

SERVICÓ DE PÓS-GRADUAÇÃO DO ICMC-USP

Nome do Candidato: João Benedito dos Santos Júnior

Assinatura: João Benedito dos Santos Júnior

Modelagem de Ambientes Interativos Conscientes de Contexto: Uma Abordagem Baseada nos Padrões MPEG-4 e MPEG-7

João Benedito dos Santos Júnior

Orientador

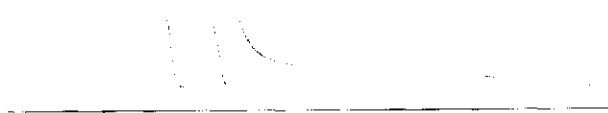
Prof. Dr. Edson dos Santos Moreira

*Tese apresentada ao Instituto de
Ciências Matemáticas e de Computação – ICMC,
da Universidade de São Paulo – USP,
como parte dos requisitos para obtenção do título de
Doutor em Ciências – Área de Ciências de Computação.*

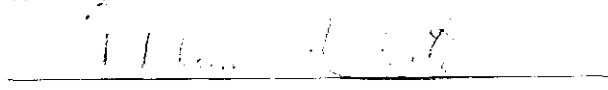
São Carlos - SP
2002

A Comissão Julgadora:

Prof. Dr. Edson dos Santos Moreira



Prof. Dr. Berthier Ribeiro de Araújo Neto



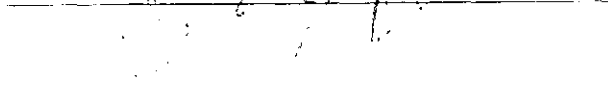
Profa. Dra. Maria da Graça Campos Pimentel



Prof. Dr. Marcelo Knörich Zuffo



Prof. Dr. Guido Lemos de Souza Filho



*Louvo a DEUS pelo dom da vida e agradeço
à minha família pela vida em comunhão com DEUS.*

*Este trabalho é fruto do que eu sou; e eu sou o fruto das
minhas origens e da minha família.*

*“Vim para a cidade, trabalhei e estudei
E por isso atualmente já me chamam de ‘doutor’
Mas o sotaque que eu sempre carreguei
Não deixa enganar que eu vim do interior*

*Mas não faz mal
Sou assim, eu sou matuto
Meu ‘ocê’ não é fajuto
Sai assim ao natural*

*E quando falo
Meu sotaque então me trai
Para surpresa e espanto
Sem querer eu digo ‘uai’*

*A vida toda eu vou falar ‘uai’, ‘uai’
A vida toda eu vou falar ‘ocê’
Assim falavam meu avô e meu pai
Sou caipira, logo já se vê...”*

José Fortuna e Marcelo Costa

*Dedico este trabalho aos meus pais,
João Benedito dos Santos e Maria de Lourdes Santos,
aos meus avós (em vida e in memoriam), aos meus irmãos e irmãs,
aos meus sobrinhos, especialmente ao **Gabriel.***

Agradecimentos

Ao bom DEUS, por me conceder saúde, por iluminar meu caminho e por me abençoar na realização de todas as etapas deste trabalho.

Ao meu orientador, o Prof. Dr. Edson dos Santos Moreira, pelo apoio, pelo incentivo constante e pela orientação. Em especial, agradeço-lhe pela liberdade que ele me permitiu na condução da minha pesquisa e pelos inúmeros e renovados votos de confiança.

À Profa. Maria da Graça Campos Pimentel, pelo apoio irrestrito quando da minha chegada ao então ICMSC-USP, em 1996.

Aos professores Edson dos Santos Moreira, Fredy João Valente, José Carlos Maldonado, Maria Cristina Ferreira de Oliveira, Maria da Graça Campos Pimentel, Sandra Maria Alúcio e Marcos José Santana por compartilharem seus conhecimentos durante o período de créditos, tanto no Mestrado quanto no Doutorado.

Agradeço aos colegas do Laboratório Intermídia, especialmente ao Luciano Martins e ao Leonardo Andrade. Um eterno agradecimento aos meus amigos Gustavo Blengini Faria e ao Rudinei Goularte, por dividirem comigo todas as responsabilidades, incertezas e conquistas da nossa pesquisa.

Às secretárias da Pós-Graduação Beth, Laura e Marília, pela amizade e presteza com que sempre me atenderam, e pelo jeito humano de tratar a todos os alunos.

A todos os colegas que compartilharam comigo os momentos de alegria e de apreensão, as vitórias e derrotas do período de créditos. À Mirlla, pela constante amizade e incentivo. À Elaine, o meu agradecimento especial pela sua amizade, preocupação e ajuda.

Ao Dr. Leonardo Chiariglione, por permitir a realização de parte desta pesquisa nos laboratórios do TILAB – Telecom Italia Lab, em Turim, Itália. Meus sinceros agradecimentos ao Dr. Mario Guglielmo pela organização da minha estadia em Turim, pela supervisão direta do trabalho e pela amizade que construímos ao longo do tempo. Agradeço também ao Dr. Gianluca De Petris, pela orientação técnica durante o período em que estive no TILAB, oferecendo-me um ambiente de pesquisa extremamente rico.

Agradecimentos especiais aos meus amigos italianos Alessandro Costa Laia e Giovanni Cordara e a seus familiares, que me ajudaram a suportar a ausência da família. Agradeço também aos colegas da TILAB Advanced Multimedia Division - Barbara Silano, Diego Gibellino, Filippo Chiariglione, Barbara Negro, Stefano Dal Lago and Tiziana Trucco – que me proporcionaram um excelente ambiente de trabalho e incentivaram a minha completa integração.

Novas formas de interação têm surgido a partir da disseminação das tecnologias associadas à multimídia interativa. O uso de objetos multimídia apresenta-se, num primeiro momento, como um bônus para muitas aplicações, porém, no contexto no qual este trabalho se insere, pretende-se investigar as possibilidades para o usuário interagir com objetos multimídia, durante a apresentação desses objetos, com características que ultrapassem os limites dos tradicionais controles VCR (Vídeo-Cassette Recorder). Adicionalmente, a apresentação de um vídeo interativo pode ser influenciada e/ou modificada por eventos que ocorrem no ambiente no qual a aplicação multimídia estiver inserida. Neste cenário, o ambiente – visto neste trabalho como sendo a infra-estrutura de comunicação, a aplicação, as mídias e os terminais para interação – pode exigir adaptações na interface do usuário e o ajuste nos parâmetros de distribuição e consumo dos streams elementares de mídia. Tecnologias, técnicas e padrões têm sido produzidos para viabilizar o desenvolvimento dessas novas aplicações multimídia; entretanto, muitas dessas novas formas de interação continuam inexploradas e não formalizadas.

O principal objetivo desta tese¹ é apresentar uma proposta para a modelagem de ambientes interativos conscientes de contexto, utilizando a linguagem UML (Unified Modelling Language), tendo como infra-estrutura tecnológica os padrões MPEG-4 e MPEG-7 da família de padrões MPEG (Moving Picture Expert Group) e considerando as informações contextuais como uma forma para facilitar e incrementar a interação do usuário com essas novas aplicações multimídia interativas.

O trabalho descreve um cenário para aplicações interativas com base nos padrões MPEG-4 e MPEG-7, no qual seja possível aplicar os conceitos de consciência de contexto como forma para oferecer suporte ao desenvolvimento de novos sistemas multimídia interativos. Neste sentido, novos critérios de interação foram estabelecidos a partir da distribuição de conteúdo multimídia (representado por objetos MPEG-4) e informação associada ao conteúdo, descrita por esquemas MPEG-7. O futuro padrão MPEG-21 está sendo investigado como potencial tecnologia para representar esses novos requisitos multimídia como itens digitais universais. Os principais conceitos de consciência de contexto foram estendidos e adicionados à modelagem proposta também para prover personalização do ambiente. O conceito de perfil de aplicação interativa foi formalizado e discutido. Como forma de validação da proposta, uma aplicação de Televisão Interativa (TVI) foi formalizada e apresentada como estudo de caso, possibilitando a discussão das limitações presentes na modelagem proposta.

¹ Parte deste trabalho foi realizada nos laboratórios da Telecom Italia Lab – TI LAB (que era conhecido como CSELT – Centro di Studi e Laboratorio Telecomunicazioni), junto a Advanced Multimedia Division, em Turim, Itália, no período de maio a dezembro de 2001.

Abstract

New forms of interaction became possible from the dissemination of the interactive video technology. The possibility for using multimedia objects in modern applications has proven to be a bonus and the we want to provide ways in which the user can interact with multimedia objects, while the presentation is occurring, with features that go beyond the regular controls of a VCR (Video-Cassette Recorder). Furthermore, the presentation of the video can be modified by events happening in the surrounding environment. In this scenario, the environment may require the redesign of the user interface and the retargeting of the elementary streams parameters in order to match specific requirements and features of the environment (application, media, network and terminals). Several new techniques and standards have been produced in order to allow for the development of new applications; however, some potential new interaction forms are still unexplored.

The main objective of this thesis is to present a proposal for Modelling of Context-Aware Interactive Environments Based on Standards of the MPEG (Moving Picture Expert Group) Family, using the UML (Unified Modelling Language) and considering the user actions from contextual information like a way to ease its interaction with interactive applications.

The work describes a scenario for interactive applications, based on the emerging MPEG-4 and MPEG-7 standards, that uses the context-awareness concepts in order to provide the necessary support to the development of more interactive and useful systems. At this point, new interaction criteria have been established from distribution of MPEG-4 media objects and MPEG-7 scene descriptions on network environments. MPEG-21 is being studied for applying to new multimedia requirements, considering multimedia objects as universal digital item. Context-awareness aspects have been extended and added for providing personalization on the interactive environment. The concept of interactive application profile is formalized and presented. As validating mechanism, an ITV (Interactive Television) application is presented as a case of study, focalizing how and which kind of contextual information could be used in the ITV environment, allowing the discussion about the main problems of the proposed modelling.

Índice Geral

Índice de Figuras	xiii
Índice de Tabelas.....	xvi
1 Introdução.....	1
1.1 Considerações iniciais	1
1.2 Contextualização.....	3
1.3 Trabalhos relacionados	3
1.4 Metodologia.....	5
1.5 Objetivos	7
1.6 Inovações.....	9
1.7 Principal contribuição original.....	10
1.8 Estrutura do trabalho	11
2 Multimédia Interativa Distribuída.....	12
2.1 Considerações iniciais	12
2.2 Tecnologias associadas à multimédia interativa: uma breve apresentação.....	12
2.2.1 SMIL	13
2.2.2 VRML	15
2.2.3 HTML e DHTML	16
2.2.4 XML.....	18
2.2.5 XSL.....	20
2.3 A problemática da distribuição de multimédia.....	21
2.4 Aspectos de qualidade de serviço em ambientes interativos	29
2.5 A infra-estrutura de integração dos sistemas digitais: uma breve descrição	33
2.5.1 Redes ATM	34
2.5.2 O padrão Fast Ethernet	35
2.5.3 O padrão Gigabit Ethernet	36
2.5.4 Comunicação sem fio (<i>wireless</i>).....	37
2.5.5 Comunicação via satellite	38
2.5.6 Microondas	38

2.5.7	Telefonia celular e comunicação de dados	39
2.5.8	RDSI - Rede Digital de Serviços Integrados	41
2.5.9	Computação e tecnologias móveis	43
2.5.10	Teleconferência e videoconferência.....	47
2.6	Os padrões MPEG-4 e MPEG-7.....	51
2.6.1	O padrão MPEG-4.....	53
2.6.1.1	Sistemas (Systems).....	55
2.6.1.2	Visual.....	56
2.6.1.3	Áudio.....	58
2.6.1.4	Framework para integração da entrega multimídia	59
2.6.1.5	Software de referência e conformidade	60
2.6.1.6	Players	60
2.6.1.7	Ferramentas de autoria	61
2.6.1.8	O formato de arquivo MPEG-4	61
2.6.2	O padrão MPEG-7.....	62
2.6.2.1	Sistemas (Systems).....	64
2.6.2.2	Linguagem de Definição de Descrição (DDL)	65
2.6.2.3	Visual.....	65
2.6.2.4	Áudio.....	65
2.6.2.5	Esquemas de Descrição de Multimídia (MDS)	66
2.6.2.6	Software de referência.....	66
2.6.2.7	Teste de conformidade (Conformance Test)	67
2.7	Apresentando o MPEG-21.....	67
2.7.1	MPEG-21 como um <i>framework</i> multimídia	68
2.7.2	O escopo do MPEG-21	69
2.7.3	O modelo do usuário.....	69
2.7.4	MPEG-21 e os itens digitais	70
2.7.5	Declaração de itens digitais	71
2.7.6	Identificação de itens digitais.....	71
2.7.7	Adaptação de itens digitais	72
2.8	Considerações finais	74
3	Modelagem de Sistemas Multimídia.....	75
3.1	Considerações iniciais	75
3.2	Métodos, técnicas, metodologias e modelos: uma breve apresentação	76
3.2.1	Relationship Management Methodology (RMM)	77
3.2.2	Object Oriented Hypermedia Design Model (OOHDM)	78
3.2.3	Enhanced Object Relationship Model (FORM)	79

3.2.4	Hypermedia Modelling Technique (HMT).....	80
3.2.5	A Database-Oriented Approach to the Design of Educational Hyperbook	80
3.2.6	Analysis and Design of Web-based Information Systems	81
3.2.7	OOMMA	82
3.3	A linguagem UML.....	83
3.3.1	Os diagramas UML e a sua adequação a esta proposta	85
3.3.2	O diagrama use-case.....	88
3.3.3	O diagrama de classes.....	88
3.3.4	O diagrama de colaboração.....	89
3.3.5	O diagrama de seqüência.....	89
3.3.6	O diagrama de estado	90
3.4	MPEG-4 e MPEG-7: justificativas para o uso desses padrões.....	91
3.4.1	A solução MPEG-4.....	92
3.4.2	Diferenças entre um codificador de mídias e uma arquitetura.....	93
3.4.3	Orientação a objetos.....	94
3.4.4	Alguns detalhes da arquitetura MPEG-4.....	96
3.4.5	Algumas vantagens e desvantagens da arquitetura MPEG-4.....	97
3.5	Considerações finais.....	98
4	Consciência de Contexto	99
4.1	Considerações iniciais	99
4.2	Consciência de contexto e suas relações com interação do usuário.....	99
4.3	Ubiquidade	103
4.4	Mobilidade como infra-estrutura para consciência de contexto	106
4.5	Uma proposta de formalismo para consciência de contexto	111
4.5.1	O modelo SPICE.....	112
4.5.1.1	Domínio.....	113
4.5.1.2	Ambiente.....	113
4.5.1.3	Usuário.....	115
4.5.1.4	Aplicação.....	116
4.5.1.5	Contexto	117
4.5.1.6	Recurso.....	119
4.5.1.7	Informação.....	120
4.5.2	A proposta das regras contextuais como extensão do modelo de Anind	120
4.6	Usando agentes para prover consciência de contexto	127
4.7	Considerações finais.....	130

5 Ambientes Interativos Conscientes de Contexto.....	132
5.1 Considerações iniciais	132
5.2 Princípios de modelagem	133
5.3 Representação de mídias	133
5.4 MPEG-1: uma tecnologia para vídeo interativo	134
5.5 MPEG-4 <i>Systems</i> : relacionando classes de objetos.....	136
5.5.1 API Scene	137
5.5.2 API Resource	138
5.5.3 API Net.....	139
5.6 A arquitetura MPEG-J.....	140
5.6.1 Uma visão geral dos recursos MPEG-J relevantes para esta proposta.....	141
5.6.2 Exemplos de aplicação do código MPEG-J	148
5.7 NECTAR: o modelo da infra-estrutura de comunicação.....	151
5.7.1 Detalhando a infra-estrutura do servidor.....	154
5.7.2 Detalhando a infra-estrutura do cliente.....	159
5.7.3 Detalhando a infra-estrutura da rede de comunicação	160
5.8 PRACTIC: o modelo da personalização.....	161
5.8.1 O programa interativo.....	162
5.8.2 O perfil de programa interativo.....	164
5.8.3 Utilização dos perfis em programas interativos	167
5.8.4 Produzindo um programa interativo.....	168
5.9 Integrando os modelos NECTAR, SPICE e PRACTIC.....	169
5.9.1 Modelagem das classes UML.....	172
5.9.1.1 A classe Person	173
5.9.1.2 A classe PDR.....	174
5.9.1.3 A classe Element	175
5.9.1.4 A classe Device.....	176
5.9.1.5 A classe Preference	176
5.9.1.6 A classe Network	176
5.9.1.7 A classe Profile	177
5.9.1.8 A classe UserPreference	178
5.9.1.9 A classe Layout	179
5.9.1.10 A classe Menu	180
5.9.1.11 A classe Control	180
5.9.1.12 A classe Communication	181
5.9.1.13 A classe Context.....	181
5.9.1.14 A classe Interaction.....	182
5.9.1.15 A classe Identification	183

5.9.1.16	A classe Location.....	184
5.9.1.17	A classe Activity.....	184
5.9.1.18	A classe Time.....	185
5.9.1.19	A classe Listener.....	186
5.9.1.20	A classe Monitor.....	187
5.9.1.21	A classe Program.....	187
5.9.1.22	A classe Sensor.....	188
5.9.1.23	A classe Schedule.....	189
5.9.1.24	A classe Element.....	189
5.9.1.25	A classe InformationElement.....	190
5.9.1.26	A classe Media.....	190
5.9.1.27	A classe Event.....	191
5.9.1.28	A classe Timeline.....	192
5.10	Integrando agentes de software para prover consciência de contexto.....	192
5.11	Discussão das relações entre esta proposta e o MPEG-21.....	198
5.12	Considerações finais.....	200
6	Estudo de Caso: Modelagem de Programas para Televisão Interativa.....	201
6.1	Considerações iniciais.....	201
6.2	Um breve relato da evolução da interação na televisão.....	201
6.3	Revisitando aspectos de consciência de contexto aplicáveis à TV interativa.....	203
6.4	Infra-estrutura para distribuição de programas de TV interativa.....	205
6.5	Requisitos para a produção de programas de TV interativa.....	206
6.6	Modelagem da aplicação de TV interativa: estrutura e implementação.....	207
6.7	Exemplos de programas de TV interativa.....	221
6.8	Aplicação dos conceitos de programa e perfil a um programa de evento esportivo.....	223
6.8.1	Definindo o programa.....	224
6.8.2	Descrevendo o programa.....	225
6.8.3	Descrevendo os perfis.....	227
6.9	Considerações finais.....	230
7	Conclusões.....	232
7.1	Considerações iniciais.....	232
7.2	Contribuições.....	234
7.2.1	Contribuições para o campo da modelagem de sistemas multimídia.....	235
7.2.2	Contribuições para o campo dos agentes de software.....	235
7.2.3	Contribuições para o campo da consciência de contexto.....	235

7.2.4 Contribuições para o desenvolvimento de aplicações multimídia interativas.....	236
7.2.5 Contribuições para o estabelecimento de novos modelos de negócios.....	236
7.3 Limitações.....	237
7.4 Trabalhos futuros.....	237
7.5 Considerações finais.....	239
Referências Bibliográficas.....	240
Apêndice A - Resumo do Relatório das Atividades Realizadas no Exterior.....	253
Apêndice B - Resumo e Abstracts dos Artigos Publicados.....	265

Índice de Figuras

Figura 1.01 - O escopo da modelagem - estrutura gráfica (Santos Jr. et al., 2001b).....	8
Figura 2.01 - Uma apresentação multimídia protocolada em SMIL.....	14
Figura 2.02 - Um "mundo virtual" construído em VRML.....	15
Figura 2.03 - Documento formatado de acordo com o padrão HTML.....	17
Figura 2.04 - Um exemplo de página HTML com recursos DHTML.....	18
Figura 2.05 - Um exemplo de documento estruturado XML para o domínio ensino.....	19
Figura 2.06 - Trecho de documento XSL para formatar um elemento.....	21
Figura 2.07 - Camadas de um sistema multimídia.....	22
Figura 2.08 - Estrutura genérica da arquitetura cliente-servidor para a Internet.....	26
Figura 2.09 - Níveis de QoS em um ambiente interativo.....	31
Figura 2.10 - Exemplo da estrutura de um <i>bitstream</i> MPEG-4 (Pereira & Ebrahimi, 2002).....	57
Figura 2.11 - Entrega de dados no MPEG-4 (ISO, 1997b).....	59
Figura 2.12 - Tela do IMI-2D.....	60
Figura 2.13 - Tela do IST MPEG-4 Video Compliant Framework.....	61
Figura 2.14 - Exemplo do formato de arquivo MPEG-4 para intercâmbio de dados (ISO, 1999a).....	62
Figura 2.15 - Arquitetura MPEG-7 (ISO, 2001d).....	64
Figura 2.16 - Estrutura do MDS (ISO, 2001g).....	66
Figura 2.17 - Exemplo de uma descrição de um vídeo em MPEG-7.....	67
Figura 2.18 - Modelo de eventos e interfaces que descrevem interações MPEG-21 (MPEG-21, 2001)....	70
Figura 2.19 - Relacionamento entre declaração, identificação e descrição de um item digital.....	72
Figura 2.20 - Esquema para o conceito de adaptação de item digital (MPEG-21).....	73
Figura 3.01 - Modelos do Database-Oriented Approach to the Design of Educational Hyperbook.....	80
Figura 3.02 - Estrutura OOMMA representada em um diagrama de classes (Engels, 2000).....	83
Figura 3.03 - Camadas de objetos em um ambiente com consciência de contexto.....	87
Figura 4.01 - Focos de interação do usuário em relação à existência ou não de consciência.....	103
Figura 4.02 - Modelo para uso de consciência associada à mobilidade.....	109
Figura 4.03 - O escopo da modelagem - estrutura gráfica (Santos Jr. et al., 2001b).....	112
Figura 4.04 - Diagrama de classes UML para formalização do elemento DOMÍNIO.....	113
Figura 4.05 - Diagrama de classes UML para formalização do elemento AMBIENTE.....	114

Figura 4.06 - Diagrama de classes UML para formalização do elemento APLICAÇÃO	117
Figura 4.07 - Diagrama de classes UML para formalização do elemento CONTEXTO	118
Figura 4.08 - Diagrama de classes UML para formalização do elemento RECURSO	119
Figura 4.09 - Diagrama de classes UML para formalização do elemento INFORMAÇÃO	120
Figura 4.10 - Modelo para formalização das regras contextuais	121
Figura 4.11 - Diagrama completo de classes UML do modelo SPICE	122
Figura 4.12 - Esquema de relações de pertinência a um contexto	124
Figura 4.13 - Esquema de relações e transições em um contexto	124
Figura 4.14 - Exemplo de formalismo e uso de uma regra contextual	125
Figura 4.15 - Operadores de regras contextuais	126
Figura 4.16 - Diagrama de classes UML para os operadores de regras contextuais	126
Figura 4.17 - Operadores de regras contextuais	127
Figura 4.18 - Modelo para relações entre contexto e elementos contextuais	127
Figura 5.01 - Arquitetura mínima para o uso do MPEG-4 em ambientes interativos	135
Figura 5.02 - Identificação das interfaces MPEG-J em uma arquitetura MPEG-4 (ISO, 1999a)	140
Figura 5.03 - Exemplo MPEGlet para manipular os nós de um grafo de uma cena (ISO, 1999a)	149
Figura 5.04 - Exemplo MPEGlet para adicionar eventos a uma cena MPEG-4 (ISO, 1999a)	150
Figura 5.05 - Exemplo MPEGlet para manipular os dados de um BIFS de uma cena (ISO, 1999a)	151
Figura 5.06 - Esquema para decodificação de <i>streams</i> MPEG-4 (ISO, 2001a)	154
Figura 5.07 - Um exemplo de DS (Description Schema) MPEG-7	157
Figura 5.08 - Esquema para apresentação de objetos MPEG-4 em um terminal (ISO, 2001a)	159
Figura 5.09 - Estrutura proposta para um programa interativo	162
Figura 5.10 - Entidades que compõem um programa interativo em uma aplicação	162
Figura 5.11 - Schema MPEG-7 para descrição de uma cena	163
Figura 5.12 - Entidades que compõem um programa interativo em uma aplicação	164
Figura 5.13 - Schema de descrição do elemento Layout	165
Figura 5.14 - Schema de descrição do elemento Communication	165
Figura 5.15 - Schema de descrição do elemento Context	166
Figura 5.16 - Schema de descrição do elemento Interaction	166
Figura 5.17 - Schema de descrição de atributos do elemento Interaction	166
Figura 5.18 - Schema de descrição do elemento UserPreferences	167
Figura 5.19 - Diagrama de seqüência UML para as entidades de um programa interativo	168
Figura 5.20 - Representação genérica da relação entre programa, perfil e cena	168
Figura 5.21 - Diagrama UML <i>use-case</i> para o modelo de acesso aos objetos de mídia	170
Figura 5.22 - Diagrama de classes UML para o modelo de acesso ao conteúdo de mídia	170
Figura 5.23 - Diagrama UML de estados para o acesso ao conteúdo associado à mídia	171
Figura 5.24 - Diagrama UML de colaboração entre agentes físicos no acesso ao conteúdo de mídia	171

Figura 5.25 - Diagrama de classes UML de um ambiente interativo consciente de contexto	172
Figura 5.26 - Esquema das interações completas em ambiente interativo consciente de contexto	193
Figura 5.27 - Diagrama UML de colaboração entre agentes em um ambiente interativo.....	194
Figura 5.28 - Comportamento de um subconjunto de agentes.....	195
Figura 5.29 - Esquema baseado em agentes para coleta de informações contextuais	196
Figura 5.30 - Diagrama UML de colaboração para subconjunto de agentes (Santos Jr. et al, 2001a)....	197
Figura 6.01 - Mapeamento de entidades contextuais em elementos de perfis de TVI	204
Figura 6.02 - Ambiente consciente de contexto para a Televisão Interativa	206
Figura 6.03 - Exemplo de <i>Schema</i> para o elemento <i>Program</i>	208
Figura 6.04 - Exemplo de cabeçalho de um programa.....	208
Figura 6.05 - Exemplo de <i>Schema</i> para descrever a cena	209
Figura 6.06 - Exemplo de <i>Schema</i> para descrição das mídias.....	212
Figura 6.07 - Exemplo de <i>Schema</i> para os elementos da linha do tempo.....	213
Figura 6.08 - Exemplo de <i>Schema</i> para o elemento <i>profile</i>	214
Figura 6.09 - Exemplo de cabeçalho de um perfil de programa.....	214
Figura 6.10 - Exemplo do <i>Schema</i> para o elemento <i>layout</i>	216
Figura 6.11 - Exemplo de <i>Schema</i> para o elemento <i>Communication</i>	217
Figura 6.12 - Exemplo de <i>Schema</i> para os elementos de consciência de contexto	218
Figura 6.13 - Exemplo do uso dos elementos de consciência de contexto	218
Figura 6.14 - Exemplo de <i>Schema</i> para os elementos de interação	219
Figura 6.15 - Exemplo do uso dos elementos de interação	220
Figura 6.16 - Exemplo de <i>Schema</i> para os elementos de preferências do usuário	220
Figura 6.17 - Exemplo do uso dos elementos de preferências do usuário	221
Figura 6.18 - Exemplo de descrição de um programa de evento esportivo.....	227
Figura 6.19 - Exemplo do perfil básico.....	227
Figura 6.20 - Exemplo do perfil intermediário.....	228
Figura 6.21 - Exemplo do perfil avançado.....	230
Figura 7.01 - O escopo da modelagem - estrutura gráfica (Santos Jr. et al., 2001)	233

Índice de Tabelas

Tabela 2.01 - Algumas mídias de distribuição e velocidades de transferência	22
Tabela 2.02 - Relação de quantidade de armazenamento para algumas mídias.....	23
Tabela 2.03 - Algumas aplicações de serviços interativos (Hashisuca, 1996).....	32
Tabela 5.01 - Pacotes da API MPEG-J.....	141
Tabela 5.02 - Interfaces da API MPEG-J.....	142
Tabela 5.03 - Métodos da interface DecoderConfigDescriptor.....	142
Tabela 5.04 - Métodos da interface ESDDescriptor	142
Tabela 5.05 - Métodos da classe MpegJTerminal.....	143
Tabela 5.06 - Métodos da interface ObjectDescriptor	144
Tabela 5.07 - Classes e interfaces do pacote Resource	144
Tabela 5.08 - Interfaces e classes do pacote Network.....	146
Tabela 5.09 - Interfaces e classes do pacote Decoder.....	146
Tabela 5.10 - Interfaces do pacote Scene.....	147
Tabela 5.11 - Métodos da classe Person	174
Tabela 5.12 - Métodos da classe PDR.....	174
Tabela 5.13 - Métodos da classe Element.....	175
Tabela 5.14 - Métodos da classe Device	176
Tabela 5.15 - Métodos da classe Preference.....	176
Tabela 5.16 - Métodos da classe Network	177
Tabela 5.17 - Métodos da classe Profile.....	177
Tabela 5.18 - Métodos da classe UserPreferences.....	178
Tabela 5.19 - Métodos da classe Layout.....	179
Tabela 5.20 - Métodos da classe Menu.....	180
Tabela 5.21 - Métodos da classe Control.....	180
Tabela 5.22 - Métodos da classe Communication.....	181
Tabela 5.23 - Métodos da classe Context	182
Tabela 5.24 - Métodos da classe Interaction	182
Tabela 5.25 - Métodos da classe identification.....	183
Tabela 5.26 - Métodos da classe Location.....	184
Tabela 5.27 - Métodos da classe Activity	185
Tabela 5.28 - Métodos da classe Time	186

Tabela 5.29 - Métodos da classe Listener	187
Tabela 5.30 - Métodos da classe Monitor	187
Tabela 5.31 - Métodos da classe Program	188
Tabela 5.32 - Métodos da classe Sensor	188
Tabela 5.33 - Métodos da classe Schedule	189
Tabela 5.34 - Métodos da classe Element	189
Tabela 5.35 - Métodos da classe InformationElement	190
Tabela 5.36 - Métodos da classe Media	191
Tabela 5.37 - Métodos da classe Event	191
Tabela 5.38 - Métodos da classe Timeline	192
Tabela 5.39 - Pacotes da plataforma JADE	197
Tabela 6.01 - Elementos do Programa	207
Tabela 6.02 - Subelementos do programa	208
Tabela 6.03 - Subelementos de mídia	209
Tabela 6.04 - Subelementos de linha do tempo	212
Tabela 6.05 - Elementos de perfil	214
Tabela 6.06 - Subelementos de layout	215
Tabela 6.07 - Subelementos de comunicação	216
Tabela 6.08 - Subelementos de consciência de contexto	217
Tabela 6.09 - Subelementos de interação	219
Tabela 6.10 - Elementos de personalização do usuário	220
Tabela 6.11 - Componentes do programa	224

1.1 Considerações iniciais

Com a disseminação das tecnologias associadas à multimídia interativa, novas aplicações e formas de interação têm se tornado possíveis. Os diferentes meios com que o usuário pode interagir com uma apresentação multimídia (ou com objetos multimídia) demandam a adaptação das aplicações (eventualmente de forma automática) a determinadas situações, com o objetivo de melhorar a interação em seus diversos níveis. Além disso, a presença da computação de forma ubíqua oferece ao usuário a expectativa de que é possível acessar informações e serviços em todo lugar e a qualquer momento e, para atender a essas expectativas, aspectos de consciência de contexto têm sido usados (Abowd, 1999; Pascoe et al., 1999; Dey & Abowd, 1999; Dey & Abowd, 2000; Schmidt, 2000).

Um vasto número de situações em que o usuário pode estar envolvido, durante uma sessão interativa, exige esforços no sentido de definir como essas interações ocorrem e quais são os melhores critérios para tratamento dessas interações, oferecendo, assim, melhores condições de uso para o usuário final (Dey & Abowd, 2000; Schmidt, 2000). Uma maneira para melhorar o suporte à interação do usuário é melhorar a comunicação entre as entidades do ambiente durante todo o tempo em que essas interações podem ocorrer. Isto pode ser possível através do uso de informações contextuais obtidas da aplicação, da infra-estrutura de comunicação, das características das mídias utilizadas pela aplicação e das características dos dispositivos e terminais utilizados pelo usuário para ter acesso à informação.

Neste sentido, torna-se interessante formalizar um cenário para aplicações interativas, considerando o estado da arte representado pela evolução e convergência dos sistemas digitais (computadores, televisão, telefones e telefonia, infra-estrutura de comunicação de dados, PDAs – *Personal Digital Assistant*, dentre outros), e propor a utilização dos padrões da família MPEG (*Moving Picture Experts Group*) (ISO, 1997a) – MPEG-4, para representação de conteúdo multimídia na forma de objetos de mídia (ISO, 1997b; ISO, 1999a; ISO, 2001a); e MPEG-7, para descrição de informação associada ao conteúdo multimídia (ISO, 1997c; ISO,

2000a) – como base tecnológica para implementação de ambientes interativos conscientes de contexto, considerando que esses padrões MPEG apresentam potencialidades para:

- oferecer formas para estruturar conteúdo multimídia e informações associadas à mídia, permitindo a implementação dos aspectos de consciência de contexto discutidos neste trabalho;
- possibilitar o uso os aspectos de consciência de contexto para prover adaptação automática às necessidades do usuário, da aplicação e da infra-estrutura;
- permitir a extensão dos conceitos de consciência do contexto a todos os níveis de entidades que compõem a infra-estrutura do ambiente interativo.

Alguns estudos, encontrados na literatura, apresentam esses padrões da família MPEG – MPEG-4 para codificação e *streaming* de mídia, e MPEG-7 para descrição de conteúdo e indexação – como tecnologias capazes de suportar a construção da infra-estrutura necessária às novas e complexas aplicações multimídia interativas e distribuídas (Chiariglione, 1999; Battista et al., 1999; Chiariglione, 2001).

Apesar da complexidade e abrangência desses padrões, existe uma carência de modelos e modelagens para esse novo universo de aplicações interativas. Adicionalmente, os trabalhos que estão sendo desenvolvidos na área raramente consideram, como parte formal da modelagem ou modelo, aspectos como **contexto** e **consciência de contexto** e adaptabilidade às necessidades do usuário como forma para melhorar a interatividade (Schmidt, 2000).

Esta tese apresenta uma proposta para modelagem, utilizando a sintaxe gráfica da linguagem UML. (*Unified Modelling Language*) (Booch et al., 1996), de um ambiente interativo consciente de contexto baseado nos padrões MPEG-4 e MPEG-7, formalizando as situações de adaptação da apresentação de cenas multimídia de acordo com os parâmetros de consciência de contexto formalizados (Santos Jr. et al., 2001b). Essa modelagem apresenta-se como uma tentativa de cobrir as interações *usuário-rede-aplicação* (*complete interactions* (Santos Jr. et al., 2001a)) que ocorrem em um ambiente interativo (Rodrigues, 1998). Neste cenário, o trabalho enfoca principalmente quais os tipos de informações contextuais podem ser utilizados e como devem ser manipulados em todos os níveis de interação presentes no ambiente. Dessa investigação, o trabalho também propõe a formalização do conceito de *perfil de aplicação interativa* e discute como uma aplicação de Televisão Interativa (TVI) poderia usufruir da modelagem proposta.

1.2 Contextualização

O uso de objetos multimídia - áudio, vídeo, texto, imagens, gráficos, animações - na construção de ambientes interativos é extremamente complexo (Battista et al., 1999). Uma vez que é possível representar mídias como objetos (objetos na concepção plena presente no Paradigma da Orientação a Objetos da OMG - *Object Management Group*), esta proposta oferece formas para que o usuário possa interagir livremente com cada objeto que compõe uma ou mais cenas em uma sessão de multimídia em um ambiente interativo.

Como resultado dessa concepção, a qualidade da multimídia apresentada ao usuário passa a ser controlada não somente pela aplicação e pelos recursos computacionais disponíveis (computadores, *players*, codificadores e decodificadores, dentre outros), mas também pelas características de todas as entidades que compõem o ambiente interativo em questão, incluindo-se a infra-estrutura de comunicação, os dispositivos sensoriais e o(s) perfil(is) do usuário. Desta forma, é possível personalizar, em diversos níveis definidos pelo projetista, a distribuição e o consumo de multimídia interativa de forma personalizada às características de um determinado ambiente. Como prática desejável para o projeto desses novos ambientes, a modelagem torna-se um elemento essencial, bem como a observação e o uso de padrões que permitam a interoperabilidade, *openness* e reusabilidade, dentre outras várias características desejáveis.

1.3 Trabalhos relacionados

A crescente evolução dos sistemas digitais integrados e o forte apelo que sempre existiu pela distribuição de áudio, vídeo e informações associadas, com boa qualidade, fizeram com que a investigação dos novos sistemas multimídia ganhasse um caráter absolutamente multidisciplinar, envolvendo, no campo das Ciências de Computação, esforços de subáreas como a Multimídia/Hipermídia (com ênfase na codificação e distribuição de mídias e informação associada, e também em esquemas de descrição e intercâmbio de dados multimídia), Engenharia de Software (com ênfase em técnicas de modelagem), Inteligência Computacional (com ênfase no campo dos agentes de software), Interação Usuário-Computador (com ênfase em consciência de contexto) e Redes de Computadores (com ênfase na infra-estrutura de comunicação), dentre outras cujo potencial de integração tem sido investigado. Neste sentido, ao apresentar alguns trabalhos relacionados a esta tese, é necessário categorizá-los na tentativa de identificar, com clareza, as fronteiras a partir das quais esses trabalhos impactam e/ou são impactados pelas contribuições aqui reportadas.

No campo da **modelagem de sistemas multimídia**, Sauer e Engels (Sauer & Engels, 1999; Engels, 2000) propuseram um modelo UML para sincronização de componentes multimídia, porém sem considerar aspectos de consciência de contexto e sem explorar a codificação das mídias como objetos. Van Setten e seus colaboradores propuseram uma modelagem de ambientes multimídia interativos a partir da definição de categorias de interatividade e níveis de personalização (Van Setten, 2001; Van Setten et al., 2001). No entanto, o trabalho ainda não contempla as relações entre a personalização e a representação de conteúdo multimídia. Consciência de contexto tem sido investigada como forma de prover personalização, exclusivamente.

No campo dos **agentes de software**, Bellefemine e seus colaboradores (Bellefemine et al., 2001) propuseram e implementaram a plataforma JADE (*Java Agents Development Framework*), que apresenta os principais recursos para desenvolvimento de aplicações baseadas em agentes e suporta os requisitos de mobilidade. No entanto, os aspectos de consciência de contexto não são ainda utilizados pelo *framework*. Além disso, mecanismos de integração dos agentes JAVA com a API MPEG-J (ISO, 1999d) não foram especificados, o que possibilita extensões na plataforma JADE a partir da modelagem proposta neste trabalho.

No campo da **consciência de contexto**, Dey e Abowd (Dey & Abowd, 1999) propuseram a representação de contexto a partir das entidades WHO, WHEN, WHAT, WHERE e HOW. A pesquisa de Dey e Abowd concentra-se, principalmente, no suporte ao desenvolvimento de aplicações que necessitem utilizar informações contextuais, sem, no entanto, mostrar esforços com a formalização dos aspectos de consciência de contexto e com a investigação dos diversos níveis e critérios de interação que podem surgir a partir do uso de consciência de contexto.

Por último, no campo das **aplicações**, mais precisamente da **televisão interativa**, iniciativas importantes foram feitas em busca de padronização e modelos para distribuição, apresentação e consumo de multimídia interativa. Dentre essas iniciativas, os trabalhos do ATVEF (*Advanced Television Enhancement Forum*) (ATVEF, 1995), do TV-Anytime Forum e do W3C Consortium merecem ser relacionados a esta tese. O W3C Consortium, observando o desejo de integração da TV com a Web, criou um novo URI (*Uniform Resource Indicator*) (Bernes-Lee et al., 1998), denominado 'tv' (Finseth, 1998; Vickress, 2000). O ATVEF especifica melhorias nas funcionalidades atuais da televisão convencional e propõe funcionalidades

novas, principalmente relacionadas à interação. O centro dos esforços do ATVEF é a união da TV atual com a Internet, especificando para cada padrão de TV (PAL, NTSC, DVB, dentre outros) como transmitir através do protocolo UHITP (*Unidirecional Hypertext Transfer Protocol*), protocolo esse definido pelo ATVEF (ATVEF, 2000). O TV-Anytime Forum busca especificar uma nova TV, com novas funções (incluindo Internet) e novas formas de interação. Esse padrão é mais completo e mais complexo que o padrão que está sendo definido pelo ATVEF, sendo totalmente digital e buscando estruturar os programas oferecidos, fornecendo formas padronizadas para acesso a conteúdo (TVAF, 2000a; TVAF, 2000b).

Apesar da complexidade e abrangência desses padrões, nenhum deles contempla modelos e modelagens para aplicações interativas. Adicionalmente, os trabalhos que estão sendo desenvolvidos na área de televisão interativa raramente levam em conta aspectos como **contexto** e **consciência de contexto**, e adaptabilidade às necessidades do usuário como forma de melhorar a interatividade.

1.4 Metodologia

O contexto no qual este projeto se insere é atual, tanto em termos da pesquisa em Ciência da Computação quanto em termos de aplicabilidade na cadeia produtiva dos complexos sistemas multimídia interativos que devem emergir. Neste ponto, é interessante e bastante razoável adotar uma estratégia de investigação baseada em uma contínua prospecção tecnológica e acompanhamento das atividades de desenvolvimento e padronização das tecnologias utilizadas neste trabalho.

Ao propor a **modelagem de ambientes interativos conscientes de contexto**, esta tese considera as seguintes definições:

- **ambiente**: pode ser visto como uma macro-entidade que contém uma aplicação multimídia, toda a infra-estrutura de comunicação necessária, elementos de hardware e software, dispositivos sensoriais e suporte à interação do usuário, além do próprio usuário;
- **interativo**: refere-se a toda e qualquer interação entre os elementos presentes em um ambiente;

- **consciente de contexto:** identifica o uso de técnicas para provimento de informações contextuais que permitam adaptabilidade do ambiente às necessidades do usuário e do próprio ambiente como um todo.

Adicionalmente, termos complementares são considerados na composição da proposta:

- **sistema:** pode ser visto como um conjunto de elementos do ambiente, sendo, portanto, arranjos de um ambiente;
- **aplicação:** uma solução de software que pode ser utilizada como um elemento componente de um ambiente ou sistema;
- **multimídia:** o conjunto de mídias (nesta proposta, *objetos de mídia*) utilizáveis no desenvolvimento de uma aplicação de software (aplicação multimídia).

Ainda no campo das definições, o conceito de **modelagem** é visto nesta proposta como sendo a especificação e integração de modelos para prover uma referência para prototipação e desenvolvimento, neste caso, de **ambientes interativos conscientes de contexto**.

A partir da compreensão dessas macro-definições, estudos exploratórios foram realizados nos primeiros 24 (vinte e quatro) meses do programa de doutoramento, com vistas a identificar o escopo da proposta e também as bases tecnológicas que poderiam suportar o desenvolvimento deste trabalho.

A revisão de literatura, conforme pode ser visto nos trabalhos relacionados na **Seção 1.3**, evidenciou que existem modelos, técnicas e metodologias para o desenvolvimento de aplicações multimídia e hpermídia (especialmente aquelas orientadas por hiperdocumentos), porém sem contemplar a definição de **ambiente** defendida neste trabalho. Adicionalmente, o conceito do que é **interativo** estrapola o universo da interação usuário-computador, uma vez que esta proposta considera a interação que ocorre em todos os níveis do ambiente (interação usuário-aplicação; interação aplicação-rede; interação usuário-ambiente, do ponto-de-vista de todos os dispositivos e agentes que fazem parte do ambiente).

A partir da definição de que a modelagem deveria cobrir as interações em todo o ambiente, a o foco da investigação centrou-se na consciência de contexto, considerando que o ambiente, para reagir de forma automática às interações, deveria ser consciente de todos os seus elementos e de todas as interações geradas por esses elementos. Da investigação de

literatura, várias propostas de uso de consciência de contexto foram consideradas. No entanto, o modelo de **Anind** foi utilizado como referência, uma vez que define as entidades contextuais e apresenta mecanismos para uso de consciência de contexto no desenvolvimento de aplicações, enquanto outras propostas têm seu foco apenas na aplicação dos conceitos de consciência de contexto (Dey & Abowd, 2000).

Analisado o modelo de **Anind**, pôde-se verificar que havia necessidade de especificar extensões nesse modelo para contemplar o uso de consciência de contexto em ambientes interativos, considerando a definição de **ambiente** proposta nesta tese.

Quanto à investigação tecnológica, os padrões MPEG-4 e MPEG-7 foram exaustivamente estudados e analisados (ISO, 1999a; ISO, 2000a), sendo que todo o processo de padronização foi observado quando do uso destas tecnologias nesta tese.

Desta metodologia de investigação científica é que surgiram as duas principais contribuições deste trabalho: a) um modelo de interações baseado em consciência de contexto e adequado à adaptabilidade de todas as interfaces do usuário como o ambiente; b) a especificação de modelos de interação e a integração desses modelos para compor a modelagem de ambientes interativos conscientes de contexto. Essas contribuições são apresentadas nas próximas seções e discutidas ao longo da tese. Adicionalmente, os padrões MPEG-4 e MPEG-7 forneceram as interfaces suficientes para suportar o desenvolvimento de aplicações multimídia e a construção de ambientes interativos com os requisitos de interação especificados pela modelagem aqui reportada.

1.5 Objetivos

A computação ubíqua proporciona ao usuário a expectativa de que é possível acessar informações e serviços de qualquer ponto, geograficamente definido, e a qualquer momento. Os requisitos de mobilidade fazem com que o contexto do usuário seja alterado de acordo com a sua localização, identidade e demais entidades que estejam envolvidas num determinado ambiente. Neste sentido, os avanços nas tecnologias associadas à multimídia interativa trazem consigo novas expectativas em relação às tradicionais formas de interação multimídia.

Conforme citado, as informações contextuais podem auxiliar na definição e execução dos ajustes necessários para uma apresentação multimídia com um determinado grau de qualidade de serviço, provendo adaptabilidade e personalização, porém conservando os níveis de interatividade esperados pelo usuário (McCanne et al., 1996; Dey & Abowd, 2000).

Este trabalho, ao ter propôr uma modelagem para ambientes interativos conscientes de contexto, especifica métodos para integração de tecnologias como MPEG-4 (ISO, 1999a; ISO, 2001a), MPEG-7 (ISO, 2000a), XML (Connolly, 1997), dentre outras, e aponta para modelos de desenvolvimento de aplicações interativas, tais como TVI, videoconferência, vídeo-sob-demanda, dentre outras. A **Figura 1.01** apresenta a estrutura gráfica do escopo coberto pela modelagem proposta. A modelagem, então, é composta por três modelos distintos, a saber:

- **NECTAR** (*Network Environment with Context-Awareness Rules*): cobre a infra-estrutura de comunicação necessária a aplicações interativas conscientes de contexto, principalmente em termos dos requisitos de rede;
- **SPICE** (*Schemes for Interaction on Context-Aware Environments*): cobre os aspectos de modelagem que definem esquemas e critérios de interação para o modelo de interações completas (*complete interaction* → *user-network-application*);
- **PRACTIC** (*Profile for Context-Aware Interactive Applications*): cobre o modelo da geração de perfis de aplicações interativas, sendo composto pelo **PRACTIC-E** (*PRACTIC Program Examples*) - que cobre a especificação de perfis para instâncias de aplicações interativas, como é o caso de programas de TVI - e pelo **PRACTIC-S** (*PRACTIC Schemata*) - que cobre a especificação de esquemas de descrição para aplicações interativas genéricas (Faria, Santos Jr., Goularte & Moreira, 2001);

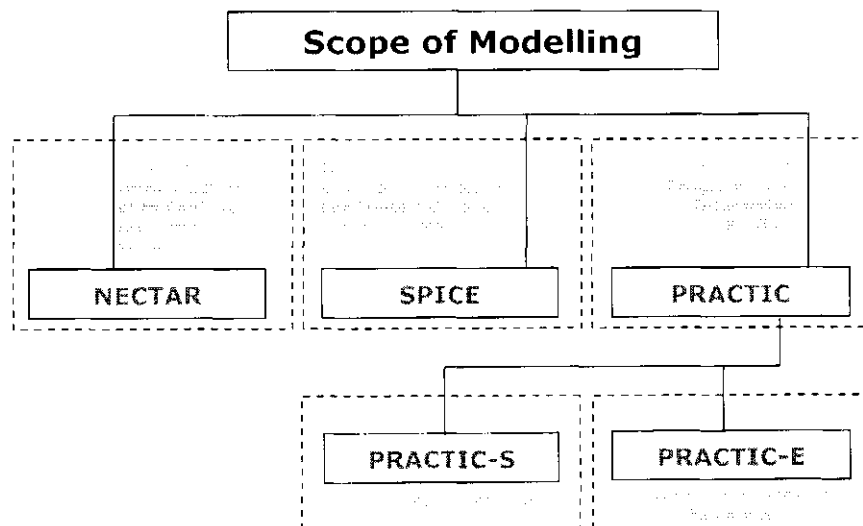


Figura 1.01: O escopo da modelagem - estrutura gráfica (Santos Jr. et al., 2001b)

Os modelos propostos consideram que o uso de aspectos de consciência de contexto pode ser dividido em três categorias distintas:

- **controle dinâmico de objetos:** a tecnologia MPEG-4 permite a composição de cenas multimídia a partir de objetos de mídia independentes; esses objetos apresentam, então, potencial para serem manipulados separadamente. Para que isto se torne possível, as descrições de objetos (forma, conteúdo, ações e relacionamentos) devem incorporar semânticas de consciência de contexto que permitam à aplicação estar sempre consciente do estado corrente de cada um desses objetos;
- **adaptação dinâmica da aplicação às condições da infra-estrutura:** as interfaces interativas devem prover mecanismos para especificação da qualidade dos componentes de uma apresentação multimídia; uma vez que o usuário especifica a qualidade, esses parâmetros devem permanecer aceitáveis durante toda a apresentação;
- **adaptação dinâmica da aplicação ao contexto do usuário:** existem situações nas quais a interface deve se adaptar automaticamente ao contexto do usuário; dependendo das informações descritas pelas semânticas contextuais e parâmetros de consciência de contexto, tais como identidade, localização, ações e tempo, serviços e funções podem ou não ser habilitados ao usuário.

É importante observar, em termos de objetivos, que a segunda categoria auxilia o trabalho de produtores de conteúdo multimídia e gerenciadores de rede. As duas outras categorias podem trazer benefícios diretamente aos usuários finais.

1.6 Inovações

As inovações presentes neste trabalho estão concentradas no fato de que os padrões, tecnologias e modelos, propostos até então, não consideram as interações existentes entre a infra-estrutura de comunicação, a aplicação e os objetos de mídia que compõem uma ou mais cenas multimídia, uma vez que a representação das mídias como objetos foi possível somente após a padronização do MPEG-4. Desta forma, os aspectos de consciência de contexto são utilizados sempre ao nível da aplicação e não há preocupação em formalizar os critérios de interação para uso da consciência de contexto e, nem mesmo, de formalizar a semântica dos elementos contextuais WHO, WHAT, WHERE e WHEN (Dey & Abowd, 2000). Ao formalizar os aspectos de consciência de contexto, dentro do universo das aplicações interativas, este trabalho representa um avanço na direção dos novos tipos de interação que são esperados para a nova geração de aplicações multimídia. Algumas funções

tradicionais de controle remoto, por exemplo, tendem a desaparecer e outros tipos de mecanismos de controle devem surgir muito em breve, o que reforça a importância da modelagem dos ambientes interativos conscientes de contexto, e não somente das aplicações multimídia.

O uso de aspectos de consciência de contexto tem sido largamente pesquisado no domínio das aplicações e das redes de comunicação de dados, porém é uma inovação na modelagem de ambientes multimídia interativos.

Este trabalho também inova na direção da definição dos conceitos de perfil de aplicação interativa e de programa interativo, que são apresentados de forma compatível com as tecnologias MPEG-4 e MPEG-7, gerando extensões a essas tecnologias de modo a possibilitar o uso integrado da aplicação com o ambiente.

1.7 Principal contribuição original

A modelagem proposta nesta tese visa definir e formalizar novos tipos de interação gerados por uma aplicação inserida em um ambiente interativo. Nesses ambientes, mecanismos tradicionais como VCR (*Video Cassette Recorder*) e STB (*Set-Top Box*) tendem a ser substituídos por dispositivos sensoriais, o que sinaliza para a interação usuário-ambiente ao invés da interação usuário-computador. A modelagem auxilia o desenvolvimento dessas aplicações interativas por apresentar um conjunto bem definido das interações geradas pela comunicação entre as entidades usuário, rede, aplicação e terminal (Santos Jr. et al., 2001a; Santos Jr. et al., 2001b). A modelagem, ao contemplar a consciência de contexto, apresenta avanços para o desenvolvimento de aplicações complexas, como é o caso dos ambientes classificados como *wearable computing* e *ubiquitous computing*.

Os modelos NECTAR, SPICE e PRACTIC, em sua concepção, consideram as potencialidades dos padrões MPEG-4 e MPEG-7. Particularmente, o MPEG-7 é utilizado para oferecer suporte à personalização do ambiente através de esquemas de descrição das cenas multimídia. Como passo em direção a um futuro ainda mais promissor, a modelagem contempla o uso do mais novo padrão da família MPEG - o MPEG-21 - como tecnologia para a construção de novos esquemas de representação de objetos de mídia como itens digitais universais. O novo padrão provavelmente acrescentará importantes elementos à modelagem proposta.

1.8 Estrutura do trabalho

Esta tese está estruturada em sete capítulos. No **Capítulo 2**, é apresentada uma visão geral do campo da multimídia interativa e distribuída, com destaque para uma descrição detalhada das características e funcionalidades dos padrões MPEG-4, MPEG-7 e MPEG-21. No **Capítulo 3**, é apresentado o estado da arte em técnicas, modelos, linguagens, metodologias e *frameworks* que podem ser utilizados para a modelagem de aplicações multimídia; uma atenção especial é dada às características da linguagem UML, escolhida como ferramenta para a construção da modelagem proposta nesta tese. A formalização dos aspectos de consciência de contexto aplicáveis a ambientes interativos é apresentada e discutida no **Capítulo 4**. No **Capítulo 5**, os modelos NECTAR, SPICE e PRACTIC, que compõem a modelagem proposta, são apresentados em detalhes. No **Capítulo 6**, é discutida a construção de uma aplicação de Televisão Interativa, servindo como estudo de caso para avaliação das principais limitações da modelagem proposta. No **Capítulo 7**, são discutidas as principais contribuições deste trabalho e apresentado um cenário para futuras pesquisas.

No **Apêndice A**, é apresentado um resumo do **Relatório das Atividades Realizadas no Exterior**, mais precisamente nos laboratórios da Telecom Italia Lab - TILAB (que era conhecido como CSELT - Centro di Studi e Laboratorio Telecomunicazioni), junto à Advanced Multimedia Division, em Turim, Itália, no período de maio a dezembro de 2001. Esse resumo do relatório foi incorporado à tese por descrever as atividades realizadas durante a estadia no exterior, atividades essas que permitiram aprofundar os estudos sobre os padrões MPEG-4 e MPEG-7, especialmente em termos do MPEG-4 Systems, API MPEG-J e integração de *streams* MPEG-7 em *streams* MPEG-4. O relatório completo encontra-se registrado e depositado nos arquivos do CSELT, mediante aprovação dos supervisores da pesquisa no exterior.

No **Apêndice B**, são apresentados os resumos e/ou abstracts dos artigos publicados como resultado deste trabalho (Santos Jr. et al., 2001a; Santos Jr. et al., 2001b; Faria, Santos Jr., Moreira & Goularte, 2001; Santos Jr. et al., 2002a; Santos Jr. et al., 2002b).

2.1 Considerações iniciais

Em meio à revolução digital que ocorre atualmente, a distribuição de multimídia é cada vez mais um foco importante na área tecnológica, quer seja no meio acadêmico quer seja na indústria.

Os aspectos que envolvem a produção, a distribuição e a apresentação de multimídia tornam-se cada vez mais complexos e a padronização apresenta-se como elemento fundamental para nortear o uso de multimídia em um mundo cada vez mais globalizado e digital.

Com o desenvolvimento contínuo das tecnologias para multimídia interativa, um vasto número de aplicações tornou-se possível. As diferentes possibilidades de interação do usuário com os objetos de mídia que compõem o vídeo trazem consigo a necessidade de modelagem dessa interação em direção aos aspectos de adaptação e personalização da interface. Adicionalmente, a convergência de mídias e entre os meios de comunicação tende a permitir a inserção do usuário em um ambiente cada vez mais interativo, passando esse usuário a fazer parte desse ambiente, juntamente com todos os componentes de hardware e software da aplicação, também inserida como componente desse ambiente (Kate, 1998).

Neste capítulo, é apresentada uma visão geral do campo da multimídia interativa e distribuída, com destaque para uma descrição detalhada das características e funcionalidades dos padrões MPEG-4 e MPEG-7.

2.2 Tecnologias associadas à multimídia interativa: uma breve apresentação

No campo da multimídia interativa, muitas são as tecnologias que podem ser utilizadas no desenvolvimento de aplicações. Em linhas gerais, essas tecnologias se aplicam às três fases de um projeto multimídia: produção, distribuição e apresentação.

As tecnologias de produção e distribuição são direcionadas à composição e entrega de objetos de mídia e documentos a um usuário ou grupo de usuários, usufruindo da infraestrutura das redes de comunicação. Existe uma coleção bastante rica de protocolos de distribuição, que contempla desde a distribuição de mídias audiovisuais simples (como é o caso de áudio e vídeo) até as descrições de ambientes complexos (como é o caso de cenas de realidade virtual). Adicionalmente, aspectos de sincronização têm sido investigados no contexto das tecnologias de distribuição, especialmente no caso de ambientes multimídia cooperativos.

As tecnologias de apresentação contemplam os aspectos de protocolos gráficos para exibição de conteúdos e informações de objetos de mídia, bem como para visualização de hiperdocumentos, especialmente no contexto da WWW.

As **Subseções 2.2.1 a 2.2.5** apresentam uma breve descrição de um subconjunto de tecnologias para produção, distribuição e apresentação de multimídia interativa.

2.2.1 SMIL

Um sistema multimídia é uma coleção de componentes de hardware e software relacionados entre si e que devem ser selecionados de forma a funcionarem juntos. De modo geral, esse conjunto de componentes é formado pela placa de vídeo, barramento de dados, controladoras, software de aquisição, software de edição, software de autoria e/ou programação e mídias de distribuição.

Os sistemas multimídia envolvem fortemente a captura, a transmissão e a apresentação de mídias, sendo desejável em muitas aplicações que técnicas de sincronização sejam aplicadas para temporizar a apresentação das múltiplas mídias em uma cena.

A linguagem SMIL, desenvolvida pelas empresas Netscape, RealNetworks, Microsoft, Phillips, Digital Equipment, além de diversas organizações de pesquisa - Columbia University, CWI e INRIA -, é uma especificação do W3C (*World Wide Web Consortium*) para criar uma maneira de enviar fluxos de mídia sincronizados através do ambiente da WWW (W3C, 1998).

A linguagem SMIL é capaz de descrever o comportamento temporal da apresentação, o *layout* da apresentação na tela, além de associar *hyperlinks* a objetos de mídia. Com o uso da linguagem SMIL, pode-se coordenar o *timing* dos objetos em uma apresentação multimídia, na qual cada objeto de mídia é acessado com um único URL, o que viabiliza apresentações de objetos que estão localizados remotamente em lugares distintos.

Num contexto mais amplo, deve-se observar que com o uso da linguagem SMIL, tem-se a possibilidade de criar, por exemplo, um texto introdutório que o espectador pode ler antes do início de um vídeo. Pode-se, também, inserir um texto publicitário sonificado acompanhando a exibição de um vídeo.

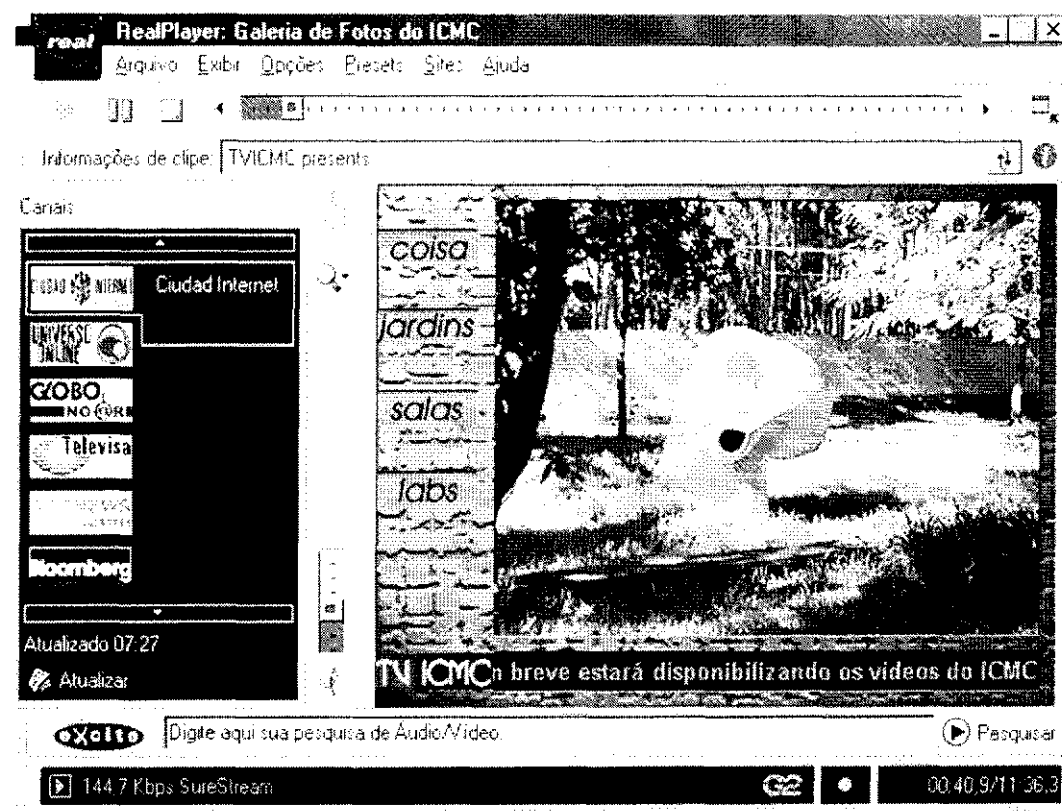


Figura 2.01 - Uma apresentação multimídia protocolada em SMIL

Atualmente, o uso da linguagem SMIL está restrito aos servidores Real, tais como RealText, RealAudio e RealVideo, conforme ilustra a Figura 2.01, mas espera-se que em breve os *browsers* de uso disseminado incorporem mecanismos interpretadores para tecnologia SMIL. Neste ponto, quanto ao uso da linguagem SMIL, pode-se produzir um texto introdutório, que será salvo como arquivo do RealText, seguido de um áudio que é manipulado pelo RealAudio. Por fim, um vídeo pode ser inserido através de um arquivo do RealVideo, e o texto final, sincronizado com o vídeo, pode ser manipulado pelo RealText.

Uma potencialidade adicional da linguagem SMIL reside no fato da apresentação de conteúdos multimídia que necessitam de visualizações seqüenciais e paralelas e que explorem, de maneira temporal, os sentidos de percepção (audição e visão) do leitor, e que contemplem, de forma associada, requisitos de consciência de contexto para identificação de propriedades de apresentação de uma cena e aspectos da infra-estrutura de comunicação (largura de banda, endereços de rede que identifiquem o idioma a ser utilizado), por exemplo.

2.2.2 VRML

A linguagem VRML (*Virtual Reality Modelling Language*) é uma notação baseada em *scripts* de texto que surgiu como um padrão para descrever formas e cenas 3D no ambiente da WWW. Pode ser vista, popularmente, como uma versão 3D da linguagem HTML, sendo que a principal diferença reside no fato de que VRML não descreve texto ou referências gráficas, mas constrói vetorialmente objetos gráficos em três dimensões e poligonais (VRML, 1997).

Deve-se observar que a linguagem VRML é na realidade um formato de arquivo e não uma biblioteca de programação para aplicativos 3D, apesar da mesma dar suporte à criação de *scripts*.

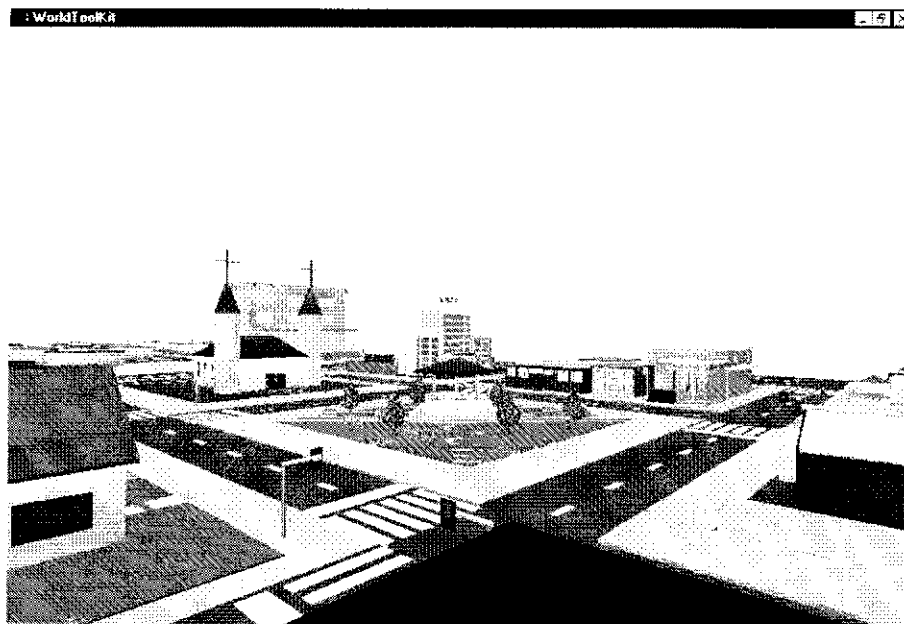


Figura 2.02 – Um “mundo virtual” construído em VRML.

Outra característica inexistente em VRML é a capacidade de interação de múltiplos usuários no mesmo “mundo virtual”. Este elemento básico para a fundamentação de realidade virtual

e comunidades virtuais *on-line* não se encontra implementado nas atuais versões da linguagem, visto que exigem protocolos de comunicação e base de dados para que essas interações possam ser realizadas eficientemente, apesar de ser uma característica que está sendo pesquisada e discutida, para futuras implementações (VRML, 1997).

Aplicações de VRML são amplas e incluem visualização distribuída, interface 3D do usuário com recursos remotos do ambiente da WWW, ambientes 3D colaborativos, simulações interativas, museus virtuais, dentre outras, conforme pode ser visto na **Figura 2.02**.

No caso particular dos ambientes interativos, pode-se aplicar a linguagem VRML na construção de ambientes 3D que permitam a interação simples do usuário através do *mouse*, estimulando a percepção do conteúdo com bom grau de realismo (Churchill, 1998).

2.2.3 HTML e DHTML

HTML (*HyperText Markup Language*) é uma linguagem, criada a partir da meta-linguagem SGML (*Standard Generalized Markup Language*) (ISO, 1986a; ISO, 1986b; Herwijnen, 1994), que se preocupa principalmente em organizar como um documento deve ser apresentado. Além disso, HTML surgiu devido à expansão da Internet e à necessidade de uma linguagem simples para especificação da forma de apresentação dos documentos no ambiente da WWW.

A linguagem HTML possui *tags* (etiquetas) pré-definidas que são utilizadas para especificar a informação e a forma como esta informação deve ser apresentada, tais como <HEAD>, para definir o cabeçalho do hiperdocumento; <H1>, para definir um tamanho de fonte; <TABLE>, para definir uma tabela. A **Figura 2.03** apresenta um documento formatado com o padrão HTML e o resultado da apresentação gráfica desse mesmo documento.

A linguagem HTML apresenta algumas características relacionadas à sua flexibilidade, que constituem seus pontos negativos (Bray et al., 1997; Johnson, 1999):

- HTML não permite a especificação de novos elementos e atributos;
- HTML representa a informação em termos de seu *layout*;
- A alteração de um documento HTML é bastante trabalhosa;
- HTML possui pouca estrutura semântica, pois seus elementos são agrupados sem seguir uma estrutura pré-definida.

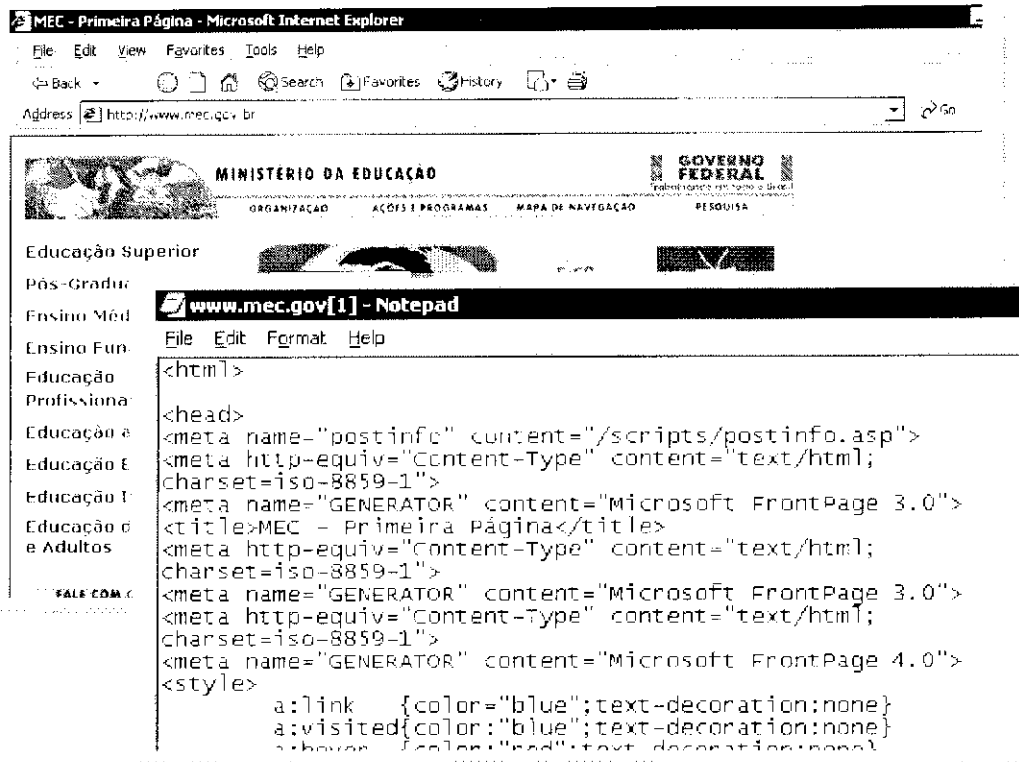


Figura 2.03 – Documento formatado de acordo com o padrão HTML

Apesar destas características, HTML é uma linguagem de singular importância, pois a simplicidade do código e a limitação no número de elementos que a constitui permitem que o seu uso seja extremamente simplificado e que qualquer pessoa com conhecimentos básicos de computação possa implementar um hiperdocumento para ser apresentado no ambiente da WWW. Para suprir as desvantagens desta linguagem, novas tecnologias, mais poderosas, têm sido propostas, como é o caso de DHTML (*Dynamic HTML*).

O advento do DHTML estabelece um novo ponto de referência com relação à interatividade no ambiente da WWW. Até então, os documentos HTML eram, em sua maioria, de uma natureza estática. Por outro lado, a inclusão de *applets* JAVA e controles ActiveX, por exemplo, introduz um certo nível de interatividade; no entanto, a página em si continua a ser algo estático.

Numa perspectiva inovadora, as páginas desenvolvidas com recursos DHTML ganham “vida” com a utilização de *scripts* simples (por exemplo, desenvolvidos em JavaScript),

fáceis de desenvolver, e que tornam verdadeiramente dinâmicos os elementos de uma página, como mostra a ilustração da Figura 2.04.

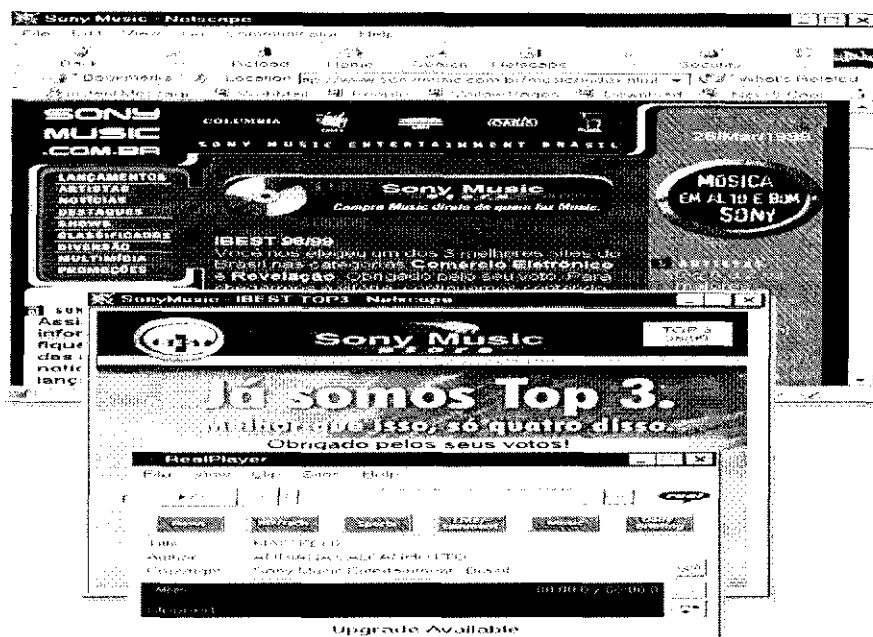


Figura 2.04 - Um exemplo de página HTML com recursos DHTML.

DHTML não é uma extensão da linguagem HTML, nem tão pouco uma nova linguagem de programação. DHTML estabelece um nível superior de interatividade entre os *scripts* e os elementos da linguagem HTML incluídos em uma página. Nesta perspectiva, qualquer um dos elementos de uma página é modificável em tempo real, abrindo novas possibilidades de interatividade como, por exemplo, ações de mostrar e ocultar textos como resposta à interação do usuário.

Dado que DHTML funciona no *browser* do usuário (leitor) - do lado cliente da comunicação cliente/servidor - uma vantagem pode ser imediatamente relacionado: a redução drástica do tempo de resposta às ações interativas. Finalmente, não sendo menos importante, esta tecnologia abre novas perspectivas no que se refere ao desenvolvimento de conteúdo multimídia para aplicações interativas no ambiente da WWW.

2.2.4 XML

A linguagem XML (*Extensible Markup Language*) tem por objetivo fornecer os benefícios inexistentes em HTML, e ser mais simples que SGML.

A linguagem XML permite a construção de marcadores (*tags*) que especificam a estrutura de um determinado documento (Johnson, 1999). A **Figura 2.05** apresenta a definição de um conjunto de *tags* para uma aula no domínio de aplicação ensino, por exemplo. Pela **Figura 2.05**, observa-se a definição de elementos que fazem parte da estrutura de uma aula, sem, no entanto, especificar como tais elementos devem ser apresentados ao leitor, no caso um estudante.

Observa-se que a modelagem é um aspecto relevante em aplicações multimídia. Percebe-se, por exemplo, que a definição de documentos estruturados permite a apresentação controlada de documentos a classes de usuários, o controle do uso do conteúdo pelo usuário, bem como o reuso de classes de documentos em mais de uma aplicação (Pimentel, Santos Jr. & Fortes, 1998).

Como o próprio nome sugere, XML é uma linguagem extensível (meta-linguagem), com a qual os projetistas podem criar seus próprios elementos de acordo com a aplicação que estiver sendo modelada, dando importância ao conteúdo e à estrutura da informação, sem se preocupar com a apresentação (Connolly, 1997). Através desta característica, a flexibilidade da linguagem é ampla, uma vez que o projetista pode criar uma nova linguagem a partir da meta-linguagem XML.

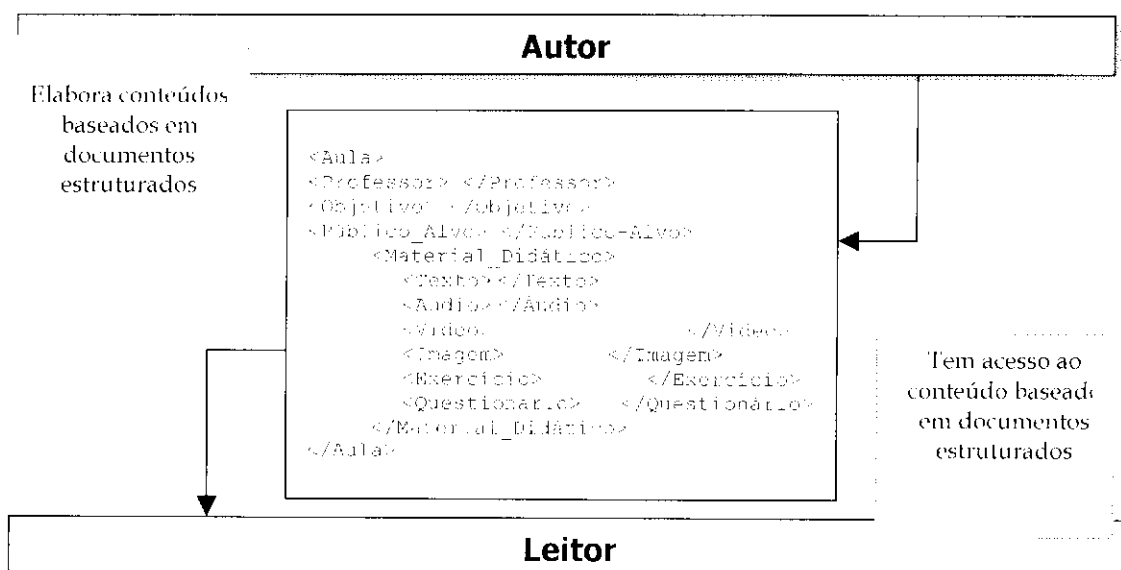


Figura 2.05 - Um exemplo de documento estruturado XML, para o domínio ensino

Através do exemplo da **Figura 2.05**, nota-se que não há nenhum elemento que define como as informações contidas no documento XML devem ser apresentadas, diferente do que acontece em um documento formatado com a linguagem HTML. Sendo assim, pode-se dizer

que XML considera apenas o conteúdo a ser apresentado. Portanto, existe a necessidade da utilização de outro recurso que seja responsável pelos atributos de apresentação. Esta tarefa pode ser realizada através da utilização de programas específicos ou linguagens apropriadas para associar estilos ao conteúdo de um documento XML, como é o caso da tecnologia XSL, apresentada na **Seção 2.2.5**.

A interpretação de um documento XML ocorre através da utilização de *parser*, que analisa o documento obtendo suas informações de conteúdo para processamento. Esta função depende do tipo de *parser* implementado, do propósito da informação e da estrutura semântica apresentada no documento XML. Cada documento XML pode ou não estar associado a uma linguagem específica que define sua estrutura; essa definição é formalizada através da criação de um DTD. Genericamente, um DTD define regras para a especificação de uma classe de documentos, tais como (Halasz, 1994):

- que tipos de elementos podem existir em um documento;
- que atributos esses elementos podem ter;
- como as instâncias desses elementos estão hierarquicamente relacionadas.

A estrutura especificada em um DTD, segundo sua definição no padrão SGML, possui uma propriedade importante: apenas a estrutura lógica de um documento é descrita, não sendo fornecida informação sobre o conteúdo e apresentação dos elementos definidos.

Se o documento XML estiver associado a um DTD, o *parser* deve verificar se o documento XML está correto (*parser* de validação). Para isto o *parser* processa o DTD que corresponde ao documento XML e obtém sua estrutura. Posteriormente, o *parser* obtém as informações do documento XML realizando a validação com a estrutura definida no DTD. Por outro lado, se o documento XML não está associado a um DTD, então apenas a estrutura sintática do documento XML é verificada (*parser* sem validação).

2.2.5 XSL

A manipulação de documentos formatados através do padrão SGML envolve o uso de uma linguagem baseada em LISP, denominada DSSSL (*Document Style Semantics and Specification Language*), para especificar como o documento deve ser apresentado, uma vez que características de apresentação não estão presentes no documento SGML. O mesmo acontece com a utilização de documentos formatados através da linguagem XML. Assim, o grupo

responsável pelo desenvolvimento da linguagem XML desenvolveu também uma linguagem de apresentação - XSL, que é utilizada para formatar o estilo de apresentação do conteúdo de um documento XML. (Johnson, 1999).

Um documento XSL (*Extended Style Language*) tem por objetivo “transformar” um documento XML em outros formatos de documentos, tais como HTML, TeX, *PostScript* e RTF (Lie, 2000). Através do uso de documentos XSL é possível criar múltiplas representações (visões) de uma mesma informação contida em um documento XML. Um exemplo de sintaxe para formatação de um documento XSL é apresentado na **Figura 2.06**.

Um documento XSL é formado por regras, denominadas *templates*. Um *template* é constituído de duas partes: *pattern*, que faz referência a um elemento; e *action*, que constitui o formato aplicado ao elemento referenciado pelo *pattern*. Os *templates* do documento XSL são aplicados ao documento XML e o resultado apresentado é a informação do documento XML formatada e organizada de acordo com as especificações do documento XSL.

```

<xsl:template match="Page">
  <DT> Homepage: </DT>
  <DD>
    P3
  </DD>
  <A>
    <xsl:attribute name="href">
      <xsl:value-of select="."/ >
    </xsl:attribute>
    <xsl:attribute name="target">
      "new"
    </xsl:attribute>
    <xsl:value-of select="."/ >
  </A>
</DD>
</xsl:template>

```

Figura 2.06 - Trecho de documento XSL para formatar um elemento

A linguagem XSL permite a utilização de documentos formatados de acordo com a linguagem XML através de alguns *browsers*, que são capazes de interpretar documentos XSL e XML e gerar apresentações destes documentos, como é o caso do *browser* Internet Explorer versão 5 da Microsoft.

2.3 A problemática da distribuição de multimídia

Um sistema multimídia é uma coleção de componentes de hardware e software relacionados entre si e que devem ser selecionados de forma a funcionarem juntos (Fluckiger, 1995). Esse conjunto de componentes é formado, basicamente, pela placa de vídeo, barramento de

dados, controladoras, software de aquisição, software de edição, software de autoria e/ou programação e mídias de distribuição, conforme ilustra a **Figura 2.07**.

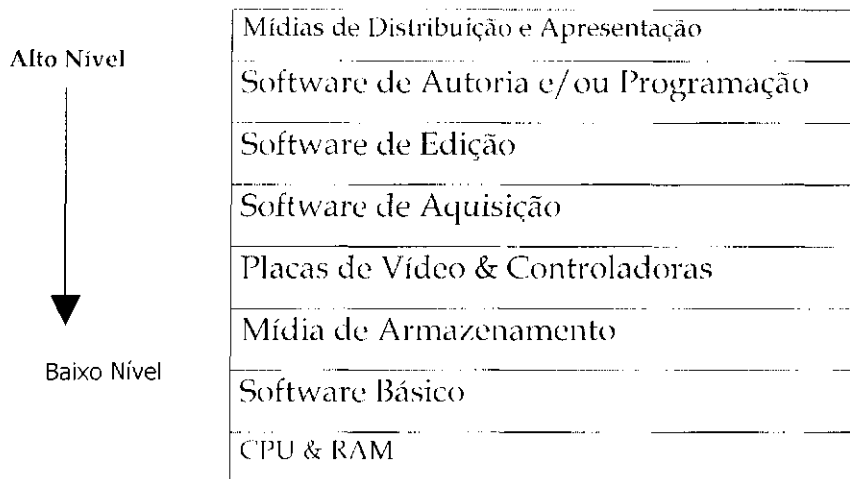


Figura 2.07 - Camadas de um sistema multimídia

O desenvolvimento de aplicações multimídia começa com a escolha e compreensão dos princípios da mídia de distribuição, uma vez que as mídias existentes possuem diferentes velocidades de transferência, conforme apresenta a **Tabela 2.01**.

Tabela 2.01 - Algumas mídias de distribuição e velocidades de transferência

Mídia de Distribuição	Velocidade
CD-ROM 2X	300 KBytes/s
CD-ROM 8X	1,2 MBytes/s
CD-ROM 10X	1,5 MBytes/s
HD (Hard Disk)	5 .. 10 MBytes/s
Memória RAM	30 .. 100 MBytes/s
REDE WAN	1,2 KBytes/s
REDE Ethernet	1,2 MBytes/s
REDE Fast Ethernet (FDDI)	12 MBytes/s

É importante, ainda, conhecer com precisão o espaço de armazenamento requisitado por um arquivo de mídia, especialmente as mídias temporais como áudio e vídeo. A **Tabela 2.02** apresenta alguns exemplos de dados que indicam a quantidade de *bytes* necessários para armazenamento de amostras de mídias como áudio e vídeo digital.

Em suma, deve-se observar dois conceitos importantes, quando do desenvolvimento de sistemas multimídia: sistema de desenvolvimento/autoria e sistema de distribuição. No sistema de desenvolvimento são necessários componentes de hardware como *scanner*, placas de captura de vídeo, câmera fotográfica digital, unidade de CD-ROM, câmeras *Cam-Corder* (câmera filmadora com capacidade de gravação [*recorder*]), microfones e outros. Já no sistema de distribuição, deve-se observar a existência de um hardware com os acessórios essenciais à aplicação. Por exemplo, se a mídia de distribuição utilizada no desenvolvimento for um CD-ROM de velocidade 8X, então no sistema de distribuição (ao nível do usuário) deve-se ter uma unidade de CD-ROM com velocidade 8X ou superior.

Tabela 2.02 - Relação das taxas de armazenamento para algumas mídias

Mídia	Taxa de Amostragem	Número de Bits	Armazenamento
Áudio	44,1 samples/s	16 bits	88,2 Kbytes/s
Vídeo	8 Msamples/s	24 bits (true color) 30 frames/s	1,6 Mbytes/s

No contexto da multimídia, potencialmente, observa-se que o desenvolvedor pode usufruir de recursos que permitam a sofisticação das aplicações. Neste ponto, existem algumas regras que podem ser usadas por usuários e desenvolvedores para nortear e avaliar sistemas multimídia/hipermídia, a saber (Harrisson, 1996):

- a regra da multimídia: um sistema hipermídia não deve manipular somente os tipos de dados básicos (caracteres, números e datas), mas também proporcionar sofisticações ao sistema sensorial do ser humano através de gráficos coloridos em 2D e 3D, áudio, imagem, vídeo, texto e *widgets* (objetos gráficos, como botões, ícones e outros);
- a regra do objeto: um sistema hipermídia deve empregar metáforas de objetos na interação, armazenamento e criação de aplicações;
- a regra dos *scripts*: um sistema hipermídia deve fornecer linguagens de *script* acessível que permitam a extensão e a modificação do comportamento do sistema e dos seus elementos, onde entende-se por *script* como sendo um programa curto que pode ser relacionado com um objeto para especificar seu comportamento sob várias condições;
- a regra multi-usuário: um sistema hipermídia deve suportar a construção cooperativa de aplicações por vários autores, e a execução dessas aplicações por

vários usuários concorrentes, onde o acesso a objetos, através de redes de computadores e ambientes cliente-servidor cooperativos, deve ser de forma concorrente e transparente para os autores e usuários;

- a regra da escalabilidade: as aplicações hipermídia devem executar, praticamente, com as mesmas características de desempenho tanto no ambiente de produção quanto no ambiente de uso real;
- a regra da interoperabilidade: as aplicações hipermídia devem proporcionar o intercâmbio de dados entre elas, e também com aplicações externas;
- a regra de *hyperlink* e *hyperview*: um sistema hipermídia deve permitir ao usuário estabelecer relacionamentos navegáveis entre objetos multimídia (*hyperlink*), explorar e navegar por esses relacionamentos de forma não-linear, determinar sua localização entre eles a qualquer momento da visualização/apresentação (*hyperview*), e retornar facilmente para qualquer localização anterior em um caminho navegado;
- a regra da independência tecnológica: um sistema hipermídia deve executar em vários ambientes operacionais distintos e garantir que as aplicações hipermídia serão executadas sem modificações à medida que a tecnologia evolui;
- a regra da extensibilidade: um sistema hipermídia deve permitir a implementação de novas funções ao sistema e modificar a interface com o usuário, de maneira a solucionar diversas classes de problemas e acomodar usuários de categorias distintas;
- a regra multi-idioma: um sistema hipermídia deve suportar interação com o usuário, armazenamento e recuperação de informação, em idiomas distintos, considerando-se os aspectos de globalização da economia, por exemplo;
- a regra do suporte ao desempenho: um sistema hipermídia deve interagir com o usuário através de uma interface gráfica amigável, empregar mídias na comunicação com o usuário e oferecer níveis de ajuda *on-line*, acomodando usuários iniciantes, intermediários e experientes;
- a regra dos padrões: um sistema hipermídia deve suportar e obedecer os principais padrões formais e de mercado, garantindo a compatibilidade com outros produtos da área.

No contexto dos sistemas distribuídos, um ambiente multimídia deve suportar a construção cooperativa de aplicações por múltiplos autores e a execução dessas aplicações e

apresentação de informações a múltiplos usuários em redes de computadores, geralmente formadas por componentes heterogêneos (Coulouris et al., 1994).

O estado da arte revela a consolidação das redes de computadores e usuários colaborando eletronicamente em ambientes cooperativos. Aplicações complexas podem ser desenvolvidas por equipes isoladas fisicamente e que utilizem computadores e redes de diferentes fabricantes com diferentes sistemas operacionais e protocolos de comunicação, formando um sistema integrado via Internet.

Assim, aplicações multimídia interativas devem estar potencialmente associadas a ambientes cliente-servidor, nos quais objetos de uma aplicação podem ser acessados por outros usuários, sendo que o sistema deve resolver conflitos quando dois ou mais usuários tentarem acessar e modificar um mesmo objeto. Deve-se considerar, ainda, que não somente a aplicação e o processamento devem ser distribuídos, mas também o armazenamento e o gerenciamento, o que exige a implementação de mecanismos que garantam transparência e permitam ao usuário utilizar a rede como se fosse o seu próprio equipamento. A Internet usa o protocolo TCP/IP (*Transmission Control Protocol – Internet Protocol*), criado em 1972, que a princípio não foi feito para acomodar a evolução tecnológica que se observa atualmente (Comer, 1995). Desta forma, a nova geração de protocolos IPng (*Internet Protocol next generation*), em fase de consolidação, deve suportar, de forma mais adequada, as exigências de tráfego da rede mundial.

Na Internet, os pacotes trafegam da origem ao destino passando por roteadores com tecnologias distintas no aspecto físico (Telefonia, Ethernet, RDSI, dentre outros), conforme apresentado na **Seção 2.5**, o que caracteriza a alta taxa de conectividade e a interconexão de componentes heterogêneos. A **Figura 2.08** apresenta uma arquitetura cliente-servidor característica da Internet.

Para que as aplicações multimídia possam ser implementadas com características cooperativas e alcançar os benefícios da computação distribuída, são necessários formatos comuns para o intercâmbio de objetos. Não é suficiente padronizar apenas os formatos de mídias individuais. As relações temporais, espaciais, estruturais e procedimentais entre os componentes de mídia são uma parte importante e também devem ser representadas.

Atualmente, sabe-se que a falta de um formato comum é um impedimento para o desenvolvimento de diversas classes de aplicações multimídia (Moreira et al., 1995).

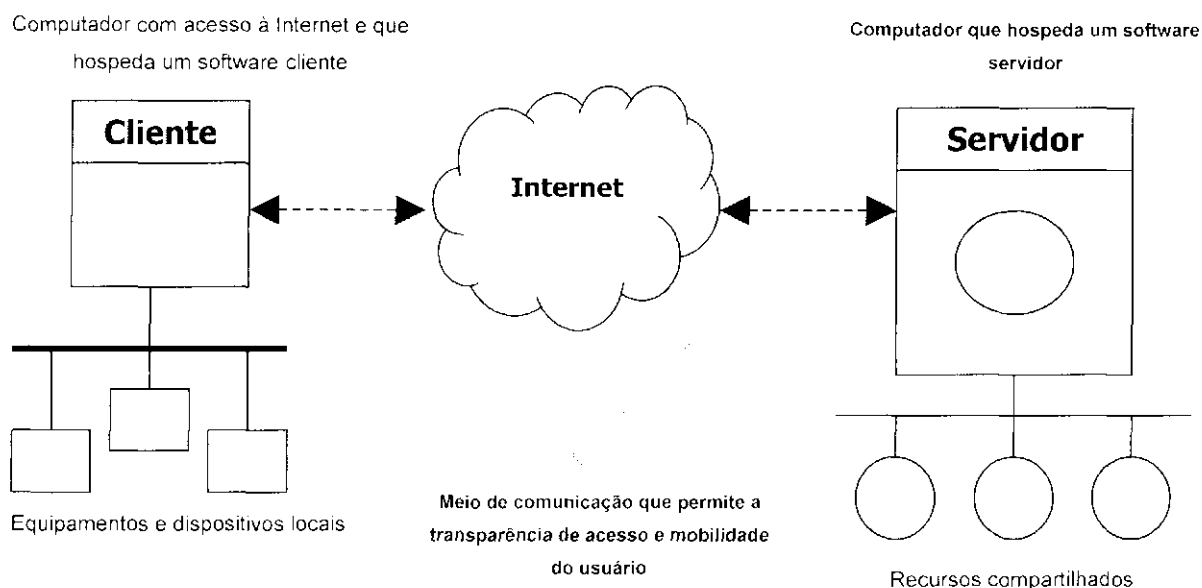


Figura 2.08 – Estrutura genérica da arquitetura cliente-servidor para a Internet

Existem diversos formatos para os objetos multimídia e que permitem o estabelecimento de intercâmbio entre as aplicações e/ou ambientes de autoria e apresentação, especialmente na troca de documentos.

Um padrão para representação de documentos deve considerar, também, a possibilidade de ligação entre documentos diferentes. Exemplos destas ligações são, por exemplo referências bibliográficas que levam ao documento original e palavras-chave que levam a documentos mais específicos. Estas ligações são chamadas de *links*. Quando o usuário ativa um *link*, ele é levado a um novo documento ou a um novo ponto dentro do documento correntemente visualizado.

Em um documento também se encontram elementos de controle, denominados *markups* (*tags*) ou etiquetas, e esses elementos devem obedecer a padrões de formatação de documentos, tais como o SGML, HTML e, mais recentemente, XML.

Para o intercâmbio de imagens também existem padrões bem definidos. Uma imagem com qualidade fotográfica possui 24 *bits* de informação de cor por *pixel*, o que resulta em um tamanho de arquivo grande e impraticável para ser enviado por uma rede. Usa-se, então, algoritmos de compressão que podem levar a uma redução no tamanho de até 99%. A

aplicação desses algoritmos pode resultar em mídias com perda (*lossy*) ou sem perda (*lossless*). Um algoritmo com perdas resulta em uma imagem de pior qualidade, mas com um tamanho menor. A compressão com perdas consiste de uma troca de qualidade por tamanho. Alguns padrões de compressão e armazenamento de imagens são o GIF (*Graphics Interchange Format*) e o JPEG (*Joint-Photograph Expert Group*).

Fotos, gráficos e desenhos são imagens de duas dimensões. Uma imagem de três dimensões requer um formato completamente diferente, pois o usuário poderá visualizar os objetos representados em qualquer ângulo que desejar. Atualmente, a linguagem VRML - uma linguagem de descrição de cenas de realidade virtual - permite o envio de objetos tridimensionais pela Internet e a interação do usuário com a cena apresentada em uma janela de visualização.

Para a mídia de áudio existem formatos definidos de acordo com a natureza do som. Um dos tipos de áudio é o áudio em forma de onda, no qual uma onda é um sinal original de áudio convertido para o formato digital através de uma placa de som. Estas informações digitais podem ser examinadas e manipuladas através dos programas de edição de áudio em forma de onda. Este tipo de áudio é produzido através do hardware da placa de som, que transforma as informações digitais novamente em sinais de áudio analógicos e os envia para um amplificador equipado com alto-falantes. Os arquivos de formas de onda possuem, nos computadores pessoais, a extensão *WAV* (*wave* = onda).

O segundo tipo de áudio é o MIDI (*Musical Instrument Digital Interface*), um padrão internacional que especifica cabos e o hardware utilizados na conexão entre computadores e instrumentos musicais eletrônicos. O padrão MIDI também inclui um protocolo de comunicação para a transferência de dados entre dispositivos. Um arquivo de som em formato MIDI contém informações que descrevem quais notas devem ser tocadas por um sintetizador MIDI de instrumentos.

O vídeo digital sempre foi um tema que atraiu muito interesse, pois reúne qualidade de imagem com facilidade de edição. Há anos, as principais companhias do mercado de sistemas iniciaram a produção de extensões multimídia que incorporavam o vídeo digital. Os formatos criados por estas companhias, entretanto, eram proprietários e os vídeos só rodavam na sua própria plataforma. Para resolver isto, os fabricantes desenvolveram *video*

players para as demais plataformas. Neste ponto, tornaram amplamente utilizados os formatos QuickTime da Apple Computers e o AVI (*Audio Video Interchange*) da Microsoft, tanto para gravação de vídeo em CD-ROM quanto para disponibilização na Internet, apesar de serem inferiores quando comparados às imagens de televisão. Devido ao alto volume de dados que um vídeo digital contém e às altas taxas de largura de banda exigidas para transferência em redes de computadores, pesquisadores procuraram, então, um novo algoritmo de compressão que pudesse trazer a qualidade dos VCRs para os computadores, sendo que o trabalho desses pesquisadores resultou no padrão MPEG, que será apresentado, em suas *releases* 4 e 7, na **Seção 2.6**.

No contexto do intercâmbio de objetos multimídia, um modelo de composição é uma lista que define a ordem em que os objetos multimídia serão acessados e visualizados, sendo usados para transportar estes objetos pela rede. É importante ressaltar que um modelo de composição não fornece formatos para armazenar uma imagem ou um vídeo, mas formatos para transportar estes objetos.

Os modelos de composição fornecem recursos para a estruturação de mídias interativas representando os relacionamentos temporais, sincronização, formatos de mídias e acesso aos objetos de mídia. Alguns modelos também permitem a integração de *scripts* aos objetos. Isto lhes dá características dinâmicas. Um bom modelo de composição deve ser eficiente em relação ao armazenamento e acesso aos objetos.

Em linhas gerais, existem dois modelos básicos: modelo de trilhas e modelo de objetos. No modelo de trilhas, a seqüência de acesso primário é temporal e a apresentação multimídia é vista como uma seqüência de trilhas. No modelo de objetos, a seqüência de acesso primário é hierárquica e a apresentação multimídia é vista como ligações entre objetos. Como principais representantes do modelo de trilhas existem o QuickTime e o OMFI (*Open Media Framework Interchange*). Já como modelo de objetos, o principal representante é o MIEG (*Multimedia Hypermedia Expert Group*) (Buford et al., 1994; ISO, 1995).

Ainda no contexto do intercâmbio de objetos multimídia, devem ser abordados os aspectos referentes a relacionamentos temporais entre objetos multimídia e distribuição dos mesmos numa apresentação.

Quando se integra as mídias numa apresentação, deve-se criar relacionamentos temporais entre essas mídias, fazendo com que sejam apresentadas seqüencialmente ou simultaneamente (em paralelo).

Para representar estes relacionamentos, utiliza-se de modelos que representam todas as possibilidades de combinações (seqüenciais ou paralelas) de objetos multimídia. Além disso, existem técnicas para combinar tais objetos visando a melhor qualidade da apresentação, principalmente no aspecto sincronização. A representação de relacionamentos pode ser feita tanto graficamente, através de *timelines* ou Redes de Petri, como por meio de *scripts*, representando combinações seqüenciais, paralelas e *loops*, através de linguagens de *scripts*, por exemplo.

As operações que um sistema multimídia deve suportar também são relevantes. Tais operações são avanço, retrocesso (apresentação reversa), pausa, retomada da apresentação, acesso aleatório, *looping* e *browsing*. Estas operações são o resultado da interação do usuário com o sistema multimídia. Uma atenção especial é dada para a operação de *browsing*, onde a interação com o usuário leva a uma apresentação não-seqüencial dos dados, na qual o usuário decide sobre o próximo trecho a ser apresentado.

Em ambientes distribuídos, o problema da distribuição dos objetos multimídia consiste em assegurar que os objetos multimídia de uma apresentação, que podem estar fisicamente em diferentes lugares, sejam apresentados com sincronismo. Assim, tecnologias podem ser combinadas e integradas de modo a assegurar que os objetos sejam apresentados no instante de tempo previsto, de modo que haja continuidade na apresentação.

2.4 Aspectos de qualidade de serviço em ambientes interativos

Sistemas multimídia utilizam tanto mídias discretas (como texto e gráficos) quanto mídias contínuas (como áudio e vídeo). Os dados de mídia contínua (ou dados baseados no tempo) são gerados em uma determinada taxa e devem ser reproduzidos nessa mesma taxa, sendo que existe um prazo (*deadline*) para que o dado seja exibido sem perda na qualidade (Buford, 1994). Como exemplo, se uma aplicação deseja exibir *frames* a uma taxa de 30fps (*frames* por segundo) e em algum momento da transmissão (imaginando que a exibição é remota) essa taxa diminui, ocorrerão tremores na imagem (*jitter*) que estiver sendo exibida. As tecnologias atuais encontram algumas dificuldades em suportar essas características de tempo real dos

dados de mídia contínua, de forma que muitas pesquisas têm sido feitas no sentido de superar essas dificuldades.

Servidores de dados multimídia devem suportar transmissão a clientes heterogêneos e garantir que o vídeo seja apresentado com a mesma qualidade com a qual foi gerado, e isto requer parâmetros de qualidade de serviço (QoS - *Quality of Service*) (Hashisuca, 1996).

Entende-se por qualidade de serviço como sendo a parametrização, para o sistema, da vontade ou necessidade do usuário. Por exemplo, se um cliente deseja transmitir um vídeo no formato NTSC, a taxa de *frames* deve ser de 30fps, e os sistemas devem garantir que essa taxa seja mantida. Os parâmetros de QoS podem estar relacionados com a aplicação, com o sistema operacional, com a rede ou com o dispositivo (Hashisuca, 1996).

Num estágio mais atual, observa-se que os aspectos de consciência de contexto também podem influenciar a qualidade de serviço, especialmente em termos do controle eficiente das ações do usuário e das reações do ambiente em face das mídias válidas nesse ambiente.

O gerenciamento de QoS para aplicações de mídia contínua pode ser subdividido em dois esquemas de controle: *estático* e *dinâmico* (Buford, 1994). No esquema de controle estático, o usuário simplesmente especifica o nível de QoS no momento da criação de uma determinada sessão, sendo que esse nível será mantido até o final dessa sessão.

Por outro lado, o esquema de controle dinâmico permite que o sistema ou o programa de usuário modifique o nível de QoS durante o andamento da sessão, o que é relevante para aplicações com consciência de contexto. A **Figura 2.09** apresenta as camadas de QoS em um ambiente interativo.

Quanto à qualidade de serviços, as aplicações VoD (*Video-on-Demand*) apresentam-se como importantes representantes dos sistemas interativos, sendo divididas, genericamente, em categorias que incluem televisão interativa e aplicações corporativas/educacionais. O foco da TVI é a apresentação de vídeo a partir de uma pequena coleção. O foco das aplicações corporativas/educacionais é a apresentação aleatória de dados de vídeo, selecionados a partir de uma coleção de vídeos de tamanhos variados. Um sistema VoD para TVI pode armazenar dados de vídeo economicamente em discos magnéticos. Contudo, devido ao grande volume de dados de vídeo armazenados para uma aplicação

corporativa/educacional, um sistema VoD para esse tipo de aplicação tipicamente usa uma combinação de discos magnéticos com outros meios de armazenamento (Berger, 1995).

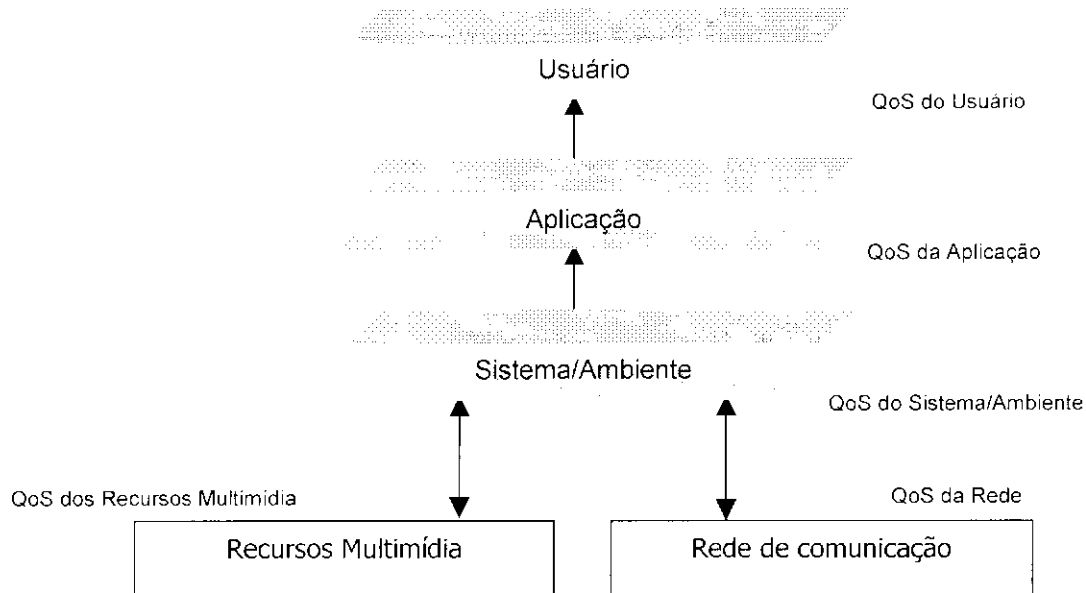


Figura 2.09 - Níveis de QoS em um ambiente interativo

No desenvolvimento de um sistema VoD, pesquisadores têm que resolver muitos problemas. Primeiro, o sistema deve estar apto a localizar um vídeo desejado. Segundo, o sistema deve armazenar e gerenciar coleções de dados de vídeos estruturados. Terceiro, o sistema deve transportar um vídeo selecionado e apresentá-lo ao usuário final (Berger, 1995).

Para localizar um vídeo desejado, uma interface amigável deve ser criada, permitindo que se realize pesquisas em uma base de metadados com vistas à localização de um vídeo em um repositório. O projeto da interface de pesquisa deve considerar quais tipos de pesquisas devem ser suportadas e como as pesquisas devem ser especificadas. Adicionalmente, aspectos de consciência do contexto do usuário podem ser considerados durante o processo de pesquisa e seleção de um vídeo.

Serviços de vídeo interativos são similares ao serviço de difusão de TV. A diferença é que a interface da TV foi projetada para suportar muitos canais simultâneos sobre um mesmo banco de dados. No sistema de TV tradicional, cada estação realiza a sua programação simultaneamente com outras estações, e o usuário seleciona um canal específico. Em contraste, um sistema interativo faz com que toda a programação seja apresentada ao usuário sem nenhuma restrição. O usuário pode selecionar qualquer parte da programação a qualquer hora (Little & Venkatesh, 1994).

De acordo com o nível de interatividade, os serviços de vídeo podem ser classificados em (Hachisuca, 1996):

- serviços *broadcast* (No-VoD): similar aos canais de televisão, em que o usuário é um participante passivo e não possui nenhum controle sobre a sessão;
- serviço *pay-per-view* (PPV): o usuário paga por uma programação específica, similar ao serviço existente em CATV (*Cable TV*);
- serviço *quasi video-on-demand* (Q-VoD): os usuários são agrupados pelo interesse comum que possuem; os usuários possuem um controle rudimentar, ativado por chaveamento para os diferentes grupos;
- serviços *near video-on-demand* (N-VoD): funções de avançar e voltar são simuladas por transições em um intervalo de tempo discreto (da ordem de 5 minutos); isto é feito utilizando-se vários canais que possuem a mesma programação, porém em tempos diferentes;
- serviço *true video-on-demand* (T-VoD): o usuário possui total controle sobre a apresentação, manipulando todas as funções de VCR, incluindo avançar, voltar, congelar e posicionar diretamente, dentre outras; é alocado um canal para cada usuário, o que limita o número de usuários do serviço.

Os serviços interativos podem ser usados nas mais diversas aplicações, como visto na **Tabela 2.03**. Um sistema VoD pode ser acessado simultaneamente por vários usuários com diferentes preferências. O sistema deve ser capaz de garantir uma qualidade de serviço satisfatória a todos os clientes. Um cenário típico de um VoD consiste em um banco de dados local, conectado a uma residência de um usuário por uma rede de comunicação. Na residência do usuário, há uma interface de rede e um monitor.

Tabela 2.03 – Algumas aplicações de serviços interativos (Hachisuca, 1996)

Aplicação	Descrição
<i>Movies-on-Demand</i> (Filmes sob Demanda)	O consumidor pode selecionar e ver o vídeo com total capacidade de interação
Vídeo-Game Interativo	Pode-se jogar sem ter que adquirir uma cópia do jogo
Noticiário de TV interativo	O consumidor é capaz de selecionar uma notícia bem como seu nível de detalhe
Catálogos interativos	O consumidor pode examinar e comprar produtos
Educação a distância	Um estudante se inscreve em um curso, remotamente. O estudante pode ter preferências e horários individuais

A interação do usuário com o sistema pode ser feita através de um controle remoto ou um teclado de computador. Esse sistema consiste da informação/arquivo de programação armazenada em um servidor central. Essa informação é transmitida a servidores regionais através de uma rede de comunicação de alta velocidade, ficando à disposição dos usuários.

Sessões VoD são geralmente longas e requerem alta banda passante para transmissão, necessitando, então, de técnicas de compressão de vídeo, tais como MPEG, MJPEG (*Motion-Joint Photographic Expert Group*), H.26, dentre outras, para diminuir o número de *bits* a serem transmitidos (Berger, 1995).

2.5 A infra-estrutura da integração de sistemas digitais: uma breve descrição

Assim como existem especificações para um computador e/ou equipamento local (*desktop*), existem especificações (requisitos) para aplicações em redes de comunicação. Dentre esses requisitos encontram-se a largura de banda adequada, a pouca variação de atraso, o pouco atraso máximo, a conexão ponto-a-ponto ou multiponto e a escolha de comutação por *circuit switching* ou *packet switching*.

O padrão LAN Ethernet é *packet switching*, permitindo a comunicação bidirecional não-determinística em relação ao tempo do trajeto dos pacotes. O padrão ATM (*Asynchronous Transfer Mode*) pode ser usado em LANs (*Local Area Network*), MANs (*Metropolititan Area Network*) e WANs (*Wide Area Network*), e atualmente sua aplicação em AANs (*All Area Network*) está em expansão, sendo do tipo *circuit switching* e determinístico (Tanenbaun, 1998). Observa-se, então, que padrões como ATM apresentam características interessantes para suportar o tráfego de objetos multimídia em ambientes interativos.

A comunicação de dados apresenta-se como um avanço tecnológico de grande importância para a humanidade, uma vez que permite a interligação de computadores, dispositivos e equipamentos diversos, possibilitando às pessoas se comunicarem de forma abrangente e transparente em termos das complexidades tecnológicas exigidas.

A base para a implantação e difusão das redes de comunicação são as redes locais de computadores. Entende-se por rede local como um suporte de comunicação para interconexão de equipamentos numa área restrita, sendo que a interconexão de

equipamentos é desejada para viabilizar o compartilhamento dos recursos computacionais de hardware, software e de informação.

A organização básica de uma rede local consiste de um número de equipamentos, tais como computadores, impressoras, terminais, servidores especializados (impressão e de disco, por exemplo) que são interligados de diversas maneiras, tais como por cabo coaxial, par trançado, fibra ótica ou mesmo por sinais de rádio (redes sem fio), permitindo o acesso por um determinado número de usuários.

Com o crescimento do número de usuários nas redes e também o com aumento de aplicativos gráficos, multimídia e cooperativos, a performance das redes locais com tecnologia Ethernet a 10Mbps (*megabits* por segundo) está sendo abalada. Para quebrar a barreira dos 10Mbps, surgiram as tecnologias de redes de alta velocidade como o ATM e o padrão IEEE 100 Base-T Fast Ethernet.

Nas **Subseções 2.5.1 a 2.5.11** são apresentadas as características principais de alguns padrões para comunicação de dados e que formam a infra-estrutura básica de comunicação em uso.

2.5.1 Redes ATM

O ATM é uma forma de tecnologia baseada na transmissão de pequenos pacotes de tamanho fixo e estrutura definida denominados células. Estas células são transmitidas através de conexões de circuitos virtuais estabelecidos, sendo sua entrega e comutação feitas pela rede de comunicação, com base na informação de seu cabeçalho.

A tecnologia ATM se adapta facilmente às exigências de uma vasta gama de tráfegos, suportando com isto diferentes tipos de serviços, o que propiciou sua escolha para oferecer suporte à implantação da Rede Digital de Serviços Integrados - Faixa Larga RDSI-FL (*Broadband Integrated Services Digital Network - B-ISDN*) (Tanenbaun, 1998).

Numa rede ATM, um canal é identificado pela posição fixa de seus *slots* dentro dos *frames*, sendo que a banda passante é dividida em segmentos fixos de informação (células). Cada célula possui um cabeçalho de informação para que o receptor possa identificar o canal ao qual uma célula pertence, e tomar as medidas cabíveis. Um canal não é mais identificado de maneira estática por uma posição no tempo, mas de maneira dinâmica através das

informações contidas nos cabeçalhos. Assim, não é necessário definir um mapeamento entre as posições no tempo e os canais envolvidos na transmissão, pois esta informação de mapeamento está contida no cabeçalho. Os pontos comutadores da rede podem também lidar de maneira homogênea com o chaveamento de todos os serviços.

Observa-se, então, que o fato de se poder alocar dinamicamente uma determinada quantidade de banda passante é de grande valia para a transmissão de informações cujo tráfego pode ser caracterizado por pacotes de dados distribuídos no tempo, evitando-se o desperdício de banda nos períodos de latência da fonte geradora.

2.5.2 O padrão Fast Ethernet

Com o crescimento do número de usuários nas redes e também com o aumento de aplicativos gráficos, multimídia e cooperativos, a performance das redes locais com tecnologia Ethernet a 10Mbps está sofrendo abalos.

Para quebrar a barreira dos 10Mbps, surgiram as tecnologias de redes de alta velocidade como o ATM e o padrão IEEE 100 Base-T Fast Ethernet (Tanenbaun, 1998). O padrão 100 Base-T de Ethernet a 100Mbps mantém as principais características do padrão Ethernet 10Mbps, tais como o formato do *frame*, a quantidade de dados que um *frame* pode transportar, e o mecanismo de controle de acesso ao meio, diferenciando do padrão original apenas na velocidade de transmissão dos pacotes, que no padrão 100 Base-T é 10 vezes maior que no original.

Os meios físicos padrão 100Base-TX e 100Base-FX, usados no Fast Ethernet, foram originalmente desenvolvidos pelo ANSI para o padrão FDDI, e são amplamente utilizados em redes locais FDDI. A conexão ao meio físico é feita através da interface dependente do meio, que consiste de um conector de par trançado ou de fibra ótica de oito pinos.

A Interface Independente do Meio (*Media Independent Interface*) é um conjunto de eletrônicos opcionais que provê uma maneira de ligar as funções de controle de acesso ao meio do dispositivo de rede com o dispositivo da camada física (PHY), o qual envia os sinais para o meio físico. A MII pode, opcionalmente, suportar tanto operações a 10Mbps como a 100 Mbps, permitindo que dispositivos de rede possam conectar tanto segmentos 10 Base-T como 100 Base-T.

É importante notar que as portas Ethernet dos repetidores não usam uma interface Ethernet. Uma porta do repetidor conecta-se ao sistema do meio FastEthernet usando os mesmos equipamentos PHY e MDI. No entanto, as portas dos repetidores operam ao nível de *bit* individual para sinais Ethernet, movendo os sinais diretamente de segmento para segmento. Entretanto, as portas dos repetidores não contêm interfaces Ethernet já que elas não operam ao nível de *frames* Ethernet.

2.5.3 O padrão Gigabit Ethernet

Em novembro de 1995, a comissão IEEE 802.3 encarregou o *Speed Study Group* para viabilizar um incremento na capacidade das redes Fast Ethernet, buscando o que foi denominado de rede *Gigabit* (Tanenbaun, 1998).

Em meados de 1996, o IEEE aprovou a solicitação do projeto de desenvolvimento para o Gigabit Ethernet, e formou o *802.3z Gigabit Ethernet Task Force*, com alguns objetivos iniciais, a saber:

- alcançar a velocidade de 1000 Mbps;
- usar o formato de *frames* do padrão Ethernet 802.3;
- alcançar os requisitos funcionais do padrão 802;
- possibilitar a comunicação entre 10Mbps, 100Mbps e 1000Mbps;
- manter o tamanho mínimo e máximo de *frames* dos padrões anteriores;
- prover operações *half* e *full-duplex*;
- suportar a topologia estrela;
- utilizar o método de acesso CSMA/CD com suporte para ao menos um repetidor por subrede;
- suportar o canal de fibra ANSI FC-1 e FC-0 e, se possível, cabo de cobre;
- fornecer uma família de especificações de nível físico;
- permitir controle de fluxo;
- especificar uma Interface Independente de Mídia Lógica (GMII).

Assim, tem-se uma tecnologia derivada do padrão Ethernet, que oferece uma largura de banda de 1000 Mbps (1 Gbps), sendo 100 (cem) vezes mais rápida que o padrão Ethernet, mas compatível com este último e usando os protocolos CSMA/CD e MAC.

2.5.4 Comunicação sem fio (*wireless*)

Muitos sistemas de comunicação fazem a transmissão dos dados utilizando fios de cobre (como par trançado e cabo coaxial) ou fibra ótica. Outros, entretanto, transmitem os dados pelo ar, não utilizando qualquer tipo de meio físico, como é o caso da transmissão por raios infravermelhos, laser, microondas e rádio. Cada uma destas técnicas é adequada a certas aplicações, que podem ser empregadas em LANs e WANs (Tanenbaun, 1998).

Na realidade, o ar (ou espaço livre) constitui-se de um meio natural para a propagação de sinais eletromagnéticos, podendo, em muitos casos, ser considerado o melhor suporte de transmissão, no que se refere à conectividade. Tal afirmação baseia-se no fato de que o ar provê uma interconexão completa e permite maior flexibilidade na localização das estações.

Existem também alguns inconvenientes com relação ao sistema de comunicação sem fio, sendo que os principais são o custo dos equipamentos e a regulamentação pública.

A escolha de canais de radiofrequência para sistemas de comunicação, de uma forma geral, é bastante complexa, pois vários fatores devem ser observados, dentre os quais pode-se citar:

- banda passante desejada;
- área de cobertura;
- disponibilidade do espectro;
- interferências e fontes de ruído;
- regulamentação pública;
- custos dos equipamentos.

Como se pode observar atualmente, o cabo já é uma alternativa concreta à TV por radiodifusão; o celular está ganhando o espaço dos telefones convencionais e, em alguns casos, o cabeamento da LAN está cedendo sua posição para ondas de rádio e infravermelho.

Muitos fabricantes competem em um mercado impulsionado pela necessidade de substitutos para sistemas com fio em instalações de redes de comunicação, e uma gama extensa de produtos tem sido desenvolvida e colocada à disposição do usuário.

2.5.5 Comunicação via satélite

Os satélites artificiais são largamente empregados em telecomunicações, podendo ser classificados em geo-estacionários ou não geo-estacionários, de acordo com sua órbita, podendo prover meios de comunicação nas seguintes categorias:

- ponto a ponto;
- ponto a multiponto;
- multiponto a ponto;
- multiponto a multiponto.

A comunicação via satélite apresenta algumas vantagens em relação aos meios tradicionais, principalmente no que diz respeito à largura de banda disponível, mas também apresenta limitações, sendo a maior delas a vida útil relativamente curta (8 a 10 anos).

Os elementos básicos do serviço de comunicação por satélite são divididos entre o segmento espacial e o segmento terrestre, e este serviço pode se dar nas bandas C e KU de frequência (Tanenbaun, 1998).

A Banda C é uma faixa de frequência utilizada na comunicação com satélite no espectro de frequência de 3.9GHz até 6.2 GHz - segundo o IEEE - e espectro de frequência comercial de 3.7GHz até 6.425GHz, sendo utilizado um sinal de frequência 6GHz para comunicação no sentido terra-satélite e 4GHz no sentido satélite-terra.

A Banda KU é uma faixa de frequência utilizada na comunicação com satélite no espectro de frequência de 15.35GHz até 17.25 GHz - segundo o IEEE - e espectro de frequência comercial de 10.7GHz até 18GHz, sendo utilizado um sinal de frequência 14GHz para comunicação no sentido terra-satélite e 12GHz no sentido satélite-terra.

2.5.6 Microondas

São sistemas de transmissão via rádio que operam na faixa de 900Mhz até 30.000Mhz no espectro de frequência, sendo também conhecidos como "*line-of-sight microwave systems*" (Tanenbaun, 1998). Estes sistemas são usados para transportar sinais analógicos ou digitais de voz e dados.

Nestas frequências, as ondas de rádio se comportam praticamente como ondas de luz e sua propagação segue uma linha reta, de onde se conclui que não devem existir obstáculos sólidos em meio a esta linha.

Uma característica importante destes sistemas é a possibilidade de se prever o nível do sinal que é recebido pelo receptor distante com uma precisão conhecida.

Em linhas gerais, a transmissão de rádio em frequências de microondas é uma alternativa interessante, sendo que sua aplicação pode ser observada em troncos para telefonia (antigamente estes mesmos troncos serviam para transmissão de TV), em *links* ponto a ponto para um backbone de uma rede WAN ou LAN, em sistemas multiponto para telefonia ou dados, e no transporte de sinais de vídeo, tais como CATV (televisão a cabo).

2.5.7 Telefonia celular e comunicação de dados

Nos últimos vinte anos, tecnologias de comunicação móvel evoluíram de protótipos experimentais ao sucesso comercial.

A telefonia celular é um sistema de telecomunicação que permite a conexão de assinantes (fixos ou móveis) através de um enlace de rádio, enquanto que na telefonia clássica os assinantes são conectados à central de comutação através da dispendiosa instalação de cabos para composição da rede telefônica.

As redes de telefonia celular existentes foram otimizadas para tráfego de voz, com taxa de transmissão de até 4,8 Kbps, tipicamente. Redes digitais, destinadas ao transporte de dados e voz digitalizada, estão em desenvolvimento na Europa e EUA, podendo elevar a taxa típica de transmissão para 9,6 Kbps (Perkins, 1997).

Existem diversas infra-estruturas de WWAN (*Wireless Wide Area Network*) disponíveis para suportar comunicação de dados. Muitas delas foram desenvolvidas inicialmente para suportar comunicação de voz e depois foram adaptadas para suportar dados. Infra-estruturas para transmissão sem fios apresentam diversas limitações em relação à infra-estrutura convencional. A taxa de erros em *links* de rádio limita o desempenho a 10 Kbps. Esses canais não oferecem estabilidade da qualidade de transmissão, podendo a mesma variar em função de "sombras de rádio" causadas por obstruções tais como edifícios,

acidentes geográficos, chuva, neblina, neve, dentre outros. Assim, a taxa efetiva de transmissão pode cair a 1 Kbps.

Na telefonia celular, os problemas de transmissão de dados tipicamente ocorrem nas fronteiras das células e quando os sinais são comutados de uma célula para outra, instante em que a transmissão pode ser interrompida por cerca de 100 milissegundos (*break-before-make switching*). Os protocolos da camada de enlace dos modems foram adaptados especificamente para poderem lidar com a qualidade variável na transmissão de dados.

Hoje o mercado de transmissão de dados sem fios é dominado por produtos que são inseridos diretamente em microcomputadores pessoais (*desktop*) de forma similar a outros periféricos tipo EISA ou adaptadores externos que utilizam as interfaces serial ou paralela. Além disso, interfaces portáteis com adaptadores do tamanho de um cartão de crédito, conhecidas como PCMCIA (*Personal Computer Memory Card International Association*), tornam-se cada vez mais populares e utilizadas.

Cada terminal móvel celular de assinante (telefone celular) é acoplado a uma antena de rádio que transmite na banda de 800MHz (padrão americano) ou 900MHz (padrão europeu). No Brasil foi instalado um sistema de telefonia móvel analógica, basicamente para transmissão de voz, baseado no padrão europeu. Recentemente, esforços foram feitos para implantação da telefonia celular digital, que já se tornou uma realidade no país, a exemplo do que aconteceu nos Estados Unidos da América e em países europeus, nos quais esforços de padronização de sistemas de telefonia celular digital foram feitos para suportar a transmissão de voz e dados.

Para a implantação da telefonia móvel celular, subdivide-se a área geográfica em pequenas células hexagonais, cada uma dispendo de uma estação de rádio-base e de conjuntos de antenas direcionais para supervisão e controle das suas radiofrequências disponíveis. Estações rádio-base são conectadas ao terminal de controle, e estes são conectados entre si e com a "Rede Nacional e Internacional de Telefonia".

Cada telefone celular, em trânsito por determinada área, quando deseja fazer uma chamada, envia uma mensagem ao rádio-base. Essa mensagem, após ser processada e aceita pelo Terminal de Controle, implica a conexão do telefone celular à estação telefônica celular

através da concessão de uma radiofrequência disponível. Todas as transmissões são *full-duplex*, com um canal de transmissão e outro de recepção.

A passagem de um assinante de um setor para outro, dentro da mesma célula, fica a cargo dos equipamentos de controle internos à célula. Quando o assinante move-se de uma célula para outra, o procedimento é diferente. Ao verificar a apresentação de um nível mínimo de recepção do sinal na célula de origem, o terminal de controle comuta o assinante para uma radiofrequência disponível em uma célula vizinha. A radiofrequência utilizada na célula de origem é desalocada e torna-se disponível para nova alocação. Muitos sistemas celulares esperam em torno de 100 milissegundos, após a liberação da radiofrequência de origem, antes de alocar nova radiofrequência na célula vizinha. Este atraso não causa descontinuidade na comunicação de voz, mas causa problemas na transmissão de dados.

2.5.8 RDSI - Rede Digital de Serviços Integrados

A introdução da RDSIBE (*Rede Digital de Serviços Integrados de Banda Estreita*), que possibilita a comunicação inclusive via Internet a 128Kbps e/ou 2Mbps, já é um avanço em termos de transmissão digital, apesar de estar disponível nos países industrializados desde o início da década de 80 (Soares et al., 1995; Tanenbaun, 1996; Tanenbaun, 1998).

Com o advento, entretanto, da fibra ótica, dos multiplexadores óticos e da tecnologia ATM, a taxa de transmissão já chega a dezenas de Gbps. Levando-se em consideração que a fibra ótica tem como matéria-prima a sílica, que é muito mais abundante em nosso planeta do que o cobre, é evidente que a fibra ótica deverá se tornar muito mais barata do que os atuais cabos de pares de cobre usados pelas concessionárias de serviços públicos de telefonia, das centrais telefônicas até o terminal do cliente. Além disto inúmeras vantagens técnicas levaram os países industrializados a utilizar fibra ótica até o terminal do cliente e foi criada a chamada RDSIBL (*Rede Digital de Serviços Integrados de Banda Larga*).

A partir do ano 2000 a comunicação multimídia (voz, dados em alta velocidade, áudio, vídeo em alta definição, gráficos, telemedicina, dentre outros) em banda larga estará disponível para um vasto número de clientes, e a previsão é as telecomunicações interativas e vídeo 3D sejam realidade até o ano de 2010.

No Brasil, a RDSIBE ainda é uma tecnologia em consolidação e a utilização extensiva de

RDSIBL ainda é um objetivo a ser atingido. Neste ponto, ao invés de realizar investimentos na instalação de cabos óticos que alcancem o cliente, apenas para a transmissão de sinais de TV a cabo e persistir nas baixas taxas de transmissão na Internet, principalmente nos enlaces interestaduais e internacionais, pode-se investir em infra-estrutura que acompanhe o desenvolvimento das técnicas de telecomunicações.

Tradicionalmente, sistemas de comunicação foram desenvolvidos para o transporte de tipos específicos de informação - o sistema telefônico para o tráfego de voz, as redes de comunicação de pacotes para dados textuais, vídeo e televisão em redes de radiodifusão ou a cabo. Essas redes foram claramente projetadas para aplicações específicas, o que gera problemas de adaptação a outros tipos de serviço. O ideal de uma única rede capaz de atender a todos esses serviços, de forma a obter uma economia devido ao compartilhamento dos recursos, veio a motivar o conceito das Redes de Serviço Integrado.

Pela primeira vez, em 1972, o CCITT (atual ITU-T), emitiu, em sua recomendação G.702, a seguinte definição para essa nova rede: “uma rede digital integrada, na qual os mesmos comutadores e caminhos digitais são usados para os diferentes serviços, por exemplo, telefonia e dados”.

Nos períodos subseqüentes de estudo, o CCITT continuou a elaborar as especificações sobre a RDSI, que culminaram em 1984 com as recomendações da Série I do Livro Vermelho. Surgiu então a seguinte definição de RDSI: “uma rede, em geral evoluída da rede digital de telefonia, que proporciona conectividade digital fim-a-fim, para suportar uma variedade de serviços vocais e não vocais, aos quais os usuários têm acesso a partir de um conjunto limitado de interfaces padronizadas.

O investimento no sistema telefônico existente é alto, exigindo que a RDSI seja concretizada por etapas ao longo de décadas, fazendo com que a RDSI coexista com o sistema analógico atual por muitos anos. Esses requisitos têm uma influência importante sobre a forma final que a RDSI assumirá e sobre a forma como o sistema corrente evoluirá gradativamente em direção à RDSI.

Um fato importante é o crescimento das redes comerciais comutadas por pacotes, crescimento esse observado desde meados dos anos setenta. Outro fator importante reside na

decisão de aproveitar a base telefônica instalada, considerando-se que é inviável substituir uma vasta malha de cabeamento por fibra ótica somente para satisfazer os requisitos da RDSI. Por outro lado, substituir troncos entre centrais por fibra ótica é factível, porque são em menor número e podem ser atualizados com maior facilidade.

Esses fatos indicam que a RDSI foi projetada, desde o início, para conviver com as limitações existentes das malhas de assinantes e das redes de voz, além das redes de comutação de pacotes.

Um novo conjunto de interfaces, que utiliza a tecnologia ADSI (*Linha de Assinantes Assimétrica Digital*) sobre linhas comuns de telefone, promete abrir canais de até 6312 Kbps. A velocidade de envio dos dados seria de até quatro vezes maior que uma linha TI e cerca de 400 vezes mais veloz que um modem de 14.4 Kbps.

Outra tecnologia que pode ser combinada com a ADSL é o HDSL, a linha digital de assinante de alta taxa de *bits*. O ADSL utiliza um fio de par trançado de mais de 5 Km. O HDSL, por sua vez, usa dois pares completos de par trançado de quase 4 Km.

A linha de assinante digital simétrica (SDSL), segundo alguns analistas, pode até ser mais atraente para alguns usuários. O SDSL permite duas larguras de banda em sentido bidirecional, o que viabiliza, por exemplo, videoconferências ao vivo em ambientes domésticos.

2.5.9 Computação e tecnologias móveis

A computação móvel é o último estágio do desenvolvimento da computação pessoal. Em 1946, a Illinois Bell Telephone Company introduziu um serviço de telefonia móvel, que permitia a usuários que estivessem conduzindo veículos, comunicar-se com o sistema telefônico. Foi a primeira iniciativa de permitir comunicação bidirecional sem fio. Atualmente, observa-se a expansão e a popularização do telefone celular, que garante mobilidade aos usuários do sistema telefônico (Bates & Gregory, 1997).

São muitos os argumentos que formam a base para o desenvolvimento de um ambiente de computação móvel. Inicialmente, deve-se pensar que nem todas as pessoas que necessitam do computador estão presentes em um determinado local físico, tal como um escritório, uma sala

de aula, dentre outros. Vendedores são exemplos de trabalhadores com alta mobilidade que necessitam acessar bases de dados remotas e executarem operações diversas, tais como a emissão de pedidos, requisição de mercadorias, dentre outras.

É possível relacionar vantagens de um sistema móvel, tais como conforto para utilização em qualquer ambiente, flexibilidade para utilização em diversas aplicações que exijam movimento, disponibilidade independente da localização do usuário. Além disso, algumas exigências também podem ser relacionadas, tais como a portabilidade para facilitar o transporte, autonomia de energia para garantir o funcionamento onde não existe disponibilidade de energia e desempenho comparável a estações fixas.

A exemplo dos computadores fixos, deseja-se garantir características de comunicação aos computadores móveis, de uma forma sem fio (*wireless*). Esta comunicação deve ser feita através de sistema de ondas de rádio, utilizando antenas para transmissão e recepção. Pela portabilidade do sistema, espera-se que as antenas não sejam grandes e nem mesmo muito potentes, para não ferir a autonomia de energia do sistema.

Um ambiente de computação móvel envolve computadores portáteis interligados em rede através de sistema de ondas de rádio, no qual denomina-se unidade móvel (UM) o elemento de rede (computador portátil conectado à rede de computação móvel). As unidades móveis são agrupadas e ligadas em rede, divididas em subredes, cada qual mantida por um grupo de antenas, as quais são chamadas ponto de acesso (PA). Um ambiente de computação móvel pode ser desenvolvido de forma independente de infra-estrutura fixa, caracterizando as redes *ad-hoc*, que são interligadas diretamente através das antenas.

A mobilidade porém, sempre implica algumas condições, típicas do ambiente, que devem ser consideradas, independente do sistema de acesso (Perkins, 1997):

- capacidade de comunicação limitada com largura de banda variável e alta taxa de erros;
- autonomia de energia limitada por baterias com limite de consumo, de forma que deve-se despender o mínimo de energia com processamento e dispositivos de apoio ao sistema;
- limites físicos de hardware para garantia de portabilidade, limitando também o poder de processamento e dispositivos;

- problemas de roteamento de pacotes, quando há variação da subrede onde está presente a UM;
- perda temporária de comunicação, quando do deslocamento entre áreas mantidas por diferentes estações de rádio e renegociação de características de acesso.

Para contornar estes problemas, diversas soluções têm sido propostas, algumas das quais já em operação, outras em pesquisa.

A princípio, pode-se utilizar o próprio sistema de telefonia celular para fazer a interconexão destes computadores em rede, porém, algumas limitações estão presentes neste tipo de acesso, especialmente no que diz respeito à baixa velocidade de acesso obtida neste sistema. Para este sistema, a unidade móvel é um computador portátil que utiliza um modem e um telefone celular para se conectar a um servidor de acesso remoto. As estações de rádio são as estações dos sistemas de telefonia celular e todas as operações de mobilidade são gerenciadas pela camada física deste sistema, provida pelo sistema de telefonia celular.

Outras formas de comunicação podem também ser usadas, como é o caso da Wavelan, que permite a utilização de taxas de transmissão mais satisfatórias, entre 1 e 2Mbps. Este sistema deve resolver problemas de roteamento, que são resolvidos naturalmente pelo sistema celular.

Como ponto importante, após o estabelecimento de mecanismos nas camadas inferiores que possibilitam o uso de computação móvel, deve-se observar as necessidades de adaptação para as camadas superiores para permitir o seu uso de forma mais ampla, caracterizando e permitindo o uso de computação móvel na Internet (a Internet móvel).

Espera-se que uma unidade móvel seja reconhecida na Internet independente da subrede em que ela se encontra. Sabe-se, porém, que o endereço de uma estação depende de sua subrede, o que exige modificações de configuração para que o acesso mantenha-se transparente ao usuário.

Para servidores móveis o problema é mais crítico ainda, pois deve ser reconhecido na Internet com um endereço fixo, independente da subrede em que se encontra. Espera-se,

ainda, que a estação fixa em comunicação com a unidade móvel não observe diferenças entre esta unidade e outra estação qualquer.

Uma unidade móvel deve ser capaz de acessar todos os recursos disponíveis em uma estação ligada a uma rede através de cabeamento, contornando todos os problemas deste ambiente. Lança-se mão, então, de técnicas para melhor aproveitamento do canal de comunicação, da energia do sistema e do processamento, de forma a limitar as tarefas executadas na unidade móvel.

Deve-se adaptar a aplicação, o protocolo utilizado ou mesmo a pilha de protocolos utilizados no ambiente de computação móvel para garantir os mesmos parâmetros de QoS que seriam garantidos em uma aplicação semelhante em uma rede fixa. Pode-se alterar a pilha de protocolos entre a unidade móvel e a estação fixa, de forma a criar uma camada intermediária que agregue as características necessárias, como por exemplo em requisições feitas ao protocolo HTTP (*HyperText Transfer Protocol*). Assim, uma camada *middleware* (agente) pode ser responsável por fazer uma adaptação do protocolo HTTP para o meio, de forma a economizar a banda e, possivelmente, o processamento. O resultado pode ser transparente para o usuário ou não, dependendo dos aspectos de contexto desse mesmo usuário (*awareness*). Neste caso, uma abordagem poderia ser a realização de um tratamento prévio sobre serviços oferecidos ao usuário, como por exemplo, as figuras poderiam ser inicialmente transmitidas e mostradas em preto-e-branco e, somente as figuras necessárias e requisitadas sob a ação do usuário seriam então transmitidas em cores.

Ainda no contexto da mobilidade, ao vislumbrar o serviço de telefonia IP (*Internet Protocol*), deseja-se adaptá-lo de forma a possibilitar seu uso em um ambiente de computação móvel. Neste ponto, o uso de diversos algoritmos de codificação de voz influi na largura de banda necessária para o serviço e na qualidade da reprodução da voz no outro extremo da conversação, de forma a ter possivelmente uma reprodução de voz mais mecanizada, com perda de alguns sons da fala.

A adaptabilidade a ser trabalhada para telefonia IP pode ser feita sobre estes algoritmos de codificação e decodificação, definidos pela ITU-T, além da possibilidade de compactação e supressão de silêncio. Assim, a estação fixa pode alterar dinamicamente estes métodos, atuando diretamente na voz, de acordo com a disponibilidade de largura de banda.

As tecnologias que permitem a implementação de sistemas móveis abrem uma perspectiva ampla para o desenvolvimento de aplicações com requisitos de consciência de contexto, especialmente em termos da interação do usuário com ambientes nos quais essas aplicações estiverem inseridas.

2.5.10 Teleconferência e videoconferência

Diversas situações de trabalho exigem que as pessoas estejam geograficamente dispersas apesar de formarem uma equipe. Isto é muito comum em grandes empresas, como as multinacionais, por exemplo. Nestes casos, a realização de reuniões para discussão de problemas pode se tornar um fator de alto custo. O transporte de pessoal, a acomodação, enfim, toda a infra-estrutura que precisa ser criada para realizar os encontros acarreta altos gastos. Além dos gastos diretos, o desperdício de tempo com o transporte para locais distantes também representa uma perda significativa. Na tentativa de melhorar este quadro, vários grupos de pesquisa direcionaram seus enfoques sobre o conjunto de serviços denominado, genericamente, teleconferência (Lemair & Shae, 1997).

As aplicações atuais para as teleconferências incluem principalmente as reuniões de negócios, os grupos de discussão e a educação a distância.

O ambiente de teleconferência é um ambiente multimídia, que torna possível o compartilhamento de diversos espaços acústicos e visuais. Nas teleconferências, os diversos participantes são capazes de compartilhar mídias estáticas como textos, diagramas e figuras e também mídias dinâmicas como áudio e vídeo. Logicamente, o fato de se utilizar canais de comunicação e equipamentos de aquisição de dados, que são limitados, cria um fator de limitação na qualidade e na interatividade das conferências (Galbreath, 1995).

Existem diversos serviços normalizados para teleconferência. Uma teleconferência pode ser definida como um conjunto de facilidades de comunicação através de meios eletrônicos para possibilitar comunicações bidirecionais entre dois ou mais usuários ou grupos de usuários. Para ser interoperável, o sistema deve seguir os padrões estabelecidos para transmissão das diversas mídias. Os organismos de padronização internacional (ISO e ITU-T) indicam a utilização da norma H.200, definindo diferentes níveis de teleconferência:

- *audioconferência* – serviço em que somente sinais de áudio e controle são transmitidos entre os participantes;

- *conferência áudio-documentacional* - similar à sessão de audioconferência, havendo entretanto o tratamento de documentos textuais;
- *conferência audiográfica* - serviço com suporte à transmissão de áudio, sinais de controle, documentos e imagens estáticas;
- *freeze-frame videoconferência* - serviço similar à conferência audiográfica acrescida do envio periódico de imagens estáticas aos participantes;
- *teleseminário* - serviço que consiste da distribuição dos eventos ocorridos num local (áudio e vídeo) para todos os demais participantes, sendo o áudio o único sinal de retorno;
- *videoconferência* - serviço similar à conferência audiográfica acrescida do envio *on-line* e em tempo real de sinais de vídeo entre os vários participantes.

Em linhas gerais, uma aplicação deve utilizar um conjunto mínimo de serviços de teleconferência. Um serviço de videoconferência que queira facilitar a realização de reuniões deve, entretanto, incluir outras ferramentas. Tais facilidades devem incluir todo um sistema denominado de pré-conferência, que deve permitir o agendamento da conferência e sua divulgação aos seus componentes. Além disto, na pré-conferência deve-se configurar o ambiente para a sessão a ser realizada.

Já durante uma sessão de videoconferência, os participantes devem se sentir o mais próximo possível de uma conferência clássica. Para isso eles devem dispor, por exemplo, de ferramentas que permitam a fácil e eficiente manipulação dos documentos compartilhados. Os usuários devem também ter oportunidade de comunicação, tanto em modo privado como em grupos. Diferentes sistemas de votação também podem ser implementados para facilitar o andamento de reuniões. Um ponto de extrema importância em uma videoconferência é sua privacidade. Deste modo, mecanismos robustos de segurança devem proteger as conferências de espionagens, impedindo a entrada de componentes não desejados. Estes itens são essenciais para uma boa aceitação dos sistemas de videoconferência pelos usuários. Adicionalmente, deve haver um limite de tempo na utilização do canal de voz por cada um dos participantes, por exemplo. Este controle de acesso pode ser realizado por um coordenador ou de modo automático pelo ambiente, o que caracteriza o uso da consciência de contexto.

Uma das maiores preocupações nos sistemas de videoconferência é a inundação de redes com o volume de dados gerado (voz e, principalmente, vídeo). Mecanismos de controle de fluxo devem ser implementados para evitar tais inundações. Os principais elementos de um sistema de videoconferência incluem (Bates & Gregory, 1997):

- **participante:** usuário da conferência com direitos, controlados pelo coordenador, à fala e demais recursos da conferência;
- **organizador:** indivíduo que tem como tarefa agendar a conferência e, se necessário, divulgar aos participantes a existência da conferência (pode ser um participante ou não);
- **coordenador:** participante com direitos especiais sobre todo o controle da conferência (uma conferência pode ser realizada sem a presença deste indivíduo, quando o controle de acessos for realizado pelo próprio sistema);
- **interlocutor:** participante que detém, em um dado instante, o direito à fala e à alteração dos documentos multimídia/hipermídia (o direito de alteração de documentos pode ser delegado a um secretário);
- **secretário:** usuário da conferência para quem se delega o direito de escrita nos documentos multimídia/hipermídia da base compartilhada (pode ser um participante ou não);
- **assento:** dispositivo lógico que pode ser preenchido por um participante ou secretário;
- **base privada:** sessão de trabalho de um usuário, de acesso e controle restrito a este usuário;
- **hiperbase ou hiperbase pública:** depósito de documentos persistentes de acesso a todos os usuários da conferência, de acordo com seus direitos;
- **base compartilhada:** depósito volátil de documentos que possibilita o trabalho cooperativo entre os participantes da conferência (é visível por todos os participantes, mas com controle de alteração realizado pelo sistema).

Observa-se, aproveitando-se dos conceitos de agentes apresentados no **Capítulo 4**, que os elementos de um sistema de videoconferência podem ser mapeados em agentes computacionais habilitados para a realização das tarefas de cada elemento e também para tarefas de cooperação, sendo essa implementação relevante para o cenário dos ambientes com consciência de contexto.

Ao nível tecnológico, as primeiras soluções de videoconferência apareceram com um contexto de controle centralizado, tendo sido uma solução natural para as conferências baseadas em comutações de circuitos, como ocorre com as linhas telefônicas públicas e na N-ISDN, por exemplo.

O crescimento da Internet vem atraindo os mais diversos tipos de aplicação, que tentam explorar ao máximo o potencial da rede mundial. Como não podia deixar de ser, os sistemas de videoconferência também passaram a se integrar neste universo. Os sistemas de videoconferência em redes de comutação de pacotes apareceram como solução para o desperdício de recursos que ocorrem nos sistemas de comutação de circuitos. Uma das soluções, adotada para otimizar o uso da rede, foi a utilização de sistemas distribuídos.

Numa arquitetura distribuída, as estações dos usuários se responsabilizam pelo tratamento dos dados. A confiabilidade do sistema é muito melhor do que na arquitetura centralizada. Os usuários se conectam em uma estrutura *multicast* que os interconecta, permitindo a troca de informações entre eles. Uma das maiores dificuldades dos sistemas distribuídos é a garantia da qualidade de serviço (QoS), causada, principalmente, pelo congestionamento nos canais de comunicação, e que gera, como consequência, sérios problemas de sincronismo e descontinuidade nas mídias de áudio e vídeo.

Para superar problemas de sincronismo, algumas técnicas são utilizadas. Inicialmente, as fontes de áudio e vídeo são tratadas independentemente. Na geração de cada mídia, existe um esquema de marcação dos pacotes com "*time stamps*" que permitirão a recuperação do sincronismo no destino. Ao serem reproduzidas, as mídias são re-sincronizadas, tendo como mestre o sinal de áudio. Desta forma, se for necessário, o sinal de vídeo descarta alguns quadros se o sinal de áudio estiver um pouco atrasado. Da mesma forma o sinal de vídeo pode repetir alguns quadros antigos se o sinal de áudio estiver adiantado em relação a ele.

Outra preocupação é a implementação de mecanismos de controle de congestionamento da rede multiponto utilizada. Deve-se lembrar que o tráfego de mídias de áudio e vídeo ocupa uma porcentagem alta da banda passante disponível. No caso da Internet, o backbone multiponto disponível, o MBONE, funciona com o protocolo UDP (*User Datagram Protocol*), que não apresenta mecanismos de controle de fluxo e de congestionamento. Desta forma este controle deve ser realizado em camadas superiores ao UDP. As soluções procuram sempre

diminuir as perdas de dados decorrentes do congestionamento da rede. Em geral técnicas de estimação do estado atual das redes são empregadas para a obtenção de suas capacidades instantâneas de transmissão.

Mesmo com a implementação de mecanismos de controle de congestionamento e de fluxo, a ocorrência de perdas em alguns instantes é praticamente inevitável. Um bom sistema de videoconferência deve ser o mais imune possível a essas perdas. Desta forma, os participantes de uma conferência devem notar o mínimo possível de degradação de serviços. Para tanto, é interessante a utilização de codificadores e decodificadores (CODECS) robustos a perdas.

Em geral, a videoconferência tradicional requer interconexão especial através do telefone com alta taxa de largura de banda. Atualmente estão sendo utilizadas redes ISDN e ATM, e com os avanços da tecnologia, proporcionando processadores mais rápidos e melhor esquemas de compressão de dados, um novo tipo de videoconferência, a conferência *desktop*, tornou-se viável. Ao contrário das videoconferências em salas especiais, exigindo equipamentos especiais e caros, a videoconferência em *desktop* pode ser realizada através da inclusão de software e hardware em computadores originariamente para uso pessoal.

2.6 Os padrões MPEG-4 e MPEG-7

O crescente número de aplicações multimídia torna complexa a representação dos dados audiovisuais, uma vez que a maioria das aplicações utiliza multimídia como característica comum para interatividade com usuário. As aplicações impõem conjuntos de especificações que variam muito de uma aplicação para outra. A diversidade de aplicações implica em diferentes conjuntos de especificações, sendo que cada aplicação pode ser caracterizada pelo de dado a ser processado (vídeo, imagem, texto), natureza do dado (natural, sintética, médica, gráfica), taxa de bits (baixa, média e alta), atraso admissível máximo, tipo de comunicação (ponto-a-ponto, multiponto), e por um conjunto de funcionalidades oferecidas (escalabilidade, manipulação de objetos, edição).

Tanto a indústria eletrônica como as operadoras de TV a cabo, companhias de telecomunicações e empresas de software e hardware têm mostrado um interesse crescente em multimídia. Esta tendência tem acelerado o crescimento da utilização de CD-ROMs, do ambiente da WWW e, mais recentemente, do DVD. Assim, os padrões que atualmente

existem para multimídia não conseguem atender adequadamente às novas expectativas e requisitos do usuário e das novas e complexas aplicações (Ebrahimi & Horne, 2000). Dentro deste contexto, dois novos grupos de trabalho MPEG foram criados para desenvolver padrões com o objetivo de atender os requisitos das aplicações multimídia atuais e futuras, sendo denominados MPEG-4 (ISO, 1997b) e MPEG-7 (ISO, 1997c).

Conforme apresentado, diversas tecnologias podem ser utilizadas para a construção de aplicações multimídia interativas. No contexto do grupo MPEG, o padrão MPEG-2 é o mais indicado para a codificação de imagens, porém, à medida que os requisitos de interatividade estão se tornando mais complexos, a demanda por tecnologias mais adequadas tem crescido (ISO, 1997d; ISO, 1997e).

O grupo MPEG, desde 1980, tem trabalhado na padronização de informação audiovisual (áudio e vídeo), tendo como resultado dois padrões mundialmente conhecidos: o MPEG-1 (IS-11172) e MPEG-2 (IS-13818). O primeiro especifica o armazenamento de áudio e vídeo a taxas de 1,5Mbps; o segundo manipula a codificação genérica de TV digital e sinais de HDTV (*High Definition Television*). Estes padrões têm, ao longo do tempo, gerado forte impacto na indústria eletrônica.

Em linhas gerais, o padrão MPEG-4 visa atender as áreas de televisão digital, aplicações gráficas interativas e WWW, além de fornecer requisitos para integrar a produção, distribuição e acesso ao conteúdo baseado em informação audiovisual. O padrão MPEG-7 tem como nome formal "Interface de Descrição do Conteúdo Multimídia" e oferece um conjunto padrão de descritores e esquemas de descrição usados para especificar o conteúdo da informação, com a finalidade de tornar a busca da informação multimídia mais rápida e eficiente.

O padrão MPEG-4 é o sucessor natural do MPEG-2, tendo sido desenvolvido para ser um padrão multimídia e não somente audiovisual (como o MPEG-2), e já existem estudos para o encapsulamento de *streams* MPEG-4 em MPEG-2, o que é utilizado pelos padrões da TV Digital.

O padrão MPEG-7, ao contrário dos demais padrões MPEG, não é relacionado à codificação de áudio e vídeo; sua função é criar um padrão para a descrição de uma cena audiovisual,

uma imagem ou sons, que não precisam estar armazenados digitalmente. O MPEG-7 pretende disponibilizar ferramentas (nesse caso, as ferramentas são os descritores, que permitem a criação das descrições) para que a busca em imagens, vídeos e arquivos sonoros seja tão fácil quanto é a busca em textos.

As **Subseções 2.6.1 e 2.6.2** descrevem as principais características destes dois padrões da família MPEG.

2.6.1 O padrão MPEG-4

O padrão MPEG-4 tem como objetivo construir uma cena audiovisual, tratando os dados como objetos de mídia. Esses objetos podem ser vídeo, áudio, objetos sintéticos (imagens e áudio gerados por computador) e objetos Web (documentos na WWW), além de sua relação com o tempo e o espaço.

As informações dos objetos, que serão especificados posteriormente, são organizadas em árvores. Outro aspecto do MPEG-4 é permitir a integração da produção, distribuição e do acesso a conteúdo (ISO, 2001a; ISO, 2001b; Battista et al., 1999). A primeira versão MPEG-4 atingiu o nível de padrão internacional em abril de 1999 (ISO, 1999a) e a segunda versão tornou-se padrão no início de 2001 (ISO, 2001a), porém, algumas extensões ainda estão em progresso.

Uma cena audiovisual MPEG-4 é descrita como a composição de objetos de mídia primitivos. Um objeto de mídia primitivo é, por exemplo, um vídeo da face de uma pessoa ou um áudio de uma voz, e correspondem às folhas em uma árvore descritiva. Já um objeto composto associa esses objetos em subárvores, de modo que se pode ter, dentro de um objeto composto, um vídeo de um rosto e o áudio relacionado a uma voz, facilitando a manipulação de vários objetos relacionados. Dentre as suas características, o MPEG-4 permite (ISO, 2001a):

- colocar objetos de mídia em qualquer posição espacial, dadas as suas coordenadas;
- aplicar transformações a objetos, mudando sua geometria ou acústica;
- criar objetos compostos, agrupando objetos primitivos;
- mudar interativamente a visualização e o som de cenas.

Assim como o MPEG-2, o MPEG-4 possui diversos *profiles*, sendo 8 para vídeo natural, 7 para vídeos sintéticos e híbridos (sintéticos + naturais), 8 para áudio, 3 para gráficos, 4 para

descritores de cena, 2 para o MPEG-J e um para a descrição de objetos (ISO, 2001a). Além destes, outros *profiles* estão sendo estudados (ISO, 2000b).

Algumas das aplicações para as quais o MPEG-4 foi desenvolvido são (ISO, 2001a; Pereira & Ebrahimi, 2002):

- *broadcast*: o MPEG-4, pelo fato de suportar uma compressão de dados muito alta, permite um uso muito bom da largura de banda disponível; também, por esse motivo, permite que diversos *streams* de vídeo sejam transmitidos em paralelo; o padrão MPEG-4 trata o vídeo como um conjunto de objetos inter-relacionados e essa característica é bastante interessante para a televisão interativa, uma vez que pode-se associar diferentes objetos a diferentes ações; o ISDB (TV Digital japonesa) prevê o uso do MPEG-4 em sua implementação;
- *visualização de cenas de modo colaborativo*: o MPEG-4 é bastante apropriado para a organização de cenas envolvendo objetos reais e virtuais, que devem ser transmitidos para participantes dispersos; isso faz do MPEG-4 uma poderosa ferramenta para a construção de aplicações de trabalho colaborativo, aplicações de realidade virtual e de realidade aumentada (*augmented reality*);
- *armazenamento e recuperação baseada em conteúdo*: pelo fato do MPEG-4 permitir uma descrição dos objetos, a busca por palavras-chave fica bastante facilitada. Essa característica é tão importante que o uso adequado de metadados (comumente chamados de dados sobre os dados) é o objetivo principal do MPEG-7, que será descrito na **Seção 2.6.2**;
- *set-top box da Televisão Digital*: a TV Digital, como foi dito anteriormente, irá mudar o modo das pessoas assistirem televisão, uma vez que ela possuirá um ambiente mais interativo; algumas das características que o MPEG-4 possui, que podem ser bastante interessantes para a TV Digital são: a) criação de *links* entre os comerciais apresentados e suas páginas na Internet; b) acesso à informação na Internet; c) jogos interativos e *video on demand*;
- *jogos*: é inegável a popularidade dos jogos de computador nos dias de hoje. A maioria dos jogos utiliza gráficos 3D tanto para o ambiente quanto para os objetos controlados pelo usuário; o uso de MPEG-4 nos jogos abre novas possibilidades com integração dos objetos sintéticos com os naturais e a animação facial;
- *computação móvel*: com a popularização dos *Personal Digital Assistants* (PDAs) e dos telefones celulares, fica evidente a importância da computação móvel nos próximos

anos; o MPEG-4, pelo fato de conseguir transmitir vídeo em redes com pequena largura de banda, mostra-se uma boa solução para a futura transmissão de vídeos para dispositivos móveis;

- produção de TV: o MPEG-4 apresenta-se como uma interessante evolução em relação ao *chroma keying* (técnica da TV analógica com a qual é possível colocar um vídeo sobre uma área de outro que possui uma cor bem definida, normalmente verde ou azul), permitindo a gravação dos objetos separadamente e com diferentes níveis de qualidade.

O padrão MPEG-4 é dividido em sete partes: Sistemas, Visual, Áudio, Teste de Conformidade, Software de Referência, DMIF e Software Otimizado para as ferramentas MPEG-7 (codificadores de vídeo), que serão apresentadas a seguir.

2.6.1.1 Sistemas (Systems)

A parte *Systems* do padrão define a descrição do relacionamento entre os componentes audiovisuais que formam a cena. Esse relacionamento é descrito em dois níveis principais.

O primeiro descritor é o BIFS (*Binary Format for Scenes*) (ISO, 2001a), que descreve o relacionamento espacial e temporal entre os componentes da cena. O BIFS é baseado no conceito do VRML tanto na estrutura quanto na funcionalidade da composição de nós de objetos. O outro descritor é o OD (*Object Descriptor*), que define o relacionamento entre os *Elementary Streams* de cada objeto da cena.

Outros aspectos relacionados à parte *Systems* do MPEG-4 são a interação, a multiplexação, o armazenamento, interfaces com redes e outros equipamentos, representação textual, sincronização e direitos autorais. A interação com e entre os objetos é possível no MPEG-4 de duas formas.

O primeiro tipo de interação é o previsto pelo padrão, como o posicionamento espacial e a retirada de objetos da cena. Para que seja possível a interação não prevista pelo MPEG-4, com e entre os objetos, a API MPEG-J, que é derivada da linguagem Java (Cornell, 1997; SUN, 1999; SUN, 2000), pode ser amplamente utilizada. MPEG-J é um sistema de controle flexível e programável que representa uma sessão audiovisual de modo a permitir que a sessão se adapte às características do terminal onde ela será apresentada. MPEG-J possui

duas importantes características: a) a capacidade de permitir uma pequena degradação quando os recursos são limitados ou variam de acordo com o tempo; b) a habilidade de responder à interação do usuário e prover funcionalidades multimídia (ISO, 1999d). Na **Seção 5.5**, o MPEG-4 Systems é descrito em detalhes e suas relações com esta proposta são discutidas; o mesmo ocorre com a API MPEG-J, na **Seção 5.6**.

2.6.1.2 Visual

O padrão de vídeo natural do MPEG-4 (ISO, 1998a; Pereira & Ebrahimi, 2002) consiste de uma coleção de ferramentas para as áreas de *shape coding* (codificação de forma), estimativa e compensação de movimento, codificação de textura, recuperação de erro, *sprite coding* e escalabilidade.

O *shape coding* pode ser feito de duas maneiras: a) em modo binário, onde a forma de cada objeto é dada por uma máscara binária; b) pelo modo de *grayscale* (tons de cinza), onde a forma do objeto é dada de modo similar ao dos canais alpha, permitindo transparência e diminuindo o *aliasing* (diferença nas bordas das figuras). A estimativa e a compensação de movimento é baseada em blocos, que podem ser de 8X8 ou 16X16 pixels. A codificação da textura é baseada no 8X8 DCT (*Discrete Cosine Transform*). Com algumas modificações, a codificação de texturas estáticas pode ser feita utilizando transformadas *wavelet* (Daubechies, 1990).

As mais importantes características do padrão de codificação visual do MPEG-4 são (Pereira & Ebrahimi, 2002; Battista et al., 1999):

- eficiência na compressão: a compressão eficiente foi um dos principais objetivos tanto do MPEG-1 quanto do MPEG-2. O MPEG-4 pretende, a partir de uma codificação melhorada e da codificação de múltiplos *streams* de mídia concorrentes, melhorar ainda mais essa compressão e, com isso, aumentar o número de aplicações e produtos que utilizam esse padrão;
- interação baseada em conteúdo: codificando e representando objetos de vídeo ao invés de quadros de vídeo, o MPEG-4 permite a criação de aplicações baseadas em conteúdo;
- acesso universal: as duas últimas funcionalidades do MPEG-4 são a robustez e a escalabilidade permitindo, assim, a transmissão de vídeos, de acordo com a largura de banda disponível; O padrão definiu três faixas de *bitrates*: abaixo de 64 Kb/s, entre

64 e 384 Kb/s e entre 384 e 4 Mb/s, cobrindo, assim, as necessidades de diferentes aplicações.

Uma cena visual do MPEG-4 é composta por um ou mais objetos de vídeo. Cada objeto de vídeo possui informação temporal e espacial através de sua forma, movimento e textura.

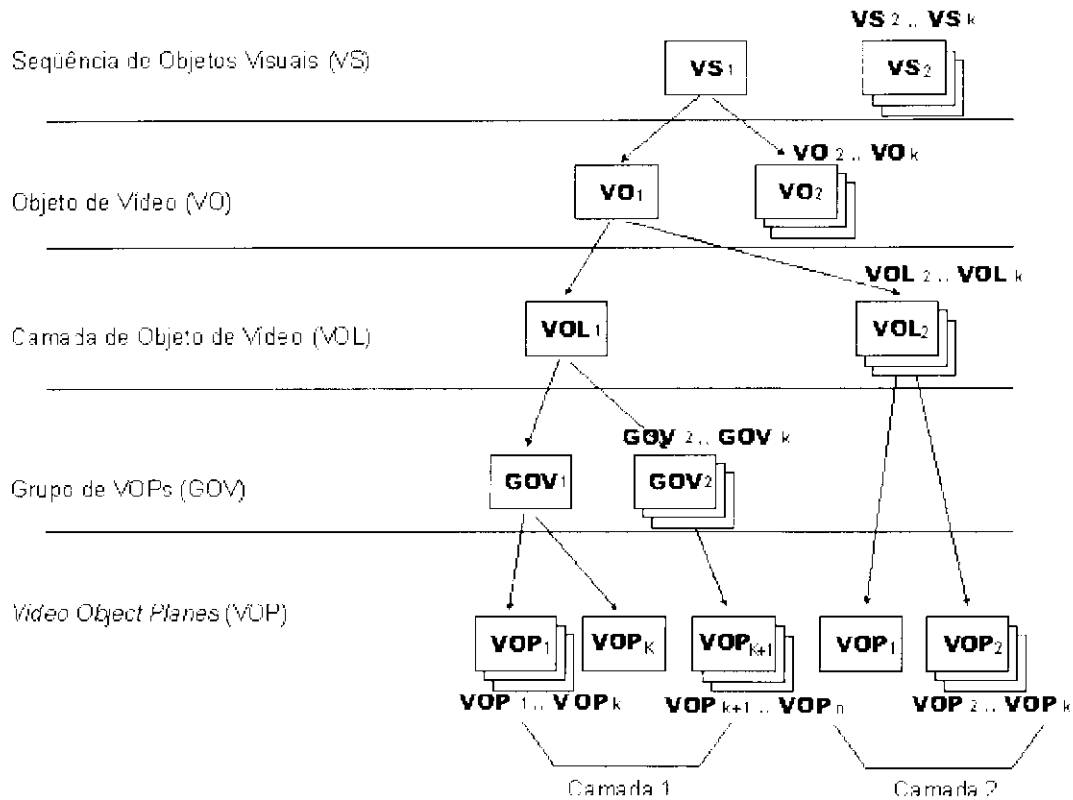


Figura 2.10 – Exemplo da estrutura de um *bitstream* MPEG-4 (Pereira & Ebrahimi, 2002)

Um *bitstream* MPEG-4 tem uma divisão hierárquica (Figura 2.10), sendo dividido em:

- Sequência de Objetos Visuais (VS): a cena completa, que pode conter objetos 2D e 3D;
- Objeto de Vídeo (VO): um objeto de vídeo é necessariamente 2D, e ele pode ter um formato retangular ou, utilizando máscaras, irregular;
- Camada do Objeto de Vídeo (VOL): um objeto de vídeo pode ser composto por diversas camadas, por exemplo, com diferentes níveis de compressão e quantização.

Cada objeto de vídeo é dividido em *frames*, chamados *Video Object Plane* (VOP), que podem ser agrupados em *Groups of Video Object Planes* (GOV).

2.6.1.3 Audio

A parte de áudio do MPEG-4 é referente tanto à codificação de áudio, com diversos *Profiles* e *Levels*, quanto à apresentação de voz sintética, incluindo a sincronização labial e a informação sobre os fonemas.

O áudio MPEG-4 permite o desenvolvimento de uma variedade de aplicações, que vão desde a conversão automática entre fala e texto (*text-to-speech*) até a definição de multicanais e sons tridimensionais. O áudio, assim como ocorre com os objetos visuais, é representado como um objeto de mídia e suas propriedades podem ser manipuladas pelas interfaces definidas pelo MPEG-4 *Systems*. Algumas possibilidades de codificação estão presentes na especificação MPEG-4 Audio.

A primeira possibilidade é a codificação de **sinais de áudio genéricos**. Neste caso, o objetivo é a codificação de sinais de áudio com alta compressão e destinados a compor *streams de mídia* para distribuição em canais com baixa largura de banda. A cobertura, neste caso, inicia-se aos 6 kbits/s num espectro de frequência de 4KHz. Parâmetros de codificação e a representação orientada a objetos permitem a manipulação de características do sinal de áudio e sua adequação a outros níveis de QoS.

A segunda possibilidade é a codificação de **sinais de áudio baseados na fala**. Neste caso, a codificação é voltada a aplicações em que o áudio deve ser capturado e reproduzido em tempo real. A codificação é gerada para transmissão a taxas que variam de 2 kbits/s a 24 kbits/s. A codificação é baseada em algoritmos HVXC, permitindo todos os controles de interação do usuário. É possível, ainda, durante a codificação, gerar ajustes de velocidade e produzir efeitos nos sinais de áudio.

A terceira possibilidade é a codificação de **sinais de áudio sintéticos**. O áudio MPEG-4 é estruturado de forma a permitir também a descrição de sons de instrumentos musicais, assim como ocorre com o formato MIDI. Neste caso, vale observar que os algoritmos não são baseados diretamente nos sons dos instrumentos musicais, mas em descrições matemáticas que permitem a codificação de escalas de notas musicais.

Por fim, é possível ainda a codificação de **sinais escaláveis de áudio baseados na fala**. O MPEG-4 Audio possui codificadores TTS que operam em taxas que variam de 200bits/s a 1,2 Kbits/s, possibilitando a codificação de fonemas e parâmetros de descrição da fala.

2.6.1.4 Framework para a Integração da Entrega Multimídia

Outro componente importante do MPEG-4 é o DMIF (*Delivery Multimedia Integration Framework*) (ISO, 2001a). O DMIF é um protocolo para o gerenciamento de *streams* multimídia sobre tecnologias de transporte. Ele é semelhante ao FTP, sendo que a principal diferença é que enquanto o FTP retorna dados, o DMIF retorna ponteiros para onde obter os *streams* de dados, ajustando a QoS do objeto às condições da rede (ISO, 2001a).

Uma vez que os objetos de mídia podem ser entregues separadamente, deve haver um mecanismo para integrar estes dados e sincronizá-los. Este mecanismo é composto por duas camadas: a camada DMIF e a camada de **Sincronização**, conforme ilustra a **Figura 2.11**. A primeira camada recebe os *streams* de dados da camada de transmissão de dados (chamada de camada **Transmux**) por meio dos canais **TransMux**, implementados na **Interface DMIF/Network** (ISO, 1997b). Estes *streams* podem ser compostos por *streams* independentes, mas com algum tipo de afinidade (normalmente a mesma QoS), e são decompostos em *streams elementares* por meio dos **FlexMuxes**.

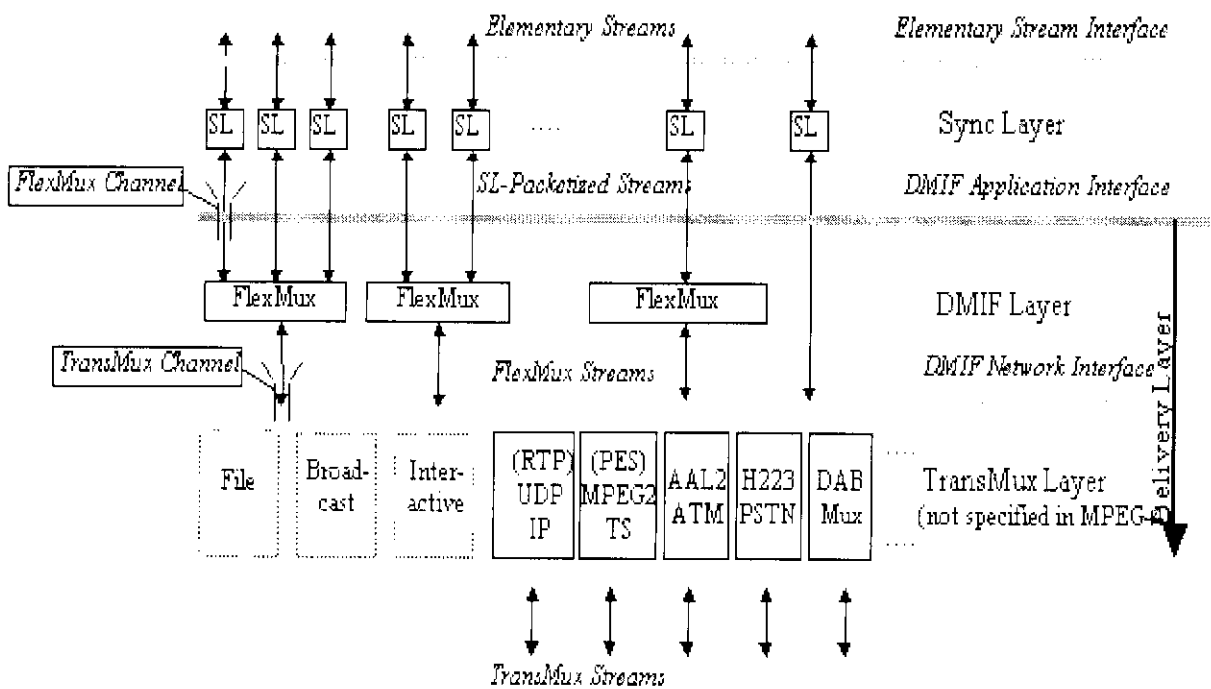


Figura 2.11 – Entrega de dados no MPEG-4 (ISO, 1997b)

A camada DMIF não é utilizada se a funcionalidade dos FlexMuxes não é requerida. Estes *streams* são então entregues à camada de sincronização, que efetua a sincronização dos *streams*, além de testes de consistência. Após a sincronização, os *streams* são entregues à camada de **Decodificação**. Nesta camada, cada *stream* de dados é tratada pela ferramenta de descompressão adequada, ou utilizado para compor a descrição da cena. Finalmente, os objetos de mídia são compostos pelo *compositor de cena*, utilizando informações de descrição de cena processadas anteriormente, de modo a reconstituir a informação multimídia.

2.6.1.5 Software de Referência e Conformidade

O *software* de referência é um *software* que deve ser utilizado para o teste de *streams* MPEG-4. O *software* desenvolvido se chama IM1-2D e pode ser encontrado no site do CSELT (www.csell.it).

O teste de conformidade visa estabelecer regras para que futuros *softwares* que utilizem o padrão MPEG-4 possam ser testados.

A sétima parte do padrão pretende suportar a construção de um *software* que se utilize das ferramentas MPEG-7, possibilitando a integração completa entre os dois padrões.

2.6.1.6 Players

Não existem muitos *players* de MPEG-4 disponíveis. O mais conhecido é o IM1-2D feito pelo CSELT (atual Telecom Italia Lab). O IM1-2D foi feito para ser o *software* de referência, relativo à parte 5 do padrão. A última versão disponível para *download* é a versão 2.7. Ela apresenta os *streams* MPEG-4 e vem com um codificador. No entanto, não foi possível implementar um exemplo de TV Interativa, pois foram encontrados problemas para realizar a codificação do vídeo. A **Figura 2.12** apresenta a tela do *player* IM1-2D.

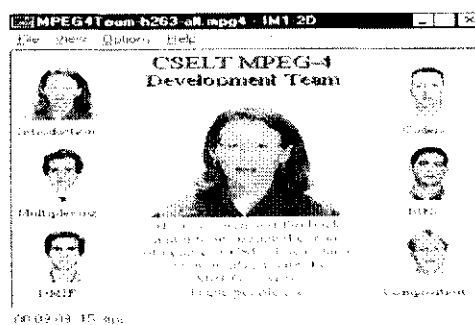


Figura 2.12 – Tela do IM1-2D

Existem *codecs* que implementaram somente a codificação do vídeo MPEG-4, no entanto, eles não são compatíveis com o IM1-2D. Esses *codecs* são o *MPEG-4 video codec* da Microsoft e o da DivX. Os vídeos codificados dessa forma podem ser visualizados no Windows Media Player ou no The Playa, *player* de vídeo feito pela equipe que desenvolveu o DivX.

2.6.1.7 Ferramentas de autoria

Ainda não existem muitas ferramentas de autoria para o MPEG-4. Uma ferramenta avaliada foi a desenvolvida pelo Instituto Superior Técnico de Portugal. O *IST MPEG-4 Video Compliant Framework* implementou diversas funcionalidades do padrão, entre elas a codificação separada dos objetos. A **Figura 2.13** apresenta uma tela do *software* na qual pode ser notada a presença da árvore de objetos, à esquerda, e da *Timeline*, abaixo.

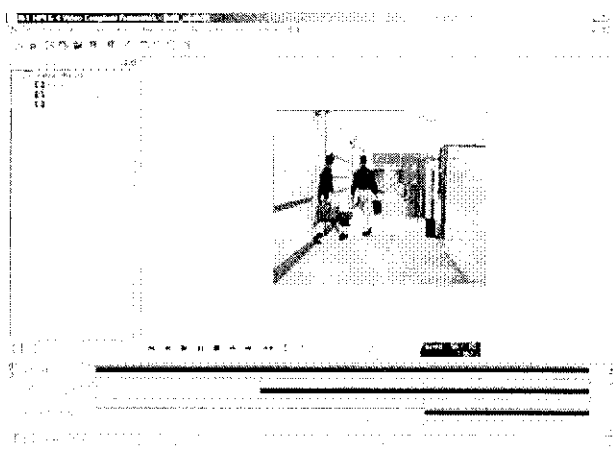


Figura 2.13 - Tela do IST MPEG-4 Video Compliant Framework

O *software* foi testado, porém, ele só trabalha com objetos já codificados, e os codificadores disponíveis não eram compatíveis com ele, bem como os *streams* gerados por ele não eram compatíveis com os *players*.

O CSELT está em fase final de desenvolvimento do Sinfonia, um *software* que compreenderá um *player*, uma ferramenta de autoria e um servidor de *streams*.

2.6.1.8 O formato de arquivo MPEG-4

O formato de arquivo do MPEG-4 (cuja extensão é **MP4**) foi projetado para conter informações, na forma de objetos, de todas as mídias que compõem uma apresentação multimídia. Trata-se de um formato flexível, extensível e que permite o intercâmbio, o gerenciamento, a edição e apresentação dos objetos de mídia. O formato permite, ainda, a reprodução local ou distribuída em uma rede de comunicação. Neste segundo caso, a

apresentação é baseada nas interfaces entre o formato de arquivo e os mecanismos de entrega de *streams* (TransMux).

O formato de arquivo foi projetado para ser independente da arquitetura de codificação, distribuição e apresentação, tendo sido o projeto baseado no formato de QuickTime® da Apple Inc (ISO, 1999a). A Figura 2.14 apresenta o esquema genérico de um arquivo MPEG-4.

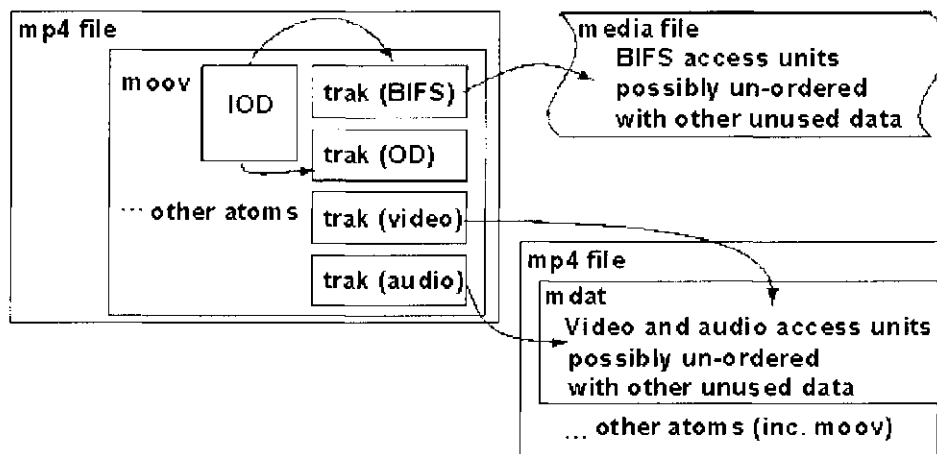


Figure 2.14 - Exemplo do formato de arquivo MPEG-4 para intercâmbio de dados (ISO, 1999a)

O formato de arquivo MPEG-4 é pode ser definido como *streamable*, uma vez que não define um protocolo de transporte, mas contém informações sobre como os *streams de mídia* devem ser transportados e apresentados. Desta forma, uma aplicação pode interagir com o formato de arquivo e obter informações para personalização de todo o ambiente (hardware, software, dispositivos) no qual o arquivo estiver sendo manipulado.

2.6.2 O padrão MPEG-7

Com o uso crescente de informação audiovisual, um modo de descrever o conteúdo dessas informações fez-se necessário, uma vez que as buscas nesses dados não eram “eficientes”. Para isso, foi utilizado o conceito de metadados, também conhecido como “dados sobre os dados”.

Diversos trabalhos nessa linha de pesquisa já foram feitos no grupo Intermídia do ICMC, como, por exemplo, o trabalho de Goularte (Goularte, 1998). O uso de metadados em conteúdo audiovisual tornará possível a busca minuciosa em grandes bases de dados, em um tempo bastante curto, semelhante à busca em texto. No entanto, para que seja possível essa busca, o conteúdo audiovisual deve ser descrito de uma forma padronizada.

O MPEG-7 (ISO, 1999c; ISO, 2000a), formalmente conhecido como “Interface de Descrição do Conteúdo Multimídia”, tem como objetivo especificar um conjunto padrão de descritores de vários tipos de informações multimídia. Essa descrição será associada ao conteúdo permitindo, assim, um modo de procura eficiente para o material de interesse do usuário.

Em termos de padronização, pretende-se definir com o MPEG-7:

- Tipos de Dados (*Datatypes*): elementos de descrição não específicos ao domínio audiovisual, que correspondem ao reuso de tipos básicos ou estruturas, utilizados por múltiplos Descritores e Esquemas de Descrição;
- Descritores (D): representam as características e definem a sintaxe e a semântica de cada representação de características;
- Esquemas de Descrição (DS): especificação da estrutura e da semântica dos relacionamentos entre seus componentes, que podem ser Ds ou DSs;
- Uma Linguagem de Definição de Descrição (*Definition Description Language*) (DDL) (ISO, 1999b): para permitir a criação de novos DSs (ou mesmo novos Ds) e permitir a extensão e modificação dos mesmos.

O MPEG-7 não especifica como é feita a extração das descrições/características do objeto, bem como não especifica um mecanismo de busca que faça uso das descrições (ISO, 2001c).

O MPEG-7 tem como principais áreas de aplicação (Nack & Lindasy, 1999):

- Educação;
- Jornalismo;
- Informação turística;
- Serviços culturais (museus, galerias de arte e exposições);
- Entretenimento (jogos);
- Sistemas geográficos;
- Aplicações biomédicas;
- *Home-Shopping* (busca por palavras-chave);
- Arquivos de filmes, vídeo e rádio (permitindo, assim, uma busca por assunto).

O ISDB (TV Digital Japonesa) já prevê o uso do MPEG-7 como modo de descrever seus programas e, com isso, tem trabalhado em conjunto com o *TV-Anytime Forum* na especificação de como utilizar o MPEG-7 na TV Interativa (TVAF, 2000c).

O padrão MPEG-7 é dividido em sete partes: Sistemas, Linguagem de Definição de Descrição, Visual, Áudio, Esquemas de Descrição Multimídia, *Software* de Referência e Teste de Conformidade, que serão apresentadas a seguir.

2.6.2.1 Sistemas (*Systems*)

A primeira parte do padrão MPEG-7 é o *Systems* (ISO, 2001b). É a arquitetura padrão do sistema, que especifica as funcionalidades para a comunicação das descrições do conteúdo multimídia.

O objetivo do *MPEG-7-Systems* é criar uma especificação não ambígua que permita aos desenvolvedores e usuários de MPEG-7 o desenvolvimento de decodificadores MPEG-7 e o preparo de Descrições MPEG-7 para o transporte e armazenamento.

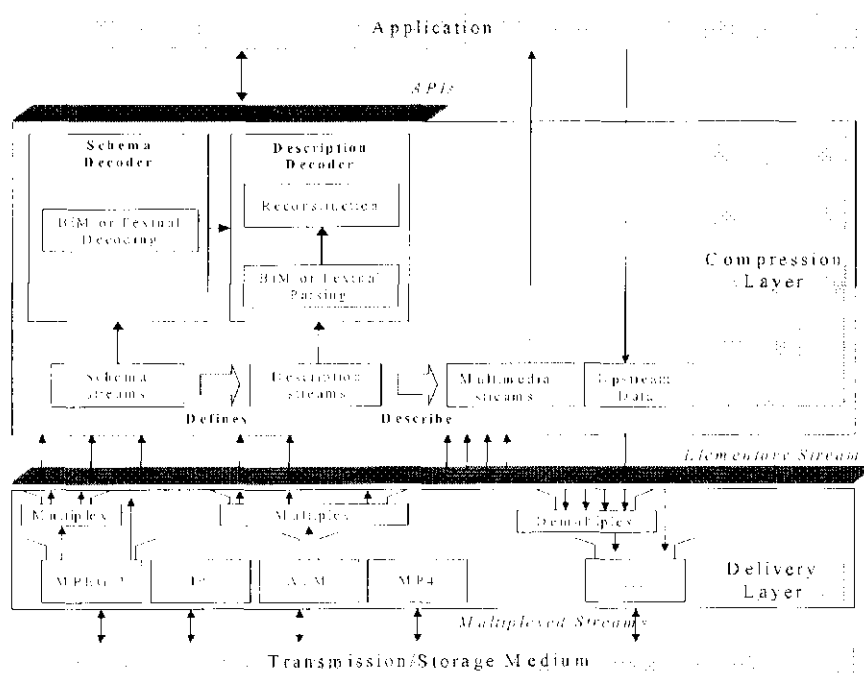


Figura 2.15 – Arquitetura MPEG-7 (ISO, 2001d)

A Figura 2.15 apresenta a arquitetura do MPEG-7. A camada mais baixa é o meio de transmissão/armazenamento (*Transmission/Storage Medium*), referente às camadas de distribuições da infra-estrutura.

O *stream* MPEG-7 pode ser transportado de diversas maneiras, como MPEG-2, ATM, IP ou MPEG-4, sendo que na Camada de Transmissão (*Delivery Layer*) é especificado como isso ocorrerá.

A Camada de Transmissão entregará *streams* elementares de MPEG-7 para a Camada de Compressão (*Compression Layer*). Esses *streams* elementares podem ser compostos por Esquemas de Descrição ou por Descrições, que podem estar representados ou no formato textual ou no formato binário (BiM – Binary Format for MPEG-7), dependendo da aplicação que o utilizará.

2.6.2.2 Linguagem de Definição de Descrição (DDL)

Para que as Descrições e Esquemas de Descrição sejam feitos, é necessária a definição de uma linguagem para a criação dessas descrições. O MPEG Group decidiu recentemente pela adoção da linguagem *XML Schema* (W3C, 2001a; W3C, 2001b; W3C, 2001c), por ser um padrão bastante difundido no mundo. Porém, como o *XML Schema* não foi feito especificamente para o conteúdo audiovisual, foram criadas extensões para o MPEG-7. Essas extensões são relativas à representação de matrizes e vetores e a tipos de dados relativos a tempo. Com isso, a DDL do MPEG-7 foi dividida em três partes: a linguagem estrutural do *XML Schema*, os tipos de dados do *XML Schema* e as extensões feitas, que podem ser encontradas no documento que especifica a DDL MPEG-7 (ISO, 2001c).

2.6.2.3 Visual

A terceira parte do padrão (ISO, 2001d) especifica ferramentas (descritores) para a descrição do conteúdo visual, como imagens, vídeo e imagens em 3D.

As ferramentas são definidas a partir de sua sintaxe em DDL e representação binária (BiM) e permitem a descrição das características visuais, como cor, textura, formato e movimento, assim como permitem a descrição da localização dos objetos descritos em uma imagem ou seqüência de vídeo.

2.6.2.4 Áudio

Esta parte do padrão MPEG-7 (ISO, 2001e) especifica ferramentas para a descrição do conteúdo sonoro.

O formato de codificação ou o meio de transmissão não são fatores limitantes, bem como não é limitado à descrição de músicas, discursos, efeitos especiais e trilhas sonoras. Também foram descritos os timbres de instrumentos e as saídas de mecanismos de reconhecimento de

voz. Através desses descritores pretende-se que seja possível a localização de áudio por semelhança.

2.6.2.5 Esquemas de Descrição de Multimídia (MDS)

A quinta parte do padrão MPEG-7 especifica as ferramentas para a descrição de informações multimídia.

As ferramentas podem ser divididas em elementos básicos (*Basic Elements*), elementos de descrição de conteúdo (*Content Description*), elementos de gerenciamento (*Content Management*), organização de conteúdo (*Content Organization*), navegação e acesso (*Navigation & Access*) e interação do usuário (*User Interaction*), como mostrado na **Figura 2.16**.

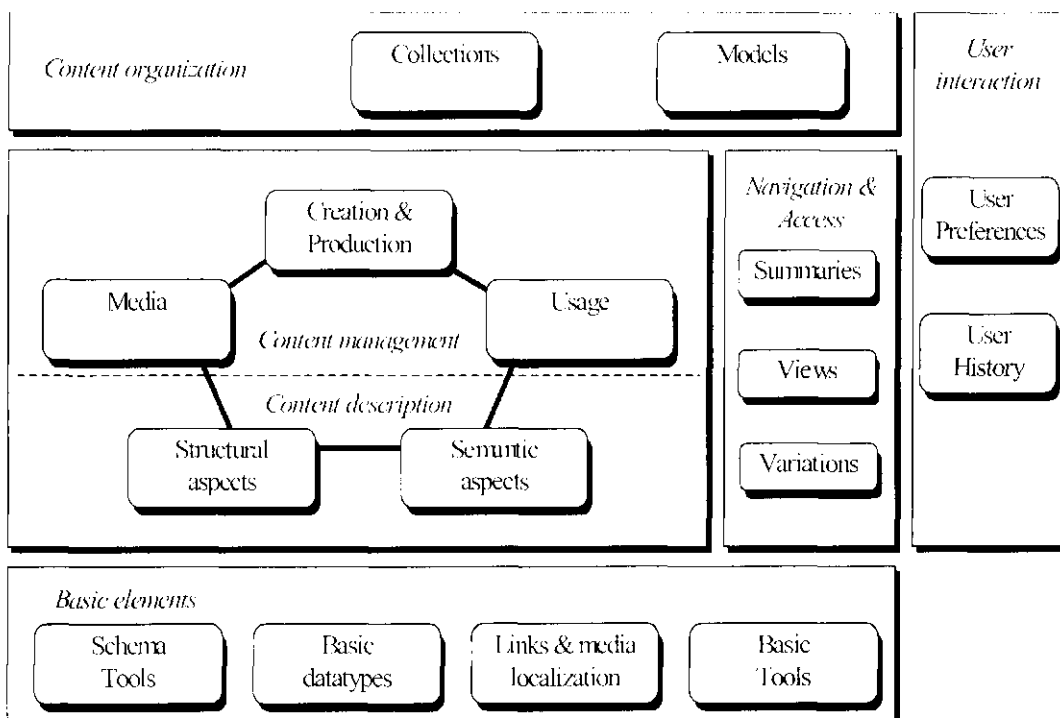


Figura 2.16 - Estrutura do MDS (ISO, 2001f)

2.6.2.6 Software de Referência

A sexta parte do padrão é a criação de um *software* de referência. Esse *software* é dividido em quatro partes:

- Um *parser* de DDL;
- Descritores visuais – o *software* deve criar descrições padrões do conteúdo visual;
- Descritores sonoros – o *software* deve criar descrições padrões do conteúdo sonoro;
- Esquemas de Descrição Multimídia – o *software* deve possuir esquemas de descrição multimídia padrões.

No entanto, ainda não foi otimizada a aquisição automática de características sonoras, visuais ou multimídia (ISO, 2001g).

2.6.2.7 Teste de Conformidade (*Conformance Test*)

A última parte do padrão definirá guias e procedimentos para o teste de conformidade com o padrão das implementações. Um exemplo da descrição de um vídeo utilizando o MPEG-7 MDS é apresentado na **Figura 2.17**.

```
<MediaInformation id="Audio Banco_dir">
  <MediaProfile>
    <MediaFormat>
      <Content>audio</Content>
      <FileFormat href="urn:mpeg:MPEG7FileFormatCS:4">
        <Name xml:lang="em">mp3</Name>
      </FileFormat>
      <Bandwidth>1220</Bandwidth>
      <AudioCoding>
        <Format href="urn:mpeg:MPEG7AudioCodingFormatCS:3">
          <Name xml:lang="em">mp3</Name>
        </Format>
        <AudioChannels side="2">2</AudioChannels>
      </AudioCoding>
    </MediaFormat>
  </MediaProfile>
</MediaInformation>
```

Figura 2.17 - Exemplo de uma descrição de um vídeo em MPEG-7

2.7 Apresentando o MPEG-21

Atualmente, existem muitas tecnologias, formatos e tipos de mídia que podem ser utilizados na construção, distribuição e consumo de conteúdo multimídia. Não há, entretanto, nenhum *framework* para descrever como esses elementos, já existentes ou em desenvolvimento, se relacionam.

O foco principal do MPEG-21 - padrão em fase inicial de desenvolvimento pelo grupo MPEG - é descrever como esses vários elementos de mídia podem ser utilizados de forma conjunta. Como resultado, espera-se que o MPEG-21 forneça uma estrutura aberta para a distribuição e consumo de conteúdo multimídia, preservando tanto os direitos do autor quanto do consumidor. Espera-se, também, que o padrão MPEG-21, enquanto *framework*, possa fornecer mecanismos padronizados para interoperabilidade de sistemas multimídia,

permitindo o uso transparente de objetos multimídia através de redes de comunicação que interliguem comunidades distintas ao redor do mundo (MPEG-21, 2001; Koenen, 2001).

2.7.1 MPEG-21 como um *framework* multimídia

Atualmente, as tecnologias multimídia suportam o uso de diferentes tipos de *players* para visualização de conteúdo. Essas tecnologias estão também associadas a mecanismos de rede, permitindo o acesso a informação e aos serviços a qualquer momento e de qualquer lugar. Neste ponto, terminais ubíquos têm sido desenvolvidos em conjunto com as tecnologias de comunicação. Entretanto, nenhuma solução pode ser considerada completa, uma vez que cada solução utiliza seus próprios modelos, procedimentos, formatos de mídia, gerando diferentes possibilidades de arquitetura e composição da infra-estrutura. De qualquer forma, pode-se dizer que os ambientes multimídia tendem a se tornar cada vez mais complexos e integrados.

Neste cenário, desenvolver uma estrutura comum para uso de multimídia facilitará a cooperação entre os diversos elementos da cadeia produtiva, que envolve os produtores de conteúdo, os provedores de acesso à rede de comunicação (geralmente a Internet), os produtores de interface, os administradores de redes de comunicação e os usuários finais.

Para contemplar os interesses desses diversos elementos, os objetos de mídia, que compõem uma aplicação multimídia, devem ser identificados, descritos, controlados e protegidos. O transporte e a entrega de um objeto de mídia deve respeitar parâmetros pré-estabelecidos durante a produção e/ou durante as negociações entre os terminais e as redes de comunicação, considerando-se os eventos que podem ocorrer durante uma sessão de interação. Adicionalmente, os aspectos de gerência de informações do usuário, tais como suas preferências, podem influenciar diretamente o consumo dos objetos de mídia e as questões que envolvem a propriedade intelectual, por exemplo. Esta abordagem permite a adaptabilidade ao nível da infra-estrutura e o controle de itens como personalização, segurança, tarifação, dentre outros.

Dentro do escopo do MPEG-21, a sintaxe e a semântica dos objetos de mídia passam a ser referenciadas no contexto dos itens digitais. Neste sentido, o MPEG-21 prevê a definição de sete subgrupos de recomendações: a) *Schema* de declaração do item digital (uma abstração de URI com potencial de interoperabilidade) l; b) identificação do item digital (natureza,

granularidade, tipo); e) manipulação de protocolos de acesso ao item digital; d) gerência da propriedade intelectual e proteção de direitos; e) interoperabilidade entre terminais e entre as redes de comunicação; f) formatos de representação de itens digitais; f) controle de eventos.

Observa-se que as recomendações MPEG-21 estão diretamente relacionadas com as questões de interoperabilidade, o que tem influenciado as propostas de contribuição durante o processo de padronização do MPEG-21.

2.7.2 O escopo do MPEG-21

O escopo do MPEG-21 pode ser definido como a integração das tecnologias e recursos para permitir a autoria, distribuição e consumo de conteúdo multimídia, além de suportar o gerenciamento da propriedade intelectual.

Do ponto de vista tecnológico, definir um *framework* exige contribuições de diversos segmentos da cadeia produtiva. Uma proposta de estrutura de aplicações multimídia é o passo inicial do MPEG-21, identificando aspectos de integração de tecnologias e sistemas, bem como novos critérios de interação em todos os níveis (ou camadas) de uma aplicação, sendo que esses níveis estão relacionados com o usuário, com a infra-estrutura de comunicação, com os dispositivos de apresentação e com a aplicação multimídia propriamente dita.

2.7.3 O modelo do usuário

No contexto do MPEG-21, um usuário é toda e qualquer entidade que interage num ambiente MPEG-21, manipulando itens digitais de mídia.

Os usuários são identificados através de seus relacionamentos com outros usuários ou por alguma interação no ambiente como um todo. Todo e qualquer tipo de usuário (produtor, distribuidor, consumidor) é identificado da mesma forma no MPEG-21, uma vez que, independente da ação do usuário, este se relaciona com itens digitais de mídia, que são o foco central da proposta do padrão.

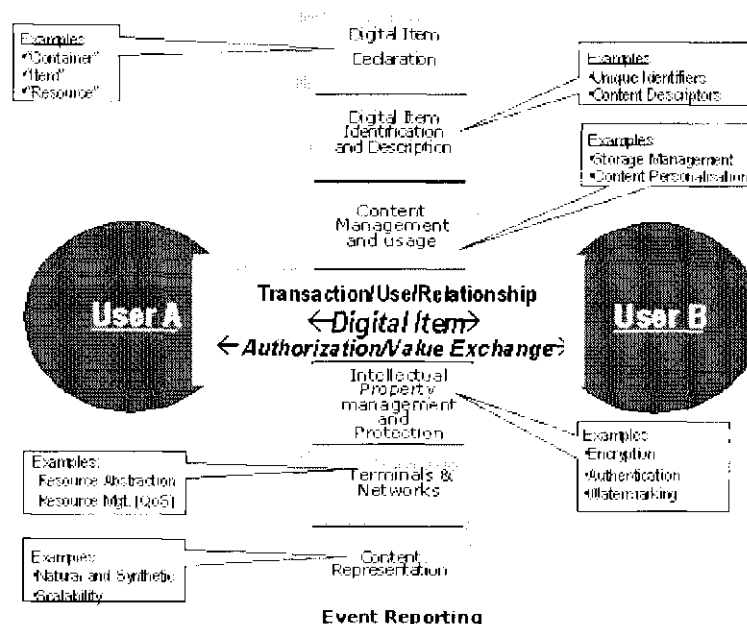


Figura 2.18 - Modelo de eventos e interfaces que descrevem interações MPEG-21 (MPEG-21, 2001)

Em seu nível mais básico, o MPEG-21 oferece uma estrutura a partir da qual um usuário interage com os itens digitais de mídia e participa da geração de interações no ambiente como um todo. Neste ponto, as interações são baseadas nos itens digitais e as restrições de interações no ambiente estão definidas na estrutura desses itens digitais, conforme ilustra a Figura 2.18.

2.7.4 MPEG-21 e os itens digitais

Um padrão, tal como MPEG-21, que propõe estruturar a integração de tecnologias em larga escala, deve descrever, de forma precisa, como os itens digitais de mídia devem ser formados, gerenciados e utilizados por todas as entidades pertencentes ao sistema.

Existem muitas maneiras para descrever um item digital de mídia. Desta forma, um desafio adicional do MPEG-21 é configurar modelos de descrição que sejam suficientemente flexíveis quanto à interoperabilidade. Tais modelos serão considerados úteis se produzirem formatos capazes de representar todo e qualquer item digital de mídia.

A especificação da declaração de um item digital (MPEG-21, 2001) consiste do uso de um dos modelos disponíveis e que seja suficientemente completo para permitir o gerenciamento desse item digital. Um exemplo pode ser visto a seguir.

Supondo a declaração de uma *web page* como um item digital. Uma *web page* consiste tipicamente de um documento HTML e seus *links* para outros documentos (hiperdocumentos). Adicionalmente, uma *web page* pode conter também elementos de mídia, tais como imagens (JPEGs e GIFs), vídeo (AVI, MOV, MPG), áudio (WAV, MIDI, MP3) e animações (VRML, Flash). Neste exemplo simples, pode-se declarar o item digital como sendo a *web page* acrescida de toda a árvore de elementos formadores. Aplicações como Java *applets* e *scripts* podem modificar a *web page* para determinar, por exemplo, qual o idioma do usuário que estiver visualizando o hiperdocumento. O ponto-chave é que a interação pode modificar o conteúdo do item digital. Desta forma, mecanismos devem permitir a separação dos elementos que compõem a estrutura do item digital de outros elementos que, potencialmente, manipulam eventos e/ou geram interações no ambiente. Observa-se, então, que algumas interações podem ser permitidas em algumas plataformas e não serem possíveis e/ou permitidas em outras plataformas. No entanto, a estrutura do item digital deve ser universal (independente da plataforma).

2.7.5 Declaração de itens digitais

O objetivo da declaração de um item digital é descrever um conjunto de operações e conceitos abstratos que instanciem esse item a um dos modelos de itens digitais disponíveis no escopo do MPEG-21.

Dentro de um modelo, um item digital passa a ser a representação digital de um objeto de mídia e suas operações que geram interações no ambiente. O modelo, por sua vez, deve ser tão flexível e genérico quanto possível, fornecendo todas as primitivas necessárias para compor a funcionalidade de uma instância desse modelo (item digital). Adicionalmente, o modelo deve fornecer conceitos comuns que permitam o mapeamento (*mapping*) entre esquemas que descrevem itens digitais de mídia de modelos diferentes.

2.7.6 Identificação de itens digitais

O escopo da identificação de um item digital inclui: a) como identificar um item digital e os recursos associados ao item; b) como identificar se um endereço de rede (por exemplo, o *IP address*) está relacionado a um ou mais itens digitais; c) como identificar esquemas de descrição relativos a modelos que instanciam itens digitais; d) como usar identificadores para associar itens digitais a informações e/ou metadados do ambiente.

A especificação de identificação de itens digitais não cobre ações que não estejam relacionadas no modelo de item digital. Os identificadores cobertos por uma identificação de item digital devem ser incluídos diretamente na identificação e relacionados aos itens permitidos pelo modelo em uso. A **Figura 2.19** apresenta as possibilidades de identificadores para os itens digitais e seus relacionamentos em um modelo de item digital.

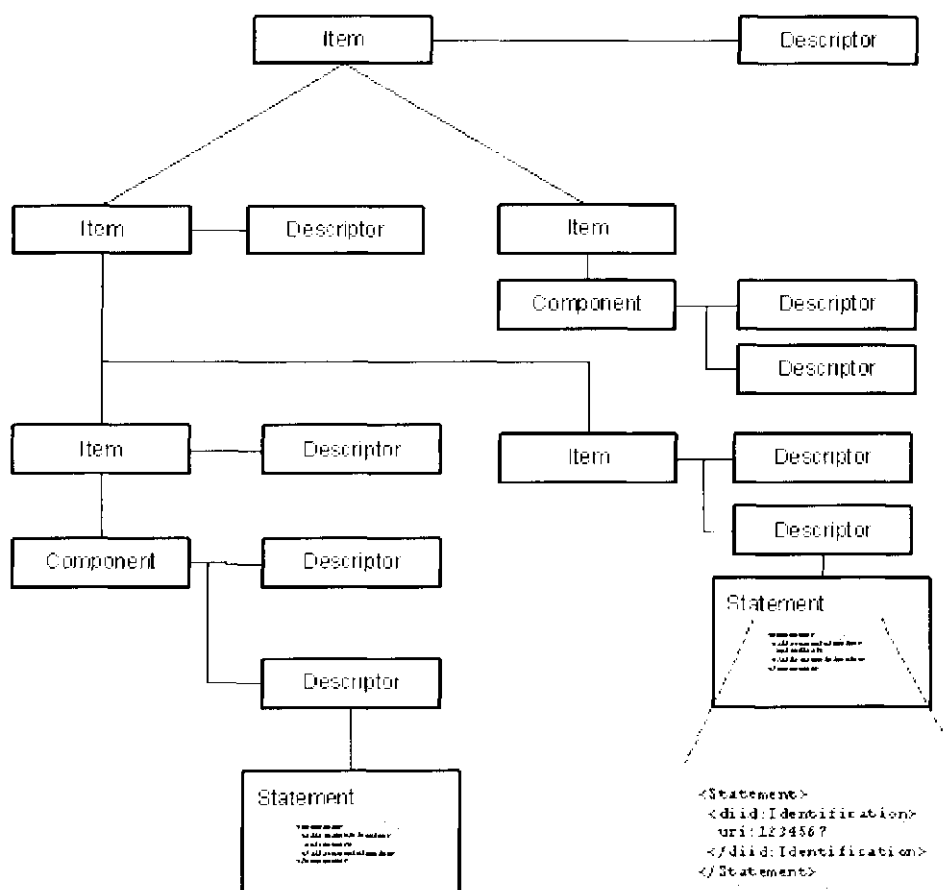


Figura 2.19 - Relacionamento entre a declaração, identificação e descrição de um item digital

2.7.7 Adaptação de itens digitais

O objetivo dos elementos associados aos terminais e à rede de comunicação é permitir o acesso transparente aos itens digitais de mídia. Isto permite a personalização do acesso a esses itens de mídia e o desenvolvimento de aplicações que utilizem informações para controle automático de ações. Para se atingir esse objetivo, é necessário o desenvolvimento de modelos de adaptação dos itens digitais ao ambiente em que estiverem sendo referenciados.

O MPEG-21 propõe o desenvolvimento de um *motor de adaptação de itens digitais*, que é composto por mecanismos e descrições que analisam as relações descritas entre os itens

digitais e os recursos associados. A Figura 2.20 apresenta um esquema para adaptação de um item digital a ambientes distintos.

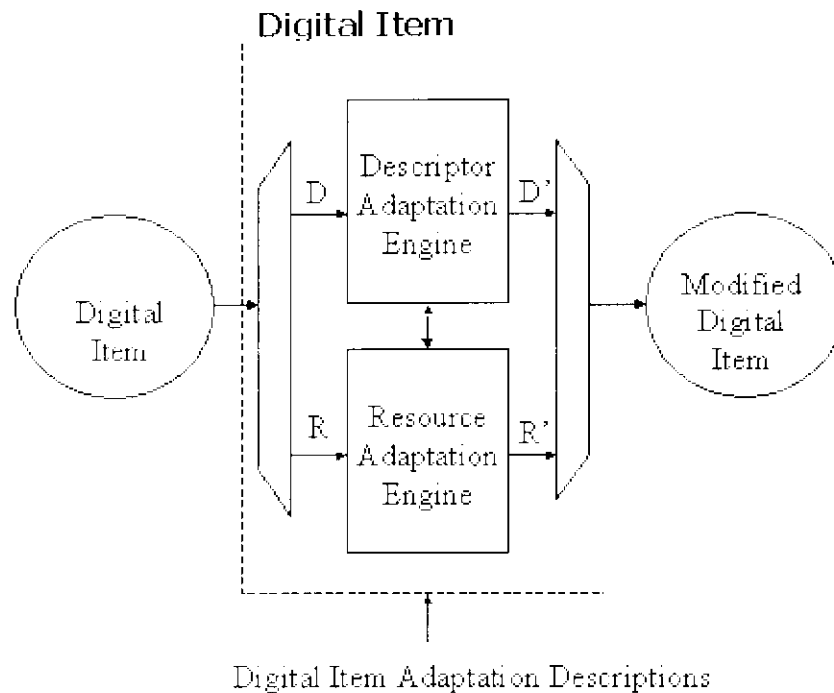


Figura 2.20 – Esquema para o conceito de adaptação de item digital (MPEG-21, 2001)

O estágio atual de desenvolvimento do MPEG-21 contempla seis categorias de investigação, para as quais novas propostas têm sido encaminhadas ao grupo MPEG, a saber:

- **características do usuário:** propostas que especifiquem ferramentas de descrição das características de um usuário, incluindo suas preferências em relação à interação e também as preferências em relação à infra-estrutura disponível; aspectos de mobilidade devem ser considerados em relação ao acesso ao item digital;
- **potencialidades dos terminais:** propostas que especifiquem ferramentas de descrição que identifiquem a potencialidade de uso de diversos tipos e configurações de terminais de exibição/interação multimídia, incluindo aspectos de decodificação, composição, protocolos de comunicação, software e demais sistemas relacionados;
- **características da rede:** propostas que especifiquem ferramentas de descrição das potencialidades para adaptabilidade de uma infra-estrutura de comunicação às necessidades de uma aplicação, considerando-se as características dos itens digitais envolvidos;

- *características naturais do ambiente*:: propostas que especifiquem ferramentas de descrição que permitam personalização das interfaces em relação às propriedades (iluminação, ruídos, área, objetos) e ao contexto de um determinado ambiente;
- *adaptability dos recursos*: propostas que especifiquem ferramentas que auxiliem a manipulação de recursos através de metadados que descrevam esses recursos; adicionalmente, essas ferramentas devem permitir ajustes nas propriedades dos recursos para os casos em que os parâmetros de qualidade de serviço (QoS) devam ser ajustáveis;
- *mobilidade da sessão*: propostas de ferramentas que especifiquem como transportar o estado de um item digital de um ponto a outro; ao nível técnico, essas propostas devem cobrir a captura do estado corrente no ponto de origem, a transferência e a reconstrução do estado no ponto de destino, envolvendo, portanto, técnicas clássicas e modernas de migração de processos em sistemas distribuídos (Coulouris et al., 1994).

2.8 Considerações finais

No contexto da multimídia interativa, novas tecnologias têm sido criadas e, gradativamente, vão sendo incorporadas às aplicações.

Ao contemplar as alternativas tecnológicas para o desenvolvimento de uma aplicação, deve-se ressaltar a importância de uma análise de requisitos para que se possa escolher as tecnologias mais adequadas a um determinado domínio de aplicação.

No contexto no qual este trabalho está inserido, uma importante contribuição reside na investigação na apresentação de uma modelagem orientada a objetos e baseada nos padrões MPEG-4 e MPEG-7, contemplando aspectos de compartilhamento, armazenamento e recuperação, distribuição e apresentação de conteúdo multimídia em aplicações associadas a ambientes interativos. A partir da apresentação tecnológica feita neste capítulo, uma visão geral das técnicas, metodologias, métodos e linguagens de modelagem, que podem ser utilizadas no desenvolvimento de aplicações multimídia, é apresentada no **Capítulo 3**.

3.1 Considerações iniciais

Devido à demanda do mercado por aplicações multimídia e interativas, com requisitos funcionais, elegantes e flexíveis, organizações de padronização como a ISO (*International Standardization Organization*) e W3C (*World Wide Web Consortium*) produziram padrões, tecnologias e recomendações para a área, tais como HTML, XML e DOM (*Document Object Model*). Devido à sua simplicidade, o padrão HTML foi rapidamente absorvido pelo mercado. Conforme as aplicações foram se tornando mais complexas e exigindo mais recursos, outros padrões e recomendações foram se juntando ao HTML para atender à demanda. Tais tecnologias, contudo, não contemplam metodologias de desenvolvimento para aplicações.

A modelagem, projeto e desenvolvimento de aplicações e sistemas computacionais é suportada por ferramentas e técnicas, advindas geralmente da Engenharia de Software, que garantem qualidade, produtividade e baixo custo de produção. Muito do que se dispõe tecnologicamente no mercado para a produção de sistemas multimídia é centrado em apresentação, demonstrando uma grande preocupação com a interface e pouca com a estrutura da informação. Isto fica evidenciado pelo número de ferramentas como HTML, DHTML, SMIL, Flash, dentre outras, que permitem a construção de visuais impressionantes e bons níveis de interação (principalmente quando utilizados em conjunto com linguagens como Java ou Java-Script), porém, não contemplam uma representação estruturada da informação, o que seria de grande valia para a o desenvolvimento de aplicações personalizadas e conscientes de contexto, por exemplo.

Do ponto de vista conceitual, cada formato de mídia pode requerer ferramentas próprias para seu manuseio, dificultando o processo de produção de uma apresentação. Como consequência, a sincronização dessas mídias torna-se um problema. Como exemplo, pode-se imaginar um vídeo de uma propaganda de um produto, onde alguém fala das características do mesmo enquanto imagens, texto e algumas animações (desenhos animados inseridos no vídeo) são apresentados. É desejável que o áudio, descrevendo a cor do produto, deve ser

tocado sincronamente com as imagens que mostram características desse mesmo produto. Da mesma forma, as animações devem ser inseridas em posições específicas e com tempos muito bem determinados. Conseguir sincronizar estas mídias pode ser um trabalho penoso. A abordagem usual utiliza todas as mídias juntas em um mesmo arquivo (garantindo a sincronização) para a transmissão em rede, porém, existem vantagens em transmitir essas mídias separadamente. Pode-se conseguir níveis diferentes de controle da apresentação durante a exibição de cada mídia individualmente, podendo-se escolher, por exemplo, se um determinado objeto irá ou não aparecer na cena e de que forma. Contudo, transmitir as mídias individualmente requer novas medidas de sincronização, uma vez que a rede não fornece garantia de tempo de entrega. Nota-se, portanto, que o processo de modelagem é decisivo na construção dos novos e complexos ambientes interativos, considerando a aplicação multimídia como parte desse ambiente, conforme propõe este trabalho.

3.2 Métodos, técnicas, metodologias e modelos: uma breve apresentação

O projeto de aplicações multimídia difere do projeto e desenvolvimento de software tradicionais em vários aspectos críticos, uma vez que aplicações multimídia apresentam dificuldades em sua modelagem e em seu projeto que não são encontradas no projeto de sistemas tradicionais. Como consequência, métodos específicos para modelagem e projeto deste tipo de aplicação são necessários, métodos estes que precisam atender as diferentes necessidades dos vários tipos de aplicações multimídia que têm surgido.

Diversos métodos para modelagem e projeto de aplicações multimídia têm sido pesquisados e propostos. Dentre esses métodos, destacam-se o HDM (*Hypermedia Design Method*) (Garzotto et al., 1993), o RMM (*Relationship Management Methodology*) (Isakovitz et al., 1995; Isakovitz et al., 1998), o OOHDM (*Object Oriented Hypermedia Design Model*) (Schwabe, 1995), o *Analysis and Design of Web-Based Information Systems* (Takahashi et al., 1997), o EORM (*Enhanced Object Relationship Model*) e o *Database-Oriented Approach to the Design of Educational Hyperbooks* (Fröhlich, 1997) e o HBMS (*Hypertext Model Based on Statecharts*) (Turine et al., 1997).

Alguns dos métodos para modelagem e projeto de aplicações multimídia, propostos na literatura, utilizam o paradigma da orientação a objetos e, conseqüentemente, permitem a construção de novos sistemas multimídia explorando o “reuso” de componentes já modelados em sistemas anteriores (Nemetz, 1997; Rossi et al., 1995; Schwabe, 1995). Porém, nenhum destes métodos para projeto e modelagem de sistemas multimídia, propostos na

literatura, é baseado em um padrão ISO para especificação do comportamento do conteúdo multimídia. Além disso, a maioria desses métodos não faz parte de uma metodologia completa de desenvolvimento que contemple os padrões, tecnologias e recomendações, tais como XML, SGML, SMIL, XSL, DOM e os padrões da família MPEG, por exemplo. Observa-se, ainda, que esses métodos concentram-se na modelagem da aplicação multimídia ao nível da estrutura e da interface, porém no contexto único da aplicação enquanto software. Nesses métodos, não há ferramentas para modelagem do ambiente interativo como um todo, visão essa que é discutida neste trabalho; o ambiente interativo passa a ser formado pela infraestrutura de comunicação, pela aplicação em si, pelo(s) usuário(s) e pelos objetos de mídia que compõem as cenas em apresentação. As **Subseções 3.2.1 a 3.2.8** apresentam uma breve descrição de cada um dos principais métodos, metodologias e técnicas de que podem ser aplicados à modelagem de sistemas multimídia.

3.2.1 Relationship Management Methodology (RMM)

O RMM é uma metodologia para projeto estruturado de hipermídia e seu nome é derivado da visão de hipermídia como um veículo para gerenciar relacionamentos entre objetos de informação. O método é descrito em (Isakovitz et al., 1995), dentro do contexto do ciclo de desenvolvimento de software completo e especifica os passos para um projeto de um sistema hipermídia desde a fase de análise e projeto até a fase de teste, focalizando os aspectos de projeto e implementação da aplicação, principalmente as etapas destinadas a especificação do domínio e dos mecanismos de navegação da aplicação.

O núcleo do RMM é um modelo de dados chamado RMDM (*Relationship Management Data Model*). O RMDM é a primeira fase do RMM, sendo composto de três passos: *E-R Design*, *Entity Design*, *Navigation Design*. Eis as principais características de um desses três passos:

- *E-R Design*, primeiro passo do RMDM e do próprio RMM, consiste em representar as informações do domínio da aplicação utilizando um diagrama de Entidade-Relacionamento; esta forma de representação foi escolhida por ser familiar aos desenvolvedores;
- *Entity Design*, segundo passo do RMM, é exclusivo de sistemas hipermídia e determina como as informações sobre as entidades, definidas no passo anterior, serão apresentadas aos usuários e como esses usuários poderão ter acesso às informações;

- *Navigation Design*, terceiro passo do RMM, permite o modelo e o projeto caminhos pelos quais o usuário poderá navegar no hipertexto; no RMDM, a representação da navegação é suportada através de diferentes primitivas, tais com *guided tour*, *index* e *indexed guided tour*.

A segunda parte da metodologia RMM não está diretamente relacionada com a modelagem e projeto de estruturas de acesso, e consiste de 4 passos: *Conversion Protocol Design*, *User-interface Design*, *Run-time Behavior Design* e *Construction and Testing*. Desses passos, os dois primeiros são os mais importantes:

- *Conversion Protocol Design*: transforma cada elemento do diagrama RMDM em um elemento da plataforma destino, na qual será implementado o sistema;
- *User-interface Design*: consiste do projeto das interfaces (telas) para todos os objetos que aparecem no diagrama RMDM, resultado final do terceiro passo.

Isakowitz propõe uma nova versão do RMM, chamada de *RMM extended* (Isakovitz et al., 1998). Essa nova versão da metodologia possui como objetivo cobrir falhas do RMM padrão (Isakovitz et al., 1995). O *RMM extended* introduz um novo elemento, denominado *M-Slice*, que possibilita maior reusabilidade de projeto.

3.2.2 Object Oriented Hypermedia Design Model (OOHDM)

Este método utiliza os mecanismos de abstração e composição para permitir a descrição concisa de itens de informação complexos (comuns em sistemas Hiperemídia) e para possibilitar a especificação de padrões de navegação e transformações de interface (Schwabe, 1995). No OOHDM, um sistema Hiperemídia é construído em quatro passos:

- *Conceptual Design* (Modelo do Domínio da Aplicação);
- *Navigational Design* (Modelo de Navegação);
- *Abstract Interface Design* (Projeto de Interface);
- *Implementation*.

O OOHDM, durante o *Conceptual Design*, faz uso dos princípios de modelagem orientada a objetos do OMT (Schwabe, 1995) e difere do RMM em dois pontos principais: a) pela ênfase no projeto do modelo de navegação e no projeto da interface abstrata que o OOHDM possui; b) pelo uso do paradigma de orientação a objetos. Neste primeiro passo, classes conceituais podem ser construídas a partir de relacionamentos de agregação e

generalização/especialização. A principal preocupação, durante o *Conceptual Design*, é modelar a semântica da aplicação, sem considerar os tipos de usuários e tarefas que realizarão.

O modelo de navegação é uma “visão”, em termos de base de dados, do modelo conceitual da aplicação e é expressado em dois esquemas: *Navigational Class Schema* e *Navigation Context Schema*. No primeiro esquema, são definidos os objetos navegáveis da aplicação. Similar ao IIDM, existe um conjunto pré-definido de classes navegacionais, tais como *nós*, *links* e estruturas de acesso (como *indexes* e *guided tour*). No OOHDM, o projeto do modelo de navegação considera os tipos de usuários e o conjunto de tarefas que realizarão durante o uso da aplicação. Diferentes modelos de navegação podem ser construídos para um mesmo modelo conceitual, uma vez que uma mesma aplicação pode possuir diversos conjuntos diferentes de usuários e tarefas.

No terceiro passo, *Abstract Interface Design*, o OOHDM define quais os objetos de interface que o usuário irá perceber, quais os objetos de interface irão ativar a navegação, a forma como os objetos multimídia da interface serão sincronizados e como as transformações de interface irão ocorrer. É possível a construção de diferentes interfaces para um mesmo modelo de navegação. O último passo, *Implementation*, consiste do mapeamento dos modelos de navegação e de interface abstrata em objetos concretos disponíveis para o ambiente de implementação escolhido.

3.2.3 Enhanced Object Relationship Model (EORM)

Este modelo, assim como o OOHDM, utiliza o OMT para definir as primitivas do modelo conceitual do sistema (Millet et al., 1996). O EORM é composto de três etapas:

- especificação da estrutura de classes da aplicação (*Class Framework*);
- definição da relação entre as classes que compõe a aplicação (*Composition Framework*);
- definição dos aspectos de interface.

O modelo do domínio da aplicação é descrito através de classes e de relacionamentos de generalização e associação em um esquema conceitual, construído a partir das primitivas do OMT.

O modelo FORM estende o modelo ORM, incluindo semântica nas relações de classes, de forma que as relações são mapeadas nas classes de ligação, que possuem uma estrutura e um comportamento próprio. Uma classe de ligação pode representar uma ou mais relações da aplicação e estas classes podem ser agrupadas em uma hierarquia, com o objetivo de indicar herança entre elas. O modelo EORM não separa de forma explícita o projeto conceitual do projeto de navegação do sistema.

3.2.4 Hypermedia Modelling Technique (HMT)

O HMT utiliza quatro modelos para descrever uma aplicação: o *Object Model*, o *Hyperobject Model*, o *Navigation Model* e *Interface Model*. O primeiro modelo descreve os objetos do domínio do sistema e seus relacionamentos, utilizando os conceitos e notações do modelo de objetos OMT (Rumbaugh, 1991).

O *Hyperobject Model* refina o *Object Model*, associando semântica aos relacionamentos entre os objetos do domínio. O *Navigation Model* define os objetos de navegação e os contextos de navegação da aplicação. Através do *Interface Model* do HMT, descreve-se como a informação será apresentada ao usuário. Essas definições são baseadas no *Navigation Model* e no *Hyperobject Model*.

3.2.5 A Database-Oriented Approach to the Design of Educational Hyperbook

A *Database-Oriented Approach to the Design of Educational Hyperbooks* é um método voltado para modelagem e projeto de sistemas educacionais “adaptativos”. Este método utiliza a linguagem de modelagem orientada a objetos *O-Telos*, que combina conceitos de orientação a objetos com regras de dedução e restrições. Segundo o método, o modelo de um sistema é composto por quatro submodelos: *Domain Model*, *Navigational Model*, *Visualization Model* e *User Model*. A **Figura 3.01** ilustra o relacionamento entre esses modelos.

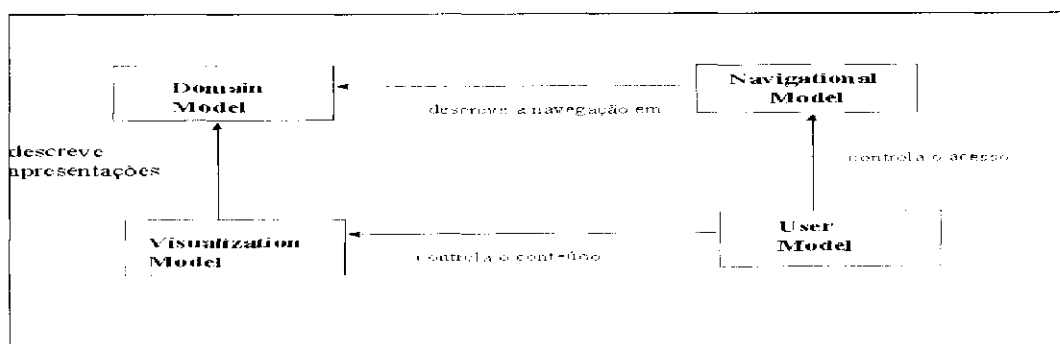


Figura 3.01: Modelos do *Database-Oriented Approach to the Design of Educational Hyperbook*

O primeiro submodelo, *Domain Model*, descreve as entidades do domínio do sistema, tratadas como classes, e os seus relacionamentos. Para isto, o método mescla conceitos e notações de orientação a objetos e de diagramas E-R (Entidade-Relacionamento).

As classes de uma aplicação em um sistema podem ser organizadas em uma hierarquia simples com relacionamentos de herança únicos entre elas. Um modelo de domínio pode possuir superclasses, subclasses, classes, instâncias e relacionamentos entre as entidades (entre classes e instâncias de classes). Este modelo de domínio é similar ao RMM, porém estende-o através de conceitos de classe e herança.

A navegação, neste modelo, deve ser suportada de duas formas:

- *Data Model Navigation*: baseado no modelo de domínio e similar à modelagem no OOHDH; no *Domain Model*, este tipo de navegação permite uma leitura seletiva e acesso rápido à informação através dos *hyperlinks*;
- *Sequential Navigation*: apresenta páginas em uma seqüência.

O terceiro submodelo, denominado *Visualization Model*, especifica como a informação será apresentada. O *User Model* adapta o conteúdo e a estrutura de navegação de *hyperbooks*. Este modelo de usuário possui suas limitações, uma vez que é baseado no fato do usuário visitar ou não as páginas, não descrevendo o quanto ele realmente absorveu do conteúdo de cada página que visitou. Entre os métodos descritos nesta breve revisão, este é o único método que possui recursos específicos para a modelagem e projeto de aplicações Hipermídia educacionais adaptativas.

3.2.6 Analysis and Design of Web-based Information Systems

Métodos para modelagem e projeto de sistemas Hipermídia como o RMM mostram-se bastante eficientes para aplicações do tipo “quiosque de informações”, nas quais o usuário navega pelas informações de uma maneira sistemática. Mas para alguns sistemas Hipermídia, o acesso a uma parte específica da informação é apenas parte dos objetivos do usuário, que também necessita de processar dados, comunicar-se e colaborar com seus colegas através da aplicação (*WBISs-Web-based Information Systems*). O método *Analysis and Design of Web-based Information Systems* (Takahashi et al., 1997) foi desenvolvido para suprir as necessidades deste tipo de aplicações Hipermídia, onde usuários necessitam processar

dados além de possuir acesso a informações. Em conjunto com este método estão sendo desenvolvidas ferramentas para suportá-lo.

Este método consiste de duas abordagens: estática e dinâmica. O modelo Entidade-Relacionamento (E-R) é utilizado para descrever a modelagem e projeto estático de WBISs. Para o modelo e projeto dinâmico, o método utiliza cenários. O modelo estático, baseado em diagramas E-R, define quais são as entidades e como elas estão relacionadas. Os cenários definem quais os recursos da Web que são utilizados e acessados e a quem estão relacionados.

Os passos do método *Analysis and Design of Web-based Information Systems* são: *E-R Analysis* (E-R), *Scenario Analysis* (cenário), *Architecture Design* (diagramas RMDMW), *Attribute Definition, Construction, Test*. Os diagramas RMDMW (*Relationship Management Data Model for WBISs*) são uma extensão dos diagramas RMDM da metodologia RMM (Isakovitz et al., 1995).

3.2.7 OOMMA

Conforme discutido, aplicações multimídia podem ser definidas como aquelas que combinam elementos de hardware e software para prover interatividade a cenas compostas por objetos de mídia independentes e que se relacionam no tempo e no espaço. Adicionalmente, a sincronização entre esses objetos pode ser organizada de forma hierárquica e através de um modelo que descreva a interação e a navegação. O modelo OOMMA (*Object Oriented Modelling of Multimedia Applications*) é uma extensão à linguagem de modelagem UML e que permite modelar aspectos de uma aplicação multimídia, gerando um modelo coerente e integrado e observando os seguintes aspectos: a) a estrutura das cenas, incluindo a arquitetura da aplicação e o cenário de uso dos objetos multimídia; b) a apresentação dos objetos no tempo e no espaço; c) o comportamento temporal dos objetos de mídia; d) a interação baseada em eventos (Engels, 2000).

Em relação às extensões UML, o modelo OOMMA provê uma linguagem (OOMMA-L) para descrever as relações temporais entre os diversos tipos de mídia. A construção de interfaces é suportada por uma notação gráfica que permite especificar o posicionamento de objetos em uma cena e sua hierarquia de apresentação. Por fim, são oferecidas *guidelines* para integrar os

diversos diagramas UML na construção de aplicações multimídia complexas (Sauer & Engels, 1999).

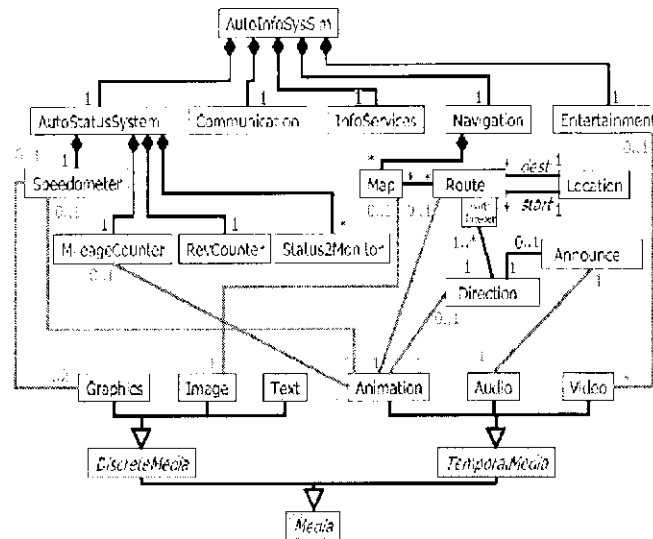


Figura 3.02 – Estrutura OOMMA representada em um diagrama de classes (Engels, 2000)

Pelo método OOMMA, a estrutura de uma aplicação multimídia por ser representada em um diagrama de classes, que descreve logicamente a hierarquia das classes que compõem a aplicação, conforme ilustrado na **Figura 3.02**. As relações temporais são descritas em um diagrama de seqüência, que possui algumas sintaxes gráficas estendidas da linguagem UML padrão. O diagrama de apresentação é usado para projetar a interface do usuário, contemplando objetos de visualização e objetos de interação. As interações são descritas através de *statecharts* (diagramas de estados).

3.3 A linguagem UML

Um dos principais problemas no desenvolvimento de novos sistemas utilizando a orientação a objetos nas fases de análise de requisitos, análise de sistemas e *design* é que não existe uma notação padronizada e realmente eficaz que abranja qualquer tipo de aplicação que se deseje. Cada simbologia existente possui seus próprios conceitos, gráficos e terminologias.

O surgimento da UML apresentou-se como uma possibilidade concreta para uma metodologia de modelagem. UML é mais que a padronização de uma notação. É também o desenvolvimento de novos conceitos não normalmente usados. Por isso e muitas outras razões, é que UML representa um passo na direção do estado da arte em modelagem de objetos.

UML permite abordar o caráter estático e dinâmico do sistema a ser analisado levando em consideração, já no período de modelagem, todas as futuras características do sistema em relação a utilização de semânticas próprias da linguagem a ser utilizada, utilização de banco de dados bem como as diversas especificações do sistema a ser desenvolvido de acordo com as métricas finais do sistema.

Os conceitos da orientação a objetos têm sido discutidos há muito tempo, desde o lançamento da primeira linguagem orientada a objetos, a SIMULA. No entanto, a literatura demonstra a dificuldade de se encontrar linguagens que possibilitem o desenvolvimento de qualquer tipo de aplicação.

Alguns conceitos foram discutidos e/ou definidos por nomes como Coad, Yourdon e Pressman, e reportam aos seguintes itens:

- a orientação a objetos é uma tecnologia para a produção de modelos que especifiquem o domínio do problema de um sistema;
- quando construídos corretamente, sistemas orientados a objetos são flexíveis a mudanças, possuem estruturas bem conhecidas e provêm a oportunidade de criar e implementar componentes totalmente reutilizáveis;
- modelos orientado a objetos são implementados convenientemente utilizando uma linguagem de programação orientada a objetos;
- a orientação a objetos não é só teoria, mas uma tecnologia de eficiência e qualidade comprovadas, usada em inúmeros projetos e para construção de diferentes tipo de sistemas.

A UML é uma tentativa de padronizar a modelagem orientada a objetos de uma forma que qualquer sistema, seja qual for o tipo, possa ser modelado corretamente, com consistência e potencial para se comunicar com outras aplicações.

Existem várias metodologias de modelagem orientada a objetos que precedem o surgimento de UML, a saber:

- Booch - o método de Grady Booch para desenvolvimento orientado a objetos está disponível em muitas versões; Booch definiu a noção de que um sistema é analisado a partir de um número de visões, onde cada visão é descrita por um número de modelos e diagramas;

- OMT - Técnica de Modelagem de Objetos (*Object Modelling Technique*) é um método desenvolvido pela GE (*General Electric*), onde James Rumbaugh trabalhava; o método é especialmente voltado para o teste dos modelos, baseado nas especificações da análise de requisitos do sistema, sendo que o modelo total do sistema baseado no método OMT é composto pela junção dos modelos de objetos, funcional e *use-cases*;
- OOSE/Objectory - Os métodos OOSE e o Objectory foram desenvolvidos com base no mesmo ponto de vista formado por Ivar Jacobson; O método OOSE é a visão de Jacobson de um método orientado a objetos; já o Objectory é usado para a construção de sistemas diversos, sem restrições quanto à natureza; ambos os métodos são baseados na utilização de *use-cases*, que definem os requisitos iniciais do sistema, vistos por um ator externo.

Cada um destes métodos possui sua própria notação (seus próprios símbolos para representar modelos orientado a objetos), processos (que atividades são desenvolvidas em diferentes partes do desenvolvimento), e ferramentas (as ferramentas CASE que suportam cada uma destas notações e processos).

UML tem potencial para dominar o campo da modelagem de sistemas e ambientes ao nível da indústria, sendo uma linguagem baseada em conceitos e padrões consolidados e que apresenta recursos para especificação de construções semânticas genéricas.

3.3.1 Os diagramas UML e sua adequação a esta proposta

Um dos principais problemas do desenvolvimento de novos sistemas utilizando a orientação a objetos nas fases de análise de requisitos, análise de sistemas e *design* é que não existe uma notação padronizada e realmente eficaz que abranja qualquer tipo de aplicação que se deseje construir.

Cada simbologia existente possui seus próprios conceitos, gráficos e terminologias, resultando, muitas vezes, em modelos confusos, especialmente para aqueles que querem utilizar a orientação a objetos não somente para visualizar o sentido de um fluxo em um relacionamento entre entidades, mas para criar modelos de qualidade para ajudá-los a construir e manter sistemas e ambientes cada vez mais eficazes. Neste contexto, metodologias como a UML surgiram para suportar a modelagem de sistemas e ambientes computacionais (Jacobson et al, 1999).

Em linhas gerais, um sistema é composto por diversos aspectos: funcional (que é sua estrutura estática e suas interações dinâmicas), não funcional (requisitos de tempo, confiabilidade, desenvolvimento) e aspectos organizacionais (organização do trabalho, mapeamento dos módulos de código). Assim, um sistema é descrito em um certo número de visões, cada uma representando uma projeção da descrição completa e mostrando aspectos particulares do sistema. Cada visão é descrita por um número de diagramas que contém informações que dão ênfase aos aspectos particulares do sistema. Em alguns casos, pode haver sobreposição entre os diagramas, o que indica que o diagrama pode fazer de duas ou mais visões.

Genericamente, as visões que compõem um sistema são:

- visão "use-case": descreve a funcionalidade do sistema desempenhada pelos atores externos do sistema (usuários);
- visão lógica: descreve como a funcionalidade do sistema será implementada;
- visão de componentes: é uma descrição da implementação dos módulos e suas dependências;
- visão de concorrência: trata a divisão do sistema em processos e processadores;
- visão de organização: mostra a organização física do sistema, os computadores, os periféricos e como eles se conectam entre si.

No contexto do desenvolvimento de aplicações distribuídas, o ambiente da WWW oferece a infra-estrutura básica para suportar tanto a criação quanto o gerenciamento de sessões de trabalho cooperativo (Bernes-Lee et al., 1994). Neste ponto, é importante considerar as propriedades do ambiente no qual a aplicação está inserida. Essas propriedades referem-se às operações associadas ao sensoriamento e à descoberta de informações contextuais, permitindo o controle da apresentação de informações e serviços ao usuário e a execução automática de ações no ambiente.

Considerando a existência de objetos de mídia em um ambiente com consciência de contexto, a manipulação eficiente desses objetos torna-se necessária, especialmente em termos do armazenamento e recuperação de *streams* e dados associados. Desta forma, tecnologias devem ser associadas às tarefas de manipulação de mídias e ao controle do contexto do usuário, da aplicação e do ambiente.

Essas tecnologias podem ser mapeadas com base no paradigma da orientação a objetos, considerando a existência de três camadas, conforme ilustra a **Figura 3.03**.

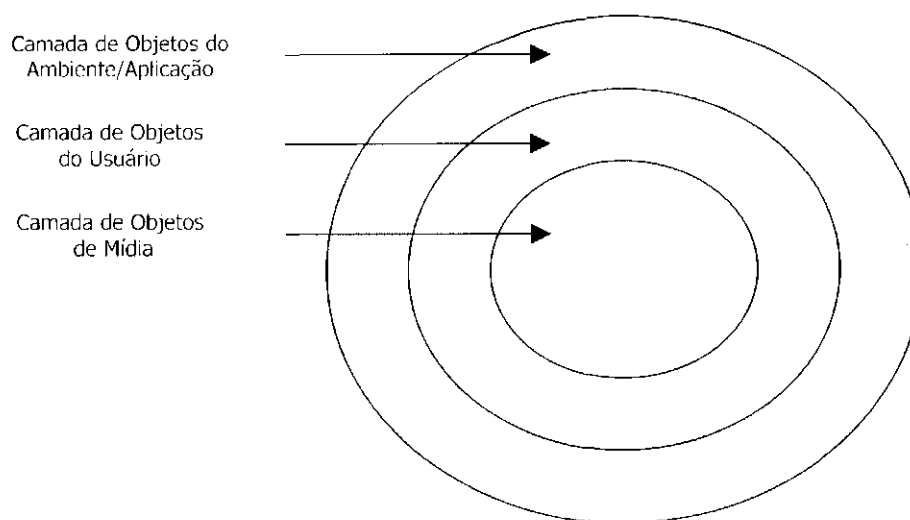


Figura 3.03 – Camadas de objetos em um ambiente com consciência de contexto

A **Figura 3.03** mostra que a camada de objetos do ambiente é a mais externa e afeta as outras duas camadas. Neste ponto, é importante observar que uma classe genérica de objetos deve ser definida nesta camada e mapeada para classes específicas de objetos, gerando instâncias em termos de tecnologias que podem ser utilizadas na implementação de aplicações para o ambiente.

Neste contexto, uma linguagem de modelagem, como é o caso da UML, permite a especificação formal dessas classes de objetos e seus relacionamentos em termos das atividades (ações) que são geradas quando o usuário interage com o ambiente.

Analisando os aspectos de consciência de contexto, é relevante verificar que propriedades como atributos, comportamento e relacionamentos (agregação, especialização e herança, por exemplo) podem ser mapeadas para o paradigma da orientação a objetos. Desta forma, é possível que classes de objetos de um ambiente com consciência de contexto sejam passíveis de mapeamento para tecnologias, protocolos, esquemas de armazenamento/recuperação e linguagens, tais como XML, MPEG-4, MPEG-7, MPEG-J, JAVA, dentre outros.

Genericamente, elementos da simbologia UML podem ser transformados em ações, atributos, condições, eventos, métodos (operações) e código, a partir do uso de um conjunto de diagramas de modelagem.

Na modelagem de um ambiente com a linguagem UML, pode-se utilizar uma seqüência de nove diagramas: diagrama de *use-case*; diagrama de classes; diagrama de objetos; diagrama de estados; diagrama de seqüência; diagrama de colaboração; diagrama de atividade; diagrama de componentes; e o diagrama de execução.

Nas **Subseções 3.3.2 a 3.3.7** é apresentada uma breve descrição de cada um dos diagramas UML aplicáveis a esta proposta.

3.3.2 O diagrama *use-case*

O diagrama *use-case* é uma técnica usada para descrever e definir os requisitos funcionais de um sistema, sendo que um requisito funcional engloba os atores externos ao sistema modelado. Os atores representam o papel de uma entidade externa ao sistema, como é o caso de um usuário ou um outro sistema que interage com o sistema modelado. Os atores iniciam a comunicação com o sistema através de *use-cases* (casos de uso), que representam uma seqüência de ações executadas pelo sistema a partir da necessidade do ator.

Genericamente, atores e *use-cases* são classes. Um ator é conectado a um ou mais *use-cases* através de associações, e tanto atores quanto *use-cases* podem possuir relacionamentos de generalização que definem um comportamento comum de herança.

O uso de *use-cases* em tarefas colaborativas é muito importante, por apresentar as interações necessárias para a execução de uma tarefa específica no sistema. Uma colaboração é descrita por diagramas de atividades e um diagrama de colaboração.

3.3.3 O diagrama de classes

O diagrama de classes demonstra a estrutura estática das classes de um sistema, no qual essas representam as "coisas" que são gerenciadas pela aplicação modelada.

Uma classes pode se relacionar com outras classes através de diversas maneiras:

- através de associação (conectadas entre si);
- através de dependência (uma classe depende ou usa outra classe);
- pela especialização (uma classe é uma especialização de outra classe);
- em pacotes (classes agrupadas por características similares).

Todos estes relacionamentos são mostrados no diagrama de classes juntamente com as suas estruturas internas, que são os atributos e operações. O diagrama de classes é considerado estático, uma vez que a estrutura descrita é sempre válida em qualquer ponto do ciclo de vida do sistema.

Uma classe presente em um diagrama pode ser implementada utilizando-se uma determinada tecnologia que suporte a orientação a objetos de forma direta ou indireta.

3.3.4 O diagrama de colaboração

Um diagrama de colaboração mostra, de maneira semelhante ao diagrama de seqüência, a colaboração dinâmica entre os objetos. Normalmente, pode-se escolher entre utilizar o diagrama de colaboração ou o diagrama de seqüência.

No diagrama de colaboração, além de mostrar a troca de mensagens entre os objetos, percebe-se também os objetos com os seus relacionamentos. Se a ênfase do diagrama for baseada na linha do tempo, é melhor escolher o diagrama de seqüência; mas se a ênfase for o contexto do sistema, é melhor dar prioridade ao diagrama de colaboração.

O diagrama de colaboração é desenhado como um diagrama de objeto, no qual os diversos objetos são mostrados juntamente com seus relacionamentos. As setas de mensagens são desenhadas entre os objetos para mostrar o fluxo de mensagens entre eles. As mensagens são nomeadas e, dentre outras propriedades, mostram a ordem em que as mensagens são enviadas. Também podem mostrar condições, interações, valores de resposta, dentre outros.

3.3.5 O diagrama de seqüência

Um diagrama de seqüência mostra a colaboração dinâmica entre os vários objetos de um sistema, sendo que o mais importante aspecto deste diagrama é que a partir dele percebe-se a seqüência de mensagens enviadas entre os objetos.

O diagrama de seqüência consiste de um número de objetos apresentados em linhas verticais, que representam uma linha do tempo. As mensagens enviadas por cada objeto são simbolizadas por setas entre os objetos que se relacionam. Os diagramas de seqüência possuem dois eixos: o eixo vertical, que mostra o tempo; e o eixo horizontal, que mostra os objetos envolvidos na seqüência de uma certa atividade. No eixo horizontal estão os objetos

envolvidos na seqüência. Cada objeto é representado por um retângulo (similar ao diagrama de objetos) e uma linha vertical pontilhada chamada de linha de vida do objeto, indicando a execução do objeto durante a seqüência.

A comunicação entre os objetos é representada com uma linha com setas horizontais, simbolizando as mensagens entre as linhas de vida dos objetos. A seta especifica se a mensagem é síncrona, assíncrona ou simples. As mensagens podem possuir também números seqüenciais para tornar mais explícita a seqüência no diagrama. Em alguns sistemas, objetos agem de forma concorrente, cada um com sua linha de execução (*thread*).

3.3.6 O diagrama de estado

O diagrama de estado é tipicamente um complemento para a descrição das classes, mostrando todos os estados possíveis em que objetos de uma certa classe podem se encontrar, além de apresentar os eventos do sistema que provocam tais mudanças de estado.

Os diagramas de estado não são escritos para todas as classes de um sistema, mas apenas para aquelas que possuem um número definido de estados conhecidos e onde o comportamento das classes é afetado e modificado pelos diferentes estados.

Diagramas de estado capturam o ciclo de vida dos objetos, subsistemas e sistemas. Eles mostram os estados que um objeto pode possuir e como os eventos (mensagens recebidas, *timer*, erros, condições) afetam estes estados ao passar do tempo.

Ao nível da notação, os diagramas de estado possuem um ponto de início e vários pontos de finalização. Um ponto de início (estado inicial) é mostrado como um círculo todo preenchido, e um ponto de finalização (estado final) é mostrado como um círculo em volta de um outro círculo menor preenchido. Um estado é mostrado como um retângulo com cantos arredondados. Entre os estados estão as transições, mostradas como uma linha com uma seta no final de um dos estados. A transição pode ser nomeada com o seu evento causador. Quando o evento acontece, a transição de um estado para outro é executada ou disparada.

Uma transição de estado normalmente possui um evento associado a ela. Se um evento é anexado a uma transição, esta será executada quando o evento ocorrer. Se uma transição não possuir um evento associado a ela, a mesma ocorrerá quando a ação interna do código do

estado for executada (se existirem ações internas como entrar, sair ou outras ações definidas pelo desenvolvedor). Então, quando todas as ações forem executadas pelo estado, a transição será disparada e serão iniciadas as atividades do próximo estado no diagrama de estados.

3.4 MPEG-4 e MPEG-7: justificativas para o uso desses padrões

Conforme citado, é crescente o interesse por novas tecnologias de *video stream*, principalmente ao nível das grandes indústrias provedoras de tecnologias multimídia, como é o caso da Apple (QuickTime), da Microsoft (Windows Media) e da Real Networks (RealMedia).

Por outro lado, os ambientes de *broadcasting* ainda necessitam de um padrão que seja capaz de unificar a programação e a difusão de multimídia, por exemplo via Internet (Finseth, 1998).

Muitos editores de conteúdo multimídia têm considerado os usuários de PDA, telefones celulares, dispositivos móveis de acesso à Internet e *set-top boxes* como um novo e vasto mercado a ser explorado. No entanto, as múltiplas tecnologias multimídia existentes e as múltiplas especificações para vários tipos de dispositivos constituem um obstáculo para o desenvolvimento e suporte às aplicações. Adicionalmente, é interessante pensar que um mecanismo simplificado e adaptável para produção e distribuição de conteúdo multimídia poderá ser interessante para uma comunidade tão diversamente configurada, permitindo que se atinja aspectos como escalabilidade e adaptabilidade, por exemplo em relação à largura de banda disponível e à qualidade das mídias utilizadas. Neste sentido, MPEG-4 surge como uma possibilidade para integração dos processos de produção e distribuição, constituindo-se no mais recente esforço do *Moving Picture Experts Group*.

Recentemente, acreditava-se que propostas como Windows Media Technologies, QuickTime e RealMedia seriam suficientes para cobrir a produção e distribuição de multimídia. No entanto, o pensamento da indústria era voltado somente para o contexto da Internet. Atualmente, existe um crescente interesse por HDTV (*High Digital TeleVision*) e, conseqüentemente, pela convergência entre Internet e televisão digital (Kate, 1998). Neste contexto, as potencialidades de padrões como MPEG-2 e MPEG-1 Layer-3 (MP3) foram revistas (ISO, 1994), permitindo que os padrões MPEG-4 e MPEG-7 fossem também potencializados quanto ao uso num cenário de televisão digital e Internet, por exemplo.

Em decorrência da revisão de pensamento, indústrias como a Apple, Cisco, Philips e Sun iniciaram altos investimentos no campo da televisão digital, sendo que o primeiro obstáculo enfrentado foi escolher um padrão que suportasse os requisitos para *streaming* de áudio e vídeo em redes IP. Neste ponto, o padrão MPEG-4 foi identificado como a melhor solução existente, devido, principalmente, ao modelo de orientação a objetos.

Fundamentalmente, observa-se que os serviços baseados em *streaming* ultrapassam os limites do “mundo IP”, atingindo territórios como a comunicação *wireless*, broadcasting e as comunicações via cabo e satélite. Adicionalmente, com avanço dos trabalhos em nível científico e industrial, e também com a evolução dos requisitos em torno do cenário de integração da televisão com a Internet, observou a necessidade de uso de um protocolo capaz de fornecer possibilidades de negociação entre as diversas camadas e mídias que compõem uma determinada aplicação. Esses requisitos identificados vão de encontro às características de projeto do MPEG-4, contrapondo-se ao uso de tecnologias proprietárias para *streaming*. Por outro lado, ainda existem os problemas relativos à qualidade do processo de *streaming* e também em relação ao processo de licenciamento para uso, visto que o referido padrão é relativamente novo.

3.4.1 A solução MPEG-4

Para compreender o escopo da solução MPEG-4 é interessante conhecer os princípios a partir dos quais a solução foi proposta. Para tal, pode-se considerar um exemplo baseado no problema da produção e distribuição de conteúdo por uma rede de televisão, que deseja também distribuir esse mesmo conteúdo multimídia via Internet. Neste sentido, alguns argumentos devem ser considerados:

- a rede de televisão tem um público que recebe a programação através de meios de comunicação de banda larga, como é o caso do sistema de antenas, cabos e satélite, nos quais a qualidade de transmissão é considerada ótima e/ou excelente;
- a rede de televisão fornece também o mesmo conteúdo em um *web site*, através de *streaming* em tempo real via Internet; neste caso, a qualidade da transmissão é razoável ou mesmo inaceitável em muitos casos, dependendo diretamente das características da conexão do usuário, especialmente em termos da velocidade;
- uma conexão a 28,8 Kbps não oferece possibilidades para uma boa transmissão de vídeo e apenas uma qualidade razoável para a transmissão de áudio.

Neste cenário, pode-se pensar também que na programação da emissora existem programas como aqueles que apresentam cotação das bolsas de valores e classificação de equipes em eventos esportivos, tais como campeonatos de futebol, corridas de fórmula 1, dentre outros. Para esses casos, a informação que se deseja transmitir é basicamente textual, que pode ser visto com boa qualidade tanto no ambiente televisivo quanto na Internet. No entanto, também esse tipo de informação pode sofrer alterações na qualidade da apresentação, visto que os vários tipos separados de mídia (áudio, vídeo e texto) são misturados em um único fluxo de informação, antes de serem enviados. Assim, tem-se a degradação de todo o conteúdo transmitido quando se usa uma conexão com pouca largura de banda.

Ao nível das aplicações, observa-se que o problema reportado impede a implementação de serviços como o *e-commerce*, visto que não existe um modo para transformar um elemento de mídia e um *hyperlink*, por exemplo. Adicionalmente, seria interessante, em termos da programação da emissora, adicionar um arquivo de informações das empresas às cotações da bolsa, por exemplo; ou, ainda, permitir ao usuário saltar as tabelas de classificação dos eventos esportivos e acessar diretamente os detalhes sobre as equipes e/ou atletas. Observa-se que o impedimento para o desenvolvimento dessas aplicações reside justamente no fato de que os elementos de mídia não são transmitidos como elementos individuais (*Elementary Streams*). Adicionalmente, não é possível também atingir a distribuição para dispositivos *wireless*, celulares, set-top boxes, dentre outros, sem ao menos produzir um *stream* específico para cada um desses terminais de recepção e interação. Porém, neste caso, conforme citado, cada proprietário dessas soluções deveria ter a liberdade para acessar um URI e manipular o conteúdo de acordo com os seus próprios procedimentos de gerenciamento do conteúdo multimídia (Bernes-Lee et al., 1998).

Neste sentido, uma solução é organizar um processo produtivo que permita a criação de um único *stream* padrão que possa ser adaptável ao maior número possível de usuários e atendendo aos requisitos de qualidade especificados. Esta foi, em suma, a proposta de projeto que deu origem ao MPEG-4.

3.4.2 Diferenças entre um codificador de mídias e uma arquitetura

Para medir a potencialidade da tecnologia MPEG-4 é importante compreender as diferenças que existem entre um codificador de mídias (codec) e uma arquitetura.

O termo *codec* (acrônimo para compressão/descompressão) se refere a qualquer algoritmo utilizado para comprimir e descomprimir fluxos de dados que formam uma mídia digital, visando obter maior eficiência no armazenamento (por exemplo, em discos como o CD-ROM), e também na transmissão em redes de comunicação. Vale ressaltar que existe um outro tipo de *codec* que se refere aos mecanismos utilizados para converter sinais analógicos em digitais e vice-versa, porém não é o foco dessa abordagem.

Por outro lado, uma arquitetura é um conceito muito mais amplo, identificando os formatos das mídias suportadas e definindo os mecanismos de sincronização em um *stream*. Adicionalmente, é responsável por descrever a estrutura dos arquivos comprimidos, as regras para transmissão dos *streams* e os protocolos de comunicação entre o servidor e o terminal do usuário.

Assim como para a arquitetura proposta para o MPEG-4, as tecnologias QuickTime, RealMedia e Windows Media suportam vários *codecs*, porém essas arquiteturas proprietárias focalizam principalmente as mídias utilizadas, a rede IP e o computador (enquanto elemento processador e terminal de exibição). Por outro lado, a arquitetura MPEG-4 suporta uma vasta gama de meios de transmissão e dispositivos de recepção. Adicionalmente, visto que MPEG-4 foi criado a partir do trabalho de um organismo internacional de padronização, no caso a ISO, envolvendo representantes de todos os setores interessados da indústria, o padrão é isento dos problemas decorrentes do desenvolvimento de uma solução proprietária.

Do ponto de vista do usuário, no entanto, o *codec* é mais importante que a arquitetura. Pode-se dizer que nenhum usuário vê ou escuta a arquitetura. Neste sentido, os algoritmos *codec* para vídeo do MPEG-4 não estão ainda ao nível dos algoritmos utilizados por soluções como da Real Networks e Microsoft Windows. Adicionalmente, a Apple também está trabalhando no desenvolvimento de um novo *codec* QuickTime, denominado On2, que promete significativas melhorias em relação ao *codec* atualmente utilizado. Assim, a arquitetura MPEG-4 é muito promissora, porém os aspectos relativos aos *codecs* necessitam ainda de aprimoramento.

3.4.3 Orientação a objetos

Em um *stream* MPEG-4, todos os elementos de mídia, como é o caso dos conteúdos de áudio e vídeo, imagens estáticas e informações textuais (como no exemplo das cotações das bolsas

de valores e competições esportivas) são mantidos como objetos independentes uns dos outros.

Em linhas mais avançadas, os componentes de um *stream* de vídeo também podem ser tratados separadamente.

Considerando o exemplo da emissora de televisão, um programa para apresentação da previsão do tempo pode ser gerado e representado como diversos objetos de vídeo. Assim, pode-se gerar as cenas da previsão do tempo com um determinado objeto que represente o fundo (*background*) para a apresentação e, posteriormente, substituir esse fundo por um outro objeto que represente, por exemplo, um mapa de uma determinada região, com animações e outras personalizações apropriadas ao caso. Em um *stream* MPEG-4, todos esses elementos de mídia podem ser representados como objetos independentes.

De forma análoga, durante os programas esportivos e de informações financeiras, exemplificados anteriormente, o apresentador e os textos informativos podem ser representados como objetos distintos e independentes. Esta abordagem da orientação a objetos apresenta dois benefícios fundamentais:

- primeiro, é que cada elemento pode ser tão interativo quanto necessário, permitindo ao usuário, por exemplo, acionar controles de pausa ou acelerar a apresentação, selecionar *links* de interesse, eliminar o objeto *apresentador* de uma determinada cena, e outras ilimitadas possibilidades de interação;
- segundo, é cada *stream* pode ser personalizado para vários tipos de conexão e dispositivos de recepção.

Neste contexto, um usuário conectado via DSL, por exemplo, poderia receber todos os elementos de um conteúdo multimídia, enquanto um usuário que verifica as condições meteorológicas usando um PDA poderia receber somente os objetos referentes ao mapa da previsão do tempo, alguma animação das condições climáticas e algumas faixas de áudio adequadas a condições de pouca largura de banda.

Ainda em termos de aplicação e interatividade, um usuário interessado nos dados das cotações das bolsas de valores poderia ouvir em seu celular o conteúdo das informações

textuais, isto após um processo de conversão dos objetos de texto em objetos de áudio, usando um dos componentes da arquitetura MPEG-4.

Em linhas gerais, observa-se que essas possibilidades de interação contribuem para colocar o MPEG-4 em contraste com as limitações das tecnologias de *streaming* oferecidas pelas soluções proprietárias e existentes atualmente. Essas tecnologias, por exemplo, se limitam a eliminar *frames* de vídeo ou a utilizar um *codec* de áudio de baixa qualidade para atingir alguns requisitos de transmissão em uma rede de comunicação.

Em termos de produção, usando MPEG-4, a indústria de *broadcasting* e os profissionais *broadcasters* podem reduzir sensivelmente os custos de produção e de armazenamento, uma vez que é possível criar apenas um único *stream* de dados multimídia e aplicar mecanismos inteligentes sobre esse *stream* para personalizar a distribuição e a apresentação.

3.4.4 Alguns detalhes da arquitetura MPEG-4

MPEG-4 é o primeiro padrão da família MPEG definido a partir do paradigma da orientação a objetos. Desta forma, novas tecnologias e mecanismos foram definidos para compor a arquitetura MPEG-4.

O primeiro desses mecanismos e tecnologias são os BIFS (*Binary Format For Scenes*) que é a linguagem especificada para descrever como os diversos objetos fazem parte de um cena. A linguagem BIFS foi inspirada em VRML, porém os arquivos BIFS podem ser binários ou textuais, enquanto arquivos VRML são essencialmente textuais. Neste ponto, usando BIFS é possível a reprodução e apresentação durante o processo de *streaming* de uma cena, o que é uma necessidade absoluta para o ambiente da Internet. Por outro lado, usando VRML, a reprodução e apresentação da cena ocorre somente depois que se completa todo o *download* dos componentes da cena.

Na arquitetura MPEG-4, cada objeto é encapsulado no seu respectivo *stream* ou em diferentes *streams*, dependendo das necessidades da aplicação. Cada *stream* individual é denominado *Elementary Stream*. Cada *elementary stream* contém as informações necessárias sobre o *decoder* requisitado e também detalhes sobre a qualidade de serviço especificada, tais como o *bit rate* mínimo necessário para transmissão. Um outro *elementary stream* contém as informações no

formato BIFS, que descrevem as relações temporais para garantir o sincronismo entre os objetos durante a apresentação.

Do ponto de vista da transmissão, a arquitetura MPEG-4 inclui dois tipos de multiplexadores que permitem a transmissão dos *streams* usando os mecanismos de transporte existentes, tais como RTP (*Real Time Protocol*) – usado na Internet ou ATM – e MPEG-2 – usado na difusão via cabo ou via satélite, por exemplo. O primeiro multiplexador, denominado FlexMux e definido no DMIF (*Delivery Multimedia Integration Framework*), define a comunicação entre o servidor MPEG-4 e o terminal capaz de reproduzir as cenas MPEG-4, negociando os níveis de QoS e ajustando os *elementary streams* para as condições de transmissão. O segundo multiplexador, denominado TransMux, é responsável pela interface com o *stream* que transporta o conteúdo multimídia e pela transmissão correta dos dados, permitindo que *streams* MPEG-4 possam ser transportadas via *broadcast*, *wireless* e Internet, ampliando, significativamente, a cobertura da arquitetura em termos de mercado.

Neste sentido, os mecanismos derivados dos multiplexadores oferecem ao MPEG-4 significativas vantagens sobre tecnologias como QuickTime (que foi o ponto de partida para a construção do formato MPEG-4) e SMIL (*Synchronized Multimedia Integration Language*), padronizado pela W3C. Em linhas gerais, observa-se nenhum desses dois formatos oferece os níveis de negociação presentes na arquitetura MPEG-4, o que não permite o uso de seus recursos em outros meios que não a Internet.

3.4.5 Algumas vantagens e desvantagens da arquitetura MPEG-4

Além dos fundamentos principais do MPEG-4, a arquitetura oferece também possibilidades para funções de animação facial que, quando combinada com interfaces de texto e voz, permitem a construção de cenas baseadas em “imagens falantes”, que abrem um interessante espectro de aplicações no “mundo virtual”, principalmente por utilizar pouca banda passante.

Outro ponto importante da arquitetura MPEG-4 é que os algoritmos para compressão de imagens estáticas são extremamente otimizados e apresentam resultados semelhantes aos algoritmos JPEG. Por outro lado, ainda persiste o problema com o *codec* de vídeo MPEG-4, que não tem produzido resultados satisfatórios, sendo muito inferior às soluções proprietárias da Microsoft e Real Networks. No caso específico de vídeo, a Real Networks, ao

apresentar o *codec* Real Video 8 (que demonstrou alto desempenho), propôs que o produto fosse inserido integralmente como parte do padrão de vídeo MPEG-4 e nas futuras extensões.

3.5 Considerações finais

De forma genérica, os ambientes interativos são considerados “ambientes enriquecidos” por utilizarem objetos de mídia de diferentes naturezas. Muitas podem ser as formas de composição das cenas multimídia a partir desses objetos de mídia. Em termos de infraestrutura de comunicação, pode-se definir vários níveis de qualidade de serviço. Ao nível do usuário, muitas são as formas de interação com as cenas em apresentação, inclusive explorando recursos sensoriais e informações contextuais.

Neste capítulo, foi discutida a necessidade da modelagem de ambientes interativos e foram apresentados alguns dos métodos, técnicas e metodologias que podem ser aplicados à modelagem de aplicações multimídia interativas. A linguagem UML foi detalhada em seus diagramas e justificada quanto ao uso na modelagem proposta por este trabalho. Uma reflexão sobre o uso de padrões, no contexto da modelagem, foi apresentada e discutida, especialmente em termos dos padrões MPEG-4 e MPEG-7.

No próximo capítulo, inicia-se a discussão da modelagem proposta por este trabalho para o caso dos ambientes interativos conscientes de contexto. Essa discussão parte da definição do que é contexto, consciência de contexto e suas relações com aplicações multimídia em ambientes interativos.

4.1 Considerações iniciais

Em meio à revolução digital que ocorre atualmente, a distribuição de multimídia é cada vez mais um foco importante na área tecnológica, quer seja no meio acadêmico quer seja na indústria.

Com o surgimento das tecnologias para multimídia interativa, um vasto número de aplicações tornou-se possível. As diferentes possibilidades de interação do usuário com os objetos de mídia que compõem o vídeo trazem consigo a necessidade de modelagem dessa interação em direção aos aspectos de adaptação e personalização da interface. Adicionalmente, a convergência de mídias e entre os meios de comunicação tende a inserir o usuário em um ambiente cada vez mais interativo e esse usuário passa a fazer parte desse ambiente, juntamente com todos os componentes de hardware e software da aplicação, também inserida como componente desse ambiente.

Neste capítulo, discute-se uma possibilidade para formalização de aspectos de consciência de contexto, como parte da modelagem proposta por este trabalho, considerando que informações contextuais podem ser extremamente úteis para a oferta de serviços personalizados nesse novo e potencial universo de aplicações multimídia interativas.

4.2 Consciência de contexto e suas relações com interação do usuário

Reconhecidamente, a interface com o usuário é um ponto cada vez mais importante à medida que o uso da tecnologia, especialmente do computador, cresce consideravelmente.

O desenvolvimento de interfaces usuário-computador é altamente criativo e diferentes projetistas usam diferentes técnicas. Por envolver um projeto multidisciplinar, várias formas de “enxergar o mundo” são aplicadas, sendo de extrema importância o foco centrado nas tarefas do usuário - “como o usuário trabalha e o que é importante para ele” (Pressman, 1995).

O ser humano possui um sistema perceptivo sensorial e paralelo. Uma especificação adequada da comunicação visual é um elemento chave de uma interface amigável. De forma geral, as informações são armazenadas no cérebro em uma memória STM (*Short Time Memory*) - que podem ser reusadas imediatamente - ou em uma memória LTM (*Long Time Memory*), onde é formado o conhecimento (Preece, 1994). Além disso, um usuário utiliza heurísticas (diretrizes, regras e estratégias) que influenciam uma informação ou conhecimento em um certo contexto da interação.

A abordagem cognitiva tradicional enfoca a utilização de cenários de execução de tarefas, nos quais deve-se enxergar o ser humano como um ator capaz de controlar as tarefas e o ambiente no qual essas tarefas estão inseridas (Shneiderman, 1998). Um sistema baseado em computador é usado para automatizar tarefas que, em suma, são formadas por eventos e relacionamentos, existindo tarefas genéricas, tais como tarefas de comunicação, tarefas de diálogo, tarefas cognitivas e tarefas de controle, que devem ser consideradas no projeto de quaisquer interações usuário-computador (Preece, 1994).

No atual estado da arte, observa-se que a interação usuário-computador tem se tornado cada vez mais complexa, em termos de análise de requisitos, e transparente e natural para os usuários.

Tecnologias e técnicas avançadas de interação têm sido criadas para minimizar o uso de dispositivos tradicionais de interação como mouse e o teclado, permitindo que o usuário interaja diretamente com o ambiente no qual a aplicação está inserida. No entanto, na criação de ambientes computacionais nos quais a aplicação possa ter a “liberdade” de sugerir ou tomar certas decisões, é necessário o uso de informações contextuais do usuário, da aplicação e do ambiente como um todo.

Segundo Abowd, um sistema é consciente de contexto se utiliza o contexto para oferecer informações relevantes e/ou serviços ao usuário, sendo que a relevância depende das tarefas que o usuário realiza (Abowd, 1999).

Em linhas gerais, entende-se por contexto como sendo qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Uma entidade, por sua vez, pode ser uma pessoa, um lugar ou objeto relevante para a interação entre um usuário e uma aplicação,

incluindo-se também o usuário e a própria aplicação (Dey & Abowd, 1999). Adicionalmente, parâmetros do comportamento humano podem caracterizar a consciência de contexto, especialmente quanto à definição das visões de contexto (superficial ou abrangente) que se deseja extrair de um ambiente.

Ao nível da aplicação, pode-se citar duas maneiras de se obter informações de contexto. Na primeira, um usuário fornece informações explicitamente, tais como um nome de usuário e uma senha em um procedimento de autenticação/autorização. Na segunda, mais complexa, o usuário informa implicitamente as informações de contexto para a aplicação, sendo função da aplicação a captura dessas informações de contexto geradas por ações do usuário.

A criação de ambientes que utilizem as informações de contexto para fornecer informações relevantes e/ou ações específicas para cada usuário, conforme o grau de liberdade definido pelo projetista, é uma tarefa que precisa ser parametrizada pela aplicação. Esta parametrização é obtida com base em cinco questionamentos:

- onde o usuário está ?
- quem é o usuário ?
- quais recursos estão em uso ?
- quando se pode realizar determinada operação ?
- o que o usuário está fazendo ?

A partir dos questionamentos citados, pode-se obter as seguintes categorias de contexto:

- localização;
- identidade;
- atividade;
- tempo.

Observa-se, então, que através da categorização é possível caracterizar a situação de uma entidade, bem como obter informações sobre o comportamento dessa entidade no tempo (Dey & Abowd, 1999). Assim, a partir das informações de contexto, uma aplicação pode realizar ações/reações como as seguintes:

- a apresentação de informações e serviços para o usuário, como por exemplo a apresentação de mídias disponíveis e que estão relacionadas com o contexto do usuário;

- a execução automática de serviços, como por exemplo o disparo de eventos relacionados com a localização corrente de uma entidade;
- a associação do contexto com a informação para posterior recuperação, como por exemplo o armazenamento de informações contextuais durante uma sessão de *chat* em uma sala de aula virtual, relacionando observações de alunos e professores (tutores) com o assunto discutido em uma aula.

As definições de consciência de contexto oferecem condições para o desenvolvimento de aplicações altamente adaptativas, o que é relevante em ambientes multimídia distribuídos, especialmente em termos das ações e reações, tanto ao nível do usuário quanto ao nível da aplicação e do ambiente. Neste ponto, o uso de dispositivos sensoriais, o uso de mecanismos e recursos para descoberta de informações e associação de dados digitais ao contexto do usuário são extremamente importantes em ambientes com consciência de contexto, viabilizando a apresentação personalizada de conteúdos, a execução automática de serviços e o armazenamento/recuperação de informações contextuais em diversos níveis de interesse.

O grau de consciência (*awareness*) de um objeto em relação a outro quantifica a importância subjetiva ou relevância do objeto para um determinado ambiente ou mesmo para uma determinada mídia utilizada em uma aplicação. Como exemplo, pode-se observar que o grau de consciência pode ser mapeado para o volume de um canal de áudio ou para o nível de detalhe de um *rendering* de uma imagem, sendo que, em linhas gerais, espera-se maior dedicação de processamento aos objetos que possuam valores de consciência mais altos.

Outro ponto importante reside no fato de que um objeto, em uma interação, pode afetar os níveis de consciência de outro objeto, exigindo, portanto, procedimentos de negociação. O objeto que observa (ou receptor) pode focar sua atenção em áreas ou objetos específicos, enquanto o objeto observado (ou transmissor) pode controlar sua “visibilidade” de forma que objetos em algumas áreas sejam mais conscientes que em outras áreas. Um bom exemplo pode ser reportado ao acesso a mídias distintas em uma determinada cena multimídia.

A criação de ambientes interativos com consciência de contexto é um desafio devido aos muitos fatores inerentes à criação de ambientes virtuais, especialmente quando se trata de interação multiusuário em ambientes distribuídos, como é o caso considerado nesta proposta.

4.3 Ubiquidade

A computação ubíqua (*Ubiquitous Computing*) tem sido vista como a próxima era da computação (*age of calm technology*) - é a ciência objetivando a invisibilidade dos recursos computacionais para as pessoas. Neste contexto, invisibilidade não significa desaparecer com os computadores, mas que os recursos computacionais irão, cada vez mais, ser embutidos em objetos do mundo real, permitindo que as pessoas interajam com computadores de uma forma natural.

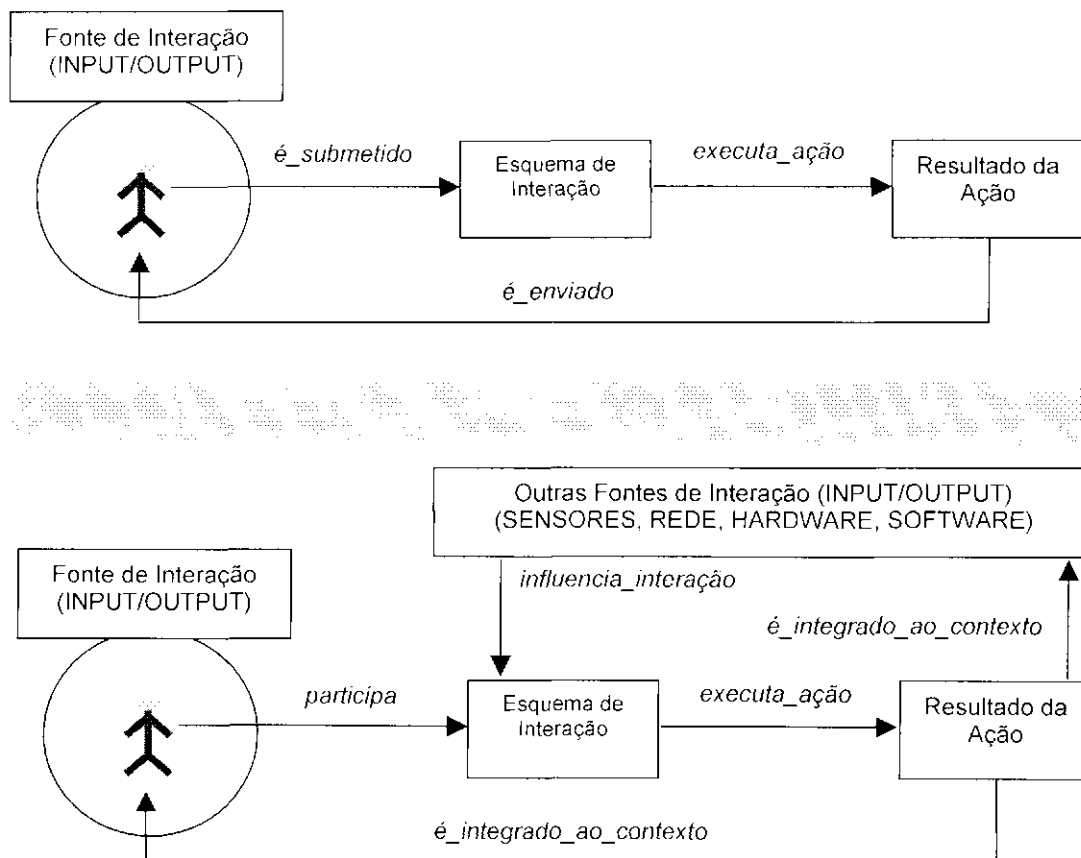


Figura 4.01: Focos de interação do usuário em relação à existência de consciência

É fato que um dos principais fundamentos da tecnologia é auxiliar o ser humano nas suas tarefas do cotidiano (*everyday tasks*), porém, ao longo do tempo, tem-se observado as dificuldades que o homem tem encontrado para se adaptar aos requisitos que determinada tecnologia impõe (*psychopathologies of everyday*), o que, de certa forma, tende a distanciar-lo do mundo real (Cooper, 1997).

A computação ubíqua provê várias das possibilidades para inserção do usuário como componente de um ambiente mais amplo, oferecendo-lhe formas para experimentar o acesso à informação independente da localização geográfica e das formas de produção dessa informação. Além disso, os elementos da computação móvel oferecem formas para captura de alguns dos dados referentes ao contexto no qual o usuário está inserido, tais como a localização geográfica, pessoas e objetos que se encontram próximos ao usuário, abrindo possibilidades concretas para melhoria na interação do usuário com o ambiente.

O vasto número de situações em que o usuário pode estar envolvido acrescenta a necessidade da adaptação a cada uma dessas situações e/ou a grupos formados por algumas dessas situações. Os requisitos de adaptabilidade, portanto, também oferecem possibilidades concretas para melhoria da interação do usuário com o ambiente (McCanne et al., 1996; Dey & Abowd, 2000).

Genericamente, uma forma para melhorar a interação do usuário é através da melhoria da comunicação entre o usuário e o ambiente durante o processo de interação, usando, por exemplo, informações contextuais.

A **Figura 4.01** apresenta um esquema que ilustra a posição do foco de interação do usuário em um ambiente consciente de contexto, em relação a um ambiente não consciente de contexto.

Observa-se que aplicações conscientes de contexto usam os aspectos associados ao contexto para fornecer informações e/ou habilitar serviços ao usuário (Salber, 1999). Adicionalmente, informações podem ser fornecidas ao ambiente como um todo, considerando que também a rede, sensores, hardware e software podem usufruir do grau de consciência de contexto do ambiente. Isto permite o desenvolvimento de adaptabilidade em diversos níveis, e o grau de automatismo pode ser definido de acordo com a aplicação, considerando, por exemplo, as condições de infra-estrutura de cada aplicação.

No caso das aplicações interativas, de forma simplificada, o suporte à consciência de contexto pode ser usado de três formas distintas: a) para controle dos objetos de mídia de uma cena; b) para adaptação dinâmica da aplicação às condições da rede de comunicação; c) para adaptação dinâmica da aplicação ao contexto do usuário.

O controle sobre os objetos de mídia é um aspecto fundamental para aplicações interativas, considerando que novas tecnologias de mídias interativas (vídeo interativo) permitem a composição de cenas multimídia formadas por diferentes objetos e originados de diferentes fontes. Esses objetos têm, naturalmente, o potencial para serem manipulados individualmente e de forma independente em uma cena. Para este caso, esquemas de descrição oferecem a semântica necessária para a construção de cada objeto e também as formas possíveis e permitidas para acesso e manipulação desses objetos. Neste sentido, observando-se as descrições semânticas, é possível obter as ações que podem ser realizadas sobre um determinado objeto e/ou grupo de objetos, e associar essas ações a requisitos de consciência de contexto.

A adaptação dinâmica da aplicação às condições da rede de comunicação e ao servidor de multimídia acrescenta aspectos de adaptabilidade que são muito importantes em aplicações interativas distribuídas. As interfaces de comunicação em rede aplicadas a serviços interativos devem fornecer mecanismos para que o usuário especifique, por exemplo, os parâmetros de qualidade da apresentação das cenas multimídia. Uma vez que o usuário tenha especificado o nível de qualidade desejado, as interfaces de comunicação devem se adaptar a esses novos parâmetros e assegurar que a qualidade desejada seja mantida durante toda a apresentação. Caso algum parâmetro precise ser reajustado, isto deve ocorrer de uma forma transparente ao usuário e também ao contexto do ambiente no qual o usuário está inserido.

A adaptação dinâmica da aplicação ao contexto do usuário é um dos pontos-chave para os serviços personalizados às necessidades do usuário. Existem muitas situações nas quais as interfaces existentes no ambiente podem se auto-adaptar ao contexto do usuário, de forma transparente. Essa adaptação pode usar, por exemplo, informações contextuais, tais como identificação do usuário, localização geográfica, ação ou ações em curso e os parâmetros de tempo (*time_stamp*). De acordo com esses parâmetros, serviços podem ser habilitados e/ou desabilitados a um determinado usuário ou grupo de usuários.

Em termos da cadeia produtiva de aplicações interativas, observa que a adaptação dinâmica da aplicação às condições da rede de comunicação e ao servidor de multimídia pode auxiliar sobremaneira aos provedores de rede e de conteúdo. Por outro lado, o usuário final será o maior beneficiado se o ambiente oferecer condições de adaptação da aplicação ao contexto

do próprio usuário e também um controle eficiente dos objetos que compõem uma aplicação interativa.

4.4 Mobilidade como infra-estrutura para consciência de contexto

No escopo deste trabalho, o conceito de mobilidade pode ser definido como sendo o elemento que fornece suporte à implementação de aplicações que podem ser utilizadas de qualquer ponto geograficamente definido. Num contexto mais restrito, pode-se pensar num exemplo de mobilidade com o uso de protocolos WAP (*Wireless Application Protocol*), através do qual é possível implementar aplicações para acesso a conteúdo de servidores *web* de qualquer ponto geográfico.

Nos últimos anos, diversos equipamentos surgiram (e estão surgindo) com o objetivo de ampliar as fronteiras da mobilidade do usuário, tais como o celular, o PDA (*Personal Digital Assistant*), *wearable computer*, dentre outros. Em adição, existem vários recursos e sistemas que têm por objetivo suportar a mobilidade do usuário, dentre os quais pode-se citar aplicações do tipo *MyDocsOnline*, que permite ao usuário, através de qualquer tipo de conexão com a Internet, acessar os seus documentos e, inclusive, imprimi-los em um terminal de FAX. Outro exemplo é a tecnologia Java que, juntamente com a tecnologia XML, possibilitam a criação de sistemas que usufruem da mobilidade do usuário

Observa-se que o conceito de ubiqüidade está ligado ao conceito de mobilidade, e pretende-se, neste trabalho, apresentar uma visão de como a mobilidade por ser modelada para oferecer suporte à ubiqüidade. Em linhas gerais, tem-se o seguinte problema: se a computação ubíqua parte do princípio que os computadores podem estar espalhados por todas as partes e em diferentes formas de objetos, por que não aproveitar estes recursos para permitir a mobilidade?

Com intuito de responder a essa questão, no enfoque deste trabalho, o modelo de mobilidade passa pelo uso dos requisitos de consciência de contexto (*context-awareness*), que, em suma, referem-se ao uso de dados contextuais (*who, where, when, what, how*) para prover informações relevantes de tarefas ou serviços para um usuário. Conforme visto, entende-se por **contexto** como sendo qualquer informação que pode ser utilizada para caracterizar uma entidade, onde uma entidade pode ser uma pessoa, um lugar ou um ambiente físico ou computacional (Dey & Abowd, 2000).

Observa-se, portanto, que os requisitos de consciência de contexto podem ser aplicados num cenário de ubiquidade e mobilidade na forma de um agente observador do ambiente (*context*) de forma sistêmica, analisando todos os objetos do ambiente tanto de forma independente como em conjunto (relacionamentos). Mudanças no ambiente podem ocorrer a partir da ocorrência de eventos, e os agentes de *context-awareness* (CaA) devem estar habilitados a percebê-las e, desta forma, auxiliar na tomada de decisões. Algumas mudanças são significativas e importantes e fazem com que o próprio CaA acione um evento. Como exemplo, pode-se imaginar o seguinte cenário: uma pessoa está assistindo a um programa com cenas de violência na televisão (contexto inicial); uma criança adentra o recinto (mudança no contexto inicial); um CaA automaticamente dispara um evento para a mudança da programação na televisão, visto que não é recomendável para uma criança assistir a programas violentos. Outras mudanças no contexto não geram grandes transformações e, portanto, o CaA pode gerar recomendações ou mesmo manter o ambiente inalterado.

Em linhas gerais, para que um CaA reconheça mudanças significativas no contexto, é importante estabelecer parâmetros para quantificar e qualificar os requisitos contextuais, bem como as suas mudanças. Estes parâmetros podem ser obtidos através de cinco argumentos:

- WHERE – determina onde uma entidade se encontra dentro do contexto (exemplo: *onde está o usuário?*); pode-se quantificar este argumento utilizando-se da posição geográfica (X,Y ou X,Y,Z) de uma entidade; um novo nível de interação pode ser estabelecido e pode-se pensar na divisão de uma entidade em subentidades (exemplo: *onde está o pé do usuário?*);
- WHO – identifica qual a entidade que está realizando uma determinada ação (exemplo: *quem é esse usuário?*); pode-se obter dados a respeito do usuário através do uso de sensores (íris, altura, temperatura) ou através de processos que identifiquem *log* em um sistema;
- WHEN – identifica, a partir de uma unidade de tempo conhecida, o horário em que uma determinada entidade do contexto realizou uma determinada ação (exemplo: *quando o usuário ligou a entidade televisão?*); pode-se quantificar este argumento através do armazenamento dos dados de tempo (*time-stamp*) associados com a ação;
- WHAT – retorna informações sobre as ações que uma determinada entidade está realizando em um determinado momento (exemplo: *o que o usuário está assistindo*

na televisão?); pode-se quantificar este argumento através dos resultados obtidos por outros argumentos de contexto, tais como WHERE/WHO/WHEN (conhecendo a posição geográfica de duas entidades distintas, pode-se descobrir as relações entre essas entidades; ainda, se uma entidade está se deslocando de um ponto geográfico para outro, e reconhecendo que essa entidade pertence a uma categoria *person*, pode-se listar as atividades permitidas naquele contexto para aquela pessoa);

- HOW – identifica como uma determinada entidade está realizando uma ação, apresentando elementos complementares a essa ação e permitindo a descoberta de relações implícitas ao contexto (exemplo: *como o usuário trabalha?*).

Observa-se que o uso de requisitos de consciência de contexto pode gerar impactos diretos na qualidade de serviço (QoS) em um ambiente. De um ambiente dotado de CaA, pode-se extrair dados que auxiliem a quantificação do grau da qualidade de um serviço, em seus diversos níveis de avaliação. O grau obtido não deve ser visto com um valor fixo, uma vez que variáveis de diferentes naturezas podem influenciar a sua medida, tais como o tráfego da rede, software, o número de usuários, o conteúdo em distribuição, dentre outras.

Num cenário mais abrangente, a expectativa de uso da mobilidade, da consciência de contexto e da ubiquidade tem seu foco central na possibilidade de um usuário ter acesso qualitativo a serviços, independente da sua localização geográfica, usufruindo das regras de transição de contexto (quando ocorrer) que permitam a manutenção de níveis aceitáveis da qualidade de serviço e mecanismos de adaptação natural e gradativa aos meios de comunicação utilizados e aos dispositivos terminais de informação.

Neste sentido, se é possível quantificar e qualificar um contexto, pode-se determinar quais mudanças são realmente significativas. Vale ressaltar que o conceito de mudança significativa é subjetivo, devendo haver mecanismos que permitam a sua configuração para um determinado ambiente e/ou cenário. Por exemplo, para um determinado usuário (WHO), o fato de uma criança entrar em um ambiente que esteja apresentando um filme com cenas de artes marciais pode ser uma mudança de contexto significativa, porém, para outras pessoas isso pode não ser significativo. Ao ocorrer uma mudança significativa no contexto, os CaA podem disparar eventos que promovam ajustes à realidade de cada entidade envolvida no contexto. Por exemplo, quando a criança adentra o recinto, automaticamente o

sistema começa a passar cenas de um desenho animado na entidade televisão, enquanto dispara um evento que continue apresentando o filme em um PDA do adulto. Neste caso, a mudança de contexto pode causar sensíveis modificações nos níveis de QoS, considerando-se as diferenças de propriedades e características entre os terminais envolvidos (terminal de televisão e PDA, por exemplo).

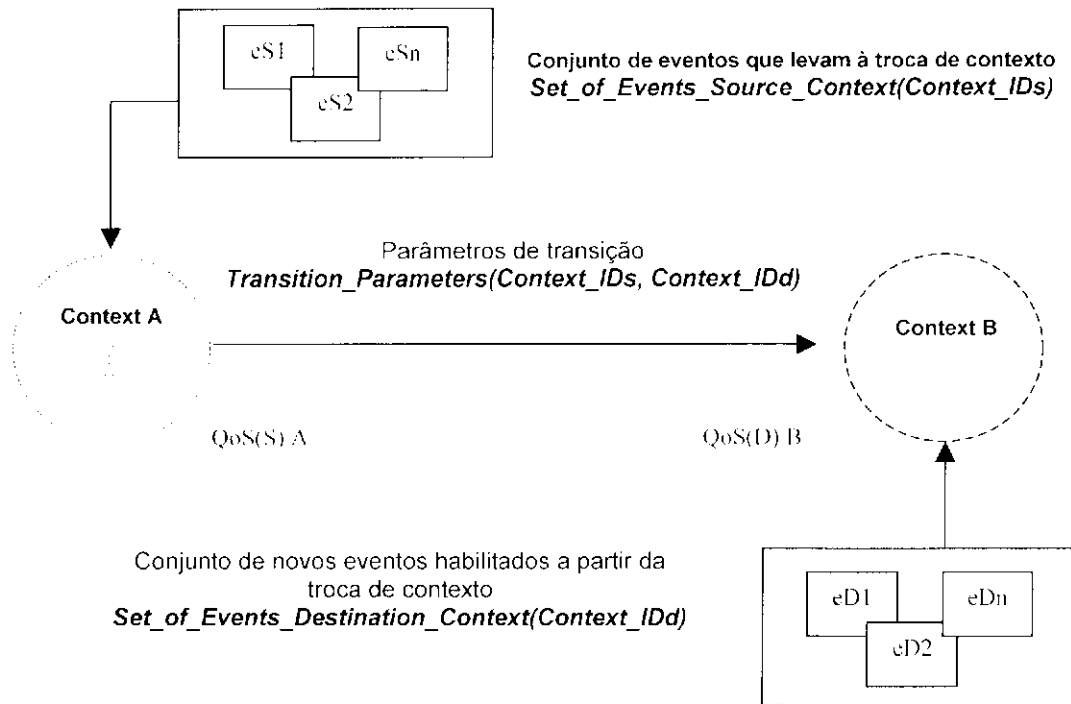


Figura 4.02 - Modelo para uso de consciência associada à mobilidade

Analisando os parâmetros de QoS, quando da troca de contexto, pode-se atingir diversos níveis de avaliação do ambiente como um todo. Com os dois valores de QoS ($QoS(S)$ e o $QoS(D)$), pode-se obter um valor que representa a diferença de qualidade de serviço entre os contextos. Este valor é chamado de Δc e é calculado pela diferença parametrizada entre o $QoS(S)$ e o $QoS(D)$. Pode-se, com isso, estabelecer **regras de contexto** que levem a situações de melhor qualidade. Por conseqüência, **critérios de interação** sistematizados podem ser definidos a partir das regras de contexto, estabelecendo formas de incrementar a interação, em todos os seus níveis, sem prejuízos aos níveis de QoS classificados como aceitáveis.

Com esta proposta, pode-se verificar quando ocorre uma alteração no QoS de um ambiente e, a partir do cálculo do Δc , pode-se saber como executar as ações associadas a essa alteração da forma mais natural e gradativa para o usuário, conforme ilustra a **Figura 4.02**. Assim,

apesar das possibilidades de queda na Qualidade de Serviço, o usuário continuará realizando a(s) tarefa(s) em curso.

No âmbito tecnológico, os padrões MPEG-4 e MPEG-7 têm sido vistos como potenciais soluções para aplicações no campo da televisão, do rádio, da Internet, dentre outros. Em adição, os estes padrões MPEG oferecem recursos para suportar a arquitetura proposta neste trabalho. Conforme visto, o padrão MPEG-4 possui diversas formas de compactação (imagem, som, texto) e representação de mídias como objetos. O MPEG-4 permite, então, a apresentação de uma mesma cena multimídia em uma variedade de equipamentos e infraestrutura de comunicação. Desta forma, quando ocorrer uma mudança de contexto, uma apresentação pode continuar sendo apresentada sem dependências com o equipamento utilizado. Em adição, no instante em que o agente CA perceber uma variação de qualidade de Serviço no contexto, o ambiente pode alterar o grau de compactação de uma apresentação MPEG-4 de forma gradativa. Assim, o usuário pode assistir a uma apresentação multimídia independente de sua localização geográfica.

Neste cenário, poderá ocorrer uma mudança na qualidade da apresentação, mas será assegurada a continuidade da apresentação multimídia. De forma complementar, mecanismos de pesquisa e busca podem ser implementados e/ou aprimorados de acordo com os itens de consciência de contexto, e o padrão MPEG-7 oferece suporte semântica para a construção desses mecanismos.

Como análise da arquitetura proposta pela **Figura 4.02**, pode-se desenvolver uma reflexão sobre o seguinte caso: "O senhor Silva está assistindo o seu filme preferido em sua confortável sala de estar. O filme é codificado e apresentado no padrão MPEG-4 e com uma qualidade excepcional (64 Kbit/s de áudio, por exemplo), uma vez que o ambiente possui bons equipamentos e boa conexão de rede. Num determinado instante, a filha do Sr. Silva lhe telefona e lhe pede para que a busque na escola. Sr. Silva se levanta rapidamente e vai buscar a sua filha. Os agentes CA verificam uma mudança significativa no contexto e imediatamente param de transmitir o filme na televisão e armazenam num servidor o momento exato em que o Sr. Silva parou de assistir o filme. No seu percurso até a escola, Sr. Silva é surpreendido por um congestionamento. Neste instante, o sistema analisa o contexto e verifica que o Sr. Silva está apto para continuar vendo o filme em seu PDA. O sistema, então, envia uma mensagem para o Sr. Silva perguntando se ele deseja dar prosseguimento

ao filme. Sr. Silva responde que sim e volta a assistir o seu filme. Como o PDA não possui alto poder de processamento e a infra-estrutura de transmissão de dados não é baseada em banda larga, o sistema sabe que deve reduzir a qualidade do filme (por exemplo, reduzindo o áudio para 2 Kbits). Mesmo assim, o Sr. Silva pôde continuar assistindo o filme”.

4.5 Uma proposta de formalismo para consciência de contexto

No **Capítulo 5**, é apresentada e discutida a modelagem completa para a proposta aqui reportada. Nesta seção, conforme ilustra a **Figura 4.03**, apresenta-se o escopo completo da modelagem e discute-se o a formalização do modelo de consciência de contexto (**SPICE**), conforme ilustra o diagrama estrutural da modelagem (**Figura 4.03**, que é uma rerepresentação da **Figura 1.01**).

A modelagem completa é composta por três modelos distintos, porém integrados, tendo cada um deles as seguintes funções de cobertura em relação à modelagem como um todo:

- **SPICE** (*Schema for Interaction on Context-Aware Environments*): este modelo cobre a modelagem da interação do usuário para o caso específico dos ambientes conscientes de contexto, especificando os métodos para interação e considerando também o tipo de aplicação a ser desenvolvida; neste caso, a interação do usuário em uma aplicação de videoconferência pode ser especificada a partir dos requisitos identificados para uma aplicação de Televisão Interativa, por exemplo;
- **NECTAR** (*Network Environment with ConText-Awareness Rules*): este modelo cobre a infra-estrutura necessária para aplicações interativas conscientes de contexto, especialmente em termos dos requisitos de rede de comunicação;
- **PRACTIC** (*Profile for Context-Aware Interactive Applications*): este modelo cobre a geração de *profiles* (perfis) específicos para aplicações interativas conscientes de contexto, sendo composto por dois submodelos distintos: a) **PRACTIC-S** (*PRACTIC Schema*), que cobre a especificação de esquemas genéricos de descrição para programas interativos (Practic, 2001a); b) **PRACTIC-E** (*PRACTIC Program Examples*), que cobre a especificação de *profiles* para instâncias de programas de televisão interativa (Practic, 2001b).

Conforme apresentado, os agentes de software são vistos como componentes do ambiente e estão presentes nos três modelos propostos (**NECTAR**, **SPICE** e **PRACTIC**), servindo, portanto, como elemento integrador desses modelos. Neste sentido, a modelagem proposta

explora os recursos da plataforma JADE (*Jaca Agents Development Framework*), que está em desenvolvimento nos laboratórios TILAB (Bellefemine et al., 2001).

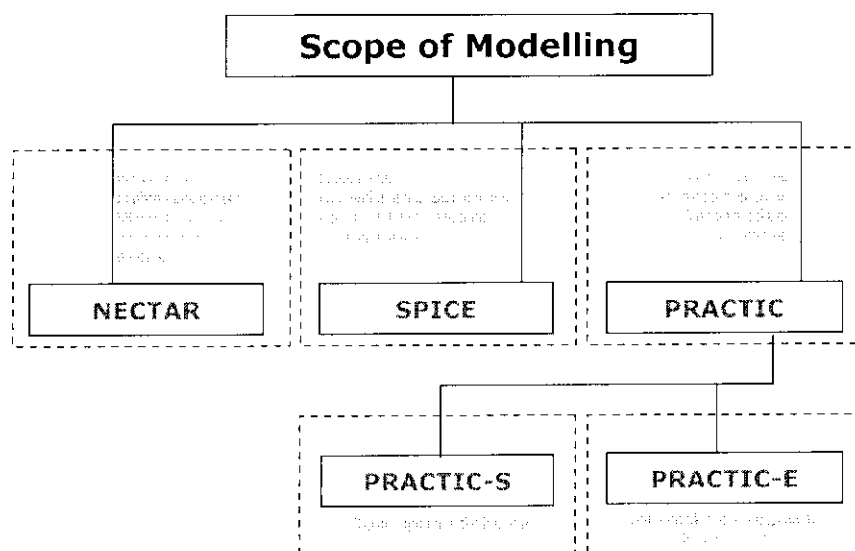


Figura 4.03: O escopo da modelagem - estrutura gráfica (Santos Jr. et al., 2001b)

Um dos focos deste trabalho está centrado em como modelar interações em ambientes interativos conscientes de contexto. Para tal, o uso de consciência de contexto, no modelo **SPICE**, é proposto como uma macro-definição que impacta todos os demais modelos da modelagem. Essa visão estende as propostas de Anind, Schmidt e Morse, que se limitam a oferecer os elementos fundamentais de consciência de contexto e a apresentar formas de uso desses elementos em aplicações (Dey & Abowd, 2000; Schmidt, 2001; Morse, 1998).

4.5.1 O modelo **SPICE**

O modelo **SPICE** cobre a modelagem da interação do usuário para o caso específico dos ambientes interativos conscientes de contexto, especificando os métodos para interação e habilitando novos critérios de interação. Tal abordagem influencia a produção, a distribuição e a apresentação de aplicações interativas, usufruindo do suporte tecnológico oferecido pelos padrões MPEG-4 e MPEG-7 (ISO, 1998b; ISO, 1999a; ISO, 2000a).

Do ponto de vista estrutural, a modelagem do modelo **SPICE** consiste dos elementos necessários para formalizar **consciência de contexto** e direcionar o seu uso em aplicações multimídia interativas. Esses elementos - Contexto, Usuário, Ambiente, Domínio, Recurso, Informação e Aplicação - foram identificados a partir da proposta de ambiente interativo defendida neste trabalho, que considera a aplicação como parte do ambiente. O conjunto

completo desses componentes é denominado CUADRIA, sendo referenciado nas seções desta tese, nas quais serão apresentadas as relações descobertas ao longo do desenvolvimento da modelagem. Nas Seções 4.5.1.1 a 4.5.1.7, são descritas as características de cada componente do modelo SPICE e sua formalização.

4.5.1.1 Domínio

Um determinado contexto existe e/ou pode existir a partir da existência de um determinado domínio de aplicação. A informação produzida em um contexto possui uma semântica associada ao domínio de aplicação ao qual pertencem as relações e entidades que definem o contexto, conforme ilustra a Figura 4.04.

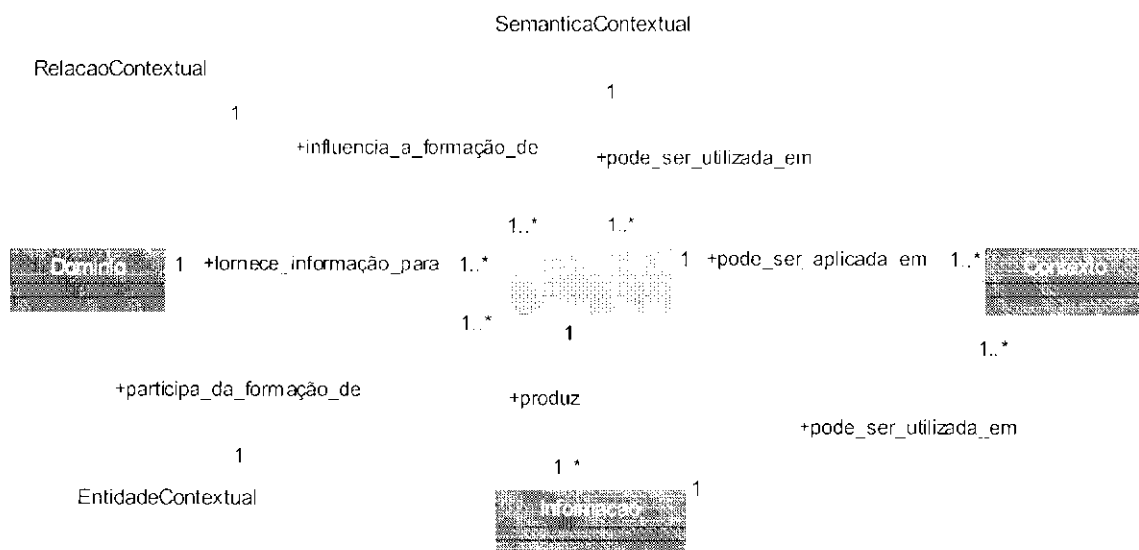


Figura 4.04 – Diagrama de classes UML para formalização do elemento DOMÍNIO

A comunicação entre domínios de aplicações se faz através de agentes que conheçam as regras de contexto de ambos os domínios e as informações que são produzidas em cada um deles. Adicionalmente, esses agentes são capazes de inserir novas semânticas aos domínios de aplicação e também modificar as semânticas existentes. Observa-se, portanto, que a qualidade da informação em um domínio de aplicação depende da qualidade das semânticas e da qualidade dos processos que executam as regras de contexto. Educação, comércio eletrônico e entretenimento são exemplos de domínios de aplicação.

4.5.1.2 Ambiente

O ambiente é formado por todas as entidades (elementos) que influenciam o usuário também durante o uso de uma aplicação em um determinado domínio.

Uma entidade, por sua vez, pode ser uma pessoa, um lugar ou objeto relevante para a interação entre um usuário e uma aplicação inserida nesse ambiente (Dey & Abowd, 2000).

Uma entidade pode ser nativa (ENTIDADE INTERNA) a um domínio ou ser inserida (ENTIDADE EXTERNA) nesse domínio a partir de uma necessidade, requisito ou evento. Desta forma, as relações que produzem as regras de contexto devem considerar também a natureza da entidade em relação ao domínio, conforme ilustra a **Figura 4.05**.

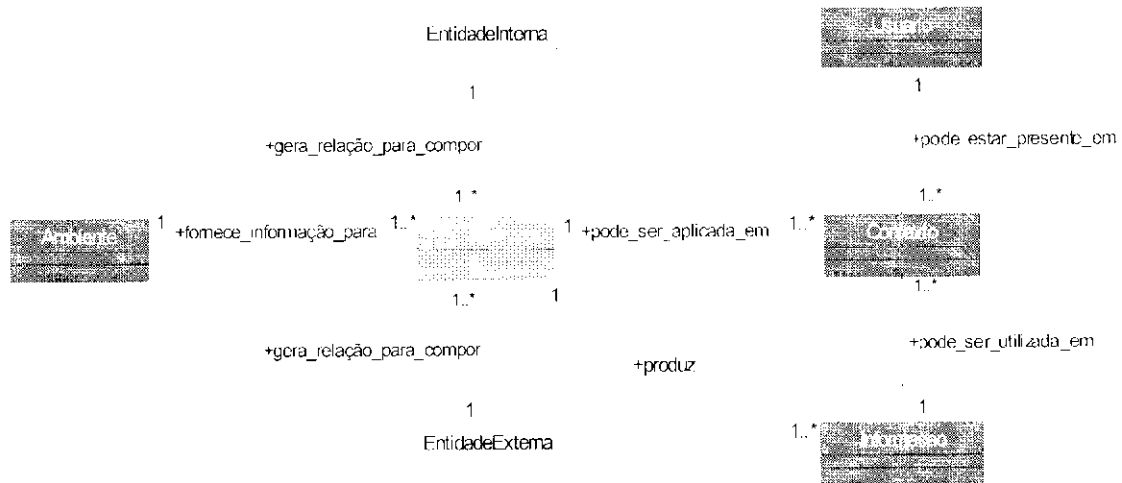


Figura 4.05 - Diagrama de classes UML para formalização do elemento AMBIENTE

Um ambiente é, portanto, uma instância de um domínio de aplicação, instância essa configurada de acordo com as entidades de ambiente disponíveis em um determinado momento. Quanto maior a flexibilidade de configuração do ambiente, maiores serão as possibilidades de uso das informações contextuais.

Assim, um ambiente é consciente de contexto se utiliza o contexto para oferecer informações relevantes e/ou serviços ao usuário, sendo que a relevância depende das tarefas que o usuário realiza (Abowd, 1999).

A criação de ambientes que utilizem as informações de contexto para fornecer informações relevantes e/ou ações específicas para cada usuário, conforme o grau de liberdade definido pelo projetista, é uma tarefa que precisa ser parametrizada para cada aplicação e considerando também o domínio de aplicação.

Esta parametrização é obtida com base em cinco questionamentos: a) onde o usuário está (WHERE); b) quem é o usuário (WHO); c) como o usuário está utilizando os recursos

(HOW); d) quando se pode realizar determinada operação (WHEN); e) o que o usuário está fazendo (WHAT).

No contexto da execução automática de tarefas e ações, pode-se considerar as quatro entidades de consciência definidas por Dey (Dey & Abowd, 2000), a partir dos questionamentos citados: a) localização (WHERE); b) identidade (WHO); c) atividade (WHAT + HOW); d) tempo (WHEN).

Uma sala de aula, uma sala de trabalho, um ambiente doméstico, um salão de convenções e um automóvel, são exemplos de ambiente.

4.5.1.3 Usuário

A presença do usuário é uma premissa básica na relação domínio-ambiente, justificando a necessidade das aplicações. Neste sentido, também a modelagem de um ambiente consciente de contexto deve ser centrada no usuário.

Em linhas gerais, existem cinco grupos de características que permitem identificar o usuário, quando este utiliza uma ou mais aplicações em um ambiente:

- características de identificação do próprio usuário: são as características do usuário enquanto indivíduo, tais como nome, idade, data de nascimento, dentre outras;
- características dos interesses do usuário: essas características formam a base para definir o porquê do interesse de uma pessoa por uma determinada informação ou parte dessa informação; por exemplo, a característica de interesse por vídeos sobre Geografia do Brasil pode fazer parte dos interesses de um usuário;
- características das preferências do usuário: formam o conjunto explícito das informações de maior relevância para o usuário; por exemplo, a preferência por uma mídia de áudio ou de vídeo, ou por uma mídia de texto, durante uma apresentação multimídia;
- características dos objetivos do usuário: as características dos objetivos do usuário permitem especificar melhor os elementos mais importantes de seus interesses e preferências; por exemplo, o usuário tem interesse em informações sobre Geografia do Brasil e preferências por receber informações em formato de áudio e vídeo, com o objetivo de estudar especificamente o nível de pluviosidade na Floresta Amazônica;

- características das tarefas do usuário: as características das tarefas do usuário permitem, juntamente com as características dos objetivos, especificar a granularidade dos interesses do usuário; neste sentido, uma lista das tarefas auxilia na definição da ordem dos processos necessários para se atingir o maior grau possível de satisfação em relação aos interesses do usuário; por exemplo: lista 1 de tarefas [a) enviar e-mail ao IBGE visando receber dados, em formato de áudio e vídeo, sobre a pluviosidade na Floresta Amazônica; b) esperar por resposta do IBGE; c) analisar os dados recebidos; d) selecionar os dados de interesse] ou lista 2 de tarefas [a) pesquisar sobre pluviosidade na Floresta Amazônica nos *sites* de busca da Internet; b) selecionar *links* de interesse; c) enviar aos destinatários selecionados a partir dos *links*, uma solicitação de dados em formato de áudio e vídeo; d) esperar por respostas dos destinatários; e) analisar os dados recebidos; f) selecionar os dados de interesse].

O usuário, ao utilizar uma aplicação inserida em um ambiente consciente de contexto, recebe diretamente os resultados produzidos pelas regras de contexto, que são geradas pelas relações domínio-ambiente-aplicação e pelas semânticas de ambiente. Neste sentido, melhorando a produção das regras de contexto pode-se ampliar os níveis de informações contextuais e, por conseqüência, fornecer mecanismos elaborados para elaboração da lista de tarefas do usuário. Assim, quanto maior o grau de otimização da lista de características das tarefas do usuário, maior será a possibilidade de melhoria nos processos de interação com o ambiente, contribuindo para o incremento do grau de satisfação do usuário.

4.5.1.4 Aplicação

Uma aplicação é toda e qualquer interface entre o usuário e o ambiente, em um determinado domínio, constituindo-se, essencialmente, dos elementos de software necessários para produção e consumo de informações nesse ambiente.

As aplicações também são responsáveis pelo estabelecimento de comunicação entre domínios diferentes, provendo mecanismos de suporte à troca de contexto, conforme apresentado pela **Figura 4.06**. Neste sentido, as regras de contexto podem ser incrementadas por semânticas mais complexas derivadas, por exemplo, do uso da tecnologia de agentes (Jennings & Wooldrige, 1995), com base em descrições de ontologias (Blásquez, 1998; Sowa, 2000).

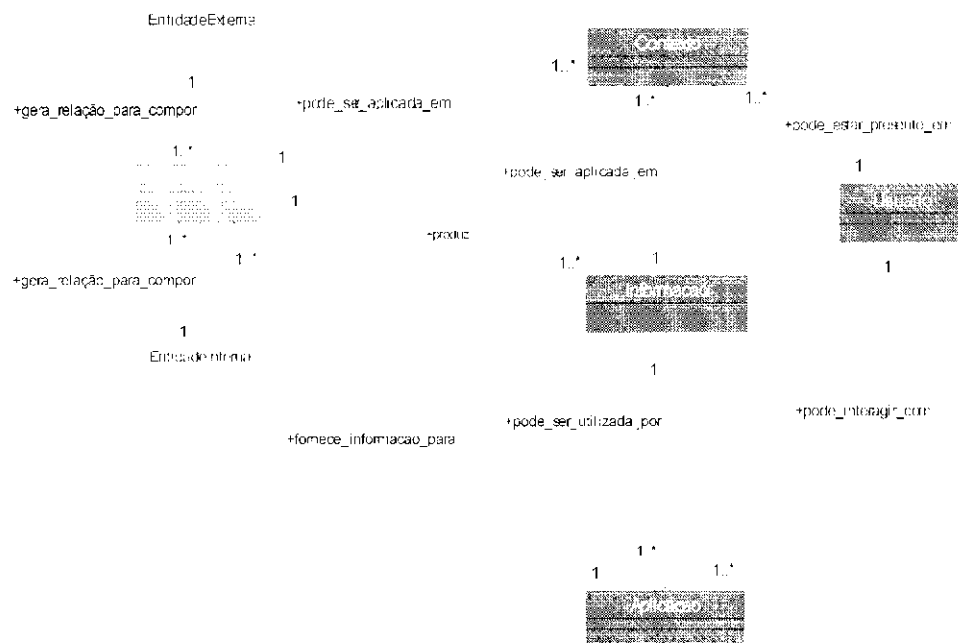


Figura 4.06 - Diagrama de classes UML para formalização do elemento APLICAÇÃO

Uma interface gráfica para interação do usuário com um computador é um exemplo de aplicação, assim como um software que recebe dados de um dispositivo de rede e configura automaticamente os requisitos de comunicação a partir dos dados recebidos.

4.5.1.5 Contexto

O contexto de uma aplicação consiste de todos os elementos associados direta ou indiretamente ao ambiente da aplicação e às informações que essa aplicação usa e/ou produz.

O contexto está diretamente associado ao domínio de aplicação no qual uma aplicação está inserida. De maneira geral, pode-se reconhecer o contexto através das relações entre o domínio de aplicação e os recursos disponíveis no ambiente, conforme ilustra a **Figura 4.07**.

Dentro de um determinado contexto, uma regra de contexto é responsável por associar um recurso a um processo baseado nos operadores INPUT (I), RUNNING (R) e OUTPUT (O), produzindo uma informação. Um operador IRO define a natureza de um recurso dentro de um ambiente. Assim, quando um operador INPUT é associado a um recurso, define-se que aquele recurso tem a natureza de gerar informações para INPUT no ambiente; se o operador associado for RUNNING, então o recurso tem a função de processamento dentro do

ambiente; por fim, se o operador associado for OUTPUT, o recurso tem natureza apropriada ao fornecimento de informações. Com esta formalização, estruturas *pipe* podem ser construídas a partir dos operadores IRO.

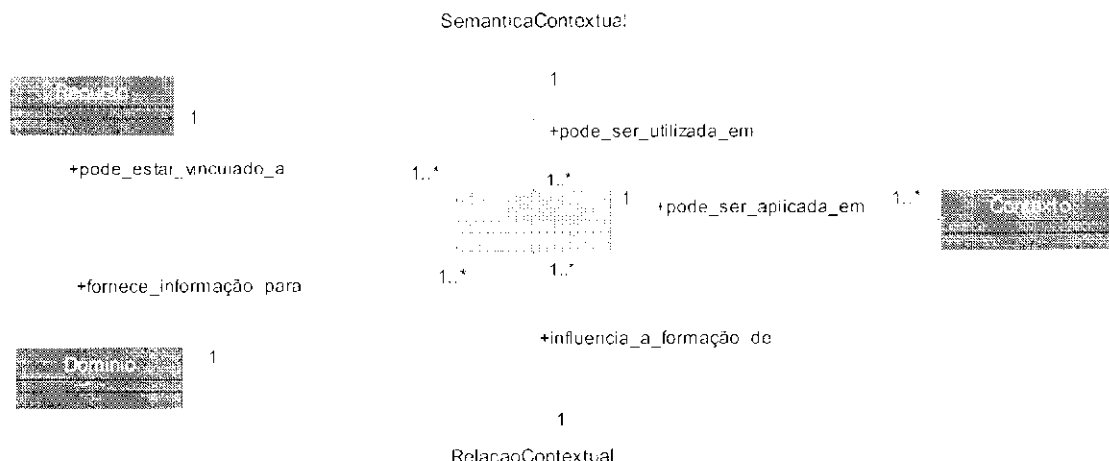


Figura 4.07 - Diagrama de classes UML para formalização do elemento CONTEXTO

Os operadores IRO são independentes do domínio de aplicação, mas herdam as relações geradas a partir da associação DOMINIO-RECURSO. Desta forma, um recurso pode estar associado a um mesmo operador em diversos domínios diferentes, realizando a mesma operação. No entanto, a semântica associada à regra de contexto pode influenciar na informação produzida pelo recurso/operador em um determinado domínio.

A troca de contexto pode ocorrer quando uma relação DOMINIO-RECURSO faz referência a outras relações que não conduzem ao conjunto de regras de contexto que forma o domínio de aplicação corrente.

Um grupo de vinte estudantes e um professor, em uma sala de aula na cidade de São Carlos, no estado de São Paulo, Brasil, usando projetor de recursos audiovisuais e um computador para acesso à Internet, em uma aula de Geografia, é um exemplo simplificado de descrição de um contexto. O dia da semana em que a aula está sendo realizada é um exemplo simples de semântica de ambiente capaz de mudar o contexto e/ou associá-lo a outros contextos.

4.5.1.6 Recurso

Um recurso é qualquer elemento de hardware ou software utilizado em um ambiente e, a partir do qual, se pode obter uma informação. Todo recurso precisa de um operador IRO para entrar em operação, como apresentado na **Figura 4.08**.

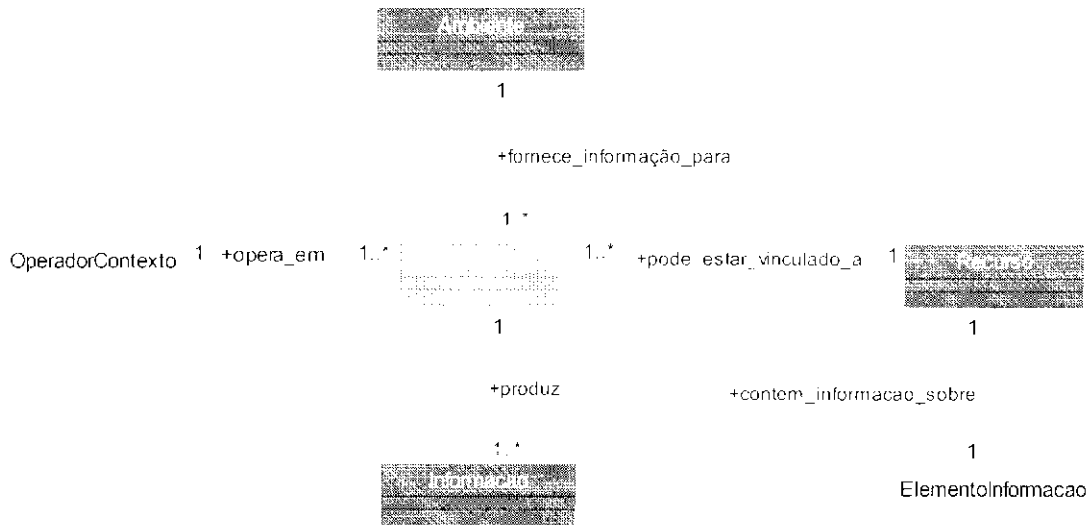


Figura 4.08 - Diagrama de classes UML para formalização do elemento RECURSO

Neste sentido, pode-se definir cinco estados básicos para um recurso: a) associação; b) ativação; c) operação; d) suspensão; e) reativação; e) desativação.

Genericamente, o recurso começa a operar quando se procede a sua associação a um operador IRO. O estado de ativação ocorre, por exemplo, a partir de um evento disparado no ambiente. O estado de operação é atingido respeitando-se a natureza do recurso e as regras de contexto que definem as relações no domínio de aplicação. O próprio ambiente e/ou outro recurso podem provocar um estado de suspensão; neste caso, o estado de reativação pode ser atingido disparando-se um novo evento. Por fim, o estado de desativação pode ser atingido a partir do estado de suspensão ou quando o recurso for considerado completo em termos das relações que definem as regras de contexto a ele associadas.

A desativação de um recurso pode ser definitiva ou temporária. No primeiro caso, todas as relações que definem as regras de contexto associadas ao recurso deixam de alimentar esse recurso. No segundo caso, uma semântica específica pode reativar o recurso quando necessário, promovendo a realimentação das relações que definem as regras de contexto.

4.5.1.7 Informação

A informação é o elemento produzido por um determinado recurso em um determinado contexto de um determinado ambiente.

O motor de execução de cada recurso é abastecido com dados de entrada (INPUT) e é responsável por algum tipo de processamento. Neste ponto, as semânticas de contexto e as relações contextuais são responsáveis por instruir ao recurso sobre as formas mais adequadas para processamento dos dados contextuais e geração das informações contextuais. Observa-se que o grau de consciência de contexto de um ambiente depende, diretamente, da granularidade dos dados de INPUT e da qualidade das regras contextuais aplicadas para produzir a informação, conforme ilustrado na **Figura 4.09**.

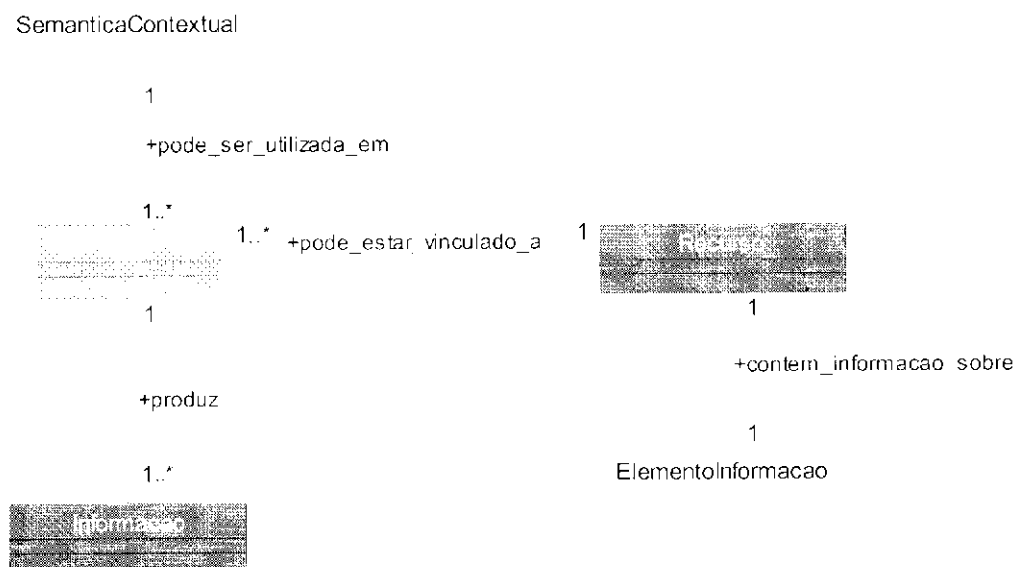


Figura 4.09 - Diagrama de classes UML para formalização do elemento INFORMAÇÃO

Considerando que um processo está associado a um recurso, pode-se dizer que uma informação é o resultado do trabalho realizado por esse recurso.

4.5.2 A proposta das regras contextuais como extensão do modelo de Anind

A formalização das relações contextuais depende diretamente da definição de três elementos: a) entidade contextual (EC), semântica contextual (SC) e regra contextual (RC). Esses elementos devem ser associados a um Motor de Contexto (MC), que é o agente responsável pela geração das relações contextuais, conforme ilustra a **Figura 4.10**.

A formalização das relações contextuais depende diretamente da definição de três elementos: a) entidade contextual (EC), semântica contextual (SC) e regra contextual (RC). Esses elementos devem ser associados a um Motor de Contexto (MC), que é o agente responsável pela geração das relações contextuais.

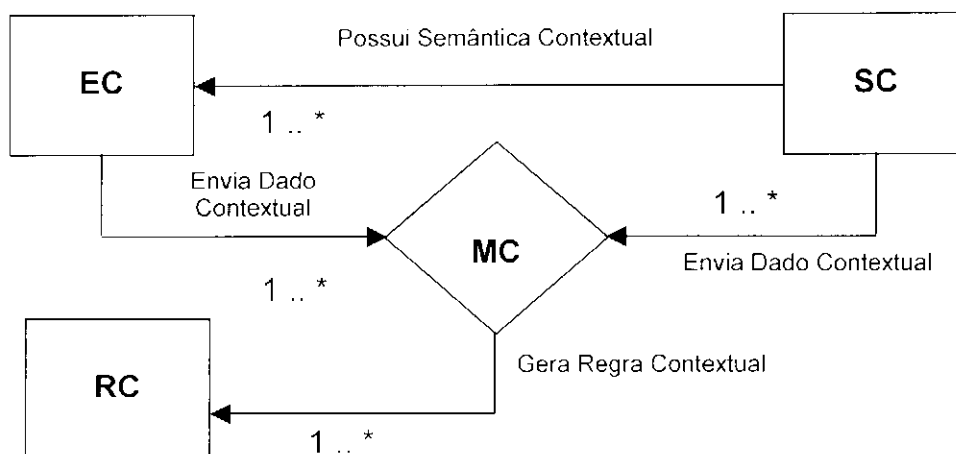


Figura 4.10 - Modelo para formalização das regras contextuais

Em termos de consciência de contexto, a modelagem proposta estende o modelo proposto por Anind (Dey & Abowd, 2000), acrescentando novos elementos formadores. Esses novos elementos são genéricos e agregam funcionalidades conceituais, independentes do domínio de aplicação e do ambiente.

Genericamente, três questionamentos conduzem a uma reflexão sobre os requisitos essenciais para o uso de consciência de contexto, por exemplo, em ambientes interativos:

- a) Quando (WHEN) se deseja produzir uma mídia (WHAT), em que (WHO + WHAT + HOW) se deve pensar?
- b) quando (WHEN) se deseja distribuir essa mídia produzida, quais (WHAT + HOW + WHERE) são os requisitos de distribuição ?
- c) quando (WHEN) se deseja apresentar essa mídia, quais (WHAT) são as opções de visualização disponíveis em um determinado momento (WHEN + WHERE) ?

No contexto deste trabalho, uma proposta para responder os questionamentos é elaborada com base no escopo CUADRIA. A **Figura 4.11** apresenta o diagrama de classes UML que relaciona todas as classes do modelo SPICE.

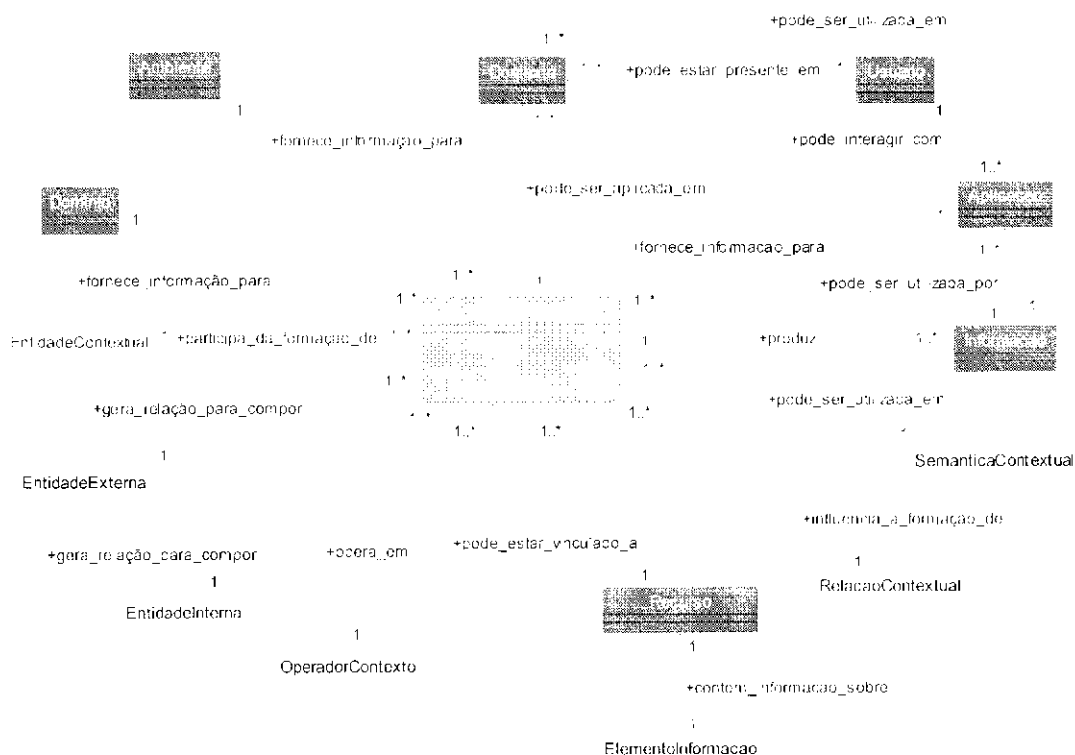


Figura 4.11 – Diagrama completo de classes UML do modelo SPICE

Para o primeiro questionamento, o elemento <WHAT> conduz a uma reflexão sobre o Conteúdo e Usuário. Para o segundo questionamento, o foco do elemento <WHAT> concentra-se sobre Recurso, Ambiente e Aplicação. No terceiro questionamento, o elemento <WHAT> refere-se ao Ambiente e Usuário.

Percebe-se, portanto, que o Contexto é um componente de ligação entre os demais componentes do escopo CUADRIA. Adicionalmente, o grau de consciência é responsável pelo nível de resposta para cada um dos questionamentos. Quanto maior o grau de consciência, mais completa será a lista de elementos que formarão cada uma das respostas. Definir mecanismos para aumentar o grau de consciência de contexto em um ambiente é a principal motivação deste trabalho. Neste sentido, estender o modelo de Anind (Dey & Abowd, 2000), muito mais do que definir novos elementos formadores de consciência de contexto implica em apresentar alternativas para construção de relações que gerem novas regras de contexto.

O modelo de Anind

O modelo de consciência de contexto proposto por Anind concentra-se no nível da interação usuário-aplicação, enfocando, principalmente, a execução automática de tarefas e ações.

O modelo é baseado em quatro entidades de consciência, geradas a partir dos cinco questionamentos fundamentais: a) localização (WHERE); b) identidade (WHO); c) atividade (WHAT + HOW); d) tempo (WHEN).

Na relação usuário-aplicação, os níveis de consciência de contexto são simplificados, uma vez que não são consideradas as relações usuário-ambiente e ambiente-aplicação, que produzem regras de contexto mais complexas.

Neste sentido, no modelo proposto por Anind não é necessário definir operadores que definam as relações para produção de regras de contexto. Usando o modelo, o projetista pode partir diretamente dos dados que são gerados pela interação usuário-aplicação e extrair o grau de consciência de contexto a partir dos cinco questionamentos básicos. A existência de relações ao nível do ambiente, da aplicação e do usuário, conduz à extensão das relações que produzem as regras de contexto. Nesta proposta, essa extensão inicia-se com a definição de operadores de consciência, que precedem o uso dos elementos <WHAT>, <WHO>, <WHEN>, <WHERE> e <HOW>.

O primeiro grupo de operadores (*operadores de pertinência ao contexto – OPC*) são dedicados às operações que verificam se uma determinada entidade ou informação é pertinente a um determinado domínio. Adicionalmente, os operadores de pertinência podem se comunicar, através de agentes, com operadores de pertinência de outros domínios, gerando mecanismos de busca por pertinência a um contexto.

Em termos das operações, os OPC podem verificar se um elemento pertence a um determinado contexto (<INCONTEXT>) ou, então, definir/configurar que um determinado elemento deve deixar de pertencer a um determinado contexto (<OUTCONTEXT>). Genericamente definidos, os OPC são aplicados às relações usuário-ambiente, aplicação-ambiente e recurso-aplicação, conforme ilustra a **Figura 4.12**.

```

<INCONTEXT>
  <ENTITY> </ENTITY>
  <INFORMATION> </INFORMATION>
  <CONTEXT ID="" />
</INCONTEXT>
<OUTCONTEXT>
  <ENTITY> </ENTITY>
  <INFORMATION> </INFORMATION>
  <CONTEXT ID="" />
</OUTCONTEXT>

```

Figura 4.12 - Esquema de relações de pertinência a um contexto

O segundo grupo de operadores (*operadores para troca de contexto – OTC*) são dedicados às operações que verificam e executam a troca entre contextos diferentes em um ambiente. Operadores OTC consideram as regras de contexto para auxiliar nas decisões que envolvam troca de contexto, identificando a(s) entidade(s) que solicita(m) uma operação de troca (*context change*) e os dados que serão necessários para essa troca, bem como a(s) entidade(s) que será(ão) responsável(is) por receber os dados no contexto destinatário.

Em termos das operações, os OTC podem verificar se um elemento pertence a um determinado contexto (<INCONTEXT>) ou, então, definir/configurar que um determinado elemento deve deixar de pertencer a um determinado contexto (<OUTCONTEXT>). Genericamente definidos, os OTC são aplicados às relações usuário-contexto, contexto-ambiente e contexto-contexto, conforme mostra a **Figura 4.13**.

```

<SAVECONTEXT>
  <ENTITY> </ENTITY>
  <DATACONTEXT> </DATACONTEXT>
  <CONTEXT ID="" />
</SAVECONTEXT>
<MOVE>
  <SOURCECONTEXT ID="" > </SOURCECONTEXT ID="" >
  <DATACONTEXT> </DATACONTEXT>
  <TARGETCONTEXT ID="" ID="" />
</MOVE>

```

Figura 4.13 - Esquema relações e transições em um contexto

O terceiro grupo de operadores (*operadores de recursos – OR*) são dedicados às operações de associação de recursos a entidades do ambiente, a partir das regras de contexto válidas para as entidades e recursos envolvidos na operação. De forma complementar, os operadores OR também são responsáveis pela liberação dos recursos associados às entidades.

Operadores OR consideram os seguintes estados de um recurso realizar as operações de associação e liberação: a) associação; b) ativação; c) operação; d) suspensão; e) reativação; e) desativação.

O estado de associação (operadores <GET> e <SET>) consiste em indicar um recurso mais apropriado para uma entidade em um determinado instante do contexto. O estado de ativação (<SEND> e <USE>) consiste em enviar os dados contextuais que ativam o motor de contexto (MC) para um determinado recurso. O estado de operação (operador <RUN>) consiste em manter o motor de contexto (MC) em estado de execução. O estado de suspensão (operador <STOP>) é atingido quando ocorre um evento no contexto ao qual pertence o recurso ou, ainda, quando o ambiente, usando as regras de contexto apropriadas, decide intervir no estado de operação de um recurso. Uma vez suspenso, um recurso pode voltar ao estado de operação através de uma nova ativação, que pode ser automática, através das regras de contexto, ou gerada por um evento. Por fim, o estado de desativação (operadores <LEAVE> e <FREE>) é atingido quando o recurso é liberado pela entidade a que estava associado. Essa liberação pode ocorrer quando se atinge o final do estado de operação ou, ainda, quando ocorre algum evento que gere estados não-recuperáveis no ambiente.

A proposta de extensão apresentada neste trabalho encapsula os elementos <WHAT>, <WHIO>, <WHEN>, <WHERE> e <HOW>, à luz dos operadores especificados, gerando classes que contém as regras de contexto.

Por exemplo, considerando o modelo proposto por Anind, a relação “Usuário_Pode Usar_Recurso” pode ser formalizada de acordo com a **Figura 4.14**. A figura mostra também exemplo de uso da relação contextual.

<ENTITY> : <OAct> : <ACTION ID=""> : <ENTITY>

<USUARIO> : <CAN> : <ACTION ID="use"> : <RESOURCE>

Figura 4.14 - Exemplo de formalismo e uso de uma regra contextual

A formalização de uma relação contextual recebe como parâmetros as entidades válidas para definição dessas relações em ambientes conscientes de contexto. Um operador de ação (OAct), representado por verbos, é usado para conectar as entidades que se deseja relacionar.

A **Figura 4.15** mostra como os operadores Oact podem ser definidos e utilizados em regras contextuais.

```

<ENTITY> : <CAN> : <ACTION ID=""> : <ENTITY>

<ENTITY> : <KNOW> : <ACTION ID=""> : <ENTITY>

<ENTITY> : <DEPEND> : <ACTION ID=""> : <ENTITY>
    
```

Figura 4.15 - Operadores de regras contextuais

Adicionalmente, as ações permitidas em <ACTION> podem ser configuradas de acordo com as semânticas habilitadas para o ambiente e que produzem as regras de contexto. A **Figura 4.16** ilustra o diagrama UML de classes que integra o uso de operadores de regras contextuais ao modelo SPICE.

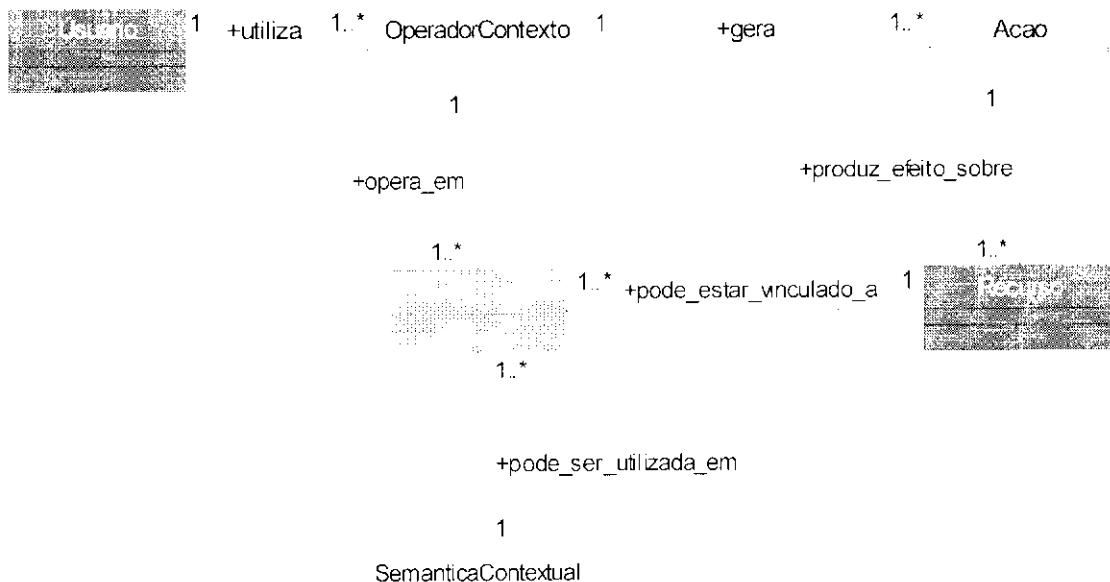


Figura 4.16 - Diagrama classes UML para os operadores de regras contextuais

A partir da formalização da relação de contexto, as regras contextuais podem ser geradas como extensões e combinações das entidades propostas no modelo de **Anind**. A **Figura 4.17** apresenta possibilidades de regra de contexto para a relação “Usuário_Pode Usar_Recurso”.

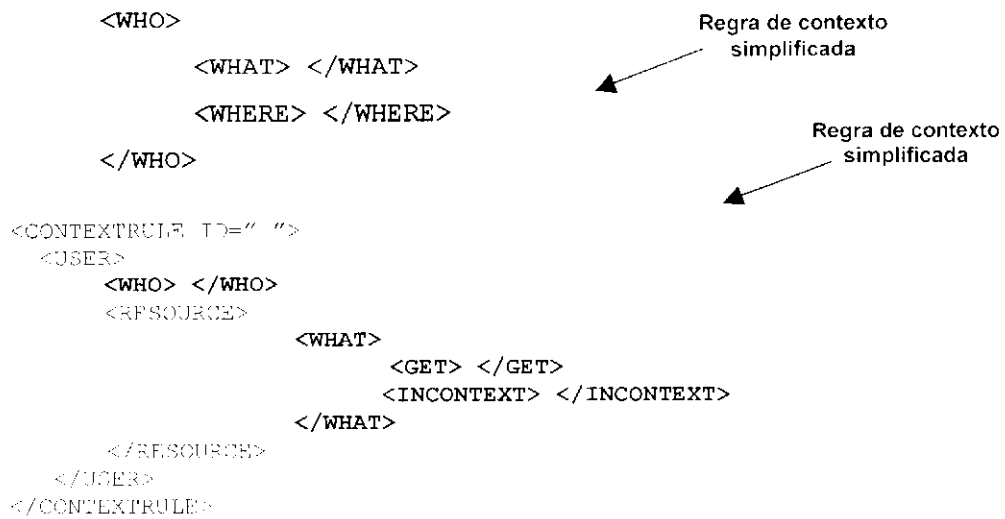


Figura 4.17 - Operadores de regras contextuais

Observa-se que a representação acima gera uma hierarquia entre as classes RESOURCE, USER e CONTEXTRULE, e sugere uma forma de encapsulamento, conforme ilustrado na Figura 4.18. Esta abordagem é coerente com a proposta teórica apresentada no escopo CUADRIA, da qual se extrai as relações entre as entidades contextuais.

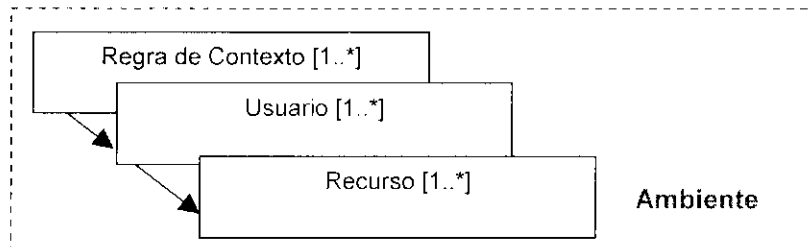


Figura 4.18 - Modelo para relações entre contexto e elementos contextuais

4.6 Usando agentes para prover consciência de contexto

Tradicionalmente, eventos e relacionamentos são analisados e modelados no desenvolvimento de interfaces para aplicações em ambientes computacionais. No entanto, ao serem considerados aspectos como a ubiqüidade, o projeto de interface exige requisitos que vão além dos eventos e relacionamentos, uma vez que a interface deve ser dotada de capacidades reativas, tanto ao nível do usuário quanto em termos da infra-estrutura formada pela própria aplicação e pela rede de comunicação, por exemplo. Adicionalmente, aspectos de consciência de contexto acrescentam novas possibilidades de interação aos usuários, aplicações e infra-estrutura.

Em linhas gerais, entende-se por contexto como sendo qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Uma entidade, por sua vez, pode ser uma pessoa, um lugar ou objeto relevante para a interação entre um usuário e uma aplicação, incluindo-se também o usuário e a aplicação em si. Assim, um sistema é consciente de contexto se utiliza o contexto para oferecer informações relevantes e/ou serviços ao usuário, sendo que a relevância depende das tarefas que o usuário realiza (Abowd, 1999).

A criação de ambientes que utilizem as informações de contexto para fornecer informações relevantes e/ou ações específicas para cada usuário, conforme o grau de liberdade definido pelo projetista, é uma tarefa que precisa ser parametrizada pela aplicação. Esta parametrização é obtida com base em cinco questionamentos: a) onde o usuário está (WHERE); b) quem é o usuário (WHO); c) como o usuário está utilizando os recursos (HOW); d) quando se pode realizar determinada operação (WHEN); e) o que o usuário está fazendo (WHAT).

No contexto da execução automática de tarefas e ações, pode-se considerar as quatro entidades de consciência definidas por Dey (Dey & Abowd, 2000), a partir dos questionamentos citados: a) localização (WHERE); b) identidade (WHO); c) atividade (WHAT + HOW); d) tempo (WHEN).

Aspectos de consciência de contexto podem ser importantes quando associados ao ambiente no qual uma aplicação está inserida, oferecendo a informações do contexto em que o usuário se encontra, as possibilidades de adaptação e informações sobre recursos disponíveis em um determinado momento. Este é o ponto-chave para a definição do conceito de *perfil de aplicação interativa* apresentado e discutido neste trabalho, e que se encontra detalhado no **Capítulo 5**.

Um ponto importante é que palavras como HOW, WHERE, WHAT, WHO e WHEN expressam semânticas que podem ser tão complexas quanto seja necessário a uma aplicação. Assim, a semântica HOW, por exemplo, ao se referir a "quais recursos estão em uso", expressa implicitamente "como o usuário está utilizando esses recursos". Neste ponto, implementando um mecanismo avaliador (agente + filtros, por exemplo) que apresente e/ou monitore uma lista dos recursos em uso, pode-se ter uma indicação de como o usuário está se sentindo no ambiente (contribuição para *emotional interfaces*). De forma complementar, a proposta de junção das semânticas HOW e WHAT oferece suporte à formação da entidade

"atividade" que, em suma, é um elemento formador do conceito de *perfil*, visto que para se ter atividade é necessário ter recursos em uso, quer seja ao nível do usuário quer seja ao nível da infra-estrutura.

Na proposta apresentada por este trabalho, a execução automática de tarefas e ações é de extrema importância. Neste sentido, agentes podem ser usados para permitir a implementação dos aspectos de consciência de contexto, considerando-se, por exemplo, as quatro entidades de consciência definidas por Dey (Dey & Abowd, 2000).

A definição fundamental de agente, obtida da literatura, diz que *um agente é capaz de colaborar, através da troca de dados com outros agentes e programas, para a resolução de problemas complexos* (Moreira & Walczowski, 1997). Complementando, pode-se dizer que *um agente computacional é um componente de software capaz de se comunicar e cooperar com outros agentes usando uma linguagem de comunicação* (Maes, 1994).

Observa-se que dispositivos como STB (*Set Top Boxes*) acrescentam novas possibilidades de interação que vão além dos dispositivos tradicionais de interação como mouse e o teclado, e a interação com o ambiente (consciência de contexto) no qual a aplicação está inserida passa a ser tão importante quanto a interação direta do usuário com uma aplicação.

O desenvolvimento de perfis para aplicações interativas é, então, uma forma de categorização que caracterize a situação de uma entidade em uma aplicação, bem como uma forma de se obter informações sobre o comportamento dessa entidade no tempo. Assim, com o uso de perfis em conjunto com informações de contexto, uma aplicação pode realizar ações e reações como as seguintes:

- a apresentação de informações e serviços para o usuário, como por exemplo a apresentação de mídias disponíveis e que estão relacionadas com o contexto do usuário;
- a execução automática de serviços, como por exemplo o disparo de eventos relacionados com a localização corrente de uma entidade;
- a associação do contexto com a informação para posterior recuperação, como por exemplo o armazenamento de informações contextuais durante uma sessão multimídia interativa, relacionando observações do(s) usuário(s) com o(s) assunto(s)-chave que faz(em) parte da descrição de uma ou mais cenas desse vídeo.

As definições de consciência de contexto oferecem condições para o desenvolvimento de aplicações altamente adaptativas, o que é relevante em ambientes interativos, como é o caso da TVI, por exemplo.

O grau de consciência (*awareness*) de um objeto em relação a outro quantifica a importância subjetiva ou relevância do objeto para um determinado ambiente ou mesmo para uma determinada mídia utilizada em uma aplicação. Como exemplo, no contexto das aplicações interativas, o grau de consciência pode ser mapeado para o volume de um canal de áudio ou para o nível de detalhe de imagem de uma seqüência de objetos formadores de uma cena em um vídeo.

Outro ponto importante reside no fato de que um objeto, em uma interação, pode afetar os níveis de consciência de outro objeto, exigindo, portanto, procedimentos de negociação. O objeto que observa (ou receptor) pode focar sua atenção em áreas ou objetos específicos, enquanto o objeto observado (ou transmissor) pode controlar sua “visibilidade” de forma que objetos em algumas áreas sejam mais conscientes que em outras áreas. Este aspecto é relevante no acesso a conteúdo de mídias em uma determinada cena, representada com tecnologias como MPEG-4 e MPEG-7, por exemplo.

4.7 Considerações finais

Conforme discutido, entende-se por contexto como sendo qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Parâmetros do comportamento humano podem caracterizar a consciência de contexto (*context-awareness*), especialmente quanto à definição das visões de contexto (superficial ou abrangente) que se deseja extrair de um ambiente. Ao nível do usuário, muitas são as formas de interação com as cenas em uma apresentação multimídia, inclusive explorando recursos sensoriais e informações contextuais.

Neste capítulo, foram apresentados os principais argumentos para definição de consciência de contexto e discutidas algumas visões de contexto e consciência reportadas na literatura. Como contribuição, apresentou-se um modelo para formalização de contexto e uma proposta para construção e formalização de regras contextuais. Aspectos dos ambientes interativos, e de seus objetos de mídia, foram considerados para a definição dos novos esquemas e

critérios de interação do usuário, considerando que essa interação acontece entre o usuário e o ambiente, e não somente ao nível da interação usuário-computador.

No próximo capítulo, discute-se a modelagem completa proposta por este trabalho para o caso dos ambientes interativos conscientes de contexto.

5.1 Considerações iniciais

Conforme discutido, tecnologias, técnicas e padrões têm sido produzidos para viabilizar o desenvolvimento de novas e complexas aplicações multimídia, especialmente aquelas inseridas no contexto da convergência de mídias e sistemas.

Nos capítulos anteriores, procurou-se justificar o uso de informações contextuais como uma importante fonte para oferecer suporte ao desenvolvimento de aplicações personalizadas às características e preferências do usuário, além de permitir diversos níveis de adaptabilidade para a infra-estrutura do ambiente no qual as aplicações estão inseridas.

Neste sentido, a modelagem de ambientes interativos conscientes de contexto apresenta-se como uma proposta capaz de servir como referência para o desenvolvimento de novas aplicações multimídia interativas e auxiliar a toda a cadeia produtiva – emissoras de *broadcasting*, *Internet Service Providers*, produtores de conteúdo, designers e usuário finais – na compreensão desse novo universo de aplicações. Os padrões MPEG-4 e MPEG-7 foram investigados e escolhidos como referência tecnológica, apesar de que a modelagem aqui reportada possui características de *openness* e pode ser utilizada para a construção de ambientes com outras bases tecnológicas.

Neste capítulo, são apresentados detalhes da arquitetura MPEG-4, especialmente do MPEG-4 *Systems* e da API MPEG-J, e discutidos os modelos NECTAR e PRACTIC, que compõem, juntamente com o modelo SPICE, a modelagem proposta. Detalhes complementares do modelo SPICE são também discutidos, uma vez que a formalização desse modelo foi apresentada no Capítulo 4. Aspectos integradores da arquitetura, representados pelos agentes de software, são também discutidos, especialmente em termos da implementação desses agentes com a plataforma JADE e com a API MPEG-J.

5.2 Princípios da modelagem

Para o desenvolvimento da modelagem de ambientes interativos conscientes de contexto, esta proposta considera o contexto dividido em duas categorias: a) o contexto do ambiente; b) o contexto do usuário. A linguagem UML fornece recursos adequados para modelagem de classes dos objetos, permitindo a especificação formal dessas classes e de seus relacionamentos em termos das ações que ocorrem em um ambiente (Booch et al., 1996). Em linhas gerais, aspectos tais como atributos, comportamento e relacionamentos (agregação, especialização e herança) podem ser representados como objetos, evidenciando as virtudes da linguagem UML. As classes de objetos podem ser mapeadas em tecnologias como XML, MPEG-4, MPEG-7, JAVA e MPEG-J, dentre outras.

O ambiente, e suas informações contextuais, pode ser modelado em diagramas UML, tais como o diagrama da classe, o diagrama da colaboração e o diagrama de seqüência. O estabelecimento de relações entre as classes de objetos do ambiente e as classes de objetos das aplicações inseridas nesse ambiente, é especificado nos diagramas UML, e essa modelagem pode ser implementada em uma linguagem de programação.

5.3 Representação de mídias

A modelagem UML permite descrever os eventos e os relacionamentos no ambiente em que uma aplicação está inserida. Por outro lado, a tecnologia MPEG-7 é usada para descrever objetos de mídia do ambiente.

O padrão MPEG-7, em desenvolvimento no grupo MPEG desde outubro de 1996, oferece recursos para a descrição dos dados associados a objetos de mídia em uma cena multimídia (ISO, 1998b; Nack & Lindsay, 1999). Conforme apresentado, o padrão MPEG-7 utiliza descritores, esquemas de descrição e uma linguagem de definição de descrição (DDL), permitindo a descrição de todos os tipos de objetos multimídia (ISO, 2001c). Neste sentido, o MPEG-7 tem características importantes para permitir a associação de informações contextuais aos objetos de mídia que compõem uma aplicação em um determinado ambiente.

As categorias de informação contextual identificadas por Dey (Dey & Abowd, 1999) podem ser mapeadas para os recursos do padrão MPEG-7 da seguinte forma:

- **Localização:** pode ser mapeada através de descritores e tipos primitivos de dados MPEG-7, tais como *string*, *float*, *url*, dentre outros; adicionalmente, esta informação contextual pode ser associada diretamente ao ID (identificador) de um objeto de mídia, descrito em MPEG-4, por exemplo;
- **Identidade:** o padrão MPEG-7 oferece suporte à definição de tipos como ENTITY e NAME através da associação de identificadores formatados como números inteiros; além disso, as entidades podem ser utilizadas em diferentes esquemas de descrição;
- **Atividade:** os tipos de dados MPEG-7 podem ser utilizados para descrever informações sobre os estados de um objeto em uma cena multimídia, utilizando-se, por exemplo, *strings* de caracteres;
- **Tempo:** o padrão MPEG-7 oferece dois tipos primitivos de dados para manipulação de informações de tempo: **date** e **time**; estes tipos primitivos são úteis na associação de dados cronológicos às informações contextuais do ambiente; esquemas de descrição estendidos podem ser definidos pela combinação dos tipos primitivos de dados, o que acrescenta possibilidades semânticas em uma cena multimídia.

Vale a pena observar que os esquemas de descrição MPEG-7 podem ser utilizados para definir hierarquia entre informações contextuais, tanto em um mesmo ambiente quanto entre ambientes distintos.

5.4 MPEG-4: uma tecnologia para vídeo interativo

Conforme apresentado, o padrão MPEG-4 (ISO, 1999a; Battista et al., 1999) foi desenvolvido pelo grupo MPEG com o objetivo principal de fornecer suporte à codificação, transmissão, apresentação e gerenciamento de objetos multimídia (áudio, vídeo, imagens, animações 2D e 3D, textos), bem como o controle da sincronização e da recuperação desses objetos em um *stream* de mídia que estiver sendo transmitido.

Outras características como o gerenciamento da interatividade e a descrição das cenas, em termos do comportamento temporal e espacial, também são suportadas pelo padrão.

Em um *stream* de mídia representado em MPEG-4 é possível a busca aleatória por seqüências áudio ou vídeo, bem como localizar e interagir com quaisquer objetos da cena. A infraestrutura mínima para uso da tecnologia MPEG-4 em um ambiente interativo pode ser vista na **Figura 5.01**.

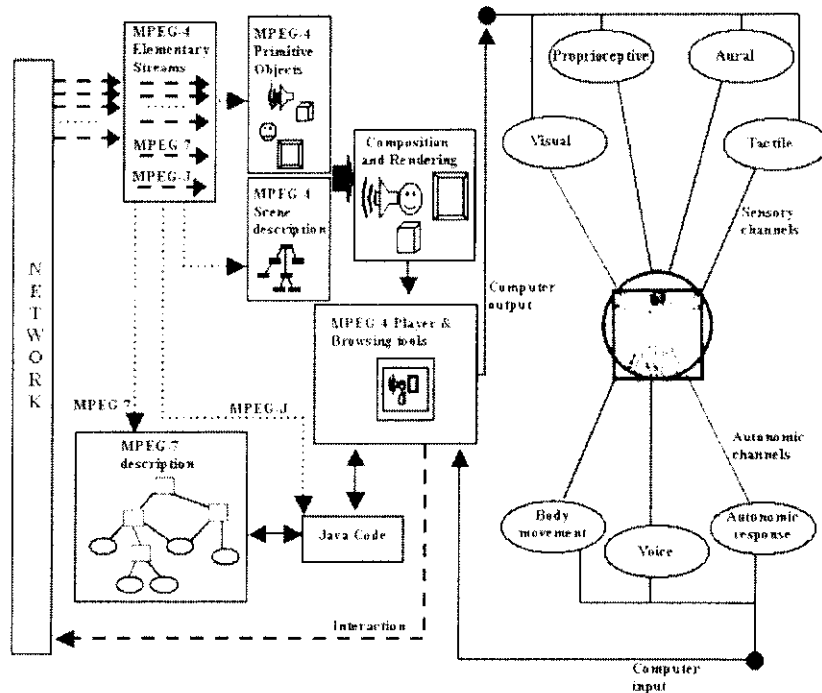


Figura 5.01 - Arquitetura mínima para uso do MPEG-4 em ambientes interativos

No cenário apresentado pela **Figura 5.01**, uma apresentação multimídia consiste de um conjunto de objetos de mídia e seus relacionamentos. A apresentação MPEG-4 é codificada e distribuída (cada objeto MPEG-4 é um *stream* de mídia elementar) através da rede (ou em um sistema local). Uma descrição MPEG-7 dos objetos de mídia e do contexto da aplicação/usuário/ambiente pode ser codificada também como um *stream* elementar em um fluxo MPEG-4.

Conforme citado, um recurso interessante do padrão MPEG-4 é a API MPEG-J (ISO, 1999d), que consiste de um conjunto de classes de objetos que permite a programação da interação entre os usuários e a aplicação multimídia. Como a API MPEG-J é inteiramente baseada em JAVA, é possível usufruir de todos os recursos da programação JAVA na construção de uma aplicação, que é denominada *MPEGlet*. A **Seção 5.6** apresenta os detalhes da API MPEG-J e que são importantes para esta proposta.

Observa-se que o desenvolvimento de interfaces MPEG-J acrescenta novas possibilidades de controle das informações contextuais e representa um caminho natural para a integração de agentes de software nesta proposta.

5.5 MPEG-4 Systems: relacionando classes de objetos

Como apresentado no **Capítulo 2**, o MPEG-4 *Systems* define um conjunto de algoritmos avançados para codificação/decodificação da informação audiovisual.

Os *streams de mídia* (*Elementary Stream* – ES) codificados podem ser transmitidos ou armazenados como objetos, e os sistemas MPEG-4 apresentam descrições e interfaces para estabelecimento de relacionamentos entre esses objetos de mídia na composição de cenas audiovisuais.

Um BIFS (*Binary Information For Scenes*) descreve os arranjos espaciais e temporais entre os objetos de mídia de uma cena. Os usuários, através de um *player* MPEG-4 podem interagir com cada um dos objetos de mídia de uma cena e gerar ações que alterem propriedades e o comportamento desses objetos. Em um nível mais baixo, os *descriptors do objeto* (ODs) definem o relacionamento entre os *streams* elementares de mídia associados a cada objeto (por exemplo, o *stream* de áudio e o *stream* de vídeo para um usuário em uma aplicação de videoconferência). Um OD fornece também informações adicionais para a decodificação de um *stream* de mídia, tais como a URL, dados de propriedade intelectual e esquemas de descrição da mídia, que podem ser especificados em MPEG-7.

Num contexto mais amplo, o MPEG-4 *Systems* contém as seguintes especificações:

- um formato padrão de arquivo para suportar o armazenamento e o intercâmbio de informação audiovisual;
- um modelo para programação de níveis de interatividade;
- um modelo geral de eventos para as cenas;
- uma API programável – MPEG-J – para desenvolvimento de controles do terminal, da aplicação e da infra-estrutura de comunicação;

Nas relações entre os terminais e a rede de comunicação, o MPEG-4 *Systems* define primitivas de mapeamento entre protocolos, tais como RTP, UDP e IP, além de esquemas para encapsulamento em nível de transporte, como é o caso de *streams* MPEG-4 em fluxos MPEG-2.

Em termos de implementação de aplicações, o MPEG-4 *Systems* oferece três APIs para o controle de todos os elementos envolvidos na apresentação de uma cena audiovisual: **Scene**

(controle de cenas), **Resource** (controle de recursos do ambiente) e **Net** (controle da rede de comunicação). Diversas interfaces e classes encontram-se especificadas nessas APIs e a programação pode ser desenvolvida inteiramente com MPEG-J, conforme apresentado na **Seção 5.6**.

5.5.1 API Scene

A API Scene fornece mecanismos para que aplicações MPEG-J possam acessar e manipular os elementos de uma cena audiovisual que estiver sendo *renderizada* em um *player*. Os BIFS fornecem os dados dos objetos de mídia e a interface BIFS-Scene é definida.

As classes e interfaces para controle de cenas encontram-se definidas na biblioteca `org.iso.mpeg.mpegj.scene` e pode-se selecionar as classes mais apropriadas para uma determinada arquitetura aplicação-rede-terminal.

O controle de eventos nos BIFS de um grafo de cenas pode ser feito através das interfaces de classes **EventIn** e **EventOut**. Adicionalmente, pode-se acrescentar monitores de eventos utilizando a classe **EventOutListener**.

A classe **EventIn** contém os métodos para associação de eventos a cada nó em um grafo de cenas MPEG-4. Essa associação é feita com base nas relações temporais e espaciais entre os objetos de mídia da cena. Do mesmo modo, a classe **EventOut** contém métodos para desabilitar eventos associados a um nó do grafo de cenas. Adicionalmente, a API oferece possibilidades de inserção de monitores de eventos através da classe **EventOutListener**, que verificam as transições entre os objetos da cena.

A interface **SceneManager** contém métodos para adicionar e remover um monitor para toda a cena (**SceneListener**). Este conceito é diferente do conceito de evento, uma vez que um **SceneListener** é capaz de gerar informações de controle de todo o comportamento da cena e não somente de um nó do grafo. A interface **SceneListener**, por sua vez, monitora diretamente o BIFS de uma cena e notifica (**notify()**) uma aplicação (por exemplo, MPEG-J) nos momentos em que houver qualquer tipo de alteração no BIFS.

A interface Scene funciona como um agente que se relaciona com o BIFS de uma cena. Um método **getNode()** retorna um descritor para um nó desejado em uma cena. Desta forma, uma segunda interface – **Node** – age para controlar diretamente as propriedades de um nó

no grafo de uma cena (`getEventOut()`, `EventOutListener`, `sendEventIn`, `getNodeType()`, `NodeType`).

5.5.2 API Resource

A API **Resource** fornece mecanismos para que aplicações MPEG-J possam acessar e manipular informações de todos os recursos associados a um terminal (**MpegjTerminal**) que interage com uma aplicação multimídia MPEG-4.

As classes e interfaces para controle de cenas encontram-se definidas na biblioteca `org.iso.mpeg.mpegj.resource` e pode-se selecionar as classes mais apropriadas para uma determinada arquitetura aplicação-rede-terminal.

O gerenciamento eficiente de recursos auxilia o controle de uma sessão MPEG-4 e permite a personalização de componentes do ambiente associados a esses recursos.

O principal componente da API **Resource** é o pacote **ResourceManager**, que é baseado no modelo de eventos MPEG-4 e é responsável pelo monitoramento dinâmico e estático de todos os *profiles* associados aos recursos de um terminal, inclusive aqueles que têm interações com a rede de comunicação (neste caso, existem interfaces entre a API Resource e a API Net). Uma coleção de classes pode ser utilizada no desenvolvimento de mecanismos de controle dos recursos. Dentre essas classes, pode-se citar a classe **Renderer**, a classe **MPDecoderMediaEvents** e a classe **MPRendererMediaEvents**.

Para cada decodificador, o **ResourceManager** gerencia os eventos associados e relaciona esses eventos com o restante dos componentes da cena. Desta forma, o gerenciamento do recurso considera o contexto da cena como um todo, e não somente as propriedades daquele recurso. Uma vez que é conhecido um nó em um grafo de nós de uma cena, as interfaces acessam os decodificadores associados a esse nó (usando o OD ou ESID), determinando, por exemplo, as prioridades entre os decodificadores.

Para manipular as potencialidades de cada recurso, a API **Resource** possui um outro conjunto de classes – **CapabilityManager** – que fornece métodos que trabalham com abstrações de recursos em relação à arquitetura de hardware e software em uso. Diferentes plataformas e cenários podem fazer parte de um ambiente interativo, e os recursos

associados a um terminal devem garantir parâmetros de QoS pré-estabelecidos. Como exemplo, pode-se citar algumas premissas:

- uma aplicação multimedia em uma infra-estrutura *wireless* pode usufruir de diferentes capacidades de comunicação e conexões;
- um computador pode estar envolvido no processamento de aplicações que não fazem parte do ambiente interativo;
- sessões audiovisuais utilizam múltiplas fontes de mídia.

Em termos do terminal em si, a API **Resource** possui um conjunto de classes – **TerminalProfileManager** – que permite a uma aplicação selecionar o melhor *profile*, com base nas especificações MPEG-4 *Systems* para uma determinada situação em que o terminal estiver envolvido. As classes oferecem avançados mecanismos de busca para seleção automática de um *profile*; neste ponto, aspectos de consciência de contexto, conforme discutido no **Capítulo 4**, podem ser associados aos recursos dos terminais.

5.5.3 API Net

A API **Net** oferece recursos para permitir o controle de todos os componentes da rede que se relacionam com um *player* MPEG-4. Isto é possível pelo fato de que as especificações DMIF contém abstrações de componentes de rede em todos os níveis de uma arquitetura. Desta forma, as aplicações podem ser **conscientes** dos detalhes de conexões de rede que estão sendo usadas (LAN, WAN, MAN) e ter acesso às propriedades dos serviços que estão sendo oferecidos. As funcionalidades da API **Net** podem ser divididas em dois grandes grupos:

- **Network query**: oferece recursos que habilitam a execução de pedidos de informações estatísticas sobre os componentes da rede; essas informações são negociadas com as interfaces DMIF e devem ser pedidos que estejam em conformidade com o MPEG-4 *Systems*;
- **Channels control**: oferece mecanismos para o controle de cada canal de transmissão de dados audiovisuais que estiver em uso pelo *player* MPEG-4; desta forma, uma aplicação MPEG-J pode habilitar/desabilitar a transmissão de um *stream* de mídia (*elementary stream*), adicionando requisitos para personalização de um ambiente.

Observa-se que as APIs **Scene**, **Resource** e **Net** constituem o núcleo principal das especificações MPEG-4 *Systems*, no que tange à parte programável das aplicações.

5.6 A arquitetura MPEG-J

MPEG-J é um subsistema MPEG-4 programável que especifica uma API (*Application Programming Interface*) que oferece classes Java para permitir a interoperabilidade entre um *player* MPEG-4 e um código Java. Combinando os objetos de mídia MPEG-4 e o código Java, é possível desenvolver mecanismos complexos de manipulação dos dados associados a esses objetos de mídia e controlar, de forma inteligente e/ou automática, a interação com esses objetos em uma sessão multimídia. A Figura 5.02 ilustra o funcionamento de um código MPEG-J em uma aplicação baseada em MPEG-4 *Systems*.

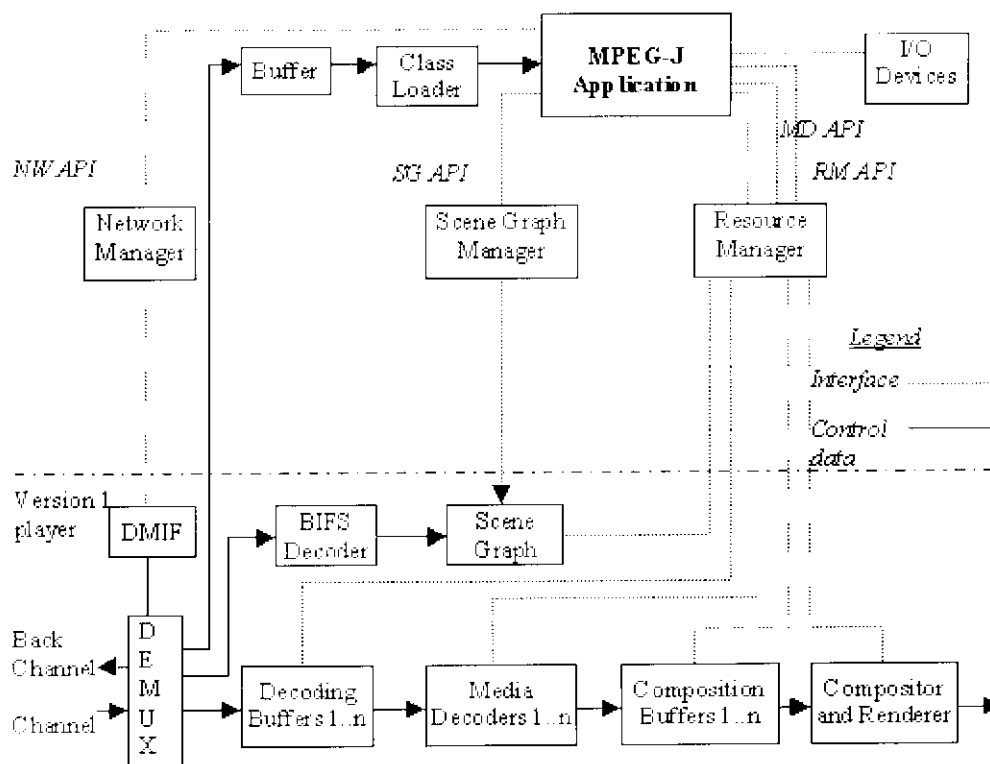


Figura 5.02 - Identificação das interfaces MPEG-J em uma arquitetura MPEG-4 System (ISO, 1999a)

Observa-se, na parte inferior da Figura 5.02, que existe um motor de apresentação de objetos audiovisuais, conforme prevê o MPEG-4 *Systems*. O subsistema MPEG-J pode controlar o motor da apresentação, influenciando diretamente a forma de apresentação e interação com os objetos de mídia.

A aplicação Java, desenvolvida com a API MPEG-J, é distribuído como um *elementar stream* (fluxo elementar), da mesma forma que ocorre com o áudio, com o vídeo e com todos os demais objetos audiovisuais.

Em um *player* MPEG-4, o código MPEG-J é decodificado como um objeto de mídia. Em termos funcionais, o código MPEG-J tem acesso ao grafo de objetos de mídia de uma cena MPEG-4, tendo permissão para consultar dados do grafo, alterar a hierarquia dos nós do grafo, remover nós do grafo e gerenciar recursos associados aos objetos de mídia representados no grafo.

O gerenciador de recursos da API MPEG-J (*ResourceManager*) oferece primitivas para acesso e controle dos terminais associados ao *player* e também dos recursos da infra-estrutura de comunicação (*NetworkManager*) associada aos terminais. A API MPEG-J permite, ainda, o controle sobre os decodificadores (*DecoderManager*) de objetos de mídia. Vale ressaltar que todas as formas de interação com a rede de comunicação são compatíveis com as especificações DMIF do MPEG-4.

Nas **Subseções 5.6.1** e **5.6.2**, são apresentadas as classes MPEG-J que estão diretamente relacionadas com as classes modeladas nesta proposta, especialmente em termos do gerenciamento das interações, com suporte dos agentes software, que ocorrem em um ambiente interativo consciente de contexto. Alguns exemplos de aplicação MPEG-J são também apresentados.

5.6.1 Uma visão geral dos recursos MPEG-J relevantes para esta proposta

Genericamente, a API MPEG-J é composta de cinco macro-pacotes especificados de acordo com a sintaxe padrão da linguagem Java, conforme pode ser visto na **Tabela 5.01**.

Tabela 5.01 - Pacotes da API MPEG-J

Pacotes	Descrição
<code>org.iso.mpeg.mpegj</code>	Oferece classes para acesso ao MpegjTerminal que fornece métodos para manipulação de diferentes classes de gerenciadores, tais como ResourceManager , SceneManager e NetworkManager
<code>org.iso.mpeg.mpegj.decoder</code>	Oferece classes para acesso e controle dos decodificadores de objetos audiovisuais
<code>org.iso.mpeg.mpegj.net</code>	Oferece classes para acesso e controle dos components de rede associados a um <i>player</i> MPEG-4
<code>org.iso.mpeg.mpegj.resource</code>	Oferece classes para permitir o gerenciamento das propriedades de um recurso presente em uma aplicação baseada no MPEG-4 <i>Systems</i>
<code>org.iso.mpeg.mpegj.scene</code>	Oferece classes para manipular o grafo de uma cena MPEG-4

A partir dos pacotes-padrão, a API MPEG-J também especifica interfaces e a classe pública para implementação de MPEGlets, conforme pode ser visto na **Tabela 5.02**.

Tabela 5.02 - Interfaces da API MPEG-J

Interfaces	Descrição
DecoderConfigDescriptor	Uma interface para implementação de classes que acessem DecoderConfigDescriptor
ObjectDescriptor	Uma interface para descritores de objetos de mídia ObjectDescriptor
MPEGlet	Interface padrão que deve ser implementada pela aplicação MPEG-J
ESDescriptor	Uma interface para descritores de <i>streams elementares de mídia</i> ESDescriptor
Classes	
MpegjTerminal	Uma classe padrão para interação com um terminal MPEG-J MpegjTerminal
Exceções	
MPEGJException	Uma classe para tratamento de exceções MPEG-J com base no modelo de exceções da language Java.
NetworkManagerNotFoundException	
SceneManagerNotFoundException	

A interface **DecoderConfigDescriptor** possui métodos que manipulam o tipo e as propriedades de um descritor de objeto de mídia, conforme pode ser visto na **Tabela 5.03**.

Tabela 5.03 - Métodos da interface DecoderConfigDescriptor

Métodos	Descrição
int getStreamType();	Retorna o tipo de um <i>elementary stream</i> de acordo com os tipos especificados no MPEG-4 <i>Systems</i>
int getBufferSizeDB();	Retorna o tamanho do buffer de decodificação para um <i>elementary stream</i>
int getMaxBitrate();	Retorna o máximo <i>bitrate</i> (bits por segundo) de um <i>elementary stream</i>
int getObjectTypeInfoIndication();	Retorna uma indicação de um objeto de mídia ou cena com vínculos com um decodificador

A interface **ESDescriptor** possui métodos que manipulam as propriedades de um descritor de *elementary stream*, conforme pode ser visto na **Tabela 5.04**.

Tabela 5.04 - Métodos da interface ESDescriptor

Métodos	Descrição
int getESID();	Retorna o ID associado a um ESDescriptor
Boolean isStreamDependent();	Retorna TRUE se há dependência entre dois ou mais <i>elementary streams</i> .

<code>int getdependsOn_ESJD();</code>	Retorna o ID (ESID) de um <i>elementary stream</i> que dependa de um outro <i>elementary stream</i>
<code>int getStreamPriority();</code>	Retorna a prioridade relativa de um <i>elementary stream</i> em uma cena
<code>DecoderConfigDescriptor getDecoderConfigDescriptor();</code>	Retorna um objeto de propriedades de um descritor de decodificador

O tratamento de exceções na API MPEG-J segue o modelo de exceções da linguagem Java, e existem formas de tratar exceções em relação à rede, aos decodificadores, aos recursos e às cenas, observando-se as seguintes bibliotecas de classes:

- `org.iso.mpeg.mpegj.net.AccessLayerException;`
- `org.iso.mpeg.mpegj.scene.BadNodeException;`
- `org.iso.mpeg.mpegj.scene.BadParameterException;`
- `org.iso.mpeg.mpegj.resource.CapabilityManagerNotFoundException;`
- `org.iso.mpeg.mpegj.resource.IllegalDecoderMediaEventsException;`
- `org.iso.mpeg.mpegj.resource.IllegalRendererMediaEventsException;`
- `org.iso.mpeg.mpegj.resource.IllegalStreamMediaEventsException;`
- `org.iso.mpeg.mpegj.scene.InvalidNodeException;`
- `org.iso.mpeg.mpegj.scene.InvalidSceneException;`
- `org.iso.mpeg.mpegj.decoder.MediaDecoderException;`
- `org.iso.mpeg.mpegj.resource.RendererNotFoundException;`
- `org.iso.mpeg.mpegj.resource.ResourceManagerNotFoundException.`

A classe **MpegjTerminal** possui construtores e métodos para o gerenciamento do terminal ao qual um *player* MPEG-4 está associado e também para gerenciamento dos recursos e da rede que suportam o terminal, conforme pode ser visto na **Tabela 5.05**.

Tabela 5.05 – Métodos da classe MpegjTerminal

Método	Descrição
<code>MpegjTerminal();</code>	Método que constrói um objeto MpegjTerminal
<code>MpegjTerminal(MPEGlet);</code>	Método que constrói um objeto MpegjTerminal
<code>ResourceManager getResourceManager();</code>	Método que obtém um gerenciador de recursos para o terminal selecionado
<code>SceneManager getSceneManager();</code>	Método que obtém um gerenciador de cenas para o terminal selecionado
<code>NetworkManager getNetworkManager();</code>	Método que obtém um gerenciador de rede para o terminal selecionado
<code>ObjectDescriptor getinitOD();</code>	Retorna o descritor do primeiro objeto de mídia (raíz) de uma cena
<code>Vector getODs(int);</code>	Retorna um vetor de objetos de descritores de objetos de mídia (OD) presentes em uma cena

A interface **ObjectDescriptor** possui métodos que manipulam as propriedades de um objeto de mídia que é descrito como *elementary stream*, conforme pode ser visto na **Tabela 5.06**.

Tabela 5.06 – Métodos da interface ObjectDescriptor

Método	Descrição
<code>int getObjectDescriptorID();</code>	Retorna o ID associado a um ObjectDescriptor
<code>Vector getESDescriptors();</code>	Retorna um vetor de <i>elementary streams</i> associados com um ObjectDescriptor .
<code>ESDescriptor getESDescriptor(int);</code>	Retorna um descritor de <i>elementary stream</i> associado a um ESID
<code>String getURL();</code>	Retorna uma URL associada a um ObjectDescriptor

O pacote **Resource** possui classes e interfaces para suportar a manipulação de todos os recursos presentes em terminal que esteja associado a um *player* MPEG-4. Dentro do pacote **Resource**, é importante apresentar os métodos da classe **Capability**, que descrevem as propriedades mais importantes de um terminal, permitindo a implementação de diversos níveis de personalização, conforme pode ser visto na **Tabela 5.07**.

Tabela 5.07 – Classes e interfaces do pacote Resource

Construtores	Descrição
<code>CapabilityManager();</code>	
Métodos	
<code>void notifyApplicationFreeMemory(long, Observer);</code>	
<code>void notifyTerminalFreeMemory(long, Observer);</code>	
<code>void notifyTerminalLoad(long, Observer);</code>	
<code>void notifyTerminalNetworkLoad(long, Observer);</code>	
<code>void deleteObserver(Observer);</code>	
<code>int getNumCPUs();</code>	
<code>int getCPUSpeed(int);</code>	
<code>String getCPUType(int);</code>	
<code>String getMouseType();</code>	
<code>int getDisplayColorDepth();</code>	
<code>String getDisplayType();</code>	
<code>String getKeyboardType();</code>	
<code>String getNetworkType();</code>	
<code>int getNumParallelPorts();</code>	
<code>int getNumSerialPorts();</code>	
<code>Dimension getScreenSize();</code>	
<code>short getScreenDepth();</code>	
<code>short getScreenResolution();</code>	
<code>String getOSLanguage();</code>	
<code>String getOSType();</code>	
<code>String getTerminalArchitecture();</code>	
<code>String[] getAudioDrivers();</code>	

String[] getMIDIDrivers();	
String[] getVideoDrivers();	
String getModemType();	
long getFreeTerminalMemory();	
long getTotalTerminalMemory();	
long getTotalApplicationMemory();	
long getFreeApplicationMemory();	
long getTerminalLoad();	
long getNetworkLoad();	
short getSceneDescriptionProfile();	
short getVisualProfile();	Retorna um <i>profile</i> objetos visuais de acordo com as especificações do MPEG-4 Visual Profile
short getAudioProfile();	Retorna um <i>profile</i> de áudio de acordo com as especificações do MPEG-4 Audio Profile
short getSceneDescriptionProfile();	Retorna um <i>profile</i> de descrição de cena, por exemplo, utilizando esquemas MPEG-7, de acordo com as especificações do MPEG-4 Scene Description Profile
short getODProfile();	Retorna um <i>profile</i> de descritor de objeto de mídia, de acordo com o MPEG-4 Object Descriptor definido em ODProfileLevelIndication
short getGraphicsProfile();	Retorna um <i>profile</i> apresentação gráfica de acordo com as especificações do MPEG-4 Visual Profile definido em graphicsProfileLevelIndication
short getMPEGJProfile();	Retorna um <i>profile</i> do MPEG-J de acordo com as especificações da API MPEG-J no MPEG-4 Systems e definido em JProfileLevelIndication

O pacote **Network (net)** possui classes e interfaces para suportar a manipulação de todos os recursos de redes de comunicação vinculados a um *player* MPEG-4, conforme pode ser visto na **Tabela 5.08**. As questões de QoS são especialmente tratadas neste pacote, principalmente em termos do controle dos canais de transmissão de *elementary streams*.

Tabela 5.08 – Interfaces e classes do pacote Network

Interfaces	Descrição
NetworkManager	
ChannelDescriptor	Interface que permite recuperar todas as informações sobre um determinado canal especificado em uma sessão audiovisual
DMIFMonitor	
ChannelController	
ServiceSessionDescriptor	Interface que permite recuperar todas as informações sobre uma sessão audiovisual
QoSDescriptor	
Classes	
Qualifier	
ChannelsDescriptor	
QoSData	
QualifierTag	Classe que contém as constants que definem as métricas de QoS em uma sessão audiovisual, de acordo com as especificações DMIF MPEG-4

O pacote **Decoder** possui classes para a manipulação de decodificadores de mídia reconhecidos por um *player* MPEG-4, conforme pode ser visto na Tabela 5.09. Cada tipo de objeto de mídia (identificados por um **ObjectDescriptor**) pode estar associado a um conjunto (vetor de decodificadores) de decodificadores válidos, o que permite a personalização a partir de ajustes nos parâmetros de QoS.

Tabela 5.09 – Interfaces e classes do pacote Decoder

Interfaces	Descrição
MPDecoder	Interface que é a base para instanciação de um <i>decoder</i> a um ou mais objetos de mídia.
DecoderType	Interface para acesso aos tipos de <i>decoder</i> válidos para um determinado objeto de mídia em relação ao contexto de uma cena
Exceções	
MediaDecoderException InvalidDecoderLevelException DecoderNotFoundException InvalidDecoderPriorityException InvalidDecoderSpeedException InvalidDecoderModeException DecoderNotRunningException DecoderAlreadyAttachedException DecoderNotAttachedException DecoderAlreadyRunningException InvalidDecoderTypeException	

O pacote **Scene** possui interfaces para descrever todas as interações possíveis em uma cena MPEG-4. Essas interações são suportadas pelo controle de eventos (*event_in* e *event_out*),

inserção e controle de sensores de cena, e monitoramento dos objetos de mídia da cena, conforme pode ser visto na Tabela 5.10.

Tabela 5.10 – Interfaces do pacote Scene

Interfaces	Descrição
Scene	Interface que atua como um <i>proxy</i> para um BIFS de uma cena
Node	Interface que atua como um <i>proxy</i> para um nó de um BIFS no grafo de nós da cena
SceneListener	Interface que permite o monitoramento das mudanças no BIFS de uma cena
NodeType	Interface que define constantes para os tipos de nós permitidos pelo MPEG-4 <i>Systems</i>
NodeValue	Interface que define um valor para um nó de um BIFS de uma cena ou especifica um novo valor quando da criação de um novo nó
FieldValue	Controlador que verifica se um objeto de mídia pode retornar valores de suas propriedades
Field	Interface que controla o acesso a campos de dados associados às propriedades de um nó
Field.Anchor	Define constantes para o defID Anchor em um nó
Field.ProximitySensor2D	Define constantes para o defID ProximitySensor2D em um nó
Field.ProximitySensor	Define constantes para o defID ProximitySensor em um nó
Field.TimeSensor	Define constantes para o defID TimeSensor em um nó
Field.TouchSensor	Define constantes para o defID TouchSensor em um nó
EventOut	Interface que controla a desativação de um controlador de evento em uma cena
EventOut.Anchor	
EventOut.TimeSensor	
EventOut.TouchSensor	
EventOutListener	Interface que implementa a notificação de desativação de eventos
EventIn	Interface que controla a ativação de um controlador de evento em uma cena
EventIn.Anchor	
EventIn.ProximitySensor2D	
EventIn.ProximitySensor	
EventIn.TimeSensor	
EventIn.TouchSensor	
EventIn.VisibilitySensor	
EventIn.WorldInfo	
Exceções	
InvalidSceneException	
BadNodeException	
BadParameterException	
InvalidNodeException	

5.6.2 Exemplos de aplicação do código MPEG-J

Um primeiro exemplo de aplicação do código MPEG-J permite visualizar as potencialidades para controle do grafo de uma cena composta por objetos de mídia MPEG-4. Pelo exemplo, ilustrado na **Figura 5.03**, observa-se que o MPEGlet pode acessar, modificar e criar nós no grafo de descrição da cena.

```
import org.isc.mpeg.mpegj.resourceManager.*;
import org.isc.mpeg.mpegj.scene.*;
import org.isc.mpeg.mpegj.*;
import org.isc.mpeg.mpegj.net.*;

public class SceneExample implements MPEGlet
{
    MpegjTerminal mpegjTerminal;
    Scene m_scene;
    MySceneListener m_sceneListener;
    MyTranslate m_translate;
    MyEventOutListener m_eventOutListener;
    MyNodeCreator m_nodeCreator;

    public void init()
    {
        mpegjTerminal = new MpegjTerminal(MPEGlet);
        // all the initialization goes here for net and resource managers

        //getting a reference to the scene object from the MpegjTerminal object
        try
        {
            m_scene = m_sceneListener.getScene( mpegjTerminal );
        }
        catch( MPEGJException mpegj_exp )
        {
        }
        catch ( InterruptedException ie )
        {
        }
    }

    public void run()
    {
        //to access and modify a field in the BIFS scene
        try
        {
            m_translate.translate(m_scene);
        }
        catch( MPEGJException mpegj_exp )
        {
        }

        //to receive notifications of changes to the scene
        try
        {
            m_eventOutListener.register(m_scene);
        }
    }
}
```

```

        catch( MPEGJException mpegj_exp )
        {
        }

        // creating and adding a new node (in this case, a Box node) to the Scene
        try
        {
            m_nodeCreator.createBox(m_scene);
        }
        catch( MPEGJException mpegj_exp )
        {
        }
    }

    public void stop()
    {
    }

    public void destroy()
    {
    }
}

```

Figura 5.03 – Exemplo MPEGlet para manipular os nós de um grafo de uma cena MPEG-4 (ISO, 1999a)

Um segundo exemplo de aplicação do código MPEG-J permite compreender a implementação de sensores de eventos (*listeners*). Esses sensores podem ser acoplados aos *TouchSensors* descritos no MPEG-4 *Systems*. Pelo exemplo, ilustrado na **Figura 5.04**, observa-se que o MPEGlet apresenta um mecanismo para se obter uma referência a um objeto de mídia em uma cena MPEG-4, considerando-se também as propriedades do objeto **MpegjTerminal**. Neste ponto, é relevante observar que essa abordagem permite estabelecer eventos de acordo com as potencialidades do terminal em uso.

```

import org.iso.mpeg.mpegj.*;
import org.iso.mpeg.mpegj.scene.*;

// Gets the scene from the MpegjTerminal
public class MySceneListener implements SceneListener
{
    private Scene m_scene;

    // waits for and returns the scene associated with the given MpegjTerminal.

    // @param terminal the MpegjTerminal containing the scene
    // @return the scene
    // @exception MPEGJException if an MPEG-J failure occurs
    // @exception InterruptedException if interrupted while waiting for the scene

    public synchronized Scene getScene(MpegjTerminal terminal)
        throws MPEGJException, InterruptedException
    {
        SceneManager mgr = terminal.getSceneManager();
    }
}

```

```

        mgr.addSceneListener(this);

        while (m_scene == null)
        {
            wait();
        }

        return m_scene;
    }

    // Called back by the SceneManager to provide the scene.
    public synchronized void notify(int what, Scene scene)
    {
        if (what == SceneListener.Message.SCENE_READY)
        {
            m_scene = scene;
            notify();
        }
    }
}

```

Figura 5.04 – Exemplo MPEGlet para adicionar eventos a uma cena MPEG-4 (ISO, 1999a)

Um terceiro exemplo de aplicação do código MPEG-J mostra como acessar e manipular um campo de dado descrito no BIFS de uma cena MPEG-4. Neste caso, a proposta é modificar o parâmetro de translação de um nó **Transform2D** no grafo da cena. A modificação do valor é feita através do envio de uma mensagem ao controlador **eventIn**, que aciona o método **scene.getNode()** com um argumento 1, que corresponde ao campo DEF ID do nó **Transform2D** em uma cena MPEG-4. A Figura 5.05 apresenta o MPEGlet para manipulação de BIFS.

```

import org.iso.mpeg.mpegj.*;
import org.iso.mpeg.mpegj.scene.*;

// Translates the translation field of a DEF 1 Transform2D node.
public class MyTranslate
{
    //Translates the translation field of node 1

    //@param scene the scene
    //@exception MPEGJException if an MPEG-J failure occurs

    public void translate(Scene scene) throws MPEGJException
    {
        Node node = scene.getNode(1);
        // Check that the node is a Transform2D.

        if (node.getNodeType() != NodeType.Transform2D)
        {
            System.err.println("Node 1 is not a Transform2D!");
            return;
        }
    }
}

```


Em linhas gerais, a infra-estrutura de comunicação proposta pelo modelo pelo **NECTAR** visa melhorar o suporte à interatividade entre o usuário e a aplicação como um todo, considerando o fato de que a aplicação interativa é parte integrante de um ambiente consciente de contexto. Essa infra-estrutura é composta por três componentes distintos: o *servidor*, a *rede de comunicação* e o *terminal*.

O *servidor* cobre as operações de armazenamento e envio de informações multimídia. Além disso, o *servidor* também realiza operações que permitem enviar objetos multimídia diferentes a usuários com preferências e interesses diferentes. Em termos conceituais, o *servidor* deve armazenar os objetos multimídia que compõem a um programa interativo e a estrutura desse programa, no que diz respeito aos objetos das cenas, as informações temporais e espaciais, e os relacionamentos entre os objetos. Nesta proposta, um **programa interativo** é definido como sendo o conjunto completo dos componentes de mídia e suas relações como o ambiente e com o usuário, sendo que o programa interativo é, então, parte integrante de uma aplicação multimídia interativa. Essa aplicação, por sua vez, é parte integrante do ambiente.

No contexto deste trabalho, o conteúdo multimídia é baseado em objetos MPEG-4 (Pascoe et al., 1999). Por outro lado, a estrutura dos programas interativos é representada por descrições MPEG-7 (Wu et al., 1997; DeRose & Durand, 1994), gerando instâncias da modelagem. Neste sentido, as descrições MPEG-7 oferecem as semânticas necessárias ao conteúdo codificado em MPEG-4, permitindo novas formas de interatividade e o uso eficiente das informações contextuais para adaptação do programa interativo às condições do servidor, por exemplo (Ebrahimi & Horne, 2000).

A *rede de comunicação*, é configurada de forma a oferecer um suporte flexível às condições de tráfego e permitir o gerenciamento da qualidade da apresentação ao usuário. Tradicionalmente, a qualidade de serviço (QoS) em uma rede de comunicação é limitada pelos esquemas de roteamento e controle da qualidade dos dados que trafegam nessa rede.

Este trabalho sugere o uso de consciência de contexto também ao nível das redes de comunicação (*Context-Aware Networks*), apresentando mecanismos para uso de regras de contexto entre as entidades que compõem a rede de comunicação e oferecendo novos esquemas de controle de QoS em duas formas diferentes: a) para descobrir entidades de rede

que sejam capazes de melhorar o processamento sobre os dados que trafegam em direção ao usuário; b) para que a aplicação do usuário possa interferir nas configurações da rede de comunicação, oferecendo, por exemplo, dados contextuais que possam melhorar o desempenho da própria rede (Li et al., 1998).

O uso de regras de consciência de contexto em nível de rede de comunicação apresenta-se como uma nova perspectiva para os esquemas de interação *usuário-rede* e *aplicação-rede*, sendo, portanto, uma inovação para a construção de aplicações interativas conscientes de contexto.

Aspectos de consciência de contexto podem ser potencialmente usados no processamento de informações contextuais sobre a rede de comunicação, sobre o servidor e sobre as aplicações, permitindo a tomada de decisões de forma dinâmica sobre as condições mais adequadas à distribuição e apresentação de programas interativos, considerando, por exemplo, o tráfego, a largura de banda, a qualidade necessária para a apresentação, o nível de interatividade exigido, as características e necessidades do programa e do usuário. Essas decisões podem ser tomadas no momento em que ocorrerem eventos relevantes ao contexto, sem necessidade de intervenção ou consulta ao servidor. Desta forma, pode-se dizer que a rede de comunicação passa a ser consciente do contexto da aplicação interativa.

O componente *terminal* é responsável por todas as interfaces diretas entre o usuário e a aplicação interativa. Em linhas gerais, as principais funções concentram-se em receber e processar as informações contextuais e negociar dinamicamente, com o servidor e com a rede de comunicação, os requisitos e *status* da aplicação.

O *terminal* é também responsável por exibir as cenas que formam o programa interativo, cenas essas compostas por objetos MPEG-4. Desta forma, o terminal possui um player MPEG-4, implementado segundo as recomendações MPEG-4 Systems, conforme visto na Seção 5.5.

Conforme discutido, a API MPEG-J pode ser usada como interface entre o player Mpeg-4 e programas JAVA (Nack & Lindsay, 1999), acrescentando possibilidades de manipulação das informações contextuais, por exemplo.

Observa-se que a infra-estrutura proposta para aplicações interativas gera um complexo universo de novas possibilidades de interação entre os componentes dessa infra-estrutura. Os relacionamentos representados pelas setas em vermelho indicam os pontos nos quais se faz necessária a comunicação entre agentes de software; por outro lado, as setas em preto indicam os relacionamentos entre elementos de hardware e software que compõem o ambiente interativo consciente de contexto.

Observando-se a infra-estrutura proposta, a comunicação entre agentes pode ser aplicada a dois propósitos distintos: a) para coletar informações contextuais dos diversos componentes da infra-estrutura; b) para operações de personalização das interfaces com o usuário.

5.7.1 Detalhando a infra-estrutura do servidor

Em linhas gerais, o servidor é responsável por armazenar e distribuir os programas interativos. Esses programas são compostos por objetos multimídia, tais como vídeo, áudio, imagens, animações, gráficos e textos, dentre outros. Além disso, um programa interativo possui também *scripts* que especificam a estrutura do programa e as relações espaciais e temporais entre os objetos que formam as cenas desse programa. Generalizando, um programa interativo é composto por conteúdo multimídia e descrição da estrutura do conteúdo multimídia.

O servidor também deve ser capaz de distribuir objetos multimídia distintos para usuários com preferências e necessidades distintas.

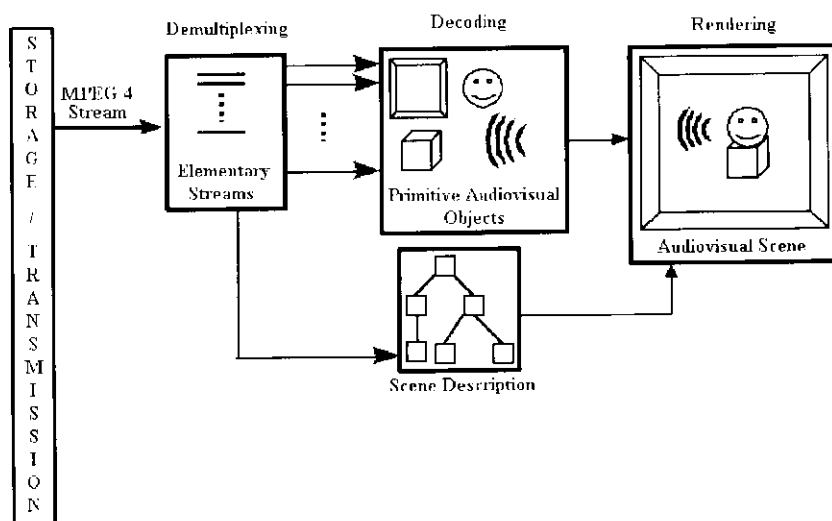


Figura 5.06: Esquema para decodificação de *streams* MPEG-4 (ISO, 2001a)

O padrão MPEG-4 oferece o suporte necessário para o conteúdo dos objetos multimídia, tendo sido desenvolvido para prover soluções aos novos tipos de aplicações interativas (Battista et al., 1999). O potencial e a funcionalidade do MPEG-4 são baseados na manipulação – codificação, decodificação e interação – de mídias como objetos, conforme é possível observar na **Figura 5.06**.

Uma apresentação MPEG-4 é baseada em cenas. Cada cena é composta por diversos objetos de mídia (vídeo, áudio, gráficos 2D e 3D, animações 2D e 3D, texto, imagens, dentre outros) e esses objetos são codificados em *streams* (*Elementary Streams* – ES). *Streams* MPEG-4 são decodificadas, permitindo a separação e reconstrução dos objetos, tornando possível a interação com cada objeto que compõe a cena, conforme ilustra a **Figura 5.06**. Neste sentido, uma possível lista de interações contém operações como alteração da posição e forma dos objetos na cena; operações de escala e rotação; alteração da velocidade de apresentação dos objetos; adição e remoção de objetos de uma cena (ISO, 2001a).

Em termos de codificação, o padrão MPEG-4 suporta diversos tipos de algoritmos e taxas de compressão. Os objetos passam por um processo de compressão e podem se multiplexados em um ou mais *streams*, tanto para armazenamento quanto para transmissão em uma rede de comunicação.

O processo de composição de cenas é bastante flexível com o uso de MPEG-4, permitindo a criação de apresentações multimídia (naturais ou sintéticos) com base em um grafo das cenas que compõem essa apresentação. Os objetos de mídia, por sua vez, são organizados hierarquicamente, permitindo que objetos de tipos mais complexos sejam formados a partir de objetos dos tipos primários.

Cada objeto de mídia é uma entidade independente, e pode ser sincronizado com outros objetos que compõem a cena. A sincronização em MPEG-4 é feita em nível de *frames* para todos os objetos. Isto é uma vantagem em relação aos mecanismos de sincronização usados por tecnologias como SMIL (W3C, 1998) e HyTime (ISO, 1992; DeRose & Durand, 1994), que geram sincronização ao nível de *streams* ou grupos de *frames*. Os mecanismos de sincronização são extremamente importantes em programas interativos, como é o caso dos programas de TVI, visto que esta proposta considera que os usuários possam manipular os

objetos durante a exibição das cenas. Além disso, o suporte automático da consciência de contexto também envolve diretamente os mecanismos de sincronização.

Outra característica interessante do padrão MPEG-4 é que os objetos de vídeo são compostos por camadas (*layers*) (Ebrahimi & Horne, 2000). Existe uma camada básica e várias camadas complementares que permitem escalabilidade temporal e espacial. As camadas complementares visam melhorar a qualidade da camada básica em termos da resolução e da taxa de transmissão de *frames*. Neste sentido, MPEG-4 permite a composição de cenas como objetos estruturados e relacionados entre si. Adicionalmente, características como facilidades de *streaming*, alto potencial de compressão, sincronização, dentre outras, fazem com que o MPEG-4 apresente-se como potencial candidato para implementação de programas de TVI. Adicionalmente, a estrutura do conteúdo MPEG-4 pode ser descrita com o uso de descrições MPEG-7 (Nack & Lindsay, 1999; ISO, 2000a), que representam instâncias de um modelo de ambiente interativo. Neste ponto, descrições MPEG-7 podem ser codificadas no mesmo *stream* que contém o conteúdo multimídia, permitindo o estabelecimento de diversos níveis de interação, especialmente aqueles associados aos mecanismos de consciência de contexto.

MPEG-7 tem por objetivo padronizar as descrições de dados referentes ao conteúdo multimídia, representado, por exemplo, em MPEG-4. Essa padronização é feita através de um conjunto de “Descritores” (*Descriptor - D*), que podem ser usados para descrever qualquer tipo de objeto de mídia. O padrão MPEG-7 especifica também estruturas de descritores pré-definidas e seus relacionamentos na descrição de cenas, oferecendo suporte também para que o usuário possa definir suas próprias estruturas. Tais estruturas são denominadas “Esquemas de Descrição” (*Descriptors Schemes - DS*).

Para a definição de um novo DS, deve-se utilizar uma linguagem específica para esse propósito, denominada “Linguagem de Definição de Descrição” (*Description Definition Language - DDL*) (ISO, 2001c). Usando uma DDL, DSs e Ds, o padrão MPEG-7 pode acrescentar ao conteúdo multimídia as mesmas características que XML (W3C, 2000) acrescenta ao conteúdo textual.

Conforme citado, os objetos MPEG-4 podem ser descritos com o uso do MPEG-7. Esta possibilidade abre um cenário interessante para o conteúdo de programa de TVI, uma vez

que oferece todas as semânticas necessárias para o gerenciamento eficiente do conteúdo, para a construção de mecanismos de pesquisa e para o desenvolvimento de novos níveis de interatividade. Deve-se observar também que MPEG-7 não é limitado à descrição de conteúdo de objetos multimídia, podendo ser aplicado também para construção de níveis mais abstratos, como é o caso de um programa interativo (um programa de TVI, por exemplo). Um exemplo dessa aplicação pode ser visto no código seguinte na **Figura 5.07**.

```
<DSType name="Program">
  <!--core-->
  <attribute name="identifier" type="integer">
  <attribute name="member_of" type="GroupRefType" minOccurs="0"
    maxOccurs="unbounded"/>
  <attribute name="generic program" type="GroupRefType" minOccurs="0"
    maxOccurs="unbounded"/>
  <attribute name="episode" type="SeriesRefType" minOccurs="0"
    maxOccurs="unbounded"/>
  <! program basic information -->
  <attribute name="title" type="ProgramTitleType" minOccurs="1"
    maxOccurs="unbounded"/>
  <attribute name="abstract" type="ProgramAbstractType" minOccurs="0"
    maxOccurs="unbounded"/>
  <attribute name="keyword" type="ProgramKeywordType" minOccurs="0"
    maxOccurs="unbounded"/>
  <attribute name="genre" type="ProgramGenreType" minOccurs="0"
    maxOccurs="unbounded"/>
</DSType>
```

Figura 5.07 – Um exemplo de DS (Description Schema) MPEG-7

Observa-se que o DS apresenta uma forma para estruturar um programa de TVI e quais os elementos, em mais alto nível, que compõem esse programa. A composição é determinada pela identificação do programa, pela associação de elementos (grupo, episódio) e pelo índice de informações (título, palavras-chave, dentre outros). Nota-se, ainda, que no DS existe um atributo *member_of* que é capaz de estabelecer um *link* com outros DS.

No cenário dos ambientes interativos conscientes de contexto, é interessante que o servidor possa se adaptar às necessidades dos serviços sob demanda. Esses serviços, tanto para um usuário quanto para um grupo de usuários, deve possuir, dentre muitas características, as seguintes:

- permitir a distribuição via *streams*;
- realizar a multiplexação de dados;
- permitir ajustes nos *streams* durante a distribuição;
- possibilitar o armazenamento e a recuperação de programas;
- realizar pesquisas com base nas descrições de conteúdo.

Os parâmetros de informação, necessários aos procedimentos de adaptação e personalização, devem ser fornecidos pela aplicação do cliente, tais como o nível de qualidade desejado, por exemplo; pela rede, tais como as condições de tráfego, banda disponível, dentre outros; e pelo servidor, tais como os dados disponíveis em determinado momento e as descrições associadas a esses dados.

O modelo de ambientes interativos conscientes de contexto apresenta as situações em que os parâmetros de personalização são necessários, especialmente em termos dos elementos de informações presentes no *Terminal Mode*. Neste ponto, esses elementos de informação são também instanciados como descrições MPEG-7, mantendo a compatibilidade com as descrições de conteúdo presentes para os objetos de mídia. Adicionalmente, os elementos de informação são usados para representar informações contextuais (*context-awareness*), tais como identidade, localização, atividade e tempo, informações essas que podem ser mapeadas por descritores e esquemas de descrição do MPEG-7.

Num cenário de TVI, diferentes usuários podem possuir diferentes tipos de preferência e níveis de qualidade para uma mesma apresentação. Desta forma, o servidor deve ser capaz de distribuir diferentes *streams* de um mesmo programa interativo. No caso particular de vídeo, a tecnologia utilizada deve ser baseada em codificação em camadas, como é o caso do MPEG-4, em que o sinal de vídeo é codificado em uma ou mais camadas com diferentes níveis de prioridade (Wu et al., 1997; McCanne et al., 1996; Li et al., 1998; Ebrahimi & Horne, 2000). Assim, o servidor pode usar codificadores MPEG-4 para oferecer as funcionalidades baseadas em camadas para os objetos de vídeo.

Completando a infra-estrutura do servidor, agentes de software podem ser usados para controlar e monitorar os processos de distribuição dos programas interativos. Esses agentes podem acessar as descrições de programa, os requisitos de interação do usuário e os parâmetros do servidor e da rede de comunicação, usufruindo dos recursos da API MPEG-J, por exemplo. De posse desses dados, os agentes podem executar operações de personalização em todo o ambiente. Como exemplo, pode-se pensar no caso de balanceamento da carga do servidor, em que agentes podem monitorar a execução das tarefas de entrega dos programas interativos e indicar possibilidades de interrupção no serviço. Nesta situação, os agentes podem consultar as descrições do programa interativo para tentar encontrar alternativas para os objetos que compõem o programa, como, por

exemplo, selecionar um novo codificador de vídeo que seja capaz de codificar um número menor de camadas, reduzindo assim o fluxo de dados na rede de comunicação, porém conservando a qualidade da apresentação ao nível do usuário.

5.7.2 Detalhando a infra-estrutura do cliente

O modelo do cliente é composto pelos agentes de terminais. O terminal é composto por um computador pessoal, conectado a uma rede, e deve exibir programas interativos, contendo um *player* MPEG-4 como a base para essa exibição.

A Figura 5.08 mostra os componentes envolvidos no processo de exibição de uma apresentação MPEG-4 pelo terminal do cliente (como é o caso de um programa de TVI). Os *streams* de mídia codificados em MPEG-4 são enviados ao terminal. Através de um demultiplexador, esses *streams* Mpeg-4 são identificados e separados como *streams* elementares. Cada *stream* elementar é decodificado em objetos de mídia e enviado ao *player* para sua renderização. O *player* utiliza a descrição da cena e as descrições dos objetos de mídia e verifica que tipos de controle podem ser aplicados a cada objeto.

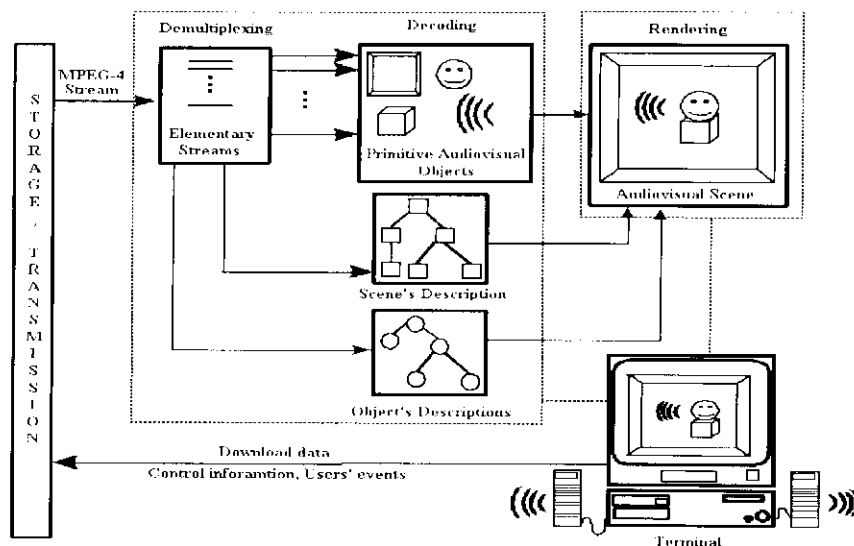


Figura 5.08: Esquema para apresentação de objetos MPEG-4 em um terminal (ISO, 2001a)

O autor de uma apresentação multimídia é quem determina esses controles que podem ser aplicados aos objetos de mídia. Cada objeto tem uma lista de controles padrão, além dos controles adicionados durante a autoria. Os tipos mais complexos de controle podem ser conseguidos usando *MPEGlets* desenvolvidos com o uso da API MPEG-J (ISO, 1999d), que

age como uma interface entre o *player*, os objetos de mídia e a aplicação como um todo, que pode ser desenvolvida em Java.

Os usuários podem interagir com a apresentação de diversas formas, partindo-se do princípio do controle de objetos na cena e das possibilidades de *searching* de características de cada objeto. Neste cenário, as interações dos usuários geram eventos. Alguns eventos têm um efeito local, como a amplificação de uma imagem, e são fáceis de processar. Outros podem afetar a rede ou os serviços do usuário, como é o caso de uma requisição por uma taxa de compressão de *frames*, por exemplo.

Os terminais podem ser conectados também a dispositivos como VCRs, estações de trabalho, terminais de TV, câmeras, dentre outros. Estes dispositivos podem possuir sensores, que capturam algum tipo da informação contextual e fornecem suporte à ubiquidade e à consciência de contexto. Neste sentido, um sensor comunica-se com os dispositivos que são controlados pelos terminais.

5.7.3 Detalhando a infra-estrutura da rede de comunicação

Um dos focus desta proposta é centrado no fato de que a rede de comunicação deve estar consciente do contexto da aplicação interativa e do ambiente como um todo (usuários, programa interativo e interações usuário-rede-aplicação), oferecendo novas possibilidades de gerenciamento das aplicações. Essas possibilidades são relacionadas às interações entre os três componentes da infra-estrutura. Para se atingir esse novo nível de gerenciamento da rede, são necessários protocolos e algoritmos que forneçam informações gerenciáveis, como é o caso do MPEG-4.

Tradicionalmente, o processamento de informações da rede é limitado à distribuição, ao controle de congestionamento e à qualidade de serviço (QoS). As novas arquiteturas de rede, como é o caso das **redes ativas**, permitem a mudança dinâmica do comportamento da aplicação. Neste sentido, os roteadores podem processar dados contextuais do usuário e da aplicação (Tennenhouse & Wetherall, 1996; Calvert et al., 1998).

O trabalho de doutorado de Goularte investiga o uso de diferentes tecnologias de rede para implementar os requisitos definidos por esta proposta, especialmente em termos do uso de informações contextuais (Goularte, 2001). O foco da proposta de Goularte é permitir que a rede tome decisões dinâmicas, baseadas em informações contextuais e regras de contexto

modeladas nesta proposta, visando melhorar a distribuição de programas interativos, sobretudo em termos do tráfego, da largura de banda, da qualidade da apresentação, do nível de interatividade, das necessidades dos programas interativos e das interações do usuário. Essas decisões podem ser tomadas no momento em que determinados eventos ocorrem, sem necessidade da intervenção do usuário ou de respostas do usuário. Neste sentido, este trabalho contribui com a proposta de Goularte quando oferece requisitos necessários para determinar que informações contextuais são relevantes às decisões da rede e que decisões contribuem realmente para a adaptação da aplicação.

Atualmente, o *toolkit* ANTS tem oferecido suporte a experimentos com redes ativas (Wetherall et al., 1998), permitindo o desenvolvimento de aplicações JAVA para o monitoramento da rede, o que é interessante para esta proposta, que considera o uso da plataforma JADE e da API MPEG-J, tecnologias baseadas em Java.

5.8 PRACTIC: o modelo da personalização

A abordagem atual para o desenvolvimento de aplicações interativas, em nível mundial (ATVEF, 2001; DVB, 2000; TVAF, 2001a), prevê somente a adição de alguns aspectos de interatividade aos tradicionais programas interativos. No entanto, com a mesma tecnologia que esses modelos utilizam, é possível obter um resultado com maior grau de adaptabilidade ao consumidor e que também permita uma melhor definição do público-alvo pelos desenvolvedores de aplicação.

Neste ponto, observa-se que é possível adaptar um mesmo programa interativo, com pequenas modificações, a espectadores com diferentes interesses, conforme ilustra a **Figura 5.09**.

Na **Figura 5.09** é apresentado um programa interativo com seus diversos objetos, porém para cada usuário ele será mostrado de uma forma diferente, dependendo dos perfis que esse usuário tem à disposição. No **Capítulo 6**, será descrita uma aplicação de TVI como exemplo dos conceitos aqui discutidos.

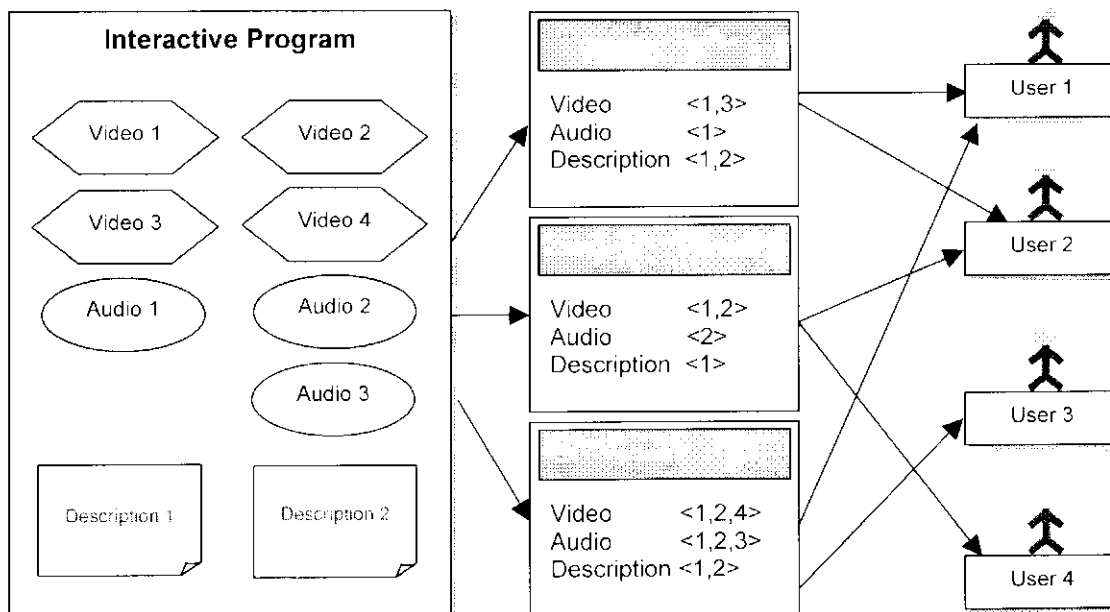


Figura 5.09: Estrutura proposta para um programa interativo

Neste contexto, é importante ressaltar que esse tipo de adaptação, considerando-se uma estrutura particular para um programa interativo, pode ser feito com o uso de aspectos de consciência de contexto. As Seções 5.8.1 e 5.8.2 apresentam descrições de programa e de perfil, respectivamente.

5.8.1 O programa interativo

Nesta proposta, o programa é o elemento central de uma aplicação interativa. O programa interativo pode ser definido como o conjunto de todas as mídias (e suas descrições), cenas (e suas descrições), um sumário (ou sinopse) e sua programação, conforme mostra a Figura 5.10.

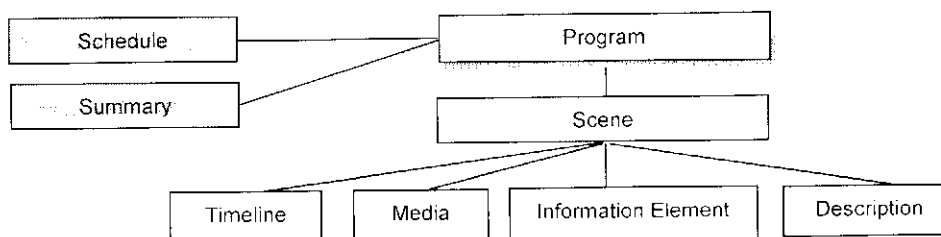


Figura 5.10: Entidades que compõem um programa interativo em uma aplicação

O programa interativo é estruturado usando a DDI MPEG-7 (ISO, 2001b) e utiliza as descrições multimídia definidas pelo MPEG-7 (ISO, 2001c). As entidades *Schedule* e *Summary* são definidas pelo TV-Anytime Forum (TVAF, 2001b; TVAF, 2001c). A seguir, uma versão simplificada do Schema de descrição de uma cena é apresentada na **Figura 5.11**. A versão completa para cenas (e programas) é encontrada em (PRACTIC, 2001b) e também no **Capítulo 6**.

```
<element name="Scene">
  <complexType complexContext="true">
    <element name="Media">
      <complexType complexContext="true">
        <attribute name="type" type="mpeg7:DSType"/>
        <attribute name="id" type="string"/>
        <attribute name="filelocation" type="mpeg7:DSType"/>
      </complexType>
    </element>
  </ComplexType>
</element>
```

Figura 5.11 – Schema MPEG-7 para descrição de uma cena

A partir das mídias e suas descrições, é possível construir diferentes interfaces, ou modos de apresentação, para um programa interativo. A esta possibilidade, dá-se a denominação de perfil (*profile*). Para a codificação das mídias e das cenas, esta proposta considera o uso da tecnologia MPEG-4, que é bastante flexível e eficiente para a compactação de vídeo e de áudio. Como característica fundamental do MPEG-4 pode-se citar o fato dos componentes de uma cena estarem separados como objetos, permitindo assim incrementos nas formas de manipulação desses componentes. Essa característica é bastante significativa para o caso da implementação de diferentes perfis de programas interativos. É importante salientar que apesar deste trabalho considerar o uso de MPEG-4 para a apresentação e transmissão do programa, a estrutura apresentada é independente do mecanismo de transporte, podendo utilizar os padrões da TV Digital (ATSC, 2001; DVB, 1999; ARIB, 2000), bem como pode utilizar o Multimídia Home Platform (DVB, 2000) para a apresentação do conteúdo multimídia.

As descrições das cenas e das mídias permitem a criação de ferramentas de busca em vídeos, áudio e nas cenas. Pelo fato deste trabalho utilizar o padrão MPEG-7 para a descrição destas, deve-se observar que ferramentas de busca para MPEG-7 poderão ser utilizadas.

Por outro lado, o *sumário* e o *schedule* têm uma destacada importância para aplicações do tipo EPG (*Eletronic Program Guides*), uma vez que é a partir do uso de EPGs que o usuário escolherá o que assistir e o que pretende gravar. Com as informações do sumário, o usuário pode escolher exatamente os programas a serem assistidos/gravados; com a informação do *schedule*, o usuário não precisa se preocupar com o horário de início do programa, uma vez que seu STB fará isso de forma transparente. Outro relevante uso do *schedule* reside no fato de que em um ambiente consciente de contexto e ubíquo o STB pode informar um equipamento telefônico ou PDA o horário de início de um determinado programa ao vivo.

5.8.2 O perfil de programa interativo

Um programa interativo precisa de uma especificação de modo de apresentação. Pelo fato de existir uma vasta quantidade de mídias disponíveis, o número de diferentes formas de apresentações desse programa tende a ser igualmente grande. Quem disponibiliza os perfis são as emissoras geradoras de programas, porém estes perfis podem ser personalizados pelos usuários. Ao fazer com que as produtoras disponibilizem os perfis, impede-se a incompatibilidade entre um perfil e um programa, além de permitir que diferentes perfis possam ser disponibilizados de acordo com o tipo e/ou categoria de usuário (por exemplo, o plano de canais interativos que um determinado usuário possui).

Para a descrição dos perfis, este trabalho utiliza o padrão MPEG-7 e a criação de novos elementos, acessível em (PRACTIC, 2001b), para descrever alguns aspectos de *layout*, requisitos e contexto, relevantes a esse perfil e suas interações, conforme ilustra a **Figura 5.12**. As informações relativas a um perfil estão dentro de um elemento <profile>. Esse elemento, por sua vez, é composto por outros elementos, que terão suas funcionalidades apresentadas a seguir.

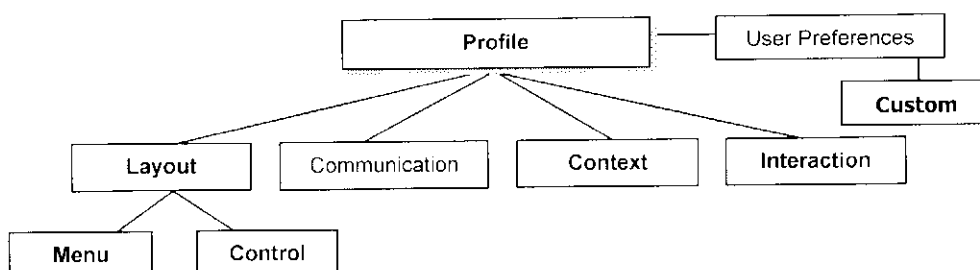


Figura 5.12: Entidades que compõem um programa interativo em uma aplicação

Os aspectos de apresentação definidos, que fazem parte do elemento <Layout>, são relativos às cores de fundo, das fontes e posições das mídias. Dentro do *layout* ainda, tem-se os controles de mídia possíveis para um determinado perfil, o tipo de menu e seu conteúdo. Uma versão simplificada do Schema de descrição do elemento <layout> é apresentada na **Figura 5.13**. A versão completa do Schema pode ser encontrada em (PRACTIC, 2001b) e também no **Capítulo 6**, quando é discutida uma instância de aplicação interativa, no caso a TVI.

```

<element name="Layout">
  <complexType complexContext="true">
    <attribute name="bgcolor" type="string"/>
    <element name="Object">
      <complexType complexContext="true">
        <attribute name="id" type="string"/>
        <element name="ObjectPosition">
          </element>
        <element name="ObjectSound">
          </element>
        </complexType>
      </element>
    <element name="Menu">
      </element>
    <element name="Control">
      </element>
    </complexType>
  </element>

```

Figura 5.13 - Schema de descrição do elemento *Layout*

Os requisitos do perfil ficam especificados e armazenados dentro do elemento <Communication>, onde estão os requisitos de comunicação de cada mídia, os decodificadores necessários para cada uma dessas mídias e requisitos do STB. Na **Figura 5.14**, é apresentada uma versão simplificada o Schema de descrição do elemento <communication>.

```

<element name="Communication">
  <complexType complexContext="true">
    <element name="CommRequirements">
      <complexType complexContext="true">
        <attribute name="medianame" type="string"/>
        <element name="Bandwith">
          <simpleType>
            <restriction base="nonNegativeInteger">
              <minInclusive value="0"/>
            </restriction>
          </simpleType>
        </element>
        <element name="Codec" type="string" />
      </complexType>
    </element>
  </complexType>
</element>

```

Figura 5.14 - Schema de descrição do elemento *Communication*

Dentro do elemento <Context>, insere-se todo o tipo de informação de contexto que pode ser relevante para um determinado perfil. Esse elemento foi definido a partir de variáveis relevantes em consciência de contexto, tais como <who>, <where>, <when> e <what> (Dey & Abowd, 2000). Esse elemento possui as ações que devem ser realizadas em determinadas situações. Na **Figura 5.15**, é apresentado um exemplo de como utilizar o Schema para a utilização em consciência de contexto.

```
<Context>
  <who = "mr. 1" > action 1 </who>
  <who = "mr. 2" >
    <where = "bedroom" >
      <when = "24:00" >
        openProgram(news1)
      </when>
      openProgram(news2)
    </where>
    openProgram(music1)
  </who>
  <what = "24:00" > turnoff() </what>
</Context>
```

Figura 5.15 – Schema de descrição do elemento Context

O último grande elemento é o <Interaction>, no qual estarão todas as ações que devem ser realizadas para cada interação definida para um perfil. Essas interações são descritas na **Figura 5.16**.

```
<element name="interaction">
  <complexType complexContext="true">
    <attribute name="on" type="string"/>
    <attribute name="media" type="pratic:Media"/>
    <attribute name="button" type="pratic:Button"/>
    <attribute name="action" type="string"/>
  </complexType>
</element>
```

Figura 5.16 – Schema de descrição do elemento Interaction

No exemplo da **Figura 5.17**, o atributo *move* é uma descrição MPEG-7, o atributo *video3* foi apresentado no programa e descrito em MPEG-7 e a ação *moveMedia(video3)* representa um comando que será dado ao STB para realizar a tarefa. Percebe-se que esse comando depende do Sistema Operacional presente no STB e das aplicações que estão presentes no mesmo, tais como MHP (Evain, 1998) ou JavaTV (SUN, 2001).

```
<Interaction>
  <on = "move" media="video3" action="moveMedia(video3)" >
  <on = "click" button="volume" action="openVolumeBox()" >
</Interaction>
```

Figura 5.17 – Schema de descrição de atributos do elemento Interaction

Além desses elementos, existe mais um elemento que segue a sintaxe dos perfis, denominado <UserPreferences>. O elemento <UserPreferences>, diferentemente dos outros elementos, fica armazenado no STB, nele estão armazenadas algumas personalizações feitas pelos usuários para um determinado perfil de um determinado programa. Um exemplo de uso desse elemento é apresentado na **Figura 5.18**.

```
<UserPreferences user="Galli">
  <MediaVolume grade="25" transformation = "ultrabass" />
  <profile="Sports3" program="NewSports">
    <layout bgcolor="ceffcc" />
  </profile>
</UserPreferences>
```

Figura 5.18 – Schema de descrição do elemento *UserPreferences*

Com os elementos apresentados e utilizando as descrições do MPEG-7 (ISO, 2001c) e do TV-Anytime Forum (TVAF, 2001b; TVAF, 2001c), pode-se especificar um amplo conjunto de características singulares, além de personalizações para os perfis e programas.

5.8.3 Utilização dos perfis em programas interativos

A utilização de perfis permite, inicialmente, uma melhor adequação de um determinado programa ao seu espectador. A **Figura 5.19** apresenta um diagrama de seqüência, em sintaxe gráfica baseada em UML (Booch et al., 1996), que mostra o funcionamento básico da ação dos perfis para a apresentação de um programa em um ambiente não consciente de contexto.

Observando-se a **Figura 5.19**, percebe-se que a seqüência de eventos tem início quando o *usuário (user)* decide qual programa interativo irá assistir; escolhido o programa, o *STB (STB)* requisita ao distribuidor (*sender*) quais são os perfis possíveis e permitidos para aquele *usuário*; o distribuidor, então, retorna esses perfis para o *STB* que apresenta as opções para o *usuário*; o *usuário* escolhe o perfil e suas modificações, e então o *STB* faz a requisição, junto ao distribuidor, dos *streams* de vídeo, áudio, dos metadados e do *stream* de controle da cena; por fim, o distribuidor enviará o programa e o perfil desejados para o *usuário*.

Uma outra utilização do conceito de perfis é em um ambiente consciente de contexto. Observando-se a **Figura 5.19**, percebe-se que a seqüência de eventos tem início quando o *usuário (user)* decide qual programa interativo irá assistir; escolhido o programa, o *STB (STB)* requisita ao distribuidor (*sender*) quais são os perfis possíveis e permitidos para

aquele *usuário*; o distribuidor, então, retorna esses perfis para o *STB* que apresenta as opções para o *usuário*; o *usuário* escolhe o perfil e suas modificações, e então o *STB* faz a requisição, junto ao distribuidor, dos *streams* de vídeo, áudio, dos metadados e do *stream* de controle da cena; por fim, o distribuidor enviará o programa e o perfil desejados para o *usuário*.

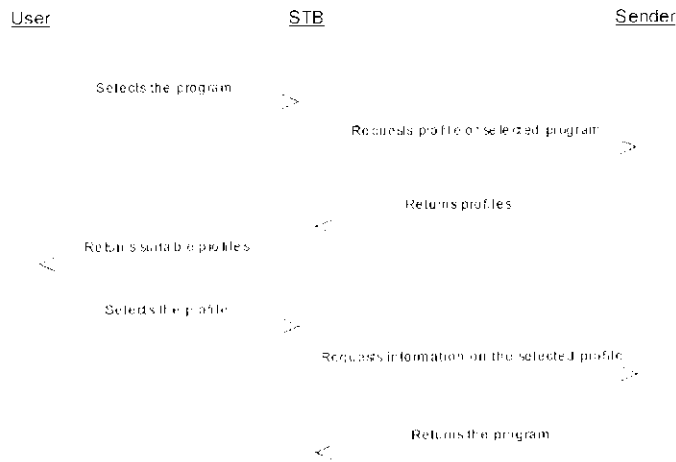


Figura 5.19: Diagrama de seqüência UML para as entidades de um programa interativo

Nesse tipo de ambiente, agentes fornecem ao STB as informações que podem ser relevantes ao perfil. O STB, por sua vez, utiliza essas informações, descritas no elemento <Context>, para a adaptação do perfil de programa.

5.8.4 Produzindo um programa interativo

Um programa pode possuir vários de **perfis**, que representam a forma e é composto por várias **cen**as, que representam o conteúdo, conforme ilustra a **Figura 5.20**.

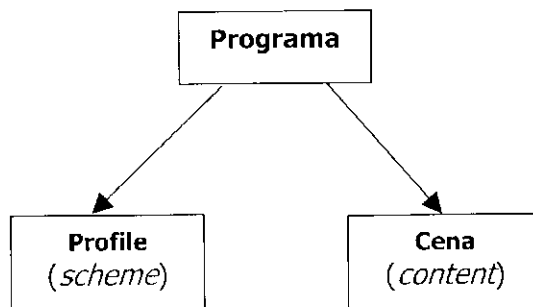


Figura 5.20 - Representação genérica da relação entre programa, perfil e cena

Um perfil de programa interativo é composto por **atores**, que podem ser: produtor, diretor, produtor de conteúdo, produtor de conteúdo *web*, e usuário final. Todos esses atores realizam **ações** (*use-cases*) e a partir dos **atores** e das **ações** de cada **perfil** pode-se determinar os **recursos** necessários. Cada perfil de programa interativo pode ser analisado e identificado a partir do uso de três parâmetros:

- **necessidade de interação**, que é o fato da interação ser necessária ou não para que o programa interativo seja assistido (pode ser **CN** - **com necessidade** ou **SN** - **sem necessidade**);
- **nível de interação**, que é a medida quantitativa da interação média esperada, e pode ser dividida em 6 níveis : **alto(6)**, **médio-alto(5)**, **médio(4)**, **médio-baixo(3)**, **baixo(2)** e **sem interação(1)**;
- **acesso a conteúdo**, que corresponde à quantidade de conteúdo que pode ser acessado, podendo ser **restrito (R)** ou **amplo (A)**.

No universo desta proposta, o programa interativo é incrementado com características de consciência de contexto e computação ubíqua.

Inicialmente, o autor (PRODUTOR) deve preparar a o conteúdo multimídia, através de objetos de mídia MPEG-4, inserindo-os em uma TIMELINE. Nessa TIMELINE, podem ser inseridos os ELEMENTOS (objetos de uma cena), os ELEMENTOS DE INFORMAÇÃO (“objetos abstratos”) e as PESSOAS que farão parte da programação.

Em termos de apresentação, o conteúdo multimídia a pode ser realizado de duas formas:

1. em tempo-real de programação (CONTEUDO_AO_VIVO);
2. como revisão do conteúdo (CONTEÚDO_REGISTRADO).

Com relação ao CONTEÚDO, na produção de um programa interativo, o PRODUTOR pode inserir referências a OBJETOS DE MÍDIA, DOCUMENTOS, HIPERDOCUMENTOS e SITES WWW, além de poder conectar, através de um LINK externo, a programação de dois ou mais ambientes distintos.

5.9 Integrando os modelos NECTAR, SPICE e PRACTIC

A modelagem das classes dos modelos NECTAR, SPICE e PRACTIC apresenta-se como uma referência para o desenvolvimento de aplicações multimídia que tenham como cenário

um ambiente interativo consciente de contexto. A **Figura 5.21** apresenta um diagrama UML *use-case* que descreve o cenário das macro-interações que devem ocorrer entre as entidades em um ambiente dessa natureza.

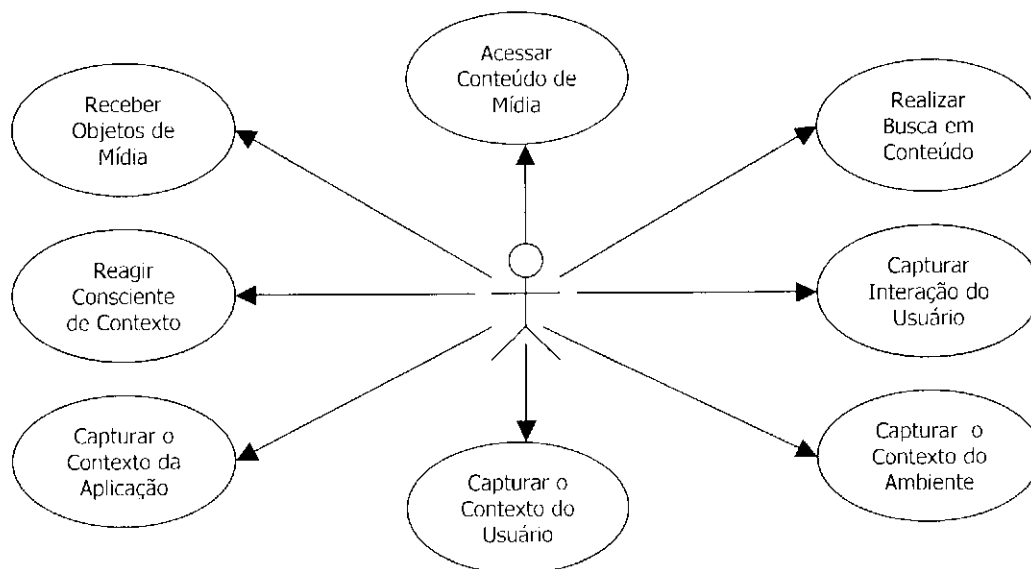


Figura 5.21: Diagrama UML *use-case* para o modelo de acesso aos objetos de mídia

A **Figura 5.22** apresenta um diagrama UML de classes que descreve as relações entre as classes envolvidas em um dos esquemas de interação possíveis no ambiente. Observa-se que a relação entre o objeto de mídia e o restante do ambiente passa pela definição dos esquemas de descrição de conteúdo, impactando diretamente o comportamento dos objetos na cena, o que é compatível com as especificações do *MPEG-4 Systems*.

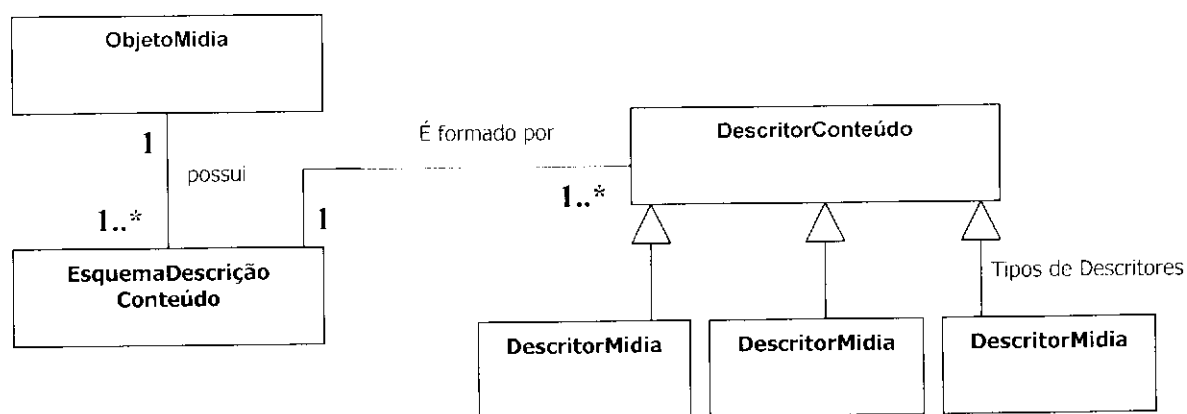


Figura 5.22: Diagrama de classes para o modelo de acesso ao conteúdo de mídia

O diagrama UML de estados apresentado pela **Figura 5.23** ilustra as ações (representadas por verbos) que geram interações entre componentes do ambiente. Essas interações

correspondem às transições entre estados e estão relacionadas aos monitores de eventos associados ao BIFS de uma cena, conforme especificações do MPEG-Systems.

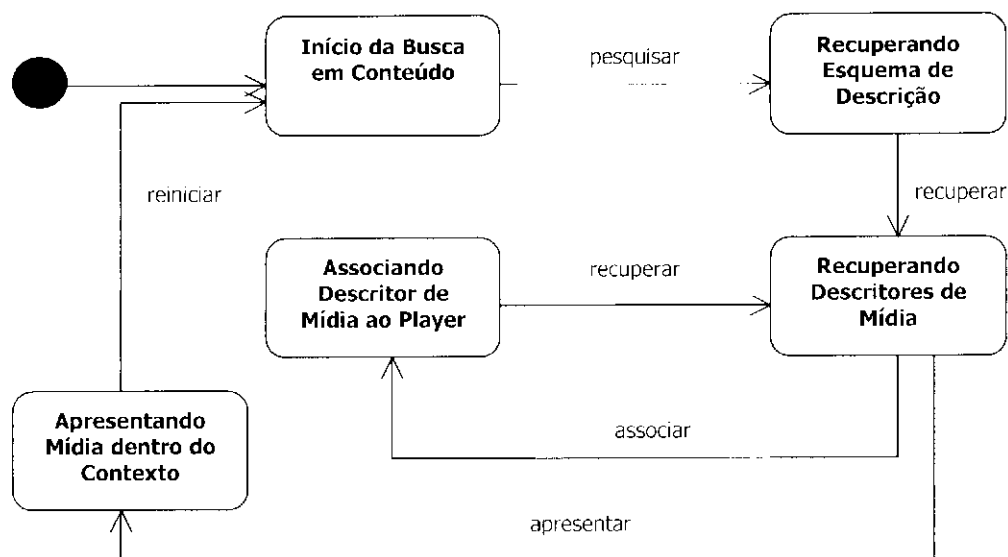


Figura 5.23: Diagrama UML de estados para o acesso ao conteúdo associado à mídia

Complementando a apresentação do cenário, a **Figura 5.24** mostra um diagrama UML de colaboração entre os componentes de hardware e software necessários para gerar as interações apresentadas na **Figura 5.21**.

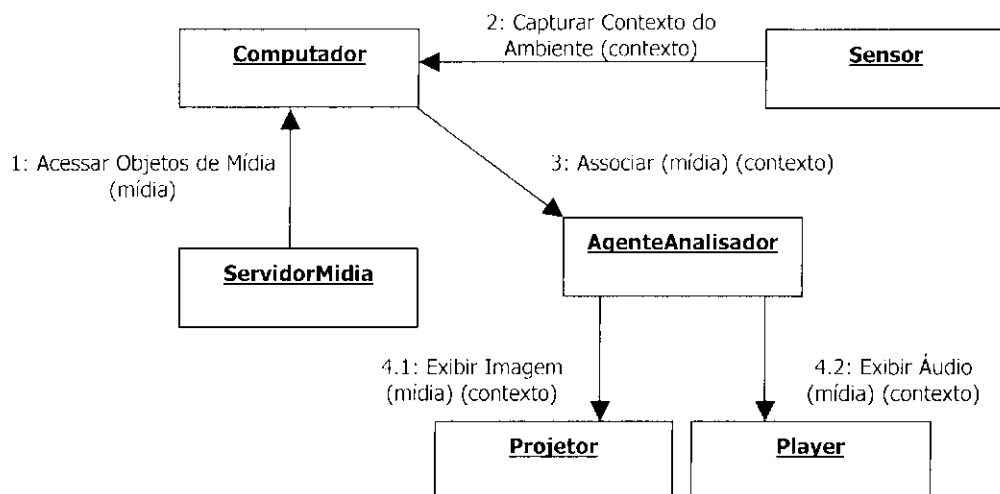


Figura 5.24: Diagrama UML de colaboração entre agentes físicos para o acesso ao conteúdo de mídia

Identificado o cenário de aplicação dos modelos **NECTAR**, **SPICE** e **PRACTIC**, as próximas subseções discutem, em detalhes, todas as classes de objetos, modeladas em UML, necessárias para suportar o desenvolvimento de aplicações multimídia para ambientes interativos conscientes de contexto.

5.9.1 Modelagem das classes UML

Na modelagem proposta por este trabalho, uma aplicação interativa é o elemento do ambiente interativo que integra os três modelos especificados: **NECTAR**, **SPICE** e **PRACTIC**. Adicionalmente, a aplicação interativa encontra-se dividida em três grandes componentes; outros dois componentes são relacionados a todas as partes do processo.

A modelagem, apresentada na **Figura 5.25**, é baseada em UML, permitindo a especificação formal das classes e seus relacionamentos, em termos das interações que são geradas durante o uso da aplicação pelo usuário.

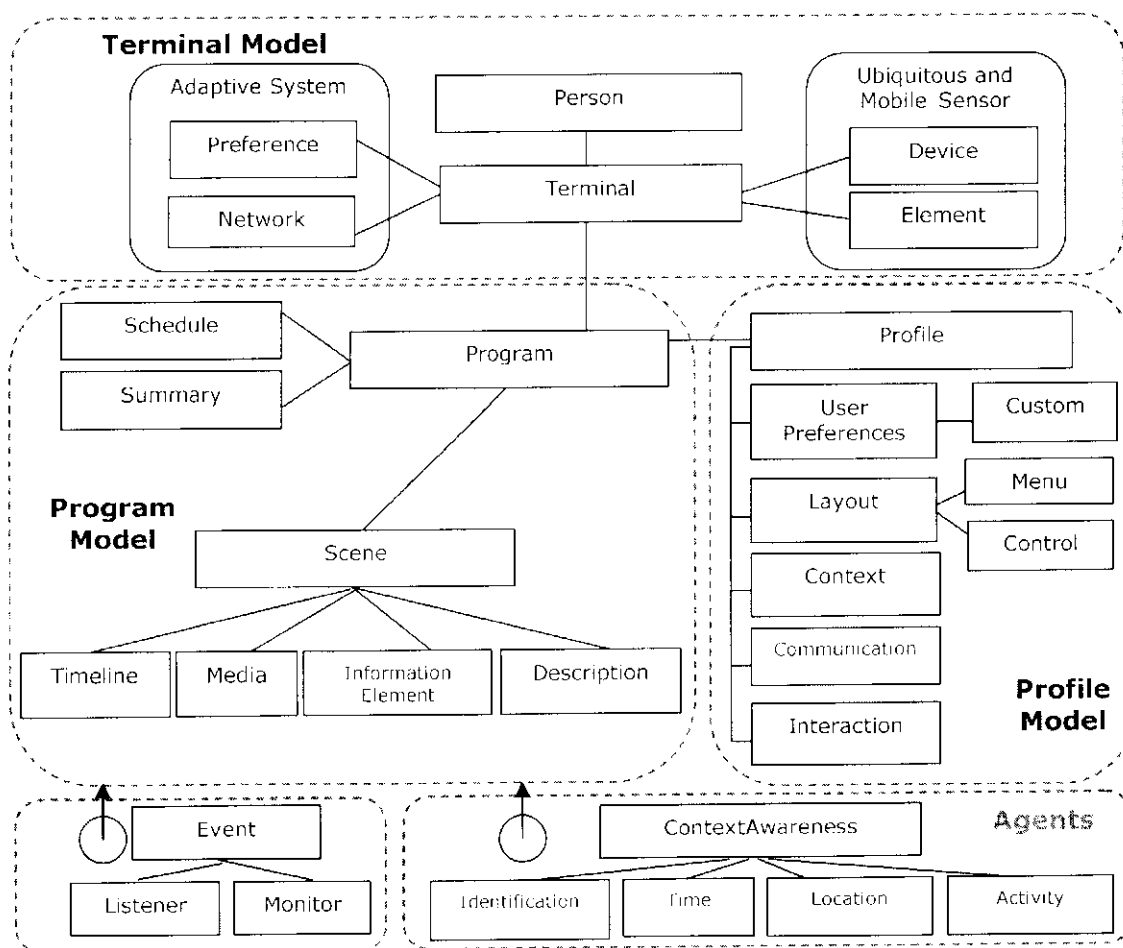


Figura 5.25 – Diagrama de classes UML para um ambiente interativo consciente de contexto (Santos Jr., 2001a)

O modelo de interação, descrito pelo **SPICE**, é composto por três principais componentes. O primeiro componente é o *Program Model*. O objeto central de uma aplicação interativa é o programa interativo (*interactive program*), que é associado às principais partes dos outros modelos (*interactive profile* e *terminal*); essas associações existem porque o terminal recebe informações de um perfil de programa (*program profile*), porém, como toda essa informação

está no *program model*, ele se comunica com o objeto *program* (a **Seção 5.8.1** apresentará o *modelo de programa interativo em detalhes*).

O segundo componente é o *Profile Model*. Cada programa é composto (e apresentado) por um ou mais perfis; um perfil pode ser visto como uma “face” diferente de um programa (a **Seção 5.8.2** apresentará o conceito de *perfil de programa interativo* com mais detalhes).

O terceiro componente é o *Terminal Model*. O objeto central desse componente é o terminal, que está conectado ao Sistema Adaptativo (*Adaptive System*) e aos sensores de movimento e equipamentos ubíquos (*Ubiquitous and Mobile Sensor*), que incluem todos os sensores e equipamentos que podem prover informação contextual útil; o terminal pode possuir equipamentos embutidos, como decodificadores (de vídeo e de áudio) e acessórios para a interação (por exemplo, teclados e *joysticks*); outro objeto que está presente no *Terminal Model* é a pessoa (*person*), que é o objeto que armazena toda a informação relevante sobre todos os usuários do sistema; exemplos de informações relevantes podem ser idade, gênero, salário, canais e programas favoritos, restrições de acesso, volume do som e ajustes da imagem; o *Adaptive System* é referente às informações relacionadas à rede, enquanto o *Ubiquitous/Mobile Sensor* é composto por objetos (*devices*) que representam os equipamentos que tornam possível a associação de consciência de contexto à ubiqüidade, como sensores e suas características.

Os outros componentes são os referentes à consciência de contexto e aos eventos que ocorrem durante o programa, conforme o modelo **SPICE** apresentado no **Capítulo 4**.

Com base no diagrama UML de classes da **Figura 5.25**, as próximas **subseções** descrevem os detalhes de todas as classes especificadas para suportar o desenvolvimento de aplicações interativas baseadas nos modelos **NECTAR**, **SPICE** e **PRACTIC** e que serão inseridas em ambientes interativos conscientes de contexto.

5.9.1.1 A classe **Person**

A classe **Person** contém a especificação de informações das pessoas que irão interagir com um PDR (*Personal Digital Recorder*) em um ambiente interativo consciente de contexto. Um PDR deve ser especificado e visto como elemento central da interação do usuário com um programa interativo. A classe **Person**, portanto, deve possuir interfaces com a classe **Profile** e

com a lista de atividades de uma instância da classe **Program**, conforme pode ser observado no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Person** é formada pelos métodos e propriedades descritos na **Tabela 5.11**.

Tabela 5.11 – Métodos da classe Person

Métodos	Definição
SetIDPerson(Identification);	Método que define uma identificação (ID) única para uma pessoa em um programa
setProfilePerson(Profile);	Método que associa um profile a uma pessoa, considerando o contexto em que a pessoa estiver inserida
Activity[] getValidActionsList();	Método que retorna a lista das ações válidas para uma pessoa, considerando-se o perfil associado à pessoa.
Identification getIDPerson();	Método que retorna um objeto de identificação (ID) para uma pessoa em um programa
setActivityPerson(Person, Activity);	Método que associa uma atividade como sendo válida para uma pessoa em um programa
Location getLocationPerson();	Método que retorna um objeto de localização que representa a posição geográfica de uma pessoa em relação ao ambiente em que um programa está inserido

5.9.1.2 A classe PDR

A classe **PDR** contém as propriedades de um equipamento do ambiente, as propriedades de personalização de um PDR e as descrições sobre a interação desse PDR com outros dispositivos do ambiente, tais como os sensores, por exemplo. Ressalta-se, novamente, que um PDR deve ser especificado e visto como elemento central da interação do usuário com um programa interativo. As interações da classe **PDR** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **PDR** é formada pelos métodos e propriedades descritos pela **Tabela 5.12**.

Tabela 5.12 – Métodos da classe PDR

Métodos	Definição
PDR getDeviceProperties();	Método que retorna as propriedades físicas de um dispositivo do ambiente, especialmente em termos da associação com um PDR
Control[] GetDeviceControls();	Método que retorna a lista dos controles válidos para um dispositivo do ambiente

Identification getIDDevice();	Método que retorna a identificação única de um dispositivo em relação ao contexto em que estiver inserido
SendCommand(Command);	Método que envia comandos para um receptor em um programa
InformationElement getElementProperties();	Método que retorna a lista de propriedades de um elemento associado a um programa
Identification getIDElement();	Método que retorna a identificação única de um elemento associado a um programa
Identification getIDPerson(PDR);	Método que retorna a identificação de uma pessoa que estiver utilizando um determinado PDR
Profile getProfilePerson(PDR);	Método que retorna o profile associado a uma pessoa que estiver utilizando um determinado PDR
Network getNetworkProperties();	Método que retorna um registro de propriedades (<i>bitrate</i> , protocolo de comunicação, dentre outros) de uma rede de comunicação à qual um PDR estiver associado
PDR getPDRPreferences();	Método que retorna as propriedades preferenciais para o funcionamento de um PDR em um ambiente
SendRequestInteraction(Interaction);	Método que envia uma requisição de interação para um PDR, habilitando o PDR a essa interação de acordo com o contexto

5.9.1.3 A classe Element

A classe **Element** contém a especificação de informações e propriedades de elementos de um programa interativo que podem interagir com um PDR. Deve-se observar que agentes podem utilizar elementos para interagir com um PDR. As interações da classe **Element** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Element** é formada pelos métodos e propriedades descritos pela **Tabela 5.13**.

Tabela 5.13 - Métodos da classe Element

Métodos	Definição
PDR setIDElement();	Método que define um ID para um elemento do ambiente
Events[] getValidEventsList();	Método que retorna a lista de eventos válidos para um elemento
setElementProperties(Properties[]);	Método que define propriedades para um elemento
String receiveMessage();	Método que retorna uma mensagem recebida por um elemento durante uma sessão interativa

5.9.1.4 A classe Device

A classe **Device** contém as propriedades de equipamentos que podem ser utilizados/acessados por um PDR. A classe **Device** gera subclasses que podem estar relacionadas com *mobile computing devices*, *sensing*, *ubiquitous computing* e demais meios eletrônicos que permitam algum tipo de interação com o ambiente. As interações da classe **Device** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Device** é formada pelos métodos e propriedades descritos pela **Tabela 5.14**.

Tabela 5.14 – Métodos da classe Device

Métodos	Definição
setDeviceProperties(Properties[]);	Método que define as propriedades de um dispositivo do ambiente
setDeviceControls(Control[]);	Método que define os controles válidos para um dispositivo
setIDDevice(Identification);	Método que define um ID para um dispositivo
Command receiveCommand();	Método que retorna um comando recebido por um dispositivo

5.9.1.5 A classe Preference

A classe **Preference** contém a descrição das propriedades de personalização de uma interação com um PDR. As interações da classe **Preference** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Preference** é formada pelos métodos e propriedades descritos pela **Tabela 5.15**.

Tabela 5.15 – Métodos da classe Preference

Método	Definição
setPDRPreferences(Preference[]);	Método que define as propriedades de preferências para um PDR
setValidInteractions(Interaction[]);	Método que define a lista de interações válidas em relação às preferências de um dispositivo
Interaction[] getValidInteractions();	Método que retorna a lista de interações válidas em relação às preferências de um dispositivo

5.9.1.6 A classe Network

A classe **Network** contém as especificações das características de uma rede de comunicação que estiver associada a um PDR. Adicionalmente, classe **Network** suporta a especificação dos recursos do ambiente associados ao terminal que contém um PDR. As interações da

classe **Network** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Network** é formada pelos métodos e propriedades descritos pela **Tabela 5.16**.

Tabela 5.16 – Métodos da classe Network

Método	Definição
<code>setNetworkRate(short)</code>	Método que define a taxa de transmissão em uma rede
<code>NetworkProperties verifyNetworkStatus();</code>	Método que retorna as propriedades de <i>status</i> de uma rede
<code>setComputerInformation(Device);</code>	Método que define as informações de configuração de um computador como um dispositivo do ambiente
<code>Preferences getNetworkRequirements();</code>	Método que retorna os requisitos de uma rede de comunicação em relação ao contexto
<code>String getNetworkName();</code>	
<code>Identification getNetworkID();</code>	Método que retorna a identificação de uma rede como sendo um elemento do ambiente
<code>String getProtocolName();</code>	
<code>short getNetworkRate();</code>	
<code>Device[] getNetworkAssociatedDeviceList();</code>	Método que retorna a lista de dispositivos associados a uma rede

5.9.1.7 A classe Profile

A classe **Profile** contém métodos e atributos que fornecem informações sobre as características de um perfil de um elemento do ambiente, bem como a lista das interações possíveis e quais os tipos de busca disponíveis. As interações da classe **Profile** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Profile** é formada pelos métodos e propriedades descritos pela **Tabela 5.17**.

Tabela 5.17 – Métodos da classe Profile

Atributos	
<code>String name;</code>	Nome do perfil
<code>String program;</code>	Nome do programa que originou um perfil
<code>boolean searchwww;</code>	Informa se é permitida a busca de informações no ambiente da WWW
<code>boolean searchscene;</code>	Informa se é permitida a busca de informações em uma cena
<code>boolean intrascenetree;</code>	Informa se é possível a interação com o grafo de objetos da cena

<code>boolean interscenetree;</code>	Informa se há conexão entre grafos de cenas diferentes, permitindo a interação entre cenas
Métodos	Definição
<code>setProfileName(String);</code>	Método que define um nome para um profile
<code>setProgramName(String);</code>	Método que define um nome para um programa que originou um profile
<code>Identification getIdPerson();</code>	Método que retorna a identificação de um usuário que pode ser associado a um profile
<code>setSearchInWWW(boolean);</code>	Método que define o estado (TRUE ou FALSE) para a busca no ambiente da WWW
<code>setSearchInScene(boolean);</code>	Método que define o estado (TRUE ou FALSE) para a busca dentro de uma cena
<code>setSearchInterScene(Scene[], boolean);</code>	Método que define o estado (TRUE ou FALSE) para a busca entre cenas
<code>SetInteractionWithSceneTree(Scene, boolean);</code>	Método que define a possibilidade de interação com o grafo de objetos de uma cena
<code>Interaction[] getInteractionList(Scene);</code>	Método que retorna a lista de interações possíveis em uma cena em relação a um profile

5.9.1.8 A classe `UserPreference`

A classe `UserPreference` armazena as preferências do usuário para um determinada profile de programa interativo, incluindo as mudanças em relação ao menu, controles de interação, layout e o profile como um todo. As interações da classe `UserPreference` podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe `UserPreference` é formada pelos métodos e propriedades descritos pela **Tabela 5.18**.

Tabela 5.18 - Métodos da classe `UserPreference`

Atributos	
<code>String filemenu;</code>	Nome do arquivo de configuração que contém as modificações do usuário em relação aos menus
<code>String filecontrol;</code>	Nome do arquivo de configuração que contém as modificações do usuário em relação aos controles de interação

<code>String filelayout;</code>	Nome do arquivo de configuração que contém as modificações do usuário em relação ao layout das interfaces gráficas
<code>String fileprofile</code>	Nome do arquivo de configuração que contém as modificações do usuário em relação ao profile como um todo
Métodos	Definição
<code>Media[] getAvailableMedia(Profile);</code>	Método que retorna uma lista das mídias disponíveis para uso em um determinado perfil de programa interativo
<code>setCustomFiles(Profile, String, String, String, String);</code>	Método que define os arquivos de personalização contendo as preferências do usuário

5.9.1.9 A classe Layout

A classe **Layout** contém as propriedades de um layout gráfico para um profile de um programa interativo. As interações da classe **Layout** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Layout** é formada pelos métodos e propriedades descritos pela **Tabela 5.19**.

Tabela 5.19 – Métodos da classe Layout

Atributos	
<code>Vector[] mediaposition;</code>	Um vetor com quatro campos: o primeiro deve conter o ID de um objeto de mídia; o segundo deve conter a posição da mídia no eixo cartesiano X; o terceiro deve conter a posição da mídia no eixo cartesiano Y; o quarto deve conter a posição da mídia no eixo cartesiano Z, quando o layout for no plano 3D.
<code>long int bgcolor;</code>	Valor da cor de fundo do layout de um programa
Métodos	Definição
<code>setMediaPosition(Identification, Vector);</code>	Método que define a posição de uma mídia em uma interface gráfica
<code>setBackgroundcolor(long int);</code>	Método que define a cor de fundo do layout de um programa

5.9.1.10 A classe Menu

A classe **Menu** contém as propriedades de um menu gráfico para um profile de um programa interativo. As interações da classe **Menu** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Menu** é formada pelos métodos e propriedades descritos pela **Tabela 5.20**.

Tabela 5.20 - Métodos da classe Menu

Atributos	
String fontname;	O nome da fonte padrão utilizada nos controles do menu
long int fontcolor;	Valor da cor da fonte dos controles do menu
short fontsize;	Valor do tamanho da fonte padrão utilizada nos controles do menu
Vector elements;	Um vetor contendo todos os elementos que compõem o menu
Métodos	
setFontName (String);	Definição
setFontColor(long int);	
setFontName (String);	
setFontSize (short);	
addElement (Element);	Método que adiciona um elemento a um menu
setElementIndex(int);	Método que define a posição do elemento no vetor de elementos do menu
Vector getElementsOfMenu ();	Método que retorna um vetor de elementos que compõem um menu

5.9.1.11 A classe Control

A classe **Control** contém as propriedades de todos os controles associados a um profile de um programa interativo. As interações da classe **Control** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Control** é formada pelos métodos e propriedades descritos pela **Tabela 5.21**.

Tabela 5.21 - Métodos da classe Control

Atributos	
Layout buttonlayout;	Atributo que contém o layout de um botão de controle
short linkpress;	Atributo que contém o efeito gráfico de acionamento de um controle; pode ser 1(efeito pressionamento de botão); 2 (efeito

	sublinhado); 3 (efeito definido pelo usuário)
Vector navigationbar;	Vetor de atributos de uma barra de navegação
Métodos	Definição
Control[] getMediaControls(Media);	Método que retorna uma lista dos controles permitidos para um tipo de mídia
setButtonLayout(Layout);	Método que define o layout de um botão de controle
setLinkPresentation(short);	Método que define a forma de apresentação de um controle
setNavigationBar(Vector);	Método que define as propriedades de uma barra de navegação

5.9.1.12 A classe Communication

A classe **Communication** contém as características necessárias tanto ao nível da rede quanto ao nível dos recursos de hardware locais para a apresentação de cada mídia e do profile como um todo. As interações da classe **Communication** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Communication** é formada pelos métodos e propriedades descritos pela **Tabela 5.22**.

Tabela 5.22 - Métodos da classe Communication

Atributos	
Vector profileband;	Um vetor de propriedades da banda passante necessária para um determinado profile
Vector profilerequirements;	Um vetor de requisitos de um recurso formado por dois campos: o primeiro deve conter o ID de um recurso; o segundo, uma lista das características desse recurso
Método	Definição
setProfileBand(Vector);	Método que define as propriedades da banda passante para um determinado profile
setProfileRequirements(Vector);	Método que define os requisitos computacionais necessários para uso de um recurso em um profile (por exemplo, codificadores, decodificadores, processadores)

5.9.1.13 A classe Context

A classe **Context** relaciona as características contextuais necessárias para um profile de programa interativo. Vale ressaltar que as classes de contexto (**Identification**, **Time**, **Activity**

e **Location**) usufurem das informações capturadas pela classe **Context**. Desta forma, a classe **Context** funciona como uma coletora de informações contextuais, construindo listas (vetores) com essas informações coletadas do ambiente. As interações da classe **Context** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Context** é formada pelos métodos e propriedades descritos pela **Tabela 5.23**.

Tabela 5.23 – Métodos da classe Context

Atributos	
Vector ndevice;	Um vetor dos dispositivos pertinentes a um determinado profile
Vector nmedia;	Um vetor das mídias pertinentes a um determinado profile
Vector nperson;	Um vetor das pessoas pertinentes a um determinado profile
Método	Definição
setNecessaryDevice(Device);	
setNecessaryPerson(Person);	
setNecessaryMedia(Media);	
Vector getNecessaryDevice();	
Vector getNecessaryPerson();	
Vector getNecessaryMedia();	

5.9.1.14 A classe Interaction

A classe **Interaction** contém as propriedades que definem uma interação entre todos os elementos do ambiente. Cada interação deve ser identificada e relacionar duas ou mais entidades do ambiente interativo consciente de contexto. Observa-se, então, que as interações estão diretamente relacionadas com as classes contextuais. As interações da classe **Interaction** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Interaction** é formada pelos métodos e propriedades descritos pela **Tabela 5.24**.

Tabela 5.24 – Métodos da classe Interaction

Método	Definição
setInteractionID(Identification);	Método que define um ID para uma interação gerada no ambiente
DefineInteraction(Interaction, Element[]);	Define uma interação como válida para um ou mais elementos do ambiente
associateInteractionToContext(Interaction, Context);	Método que associa uma interação a um contexto
GetInteractionList [] (Context);	Método que retorna a lista de interações associadas a todos os elementos em um determinado contexto

5.9.1.15 A classe Identification

A classe **Identification** é responsável pelas operações de identificação de pessoas e elementos que fazem parte de um programa interativo inserido em um ambiente consciente de contexto. Neste sentido, a classe **Identification** provê mecanismos que identificam sumariamente esses elementos, permitindo a execução de ações associadas a uma lista de **atividades** e considerando as condições de **localização** e **tempo**. As interações da classe **Identification** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Identification** é formada pelos métodos e propriedades descritos pela **Tabela 5.25**.

Tabela 5.25 - Métodos da classe Identification

Método	Definição
<code>Identification getIdPerson(Person);</code>	Método que retorna a identificação única de uma pessoa em um programa, a partir da interação com um PDR
<code>Identification getIdElement(Element);</code>	Método que retorna a identificação única de um elemento em uma cena, permitindo a associação com descrições de cena e modelos de eventos
<code>Identification getIdInformationElement(InformationElement);</code>	Método que retorna a identificação única de um elemento de informação em uma cena, permitindo a associação com modelos de eventos.
<code>Identification getIdDevice();</code>	Método que retorna a identificação única de um device que permite INPUT/OUTPUT em um programa.
<code>associateIDLocation(Identification, Location);</code>	Método que permite a associação de dados de identificação a um mapa de localização de pessoas, elementos e devices.
<code>associateIDActivity(Identification, Activity);</code>	Método que permite a associação de uma identificação a uma atividade ou lista de atividades.
<code>Identification[] getListIDTime(Time, Timeline);</code>	Método que retorna uma lista de identificadores de elementos que realizam alguma atividade em um instante de tempo <i>T</i> em uma <i>timeline</i>

5.9.1.16 A classe Location

A classe **Location** é responsável pelas operações de localização de pessoas e elementos que fazem parte de um programa interativo inserido em um ambiente consciente de contexto. A classe **Identification** manipula, conceitualmente, um mapa de localização, que contém os elementos diretamente associados ao programa e também os elementos do ambiente como um todo, como é o caso da localização de **devices** que permitem a interação com o programa através de um PDR. As interações da classe **Location** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Location** é formada pelos métodos e propriedades descritos pela **Tabela 5.26**.

Tabela 5.26 - Métodos da classe Location

Método	Definição
Location GetPositionPersonScene(Person);	Método que permite obter a posição corrente de uma pessoa em uma cena de um programa.
Location GetPositionElementScene(Element);	Método que permite obter a posição corrente de um elemento em uma cena de um programa.
Location GetPositionDevice(Device);	Método que permite obter a posição corrente de um device associado a um programa através de um PDR.
EnablePersonLocationMonitor(Person, boolean);	Método que habilita/desabilita a associação de um monitor de presença para uma pessoa em um programa.
EnableElementLocationMonitor(Monitor, boolean);	Método que habilita/desabilita a associação de um monitor de presença para um elemento em um programa.
EnableDeviceLocationMonitor(Device, boolean);	Método que habilita/desabilita a associação de um monitor de presença para um device em um programa.
Location getLocationDataSensor(Sensor);	Método que permite obter a posição corrente de um sensor de dados no ambiente

5.9.1.17 A classe Activity

A classe **Activity** é responsável por operações que permitem o acesso às atividades de um programa. Entende-se por atividade como sendo um item de programação em execução, a partir do dispara de eventos ou ações pré-programadas. A classe **Activity** relaciona-se diretamente com as classes **Identification**, **Location** e **Time**, resultando em operações conjugadas, quanto da definição dos níveis de consciência de contexto associados a uma

atividade ou lista de atividades. As interações da classe **Activity** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Activity** é formada pelos métodos e propriedades descritos pela **Tabela 5.27**.

Tabela 5.27 – Métodos da classe Activity

Método	Definição
Identification getActivityID();	Método que retorna a identificação sumária de uma atividade em uma cena de um programa.
Activity[] getActivitiesListPerson(Person);	Método que retorna a lista de atividades associadas a uma pessoa em uma cena de um programa.
Activity[] getActivitiesListScene();	Método que retorna o <i>schedule</i> de todas as atividades programadas para uma cena em um programa.
Activity getCurrentActivityPerson();	Método que retorna a identificação e as propriedades da atividade correntemente associada a uma pessoa.
Vector getTimeDateActivity();	Método que retorna os dados temporais (tempo e data) associados a uma atividade, especialmente em termos de cronometragem.
updateActivityPerson(Person, Activity);	Método que permite atualizar a associação de uma atividade a uma pessoa, considerando-se aspectos de consciência de contexto como a localização, a identificação e o tempo.
updateEventListForActivity(Event [], Activity);	Método que permite atualizar a lista de eventos válidos para uma atividade, considerando-se aspectos de consciência de contexto como a localização, a identificação e o tempo.
Person[] selectPersonsForActivity();	Método que retorna a lista de pessoas associadas a uma atividade.
associateActivityElement(Element, Activity);	Método que associa uma atividade a uma pessoa ou elemento do ambiente

5.9.1.18 A classe Time

A classe **Time** é responsável por operações que permitem o acesso aos dados temporais de uma cena em um programa. Entende-se por dados temporais como sendo os valores definidos na pré-programação de uma *timeline*, e também os valores de tempo associados à execução de atividades, localização de pessoas, elementos e devices. Observa-se que a classe **Time** tem relacionamento direto com a pré-programação baseada na *timeline* e também com

as preferências associadas a um PDR. As interações da classe **Time** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Time** é formada pelos métodos e propriedades descritos pela **Tabela 5.28**.

Tabela 5.28 – Métodos da classe Time

Método	Definição
getTime();	Método que retorna a hora corrente, considerando-se um padrão de referência previamente definido no programa, como por exemplo o GMT.
getTemporalData();	Método que retorna os dados temporais associados a qualquer pessoa (como por exemplo, o tempo total de interação entre uma pessoa e um device), elemento, agente ou device em um programa.
setTime();	
setTemporalData();	Método que retorna os dados associados a pré-programação de datas para ocorrência de eventos e/ou execução de atividades em um programa.
getDate();	
setDate();	
getTimeDateListPerson();	Método que retorna a lista de todos os dados temporais (time e date) associados a uma pessoa.
getTimeDateListElement();	Método que retorna a lista de todos os dados temporais (time e date) associados a um elemento
getTimeDateListDevice();	Método que retorna a lista de todos os dados temporais (time e date) associados a um device.

5.9.1.19 A classe Listener

A classe **Listener** é uma interface que pode ser associada a qualquer atividade programada para uma cena, permitindo a notificação de ocorrência de eventos. As interações da classe **Listener** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Listener** é formada pelos métodos e propriedades descritos pela **Tabela 5.29**.

Tabela 5.29 – Métodos da classe Listener

Método	Definição
getIDEvent();	Método que retorna a identificação única de um evento associado a uma atividade em uma cena.
add();	Método que associa (adiciona) um

	<i>listener</i> (notificador de eventos) a uma atividade em uma cena.
<code>remove()</code> ;	Método que remove um <i>listener</i> (notificador de eventos) de uma atividade em uma cena.
<code>equals()</code> ;	Método que verifica se um <i>listener</i> é igual a outro <i>listener</i> presente em uma cena.

5.9.1.20 A classe Monitor

A classe **Monitor** é uma interface que pode monitorar qualquer ator (pessoas, elementos, agentes e devices) em um programa interativo inserido em um ambiente consciente de contexto e notificar dados e propriedades desses atores, seja em estado de inércia ou quando em estado de execução de alguma atividade. As interações da classe **Monitor** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Monitor** é formada pelos métodos e propriedades descritos pela **Tabela 5.30**.

Tabela 5.30 – Métodos da classe Monitor

Método	Definição
<code>getActorAssociatedToMonitor()</code> ;	Método que retorna a identificação do ator ao qual um monitor está associado.
<code>add()</code> ;	Método que associa (adiciona) um <i>monitor</i> a um ator em um programa.
<code>remove()</code> ;	Método que remove um <i>monitor</i> de um programa.
<code>getListMonitorFunctions()</code> ;	Método que retorna a lista de funções de um monitor.

5.9.1.21 A classe Program

A classe **Program** contém as propriedades mais genéricas de um programa interativo, identificando as características que catalogam um programa. As interações da classe **Program** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Program** é formada pelos métodos e propriedades descritos pela **Tabela 5.31**.

Tabela 5.31 – Métodos da classe Program

Método	Definição
<code>Time getDuration()</code> ;	Método que retorna dados de tempo de duração de um programa
<code>Time getEndTime()</code> ;	Método que retorna dados de tempo quando do encerramento de um programa
<code>String getProgramName()</code> ;	Método que retorna o nome de um programa
<code>Scene[] getScenesTDLList()</code> ;	Método que retorna uma lista de

	identificadores das cenas de um programa
<code>Time getStartTime();</code>	Método que retorna dados de tempo quando do início de um programa
<code>Person[] getPersonsList();</code>	Método que retorna uma lista de pessoas que interagem com um programa
<code>Element[] getElementSIDList();</code>	Método que retorna uma lista de identificadores dos elementos associados a um programa
<code>Vector getMetaInformationAuthors();</code>	Método que retorna as propriedades, na forma de lista de meta-informação, dos autores de um programa

5.9.1.22 A classe Sensor

A classe **Sensor** contém as propriedades de um sensor, que é identificado como um elemento do ambiente. Um sensor fornece dados do ambiente e auxilia no controle desse ambiente através dos aspectos de consciência de contexto. As interações da classe **Sensor** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Sensor** é formada pelos métodos e propriedades descritos pela **Tabela 5.32**.

Tabela 5.32 - Métodos da classe Sensor

Método	Definição
<code>short getSensorType();</code>	Método que retorna o tipo de sensor; 1, para sensores contextuais; 2, para sensores embutidos em dispositivos do ambiente
<code>Identification getSensorID();</code>	Método que retorna o ID de um sensor
<code>Context[] getSensorRange();</code>	Método que retorna o contexto no qual a atuação de um sensor é válida
<code>enableSensor();</code>	Método que habilita o uso de um sensor no ambiente
<code>disableSensor();</code>	Método que desabilita o uso de um sensor no ambiente
<code>associateSensorMonitor(Monitor);</code>	Método que associa um sensor a um monitor de eventos

5.9.1.23 A classe Schedule

A classe **Schedule** fornece métodos para acesso às propriedades de controle de uma programação, que contém um programa interativo. As interações da classe **Schedule** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Schedule** é formada pelos métodos e propriedades descritos pela **Tabela 5.33**.

Tabela 5.33 – Métodos da classe Schedule

Método	Definição
<code>addActivity(Activity);</code>	Método que adiciona uma atividade ao schedule de um programa
<code>Control[] getDeviceControls();</code>	Método que retorna a lista dos controles presentes em um programa
<code>Identification[] getIDDevice();</code>	Método que retorna a lista de identificadores de dispositivos presentes em um programa
<code>addListenerForActivity(Listener);</code>	Método que adiciona um <i>listener</i> para uma atividade em um programa
<code>Event[] getProgramEvents();</code>	Método que retorna uma lista dos eventos associados a um programa
<code>Activity[] getProgramActivities();</code>	Método que retorna uma lista de atividades associadas a um programa

5.9.1.24 A classe Element

A classe **Element** contém as propriedades singulares de um elemento do ambiente, sendo que o elemento é definido como sendo qualquer entidade do ambiente. As interações da classe **Element** podem ser vistas no diagrama UML de classes apresentado pela Figura 5.25. Em termos de definição, a classe **Element** é formada pelos métodos e propriedades descritos pela Tabela 5.34.

Tabela 5.34 - Métodos da classe Element

Método	Definição
<code>Identification getIDElement();</code>	Método que retorna o ID de um elemento
<code>setIDElement(Identification);</code>	Método que define um ID para um elemento
<code>Location getLocationElement();</code>	Método que retorna a localização de um elemento no ambiente
<code>Event[] getValidEventsList();</code>	Método que retorna a lista de eventos associados a um elemento do ambiente
<code>Properties getProperties();</code>	Método que retorna as propriedades de um elemento
<code>boolean equalsTo(Element);</code>	Método que verifica se um elemento é igual a outro elemento

5.9.1.25 A classe InformationElement

A classe **InformationElement** contém informações de controle de todos os elementos de um ambiente. Vale ressaltar que um elemento de informação (**InformationElement**) difere-se de um **element** (**Element**) em termos conceituais, uma vez que um elemento refere-se diretamente a uma entidade do ambiente. As interações da classe **InformationElement** podem ser vistas no diagrama UML de classes apresentado pela Figura 5.25. Em termos de

definição, a classe **InformationElement** é formada pelos métodos e propriedades descritos pela **Tabela 5.35**.

Tabela 5.35 – Métodos da classe InformationElement

Método	Definição
<code>getIDInformationElement();</code>	Método que retorna o identificador de um elemento de informação
<code>setIDInformationElement(Identification);</code>	Método que define um ID para um elemento de informação
<code>informationElement getInformationType();</code>	Método que retorna as informações de ambiente associadas a um elemento de informação
<code>Properties getProperties();</code>	Método que retorna as propriedades de um elemento de informação
<code>EqualsTo(InformationElement);</code>	Método que verifica se um elemento de informação é igual a outro
<code>associateInformationElementToPerson(Person);</code>	Método que associa um elemento de informação a uma pessoa
<code>associateInformationElementToElement(Element);</code>	Método que associa um elemento de informação a um elemento do ambiente
<code>associateInformationElementToDevice(Device);</code>	Método que associa um elemento de informação a um dispositivo do ambiente
<code>associateInformationElementToMedia(Media);</code>	Método que associa um elemento de informação a uma mídia em uma cena

5.9.1.26 A classe Media

A classe **Media** permite o acesso às macro-informações de uma mídia (objeto de mídia) em uma cena multimídia. Essas macro-informações não se referem às características técnicas de uma mídia, mas descrevem quais são as relações de uma mídia em uma cena e quais interações são permitidas. As interações da classe **Media** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Media** é formada pelos métodos e propriedades descritos pela **Tabela 5.36**.

Tabela 5.36 – Métodos da classe Media

Método	Definição
<code>defineMediaType(short);</code>	Método que define um tipo (Object Descriptor) para uma mídia
<code>monitoringAddMediaScene();</code>	Método que monitora a inserção de uma mídia a uma

	cena
Media getMediaTargetInteraction(Interaction);	Método que retorna qual mídia está sendo alvo de uma interação
monitoringRemoveMediaScene();	Método que monitora a remoção de uma mídia de uma cena
Media[] selectMediaListOfScene();	Método que retorna uma lista de todas as mídias inseridas em uma cena
addEventListForMedia(Event[]);	Método que adiciona uma lista de eventos a uma mídia
addInformationForMedia(InformationElement);	Método que adiciona um elemento de informação a uma mídia
Control[] getControls();	Método que retorna uma lista de todos os controles permitidos sobre uma mídia
InformationElement[] captureInformationOfMedia();	Método que captura todas as informações de uma mídia como elemento de um ambiente
Properties getProperties();	Método que retorna as propriedades físicas de uma mídia

5.9.1.27 A classe Event

A classe **Event** permite o acesso às informações sobre os eventos que ocorrem no ambiente interativo consciente de contexto. Desta forma, os eventos não são simplesmente disparadores de ações geradas a partir da interação do usuário, mas ações que dependem das variáveis de contexto. As interações da classe **Event** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Event** é formada pelos métodos e propriedades descritos pela **Tabela 5.37**.

Tabela 5.37 – Métodos da classe Event

Método	Definição
Identification getIdEvent();	Método que retorna um identificador de um evento
setEventToActivity(Event[], Activity[]);	Método que define um evento ou lista de eventos permitidos para uma atividade ou lista de atividades
SetEventToMedia(Event[], Media[]);	Método que define um evento ou lista de eventos permitidos para uma mídia ou lista de mídias
boolean equalsTo(Event);	Método que verifica se um evento é igual a outro evento
Event getLastEvent();	Método que retorna o último

	evento que ocorreu no ambiente como um todo
Event [] <code>getEventTime(Time);</code>	Método que retorna o evento (ou lista de eventos) associado a um elemento de tempo no ambiente
Event [] <code>getEventList(Element);</code>	Método que retorna a lista de todos os eventos ocorridos em um elemento do ambiente

5.9.1.28 A classe Timeline

A classe **Timeline** contém as propriedades de um equipamento do ambiente, as propriedades de personalização de um PDR e as descrições sobre a interação desse PDR com outros dispositivos do ambiente, tais como os sensores, por exemplo. Ressalta-se, novamente, que um PDR deve ser especificado e visto como elemento central da interação do usuário com um programa interativo. As interações da classe **Timeline** podem ser vistas no diagrama UML de classes apresentado pela **Figura 5.25**. Em termos de definição, a classe **Timeline** é formada pelos métodos e propriedades descritos pela **Tabela 5.38**.

Tabela 5.38 – Métodos da classe Timeline

Método	Definição
Time <code>getDuration();</code>	Método que retorna o tempo total de duração de uma timeline para uma cena
Time <code>setDuration();</code>	Método que define o tempo total de duração de uma timeline para uma cena
Element [] <code>getParallelElementList(Time);</code>	Método que retorna a lista de todos os elementos que se relacionam no ambiente em um determinado instante de tempo <i>T</i> de uma timeline
Element [] <code>getSequentialElementList();</code>	Método que retorna a lista de todos os elementos relacionados em uma timeline
<code>monitoringAddElement();</code>	Método que monitora a inserção de um elemento a uma timeline
<code>monitoringRemoveComponent();</code>	Método que monitora a remoção de um elemento de uma timeline
<code>addMonitorToTime(Time);</code>	Método que adiciona um monitor de eventos para uma unidade de tempo em uma timeline

5.10 Integrando agentes de software para prover consciência de contexto

Os agentes de software podem ser utilizados para prover consciência de contexto ao ambiente interativo, considerando-se as definições de agentes reportadas na literatura (Moreira & Walczowski, 1997; Ketchpel & Genesereth, 1994). Os relacionamentos entre

agentes caracterizam as interações que ocorrem entre Usuário-Rede-Aplicação, conforme ilustrado na **Figura 5.26**.

A **Figura 5.26** apresenta um esquema para representar as interações entre entidades em uma aplicação inserida em um ambiente interativo consciente de contexto. No esquema, é possível observar a existência de interações bem definidas entre Usuário-Rede, Aplicação-Rede, e Usuário-Aplicação.

As interações entre a rede e o usuário ocorrem quando as ações do usuário disparam eventos sobre componentes de software nativos da infra-estrutura de comunicação. As interações entre a aplicação e a rede cobrem o relacionamento entre todas as aplicações (agentes, programas, *daemons*, *drivers*, dentre outras) e a rede de comunicação (especialmente ao nível dos protocolos de comunicação). As interações entre o usuário e a aplicação ocorrem sempre que o usuário manipular, através da aplicação, um elemento do terminal que gera um evento no ambiente como um todo. Complementando, denomina-se **interação completa** quando houver relacionamentos entre as três entidades de interação.

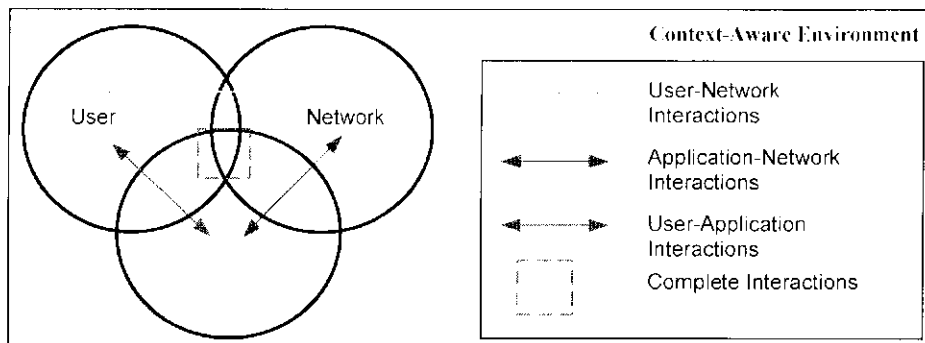


Figura 5.26: Esquema de interação completas em um ambiente consciente de contexto

Essas interações completas podem ser vistas no modelo de integração de agentes apresentado na **Figura 5.27**.

A seqüência de transições apresentada na **Figura 5.27** inicia-se com o usuário ligando o terminal e escolhendo um programa (1); em seguida, o terminal fará, simultaneamente, a captura de informação sobre o ambiente, começando por informações sobre os dispositivos ligados e suas propriedades (2.1), seguindo com a captura das informações dadas pelos sensores (2.2) e as informações contextuais, como dia, horário e eventos relacionados (2.3). Depois disso, com as informações capturadas, o terminal recebe o perfil do programa

adequado para aquele contexto (3). Então, as informações são passadas para a aplicação de TV Interativa, que configurará o programa de TVI para ser recebido (4). Em seguida, as descrições do programa são recebidas (5), seguidas pelo estado atual da rede (6); então, serão definidas as melhores características para a transmissão do programa na rede (7) e o estado do servidor é recebido. Finalizando, a aplicação envia o programa para o terminal apresentar. Existem ainda os agentes de eventos, que têm a função de capturar eventos durante a transmissão e que podem mudar algumas características do programa (10).

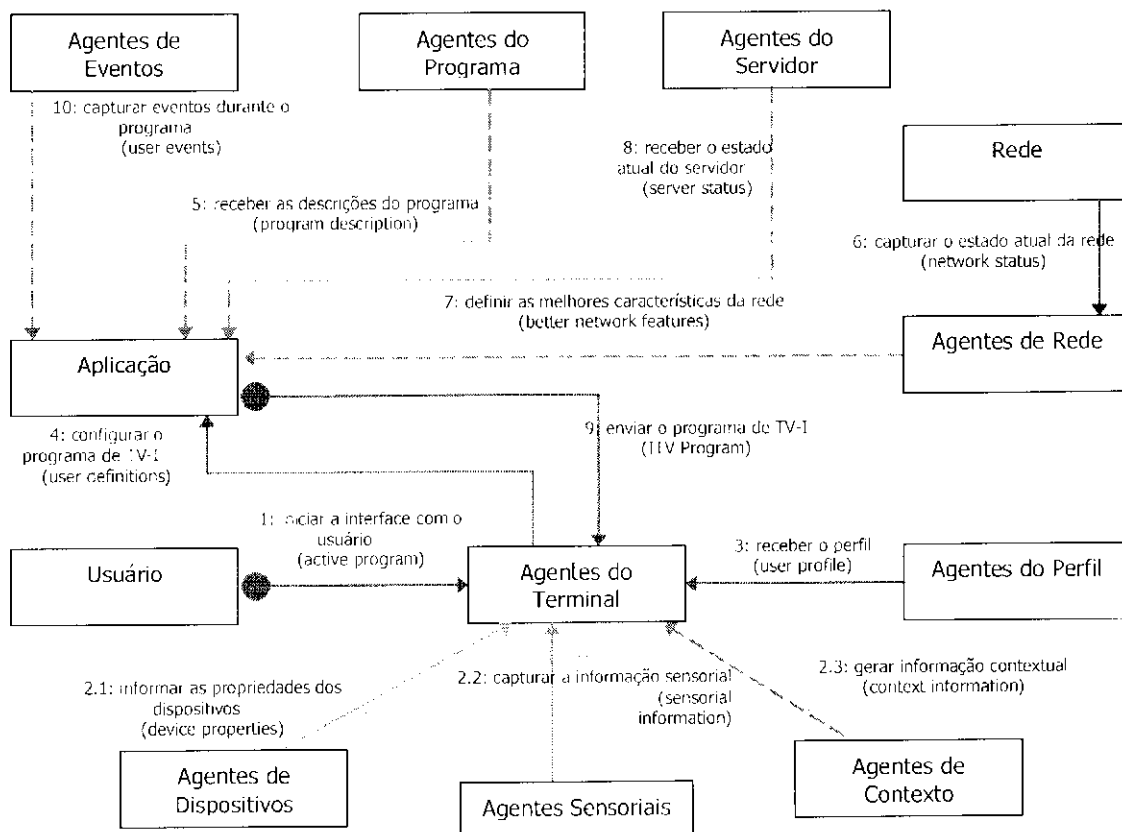


Figura 5.27 – Diagrama de colaboração UMI entre agentes em um ambiente consciente de contexto

Ao nível do desenvolvimento de aplicações para ambientes interativos conscientes de contexto, conforme discutido no **Capítulo 4**, as relações contextuais surgem das interações que ocorrem entre as entidades do ambiente e descrevem, de modo geral, os eventos e as ações que ocorrem no ambiente.

Considerando que agentes de software podem ser utilizados para prover informações contextuais em todo o ambiente, essas relações contextuais podem ser mapeadas em um esquema de descrição de eventos baseado em agentes, conforme ilustra a **Figura 5.27**.

O uso de um agente de software em um ambiente interativo consciente de contexto pode ser mapeado através de um diagrama UML da, como mostrado na **Figura 5.27**. No diagrama, os relacionamentos entre agentes caracterizam as interações que ocorrem entre as entidades Usuário-Aplicação-Rede, evidenciando que os agentes podem ser distribuídos por todo o ambiente e coletar informações contextuais, especialmente aquelas que permitam adaptação automática do conteúdo multimídia em relação ao contexto e interesses (mapeados de acordo com o perfil e preferências) do usuário.

A capacidade social dos agentes computacionais é usada para o estabelecimento da cooperação na solução de problemas complexos, como é o caso das interações completas em ambientes interativos conscientes de contexto.

O comportamento de um agente de software deve ser descrito com base em uma ontologia, conforme prevê o padrão a ACL (*Agent Communication Language*) do padrão FIPA (*Foundation for Intelligent Physical Agents*) (Bellifemine et al., 2001).

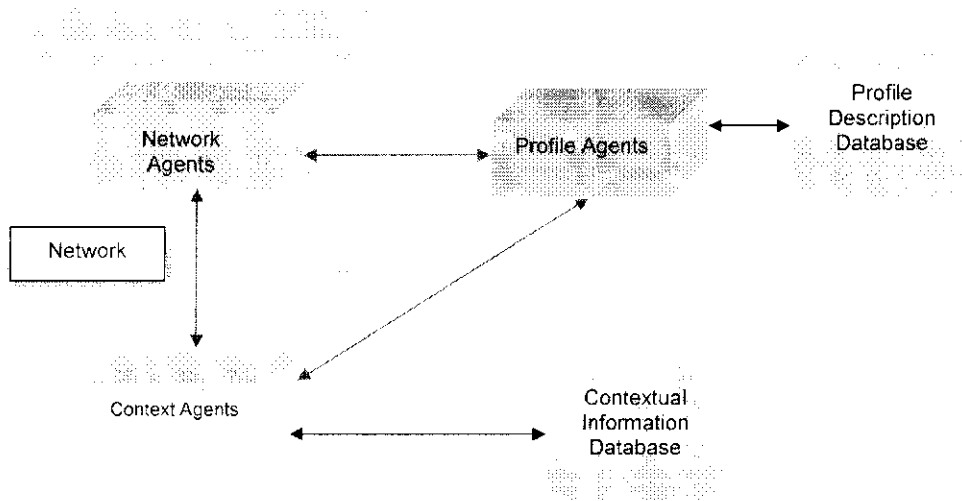


Figura 5.28: Comportamento de um subconjunto de agentes

A **Figura 5.28** mostra um subconjunto dos agentes que compõem o diagrama completo apresentado na **Figura 5.27**, enquanto a **Figura 5.29** apresenta um esquema para definição de uma ontologia em um ambiente interativo consciente de contexto (Santos Jr. et al., 2001b).

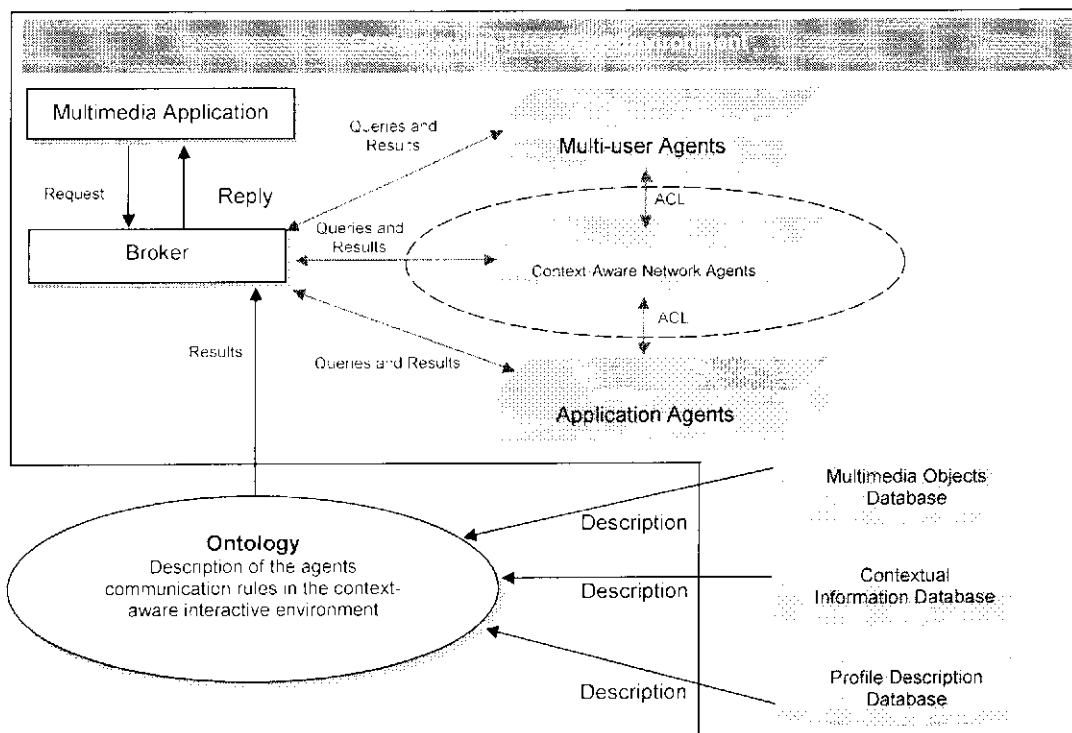


Figura 5.29: Esquema baseado em agentes para coleta de informações contextuais (Santos Jr. et al., 2001b)

Conforme ilustra a **Figura 5.29**, a ontologia descreve como organizar as ações dos agentes. A ontologia utiliza esquemas de descrição baseados em XML (ou MPEG-7), que informam como extrair informação das bases de dados; os resultados produzidos pela cooperação entre os agentes podem ser utilizados como parâmetro de entrada para a negociação entre um *broker* (*middleware*) e a aplicação multimídia.

O diagrama UML de colaboração, apresentado pela **Figura 5.30**, mostra o comportamento dos agentes da rede, agentes do contexto e agentes do perfil em relação à ontologia definida. Os agentes da rede são ativados com um pedido do usuário de um programa interativo; os agentes da rede estabelecem comunicação com os agentes do contexto e com os agentes do perfil; com agentes de contexto, os agentes da rede negociam sobre as características contextuais que impactam a qualidade de serviço da aplicação; com agentes do perfil, os agentes da rede negociam sobre as características do perfil de usuário que são relevantes para a adaptabilidade dos aspectos de QoS na rede; os agentes de contexto negociam com os agentes de perfil sobre as características do perfil de usuário que são relevantes para o contexto em que a aplicação está inserida.

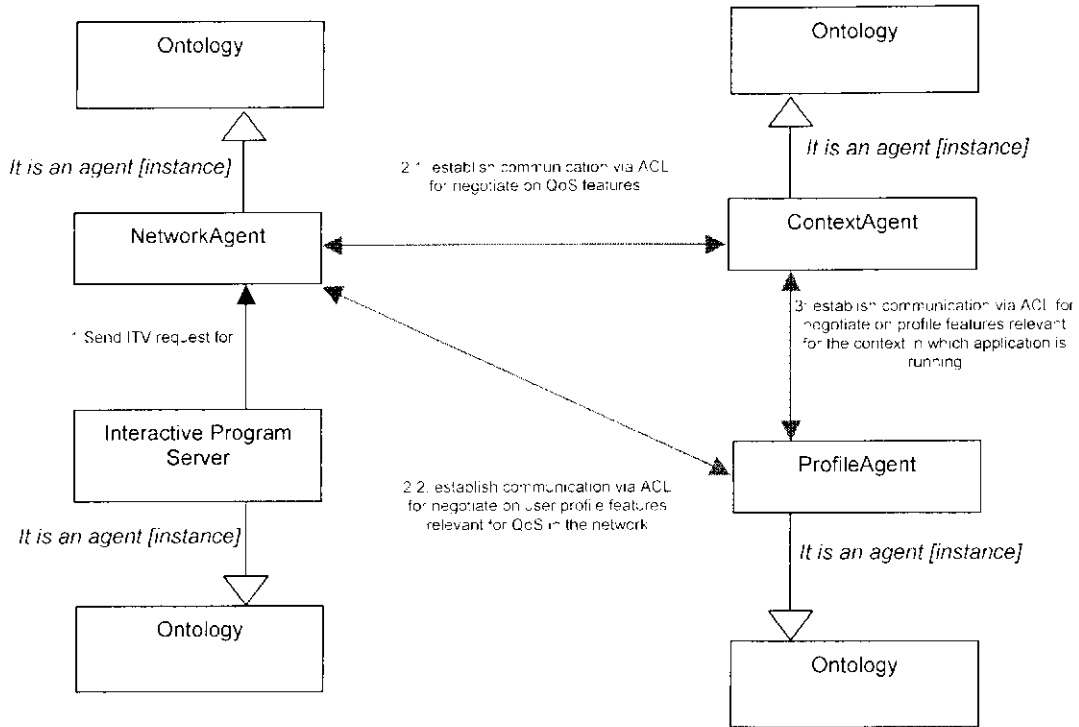


Figura 5.30: Um diagrama UML de colaboração para um subconjunto de agentes (Figura 5.25) (Santos Jr. et al., 2001a)

Para o escopo desta tese, diversas plataformas de desenvolvimento de agentes oferecem suporte à implementação. A plataforma JADE, também desenvolvida pelo grupo MPEG, oferece todos os recursos necessários para definição de ontologias, criação de agentes (móveis ou não) e para o estabelecimento da comunicação entre agentes. Adicionalmente, o fato da plataforma JADE ser inteiramente desenvolvida em JAVA permite a sua integração com aplicações *MPEGlet* desenvolvidas com a API MPEG-J. A Tabela 5.39 apresenta uma visão geral dos pacotes das classes da plataforma JADE.

Tabela 5.39 – Pacotes da plataforma JADE

Pacote	Descrição
jade.content.acl	Contém classes para estabelecimento de comunicação entre agentes
jade.content.lang	Contém classes que especificam a linguagem de comunicação entre os agentes
jade.content.onto	Contém classes para definição de uma ontologia
jade.content.schema	Contém classes para definição de esquemas de descrição do conteúdo a ser manipulado por um agente
jade.core	Contém o <i>microkernel</i> da plataforma JADE
jade.core.behaviours	Contém classes para definir o

	comportamento de agentes
<code>jade.core.event</code>	Contém classes para manipular eventos gerados por agentes
<code>jade.domain</code>	Contém classes para especificação do escopo de uma ontologia
<code>jade.domain.FIPAAgentManagement</code>	Contém a definição da ontologia FIPA-Agent-Management de acordo com o padrão FIPA, apresentado como deve ser o monitoramento dos agentes na plataforma JADE
<code>jade.domain.mobility</code>	Contém classes que permitem definir a ontologia JADE-mobility, utilizada no desenvolvimento de agentes móveis
<code>jade.gui</code>	Contém classes para o desenvolvimento de interfaces gráficas baseadas no modelo swing da linguagem Java
<code>jade.proto</code>	Contém as regras dos protocolos que regulam o comportamento dos agentes, de acordo com o padrão FIPA
<code>jade.util</code>	Contém classes que permitem a manipulação das propriedades dos agentes
<code>jade.wrapper</code>	Oferece suporte ao desenvolvimento de aplicações Java externas ao <i>mundo dos agentes</i> , como é o caso das aplicações MPEG-J

5.11 Discussão de relações entre esta proposta e o MPEG-21

Na seção 2.7, o MPEG-21 foi apresentado como sendo o novo projeto de padrão do grupo MPEG, tendo sido iniciado no final do ano de 2001. As principais características do MPEG-21 foram apresentadas e relacionadas à definição dos **itens digitais** universais (Koenen, 2001).

Uma vez apresentada a modelagem proposta por este trabalho, é importante discutir algumas relações que existem entre esta proposta e aquilo que está sendo especificado nos cenários do MPEG-21.

A essência de todos os esforços do grupo MPEG é a interoperabilidade, especialmente para o consumidor de multimídia. A interoperabilidade, neste caso, significa que os consumidores podem usufruir de conteúdo multimídia sem qualquer preocupação com formatos de arquivo, *codecs*, modelos de meta-informação, arquiteturas de hardware e software, dentre outros elementos envolvidos na apresentação multimídia.

De acordo com a proposta técnica, o objetivo do MPEG-21.s é descrever um *framework* que referencie como diferentes elementos de hardware e software, no contexto do conteúdo multimídia, podem ser utilizados na construção de uma infra-estrutura para a distribuição e o consumo desse conteúdo multimídia. Neste cenário, o grupo do MPEG-21 absorve esforços de representantes da indústria fonográfica (música), da indústria de produção de mídias e dos fornecedores de tecnologia multimídia. O aspecto **integração** é o elemento-chave da proposta do MPEG-21 e, segundo estimativas do grupo MPEG, há 6 (seis) bilhões de potenciais usuários para a proposta MPEG-21 (Koenen, 2001).

O projeto do MPEG-21 procura usar padrões existentes, quando possível, e apresentar mecanismos para facilitar sua integração em uma infra-estrutura. Essa integração é a base para a primeira parte do MPEG-21: a **declaração de item digital**, conforme visto na **Seção 2.7.5**. Essa declaração é baseada em um XML-*Schema*, considerando que XML é um poderoso padrão para intercâmbio de dados. No entanto, a parte mais complexa do MPEG-21 é a **identificação** e a **descrição** dos itens digitais. Para este caso, o foco do MPEG-21 é desenvolver mecanismos que permitam a adaptação do conteúdo ao ambiente. Vale observar que para o grupo MPEG o conceito de **ambiente** é o mesmo desenvolvido nesta tese. Neste caso, o conteúdo multimídia será visto como uma coleção de **itens digitais** associada a um ambiente, sendo que o ambiente inclui as redes de comunicação, os terminais de exibição e as aplicações para acesso ao conteúdo.

Neste contexto, o grupo MPEG-21 tem sinalizado a necessidade de pesquisas e propostas que incluam: a) a representação de conteúdo multimídia como coleção de **itens digitais**, observando-se as representações já existentes em outros padrões MPEG, como é o caso dos objetos de mídia do padrão MPEG-4; b) os modelos de interação que descrevam o comportamento desses **itens digitais** no ambiente; c) os modelos que conceituem e caracterizem os novos ambientes interativos; d) os modelos que controlem eventos e monitores para personalização nesses novos ambientes interativos.

Cronologicamente, esta tese teve início em 1998, atingindo um estágio de maturidade no ano de 2001. Observa-se pelos argumentos descritos, que a proposta, aqui reportada, tem relações com as expectativas de projeto do MPEG-21.

O conceito **interativo**, ao se referir a todas as interações que ocorrem entre os mais diversos elementos de uma infra-estrutura, pode ser aplicado aos requisitos do MPEG-21. Adicionalmente, o conceito de **ambiente**, ao se referir à composição da infra-estrutura de rede, terminais, aplicações e usuários, também se enquadra na proposta do MPEG-21.

Ao nível do conteúdo multimídia, a proposta de personalização de conteúdo, através de *profiles*, é diretamente aplicável ao MPEG-21, sendo que os aspectos de consciência de contexto, formalizados nesta tese, representam uma nova contribuição para o projeto do novo padrão. De forma complementar, as classes UML propostas nesta tese formalizam os requisitos das *interações completas* (rede-aplicação-usuário), e esse requisitos podem ser mapeados e/ou configurados para auxiliar a **descrição de itens digitais**, que é um dos pontos de investigação recomendados para futuras propostas de contribuições ao MPEG-21.

5.12 Considerações finais

As tecnologias MPEG-4 e MPEG-7, desenvolvidas para serem padrões internacionais, são potenciais para o desenvolvimento de aplicações interativas nesse novo e complexo universo de sistemas digitais integrados.

Enquanto tecnologias, as características desses padrões evidenciam sua aplicabilidade na codificação, distribuição e apresentação de multimídia. Por representarem mídias como objetos e permitirem a descrição de conteúdo multimídia em uma cena, essas tecnologias se relacionam diretamente com o paradigma da orientação a objetos. Neste sentido, a proposta de modelagem de ambientes interativos conscientes de contexto apresenta-se como uma forma de orientação para a cadeia produtiva dos novos programas interativos, como é o caso dos programas de TVI.

Neste capítulo, foram apresentados e discutidos os modelos que compõem a modelagem de ambientes interativos conscientes de contexto. Como contribuição, foram apresentadas as formas de integração dos modelos propostos com as tecnologias MPEG-4 e MPEG-7, identificadas como base tecnológica para sustentar esta proposta.

No próximo capítulo, discute-se a modelagem de uma aplicação de TVI como forma de validação da modelagem proposta neste capítulo.

6.1 Considerações iniciais

A televisão é um meio de comunicação extremamente popular e que possui linguagem e funcionalidades próprias. Atualmente, a televisão passa por um período de revolução; a primeira grande mudança está sendo a digitalização do sinal e a segunda é o novo universo de possibilidades de interação que se pretende criar.

Essas duas mudanças tendem a impactar diretamente o modo como as pessoas assistem a um programa de TV. A Televisão Interativa (TVI) é uma aplicação multimídia que insere novas funcionalidades em um ambiente anteriormente não interativo (Backer et al., 1998). Um grande número de tecnologias pode ser usado em sua implementação e, devido ao crescente interesse da indústria, diversos esforços têm sido feitos para sua padronização (Anatel, 2001).

Neste capítulo, é apresentado e discutido um novo conceito para criar e distribuir programas interativos, representando um avanço na adaptação dos programas às necessidades ou expectativas dos usuários. Na próxima seção, será apresentada uma breve descrição sobre TVI. Na sequência, serão revisitados os principais aspectos de consciência de contexto relevantes a aplicações de TVI e discutidos os aspectos da infra-estrutura necessária para a distribuição, os requisitos para a produção, a modelagem do ambiente de TVI consciente de contexto, e a modelagem e implementação de uma aplicação de TVI, como é o caso de um evento esportivo.

6.2 Um breve relato da evolução da interação na televisão

A possibilidade da interação com um programa de televisão é um desejo antigo. Diversos programas incentivam a interação dos espectadores, na maioria das vezes por telefone, e essa interação ocorre tanto em *game shows* quanto para fazer pesquisa de opinião, como é o caso da escolha do desfecho de um programa. Um aspecto inconveniente é que essa interação é bastante limitada, uma vez que a TV e o telefone são aparelhos que não se comunicam de forma integrada. Com a evolução da tecnologia e o aparecimento das URDs (*Unidade*

Receptora-Decodificadora), a ligação do aparelho de TV com o telefone tornou-se possível, possibilitando o lançamento de diversos modelos e produtos de TV Interativa.

A transmissão da TV Interativa pode ser feita de três modos: a) via *broadcast*, em que todos os usuários recebem toda a informação possível, porém as URDs só apresentam o necessário; b) via *unicast*, que permite a entrega individual da informação para o usuário (tem a vantagem de que só a informação que interessa ao usuário é disponibilizada, porém tem a desvantagem no caso da transmissão de vídeo de vários *threads*, necessitando muitos recursos computacionais dos servidores); c) via *multicast*, em que a transmissão de cada objeto exige um *thread* do servidor, independente do número de receptores (nesse modo, cada receptor entra num grupo para receber o objeto). Martins mostra uma comparação de desempenho entre um servidor de vídeo *unicast* e um *multicast*, na qual fica evidente a melhor utilização da banda e o menor processamento necessário no servidor *multicast* (Martins, 2001).

O canal de retorno é um outro fator que categoriza a TV Interativa em três grupos. O primeiro grupo é o que não possui canal de retorno, sendo que toda a informação com a qual o usuário pode interagir sempre é entregue ao usuário. Esse tipo de interação apresenta alto custo em termos de transmissão e apresenta limitações para o usuário. O segundo grupo possui canal de retorno não-dedicado; a linha telefônica é utilizada como o canal de retorno. A interação fica mais “rica” e menos limitada. A maioria dos modelos de *Enhanced TV* pode ser inserida nesse grupo. Como parte do último grupo estão as TVs Interativas com canal de retorno dedicado. Deste grupo fazem parte as TVs Interativas por Cabo ou que utilizam linhas DSL para o retorno. A existência de um canal dedicado permite uma interação completa, uma vez que a possibilidade de interação está sempre presente e o tempo de resposta, normalmente, é menor.

Atualmente, existem diversas soluções de TV Interativa disponíveis. Essas soluções, apesar de permitirem alguma interação do usuário, são bastante limitadas em relação às expectativas dos usuários, acostumados à interação existente na WWW, por exemplo (Abert, 2000). Existem nomenclaturas diferentes para diferentes implementações de TV Interativa em relação à sua funcionalidade, a saber:

- *Enhanced TV*: os programas podem possuir conteúdo relacionado que, normalmente, é acessado através de um clique em um ícone;

- *Individualized TV*: o usuário possui opções de câmera e áudio, semelhante às funcionalidades de um DVD, por exemplo;
- *Internet TV*: permite o acesso à Internet via televisor, utilizando a linha telefônica como meio de comunicação;
- *On demand TV*: é a TV que permite o *Video on Demand* (ou a sua variação *Near Video on Demand*);
- *Personal TV*: é o tipo de TV que permite ao usuário o controle do vídeo, sendo bastante conhecida e funciona a partir dos PDRs (*Personal Data Recorders*).

Uma das aplicações mais conhecidas e úteis da TV Interativa é o Guia Eletrônico de Programação (EPG – *Electronic Program Guide*), que é uma aplicação presente tanto nos modelos de *Enhanced TV* quanto em *Interactive TV*. Tanta importância deve-se ao fato de que o EPG é o programa centralizador de toda a informação sobre a programação dos canais de TV. Os EPGs são aplicações que diferenciam as funcionalidades entre modelos de TV Interativa, podendo ser simples indicadores da programação atual (como os utilizados nas TVs por satélite), como controladores da gravação de programas (como os usados nos PVRs), ou para fazer recomendações de programação (Barbieri, 2001; Kurapati et al., 2001).

Como foi apresentado, existem diversas implementações e definições de TV Interativa. Neste trabalho, considera-se como TV Interativa a TV que possui as seguintes características:

- recepção digital dos sinais de áudio e de vídeo;
- canal de retorno dedicado;
- possibilidade de transmissão e recepção de dados;
- possibilidade de comunicação com outros equipamentos eletrônicos;
- possibilidade de interação na rede e na aplicação através do uso de agentes de software.

A partir dessas características e dos modelos apresentados e discutidos no **Capítulo 5**, uma aplicação de TV Interativa foi modelada e implementada, sendo que os detalhes dessa modelagem e implementação serão discutidos nas próximas seções.

6.3 Revisitando aspectos de consciência de contexto aplicáveis à TV Interativa

Conforme discutido no **Capítulo 4**, aspectos de consciência de contexto podem ser importantes quando associados ao ambiente no qual uma aplicação está inserida, oferecendo

as informações do contexto em que o usuário se encontra, as possibilidades de adaptação e informações sobre recursos disponíveis em um determinado momento.

Ainda em relação ao **Capítulo 4**, observou-se que palavras como HOW, WHERE, WHAT, WHO e WHEN expressam semânticas que podem ser tão complexas quanto seja necessário a uma aplicação. Conforme exposto, uma semântica HOW, por exemplo, ao se referir a “quais recursos estão em uso”, expressa implicitamente “como o usuário está utilizando esses recursos”. A implementação de um mecanismo avaliador (agente + filtros, por exemplo) que apresente e/ou monitore uma lista dos recursos em uso pode indicar como o usuário está se sentindo no ambiente.

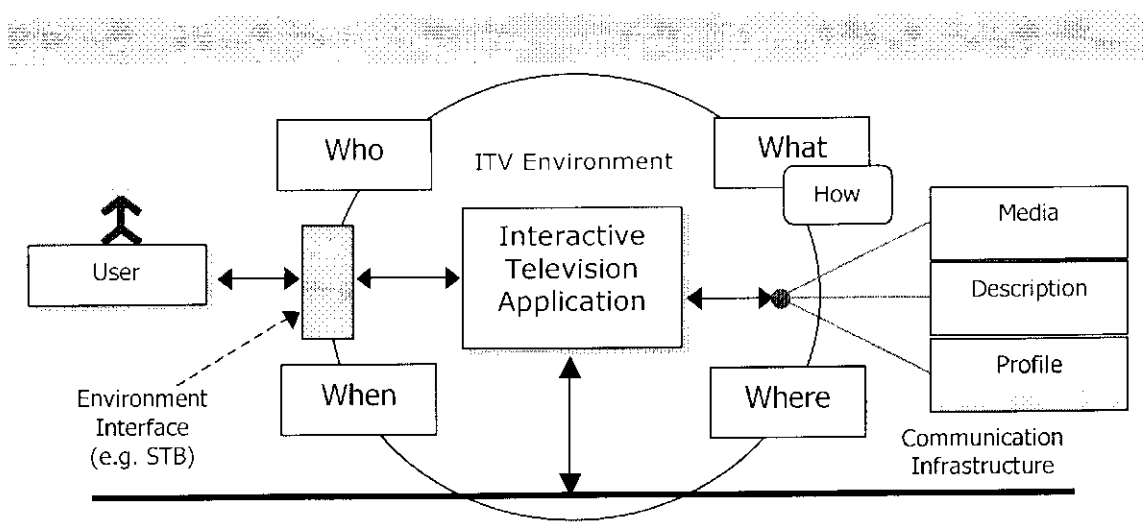


Figura 6.01 - Mapeamento de entidades de consciência de contexto em elementos de perfis de aplicações de TVI

De forma complementar, a proposta de junção das semânticas HOW e WHAT oferece suporte à formação da entidade “atividade” que, em suma, é um elemento formador do conceito de perfil, conforme ilustrado na **Figura 6.01**.

No contexto proposto por este trabalho para as aplicações de TVI, a execução automática de tarefas e ações é de extrema importância. Nesse sentido, agentes podem ser usados para permitir a implementação dos aspectos de consciência de contexto, considerando-se, por exemplo, as quatro entidades de consciência definidas por Dey e Abowd (Dey & Abowd, 2000).

Conforme discutido, o desenvolvimento de perfis para aplicações de TVI é, então, uma forma de categorização que caracteriza a situação de uma entidade em uma aplicação, bem

como uma forma de se obter informações sobre o comportamento dessa entidade no tempo. Assim, com o uso de perfis em conjunto com informações de contexto, uma aplicação pode realizar ações e reações, como as seguintes:

- a apresentação de informações e serviços para o usuário, como, por exemplo, a apresentação de mídias disponíveis e que estão relacionadas com o contexto do usuário;
- a execução automática de serviços, como, por exemplo, o disparo de eventos relacionados com a localização corrente de uma entidade;
- a associação do contexto com a informação para posterior recuperação, como, por exemplo, o armazenamento de informações contextuais durante uma sessão de multimídia interativa, relacionando observações do(s) usuário(s) com o(s) assunto(s)-chave que faz(em) parte da descrição de uma ou mais cenas desse vídeo.

Outro ponto importante reside no fato de que um objeto, em uma interação, pode afetar os níveis de consciência de outro objeto, exigindo, portanto, procedimentos de negociação. Esse aspecto é relevante no acesso a conteúdo de mídias em uma determinada cena, representada com tecnologias como MPEG-4, por exemplo.

6.4 Infra-estrutura para distribuição de programas de TV Interativa

A aplicação de TVI modelada neste trabalho tem como base uma infra-estrutura para a distribuição desenvolvida especialmente para a TVI. Essa arquitetura segue o modelo cliente-servidor e contempla requisitos de consciência de contexto. A **Figura 6.02** apresenta essa infra-estrutura, que encontra-se dividida em:

- *Server*: o servidor tem como função armazenar e distribuir programas de TVI; esses programas são compostos por objetos multimídia, como vídeo, áudio, imagens, animações, gráficos e textos, dentre outros; além disso, um programa deve possuir um *script* indicando como esse programa deve ser apresentado, sua estrutura e as relações espaciais e temporais entre seus objetos (conteúdo e estrutura);
- *Network*: existem duas redes no ambiente apresentado; a rede que faz a ligação entre o servidor e a rede “caseira” (*home network*), e a própria rede “caseira”. A primeira rede deve ser capaz de mudar seu comportamento de modo a transmitir os dados eficientemente, mudando dinamicamente seus protocolos e algoritmos; já a rede “caseira” deve ser capaz de realizar a comunicação entre os diferentes equipamentos (e sensores) e o terminal de exibição;

- *Client*: o cliente é composto pelo terminal e seus agentes; o terminal é composto por um computador pessoal conectado à rede e deve exibir os programas de TVI; esse terminal também deve executar um *player* compatível com o padrão MPEG-4.

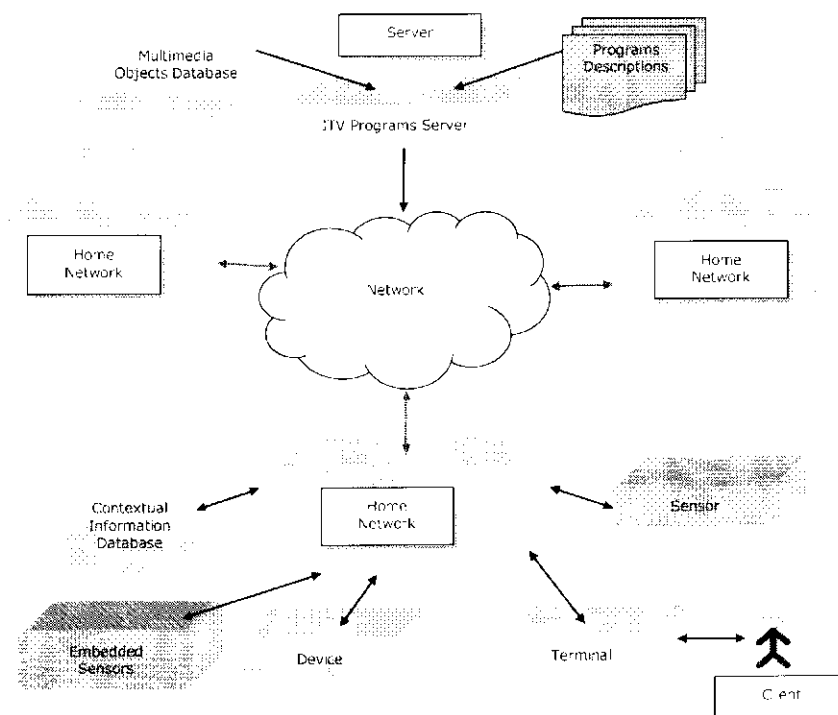


Figura 6.02 - Ambiente consciente de contexto para a TV Interativa

6.5 Requisitos necessários para a produção de programas de TV Interativa

Uma preocupação adicional deste trabalho é minimizar as mudanças na infra-estrutura, necessárias para a produção de um programa de TVI em relação a um programa para a TV Digital. Com isso, além dos equipamentos presentes em um estúdio de TV Digital, como câmeras e VTRs digitais, é necessário acrescentar somente dois novos *softwares*.

O primeiro é uma ferramenta para a edição dos *streams* MPEG-4; essa ferramenta deve possuir funcionalidades para manipular vários objetos de mídia. O segundo *software* necessário é um editor de metadados que suporte o padrão MPEG-7 e os elementos especificados pelo *TV-Anytime Forum* e por este trabalho (que serão discutidos a partir da **Seção 6.7**). Esse segundo *software*, preferencialmente, deve estar integrado a um editor ou *player* de *streams* MPEG-4, de modo a facilitar a inserção de metadados.

A mudança mais significativa está no modo de criar os programas, já que na TVI, além da produção do conteúdo audiovisual tradicional, é necessária a produção de conteúdo audiovisual alternativo, dos metadados e de como será a interação do usuário.

6.6 Modelagem da aplicação de TV Interativa: estrutura e implementação

A abordagem atual da TV Interativa, em nível mundial, prevê somente a adição de alguns aspectos de interatividade aos tradicionais programas de televisão (ATVEF, 2000; DVB, 2000; ARIB, 2000; TVAF, 2001a). No entanto, com a mesma tecnologia que esses modelos utilizam, é possível obter um resultado com maior grau de adaptabilidade ao consumidor e que também permita uma melhor definição do público-alvo pela emissora de televisão.

A aplicação de TVI, apresentada nas próximas seções, é especificada a partir da modelagem proposta no **Capítulo 5**, e considera a integração dos modelos **NECTAR**, **SPICE** e **PRACTIC**, conforme discutido na **Seção 5.9**.

Conforme visto na **Seção 5.8.1**, o programa interativo é o elemento central de uma aplicação de TVI, sendo estruturado de acordo com as especificações da DDL MPEG-7 (ISO, 2001c) e as descrições multimídia definidas pelo MPEG-7 (ISO, 2000a). As entidades *Schedule* e *Summary* são definidas pelo *TV-Anytime Forum* (TVAF, 2001b; TVAF, 2001d) e, por isso, não serão apresentadas aqui com maior detalhamento. A **Tabela 6.01** apresenta os elementos de um programa.

Tabela 6.01 - Elementos do Programa

Nome	Definição
Program	É o principal elemento de um programa, sendo do tipo ProgramType
ProgramType	É um complexType , dentro do qual estão a cena, a programação (Schedule) e a sinopse (Summary)
Information	Apresenta informações sobre o programa
ProgramInformationType	É um simpleType que contém informações sobre o programa
name	Nome do programa, sendo do tipo mpeg7:TitleType
id	Identificação única do programa - é utilizado o identificador de programas do TV Anytime Forum CRIDType
Scene	Descreve os objetos presentes na cena, sendo do tipo SceneType
Summary	Apresenta a sinopse do programa - foi definido pelo TV Anytime Forum como o tipo SynopsisType
Schedule	É um elemento que apresenta a programação do programa, sendo do tipo tva:ScheduleEventType

A **Figura 6.03** apresenta um exemplo de *Schema* MPEG-7 para descrição do programa.


```

<element name="Program" type="practio:ProgramType"/>

<complexType name="ProgramType">
  <sequence>
    <element name="Information"
      type="practio:ProgramInformationType"/>
    <element name="Scene" type="practio:SceneType"/>
    <element name="Summary" type="tva:SynopsisType"/>
    <element name="Schedule" type="tva:ScheduleEventType"/>
  </sequence>
</complexType>

<simpleType name="ProgramInformationType">
  <attribute name="name" type="mpeg7:TitleType"/>
  <attribute name="id" type="tva:CRIDType"/>
</simpleType>

```

Figura 6.03 - Exemplo de *Schema* para o elemento *Program*

O programa deve ser definido e disponibilizado pela emissora, de modo que seja possível sua apresentação de diferentes maneiras. No programa, estão presentes as informações fundamentais para o funcionamento do mesmo, bem como a sua composição por mídias, que podem ser descritas em MPEG-7. Um exemplo do uso do cabeçalho é apresentado na **Figura 6.04**.

```

<Program>
  <Information name="Jogo_Sao_Paulo_x_Palmeiras"
    id="crid://www.practio.tv/shows/Jogo1-2001-04-05"/>
  <Scene>
    ...
  </Scene>
  <Summary>
    ...
  </Summary>
  <Schedule>
    ...
  </Schedule>
</Program>

```

Figura 6.04- Exemplo de cabeçalho de um programa

Como foi discutido na **Seção 5.8.1**, um programa é composto por diversos subelementos, cuja definição, bem como de seus atributos, é apresentada na **Tabela 6.02**.

Tabela 6.02 - Subelementos do programa

Nome	Definição
SceneType	É o tipo de dado que possui as informações sobre a cena
Media	É um elemento que indica o tipo de mídia existente na cena e sua localização, sendo do tipo MediaType

Timeline	É um elemento que indica a localização da descrição da linha do tempo da cena, sendo do tipo TimelineType
Telement	É o elemento responsável pela interação via InformationElement , e é nesse elemento que estão as ações que devem ser feitas de acordo com o contexto; esse elemento, bem como seus subelementos, é do tipo IelementType
Description	É um elemento destinado à descrição de mídias independentes ou da cena; é um tipo de dado MPEG-7, que pode ser um XML ou um HTML, e seu tipo é indicado na descrição da cena; esse elemento deve ser retirado de IPCTMimeTypeCS (ISO, 2001g [seção 17.1.2.1.2])

A **Figura 6.05** apresenta um exemplo de *Schema* MPEG-7 para descrição de cena.

```
<complexType name="SceneType">
  <sequence>
    <element name="Media" type="practic:MediaType"/>
    <element name="Timeline" type="practic:TimelineType"/>
    <element name="Telement" type="practic:IelementType"/>
    <element name="Description" type="mpeg7:ControlledFormUseType"/>
  </sequence>
</complexType>
```

Figura 6.05 - Exemplo do *Schema* para descrever a cena

O programa pode ser composto por vários objetos de mídia, que devem ser descritos. A descrição desses objetos de mídia é apresentada na **Tabela 6.03**, e o seu *Schema* de representação é apresentado na **Figura 6.06**.

Tabela 6.03 - Subelementos de mídia

Nome	Definição
MediaType	É o tipo de dado que descreve as características da mídia
MediaID	É o identificador do objeto de mídia, sendo do tipo mpeg7:UniqueIDType ; é utilizado para a ligação a dados relacionados à mídia descritos em MPEG-7
ObjectID	É o identificador do objeto de mídia, sendo do tipo PracticIDType ; é utilizado para a manipulação do objeto mídia no layout
FileFormat	É um elemento que indica o formato da mídia, sendo do tipo de dado MPEG-7 que deve ser retirado de MPEG7FileFormatCS (ISO, 2001g [seção 17.1.2.1.1])
Location	Informa a localização do arquivo, sendo do tipo mpeg7:MediaLocatorType
FileSize	Informa o tamanho do arquivo
System	Informa o tipo de sistema de cores utilizado, por exemplo, PAL, NTSC e DVB, sendo de um tipo de dado MPEG-7 que deve ser retirado de MPEG7SystemCS (ISO, 2001g [seção 17.1.2.3.1])
Bandwidth	Informa a largura de banda necessária para a mídia ser

	apresentada
Bitrate	Informa a taxa de bits necessária para a apresentação da mídia
variable	Utilizado se a taxa de bits é variável
minimum	Indica a taxa de bits mínima necessária para a apresentação da mídia
average	Indica a taxa de bits média para a apresentação da mídia
maximum	Indica a taxa de bits máxima necessária para a apresentação da mídia
AudioAttributes	Descreve as características sonoras da mídia
Coding	Indica o tipo de codificação utilizado na mídia, sendo de um tipo de dado MPEG-7 que deve ser retirado de MPEG7AudioCodingFormatCS (ISO, 2001g [seção 17.1.2.6.1])
Channels	Indica o número de canais sonoros da mídia em cada posição
front	Número de canais de áudio frontais
rear	Número de canais de áudio traseiros
side	Número de canais de áudio laterais
lfe	Número de canais de áudio de baixa frequência
track	Número de faixas nas quais o som está gravado
MixType	Indica o tipo de mixagem utilizado pela mídia, sendo de um tipo de dado MPEG-7 que deve ser retirado de MPEG7AudioPresentationCS (ISO, 2001g [seção 17.1.2.5.1])
VideoAttributes	Descreve as características visuais da mídia
Coding	Indica o tipo de codificação utilizado na mídia, sendo de um tipo de dado MPEG-7 que deve ser retirado de MPEG7VideoCodingFormatCS (ISO, 2001g [seção 17.1.2.4.1])
Scan	Indica o tipo de “scan” utilizado pela mídia (por exemplo, progressivo, entrelaçado), sendo do tipo de dado tva:ScanType
HorizontalSize	Indica a tamanho horizontal (em pixels) da mídia
VerticalSize	Indica a tamanho vertical (em pixels) da mídia
AspectRatio	Indica o aspecto da imagem (a proporção do tamanho horizontal e do vertical), sendo do tipo de dado tva:AspectRatioType
Color	Indica como é a coloração da mídia (branco e preta, colorida artificialmente...), sendo do tipo de dado tva:ColorType

```

<complexType name="MediaIType">
  <sequence>
    <element name="MediaID" type="mpeg7:UniqueIDType" use="required" />
    <element name="ObjectID" type="practio:PractioIDType"
      use="required"/>
    <element name="FileFormat" type="mpeg7:ControlledTermUseType"
      minOccurs="0"/>
    <element name="Location" type="mpeg7:MediaLocatorType"/>
    <element name="FileSize" type="nonNegativeInteger" minOccurs="0"/>
    <element name="System" type="mpeg7:ControlledTermUseType"
      minOccurs="0"/>
    <element name="Bandwidth" type="nonNegativeInteger"
      minOccurs="0"/>
    <element name="BitRate" minOccurs="0">
      <complexType>
        <simpleContent>
          <extension base="nonNegativeInteger">

```

```

        <attribute name="variable" type="boolean" use="default"
            value="false"/>
        <attribute name="minimum" type="nonNegativeInteger"
            use="optional"/>
        <attribute name="average" type="nonNegativeInteger"
            use="optional"/>

        <attribute name="maximum" type="nonNegativeInteger"
            use="optional"/>
    </extension>
</simpleContent>
</complexType>
</element>

<element name="AudioAttributes" minOccurs="0">
    <complexType>
        <complexContent>
            <restriction base="anyType">
                <sequence>
                    <element name="Coding"
                        type="mpeg7:ControlledTermUseType"
                        minOccurs="0"/>
                    <element name="Channels" minOccurs="0">
                        <complexType>
                            <simpleContent>
                                <extension base="nonNegativeInteger">
                                    <attribute name="front"
                                        type="nonNegativeInteger"
                                        use="optional"/>
                                    <attribute name="side" type="nonNegativeInteger"
                                        use="optional"/>
                                    <attribute name="rear" type="nonNegativeInteger"
                                        use="optional"/>
                                    <attribute name="lfe" type="nonNegativeInteger"
                                        use="optional"/>
                                    <attribute name="track"
                                        type="nonNegativeInteger"
                                        use="optional"/>
                                </extension>
                            </simpleContent>
                        </complexType>
                    </element>
                    <element name="MixType"
                        type="mpeg7:ControlledTermUseType"
                        minOccurs="0"/>
                </sequence>
            </restriction>
        </complexContent>
    </complexType>
</element>

<element name="VideoAttributes" minOccurs="0">
    <complexType>
        <complexContent>
            <restriction base="anyType">
                <sequence>
                    <element name="Coding"
                        type="mpeg7:ControlledTermUseType"
                        minOccurs="0"/>
                    <element name="Scan" type="tva:ScanType" minOccurs="0"/>
                    <element name="HorizontalSize" type="positiveInteger"

```

```

        minOccurs="0"/>
    <element name="VerticalSize" type="positiveInteger"
        minOccurs="0"/>
    <element name="AspectRatio" type="tva:AspectRatioType"
        minOccurs="0"/>
    <element name="Color" type="tva:ColorType"
        minOccurs="0"/>
    </sequence>
</restrictor>
</complexContent>
</complexType>
</element>
</sequence>
</complexType>

```

Figura 6.06 - Exemplo do Schema para a descrição das mídias

A partir das mídias e suas descrições, é possível construir diferentes interfaces (ou modos de apresentação) para um programa. A essa possibilidade dá-se a denominação de perfil. Para a codificação das mídias e das cenas, essa proposta considera o uso da tecnologia MPEG-4, que é bastante flexível e eficiente para a compactação de vídeo e de áudio. Como característica fundamental do MPEG-4, pode-se citar o fato dos componentes de uma cena estarem separados como objetos, permitindo, assim, incrementos nas formas de manipulação desses componentes. Essa característica é bastante significativa para o caso da implementação de diferentes perfis de programas de TVI.

É importante salientar que apesar deste trabalho considerar o uso de MPEG-4 para a apresentação e transmissão do programa, a estrutura apresentada é independente do mecanismo de transporte, podendo utilizar os padrões da TV Digital (ATSC, 2001; DVB, 1999; ARIB, 2000), bem como o MHP (*Multimedia Home Platform*) para a apresentação do conteúdo multimídia (DVB, 2000). As descrições das cenas e das mídias permitem a criação de ferramentas de busca em vídeos, em áudio e nas cenas.

Outro elemento importante da cena é o *timeline* (linha do tempo), que informa a posição temporal das mídias na cena. Esse elemento é do tipo *TimelineType*, que tem seus elementos apresentados na Tabela 6.04 e seu Schema, na Figura 6.07.

Tabela 6.04 - Subelementos de linha do tempo

Nome	Definição
TimelineType	É o tipo de dado que possui as informações sobre a linha do tempo da cena

Tobject	É um elemento que indica o objeto que terá sua ação temporal descrita
ID	É um identificador do objeto, sendo do tipo PracticIDType
MediaTime	É um elemento que informa o ponto em que a mídia deve começar e sua duração, sendo um elemento MPEG-7 do tipo mpeg7:MediaTimeType
SceneTimePoint	É um elemento que indica o instante da cena em que o objeto deve ser inserido, sendo um elemento MPEG-7 do tipo mpeg7:TimePointType

```

<complexType name="TimelineType">
  <sequence>
    <element name="TObject">
      <complexType>
        <complexContent>
          <restriction base="anytype">
            <sequence>
              <element name="MediaTime" type="mpeg7:MediaTimeType"/>
              <element name="SceneTimePoint"
                type="mpeg7:TimePointType"/>
            </sequence>
            <attribute name="ID" type="practic:PracticIDType"/>
          </restriction>
        </complexContent>
      </complexType>
    </element>
  </sequence>
</complexType>

```

Figura 6.07 – Exemplo do *Schemata* para os elementos da linha do tempo

O sumário e o *schedule* têm uma destacada importância para aplicações do tipo EPG (*Electronic Program Guides*), uma vez que é a partir do uso de EPGs que o usuário escolherá o que assistir e o que pretende gravar. Com as informações do sumário, o usuário pode escolher exatamente os programas a serem assistidos/gravados; com a informação do *schedule*, o usuário não precisa se preocupar com o horário de início do programa, uma vez que sua URD fará isso de forma transparente.

Outro uso relevante do *schedule* reside no fato de que, em um ambiente consciente de contexto e ubíquo, a URD pode informar, automaticamente, a um equipamento telefônico ou PDA (*Personal Digital Assistant*) o horário de início de um determinado programa.

Na **Seção 5.8.2**, discutiu-se a integração de um ou mais perfis de programa interativo ao desenvolvimento de programas interativos. Este trabalho utiliza o padrão MPEG-7 para a criação de novos elementos e para descrever alguns aspectos de layout, requisitos e contexto,

relevantes a esse perfil e suas interações. As informações relativas a um perfil estão dentro de um elemento *profile*. Esse elemento, por sua vez, é composto por outros elementos, que terão suas funcionalidades apresentadas na Tabela 6.05 e seu *Schema*, na Figura 6.08. A Figura 6.09 apresenta um exemplo de uso do cabeçalho.

Tabela 6.05 - Elementos de perfil

Nome	Definição
Profile	É o elemento que descreve as características do perfil do programa, sendo do tipo ProfileType
ProfileType	É um complexType , dentro do qual estão todos os elementos do perfil.
Layout	É um elemento, dentro do qual estará especificada a disposição dos elementos na tela, sendo do tipo LayoutType
Communication	É um elemento que informa as características necessárias para a transmissão e visualização do perfil, sendo do tipo CommunicationType
Context	É um elemento que informa como devem ser tratadas informações de contexto relevantes
name	Identifica o nome do perfil
id	Identificador único do programa, sendo do PracticIDType
programid	Indica o id do programa ao qual aquele perfil pertence, sendo do tipo tva:CRIDType

```
<element name="Profile" type="practic:ProfileType"/>
<complexType name="ProfileType">
  <sequence>
    <element name="Layout" type="practic:LayoutType"/>
    <element name="Communication" type="practic:CommunicationType"/>
    <element name="Context" type="practic:ContextType"/>
    <element name="Interaction" type="practic:InteractionType"/>
  </sequence>
  <attribute name="name" type="string"/>
  <attribute name="id" type="practic:PracticIDType"/>
  <attribute name="programid" type="tva:CRIDType"/>
</complexType>
```

Figura 6.08 - Exemplo do *Schema* para o elemento *profile*

```
<Profile name="Jogo_1" id="PR-00001"
  programid="crId://www.practic.tv.com/snowa/Jogol-2001-04-05/">
  <Layout bgcolor="#000000">
    ...
  </Layout>
  <Communication>
    ...
  </Communication>
  <Context>
    ...
  </Context>
```

```

</Context>
<Interaction>
...
</Interaction>
</Profile>

```

Figura 6.09 - Exemplo do cabeçalho de um perfil de programa

Os aspectos de apresentação definidos, que fazem parte do elemento Layout, são relativos às cores de fundo, às fontes e posições das mídias. Dentro do *layout* ainda, têm-se os controles de mídia possíveis para um determinado perfil, o tipo de menu e seu conteúdo. Uma versão simplificada do *Schema* de descrição do elemento *Layout* é apresentada na Figura 6.10 e a descrição de seus subelementos, na Tabela 6.06.

Tabela 6.06 - Subelementos de layout

Nome	Definição
LayoutType	É um complexType que indica as características do layout do perfil
bgcolor	Indica a cor de fundo padrão. É do tipo hexadecimal
Object	É o elemento que indica um objeto na cena. Um objeto pode ser um elemento de mídia ou uma descrição
id	Identificador único de objetos, sendo do tipo PracticIDType
xposition	Indica a posição horizontal do objeto
yposition	Indica a posição vertical do objeto
volume	Indica o volume do objeto
effect	Indica o efeito sonoro aplicado àquele objeto
Button	Esse é um elemento que apresenta um botão padrão na tela
id	É um identificador do botão, sendo do tipo PracticIDType
action	Indica a ação que deve ser disparada quando o botão é clicado
xposition	Define a posição na tela da margem esquerda do botão
yposition	Define a posição na tela da margem acima do botão
Menu	É o elemento responsável pelo menu do programa, sendo do tipo MenuType
Control	São os controles do programa (play, stop, rewind, forward, replay, change subtitle, change language, dentre outros)
name	É o nome do comando, sendo de um tipo de dado MPEG-7 que deve ser retirado de MPEG7ActionTypeCS (ISO, 2001g [seção 17.4.1.1.1])
id	É o identificador do comando
Action	É a ação disparada quando o comando é executado
MenuType	É um ComplexType que define um tipo de menu disponível para aquele programa de TVI
Selector	É um tipo de menu, sendo do tipo MenuSelector
Popup	É um tipo de menu, sendo do tipo MenuPopup
Bar	É um tipo de menu, sendo do tipo MenuBar
type	Identificador do tipo de menu (popup, selector, dentre outros)
position	Indica a posição do menu (top, down, left, right)
color	Indica a cor do menu. É do tipo hexadecimal


```

<complexType name="LayoutType">
  <sequence>
    <element name="Object">
      <complexType>
        <attribute name="id" type="practic:PracticIDType"/>
        <attribute name="xposition" type="nonNegativeInteger"/>
        <attribute name="yposition" type="nonNegativeInteger"/>
        <attribute name="vclume" type="nonNegativeInteger"/>
        <attribute name="effect" type="mpeg7:ControlledTurnUseType"/>
      </complexType>
    </element>
    <element name="Button">
      <complexType>
        <attribute name="id" type="practic:PracticIDType"/>
        <attribute name="action" type="string"/>
        <attribute name="xposition" type="nonNegativeInteger"/>
        <attribute name="yposition" type="nonNegativeInteger"/>
      </complexType>
    </element>
    <element name="Menu" type="practic:MenuType"/>
    <element name="Control">
      <complexType>
        <attribute name="name" type="string"/>
        <attribute name="id" type="nonNegativeInteger"/>
        <attribute name="action" type="string"/>
      </complexType>
    </element>
  </sequence>
  <attribute name="bgcolor" type="hexBinary"/>
</complexType>

<complexType name="MenuType">
  <choice>
    <element name="Popup" type="practic:MenuPopup"/>
    <element name="Selector" type="practic:MenuSelector"/>
    <element name="Bar" type="practic:MenuBar"/>
  </choice>
  <attribute name="type" type="practic:MenuTypeType"/>
  <attribute name="position" type="practic:MenuPositionType"/>
  <attribute name="color" type="hexBinary"/>
</complexType>

```

Figura 6.10 - Exemplo do *Schema* para o elemento layout

Os requisitos do perfil ficam especificados e armazenados dentro do elemento *Communication*, onde estão os requisitos de comunicação de cada mídia, os decodificadores necessários para cada uma dessas mídias e requisitos técnicos da URD. Na **Tabela 6.07** são apresentados os subelementos do elemento *Communication* e na **Figura 6.11** é apresentada uma versão simplificada do *Schema* de descrição do tipo *CommunicationType*.

Tabela 6.07 - Subelementos de comunicação

Nome	Definição
CommunicationType	É um complexType que contém elementos que indicam as características necessárias para a apresentação do perfil
CommRequirements	É um complexType onde estão os requisitos de cada mídia
Bandwidth	Quantidade de banda de rede necessária para a apresentação do perfil
Codec	Decodificadores e codificadores necessários para a apresentação da mídia, sendo um tipo MPEG-7 que deve ser retirado de MPEG7VideoCodingFormatCS (ISO, 2001g [seção 17.1.2.4.1]) ou de MPEG7AudioCodingFormatCS (ISO, 2001g [seção 17.1.2.6.1])

```

<complexType name="CommunicationType">
  <sequence>
    <element name="CommRequirements">
      <complexType complexContext="true">
        <sequence>
          <element name="Bandwidth">
            <simpleType>
              <restriction base="nonNegativeInteger" value="0"/>
            </simpleType>
          </element>
          <element name="Codec" type="mpeg7:ControlledTermUseType" />
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>

```

Figura 6.11 - Exemplo de Schema para o elemento *Communication*

Dentro do elemento *Context*, insere-se todo o tipo de informação de contexto que pode ser relevante para um determinado perfil. Esse elemento foi definido a partir de variáveis relevantes em consciência de contexto, tais como WHO, WHERE, WHEN e WHAT (Dey & Abowd, 2000), e discutidas no **Capítulo 4**. Esse elemento possui as ações que devem ser realizadas em determinadas situações. A **Tabela 6.08** apresenta a descrição desses elementos, e a **Figura 6.12** apresenta um exemplo de como utilizar o *Schema* para a contemplar consciência de contexto a partir do tipo *ContextType*.

Tabela 6.08 - Subelementos de consciência de contexto

Nome	Definição
who	Elemento que indica quem (é utilizado para a ação dos agentes no ambiente) e dependendo deste realizará uma ação
RefId	Identificador de who, sendo do tipo mpeg7:UserIdentifierType
where	Elemento que indica localização e dependendo desta realizará uma ação

RefId	Identificador do local
When	Elemento que indica tempo e dependendo deste realizará uma ação
RefId	Identificador de tempo, sendo do tipo <code>mpeg7:TimeType</code>
What	Elemento que indica o que está sendo feito e dependendo deste realizará uma ação
RefId	Identificador de “ocorrência”

```

<complexType name="ContextType">
  <complexContent>
    <extension base="string">
      <sequence>
        <element name="who">
          <complexType>
            <complexContent>
              <extension base="ContextType">
                <attribute name="RefId" type="mpeg7:UserIdentifierType" />
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="what">
          <complexType>
            <complexContent>
              <extension base="ContextType">
                <attribute name="RefId" type="string" />
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="when">
          <complexType>
            <complexContent>
              <extension base="ContextType">
                <attribute name="RefId" type="mpeg7:TimeType" />
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="where">
          <complexType>
            <complexContent>
              <extension base="ContextType">
                <attribute name="RefId" type="string" />
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Figura 6.12 – Exemplo de *Schema* para os elementos de consciência de contexto

A utilização dos subelementos do elemento *Context* se dá através do aninhamento dos mesmos, de modo semelhante à utilização do “*if-then-else*” em linguagens de programação.

Desse modo, é possível associar ações às situações relevantes que acontecem no ambiente.

Um exemplo é mostrado na **Figura 6.13**.

```
<Context>
  <who RefId = "son">hide(video1)</who>
  <who RefId = "m. 2">
    <where RefId = "bedroom">
      <when RefId = "24:00"> openProfile(news1) </when>
      openProgram(news2)
    </where>
    openProgram(sportnews)
  </who>
  <what RefId = "empty"> turnoff() </what>
</Context>
```

Figura 6.13 – Exemplo do uso dos elementos de consciência de contexto

No exemplo anterior, é mostrado o uso da consciência de contexto dentro de um perfil; no entanto, os elementos de consciência de contexto podem ser utilizados tanto no elemento *<program>* quanto no elemento *userpreferences*, indicando qual é o perfil mais adequado de um programa para uma determinada pessoa. No elemento *<userpreferences>*, o elemento *<context>* é utilizado com a mesma sintaxe e, no caso do programa, ele será parte do *InformationElement*, que representa uma contribuição deste trabalho.

O último importante elemento é o *Interaction*, no qual encontram-se todas as ações que podem ser realizadas para cada interação definida para um perfil. Essas interações são descritas segundo a **Tabela 6.09**; na **Figura 6.14** são mostrados os elementos de interação.

Tabela 6.09 – Subelementos de interação

Nome	Definição
InteractionType	É um complexType , que possui a seqüência de interações dadas pelo elemento IT.
IT	É o elemento necessário a cada interação, sendo do tipo ITType
ITType	É um complexType que descreve o que deve ser feito quando uma ação é realizada
On	É um atributo que indica que tipo de interação foi feita (click, move, double-click, dentre outros)
Media	É um atributo que indica a mídia com a qual ocorreu a interação
Button	É um atributo que indica o botão (de um controle remoto) com o qual ocorreu a interação
action	É a ação que deve ocorrer (função que será chamada)

```
<complexType name="InteractionType">
  <sequence>
    <element name="IT" type="practicITType"/>
  </sequence>
</complexType>
```

```

<complexType name="ITType">
  <attribute name="on" type="string"/>
  <attribute name="media" type="practic:PracticIDType"/>
  <attribute name="button" type="practic:PracticIDType"/>
  <attribute name="action" type="string"/>
</complexType>

```

Figura 6.14 – Exemplo do *Schema* para os elementos de interação

No exemplo da **Figura 6.15**, o atributo *move* é uma descrição MPEG-7, o atributo *video3* foi apresentado no programa e descrito em MPEG-7 e a ação *moveMedia(video3)* representa um comando que será dado à URD para realizar a tarefa. É importante notar que esse comando depende do Sistema Operacional presente na URD e das aplicações que estão presentes no ambiente, tais como MHP ou JavaTV (Sun, 2001).

```

<Interaction>
  <IT on="move" media="video3" action="moveMedia(video3)">
  <IT on="click" button="volume" action="openVolumeBox()">
</Interaction>

```

Figura 6.15 – Exemplo do uso dos elementos de interação

Além dos elementos descritos, existe mais um elemento que segue a sintaxe dos perfis, denominado *UserPreferences*. O elemento *UserPreferences*, diferentemente dos outros elementos, fica armazenado na URD; nele estão armazenadas algumas personalizações feitas pelos usuários para um determinado perfil de um programa. Os elementos de personalização do usuário são descritos na **Tabela 6.10**; o *Schema* é apresentado na **Figura 6.16** e um exemplo de uso desses elementos é apresentado na **Figura 6.17**.

Tabela 6.10 – Elementos de personalização do usuário

Nome	Definição
UserPreferences	É um elemento que armazenará as personalizações dos perfis e programas
UserPreferencesType	É um complexType onde serão armazenadas todas as personalizações de perfis, de programas e de hábitos para assistir televisão
user	É um atributo com o nome do usuário, sendo um tipo de dado mpeg7:UserIdentifierType
Profile	É o elemento que conterà as mudanças no perfil, sendo do tipo ProfileType
CustomPreferences	É um elemento que informa outras personalizações não relativas somente a um programa ou perfil, e sim relacionadas a situações ou, então, personalizações válidas para todos os perfis e/ou programas.

```

<element name="UserPreferences" type="UserPreferencesType">
  <complexType name="UserPreferencesType">
    <sequence>
      <element name="Profile" type="ProfileType">
        <element name="CustomPreferences" type="TelementType">
        </element>
      </sequence>
      <attribute name="user" type="mpeg7:UserIdentifierType"
        user "required"/>
    </complexType>
  </element>

```

Figura 6.16 – Exemplo de Schema para os elementos de preferências do usuário

```

<UserPreferences user="Calli">
  <MediaVolume grade="25" effect = "ultrabass" />
  <profile="Sports3" program="NewSports">
    <Layout bgcolor="ccffcc" />
  </profile>
</UserPreferences>

```

Figura 6.17 – Exemplo do uso dos elementos de preferências do usuário

Com os elementos apresentados e utilizando as descrições do MPEG-7 (ISO, 2001g; ISO, 2001h) e do *TV-Anytime Forum* (TVAF, 2001b; TVAF, 2001c), pode-se especificar um amplo conjunto de características singulares, além de personalizações para os perfis e programas.

6.7 Exemplos de programas de TV Interativa

A linguagem visual para a TV Interativa deve ter características da linguagem já utilizada nos programas da televisão tradicional, da linguagem utilizada nos PDRs, e da linguagem presente em *sites* da WWW. Com isso, os programas adaptar-se-ão às novas funcionalidades disponíveis. A seguir, serão apresentados alguns exemplos de programas da TV tradicional e algumas mudanças que serão possíveis a partir da proposta aqui reportada.

Na parte visual em um evento, como um jogo de futebol, vôlei ou basquete, deve-se permitir a presença de múltiplas câmeras fornecendo vários ângulos da partida; na parte de áudio, a escolha da narração entre os vários narradores disponibilizados pela emissora do programa, ou o áudio original, se o programa for estrangeiro; na parte de dados relacionados, links para dados dos jogadores, dos times, das partidas e dos campeonatos, sendo que alguns links podem ser exibidos em determinados momentos, como, por exemplo, após um gol em uma partida de futebol pode-se exibir links para replay, dados do jogador e da partida, dentre

outros. Alguns desses exemplos já estão disponíveis na *Enhanced TV*, porém de uma forma diferente e muito limitada.

No caso dos filmes, as possíveis mudanças são em direção a múltiplos ângulos, cenas alternativas e dados relacionados ao filme e aos atores. Esse tipo de interação já é possível e, em alguns casos, comum, como ocorre com os DVDs. Comandos como retroceder e avançar um filme, comuns nos PDRs, também podem ser disponibilizados pela emissora. Como complemento, as produtoras de filmes podem fazer um novo tipo de *merchandising* criando links aos seus produtos à medida que eles aparecem na tela, ou, então, mudando os produtos de acordo com as características do usuário.

Telejornais podem ter diversas “faces”, ora privilegiando imagens, ora privilegiando notícias relacionadas, programas relacionados ou links na WWW relacionados com cada notícia. Cada notícia também pode ter uma versão expandida com maior duração que aquela mostrada no telejornal. O usuário, por sua vez, também pode personalizar seu telejornal a alguns assuntos de seu interesse.

Outro tipo de programa tradicional que pode sofrer mudanças são os programas infantis, que podem incentivar a interação das crianças para a resposta de perguntas ou oferecerem mais informações sobre objetos na tela, tornando-se, assim, educativos. Já existem programas infantis para a *Enhanced TV* em transmissão nos EUA.

Um programa educativo pode utilizar as técnicas, já bastante estudadas, da educação por computador à distância e adaptá-las para a tela de TV e para os diferentes controles da TV Interativa. A TV pode ser uma interface mais interessante e acessível às pessoas em geral, ao contrário do que ainda ocorre com o computador tradicional.

Programas de *Home Shopping* podem sofrer uma melhoria significativa de qualidade e eficiência com a interação, uma vez que o consumidor pode obter informações sobre o produto e comprá-lo. Além disso, é possível personalizar o canal de *Home Shopping*, de modo a oferecer produtos atraentes, segundo o perfil do usuário. Os programas de *Home Shopping* podem ser adaptados para serem chamados a partir do link nos produtos em filmes, comerciais, notícias e outros programas de TV.

Jogos Interativos também são uma parte motivadora da TV Interativa; os novos aparelhos, como o Xbox, PlayStation 2 e o Game Cube já têm características para jogos em redes e seus fabricantes pensam em adaptá-los para a TV Interativa. *Gameshows* são outros programas que podem ter um ganho bastante grande com a possibilidade de interação. Já existem programas interativos nos quais o usuário pode responder às perguntas ao mesmo tempo em que o participante do programa real e, no final do mês, o telespectador que mais acumular pontos ganha prêmios. Esses programas fazem uso dos recursos da *Enhanced Television* e, para a interação (que é bastante limitada), utilizam a linha telefônica. Considerando a TV Interativa com um canal dedicado, muitas outras características podem surgir, como a formação de equipes para competir, auxílio da rede para o participante, dentre outras.

A propaganda para a TV Interativa deve ter um novo paradigma, uma vez que diversas novas variáveis serão introduzidas, como a propaganda direcionada (só alguns recebem a propaganda), a propaganda adaptável (dependendo de quem está assistindo, a propaganda é diferente) e a possibilidade de interação com o produto e, até mesmo, a compra via TV.

6.8 Aplicação dos conceitos de programa e de perfil a um programa de evento esportivo

Nesta seção, é apresentada uma proposta para um *programa esportivo* que tenha como conteúdo uma partida de futebol, possibilitando a aplicação dos conceitos de *perfis* em um programa de televisão interativa. A situação proposta para esse programa permite visualizar as novas formas de interação que o uso de *perfis* acrescenta à interação do usuário em TVI, evidenciando que tanto o usuário quanto a própria aplicação interagem com a infra-estrutura descrita na **Seção 6.4**.

Na exibição de um evento esportivo em TVI, algumas características são esperadas no contexto da interação do usuário, dentre as quais pode-se destacar: a) oferta de diferentes ângulos de câmeras; b) oferta de várias opções de áudio, tais como diferentes locutores e idiomas; c) acesso a informações estatísticas dos times que estão jogando, do próprio jogo e dos jogadores; d) oferta de situações de *replay*, além do convencional; e) oferta de informações sobre assuntos relacionados, como outros jogos simultâneos, por exemplo; f) acesso a informações sobre o programa (sumário) e a programação; g) oferta de programas relacionados, como jogos do mesmo time na programação do mês, por exemplo.

Em termos de construção da aplicação, essas características devem ser implementadas e descritas segundo os elementos do programa e do perfil apresentados na **Seção 6.7**. Cada perfil pode possuir características únicas que devem ser especificadas. Nesse ponto, alguns exemplos de variáveis de um perfil podem ser como os seguintes:

- *layout*: aspectos de cor de fundo e das imagens/figuras; estilo dos botões, estilo de menu, posição do menu, posição das mídias na interface do usuário (quando possível); quais mídias devem ser apresentadas, quais mídias podem ter suas posições mudadas e quais podem ser as posições finais;
- *communication*: quais são os *codecs* necessários para cada mídia, bem como as taxas de largura de banda necessárias para cada mídia;
- *context*: quais são os comandos de decisão que utilizam as variáveis de contexto, tais como *quem* está usando, *qual* o horário de uso, *quais* outros aparelhos estão ligados e/ou conectados, *o que* já foi visto até um determinado momento;
- *interaction*: quais são as mudanças em relação às mídias e ao *layout*, por exemplo, a partir da interação do usuário com determinados grupos de botões/controles/menus associados a essas mídias e *layout*.

Vale ressaltar que a definição de uma instância de um programa, como é o caso de um *evento esportivo*, não limita a aplicação do conceito de *perfis* em TVI. Deve-se observar que tanto a infra-estrutura como os *perfis* podem ser aplicados, por exemplo, em ambientes interativos que contenham aplicações diferentes da TVI, como é o caso da videoconferência em um ambiente de ensino-aprendizagem.

6.8.1 Definindo o programa

Conforme apresentado, o programa é o elemento central de uma aplicação de TVI, sendo responsável por agrupar os macro-elementos que descrevem a aplicação.

Para uma instância de programa, como é o caso do evento esportivo, observa-se que a definição dos elementos deve contemplar características específicas da aplicação, porém de forma associada aos elementos genéricos modelados. Os componentes do programa de evento esportivo são apresentados **Tabela 6.11**.

Tabela 6.11 – Componentes do programa

Objeto	Descrição
Video_Principal	É o <i>stream</i> de vídeo principal da transmissão, semelhante à transmissão atual de jogos de futebol. Composto por imagens de diversas câmeras
Video_Topo	É o <i>stream</i> de vídeo das imagens da câmera que fica acima do estádio, em um dirigível.
Video_Gol_esq	É o <i>stream</i> de vídeo da imagem do gol da esquerda. São utilizadas mais de uma câmera para esse <i>stream</i> de vídeo: uma atrás da trave, uma acima da trave e uma lateralmente (para escanteios e impedimentos)
Video_Gol_dir	É o <i>stream</i> de vídeo da imagem por trás da trave do gol da direita; assim como o vídeo anterior, esse é composto por imagens de mais de uma câmera
Video_Amplo	Apresenta a imagem do jogo de longe e é útil para examinar a tática do jogo
Video_Zoom	Apresenta a imagem do jogo de perto, focalizando jogadores e jogadas
Audio_Principal	Canal de áudio principal com locução e comentários
Audio_Locucacao2	Canal de áudio do locutor alternativo
Audio_Banco_esq	Canal de áudio com o som do banco de reservas do lado esquerdo
Audio_Banco_dir	Canal de áudio com o som do banco de reservas do lado direito
Audio_Ambiente	Canal de áudio com o som ambiente da partida, sem locução
estatistica_time1	Dados e links referentes ao time 1
estatistica_time2	Dados e links referentes ao time 2
estatistica_jogo	Estatísticas do jogo
Campeonato	Informações online de outros jogos do campeonato

6.8.2 Descrevendo o programa

A descrição MPEG-7 do programa de evento esportivo deve contemplar todos os elementos definidos no modelo de programa, a começar pelo elemento *Scene*, a partir do qual é construída a árvore de elementos.

Uma vez descrito o elemento *Scene*, pode-se instanciar elementos de mídia, de interação, de *timeline*, de comunicação, de *layout*, e os elementos de composição, que são o *Summary* e o *Schedule*. A **Figura 6.18** apresenta a descrição MPEG-7 para o programa de evento esportivo.

```
<Program>
  <Information name "jogo Sao Paulo x Palmeiras"
    id "crld://www.practictv.com/shows/Jogo1-2001-04-05/" />
  <Scene>
    <Media>
      <MediaID>
        <IDOrganization>
          <FreeTerm>RedeGlobo</FreeTerm>
        </IDOrganization>
```

```

    <IDName>
      <FreeTerm>Video_Principal</FreeTerm>
    </IDName>
    <UniqueID>CM-01-2001-04-05</UniqueID>
  </MediaID>
  <ObjectID>CM-00001</ObjectID>
  <FileFormat href="urn:mpeg:MP EG7FileFormatCS:3">
    <Name xml:lang="en">mp4</Name>
  </FileFormat>
  <Location>
    <MediaUri>http://www.practictv.com/video/video2.mp4</MediaUri>
    <StreamID>1</StreamID>
  </Location>
  <System href="urn:mpeg:MP EG7SystemCS:1">
    <Name xml:lang="en">PAL</Name>
  </System>
  <Bandwidth>968</Bandwidth>
  <BitRate minimum="62"/>
  <VideoAttributes>
    <Coding href="urn:mpeg:MP EG7VisualCodingFormatCS:4"
      colorDomain="color">
      <Name xml:lang="en">MP EG-4</Name>
    </Coding>
    <Scan>progressive</Scan>
    <HorizontalSize>640</HorizontalSize>
    <VerticalSize>480</VerticalSize>
    <AspectRatio>4:3</AspectRatio>
    <Color type="color"/>
  </VideoAttributes>
</Media>
<Media>
<Media>
<Media>
</Media>
<Media>
<Media>
</Media>
<Timeline>
  <Object ID="CM-01-2001-04-05">
    <MediaTime>
      <MediaTimePoint>T00:00:00:00:0F0</MediaTimePoint>
      <MediaDuration>T3530M</MediaDuration>
    </MediaTime>
    <SceneTimePoint>2001-04-05T18:00:00:0F0</SceneTimePoint>
  </Object>
  <Object ID="CM-02-2001-04-05">
  </Object>
  <Object ID="CM-03-2001-04-05">
  </Object>
  <Object ID="CM-04-2001-04-05">
  </Object>
  <Object ID="CM-05-2001-04-05">
  </Object>
  <Object ID="CM-06-2001-04-05">
  </Object>
  <Object ID="AU-01-2001-04-05">
  </Object>
  <Object ID="AU-02-2001-04-05">
  </Object>
  <Object ID="AU-03-2001-04-05">
  </Object>
  <Object ID="AU-04-2001-04-05">
  </Object>

```

```

    <Object ID="AU-05-2001-04-05">
    </Object>
</Timeline>
<Element> </Element>
<Description> </Description>
</Scene>
<Summary> </Summary>
<Schedule> </Schedule>
</Program>

```

Figura 6.18 - Exemplo da descrição de um programa de evento esportivo

6.8.3 Descrevendo os Perfis

A partir do conteúdo disponibilizado pelo programa, foram definidos três perfis para o programa de evento esportivo. O primeiro é o perfil básico, bastante semelhante à transmissão atual, com pequena possibilidade de interação. O segundo perfil contempla pequena possibilidade de interação e troca de imagens e sons, porém com grande ênfase ao acesso às informações relacionadas ao evento. O terceiro perfil é o mais completo, com total possibilidade de interação por parte do usuário e com muitas opções de imagens e de sons. A seguir, é apresentada, na **Figura 6.19**, a implementação do primeiro perfil; a **Figura 6.20** mostra o segundo perfil, enquanto a **Figura 6.21** apresenta o terceiro e mais completo dos perfis especificados.

```

<Profile name="basico" id="PR-00001">
  <Layout bgcolor="#000000">
    <Object id="CX-00001" xposition="0" yposition "0"/>
    <Object id="AU-00001" volume="35" effect="loudness"/>
    <Button id="BT-00001" action "show(Descricao_Time_1)"
      xposition="40" yposition "430">Palmeiras</Button>
    <Button id="BT-00002" action "show(Descricao_Time_2)"
      xposition="560" yposition "430">São Paulo</Button>
    <Control/>
  </Layout>
  <Communication>
    <CommRequirements>
      <Bandwidth>986</Bandwidth>
      <Codec>MPEG4 Video Codec</Codec>
      <Codec>MPEG4 Audio</Codec>
    </CommRequirements>
  </Communication>
  <Interaction>
    <If on=" " midia=" " action=" " />
  </Interaction>
</Profile>

```

Figura 6.19 - Exemplo do perfil básico

A seguir, será apresentado o segundo perfil, que possui diversas características de interação, principalmente com conteúdo relacionado.

```

<Profile name="intermediario" id="PR-00002"
  programid="cr:d://www.practicv.com/shows/Jogol-2001-04-05/"
  xmlns="http://intermedia.icmc.sc.usp.br/SCOTIMM/PRACTIC/PRACTICs.xsd"
  xmlns:xml="http://www.w3.org/2000/10/xml.xsd">
  <Layout bgcolor="#000000">
    <Object id="CM-00001" xposition="0" yposition="0"/>
    <Menu type="Popup" position="down" color="000000">
      <Head id="Som"> Som
      <Item id="IT-01001" action="opensound(Audio_Principal)">
        principal (Galvão Bueno)
      </Item>
      <Item id="IT-01002" action="opensound(Audio_Locucac2)">
        alternativo (Sílvio Luis)
      </Item>
    </Head>
    <Head id="Info">Informações
      <Item id="IT-02001"
        action="show(Descriçao_Time_1)">Palmeiras</Item>
      <Item id="IT-02002" action="show(Descriçao_Time_2)">
        São Paulo
      </Item>
      <Item id="IT-02003" action="show(Campeonato)">outros jogos
      </Item>
    </Head>
    <Head id="estatisticas">Estatísticas
    <Item id="IT-03001" action="show(Estatisticas.bolaemjogo)">
      Tempo de bola em jogo</Item>
      <Item id="IT-03002" action="show(Estatisticas.impedimentos)">
      Impedimentos</Item>
      <Item id="IT-03003" action="show(Estatisticas.passescertos)">
      Passes Certos</Item>
      <Item id="IT-03004" action="show(Estatisticas.passeserrados)">
      Passes Errados</Item>
      <Item id="IT-03005" action="show(Estatisticas.finalizacoes)">
      Finalizações</Item>
    </Head>
  </Menu>
  <Control/>
</Layout>
<Communication>
  <CommRequirements>
    <Bandwidth>986</Bandwidth>
    <Codec>MPEG4_Video_Codec</Codec>
    <Codec>MPEG4_Audio</Codec>
  </CommRequirements>
</Communication>
<Interaction>
  <IT on=" " midia=" " action=" "/>
</Interaction>
</Profile>

```

Figura 6.20 – Exemplo do perfil intermediário

O terceiro perfil é o avançado, possuindo como característica uma grande diversidade de imagens e sons.

```

<Profile name="avancado" id="PR-00003"
  programId="crid://www.practictv.com/shows/Jogo1-2001-04-05/"

xmlns="http://intermedia.icmc.usp.br/SCOTIYM/PRACTIC/PRACTICs.xsd"
xmlns:xml "http://www.w3.org/2000/10/xml.xsd">
<Layout bgcolor="#0000CC">
  <Object id "CY-00001" xposition "0" yposition="0"/>
  <Menu type "Popup" position="down" color="0000CC">
    <Head id "Som"> Som
      <Item id="IT-01001" action="opensound(Audio_Principal)">
        principal (Galvão Bueno)</Item>
      <Item id "IT-01002" action="opensound(Audio_Locucão)">
        alternativo (Sílvio Luís)</Item>
      <Item id="IT-01003" action="opensound(Audio_Banco_esq)">
        Banco de reservas do Palmeiras</Item>
      <Item id "IT-01004" action="opensound(Audio_Banco_dir)">
        Banco de reservas do SãoPaulo</Item>
      <Item id "IT-01005" action="opensound(Audio_Ambiente)">
        Som Locução (Som Ambiente)</Item>
    </Head>
    <Head id "Camera"> Câmera
      <Item id "IT-02001" action="open(Video_Principal)">
        principal </Item>
      <Item id "IT-02002" action="open(Video_Principal,380,320)">
        principal em janela</Item>
      <Item id="IT-02003" action="open(Video_Topo)">
        Câmera do Dirigível</Item>
      <Item id="IT-02004" action="open(Video_Topo,380,320)">
        Câmera do Dirigível em janela</Item>
      <Item id="IT-02005" action="open(Video_Gol_esq)">
        Lado Esquerdo do Campo</Item>
      <Item id="IT-02006" action="openfull(Video_Gol_esq,380,320)">
        Lado Esquerdo do Campo em janela</Item>
      <Item id="IT-02007" action="open(Video_Gol_dir)">
        Lado Direito do Campo</Item>
      <Item id "IT-02008" action="open(Video_Gol_dir,380,320)">
        Lado Direito do Campo em janela</Item>
      <Item id "IT-02009" action="open(Video_Amplo)">Campo
        Inteiro</Item>
      <Item id "IT-02010" action="open(Video_Amplo)">Campo
        Inteiro</Item>
      <Item id="IT-02011" action="open(Video_Zoom)">Zoom</Item>
    </Head>
    <Head id "Info">Informações
      <Item id="IT-03001" action="show(Descricao_Time_1)">
        Palmeiras</Item>
      <Item id="IT-03002" action="show(Descricao_Time_2)">
        São Paulo</Item>
      <Item id="IT-03003" action="show(Campeonato)">outros
        jogos</Item>
    </Head>
    <Head id="estatisticas">Estatísticas
      <Item id "IT-04001" action="show(Estatisticas.bolaemjogo)">
        Tempo de bola em jogo</Item>
      <Item id="IT-04002" action="show(Estatisticas.impedimentos)">
        Impedimentos</Item>
      <Item id "IT-04003" action="show(Estatisticas.passeocortes)">

```

```

        Passes Certos</Item>
    <Item id "IT-04004" action="show(Estatisticas.passeseerrados)">
        Passes Errados</Item>
    <Item id "IT-04005" action="show(Estatisticas.finalizacoes)">
        Finalizações</Item>
</Feed>
</Menu>
<Control/>
</Layout>
<Communication>
    <CommRequirements>
        <Bandwidth>986</Bandwidth>
        <Codec>MPEG4 Video Codec</Codec>
        <Codec>QuickTime Video Codec</Codec>
        <Codec>Windows Media Video Codec</Codec>
        <Codec>Windows Video Codec</Codec>
        <Codec>WAV Codec</Codec>
        <Codec>MP3-1 Layer3</Codec>
    </CommRequirements>
</Communication>
<Interaction>
    <IT on=" " midia " " action=" " />
</Interaction>
</Profile>

```

Figura 6.21 – Exemplo do perfil avançado

No terceiro exemplo de perfil, pode-se observar a utilização dos elementos de contexto que não foram utilizados nos outros dois perfis. A utilização desses elementos não é obrigatória e, apesar de no exemplo terem sido usados para persuadir os usuários a comprarem produtos, esses elementos podem ser utilizados de modo a melhorar a interação do usuário com a televisão.

Observando os três exemplos, percebe-se que, utilizando os conceitos de programa e perfil, é possível o reaproveitamento de conteúdo para a apresentação de diferentes programas, de um modo relativamente simples. Esses conceitos também são úteis na personalização da TV e permitem a utilização de consciência de contexto nesse ambiente.

6.9 Considerações finais

Quando se pensa na televisão interativa inserida em um ambiente consciente de contexto, pode-se vislumbrar inúmeras novas aplicações.

É possível pensar que as redes de televisão conseguirão fazer uma transmissão diferenciada para diferentes contextos; os anunciantes poderão transmitir suas propagandas somente para seu público-alvo; certas cenas poderão ser excluídas em determinadas situações (a presença

de uma criança, por exemplo). No entanto, deve-se observar que não só a transmissão mudará, mas também a apresentação passará por sensíveis mudanças.

Neste capítulo, foi apresentada, como estudo de caso, a modelagem completa de uma aplicação para TVI com base tecnológica centrada nos padrões MPEG-4 e MPEG-7. Como contribuição, apresentou-se um modelo para composição de um programa interativo a partir dos conceitos de programa e perfil, também formalizados por este trabalho.

No próximo e último capítulo desta tese, são apresentadas e discutidas as conclusões deste trabalho, bem como descritas as limitações evidenciadas, que permitem a identificação de trabalhos futuros e de um cenário para o desenvolvimento de novas pesquisas.

7.1 Considerações iniciais

Conforme discutido neste trabalho, a evolução e integração dos sistemas digitais e o surgimento de novas e avançadas tecnologias multimídia têm oferecido espaço para o desenvolvimento de novas e complexas aplicações interativas.

Ao considerar o estado da arte representado pela evolução e convergência dos sistemas digitais (computadores, televisão, telefones e telefonia, infra-estrutura de comunicação de dados, PDAs - *Personal Digital Assistant*, dentre outros) e propor a utilização dos padrões da família MPEG - MPEG-4, para representação de conteúdo multimídia na forma de objetos de mídia (ISO, 2001a); e MPEG-7, para descrição de informação associada ao conteúdo multimídia (ISO, 2001b) - este trabalho ofereceu contribuições singulares para o campo de modelagem de ambientes interativos conscientes de contexto, uma vez que ainda são poucos os trabalhos científicos dessa natureza.

O foco deste trabalho é a proposta de uma modelagem para ambientes interativos conscientes de contexto, especificando métodos para integração de tecnologias como MPEG-4 (ISO, 2001a), MPEG-7 (ISO, 2001b), XML. (Connolly, 1997), dentre outras, e apontando para modelos de desenvolvimento de aplicações interativas, tais como televisão interativa, videoconferência, vídeo-sob-demanda, dentre outras. A **Figura 7.01** - que é uma reprodução da **Figura 1.01**, apresentada no **Capítulo 1** e recolocada aqui para auxiliar na compreensão do texto - apresenta a estrutura gráfica do escopo que foi coberto pela modelagem proposta, sendo importante relembrar os três modelos distintos que compõem a modelagem:

- **NECTAR** (*Network Environment with Context-Awareness Rules*): cobre a infra-estrutura de comunicação necessária a aplicações interativas conscientes de contexto, principalmente em termos dos requisitos de rede;
- **SPICE** (*Schemes for Interaction on Context-Aware Environments*): cobre os aspectos de modelagem que definem esquemas e critérios de interação para o modelo de interações completas (*complete interaction* → *user-network-application*);

- PRACTIC** (*Profile for Context-Aware Interactive Applications*): cobre o modelo da geração de perfis de aplicações interativas, sendo composto pelo **PRACTIC-E** (*PRACTIC Program Examples*) – que cobre a especificação de perfis para instâncias de aplicações interativas, como é o caso de programas de TVI – e pelo **PRACTIC-S** (*PRACTIC Schema*) – que cobre a especificação de esquemas de descrição para aplicações interativas genéricas;

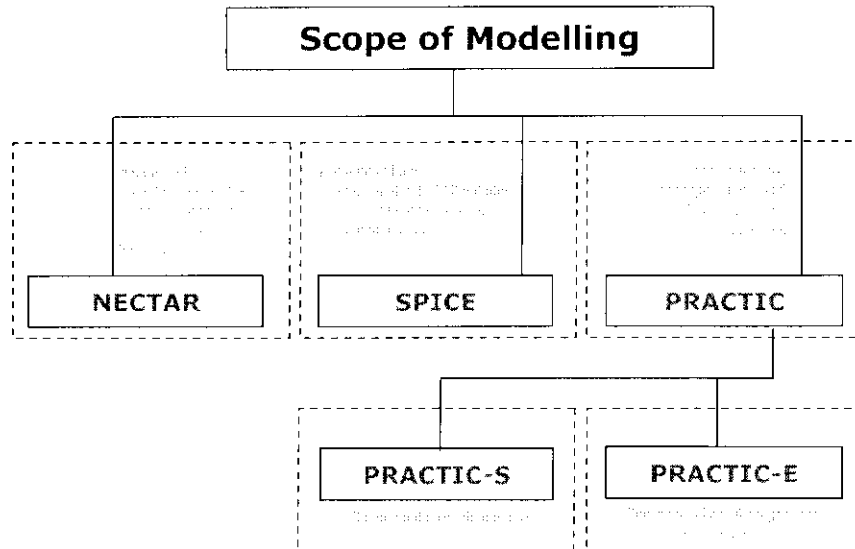


Figura 7.01: O escopo da modelagem – estrutura gráfica (Santos Jr. et al., 2001b)

Nesta proposta, os agentes de software são vistos como componentes do ambiente e estão presentes nos três modelos propostos (**NECTAR**, **SPICE** e **PRACTIC**), servindo, portanto, como elemento integrador desses modelos. Neste sentido, a modelagem proposta explora os recursos da plataforma JADE (*Java Agents Development Framework*), que está em desenvolvimento nos laboratórios TILAB (Bellefemine et al., 1999; Bellefemine et al., 2001). Portanto, no universo desta proposta, requisitos para o uso de agentes foram identificados e classificados, regulando as formas de provimento de informações contextuais para os elementos que compõem as aplicações interativas conscientes de contexto, quer seja ao nível da infra-estrutura de comunicação (**NECTAR**), ao nível da interação usuário-ambiente (**SPICE**) ou ao nível da personalização do ambiente (**PRACTIC**).

Neste capítulo, são discutidas as contribuições desta tese, o que ocorre na **Seção 7.2**. Em seguida, na **Seção 7.3**, são apontadas algumas das limitações do trabalho e, na **Seção 7.4**, são descritos alguns dos trabalhos futuros. Na **Seção 7.5**, são apresentadas algumas considerações finais.

7.2 Contribuições

Ao longo da tese, observou-se que a principal contribuição original deste trabalho é a modelagem que visa definir e formalizar novos tipos de interação gerados por uma aplicação inserida em um ambiente interativo consciente de contexto. Nesses ambientes, mecanismos tradicionais como VCR e STB (*Set-Top Box*) tendem a ser substituídos por dispositivos sensoriais, o que sinaliza para a interação usuário-ambiente ao invés da interação usuário-computador, simplesmente. A modelagem auxilia o desenvolvimento dessas aplicações interativas por apresentar um conjunto bem definido das interações geradas pela comunicação entre as entidades usuário, rede, aplicação e terminal (Santos Jr. et al., 2001a; Santos Jr. et al., 2001b). A modelagem, ao contemplar a consciência de contexto, apresenta avanços para o desenvolvimento de aplicações complexas, como é o caso dos ambientes classificados como *wearable computing* e *ubiquitous computing*.

Os modelos **NECTAR**, **SPICE** e **PRACTIC** consideram as potencialidades dos padrões MPEG-4 e MPEG-7. Particularmente, o MPEG-7 é utilizado para oferecer suporte à personalização do ambiente através de esquemas de descrição das cenas multimídia.

As inovações presentes neste trabalho estão concentradas no fato de que os padrões, tecnologias e modelos, propostos até então, não consideram as interações existentes entre a infra-estrutura de comunicação, a aplicação e os objetos de mídia que compõem uma ou mais cenas multimídia, uma vez que a representação das mídias como objetos foi possível somente após a padronização do MPEG-4. Desta forma, os aspectos de consciência de contexto são utilizados sempre ao nível da aplicação e não há preocupação em formalizar os critérios de interação para uso da consciência de contexto e, nem mesmo, de formalizar a semântica dos elementos contextuais WHO, WHAT, WHERE e WHEN (Dey & Abowd, 2000). Ao formalizar os aspectos de consciência de contexto, dentro do universo das aplicações interativas, este trabalho representa um avanço na direção dos novos tipos de interação que são esperados para a nova geração de aplicações multimídia. Algumas funções tradicionais de controle remoto, por exemplo, tendem a desaparecer e outros tipos de mecanismos de controle devem surgir muito em breve, o que reforça a importância da modelagem dos ambientes interativos conscientes de contexto, e não somente das aplicações multimídia.

Este trabalho também inovou na direção da definição dos conceitos de perfil de aplicação interativa e de programa interativo, que são apresentados de forma compatível com as tecnologias MPEG-4 e MPEG-7, gerando extensões a essas tecnologias de modo a possibilitar o uso integrado da aplicação com o ambiente.

A pesquisa realizada nesta tese abrangeu tópicos de diversas subáreas da Ciência da Computação e, portanto, as contribuições são tratadas separadamente nas subseções seguintes.

7.2.1 Contribuições para o campo da modelagem de sistemas multimídia

No campo da **modelagem de sistemas multimídia**, a modelagem proposta estende o modelo proposto por Sauer e Engels (Sauer e Engels, 2000), uma vez que considera aspectos de consciência de contexto e explorar a codificação das mídias como objetos, através do suporte oferecido pelo padrão MPEG-4. Adicionalmente, a modelagem proposta também estende as categorias de interatividade e níveis de personalização propostas por Van Setten e seus colaboradores (Van Setten et al., 2001), especialmente através da proposta de formalização de consciência de contexto pela concepção das regras contextuais.

7.2.2 Contribuições para o campo dos agentes de software

No campo dos **agentes de software**, os requisitos para uso de agentes de software identificados nesta proposta impactam diretamente o desenvolvimento da plataforma JADE (*Java Agents Development Framework*), proposta por Bellefemine e seus colaboradores (Bellefemine et al., 2001). Novas API têm sido desenvolvidas para suportar consciência de contexto e a produção de regras contextuais. Os requisitos aqui identificados têm sido incorporados à plataforma JADE através de aplicações MPEG-J, conforme discutido no **Capítulo 5**.

7.2.3 Contribuições para o campo da consciência de contexto

No campo da **consciência de contexto**, o modelo SPICE estende o modelo proposto por Dey e Abowd (Dey & Abowd, 1999) oferecendo suporte ao uso de informações contextuais em todos os níveis (aplicação, infra-estrutura de comunicação, interação do usuário, mídias) da composição estrutural de um ambiente consciente de contexto.

7.2.4 Contribuições para o desenvolvimento de aplicações multimídia interativas

Por último, no campo das **aplicações**, mais precisamente da **televisão interativa**, a modelagem proposta especifica como produzir, distribuir e apresentar um programa de TVI, adicionando o conceito de *profile interativo* a esse programa, o que caracteriza uma inovação no campo da Televisão Interativa.

A modelagem também agrega valor aos padrões MPEG-4 e MPEG-7, ao apresentar uma visão coerente sobre como integrar essas tecnologias no desenvolvimento de aplicações interativas. Essa visão, baseada nos conceitos de programa e perfil, oferece esquemas eficientes para representação de conteúdo multimídia e meta-informação associada ao conteúdo, que são objetos de trabalho para toda a cadeia produtiva de aplicações de TVI.

7.2.5 Contribuições para o estabelecimento de novos modelos de negócios

O desenvolvimento das novas tecnologias de informação é uma oportunidade estratégica para as organizações que desejam aumentar sua competitividade na sociedade da informação. Frequentemente, surgem notícias que demonstram o impacto estratégico dessas novas tecnologias, desde as mais inovadoras que permitem a compra de bilhetes de cinema a partir de dispositivos móveis de comunicação, à evolução dos modelos de negócio mais tradicionais, como é o caso dos bancos, por exemplo. O comércio eletrônico (*e-commerce*) e os negócios eletrônicos (*e-business*) são hoje a prova tangível de que empresas e setores da economia fazem parte de um ambiente competitivo baseado na utilização das novas tecnologias e da Internet (Amor, 2000).

O gestor daquela empresa que pretende ter sucesso num futuro próximo enfrenta assim novos desafios, e precisa compreender profundamente as estratégias que podem resultar da utilização das novas tecnologias.

Neste cenário, a modelagem aqui proposta, quando apresenta uma referência para o desenvolvimento de aplicações inseridas em um ambiente interativo consciente de contexto e especifica as entidades (e seus comportamentos e relacionamentos) que formam ambientes dessa natureza, contribui ao oferecer os elementos necessários à compressão dessas novas tecnologias, especialmente em como integrá-las, considerando a evolução natural dos sistemas digitais.

Observa-se que a apresentação do MPEG-21 nesta tese sinaliza para o fato de que itens digitais universais poderão ser utilizados como elementos independentes, porém integrados, na construção de aplicações, acrescentando novos requisitos tecnológicos. O tratamento da informação poderá ser globalizado, criando novos cenários virtuais com maior grau de realismo, em relação aos procedimentos que são utilizados nos modelos de negócios já consolidados. Isto é um fato relevante, uma vez que muitas organizações ainda resistem ao uso das novas tecnologias quando encontram dificuldades em representar seus modelos tradicionais de negócios (muitas vezes de sucesso) em modelos digitais.

7.3 Limitações

Uma limitação inicial está relacionada à escassez de ferramentas de autoria que permitam a implementação de todos os elementos e componentes propostos pela modelagem. No entanto, esta limitação é temporária, uma vez que há crescente interesse da indústria em produzir ferramentas para suportar os padrões MPEG-4 e MPEG-7 (Envivio, 2001; Sinfonia, 2001).

Quanto à modelagem, especialmente no campo dos agentes, existem limitações na definição de ontologias que validem por completo os diagramas *use-case* discutidos no **Capítulo 5**. Observa-se que os requisitos identificados para o uso de agentes descrevem somente as primitivas fundamentais para suportar o uso desses agentes no fornecimento de informações contextuais e na personalização do ambiente interativo. Neste sentido, novas formas de uso de agentes podem ser necessárias em ambientes com novas complexidades, especialmente para que o uso de agentes seja compatível com as especificações de padrões como o FIPA (*Foundation for Intelligent Physical Agents*).

Em relação à infra-estrutura de comunicação, o modelo **NECTAR** contempla os elementos fundamentais para prover informações contextuais sobre a rede e sobre os dispositivos do ambiente. No entanto, é necessário expandir o conceito de personalização a tecnologias que contemplem maior grau de inteligência e não simplesmente o automatismo. Assim, a modelagem não contempla especificações de *redes ativas*, por exemplo, o que limita os níveis de personalização às informações contextuais disponíveis.

7.4 Trabalhos Futuros

Considerando a modelagem desenvolvida e reportada ao longo dos capítulos desta tese, pode-se discutir a continuidade do trabalho aqui reportado, enfocando, assim como ocorreu

nas contribuições, as diversas subáreas da Ciência da Computação que podem ser impactadas por pesquisas futuras.

No campo da **consciência de contexto**, pode-se realizar extensões no modelo **SPICE**, visando sua completude em termos de critérios de interação. Em adição, extensões podem ser feitas em termos da inserção de novos componentes estruturais nas classes contextuais, expandindo a cobertura dos elementos de contexto a partir dos formalismos das regras contextuais. Complementando, pode-se especificar novas descrições MPEG-7 que possam ser utilizadas como classes de critérios de interação.

Estendendo os critérios de interação, espera-se que novas contribuições sejam dados no campo dos **ambientes personalizados**. Neste sentido, os conceitos de **programa** e **perfil**, definidos neste trabalho, podem ser estendidos para suportar novos elementos de personalização, especialmente aqueles que se relacionam com a infra-estrutura de comunicação, estabelecendo novas associações entre os modelos **NECTAR** e **PRACTIC**. Neste ponto, a investigação sobre propriedades das redes ativas pode acrescentar novos itens de personalização e permitir a definição de níveis diferenciados de qualidade de serviço.

Observa-se, então, que no campo das **redes de computadores e infra-estrutura de comunicação** uma contribuição importante poderia ser a implementação de uma infra-estrutura com suporte a redes ativas em ambiente *multicasting*, investigando novos níveis de interação entre a aplicação interativa e o ambiente em que essa aplicação está inserida. Tais contribuições poderiam estender as funcionalidades previstas no modelo **NECTAR**.

No campo da **hipermídia**, muitos trabalhos de pesquisa podem ser definidos, indo desde a implementação de EPGs à criação de programas interativos, incluindo ferramentas de busca (*searching tools*) em vídeos que utilizem MPEG-7 ou XML para a descrição de meta-informação associada às cenas.

No campo dos **agentes de software**, espera-se que extensões possam ser feitas para os requisitos de uso desses agentes. Ontologias podem ser definidas a partir do modelo proposto para interação com os agentes, e essas ontologias podem ser criadas para suportar níveis de automatismo em aplicações que recomendam programas ao usuário, de acordo com suas preferências e comportamento.

No campo da **interação usuário-ambiente**, espera-se que extensões sejam produzidas para os *Schema* MPEG-7, de modo a suportar outros elementos de layout, tais como novos tipos de menu. Adicionalmente, tornam-se relevantes trabalhos que comparem o comportamento do usuário em um programa de TVI ao seu comportamento num *site* na WWW e/ou assistindo a um programa da TV tradicional. Tais contribuições podem ser significativas num cenário de *emotional interfaces*, por exemplo.

Por fim, no campo de **banco de dados**, pode-se trabalhar na estruturação de um modelo de banco de dados multimídia com suporte às novas tecnologias interativas, como é o caso do MPEG-4, oferecendo suporte ao armazenamento e recuperação de objetos de mídia. Trabalhos dessa natureza podem contribuir sobremaneira para o desenvolvimento de aplicações de comércio eletrônico, por exemplo.

7.5 Considerações finais

Neste capítulo foram apresentadas as conclusões deste trabalho, ressaltando-se também alguns dos trabalhos que podem ser realizados como continuidade deste.

Quando se pensa em aplicações interativas inseridas em um ambiente consciente de contexto, pode-se vislumbrar inúmeras novas aplicações. É possível pensar, por exemplo, que as redes de televisão conseguirão fazer uma transmissão diferenciada para diferentes contextos; os anunciantes poderão transmitir suas propagandas somente para seu público-alvo; certas cenas poderão ser excluídas em determinadas situações (a presença de uma criança, por exemplo). No entanto, deve-se observar que não só a transmissão mudará, mas também a apresentação passará por sensíveis mudanças. Portanto, a modelagem de um ambiente completo é justificada pela necessidade de compreensão e formalização desse novo universo de potenciais aplicações.

Por fim, a importância desse trabalho foi reconhecida na comunidade pela aceitação, no Brasil e no exterior, de cinco trabalhos científicos (Santos Jr. et al., 2001a; Santos Jr. et al., 2001b; Faria, Santos Jr., Moreira & Goularte, 2001; Santos Jr. et al., 2002a; Santos Jr. et al., 2002b).



Referências Bibliográficas

(Abert, 2000)

ABERT. *Testes em Sistemas de Televisão Digital* (Fevereiro de 2000) - [On Line] http://www.anatel.gov.br/biblioteca/publicacao/relatorio_tvdigital_cp216.htm

(Abowd, 1999)

Abowd, G.D. (1999): *Software Engineering Issues for Ubiquitous Computing*. Proceedings of International Conference on Software Engineering (ICSE'99), pp.75-84, 1999-May.

(Amor, 2000)

Amor, D.: *The e-Business (R) Evolution: Living and Working in na Interconnected World*. 1st Edition. Upper Saddle River, NJ: Prentice Hall, 2000. p. 636.

(ATVEF, 1995)

Advanced Television Systems I - *Digital Audio Compression Standard (AC-3) - (A/52)* (Dezembro de 1995) - [On Line] - http://www.atsc.org/Standards/A52/A_52A.pdf

(ATVEF, 2000)

Advanced Television Enhancement Forum - *Enhanced Content Specification* (2000) - [On Line] http://www.atvef.com/library/spec1_1a.html

(ATVEF, 2001)

Advanced Television Systems I - *ATSC Digital Television Standard (Revision A) - (A/53)* (Abril de 2001) - [On Line] - http://www.atsc.org/Standards/A53/a_52.pdf

(Anatel, 2001)

Agência Nacional de Telecomunicações (ANATEL) - *Relatório Integrador dos Aspectos Técnicos e Mercadológicos da TV Digital - versão 1.0 - (28/03/2001)* - [On Line] http://www.anatel.gov.br/eventos_publicos/Consulta/2001/consulta_291/Relatorio_Integrador.exe

(ARIB, 2000)

Association of Radio Industries and Businesses (ARIB). - ITU Document 6M/J2-E - *Operational Guidelines of ARIB's XMI-based Multimedia Coding Scheme(Ver.1.2)* - (Setembro de 2000) - [On Line] - <http://www.itu.int/itudoc/itu-r/sg6/docs/wp6m/2000-03/contrib/009e.html>

(ATSC, 2001)

ATS (2001): Advanced Television Systems Committee: *ATSC DIGITAL TELEVISION STANDARD (Revision A)* [2001-April] <http://www.atsc.org/Standards/A53/>. Available on 2001.

(Backer et al., 1998)

Backer, G.; Bennington, J.; Boltax, J.; Gage, K.; Garcia, M.; Richmond, M.; Vickers, M.; Watson, S.; Zigmond, D. - *A Framework for Interactive Television Based on Internet Standards* - Apresentado no Workshop Television and the Web - (Julho de 1998) - [On Line] http://www.atvef.com/library/9806_framework.html

(Barbieri, 2001)

Barbieri, M.; Ceccarelli, M.; Mekenkamp, G.; Nesvadba, J. – *A Personal TV Receiver with Storage and Retrieval Capabilities* – Proceedings of the 1st Workshop on Personalization in Future TV. Sonthofen, Germany, (Julho de 2001).
<http://www.di.unito.it/~liliana/LM01/barbieri.pdf>

(Bates & Gregory, 1997)

Bates, R. J. and Gregory, D. (1997): *Voice and Data Communications Handbook*. McGraw-Hill Series on Computer Communications, 1997.

(Battista et al., 1999)

Battista, S.; Casalino, F.; Lande, C. *MPEG-4: A Multimedia Standard for the Third Millennium, Part1* – IEEE Multimedia, págs 74-83 (Outubro-Dezembro 1999).

(Bellifemine et al., 1999)

Bellifemine, F. et al. (1999): *JADE A FIPA-compliant agent framework*. Proceedings of PAAM'99. London, pp.97-108. 1999-August.

(Bellifemine et al., 2001)

Bellifemine, F. et al (2001): *Developing multi agent systems with a FIPA-compliant agent framework*. In: Software - Practice And Experience, N.31, pp.03-128. 2001- February.

(Berger, 1995)

Berger, D. (1995): *Video-on-Demand Metadata Query Interfaces*. Master Thesis. University of California at Berkeley, 1995.

(Bernes-Lee et al., 1994)

Berners-Lee, T.; Cailliau, R.; Luotonen, A.; Henrik, N. and Secret, A. (1994): *The World Wide Web*. Communications of the ACM, N.8, V.37, pp.76-82, 1994.

(Bernes-Lee et al., 1998)

Berners-Lee, T. et al. (1998): *Uniform Resource Identifiers (URI): Generic Syntax*. Internet RFC 2396. Available on 1998.

(Blásquez, 1998)

Blásquez, M. (1998): *Building Ontologies at the Knowledge Level using Ontology Design Environment*. KAW'98, Canada. 1998.

(Booch et al., 1996)

Booch, G.; Rumbaugh, J.; Jacobson, I. – *UML: Unified Modelling Language for Object-Oriented Development Documentation set. Addendum*, Santa Clara, California, Rational Software Corp, July 1996.

(Bray et al., 1997)

Bray, T. et al. (1997): *Extensible Markup Language (XML) – Principles, Tools and Techniques*. World Wide Web Journal, V.2, N.4, pp.29-66, 1997.

(Buford, 1994)

Buford, J. (1994): *Uses of Multimedia Information in Multimedia Systems*. Addison-Wesley, 1994.

(Buford et al., 1994)

Buford, J.; Rutledge, L. and Keskin, C. (1994): *HyOctane: a HyTime Engine for an MMIS*. Multimedia Systems, N.4, V.1, pp.173-185, 1994.

(Calvert et al., 1998)

Calvert, K. L. et al. (1998): *Directions in Active Networks*. IEEE Communications Magazine 36(10), pp.72-78, 1998.

(Chiariglione, 1999)

Chiariglione, L.: MPEG-4 - The fusion of natural and synthetic audio and video. Spring Conference in Computer Graphics 99, Budmerice, May-1999.

(Chiariglione, 2001)

Chiariglione, L.: Standard technologies to develop the multimedia applications. MPEG-4 Conference, Paris, 2001.

(Churchill, 1998)

Churchill, E. F. and Snowdown, D. (1998): *Collaborative Virtual Environments: An Introductory Review of Issues and Systems*. Virtual Reality, V.3, N.1, 1998.

(Comer, 1995)

Comer, D. E. (1995): *Internetworking with TCP/IP*. 3rd Edition, Prentice-Hall, 1995.

(Connolly, 1997)

Connolly, D. (1997): *XML. Principles, Tools and Techniques*. World Wide Web Journal. O'Reilly, V.2, Issue 4, 1997.

(Cooper, 1997)

Cooper, A. (1997): *About face: The Essential of User Interface Design*. IDG Books, 1997.

(Cornell, 1997)

Cornell, G. and Horstmann, C.S. (1997): *Core JAVA - Guia Autorizado da SUN Microsystems*. Editora Makron Books, São Paulo, 1997.

(Coulouris et al., 1994)

Coulouris, G.; Dollimore, J. and Kindberg, T. (1994): *Distributed Systems: Concepts and Design*. 2nd Edition, Addison-Wesley Publishing Company, 1994.

(Daubechies, 1990)

Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961-1005, September 1990.

(DeRose & Durand, 1994)

DeRose, S. J.; Duran, D. G. *Making Hypermedia Work: A User's Guide to HyTime*, Kluwer Academic Publishers, Massachusetts, 1994.

(Dey & Abowd, 1999)

Dey, A. K and Abowd, G. D. (1999): *CyberDesk: The Use of Perception in Context Aware Computing*. 1st Workshop on Perceptual User Interfaces, pp.26-27, 1999.

(Dey & Abowd, 2000)

Dey, A. K. and Abowd G.D. (2000): *Towards a Better Understanding of Context and Context-Awareness*. CHI 2000 - Workshop on The What, Who, Where, When, and How of Context-Awareness. The Netherlands, 2000-April.

(DVB, 1999)

Digital Video Broadcasting (DVB) - ETSI EN 300 744, - *Framing structure, channel coding and modulation for digital terrestrial television versão 1.2.1* (07/99) [On line] <http://www.etsi.org/>

(DVB, 2000)

Digital Video Broadcasting (DVB) - ETSI TS 101 812, - *Multimedia Home Platform (MHP) Specification 1.1 (Draft)* [On Line] <http://www.mhp.org/>

(Ebrahimi & Horne, 2000)

Ebrahimi, T. and Horne, C. (2000): *MPEG-4 Natural Video Coding - an Overview*. Signal Processing, Image Communication 15(4-5), pp.365-385. 2000.

(Engels, 2000)

Engels, G.: *Object-Oriented Modelling of Multimedia-Applications*. Handbook of Interaction on Multimedia Applications. 2000. [On-line] <ftp://cs.pitt.edu/chang/handbook/54b.pdf>

(Envivio, 2001)

The Envivio Tools for MPEG-4 Standard and Broadcast. The Envivio Corporation. [On-line] <http://www.envivio.com>

(Evain, 1998)

Evain, J. P. - *The Multimedia Home Platform - an overview* - [On Line] Extensible Markup Language (XML) 1.0 , W3C Recommendation, <http://www.w3c.org> February 1998.

(Faria, Santos Jr., Goularte & Moreira, 2001)

Blengini, G. F., Junior, J. B. S., Goularte, R., Moreira, E. S.. *Uso de Perfis em Aplicações de Televisão Interativa Conscientes de Contexto*. Anais do SBMidia 2001 - VII Simpósio Brasileiro de Sistemas Multimídia e Hiperídia, pp. 139-154. Florianópolis - SC, Brasil, outubro de 2001.(Impresso no Brasil - Eds: Roberto Willrich e Cesar Augusto Camilo Teixeira).

(Finseth, 1998)

Finseth, C.; Thomas, G.- *Guide to TV Broadcast URLs-* [On Line] <http://ietf.org/internet-drafts/draft-finseth-guide-01.txt>

(Fluckiger, 1995)

Fluckiger, F. (1995): *Understanding Networked Multimedia: Applications and Technology*. Editora Prentice Hall, 1995.

(Fröhlich, 1997)

Fröhlich, P.; Wolfgang, N. *A Database-Oriented Approach to the Design of Educational Hyperbooks*, Proceedings of the Workshop "Intelligent Systems on the World Wide Web", 8th World Conference of the AIED society, Kobe, Japan, August 1997.

(Galbreath, 1995)

Galbreath, J. (1995): *Compressed Digital Videoconferencing*. Educational Technology, pp.31-38, 1995.

(Garzotto et al., 1993)

Garzotto, F. et al. (1993): *HDM: A Model-based Approach to Hypertext Application Design*. The ACM Transactions on Information Systems, V.11, N.2, pp.1-26, 1993-January.

(Goularte, 1998)

Goularte, R. – *Utilização de Metadados no Gerenciamento de Acesso a Servidores de Vídeo* (Fevereiro de 1998) – Dissertação de mestrado, ICMC/USP, São Carlos 1998.

(Halasz, 1994)

Halasz, F. G. and Schwartz, M. (1994): *The Dexter Hypertext Reference Model*. Communications of the ACM, V.37, N.2, pp.30-39, 1994-February.

(Harrison, 1996)

Harrison, M. (1996): *The Essentials Elements of Hypermedia*. Academic Press, 1996.

(Hashisuca, 1996)

Hashisuca, A. M. M. (1996): *Desenvolvimento de um Sistema Servidor de Aplicações Multimídia*. Dissertação (Mestrado), 1996. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

(Herwijnen, 1994)

Herwijnen, E. van (1994): *Practical SGML*. Kluwer Academic Publishers, 2a Edition, 1994.

(Isakovitz et al., 1995)

Isakovitz, T. et al. (1995): *RMM: A Methodology for Structured Hypermedia Design*. Communications of the ACM, V.38, N.8, pp.34-44, 1995-August.

(Isakovitz et al., 1998)

Isakovitz T.; Kamis, A.A; Koufaris, M. *The Extended RMM Methodology for Web*, Publishing Working paper, I S-98-18, Center for Research on Information Systems, 1998.

(ISO, 1986a)

International Organization for Standardization, ISO/IEC:IS-8879, *Information Processing- Text and Office System- SGML*, 1986.

(ISO, 1986b)

ISO/IEC IS 8879. *Information Processing - Text and Office Systems – Standards Generalized Markup Language (SGML)*. Available on 1986.

(ISO, 1992)

International Organization for Standardization, ISO/IEC: 88744, *Hypermedia/Time-Based Structuring Language (HyTime)*, 1992.

(ISO, 1994)

ISO/IEC 13818-3, *Information Technology: Generic coding of Moving pictures and associated audio: Part 3 – Audio – (1994)*

(ISO, 1995)

International Organization for Standardization, ISO/IEC: 13522-1, *Information Technology Coding of Multimedia and Hypermedia Information*, Part. 5: MHEG, 1995.

(ISO, 1997a)

ISO/IEC IS 13818, *MPEG-2: Generic Coding of Moving Pictures and Audio Information*. Available on 1997.

(ISO, 1997b)

ISO/IEC JTC1/SC29/WG11 N1909, *Overview of the MPEG-4 Version 1 Standard*. <http://tilab.com/mpeg/public/w1909.htm>. December-1997.

(ISO, 1997c)

ISO/IEC JTC1/SC29/WG11 N1921, *Third Draft of MPEG-7 Requirements*. <http://tilab.com/mpeg/public/w1921.htm>. Available on December-1997.

(ISO, 1997d)

ISO/IEC 13818-7 - *Information Technology: Generic coding of Moving pictures and associated audio: Part 7 - Advanced Audio Coding (AAC) - (1997)*

(ISO, 1997e)

ISO/IEC IS 13818 - *MPEG-2: Generic coding of moving pictures and audio information - (1997)*

(ISO, 1998a)

ISO/IEC JTC1/SC29/WG11 N1902 - *Generic Coding of Audio-Visual Objects: Part 2 - Visual - (Novembro de 1998)*.

(ISO, 1998b)

ISO/IEC JTC1/SC29/WG11 N2460 - *MPEG-7: Context and Objectives (version - 10 Atlantic City) - (Outubro de 1998)*.

(ISO, 1999a)

ISO/IEC JTC1/SC29/WG11 N2725 - *MPEG-4 Overview - (Seoul Version) - (Março de 1999)*.

(ISO, 1999b)

ISO/MPEG N3201, *DDL Working Draft 1.0, MPEG Systems subgroup*. Maui, http://tilab.com/mpeg/public/mpeg-7_dll.zip. December-1999.

(ISO, 1999c)

ISO/MPEG N3158, *Overview of the MPEG-7 Standard, MPEG Requirements Group*. Maui. <http://tilab.com/mpeg/standards/mpeg-7/mpeg-7.htm>. December-1999.

(ISO, 1999d)

ISO/MPEG N2739 ISO/IEC 14496-1/PDAM1 subpart 3 - *MPEG-4 version 2 MPEG-J* (Março de 1999)

(ISO, 2000a)

ISO/MPEG N3445 *MPEG-7 overview (version 3.0)*. <http://tilab.com/mpeg/standards/mpeg-7/mpeg-7.htm>. Available on 2000.

(ISO, 2000b)

ISO/IEC JTC1/SC29/WG11 N3537 - *New MPEG-4 Profiles Under Consideration* - (Julho de 2000).

(ISO, 2001a)

ISO/MPEG N3536, *MPEG-4 Overview (v.15 - beijing version)*.
<http://tilab.com/mpeg/standards/mpeg-4/mpeg-4.htm>. Available on 2001.

(ISO, 2001b)

ISO/MPEG N4001, ISO/IEC FCD 15938-1 *Information Technology - Multimedia Content Description Interface - Part 1 Systems*, MPEG Systems SubGroup, [Março de 2001].
http://www.cselt.it/mpeg/public/mpeg-7_systems_fcd.zip

(ISO, 2001c)

ISO/MPEG N4002, ISO/IEC FCD 15938-2 *Information Technology - Multimedia Content Description Interface - Part 2 Description Definition Language*, MPEG Systems SubGroup, [15/01/2001]. http://www.cselt.it/mpeg/public/mpeg-7_ddl_fcd.zip

(ISO, 2001d)

ISO/MPEG N4062, ISO/IEC FCD 15938-3 *Information Technology - Multimedia Content Description Interface - Part 3 Visual*, MPEG Visual Group, [Março de 2001].
http://www.cselt.it/mpeg/public/mpeg-7_visual_fcd.zip

(ISO, 2001e)

ISO/MPEG N4004, ISO/IEC FCD 15938-4 *Information Technology - Multimedia Content Description Interface - Part 4 Audio*, MPEG Audio Group, [Março de 2001].
http://www.cselt.it/mpeg/public/mpeg-7_audio_fcd.zip

(ISO, 2001f)

ISO/MPEG N3996, ISO/IEC FCD 15938-5 *Information Technology - Multimedia Content Description Interface - Part 5 Multimedia Description Schemes*, Multimedia Description Schemes (MDS) Group [Março de 2001]. http://www.cselt.it/mpeg/public/mpeg-7_mds_fcd.zip.

(ISO, 2001g)

ISO/MPEG N4006, ISO/IEC FCD 15938-6 *Information Technology - Multimedia Content Description Interface - Part 6 Reference Software*, Implementation Studies Group [Março de 2001]. http://www.cselt.it/mpeg/public/mpeg-7_reference_software_fcd.zip.

(ISO, 2001h)

ISO/MPEG N4031, *Overview of the MPEG-7 Standard (version 5.0)*, MPEG Requirements Group, ISO, Singapore. (Março de 2001). <http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm>. [5/13/01]

(Jacobson et al., 1999)

Jacobson, I.; Booch, G. and Rumbaugh, J. (1999): *The Unified Software Development Process*. Addison-Wesley Object Technology Series, 1999.

(Jennings & Wooldrige, 1995)

Jennings, N. R. and Wooldrige, M. (1995): *Agents Theories, Architectures and Languages: a Survey*. Intelligent Agents, pp.55-67, 1995.

(Johnson, 1999)

Johnson, M. (1999): *XML for the Absolute Beginners*. The JAVA World, 1999-April. Disponível on-line (URL): <http://www.javaworld.com/javaworld/jw-04-1999/jw-04-xml-p.html>

(Kate, 1998)

Kate, W.; Radha, H. - *Bringing the Web to the TV: Convergence Scenarios* - (julho de 1998) - Apresentado no Workshop Television and the Web [On Line] - <http://www.w3.org/Architecture/1998/06/Workshop/paper38/>

(Ketchpel & Genesereth, 1994)

Ketchpel, S. and Genesereth, M. (1994): *Software Agents*. Communications of the ACM, p.48-53, 1994-July.

(Koenen, 2001)

Rob Koenen, Object-based MPEG offers flexibility, EE Times, 12 November 2001, CMP <http://www.eetimes.com/story/OEG20011112S0042>.

(Kurapati et al., 2001)

Kurapati, K.; Gutta, S.; Schaffer, D.; Martino, J.; Zimmerman, J. - *A multi-agent TV recommender* - Proceedings of the 1st Workshop on Personalization in Future TV. Sonthofen, Germany, (Julho de 2001). [on-line] <http://www.di.unito.it/~liliana/UM01/kurapati.pdf>

(Lemair & Shae, 1997)

Lemair, M. H. W. and Shae, Z. Y. (1997): *Videoconferencing over Packet-Based Networks*. IEEE JSAC, V.15, N.1, 1997-August.

(Li et al., 1998)

Li, X. et al. (1998): *Layered Video Multicast with Retransmission (LVMR): Evaluation of hierarchical rate control*. IEEE INFOCOM'98, pp.1062-1072. 1998.

(Lie, 2000)

Lie, H. W., Bos, B. - *Cascading Style Sheets, level1*- [On Line] <http://www.w3.org/TR/1999/REC-CSS1-19990111>

(Little & Venkatesh, 1994)

Little, T. D. C. and Venkatesh, D. (1994): *Client Server Metadata Management for the Delivery of Movies in a Video-On-Demand System*. 1st International Workshop on Services in Distributed and Networked Enviroments, Prague, Czech Republic, 1994-June.

(Maes, 1994)

Maes, P. (1994): *Agents that Reduce Work and Information Overload*. Communications of the ACM, pp.31-40, 1994-July.

(Martins, 2001)

Martins, L. *Transmissão de Vídeo Usando IPe6 e Multicasting em Redes de Alto Desempenho*. Dissertação de Mestrado, ICMC/USP, 2001.

(McCanne et al., 1996)

McCanne, S. et al. (1996): *Receiver-driven layered multicast*. SIGCOMM'96. 1996.

(Millet et al., 1996)

Millet, J. R.; Schwabe, D.; Lanzelotte R. *Autoria de Aplicações Hiperídia Utilizando um Banco de Dados Orientado a Objetos*, XI Simpósio Brasileiro de Banco de Dados, São Carlos, 1996.

(Moreira & Walczowski, 1997)

Moreira, D. A. and Walczowski, L. T. (1997): *Using Software Agents to Generate VLSI Layouts*. IEEE Expert Intelligent Systems, pp.26-32. 1997-November.

(Moreira et al., 1995)

Moreira, E. S.; Nunes, M. G. V.; Pimentel, M. G. C. (1995): *Design Issues for a Distributed Hypermedia-based Tutoring System (HyDTS)*. International Conference on Computer Application in Industry, Cairo, Egypt, 1995-December.

(Morse, 1998)

Morse, D.R., Ryan, N.S. & Pascoe, J. (1998): Enhanced reality fieldwork using hand-held computers in the field. *Life Sciences Educational Computing*, 9(3): 18-20.

(MPEG-21, 2001)

MPEG-21: Requirements: From MPEG-1 to MPEG-21: Creating an Interoperable Multimedia Infrastructure. N4518. ISO/IEC. Pattaya, Thailand. December-2001.

(Nack & Lindsay, 1999)

Nack, F. and Lindsay, A. T. (1999): *Everything You Wanted to Know About MPEG-7: Part 1*. IEEE Multimedia, pp.65-77.

(Nemetz, 1997)

Nemetz, F. *Hypermedia Modelling Technique: An Object-Oriented Design Model for Hypermedia Applications*, CHI 97, 1997.

(Pascoe et al., 1999)

Pascoe, J. et al. (1999): *Issues in Developing Context-Aware Computing*. In Gellersen, H. (ed.): *Handheld and Ubiquitous Computing*, Lecture Notes in Computer Science, V.1707. Springer-Verlag, Berlin Heidelberg New York, pp.208-221. 1999.

(Pereira & Ebrahimi, 2002)

Pereira, F. & Ebrahimi, T. (2002): *The MPEG-4 Book*. IMSC Press Multimedia Series. 2002.

(Perkins, 1997)

Perkins, C. (1997): *Mobile Networking Through Mobile IP*. IEEE Internet Computing Online, 1997.

(Pimentel, Santos Jr. & Fortes, 1998)

Pimentel, M. G. C.; Santos Jr., J. B. dos & Fortes, R. P. M. (1998): *Modelagem, Autoria e Apresentação de Documentos Didáticos Estruturados*. Anais do IV Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia. Rio de Janeiro, V.1, p.211-224, maio de 1998.

(Practic, 2001a)

PRACTIC (2001a): PRACTIC Program Example [On line]
<http://www.icmc.sc.usp.br/futuretv/practic.html>. Available on 2001.

(Practic, 2001b)

PRACTIC (2001b): PRACTIC Schema [On line]
<http://www.icmc.sc.usp.br/futuretv/practic.html>. Available on 2001.

(Preece, 1994)

Preece, J. (1994): *Human-computer Interaction*. Addison-Wesley, 1994.

(Pressman, 1995)

Pressman, R. S. (1995): *Engenharia de Software*. Makron Books, 1995.

(Rodrigues, 1998)

Rodrigues, P.; Gafsi, J.; Nonnenmacher, J. - *A More Attractive and Interactive TV* - (Julho de 1998) - Apresentado no Workshop Television and the Web [On Line] - <http://www.w3.org/Architecture/1998/06/Workshop/paper08>

(Rossi et al., 1995)

Rossi G.; Garrido A.; Carvalho S. *Object-Oriented Patterns for Hypermedia Applications*, Proceedings of Patterns Languages of Programs (PLOP'95), 1995.

(Rumbaugh, 1991)

Rumbaugh, J. et al. *Object-Oriented Modelling and Design*, Prentice-Hall, 1991.

(Salber, 1999)

Salber, D. et al. (1999): *The Context Toolkit: Aiding the Development of context-Enabled Applications*. In proceedings of CHI'99, pp.434-441. April-1999.

(Santos Jr. et al., 2001a)

Santos Jr., J.B. dos; Goularte, R.; Faria, G.; Moreira, E.S.: Modelling of User Interaction in Context-Aware Interactive Television Application on Distributed Environments. Proceedings. First Workshop on Future Television at 8thInternational Conference on User Modelling (UM'2001). Sonthofen, Germany. July-2001. LNAI 2001 Springer Verlag, N1, V2, pp.49-56. (Printed in Germany).

(Santos Jr. et al., 2001b)

Santos Jr., J.B. dos; Goularte, R.; Moreira, E. S.; Faria, G. B.:The Modelling of Structured Context-Aware Interactive Environments. SDPS Journal. Transactions of the SDPS. Issue 5, Number 4, pp. 77-93. December-2001. (Printed in USA).

(Santos Jr. et al., 2002a)

Santos Jr., J.B. dos; Guglielmo, M.; De Petris, G.; Moreira, E. S.; Goularte, R.; Faria, G. B.; Cordara, G.; Laia, A. C.: Production, Distribution and Presentation of Educational Multimedia Content in Personalized Environments Using the New Standards of the MPEG Family. Proceedings of IDPT2002 - 6th International Conference on Integrated Design and Process Technology. Pasadena, USA, June-2002. (Printed in USA).

(Santos Jr. et al., 2002b)

Santos Jr., J.B. dos; Moreira, E. S.; Goularte, R.; Faria, G. B.; Guglielmo, M.; De Petris, G.: Describing a Scenario and Technologies for Building Context-Aware Interactive Environments: Towards to Definition of New Interaction Criteria. **To appear in** Proceedings of CLEI'2002 - XXVIII Conferencia Internacional di Ingenieria Informatica. Montevideo, Uruguay, November-2002.

(Sauer & Engels, 1999)

Sauer, S. & Engels, G.: Extending UML for modelling of multimedia applications. In P. Mussio M. Hirakawa, editor, Proc. IEEE Symposium on Visual Languages (VL'99), Tokyo,

Japan, September-1999. [On-line]
<http://computer.org/proceedings/v1/0216/02160080abs.htm>

(Schmidt, 2000)

Schmidt, A. (2000): *Implicit Human Computer Interaction Through Context*. Personal Technologies Volume 4 (2&3), pp.191-199, June-2000.

(Schwabe, 1995)

Schwabe D.; Rossi G. *The Object -Oriented Hypermedia Design Model*, Communications ACM, Vol 38, August 1995.

(Shneiderman, 1998)

Shneiderman, B. (1998): *Design the User Interface: Strategies for Effective Human-computer Interaction*. Addison-Wesley, 3rd Edition, 1998.

(Sinfonia, 2001)

The Sinfonia Tools for MPEG-4 Standard. Multimedia Division of TILAB – Telecom Italia Lab. 2001. [On-line] <http://www.tilab.com>

(Soares et al., 1995)

Soares, L. F G.; Lemos, G.; Colches, S. *Redes de Computadores: das LANs, MANs e WANs às Redes ATM*. 2ª. Ed. Rio de Janeiro: Campus, 1995.

(Sowa, 2000)

Sowa, J. F. (2000): *Ontology, metadata and semiotics*. Springer-Verlag, Berlin, pp.55-81. 2000-June.

(SUN, 1999)

SUN (1999): *JAVA Tutorial*. Disponível on-line (URL): <http://www.sun.com>. Available on 1999.

(SUN, 2000)

SUN (2000): *JAVA Servlets Tutorial*. Disponível on-line (URL): <http://www.sun.com>. Available on 2000.

(SUN, 2001)

SUN (2001): *The Sun Microsystems Java TV API 1.0 specification*. [On Line] <http://java.sun.com/products/javatv/>. Available on 2001.

(Takahashi et al., 1997)

Takahashi, K; Liang, E. *Analysis and Design of Web-based Information Systems*, proceedings of the Sixth International World Wide Web Conference, 1997.

(Tanenbaun, 1996)

Tanembaun, A. S. (1996): *Distributed Operating Systems*. Prentice Hall, 1996.

(Tanenbaun, 1998)

Tanembaun, A. S. (1998): *Computer Networks*. Prentice-Hall, 1998.

(Tennenhouse & Wetherall, 1996)

Tennenhouse, D. L. and Wetherall, D. J. (1996): *Towards an Active Network Architecture*. Multimedia Computing and Networking (MMCN '96), San Jose, CA, 1996.

(Turine et al., 1997)

Turine, M., A. S.; Oliveira, M. C. F.; Masiero, P. C. *Hypertext Model Based on Statecharts*, Proceedings of the 8th ACM Conference on Hypertext (Hypertext'97), Southampton, UK, Abril 6-11, 1997, pp.102-111,1997.

(TVAF, 2000a)

TV Anytime Forum. – *Requirements Series:R1 : The TV-Anytime Environment (Informative)*– (Agosto de 2000) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Plenary/TV035r6.zip>

(TVAF, 2000b)

TV Anytime Forum. – *Requirements Series:R2 : The System Description (Informative)*– (Abril de 2000) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Plenary/TV036r2.zip>

(TVAF, 2000c)

TV Anytime Forum. – *Requirements Series:R3 : Metadata Requirements (Normative)* – (Abril de 2000) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Plenary/TV037r3.zip>

(TVAF, 2001a)

TV Anytime Forum. – *Specification Series:S1 on Phase 1 – Benchmark Applications (Informative)*– (Fevereiro de 2001) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Specifications/SP001v10.zip>

(TVAF, 2001b)

TV Anytime Forum. – *Specification Series:S2 on System Description (Informative)*– (Junho de 2001) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Specifications/SP002v11.zip>

(TVAF, 2001c)

TV Anytime Forum. – *Specification Series:S3 on Metadata (Normative)*– (Abril de 2001) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Specifications/SP003v10r1.zip>

(TVAF, 2001d)

TV Anytime Forum. – *Specification Series:S4 : Content Referencing (Normative)*– (Abril de 2001) –[On Line] – <ftp://tva:tva@ftp.bbc.co.uk/pub/Specifications/SP004v11.zip>

(Van Setten, 2001)

Van Setten, M. et al.: Gigaport: Personalized Information Systems. PhD Thesis. Telematica Institute. The Netherlands. June-2001.

(Van Setten et al., 2001)

Van Setten, M.; Tokmakoff, A.; van Vliet, H.: Designing Personalized Information Systems. Proceedings. First Workshop on Future Television at 8th International Conference on User Modelling (UM'2001). Sonthofen, Germany. July-2001. LNAI 2001 Springer Verlag, N1, V2 (Printed in Germany).

(Vickres, 2000)

Vickres, M; Zigmond, D- *Uniform Resource Identifiers for Television Broadcasts* (21 de Fevereiro de 2000).

(VRML, 1997)

The Virtual Reality Modelling Language (1997): ISO/IEC 14772:1997. [on-line] <http://www.vrml.org/technicalinfo/specifications/vrml97>.

(W3C, 1998)

W3C Working Draft - *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification* (15 de Junho de 1998) - [On Line] - <http://www.w3.org/TR/1998/REC-smil-19980615>

(W3C, 2000)

W3C Working Draft - *Document Object Model (DOM) Level 1 Specification (Second Edition) Versão 1.0* (29 de Dezembro de 2000) - [On Line] - <http://www.w3.org/TR/2000/WI-DOM-Level-1-20000929>

(W3C, 2001a)

W3C Recommendation - *XML Schema Part 0: Primer* (2 de Maio de 2001) - [On Line] - <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>

(W3C, 2001b)

W3C Recommendation - *XML Schema Part 1: Structure* (2 de Maio de 2001) - [On Line] - <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

(W3C, 2001c)

W3C Recommendation - *XML Schema Part 2: Data Types* (2 de Maio de 2001) - [On Line] - <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

(Wetherall et al., 1998)

Wetherall, J. et al (1998): *ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols*. Proc. IEEE OpenArch'98, San Francisco, CA, 1998.

(Wu et al., 1997)

Wu, L. et al. (1997): *Thinstreams: An Architecture for Multicasting Layered Video*. NOSSDAV'97, 1997.



Identification of the Interaction Criteria and Tools Suitable to Provide the most Efficient Use of Multimedia Educational Material and Implementation of a Telelearning Application over a Multicast/Broadcast Scenario ¹

Final Report of the Research Activities

Author: João Benedito dos Santos Junior
(*named Visiting Researcher in this document*)

Local: TILAB – Telecom Italia Laboratory

City: Turin, Italy

Division: Advanced Multimedia

Boss: Leonardo Chiariglione

Supervisor: Mario Guglielmo

Project Manager: Gianluca De Petris

Project: Telelearning System

Period of Research: June-1st to November-30th

Date of Writing: November-05th to 12th, 2001

The TILAB research proposal

The proposal for 06 (six) months of research activity at TILAB has been presented by Mario Guglielmo and Gianluca De Petris on April 2001, after a period of talking with João Benedito dos Santos Junior – the visiting researcher. This proposal has been elaborated also for contributing with João Benedito dos Santos Junior in his PhD Thesis at the Institute of Mathematics and Computing, at University of Sao Paulo, Brazil [1], as ilustraded in the **Table 01**.

The title of the research has been defined “Identification of the interaction criteria and tools suitable to provide the most efficient use of multimedia educational material and implementation of a telelearning application over a multicast/broadcast scenario”, having the “Distance Learning” like area of application.

The technologies available for using in the project were the following: a) MPEG-4 object coding; b) scene control; c) user interaction; d) content description and identification; e) transport of MPEG-4 over IP and MPEG-2 TS. Furthermore, the available results were the following: a) skills on MPEG-4 technologies (mainly object coding and Systems); b) knowledge about the architecture of a prototypical telelearning platform; c) software for integration with commercial systems to send lessons; d) software to receive and handle lessons from the satellite.

In general lines, the research description was the following:

TILAB is using MPEG-4 technology to build applications to be used in real environments [6]. One of these applications allows a teacher to send real-time lessons to this/her students or to record them. The Tele-learning system under development at TILAB includes:

- on the teacher side: a recording workstation with two cameras, microphone, specific MPEG-4 software;
- an IP network or an MPEG-2 TS satellite link;
- on the student side: a PC with special MPEG-4/2 software and special boards, or a Set Top Box.

The research work would focus on the broadcast scenario where a satellite board is used in a PC or on an MHP set top box. The aim of this work is to send the lesson even to a student that is not connected to the intranet, using a satellite link, either over IP embedded in the

MPEG-2 TS or directly over MPEG-2 TS. For the security part it may be necessary to have a low-band return channel implemented e.g. through a mobile phone.

The satellite environment may require the redesign of the User Interface and the retargeting of the elementary streams parameters in order to match specific requirements and features of the medium. The **Figure 01** shows the architecture of the Telelearning system and presents the scenario that is expected when this research has been started [5].

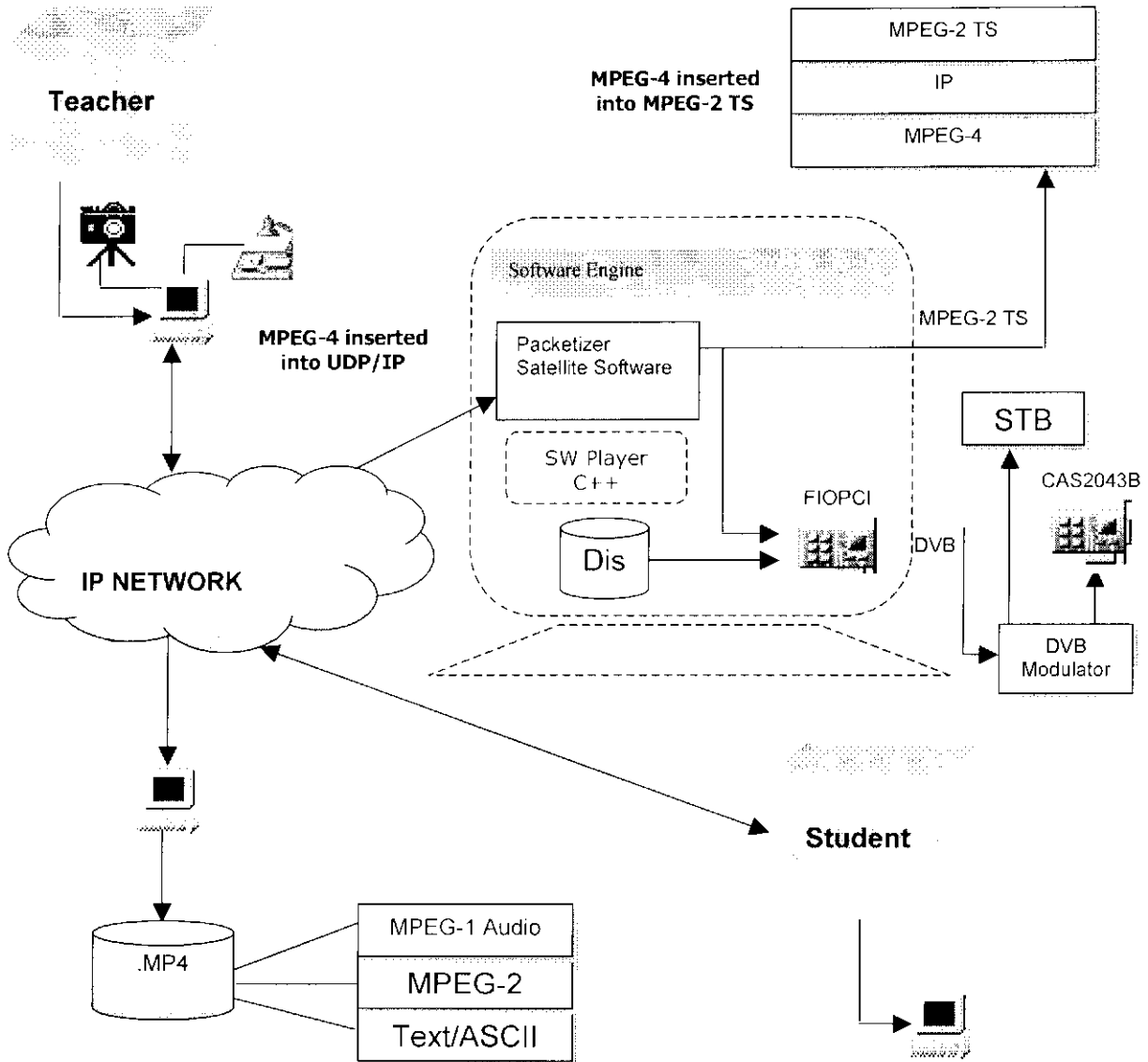


Figure 01: The Telelearning Architecture

The satellite environment may require the redesign of the User Interface and the retargeting of the elementary streams parameters in order to match specific requirements and features of the medium.

Table 01: Research Activities - 1st June-2001 to 30th November-2001

Month	Activity
June	<ol style="list-style-type: none"> 1. Study of the DVB, MPEG-2 and DSMCC specifications 2. Study of resources of the Microsoft Foundation Classes 3. Experiment of the resources of the Microsoft C++ Integrated Development Environment Version 6.0
July	<ol style="list-style-type: none"> 4. Implementation of a GUI for the TS File Player 5. Implementation of controller for TS File Player
August	<ol style="list-style-type: none"> 6. Study and definition of the new interaction criteria for Telelearning System Architecture 7. Implementation of the new criteria of interaction for the both teacher and student
September	<ol style="list-style-type: none"> 8. Implementation of the Teacher Chat and Student Chat applications 9. Definition of a XML/MPEG-7 model for integration of the chat applications with de MPEG-4 scene control 10. Study and implementation of the MSDP interpreter
October	<ol style="list-style-type: none"> 11. Study and implementation of the MUX IP Datagram Section for MPEG-4 streams over MPEG-2 Transport Stream - an extension to the DVB MUX basic application (it has been implemented in TILAB)
November	<ol style="list-style-type: none"> 12. Study and implementation of the some mechanisms for providing support to transmission of MPEG-4 streams over MPEG-2 Transport Stream using DVB standard. 13. Evaluation and Tests 14. Ending

The development of software applications

The activities of development of software applications have been divided into three phases: a) in the first phase, during July, it has been developed a player for MPEG-2 TS files; b) in the second phase, during August and September, it have been developed new interaction criteria and chat tools for the Telelearning scenario; c) in the last phase, during October and November, it have been implemented some mechanisms for providing support to transmission of MPEG-4 streams over MPEG-2 TS using DVB standard.

The additional contributions for TILAB researches

As described in the Section 3.3, when the teacher chat application is turned active, is also triggered a subprocess for recording all actions realized by teacher. This record is generated through an XML-compliant model, as show the schema of the Figure 02.

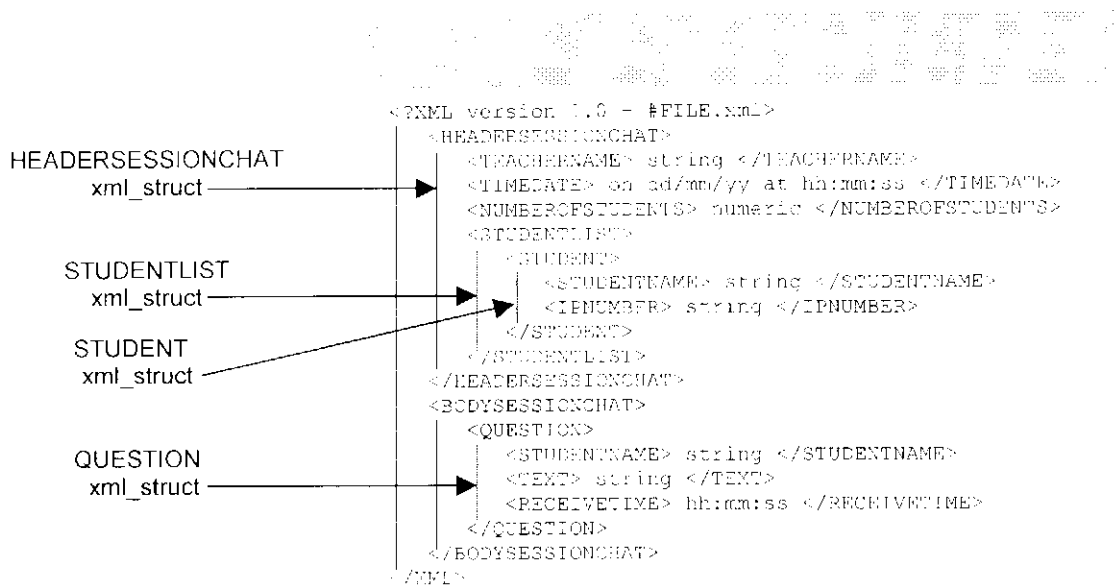


Figure 02: The XML-compliant model for integration of chat application at Telelearning architecture

At the level of implementation, the field <TIMEDATE> of the HEADERSSESSIONCHAT xml_struct can be divided into sub fields, allowing more granularity in the synchronization processes with the MPEG-4 elementary streams. For example, a suggestion is the specification of the following fields: <day>, <month>, <year>, <hour>, <minute> and <second>.

As described, TILAB is using MPEG-4 technology in the Telelearning project. In this context, the technologies available allowing the MPEG-4 object coding, the scene control and user interaction and transport of MPEG-4 over IP and MPEG-2 TS. This additional contribution presents a proposal for content description and personalization of the Telelearning environment.

Considering the log files of the Teacher Chat Application is possible to integrate the data with DBMS (*DataBase Management System*) and providing new controls for the both student

and teacher interfaces. The **Figures 03** and **04** present examples of XML files generated by chat application.

```

<?XML version 1.0 - #CMTR.xml>
  <HEADERSSESSIONCHAT>
    <TEACHERNAME>João</TEACHERNAME>
    <TIMEDATE>on 10/Oct/01 at 11:45:18</TIMEDATE>
    <NUMBEROFSTUDENTS>1</NUMBEROFSTUDENTS>
    <STUDENTLIST>
      <STUDENT>
        <STUDENTNAME>Gabriel</STUDENTNAME>
        <IPNUMBER>163.162.130.53</IPNUMBER>
      </STUDENT>
    </STUDENTLIST>
  </HEADERSSESSIONCHAT>
  <BODYSESSIONCHAT>
    <QUESTION>
      <STUDENTNAME>Gabriel</STUDENTNAME>
      <TEXT> Demanda 1 </TEXT>
      <RECEIVETIME>11:45:48</RECEIVETIME>
    </QUESTION>
  </BODYSESSIONCHAT>
</XML>

```

Figure 03: An example of XML file generated for Teacher Chat Application

```

<?XML version 1.0 - #CMTR.xml>
  <HEADERSSESSIONCHAT>
    <TEACHERNAME>João</TEACHERNAME>
    <TIMEDATE>on 10/Oct/01 at 11:45:18</TIMEDATE>
    <NUMBEROFSTUDENTS>1</NUMBEROFSTUDENTS>
    <STUDENTLIST>
      <STUDENT>
        <STUDENTNAME>Gabriel</STUDENTNAME>
        <IPNUMBER>163.162.130.53</IPNUMBER>
      </STUDENT>
    </STUDENTLIST>
  </HEADERSSESSIONCHAT>
  <BODYSESSIONCHAT>
    <REPLY>
      <STUDENTNAME>Gabriel</STUDENTNAME>
      <TEXT> Rispota 1 </TEXT>
      <SENDTIME>11:45:55</SENDTIME>
    </REPLY>
  </BODYSESSIONCHAT>
</XML>

```

Figure 04: An example of XML file generated for Teacher Chat Application

The **Figure 05** shows the basic graphic interface of the student side in the Telelearning project. Each graphic object is a representation of each MPEG-4 elementary stream. The chat application can be inserted as an external application, but triggered by a special control inserted into a MPEG-4 stream. In this context, a synchronization process can be implemented for providing advanced controller for the user.

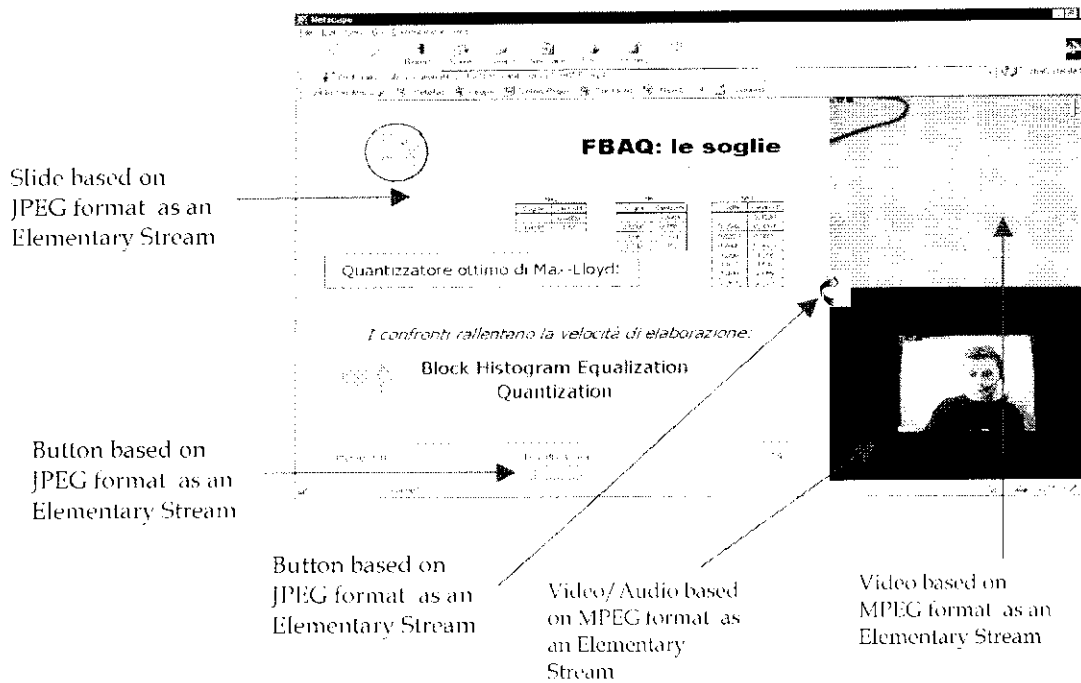


Figure 05: An example of XML file generated for Teacher Chat Application

As a concrete example, the XML structure, generated by chat teacher application, can be used for synchronizing the MPEG-4 objects with the questions that have been formulated by the students during the lesson. Thus, a publisher process can be used for presenting statistics about the lesson, considering the temporal relations described by chat time parameters and time stamp of the MPEG-4 objects, as illustrated in the Figure 06.

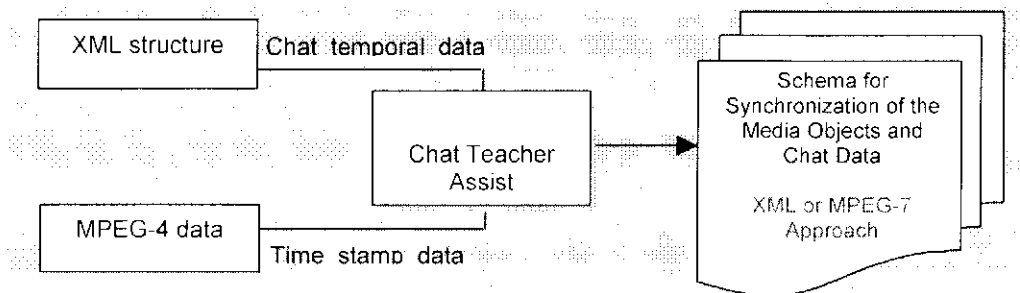


Figure 06: Basic scenario for integration of the new interaction criteria

Furthermore, MPEG-7 scene descriptions can be used for adding new personalization levels to the Telelearning environment [7]. This is one point that will be explored by the visiting researcher in his PhD Thesis and the results will be reported to TILAB [8].

The skills for the visiting researcher

TILAB – Telecom Italia Lab – is using MPEG-4 technology to build applications to be used in real environments. One of these applications allows a teacher to send real-time lessons to this/her students or to record them. The Tele-Learning system under development at TILAB includes:

The technologies available for using in the project were the following: a) MPEG-4 object coding; b) scene control; c) user interaction; d) content description and identification; e) transport of MPEG-4 over IP and MPEG-2 TS. Furthermore, the available results were the following: a) skills on MPEG-4 technologies (mainly object coding and Systems); b) knowledge about the architecture of a prototypical Telelearning platform; c) software for integration with commercial systems to send lessons; d) software to receive and handle lessons from the satellite.

João Benedito dos Santos Junior (joao@icmc.sc.usp.br) is a PhD Researcher, advising by Prof. Dr. Edson dos Santos Moreira (edson@icmc.sc.usp.br, PhD at Manchester University, UK), at Institute of Mathematics and Computing at University of Sao Paulo, Brazil.

In his PhD Thesis – “*Modelling of Context-Awareness Interactive Environments: An Approach Based on New Standards of the MPEG Family*”, started in 1998, he is working with MPEG-4 and MPEG-7 standards, as shows the following abstract [1]:

The possibility of using multimedia objects in modern applications has proven to be a bonus. We want to provide ways in which the user can interact with multimedia objects as they are presented with features that go beyond the regular controls of a VCR. Furthermore, the presentation of the video can be modified by events happening in the surrounding environment. This research presents a modelling for Context-Aware Interactive Television Applications as a proposal for producing, distributing and presenting multimedia at the user, considering the actions from contextual information, in a way to ease its interaction with context-aware interactive applications. MPEG-4 and MPEG-7 standards are being used for providing the technological support for producing multimedia content and media description, respectively. The work also discusses how and which kind of contextual information could be used in one ITV environments that have an active network as the underlying

infrastructure. The concept of profile applied to ITV is presented and methods for using profiles in ITV programs are discussed.

In this context, the period of six months at TILAB has provided to João the possibilities for to know better the MPEG-4 and MPEG-7 standards and to evaluate the application of these standards to his PhD Thesis. Furthermore, the visiting researcher has used the tools that compound the TILAB MPEG Solutions for carrying out experiments suitable to the purposes of his research work.

According to PhD Program Timetable of the visiting researcher, he must finish his Thesis in March-2002, reporting the main results at the TILAB, through Dr. Mario Guglielmo.

Perspectives and future works

TILAB - Telecom Italia Lab - is using MPEG-4 technology to build applications to be used in real environments. One of these applications allows a teacher to send real-time lessons to this/her students or to record them. The Tele-Learning system under development at TILAB includes:

- on the teacher side: a recording workstation with two cameras, microphone, specific MPEG-4 software;
- an IP network or an MPEG-2 TS satellite link;
- on the student side: a PC with special MPEG-4/2 software and special boards, or a Set Top Box.

Table 02: Timetable for the experiment

Activity	Period
Visit of TILAB Researchers in Brazil, specially to Catholic University of Minas Gerais (<i>where the author is lecturer and researcher in computing</i>)	September-2002
Providing of the TILAB-BR infrastructure	January-2003
Event for starting the experiment	First week of February-2003
Production of the Multimedia Content Compliant to Tele-Learning Platform	February and March-2003
Experiment	March, April and May-2003
Evaluation	June-2003

In this context, TILAB has interest on carry out an experiment in Brazil for evaluating the capabilities of the Tele-Learning Platform in the Brazilian Distance Learning Scenario. The Table 02 presents a suggested timetable for the experiment, starting on February-2003 and

ending on June-2003. Furthermore, the visiting researcher has submitted an abstract to IDPT'2002 - International Conference on Integrated Design and Process Technology - Session on Modelling and Development of Multimedia Systems (June 23-27, 2002, Pasadena, USA), as presented in the following.

Mario Guglielmo, Gianluca De Petris, Alessandro Costa Laia and Giovanni Cordara are co-authors of the follow abstract.

This abstract has been accepted and the authors must write the full paper by 15th January-2002. Thus, a specific timetable will be defined and the authors will work in the paper via Internet.

Using MPEG-4 and MPEG-7 for Production, Distribution and Presentation of Educational Multimedia on Personalized Environments

João Benedito dos Santos Junior¹, Gianluca De Petris², Mario Guglielmo²,
Edson dos Santos Moreira ¹, Rudinei Goularte ¹,
Alessandro Costa Laia² and Giovanni Cordara²
{joao, edson, rudinei}@icmc.sc.usp.br
{gianluca.depetris, mario.guglielmo, alessandro.costalaia, giovanni.cordara@tilab.com}

¹Institute of Mathematics and Computing - ICMC
University of Sao Paulo - USP
P.O.Box 668 - Sao Carlos - SP - Brazil
Phone: +55 16 273 9668 FAX: +55 16 273 9751

²TILAB - Telecom Italia Lab
Multimedia Division - Advanced Application Development
Via G. Reiss Romoli, 274
10148 - Turin - Italy
Phone: +39 011 228 6115 FAX: +39 011 228 6299

Abstract. TILAB is using MPEG-4 technology to build applications to be used in real environments. One of these applications allows a teacher to send real-time lessons to this/her students or to record them. The Tele-Learning System under development at TILAB includes: a) on the teacher side: a recording workstation with two cameras, microphone, specific MPEG-4 software; b) an IP network or an MPEG-2 TS satellite link; c) on the student side: a PC with special MPEG-4/2 software and special boards, or a Set Top Box. This research presents a focus on the broadcast scenario where a satellite board is used in a PC or on an MHP set top box. The aim of this work is to send the lesson even to a student that is not connected to the intranet, using a satellite link, either over IP embedded in the MPEG-2 TS or directly over MPEG-2 TS. For the security part it may be necessary to have a low-band return channel implemented, for example, through a mobile phone. The satellite environment may require the redesign of the User Interface and the retargeting of the elementary streams parameters in order to match specific requirements and features of the medium. At this point, new interaction criteria have been established from distribution of MPEG-4 media objects and MPEG-7 scene descriptions on network environments. Furthermore, context-awareness aspects are being added for providing personalization on the teaching-learning environment.

Conclusions

MPEG-4 is a standard that builds on the successes of MPEG-1 and MPEG-2, two standards that have changed the audio-visual landscape. MPEG-4 has been developed by the best experts provided by all industries with a stake in multimedia.

It is a powerful standard, rich in functionality, encompassing other successful standards. It can be customized to serve the needs of specific industries while preserving a high level of interoperability across applications of different industries.

It is a living standard in the sense that new features are being added preserving compatibility with existing applications to respond to the demands of the application world and supported by the expanding possibilities of technology.

In this context, the period of six months at TILAB has provided to visiting researcher the possibilities for knowing better the MPEG-4 and MPEG-7 standards and to evaluate the application of these standards to his PhD Thesis. Furthermore, the visiting researcher has used the tools that compound the TILAB MPEG Solutions for carrying out experiments suitable to the purposes of his research work. At this point, the visiting researcher is able to help TILAB for providing technological solutions based on MPEG-4 in the both brazilian academic and industry market.

Acknowledgments

I would like to thank Leonardo Chiariglione for allowing the carrying of part of my PhD research at TILAB.

I am very thankful to Mario Guglielmo by the organization of the visit, advising of the research and by the friendship. Thanks also to Dr. Edson dos Santos Moreira (USP, Brazil) for his supervision and for the freedom he allowed me in conducting my research.

I also would like to thank Gianluca De Petris for his technical advising during the research activities and for providing the sufficient environment for experiments. Special thanks to Alessandro Costa Laia and Giovanni Cordara for the both technical support and friendship, what became more pleasant my stay in Turin.

I would like to thank my colleagues at the Multimedia Division of TILAB for providing a very pleasant integration and excellent work environment, especially to Barbara Silano, Diego Gibellino, Filippo Chiariglione, Barbara Negro, Stefano Dal Lago and Tiziana Trucco. Finally, I would like to express my gratitude to TILAB for the financial support for this work.

References

- [1] Santos Jr., J. B. dos; Goularte, R.; Faria, G. B.; Moreira, E. S.: Modelling of user interac
- [2] Digital Video Broadcasting (DVB). DVB Specification for Data Broadcasting. EN 301 192. 1999.
- [3] Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 6: Extensions for DSM-CC. ISO/IEC 13818-6.
- [4] Information Technology – Generic Coding of Moving Pictures and Associated Audio. MPEG-2 Systems. 1994.
- [5] The Telelearning Architecture. TILAB reports.
- [6] Information Technology – Coding of Audio-visual Objects. MPEG-4 Systems. 2000.
- [7] Overview of the MPEG-7 Standard (version 5.0). 2001.
- [8] Santos Jr., J. B. dos: Modelling of Context-Aware Interactive Environments: An Approach Based on New Standards of the MPEG Family. PhD Thesis at the Institute of Mathematics and Computing at University of Sao Paulo, Brazil.



Artigos Publicados

Paper 01 - Proceedings of International Conference (July, 2001)

UM'2001 - 8th International Conference on User Modelling

1st Workshop on Personalization in Future TV

Santos Jr., J.B. dos; Goularte, R.; Faria, G.; Moreira, E.S.: Modelling of User Interaction in Context-Aware Interactive Television Application on Distributed Environments. Proceedings. First Workshop on Future Television at 8th International Conference on User Modelling (UM'2001). Sonthofen, Germany. July-2001. LNAI 2001 Springer Verlag, N1, V2, pp.49-56. (Printed in Germany).

Disponível on-line: <http://www.di.unito.it/~liliana/UM2001/>

Paper 02 - Anais de Congresso no Brasil (Outubro, 2001)

SBMidia'2001 - Simpósio Brasileiro de Sistemas Multimídia e Hipermídia

Blengini, G. F., Junior, J. B. S., Goularte, R., Moreira, E. S.. Uso de Perfis em Aplicações de Televisão Interativa Conscientes de Contexto. Anais do SBMidia 2001 - VII Simpósio Brasileiro de Sistemas Multimídia e Hipermídia, pp. 139-154. Florianópolis - SC, Brasil, outubro de 2001.(Impresso no Brasil - Eds: Roberto Willrich e Cesar Augusto Camilo Teixeira).

Paper 03 - International Journal (December, 2001)

SDPS Journal - Society for Design and Process Science

Santos Jr., J.B. dos; Goularte, R.; Moreira, E. S.; Faria, G. B.:The Modelling of Structured Context-Aware Interactive Environments. SDPS Journal. Transactions of the SDPS. Issue 5, Number 4, pp. 77-93. December-2001. (Printed in USA).

Disponível on-line: <http://www.sdpsnet.org/vol5-4.htm>

Paper 04 - Proceedings of International Conference (June, 2002)

IDPT'2002 - International Conference on Integrated Design and Process Technology

Santos Jr., J.B. dos; Guglielmo, M.; De Petris, G.; Moreira, E. S.; Goularte, R.; Faria, G. B.; Cordara, G.; Laia, A. C.: Production, Distribution and Presentation of Educational Multimedia Content in Personalized Environments Using the New Standards of the MPEG Family. Proceedings of IDPT'2002 - 6th International Conference on Integrated Design and Process Technology. Pasadena, USA, June-2002. (Printed in USA).

Paper 05 - Proceedings of International Conference (November, 2002)

CLEI'2002 - XXVIII Conferencia Internacional di Ingeneria Informatica

Santos Jr., J.B. dos; Moreira, E. S.; Goularte, R.; Faria, G. B.; Guglielmo, M.; De Petris, G.: Describing a Scenario and Technologies for Building Context-Aware Interactive Environments: Towards to Definition of New Interaction Criteria. **To appear in** Proceedings of CLEI'2002 - XXVIII Conferencia Internacional di Ingeneria Informatica. Montevideo, Uruguay, November-2002.

Proceedings of International Conference (July, 2001)

UM'2001 - 8th International Conference on User Modelling

1st Workshop on Personalization in Future TV

Santos Jr., J.B. dos; Goularte, R.; Faria, G.; Moreira, E.S.: Modelling of User Interaction in Context-Aware Interactive Television Application on Distributed Environments. Proceedings. First Workshop on Future Television at 8thInternational Conference on User Modelling (UM'2001). Sonthofen, Germany. July-2001. LNAI 2001 Springer Verlag, N1, V2, pp.49-56. (Printed in Germany).

Disponível on-line: <http://www.di.unito.it/~liliana/UM2001/>

Modelling of User Interaction in Context-Aware Interactive Television Application on Distributed Environments

**João B. dos Santos Jr, Rudinei Goularte,
Gustavo B. Faria and Edson dos S. Moreira**
{joao, rudinei, gusfaria, edson}@icmc.sc.usp.br
Institute of Mathematics and Computing - ICMC
University of Sao Paulo - USP
13566-570, São Carlos-SP, Brazil, PO Box 668

Abstract. The possibility of using multimedia objects in modern applications has proven to be a bonus. We want to provide ways in which the user can interact with multimedia objects while it is been presented with features that go beyond the regular controls of a VCR. Furthermore, the presentation of the video can be modified by events happening in the surrounding environment. This paper presents a proposal to model the user actions from contextual information, in a way to ease its interaction with context-aware interactive applications like Interactive TV (ITV). The paper also discusses how and which kind of contextual information could be used in one ITV environments that have an active network as the underlying infrastructure.

Anais de Congresso no Brasil (Outubro, 2001)

SBMídia'2001 - Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia

Blengini, G. F., Junior, J. B. S., Goularte, R., Moreira, E. S.. Uso de Perfis em Aplicações de Televisão Interativa Conscientes de Contexto. Anais do SBMídia 2001 - VII Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, pp. 139-154. Florianópolis - SC, Brasil, outubro de 2001.(Impresso no Brasil - Eds: Roberto Willrich e Cesar Augusto Camilo Teixeira).

**Uso de Perfis em Aplicações de Televisão Interativa
Conscientes de Contexto**

**Gustavo Blengini Faria, João Benedito dos Santos Junior,
Rudinei Goularte e Edson dos Santos Moreira**
{gusfaria, joao, rudinei, edson}@icmc.sc.usp.br
Universidade de São Paulo - USP
Instituto de Ciências Matemáticas e de Computação - ICMC
13566-570, São Carlos, Brasil, Cx. Postal 668

Resumo. A TV Interativa é uma aplicação multimídia com características peculiares, tais como interações em um ambiente anteriormente não interativo. Diversas tecnologias podem ser utilizadas em sua implementação e, devido ao crescente interesse da indústria, diversos esforços são feitos para sua padronização. Este artigo apresenta um novo conceito para programas interativos que provê uma melhor adaptação dos programas às necessidades ou expectativas dos usuários, utilizando os padrões e tecnologias para a TV Interativa e estendendo-os para o uso em ambientes conscientes de contexto.

Abstract. The Interactive TV is a multimedia application with peculiar characteristics like interactions in an environment where interactions were not possible previously. A number of technologies can be used on its implementation and there are some efforts to build standards. This paper presents a new concept to interactive programs that provides a better adaptation of the programs to the viewers using Interactive TV standards and technologies and extending them to the use on context aware environments.

Palavras-chave. Televisão Interativa, Interação do Usuário, Consciência de Contexto. MPEG-4, MPEG-7.

Áreas de Adequação. Interação do usuário com aplicações multimídia e hiperemídia. Ferramentas, ambientes e infra-estrutura para sistemas multimídia e hiperemídia.

International Journal (December, 2001)

SDPS Journal – Society for Design and Process Science

Santos Jr., J.B. dos; Goularte, R.; Moreira, E. S.; Faria, G. B.:The Modelling of Structured Context-Aware Interactive Environments. SDPS Journal. Transactions of the SDPS. Issue 5, Number 4, pp. 77-93. December-2001. (Printed in USA).

Disponível on-line: <http://www.sdpsnet.org/vol5-4.htm>

The Modelling of Structured Context-Aware Interactive Environments

**João Benedito dos Santos Junior, Rudinei Goularte,
Edson dos Santos Moreira and Gustavo Blengini Faria**

Institute of Mathematics and Computing
University of São Paulo – USP
P.O.BOX 668 – São Carlos – SP – Brazil
{joao, rudinei, edson, gusfaria}@icmc.sc.usp.br

Abstract. Using multimedia video objects in building rich, interactive and distributed environments is complex. We want to provide ways in which the user can interact freely with those objects. As a result, the quality of the applications is not only ruled by the presentation machine and its neighborhood but by the traffic conditions and the status of the server as well. As a good practice of design, modelling is essential, as well as the observation of standards. This paper presents a proposal for modelling Structured Context-Aware Interactive Environments, as an approach for covering the user-network-application interactions that occur in an interactive environment. In this scenario, the work discusses mainly which type of contextual information (both at the terminal and the network) could be used in an interactive environment and how it should be manipulated. The concept of program profile is also presented and methods for using profiles in interactive video are discussed. Interactive Television (ITV) programs are presented as an example of application that can use our modelling.

Paper 04 - Proceedings of International Conference (June, 2002)

IDPT'2002 - International Conference on Integrated Design and Process Technology

Santos Jr., J.B. dos; Guglielmo, M.; De Petris, G.; Moreira, E. S.; Goularte, R.; Faria, G. B.; Cordara, G.; Laia, A. C.: Production, Distribution and Presentation of Educational Multimedia Content in Personalized Environments Using the New Standards of the MPEG Family. Proceedings of IDPT'2002 - 6th International Conference on Integrated Design and Process Technology. Pasadena, USA, June-2002. (Printed in USA).

Production, Distribution and Presentation of Educational Multimedia Content in Personalized Environments Using the New Standards of the MPEG Family

**João Benedito dos Santos Junior¹, Mario Guglielmo², Gianluca De Petris²,
Edson dos Santos Moreira³, Rudinei Goularte³, Gustavo Blengini Faria³,
Alessandro Costa Laia², Giovanni Cordara²**

{joao, edson, rudinei, gusfaria}@icmc.sc.usp.br

{mario.guglielmo, gianluca.depétris, alessandro.costalaia, giovanni.cordara}@tilab.com

^{1,3} University of Sao Paulo - USP
Institute of Mathematics and Computing - ICMC
P.O.Box 668 - Sao Carlos - SP - Brazil
Phone: +55 16 273 9668 FAX: +55 16 273 9751

² TILAB - Telecom Italia Lab
Multimedia Division - Advanced Application Development
Via G. Reiss Romoli, 274 - 10148 - Turin - Italy
Phone: +39 011 228 6115 FAX: +39 011 228 6299

Abstract. TILAB is using MPEG-4 technology to build applications to be used in real environments. One of these applications allows a teacher to send real-time lessons to this/her students or to record them. The Tele-Learning System under development includes: a) on the teacher side: a recording workstation with two cameras, microphone, specific MPEG-4 software; b) an IP network or an MPEG-2 TS satellite link; c) on the student side: a PC with special MPEG software, and a special board if receiving from satellite. This research focuses on the broadcast scenario where a satellite board is used in a PC. Thus, the work covers how to send the lesson even to a student that is not connected to the intranet, using a satellite link, either over IP embedded in the MPEG-2 TS or directly over MPEG-2 TS. For the security part it may be necessary to have a low-band return channel implemented, for example, through a mobile phone. The satellite environment may require the redesign of the User Interface and the retargeting of the elementary streams parameters in order to match specific requirements and features of the medium. At this point, new interaction criteria have been established from distribution of MPEG-4 media objects and MPEG-7 scene descriptions on network environments. Furthermore, context-awareness aspects are being added for providing personalization on the teaching-learning environment and MPEG-21 is being studied for applying to new multimedia requirements.

Paper 05 - Proceedings of International Conference (November, 2002)

CLEI'2002 - XXVIII Conferencia Internacional di Ingeneria Informatica

Santos Jr., J.B. dos; Moreira, E. S.; Goularte, R.; Faria, G. B.; Guglielmo, M.; De Petris, G.: Describing a Scenario and Technologies for Building Context-Aware Interactive Environments: Towards to Definition of New Interaction Criteria. **To appear in** Proceedings of CLEI'2002 - XXVIII Conferencia Internacional di Ingeneria Informatica. Montevideo, Uruguay, November-2002.

Describing a Scenario and Technologies for Building Context-Aware Interactive Environments: Towards to Definition of New Interaction Criteria

João Benedito dos Santos Junior¹

Catholic University of Minas Gerais - PUC Minas

Department of Computer Science

Poços de Caldas - MG - Brazil

Phone (voice): +55 35 3697 3027 FAX: +55 35 3697 3001

joao@pucpcaldas.br

Edson dos Santos Moreira², Rudinei Goularte², Gustavo Blengini Faria²

University of São Paulo - USP

Institute of Mathematics and Computing - ICMC

P.O.Box 668 - São Carlos - SP - Brazil

Phone (voice): +55 16 273 9668 FAX: +55 16 273 9751

Mario Guglielmo³, Gianluca De Petris³

TILAB - Telecom Italia Lab

Multimedia Division - Advanced Application Development

Via Guglielmo Reiss Romoli, 274 - Turin - Italy

Phone (voice): +39 011 228 6115 FAX: +39 011 228 6299

Abstract

As the interactive video technology has been disseminated, new forms of interaction became possible. A number of new techniques and standards have been produced in order to allow for the development of new applications; however, some potential new interaction forms are still unexplored. This paper describes a scenario for interactive applications, based on the emerging MPEG-4 and MPEG-7 standards, that uses the context-awareness concepts in order to provide the necessary support to the development of more interactive and useful systems. In this scenario, the environment may require the redesign of the user interface and the retargeting of the elementary streams parameters in order to match specific requirements and features of the environment (application, media, network, terminal). At this point, new interaction criteria can be established from distribution of MPEG-4 media objects and MPEG-7 scene descriptions on network environments. Context-awareness aspects are being added for providing personalization on the interactive environment.

Keywords: Human-computer Interaction; Human-environment Interaction; Ubiquitous Computing; Context-awareness; Distributed Interactive Applications; Multimedia Standards.