

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO



Dissertação de Mestrado

**Metadados em Multimídia: Aplicações e Conceitos em
MPEG-7**

Autor: Luis André Villanueva da Costa Ferreira

Orientador: Prof. Dr. Luiz César Martini

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Banca Examinadora

Prof. Dr. Luiz César Martini (DECOM-FEEC-UNICAMP)

Prof. Dr. Renato Baldini Filho (DECOM-FEEC-UNICAMP)

Prof. Dr. Yuzo Iano (DECOM-FEEC-UNICAMP)

Prof. Dr. Fujio Yamada (Universidade Mackenzie)

Campinas, SP-Brasil

Março/ 2007

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

F413m Ferreira, Luis André Villanueva da Costa
Metadados em multimídia: aplicações e conceitos em
MPEG-7 / Luis André Villanueva da Costa Ferreira. --
Campinas, SP: [s.n.], 2007.

Orientador: Luiz César Martini.
Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Videodigital. 2. MPEG (Padrão de codificação de
vídeo). 3. Metadados. 4. XML (Linguagem de marcação de
documento). 5. Vídeo interativo. 6. Sistemas multimídia. I.
Martini, Luiz César. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Título em Inglês: Multimedia metadata concepts and applications in MPEG-7.

Palavras-chave em Inglês: Vídeo manipulation, MPEG-7, Metadata, Soccer
Ontology.

Área de concentração: Telecomunicações e telemática.

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Renato Baldini Filho, Yuzo Iano, Fujio Yamada.

Data da defesa: 16/03/2007

Programa de Pós-Graduação: Engenharia Elétrica

Resumo

Neste trabalho, uma pesquisa foi realizada com o objetivo de apresentar os metadados (ou dados sobre dados) e especificamente metadados em multimídia com o padrão MPEG-7 como sendo a base. Inicialmente, a teoria envolvendo os metadados em geral e metadados em multimídia em particular são apresentadas, com o propósito de preparar o leitor para suas aplicações, especialmente em vista da explosão de mídia estar sendo atualmente disponibilizada *online*. O usuário deve ser capaz de localizar a mídia que necessita de forma eficiente dentre esse mar de informação, e esse é o principal objetivo dos metadados. Com isso em mente, o MPEG-7 foi desenvolvido pela ISO (*International Organization for Standardization* ou Organização Internacional para Padronização). O MPEG-7 é padrão aberto baseado no formato XML (*Extensible Markup Language* ou Linguagem de Marcação Extensível) que permite que aperfeiçoamentos sejam desenvolvidos continuamente, o que é muito útil considerando o fluxo de novas mídias sendo disponibilizadas aos usuários diariamente. Este trabalho portanto apresenta os dois tipos de semânticas para metadados em multimídia, de baixo e alto nível. As semânticas de baixo nível incluem informações concretas sobre uma mídia como cores, contraste, pixels, etc. Por outro lado, as semânticas de alto nível são mais abstratas, e levam em consideração o que de fato está sendo exibido pela mídia, como pedras, árvores, pessoas, rios, etc. O intervalo semântico entre esses dois tipos de metadados apresenta um grande problema para a programação de metas. O principal objetivo deste trabalho é demonstrar como, utilizando a aplicação IBM *VideoAnnex* desenvolvida baseada no padrão MPEG-7, metadados em

multimídia de baixo e alto nível semântico podem ser criados e anexados a um arquivo de vídeo em MPEG-2. A maior parte do processo de anexação é realizada diretamente utilizando a aplicação *VideoAnnex* como será mostrado, mas diversos metadados de alto nível devem ser inseridos diretamente no arquivo XML. Como as aplicações de metadados em multimídia são tão vastas quanto a mídia que eles representam, este trabalho apresentará primeiro uma abordagem mais geral para uso com um filme qualquer, e obviamente muito menos completa, e em seguida uma abordagem mais específica utilizando como objetivo uma partida de futebol. Estudos na área de metadados em multimídia ainda estão na fase inicial, então este trabalho não apresenta uma abordagem final, mas sim uma nova opção no uso de metadados em multimídia, especialmente importantes quando se considera a atual difusão da Internet e TV Digital.

Palavras-chave: Manipulação de vídeos, MPEG-7, metadados, DDL, *Soccer Ontology*.

Abstract

In this work, a research was made in order to present Metadata (or data about data) and specifically Multimedia Metadata applications with the MPEG-7 standard as the basis. At first, the theory concerning Metadata in general and Multimedia Metadata in particular is presented, with the purpose to prepare the reader for its uses, specially with the current explosion of Media being made available online. The user must be able to locate the media he requires amongst this sea of information, and that is the Metadata main objective. With that in mind, MPEG-7 was developed by ISO (International Organization for Standardization). MPEG-7 is an open standard based on a XML (Extensible Markup Language) format that allows for improvements to be made continuously, which is very helpful considering new media is being made available to users every day. This work then presents the two types of Multimedia Metadata semantics, low level and high level. The low level semantics include concrete information about a movie or image like color, contrast, pixels, etc. On the other hand, high-level semantics take a more abstract approach, and take into consideration what is actually on a scene or image like rocks, sky, trees, actions, etc. The Semantic Gap between these two Metadata types presents a problem for most Metadata programming. This work's main objective is to present how, using the IBM Videoannex application, developed based on the MPEG-7 standard, Multimedia Metadata of both low and high level semantics can be annexed to a MPEG-2 video file. Most of the annexation process can be made directly through the Videoannex application as will be shown, but several high level Metadata need to be inserted directly into the XML file. Since Multimedia Metadata uses are as vast as the media they represent, this work will present

first an approach more general to any movie, but obviously much less complete and then a more focused approach, using a soccer match as target. Multimedia Metadata studies are still in the beginning stage, so this work doesn't present an end approach, but more of a different option to the use of Multimedia Metadata, specially important with the current spread of the Internet and Digital TV.

Keywords: Video manipulation, MPEG-7, metadata, Soccer Ontology.

Agradecimentos

‘Agradeço em primeiro lugar ao Prof. Dr. Luiz César Martini, meu orientador e companheiro no desenvolvimento desta tese, por todo o seu auxílio e colaboração durante todos os processo da criação e desenvolvimento deste trabalho.’

‘Agradeço a minha família pela paciência e ajuda durante todo o caminho da pós-graduação, e pela sua presença tanto nas dificuldades que apareceram quanto nas vitórias obtidas.’

‘Agradeço a Capes pelo auxílio financeiro, sem o qual essa tese não teria sido possível.’

‘Agradeço a todos os amigos do DECOM e de outras universidades pelo apoio, incentivo, e paciência.’

‘Por fim, agradeço ao DECOM, FEEC e UNICAMP pela oportunidade de desenvolver essa tese de mestrado.’

Sumário

Lista de Figuras.....	xi
Lista de Tabelas.....	xiii
Abreviaturas e Símbolos.....	xv
Capítulo 1 Introdução.....	1
Introdução.....	1
1.1 Motivação.....	1
1.2 Descrição do Problema.....	2
1.3 Objetivos.....	4
1.4 Metodologia.....	4
1.5 Organização da Dissertação.....	5
Capítulo 2 Metadados: Definições e Conceitos.....	7
2.1 Metadados: Histórico.....	7
2.2 Metadados: Definições.....	8
2.3 Metadados: Determinação dos Requerimentos.....	9
2.4 Metadados: Tipos Gerais.....	11
2.4.1 Metadados Específicos.....	11
2.4.2 Metadados Únicos.....	12
2.4.3 Metadados Comuns ou Gerais.....	12
2.5 Metadados: Vantagens e Desvantagens.....	13
2.6 Metadados com Semântica de Alto Nível.....	15
Capítulo 3 MPEG-7: Conceitos.....	17
3.1 MPEG-7: Histórico e Características.....	19
3.1.1 MPEG-7: Características Básicas.....	20
3.1.2 MPEG-7: Princípios Fundamentais.....	21

3.2 MPEG-7 e W3C <i>Semantic Web</i>	23
3.3 MPEG-7: Análise Técnica.....	25
3.3.1 Descritores.....	26
3.3.2 Esquemas de Descrição (DS).....	28
3.3.2.1 MDS: Tipos e Aplicações.....	29
Capítulo 4 DDL: Conceitos.....	33
4.1 DDL: Histórico.....	34
4.2 XML <i>Schema</i> : Componentes.....	35
4.3 XML <i>Schema</i> : <i>Namespace</i> e Envoltória.....	36
4.4 XML <i>Schema</i> : Definições de Tipo.....	37
4.4.1 Definições de Tipo Simples.....	37
4.4.2 Definições de Tipo Complexo.....	38
4.4.3 Definições de Tipo Derivado.....	39
4.5 XML <i>Schema</i> : Tipos de Dados.....	41
4.5.1 Tipos de Dados Primitivos Internos.....	41
4.5.2 Tipos de Dados Derivados Internos.....	42
4.6 Extensões Específicas para o MPEG-7.....	43
4.6.1 Tipos de Dados <i>Array</i> e Matrizes.....	43
4.6.2 Tipos de Dados Derivados Internos para o MPEG-7.....	44
Capítulo 5 Aplicação de metadados.....	46
5.1 Ferramenta Utilizada: IBM <i>VideoAnnex Annotation Tool</i>	47
5.1.1 Instalação do Programa.....	47
5.1.2 IBM <i>VideoAnnex</i> : Funcionamento.....	48
5.1.3 Exemplo de Anotação Utilizando o <i>VideoAnnex</i>	51
5.2 Aplicando Metadados de Semântica Elevada: A <i>Soccer Ontology</i>	54
5.2.1 Início do Desenvolvimento da <i>Soccer Ontology</i>	55
5.2.2 Cabeçalho da <i>Soccer Ontology</i>	57
5.2.3 <i>Soccer Ontology</i> : Desenvolvimento dos Metadados.....	60
5.2.4 Aplicando a <i>Soccer Ontology</i>	65
5.3 Inserindo Metadados Diretamente no Arquivo MPEG-7.....	71
5.3.1 Desenvolvimento dos Metadados no Ambiente XML.....	71

5.4 Conclusões.....	77
Capítulo 6 Conclusões e Trabalhos Futuros.....	79
6.1 Conclusões.....	79
6.2 Trabalhos Futuros.....	81
Referências Bibliográficas.....	83
Anexo I.....	88
Anexo II.....	96
Artigos Submetidos.....	106

Lista de Figuras

Capítulo 3

Figura 3.1. Exemplo de Cadeia MPEG-7.....	20
Figura 3.2. Esquema dos Elementos do MPEG-7.....	26
Figura 3.3. Distribuição dos MDS em MPEG-7.....	30
Figura 3.4. <i>Generic Audiovisual DS</i>	32

Capítulo 5

Figura 5.1. Tela Principal do IBM <i>VideoAnnex Annotation Tool</i>	48
Figura 5.2. Painel <i>Shot Annotation</i>	49
Figura 5.3. Abertura de Vídeo no <i>VideoAnnex</i>	52
Figura 5.4. Visualização do Vídeo no <i>VideoAnnex</i>	53
Figura 5.5. Visualização das Anotações.....	53
Figura 5.6. Menu Lexicon do <i>VideoAnnex</i>	56
Figura 5.7. <i>VideoAnnex</i> utilizando a <i>Soccer Ontology</i>	65
Figura 5.8. Marcação de uma Partida de Futebol utilizando a <i>Soccer Ontology</i>	66
Figura 5.9. Descrevendo um Gol com a <i>Soccer Ontology</i>	68
Figura 5.10. Marcando um Evento Estático: Jogador Expulso.....	69

Lista de Tabelas

Capítulo 3

Tabela 3.1 Lista de Descritores em Uso Atualmente.....	27
--	----

Abreviaturas e Símbolos

AI	Artificial Inteligence
CD	Compact Disc
CRUD	Create, Read, Update and Delete
DDL	Description Definition Language
DL	Digital Library
DS	Description Scheme
DSTC	Distributed Systems Technology Center
DVD	Digital Video Disc
IANA	Internet Assigned Numbers Authority
IBM	International Business Machines Corporation
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
KR	Knowledge Representation
HDTV	High Definition Television
HTML	HyperText Markup Language
MDS	Multimedia Description Schemes
MIME	Multipurpose Internet Mail Extensions
NTSC	National Television Systems Committee
XML	Extensible Markup Language
XMLNS	XML Namespace
URI	Uniform Resource Identifier

Capítulo 1 Introdução

1.1 Motivação

Metadados como o nome já indica, são descrições inseridas dentro dos próprios arquivos de mídia, desenvolvidos com o objetivo de facilitar a busca e a recuperação desses arquivos, e são manipulados para se conformar com mudanças no arquivo que descrevem.

O desenvolvimento de metadados para a aplicação com arquivos digitais de mídia é um processo em atual expansão, a distribuição de arquivos de mídia digital pela Internet, rádio e outros meios está cada vez está generalizada. A quantidade de mídias disponíveis ao usuário está explodindo diariamente, e a necessidade do uso de metadados para organizar essa enorme quantidade de informação é um fato consolidado.

Aplicações para o uso de metadados são infindáveis, divididas em diversas áreas como biomédica (a marcação de arquivos digitais como ultra-sonografias para futura análise), arquitetura (descrição de arquivos delineando a construção de edifícios), e a mais básica aplicação, a simples marcação de um arquivo de vídeo para o usuário comum.

O uso de metadados na transmissão, recuperação, manipulação e descrição de arquivos de mídias permite a localização mais rápida e eficiente da informação buscada, por incluir a descrição da mídia dentro do próprio arquivo digital, evitando também problemas como subjetividade no campo de busca.

Com o desenvolvimento de diversas ferramentas que se baseiam no conceito de metadados e os utilizam na descrição e distribuição de arquivos de mídia, como o IBM *Annotation VideoAnnex*, que se baseia no formato MPEG-7 da ISO e o *Semantic Web*, o

desenvolvimento de metadados continua se intensificando. Como a quantidade de mídia disponível continua aumentando exponencialmente, a quantidade de metadados também aumenta e é necessário de que eles sejam corretamente programados e distribuídos, seguindo um mesmo padrão de desenvolvimento que permita que eles sejam ligados um aos outros. O padrão mais aceito e utilizado nesta tese é exatamente o MPEG-7, desenvolvido pela ISO com esse objetivo.

Os metadados obviamente também têm problemas, que incluem seu caro desenvolvimento, a grande quantidade de tempo necessária para serem programados e inseridos nos arquivos de mídia, além de sua programação ser demasiada complicada para o usuário comum utilizar. Alternativas buscando minimizar esses problemas estão sendo desenvolvidas.

Por fim, os metadados são as melhores alternativas existentes para a descrição de arquivos de mídia e distribuição de arquivos de mídia, e portanto o desenvolvimento desta tese se baseia neles.

1.2 Descrição do problema

Um dos principais problemas na utilização de metadados para a marcação de arquivos de mídia é a *Semantic Gap*, ou intervalo semântico. A *Semantic Gap* indica a diferença entre a descrição de um objeto e a realidade deste objeto, ou seja, quando a descrição é inadequada para descrever todas as nuances de um determinado objeto. Um exemplo comum desse problema é a descrição de obras de arte em que a menor diferença entre luz e sombra é importantíssima, mas a descrição não consegue demonstrar essas diferenças. Esse

problema é acentuado, pois o modo de metadado mais comumente utilizado é o de semântica de nível baixo, que será apresentado com mais detalhes no próximo capítulo.

Esse metadado é reconhecido automaticamente pela máquina, e é muito mais fácil de programar, pois baseia-se em informações concretas sobre um vídeo, como cor, contraste, resolução, etc. Mas como se percebe, utilizar esse tipo de metadados não é remotamente suficiente para descrever e permitir que um arquivo de vídeo seja localizado e recuperado, portanto o foco de desenvolvimento se direcionou para os metadados com um alto nível semântico [2].

Os metadados de alto nível semântico se focam em uma aproximação abstrata, ou seja, eles procuram descrever o que de fato é mostrado em um determinado vídeo além de pixels, ou seja, árvores, pessoas, animais, nuvens, etc.

Com o desenvolvimento de metadados de semântica de nível elevado, criou-se um novo problema. Como é possível se criar um metadado para se descrever praticamente tudo que existe, surgiu um mar de metadados genéricos que não são indicados a nenhuma aplicação mais avançada. Para minimizar esse problema, foi aplicado o conceito de ontologias, que recomenda o desenvolvimento de metadados destinados a uma aplicação específica, o que permite que esses metadados sejam utilizados por usuários em áreas especializadas, como biomédicas, arquiteturas, etc. Este trabalho portanto se baseia no desenvolvimento de ontologias, formadas por metadados de alto nível semântico, apesar de seu trabalhoso e muitas vezes custoso uso.

1.3 Objetivos

O desenvolvimento de metadados não direcionados a nenhuma aplicação específica impede seu uso em uma aplicação mais avançada, devido a necessidade de semântica de alto nível. Para minimizar esse problema, o foco atual são as ontologias, conjuntos de metadados desenvolvidos para uma aplicação particular.

Este trabalho tem como objetivo apresentar o desenvolvimento e utilização de uma ontologia em particular, criada para a descrição em alto nível semântico de uma partida de futebol. Essa ontologia, denominada de *Soccer Ontology*, permite aos usuários que descrevam em detalhes, partidas deste esporte, de modo rápido e fácil. A *Soccer Ontology* permite que os usuários se concentrem apenas nas descrições, já que todos os principais metadados necessários para a marcação de uma partida já estão inclusos. Caso o usuário necessite de descrições extras, ele pode acrescentá-los à ontologia pelo menu Lexicon.

A *Soccer Ontology* foi desenvolvida para trabalhar em conjunto com o *IBM VideoAnnex Annotation Tool*, ferramenta gratuita disponível para a marcação de vídeos utilizando o padrão MPEG-7, que também é a base da *Soccer Ontology*.

1.4 Metodologia

O desenvolvimento da *Soccer Ontology* é baseado em informações teóricas sobre o MPEG-7 e os metadados em si, e portanto essas informações serão apresentadas no início do trabalho, nos capítulos 2 e 3. Após a apresentação do MPEG-7 e seu funcionamento, e como metadados em geral são desenvolvidos, o trabalho apresenta o DDL, *Description*

Definition Language ou linguagem de definição de descrições.

O DDL é uma linguagem de programação baseada no formato XML desenvolvida para uso pelo padrão MPEG-7, sendo a linguagem de programação utilizada para o desenvolvimento da *Soccer Ontology*. Portanto o DDL é apresentado e suas principais rotinas são descritas.

Com a base teórica informada nos 3 primeiros capítulos, começa o desenvolvimento da ontologia em si. Em primeiro lugar, o desenvolvimento da ontologia é realizado passo a passo para facilitar ao leitor o entendimento do procedimento feito. São incluídos comentários e informações para comparação com outras ontologias e o desenvolvimento de ontologias extras caso seja necessário.

Após o desenvolvimento da *Soccer Ontology* no ambiente DDL, são feitos testes para ilustrar seu funcionamento. Todos os testes foram feitos no *IBM VideoAnnex Annotation Tool*, que é a ferramenta que foi utilizada como base durante o desenvolvimento da *Soccer Ontology*. Por fim, é indicado ao usuário como criar novos metadados para utilização na *Soccer Ontology*, demonstrando sua flexibilidade, e caso necessário inseri-los diretamente no arquivo MPEG-7, em vista das limitações do *IBM VideoAnnex*.

1.5 Organização da dissertação

O capítulo 2 deste trabalho trata dos conceitos envolvidos com a criação, utilização, requerimentos e em especial os tipos de metadados existentes. As diferenças

entre metadados de baixo a alto nível semântico são ilustradas, e por que os metadados de alto nível semântico foram escolhidos para uso nessa tese. Também é introduzido o conceito de ontologias.

O capítulo 3 se foca no padrão MPEG-7, seu histórico, funcionamento, as características principais do formato e suas diferenças para o W3C *Semantic Web*. Neste capítulo também é feita uma análise técnica do padrão MPEG-7, com a apresentação dos esquemas de metadados ou descritores e os metadados em si, e como eles trabalham dentro do formato.

O capítulo 4 é baseado na linguagem de programação DDL, *Description Definition Language*, ou linguagem de definição de descrições, utilizada neste trabalho para a programação de metadados. Neste capítulo, é apresentado o histórico da linguagem, seu desenvolvimento a partir do formato XML, suas principais ferramentas, os tipos de definição utilizados pelo DDL, e suas extensões desenvolvidas especialmente para o MPEG-7.

O capítulo 5 começa a parte prática deste trabalho, introduzindo o desenvolvimento da *Soccer Ontology*, como ela foi programada, testes de seu uso, a sua utilização com a *IBM VideoAnnex Annotation Tool*, como alterar a ontologia, etc.

O Capítulo 6 por fim apresenta as conclusões chegadas sobre este trabalho, incluindo sugestões de trabalhos futuros.

Capítulo 2 Metadados: Definições e Conceitos

Este capítulo define metadados, suas funções, conceitos e aplicações. Os metadados especialmente aplicados em multimídia são um dos principais temas desta tese.

2.1 Metadados: Histórico

Metadados tem se tornado um assunto muito discutido desde o ano 2000, mas porque esse assunto é tão debatido? Porque os metadados podem ser a solução para vários problemas que existem atualmente para gerenciar a enorme quantidade de informação disponível para qualquer usuário com um computador, desde a década de 90 [6]. Esses problemas incluem:

- Localizar os dados que o usuário necessita em meio ao oceano de informações disponíveis.
- Organizar esses dados em um banco de dados de fácil acesso.
- As diferenças entre o dado original e aquele após as alterações efetuadas pelo usuário, etc.

Toda essa informação disponível não tem nenhum valor se o usuário não é capaz de encontrar e utilizar aquilo que necessita [4]. Como se produz cada vez mais conteúdo de mídia, mais complexo vai se o problema de gerenciar todo este conteúdo.

Para localizar de forma eficiente a informação desejada, é absolutamente necessário ter-se as informações sobre o conteúdo buscado [1].

Atualmente, se utilizam *websites* de busca utilizando texto para localizar multimídia, com uma taxa muito baixa de acerto [3]. O usuário insere palavras no *browser* sobre a mídia desejada, e o mecanismo de busca tenta localizar a mídia entre milhões de outros possíveis resultados. Isso gera vários problemas, entre eles dois especialmente sérios. Primeiro, é um processo caro e complexo, e quanto mais conteúdo disponível, mais complexo o processo se torna. Segundo, localizar a mídia depende das palavras inseridas pelo usuário, que são obviamente subjetivas. Esses problemas não são um assunto novo, e para resolvê-los, metadados são a saída mais óbvia, pois eles permitem que as descrições dos conteúdos sejam automaticamente geradas, que elas sejam mais objetivas e independentes do conceito que o usuário utilizou no momento da busca, e estarão inseridas dentro do próprio conteúdo de mídia, facilitando sua localização e manipulação [2].

2.2 Metadados: Definições

A palavra meta, que em grego significa “sobre”, portanto metadados seriam dados sobre dados. Essa definição, entretanto, é muito genérica, pois o “sobre” neste caso está aberto a diversas interpretações, especialmente para o usuário dos dados.

Uma definição de metadados utilizada por diversas organizações atualmente é:

“Metadados representam informações descritivas em detalhes, como nomes dos elementos envolvidos na mídia, tamanho da mídia, definições, etc. em relação aos campos de dados utilizados [4]”.

Obviamente, essa definição também não consegue incluir tudo que envolve os metadados, que por sua própria natureza são flexíveis e a escolha dos elementos descritos fica sempre sob a responsabilidade de um especialista. Para que se obtenha o máximo desses descritores, é necessário cada vez mais a inclusão de dados de maior complexidade. Esses metadados mais complexos são os com semântica de alto nível, que são descritos na seção 2.5.

Os dados de mídia são inúteis sem os metadados, pois se os dados não podem ser encontrados, é como se eles não existissem [3]. O desenvolvimento de metadados deve levar em consideração portanto as cinco questões formuladas pela *Database Solutions Inc.* [4].

- 1 - Quais são os dados que se possui?
- 2 – O que eles significam?
- 3 – Onde eles se encontram?
- 4 – Como eles chegaram lá?
- 5 – Como se pode obtê-lo?

Para o bom desenvolvimento de metadados, essas perguntas devem ser respondidas. A seguir, se entra em mais detalhes sobre como funciona o desenvolvimento dos metadados.

2.3 Metadados: Determinação dos Requerimentos

O primeiro procedimento para o desenvolvimento de metadados é adotar o seguinte processo de requerimentos [4]. Eles são apresentados a seguir:

- 1 – Identificação dos receptores dos metadados: Neste passo, ocorre a identificação de

todos os potenciais usuários e geradores dos metadados em questão. Esses usuários ou geradores podem ser pessoas, aplicações, bancos de dados de metadados, etc.

2 – Listar as necessidades de cada usuário: Realizar a análise do que os receptores realmente precisarão.

3 – A fonte do metadado deve ser determinada: Através da determinação da fonte de cada requerimento, o metadado pode ser nomeado de acordo com a necessidade. Outros requerimentos também são determinados neste passo.

4 – Planejamento da arquitetura preliminar e o alcance do metadado: Baseado nos resultados dos passos anteriores, uma caixa (*Box*) é gerada em torno dos metadados a serem incluídos na solução inicial.

5 – Recategorização dos metadados: Executa-se o refinamento dos requerimentos dos metadados para incluir categorias de utilização baseadas na necessidade de que todos os metadados sejam específicos, únicos ou comuns.

6 – Metamodelos: O metadado é organizado e identificado, similarmente aos modelos de dados, auxiliando na determinação do local de armazenagem final do metadado.

7 – Consideração de acesso e exibição do metadado: Determinação de como o metadado será acessado utilizando o CRUD, ou *Create, Read, Update, Delete* ou Cria, Lê, Atualiza, Apaga, que indica todas as funções que devem ser implementadas em uma base de dados para que esta seja considerada completa. É determinado como ele será mantido em sincronia caso necessário, e quais mecanismos de exibição serão utilizados.

8 – Antecipação do processamento do metadado: O tipo de utilização (Web, bancos de metadados, aplicações customizadas, etc.) determina os requisitos restantes, que podem ser quaisquer não planejados desde o início do processo, mas que são importantes para o processamento mecânico.

9 – Considerar os requerimentos que não são ligados diretamente ao metadado: A última parte do processo onde ocorre a avaliação dos outros requisitos. Tem-se então o desenvolvimento do metadado, que pode ser uma semântica de baixo ou alto nível.

No próximo capítulo, mais detalhes sobre o processo serão apresentados.

2.4 Metadados: Tipos Gerais

Os metadados devem ser ordenados em grupos de dados similares para facilitar sua localização [8]. Esta categorização leva em conta os requisitos específicos de cada metadado e os agrupam. Existem várias possíveis divisões de metadados, por exemplo o conteúdo onde o metadado está sendo usado (Ex: Imagens, programas, etc.), Bancos de Dados, etc. A divisão apresentada neste capítulo se baseia na categoria dos metadados, e é a mais geral possível.

Os três tipos de metadados são específicos, únicos e comuns ou gerais.

2.4.1 Metadados Específicos

Metadados específicos são qualquer metadado que será utilizado por apenas uma categoria de usuários, sendo precisa e específica para esta categoria [3]. Metadados específicos também se originam de uma categoria que espelha a própria categoria dos usuários. Por fim, metadados específicos são todos os metadados separados por seus propósitos específicos, tanto na forma de design dos metadados quanto ao seu uso. Outros nomes para esta categoria são metadado especialista ou metadado aplicado.

Alguns exemplos desta categoria são: metadados para gerenciamento de configurações específicas, metadados utilizados para relato de problemas no desenvolvimento de uma aplicação, etc.

Mais uma vez, para definir um metadado como específico, é necessário verificar a origem do metadado e os usuários, sejam pessoas, aplicações, etc. [1].

2.4.2 Metadados Únicos

Os metadados únicos também são desenvolvidos para um usuário específico, entretanto sua fonte de origem não espelha (ou seja, não se origina no mesmo componente da arquitetura) a categoria dos usuários. Os metadados únicos podem ter múltiplas fontes, mas essas fontes devem ser sempre diferentes entre si [4]. Por essa distinção, os metadados únicos são mais comuns que os específicos.

Em situações de dúvida se um metadado é de fato específico ou único, recomenda-se realizar uma segunda verificação dos requerimentos citados na seção 2.3.

Essa verificação, confirmando a origem de fato do metadado, indica com segurança se é específico ou único.

2.4.3 Metadados Comuns ou Gerais

Esta categoria inclui todos os metadados que não se enquadram em uma das duas categorias anteriores. Eles podem ser utilizados por todas as categorias de usuários finais. Apesar de ser chamado comum, metadados que podem ser usados por todas as categorias

de usuários são raros [2]. Portanto, metadados comuns podem ser considerados simplesmente aqueles que não se enquadram nas categorias único e específico.

Um exemplo de metadado comum é uma aplicação que recupera o nome completo de um usuário de um banco de dados utilizando metadados para efetuar a localização do nome.

Neste capítulo também foi comentado a origem dos metadados, as definições utilizadas, o início de como metadados são desenvolvidos. As questões finais a serem respondidas são: Realmente se precisam dos metadados, especialmente dos metadados em multimídia que são o tema desta tese? Quais são suas verdadeiras vantagens em relação a outras tecnologias?

2.5 Metadados: Vantagens e Desvantagens

O contínuo aumento de mídia disponível aos usuários gerou a necessidade de um meio de descrever a mídia com mais detalhes para facilitar sua localização ao usuário. Mas os metadados são realmente melhores que outras tecnologias para este objetivo?

Os fatos são: É necessário que a descrição do conteúdo em questão seja de forma semântica e deve ser compreensível para a máquina [2]. Esses dois problemas são resolvidos pelos metadados, que descrevem o conteúdo de forma semântica (baseada em contexto). Metadados são menos subjetivos que outras formas de descrição de mídia, estão inseridos dentro da própria mídia e são projetados, levando em conta que os mesmos devem ser reconhecidos mecanicamente [8].

Os metadados são extremamente complexos, seu desenvolvimento tende a ser caro e inserir metadados em um arquivo de mídia aumenta muito sua complexidade. Metadados

de baixo nível que só levam em conta as qualidades físicas de uma imagem, arquivo de vídeo ou áudio, não são suficientes para os usuários atuais devido à necessidade de se trabalhar com informações abstratas sobre o conteúdo [10]. O maior problema dos metadados de baixo nível é a diferença entre o que o usuário percebe (seus limites visuais, ou seja, diferenças de cores, contraste, sombras, etc. que o usuário não é capaz de perceber através dos seus olhos) e o que a máquina percebe, trabalhando em bits. Isso é um dos problemas que gera a necessidade pelos metadados com semântica de alto nível [11].

Desenvolveram-se então formatos que realizavam os seguintes objetivos na manipulação de metadados para multimídia, disseminando seu uso.

- Facilitar o desenvolvimento e a codificação de metadados para multimídia.
- Possuir mecanismos que estabeleçam contexto em um componente de mídia.
- Dispor de esquemas de descrição flexíveis levando em consideração o contexto em que são usados.
- A semântica também deve ser sempre mutável pois a criação da descrição da mídia é um processo dinâmico e sempre deve ser aperfeiçoado.

Levando essas necessidades em consideração, dois formatos foram criados para a descrição de conteúdo multimídia legível por máquinas: O W3C *Semantic Web* (Rede Semântica W3C), e o da ISO (*International Organization for Standardization* ou Organização Internacional para Padronização) denominado *Multimedia Content Description Interface* (Interface de Descrição de Conteúdo de Multimídia) *MPEG-7* [6] que é o formato utilizado nesta tese.

2.6 Metadados com Semântica de Alto Nível

Metadados com semântica de alto nível ou metadados considerando contexto são o caminho necessário para o desenvolvimento dos metadados. Metadados com semântica de alto nível são metadados que descrevem o conteúdo de mídia com mais detalhes [14], uma vez que a própria mídia está se tornando mais e mais complexa.

Os metadados com semântica de baixo nível têm vários problemas, tais como a descrição de baixo nível de um arquivo de mídia se relaciona às características físicas da mídia; o grande gap entre o que o usuário percebe na mídia e o que a máquina codifica, decodifica e localiza, chamada gap sensorial e de dados; e o gap semântico [10], que é a diferença entre o que o metadado indica (informações de baixo nível) sobre a mídia procurada pelo usuário, e o que o usuário precisa para facilitar a localização e utilização da mídia desejada.

Um exemplo de gap semântico pode estar na busca pela imagem de um dos quadros de Da Vinci. Seu desenho cheio de detalhes, variações de cores e diferenças de contraste pode causar confusão na busca se o metadado não for específico para o conteúdo, gerando resultados incorretos.

Com a necessidade do desenvolvimento de tecnologias que utilizem metadados relacionados ao conteúdo da mídia, vários formatos foram desenvolvidos como o W3C *Semantic Web* [20] e o MPEG-7 da ISO [12].

Cada um desses empreendimentos considera o processo de atribuição de metadados a um componente de mídia como um processo terminal, onde um usuário humano auxiliado por um sistema (um computador na maioria dos casos) efetua as computações necessárias,

designa o metadado aos itens de mídia e então o material é distribuído para outros usuários.

Essa abordagem gera um novo problema, pois é na sua maior parte estática. Ela não reflete o processo contínuo de interpretação e compreensão dos componentes sintáticos e semânticos da mídia. Mídia continua sendo desenvolvida para um propósito específico, e é necessário é que os metadados sejam flexíveis, o que ainda não é feito com frequência devido à ausência da tecnologia necessária.

Para a resolução desse problema, necessita-se de uma maior discussão entre usuários e criadores sobre um modelo de desenvolvimento de metadados com base semântica e reconhecíveis por máquinas mais acessível aos usuários em geral.

Essa tese trata do MPEG-7 da ISO, formato mais popular atualmente na criação e implementação de metas e como este é utilizado para o desenvolvimento de metadados semânticos. No próximo capítulo, serão apresentados mais detalhes sobre o MPEG-7.

Capítulo 3 MPEG-7: Conceitos

O valor da informação depende de quão facilmente ela pode ser encontrada, recuperada, acessada e gerenciada. Com o crescimento cada vez maior da quantidade de mídia disponível aos usuários, um padrão para a busca e uso da mídia se faz necessário. Um desses padrões é o MPEG-7 (*Motion Picture Experts Group* ou Grupo de Especialistas em Imagens em Movimento), desenvolvido pela ISO.

O MPEG-7 é um padrão ISO/IEC, que também desenvolveu os padrões MPEG-1, MPEG-2, MPEG-4 e MPEG-21 [12]. Ele foi aceito no ano de 2001, com o nome formal de Interface de Descrições de Conteúdos de Multimídia e tem como objetivo a descrição de conteúdos de mídia utilizando a anexação de metadados ao conteúdo de mídia. O MPEG-7 fornece um conjunto de ferramentas descritivas que podem ser utilizadas para a descrição de vários tipos de arquivos de multimídia. Ele trabalha em conjunto com os outros formatos MPEG, adicionando funcionalidades a eles. [8] Com os antigos formatos MPEG, tem-se os arquivos de mídia e com o MPEG-7, pode-se localizá-los com o auxílio de metadados.

O mais recente formato ISO, o MPEG-21, trabalha gerando um padrão onde os diferentes elementos formam uma infra-estrutura, permitindo que a entrega e consumo de conteúdos de mídia trabalhem em conjunto. O MPEG-21 também tem como objetivo diminuir a distribuição ilegal de arquivos entre usuários na web (*file-sharing*). O padrão MPEG-21 ainda está em fase experimental e seu uso ainda não está disseminado.

O assunto deste capítulo é o padrão MPEG-7, suas características, funcionamento, aplicações, e sua relação com os metadados. A utilização do MPEG-7 para este fim é o principal foco deste trabalho.

3.1 MPEG-7: Histórico e Características

Fez-se necessário o desenvolvimento de um padrão de descrições que, utilizando ferramentas desenvolvidas para o padrão, fossem objetivas, mídia-específicas e reconhecidas pela máquina. Esse desenvolvimento foi iniciado pelo grupo MPEG no mês de julho de 1996, quando foi decidida na reunião em Tampere, Finlândia a criação de um projeto de padronização conhecido publicamente como MPEG-7 [12]. Ele é formalmente chamado de Interface de Descrição de Conteúdos de Multimídia (ISO/IEC 15938).

O principal objetivo do MPEG-7 foi a criação de um modo para a descrição de vários tipos de multimídia, seja em partes ou trabalhos completos, não importando o seu formato e o local de armazenagem. Assim era facilitado o acesso rápido e o gerenciamento da mídia necessária ao usuário.

As descrições (os metadados) utilizadas pelo MPEG-7 podem ser em forma de texto (anotações sobre a mídia, nome dos criadores, etc) e dados não-textuais (informações estatísticas, informações sobre os pontos de filmagem da mídia, etc.). O MPEG-7, assim como os demais formatos MPEG desenvolvidos pela ISO (MPEG-1, MPEG-2 e MPEG-4), definem uma representação padronizada de multimídia devendo satisfazer um conjunto de regras bem definidas.

A principal diferença entre o MPEG-7 e os padrões anteriores é que o MPEG-7, ao invés de representar a mídia em si (o que já era feito pelos outros padrões), representa o conteúdo sobre a mídia, os metadados ou descritores [21]. Os padrões podem se interconectar, apesar dos requisitos serem muito diferentes entre si. Por exemplo, é possível trabalhar com um arquivo de vídeo codificado em MPEG-4 e inserir metadados

para descrevê-lo utilizando o MPEG-7. Mostrar como esse processo é feito está na parte prática desta tese.

O MPEG-7 também foi desenvolvido com o objetivo de criar um padrão que facilitaria a troca de descrições entre diferentes sistemas, evitando problemas de interoperabilidade [2]. O padrão também permite que ferramentas desenvolvidas por diferentes geradores de conteúdo sejam utilizadas em conjunto.

Com o desenvolvimento do MPEG-7, o principal resultado obtido é esta interoperabilidade, o que permite a geração de mídia de baixo custo utilizando o padrão e seus serviços, principalmente levando em conta o aumento cada vez maior do volume de mídia disponível.

3.1.1 MPEG-7: Características Básicas

Para criar os metadados, o MPEG-7 oferece um conjunto de elementos de metadados audiovisuais, e sua estrutura e funcionamento entre si. Esse conjunto de metas é definido na forma de *Descriptors* e *Description Schemes*. O MPEG-7 possui a capacidade de descrever as características semânticas, estruturais e físicas de qualquer arquivo de mídia.

A cadeia de processos do MPEG-7 inclui a extração da informação, a descrição em si, e a aplicação fazendo uso da descrição [6]. A parte da descrição (metadados) é onde se concentra o MPEG-7 e esta tese. Não é incluído no padrão a recuperação da mídia ou a ferramenta de busca para localizar a mídia. Essas aplicações ficam sob a responsabilidade dos criadores de software, que utilizando o MPEG-7, desenvolvem as aplicações para ele. Algumas aplicações já existentes são o TU Berlim Analisador de Áudio MPEG-7, o

Codificador de Áudio MPEG-7 e a IBM *Videoannex Annotation Tool* [19], que cria documentos MPEG-7 para arquivos de vídeo descrevendo a estrutura e fornecendo palavras-chave. A IBM *Videoannex Annotation Tool* é a ferramenta que será utilizada para a inserção de metadados em arquivos de mídia nesta tese. A figura 3.1 apresenta um exemplo de uma cadeia MPEG-7, que é explicada em mais detalhes abaixo:

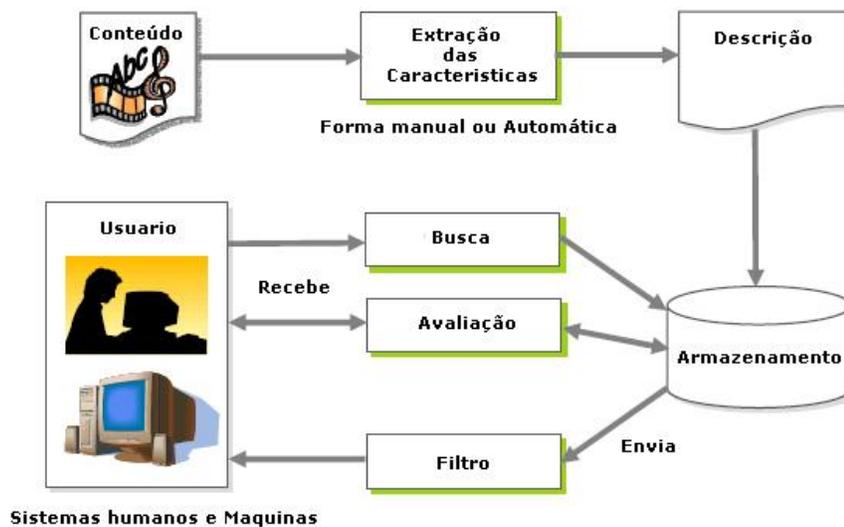


Figura 3.1. Exemplo de cadeia MPEG-7. ([12], Editado)

A descrição, como mostrado na figura 3.1, é obtida do conteúdo de mídia por extração manual ou automática [12]. Os metadados são armazenados para futuro uso. No cenário “recebe”, um conjunto de descrições que está de acordo com o pedido do usuário, é enviado a ele para análise. No cenário “envia”, um agente automático filtra as descrições e executa ações programadas, como mudar de canal de TV ou gravar uma determinada transmissão.

A descrição MPEG-7 pode ser fisicamente inserida com a mídia, ou na mesma seqüência de dados ou no mesmo local de armazenagem. É possível também que os metadados sejam localizados em outro ponto na mesma rede. Neste caso, um mecanismo

que ligue a mídia e os metadados é necessário. Um ponto interessante é que o MPEG-7 não é dependente dos outros padrões MPEG, podendo ser usado em outras aplicações como em um filme analógico.

3.1.2 MPEG-7: Princípios Fundamentais

Quando do desenvolvimento do MPEG-7, foi decidido que certos princípios fundamentais seriam seguidos por todos os envolvidos. Esses princípios expressam a própria visão por trás do MPEG-7 [23], eles guiaram todo o processo de formação dos requisitos e ferramentas de utilização para ele.

Os princípios fundamentais, portanto, são:

1 – Base Larga de Aplicação: MPEG-7 será aplicável ao conteúdo associado a qualquer domínio de aplicações, sejam geradas em tempo real ou não. O MPEG-7 não será destinado para apenas um tipo específico de aplicação. Além disso, o conteúdo pode ser armazenado e ser feito disponível pela Internet, off-line ou por meio de *stream*.

2 – Relação com o Conteúdo: O MPEG-7 permitirá a criação de descrições [35] a serem usadas:

- Isoladamente, por exemplo, fornecendo um básico sumário do conteúdo.
- Multiplexado com o próprio conteúdo, em casos como quando transmitido em conjunto com a mídia.
- Ligado a uma ou mais versões do conteúdo, por exemplo em mídias baseadas on-line.

3 – Grande Disposição de Tipos de Mídia: O MPEG-7 deverá considerar uma grande variedade de tipos de mídia, como fala, imagens, modelos 3-D, etc. Ferramentas descritivas

para arquivos textuais não serão necessárias, pois o MPEG-7 tem o objetivo de se concentrar em arquivos audiovisuais. Portanto, serão usadas ferramentas existentes para estes arquivos, como XML [26] (*Extensible Markup Language* ou Linguagem de Marcadores Extensiva).

4 – Independência de Mídia: O MPEG-7 deverá ser aplicável independentemente do meio que fornece o conteúdo. Mídia neste caso pode incluir filme fotográfico, fitas, CDs, DVDs, um disco rígido, uma transmissão pela Internet, etc.

5 – Baseado em Objetos: MPEG-7 deverá permitir a descrição baseada em objetos de conteúdo. O conteúdo pode ser representado ou descrito como uma composição de objetos multimídia e deverá ser possível acessar independentemente os dados descritivos de cada objeto específico presente no conteúdo de mídia.

6 – Independência do Formato: O MPEG-7 deve ser aplicável independentemente do formato de representação do conteúdo, seja analógico ou digital, comprimido ou descomprimido. Esses formatos incluem NTSC (*National Television Standards Committee*), MPEG-1, MPEG-2, MPEG-4, etc. Mas, como o MPEG-7 e o MPEG-4 são padrões baseados em análise de objetos, é muito interessante utilizá-los em conjunto [34]. Eles se complementam muito bem e permitem que diversas aplicações sejam desenvolvidas.

7 – Nível de Abstração: O MPEG-7 deverá incluir capacidades de descrição com diferentes níveis de abstração, de nível baixo a descrições de nível alto que apresentam significado semântico [15], como foi discutido na seção 2.6. Normalmente, as descrições de nível baixo podem ser extraídas da mídia automaticamente, enquanto que as de nível alto precisam ser extraídas manualmente ou de forma semi-automática. É importante lembrar que descrições de nível alto normalmente se baseiam em informações de nível baixo.

8 – Extensibilidade: O MPEG-7 deverá permitir a extensão do conjunto central de

ferramentas de descrição de uma forma padronizada. É reconhecido que um padrão como o MPEG-7 nunca será capaz de resolver todas as necessidades de cada aplicação possível, então deve ser possível aprimorar o padrão de forma a permitir o máximo de interoperabilidade possível.

Esses são os princípios fundamentais do MPEG-7, que direcionaram todo o desenvolvimento desse padrão. Mas quais são as principais diferenças entre o MPEG-7 e o outro padrão desenvolvido com o mesmo objetivo, descrição de mídia utilizando metadados, o W3C *Semantic Web*? Essa comparação é o tema da próxima seção.

3.2 MPEG-7 e W3C *Semantic Web*

Esta seção apresenta uma comparação elementar entre o W3C e o MPEG-7, pois uma comparação aprofundada levando em consideração todas as características de cada padrão não é o objetivo desta tese. Outros trabalhos já foram desenvolvidos sobre este específico tema.

Para iniciar uma comparação, é necessário considerar os objetivos quando do desenvolvimento do W3C *Semantic Web* [20]. Problemas na utilização de metadados envolvem todas as formas de informação, em especial a DL (*Digital Library* ou Biblioteca Digital) e a KR (*Knowledge Representation* ou Representação de Conhecimento), além da Inteligência Artificial (AI) que interpreta, manipula e gera a multimídia. A idéia do W3C durante o desenvolvimento do *Semantic Web* foi desenvolver um padrão que considerasse o DL e o KR aplicado a rede. O MPEG-7 incorpora aspectos da DL, KR e AI.

A comunidade usuária do DL vê os metadados como um modo de catalogar e recuperar

informação em grandes bancos de dados, e seus padrões são direcionados para estes fins. O principal exemplo é o Dublin Core [24], que padroniza o conjunto de metadados que aparece mais freqüentemente nas diversas bibliotecas existentes. A padronização inclui o título, assunto e data da criação.

Já a comunidade usuária do KR tem um foco centrado em representar o conteúdo implícito em vez de descrever o documento contendo o conteúdo, como o *Digital Library*. Para a comunidade KR, o desenvolvimento de uma infra-estrutura robusta para a adição de metadados a documentos disponíveis na rede é o principal objetivo.

Apesar das diferenças entre as comunidades, é obvio que várias aplicações necessitam de elementos de ambos os padrões. Aqui, tem-se o conceito de Ontologia [16] como define Schreiber, uma “especificação de uma conceituação”. Ou seja, uma definição geralmente utilizada dos objetos e conceitos de uma determinada área. Ao se comprometer com uma determinada ontologia, os usuários podem fazer anotações com mais consistência e de forma sistemática, além da possibilidade do uso do conhecimento anterior sobre a ontologia em conjunto com os metadados.

O MPEG-7 considera o uso dos metadados de forma similar ao W3C, mas como os documentos MPEG-7 são geralmente arquivos de Multimídia complexos, seu objetivo durante o desenvolvimento foi gerar uma interface comum para a descrição de arquivos de mídia. Isso significa representar informação sobre o conteúdo, mas não o próprio conteúdo.

O padrão MPEG-7 endereça a interoperabilidade e globalização das aplicações com metadados e a flexibilidade da manipulação dos dados. Para esse objetivo, o MPEG-7 tenta lidar com as propostas das três comunidades (DL, KR e AI), enquanto que o W3C *Semantic Web* lida com apenas o DL e o KR. Tentar atender às necessidades das três comunidades é um dos maiores desafios do MPEG-7, mas essa direção foi um dos principais motivos da

escolha dele para uso nessa tese, pois a inteligência artificial que gera, manipula e interpreta a mídia é um dos principais focos no desenvolvimento de metas para aplicação em arquivos de Multimídia atualmente, especialmente em aplicações com metadados de alto nível.

3.3 MPEG-7: Análise Técnica

Tendo em mente o desenvolvimento de metadados para a descrição de elementos multimídia, as principais ferramentas do MPEG-7 são:

- 1 – Descritores (D)
- 2 – Esquemas de Descrição (*Description Schemes* ou DS)
- 3 – Linguagem de Definição de Descrição (*Description Definition Language* ou DDL)

A figura 3.2 apresenta um dos possíveis relacionamentos entre os diferentes elementos. O DDL permite a definição de descritores e esquemas de descrição, fornecendo os meios para que os descritores sejam inseridos nos esquemas. O DDL [7] também permite que certos esquemas possam ser aperfeiçoados para aplicações de uso específico. As descrições e esquemas são visualizadas como descrições no formato XML (*Extensible Markup Language* ou Linguagem de Marcação Extensível), de acordo com o padrão MPEG-7.

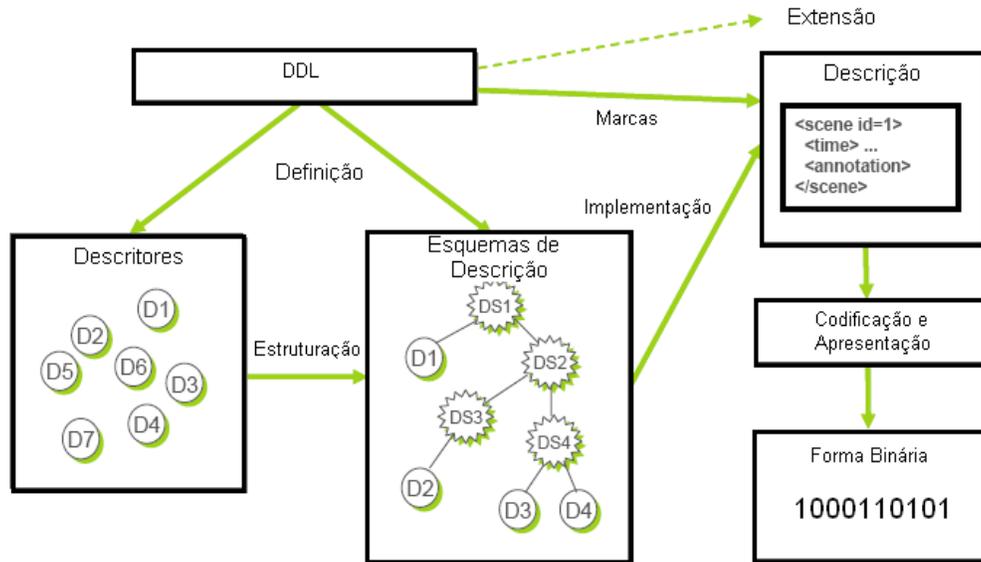


Figura 3.2. Esquema dos Elementos do MPEG-7 [12].

Esta seção apresenta mais detalhes sobre os descritores e os esquemas de descrição. O Cap. 4 se foca exclusivamente no DDL, linguagem de programação utilizada para o MPEG-7 e portanto de extrema importância para este trabalho.

3.3.1 Descritores

Os Descritores definem a sintaxe e semântica de cada mídia. Por exemplo, para a característica de brilho de uma imagem, o histograma ou texto do título é o descritor. O criador de um documento audiovisual ou uma textura em um único quadro também é um exemplo de descritor [3].

A sintaxe e semântica da mídia neste caso se referem aos seguintes tipos de informação: características audiovisuais de baixo nível, como cor, textura, brilho, etc.; características de alto nível de objetos semânticos, eventos e conceitos abstratos; processos

de gerenciamento do conteúdo; informações sobre a mídia armazenadora, etc. Como já foi citado anteriormente, os descritores ligados a características de baixo nível são recuperados automaticamente, enquanto os descritores ligados a características de alto nível devem ser recuperados com a intervenção do usuário humano. A tabela 3.1 apresenta alguns dos atuais descritores que foram incorporados para uso com o MPEG-7:

Tabela 3.1: Lista de descritores em uso atualmente.

Tipo	Característica	Descritores
Visual	Estruturas Básicas	Layout da Grid
		Histograma
	Cor	Espaço da cor
		Cor dominante
		Histograma de cor
		Quantização de cor
	Textura	Distribuição espacial da intensidade da imagem
		Textura homogênea
	Forma	Caixa limitadora de objeto
		Forma baseada na região
		Forma baseada no contorno
		Descritor de forma 3D
	Movimento	Movimento da câmera
		Trajetória do objeto em movimento
		Movimento do objeto paramétrico na imagem
		Atividade de movimento

		Características da trajetória do movimento como velocidade
Áudio	Anotação da fala	Conjunto de palavras e fonemas incluindo metadados
	Timbre	Razão de harmônicas par a ímpar
		Coerência de ataque harmônico
	Melodia	Contorno melódico e ritmo

3.3.2 Esquemas de Descrição (DS)

Um Esquema de Descrição (DS) especifica a estrutura e a semântica dos relacionamentos entre seus componentes, que podem ser descritores ou esquemas de descrição [2].

Um exemplo de DS, o DS *ImageText* descreve uma região no espaço de uma imagem ou quadro que corresponde a um texto. Ou seja, o *ImageText DS* pode descrever os caracteres textuais, linguagem, fonte, em conjunto a outras propriedades do DS de uso geral.

Uma definição mais completa para os esquemas de descrição é que eles expandem os descritores do MPEG-7 combinando descritores individuais e outros DS no interior de estruturas mais complexas, e definindo os relacionamentos entre os descritores e os esquemas de descrição [25]. Para o MPEG-7 os DS são distribuídos considerando se eles se relacionam especificamente com áudio ou imagens, ou se eles se relacionam à descrição de multimídia.

Os seguintes conceitos são utilizados dentro do grupo DS para a descrição do conteúdo

audiovisual, de acordo com [2]:

- Estrutura Sintática – Estrutura física e lógica do conteúdo audiovisual.
- Estrutura Semântica – Descrição baseada no significado semântico, por exemplo, estruturas baseadas em eventos temporais e/ou objetos espaciais.
- Links Sintático-Semânticos – As associações entre elementos sintáticos e semânticos.

Normalmente, os Esquemas de Descrição de Multimídia (MDS) se referem a todos os tipos de mídia, que consistem de dados de áudio, vídeo e textual, enquanto os descritores específicos de domínio, tais como cor, textura, etc. se referem especificamente aos domínios de áudio e vídeo.

No próximo item, são apresentados os esquemas de descrição de acordo com seu uso.

3.3.2.1 MDS: Tipos e Aplicações

Os Esquemas de Descrição de Multimídia são divididos em diversos tipos de acordo com seu uso. Os tipos em questão são: Elementos Básicos, Descrição de Conteúdo, Gerenciamento de Conteúdo, Organização de Conteúdo, Navegação e Acesso e Interação com Usuário [29]. A figura 3.3 apresenta a distribuição dos MDS do MPEG-7.

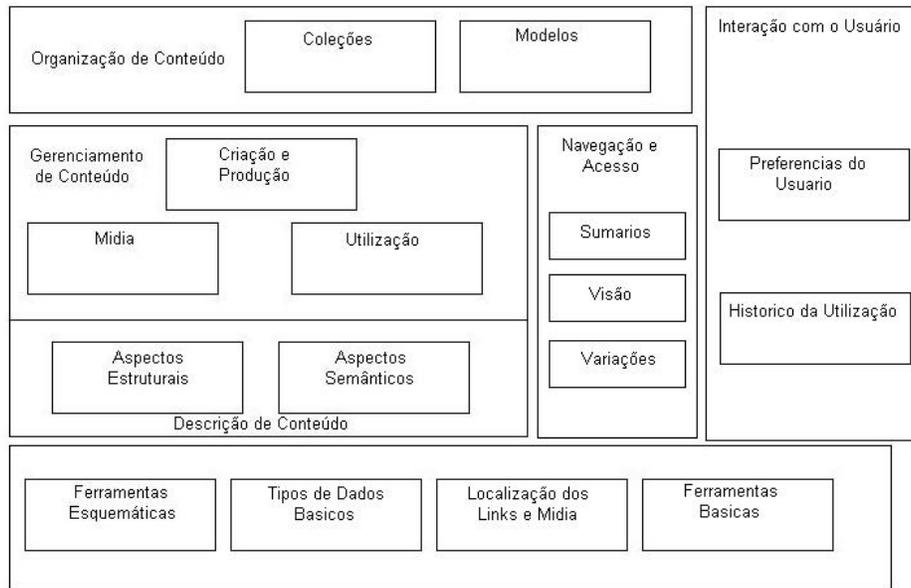


Figura 3.3. Distribuição dos MDS em MPEG-7 [5].

A especificação de MDS para o MPEG-7 define um numero de elementos básicos que serão usados como base fundamental durante toda a definição dos esquemas de descrição do MPEG-7 [32]. Abaixo, lista dos tipos de esquemas de descrições de multimídia e suas utilizações:

- **Elementos Básicos:** Esses elementos incluem ferramentas esquemáticas, tipos de dados básicos, ferramentas de localização e identificação da mídia, e ferramentas básicas de descrição.
- **Descrição de Conteúdo:** Esses elementos descrevem a estrutura (regiões, quadros de vídeo e segmentos de áudio) e semântica (objetos, descrições abstratas).
- **Gerenciamento de Conteúdo:** Esses elementos são os responsáveis pela descrição dos aspectos de criação e produção, codificação da mídia, formatos de armazenamento e dos dados e utilização do conteúdo.
- **Navegação e Acesso:** Esses esquemas de descrição tem o objetivo de facilitar a navegação e recuperação da mídia, seja pela criação de sumários, observações e variações

do conteúdo de mídia. Os sumários fornecem informações compactas sobre o conteúdo para facilitar sua localização e uso, as observações descrevem decomposições dos sinais de mídia no espaço, tempo e frequência.

- **Organização de Conteúdo:** Esses esquemas de descrição organizam e modelam coleções de conteúdo de mídia, para acesso mais rápido pelo usuário. As coleções podem ser descritas como um todo utilizando como base as suas propriedades em comum.

- **Interação com o Usuário:** Este esquema descreve as preferências dos usuários e histórico de utilização relacionada ao uso dos arquivos de mídia. Esse histórico permite, entre outros usos, a personalização do conteúdo em MPEG-7 para as necessidades do usuário de acordo com suas necessidades anteriores.

Nesta seção, foram apresentados os tipos e aplicações dos MDS para o MPEG-7 [23]. O DS Genérico Audiovisual representa a mais utilizada integração de todas as propostas de DS em um único esquema, como apresentado em [2]. Ele consiste de:

- 1) Uma coleção de DS estruturais sintáticos responsáveis por descrever elementos tais como segmentos, regiões, cores, etc.

- 2) Uma coleção de DS estruturais semânticos descrevendo objetos, atores, eventos, etc.

- 3) DS de ligação semântica e sintática, conectando os elementos sintáticos e semânticos.

- 4) DS de sumário, utilizado para permitir a navegação nos vários níveis de descrição.

- 5) DS de MidiaInfo, contendo descritores relacionados a mídia de armazenagem, como formato do arquivo, tipo de compressão, etc.

- 6) DS de MetaInfo, contendo descritores com informação sobre o criador ou distribuidor da mídia. Ex: *ContentDS*, *CreditDS*, *RightsDS*, etc.

- 7) DS de modelo, descreve os métodos de classificação para os dados de multimídia, e a

correspondência entre a mídia atual e outros conteúdos utilizando diferentes modelo.

Na figura 3.4, é mostrada a estrutura e como são conectados os diversos DS dentro do esquema Genérico Audiovisual:

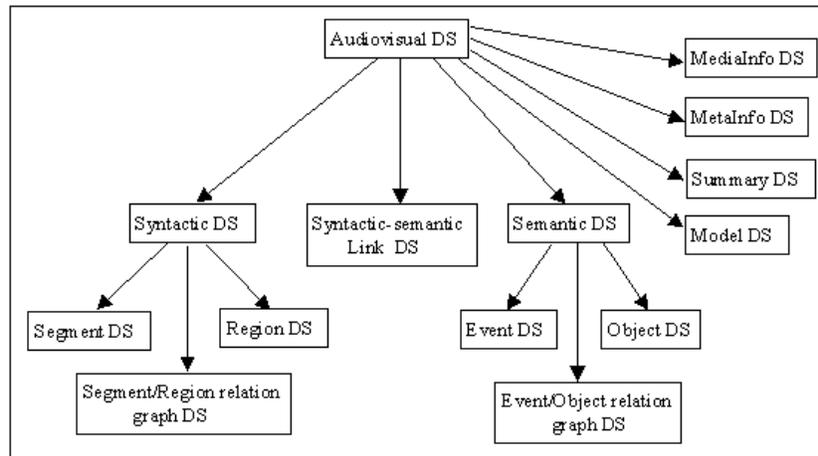


Figura 3.4. *Generic Audiovisual DS* [1].

Apesar de ser o mais amplamente utilizado, o DS Genérico Audiovisual é extremamente complexo e existe muita redundância e funcionalidades sobrepostas dentro dele. Isso ocorre devido à inclusão de propostas de DS diferentes no padrão, o que levou a conflitos. Ainda, algumas das propostas inseridas no DS Audiovisual Genérico têm aplicação duvidosa, por serem de uso muito complexo, o que afasta o usuário que não tem um conhecimento anterior do MPEG-7 da utilização desse esquema.

Nos últimos anos, o desenvolvimento de DS com funcionamento mais simples para o usuário que deseja apenas manipular metadados em mídia, sem ter que lidar com todos os esquemas no DS Audiovisual Genérico, foi extremamente ampliado. Com esses DS de uso mais simples, o uso do MPEG-7 aumentou sua presença na descrição de conteúdos de mídia na Web [25].

O próximo capítulo lidará exclusivamente com o DDL e seus conceitos, que são de uso fundamental para o desenvolvimento da parte prática desta tese.

Capítulo 4 DDL: Conceitos

O DDL (*Description Definition Language* ou Linguagem de Definição de Descrições) é uma linguagem que permite a criação e desenvolvimento de novos descritores em metadados para o MPEG-7, além de novos DS [7]. Ele também permite a extensão e modificação dos DS já existentes, como a *ImageText* DS [5]. O DDL é uma linguagem *schema* (*schema* é a base que define uma classe de documentos XML) que representa os resultados da manipulação de arquivos multimídia como um conjunto de restrições sintáticas, estruturais e de valores. Os descritores, DS e descrições devem ser capazes de compreender e utilizar esta linguagem. O DDL também fornece as regras que os usuários devem seguir para combinarem, estenderem e aprimorarem os DS e descritores existentes para o desenvolvimento de aplicações específicas [9].

Um *schema* DDL, como já foi dito, define as restrições que uma descrição MPEG-7 deve seguir. Ele é codificado em XML e inserido no documento diretamente pelo uso de marcadores. Um documento XML tem similaridades com o HTML, mas possui marcas definidas pelo usuário [20]. Por fim, MPEG-7 *instances* ou instâncias são documentos XML que se conformam a um esquema MPEG-7 particular, expresso em DDL, e que descreve o conteúdo multimídia. O DDL para MPEG-7 utiliza a linguagem W3C XML *Schema Language* como base e acrescenta mecanismos específicos ligados ao MPEG-7, como tipos de dados específicos, a ela [26].

Os requisitos DDL para o MPEG-7, de acordo com [23], devem ser capazes de expressar relacionamentos estruturais, espaciais, temporais, espaço-temporais, familiares e conceituais entre os elementos dentro de um DS e entre diversos esquemas. Ele também

devem fornecer um modelo para ligações entre uma ou mais descrições e o dado descrito. Obviamente, ele deve funcionar independente da plataforma e da aplicação, ser reconhecido pela máquina e de preferência pelo homem. Ainda, ele deve ser capaz de especificar o tipo de dado do descritor, seja primitivo (texto, números inteiros, etc) ou composto (histogramas).

Um *Parser* DDL deve ser utilizado para validar a sintaxe dos *schemas* MPEG-7, que incluirão os DS e descritores. Dada uma descrição MPEG-7 codificada em XML, o *Parser* deve ser capaz de verificar sua conformidade com as regras definidas no *schema* MPEG-7 correspondente [7].

Atualmente, existem poucos *Parsers* específicos para o MPEG-7 disponíveis, mas *Parsers* do XML podem ser utilizados para a validação, se as características específicas ao MPEG-7 não forem utilizadas ou forem removidas previamente.

4.1 DDL: Histórico

A decisão de utilizar o formato XML para a sintaxe DDL no MPEG-7 ocorreu após a discussão de diversas propostas no encontro de avaliação do MPEG-7, em Lancaster, 1999 [27]. Após a conferência, também foi acordado que o DDL para o MPEG7 deve ser capaz de validar as restrições já discutidas anteriormente, além do acréscimo de novas restrições caso necessárias.

A base do DDL para o MPEG-7 foi baseada na proposta do Centro de Tecnologia de Sistemas Distribuídos (DSTC) em 1999, em conjunto com as idéias de outros participantes [28]. Essa base deveria também ser ligada com os recentes desenvolvimentos da

comunidade W3C, como o XML *Schema* [26].

Entretanto, durante o desenvolvimento, foram levantadas divergências sobre a dependência do MPEG-7 nos resultados gerados pelo W3C para o XML *Schema*, parte essencial do MPEG-7 DDL. Assim, iniciou-se o desenvolvimento de um novo formato de XML *Schema* separado do W3C com o objetivo de ser exclusivo para o MPEG-7, utilizando conceitos específicos como descritores e DS.

Mas, no fim, foi decidido no encontro de março de 2000 pelo uso do XML *Schema* desenvolvido pelo W3C, acrescido de algumas extensões específicas para o MPEG-7. Essa decisão se baseou no uso estável comprovado do W3C *Schema*, e na sua provável aceitação pelos usuários. Outros fatores que contribuíram para essa escolha foram os Parsers do XML *Schema* serem de uso aberto aos usuários, e a possibilidade de inserção das extensões para o MPEG-7 para a solução dos problemas que viessem a surgir no uso continuado do *Schema* [30]. Essas extensões são validadas pelos Parsers existentes no *Schema*, que é o tema da próxima seção.

4.2 XML *SCHEMA*: Componentes

Um *Schema* XML consiste de um conjunto de componentes gerados para definir as estruturas e atributos dos descritores e DS [26]. Esses componentes são divididos em três grupos:

- Primários: O *schema* em si, *namespaces*, definições de tipo simples e complexo, declarações de atributo e elemento.

- Secundários: Definições de grupos de atributos, definições de restrições de

identidades, declarações de notação e definições do modelo.

- Auxiliares: Estes são dependentes dos outros componentes e não podem funcionar isoladamente. Exemplos: Anotações, grupos de modelos, partículas, etc.

4.3 XML Schema: Namespace e Envoltória

XML *Namespaces* fornecem um método simples para designar nomes únicos a tipos de elementos e atributos no interior de um documento XML, permitindo sua reutilização em outros documentos baseados neste padrão [31]. Um XML *Namespace* é uma coleção de nomes, identificados por um Identificador de Recursos Uniforme (URI), que são então utilizados em XML como tipos de elementos e atributos.

Para o MPEG-7, o mecanismo *namespace* permite que descritores e esquemas de descrição de múltiplos *schemas* diferentes sejam reutilizados e combinados para a criação de novos *schemas*. No início de cada *schema* MPEG-7, é obrigatório um cabeçalho identificando o *namespace* utilizado. Esse cabeçalho consiste de um *schema* XML, que inclui os seguintes atributos:

- *xmns*: Referência ao XML *Namespace*.
- *TargetNamespace*: A URI pela qual o *schema* atual é verificado.
- *xmns:MPEG-7*: A URI do *schema* MPEG-7 que será usado para a validação dos descritores e DS.
- *Version*: Especificação da versão do *schema*.

Abaixo, exemplo de um cabeçalho obrigatório para o *schema* MPEG-7:

```
<schema xmlns="http://www.w3.org/1999/XMLSchema"
        xmlns:mpeg-7="http://www.mpeg7.org/2000/MPEG7_schema"
```

```
targetNamespace="http://www.mpeg7.org/2000/MPEG7_schema"
.... (informações adicionais)
</schema>
```

4.4 XML Schema: Definições de Tipo

As definições de tipo (*Type Definition*) no XML Schema indicam os componentes internos que podem ser utilizados em outros componentes schema como os elementos, atributos ou outras definições de tipo. O XML Schema apresenta três componentes com definição de tipo: simples, complexo e derivado [30]. Os três tipos são explicados em mais detalhes na próxima seção.

4.4.1 Definições de Tipo Simples

As definições de tipo simples não podem carregar atributos ou possuir elementos de conteúdo. Mas tanto elementos quanto atributos podem ser declarados como tendo definições de tipo simples. Uma grande quantidade de definições de tipos simples é fornecida com o XML Schema, sendo tipos simples inseridos no *schema* ou derivados de outros tipos primitivos [27].

Além dos tipos simples já disponíveis no interior do *schema*, novos tipos simples podem ser criados utilizando restrições nos já existentes. Abaixo, um exemplo de nova definição de tipo simples utilizando restrições [30]:

```
<simpleType name="6bitInteger" base="nonNegativeInteger">
  <minInclusive value="0"/>
  <maxInclusive value="63"/>
</simpleType>
```

```

<simpleType name="DirectionType" base="string">
  <enumeration value="unidirectional"/>
  <enumeration value="bi-directional"/>
</simpleType>
<simpleType name='listOfString' base='string' derivedBy='list'/>

```

4.4.2 Definições de Tipo Complexo

Diferentemente das definições de tipo simples, as de tipo complexo podem carregar atributos ou possuir elementos de conteúdo. As definições de tipo complexo fornecem:

- Restrições à aparência e natureza dos atributos
- Restrições à aparência e natureza dos elementos de conteúdo.
- Derivação de tipos complexos para outros simples ou complexos através de extensões ou restrições.

Novas definições de tipo complexo são desenvolvidas utilizando o mecanismo *complexType* e essas definições normalmente contêm um conjunto de declarações e referências de elementos, além de declarações de atributos. Essas declarações não são tipos em si, mas sim uma associação entre um nome e restrições que governam a aparência daquele nome em documentos que utilizam o *schema* associado [30].

Abaixo, exemplo de uma definição de tipo complexo:

```

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

```

Abaixo, exemplo de uma utilização válida dessa definição:

```
<ProdComp id="040770">
  <OrgNome>Decom</OrgNome>
  <ContactarPessoa>Luis Paulo</ContactarPessoa>
  <Endereco>Unicamp</Endereço>
</ProdComp>
```

Dessa definição, obtém-se que quaisquer elementos *ProdComp* aparecendo em um documento MPEG-7 devem consistir de um elemento *OrgNome*, um ou mais elementos *ContactarPessoa* e um elemento *Endereco*. O primeiro desses elementos vai conter uma *string*, o segundo vai conter o *complexType IndividualType* e o terceiro o *complexType PlaceType*. Por fim, qualquer elemento cujo tipo seja declarado *Organization* deve aparecer com o atributo ID, que contém um número de identificação único.

4.4.3 Definições de Tipo Derivado

É possível derivar novos *complexTypes* por extensão de uma definição simples ou complexa ou por restrição de uma definição complexa. Uma definição de tipo complexo que por extensão gera outra faz isso através da adição de partes de modelos de conteúdo no fim dos outros modelos. A presença de declarações de atributos adicionais também pode gerar a extensão, ou em alguns casos, ambos.

O novo *complexType* pode ser derivado pela extensão de um *simpleType*, que ocorre pela adição de atributos, como é mostrado no exemplo abaixo [3] :

```
<complexType name="RelationType">
  <simpleContent>
    <extension base="string">
      <attribute name="source" type="IDREF" use="optional"/>
      <attribute name="target" type="IDREF" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<element name="Relation" type="RelationType"/>
<Relation source="edge1" target="edge2">morph</Relation>
```

No exemplo acima, os atributos *target* e *source* são adicionados a base simples para definir o *RelationType*, gerando a derivação.

O novo *complexType* pode ser derivado também pela extensão de um *complexType* já existente. No exemplo abaixo, o elemento *role*, que indica uma nova função, é utilizado para estender o *complexType Creator* [7]:

```
<complexType name="Creator" base="Person" derivedBy="extension">
  <element name="role" type="controlledTerm"/>
</complexType>
```

Uma definição de tipo cujas declarações estão em uma relação 1 para 1 com as declarações de outra definição, e restringindo as possibilidades daquele a que corresponde, é considerada uma restrição. O uso de restrições para derivação é apresentado no exemplo abaixo:

```
<complexType name="personName">
  <element name="title" minOccurs="0"/>
  <element name="forename" minOccurs="0" maxOccurs =
"unbounded"/>
</complexType>

<complexType name="simpleName" base="personName"
derivedBy="restriction">
  <element name="title" maxOccurs="0"/>
  <element name="forename" minOccurs="1" maxOccurs="1"/>
</complexType>
```

No exemplo acima, a definição de tipo complexo *simpleName* é derivada da definição *personName* pela restrição ao número de máximas ocorrências (*maxOccurs*) do elemento *forename*, de ilimitada para 1.

4.5 XML Schema: Tipos de Dados

Nesta seção, são definidos os tipos de dados do XML Schema. Esses tipos incluem os primitivos internos, os derivados internos, as facetas, e as listas e de união. Abaixo, apresentação mais detalhada dos diferentes tipos.

4.5.1 Tipos de Dados Primitivos Internos

Os tipos de dados primitivos internos são fornecidos dentro dos *datatypes* do XML Schema. Eles são:

- *string*: seqüências de caracteres.
- *boolean*: simples verdadeiro ou falso.
- *decimal*: seqüência finita de dígitos decimais.
- *float*: tipo de dado de ponto flutuante de 32 bits com precisão *single*.
- *double*: tipo de dado de ponto flutuante de 64 bits com precisão *double*.
- *duration*: duração de tempo no formato ISO8601 PnYnDTnHnMnS, onde nY=número de anos, nD=número de dias, T=data, nH=número de horas, nM=número de minutos e nS=número de segundos.
- *dateTime*: instante específico de tempo.
- *time*: um instante de tempo que ocorre todos os dias.
- *date*: data do calendário.
- *yearMonth*: mês e anos específicos do calendário Gregoriano.
- *year*: ano específico do calendário Gregoriano.

- *gMonthDay*: data Gregoriana que se repete todos os meses, como o dia dez do mês.
- *gMonth*: mês Gregoriano que ocorre todos os anos.
- *base64Binary*: dados binários codificados em base 64.
- *hexBinary*: dados binários codificados em base hexadecimal.
- *Qname*: representa nomes qualificados em XML no formato <nome do *namespace*, nome local> .
- *NOTATION*: esse tipo de dado contém o conjunto de todas as notações declaradas no *schema* atual. *NOTATION* deve ser usado somente para atributos, nunca diretamente no *schema*.

4.5.2 Tipos de Dados Derivados Internos

Os tipos de dados que foram derivados dos tipos primitivos, também são fornecidos pelo XML *Schema*. Os principais são [42]:

- *normalizedString*: *string* normalizado de espaço em branco.
- *token*: *strings* de símbolos (*tokens*) que não possuem seqüências internas de dois ou mais espaços.
- *language*: o conjunto de *strings* que são identificadores de linguagem válidos (RFC 1766, 1995).
- *NMTOKEN*: conjunto de símbolos que estão de acordo com a recomendação XML 1.0 de outubro de 2000.
- *Name*: conjunto de todas as *strings* que estão de acordo com a parte *Name* do XML 1.0.

- *NCName*: nomes “não colonizados” ou nomes não qualificados com um *Namespace* XML, de acordo com a recomendação *NCName* de [31].

- *ID*: o tipo de atributo ID de acordo com o XML 1.0.

- *ENTITY, ENTITIES*: atributo que indica entidade, entidades no XML 1.0.

- *Integer, nonPositiveInteger, negativeInteger, nonNegativeInteger, positiveInteger*: atributos para numerais inteiros, tanto negativos como positivos.

- *long, short*: limites positivos e negativos para os numerais inteiros, |9223372036854| para *long* e |32767| para *short*.

- *byte*: derivado de *short*, estabelece o limite positivo a 127 e o negativo a -128.

4.6 Extensões Específicas Para o MPEG-7

Para que o XML *Schema* se adequar às necessidades do MPEG-7, algumas extensões se fazem necessárias. Essas extensões se dividem em dois tipos, tipos de dados de vetores e matrizes, e tipos de dados derivados internos, que se somam aos derivados já existentes no XML *Schema* [28].

4.6.1 Tipos de Dados *Array* e Matrizes

O MPEG-7 necessita ser capaz de fornecer um mecanismo que force o *Parser* a restringir o tamanho de vetores e matrizes, seja para um valor predefinido de uma faceta em um *schema*, ou um atributo no momento da sua indicação. Para vetores unidimensionais e matrizes multidimensionais, utilizando o tipo de dado *list*, foram

desenvolvidos dois métodos para definir suas dimensões. O novo atributo *mpeg7:dimension*, que é uma lista de inteiros positivos, especifica as dimensões do vetor ou matriz.

Abaixo, um exemplo da utilização desse atributo na geração de uma matriz:

```
<simpleType name="IntegerMatrix3x4" base="integer" derivedBy="list">
  <mpeg7:dimension value="3 4" />
</simpleType>
<element name='IntegerMatrix3x4' type='IntegerMatrix3x4' />

<IntegerMatrix3x4>
  5    8  9  4
    6    7  8  2
    7    1  3  5
</IntegerMatrix3x4>
```

No caso de vetores e matrizes parametrizados, também tem-se o atributo *mpeg7:dim*.

Ele especifica as dimensões a serem aplicadas a um tipo *list* no momento da instanciação.

Abaixo, exemplo do uso do atributo *mpeg7:dim* :

```
<simpleType name="dim" base="integer" derivedBy="list" />
<complexType name="NDimIntegerArray" base="listOfInteger"
derivedBy="extension" > <attribute ref="mpeg7:dim" />
</complexType>
<element name="IntegerMatrix" type="NDimIntegerArray" />
<IntegerMatrix mpeg7:dim="3 4">  1  2  3  4
                                5  6  7  8
                                9 10 11 12</IntegerMatrix>
```

4.6.2 Tipos de Dados Derivados Internos

Além dos já existentes tipos de dados internos do XML *Schema*, são fornecidos internamente os seguintes tipos de dados necessários na utilização do MPEG-7 [42]:

MimeType ou tipo Mime (*Multipart Internet Mail Extension* ou Extensão de Correio na Internet em Multipartes) e *Character Set Code* ou Código de Conjunto de Caracteres, ambos desenvolvidos pela IANA (*Internet Assigned Numbers Authority* ou Autoridade dos Números Designados na Internet), e da ISO tem-se o *Country Code* ou Código de País (código ISO3166-1), o *Region Code* ou Código de Região (código ISO3166-2) e por fim, o *Currency Code* ou Código de Moeda (código ISO4217).

No próximo capítulo, será apresentada a implementação dos metadados utilizando o conceito de Ontologia, e como os metadados foram desenvolvidos.

Capítulo 5 Aplicação de Metadados

A parte prática desta tese apresenta primeiro o funcionamento do IBM *VideoAnnex*, que será a principal aplicação para a utilização dos metadados de semântica elevada. Em seguida, sé mostrado como o *VideoAnnex* foi utilizado para a inserção de metadados gerais, não levando em conta uma ontologia. Finalmente, utilizando o conceito de ontologia, foi desenvolvido um conjunto de metadados para serem utilizados na descrição de partidas de futebol. Todo o programa foi desenvolvido em inglês, devido a utilização do MPEG-7 ser mais difundida em países de língua inglesa, e para permitir a publicação desta ontologia futuramente.

São apresentadas as criações dos metadados, utilizando a linguagem de programação DDL e como são inseridos no arquivo de mídia através do *VideoAnnex*. Alguns metadados mais avançados que não podem ser inseridos pelo *VideoAnnex* são inseridos diretamente no arquivo XML. Através destes processos, é apresentada a criação e inserção de metadados de semântica elevada em um arquivo de mídia, levando-se em conta o conceito de ontologia [13].

5.1 Ferramenta Utilizada: IBM *VideoAnnex Annotation Tool*

A IBM *VideoAnnex Annotation Tool* [36] ou Ferramenta de Anotação e Anexação em Vídeo foi desenvolvida pela IBM para auxiliar os autores na anotação de vídeo com

metadados utilizando o formato MPEG-7. Ela foi utilizada nesta tese devido a sua facilidade de uso por inserir diretamente os metadados no arquivo MPEG, sua utilização ser gratuita e disponível para todos os interessados a partir do site: <http://www.alphaworks.ibm.com/tech/videoannex>. Também não é necessária a instalação de nenhum programa além do *VideoAnnex* para seu funcionamento (outros anexadores de metadados necessitam de Java instalado na máquina do usuário). O arquivo final é salvo no formato MPEG-7 diretamente, sem a necessidade de conversões, etc.

5.1.1 Instalação do Programa

É necessário no mínimo que o usuário possua um Pentium 2 com 128 MB de RAM e o sistema operacional Windows 95 ou superior instalado para o funcionamento do IBM *Videoannex*. Percebe-se que o programa não necessita de um computador com grandes recursos, o que facilita o seu uso por uma maior população de usuários. Após o download do programa da página <http://www.alphaworks.ibm.com/tech/videoannex/download>, deve-se descompactar o arquivo .zip em uma nova pasta *VideoAnnex v151*, que é a versão utilizada nesta tese. Após a descompactação, simplesmente clique no ícone *VideoAnnEx* e o programa irá ser iniciado. Qualquer problema durante a instalação, o usuário pode entrar em contato com a IBM pela página <http://www.alphaworks.ibm.com/tech/videoannex>.

5.1.2 IBM VideoAnnex: Funcionamento

Imagem da tela principal do IBM VideoAnnex Annotation Tool:

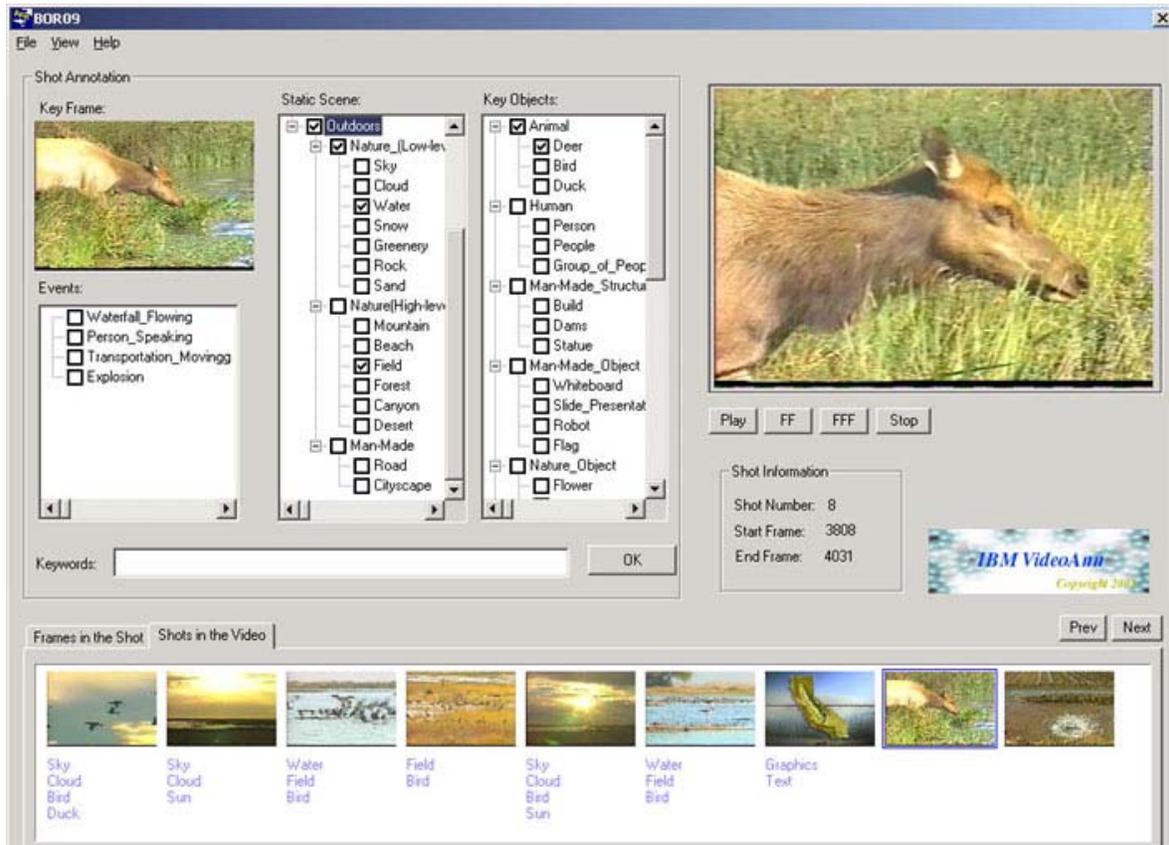


Figura 5.1. Tela Principal do IBM VideoAnnex Annotation Tool [32].

O VideoAnnex utiliza arquivos de vídeo somente nos formatos MPEG-1 ou MPEG-2. Ao abrir o arquivo MPEG, o VideoAnnex divide o vídeo em *shots* ou cenas, analisando variações de cor, mudanças de cena, etc. Então, ele salva esse conjunto de *shots* junto ao arquivo original, no formato .FRP, que é o arquivo associado com a ferramenta PerForm Pro Plus. Essa extensão foi escolhida por opção dos programadores do IBM VideoAnnex.

Como se percebe na figura 5.1, o VideoAnnex é dividido em quatro seções. No canto direito superior, a janela de exibição de vídeo, com informações sobre as *shots*. No campo

superior esquerdo, é a seção *Shot Annotation* com uma *key frame* (quadro-chave) sendo exibida. Esse quadro pode ser alterado de acordo com a preferência do usuário no painel *frames in the shot*.

Na parte de baixo do *VideoAnnex* está o *Views Panel*, que apresenta a seqüência de *shots* e os metadados já inseridos em cada *shot*. A seção de *Region Annotation* não aparece na imagem 5.1, ela é utilizada caso o usuário deseje inserir os metadados localizando as regiões espacialmente na *shot*. Utilizando essas quatro seções, o usuário poderá inserir os metadados no arquivo MPEG de forma interativa.

A parte principal do IBM *VideoAnnex Annotation Tool* é a parte de *Shot Annotation*, onde de fato a inserção de metadados será realizada em cada *shot*. Essa parte do *VideoAnnex* é apresentada na figura 5.2.

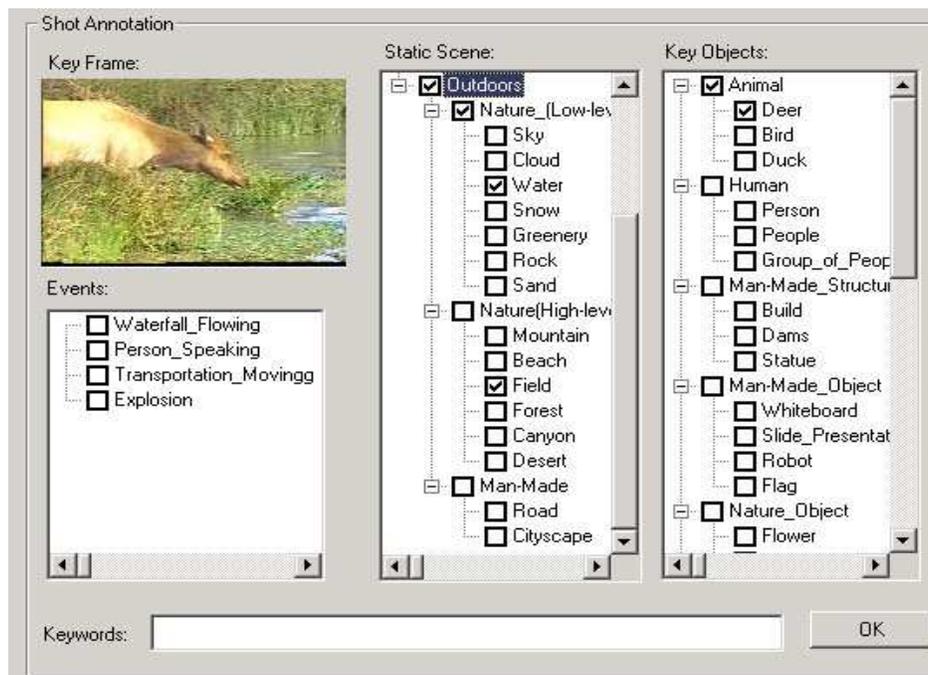


Figura 5.2. Painel *Shot Annotation* [32].

Como se percebe na imagem 5.2, o quadro chave da atual *shot* é exibido na janela de

mesmo nome. O quadro chave é uma imagem escolhida aleatoriamente entre as existentes dentro da *shot* para auxiliar o usuário a inserir os metadados, de acordo com o que ocorre nessa determinada *shot*. O usuário pode alterar o quadro chave de acordo com suas preferências no painel *Frames in the Shot*.

No painel *Shot Annotation*, o Lexicon de anotação é exibido. Lexicon é o arquivo em XML que gera os metadados utilizados pelo *VideoAnnex*, e é através da manipulação do lexicon que são desenvolvidos os metadados específicos. O *VideoAnnex* no momento da instalação já possui um conjunto de metadados desenvolvidos pelos criadores do programa, que é direcionado a um uso genérico para qualquer tipo de filme. Obviamente, esses metadados são limitados para aplicações de tipo mais avançado, então a manipulação do lexicon é absolutamente necessária para o uso do *VideoAnnex*.

No lexicon original do *VideoAnnex*, os metadados de alto nível semântico podem ser inseridos de acordo com as seguintes divisões, vide a imagem 5.2:

- *Events*: Lista de eventos ou ações que podem ser utilizados para descrever o que ocorre na *shot*.
- *Static Scene*: Lista de cenas estáticas no fundo do vídeo que podem ser utilizadas para a descrição das *shots*.
- *Key Objects*: Lista dos objetos de importância vistos na *shot*.

Os nomes das divisões também podem ser alterados, de acordo com o interesse do usuário. Em cada uma das divisões, as descrições estão organizadas em uma estrutura hierárquica em formato árvore. Essas descrições possuem pontos de marcação correspondentes para o usuário selecionar de acordo com a *shot*. A barra *Keywords* também permite a inserção de descrições em formato textual.

Várias descrições simultâneas podem ser inseridas na barra *Keywords* desde que separadas por vírgulas. Uma vez escolhidas as descrições, clicando no botão OK realiza a inserção dos metadados naquela determinada *shot*, e passa para a marcação da próxima *shot*.

Selecionando a opção *Region Annotation* na parte *Tools* do *VideoAnnex*, quando o botão OK for clicado aparecerá uma nova janela para que o usuário selecione a área de interesse para marcação espacial na *shot* para os metadados escolhidos.

Para facilitar o trabalho de inserir metadados *shot a shot*, foi desenvolvida a opção *Annotation Learning* no menu *Tools*. Após o usuário selecionar essa opção, *shots* similares entre si terão as mesmas descrições, sem que o usuário tenha que inserí-las novamente. Este processo entretanto não é confiável, portanto se recomenda que o usuário verifique se as descrições foram inseridas corretamente.

As outras opções do menu *Tools* se referem a desenvolver novos Lexicons e manipular os já existentes.

Após a marcação das *shots*, que de acordo com o tamanho do filme podem ser centenas, o usuário pode salvar o trabalho já feito como um arquivo MPEG-7 na extensão .MP7, através do menu *file*.

5.1.3 Exemplo de Anotação Utilizando o *VideoAnnex*

Nesta seção, será apresentado um exemplo de marcação de metadados em um arquivo de vídeo no formato MPEG-2, criando um arquivo XML no formato MPEG-7 de forma

básica. Não é o objetivo dessa tese apresentar utilizações avançadas do *VideoAnnex*, portanto para esse fim recomenda-se a procura por outros trabalhos como [14] e [32].

O primeiro passo é carregar o arquivo de vídeo (lembrando que o formato deve ser MPEG-1 ou MPEG-2), no menu *File* pela opção *Open*:

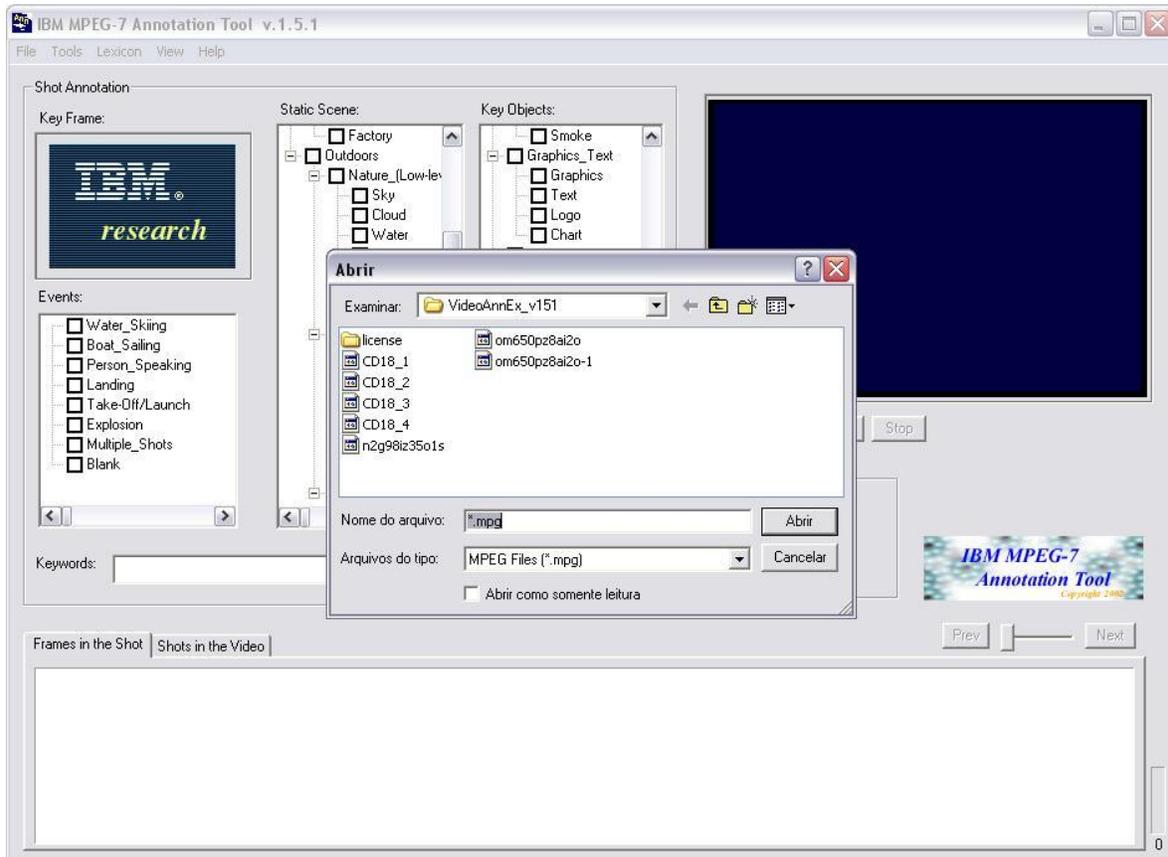


Figura 5.3. Abertura de Vídeo MPEG-2 no *VideoAnnex*.

Em seguida, assistindo o vídeo pela janela *Vídeo Playback*, o usuário tem uma idéia inicial dos metadados que serão necessários para este filme. Ao abrir o filme, também é criado o arquivo FRP com as informações sobre as *shots*, que são exibidas na parte de baixo do programa, junto com os quadros existentes dentro de cada *shot*, como mostrado na figura 5.4.

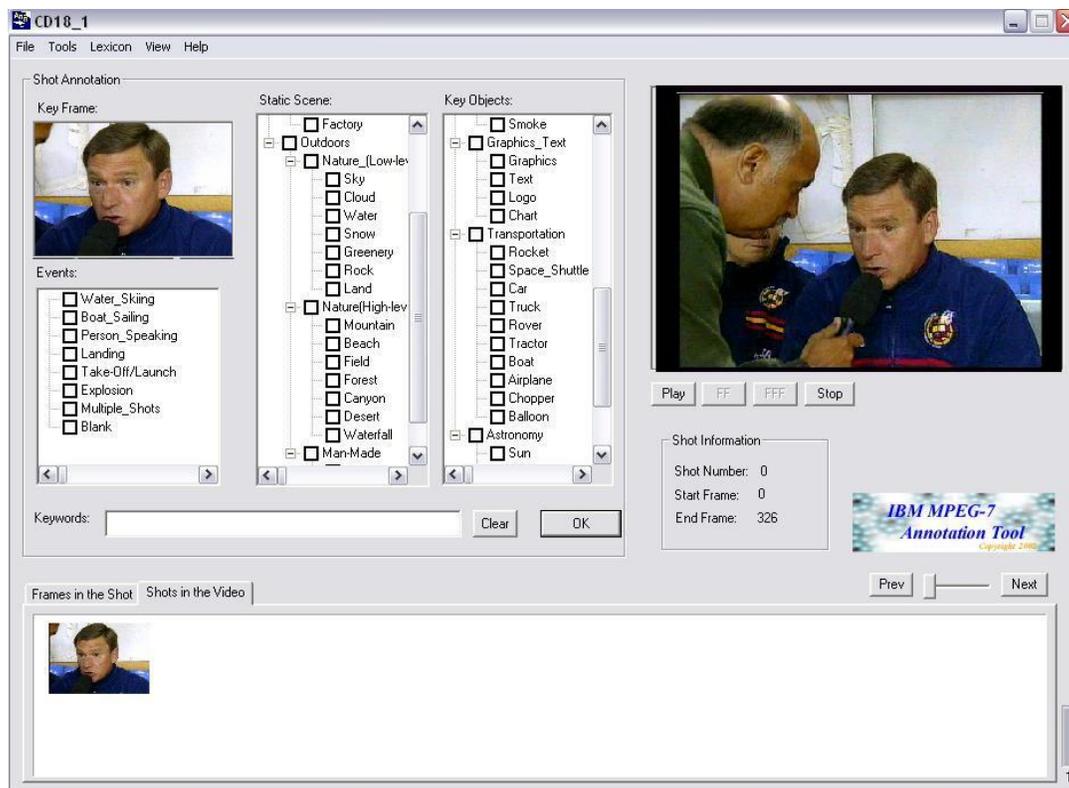


Figura 5.4. Visualização do Vídeo no *VideoAnnex*.

Agora, o usuário pode iniciar a escolha dos metadados necessários para a *shot* atual nos painéis *Events*, *Static Scene* e *Key Objects*. Após a escolha, são marcados os *slots* adequados. Caso seja necessário, o usuário utiliza o painel *Keywords* para a inserção de metadados em forma de texto, separados por vírgulas. Finalizada a marcação e a digitação do texto, o usuário passa para o próximo *shot* com o botão OK. As anotações já feitas podem ser visualizadas no painel *Shots in the Vídeo*, como mostra a figura 5.5.

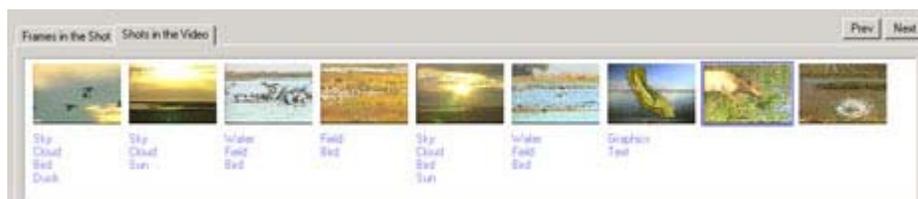


Figura 5.5. Visualização das Anotações.

Para salvar as anotações já feitas, no menu *File* o usuário deve clicar a opção *Save*

MPEG-7 XML e salvar com o nome desejado. Para carregar as anotações, no mesmo menu o usuário seleciona a opção *Load MPEG-7 XML* e escolhe o arquivo MP7 desejado, lembrando que o usuário deve carregar o filme MPEG antes das anotações. Por fim, para alterar anotações já feitas, o usuário seleciona a *shot* que ele deseja alterar no painel *Shots in the Vídeo* clicando duas vezes, e as anotações vão ser exibidas. O usuário então pode alterá-las e quando terminar, clica em OK para manter as alterações.

Com esses passos, o usuário já é capaz de criar um arquivo MPEG-7 com os metadados de semântica elevada necessários. Os metadados disponíveis no *VideoAnnex* inicialmente não são suficientes para uma aplicação de um nível mais avançado. Na próxima seção, é apresentado o desenvolvimento de um conjunto de metadados baseado no conceito de ontologia destinado a aplicação em partidas de futebol.

5.2 Aplicando Metadados de Semântica Elevada: A *Soccer*

Ontology

A utilização de metadados de semântica elevada é o principal foco do desenvolvimento de metadados atualmente, e esta tese também tem esse objetivo. Foi desenvolvido para o *VideoAnnex* um conjunto de metadados levando em conta o conceito de ontologia, apresentado no item 3.2. Ontologia como definido por Schreiber, é a “especificação de uma conceituação”. Aplicado aos metadados, indica o desenvolvimento de descrições direcionadas a um fim específico. Como as aplicações para os metadados de semântica elevada são diversas, desenvolver um conjunto de metas para uma determinada aplicação é

muito interessante, pois permite ao usuário mais liberdade para trabalhar em uma área específica. Com diversos grupos desenvolvendo metadados específicos utilizando o formato MPEG-7, é possível reunir os metadados já desenvolvidos para aplicações mais gerais.

Nesta tese, foi desenvolvido um conjunto de metadados em semântica elevada para utilização no *VideoAnnex* direcionados especificamente para a descrição de uma partida de futebol. Com a transmissão de partidas de futebol em HDTV [5], o interesse pelo esporte em todo o mundo e a possibilidade das partidas serem exibidas pela Internet, etc. é obvio o interesse por metadados focados especialmente no esporte futebol. Já houve outras tentativas de desenvolvimento de ontologias para o futebol, como o desenvolvido pelo *SmartWeb project* [11], mas essas tentativas tinham o objetivo de trabalhar com outras ferramentas como o *Semantic Web*. Por esse motivo, a ontologia para essa tese foi desenvolvida inteiramente para o uso com o *VideoAnnex* baseado no formato MPEG-7. Essa ontologia para facilitar ao usuário será chamada de *Soccer Ontology*. A *Soccer Ontology* está disponível em sua forma completa no Apêndice 2. Na próxima seção, descreve-se como foi desenvolvida a ontologia no campo XML.

5.2.1 Início do Desenvolvimento da *Soccer Ontology*

O desenvolvimento da *Soccer Ontology* foi realizado no ambiente DDL, que tem como base o formato XML da *Semantic Web* com algumas melhorias específicas para o MPEG-7. Em primeiro lugar, utilizando o lexicon ou conjunto de metadados original do *VideoAnnex*

é possível fazer modificações simples nos metas utilizando o menu *Lexicon*, como é visto na figura 5.6.

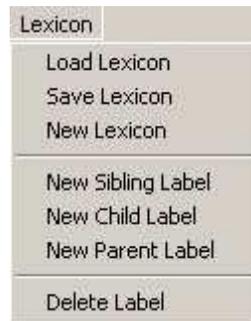


Figura 5.6. Menu *Lexicon* do *VideoAnnex*.

As opções disponíveis são:

- *Load Lexicon*: Carrega o lexicon desenvolvido pelo usuário. Quando o programa é iniciado, será carregado sempre o lexicon padrão. É o que vai permitir que o usuário desenvolva *Lexicons* específicos para determinadas aplicações (ontologias), como a *Soccer Ontology*.

- *Save Lexicon*: Salva as alterações efetuadas pelo usuário nos metadados utilizando os comandos abaixo.

- *New Lexicon*: Gera um lexicon vazio, o que permite ao usuário criar do zero um novo lexicon, definindo as hierarquias de acordo com suas preferências.

- *New Sibling Label*: Cria uma nova entrada no lexicon, que é “irmã” da atual. Ou seja, cria-se uma nova entrada de metadados no mesmo nível hierárquico da atual.

- *New Child Label*: Cria uma nova entrada no lexicon, que é “filha” da atual. Ou seja, cria-se uma nova entrada de metadados no nível hierárquico diretamente inferior a atual.

- *New Parent Label*: Cria uma nova entrada no lexicon, que é “pai” da atual. Ou seja, cria-se uma nova entrada de metadados no nível hierárquico diretamente superior a atual.

O Menu *Lexicon* é muito útil quando se deseja fazer modificações em um lexicon já

existente, mas para desenvolver um novo lexicon do zero como foi feito nessa tese, ele não é funcional, pois cada uma das entradas deve ser desenvolvida individualmente. Utilizando o Menu *Lexicon*, foi desenvolvido um lexicon mais completo direcionado ao uso geral de marcação de vídeos, baseado no lexicon original desenvolvido para o *VideoAnnex*. Obviamente, esse lexicon está longe de ser perfeito devido à quantidade de metas possíveis e seus usos são praticamente infinitos, mas para o usuário que deseja iniciar a marcação de vídeos utilizando o *VideoAnnex* é um bom começo. Esse lexicon encontra-se no Anexo 1.

Para se desenvolver um novo lexicon do zero, é mais interessante trabalhar diretamente no ambiente DDL, que permitirá mais flexibilidade ao usuário no momento da criação dos metadados necessários. Em primeiro lugar, o usuário deve criar ou utilizar um arquivo já existente .LEX. Essa terminologia é importante para que o lexicon seja reconhecido pelo *VideoAnnex*. Após a criação do arquivo, sua manipulação pode ser feita no *Word*, *Notepad*, ou outra ferramenta de manipulação de texto. Começa-se com o cabeçalho do lexicon.

5.2.2 Cabeçalho da *Soccer Ontology*

O lexicon vai ser programado em DDL, então como foi apresentado no ítem 4.3, deve ser indicado um *Namespace* para que esse Lexicon possa ser utilizado por outras aplicações. No caso da *Soccer Ontology*, o *Namespace* ficou da seguinte forma:

```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"  
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
```

O *Namespace* pode ser mantido o mesmo do lexicon padrão do *VideoAnnex*, caso seja do interesse do usuário. No caso da *Soccer Ontology* foi isso que ocorreu, mas caso o

usuário deseje utilizar uma URI diferente para a validação dos esquemas e descritores MPEG-7, isso também é possível.

Em seguida, começa o cabeçalho em si. Utilizando a rotina *header*, o usuário pode começar a inserir as informações específicas a este lexicon [17]. Em primeiro lugar, o usuário pode inserir as suas informações pessoais para indicar que foi o responsável pelo desenvolvimento do lexicon, lembrando que essas informações devem ser inseridas utilizando o formato DDL. Na *Soccer Ontology*, foram usados como base os créditos aos criadores do *VideoAnnex*, e a partir deles foram desenvolvidos os créditos ao usuário que desenvolveu a *Soccer Ontology*. Os créditos então ficaram da seguinte forma:

```
<Creator>
  <Role href="creatorCS">
    <Name>Creator</Name>
  </Role>
  <Agent xsi:type="OrganizationType">
    <Name>Tese DECOM</Name>
  <Contact xsi:type="PersonType">
    <Name>
      <GivenName>Luis Andre</GivenName>
      <FamilyName>Ferreira</FamilyName>
    </Name>
    <ElectronicAddress>
      <Email>luisandre1@yahoo.com.br</Email>
    </ElectronicAddress>
  </Contact>
</Agent>
</Creator>
```

Percebe-se que foi utilizada a rotina *Creator*, onde as informações foram inseridas de acordo com as normas básicas do DDL, como foram apresentadas no capítulo anterior.

É interessante ao usuário acrescentar mais informações ao cabeçalho para que o conjunto de metadados fique diferenciado, além de detalhes sobre a reprodução desse conjunto. Ou seja, no caso da *Soccer Ontology* foi indicado a data de criação da ontologia e informações sobre sua utilização utilizando a rotina *Description Metadata*.

Em DDL, temos:

```
<DescriptionMetadata>
  <Comment>
<FreeTextAnnotation>Description of a Soccer Match.</FreeTextAnnotation>
  </Comment>
  <CreationTime>2006-01-05T13:13:7-03:00</CreationTime>
</DescriptionMetadata>
```

A informação *Description of a Soccer Match* foi inserida como uma anotação de texto, permitindo uma fácil identificação sobre o uso dessa específica ontologia. A data de criação da *Soccer Ontology* foi inserida no formato reconhecido pelo MPEG-7, o ISO 8601 [33]. Nesse caso, tem-se a data completa com o ano, mês e dia, horário em horas, minutos e segundos e fuso horário. Portanto, a *Soccer Ontology* foi finalizada no dia 5 de janeiro de 2006, 13:13:07 da tarde, no horário de Brasília. Percebe-se que a indicação da data indica ao usuário dos metadados se as informações que ele está utilizando são atuais.

Também a permissão de reprodução dos metadados desenvolvidos deve ser indicada, para evitar a reprodução não autorizada. A *Soccer Ontology* foi desenvolvida com o objetivo de permitir aos usuários trabalharem com a marcação de jogos de futebol sem restrições. Portanto, utilizando a rotina *Usage Information*, em DDL tem-se:

```
<UsageInformation>
  <Rights>
    <RightsID organization="DECOM" type="4MCopyrights">Unrestricted
access. This object may be copyright-protected. Contact the owner or
rights holder for permission to reproduce.</RightsID>
  </Rights>
</UsageInformation>
```

Foi indicada a organização para a qual foi desenvolvida a ontologia, o DECOM, o tipo *4MCopyrights* indica a proteção do conteúdo MPEG-7, essa ontologia tem acesso irrestrito, mas a permissão de reprodução deve ser requerida junto ao criador. Quando se trabalha com vídeos disponíveis na Internet e outras fontes, a questão dos direitos autorais é

importantíssima, devido a cópias não autorizadas, envio de arquivos, etc. Portanto, é fundamental a indicação dos direitos na criação de novos artigos utilizando o MPEG-7 como a *Soccer Ontology*.

Por fim, é necessária a indicação da ferramenta que utilizará este lexicon, no caso da *Soccer Ontology* o *IBM VideoAnnex*. Portanto, utilizando a rotina *Tool*, no arquivo XML ficou assim:

```
<Instrument>
  <Tool>
    <Name>IBM VideoAnnEx MPEG-7 Annotation Tool</Name>
  </Tool>
</Instrument>
```

A indicação da ferramenta não é obrigatória, o *VideoAnnex* vai conseguir utilizar o lexicon mesmo faltando a indicação, mas é interessante para outros usuários que venham utilizar o lexicon saibam para qual aplicação ele foi desenvolvido.

Por fim, o cabeçalho está completo. Foram indicadas as informações sobre a *Namespace* utilizada, o criador da *Soccer Ontology*, data de criação, informações sobre a ontologia, direitos autorais e a ferramenta que vai utilizar a ontologia. Na próxima seção, será apresentado o desenvolvimento da *Soccer Ontology*, com a criação e programação dos metadados destinados a descrição semântica de partidas de futebol.

5.2.3 *Soccer Ontology*: Desenvolvimento dos Metadados

Os metadados que serão utilizados para a marcação de partidas de futebol utilizando o *VideoAnnex* devem ser criados no ambiente XML, levando em consideração as normas de desenvolvimento do MPEG-7, que já foram apresentadas em capítulos anteriores. Como se sabe, o *VideoAnnex* divide os metadados em três painéis, cada um com

específicas descrições em seu interior. O lexicon padrão do *VideoAnnex* denomina essas divisões como *Events*, *Static Scene* e *Key Objects*. Para a *Soccer Ontology* essas divisões foram renomeadas visando a aplicação para partidas de futebol. Ou seja, as divisões são chamadas *Events Soccer Match* (Eventos de uma partida de futebol), *Soccer Match Plays* (Jogadas em uma Partida de Futebol) e *Key Strikes and Objects* (Ataques e Objetos Principais). Cada painel tem as seguintes especificações:

- *Events Soccer Match*: Os principais eventos que podem ocorrer durante uma partida de futebol. Exemplos: Aquecimento, início da partida, prorrogação, falta de luz, etc. Neste painel também se incluem as condições do clima durante a partida.

- *Soccer Match Plays*: As principais jogadas que ocorrem durante a disputa de uma partida. Elas são divididas em estáticas ou dinâmicas. Exemplos: Passes, cruzamentos, impedimentos, faltas, etc.

- *Key Strikes and Objects*: Este painel inclui os principais ataques das duas equipes, incluindo o principal objetivo do futebol, os gols, e os objetos principais utilizados durante uma partida de que podem ser visualizados durante o vídeo. Exemplos desses objetos: a bola, bandeira de escanteio, traves, etc.

Para efetuar a mudança dos títulos dos painéis do *VideoAnnex*, foi iniciada a programação do lexicon no ambiente DDL. Após a criação do cabeçalho, o primeiro painel foi desenvolvido, utilizando como base os conhecimentos obtidos sobre a programação XML. Para a mudança do título do painel na *Soccer Ontology*, tem-se as seguintes linhas de programação:

```
<Term termID="Events_Soccer_Match">  
  <Name xml:lang="en">Events Soccer Match</Name>  
  ... (inserção dos metadados deste painel)  
</Term>
```

Percebe-se mais uma vez as similaridades entre o XML e a linguagem HTML. Utilizando a rotina *Term*, foi inserido o nome do painel em inglês (*xml:lang= "en"*) *Events Soccer Match*. Após a inserção do título, são inseridos os metadados deste painel utilizando o mesmo formato, lembrando da hierarquia dos metadados no *VideoAnnex*. E por fim, fecha-se esse determinado painel com *</Term>*.

Os outros dois painéis na *Soccer Ontology*, *Soccer Match Plays* e *Key Strikes and Objects* foram programados da mesma maneira. Deve-se prestar muita atenção no momento da programação, a menor diferença entre o começo e o fim de uma rotina ou um pequeno erro de concordância durante a programação e o *VideoAnnex* não irá reconhecer o lexicon.

Após a criação dos painéis, inicia-se o desenvolvimento dos metadados que serão realmente inseridos em cada um dos painéis para o uso na marcação de partidas de futebol. Abaixo, cada um dos painéis e os metadados que foram inseridos em cada um deles utilizando a programação XML.

- *Events Soccer Match*: Os principais metadados que estão disponíveis dentro deste painel na *Soccer Ontology* são: *Weather Conditions*, *Voice Over*, *Fight*, *National Anthem Plays*, *Warm-Up*, *First Half Begins*, *Half Time*, *Second Half Begins*, *End of the Second Half*, *Overtime*, *Penalty Kicks*, *Power Outage* e *Blank*. Quando necessário, alguns desses metadados foram divididos em mais metadados utilizando o conceito de árvore hierárquica do *VideoAnnex*. Por exemplo, o metadado *Weather Conditions* se divide nas seguintes opções: *Day*, *Night*, *Rain*, *Snow*, *Clouds* e *Fog*. Utilizando estes metas, o usuário pode descrever os principais eventos que ocorrem durante uma partida de futebol.

- *Soccer Match Plays*: Este painel se divide em duas categorias principais, *Static Plays* (jogadas estáticas) e *Actual Plays* (jogadas de fato ou dinâmicas). Dentro do conjunto de

metadados *Static Plays*, tem-se os seguintes metadados: *Kick-Off*, *Corner Kick*, *Goal Kick*, *Throw In*, *Foul*, *Two Touch Foul*, *Offside*, *Yellow Card*, *Red Card*, *Hand Touched Ball*, *Crowd View*, *Stadium View*, *Goalie/Player Warnings* e *Player Substitution*. Como no painel anterior, muitos desses metadados se dividem em mais metas para aprimorar a marcação dos jogos. Por exemplo, o Metadado *Goal Kick* se divide em *For the Home Team* e *For the Away Team*, indicando qual o time que cobrará o tiro de meta. Dentro do segundo conjunto, *Actual Plays*, tem-se: *Home Team Attacking*, *Away Team Attacking*, *Player Passing*, *Wrong Pass*, *Long Range Pass*, *Head Dispute*, *Ball Stolen*, *Goalie Defense*, *Goalie Pass*, *Player Carries the Ball*, *Player Crosses the Ball*, *Player Loses the Ball*, *Ball Hits Corner Flag*. Com os metadados deste painel, o usuário pode descrever as principais jogadas estáticas e dinâmicas de uma partida de futebol, lembrando que os principais ataques e gols se localizam no terceiro painel.

- *Key Strikes and Objects*: Este painel se divide em cinco categorias principais, *Goal*, *Shots*, *Human*, *Soccer Key Objects* e *Graphics and Text*. Em *Goal*, como o nome já indica, são descritos os gols da partida. Dentro desta categoria, tem-se os seguintes metadados: *For the Home Team*, *For the Away Team*, *Header*, *Long Range Kick*, *Médium Range Kick*, *Close Range Kick*, *Own Goal*, *Players Celebrating*, *Goal Overturned*. Em *Shots*, são descritos os principais chutes a gol da partida que não terminam em gols. Dentro desta categoria, se encontram os seguintes metadados: *For the Home Team*, *For the Away Team*, *Over the Goal*, *Off the Side*, *Net Hit from the Outside*, *Goalie Defense*, *Hit Bar Pole*, *Hit Opposing Player*, *Hit Friendly Player*. Em *Human*, é descrito o elemento humano em campo, jogadores, técnicos, juiz, assistentes, etc. Portanto, incluídos nesta categoria tem-se os seguintes metadados: *Player (w/Front View Face)*, *Player (w/Side View Face)*, *Player (w/o Face)*, *Reserve Players*, *Refs(w/Front View Face)*, *Refs(w/Side View Face)*, *Refs(w/o*

Face), *People Outside the Game*, *Player Body Parts*, *Player Interviews*, *Coach/Other Interviews*. Em *Soccer Key Objects*, aparecem os principais objetos disponíveis em uma partida de futebol que são visíveis em uma determinada shot. Tem-se, então os seguintes metadados: *Ball*, *Goal Poles*, *Soccer Stadium*, *Grass*, *Corner Flag*, *Gurney*, *Ambulance*, *Reserves Bench*, *Press Seats*, *Score Board* e *Crowd Shot*. Por fim, tem-se a categoria *Graphics and Text*, que irão descrever os gráficos e textos exibidos normalmente durante uma partida. Os metadados pertencentes a essa categoria são: *Graphics*, *Replay*, *Text*, *Logo*, *Chart*, *Game Time*, *Yellow/Red Cards*.

Percebe-se que utilizando a *Soccer Ontology*, o usuário tem a sua disposição uma grande variedade de metadados específicos com os quais pode descrever em nível de semântica elevada uma partida de futebol. A *Soccer Ontology* permite a descrição detalhada de todos os eventos envolvendo a partida, todos os participantes dentro e fora do campo, os principais lances, etc. Mas percebe-se que alguns metadados não podem ser inseridos utilizando os metadados presentes na *Soccer Ontology*, como nomes dos jogadores, detalhes da partida específica que se está descrevendo como data, local, etc e outros. Para esses metadados em especial, o usuário pode inserir a informação como texto utilizando o painel *Keywords* do *VideoAnnex*, ou inserir os metadados diretamente no arquivo final .MP7, que é criado após a marcação utilizando o *VideoAnnex*. Na próxima seção, onde será apresentada a utilização de fato da *Soccer Ontology*, essa opção será apresentada.

5.2.4 Aplicando a Soccer Ontology

Abaixo, na figura 5.7, tem-se a *Soccer Ontology* carregada para a marcação de uma partida de futebol.

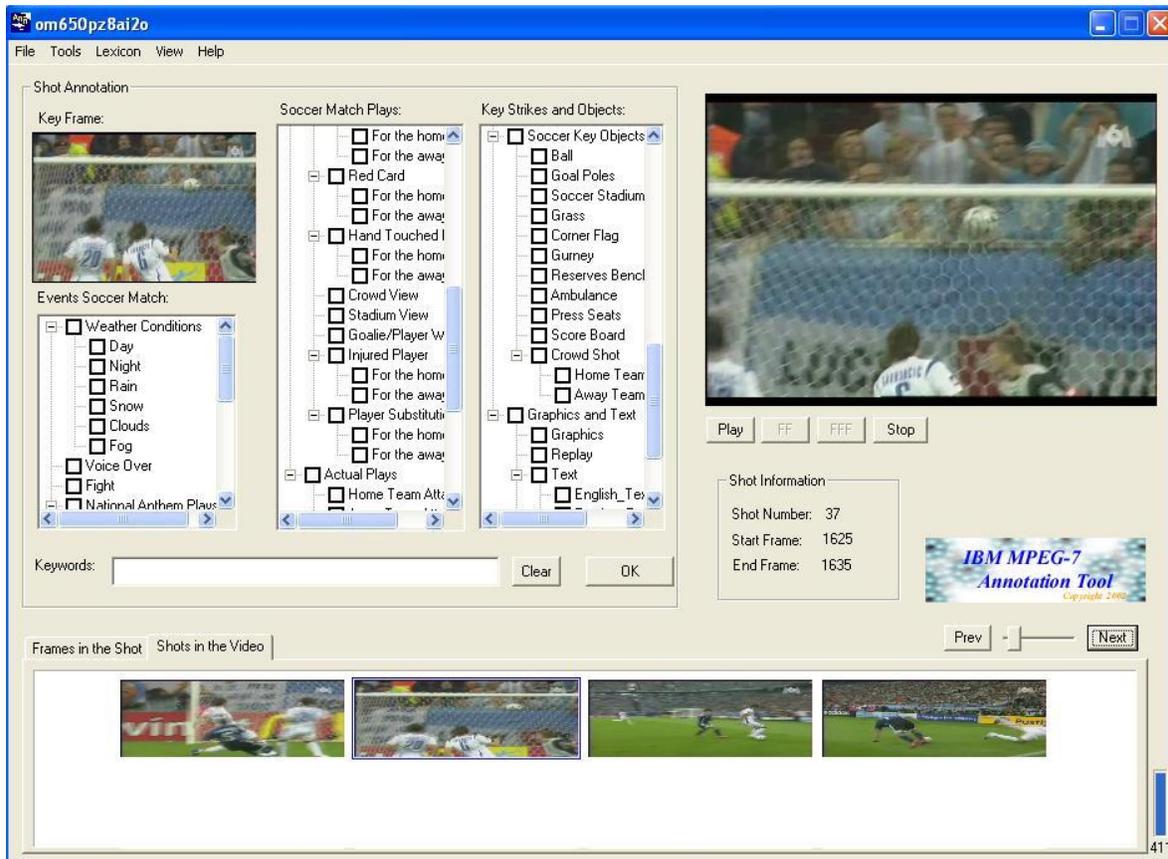


Figura 5.7. VideoAnnex utilizando a Soccer Ontology.

A partida que será utilizada neste exemplo da utilização da *Soccer Ontology* é Argentina x Servia e Montenegro, realizada no dia 16/06/2006 pela Copa do Mundo na Alemanha.

Após carregar o arquivo de vídeo da partida, que será dividida em *shots*, é aberto o lexicon da *Soccer Ontology* como mostra a imagem 5.7.

Com o lexicon carregado, pode-se iniciar a marcação do vídeo. Na primeira shot, por exemplo, tem-se uma jogada simples de passes no meio de campo, como é mostrado na figura 5.8.

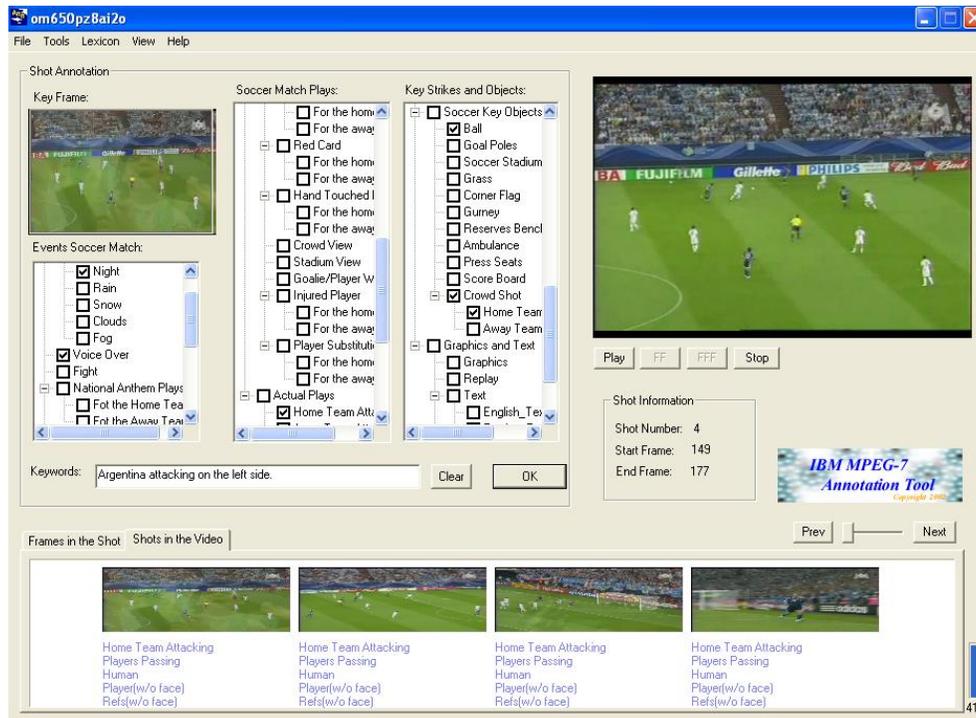


Figura 5.8. Marcação de uma partida de futebol utilizando a *Soccer Ontology*.

Selecionando então os metadados apropriados, tem-se a seguinte seqüência no arquivo final MPEG-7 para esta *shot*:

```
<TemporalDecomposition>
<VideoSegment>
<MediaTime>
<MediaTimePoint>T00:00:04:1636F30000</MediaTimePoint>
</MediaTime>
</VideoSegment>
</TemporalDecomposition>
</VideoSegment>
<VideoSegment>
<TextAnnotation type="scene" relevance="1" confidence="1">
<FreeTextAnnotation>Home Team Attacking</FreeTextAnnotation>
<FreeTextAnnotation>Players Passing</FreeTextAnnotation>
<FreeTextAnnotation>Human</FreeTextAnnotation>
<FreeTextAnnotation>Player(w/o face)</FreeTextAnnotation>
<FreeTextAnnotation>Refs(w/o face)</FreeTextAnnotation>
<FreeTextAnnotation>People Outside the Game</FreeTextAnnotation>
<FreeTextAnnotation>Ball</FreeTextAnnotation>
```

```
<FreeTextAnnotation>Crowd Shot</FreeTextAnnotation>  
<FreeTextAnnotation>Home Team Supporters</FreeTextAnnotation>  
<FreeTextAnnotation>Weather Conditions</FreeTextAnnotation>  
<FreeTextAnnotation>Night</FreeTextAnnotation>  
<FreeTextAnnotation>Voice Over</FreeTextAnnotation>  
</TextAnnotation>
```

O termo *TemporalDecomposition* se refere a divisão do arquivo MPEG-2 em *shots*. Esta específica *shot* se inicia no *MediaTimePoint* (Ponto de Tempo da Mídia) T00:00:04:1636F30000 ou 4 segundos e 1636 unidades de 30000. Começa então a serem apresentados os metadados que foram inseridos pelo *VideoAnnex* para esta específica *shot*, como *Players Passing*, *Home Team Attacking*, *Weather Conditions*, etc. Todos os metadados inseridos pelo *VideoAnnex* através dos painéis serão denominados como *FreeTextAnnotation* ou Anotação de Texto Livre para o arquivo MPEG-7 final.

Caso sejam inseridas informações pelo painel *Keywords* do *VideoAnnex*, por exemplo “*Argentina attacking on the right side*” (Argentina atacando pelo lado direito), no MPEG-7 de saída aparece esse texto como mais um *FreeTextAnnotation*, ou seja:

```
<FreeTextAnnotation>Argentina attacking on the left side.  
</FreeTextAnnotation>
```

Os metadados inseridos pelo painel *Keywords* são indistinguíveis dos inseridos pelos outros painéis na *Soccer Ontology*. O uso das *Keywords* é muito interessante para inserir informações específicas a uma determinada *shot*, como nomes dos jogadores envolvidos em um determinado lance, ação específica do árbitro, o jogador que foi expulso, o que foi punido com o cartão amarelo, o que fez o gol e diversas outras situações que ocorrem em uma partida e que não podem ser descritos com os metadados já presentes na ontologia. É muito interessante ao usuário utilizar a opção *Annotation Learning* e na primeira *shot*, marcar os metadados que serão utilizados em todas as outras *shots*, facilitando seu trabalho

e evitando que ele tenha que selecionar os mesmos metadados repetidas vezes.

Mais um exemplo do trabalho com a *Soccer Ontology*, agora com o momento mais importante de uma partida, o gol. Neste caso, o primeiro gol da Argentina contra a Servia e Montenegro. Com a *Soccer Ontology* carregada, a específica *shot* em que ocorre o gol pode ser marcada, como mostra a figura 5.9.

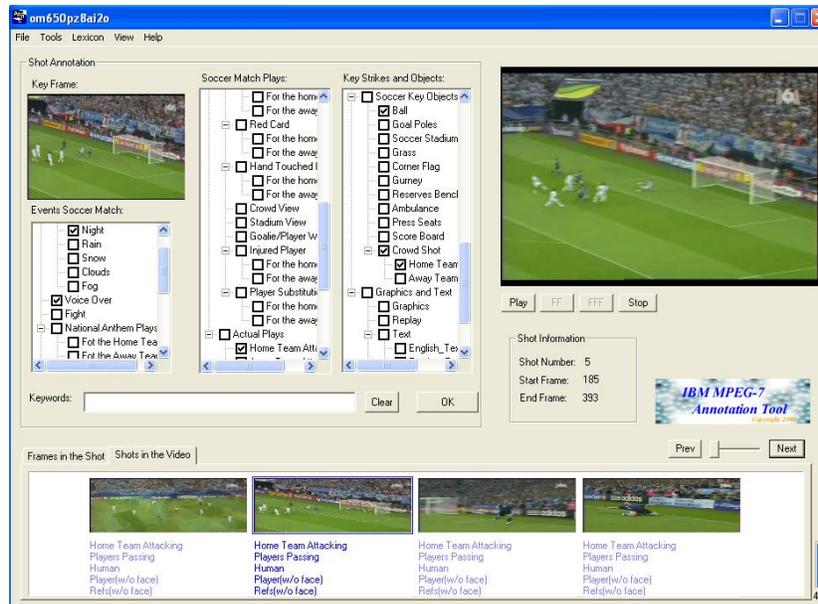


Figura 5.9. Descrevendo um gol com a *Soccer Ontology*.

Para este gol especificamente, foram escolhidos para descrevê-lo os seguintes metadados: *Weather Conditions*, *Night*, *Voice Over* (narração do jogo), *Home Team Attacking*, *Players Passing*, *GOAL*, *For the Home Team*, *Players Celebrating*, *Human*, *Player (w/side view face)*, *Player (w/face)*, *Refs (w/face)*, *People Outside the Game*, *Soccer Key Objects*, *Ball* e por fim *Goal Poles*. Também foi inserida uma anotação utilizando o painel *Keywords* para a indicação do jogador que marcou o gol, Max Rodriguez. Esses metadados irão aparecer no arquivo MPEG-7 final como:

```
<TemporalDecomposition>
  <VideoSegment>
    <MediaTime>
      <MediaTimePoint>T00:00:05:12162F30000</MediaTimePoint>
```

```

    </MediaTime>
  </VideoSegment>
</TemporalDecomposition>
  </VideoSegment>
<VideoSegment>
<TextAnnotation type="scene" relevance="1" confidence="1">
<FreeTextAnnotation>Home Team Attacking</FreeTextAnnotation>
<FreeTextAnnotation>Players Passing</FreeTextAnnotation>
<FreeTextAnnotation>Human</FreeTextAnnotation>
<FreeTextAnnotation>Player (w/o face)</FreeTextAnnotation>
<FreeTextAnnotation>Refs (w/o face)</FreeTextAnnotation>
<FreeTextAnnotation>People Outside the Game</FreeTextAnnotation>
<FreeTextAnnotation>Ball</FreeTextAnnotation>
<FreeTextAnnotation>Crowd Shot</FreeTextAnnotation>
<FreeTextAnnotation>Home Team Supporters</FreeTextAnnotation>
<FreeTextAnnotation>Weather Conditions</FreeTextAnnotation>
<FreeTextAnnotation>Night</FreeTextAnnotation>
<FreeTextAnnotation>Voice Over</FreeTextAnnotation>
<FreeTextAnnotation>Max Rodriguez scores for Argentina.
1x0.</FreeTextAnnotation>
</TextAnnotation>

```

Percebe-se que com esses metadados, foi feita uma descrição diversa dos eventos envolvendo o gol. Inclusive é facilitada uma busca posterior em bancos de dados pois todos os metadados inseridos no arquivo MPEG-7 foram inseridos como *FreeTextAnnotation*.

Agora, é apresentado como é feita a marcação de um evento estático, no caso um jogador Sérvio é expulso, como é apresentado na figura 5.10.

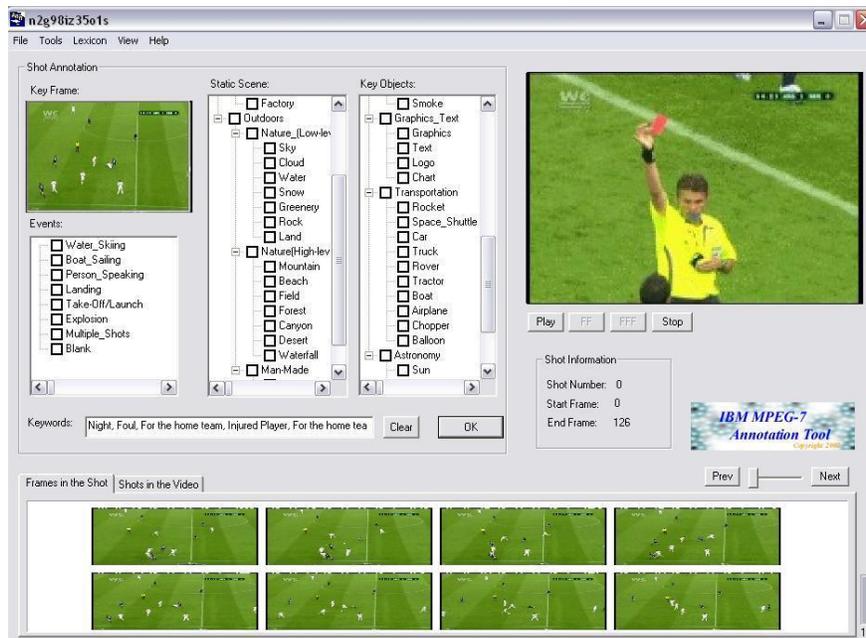


Figura 5.10. Marcando um evento estático: jogador é expulso.

Percebe-se pela *shot* o momento da expulsão do jogador. Para esse evento em particular, foram escolhidos os seguintes metadados para marcação: *Weather Conditions*, *Night*, *Voice Over* (esses três metadados são inseridos em todas as *shots*, os dois primeiros indicam as características ambientais da partida que não se alteraram durante o jogo, e o último que o jogo está sendo narrado), *Red Card*, *For the Away Team*, *Human*, *Player(w/ side view face)*, *Player(w/o face)*, *Refs(w/ front view face)* e utilizando o painel *Keywords*, *Mateja Kesman expelled from the match*, que indica o jogador expulso. Após a marcação, o arquivo resultante MPEG-7 tem para essa específica *shot*:

```
<MediaTime>
  <MediaTimePoint>T00:01:35:26874F30000</MediaTimePoint>
</MediaTime>
</VideoSegment>
</TemporalDecomposition>
</VideoSegment>
<VideoSegment>
<TextAnnotation type="scene" relevance="1" confidence="1">
  <FreeTextAnnotation>Red Card</FreeTextAnnotation>
  <FreeTextAnnotation>For the away team</FreeTextAnnotation>
  <FreeTextAnnotation>Human</FreeTextAnnotation>
  <FreeTextAnnotation>Player(w/side view face)</FreeTextAnnotation>
  <FreeTextAnnotation>Player(w/o face)</FreeTextAnnotation>
  <FreeTextAnnotation>Refs(w/front view face)</FreeTextAnnotation>
  <FreeTextAnnotation>Weather Conditions</FreeTextAnnotation>
  <FreeTextAnnotation>Night</FreeTextAnnotation>
  <FreeTextAnnotation>Voice Over</FreeTextAnnotation>
  <FreeTextAnnotation>Mateja Kesman expelled from the
match.</FreeTextAnnotation>
</TextAnnotation>
```

Essas são apenas algumas das aplicações da *Soccer Ontology*, os eventos de um jogo de futebol podem ocorrer de diversas maneiras, mas com a ontologia esses eventos podem ser descritos sem dificuldades para os usuários, exceto nos casos em que se faz necessária a inserção dos metadados diretamente no arquivo MPEG-7, exceção que é explicada na próxima seção.

5.3 Inserindo Metadados Diretamente no Arquivo MPEG-7

Como já foi visto, a utilização da *Soccer Ontology* permite a marcação da grande maioria dos eventos ocorridos em uma partida de futebol, mas existem metadados mais complexos que necessitam ser inseridos diretamente no arquivo final MPEG-7. Alguns exemplos são: nome do estádio em que ocorre a partida, escalação dos jogadores, data do jogo, etc. Esses metadados podem ser inseridos pelo painel *Keywords*, mas não se tem uma descrição apropriada utilizando esse método.

Portanto, para esses metadados que são muito importantes para uma partida de futebol, será apresentado como é possível inseri-los diretamente no arquivo MPEG-7.

5.3.1 Desenvolvimento dos Metadados no Ambiente XML

Em primeiro lugar, é necessário indicar o local onde a partida está sendo realizada. No caso do exemplo utilizado, se trata de um estádio da Copa do Mundo. Portanto, para a formatação XML e a rotina *Place*, tem-se:

```
<Place>
  <Name xml:lang="en">Fifa_World_Cup_Stadium_Gelsenkirchen</Name>
  <Role><Name>Fifa World Cup Stadium Gelsenkirchen</Name></Role>
</Place>
```

Percebe-se que o usuário pode simplesmente mudar o nome do estádio neste modelo para qualquer local desejado. Essa informação pode também ser inserida pelo painel *Keywords* como uma *Free Text Annotation* sem muitas dificuldades.

Agora, serão indicados os 11 jogadores titulares de ambos os times. Percebe-se que essa informação seria trabalhosa para ser inserida pelo painel *Keywords*, portanto é mais

interessante inserir esse importantíssimo conjunto de metadados diretamente no arquivo MPEG-7. Pela formatação XML, utilizando a rotina *PersonGroup* [5] descrita anteriormente, para a partida exemplo, tem-se:

```
<PersonGroup>
  <Name>Home Team</Name>
  <Kind><Name>Soccer Team</Name></Kind>
  <Member xsi:type="PersonType">
    <FirstName>Roberto</FirstName>
    <LastName>Abbondanzieri</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Nicolas</FirstName>
    <LastName>Burdisso</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Roberto</FirstName>
    <LastName>Ayala</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Gabriel</FirstName>
    <LastName>Heinze</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Juan</FirstName>
    <LastName>Sorin</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Maxi</FirstName>
    <LastName>Rodriguez</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Javier</FirstName>
    <LastName>Mascherano</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Luis</FirstName>
    <LastName>Gonzalez</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Juan</FirstName>
    <LastName>Riquelme</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Javier</FirstName>
    <LastName>Saviola</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Hernan</FirstName>
    <LastName>Crespo</LastName>
  </Member>
</PersonGroup>
```

Com isso, tem-se a escalação do time que foi considerado o mandante (*Home Team*), a Argentina, em uma rotina XML. Agora para os visitantes, utilizando a mesma rotina, tem-se para a Sérvia e Montenegro:

```
<PersonGroup>
  <Name>Away Team</Name>
  <Kind><Name>Soccer Team</Name></Kind>
  <Member xsi:type="PersonType">
    <FirstName>Dragoslav</FirstName>
    <LastName>Jevric</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Igor</FirstName>
    <LastName>Dul'jaj</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Goran</FirstName>
    <LastName>Gavrancic</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Mladen</FirstName>
    <LastName>Krstajic</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Milan</FirstName>
    <LastName>Dudic</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Ognjem</FirstName>
    <LastName>Koroman</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Dejan</FirstName>
    <LastName>Stankovic</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Albert</FirstName>
    <LastName>Nadj</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Predrag</FirstName>
    <LastName>Djorjevic</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Mateja</FirstName>
    <LastName>Kezman</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Savo</FirstName>
    <LastName>Milosevic</LastName>
  </Member>
</PersonGroup>
```

E assim foram inseridas as escalações iniciais dos dois times que disputaram a partida utilizada como exemplo. Podem ser inseridas informações como substituições, também, caso seja de interesse do usuário.

Agora, serão indicados os nomes dos responsáveis pela partida, o arbitro e seus assistentes. Novamente, a rotina *PersonGroup* é interessante de ser utilizada aqui.

```
<PersonGroup>
  <Name>Referee</Name>
  <Member xsi:type="PersonType">
    <FirstName>Roberto</FirstName>
    <LastName>Rossetti</LastName>
  </Member>
  <Name>Assistants</Name>
  <Member xsi:type="PersonType">
    <FirstName>Cristiano</FirstName>
    <LastName>Copelli</LastName>
  </Member>
  <Member xsi:type="PersonType">
    <FirstName>Alessandro</FirstName>
    <LastName>Stagnoli</LastName>
  </Member>
</PersonGroup>
```

Para finalizar com a rotina *PersonGroup*, são indicados os nomes dos treinadores das duas equipes disputando a partida no arquivo MPEG-7:

```
<PersonGroup>
<Name>Home Team</Name>
  <Kind><Name>Coach</Name></Kind>
  <Member xsi:type="PersonType">
    <FirstName>Jose</FirstName>
    <LastName>Pekerman</LastName>
  </Member>
<Name>Away Team</Name>
<Kind><Name>Coach</Name></Kind>
  <Member xsi:type="PersonType">
    <FirstName>Ilija</FirstName>
    <LastName>Petkovic</LastName>
  </Member>
</PersonGroup>
```

A data e horário da partida é uma informação imprescindível que deve ser inserida no arquivo MPEG-7, seja pelo painel *Keywords* ou diretamente.

Utilizando a rotina *datetime* para a indicação da data, tem-se em XML:

```
<event>
  <title>Argentina x Serbia and Montenegro WC Match</title>
  <abstract>Date and time of the match</abstract>
  <dateTime>June 16, 2006 at 3:00 PM</dateTime>
  <duration>2 hours</duration>
</event>
```

A rotina *datetime* facilita a inserção de datas em XML, por ser mais intuitiva ao usuário. Não é obrigatório o uso do formato ISO de datas, como o utilizado na programação da *Soccer Ontology*, mas para facilitar posterior a leitura mecânica por um computador, seu uso é recomendado. Portanto, utilizando o formato ISO para datas tem-se:

```
<event>
  <title>Argentina x Serbia and Montenegro WC Match</title>
  <abstract>Date and time of the match</abstract>
  <dateTime>2006-06-16T15:00:0+01:00</dateTime>
  <duration>PT2H</duration>
</event>
```

Por fim, as estatísticas da partida como número de faltas, cartões amarelos, vermelhos etc. também podem ser inseridos no arquivo MPEG-7, utilizando a rotina de designação de atributos já mencionada durante a discussão teórica de definições de tipo derivado. Portanto, tem-se em XML:

```
<Stats> <!Statistics for the Game>

  <attributeGroup name="Game_Stats_Home_Team">

    <attributename="Number_of_Fouls" type="integer"/>

    <attributename="Number_of_Yellow_Cards" type="integer"/>

    <attributename="Number_of_Red_Cards" type="integer"/>

    <attributename="Number_of_Offsides" type="integer"/>

    <attributename="Number_of_Shots" type="integer"/>

  </attributeGroup>
```

```
<attributeGroup name="Game_Stats_Away_Team">
    <attributename="Number_of_Fouls" type="integer"/>
    <attributename="Number_of_Yellow_Cards" type="integer"/>
    <attributename="Number_of_Red_Cards" type="integer"/>
    <attributename="Number_of_Offsides" type="integer"/>
    <attributename="Number_of_Shots" type="integer"/>
</attributeGroup>
</Stats>
```

É possível perceber que podem ser inseridas mais estatísticas além das apresentadas acima, que são apenas exemplos de metadados interessantes a serem utilizados para a marcação de uma partida de futebol, e que não podem ser inseridos através do *VideoAnnex*.

Uma observação importante sobre a inserção direta de metadados no arquivo MPEG-7. O *VideoAnnex* não reconhece estes metadados como parte do arquivo MPEG-7, portanto se o usuário abrir o arquivo MPEG-7 pelo *VideoAnnex* com os metadados extras inseridos, a aplicação apagará esses metadados, e o trabalho será perdido. Portanto, o usuário deve inserir os metadados diretamente após finalizar o trabalho no *VideoAnnex*. É interessante também se manter um arquivo separado com estes metadados, para evitar a perda deles.

Com isso, se encerra a inserção de metadados diretamente no arquivo MPEG-7. Como foi apresentado, estes metadados são muito úteis na marcação de uma partida de futebol. E o painel *Keywords* é uma alternativa interessante para a marcação direta.

5.4 Conclusões

Neste capítulo, foram apresentadas as principais aplicações desenvolvidas para esta tese utilizando o conceito de metadados, semântica e ontologias. A partir da utilização da aplicação IBM *VideoAnnex*, foi demonstrado em primeiro lugar como o usuário pode utilizá-la para a inserção de metadados previamente programados em um arquivo de vídeo, criando um arquivo MPEG-7. Esses metadados são do nível de semântica elevada, e necessitam de um usuário humano para a inserção [38]. Essa é a aplicação principal do *VideoAnnex* e seu principal uso nesta tese.

Com os metadados programados originalmente no *VideoAnnex*, qualquer tipo de aplicação mais avançada é impossível, pois esses metas são extremamente genéricos, não destinados a nenhuma aplicação mais específica. Por esse motivo, foi apresentado como se deve desenvolver um lexicon que contenha os metadados apropriados para uma determinada aplicação seguindo o conceito de ontologias.

Aplicando esse conceito, finalmente tem-se o ponto central desta tese: o desenvolvimento e utilização da *Soccer Ontology*, a ontologia desenvolvida especificamente para permitir a marcação de uma partida de futebol, considerando todos os fatores envolvidos durante essa partida. Foi apresentado neste capítulo como foi desenvolvida a *Soccer Ontology*, do início de sua evolução a sua forma atual, sua programação em DDL, e como foi aplicada, utilizando o *VideoAnnex* como base, através da utilização de diversos exemplos.

Por fim, foram apresentados metadados relacionados ao futebol que devem ser inseridos diretamente no arquivo MPEG-7 devido a limitações inerentes do *VideoAnnex*.

A *Soccer Ontology* não se destina a ser a aplicação final para a marcação de partidas de futebol. O fato de alguns metadados precisarem ser inseridos diretamente no arquivo MPEG-7 indica que o programa ainda precisa de melhorias, em especial a interface do *VideoAnnex*, que infelizmente não foi atualizada pela IBM depois de seu desenvolvimento inicial o que limita os metadados possíveis de serem inseridos.

Mas para a marcação da grande maioria das partidas, a *Soccer Ontology* representa uma opção viável e muito interessante, devido ao seu foco e possibilidade de marcações de nível de semântica elevado. A *Soccer Ontology* também é extremamente flexível, o que é especialmente importante no atual fluxo de produção e liberação de vídeos digitais.

O desenvolvimento da *Soccer Ontology* é especialmente atual e oportuno, devido a recentes mudanças na área de HDTV e TV Digital e o início da transmissão e exibição de partidas em forma digital, como na Copa do Mundo de 2006. Essa tendência só tende a aumentar nos anos vindouros, com a maior disponibilização de arquivos de mídia pela Internet e outras fontes, e a disseminação da TV Digital.

Capítulo 6 Conclusões e Trabalhos Futuros

O trabalho com a programação e utilização de metadados para a descrição de mídia continua aumentando de forma contínua, e o uso de metadados com um nível elevado de semântica que permitam uma descrição mais eficaz e interessante de um arquivo de mídia é um fato comprovado. Mas a sua complexidade de programação e alto custo de utilização são obstáculos que os programadores de metadados enfrentam freqüentemente. Devido a essas dificuldades, novos meios de utilização para esses metadados estão atualmente sendo desenvolvidos em diversos centros de pesquisa em todo o mundo.

6.1 Conclusões

Este trabalho apresentado tem o objetivo de apresentar o desenvolvimento de uma ontologia específica, a *Soccer Ontology*, e sua utilização em conjunto com a *IBM VideoAnnex Annotation Tool* como uma alternativa na descrição em alto nível semântico de uma partida de futebol. A utilização do *IBM VideoAnnex* como base da *Soccer Ontology* se justifica devido a sua fácil disponibilidade via Internet, ser uma ferramenta gratuita e de grande facilidade de uso comparada com outras aplicações baseadas no formato MPEG-7 [22].

No capítulo 2, os metadados foram exibidos e suas definições e aplicações indicadas. Neste capítulo também foram demonstrados os problemas no uso discriminado de metadados para a marcação de vídeos. Entre eles tem-se seu alto custo, dificuldade de

programação e o gap semântico.

Os metadados de alto nível semântico permitem uma descrição abstrata dos eventos de vídeo, sendo muito mais apropriados, mas também problemáticos em seu uso. A programação desse tipo de metadado é extremamente cara e geralmente demorada [39]. Para diminuir esses problemas, é introduzido o conceito de ontologias, o que forma a base do desenvolvimento da *Soccer Ontology*.

Tendo o conceito das ontologias, como a *Soccer Ontology* foi desenvolvida é de grande interesse para a programação de outras ontologias. As informações teóricas sobre o padrão MPEG-7, padrão do uso de metadados nesta tese, e sobre o DDL, linguagem de programação utilizada para o desenvolvimento da *Soccer Ontology*, se fazem necessárias e são devidamente indicadas nos capítulos 3 e 4.

Por fim, no capítulo 5 é apresentada a *Soccer Ontology*. Com sua utilização, a marcação de uma partida de futebol é enormemente facilitada, permitindo aos usuários que descrevam a partida com um alto nível semântico, e criando um arquivo MPEG-7 com os diversos metadados já inseridos. Os exemplos de utilização da *Soccer Ontology* em conjunto com a *IBM VideoAnnex Annotation Tool* neste capítulo apresentam sua facilidade de uso, além de sua flexibilidade. Isso ocorre porque a *Soccer Ontology* permite ao usuário que acrescente ou remova metadados da ontologia de acordo com suas necessidades. Isso é muito importante pois o desenvolvimento de novos metadados é um fator constante que deve ser levado em conta durante o desenvolvimento de uma ontologia.

Com os exemplos da utilização da *Soccer Ontology*, as vantagens do uso de ontologias ficam claras. Desenvolvendo a ontologia com a marcação de uma partida de futebol, foi possível desenvolver uma série de metadados específicos a esse esporte, permitindo que o usuário realize uma descrição mais completa da partida, e também a ligação com outros

arquivos de mídia marcados pela mesma ontologia.

A escolha do futebol para a descrição pela ontologia desenvolvida se explica pelo advento das transmissões digitais de partidas em alta definição, como o ocorrido na Copa da Alemanha em 2006. Isto expõe a necessidade da descrição dos arquivos de mídia para futura referência. A popularidade comprovada desse esporte por todo o mundo também foi um grande fator na escolha. A *Soccer Ontology* foi desenvolvida para que qualquer usuário com um computador de configuração básica e a aplicação gratuita *IBM VideoAnnex Annotation Tool* instalada possam utilizá-la.

6.2 Trabalhos Futuros

Durante o desenvolvimento desta tese, foi possível visualizar algumas possibilidades para trabalhos futuros nesta área.

Uma das possibilidades seria o desenvolvimento de um banco de dados que armazenasse os arquivos MPEG-7 após a marcação dos mesmos, e permitisse a recuperação rápida desses arquivos a partir da busca por um determinado metadado. Por exemplo, um usuário tem interesse em visualizar os gols marcados por um determinado jogador, bastaria inserir o nome do jogador e o metadado “Goals” no mecanismo de busca, e os arquivos MPEG-7 correspondentes seriam exibidos. Obviamente, esse banco de dados seria específico para a *Soccer Ontology*, mas essa idéia pode ser utilizada em diversas outras ontologias.

Outra possibilidade seria o desenvolvimento de outras ontologias baseadas na criação da *Soccer Ontology*, o que permitiria que um conjunto maior de arquivos de vídeo fosse

descrito com metadados em alto nível semântico.

O desenvolvimento de uma aplicação que permitisse a diminuição do tempo de descrição de um vídeo, um dos maiores problemas do uso de metadados de alto nível semântico, também seria algo desejado. A utilização de inteligência artificial para o reconhecimento automático pela máquina das figuras sendo exibidas no vídeo e em seguida a escolha do metadado apropriado seria uma solução interessante. Obviamente, o usuário humano ainda seria necessário para verificações, entretanto a diminuição significativa no tempo que o usuário de metadados atualmente leva na descrição de um arquivo de vídeo seria muito útil. Algumas idéias nesse sentido já estão sendo desenvolvidas, como a apresentada em [18], mas sua concepção ainda está na fase inicial e distante de uma aplicação prática.

Uma última sugestão para um futuro trabalho seria o desenvolvimento de outras ontologias para diversas atividades, como para outros esportes ou para outras áreas de estudo como a área biomédica, arte, arquitetura, etc. Com mais ontologias se tornando disponíveis, a marcação de arquivos de mídia se torna mais específica e interessante para aplicações avançadas utilizando os metadados de semântica elevada [40].

Referências Bibliográficas

- [1] Hunter, J. ‘*MPEG-7: Behind the Scenes*’. *D-LIB Magazine*, volume 5, número 9, Setembro 1999.
- [2] Martinez, J. M., Koenen, R., Pereira, F. ‘*MPEG-7: the generic Multimedia Content Description Standard*’. IEEE 2002.
http://www.img.lx.it.pt/~fp/cav/ano2005_2006/Additional_material/MPEG7_overview_2.pdf
- [3] Tannenbaum, A. ‘*Metadata Solutions*’. New Jersey: Addison-Wesley, 2001.
- [4] Martínez, J. M. (UAM-EPS-GTI, ES). ‘*Introducing MPEG-7 DDL – A Overview*’. ISO – *International Standardization Society*, Outubro 2005.
<http://www.chiariglione.org/mpeg/technologies/mp07-ddl/index.htm>.
- [5] Manjunath, B. S., Salembier, P, Sikora, T. ‘*Introduction to MPEG-7, Multimedia Content Description Interface*’. England: John Wiley & Sons, 2002.
- [6] Sonera Medialab. ‘*MPEG-7 White Paper*’. Outubro 2003.
www.medialab.sonera.fi/workspace/MPEG7WhitePaper.pdf
http://www.img.lx.it.pt/~fp/cav/ano2005_2006/Additional_material/MPEG7_overview_2.pdf
- [7] Ganguly, P. Rabhi, F. A., Ray, P. K. ‘*Bridging Semantic Gap*’. Third Asian Pacific Conference on Pattern Languages of Programs, Australia, 2002.
<http://crpit.com/confpapers/CRPITV13Ganguly.pdf>
- [8] Vembu, S., Kiesel, M., Sintek, M., Baumann, S. ‘*Towards Bridging the Semantic Gap in Multimedia Annotation and Retrieval*’. Knowledge Management Lab, Alemanha, 2005.
- [9] Benitez, A.B., Rising, H., Jorgensen, C., Leonardi, R., Bugatti, A., et. al. ‘*Semantics of Multimedia in MPEG-7*’. Proceedings of the IEEE International Conference on Image Processing, EUA, 2002.

- [10] Garcia, R., Celma, O. '*Semantic Integration and Retrieval of Multimedia Metadata*'. Proceedings of the Fifth International Workshop on Knowledge Markup and Semantic Annotation at the Fourth International Semantic Web Conference, Irlanda, 2005.
- [11] Decker, S., Harmelen, F., Broekstra, J., Erdmann, M., Fensel, D., et. al. '*The Semantic Web: The Roles of XML and RDF*'. Proc. 11th International World Wide Web Conference, ACM Press, 2002.
<http://www.ontoknowledge.org/oil/down1/IEEE00.pdf>
- [12] Chang, S.F., Sikora, T., Puri, A. '*Overview of the MPEG-7 Standard*'. IEEE Transactions on Circuits and Systems for Video Technology, 2001.
<http://ieeexplore.ieee.org/iel5/76/20050/00927421.pdf>
- [13] Smith, J. R. '*MPEG-7 Multimedia Content Description Standard*'. Multimedia Information Retrieval and Management, Part 1, Springer Verlag, 2003.
<http://www.cic.uiuc.edu/groups/VideoWorkingGroup/archive/MeetingNotes/Mpeg7Report.ppt>
- [14] *IBM VideoAnnex Annotation Tool User Manual*.
<http://www.research.ibm.com/VideoAnnEx/>
- [15] *MPEG-7 Requirements Document (v. 15)*, ISO IEC Standard JTC1/SC29/WG11/N4327, ISO/IEC, 2001.
- [16] ISO/IEC 15938-8:2002, '*Multimedia Content Description Interface – Part 8: Extraction and Use of MPEG-7 Descriptions*'. 2002.
- [17] '*XML Schema Part 0: Primer*'. W3C Candidate Recommendation. Maio de 2001
<http://www.w3.org/TR/xmlschema-0/>
- [18] Chiriaglione, L. '*The Challenges of Multimedia Standardization*'. IEEE Multimedia, 1997.
<http://doi.ieeecomputersociety.org/10.1109/93.591141>

- [19] Dublin Core Metadata Initiative.
<http://dublincore.org>
- [20] Pease, A., Niles, I., Li, J. '*Origins of the IEEE Standard Upper Ontology*'. Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Canadá, 2002.
<http://www.ontologyportal.org/pubs/IJCAI2001.pdf>
- [21] DDL FAQ.
<http://xml.coverpages.org/mpeg7-dll-faq.doc>
- [22] Day, N. '*MPEG-7: Daring to Describe Multimedia Content*'. XML-Journal, Volume 1, Edição 6, 2000.
<http://xml.sys-con.com/read/40096.htm>
- [23] ISO/IEC 15938-5:2001. '*Multimedia Content Description Interface – Part 5: Multimedia Description Schemes*'.
- [24] MPEG MDS Group. '*MPEG-7 Multimedia Description Schemes XM (v 8.0)*'. ISO/MPEGN4243, MPEG Meeting, Cingapura, 2001.
- [25] Hunter, J. '*Information technology – Multimedia Content Description Interface – Part 2: Description Definition Language*'. Maio 2005.
http://www.image.ntua.gr/oresteia/private/technical_docs/XML/ddl-w3702.pdf
- [26] '*XML: Extensible Markup Language 1.0*'. W3C Candidate Recommendation. Novembro de 1999.
<http://www.w3.org/TR/REC-xml/>
- [27] Hunter, J., DSTC. '*A Proposal for an MPEG-7 DDL*'. Test and Evaluation Meeting, Lancaster, 1999.
- [28] '*MPEG-7 DDL Response to XML Schema Last Call for Review*'. Maio de 2000.
<http://archive.dstc.edu.au/mpeg7-ddl/issues.html>

- [29] ‘*Namespaces in XML*’. W3C Recommendation. Janeiro de 1999.
<http://www.w3.org/TR/REC-xml-names/>
- [30] ‘*XML Schema Part 2: Data Types*’. W3C Candidate Recommendation. Outubro de 2000.
<http://www.w3.org/TR/xmlschema-2/>
- [31] Hunter, J. ‘*Adding Multimedia to the Semantic Web - Building a MPEG-7 Ontology*’. Proceedings of the International Semantic Web Working Symposium, Stanford University, EUA, 2001.
- [32] AlphaWorks. ‘*IBM MPEG-7 Annotation Tool*’, 2003.
<http://www.alphaworks.ibm.com/tech/videoannex>
- [33] Whitaker, J. C. *DTV Handbook – The Revolution in Digital Video*, 3rd Edition. USA: McGraw-Hill, 2001.
- [34] Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D. ‘*Object recognition as machine translation: Learning a Lexicon for a Fixed Image Vocabulary*’. Proceedings of the Seventh European Conference on Computer Vision, Dinamarca, 2002.
- [35] ISO 8601, Representações de Datas e Horas. Junho de 1988.
- [36] Garcia, R., Celma, O. ‘*Semantic Integration and Retrieval of Multimedia Metadata.*’ Proceedings of the Fifth International Workshop on Knowledge Markup and Semantic Annotation at the Fourth International Semantic Web Conference, Irlanda, 2005.
<http://www.iua.upf.edu/mtg/publications/d450b9-ISWC2005-GarciaCelma-SemAnnot2005.pdf>
- [37] Ryu, J., Sohn, Y., Kim, M. ‘*MPEG-7 Metadata Authoring Tool*’. Proc. 10th ACM International Conference Multimedia, ACM Press, 200.
<http://portal.acm.org/citation.cfm?id=641061&dl=GUIDE&coll=GUIDE>

- [38] Lavrenko, V., Feng, S., Manmatha, R. '*Statistical models for automatic video annotation and retrieval*'. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Canadá, 2004.
- [39] Bloehdorn, S., Petridis, K., Saatho, C., Simou, N., Tzouvaras, V., et. al. '*Semantic Annotation of Images and Videos for Multimedia Analysis*'. Proceedings of the Second European Semantic Web Conference, Grécia, 2005.
<http://www.acemedia.org/aceMedia/files/resource/eswc05.pdf>
- [40] Troncy, R. '*Integrating Structure and Semantics into Audio-Visual Documents*'. Proceedings of the Second International Semantic Web Conference, EUA, 2003.

Anexo I

Abaixo, versão aprimorada do Lexicon que é disponibilizado junto com o *VideoAnnex*.
<!...> indica comentários.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:example:MPEG7">
      <Header xsi:type="DescriptionMetadataType">
        <Creator>
          <Role href="creatorCS">
            <Name>Creator</Name>
          </Role>
          <Agent xsi:type="OrganizationType">
            <Name>IBM T. J. Watson Research Center</Name>
            <Contact xsi:type="PersonType">
              <Name>
                <GivenName>Ching-Yung</GivenName>
                <FamilyName>Lin</FamilyName>
              </Name>
              <ElectronicAddress>
                <Email>cylin@watson.ibm.com</Email>
              </ElectronicAddress>
            </Contact>
          </Agent>
        </Creator>
      <Changes>
        <Role href="changesCS">
          <Name>Changes</Name>
        </Role>
        <Agent xsi:type="OrganizationType">
          <Name>Tese DECOM</Name>
          <Contact xsi:type="PersonType">
            <Name>
              <GivenName>Luis Andre</GivenName>
              <FamilyName>Ferreira</FamilyName>
            </Name>
            <ElectronicAddress>
              <Email>luisandre1@hotmail.com</Email>
            </ElectronicAddress>
          </Contact>
        </Agent>
      </Changes>
    </Instrument>
    <Tool>
      <Name>IBM VideoAnnEx MPEG-7 Annotation Tool</Name>
    </Tool>
  </Instrument>
</Header>
```

```

<Term termID="Events">
  <Name xml:lang="en">Events</Name>
  <Term termID="Voice_Over">
    <Name xml:lang="en">Voice_Over</Name>
  </Term>
  <Term termID="Water_Skiing">
    <Name xml:lang="en">Water_Skiing</Name>
  </Term>
<Term termID="Fight">
  <Name xml:lang="en">Fight</Name>
</Term>
  <Term termID="Boat_Sailing">
    <Name xml:lang="en">Boat_Sailing</Name>
  </Term>
  <Term termID="Person(s)_Speaking">
    <Name xml:lang="en">Person(s)_Speaking</Name>
  </Term>
  <Term termID="Person(s)_Standing_Still">
    <Name xml:lang="en">Person(s)_Standing_Still</Name>
  </Term>
  <Term termID="Person(s)_Walking">
    <Name xml:lang="en">Person(s)_Walking</Name>
  </Term>
  <Term termID="Person(s)_Running">
    <Name xml:lang="en">Person(s)_Running</Name>
  </Term>
<Term termID="Person(s)_Falling">
  <Name xml:lang="en">Person(s)_Falling</Name>
</Term>
<Term termID="Person(s)_Bathing">
  <Name xml:lang="en">Person(s)_Bathing</Name>
</Term>
  <Term termID="Landing">
    <Name xml:lang="en">Landing</Name>
  </Term>
  <Term termID="Take-Off_Launch">
    <Name xml:lang="en">Take-Off/Launch</Name>
  </Term>
<Term termID="Soccer_Game">
  <Name xml:lang="en">Soccer_Game</Name>
</Term>
  <Term termID="Explosion">
    <Name xml:lang="en">Explosion</Name>
  </Term>
<Term termID="Sports_Event">
  <Name xml:lang="en">Sports_Event</Name>
</Term>
<Term termID="Rain_Storm">
  <Name xml:lang="en">Rain_Storm</Name>
</Term>
  <Term termID="Multiple_Shots">
    <Name xml:lang="en">Multiple_Shots</Name>
  </Term>
  <Term termID="Blank">
    <Name xml:lang="en">Blank</Name>
  </Term>
</Term>

```

```
<Term termID="Static_Scene">
  <Name xml:lang="en">Static Scene</Name>
  <Term termID="Outer_Space">
    <Name xml:lang="en">Outer_Space</Name>
    <Term termID="Moon">
      <Name xml:lang="en">Moon</Name>
    </Term>
    <Term termID="Mars">
      <Name xml:lang="en">Mars</Name>
    </Term>
  </Term>
  <Term termID="Indoors">
    <Name xml:lang="en">Indoors</Name>
    <Term termID="Classroom">
      <Name xml:lang="en">Classroom</Name>
    </Term>
    <Term termID="Meeting_Room">
      <Name xml:lang="en">Meeting_Room</Name>
    </Term>
    <Term termID="Laboratory">
      <Name xml:lang="en">Laboratory</Name>
    </Term>
    <Term termID="Factory">
      <Name xml:lang="en">Factory</Name>
    </Term>
  <Term termID="In-House">
    <Name xml:lang="en">In-House</Name>
  </Term>
  <Term termID="Outdoors">
    <Name xml:lang="en">Outdoors</Name>
    <Term termID="Nature__Low-level_">
      <Name xml:lang="en">Nature_(Low-level)</Name>
      <Term termID="Sky">
        <Name xml:lang="en">Sky</Name>
      </Term>
      <Term termID="Cloud">
        <Name xml:lang="en">Cloud</Name>
      </Term>
      <Term termID="Water">
        <Name xml:lang="en">Water</Name>
      </Term>
      <Term termID="Snow">
        <Name xml:lang="en">Snow</Name>
      </Term>
      <Term termID="Greenery">
        <Name xml:lang="en">Greenery</Name>
      </Term>
      <Term termID="Rock">
        <Name xml:lang="en">Rock</Name>
      </Term>
      <Term termID="Land">
        <Name xml:lang="en">Land</Name>
      </Term>
    <Term termID="Tree">
      <Name xml:lang="en">Tree</Name>
    </Term>
  </Term>
</Term>
```

```

</Term>
<Term termID="Nature_High-level_">
  <Name xml:lang="en">Nature (High-level) </Name>
  <Term termID="Mountain">
    <Name xml:lang="en">Mountain</Name>
  </Term>
  <Term termID="Beach">
    <Name xml:lang="en">Beach</Name>
  </Term>
  <Term termID="Field">
    <Name xml:lang="en">Field</Name>
  </Term>
  <Term termID="Forest">
    <Name xml:lang="en">Forest</Name>
  </Term>
  <Term termID="Canyon">
    <Name xml:lang="en">Canyon</Name>
  </Term>
  <Term termID="Desert">
    <Name xml:lang="en">Desert</Name>
  </Term>
  <Term termID="Waterfall">
    <Name xml:lang="en">Waterfall</Name>
  </Term>
</Term>
<Term termID="Man-Made">
  <Name xml:lang="en">Man-Made</Name>
  <Term termID="Road">
    <Name xml:lang="en">Road</Name>
  </Term>
<Term termID="Street">
  <Name xml:lang="en">Street</Name>
</Term>
  </Term><Term termID="Cemetery">
    <Name xml:lang="en">Cemetery</Name>
  </Term>
  <Term termID="Cityscape">
    <Name xml:lang="en">Cityscape</Name>
  </Term>
<Term termID="House">
  <Name xml:lang="en">House</Name>
  </Term>
</Term>
</Term>

<Term termID="Key_Objects">
  <Name xml:lang="en">Key Objects</Name>
  <Term termID="Animal">
    <Name xml:lang="en">Animal</Name>
    <Term termID="Deer">
      <Name xml:lang="en">Deer</Name>
    </Term>
  </Term>
  <Term termID="Fish">
    <Name xml:lang="en">Fish</Name>
  </Term>
  <Term termID="Bird">

```

```

        <Name xml:lang="en">Bird</Name>
    </Term>
    <Term termID="Duck">
        <Name xml:lang="en">Duck</Name>
    </Term>
</Term>
<Term termID="Human">
    <Name xml:lang="en">Human</Name>
    <Term termID="Person_w_frontal_face_">
        <Name xml:lang="en">Person (w/frontal_face) </Name>
    </Term>
    <Term termID="Person_w_side_view_face_">
        <Name xml:lang="en">Person (w/side_view_face) </Name>
    </Term>
    <Term termID="Person_w_o_face_">
        <Name xml:lang="en">Person (w/o_face) </Name>
    </Term>
    <Term termID="People_countable_">
        <Name xml:lang="en">People (countable) </Name>
    </Term>
    <Term termID="People_uncountable_">
        <Name xml:lang="en">People (uncountable) </Name>
    </Term>
<Term termID="Body_Parts">
    <Name xml:lang="en">Body_Parts</Name>
</Term>
<Term termID="Man-Made_Structure">
    <Name xml:lang="en">Man-Made_Structure</Name>
    <Term termID="Building">
        <Name xml:lang="en">Building</Name>
    </Term>
    <Term termID="Dams">
        <Name xml:lang="en">Dams</Name>
    </Term>
<Term termID="Dams">
    <Name xml:lang="en">Soccer_Stadium</Name>
</Term>
    <Term termID="Statue">
        <Name xml:lang="en">Statue</Name>
    </Term>
    <Term termID="Swimming_Pool">
        <Name xml:lang="en">Swimming_Pool</Name>
    </Term>
</Term>
<Term termID="Man-Made_Object">
    <Name xml:lang="en">Man-Made_Object</Name>
    <Term termID="Whiteboard">
        <Name xml:lang="en">Whiteboard</Name>
    </Term>
    <Term termID="Slide_Presentation">
        <Name xml:lang="en">Slide_Presentation</Name>
    </Term>
    <Term termID="Robot">
        <Name xml:lang="en">Robot</Name>
    </Term>
<Term termID="Bottle">

```

```
        <Name xml:lang="en">Bottle</Name>
    </Term>
    <Term termID="Flag">
        <Name xml:lang="en">Flag</Name>
    </Term>
<Term termID="Umbrella">
    <Name xml:lang="en">Umbrella</Name>
</Term>
<Term termID="Wheel">
    <Name xml:lang="en">Wheel</Name>
</Term>
<Term termID="Toy">
    <Name xml:lang="en">Toy</Name>
</Term>
<Term termID="Lantern">
    <Name xml:lang="en">Lantern</Name>
</Term>
<Term termID="Statue">
    <Name xml:lang="en">Statue</Name>
</Term>
<Term termID="Weapon">
    <Name xml:lang="en">Weapon</Name>
</Term>
</Term>
<Term termID="Nature_Object">
    <Name xml:lang="en">Nature_Object</Name>
    <Term termID="Flower">
        <Name xml:lang="en">Flower</Name>
    </Term>
    <Term termID="Tree">
        <Name xml:lang="en">Tree</Name>
    </Term>
    <Term termID="Fire">
        <Name xml:lang="en">Fire</Name>
    </Term>
    <Term termID="Smoke">
        <Name xml:lang="en">Smoke</Name>
    </Term>
</Term>
<Term termID="Graphics_Text">
    <Name xml:lang="en">Graphics_Text</Name>
    <Term termID="Graphics">
        <Name xml:lang="en">Graphics</Name>
    </Term>
    <Term termID="Text">
        <Name xml:lang="en">Text</Name>
</Term>
<Term termID="English_Text">
<Name xml:lang="en">English_Text</Name>
</Term>
<Term termID="Foreign_Text">
<Name xml:lang="en">Foreign_Text</Name>
</Term>
</Term>
    <Term termID="Logo">
        <Name xml:lang="en">Logo</Name>
    </Term>
    <Term termID="Chart">
```

```

        <Name xml:lang="en">Chart</Name>
    </Term>
</Term>
<Term termID="Transportation">
    <Name xml:lang="en">Transportation</Name>
    <Term termID="Rocket">
        <Name xml:lang="en">Rocket</Name>
    </Term>
    <Term termID="Space_Shuttle">
        <Name xml:lang="en">Space_Shuttle</Name>
    </Term>
    <Term termID="Car">
        <Name xml:lang="en">Car</Name>
    </Term>
<Term termID="Wagon">
    <Name xml:lang="en">Wagon</Name>
</Term>
<Term termID="Truck">
    <Name xml:lang="en">Truck</Name>
</Term>
<Term termID="Rover">
    <Name xml:lang="en">Rover</Name>
</Term>
<Term termID="Tractor">
    <Name xml:lang="en">Tractor</Name>
</Term>
<Term termID="Boat">
    <Name xml:lang="en">Boat</Name>
</Term>
<Term termID="Airplane">
    <Name xml:lang="en">Airplane</Name>
</Term>
<Term termID="Chopper">
    <Name xml:lang="en">Chopper</Name>
</Term>
<Term termID="Balloon">
    <Name xml:lang="en">Balloon</Name>
</Term>
</Term>
<Term termID="Astronomy">
    <Name xml:lang="en">Astronomy</Name>
    <Term termID="Sun">
        <Name xml:lang="en">Sun</Name>
    </Term>
    <Term termID="Moon">
        <Name xml:lang="en">Moon</Name>
    </Term>
    <Term termID="Earth">
        <Name xml:lang="en">Earth</Name>
    </Term>
    <Term termID="Mars">
        <Name xml:lang="en">Mars</Name>
    </Term>
    <Term termID="Jupiter">
        <Name xml:lang="en">Jupiter</Name>
    </Term>
    <Term termID="Saturn">

```

```
        <Name xml:lang="en">Saturn</Name>
      </Term>
      <Term termID="Other_Planet">
        <Name xml:lang="en">Other_Planet</Name>
      </Term>
    </Term>
  </Term>
</ClassificationScheme>
</Description>
</Mpeg7>
```

Anexo II

Abaixo, a *Soccer Ontology* em sua forma completa (<!--> indicam comentários):

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:example:MPEG7">
      <Header xsi:type="DescriptionMetadataType">
        <Creator>
          <Role href="creatorCS">
            <Name>Creator</Name>
          </Role>
          <Agent xsi:type="OrganizationType">
            <Name>Tese DECOM</Name>
            <Contact xsi:type="PersonType">
              <Name>
                <GivenName>Luis Andre</GivenName>
                <FamilyName>Ferreira</FamilyName>
              </Name>
              <ElectronicAddress>
                <Email>luisandrel@yahoo.com.br</Email>
              </ElectronicAddress>
            </Contact>
          </Agent>
        </Creator>
      </DescriptionMetadata>
      <Comment>
        <FreeTextAnnotation>Description of Soccer Match.</FreeTextAnnotation>
      </Comment>
      <CreationTime>2006-01-05T13:13:7+01:00</CreationTime>
    </DescriptionMetadata>
    <UsageInformation>
      <Rights>
        <RightsID organization="DECOM" type="4MCopyrights">Unrestricted access.
        This object may be copyright-protected.Contact the owner or rights holder
        for permission to reproduce.</RightsID>
      </Rights>
    </UsageInformation>
    <Tournament> <!--Tournament the teams are playing. Example:World Cup>
      <Name xml:lang="en">Insert_Tournament_Name_Here</Name>
      <Role><Name>Tournament being played</Name></Role>
    </Tournament>
    <Match> <!--Teams Playing in the Soccer Match>
      <Name xml:lang="en">Insert_teams_here:Home_TeamxAway_team</Name>
      <Role><Name>Teams Playing the Match</Name></Role>
    </Match>
    <Place> <!--Location of the match>
      <Name xml:lang="en">Insert_Stadium_Name_Here</Name>
      <Role><Name>Soccer Match Location</Name></Role>
    </Place>
```

```

<Date>    <!Date of the match>
    <Name xml:lang="en">Insert Date Here in month/date/year</Name>
    <Role><Name>Date of the Match</Name></Role>
</Date>
<PersonGroup><!Identify Ref and assistants>
    <Name>Referee</Name>
    <Member xsi:type="PersonType">
        <FirstName>First Name Here</FirstName>
        <LastName>Last Name Here</LastName>
    </Member>
    <Name>Assistants</Name>
    <Member xsi:type="PersonType">
        <FirstName>First Name Here</FirstName>
        <LastName>Last Name Here</LastName>
    </Member>
</PersonGroup>
<PersonGroup> <!Identify all 11 or if necessary 22 players of both
teams>
    <Name>Home Team</Name>
    <Kind><Name>Soccer Team</Name></Kind>
    <Member xsi:type="PersonType">
        <FirstName>First Name Here</FirstName>
        <LastName>Last Name Here</LastName>
    </Member>
    /Repeat for all the players of the Home team
    <Name>Away Team</Name>
    <Kind><Name>Soccer Team</Name></Kind>
    <Member xsi:type="PersonType">
        <FirstName>First Name Here</FirstName>
        <LastName>Last Name Here</LastName>
    </Member>
    /Repeat for all the players of the Away team
</PersonGroup>
<PersonGroup> <!Identify the coaches for both teams>
    <Name>Home Team</Name>
    <Kind><Name>Coach</Name></Kind>
    <Member xsi:type="PersonType">
        <FirstName>First Name Here</FirstName>
        <LastName>Last Name Here</LastName>
    </Member>
    <Name>Away Team</Name>
    <Kind><Name>Coach</Name></Kind>
    <Member xsi:type="PersonType">
        <FirstName>First Name Here</FirstName>
        <LastName>Last Name Here</LastName>
    </Member>
</PersonGroup>
<Stats> <!Statistics for the Game>
    <attributeGroup name="Game_Stats_Home_Team">
        <attributename="Number_of_Fouls" type="integer"/>
        <attributename="Number_of_Yellow_Cards" type="integer"/>
        <attributename="Number_of_Red_Cards" type="integer"/>
        <attributename="Number_of_Offsides" type="integer"/>
        <attributename="Number_of_Shots" type="integer"/>
    </attributeGroup>
    <attributeGroup name="Game_Stats_Away_Team">
        <attributename="Number_of_Fouls" type="integer"/>

```

```

    <attributename="Number_of_Yellow_Cards" type="integer"/>
    <attributename="Number_of_Red_Cards" type="integer"/>
    <attributename="Number_of_Offsides" type="integer"/>
    <attributename="Number_of_Shots" type="integer"/>
  </attributeGroup>
</Stats>
  <Instrument>
    <Tool>
      <Name>IBM VideoAnnEx MPEG-7 Annotation Tool</Name>
    </Tool>
  </Instrument>
</Header>
  <Term termID="Events_Soccer_Match">
    <Name xml:lang="en">Events Soccer Match</Name>
  <Term termID="Weather_Conditions">
    <Name xml:lang="en">Weather Conditions</Name>
    <Term termID="Day">
      <Name xml:lang="en">Day</Name>
    </Term>
    </Term><Term termID="Night">
      <Name xml:lang="en">Night</Name>
    </Term>
  <Term termID="Rain">
    <Name xml:lang="en">Rain</Name>
  </term>
  </Term><Term termID="Snow">
    <Name xml:lang="en">Snow</Name>
  </Term>
  </Term><Term termID="Clouds">
    <Name xml:lang="en">Clouds</Name>
  </Term>
  </Term><Term termID="Fog">
    <Name xml:lang="en">Fog</Name>
  </Term>
</Term>
  <Term termID="Voice_Over">
    <Name xml:lang="en">Voice Over</Name>
  </Term>
  <Term termID="Fight">
    <Name xml:lang="en">Fight</Name>
  </Term>
  <Term termID="National_Anthem_Plays">
    <Name xml:lang="en">National Anthem Plays</Name>
  <Term termID="For_The_Home_Team">
    <Name xml:lang="en">Fot the Home Team</Name>
  </Term>
  <Term termID="For_The_Away_Team">
    <Name xml:lang="en">Fot the Away Team</Name>
  </Term>
</Term>
  <Term termID="Warm-Up">
    <Name xml:lang="en">Warm-Up</Name>
  </Term>
  <Term termID="First_Half_Begins">
    <Name xml:lang="en">First Half Begins</Name>
  </Term>
  <Term termID="Half_Time">

```

```

<Name xml:lang="en">Half Time</Name>
  </Term>
<Term termID="Second_Half_Begins">
<Name xml:lang="en">Second Half Begins</Name>
  </Term>
<Term termID="End_of_the_Second_Half">
<Name xml:lang="en">End of the Second Half</Name>
  </Term>
<Term termID="Overtime">
<Name xml:lang="en">Overtime</Name>
  </Term>
<Term termID="Penalty_Kicks">
<Name xml:lang="en">Penalty Kicks</Name>
  </Term>
<Term termID="Power_Outage">
<Name xml:lang="en">Power Outage</Name>
  </Term>
<Term termID="Blank">
  <Name xml:lang="en">Blank</Name>
  </Term>
</Term>
<Term termID="Soccer_Match_Plays">
  <Name xml:lang="en">Soccer Match Plays</Name>
<Term termID="Static_Plays"> <!--Static plays in a soccer match-->
  <Name xml:lang="en">Static Plays</Name>
  <Term termID="Kick-Off">
    <Name xml:lang="en">Kick-Off</Name>
  </Term>
  <Term termID="Corner_Kick">
    <Name xml:lang="en">Corner Kick</Name>
    <Term termID="For_the_home_team">
      <Name xml:lang="en">For the home team</Name>
    </term>
    </Term><Term termID="For_the_away_team">
      <Name xml:lang="en">For the away team</Name>
    </Term>
  </term>
  <Term termID="Goal_Kick">
    <Name xml:lang="en">Goal Kick</Name>
    <Term termID="For_the_home_team">
      <Name xml:lang="en">For the home team</Name>
    </term>
    </Term><Term termID="For_the_away_team">
      <Name xml:lang="en">For the away team</Name>
    </Term>
  </term>
  </term>
<Term termID="Throw_In">
  <Name xml:lang="en">Throw In</Name>
  <Term termID="For_the_home_team">
    <Name xml:lang="en">For the home team</Name>
  </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
  </term>
  <Term termID="Foul">
    <Name xml:lang="en">Foul</Name>
  </term>

```

```
<Term termID="For_the_home_team">
  <Name xml:lang="en">For the home team</Name>
</term>
</Term><Term termID="For_the_away_team">
  <Name xml:lang="en">For the away team</Name>
</Term>
</term>
<Term termID="Two_Touch_Foul">
  <Name xml:lang="en">Two Touch Foul</Name>
  <Term termID="For_the_home_team">
    <Name xml:lang="en">For the home team</Name>
  </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</term>
<Term termID="Offside">
  <Name xml:lang="en">Offside</Name>
  <Term termID="For_the_home_team">
    <Name xml:lang="en">For the home team</Name>
  </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</term>
<Term termID="Yellow_Card">
  <Name xml:lang="en">Yellow Card</Name>
  <Term termID="For_the_home_team">
    <Name xml:lang="en">For the home team</Name>
  </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</term>
<Term termID="Red_Card">
  <Name xml:lang="en">Red Card</Name>
  <Term termID="For_the_home_team">
    <Name xml:lang="en">For the home team</Name>
  </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</term>
<Term termID="Hand_Touched_Ball">
  <Name xml:lang="en">Hand Touched Ball</Name>
  <Term termID="For_the_home_team">
    <Name xml:lang="en">For the home team</Name>
  </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</term>
<Term termID="Crowd_View">
  <Name xml:lang="en">Crowd View</Name>
  </Term>
<Term termID="Stadium_View">
  <Name xml:lang="en">Stadium View</Name>
</Term>
```

```

</Term>
<Term termID="Goalie/Player_Warnings">
  <Name xml:lang="en">Goalie/Player Warnings</Name>
</Term>
  <Term termID="Injured_Player">
    <Name xml:lang="en">Injured Player</Name>
    <Term termID="For_the_home_team">
      <Name xml:lang="en">For the home team</Name>
    </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</Term>
  <Term termID="Player_Substitution">
    <Name xml:lang="en">Player Substitution</Name>
    <Term termID="For_the_home_team">
      <Name xml:lang="en">For the home team</Name>
    </term>
  </Term><Term termID="For_the_away_team">
    <Name xml:lang="en">For the away team</Name>
  </Term>
</Term>
</Term>
<Term termID="Actual_Plays"> <!Plays that occur during a soccer
match>
  <Name xml:lang="en">Actual Plays</Name>
  <Term termID="Home_Team_Attacking">
    <Name xml:lang="en">Home Team Attacking</Name>
  </Term>
  <Term termID="Away_Team_Attacking">
    <Name xml:lang="en">Away Team Attacking</Name>
  </Term>
  <Term termID="Players_passing">
    <Name xml:lang="en">Players Passing</Name>
  </Term>
  <Term termID="Wrong_Pass">
    <Name xml:lang="en">Wrong Pass</Name>
  </Term>
  <Term termID="Long_Range_Pass">
    <Name xml:lang="en">Long Range Pass</Name>
  </Term>
  <Term termID="Head_Dispute">
    <Name xml:lang="en">Head Dispute</Name>
  </Term>
  <Term termID="Ball_Stolen">
    <Name xml:lang="en">Ball Stolen</Name>
  </Term>
  <Term termID="Goalie_Defense">
    <Name xml:lang="en">Goalie Defense</Name>
  </Term>
  <Term termID="Goalie_Pass">
    <Name xml:lang="en">Goalie Pass</Name>
  </Term>
  <Term termID="Player_Carries_The_Ball">
    <Name xml:lang="en">Player Carries The Ball</Name>
  </Term>
  <Term termID="Player_Crosses_The_Ball">

```

```

        <Name xml:lang="en">Player Crosses The Ball</Name>
    </Term>
<Term termID="Player_Loses_the_Ball">
    <Name xml:lang="en">Player Loses the Ball</Name>
    </Term>
    <Term termID="Ball_Hits_Corner_Flag">
        <Name xml:lang="en">Ball Hits Corner Flag</Name>
    </Term>
</Term>
<Term termID="Key_Strike_and_Objects">
    <Name xml:lang="en">Key Strikes and Objects</Name>
    <!Most important strikes and Objects in the game>
<Term termID="GOAL">
    <Name xml:lang="en">GOAL</Name>
    <Term termID="For_the_home_team">
        <Name xml:lang="en">For the home team</Name>
    </term>
        </Term><Term termID="For_the_away_team">
            <Name xml:lang="en">For the away team</Name>
        </Term>
<Term termID="Header">
    <Name xml:lang="en">Header</Name>
</Term>
<Term termID="Long_Range_Kick">
    <Name xml:lang="en">Long Range Kick</Name>
</Term>
    <Term termID="Medium_Range_Kick">
        <Name xml:lang="en">Medium Range Kick</Name>
    </Term>
<Term termID="Short_Range_Kick">
    <Name xml:lang="en">Close Range Kick</Name>
</Term>
<Term termID="Own_Goal">
    <Name xml:lang="en">Own Goal</Name>
</Term>
    <Term termID="Players_Celebrating">
        <Name xml:lang="en">Players Celebrating</Name>
    </Term>
<Term termID="Goal_Overturned">
    <Name xml:lang="en">Goal Overturned</Name>
</Term>
</Term>
<Term termID="Shots">
    <Name xml:lang="en">Shots</Name>
    <Term termID="For_the_home_team">
        <Name xml:lang="en">For the home team</Name>
    </term>
        </Term><Term termID="For_the_away_team">
            <Name xml:lang="en">For the away team</Name>
        </Term>
            </Term><Term termID="Over_the_goal">
                <Name xml:lang="en">Over the Goal</Name>
            </Term>
                </Term><Term termID="Off_the_Side">
                    <Name xml:lang="en">Off the side</Name>
                </Term>
                    </Term><Term termID="Net_Hit_From_the_Outside">

```

```

<Name xml:lang="en">Net Hit From the Outside</Name>
  </Term>
</Term><Term termID="Goalie_Defense">
  <Name xml:lang="en">Goalie Defense</Name>
  </Term>
</Term><Term termID="Hit_Bar_Pole">
  <Name xml:lang="en">Hit bar Pole</Name>
  </Term>
</Term><Term termID="Hit_Opposing_Player">
  <Name xml:lang="en">Hit Opposing Player</Name>
  </Term>
</Term><Term termID="Hit_Friendly_Player">
  <Name xml:lang="en">Hit Friendly Player</Name>
  </Term>
<Term termID="Human">
  <Name xml:lang="en">Human</Name>
  <Term termID="Player_w_Front_view_face_">
    <Name xml:lang="en">Player(w/front view face)</Name>
    </Term>
  <Term termID="Player_w_side_view_face_">
    <Name xml:lang="en">Player(w/side view face)</Name>
    </Term>
  <Term termID="Player_w_o_face_">
    <Name xml:lang="en">Player(w/o face)</Name>
    </Term>
  <Term termID="Reserve_Players">
    <Name xml:lang="en">Reserve Players</Name>
    </Term>
  <Term termID="Refs_w_front_view_face_">
    <Name xml:lang="en">Refs(w/front view face)</Name>
    </Term>
  <Term termID="Refs_w_side_view_face_">
    <Name xml:lang="en">Refs(w/side view face)</Name>
    </Term>
  <Term termID="Refs_w_o_face_">
    <Name xml:lang="en">Refs(w/o face)</Name>
    </Term>
  <Term termID="People_Outside_the_Game_(Coaches, Press, Assistants, etc.)">
    <Name xml:lang="en">People Outside the Game</Name>
    </Term>
  <Term termID="Player_Body_Parts">
    <Name xml:lang="en">Player Body Parts</Name>
    </Term>
  <Term termID="Player_Interview">
    <Name xml:lang="en">Player Interview</Name>
    </Term>
  <Term termID="Coach/Other_Interview">
    <Name xml:lang="en">Coach/Other Interview</Name>
    </Term>
</term>
  <Term termID="Soccer_Key_Objects">
<Name xml:lang="en">Soccer Key Objects</Name>
  <Term termID="Ball">
    <Name xml:lang="en">Ball</Name>
    </Term>
  <Term termID="Goal_Poles">

```

```
<Name xml:lang="en">Goal Poles</Name>
  </Term>
  <Term termID="Soccer_Stadium">
    <Name xml:lang="en">Soccer Stadium</Name>
  </Term>
  <Term termID="Grass">
    <Name xml:lang="en">Grass</Name>
  </Term>
  <Term termID="Corner_Flag">
    <Name xml:lang="en">Corner Flag</Name>
  </Term>
  <Term termID="Gurney">
    <Name xml:lang="en">Gurney</Name>
  </Term>
  <Term termID="Reserves_Bench">
    <Name xml:lang="en">Reserves Bench</Name>
  </Term>
  <Term termID="Ambulance">
    <Name xml:lang="en">Ambulance</Name>
  </Term>
  <Term termID="Press_Seats">
    <Name xml:lang="en">Press Seats</Name>
  </Term>
  <Term termID="Score_Board">
    <Name xml:lang="en">Score Board</Name>
  </Term>
  <Term termID="Crowd_Shot">
    <Name xml:lang="en">Crowd Shot</Name>
  </Term>
  <Term termID="Home_Team_Supporters">
    <Name xml:lang="en">Home Team Supporters</Name>
  </Term>
  <Term termID="Away_Team_Supporters">
    <Name xml:lang="en">Away Team Supporters</Name>
  </Term>
</Term>
</Term>
  <Term termID="Graphics_and_Text">
    <Name xml:lang="en">Graphics and Text</Name>
    <Term termID="Graphics">
      <Name xml:lang="en">Graphics</Name>
    </Term>
  </Term>
  <Term termID="Replay">
    <Name xml:lang="en">Replay</Name>
  </Term>
  <Term termID="Text">
    <Name xml:lang="en">Text</Name>
  </Term>
  <Term termID="English_Text">
    <Name xml:lang="en">English_Text</Name>
  </Term>
  <Term termID="Foreign_Text">
    <Name xml:lang="en">Foreign_Text</Name>
  </Term>
</Term>
  <Term termID="Logo">
    <Name xml:lang="en">Logo</Name>
  </Term>
  <Term termID="Chart">
```

```
<Name xml:lang="en">Chart</Name>
  </Term>
  <Term termID="Game_Time">
    <Name xml:lang="en">Game Time</Name>
  </Term>
  <Term termID="Yellow/Red_Cards">
    <Name xml:lang="en">Yellow/Red Cards</Name>
  </Term>
</Term>
</Term>
</ClassificationScheme>
</Description>
</Mpeg7>
```

Artigos Publicados