

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

AVANTV: UMA ABORDAGEM PARA PERSONALIZAÇÃO DO
CONTEÚDO DE APLICAÇÕES DE TV DIGITAL INTERATIVA
SENSÍVEL AO CONTEXTO

FABIANA FERREIRA DO NASCIMENTO

João Pessoa, Paraíba

Agosto – 2011

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

FABIANA FERREIRA DO NASCIMENTO

**AVANTV: UMA ABORDAGEM PARA PERSONALIZAÇÃO DO
CONTEÚDO DE APLICAÇÕES DE TV DIGITAL INTERATIVA
SENSÍVEL AO CONTEXTO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Ed Porto Bezerra

João Pessoa, Paraíba

Agosto – 2011

N244a Nascimento, Fabiana Ferreira do.

AvanTV: Uma Abordagem para Personalização do Conteúdo de Aplicações de TV Digital Interativa Sensível ao Contexto / Fabiana Ferreira do Nascimento. – João Pessoa, 2011.
124 f. : il.

Dissertação (Mestrado) – UFPB/CCEN.
Orientador: Ed Porto Bezerra.

1. TV Digital Interativa. 2. Personalização de Conteúdo. 3. Modelagem de Contexto. 4. Aplicações Sensíveis ao Contexto.

CDU – 004.03(043)

Catálogo bibliográfica: Rejane Medeiros Borges (CRB 15)

AGRADECIMENTOS

Agradeço, antes de tudo, ao Pai Criador que me trouxe iluminação nos momentos difíceis e caminhou comigo ao longo de toda a jornada.

Agradeço aos companheiros do NTI que me estimularam na realização do mestrado. Desde o consentimento da ex-diretora, professora Ismênia Mangueira, e do atual, prof. Sérgio Albuquerque, até o apoio essencial de todos os colegas, em especial, a Clístenes e Creusa.

Agradeço ao meu orientador Ed Porto, que contribuiu nas direções a serem seguidas neste trabalho e pela paciência durante período de desânimo.

Muito obrigada pela colaboração: dos professores do PPGI, sobretudo à professora Tatiana; ao professor Carlos Ferraz do Cin/UFPE pelas importantes sugestões; aos colegas do LAVID pelas trocas de conhecimento, principalmente Erisvaldo, Álan e Júlio; e àqueles que cooperaram direta ou indiretamente com palavras de incentivo e conforto, assim como a Anne, Samara, Viviane, Tiago, Ricardo, Jonatas, Ramos, aos bibliotecários, principalmente Rejane, e tantos outros.

Agradeço aos meus pais, Hamilton e Gracinha (*in memoriam*), por sempre acreditarem nos meus ideais e me ensinarem que todo trabalho é concretizado com perseverança. E ainda aos meus queridos irmãos Fabinho e Flavinho, cunhadas e familiares pela força e carinho incondicional.

Agradeço aos meus anjos da guarda Mário, companheiro de todas as horas, e Yasmin, filha de coração, por tanta generosidade e cumplicidade. A vida tem outro sentido ao lado de vocês.

Graças a todos vocês e outros que porventura eu possa ter esquecido, que com gestos simples e palavras amigas amenizaram tantas aflições, foi possível vencer este desafio. Muitíssimo obrigada!

RESUMO

Graças ao modo peculiar de interatividade realizada por aplicativos na TV, são necessários mecanismos que possibilitem recuperar informações além daquelas fornecidas diretamente. Neste sentido, aplicações sensíveis ao contexto utilizam informações consideradas relevantes para fornecer suporte à realização de tarefas. Desenvolver aplicações desta natureza apresenta desafios quanto a captura de dados a partir de diferentes fontes, tais como sensores; quanto a representação mais adequada para realizar comportamento sensível ao contexto; e capacitar a inferência de conhecimentos. Este trabalho propõe uma abordagem para personalização do conteúdo de aplicações de TV Digital Interativa através da manipulação de informações de contexto. Para tanto, foi realizada a especificação de um modelo contextual que descreve semântica de informações do usuário e de conteúdo esportivo de maneira integrada, e foram desenvolvidos serviços cujas funcionalidades oferecem suporte ao uso de contexto.

Palavras-chave: TV Digital Interativa. Personalização de Conteúdo. Modelagem de Contexto. Aplicações Sensíveis ao Contexto. Análise Semântica.

ABSTRACT

Due to particular interactivity mode performed by TV services, engines are required that allow to retrieve informations beyond those provided directly. In this sense, context-aware applications use relevant informations to provide support in tasks execution. To develop these kind of applications presents challenges in context capture from heterogeneous sources, as sensors; by representation more adjusted to perform context-aware behavior; and to enable infer knowledges. This dissertation proposes an approach for content personalization of Interactive TV Applications by context handling. To this end, a context modelling was achieved to describe the user information and sports content information semantic in an integrated way and services were developed whose features provides support to context usage.

Keywords: *Interactive Digital TV. Content Personalization. Context Modeling. Context-Aware Applications. Semantic Analysis.*

SUMÁRIO

CAPÍTULO I. INTRODUÇÃO	1
1.1 MOTIVAÇÃO	2
1.2 OBJETIVOS.....	3
1.3 ESTRUTURA DA DISSERTAÇÃO.....	4
CAPÍTULO II. FUNDAMENTAÇÃO TEÓRICA	5
2.1 REPRESENTAÇÃO DO CONHECIMENTO	5
2.2 TELEVISÃO DIGITAL INTERATIVA.....	11
2.2.1 ARQUITETURA E PADRÕES	12
2.2.2 SISTEMA BRASILEIRO DE TV DIGITAL	13
2.2.3 INTERATIVIDADE NA TV.....	16
2.2.4 METADADOS NA TVDI.....	21
A) MPEG-7.....	21
B) TV-ANYTIME	22
C) MPEG-21.....	23
2.3 SISTEMAS SENSÍVEIS AO CONTEXTO	24
2.3.1 CONCEITOS BÁSICOS	25
2.3.2 ESTRATÉGIAS DE MODELAGEM DE CONTEXTO	26
2.3.3 CONSTRUÇÃO DE APLICAÇÕES SENSÍVEIS AO CONTEXTO	28
2.3.4 EXEMPLOS DE APLICAÇÕES SENSÍVEIS AO CONTEXTO	31
2.4 ADAPTAÇÃO DO CONTEÚDO	32
2.5 CONSIDERAÇÕES FINAIS	36
CAPÍTULO III. TRABALHOS RELACIONADOS	38
3.1 PROFILETV	38
3.2 COLLABORATVWARE	40
3.3 FRAMEWORK PARA CUSTOMIZAÇÃO DE VÍDEOS ESPORTIVOS PERSONALIZADOS.....	41
3.4 PERSONALTVWARE	43
3.5 ANÁLISE COMPARATIVA	44
3.6 CONSIDERAÇÕES FINAIS	46
CAPÍTULO IV. INTEGRANDO MODELO CONTEXTUAL E CONTEÚDO SEMÂNTICO	48
4.1 INTRODUÇÃO	48
4.2 MODELO DE DOMÍNIO	49
4.2.1 ESCOPO.....	50
4.2.2 ONTOLOGIA <i>SPTC SEMANTIC</i>	51
4.3 MODELO DE USUÁRIO	55
4.3.1 REPRESENTAÇÃO DO MU POR METADADOS MPEG-7	56
4.4 MODELO DE ADAPTAÇÃO	59
4.5 PROCEDIMENTOS DE INFERÊNCIA	65
4.6 AVALIAÇÃO DAS ONTOLOGIAS.....	69
4.7 CONSIDERAÇÕES FINAIS	70

<u>CAPÍTULO V. PROTÓTIPO AVANTV: APLICAÇÕES DE TVDI SENSÍVEIS AO CONTEXTO</u>	<u>72</u>
5.1 IMPLANTAÇÃO.....	72
5.2 INTERFACE COM APLICAÇÕES.....	76
5.3 MÓDULO DE AQUISIÇÃO	78
5.4 MÓDULO DE GERENCIAMENTO.....	80
5.5 MÓDULO DE RACIOCÍNIO	83
5.6 IMPLEMENTAÇÃO DO PROTÓTIPO AVANTV	87
5.6.1 FILTRAGEM BASEADA EM CONTEÚDO UTILIZANDO ONTOLOGIAS	92
5.7 EXEMPLOS DE USO	94
5.7.1 AGREGADOR DE NOTÍCIAS	95
5.7.2 LISTAGEM DE EVENTOS ESPORTIVOS	99
5.8 CONSIDERAÇÕES FINAIS	101
 <u>CAPÍTULO VI. CONCLUSÃO</u>	 <u>102</u>
6.1 PRINCIPAIS CONTRIBUIÇÕES.....	102
6.2 TRABALHOS FUTUROS.....	103
 <u>REFERÊNCIAS.....</u>	 <u>105</u>

LISTA DE FIGURAS

Figura 2.1: Representação em grafo de declaração RDF	7
Figura 2.2: Descrição da classe Competition da ontologia de esportes na ferramenta de edição Protégé-OWL 4.	11
Figura 2.3: Visão geral do sistema de difusão	12
Figura 2.4: Arquitetura Ginga (Fonte: GINGANCL, 2007).....	14
Figura 2.5: Estrutura básica de um Xlet Java.....	16
Figura 2.6: Programas interativos em outros meios	18
Figura 2.7: Exemplos de interatividade local	19
Figura 2.8: Exemplo de um guia eletrônico de programação (EPG).....	19
Figura 2.9: Aplicações interativas	20
Figura 3.1: Definição contextual dos serviços interativos testados no ProfileTV (Fonte: COELHO, 2008)	39
Figura 3.2: Arquitetura do CollaboraTVware. (Fonte: ALVES, 2008).....	41
Figura 3.3: Framework para customização de vídeos esportivos personalizados (Fonte: LIANG et al., 2009).....	42
Figura 3.4: Infraestrutura PersonalTVware (Fonte: SILVA et al., 2009).....	44
Figura 4.1: Representação do Conteúdo Esportivo (SPTC)	52
Figura 4.2: Descrição de serviços esportivos no modelo SPTC	53
Figura 4.3: Ontologia de esportes (SPTC Semantic)	55

Figura 4.4: Elementos do UserDescriptionType com restrições de complexidade sobre o padrão MPEG-7	57
Figura 4.5: UserPreferences DS simplificado no MPEG-7	58
Figura 4.6: UsageHistory DS no MPEG-7	58
Figura 4.7: Ontologia de Adaptação Contextual AvantvCTX.....	60
Figura 4.8: Definição das propriedades de dados hasContextType e hasAcquisitionType	61
Figura 5.1: Casos de uso da AvanTV	73
Figura 5.2: Ambiente de execução Ginga-J estendido.....	75
Figura 5.3: Arquitetura da AvanTV	76
Figura 5.4: Diagrama de interação entre aplicação e serviços AvanTV	78
Figura 5.5: Classes do Módulo de Aquisição	79
Figura 5.6: Classes do Módulo de Gerenciamento dos Modelos	80
Figura 5.7: Fragmento de documento SPTC.....	81
Figura 5.8: Instâncias do modelo de usuário	82
Figura 5.9: Classes do Módulo de Raciocínio	83
Figura 5.10: Diagrama de classes dos objetos de controle de requisições	85
Figura 5.11: Ontologia da classe Application	86
Figura 5.12: Descrição XML de Serviço de Informação.....	88
Figura 5.13: Diagrama de seqüência para requisições ao Reason	91
Figura 5.14: Processamento de adaptação de conteúdo	94
Figura 5.15: Aplicação de Notícias com opção para usuário não registrado.....	97

Figura 5.16: Aplicação de notícias com opções para usuários registrados	97
Figura 5.17: Aplicação de notícias com conteúdo personalizado	98
Figura 5.18: Listagem de eventos esportivos com realce de conteúdo	100
Figura 5.19: Listagem de eventos esportivos com anúncio personalizado	101

LISTA DE TABELAS

Tabela 2.1: Resumo das estratégias de modelagem de contexto	28
Tabela 3.1: Comparação entre sistemas analisados	45
Tabela 4.1: Dados utilizadas no Modelo de Usuário	56
Tabela 4.2: Representação da Classe User	62
Tabela 4.3: Representação da Classe Content	63
Tabela 4.4: Representação da Classe Activity	64
Tabela 4.5: Características das ontologias relacionadas ao modelo de adaptação contextual AvantvCTX.....	69

LISTA DE ABREVIATURAS

API	Application Programming Interface
BNF	Backus-Naur Form
CRID	Content Reference Identifier
DI	Digital Item
EPG	Electronic Programming Guide
MDS	Multimedia Description Schemes
MPEG	Moving Picture Experts Group
OWL	Web Ontology Language
PDR	Personal Digital Recorder
PSI	Program-Specific Information
RDF	Resource Description Framework
SCC	Sistema Ciente de Contexto
SSC	Sistema Sensível ao Contexto
STB	Set –Top Box
TVDI	Televisão Digital Interativa
UDP	User Description Profile
URI	Uniform Resource Identifier
XML	Extensible Markup Language
UML	Unified Modeling Language

Capítulo 1.

INTRODUÇÃO

A ssistir televisão não mais corresponde a uma prática passiva no que se refere às possibilidades de interação do telespectador. Além da melhoria na qualidade de imagem e áudio, na TV Digital Interativa (TVDI), o telespectador assume o papel de um usuário computacional, capaz de obter mais informações além da programação de canais e realizar diversas ações que aprimorem sua experiência televisiva. No entanto, a maioria das aplicações de TVDI é projetada sem considerar as particularidades da audiência e segue a estrutura de navegação genérica. Uma solução para este problema é o emprego de mecanismos que permitam fornecer conteúdo relevante de forma particular ao usuário.

Em geral, sistemas interativos tendem a ser personalizados de diferentes maneiras tais como o ajuste da interface do usuário quanto à cor, fonte e posição de itens (apresentação) ou quanto à produção de conteúdos distintos para cada usuário (navegação). Ambos requerem algum conhecimento acerca dos elementos que caracterizam o usuário, determinado pelo domínio do sistema como um todo. Apesar de parecer contraditório o fato de endereçar conteúdo de gênero pessoal num meio de difusão (*broadcast*), este é um tema explorado em várias pesquisas.

Modelos de usuários são resultantes de informações de perfis e configurações fornecidas pelo usuário ou extraídas pelo sistema a partir da análise de suas interações e do ambiente em que estão inseridos. A combinação de descritores do modelo de usuários e do conteúdo possibilita oferecer uma TV personalizada. Alguns exemplos abrangem recomendações baseadas em guias eletrônicos de programação (*Electronic Program Guides – EPG*) (ARDISSONO *et al.*, 2004; BUCZAK *et al.*, 2002), em seleção de anúncios interativos dirigidos (THAWANI *et al.*, 2004) e em adaptação do conteúdo que enfatiza a programação (GOREN-BAR e GLINANSKY, 2002; GOULARTE, 2003).

Estudos têm aplicado contexto na assimilação do que é importante ao usuário para construir sistemas adaptáveis. O contexto de um usuário é um conceito amplo que pode incluir, dentre outros, conhecimento sobre localização, tempo, dispositivo, ambiente físico e social. De maneira geral, é definido como qualquer informação empregada para caracterizar a situação de uma entidade (DEY *et al.*, 1999). Um sistema é dito Ciente de Contexto (SCC) ou Sensível ao Contexto (SSC) quando utiliza estas informações para mudar seu comportamento conforme o interesse dos envolvidos numa tarefa. No entanto, questões relacionadas à utilização de contexto por aplicações em sistemas televisivos ainda são desafiadoras.

1.1 Motivação

As reações à programação, o compartilhamento do que está sendo exibido ou a busca por informações adicionais podem ser realizadas através do próprio receptor (*set-top box* ou terminal de acesso) da TVDI. Dentre os tipos básicos de serviços fornecidos, a forma mais comum ocorre pela inclusão de conteúdo sobreposto aos programas. Seja pela interatividade local ou através de canal de comunicação, o conteúdo suplementar reforça o modelo de busca da informação pelo usuário, contraposto ao comportamento habitual caracterizado por uma atitude inerte (*lean back*) de consumo de mídia. Exemplos desta ênfase na TV incluem a navegação por tabelas de jogos e estatísticas em partidas de futebol.

O reduzido volume de interação nos terminais de recepção, a distração do telespectador, o caráter de difusão em sistemas televisivos e a sobrecarga de informações se apresentam como desafios à disponibilização destes serviços interativos. Um aspecto desejável é a personalização de aplicações, de maneira a oferecer informações ajustadas aos interesses de cada usuário.

Para se obter uma TV personalizada, é necessário manter algum tipo de conhecimento sobre o usuário. Nesse sentido, o suporte à representação e recuperação automática de informações do usuário é fundamental. Uma solução, neste caso, é o uso conjunto de informações do contexto e a seleção de preferências explicitadas pelo usuário para identificar quem está em frente à TV.

Aplicações sensíveis ao contexto podem ser projetadas de diferentes maneiras. No entanto, a literatura científica propõe a separação entre processos de aquisição e processos de uso de contexto, consideradas tarefas relacionadas ao propósito das aplicações, a fim de melhorar o reuso e a extensibilidade de sistemas (DEY, 2000).

O presente trabalho apresenta, portanto, uma abordagem, chamada AvanTV¹, para adequação do conteúdo de aplicações de TVDI sensíveis ao contexto. Tal abordagem consiste na especificação de informações de contexto para captura e acesso para personalização por aplicações interativas. Para isso, foi desenvolvido um modelo capaz de descrever as características do usuário e de organização do conteúdo. A correspondência entre estas estruturas é realizada por conceitos de uma ontologia de contexto, que expressa os possíveis interesses do usuário pela incorporação de informações semânticas. Neste trabalho, a semântica de conteúdo é específica às transmissões esportivas. Também desenvolvemos um protótipo da abordagem AvanTV, integrado à plataforma de *middleware* Ginga presente nos receptores, de modo a fornecer mecanismos para uso de contexto.

1.2 Objetivos

O principal objetivo deste trabalho consiste em desenvolver uma abordagem para personalização do conteúdo de aplicações de TVDI que considere a sensibilidade ao contexto do usuário.

De modo a alcançar este objetivo geral, destacamos os seguintes objetivos específicos:

- Investigar a personalização de aplicações interativas na literatura de Sistemas Sensíveis ao Contexto e TVDI;
- Analisar modelos para representação de informações sobre telespectadores/usuários;

¹ AvanTV foi denominada com o sentido de seguir adiante ou evoluir os serviços de TV.

- Verificar técnicas de adaptação a serem utilizadas para personalização de conteúdo em aplicações de TVDI;
- Definir uma metodologia para representação de informações contextuais da abordagem AvanTV;
- Desenvolver um protótipo para fornecer serviços personalizados de TVDI conforme as preferências derivadas do perfil do usuário. Os objetivos do protótipo são os seguintes: apresentar um modelo de domínio dos interesses específicos aos esportes pela audiência; adaptar a seleção do conteúdo de acordo com a identificação do contexto do usuário; e validar a abordagem AvanTV.

1.3 Estrutura da Dissertação

Os demais capítulos desta dissertação estão assim organizados:

No capítulo II são apresentados os conceitos fundamentais acerca do estado da arte sobre Sistemas de TVDI e sobre Computação Sensível ao Contexto, assim como métodos de modelagem e de adaptação pertinentes.

O capítulo III apresenta alguns trabalhos relacionados que exploram a sensibilidade ao contexto para aperfeiçoar serviços interativos. Considerações acerca das características de cada abordagem são feitas de maneira comparativa.

O capítulo IV discute sobre a organização das informações do usuário e do conteúdo para modelagem contextual proposta pela abordagem AvanTV.

No capítulo V são descritos projeto e implementação do protótipo que viabiliza o uso de contexto por aplicações, em termos da infraestrutura adotada, ferramentas utilizadas e serviços oferecidos. A seguir, um estudo de caso é ilustrado para validação dos resultados.

Finalmente, o capítulo VI apresenta as conclusões pelo resumo das questões tratadas e contribuições obtidas, além de ponderar sobre trabalhos futuros.

Capítulo II.

FUNDAMENTAÇÃO TEÓRICA

Individualizar informações através de sistemas computacionais abrange tecnologias e paradigmas diversos. De modo a tornar o conteúdo televisivo pessoal, considerou-se o emprego de informações de contexto e técnicas de adaptação para aperfeiçoar aplicações interativas. A seguir são introduzidos fundamentos básicos para o desenvolvimento deste trabalho. A seção 2.1 descreve ferramentas para representação do conhecimento. A seção 2.2 trata da TVDI. A seção 2.3 apresenta os Sistemas Sensíveis ao Contexto. Na seção 2.4 são resumidas técnicas de adaptação do conteúdo. Nas considerações finais, os conceitos são relacionados ao tema deste trabalho.

2.1 Representação do Conhecimento

O problema em decidir quais estruturas lógicas utilizar para representar um conjunto de dados tem impacto direto na concepção, na recuperação e no gerenciamento da informação. A Web Semântica, uma extensão da *web* atual que propõe integrar e vincular conteúdo entre aplicações pela atribuição de significado aos dados (BERNERS-LEE *et al.*, 2001). Para isso, documentos com conteúdo estruturado, compreendidos por humanos, são acrescidos de metadados e ontologias para interpretação por sistemas computacionais.

A metalinguagem XML (*eXtensible Markup Language*), um padrão para descrição hierárquica de dados, é amplamente utilizada para intercâmbio entre aplicações (XML W3C, 2008). Documentos XML são constituídos por *tags* com seus atributos e conteúdo. Um esquema XML (ou *XML Schema*²) descreve e restringe o conteúdo desses documentos, de modo a serem validados segundo regras específicas (LUGMAYR *et al.*, 2004). Por exemplo, o documento

² Define a estrutura de documentos XML: <http://www.w3.org/XML/Schema>

listaEsportes.xml a seguir pode ser descrito pelo esquema esporte.xsd mais abaixo:

```
<!-- listaEsportes.xml -->
<?xml version="1.0" encoding="ISO-8859-1"?>
<esportes xmlns="urn:avantv:sport:schema:2010"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:avantv:sport:schema:2010
    ../model/esportes.xsd">
  <esporte categoria="Artes Marciais e Luta">
    <nome>Karatê</nome>
  </esporte>
  <esporte categoria="Esporte Coletivo">
    <nome>Futebol de campo</nome>
  </esporte>
</esportes>
```

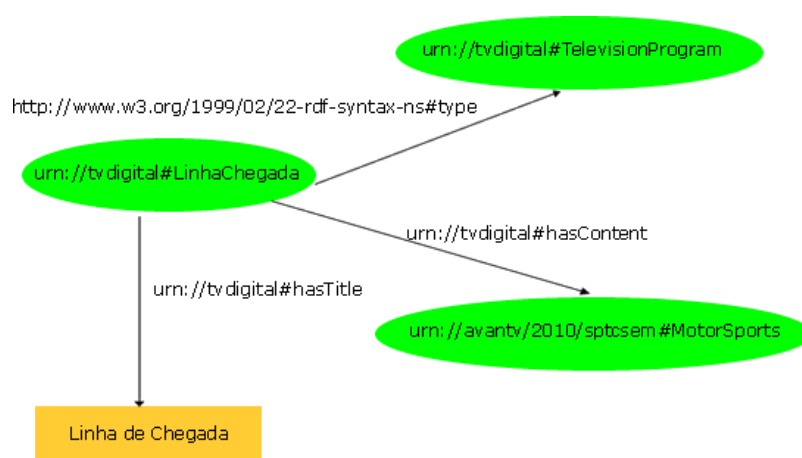
O elemento raiz `esportes` é constituído de sequência de elementos `esporte`, contendo um sub-elemento `nome` e um atributo `categoria`.

```
<!-- esportes.xsd -->
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:avantv:sport:schema:2010"
xmlns="urn:avantv:sport:schema:2010">
<xs:element name="esportes">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="esporte" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="nome" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="categoria" type="xs:string"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Integrar dados descritos em XML não é tão simples, por isso são incluídos níveis de conhecimento às informações para oferecer maior compreensão por sistemas. O padrão RDF (*Resource Definition Framework*) facilita essa integração, inclusive em esquemas distintos, pela descrição de metadados associados a recursos (RDF W3C, 2004). O modelo de dados RDF consiste em fazer afirmações (*statements*) sobre recursos na forma de triplas (sujeito, predicado,

objeto), onde sujeito e predicado são recursos identificados pelo padrão URI (*Uniform Resource Identifier*³) e objeto pode ser um URI ou um literal. Deste modo, é possível relacionar dois recursos através de uma propriedade. Por exemplo, a declaração “o programa *Linha de Chegada* apresenta conteúdo sobre esportes motorizados” pode ser expressa pelo grafo RDF da Figura 2.1. Os recursos são exibidos como elipses e as propriedades, como arcos. Valores literais são representados por retângulos.

Figura 2.1: Representação em grafo de declaração RDF



Dentre as formas de serializar as declarações RDF, a mais comum é através da XML, como apresentado no exemplo de sintaxe abaixo para o grafo anterior. O prefixo `tv` representa o espaço de nomes definido pela URI `urn://tvdigital` e o `rdf` referencia os termos do vocabulário RDF.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:tv="urn://tvdigital" >
  <tv:TelevisionProgram rdf:ID="LinhaChegada">
    <tv:hasTitle>Linha de Chegada</tv:hasTitle>
    <tv:hasContent
rdf:resource="urn://avantv/2010/sptcsem#MotorSports"/>
  </tv:TelevisionProgram>
</rdf:RDF
```

³ Padrão para identificação única de recursos <http://www.ietf.org/rfc/rfc1630.txt>. Classifica-se em URL (*Uniform Resource Locator*) ou URN (*Uniform Resource Name*).

A definição de vocabulários RDF é realizada por esquemas RDF⁴ (ou *RDF Schemas* - RDFS). Um esquema RDF é uma extensão semântica do RDF que provê mecanismos para descrever grupos de recursos relacionados (classes) e o relacionamento entre os mesmos (propriedades). A declaração de que um recurso é instância de uma classe é assinalada pela propriedade `rdf:type`. A especificação das classes de recursos à que as propriedades se aplicam é feita pelas propriedades RDF `domain` e `range`. O `rdfs:domain` é utilizado para indicar que determinado recurso (sujeito) que contém uma dada propriedade é instância de uma ou mais classes. O `rdfs:range` designa quais valores de uma propriedade (objeto) são instâncias de uma ou mais classes. Para o exemplo anterior, o fragmento de código a seguir descreve que a propriedade `hasTitle` tem como domínio as instâncias da classe `TelevisionProgram` e os objetos são valores literais:

```
<rdf:Property rdf:ID="hasTitle">
  <rdfs:domain rdf:resource="#TelevisionProgram"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
```

Além destas, RDFS contém outras propriedades que enriquecem a semântica de modelos RDF, como `rdfs:subClassOf` e `rdfs:subPropertyOf`. Estas propriedades são usadas para incluir sub-classes e sub-propriedades de classes e propriedades, respectivamente.

A diferença entre estas descrições e a representação XML é que as declarações junto às propriedades semânticas adicionais (como simetria) possibilitam a interpretação por sistemas e podem ser usadas para derivação de relacionamentos dinâmicos entre recursos. Além disso, interpretadores de resultados podem ser usados por aplicações, onde afirmações sobre recursos podem ser acumulados de diferentes origens, se identificados pelo mesmo URI. A essa definição de conceitos e capacidade de prover raciocínio a artefatos de *software* é atribuído o termo **Ontologia**.

⁴ Linguagem para descrição de vocabulário RDF <http://www.w3.org/TR/rdf-schema/>

Uma ontologia é uma especificação explícita formal de conceitos em um domínio de discurso (classes), propriedades de cada conceito descrevendo características e atributos dos conceitos com respectivas restrições (NOY e MCGUINNESS, 2001). Uma ontologia junto com um conjunto de instâncias (ou indivíduos) de classes constitui uma base de conhecimento. Existem diferentes metodologias para construção de ontologias que, em geral, consideram (1) a determinação do domínio e escopo a que se propõem, (2) o reuso de ontologias existentes e (3) definições iterativas de seus elementos (classes, propriedades, hierarquias, restrições e indivíduos).

A linguagem RDFS provê descrição de ontologias. Ela foi estendida, de modo a ampliar a expressividade, através da linguagem OWL (*Web Ontology Language*). Algumas das características originais da OWL são proposições entre classes (como disjunção e equivalência), cardinalidade (por exemplo, “exatamente um”), equivalências e características de propriedades (por exemplo, transitividade, sub-propriedades). A capacidade em inferir novas informações é ocasionada pela definição de conceitos, conjunto de declarações e regras que constituem a ontologia. De modo a limitar a complexidade da OWL conforme requisitos das aplicações são definidos os seguintes subconjuntos (OWL W3C, 2004):

- *OWL Lite*: provê construtores mais simples pela restrição da linguagem, como permitir apenas valores de cardinalidade 0 ou 1. Apesar de reduzir a expressividade das ontologias, facilita a implementação e otimiza o processo de inferência.
- *OWL DL*: estende *OWL Lite* e inclui todos os construtores da linguagem OWL, mas sob restrições da lógica de descrição de modo a preservar completude e capacidade de decisão;
- *OWL Full*: garante o máximo de expressividade e liberdade sintática do vocabulário RDF ou OWL pré-definido, mas sem garantias computacionais.

Quando ontologias são orientadas a um assunto específico denominam-se ontologias de domínio ou de nível inferior. Por outro lado, quando são genéricas designam ontologias de nível superior ou independentes de domínio (GU *et al.*, 2005). Neste trabalho, o desenvolvimento de ontologias para descrição de conceitos esportivos e do contexto do usuário utilizou a OWL. Parte dos vocabulários são reusados de ontologias existentes. Ao invés de importá-las, alguns dos seus termos são mapeados por construtores de equivalência. Deste modo, a sobrecarga do processamento de inferências é reduzida. As ontologias referenciadas são:

- A ontologia de TV Digital (TVD), desenvolvida por LEITE *et al.*, (2007), que trata conceitos que descrevem componentes inerentes, tais como recursos computacionais, aplicação e programa televisivo.
- A descrição de pessoas e relacionamentos interpessoais estende termos da ontologia FOAF (*Friend of a Friend*)⁵.
- Além disso, vocabulários empregados na descrição de coleções de recursos pelo padrão *Dublin Core*⁶ foram úteis à caracterização de elementos do modelo.

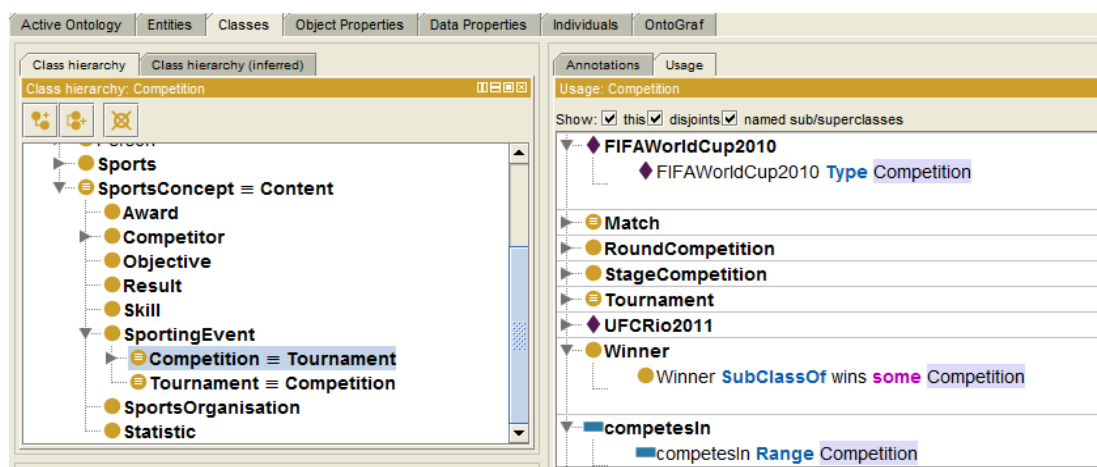
O processo de construção das ontologias foi apoiado pelas ferramentas de edição Protégé (PROTEGE, 2011) e o *framework* Jena (JENA API, 2011).

O ambiente Protégé provê um conjunto de estruturas para modelagem de conhecimento que auxiliam na criação, visualização e manipulação de ontologias em diferentes representações. Além disso, seu editor fornece facilidades para navegação e inferência em modelos OWL e pode ser estendido pela integração de *plugins*. A Figura 2.2 ilustra a descrição da classe *Competition*, da ontologia de domínio dos esportes. O detalhamento de uso mostra indivíduos (FIFAWorldCup2010 e UFCRio2011), subclasses (Match, RoundCompetition e StageCompetition), classes equivalentes (Tournament) e uma propriedade de objetos (competesIn).

⁵ The Friend of a Friend (FOAF) Project: <http://www.foaf-project.org/>

⁶ Dublin Core Metadata Element Set: <http://dublincore.org/documents/dces/>

Figura 2.2: Descrição da classe Competition da ontologia de esportes na ferramenta de edição Protégé-OWL 4.



A API Java para construção de aplicações da Web Semântica do Jena provê componentes para manipulação de modelos RDFS e OWL, inferências baseadas em regras, serialização em formatos variados (por exemplo, RDF/XML, N3 e N-Triples), além de processamento de consultas.

2.2 Televisão Digital Interativa

Desde sua concepção na década de 50 na América do Norte, com programação, principalmente, de notícias, entretenimento e esportes, poucas mudanças foram adicionadas à televisão, que permaneceu, essencialmente, inalterada, até o advento da digitalização do seu sinal. Kosch *et al.*, (2008) declarou que, não apenas emissoras, mas qualquer pessoa poderia disseminar e, certamente, receber qualquer tipo de mensagem a qualquer momento com qualquer dispositivo. Deste prenunciado cenário de difusão mais “democrática”, a TVDI é a propulsora da integração das diversas mídias.

Esta seção trata dos fundamentos da TVDI. É apresentada sua arquitetura geral e os padrões existentes, inclusive aqueles adotados no sistema brasileiro. São discutidos aspectos relacionados à interatividade e descritos os metadados utilizados em sistemas televisivos.

2.2.1 Arquitetura e Padrões

Em geral, um sistema de TVDI é constituído de elementos capazes de realizar desde a produção de programas a serem veiculados até o processamento para respectiva apresentação. O padrão ISO MPEG-2, instituído pelo *Moving Picture Experts Group* (MPEG), um consenso entre os principais sistemas de televisão digital, define um conjunto de tecnologias para compressão, codificação e multiplexação (MORRIS e SMITH-CHAIGNEAU, 2005). Os dados capturados são convertidos em formato digital e submetidos às técnicas de compressão para reduzir a quantidade de dados combinados ao sinal e produzir fluxos elementares (*Elementary Streams* – ES). O sinal é, então, transmitido como uma estrutura MPEG-2 *Transport Stream* – TS, que combina pacotes ES e dados, multiplexados e associados a identificadores que indicam o tipo de conteúdo e outros descritores sobre a organização do fluxo. Um conjunto de estruturas com essas informações é definido pelas tabelas *Programa-Specific Information* (PSI). Uma vez que o conteúdo esteja pronto para transmissão, o fluxo de bit digital é convertido na forma de onda própria ao meio de transmissão (terrestre, satélite, cabo), denominado modulação. De forma geral, a Figura 2.3 ilustra esse processo.

Figura 2.3: Visão geral do sistema de difusão



Por se tratar de um meio de transmissão primariamente unilateral, as redes de TVDI utilizam o conceito de carrossel de dados e de objetos do protocolo *Digital Storage Media – Command and Control* para enviar dados para os clientes processarem e garantir a entrega da informação a qualquer instante de sintonização, através de um sistema de arquivos que transmite periodicamente todos os arquivos em ciclos (FERNANDES *et al.*, 2004).

Para a devida apresentação do programa e conteúdo vinculado é necessário um cliente responsável pela recepção, demodulação e decodificação do sinal televisivo (ex. *set-top box* – STB, aparelhos móveis ou TVs integradas). Os receptores são caracterizados por limitada capacidade de recursos computacionais e interação por controle remoto. De modo a possibilitar a independência sobre hardware e software heterogêneos, é fornecida uma API (*Application Programming Interface*) genérica e padronizada que facilita a portabilidade das aplicações, denominada *middleware*. Conseqüentemente, as aplicações não necessitam de múltiplas versões para acessar as especificidades de cada plataforma, porém elas se tornam dependentes da API fornecida (FERNANDES *et al.*, 2004).

A compatibilidade entre sistemas é obtida pela adoção de padrões e, para cada sistema, são empregados modelos decorrentes de questões comerciais e políticas. O padrão mais usado é o europeu *Digital Video Broadcast* (DVB), que criou especificações para todos os tipos de redes e do *middleware Multimedia Home Platform* (MHP), uma plataforma aberta em que aplicações do tipo declarativo DVB-HTML e procedural DVB-J executam (DVB, 2007). Os padrões desenvolvidos pelo grupo americano *Advanced Television System Committee* (ATSC) se referem ao *middleware DTV Application Software Environment* (DASE), que oferece suporte às aplicações Java TV e HTML (ATSC, 2003). Enquanto o sistema japonês *Integrated Services Digital Broadcasting* (ISDB), baseado no sistema europeu, tem plataforma mantida pela organização *Association of Radio Industries e Businesses* (ARIB), e favorece a recepção móvel e portátil, além de definir a linguagem declarativa BML (*Broadcast Markup Language*) (ARIB, 2007). Assim como o MPEG-2 é o elemento comum no processo de difusão, a especificação GEM (*Globally Executable MHP*) (GEM, 2010), proposta pelo grupo DVB, é compartilhada pelos *middlewares* para execução de aplicações.

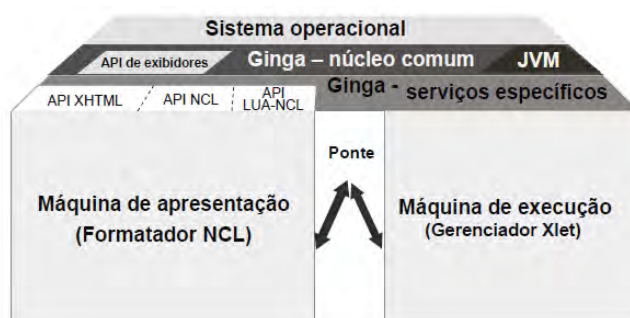
2.2.2 Sistema Brasileiro de TV Digital

Considerado um dos principais meios de comunicação no Brasil, a televisão está presente em 98% dos domicílios, enquanto apenas 35% possuem

computadores e menos de 28% acesso à Internet (IBGE, 2009). Após avaliação dos padrões existentes, o Sistema Brasileiro de TV Digital (SBTVD) adotou o padrão japonês para transmissões terrestres devido, principalmente, às suas características técnicas mais amplas e referentes à mobilidade. Estabelecida em 2006, a cobertura do sinal já atinge 46% da população e há previsão de que todos os radiodifusores transmitam no formato até 2013 e o desligamento dos sinais de TV analógica ocorra em 2016 (SBTVD, 2011). O padrão brasileiro já é aceito por diversos países, tais como Peru, Argentina, Chile, Uruguai, Nicarágua e Costa Rica.

O principal fator considerado nas pesquisas e na indústria refere-se à interatividade. Neste caso, o componente fundamental para o desenvolvimento de serviços interativos é o *middleware*, denominado Ginga (SOARES e LEMOS, 2007), cuja especificação de referência foi definida pelo LAVID/UFPB⁷ e Telemídia/PUC-Rio⁸. Assim como os demais padrões, ele é capaz de processar tanto aplicações declarativas, pelo ambiente Ginga-NCL, quanto imperativas, pelo Ginga-J. Ambos são visualizados na arquitetura do Ginga, ilustrada na Figura 2.4.

Figura 2.4: Arquitetura Ginga (Fonte: GINGANCL, 2007)



A máquina de apresentação Ginga-NCL (GINGANCL, 2007) processa documentos NCL (*Nested Context Language* – Linguagem de Contextos Aninhados), capazes de descrever o comportamento temporal de uma apresentação multimídia, associar interação do usuário a objetos de mídia, definir alternativas para apresentação e exibição em múltiplos dispositivos. Este módulo é composto pelo

⁷ Laboratório de Vídeo Digital na Universidade Federal da PB. <http://www.lavid.ufpb.br/>

⁸ Laboratório Telemídia na PUC-Rio. <http://www.telemidia.puc-rio.br/>

Formatador NCL e o Gerenciador de Base Privada, além de exibidor XHTML e interpretadores de scripts Lua e estilos. O formatador NCL é responsável por receber a especificação de um documento e controlar sua apresentação. Os documentos em uma base privada podem ser iniciados, pausados, recuperados, parados e referenciados entre si.

O subsistema Ginga-J (GINGAJ, 2010) processa conteúdo de aplicações procedurais (*Xlets Java*) através de uma máquina virtual Java (JVM – *Java Virtual Machine*). Devido a questões de propriedade intelectual, a *Sun* em parceria com o fórum SBTVD, propôs a especificação JavaDTV (JAVADTV, 2010), em substituição ao GEM, composta pelas APIs JavaTV, componentes gráficos *Lwuit* e serviços da plataforma *Java Micro Edition*. Informações nas tabelas de serviços (SI – *Service Information*) que descrevem programas e serviços presentes em fluxos MPEG-2 são acessíveis por métodos das APIs. O modelo de aplicações contempla a mudança de ciclo de vida dos *Xlets* através de quatro estados (*Loaded*, *Paused*, *Active* e *Destroyed*). Projetado para ser extensível, oferece suporte a aplicações concorrentes que cooperam ao se comunicar entre si e compartilhar recursos, ou independentes e competindo por recursos. A Figura 2.5 ilustra uma classe *Xlet* com métodos que provêm atualização de estados (métodos da interface *Xlet*), notificação sobre liberação de um determinado recurso (métodos da interface *ScarcResourceListener*) e eventos de entrada do usuário (método da interface *UserInputEventListener*).

A arquitetura Ginga é composta ainda por um núcleo comum (*Ginga Common Core*) constituído por decodificadores e procedimentos para obter conteúdo dos fluxos de transporte ou do canal de interatividade⁹. O controle de ambos os ambientes é responsabilidade deste subsistema que apresenta, inclusive, uma ponte para construção de aplicações híbridas. A ligação é realizada tanto por uma

⁹ Canal de comunicação que provê interconexão das redes de televisão com as redes de telecomunicações. Também conhecido como canal de retorno.

aplicação declarativa que contém conteúdo de objeto ativo quanto por uma aplicação procedural com conteúdo declarativo.

Figura 2.5: Estrutura básica de um Xlet Java

```
public class InitXlet implements Xlet, ScarceResourceListener, UserInputEventListener {

    // métodos chamados para mudança de ciclo de vida do Xlet
    public void initXlet(XletContext xletContext) throws XletStateChangeException {
    }

    public void startXlet() throws XletStateChangeException {
    }

    public void pauseXlet() {
    }

    public void destroyXlet(boolean flag) throws XletStateChangeException {
    }

    // metodos herdados de ScarceResourceListener
    public void releaseForced(ScarceResource arg0) {
    }

    public boolean releaseRequested(ScarceResource arg0) {
        return false;
    }

    public void released(ScarceResource arg0) {
    }

    // metodo herdado de UserInputEventListener
    public void userInputEventReceived(UserInputEvent event) {
    }
}
```

As APIs do *middleware* Ginga oferecem ainda funcionalidades avançadas que permitem o uso de múltiplos dispositivos de interação. O mecanismo de identificação da origem de cada interação viabiliza aplicações que envolvam diversos usuários. Além disso, em virtude da necessidade de descrição adicional do conteúdo são definidos elementos que acoplam metadados.

2.2.3 Interatividade na TV

A propriedade que caracteriza um sistema como interativo é sua capacidade de reciprocidade direta e temporal de ações com os usuários (SCHMIDT, 2002). Aplicações desta natureza não são mais restritas aos convencionais sistemas *desktop* ou à infraestrutura *web*. Na proporção em que dispositivos eletrônicos são dotados de processamento, são incluídos mecanismos para atribuir eventos às interações realizadas pelo usuário, cabendo aos aplicativos associar ações aos

mesmos. Projetar esta classe de serviços na TV possui seus próprios desafios. Transpor métodos e técnicas de desenvolvimento de aplicações entre esses domínios não funciona por diversos motivos, que incluem questões técnicas e conduta da audiência.

Os elementos gráficos da interface televisiva devem ser simples e diretos, evitando grande volume de informações que causem confusão. Além disso, o modelo de distribuição predominantemente unidirecional restringe a independência temporal de consumo de mídia. A impressão de compartilhar o que está sendo assistido com outros telespectadores é uma característica atraente para muitos. Outro contraste refere-se ao relacionamento com o meio. Normalmente, as pessoas assistem à televisão na companhia de outras, como atividade social e familiar, envolvidas em tarefas cotidianas, orientadas ao lazer ao invés do desempenho de tarefas ou apenas mudando entre canais. Diferentemente do usuário perante o computador, cuja postura é atenta e consultiva, diante da TV o mesmo assume uma atitude relaxada e absorta. Esse estilo de consumo de mídia é, freqüentemente, caracterizado como *leaning back* (MASTHOFF e PEMBERTON, 2005). As aplicações destinadas à TV, portanto, requerem um modo de interatividade particularmente projetado para as restrições apresentadas.

Permitir que telespectadores tenham informações adicionais sobre a programação, selecionar ângulos de câmera, acessar a internet ou enviar e-mails diretamente na televisão são algumas das oportunidades de interação ocasionadas pela digitalização do sinal de *broadcast*. Até então, a comunicação era realizada por meios alternativos, como ligações telefônicas, envio de mensagens por celular (SMS) e internet. Na década de 90, por exemplo, a emissora Rede Globo possibilitava que o público votasse por telefone para decidir o desfecho de histórias no programa *Você Decide* (vide Figura 2.6.a) ou a exibição de filmes no dia seguinte (vide Figura 2.6.b) (MEMÓRIAGLOBO,2011). A participação por tecnologia SMS tem sido amplamente utilizada em votações e chats. A internet também é empregada para receber além de texto, áudio e imagem, a exemplo do *Você no JPB* (vide Figura 2.6.c), que o telespectador faz a notícia sobre sua comunidade.

Figura 2.6: Programas interativos em outros meios



O suporte a execução de aplicativos na TV dispensa a necessidade de um meio separado do ambiente de *broadcast* para troca de informações. Para Morris e Smith-Chaigneau (2005), essas aplicações são categorizadas segundo a dependência de canal de interatividade em:

- *Enhanced Broadcast*: interações ocorrem apenas entre o usuário e o receptor, com dados embutidos e não requer canal de comunicação. Destinado aos receptores de baixo custo, fornece funcionalidades do *middleware* e aplicativos que ele processa.
- *Interactive Broadcast*: oferece ao usuário serviços que necessitam comunicação através de um canal de interatividade. Neste caso, as aplicações podem obter mais dados além do fluxo de transmissão e requer uma API para controle de acesso.
- *Internet TV*: permite acesso às funcionalidades Web (como navegação e correio eletrônico) através da TV. O usuário pode atuar integradamente com aplicações da emissora e da Internet.

O conteúdo sobreposto ao vídeo principal, na forma de texto ou de gráficos, pode ser sincronizado ou independente da programação. No primeiro caso, o telespectador tem acesso às informações complementares à exibição de um programa, como a inclusão de estatísticas sobre um evento esportivo (vide Figura 2.7a). No segundo caso, os serviços desconexos normalmente incluem previsão do tempo, boletins de notícias ou governo eletrônico, como o serviço de consulta ao calendário de pagamento da previdência ilustrada na Figura 2.7b. Guias Eletrônicos

de Programação (EPG), como ilustrado na Figura 2.8, são os serviços de TVDI mais elementares que incluem informações sobre a programação. Geralmente, apresenta mecanismo de busca e filtragem de conteúdo baseada no gênero do programa.

Figura 2.7: Exemplos de interatividade local

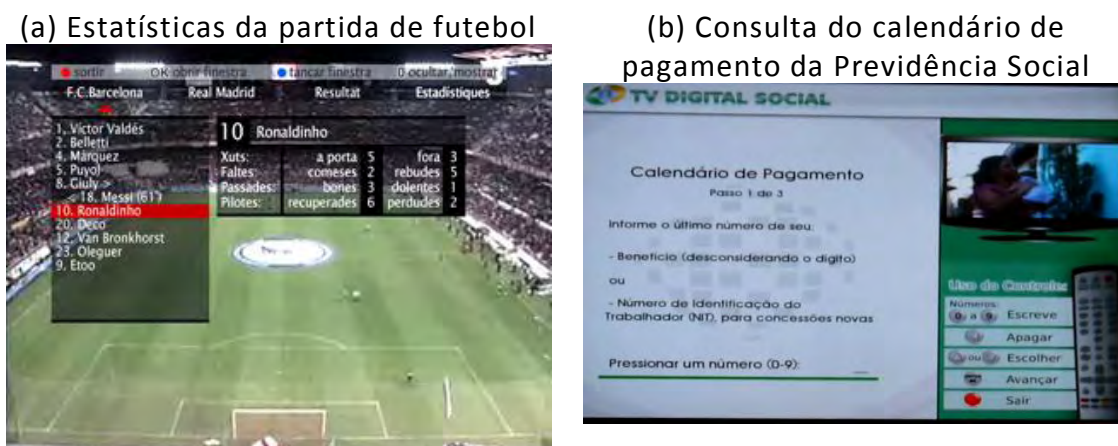


Figura 2.8: Exemplo de um guia eletrônico de programação (EPG)



A presença de canal de comunicação amplia a capacidade de interatividade do telespectador e permite que provedores de serviços e emissoras recebam o resultado da interação. Aplicações que requerem participação tais como votação (vide exemplo do *Big Brother* na Figura 2.9.a, *chat* ou *enquete* ilustram este formato. O modelo de receita baseado em espaços publicitários é predominante no Brasil e se beneficia da interatividade na TV para que o telespectador obtenha mais informações relativas a um produto ou marca, expresse opinião, efetue compra ou participe de promoções durante o intervalo comercial (Figura 2.9.b). Em pesquisa conduzida pela *Nielsen Company* (NIELSEN, 2009), foi identificado que os latino-

americanos são mais propensos a se sentirem informados (68%) em relação aos comerciais de TV, do que em relação aos vídeos publicitários *online*. Isto demonstra o alto nível de envolvimento a ser atingido pela publicidade e pela propaganda interativa através deste meio.

De modo a aprimorar a experiência do telespectador e agregar valor aos programas, os serviços de TVDI precisam ser oferecidos de maneira pessoal. Um aspecto desejável é a personalização do conteúdo para proporcionar informações diferenciadas a cada usuário. Nesse sentido, construir uma TV personalizada exige o conhecimento sobre comportamento e interesses do usuário. Buczak *et al.* (2002) propõe, por exemplo, a combinação das preferências do usuário a mecanismos de recomendação para o desenvolvimento de um EPG personalizado. Para refletir o comportamento social de interação na TV, Goren-Bar e Glinansky (2002) utilizam o conceito de estereótipos para agrupar preferências dos usuários e sugerir programas. Adaptar conteúdo, segundo interesses da audiência, requer gerenciamento de informações que descrevam sua semântica e também métodos de correspondência. Por exemplo, para possibilitar seleção personalizada de conteúdo e integrá-los em diferentes formatos a partir de múltiplos dispositivos, o projeto NoTube explora o desenvolvimento de serviços e ferramentas baseados na semântica dos dados (AROYO *et al.*, 2009).

Figura 2.9: Aplicações interativas

(a) Votação que requer interatividade plena



(b) Anúncio interativo de produtos e serviços



2.2.4 Metadados na TVDI

O enriquecimento de dados é realizado através de metadados. Na TVDI, a diversidade de conteúdos e de serviços impõe desafios quanto à complexidade e sobrecarga de dados. O uso de metadados neste meio oferece suporte à gerência da informação (LUGMAYR *et al.*, 2004).

Para Cesar e Chorianopoulos (2008) a modelagem semântica, que associa metadados aos dados de mídia digital, é o primeiro passo no desenvolvimento de serviços interativos, sendo essencial para busca e seleção de conteúdo. No entanto, a literatura tem se concentrado quase que exclusivamente na distribuição de “dados reais” e, se descritores são necessários, sua disponibilidade é simplesmente assumida. Kosch *et al.* (2008) destaca que, em cenários dinâmicos, sem gerenciamento de metadados os usuários não conseguem adquirir os dados que precisam.

Um dos padrões amplamente adotado em sistemas de TV Digital é o MPEG-2, através do uso de descritores de baixo nível, tais como as tabelas PSI/SI. Conforme Lugmayr *et al.* (2004), apesar de mais fáceis de manipular e apropriadas para *streaming*, a rigidez desses descritores dificulta a oferta de serviços mais avançados, como os de personalização. Portanto, padrões mais flexíveis, como MPEG-7, TV-Anytime e MPEG-21, são utilizados no meio televisivo.

a) MPEG-7

O MPEG-7 é um padrão ISO/IEC, desenvolvido pelo grupo MPEG, que fornece um conjunto de descritores para conteúdo multimídia. Ele tem como objetivo prover algum grau de interpretação no significado da informação (MARTINEZ, 2004). Para diferentes domínios, seus elementos têm variadas definições. Os principais elementos definem as características e a estrutura de relacionamento entre seus componentes. A representação dos dados é uniformizada, porém o processo de empregá-los e de adquiri-los não. Em meio às funcionalidades que constituem o padrão, destacam-se:

- A linguagem DDL (*Description Definition Language*), baseada na linguagem *XML Schema*, projetada especificamente para descrição de conteúdo audiovisual, define a sintaxe dos descritores e permite estender e modificar esquemas existentes.
- Esquemas de descrição audiovisual: estruturas que descrevem sinal, timbre, instrumentos, entre outros aspectos para áudio; e cor, forma, movimento, entre outros para componente visual.
- Esquemas de descrição multimídia (*Multimedia Description Schemes – MDS*): apresentam conjunto de descritores para manipular características genéricas às demais mídias, tais como vocabulários controlados e informação espaço-temporal. Além disso, definem classes de funcionalidades mais avançadas referentes à organização do conteúdo e interação do usuário.
- Perfis e níveis: são definidos como subconjuntos e restrições sobre os demais descritores aplicados a propósitos específicos. Existem atualmente três perfis padronizados: *Simple Metadata Profile* (SMP) considera instancias individuais de imagem, áudio ou vídeo utilizadas, normalmente, em aplicações móveis; *User Description Profile* (UDP) provê melhoria ao acesso de dispositivos multimídia através da descrição de preferências pessoais e padrões de uso pelo usuário; *Core Description Profile* (CDP) inclui organização de coleções de conteúdo multimídia.

Segundo definição do ciclo de vida proposta por Lugmayr *et al.* (2004), o padrão MPEG-7 atua como descritor de tipos de dados multimídia, principalmente, nas etapas de produção e consumo do conteúdo.

b) TV-Anytime

Resultado dos trabalhos do grupo de organizações TV-Anytime Forum, o conjunto de especificações *TV-Anytime* possibilita busca, seleção, aquisição e

definição de direitos de uso de conteúdo em sistemas de armazenamento digital local ou remoto de serviços on-line e por difusão, essencialmente pelo uso de *Personal Digital Recorder* (PDR) (TV-ANYTIME, 2010).

Foram definidas funcionalidades suportadas pelas especificações normativas de metadados, referência ao conteúdo e gerenciamento/proteção de direitos autorais. O conjunto de metadados baseia-se no padrão MPEG-7 e em terminologias para classificação de conteúdo (*Classification Schemes*). Para referenciar um conteúdo independentemente da localização e momento de disponibilidade, é atribuído um identificador CRID (*Content Reference Identifier*) formado por um domínio segundo regras de nomeação que garantem exclusividade e uma parte de dados em formato textual livre em conformidade com o padrão URI. As operações sob conteúdo de programas e serviços resultam em índices com sintaxe: **CRID://<authority>/<data>**. Foram especificados, ainda, padrões relacionados a serviços sobre redes bidirecionais que incluem áreas como segmentação, redistribuição, intercâmbio de novos formatos e programação remota de PDR.

c) MPEG-21

Diferentemente dos demais padrões publicados pelo MPEG, o MPEG-21 tem como objetivo o uso transparente, protegido e interoperável de recursos multimídia através da cadeia de produção e consumo digital (BURNETT *et al.*, 2006). Foi introduzido o conceito de objeto digital estruturado (*Digital Item* - DI), normatizado como unidade de distribuição e transação dentro de um *framework*. Deste modo, qualquer conteúdo no MPEG-21 é envolvido em um “container virtual”, que permite a adição de outros recursos. Sua padronização contém atualmente vinte partes (MPEG21, 2011), das quais destacamos:

- *Digital Item Declaration* (DID) – especifica um conjunto de termos e conceitos que constituem o modelo de definição de DI.

- *Digital Item Identification* (DII) – apresenta elementos que reusam identificadores únicos existentes na DID, como ISBN para livros.
- *Intellectual Property Management and Protection* (IPMP) – fornece meios para proteção dos DI e seus direitos autorais.
- *Digital Item Adaptation* (DIA) – define a sintaxe e semântica de metadados utilizados na adaptação de DI, de modo a satisfazer restrições de transmissão, armazenamento e consumo.
- *Digital Item Processing* (DIP) – provê métodos para que o autor de um DI sugira como um usuário pode interagir com o DI

Apesar de recente, as vantagens decorrentes do uso de DI para o meio de difusão, como formato de arquivamento de conteúdos e reutilização em outros canais de comunicação (*crossmedia*) com garantias de proteção, tais como web, dispositivos móveis e *streaming*, já são incluídas em novos modelos, como o *Digital Broadcast Item Model* (LUGMAYR *et al.*, 2004).

2.3 Sistemas Sensíveis ao Contexto

Interpretar situações nas quais interações com sistemas computacionais estão inseridas possibilita a construção de aplicações em que a tecnologia desaparece da percepção do usuário. Com o aumento da quantidade de serviços proporcionados pela TVDI, pesquisas têm buscado soluções que considerem as particularidades da audiência e obtenham maior riqueza de dados enquanto mantêm a simplicidade de interação. Estas são algumas das questões endereçadas pelo paradigma de sensibilidade ao contexto.

A seguir, são apresentados conceitos básicos; algumas estratégias de modelagem de contexto; questões de projeto de aplicações sensíveis ao contexto; e, por fim, exemplos de uso de contexto.

2.3.1 Conceitos Básicos

Schilit *et al.* (1994) introduziu o termo Computação Ciente de Contexto (ou Computação Sensível ao Contexto) para designar sistemas capazes de se adaptar conforme localidade de uso, pessoas e recursos próximos, assim como mudanças desses objetos ao longo do tempo. O *Active Badge* (WANT *et al.*, 1992) foi o primeiro trabalho que utilizou a informação contextual de localização do usuário para redirecionar chamadas telefônicas.

Diversas aplicações que fornecem informação de acordo com o posicionamento foram desenvolvidas desde então, como os projetos *Cyberguide* (ABOWD *et al.*, 1997) e *GUIDE* (CHEVERST *et al.*, 2000) que oferecem roteiros turísticos cientes de localização. Embora a informação sobre localização tenha sido um dos atributos mais utilizados, a definição de **contexto** mais consagrada foi dada por Dey *et al.* (1999):

“qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo os próprios usuário e aplicação.”

É responsabilidade de um projetista decidir quais informações podem ser informações de contexto. Em geral, os atributos de contexto são descritos segundo quem (*who*), onde (*where*), quando (*when*), como (*how*) e o que (*what*) das entidades, para determinar por que (*why*) uma condição está acontecendo. A literatura apresenta categorias de contexto alinhadas a essas questões. Segundo Dey (2000) os tipos primários de contexto compreendem localização, identidade, atividade e tempo. Além de distinguir a situação de uma entidade, estes tipos atuam como índices na busca de dados secundários. De modo mais abrangente, Schmidt (2002) propõe uma organização para contexto relacionado a *fatores humanos* e ao *ambiente físico*. Cada uma dessas classificadas, respectivamente, em três grupos:

- *Pessoal* (por exemplo, preferências, hábitos, habilidades, estado emocional, etc.), *relacionamento social* (por exemplo, interação social, grupos dinâmicos, co-localização, etc.) e *atividade* (por exemplo, trabalhando, descansando, em movimento, etc.);
- *Condições* (por exemplo, iluminação, temperatura, etc.), *infraestrutura* (por exemplo, recursos computacionais, etc.) e *localização* (por exemplo, posição absoluta ou relativa, etc.).

Vieira (2008) define os dados que caracterizam uma entidade em um domínio como *elementos contextuais*, enquanto o conjunto das instâncias para apoiar na execução de alguma tarefa como o *contexto* em si. O elemento contextual é um conceito estático, definido durante o projeto, e usado para compor o contexto que, por sua vez, é dinâmico e construído durante a execução de uma interação. Portanto, a existência de contexto está condicionada à possibilidade de relação com outros contextos e à associação de elementos relevantes durante a realização de uma tarefa.

2.3.2 Estratégias de Modelagem de Contexto

O compartilhamento das informações de contexto requer formas padronizadas de representação. As principais estratégias para modelagem de contexto, graças às estruturas de dados utilizadas, podem ser resumidas a seguir (STRANG e LINNHOF-POPIEN, 2004):

- *Pares chave-valor*: são as estruturas mais simples para especificar contexto. Uma chave identifica o contexto, descrito por um valor associado, porém a recuperação por correspondência textual é inadequada em problemas complexos.
- *Linguagem de Marcação*: utiliza estrutura hierárquica do padrão XML. Apresenta problemas na captura de relacionamentos contextuais e limitação na resolução de ambigüidade.

- *Modelos gráficos*: úteis em estruturar informação contextual utilizando elementos visuais, tais como UML¹⁰ e ORM¹¹. Se por um lado facilita a especificação de conceitos e requisitos contextuais, por outro necessita de estruturas adicionais para processá-los.
- *Orientação a objetos*: são aproveitadas as potencialidades do paradigma (ex: herança, etc.) para representar diferentes tipos de contexto. Provê interfaces para acesso e processamento de contexto que, entretanto, dificulta o formalismo do modelo.
- *Baseado em lógica*: fatos, expressões e regras são usados para definir o modelo contextual. Um sistema baseado em lógica é usado para gerenciar os termos.
- *Baseado em ontologias*: descreve contexto em termos de conceitos e relacionamentos. Possui alta expressividade e formalismo, o que facilita o compartilhamento e interpretação de conhecimento. Pesquisas que utilizam este modelo (BULCÃO NETO, 2006; GU et al., 2005) representam contexto por linguagens de ontologias, como RDF e OWL.

Para avaliação destas estratégias, Strang e Linnhoff-Popien (2004) consideraram os seguintes aspectos: possibilidade de composição distribuída (*dc*), capacidade de validação parcial do conhecimento (*pv*), suporte à indicação do nível de qualidade da informação (*qua*), habilidade em gerenciar dados incompletos e ambíguos (*inc*), nível de formalidade (*for*) e aplicabilidade em ambientes existentes (*app*). A Tabela 2.1 resume a adequação a essas métricas.

Apesar das limitações, é preciso ponderar o emprego de XML como padrão para intercâmbio de dados estruturados em vastos domínios, como os metadados de TVDI. Segundo a análise, linguagens de marcação apresentam notável propensão a serem aplicadas e proverem validação na estrutura em uso. As

¹⁰ Unified Modeling Language - <http://www.uml.org/>

¹¹ Object Role Modeling - <http://www.orm.net/>

representações baseadas em ontologias são as mais propícias por melhor atender às condições enumeradas. Devido à sua capacidade em realizar raciocínio e ao conjunto de benefícios apresentados, é a modelagem mais utilizada no projeto de sistemas sensíveis ao contexto.

Tabela 2.1: Resumo das estratégias de modelagem de contexto

Estratégias de modelagem	dc	pv	qua	inc	for	app
Pares chave-valor	-	-	-	-	-	+
Esquemas de marcação	+	++	-	-	+	++
Gráficos	-	-	+	-	+	+
Orientados a objetos	++	+	+	+	+	+
Baseados em lógica	++	-	-	-	++	-
Baseados em ontologias	++	++	+	+	++	+

O modelo semântico de informações de contexto baseado em ontologias de nível superior proposto por Bulcão Neto (2006) é composto das dimensões de identidade, localização, tempo, atividade e modo de captura e acesso. Os conceitos associados à estrutura de sistemas sensíveis ao contexto definidos por gráficos em Vieira (2008) provê suporte à criação de modelos específicos ao domínio de aplicações. Conduzidos por estas pesquisas, propomos a captura e acesso de descritores de marcação, característicos aos metadados de TVDI, para especificação de um modelo contextual representado por ontologias.

2.3.3 Construção de Aplicações Sensíveis ao Contexto

Mais do que prover adaptação, Sistemas Sensíveis ao Contexto (SSC) utilizam contexto para fornecer informações e serviços relevantes aos seus usuários (DEY, 2000). As informações relevantes para uma dada situação podem ser definidas explicitamente, porém o objetivo de aplicações que usam contexto deve ser tornar as interações mais fáceis. Assim, o foco das pesquisas tem sido a aquisição automática de dados, ou seja, sem a intervenção do usuário, de modo que o

projetista da aplicação é o responsável por determinar as informações contextuais de maior importância e os métodos de reação a elas.

A falta de generalidade no desenvolvimento *ad-hoc* de aplicações sensíveis ao contexto tem motivado pesquisas que propõem a separação de interesses no processo de construção das mesmas (DEY, 2000). Em geral, as tarefas relacionadas à manipulação de contexto são: especificação, aquisição, processamento, distribuição e uso de contexto.

A *especificação* consiste em definir o comportamento sensível ao contexto e determinar quais os tipos de contexto necessários para ativá-lo. Tarefa esta realizada durante processo de modelagem de contexto.

A *aquisição* de informações contextuais é o processo que monitora e captura informações contextuais de fontes heterogêneas e distribuídas, tais como sensores físicos (por exemplo, termômetros, câmeras, etc.), lógicos (por exemplo, agentes inteligentes) ou virtuais (por exemplo, extratores de informações em outras aplicações, monitor de atividades de entrada do usuário). Três métodos para obter informação contextual se destacam (CHEN, 2004): *acesso direto aos sensores*, *baseado em middleware* e *servidor de contexto*. No método de *acesso direto aos sensores*, aplicações clientes adquirem a informação desejada diretamente de sensores embutidos nos dispositivos, o que possibilita melhor controle das operações dos mesmos, porém o alto acoplamento dificulta a manutenção. No método *baseado em middleware*, um componente gerencia o acesso concorrente a múltiplos sensores, o que facilita a extensibilidade, mas impõe aumento de carga computacional (isto pode ser um problema quando aplicado em dispositivos com menos recursos). No método por *servidor de contexto*, a arquitetura baseada em *middleware* é estendida pela introdução de um componente para gerenciamento remoto de acesso que atenua o consumo de recursos escassos.

Em trabalhos mais recentes (VIEIRA, 2008; AROYO *et al.*, 2009), as atividades de *aquisição*, *processamento*, *armazenamento* e *distribuição* são agrupadas numa infraestrutura intermediária de *gerenciamento de contexto* entre fornecedores e

consumidores de contexto. Consumidores podem acessá-lo de forma síncrona, quando o cliente envia mensagens solicitando algum tipo de dado oferecido e aguarda até receber resposta do servidor, ou assíncrona, quando cada cliente se inscreve em eventos específicos de seu interesse a fim de ser notificado ou ter um método acionado. A segunda forma de acesso é mais adequada nos casos de mudanças constantes e recursos limitados.

O gerenciamento de contexto pode ser modularizado e tratado independente de domínio. Geralmente, ele é projetado por uma arquitetura intermediária, útil em amenizar restrições de processamento. Como no *Context Toolkit* (DEY, 2000), cujos componentes denominados *widgets* abstraem detalhes dos mecanismos de sensoriamento das aplicações que utilizam contexto, fornece informações condensadas e interpreta informações elementares em significados de mais alto nível. No entanto, seu modelo baseado em pares chave-valor dificulta a inferência de novos fatos. Por outro lado, o projeto *Service-Oriented Context-Aware Middleware* (SOCAM) introduzido por Gu *et al.*, (2005) possibilita resolver conflitos e manter a consistência através de um componente de raciocínio sobre uma base de conhecimento. O SOCAM utiliza um servidor central, chamado interpretador de contexto, para coletar dados de contexto através dos fornecedores e banco de dados contextual.

Disponibilizar e enfatizar recursos ou executar serviços automaticamente são alguns exemplos de *uso de contexto*. O comportamento sensível ao contexto das aplicações, normalmente, é acionado por mecanismos de consulta e/ou notificação, sendo esta atividade considerada essencial e orientada ao propósito (DEY, 2000). Numa determinada situação, nenhum, um ou mais contextos podem satisfazer a relação de que a situação pertence ao contexto. Diferentes mecanismos para indicar qual contexto uma aplicação requer são encontrados na literatura. Chen (2004) propõe algoritmos baseados em regras para prover inferência de contexto independente do comportamento do sistema. Para contextos estáveis ao longo do tempo e associados à infraestrutura, Gu *et al.* (2005) descreve mecanismos acionados por regras dinâmicas para prover aprendizagem sobre o dispositivo que

determina as situações de contexto. Porém, a avaliação das condições pode ser aperfeiçoada pela atribuição de um grau de certeza às decisões, como por regras baseadas em lógica *fuzzy*, adequadas à atualização periódica de contextos que mudam ao longo do tempo, utilizadas por Leite *et al.* (2007).

2.3.4 Exemplos de Aplicações Sensíveis ao Contexto

Investigações em sistemas sensíveis ao contexto são encontradas tanto na aplicação prática do uso de informações de contexto para aperfeiçoar serviços e funcionalidades, quanto no tratamento computacional através de formalizações, *frameworks* e metodologias. Os resultados desses estudos são empregados nas pesquisas endereçadas por aquelas experiências práticas. Esta dissertação se refere ao primeiro grupo para o cenário de TVDI.

Dados de posicionamento têm sido amplamente utilizados para prover serviços cientes de localização (ABOUD *et al.*, 1997)(CHEVERST *et al.*, 2000), principalmente devido à propagação de dispositivos móveis. No entanto, diferentes tipos de contexto têm sido combinados para oferecer sistemas mais adaptativos.

Por exemplo, no *EasyMeeting* (CHEN, 2004) pessoas presentes numa palestra, seus percursos até o local da conferência e seus dispositivos habilitados com *bluetooth* (tais como celulares e PDAs) são utilizados para oferecer serviços ao palestrante e ao público, tais como configurações da apresentação e envio de notificações. As informações são mantidas numa base de conhecimento, representadas como triplas RDF. Baseado nas informações sobre o perfil das pessoas, um agente inteligente é capaz de detectar a chegada dos participantes e distribuir recursos sobre a apresentação (*slides* e URL para *download* de material).

Em outro exemplo, informações nutricionais de alimentos associadas ao horário de preparo das refeições e ao perfil (peso, altura, idade, sexo e nível de atividade) dos membros de uma família são obtidos por uma cozinha que realiza recomendações, como o uso de ingredientes e balanceamento de calorias (CHEN *et al.*, 2010). Para exibir valores nutricionais dos alimentos são combinadas

informações baseadas nas últimas ações de preparo registradas e dados de contexto relacionados ao período do dia e hábitos dos usuários.

Thawani *et al.* (2004) propõe a seleção de anúncios interativos que considerem as informações contextuais atuais e históricas do usuário, além das entidades de localização, identidade, atividade e tempo das aplicações. Contextos passados são analisados e agregados para derivar preferências e comportamento de uso do usuário na forma de regras. O modelo para representação de contexto baseia-se no padrão MPEG-7 para descrever informações sobre anúncios, perfis dos usuários e rede de difusão. Um algoritmo de análise dos canais e características de exibição do usuário é responsável por prever a natureza do anúncio com maior impacto.

Goularte (2003) também utiliza uma representação baseada em MPEG-7 para os tipos de contexto referentes à infra-estrutura de comunicação entre aplicação e dispositivo; ao contexto do usuário e o estado dos dispositivos do sistema; a semântica do domínio da aplicação e relacionamentos entre dispositivos e usuários; e informações sobre o ambiente onde uma determinada entidade se encontra. Um serviço de personalização de conteúdo foi desenvolvido com base na comparação das descrições dos programas às preferências do usuário.

2.4 Adaptação do conteúdo

Existe uma estreita relação entre sensibilidade ao contexto e capacidade de adaptação em aplicações. Adaptações podem ser realizadas no *comportamento* da aplicação, por mudanças no ambiente; na *interface* de apresentação ao usuário; na *interação* com outras entidades que se comunicam com a aplicação; ou ainda, no *conteúdo* das informações exploradas pela aplicação (WU, 2002). Enquanto os usuários têm preferências distintas por exibição de mídias ou quanto ao tamanho das informações, ajustes de conteúdo abrangem ainda alternativas em incluir, suprimir ou modificar os dados originais segundo o ambiente de uso. Limitaremos nossa abordagem à adaptação do conteúdo de aplicações no escopo da TVDI.

Em geral, a adaptação de conteúdo se refere ao ajuste do mesmo ao ambiente de uso. A natureza do conteúdo determina as operações envolvidas, seja pela adaptação no nível de sinal ou de semântica (BURNETT *et al.*, 2006). No primeiro caso, são realizadas modificações de resolução, taxa de *bits*, largura de banda ou recodificação segundo restrições de rede e dispositivo. No outro caso, o conteúdo é modificado pelo significado dos dados e, juntamente com informações do ambiente e do usuário, proporciona efetiva personalização. Pesquisas na área de Hipermedia Adaptativa (HA) propõem a construção de aplicações a partir da divisão em três modelos (WU, 2002):

- **Modelo de domínio:** consiste num conjunto de conceitos e relacionamentos correspondentes às informações específicas que descrevem os interesses do usuário, tais como tipos de mídia e vocabulário do conteúdo.
- **Modelo de usuário:** mantém conhecimento sobre o usuário e sua relação com o conteúdo, como preferências explícitas ou derivadas das interações, assim como características pessoais e do contexto.
- **Modelo de adaptação:** define como a aplicação se adapta às informações do usuário e como o modelo de usuário é atualizado. Mecanismos por regras são comuns para expressar estratégias de adaptação.

Essa categorização é análoga à especificação de adaptação de conteúdo digital do padrão MPEG-21, porém mantém os mecanismos de adequação fora do escopo de padronização. De modo a orientar as adaptações, consideramos a divisão acima para descrever quais informações de contexto são necessárias à execução de comportamento sensível ao contexto. Algumas técnicas de adaptação de HA foram estendidas por Masthoff e Pemberton (2005) para lidar com a TV interativa. Destacamos as três técnicas seguintes:

- *Exibição e ocultação seletiva:* consiste em apresentar ou suprimir itens do conteúdo. Por exemplo, notícias podem ser incluídas ou não

dependendo da relevância ou interesse pelo usuário ao invés de exibir as mesmas a todos;

- *Resumo e ampliação de objetos de mídia*: considera a apresentação de itens do conteúdo de forma sucinta com a opção de acesso a mais detalhes. Normalmente, combinam-se métodos de filtragem baseada em conteúdo para resumir a exibição dos itens mais apropriados. Cada item requer objeto complementar com a versão condensada indicando até quando efetuar a exibição, qual ação realizar para obter mais informação e o que apresentar para evitar exceder o tempo da programação.
- *Ordenação de itens*: corresponde à realocação da apresentação de itens mais importantes conforme modelo de usuário. A ordenação espacial pode ser aplicada quando múltiplos itens são exibidos simultaneamente.

Além dos benefícios de personalização já discutidos, a redução de dados propiciada pela maioria das adaptações de conteúdo oferece suporte a muitas das restrições de interatividade na TV apresentados na seção 2.2, porém nem todas as técnicas são aplicáveis a qualquer tipo de mídia. Como a indicação do que é relevante depende das entidades envolvidas numa interação, na AvanTV especificamos um conjunto de operações relacionadas às técnicas de adaptação de HA segundo a identificação de conjuntos de contexto determinados pelos modelos.

O método contemplado para realizar seleção e ordenação de itens com base na similaridade entre informações do usuário e do conteúdo é o de *filtragem baseado em conteúdo*. No entanto, devido a problemas de ambigüidade na correspondência textual (por exemplo, com palavras-chave), uma solução é fazer uso de termos de ontologias para representar estes perfis. Dado que o conteúdo é representado por um conjunto de conceitos da ontologia definidos pelos seus itens e o usuário, por uma lista ponderada de conceitos representando seus interesses, Shoal *et al.* (2008) propõe que a semelhança entre eles pode ser obtida se incluem conceitos iguais ou algum relacionamento hierárquico. Considerando três níveis a partir desta análise, os possíveis casos são:

- Correspondência perfeita (*P*): o mesmo conceito aparece em ambos os perfis.
- Correspondência forte (*F*): um termo aparece apenas em um perfil, enquanto um pai ou filho daquele vocábulo aparece no outro. Neste cenário, ocorrem duas situações: o interesse do usuário é mais específico ou mais geral que o item.
- Correspondência fraca (*R*): conceito aparece num perfil e um conceito avô ou neto aparece no outro. Também se subdivide nas situações de interesse do usuário mais específico ou mais geral que o item.

Pela atribuição de pontos (*scores*) a cada um desses casos, o índice de similaridade (*IS*) é medido como segue:

$$IS = \frac{\sum_{i \in Z} N_i \cdot S_i}{\sum_{j \in U} N_j}$$

Onde:

Z – número de conceitos na representação de itens

U – número de conceitos na representação de usuário

i – índice dos conceitos na lista de itens

j – índice dos conceitos na lista de usuário

S_i – pontuação para cada tipo de equivalência (*P*), (*F*) ou (*R*) entre o conceito *i* ao respectivo conceito no perfil de usuário (*S_i* = 0, caso não haja relação)

N_j – contagem de acessos pelo usuário ao conceito *j* (usado no cálculo do peso *w*)

Assumindo a seguinte pontuação para cada caso: **(P) = 1**; **(F)_b=2/5** (conceito do usuário mais específico); **(F)_c=2/3** (conceito do usuário mais geral); **(R)_d=1/3** (conceito do usuário muito mais específico); e **(R)_e =1/2** (conceito do usuário muito mais geral). O algoritmo a seguir mostra o processo para calcular a similaridade entre um perfil de item e um de usuário:


```

Matching (Item-List, User-List) {
Score = 0
Repeat for each I-concept in Item-List {
    Do case {
        - if I-concept in User-List then Score= ++ 1*w // caso (P)
        - if parent of I-concept in User-List then Score= ++2/3*w //caso (F)_c
        - if grandparent of I-concept in User-List then Score= ++1/2*w //(R)_e
    }
}
Repeat for each U-concept in User-List {
    Do case { // não é preciso checar correspondência perfeita novamente
        - if parent of U-concept in Item-List then Score= ++2/5*w //caso (F)_b
        - if grandparent of U-concept in Item-List then Score= ++1/3*w //(R)_d
    }
}
}

```

As pontuações assumidas são dadas como exemplos, para aprimorar o desempenho nem todos os conceitos do usuário devem participar da computação, apenas os mais relevantes. Para ampliar a expressividade deste algoritmo, o mesmo foi estendido para conceber um caso de correspondência semântica e é descrito no capítulo V, quando apresentamos o uso de contexto por aplicações segundo a abordagem proposta.

2.5 Considerações Finais

Neste capítulo foi abordada a fundamentação teórica relacionada ao tema desta dissertação. Foram descritos os sistemas e desafios de interatividade da TVDI, além de apresentados padrões e ferramentas para representação de conhecimento. A seguir, foi discutida a importância do paradigma de sensibilidade ao contexto no

projeto de aplicações que oferecem serviços relevantes. Por fim, técnicas de adaptação foram apresentadas considerando aspectos de contexto.

Este trabalho busca aperfeiçoar serviços interativos oferecidos na TV pela personalização do conteúdo aos interesses do usuário. A abordagem AvanTV proposta considera informações de contexto na concepção de aplicações segundo vinculação semântica aos dados. A representação de informações particulares do conteúdo, do perfil do usuário e descrição de adaptações é feita, principalmente, por ontologias. Como o escopo compreendido se refere à transmissão de conteúdo para interação na TV, são reusados termos das ontologias TVD e FOAF, bem como parte dos metadados *Dublin Core* e do padrão MPEG-7.

O MPEG-7 foi preferível aos demais metadados utilizados na TV devido ao conjunto de descritores designados para manipular características da interação do usuário e do conteúdo multimídia, assim como sua equivalência com esquemas de classificação do TV-Anytime e maior maturidade em relação ao MPEG-21.

Em decorrência do aumento de expressividade pela combinação dessas estratégias, os processo de aquisição e uso de contexto pelas aplicações são facilitados por mecanismos de avaliação de contexto e inferência de novos fatos expressos por regras que constituem as ontologias. Portanto, para adaptação do conteúdo de aplicações interativas contemplamos um conjunto de operações relacionadas às técnicas de adaptação de HA segundo a identificação de conjuntos de contexto determinados pelos modelos. O funcionamento destas operações é orientado pela extensão de um algoritmo de filtragem baseada em conteúdo conforme descrição de termos das ontologias.

Capítulo III.

TRABALHOS RELACIONADOS

A pesar das diversas pesquisas em sistemas sensíveis ao contexto, foi identificada a necessidade de avanços na experimentação de informações contextuais para aperfeiçoar serviços. Este capítulo descreve alguns esforços desenvolvidos que utilizam a sensibilidade ao contexto para facilitar a interatividade com aplicações na TV. A seção 3.1 descreve o sistema *ProfileTV*. A seção 3.2 apresenta o sistema colaborativo *CollaboraTVware*. A seção 3.3 descreve um *framework* para customização de vídeo esportivo. A seção 3.4 delinea a infraestrutura de suporte a recomendação *PersonalTVware*. Por fim, a seção 3.5 apresenta uma análise comparativa entre os trabalhos elencados.

3.1 ProfileTV

Motivado, principalmente, por problemas quanto à escalabilidade, segurança e portabilidade dos dados capturados da interação entre usuário, dispositivo e serviços, Coelho (2008) propõe um sistema para gerenciamento de perfis em TVDI, denominado *ProfileTV*. Dentre outros objetivos, ele permite a definição e atualização de contexto para cada interação entre telespectador e serviço interativo.

As principais funcionalidades fornecidas consistem na criação, atualização, sincronização e importação/exportação de perfis associados a usuários, dispositivos e serviços. O gerenciamento é realizado remotamente para atender serviços de clientes *web* ou embarcado em dispositivos interativos com *middleware* MHP.

Para configuração do sistema, é feita a definição de contexto através de arquivos XML, de modo que o administrador do sistema deve descrever as informações relevantes para cada tipo de interação a coletar dados. A gramática dos documentos é definida por categorias e propriedades que representam dispositivos,

serviços e características pessoais dos usuários, conforme ilustra a Figura 3.1 que mostra os dados a serem coletados nos serviços interativos experimentados.

Figura 3.1: Definição contextual dos serviços interativos testados no *ProfileTV*
(Fonte: COELHO, 2008)

```

110 <category id="PAN" name="Pan-American Games" parent="SERVICES">
111   <property id="PAN-AREA" name="area" type="MULTI"
112     valueType="TEXT" pattern="mural|medalhas|sobrePan|agenda|noticias" />
113   <property id="PAN-ELAPSED-TIME" name="elapsed_time" type="MULTI"
114     valueType="TEXT" pattern="mural|medalhas|sobrePan|agenda|noticias"
115     related="PAN-AREA" />
116   <property id="PAN-Sport" canBeNull="FALSE" name="sport" type="SINGLE"
117     valueType="TEXT">
118     <value>Futebol</value>
119   </property>
120 </category>
121 <category id="CCB" name="Campeonato Brasileiro" parent="SERVICES">
122   <property id="CCB-AREA" name="area" type="MULTI"
123     valueType="TEXT"
124     pattern="artilharia|tabela|campeoes|classificacao|estadio|estatisticas" />
125   <property id="CCB-ELAPSED-TIME" name="elapsed_time" type="MULTI"
126     valueType="TEXT" pattern="artilharia|tabela|campeoes|classificacao|estadio|estatisticas"
127     related="CCB-AREA" />
128   <property id="CCB-Club" name="Clube" type="SINGLE"
129     valueType="TEXT">
130     <value>SPORT</value>
131   </property>
132 </category>
133 <category id="PG" name="Portal da Globo" parent="SERVICES">
134   <property id="PG-AREA" name="area" type="MULTI"
135     valueType="TEXT"
136     pattern="jornalismo|esportes|entretenimento" />
137   <property id="PG-ELAPSED-TIME" name="elapsed_time" type="MULTI"
138     valueType="TEXT" pattern="jornalismo|esportes|entretenimento"
139     related="PG-AREA" />
140 </category>

```

Apesar da ampla aceitação do uso de linguagem de marcação, seu uso exclusivo é restritivo quanto à capacidade de completude e desambiguação das informações, conforme pode ser observado pela descrição da propriedade *area* distinguida apenas pelo identificador da categoria e padrão de uso da mesma.

Pelo monitoramento das interações do usuário são capturadas as características de cada categoria de perfil e mantidas localmente ou exportadas a um servidor para diferentes propósitos (ex: entrega de conteúdo personalizado e gerenciamento de conta pelo usuário). O mapeamento específico às informações contextuais é utilizado para aperfeiçoar a apresentação por parte das aplicações, como o caso de identificação da área preferida do usuário na aplicação sobre informações dos jogos do *Pan-Americano* conforme tempo gasto de visualização.

3.2 CollaboraTVware

A dificuldade no acesso transparente ao crescente volume e diversidade de conteúdo na TVDI é a principal motivação para o sistema *CollaboraTVware* (ALVES, 2008), que provê orientação na escolha de programas e serviços interativos pelos usuários a partir da participação colaborativa de outros usuários com perfis e contextos similares.

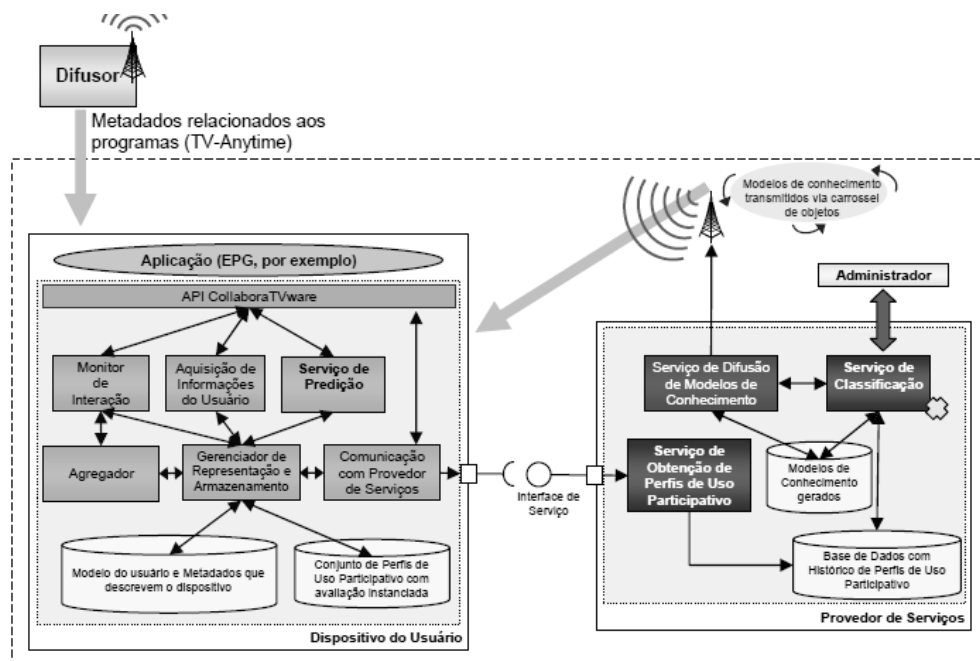
A representação do modelo de usuário e de informações contextuais é feita por esquemas XML derivados dos padrões de metadados da TVDI, além do padrão MPEG-21 para descrição de dispositivos. As entidades contextuais consideradas foram o usuário, o dispositivo e o programa em função das dimensões identidade, localização e tempo. A sensibilidade ao contexto é utilizada para orientar os usuários na escolha de programas e serviços interativos.

As similaridades de perfis e contextos entre usuários são definidas por técnicas de mineração de dados, responsáveis pela geração de um modelo de conhecimento a partir da avaliação de um determinado programa. Os perfis de uso participativo com avaliações atribuídas pelo usuário são enviados por canal de retorno ao provedor de serviço para realizar a etapa de classificação, conforme apresentado pelos subsistemas envolvidos na figura 3.2.

A descoberta da avaliação para um determinado usuário é orientada pelo processo de predição, implantado no dispositivo de recepção do usuário, sobre o modelo de conhecimento difundido.

A construção de modelos de conhecimento depende da interação do usuário que atribui avaliação ao programa, para orientar outros usuários em contextos semelhantes, porém não considera técnicas de filtragem de conteúdo ou mecanismos de adaptação associados ao contexto do usuário. Não faz parte do escopo a proposição de tecnologias para aquisição e interpretação destas informações, cujas funcionalidades são simuladas por arquivos de configuração.

Figura 3.2: Arquitetura do *CollaboraTVware*. (Fonte: ALVES, 2008)



3.3 Framework para Customização de Vídeos Esportivos Personalizados

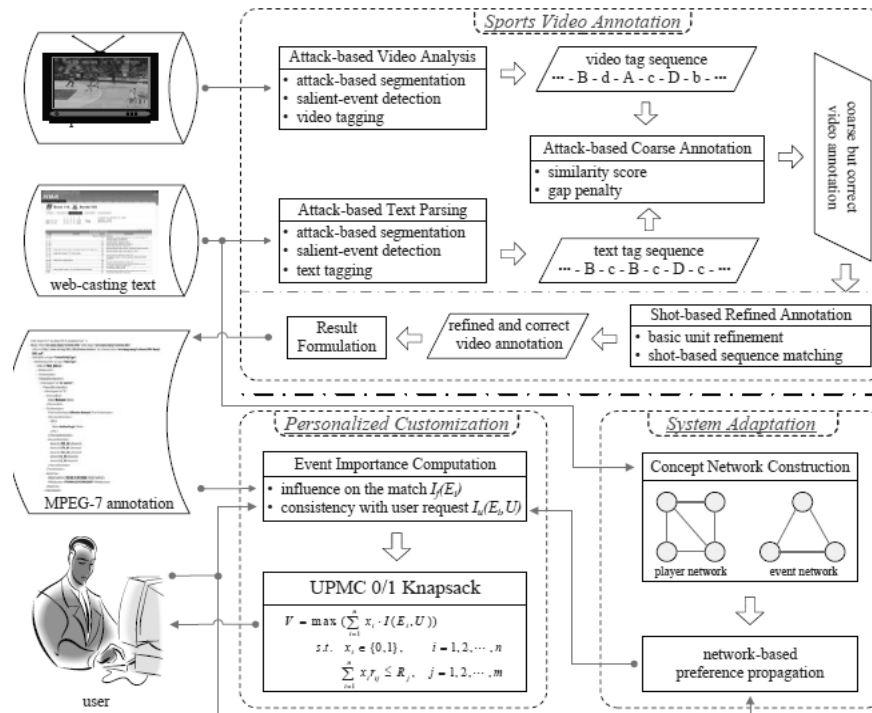
O *framework* para personalização de vídeo esportivo proposto por Liang *et al.* (2009), em que os usuários podem utilizar segmentos de vídeo refinados contendo suas preferências (jogador, time, etc.) acerca das partidas esportivas, endereça questões de anotação semântica de vídeo, recuperação e resumo de vídeo personalizado, além de adaptação de sistema.

Uma abordagem de correspondência semântica pelas seqüências temporais associa o conteúdo do vídeo a textos de *web-casting* de modo a gerar anotação detalhada de vídeo, armazenada no formato do padrão MPEG-7, incluindo os jogadores envolvidos, tipos de eventos, pontuação resultante e o segmento de vídeo correspondente, segundo ilustra a Figura 3.3.

O algoritmo para customização de conteúdo de vídeo ajusta as preferências do usuário às limitações do ambiente de maneira a selecionar os segmentos de vídeo com conteúdo mais importante. A aprendizagem implícita sobre as escolhas do

usuário é obtida com o uso contínuo, para tanto, é proposta uma abordagem baseada em rede social de semânticas de vídeo para identificar as preferências ocultas do usuário sem interações adicionais.

Figura 3.3: Framework para customização de vídeos esportivos personalizados (Fonte: LIANG *et al.*, 2009)



Anotações semânticas de vídeo são realizadas por Rede Bayesiana e algoritmos de equivalência aplicados para alinhar sequência de vídeo e texto. A partir da definição de duas redes de conceitos, para jogadores e eventos, são identificados conceitos do contexto mútuo da partida e descritores do grau de interesse do usuário por um determinado conceito da partida. Assim, a preferência do usuário é inferida e ajusta sua estrutura e/ou parâmetros de forma mais conveniente para uso futuro. A personalização de vídeo foca em seleção de conteúdo, sendo orientadas adaptações ao canal de comunicação e exibição em dispositivos. Os componentes do sistema possuem alto acoplamento que dificulta a reutilização com outros mecanismos de raciocínio e não são aplicados diretamente no meio televisivo.

3.4 PersonalTVware

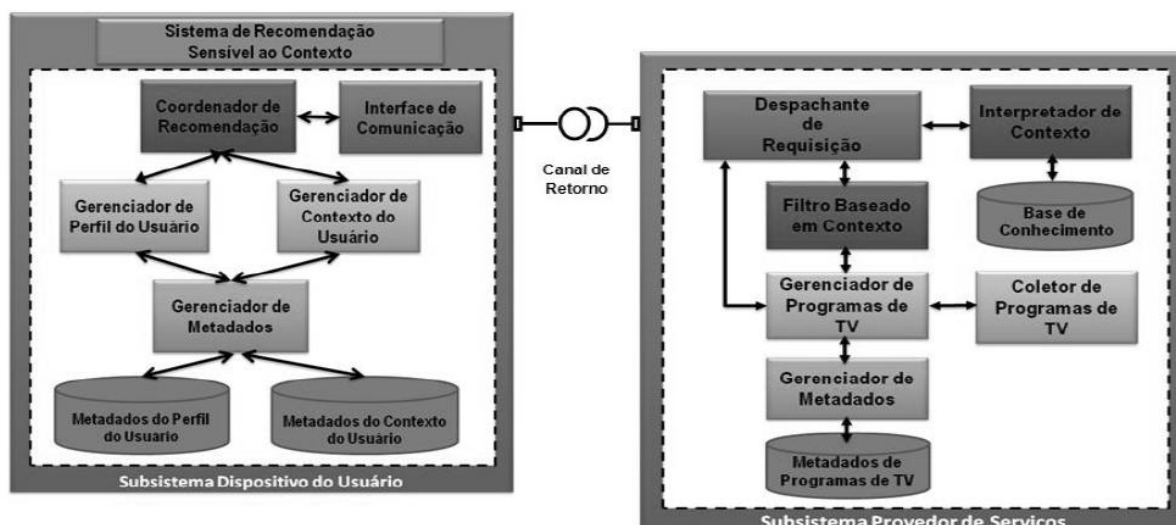
Silva *et al.* (2009) propõe uma infraestrutura para suporte a recomendação personalizada de conteúdo para TV Digital, denominada *PersonalTVware*, de forma a facilitar a busca de conteúdo de interesse do usuário em meio ao alto volume de conteúdo oferecido. Conforme ilustrada a Figura 3.4, a mesma é distribuída entre o subsistema do usuário (dispositivo contendo um *middleware* de recepção de sinal digital) e o provedor de serviços, e utiliza *Web Services* para comunicação bidirecional sobre o canal de interatividade.

As informações de contexto do usuário, tais como identificação, dia, hora, local e origem do programa de TV, adquiridas de forma implícita são representadas por documentos XML, enquanto os dados do perfil fornecidos pelo usuário são descritos por metadados do padrão TV-Anytime. O acesso a estas informações é responsabilidade dos gerenciadores de contexto e de perfil do usuário, respectivamente.

A partir das informações de contexto atuais e passadas, o interpretador realiza a inferência de preferências implícitas por canais e gêneros com base nos métodos de aprendizagem de máquina por raciocínio baseado em casos, árvores de decisão, rede bayesiana e redes neurais. As recomendações são feitas pela filtragem de conteúdo baseado em contexto, localizado no provedor de serviços, que explora as variáveis contextuais e as descrições dos conteúdos dos programas de TV, assim como a avaliação de relevância dada pelo usuário.

O projeto do sistema é constituído de um subsistema no dispositivo do usuário e outro num provedor de serviços, ambos implementados em Java. Os módulos do dispositivo do usuário são responsáveis por gerenciar o perfil e contexto do usuário, manutenção de metadados e controle de recomendações. Para o provedor de serviço, foram projetados componentes para interpretação de contexto, controle de requisições e gerência de informações sobre programas de TV.

Figura 3.4: Infraestrutura *PersonalTVware* (Fonte: SILVA *et al.*, 2009) SILVA, F. S.; ALVES, L. G. P.; BRESSAN, G. (2009)



Para validação foi desenvolvido um aplicativo *Xlet* que utiliza a API desenvolvida para prover recomendações de programas de TV. Apesar de considerar o armazenamento do perfil do usuário no próprio dispositivo, informações de contexto são transmitidas para processamento no provedor de serviços. Segundo experimentos realizados, os melhores resultados quanto a qualidade das recomendações foram observados pelo método de raciocínio baseado em casos.

3.5 Análise Comparativa

Os trabalhos apresentados foram comparados conforme característica para desenvolvimento de aplicações sensíveis ao contexto. Os principais aspectos observados foram:

1. Especificação de informações de contexto: para determinar o comportamento contextual a ser executado pelas aplicações, é necessário prover os tipos de contexto requeridos;
2. Monitoração e análise do perfil do usuário: conhecimento sobre o usuário é indispensável para aplicativos interativos de TV que pretendem realizar algum tipo de adaptação;

3. Aquisição de contexto: para perceber alterações no contexto, as informações precisam ser monitoradas e capturadas de fontes diversas;
4. Capacidade em inferir novas informações: dado o modo de interatividade inerente à TV, é desejável oferecer mecanismos que ampliem o conhecimento a partir das informações já obtidas.
5. Disponibilização e uso de contexto: meios para disponibilizar ou executar comportamento sensível ao contexto às aplicações.

A Tabela 3.1 apresenta as estratégias utilizadas por cada um dos trabalhos discutidos conforme características elencadas anteriormente. Os campos sem comentários indicam que o sistema não apresenta a funcionalidade correspondente.

Tabela 3.1: Comparação entre sistemas analisados

	1	2	3	4	5
<i>ProfileTV</i>	Arquivos XML	Captura da interação e transformação em perfil	Obtenção por perfis pré-definidos	-	Apóia na personalização das interações do telespectador
<i>CollaboraTVware</i>	Esquemas de marcação (metadados)	Avaliação atribuída pelo usuário, identificado por simulação	-	Técnicas de mineração de dados para predição da avaliação de programas	Auxilia na escolha de programas e serviços interativo
<i>Framework</i> para customização de vídeos esportivos	Esquemas de marcação (metadados)	Preferências do usuário são apreendidas das interações com vídeos	Aquisição por segmentos de vídeos e anotação textual	Classificador Bayesiano para extração e anotação de vídeo	Fornecer conteúdo e seleção de segmentos de vídeo adaptado
<i>PersonalTVware</i>	Arquivos XML	Observação de consumo de programa de TV e <i>feedback</i> explícito	Aquisição por componentes distribuídos	Métodos de aprendizagem para inferir preferências por gêneros	Uso para recomendação de programas de TV

Em geral, os modelos de representação contextual utilizam linguagens de marcação para descrever os tipos e informações de contexto. Esta opção é

justificada, principalmente, pelo emprego de metadados intrínsecos ao meio, como descritores do padrão MPEG-7. As soluções para TV requerem disponibilidade de canal de interatividade para efetivar o processamento de contexto.

Quando realizada, a captura de contexto ocorre diretamente associada às fontes que especificam cada tipo de informação. Enquanto os mecanismos de inferência de contexto empregados são diferenciados para cada situação. Tanto o *CollaboraTVware* quanto *PersonalTVware* utilizam métodos de aprendizagem, porém com técnicas distintas. No *framework* para customização de vídeos esportivos, são extraídos termos para anotação semântica baseado em redes *Bayesianas*. Técnicas de inferência estão fora do escopo do *ProfileTV*.

Outra consideração importante é com relação ao uso de contexto nas propostas que se referem à recomendação de serviços, personalização da apresentação ou avaliação de contexto.

3.6 Considerações Finais

Neste capítulo, foram descritos trabalhos que utilizam sensibilidade ao contexto para aperfeiçoar serviços. Uma análise comparativa foi realizada para verificar as características atendidas por cada sistema. Exceto pelo *framework* apresentado na seção 3.3, todos compartilham do objetivo de apoiar as interações na TVDI.

Na abordagem AvanTV, por outro lado, as informações de contexto são especificadas através da combinação de metadados do padrão MPEG-7 e ontologias de domínio, de modo a ponderar a capacidade de intercâmbio de dados à habilidade em prover interpretação e dedução de novos conhecimentos.

Assim como a maior parte dos trabalhos analisados, esta proposta realiza monitoração das interações do usuário para, juntamente com as preferências apreendidas, descrever o contexto com base na extração de termos semânticos. Para uso de contexto, é explorada a lacuna quanto à utilização de técnicas de adaptação e, conseqüente, personalização do conteúdo de aplicações. Para tanto, é necessário entendimento do significado dos dados originais de modo a ajustá-los

pela inclusão, supressão ou modificação do seu conteúdo. Neste sentido, convém a análise dos mecanismos de construção de redes semânticas sobre o conteúdo do vídeo, propostos por Liang *et al.* (2009), que neste trabalho, de maneira análoga, é definido para o conteúdo das aplicações e suportados por ontologias.

No capítulo IV serão apresentados os tipos de informações e respectivas representações para definição do modelo contextual. A seguir, o capítulo V descreve como essas informações foram utilizadas para fornecer personalização do conteúdo de aplicações de TVDI. Para isso, foram desenvolvidas funcionalidades que capacitam aplicações interativas de sensibilidade ao contexto em dispositivos com *middleware* Ginga.

Capítulo IV.

INTEGRANDO MODELO CONTEXTUAL E CONTEÚDO SEMÂNTICO

Descrevemos a importância em utilizar contexto para fornecer informações mais relevantes aos usuários. Além disso, observamos que o compartilhamento de informações contextuais requer formas padronizadas de representação. Quando conduzida implicitamente pelo sistema, adaptações do conteúdo são orientadas pelas representações das necessidades do usuário, das informações específicas do domínio e da descrição de como realizar tais adaptações.

Este capítulo apresenta os tipos de informações e respectivas representações para definição do modelo de contexto proposto pela abordagem AvanTV. A seção 4.1 introduz a metodologia adotada e os requisitos para realizar adaptações. A seção 4.2 descreve o Modelo de Domínio (MD), que inclui conceitos e relacionamentos associados, principalmente, ao gênero esportivo. Na seção 4.3 é apresentado o Modelo de Usuário (MU) como uma extensão dos padrões de metadados utilizados na TV e relacionados às informações de contexto. Na seção 4.4 é descrita a especificação de contexto representada por ontologias que possibilita ajustar informações aos interesses e contexto do usuário. A seção 4.5 apresenta mecanismos utilizados para inferência de contexto. Uma avaliação das ontologias utilizadas é delineada na seção 4.6. Por fim, a seção 4.7 relaciona algumas considerações quanto a abordagem proposta.

4.1 Introdução

Inicialmente, a abordagem AvanTV relaciona atividades para suporte à adaptação de serviços de TVDI através de um modelo baseado em metadados e ontologia de domínio. Posteriormente, são descritos os elementos necessários ao tratamento de comportamento sensível ao contexto. A modelagem é baseada em

conceitos relacionados à manipulação de contexto proposta por Vieira (2008) e nas dimensões semânticas de informação contextual apresentadas na seção 2.3.

Como proposto por Noy e McGuinness (2001), foi explorado o reuso de ontologias existentes na construção dos modelos. Os metadados e ontologias, descritos no capítulo II, relacionados neste trabalho são: de TV Digital (TVD), desenvolvida em OWL por Leite *et al.* (2007); de pessoas e relacionamentos interpessoais FOAF; vocabulários do padrão *Dublin Core*, para descrição de coleções de recursos; e do padrão MPEG-7, para descrição de preferências e interações do usuário, bem como do conteúdo das aplicações.

Alguns requisitos relacionados à utilização de informações de contexto para adaptação das aplicações são: criação e identificação de usuários e grupos; adição de preferências; manutenção do histórico de uso; e atribuição de contexto às interações. Quando uma aplicação requer personalização, ela pode fazê-lo pela utilização de operações de adaptação, por registro de serviços para monitoramento de interações ou pela determinação de regras sobre o modelo. Tais funcionalidades requerem delimitação do escopo influenciado por contexto ao domínio de conhecimento, tratado pela definição de terminologias para compartilhamento de significado. Neste caso, propomos uma representação para conteúdo esportivo. Analogamente à definição dada por Dey *et al.* (1999), para a difusão de conteúdo esportivo consideramos contexto como *qualquer informação que pode ser usada para caracterizar a situação de uma entidade em uma aplicação esportiva, onde uma entidade é um telespectador, a programação televisiva, recursos e o próprio conteúdo, considerados relevantes para as interações com a aplicação.*

4.2 Modelo de Domínio

A interpretação acerca das condições que definem a situação em que as interações são realizadas é definida pelo domínio da aplicação. De modo a identificar o vocabulário empregado nos interesses do usuário e gerar novas afirmações, é necessário construir modelos de propósito específicos, enquanto o gerenciamento continua a ser tratado de forma genérica.

Seguindo a hierarquia proposta por Wu (2002), tal informação é descrita em termos da granularidade de *itens*, *conteúdo* e *componentes compostos*. Os ***itens*** (ou *fragmentos*) constituem conceitos atômicos, que não podem ser modificados, mas fornecem primitivas do modelo. O ***conteúdo*** (ou *páginas*) consiste no agrupamento de itens que podem ser incluídos ou omitidos, e ter sua disposição alternada. ***Componentes compostos*** agregam diversos conteúdos e/ou partes menores de outros componentes de modo a combinar elementos sobre o mesmo assunto. A seguir, apresentamos a semântica de conteúdo esportivo para serviços de TVDI.

4.2.1 Escopo

Apesar das inúmeras possibilidades de mídia para acesso ao conteúdo audiovisual, os eventos esportivos continuam sendo consumidos, preferencialmente, pela TV. Durante a Copa do Mundo de Futebol 2010, por exemplo, houve redução proporcional em consultas na web durante as partidas de cada país, enquanto ocorreu crescimento expressivo da audiência televisiva (OLDHAN e SNEDEGAR, 2010; IBOPE, 2010).

Os eventos esportivos são caracterizados pela competitividade e grande quantidade de dados que podem ser difundidos aos usuários. Em sistemas de TVDI os produtores podem inserir conteúdo durante a exibição de programas. Dentre os serviços oferecidos, destacam-se a apresentação de informações referentes aos resultados, tabelas de jogos, classificações e estatísticas de competições, além de noticiários e identificação de expressões específicas a cada modalidade (por exemplo, a marcação de um gol no futebol). As transmissões geralmente atraem anunciantes e uma proeminente receita publicitária, que utiliza diferentes estratégias de publicidade, considerando local da competição, uniformes dos atletas, posicionamento do(s) equipamento(s), local destinado ao público e demais atuantes.

A ocorrência dessas informações, muitas vezes, caracteriza o interesse da audiência e requer mecanismos capazes de representá-las. Questões referentes a como realçar o favoritismo por times, atletas ou competições, além de beneficiar a exibição de anúncios mais adequada às características de cada segmento na

difusão esportiva, são endereçadas por aplicações interativas. Informativos, tais como boletins, notícias e comentários, ofertados como serviços interativos permitem ao usuário acessar, a qualquer instante, estatísticas do jogo, perfil dos jogadores, normas da modalidade, áudio do hino etc. Aplicações de entretenimento, como jogos, animações e serviços de participação, estimulam a interatividade e produzem alto grau de imersão, a exemplo de serviços de aposta que registram palpites dos telespectadores sobre resultados de uma partida. Ainda, a publicidade interativa, que inclui o *t-commerce*¹², permite, entre outras ações, a compra imediata de produtos veiculados na TV e anúncios baseados nas preferências do usuário.

4.2.2 Ontologia *SPTC Semantic*

Neste trabalho, as informações específicas ao domínio são representadas por uma estrutura sintática acerca do conteúdo e semântica dos conceitos. O modelo de dados SPTC (*Sporting Content*) proposto explora conceitos dos esportes para promover a interoperabilidade de informações esportivas entre sistemas, normalmente associados à difusão de conteúdo, tal como relatado em Nascimento e Bezerra (2011a). Provê, assim, validação computacional de documentos relacionados a objetos de interesse do usuário em aplicações esportivas e trata, uniformemente, suas características publicitárias. Objetos de interesse são constituídos por elementos preferenciais, tais como competidores (times, atletas, etc.), eventos e serviços esportivos.

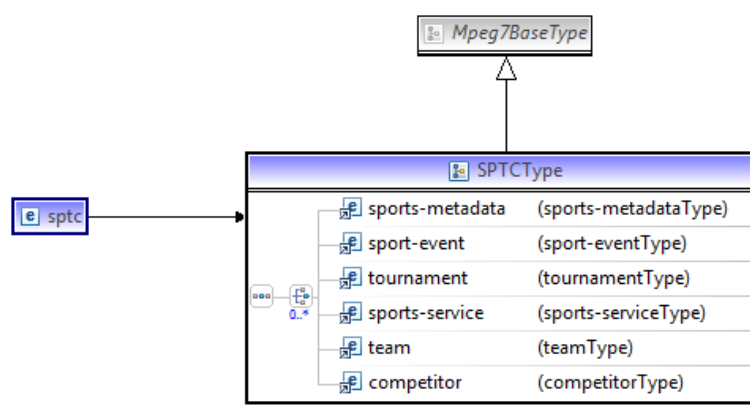
Esses eventos são caracterizados pela disputa entre competidores, através dos respectivos times e/ou atletas, delimitadas temporalmente e realizadas em instalações, por exemplo, estádios, ginásios ou arenas. Eventos esportivos podem ser agrupados em competições organizadas, por sua vez, em períodos, chaves e/ou etapas, a exemplo dos grupos na Copa do Mundo de Futebol e rodadas do Campeonato Brasileiro de Futebol. Em geral, as competições culminam com premiações e títulos aos vitoriosos.

¹² O serviço *T-Commerce* se refere ao comércio eletrônico na TVDI.

Diferentemente de propostas semelhantes (como o *SportML*¹³), os dados descritos no SPTC são combinados à especificação semântica do contexto do usuário. O conteúdo esportivo é representado por documentos XML, definidos por um esquema de dados que estende o padrão MPEG-7, enquanto os itens que caracterizam os objetos de interesse são representados por termos da ontologia de esportes, descrita em OWL.

O elemento raiz `sptc` do esquema XML é definido pelo tipo `SPTCType`, ilustrado na Figura 4.1, uma extensão do tipo `Mpeg7BaseType` do padrão MPEG-7. Este tipo é constituído por uma sequência alternativa dos objetos de interesse: eventos esportivos (`sport-event`), competições (`tournament`), metadados (`sports-metadata`), serviços a serem processados (`sports-service`) ou descrição de competidores (`team` ou `competitor`). A identificação e referência a objetos do modelo utilizam URI, que podem endereçar termos da ontologia.

Figura 4.1: Representação do Conteúdo Esportivo (SPTC)

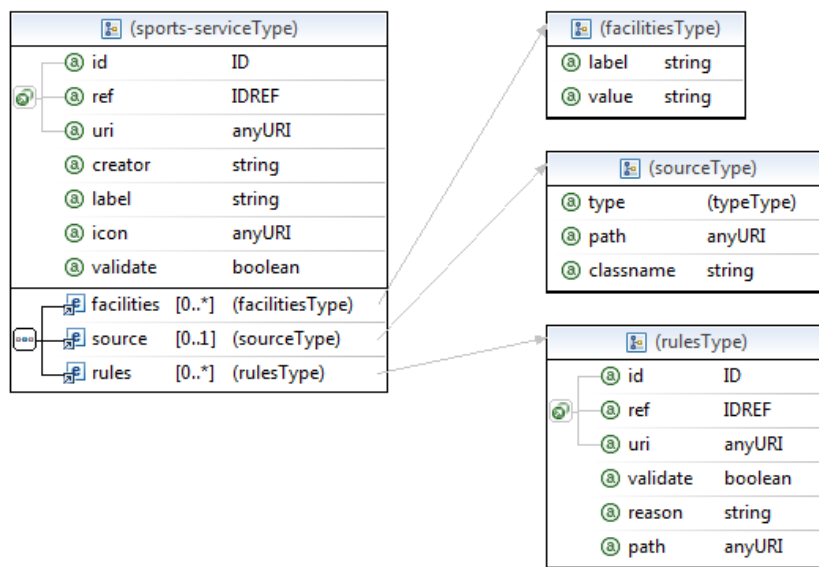


Prover metadados aos serviços esportivos atende à necessidade das aplicações em especificar as informações contextuais de maior importância e métodos de reagir a elas. Portanto, configurações acerca dos atributos de contexto, fontes para aquisição de informação contextual e conjunto de regras que indicam as condições e ações a serem satisfeitas são descritas pelo tipo `sports-serviceType`, apresentado na Figura 4.2.

¹³ IPTC Council: <http://www.iptc.org/cms/site/index.html?channel=CH0105>

São contemplados, ainda, descritores extensíveis e inerentes às modalidades para estatísticas (*event-stats* e *competitor-stats*, por exemplo), ações defensivas e ofensivas (*event-actions-soccer*, por exemplo), publicidade de patrocinadores (*sponsor*), dentre outros.

Figura 4.2: Descrição de serviços esportivos no modelo SPTC



O gênero esportivo tem sido tratado segundo esforços independentes para descrever eventos ao vivo. No entanto, poucos têm sido direcionados à especificação semântica. Nesse sentido, a BBC (BBC, 2011) iniciou recentemente a proposição de uma ontologia em RDF para esportes, centrada na cobertura de eventos competitivos.

Propomos a concepção da ontologia *SPTC Semantic*¹⁴ que mapeia os principais conceitos da difusão de conteúdo esportivo em itens que constituem objetos de interesse do usuário. As definições de aplicação e programa televisivo da ontologia *TVD* foram estendidas para definir o conteúdo, de maneira geral, como restrição à propriedade que agrega fluxos elementares.

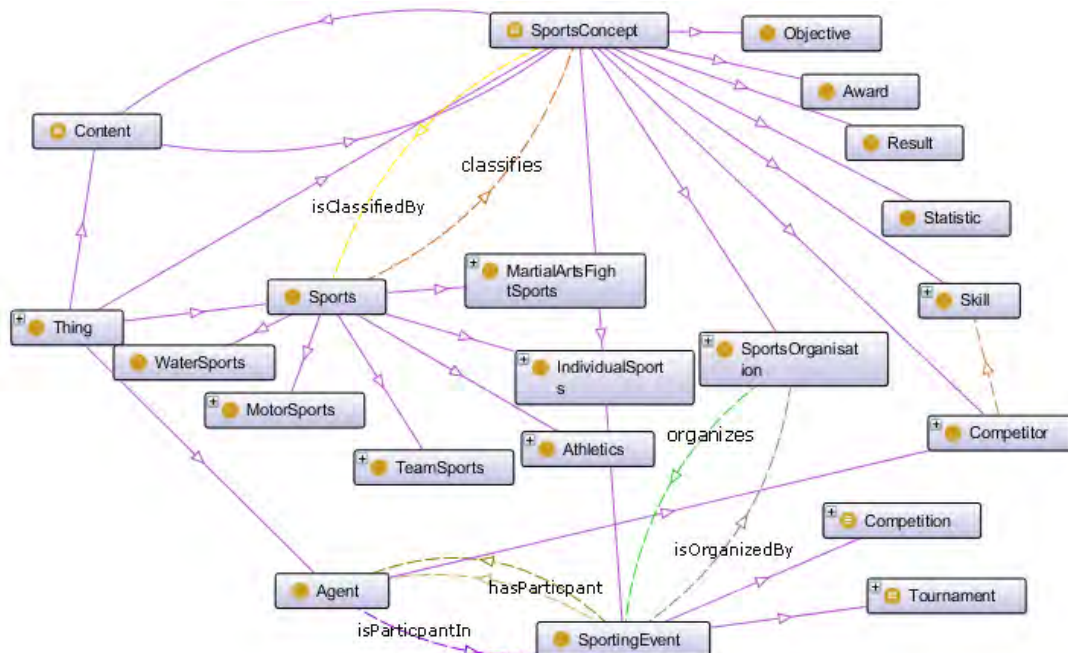
¹⁴ Disponível em <http://code.google.com/p/sport-ontology/>

A ontologia *SPTC Semantic*, identificada pela URI (<urn://avantv/2010/sptcsem>), é constituída de 39 classes e 29 propriedades, resumidas na Figura 4.3. Ela é organizada da seguinte forma:

- Todos os termos são subclasses de conceito esportivo (*SportsConcept*);
- *SportsConcept* é uma subclasse de conteúdo (*Content*) que designa o agrupamento de itens (ou termos do vocabulário);
- A classe *Content* provem da ontologia de Adaptação Contextual (seção 4.4) e possui relação com a definição de componentes das classes Programa Televisivo (*TelevisionProgram*) e Aplicação (*Application*) da ontologia TVD;
- O termo *Agent* advém da ontologia FOAF para definir pessoas do modelo, participantes em eventos esportivos (por exemplo, *Athlete* ou *Team*, subclasses de *Competitor*);
- A classe esportes (*Sports*) classifica os termos do vocabulário segundo as categorias, por exemplo, Atletismo (*Athletics*), Artes Marciais e Lutas (*MartialArtsFightSports*), Esportes Individuais (*IndividualSports*) e Coletivos (*TeamSports*).
- As principais propriedades são: classificação de conceitos (*classifies/isClassifiedBy*), participação (*hasParticipant/isParticipantIn*) e organização de eventos (*organizes/isOrganizedBy*).

Primeiramente, o projeto da ontologia de esportes foi desenvolvido puramente em RDF, no entanto, a definição de algumas restrições, tais como a cardinalidade de propriedades e operações de agregação sobre classes, demandou o uso do formalismo da OWL para sua concepção.

Figura 4.3: Ontologia de esportes (SPTC Semantic)



4.3 Modelo de Usuário

Algumas diferenças entre usuários devem influenciar na utilidade individual do serviço fornecido. A preocupação em definir como a informação do usuário pode ser obtida por sistemas automatizados e utilizada para melhorar o desempenho dos mesmos é intuito da Modelagem de Usuário. A obtenção destas informações pode ocorrer de maneira direta, quando expressas explicitamente pelo usuário, por exemplo, através de diálogos ou formulários; ou indireta, quando extraídas de maneira implícita, por exemplo, através do comportamento histórico ou presença do usuário num ambiente (CHEN, 2004; CESAR e CHORIANOPOULOS, 2008).

Conforme discutido na seção 2.2.3, é necessário conhecimento acerca das características dos telespectadores para prover personalização. Esse conhecimento é caracterizado tanto por modelos individuais (BUCZAK *et al.*, 2002; THAWANI *et al.*, 2004) quanto por classes de características predefinidas (GOREN-BAR e GLINANSKY, 2002; ARDISSONO *et al.*, 2004). Logo, a adaptação do conteúdo de aplicações de TVDI deve prover um Modelo de Usuário (MU) particular que considere atividades conjuntas e que explore, além dos dados pessoais, as

preferências sobre conceitos do modelo de domínio e a relação com fatores do ambiente. As informações obtidas precisam ser representadas de modo a serem compartilhadas. Neste caso, a sobreposição entre conceitos presentes no MU e no MD relaciona interesses do perfil do usuário que, normalmente, são associados a atributos indicando sua relevância. Por exemplo, o uso de valores booleanos pode denotar que um conceito é importante ou não para o usuário. Ou ainda, de maneira mais significativa, pela atribuição de valores que variam numericamente, como um real entre 0 e 1, ou em classes, tais como ‘Ruim’, ‘Bom’ e ‘Ótimo’.

A Tabela 4.1 apresenta os dados do telespectador contemplados nesta dissertação, respectivas técnicas utilizadas para aquisição e estratégia para representação do modelo. O MU proposto mantém descritores de metadados MPEG-7 MDS, além de um perfil com termos relacionados semanticamente às ontologias de domínio e de informações contextuais.

Tabela 4.1: Dados utilizadas no Modelo de Usuário

Dados utilizados	Técnica de Aquisição	Estratégia para Representação
Pessoais (nome, idade, sexo, contato, local, grupos)	Diretamente fornecida pelo usuário	MPEG-7 MDS
Preferências	Híbrida	MPEG-7 MDS e perfil do usuário sobre ontologias
Histórico de uso	Indiretamente pela interação do usuário	Arquivos intermediários e MPEG-7 MDS
Presença e assiduidade	Indiretamente por abstração de sensores	Perfil do usuário sobre ontologias

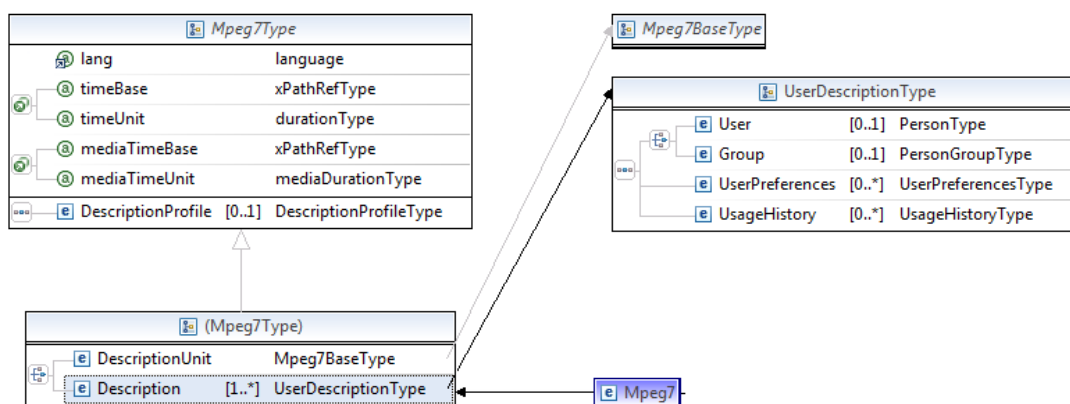
4.3.1 Representação do MU por Metadados MPEG-7

O esquema de descrição para Interação do Usuário (*User Interaction DS*) é um subconjunto dos esquemas MDS do MPEG-7 que descreve preferências do

usuário e histórico de uso sobre conteúdo multimídia (MARTINEZ, 2004). As preferências (*UserPreference DS*) são constituídas pelos diferentes tipos de conteúdo e modos de navegação, enquanto o histórico de uso (*UsageHistory DS*) é constituído por uma lista de ações do usuário dentro das aplicações.

A representação das informações do MU é definida por metadados do perfil *User Description Profile* (UDP) combinado a algumas restrições de complexidade. A primeira delas diz respeito a tornar explícita a escolha pelo tipo de agente do descritor de usuários *UserDescriptionType*, anteriormente realizada pela extensão do tipo abstrato *AgentType*. Assim, a distinção entre uma pessoa (*PersonType*) e um grupo de pessoas (*PersonGroupType*) passa a ser descrito como apresentado na Figura 4.4, mantendo a compatibilidade com ferramentas para validação de esquemas XML existentes.

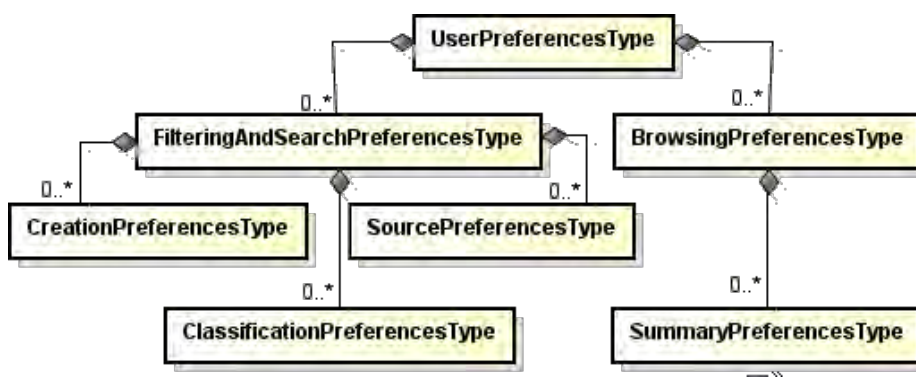
Figura 4.4: Elementos do *UserDescriptionType* com restrições de complexidade sobre o padrão MPEG-7



O *UserPreference DS* define as preferências do usuário segundo a criação, a classificação e uso de conteúdo (*FilteringAndSearchPreferences DS*) e a navegação sobre o mesmo (*BrowsingPreferences DS*). A estrutura foi modificada de modo a evitar recursividade na definição de preferências, aprimorando o desempenho das ferramentas de *parser* e mapeamento de termos. Um esquema de descrição de preferências simplificado é apresentado na Figura 4.5. O valor das preferências varia no intervalo $[-100, 100]$, onde zero representa um

valor neutro, valores abaixo indicam um grau negativo em relação a um conteúdo, e valores acima sugerem um grau positivo.

Figura 4.5: UserPreferences DS simplificado no MPEG-7



As ações de consumo pelo usuário são mantidas pelo descritor `UsageHistory` DS, composto por elementos `UserActionHistory` que descrevem um conjunto de elementos `UserActionList` constituído de objetos `UserAction`, segundo ilustra a Figura 4.6. Identificadores das aplicações e referências aos conteúdos associados a cada ação podem ser incluídos, assim como o período de observação e as recorrências da interação.

Figura 4.6: UsageHistory DS no MPEG-7



A partir do registro direto de interesses do usuário, tais como gênero e formato preferenciais, os valores são atualizados automaticamente pela análise da frequência de uso obtida no histórico. No entanto, a relação entre preferências do usuário e conteúdo se limita aos metadados oferecidos pelos esquemas de descrição, em sua maioria pela atribuição de palavras-chave.

Uma solução foi explorar os campos de referências URI para incluir conceitos do domínio. As correlações entre descritores do usuário e diferentes tipos de conteúdo são realizadas pelo método de filtragem baseada em conteúdo utilizando ontologias proposto por Shoval *et al.* (2008), a ser aplicada no Capítulo V. Ao contrário da tradicional filtragem baseada em conteúdo, em que a similaridade entre

o perfil do usuário e itens do conteúdo é obtida pela correspondência de termos de diferentes origens com sentidos variados, esta solução transpõe problemas de ambigüidade pela equivalência entre conceitos semanticamente relacionados.

4.4 Modelo de Adaptação

Os procedimentos que definem como aspectos do modelo de usuário e de domínio serão utilizados para oferecer funcionalidades adaptativas às aplicações são encontrados no Modelo de Adaptação. Estendemos esta definição incluindo a análise de comportamento sensível ao contexto e a indicação de como este comportamento deve ser executado.

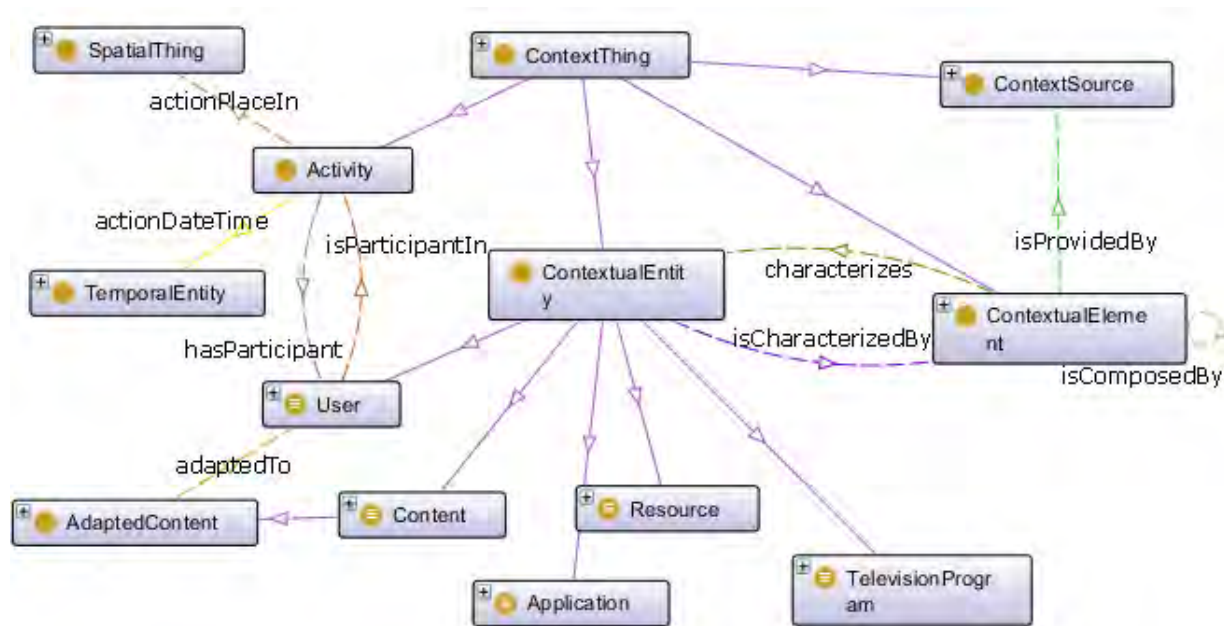
As estratégias de adaptação propostas são expressas segundo a representação *AvantvCTX* baseada em ontologia apresentada na Figura 4.7. A associação entre declarações em regras, quando satisfeitas, acionam comportamentos pré-definidos. Nascimento e Bezerra (2011b) relatam um conjunto de afirmações provenientes da convergência entre essa ontologia e conceitos dos modelos de usuário e domínio para construção de uma base de conhecimento. Assim, através de regras de inferência, é possível derivar novos fatos conforme condições vinculadas às informações de contexto.

As principais classes e relações da ontologia de adaptação contextual *AvantvCTX*, identificada por `urn://avantv/2010/ctx`, são organizadas da seguinte forma:

- *ContextThing*: representa os principais termos relacionados aos elementos estruturais de uma aplicação sensível ao contexto, tais como entidades (*ContextualEntity*), atributos (*ContextualElement*) e fontes (*ContextSource*) de contexto.
- *ContextualEntity*: representa as entidades consideradas relevantes para descrição e identificação de contexto. Nas interações com aplicações esportivas de TVDI são: usuário (*User*), programação da TV (*TelevisionProgram*), aplicação (*Application*), conteúdo

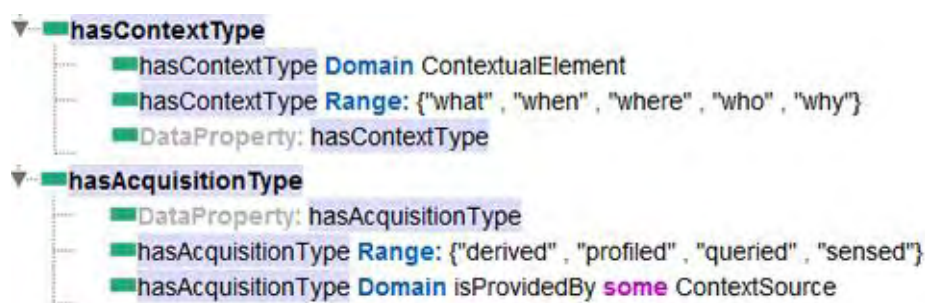
(Content) e recurso computacional (Resource). Cada contexto é constituído durante a execução pela relação de suas propriedades com os demais contextos.

Figura 4.7: Ontologia de Adaptação Contextual AvantvCTX



- **ContextualElement**: representa as propriedades (atributos e relacionamentos) que caracterizam (characterizes/isCharacterizedBy) uma entidade de contexto, tais como a *presença e preferências* de usuários ou *itens* do conteúdo. Descrevem a categoria de contexto a que se refere pelo relacionamento *hasContextType* (Figura 4.8) e podem ser compostos por outros elementos contextuais através do relacionamento *isComposedBy*.
- **ContextSource**: representa a origem dos valores de elementos contextuais, tais como sensores virtuais, agregadores de notícias e bases de dados externas. O método de aquisição é definido pelo atributo *hasAcquisitionType* (Figura 4.8) associado ao relacionamento *isProvidedBy*.

Figura 4.8: Definição das propriedades de dados *hasContextType* e *hasAcquisitionType*



Uma das principais dificuldades ao uso de contexto diz respeito à heterogeneidade e distribuição das fontes pelas quais se obtém contexto. Detectar a presença de pessoas num ambiente, por exemplo, pode requerer a combinação dos resultados de diversas técnicas tais como processamento de imagens, sensores de movimento etc. Apesar de ainda não haver padrões quanto à coleta dessas informações, o emprego de tipos de contexto pode atuar como índices na integração de outras fontes. Por exemplo, pela identidade de uma pessoa se obtém as relações com os presentes.

Um membro da classe de Usuários (*User*) é caracterizado pelas propriedades idade (*age*), sexo (*genre*), localização (*location*) e ligação a grupos (*member*) dos dados pessoais, assim como preferências (*Preferences*), presença (*Presence*), histórico (*history*) e frequência de uso (*frequencyUsage*), instanciadas a partir do MU. A Tabela 4.2 apresenta essas propriedades e outros descritores da classe *User* como entidade da ontologia de adaptação contextual, gerada pelo *plugin* OWLDoc da ferramenta de edição Protégé.

Usuários são entidades de contexto modelados como classe equivalente a pessoas (*foaf:Person*) e grupos (*foaf:Group*) da ontologia FOAF. Assim, suas propriedades são importadas para compor a definição, tais como *age*, *gender* e *member*, cujas especificações foram estendidas para descrever atributos de contexto. As classes *Preference* e *Presence* são subclasses de *ContextualElement* restritas à caracterização de usuários através das relações *hasPreference* e *isPresent*, respectivamente, assim como *location* é sub-

propriedade de `characterizes` que vincula um conceito espacial (`SpatialThing`) ao usuário, também reusado da ontologia FOAF.

Tabela 4.2: Representação da Classe User

Class: User urn://avantv/2010/ctx#User	Usage (15)
Class Hierarchy + Thing + ContextThing + ContextualEntity User	• Class: User • Preference $\equiv$ characterizes only User (Class) • Presence $\equiv$ characterizes only User (Class) ◦ $\equiv$ isPresent some Literal • hasPreference Domain User (ObjectProperty) ◦ SubPropertyOf isCharacterizedBy • member Range (characterizes some User) (ObjectProperty) ◦ EquivalentProperties foaf:member • age Domain (characterizes some User) (DataProperty) ◦ EquivalentProperties foaf:age • gender Domain (characterizes some User) (DataProperty) ◦ EquivalentProperties foaf:gender
Equivalents (1) • foaf:Group or foaf:Person • tvd:User	• frequencyUsage Domain (hasParticipant some (characterizes some User)) (DataProperty) • isPresent Domain (characterizes some User) (DataProperty) • history Domain User (ObjectProperty) • location Range User (ObjectProperty) • participateIn Domain User (ObjectProperty) • adaptedTo Range User (ObjectProperty) • hasParticipant Range User (ObjectProperty) • isRelevantTo Range (Activity and actionDateTime some TemporalEntity and actionPlaceIn some SpatialThing and hasParticipant some User) (ObjectProperty)

Existem ainda propriedades associadas aos metadados Mpeg7, tais como equivalência entre a classe `Preference` e `mpeg7:UserPreferenceType`, além do relacionamento `history` de `User` com `mpeg7:UsageHistoryType`. As demais propriedades são descritores da interação do usuário com aplicações (`participateIn`, `hasParticipant` e `isRelevantTo`) e adaptação do conteúdo (`adaptedTo`), descritas na Tabela 4.4.

Para modelar o conteúdo (*Content*), foram estendidos termos do padrão de metadados *Dublin Core*. As propriedades título (*title*), assunto (*subject*), data (*date*) e tipo (*type*) descrevem atributos básicos que caracterizam *Content*, conforme ilustra a Tabela 4.3. Assunto é representado por terminologias de classificação de conteúdo, enquanto o tipo é derivado da mídia descrita pelos itens.

Tabela 4.3: Representação da Classe *Content*

Class: Content urn://avantv/2010/ctx#Content	Usage (9)
Class Hierarchy + Thing + ContextThing + ContextualEntity Content	• Class: Content • Item $\equiv$ characterizes only Content (Class) • hasPart Domain Content (ObjectProperty) ◦ SubPropertyOf isCharacterizedBy • hasContent Range Content (ObjectProperty) ◦ Domain Application or TelevisionProgram • date Domain (characterizes some Content) (DataProperty) ◦ EquivalentProperties dc:date • subject Domain (characterizes some Content) (DataProperty) ◦ EquivalentProperties dc:subject • title Domain (characterizes some Content) (DataProperty) ◦ EquivalentProperties dc:title • type Domain (characterizes some Content) (DataProperty) ◦ EquivalentProperties dc:type • AdaptedContent SubClassOf Content ◦ adaptedIn Range InstantThing ◦ adaptedTo Range User
Equivalents (1) • tvd:aggregation some tvd:ElementaryStream	

Como o conteúdo das aplicações é o foco de personalização do modelo, seus itens são descritos pelo elemento contextual *Item*, associado pelo relacionamento *hasPart*. O tipo de mídia (atributo *mediaType*) e os conceitos, estendidos do modelo de domínio, são propriedades dos itens que orientam as operações de adaptação. A classe que representa o conteúdo adaptado (*AdaptedContent*), subclasse de *Content*, é relacionada à descrição do instante (*adaptedIn*) e

usuário (`adaptedTo`) a que foi ajustado. A classe `Content` se relaciona às classes `TelevisionProgram` e `Application` da ontologia TVD pela propriedade `hasContent` e equivale à restrição que agrega fluxos elementares.

Aplicações (`Application`), programas de TV (`TelevisionProgram`) e recursos computacionais (`Resource`), estendidos da ontologia TVD, também são entidades contextuais. O tipo das aplicações é utilizado para definir o conjunto de operações adaptativas a serem empregadas. Recursos nos dispositivos são definidos pela avaliação de propriedades da infraestrutura, por exemplo, memória livre versus total, espaço de armazenamento e existência de canal de interatividade.

As ações realizadas pelos usuários de aplicações que requerem adaptação são descritas por atividades (`Activity`) quanto ao instante (`TemporalEntity`), local (`SpatialThing`) e agente (`User`) da interação. Através da realização das atividades é possível determinar quais propriedades de contexto devem ser instanciadas (`isRelevantTo/relevancyCE`) e qual a relevância das mesmas para a interação (`relevancy`), conforme apresenta a Tabela 4.4.

Tabela 4.4: Representação da Classe Activity

Class: Activity urn://avantv/2010/ctx#Activity	Usage (6)
Class Hierarchy + Thing + ContextThing Activity	• Class: Activity • actionDateTime Domain Activity (<i>ObjectProperty</i>) • actionPlaceIn Domain Activity (<i>ObjectProperty</i>) • hasParticipant Domain Activity (<i>ObjectProperty</i>) • isRelevantTo Range (Activity and actionDateTime some TemporalEntity and actionPlaceIn some SpatialThing and hasParticipant some User) (<i>ObjectProperty</i>) ○ Domain ContextualElement ○ Inverse relevancyCE (<i>ObjectProperty</i>) ○ relevancy Domain (relevancyCE some ContextualElement) (<i>DataProperty</i>) • participateIn Range Activity (<i>ObjectProperty</i>) ○ Inverse hasParticipant

Declarações sobre essas ontologias podem compor regras a serem processadas para dedução de novos conhecimentos. Por exemplo, a classificação e a participação de itens do conteúdo esportivo, descritas pelas propriedades do modelo de domínio `isClassifiedBy` e `isParticipantIn`, respectivamente, podem caracterizar uma situação do conteúdo relevante para identificação de eventos esportivos, quando relacionados aos itens como elementos contextuais.

4.5 Procedimentos de Inferência

Para realizar inferências sobre informações de contexto, utilizamos mecanismo de raciocínio baseado em ontologias e regras do Jena. Uma regra é representada como um conjunto de condições e um conjunto de ações que indicam procedimentos a serem executados quando as condições são satisfeitas. Portanto, o comportamento contextual, resultante desta avaliação, pode modificar o modelo de usuário, atualizar o valor de relevância para a relação entre elementos de contexto e atividade de interação, ou realizar adaptação conforme a base de conhecimento.

A API do Jena possui um subsistema de inferência que permite derivar novos conhecimentos da ontologia (JENA API, 2011). A partir da descrição de classes e instâncias codificadas em RDF ou OWL é possível inferir fatos adicionais. Embora o mecanismo que realiza esta tarefa, chamado *reasoner*, seja projetado de modo geral, há suporte para inclusão de construtores específicos. Os principais *reasoners* definidos pela arquitetura do Jena são:

- *Transitive reasoner*: considera apenas propriedades simétricas e transitivas. Devido à simplicidade, apresenta alto desempenho e eficiente uso de memória, usado por *reasoners* mais complexos.
- *RDFS reasoners*: trata um subconjunto configurável de especificações RDFS, cujo desempenho varia segundo os níveis *full*, *default* ou *simple*.
- *OWL reasoners*: trata inferências baseadas em regras sobre especificações OWL. Em decorrência da complexidade da linguagem, são fornecidos três *reasoners*: um pré-definido *full*, um menor *mini* e um mínimo, porém mais rápido *micro*. A escolha decorre de requisitos da

aplicação e performance que oferecem. Exceto por limitações sobre as restrições de cardinalidade `minCardinality` e `someValuesFrom`, o *OWLMini* abrange todos os construtores suportados e incluídos no tipo padrão *OWLFull*, o que evita expansões infinitas introduzidas por recursos sem URI (*blank nodes*). *OWLMicro* utiliza construtores RDFS mais descritores de classes e propriedades, tais como união e equivalência. Pela modificação do conjunto de regras é possível ainda incluir construtores além dos suportados.

- *Generic Rule Reasoner*: possibilita customizar os demais *reasoners* pela adição de novas regras e primitivas.

A associação de algum *reasoner* a um conjunto de dados é efetuada durante a criação de um dos tipos de modelos oferecidos através da classe `ModelFactory` da API ou registrados para ativação dinâmica. Assim, além das declarações sobre os esquemas e dados originais, consultas ao modelo retornam afirmações adicionais derivadas dos dados usando mecanismos de inferência implementados pelo *reasoner*. No entanto, é possível vincular *reasoners* mais especializados a diferentes conjuntos de instâncias de dados (chamado *abox*), enquanto inferências sobre informações do esquema (denominado *tbox*) são reusadas.

Na abordagem AvanTV são combinadas a implementação mais leve do OWL *reasoner* para inferências sobre classes e a adição de regras de adaptação sobre os indivíduos através da classe `Rule`, para inferência pelo *Generic Reasoner*. Para descrição das regras uma seqüência de termos (*bare-rule*) divide as premissas da conclusão pela representação direcional `<-` ou `->`, conforme sintaxe simplificada baseada na notação BNF:

```
Rule      :=  bare-rule .
           or  [ bare-rule ]
           or  [ ruleName : bare-rule ]

bare-rule :=  term, ... term -> hterm, ... hterm
           or  bhterm <- term, ... term

hterm     :=  term
           or  [ bare-rule ]
```

```

term      := (node, node, node)
           or (node, node, functor)
           or builtin(node, ... node)

bhterm    := (node, node, node)

functor   := functorName(node, ... node)

node      := uri-ref
           or prefix:localname
           or <uri-ref>
           or ?varname
           or 'a literal'
           or 'lex' ^^typeURI
           or number

```

Cada termo (*term*) da regra representa uma declaração na forma de tripla ou uma chamada a uma primitiva procedural. Os componentes da declaração e os parâmetros são representados por nós (*node*), que podem expressar um URI, referências por prefixos, uma variável da regra, um literal simples ou tipificado, ou um número. A conclusão da regra (*hterm*) pode ser constituída por um termo ou outras regras. Quando do encadeamento inverso (*backward*), a conclusão da regra (*bhterm*) é formada apenas por triplas de nós. Uma coleção de primitivas, úteis na realização de comparações, é predefinida pelo Jena, tais como *lessThan()*, *sum()* e *equal()*, contudo outras podem ser criadas e registradas.

Algumas regras foram previamente incorporadas ao modelo, considerando o comportamento contextual e assumindo os seguintes prefixos para as ontologias: *tv*d para TV digital (*urn://tvdigital*); *sps* para o domínio *SPTC Semantic* (*urn://avantv/2010/sptcsem*); e *ctx* para adaptação contextual (*urn://avantv/2010/ctx*). Porém, conforme discutido na seção 4.2.2, é possível adicionar novas regras dinamicamente. Como exemplo, citamos:

R1: quando ocorre interação entre aplicação e usuário, que não é identificado ou não há nenhum usuário registrado (*isBNode*), cria instância de um usuário presente (*makeInstance*) e produz a regra inversa (*backward*) *rule1A*, que o identifica como anônimo e inicializa sua frequência de uso.

```

[rule1: (?V ctx:participateIn ?A), (?A ctx:actionDateTime now(?x)),
isBNode(?V) -> makeInstance('true',ctx:isPresent, ctx:User, ?V)

```



```
[rule1A: (?V ctx:isRelevantTo ?A) <-(?V ctx:id 'anonymous'), (?C
owl:onProperty ctx:characterizes), (?C owl:someValuesFrom all(ctx:User,
?V)), (?C ctx:frequencyUsage 'low')]]
```

R2: verifica presença de grupo à partir dos seus membros, o que altera o significado da atualização do MU.

```
[rule2: (?A ctx:isPresent ?C), (?B ctx:isPresent ?C), notEqual(?A, ?B), (?G
ctx:member ?A), (?G ctx:member ?B) -> (?G ctx:isPresent ?C)]
```

R3: dada uma declaração de preferência do usuário por um conceito esportivo presente como elemento contextual da atividade de interação, pontua relevância do conceito do MD com peso do valor da propriedade sobre a preferência (provavelmente, decorrida dos metadados do MU).

```
[rule3: (?V ?P ?S), (?P owl:onProperty ctx:hasPreference), (?S rdf:type
sps:SportsConcept), (?V ctx:participateIn ?A), (?A ctx:relevancyCE ?S) -> (?S
relevancy ?R), sum(?P, ?R, ?R)]
```

R4: alerta sobre presença de um competidor de interesse do usuário (relevância maior que 50%) no programa televisivo sendo assistido.

```
[rule4: (?T ctx:hasContent ?C), (?V tvd:assiste ?T), (?C ctx:hasPart ?I), (?V
ctx:isPresent 'true'), (?V ?X ?I), (?X relevancy ?R), greaterThan(?R, 0.5), (?I
sps:hasParticipant ?G) -> print('See in program', ?G)]
```

R5: alerta utilizado para orientar recomendações de conteúdo similar.

```
[rule5: (?V ctx:isPresent 'true'), (?V ctx:hasPreference ?I), (?T
ctx:hasContent ?S), (?S rdf:type sps:SportsConcept), (?I rdfs:subClassOf
?S) -> print('Content for you', ?V, ?S)]
```

Obviamente, a utilidade destas e outras regras incluídas dinamicamente dependem da devida criação de instâncias que componham a base de conhecimento. Vale ainda ressaltar que o aumento da expressividade na representação de conceitos ocasiona também acréscimo na complexidade do processo de inferência.

4.6 Avaliação das Ontologias

Para avaliação do modelo de adaptação contextual da abordagem AvanTV foram construídas instâncias para uso pelas aplicações e coletadas métricas para comparação com os modelos de domínio relacionados.

Medições sobre ontologias no Protégé (PROTEGE, 2011) consideram a expressividade baseada em lógica de descrição (DL), número de classes, propriedades de objetos, propriedades de dados, indivíduos e proposições. Além destas, consideramos ainda o subconjunto da linguagem OWL empregado na construção das ontologias, como ilustra a Tabela 4.5.

Tabela 4.5: Características das ontologias relacionadas ao modelo de adaptação contextual AvantvCTX

Ontologia	Expressividade	OWL	Classes	Propriedades	Indivíduos	Proposições
AvantvCTX	SHOI(D)	DL	27	43	28	112
SPTCSem	SHI(D)	DL	39	29	27	115
TVD	SO(D)	DL	63	52	33	191
Foaf	ALCHIF(D)	DL	19	71	0	160

Foi preciso manter o compromisso entre os conceitos que descrevem as ontologias *SPTC Semantic* e *AvantvCTX* e o quão complexo seria para realizar inferência. A expressividade DL é codificada pela seguinte convenção de nomes:

- AL – lógica sobre atributos que permite negação, conjunção, restrições de valor universal, quantificação existencial limitada;
- C – negação de conceito complexo;
- S – abreviação para AL e C com propriedades transitivas;
- H – hierarquia de propriedades;
- O – nominais, tais como enumeração de classes ou de valores;
- I – propriedade inversa;
- N – restrições de cardinalidade;

- F – propriedades funcionais;
- (D) – uso de propriedades de tipo e valores de dados.

Dado que todas as ontologias utilizam o subconjunto *OWL-DL* da linguagem, os mecanismos de inferência considerados atendem a complexidade computacional imposta. É possível verificar que apesar do reduzido número de classes, a ontologia *AvantvCTX* abrange maior poder de expressão (*SHOI(D)*) e, conseqüentemente possui custo mais elevado para processamento de inferência. Porém, esse valor é amenizado por não combinar as definições de propriedade inversa e de restrições de cardinalidade, que ocasionam propagações indesejadas como no caso dos *OWL reasoners* de Jena (JENA API, 2011). Cabe salientar que a importação de outras ontologias contribui diretamente para aumentar a expressividade do modelo.

Nas ontologias *SPTCSem*, *TVD* e *Foaf*, as declarações atendem às definições lógicas *SHI(D)*, *SO(D)* e *ALCHIF(D)*, respectivamente, cujas cargas relacionadas são menores, porém têm impacto elevado pelo número de classes, propriedades e proposições dos modelos. *SPTCSem* não possuem nominais nem propriedades funcionais, mas concebe propriedades transitivas úteis na derivação de fatos relevantes do domínio. Por outro lado, a ontologia *TVD* não inclui hierarquia de propriedades que, apesar de simplificar a inferência, limita a combinação de regras. Como o projeto da ontologia *Foaf* foi concebido para emprego em diversas situações, as propriedades transitivas são eliminadas, mas preserva as funcionais.

Quanto maior o número de acepções aumenta a habilidade em sistemas incluírem afirmações semânticas, no entanto o desempenho nos tempos de resposta ao acesso das mesmas é prejudicado. A maioria das instâncias (indivíduos) geradas pelo protótipo *AvanTV* são tratadas e mantidas de maneira distinta, o que atenua este problema e capacita a melhoria das deduções.

4.7 Considerações Finais

Este capítulo apresentou a especificação utilizada pela abordagem *AvanTV* para prover conteúdo personalizado. Com base em metadados e ontologias, foram

representados conceitos da difusão de conteúdo esportivo, informações sobre usuário e como prover comportamento sensível ao contexto pela correspondência entre domínio e interesses do usuário.

Considerando pesquisas e particularidades relacionadas aos esportes, foi desenvolvido o modelo de dados *SPTC*, capaz de estruturar elementos que descrevem objetos de interesse específicos. Além disso, descritores semânticos foram definidos pelo *SPTC Semantic* para dar suporte à extração de novos conhecimentos, através de uma ontologia utilizando a linguagem OWL. Informações que caracterizam telespectadores foram descritas pelo modelo de usuário baseado, principalmente, no padrão de metadados MPEG-7. Pela extensão de seu perfil UDP, dados pessoais, histórico de uso, preferências sobre conceitos do modelo de domínio e relação com fatores do ambiente foram associados à ontologia de adaptação contextual.

Devido à integração de terminologias dos modelos propostos numa representação orientada à manipulação de contexto, foi possível realizar inferências por regras sobre as ontologias. No entanto, este é um processo contínuo e exige mecanismos que permitam, dinamicamente, adicionar regras que convenham às aplicações. Portanto, foi necessário combinar *reasoners* do *framework* Jena que atuem tanto sobre o esquema de dados quanto sobre instâncias do modelo. Apesar de esse processo ser constituído através de programação, retirando a capacidade nativa de raciocínio, por outro lado mantém a integridade do modelo para uso em outros cenários. Além disso, o reuso de outras ontologias na construção dos modelos propostos foi feito de forma modular, para evitar a sobrecarga na inferência.

O capítulo V delinea a infraestrutura necessária para análise de contexto e oferece funcionalidades de personalização por sensibilidade ao contexto.

Capítulo V.

PROTÓTIPO AVANTV: APLICAÇÕES DE TVDI SENSÍVEIS AO CONTEXTO

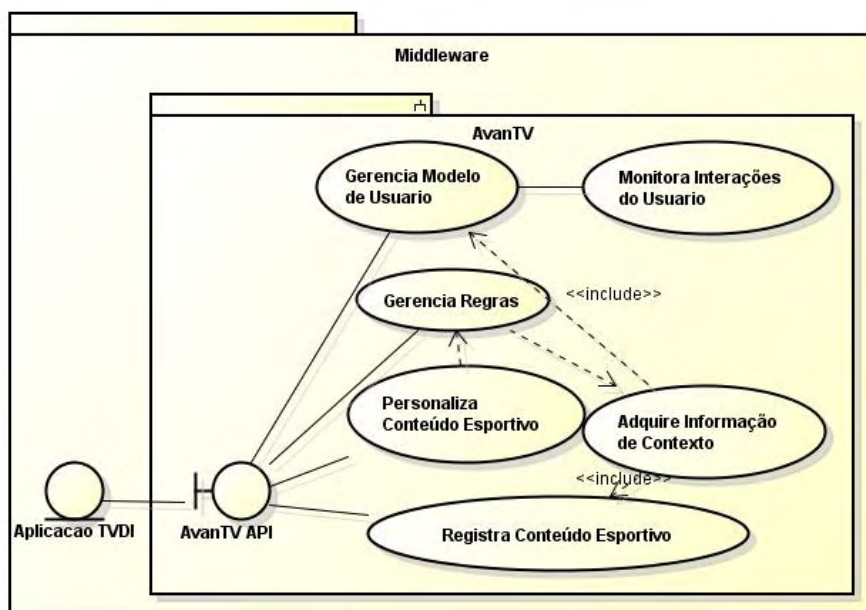
O processamento da semântica de informação contextual para apoiar personalizações na TV é realizado por funcionalidades incluídas num ambiente de recepção do sinal de *broadcast*. Este capítulo apresenta a implantação do protótipo desenvolvido na seção 5.1. A seguir, são descritos os módulos necessários para: fornecer serviços às aplicações, na seção 5.2; capturar informações de contexto, na seção 5.3; gerenciar informações, na seção 5.4; e realizar processo de inferência, na seção 5.5. Detalhes de implementação são apresentados na seção 5.6. Nosso objetivo é demonstrar a utilidade dos modelos de domínio, de usuário e de adaptação propostos na construção de aplicações sensíveis ao contexto. Para tanto foram desenvolvidos aplicativos de TVDI que ajustam seu conteúdo aos interesses da audiência, descritos na seção 5.7. Finalmente, as últimas considerações são relatadas na seção 5.8.

5.1 Implantação

Foram desenvolvidas funcionalidades comuns às aplicações sensíveis ao contexto, segundo os modelos definidos anteriormente, de modo a prover uma validação da abordagem AvanTV. A Figura 5.1 apresenta alguns casos de uso.

Para coleta de informações de contexto, é suficiente a definição das fontes e métodos para tratá-las. No entanto, as vantagens alcançadas pelas aplicações para personalização dependem diretamente da manutenção dos dados do usuário e da vinculação dos mesmos ao conteúdo. Nesse sentido, estabelecer mecanismos para exploração de novos conhecimentos e procedimentos que realizem as devidas correspondências se torna crucial. De modo a verificar o compartilhamento semântico no acesso às informações de contexto, foram desenvolvidos serviços para atender às requisições de personalização das aplicações.

Figura 5.1: Casos de uso da AvanTV



Tais artefatos são operados num ambiente de recepção de sinal televisivo (ex. *set-top box*). As aplicações que oferecem interatividade aos usuários podem solicitar diferentes serviços à interface de acesso para realizar personalização do conteúdo. Para isso, os requisitos funcionais abrangidos são apresentados a seguir:

RF1 : Manter informações sobre usuários

Deve permitir a inclusão de dados sobre telespectadores, ou grupos desses, que os identifique para registrar suas preferências e histórico de uso, assim como o contexto de suas interações.

RF2 : Perceber e identificar usuários

Dado o modo de interatividade diferenciado em plataformas de TV, é adequado oferecer autenticação indireta do usuário como alternativa para conservação do foco no entretenimento. Assim como o reconhecimento de grupos para prover adaptações afins.

RF3 : Especificar informações de contexto

Aplicações e programas televisivos com conteúdo relacionado ao

comportamento (histórico e preferências) do usuário devem ser instanciados segundo entidades contextuais.

RF4 : Prover mecanismos de personalização

Operações que realizem filtragem, ordenação e síntese do conteúdo baseadas na relevância do domínio da informação para o usuário devem ser oferecidas às aplicações.

RF5 : Habilitar descoberta de conhecimento

Para analisar e ativar contexto, devem ser proporcionadas estruturas que lidam com a construção de regras predefinidas e dinâmicas capazes de deduzir novos fatos, assim como customização das propriedades semânticas utilizadas pelos mecanismos de adaptação.

Os requisitos não-funcionais que dão suporte a essas condições são os seguintes:

RNF1 : Interoperabilidade

Para especificar quais informações as aplicações necessitam, de modo a garantir homogeneidade em diferentes plataformas, deve ser oferecida uma representação padronizada.

RNF2 : Transparência na Comunicação

Devido à capacidade reduzida de recursos computacionais do receptor, idealmente, os dados adquiridos do ambiente e a execução de ações deve ter comunicação transparente entre sensores e aplicações.

RNF3 : Extensibilidade

Por causa da variação dos níveis de interatividade, os serviços devem ser oferecidos como extensão ao *middleware* do dispositivo

de recepção. Além disso, deve ser possível ampliar a conceituação das ontologias sem prejudicar sua manutenção.

Apesar das aplicações poderem acessar o modelo de adaptação diretamente através de ontologias, por causa das restrições de recursos computacionais em dispositivos receptores esse processo é realizado por serviços oferecidos pelo subsistema AvanTV, incorporado ao ambiente de execução Ginga-J (GINGAJ, 2010) ilustrado na Figura 5.2. Essa extensão consiste em dotar a máquina de execução de uma base de conhecimento representada pelos modelos propostos e provê serviços às aplicações interativas que disponibilizem e interpretem as informações mantidas.

Figura 5.2: Ambiente de execução Ginga-J estendido

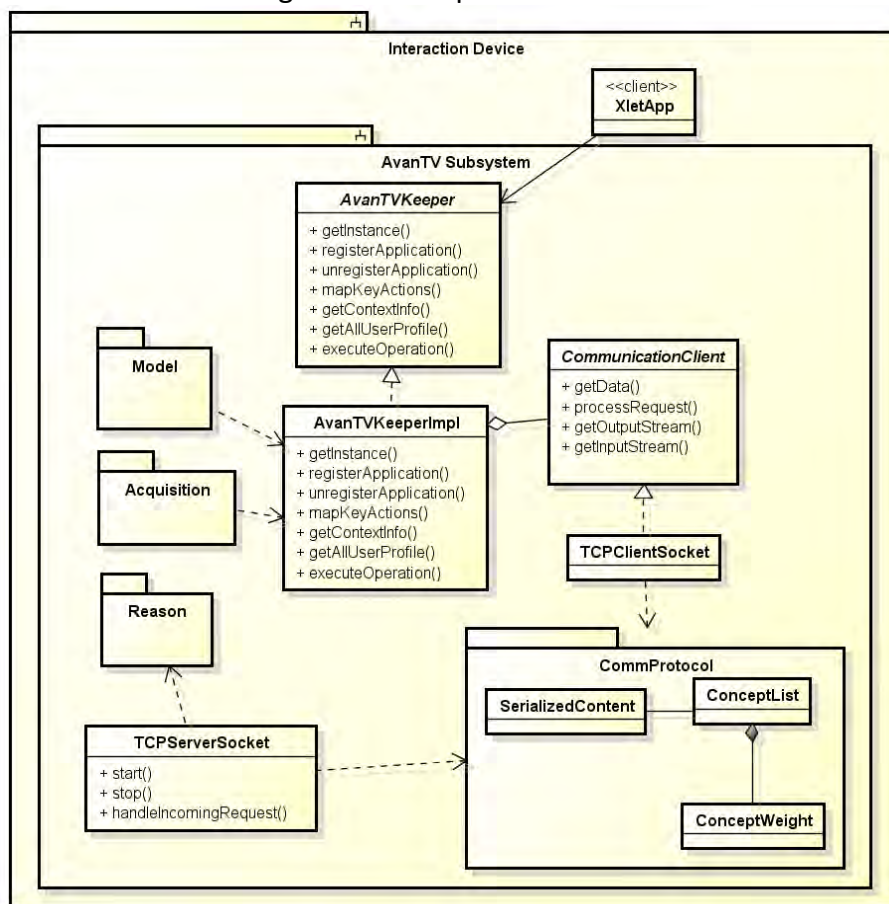


Deste modo, a manipulação de ontologias, assim como o processo de aquisição e representação de informações do usuário e do conteúdo de aplicações, é efetuada por módulos intermediários. A arquitetura, ilustrada na Figura 5.3, é dividida em três módulos principais: aquisição (*Acquisition*), gerenciamento das representações (*Model*) e raciocínio (*Reason*). Além destes, destacam-se os serviços de disponibilização de informações às aplicações (*AvanTVKeeper*) e de comunicação. Esta organização segue o processo essencial definido por Dey (2000), no qual a definição de quais informações de contexto são necessárias e como solicitá-las para uso é alcançada pela descrição do modelo; o procedimento de obtenção de contexto é realizado pelo componente de aquisição; e o comportamento sensível ao contexto é obtido pela conjunção entre regras de inferência e técnicas de adaptação. As próximas subseções detalham esses subsistemas.

5.2 Interface com Aplicações

Para que a informação esteja disponível às aplicações, foi definida uma interface que fornece acesso aos demais subsistemas através da *Facade*¹⁵ *AvanTVKeeper*. Quando esse componente é inicializado junto com a máquina de execução do *middleware*, são criados objetos que interconectam os demais módulos. A classe *AvanTVKeeperImpl* cria objetos para gerenciamento das fontes contextuais (*Acquisition*) e das representações de dados (*Model*), e se comunica com o módulo de manipulação das ontologias (*Reason*) através das estruturas de compartilhamento *TCPClientSocket* e *TCPServerSocket*.

Figura 5.3: Arquitetura da AvanTV



¹⁵ Padrão de projeto que permite isolar as funcionalidades internas do subsistema AvanTV das aplicações que requerem personalização.

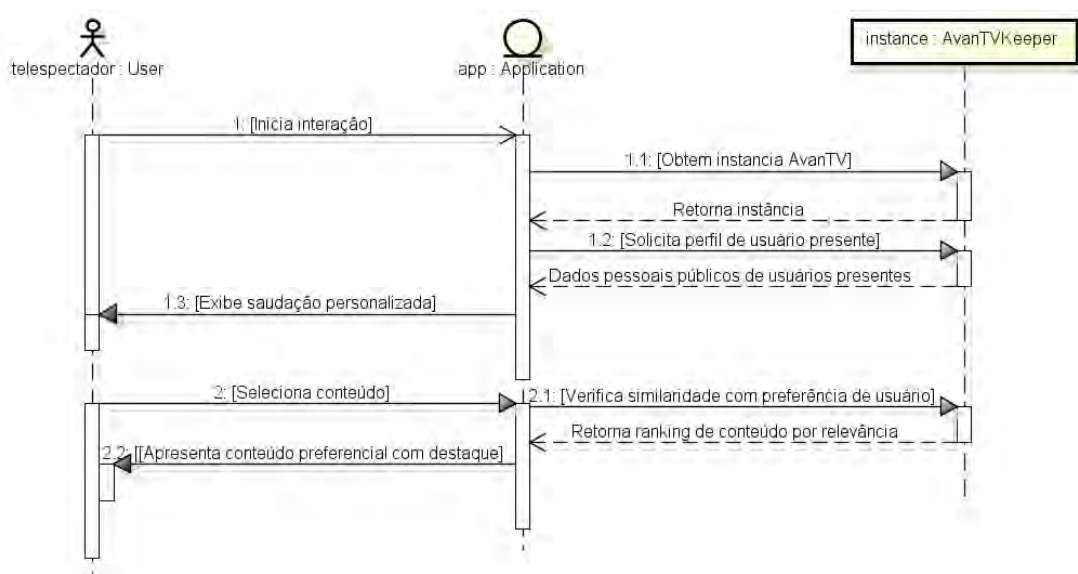
Os objetos trafegados através das estruturas de comunicação são definidos pela classe `SerializedContent`, associada a listas de conceitos (`ConceptList`). Uma `ConceptList` representa vocabulários de perfis de usuários ou itens do conteúdo na forma de termos ponderados (`ConceptWeight`). Os pesos atribuídos indicam a relevância dos termos para cada perfil, úteis na determinação de comportamento sensível ao contexto. A serialização do conteúdo é necessária para o envio de um objeto entre aplicações utilizando `sockets`¹⁶ e encapsula, além dos perfis descritos, objetos contendo um conjunto de operações do protocolo recebidos pelo servidor de comunicação (`TCPServerSocket`) e processados pelo módulo de raciocínio. Além dos métodos exibidos, classes auxiliares executam procedimentos internos, como requisições de processamento sobre as ontologias.

Assim, qualquer aplicação que acesse a arquitetura pode obter o perfil público do usuário e consultar informações gerais de contexto. Em um nível intermediário, são oferecidos serviços de personalização sobre o modelo de usuário e monitoramento das interações. Para isso, é necessário prover mapeamento da captura de teclas. Em um nível mais avançado, as aplicações podem realizar personalização baseada em contexto, pelo acionamento de inferências baseadas em regras e uso de mecanismos de adaptação sobre relacionamentos semânticos de contexto. O diagrama de seqüência da Figura 5.4 sintetiza o processo de interação básica de uma aplicação com os serviços disponibilizados.

Desse modo, para uma aplicação ter seu conteúdo personalizado, é necessário apenas solicitar informações à interface AvanTV, sem se preocupar com questões relacionadas à aquisição e representação das mesmas. No entanto, quando uma aplicação requer comportamento sensível ao contexto, ou contribui para o enriquecimento semântico das ontologias com informações do domínio, é necessário registrar configurações para cada conjunto de serviços, neste caso, descritas pelo elemento `sports-serviceType` do modelo SPTC.

¹⁶ Componente de software que provê interface de rede para uma aplicação, escondendo detalhes de baixo nível, tais como tipo de transmissão, tamanho de pacote, etc.

Figura 5.4: Diagrama de interação entre aplicação e serviços AvanTV



Apesar de relacionados, o suporte à descoberta de recursos, disseminação e gerenciamento histórico de contexto não é foco deste trabalho. Estes requisitos normalmente são desempenhados por componentes de *frameworks* que tratam a computação sensível ao contexto para diferentes propósitos.

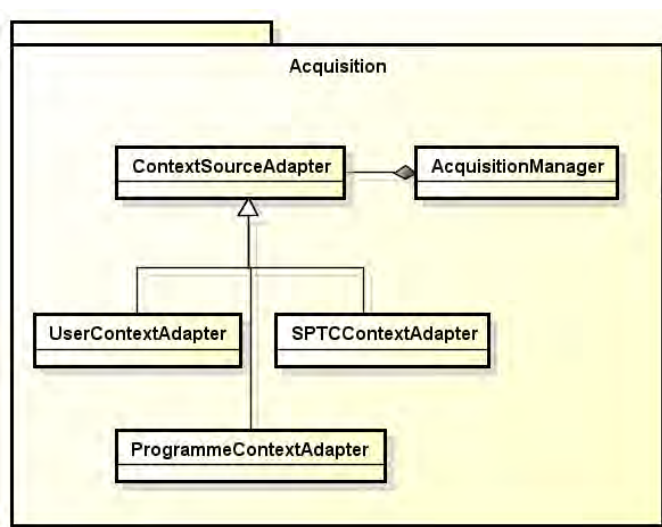
5.3 Módulo de Aquisição

Neste módulo, cuja organização é apresentada na Figura 5.5, as fontes de informação contextual são gerenciadas pela classe `AcquisitionManager`, que trata estruturas para obtenção de contexto. Particularidades de cada fonte são tratadas por extensões à classe `ContextSourceAdapter`, responsável por manter a compatibilidade entre operações. Deste modo, subclasses implementam um conjunto de operações solicitadas pelo `AcquisitionManager` à estrutura hierárquica, onde diferentes estratégias são orientadas pelos tipos dos objetos.

Adaptadores que realizam conversões a partir de fontes de contexto do usuário (`UserContextAdapter`), de serviços esportivos (`SPTCContextAdapter`) e da programação televisiva (`ProgrammeContextAdapter`) foram projetados segundo os componentes de coleta de dados de contexto:

- **Sensor de presença:** seguindo a especificação do protocolo CPP (*Common Profile for Presence*) (CPP, 2004) para transporte de informações de presença entre servidor e clientes, é disponibilizado um sensor virtual que efetiva as operações de inscrição e notificação do serviço para verificação de telespectadores presentes.

Figura 5.5: Classes do Módulo de Aquisição



- **Base de dados de serviços esportivos:** parte das inferências realizadas sobre o conteúdo esportivo depende da concretização de um modelo de conhecimento. As instâncias de objetos de interesse do MD derivadas de serviços esportivos são mantidas em documentos com declarações sobre indivíduos.
- **Eventos da programação:** para cada sistema de TVDI, é definido um conjunto específico da API que representa elementos das tabelas SI disponíveis na rede. Foi desenvolvido um adaptador para manipular eventos da programação utilizando estas interfaces.

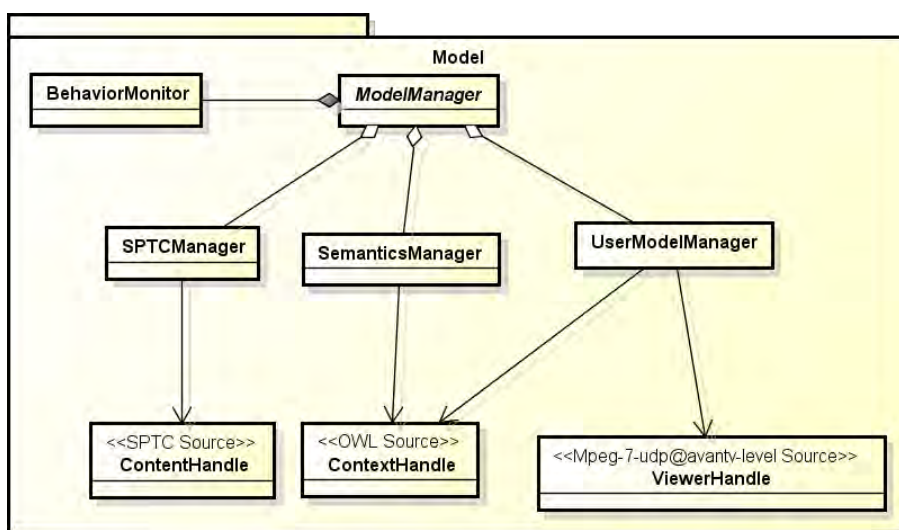
Se dados de outros sensores precisam ser capturados, não é preciso endereçar cada uma das fontes. Através do `AcquisitionManager` podem ser adicionados e consultados novos adaptadores de fontes contextuais segundo especificação da classe `ContextSourceAdapter`. A comunicação com o modelo contextual é intermediada pelas estruturas de compartilhamento, acionadas quando

da atualização de um elemento contextual ou, na direção contrária, pela consulta a valores de contexto através do `AcquisitionManager`.

5.4 Módulo de Gerenciamento

Uma vez que as informações estejam em formato válido, condizentes com as representações propostas no capítulo IV, as classes `SPTCManager`, `SemanticsManager` e `UserModelManager`, esboçadas na Figura 5.6, gerenciam-nas pela realização de operações de *parser*, de atualização ou de monitoração. Os serviços prestados podem ser acessados através da `ModelManager`, que além de agregar as funcionalidades de cada gerenciador, é composta pela classe `BehaviorMonitor`, responsável pela captura das interações do usuário.

Figura 5.6: Classes do Módulo de Gerenciamento dos Modelos



Quando aplicações interativas solicitam serviços, o conteúdo definido por objetos de interesse descritos em documento XML é utilizado para criar instâncias do modelo SPTC por meio da classe `ContentHandle`. Deste modo, tanto características do modelo de usuário que fazem uso desses objetos quanto serviços de personalização sobre o conteúdo podem ser mapeados em termos da ontologia *SPTC Semantic* e processados pelo Módulo de Raciocínio (subseção 5.5). Por exemplo, a Figura 5.7 contém trecho de um documento que descreve o serviço da tabela de classificação do campeonato brasileiro de futebol (`sports-service`).

Além da descrição de metadados (`sports-metadata`), ele inclui dados sobre o time do Corinthians (`team`) utilizado como conteúdo da aplicação. Conceitos relevantes são mapeados pelos gerenciadores, considerando URI e semelhança entre termos.

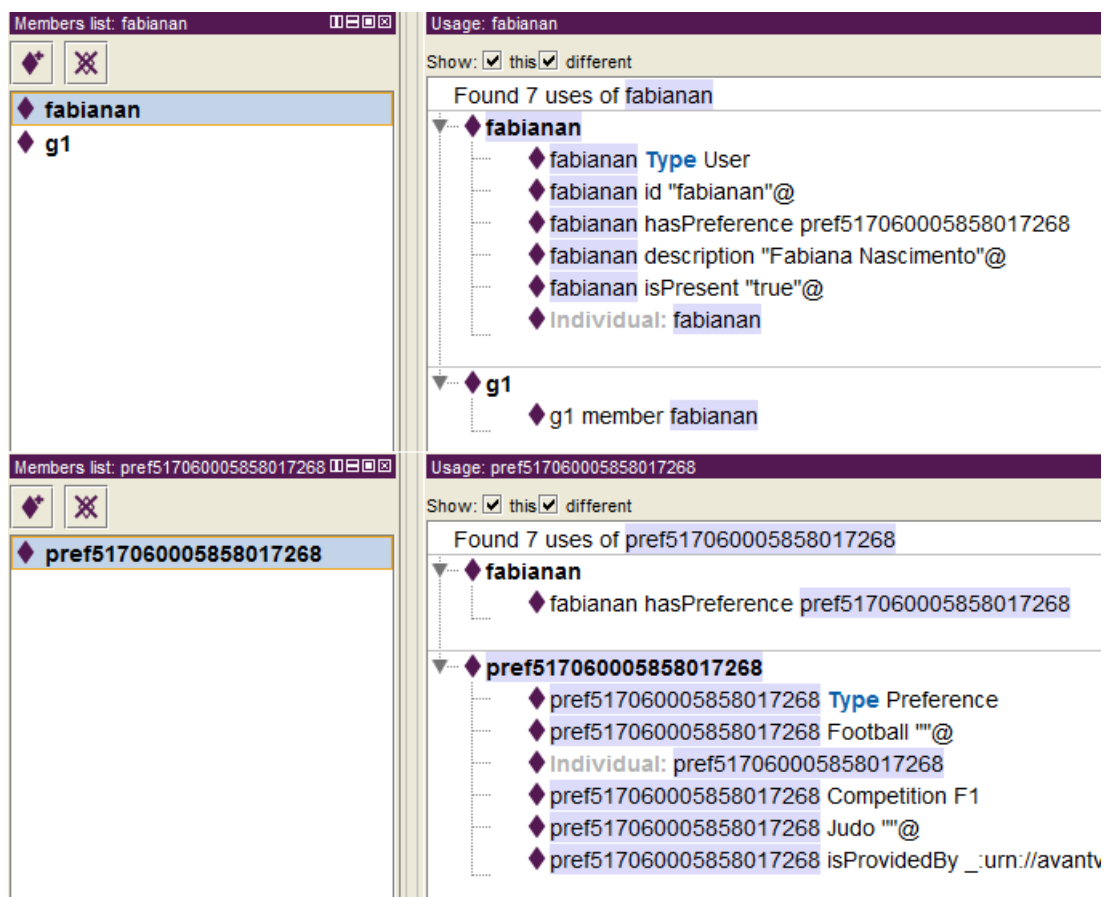
Figura 5.7: Fragmento de documento SPTC

```
<?xml version="1.0" encoding="UTF-8"?>
<sptc xmlns="urn:avantv:sptc:schema:2010"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:avantv:sptc:schema:2010 ../model/sptc.xsd">
  <sports-metadata sports-title="Campeonato Brasileiro 2011"
    date-time="2011-04-23T10:13:38-03:00" language="pt-BR" kind="futebol"
    sports-label="Brasileirao 2011 - serie A" gender="masculino"
    id="br2011.A001432"
    modality="futebol de campo" />
  <team id="br.sp.corinthians" uri="urn://avantv/2010/sptcsem#timecorinthians"
    icon="corinthians.png">
    <name first="Corinthians" nick="Timao" />
    <team-stats period-type="competicao" period-value="br2011.A001432"
      events-played="4" wins="3" ties="1" losses="0" rank="2">
      <stats name="GP" label="Gols pró" value="7"/>
      <stats name="GC" label="Gols contra" value="3"/>
    </team-stats>
  </team>
  <sports-service id="classificacaoobr2011.1" creator="CBF"
    label="Tabela de Classificacao Brasileiro 2011">
    <source type="java"/>
    <rules reason="OWL">...</rules>
  </sports-service>
</sptc>
```

Dados pessoais e preferências explicitadas pelo usuário são interpretados e validados segundo descritores do perfil UDP de MPEG-7 com restrições de complexidade descritas na seção 4.3. Adição, remoção e atualização de dados sobre pessoas, grupos, membros, preferências e histórico de uso são realizadas pelo gerenciador do modelo de usuário (`UserModelManager`). Alterações sobre informações contextuais relacionadas ao usuário também são refletidas nas operações deste componente, assim como o mapeamento dos dados mantidos em termos da ontologia do modelo contextual. Por exemplo, de uma instância do usuário *'fabianan'*, membro do grupo *'g1'* e cujas preferências por *'Judô'*, *'Futebol'* e competições da *'F1'* fornecida pelo conjunto de metadados, são obtidas entidades

de contexto da classe `User` e elemento contextual da classe `Preference`, dentre outras propriedades (vide Figura 5.8).

Figura 5.8: Instâncias do modelo de usuário

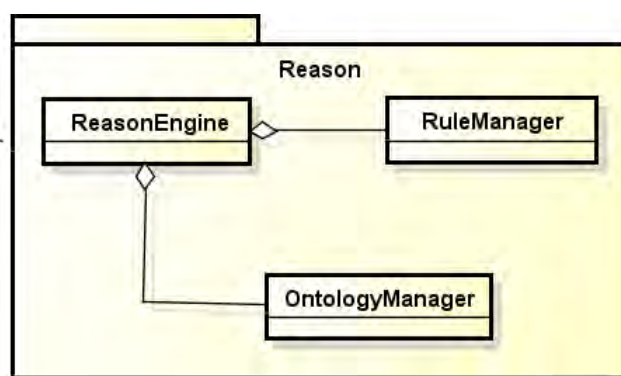


Durante monitoramento das interações do usuário com a TV, através do `BehaviorMonitor`, também são coletadas informações contextuais sobre perfis de usuários e grupos (*who*), tempo (*when*) e local de interação (*where*), assim como a que conteúdo se refere (*what*) e qual o tipo de dispositivo (*how*). Após análises periódicas, a classe `ModelManager` é encarregada de armazenar o agrupamento desses dados na forma de histórico de uso dos metadados do MU e construir instâncias de termos provenientes da informação contextual através da classe `SemanticsManager`.

5.5 Módulo de Raciocínio

Toda informação manipulada pela arquitetura AvanTV relevante para a personalização de conteúdo é mantida como instância das fontes de metadados bem como das ontologias. Um elemento fundamental para o processamento de inferências e correspondência entre os modelos é o componente de Raciocínio delineado na Figura 5.9. Ele foi desenvolvido como um subsistema remoto por questões de reuso e de extensibilidade. Os serviços compreendidos pelo mesmo são: criação e recuperação dos modelos ontológicos sobre esquemas e instâncias de dados, vinculação de *reasoners* a diferentes conjuntos, gerência de recursos e declarações, assim como de regras de inferência.

Figura 5.9: Classes do Módulo de Raciocínio



A classe `OntologyManager` é responsável pela criação e recuperação do modelo base que contém sentenças definindo classes e relações das ontologias de adaptação contextual e *SPTC Semantics (tBox Model)*, e afirmações sobre indivíduos no domínio (*aBox Model*). O modelo *tBox* padrão é vinculado às instâncias por meio da especificação para modelos OWL do Jena e utiliza mecanismo de inferências por regras *OWLMicro* para implicações adicionais sobre a base de conhecimento. Pela extensão da mesma, é possível integrar outros esquemas com respectivos *reasoners*.

Configurações de informações contextuais e métodos de reação a elas são estabelecidas pela `RuleManager`. Responsáveis por adicionar regras que indiquem condições e ações ativadas de acordo com o contexto, os serviços fornecidos por

esta e por outras classes acessórias produzem declarações que constituem tanto a base de conhecimento quanto mecanismos de inferência.

Informações contextuais mantidas pela ontologia integram dados do usuário e de itens de conteúdo a ser adaptado. Terminologias e elementos que caracterizam interesses e contexto dos telespectadores constituem indivíduos do perfil do usuário. Quando uma aplicação é registrada, seu conteúdo é mapeado em termos definidos pelo MD. Deste modo, as operações de adaptação e regras de inferência consideram a semântica dos dados de entrada referentes ao contexto, ao conteúdo, aos usuários assim como seus relacionamentos.

Estas questões e a gerência dos componentes descritos anteriormente são responsabilidade do principal elemento deste módulo: o `ReasonEngine`. O processo de manipulação das informações presentes nas ontologias acontece da seguinte forma:

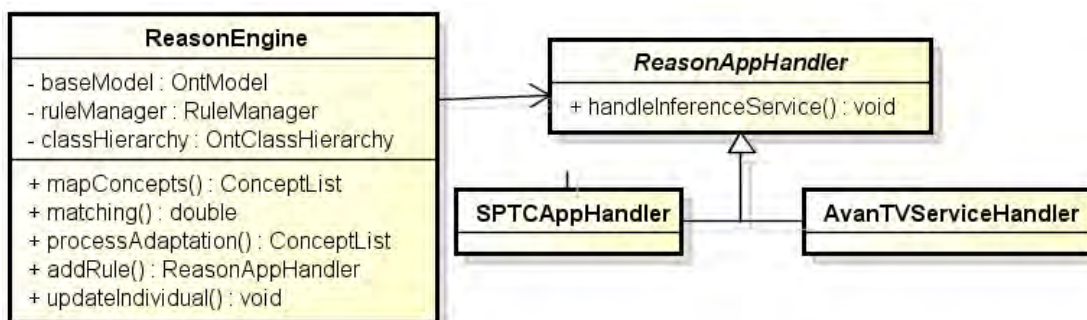
- **Criação/Atualização das ontologias:** instâncias de `OntologyManager` e do gerenciador de regras são obtidas durante a inicialização deste componente para construção do modelo base do tipo `OntModel`¹⁷ e inclusão de regras predefinidas que auxiliem a derivar fatos adicionais. Através do componente de comunicação, fontes contextuais inscritas no módulo de aquisição e conceitos mapeados constituem indivíduos a serem atualizados na ontologia sempre que necessário.
- **Mapeamento de conceitos dos modelos:** para identificar termos referentes ao perfil dos telespectadores (preferências, histórico de uso ou sensores) ou dos conteúdos de aplicações, são oferecidos métodos para construção de listas de conceitos (`ConceptList`) disponibilizadas ao módulo de gerenciamento.
- **Controle de Requisições:** as requisições de cada aplicação são atendidas por objetos definidos pela interface `ReasonAppHandler`, conforme diagrama

¹⁷ `OntModel` é uma classe da API Jena que permite acesso aos *statements* da coleção de classes, propriedades e indivíduos das ontologias.

da Figura 5.10. Regras específicas, relacionamentos semânticos considerados nas operações de adaptação e estado das informações de contexto que satisfazem as condições de interesse das aplicações são controlados por estes objetos.

- **Operações de adaptação contextual:** comportamento sensível ao contexto pode ser tratado diretamente pelas aplicações, porém, opcionalmente, podem utilizar operações baseadas nas técnicas de HA para filtragem, ordenação e mapeamento de itens do conteúdo que considerem ainda terminologias do modelo contextual.

Figura 5.10: Diagrama de classes dos objetos de controle de requisições



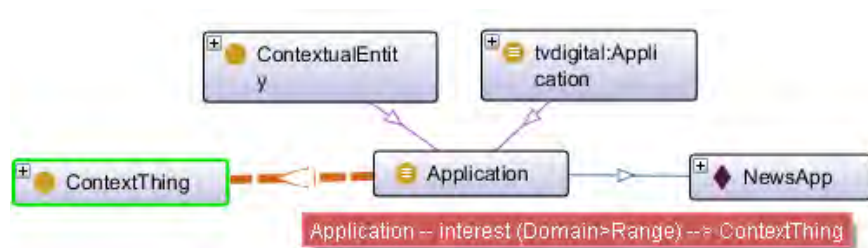
Para evitar que alguma requisição não seja atendida por causa da ausência de informação na ontologia, um conjunto de regras e relacionamentos genéricos são pré-definidos e tratados pelo `AvanTVServiceHandler`. Outras classes de controle devem ser definidas quando dados de outras ontologias são integrados.

Com o objetivo de reduzir a carga no processamento de inferências, a parte das regras relativa a particularidades das aplicações é abordada de forma diferente em relação às outras regras de contexto predefinidas. Depois que a regra é obtida da descrição do documento SPTC (por meio do elemento `sports-serviceType`), é validada e adicionada pelo objeto `RuleManager` com um vínculo ao *handler* apropriado. Ao invés de compor a base de conhecimento, as declarações derivadas desse tipo de regras são retornadas ao controlador.

Devido a um artifício do próprio modelo de adaptação contextual proposto na seção 4.4, as aplicações que possuem requisições atendidas também são

mapeadas como indivíduos da classe `Application`, subclasse de `ContextualEntity`. Através do predicado próprio `interest`, apresentado na Figura 5.11, é possível monitorar toda informação de contexto representada pela classe `ContextThing`, como fontes de contexto (`ContextSource`) e elementos contextuais (`ContextualElement`).

Figura 5.11: Ontologia da classe `Application`



A propriedade `interest` tem como sujeito das declarações a classe `Application` e como objeto a classe `ContextThing` e suas subclasses. Através de sentenças deste tipo, é possível determinar o que deve ser considerado como relevante no contexto de uma atividade. Portanto, o processamento do subconjunto de instâncias relevantes ao foco de interesse das aplicações é dado justamente pelas regras que ativam comportamentos contextuais pré-definidos. Por exemplo, para a filtragem do conteúdo da aplicação `NewsApp`, segundo os elementos contextuais de presença (`Presence`) e preferências (`Preference`) da entidade usuário (`User`), é indicado o interesse nas mudanças das mesmas pela inclusão dos *statements* (`NewsApp`, `interest`, `Presence`) e (`NewsApp`, `interest`, `Preference`) na ontologia. Quando condições definidas pela regra que aciona a filtragem do conteúdo são atendidas, a lista de conceitos mapeada para itens do conteúdo de `NewsApp` é submetida à respectiva operação da `ReasonEngine`. Todos os parâmetros específicos à aplicação são gerenciados pelo objeto de controle (*handler*) vinculado.

5.6 Implementação do Protótipo AvanTV

Os subsistemas foram concebidos e testados no ambiente de execução *Ginga-J Emulator* (SARAIVA JUNIOR, 2010), baseado no emulador *XletView* para visualização de aplicações *Xlets* do MHP em um computador, mas que observa a especificação JavaDTV. Os módulos da arquitetura foram incorporados como funcionalidades complementares, exceto pelo módulo de raciocínio que é ativado independentemente pelo servidor de comunicação. Desse modo, o protótipo foi concebido para atender aplicações Java *Xlets* coerentes com a norma Ginga-J (GINGAJ, 2010).

A classe `AcquisitionManager` é constituída, principalmente, de uma coleção de adaptadores adicionados para simular o comportamento de sensores na tarefa de coletar dados do contexto. Cada adaptador herda, necessariamente, da classe abstrata `ContextSourceAdapter` que provê operações de inscrição e notificação orientadas pelo protocolo CPP (CPP, 2004). Para o contexto do usuário, a classe `UserPresenceSourceAdapter`, uma especialização de `UserContextAdapter`, inicializa os usuários presentes a partir de um sensor virtual que implementa *PresenceListener* e controla as mudanças de estado quanto à presença/ausência de usuários. Alterações das propriedades que incluem preferências do usuário específicas da plataforma também poderiam ser tratadas como fontes contextuais pelo uso da interface `com.sun.dtv.platform.UserPropertyListener` da API JavaDTV. Elas não foram consideradas, pois as classes do pacote ainda estão incompletas no emulador e, para fins de validação, o adaptador para contexto de presença do usuário foi suficiente. Além desse, conteúdo de aplicações fornecedoras de informações de contexto, segundo modelo SPTC, são coletados por `SPTCContextAdapter`, útil para criação de instâncias de interesse do usuário segundo ontologia esportiva *SPTC Semantic*. De modo análogo, outros domínios devem considerar aplicações e programas televisivos bases para aquisição de contexto pelos respectivos adaptadores.

Para criar coletores da programação foi utilizado o suporte da API Java TV que compõe a estrutura do modelo de aplicação Ginga-J. Os pacotes dessa API fornecem controle sobre funcionalidades únicas para receptores de televisão, inclusive para manusear serviços de informação. Na ausência de uma base de dados SI (*SI Database*) concreta de acesso dependente do protocolo de transporte, foi utilizado arquivo XML segundo gramática definida pela *Sun* empregada em exemplos de amostra, conforme trecho ilustrado na Figura 5.12. Deste modo, metadados de serviços e detalhes da programação puderam ser obtidos pelas classes e interfaces dos pacotes `javax.tv.service.*` como mudanças capturadas pelo adaptador `ProgrammeContextAdapter` do módulo de aquisição. Esta classe monitora eventos de um programa de TV na grade de programação definidos pela interface `ProgramScheduleListener` da API JavaTV.

Figura 5.12: Descrição XML de Serviço de Informação

```
<Service
  NAME="SERVI"
  SERVICETYPE="DIGITAL_TV"
  SITYPE="SBTVD"
  NUMBER="6"
  MINORNUMBER="1"
  CALIMIT="interrupt"
  SIMULATION="file:../video/Sample.mov">
  <ServiceDetails
    PROVIDERNAME="Globo"
    LONGNAME="Rede Globo"
    DESCRIPTION="Rede Globo de Televisao"
    DELIVERYSYSTEMTYPE="TERRESTRIAL">
    <ProgramSchedule>
      <ProgramEvent
        NAME="Jornal Hoje"
        DURATION="120"
        DESCRIPTION="Acidente Aereo em Recife"
        STARTTIME="sysdate"/>
      <ProgramEvent
        NAME="Globo Esporte"
        DURATION="120"
        DESCRIPTION="Crise na Selecao Brasileira"
        STARTTIME="+120" />
      <ProgramEvent
```

Desde que uma das responsabilidades do módulo de gerenciamento dos modelos é prover informações sobre usuários e conteúdo representados pelos esquemas de dados, a realização de *parser* desses formatos é indispensável. A ferramenta de linha de comando *scomp* disponibilizada pela coleção de bibliotecas

*XMLBeans*¹⁸ consegue transformar *XML Schemas* em classes e interfaces Java. Com os tipos compilados dos modelos UDP MPEG-7 e SPTC, foram escritos os códigos-fontes das respectivas classes de manipulação de dados, *UserModelManager* e *SPTCManager*. Exemplos de métodos que interpretam a instância do modelo de usuário e adicionam uma pessoa como usuário são apresentados no trecho de código abaixo:

```
private Mpeg7Document mpeg7Doc;
private Mpeg7 root;

/**
 * Instantiate and parses user model instance from usersFile param
 */
public void parseUserModel(String userFile) throws Exception {
    mpeg7Doc = Mpeg7Document.Factory.parse(new
        FileInputStream(userFile));
    root = mpeg7Doc.getMpeg7();
}

/**
 * Add a new person (no duplicates)
 * @param person The person to be added
 */
public void addPerson(PersonType person) {
    if (person!=null && person.getId()!=null &&
        !containsObjectInModel(person)) {

        // insert a new description at begins
        UserDescriptionType newudt = root.insertNewDescription(0);
        newudt.setUser(person); // set a person in new description
        saveModel(); // save to user model
    }
}
```

Além disso, a classe de acesso *ModelManager* também possui a capacidade de capturar as interações do usuário com o controle remoto por meio do *BehaviorMonitor*, pela implementação da interface *com.sun.dtv.ui.event.UserInputEventListener*. Tais observações ocorrem por uma *thread* independente acionada quando uma instância *AvanTV* é solicitada, ou diretamente pelo usuário. Dados de contexto e detalhes de interatividade com aplicações também são armazenados quando da atribuição pelo subsistema e mapeamento dado pelas aplicações. Mesmo quando desconhecido, a

¹⁸ Disponível em <http://xmlbeans.apache.org/>

dimensão *who* mantém identificação de usuário anônimo (*anonymous*), e a dimensão *when* define Intervalos de anotação, cujos parâmetros *from* e *to* indicam marcos de tempo, conforme ilustra o seguinte fragmento:

```
<when from=21-02-2011 13:27:56 to=21-02-2011 13:28:03>
<who userId=anonymous>
<where locationId=PB_BR>
<what contentId=br.ufpb.avantv.xlets.apps.GameTableApp>
<how resourceId=avantv>
21-02-2011 13:27:57 >> browseTable - Key_Pressed - KeyCode=VK_DOWN
21-02-2011 13:27:58 >> browseTable - Key_Pressed - KeyCode=VK_DOWN
21-02-2011 13:28:01 >> selectItem - Key_Pressed - KeyCode=VK_ENTER
```

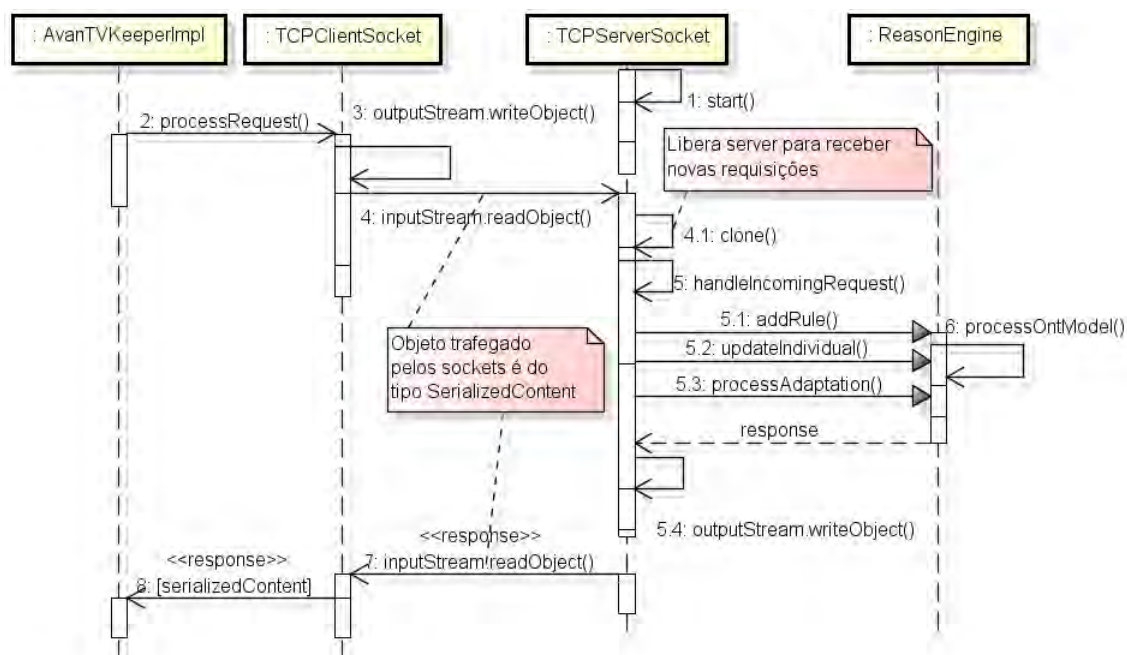
Esses dados são registrados em arquivo, sem tratamento, e analisados periodicamente, segundo parâmetro de configuração `br.avantv.behavior.interval` medido em segundos ou suspensão da *thread*, ou seja, o que ocorrer primeiro. A análise é realizada pela classe `SemanticManager` que armazena em um histórico de uso (`UsageHistoryType`) do respectivo usuário: (1) o período de observação (*when*), (2) a lista de ações do usuário (`UserActionList`) com os respectivos tipos de interações do usuário (`UserActionType`), (3) contendo identificador do conteúdo (dimensão *what*), tipos de ações (*browseTable*, *selectItem*, *changeChannel* etc.), número de instâncias (contagem de registros iguais) e duração total. Apesar de ser uma estratégia simplista por não considerar tipo de mídia e por desprezar localização (*where*) e modo (*how*) de interação, ela disponibiliza um conjunto de dados que, após o emprego de técnicas de mineração e filtragem, habilita serviços de recomendação e personalização, porém não atendidos por esta implementação.

Para acessar as ontologias, a classe principal `ReasonEngine` do módulo de raciocínio faz uso de um modelo base do tipo `OntModel`, fornecido pelo Jena, que permite manipular as ontologias *SPTC Semantic* e *AvantvCTX*. A vinculação entre esquemas (*tbox*) e instâncias (*abox*) é realizada por meio do registro do *reasoner* no construtor do `OntologyManager`.

Este subsistema é acionado separadamente pelo servidor de comunicação `TCPServerSocket`, que trata as solicitações recebidas de clientes

TCPClientSocket por múltiplos processos em paralelo. As requisições obtidas na forma de objetos SerializedContent podem ser para criação e atualização de indivíduos, adição de regras e operações de adaptação, seguindo processo descrito pelo diagrama de seqüência da Figura 5.13.

Figura 5.13: Diagrama de colaboração para requisições ao Reason



Quando o cliente de comunicação recebe a requisição da interface AvanTV através do método `processRequest()`, o objeto serializado, contendo protocolo com o tipo do processamento a ser realizado e respectivos parâmetros, é transmitido ao servidor, pela escrita e leitura nos fluxos de saída e entrada (`outputStream.writeObject()` do cliente e `inputStream.readObject()` do servidor). Ao receber dados, o servidor replica uma nova instância de si próprio e, enquanto a nova *thread* lida com a solicitação de entrada, o processo corrente permanece ativo aguardando novas requisições. A depender dos parâmetros de `SerializedContent`, as seguintes funções podem ser realizadas:

- Adição de regra pelo método `addRule()` da `ReasonEngine` é acionado para validar os valores a serem comparados com termos da ontologia (por

exemplo, frequência de uso alta, interesse em usuários presentes, entre outros.) e salvar pelo `RuleManager` com objeto de controle (*handler*) vinculado. Esse processo utiliza uma especificação de regra para inferência genérica do Jena que analisa *string* e retorna uma lista de regras: `List<Rule>`

```
rules = Rule.parseRules(rulesrc).
```

- Criação ou atualização de instância da ontologia através de `updateIndividual()` submete a lista de conceitos parametrizada para criação de *statements* no modelo base. Duplicações são evitadas pela verificação da existência dos recursos do *statement* no modelo. Para buscar todas as declarações que combinem com a tripla <sujeito, predicado, objeto>, utiliza-se a API Jena: `StmtIterator iterator = model.listStatements(sujeito, predicado, objeto)`. E para criar uma nova declaração: `Statement aStatement = model.createStatement(sujeito, predicado, objeto)`.
- Processamento de adaptação do conteúdo pelo método `processAdaptation()` é executado sob listas de conceitos recebidos para diferentes propósitos. A correspondência entre conceitos do perfil do usuário e de itens do domínio é realizada pela extensão do mecanismo de filtragem baseada em conteúdo utilizando ontologias proposto em Shoval *et al.* (2008).

5.6.1 Filtragem baseada em conteúdo utilizando ontologias

Conforme introduzido na seção 2.4, o método para filtragem baseada em conteúdo desenvolvido por Shoval *et al.* (2008), soluciona problemas de ambigüidade na correspondência textual pelo uso de conceitos de ontologias para representar perfis de usuário e de itens do conteúdo.

Por exemplo, sejam os termos $U1 = \text{'Corinthians'}$ e $U2 = \text{'Campeonato Brasileiro'}$ no perfil de um usuário haverá um grau de similaridade com itens $I1 = \text{'esporte'}$ e $I2 = \text{'competição'}$ de um conteúdo, onde: $U1 = I1$ por correspondência fraca (R) da hierarquia `Team` $\rightarrow$ `Competitor` $\rightarrow$ `SportsConcept` da ontologia *SPTC Semantics*; e

$U2=I2$ por correspondência perfeita (P) visto que $U2$ é um indivíduo da classe representada por $I2$, e $U2=I1$ também por (R).

As pontuações assumidas para cada caso são dadas como exemplos e para aprimorar o desempenho nem todos os conceitos do usuário devem participar da computação, apenas os mais relevantes. Porém, para o exemplo anterior, é possível observar que não é obtida correspondência entre $U1$ e $I2$, mesmo sabendo que ‘*Corinthians*’ é um time e que, certamente, tem correlação com alguma ‘*competição*’. Para ampliar a expressividade deste algoritmo, foi concebido um novo caso, o de **correspondência semântica (Y)**, cuja pontuação determinada experimentalmente é igual a $1/5$.

Para atender esta situação, o algoritmo original (*Matching (Item-List, User-List)*) foi alterado da seguinte maneira: uma lista de conceitos tem a habilidade de incluir predicados que relacionam conceitos semanticamente (*Related-properties*) e uma nova condição foi adicionada no laço sobre perfil do usuário que avalia a existência de declarações do modelo, onde *U-concept* é o sujeito, *Related-properties* são os predicados testados e *I-concept*, o objeto. Para nosso exemplo, é relevante incluir as propriedades *SptcSem.competesIn* e *SptcSem.performedBy*, subpropriedades de *SptcSem.isParticipantIn* e *SptcSem.hasParticipant*, respectivamente. Deste modo, como o modelo base contém o *statement* (*Competitor, competesIn, Competition*), a avaliação de equivalência entre $U1$ (*‘Corinthians’* → *Competitor*) e $I2$ (*‘competição’* → *Competition*) será verdadeira.

Esta estratégia pode causar inconsistência ao considerar propriedades e restrições dos esquemas OWL e RDF. Por exemplo, se *U-concept* tiver propriedade *owl#equivalentClass* com um *blank node*, estes depreciam o cálculo de similaridade. Para evitar esse problema, tais predicados são desconsiderados durante equivalências do tipo (Y).

Logo, a preparação do algoritmo processado pelo Módulo de Raciocínio (Reason) exige a construção de relacionamentos hierárquicos e a atribuição de predicados semânticos relevantes dos modelos. O resultado da execução de tais correspondências é a base para as operações de adaptação (vide Figura 5.14): **ordenação de itens** considera a relevância entre perfis para estabelecer a disposição; **filtragem** recebe um parâmetro que indica a condição dos itens retornados (por exemplo, similaridade entre faixa de valores); e **síntese** define graduações de realce para distribuir itens do conteúdo.

Figura 5.14: Processamento de adaptação de conteúdo

```

83  /**
84   * Process an adaption technique given in obj.getCommand() parameter over
85   * obj.getContents()
86   * @param obj The protocol object to be interpreted
87   */
88  public void processAdaptation(SerializedContent obj) {
89      // calculate and update each item weight relevance from content
90      matching(obj.getContents());
91      switch (obj.getCommand()) {
92          case CommandType.FILTERING:
93              // condition defines a filter over contents
94              processFiltering(obj.getContents(), obj.getCondition());
95              break;
96          case CommandType.ORDERING:
97              processOrdering(obj.getContents());
98              break;
99          case CommandType.RESUME:
100              // condition defines the number of ranges to detail
101              processStretching(obj.getContents(), obj.getCondition());
102              break;
103          default:
104              break;
105      }
106  }

```

5.7 Exemplos de uso

A validação das funcionalidades oferecidas pelos serviços do protótipo da AvanTV foi feita pelo desenvolvimento de duas aplicações *Xlets*: um agregador de notícias, apresentado na subseção 5.7.1, e um seletor listagem de eventos esportivos com detalhamento de competidores, na subseção 5.7.2.

Considerando que existem perfis de usuários registrados num receptor doméstico segundo modelo de usuário definido na seção 4.3. O sensor virtual de presença detecta a chegada e saída de pessoas do ambiente em que o dispositivo se encontra. Preferências são capturadas por aplicações orientadas ao diálogo, onde o usuário indica o grau de interesses em conceitos do domínio diretamente, e o

histórico de uso é obtido de maneira implícita segundo identificação fornecida pelo módulo de aquisição e monitoramento das interações. A verificação de presença de grupos, definida pela regra `rule3` do modelo de adaptação, aciona alterações de descritores identificados pelo grupo e não por seus membros. Caso não haja registro de qualquer usuário ou grupo, as informações capturadas são atribuídas a um usuário anônimo e utilizadas também em inferências futuras.

Baseado nos perfis de usuários mantidos pelo módulo de gerenciamento, o protótipo AvanTV atualiza as informações do modelo conforme as demais regras de inferência ativadas por componentes do módulo de aquisição ou por solicitações feitas pelas aplicações. Dados coletados quando os usuários interagem com serviços cujo conteúdo é definido pelo modelo de domínio ou com programas de TV, também constituem novas declarações da base de conhecimento.

Os perfis de usuários para teste das aplicações nas próximas subseções são de usuários descritos como segue:

- *Pedro*: pertence ao grupo ‘teen’, curte lutas marciais.
- *Julia*: do mesmo grupo de Pedro, tem 15 anos e costuma assistir programas relacionados à música eletrônica e beleza.
- *Mariana*: é membro do grupo ‘família’ que inclui todos os demais usuários descritos. Prefere assistir TV à tarde cuja programação é relacionada com novelas e filmes. Porém, é torcedora do Santos FC e costuma acompanhar informações sobre campeonatos pela *web*.
- *Francisco*: tem 40 anos, torce pelo time do Corinthians e gosta de noticiários. Costuma assistir TV apenas no início da manhã e à noite.

5.7.1 Agregador de Notícias

A exibição de notícias agregadas de diferentes fontes (ou *feeds*) é prática comum na Web. Isto evita a visita a diversos endereços que disponibilizam tais conteúdos, permitindo que informações diversas sejam acessíveis de maneira interligada num único local, ou ainda, através de diferentes dispositivos. O formato

em que são oferecidos é essencialmente por documento XML no padrão RSS (*Rich Site Summary* ou *Really Simple Syndication*) constituído por elementos que descrevem informações do canal que fornece o conteúdo e respectivos itens.

Um aplicativo com esta funcionalidade para TV foi desenvolvido com o objetivo de prover notícias personalizadas aos telespectadores e, assim, validar o protótipo AvanTV. Os procedimentos para acesso ao agregador de notícias são os seguintes:

1. Após iniciada por controle remoto, são submetidas requisições aos *feeds* endereçados no receptor para busca dos referentes conjuntos de notícias. De modo a transpor a ausência de canal de interatividade, são transmitidos alguns documentos de canais predefinidos por *broadcast* juntamente com a aplicação. Inscrições em canais são mantidas por arquivo de configuração;
2. É feito o *parser* dos documentos RSS obtidos e agregados numa coleção de itens para exibição;
3. A aplicação solicita à interface *AvanTVKeeper* a lista de pessoas presentes e apresenta tela padrão com saudação personalizada mais quadro com todas as notícias indicando a quantidade de páginas disponíveis.
4. Uma nova requisição é feita à interface para submeter o volume total de notícias para adaptação por filtragem dos itens relacionados ao perfil do usuário (pessoa ou grupo presente). Através de mensagem diferenciada, a aplicação oferece opção para usuário apertar botão e reduzir o conteúdo àqueles de seu interesse.
5. Se opção for selecionada, o quadro de notícias é atualizado com conteúdo filtrado. O diagrama de interação da Figura 5.4 ilustra parte desse processo.

Algumas telas da aplicação são apresentadas na Figura 5.15 e na Figura 5.16 incluindo todos os itens dos *feeds* disponíveis (veja detalhe do número de páginas). No painel de notícia é exibido o título da informação com data de publicação, cuja navegação por setas do controle traz mais detalhes noutro painel abaixo. Observe a janela auxiliar, carregada pelo módulo de aquisição para simular os usuários registrados no modelo, assim como o estado de presença/ausência em que se encontram. Quando identificada a presença de um usuário não registrado, é

acionado o alerta com opção para preencher perfil básico e demais alternativas desabilitadas (vide Figura 5.15). O diálogo para os usuários registrados *Francisco* e *Mariana* oferece opções de personalização (vide Figura 5.16).

Figura 5.15: Aplicação de Notícias com opção para usuário não registrado



Figura 5.16: Aplicação de notícias com opções para usuários registrados



Como *Francisco* e *Mariana* têm pouco tempo para ler tantas notícias, ativam opção para ter acesso a conteúdo exclusivo (pressionando o botão verde do controle remoto). Para oferecer esse tipo de personalização, a requisição submetida pela aplicação no passo 4 dos procedimentos para interação com o agregador de notícias é enviada ao módulo de Raciocínio. Através da *ReasonEngine* são mapeados os conceitos do conteúdo para execução da operação de filtragem que mede a relevância dos mesmos conforme os perfis dos usuários presentes. De posse dos itens filtrados e com as respectivas medidas de relevância atribuídas, a aplicação reordena e, se necessário, exibe apenas o conteúdo com pontuação mais alta.

O resultado do acesso ao conteúdo exclusivo para os usuários pode ser visto na Figura 5.17. Além da redução na quantidade de notícias, que pode ser limitada por parâmetro de configuração, as primeiras notícias são diretamente relacionadas às preferências dos usuários.

Figura 5.17: Aplicação de notícias com conteúdo personalizado



5.7.2 Listagem de Eventos Esportivos

Dentre as aplicações esportivas, consultar a programação de eventos esportivos e tabelas de classificação de campeonatos são as mais requisitadas. Para validação do modelo SPTC assim como de inclusão de regras pelo serviço do domínio, a primeira foi desenvolvida.

No fragmento abaixo, encontram-se metadados que exemplificam a descrição das competições apresentadas por esta aplicação, através de documento SPTC:

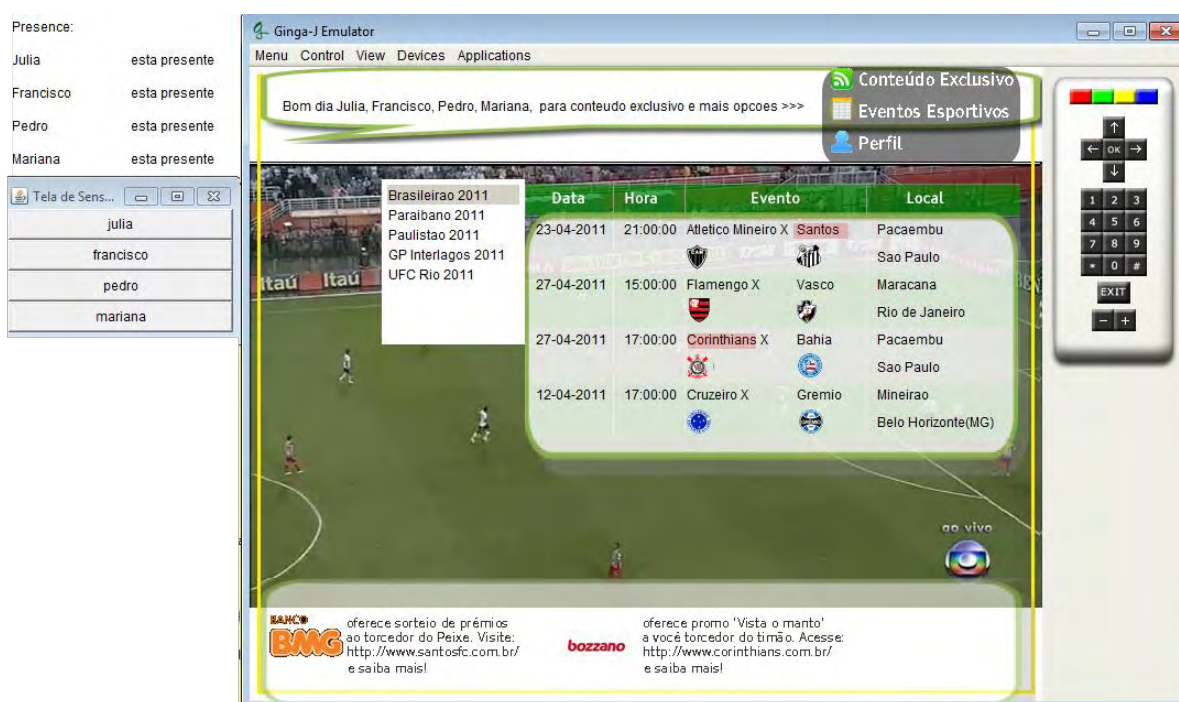
```
<sptc xmlns="urn:avantv:sptc:schema:2010">
<tournament> <event-metadata uri="urn://avantv/2010/sptcsem#Football" evt-
name="Brasileirao 2011" />
  <sport-event>
    <event-metadata start-date-time="2011-04-23T21:00:00-03:00">
    <location id="br.sp.pacaembu" type="stadium">
      <name full="Pacaembu" />
      <address city="Sao Paulo" /></location></event-metadata>
    <team uri="urn://avantv/2010/sptcsem#timeatleticomg"
      icon="atleticomg.png">
      <name first="Atletico Mineiro" nick="Galo" />
      <sponsor uri="urn://avantv/2010/sptcsem#bancobmg"
        name="Banco BMG" icon="bmg.png" />
    </team>
    <team uri="urn://avantv/2010/sptcsem#timesantos"
      icon="santos.png">
      <name first="Santos" nick="Peixe" />
      <sponsor uri="urn://avantv/2010/sptcsem#seara"
        name="Seara" icon="seara.png" /></team>
  </sport-event>
</tournament>
<sports-service id="tablebr2011.1" creator="servicePromoted"
  label="Eventos Esportivos" icon="SportData.png">
<facilities label="relatedProperty" value="sps#isClassifiedBy"/>
<facilities label="relatedProperty" value="sps#classifies"/>
<source type="java" classname="br.ufpb.avantv.xlets.GameTableApp" />
<rules reason="OWL"> @prefix ctx: <urn://avantv/2010/ctx#>.
  @prefix sps: <urn://avantv/2010/sptcsem#>.
  [adteamrule: (?V ctx#isPresent 'true'), (?V ctx#hasPreference ?T), (?T
rdf#type sps:Team) (?T sps#sponsor ?S)-> print('Anuncie para', ?V,?S)]
</rules> </sports-service>
</sptc>
```

Na descrição do serviço esportivo (sports-service) a aplicação define duas propriedades semânticas relacionadas, a serem consideradas no processamento de equivalência das operações de adaptação do módulo de raciocínio: classifies/isClassifiedBy. E ainda, acrescenta a regra adteamrule para

processamento de anúncio publicitário caso um usuário (?V) esteja presente e tenha um time (?T) exibido na listagem de competições esportivas como predileto, aciona a primitiva que recomenda exibição do patrocinador (?S) ao usuário. Estes descritores são tratados pelo Módulo de Gerenciamento que encaminha as regras ao Módulo de Raciocínio para controle de requisições das aplicações (*handlers*).

O resultado da personalização do conteúdo após correspondência do conteúdo aos interesses do usuário e ativação da regra definida é apresentado na Figura 5.18. Observe que os times de preferência de usuários presentes são realçados e a conclusão da regra aciona o anúncio publicitário dos respectivos times. Como o usuário *Pedro*, que curte lutas marciais, também está presente, um anúncio personalizado é ainda apresentado na Figura 5.19.

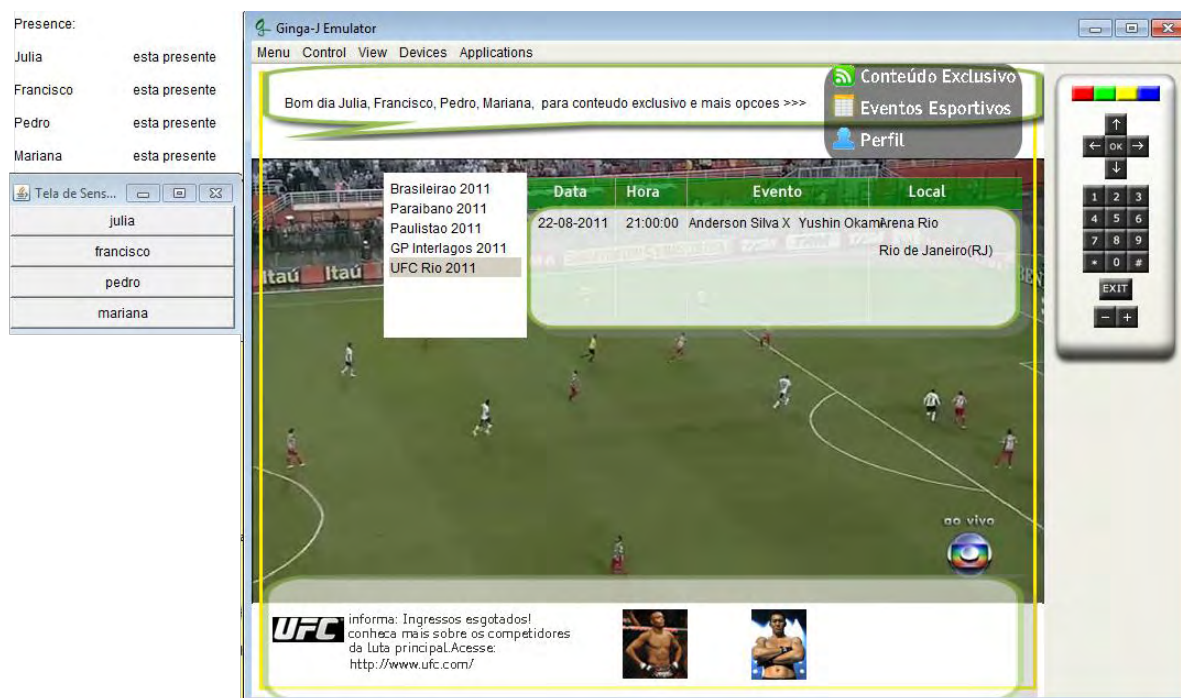
Figura 5.18: Listagem de eventos esportivos com realce de conteúdo



A competição da UFC, na ontologia SPTC Semantic, é um conceito esportivo classificado como Artes Marciais e Lutas (*MartialArtsFightSports*). Para satisfação da regra *adteamrule*, a instanciação de termos do conteúdo da

aplicação permite inferir os conceitos classificados deste tipo como equivalentes às preferências de algum dos usuários presentes pela ontologia de contexto.

Figura 5.19: Listagem de eventos esportivos com anúncio personalizado



5.8 Considerações Finais

Este capítulo apresentou a arquitetura do protótipo AvanTV que provê funcionalidades para aplicações sensíveis ao contexto. Os módulos responsáveis pela aquisição de contexto, pelo gerenciamento de informações dos modelos propostos no capítulo IV, pela realização de inferência por regras e pela disponibilização de acesso foram detalhados.

O protótipo foi projetado como uma extensão da máquina de execução Ginga-J e implementado num emulador. Operações para adaptação de conteúdo foram descritas em termos de um método para filtragem baseada em conteúdo utilizando ontologias. O algoritmo deste método foi estendido para considerar, além dos relacionamentos hierárquicos, correspondências por similaridade semântica definidas por predicados das ontologias. Para validação, foram desenvolvidos aplicativos *Xlets* que realizaram personalização do conteúdo.

Capítulo VI.

CONCLUSÃO

Este trabalho trata a necessidade de adaptação do conteúdo de aplicações de TVDI segundo o paradigma de sensibilidade ao contexto. Considerando, principalmente, informações do usuário, da programação e do conteúdo, foi possível definir uma abordagem que incorpora semântica à manipulação de informações de contexto.

Neste capítulo, são resumidos os resultados dos capítulos anteriores e apresenta indicações para o desenvolvimento de trabalhos futuros.

6.1 Principais Contribuições

Investigamos o problema de personalização do conteúdo de aplicações interativas para TV de modo a facilitar a recuperação automática de informações do usuário. Depois de identificadas questões relacionadas à representação de conhecimento, ao modo de interatividade na TV, à construção de aplicações sensíveis ao contexto e de técnicas para adaptação de conteúdo, foi concebida a abordagem AvanTV, principal contribuição desta dissertação. A realização da mesma possibilita:

- Representar informações de contexto usando ontologias, que apóia o compartilhamento semântico entre aplicações;
- Modelar informações de usuário, que permite associar seus interesses e monitorar o contexto das interações que realiza;
- Oferecer solução para aquisição de informações de contexto a partir de fontes heterogêneas e de maneira extensível;
- Prover gerenciamento comum dos dados representados;
- Realizar inferência por regras de informações de contexto, não apenas limitadas às ontologias propostas, mas também pela inclusão de outras;

- Disponibilizar operações de adaptação do conteúdo considerando a semântica das relações entre elementos contextuais;

Além de facilitar a integração de dados contextuais, a utilização de padrões tanto na modelagem de informações do usuário quanto do domínio proporciona interoperabilidade entre diferentes plataformas. A expressividade pelo enriquecimento das ontologias pode ser ampliado, desde que observado o equilíbrio com custos computacionais de processamento.

Devido à modularidade das funções desempenhadas pelos serviços desenvolvidos no protótipo, os mesmos podem ser oferecidos de forma independente. Como no oferecimento de recomendações ou interações colaborativas através de regras de inferência apropriadas. A realização de adaptação de conteúdo através de sensibilidade ao contexto como uma extensão à máquina de execução do *middleware* brasileiro Ginga descreve uma funcionalidade inovadora, que requer exploração pelo subsistema declarativo.

6.2 Trabalhos Futuros

A abordagem AvanTV proposta foi idealizada para o meio televisivo de modo a realizar adaptações do conteúdo no domínio esportivo. Com o objetivo de transpor limitações, são sugeridos os seguintes trabalhos futuros:

- Expandir o uso para outras plataformas e áreas de aplicação: além dos dispositivos de interação de TV, é necessário envolver diferentes meios para produzir situações de atuação de contexto mais completas. Desse modo, a especificação de domínios distintos motiva condições mais próximas à realidade.
- Abranger outros conhecimentos através de representações semânticas, segundo padrões existentes: à medida que se amplia a expressividade dos modelos contextuais, eleva-se o grau de complexidade. O equilíbrio entre essas características pode ser alcançado pela ligação entre dados padronizados em uso segundo demanda de raciocínio;

- Formalizar a descrição das operações de adaptação num nível mais abstrato: as mudanças segundo o contexto não se referem apenas ao conteúdo e, portanto, é relevante a generalização de tais procedimentos;
- Realizar adaptação também no comportamento, na interface e na interação com outras entidades que se comunicam com a aplicação;
- Oferecer ferramentas que facilitem a criação e manutenção de instâncias dos modelos.

REFERÊNCIAS

ARDISSONO, L.; GENA, C.; TORASSO, P.; BELLIFEMINE, F.; DIFINO, A.; NEGRO, B. (2004) *User Modeling and Recommendation Techniques for Personalized Electronic Program Guides*. In: *Personalized Digital Television - Targeting programs to individual users*, Vol. 6, chapter 1 pp. 3-26, Kluwer Academic Publishers, 2004.

ABOWD, G.D.; ATKESON, C.G.; HONG, J.; LONG, S.; KOOPER, R.; PINKERTON, M. (1997) *Cyberguide: a mobile context-aware tour guide*. *Wireless Networks*, Vol. 3, No. 5, Kluwer Academic Publishers 1997.

ALVES, L. G. P. (2008) *CollaboraTVware: uma infraestrutura ciente de contexto para suporte a participação colaborativa no cenário de TV Digital Interativa*. Dissertação (Mestrado), 203p. – Depto. de Engenharia de Computação e Sistemas / Escola Politécnica USP, 2008.

ARIB (2007) Association of Radio Industries and Businesses. ARIB Standard STD-B24 Data Coding and Transmission Specification for Digital Broadcasting, version 5.1, 2007.

AROYO, L.; CONCONI, A.; DIETZE, S.; KAPTEIN, A.; et al. (2009). NoTube – making TV a medium for personalized interaction. In: *EuroITV2009 - European Interactive TV Conference*, Leuven, Belgium, 03-05 Jun 2009.

ATSC (2003) ATSC Advanced Television Systems Committee. ATSC Data Broadcast Standard A/100 – DTV Application Software Environment – Level 1. 2003.

BBC (2011) BBC Ontologies – The Sport Ontology. Disponível em <<http://www.bbc.co.uk/ontologies/sport/2011-02-17.shtml>>. Acesso em 01/07/2011.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. (2001) The Semantic Web. Scientific American Magazine. Disponível em <<http://www.scientificamerican.com/article.cfm?id=the-semantic-web>>. Acesso em 01/07/2011.

BUCZAK, A. L.; ZIMMERMAN, J.; KURAPATI, K. (2002) Personalization: Improving Ease-of-Use, Trust and Accuracy of a TV Show Recommender, Proc. Workshop on Personalization in Future TV. Málaga, Spain, May 2002

BULCÃO NETO, R. F. (2006) Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto, Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação – ICMC-USP, 2006.

BURNETT, I. S.; PEREIRA, F; WALLE, R. V; KOENEN, R. (2006) The Mpeg-21 Book. John Wiley & Sons, Ltd. Jun 2006.

CESAR, P.; CHORIANOPOULOS, K. (2008) The Evolution of TV Systems, Content, and Users Toward Interactivity. Foundations and Trends in Human–Computer Interaction, Vol. 2, No. 4, pp. 279–373, 2008.

CHEN, H. (2004) An Intelligent Broker Architecture for Pervasive Context-Aware Systems, PhD Thesis, University of Maryland, Baltimore, 2004.

CHEN, J.H.; CHI, P.P.Y.; CHU, H.H.; CHEN, C.C.H.; HUANG, P (2010) A Smart Kitchen for Nutrition-Aware Cooking. IEEE Pervasive Computing 9, 4 pp. 58-65, October, 2010.

CHEVERST, K.; DAVIES, N.; MITCHELL, K.; FRIDAY, A.; EFS-TRATIOU, C. (2000) Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, New York, USA, pp.17–24, 2000.

COELHO, A. S. S. (2008) ProfileTV: Um Sistema de Gerenciamento de Perfis em TVDI. Dissertação de Mestrado 164p. – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2008.

CPP (2004) CROCKER, D; PETERSON, J. Common Profile for Presence – CPP. RFC 3859, IETF, August 2004. Disponível em <<http://tools.ietf.org/html/rfc3859>>. Acesso em 01/07/2011.

DEY, A. K.; ABOWD G. D.; BROWN P. J.; DAVIES N.; SMITH M.; STEGGLES, P. (1999) Towards a Better Understanding of Context and Context-Awareness. In: Proc. of the 1st international Symposium on Handheld and Ubiquitous Computing - Karlsruhe, Germany, Sep., 1999.

DEY, A.K. (2000) Providing architectural support for building context-aware applications. PhD Thesis, College of Computing, Georgia Institute of Technology. Disponível em <<http://www-static.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf>>. Acesso 01/07/2011.

DVB (2007) DVB Digital Video Broadcasting. TS 102 812 V1.3.1 Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification. DVB Bluebook A068 Rev. 3, 2007.

FERNANDES, J.; LEMOS, G.; SILVEIRA, G. (2004) Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas. Jornada de Atualização em Informática do Congresso da SBC, 2004.

GEM (2010) ETSI TS 102 728 V.1.1.1: Digital Video Broadcasting (DVB) Globally Executable MHP version 1.2.2. ETSI Standard, 2010. Disponível <http://www.etsi.org/deliver/etsi_ts/102700_102799/102728/01.01.01_60/ts_102728_v010101p.pdf>. Acesso em 01/07/2011.

GINGANCL (2007) ABNT NBR 15606-2 – Associação Brasileira de Normas Técnicas. Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de

aplicação XML para codificação de aplicações. SBTVD, NBR 15606-2 2007, v.3 2009.

GINGAJ (2010) ABNT NBR 15606-4 – Associação Brasileira de Normas Técnicas. Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 4: Ginga-J – Ambiente para a execução de aplicações procedurais. SBTVD, NBR 15606-4 2010.

GOREN-BAR D.; GLINANSKY, O. (2002) Family Stereotyping – A Model to Filter TV Programs for Multiple Viewers, Proc. Workshop on Personalization in Future TV, Málaga, Spain, May 2002

GOULARTE, R. (2003) Personalização e Adaptação de Conteúdo Baseadas em Contexto para TV Interativa. 262p. Tese (Doutorado) – Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2003.

GU, T.; PUNG, H. K.; ZHANG, D. Q. (2005) A Service-Oriented Middleware for Building Context-Aware Services, Elsevier Journal of Network and Computer Applications (JNCA), v. 28, n. 1, pp. 1-18, 2005.

IBGE (2009) Instituto Brasileiro de Geografia e Estatística - IBGE. Síntese de Indicadores. Pesquisa Nacional por Amostra de Domicílios PNAD 2009. Disponível em
<http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2009/pnad_sintese_2009.pdf>. Acesso em 01/07/2011.

IBOPE (2010) Índice de Audiência Domiciliar e Individual Copa 2010. Grupo IBOPE - IBOPE Mídia, jul/2010. Disponível em
<http://www.ibope.com.br/download/copa2010/Copa_2010_Jogos_PNT.pdf>. Acesso em 01/07/2011.

JAVADTV (2010) ABNT NBR 15606-6 – Associação Brasileira de Normas Técnicas. Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 6: Java DTV 1.3. SBTVD, NBR 15606-6 2010.

JENA API (2011) Jena – A Semantic Web Framework for Java. Talis Systems Ltd./Epimorphics Ltd. Disponível em <<http://www.openjena.org/>>. Acesso em 01/07/2011.

KOSCH, H.; BÖSZÖRMÉNYI, L.; HÖBLING, G.; COQUIL, D.; HEUER, J. (2008) Personalization of Mobile Multimedia Broadcasting, Journal of Digital Multimedia Broadcasting, August, 2008.

LEITE, L. E. C.; LIMA, O.; SOUZA FILHO, G. L.; MEIRA, S. R. L.; TEDESCO, P. (2007) Uma Arquitetura de Serviço para Avaliação de Contextos em Redes de TV Digital. XXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC'2007, Belém/PA, 2007.

LIANG, C.; XU, C.; LU, H. (2009) Personalized Sports Video Customization Using Content and Context Analysis. 2009. In: International Journal of Digital Multimedia Broadcasting, Volume 2010

LUGMAYR, A.; NIIRANEN, S.; KALLI, S. (2004) Digital Interactive TV and Metadata: Future Broadcast Multimedia. Springer-Verlag NY, 2004.

MARTINEZ, J. M. (2004) MPEG-7 Overview ISO/IEC JTC1/SC29/WG11 N6828, Tech. V.10. Palma de Mallorca, October, 2004.

MASTHOFF, J.; PEMBERTON, L. (2005) Adaptive Hypermedia for Personalized TV. In: Adaptable and Adaptive Hypermedia Systems. IDEA group publishing, pp. 246-263, 2005.

MEMÓRIAGLOBO (2011) Memória Globo – Globo Comunicação e Participações S.A. Disponível em <<http://memoriaglobo.globo.com/>> . Acesso em 01/07/2011.

MORRIS, S.; SMITH-CHAIGNEAU, A. (2005) Interactive TV Standards: A Guide to MHP, OCAP and JavaTV. Focal Press, 2005.

MPEG21 (2011) MPEG-21 Multimedia Framework ISO/IEC JTC 1/SC 29. Disponível em <<http://www.itscj.ipsj.or.jp/sc29/29w42911.htm#MPEG-21>>. Acesso em 01/07/2011.

NIELSEN (2009) Pesquisa Online Global de Consumidores da Nielsen – Confiança, Valor e Envolvimento com Publicidade. Adweek Media, The Nielsen Company, Julho/2009.

NASCIMENTO, F. F.; BEZERRA, E. P. (2011a) AvanTV: Uma Abordagem para Personalização do Conteúdo de Aplicações de TV Digital Interativa Sensíveis ao Contexto. In: I Escola Paraibana de Informática, Workshop de Pós-Graduação (WPG), João Pessoa/PB, 2011.

NASCIMENTO, F. F.; BEZERRA, E. P. (2011b) An Approach for Content Personalization of Context-Sensitive Interactive TV Applications. In: Second Euro ITV Workshop on Interactive Digital TV in Emergent Economies, EuroITV. Lisboa, Portugal, 2011

NOY, N. F.; MCGUINNESS, D. L. (2001) Ontology Development 101: A Guide to Creating Your First Ontology Stanford University Knowledge Systems Laboratory Technical Report KSL-01-05, March 2001. Disponível em <http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html>.

OLDHAN, J.D.; SNEDEGAR, R. (2010) The most World Cup-crazy countries. The Official Google Blog, 15-07-2010. Disponível em <<http://googleblog.blogspot.com/2010/07/most-world-cup-crazy-countries.html>>. Acesso em 01/07/2011.

OWL W3C (2004) OWL Web Ontology Language. W3C Recommendation, february/2004. Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso 01/07/2011.

PROTEGE (2011) The Protégé Project. Stanford University. Disponível em <<http://protege.stanford.edu>>. Acesso em 01/07/2011.

RDF W3C (2004) Resource Description Framework – RDF Primer. W3C Recommendation, Feb/2004. Disponível em: <<http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>>. Acesso 01/07/2011.

SBTVD (2011) Fórum do Sistema Brasileiro de TV Digital. Disponível em <<http://forumsbtvd.org.br/>>. Acesso em 01/07/2011.

SARAIVA JUNIOR, E.G. (2010) Ginga-J Emulator: Uma Ferramenta de Execução de Aplicações Imperativas para o Middleware Ginga. Trabalho de Conclusão de Curso. Departamento de Informática, UFPB, 2010

SCHILIT, B.; ADAMS, N.; WANT, R. (1994) Context-Aware Computing Applications. In: Proc. of 1st International Workshop on Mobile Computing Systems and Applications, pp. 85-90, Santa Cruz, CA, 1994.

SCHMIDT, A. (2002) Ubiquitous Computing – Computing in Context. PhD Thesis, Lancaster University, U.K., 2002.

SHOVAL, P.; MAIDEL, V.; SHAPIRA, B. (2008) An Ontology-Content-Based Filtering Method. International Journal Information Theories and Applications, Vol. 15, pp. 303-314, 2008.

SILVA, F. S.; ALVES, L. G. P.; BRESSAN, G. (2009) PersonalTVware: Uma Proposta de Arquitetura Sensível ao Contexto para Suporte a Recomendação Personalizada de Conteúdo no Cenário da TV Digital Interativa. Simpósio Brasileiro de Computação Ubíqua e Pervasiva, Bento Gonçalves/RS, 2009.

SOARES, L. F.; LEMOS, G. (2007) Interactive Television in Brazil: System Software and the Digital Divide. In European Interactive TV Conference - EuroITV2007. Amsterdam, 2007.

SOUZA FILHO, G. L.; LEITE, L. E. C.; BATISTA, C. E. C. F. (2007) Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. Journal of the Brazilian Computer Society.Vol.13.p.47-56. Porto Alegre/RS, 2007.

STRANG, T.; LINNHOF-POPIEN, C. (2004) A Context Modeling Survey. In: First International Workshop on Advanced Context Modelling, Reasoning and Management, 6th UbiComp, England, 2004.

THAWANI, A.; GOPALAN, S.; SRIDHAR, V. (2004) Context Aware Personalized Ad Insertion in a Interactive TV Environment. In: Adaptive Hypermedia: 4th Workshop Personalization in Future TV, Holanda, 2004.

TV-ANYTIME (2010) ETSI TS 102 822 - Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems (TV-Anytime). Disponível em <<http://tech.ebu.ch/tvanytime/>>. Acesso em 01/07/2011.

WANT, R.; HOPPER, A.; FALCAO, V.; GIBBONS, J. (1992) The active badge location system. ACM Transactions on Information Systems, Vol. 10, No. 1, pp.91–102, 1992.

WU, H. (2002) A Reference Architecture for Adaptive Hypermedia Applications. Dissertation, Computer Science Department, University of Amsterdam. Eindhoven, The Netherlands. November, 2002.

VIEIRA, V. (2008) CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive System. Tese (Doutorado) 187p. – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2008.

XML W3C (2008) Especificação XML 1.0. World Wide Web Consortium Disponível em <<http://www.w3.org/TR/REC-xml>>. Acesso em 01/07/2011.