

INSTITUTO MILITAR DE ENGENHARIA

MODELO DE METADADOS PARA A DESCRIÇÃO DE DOCUMENTOS
ELETRÔNICOS NA WEB

POR

CÁSSIA MARIA BARRETO

DISSERTAÇÃO SUBMETIDA

COMO REQUISITO PARCIAL

PARA A OBTENÇÃO DO GRAU DE

MESTRE EM CIÊNCIAS EM SISTEMAS E COMPUTAÇÃO

Assinatura da Orientadora da Dissertação

ANA MARIA DE CARVALHO MOURA – Dr.Ing.

Assinatura da Co-Orientadora da Dissertação

MARIA LUÍZA MACHADO CAMPOS – Ph.D.

Rio de Janeiro - RJ

Agosto de 1999

A Deus, pela graça alcançada.

À minha mãe, por tudo....

Ao meu irmão Cláudio, sua esposa Conceição

e meus sobrinhos Natália e Vítor.

A meus chefes e companheiros de trabalho da DPMM

e todos os meus amigos.

AGRADECIMENTOS

A Deus, por cada minuto de graça concedida para que eu chegasse até aqui.

À Marinha do Brasil, na figura do Excelentíssimo Sr. Vice-almirante Luiz Eugênio de Albuquerque Lobo (*in memoriam*), pela confiança concedida para que eu pudesse aprimorar meus conhecimentos na área de informática.

À minha família, em especial à Maria Francisca, minha mãe, pela paciência, dedicação e apoio em todos os momentos, e as minhas tias Rita, Edith e Albertina, pelo cuidado e preocupação constantes com meu bem-estar.

Ao Instituto Militar de Engenharia, por me aceitar como aluna e pelos valiosos ensinamentos ministrados.

À professora Ana Maria do IME, por ter confiado em mim e permitido a minha permanência no curso, por todo o conhecimento transmitido, pela orientação dedicada, sempre segura e correta e, principalmente, pela amizade e incentivo para que eu terminasse este trabalho.

À professora Maria Luíza da UFRJ, pelas inestimáveis sugestões e orientação segura durante todo o desenvolvimento desta dissertação.

A todos os meus amigos da Diretoria do Pessoal Militar da Marinha, em especial à chefia, aos oficiais, às praças e aos civis do Departamento de Sistemas pela compreensão, incentivo e ambiente cooperativo.

Ao Comandante Carlos Eduardo Araújo Motta, vice-diretor da DPMM, pela confiança que sempre depositou em meu trabalho, pelo exemplo de determinação e liderança, pelo apoio, compreensão e incentivo para que eu terminasse este trabalho.

Às Tenentes Márcia Regina, Nadima, Nair e Comandante Audrey, pelo afeto, amizade, incentivo, companheirismo e suporte nas horas mais difíceis.

Ao Comandante Barros, da Diretoria de Administração da Marinha, pela amizade e incentivo para que eu continuasse este trabalho.

A todos os meus amigos do IME, em especial à Simone Garcia e Luiz Alexandre, pelo incentivo constante, ao Tenente Meyer e Hamilton pelos conhecimentos transmitidos e o auxílio prestado durante o difícil período de créditos.

Aos professores Rubens Melo, da PUC-RJ, Geraldo Xexéo, da UFRJ, e Cláudia Oliveira, do IME, por participarem da banca para avaliação desta dissertação.

Por fim, a todos aqueles que contribuíram, direta ou indiretamente, para a realização deste trabalho.

RESUMO

O crescente *status* alcançado pela Web como ferramenta para publicação de recursos, realçado pelos avanços na área de recuperação de informações tem conduzido a uma sobrecarga de informações de difícil gerenciamento. Nesse contexto, a utilização de metadados fornecidos pelos próprios autores de documentos da Web tem sido considerado por diversos pesquisadores como a solução mais adequada para tornar mais precisa a descoberta de documentos eletrônicos, uma vez que as estratégias de indexação automática dos conteúdos analisados, empregadas pela maioria das ferramentas atuais de pesquisa, não têm sido suficiente para atender à demanda cada vez maior de informações.

A iniciativa de metadados descrita neste trabalho é denominada Modelo de Objetos Digitais para Documentos Eletrônicos (MODDE). Seu objetivo é possibilitar a gerência de documentos da Web com base no seu conteúdo intelectual, além de auxiliar o usuário leigo na descrição adequada de suas propriedades segundo diversos padrões de metadados.

O modelo proposto é baseado em projetos de metadados recentes, tais como a arquitetura Warwick, desenvolvida para o ambiente da Web, e no conceito de objeto digital proposto pela arquitetura de Kahn e Wilensky para a representação de recursos de informação gerenciados por Bibliotecas Digitais distribuídas. Sua estrutura hierárquica multi-camadas possibilita a organização coerente e padronizada do conteúdo do documento juntamente com as descrições associadas a esse conteúdo.

ABSTRACT

The increasing status of the Web as a tool for resources publishing, emphasized by the progress in the information retrieval domain, has led to an information overhead, difficult to manage. In this context, the use of metadata provided by the own authors' Web documents has been considered by many researchers as the adequate solution to turn out to be more accurate the research of electronic documents. The automatic indexing strategies of documents contents used by most of the current search tools have not been adequate to answer to the information increasing requirements.

The metadata proposal described in this work is named Digital Objects Model for Electronic Documents (MODDE). Its purpose is to allow the management of documents on the Web, based on their intellectual content, helping the non-expert user to adequately describe his documents properties according to several metadata standards.

The proposed model is based on recent metadata projects, such as Warwick architecture, developed for the Web environment, and on the digital object concept proposed by Kahn and Wilensky architecture to represent information resources managed by distributed digital libraries. Its hierarchical multi-layer structure allows a coherent and standardized organization of the document content integrated to the descriptions associated to this content.

SUMÁRIO

RESUMO	iv
ABSTRACT	v
SUMÁRIO	vi
LISTA DE ILUSTRAÇÕES	x
LISTA DE TABELAS	xii
LISTA DE ABREVIATURAS E SÍMBOLOS	xiii
1 – INTRODUÇÃO	01
1.1 – OBJETIVO DA DISSERTAÇÃO	03
1.2 – ORGANIZAÇÃO DA DISSERTAÇÃO	04
1 – O AMBIENTE DE RECUPERAÇÃO DE INFORMAÇÕES DA WEB	06
1.3 – INTRODUÇÃO	06
1.4 – DOCUMENTOS ELETRÔNICOS NA WEB	08
2.2.1 – Conceitos Básicos	08
2.2.2 – Documentos SGML	10
2.2.3 – Documentos HTML	12
2.2.4 – Documentos XML	13
2.2.5 – Coleções de Documentos na Web	14
1.5 – BUSCA E LOCALIZAÇÃO DE DOCUMENTOS NA WEB	15
2.3.1 – Conceitos Básicos	15
2.3.2 – Principais Problemas	17
2.3.3 - O Papel do Metadado na Web	18
1.6 – CONSIDERAÇÕES FINAIS	20
2 – O EMPREGO DE METADADOS PARA A DESCRIÇÃO DE DOCUMENTOS ELETRÔNICOS NA WEB	22
1.7 – INTRODUÇÃO	22
1.8 – CLASSIFICAÇÕES DE METADADOS	23
1.8.1 – Classificação Geral	23
1.8.2 – Classificação Funcional	24
1.9 – EMPREGO DE METADADOS NA WEB	25

1.9.1	– Processo de Identificação de Recursos	26
1.9.2	– Processo de Descoberta de Recursos	28
1.9.3	– Processo de Recuperação de Recursos	29
1.10	– OBTENÇÃO DE METADADOS NA WEB	30
1.10.1	– Abordagem Automática	31
1.10.2	– Abordagem Manual	34
3.5	– CONSIDERAÇÕES FINAIS	36
3	– PADRÕES DE METADADOS PARA A DESCRIÇÃO DE CONTEÚDO	37
1.11	– INTRODUÇÃO	37
1.12	– PADRÕES PARA DOCUMENTAÇÃO DE ACERVOS ESPECÍFICOS	38
1.12.1	– <i>Machine Readable Catalogue</i> (MARC)	
	38	
1.12.1.1	– Análise do Padrão	41
1.12.2	– <i>Text Encoding Initiative Independent Headers</i> (TEI)	41
1.12.2.1	– Análise do Padrão	44
1.12.3	– <i>Encoded Archival Description</i> (EAD)	45
1.12.3.1	– Análise do Padrão	47
1.12.4	– <i>Government Information Locator Service</i> (GILS)	48
1.12.4.1	– Análise do Padrão	50
1.12.5	– Quadro Comparativo entre os Padrões da Categoria	50
1.13	– PADRÕES PARA A DESCOBERTA DE RECURSOS DA WEB	51
1.13.1	– <i>Internet Anonymous Ftp Archive</i> (IAFA)	
	52	
1.13.1.1	– Análise do Padrão	55
1.13.2	– <i>Summary Object Interchange Format</i> (SOIF)	55
1.13.2.1	– Análise do Padrão	57
1.13.3	– <i>Dublin Core</i>	
	57	
1.13.3.1	– Análise do Padrão	60
1.13.4	– Quadro Comparativo entre os Padrões da Categoria	60
1.14	– PROPOSTA DE CLASSIFICAÇÃO FUNCIONAL DO METADADO	61
1.15	– ANÁLISE COMPARATIVA DOS PADRÕES DE METADADOS	67

1.16–	CONSIDERAÇÕES FINAIS	68
4 –	ARQUITETURAS GENÉRICAS DE METADADOS	70
1.17–	INTRODUÇÃO	70
1.18–	ARQUITETURA WARWICK	71
5.2.1 –	Conceitos Básicos	71
5.2.2 –	Modelo de Dados	72
5.2.3 –	Implementação na Linguagem HTML	74
5.2.4 –	Implementação segundo a Orientação a Objeto	75
5.2.5 –	Implementação segundo Conceitos de Objetos Distribuídos	76
5.2.6 –	Arquitetura de Kahn e Wilensky segundo a Arquitetura Warwick	79
5.2.7 –	Extensão da Arquitetura Warwick (Modelo “DARs”)	80
1.19–	META CONTENT FRAMEWORK (MCF)	82
1.19.1 –	Conceitos Básicos	82
1.19.2 –	Modelo de Dados	83
1.19.3 –	Sistema Inicial de Tipos da MCF	84
1.19.4 –	Implementação da MCF na Linguagem XML	87
1.20–	RESOURCE DESCRIPTION FORMAT	88
1.20.1 –	Conceitos Básicos	88
1.20.2 –	Modelo de Dados	89
1.20.3 –	Implementação da RDF na Linguagem XML	91
1.21–	ESQUEMA RDF	92
1.21.1 –	Sistema Inicial de Tipos do Esquema RDF	92
1.22–	ARQUITETURA DE MODELAGEM DE QUATRO NÍVEIS	95
1.23–	CONSIDERAÇÕES FINAIS	96

5 – PROPOSTA DE UM MODELO DE METADADOS PARA A DESCRIÇÃO DE DOCUMENTOS NA WEB	100
1.24– INTRODUÇÃO	100
1.25– ASPECTOS CONSIDERADOS	101
1.25.1 – Organização Externa	103
6.2.2 – Conteúdo Intelectual	103
6.2.3 – Estrutura	104
6.2.4 – Organização Interna	105
6.2.5 – Formato	106
6.2.6 – Relacionamentos com Outros Recursos	106
1.26- PROPOSTA DE UMA ESTRUTURA ORGANIZACIONAL PARA DOCUMENTOS ELETRÔNICOS NA WEB	107
1.27– MODELO DE OBJETOS DIGITAIS PARA DOCUMENTOS ELETRÔNICOS (MODDE)	110
1.28– EXEMPLO DE UTILIZAÇÃO DO MODDE NA WEB	116
6.5.1 – Estudo de Caso	116
6.5.2 – Análise do Exemplo	119
6.5.3 – Modelando o Exemplo Passo a Passo	121
1.29– CONSIDERAÇÕES FINAIS	131
6 – DEFINIÇÃO DE UM AMBIENTE PARA INSTANCIAÇÃO E UTILIZAÇÃO DO MODDE	133
1.30– INTRODUÇÃO	133
1.31– DEFINIÇÃO DE UMA ARQUITETURA DE METADADOS PARA O MODDE	134
1.31.1 – Conceitos Básicos	134
1.31.2 – Sistema Inicial da Arquitetura Proposta	137
1.31.3 – Sistema de Tipos Definido para o MODDE	142
1.31.4 – Definição de Modelos para Descrição do MODDE	144
1.32– PROPOSTA DE UM AMBIENTE PARA INSTANCIAÇÃO DO MODDE	150

1.32.1 – Ambiente de Instanciação do Registro de Metadados	151
1.32.2 – Ambiente de Instanciação do Recipiente de Pacotes	152
1.32.3 – Ambiente de Instanciação do Repositório de Objetos Digitais	157
1.33– UTILIZAÇÃO DO MODDE NA WEB	158
1.33.1 – Como Arquivo Independente	160
1.33.2 – Embutido em um Documento HTML	161
1.34– CONSIDERAÇÕES FINAIS	161
7 – CONCLUSÃO	162
1.35– PRINCIPAIS CONTRIBUIÇÕES	163
1.36– SUGESTÕES PARA TRABALHOS FUTUROS	165
REFERÊNCIAS BIBLIOGRÁFICAS	167

LISTA DE ILUSTRAÇÕES

FIGURA 2.1 – Hierarquia de Objetos de Hipertexto	08
FIGURA 2.2 – Documento com Estrutura de Ligações não Tipadas	09
FIGURA 2.3 – Documento com Estrutura de Ligações Tipadas	11
FIGURA 2.4 – Estrutura de um Documento XML	14
FIGURA 2.5 – Exibição Parcial da Árvore de Pesquisa YAHOO	15
FIGURA 4.1 - Formato de um Registro MARC	39
FIGURA 5.1 – Componentes da Arquitetura Warwick	73
FIGURA 5.2 – Recipiente Internamente Referenciado	73
FIGURA 5.3 – Recipiente Externamente Referenciado	74
FIGURA 5.4 – Componentes da Arquitetura Warwick segundo a Orientação a Objeto	76
FIGURA 5.5 – Estrutura de um Objeto Digital	78
FIGURA 5.6 – MetaObjeto ou ObjetoCatálogo	78
FIGURA 5.7 – Estrutura de um Objeto Digital segundo a Arquitetura Warwick	80
FIGURA 5.8 – Contextos Distintos de Uso do Dado / Metadado	80
FIGURA 5.9 – Componente Catálogo da Arquitetura Warwick	82

FIGURA 5.10 – Modelo de Dados da MCF	83
FIGURA 5.11 – Definição de um Esquema de Metadados segundo Conceitos da MCF	86
FIGURA 5.12 – Representação de um Esquema de Metadados Instanciado	86
FIGURA 5.13 – Elemento XML	87
FIGURA 5.14 – Declaração RDF	89
FIGURA 5.15 – Recurso RDF	90
FIGURA 5.16 – Coleções RDF	91
FIGURA 5.17 – Hierarquia de Classes do Esquema RDF	93
FIGURA 5.18 – Arquitetura de Modelagem de Quatro Níveis	95
FIGURA 6.1 – Documento Eletrônico segundo a Estrutura Organizacional Proposta	102
FIGURA 6.2 – Estruturas Lógicas e Físicas em um Documento XML	104
FIGURA 6.3 – Estrutura Organizacional para Documentos Eletrônicos na Web	107
FIGURA 6.4 – Documento Eletrônico Multi-camadas	110
FIGURA 6.5 – Tipos de Objetos Digitais	111
FIGURA 6.6 – Exemplo de Relacionamentos entre Recursos da Web	117
FIGURA 6.7 – Representação do Exemplo segundo o Modelo de Documento Proposto	120
FIGURA 6.8 – Modelagem dos ObjetoColeção e ObjetoAssociação	121
FIGURA 6.9 – Modelagem do ObjetoIntelectual	123
FIGURA 6.10 – Modelagem do ObjetoDeExpressão	124
FIGURA 6.11 – Modelagem do ObjetoDeExpressão – continuação	126
FIGURA 6.12 – Modelagem dos ObjetoDeComposição e ObjetoEstrutural	127
FIGURA 6.13 – Modelagem dos ObjetoDeComposição e ObjetoEstrutural – continuação	128
FIGURA 6.14 – Modelagem do ObjetoInstância.	129
FIGURA 7.1 – Camadas da Arquitetura de Metadados Proposta	135
FIGURA 7.2 – Exemplo da Arquitetura de Metadados Proposta	137
FIGURA 7.3 – Sistema Inicial da Arquitetura de Metadados Proposta	138
FIGURA 7.4 – Componentes da Arquitetura de Metadados Proposta	150
FIGURA 7.5 - Ambiente de Instanciação do MODDE	151
FIGURA 7.6 – Classe Descritor	152
FIGURA 7.7 – Classe Pacote	152
FIGURA 7.8 – Classe PacoteDeMetadados	153

LISTA DE TABELAS

TABELA 4.1 - Categoria Documentação (Quadro Comparativo)	50
TABELA 4.1 - Categoria Documentação (Quadro Comparativo) – continuação	51
TABELA 4.2 - Elementos do Modelo Dublin Core	59
TABELA 4.3 - Categoria Descoberta (Quadro Comparativo)	60
TABELA 4.3 - Categoria Descoberta (Quadro Comparativo) - continuação	61
TABELA 4.4 – Metadados segundo Classificação Funcional Proposta	66
TABELA 4.4 – Metadados segundo Classificação Funcional Proposta – continuação	67

LISTA DE ABREVIATURAS E SÍMBOLOS

CDIF	("CASE Data Interchange Format")
CGI	("Common Gateway Interface")
CORBA	("Common Object Request Broker Architecture")
DARs	("Distributed Active Relationship")
DC	("Dublin Core")
DLG	("Directed Linked Graph")
DTD	("Document Type Definition")
EAD	("Encoded Archival Description")
FGDC	("Federal Geographic Data Committee")
FTP	("File Transfer Protocol")
GIF	("Graphics Interchange Format")
GILS	("Government Information Locator Service")
HTML	("HyperText Markup Language")
HTTP	("HyperText Transfer Protocol")
IAFA/Whois++	("Internet Anonymous Ftp Archive with Whois++ Protocol")
IETF	("Internet Engineering Task Force")

IFLA	("International Federation Library Association")
IR	("Information Retrieval")
MARC	("Machine Readable Catalogue")
MCF	("Meta Content Framework")
MIME	("Multipurpose Internet Mail Extensions")
MODDE	Modelo de Objetos Digitais para Documentos Eletrônicos
MOF	("Meta Object Facility")
NCSA	("National Center for Supercomputing Applications")
OCLC	("Online Computer Library Center")
PICS	("Platform for Internet Content Selection")
RAP	("Repository Access Protocol")
RDF	("Resource Description Framework")
ROADS	("Resource Organisation and Discovery in Subject-based Services")
SAIF	("Spatial Archive and Interchange Format")
SGBD	Sistema Gerenciador de Banco de Dados
SGML	("Standard Generalized Markup Language")
SOIF	("Summary Object Interchange Format")
SQL	("Structured Query Language")
TEI	("Text Encoding Initiative")
UML	("Unified Modeling Language")
URI	("Universal Resource Identifier")
URN	("Uniform Resource Name")
URL	("Universal Resource Locator")
WWW	("World Wide Web")
W3C	("World Wide Web Consortium")
XML	("Extensible Markup Language")

CAPÍTULO 1

INTRODUÇÃO

Desde o advento da Web, em 1993, a Internet tem evoluído para um ambiente de computação distribuída de âmbito global, caracterizado, principalmente, por mudanças e crescimento acelerados, possuindo a capacidade de integrar recursos de informação que estão disponíveis nas mais diversas mídias (som, vídeo, imagens, textos, etc.) e meios de armazenamento.

Um dos efeitos diretos desse processo de globalização da informação são os projetos que envolvem recursos de informação oriundos de áreas diversas. Sem uma documentação que descreva esses recursos, a compreensão da informação armazenada fica restrita aos especialistas em suas respectivas áreas de atuação. Dados armazenados somente são úteis quando inseridos em um contexto conhecido e bem especificado. A documentação que descreve o dado armazenado é denominada *metadado* e é essencial a plena utilização dos recursos disponíveis.

Em uma abordagem tradicional, o propósito do uso de metadados é divulgar e manter sob controle os dados produzidos e manipulados no âmbito corporativo. Sob uma ótica mais atual, a importância de utilização de metadados pode ser observada em

diferentes áreas, aplicações e tecnologia emergentes. De uma forma genérica, metadados possuem usos distintos, fortemente relacionados ao ambiente no qual o dado é armazenado e organizado. Em sistemas gerenciadores de arquivos, por exemplo, o uso dos dados armazenados depende basicamente da descrição de seu formato, o que pode ser considerado um exemplo rudimentar de metadado. No contexto dos bancos de dados não-convencionais e dos gerenciadores de dados multimídia, o uso de metadados específicos ao tipo da mídia possibilita a implementação de consultas baseadas em conteúdo (JAIN & HAMPAPUR, 1994) e a integração de informações oriundas de mídias distintas. Metadados também podem ser usados em ambientes de banco de dados (heterogêneos ou federados) como componente de interoperabilidade, integrando esquemas, visões e provendo a troca de informações entre ambientes e plataformas distintas (SELIGMAN & ROSENTHAL, 1996) (PICCININI, 1998).

Mais recentemente, metadados têm sido considerados essenciais aos ambientes de Data Warehouses, possibilitando determinar a origem de um item de dado em particular; para onde os dados foram migrados; as transformações ocorridas nos dados, etc. E nas corporações cujos ambientes são baseados na computação cliente/servidor, metadados emergem como elementos críticos para promover o intercâmbio de informações entre ferramentas de vendedores distintos (MDIS).

Na Web, metadados têm ganhado destaque em diversos projetos envolvendo dados georeferenciados, devido à variedade e complexidade dos tipos de dados envolvidos, bem como a diversidade dos relacionamentos entre eles. Para enfatizar a importância do uso de metadados nos SIGs, diversos padrões específicos para o dado georeferenciado foram desenvolvidas. FGDC (*“Content Standards for Digital Geospatial Metadata”*) (FGDC) e SAIF (*“Spatial Archive and Interchange Format”*) (SAIF) são exemplos recentes. O projeto SEQUOIA 2000 (ANDERSON & STONEBRAKER, 1994), por exemplo, desenvolveu um esquema para a descrição de imagens de satélites baseado no padrão SAIF.

No contexto desta dissertação, o enfoque do emprego de metadados tem por objetivo tornar mais precisa a descoberta de documentos eletrônicos, que representam a maioria dos recursos de informação da Web, uma vez que as estratégias empregadas pelas ferramentas atuais de indexação e pesquisa automática de recursos da Web (Alta Vista,

por exemplo) não tem sido suficientes para atender a demanda cada vez maior de informações.

Uma abordagem baseada em metadados fornecidos pelos próprios autores de documentos têm sido considerada por diversos pesquisadores de metadados como a mais adequada em função das características mais evidentes do ambiente da Web que implica na impossibilidade de estabelecimento de um registro central de metadados.

Diversos padrões de metadados tem sido propostos ou adaptados para emprego na Web. O problema na utilização de alguns padrões decorre da dificuldade encontrada pelo usuário leigo no emprego de regras de catalogação que, na maioria das vezes, não se aplicam de forma adequada ao ambiente da Web. Desde 1995, no entanto, a partir de um esforço conduzido por diversas comunidades utilizadoras de recursos da Web, o modelo Dublin Core (WEIBEL et alii, 1995), projetado para uso específico na Web, vem sendo progressivamente refinado com o objetivo de se tornar o padrão de direito e de fato para a descrição dos recursos da Web.

A despeito desses esforços na direção de um padrão de metadados único e monolítico, outras comunidades da Web têm investido na direção das arquiteturas de metadados, na qual os diversos padrões propostos para a Web são considerados como abordagens complementares. Essas arquiteturas baseiam-se na implementação de mecanismos que possibilitam a tradução de um termo oriundo de um padrão em outro de forma a prover interoperabilidade entre aplicações distintas. A favor desta abordagem estão os inúmeros acervos já documentados e catalogados por padrões tradicionais, tais como MARC (USMARC, 1996) e que são tornados disponíveis através da Web.

A tendência atual reforça a utilização dos descritores do padrão Dublin Core, por parte dos autores de documento, embutidos, principalmente, nos cabeçalhos dos documentos HTML. Entretanto, devido à necessidade crescente de inúmeras aplicações trocarem informações estruturadas na Web, a arquitetura de metadados que está sendo desenvolvida sob os auspícios do consórcio W3C (W3C) para este ambiente objetiva ser flexível o bastante a fim de se ajustar facilmente às novas requisições dos usuários. Nesse sentido, o mais provável é que a solução adotada empregue um misto de padrões para a descrição de conteúdo e de linguagens para a representação de dados, tal como a XML (XML, 1996), segundo algum modelo (baseado em grafos, por exemplo) para a representação do conhecimento.

1.1 – OBJETIVO DA DISSERTAÇÃO

O objetivo deste trabalho é desenvolver um modelo baseado em metadados para a descrição e representação das principais características de um documento eletrônico na Web, de modo a tornar mais efetiva a sua localização pelas ferramentas de pesquisa da Web.

O modelo proposto deve possibilitar a organização coerente e padronizada das descrições relacionadas ao documento (juntamente com o seu conteúdo), maximizando a interoperabilidade e propiciando o compartilhamento e reutilização dessas descrições. Como é bastante improvável a adoção de um padrão único de metadados para a Web, esse modelo deve possibilitar a descrição das propriedades de um documento segundo diversas abordagens descritivas e ser flexível o bastante a fim de se ajustar facilmente aos novos requisitos das aplicações dos usuários. Tendo em vista uma futura implementação desse modelo em ambiente de banco de dados, estruturas específicas serão também definidas.

Para o seu desenvolvimento, serão considerados aspectos tais como: características de um documento eletrônico na Web; padrões de metadados existentes e em desenvolvimento para a Web; tipos de metadados para a descrição de documentos eletrônicos na Web; estratégias usadas para obter, manualmente e/ou automaticamente, o metadado; formas de utilização do metadado; sintaxes para intercâmbio de metadados; formas de relacionamento entre dado e metadado; e formas de armazenamento do metadado.

1.2 - ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada em oito capítulos, da seguinte maneira:

No Capítulo 1 é apresentada a introdução do trabalho contendo a motivação para o seu desenvolvimento e definição do objetivo a ser alcançado.

No Capítulo 2 é descrito o ambiente de recuperação de informações da Web, destacando-se algumas características de seus recursos de informações, em particular os documentos eletrônicos. Também são apresentadas as principais abordagens para localização de documentos eletrônicos na Web, os problemas decorrentes de cada

abordagem e os fatores que justificam o emprego de metadados como auxílio a diversas funcionalidades implementadas nesse ambiente.

No Capítulo 3 o metadado é descrito e organizado segundo as principais classificações propostas na literatura. Também são descritas as formas de utilização do metadado nos processos de identificação, descoberta e recuperação de recursos e as abordagens empregadas na Web para a sua obtenção.

No Capítulo 4 são descritos os principais padrões de metadados citados na literatura para a descrição de documentos eletrônicos e outros recursos de informação. Na Seção 4.4 é apresentada uma proposta de classificação funcional de metadados que foi baseada nos componentes oriundos dos padrões de metadados analisados.

No Capítulo 5 são apresentadas as principais arquiteturas de metadados que estão sendo propostas para a implementação no ambiente da Web. Algumas enfocam requisitos de comunidades específicas, tais como a de Bibliotecas Digitais. Entretanto, todas possuem um requisito comum: prover interoperabilidade entre aplicações distintas.

No Capítulo 6 é apresentado o Modelo de Objetos Digitais para Documentos Eletrônico (MODDE) que possibilita a gerência de um documento ou de uma coleção de documentos com base no seu conteúdo intelectual. Um estudo de caso demonstrando a funcionalidade do modelo na Web também é apresentado.

No Capítulo 7 é descrito um ambiente para instanciação e utilização do MODDE. São componentes desse ambiente: uma arquitetura de metadados multi-camadas e um sistema inicial para a declaração de descritores oriundos de diversas abordagens descritivas; o *registro* de metadados para armazenamento dos descritores declarados; o componente *recipiente* para o armazenamento de pacotes contendo modelos, meta-modelos e meta-meta-modelos instanciados; e o componente *repositório* para armazenamento dos objetos digitais criados.

Finalmente, o Capítulo 8 contém a conclusão do trabalho proposto. Também são apresentadas algumas sugestões para trabalhos futuros relacionados ao tema desta dissertação.

CAPÍTULO 2

O AMBIENTE DE RECUPERAÇÃO DE INFORMAÇÕES DA WEB

2.1 – INTRODUÇÃO

A Internet é hospedeira de um conjunto de recursos que são gerenciados de forma autônoma e distribuída. Nesse ambiente, existe uma tendência cultural forte contra o estabelecimento de controle centralizado ou mesmo de um registro centralizado de recursos. Essa tendência vai além dos aspectos técnicos, embora por si só já sirvam para justificá-la, pois envolvem questões relacionadas à escalabilidade e à segurança da rede.

Como decorrência dessa descentralização, os recursos de informação acessíveis através da Web são extremamente heterogêneos quanto à natureza, granularidade, tamanho e cobertura. Esses recursos incluem (NIDR, 1994): serviços em geral, tais como os de correspondência eletrônica, previsão de tempo, seções interativas de áudio e vídeo, etc.; agregações de informações (organizadas ou não de forma hierárquica), tais como banco de dados, arquivos acessíveis através do protocolo Ftp, arquivos de listas de correspondência, grupos de notícias (“*newsgroups*”), etc.; e objetos de informação, que são acessíveis através de algum protocolo da Internet tais como Ftp, Telnet, Z39.50, HTTP, etc. Objetos de informação são recursos tais como os arquivos de imagens, programas,

áudio, vídeo e, em maior número, documentos eletrônicos, que são o enfoque deste trabalho.

Diariamente, uma grande diversidade de usuários tenta localizar os recursos disponíveis através da Web. De uma forma genérica, em termos da qualidade, consistência e coerência das informações pesquisadas, as expectativas desses usuários são baseadas geralmente em ambientes tais como os bancos de dados comerciais ou os catálogos de bibliotecas “on-line” que, ao contrário do ambiente distribuído e autônomo da Internet, são ambientes projetados para a recuperação de informações.

O problema em se tentar estender as metodologias empregadas pelos sistemas tradicionais de recuperação de informações ao ambiente da Web se deve ao fato de que a faixa de consultas possíveis é muito ampla, bem como a expectativa de seus usuários. Nesse ambiente, a busca por informações é geralmente um processo demorado e não a formulação de uma consulta simples. A consulta de um usuário é refinada de forma interativa à medida que este obtém uma melhor compreensão dos tipos de recursos de informação que são relevantes às suas necessidades em um contexto em particular.

Como a Internet não se constitui em um corpo de informação organizado, selecionado e gerenciado de forma única, os sistemas de recuperação de informações usados pelas ferramentas de pesquisa que operam em seus espaços são geralmente projetados para suportar uma maior precisão na especificação dos critérios de seleção, bem como para auxiliar o usuário a gerenciar conjuntos grandes de resultados, seja através de sucessivos refinamentos da consulta ou resumindo-se esses conjuntos de várias formas.

O aumento exponencial de recursos de informação na Web, no entanto, vem acarretando diversos problemas relacionadas ao uso dessas ferramentas, dentre os quais destaca-se a pouca precisão do conjunto resultado de uma consulta ocasionado principalmente pela falta de informações contextuais sobre os dados pesquisados. Nesse caso, a utilização de metadados fornecidos pelos próprios autores de documentos vem sendo considerada a solução mais adequada para tornar mais efetiva a busca de informações na Web.

De maneira similar aos sistemas de recuperação mais antigos que operavam sobre registros de metadados muito simples, baseados na catalogação descritiva tradicional, o uso crescente de informações estruturada na Web pode ser considerado um fator muito significativo no projeto dos futuros sistemas de recuperação de informações, tal como foi o

emprego dos algoritmos de análise estatística de texto pelas ferramentas de pesquisa da Web.

Neste capítulo serão apresentadas inicialmente, na Seção 2.2, as principais características de um documento que possam influenciar no projeto futuro das ferramentas de pesquisa e recuperação de recursos, considerando o ambiente heterogêneo e distribuído da Web. Também são descritos alguns padrões relacionados a documentos eletrônicos que estão sendo propostos ou que possuem uso corrente na Web tendo em vista a forma como a informação contida nesses documentos é estruturada.

Na Seção 2.3 serão apresentadas as principais abordagens para localização e acesso a documentos eletrônicos na Web, bem como os problemas decorrentes de cada abordagem. O enfoque principal da seção é o papel do metadado provendo suporte a diversas funcionalidades da Web de forma a torná-las mais efetivas.

2.2 – DOCUMENTOS ELETRÔNICOS NA WEB

2.2.1 – Conceitos Básicos

Documentos eletrônicos constituem a grande maioria dos recursos de informação da Web. Segundo Thuring (THURING, 1991), um documento eletrônico consiste de uma parte de conteúdo, uma de organização e outra de apresentação. É composto por um conjunto de objetos denominados objetos de hipertexto, que representam os nós e as ligações desse documento.

Conforme pode ser visualizado na Figura 2.1, nós e ligações podem representar tanto o conteúdo quanto a estrutura do documento, sendo denominados, respectivamente, nó (ou ligação) de conteúdo e nó (ou ligação) de estrutura.

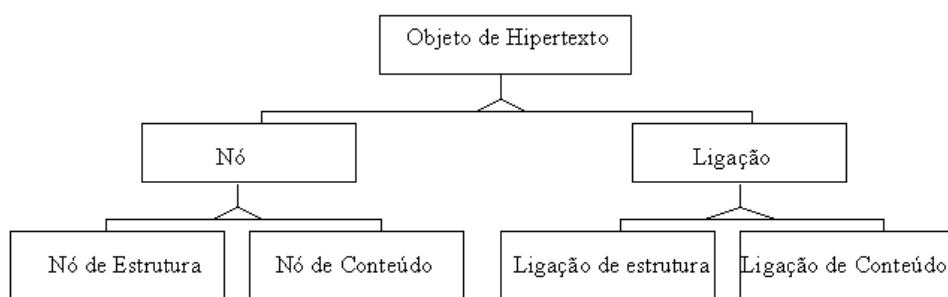


FIGURA 2.1 - Hierarquia de Objetos de Hipertexto

Um nó de conteúdo pode ser do tipo composto ou atômico, sendo que a composição de nós desse tipo é empregada para estruturar, de forma hierárquica, o conteúdo do documento em sub-unidades de informação, segundo um domínio específico. Uma ligação de conteúdo, por sua vez, conecta nós de conteúdo segundo relacionamentos semânticos. Pode ser classificada como *ligação sem rótulo* quando não especifica a natureza do relacionamento, caso da maioria dos documentos acessíveis através da Web; *ligação com rótulo genérico*, quando descreve o relacionamento de uma forma global, usando termos genéricos do tipo “ÉDiscutidoPor”, por exemplo; ou *ligação com rótulo específico*, tal como “ÉCríticoPor”, quando descreve a semântica do relacionamento dentro de um domínio de aplicação específico. Já um nó de estrutura serve para organizar nós e ligações de conteúdo enquanto que uma ligação de estrutura conecta nós de estrutura. A forma como a estrutura e o conteúdo de um documento são exibidos, quer seja na forma textual, gráfica, ou combinada, se refere a parte de apresentação de um documento.

Diversos aspectos relacionados a documentos eletrônicos são relevantes ao projeto das ferramentas de pesquisa e recuperação de recursos que operam no ambiente da Web, conforme destacado a seguir (WARD & WOOD):

- *Estrutura interna*: na Web, documentos eletrônicos variam sobremaneira quanto à estrutura interna, podendo consistir de páginas, capítulos, páginas de rosto, índice, ilustrações, etc., e se relacionar, de forma arbitrária, com vários outros recursos de informação, conforme o documento com estrutura de ligações não tipadas apresentado na Figura 2.1. De uma forma geral, a estrutura das ligações da maioria dos documentos da Web não permite distinguir quais componentes fazem parte de sua estrutura interna daqueles que são apenas referências externas (FERBER, 1997).

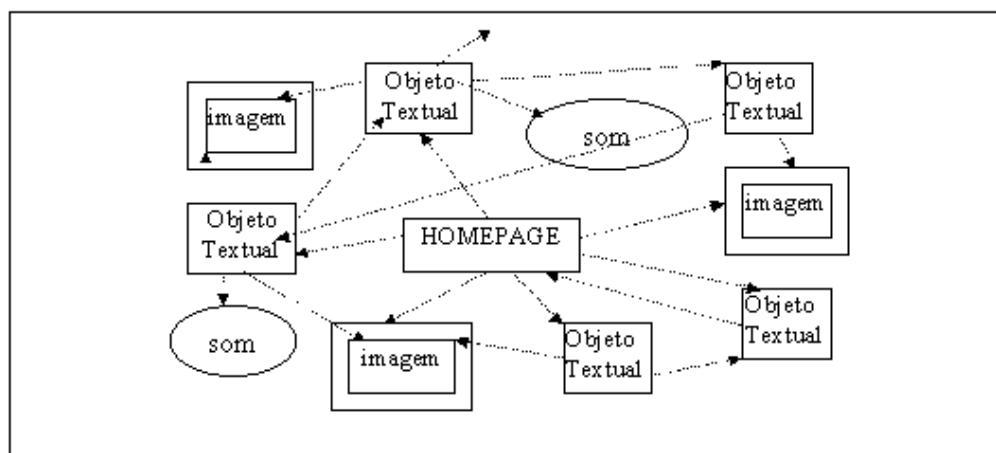


FIGURA 2.2 - Documento com Estrutura de Ligações não Tipadas

- *Versões*: devido à facilidade de mudanças no ambiente de publicação da rede, diferentes versões de documentos digitais são criadas continuamente. Essas versões podem diferir em um único bit ou serem completamente diferentes;
- *Formato*: apesar da maioria dos documentos da Web ser publicada no formato HTML, para algumas aplicações outros formatos (PDF, RTF, etc.) são mais apropriados. Recursos de informação mais antigos, por exemplo, se encontram disponíveis apenas em papel, fazendo com que sua liberação para o meio “on line” só seja possível na forma de imagens de páginas escaneadas. Para a indexação do conteúdo de um documento, por exemplo, o formato puramente textual é mais apropriado do que o formato postScript. Para a exibição página a página, uma coleção de imagens do documento é mais apropriada do que outra forma de apresentação.
- *Qualidade*: existe uma variação grande na qualidade de conteúdo dos recursos disponíveis através da Web. Enquanto alguns recursos são mantidos por organizações (comerciais ou privadas) como parte de suas missões ou atividades básicas, outros são tornados disponíveis através de contribuições individuais e armazenados nos diversos provedores de informação disponíveis na Web sem qualquer compromisso com a qualidade da informação que está sendo prestada.
- *Volatilidade*: a Internet constitui atualmente um ambiente de computação distribuída de grande escala que é caracterizado pelo crescimento e mudanças aceleradas. Seus recursos são extremamente voláteis, sendo constantemente criados, movimentados de um servidor para outro, algumas vezes desaparecem ou se tornam obsoletos devido a falta de manutenção e suporte. Basicamente, esses recursos não seguem nenhum tipo de política definida para o escopo de conteúdo.

Nas próximas seções serão descritos alguns padrões de documentos propostos ou que possuem uso corrente na Web. Alguns lidam com a estrutura conceitual do documento, como o padrão SGML, enquanto outros lidam com a estrutura do documento para apresentação no hipertexto, como as linguagens HTML e XML.

2.2.2 - Documentos SGML

A linguagem SGML (*Standard Generalized Markup Language*) (SGML) é um padrão internacional para a representação conceitual de documentos. Representa,

basicamente, um sistema de marcação generalizada e não um *software*. Marcação é qualquer informação adicional acrescentada ao texto de um documento. A maioria dos processadores de textos inserem marcadores em formato binário nos documentos criados para descrever a sua aparência. Já a marcação generalizada ou descritiva é explícita e indica a importância estrutural de uma porção de texto dentro de um documento.

A SGML fornece um esquema de marcação simples, independente de plataforma e extremamente flexível para a descrição da estrutura lógica de um documento. Já a formatação (ou *layout*) do documento tem que ser anexada por uma aplicação específica, pois a linguagem não possui mecanismo para definição de estilo.

Em um documento SGML é possível distinguir as partes relativas a sua estrutura interna (tais como capítulos, partes, tópicos e figuras) daquelas pertencentes a documentos distintos conforme pode ser visualizado na Figura 2.3. Nesse exemplo é apresentado um documento cuja estrutura das ligações é tipada permitindo distinguir a hierarquia existente entre os componentes do documento.

Cada componente estrutural em um documento SGML é denominado de *elemento*. Um elemento pode conter (de forma recursiva) vários outros elementos, sendo que cada elemento é marcado por rótulos de início ("`<nomeElemento>`") e fim ("`</nomeElemento>`"). O nome do elemento identifica o seu tipo. Um elemento pode ser vazio ou conter um valor. Um elemento pode conter um ou vários atributos. Cada atributo possui um nome e um valor associado.

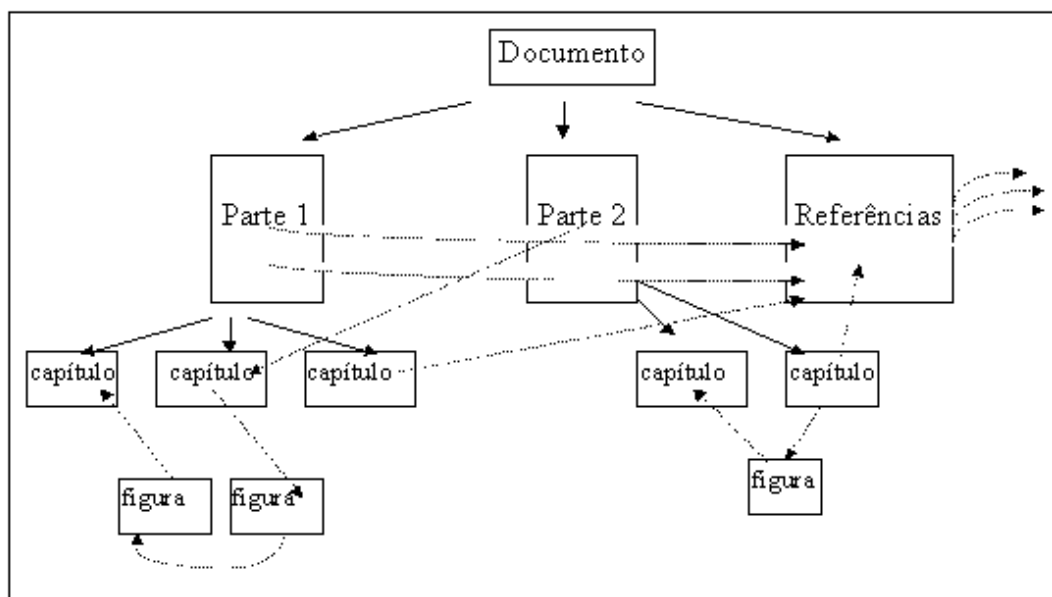


FIGURA 2.3 - Documento com Estrutura de Ligações Tipadas

O conjunto de elementos que define a estrutura lógica de uma classe (ou tipo) de documento SGML é definido através de um componente denominado Definição de Tipo de Documento (DTD - “*Document Type Definition*”) que é, essencialmente, uma gramática livre de contexto na qual o lado direito das regras corresponde a expressões regulares.

A SGML não impõe qualquer conjunto específico de tipos de elementos a seus usuários. Em vez disso, o padrão possibilita que estes criem seus próprios tipos de elementos, por isso é denominada de *meta-linguagem*, ou seja, uma linguagem para definir linguagens de marcação. As linguagens HTML e XML, descritas a seguir, são exemplos de aplicações baseadas na SGML.

2.2.3 - Documentos HTML

No contexto da Internet, a Web não é uma rede de computadores, mas sim uma aplicação que utiliza os seus recursos como meio de comunicação e transporte de informações. O poder da Web se deve a capacidade de interligar recursos de informação através de um sistema de hipertextos implementado por ferramentas de navegação (*Netscape Navigator*, *Internet Explorer*, *Mosaic*, etc.), que se comunicam com servidores especializados.

Uma linguagem de marcação, denominada HTML (*“HyperText Markup Language”*) (HTML), baseada na linguagem SGML, foi criada especialmente para codificação dos documentos disponíveis através dos servidores Web e interpretados pelos navegadores. Na Web, o protocolo HTTP (*“HyperText Transfer Protocol”*) é utilizado para o transporte de documentos HTML. É orientado a objeto, sem controle de estados e foi desenvolvido especialmente para o ambiente da Web: possui a velocidade necessária ao estabelecimento de conexões cliente-servidor em um sistema de informação hipermídia distribuído, provendo um tempo de resposta em torno de 100 ns.

A principal vantagem da linguagem HTML é a facilidade que oferece para construção de documentos hipermídia e, de alguma forma, a separação que esta provê dos conceitos de apresentação e de conteúdo de um documento. A forma como este é visualizado pelo usuário depende apenas de ajustes nos parâmetros de ambiente do navegador.

Ao contrário da SGML, a linguagem HTML oferece um conjunto de rótulos de marcação padrão, não permitindo aos usuários a criação de seus próprios rótulos. Também não possibilita a quebra de páginas longas em seções lógicas (a não ser com o uso de *frames*, o qual não é aceito por alguns navegadores). A HTML mantém a informação de apresentação (*“layout”*) como parte da marcação da página o que, em alguns casos, faz com o documento não tenha a aparência correta em determinadas plataformas. Também não faz uso de padrões para a especificação dos tipos das hiperligações, de forma que o navegador pudesse propiciar uma melhor navegação.

2.2.4 - Documentos XML

Em meados de 1996, um grupo de aproximadamente 80 peritos na linguagem SGML uniu forças ao Consórcio W3C (*World Wide Web Consortium*) para formar um grupo de trabalho com o objetivo de definir uma linguagem de marcação que tivesse o poder da SGML e, ao mesmo tempo, fosse fácil de ser implementada na Web. Os requisitos definidos para a linguagem foram: dar suporte à marcação generalizada na Web; produzir documentos válidos segundo as regras da SGML; fornecer suporte à hiperligações compatíveis com a abordagem implementada pela linguagem HTML; e fornecer um mecanismo de folha de estilo genérico e poderoso.

A especificação inicial da linguagem foi emitida em novembro de 1996. Formalmente, a XML (“Extensible Markup Language”) (XML) é um *perfil* (ou subgrupo) da SGML, o que significa que foram selecionados dessa linguagem apenas os recursos necessários para a Web, de modo a tornar disponível aos seus usuários a marcação generalizada. Com a XML cada usuário pode definir seus próprios conjuntos de rótulos, ou seja, produzir documentos auto-descritivos, que podem ser validados.

O modelo de dados da XML é muito simples, provendo a base para a construção de documentos mais complexos. Basicamente, um documento XML pode ser visto como uma linearização da estrutura de uma árvore, onde cada nó contém uma série de caracteres. A estrutura da árvore e o conteúdo de seus nós formam o conteúdo de informação do documento XML. Apesar de possuir estrutura essencialmente hierárquica, a XML possibilita a conexão entre nós arbitrários, conforme apresentado na Figura 2.4 onde um dos elementos da árvore *p* faz referência a um outro elemento da árvore cujo identificador é “X7”.

A aparência de um documento XML é determinada por uma ou mais folhas de estilo, denominadas XS, que podem ser anexadas ao documento e selecionadas pelo usuário para a exibição de acordo com o seu ambiente.

O emprego da XML será mais adequado para pessoas ou organizações que possuam recursos de informação que não se encaixam nos moldes da HTML e que porventura desejam tornar esses dados disponíveis para a Web. São exemplo de dados típicos a serem representados na XML: livros, transações financeiras (EDI), manuais técnicos, fórmulas químicas, registros médicos, etc.

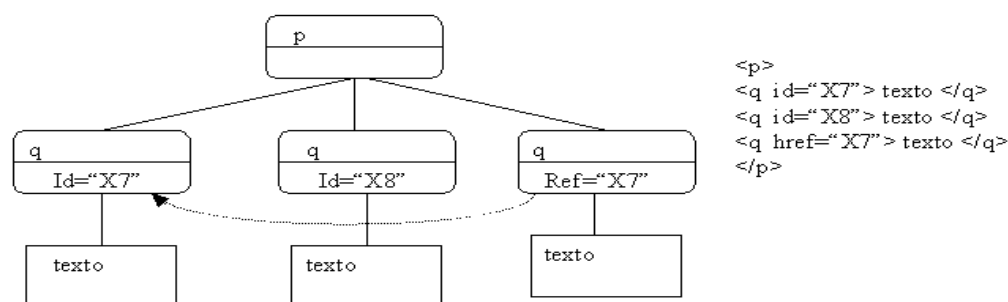


FIGURA 2.4 -Estrutura de um Documento XML

2.2.5 - Coleções de Documentos na Web

Na Web uma coleção de documentos consiste em um agrupamento de objetos e/ou coleções relacionados que podem estar distribuídos por várias localidades e conter vários tipos de mídias distintas. Tecnologias como a Web fazem com que a reprodução e a transmissão de documentos sejam muito facilitadas na Internet. Entretanto, pouca estrutura é provida à gerência de coleções (LAGOZE & DAVIS, 1995). A maioria dessas tecnologias apresenta a mesma abstração: *uma hierarquia de arquivos*. A localização de um arquivo específico dentro dessa estrutura hierárquica nem sempre é facilitada, pois cada sítio provê formas de organização distintas. Alguns agrupam os recursos por ano, por nome de projeto, por assunto, por departamento, etc. Mesmo se todos organizassem seus recursos de forma semelhante, ainda assim não estariam facilitando em muito a localização do arquivo desejado, pois cada sítio emprega convenções muito particulares para a nomeação de seus arquivos e indicação do formato dos dados. Com a forma atual de organização de recursos, se vários arquivos juntos constituem um único documento (várias páginas escaneadas, por exemplo), a relação entre esses arquivos não é identificada de forma explícita, nem pelo formato e nem pelo tipo do arquivo. Apesar de ser útil agrupar arquivos relacionados dentro do mesmo diretório, a falta de padronização desses agrupamentos é um outro problema.

Coleções de documentos são geralmente organizadas segundo *esquemas de classificação*, que provêem um modo uniforme de denotar o assunto principal de um recurso de forma que este possa ser agrupado com itens similares.

Na Web, o serviço YAHOO! (YAHOO) utiliza uma taxonomia própria para agrupar recursos de informação de forma hierárquica. Sua página inicial apresenta uma lista alfabética com diversas áreas de assuntos gerais, que podem ser percorridas pelo usuário até a localização do tópico desejado. Cada entrada no catálogo YAHOO! contém o endereço e um breve texto descritivo sobre o recurso. O resultado de uma pesquisa efetuada sobre o tema “Uso da Web nas escolas”, por exemplo, poderia ser encontrado percorrendo-se os termos da árvore de pesquisa apresentada na Figura 2.5.

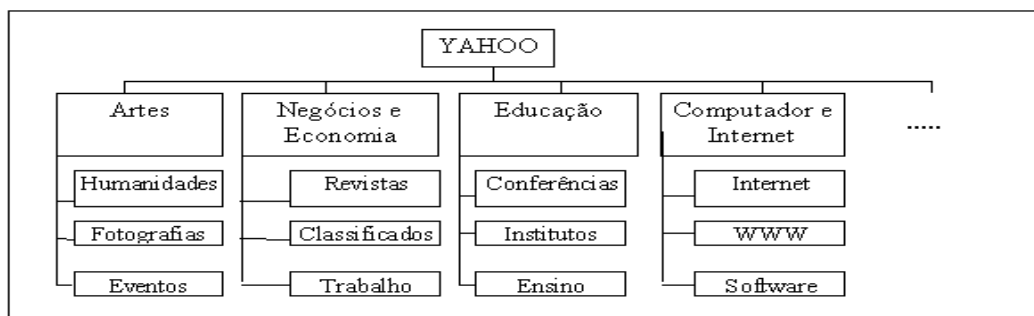


FIGURA 2.5 - Exibição Parcial da Árvore de Pesquisa YAHOO

Diversos outros esquemas de classificação estão sendo propostos para a organização dos recursos da Web (VIZINE-GOETZ, 1996). Alguns são amplamente utilizados no contexto das bibliotecas tradicionais, tais como o *Decimal Dewey* (CHAN et alii, 1996), um esquema hierárquico onde cada conceito é expresso por um número que identifica, de forma concisa, a sua posição na hierarquia.

2.3 – BUSCA E LOCALIZAÇÃO DE DOCUMENTOS NA WEB

2.3.1 – Conceitos Básicos

No ambiente da Web, o usuário faz uso dos vários recursos de informação assistido por diversos tipos de *softwares* denominados clientes que provêm uma interface gráfica que permite aos usuários um controle interativo e uma interação direta com as funcionalidades disponíveis. Embora as necessidades de informação desses usuários possam variar de forma bastante ampla, em todos os casos será localizado um conjunto de recursos de informação relevantes mediante o emprego de diversos serviços e ferramentas de recuperação de informações. Basicamente, os usuários da Web podem acessar seus recursos de informação mediante três tipos de operação de navegação pela teia do hipertexto: surfe, folheio ou pesquisa (KOCH et alii, 1996).

Surfe é a operação de navegação que, em termos práticos, é similar à operação de folhear seguidamente as páginas de um livro a procura de um tópico qualquer, sem antes tentar localizá-lo através de um índice. Na Web, o surfe só é viável dentro de contextos

limitados (dentro de um documento HTML, por exemplo), caso contrário é possível o usuário ficar "*perdido na teia do hipertexto*".

Folheio é a operação de navegação caracterizada pela visualização prévia da estrutura do hipertexto, o que torna possível o percurso de suas ligações de forma sistemática. É uma operação similar à atividade de pesquisa à tabela de conteúdo (ou índice) de um livro a procura de um determinado capítulo que contenha um tópico desejado. Na Web, serviços de folheio podem ser classificados, quanto à organização da informação e a forma de folheio, em cronológicos (CUI World Wide Web Catalog, What's New Too!, etc.), geográficos (W3 Servers, etc.) e por assunto, sendo que o catálogo YAHOO! é o exemplo mais expressivo da categoria.

Pesquisa, por sua vez, é a operação de navegação auxiliada por banco de dados de índices (por palavras-chaves, em sua maioria) criados para este propósito, de forma a possibilitar o acesso direto ao recurso desejado. Em termos práticos, é uma atividade similar à procura de um item em um livro a partir de seu índice remissivo. Na Web, serviços de pesquisa são implementados por *softwares* que coletam, de forma recursiva, informações para indexação nos diversos servidores ligados a Web, para atualização de seus índices. Consistem, basicamente, de: um mecanismo de coleta de informações (ou "robô"), usado para localizar novos recursos de informação ou atualizar informações antigas; um processo de indexação; um sistema de recuperação; e um mecanismo para formulação de pesquisa (ou interface do usuário).

Segundo Koch (KOCH et alii, 1996), serviços de pesquisa da Web podem ser diferenciados pelo tipo de recurso indexado; a audiência de atendimento (se públicos, coletivos, individuais, privados, etc.); o mecanismo de pesquisa, que pode ser único (Alta Vista, Infoseek, etc.) ou combinar vários serviços de pesquisa através de uma interface de pesquisa unificada (MetaCrawler, SavvySearch, etc.); a cobertura do índice, que podem ser genéricos, regionais ou baseados em assunto. Maiores detalhes podem ser obtidos em (MOURA et alii, 1999).

A tendência atual na Web, no entanto, objetiva a integração dos mecanismos de pesquisa e folheio de forma a torná-los mais efetivos. Como exemplos de ferramentas que adotam essa abordagem pode-se citar, dentre outras, Excite, YAHOO! e Lycos.

2.3.2 – Principais Problemas

A grande dificuldade de desenvolvimento de ferramentas de pesquisa de informações para a Web advém da heterogeneidade dos espaços de informação, protocolos associados, recursos de informação, bem como os mecanismos de indexação e acesso. O fator mais relevante, no entanto, é a quantidade cada vez maior de recursos disponíveis para pesquisa na Internet. Isto demanda uma ênfase crescente na precisão da pesquisa e na seleção e ordenação dos recursos pela qualidade ao invés da simples identificação da informação que tem probabilidade de ser relevante para o usuário. O tempo disponível pelo usuário representa, sem dúvida, um fator restritivo na decisão de quais recursos deverá examinar.

Um dos principais problemas das atuais ferramentas de pesquisa da Web está relacionado ao mecanismo de indexação que não consegue distinguir quais conjuntos de páginas em um servidor constituem um documento único, e por esse motivo indexa o conteúdo de páginas individuais e não de recursos de informação. Outro aspecto relevante se refere ao fato de que as palavras contidas nos documentos são indexadas sem considerações sobre a semântica do contexto no qual estão inseridas. É possível, por exemplo, pesquisar documentos que contenham referências a "Tim Berners-Lee", porém não é possível pesquisar quais documentos foram *escritos por* "Tim Berners-Lee". Por esses motivos o conjunto resultado de uma pesquisa não é muito preciso, incorporando um percentual alto de documentos não relevantes, o que obriga o usuário a despendar tempo valioso na procura por documentos de interesse. Também podem ser destacadas a inexistência de informações padronizadas descrevendo cada item do conjunto resultado de uma pesquisa, o que dificulta a sua análise por parte do usuário e a inexistência de controle quanto à qualidade dos recursos indexados (páginas com erros ou inacabadas, de origem duvidosa ou desconhecida, etc.).

De uma forma genérica, as informações armazenadas nos índices criados pelas ferramentas de pesquisa da Web limitam-se àquelas necessárias à descoberta do recurso. Não possuem semântica explícita nem formato estruturado e padronizado, o que impossibilita a pesquisa campo a campo e o intercâmbio de informações entre os diversos índices da Web de forma colaborativa.

Ao contrário das ferramentas de pesquisa, os serviços de folheio se destacam pelo processo de seleção dos recursos onde apenas os itens considerados relevantes por seus

autores são catalogados e também pelos bons sumários de recursos que são produzidos. Dentre as desvantagens desses serviços pode-se citar a manutenção dos catálogos é de alto custo devido ao ambiente extremamente dinâmico da Web o que acarreta em entradas obsoletas no catálogo, ocasionadas pela falta de manutenção automática nos servidores. O principal problema, no entanto, se deve ao fato dessas ferramentas cobrirem apenas uma pequena fração dos recursos da Web.

Uma abordagem mais coerente com o ambiente extremamente dinâmico da Web é a provisão de metadados pelos próprios autores de documentos. A implementação de pesquisas do tipo **autor** = "**Tim Berners-Lee**", típica em ambiente de banco de dados, só seria possível a partir da captação de metadados pelas ferramentas de indexação de recursos. Na seção seguinte esse assunto será explorado mais detalhadamente.

2.3.3 – O Papel do Metadado na Web

Na Web, diversos tipos de metadados podem ser extraído do conteúdo de um documento eletrônico. Esses metadados podem ser obtidos a partir de diversos algoritmos de análise automática de conteúdo (alguns dos quais serão comentados no próximo capítulo) ou serem criados de forma implícita por ocasião da criação do dado bruto contido na mídia (SHETH & KLAS, 1998).

Um editor SGML, por exemplo, gera metadados de acordo com o tipo do documento quando este é editado. De fato, a linguagem SGML não provê uma distinção explícita entre o dado bruto relativo ao conteúdo do documento e o metadado que o descreve. A única distinção existente se refere ao tipo de marcação contida no documento que pode ser estrutural, quando for relativa a uma porção com semântica explícita definida no texto (um parágrafo ou seção, por exemplo), ou não estrutural, quando for relativa aos demais elementos do documento. Em um editor de documentos HTML, por sua vez, além do conteúdo do documento, são geradas várias informações adicionais sobre a página e seu autor segundo as convenções usadas pelos mecanismos de pesquisas.

Tomando como base os metadados que podem ser obtidos a partir de um documento (ou de um recurso qualquer da Web), são diversos os fatores que justificam o seu emprego na Web, alguns dos quais são citados a seguir (NIDR, 1994):

- **desempenho:** registros de metadados são, geralmente, muito menores do que objetos que descrevem, requerendo menos recursos na sua transmissão, pesquisa e armazenamento;
- **arquitetura:** a maioria dos protocolos de recuperação da rede (HTTP, por exemplo) não permitem que partes ou sub-componentes de objetos sejam recuperados separadamente. No entanto, todo o conteúdo do documento pode representar mais informação do que o usuário necessita. Metadados podem ser usados para prover contexto à unidades específicas do documento, por ocasião de uma pesquisa, embora os sistemas de recuperação de informações convencionais não suportem de forma explícita a estrutura lógica de um documento;
- **escopo:** metadados podem descrever recursos que não estão disponíveis no ambiente da rede. Esses recursos podem existir em alguma outra forma que não a digital (impresso, por exemplo) ou em algum dispositivo externo de armazenamento, tal como um CD;
- **conteúdo:** alguns metadados não podem ser extraídos do conteúdo do objeto o qual descrevem. O assunto de um documento, por exemplo, é atribuído mediante uma análise intelectual de seu conteúdo. Da mesma forma, alguns metadados podem requerer métodos de extração complexos, sendo mais prático e menos custoso armazená-los logo após sua obtenção;
- **econômico:** atualmente, a maioria das informações da Web são usadas sem a cobrança de qualquer encargo. No futuro, entretanto, a tendência é que algumas informações só estejam disponíveis aos usuários mediante alguma forma de pagamento. Nesse contexto, os sistemas de recuperação de informações deverão identificar, com base nos metadados disponíveis, quais informações estão livres de encargos e quais podem ser recuperadas mediante alguma forma de pagamento segundo um critério de seleção que considere, por exemplo, a qualidade e o custo da informação;
- **privacidade:** metadados expressam de forma mais adequada os termos e condições que especificam o direito a propriedade intelectual por parte dos autores de recursos; e
- **information retrieval:** recursos de informação nem sempre são facilmente processáveis pois seu formato pode não ser familiar (arquivos compactados, por exemplo), seu conteúdo pode estar criptografado, o recurso pode ser grande, tornando o acesso

custoso para transferência, etc. De uma forma geral, recursos do tipo imagem, vídeo e áudio não são indexados pela maioria dos mecanismos de indexação disponíveis.

2.4 – CONSIDERAÇÕES FINAIS

Os serviços de pesquisa da Internet apresentados neste capítulo surgiram a partir de 1994 e, desde então, têm se expandindo rapidamente. De uma forma genérica, esses mecanismos diferem principalmente quanto a seleção dos documentos indexados e o tamanho (ou cobertura) e frequência de atualização dos bancos de índices criados (KOCH et alii, 1996). Nesse contexto, a efetividade de uma consulta é altamente dependente de um conhecimento prévio do usuário a respeito do mecanismo de pesquisa utilizado.

Um problema que vem se destacando atualmente se refere ao fato da arquitetura atual dos sistemas de coleta, indexação, pesquisa e recuperação não estarem bem ajustados ao número de documentos e a taxa de crescimento na Web: o enfoque principal não é *qualidade* (ou precisão) dos itens selecionados por ocasião de uma pesquisa e sim a *quantidade* dos itens recuperados. Nesse contexto, conforme destacado na Seção 2.3.3, o emprego de metadados tem sido considerado essencial em diversas funcionalidades da Web, dentre elas a localização efetiva de documentos.

Tendo em vista os diversos tipos de documentos estudados neste capítulo, três tipos distintos de metadados podem ser identificados na Web (SHETH & KLAS, 1998):

- aqueles que são contidos explicitamente no documento e identificados por rótulos específicos, tais como *autor*, *título*, etc., que são capturados por alguns poucos serviços de pesquisa;
- aqueles obtidos a partir de uma análise efetuada sobre o conteúdo do documento, tais como: o tamanho do documento, seu formato, uma lista de palavras-chaves, etc. Esta abordagem é adotada pela maioria dos serviços de pesquisa; e
- mais recentemente, aqueles metadados que são providos de forma manual, cujo objetivo é tornar mais preciso o resultado de uma pesquisa.

Neste último caso, o uso adequado de metadados pelas diversas ferramentas de pesquisa ainda depende do desenvolvimento ou da adaptação de padrões para descrição de recursos da Web e para o intercâmbio de dados estruturados nesse ambiente. No próximo

capítulo será apresentado um enfoque mais detalhado do emprego de metadados para tornar mais efetiva a busca e localização de documentos na Web.

CAPÍTULO 3

O EMPREGO DE METADADOS PARA A DESCRIÇÃO DE DOCUMENTOS ELETRÔNICOS NA WEB

3.1 - INTRODUÇÃO

Grande parte das discussões relacionadas à descoberta e recuperação de informações na Web envolve considerações a respeito do emprego de metadados. Literalmente, metadado significa “dado sobre o dado”, embora a sua origem seja obscura, remontando ao ano de 1976 (NIDR, 1994). Como o termo vem sendo amplamente empregado em diversas áreas de estudo, qualquer trabalho relacionado a metadados deve definir, de forma explícita, o seu contexto de uso, de forma a propiciar a sua correta interpretação.

O enfoque básico deste capítulo é o estudo do metadado como requisito essencial para promover a localização efetiva dos recursos de informação, em particular os documentos eletrônicos, em um ambiente expansível, heterogêneo e distribuído como a Web. Essa necessidade decorre do aumento exponencial de recursos na Web e da demanda por soluções mais efetivas do que aquelas implementadas pelas ferramentas de pesquisa que empregam a indexação automática do conteúdo dos recursos analisados.

Com o objetivo de se identificar os tipos de metadados que podem ser requeridos para a gerência de documentos eletrônicos na Web, serão descritas, inicialmente, na Seção 3.2, duas propostas de classificação de metadados. A proposta geral foi definida com base nos tipos de metadados que são requeridos para acesso a tipos distintos de mídias digitais. A proposta funcional enfoca os metadados necessários para se descrever um documento eletrônico. Ambas classificações serão utilizadas como base para a classificação funcional de metadados apresentada na Seção 4.6 que, por sua vez, será empregada como subsídio para a comparação dos padrões de metadados analisados no Capítulo 4.

3.2 – CLASSIFICAÇÕES DE METADADOS

Diversos trabalhos relacionados a metadados fazem uso de classificações com o objetivo de identificar a natureza de seus elementos e o papel desempenhado por cada elemento dentro do contexto de informação que está sendo considerado. Neste capítulo, a

classificação geral de metadados apresentada tem por objetivo enfatizar: o modo como os metadados são obtidos, se de forma automática ou manual; se são baseados ou não no conteúdo do recurso; se são relacionados ou não ao tipo de mídia; e se são dependentes ou não de um domínio qualquer de informação. Quanto à relevância da classificação funcional apresentada na Subseção 3.2.2, esta se deve ao fato do seu enfoque estar voltado aos metadados necessários para a descrição de um documento eletrônico composto por itens multimídia.

3.2.1 - Classificação Geral

A classificação de metadados descrita a seguir foi proposta por KASHYAP (KASHYAP et alii, 1995). Um aspecto que se destaca nesta proposta está relacionado aos metadados descritivos de conteúdo: a classificação subentende que esses metadados também podem ser classificados como dependentes de conteúdo no caso de serem extraídos por processos de análise automática sobre o conteúdo do recurso.

- **Metadados dependentes de conteúdo:** são aqueles termos extraídos por processos automáticos, inteiramente guiados pelo tipo da mídia, que tomam como base o conteúdo do dado original. Um extrator automático de metadados para arquivos de e-mail, por exemplo, pode filtrar informações tais como: quem enviou a mensagem, a data, o assunto, para quem foi enviada a mensagem, etc.; e
- **Metadados descritivos de conteúdo:** são termos descritivos que são associados ao conteúdo do recurso. Podem ser específicos da mídia em questão como, por exemplo, os termos que descrevem a cor predominante, o formato e a textura de uma imagem, ou independentes da mídia tais como os termos que classificam o conteúdo (ou assunto) de um documento. Também podem ser classificados em:
 - ◊ **dependentes de domínio:** são termos que empregam conceitos relacionados a algum domínio específico como, por exemplo, aqueles que descrevem a interpretação de uma imagem de satélite (vegetação, uso territorial, etc.);
 - ◊ **independente de domínio:** são termos descritivos genéricos que não são baseados em conceitos específicos extraídos do domínio de informação no qual o recurso analisado esteja inserido. Podem ser enquadrados nesta categoria os metadados que descrevem a estrutura de um documento multimídia, por exemplo; e

- ◇ **independentes de conteúdo:** são aqueles termos descritivos que não podem ser extraídos diretamente do conteúdo do recurso ao qual se referem. Podem ser citados como exemplos a data/hora da última modificação e o local de armazenamento de um documento.

3.2.2 - Classificação Funcional

A classificação funcional de metadados apresentada a seguir foi proposta por BOHM (BOHM & RAKOW, 1994) para utilização no projeto HyperStorM (“*Hypermedia Document Storage and Modeling*”) que provê uma infra-estrutura genérica para a gerência (com base em metadados) de diversos tipos de documentos estruturados. É composta por seis categorias que são descritas a seguir:

- **Metadados para a representação de tipos de mídia:** provêem informação relacionadas à apresentação do dado multimídia, ao idioma, ao formato e a técnicas de compressão;
- **Metadados descritivos de conteúdo:** são os metadados usados para descrever o conteúdo de um documento multimídia. Podem ser dependentes da mídia;
- **Metadados para classificação de conteúdo:** correspondem a informações adicionais que podem ser derivadas do conteúdo do documento auxiliando em algum tipo de classificação que possa ser utilizada em sistemas de filtragem. No contexto médico, por exemplo, estas informações podem ser úteis para a determinação das pessoas autorizadas a lerem um documento específico;
- **Metadados para a composição do documento:** são metadados que descrevem os relacionamentos entre os componentes do documento e o papel de cada componente dentro de um determinado contexto. Possibilitam a implementação de métodos envolvendo sua composição, tal como *ObtemPróximoComponente*;
- **Metadados para histórico do documento:** provêem informações históricas e de acompanhamento relativas à natureza evolucionária e dinâmica de um documento, observando seus componentes individuais. São exemplos de metadados desta categoria: “aprovadorPor”, “nãoAprovado” e “dataDaÚltimaAtualização”; e
- **Metadados para a localização do documento:** são aqueles que possibilitam o acesso a um determinado documento.

3.3 – EMPREGO DE METADADOS NA WEB

Desde o seu surgimento, a forma mais comum de emprego de metadados na Web ocorre por ocasião da transferência de um recurso de informação do cliente para o servidor HTTP, ou vice-versa. Nessa ocasião, metadados são *anexados* ao recurso transferido através dos cabeçalhos do protocolo HTTP, que são emitidos pelo servidor em resposta à uma solicitação recebida de um cliente (um navegador Web, por exemplo). Existem vários tipos de cabeçalhos HTTP, dentre eles: *expires*, que informa a data em que um documento se torna obsoleto; *refresh*, que especifica um período (“delay”) em segundos antes que o navegador recarregue o documento automaticamente; e *content-language*, que especifica o idioma relativo ao conteúdo do documento, etc. (VANCOUVER WEBPAGES, 1997).

Com o advento da linguagem HTML, no entanto, tornou-se também possível *embutir* metadados nos documentos HTML, dentro dos rótulos META, para que estes fossem coletados pelos mecanismos de indexação que atuam na Web. A declaração META pode funcionar como um recipiente para metadados, armazenando as propriedades do documento na forma de pares (nome-atributo, valor-atributo). Possui dois atributos: “*name*”, que especifica a propriedade do documento e “*content*”, que especifica o valor dessa propriedade. Sua sintaxe básica é: <META HTTP-EQUIV=“*name*” CONTENT = “*content*”> ou <META NAME = “*name*” CONTENT = “*content*”>. O atributo HTTP-EQUIV torna uma declaração META equivalente a um cabeçalho HTTP, sendo este responsável por controlar a ação do navegador. No exemplo a seguir, o cabeçalho HTTP “Expires: Mon, 13 May 1998 09:13:25 GMT” é equivalente à declaração: <META HTTP-EQUIV=“*expires*” CONTENT=“Mon, 14 May 1998 09:13:25 GMT”>.

Mais recentemente, com o desenvolvimento da plataforma PICS (*Platform for Internet Content Selection*) (PICS, 1996) (PICS, 1998), um mecanismo para a comunicação, do servidor para o cliente, de rótulos de avaliação (*rating labels*) sobre o conteúdo de páginas HTML, deu-se início, na Web, ao emprego de metadados que são armazenados *independentemente* do recurso ao qual se referem. Neste contexto, uma definição mais apropriada para metadado seria “o *dado estruturado sobre o dado que é interpretado de forma automática*” (W3C). A estrutura do metadado pode ser usada, na

Web, para tornar mais efetivos diversos processos, tais como a *identificação, descoberta e recuperação de recursos de informação*, descritos mais detalhadamente a seguir.

3.3.1 – Processo de Identificação de Recursos

Segundo Berners-lee (BERNERS-LEE, 1994), um recurso de informação representa qualquer objeto passível de receber um identificador único. Atualmente, a base para identificação de um recurso na Web é o URL (*“Uniform Resource Locator”*), um metadado que expressa a sua localização em um determinado espaço de informação da Internet.

Apesar de existirem variações, o formato básico de um URL corresponde a: *esquema://host:porta/caminho*. *Esquema* identifica o protocolo de recuperação que será utilizado (HTTP, FTP, Gopher, Telnet, etc.); *host* é o computador hospedeiro no qual o recurso reside; *porta* é um número que identifica o serviço que está sendo requisitado ao computador hospedeiro; e *caminho* é a identificação da localização de um recurso no espaço de armazenamento do computador hospedeiro. São exemplos de URLs válidos:

- ftp://nic.merit.edu/documents/fyi/fyi_20.txt
- <gopher://gopher.cic.net:2000/11/hunt>
- <telnet://locis.loc.gov>
- <news://comp.infosystems.www.users>
- <http://www.rpi.edu/~decemj/works/wwwu/contents.html>

Como consequência da abordagem de nomeação baseada no endereço ou localização do recurso na Web, pode-se citar: ocorrência de referências pendentes (recursos não encontrados) ocasionadas por manutenções efetuadas nos servidores, movimentando de um local para outro ou eliminando recursos; inexistência de mecanismos de balanceamento automático de carga da rede, redirecionando as solicitações de um servidor sobrecarregado para outro que possua cópia dos recursos; baixa tolerância do sistema a falhas devido à impossibilidade de se redirecionar solicitações para outros servidores que estejam ativos; e a lentidão na recuperação de recursos devido a inexistência de um sistema de memórias *caches* para armazenar os recursos mais solicitados, por exemplo.

Com o objetivo de solucionar esses problemas, um grupo IETF ("*Internet Engineering Task Force*") vem desenvolvendo uma arquitetura geral para identificação única de recurso na rede. Faz parte dessa arquitetura um mecanismo de nomeação de recursos denominado URN ("*Uniform Resource Name*") (URN, 1996) que é persistente, imutável, globalmente único e independente da localização do recurso. Para a utilização desse mecanismo, um serviço da rede será responsável por prover o seu mapeamento com o recurso correspondente. Este processo, denominado *resolução*, envolverá os seguintes componentes:

- **esquema de nomeação:** define um procedimento padrão para criação e assinalamento de URNs únicos definidos segundo uma sintaxe especificada;
- **autoridade de nomeação:** é a organização responsável por atribuir URNs segundo um determinado esquema de nomeação;
- **sistema de resolução:** é o serviço responsável por armazenar e resolver URNs. Um sistema de resolução deverá retornar a localidade atual de um recurso ou uma lista de localidades desse recurso; e
- **registro de URN:** é o serviço responsável por armazenar dados acerca de esquemas, autoridades de nomeação e sistemas de resolução. Um registro provê dois tipos de funcionalidade: fornecer regras para a extração da autoridade de nomeação de um URN em um esquema de nomeação em particular; e determinar qual sistema de resolução é capaz de resolver um dado URN.

Por definição, um esquema de nomeação não é amarrado a um sistema de resolução específico. São conceitos de implementação independentes. Dessa forma, um sistema de resolução deverá ser potencialmente capaz de resolver qualquer URN.

A sintaxe proposta para um URN identifica o esquema de nomeação, seguido por um conjunto de caracteres relativos a uma sintaxe específica definida por uma autoridade qualquer, conforme exemplificado a seguir: *urn:hdl:cnri.dlib/agosto95*, onde: "hdl" identifica o esquema de nomeação como sendo o sistema *handle*; "cnri.dlib" identifica a autoridade de nomeação e "agosto95" é o conjunto de caracteres único que identifica o recurso, cuja sintaxe é definida pelo esquema de nomeação específico.

3.3.2 – Processo de Descoberta de Recursos

Descoberta é um termo amplo usado para abranger todo o processo de identificação dos recursos de informação da rede que possam estar disponíveis para o usuário (NIDR, 1994). Não é um processo simples e linear como uma consulta a um diretório de recursos de informação da rede. Ao contrário, é um processo complexo, interativo e altamente dinâmico, centrado primordialmente no usuário e modelado pela aquisição incremental de informação e conhecimento, agregando também experiências e preferências pessoais. O processo de descoberta de recursos envolve um grande número de atividades táticas que são executadas de forma interativa. De uma forma genérica, pode-se considerar o processo de descoberta de recursos como uma série de movimentos entre duas fases:

- **fase de localização:** é a fase durante a qual uma pessoa formula um conjunto de critérios de seleção e, a partir desses critérios, obtém um conjunto de recursos candidatos. Um aspecto importante a ser considerado nesta fase é a granularidade do conjunto resultado que depende diretamente do nível de especificidade dos termos usados pelo pesquisador no vocabulário de consulta. Quanto menos específicos forem os termos usados nos critérios de seleção, mais resultados “falsos” podem ser incorporados ao resultado da pesquisa; e
- **fase de exame:** é a fase na qual o usuário verifica o conjunto resultado da pesquisa para seleção dos recursos candidatos a recuperação. Esta seleção pode ser feita com base em registros de catalogação formal (segundo um determinado padrão de descrição) ou em uma forma de descrição mais trivial como os resumos exibidos pelas ferramentas de pesquisa. A partir do exame do conjunto resposta é possível passar a fase de recuperação dos recursos desejados caso o conjunto resposta seja satisfatório ou, então, retornar à fase de localização submetendo-se critérios de seleção modificados. Apesar desse não ser um processo essencialmente linear, seu princípio básico consiste do refinamento sucessivo do conjunto resultado até que este seja satisfatório.

No contexto da descoberta, metadados podem ser empregados para auxiliar os usuários a encontrar informações *relevantes* a respeito de um tópico. Por informação relevante pode-se considerar: a mais atual; a que possua o nível adequado de detalhe, aquela cuja qualidade seja controlada, ou aquela que reflita uma perspectiva desejada pelo usuário; encontrar *uma* informação já conhecida; encontrar *todas* as informações disponíveis a respeito de um tópico; encontrar *novas* informações a respeito de um tópico; identificar recursos similares, distintos, equivalentes, relacionados, etc.; e analisar a

conveniência de uso do recurso, que implica na disponibilidade prévia do software e do hardware necessário ao seu uso.

Metadados também podem ser usados para prover visões abstratas de um conjunto resultado muito grande. Essas visões podem corresponder a sumários de documentos, versões reduzidas de imagens (“*thumbnails*”), etc.

3.3.3 – Processo de Recuperação de Recursos

Recuperação é um processo complementar à descoberta e envolve o acesso aos recursos de informação constantes do conjunto resultado obtido no processo de descoberta. Na Internet, os sistemas de recuperação foram desenvolvidos muito antes dos sistemas de descoberta de recursos na forma de seus inúmeros protocolos de comunicação (Gopher, FTP anônimo, Telnet, etc.) (NIDR, 1994).

Historicamente, protocolos de recuperação obtém visões muito simples dos objetos recuperados. O protocolo FTP, por exemplo, compreende basicamente objetos binários e textuais (ASCII). Protocolos mais recentes como o HTTP obtém um conhecimento maior a respeito do tipo do objeto recuperado, porém esse conhecimento não é muito utilizado como auxílio ao processo de recuperação.

De uma forma genérica, transferir um recurso constitui-se, basicamente, de duas etapas: avaliação do endereço (URL) do recurso e movimentação de um conjunto de bits localizado em um endereço remoto para um cliente local. Um designador do tipo do recurso é movido como parte da operação de transferência de forma que o receptor possa determinar o tipo de software que deverá ser invocado para processar o conjunto de bits recebido: as funções de visualização e “*download*” são dependentes do conteúdo e estrutura do recurso.

Para que o mecanismo de recuperação funcione adequadamente na Web, recursos de informação devem ser armazenados em múltiplas versões. Isto se dá pelo fato da maioria dos protocolos de recuperação não incluírem funcionalidades para a negociação do formato do recurso solicitado pelo cliente ao servidor, aliado ao fato de que os servidores funcionam geralmente como simples fornecedores de conjuntos de bits. Dessa forma, é comum encontrar documentos armazenados em vários formatos em um mesmo servidor, sem nenhuma indicação de qual versão é a original ou a mais atual. O ideal seria a

existência de uma única cópia por objeto que pudesse ser liberada em vários formatos via conversão efetuada no ambiente servidor (PRICE-WILKIN, 1997). Uma imagem, por exemplo, poderia ser liberada em três versões: uma reduzida (“*thumbnail*”) que seria usada para propósitos de folheio; uma versão para visualização na tela; e outra versão de alta resolução para impressão ou exame mais detalhado na tela.

No contexto da recuperação, metadados podem ser usados para informar qual é a versão original de um recurso e o quanto de degradação poderá ocorrer devido a uma conversão de formato no servidor. Um protocolo de recuperação melhor adaptado às funções de recuperação de recursos deveria possuir habilidade para negociar os custos de acesso a um objeto sem que uma transação de recuperação tivesse que ser inicializada. Metadados para avaliação do custo de recuperação de um recurso podem descrever: o custo básico atribuído ao objeto; quem está solicitando acesso ao objeto (alguns usuários podem acessá-lo livremente, enquanto que outros necessitam de uma autorização ou uma confirmação de pagamento efetuado, por exemplo); o formato em que está sendo solicitado o objeto (uma imagem na versão reduzida pode ser liberada livremente enquanto que na versão de alta resolução pode ter um custo alto de liberação); o período em que o objeto está sendo acessado (horários de pico podem representar um custo maior de acesso); o tempo de acesso ao recurso, no caso deste ser interativo; etc.

3.4 – OBTENÇÃO DE METADADOS NA WEB

Existem, basicamente, duas abordagens para a captação de metadados na Web: *automática* e *manual* (IANELLA, 1997). A abordagem automática baseia-se na extração de metadados através de ferramentas desenvolvidas para esse fim específico. A abordagem manual é baseada na obtenção de metadados através formulários especificados por padrões para a descrição de conteúdo. A opção por determinada abordagem irá depender de fatores tais como: valor do acervo a ser resguardado; necessidade de informações selecionadas segundo critérios de qualidade; quantidade de recursos a serem descritos, etc. É uma função custo-benefício: metadados obtidos de forma manual são mais difíceis de manter enquanto que metadados capturados de forma automática produzem índices de menor qualidade e, conseqüentemente, informações menos precisas. A seguir, essas abordagens serão descritas de forma mais detalhada.

3.4.1 - Abordagem Automática

Documentos eletrônicos no formato HTML representam a maioria dos objetos acessados na Web. Geralmente, esses documentos não possuem nenhuma informação adicional associada para tornar mais precisa a sua recuperação, ficando na dependência exclusiva dos mecanismos de indexação disponíveis. Alguns métodos de análise de conteúdo mais utilizados para extração automática de metadados de recursos textuais na Web são descritos a seguir:

- **Palavras-chaves** - é a abordagem mais comum de análise de texto na Web. Palavras-chaves são termos que expressam algum significado semântico do texto do qual são derivadas. Algumas ferramentas obtêm palavras-chaves de parte do conteúdo do documento, selecionando títulos, cabeçalhos, sub-cabeçalhos, textos âncoras, palavras de maior ocorrência no texto, linhas iniciais do texto, etc. *Magellan Internet Guide*, *WWW Worm* e *Inktomi* são exemplos de mecanismos de indexam partes do documento. Outras ferramentas utilizam o método de *indexação completa de texto* (“full-text”). Neste caso, cada palavra do conteúdo analisado aparece no índice criado. Para preparar um documento para indexação, um analisador léxico quebra o texto em um conjunto de palavras. Estas palavras podem ser confrontadas com uma lista que evita a indexação de palavras comuns tais como artigos e preposições. Um das desvantagens desse método é o índice gerado, comparável, em tamanho, aos recursos que indexa. Alguns algoritmos foram criados para diminuir o número de termos indexados. A maioria associa a cada palavra um peso que verifica a sua relevância dentro do texto, medida, na maioria da vezes, pela relação entre a frequência do termo no documento e sua frequência no conjunto de todos os documentos recuperados (KOCH et alii, 1996).
- **Indexação invertida** – o emprego desta abordagem provê um índice contendo todos os termos de referência (ou palavras-chaves) que ocorrem em todos os documentos de uma coleção (SALTON, 1983). Cada termo é armazenado com uma lista que referencia cada documento individual que contenha uma ou mais ocorrências desse termo. Uma das desvantagens do método é a grande quantidade de processamento exigida para

manter o índice atualizado, compartilhando da mesma desvantagem do método anterior quanto ao espaço exigido para armazená-lo.

- **Indexação semântica** – esta abordagem baseia-se em um conhecimento prévio da estrutura do arquivo que será indexado a fim de conduzir um processo de extração de metadados que seja específico para cada tipo. A partir da identificação do tipo do arquivo, um processo de análise específico, denominado *sumarização* (HARDY & SCHWARTZ, 1993), é selecionado e aplicado ao recurso. Esse processo toma como base um catálogo que armazena as principais características dos tipos distintos de arquivos. Uma das vantagens deste método é a modularização introduzida no processo de extração de metadados, tornando-o mais preciso e facilmente adaptável a novos formatos de dados. O sistema HARVEST, desenvolvido na Universidade do Colorado, EUA, possui um sub-sistema, denominado *Essence*, que utiliza esse método com o objetivo de extrair informações para indexação de vários tipos de dados, inclusive compactados.
- **Indexação semântica latente** - esta proposta observa que um conjunto de termos extraídos de forma aleatória do conteúdo de um documento fornece uma representação incompleta e não confiável desse documento (DEERWETER, 1990). Segundo a abordagem de indexação semântica latente, os outros mecanismos de indexação ignoram dois conceitos importantes: *sinonímia*, que significa mais de uma palavra relacionada ao mesmo conceito (colégio e escola, por exemplo); e *homonímia*, que significa uma palavra relacionada a mais de um conceito. A proposta de indexação semântica latente argumenta que uma análise adicional é necessária para representar a semântica de um texto de forma mais acurada. Representa um meio termo entre a catalogação manual e a abordagem de indexação automática, introduzindo a *indexação por conceito*. Nessa abordagem, o resultado de uma pesquisa é representado por uma matriz onde cada linha representa um conceito (ao invés de um documento). O mecanismo de pesquisa *Excite* (EXCITE), baseado nessa abordagem, utiliza um esquema de classificação baseado na análise estatística dos documentos correntes, com o objetivo de agrupar documentos com perfis similares. Seu algoritmo “aprende” a categorizar um documento de baixo para cima, partindo de um conjunto de documentos

para chegar a uma categoria de assunto. É um sistema que se organiza automaticamente, ou seja, a categoria de assunto é determinada por uma assunção básica: palavras que ocorrem freqüentemente juntas estão, de alguma forma, relacionadas. O esquema de classificação se ajusta automaticamente às mudanças refletidas nos textos que venham a sugerir novas conexões entre palavras. Esta abordagem elimina duas grandes críticas aos sistemas tradicionais de classificação bibliotecária: que todo esquema de classificação tem um ponto de vista (de alguém ou de alguma organização) e que todo esquema luta constantemente contra a obsolescência.

- **Classificação semântica** – ao contrário das abordagens onde o conteúdo extraído do documento é indexado sem qualquer informação contextual associada, nesta abordagem uma classificação do conteúdo (ou assunto) do documento é atribuída, de forma automática, com base em categorias previamente definidas que expressam o seu significado semântico. Um exemplo de utilização dessa abordagem é o projeto SCORPION (SHAFER, 1997) que enfoca o reconhecimento automático do assunto de um documento através da combinação de técnicas de recuperação da informação com a ciência da catalogação. O algoritmo utilizado é baseado no esquema de Classificação *Decimal Dewey* (THOMPSON et alii, 1997).
- **Indexação de meta-informações** – esta abordagem é a mais recente, empregada por algumas ferramentas de indexação automática de recursos, tais como *AltaVista*, *Infoseek*, *Metadata Search Engine* e *Harvest* que além do conteúdo do documento, capturam também as informações contidas nos rótulos META na parte de cabeçalho da linguagem HTML. Essas informações são fornecidas pelo próprio autor do documento, com a finalidade de auxiliar na sua localização (KOCH et alii, 1996). A utilização dessa facilidade tem importância vital quando o documento utiliza “frames” (quadros) ou quando é composto por muitos elementos gráficos (VANCOUVER WEBPAGES, 1997). Nesse último caso, existe pouco ou nenhum texto a ser indexado além daquele contido no atributo ALT da declaração IMG, que conecta uma imagem a uma página HTML.

3.4.2 - Abordagem Manual

A obtenção manual de metadados pode envolver a atividade de catalogação de recursos. Segundo Levy (LEVY, 1996), catalogar é a prática de organizar uma coleção de itens bibliográficos para facilitar sua identificação, localização, acesso e uso. A catalogação cria uma ilusão de ordem (“a ordem dos livros”) em um universo de informação caótico.

Quando a quantidade de itens é suficientemente grande, ocorre a necessidade de um catálogo, que consiste de um conjunto de registros de catalogação, cada qual relacionado a um determinado item da coleção. Um catálogo também pode ser considerado uma coleção cujos objetos são os registros de catalogação que descrevem certas características de um item, tais como o seu autor, assunto, etc.

Aparentemente trivial, catalogar não é simplesmente ler as propriedades evidentes de um item, mas uma atividade que exige habilidade para interpretar as propriedades pouco evidentes desse item. Existem dois tipos de atividades de catalogação (LEVY, 1996):

- **catalogação descritiva:** se refere à atividade de criação de registros de catalogação para um item, descrevendo suas características observáveis (título, autor, etc.); e
- **catalogação de assunto:** se refere à atividade de classificação do conteúdo intelectual de um item. Geralmente, esta classificação irá determinar sua localização em um sistema de armazenamento qualquer.

De uma forma genérica, o processo de catalogação envolve um conjunto de instituições inter-relacionadas responsáveis pela produção, distribuição, conservação e uso do material bibliográfico; um conjunto de termos para a classificação do material, dando origem às noções de autor, edição, trabalho intelectual, etc.; mecanismos de regulamentação (*copyright*, etc.) para a gerência e uso do material controlado. É uma atividade de normalização e regularização que utiliza categorias padrões de descrição a fim de obter um produto final que corresponde a uma idealização do item descrito.

Na Web, a atividade de catalogação de recursos é bem recente, tendo iniciado a partir da constatação de que o acesso efetivo aos recursos de informação disponíveis só seria possível com a construção de índices que pudessem conduzir os usuários à localização de informações de qualidade. A principal característica dessas atividades se refere ao o enfoque (ou granularidade) da descrição que se restringe à produção de um resumo do conteúdo do servidor, não provendo uma documentação detalhada de cada item disponível. Posteriormente, esses resumos são disponibilizados para as ferramentas

automáticas de recursos que indexam informações captadas de forma manual. Nesse sentido, dois projetos de catalogação se destacam na Web:

- *ROADS*: “*Resource Organisation and Discovery in Subject-based Services*” (HEERY, 1996b) é um dos projetos do Programa de Biblioteca Eletrônica (eLib) do Reino Unido que objetiva a modernização dos serviços de biblioteca oferecidos ao nível universitário. *ROADS* investiga formas para criação, armazenamento e distribuição de metadados sobre os recursos de informação da Web. O projeto usa uma variação dos formulários do padrão IAFA (DEUTSCH et alii, 1995), descrito no próximo capítulo, para captação manual de metadados. As facilidades do projeto *ROADS* incluem: um mecanismo de pesquisa; várias ferramentas para a criação e processamento dos registros *ROADS*; uma interface para a Web e um mecanismo que seleciona quais servidores devem ser pesquisados; e
- *DESIRE*: é outro projeto do programa eLib que também usa uma variação dos formulários IAFA. *DESIRE* (DESIRE) objetiva o desenvolvimento de índices automáticos da Web (baseados em robôs) e a construção de catálogos organizados por assunto (também denominados de “*gateways*”) cujos recursos são catalogados manualmente. Um *gateway* enfoca primordialmente a qualidade de seus recursos. Para tal, define uma série de critérios de seleção. Dos *gateways* disponíveis, pode-se citar: *SOSIG* (“*Social Science Information Gateway*”); *OMNI* (“*Organising Medical Networked Information*”); *EEVL* (“*Edinburg Engineering Virtual Library*”); *ADAM* (“*Art, Design, Architecture and Media Information Gateway*”) e *IHR-info* (“*Institute of Historical Research*”).

Um problema de ambas iniciativas é que o fornecimento de metadados fica restrito exclusivamente aos provedores de informações e gerentes de sítios da Internet, não possibilitando aos criadores de documentos prover metadados a respeito de seus próprios recursos de informação. A partir de 1995, no entanto, com o propósito de promover a integração de esforços para alcançar uma solução comum para o problema de descrição de recursos na Internet, organizações tais como a OCLC (“*Online Computer Library Center*”) e o NCSA (“*National Center for Supercomputing Applications*”) organizaram o primeiro “*workshop*” em metadados, do qual participaram profissionais da área da computação, biblioteconomia, codificação de textos e outras áreas afins. O objetivo do encontro foi identificar e definir um conjunto de elementos simples e genérico para a descrição de uma

faixa ampla recursos da Web, menos complexo do que os padrões tradicionais de catalogação bibliotecária, de forma a encorajar autores e publicadores a prover metadados que possam ser captados pelas ferramentas de coleta automática. O modelo Dublin Core, descrito no capítulo seguinte, representa um consenso de diversas comunidades nessa direção (WEIBEL et alii, 1995).

3.5 – CONSIDERAÇÕES FINAIS

A medida que proliferam os recursos de informação na Web, torna-se cada vez mais premente a adoção de meios sistemáticos para a identificação, localização e descrição desses recursos. De uma forma geral, apesar de algumas sofisticacões implementadas pelas ferramentas de pesquisa disponíveis, conforme descrito neste capítulo, estas ainda retornam resultados bastante imprevisíveis por ocasião de uma consulta. O principal problema é a perda do contexto da informação por ocasião do processo de indexação.

Bibliotecários vêm empregando metadados como método para a descrição de recursos por décadas. O que é novo sobre metadados, no entanto, é o cenário no qual estes serão usados: recursos de informação da Web, ao contrário do acervo das bibliotecas tradicionais, são objetos em constante movimento.

Inicialmente, a comunidade bibliotecária necessitará estender as práticas da catalogação tradicional aos recursos acessíveis através da Web, para o emprego adequado do metadado. É especialmente relevante o emprego dos métodos utilizados por essa comunidade, uma vez que esses tem-se mostrado bastante adequados à tarefa de organização da informação.

A complexidade do problema a ser resolvido, entretanto, sugere uma variedade de soluções, conforme poderá ser observado no próximo capítulo, onde serão descritos alguns dos padrões de metadados que estão sendo propostos para emprego na Web.

CAPÍTULO 4

PADRÕES DE METADADOS PARA A DESCRIÇÃO DE CONTEÚDO

4.1 – INTRODUÇÃO

Padrões representam um meio importante para se obter esquemas de representação comuns e interoperabilidade entre sistemas (SHETH & KLAS, 1998).

Um padrão de metadados provê um conjunto de elementos descritores organizados segundo um modelo de dados que é empregado para descrever objetos do domínio de informação do usuário. Um modelo de dados, por sua vez, pode ser definido como um conjunto de conceitos e regras de composição associadas que são usadas para descrever tipos e relacionamentos entre as instâncias desses tipos. No contexto relacional, o modelo de dados básico consiste de tabelas, registros e atributos. No contexto da orientação a objetos, o modelo de dados básico consiste de classes e objetos. Em ambos os modelos, regras de composição distintas se aplicam. No modelo orientado a objetos, essas regras se referem à herança entre classes, polimorfismo, domínio de atributos e agregação. No modelo relacional, essas regras especificam, por exemplo, que um registro deve conter um número qualquer de atributos e que cada atributo deve ser definido em termos de um tipo primitivo de dado.

De uma forma genérica, a complexidade do modelo de dados depende diretamente da complexidade estrutural da informação que se quer representar. Um modelo que consiste em conjunto de pares (nome-atributo, valor-atributo) não possibilita a representação dos tipos de relacionamentos encontrados em um sistema de arquivos, por exemplo. Neste caso, modelos de dados mais complexos (baseados em grafos, árvores, etc.) seriam mais apropriados.

Neste capítulo são analisados alguns dos padrões de metadados que estão sendo desenvolvidos ou adaptados para a descrição dos recursos da Web, em particular, os documentos eletrônicos. Foram agrupados segundo as suas características mais evidentes por Heery (HEERY, 1996a) em padrões para a documentação de acervos específicos, descritos na Seção 4.2, e padrões para a descoberta de recursos na Web, descritos na Seção 4.3. Na Seção 4.6 é apresentada uma classificação funcional de metadados que será empregada como base para a comparação desses padrões tendo em vista a possibilidade de emprego na Web.

O termo “padrão” será empregado de forma indistinta neste capítulo, devido ao fato das iniciativas de metadados descritas se originarem de um consenso formado por uma determinada comunidade, não correspondendo a iniciativas isoladas.

4.2 – PADRÕES DE METADADOS PARA DOCUMENTAÇÃO DE ACERVOS ESPECÍFICOS

Os padrões desta categoria provêm informações bibliográficas completas a respeito de um item e possibilitam a descrição de relacionamentos e hierarquias complexas entre objetos. Alguns são bem antigos, tendo sido projetados, de uma forma geral, para atender às necessidades específicas de diferentes comunidades (museus e bibliotecas, por exemplo) que, em comum, possuem um acervo valioso e de caráter permanente que deve ser resguardado.

Quanto ao emprego na Web, a tendência atual reforça o uso desses padrões por usuários especializados, mas prevê a liberação desses metadados de forma resumida para as ferramentas de descoberta de recursos, já que o acervo disponível é muito grande. Essa liberação seria possível mediante o estabelecimento de regras de conversão do conjunto mais amplo de metadados para um mais simples e genérico (tal como o padrão Dublin Core, descrito adiante).

Os padrões MARC, TEI, EAD e GILS, enquadrados nesta categoria, são descritos em ordem cronológica de desenvolvimento mais adiante.

4.2.1 - *Machine Readable Catalogue* (MARC)

O padrão MARC (USMARC, 1996), criado na década de 60 nos EUA, foi desenvolvido com o objetivo de possibilitar o intercâmbio de registros de catalogação entre bibliotecas, devido a informatização de seus serviços. A versão original do padrão é denominada de USMARC. Outras versões foram desenvolvidas em consequência das necessidades locais de cada biblioteca (sistemas de numeração local, idioma, etc.) e do tipo de material controlado, entre elas: UKMARC, UNIMARC, CANMARC e PicaMARC. Atualmente, esforços estão sendo despendidos na tentativa de unificação dessas versões, objetivando o intercâmbio de registros MARC a nível global.

São definidos formatos para cinco tipos de dados:

- **bibliográfico:** contém especificações de formato para a codificação de elementos de dados que são necessários para descrever, recuperar e controlar várias formas de material bibliográfico, tais como: livros, periódicos, arquivos de computador, mapas, músicas, materiais visuais, etc.;
- **armazenamento e localização:** contém especificações de formato para a codificação de elementos de dados relacionados ao armazenamento e localização de toda forma de material;
- **autoridade:** contém especificações de formato para a codificação de elementos de dados que identificam ou controlam o conteúdo e a designação de conteúdo das partes dos registros bibliográficos que são sujeitas a controle de autoridade. Uma autoridade define nomes e/ou valores que podem ser associados a um determinado elemento de dado;
- **classificação:** contém especificações de formato para a codificação de elementos de dados relacionados a números de classificação e a seus títulos associados. Registros de classificação são usados para manutenção e desenvolvimento de esquemas de classificação; e
- **informação comunitária:** contém especificações de formato para registros contendo informações acerca de eventos, programas e serviços.

A estrutura dos registros USMARC segue padrões nacionais e internacionais, dentre eles os padrões ANSI Z39.2 (Formato de Intercâmbio de Informações) e ISO 2709 (Formato para Troca de Informações). O formato de um registro MARC possui algumas divisões básicas, conforme exemplificado na Figura 4.1 a seguir:

Guia	Diretório	Campos de Controle Variáveis	Campos de Dados Variáveis
------	-----------	------------------------------	---------------------------

FIGURA 4.1 - Formato de um Registro MARC

Os 24 bytes iniciais correspondem à guia do registro, projetada para fornecer informações sobre o tamanho total do registro MARC e de algumas estruturas de dados do diretório e dos campos de dados. A guia também provê algumas informações básicas a respeito do tipo de material que o registro MARC descreve. O diretório é usado para indicar onde cada campo inicia e o tamanho de cada um. Pode possuir tamanho variável

dependendo do número de campos do registro. Cada entrada de doze bytes no diretório corresponde a um campo do registro MARC.

São poucos os campos de controle em um registro MARC. Eles são usados, em alguns sistemas, para armazenar informações tais como o número de controle do material descrito. Os campos de dados são mais numerosos pois armazenam informações bibliográficas sobre o material descrito. A estrutura dos campos de dados e de controle são idênticas: ambos inicializam com um indicador de campo seguidos por um número de sub-campos. Cada sub-campo é separado por um delimitador de sub-campo que, em muitos sistemas, é representado pelo caracter “\$” (internamente, entretanto, é armazenada apenas a forma binária do caracter). Ao final de cada campo do registro existe um delimitador de fim de campo e um delimitador de fim de registro, se for o caso.

Cada campo de dado em um registro USMARC é identificado por um rótulo de três caracteres. Cada rótulo é armazenado em sua entrada correspondente no diretório e não no próprio campo de dado. Campos de dados e de controle são agrupados em blocos de acordo com o primeiro caracter do rótulo, que identifica a função do dado dentro do registro USMARC. Os caracteres restantes do rótulo especializam o tipo da informação contida no campo (se é um nome de pessoa ou corporação, por exemplo).

Teoricamente, todos os campos do registro podem ser repetidos, com exceção dos campos número de controle, identificador do número de controle e a data/hora da última transação.

O conteúdo da maioria dos elementos de dados dos registros USMARC é definido por padrões fora do escopo do padrão USMARC, tais como as Regras de Catalogação Anglo-Americana (AACR2) ou os Cabeçalhos de Assunto da Biblioteca do Congresso. É a aplicação dessas regras que torna o padrão complexo e não a estrutura de seus registros em si. A seguir, um exemplo de utilização do modelo MARC (formato bibliográfico):

```
084 813.412
100 Ana Maria Moura
245 00 Metadata: The Essential Requirements... $h [computer file] / $c Ana Maria Moura
256 Computer file
260 [s.1.] : $b Departamento de Engenharia de Sistemas, Instituto Militar de Engenharia, $c [n.d.]
520 An article on metadata standards and models ....
856 7 $2 http $u www.des.ime.eb.br $d ~/~anamoura/publicacoes $f publicacoes.html $u
http:// www.des.ime.eb.br/~anamoura/publicacoes.html
```

Quanto aos projetos que utilizam o padrão MARC na Web, pode-se citar o projeto INTERCAT da OCLC (OCLC). Seu objetivo é averiguar a possibilidade de utilização das regras de catalogação AACR2 e do padrão MARC na construção de descrições bibliográficas dos itens acessados através da Internet. A OCLC emprega uma variação do padrão denominada OCLC MARC.

4.2.1.1 – Análise do Padrão

De uma forma genérica, modelos antigos, como MARC, oferecem uma visão muito complexa e rígida dos objetos descritos não possibilitando a criação “ad-hoc” de metadados de forma a atender a propósitos de comunidades específicas ou a necessidades ocasionadas pela própria evolução de conceitos. Qualquer alteração no padrão MARC (para inclusão de um novo campo, por exemplo) exige um consenso de sua comunidade internacional.

MARC herda também algumas limitações do estado da arte da computação da época em que foi desenvolvido (basicamente, é um formato binário projetado para armazenamento em fita magnética, tendo como principal objetivo a economia de memória do computador), fornecendo uma visão muito plana dos objetos descritos. Este fato dificulta o seu emprego para a descrição e representação de relacionamentos complexos entre coleções e/ou entre itens isolados.

4.2.2 - *Text Encoding Initiative Independent Headers (TEI)*

O padrão TEI (BURNARD, 1994) é um projeto cooperativo internacional que estabelece linhas de ação para a preparação e intercâmbio de textos eletrônicos. Teve sua origem em 1987 a partir de uma conferência organizada pela Associação de Computadores e Humanidades (ACH), cujo objetivo foi desenvolver um esquema padrão de codificação independente de *software*, *hardware* e área de aplicação e uma forma de representação explícita das principais características de um texto de modo a possibilitar o seu processamento por computador.

As diretivas TEI, publicadas em 1994, se aplicam a textos em qualquer idioma, qualquer gênero literário ou tipo de texto, sem restrições quanto a sua forma ou conteúdo.

Apesar de ter sido desenvolvido para atender às requisições da comunidade de pesquisa erudita, TEI pode ser útil também aos bibliotecários responsáveis por manter e documentar materiais eletrônicos. Embora o enfoque seja a representação eletrônica de textos que porventura já existam em qualquer tipo de mídia, é possível utilizar o padrão para a criação de novos textos.

O padrão TEI é uma aplicação da linguagem SGML, utilizada para definir um esquema de codificação de textos em termos de *elementos* e *atributos*. Elemento é uma unidade textual tal como um parágrafo ou o título de um documento. Os vários elementos do padrão são agrupados pelos seguintes conjuntos extensíveis e rotulados: **conjuntos núcleo** - contém os elementos do modelo necessários a todos os tipos de documentos; **conjuntos base** - são conjuntos de elementos específicos para documentos de uma classe em particular, tais como verso, prosa, poema, drama, etc.; **conjuntos adicionais** - são elementos apropriados para o tratamento especializado ou detalhado do texto em classes diferentes de documentos; **conjuntos auxiliares** - são elementos com papéis especializados.

As diretivas TEI especificam que todo texto codificado TEI deve ser precedido por um cabeçalho TEI descrevendo seu texto, codificação, fontes e revisões. A especificação do cabeçalho foi formulada como parte do projeto do *Comitê sobre Documentação de Textos*, formado por bibliotecários e arquivistas da Europa e da América do Norte, refletindo, portanto, em dogmas da catalogação tradicional.

No cabeçalho TEI, que é codificado como <teiHeader>, estas descrições são divididas em quatro partes principais:

- **descrição de arquivo:** codificada como <fileDesc>, contém uma descrição bibliográfica completa do arquivo e suas fontes, a partir da qual o usuário do texto pode derivar uma citação bibliográfica ou um bibliotecário pode criar de um registro de catálogo para o texto;
- **descrição da codificação:** codificada como <encodingDesc>, fornece uma descrição detalhada sobre como o texto foi normalizado durante sua codificação, como as ambigüidades da fonte foram resolvidas, quais níveis de codificação foram aplicados, etc.;

- **descrição do perfil:** codificada como <profileDesc>, contém informação contextual e classificatória sobre o texto, tal como seu assunto, a situação em que foi produzido, as pessoas que participaram de sua produção, etc.; e
- **descrição da revisão:** codificada como <revisionDesc>, contém informações sobre o histórico das alterações feitas durante o desenvolvimento do texto eletrônico, importantes para o controle de versões.

Os cabeçalhos TEI podem ser usados de três formas:

- codificados na mesma estrutura do texto TEI codificado. Neste caso podem ser criados pelo autor do documento ou, então, introduzidos por ocasião de sua codificação para fins de arquivamento;
- criados por terceiros como parte de um processo de controle bibliográfico, e mantidos separadamente do texto TEI que descrevem; e
- como cabeçalhos independentes, usados para descrever recursos que não sejam textos TEI codificados. É neste contexto que podem ser empregados para descrever qualquer recurso da rede.

A seguir, um exemplo de utilização do modelo TEI:

```

<TEIHEADER>
  <FILEDESC>
    <TITLESTMT>
      <TITLE> Metadata: the Essential Requirements..... </TITLE>
      <RESPSTMT><RESP>compiled by</RESP>
        <NAME>Ana Maria Moura</NAME>
      </RESPSTMT>
    </TITLESTMT>
    <PUBLICATIONSTMT>
      <DISTRIBUTOR>Instituto Militar de Engenharia</DISTRIBUTOR>
      <SOURCEDESC>
        <BIBL> Master Thesis Monography </BIBL>
      </SOURCEDESC>
    </PUBLICATIONSTMT>
  </FILEDESC>
</TEIHEADER>

```

A utilização do padrão TEI pela biblioteca da Universidade de Virgínia, EUA, faz parte de um projeto cujo objetivo é tornar acessível ao público, textos baseados no padrão SGML (SEAMAN, 1994). TEI provê um conjunto de rótulos, uma metodologia e várias DTDs que possibilitam uma descrição detalhada do conteúdo intelectual, da estrutura e da forma tipográfica de um trabalho disponível para o meio “on-line”. Após codificado, esse trabalho é adicionado a um banco de dados de textos, podendo ser pesquisado e analisado

como uma entidade individual ou como parte de um contexto mais abrangente. O projeto também provê conversão automática de textos TEI para a linguagem HTML e mecanismos para a inclusão de um cabeçalho TEI contendo informações bibliográficas básicas dentro do código binário de um arquivo de imagem. Dessa forma, todo texto ou imagem recuperada do banco de dados da biblioteca possui um conjunto de informações bibliográficas que são transferidas ao cliente como um anexo.

4.2.2.1 – Análise do Padrão

Um cabeçalho TEI provê informações bibliográficas completas da mesma forma que um registro de catalogação MARC. Entretanto, TEI fornece também um conjunto de informações não bibliográficas que descrevem o perfil, a codificação e o histórico de revisão do texto TEI codificado. Estas informações possibilitam a análise, manual e automática, do texto eletrônico documentado pelo cabeçalho. No padrão MARC essas informações só podem ser fornecidas em campos de anotações (contendo textos livres), que não fornecem a estrutura adequada às operações de pesquisa e recuperação.

A dificuldade de utilização do padrão depende do montante de informação detalhada contida no cabeçalho TEI e da conformidade de seu conteúdo com regras externas tais como a AACR2. Se o cabeçalho for criado da mesma forma que um registro MARC, segundo as mesmas práticas de catalogação, habilidades específicas serão exigidas. Entretanto, é possível criar cabeçalhos TEI simplificados cujo conteúdo não esteja de acordo com a AACR2.

Um cabeçalho TEI pode ser manipulado, pesquisado e recuperado por qualquer software que interprete registros SGML, tal como o navegador Panorama. O problema da utilização do padrão TEI para descrição dos recursos da Web advém do fato de que este foi projetado para codificar textos literários não incluindo, portanto, o tipo de metadado necessário às ferramentas auxílio à localização de recursos, ao contrário do padrão EAD (descrito a seguir) cujo principal enfoque são os metadados para acesso a acervos de documentos na forma eletrônica (GAYNOR, 1996).

4.2.3 - *Encoded Archival Description (EAD)*

O projeto EAD (SWETLAND, 1996) teve início em 1993 na Universidade da Califórnia, em *Berkeley*. O objetivo principal do projeto foi o desenvolvimento de um padrão não proprietário para a codificação de inventários, registros, índices e outras ferramentas similares criadas por museus, bibliotecas e arquivos que são utilizados para possibilitar a organização, localização e uso de seus acervos. Estas ferramentas descritivas, geralmente automatizadas, utilizam, em sua maioria, uma abordagem de catalogação tradicional baseada no padrão MARC.

Embora o termo “*finding aids*” esteja relacionado a uma grande variedade de ferramentas empregadas para descrever, controlar e prover acesso a arquivos e coleções, o modelo EAD enfoca primordialmente as funcionalidades dos inventários e registros, provendo elementos que possam suportar as funções de descrição, controle, navegação, indexação e apresentação dos materiais arquivados.

Seu esquema de codificação é baseado no padrão SGML e é representado na forma de uma definição de tipo de documento (DTD) denominada DTD EAD. Dessa forma, é possível pensar no modelo EAD como uma linguagem para intercâmbio e portabilidade entre instituições, permitindo o compartilhamento de dados entre ferramentas de auxílio a descoberta de recursos.

A DTD EAD possibilita quebrar a estrutura de um documento textual em *elementos* (também denominados de *rótulos EAD*) e, posteriormente, organizá-los segundo uma estrutura hierárquica, onde os elementos do nível mais alto fornecem uma visão mais genérica do material descrito. À medida em que se aprofunda na hierarquia dos elementos, um detalhamento maior do material descrito é obtido.

O modelo comporta dois tipos de elementos: *descritivos* e *genéricos*. Elementos descritivos ocorrem em pontos específicos do documento (ou ferramenta) descrito. Elementos genéricos podem ocorrer em qualquer local e são geralmente embutidos dentro dos elementos descritivos. Os principais elementos do modelo EAD são:

- **<ead>** - é o primeiro elemento de qualquer codificação EAD, funcionando como uma cápsula para os demais elementos do modelo;
- **<eadheader>** - contém informações de cabeçalho que descrevem a própria ferramenta descritiva;

- **<frontmatter>** - contém informações introdutórias (página de título, prefácio, dedicatória, introdução, etc.) que são necessárias para a publicação formal de uma ferramenta descritiva; e
- **<findaid>** - contém a descrição do conteúdo, contexto e extensão do material arquivado. É a parte mais complexa do modelo, devido ao número de elementos e sub-elementos que contém e também pelas decisões que devem ser tomadas quanto ao tipo de estrutura de elementos a ser empregada de forma a melhor refletir a organização do material descrito que dependerá também das práticas locais adotadas para o desenvolvimento de ferramentas descritivas. O elemento *componente* (codificado como “<c>”) é a parte da ferramenta descritiva que descreve o conteúdo, contexto e extensão de cada componente do material descrito.

Os elementos EAD são envoltos pelos caracteres “<” e “/>”, indicando o início e o término de cada elemento. Um elemento pode possuir vários atributos e cada atributo um valor. As diretivas EAD definem trinta atributos aproximadamente. Para cada atributo são especificados os elementos em que podem ser aplicados. No modelo EAD, qualquer atributo ou valor usado para descrever o material em um nível é automaticamente herdado pelos níveis subordinados, a menos que as diretivas especifiquem o contrário.

Um banco de dados com informações descritivas codificadas com base nas especificações EAD pode ser consultado no ambiente da Web. Para tal, é necessário que estas descrições sejam armazenadas em um servidor Internet. Também deve ser provido um mecanismo de busca associado a uma interface de usuário. A criação de registros EAD é possível através de um editor SGML ou através de um processador de textos.

A seguir um exemplo de utilização do modelo EAD:

```

<EAD>
  <EADHEADER> (contém informações descritivas e declarativas a respeito da própria ferramenta de
    auxílio à localização)
    <EADID>98765432</EADID>
    <FILEDESC>
      <TITLESTMT>
        <TITLEPROPER> Metadata: the Essential Requirement to Provide....</TITLEPROPER>
      </TITLESTMT>
    </FILEDESC>
  </EADHEADER>
  <FRONTMATTER> (contém informações introdutórias necessárias à publicação da ferramenta de auxílio
    à localização)
    <TITLEPAGE>
      <TITLEPROPER> Metadata: the Essential Requirement to Provide Effective Access to Resources in

```

```

the Web </TITLEPROPER>
  <PUBLISHER> System Engineering Department </PUBLISHER>
  </TITLEPAGE>
</FRONTMATTER>
<FINDAID> (contém informações administrativas e intelectuais a respeito do material descrito)
  <ARCHDESC AUDIENCE= "external" LEVEL="item">
  <DID>
    <UNITTITLE> Metadata: the Essential Requirement to Provide Effective Access to Resources in the
Web<UNITDATE>1997</UNITDATE></UNITTITLE>
    <PHYSDESC><EXTENT>3 linear feet.</EXTENT></PHYSDESC>
  </DID>
  <ADMININFO>
    <ACQINFO>
      <P> work produced from Master Thesis </P>
    </ACQINFO>
    <ACCESSRESTRIC>
      <P>The copywrite interests in the Moura, Ana Maria & Hess Lilia.....</P>
    </ACCESSRESTRIC>
  </ADMININFO>
  .....
</FINDAID>
  .....
</EAD>

```

Os principais projetos envolvendo o padrão EAD estão sob a gerência da Biblioteca do Congresso e de algumas universidades americanas, dentre elas: Berkeley, Yale, Harvard, Duke, Stanford e Virgínia.

4.2.3.1 – Análise do Padrão

EAD pretende ir um passo adiante da abordagem tradicional, provendo habilidade para apresentar informações descritivas relacionada às ferramentas para localização de recursos; preservar os relacionamentos hierárquicos existentes entre níveis distintos de descrição; representar a informação descritiva que foi herdada de outro nível hierárquico; e possibilitar a indexação e recuperação de elementos específicos.

Ao contrário do padrão MARC que enfoca na descrição bibliográfica completa de itens isolados, o padrão EAD foi projetado para descrever, de forma detalhada, recursos de informação estruturados hierarquicamente. A DTD EAD implementa uma estrutura recursiva, possibilitando a repetição de marcações em diversos níveis da hierarquia que está sendo descrita (coleção, série, pasta, item, etc.). Pode-se dizer que os padrões MARC e EAD representam abordagens descritivas complementares, na medida em que os registros MARC provêm uma representação condensada de uma coleção, enquanto que o

padrão EAD provê a estrutura necessária para a descrição detalhada da coleção e de seus itens. Na prática, o padrão EAD reconhece as especificações dos padrões TEI e USMARC como linhas de ação complementares.

4.2.4 - *Government Information Locator Service (GILS)*

O desenvolvimento do padrão GILS (GILS, 1994) teve início em 1994 incentivado pelo Governo Federal Americano, com a intenção de prover, ao público em geral e aos seus próprios funcionários, um serviço para localização de informações úteis, produzidas por diversas agências governamentais.

GILS foi projetado para cobrir uma grande diversidade de contextos e aplicações em diversas áreas (social, política, econômica ou temática) e tipos distintos de informações, incluindo pessoas, organizações, eventos, artefatos e recursos de informação da Web, tendo em vista a sua utilização pelo Sistema de Informação Global.

Formalmente, o padrão GILS é definido como uma coleção descentralizada de serviços e recursos de localização (“*locators*”). Um recurso de localização é definido como um recurso de informação para a identificação de outro recurso de informação, descrevendo seu conteúdo e possibilitando a sua localização. Cada recurso de localização pode conter informações de qualquer tipo, em qualquer nível de granularidade: um servidor de recursos GILS, por exemplo, pode ser visto como um recurso de informação descrito por outros recursos GILS.

Recursos de localização podem conter um ou mais atributos GILS. O núcleo básico do modelo é composto por aproximadamente 70 atributos, sendo que outros 100 são herdados do conjunto de atributos bibliográficos do protocolo Z39.50 (Z39.50), denominado Bib-1. O modelo GILS também possibilita a criação de atributos “ad-hoc” e a definição de recursos GILS sem nenhum atributo. Neste caso, é criado um atributo *default*, denominado “*anywhere*”, através do qual é possível efetuar uma pesquisa do tipo *full-text* sobre esse campo. A semântica dos elementos GILS é registrada em um componente de registro, não sendo necessário definir um formato “canônico” de transmissão nem uma estrutura para o metadado.

A interoperabilidade entre vários servidores GILS é garantida através de uma *profile* de aplicação. Uma *profile* declara uma função e o ambiente dentro do qual esta

função será utilizada, definindo um conjunto ou mais de padrões e identificando as classes, subconjuntos, opções e parâmetros desses padrões, de forma a garantir interoperabilidade em um contexto específico e limitado.

A *profile* GILS define como devem se comportar, para fins de comunicação, *softwares* clientes e *servidores*. Para tal, utiliza o protocolo aberto ANSI Z39.50 para especificar como as pesquisas a esses servidores podem ser expressas e como os resultados devem ser retornados. Z39.50 possibilita que bancos de dados distintos sejam pesquisados independentemente do tipo de interface de usuário utilizado.

Na *profile* GILS a definição do comportamento do servidor é feita de forma abstrata, ou seja, independentemente de como as informações são fisicamente gerenciadas pelo servidor. Dessa forma, cada empresa tem liberdade para organizar livremente suas informações com a garantia de interoperabilidade para pesquisa. Um servidor GILS aparece para um *software* cliente como um conjunto de recursos de localização que podem ser pesquisados.

As informações contidas no recursos de localização podem ser pesquisadas campo a campo, ou através de um mecanismo de pesquisa do tipo *full-text* (Alta Vista, por exemplo). Como a *profile* GILS não possui nenhuma interface de usuário, os servidores GILS devem ser acessados através de *gateways* (Web e X.500), clientes ou agentes.

Os serviços da *profile* GILS também possibilitam a consulta de informações em diversas estruturas: HTML, SGML, X.500, banco de dados SQL, Dublin Core, SOIF, IAFA, metadados espaciais, etc. Sempre que necessário, os servidores GILS efetuam um mapeamento da semântica local com os elementos GILS registrados. A versão 3 do protocolo Z39.50 possui uma facilidade denominada “*explain*” através da qual é possível descobrir, de forma dinâmica, quais atributos estão disponíveis para consulta em um servidor GILS. A pesquisa em vários idiomas também é favorecida pois os elementos do modelo são referenciados por números. Dessa forma, torna-se tarefa da interface do usuário traduzir o número do elemento para um idioma qualquer.

A seguir um exemplo de utilização do modelo GILS:

Title:	Metadata: The Essential Requirement to Provide...
Originator:	Ana Maria Moura
ContactNetworkAddress:	Anamoura@ime.eb.br
ContactTelephone:	55-21-295 3232
Abstract:	Metadata, or “information that makes data useful”, have been
ContactOrganization:	Instituto Militar de Engenharia
UncontrolledTerm:	Metadata, resource description

LinkageType: Text/plain
Linkage: <http://www.des.ime.eb.br/~anamoura/publicacoes.html>
LanguageOfResource: English

Além de ser utilizado pela maioria dos órgãos e agências governamentais nos EUA, diversos países (entre eles o Canadá e a Austrália) tem manifestado interesse na utilização do padrão GILS, bem como algumas companhias não governamentais, no desenvolvimento de servidores Z39.50 concordantes com a *profile* GILS.

4.2.4.1 – Análise do Padrão

GILS é um padrão complexo devido a sua amplitude de cobertura (inclui elementos para a descrição de dados geoespaciais e temporais simples, por exemplo) e por seu projeto ter sido bastante influenciado pelo padrão MARC. Embora seja possível a criação de registros GILS por pessoas não especializadas, GILS também possibilita a criação de registros de localização bastante ricos e complexos. Sua documentação provê, inclusive, um mapeamento com o padrão MARC. Quanto a sua utilização na Web, um aspecto bastante relevante é a facilidade que introduz à criação de metadados definidos localmente, ou seja, que não fazem parte o núcleo comum do padrão GILS.

4.2.5 – Quadro Comparativo entre os Padrões da Categoria

A Tabela 4.1 a seguir, apresenta, de forma resumida, as principais características dos padrões para documentação de acervos específicos, considerando a perspectiva de utilização desses padrões para a descrição de seus recursos de informação da Web.

TABELA 4.1 – Categoria Documentação (Quadro Comparativo)

Aspecto considerado	MARC	TEI	EAD	GILS
Comunidade utilizadora do padrão	- Bibliotecários, - Arquivistas, - Museus, etc.	- Comunidades acadêmicas dos EUA e da Europa, - Área de humanas, etc.	- Biblioteca do Congresso Americano, - Bibliotecas de diversas universidades nos EUA	- Agências e órgãos governamentais

TABELA 4.1 – Categoria Documentação (Quadro Comparativo) - continuação

Aspecto considerado	MARC	TEI	EAD	GILS
Facilidade de utilização do padrão	- Exige treinamento prévio e conhecimento especializado	- Depende do grau de detalhe empregado na descrição	- Exige experiência no manuseio de coleções e prática de arquivamento	- Pode exigir conhecimento especializado, embora não seja mandatário
Cobertura do padrão	- Documentos, vídeos, gravações sonoras, mapas, periódicos, etc.	- Qualquer tipo de texto em formato eletrônico	- Coleções de documentos, - Registros, - Inventários, etc.	- Possui cobertura muito ampla
Sintaxe / formato para transporte dos metadados	- ISO 2709	- DTD TEI	- DTD EAD	- USMARC, SUTRS, GRS-1
Estrutura Dos elementos	- Registros (formato interno binário)	- Rótulos SGML (composto por <i>elementos e atributos</i>)	- Rótulos SGML (composto por <i>elementos e atributos</i>)	- Pares (nome-atributo / valor-atributo)
Representação de relacionamentos entre recursos	- Descreve (mas não representa) relacionamentos hierárquicos (verticais) ou não (horizontais).	- Descreve e representa através do elemento “ <i>source description</i> ”.	- Descreve e representa relacionamentos (essencialmente hierárquicos) das coleções e de seus itens.	- Descreve (mas não representa) através do elemento “ <i>cross reference</i> ”.
Utilização de regras de conteúdo Para os elementos de dados	- AACR2, ISBD, Dewey Decimal, etc. (uso obrigatório)	- podem ser usadas (não são obrigatórias).	- Podem ser usadas (não são obrigatórias).	- podem ser usadas (não são obrigatórias).
Protocolos associados	- Z39.50	- não	- não	- não

4.3 - PADRÕES PARA A DESCOBERTA DE RECURSOS DA WEB

Alguns dos modelos desta categoria são bem recentes. Foram desenvolvidos a partir de 1994 com o objetivo de prover uma abordagem estruturada e padronizada aos índices de recursos criados e mantidos pelas ferramentas de coleta automática de metadados (HEERY, 1996a).

O enfoque dos padrões desta categoria é tornar mais eficiente o processo de descoberta de recursos, baseando-se em elementos descritores simples, que podem ser fornecidos pelo próprio autor do documento ou pelos gerentes de sítios. Esses elementos descritivos são bastante genéricos de forma a cobrir uma faixa ampla de recursos e

domínios distintos. O modelo de dados utilizado por esses padrões é, em sua maioria, simples, na forma de pares (nome-atributo, valor-tributo), podendo ser utilizados para pesquisa campo a campo (autor, assunto, etc.). Não foram projetados para expressar, de forma isolada, relacionamentos e hierarquias complexas entre objetos e sim para cobrir o intervalo existente entre os índices não estruturados e os padrões mais ricos e complexos, que exigem conhecimento especializado. Devido à similaridade de formato, a conversão entre padrões desta categoria é perfeitamente viável, tornando-os altamente interoperáveis.

Quanto ao emprego na Web, a tendência atual reforça a especificação de uma sintaxe comum (SOIF, RDF, etc.) para esses padrões, que possa ser interpretada pelos serviços de coleta e indexação automática de recursos.

Os padrões IAFA *Templates*, SOIF e Dublin Core, pertencentes à categoria, serão descritos (em ordem cronológica) a seguir:.

4.3.1 - Internet Anonymous Ftp Archive (IAFA)

FTP anônimo é um serviço, a nível de aplicação, que utiliza o protocolo FTP (“*File Transfer Protocol*”), um dos componentes do conjunto de protocolos TCP/IP. Antes do advento dos servidores HTTP (“*HyperText Transfer Protocol*”), o espaço FTP anônimo era o principal método de publicação de informações no ambiente Internet. Atualmente, os recursos FTP estão integrados aos demais recursos da Web através de diversas ferramentas de busca e recuperação de recursos.

Os modelos IAFA (DEUTSCH, 1994), desenvolvidos por um grupo IETF (“*Internet Engineering Task Force*”), teve como principal objetivo prover aos administradores do espaço FTP um conjunto de descritores para a documentação dos recursos de informação disponíveis em seus arquivos, tornando possível a sua utilização pelos mecanismos de indexação de recursos.

As diretivas do modelo, publicadas em 1995, especificam um tipo de formulário (ou “*template*”) para cada categoria de recurso disponível no espaço FTP, tais como: documentos, arquivos de dados, arquivos de listas de correspondência, arquivos Usenet e de E-mail, pacotes de software e arquivos de imagem. Alguns modelos são específicos para a gerência e administração de um sítio FTP, provendo informações sobre sua configuração, os serviços disponíveis (catálogos “on line”, servidores de informação, etc.),

etc. Os arquivos de índices baseados nos modelos IAFA são identificados pela extensão “.AFA” e podem existir em qualquer local da árvore de diretório do sítio.

Quanto a estrutura de dado empregada no modelo, cada elemento de dado é estruturado na forma de pares (nome-atributo, valor-atributo). Todos os elementos de dados podem ser repetidos com exceção do identificador único do recurso e dos campos “template-type” (tipo-do-formulário) e “template-version” (versão-do-formulário). O campo “template-type” identifica o tipo do recurso que está sendo descrito, podendo corresponder a:

- “SITEINFO” - provê informações sobre a configuração de um sítio FTP. Deve haver apenas uma instância deste formulário em cada arquivo “.AFA”;
- “ARCHIVE” - um arquivo físico pode conter vários arquivos lógicos. Cada arquivo lógico representa uma área dentro do arquivo físico que é gerenciada por pessoas ou instituições diferentes, por exemplo. Para os *softwares* que processam os modelos IAFA, o diretório no qual este arquivo se encontra localizado e todos os seus sub-diretórios fazem parte do arquivo lógico descrito;
- “MIRROR” - informações para atualização automática de arquivos;
- “SERVICE” - contém a descrição dos serviços oferecidos pelo sítio;
- “DOCUMENT”, “IMAGE”, “SOFTWARE”, “MAILARCHIVE”, “USENET”, “SOUND”, “VIDEO”, “FAQ”, etc. - estes formulários contém as mesmas informações. O elemento de dado “category” (categoria) serve para especializar o tipo de recurso descrito. “Relatório Técnico” e “Manual de Usuário” são exemplos de categorias que podem ser empregadas no formulário do tipo “DOCUMENT”.

Cada formulário do modelo IAFA corresponde a um agrupamento lógico de um ou mais elementos de dados. Nesse contexto, um elemento de dado é definido como uma porção de informação não necessariamente atômica.

Alguns elementos de dados que sempre ocorrem juntos em mais de um formulário IAFA são agrupados em estruturas denominadas *clusters*. Nos modelos IAFA, essas estruturas agrupam informações descritivas sobre *pessoas*, *organizações* e *recursos de informação*. Cada *cluster* recebe um identificador único a partir do qual é referenciado e um papel específico (tal como “USER” ou “ORGANIZATION”), sendo posteriormente associado a um determinado elemento de dado em um formulário IAFA tal como Publisher-(ORGANIZATION*) ou Author-(USER*). Este último, por exemplo, agrupa

informações sobre uma pessoa que exerce o “papel” de autor do documento, tais como “Author-name”, “Author-address”, “Author-Work-Fax”, etc.

Dentre as funcionalidades do modelo IAFA estão incluídos os campos “format-v*”, “language-v*” e “URI-v*” que são denominados de *campos variantes*. Estes campos são utilizados no modelo para representar o fato de que um mesmo conteúdo intelectual pode variar quanto ao *formato* (postScript, HTML, etc.), o idioma (inglês, português, etc.) e o *local de publicação* (URI). A sintaxe “-v*” (onde “*” representa um número seqüencial) possibilita agrupar um conjunto de descritores relativos a uma determinada versão ou cópia do recurso.

Um exemplo de codificação do padrão IAFA é apresentado a seguir:

Template-Type:	DOCUMENT
Template-Version:	1
Title:	Metadata: The Essential Requirements to Provide Effective
Author-Name:	Ana Maria Moura
Author-Email:	Anamoura@ime.eb.br
Author-Work-Phone:	55-21-295 3232
Last-Revision-Date:	15 July 1997
Description:	This paper describes the most important metadata standards and models.
Publisher-Organization Name:	Instituto Militar de Engenharia
Keywords:	Metadata, resource description
Format-v0:	Text/plain
URI-v0:	http://www.des.ime.eb.br/~anamoura/meta97.gz
Size-v0:	4 pages
Language-v0:	Portuguese
Format-v1:	Application/postscript
URI-v1:	http://www.des.ime.eb.br/~anamoura/meta97.ps.gz
Size-v1:	5 pages
Language-v1:	English

Diversas iniciativas baseadas no padrão IAFA estão em andamento na Web. Dentre elas destacam-se os projetos ROADS e DESIRE, mencionados no capítulo anterior (Seção 3.4.2), e o projeto *WHOIS++*, um protocolo e uma arquitetura de pesquisa distribuída que está sendo definida para a Web (FALTSTROM et alii, 1998). Ao contrário da maioria dos bancos de dados de índices criados pelos mecanismos de indexação de recursos da Web, as informações contidas nos bancos de dados Whois++ são estruturadas segundo um modelo de dados baseado nos formulários IAFA. Essa característica permite aos índices Whois++ trabalharem de forma colaborativa, trocando informações estruturadas entre clientes e servidores. Também possibilita a implementação e consultas por tipo de formulário, atributos específicos, etc.

4.3.1.1 - Análise do Padrão

IAFA possui a vantagem de ter sido projetado especialmente para uso na Web, não sendo indicado para descrever recursos em meio não eletrônico. Seu desenvolvimento foi incentivado por empresas privadas como parte de seus projetos relacionados à construção de ferramentas de navegação e serviços de diretórios da Web.

A principal vantagem do padrão é a sua facilidade de uso, cujo objetivo é possibilitar a cada provedor de recursos a provisão de seus próprios metadados, já que uma abordagem centralizada seria extremamente custosa em um ambiente como a Web. Por esse motivo enfoca apenas campos bibliográficos básicos, tais como título ou autor.

Quanto a codificação dos registros IAFA, estes são armazenados em ASCII. Cada elemento de dado IAFA, definido como um *par* (atributo-nome, atributo-valor), possui tamanho variável, sendo delimitado por caracteres especiais (“+” ou “-“, por exemplo) incluídos no texto. Para fins de recuperação, o conteúdo da maioria dos elementos de dado IAFA seguem padrões estabelecidos para a Web, tais como a RFC 822, para endereços eletrônicos (*e-mail*) e datas/horas, ou RFC 1521 para formatos de recursos.

4.3.2 - *Summary Object Interchange Format (SOIF)*

O padrão SOIF foi concebido como parte da arquitetura do sistema Harvest (BOWMAN et alii, 1994) na Universidade do Colorado, EUA, em 1994.

Harvest é um conjunto de ferramentas integradas para coletar, extrair, organizar, pesquisar, “*cachear*” e replicar informações relevantes na Internet. É composto por vários sub-sistemas que implementam uma arquitetura distribuída para captação (coleta e indexação) de metadados e consulta (campo a campo) a diversos tipos de recursos de informação da Web.

Desenvolvido como parte dessa arquitetura, SOIF especifica vários elementos descritores dos quais dezessete são comuns a todos os tipos de dados coletados, tais como *abstract*, *author*, *description* e *title*, além de uma sintaxe padrão para o intercâmbio de informações estruturadas entre os diversos sub-sistemas Harvest.

Na arquitetura Harvest, as informações captadas pelo sub-sistema *Coletor* são passadas a um sub-sistema, denominado *Essence*, que determina o tipo do dado recebido

para então aplicar um algoritmo de resumo que é específico para a extração de informações daquele tipo de dado. O resultado final do processo é um “sumário” (ou resumo) do conteúdo do recurso. Cada sumário de recurso é estruturado na forma de pares (atributo-nome, atributo-valor) e transportado segundo a sintaxe do padrão, conforme exemplificado a seguir. O atributo “*type*” contido em cada sumário identifica o tipo do recurso resumido (imagem, texto, compactado, e-mail, etc.). Um exemplo de codificação do padrão SOIF é apresentado a seguir:

```
@DELETE { }
@REFRESH { }
@UPDATE {
@FILE { ftp://www.des.ime.eb.br/~anamoura/hype96_7r_ps.gz
Title{84}: A Framework for Hyperdocument Generation in an Object Oriented
Database Environment
Author{34}: Moura, Ana Maria e Quadros, João
Time-to-live{7}: 9676800
Last-Modification-Time{9}: 774988159
Refresh-Rate{7}: 2419200
Gatherer-Name{50}: Personal Publications
Gatherer-Host{21}: leblon.ime.eb.br
Gatherer-Version{3}: 1.0
Type{10}: Compressed
Update-Time{9}: 774988159
File-Size{6}: 164373
MD5{32}: 43193942d53f5a8e4a7b4bcff7a415
Embed<1>-Nested-Filename{13}: hype96_7r_ps
Embed<1>-Type{10}: PostScript
Embed<1>-File-Size{6}: 28233
Embed<1>-MD5{32}: 84c123582c3d0754a39a78a78a7e2fb6d23
Embed<1>-Keywords{52}: hypermedia/hyperdocument/object oriented database
}
```

No exemplo, @DELETE, @REFRESH e @UPDATE são comandos que fazem parte da interface do sub-sistema Corretor. Embutidos dentro de um único comando @UPDATE e delimitados pelos caracteres “{“ e “}” podem constar vários elementos @FILE também delimitados por “{“ e “}“. Cada elemento @FILE contém um sumário do arquivo identificado pelo URL em questão. Vários atributos SOIF podem estar contidos dentro de um elemento @FILE, alguns específicos ao tipo de dado que está sendo descrito. O número entre chaves ao lado de cada atributo SOIF representa o tamanho do valor do atributo, o que torna possível o armazenamento de dados binários de tamanho arbitrário. O campo *Embed* pode ocorrer várias vezes, indicando o agrupamento de informações correlatas. No exemplo em questão, *Embed* agrupa os campos *Nested-filename*, *Type*, *File-size*, *MD5* e *Keywords*, contendo informações relativas a operação de descompressão efetuada no arquivo original durante o processo de resumo de seu conteúdo (observa-se que o valor do atributo *Type* para o arquivo em questão é “*compressed*”). Os caracteres “<” e “>” envolvem um número sequencial que serve para indicar um agrupamento específico.

Quanto a utilização do padrão na Web, a despeito de ter sido desenvolvido como parte da arquitetura Harvest, a sintaxe do padrão SOIF está sendo cogitada pelo projeto STARTS (“*The Stanford Protocol Proposal for Internet Retrieval and Search*”) iniciado em 1995, na Universidade de *Stanford* (USA), para transporte de informações estruturadas (metadados) entre meta-pesquisadores (tal como *savvy search*). Esses mecanismos foram criados com o objetivo de prover uma interface unificada para vários mecanismos de pesquisa distintos (Alta Vista, Excite, Infoseek, etc.), dando ao usuário a ilusão de uma fonte única de informação pesquisável (GRAVANO et alii, 1997).

4.3.2.1 – Análise do Padrão

O padrão SOIF é baseado em uma combinação dos padrões IAFA e BIBTEX (BIBTEX). Entretanto, ao contrário do padrão IAFA, SOIF provê suporte a coleções de sumários compactados contendo resumos de diversos recursos de informação. Essas coleções de sumários são trocadas entre os vários sub-sistemas Harvest, possibilitando que a arquitetura seja muito mais eficiente quando aos recursos consumidos na transmissão de informações e no espaço gasto para a indexação de recursos. SOIF também suporta

atributos de conteúdo arbitrário, podendo conter caracteres que não sejam ASCII ou dados em formato binário.

4.3.3 - *Dublin Core*

O modelo Dublin Core (WEIBEL et alii, 1995) é o resultado do primeiro “workshop” em metadados, ocorrido em março de 1995, em Dublin, EUA. Foi desenvolvido para descrever recursos do tipo documento, sendo composto, inicialmente, por treze elementos oriundos da catalogação descritiva.

Apesar de toda a simplicidade do modelo, alguns objetivos foram previamente definidos, tendo em vista um futuro mapeamento com outros esquemas. São eles:

- **Intrinsecabilidade:** se concentra na descrição das propriedades intrínsecas de um objeto que são descobertas a partir de uma simples inspeção visual;
- **Extensibilidade:** permite que o conjunto de elementos seja expandido para atender a propósitos especializados ou a futuras necessidades observadas do modelo Dublin;
- **Independência de sintaxe:** possibilita a utilização do modelo Dublin em uma grande faixa de disciplinas e aplicações;
- **Não obrigatoriedade:** alguns elementos não se aplicam de forma adequada a todo tipo de recurso;
- **Capacidade de repetição:** todos os elementos do conjunto podem ser repetidos; e
- **Capacidade de modificação:** permite que cada elemento do conjunto seja especializado de forma a atender a diferentes necessidades de diferentes comunidades.

Em janeiro de 1997, foi realizado em Dublin, Ohio, o terceiro “workshop” em metadados, organizado pela CNI (“*Coalition for Networked Information*”) e pela OCLC (WEIBEL & MILLER, 1997). O objetivo principal do encontro foi prover o modelo Dublin de um conjunto de descritores para imagens. Segundo o consenso de seus participantes, dentro do escopo do modelo e considerando a sua simplicidade, imagens e textos podem ser documentados de forma semelhante. São quinze os elementos do modelo definidos a partir de então, e relacionados na Tabela 4.2.

O quarto *workshop* em metadados (WEIBEL et alii, 1997), ocorrido na Austrália, em março de 1997, teve como objetivo deliberar sobre a estrutura dos elementos do modelo Dublin, seus mecanismos de extensibilidade e refinamento da semântica de cada

elemento. Basicamente, a discussão girou em torno de qualificadores, definidos como: um conjunto não ordenado de pares (nome-elemento, valor-elemento) únicos, que podem ser ligados a cada par (nome-elemento, valor-elemento) do modelo Dublin (KNIGHT & HAMILTON, 1996). Seu uso tem o propósito de informar ao usuário como ver ou interpretar o valor de um elemento ou também de estreitar ou tornar mais específica a semântica desse elemento.

Os qualificadores definidos para o modelo Dublin são descritos a seguir:

- **type (tipo)** - é usado para generalizar ou especializar a definição semântica de um elemento do modelo Dublin. Através do qualificador “type”, por exemplo, o elemento “author” pode ser complementado com outras informações relacionadas ao autor, tais como: seu número de telefone, email, fax, etc. Por exemplo: Creator (type=email) = “cassia@taurus.ime.eb.br”;
- **scheme (esquema)** – representa uma forma de introduzir padronização ao conteúdo dos elementos do padrão Dublin. O qualificador *scheme* possibilita que o valor de um elemento seja interpretado segundo o esquema (de codificação, de classificação, etc.) por ele especificado. Por exemplo: Title (scheme=AACR2) = “Metadados: a chave para organização dos recursos da rede”.

TABELA 4.2 – Elementos do Modelo Dublin Core

Nome do Elemento	Descrição do Elemento
<i>Subject</i>	Tópico relacionado ao objeto descrito.
<i>Title</i>	Nome do objeto.
<i>Author</i>	Pessoas diretamente responsáveis pelo conteúdo intelectual do objeto.
<i>Publisher</i>	Agente ou a agência responsável por tornar o objeto disponível.
<i>OtherAgent</i>	Pessoas, tais como editores e ilustradores, que contribuíram de forma significativa para o conteúdo intelectual do objeto.
<i>Date</i>	Data de publicação.
<i>ObjectType</i>	Gênero do objeto, tal como uma novela, um poema ou dicionário.
<i>Form</i>	Formato de dado do objeto, tal como postScript, HTML, etc.
<i>Identifier</i>	Números ou caracteres usados para identificar o recurso de forma única.
<i>Relation</i>	Categoria de relacionamento do objeto com outros objetos.
<i>Source</i>	Objetos dos quais o objeto descrito é derivado.
<i>Language</i>	Idioma relativo ao conteúdo intelectual do objeto.
<i>Coverage</i>	Localização espacial e duração temporal do objeto.
<i>Rights</i>	Contém ou referencia o direito de propriedade sobre o objeto.
<i>Description</i>	Contém uma descrição textual do objeto ou uma referência para essa descrição.

Exemplos de codificação do padrão Dublin Core serão apresentados no próximo capítulo, por ocasião da descrição da arquitetura Warwick.

Quanto aos projetos que utilizam o padrão, pode-se citar o projeto NÓRDICO (NÓRDICO, 1996), que envolve várias instituições da Escandinávia com o objetivo de criar um ambiente para a coleta, indexação e produção de metadados para a Web. As tarefas do projeto NÓRDICO incluem: a definição mais precisa do modelo Dublin Core; a conversão do modelo Dublin para os formatos MARC (versão nórdica) e vice-versa; a criação de diretivas para a sintaxe do modelo Dublin na linguagem HTML; a definição de algumas requisições para interfaces de usuário e o desenvolvimento de serviços de pesquisa baseados em metadados.

4.3.3.1 – Análise do Padrão

O padrão Dublin Core é a iniciativa baseada em metadados mais recente descrita neste capítulo e que, sem dúvida, foi o ponto de partida para os demais projetos envolvendo a especificação de uma arquitetura de metadados para a Web. Apesar de ser extremamente simples, seu projeto é fruto da grande experiência de profissionais envolvidos com a catalogação de recursos. É certo que o padrão MARC teve grande influência em seu desenvolvimento, porém foram extraídos desse padrão somente aqueles elementos essenciais à descrição e ao acesso de uma grande variedade de recursos de informação. Sua simplicidade é o fator chave para a rápida utilização na Web, embora já existam propostas para a utilização do padrão fora desse contexto na forma de uma aplicação integrando todos os tipos de recursos de informação, inclusive aqueles não disponíveis no meio eletrônico (RLG).

4.3.4 – Quadro Comparativo entre os Padrões da Categoria

As principais características dos padrões desta categoria são apresentadas, de forma resumida, na Tabela 4.3 a seguir, considerando a perspectiva de utilização desses padrões para a descrição de seus recursos de informação da Web.

TABELA 4.3 – Categoria Descoberta (Quadro Comparativo)

Aspecto considerado	IAFA/Whois++	SOIF	DUBLIN CORE
Comunidade utilizadora do padrão	- Provedores / gerentes de recursos de sítios Ftp	- Arquitetura distribuída Harvest, para coleta e indexação automática de recursos da Web.	- Autores de documentos da Web em geral
Facilidade de utilização do padrão	- Não exige treinamento nem conhecimento prévio	- O metadado é gerado de forma automática por um sub-sistema Harvest ("Essence")	- Não exige treinamento prévio nem conhecimento especializado
Cobertura do padrão	- Recursos gerenciados por um sítio Ftp em geral (documentos, imagens, configuração do sítio, etc.).	- Dados em qualquer tipo de formato, inclusive compactados	- Objetos do tipo documento (DLOs)

TABELA 4.3 – Categoria Descoberta (Quadro Comparativo) - continuação

Aspecto considerado	IAFA/Whois++	SOIF	DUBLIN CORE
Sintaxe / formato para transporte dos metadados	- Não possui	- O padrão fornece uma sintaxe também denominada SOIF	- É independente de sintaxe por concepção.
Estrutura Dos elementos	- Pares (nome-atributo, valor-atributo).	- Pares (nome-atributo, valor-atributo).	- Pares (nome-atributo, valor-atributo).
Representação de relacionamentos entre recursos	- Não.	- Não. Entretanto, possui o elemento "URL-references" através do qual interliga recursos distintos.	- Através do elemento "relation" é possível descrever relacionamentos hierárquicos ou não. Deve fazer uso das arquiteturas de metadados da Web para representá-los.
Utilização de regras de conteúdo Dos elementos de dados	- Utiliza alguns padrões de conteúdo da Web (MIME, RFC-822, etc.), a fim de facilitar a recuperação desses elementos.	- Alguns dos conteúdos gerados de forma automática podem ser concordes com padrões usados na Web.	- Através dos <i>qualificadores</i> é possível adicionar padrões de conteúdo aos elementos Dublin.
Protocolos associados	- Whois++	- Não.	- Z39.50.

4.4 – PROPOSTA DE CLASSIFICAÇÃO FUNCIONAL DO METADADO

Nesta seção será apresentada uma classificação funcional de metadados cujo propósito é servir como subsídio para uma comparação entre os diversos padrões analisados neste capítulo, tendo em vista a possibilidade de emprego na Web. A

classificação foi baseada nos componentes de metadados providos por cada padrão, sendo que o seu escopo se restringe aos metadados necessários para a descrição de um documento eletrônico no ambiente da Web. Estes elementos foram classificados, quanto a finalidade, segundo quatro categorias básicas, descritas a seguir:

1) Metadados para descoberta do recurso: esta categoria se refere aos elementos necessários à descoberta e identificação de um recurso de informação da rede. Toma como base os principais pontos de acesso a um recurso, identificados pelo comitê MARBI (MARBI, 1995) como: *informações bibliográfica básicas*, tais como título, autor(es), assunto e identificador único do recurso (URL, URI, etc.); *informações bibliográficas adicionais*, que servem para garantir que o recurso em questão é, de fato, aquele que se deseja recuperar: tipo da mídia (documento, imagem, vídeo, etc.) e gênero do recurso (relatório técnico, monografia, etc.), datas relacionadas (criação do recurso, por exemplo) e seus contribuintes (editores, compiladores, etc.); *características técnicas*, tais como o idioma e o formato do recurso, que auxiliam na seleção de uma instância que atenda aos requisitos de utilização do usuário; e *informações complementares*, que permitem relacionar o recursos com outros (suas referências bibliográficas, por exemplo). Analisando os padrões anteriormente descritos, observa-se que este conjunto de elementos varia sobremaneira com o propósito do padrão. De uma forma genérica, incluem:

- **Metadados para descrição bibliográfica básica:** são os elementos que apresentam maior variação de um modelo para outro devido ao gênero de recursos descritos. O padrão MARC, por exemplo, é, provavelmente, o mais completo enquanto que alguns dos padrões de metadados mais recentes, tais como Dublin Core e SOIF, possuem o mínimo de termos necessários para descreverem uma grande faixa de recursos. De uma forma genérica, este conjunto fornece elementos para a identificação, atribuição de responsabilidades e caracterização do recurso. Como exemplo de elementos desta categoria pode-se citar: título, autor, etc.;
- **Metadados para a identificação única do recurso na rede:** na Web, estes elementos são usados para prover uma identificação única a um recurso de informação;
- **Metadados para descrição de conteúdo:** são os elementos que descrevem o conteúdo intelectual de um recurso. Os modelos Dublin Core e IAFA/Whois++ incluem o campo

descrição para este propósito. Este campo pode conter um resumo, em prosa descritiva, de um recurso textual ou ser usado para referenciar um esquema descritivo mais complexo para mídias não textuais (uma imagem, por exemplo) ou referenciar uma descrição disponível em outro tipo de mídia que não a textual (um vídeo contendo uma exibição explicativa, por exemplo);

- **Metadados para descrição de assunto:** correspondem a um conjunto de termos (controlados ou não por um esquema de classificação) usados para se caracterizar o conteúdo de informação de um recurso. Para os recursos com características espaciais ou temporais, estas informações de referem aos limites de latitude e longitude cobertos e a duração;
- **Metadados para descrição de estrutura:** são os metadados usados para descrever a composição interna de um recurso de informação (imagens, gráficos, capítulos, etc.). No ambiente da Web uma tabela de conteúdos é geralmente usada para descrever a estrutura de um documento, fazendo uso de sua própria estrutura de hiperligações;
- **Metadados para descrição de relacionamentos:** estes metadados descrevem os tipos de associações entre recursos de informação. Por exemplo: o relacionamento de um recurso com suas referências bibliográficas, cópias, versões e instâncias (mesmo conteúdo intelectual em diferentes formas de apresentação);
- **Metadados para a descrição da proveniência do recurso:** esses metadados identificam os provedores (ou as fontes primárias) do recurso. O modelo Dublin Core considera a proveniência um tipo especial de relacionamento de um recurso com que compartilham do mesmo conteúdo intelectual; e
- **Metadados para descrição de mídia e de formato:** são elementos que descrevem as características de apresentação e a representação de dados dos componentes de um recurso, tais como: tipo de mídia, tamanho, formato, técnicas de compressão, etc. Além dos elementos comuns a todos os tipos de mídia, informações específicas podem ser incluídas, tais como o número de canais e a taxa de amostragem para um áudio ou, o número de quadros e tomadas de um vídeo. Bohm & Rakow caracterizaram o atributo idioma como específico das mídias textuais (BOHM & RAKOW, 1994). Alguns modelos (Dublin Core, por exemplo) associam o formato de um recurso a um tipo de conteúdo MIME (MIME), podendo assumir valores tais como: text/sgml, text/html or image/gif.

2) Metadados para Disponibilidade do Recurso: estes metadados definem os termos e condições requeridos para o acesso e a recuperação de um recurso, de forma restrita ou não. De uma forma genérica, incluem:

- **Metadados para distribuição do recurso:** esses elementos descrevem como o recurso de informação é tornado disponível aos usuários, especificando formatos, distribuidores, detalhes da publicação, informações para contato e formas de solicitação do recurso, etc. O padrão FGDC, por exemplo, possui uma seção de distribuição que informa como obter o recurso nas formas digital e não digital;
- **Termos e condições para acesso ao recurso:** são elementos que especificam se existem requisitos prévios que devem ser preenchidos para acesso ao recurso. São metadados que garantem o direito à privacidade e à propriedade intelectual. Podem ser expressos como uma requisição de permissão para acesso ou de identificação mediante a informação de um usuário e de uma senha;
- **Termos e condições para uso do recurso:** são elementos que especificam os termos e condições para utilização de um recurso. São geralmente expressos na forma de uma notícia de *copyright* que pode especificar políticas para se copiar, modificar ou remover um recurso, por exemplo;
- **Metadados sobre as requisições do recurso:** são metadados que especificam os requisitos de software e de hardware para uso do recurso, tais como os ajustes e configurações do ambiente e os aplicativos que devem ser utilizados para se visualizar ou carregar um recurso. O objetivo é evitar a transferência de dados pela Web sem as condições necessárias ao seu uso ou exibição;
- **Metadados para localização do recurso:** são metadados que provêm a informação necessária para se transferir um recurso de um servidor para um cliente. O propósito principal é possibilitar ao sistema selecionar uma cópia apropriada do recurso de forma a reduzir o tráfego de informações na rede. Para tal, pode ser utilizada uma função que efetue considerações sobre o custo de transmissão versus a localização do recurso;
- **Metadados sobre a autenticidade do recurso:** esses metadados servem para garantir que um recurso não seja modificado de forma indevida ou seja modificado por pessoas não autorizadas. Pode incluir desde os mecanismos mais simples (baseados no tamanho

do arquivo) até os mais complexos (baseados em esquemas de assinatura digital). O modelo SOIF, por exemplo, inclui o elemento MD5 para este propósito; e

- **Metadados para classificação de conteúdo do recurso:** esses metadados baseiam-se no conteúdo do recurso para classificá-lo segundo algum esquema provido por uma autoridade de classificação. A plataforma PICS, por exemplo, provê um mecanismo para associar rótulos de classificação de conteúdo a um documento da rede.

3) Metadados para Utilização do Recurso: são informações adicionais que possibilitam a utilização adequada do recurso. De uma forma genérica, incluem:

- **Metadados para a descrição da qualidade do dado do recurso:** esses metadados são particularmente importantes para determinadas aplicações tais como os Sistemas de Informações Geográficas. Os metadados desta categoria descrevem a validade, o percentual de acerto e a estimativa de erro das informações contidas em um recurso. Informações sobre a linhagem do recurso do recurso também podem ser providas (como no padrão SAIF, por exemplo);
- **Metadados para a descrição do propósito do recurso:** esses metadados descrevem o porquê do recurso de informação estar sendo liberado aos usuários, identificando programas, projetos, *forums* de discussão, etc. relacionados a esse recurso; e
- **Metadados para descrição contextual do recurso:** esses metadados provêm informações a respeito de eventos, situações, ambiente, etc., relacionados à produção do recurso. O padrão TEI, por exemplo, fornece o elemento <textDesc> que descreve a situação na qual um texto foi produzido; o elemento <partDesc> descreve os oradores, as vozes e participantes identificados em uma interação lingüística qualquer; e o elemento <settingDesc> descreve os parâmetros e ajustes do ambiente em que a interação lingüística ocorreu.

4) Metadados para Administração e Controle do Recurso: estes metadados provêm informações para controle e auditoria de um recurso de informação e de seus metadados associados. De uma forma genérica, incluem:

- **Metadados para controle das modificações do recurso:** são metadados usados para controle de versão do recurso, podendo descrever: as modificações introduzidas, datas relacionadas, responsáveis pelas modificações, informações para contato com esses

responsáveis, etc. O padrão TEI, por exemplo, provê o elemento <revisionDesc> para armazenar um registro ou histórico detalhado das mudanças ou revisões de um texto TEI;

- **Metadados para administração do recurso:** são metadados que permitem gerenciar e controlar o recurso, tais como: sua data de criação, período de validade, identificação de seus administradores, informações para contato, etc.;
- **Metadados para histórico de uso do recurso:** são metadados operacionais reservados para armazenar informações a cerca das operações executadas sobre o recurso (cópias, edições, exclusão, etc.), tais como: identificações dos responsáveis, data/hora das operações, etc. (BEARMAN, 1996); e
- **Metadados para administração do metadado:** são informações que permitem gerenciar e controlar os metadados associados a um recurso, tais como: sua data criação, última revisão, próxima revisão, identificação de seus administradores, informações para contato, nome e versão dos padrões usados, etc.

A partir da classificação funcional apresentada na seção anterior, é possível comparar os padrões descritos quanto aos conjuntos funcionais de metadados providos por cada um. O resultado da comparação é apresentado na Tabela 4.4 a seguir:

TABELA 4.4 - Metadados segundo Classificação Funcional Proposta

Conjunto Funcional de Metadados	IAFA/ WHOIS++	USMARC	Dublin Core	TEI	SOIF	EAD	GILS
Bibliográficos Básicos	x	x	x	x	x	x	x
Identificação Única do Recurso	x	x	x	x	x	x	x
Descrição do Conteúdo do Recurso	x	x	x		x	x	x
Descrição do Assunto	x	x	x	x		x	x
Descrição da Estrutura do Recurso		x				x	x
Descrição dos Relacionamentos		x	x	x	x	x	x
Descrição da Proveniência do Recurso	x	x	x	x	x	x	x
Descrição do Formato e da Mídia	x	x	x	x	x	x	x
Termos para a Distribuição do Recurso	x	x	x	x		x	x
Termos e Condições para Acesso		x				x	x
Termos e Condições para Uso do Recurso	x	x	x		x		x
Classificação de Conteúdo do Recurso		x					x
Requisições para Uso do Recurso	x	x			x		x
Localização do Recurso	x	x	x		x		x
Autenticidade do Recurso					x		
Descrição da Qualidade do Dado		x					x
Descrição do Propósito do Recurso		x					x
Descrição Contextual do Recurso				x		x	
Controle de Modificação do Recurso	x		x	x	x	x	

TABELA 4.4 - Metadados segundo Classificação Funcional Proposta - continuação

Conjunto Funcional de Metadados	IAFA/ WHOIS++	USMARC	Dublin Core	TEI	SOIF	EAD	GILS
Administração do Recurso	x		x			x	
Histórico de Uso do Recurso							
Administração do Metadado	x	x			x		x

4.5 –ANÁLISE COMPARATIVA DOS PADRÕES DE METADADOS

Quanto ao empregos dos padrões analisados para a descrição dos recursos de informação da Web, pode-se constatar, pela Tabela 4.4, que os padrões mais antigos não provêem metadados específicos ao uso na Web, tais como: termos e condições para acesso ao recurso; verificação da autenticidade do recurso, da qualidade dos dados ou a classificação do conteúdo. De uma forma geral, o emprego desses padrões na Web requer algumas considerações adicionais:

- **MARC:** apesar de seu formato arcaico, que requer sistemas especializados para a criação e intercâmbio de seus registros, o acervo disponível é muito grande, compensando o investimento em ferramentas que convertam esses registros para formatos mais simples e apropriados ao ambiente da Web (UKOLN). Nesse contexto, os elementos do padrão Dublin Core podem ser empregados como componentes de interoperabilidade, através de um mapeamento formal com padrões mais complexos e especializados, tal como MARC. A principal vantagem dessa abordagem é a garantia da precisão, a organização e a classificação, oferecidos pelos sistemas de catalogação bibliotecária;
- **TEI e EAD:** padrões baseados na SGML, de uma forma geral, são mais adequadas para a representação de relacionamentos hierárquicos, tais como aqueles existentes em sistemas de arquivos. Na Web, esses padrões possibilitam a implementação de catálogos multi-níveis, flexíveis e adaptáveis, onde os recursos são organizados segundo níveis crescentes de detalhamento. Esse tipo de organização permite ao usuário deslocar-se de um nível de detalhe para outro de forma a obter um determinado trabalho, suas versões, partes componentes, trabalhos relacionados, etc.;
- **GILS:** a amplitude de abrangência do padrão é, sem dúvida, um dos fatores que impulsionam o seu emprego na Web. Dentre outras vantagens do padrão, pode-se citar: o uso do protocolo Z39.50 para registro dos elementos do padrão; a possibilidade

de criação de metadados definidos segundo as requisições dos usuários e o mapeamento formal que provê com o padrão MARC; a transparência quanto ao formato de armazenamento dos dados pelo servidor GILS, etc. O aspecto mais importante, no entanto, é que algumas instituições já provêem acesso aos registros GILS por outros meios além do protocolo Z39.50, mais especificamente em servidores HTTP ou WAIS. Em alguns agências, registros GILS são usados como base para a geração automática de documentos HTML. Outras provêem acesso, via a tecnologia CGI (*“Commom Gateway Interface”*), aos servidores Z39.50, ou carregam os registros GILS em banco de dados acessíveis na Web;

- **IAFA/Whois++, SOIF e Dublin Core:** são os padrões mais adequados para emprego imediato na Web, devido a simplicidade de utilização. Além dos metadados bibliográficos básicos que são comuns a vários tipos de recursos, esses padrões provêem algumas categorias de metadados que são específicas à gerência de recursos de informação no ambiente da Web. O que se espera é que com o estabelecimento de arquitetura de metadados para a Web, esses padrões venham a ser amplamente empregados.

4.6 – CONSIDERAÇÕES FINAIS

Neste capítulo foram descritos os principais padrões de metadados que vêm sendo propostos para uso na Web. A princípio pôde-se constatar que a catalogação descritiva (ou bibliográfica), enfocada pela maioria dos padrões analisados, é apenas uma dentre as várias categorias de metadados que podem ser empregadas na Web. Na verdade, devido a sua natureza heterogênea e descentralizada, requisitos de metadados podem variar substancialmente quanto ao tipo de recurso descrito (que pode demandar elementos de domínio específico tal como “cobertura”, para dados espaciais e temporais, e “proveniência” para recursos em formato não digital) ou quanto propósito de criação do metadado. Autores de documentos da Web, por exemplo, fornecem metadados que são coletados pelos programas de indexação automática de recursos. Gerentes de sítios da Web, por sua vez, produzem metadados que auxiliam nas atividades de gerência, descoberta, localização, avaliação e seleção dos recursos de informação sob suas responsabilidades. O nível de detalhe e o enfoque da descrição também deve ser

considerado, uma vez que alguns padrões enfocam a descrição de itens isolados, enquanto que outros são projetados para descrever recursos organizados de forma hierárquica.

Até então, toda a experiência na utilização de metadados e de padrões de descrição se originou na catalogação bibliotecária. Mais especificamente, no emprego do padrão MARC e das regras de catalogação AACR2, ambos extremamente complexos para a maioria dos usuários leigos. O que se observa é que todo o formalismo necessário ao uso efetivo dos registros de catalogação bibliotecária pelos sistemas de biblioteca não se aplica de forma adequada ao ambiente da Web. Por outro lado, a simplicidade exigida de um padrão de metadados para o usuário leigo da Web pode resultar na pouca efetividade das descrições produzidas.

Conforme observado por diversos pesquisadores de padrões de metadados, é bastante improvável a adoção de um padrão único e monolítico para a descrição dos recursos de informação da Web. Qualquer esquema projetado para a descrição de tipos distintos de recursos deve ser flexível o bastante para capturar informações descritivas para uma variedade de propósitos. Esta diversidade de propósitos é refletida inclusive na terminologia empregada: termos tais como “coleções”, “esquemas”, “modelos” e “recursos”, encontrados na literatura, lidam com os mesmos problemas.

Também é certo que a necessidade por novos tipos de metadados continuará a se expandir a medida em que a infra-estrutura de informação da Web se tornar mais madura, requerendo outros tipos de aplicações (comércio eletrônico, por exemplo). Um padrão de metadados para a Web deve ser suficientemente flexível a fim de incorporar novas classes de metadados, cada qual representando o interesse de uma comunidade específica.

Nesse contexto, uma alternativa mais adequada é aquela que considera os diversos padrões de metadados como abordagens complementares, que podem ser integradas a fim de atender aos requisitos de comunidades de usuários distintas. Este é enfoque das arquiteturas de metadados que serão descritas no próximo capítulo: prover interoperabilidade semântica, estrutural e sintática ao universo de discurso do metadado.

CAPÍTULO 5

ARQUITETURAS GENÉRICAS DE METADADOS

5.1 – INTRODUÇÃO

Segundo Ianella (IANELLA, 1998), o principal propósito de uma arquitetura de metadados é prover suporte à codificação e ao transporte de uma grande variedade de metadados desenvolvidos de forma independente, maximizando a interoperabilidade do sistema através do uso de convenções comuns a respeito da *semântica*, *sintaxe* e *estrutura* do metadado.

O primeiro aspecto tratado por uma arquitetura de metadados, *interoperabilidade semântica*, possibilita a compreensão do significado de cada elemento componente dos diversos padrões de metadados definidos segundo às convenções estabelecidas pela arquitetura em questão. A semântica do elemento *creator* do padrão Dublin Core, por exemplo, pode ser definida como “a pessoa ou organização responsável pela criação do conteúdo intelectual de um recurso de informação”.

Duas estratégias são geralmente empregadas para se obter interoperabilidade semântica (KERHERVÉ, 1997): *bottom-up*, na qual são oferecidas ferramentas que provêem a integração de conjuntos distintos de metadados e dos procedimentos relativos a cada conjunto. Nessa estratégia, parte-se de diversos conjuntos de metadados projetados para atender às necessidades de uma determinada comunidade e tenta-se obter um conjunto integrado, geralmente bastante reduzido, que possa ser empregado de forma comum por esta comunidade. O padrão *Dublin Core*, na área de bibliotecas digitais, é um exemplo dessa abordagem; e *top-down*, que parte de um conjunto grande de metadados, bastante genérico, que é especializado ou adaptado para atender às necessidades de diversas comunidades e aplicações distintas. Neste sentido, é uma abordagem mais flexível e adaptável ao requisitos dos usuários, que sendo empregada pelas arquiteturas de metadados que estão sendo desenvolvidas para a Web (RDF, por exemplo).

O segundo aspecto tratado por um arquitetura de metadados, *interoperabilidade estrutural*, define cada elemento componente de um padrão de metadados, descreve os seus tipos, a escala de valores possíveis para esses elementos e os mecanismos utilizados para se agrupar (ou relacionar) esses elementos de modo a que possam ser processados de forma

automática. Quanto mais complexa for a estrutura do padrão de metadados, mais complexo deve ser o *modelo de dados* empregado para descrevê-la. Interoperabilidade estrutural implica na possibilidade de representação de estruturas de metadados simples do tipo par (nome-elemento, valor-elemento) empregadas por modelos tais como Dublin Core; na representação dos registros de metadados providos por padrões tais como MARC ou IAFA; ou mesmo, na representação de hierarquias de classes e de composição de classes complexas, tais como as que podem ocorrer entre os recursos de informação da Web.

O terceiro aspecto, *interoperabilidade sintática*, se refere a forma como metadados são codificados para transferência. A sintaxe provê uma linguagem comum para representar a estrutura do metadado. Nesse sentido, a linguagem XML vem sendo utilizada em diversas aplicações que necessitem trocar informações estruturadas na Web.

Neste capítulo são apresentadas as principais arquiteturas de metadados que estão sendo propostas para a implementação no ambiente da Web. Algumas enfocam requisitos de comunidades específicas, tais como a de Bibliotecas Digitais. Entretanto, todas possuem um requisito comum que é prover interoperabilidade entre aplicações distintas.

5.2 - ARQUITETURA WARWICK

5.2.1 – Conceitos Básicos

A arquitetura Warwick foi proposta como resultado do segundo *workshop* em metadados ocorrido em abril de 1996, em Warwick, no Reino Unido. Dentre os motivos citados para o seu desenvolvimento, pode-se destacar (LAGOZE et alii, 1996): promover interoperabilidade semântica através de disciplinas e linguagens e definir mecanismos de extensibilidade que dêem suporte simultâneo a modelos de descrição mais ricos e elaborados (tal como MARC) e a modelos descritivos mais simples (tal como Dublin Core).

Também denominada de “Arquitetura de Recipientes”, a arquitetura Warwick não define um conjunto de descritores, mas sim a maneira como dado e metadado podem se relacionar dentro de estruturas de dados específicas, projetadas para fins de modularidade do metadado.

5.2.2 – Modelo de Dados

Os principais componentes do modelo de dados da arquitetura Warwick estão descritos a seguir:

- **Recipiente:** é a estrutura básica para agregação de conjuntos de *pacotes*. Quanto à forma de utilização do recipiente, a arquitetura define duas categorias:
 - ❑ **Transiente:** quando o recipiente é criado apenas como meio de transporte entre repositórios e clientes, não necessitando de um identificador único de objeto;
 - ❑ **Persistente:** quando o recipiente possui prerrogativas de objeto de primeira-classe na infra-estrutura da informação. Neste caso, recebe um identificador único, através do qual é referenciado.
- **Pacote:** é uma estrutura de dado destinada a armazenar metadados de um determinado tipo. O nome do padrão de metadado em questão determina o tipo que é atribuído ao pacote. Dentro de um recipiente, um pacote representa uma seqüência obscura de bits, podendo ser criptografado para transmissão. Três tipos de pacotes são definidos pela arquitetura :
 - ❑ **Pacote de conteúdo:** contém metadados de um determinado tipo (Dublin Core, por exemplo);
 - ❑ **Pacote indireto:** implementa uma referência indireta a um outro objeto que pode estar armazenado em um espaço de informação qualquer;
 - ❑ **Pacote recipiente:** é um pacote que é também um recipiente, armazenando ou servindo como meio de transporte para outros pacotes. Essa possibilidade denota a característica recursiva da arquitetura Warwick.

A Figura 5.1 apresenta um recipiente da arquitetura Warwick contendo os seguintes objetos: um pacote de conteúdo, contendo metadados do padrão Dublin Core; um pacote de conteúdo, contendo metadados do padrão MARC; e um pacote indireto, referenciando o identificador único (URI) de um pacote de conteúdo externo que contém termos e condições para acesso a algum recurso.

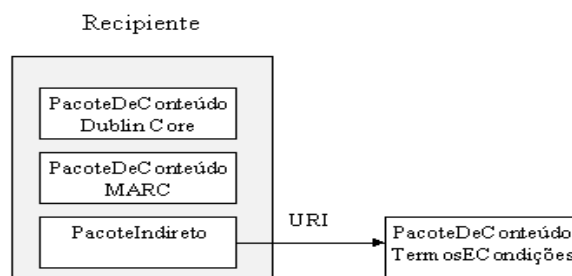


FIGURA 5.1 - Componentes da Arquitetura Warwick

A arquitetura Warwick possibilita diferenciar o metadado provido pelo autor do recurso, daquele metadado que é definido por um agente de indexação ou serviço de catalogação, cuja existência o responsável pelo recurso ignora, na maioria das vezes. Essa distinção é feita de acordo com o tipo de associação recipiente-objeto, determinada por três categorias de recipientes distintas:

- **Recipiente de metadados internamente referenciado:** é o metadado que o autor ou responsável pelo objeto selecionou para descrevê-lo ou que, pelo menos, tem conhecimento de sua existência. Neste caso, a associação recipiente-objeto pode ser feita de duas formas:
 - ◊ embutindo o recipiente de metadados na mesma estrutura do objeto. Neste caso o pacote de metadado não necessita de um identificador único, conforme apresentado na Figura 5.2 (objeto 1); ou
 - ◊ mantendo dado e metadado separados, mas ligados através de uma referência ao identificador único do recipiente de metadados (URI) que é embutido dentro da estrutura do objeto, conforme Figura 5.2 (objeto 2). Neste caso, o recipiente será obrigatoriamente um objeto de primeira classe, podendo ter seus próprios metadados associados.

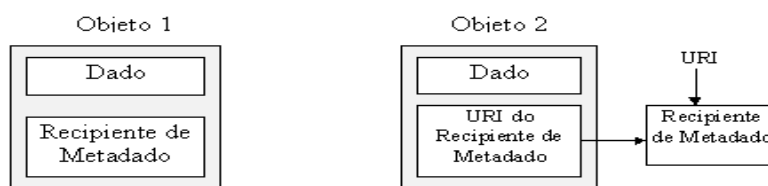


FIGURA 5.2 – Recipiente Internamente Referenciado

- **Recipiente de metadado externamente referenciado:** é o metadado criado e mantido por terceiros sem o conhecimento ou o consentimento prévio do responsável ou criador do objeto. Uma quantidade ilimitada de recipientes externamente referenciados podem fazer referência a um mesmo objeto. Cada pacote dentro do recipiente externamente referenciado é um objeto de primeira-classe, possuindo identificador único e, possivelmente, conjuntos de metadados a ele associados. Neste caso, a ligação do recipiente com o objeto que descreve é implementado através de um atributo ou *pacote de ligação* contido dentro do recipiente apenas. A Figura 5.3 exemplifica esse conceito.

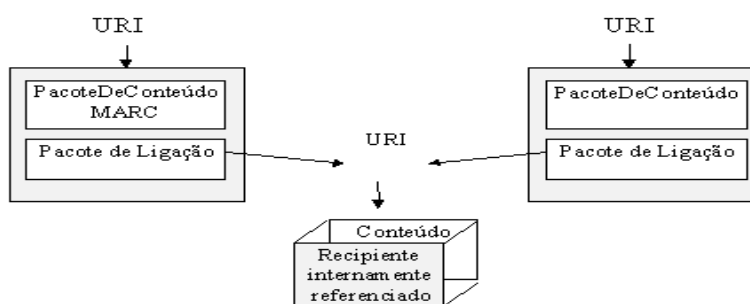


FIGURA 5.3 - Recipiente Externamente Referenciado

O mecanismo de associação de um recipiente de metadados com o objeto ao qual descreve irá depender da forma de implementação da arquitetura, cujas principais propostas encontradas na literatura são descritas a seguir.

5.2.3 – Implementação na linguagem HTML

A implementação em HTML da arquitetura Warwick é considerada a mais simples por não requerer alteração nos navegadores ou mecanismos de pesquisa existentes para a captação imediata do metadado contidos ou referenciados nos documentos HTML. Propostas para a associação dado-metadado utilizando o padrão Dublin Core são descritas em (MILLER, 1996a) (MILLER, 1996b) e (BURNARD et alii, 1996).

Um exemplo de utilização do modelo Dublin Core onde a estrutura pacote é embutida no cabeçalho do documento HTML é apresentado a seguir. Como a sintaxe do elemento META não foi modificada, a ferramenta de captação deve interpretar o conteúdo

da declaração META, que pode conter o elemento TYPE indicando o início (*begin*) ou fim (*end*) de um pacote.

```
<HEAD>
<TITLE>Metadata for the masses</TITLE>
<META NAME="package"      CONTENT="(TYPE=begin) Dublin Core">
<META NAME="DC.title"     CONTENT="Metadata for the masses">
<META NAME="DC.subject"   CONTENT="Dublin Core, Metadata">
<META NAME="DC.autor"     CONTENT="Paul Miller">
<META NAME="DC.autor"     CONTENT="(TYPE=phone) +44 191 222 8212">
<META NAME="package"      CONTENT="(TYPE=end) Dublin Core">
<META NAME="package"      CONTENT="(TYPE=begin) ahdsDescriptor">
<META NAME="AD.precision" CONTENT="(TYPE=spatial) 2">
<META NAME="package"      CONTENT="(TYPE=end) ahdsDescriptor">
</HEAD>
<BODY>
```

5.2.4 - Implementação segundo a Orientação a Objeto

A proposta orientada a objeto da arquitetura Warwick (LAGOZE et alii, 1996) define uma super-classe denominada Pacote a partir do qual as classes PacoteRecipiente, PacoteIndireto e PacoteConteúdo são especializadas, conforme apresentado na Figura 5.4.

- **classe Pacote:** é a super-classe para todos os tipos de componentes da arquitetura Warwick. Nenhum método foi especificado para esta classe;
- **classe PacoteRecipiente:** é a classe que implementa o componente recipiente da arquitetura Warwick. O método *ObterPacotes* definido para esta classe retorna um conjunto de referências aos pacotes armazenados no recipiente, cuja ordenação não é relevante para a arquitetura. O acesso ao conteúdo de cada pacote contido no recipiente deve ser obtido segundo os métodos definidos para cada classe correspondente;
- **classe PacoteIndireto:** é a classe que implementa o componente pacote indireto da arquitetura Warwick. O método *ObterIdentificadorDaIndireção* retorna o identificador do objeto que é referenciado de forma indireta. Um PacoteIndireto pode referenciar outro PacoteIndireto que, por sua vez, também pode referenciar um outro PacoteIndireto e assim sucessivamente. Por esse motivo, a arquitetura define também o método *ObterIndireção*, que resolve de forma recursiva a referência indireta até a obtenção do objeto “folha”; e

- **classe PacoteDeConteúdo:** implementa o componente pacote de conteúdo da arquitetura Warwick. Os métodos definidos para esta classe são dependentes do tipo de conteúdo do pacote.

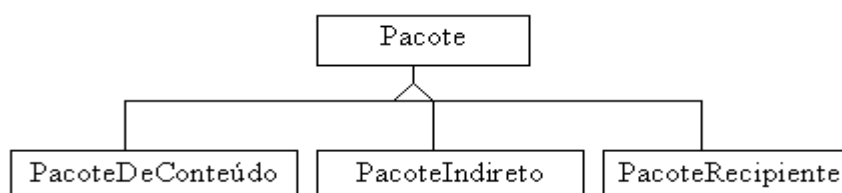


FIGURA 5.4 – Componentes da Arquitetura Warwick segundo a Orientação a Objetos

5.2.5 - Implementação segundo Conceitos de Objetos Distribuídos

A implementação da arquitetura Warwick segundo conceitos de objetos distribuídos (CORBA, DCOM, etc.) pressupõe uma infra-estrutura de informação bastante diferente da que existe atualmente na Web (LAGOZE et alii, 1996). Apesar de definir estruturas que permitem agrupar e associar dados e metadados de forma modularizada, a arquitetura Warwick não especifica como essas estruturas podem ser armazenadas ou mesmo negociadas entre clientes e repositórios em um ambiente distribuído de larga escala como a Web.

A arquitetura proposta por (KAHN & WILENSKY, 1995) define uma base para implementação dos pacotes e recipientes da arquitetura Warwick no contexto dos objetos distribuídos, definindo convenções para a nomeação, localização, segurança, autenticação e acesso aos componentes de uma biblioteca digital. Os principais elementos dessa arquitetura estão descritos a seguir:

a) Repositório

A estrutura do repositório da arquitetura de Kahn e Wilensky difere da estrutura de outros recipientes de conteúdo (banco de dados ou um sistemas de arquivos, por exemplo) na medida em que incorpora alguns metadados específicos, dentre os quais os mais importantes são: um identificador único para o repositório denominado *handle*, que é

fornecido por autoridade de nomeação qualquer; e *termos e condições*, que encapsulam regras para a obtenção de acesso ao repositório.

Uma segunda diferença em relação aos recipientes de conteúdo tradicionais, é que o repositório da arquitetura de Kahn e Wilensky armazena *objetos digitais*. Um objeto digital não é um objeto qualquer na forma digital, como um arquivo de imagem. Um objeto digital é uma estrutura de dados específica projetada para representar uma vasta categoria de objetos em formato digital que são acessados através da Web (ARMS, 1997) (BOWMAN & CAMARGO, 1998).

O acesso ou depósito dos objetos digitais contidos no repositório são feitos através de um protocolo simples definido pela arquitetura de Kahn e Wilensky, denominado RAP (*Repository Access Protocol*). Esse protocolo implementa dois métodos básicos: *DepositarObjetoDigital*, através do qual um novo objeto é armazenado no repositório; e *AcessarObjetoDigital*, que retorna uma referência a um objeto digital armazenado no repositório.

b) Objeto Digital

Basicamente, a estrutura de um objeto digital apresentada na Figura 5.5 é composta por *dados* e *metadados*. A porção *dado* corresponde ao conteúdo do material digital que está sendo representado pelo objeto digital. No caso do objeto digital representar uma imagem, por exemplo, sua porção *dado* poderá conter ou referenciar o conjunto de bits que compõem essa imagem. A porção *metadado* do objeto digital é composta por:

- um *registro de propriedades*, contendo um conjunto de metadados que descrevem o objeto representado pelo objeto digital. A arquitetura não especifica quais metadados devem fazer parte da composição do registro de propriedades. O único metadado obrigatório definido na arquitetura é denominado metadado-chave e é usado para identificar, de forma única, o objeto digital;
- um *registro de transações*, que deve armazenar as operações efetuadas sobre o objeto digital. Não é um componente obrigatório; e
- a *assinatura digital* do objeto, um metadado opcional que tem por objetivo garantir sua autenticidade.

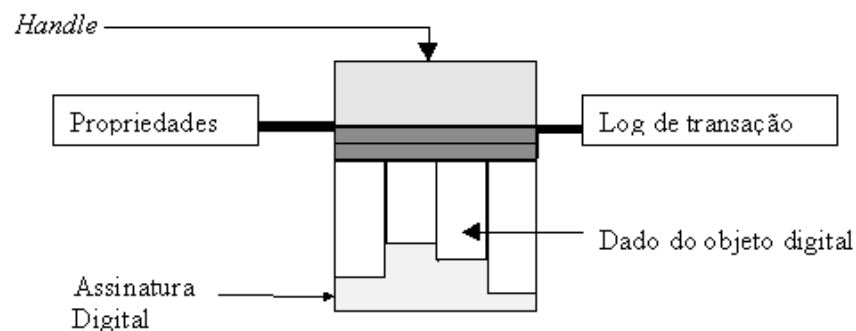


FIGURA 5.5 – Estrutura de um Objeto Digital

A arquitetura define o objeto digital denominado *MetaObjeto*, apresentado na Figura 5.6. Este armazena um conjunto de referências a vários outros objetos digitais, podendo ser usado para representar objetos de estrutura complexa, tal como um sítio Web ou um banco de dados, que são compostos por vários objetos de estruturas distintas.

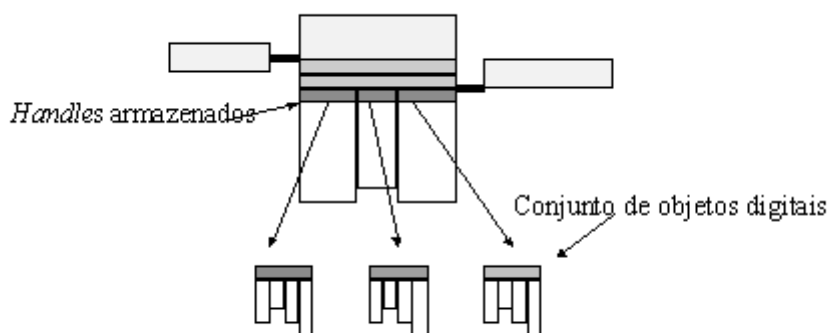


FIGURA 5.6 - MetaObjeto ou Objeto Catálogo

O dado de um objeto digital possui um tipo. Os tipos de dados definidos na arquitetura de Kahn e Wilensky são: *bit-sequence*, que corresponde a um conteúdo digital de recurso de informação qualquer, tal como um texto em “postScript”, uma imagem GIF ou um *applet* java; *digital-object*, indicando que um objeto digital contém outro objeto digital; *handle* indicando que o conteúdo de dado do objeto digital corresponde a uma referência para um outro objeto digital; *set-of-bit-sequence*, *set-of-digital-object* e *set-of-handles*, que indicam que o conteúdo do objeto digital corresponde a um conjunto de dados do tipo *bit-sequence*, *digital-object* e *handle*, respectivamente.

A arquitetura define dois operadores para a construção de tipos: *set-of* e *compose*, que opera sobre os tipos básicos definidos previamente. Cada tipo criado é cadastrado em

um registro global de tipos. É possível também criar sub-tipos de objetos digitais a partir da introdução de novos metadados no registro de propriedades de um objeto digital. Por exemplo, um objeto digital denominado “relatórioTécnico” pode conter metadados do tipo “autor”, “instituição”, “série”, etc.

Através do protocolo RAP é possível obter *disseminações* de um objeto digital. Disseminação é o resultado de uma operação de acesso sobre o conteúdo do objeto digital. A arquitetura prevê que o conteúdo do objeto digital armazenado não precisa ser (e na maioria das vezes não é) igual ao conteúdo que é disseminado ao cliente. Um objeto digital representando um documento eletrônico (armazenado como texto simples, por exemplo) pode prover disseminações desse documento nas versões PostScript e HTML. Uma imagem representada por um objeto digital pode ser disseminada nas versões *thumbnail*, mais apropriada à operação de folheio, e de alta resolução, para uma análise mais apurada de seu conteúdo. Ambas disseminações podem ser obtidas a partir da imagem original, armazenada na porção “dado” do objeto digital. Esse tipo de funcionalidade (conversão automática de formato) não é provida pelos recipientes de conteúdo tradicionais ou pelos diversos servidores da rede.

5.2.6 - Arquitetura de Kahn e Wilensky segundo a Arquitetura Warwick

A estrutura de um objeto digital interpretada segundo os conceitos da arquitetura Warwick (LAGOZE et alii, 1996) é apresentada na Figura 5.7. É composta pelos atributos *RecipienteDeMetadado* e *RecipienteDeDado*, ambos descritos a seguir.

- **RecipienteDeMetadado:** pode conter ou referenciar um conjunto de PacoteDeMetadados. Um PacoteDeMetadado contém metadados de um determinado tipo (MARC ou Dublin Core, por exemplo);
- **RecipienteDeDado:** contém um conjunto de *elementoDeConteúdo*, cada qual representando um componente estrutural do recurso de informação representado pelo objeto digital. Um ElementoDeConteúdo, por sua vez, pode conter seu conjunto próprio de descritores, armazenados em seu RecipienteDeMetadado. A porção “dado” de um elemento de conteúdo está representada por seu componente *PacoteDeConteúdo*. Um PacoteDeConteúdo pode conter dado de um determinado

tipo, tal como um texto ou uma imagem, ou possuir, de forma recursiva, a mesma estrutura de um ElementoDeConteúdo.

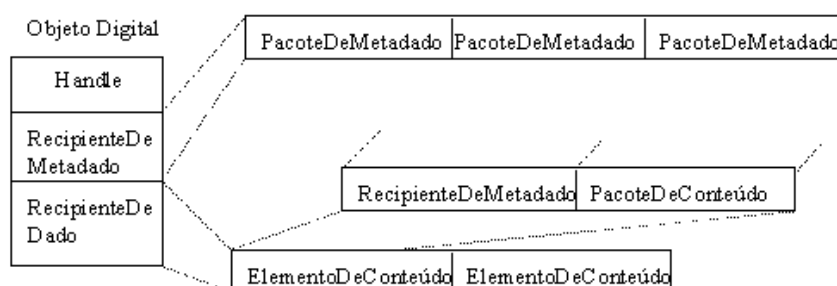


FIGURA 5.7: Estrutura de um Objeto Digital Segundo a Arquitetura Warwick

5.2.7 - Extensão da Arquitetura Warwick (Modelo “DARs”)

Uma extensão da arquitetura Warwick foi proposta em (DANIEL & LAGOZE, 1997). Essa extensão foi denominada de DARs - *Distributed Active Relationships* (Relacionamentos Ativos Distribuídos) e tem por objetivo definir um modelo para expressar relacionamentos entre os recursos da Web. O modelo DARs possibilita que esses relacionamentos sejam executáveis a partir de qualquer local e carregados de forma dinâmica. Seus princípios básicos estão descritos a seguir:

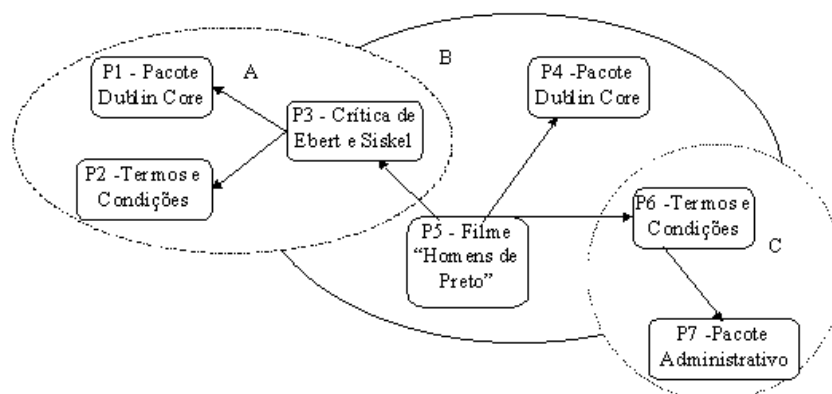


FIGURA 5.8 - Contextos Distintos de Uso do Dado / Metadado

- não existe distinção evidente entre dado e metadado. Um pacote pode estar no papel de “dado” ou no papel de “metadado”. A distinção é feita pelo tipo de *relacionamento* existente entre dois ou mais recursos de informação. A Figura 5.8 exemplifica esse

conceito. Três contextos distintos são demarcados: A, B e C. Dentro do contexto A, o pacote P3, contendo a crítica cinematográfica do filme “Homens de Preto”, está sendo descrito pelo pacote P1 (Dublin Core) e tem seus termos e condições para acesso descritos no pacote P2. Dentro do contexto A, portanto, P3 está atuando no papel de “dado”. No entanto, dentro do contexto B, O mesmo pacote P3 está no papel de “metadado” em relação ao pacote P5 que contém ou referencia o filme “Homens de Preto”, criticado por P3. De forma semelhante, o pacote P6 contém termos e condições para acesso ao pacote P5 no contexto B, enquanto que, no contexto C, está sendo descrito pelo pacote P7 representando, portanto, o papel de “dado”.

- podem existir vários tipos de relacionamentos entre recursos de informação. Um recurso pode ser uma agregação (ou um recipiente) de pacotes de dados que são conectados via relacionamentos. Daí o nome “distribuído” do modelo DARs;
- recursos podem estar relacionados independentemente da localização de cada um. A característica de *conectividade* da rede torna isso possível. Dessa forma, um recipiente da arquitetura Warwick, pode conter fisicamente ou referenciar um recurso localizado em qualquer espaço de informação da rede;
- o *poder computacional* do ambiente da rede possibilita a implementação de relacionamentos “ativos” (ou dinâmicos) entre recursos de informação. Este conceito implica em que o dado (ou metadado) não necessita ter existência física, podendo ser obtido a partir de algum processamento prévio.
- o *poder computacional* do ambiente da rede também possibilita que relacionamentos sejam componentes executáveis e não estáticos (tal como autor = “Cassia Barreto”). Essa abordagem é bem adequada para se tratar metadados complexos, tais como aqueles relacionados à gerência de direitos de recursos de informação, que podem requerer esquemas de acessos complexos envolvendo interação e negociação com serviços de autenticação, agentes, etc.

A fim de possibilitar a descrição dos relacionamentos existentes entre os pacotes dentro um recipiente da arquitetura Warwick, um novo componente denominado *catálogo* foi definido. Um catálogo é expresso como um pacote que contém uma lista de asserções sobre pacotes individuais e sobre o relacionamento entre esses pacotes. A Figura 5.9 exemplifica esse conceito. O catálogo do recipiente contém uma asserção especificando que o pacote P2 contém descrições bibliográficas sobre o pacote P1.

A arquitetura define que a descrição do relacionamento pode estar contida fisicamente no pacote catálogo ou ser apenas referenciada dentro deste, conforme apresentado na Figura 5.9. Um aspecto interessante sugerido pela extensão da arquitetura Warwick é a possibilidade de identificação de relacionamentos através de URIs. Esta abordagem evitaria a ocorrência de nomes coincidentes e possibilitaria o tratamento de relacionamentos como objetos de primeira classe, com direito a seus próprios metadados. Apesar de ser ainda muito incipiente, uma possível taxonomia para relacionamentos entre recursos foi proposta por Daniel (DANIEL, 1997).

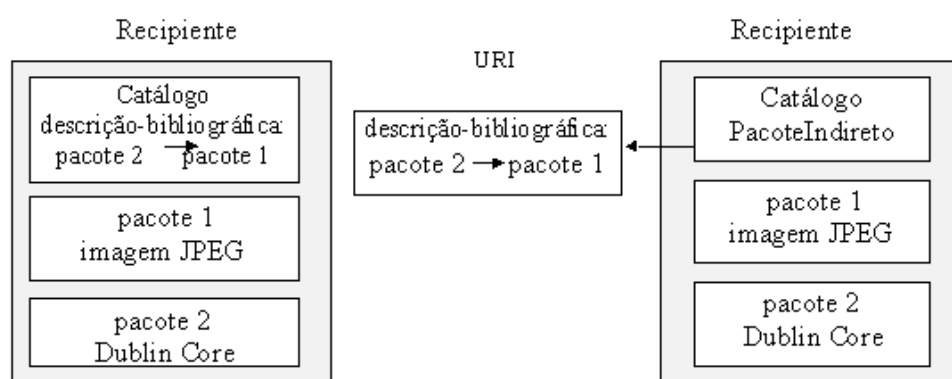


FIGURA 5.9 - Componente Catálogo da Arquitetura Warwick

A implementação dos conceitos da arquitetura Warwick estendida, associada aos conceitos da arquitetura de Kahn e Wilensky para bibliotecas digitais é descrita em (PAYETTE & LAGOZE, 1998). O protótipo em questão, denominado FEDORA (*Flexible and Extensible Digital Object Repository Architecture*), é baseado na arquitetura CORBA para objetos distribuídos.

5.3 - META CONTENT FRAMEWORK

5.3.1 - Conceitos Básicos

Meta Content Framework (MCF) (GUHA, 1997), desenvolvida sob os auspícios da Apple Computer Inc., é uma arquitetura aberta que provê um sistema de tipos para a representação da estrutura de organização da informação em diversos ambientes: Internet, Intranet, em redes locais ou mesmo em uma simples estação pessoal de trabalho (*desktop*).

Por estrutura de organização da informação entende-se que a MCF provê conceitos que permitem descrever propriedades, representar hierarquias e composições complexas de objetos tais como sistemas de arquivos, categorias de assuntos, diretórios de pessoas, páginas Web, etc.

No contexto da Web informações estruturadas podem servir para múltiplos propósitos, tais como: informar aos mecanismos de indexação quais informações devem ser indexadas; possibilitar a um navegador apresentar, de forma automática, o mapa de um sítio; possibilitar a efetivação de transações comerciais na rede; integrar fontes de informações distintas (*desktop*, *Web*, *email*, etc.) que são visualizadas através de metáforas distintas (*tree views*, *Web views*, *flythroughs*, etc.), etc.

5.3.2 - Modelo de Dados

A arquitetura MCF propõe um modelo de primeira ordem que possibilita a representação de objetos, seus atributos e relacionamentos com outros objetos segundo um conjunto de tuplas de aridade n (geralmente $n = 3$), onde cada tupla corresponde a uma *asserção* que declara a existência de uma propriedade relacionada a um objeto.

O modelo de dados MCF implementa seu modelo conceitual segundo a estrutura de um DLG (*Directed Labeled Graph*) cujos elementos básicos apresentados na Figura 5.10 são os *nós*, *rótulos* e *arcos*. Basicamente, o modelo de dados MCF compreende:

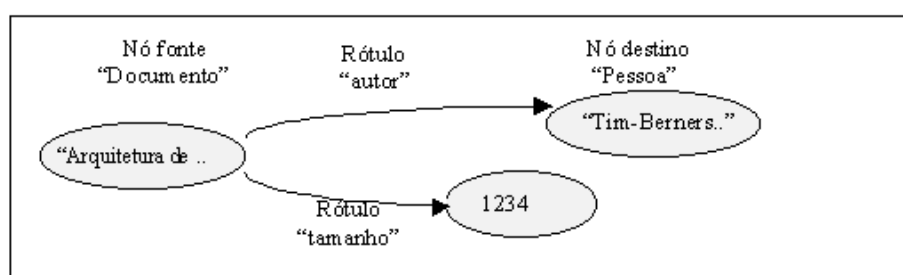


FIGURA 5.10 – Modelo de Dados da MCF

- um *conjunto de nós*, onde cada nó representa um objeto. Um nó pode corresponder a um tipo primitivo de dado (tal como inteiro, caracter, etc.) ou uma entidade do mundo real (um documento, uma imagem, uma pessoa, etc.);
- um *conjunto de rótulos*, onde cada rótulo corresponde a um *nome de propriedade*, tal como “autor” ou “assunto”, que pode ser associada a um nó; e

- um *conjunto de arcos*, onde cada arco corresponde a uma associação efetiva entre nós. A estrutura de dados de um arco corresponde a uma tripla consistindo de um nó fonte, um nó destino e um rótulo que nomeia essa associação.

5.3.3 – Sistema Inicial de Tipos da MCF

Os conceitos *nó*, *rótulo* e *arco* são implementados, respectivamente, pelos conceitos *Category* (Categoria), *PropertyType* (TipoDePropriedade) / *FunctionalPropertyType* (TipoDePropriedadeFuncional) e *Property* (Propriedade) que, com semânticas previamente definidas, inicializam o sistema de tipos da MCF.

Unit (Unidade) é o conceito mais genérico, representando a super-classe, implícita ou explícita, de todos os demais tipos definidos no modelo MCF. *Categoria* corresponde ao conceito de classe nos modelos orientados a objeto. É a partir desta unidade que os usuários definem os objetos de seu domínio de estudo. No modelo MCF, o nome de uma Categoria (“Pessoa”, por exemplo) expressa o tipo do objeto. Unidades e Categorias são sempre expressas em letras maiúsculas no modelo MCF.

TipoDePropriedade é uma Unidade que corresponde a um nome de propriedade (representado em letras minúsculas) que pode ser associada a uma Categoria qualquer, tal como “autor”, “assunto”, etc.

TipoDePropriedadeFuncional é uma Unidade que funciona como uma restrição de cardinalidade do modelo de dados MCF especificando que uma propriedade só pode se relacionar uma única vez com a Categoria na qual se aplica. Por exemplo, a propriedade “dataDeCriação” é um tipoDePropriedadeFuncional, pois só faz sentido uma única ocorrência dessa informação para um documento, por exemplo. A propriedade “autor”, no entanto, não é um tipoDePropriedadeFuncional já que um documento pode possuir vários autores.

Propriedade representa uma associação efetiva entre Categorias. No exemplo da Figura 5.10, “Tim Berners-Lee” (categoria “Pessoa”, destino da associação) é o “autor” (tipoDePropriedade) de “Arquitetura de Metadados para a Web” (categoria “Documento”, fonte da associação).

Para cada conceito do modelo de dados da MCF, algumas propriedades são previamente definidas. As propriedades associadas ao conceito Unidade são descritas a seguir:

- **typeOf (tipoDe):** é o tipoDePropriedade usado para se atribuir tipo a um objeto. Por exemplo, o tipoDe do objeto “Tim Berners-lee” é a Categoria “Pessoa”. O modelo MCF permite que um objeto pertença ao mesmo tempo a várias categorias;
- **id (identificador):** é o tipoDePropriedade usado para identificar uma unidade de forma única. Não é obrigatório;
- **name (nome):** é o tipoDePropriedade usado para nomear uma unidade. Não é obrigatório, porém as unidades TipoDePropriedade, TipoDePropriedadeFuncional e Categoria devem possuir nomes; e
- **description (descrição):** é o tipoDePropriedade usado para descrever uma unidade.

As propriedades previamente definidas para os conceitos TipoDePropriedade / TipoDePropriedadeFuncional são:

- **domain (domínio):** é o tipoDePropriedade usado para especificar a Categoria na qual se aplica uma determinada propriedade. Por exemplo, o domínio do tipoDePropriedade “homePage” é a categoria “Pessoa”. Isto significa que “homePage” é uma propriedade válida da categoria “Pessoa”;
- **range (faixa):** é o tipoDePropriedade usado para especificar restrições de tipo ao valor de uma propriedade. Por exemplo: a faixa do tipoDePropriedade “autor” é a categoria “Pessoa” (“Berners-Lee”, por exemplo);
- **superPropertyType (tipoDePropriedadeSuperior):** é o tipoDePropriedade usado para expressar uma relação de generalização entre propriedades. Por exemplo: “paisBiológicos” é tipoDePropriedadeSuperior do tipoDePropriedade “mãeBiológica” e do tipoDePropriedade “paiBiológico”;

As propriedades previamente definidas para o conceito Categoria são:

- **superType (tipoSuperior):** é o tipoDePropriedade usado para indicar uma relação de super-conjunto (ou especialização) entre Categorias. Por exemplo, “Pessoa” é tipoSuperior da categoria “Doutor”, porque um objeto da categoria “Doutor” também é um objeto da categoria “Pessoa”;
- **MutuallyDisjoint (disjuntoMutuamente):** é o tipoDePropriedade usado para representar uma relação de exclusividade entre duas Categorias especificando que um

determinado objeto não pode pertencer a ambas simultaneamente. Por exemplo, os objetos da categoria “*Inteiro*” são disjuntivos dos objetos da categoria “*Caracter*”; e

- **parent (pais):** é o tipoDePropriedade usado para representar a relação mais genérica entre categorias (é-um). No modelo MCF esta propriedade tem o sentido de “inclusão”, expressando que uma Categoria pode estar contida dentro de outra. A propriedade *parent* poderia ser utilizada para expressar o relacionamento de um sítio Web contendo várias páginas HTML, por exemplo.

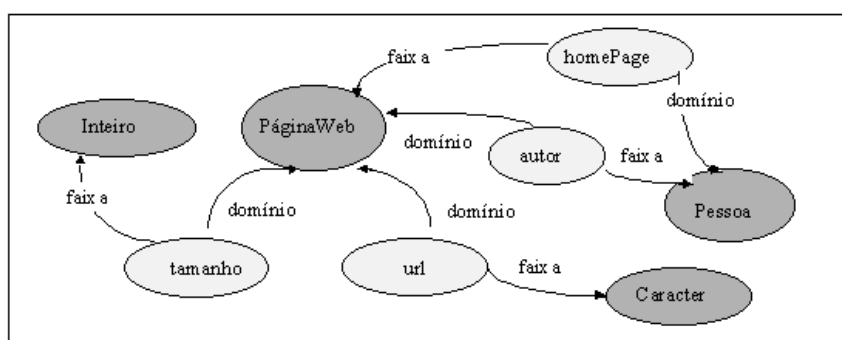


FIGURA 5.11 - Definição de um Esquema de Metadados segundo Conceitos da MCF

A Figura 5.11 representa um esquema de metadados definido segundo os conceitos da MCF: “PáginaWeb”, “Pessoa”, “Inteiro” e “Caracter” representam Categorias; “url” e “tamanho” são exemplos de tipoDePropriedadeFuncional; e “autor” e “homePage” são exemplos de tipoDePropriedade.

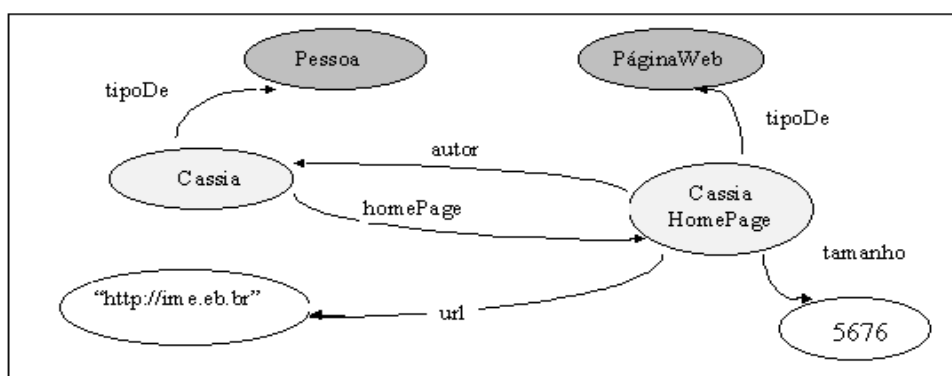


FIGURA 5.12 - Representação de um Esquema de Metadados Instanciado

Na Figura 5.12 o esquema anteriormente definido foi instanciado. O objeto “Cassia”, é uma instância da categoria “Pessoa”, o objeto “Cassia HomePage” é uma

instância da categoria “PáginaWeb” . O atributo “autor” recebe como valor o objeto “Cassia” e “homePage” , o objeto “CassiaHomePage”. O valor dos atributos “url” e “tamanho” são instâncias das categorias Inteiro (5676) e Caracter (<http://ime.eb.br>).

5.3.4 - Implementação da MCF na Linguagem XML

Objetos, categorias e propriedades necessitam de uma sintaxe que possibilite sua representação para fins de intercâmbio. Guha (GUHA & BRAY, 1997) propõe a utilização da linguagem XML para a representação dos conceitos da MCF.

MCF usa os *elementos* da XML para representar objetos. As propriedades dos objetos são representadas por elementos contidos dentro do elemento que representa o objeto. O tipo do elemento representa a categoria a qual pertence o objeto. Se um objeto pertencer a mais de uma categoria, qualquer uma servirá para caracterizar o seu tipo. A Figura 5.13 contém um exemplo de um elemento XML de nome “P” que possui um atributo denominado “atrib”, cujo valor é igual a “valor-atributo”.

<code><P atrib = “valor-atributo” ></code>
--

FIGURA 5.13 - Elemento XML

O exemplo a seguir, baseado na Figura 5.11, demonstra como os conceitos da MCF podem ser expressos na linguagem XML. O atributo “unit” associa uma propriedade a um outro objeto identificado pelo valor desse atributo.

```
<Category id = “PáginaWeb”>
  <description> esta categoria representam todas as páginas Web criadas</description>
</Category>
<Category id = “Pessoa”>
  <description> esta categoria representa uma pessoa qualquer</description>
</Category>
<Category id = “Caracter”>
  <description> esta categoria representam um tipo primitivo de dado do tipo caracter
</description>
</Category>
<Category id = “Inteiro”>
  <description> esta categoria representam um tipo primitivo de dado do tipo inteiro
</description>
</Category>
```

```

<PropertyType id = "autor">
  <domain unit = "PáginaWeb"/>
  <range unit = "Pessoa"/>
</PropertyType>
<FunctionalPropertyType id = "url">
  <domain unit = "PaginaWeb"/>
  <range unit = "Caracter"/>
</FunctionalPropertyType>
<FunctionalPropertyType id = "tamanho">
  <domain unit = "PaginaWeb"/>
  <range unit = "Inteiro"/>
</FunctionalPropertyType>
<PropertyType id = "email">
  <domain unit = "Pessoa"/>
  <range unit = "Caracter"/>
</PropertyType>
<PropertyType id = "homePage">
  <domain unit = "Pessoa"/>
  <range unit = "PáginaWeb"/>
</PropertyType>

```

No exemplo a seguir, o esquema instanciado apresentado na Figura 5.12 é representado na sintaxe XML.

```

<PáginaWeb id = "CassiaHomePage"
  <url>http://ime.eb.br/tese.html</url>
  <autor unit = "Cassia"/>
</PáginaWeb>
<Pessoa id = "Cassia">
  <email>cassia@taurus.ime.eb.br</email>
  <homePage unit = "CassiaHomePage"/>
</Pessoa>

```

5.4 - RESOURCE DESCRIPTION FORMAT (RDF)

5.4.1 - Conceitos Básicos

A arquitetura RDF (RDF, 1997) é uma iniciativa em desenvolvimento sob os auspícios do consórcio W3C que tem por objetivo prover interoperabilidade entre aplicações que necessitam trocar informações estruturadas na Web.

Na área de descoberta de recursos, por exemplo, RDF possibilita a implementação de mecanismos de pesquisa mais eficientes. Na área de catalogação RDF pode ser usado para descrever os recursos de informação em um sítio da Web, em uma biblioteca digital,

etc. Na área de agentes inteligentes RDF pode facilitar o intercâmbio de informações e o compartilhamento de conhecimentos.

5.4.2 - Modelo de Dados

A principal finalidade do modelo de dados da RDF é possibilitar a representação das propriedades de recursos de informação e de seus valores. O modelo RDF básico consiste de três tipos de objetos:

- **Recurso:** representa tudo que é descrito segundo o modelo RDF;
- **Propriedade:** representa um aspecto específico, característica, atributo ou relação usada para descrever um recurso; e
- **Declaração:** corresponde a associação de um recurso específico, uma propriedade e o valor dessa propriedade para esse recurso.

A representação formal de seu modelo de dados está descrita a seguir:

- existe um conjunto denominado Recursos (*Resources*);
- existe um conjunto denominado Literais (*Literals*);
- existe um subconjunto de Recursos denominado Propriedades (*Properties*);
- existe um conjunto denominado Declarações (*Statements*). Cada elemento desse conjunto representa uma tripla na forma [pred, sub, obj], onde: *pred* é uma propriedade (membro do conjunto Propriedades; *sub* é um recurso (membro de recursos); e *obj* é ao mesmo tempo um recurso ou um literal (membro de Literais);

Pode-se visualizar um conjunto de declarações (membros do conjunto Declarações) como um DLG (*Directed Labeled Graph*): cada recurso ou literal representa um vértice; e a tripla {pred, sub, obj} é um arco partindo de sub para obj e rotulada por pred.

No exemplo da Figura 5.14, a declaração “*Cassia Barreto é autora do documento cujo URL é http://www.ime.eb.br/cassia/metadados.html*” pode ser representada como um grafo da seguinte forma:



FIGURA 5.14 - Declaração RDF

Representada segundo o modelo de tuplas ternárias (que é equivalente a representação anterior), a mesma declaração daria origem a seguinte tupla: {autora, [http://www.ime.eb.br/cassia/metadados.html], “Cassia Barreto”, onde “autora” representa o arco que une os nós fonte e destino. A relação expressa pelo arco (ou declaração RDF) pode ser convertida em um recurso RDF (ou nó “X”) da seguinte forma:

```
X ---- type (tipo) ----- > [RDF:Statement]
X ---- predicate (predicado) ----- > [autora]
X ---- subject (sujeito) ----- > [http://www.ime.eb.br/cassia/metadados.html ]
X ---- object (objeto) -----> “Cassia Barreto”
```

Representado segundo a estrutura de um DLG (Figura 5.15), o nó criado (recurso “X”) deu origem a quatro novas tuplas ternárias que são membros do conjunto Declarações. No modelo RDF os elementos dessas tuplas são conhecidos formalmente por: *RDF:type*, o qual provê um modo de se atribuir um tipo a um nó (que no exemplo em questão é uma declaração); *RDF:predicate*, identificando o nome da propriedade; *RDF:subject*, identificando o objeto que possui a propriedade; e *RDF:Object*, identificando o valor da propriedade.

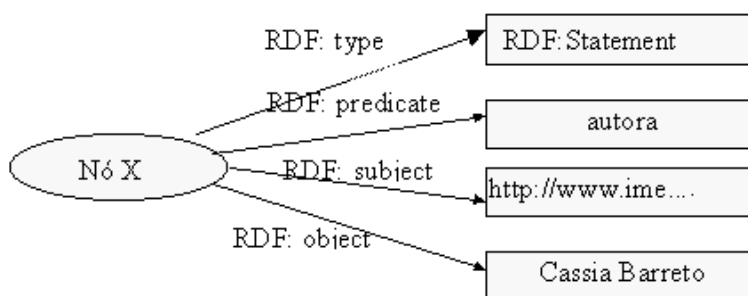


FIGURA 5.15 - Transformação de uma Declaração RDF em um Recurso RDF

A definição formal de *tipo* no modelo RDF é:

- existe um elemento do conjunto Propriedades conhecido como *RDF:type*;
- membros do conjunto Declarações na forma {*RDF:type*, sub, obj} devem satisfazer ao que se segue: sub e obj são membros do conjunto Recursos. Nenhuma outra restrição é imposta quanto ao uso da propriedade *type*.

O modelo RDF permite criar coleções de recursos ou de literais. Coleções permitem declarar que um trabalho foi criado por vários autores, listar os estudantes de um curso, ou os módulos de um pacote de software. Três tipos de coleções RDF são definidas: listas ordenadas, denominadas Sequências (*Sequences*); listas não ordenadas, denominadas Malas (*Bags*); e listas que apresentam alternativas para os valores prováveis de uma propriedade, denominada Alternativas (*Alternatives*).

A Figura 5.16 exemplifica duas coleções RDF do tipo “RDF:seq”. Uma coleção organiza seus documentos por data e a outra por assunto. Os arcos que partem do nó criado apontam para cada membro da coleção e são rotulados de forma única através dos elementos do conjunto de ordinais (1,2,3...) que, no modelo RDF, é denominado Ord. Os elementos do conjunto Ord são denominados de RDF:_1, RDF:_2, ..., RDF:_n.

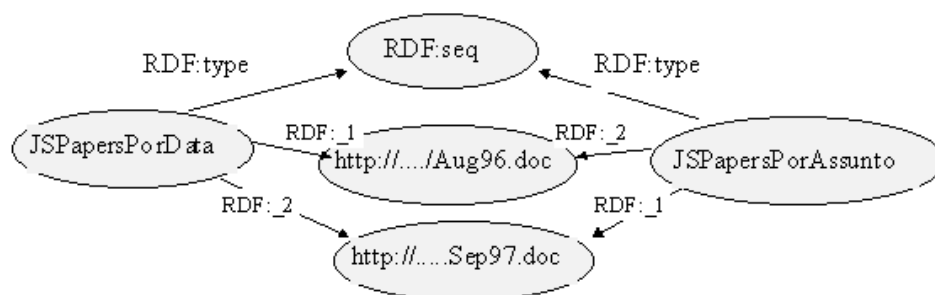


FIGURA 5.16 - Coleções RDF

5.4.3 – Implementação da RDF na Linguagem XML

RDF usa a linguagem XML como linguagem de codificação para o intercâmbio de metadados criados segundo o seu modelo de dados. Duas sintaxes são propostas: de *serialização*, que expressa toda a potencialidade do modelo RDF; e *abreviada*, que inclui construtores adicionais que expressam de forma mais compacta o modelo de dados RDF.

Na Web, descrições RDF podem ser associadas aos recursos que descrevem de quatro formas:

- a descrição pode estar contida dentro do recurso, embutida no cabeçalho de um documento HTML ou referenciada através do elemento <LINK> da seguinte forma:
<LINK rel="meta" href="mydocMetadata.DC.RDF">;

- a descrição pode ser externa ao recurso, porém fornecida pelo mecanismo de transferência (o protocolo HTTP, por exemplo) na mesma transação de recuperação do recurso;
- a descrição pode ser recuperada de forma independente do recurso e obtida de uma fonte distinta (um serviço de “*bureau*”, por exemplo); e
- a descrição pode conter (ou envolver) o próprio recurso.

A seguir serão demonstrados algumas potencialidades do modelo RDF utilizando a sintaxe RDF para expressar esquemas instanciados contendo:

a) Declarações simples: “*Cassia Barreto é autora do documento cujo URL é <http://www.ime.eb.br/cassia/metadados.html>*”

```
<rdf: RDF>
  <rdf: Description about = “http://www.ime.eb...../metadados.html”>
    <s: autora>Cassia Barreto</s: autora>
  </rdf: Description>
</rdf: RDF>
```

b) Coleções: “*Os estudantes do curso de inglês são João, Maria, José, Pedro e Ana*”

```
<rdf: RDF>
  <rdf: Description about = “http://meusColegas.edu/curso/inglês”>
    <s: estudantes>
      <rdf: Bag>
        <rdf: li resource=”http://meusColegas.edu/estudantes/Joao/>
        <rdf: li resource=”http://meusColegas.edu/estudantes/Maria/>
        <rdf: li resource=”http://meusColegas.edu/estudantes/Jose/>
        <rdf: li resource=”http://meusColegas.edu/estudantes/Pedro/>
        <rdf: li resource=”http://meusColegas.edu/estudantes/Ana/>
      </rdf: Bag>
    <s: estudantes>Cassia Barreto</s:Creator>
  </rdf: Description>
</rdf: RDF>
```

5.5 – ESQUEMA RDF

5.5.1 – Sistema Inicial de Tipos do Esquema RDF

O Esquema RDF (RDF SCHEMA, 1998) provê uma mecanismo para a declaração de propriedades de recursos (título, autor, etc.), dos relacionamento entre essas

propriedades, das classes de recursos que podem ser descritas (livros, pessoas, páginas Web, etc.), das combinações possíveis entre classes e propriedades, etc. Basicamente, o esquema RDF provê um sistema de tipos básicos que é empregado na construção dos esquemas RDF.

O sistema de tipos é especificado nos termos do modelo de dados básico da RDF: como *recursos* e *propriedades*, conforme representado na Figura 5.17. As setas indicam que o objeto da qual elas partem possui o tipo do objeto que está sendo apontado. Como o esquema RDF é baseado na proposta MCF, este será descrito de forma sucinta a seguir:

- **rdfs: Resource** – representa qualquer objeto descrito por expressões RDF;
- **rdf: Property** – é um tipo de recurso que representa uma propriedade qualquer de um recurso;
- **rdf: Class** – é um tipo de recurso que representa o conceito de classe de objetos, como nos sistemas orientados a objeto;
- **rdf: type** – é um tipo de propriedade que declara que um recurso é membro de uma classe, possuindo todas as suas características. Um recurso no modelo RDF pode ser uma instância de mais de uma classe;

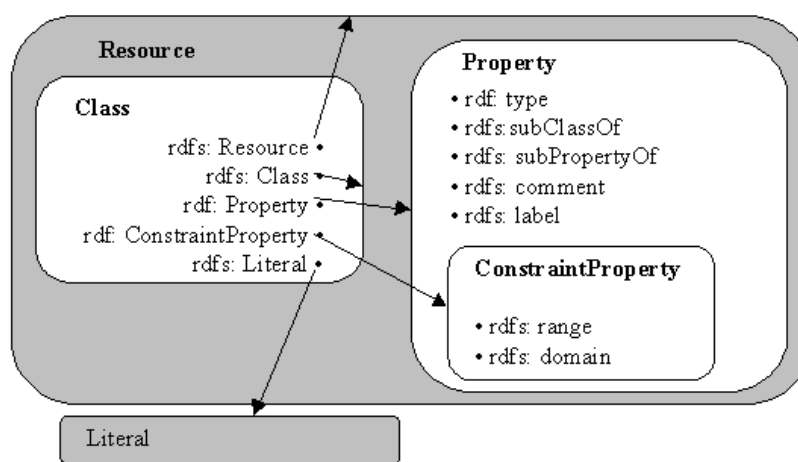


FIGURA 5.17 - Hierarquia de Classes do Esquema RDF

- **rdfs: subClassOf** – é um tipo de propriedade que declara a relação de sub-classe / super-classe entre duas classes. Uma classe pode ser sub-classe de mais de uma classe no modelo RDF;
- **rdfs: subPropertyOf** – é uma propriedade usada para declarar um relacionamento de especialização entre duas propriedades;

- **rdfs: ConstraintResource** – é uma sub-classe de Resource. O propósito desta classe é prover um mecanismo que possibilite ao processador RDF determinar as demais classes definidas que estabelecem regras de restrições;
- **rdfs:ConstraintProperty** – é um recurso sub-classe de rdfs:property. Todas as instâncias desta classe são usadas para especificar restrições. É também sub-classe de rdfs:ConstraintResource, correspondendo a um sub-conjunto que representa propriedades;
- **rdfs:range**: é uma instância da classe rdfs:ConstraintProperty que é usada para restringir valores de propriedades. O valor da propriedade *range* é sempre uma classe; e
- **rdfs:domain** – é uma instância da classe rdfs:ConstraintProperty usada para especificar a classe na qual uma determinada propriedade pode ser aplicada.

A seguir, é apresentado um exemplo de utilização do esquema RDF para modelar o relacionamento entre três classes: “Automóvel” (a mais genérica) e as sub-classes “Caminhão” e “VeículoDePassageiro”.

```
<rdf: RDF
  xmlns: rdf="http://www.w3.org/TR/WD-rdf-syntax#"
  xmlns: rdfs="http://www.w3.org/TR/WD-rdf-schema#" ">

<rdf: Description ID="Automóvel">
  <rdf:type resource="http://www.w3.org/TR/WD-rdf-schema#Class/>
  <rdf: subClassOf resource="http://www.w3.org/TR/WD-rdf-schema#Class/>
</rdf: Description ID="Automóvel">

<rdf: Description ID="VeículoDePassageiro">
  <rdf:type resource="http://www.w3.org/TR/WD-rdf-schema#Class/>
  <rdf: subClassOf resource="#Automóvel"/>
</rdf: Description>

<rdf: Description ID="Caminhão">
  <rdf:type resource="http://www.w3.org/TR/WD-rdf-schema#Class/>
  <rdf: subClassOf resource="#Automóvel"/>
</rdf: Description>

</rdf: RDF>
```

5.6 – ARQUITETURA DE MODELAGEM DE QUATRO NÍVEIS

A arquitetura de modelagem de quatro níveis proposta por Kerhervé (KERHERVÉ, 1997) é baseada na constatação de que a tendência atual conduz à integração, federação e inter-operação de diversas propostas e padrões de metadados existentes.

Segundo Kerhervé, se o objetivo é a extensibilidade do modelo de metadados, deve-se concentrar esforços em um nível de modelagem superior, ou seja nos conceitos usados para se definir esse modelo. Para tal, propõe uma arquitetura de meta-modelagem de quatro níveis que é apresentada na Figura 5.18 e descrita a seguir:

- **nível de dados:** corresponde às instâncias manipuladas pelo sistema. No modelo relacional, por exemplo, tuplas são exemplos de instâncias dos esquemas definidos no nível de modelo. O metadado extraído neste nível se refere a aspectos físicos dos dados, tais como o seu volume e localização do conteúdo;
- **nível de modelo:** se refere à descrição dos elementos usados nas aplicações dos usuários que constituem um modelo da informação do nível imediatamente abaixo. O metadado extraído neste nível se refere a estrutura (ou esquema) desses elementos;
- **nível de metamodelo:** define o formalismo e a especificidade empregados na definição

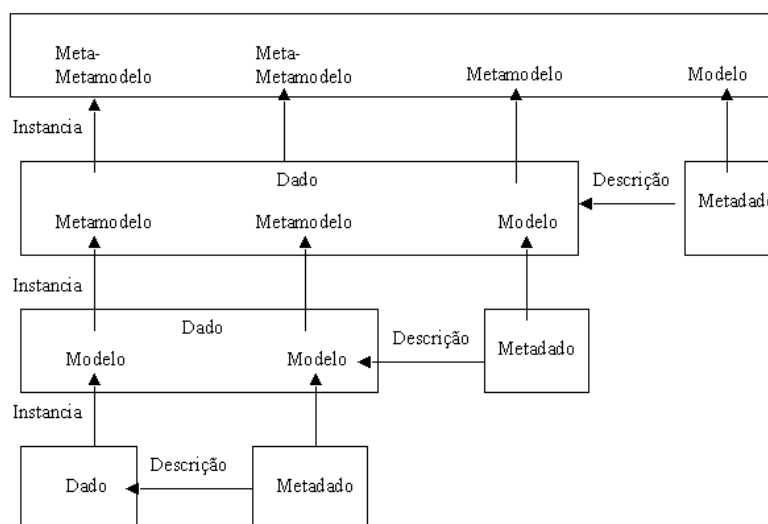


FIGURA 5.18 - Arquitetura de Modelagem de Quatro Níveis

dos modelos da camada imediatamente abaixo. Nos sistemas relacionais, por exemplo, se refere aos conceitos usados para implementar o próprio modelo relacional. O metadado extraído neste nível pode ser empregado para prover interoperabilidade entre ferramentas (CASE, por exemplo), métodos e sistemas; e

- **nível de meta-metamodelo:** é o nível de modelagem raiz, o qual possibilita a representação homogênea e a interoperabilidade nos níveis inferiores.

Kerhervé argumenta também sobre a necessidade de se escolher um formalismo de representação que possibilite expressar, de forma homogênea, a informação nos diferentes níveis de modelagem. Para tal, sugere a utilização de *grafos conceituais* devido a expressividade na representação e na manipulação de tipos e instâncias. Grafos conceituais modelam o universo de discurso como uma série de conceitos e de relações entre esses conceitos.

5.7 – CONSIDERAÇÕES FINAIS

Neste capítulo foram descritas as principais arquiteturas de metadados que estão sendo propostas para implementação na Web. Diversos requisitos são comuns a essas arquiteturas, dentre os quais:

- possibilitar aos projetistas de padrões de metadados enfocar apenas os requisitos específicos de suas áreas de conhecimento;
- permitir que a sintaxe do metadado varie conforme requisições semânticas, práticas de uma determinada comunidade ou requisições funcionais relacionadas ao padrão em questão;
- possibilitar a distribuição da responsabilidade pela gerência do metadado segundo sua comunidade de especialistas;
- promover interoperabilidade semântica e extensibilidade de modo que as ferramentas de pesquisa possam acessar e manipular metadados de forma seletiva;
- permitir controlar, de forma independente, o acesso a conjuntos de metadados distintos que se referem a um mesmo objeto; e
- acomodar, de forma flexível, novos conjuntos de metadados, sem exigir mudanças nos metadados existentes, nem nos programas que os utilizam.

De uma forma genérica, essas arquiteturas enfocam pelos menos um aspecto relacionado à interoperabilidade do metadado, conforme pode ser observado a seguir:

- **Arquitetura Warwick:** seu enfoque é a definição de estruturas específicas para o armazenamento e transporte modularizado de metadados oriundos de padrões distintos. A arquitetura não especifica como devem ser estruturadas as informações contidas nos

pacotes e/ou recipientes, nem uma sintaxe padrão para o seu intercâmbio. Como nenhum mecanismo foi especificado para registro dos metadados contidos nos pacotes, também não foi definida a forma de tratamento das sobreposições e interações semânticas que podem ocorrer entre conjuntos distintos de metadados dentro dos pacotes. Outro aspecto não definido pela arquitetura Warwick se refere a uma taxonomia padrão para descrição dos relacionamentos entre os pacotes contidos no recipiente, provavelmente pela dependência quanto ao domínio de aplicação;

- **Arquiteturas MCF:** seu enfoque é a representação da estrutura de organização da informação. Para tal, utiliza um modelo de dados baseado em grafos, adequado a representação dos relacionamentos simultâneos que podem ocorrer entre os recursos de informação da Web. A arquitetura MCF provê um sistema inicial de tipos que se constitui de um vocabulário básico a partir do qual é possível descrever a estrutura de diversas classes de recursos de informação. A sintaxe proposta pela MCF para o seu modelo de dados é baseada na linguagem XML, possibilitando representar, de forma isomorfa, a definição de um esquema e as instâncias obtidas a partir desse esquema pois, ao contrário da arquitetura RDF, a MCF não emprega modelos distintos para a representação de dados e metadados. A arquitetura utiliza um mecanismo para registro de metadados denominado “espaço de nomes da XML” (XML NAMES SPACE), desenvolvido como parte da linguagem XML. Esse mecanismo possibilita a determinação da semântica de qualquer elemento contido em um bloco XML desde que este esteja qualificado por um prefixo XML válido (DC:creator, por exemplo, identificando o elemento *creator* do padrão Dublin Core). O prefixo em questão é associado a um endereço válido da Web que armazena o esquema a partir do qual obtém-se a descrição semântica do elemento qualificado. Essa abordagem é flexível o bastante para possibilitar o compartilhamento de metadados criados de forma autônoma e descentralizada;
- **Arquitetura RDF:** o propósito da arquitetura é prover uma infra-estrutura sintática e estrutural comum a uma grande diversidade de metadados de forma a maximizar a interoperabilidade. Para a representação da estrutura de organização da informação a arquitetura também utiliza um modelo de dados baseado em grafos. Por outro lado, RDF é uma iniciativa desenvolvida em duas etapas: a primeira etapa enfoca o desenvolvimento de um modelo para a representação das propriedades de recursos e de

seus valores (ou seja, das instâncias de um esquema RDF); a segunda etapa, denominada esquema RDF, enfoca o desenvolvimento de um sistema inicial de tipos, baseado em alguns conceitos providos pela MCF, para descrever a estrutura de organização da informação. Embora adote a abordagem modularizada proposta pela arquitetura Warwick, a RDF representa uma evolução desse modelo, possibilitando o compartilhamento de vocabulários distintos dentro de um mesmo pacote. A descrição semântica dos elementos referenciados no esquema é obtida de maneira semelhante a MCF, ou seja, através do registro em um espaço de nomes da XML; e

- **Arquitetura de modelagem de quatro níveis:** a arquitetura proposta por Kerhervé provê uma base conceitual para a integração de modelos formais de metadados segundo uma hierarquia de descrição onde os mais genéricos, tais como os modelos MCF e RDF, possibilitam a definição de modelos mais específicos tais como Dublin Core ou EAD.

De uma forma genérica, as arquiteturas de metadados estudadas neste capítulo podem ser enfocadas como abordagens de metadados complementares, uma vez que:

- **possibilitam a representação de um recurso de informação segundo o conteúdo intelectual:** a estrutura do objeto digital proposto pela arquitetura de Kahn e Wilensky e expresso segundo a arquitetura Warwick provê um modelo de dados genérico para a representação de diversos tipos de recursos de informação, independentemente de sua complexidade estrutural. Um objeto digital representa um recurso de informação como uma unidade de informação auto-descritiva, onde dados e metadados são associados segundo um contexto de uso, dentro de estruturas denominadas pacotes;
- **possibilitam a representação das propriedades e dos relacionamentos de um recurso de informação:** as arquiteturas RDF e MCF fornecem um modelo de dados concreto para a representação dos dados e metadados armazenados nos pacotes que compõem o objeto digital; e
- **possibilitam o compartilhamento de diversas iniciativas de metadados para a descrição das diversas características de um recurso de informação:** a arquitetura de modelagem multi-nível proposta por Kerhervé possibilita a integração de modelos formais de metadados segundo uma hierarquia de descrição.

. No próximo capítulo, esses aspectos serão melhor detalhados com a apresentação de uma proposta de modelo de metadados para a representação de documentos eletrônicos

voltada para o ambiente da Web, levando em consideração alguns dos conceitos estudados nas arquiteturas de metadados descritas neste capítulo.

CAPÍTULO 6

PROPOSTA DE UM MODELO DE METADADOS PARA A DESCRIÇÃO DE DOCUMENTOS ELETRÔNICOS NA WEB

6.1 – INTRODUÇÃO

Até pouco tempo, a produção de metadados era tarefa exclusiva de alguns especialistas, tais como os bibliotecários e gerentes de sítios da Web. Entretanto, com a explosão de informações na Web, a tendência atual é que pessoas *não especialistas* passem a fornecer metadados a respeito de seus próprios recursos de informação. Para que essa tarefa seja facilitada, é necessário o desenvolvimento de ferramentas para a criação, gerência e uso de metadados.

De uma forma genérica, observa-se que o enfoque da maioria das arquiteturas de metadados analisadas no capítulo anterior baseia-se na definição de uma representação comum para uma ampla variedade de recursos de informação e de suas propriedades. No entanto, pouca atenção é dada ao fato de que cada recurso possui características muito particulares que devem ser consideradas quando se deseja prover uma representação adequada da informação dentro de um determinado contexto. A iniciativa de metadados descrita neste capítulo tem por objetivo cobrir esta lacuna, na medida em que propõe um modelo específico para a representação das principais características de um documento eletrônico na Web. Esse modelo pode ser empregado como base para a construção de ferramentas que auxiliem o usuário leigo na descrição das propriedades de um documento ou de uma coleção de documentos com base em seu conteúdo intelectual.

Na proposta descrita neste capítulo, os seguintes aspectos relacionados a documentos eletrônicos na Web são considerados:

- *organização externa*: se refere à disposição do documento em relação a outros itens com os quais compartilhe de alguma similaridade;
- *conteúdo intelectual*: expressa a semântica do conteúdo do documento e o contexto no qual este se insere;
- *estrutura*: se refere às formas utilizadas para se segmentar um documento em componentes ou unidades menores que podem ser independentes ou não do documento como um todo;
- *organização interna*: se refere à disposição interna das unidades componentes de um documento que define uma hierarquia particular para cada tipo de documento;

- *formato*: expressa as regras definidas para a apresentação da organização interna de um documento; e
- *relacionamentos*: expressam as associações do documento com outros recursos por ele referenciados ou que o referenciam.

Neste capítulo, é apresentado um modelo multi-camadas para a representação de um documento eletrônico na Web. Este modelo foi proposto com base nos aspectos relacionados anteriormente, tratados de forma ampla na Seção 6.2. O modelo tem por objetivo prover uma estrutura organizacional comum para os dados e metadados relacionados a um documento (ou a uma coleção de documentos), tendo como base a estrutura de um objeto digital segundo a arquitetura Warwick, conforme apresentado no Capítulo 5. Foi denominado *Modelo de Objetos Digitais para Documentos Eletrônicos (MODDE)*.

Por fim, na Seção 6.5, é apresentado um estudo de caso que objetiva demonstrar a funcionalidade do modelo de objetos digitais para a organização das descrições relativas a um único documento ou a uma coleção de documentos na Web.

6.2 – ASPECTOS CONSIDERADOS

Ao contrário dos sistemas de catalogação tradicional, que enfocam basicamente as características físicas dos itens disponíveis a mão, os usuários de recursos acessados remotamente, a despeito dos diversos formatos disponíveis na Web, estão mais interessados na identificação e localização de um trabalho, ou mais especificamente, no seu conteúdo intelectual, através do qual o autor documentou suas idéias e compartilhou suas criações artísticas e intelectuais.

Tomando como base essas necessidades, a IFLA (*“International Federation Library Association”*) (TILLET, 1996) (IFLA) vem elaborando alguns requisitos funcionais que devem ser atendidos por um registro bibliográfico em função da evolução da tecnologia de publicação para o ambiente da Web. Esses requisitos estão sendo delineados para os seguintes tipos de entidades abstratas que foram identificados para a maioria dos modelos de catalogação bibliográfica, mesmo aqueles desenvolvidos de forma independente:

- *trabalho*, correspondendo a uma abstração do documento que expressa o seu conteúdo intelectual e artístico;
- *expressão de conteúdo*, correspondendo a uma abstração do documento na qual conteúdo intelectual ou artístico de um trabalho é realizada mediante uma forma textual, visual, musical, etc.;
- *personificação física*, correspondendo a uma abstração que se aplica a todas as instâncias do documento que compartilham das mesmas características físicas; e
- *item*, correspondendo a uma cópia individual do trabalho a qual se aplicam as abstrações relacionadas acima.

Um documento descrito segundo as entidades identificadas pela IFLA é apresentado na Figura 6.1. Nesse exemplo, o conteúdo intelectual, expressão de conteúdo, personificação e os componentes estruturais do documento identificado por “<http://www.dpmm.org/meta.html>” são descritos segundo os termos do padrão Dublin Core. Pode-se observar que conjuntos de termos distintos são necessários para caracterizar de forma mais adequada as abstrações do documento.

Segundo Bearman (BEARMAN et alii, 1999), essas entidades abstratas provêm um *modelo lógico comum* para o emprego de metadados na Web. O principal argumento a seu favor, no entanto, é a observação de que mais importante do que a habilidade em se

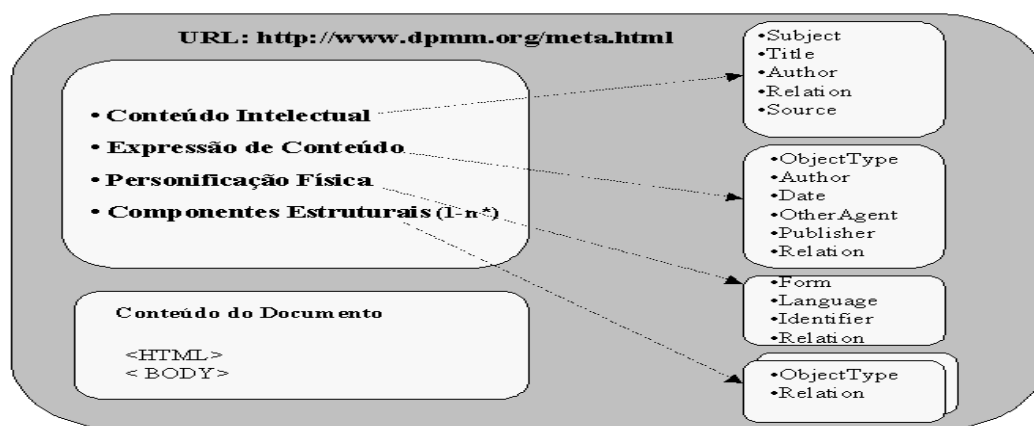


FIGURA 6.1 - Documento Eletrônico segundo a Estrutura Conceitual Proposta

gerar e manipular quantidades cada vez maiores de metadados é a habilidade em organizá-los de forma coerente.

A estrutura organizacional apresentada neste capítulo representa uma extensão do modelo proposto pela IFLA, conforme poderá ser observado nas subseções a seguir.

6.2.1 – Organização Externa

A organização externa de um documento torna explícita a forma como este é agrupado (ou relacionado) com outros itens com o qual compartilha de alguma similaridade. Esses agrupamentos são denominados, de uma forma genérica, de coleções, e objetivam prover uma organização coerente sobre os dados de forma a proporcionar visões que possibilitem uma melhor navegação por esse componentes.

Em sistemas tradicionais de bibliotecas, coleções fornecem visões estáticas de seus títulos segundo organizações bem familiares (tais como: autor, assunto ou título), utilizando-se de algum esquema de classificação. De uma forma geral, essas organizações facilitam a consulta pois diminuem a complexidade de seu manuseio ao agrupar itens similares. Em ambientes como a Web, um único documento pode fazer parte, simultaneamente, de várias coleções. A organização de uma coleção pode também ser alterada e exibida de forma dinâmica ao usuário, a fim de que este possa observá-la sob pontos de vista distintos. Para coleções organizadas pelo endereço (URL) dos documentos, por exemplo, é possível agrupar os resultados que se referem ao mesmo sítio ou então, agrupar os endereços que se referem à mesma coleção em um sítio.

6.2.2 – Conteúdo Intelectual

Segundo Tillet (TILLET, 1996), o conteúdo intelectual de um documento expressa as idéias e opiniões de seu autor a respeito de algum assunto. O conteúdo intelectual caracteriza a “obra” ou o trabalho em si e independe do gênero de recurso, mídia e formato de publicação. O título “Modelo de Metadados para a Descrição de Documentos Eletrônicos na Web”, por exemplo, pode ser empregado para caracterizar o conteúdo intelectual desta dissertação. É o nível mais alto de abstração que pode ser empregado para caracterizar um documento.

6.2.3 – Estrutura

A despeito da estrutura de hiperligações decorrente do próprio ambiente da Web, o conteúdo textual de cada unidade componente do documento possui estrutura, ou seja, não é uma seqüência indiferenciada de palavras e muito menos, de bytes. Dependendo do

propósito, um documento pode ser dividido em unidades distintas, de tipos e tamanhos também distintos. Um texto em prosa como esta dissertação, por exemplo, pode ser dividido em seções, capítulos, parágrafos e sentenças. Um verso, no entanto, poderia ser dividido em cantos, estrofes e linhas.

Uma vez impressas, essas unidades podem ser divididas em volumes e páginas, por exemplo. De uma forma geral, unidades estruturais são usadas para identificar localidades ou pontos específicos dentro de um texto: por exemplo, “a terceira sentença do segundo parágrafo no capítulo décimo” ou “a página 10”. Também podem ser empregadas para dividir um texto em fragmentos significativos para fins analíticos: “quantos parágrafos separam cada ocorrência da palavra *metadado* ? quantas páginas ?. Unidades textuais podem se sobrepor de maneira complexa e imprevisível. Em um documento XML, por exemplo, as estruturas lógicas são indicadas pela marcação que ele contém, enquanto que as estruturas físicas podem representar unidades físicas de armazenagem, tal como um arquivo em disco. Entretanto, as estruturas lógicas e físicas em um documento XML devem ser síncronas, ou seja, precisam se aninhar. Elementos precisam começar e terminar em uma mesma entidade. Cada entidade deve conter estruturas lógicas completas e não parciais, conforme apresentado na Figura 6.2.

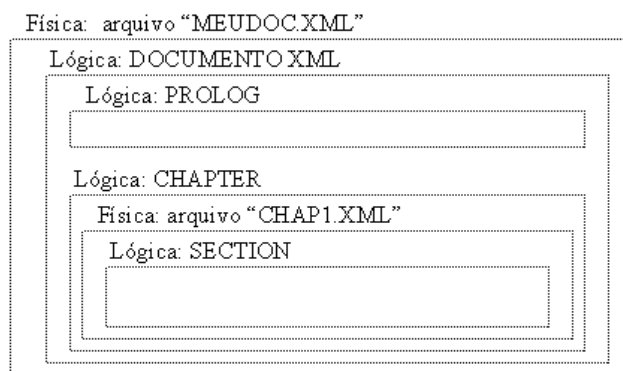


FIGURA 6.2 - Estruturas Lógicas e Físicas em um Documento XML

Na Web, unidades estruturais podem ser utilizadas para: auxiliar na determinação do tamanho do documento; melhorar a indexação - termos de diferentes partes de um documento podem ser pesados de acordo com a sua importância no documento (resumo e notas de rodapé, por exemplo); possibilitar o acesso direto a partes específicas de um documento; distinguir níveis de detalhes diferentes; restringir a recuperação a partes específicas do documento ou a determinado nível de detalhe; efetuar pesquisas combinando tipos específicos de mídias, etc. (FERBER, 1997).

6.2.4 – Organização Interna

A organização interna de um documento caracteriza o seu *tipo*. Formalmente, o tipo de um documento define sua estrutura interna e organização, ou seja, define quais unidades estruturais um documento pode conter e como essas unidades são organizadas internamente (SGML, 1996). Segundo Tillet (TILLET, 1996), o tipo de um documento representa uma *expressão de conteúdo* do documento.

A organização de documentos em tipos é geralmente usada para fins de verificação quanto a sua validade estrutural. Um documento do tipo *relatório técnico*, por exemplo, pode ser definido como “um documento composto por um título, seguido de um resumo e de uma seqüência de um ou mais parágrafos”. Neste caso, qualquer outro tipo de documento que não possuir um título não será considerado um relatório técnico como, por exemplo, “um documento contendo uma seqüência de parágrafos seguido por um resumo”.

Na Web, o padrão BIBTEX (BIBTEX) provê uma taxonomia simples para a organização de documentos em tipos. BIBTEX é um programa e um formato de arquivo projetado em 1985 para o sistema de preparação de documentos (LATEX). A taxonomia BIBTEX especifica os tipos *article*, *book*, *booklet*, *inBook*, *inCollection*, *inProceeding*, *manual*, *MasterThesis*, *PhDThesis*, *proceedings*, *techReport* e *unpublished* e atributos tais como *author*, *title*, *volume*, *number*, etc. que, quando combinados, formam um tipo de recurso. BIBTEX define, por exemplo, que um documento do tipo “tese de mestrado” deve conter informação sobre a universidade na qual foi conduzida, conforme apresentado no exemplo a seguir:

```
@PhdThesis {La88,  
  author = “K. Larsen”  
  year   = “1988”,  
  type   = “Master Thesis”  
  school = “Aarhus Univerty, Denmark”  
  title  = “A Fully Abstract Model for a Process Algebra with Refinements”}.
```

Para documentos baseados no padrão SGML, uma DTD (“*Document Type Definition*”) define o esquema do documento, especificando os tipos e as características (conteúdo, atributos, etc.) de cada elemento contido no documento e as notações e entidades que podem ser encontradas nesse documento.

6.2.5 – Formato

A Web é, essencialmente, um ambiente de publicação sendo, portanto, grande o número de formatos para a disponibilidade de um recurso de informação. O padrão MIME, por exemplo, organiza recursos de informação quanto ao formato segundo uma hierarquia de dois níveis, onde o primeiro agrupa recursos por tipo de mídia distintos (texto, imagem, aplicação, etc.) e o segundo os formatos possíveis para cada mídia (jpeg, html, postScript, etc.). Segundo Tillet (TILLET, 1996), os vários formatos de publicação de um documento caracterizam suas *personificações físicas*, cada qual representando uma forma usada para disseminar o seu conteúdo.

6.2.6 – Relacionamentos com Outros Recursos

No modelo proposto duas categorias de relacionamentos entre recursos serão consideradas: *estruturais* e *contextuais*. Relacionamentos estruturais são essencialmente hierárquicos, ou seja, ocorrem entre recursos de informação organizados segundo uma estrutura hierárquica qualquer, e por esse motivo também podem ser denominados *relacionamentos verticais* (USMARC, 1996). Neste modelo, relacionamentos estruturais podem ser empregados para representar a associação entre: uma coleção e os conteúdos intelectuais e/ou sub-coleções que a compõem; um determinado conteúdo intelectual e as várias formas de expressão desse conteúdo; uma determinada forma de expressão de conteúdo e suas várias personificações físicas; e uma determinada personificação física e suas várias unidades estruturais.

Relacionamentos contextuais, por sua vez, representam a associação do documento com outros recursos complementares tais como: suas anotações; referências bibliográficas; registros de catalogação; versões distintas do conteúdo intelectual; cópias de uma determinada personificação física; termos e condições para acesso a seu conteúdo, etc. Também podem ser chamados de *relacionamentos horizontais* (USMARC, 1996), pois permitem relacionar recursos independentemente da forma como esses estão organizados. A descrição dos relacionamentos dessa categoria ainda depende da definição de uma taxonomia que está sendo investigada pelos grupos ligados a série de *workshops* em metadados para a Web.

6.3 – PROPOSTA DE UMA ESTRUTURA ORGANIZACIONAL PARA DOCUMENTOS ELETRÔNICOS NO AMBIENTE DA WEB

A fim de poder representar os principais aspectos de um documento eletrônico na Web, uma estrutura multi-camadas será proposta, uma vez que, intuitivamente, os conceitos de conteúdo intelectual, organização externa, interna e a própria estrutura de um documento na Web conduzem a uma representação hierárquica.

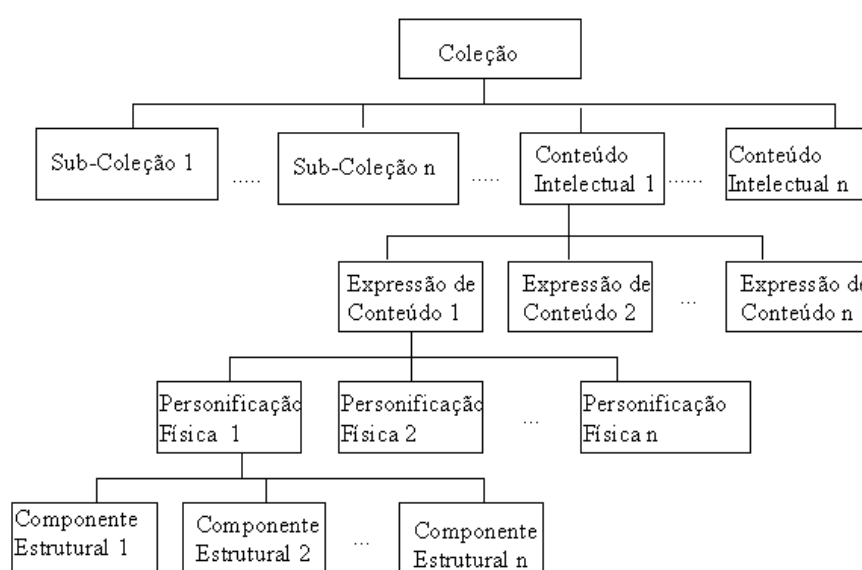


FIGURA 6.3: Estrutura organizacional para um Documento Eletrônico na Web

De acordo a estrutura proposta, um documento pode ser visualizado como um conjunto de conceitos inter-relacionados organizados segundo uma hierarquia de quatro camadas, apresentada na Figura 6.3, onde categorias distintas de metadados são necessárias para descrever os conceitos da cada camada. De uma forma geral, metadados definidos em uma camada são herdados pelas camadas seguintes.

- **Camada de coleção:** expressa a organização externa de um documento. Nesta camada, documentos podem ser organizados segundo esquemas padrões de classificação ou formas menos comuns de agrupamento. Metadados para descrever a coleção podem enfatizar: a descrição de seu conteúdo, assunto, total de itens disponíveis, título (ou nome) da coleção, seus responsáveis, etc. Quanto aos relacionamentos contextuais, para cada coleção pode haver outra superior, subordinada, derivada, equivalente, complementar, similar, precedente, sucessora, etc.;

- **Camada de conteúdo intelectual:** nesta camada, um documento é representado pelas idéias e conceitos de seus autores que são expressas em seu conteúdo. O metadado necessário para descrevê-lo constitui-se de informações bibliográficas básicas (tais como título, autores, outros contribuintes, etc.), da descrição de seu conteúdo intelectual, do assunto, etc. Quanto aos relacionamentos contextuais, um documento representado nesta camada pode referenciar: suas referências bibliográficas, outros documentos complementares ou de alguma forma similares, versões anteriores do mesmo conteúdo intelectual, documentos ou recursos do qual o seu conteúdo seja derivado, anotações, etc.;
- **Camada de expressão de conteúdo:** nesta camada o conteúdo intelectual de um documento é organizado segundo uma ou mais formas de expressões desse conteúdo. Uma expressão de conteúdo define um tipo de documento, tal como “tese de mestrado”, “relatório técnico”, “tese de doutorado”, etc. O metadado para descrever esse tipo pode focar na descrição de sua estrutura e organização (fazendo uso de um esquema, por exemplo), nos aspectos da publicação (local, data, publicadores, outros contribuintes, etc.), na descrição do contexto de preparação do documento, etc. Quanto aos relacionamentos contextuais, nesta camada podem focar outras versões daquela expressão de conteúdo, termos e condições para acesso ao conteúdo do documento, etc.;
- **Camada de personificação física:** nesta camada, uma determinada expressão de conteúdo de um documento pode ser materializada por uma ou mais personificações físicas (ou instâncias) dessa expressão. Uma personificação física representa o conteúdo do documento segundo um determinado formato de publicação da Web (postScript e HTML, por exemplo). O metadado para descrevê-lo pode focar: o tipo da mídia; formato; endereço (URL) do documento; idioma no qual o seu conteúdo foi expresso; modo de acesso; custos envolvidos na recuperação; detalhes da distribuição, etc. Quanto aos relacionamentos contextuais, podem focar: as traduções do documento para outros idiomas; edições suplementares; cópias disponíveis em outros servidores, etc.;
- **Camada de estrutura:** na última camada do modelo proposto, uma determinada personificação física de um documento pode ser visualizada como um conjunto de *visões estruturais* (RUS & SUBRAMANIAN, 1997), cada qual representando uma

forma pela qual essa personificação pode ser segmentada para exibição ou pesquisa. São quatro as visões estruturais definidas para o modelo proposto e descritas a seguir:

- ❑ **Visão física ou de representação:** representa a segmentação de uma instância do documento em partes físicas tais como páginas, blocos, quadros ou mesmo coordenadas físicas (GOODCHILD, 1996);
- ❑ **Visão lógica:** representa a segmentação de uma instância do documento em partes lógicas (tais como cabeçalhos, parágrafos, seções, etc.), cada qual correspondendo a um componente semântico distinto, organizadas segundo uma hierarquia previamente definida;
- ❑ **Visão de conteúdo:** representa abstrações ou conceitos extraídos do conteúdo do documento segundo um determinado domínio de aplicação. Em um trabalho científico, por exemplo, *teoremas, lemas e provas* são exemplos de abstrações baseadas em conteúdo que podem ser extraídas do documento; e
- ❑ **Visão hipermídia:** representa a segmentação de um documento por seus componentes hipermídia tais como textos, imagens, vídeos, etc. Esta visão descreve a semântica das hiperligações de um documento na Web.

O modelo de documento multi-camadas proposto nesta seção provê a estrutura necessária para a gerência de um documento (ou de uma coleção de documentos) com base em seu conteúdo intelectual. Para que a sua implementação seja viável, considerando o grande acervo disponível na Web, a estrutura de dados selecionada para representá-lo deve possibilitar a organização de um documento (e de seus componentes) segundo o modelo conceitual proposto, independentemente de sua localização e forma de armazenamento. Nesse contexto, um documento pode ser visualizado como uma entidade de alto nível, cujo identificador é único e independente de localização, conforme apresentado na Figura 6.4.

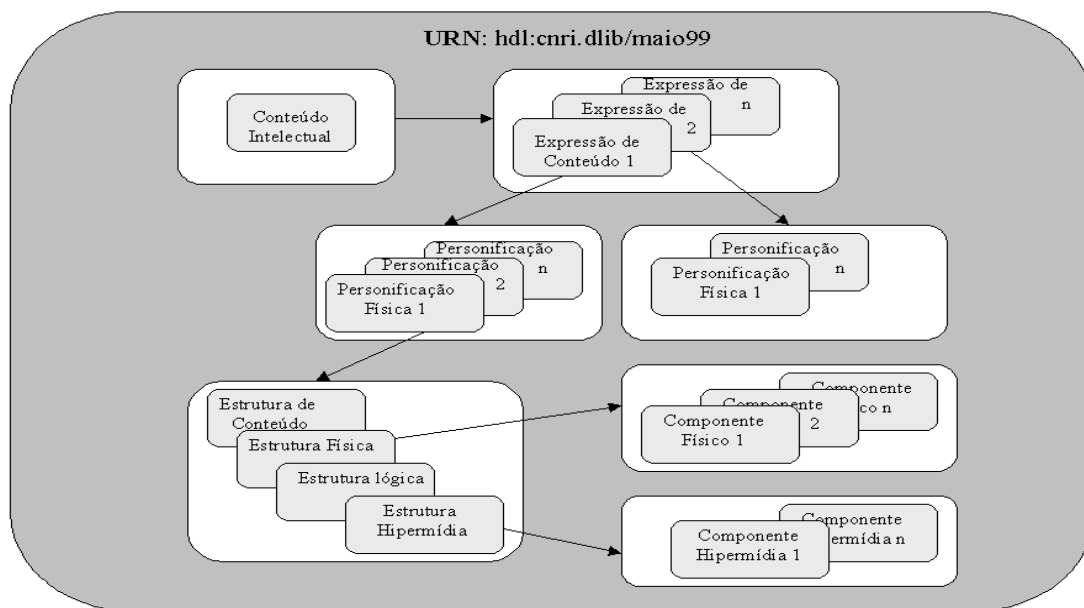


FIGURA 6.4 - Documento Eletrônico Multi-camadas

6.4 – MODELO DE OBJETOS DIGITAIS PARA DOCUMENTOS ELETRÔNICOS (MODDE)

Segundo Lagoze (LAGOZE, 1995), uma solução que possibilitaria a gerência de recursos com base no conteúdo intelectual seria o emprego de uma estrutura abstrata que escondesse dos usuários os detalhes da estrutura física dos arquivos. O objeto digital proposto por Kahn e Wilensky para a gerência de recursos em uma biblioteca digital provê essa estrutura abstrata, possibilitando a representação de um recurso de informação como uma unidade auto-descrita de informação, na qual os metadados necessários para a interpretação adequada do seu conteúdo está contido (ou referenciado) em sua própria estrutura. O objeto digital funciona, portanto, como uma cápsula para dados e metadados que são organizados segundo um contexto bem definido de uso.

A fim de possibilitar a representação da estrutura de diversos recursos de informação, modelos de dados são usados para se atribuir semântica a um objeto digital. No contexto em questão, a estrutura de um objeto digital segundo a arquitetura Warwick será empregada por ser bem adequada à representação de recursos de informação organizados de forma hierárquica.

Na proposta descrita nesta seção, denominada Modelo de Objetos Digitais para Documentos Eletrônicos (MODDE), objetos digitais são organizados segundo uma representação hierárquica recursiva. Nessa hierarquia, com a exceção do ObjetoAssociação, cada tipo de objeto digital representa uma camada ou um componente de uma das camadas da estrutura organizacional proposta na seção anterior.

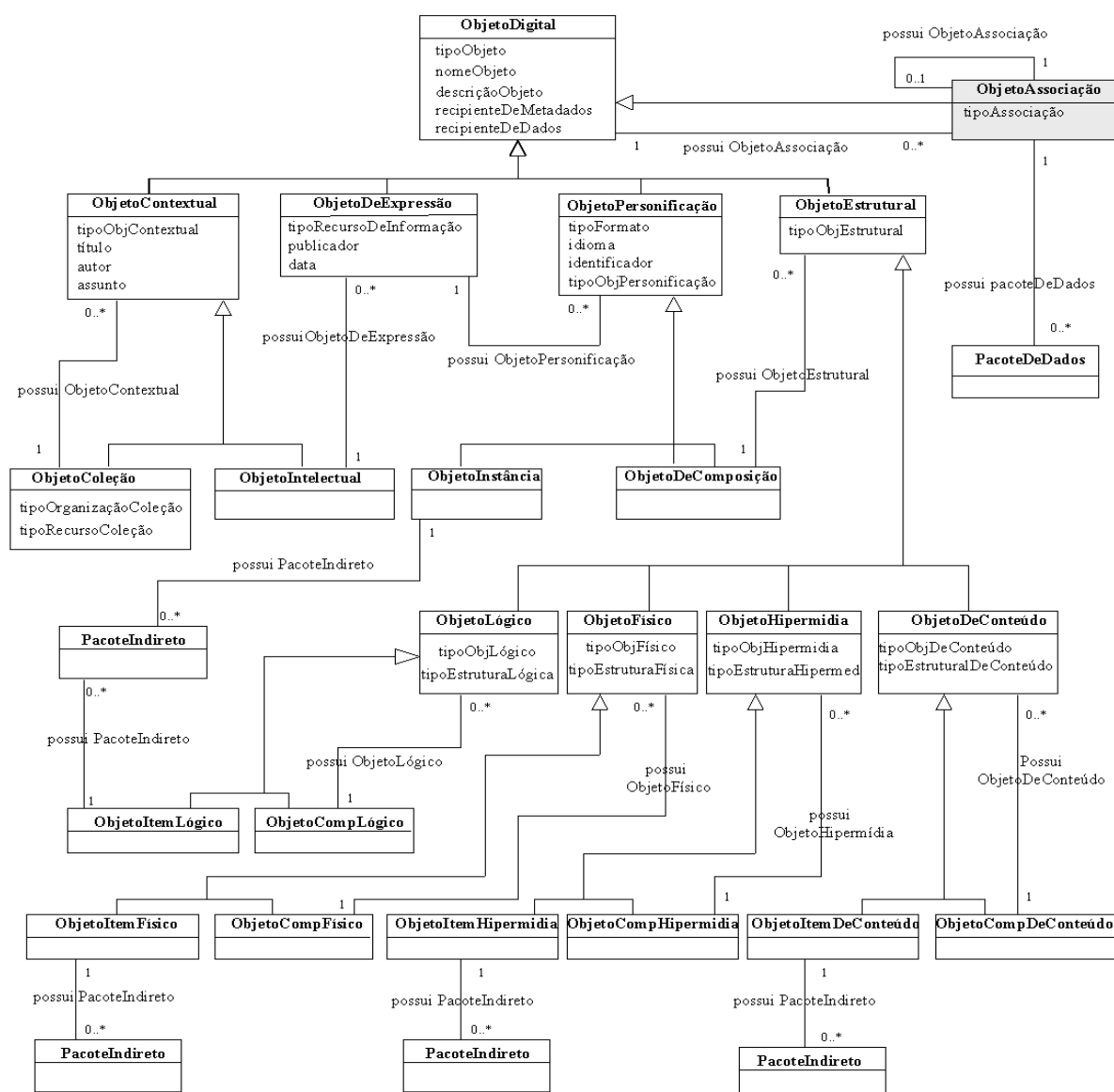


FIGURA 6.5- Tipos de Objetos Digitais

Conforme apresentado na Figura 6.5, a classe ObjetoDigital representa a super-classe de todos os objetos definidos. A estrutura das classes *PacoteDeDados* e *PacoteIndireto* referenciadas por alguns objetos digitais será especificada no Capítulo 7,

que define um ambiente para instanciação e utilização do MODDE. Os atributos que são comuns a todos os objetos digitais da hierarquia proposta são descritos a seguir:

- **tipoObjeto:** descreve o tipo do objeto digital, de acordo com as sub-classes da hierarquia definida;
- **nomeObjeto:** é o atributo empregado para a identificação do objeto digital;
- **descriçãoObjeto:** possibilita uma descrição sucinta do objeto digital;
- **recipienteDeDados:** é o atributo do objeto digital que descreve os relacionamentos estruturais que podem ocorrer entre recursos de informação (do tipo coleção ou documento) representados segundo as camadas da estrutura organizacional proposta na Seção 6.3. Por esse motivo, o valor do atributo recipienteDeDados é distinto para cada tipo de objeto digital; e
- **recipienteDeMetadados:** é o atributo do objeto digital que descreve os relacionamentos contextuais que podem ocorrer entre recursos de informação nas diversas camadas da estrutura organizacional proposta na Seção 6.3. O valor atribuído a este atributo referencia informações adicionais que não fazem parte do conteúdo do recurso representado pelo objeto digital.

Basicamente, são definidos cinco tipos de objetos digitais, cada qual implementando um conjunto de referências a outros tipos de objetos digitais ou a um conjunto de pacotes, conforme descrito a seguir.

- **ObjetoAssociação:** este objeto permite a representação dos relacionamentos contextuais que podem ocorrer entre os recursos de informação da Web. Esses relacionamentos são implementados como objetos de primeira-classe, podendo ter metadados próprios associados e serem executados a partir de qualquer local da Web. Os atributos desse objeto são: *tipoAssociação*, que identifica a categoria de relacionamento contextual que está sendo representada pelo ObjetoAssociação, tais como “Termos e Condições para Acesso”, “Referências Bibliográficas”, “Descrição Bibliográfica”, etc.; *recipienteDeDados*, que referencia um ou mais PacoteDeDados, cada qual contendo metadados (oriundos ou não de diversos padrões) relativos à categoria de relacionamento contextual que está sendo representada pelo ObjetoAssociação; *recipienteDeMetadados*, que referencia, de forma recursiva, um conjunto de objetos digitais também do tipo ObjetoAssociação, através do qual é possível descrever os relacionamentos contextuais do próprio ObjetoAssociação. É

importante observar que as informações contidas dentro de um PacoteDeDados cumprem o papel de “metadado” para o objeto digital que faz referência ao ObjetoAssociação em questão, embora essas informações façam o papel de “dado” para o ObjetoAssociação que as contém ou referencia;

- **ObjetoContextual:** define um contexto bem definido de uso para a informação do usuário. Um objetoColeção pode ser empregado para representar recursos de informação da Web que agrupam coleções de documentos eletrônicos (tal como uma *homepage*) ou então, ser empregado para representar um único documento modelado segundo a estrutura organizacional proposta na Seção 6.3. Os atributos do ObjetoContextual são descritos a seguir: *título*, que identifica o conteúdo intelectual da coleção ou do documento representado pelo ObjetoContextual; *autor(es)*, que identifica(m) o(s) responsável(is) pelo conteúdo intelectual da coleção ou do documento representado pelo ObjetoContextual; *assunto*, que identifica o tópico relacionado à coleção ou ao documento. Pode ser qualificado por um *esquemaDeClassificação* que especifica o vocabulário (Decimal Dewey, por exemplo) que foi empregado para classificar o conteúdo intelectual do documento e/ou da coleção, de modo que estes possam ser agrupados a outros recursos similares. Conforme identificado pelo atributo *tipoObjContextual*, ObjetoContextual é super-classe para os objetos digitais do tipo ObjetoColeção e ObjetoIntelectual, descritos a seguir:
 - **ObjetoColeção:** este objeto possibilita a representação de recursos de informação do tipo coleção. Pode ser empregado para representar, por exemplo, um conjunto de poemas de autores diferentes agrupados por um assunto comum. Seus atributos são descritos a seguir: *tipoOrganizaçãoColeção*, que especifica a forma de organização do conteúdo da coleção (por autor, assunto, período histórico, etc.); *tipoRecursoColeção*, que associa o ObjetoColeção à categoria de recurso de informação da Web que está sendo representado pelo ObjetoColeção (por exemplo, a *homepage* de um sítio, um catálogo “on-line” de recursos por assunto tal como YAHOO!, uma tabela de conteúdo de um sítio, etc.); *recipienteDeDados*, que referencia, de forma recursiva, um conjunto de objetos digitais do tipo ObjetoContextual, o que possibilita a representação dos relacionamentos estruturais que podem existir entre uma coleção e seus itens

(sub-coleções e/ou conteúdos intelectuais distintos); e *recipienteDeMetadados*, que referencia um conjunto de objetos digitais do tipo ObjetoAssociação, para descrever os relacionamentos contextuais desta camada; e

- **ObjetoIntelectual:** este objeto representa um documento segundo o seu conteúdo intelectual. Seus atributos são: *recipienteDeDados*, que referencia um conjunto de objetos do tipo ObjetoDeExpressão, descrito a seguir, cada qual descrevendo os relacionamentos (estruturais) do conteúdo intelectual em questão com suas várias formas de expressão de conteúdo; e *recipienteDeMetadados*, que contém um conjunto de referências a objetos digitais do tipo ObjetoAssociação, que descrevem os relacionamentos contextuais que podem ocorrer nesta camada.
- **ObjetoDeExpressão:** representa um documento na camada de expressão de conteúdo. Seus atributos são: *tipoRecursoDeInformação*, que associa uma categoria ou gênero de documento, tal como “artigo” e “tese de mestrado”, ao ObjetoDeExpressão. Essa categoria pode ser provida por um padrão tal como BIBTEX, por exemplo; *publicador*, que referencia a instituição ou pessoa responsável por tornar disponível a expressão de conteúdo representada pelo ObjetoDeExpressão; *data*, que indica a data de publicação da expressão de conteúdo representada pelo ObjetoDeExpressão; *recipienteDeMetadados*, que referencia um conjunto de objetos do tipo ObjetoAssociação, cada qual descrevendo um relacionamento contextual que pode ocorrer nesta camada; e *recipienteDeDados*, que referencia um conjunto de objetos digitais do tipo ObjetoPersonificação, cada qual descrevendo o relacionamento (estrutural) da expressão de conteúdo representada pelo ObjetoDeExpressão com suas várias personificações físicas;
- **ObjetoPersonificação:** representa um documento na camada de personificação física. Seus atributos são: *tipoFormato*, que referencia o tipo de formato (tal como text/HTML e application/postScript providos pelo padrão MIME) da personificação física representada pelo ObjetoPersonificação; *idioma*, que é relativo ao conteúdo da personificação física representada pelo ObjetoPersonificação; *identificador*, que referencia um conjunto de caracteres empregados para identificar a personificação física representada pelo ObjetoPersonificação; e *recipienteDeMetadados*, que contém um conjunto de referências a objetos digitais do tipo ObjetoAssociação. Esses objetos descrevem os relacionamentos contextuais que podem ocorrer nesta camada.

ObjetoPersonificação é super-classe para os objetos ObjetoInstância e ObjetoDeComposição, descritos a seguir, conforme identificado pelo atributo *tipoObjetoPersonificação*;

- **ObjetoInstância:** representa uma determinada personificação física de um documento como uma unidade (ou item) indivisível de informação, ou seja, sem levar em consideração os componentes estruturais que podem ser obtidos a partir de uma segmentação dessa personificação física. Seu atributo *recipienteDeDados* referencia um conjunto de *PacoteIndireto*, cada qual referenciando, por sua vez, a porção “dado” ou o conteúdo da personificação física representada pelo ObjetoInstância; e
- **ObjetoDeComposição:** representa uma determinada personificação física levando em consideração os componentes estruturais que podem ser obtidos a partir da segmentação de um documento em unidades decorrentes de suas visões estruturais lógica, física, de conteúdo ou hipermídia. Seu atributo *recipienteDeDados* referencia um conjunto de objetos digitais do tipo ObjetoEstrutural, cada qual representado uma visão estrutural da personificação física em questão.
- **ObjetoEstrutural:** representa um documento na camada de estrutura. ObjetoEstrutural possibilita a representação dos vários tipos de segmentação que podem ser aplicados a uma determinada personificação física de um documento. É super-classe de ObjetoFísico, ObjetoLógico, ObjetoDeConteúdo e ObjetoHiperímia, conforme indicado pelo atributo *tipoObjetoEstrutural*. Cada tipo distinto de ObjetoEstrutural representa uma *visão estrutural* do ObjetoInstância que faz referência ao ObjetoEstrutural em questão. Cada tipo de ObjetoEstrutural pode ser do tipo Item ou do tipo Composto. O atributo *recipienteDeDados* de cada objeto digital do tipo Composto referencia, de forma recursiva, um conjunto de objetos digitais do tipo Item ou Composto. Pode ser empregado para representar uma seção contendo vários parágrafos, no caso da representação de uma visão lógica do documento. Um ObjetoEstrutural do tipo Item (ou objeto “folha”), por sua vez, representa um determinado componente decorrente da segmentação do documento como uma unidade de informação estruturalmente indivisível. O atributo *recipienteDeDados* de cada tipo de ObjetoEstrutural do tipo Item referencia um ou mais *PacoteIndireto*, cada qual contendo ou referenciando a porção “dado” (ou de conteúdo) da unidade de informação

representada pelo ObjetoEstrutural. Conforme identificado pelo atributo *tipoObjEstrutural*, ObjetoEstrutural é super-classe dos objetos descritos a seguir:

- ❑ **ObjetoFísico:** é super-classe de ObjetoCompFísico e ObjetoItemFísico. Seu atributo *tipoEstruturalFísico* pode especificar uma unidade estrutural física tal como uma página ou um bloco;
- ❑ **ObjetoLógico:** é super-classe de ObjetoCompLógico e ObjetoItemLógico. Seu atributo *tipoEstruturalLógico* pode especificar uma unidade estrutural lógica tal como uma seção ou um parágrafo;
- ❑ **ObjetoDeConteúdo:** é super-classe de ObjetoCompDeConteúdo e ObjetoItemDeConteúdo. Seu atributo *tipoEstruturalDeConteúdo* pode especificar uma unidade estrutural extraída do conteúdo do documento, tal como uma pessoa ou um carro, por exemplo; e
- ❑ **ObjetoHiperímia:** é super-classe de ObjetoCompHiperímia e ObjetoItemHiperímia. Seu atributo *tipoEstruturalHiperímia* pode especificar uma unidade estrutural hiperímia, tal como uma imagem ou um texto.

6.5 – EXEMPLO DE UTILIZAÇÃO DO MODDE NA WEB

Nesta seção é apresentado um exemplo da funcionalidade do MODDE descrito na Seção 6.4. Inicialmente, é descrito um estudo de caso no qual vários tipos de recursos de informação da Web são interligados segundo diversas categorias de relacionamentos. Na Subseção 6.5.3, o modelo de objetos digitais é empregado para representar e descrever essas associações.

6.5.1 - Estudo de Caso

A Figura 6.6 apresenta vários tipos de recursos de informação conectados por setas. Cada seta indica que o recurso da qual se origina faz referência ao recurso por ela apontado. Os retângulos em negrito representam o conjunto de documentos que se deseja modelar segundo a estrutura organizacional proposta na Seção 6.3. Os demais retângulos representam recursos complementares associados a esses documentos por diversas categorias de relacionamentos, conforme descrito a seguir.

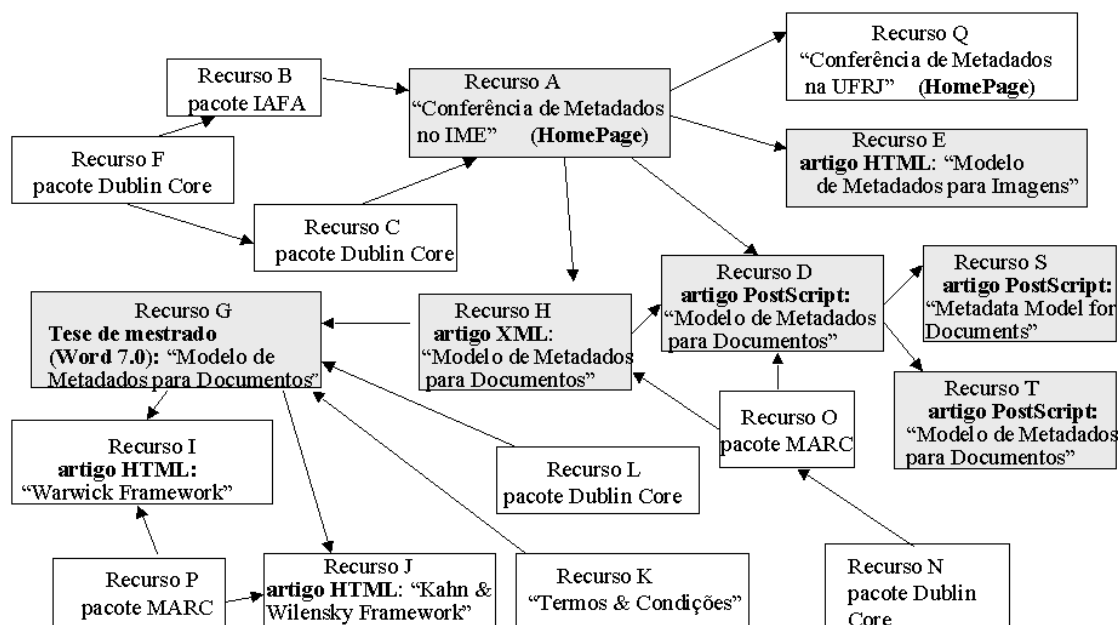


FIGURA 6.6: Exemplo de Relacionamentos entre Recursos da Web

- **Setas partindo do recurso “A”:** na parte superior do exemplo, observa-se o recurso de informação “A”, uma *homepage* intitulada “Conferência de Metadados no IME” através da qual tem-se acesso a coleção de trabalhos sobre metadados desenvolvidos na IME. No exemplo, “A” referencia os recursos “Q”, “E”, “D” e “H”. O recurso “E” representa um artigo em HTML intitulado “Modelo de Metadados para Imagens”. “D” e “H” representam dois artigos, um codificado em postScript e o outro em XML, sendo ambos intitulados “Modelo de Metadados para Documentos”. O recurso Q, por sua vez, é uma *homepage* intitulada “Conferência de Metadados na UFRJ”, que também referencia trabalhos sobre o tema metadados. Nesse contexto, “Q” pode ser considerada uma coleção complementar de “A”;
- **Seta que parte do recurso “B”:** “B” é um pacote contendo descritores do padrão IAFA, que aponta para o recurso “A”. Os descritores contidos em “B” provêm informações bibliográficas (tais como autores, assunto, descrição, etc.) sobre o recurso “A”;
- **Seta que parte do recurso “C”:** “C” é um pacote contendo descritores do padrão Dublin Core que também aponta para o recurso “A”. Tal como o recurso “B”, os descritores contidos em “C” provêm informações bibliográficas adicionais a respeito de “A”;

- **Setas que partem do recurso “F”:** “F” é um pacote contendo metadados do padrão Dublin Core. No exemplo da Figura 6.6, “F” é empregado para fornecer informações administrativas a respeito dos recursos “B” e “C” (tais como, por exemplo, a data de criação e os responsáveis pela manutenção do recurso);
- **Setas que partem do recurso “H”:** “H” é um artigo na linguagem XML que referencia os recursos “D” e “G”, respectivamente, um artigo e uma tese de mestrado. Na Figura 6.6 observa-se que “D”, “G” e “H” compartilham o mesmo conteúdo intelectual, caracterizado pelo título “Modelo de Metadados para Documentos”. Também se observa que os recursos “H” e “D” são artigos expressos segundo formatos distintos de apresentação (XML e postScript) e que “H” e “G”, por sua vez, representam expressões de conteúdo distintas (tese de mestrado e artigo) para o mesmo conteúdo intelectual (“Modelo de Metadados para Documentos”);
- **Setas que partem do recurso “D”:** “D” aponta para os recursos “S” e “T”. O recurso “S” representa uma tradução de “D” para o idioma inglês e “T” uma cópia de “D” residente em um outro sítio qualquer da Web. Nesse exemplo, pode-se considerar, portanto, que o conteúdo do recurso “S” foi *derivado* do conteúdo do recurso “D” e que o conteúdo do recurso “T” é *identico* ao conteúdo de “D”;
- **Setas que partem do recurso “O”:** “O” fornece metadados bibliográficos (padrão MARC) a respeito dos recursos “D” e “H”;
- **Seta que parte do recurso “N”:** “N” é um pacote contendo metadados do padrão Dublin Core que provêem informações administrativas a respeito do recurso “O”;
- **Setas que partem do recurso “G”:** “G” aponta para os recursos “I” e “J”, dois artigos em HTML intitulados, respectivamente, “Warwick Framework” e “Kahn & Wilensky Framework”. Ambos artigos são referências bibliográficas do recurso “G”, entitulado “Modelo de Metadados para Documentos Eletrônicos”;
- **Setas que partem do recurso “P”:** “P” representa um pacote contendo descritores do padrão MARC. Esses descritores provêem informações bibliográficas sobre os recursos “I” e “J”; e
- **Seta que parte do recurso “L”:** “L” é um pacote que contém descritores do padrão Dublin Core. Esses descritores provêem informações bibliográficas sobre o recurso “G”;

- **Seta que parte do recurso “K”:** “K” contém termos e condições para acesso ao conteúdo da tese de mestrado representada pelo recurso “G”.

6.5.2 – Análise do Exemplo

Sem o auxílio de uma ferramenta capaz de guiar o usuário através das diversas camadas do modelo proposto, a modelagem do exemplo descrito anteriormente exige um considerável esforço inicial na identificação das seguintes entidades abstratas: coleções e sub-coleções distintas; conteúdos intelectuais distintos; expressões de conteúdo distintas relativas a cada conteúdo intelectual identificado; personificações físicas distintas relativas a cada expressão de conteúdo identificada; e as unidades estruturais que se deseja representar para cada personificação física identificada. No exemplo descrito acima são identificados:

- **Coleções e sub-coleções:** foi identificada apenas a coleção representada pelo recurso “A” (*homepage* do IME). No exemplo em questão, “Q” representa uma coleção complementar e não uma sub-coleção de “A”;
- **Conteúdos intelectuais:** são identificados os conteúdos intelectuais intitulados “Modelo de Metadados para Imagem” (recurso “E”) e “Modelo de Metadados para Documentos” (recursos “H”, “G”, “D”, “S” e “T”) associados à coleção intitulada “Conferência de Metadados no IME” (recurso “A”);
- **Formas de expressão de conteúdo:** são identificados um “artigo” (recursos “H”, “D”, “S” e “T”) e uma “tese de mestrado” (recurso “G”) para o conteúdo intelectual intitulado “Modelo de Metadados para Documentos” e um “artigo” (recurso “E”) para o conteúdo intelectual intitulado “Modelo de Metadados para Imagem”;
- **Personificações físicas:** são identificadas duas personificações físicas para o artigo intitulado “Modelo de Metadados para Documentos”, uma em XML (recurso “H”) e outra em postScript (representada pelos recursos “D”, “S” e “T”). Para a tese de mestrado intitulada “Modelo de Metadados para Documentos” é identificada uma personificação física em Word 7.0 (recurso “G”) e para o artigo intitulado “Modelo de Metadados para Imagens”, uma personificação física em HTML (recurso “E”).

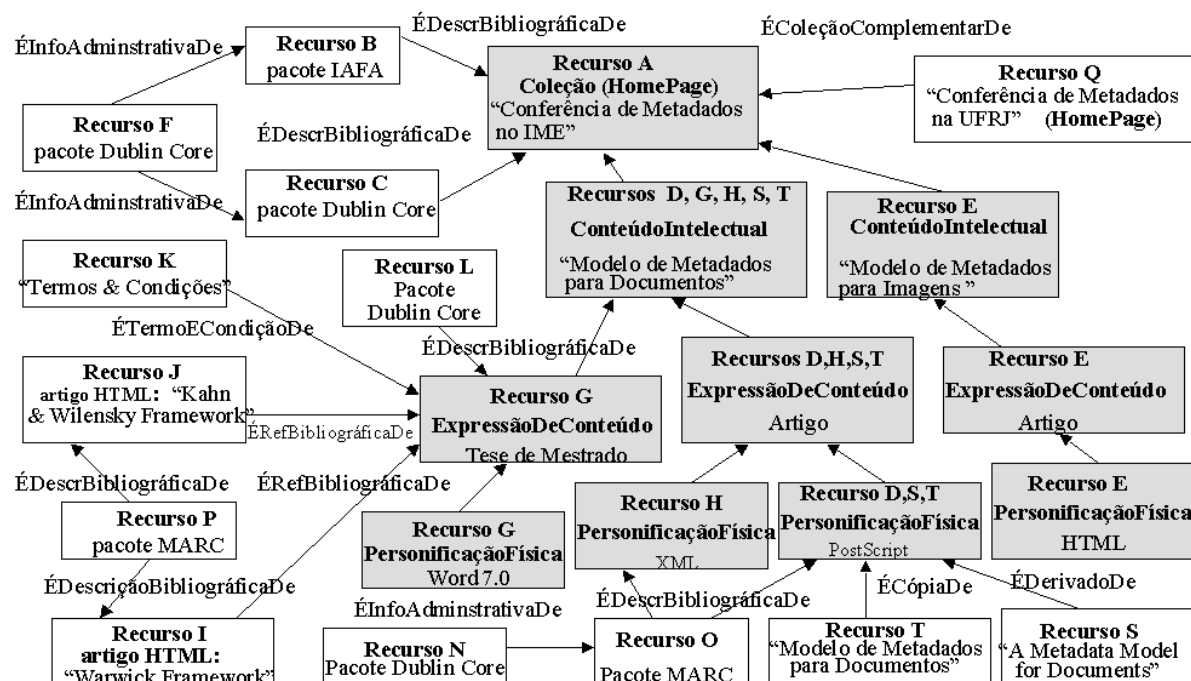


FIGURA 6.7. Representação do Exemplo segundo o Modelo de Documento Proposto

Na Figura 6.7, o exemplo descrito na seção anterior é representado segundo a estrutura organizacional proposta na Seção 6.3. Os retângulos em negrito representam a hierarquia existente entre as entidades abstratas que foram identificadas anteriormente. Nessa hierarquia, cada seta apontada de uma camada para outra representa um tipo de relacionamento estrutural (tal como “ÉPersonificaçãoFísicaDe” ou “ÉConteúdoIntelectualDe”) que não foram representados para não sobrecarregar a Figura. Os demais retângulos expressam os relacionamentos contextuais identificados em cada camada da hierarquia proposta, conforme descrito a seguir:

- “ÉColecçãoComplementarDe”: entre os recursos “A” e “Q”;
- “ÉReferênciaBibliográficaDe”: entre os recursos “G”, “I” e “J”;
- “ÉDescriçãoBibliográficaDe”: entre os recursos “B” e “A”, “C” e “A”, “O” e “D”, “O” e “H”, “L” e “G”, “P” e “I”, “P” e “J”;
- “ÉTermoECondiçãoParaAcessoDe”: entre os recursos “K” e “G”;
- “ÉInformaçãoAdministrativaDe”: entre os recursos “F” e “B”, “F” e “C”, “N” e “O”;
- “ÉCópiaDe”: entre os recursos “D” e “S”; e
- “ÉDerivadoDe”: entre os recursos “D” e “T”.

6.5.3 – Modelando o Exemplo Passo a Passo

Nesta seção, o exemplo da Figura 6.7 é representado segundo a hierarquia de objetos digitais proposta na Seção 6.4 deste trabalho.

1) Camada de Coleção

Nesta camada é modelada a coleção identificada na Subseção 6.5.2, que é representada pelo recurso “A”. Na Figura 6.8, o objeto digital identificado por OiDCol1, do tipo ObjetoColeção, é empregado para representar essa coleção. Os atributos do objeto OiDCol1 são descritos a seguir: *descriçãoObjColeção* provê uma descrição textual da coleção; *tipoRecursoColeção* informa que o ObjetoColeção está representando um recurso de informação do tipo *homepage*, cujo autor é “Barreto,C.M.”; *tipoOrganizaçãoColeção* expressa a forma de organização do conteúdo da coleção que, no exemplo em questão, é por assunto; *título* identifica o conteúdo intelectual da coleção como “Conferência de Metadados no IME”; e, por fim, *assunto* especifica o tópico “metadados” para caracterizar o conteúdo dos artigos referenciados pela coleção.

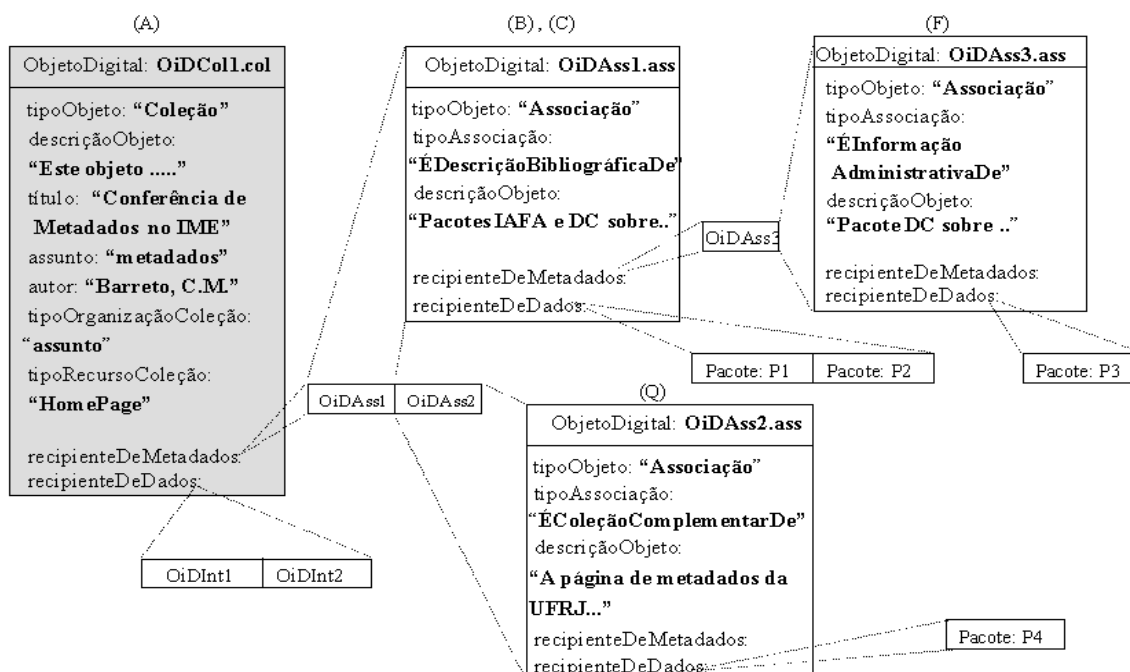


FIGURA 6.8: Modelagem dos ObjetoColeção e ObjetoAssociação

1.1) Relacionamentos Estruturais na Camada de Coleção

Quanto aos relacionamentos estruturais do objeto OiDCol1, conforme pode ser observado na Figura 6.8, seu atributo *recipienteDeDados* referencia dois objetos digitais do tipo ObjetoIntelectual, identificados por OiDInt1 e OiDInt2, que representam, respectivamente, os conteúdos intelectuais “Modelo de Metadados para a Descrição de Documentos” e “Modelo de Metadados para a Descrição de Imagens”, ambos associados à coleção representada por OiDCol1.

1.2) Relacionamentos Contextuais na Camada de Coleção

Conforme apresentado na Figura 6.8, o atributo *recipienteDeMetadados* do objeto OiDCol1 (*homepage* do IME) faz referência a dois objetos digitais do tipo ObjetoAssociação, identificados por OiDAss1 e OiDAss2.

O objeto OiDAss1 descreve o relacionamento “ÉDescriçãoBibliográficaDe” do recurso “A” com os recursos “B” e “C”. O atributo *recipienteDeDados* de OiDAss1 faz referência aos pacotes P1 e P2. Esses pacotes, por sua vez, referenciam, respectivamente, o recurso “B”, um pacote contendo descritores do padrão IAFA, e o recurso “C”, um pacote contendo descritores do padrão Dublin Core. Ambos pacotes provêm metadados bibliográficos a respeito da coleção “A”. Quanto ao atributo *recipienteDeMetadados* de OiDAss1, este referencia, recursivamente, outro ObjetoAssociação, identificado por OiDAss3. Este objeto representa o relacionamento “ÉInformaçãoAdministrativaDe” do recurso “F”, um pacote Dublin Core, com os recursos “B” e “C” (ou, mais especificamente, com o objeto OiDAss1).

Quanto ao objeto OiDAss2, também referenciado por OiDCol1, este representa o relacionamento “ÉColeçãoComplementarDe” do recurso “A” (*homepage* do IME) com o recurso “Q” (*homepage* da UFRJ), referenciado pelo pacote P4. Ao contrário do objeto OiDAss1, nenhum relacionamento contextual foi identificado para o objeto OiDAss2.

2) Camada de Conteúdo Intelectual

Nesta camada são modelados os conteúdos intelectuais distintos identificados na Subseção 6.5.2. Esses conteúdos intelectuais, apresentados na Figura 6.9, são representados por dois objetos do tipo ObjetoIntelectual, identificados por OiDInt1 e OiDInt2.

O Objeto OiDInt1 expressa o conteúdo intelectual dos recursos “H”, “D” e “G”, cujo autor é “Barreto, C.M.”. É identificado pelo atributo título como “Modelo de Metadados para Documentos” e relacionado aos tópicos “metadados, documentos”. Também apresentado na Figura 6.9, o objeto OiDInt2 representa o conteúdo intelectual do recurso “E”, caracterizado pelo título “Modelo de Metadados para Imagens”.

2.1) Relacionamentos Estruturais na Camada de Conteúdo Intelectual

Nesta camada, o atributo *recipienteDeDados* do objeto OiDInt1 referencia os objetos digitais OiDExpr1 e OiDExpr2 que representam, respectivamente, as expressões de conteúdo “artigo” e “tese de mestrado” associadas ao conteúdo intelectual representado pelo objeto OiDInt1. OiDInt2, por sua vez, referencia o objeto digital OiDExpr3 que representa a expressão de conteúdo “artigo” associada a OiDInt2.

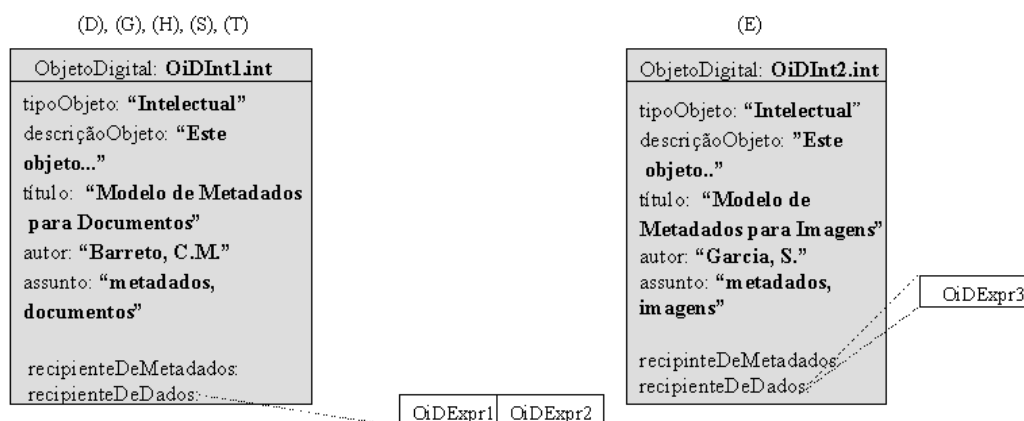


FIGURA 6.9: Modelagem do ObjetoIntelectual

2.1) Relacionamentos Contextuais na Camada de Conteúdo Intelectual

Nenhum relacionamento contextual foi identificado para os objetos OiDInt1 e OiDInt2, conforme pode ser observado na Figura 6.9.

3) Camada de Expressão de Conteúdo

Nesta camada são representadas as expressões de conteúdo distintas identificadas na Subseção 6.5.2 para os conteúdos intelectuais representados pelo objetos OiDCol1 e OiDCol2, modelados na camada anterior.

Na Figura 6.10, é apresentada a modelagem do ObjetoDeExpressão identificado por OiDExpr1 que representa a expressão de conteúdo “artigo”, publicada pelo IME em 10/10/98”, referente aos recursos “H”, “D”, “S” e “T” da Figura 6.7. O atributo recipienteDeDados de OiDExpr1 referencia os objetos OiDComp1 e OiDComp2 que representam, respectivamente, as personificações físicas postScript e XML associadas à expressão de conteúdo representada por OiDExpr1.

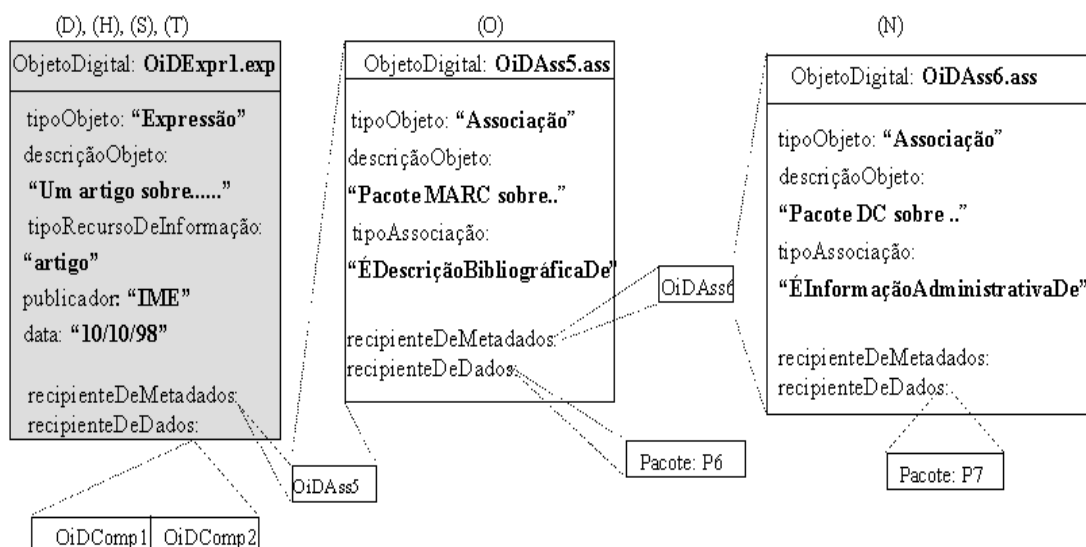


FIGURA 6.10: Modelagem do ObjetoDeExpressão

Na Figura 6.11, por sua vez, são modelados os objetos OiDExpr2 e OiDExpr3 que representam, respectivamente, a expressão de conteúdo “tese de mestrado” referente ao recurso “G” (“Modelo de Metadados p/ Documentos”) e a expressão de conteúdo “artigo” referente ao recurso “E” (“Modelo de Metadados para Imagens”), conforme apresentado na

Figura 6.7. Uma descrição textual de OiDExpr1, OiDExpr2 e OiDExpr3 é fornecida pelo atributo *descriçãoObjeto* de cada objeto.

3.2) Relacionamentos Contextuais na Camada de Expressão de Conteúdo

Nesta camada, conforme apresentado na Figura 6.10, o atributo *recipienteDeDados* de OiDExpr1 referencia os objetos OiDComp1 e OiDComp2. OiDComp1 representa a personificação física “postScript” e OiDComp2, a personificação física “XML”, ambas associadas à expressão de conteúdo “artigo”, representada pelo objeto OiDExpr1.

Na Figura 6.11, o atributo *recipienteDeDados* de OiDExpr2 referencia o objeto OiDInst1 que representa a personificação física “Word 7.0” (recurso “G”, na Figura 6.7) associada à expressão de conteúdo “tese de mestrado”, representada por OiDExpr2.

Também apresentado na Figura 6.11, o atributo *recipienteDeDados* do objeto OiDExpr3, por sua vez, referencia o objeto OiDInst2 que representa a personificação física “HTML” (recurso “E”, na Figura 6.7) associada à expressão de conteúdo “artigo” representada por OiDExpr3.

3.2) Relacionamentos Contextuais na Camada de Expressão de Conteúdo

Na Figura 6.10, observa-se que o atributo *recipienteDeMetadados* do objeto OiDExpr1 referencia o objeto identificado por OiDAss5, do tipo ObjetoAssociação. OiDAss5 descreve o relacionamento “ÉDescriçãoBibliográficaDe” da expressão de conteúdo “artigo”, representada por OiDExpr1, com o recurso “O”, um pacote (P6) contendo descritores do padrão MARC que provê informações bibliográficas a respeito de todas as personificações físicas associadas ao artigo em questão (recursos “H”, “D”, “S” e “T” da Figura 6.7). O atributo *recipienteDeMetadados* de OiDAss5 referencia, recursivamente, o objeto do tipo ObjetoAssociação identificado por OiDAss6 que descreve o relacionamento “ÉInformaçãoAdministrativaDe” entre os recursos “O” e “N”, este último um pacote (P7) contendo descritores do padrão Dublin Core que provê informações administrativas a respeito de “O”.

Na Figura 6.11, observa-se o objeto OiDExpr2. Seu atributo *recipienteDeMetadados* referencia três objetos do tipo ObjetoAssociação: OiDAss7,

OiDAss8 e OiDAss9. O objeto OiDAss7 descreve o relacionamento “ÉTermoECondiçãoParaAcessoDe” entre os recursos “G” (“tese de mestrado”) e “K”, um recurso contendo termos e condições para acesso ao conteúdo de “G”. Neste exemplo, o recurso “K” é referenciado pelo pacote P8.

O objeto OiDAss8 descreve o relacionamento “ÉDescriçãoBibliográficaDe” entre os recursos “G” e “L”. O recurso “L” contém descritores do padrão Dublin Core provendo informações bibliográficas a respeito de “G”. É referenciado pelo pacote P9, conforme pode ser observado na Figura 6.11.

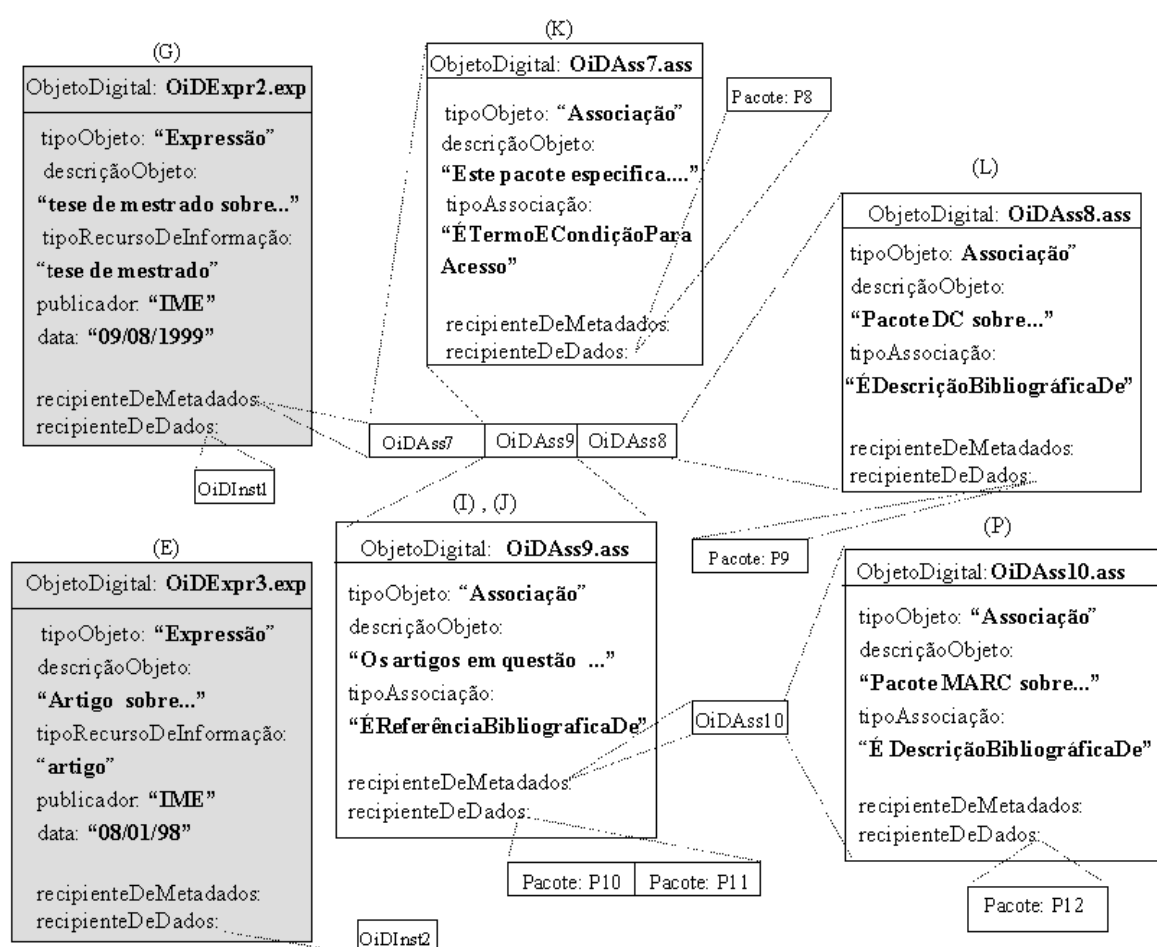


FIGURA 6.11: Modelagem do ObjetoDeExpressão - continuação

Por fim, o objeto OiDAss9 descreve o relacionamento “ÉReferênciaBibliográficaDe” entre os recursos “G” (uma “tese de mestrado”), “I” (um artigo intitulado “Warwick Framework”) e “J” (outro artigo intitulado “Kahn & Wilensky Framework”). O atributo *recipienteDeDados* de OiDAss9 referencia os pacotes P10 e P11

que, por sua vez, referenciam os recursos “I” e “J”. O atributo *recipienteDeMetadados* do objeto OiDAss9 referencia o objeto digital OiDAss10. Esse objeto descreve o relacionamento “ÉDescriçãoBibliográficaDe” entre os recursos “P”, “I” e “J”. O recurso “P”, referenciado pelo pacote P12, contém descritores do padrão MARC que fornecem informações bibliográficas sobre “I” e “J”.

Nenhum relacionamento contextual foi identificado para o objeto OiDExpr3, na Figura 6.11.

4) Camada de Personificação Física

Nesta camada são representadas as personificações físicas distintas identificadas para as expressão de conteúdo OiDExpr1, OiDExpr2 e OiDExpr3, modeladas na camada anterior.

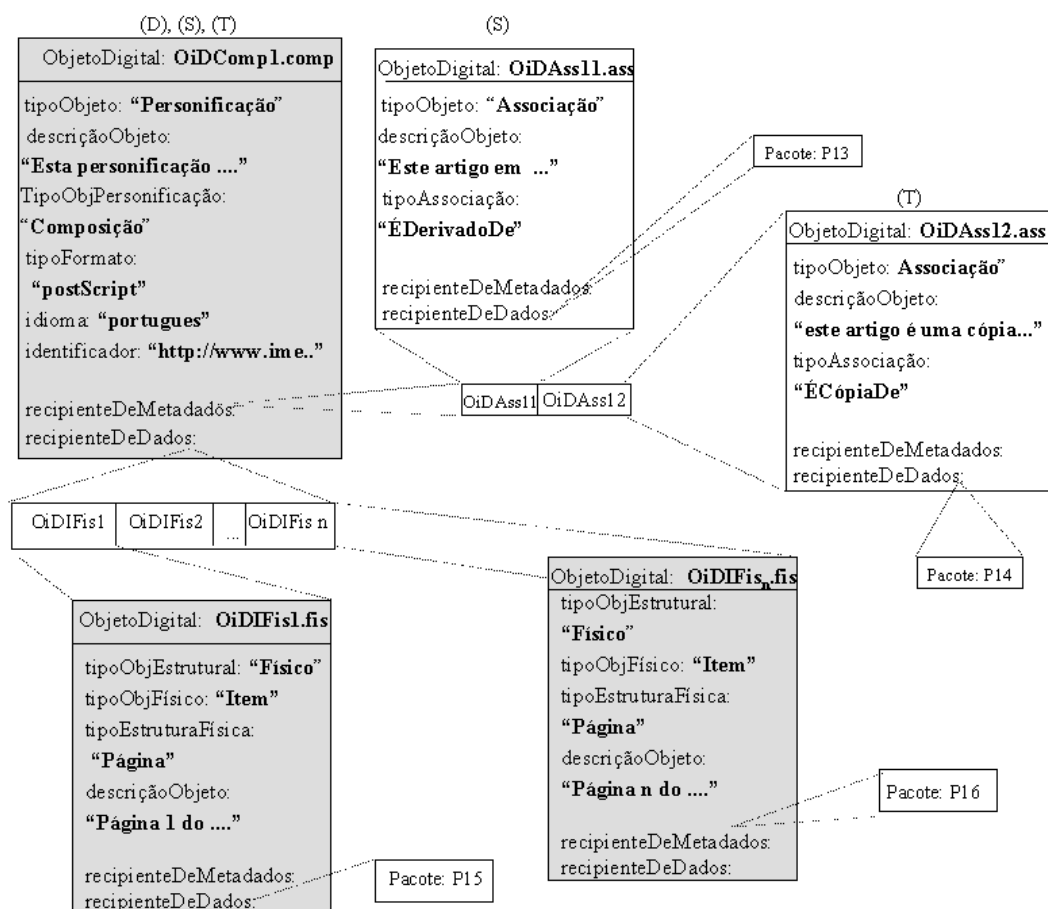


FIGURA 6.12: Modelagem dos ObjetoDeComposição e ObjetoEstrutural

4.1) Relacionamentos Estruturais na Camada de Personificação Física

Nesta camada, o atributo *recipienteDeDados* de OiDComp1, apresentado na Figura 6.12, referencia um conjunto de objetos digitais do tipo ObjetoItemFísico, cada qual identificado sequencialmente de OiDFis₁ a OiDFis_n. Esses objetos representam a visão estrutural física da personificação “postScript”, representada pelo objeto OiDComp1.

O atributo *recipienteDeDados* do objeto OiDComp2, apresentado na Figura 6.13, referencia um objeto digital do tipo ObjetoCompHipermedia (OiDCHip1) e um objeto digital do tipo ObjetoCompLógico (OiDCLog1). OiDCHip1 e OiDCLog1 representam visões estruturais distintas da personificação física “XML” representada pelo objeto OiDComp2.

Quanto aos objetos OiDInst1 e OiDInst2 da Figura 6.14, estes referenciam os pacotes P19 e P20 que, por sua vez, referenciam a porção “dado” dos documentos cujos identificadores são informados pelo atributo *identificador* de ambos objetos.

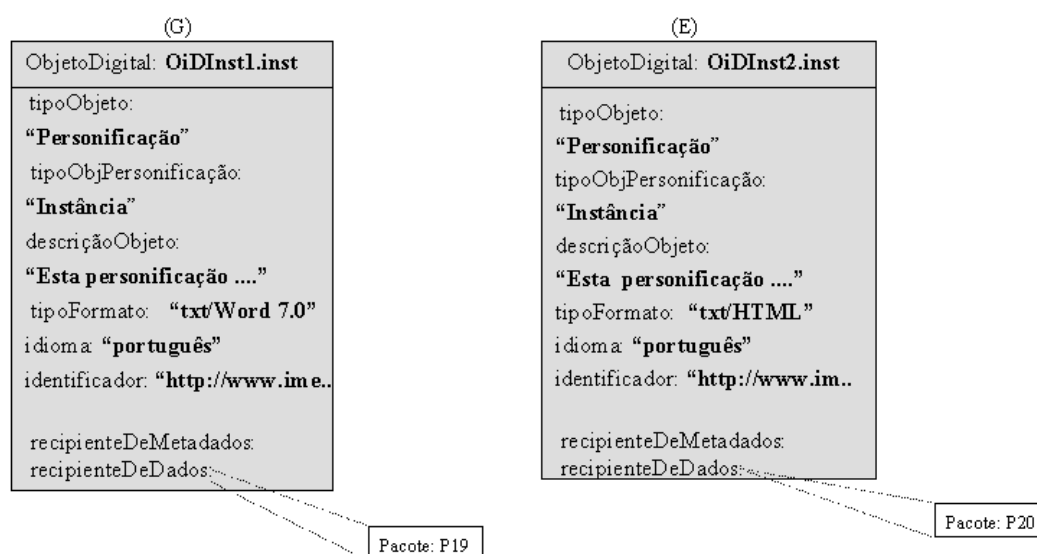


FIGURA 6.14: Modelagem do ObjetoInstância

4.1) Relacionamentos Contextuais na Camada de Personificação Física

O atributo *recipienteDeMetadados* do objeto OiDComp1, apresentado na Figura 6.12, referencia os objetos digitais, do tipo ObjetoAssociação, identificados por OiDAss11 e OiDAss12.

OiDAss11 descreve o relacionamento “ÉDerivadoDe” da personificação física representada por OiDComp1 (recurso “D”) com o recurso “S”, na Figura 6.7. O atributo *recipienteDeDados* de OiDAss11 referencia o pacote P13 que, por sua vez, referencia o recurso “S”. OiDAss12 descreve o relacionamento “ÉCópiaDe” do recurso “D” com o recurso “T”, na Figura 6.7. O atributo *recipienteDeDados* de OiDAss12 referencia o pacote P14, que referencia o recurso “S”.

Nenhum relacionamento contextual foi identificado para os objetos OiDComp2, OiDInst1 e OiDInst2, conforme pode ser observado nas Figuras 6.13 e 6.14.

5) Camada de Estrutura

Nesta camada são representadas as diversas visões estruturais de cada personificação física distinta modelada na camada anterior. Os componentes estruturais decorrentes da segmentação de um recurso não foram apresentados na Figuras 6.6 e 6.7 para fins de simplificação do exemplo.

Na Figura 6.12, cada objeto digital do tipo ObjetoItemFísico referenciado por OiDComp1 representa uma unidade estrutural física (no caso “páginas”, conforme descrito pelo atributo *tipoEstruturaFísica*) da personificação física “postScript”, representada pelo objeto OiDComp1. O atributo *recipienteDeDados* de cada ObjetoItemFísico ($OiDIFis_1$ a $OiDIFis_n$) referencia um pacote que, por sua vez, referencia a porção “dado” da unidade estrutural física representada pelo ObjetoItemFísico.

Quanto ao objeto OiDComp2, apresentado na Figura 6.13, seu atributo *recipienteDeDados* referencia os objetos OiDCHip1 e OiDCLog1. O objeto OiDCHip1 é do tipo ObjetoCompHipermedia. Seu atributo *tipoEstruturaHipermedia* referencia o tipo MIME “*multipart/mixed*”, indicando que o componente hipermedia que está sendo representado pelo objeto digital pode conter componentes estruturais de tipos de mídias distintos (tais como textos, imagens, etc.). Esses componentes são referenciados pelo atributo *recipienteDeDados* de OiDCHip1. Quanto ao objeto OiDCLog1, este é um objeto digital do tipo ObjetoCompLógico que está representando um componente lógico do tipo “capítulo”, conforme identificado pelo atributo *tipoEstruturaLógico*. Seu atributo *recipienteDeDados* referencia um conjunto de objetos digitais do tipo ObjetoItemLógico,

cada qual representando um componente lógico (seções, parágrafos, etc.) que compõem o capítulo representado por OiDCLog1.

5.1) Relacionamentos Contextuais na Camada de Estrutura

Observando o exemplo apresentado na Figura 6.7, verifica-se que nenhum relacionamento contextual foi identificado nesta camada. Entretanto, um exemplo típico de relacionamento contextual nesta camada pode ser representado pela associação dos componentes estruturais do documento (lógico, físico, hipermídia, etc.) com anotações efetuadas pelo usuário da Web (críticas, revisões, informações adicionais, etc.) a respeito do conteúdo desses componentes.

6.6 – CONSIDERAÇÕES FINAIS

Neste capítulo foi descrito um modelo de objetos digitais para a representação de um documento eletrônico na Web. É baseado em arquiteturas de metadados recentes, cujas principais contribuições para o seu desenvolvimento são comentadas a seguir:

- **Arquitetura de Kahn e Wilensky:** define o conceito de objeto digital, uma estrutura abstrata, identificada de forma única, para a representação de diversos recursos de informação com base em metadados. No MODDE, objetos digitais são usados para possibilitar a gerência de um documento com base em seu conteúdo intelectual;
- **Arquitetura Warwick:** é utilizada para prover uma estrutura ao objeto digital, possibilitando associar dados e metadados segundo um determinado contexto de uso, que define o papel da informação contida nos pacotes (se dado ou metadado). No MODDE, um objeto digital segundo a arquitetura Warwick provê uma representação hierárquica de um documento na Web. Cada camada da hierarquia proposta representa uma entidade abstrata do documento que se deseja descrever, tais como seu conteúdo intelectual, organização externa, formas de expressão do conteúdo intelectual, personificações físicas e componentes estruturais; e
- **Modelo “DARs”:** no modelo proposto, relacionamentos são representados como objetos de primeira classe, recebendo um identificador único, podendo ter metadados próprios associados e serem executados a partir de qualquer local da Web.

Apesar de alguns atributos (baseados no Modelo Dublin Core) terem sido definidos para cada tipo de objeto digital, o propósito desses atributos, no entanto, está limitado ao contexto da descoberta e localização do recurso de informação (documento ou coleção) representado pelo MODDE.

Conforme observado nos capítulos anteriores, diversos tipos de metadados são necessários a fim de possibilitar a gerência adequada de recursos de informação na Web. Tomando como base esse requisito, no capítulo seguinte desta dissertação é apresentada uma arquitetura de metadados que possibilita a definição de modelos (ou esquemas) a partir da concatenação de termos oriundos de diversos padrões. Esses modelos serão empregados para especializar o MODDE na descrição de expressões de conteúdo e personificações físicas distintas tais como artigos, relatórios técnicos, teses de mestrado, documentos HTML, XML, postScript, etc. que, por sua vez, exigem conjuntos de metadados também distintos. Essa abordagem possibilita o desenvolvimento de ferramentas para a instanciação das propriedades de um documento com a flexibilidade exigida em um ambiente com as características da Web.

No MODDE, as propriedades de um documento são anexadas, em cada camada da hierarquia definida, por um ou mais objetos do tipo ObjetoAssociação. Esse objeto faz referência, através de seu atributo *recipienteDeDados*, a uma estrutura denominada *pacoteDeDados*, que é baseada em conceitos da arquitetura Warwick e cujo objetivo é armazenar as propriedades instanciadas do documento em cada camada da estrutura organizacional do documento. Outra estrutura, denominada *PacoteIndireto*, também é referenciada por alguns tipos de objetos digitais. No modelo proposto, *PacoteIndireto* implementa uma referência indireta a um conteúdo de informação (o conteúdo do documento, por exemplo) que pode estar localizado em qualquer local da Web. Como a arquitetura Warwick não define a estrutura de conteúdo de cada pacote, esse assunto também será tratado no próximo capítulo, com a definição de um ambiente para instanciação e utilização do MODDE.

CAPÍTULO 7

DEFINIÇÃO DE UM AMBIENTE PARA INSTANCIAÇÃO E UTILIZAÇÃO DO MODDE

7.1 – INTRODUÇÃO

A tecnologia WWW tem possibilitado um acesso sem precedentes a uma gama de informações distribuídas globalmente. Segundo Lagoze (LAGOZE & FIELDING, 1998), dentre as características mais evidentes desse ambiente de publicação se destacam: a *universalidade*, que possibilita a participação de qualquer pessoa com um investimento mínimo em conhecimento e tecnologia; *uniformidade*, que implica na coexistência de recursos de informação distintos quanto a finalidade, granularidade, tamanho e cobertura em um espaço de informação comum; e a *descentralização*, que implica na impossibilidade de estabelecimento de mecanismos únicos de controle e gerência de recursos.

Na Web, essas características se traduzem na inexistência da estrutura organizacional que é necessária para a garantia de acesso à informação contida nos documentos disponíveis. No capítulo anterior foi apresentada uma proposta de tal estrutura organizacional que possibilita a gerência de um documento ou de uma coleção de documentos com base no seu conteúdo intelectual, independentemente da forma como esses documentos estão distribuídos fisicamente na Web.

Para que se tenha um acesso eficiente a esses documentos, no entanto, é necessário que haja uma representação adequada de suas propriedades. A abordagem descrita neste capítulo enfoca o emprego de metadados para a representação das propriedades de um documento tomando como base as entidades abstratas *conteúdo intelectual*, *expressão de conteúdo*, *personificação física* e *componentes estruturais*, que são empregadas para organizar, de forma coerente e padronizada, essas propriedades.

Inicialmente, na Seção 7.2, é apresentada uma arquitetura de metadados cujo enfoque é o compartilhamento de metadados oriundos de padrões distintos. Esses

metadados podem ser empregados na definição de diversos modelos descritivos para o MODDE. Na Seção 7.3 é apresentada uma proposta de ambiente para instanciação do MODDE, tendo em vista uma futura implementação desse modelo em ambiente de banco de dados. Finalmente, na Seção 7.4, são apresentados os possíveis cenários de utilização do MODDE na Web.

7.2 – DEFINIÇÃO DE UMA ARQUITETURA DE METADADOS PARA O MODDE

7.2.1 – Conceitos Básicos

Na Web, o projeto de modelos descritivos com o objetivo de prover suporte ao acesso eficiente da informação disponível nos diversos documentos torna-se bastante difícil devido a falta de consenso entre provedores e usuários quanto ao conteúdo, representação e intercâmbio de metadados. Essa diversidade é justificada pela variedade de tipos de documentos e pelos aspectos distintos que devem ser modelados pelo metadado.

Por esses motivos, uma abordagem de metadados cujo enfoque fosse o desenvolvimento de um modelo único para o ambiente da Web seria muito complexa, devendo ser genérica o bastante de forma a abranger diversas estruturas de metadados e o conjunto de operações usadas para manipular essas estruturas no mesmo modelo.

Uma abordagem mais flexível e adaptável às requisições dos usuários da Web é aquela que enfoca o desenvolvimento de modelos de metadados distintos, cada qual projetado para o tratamento de um determinado aspecto do domínio de informação do usuário. Essa estratégia representa uma abordagem do tipo “dividir para conquistar”, possibilitando a construção de modelos (ou esquemas descritivos) a partir da concatenação de termos oriundos de diversos padrões.

Para que diferentes padrões de metadados possam coexistir, no entanto, é necessário o estabelecimento de uma infra-estrutura flexível, na qual os diversos termos que compõem esses padrões sejam organizados de forma coerente segundo uma hierarquia de descrição. Para a implementação dessa hierarquia foi considerada a arquitetura de quatro níveis proposta por Kerhervé (KERHERVÉ, 1997) que tem sido empregada por

diversas outras iniciativas envolvendo metadados tais como *CASE Data Interchange Format* (CDIF), *Unified Modeling Language* (UML) e *Meta Object Facility* (MOF).

Segundo Kerhervé, quando o objetivo do projeto de metadados é a interoperabilidade entre abordagens descritivas distintas, o sistema de gerenciamento deve suportar diversos níveis de modelagem, uma vez que metadados são produzidos para atender a propósitos diversos. Na arquitetura de metadados apresentada neste capítulo, as quatro camadas propostas por Kerhervé são empregadas com o propósito de prover uma estrutura organizacional comum para o tratamento de metadados oriundos de padrões diversos. A arquitetura estabelece uma hierarquia, apresentada na Figura 7.1, na qual os termos oriundos das iniciativas mais genéricas, tais como a arquiteturas RDF e MCF, são empregados para descrever os termos oriundos das iniciativas mais específicas, tais como os padrões Dublin Core e MARC. As quatro camadas que compõem essa arquitetura são descritas a seguir:

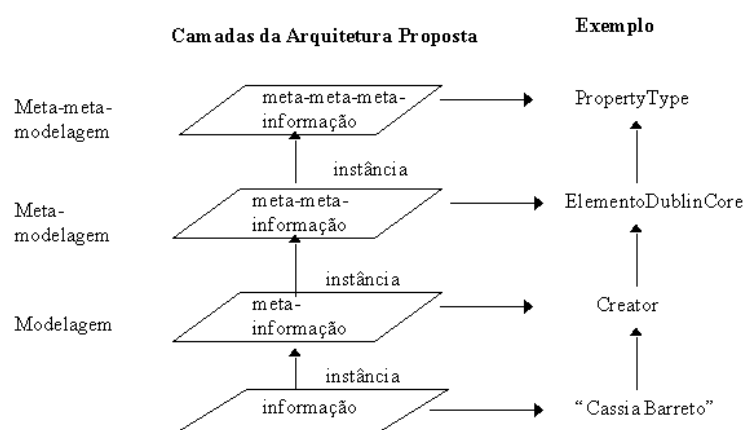


FIGURA 7.1 - Camadas da Arquitetura de Metadados Proposta

- **Camada de dados do usuário:** é a camada mais inferior da arquitetura proposta. Os objetos dessa camada correspondem aos objetos do domínio de informação do usuário que são representados em sua forma bruta, ou seja, sem qualquer descrição associada, tal como “Cassia Barreto”, no exemplo da Figura 7.1. Nesta camada, a informação representa uma instância da informação da camada de modelagem;
- **Camada de modelagem:** provê os conceitos que descrevem a informação da camada de dados do usuário. Esses conceitos compõem os diversos modelos projetados para a descrição da informação na camada de dados do usuário. São exemplos de conceitos desta camada os termos “creator”, “format” e “date” oriundos do padrão Dublin Core e

as classes de recursos “documento” e “imagem” que podem ser descritas por esses termos. A informação nesta camada, denominada meta-informação, é instância da informação da camada de meta-modelagem, conforme apresentado na Figura 7.1;

- **Camada de meta-modelagem:** provê os conceitos que descrevem a informação da camada de modelagem. Conforme pode ser observado na Figura 7.1, esses conceitos correspondem à meta-meta-informações na arquitetura proposta, sendo empregados para descrever o formalismo e a especificidade adotado por um determinado padrão de metadados na modelagem de uma parcela do domínio de informação do usuário. Podem ser citados como exemplos de conceitos desta camada os termos “template” e “cluster”, empregados para referenciar as classes de recursos que podem ser descritas pelo padrão IAFA. A informação nesta camada é instância da informação da camada de meta-meta-modelagem; e
- **Camada de meta-meta-modelagem:** provê os conceitos que descrevem os objetos da camada de meta-modelagem. Esses conceitos correspondem à meta-meta-meta-informações na hierarquia proposta, conforme apresentado na Figura 7.1. Nesta camada, também denominada *camada raiz* por Kerhervé, os conceitos são empregados para descrever a estrutura da informação nas camadas de meta-modelagem e modelagem. Basicamente, os conceitos desta camada (providos por arquiteturas tais como MCF e esquema RDF) são genéricos o bastante de forma a expressar: *hierarquias de classes e de composição de classes*, que possibilitam a organização de conceitos em categorias; *restrição sobre o tipo de dado*, que especifica a classe de objeto relativa ao conteúdo válido de um elemento (inteiro, caracter, booleano, etc.); etc. Na Figura 7.1, por exemplo, o conceito “PropertyType” provido pela arquitetura MCF é empregado para descrever o conceito “*ElementoDublinCore*”. De uma forma geral, os conceitos da camada de meta-meta-modelagem são os mais genéricos da arquitetura de metadados proposta, daí a dificuldade em se estendê-la além do quarto nível. São usados, basicamente, para a unificação dos conceitos empregados para descrever classes de objetos (ou recursos de informação) e propriedades de objetos em cada camada da arquitetura de metadados proposta.

Um exemplo da arquitetura de modelagem proposta aplicada aos padrões da Web é apresentado na Figura 7.2. Da esquerda para a direita, os conceitos de cada camada são associados através do relacionamento “descreve”. De cima para baixo, são associados pelo

relacionamento “instância de”. Observa-se, no exemplo, que não existe uma definição clara do que é “dado” e do que é “metadado”. De fato, a informação em cada camada pode assumir o “papel” de dado ou de metadado, dependendo do contexto no qual esteja inserida. Na camada de meta-modelagem, por exemplo, os conceitos “IAFATemplate” e “IAFADataElement” são instâncias, respectivamente, dos conceitos “MCF:Category” e “MCF:PropertyType” da camada de meta-meta-modelagem. Portanto, os conceitos “IAFATemplate” e “IAFADataElement” estão no papel de “dado” em relação a camada de meta-meta-modelagem. Para a camada de modelagem, no entanto, ambos os conceitos estão no papel de “metadado”, uma vez que descrevem, respectivamente, os conceitos “IAFADocument” e “Author”.

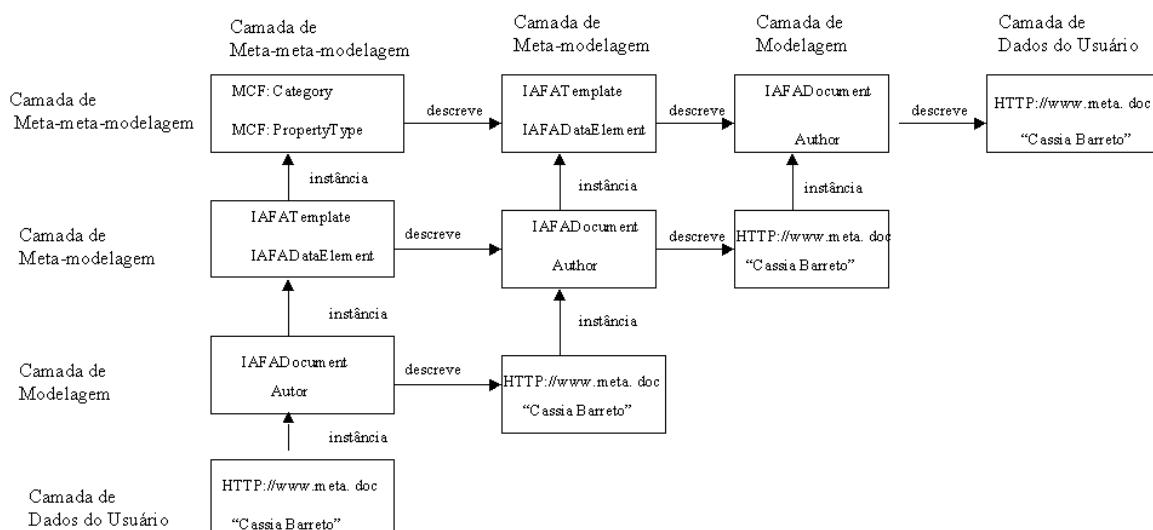


FIGURA 7.2 - Exemplo da Arquitetura de Metadados Proposta

7.2.2 - Sistema Inicial da Arquitetura Proposta

O propósito principal da arquitetura de metadados definida neste capítulo é prover interoperabilidade entre abordagens descritivas distintas. Para tal, deve possibilitar a declaração de metadados oriundos dos diversos padrões propostos para a Web segundo a hierarquia definida na seção anterior. Essa declaração é possível através de conceitos iniciais que são providos por cada camada. Esses conceitos constituem o sistema inicial (ou de inicialização) da arquitetura de metadados proposta e objetivam prover um formalismo para a representação homogênea da estrutura da informação em cada camada.

Na arquitetura proposta, o sistema inicial definido para cada camada é baseado nas linguagens declarativas para a representação de conhecimento adotadas pelas arquiteturas MCF e Esquema RDF. Em uma linguagem declarativa, propriedades são definidas em termos das classes de objetos em que se aplicam, ao contrário dos sistemas orientados a objeto onde uma classe é definida em termos das propriedades que suas instâncias podem ter. Esse enfoque centrado na propriedade objetiva atender a um axioma definido para a Web que implica na possibilidade de uma aplicação qualquer declarar uma propriedade a respeito de um recurso de informação existente, independentemente de um mecanismo de registro de propriedades centralizado.

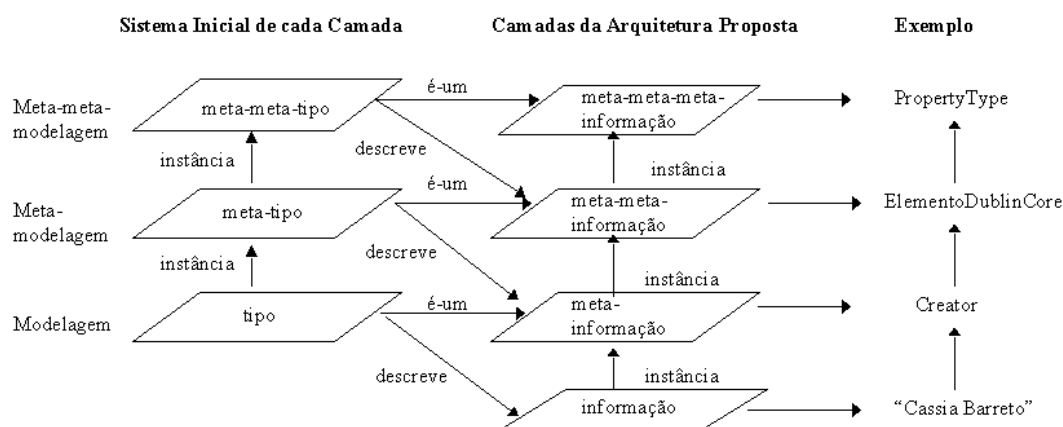


FIGURA 7.3 - Sistema Inicial da Arquitetura de Metadados Proposta

Conforme pode ser observado na Figura 7.3, na arquitetura de metadados proposta, o sistema inicial das camadas de modelagem, meta-modelagem e meta-meta-modelagem é constituído, respectivamente, por Tipos, MetaTipos e MetaMetaTipos que são empregados para especificar ou restringir a semântica da informação em suas respectivas camadas. Os componentes desse sistema inicial são descritos a seguir:

- **Descritor:** representa um conceito qualquer que pode ser empregado para descrever informações em qualquer camada da hierarquia apresentada na Figura 7.3. Seus atributos são: *nome*, que serve para especificar ou restringir a semântica do descritor; *descrição*, que provê um texto livre para o detalhamento de seu significado semântico; *subClasseDe*, que referencia o descritor que é super-classe do descritor que está sendo declarado; e *instânciaDe*, que referencia o descritor do qual aquele que está sendo declarado é uma instância. A classe Descritor é super-classe de MetaMetaTipo, MetaTipo e Tipo, descritas a seguir.

- **MetaMetaTipo:** representa um conceito qualquer da camada de meta-meta-modelagem. É super-classe de:
 - **MetaMetaTipoDeCategoria:** é um MetaMetaTipo que representa uma classe qualquer de recurso de informação (ou de objetos) que pode ser descrita na camada de meta-meta-modelagem; e
 - **MetaMetaTipoDePropriedade:** é um MetaMetaTipo que representa um nome de propriedade (ou relacionamento) que pode ser aplicada para descrever uma ou mais classes de recursos de informação da camada de meta-meta-modelagem.
- **MetaTipo:** representa um conceito qualquer da camada de meta-modelagem. Um MetaTipo representa uma instância do MetaMetaTipo referenciado pelo atributo *instânciaDe*. A classe MetaTipo é super-classe de:
 - **MetaTipoDeCategoria:** é um MetaTipo que representa uma classe qualquer de recurso de informação (ou de objetos) da camada de meta-modelagem; e
 - **MetaTipoDePropriedade:** é um MetaTipo que representa um nome de propriedade (ou relacionamento) que pode ser aplicada a uma ou mais classes de recursos de informação da camada de meta-modelagem.
- **Tipo:** representa um conceito qualquer da camada de modelagem. Um Tipo representa uma instância do MetaTipo que é referenciado pelo atributo *instânciaDe*. A classe Tipo é super-classe de:
 - **TipoDeCategoria:** é um Tipo que representa uma classe qualquer de recurso de informação (ou de objetos) da camada de modelagem. É super-classe de:
 - **TipoPrimitivo:** é um TipoDeCategoria que representa um tipo primitivo de dado. É super-classe para os TipoDeCategoria Inteiro e Caracter. Pode ser facilmente estendida para outros tipos de dados;
 - **Pessoa:** é um TipoDeCategoria que representa a super-classe para qualquer recurso de informação do tipo pessoa; e
 - **Organização:** é um TipoDeCategoria que representa a super-classe para qualquer recurso de informação do tipo organização;
 - **TipoDePropriedade:** é um Tipo que representa um nome de propriedade (ou relacionamento) que pode ser aplicada a uma ou mais classes de recursos de informação da camada de modelagem. É super-classe de:

- **Agregado:** é um TipoDePropriedade que representa a super-classe para qualquer tipo de propriedade cujo valor corresponde a uma agregação de valores de outras propriedades. Endereço é um exemplo de propriedade do tipo Agregado, pois seu valor agrega o valor das propriedades “rua”, “número”, etc.;
- **Elemento:** é um TipoDePropriedade que representa a super-classe para qualquer tipo de propriedade que possa ser empregada para descrever uma característica ou relacionamento de um recurso de informação na camada de modelagem; e
- **Qualificador:** é um TipoDePropriedade que representa a super-classe para qualquer tipo de propriedade que possa ser empregada para descrever uma outra propriedade na camada de modelagem.

Um exemplo do sistema inicial definido para cada camada da arquitetura de metadados proposta é apresentado a seguir. Foi codificado na XML, uma linguagem de propósito geral para a representação de dados. Segundo Guha (GUHA, 1997), o principal componente de uma linguagem de representação de dados é um sistema de Tipos. Na linguagem XML um sistema de Tipos é expresso em termos de *elementos* e *atributos*, conforme apresentado a seguir:

```
<xml>
  <Descritor id="MetaMetaTipo" >
    <nome>MetaMetaTipo</nome>
    <descrição>representa a super-classe para todos os MetaMetaTipos criados
  </descrição>
  </Descritor>

  <Descritor id="MetaMetaTipoDeCategoria" >
    <nome>MetaMetaTipoDeCategoria</nome>
    <subClasseDe href="#MetaMetaTipo"/>
    <descrição>representa a super-classe para todas as MetaMetaTipoDeCategorias
criadas</descrição>
  </Descritor>

  <Descritor id = "MetaMetaTipoDePropriedade" >
    <nome>MetaMetaTipoDePropriedade</nome>
    <subClasseDe href="#MetaMetaTipo"/>
    <descrição>representa a super-classe para todas as MetaMetaTipoDePropriedades
criadas</descrição>
  </Descritor>
```

```

<Descritor id = "MetaTipo" >
  <nome>MetaTipo</nome>
  <descrição>representa a super-classe para todas os MetaTipos criados </descrição>
  <instânciaDe href="# MetaMetaTipo"/>
</Descritor>

<Descritor id="MetaTipoDeCategoria" >
  <nome>MetaTipoDeCategoria</nome>
  <subClasseDe href="#MetaTipo"/>
  <descrição>representa a super-classe para todas as MetaTipoDeCategorias
criadas</descrição>
  <instânciaDe href="# MetaMetaTipoDeCategoria"/>
</Descritor>

<Descritor id="MetaTipoDePropriedade" >
  <nome>MetaTipoDePropriedade</nome>
  <subClasseDe href="#MetaTipo"/>
  <instânciaDe href="# MetaMetaTipoDePropriedade"/>
  <descrição>representa a super-classe para todas as MetaTipoDePropriedades
criadas</descrição>
</Descritor>

<Descritor id="Tipo" >
  <nome>Tipo</nome>
  <descrição>representa a super-classe para todas os Tipos criados </descrição>
  <instânciaDe href="#MetaTipo"/>
</Descritor>

<Descritor id="TipoDeCategoria" >
  <nome>TipoDeCategoria</nome>
  <subClasseDe href="#Tipo" />
  <instânciaDe href="#MetaTipoDeCategoria"/>
  <descrição>representa a super-classe para todas os TipoDeCategorias
criados</descrição>
</Descritor>

<Descritor id="TipoDePropriedade" >
  <nome>TipoDePropriedade</nome>
  <subClasseDe href="#Tipo" />
  <instânciaDe href="#MetaTipoDePropriedade"/>
  <descrição>representa a super-classe para todas os TipoDePropriedades
criados</descrição>
</Descritor>

<Descritor id="Pessoa">
  <nome>Pessoa</nome>

```

```

    <descrição>representa a super-classe para todos os tipos de recurso de informação do
    tipo pessoa </descrição>
    <subClasseDe href="#TipoDeCategoria"/>
    <instânciaDe href="#MetaTipoDeCategoria"/>
  </Descritor>

  <Descritor id="Organização">
    <nome>Organização</nome>
    <descrição>representa a super-classe para todos os tipos de recurso de informação do
    tipo organização </descrição>
    <subClasseDe href="#TipoDeCategoria"/>
    <instânciaDe href="#MetaTipoDeCategoria"/>
  </Descritor>

  <Descritor id="TipoPrimitivo">
    <nome>TipoPrimitivo</nome>
    <descrição>representa a super-classe para todos os tipos primitivos de
    dado</descrição>
    <subClasseDe href="#TipoDeCategoria"/>
    <instânciaDe href="#MetaTipoDeCategoria"/>
  </Descritor>
</xml>

```

7.2.3 - Sistema de Tipos Definido para o MODDE

Apesar da arquitetura de metadados definida neste capítulo poder ser empregada para descrever as propriedades que qualquer categoria de recurso (ou objeto) do domínio de informação do usuário, o seu propósito principal é possibilitar a descrição das propriedades de um documento em cada camada da estrutura organizacional proposta no Capítulo 6 segundo diversas abordagens descritivas. Para tal, as entidades abstratas *conteúdo intelectual*, *expressão de conteúdo*, *personificação física* e *componente estrutural* que compõem essa estrutura devem ser representadas, na camada de modelagem da arquitetura de metadados proposta, como classes de recursos distintas, que podem ser descritas por conjuntos de metadados (modelos ou esquemas) também distintos. Uma classe qualquer de recurso de informação é declarada, na camada de modelagem, tomando como base o seu sistema inicial, como uma subclasse de TipoDeCategoria, conforme descrito a seguir:

- **Coleção:** é a subclasse de TipoDeCategoria que representa a super-classe para qualquer recurso de informação do tipo coleção. São exemplos de coleções: sítios, *homepages*, tabelas de conteúdos, categorias de assunto do catálogo YAHOO!, etc.;

- **Documento:** é a subclasse de TipoDeCategoria que representa a super-classe para qualquer recurso de informação do tipo documento. São sub-classes de Documento os TipoDeCategorias descritos a seguir:
 - **ConteúdoIntelectual:** é a subclasse de Documento que representa a super-classe para qualquer recurso de informação representado na camada de conteúdo intelectual. Como exemplo de recurso de informação definido nesta camada, pode-se citar o conteúdo intelectual identificado por “Modelo de Metadados para Documentos Eletrônicos”;
 - **ExpressãoDeConteúdo:** é a subclasse de Documento que representa a super-classe para qualquer recurso de informação representado na camada de expressão de conteúdo. Subordinados à classe ExpressãoDeConteúdo podem ser criados outros TipoDeCategorias tais como TeseDeMestrado, Artigo, RelatórioTécnico, etc.;
 - **PersonificaçãoFísica:** é a subclasse de Documento que representa a super-classe para qualquer recurso de informação representado na camada de personificação física. Subordinados à classe PersonificaçãoFísica podem ser criados outros TipoDeCategorias tais como DocumentoHTML, DocumentoXML, etc.; e
 - **ComponenteEstrutural:** é a subclasse de Documento que representa a super-classe para qualquer componente estrutural do modelo conceitual de documento proposto. Este TipoDeCategoria pode ser especializado em: *ComponenteEstruturalLógico*, *ComponenteEstruturalFísico*, etc.;
- **Associação:** é a subclasse de TipoDeCategoria que representa a super-classe para qualquer categoria de relacionamento entre recursos de informação.

Um exemplo codificado na linguagem XML do sistema de Tipos definido para o MODDE é apresentado a seguir:

```
<xml>
  <Tipo id="Documento">
    <nome>Documento</nome>
    <descrição>representa a super-classe para todos os tipos de documentos</descrição>
    <subClasseDe href="#TipoDeCategoria"/>
    <instânciaDe href="#MetaTipoDeCategoria"/>
  </Tipo>

  <Tipo id="ConteúdoIntelectual">
    <nome>ConteúdoIntelectual</nome>
    <descrição>representa a super-classe para todos os recursos do tipo conteúdo
      intelectual</descrição>
```

```

    <subClasseDe href="#Documento"/>
    <instânciaDe href="#MetaTipoDeCategoria" />
</Tipo>

<Tipo id="ExpressãoDeConteúdo">
  <nome>ExpressãoDeConteúdo</nome>
  <descrição>representa a super-classe para todos os recursos do tipo expressão de
    conteúdo</descrição>
  <subClasseDe href="#Documento"/>
  <instânciaDe href="#MetaTipoDeCategoria"/>
</Tipo>

<Tipo id="PersonificaçãoFísica">
  <nome>PersonificaçãoFísica</nome>
  <descrição>representa a super-classe para todos os recursos do tipo personificação
    física</descrição>
  <subClasseDe href="#Documento"/>
  <instânciaDe href="#MetaTipoDeCategoria"/>
</Tipo>

<Tipo id="ComponenteEstrutural">
  <nome>ComponenteEstrutural</nome>
  <descrição>representa a super-classe para todos os tipos de componentes associados
as diversas visões estruturais de um documento</descrição>
  <subClasseDe href="# Documento"/>
  <instânciaDe href="#MetaTipoDeCategoria"/>
</Tipo>
</xml>

```

7.2.4 – Definição de Modelos para Descrição do MODDE

Na arquitetura de metadados proposta, modelos descritivos para o MODDE podem ser definidos a partir dos conceitos providos pela camada de modelagem, tomando como base o seu sistema de Tipos (especificado na subseção anterior). Considerando o ambiente de escala global da Web, um modelo deve funcionar como um *dicionário*, possibilitando a interpretação correta dos conceitos nele contidos por qualquer agente de coleta de metadados independentemente dos aspectos culturais envolvidos.

Basicamente, um modelo definido na camada de modelagem faz *referência* a um conjunto de Tipos (TipoDeCategorias e TipoDePropriedades) e *declara* as relações que são válidas entre esses Tipos. Os *Tipos* referenciados descrevem a semântica das classes de recursos de informação descritas pelo modelo e a semântica das propriedades que podem

ser empregadas para descrever esses recursos. As *relações* descrevem a estrutura da informação descrita pelo modelo em questão, especificando as hierarquias (de composição, de generalização/especialização, etc.) existentes entre os Tipos por ele referenciados, as combinações válidas entre TipoDeCategorias e TipoDePropriedades e os valores válidos que podem ser atribuídos a esses TipoDePropriedades. Os conceitos empregados para descrever as relações existentes entre os Tipos referenciados por um modelo são obtidos da camada de meta-meta-modelagem.

Um exemplo de um modelo descritivo para o MODDE, codificado na linguagem XML, é apresentado a seguir:

```
<xml>
  <Modelo id="MODDE" >
    <TipoDeCategoria id="Documento">
      <nome>Documento</nome>
      <descrição>representa a super-classe para todos os tipos de documentos</descrição>
      <instânciaDe href="#RecursoDublinCore"/>
    </TipoDeCategoria>

    <TipoDeCategoria id="ConteúdoIntelectual">
      <nome>ConteúdoIntelectual</nome>
      <descrição>representa a super-classe para todos os recursos do tipo conteúdo
        intelectual</descrição>
      <subClasseDe href="#Documento"/>
      <instânciaDe href="#RecursoDublinCore" />
    </TipoDeCategoria>

    <TipoDePropriedade id="Identifier">
      <nome>Subject</nome>
      <descrição>representa o elemento "identifier" do padrão Dublin Core</descrição>
      <subClasseDe href="#Elemento"/>
      <instânciaDe href="#ElementoDublinCore"/>
    </TipoDePropriedade>

    <TipoDePropriedade id="Subject">
      <nome>Subject</nome>
      <descrição>representa o elemento "subject" do padrão Dublin Core</descrição>
      <subClasseDe href="#Elemento"/>
      <instânciaDe href="#ElementoDublinCore"/>
    </TipoDePropriedade>

    <TipoDePropriedade id="Scheme">
      <nome>Scheme</nome>
      <descrição>representa o qualificador "scheme" do padrão Dublin Core</descrição>
      <subClasseDe href="#Qualificador"/>
    </TipoDePropriedade>
  </Modelo>
</xml>
```

```

    <instânciaDe href="#QualificadorDublinCore"/>
</TipoDePropriedade>

<TipoDePropriedade id="Description">
    <nome>Description</nome>
    <descrição>representa o elemento "language" do padrão Dublin Core</descrição>
    <subClasseDe href="#Elemento"/>
    <instânciaDe href="#ElementoDublinCore"/>
</MetaTipoDePropriedade>

<TipoDePropriedade id="Title">
    <nome>Title</nome>
    <descrição>representa o elemento "title" do padrão Dublin Core</descrição>
    <subClasseDe href="#Elemento"/>
    <instânciaDe href="#ElementoDublinCore"/>
</TipoDePropriedade>

<TipoDePropriedade id="Creator">
    <nome>Creator</nome>
    <descrição>representa o elemento "creator" do padrão Dublin Core</descrição>
    <subClasseDe href="#Elemento"/>
    <instânciaDe href="#ElementoDublinCore"/>
</TipoDePropriedade>

<Relação>
    <descritorFonte href="#ConteúdoIntelectual"/>
    <arcoDeRelacionamento href="#parent" />
    <descritorDestino href="#Documento" />
</Relação>

<Relação>
    <descritorFonte href="#Identifier"/>
    <arcoDeRelacionamento href="#domain" />
    <descritorDestino href="#Documento" />
</Relação>

<Relação>
    <descritorFonte href="#Description"/>
    <arcoDeRelacionamento href="#domain" />
    <descritorDestino href="#ConteúdoIntelectual" />
</Relação>

<Relação>
    <descritorFonte href="#Subject"/>
    <arcoDeRelacionamento href="#domain" />
    <descritorDestino href="#ConteúdoIntelectual" />
</Relação>

```

```

<Relação>
  <descriptorFonte href="#Title"/>
  <arcoDeRelacionamento href="#domain" />
  <descriptorDestino href="#ConteúdoIntelectual" />
</Relação>

<Relação>
  <descriptorFonte href="#Creator"/>
  <arcoDeRelacionamento href="#domain" />
  <descriptorDestino href="#ConteúdoIntelectual" />
</Relação>
</modelo>
</xml>

```

O modelo definido acima pode ser empregado para verificar a validade das descrições referentes ao documento identificado por “hdl:cnri.dlib/agosto95” que são apresentadas a seguir. Neste exemplo, verifica-se que os termos (ou TipoDePropriedades) “*creator*”, “*title*”, “*description*” e “*subject*” (oriundos do padrão Dublin Core) são empregados para descrever a entidade abstrata (ou TipoDeCategoria) *ConteúdoIntelectual* relacionada ao documento em questão. Conforme declarado nas relações constantes do modelo definido anteriormente, essas propriedades são válidas e se aplicam de forma apropriada à entidade *ConteúdoIntelectual*. Nesse modelo, o conceito “*domain*”, oriundo da arquitetura MCF, serve para indicar, nas diversas relações, que a propriedade referenciada pelo elemento *descriptorFonte* (ex.: *creator*) se aplica ao recurso referenciado pelo *descriptorDestino* (ex.: *ConteúdoIntelectual*). O conceito “*parent*”, por sua vez, tem sentido de inclusão, indicando que a classe de recurso referenciada pelo *descriptorFonte* (ex.: *ConteúdoIntelectual*) pode estar contida na classe de recurso referenciada pelo *descriptorDestino* (ex.: *Documento*).

```
<xml>
```

```
<Documento>
```

```
<identifier>hdl:cnri.dlib/agosto95</identifier>
```

```
<ConteúdoIntelectual>
```

```
<creator>Cassia Barreto</creator>
```

```
<title>Modelo de Metadados para Documentos Eletrônicos</title>
```

```
<description>Este trabalho provê um modelo formal para a representação de documentos na Web</description>
```

```
<subject scheme = "LCCH" >metadados, arquiteturas e padrões</subject>
```

```

    </ConteúdoIntelectual>
  </Documento>
</xml>

```

Na arquitetura de metadados proposta, um modelo é constituído por Tipos que são instâncias de MetaTipos definidos na camada de meta-modelagem. No modelo definido anteriormente, por exemplo, observa-se que os TipoDeCategorias “Documento” e “ConteúdoIntelectual” são instâncias do MetaTipoDeCategoria “RecursoDublinCore”. Os TipoDePropriedades “*subject*”, “*description*”, “*title*” e “*creator*”, por sua vez, são instâncias do MetaTipoDePropriedade “ElementoDublinCore” e, por fim, “*scheme*” é instância do MetaTipoDePropriedade “QualificadorDublinCore”.

Na camada de meta-modelagem, a correta interpretação dos conceitos “RecursoDublinCore”, “ElementoDublinCore” e “QualificadorDublinCore” pode ser obtida através da referência a um *meta-modelo*. De forma semelhante a um modelo, um meta-modelo *referencia* um conjunto de MetaTipos (MetaTipoDeCategorias e MetaTipoDePropriedades) e *declara* as relações (denominadas meta-relações, nesta camada) que são válidas entre esses MetaTipos.

Um exemplo de um meta-modelo é apresentado a seguir:

```

<xml>

  <MetaModelo>

    <MetaTipoDeCategoria id="RecursoDublinCore">
      <nome>RecursoDublinCore</nome>
      <descrição>representa um recurso de informação qualquer que pode ser descrito pelos
elementos do padrão Dublin Core</descrição>
      <instânciaDe href="#Category"/>
    </MetaTipoDeCategoria>

    <MetaTipoDePropriedade id="ElementoDublin">
      <nome>ElementoDublin</nome>
      <descrição>representa um determinado elemento do padrão Dublin Core</descrição>
      <instânciaDe href="#PropertyType"/>
    </MetaTipoDePropriedade>

    <MetaTipoDePropriedade id="QualificadorDublinCore">
      <nome>QualificarDublinCore</nome>
      <subClasseDe href="#ElementoDublinCore"/>
      <instânciaDe href="#PropertyType"/>
      <descrição>representa um qualificador de elementos do padrão Dublin Core
</descrição>

```

```

</MetaTipo>

<MetaRelação>
  <descriptorFonte href="#QualificadorDublin"/>
  <arcoDeRelacionamento href="#domain" />
  <descriptorDestino href="#ElementoDublin" />
</MetaRelação>

<MetaRelação>
  <descriptorFonte href="#ElementoDublinCore"/>
  <arcoDeRelacionamento href="#domain" />
  <descriptorDestino href="#RecursoDublinCore" />
</MetaRelação>

</MetaModelo>

</xml>

```

Observando o meta-modelo definido anteriormente, verifica-se que os MetaTipos que o compõem são instâncias dos MetaMetaTipos “*Category*” e “*PropertyType*” que são definidos na camada de meta-meta-modelagem. A correta interpretação desses conceitos pode ser obtida através da referência a um *meta-meta-modelo*, que referencia um conjunto de MetaMetaTipos e declara as relações (denominadas *meta-meta-relações*, nesta camada) válidas entre esses MetaMetaTipos.

Um exemplo de um meta-meta-modelo é apresentado a seguir.

```

<xml>

<MetaMetaModelo>

<MetaMetaTipoDeCategoria id="Category" >
  <nome>Category</nome>
  <descrição>representa a super-classe para todas as categorias criadas segundo a
arquitetura MCF</descrição>
</MetaMetaTipoDeCategoria>

<MetaMetaTipoDePropriedade id="PropertyType" >
  <nome>PropertyType</nome>
  <descrição>representa a super-classe para todas as propriedades criadas segundo a
arquitetura MCF</descrição>
</MetaMetaTipoDePropriedade>

<MetaMetaTipoDePropriedade id="Domain" >
  <nome>Domain</nome>
  <descrição>representa a propriedade “domain” da MCF</descrição>
</ MetaMetaTipoDePropriedade>

```

```

<MetaMetaRelação>
  <descriptorFonte href="#PropertyType"/>
  <arcoDeRelacionamento href="#domain" />
  <descriptorDestino href="#Category" />
</MetaMetaRelação>

</MetaMetaModelo>

</xml>

```

Conforme apresentado na Figura 7.4, os meta-meta-modelos definidos na camada de meta-meta-modelagem são empregados para descrever os meta-modelos definidos na camada de meta-modelagem que, por sua vez, descrevem o modelos definidos na camada de modelagem. Esses modelos são empregados para descrever os objetos do domínio de informação do usuário.

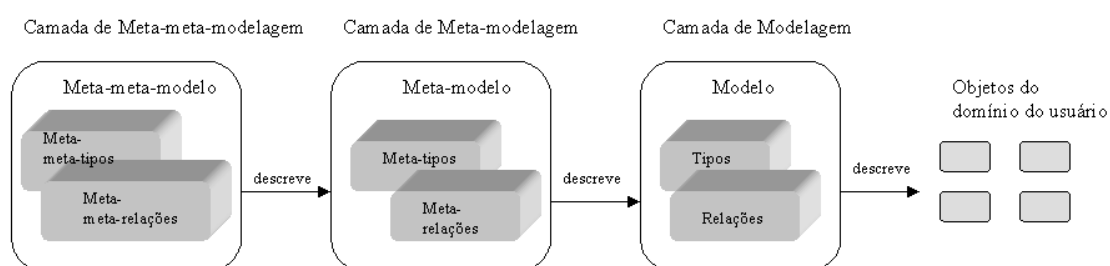


FIGURA 7.4 - Componentes da Arquitetura de Metadados Proposta

De uma forma genérica, na arquitetura de metadados proposta Tipos, MetaTipos e MetaMetaTipos constituem uma linguagem padrão para a definição de modelos, meta-modelos e meta-meta-modelos.

7.3 - PROPOSTA DE UM AMBIENTE PARA INSTANCIÇÃO DO MODDE

Na Seção 7.2 foram apresentados os principais componentes da arquitetura de metadados proposta para a descrição das propriedades de um documento em cada camada da hierarquia de objetos digitais definida no Capítulo 6. Seu enfoque é o compartilhamento de descritores oriundos de padrões distintos que podem ser empregados na definição de esquemas descritivos nas diferentes camadas da arquitetura. Não foi

especificado, porém, como os componentes dessa arquitetura, após instanciados, podem ser armazenados ou transmitidos para a sua utilização na Web.

O propósito desta seção é a definição de três ambientes distintos para uma futura implementação do MODDE em um ambiente de banco de dados. Seus componentes, apresentados na Figura 7.5 e descritos nas subseções seguintes, são baseados em conceitos definidos pelas arquiteturas Warwick e de Kahn e Wilensky. A interface é o componente a partir do qual o usuário tem acesso às funções distintas que são providas por cada ambiente.

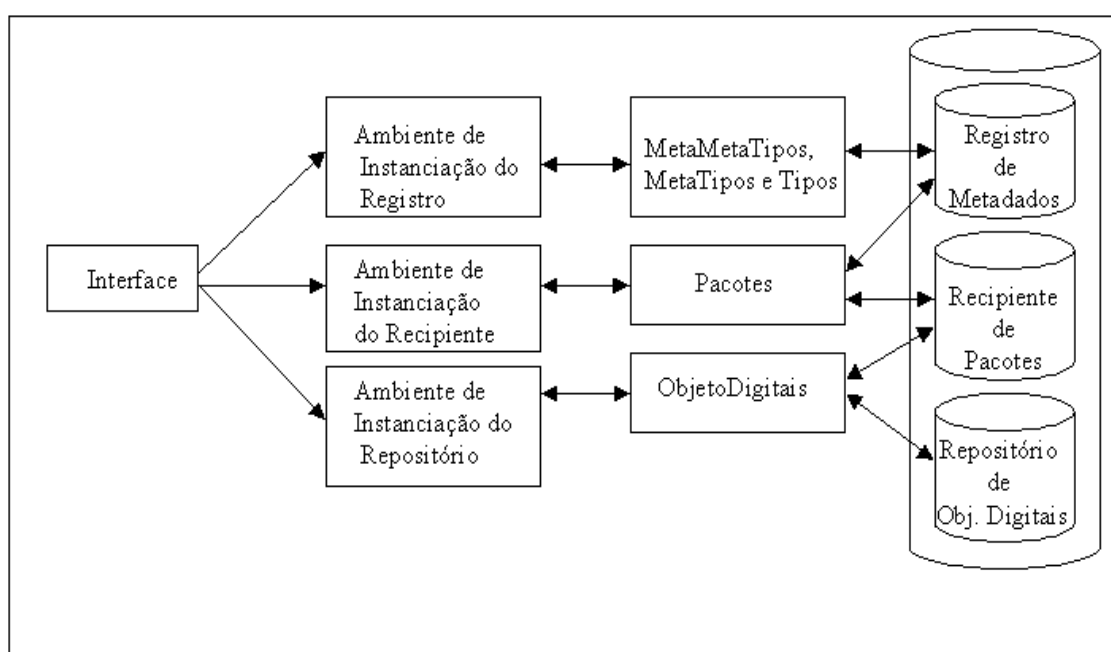


FIGURA 7.5 - Ambiente de Instanciação do MODDE

7.3.1 - Ambiente de Instanciação do Registro de Metadados

No ambiente proposto, *registro de metadados* (BARGMEYER, 1997) é o componente de uma arquitetura de metadados que provê funções para a declaração, armazenamento e recuperação de MetaMetaTipos, MetaTipos e Tipos. A sua principal finalidade, no entanto, é prover informações a respeito da *semântica* desses descritores. As classes componentes do repositório, baseadas no sistema inicial definido na Seção 7.2.2 para a arquitetura de metadados proposta, são apresentadas na Figura 7.6.

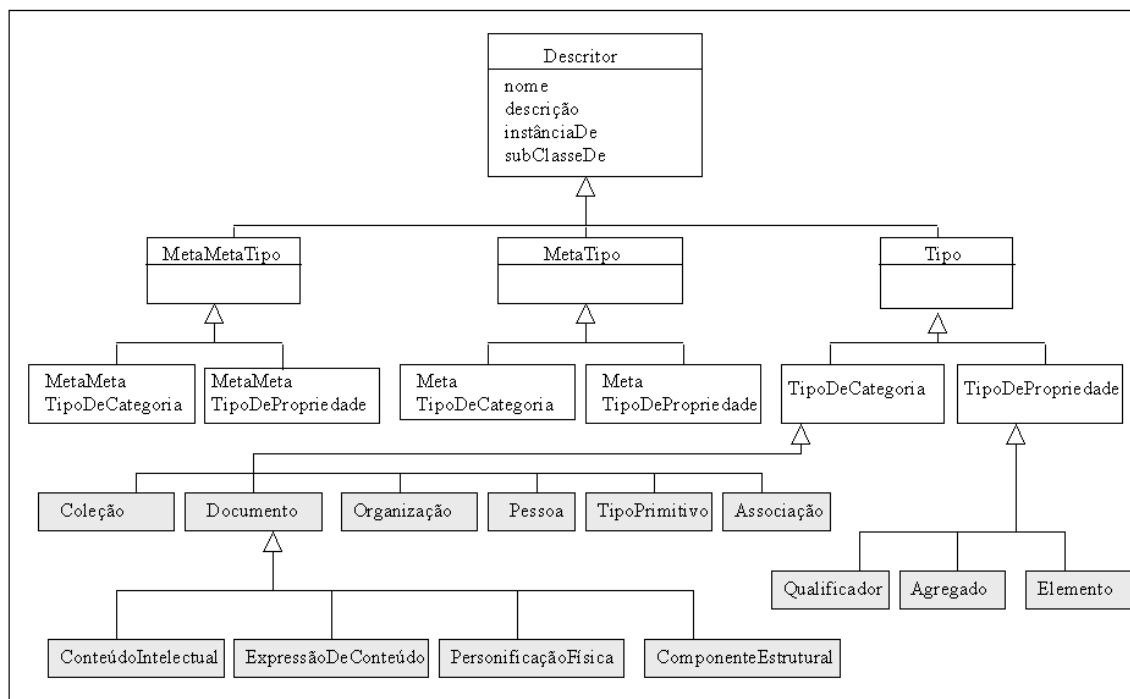


FIGURA 7.6 - Classe Descritor

7.3.2 - Ambiente de Instanciação do Recipiente de Pacotes

O recipiente de pacotes é o componente do ambiente proposto que provê funções para a definição e instanciação de meta-meta-modelos, meta-modelos e modelos a partir de descritores (MetaMetaTipos, MetaTipos e Tipos) selecionados do componente registro de metadados.

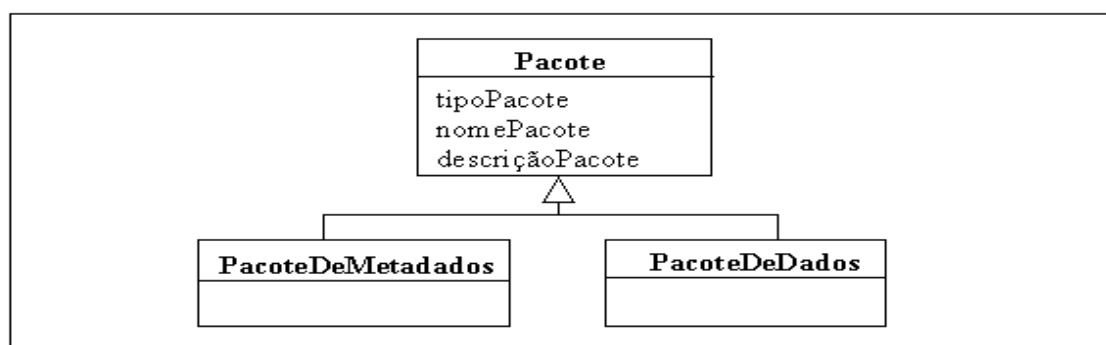


FIGURA 7.7 - Classe Pacote

Na hierarquia de objetos digitais apresentada na Figura 6.5, verifica-se que os atributos *recipienteDeDados* e *recipienteDeMetadados* de alguns objetos digitais fazem referência a uma estrutura denominada *Pacote* (mais especificamente, *PacoteIndireto* e *PacoteDeDados*). De acordo com a arquitetura Warwick, essa estrutura possibilita o tratamento modularizado do dado e do metadado em um ambiente heterogêneo e descentralizado como a Web. No ambiente proposto, Pacote é super-classe de PacoteDeDados e PacoteDeMetadados, apresentadas na Figura 7.7.

Os atributos da classe Pacote são: *nomePacote*, que provê um conjunto de caracteres para nomear o Pacote; *descriçãoPacote*, que provê um texto livre para a descrição do conteúdo do pacote; e *tipoPacote*, que especifica se o Pacote é do tipo PacoteDeMetadados (apresentada na Figura 7.8) ou do tipo PacoteDeDados (apresentada na Figura 7.9).

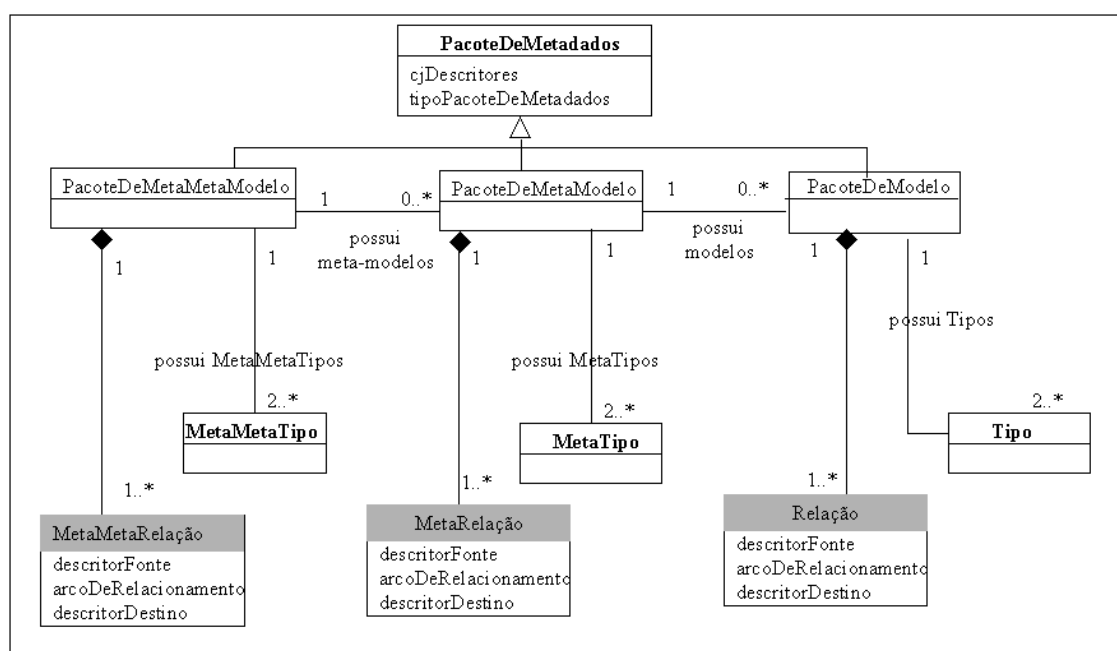


FIGURA 7.8- Classe PacoteDeMetadados

A classe PacoteDeMetadados provê o ambiente para a instanciação de meta-meta-modelos, meta-modelos e modelos. É super-classe de PacoteDeMetaMetaModelo, PacoteDeMetaModelo e PacoteDeModelo, conforme descrito a seguir:

- **PacoteDeMetaMetaModelo:** provê funcionalidade para a definição de um meta-meta-modelo. O atributo *cjDescritores* referencia os MetaMetaTipos (selecionados a partir do registro de metadados) que irão compor o meta-meta-modelo em questão. Os

relacionamentos válidos entre esses MetaMetaTipos são declarados por um conjunto de estruturas denominadas **MetaMetaRelação** (apresentada em negrito na Figura 7.8). Uma MetaMetaRelação é uma tripla da qual fazem parte os seguintes atributos: *descriptorFonte*, que referencia o MetaMetaTipo que é origem do relacionamento; *arcoDeRelacionamento*, que referencia o *MetaMetaTipoDePropriedade* que descreve o relacionamento; e *descriptorDestino*, que referencia o MetaMetaTipo que é destino do relacionamento. No ambiente proposto, PacoteDeMetaMetaModelo é empregado como um “template” para a definição de um PacoteDeMetaModelo;

- **PacoteDeMetaModelo:** provê funcionalidade para a definição de um meta-modelo. O atributo *cjDescritores* referencia os MetaTipos (selecionados a partir do componente registro de metadados) que irão compor o meta-modelo em questão. Os relacionamentos válidos entre esses MetaTipos são declarados por um conjunto de estruturas denominadas **MetaRelação** (apresentada em negrito na Figura 7.8). Uma MetaRelação é uma tripla da qual fazem parte os seguintes atributos: *descriptorFonte*, que referencia o MetaTipo que é origem do relacionamento; *arcoDeRelacionamento*, que referencia o *MetaMetaTipoDePropriedade* que nomeia o relacionamento; e *descriptorDestino*, que referencia o MetaTipo que é destino do relacionamento. Um PacoteDeMetaModelo é uma instância de um PacoteDeMetaMetaModelo. No ambiente proposto, um PacoteDeMetaModelo é empregado como um “template” para a criação de diversos PacoteDeModelos na camada de modelo;
- **PacoteDeModelo:** provê funcionalidade para a definição de um modelo. O atributo *cjDescritores* referencia os Tipos (constantes do registro de metadados) que serão empregados para compor o modelo em questão. Cada relacionamento válido entre Tipos constantes do modelo em questão é declarado por uma estrutura denominada **Relação** (apresentada em negrito na Figura 7.8). Cada Relação é uma tripla da qual fazem parte os seguintes atributos: *descriptorFonte*, que referencia o Tipo que é origem do relacionamento; *arcoDeRelacionamento*, que referencia o *MetaMetaTipoDePropriedade* que nomeia o relacionamento; e *descriptorDestino*, que referencia o Tipo que é destino do relacionamento. Um PacoteDeModelo representa uma instância de um PacoteDeMetaModelo. No ambiente proposto, um PacoteDeModelo é empregado para descrever a estrutura da informação na camada do usuário (um documento, uma imagem, etc.).

A classe *PacoteDeDados*, por sua vez, apresentada na Figura 7.9, provê funcionalidade para o armazenamento das propriedades instanciadas de um documento ou de uma coleção de documentos. *PacoteDeDados* referencia a instância da classe *PacoteDeModelo* (ou “template”) que contém a descrição semântica de todas as propriedades constantes do *PacoteDeDados*, seus domínios e as faixas de valores possíveis. *PacoteDeDados* é super-classe de *PacoteReceptivo* e *PacoteDeConteúdo*, descritas a seguir.

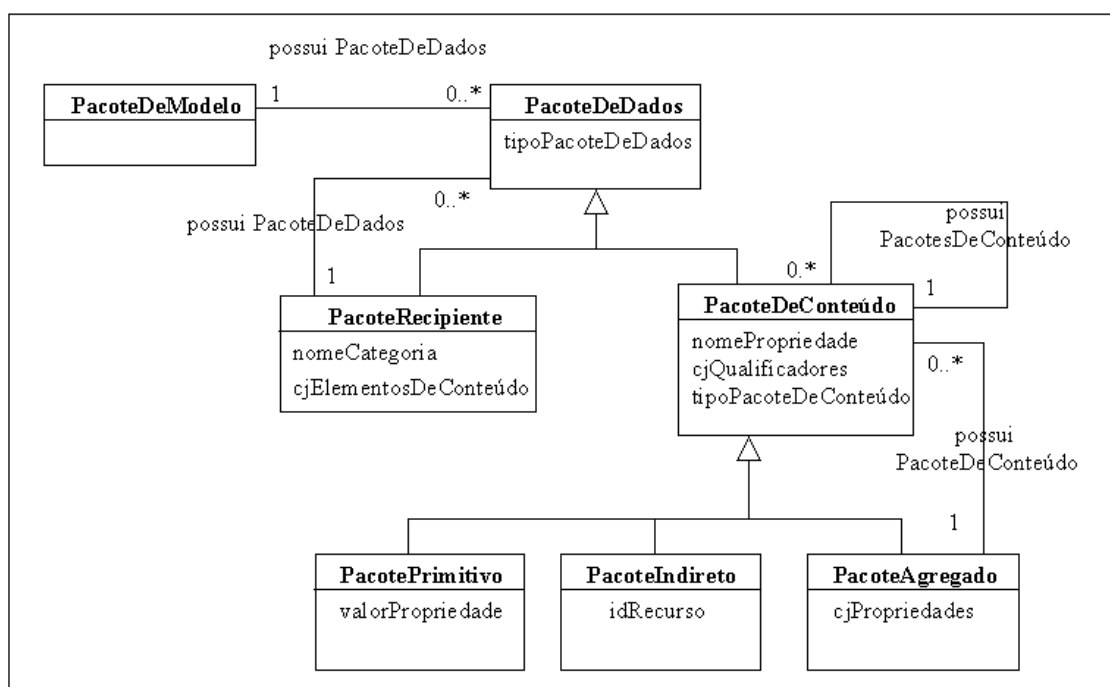


FIGURA 7.9 - Classe PacoteDeDados

- **PacoteReceptivo:** é um tipo de *PacoteDeDados* que armazena um conjunto de propriedades que descrevem uma determinada categoria de recurso de informação (por exemplo: um documento, uma coleção de documentos, as abstrações conteúdo intelectual, expressão de conteúdo, personificação física ou os componentes estruturais do documento). O atributo *nomeCategoria* identifica o nome do *TipoDeCategoria* (constante do registro de metadados) que nomeia a classe de recurso descrita pelo *PacoteReceptivo*. O atributo *cjElementosDeConteúdo* referencia, recursivamente, um conjunto de *PacoteDeDados*. Um exemplo de *PacoteReceptivo* descrevendo o recurso (ou *TipoDeCategoria*) “ConteúdoIntelectual” é apresentado a seguir:

<xml>

```

<PacoteRecipiente>
  <ConteúdoIntelectual>
    <creator>Simone Garcia</creator>
    <title>Modelo de Metadados para Imagem</title>
    <description>Este trabalho provê um ....</description>
    <subject scheme = "LCCH" >metadados, arquiteturas e
      padrões</subject>
  </ConteúdoIntelectual>
</PacoteRecipiente>
</xml>

```

- **PacoteDeConteúdo:** é um tipo de PacoteDeDados que associa um nome de propriedade (autor, por exemplo) a um valor (“Cassia Barreto”, por exemplo). Seus atributos são: *nomePropriedade*, que referencia o TipoDePropriedade constante do registro de metadados que nomeia a propriedade que está sendo instanciada; *cjQualificadores*, cujo valor corresponde, recursivamente, a um conjunto de PacoteDeConteúdo, cada qual fornecendo informações adicionais sobre o valor da propriedade referenciada pelo atributo nomePropriedade (tal como o atributo “scheme” no padrão Dublin Core, qualificando o elemento “subject” no exemplo acima). PacoteDeConteúdo pode ser do tipo:

- **PacotePrimitivo:** é um tipo de PacoteDeConteúdo cujo valor atribuído ao atributo valorPropriedade corresponde a um tipo primitivo de dado. Um exemplo de um PacotePrimitivo instanciado é apresentado a seguir:

```

<xml>
  <PacotePrimitivo>
    <autor>Cassia Maria Barreto</autor>
  </PacotePrimitivo>
</xml>

```

- **PacoteAgregado:** é um tipo de PacoteDeConteúdo cujo atributo cjPropriedades referencia, recursivamente, um conjunto de *PacoteDeConteúdo*. PacoteAgregado pode ser usado para armazenar valores de propriedades que agregam outras propriedades. A propriedade endereço, no exemplo a seguir, agrega outras propriedades tais como “rua”, “número” e “bairro”. Um exemplo de PacoteAgregado instanciado é apresentado a seguir:

```

<xml>
  <PacoteAgregado>
    <endereço>
      <rua>Princesa Isabel</rua>
      <número>121 apto 175</numero>
    </endereço>
  </PacoteAgregado>
</xml>

```

```

        <bairro>Copacabana</bairro>
    </endereço>
</PacoteAgregado>
</xml>

```

- **PacoteIndireto:** é um tipo de PacoteDeConteúdo que implementa uma referência indireta (através do atributo *idRecurso*) a um outro recurso de informação que pode estar localizado em qualquer local da Web. Um exemplo de um PacoteIndireto instanciado é apresentado no exemplo a seguir:

```

<xml>
  <PacoteIndireto>
    <editor href="http://www.ime.org/homepage.html"/>.
  </PacoteIndireto>
</xml>

```

7.3.3 - Ambiente de Instanciação do Repositório de Objetos Digitais

Repositório é o componente do ambiente proposto que provê funções para a definição, instanciação e recuperação de objetos digitais. Basicamente, a instanciação de um objeto digital consiste na:

- seleção do tipo de objeto (coleção, intelectual, informação, instância, estrutural ou associação) que se deseja instanciar;
- instanciação dos atributos específicos ao objeto selecionado;
- instanciação do atributo *recipienteDeDados*, comuns a todos os tipos de objetos digitais, conforme descrito a seguir:
 - (a) *objetoAssociação*: seu atributo *recipienteDeDados* referencia um conjunto de PacoteDeDados. Esses pacotes são selecionados a partir do componente Recipiente e correspondem a modelos instanciados. Os dados contidos nesses pacotes estão no papel de “metadado” para o objeto digital que referencia o objeto associação em questão;
 - (b) *objetoColeção*: seu atributo *recipienteDeDados* referencia, recursivamente, um conjunto de objetos digitais do tipo ObjetoContextual. Esse tipo de objeto é selecionado a partir do componente Repositório;
 - (c) *objetoIntelectual*: seu atributo *recipienteDeDados* referencia um conjunto de objetos digitais do tipo ObjetoDeExpressão. Esse tipo de objeto é selecionado a partir do componente Repositório;

- (d) *objetoDeExpressão*: seu atributo *recipienteDeDados* referencia um conjunto de objeto digitais do tipo *ObjetoPersonificação*. Esse tipo de objeto é selecionado a partir do componente *Repositório*;
- (e) *objetoDeComposição*: seu atributo *recipienteDeDados* referencia um conjunto de objeto digitais do tipo *ObjetoEstrutural*. Esse tipo de objeto é selecionado a partir do componente *Repositório*;
- (f) *objetoInstância*: seu atributo *recipienteDeDados* referencia um conjunto de *PacoteIndireto*. Esse tipo de pacote é obtido a partir do componente *Recipiente*;
- (g) *objetoEstrutural* (do tipo composto): seu atributo *recipienteDeDados* referencia um conjunto de *ObjetoEstrutural* do tipo *Composto*. Esse tipo de objeto é selecionado a partir do componente *Repositório*; e
- (h) *objetoEstrutural* (do tipo item): seu atributo *recipienteDeDados* referencia um conjunto de pacotes do tipo *PacoteIndireto*. Esses pacotes são obtidos a partir do componente *Recipiente*.
- instanciação do atributo *recipienteDeMetadados* para todos os objetos digitais definidos para o MODDE consiste na seleção dos objetos do tipo *ObjetoAssociação* a partir do componente *Repositório*.

7.4 – UTILIZAÇÃO DO MODDE NO AMBIENTE DA WEB

Na seção anterior foram descritas várias estruturas de dados projetadas especificamente para uma futura implementação do MODDE em ambiente de banco de dados, uma vez que a estrutura do metadado o transforma em um excelente candidato para ser consultado e gerenciado através das técnicas implementadas nesse ambiente.

Para o uso imediato do modelo na Web, no entanto, é necessário torná-lo disponível segundo uma linguagem de representação de dados que possa ser interpretada pelos mecanismos de indexação automática de recursos da Web, tal como a XML, empregada para prover uma representação do MODDE segundo o padrão Dublin Core, no exemplo a seguir.

<Coleção>

<objectType>homepage</objectType>

<description>Coleção das Teses de Mestrado conduzidas no IME</description>

<subject>tese, mestrado, pos-graduação</subject>

<ConteúdoIntellectual>

<creator>Cassia Barreto</creator>

<title>Modelo de Metadados para Documentos Eletrônicos</title>

<description>Provê um modelo formal para a representação de documentos na Web</description>

<subject>metadados, arquiteturas, modelos, padrões</subject>

<Associação>

<objectType scheme = “Referência Bibliográfica”

href= “http://www.dpmm.org/warwick .html” />

</Associação>

<ExpressãoDeConteúdo>

<objectType>Tese de Mestrado</objectType>

<publisher>IME</publisher>

<description>odelo formal para a representação de documentos na Web</description>

<PersonificaçãoFísica>

<objectType>Word 7.0</objectType>

</PersonificaçãoFísica>

</ExpressãoDeConteúdo>

<ExpressãoDeConteúdo>

<objectType>Artigo</objectType>

<publisher>IEEE Metadata Conference</publisher>

<title>A Metadata Model for Describing and Representing Electronic Documents on the Web</title>

<description>Provê um modelo formal para a representação de documentos na Web</description>

<PersonificaçãoFísica>

<objectType>HTML</objectType>

</PersonificaçãoFísica>

<PersonificaçãoFísica>

<objectType>PostScript</objectType>

```

    </PersonificaçãoFísica>
  </ExpressãoDeConteúdo>
</ConteúdoIntellectual>
<ConteúdoIntellectual>
  <creator>Simone Garcia</creator>
  <title>Modelo de Metadados para Descrição de Imagens</title>
  <description> provê um modelo formal....</description>
  <subject>metadados, arquitetura, modelos, padrões<
</ConteúdoIntellectual>
</Coleção>

```

7.4.1 – Uso do MODDE como Arquivo Independente

Na Web, o modelo de objetos digitais instanciado e codificado na linguagem XML poderia ser disponibilizado em arquivos texto cujo tipo MIME fosse “*application/modde*”. Isto tornaria possível aos agentes de coleta que porventura visitassem o sítio em questão, a captura das informações contidas nesses arquivos para diversos propósitos, tais como indexação das informações, exibição automática do mapa do sítio, etc.

7.4.2 – Uso do MODDE Embutido em um Documento HTML

A utilização do MODDE embutido dentro de um documento HTML dependeria, inicialmente, da possibilidade de se embutir comandos XML dentro de um documento HTML. Como esta proposta parece ser perfeitamente viável nas próximas versões da HTML, a inclusão do modelo de objetos digitais para documentos eletrônicos dentro de um documento HTML se daria dentro de um bloco iniciando com o rótulo <XML> e terminando com </XML>. Esse bloco poderia constar em qualquer parte do documento HTML. A princípio, como a HTML despreza rótulos desconhecidos, esses comandos não causariam problemas aos atuais navegadores. Para a utilização imediata do modelo, no entanto, seria necessário o desenvolvimento de “plugins” (semelhante ao “plugin” mcf) instalados nos navegadores para a visualização adequada do conteúdo do bloco XML.

Uma outra forma de utilização do modelo seria através do rótulo <LINK> da linguagem HTML. Esse rótulo possui um atributo REL através do qual é possível especificar o relacionamento de um arquivo XML contendo o modelo instanciado com o documento em questão.

7.5 – CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada uma proposta de arquitetura de metadados para a descrição do MODDE. Seu objetivo é prover uma linguagem organizacional comum para o tratamento dos elementos distintos que podem existir segundo uma hierarquia de descrição. A arquitetura implementa um modelo extensível, multi-camadas, que se adapta facilmente à evolução dos padrões de metadados e aos novos requisitos das aplicações dos usuários. Esta abordagem possibilita a interoperabilidade entre aplicações, a tradução de um modelo de metadados em outro, e o compartilhamento de metadados oriundos de diversos domínios distintos. Seu sistema inicial é baseado nas linguagens declarativas para representação de conhecimento adotadas pelas arquiteturas MCF e Esquema RDF.

Também foi apresentado um ambiente para instanciação do MODDE. Este ambiente, baseado em conceitos das arquiteturas Warwick e de Kahn e Wilensky, provê estruturas específicas para a implementação desse modelo em um ambiente de banco de dados.

CAPÍTULO 8

CONCLUSÃO

Apesar de possuir o potencial de integrar recursos de informações, coleções e catálogos em um espaço de informação global, a Web representa, atualmente, um meio simples de acesso a textos e Figuras, fornecendo a infra-estrutura para uma conectividade básica e não apenas uma plataforma padrão para a integração de dados. Em algumas situações, níveis maiores de organização e de seleção da informação são mais apropriados.

Segundo Layman (LAYMAN et alii, 1997), para que se pudesse utilizar todo o potencial da Web seria necessário incorporar ao cenário atual o intercâmbio de dados estruturados. Essa funcionalidade transformaria a Web em uma plataforma de aplicação, podendo, no futuro, evoluir para um ambiente onde a interação com os servidores Web seja totalmente automatizada. Se hoje esses servidores fornecem dados apenas aos navegadores, no futuro poderão fornecer dados para diversas aplicações.

A abordagem descrita neste trabalho representa um passo nessa direção, na medida em que possibilita a organização coerente do conteúdo do documento e dos metadados que descrevem esse conteúdo, provendo um contexto semântico e estrutura à informação contida nos documentos da Web. Com o desenvolvimento de *softwares* específicos que possam explorar essa semântica e estrutura em diversos serviços para a Web, será possível, por exemplo, a recuperação de partes específicas do documento que porventura sejam relevantes para os usuários, uma vez que todo o conteúdo do documento pode representar mais informações do que o usuário realmente necessita. Essa verificação quanto a relevância de uma determinada porção de informação só é possível com o uso de metadados provendo um contexto para a sua correta interpretação.

O enfoque básico desta dissertação foi o estudo do metadado como requisito essencial para promover a localização efetiva dos recursos de informação, em particular os documentos eletrônicos, em um ambiente expansível, heterogêneo e distribuído como a

Web. Essa necessidade decorre do aumento exponencial de recursos na Web e da demanda por soluções mais efetivas do que aquelas implementadas pelas ferramentas de pesquisa que empregam a indexação automática do conteúdo dos recursos analisados. O que se espera é que, quando a utilização de metadados se tornar comum na Web, quer seja através dos rótulos META dos documentos HTML, coletados pelos meta-pesquisadores, ou obtidos de um repositório de metadados em separado, será possível observar um aumento significativo na efetividade dos mecanismos de indexação, pesquisa e recuperação de informações (MOURA et alii, 1999).

8.1 – PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições deste trabalho são comentadas a seguir:

- **Desenvolvimento de uma taxonomia funcional de metadados para emprego na Web:** o tema abordado por esta dissertação ainda é bastante recente. Apesar de ser amplamente utilizado em diversas áreas e aplicações, o estudo de metadados para emprego na Web teve início em 1994, a partir de uma investigação conduzida pelo comitê MARBI (MARBI, 1995) com o propósito de identificar as classes de metadados oriundas do padrão bibliotecário MARC que poderiam ser aplicadas aos recursos de informação da Web. Nesse sentido, este trabalho contribuiu com uma possível taxonomia funcional de metadados que foi obtida com base em um estudo efetuado nos componentes dos diversos padrões de metadados propostos para a Web. Essa taxonomia pode ser empregada para categorizar os descritores (ou Tipos) definidos na camada de modelagem. Apesar de não esgotar o tema, uma vez que novos requisitos estarão sempre surgindo, a organização de metadados em classes provê um mecanismo simples que possibilita expressar mapeamentos formais entre termos oriundos de padrões distintos;
- **Desenvolvimento de uma estrutura organizacional para documentos e coleções de documento na Web que é independente dos aspectos de localização e armazenamento:** segundo (LAGOZE et alii, 1999), devido as suas características mais evidentes, em particular a descentralização, qualquer estrutura projetada para organizar o conteúdo da Web deve ser flexível, extensível e independente da forma como os recursos estejam distribuídos fisicamente ou de como são gerenciados. No modelo de

objetos digitais proposto, essa estrutura é baseada em entidades abstratas que refletem uma hierarquia particular que pode ser empregada para organizar, de forma coerente e padronizada, as descrições relacionadas ao documento. Considerando o fato de que a maioria dos usuários da Web não possui o conhecimento especializado que é necessário para se produzir descrições coerentes, o modelo de objetos proposto, caso implementado, pode ser empregado em ferramentas que possam guiar o usuário na descrição de seus documentos. A abordagem adotada promove a padronização e a reutilização do metadado usado na Web e minimiza os esforços no desenvolvimento de novos padrões;

- **Estudo da arquitetura Warwick tendo em vista a sua implementação na Web:** o alto nível conceitual das arquiteturas Warwick omite diversos aspectos que são necessários à implementação de uma arquitetura de metadados para a Web. Dentre os aspectos que foram considerados no desenvolvimento do modelo de objetos digitais, pode-se citar o projeto de estruturas específicas para armazenamento e utilização do MODDE na Web, baseadas nos conceitos de pacotes e recipientes da arquitetura;
- **Emprego do conceito Objeto Digital para a representação da estrutura de um documento eletrônico ou de uma coleção de documentos na Web:** um objeto digital na arquitetura FEDORA, um protótipo da arquitetura Warwick e de Kahn & Wilenski implementada no contexto das bibliotecas digitais distribuídas, tem como metáfora uma célula. Seu núcleo principal contém dados essenciais que compõem a estrutura do objeto digital, sendo que esses dados são opacos, ou seja, não possuem semântica associada. Em torno desse núcleo existe uma camada funcional contendo uma série de componentes denominados *disseminadores* que transformam o núcleo opaco de dados do objeto digital em entidades com semântica associada tais como livros, enciclopédias, etc. Tanto o núcleo quanto os disseminadores são contidos pela cápsula do objeto digital que recebe um identificador único. No modelo de objetos digitais proposto, o enfoque é a *estrutura* do objeto digital para a representação dos principais aspectos relacionados a documentos eletrônicos. Esta abordagem é mais a apropriada para implementação na Web, uma vez que essa estrutura será representada por alguma linguagem de representação de dados de propósito geral;
- **Desenvolvimento de uma estrutura para o tratamento de relacionamentos:** um aspecto de grande relevância enfatizado neste trabalho se refere ao tratamento dos

relacionamentos que podem ocorrer entre recursos de informação considerando o ambiente de larga escala da Web. No MODDE, esses relacionamentos foram organizados segundo a hierarquia definida no modelo de objetos digitais. Os relacionamentos que ocorrem em uma camada do modelo são automaticamente herdados pelas camadas subsequentes, de forma que as referências bibliográficas relacionadas ao conteúdo intelectual de um documento são também associadas às suas expressões de conteúdo e personificações físicas. Um outro aspecto relevante se refere à definição explícita, no modelo, daqueles relacionamentos que podem ocorrer entre os componentes internos (incluindo as entidades abstratas conteúdo intelectual, personificação física, etc.) que são diretamente relacionados ao conteúdo do documento (denominados estruturais), daqueles relacionamentos com outros recursos de informação distintos (denominados contextuais); e

- **Desenvolvimento de uma arquitetura genérica para metadados que pode ser empregada para descrever diversos recursos de informação da Web, e não apenas os documentos eletrônicos:** se o consenso atual é a improbabilidade de desenvolvimento de um esquema padrão para a modelagem de todos os aspectos de um documento que possa ser aceito de forma ampla pelos usuários da Web, torna-se necessário o desenvolvimento de uma estrutura que não imponha restrições quanto ao conteúdo, representação e intercâmbio de tais esquemas. Apesar de ter sido projetada visando a produção de metadados para a descrição de documentos eletrônicos, a arquitetura de metadados proposta é flexível e extensível o suficiente para se adequar a novas aplicações de metadados. Iniciativas recentes da Web (a arquitetura RDF, por exemplo) estabelecem requisitos nesse sentido.

8.2 – SUGESTÕES PARA TRABALHOS FUTUROS

Como sugestões para trabalhos futuros em continuação a este estudo, pode-se citar:

- desenvolver uma ferramenta de metadados que implemente a arquitetura de metadados proposta. Na camada de meta-modelo, tomando como base o sistema inicial provido pela camada de meta-meta-modelagem, uma ferramenta de metadados deve prover recursos que propiciem a declaração de metadados oriundos de diversas abordagens descritivas. Na camada de modelo, a ferramenta de modelagem deve propiciar a

definição de modelos para a descrição dos objetos de interesse do usuário, com base nos meta-modelos definidos na camada de meta-modelagem;

- desenvolver uma ferramenta de metadados voltada para o usuário final da Web que deseja descrever os seus recursos de informação com base nos modelos existentes a fim de que essas descrições sejam capturadas pelos mecanismos de indexação da Web;
- estender o modelo de objetos digitais para contemplar outros tipos de recursos de informação além dos documentos eletrônicos;
- promover um estudo de viabilidade no sentido da utilização do modelo proposto para o desenvolvimento de bibliotecas digitais;
- implementar o ambiente de instanciação e utilização do MODDE em ambiente de banco de dados; e
- definir uma sintaxe mais apropriada, baseada na linguagem XML, com ênfase na utilização do modelo em aplicações da Web.

Dentre as vantagens que podem ser introduzidas com uma futura implementação do modelo proposto no ambiente da Web, pode-se citar:

- possibilidade de indexação a nível de recursos individuais (e não de páginas), segundo um contexto de informação bem definido, uma vez que todo o metadado necessário a descrição do conteúdo do documento se encontra embutido junto a esse conteúdo na estrutura provida pelo objeto digital;
- se implementado pelas ferramentas de pesquisa da Web, proporcionará um aumento na qualidade dos resultados recebidos pelo usuário por ocasião de uma consulta. Metadados são empregados, nesse contexto, como um mecanismo de suporte ao acesso eficiente da informação apropriada;
- se implementado pelas ferramentas de folheio, possibilitará a construção de visões hierárquicas dos dados da Web, que podem ser oferecidos aos usuários na forma de uma interface organizada segundo diversos critérios, independente da localização física do recurso; e
- possibilidade de se efetuar consultas campo a campo (autor, assunto, etc.) a partir de formulários de pesquisas específicos.

REFERÊNCIAS BIBLIOGRÁFICAS

- AACR2 - “Anglo-American Cataloguing Rules”, Second Edition, Revision, 1998.
- ANDERSON, J. T. & STONEBRAKER, M. - “Sequoia 2000 Metadata Schema for Satellite Images”, *Sigmod Record* 23, 4, 42-48, 1994.
- ARMS, William Y., “An Architecture for Information in Digital Libraries”, (www.dlib.org/dlib/february97/cnri/02arms1.html), fevereiro 1997.
- BARGMEYER, B. – “Joint Workshop on Metadata Registries – Workshop Report”, (<http://www.lbl.gov/ölken/EPA/Workshop/report.html>), julho 1997.
- BEARMAN, David -, “Metadata Requirements for Evidence”, (<http://www.sis.pitt.edu/~nhprc/>), 1996.
- BEARMAN, D. et alii - “A Common Model to Support Interoperable Metadata”, (<http://www.dlib.org/dlib/january99/bearman/01bearman.html>), Janeiro, 1999.
- BECKETT, D. J. - “IAFA Templates in use as Internet Metadata”, (<http://www.hensa.ac.uk/tools/www/iafataools/paper/paper.html>), 1995.
- BERNERS-LEE, Tim - “Metadata Architecture”, (<http://www.w3.org/Metadata/>), 1994.
- BIBTEX - “The BibTeX Format”, (<http://www2.ecst.csuchico.edu/~jacobsd/bib/formats/bibtex.html>).
- BOHM, Klaus & RAKOW, T.C. - “Metadata for Multimedia Documents”, *Sigmod Record* 23, 4, 21-26, 1994.
- BOWMAN, C. M. et alii - “Scalable Internet Resource Discovery: Research Problems and Approaches”, *Communications of the ACM* 37, 8, pp 98-107, 1994.
- BOWMAN, C.M. & CAMARGO, W. – “Digital Libraries – The Next Generation in File System Technology”, (<http://www.dlib.org/dlib/february98/bowman/02bowman.html>), fevereiro 1998.
- BURNARD, Lou - “The Text Encoding Initiative Guidelines”, (<http://www.uic.edu/orgs/tei>), 1994.
- BURNARD, Lou et alii - “A Syntax for Dublin Core Metadata: Recommendations from the Second Metadata Workshop”, (<http://users.ox.ac.uk/~lou/wip/metadata.syntax.html>), 1996.

- CDIF – CASE Data Interchange Format, (<http://www.eigroup.org/cdif/overview/TransferFormat.html>).
- CHAN, L. M. et alii – “Dewey Decimal Classification: A Practical Guide, 2nd edition, OCLC Forest Press, Albany NY, 1996.
- DANIEL, Ron. Jr. – “Relationship Registry”, (<http://www.acl.lanl.gov/~rdaniel/IIA/relationshipRegistry.html>), 1997.
- DANIEL, Ron. Jr. & LAGOZE, Carl. - “Extending the Warwick Framework – from Metadata Containers to Active Digital Objects”, *D-Lib Magazine*, (<http://www.dlib.org/dlib/november97/daniel/11daniel.html>), novembro 1997.
- DEERWETER, S. et alii - “Indexing by Latent Semantic Analysis”, *Journal of the American Society for Information Science*, 41(6):391-407, 1990.
- DEMPSEY, L. - “Roads to Desire”, *D-Lib Magazine*, (<http://www.dlib.org/dlib/july96/07dempsey.html>), agosto 1996.
- DESIRE, “The DESIRE Project”, (<http://www.nic.surfnet.nl/surfnet/projects/desire/>).
- DEUTSCH, P. et alii - “Publishing Information on the Internet with Anonymous FTP”, (<http://info.Webcrawler.com/mak/projects/iafa/iafa.txt>) , 1995.
- EXCITE – (<http://www.excite.com>)
- FALTSTROM, Peter et alii. - “Architecture of the Whois++ Service”, (<http://www.ietf.org/internet-drafts/draft-ietf-asid-whoispp-02.txt>), 1998.
- FERBER, R. - "Hypertext and Metadata", (<http://www-cui.darmstadt.gmd.de/mind/>), 1997.
- FGDC - Content Standard for Digital Geospatial Metadata, Federal Geographic Data Committee, (<http://geochange.er.usgs.gov/pub/tools/metadata/standard/metadata.html>).
- GAYNOR, Edward – “From MARC to Markup: SGML and Online Library Systems”, (http://www.lib.virginia.edu/speccol/scdc/articles/alcts_brief.html), 1996.
- GILS - Version 2 of “Application Profile for the Government Information Locate Service”, (http://www.usgs.gov/gils/prof_v2.html), 1994.
- GOODCHILD, M. F. - “Alexandria Digital Library”, (http://alexandria.sdc.ucsb.edu/public-documents/metadata/metadata_ws.html), 1996.
- GRAVANO, L. et alii - “STARTS: Stanford Protocol Proposal for Internet Retrieval and

- Search”, (http://www.db.stanford.edu/~gravano/starts_home.html), 1997.
- GUHA, R. & BRAY, T. “Meta Content Framework Using XML”, (<http://www.w3.org/TR/NOTE-MCF-XML-970606>), 1997.
- HARDY, D.R. & SCHWARTZ, M.F. - “Essence: A Resource Discovery System Based on Semantic File Indexing”, Winter USENIX – January 25-29, San Diego, CA, 1993.
- HEERY, Rachel - “Review of Metadata Formats”, Program vol 30, Issue n° 4, preprint draft, (<http://www.ukoln.ac.uk/metadata/review.html>), 1996a.
- HEERY, Rachel - “ROADS: Resource Organisation and Discovery in Subject-based Services”, (<http://www.ukoln.ac.uk/ariadne/issue3/roads>), 1996b.
- HTML - “HTML 4.0, W3C’s next version of HTML”, (<http://www.w3.org/TR/REC-html40>).
- IANNELLA, Ricardo & WAUGH, A. - “Metadata: Enabling the Internet”, (<http://www.dstc.edu.au/RDU/reports/CAUSE97>), 1997.
- IANNELLA, Ricardo - “Mostly Metadata; A Bit Smarter Technology” (<http://archive.dstc.edu.au/RDU/reports/VALA1998>), 1998.
- IFLA – International Federation of Library Associations and Institutions – (<http://www.nlc-bnc.ca/ifla/II/metadata.htm>).
- ISBD - International Federation of Library Associations and Institutions, “ISBD(G): General International Standard Bibliographic Description: annotated text”, London: IFLA International Office for UBC, 1, 1997.
- JAIN, R. & HAMPAPUR, A. - “Metadata in Video Databases”, *Sigmod Record* 23, 4, 27-33, 1994.
- KAHN, Robert & WILENSKY, Robert - “A Framework for Distributed Object Services”, (<http://WWW.cnri.reston.va.us/home/cstr/arch/k-w.html>) , 1995.
- KASHYAP, V. et alii - “Metadata for Building the Multimedia Patch Quilt”, *Multimedia Database System: Issues and Research Directions*, Springer-Verlag, 1995.
- KERHERVÉ, Brigitte – “Models for Metadata or Metamodels for Data?” – Second IEEE Metadata Conference, Silver Spring, Maryland, (<http://computer.org/conferen/proceed/meta97/papers/bkerherve/bkerherve.html>), settembre 1997.
- KNIGHT, Jon & HAMILTON, Martin - “Dublin Core Qualifiers”, ROADS project,

- Department of Computer Studies, Loughborough University,
(<http://www.roads.lut.ac.uk/Metadata/DC-Qualifiers.html>) , 1996a.
- KOCH, T. et alii - “The Building and Maintenance of Robot Based Internet Search Services: A Review of Current Indexing and Data Collection Methods”,
(<http://www.ub.lu.se/desire/radar/reports/D3.11/>) , 1996.
- LAGOZE, Carl & DAVIS, J.R. - “Dienst - An Architecture for Distributed Document Libraries”, *Communications of the ACM* 38, 4 pp 47, 1995.
- LAGOZE, Carl et alii - “The Warwick Framework - A Container Architecture for Aggregating Sets of Metadata”, (<http://cs-tr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell/TR96-1593?abstract=>) ou
(<http://www.dlib.org/dlib/july96/lagoze/07lagoze.html>), 1996.
- LAGOZE, Carl & FIELDING, David - “Defining Collections in Distributed Digital Libraries”, *D-Lib Magazine*, Novembro, (<http://www.dlib.org/dlib/november98/lagoze/11lagoze.html>), 1998.
- LATEX – “A Document Preparation System”, Leslie Lamport, Addison-Wesley, 1986.
- LAYMAN et alii – “Specification for XML-Data”,
(<http://www.microsoft.com/standards/xml/xmldata.htm>), 1997
- LEVY, David M. - “Cataloging in Digital Order”, Proceedings of Digital Libraries 1995: The Second Annual Conference on the Theory and Practice of Digital Libraries – Austin, Texas, USA, (<http://csdl.tamu.edu/DL95/contents.html>), junho 1995.
- MARBI Discussion, Paper No . 86 - “Mapping the Dublin Core Metadata Elements to USMARC”, (<gopher://marvel.loc.gov:70/00/.listarch/usmarc/dp86.doc>) , 1995.
- MCF - “Meta Content Framework Using XML”, (<http://www.w3.org/XML/TR/NOTE-MCF-XML-970606>).
- MDIS - “Leading the Industry Initiative for Metadata Interchange”,
(<http://www.he.net/~metadata>).
- MILLER, E. J.- “An approach for Packaging Dublin Core Metadata in HTML 2.0”,
(<http://www.oclc.org:5046/~emiller/publications/metadata/minimal.html>) , 1996a.
- MILLER, E.J. - “Issues of Document Description in HTML”,
(<http://www.oclc.org:5046/~emiller/publications/metadata/issues.html>) , 1996b.
- MIME – Multipurpose Internet Mail Extensions, RFC-1521, Nathaniel S. Borenstein & Ned Freed.

MOF - Meta Object Facility Home Page, (<http://www.dstc.edu.au/Meta-Object-Facility/>).

MOURA, Ana Maria de Carvalho, Campos, Maria Luíza Machado, Barreto, Cassia Maria -
 “A Survey on Metadata for Describing and Retrieving Internet Resources”, World
 Wide Web Journal, Vol 1, Baltzer Science Publishers BV, 221-240, janeiro 1999.

NETSCAPE - (<http://home.netscape.com>).

NIDR – “CNI White Paper on Networked Information Discovery and Retrieval”,
 Incomplete Draft, Clifford Lynch, Avra Michelson, Cecilia Preston and Craig A.
 Summerhill, (<http://www.cnri.org/projects/nidr/www/toc.html>), 1994.

NORDIC - “The Nordic Metadata Project”, (<http://linnea.helsinki.fi/meta/index.html>) ,
 1996.

OCLC - Online Computer Library Center Homepage, “Building a Catalog of Internet-
 Accessible Materials”, (<http://www.oclc.org/oclc/man/catproj/catq-a.htm>).

PAYETTE, Sandra & LAGOZE, Carl - “ Flexible and Extensible Digital Object and
 Repository Architecture (FEDORA)”, European Digital Library Conference,
 (<http://www.dlib.org/dlib/may99/payette/05payette.html>), 1998.

PICCININI, Helena Serrão - “Integrando Acervos da Internet/Intranet a Banco de Dados
 Heterogêneos” , Mestrado em Informática: Ciência da Computação, Pontifícia
 Universidade Católica do Rio de Janeiro, março 1998.

PICS - “PICS: Internet Access Controls Without Censorship”, *Communications of the
 ACM* 39, 10, 1996.

PICS - (<http://www.w3.org/PICS/refcode/>), 1998.

PRICE-WILKIN, John – “Just-in-time Conversion, Just-in-case Collections”, D-lib
 Magazine, (<http://www.dlib.org/dlib/may97/michigan/05pricewilkin.html>), maio
 1997.

RDF - “Resource Description Framework Model and Syntax”, (<http://www.w3.org/TR/WD-rdf-syntax/>), 1997.

RDF SCHEMA – “Resource Description Framework Schema Specification”,
 (<http://www.w3.org/TR/WD-rdf-schema/>), 1998.

RLG - “Guidelines for Extending the Use of Dublin Core Elements to Create a Generic
 Application Integrating all Kinds of Information Resources”, (<http://www.rlg.org/metawg.html>), 1997.

RUS & SUBRAMANIAN – “Customizing Information Capture and Access”, ACM

- Transactions on Information Systems, Vol. 15, N° 1, 67-101, janeiro 1997.
- SAIF - “Spatial Archive and Interchange Format: Formal Definition”,
(<http://www.wimsey.com/~infosafe/saif/SaifHome.html>).
- SALTON, G. – “Introduction to Modern Information Retrieval”, McGraw-Hill
International Book Company, 1983.
- SEAMAN, David - “Campus Publishing in Standardized Electronic Formats – HTML and
TEI”, (<http://etext.lib.virginia.edu/articles/arl/dms-arl94.html>), 1994.
- SELIGMAN, L. & ROSENTHAL, A. - “A Metadata Resource to Promote Data
Integration” in *Proceedings of IEEE Metadata Conference*, Silver Spring, MD,
1996.
- SGML – An Introduction to the Standard Generalized Markup Language (SGML),
(http://arts.uwaterloo.ca/ENGL/courses/eng1210E/210e/assigns/4/html/c2yip/sec_2.htm).
- SHAFER, K – “The Scorpion Project”, (<http://purl.oclc.org/scorpion>), 1997.
- SHETH, Amit & KLAS, Wolfgang - “Multimedia Data Management – using Metadata
to Integrate and Apply Digital Media”, McGraw-Hill, 1998.
- SOIF - “Summary Object Interchange Format”, (<http://harvest.cs.colorado.edu/>).
- SWETLAND, A. J. G. - “Encoded Archival Description Document Type Definition
(DTD)”, final draft, (<http://scriptorium.lib.duke.edu/people/tom/guidelines/>), 1996.
- THOMPSON, Roger et alii – “Evaluating Dewey Concepts as a Knowledge Base for
Automatic Subject Assignment”, (http://orc.rsch.oclc.org:6109/eval_dc.html),
1997.
- TILLET, Barbara B. – “Cataloging Rules and Conceptual Models”, OCLC Distinguished
Seminar Series, (<http://www.oclc.org:5046/~emiller/misc/tillett.html>), janeiro 1996.
- TURING, Manfred et alii – Hypertext’91 Proceedings, 1991.
- UML – “Modelagem de Objetos Através da UML: The Unified Modeling Language”,
José Davi Furlan, MAKRON Books, 1998.
- UKOLN - “Metadata - Mapping Between Metadata Formats”,
(<http://www.ukoln.ac.uk/metadata/interoperability/>)
- URN - The URN Implementors, “Uniform Resource Names”, Progress Report, *D-Lib
Magazine*, January, (<http://www.dlib.org/dlib/february96/02arms.html>), 1996.
- USMARC - “Marbi, the USMARC Formats: Background and Principles”, American

Library Association's ALCTS/LITA/RUSA, Machine-Readable Bibliographic Information Committee, (<http://lcweb.loc.gov/marc/>), 1996.

VANCOUVER WEBPAGES - (<http://vancouver-webpages.com>), 1997.

VIZINE-GOETZ, Diane - "Using Library Classification Schemes for Internet Resources", (<http://www.uk.oclc.org/oclc/man/colloq/v-g.htm>), 1996.

WARD, N. & WOOD, A. - "Emerging Technologies for Networked Information Discovery: Beyond Z39.50", (http://www.dstc.edu/RDU/publications/e_vala98/).

WEIBEL, S. - "A Proposed Convention for Embedding Metadata in HTML", (<http://www.oclc.org:5046/~weibel/html-meta.html>), 1996.

WEIBEL, S. & MILLER, E. - "Image Description on the Internet", *D-Lib Magazine*, (<http://www.dlib.org/dlib/january97/oclc/01weibel.html>), janeiro 1997.

WEIBEL, S. et alii - "OCLC/NCSA Metadata Workshop Report", Dublin, EUA, (http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_core_report.html), 1995.

WEIBEL et alii - "The 4th Dublin Core Metadata Workshop Report", DC-4, National Library of Australia, Canberra, *D-Lib Magazine*, (<http://www.dlib.org/dlib/june97/metadata/06weibel.html>), junho 1997.

W3C – W3C Metadata Activity Home Page, (<http://www.w3c.org/Metadata/Activity.html>).

XML – "XML Overview" - (<http://sunsite.unc.edu/pub/sun-info/standards/xml/pres/9797ja/sld02000.htm>)

XML NAMES SPACES – Namespaces in XML; Bray, Hollander et alii (<http://www.w3.org/TR/1998/WD-xml-names-19980916>).

YAHOO! - (<http://www.yahoo.com>).

Z3950, "Z3950: The Search and Retrieval Protocol", (<http://lcweb.loc.gov/z3950/>).

Dissertação apresentada por

CASSIA MARIA BARRETO

e aprovada pelos Srs.:

ANA MARIA DE CARVALHO MOURA – Dr. Ing.

MARIA LUÍZA MACHADO CAMPOS – Ph.D.

RUBENS NASCIMENTO MELO – D.C.

GERALDO BONORINO XEXÉO – D.C.

CLÁUDIA MARIA OLIVEIRA – Ph.D.

IME, Rio de Janeiro, 09 de agosto de 1999.