

**Uso de Modelos Hiper­m­dia em Bibliotecas
Digitais para Dados Geogr­ficos**

Marcos Andr­e Gon­calves

Disserta­o de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Uso de Modelos Hipermídia em Bibliotecas Digitais para Dados Geográficos

Marcos André Gonçalves¹

Outubro de 1997

Banca Examinadora:

- Claudia M. Bauzer Medeiros (Orientadora)
- Daniel Schwabe
Departamento de Informática - PUC-Rio
- Maria Beatriz Felgar de Toledo
Instituto de Computação - Universidade Estadual de Campinas
- Edmundo R. M. Madeira (Suplente)
Instituto de Computação - Universidade Estadual de Campinas

¹O autor é bacharel em Ciências da Computação pela Universidade Federal do Ceará.



9802630

UNIDADE	30
N.º CHAMADA:	1/11/10/11/12
V.	Ex. 30999
COMBO BC	393/98
PROJ.	
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREGO	R\$ 11,00
DATA	10/03/98
N.º CPD	

CM-00106420-5

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Gonçalves, Marcos André

G586u Uso de modelos hipermídia em bibliotecas digitais para dados geográficos / Marcos André Gonçalves -- Campinas, [S.P. :s.n.], 1997.

Orientador : Claudia Maria Bauzer Medeiros

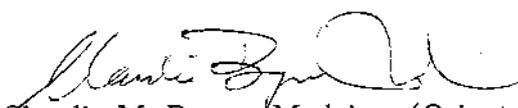
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Banco de dados orientado a objetos. 2. Sistemas de hipermídia. 3. Catálogos de bibliotecas. I. Medeiros, Claudia Maria Bauzer. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Uso de Modelos Hipermedia em Bibliotecas Digitais para Dados Geográficos

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Marcos André Gonçalves e aprovada pela
Banca Examinadora.

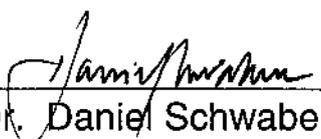
Campinas, 16 de dezembro de 1997.

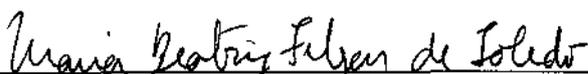


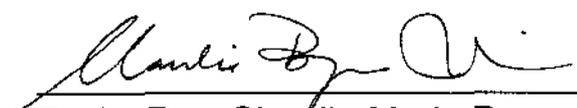
Claudia M. Bauzer Medeiros (Orientadora)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

Tese de Mestrado defendida e aprovada em 16 de dezembro de 1997 pela Banca Examinadora composta pelos Professores Doutores


Prof. Dr. Daniel Schwabe


Profa. Dra. Maria Beatriz Felgar de Toledo


Profa. Dra. Claudia Maria Bauzer Medeiros

© Marcos André Gonçalves, 1997.
Todos os direitos reservados.

*Aos meus pais, Tompson e Fanca,
fontes de admiração e inspiração maiores.*

Agradecimentos

À minha orientadora e professora, Claudia Bauzer Medeiros, por ter me recebido como orientando, por seus conhecimentos e ensinamentos valiosos, e por sua amizade e incentivo constantes.

À Cristiane Grando, cujo amor e carinho tornaram essa caminhada menos árdua e os desafios menos intransponíveis.

Aos meus familiares, pela fé inabalável em minha capacidade.

Aos companheiros do grupo de banco de dados do IC, Alexandre, Cereja, Fátima, Glaucia, Hélio Rubens, Jefferson, Juliano, Laura, Luis Mariano, Marco Antônio e Walter, pelas oportunidades de explanações e discussões que muito me enriqueceram.

Aos amigos Juliano, Luis Mariano, Glaucia e Delano, sem os quais muitos “erros” ainda permaneceriam neste trabalho.

Aos amigos Alessandra, Alexandre, Amanda, Aminadab, André, Cleidson, Cristina, Delano, Elder, Gisele, Glaucia, Guilherme, Gutemberg, Janne, Jerônimo, Karina, Laura, Lucas, Marcelo, Maria do Socorro, Marília, Mário, Nivando, Patrícia, Ralph, Roberto, Sávio, Selma, e demais que não estão aqui citados não por esquecimento, mas sim por falta de espaço, com os quais muito aprendi e compartilhei.

Aos meus professores, Ana Eliza, Azim, Claudine Chaouyia, Riverson Rios, Tarcísio Pequeno, Vânia Vidal - (Ceará)-, Célia Picinin, Claudia Bauzer Medeiros, Geovane Magalhães, Luiz Eduardo Buzato, Nelson Fonseca - (IC) -, que me proporcionaram o conhecimento e os ensinamentos necessários à minha formação e ao próprio desenvolvimento desta dissertação.

Aos amigos de República, Maria do Socorro, Nivando - (Fase 1) -, Alexandre, Cleidson, Delano - (Fase 2)- por terem aturado minhas “bagunças”.

Aos funcionários do IC, pela atenção e pronto atendimento.

Aos meus amigos residentes no Ceará, que mesmo longe, fizeram-se sentir presentes.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo suporte financeiro a esta pesquisa, consolidado em forma de bolsa de mestrado e de recursos para o projeto, e aos projetos PROTEM-GEOTEC e CEE-GEOTOOLS.

E finalmente a Deus, por estar ao meu lado em todos os momentos.

O meu mais sincero obrigado !!!

Recordação na calma da noite

Diante da minha cama, estende-se o luar.

Parece geada no chão.

Levanto a cabeça: avisto a montanha e a lua.

Torno a deitar-me. E penso na minha terra natal.

Li Po (poeta chinês)

Resumo

Esta dissertação apresenta um modelo e uma metodologia para a construção de bibliotecas digitais. Uma biblioteca digital foi aqui considerada como sendo uma aplicação hipermídia baseada em um ambiente SGBD Hipermídia Orientado a Objetos. Modelo e metodologia foram empregados na modelagem de uma aplicação específica: uma Biblioteca Digital Geográfica, cujo objetivo é colecionar e prover acesso a uma grande quantidade de dados geográficos e convencionais.

A construção desta biblioteca demandou a definição de um conjunto especial de metadados, o qual agrega diversos padrões existentes. A Biblioteca Digital Geográfica contempla dois modos de interação: folheamento (no sentido tradicional) e consulta (suportada pelo SGBD subjacente). Ambos os modos de interação são fortemente integrados no ambiente da biblioteca.

O modelo da dissertação integra o modelo do BD OOHDM de Milet et al. com o modelo de Dexter Estendido, aplicando extensões a esta integração. A metodologia estende aquela proposta pelo OOHDM, adaptando-a para permitir a modelagem de bibliotecas digitais.

As principais contribuições da dissertação são: a) uma revisão bibliográfica detalhada de requisitos de bibliotecas digitais, de modelos de dados para sistemas e para autoria de aplicações hipermídia, ambas as revisões apresentadas em uma taxonomia unificada; b) um modelo hipermídia orientado a objetos para bibliotecas digitais; c) uma metodologia que usa o modelo para construção de tais bibliotecas; d) uma especificação detalhada de como construir bibliotecas digitais geográficas, usando modelo e metodologia.

Abstract

This dissertation presents a model and a methodology for the construction of digital libraries. A digital library was here considered to be a hypermedia application, based on a Object Oriented Hypermedia DBMS environment. Model and methodology were used to model a specific application - a Geographic Digital Library, whose goal is to collect and provide access to a large volume of geographic and conventional data.

The construction of this library demanded the definition of a special set of metadata, which aggregates several existing standards. The geographic digital library contemplates two modes of interaction: browsing (in the traditional sense) and querying (supported by the underlying DBMS).

The model integrates the OOHDM database model of Milet et al, with the Extended Dexter model, and applies extensions to this integration. The methodology extends the proposal of OOHDM, adapting it to allow modelling of digital libraries.

The main contributions of the dissertation are: (a) A detailed survey of requirements of digital libraries, and of hypermedia data and authoring models, presented in a unified taxonomy; (b) An object oriented hypermedia model for digital libraries; (c) A methodology which uses the model for construction of such libraries; and (d) Detailed specification of how to build geographic digital libraries, using model and methodology.

Conteúdo

Agradecimentos	vi
Resumo	ix
Abstract	x
1 Introdução	1
2 Bibliotecas Digitais	4
2.1 Bibliotecas Digitais	4
2.1.1 Fundamentos	4
2.1.2 Tecnologia Hipermissão	5
2.1.3 Requisitos e Características de Bibliotecas Digitais	8
2.2 Dados Geográficos e Bibliotecas Digitais	15
2.2.1 Dados Geográficos	15
2.2.2 Modelagem e Tratamento de Dados e Metadados	17
2.2.3 Interface com o Usuário	19
2.2.4 Sistema de Armazenamento	20
2.2.5 Comunicação e Redes	21
2.3 Bibliotecas Digitais voltadas a Dados Geográficos	21
2.3.1 O Projeto Biblioteca Digital Alexandria	21
2.3.2 O Projeto da Biblioteca Digital da Universidade de Berkeley	23
2.3.3 O Projeto GeoScope	24
2.3.4 O Projeto EOSDIS	26
2.3.5 O Projeto do Parque Adirondack	27
2.4 Resumo	28
3 Modelos Hipermissão: Visão Geral e Análise de Requisitos para Bibliotecas Digitais	29
3.1 Introdução	29

3.2	Características Gerais de Modelos de Sistemas Hipermídia para Bibliotecas Digitais	30
3.2.1	Elementos Básicos dos Modelos	30
3.2.2	Abertura do Sistema	31
3.2.3	Suporte a Ligações Dinâmicas	33
3.3	Modelos de Sistemas Hipermídia	34
3.3.1	Modelo de Referência Dexter	34
3.3.2	Hypertext Abstract Machine (HAM)	38
3.3.3	Hyperbase	39
3.3.4	Hyperform	40
3.3.5	HB3	42
3.3.6	HyperStorm	43
3.3.7	Sistema Gerenciador de Banco de Dados OOHDM	45
3.3.8	Modelos Abertos de Ligações Dinâmicas	45
3.4	Modelos para Autoria de Aplicações Hipermídia	50
3.4.1	Hypertext Design Model (HDM)	50
3.4.2	Enhanced Object-Relationship Model (EORM)	52
3.4.3	Relationship Management Methodology (RMM)	54
3.4.4	Object Oriented Hypermedia Design Methodology (OOHDM)	58
3.5	Resumo e Conclusões do Capítulo	61
4	Modelo de Banco de Dados Hipermídia para Biblioteca Digital	64
4.1	Escolha da Metodologia e da Plataforma Base	64
4.1.1	Metodologia de Projeto de Aplicação Hipermídia	64
4.1.2	Plataforma Hipermídia	65
4.2	Modelo do Sistema Gerenciador do Banco de Dados Hipermídia	66
4.2.1	Localizador de Conteúdo e Especificador de Localização	67
4.2.2	Âncoras e Extremos	68
4.2.3	Nós e Ligações	70
4.2.4	Classe Metadados	70
4.2.5	Contexto de Navegação	71
4.2.6	Informação Contextual	76
4.2.7	Nó de Contexto	77
4.2.8	Ligações de Contexto	78
4.2.9	ADV	79
4.3	Descrição da Navegação	80
4.4	Considerações sobre o Modelo	82
4.4.1	Trabalhos Relacionados	82

4.4.2	Características Específicas do Modelo	83
4.5	Resumo	85
5	Modelagem da Biblioteca Digital Geográfica	87
5.1	Introdução	87
5.2	Modelagem da Biblioteca Digital	89
5.2.1	Projeto Conceitual da Biblioteca Digital	89
5.2.2	Projeto de Navegação da Biblioteca Digital	94
5.2.3	Projeto da Interface	105
5.3	Mapeamento para o Esquema do SGBD Hiperídia	112
5.3.1	Estrutura das Classes de Navegação	112
5.3.2	Estrutura das Ligações e Ligações de Contexto	114
5.3.3	Estrutura dos Contextos de Navegação	114
5.3.4	Estrutura das Classes Informação Contextual	120
5.3.5	Estrutura das ADVs	120
5.4	Resumo	123
6	O Ambiente de Catálogo da Biblioteca Digital Geográfica	124
6.1	Introdução	124
6.2	O Catálogo de uma Biblioteca Digital	125
6.3	Arquitetura do Catálogo	126
6.4	A Hierarquia de Classes de Metadados da Biblioteca Digital Geográfica	129
6.4.1	Classe Dublin Core	133
6.4.2	FGDC_RefEspacial	136
6.4.3	FGDC_Qualidade	137
6.4.4	Cobertura Temporal	138
6.4.5	Metadados de Geo-Classes	139
6.4.6	Metadados Convencionais	139
6.4.7	Metadado Descrição	139
6.4.8	Metadados Administrativos	139
6.4.9	Referência Indireta	139
6.5	Modelo de Consulta da Biblioteca Digital Geográfica	140
6.5.1	Folheamento da Biblioteca	140
6.5.2	Consultas	141
6.6	Considerações sobre as Classes de Metadados	147
6.6.1	Concordância com o Arcabouço Warwick	147
6.6.2	Orientação a Objetos	147
6.6.3	Trabalhos Relacionados	148
6.7	Resumo	149

7	Conclusões e Extensões	151
7.1	Conclusões	151
7.2	Extensões	152
7.2.1	Extensões ao Modelo	152
7.2.2	Extensões ao Ambiente da Biblioteca	154
7.2.3	Extensões ao Catálogo	156
	Bibliografia	158

Lista de Figuras

1.1	Organização da Dissertação	3
3.1	As Camadas do Modelo de Dexter e suas Interfaces	35
3.2	Estrutura dos Componentes na Camada de Armazenamento	37
3.3	A Arquitetura de um Sistema de Hipertexto usando HAM	39
3.4	A Hierarquia das Classes Internas de Hyperform	41
3.5	O Modelo de Dados da Arquitetura SP3	43
3.6	Visão Geral da Abordagem seguida no Projeto de HyperStorm	44
3.7	Visão Geral do Método de Projeto EORM	53
3.8	Metodologia de Projeto RMM	56
4.1	Modelo do Banco de Dados Hipermissão	66
4.2	Classes Localizador de Conteúdo e Especificador de Localização	67
4.3	Classe Âncora	68
4.4	Classe Extremo	69
4.5	Classe Nó	70
4.6	Classe Ligação	71
4.7	Classe Metadados	72
4.8	A definição da classe de visão Contexto de Navegação	75
4.9	Definição de uma classe Índice e de um exemplo de subclasse definindo índices com imagens	75
4.10	Classe Informação Contextual	76
4.11	Classe Nó de Contexto	77
4.12	Classe Ligação de Contexto	78
4.13	Classe ADV	80
4.14	Descrição da Navegação	81
5.1	Projeto Conceitual - Esquema Conceitual	92
5.2	Projeto de Navegação - Esquema das Classes de Navegação	95
5.3	Notação para a definição do Esquema de Contextos de Navegação	97
5.4	Esquema dos Contextos de Navegação da Biblioteca Digital	98

5.5	Cartão de Navegação Genérico	100
5.6	Exemplo do cartão de contexto Geo-Classe por Localização e Tempo	100
5.7	Exemplo do contexto derivado de classe Mapa Temático por Tema	101
5.8	Exemplo do contexto derivado de ligação Geo-Classe por Versão	101
5.9	Projeto de Navegação - Os Cartões dos Contextos de Navegação	102
5.10	Classes Informação Contextual da Biblioteca	103
5.11	ADV's da Biblioteca Digital	107
5.12	Diagrama de Configuração Genérico para os nós derivados de Geo-Classe	108
5.13	ADV para as informações de uma classe Informação Contextual da aplicação	109
5.14	Diagrama de Configuração para os nós da Classe N_Metadados	110
5.15	ADVChart para a ADV genérica Geo-Classe	111
5.16	ADVChart para a ADV da Classe N_Metadados	112
5.17	A Estrutura das Classes de Navegação mapeadas para o Esquema do SGBDH	113
5.18	Hierarquia dos Contextos de Navegação da Biblioteca Digital	115
5.19	Hierarquia de Classes Informação Contextual da Biblioteca Digital Geográfica	120
5.20	A Estrutura das ADV's da Biblioteca Digital Geográfica	121
5.21	A Definição da classe ADV_Imagem	122
6.1	A Arquitetura do Catálogo Digital	127
6.2	Os Serviços do Catálogo	128
6.3	A Hierarquia de Classes de Metadados	132
6.4	A Classe Dublin Core	135
6.5	Classe FGDC_RefEspacial	137
6.6	Classe FGDC_Qualidade	137
6.7	Classe Cobertura Temporal	138
6.8	Exemplo de Visualizador de Mapas	145
6.9	Exemplo de Formulário de Tempo	146

Capítulo 1

Introdução

O objetivo desta dissertação é integrar três tecnologias: a de bibliotecas digitais, a de bancos de dados geográficos e a de hipermídia. Particularmente, o trabalho é voltado à especificação de uma biblioteca digital para dar acesso a dados geográficos. A solução é baseada na definição da biblioteca sobre um banco de dados orientado a objetos e na extensão de modelos, mecanismos e metodologias para hipermídia. Estas extensões objetivam dar suporte à modelagem da biblioteca digital, permitindo o folheamento e a consulta aos dados. Um outro componente da solução inclui a definição de um esquema de catálogo eletrônico, semelhante ao catálogo de bibliotecas tradicionais, a fim de permitir buscas eficientes e rápido acesso aos dados disponibilizados pela biblioteca.

Bibliotecas Digitais oferecem um arcabouço para criar, armazenar, processar, organizar, selecionar e distribuir informação sob forma digital. A natureza desse tipo de informação é essencialmente multimídia: em adição a textos, inclui imagens, vídeos, áudios, programas de animação e simulação, jogos de computador, dentre outros. A disseminação do uso da Internet e de mecanismos de navegação via WWW vem tornando o uso de bibliotecas digitais cada vez mais freqüente, motivando várias iniciativas de organização e padronização na área.

No caso de dados geográficos, a organização de bibliotecas digitais é ainda um tópico que necessita de muito desenvolvimento. De fato, o problema não é apenas o de organização de dados, mas também envolve coleta, integração e disponibilização. Dados geográficos caracterizam-se por sua heterogeneidade (tanto do ponto de vista de formatos quanto de formas e periodicidade de coleta), pelo grande volume e por necessidades específicas de manipulação e de visualização. As pessoas que os manipulam têm diferentes perfis de formação e a nomenclatura utilizada para consultar os dados varia com sua especialização e objetivos.

O desenvolvimento de modelos de dados hipermídia para bibliotecas digitais é um grande passo no sentido de solucionar o problema de recuperação e acesso eficiente aos

dados não estruturados de uma biblioteca digital. Modelos de dados hipermídia provêm, sob uma perspectiva de sistemas, mecanismos para descrever a arquitetura interna de um sistema hipermídia e para dar suporte ao desenvolvimento de aplicações. Um modelo hipermídia para o desenvolvimento de aplicações, por sua vez, introduz abstrações para descrever a informação armazenada, os mecanismos de navegação e os aspectos de interface.

Sistemas Gerenciadores de Bancos de Dados (SGBD) fornecem uma plataforma para o desenvolvimento de bibliotecas digitais que oferece funcionalidades tais como armazenamento e gerenciamento de grandes quantidades de dados, modelos de dados para estruturação do conhecimento, acesso aos dados através de consultas baseadas nos atributos, transações concorrentes, recuperação de erros e integridade.

As principais contribuições desta dissertação são:

- proposta de um modelo de dados orientados a objetos para navegação hipermídia em uma biblioteca digital;
- especialização deste modelo, do ponto de vista de classes de dados e de navegação, para bibliotecas digitais geográficas, propondo uma metodologia para esta especificação, e mapeando o modelo em um SGBDOO;
- especificação de um modelo de metadados para permitir a navegação e consulta nesta última.

A dissertação está organizada da seguinte forma. O capítulo 2 efetua um levantamento bibliográfico sobre diversos aspectos relacionados à organização e ao gerenciamento de dados em bibliotecas digitais. O estudo classifica estes aspectos segundo quatro tipos principais: interface com o usuário; modelagem e tratamento de dados; interoperabilidade e redes de computadores; e armazenamento de dados geográficos. As implicações da utilização de dados geográficos em cada uma destas categorias é analisada.

O capítulo 3 realiza um segundo levantamento bibliográfico que considera aspectos da tecnologia hipermídia, divididos em duas grandes vertentes: sistemas hipermídia e aplicações hipermídia. Do ponto de vista do sistema hipermídia, são avaliados aspectos relativos ao suporte a funcionalidades de banco de dados, à abertura do sistema e o suporte à evolução e ao crescimento da biblioteca. Do ponto de vista de modelos de autoria para aplicações hipermídia, são levadas em consideração características tais como a capacidade de modelar os dados de uma biblioteca digital e a forma de modelar os aspectos de navegação e de interface da biblioteca.

O capítulo 4 apresenta o modelo do SGDB Hipermídia que surge como resultado dos estudos efetuados. O modelo visa satisfazer os requisitos de um modelo hipermídia para

bibliotecas digitais levantados na revisão bibliográfica, adequando-o a um ambiente de biblioteca digital.

O capítulo 5 utiliza o modelo desenvolvido no capítulo 4 para modelar uma aplicação hipermídia específica utilizando uma metodologia hipermídia estendida. A aplicação é uma biblioteca digital geográfica cujo objetivo é gerenciar, coleccionar e prover acesso a uma grande quantidade de dados geográficos e de dados convencionais armazenados em rede.

O capítulo 6 define o modelo do catálogo da biblioteca digital geográfica responsável pelo suporte às consultas sobre os dados da biblioteca. O modelo do catálogo envolve o modelo de metadados e a descrição de como consultar e navegar na biblioteca.

Finalmente, o capítulo 7 discute as conclusões finais da dissertação e as direções para trabalhos futuros.

A figura a seguir mostra a organização da dissertação, indicando a contribuição de cada capítulo e a seqüência dos resultados obtidos:

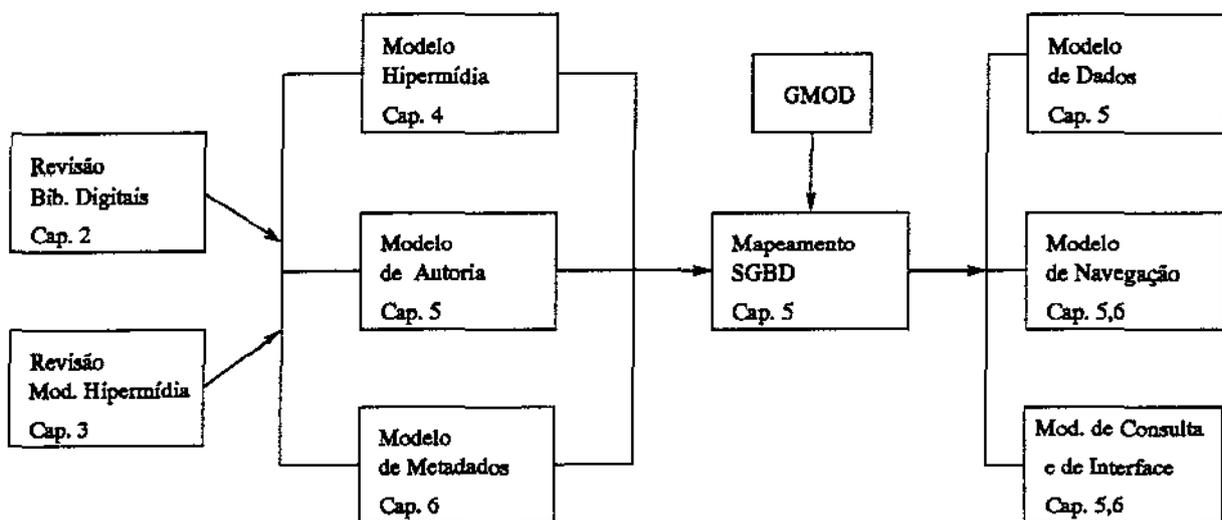


Figura 1.1: Organização da Dissertação

Capítulo 2

Bibliotecas Digitais

O termo *biblioteca digital* tem sido amplamente utilizado para designar ambientes eletrônicos que provêem aos usuários uma larga faixa de materiais e de serviços heterogêneos, cuja estrutura se assemelha à de uma biblioteca tradicional. [DA97], por exemplo, define uma biblioteca digital como um ambiente que agrega conjuntos, serviços e pessoas no suporte ao ciclo de vida completo de criação, disseminação, uso e preservação de dados, informação e conhecimento. Este capítulo faz uma revisão bibliográfica sobre os principais conceitos de bibliotecas digitais e apresenta uma taxonomia dos componentes constituintes destas bibliotecas. Os problemas estão centrados em quatro tipos principais de categoria: interface com o usuário; modelagem e tratamento de dados; interoperabilidade e redes; e, finalmente, armazenamento de dados. Além disso, o capítulo avalia o impacto do uso de dados geográficos em cada uma das categorias anteriormente citadas.

O capítulo está organizado da seguinte forma. A seção 2.1 efetua um estudo sobre os principais conceitos relativos a bibliotecas digitais. A seção 2.2 analisa estes conceitos no contexto de dados geográficos. A seção 2.3 apresenta algumas iniciativas de bibliotecas digitais geográficas que têm sido desenvolvidas. A seção 2.4 sumariza o capítulo.

2.1 Bibliotecas Digitais

2.1.1 Fundamentos

Bibliotecas Digitais devem suportar as principais funcionalidades exibidas por bibliotecas convencionais. Estas funcionalidades podem ser classificadas dentro de quatro áreas principais: coleta; organização e representação; acesso e recuperação; e análise, síntese e disseminação da informação [GBA⁺94]. Bibliotecas digitais apresentam diversas vantagens sobre bibliotecas convencionais. Dentre as vantagens, podemos citar: independência de localização; acesso a informações previamente inacessíveis; disponibilização de informações

que não são bem representadas ou efetivamente úteis fora de um ambiente digital; velocidade de acesso; possibilidades de anotação mais eficientes e sem transtornos para usuários; novas possibilidades de busca; solução do problema de circulação em relação à limitação do número de cópias.

Uma característica básica dos dados armazenados nestas bibliotecas digitais é sua interconexão, gerando a noção de “documentos hipermídia”. Ligações inter e intra documentos são mantidas por longos períodos entre localizações possivelmente distantes [BAN94].

Muitas das vantagens de bibliotecas digitais citadas anteriormente são resultado imediato da tecnologia hipermídia. Hipermídia é a ciência dos relacionamentos [BI95]. Seus conceitos tratam de meios de estruturar uma base de informação, oferecendo ao usuário acesso direto ao conteúdo e às interconexões dentro de um domínio de informação. Funcionalidades hipermídia tais como navegação (incluindo folheamento e *backtracking*), anotação e visão geral da informação são essenciais no contexto de Bibliotecas Digitais.

Bancos de Dados são outra tecnologia de fundamental importância no suporte a Bibliotecas Digitais. Trabalhos como [AFY94, AHN94, BAN94, GBA⁺94] discutem o papel desempenhado por SGBDs em bibliotecas digitais, apresentando características de sistemas de banco de dados úteis nesse contexto, tais como controle de concorrência, autorização, recuperação de falhas e persistência. Explicitam também características que devem ser modificadas, melhoradas ou adicionadas de forma a atender completamente os requisitos de funcionalidade pretendidos. Dentre as limitações de SGBDs que esbarram nos requisitos de Bibliotecas Digitais, citam-se a não interpretação dos dados, as limitações nas consultas a metadados, e a dificuldade de tratar eficientemente mudanças no esquema.

Na próxima seção, serão descritos os conceitos e as características básicas da tecnologia hipermídia. A seguir, serão discutidas mais profundamente as necessidades de bibliotecas digitais em relação aos objetivos a que elas se propõem. Conceitos e características de banco de dados estarão embutidos em ambas as discussões.

2.1.2 Tecnologia Hipermídia

Hipermídia pode ser definida como “uma abordagem para gerenciamento de informação no qual o dado é armazenado em uma rede de nós conectados por ligações” [SW88]. Nós podem conter qualquer tipo de dado multimídia, como textos, gráficos, áudio, vídeo e outras formas de informação.

A tecnologia hipermídia pode ser encarada sob várias perspectivas: pode ser vista sob uma abordagem de interface, na qual ícones ou marcadores são inseridos como ligações para facilidades de navegação; pode ser encarada como um método de banco de dados para acessar e gerenciar os dados; alternativamente, pode corresponder a um esquema de representação baseado em uma rede semântica. Em suma, um sistema hipermídia pode

ser considerado um sistema de banco de dados que provê um método particular de acessar informação [Bal93].

As características básicas de um sistema hipermídia são [Bal93]:

1. uma Interface Gráfica de Usuário, com *browsers* e diagramas de visualização, que ajudam usuários a navegar através de grandes quantidades de informação pela ativação de ligações e leitura do conteúdo dos nós;
2. mecanismos tradicionais de recuperação de informação tais como busca por palavras-chave, busca por autores, etc. Existem também propostas de incorporar consultas por estrutura a consultas por conteúdo. Estas consultas proveriam recuperação de uma parte da rede hipermídia baseado em algum critério especificado pelo usuário;
3. uma máquina hipermídia para gerenciar a informação sobre nós e ligações;
4. um sistema de armazenamento que pode ser um sistema de arquivos, uma base de conhecimento ou um SGBD.

Nós e Ligações

Segundo [Bal93], um sistema hipermídia é constituído por *nós* (conceitos) e *ligações* (relacionamentos). Um nó geralmente representa um conceito ou uma idéia. Ele pode conter textos, gráficos, animações, áudio, vídeo, imagens ou programas. Pode, além disso, possuir atributos descritivos. Nós são conectados a outros por ligações, cujo papel é conectar conceitos relacionados. O nó no qual uma ligação se origina é chamado *origem* e o nó no qual uma ligação finaliza é chamado *destino*. O conteúdo de um nó é apresentado pela ativação das ligações.

Ligações podem ser unidirecionais ou bidirecionais, facilitando, assim, trilhas inversas. Ligações também podem conter atributos descritivos como, por exemplo, especificação da ligação, elaboração da ligação, membros da ligação, ligação oposta, especificando a natureza do relacionamento.

Banco de Dados para Hipermídia

Sistemas Gerenciadores de Banco de Dados têm se mostrado úteis como mecanismos de armazenamento para sistemas hipermídia. O modelo de SGBDs orientados a objetos costuma ser preferido, pois inclui habilidade para modelar dados complexos e para representar de forma unificada o domínio do problema. Outras vantagens oferecidas pelo modelo OO são facilidades de navegação por objetos aninhados e hierarquias de classes,

conceitos de abstração de dados, encapsulamento e agrupamento através de padrões estruturais e comportamentais. O modelo OO fornece, assim, uma abordagem natural para a modelagem da estrutura de composição e ligação hipermédia.

Definiremos o conceito de *SGBD Hipermédia* (SGBDH) como sendo um Sistema Gerenciador de Banco de Dados que suporta hipermédia. Utilizando o conceito de máquina hipermédia dado em [SS90] e em [BWAH96], que a definem como sendo “uma máquina abstrata que provê uma interface para acessar e manipular as estruturas hipermédia”, um SGBDH é uma máquina hipermédia construída sobre um SGDB.

Vários requisitos são desejáveis para SGBDs Hipermédia e podem ser implementados em um SGBDOO:

1. Abertura e Distribuição: um sistema hipermédia deve ser facilmente integrável com outros sistemas de informação;
2. Suporte para Trabalho Cooperativo: hipermédia é a tecnologia mais apropriada para representar a atividade desestruturada do trabalho cooperativo e o compartilhamento de meios. Para suportar essas atividades, devem ser providas características como acesso multiusuário, mecanismos robustos de controle de concorrência, *broadcasting* de mudanças e rastreamento de contribuições feitas por elementos do grupo;
3. Integridade e Corretude dos dados: sistemas hipermédia precisam prever mecanismos especiais de integridade como, por exemplo, exclusão de referências “perdidas”, isto é, que não têm um destino válido;
4. Componente Dinâmica: um SGBDH deve dar suporte a estruturas virtuais ou dinâmicas com capacidade de reconfigurar dinamicamente a rede hipermédia para adaptação a mudanças na informação;
5. Mecanismos de Consulta e de Pesquisa: além do acesso navegacional, consultas por estrutura e conteúdo devem ser fornecidas.

Mecanismos de regras, suporte a relacionamentos *part-of*, mecanismos de versões e suporte a evolução do esquema também são características críticas de SGBDHs, que podem ser implementadas usando um SGBDOO.

Características de Sistemas de Recuperação de Informação

Navegação ou folheamento são suficientes apenas para pequenos sistemas hipermédia. Técnicas mais poderosas de busca de informação são necessárias em SGBDHs de grande porte. Consultas por conteúdo podem ser usadas para retornar o conteúdo dos nós,

enquanto consultas estruturais podem ser usadas para retornar subgrafos da rede hipermídia que casam com um dado padrão. Neste contexto, pesquisas envolvem estruturas de indexação e extensões de linguagens de consulta, tanto para executar consultas estruturais quanto para consultar dados semi-estruturados ou sem estrutura. Por dado *semi-estruturado* ou *sem estrutura* queremos designar dados que não possuem um esquema fixo, cuja estrutura pode ser irregular ou incompleta.

Sistemas de Recuperação de Informação (SRI) são sistemas que tratam de dados desestruturados como textos, imagens e áudio, e que utilizam linguagens de consulta com extensões probabilísticas, conceitos de incerteza e técnicas de casamento aproximado para recuperação de informação. Desenvolvimento de técnicas de indexação e recuperação de informação heterogênea são tópicos inerentes a SRI.

O uso de técnicas avançadas de SGBDs com SRIs vem sendo adotado cada vez mais. Exemplos são as pesquisas citadas em [QRS⁺95, Fuh96, Son96, VCC96]. [QRS⁺95] apresenta uma linguagem de consulta para recuperação de informação heterogênea semi-estruturada. A linguagem provê resultados significativos para casos nos quais alguns dados estão ausentes, quando conceitos são representados usando diferentes tipos ou quando conjuntos heterogêneos estão presentes.

[Fuh96] descreve uma linguagem de consulta para sistemas de recuperação de informação que utiliza a idéia de predicados vagos e de modelos de dados com incerteza para um sistema integrado SGBD-SRI.

[Son96, VCC96] apresentam arcabouços para integração de Sistemas de Recuperação de Informação a SGBDs. O primeiro enfatiza o projeto e a implementação de um índice-SRI e de um sistema eficiente de recuperação baseado em facilidades de otimização de um SGBD orientado a objetos. O segundo implementa um SQL relacional estendido com operadores probabilísticos para recuperação de dados desestruturados.

2.1.3 Requisitos e Características de Bibliotecas Digitais

O projeto de uma Biblioteca Digital deve se concentrar no suporte para extensões digitais dos requisitos e das funcionalidades de bibliotecas tradicionais. Dentro deste objetivo, tendo a tecnologia hipermídia como suporte, pode-se dividir o arcabouço de bibliotecas digitais em quatro grandes áreas de estudo [MG97]:

1. Interface com o Usuário;
2. Modelagem e Tratamento de Dados e Metadados;
3. Interoperabilidade e Redes;
4. Sistema de Armazenamento.

Com base nesta divisão, a discussão a seguir aborda os principais aspectos de interesse pertinentes a estrutura, requisitos, classes de aplicações e aspectos de arquitetura de uma biblioteca digital.

Há de se ressaltar que os limites entre estas áreas são muito tênues, às vezes inexistentes, havendo vários aspectos nos quais elas se interceptam. Tal divisão tem um aspecto puramente didático.

Interface com o Usuário

A interface de uma Biblioteca Digital deve apresentar as seguintes funcionalidades:

1. Consultas por Conteúdo;
2. Consulta por Folheamento;
3. Apresentação dos Dados.

Uma interface baseada em hipermídia deve permitir navegar livremente por grande quantidade de dados complexos, conectados e referenciados entre si. Contudo, redes hipermídia são sistemas complexos de redes desorganizadas, trazendo problemas em relação à sua navegação. Os principais problemas a serem tratados dizem respeito à *desorientação* e à *sobrecarga cognitiva* [Bal93]. A desorientação, ou o “se sentir perdido no espaço”, surge da necessidade de se localizar dentro da rede, de onde ir, e de como chegar em algum lugar na rede hipermídia. A sobrecarga cognitiva aparece devido à sobrecarga na tomada de decisões sobre quais ligações seguir e quais abandonar dado um grande número de escolhas.

É preciso desenvolver ferramentas para auxiliar a navegação, métodos formais para avaliar a utilidade e a adequação da interface e abordagens compreensivas e sistemáticas para projetos de interface.

Modelagem e Tratamento de Dados e Metadados

Modelagem de Dados A modelagem dos dados é de fundamental importância para bibliotecas digitais. Ela é responsável pelo suporte às facilidades de consulta e de folheamento, e pela definição do grau de independência dessas características com o sistema de armazenamento de objetos.

Um modelo de dados deve fornecer ferramentas para descrever a organização lógica de bancos de dados e para definir as operações de manipulação de dados permitidas, oferecendo, assim, uma abstração do mundo real. Um modelo de dados hipermídia provê meios para descrever informações sobre objetos e sua estrutura interna, e também para definir mecanismos de navegação. O modelo de dados hipermídia é uma boa fundação

para modelar a estrutura dos documentos armazenados em uma biblioteca digital. O modelo hipermídia da biblioteca digital deve ser flexível o suficiente para evitar lacunas entre o modelo de dados da aplicação e a estrutura dos dados do SGBDH que provê a interface para acessar e manipular as estruturas hipermídia da biblioteca.

[BWAH96] apresenta um conjunto de requisitos que devem ser atendidos por um modelo hipermídia. Um modelo hipermídia deve:

- prover um conjunto de tipos básicos de objetos hipermídia. Geralmente eles incluem nós atômicos e compostos, ligações, conteúdos, âncoras, etc. Extensões para algumas aplicações podem ser necessárias, como por exemplo ligações que podem se conectar a outras ligações;
- suportar a definição e a manutenção de restrições de integridade entre os tipos de objetos hipermídia, além de oferecer operações complexas para manipulação dos documentos hipermídia que mantêm essas restrições. Por exemplo, restrições podem evitar que objetos compostos conttenham a si próprios recursivamente ou podem impor que ligações devam sempre ter um destino válido;
- suportar sistemas abertos, ou seja, ser capaz de integrar ferramentas externas e dados externos gerados por tais ferramentas.

Metadados e Projeto de Catálogo Em Bibliotecas Digitais, é crucial suportar, de forma eficiente, acessos a dados por conteúdo. Problemas a serem tratados devido a essa última característica incluem: gerenciamento de grandes quantidades de dados; desenvolvimento de técnicas e ferramentas para descoberta e busca de dados; e compartilhamento e integração de dados estruturados, semi-estruturados ou sem estrutura.

Um aspecto chave é o uso de metadados, isto é, de dados sobre os dados. Chen et al. [CMK⁺94] definem metadados sobre dados multimídia como propriedades derivadas da mídia que são úteis para o acesso e a recuperação da informação. Dentro do contexto de Bibliotecas Digitais, [SGG96] define metadados ou meta-informação como informação armazenada em catálogos eletrônicos de bibliotecas digitais que modelam documentos e consultas de usuários cujo principal objetivo é suportar o acesso eficiente do usuário aos documentos da biblioteca. Neste mesmo contexto, [FCF⁺95] apresenta um catálogo eletrônico como sendo um banco de dados de metadados e uma máquina de pesquisa que é análoga ao sistema de catálogos de uma biblioteca padrão.

Vários são os trabalhos que têm incorporado o uso de metadados para obter funcionalidades de recuperação para as mais diferentes mídias. Citamos a seguir alguns exemplos. Em [KKH94] é apresentado um método para obtenção dinâmica de imagens através do uso de palavras-chave, armazenadas como metadados, representando o ponto de vista do

usuário e o conteúdo da imagem. A busca é executada computando equivalência semântica ou similaridades entre os metadados sobre as imagens e as palavras-chave.

[AS94] enfoca um esquema de metadados para imagens de satélites, discutindo os padrões geoespaciais utilizados e descrevendo um protótipo SQL. [JH94] explora a natureza de metadados para vídeo, com o objetivo de organizar bancos de dados de vídeo de forma a assegurar acesso por conteúdo.

[Las96] apresenta uma revisão sobre os metadados utilizados nos seis projetos DLI (Digital Library Initiative) que têm por objetivo o desenvolvimento de novas tecnologias para facilitar o acesso público aos dados da NASA via rede de computadores .

Todas essas abordagens mostram a importância do uso de metadados no gerenciamento de dados multimídia. [KS94] apresenta algumas das razões para essa importância, dentre as quais podemos citar:

- impossibilidade de fazer casamento de consulta exato em muitos casos;
- inabilidade para realizar busca por conteúdo em casos que envolvem grandes conjuntos de dados que são difíceis de analisar;
- problemas de desempenho de busca por conteúdo devido a critérios de busca específicos de modalidade ou de aplicação;
- importância de dados derivados ou interpretados, de contexto e de semântica ao tratar com dados áudio-visuais.

Um outro aspecto a ser considerado sobre metadados diz respeito à sua natureza e organização, tanto dentro de catálogos de uma biblioteca individual (intraoperabilidade de catálogo) quanto entre catálogos de diferentes bibliotecas (interoperabilidade de catálogos).

[SGG96] apresenta um arcabouço geral para suportar tanto a modelagem de documentos digitais quanto consultas de usuários em termos de metadados. Uma aplicação útil desse arcabouço seria justamente a habilidade de facilitar o projeto de modelos gerais e extensíveis de documentos, de consultas de usuário e de catálogos de bibliotecas digitais, facilitando interoperabilidade.

[BR94] apresenta uma classificação de metadados de acordo com a sua natureza e com os diferentes objetivos aos quais se propõem. A classificação é a seguinte:

1. Metadados para a Representação de Tipos de Mídias: informação adicional para a apresentação de dados multimídia, o que inclui, por exemplo, informações sobre formato, codificação, técnicas de compressão. Outro exemplo seria a definição dos metadados associados a imagens de satélites em [AS94].

2. Metadados Descritivos de Conteúdo: metadados que descrevem o conteúdo da mídia. Estes são determinados intelectualmente ou por meios de métodos semi-automáticos ou automáticos. Nos dois últimos casos, os métodos são específicos para o tipo de mídia. Exemplos incluem tamanho, posse, data de criação, etc. Esses metadados são definidos sem nenhuma análise semântica. Contudo, se for necessário a captura da informação semântica para descrever o significado ou os possíveis usos do documento, uma subdivisão deve ser efetuada nesta classificação, particionando entre metadados dependentes e independentes do domínio da aplicação. Em [KKH94], por exemplo, imagens são associadas a palavras-chave derivadas da descrição da imagem.
3. Metadados para Classificação do Conteúdo: metadados extraídos ou derivados da informação armazenada baseada no seu conteúdo. Exemplos são índices de vetores para os documentos construídos através da localização de palavras-chave dentro de um documento. As características dos dados em [JH94] também seguem essa abordagem.
4. Metadados para Composição do Documento: papéis assumidos pelos diversos componentes do documento e características dos relacionamentos entre esses componentes. Podem ser modelados por uma linguagem de estruturação de documentos, por conceitos de hipermídia ou por terminologias de modelagem semântica de dados como, por exemplo, relacionamento *partOf*.
5. Metadados para a História do Documento: dados para modelar a evolução e as mudanças através do tempo. Incluem atributos como nome do autor ou do revisor, última atualização ou um relacionamento *DerivadoDe* para modelos de versionamento.
6. Metadados para Localização do Documento: atributos para localizar documentos multimídia de forma única.
7. Metadados sobre Coleções de Documentos: metadados estatísticos. Exemplos seriam número de documentos escritos por um certo autor, frequência de documentos ou de componentes com uma certa característica na base de dados.

Padrões como SGML (Standard Generalized Markup Language) e ODA (Office Document Architecture) vêm sendo utilizados para modelar a estrutura interna ou lógica de documentos, através de metadados. Exemplos desse uso são: [KK96], que utiliza um misto dos dois padrões para dar acesso *on-line* a jornais eletrônicos; e [BAN94], que descreve um meio de armazenar documentos estruturados como documentos SGML em um banco de dados orientado a objetos.

Sistema de Armazenamento

A quantidade e o tamanho dos objetos a serem armazenados em uma biblioteca digital são potencialmente enormes. Além disso, o objetivo é fornecer essa quantidade de informação ao maior número possível de usuários, concorrente e independentemente de sua localização. Esses, entre inúmeros outros fatores, indicam a necessidade de um sistema de armazenamento distribuído.

Contudo, a utilização de um sistema de armazenamento distribuído traz consigo uma série de outros problemas. O sistema deve prover um identificador universal de objetos (OID) único para cada objeto armazenado e deve existir um esquema de nomeação global para objetos digitais. Esse esquema permitiria a um objeto ser referenciado unicamente de forma global dentro de qualquer domínio. Essa referência global faria parte do conjunto de metadados do objeto e possibilitaria a concepção de “servidores de referência” que aceitariam referências e retornariam objetos. Servidores de referência também necessitariam de um esquema de nomeação.

Tais características são de vital importância, tanto para o desenvolvimento de protocolos de acesso aos repositórios, quanto para o tratamento de questões de interoperabilidade [LGM95]. O conceito de interoperabilidade entre Bibliotecas Digitais trata de meios de facilitar o compartilhamento de recursos e de ligações, bem como da integração entre sistemas de bibliotecas digitais sem o conhecimento de detalhes de cada modelo de armazenamento empregado.

O uso de um modelo de dados hipermídia levanta algumas questões. Em primeiro lugar, deve-se utilizar um SGBDH como sistema de armazenamento para dar suporte à modelagem e à manutenção das estruturas hipermídia. Em segundo lugar, caso o modelo de dados não ofereça o suporte necessário, o sistema deve se responsabilizar pelo tratamento de questões como a manutenção de restrições definidas pelos tipos de objetos hipermídia modelados (e.g, restrições de cardinalidade, conectividade, propriedades estruturais) e pelo fornecimento de operações específicas para o tratamento da aplicação.

Por fim, deve-se garantir as propriedades de distribuição, abertura e gerenciamento da informação: armazenamento persistente; acesso multiusuário, incluindo controle de concorrência; recuperação de falhas; integração de tipos de dados multimídia; interoperabilidade; funcionalidades de consulta; tipagem de nós e ligações; e manutenção de ligações distribuídas no caso de um modelo hipermídia. A maioria dessas características pode ser tratada com a utilização de técnicas de orientação a objetos e o uso de um Sistema de Gerenciamento de Banco de dados Orientado a Objetos como sistema de armazenamento.

Um último componente a ser discutido seria o subsistema de alimentação de dados. Esse subsistema deveria tratar problemas tais como armazenamento de novos tipos de dados, extração automática de metadados e adição destes ao catálogo, além de geração automática de ligações.

[SSKS] apresenta um método para extração automática de metadados e sua respectiva associação à informação original, o que assegura a criação automática e rápida de grandes repositórios de informação. Uma hierarquia de classes abstratas é proposta para o processamento e representação dos diferentes tipos de informação e para suportar naturalmente a inclusão de novos tipos.

Interoperabilidade e Redes

Sistemas de Bibliotecas Digitais necessitam do desenvolvimento de conjuntos de protocolos para dar suporte à interação entre tipos de dados, intercomunicação dos componentes da biblioteca, busca distribuída (incluindo habilidade para busca através de bancos de dados heterogêneos, com algum nível de consistência semântica) e intercâmbio de dados.

Muitas destas questões, incluindo a questão da interoperabilidade entre bibliotecas digitais, passam pelo desenvolvimento de padrões ou de linguagens intermediárias, além de conjuntos de esquemas de mapeamento entre modelos. [SGG96], por exemplo, apresenta abordagens de testes de interoperabilidade entre catálogos, que vêm sendo feitos entre a Biblioteca Digital Alexandria e o Projeto DLI da Universidade de Berkeley, utilizando componentes da linguagem do protocolo Z39.50 como linguagem intermediária e esse mesmo protocolo para fins de busca. Z39.50 é o protocolo *on-line* padrão para catálogos de bibliotecas tradicionais e é também o atual protocolo padrão de busca para o National Spatial Data Infrastructure.

A maioria dos sistemas de bibliotecas digitais, atualmente, usa protocolos Internet, tais como o HTTP (Hypertext Transfer Protocol) para dar acesso efetivo à informação. Contudo, as várias limitações desse protocolo, incluindo sua característica de ser “sem estados” (*stateless*), isto é, de não guardar informação sobre transações efetuadas, demonstram sua inabilidade para o tratamento das questões levantadas.

Para o tratamento de questões de interoperabilidade e acesso distribuído aos dados, o Sistema EOSDIS (Earth Observing Earth/ Data and Information System) [EMM94] propõe uma arquitetura baseada em camadas, com gerenciadores de serviços distribuídos que utilizam facilidades de anúncio de serviços dos provedores locais.

O emprego de criptografia com chave pública, que inclui o desenvolvimento de um sistema de servidores de chave e a definição de padrões e de protocolos, é essencial para dar suporte a funções como segurança, autenticação, privacidade, gerenciamento de direitos e pagamentos pelo uso de propriedade intelectual. A ausência dessa infra-estrutura tem atuado como uma severa barreira à entrada de editores comerciais e de outros fornecedores de informação “*copyrighted*” no universo de bibliotecas digitais.

Um trabalho interessante em relação à perspectiva da nossa pesquisa é apresentado em [NS91]. O trabalho utiliza uma abordagem de hipertexto distribuído para integração de diversos repositórios de informação. A solução é baseada em prover organização, trans-

parência e autonomia, através das características combinadas de representação dos dados, mecanismos de armazenamento e interação com o usuário do paradigma hipermídia.

2.2 Dados Geográficos e Bibliotecas Digitais

Esta seção analisa os tópicos cobertos na seção 2.1, à luz dos problemas inerentes a Bancos de Dados Geográficos.

2.2.1 Dados Geográficos

Dados georeferenciados são dados sobre fatos, objetos e fenômenos do globo terrestre associados à sua localização sobre a superfície da Terra num certo instante de tempo. Esses dados têm quatro características principais [MP94]: sua posição geográfica (coordenadas), seus atributos (valores de dados), seus relacionamentos topológicos, e sua componente de tempo. Podem ser consideradas três categorias desses dados:

- dados convencionais: atributos alfanuméricos convencionais que descrevem as propriedades do objeto geográfico;
- dados espaciais: atributos que descrevem a geometria e localização do objeto geográfico e suas propriedades topológicas;
- dados gráficos: atributos que armazenam imagens.

Modelos de Dados Geográficos

Um modelo de dados é um conjunto de conceitos que pode ser usado para descrever os tipos de dados, relacionamentos e restrições pertinentes a esses dados. A modelagem de dados geográficos é fundamentada em duas percepções complementares do mundo [MP94, CCH⁺96]: o modelo de campos e o modelo de objetos.

O modelo de campos enxerga o mundo como uma superfície contínua, sobre a qual os fenômenos geográficos a serem observados variam segundo diferentes distribuições. Cada camada corresponde a um diferente tema de interesse (e.g., relevo, vegetação). Campos são frequentemente representados em um formato matricial, no qual os elementos da matriz são unidades poligonais regulares do espaço (células). O termo *raster* designa células retangulares e é também utilizado para designar genericamente representações matriciais. A cada célula corresponde um valor do campo representado. O interesse principal reside no conteúdo desses valores, que são tomados como válidos para toda a região, desconsiderando noções de limites.

O modelo de objetos representa o mundo como uma superfície ocupada por objetos identificáveis, com geometria e características próprias. Dois objetos podem ocupar o mesmo espaço e não precisam estar associados a um tema específico. Esses objetos geográficos são representados em formato vetorial, ou seja, sua geometria é descrita utilizando-se pontos, linhas e polígonos.

Aspectos de Arquitetura de Sistemas de Armazenamento de Dados Geográficos

Uma característica chave de dados geográficos diz respeito à forma de armazenar os dados e como integrar a representação de dados espaciais e não-espaciais. A maioria dos sistemas atuais é baseada em SGBDs acoplados a conjuntos de métodos para manusear imagens e dados espaciais. As arquiteturas para integração de um SGBD a dados espaciais são divididas em três grandes categorias [MP94, Sam95]:

1. **Sistemas Proprietários:** nos quais um banco de dados de propósito específico é associado a módulos para o processamento de dados espaciais. Usuários não podem acessar o banco de dados diretamente e a interoperabilidade com outros SGBDs é uma atividade complexa;
2. **Sistemas em Camadas:** nos quais um SGBD convencional é usado como base para a representação dos dados espaciais. Pode haver uma distinção clara entre os dados espaciais e os não-espaciais, que são implementados em diferentes modelos, ou que são armazenados em um único SGBD utilizando uma estratégia de campos longos. O acesso à informação espacial é feita através de interfaces específicas;
3. **Sistemas Extensíveis:** nos quais são utilizadas facilidades de SGBDs extensíveis ou orientados a objetos. Nestes, a dimensão espacial pode ser modelada diretamente no SGBD através da definição de tipos abstratos de dados para suporte a extensões espaciais. A formulação de consultas espaciais pode ser suportada diretamente pela própria linguagem de consulta extensível.

O acesso efetivo e eficiente aos dados geográficos armazenados no SGBD passa necessariamente pela definição de estruturas de dados espaciais para a criação de índices e métodos de acessos espaciais. As estratégias para definição dessas estruturas podem ser divididas em duas principais abordagens [MP94, Sam95]:

- agrupamento de dados geográficos em hierarquias, através do conceito de *Minimum Bounding Rectangle (MBR)*. Árvores-R são as estruturas mais utilizadas para representar esse grupo;

- divisão do espaço em células disjuntas. Cada objeto é armazenado na célula correspondente ao seu *bucket*. Objetos pontuais, retângulos, polígonos e regiões podem ser estruturados segundo um conjunto de variações da estrutura *Quadtree* (termo utilizado para descrever estruturas de árvores de decomposição recursiva). Outras estruturas, como *Gridfiles*, podem ser usadas.

2.2.2 Modelagem e Tratamento de Dados e Metadados

Discutiremos aqui apenas as extensões necessárias à acomodação de dados indexados espacialmente tais como dados geográficos em uma estrutura de bibliotecas digitais.

O modelo de dados hipermídia é composto de :

1. Ligações;
2. Nós;
3. Extensões com Metadados.

A seguir, serão analisadas aplicações de cada uma dessas características na definição de um modelo de dados geográficos em bibliotecas digitais.

Metadados

A especificação dos metadados apropriados depende da definição das consultas a serem suportadas pelo sistema. [FCF+95] considera que é possível distinguir duas classes principais de consultas a dados geográficos:

1. Consultas “O que está aqui?”, nas quais os usuários querem descobrir informações sobre fenômenos que ocorrem em localizações especificadas;
2. Consultas “Onde está isso?”, nas quais usuários querem descobrir as localizações no qual um fenômeno específico ocorre.

Uma importante extensão à componente de catálogo da biblioteca digital pode ser a incorporação de *gazetteers* [ALD+95]. Um *gazetteer* é basicamente uma lista de características geográficas (como cidades, rios), um conjunto de instâncias nomeadas dessas características e uma localização geográfica para cada instância. O uso de *gazetteers* permite a recuperação de dados baseados em sobreposição de localizações geográficas dos itens da coleção com as das instâncias nomeadas dos itens do *gazetteer*.

Há propostas recentes de padrões para representação de dados e metadados geográficos. O uso de padrões facilita o compartilhamento, a integração e a transferência de dados, assumindo um importante papel no tratamento de questões de interoperabilidade entre catálogos e intercâmbio de dados [KPD+95].

O Federal Content Standards for Digital Geospatial Metadata [Com] adotado pelo Federal Geographic Data Committee tem sido um dos padrões de metadados mais utilizados pelos atuais serviços de fornecimento de dados geográficos.

O FGDC desenvolveu seu padrão baseado em quatro características que definem o papel de metadados: disponibilidade, conveniência para uso, acesso e transferência. Especificamente, o padrão define os elementos de dados para os seguintes tópicos:

1. Identificação;
2. Qualidade de dados;
3. Organização de dados espaciais;
4. Informação de referência espacial;
5. Descrições de Entidades e Atributos;
6. Informação de Distribuição;
7. Informação de Referência de Metadados.

O padrão não especifica, contudo, como a informação é organizada em um sistema de computação ou na transferência de dados, os meios pelos quais essa informação é transmitida, nem os meios pelos quais essa informação é comunicada ao usuário. Questões como modelagem dos dados, intercâmbio de informação, e interfaces, são decididas pelos projetistas do sistema.

Ligações

Ligações conectam dados que têm alguma relação entre si de acordo com critérios do usuário. Ligações podem conectar, por exemplo, os dados à sua informação de metadados, o que pode servir tanto para fins de análise dos metadados quanto para construção de consultas. Hyperquery [Yan94] é uma linguagem que utiliza esse princípio. O usuário pode escolher uma característica para consulta e pode ir refinando sua consulta pelo preenchimento de características que são definidas hierarquicamente numa estrutura de árvore, acessadas através de seleções de cliques de mouse que expandem níveis de profundidade. Um exemplo de navegação em dados geográficos seria definir um determinado fenômeno (e.g., rios) e refiná-lo aos poucos (e.g., rios.volumeDeágua). Consultas a dados definidos hierarquicamente, como é o caso dos metadados do padrão FGDC, poderiam se beneficiar de uma linguagem desse tipo.

Ligações podem ainda conectar relacionamentos estruturais e de grupos entre objetos geográficos. Exemplos são: componentes de um fenômeno geográfico composto; diferentes versões do mesmo objeto; relacionamentos espaciais, temporais, e relacionamentos entre processos (por exemplo, o próximo passo em um *workflow* ou procedimento).

Nós

Os nós do sistema hipermídia geográfico podem conter qualquer dado (georeferenciado ou não). Isso pode incluir, como foi visto, mapas digitalizados, imagens de sensoriamento remoto, conjuntos de dados *raster* e *vetor*, textos, vídeos e áudios. Podem ainda incluir softwares ou programas de aplicação que atuem nos dados geográficos.

2.2.3 Interface com o Usuário

O componente de interface desempenha um papel fundamental em uma biblioteca digital de dados geográficos. Ele é responsável por refletir todas as funcionalidades oferecidas pelo sistema e por definir a forma como os usuários procederão suas consultas. A interface de uma biblioteca digital para dados geográficos deve apresentar características gerais de interfaces e soluções para os problemas citados na seção 2.1.3. No caso de dados geográficos, essa interface deve apresentar as seguintes funcionalidades específicas:

1. Consultas por Conteúdo

Este item depende do conjunto de consultas que foram previstas para a definição dos metadados.

Para consultas “Onde está isso?”, a interface deve fornecer meios de entrada para valores de metadados descrevendo os atributos desejados. Consultas “O que está aqui?” podem ser executadas sobre mapas base que podem ser sucessivamente refinados em termos de resolução até um ponto satisfatório para o usuário. Os dados recuperados se referem à região indicada. Outra forma de definir regiões de busca é a entrada explícita dos valores da extensão espacial desejada (por exemplo, latitude/longitude) sobre os quais serão efetuadas as buscas. Finalmente, a região geográfica que limita a consulta pode ser especificada implicitamente através da definição das localizações geográficas dos elementos de um *gazetteer*.

2. Consultas por Folheamento

Consultas por Folheamento tomam a forma de sessões de longa duração, nas quais o usuário navega através de emaranhados de informação com o objetivo de convergir para um conjunto satisfatório de dados. Esta forma de consulta é essencial, a partir do momento em que o usuário não sabe exatamente o que quer ou quando tem apenas uma vaga idéia, estando o âmbito da pesquisa concentrado em um conjunto enorme de dados heterogêneos, como é o caso de uma Biblioteca Digital para dados geográficos.

3. Apresentação dos Dados

Não é suficiente apenas saber que os dados existem e onde eles se encontram; usuários querem examinar os mapas, as fotos aéreas, etc, e realizar possíveis refinamentos em suas consultas, baseados nos dados visualizados. Assim, consultas implicam em um retorno de dados ao usuário, em um ambiente no qual os dados possam ser mostrados. Novas consultas, por conteúdo ou folheamento, podem ser executadas a partir dessa apresentação.

A utilização de conceitos de hipermídia no projeto da interface auxilia enormemente as atividades de apresentação e de navegação. Hipermídia pode, por exemplo, prover ligações que ativam análises detalhadas de dados geográficos providas por um SIG, ligações explicativas sobre tais análises, e ligações aos dados fonte. A interligação e a apresentação integrada de mapas, textos, imagens, dados tabulares podem ser gerenciadas por ligações hipermídia.

Algumas características gerais de interfaces hipermídia que auxiliariam o processo de navegação entre dados geográficos incluem:

1. Folheamento: que permite usuários selecionarem e percorrerem ligações;
2. *Backtracking*: que capacita usuários a refazerem as trilhas de passos seguidos;
3. Funcionalidades de anotação: que habilitam usuários a adicionarem novas ligações e comentários à estrutura do hiperdocumento, além daqueles fornecidos pela estrutura inicial;
4. Mecanismos de “Visão Geral” da Informação: que apresentam as âncoras e ligações como inter-relacionamentos numa estrutura de grafos;
5. Caminhos e “*Guided Tours*”: que provêem trilhas recomendadas através da rede de hiperdocumentos.

Todas essas características auxiliam a resolução de problemas típicos de navegação hipermídia vistos na seção 2.1.3.

2.2.4 Sistema de Armazenamento

Sistemas de Armazenamento para Dados Geográficos são baseados nas arquiteturas vistas na seção 2.2.1, que descreve as diferentes formas de acoplamento de um SIG a um SGBD. Dentro do SGBD são diferenciados ao menos os seguintes componentes:

- Componente de Catálogo: que armazena e gerencia os metadados.
- Componente de Índices: que auxiliam as buscas pelos dados.

- Componente de Dados: que, em geral, são separados em dados espaciais e em não espaciais.

O componente de catálogo é geralmente armazenado em um SGBD. Já os dados espaciais podem ser armazenados em sistemas de arquivos ou SGBDs, e ligados aos seus conjuntos de metadados através de mecanismos de chave estrangeira e identificadores de objetos espaciais, ou podem ser armazenados no próprio SGBD, utilizando um sistema extensível. O sistema de armazenamento da Biblioteca Digital Alexandria [ALD⁺95] é um exemplo do primeiro caso e o do Sistema Geoscope [SCB95] é um exemplo do segundo.

Como a quantidade de metadados no catálogo cresce rapidamente, é crítico prover suporte eficiente a diferentes tipos de consultas com o uso de métodos apropriados de indexação espacial. As estratégias de índices também são baseadas naquelas vistas na seção 2.2.1. Geoscope [SCB95], por exemplo, possui uma hierarquia de classes abstratas que dá suporte a múltiplos índices.

2.2.5 Comunicação e Redes

Formatos e protocolos para trocas de dados geográficos precisam ser definidos para o intercâmbio da informação. Padrões como o SAIF (Spatial Archive and Interchange Format), o SDTS (Spatial Data Transfer Standard) e o OGIS (Open Geodata Interoperability Specification) têm surgido para resolver o problema.

O SAIF é uma linguagem OO para representação e troca dos dados espaciais, independente do esquema binário de representação. O SDTS especifica construtores de troca, formatos de endereçamento, estrutura e conteúdo para dados geo-referenciados, vetoriais e matriciais. O OGIS se baseia em uma biblioteca de classes para modelar objetos e funções de aplicação, implementando serviços de comunicação.

Protocolos de transferência de arquivos padrões como o FTP (File Transfer Protocol) podem também ser utilizados para captura de dados geográficos, cujo formato é então ignorado.

2.3 Bibliotecas Digitais voltadas a Dados Geográficos

2.3.1 O Projeto Biblioteca Digital Alexandria

O Projeto da Biblioteca Digital Alexandria (ADL) [ALD⁺95, Tea97, FCF⁺95] tem por objetivo prover amplo acesso a informações espacialmente indexadas. O projeto envolve um consórcio de universidades, instituições públicas e corporações privadas, encabeçadas pela Universidade da Califórnia em Santa Barbara.

A arquitetura do sistema deriva dos quatro principais componentes de uma biblioteca tradicional: coleções, catálogos, interfaces e facilidades de aquisição. O enfoque principal está concentrado no componente de catálogo e no projeto da interface com o usuário que suporta as funcionalidades da biblioteca. O atual estágio de desenvolvimento envolve a expansão das funcionalidades do protótipo para oferecer acesso através da World-Wide-Web (WWW).

O componente de armazenamento contém coleções de objetos digitais. Aspectos desse componente incluem representação binária da informação, identificadores universais de objetos e procedimentos para interpretar e retornar os dados. A ADL está empregando um sistema de armazenamento hierárquico distribuído para armazenar suas próprias coleções. À medida que locais de armazenamento vão sendo adicionados continuamente, a ADL vem executando vários testes de interoperabilidade com outras bibliotecas. As técnicas utilizadas, envolvendo transformações de linguagens, são descritas em [SGG96].

O componente de catálogo inclui metadados e máquinas de busca que permitem aos usuários identificar objetos de interesse. Esse componente incorpora elementos dos padrões de metadados USMARC e FGDC para representar a informação espacialmente indexada. O projeto do catálogo está intimamente ligado às classes de consultas suportáveis e à questão da interoperabilidade de catálogos.

A Interface com o Usuário pretende prover fácil acesso a um conjunto de central de funcionalidades para uma população de usuários heterogêneos, visando englobar as principais funcionalidades descritas no modelo de interfaces na seção 2.2.3. Contudo, algumas limitações impostas pela tecnologia WWW dificultam o alcance desses objetivos:

1. as atuais implementações de HTML não provêem mecanismos para apresentar dados vetoriais e provêem um fraco suporte à entrada e à apresentação de dados espacialmente indexados;
2. o protocolo HTTP (Hypertext Transfer Protocol) da WWW é sem estados, i.e., ele não guarda memória de suas transações e é projetado para transações pequenas e rápidas;
3. os atuais *browsers* WWW não permitem um alto grau de interatividade.

As abordagens empregadas pelo projeto ADL para o tratamento de cada um desses problemas podem ser encontradas em [ALD⁺95].

A interface ADL é basicamente uma coleção de páginas HTML que implementam três principais capacidades: o visualizador de mapas, consultas via *gazetteer*, e consultas gerais ao catálogo. A função primária do visualizador de mapas e das páginas do *gazetteer* é permitir ao usuário definir extensões espaciais ou regiões para a busca no catálogo de metadados. O visualizador permite definir regiões de busca explicitamente (aplicando

processos de *zoom* e *pan* no mapa base), enquanto o gazetteer as define implicitamente como localizações geográficas correspondendo aos nomes de lugares e aos tipos de características. O visualizador de mapas também é um *plano de fundo* no qual os itens do gazetteer e as extensões espaciais dos resultados das buscas são desenhados. Detalhes de implementação podem ser encontrados também em [ALD⁺95] e em [Tea97].

Consultas ao catálogo são formuladas via uma representação genérica em forma normal conjuntiva e traduzidas para a linguagem SQL do SGBD do catálogo de metadados. As hiperligações são utilizadas apenas para mostrar as imagens ou os mapas associados aos resultados da consulta, que são mostrados como uma tabela HTML, não desempenhando papel de interligação das informações.

O componente de entrada visa estender o esquema de metadados, incorporar conjuntos de dados existentes e criar novos registros de metadados para itens selecionados. A interface desse componente é baseada numa interface SGBDR padrão.

Um conjunto de linguagens e protocolos oferece a infra-estrutura de comunicação necessária, tanto para o acesso externo à biblioteca quanto para intercomunicação entre os componentes do sistema. O sistema WWW utiliza o servidor HTTP NCSA para a comunicação com os browsers WWW clientes e utiliza também um conjunto de aplicações CGI (Common Gateway Interface) para geração dinâmica dos mapas base e para processamento da consulta. Consultas são traduzidas em diversas sintaxes de linguagens de consulta (e.g., SQL ou Z39.50) para busca local e em catálogos remotos. O código HTML para retorno da resposta é gerado automaticamente por cerca de 15K linhas de código rodando no servidor. Protocolos padrão Internet para transferência de dados como o FTP (File Transfer Protocol) são utilizados para "download" dos dados.

2.3.2 O Projeto da Biblioteca Digital da Universidade de Berkeley

O Projeto da Biblioteca Digital da Universidade de Berkeley [Wil96], denominado Sistema de Informação Digital Ambiental da Califórnia, tem desenvolvido uma biblioteca digital que inclui diversos tipos distintos de dados ambientais. A biblioteca conterà: todos os relatórios técnicos ambientais do estado da Califórnia; modelos computacionais que simulam fatores ambientais como tráfego e uso da água; fotografias aéreas e terrestres; mapas topográficos do uso do solo; vídeos e bancos de dados sobre a distribuição e classificação sobre recursos de plantas. A biblioteca digital de Berkeley tem atuado em conjunto com um consórcio de usuários e provedores de dados, liderados pela Agência de Recursos Ambientais do Estado da Califórnia.

O projeto da Biblioteca Digital de Berkeley é centrado na provisão de serviços que tratam das necessidades de recuperação de informação de grupos de trabalho. O sistema

tem uma arquitetura simples consistindo de repositórios, clientes, métodos de indexação e busca, protocolos; buscando, neste último, interoperabilidade. *Repositórios* são implementados como bancos de dados que suportam funções e métodos de acesso definidos pelo usuário. *Clientes* são browsers projetados para diferentes tipos de documentos tais como imagens, conjuntos de dados de SIGs e documentos de texto. *Indexação e Busca* envolvem o uso de processamento de linguagem natural, visão computacional e técnicas de SIGs para melhorar o acesso à informação orientada a mapas, imagens e textos. Os repositórios atuam como seus próprios servidores de indexação. *Protocolos* permitem a comunicação dos clientes com os servidores. Berkeley tem utilizado o HTTP, mas está projetando seu próprio protocolo, denominado ZQL, cujo nome é derivado da combinação do protocolo padrão Z39.50 e de SQL. A *Interoperabilidade* é considerada em diversos níveis: no nível de repositório, no qual é provido acesso de baixo nível para as coleções; no nível de de esquema, no qual os clientes são aplicados aos repositórios; e no nível externo, no qual visões dos esquemas de metadados da ADL e de Berkeley são criadas para permitir que clientes rodem sobre a união dos bancos de dados de fotografias aéreas de ambos os projetos.

2.3.3 O Projeto GeoScope

O objetivo do projeto Geoscope [SCB95, CT95] é tornar disponível imagens de sensoriamento remoto além de outros dados geoespaciais para uma larga faixa de público na Internet, enquanto simultaneamente provê um arcabouço para resolver características de redes e interoperabilidade de dados.

O trabalho está sendo desenvolvido para a NASA pelo USDAC (Universal Spatial Data Access Consortium), envolvendo um projeto de três anos e de 2.3M dólares de recursos cooperativos. Membros da USDAC incluem Bellcore, Camber Corporation, California Resource Agency, the Open GIS Consortium, Rutgers University's Center for Remote Sensing and Spatial Analysis, e NASA's Space Science Data Operation Office.

A ênfase do trabalho é o projeto de um servidor de catálogos que suporta um esquema dinâmico de metadados. O projeto é baseado em GeoHarness, uma extensão de InfoHarness - um sistema que suporta busca e recuperação de informação heterogênea sem qualquer relocação ou reformatação de dados.

GeoHarness utiliza um modelo de objetos que é capaz de encapsular dados não formatados, sendo também capaz de suportar armazenamento e recuperação de atributos hierárquicos heterogêneos. A idéia por trás da linguagem é prover uma visão de dados semanticamente ligados e facilitar busca e folheamento, impondo relacionamentos de *contém*, *está-contido* e *parte-de*. O trabalho considera o papel desempenhado pela teoria dos conjuntos em prover estruturação dos dados.

Um repositório Geoharness é um conjunto de Objetos que são conhecidos por um único

servidor GeoHarness. Objetos têm identificadores únicos mantidos pelo sistema e devem possuir informações de tipagem indicando o método de recuperação. Objetos GeoHarness incluem objetos simples (e.g., uma imagem de satélite) e sua meta-descrição; objetos compostos, que combinam objetos simples e um conjunto de referências a outros objetos componentes; e coleções, compostas de um conjunto de referências a outros objetos (seus filhos) e de um possível conjunto de referências para índices independentes.

A hierarquia de classes GeoHarness possui *classes abstratas* que provêm métodos compartilhados por grupos de classes terminais. As *classes terminais* são classes que não podem ser subtipadas e são instanciadas como objetos GeoHarness. Sua definição é voltada aos dados e não são parte da implementação. Exemplos de classes terminais são ferramentas para acessar e mostrar os dados (e.g., visualizadores como o *xv* e a camada de interface de GIS) e para indexar a informação (e.g., os índices WAIS e LSI).

A hierarquia abstrata tem os seguintes objetivos básicos:

1. suportar a execução de consultas a índices independentes e cuidar da apresentação dos resultados dessas consultas;
2. suportar, em tempo de execução, a apresentação da informação encapsulada pelos metadados.

Uma descrição detalhada dessa hierarquia pode ser encontrada em [SCB95]. Os principais componentes da arquitetura incluem:

1. um servidor GeoHarness (GH Server), que usa metadados;
2. um gateway HTTP, que é usado para conexão clientes HTTP / servidor;
3. o Gerador de Metadados [SSKS], que é usado para geração *off-line* de metadados que representam a visão desejável. Esses metadados são usados pelo servidor para busca e recuperação em tempo de execução.

Em tempo de execução, os clientes HTTP podem enviar suas consultas, percorrer ou retornar pedidos de dados que são passados para o gateway. Este executa as seguintes operações:

1. verifica o pedido e lê a informação de entrada quando o pedido é associado a um formulário HTML;
2. estabelece uma conexão de soquete com o servidor GH, gera e manda uma solicitação de dados - que são buscados através da informação de metadados do catálogo - e espera a resposta;
3. verifica a resposta, converte-a para uma combinação de páginas HTML, adiciona o cabeçalho HTTP e passa a resposta transformada para o *browser*.

2.3.4 O Projeto EOSDIS

O EOSDIS (Earth Observing System/ Data and Information System) [EMM94, EW94] é um projeto que tem por objetivo prover acesso fácil a dados do EOS e a outros dados científicos sobre a Terra. O EOS é uma missão de pesquisa multidisciplinar, com objetivos a longo prazo para o estudo dos processos de mudanças globais e para o desenvolvimento da capacidade de prever a evolução do sistema da Terra em escalas de tempo de décadas a séculos. O EOSDIS irá conter conjuntos de dados de uso potencial para uma larga faixa de usuários. O sistema irá prover para os usuários informações e ferramentas de buscas para localização, seleção e acesso aos dados. Usuários serão capazes de acessar o sistema, algoritmos, publicações, softwares desenvolvidos pela comunidade científica e dados geográficos em geral. Dados nos locais de armazenamento serão organizados em uma ou mais coleções dos itens relacionados, contendo dados e metadados.

Os principais objetivos da arquitetura do sistema são a definição de :

1. facilidades que levem os cientistas a localizar, obter ou usar recursos que estão disponíveis em rede;
2. características que ajudariam um cientista a enfrentar os problemas decorrentes do primeiro item, e.g., diferenças em formatos de dados, terminologia e ferramentas para entrada e saída;
3. funções que tornariam mais fácil para cientistas colaborarem em projetos de pesquisa através da rede.

A arquitetura conecta os serviços oferecidos pelos provedores às necessidades expressas pelos usuários em termos de acessos ao sistema. Três classes de softwares são consideradas nessa camada: o serviço de anúncio, que publica os serviços oferecidos pelos provedores na rede; o serviço de intermediação de pedidos, que executa o casamento entre as solicitações dos usuários e os serviços oferecidos; e o gerenciador de informação distribuída. Este último será responsável pela decomposição da consulta, se necessário, e pela geração de um plano contendo subconsultas, as quais serão processadas pelos diversos locais de armazenamento e pelo gerenciamento dos resultados intermediários.

Cada provedor local escolhe os serviços que irá prover à comunidade. As principais características a serem tratadas no projeto das interfaces do provedor com o sistema envolvem: organização interna autônoma; anúncio de serviços aos usuários; suporte a buscas e consultas; suporte à notificação para usuários da chegada de novos dados e serviços; suporte ao gerenciamento de incompatibilidades.

Uma última camada é a de interface com o usuário para dar acesso aos dados, serviços e protocolos. A interface EOSDIS irá guiar usuários para construir consultas que identifiquem dados desejados sem conhecimento da arquitetura do sistema. Usuários serão

capazes de executar buscas incrementais refinando ou adicionando critérios para identificar dados de maior interesse.

Usuários poderão pesquisar dados geográficos usando combinações de: referências de coordenadas geográficas, polígono circundante, ponto e raio, nomes geográficos comuns e conteúdo dos dados (baseados nos valores de metadados). Serão suportadas consultas a dados distribuídos e resolução de conflitos em relação a características espaço/temporais.

2.3.5 O Projeto do Parque Adirondack

O Projeto do Parque Adirondack [AAB⁺95] corresponde ao desenvolvimento de um sistema para organização, busca e uso de informações disponíveis sobre o Parque Adirondack de Nova York. Esse sistema deve permitir à agência de controle do Parque responder imediatamente a consultas do público, que atualmente requerem buscas de dados caras e demoradas. O sistema deve também permitir acesso *on-line* a informações sobre a agência. A partir desses objetivos, os projetistas enfocaram quatro subobjetivos:

1. criar um protótipo que integra informação geográfica, bancos de dados e documentos;
2. Converter dados existentes da agência para uso no protótipo;
3. estimar os custos e as características de performance de um sistema completo e projetar uma estratégia de conversão dos dados para populacionar o sistema completo;
4. identificar as vantagens do serviço adicional ao cliente e outros benefícios que o sistema pode trazer.

O sistema Adirondack foi construído para integrar quatro tecnologias: um sistema de informação geográfica, um sistema de gerenciamento de documentos, um sistema de *workflow* e um sistema de banco de dados para servir como repositório de dados e ligação de comunicação entre os vários sistemas.

Ações possíveis no sistema são: localizar uma propriedade no mapa e vê-la em combinação com outros dados geográficos; acessar documentos ou imagens escaneadas relacionadas à propriedade; obter acesso a dados do banco de dados relacionados à propriedade. Uma análise completa e avaliações de custo/performance, custo/benefício e vantagens alcançadas com a utilização do sistema podem ser encontradas em [AAB⁺95].

Parcelas no mapa podem ser identificadas apontando para a localização no mapa eletrônico, selecionando a parcela via o IDMap (Identificador), pesquisando em um índice por um nome associado com a propriedade, ou usando uma característica de busca que identifica as parcelas que estão dentro de uma distância especificada de uma localização conhecida.

A principal vantagem do sistema é sua capacidade de tirar proveito da integração de tecnologia, como poder mostrar mapas e documentos em um único sistema integrado. O protótipo, contudo, é limitado em relação aos objetivos pretendidos. Entre os principais problemas, podemos citar, por exemplo, a inabilidade de criar e armazenar novos produtos. Um usuário pode localizar um terreno e ver os documentos associados a ele, mas não pode armazenar esses dados juntos em um novo *“folder”*. O protótipo não tira vantagem de muitas das possíveis ligações entre documentos, e necessita de um SIG para mostrar sobreposições e relacionamentos. Em adição, a ausência de comunicação interprocesso entre o SIG e o software de documentos não permite ao usuário reusar velhas sessões para olhar novos documentos.

Grande parte do custo do sistema foi associado à localização e à aquisição dos dados, à sua conversão para uma forma digital e à sua integração dentro do SIG. Problemas relacionados à integração de dados espaciais existentes envolveram sua heterogeneidade de representação, precisão, inconsistência temporal, variações de escala. Por fim, um conjunto de pontos positivos e negativos e o delineamento de estratégias para a implementação do sistema completo foram obtidos da análise do protótipo.

Alguns aspectos resultantes da análise das características técnicas foram:

1. um protótipo possui alto valor como ferramenta de análise de requisitos e objetivos;
2. uma estratégia coerente de desenvolvimento e de gerenciamento de dados é essencial em qualquer sistema que trata com uma variedade de formatos de dados;
3. os metadados são uma adição útil a dados espaciais, tornando o compartilhamento de dados mais fácil;
4. a modelagem de dados é uma ferramenta útil para o exame de relacionamentos existentes entre elementos de dados e mostra onde outros devem ser criados;
5. aspectos temporais de dados espaciais são uma fonte de dificuldades.

2.4 Resumo

Este capítulo discutiu os principais conceitos relacionados a bibliotecas digitais, organizados em quatro grandes áreas de estudo: interface com o usuário, modelagem e tratamento de dados, interoperabilidade e redes, e armazenamento de dados.

O capítulo avaliou também o impacto do uso de dados geográficos em bibliotecas digitais, analisados à luz das quatro áreas anteriormente relacionadas. Finalmente, foram apresentados alguns dos projetos de bibliotecas digitais voltadas a dados geográficos que têm sido desenvolvidos.

Capítulo 3

Modelos Hipermedia: Visão Geral e Análise de Requisitos para Bibliotecas Digitais

3.1 Introdução

Aplicações hipermedia devem ser desenvolvidas a partir de modelos de dados hipermedia (e.g., [SRB96, GPS93, ISB95]). Vários são os aspectos que contribuem para esta necessidade [BI95]. Primeiramente, sob uma perspectiva de arquitetura, um modelo de dados isola os aspectos do sistema hipermedia dos detalhes de implementação física, permitindo que vários serviços de apresentação e aplicações sejam construídos sobre um único Sistema de Banco de Dados Hipermedia (SGBDH). Em segundo lugar, aplicações hipermedia envolvem muitos componentes distintos, tais como apoio à navegação, interface com o usuário, armazenamento do conteúdo dos documentos, requerendo, desta forma, modelos de dados especializados. Em terceiro lugar, como consequência, modelos de dados tradicionais não representam bem os aspectos intrínsecos de uma aplicação hipermedia. Por último, modelos hipermedia atuam como base para a descrição de aplicações, possibilitando diversas avaliações e análises em aspectos variados como estrutura, dinâmica e interação. Como muitas das páginas WWW atuais podem atestar, poucos projetistas têm experiência em projetar e avaliar a utilização de aspectos hipermedia em uma aplicação.

É importante destacar a diferença entre um sistema e uma aplicação hipermedia. Segundo [ISB95], o primeiro é um ambiente que facilita a criação do último. Um modelo de dados para um sistema hipermedia detalha sua arquitetura interna, mas é de pouco valor para modelar aplicações. Já um modelo de dados voltado à autoria de aplicações introduz abstrações úteis para descrever a informação armazenada, os mecanismos de navegação e os aspectos de interface.

De especial destaque, nesse sentido, foi o trabalho de Milet et al. [MSL96], o qual propõe um metaesquema para implementar os conceitos básicos do modelo de autoria OOHDM [SRB96] sobre um banco de dados orientados a objetos, diminuindo assim a distância semântica entre ambas as abordagens. O trabalho fornece um arcabouço uniforme para implementação de aplicações, através de especializações do metaesquema, beneficiando-se das características do banco de dados orientado a objetos para o gerenciamento das estruturas hipermídia.

Este capítulo aborda os aspectos pertinentes à construção do modelo hipermídia de uma bibliotecas digital. A discussão dar-se-á em dois níveis básicos: o nível de sistema e o de aplicação. Dentro da discussão em cada nível, destacaremos as necessidades intrínsecas de bibliotecas digitais a serem atendidas pelos modelos. Esta análise servirá como base para o desenvolvimento do modelo a ser utilizado em nosso trabalho. A seção 3.2 discute características gerais de modelos de sistemas hipermídia, com ênfase nos requisitos de bibliotecas digitais. A seção 3.3 apresenta um *survey* dos modelos de sistemas hipermídia propriamente ditos. A seção 3.4 avalia modelos para autoria de aplicações hipermídia. A seção 3.5 resume e apresenta as conclusões do capítulo.

3.2 Características Gerais de Modelos de Sistemas Hipermídia para Bibliotecas Digitais

Esta seção discute os principais aspectos de sistemas hipermídia para o desenvolvimento de aplicações voltadas a bibliotecas digitais.

3.2.1 Elementos Básicos dos Modelos

Os elementos básicos do modelo de dados de um sistema hipermídia provêm um “denominador comum” das características presentes na maioria destes sistemas [Boo92]. A maioria dos modelos utiliza três conceitos principais: um conjunto de *objetos primitivos*; algumas *abstrações* de mais alto nível que servem como coletoras para estes objetos; e um conjunto básico de *operações* para visualizar e editar objetos. Consideraremos a seguir cada um desses aspectos.

1. Abstrações de Baixo Nível

São entidades estruturais independentes de seu conteúdo, da aplicação e do ambiente do usuário. No modelo hipermídia, essas entidades básicas correspondem a *nós* e *ligações*, que podem ser refinadas através da especificação de pares atributo-valor.

Definições de ligações podem suportar diversas características diferenciadoras como tipo, direção, aridade, atributos, âncoras e relacionamentos com o nó. Geralmente, ligações são definidas no mesmo nível de importância semântica dos nós.

Âncoras são, em geral, definidas como extremos de ligações e referenciam alguma porção do conteúdo de um nó. Âncoras são consideradas entidades pertencentes à estrutura do nó, sendo responsáveis pela manutenção da separação entre essa estrutura e o conteúdo do nó.

2. Abstrações de Alto Nível

Permitem agregação dos componentes básicos e hierarquias de generalização/especialização. Em modelos baseados em grafos, por exemplo, tais abstrações correspondem a grafos e subgrafos. Abstrações adicionais podem ainda incluir versões e contextos.

3. Operações

Incluem, geralmente, especificações para funções genéricas de gerenciamento, tais como visualização e edição, operações básicas de inserção e eliminação de dados. Cada modelo define outras operações particulares. Exemplos são operações para manter restrições de integridade, para assegurar unicidade do objeto, para desabilitar a definição recursiva de objetos complexos, além de operações para detectar conflitos entre versões de objetos.

4. Relacionamento com a arquitetura subjacente

Uma arquitetura geral inclui três elementos básicos: a interface com o usuário, as aplicações e o gerenciamento dos dados. Em sistemas hipermídia, uma separação clara entre essas três camadas é especialmente importante, devendo-se refinar a camada de gerenciamento de dados de forma a distinguir claramente a estrutura dos dados do conteúdo desses dados. Isto é essencial no caso de sistemas hipermídia, pois a mesma estrutura pode ser usada para representar diversas mídias. Por outro lado, um mesmo tipo de mídia pode ser mapeado para múltiplas estruturas de maneira a suportar diferentes visões de usuários ou dispositivos de visualização.

3.2.2 Abertura do Sistema

A abertura do sistema é um aspecto crucial. Um sistema hipermídia aberto é capaz de prover múltiplas ligações extensíveis entre múltiplos tipos de documentos de uma maneira robusta e transparente [VC94].

Dentro do contexto de bibliotecas digitais, uma biblioteca tradicional pode ser considerada um sistema fechado, em que livros ou outros materiais são incluídos através de um sistema de admissão que registra e classifica estes itens para catálogo e empréstimo. Em

contraste, bibliotecas digitais não são sistemas estáticos e imutáveis, mas sim ambientes dinâmicos capazes de gerenciar um crescente fluxo de documentos, muitos dos quais serão também dinâmicos e passíveis de revisão freqüente. Estas características, juntamente com a necessidade de suporte a múltiplas mídias e com a alta volatilidade de bibliotecas digitais, pressupõem um ambiente mais aberto e dinâmico.

[VC94, DHH⁺92] oferecem uma visão geral dos requisitos de um sistema hipermedia aberto. São eles:

1. Extensibilidade de mídias e suporte a ligações

Esta característica é essencial para sistemas hipermedia não-estáticos, como é particularmente o caso de bibliotecas digitais. A extensão da base de informação hipermedia é permitida através da criação de novas ligações dentro de documentos já existentes, da adição de novos documentos e da criação de ligações de qualquer documento para qualquer outro, mesmo de tipos distintos.

Sistemas hipermedia abertos devem ainda permitir suporte tanto a dados textuais quanto a dados multimídia complexos. Para isso, eles devem fornecer meios de integrar todos os tipos de arquivos e de documentos.

2. Independência entre a ligação e o conteúdo da mídia

Esta característica permite responder questões estruturais a respeito da rede hipermedia ao mesmo tempo que possibilita a interligação de informações contidas em arquivos protegidos ou em arquivos "read-only" como, por exemplo, CD-ROMs.

A independência não é contemplada por muitos dos sistemas hipermedia atuais. Estes representam ligações como alguma forma de marcação armazenada dentro do arquivo (e.g., tags HTML em páginas WWW). Isso gera a noção de dados "fechados" ou "privados", i.e., a noção de que não é possível processar o dado fora do sistema hipermedia sem alterar o formato privado.

3. Distribuição

Um sistema aberto deve permitir integração de ferramentas que rodem em sistemas heterogêneos, bem como suportar a distribuição dos dados e processos através de redes de diversas plataformas de hardware.

4. Abstração de endereçamento

Abstrações de endereçamento provêm esquemas independentes de protocolos e de redes para a recuperação de documentos e para a identificação de âncoras. Sistemas hipermedia abertos em ambientes de redes heterogêneas devem permitir referenciar documentos em outros sistemas, transferir documentos entre estes e coexistir com outras aplicações.

5. Ligações robustas

Um ambiente aberto deve permitir a manutenção da integridade das ligações frente a mudanças no ambiente. Para isso, ele deve ser capaz de evitar ou de resolver possíveis mudanças na localização física de documentos, mudanças na localização de âncoras ou no conteúdo do documento. Um ambiente hipermídia aberto deve também ser capaz de monitorar a edição de documentos fora do ambiente hipermídia. Esta última característica pode causar incoerências entre a posição da âncora armazenada na hiperbase e a sua localização atual no documento.

3.2.3 Suporte a Ligações Dinâmicas

À medida que uma biblioteca digital cresce com a adição de dados e de novos *sites*, torna-se cada vez mais difícil criar e manter a rede hipermídia, bem como gerenciar a sobrecarga envolvida na especificação manual de todas as ligações dentro da biblioteca. Assim, outro aspecto de fundamental importância num modelo hipermídia para bibliotecas digitais é a questão do suporte a *ligações dinâmicas*. Ligações dinâmicas são aquelas que têm a capacidade de automaticamente se adaptarem ao crescimento e à evolução da base de informação. No modelo de ligações dinâmicas, tanto a origem quanto o destino da ligação podem ser dinamicamente computados.

O modelo tradicional usado para especificar uma ligação, conhecido como *modelo de ligações declaradas*, permite modelar apenas relacionamentos cuja origem e destino são explicitamente enumerados. Apesar da importância de ligações declaradas na representação de relacionamentos de natureza específica (como, por exemplo, aqueles derivados de uma modelagem conceitual), sua estrutura fixa deve ser expandida para dar suporte às ligações dinâmicas.

O desenvolvimento do modelo de ligação dinâmica contempla os seguintes aspectos [AV94]:

1. Papel desempenhado pela ligação

Uma ligação representa um relacionamento estrutural, semântico ou referencial. Tal relacionamento não precisa ser permanente e, em alguns casos, pode ser determinado como resultado de computação (e.g., através de consulta à hiperbase).

2. Crescimento e evolução da hiperbase

Como o tamanho da hiperbase cresce, existe uma necessidade de geração automática de ligações e de tipos de ligações. Além disto, a informação contida na hiperbase pode mudar freqüentemente em alguns sistemas, necessitando de ajustamento dinâmico do escopo da ligação, englobando problemas de versionamento e de edição, por exemplo.

No caso de bibliotecas digitais, este aspecto envolve também a manutenção constante dos catálogos que descrevem os documentos, sem o qual a consistência da biblioteca não pode ser garantida.

Dentre as vantagens oferecidas pelo paradigma de ligações dinâmicas podemos citar:

1. Extensibilidade

Os novos componentes adicionados à base de informação são automaticamente incluídos em uma operação de ligação dinâmica sem qualquer necessidade de marcadores especiais (e.g., *tags* HTML) ou de modificações nas estruturas de dados. Modificações nos dados também são automaticamente absorvidas.

Ligações dinâmicas podem ter extremos declarados explicitamente ou calculados dinamicamente, o que permite que documentos adicionados a uma hiperbase tenham todas as suas ligações geradas automaticamente.

2. Flexibilidade

Novas ligações podem ser criadas sem necessidade de alterar o sistema ou suas estruturas de dados.

3. Portabilidade

Os tipos de ligações definidas podem ser usados em outros sistemas capazes de entender a linguagem na qual elas foram especificadas.

3.3 Modelos de Sistemas Hipermissão

Antes de descrevermos os sistemas hipermissão propriamente ditos, introduziremos o chamado *Modelo de Referência Dexter*. Esse é um modelo de dados e processos aqui usado como um padrão de referência a partir do qual diversos sistemas hipermissão serão comparados e contrastados.

3.3.1 Modelo de Referência Dexter

O Modelo de Referência Dexter [HS94] é uma tentativa de capturar, de uma maneira formal, as abstrações mais importantes encontradas em sistemas hipermissão. O modelo serve como um padrão para comparar diversos sistemas hipermissão ao mesmo tempo que oferece uma infra-estrutura para o desenvolvimento de padrões para o intercâmbio e a interoperabilidade entre esses sistemas. Ele é resultado de dois pequenos *workshops* em hipertexto, que reuniram discussões sobre os sistemas hipermissão mais representativos da

época. O primeiro ocorreu em 1988 em Dexter Inn (New Hampshire), dando origem a seu nome.

O modelo provê uma terminologia hiperfídia padrão e formaliza as abstrações encontradas em sistemas hiperfídia. Ele divide um sistema hiperfídia em três camadas: a *camada interior aos componentes*, a *camada de armazenamento* e a *camada de tempo de execução*.

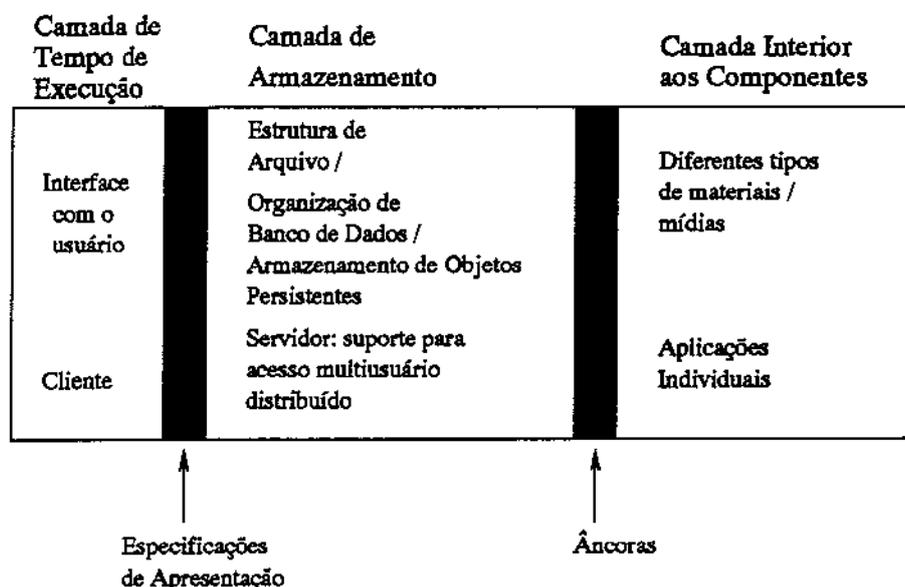


Figura 3.1: As Camadas do Modelo de Dexter e suas Interfaces

Camada Interior aos Componentes

Esta camada preocupa-se com o conteúdo e as estruturas dentro dos componentes da rede de nós e ligações, denominada rede hiperfídia. Componentes contêm os dados que formam o conteúdo básico da rede hiperfídia como, por exemplo, textos, gráficos, imagens e animações.

Essa camada é propositadamente não-elaborada dentro do modelo, devido à enorme variedade de possíveis tipos de dados passíveis de serem utilizados para descrever a estrutura interna e o conteúdo dos componentes. ODA e SGML são exemplos de padrões para a descrição desta estrutura e do conteúdo dos componentes.

Camada de Armazenamento

É o principal foco do modelo, responsável por modelar a estrutura da rede básica de nós e ligações. Esta camada exerce o papel de um “banco de dados” que é composto de uma hierarquia de componentes interconectados por ligações relacionais. A camada em questão possui ainda duas funções adicionais: uma função de mapeamento (denominada *resolver*) e uma função de acesso aos dados (*accessor*). Essas duas funções em conjunto

são responsáveis pela recuperação de componentes.

O componente é a unidade básica endereçável desta camada. Ele pode ser um *átomo*, uma *ligação* ou uma *entidade composta* por outros componentes. Cada componente na camada de armazenamento atua como um *container* genérico de dados, não havendo preocupação em descrever qualquer estrutura relativa à mídia armazenada nesse *container*. Todo componente tem um *identificador único (UID)*. A função de mapeamento é responsável por transformar uma especificação de componente em um UID correspondente. Este UID é passado à função de acesso que então recupera o componente especificado.

Ligações representam relacionamentos entre componentes e são formadas por duas ou mais especificações de extremos, cada uma das quais se referindo a um componente completo ou a uma parte dele.

Âncoras são um mecanismo de endereçamento indireto utilizado para endereçar localizações ou itens dentro de um componente individual. Elas são responsáveis por manter a separação lógica entre a camada de *armazenamento* e a camada *interior aos componentes*. Uma âncora é composta de duas partes: um *identificador* e um *valor*. O valor da âncora é arbitrário e especifica alguma localização, região, item ou subestrutura dentro do componente. Este valor é interpretável apenas pelas aplicações responsáveis por manusear o conteúdo/estrutura do componente. O identificador da âncora a identifica dentro do escopo de seu componente.

O mecanismo de identificação da âncora pode ser combinado com o mecanismo de especificação de componentes de forma a especificar os extremos de uma ligação. No modelo de Dexter, esta combinação é capturada por uma entidade chamada *especificador*. O especificador especifica um componente e uma âncora dentro deste componente que serve como extremo de uma ligação. O especificador determina ainda uma *direção* e uma *especificação de apresentação*. A direção indica se o extremo deve ser considerado origem ou destino de uma ligação, ambos ou nenhum deles (codificados respectivamente pelos valores FROM, TO, BIDIRECT e NONE). A especificação de apresentação é uma interface entre a camada de armazenamento e a de tempo de execução. Especificações de apresentação são um mecanismo que permite especificar, dentro da camada de armazenamento, como um componente deve ser apresentado ao usuário. Este conceito será discutido mais adiante ao tratarmos da camada de tempo de execução.

As definições anteriores permitem descrever a estrutura de uma ligação de forma mais precisa. Assim, uma ligação é constituída por dois ou mais especificadores, provendo à ligação uma aridade arbitrária (Figura 3.2). O campo *direção* do especificador é responsável por tratar os aspectos de direcionalidade da ligação. A única restrição imposta a essas direcionalidades é que pelo menos um especificador de uma ligação tenha uma direção TO ou BIDIRECT.

Tipos de componentes são introduzidos no modelo pela adição de um atributo *tipo* ao

componente. O valor deste atributo deve pertencer a um conjunto predefinido de *especificações de tipo*. Estas especificações contêm uma seqüência de âncoras, uma especificação de apresentação que indica à camada de tempo de execução como o componente deve ser apresentado, e um conjunto de pares arbitrários *atributo /valor* (Figura 3.2).

Em adição ao modelo de dados, a camada de armazenamento define um pequeno conjunto de operações para acessar e/ou modificar os dados e as estruturas hiperfídia. Essas operações abrangem inclusão e remoção de um componente, modificação do conteúdo ou da informação auxiliar de um componente (tais como suas âncoras e atributos), recuperação de um componente dado seu UID ou qualquer especificador que possa ser resolvido para esse UID, e operações para auxiliar a determinação da interconectividade da rede hiperfídia, dentre outras.

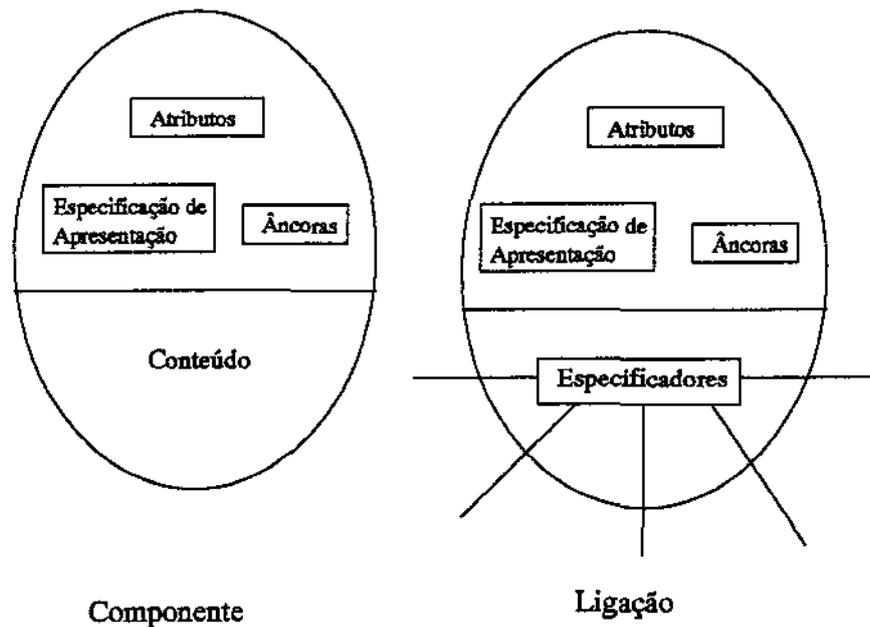


Figura 3.2: Estrutura dos Componentes na Camada de Armazenamento

Camada de Tempo de Execução

O conceito fundamental desta camada é o da *instanciação* de um componente. Uma instanciação é uma apresentação do valor do componente a partir de uma cópia deste componente. Esta cópia pode ser visualizada e/ou editada pelo usuário e as alterações são gravadas na camada de armazenamento. É permitida mais de uma instanciação simultânea do mesmo componente, sendo atribuído a cada uma dessas instanciações um *identificador de instâncias (IID)* diferente.

A instanciação de um componente implica a instanciação de suas âncoras. Uma âncora instanciada é denominada *marcador de ligação*. Este marcador é uma manifestação visível

daquela âncora em um documento apresentado. Este conceito é suportado no modelo de Dexter através da noção de *instanciação base*. A instanciação base modela uma instanciação como uma entidade complexa, contendo uma seqüência de marcadores de ligação e uma função que faz o mapeamento entre os marcadores e as âncoras que eles instanciam.

Uma entidade chamada *sessão* é incluída no modelo para manter o mapeamento entre os componentes e suas várias instanciações. Um usuário que pretende visualizar ou editar instanciações de componentes deve explicitamente abrir uma sessão. Instanciações são criadas através da ação chamada *apresentar o componente*, que permite ao usuário editar, modificar e destruir a instanciação.

No modelo de Dexter, a entidade sessão contém as estruturas hiperfídia sendo acessadas, um mapeamento dos IIDs das instanciações da sessão corrente para seus respectivos componentes, uma *história*, uma *função de mapeamento* e uma *função instanciadora*. A história é a seqüência de todas as operações executadas desde a abertura da sessão. A função de mapeamento mapeia especificadores para UIDs de componentes.

A *função instanciadora* da sessão é o coração da camada de tempo de execução. Ela é responsável por retornar uma instanciação do componente como parte da sessão, dados o UID do componente e uma especificação de apresentação. A especificação de apresentação contém informações que definem como o componente instanciado deve ser apresentado pelo sistema durante a instanciação.

A função instanciadora também é o cerne da operação *apresentar o componente*, a qual toma um especificador de componente (junto com a sessão e a especificação de apresentação) e chama a instanciadora usando o UID do componente. *Apresentar o componente*, por sua vez, é o cerne da operação *seguir a ligação*. *Seguir ligação* toma o IID de uma instanciação junto com o marcador de ligação contido dentro da instanciação e apresenta os componentes que são o destino daquela ligação.

3.3.2 Hypertext Abstract Machine (HAM)

O Hypertext Abstract Machine (HAM) [CG88] é um servidor multiusuário de propósito geral para um sistema de armazenamento hiperfídia. Ele foi o primeiro sistema que tentou prover alguma funcionalidade de bancos de dados hiperfídia para dar suporte ao desenvolvimento de aplicações.

Sendo uma máquina de armazenamento de baixo nível, o HAM provê um modelo de dados geral que pode ser usado para construir aplicações. Toda a informação sobre as aplicações descritas no modelo de HAM é armazenada em sistemas de arquivos da máquina servidora. Como muitos dos sistemas anteriores à definição do Modelo de Dexter, o HAM define suas próprias primitivas sem preocupação com qualquer padronização.

O modelo de dados de HAM possui cinco objetos básicos. Um *grafo* é um objeto de alto nível que contém os outros objetos HAM: contextos, nós, ligações e atributos. *Contextos*

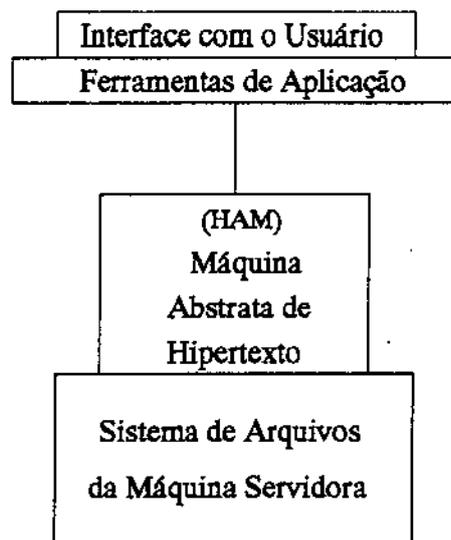


Figura 3.3: A Arquitetura de um Sistema de Hipertexto usando HAM

são usados para dividir os dados dentro do grafo. Um *nó* contém dados arbitrários. *Ligações* são relacionamentos entre nós. Contextos, nós e ligações podem conter atributos que são responsáveis pela descrição da semântica específica da aplicação.

O HAM provê ainda um mecanismo automático para o armazenamento da história das versões dos objetos (atualizada toda vez que o objeto é modificado), provê um mecanismo de filtro que permite extrair subconjuntos de objetos HAM a partir de grandes grafos e provê também mecanismos de controle de acesso opcionais que asseguram a segurança ao acessar os objetos.

O HAM oferece um conjunto básico de operações agrupadas em sete categorias distintas. Assim, existem operações para criar, eliminar e destruir objetos (elimina o objeto e a sua história); operações para modificar os dados associados aos objetos; operações para retornar dados de objetos e para recuperar informações sobre o grafo; e operações especiais, como busca em strings, que não se encaixam em nenhuma das categorias anteriores.

O suporte ao desenvolvimento de aplicações no HAM tem várias deficiências. A definição e a implementação de toda a funcionalidade específica da aplicação são deixadas completamente a cargo do desenvolvedor da aplicação. A única funcionalidade de SGBD considerada é a persistência dos dados. As demais são ignoradas.

3.3.3 Hyperbase

Hyperbase [SS90] é um sistema para autoria de hipermídia que já apresenta avanços em relação a HAM. Ele provê ao desenvolvedor de aplicações hipermídia uma interface de aplicação que atua como uma camada intermediária entre a aplicação e o SGBD (Sybase)

responsável pelo armazenamento persistente dos objetos. Para isso, Hyperbase define um modelo de dados independente da aplicação. Este modelo é responsável pela separação entre o armazenamento dos objetos, a sua interpretação e a sua apresentação. Contudo, a semântica dependente de aplicação dos objetos hiperfídia deve ainda ser definida como parte do programa de aplicação.

Muitas das idéias de Hyperbase são baseadas nas idéias de HAM e do modelo de Dexter. O modelo de dados de Hyperbase, como em HAM, contém nós, ligações, composições e atributos. A maior diferença entre as abordagens de Hyperbase e a de HAM está na forma de tratamento de coleções. Em HAM, coleções são chamadas contextos. Uma importante função destes contextos é manter uma história das versões dos objetos. Em Hyperbase, diferentemente de HAM, coleções são consideradas objetos de primeira classe, podendo ser referenciadas por ligações ou podendo estar contidas em outras coleções. Elas podem também ser derivadas de várias coleções ao mesmo tempo (e.g., como o resultado de uma consulta sobre um conjunto de coleções).

Com relação a Dexter, Hyperbase compartilha da preocupação no tratamento da camada de armazenamento e deixa o conteúdo/estrutura interna dos nós (camada interior aos componentes) e a apresentação (camada de tempo de execução) para serem tratados fora do modelo de dados.

No que diz respeito à camada interior aos componentes, a principal diferença entre os modelos de Hyperbase e de Dexter é que este último suporta ligações com múltiplas origens e destinos enquanto Hyperbase provê apenas ligações binárias. Ao contrário de Dexter, Hyperbase suporta o armazenamento da história de um componente. Por último, apesar da similaridade entre as primitivas de coleções de Dexter e de Hyperbase, este último define as operações disponíveis para as coleções, o que não ocorre em Dexter.

No nível conceitual, a principal diferença entre Dexter e Hyperbase é que Dexter constrói seu modelo usando algumas primitivas com domínios não-especificados (UIDs, Especificadores e Valores de Âncoras) enquanto que em Hyperbase são explicitados todos os domínios para cada abstração introduzida.

3.3.4 Hyperform

Hyperform [WL92] é um sistema projetado para o desenvolvimento de bases de dados hiperfídia (hiperbases). Ele provê uma linguagem orientada a objetos que permite adicionar, em tempo de execução, novos objetos e operações à hiperbase.

A idéia básica que se encontra por trás de Hyperform é a de extensibilidade. Para o alcance desse objetivo, Hyperform provê um conjunto de classes internas, usadas para enriquecer um modelo de dados hiperfídia com funcionalidades de bancos de dados.

O modelo de dados de Hyperform possui três classes básicas (figura 3.4): *MetaClasse*, *Objeto do Sistema*, e *Objeto*, e cinco subclasses de *Objeto*. *MetaClasse* provê operações

básicas para operar na hierarquia de classes. A classe Objeto do Sistema modela facilidades do sistema, como acesso ao banco de dados, caching, permissões para a criação de classes, grupos de usuários, etc. A classe Objeto modela os objetos do sistema hiperfídia que consistem basicamente de um conjunto de atributos *default*, alguns atributos alocados dinamicamente e um conjunto de operações (métodos) sobre esses atributos.

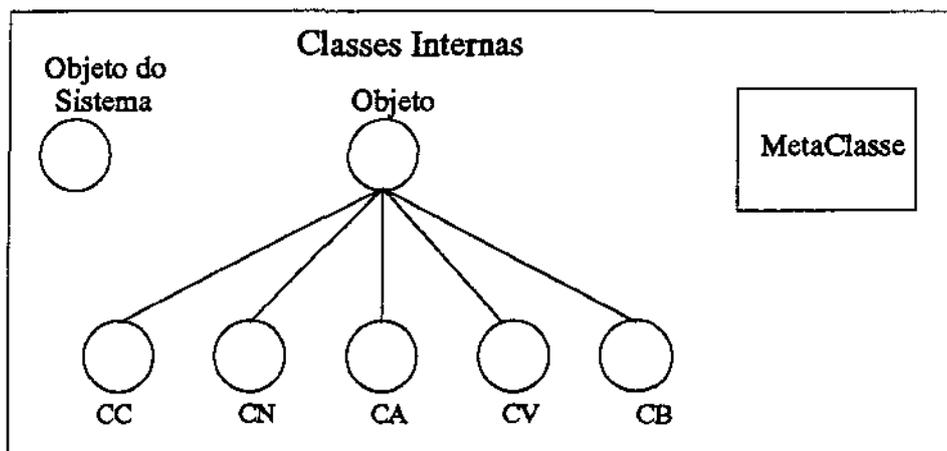


Figura 3.4: A Hierarquia das Classes Internas de Hyperform

As cinco subclasses de Objeto provêm a funcionalidade básica para cinco das principais características de aplicações hiperfídia: controle de concorrência (CC), controle de notificação (CN), controle de acesso (CA), controle de versão (CV), consulta e busca (CB). Através de especializações, o desenvolvedor da aplicação cria as classes necessárias de forma a especificar o modelo de dados da aplicação, e pode, por meio de herança múltipla, enriquecer o modelo de dados com a funcionalidade provida por estas cinco subclasses de Objeto.

As facilidades oferecidas por Hyperform podem ser usadas, por exemplo, para simular HAM, usando as subclasses de Objeto para criar o modelo de dados de HAM. O passo final é criar uma biblioteca C que proverá a interface HAM.

Comparado com HAM, Hyperbase e outros sistemas dessa geração, Hyperform apresenta uma série de vantagens. Em primeiro lugar, Hyperform não possui um modelo de dados hiperfídia fixo. Ao contrário, ele provê um conjunto de blocos de construção que pode ser estendido para prover virtualmente qualquer tipo de suporte a uma hiperbase. Além disso, Hyperform também não possui nenhuma política de projeto específica, o que permite prototipação rápida. O uso de orientação a objetos na modelagem dos dados hiperfídia permite uma implementação bem mais natural dos conceitos de extensibilidade e de direcionamento da aplicação. Outra vantagem de Hyperform é a incorporação de funcionalidades de bancos de dados, tais como controle de concorrência e controle de versionamento ao sistema hiperbase.

Alguns problemas, contudo, ainda persistem. Em Hyperform, o modelo de dados da aplicação tem que ser definido completamente pelo desenvolvedor. Não existem operações para a definição e manutenção de restrições estruturais entre os tipos de objetos hiperfídia. Outro problema é que Hyperform não incorpora realmente um SGBD ao sistema. Devido às suas características orientadas a objetos, Hyperform é capaz de oferecer meios de integrar ou mesmo de estender funcionalidades de SGBD pré-definidas. Contudo, ele não força as aplicações hiperfídia a fazerem isso.

3.3.5 HB3

HB3 [LS94] é um sistema de armazenamento hiperfídia multiusuário projetado para atender às necessidades de armazenamento de SP3. O SP3 é um sistema hiperfídia que suporta modelos baseados em processos como, por exemplo, o modelo de Dexter, tendo por objetivo prover funcionalidades hiperfídia a um conjunto de aplicações em um ambiente de rede.

Na arquitetura do SP3, o modelo de dados hiperfídia do sistema é localizado fora do HB3. Este modelo (figura 3.5) consiste de seis elementos: aplicações, componentes, seleções persistentes, âncoras, ligações e associações. *Aplicações* são programas. *Componentes* têm a mesma semântica de Dexter. *Seleções persistentes* representam seleções dentro dos componentes que persistem entre sessões de aplicações, sendo semelhantes às âncoras de Dexter. *Âncoras e ligações* são processos do sistema operacional responsáveis pela implementação dos comportamentos navegacionais do sistema. *Âncoras* são associadas a seleções persistentes e *ligações* são associadas a âncoras, definindo desse modo conexões entre seleções persistentes. *Associações* são coleções de especificadores, sendo semelhantes às composições de Dexter. Associações são coletadas em *conjuntos de associações* que são usados para oferecer a noção de contextos às operações hiperfídia.

O HB3 é implementado como um conjunto de gerenciadores: o *gerenciador de conjuntos de associações*, o *gerenciador de objetos versionados* e o *gerenciador de armazenamento*. O gerenciador de conjuntos de associações provê o armazenamento persistente e compartilhável para associações e para conjuntos de associações. O gerenciador de objetos versionáveis fornece uma história das versões para todos os objetos do repositório. O gerenciador de armazenamento mapeia o modelo de dados de HB3 para o armazenamento físico. As funcionalidades de bancos de dados residem dentro de um SGBD subjacente, acessível a partir de HB3 via o gerenciador de armazenamento.

Qualquer modelo de dados hiperfídia pode ser definido usando o HB3 através da interação com o gerenciador de *serviços de ligações*. Este gerenciador é o responsável pelo modelo de dados de SP3 e provê o suporte necessário para implementar as funcionalidades hiperfídia entre as aplicações do ambiente. Este suporte a ligações interaplicações é alcançado através de diversas atividades tais como: conexão e desconexão de âncoras

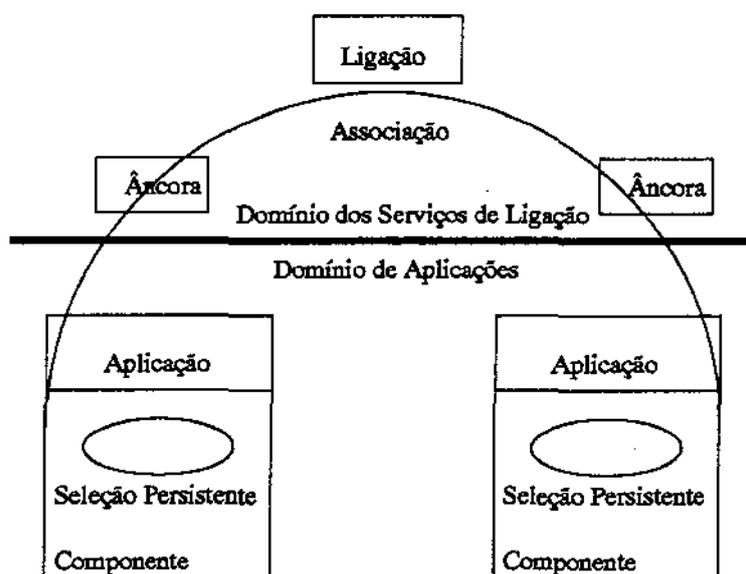


Figura 3.5: O Modelo de Dados da Arquitetura SP3

e ligações; direcionamento de requisições para as aplicações mostrarem marcadores de âncoras e ligações; definição de operações de folheamento e de operações para construção de associações e de conjuntos de associações.

3.3.6 HyperStorm

O HyperStorm [BWAH96] é uma máquina hipermídia extensível que suporta a especificação da semântica de aplicações através da definição de classes de aplicação dentro de um Sistema Gerenciador de Banco de Dados Orientado a Objetos. Esta abordagem suporta a definição de operações para a manutenção de restrições complexas (dependentes e independentes da aplicação) entre os objetos da aplicação. Como as funcionalidades hipermídia da aplicação são implementadas no SGBDOO, as operações de manutenção de restrição são executadas pelo sistema gerenciador de banco de dados.

Em HyperStorm, tanto a camada de armazenamento quanto a camada de aplicações do sistema hipermídia são implementadas dentro do SGBDOO VODAK. Apenas a camada de apresentação é deixada a cargo da aplicação. Isso permite o desenvolvimento de diferentes interfaces de usuários para uma mesma aplicação hipermídia. A abordagem de HyperStorm permite que aplicações usem funcionalidades de bancos de dados (e.g., controle de concorrência, recuperação de falhas, consultas), além de oferecer a flexibilidade necessária para suportar diferentes aplicações sobre um mesmo SGBD Hipermídia.

A filosofia do HyperStorm é dar suporte à modelagem de relacionamentos semânticos entre classes de aplicação hipermídia através do desenvolvimento de uma camada de

metaclasses (figura 3.6). Existem, em HyperStorm, metaclasses para prover objetos hipermissão básicos como nós, ligações, composições e objetos atômicos. Estas metaclasses incluem operações complexas (métodos) para manter a consistência de hiperdocumentos e podem ser direcionadas pelo projetista para atender às necessidades específicas de uma aplicação. Isto é alcançado pela classificação de objetos em tipos específicos da aplicação, pelas especializações das metaclasses e pela definição de restrições de integridade da aplicação.

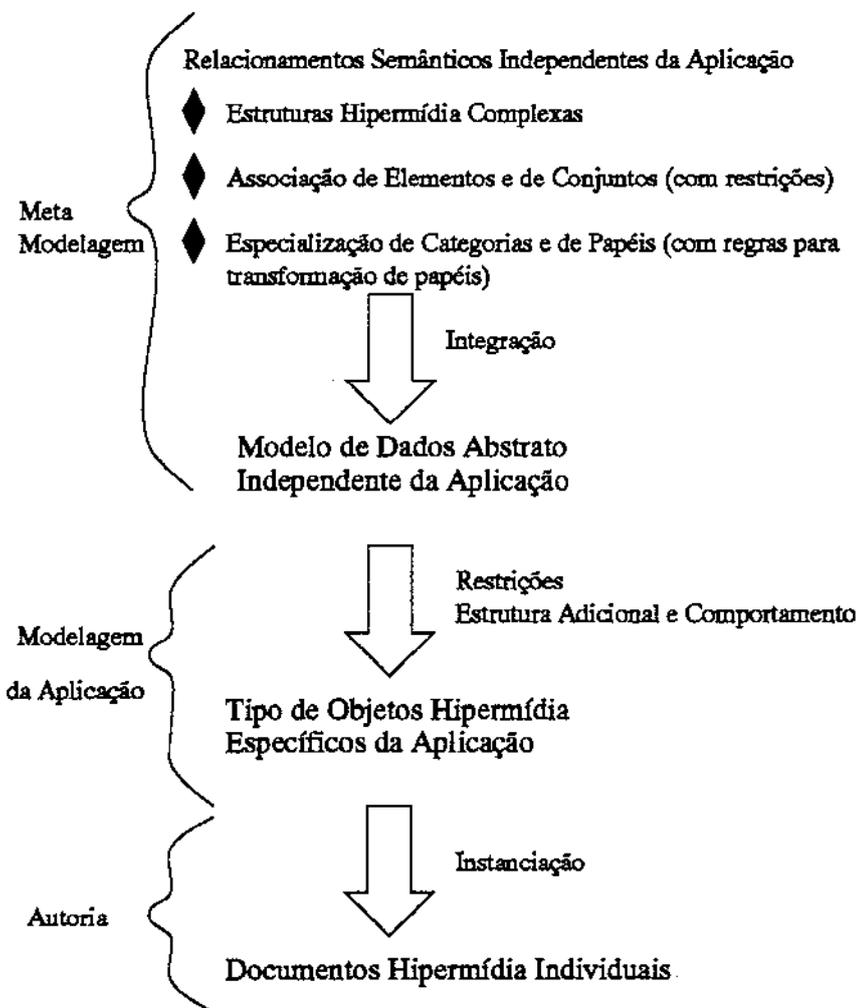


Figura 3.6: Visão Geral da Abordagem seguida no Projeto de HyperStorm

Dentre os relacionamentos semânticos básicos providos pelas metaclasses HyperStorm, podemos citar a definição de restrições para a construção de estruturas hipermissão complexas (e.g., diferentes tipos de ligações dentro de uma mesma hierarquia hipermissão), a definição de restrições para associações de elementos e de conjuntos que modelam composições, e a definição de especializações de categoria e de papéis (com regras para

transformação de papéis que permitem usar os mesmos objetos hipermídia em diferentes contextos).

3.3.7 Sistema Gerenciador de Banco de Dados OOHDM

O SGBD OOHDM [MSL96] compartilha os mesmos objetivos e aplica uma abordagem bastante semelhante à de HyperStorm para o desenvolvimento de um ambiente de implementação de aplicações hipermídia. Este trabalho também propõe um metaesquema que é implementado sobre um SGBDOO. O metaesquema pode ser estendido, via especializações, para a implementação da semântica específica de uma aplicação.

O trabalho em questão difere das propostas anteriores porque a aplicação hipermídia pode ser desenvolvida explorando-se unicamente as ferramentas de um SGBDOO. Diferentemente de HyperStorm, todas as camadas de um sistema hipermídia, inclusive a de apresentação, são levadas para dentro do SGBDOO, sendo este o responsável pelo gerenciamento destas camadas e das suas estruturas.

Outra característica diferenciadora e extremamente relevante deste trabalho é que o metaesquema proposto é baseado num mapeamento dos conceitos de um método de autoria, o OOHDM, para o esquema do banco de dados. As aplicações desenvolvidas neste método de autoria podem ser diretamente implementadas no SGBDOO através de simples extensões das metaclasses. O OOHDM e a metodologia associada são discutidos com mais detalhes na seção 3.4.4

3.3.8 Modelos Abertos de Ligações Dinâmicas

Nesta subseção são analisadas as principais propostas encontradas na literatura que implementam de forma parcial ou total as noções de ligações dinâmicas e de abertura.

Microcosm

O modelo Microcosm [DHH⁺92, LDH92] é baseado em um conjunto de processos autônomos que se comunicam entre si através de mensagens, preocupando-se com a questão da abertura do sistema, não impondo qualquer informação adicional aos dados. O sistema gera informações sobre âncoras e sobre ligações e permite a navegação em grandes volumes de dados enquanto provê facilidades de integração de ferramentas do sistema operacional subjacente.

O modelo de “seleção/ação” é o paradigma básico da navegação no sistema. O usuário seleciona dados (um trecho de texto, por exemplo) e executa alguma ação sobre eles. Um botão simplesmente amarra uma seleção específica a uma ação particular (geralmente “seguir a ligação”).

Visualizadores são utilizados para a interação com o usuário. Filtros são usados para manipular mensagens entre os visualizadores e o Microcosm. Alguns filtros que são providos por Microcosm incluem:

1. Filtros “Bancos de Dados de Ligações”

São responsáveis por armazenar toda informação referente às ligações.

2. Filtros “Mostre as Ligações”

Organizam o “*display*” de todas as ligações disponíveis quando o usuário seleciona um trecho de texto e usa a ação “Mostre as Ligações”.

3. Filtros “Compute Ligações”

Permitem ao usuário organizar um conjunto de arquivos de texto e indexá-los. A partir desta indexação, um bloco de texto pode ser selecionado e a ação “compute a ligação” pode ser escolhida. O sistema retornará um conjunto de outros documentos que tem um vocabulário similar ao do bloco selecionado.

O aspecto mais importante do sistema é a característica dinâmica de suas ligações. Existem três tipos básicos de ligações em Microcosm:

1. Específica

Permite ao usuário seguir a ligação após ter selecionado uma âncora numa dada localização específica em um documento corrente

2. Local

Permite ao usuário seguir a ligação após ter selecionado uma dada âncora em qualquer ponto do documento corrente.

3. Genérico

Permite ao usuário seguir uma ligação após ter selecionado uma dada âncora em qualquer ponto de qualquer documento.

Estes tipos de ligações possuem destinos estáticos. No entanto, ligações locais e genéricas têm âncoras com origem dinâmica. As ligações locais e genéricas permitem que o destino de uma ligação seja fixado apenas uma vez e, subsequentemente, essa ligação poderá ser seguida de qualquer objeto-origem apropriado. Qualquer documento criado pelo autor *a posteriori* tem acesso imediato a todas as ligações genéricas que tenham sido definidas para o sistema.

As ligações podem ser criadas manualmente ou computadas. Os três tipos de ligação podem ser criados manualmente, sendo que as ligações locais e genéricas requerem bem menos esforço por parte do autor. Ligações computadas, cuja origem e/ou cujo destino não foram definidos até que a ligação seja seguida, podem ser automaticamente geradas pelo sistema. Uma vez geradas, ligações podem ser incorporadas ao banco de dados de ligações.

Ligações específicas, como seu próprio nome sugere, fornecem o acesso mais específico aos dados. Estas ligações são claramente definidas entre dois pontos determinados e fixos de informação. Contudo, tais ligações requerem um considerável esforço manual para sua geração. Ligações genéricas e locais oferecem uma precisão menor na qualidade da informação retornada, mas reduzem o esforço de criação do autor. Este tem de se preocupar em gerar apenas a definição do destino. A ligação, então, estará disponível para o usuário em qualquer documento específico ou genérico adequado.

Outros tipos de ligações incluem ligações de relevância (que ligam documentos indexados por similaridade) e ligações de recuperação de texto (dinamicamente computadas quando são requisitadas, fazendo uso de primitivas do sistema ou de métodos de recuperação de informação).

DeVise Hypermedia Model - DHM

DHM [GT94] é um sistema aberto baseado no Modelo de Dexter, buscando fornecer soluções para as lacunas desse modelo.

O DHM possui um projeto de componentes orientados a objeto que suporta ligações computadas e estáticas, bem como a noção explícita de ligações *dangling*. Estas ligações têm zero ou um extremo, relaxando a semântica do modelo de Dexter, que obriga uma ligação a ter pelo menos dois extremos. As ligações *dangling* podem aparecer como resultado de algum processo sobre os objetos da hipermedia como, por exemplo, a eliminação de âncoras ou a indisponibilidade de objetos relevantes referidos pelo conteúdo do componente.

O DHM considera três noções para direcionalidade das ligações: *direção semântica*, que representa um relacionamento semântico entre os componentes; *direção de criação*, correspondendo à ordem segundo a qual os extremos foram criados; e *direção de percurso*, que especifica como a ligação pode ser percorrida - se apenas da origem para o destino, o inverso ou ambos.

A direcionalidade em DHM é parcialmente baseada na proposta de Dexter e suporta explicitamente as ligações de criação e de percurso. Ao criar uma ligação, adicionar ou editar extremos, o usuário pode escolher explicitamente se o extremo é origem, destino ou ambos. Isso equivale a definir FROM, TO ou BIDIRECT como valores do atributo direção em um especificador de Dexter. Por *default*, extremos criados com o uso da

operação “*New Link*” são origens e aqueles criados com “*Add Endpoints*” são destinos.

O DHM estende o modelo de âncora de Dexter de diversas formas. Primeiro, ele usa referências dinâmicas (ponteiros) ao invés de IDs de âncoras, o que evita a necessidade de utilizar a função de acesso aos dados, pois o especificador da ligação aponta diretamente para as âncoras do componente. A utilização de um SGBDOO no papel do banco de dados da rede hiperfídia constitui uma extensão ao modelo de Dexter e permite a persistência dos ponteiros das âncoras.

DHM distingue basicamente três tipos de âncoras, independentes do tipo de componente que as inclui: âncoras para componente-completa, âncoras marcadas, e âncoras não marcadas. Âncoras para *componente-completa* suportam o caso de extremos de ligações não ancoradas dentro de um componente. Todas as ligações com extremos componente-completa compartilham uma única âncora que aponta para aquele componente por inteiro.

Âncoras *marcadas* são associadas com objetos particulares inseridos no conteúdo do componente, sendo equivalentes ao conceito de âncora de Dexter. Esta âncora pode “rastrear” mudanças de edição no componente e na própria seleção do conteúdo. A instanciação decide se torna visível ou não o marcador da âncora para o usuário. Um *click* na região invoca a ação de seguir as ligações da correspondente âncora.

Âncoras *não marcadas* requerem uma busca no conteúdo do componente (e.g., por palavra-chave). A sua localização dentro de um componente não é definida *a priori* e é sempre computada. Este tipo de âncora fornece a abstração necessária às ligações dinâmicas. O que distingue basicamente uma âncora não marcada de uma marcada é a sua habilidade de recuperar âncoras diretamente de uma seleção em um editor de componentes.

DHM também estende algumas características de componentes e composições no modelo de Dexter, suportando construções complexas para aplicações particulares. Existem várias dimensões de componentes oferecidas que podem ser utilizadas pelo usuário ao adicionar novos componentes:

- componentes virtuais/ não virtuais

Qualquer componente pode transformar-se ou ser definido como virtual setando um *flag* apropriado. Tais componentes assemelham-se a componentes normais, mas só são salvos no banco de dados se referenciados por qualquer outro componente (digamos, uma ligação). Por exemplo, um componente virtual pode ser criado automaticamente para mostrar os resultados de uma busca executada pelo usuário sobre os componentes da hiperfídia.

- Componentes computados/estáticos

Qualquer tipo de componente pode ser resultado de uma computação, ao invés de ser manualmente criado pelo usuário. Um exemplo típico é o componente criado

através da avaliação de uma consulta. Um atributo contém a informação para executar a computação, que pode ser depois recomputada, ou por demanda, ou periodicamente.

- Conteúdo de componentes

Componentes podem incluir objetos externos ou referências a outros componentes. Um exemplo são *browsers* de ligações, que são implementados como composições virtuais, computadas, consistindo de listas de referências para ligações.

Finalmente, o DHM oferece um conjunto de estruturas para componentes e composições mais rico que Dexter.

Modelo Funcional da Ligação

O Modelo Funcional da Ligação (*Functional Model of the Link* [AV94, AVC94]) é um modelo formal projetado para descrever a ligação hiperfídia. Ele não impõe quaisquer restrições à implementação da ligação e é completamente independente do Sistema de Gerenciamento Hiperfídia, do armazenamento da ligação, da plataforma, e do sistema operacional subjacentes.

O modelo funcional expressa um tipo de ligação hiperfídia como regras a partir das quais todas as instâncias deste tipo podem ser construídas. As computações definidas nas regras da ligação denotam um relacionamento semântico entre conjuntos de componentes, bem como dão uma especificação flexível para a relação matemática projetada pelo tipo da ligação.

Como no modelo de Dexter, o termo *componente* é utilizado para qualquer unidade de informação que possa ser mostrada de forma independente. A âncora de Dexter recebe o nome de *ponto de conexão*. Um *ambiente* é um conjunto de fatores que influenciam a interpretação ou a disponibilidade da ligação.

A especificação de uma ligação consiste de um par de funções (s,r) . O *predicado origem* s é uma função que determina se o ponto de conexão candidato no ambiente corrente participa ou não daquela ligação. Se o ponto de conexão candidato satisfaz às condições embutidas no predicado, então aquela ligação está disponível a partir do ponto de conexão. A ligação é resolvida através de uma chamada à função *resolução* r que calcula os pontos de conexão destino apropriados para aquela origem no ambiente corrente. Alguns tipos de ligações não são influenciados pelo ambiente corrente, mas existem outros que podem depender de fatores de ambiente como tempo, privilégios de usuários e história de navegação do usuário.

O predicado origem e a função resolução são a especificação necessária e suficiente do comportamento navegacional de qualquer ligação. As ligações genéricas de Microcosm,

por exemplo, podem ser definidas no arcabouço do modelo funcional. O predicado origem da ligação envolve uma busca por string para verificar se um dado texto selecionado possui alguma ligação definida para ele. Este texto selecionado atua como o ponto de conexão para uma ligação. A função resolução é uma função de enumeração explícita dos destinos da respectiva ligação.

Outro exemplo do uso do modelo funcional para expressar ligações são as ligações *implícitas* do Geographic Hypermedia Information System descrito em [AC93]. Estas ligações implícitas são definidas entre tabelas de bancos de dados e dados externos, e são baseadas em dados espaciais e não-espaciais armazenados no banco. Por exemplo, as cidades de um mapa podem ser conectadas a arquivos multimídia, sendo a ligação resolvida dinamicamente entre tabelas de cidades do banco de dados e arquivos multimídia.

3.4 Modelos para Autoria de Aplicações Hiperfídia

A seção anterior descreveu modelos de dados de sistemas hiperfídia, com ênfase no suporte ao desenvolvimento de aplicações, no suporte a ligações estáticas e dinâmicas, bem como em noções de abertura. Modelos de autoria são na verdade modelos de dados e metodologias para desenvolvimentos de aplicações hiperfídia necessários principalmente no caso de aplicações grandes e complexas, como é o caso de bibliotecas digitais. Do ponto de vista da tecnologia de bibliotecas digitais, as características a serem avaliadas incluem a capacidade de modelar os dados de uma biblioteca digital e a maneira de modelar a navegação e os aspectos de interface da biblioteca.

No restante desta seção, apresentaremos algumas das principais abordagens de projeto para aplicações hiperfídia surgidas nos últimos anos.

3.4.1 Hypertext Design Model (HDM)

O HDM [GPS93, GMP95, GMP94b] é uma abordagem baseada em modelo para projetar aplicações hiperfídia de larga escala. Ele foi o primeiro modelo que se preocupava unicamente com a questão do projeto de aplicações, provendo diversas abstrações para este fim. Estas abstrações permitem a descrição de diversas classes de elementos de informação e de estruturas navegacionais para aplicações complexas.

Como qualquer modelo para descrever aplicações, HDM é constituído por um conjunto de primitivas. *Unidades* são a menor porção de informação e são os repositórios efetivos desta informação. Unidades são agrupadas em *componentes* que, por sua vez, são organizadas em hierarquias semânticas, formando *entidades*. Uma aplicação HDM é definida em termos destas entidades, que correspondem aos objetos físicos ou conceituais do domínio da aplicação.

Cada unidade apresenta o contedo de um componente sob uma *perspectiva* particular. Perspectivas s3o dispositivos sint3ticos que organizam a informa33o. Unidades possuem uma rela33o um-para-um com cada perspectiva especfca. Uma unidade 3 caracterizada por um *nome* (seu identificador) e por um *corpo*. Corpos de unidades contem os dados da aplica33o HDM. Como exemplos de unidades, podemos ter “Musical/som” e “Musical/texto” que permitem ao leitor escolher entre ouvir ou ler a letra da Mfscial.

As estruturas podem ser interconectadas por *liga33es*. Existem basicamente tr3s tipos de liga33es. *Liga33es Estruturais* conectam componentes que pertencem a uma mesma entidade. *Liga33es de Perspectivas* conectam as diferentes unidades que correspondem a um mesmo componente. *Liga33es de Aplica33o* representam relacionamentos dependentes de domfnio e conectam componentes ou entidades, do mesmo ou de diferentes tipos, em padr3es arbitr3rios definidos pelo autor. Liga33es de aplica33o s3o organizadas em tipos que s3o especificados por um nome, um conjunto de tipos de entidades origem e outro de entidades destino, al3m de um atributo *simetria* indicando se o tipo da liga33o tem uma inversa. As liga33es de perspectiva e a maioria das liga33es estruturais n3o necessitam ser definidas explicitamente pelo autor. Elas podem ser derivadas automaticamente da estrutura da entidade.

Como qualquer modelo de projeto, HDM faz uma distin33o clara entre as no33es de *Esquema* e a de *Inst3ncia do Esquema*. O *Esquema da Aplica33o* 3 um conjunto de defini33es de tipos que o descrevem em um nfvvel global. A *Inst3ncia do Esquema* 3 uma cole33o de entidades, componentes, unidades e liga33es que satisfazem as defini33es do esquema.

Em HDM, uma aplica33o hiperfídia 3 constitufda por uma *hiperbases* e por um conjunto de *estruturas de acesso*. A hiperbase 3 o centro da aplica33o e 3 constitufda a partir das primitivas definidas anteriormente. As estruturas de acesso s3o respons3veis por oferecer ao usu3rio uma vis3o da hiperbase e por permitir-lhe selecionar pontos de entrada para a navega33o. HDM oferece duas primitivas de acesso: os *fndices* e os *guided tours*. Um fndice atua como uma tabela cujas entradas apontam para um conjunto de inst3ncias de entidades, provendo acesso direto a cada um dos itens listados, e fornecendo tamb3m liga33es de retorno de cada participante para o fndice. O fndice pode ainda referenciar outros fndices, gerando fndices aninhados. O *guided tour* implementa um caminho linear atrav3s de uma cole33o de itens, permitindo ao usu3rio navegar para frente e para tr3s no caminho.

O uso da aplica33o 3 definido por sua “*sem3ntica de folheamento*”. Esta sem3ntica ser3 respons3vel por determinar a forma como as estruturas de informa33o ser3o visualizadas e por definir como o usu3rio poder3 navegar atrav3s delas.

Diversas sem3nticas de folheamento podem ser aplicadas sobre uma mesma especifica33o est3tica HDM, uma vez que os mapeamentos das primitivas HDM para as estru-

turas de implementação tenham sido providos. Isso gera a noção de diferentes *versões* da aplicação. Cada uma dessas semânticas é caracterizada por aspectos visuais e comportamentos dinâmicos diferentes, podendo rodar em sistemas hipermissão distintos. HDM possui uma semântica de folheamento *default* baseada nos conceitos de nós e de ligações binárias como sendo os objetos perceptíveis ao usuário.

Uma importante abstração introduzida no modelo HDM em [GMP94a] foi o conceito de coleções. Uma coleção é um *container* que inclui diversos nós. Nós são inseridos em coleções por critérios semânticos ou por necessidade de organização de uma apresentação específica. Coleções permitem uma modificação contínua da organização e da apresentação de uma aplicação, sem ter que alterar a informação representada pelos nós.

Coleções podem incluir outras coleções como membros. Uma coleção pode conter ainda informação adicional (e.g., nome, texto introdutório, figuras introdutórias). Essa informação é colocada em um *nó-coleção*.

Existem três operações necessárias para se criar uma coleção: definição do conjunto de membros, usando diferentes métodos pré-definidos; definição da estrutura interna da coleção; e definição do nó associado à coleção. Esse nó-coleção associado desempenha três papéis distintos: ajuda o leitor a entender o conteúdo da coleção; provê informação adicional para a coleção; e fornece mecanismos para acessar os membros da coleção.

3.4.2 Enhanced Object-Relationship Model (EORM)

O *Enhanced Object Relationship Model (EORM)* [Lan94] é uma abordagem orientada a objetos para o projeto de aplicações voltadas a sistemas de informação hipermissão. A abordagem utiliza como base uma metodologia de análise orientada a objetos - a *Object Modeling Technique (OMT)* - e estende essa metodologia com construções para capturar a semântica das interações entre objetos.

A idéia central da metodologia é representar explicitamente relacionamentos semanticamente ricos no modelo orientado a objetos. Estes relacionamentos são definidos como classes que têm estrutura e comportamento próprios. Desta forma, os relacionamentos são capazes de capturar não apenas as restrições estruturais básicas dos relacionamentos OMT, mas também complexas propriedades destes relacionamentos, tais como a semântica de ligações hipermissão. A semântica hipermissão de um relacionamento é modelada em uma aplicação definindo-se classes e hierarquias de ligações e associando as semânticas dessas ligações aos relacionamentos.

O método de projeto EORM consiste em três componentes (figura 3.7): *o arcabouço de classes, o arcabouço de composição e o arcabouço GUI*. O arcabouço de classes é formado por uma biblioteca de definições de classes reusáveis advindas da modelagem orientada a objetos do domínio da aplicação.

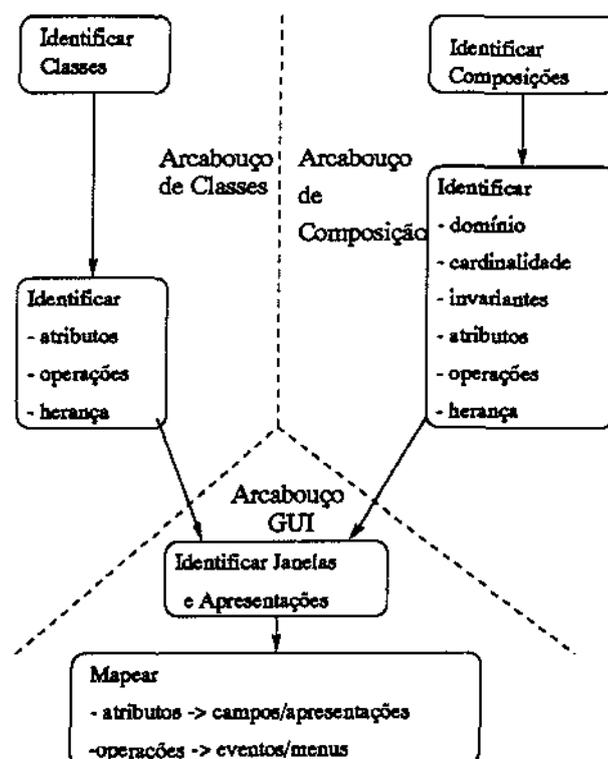


Figura 3.7: Visão Geral do Método de Projeto EORM

O arcabouço de composição consiste na especificação de uma biblioteca de definições de composições. A construção deste arcabouço envolve duas atividades: a) a identificação de composições; e b) o refinamento das composições identificadas. A primeira atividade identifica as composições dentro do domínio da aplicação. Composições são constituídas por um relacionamento, pelas classes participantes deste relacionamento e suas cardinalidades, e por uma classe de ligação especificando a semântica de navegação do relacionamento. A segunda atividade envolve o refinamento das composições identificadas através da definição explícita dos domínios e das cardinalidades dos participantes, pela definição das restrições de integridade e pela definição da semântica da ligação associada à composição. A definição da semântica das ligações resulta em uma hierarquia de classes de ligações definidas com base em critérios tais como direção, propriedades estáticas e dinâmicas, densidade da população dos conjuntos conectados, comportamento, dentre outros critérios.

O arcabouço GUI é formado por uma biblioteca reusável de definições de apresentação. As atividades relacionadas a esse arcabouço são: a) a identificação das janelas e da apresentação; e b) o mapeamento de classes e composições para apresentações. Uma apresentação consiste de um ou mais *widgets* (e.g., strings, botões) no qual um objeto pode apresentar seu conteúdo. A definição de uma apresentação envolve ainda a especificação

de eventos e de funções que podem ser associadas a esses eventos. Uma janela mostra uma apresentação na tela. Janelas podem conter também menus que são usados para assinalar comportamento à janela, fazendo-a aparecer e atuar como um programa de aplicação independente. A atividade de mapeamento considera como os atributos e as operações de classes são mapeados para a apresentação, como os eventos são conectados às operações e como a funcionalidade é assinalada aos menus.

3.4.3 Relationship Management Methodology (RMM)

A Metodologia de Gerenciamento de Relacionamentos (Relationship Management Methodology (RMM)) [ISB95] é um conjunto de técnicas e um modelo formal para o projeto e construção de aplicações hiperfídia. RMM é particularmente útil para projetar aplicações que exibem uma estrutura regular para um determinado domínio de interesse.

A metodologia inclui seqüência recomendada de diversos passos, formalismos para definição de estruturas de acesso adicionais, ênfase nas representações gráficas e procedimento detalhado, passo-a-passo, para o projeto e desenvolvimento de hiperfídia.

O modelo de dados de RMM, denominado RMDM (Relationship Management Data Model), é parcialmente baseado em HDM e em modelos de banco de dados. O RMDM é constituído por um conjunto de *primitivas de domínio*, as quais modelam a informação sobre o domínio da aplicação. *Entidades* RMDM são equivalentes às entidades de HDM, não havendo, contudo, as noções de componentes, unidades e perspectivas. A noção de *tipos de entidades* é a comumente utilizada em banco de dados. *Slices* são agrupamentos de atributos, definidos como subdivisões de entidades, de forma a tornar a apresentação da entidade mais prática e legível. Uma entidade pode ser apresentada por diversos *slices*. *Relacionamentos associativos* descrevem associações entre diferentes tipos de entidades. Eles podem ser *um-para-um* ou *um-para-muitos*. Na fase de *Projeto Navegacional* da metodologia, esses relacionamentos serão transformados em ligações semanticamente equivalentes às ligações de aplicação HDM, sendo estas últimas acessadas através de *primitivas* ou *estruturas de acesso*.

A navegação é suportada em RMDM por seis primitivas de acesso. As *ligações unidirecionais* e *bidirecionais* são usadas para especificar acessos entre *slices* de uma mesma entidade, sendo utilizadas para navegação dentro desta entidade. Essas ligações correspondem às ligações estruturais de HDM. A navegação através de diferentes entidades é suportada por *índices*, *guided tours* e *agrupamentos*. Índices e *guided tours* são equivalentes às respectivas primitivas de acesso HDM. *Guided tours condicionais indexados* combinam os paradigmas de navegação de ambas as primitivas numa estrutura mais rica. Um agrupamento é um mecanismo baseado em menus que permite acessar outras partes da hiperfídia. O diagrama RMDM completo descreve como os usuários irão navegar dentro da aplicação hiperfídia.

A metodologia RMM é mostrada na figura 3.8. O foco principal de RMM está nas fases de projeto, de desenvolvimento e de construção, representadas na figura pela área envolvida pelo retângulo tracejado. As construções geradas a cada passo da metodologia são representadas na figura pelos *labels* das setas. *Loops de feedback* (na figura representadas por linhas tracejadas) podem existir em qualquer situação. Descrevemos a seguir um esboço da metodologia e algumas diretrizes de projeto para os principais passos.

Passo S1: Projeto Entidade-Relacionamento

Neste primeiro passo, o domínio da aplicação é descrito através de um diagrama (E-R). Modelos E-R já disponíveis podem ser utilizados para desenvolver aplicações que irão atuar como interfaces hiperfídia para bancos de dados legados. Neste caso, possíveis relacionamentos muitos-para-muitos existentes no diagrama E-R devem ser divididos em diversos relacionamentos um-para-muitos equivalentes.

O objetivo do projeto de aplicações hiperfídia é tornar explícitas as ligações entre objetos. A análise do diagrama E-R ajuda a identificar relacionamentos importantes através dos quais a navegação pode ser suportada: se um caminho de navegação deve existir entre duas entidades, um relacionamento correspondente deve existir no projeto E-R.

Passo S2: Projeto de Entidades (ou Projeto de Slices)

Este passo é característico de aplicações hiperfídia e determina como a informação nas entidades escolhidas será acessada pelos usuários, o que envolve dividir uma entidade do modelo E-R em *slices* e organizá-los em uma rede hiperfídia. Assim, uma entidade pode ser apresentada como um conjunto de partes interrelacionadas entre si.

Esta fase de organização de *slices* é chamada *fase de projeto de slices*, gerando como resultado um *diagrama de slices*. A divisão da entidade em slices deve considerar que cada slice representará um “todo” para o usuário. Assim, slices não devem conter muita informação. A saída da fase de *projeto de entidades* é um diagrama E-R estendido, denotado E-R+. Ele é obtido do diagrama E-R pelo preenchimento de cada entidade com seu diagrama de slices.

Ligações estruturais, como foi definido anteriormente, conectam *slices* de uma mesma entidade. Sob o ponto de vista navegacional, a diferença semântica entre ligações estruturais e ligações de aplicação é crucial. O ato de percorrer uma ligação de aplicação (ou associativa) causa uma mudança no contexto da informação, enquanto que uma ligação estrutural mantém o contexto da informação dentro da mesma entidade.

Passo S3: Projeto Navegacional

Neste passo são projetados os caminhos que irão permitir a navegação hiperfídia. Todos os caminhos de navegação são especificados em termos genéricos para suportar a característica de atualização freqüente da base de informação.

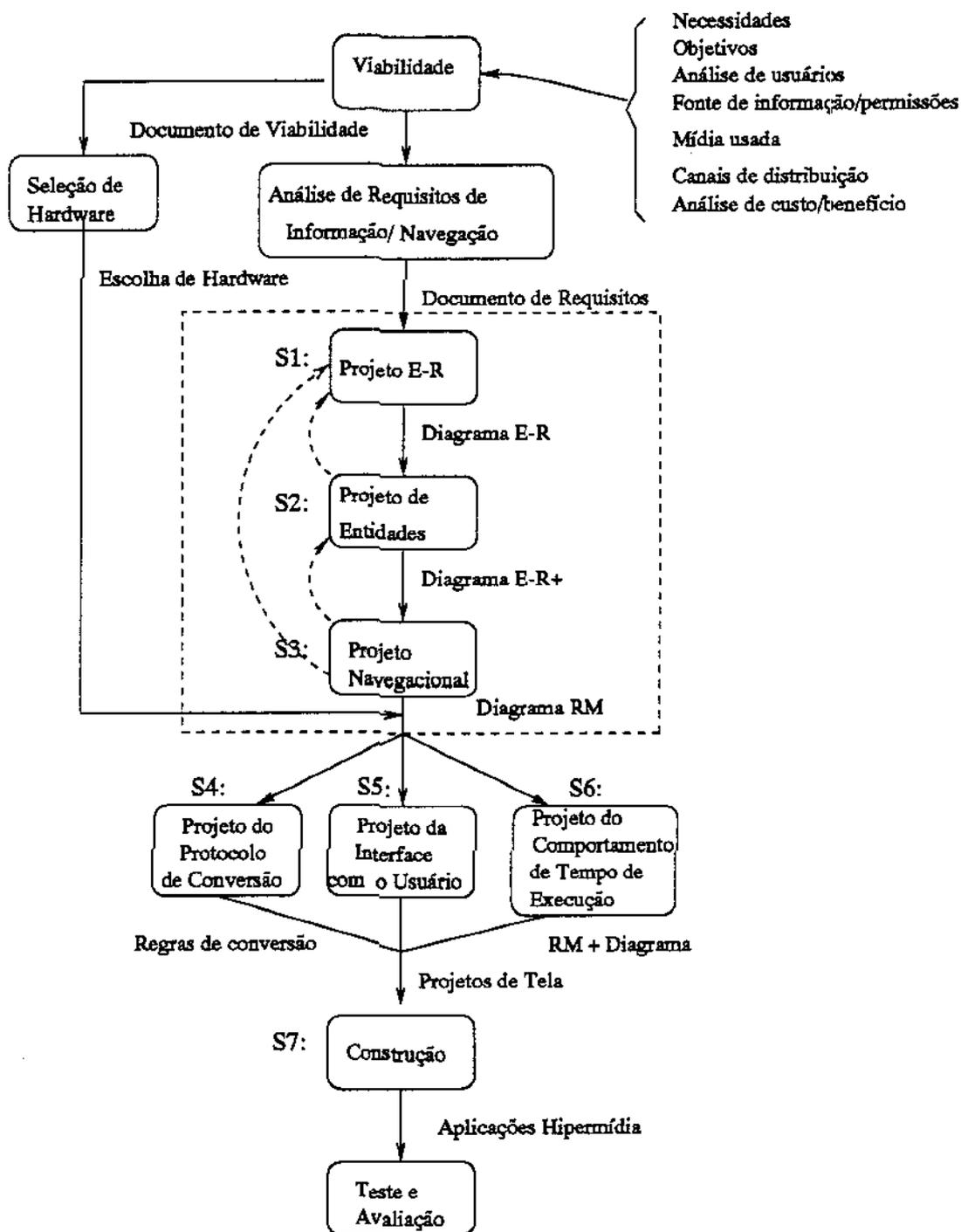


Figura 3.8: Metodologia de Projeto RMM

A primeira tarefa desta fase é projetar a navegação entre entidades baseada nos relacionamentos associativos. Dependendo da semântica que se queira adotar para a navegação, pode-se usar adequadamente quaisquer das estruturas oferecidas pelo modelo (índices, *guided tours* e os *guided tours* indexados).

Em seguida, são projetadas estruturas de acesso de alto nível pelo agrupamento de itens de interesse. Por exemplo, *menus* hierárquicos são uma alternativa para agrupar outras estruturas de acesso aninhadas. Por *default*, o acesso a uma entidade é feito via seu cabeçalho - um *slice* diferenciado que é usado como âncora para ligações chegando à entidade. Contudo, pontos de entrada diferentes podem ser definidos. No fim do estágio de projeto navegacional, o diagrama E-R+ é transformado em um diagrama RMDM que descreve todas as estruturas de acesso a serem implementadas no sistema.

Relacionamentos um-para-um são implementados via ligações bidirecionais. Há várias opções para relacionamentos um-para-muitos. Um *guided tour* é preferido quando o número de participantes é relativamente pequeno (digamos menos que 10) e quando não existem chaves para indexação que possam auxiliar o usuário. Por outro lado, quando existe um grande número de entidades e uma possível chave de indexação, um índice é uma melhor escolha. *Guided tours* indexados são um híbrido geralmente usado quando existe uma chave útil e quando alguma navegação local é desejada.

Projeto da Interface com o Usuário e Construção

Descrevemos a seguir, rapidamente, os quatro passos remanescentes da metodologia RMM. O passo S4, *projeto do protocolo de conversão*, usa um conjunto de regras para transformar cada elemento do diagrama RMDM em um objeto na plataforma destino.

O passo S5, *projeto da interface com o usuário*, envolve o projeto de *lay-outs* de tela para todo o objeto que aparece no diagrama RMDM obtido no passo S3, o que inclui *lay-out* de botões, aparência dos nós e índices, e localização de auxílios navegacionais. Decisões sobre a forma de percorrer a ligação, e como a história, o *backtracking* e os mecanismos de navegação são implementados, são feitas durante o passo S6, o *projeto do comportamento de tempo de execução*. Também durante este estágio, os desenvolvedores consideram a volatilidade e o volume dos dados manipulados para decidir se o conteúdo dos nós e dos extremos de ligações serão construídos durante o desenvolvimento da aplicação ou se serão construídos dinamicamente, sob demanda, em tempo de execução. Finalmente, o passo S7 consiste na *construção e teste*, como em projetos tradicionais de engenharia de software

3.4.4 Object Oriented Hypermedia Design Methodology (OOHDM)

O OOHDM [SR94, SB94, SR95, RSLC95, MSL96, SRB96] é outra abordagem baseada em modelos para construção de aplicações hipermedia. Ele é derivado diretamente de HDM, mas difere deste último por sua natureza orientada a objetos e por incluir primitivas de modelagem especiais para o projeto navegacional e de interface. Ele compreende quatro atividades diferentes: *Projeto Conceitual*, *Projeto Navegacional*, *Projeto da Interface Abstrata* e *Implementação*. As três primeiras fases constroem modelos orientados a objetos para descrever os esquemas correspondentes.

Descrevemos a seguir cada uma dessas atividades com maiores detalhes.

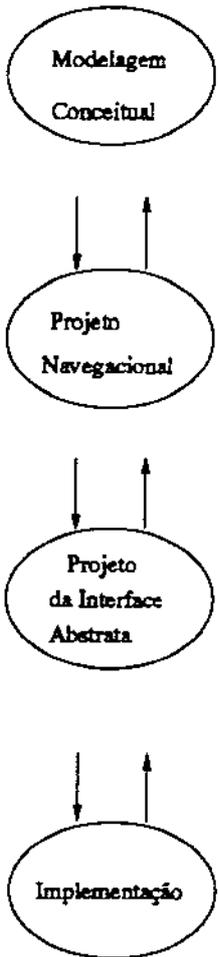
Passos	Produtos	Mecanismos	Interesses do Projeto
	Classes, relacionamentos, perspectivas	Classificação, composição, generalização e especialização	Modelagem da semântica do domínio de aplicação
	Nós, ligações, estruturas de acesso, contextos de navegação e transformações navegacionais	Mapeamento entre os objetos conceituais e de Navegação	Leva em conta o perfil do usuário e as tarefas. Ênfase nos aspectos cognitivos.
	Objetos da Interface Abstrata, respostas a eventos externos, transformações de interface	Mapeamento entre objetos de navegação e objetos perceptíveis	Modelagem dos objetos perceptíveis e implementação das metáforas escolhidas. Descrição da interface para os objetos de Navegação
	Aplicação em execução	Aqueles providos pelo ambiente destino	Completeness, desempenho

Tabela 3.1: Resumo do Método OOHDM

Projeto Conceitual

Nesta fase, um modelo do domÍnio da aplicaçaõ é construído utilizando-se princípios de modelagem orientada a objetos. São utilizados conceitos de classes, relacionamentos, hierarquias de agregação e generalização/especialização. Outras abstrações, como perspectivas (semelhantes à HDM), são também utilizadas.

Esta fase tem por objetivo capturar a semântica do domÍnio de aplicaçaõ, havendo pouca preocupação com os tipos de usuário-alvo e as tarefas a serem desempenhadas pelo sistema. O produto final gerado aqui é o esquema conceitual da aplicaçaõ com suas classes e relacionamentos.

Projeto Navegacional

Uma aplicaçaõ OOHDM é considerada uma visãõ navegacional do modelo conceitual. O projeto do esquema navegacional reorganiza o esquema conceitual tendo em vista os tipos de usuários pretendidos e o conjunto de tarefas que serão executadas ao se utilizar a aplicaçaõ.

A estrutura navegacional da aplicaçaõ hiperfídia é expressa em dois esquemas: o *esquema das classes de navegaçaõ* e o *esquema dos contextos de navegaçaõ*.

Os objetos das classes de navegaçaõ são definidos em termos de um conjunto de tipos predefinidos de classes de navegaçaõ, de maneira similar ao HDM e a outros modelos. Estas classes incluem nós, ligações e estruturas de acesso. Nós e ligações têm a semântica usual de aplicações hiperfídia, podendo ser definidos, respectivamente, como visões orientadas a objetos de classes e de relacionamentos do esquema conceitual. Estruturas de acesso, tais como índices e "guided tours", representam possíveis maneiras de acessar o nó.

Contextos de navegaçaõ desempenham um papel semelhante às coleções em HDM. Eles são a principal primitiva de estruturaçaõ do esquema de navegaçaõ. Eles são definidos como conjuntos de nós organizados segundo algum critério e incluem ligações que são responsáveis pela ordenaçãõ destes nós em um *caminho de navegaçaõ*. A especificaçãõ do caminho de navegaçaõ estabelece uma seqüência através dos elementos do contexto que pode ser seguida pelo usuário da aplicaçaõ ao folhear dentro deste contexto. Estes caminhos podem ser *seqüências*, *ramificações* ou *condições*. O contexto de navegaçaõ possui ainda atributos usados para representar metainformaçãõ, um ou mais pontos de entrada, e uma *descriçaõ comportamental* (que descreve o controle da navegaçaõ dentro do contexto). O controle pode ser *automático* ou *passo-a-passo*. A descriçaõ comportamental descreve também os tipos de restrições que são aplicáveis ao contexto, podendo estas restrições assumir os valores *livre*, *acesso direto*, *exclusivo*, ou *parcial*. Para mais detalhes veja [SB94].

Contextos de navegaçaõ podem ser de seis tipos diferentes: arbitrários, derivados de classes, derivados de ligaçaõ, índices, sessões, e derivados da estrutura. Nos contextos

arbitrários, o autor manualmente seleciona os membros (objetos de navegação) de seu interesse, um a um, segundo algum critério de escolha. Contextos *derivados de classe* são associados e derivados diretamente de cada tipo de nó existente na aplicação. Eles são formados pelo conjunto de todos os nós de um determinado tipo.

Contextos *derivados de ligação* são baseados na existência de relacionamentos um-para-muitos que são gerados na modelagem conceitual. *Índices* são contextos que possuem a semântica de navegação usual desta estrutura de acesso. *Sessões* são construídas dinamicamente à medida que o usuário folheia a informação. Elas são úteis se o usuário quiser ver todos os nós já visitados ou todas as anotações que tenham sido feitas durante a navegação.

Finalmente, os contextos de navegação *derivados de estrutura* são induzidos pela existência de nós compostos. Este tipo de contexto coleta todos os nós pertencentes a um mesmo objeto composto; as ligações, nesse contexto, irão permitir a navegação estrutural dentro da composição.

Os contextos de navegação podem ser aninhados, formando uma organização hierárquica, em que cada nível de classificação corresponde a um contexto de navegação.

Classes em Contexto complementam a definição de uma classe de navegação (um nó) com a adição de atributos dependentes de algum contexto. Estes atributos explicitam qual é a informação sendo mostrada e quais âncoras estão disponíveis quando o objeto é acessado em um contexto particular. Essas informações adicionais são visíveis apenas dentro do contexto especificado. A definição de uma Classe em Contexto inclui uma definição do seu escopo (o conjunto dos contextos de navegação no qual a Classe em Contexto é válida) e de seus participantes (o conjunto dos objetos naqueles contextos que serão modificados).

Uma vez definidos todos os contextos de navegação da aplicação, é possível definir sua estrutura navegacional especificando o *Esquema de Contextos de Navegação*. Este esquema, além de apresentar a definição dos contextos de navegação individuais, mostra também as ligações que conectam os contextos, permitindo um rápido entendimento do espaço navegacional disponível ao usuário.

A especificação da semântica de navegação (folheamento) da aplicação pode ser realizada definindo as transformações navegacionais que ocorrem quando ligações são percorridas. Em OOHDM, isso é especificado através dos *Navigation Charts* derivados dos *Statecharts* que permitem expressar o comportamento navegacional dinâmico.

Projeto da Interface Abstrata

Uma vez definida a estrutura de navegação, é possível especificar quais objetos serão perceptíveis ao usuário, qual a sua forma de apresentação, quais objetos ativarão a navegação, qual é a forma de sincronização dos objetos da interface multimídia e quais transformações ocorrem na interface. Estas definições constituem a *Especificação da In-*

terface Abstrata.

Em OOHDH é usada a abordagem das *Visões de Dados Abstratos (Abstract Data Views - ADV)* para descrever a interface com o usuário. Estas visões são modelos orientados a objetos que capturam os aspectos estruturais e dinâmicos de uma interface com o usuário. Elas são especificadas apresentando:

- a estrutura das ADVs. Para isso são utilizadas abstrações de agregação e generalização/ especialização. ADVs expressam a estrutura do *lay-out* estático da interface ao definir a aparência dos objetos de navegação e outros objetos de interface úteis (tais como *menus* e botões);
- os relacionamentos entre os objetos perceptíveis da interface e os objetos de navegação. Diagramas de configuração são usados para expressar estes relacionamentos;
- o comportamento das ADVS face a eventos externos (como tais eventos disparam a navegação e quais transformações ocorrem na interface quando o usuário interage com a aplicação). Outra generalização dos *Statecharts*, os *ADV Charts*, é usada para especificar este comportamento dinâmico. Para mais detalhes e exemplos veja [RSLC95].

Implementação

A partir de todas as especificações do sistema, o passo final para gerar uma aplicação é mapear os modelos de navegação e da interface abstrata para objetos concretos existentes em um ambiente de implementação. Diversas plataformas hipermídia podem ser utilizadas nesta atividade.

3.5 Resumo e Conclusões do Capítulo

Este capítulo apresentou um estudo de diversas características hipermídia, tanto do ponto de vista de sistemas hipermídia quanto sob a ótica de modelos de autoria para aplicações hipermídia. O estudo esteve centrado na análise e levantamento de requisitos para a definição de um modelo hipermídia voltado a bibliotecas digitais.

Foram considerados os elementos básicos dos sistemas tais como nós, ligações, abstrações de alto nível, operações e aspectos da arquitetura dos sistemas. Questões relacionadas à abertura do sistema e ao suporte a ligações dinâmicas foram identificadas como sendo requisitos essenciais de um modelo hipermídia para bibliotecas digitais.

A seção 3.3 apresentou um *survey* dos sistemas hipermídia propriamente ditos, seguindo uma evolução cronológica dos sistemas em diversos aspectos. Em primeiro lugar,

a análise revelou uma evolução gradativa da camada de armazenamento hiperfídia destes sistemas, de simples sistemas de arquivos (*filesystems*) à utilização de SGBDs relacionais, passando pela incorporação de funcionalidades orientadas a objetos nos sistemas e, finalmente, ao completo uso de SGBDOOs. Em segundo lugar, a utilização de SGBDOOs permitiu a evolução do papel de SGBDs em sistemas hiperfídia, que transcederam de simples sistemas de armazenamento persistentes para completos ambientes de desenvolvimento de aplicações. Em terceiro lugar, como consequência, os modelos de dados destes sistemas hiperfídia evoluíram concomitantemente, incorporando características relativas à modelagem do domínio da aplicação, à modelagem dos aspectos navegacionais e de interface. Estas características deram suporte à definição de estruturas hiperfídia complexas e à manutenção de restrições estruturais dependentes e independentes da aplicação dentro do próprio sistema hiperfídia.

A análise dos modelos abertos de ligações dinâmicas indicou importantes características que devem ser incorporadas aos sistemas. Dentre estas, citam-se a utilização do paradigma de navegação baseado em seleção, e a incorporação de ligações dinâmicas através da utilização de consultas ou de computações, tanto na origem quanto nos destinos da ligação.

A seção 3.4 analisou modelos de autoria para aplicações hiperfídia no âmbito da tecnologia de bibliotecas digitais. As características avaliadas foram a capacidade de modelar os dados de uma biblioteca digital e a maneira de modelar a navegação e os aspectos de interface da biblioteca. As metodologias revelaram diversas similaridades: a definição de modelos de dados para a especificação das hiperbases, semelhantes a modelos de dados para bancos de dados; a incorporação de estruturas de acesso para dar apoio à navegação; a consideração da associação semântica existente entre ligações hiperfídia e relacionamentos conceituais e estruturais do domínio da aplicação. Contudo, diversas diferenças existem, a saber:

- Metodologia Hiperfídia

O HDM se concentra apenas no modelo de dados e não sugere uma metodologia de projeto associada. O EORM utiliza uma metodologia orientada a objetos conhecida e a estende com uma fase de incorporação de semântica de ligações hiperfídia aos relacionamentos. A metodologia do RMM é talvez a mais completa, englobando todas as fases de projeto de uma aplicação hiperfídia. O OOHDm tem a vantagem de separar claramente as etapas de modelagem conceitual, navegacional, de interface e de implementação, obtendo assim uma maior modularidade.

- Modelo de Dados Hiperfídia

O modelo do HDM é o primeiro que se preocupa exclusivamente com o projeto de aplicações hiperfídia. Devido a este pioneirismo, possui abstrações mais pobres

que os outros modelos, mas funciona como uma base para os demais. O RMM é baseado em uma metodologia E-R. Já o EORM e o OOHDM utilizam abstrações orientadas a objetos que são extremamente úteis no projeto de aplicações hipermídia como, por exemplo, o uso de métodos em classes para prover comportamento aos objetos e hierarquias de especialização e agregação para modelar objetos hipermídia complexos.

- Projeto de Interface

O OOHDM é o único que separa claramente o projeto da interface das demais fases de projeto e é também o único que oferece ferramentas formais para a descrição do comportamento da interface de modo independente da implementação. Os demais relegam o projeto de interface para a etapa de implementação.

- Organização da aplicação

A organização da aplicação em HDM e em RMM é efetuada através da utilização de estruturas de acesso tais como índices e *guided tours*. O RMM possibilita ainda a utilização de condições nas estruturas de acesso, oferecendo algum grau de dinamismo nas aplicações. A especificação de determinadas classes de ligações em EORM permite organizar a navegação hipermídia. O OOHDM introduz os contextos de navegação que são primitivas arquiteturais de alto nível que incorporam consultas e que oferecem talvez a melhor forma de organização da aplicação.

Capítulo 4

Modelo de Banco de Dados Hipermedia para Biblioteca Digital

Este capítulo apresenta um dos resultados da dissertação - o modelo do Sistema Gerenciador de Banco de Dados Hipermedia que estende o modelo do SGBD OOHD de Milet et al. [MSL96]. As extensões propostas visam satisfazer os requisitos de um modelo hipermedia para bibliotecas digitais, apresentados no capítulo anterior, adequando o modelo a um ambiente de biblioteca digital. Além disto, o modelo diminui a distância semântica entre as abordagens de modelos de dados para sistemas hipermedia e de modelos para desenvolvimento de aplicações hipermedia. Este objetivo é alcançado através da integração de conceitos do modelo de autoria OOHD com conceitos do modelo de Dexter Estendido voltado ao projeto de sistemas hipermedia [GT96]. A integração é consolidada em um único modelo a ser implementado sobre um SGBDOO, que é a plataforma proposta para o desenvolvimento de aplicações de bibliotecas digitais. O capítulo está organizado da seguinte forma. A seção 4.1 justifica as escolhas de metodologia e de plataforma hipermedia adotadas. A seção 4.2 detalha o modelo em termos de suas classes e das especificações destas classes. A seção 4.3 discute a navegação utilizando o modelo. A seção 4.4 avalia o modelo apresentado. A seção 4.5 resume o capítulo.

4.1 Escolha da Metodologia e da Plataforma Base

4.1.1 Metodologia de Projeto de Aplicação Hipermedia

A metodologia hipermedia a ser adotada nesta dissertação é a do OOHD. Diversos aspectos influenciaram esta escolha. O primeiro aspecto considerado relevante foi o fato de que a metodologia utiliza um modelo de dados orientado a objetos que possui abstrações adequadas à modelagem dos dados de bibliotecas digitais.

O segundo aspecto considerado é que a metodologia separa claramente as etapas de modelagem conceitual, de navegação e de interface. A separação entre o modelo conceitual e o modelo navegacional em OOHDM permite a construção de mais de uma aplicação de biblioteca para o mesmo domínio. A separação entre o projeto de navegação e o projeto da interface permite construir mais de uma interface para o mesmo modelo navegacional, facilitando sua customização.

Outra característica relevante é a abrangência do modelo do OOHDM. Este modelo é uma versão aprimorada do modelo do HDM, possuindo primitivas especiais para os projetos navegacional e de interface. Ele incorpora e estende as principais idéias referentes à organização da aplicação como, por exemplo, as coleções propostas por Garzotto et al. [GMP94a] e os contextos aninhados de Soares et al. [SRC95].

4.1.2 Plataforma Hipermissão

Os benefícios da utilização de um SGBDOO como plataforma de implementação de um modelo hipermissão para o desenvolvimento de aplicações são inúmeros [BWAH96, MSL96]. A primeira e mais óbvia destas vantagens é dispor de uma série de funcionalidades comuns de bancos de dados úteis para aplicações hipermissão, e em particular para uma biblioteca digital, tais como controle de concorrência, recuperação de falhas, tipos de dados para multimídia e acesso declarativo aos objetos hipermissão (através de mecanismos de consultas do SGBD subjacente). A construção das aplicações usando o SGBDOO facilita a manutenção de restrições de integridade entre os objetos hipermissão, inclusive nos acessos multiusuário. O segundo conjunto de vantagens advém das propriedades do modelo orientado a objetos, já comentadas anteriormente.

Desta forma, consideramos que o SGBD Hipermissão será um SGBDOO. Esta implementação far-se-á através da definição de metaclasses e de construtores de alto nível, que modelam a semântica dos conceitos hipermissão presentes no modelo da metodologia escolhida. Cada aplicação será, assim, desenvolvida a partir da especialização destas metaclasses. A viabilidade desta abordagem já foi demonstrada em [MSL96].

O uso de construtores de modelagem de alto nível no modelo facilita a tarefa de desenvolvimento de aplicações. Esta característica libera os desenvolvedores da necessidade de definir uma semântica hipermissão complexa para a aplicação, já que eles podem adaptar as metaclasses para as suas necessidades via mecanismos de especialização.

Outra característica relevante é que as camadas relativas ao desenvolvimento da aplicação hipermissão podem ser integradas no SGBD, reduzindo o número de interfaces entre componentes e a sobrecarga envolvida no mapeamento de conceitos entre as camadas.

4.2 Modelo do Sistema Gerenciador do Banco de Dados Hipermedia

Esta seção descreve o modelo de dados do SGBDH. Detalhamos, primeiramente, a estrutura de âncoras e de ligações. A seguir, serão explicadas as estruturas correspondentes às classes de navegação e de contexto.

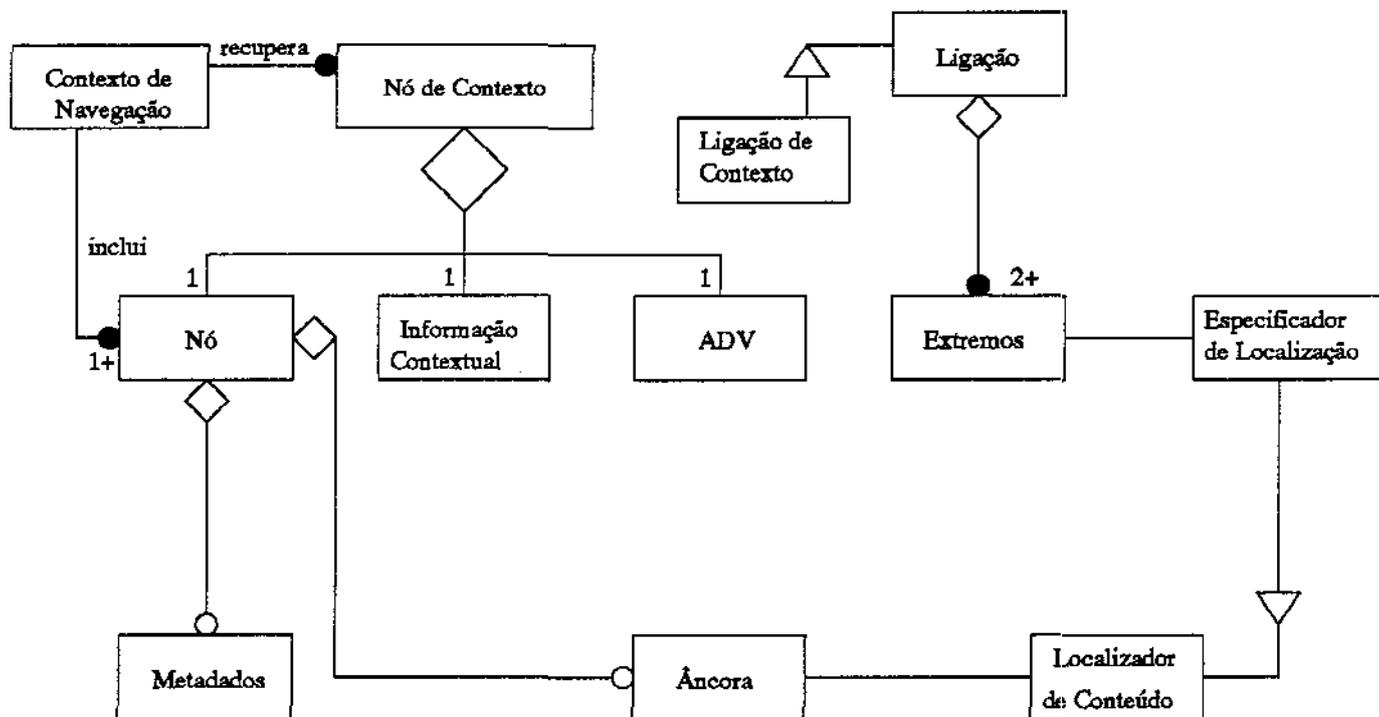


Figura 4.1: Modelo do Banco de Dados Hipermedia

O modelo é mostrado na figura 4.1, em alto nível, usando a notação da metodologia OMT. O modelo é baseado no modelo do SGBD do OOHDH [MSL96]. A figura retrata as extensões aqui propostas a este último modelo, algumas das quais são inspiradas no trabalho de Gronbaek e Trigg [GT96].

As principais extensões são: a) introdução dos conceitos de localizador de conteúdo e especificador de localização; b) separação entre os conceitos de âncoras e extremos de ligações e definição da associação indireta entre estes conceitos; c) introdução de ligações para a navegação baseada em seleção, e definição de ligações de contexto como especializações da classe Ligação; d) uso do conceito de visões em banco de dados para definir contextos de navegação sobre o esquema de classes de navegação; e e) inclusão de uma lista de metadados para cada nó. A seguir, detalhamos estas extensões.

4.2.1 Localizador de Conteúdo e Especificador de Localização

Um *Localizador de Conteúdo* (*LocCont*) (chamado Especificador de Localização em [GT96]) é uma classe básica do modelo hipermedia proposto, sendo responsável por especificar localizações dentro de um documento (nó).

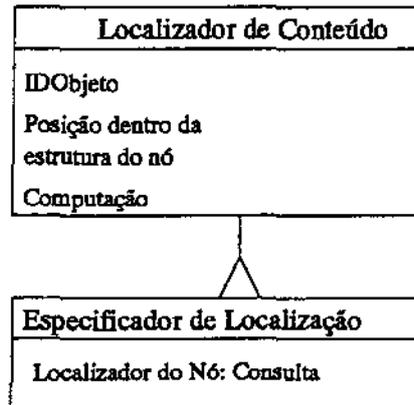


Figura 4.2: Classes Localizador de Conteúdo e Especificador de Localização

Os atributos de um Localizador de Conteúdo permitem obter uma localização dentro de um nó de formas distintas: por identificação absoluta (ou nomeada) (*atributo IDObjeto*) (e.g., um identificador de âncora, um nome HTML definido); por especificação de uma estrutura dentro do nó (*atributo Posição dentro da estrutura do nó*) (e.g., *offset*, *frame* de vídeo, um capítulo em um documento com marcações SGML); ou pela especificação de um critério de busca ou *script* a ser aplicado sobre o conteúdo dos nós (*atributo Computação*) (e.g., uma consulta, métodos multimídia, uma busca por uma string de texto, um *script* executável). Estas especificações atuam de forma conjuntiva, isto é, se mais de uma delas é definida, todas as especificações devem ser verdadeiras ao mesmo tempo. O LocCont permite manter uma separação lógica entre o nó e o seu conteúdo (mídia encapsulada pelo nó).

A classe *Especificador de Localização* é uma especialização de LocCont que identifica localizações associadas a um nó ou a um contexto de navegação no qual estas se encontram. O EspLoc é descrito por uma consulta que identifica nós - através de nomeação explícita, através de computações (métodos) aplicados sobre o nó, ou através de seus atributos - ou que indica um contexto ao qual está associada a localização. A consulta é denominada *atributo localizador do nó*.

A classe LocCont substitui o valor da âncora de Dexter, enquanto o Especificador de Localização assume um papel semelhante aos conceitos de ID da âncora e de especificação de componentes dentro dos especificadores de Dexter.

Os conceitos de LocCont e de EspLoc podem ser usados, por exemplo, para modelar

o mecanismo de endereamento URL da WWW. Este mecanismo localiza páginas web fixas codificando um protocolo servidor, um nome de um *host*, e um nome de um caminho (identificador do Nó). URLs podem localizar pontos de entrada particulares em páginas WWW, se estes foram predefinidos na página HTML (IDObjeto) e podem também invocar um programa em um servidor passando argumentos (Computação).

4.2.2 Âncoras e Extremos

LocConts e EspLocs podem existir independentemente do sistema de banco de dados hiperfídia corrente. Âncoras e Extremos modelam referências que aparecem dentro do BD no qual os nós estão armazenados. Âncoras estão associadas a nós ao passo que extremos relacionam-se a ligações.

Âncoras residem em nós e identificam localizações dentro destes. Uma âncora inclui um nome, um Localizador de Conteúdo e um identificador do nó que o contém (*atributo IDPai*). Ela inclui ainda um método *IgualLocCont* que aceita um EspLoc como argumento e determina se ele possui uma estrutura idêntica (i.e., se define uma mesma localização) à do LocCont pertencente à âncora. Esta operação é usada durante o processo de navegação.

Âncora
Nome:string
IDPai:oid
Localizador de Conteúdo:LocCont
IgualLocCont

Figura 4.3: Classe Âncora

Âncoras servem como objetos de referência para ligações. O LocCont dentro das âncoras permite especificar localizações que não estão vinculadas diretamente ao conteúdo / mídia do nó. A capacidade de localização da âncora é limitada apenas pelos métodos providos pelos tipos de dados multimídia para acessar partes dos dados. Âncoras podem estar localizadas, por exemplo, em segmentos de som e vídeo, em regiões de imagens, ou em tabelas de bancos de dados relacionais.

Âncoras Transientes são construídas em tempo de execução, geralmente baseadas em uma seleção no conteúdo de algum nó. Elas não têm associação com âncoras existentes e tipicamente existem apenas durante uma operação de seguir a ligação.

A semântica da âncora do nosso modelo é bem distinta daquela associada às âncoras do SGBD do OOHDM. Neste último modelo, toda âncora deve ser definida a priori dentro do nó e, para ser acessível, deve ser representada como um objeto visível na interface do

nó. A noção de âncoras transientes não existe no modelo do OOHDM. Âncoras transientes são a abstração que permite a navegação baseada em seleção de dados dos nós. A associação indireta existente entre âncoras e ligações em nosso modelo permite definir âncoras universais que são compartilhadas por todas as ligações.

Extremos contêm o identificador da ligação que os encapsulam (*atributo IDPai*), um *EspLoc*, e um atributo “*tipo*”, cujos valores incluem: ORIGEM, DESTINO ou AMBOS. Da mesma forma que em Dexter, estes valores determinam como o extremo irá se comportar quando ligações são seguidas.

Extremo
tipo: (ORIGEM, DESTINO ou AMBOS)
IDPai:oid
Especificador de Localização:EspLoc
IgualEspLoc
SetType
GetType

Figura 4.4: Classe Extremo

Extremos contêm também um método *IgualEspLoc* - semelhante ao da Âncora, no sentido de que este método possui a mesma funcionalidade do respectivo método da Âncora, efetuando o processo de casamento com o *EspLoc* do Extremo ao invés de com um *LocCont* - além de métodos para atribuir e capturar os valores de *tipo* (respectivamente, método *SetType* e *GetType*).

Âncoras e Extremos podem existir em apenas um único nó ou em uma única ligação, respectivamente, não sendo assim compartilháveis. Tal restrição não existe para *LocConts* e *EspLocs*, pois múltiplas Âncoras / Ligações podem compartilhar os mesmos *LocConts* / *EspLocs*.

Estas classes são as abstrações que permitem modelar uma ligação como objeto separado, de forma desvinculada do conteúdo dos nós do BD. As vantagens deste tipo de abordagem foram discutidas no capítulo anterior.

Um dos aspectos mais importantes destas classes é o seu poder de suportar ligações dinâmicas, cujo destino ou origem não são explicitamente representados, mas podem ser computados sob demanda, como é o caso de páginas WWW em *browsers*. As noções de *LocConts* e *EspLocs*, dentro de âncoras e extremos respectivamente, podem representar explicitamente buscas ou computações sobre o SGBDH que são a base da implementação das ligações dinâmicas.

Os conceitos de Microcosm, por exemplo, podem ser mapeados para as estruturas das classes do modelo aqui proposto: ligações *específicas* podem ser modeladas como um

extremo origem contendo um Localizador do Nó no seu EspLoc e uma posição definida em seu atributo “Posição dentro do nó”; ligações *locais* são modeláveis como um extremo contendo um atributo Localizador de Nó no seu EspLoc e um atributo “Computação” como uma consulta ao SGBDH; e ligações *genéricas*, modeláveis como um extremo contendo um atributo Localizador do Nó vazio em seu EspLoc e uma consulta dentro do conteúdo dos nós.

4.2.3 Nós e Ligações

Nós - denominados classes de navegação no modelo do OOHDH - e Ligações são unidades semânticas para a navegação. Um nó é constituído por um nome, uma lista de âncoras e uma lista de objetos da classe Metadados. O nome identifica o nó. A lista de metadados descreve propriedades do nó. As âncoras definem localizações dentro do nó. O conteúdo do nó - atributos e dados multimídia - é inserido nos nós específicos da aplicação, modelados como subclasses de Nó.

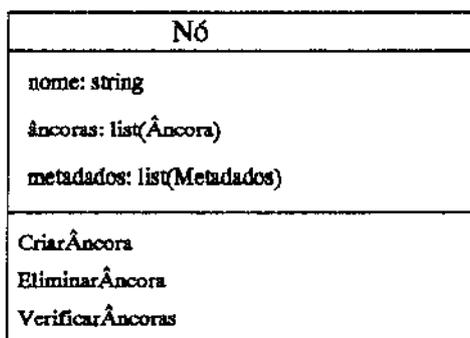


Figura 4.5: Classe Nó

Ligações contêm um nome, uma lista de atributos descritivos, e uma lista de extremos. Na biblioteca digital, não são efetuadas consultas sobre estas ligações. Esta decisão de projeto pode ser revista em sistemas nos quais não há uma distinção entre usuário e projetista. Nestes sistemas, o usuário tem a possibilidade de poder criar suas próprias ligações no SGBDH e de efetuar consultas sobre estas ligações.

4.2.4 Classe Metadados

A classe Metadados é raiz de uma hierarquia de classes de metadados. Esta hierarquia é necessária para facilitar a navegação do usuário e a descoberta de dados. Cada nó possui uma lista de metadados que descrevem diversas características deste nó. As subclasses de Metadados permitem descrever tanto o conteúdo do objeto quanto os aspectos extrínsecos

ao n3. Exemplos destes aspectos s3o informa33es associadas a termos, condi33es e custos de acesso ao n3; informa33es administrativas relativas ao gerenciamento de um n3 particular; informa33es relativas 3a proveni33ncia, adequa33o e uso correto dos dados. A manuten33o em separado de Metadados possibilita a utiliza33o de um ou mesmo de v3rios padr3es (e.g., FGDC e USMARC) para a descri33o dos objetos, facilitando o aspecto de interoperabilidade. A classe Metadados 3e utilizada na constru33o do cat3logo de biblioteca e ser3 discutida em maiores detalhes no cap3tulo 6.

4.2.5 Contexto de Navega33o

Um contexto de navega33o 3e um mecanismo de estrutura33o da aplica33o hipermdia. Em [Ros96] e [SRG97], por exemplo, Rossi e Schwabe definem os contextos de navega33o em termos de padr3es de projeto. A id3ia explorada atrav3s dos contextos de navega33o 3e a de facilitar o entendimento do ambiente de navega33o pelo usu3rio [SB94]. Contextos de navega33o podem minimizar o problema de desorienta33o do usu3rio e podem reduzir a carga cognitiva imposta pela navega33o atrav3s de hiperbases complexas. Outra importante caracter3stica dos contextos, que 3e tamb3m compartilhada pelas cole33es de Garzotto, 3e a sua habilidade de permitir a modifica33o da organiza33o e da apresenta33o de uma aplica33o, sem haver necessidade de altera33o das informa33es contidas nos n3s.

Contextos organizam a informa33o gerando particionamentos l3gicos do conjunto de n3s da biblioteca. Esta defini33o de contexto pode ser comparada 3a de **vis3o em bancos de dados**.

Uma vis3o em BDs orientados a objetos 3e comumente considerada um *banco de dados virtual*, ou seja, tem *classes virtuais* e *objetos n3o persistentes*. Classes virtuais s3o definidas por um esquema (atributos e m3todos). Atributos e m3todos da vis3o podem ser id3nticos aos do BD ou podem ser computados (virtuais). A extens3o da vis3o corresponde ao seu conte3do e 3e especificada por uma consulta sobre o SGBD. A consulta pode envolver classes do banco de dados ou de outras vis3es. Apenas a defini33o da vis3o

Liga33o
nome: string
atributos: list(atributo)
extremos: list(extremos)
CriarExtremo
EliminarExtremo
VerificarExtremos

Figura 4.6: Classe Liga33o

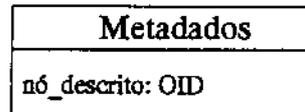


Figura 4.7: Classe Metadados

é armazenada, sendo que as suas instâncias são obtidas apenas após a avaliação da consulta. A avaliação da consulta e a determinação dos dados da visão correspondem ao que [Cer96] chama de *ativação*.

Aqui, propomos especificar contextos de navegação utilizando visões. A definição dos contextos de navegação como visões sobre o SGBDH permite à biblioteca digital explorar o mecanismo de visões do SGBDOO subjacente. Dentre as funcionalidades de um mecanismo de visões úteis a uma biblioteca digital, podemos citar:

1. Reestruturação

Visões provêm mecanismos para reestruturar a hierarquia de classes e para modificar o comportamento e a estrutura dos objetos, permitindo especificar valores de atributos implicitamente, ao invés de armazená-los. Visões também possibilitam a criação de novas classes, provendo ao usuário uma hierarquia de classes que lhe é mais apropriada do que a hierarquia realmente armazenada no banco de dados. Esta é uma das principais características de visões exploradas pelos contextos.

2. Limitação e Particionamento dos dados

Visões possibilitam o particionamento das extensões das classes do banco de dados, permitindo a seleção das entidades que satisfazem determinadas condições. Este trabalho de particionamento é delegado ao processador de consultas. Esta característica é importante no particionamento dos objetos digitais da biblioteca pelos contextos de navegação e na implementação das condições de seleção de nós declaradas na definição dos contextos de navegação.

3. Utilização de objetos virtuais e Gerenciamento das extensões da visão

Objetos virtuais são aqueles criados durante a ativação da visão, gerenciados pelo mecanismo de visões durante o ciclo de vida da visão, e destruídos em sua desativação. Esta característica libera o contexto de navegação de ter que gerenciar listas persistentes de nós de contextos, as quais duplicam a informação, como ocorria no contextos de navegação de Milet et al. [MSL96].

4. Segurança e Integridade do Banco de Dados Hipermissão

Visões estão também ligadas à limitaçoão e à proteçoão de informaçoẽs, impedindo o acesso de usuários que não têm autorizaçoão para acessar certos dados e limitando informaçoẽs acessadas. A garantia de proteçoão contribui para a manutençoão da integridade dos dados. Esta característica torna-se essencial quando considerando aspectos de bibliotecas digitais tais como controle de acesso à propriedade intelectual, restriçoão de visualizaçoão de dados “*copyright*” e acesso restrito e condicionado à pagamento.

5. Suporte à atualizaçoão e manutençoão das visões

Atualizaçoẽs nas classes armazenadas do banco de dados devem ser refletidas nas visões (*manutençoão de visões*) [Cer96]. Analogamente, contextos devem refletir atualizaçoẽs dos nós. Estas atualizaçoẽs podem ser propagadas para os contextos periodicamente ou sob solicitaçoão da aplicaçoão (*request for refresh*). Esta tarefa deve ser delegada a um *Gerenciador de Atualizaçoẽs*, que deve fazer parte do mecanismo de visões do SGBDH, como definido em [Cer96].

Assim sendo, adaptamos a noçoão de visão para uso em navegaçoão hipermidia, definindo contextos de navegaçoão como visões especiais em BDOO, tendo as seguintes características:

- Contextos não permitem atualizaçoẽs sobre nós

Atualizaçoẽs sobre nós a partir dos contextos de navegaçoão não são permitidas. Esta característica libera o banco de dados ou as aplicaçoẽs da biblioteca de terem que implementar complicados métodos de *atualizaçoẽs de visões* para o banco de dados subjacente.

- Contextos facilitam a reestruturaçoão da hierarquia de classes e a definiçoão de critérios de seleçoão

Contextos são fundamentais na reestruturaçoão da hierarquia de Classes de Navegaçoão, e são baseados em critérios de seleçoão para fins de particionamento das extensões destas classes. A reestruturaçoão da hierarquia é efetuada através da criaçoão de novas classes virtuais.

- Contextos têm parte de sua fundamentaçoão na informaçoão dos metadados

Metadados fornecem algumas das informaçoẽs básicas para o particionamento dos objetos pelos contextos, como por exemplo, os retângulos envolventes dos objetos geográficos, a validade temporal e relacionamentos com outros objetos.

- Contextos têm populaçoão virtual

O uso de visões para descrever contextos facilita a especificação do Projeto do Esquema do Contextos de Navegação da metodologia do OOHDM (adotada nesta dissertação) e o seu posterior mapeamento para o esquema do SGBDH. Falta, no entanto, a dimensão dinâmica que permite gerar novos contextos a partir das necessidades dos usuários e da evolução da biblioteca digital. Para isto, utilizamos a noção de visão parametrizada de [AB91]. Uma visão parametrizada possui parâmetros que são avaliados no processamento da consulta que cria a extensão da visão, de forma a determinar as instâncias da visão. Apenas os objetos que satisfazem o predicado da consulta em relação ao parâmetro fornecido farão parte dos dados da visão. Quando objetos são removidos ou adicionados à biblioteca, o mecanismo de parametrização, conjuntamente com o mecanismo do Gerenciador de Atualizações, permite que contextos sejam automaticamente removidos ou criados.

Nem sempre visões devem ser virtuais. Há casos em que elas podem ser *materializadas*, ou seja, tornadas persistentes. Nestes casos, contextos passariam a ser permanentes - por exemplo, se determinado usuário tem padrões fixos de acesso ou se não são esperadas grandes modificações no conteúdo da biblioteca. A materialização de visões traz vários problemas, em especial de manutenção de consistência.

O uso das visões parametrizadas aumenta consideravelmente a flexibilidade na definição de contextos de navegação. A abordagem permite a construção de contextos a partir de construção de visões em um processo *top-down* baseado em especializações, no qual as subclasses são construídas particionando sua superclasse de acordo com algum parâmetro. Por exemplo, um grupo de contextos de navegação *ImagensPorTema* pode ser definido através da seguinte declaração de visão parametrizada:

```
view ImagensPorTema(T) includes
(select i from Imagens
where i.tema=T)
```

Assim, *ImagensPorTema(Vegetação)* e *ImagensPorTema(Hidrografia)* representam dois contextos de navegação que particionam a classe *Imagens* do Banco de Dados através de um atributo *tema*. Esta forma de declaração é bem mais conveniente que declarar um contexto de navegação separado para cada possível tema. A discussão dos problemas de geração de visões parametrizadas se encontra em [AB91].

A figura 4.8 mostra a classe Contexto de Navegação definida como uma visão. Os contextos de navegação da aplicação são definidos como subclasses dessa visão. A classe Contexto de Navegação herda suas funcionalidades da classe de visão *ObjetoVirtual* [dS95], sendo esta classe definida pelo mecanismo de visões do SGBDOO. A Classe *ObjetoVirtual* é a responsável pela implementação das funcionalidades de uma classe virtual. Dentre as

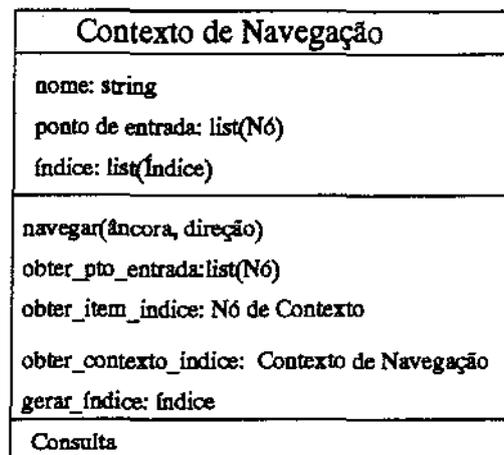


Figura 4.8: A definição da classe de visão Contexto de Navegação

funcionalidades providas por esta classe cita-se: o mapeamento entre o objeto virtual e o objeto real armazenado no BD, a partir do qual o objeto virtual é gerado; provisão de comandos para avaliação da consulta da visão; definição de métodos para a verificação de inconsistências; dentre outras.

A classe Contexto de Navegação é composta pelos atributos de classe *nome*, *ponto de entrada* e *índice*. O ponto de entrada é um atributo obrigatório que indica quais nós de contexto possibilitam o acesso ao contexto. O ponto de entrada pode ser um único nó ou mais de um nó, inclusive todos daquele contexto. O atributo índice é uma estrutura hiperfídia de acesso aos nós, semelhante a um menu que possui itens, possibilitando o acesso a um nó específico ou a um outro contexto aninhado no caso de menus hierárquicos. Quando um item específico é selecionado, o nó ou o contexto associado é recuperado. Um índice pode ser definido *a priori* ou pode ser gerado após a avaliação da consulta que determina os dados pertencentes ao contexto através do método *gerar_indice*.

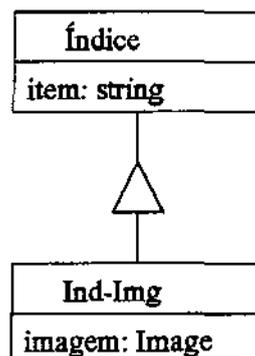


Figura 4.9: Definição de uma classe Índice e de um exemplo de subclasse definindo índices com imagens

Como no modelo do SGBD OOHD, uma classe Índice (figura 4.9) deve ser definida. A estrutura da classe Índice contém um atributo item que está relacionado ao objeto ao qual ele dá acesso. O item pode ser, por exemplo, o nome de um objeto ou de algum contexto. Subclasses da classe Índice podem definir índices que utilizam imagens, informações parciais de metadados, ícones, etc. A figura 4.9 apresenta um exemplo no qual a classe Ind-Img é uma subclasse de Índice que apresenta imagens como entradas do índice. Um índice pode também ser criado dinamicamente por aplicações para apresentar resultados de consultas.

4.2.6 Informação Contextual

A classe Informação Contextual modela informações que são usadas nos nós de um contexto para facilitar a navegação inter e intracontexto e informações que complementam a informação básica do nó quando este aparece em um contexto específico. A classe Informação Contextual corresponde à Classe em Contexto do modelo do OOHD. A classe Informação Contextual possui um método obter_objeto_contexto que é utilizado para instanciar um objeto desta classe.

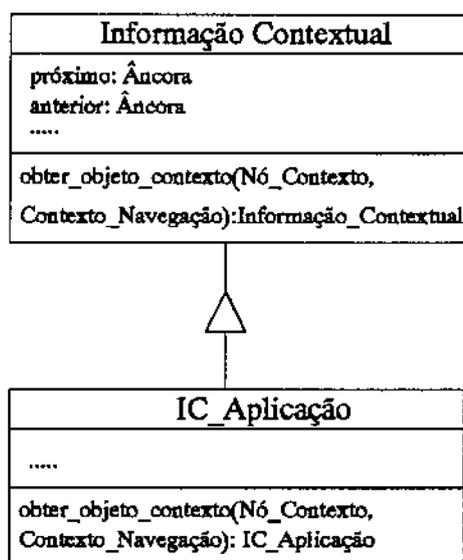


Figura 4.10: Classe Informação Contextual

Esta classe é definida como uma classe abstrata que deve ser especializada pelas classes de Informação Contextual da aplicação. Por exemplo, um objeto de Informação Contextual da aplicação pode modelar âncoras *próximo* e *anterior* para permitir a navegação dentro de um contexto de navegação, pode armazenar informações sobre a posição do nó dentro de um contexto e o total de nós deste contexto, ou pode armazenar informação

relativa ao relacionamento envolvido na navegação entre contextos.

4.2.7 Nó de Contexto

Um nó no BD (classe Nó) pode aparecer dentro de diferentes contextos. Em cada contexto, o nó é associado a diferentes informações e a apresentações de controle distintas. A tripla <Nó, Especificador de Apresentação (adv), Instância de Informação Contextual (ic)> recebe o nome de Nó de Contexto. Em particular, um mesmo Nó pode pertencer a duas ou mais visões definidas a partir de critérios diferentes, abrindo assim a possibilidade de sobreposição de visões (interseção das extensões) e conseqüente reuso [GMP95] de nós entre contextos. Em contrapartida, cada Nó de Contexto é específico para algum contexto particular. A apresentação de um mesmo nó, sob diferentes perspectivas e pontos de vista, aumenta a flexibilidade na especificação de uma aplicação ao permitir que um nó possa ser encontrado através de diferentes caminhos de acesso, em diferentes contextos e sob diferentes pontos de vista.

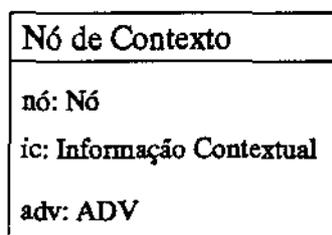


Figura 4.11: Classe Nó de Contexto

Como contextos são visões, as instâncias da classe Nó de Contexto não são persistentes (exceto nos casos já mencionados anteriormente, de usuários especiais ou de baixa volatilidade da biblioteca). Em geral, visões são geradas apenas para efeito de apresentação do nó no contexto. A persistência do nó de contexto requereria a materialização dos objetos virtuais da visão. A possibilidade de utilizar dados persistentes na extensão de uma visão traz problemas no compartilhamento de objetos, no controle de concorrência e na reestruturação dos dados [Cer96].

Em segundo lugar, a volatilidade inerente a uma biblioteca digital pode causar uma invalidação das informações referentes aos objetos da classe Informação Contextual (ic), tais como posição do nó dentro do contexto, ou total de nós deste contexto. Esta característica desencoraja a definição do objeto de contexto (ic) como persistente, condição necessária à persistência do nó de contexto.

Ressalte-se que a virtualidade dos contextos não significa que a informação não esteja armazenada. Virtualidade significa que os dados armazenados podem ser (a) organizados

e apresentados de formas diferentes e (b) combinados gerando dados derivados, estes, sim, não armazenados.

4.2.8 Ligações de Contexto

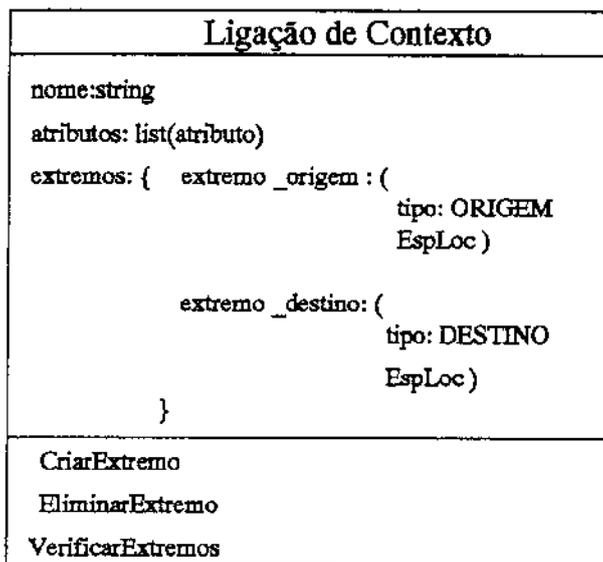


Figura 4.12: Classe Ligação de Contexto

A classe *Ligação de Contexto* é uma especialização da classe *Ligação* e representa as ligações da aplicação que ocorrem dentro e entre contextos de navegação. Uma ligação de contexto é composta por um extremo do tipo origem e um outro do tipo destino. O *EspLoc* destes extremos dita buscas sobre âncoras específicas para origem da ligação ou define nós ou contextos que são destino da ligação.

Existem dois tipos de navegação relacionadas aos contextos: (a) navegação dentro do contexto; e (b) navegação entre contextos. As Ligações de Contexto para navegação dentro do contexto são modeladas com um *extremo_origem* cujo *EspLoc* especifica âncoras próximo e anterior, atributos da Classe Informação Contextual. O *EspLoc* do *extremo_destino* define um Localizador do Nó, que utiliza um método. Este método recebe como parâmetro a direção de navegação e nó em que ocorreu a ativação da âncora e retorna como resposta o nó destino. Este método é o responsável pela implementação do caminho de navegação especificado para o contexto.

Para a navegação intercontextos, o *extremo_origem* de uma ligação de contexto tem forma semelhante. O *EspLoc* do *extremo_destino* armazena uma chamada a uma visão parametrizada referente ao (Nó de) Contexto destino da ligação. Caso o contexto de navegação destino da ligação não exista, ele deve ser ativado durante este processo de

navegação.

Por exemplo, considere a conexão de um contexto de Imagens por Tema para um contexto relativo às versões de uma imagem com uma ligação dinâmica genérica, supondo que esta ligação seja modelada com o extremo_ origem definindo uma âncora *PorVersão*, presente em todos os nós de imagem da biblioteca, e com o extremo_ destino ativando uma visão parametrizada *ImagemPorVersão(nó)*. No momento da navegação, a Imagem na qual foi selecionada a âncora é passada como o parâmetro *nó* da visão, cuja extensão é criada com todas as versões desta imagem.

Este estilo de especificações de ligações contempla o aspecto dinâmico que se deseja alcançar em nossa biblioteca. Para a navegação dentro do contexto, duas únicas ligações de contexto são necessárias para cada contexto. Estas duas ligações estão indiretamente conectadas aos atributos do tipo âncora *próximo* e *anterior* da respectiva classe Informação Contextual.

Para a navegação intercontextos, a especificação das ligações de contextos definem “tipos” de ligações. Um tipo de ligação expressa um relacionamento semântico entre nós e / ou contextos de navegação, sendo modelado por ligações dinâmicas genéricas em nosso modelo. Os tipos são definidos em termos das âncoras inseridas nos nós para extremo_ origem e das visões parametrizadas referentes aos contextos destinos da ligação. Como estas âncoras são definidas para todos os nós, elas permitem a generalização e o compartilhamento do tipo de ligação entre todos os nós da biblioteca. Desta forma, qualquer nó inserido no sistema *a posteriori* já teria as ligações de contexto definidas para as suas respectivas âncoras. Este estilo de definição de ligações através de ligações de contexto dinâmicas genéricas dispensa a necessidade da função “obter âncoras”, presente no metamodelo de Milet.

4.2.9 ADV

A ADV é uma classe genérica responsável pela apresentação de um nó dentro de um contexto de navegação. Ela é definida como uma classe abstrata para ser especializada pelas classes específicas da aplicação a serem apresentadas.

Uma ADV da aplicação (exemplificada na figura 4.13 pela subclasse *ADV_Aplicação*) é uma especialização da classe ADV. O método *apresentar* da classe ADV gerencia a apresentação dos nós. O método analisa se a ação do usuário sobre a interface se refere a trocas de mensagens entre a ADV principal e outras ADVs aninhadas que formam uma composição, ou se está associada a alguma âncora de navegação. A composição possui objetos de interface, tais como menus e botões, e pode possuir outras ADVs aninhadas que estão associadas aos atributos do nó. Os objetos de interface podem ser instanciados a partir das classes disponíveis em uma biblioteca de interface de um SGBDOO como, por exemplo, classes *Label*, *Image*, *Text* e *Button*. Âncoras de navegação indicam que

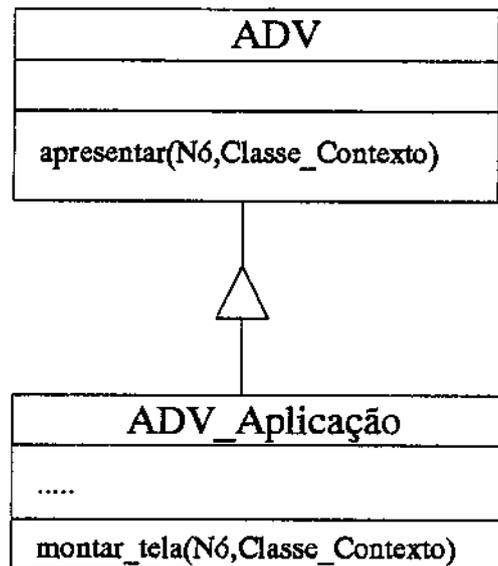


Figura 4.13: Classe ADV

a mensagem “navegar” deve ser passada para o contexto de navegação, a fim de obter o próximo nó. Âncoras transientes são tratadas como âncoras de navegação, onde menus de opções auxiliam esta navegação.

Os eventos de interface especificados no Projeto da Interface Abstrata são implementados através de métodos das classes *ADV_Aplicação*.

4.3 Descrição da Navegação

A navegação pode ser baseada em contextos ou baseada em seleção. O primeiro caso contempla a navegação baseada em botões e em âncoras fixas que ocorre inter e intra contextos de navegação. O segundo caso contempla a navegação baseada em seleções no conteúdo dos nós ou em consultas efetuadas sobre o SGBDH. Neste segundo caso, o nó pode ser apresentado dentro de um contexto *default*, sendo este geralmente um contexto derivado de classe do nó.

A navegação baseada em contexto é implementada como método dentro dos contextos de navegação, acionado pela ativação de âncoras de navegação. A navegação baseada em seleção é implementada como funções associadas a um menu de navegação na interface do nó. Ambas utilizam o mesmo mecanismo, descrito a seguir.

A navegação ocorre em 5 passos, como mostrado na figura 4.14:

1. Quando o usuário solicita alguma ação de navegação sobre a interface, uma âncora apropriada deve ser selecionada. A ação do usuário pode tomar duas formas. Se

a ação é baseada na escolha de alguma opção sobre a ADV (e.g., pressionar um botão), a âncora correspondente ao botão é escolhida. Por outro lado, se a ação é baseada em uma seleção sobre o conteúdo do nó, representado pela ADV, uma âncora transiente é gerada.

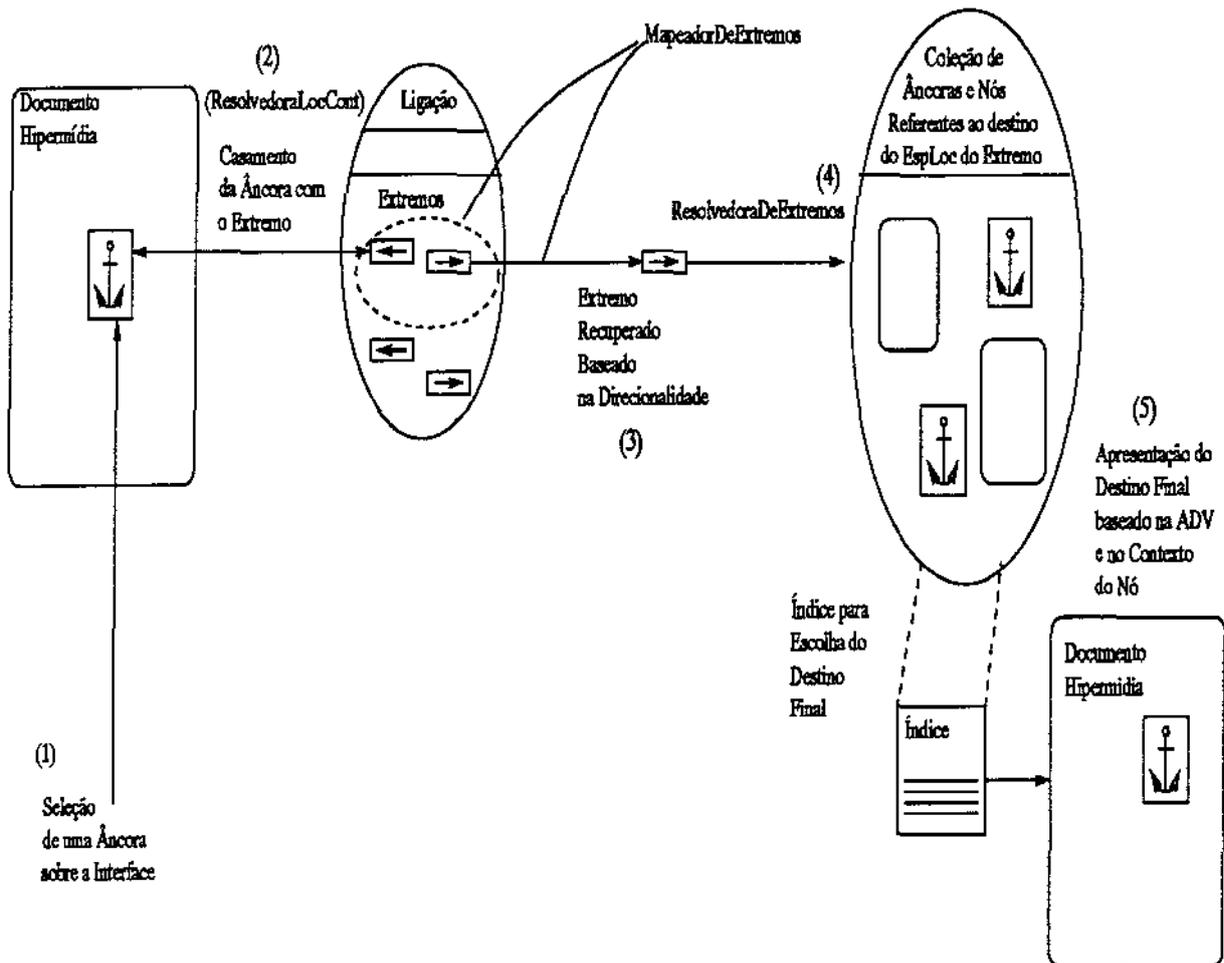


Figura 4.14: Descrição da Navegação

- Este passo toma a âncora derivada do passo anterior e recupera o conjunto de todos os extremos que casam com esta âncora. A partir destes extremos, é fácil alcançar a ligação correspondente através do atributo `IDPai`. O processo de casamento da âncora com o extremo usa o método `IgualLocCont` da âncora. Este passo utiliza uma operação primitiva chamada `ResolvedoraLocCont`.
- Este passo recebe um conjunto dos extremos da ligação e retorna um outro subconjunto dos extremos desta mesma ligação. Por exemplo, um extremo origem pode ser mapeado para extremos que são destinos daquela ligação. Este passo depende de

uma operação chamada *MapeadorDeExtremos* que usa um parâmetro “ADIANTE/DE VOLTA/ QUALQUER”, determinando a direcionalidade do percurso.

4. Este passo utiliza os *EspLocs* dos extremos gerados no passo 3 e segue as instruções definidas pelos atributos deste *EspLoc* para procurar o contexto / nó / âncora que são destinos ou para executar a computação especificada no *EspLoc*. Isto é feito através de uma operação denominada *ResolvedoraDeExtremos*. O cálculo dos destinos pode utilizar parâmetros como, por exemplo, o nó em que a âncora foi acionada. O resultado deste passo é uma coleção de nós e/ou localizações dentro dos nós.
5. Neste passo, os destinos da ligação são apresentados segundo as especificações das ADVs correspondentes. Caso haja mais de um destino possível, pode ser necessário apresentar um índice intermediário, através do qual o usuário poderá escolher o nó destino final de seu interesse. A apresentação do destino final implica na instanciação do Nó de Contexto referente ao objeto de navegação (nó) requerido. Esta apresentação pode solicitar a ativação da visão correspondente ao contexto, caso esta não esteja disponível. A apresentação é efetuada instanciando um objeto da classe Nó de Contexto e preenchendo-o com o conteúdo do seu respectivo nó e através da criação e da instanciação dos objetos de Informação Contextual (*ic*) e de ADV (*adv*) para o Nó de Contexto em questão. A mensagem *obter-obj-cont* é enviada à classe Informação Contextual para a devida instanciação dos atributos do *ic*. A apresentação do Nó de Contexto propriamente dita ocorre através do envio da mensagem *apresentar* para a ADV do Nó de Contexto.

4.4 Considerações sobre o Modelo

4.4.1 Trabalhos Relacionados

Alguns ambientes para desenvolvimento de aplicações hiperfídia (e.g., [BWAH96, MSL96]) utilizam SGBDOOs como meio de armazenamento. O modelo aqui proposto estende aquele apresentado em [MSL96], que implementa os conceitos básicos do modelo do OOHDm sobre um SGBDOO. A proposta de [MSL96] possibilita o desenvolvimento de uma aplicação hiperfídia explorando-se unicamente as ferramentas de um SGBDOO. O acesso ao modelo de dados do SGBDOO, em forma de metadesquema, permite ao projetista da aplicação implementar os modelos orientados a objetos de sua aplicação hiperfídia, projetados com a metodologia do OOHDm, diretamente sobre o ambiente, sem precisar adaptar o seu modelo aos mecanismos de implementação oferecidos.

O presente trabalho conserva estes aspectos, mas o faz estendendo o modelo anterior de forma a adaptá-lo a um ambiente de biblioteca digital. Estas extensões estão focadas no atendimento dos requisitos de um modelo hipermídia para uma biblioteca digital levantados no capítulo anterior. Dentre os requisitos atendidos citamos: a abertura do sistema; o suporte a ligações dinâmicas e genéricas; o suporte a consultas sobre o conteúdo dos nós baseadas na informação de metadados.

O trabalho de Gronbaek e Trigg [GT96] foi uma fonte inspiradora para o atendimento de alguns destes requisitos. [GT96] descreve um modelo baseado em Dexter estendido, oferecendo conceitos tais como EspLocs e Especificadores de Referências, capazes de modelar ligações dinâmicas, ligações que são embutidas no conteúdo do nó, além de navegação baseada em seleção de conteúdo. Contudo, o modelo de dados é voltado ao projeto de *sistemas hipermídia*, sem considerar as especificidades do projeto de uma aplicação de biblioteca digital, interesse do nosso trabalho.

Outro trabalho relacionado é o de [FHN97], que apresenta um metamodelo implementado sobre um BDOO para desenvolvimento de aplicações hipermídia. Contudo, este modelo não tem uma metodologia de projeto clara associada, é baseado em modelagem ER, possui um conjunto limitado de mecanismos de estruturação e de organização da aplicação, além de não possuir qualquer noção de projeto de interface.

4.4.2 Características Específicas do Modelo

Uma das principais contribuições de nosso trabalho é integrar as abordagens de modelos de dados para sistemas hipermídia e para aplicações hipermídia de [GT96] e [MSL96] em um único modelo passível de implementação em um SGBDOO, procurando diminuir a distância semântica entre elas. Esta integração implica em profundas modificações em ambas as propostas. Estas modificações e extensões, por sua vez, são direcionadas a adequar o modelo resultante a um ambiente de biblioteca digital. Com este objetivo, são introduzidas soluções originais como: a definição dos contextos de navegação como visões de banco de dados; a inclusão de uma classe Metadados, raiz de uma hierarquia de subclasses de metadados, que é imprescindível ao suporte a consultas por conteúdo e à interoperabilidade; e ligações dinâmicas que modelam ligações de contextos e caminhos de navegação dentro dos contextos. Algumas destas características são discutidas a seguir.

Âncoras e Ligações

No modelo aqui proposto, foram alterados os conceitos hipermídia básicos de âncoras e ligações do modelo do SGBD OOHD. Estas alterações utilizam as idéias propostas por Gronbaek e Trigg em [GT96]. As alterações visam reduzir o esforço de autoria em aplicações hipermídia de grande porte (no caso em particular, uma biblioteca digital),

tornando-as mais fáceis de modificar, customizar e estender.

As alterações permitem representar e armazenar ligações como objetos separados dos nós que elas conectam através de associações indiretas entre âncoras e extremos de ligações. Permitem também representar ligações para objetos que existem fora da hiperbase como, por exemplo, URLs WWW, contribuindo para a abertura do sistema. Além disto, estas alterações fornecem métodos para especificar ligações embutidas em um nó ou documento sem necessidade de marcações.

Contextos de Navegação e Ligações de Contexto

Na abordagem de Milet et al. [MSL96], os Contextos de Navegação são classes do banco de dados que armazenam um conjunto de ligações de contexto como atributos, possuindo métodos para gerar tais ligações. Existem ainda métodos para obter os nós de contexto que sucedem e antecedem o nó de contexto que ativou a navegação. Os métodos e ligações de contexto são os responsáveis pela implementação dos caminhos de navegação.

Diferentemente, em nossa abordagem, duas únicas ligações de contexto são utilizadas para implementar o caminho de navegação dentro do contexto, dispensando a necessidade de manter listas de ligações de contextos. Estas listas eram necessárias em Milet porque para cada âncora em um nó deveria haver uma ligação correspondente. Estas listas poderiam crescer enormemente frente a uma expansão da biblioteca. A chave para o projeto de nossas ligações é o seu aspecto dinâmico e genérico. A estrutura genérica da ligação de contexto para a navegação intra e intercontextos é partilhada por todos os nós do contexto de navegação. Estas ligações de contexto estão indiretamente associadas às âncoras para navegação, permitindo o compartilhamento de tipos de ligações. A estrutura destas ligações automaticamente comporta o crescimento e as alterações na base de informação da biblioteca. Isso é alcançado determinando a origem das ligações de contextos em termos das âncoras de navegação presentes em todos os nós, ao invés de estarem associadas a nós específicos, e especificando o caminho de navegação em termos genéricos dentro do método para o cálculo do caminho. Este método é aplicado às instâncias dos nós presentes no contexto para determinação do nó de contexto destino.

Outra característica diferenciadora da nossa abordagem é que a semântica dos métodos *obter-próximo* e *obter-anterior*, presentes nos contextos de navegação do metamodelo de Milet e responsáveis pela implementação do caminho de navegação seqüencial dentro do contexto, é incorporada à ligação de contexto. Esta característica é coerente com o papel da ligação de contexto em fornecer ordenação e em permitir a navegação pelos nós de contexto. Os métodos podem fazer uso de comandos SQL do próprio SGBDH sobre as instâncias presentes na extensão da visão.

Esta forma de definição de Contextos de Navegação e de Ligações de Contexto visa permitir a definição do modelo para implementação do Contexto de Navegação como

visões de bancos de dados. A nossa abordagem traz vantagens em relação a aspectos como espaço de armazenamento e gerenciamento de atualizações nas extensões das classes armazenadas, além de flexibilizar a definição dos contextos e de sua hierarquia.

Dinamismo

O modelo proposto oferece estruturas hipermídia dinâmicas a partir do uso de ligações computadas e genéricas. Estas ligações são, por sua vez, cruciais no suporte à hipermídia baseada em conteúdo [LDG+96]. Ligações baseadas em seleções de porções de texto, em regiões de imagens, ou em componentes de objetos compostos são bem suportadas pela abordagem.

As características do modelo são particularmente úteis em sistemas nos quais não existe distinção entre usuários e autores. Usuários poderiam gerar quaisquer tipos de ligações a seu bel prazer (inclusive com consultas à hiperbase), sendo que o próprio usuário seria responsável por manter a organização e a legibilidade de suas ligações.

A necessidade de ligações dinâmicas aparece naturalmente em uma fase posterior do ciclo de vida da aplicação para atender necessidades surgidas no decorrer da utilização do sistema.

A utilização de ligações dinâmicas e genéricas traz benefícios tanto para o usuário quanto para o autor. O usuário tem a liberdade de poder selecionar qualquer informação que considere relevante e de poder verificar se existem ligações que partem desta origem específica, livrando-se da "tirania do botão". Os autores têm, por sua vez, a liberdade de fornecer ligações genéricas provendo serviços padrões para todo o domínio de informação ao invés de prover botões explícitos para a para toda informação relevante.

O modelo suporta bem, ainda, a questão da evolução do esquema da aplicação através de sua característica de extensibilidade, pois a inclusão de novas ligações em objetos previamente existentes e de novas classes de navegação tem um tratamento mais adequado através do uso de ligações genéricas e de contextos de navegação como visões de bancos de dados.

4.5 Resumo

Este capítulo apresentou o modelo de dados do SGBD Hipermídia para uma Biblioteca Digital. O modelo apresenta características diferenciadoras tais como o uso de um SGBDOO para garantir funcionalidades de SGBDs a aplicações hipermídia de biblioteca digital; integração das camadas da aplicação hipermídia; manutenção de restrições de integridade entre objetos da biblioteca; facilidade de desenvolvimento de aplicações. Dentre as vantagens e contribuições do modelo, podemos citar:



1. Diminuição da distância semântica entre as abordagens de modelos de dados para sistemas hipermídia e para aplicações hipermídia

Alcançada através da integração de conceitos do modelo para aplicação OOHDM com conceitos do Modelo de Dexter Estendido para projeto de sistemas hipermídia.

2. Abertura do Sistema Hipermídia da Biblioteca Digital

Efetuada através de: a) separação das estruturas dos nós, âncoras, extremos e ligações; b) ausência de necessidade de formatos específicos para descrição de conteúdo (e.g., HTML, VRML) ou de marcadores dentro dos nós; c) ligações que suportam modificações e alterações no conteúdo dos dados, que suportam vários tipos de mídia e ligações sobre estas mídias (referenciamento de âncoras em porções de dados multimídia)

3. Ligações Dinâmicas

Suportadas por consultas dentro dos EspLocs e por casamento indireto entre âncoras e extremos

4. Flexibilidade na definição de Contexto de Navegação como Visões em bancos de dados

Obtidas através da definição de contextos como visões. Além disto, as visões permitem particionamento lógico dos objetos digitais da biblioteca, gerenciamento do contexto através do mecanismo de visões do BD subjacente, suporte à segurança e integridade dos objetos digitais.

5. Inclusão do Registro de Metadados

Fornecendo suporte a consultas por conteúdo em dados multimídia e consultas a aspectos extrínsecos destes dados; possibilidade de uso de vários padrões de descrições, aumentando a interoperabilidade; e gerenciamento facilitado e uniforme de consultas sobre os pontos de vista de usuário e sistema.

O próximo capítulo mostra como o modelo pode ser utilizado na definição de uma aplicação específica - uma Biblioteca Digital Geográfica.

Capítulo 5

Modelagem da Biblioteca Digital Geográfica

O gerenciamento de informação geográfica diz respeito à organização e ao acesso à informação geo-referenciada através de características geográficas tais como latitude, longitude, regiões envolventes ou limites jurisdicionais [KJSK95]. O capítulo 3 discutiu modelos e metodologias para aplicações hipermídia e o capítulo 4 propôs um modelo para bibliotecas digitais. Este capítulo mostra como modelar bibliotecas digitais geográficas utilizando a metodologia escolhida e apresenta um mapeamento deste modelo para o SGBDH subjacente.

O capítulo está organizado da seguinte forma. A seção 1 introduz as extensões propostas à metodologia do OOHDH com o objetivo de utilizá-lo na modelagem de uma biblioteca digital. A seção 2 descreve os diversos passos da metodologia OOHDH estendida, aplicada à modelagem da biblioteca digital geográfica. A seção 3 apresenta o mapeamento para o esquema de um SGBDOO. A seção 4 sumariza o capítulo.

5.1 Introdução

O capítulo anterior apresentou um modelo hipermídia que estende o do OOHDH para bibliotecas digitais. Propomos aqui algumas extensões à metodologia do OOHDH para utilização do novo modelo. Estas extensões correspondem a: 1) especificação de novas possibilidades para consulta usando o SGBD e formulários específicos para o domínio da aplicação (no caso em particular, para dados geográficos); 2) ampliação das formas de navegação, considerando que o modelo permite consulta baseada em conteúdo. Estas extensões, na verdade, não modificam a metodologia, mas estendem o conceito de navegação suportado por esta metodologia, de forma a considerar um maior leque de possibilidades de interação. Não são, assim, propostas novas primitivas de modelagem, mas sim no-

vas formas de uso da metodologia para considerar bibliotecas digitais geográficas. As seguintes extensões são propostas:

1. Modelagem dos aspectos de consulta à biblioteca digital

O aspecto de modelagem de consultas é pouco coberto em OOHDM. Em contrapartida, no âmbito de bibliotecas digitais este aspecto é tão ou mais importante que o aspecto navegacional. De forma a cobrir esta lacuna, sugerimos a introdução de diversas extensões nas fases do OOHDM. Estas extensões são:

(a) Projeto Conceitual

Nesta fase, é necessário definir conjuntos de metadados para a descrição dos objetos da biblioteca. Esta extensão objetiva dar suporte a consultas baseadas no conteúdo dos objetos e em suas propriedades extrínsecas. Como vimos no capítulo 2, metadados são fundamentais para acesso a bibliotecas digitais.

(b) Projeto Navegacional

Uma atividade extra desta fase envolve a definição de conjuntos de serviços de busca e de consulta que devem ser providos pela biblioteca, conforme visto no capítulo 2. Estes serviços são acessados via navegação hipermídia e devem ser fornecidos por aplicações dedicadas conectadas à biblioteca (aqui denominadas aplicações-serviço). Sendo assim, extensões adicionais devem cobrir a introdução de âncoras e ligações para permitir o acesso às aplicações-serviço e a especificação de como estas são integradas com o paradigma navegacional do OOHDM e, conseqüentemente, com o nosso modelo.

(c) Projeto de Interface

Nesta fase, as extensões envolvem a especificação, nas ADVs, de *Interfaces de Consulta* formadas pelas âncoras de acesso às aplicações-serviço.

2. Modelagem do Paradigma de Navegação “Seleção/Ação”

Outra extensão proposta, que diz respeito aos tipos de navegação considerados no OOHDM e que tem reflexos no Projeto Navegacional e no Projeto de Interface, inclui a modelagem da navegação baseada no paradigma “seleção/ação”. Neste estilo de navegação, o usuário seleciona porções dos dados de um nó (por exemplo, um trecho de texto ou uma região de uma imagem) e efetua alguma ação sobre ela. Uma ação típica é a requisição de navegação para informações relacionadas à porção do dado selecionada. Este estilo de navegação estende o paradigma navegacional do OOHDM - que é baseado unicamente em “clicks” sobre âncoras específicas modeladas dentro do nó e representadas na interface. O estilo em questão utiliza o conceito de âncoras transientes. A navegação baseada em seleção usa o conteúdo do nó como a base para

a navegação. Ela aumenta as possibilidades de navegação do usuário, atendendo a necessidades específicas, e explora ligações dinâmicas genéricas. A modelagem de navegação baseada em seleção pressupõe atividades extras, tais como definir conjuntos de informações que são reativas, especificar as ligações dinâmicas genéricas para as âncoras transientes, introduzir um Menu de Navegação na ADV para dar suporte a este estilo de navegação e especificar quais ações sobre a ADV dispõem esta navegação.

O restante deste capítulo apresenta a modelagem de uma Biblioteca Digital Geográfica utilizando o modelo do capítulo 4. Aspectos referentes ao uso de metadados e de consultas são tratados no próximo capítulo.

5.2 Modelagem da Biblioteca Digital

5.2.1 Projeto Conceitual da Biblioteca Digital

Definimos uma biblioteca digital geográfica como sendo composta de uma coleção de dados convencionais e geo-referenciados, descritos através das visões de campos e objetos (vide capítulo 2). As principais classes de dados geográficos disponibilizadas pela biblioteca incluem: mapas temáticos, modelos numéricos de terreno, imagens, e objetos geográficos complexos ou elementares. Dentre as classes convencionais oferecidas, existem documentos, softwares de aplicação e bancos de dados. A seguir, descreveremos cada um dos tipos de dados disponibilizados com maiores detalhes.

- Mapas Temáticos

São mapas que mostram uma região geográfica particionada em polígonos segundo os valores relativos a um tema (por exemplo, uso do solo ou aptidão agrícola). Os polígonos são resultados de funções de análise e classificação dos dados e não correspondem a elementos identificáveis do mundo real.

- Modelos Numéricos de Terreno

São representações de grandezas que variam continuamente no espaço. São comumente associados a altimetria e também podem ser utilizados para modelar unidades geológicas, como teor de minerais ou propriedades do solo ou do subsolo.

- Imagens Geográficas

São imagens que têm uma semântica geográfica associada. Estas imagens são obtidas através de técnicas usadas para medir propriedades eletromagnéticas de uma superfície ou de um objeto, sem que haja contato entre o objeto e o equipamento

sensor. Com o desenvolvimento das áreas de Sensoriamento Remoto e de Processamento de Imagens, imagens vêm sendo utilizadas cada vez mais em aplicações geográficas.

- **Objetos Geográficos**

São dados descritos por um modelo vetorial, podendo ser elementares ou compostos. Podem também ser usados para modelar *redes*. Uma rede é representada conceitualmente por um grafo no qual cada vértice é um objeto geográfico, apresentando restrições de conectividade acrescentadas ao grafo. Estas redes estão geralmente associadas a serviços de utilidade pública tais como água, luz e telefone.

- **Documentos**

São dados que incluem qualquer tipo de documento multimídia como textos, áudios, vídeos, em qualquer formato; quaisquer informações geradas por aplicações *third-party* e mídias *read-only* como CD-ROMs. Estes dados não têm semântica geográfica.

- **Softwares de Aplicação**

São programas executáveis ou códigos fonte compiláveis, que efetuam processamentos sobre os dados da biblioteca como um todo. Podem incluir, por exemplo, softwares de manipulação de imagens, programas para gerenciar mapas, Sistemas de Informações Geográficas (SIGs) completos, *viewers* de imagens, programas para *display* de áudio e vídeo.

- **Bancos de Dados**

São referências a bases de dados de centros de dados nos moldes de [dCG97]. Estas bases podem ser modeladas no próprio SGBDH ou em bancos de dados externos. A ativação de ligações relacionadas a estas bases permite acesso direto à informação dos centros ou das bases locais.

A primeira fase de metodologia do OOHDM corresponde à modelagem conceitual dos dados da aplicação. Analogamente, os dados da biblioteca digital geográfica devem ser modelados segundo algum modelo de dados específico para dados geográficos. Existem, na literatura, diversos modelos de dados para descrição de dados geográficos (e.g., [OPM97, CCH+96]). Nesta dissertação escolhemos o GMOD [OPM97] como ponto de partida, pois suas características se mostraram adequadas ao atendimento dos requisitos de nossa biblioteca digital. Dentre estas características, citamos a habilidade em definir fenômenos geográficos de acordo com as visões de campos e objetos em um modelo OO; a possibilidade de descrever relacionamentos entre classes de objetos; e a capacidade de

modelar a dimensão temporal. Supomos, assim, que os dados da biblioteca digital são modelados a partir do modelo GMOD modificado para atender às necessidades da biblioteca digital.

A figura 5.1 mostra a estrutura de classes do Projeto Conceitual da biblioteca digital, a partir de modificações do GMOD. As modificações introduzidas no GMOD foram: 1) acréscimo da classe *Metadados*; 2) definição específica de dados multimídia na Classe Convencional, uma classe de objetos não necessariamente geo-referenciados; 3) supressão de alguns relacionamentos previstos no GMOD que não foram considerados relevantes para a nossa biblioteca, pois se prestam mais ao desenvolvimento de aplicações geográficas ambientais. A seguir, descreveremos o modelo GMOD modificado que atuará como o esquema conceitual dos dados de nossa biblioteca.

O GMOD possui duas classes básicas: *Geo-Classe* e *Classe Convencional*. Introduzimos a classe *Metadados*, raiz de uma hierarquia de subclasses que descrevem propriedades dos objetos tanto de Geo-Classes quanto de Classe Convencional. Esta hierarquia será definida no capítulo 6. O único atributo da Classe *Metadados* é o OID do objeto que ela descreve.

A Classe Convencional descreve objetos que não são georeferenciados. Classes Convencionais podem ser de três tipos básicos: *Documentos*, *Softwares de Aplicação* e *Bancos de Dados*. Estas classes modelam os tipos de dados convencionais disponibilizados pela biblioteca. Estes tipos são descritos por diferentes objetos de metadados.

A Geo-Classe modela objetos com alguma dimensão espacial descrita pelo atributo *Localização*, que é do tipo *Geo-Região*. Objetos da classe *Geo-Região* descrevem alguma região da superfície da Terra de acordo com os atributos *projeção* e *escala*. O atributo *descrição* de uma *Geo-Região* é modelado pela classe *Representação*. *Metadados* têm a mesma semântica que nas Classes Convencionais.

Geo-Classes são especializadas em classes de Geo-Objetos, que descrevem entidades do mundo de acordo com algum modelo de objetos, e de Geo-Campos que correspondem a uma descrição da visão de campos. Geo-Campos possuem, além do atributo *Localização* que descreve a região a que eles se referem, dois atributos adicionais: *contradomínio* e *mapeamento*. Geo-Campo é especializado nas classes *Temático*, *Numérico*, e *Sensoriamento Remoto*, modelando respectivamente dados do tipo mapa temático, modelos numéricos de terreno, e imagens. Geo-Objetos podem ser de dois tipos: *Elementares* ou *Compostos*. Geo-Objetos Compostos são constituídos de outros Geo-Objetos (representado na figura 5.1 pelo relacionamento de agregação).

A dimensão temporal dos dados da biblioteca é modelada pela classe *Tempo* que se relaciona com todas as demais classes (representado na figura pelo relacionamento com o retângulo envolvente às classes).

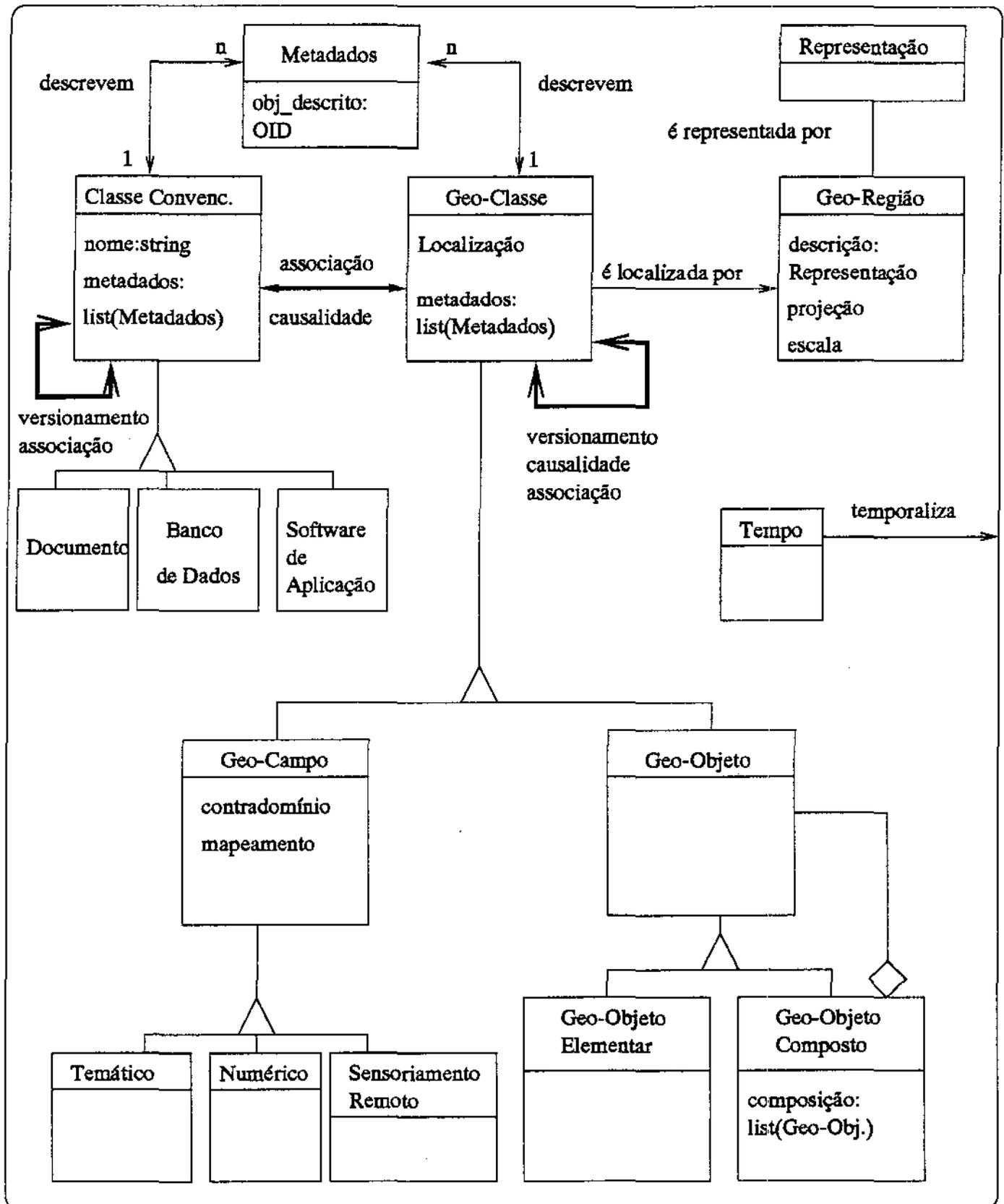


Figura 5.1: Projeto Conceitual - Esquema Conceitual

As linhas mais grossas, na figura, representam relacionamentos que podem existir entre as próprias classes (e.g., *auto-relacionamento de versionamento*) e entre Geo-Classes e Classes Convencionais (e.g., *relacionamentos de associação e causalidade*).

A seguir descreveremos as funcionalidades específicas de algumas das classes e dos relacionamentos do Projeto Conceitual.

1. Metadados

São conjuntos de termos, definições e valores que descrevem dados espaciais e convencionais. Os objetos de metadados abstraem o conteúdo de um objeto da biblioteca, permitem aos usuários encontrarem e acessarem os dados. Mais detalhes sobre esta classe se encontra no capítulo 6.

2. Geo-Campos

Geo-Campos seguem a definição de [CCH⁺96], tendo atributos contradomínio e mapeamento para modelar a visão de campos. Na classe *Temático*, por exemplo, o contradomínio V é um conjunto finito arbitrário, cujos elementos definem os temas do Geo-Campo. [CCH⁺96] cita um exemplo em que um Geo-Campo de pedologia de uma Geo-Região é caracterizado pelo conjunto de temas $V = \{\text{latosolo roxo, litosolo, cambisol, ...}\}$, conforme os tipos de solos encontrados.

3. Representação

Esta classe é raiz de uma hierarquia de classes que modelam representações geométricas e topológicas das propriedades espaciais dos objetos.

4. Tempo

As classes do banco de dados podem ser temporais ou atemporais de acordo com a possibilidade de suas instâncias variarem ou não com o tempo. O tempo é um aspecto fundamental no que tange dados geográficos. A dimensão temporal pode ser considerada sobre dois eixos de tempo: o *tempo de transação* (que descreve quando o dado foi realmente armazenado) e o *tempo válido* (que descreve como os valores variaram no mundo real).

5. Relacionamento de Associação

O relacionamento de associação corresponde a relacionamentos do modelo ER, permitindo estabelecer diferentes tipos de conexões entre objetos da biblioteca. Um relacionamento de associação pode ser estabelecido entre quaisquer duas instâncias da biblioteca.

6. Relacionamento de Causalidade

Relacionamentos de causalidade estabelecem ligações de causa e efeito entre objetos modelados na biblioteca, correspondendo, na prática, à modelagem dinâmica. Por exemplo, uma imagem de satélite de uma tempestade pode ter um relacionamento de causalidade com um mapa temático de índice pluviométrico de uma determinada região.

7. Relacionamento de Versionamento

Relacionamentos de versionamento permitem conectar versões de um mesmo conceito. Em nossa biblioteca, consideramos apenas *versões temporais* dos objetos digitais. O conjunto das versões interconectadas de um objeto digital constitui a *história* deste objeto. Assim sendo, o relacionamento de versionamento está restrito a um auto-relacionamento entre instâncias de uma mesma classe.

8. Relacionamentos de Metadados

Estes relacionamentos representam uma conexão existente entre um objeto da biblioteca digital e os diferentes objetos de metadados que o descrevem.

5.2.2 Projeto de Navegação da Biblioteca Digital

Esta seção define o esquema das classes de navegação e o esquema de contextos de navegação, de acordo com os passos da metodologia OOHDm.

Esquema das Classes de Navegação

O modelo GMOD estendido serviu como modelo conceitual da fase de Projeto Conceitual da metodologia do OOHDm. O modelo mostra como modelar dados na biblioteca geográfica. O Projeto de Navegação, por sua vez, define as classes necessárias à navegação hipermídia - as *Classes de Navegação*.

A figura 5.2 mostra o esquema das classes de navegação da biblioteca digital geográfica. A construção das classes de navegação é baseada na incorporação dos atributos referentes aos nós - tais como âncoras - às classes do modelo estendido e na reestruturação destas últimas por processos semelhantes ao uso de visões em BDOO [Ros96]. O conteúdo das âncoras é determinado pelos relacionamentos do GMOD estendido que são, por sua vez, transformados em ligações no Projeto Navegacional. As ligações são representadas pelas linhas mais grossas na figura 5.2.

Por exemplo, a classe de Navegação Geo-Classe, na figura 5.2, tem âncoras *associação*, *causal*, *versão*, *meta_anc* correspondendo respectivamente às ligações *associação*, *causalidade*, *versionamento* e *metainformação*. Em particular, as âncoras *temp_apl* e *loc_apl*,

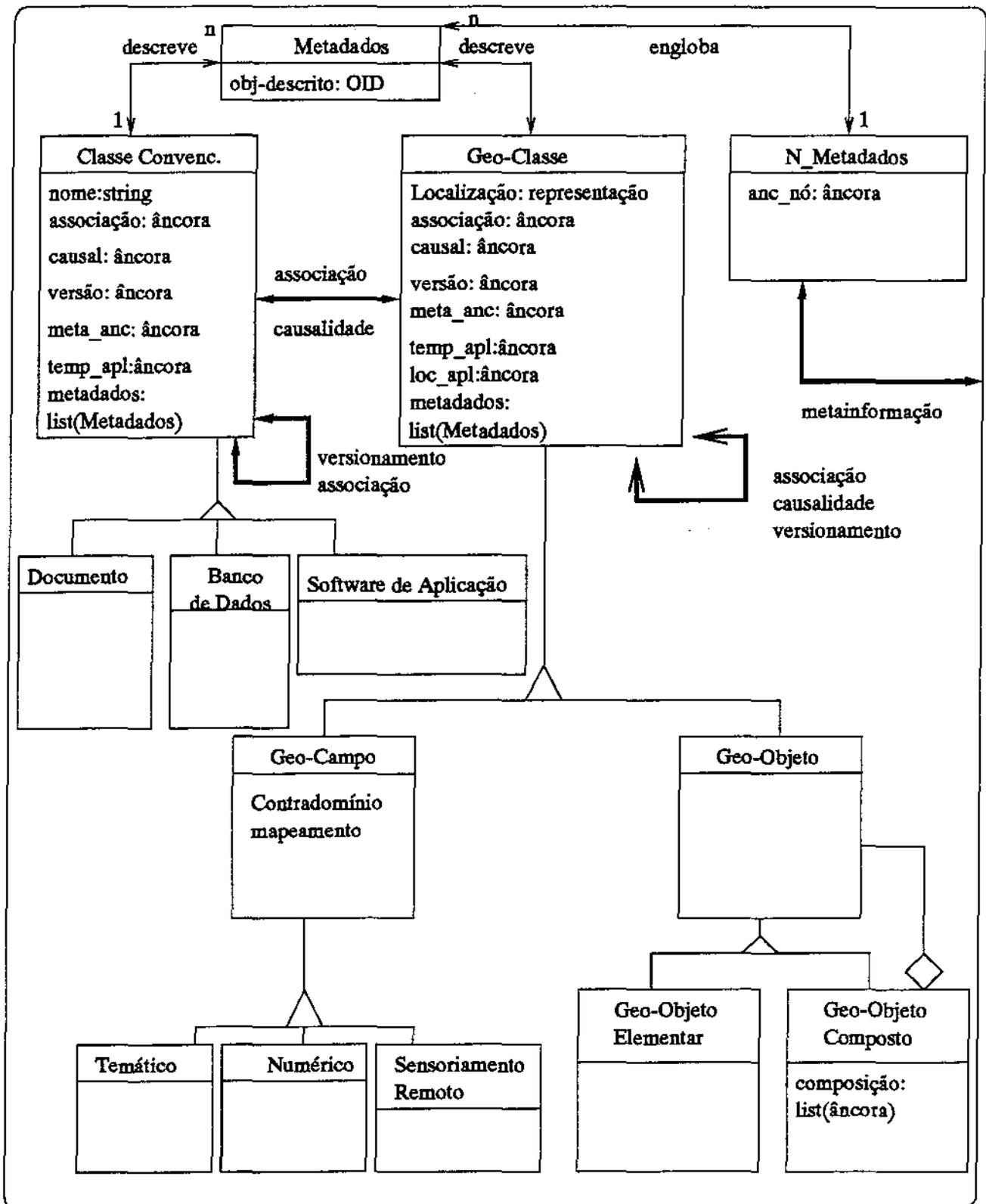


Figura 5.2: Projeto de Navegação - Esquema das Classes de Navegação

que estão associadas aos relacionamentos *é localizada por*, *é representada por*, e *temporaliza* do modelo conceitual, dão acesso a aplicações-serviço específicas que efetuam serviços para a biblioteca. Estas âncoras fazem parte de uma *Interface de Consulta* de extrema importância em nosso projeto. Esta interface e as aplicações associadas fazem parte das extensões propostas ao OOADM e são descritas no próximo capítulo.

No Projeto Navegacional da biblioteca digital geográfica, as classes do modelo conceitual *Representação*, *Geo-Região* e *Tempo* desaparecem, sendo incorporadas à classe de Navegação *Geo-Classe* e à hierarquia de metadados. Os objetos de navegação *N_Metadados* encapsulam a lista de metadados de um nó. O atributo do tipo âncora *anc_nó* permite alcançar o nó ao qual o objeto de navegação *N_Metadados* está associado.

As decisões de projeto do Esquema das Classes de Navegação foram norteadas por algumas idéias básicas. Primeiramente, considerou-se desnecessário definir as classes conceituais *Geo-Região* e *Tempo* como classes de navegação isoladas. Em segundo lugar, as dimensões temporais e espaciais foram consideradas como informações descritivas que permitem encontrar e acessar os dados da biblioteca, configurando, desta forma, o papel de metadado.

Uma conseqüência destas opções de projeto é que possíveis consultas de ligações dinâmicas, bem como de visões referentes aos contextos de navegação são efetuadas tanto sobre os atributos dos objetos quanto sobre seus metadados, ambos modelados no próprio SGBDH. Isto gera um tratamento uniforme das consultas, tanto no nível de sistema para a navegação quanto no nível de usuário para consulta, facilitando o projeto dos componentes de consulta da biblioteca. Outra conseqüência é que o projeto de navegação da biblioteca fica agora intimamente relacionado à informação armazenada nos metadados.

Esquema dos Contextos de Navegação

O esquema dos contextos de navegação mostra como os contextos estão organizados e interligados dando uma noção de possíveis caminhos de navegação. O esquema é organizado com base em alguns pressupostos:

1. navegação e consulta constituem as formas de interação com a biblioteca. Estes mecanismos são fortemente integrados, podendo ser intercambiados arbitrariamente;
2. os contextos organizam os dados da biblioteca, facilitando ao usuário navegar em uma biblioteca digital;
3. O usuário pode fazer consultas sobre a biblioteca como um todo ou dentro de um contexto particular, permitindo uma maior objetividade em suas consultas.

As notação utilizada para definir o esquema dos contextos de navegação, baseada em [Ros96], é apresentada na figura 5.3.

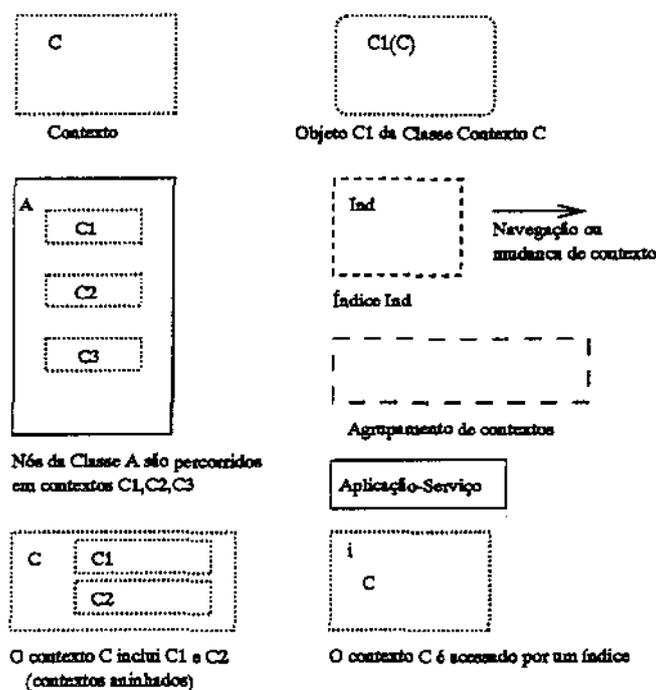


Figura 5.3: Notação para a definição do Esquema de Contextos de Navegação

A figura 5.4 apresenta os contextos de navegação, suas interligações, as aplicações-serviço e os índices de acesso aos nós da biblioteca. A figura está dividida em três blocos principais. O bloco mais à esquerda corresponde ao menu principal para acesso a índices hierárquicos ou a aplicações. O bloco mais à direita descreve os contextos de navegação e suas interligações. As setas indicam as possibilidades de navegação. A notação gráfica utilizada está descrita na figura 5.3.

Os dois primeiros blocos são auto-explicativos. Por exemplo, se o usuário seleciona “Tempo” no menu principal, uma aplicação “Formulário de Tempo” é acionada. Se o usuário, por outro lado, seleciona alguma âncora para índice, é possibilitado o acesso a algum contexto (terceiro bloco).

A chave para o entendimento da figura é a organização dos contextos. A figura mostra que estes estão divididos em três conjuntos básicos: Geo-Classes, Classes Convencionais e Metadados.

O contexto de navegação *Geo-Classe por Localização e Tempo* engloba todos os contextos referentes às Geo-Classes. Esta organização reflete uma semântica que dita que os objetos navegáveis a partir destes contextos estão restritos àqueles que recaem dentro dos intervalos de tempo e localização geográfica especificados pelo usuário.

De maneira análoga, os contextos relativos às Classes Convencionais são definidos dentro do contexto *Classe Convencional por Tempo*. Os contextos que contêm Geo-Classes e Classes Convencionais herdam esta semântica, incorporando apenas objetos contidos nos

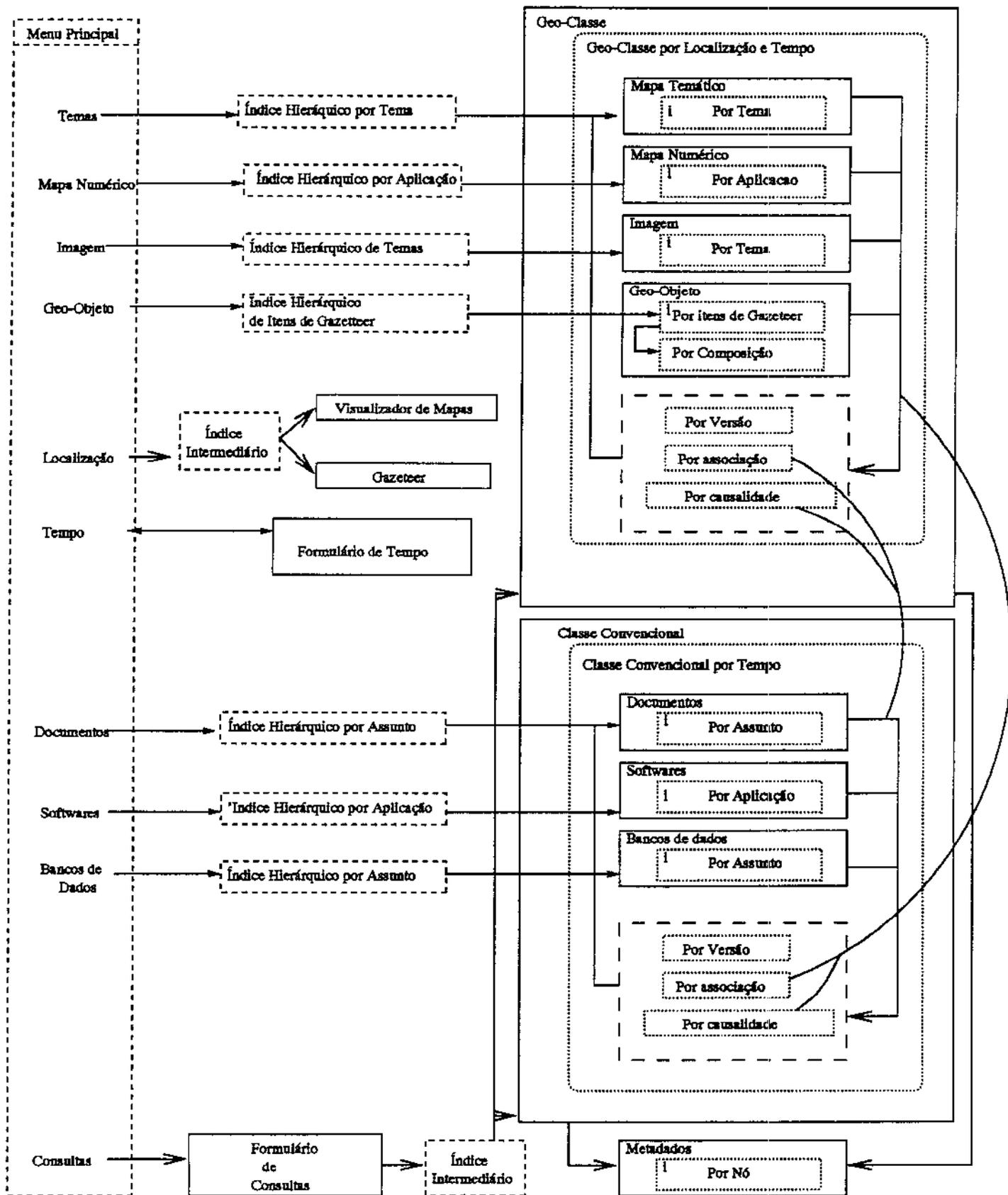


Figura 5.4: Esquema dos Contextos de Navegação da Biblioteca Digital

contextos *Geo-Classe por Localização e Tempo* e *Classe Convencional por Tempo*.

Cada Contexto *Metadados por Nó* dá acesso ao objeto de navegação N_Metadados relativo a um dado nó da biblioteca. A figura 5.4 mostra que contextos desta classe são acessados a partir de qualquer objeto em algum outro contexto da biblioteca.

Os contextos *mapas temáticos e imagens por tema, mapas numéricos por aplicação, documentos e bancos de dados por assunto, softwares por aplicação e metadados por nó* são contextos derivados de classe, sendo acessados via índices ou âncoras nos objetos. Os índices estão relacionados aos contextos e podem estar organizados em diversos níveis de profundidade a partir do menu principal.

Os contextos derivados de ligação são: *Por versão, Por associação e Por causalidade*. Estes contextos estão associados, respectivamente, aos relacionamentos de versionamento, associação e causalidade existentes entre os objetos da biblioteca, quer de Geo-Classes, quer de Classes convencionais. Os objetos destas classes Contextos permitem a mudança de contexto no qual se está acessando um nó; sendo assim denominados *contextos mutáveis*. Como em [Ros96], a figura 5.4 usa uma notação em caixa listrada que define os *contextos mutáveis* e os agrupa, evitando assim a definição das setas de navegação partindo de cada um dos contextos mutáveis para todos os outros contextos da biblioteca. Desta forma, pode-se, por exemplo, navegar do contexto de *Imagens por Versão* para um contexto de *Imagens por Tema* e vice-versa, já que toda imagem tem um tema e uma versão. Não é possível, contudo, navegar de *Imagens por Tema* para *Mapa Numérico de Terreno por Aplicação*, haja visto que não existem imagens no contexto de Mapa Numérico de Terreno.

A seguir, descreveremos os contextos de navegação sob o ponto de vista de estrutura de classes. O aspecto de navegação dinâmica será coberto no próximo capítulo, quando descreveremos o modelo de consulta da biblioteca.

A especificação de um contexto de navegação é feita com o uso da notação de *cartões de navegação* como foi definido em [SB94]. Esta especificação será transformada na definição da visão correspondente ao contexto durante o mapeamento para o esquema do banco de dados. A estrutura de um cartão de navegação genérico é mostrado na figura 5.5.

O *nome* do contexto é escrito na parte superior do cartão, juntamente com o tipo de contexto. A seção *Inclui* do cartão indica quais nós farão parte do conteúdo. Ela pode ser definida por extensão (nomeando-se cada um dos membros do contexto) ou por intensão, usando uma linguagem de consulta. Deve-se explicitar também, no cartão, qual a classe Informação Contextual que está relacionada ao contexto de navegação. Os *pontos de entrada* indicam quais nós do contexto de navegação podem ser acessados de fora do contexto (geralmente o primeiro nó no caminho de navegação). O *caminho* refere-se ao modo como os elementos serão ordenados para a navegação. O contexto pode ainda conter atributos tais como índices que aparecem quando o contexto de navegação é acessado ou

Nome do Contexto de Navegação	Tipo do Contexto
Inclui:	Informação Contextual: (Classes de Contexto, Classes, Objetos)
Pontos de Entrada:	
Caminho:	
Atributos (incluindo metainformação e métodos)	

Figura 5.5: Cartão de Navegação Genérico

métodos específicos da visão correspondente ao contexto.

A figura 5.6 mostra o cartão do Contexto *Geo-Classe por Localização e Tempo*, que exemplifica a utilização destes cartões. Notemos que os parâmetros do retângulo envolvente do objeto e a informação de cobertura temporal foram recuperados dos metadados das Geo-Classes.

Geo-Classe por Localização e Tempo	de classe
Inclui:	Informação Contextual:
\forall g in Geo-Classe g.metadados.FGDC.MBR (intersecciona / esta contido) in Localização() and g.metadados.Cobertura Temporal (intersecciona/ esta contido) in Tempo()	
Pontos de Entrada:	
Caminho:	

Figura 5.6: Exemplo do cartão de contexto Geo-Classe por Localização e Tempo

De forma semelhante, todos os objetos de uma classe Contexto são especificados genericamente por um mesmo cartão. A figura 5.7 mostra um exemplo de uma destas classes. Dado um tema, pode-se pensar em todas os mapas temáticos sobre este tema.

Os contextos derivados de ligação *Por associação* e *Por causalidade* são acessados via índices hierárquicos. O nível mais externo de profundidade deste índice possui entradas que refletem as classes disponibilizadas pela biblioteca, configurando a semântica dos respectivos relacionamentos conceituais associados às ligações. Estes relacionamentos são do tipo *um-para-muitos* e podem ocorrer entre quaisquer classes da biblioteca. No nível mais externo, pode-se escolher a classe de entrada. O nível mais interno permite selecionar

Mapa Temático por Tema	de classe
Inclui:	Informação Contextual:IC_Ordenado
$\forall m \text{ in Mapa Temático} \mid m.\text{metadados.tema}=\langle \text{tema} \rangle$	
Pontos de Entrada: índice ou 1o. no	
Caminho:	
sequencial (ordem : m.metadado.tempo ascendente, m in Mapa Temático)	

Figura 5.7: Exemplo do contexto derivado de classe Mapa Temático por Tema

um ponto de entrada específico dentro do contexto selecionado.

O contexto *Geo-Classe por Versão*, representado na figura 5.8, reflete a semântica de um auto-relacionamento de versão, no qual qualquer Geo-Classe (ou Classe Convencional) pode possuir várias versões. O ponto de entrada no contexto é o nó correspondente à versão mais recente (temporalmente) da instância. Em seguida, o usuário pode navegar sequencialmente através dos outros nós do contexto, organizados em ordem descendente de tempo. A semântica dos contextos Por Versão é inspirada nos *contextos de versão* de Soares et al. [SRC95]. O contexto agrupa nós que representam versões históricas de um mesmo objeto: o relacionamento semântico fundamental entre versões de um objeto é a sucessão temporal.

Geo-Classe por Versão	derivado de ligação
Inclui:	Informação Contextual:IC_Der_Ligação
$\forall g1 \text{ in Geo-Classe} \quad \exists v \mid v=\{g2 \text{ in Geo-Classe} \wedge \text{versão}(g1,g2)\}$	
Pontos de Entrada: 1o. no (default)	
Caminho:	
sequencial (ordem : g2.metadado.tempo descendente, g2 in Geo-Classe)	

Figura 5.8: Exemplo do contexto derivado de ligação Geo-Classe por Versão

A navegação entre contextos permite mudar o contexto no qual está se acessando um nó. Por exemplo, enquanto se está folheando através das versões de uma imagem, pode-se querer ver o conjunto das imagens da biblioteca que possuem o mesmo tema de uma destas versões. A condição para permitir tais mudanças é que a definição do contexto alvo

envolva o conjunto completo das instâncias de uma dada classe [Ros96]. As instâncias de Geo-Classe e Classe Convencional mantêm relacionamentos de versionamento, causalidade e associação de forma a permitir que os contextos derivados de ligação definam partições lógicas do conjunto total das instâncias da biblioteca, possibilitando a navegação intercontextos. Este fato permite a mudança de contexto quando o nó é acessado a partir dos contextos Por versão, Por associação e Por causalidade.

A figura 5.9 apresenta os cartões de contexto para a aplicação da biblioteca em termos genéricos. As informações presentes nestes cartões serão discutidas e retomadas na seção 3 deste capítulo, no momento em que os cartões forem mapeados em visões.

Geo-Classe por Localização e Tempo	de classe	Classe Convencional por Tempo	de classe
Inclui:	Informação Contextual:	Inclui:	Informação Contextual:
$\forall g$ in Geo-Classe g.metadado.FGDC.MBR (intersecciona / está contido) in Localização() and g.metadado.Cobertura_Temporal (intersecciona / está contido) in Tempo()		$\forall c$ in Classe Convencional g.metadado.Cobertura_Temporal(intersecciona / está contido in Tempo)	
Pontos de Entrada:		Pontos de Entrada:	
Caminho:		Caminho:	
Metadado por Nó	de classe	<Classe> por <Parâmetro>	de classe
Inclui:	Informação Contextual: IC_Metadados	Inclui:	Informação Contextual: IC_Ordenado
$\forall m$ in Metadado m.obj_descrito = <IDObjeto>		$\forall c$ in <Classe> c.<parâmetro> = <valor-parâmetro>	
Pontos de Entrada:		Pontos de Entrada: índice ou 1o. nó	
		Caminho:	circular, ordenado por c.metadado.tempo
<Classe> por <Relacionamento>	derivado de ligação	Geo-Objeto por Composição	derivado de agregado
Inclui:	Informação Contextual: IC_Der_Ligação	Inclui:	Informação Contextual: IC_Der_Ligação
$\forall c$ in <Classe> c.metadado.Relacionamento.tipo = <tipo_rel> c.metadado.Relacionamento.Identificador = nó-origem()		$\forall g$ in Geo-Objeto g.metadado.Relacionamento.tipo = agregação c.metadado.Relacionamento.Identificador = nó-origem()	
Pontos de Entrada: Índice Hierárquico		Pontos de Entrada:	
Caminho:			
circular, ordenado por c.metadado.tempo			

Figura 5.9: Projeto de Navegação - Os Cartões dos Contextos de Navegação

No capítulo 4, vimos que um nó de contexto é composto por três componentes: nó, informação contextual e adv. Os objetos da classe Informação Contextual complementam a especificação de um nó com informações que serão definidas para este nó quando ele for acessado em um determinado contexto. Estas informações incluem geralmente âncoras *próximo* e *anterior* para suportar o percurso no sentido especificado para o contexto, e informações descritivas (por exemplo, indicação em linguagem natural do relacionamento envolvido na navegação, posição do nó dentro do contexto). Como, na biblioteca, a maioria dos contextos é acessada via seus índices e via folheamento intercontexto baseado em índices, não existe a necessidade de modelar âncoras adicionais às já especificadas nos nós para a navegação intercontexto e às âncoras *próximo* e *anterior* para navegação intracontexto. Não existem, por exemplo, âncoras *próximo* e *anterior* ligadas a outras âncoras da interface de contexto (e.g., âncoras *próxima_associação*, *anterior_associação*) como existem na interface de contexto do Projeto Portinari [Por]. Apenas três âncoras adicionais são inseridas em cada nó: uma âncora para dar acesso ao índice do contexto no qual o nó está sendo acessado (âncora *anc_ind*), uma âncora para o acesso ao Menu Principal da biblioteca (âncora *menu_princ*), e uma terceira âncora para permitir o acesso ao Formulário de Consulta (âncora *consulta*). Em nossa biblioteca, são especificados apenas três tipos de classes Informação Contextual que são compartilhadas pelos nós. A estrutura destas classes é apresentada na figura 5.10, contendo, além do nome da classe e de seus atributos, informações sobre o contexto (atributo *escopo*) e sobre a classe de navegação (atributo *parte_de*), em que estas informações serão acrescentadas.

IC_Ordenado	IC_Der_Ligação	IC_Metadados
anterior:Âncora próximo:Âncora anc_ind:Âncora menu_princ: Âncora consulta: Âncora posição_nó:integer total_nós:integer escopo: todos os contextos de classe (exceto Metadados por Objeto) parte_de:todos os nós destes contextos	anterior:Âncora próximo:Âncora anc_ind:Âncora menu_princ: Âncora consulta: Âncora posição_nó:integer total_nós:integer subtítulo:string escopo: todos os contextos de ligação parte_de:todos os nós destes contextos	menu_princ: Âncora consulta: Âncora escopo: Metadados por Nó parte_de: N_Metadados

Figura 5.10: Classes Informação Contextual da Biblioteca

A classe *IC_Ordenado* é associada aos contextos de navegação derivados de classe e possui atributos *posição_nó* e *total_nós* que armazenam informações a respeito dos valores de posição do nó dentro de um contexto e do total de nós deste contexto. A classe *IC_Metadados* fornece informação relativa ao contexto Metadados por Nó. A classe *IC_Der_Ligação*, por sua vez, é compartilhada por todos os nós inseridos nos contextos derivados de ligação e possui um atributo *subtítulo*, que descreve o relacionamento entre as classes envolvidas na navegação com informações sobre o relacionamento envolvido, o nó origem da navegação e a classe deste nó. Um exemplo de um subtítulo seria “Versões da imagem 1”, indicando que a navegação ocorreu através de um relacionamento de versionamento com um objeto da classe Imagem cujo identificador é o número 1.

Exemplo de Navegação na Biblioteca Digital Geográfica

Nesta seção apresentamos um exemplo que mostra vários tipos de interação de um usuário com a biblioteca. Suponha que o usuário inicia sua navegação na biblioteca digital geográfica selecionando a opção para acesso ao contexto de Imagens por Tema, relativo a uma dada extensão espaço-temporal, para um tema = vegetação. Isto significa que o usuário deseja navegar por conjuntos de imagens, selecionando aqueles que estejam organizados segundo o tipo de vegetação considerado.

A partir desta solicitação de navegação, o SGBD Hipermissão é acionado para construir o correspondente contexto, retornando o conjunto de imagens de vegetação que se encontram dentro da respectiva extensão espaço-temporal. O sistema apresenta então, ao usuário, um índice com um ponto de entrada para cada imagem, que será um objeto da visão correspondente.

Selecionado um ponto de entrada específico, o usuário pode agora navegar através de diversas imagens de vegetação seguindo as ligações associadas às âncoras *próximo* e *anterior* para aquele contexto. Durante a navegação e a visualização dos dados, o usuário tem a oportunidade de averiguar a adequação de suas escolhas e do escopo do contexto. Suponha que o usuário tenha se interessado por uma dada imagem enquanto folheava dentro do contexto. Ele deseja então obter uma perspectiva da evolução histórica da vegetação daquele local através das versões temporais daquela imagem. Isso é obtido navegando-se para o contexto Imagens por Versão a partir da imagem de interesse. Do ponto de vista do SGBD, isto significa “desativar” a visão relativa ao contexto Imagem por Tema e gerar uma nova visão correspondente ao contexto Imagem por Versão, onde o ponto de entrada será exatamente a imagem (por tema) a partir da qual o usuário deseja fazer a nova navegação. Novamente, o usuário pode continuar folheando dentro deste contexto ou navegar para outro contexto de interesse.

Em bibliotecas digitais geográficas convencionais, este tipo de mudança de contexto não é possível. O que ocorre é que o usuário navega em profundidade (por exemplo,

pesquisando todos os tipos de dados possíveis sobre um determinado tipo de imagem) mas as imagens não estão interligadas entre si. A mudança de contexto não é possível já que a noção de contexto não é contemplada, existindo apenas vários índices isolados para cada tipo de tema.

Continuando o exemplo, suponha que, em algum momento, o usuário decida efetuar consultas sobre os metadados de Imagens — por exemplo, todas as imagens cujo tipo de cobertura vegetal seja o mesmo que o da imagem sendo acessada. Isto corresponde a invocar a aplicação-serviço *Formulário de Consultas* a partir do Contexto Imagens por Tema (ou Imagens por Versão). Esta mudança do modo de interação (de folheamento para consulta) é transparente ao usuário. De fato, a ativação do Formulário de Consulta é obtida clicando em informações dependentes de contexto (e.g., uma âncora *Consulta* específica para o contexto).

No caso de bibliotecas digitais geográficas tradicionais, não há esta possibilidade de alternância. No máximo, o usuário pode decidir verificar informações sobre os metadados de *uma* imagem específica, quando então precisa mudar de modo de interação com a biblioteca.

Finalmente, suponha que a consulta sobre metadados retorne para o usuário um conjunto de imagens que satisfaça a consulta (no caso, o OID de cada imagem). O usuário pode agora selecionar alguma destas imagens e, a partir dela, navegar dentro de um dos contextos de imagem. Outra opção é selecionar alguma imagem e utilizá-la como parâmetro de alguma consulta (por exemplo, imagens que tenham sido geradas pelo mesmo autor). Idealmente, o usuário pode mesmo utilizar a imagem para consultas associativas, desde que o SGBD as tenha indexado para permitir, por exemplo, consulta por conteúdo.

5.2.3 Projeto da Interface

De posse das especificações dos nós obtidas durante o Projeto Navegacional, especificaremos, nesta fase, o modelo da interface para cada nó definido para a biblioteca.

De acordo com Vaanamem, citado em [RSLC95], para especificar um modelo de interface, é necessário definir metáforas de interface, descrever suas propriedades estáticas e dinâmicas e seus relacionamentos com o modelo navegacional, de uma maneira independente de implementação. Para o alcance destes objetivos, é necessário especificar:

1. A aparência de cada objeto de navegação (nó) a ser visto pelo usuário.

Um mesmo nó pode ter diferentes apresentações em diferentes situações / contextos.

2. Os objetos auxiliares de interface

Estes objetos podem refletir várias funções da aplicação. Podem aparecer, por exemplo, como barras de menu e botões de controle.

3. Os relacionamentos entre a interface e os objetos de navegação (nós)

A maneira como os eventos externos sobre a interface (e.g., um *click* do mouse sobre o objeto) afetam a navegação e a apresentação do nó.

4. As transformações que ocorrem na interface

Estas transformações podem ser decorrentes da navegação ou de eventos externos, podendo afetar o comportamento de diferentes objetos de interface.

Em nossa biblioteca, não consideramos a ativação e a apresentação de mídia ativas tais como áudio e vídeo. Em sistemas possuindo esta facilidade, aspectos de sincronização de objetos de interface devem ser considerados.

São utilizadas, nesta fase da metodologia OOHDm, as Visões de Dados Abstratos (ADVs) [RSLC95] para especificação da interface. Cada objeto de navegação tem uma ADV. As ADVs são objetos que têm um estado e uma interface. Esta interface é responsável por responder a eventos internos ou externos de entrada e saída. No caso de projeto hipermídia, uma ADV modela um conjunto de atributos que definem as propriedades da interface a serem percebidas pelo usuário e um conjunto de eventos gerados pelo usuário como, por exemplo, solicitações para navegação. Em OOHDm, ADVs estão associadas a nós, ligações, âncoras ou índices, nos quais elas representam algum aspecto destes objetos para o “mundo externo”. ADVs podem ser construídas a partir de outras ADVs, usando abstrações de agregação, generalização e especialização.

Cada ADV de um nó é especificada por um *Diagrama de Configuração*. O Diagrama de Configuração especifica aspectos tais como os objetos de interface que irão compor a ADV, a forma como o usuário irá interagir com o nó, e a definição de objetos de interface que irão ativar a navegação. Os Diagramas de Configuração expressam também padrões de comunicação. Estes padrões explicitam os eventos externos (gerados pelo usuário) tratados por uma ADV, os serviços providos pela ADV (e.g., Display), e as comunicações que ocorrem entre as ADVs e os nós que elas representam.

A estrutura de uma ADV deve refletir a estrutura do nó apresentado. Na representação gráfica de uma ADV, a ADV envolvente mais externa representa um nó inteiro e as ADVs aninhadas são usadas para cada atributo visível do nó. As âncoras que se encontram dentro dos nós são associadas a ADVs ativas aninhadas (por exemplo, botões para a navegação).

Na construção do Diagrama de Configuração de uma ADV, podem ser utilizadas ADVs padrão presentes em bibliotecas de interface tais como textos, imagens, ícones ou botões.

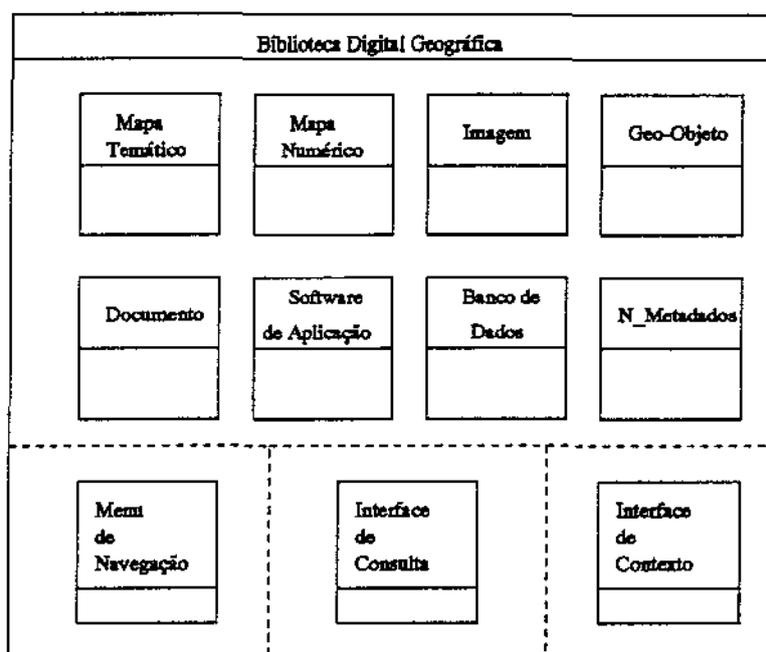


Figura 5.11: ADVs da Biblioteca Digital

As ADVs da biblioteca digital são mostradas na figura 5.11, na qual cada ADV é associada a uma classe de navegação. As ADVs das classes da aplicação nunca são apresentadas ao usuário ao mesmo tempo. As ADVs *Menu de Navegação*, *Interface de Consulta* e *Interface de Contexto*, por sua vez, serão sempre perceptíveis durante toda a navegação, podendo ocorrer supressões, alterações ou variações nas ADVs aninhadas dentro destas.

ADV Geo-Classe A figura 5.12 mostra a ADV <Geo-Classe> composta de diversas ADVs aninhadas: Imagem, Descrição, Menu de Navegação, Interface de Contexto, Interface de Consulta e strings relativos aos atributos da Geo-Classe além de alguns atributos dos seus metadados. O texto relativo à ADV Descrição também é capturado dos metadados do objeto. As ações externas que podem ser efetuadas sobre a ADV estão representadas pelos eventos *Display*, *MouseOn*, *MouseClicked*, *MouseDoubleClicked*, *SelectionOnText*. O botão Metadados da ADV Interface de Contexto permite ao usuário navegar para o contexto de navegação *Metadados por Nó*, onde a informação completa de metadados do nó se encontra disponível.

Os valores a serem mostrados são obtidos usando métodos *obter atributos* e *obter metadados*. Por fins de clareza, resolveu-se “condensar” todas as mensagens referentes ao processo de captura dos valores de atributos nestas mensagens. Concretamente, estas duas mensagens devem ser compreendidas como um conjunto de mensagens *obter nome*, *obter classe*, *obter data*, *obter descrição*, etc. Estas mensagens usam o processamento de

consultas do SGBDOO.

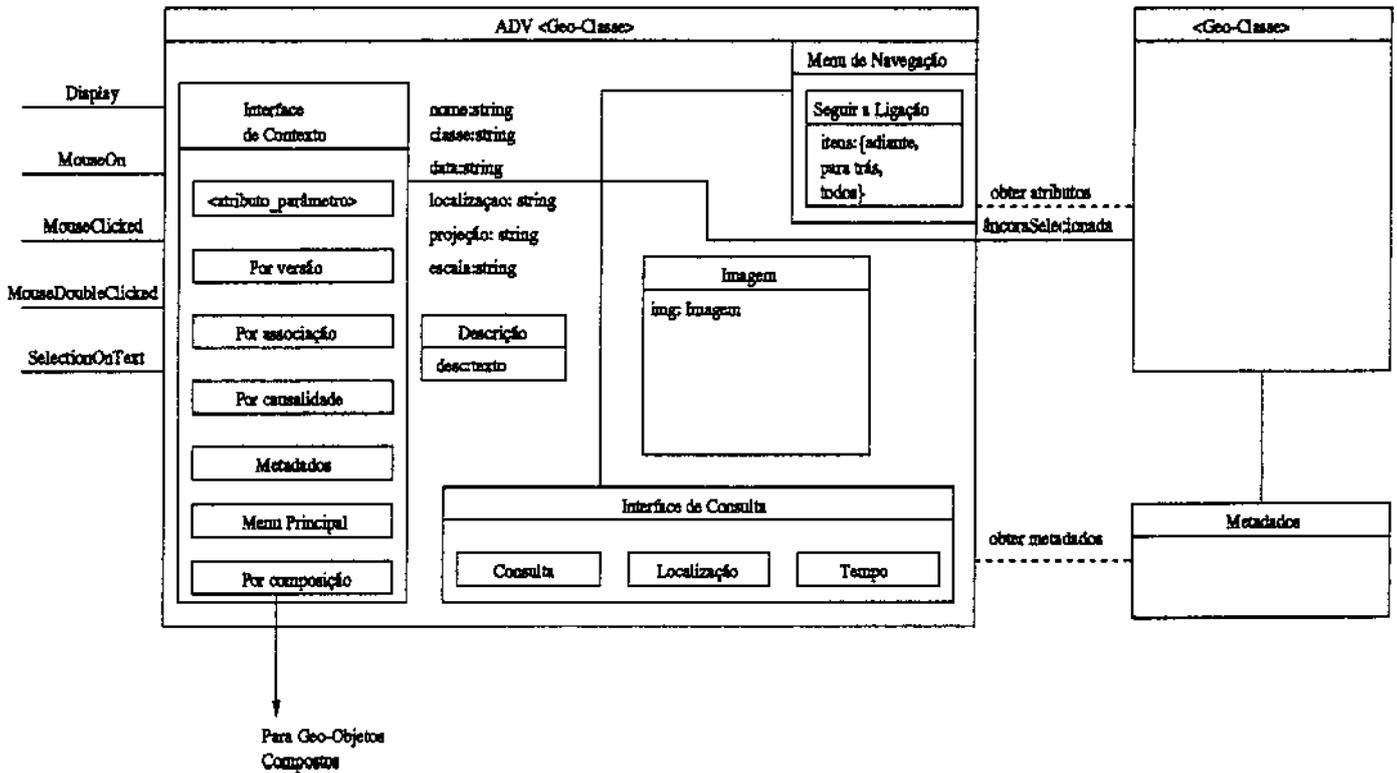


Figura 5.12: Diagrama de Configuração Genérico para os nós derivados de Geo-Classe

A mensagem `âncoraSelecionada` é enviada em duas situações: quando um botão de navegação ou de consulta é pressionado (evento `MouseClicked` com foco no botão) ou quando uma porção de texto ou da imagem no nó é selecionada e é requisitada uma navegação através do menu de navegação (âncora transiente). Em particular, todos os nós reagem à mensagem `âncoraSelecionada(A,direção)`, empacotando a âncora *A* selecionada, conjuntamente com o parâmetro de direcionalidade. O envio desta mensagem inicia o processo de navegação através da busca dos extremos das ligações associadas à âncora *A*. Navegações intra e intercontextos, associadas às ligações de contexto e botões de navegação, possuem sempre o valor `ADIANTE` para o seu parâmetro de direcionalidade.

A ADV `Interface de Contexto` contém ADVs que representam âncoras para a navegação intra e intercontextos de navegação. A ADV `<atributo-parâmetro>` possui um nome que é definido particularmente para cada subclasse de `Geo-Classe` com base no parâmetro da consulta do seu contexto de classe (e.g., `Mapa Temático por Tema`). A ADV para `Geo-Objetos` compostos contém, ainda, um botão adicional *Por Composição*, inserido dentro da `Interface de Contexto` para permitir a navegação em seu contexto derivado de agregado.

O `Menu de Navegação` permite ao usuário seguir ligações baseadas em seleções no

conteúdo do nó (mensagem SelectionOnText) nos sentidos *adiante*, *para trás* ou *todos*, correspondendo, respectivamente, aos valores de direcionalidade ADIANTE, DE VOLTA e QUALQUER.

Quando um nó de contexto é acessado, as informações da classe Informação Contextual são apresentadas. Por exemplo, quando o usuário navega de um dos contextos de classe para um contexto por versão, são introduzidas ADVs aninhadas relativas à informação da classe Informação Contextual, como na figura a seguir. Estas ADVs são apresentadas na Interface de Contexto, em substituição ao botão da âncora *Por versão*.

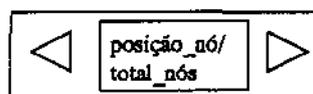


Figura 5.13: ADV para as informações de uma classe Informação Contextual da aplicação

Na figura 5.13 estão configuradas as ADVs dos botões anterior e próximo (setas), para navegação ao longo da história do objeto, as informações relativas ao número de nós no contexto, a posição do nó sendo visto e a ADV do botão para a âncora *anc_ind* ligada ao índice do contexto (retângulo menor com texto na figura 5.13). A navegação usando este botão apresenta um índice com todos os nós do contexto Por Versão.

A Interface de Consulta possui os botões para as âncoras que permitem o acesso às aplicações-serviço referentes ao visualizador de mapas, ao gazetteer e aos formulários de tempo e de consulta da biblioteca.

ADV Classe Convencional As considerações anteriores são válidas para o Diagrama de Configuração da ADV relativa às Classes Convencionais. As principais diferenças em relação à ADV das Geo-Classes são:

1. Atributos

Inclusão de atributos específicos, por exemplo, *publicador* e *tipo de documento* para a classe Documentos; *forma* (e.g., *postscript*, *pdf*, *arquivo executável Windows ou Unix*) para as classes Documento e Softwares de Aplicação; *plataforma* para Softwares de Aplicação; informações sobre o *esquema* para a classe BancodeDados.

2. Interface de Consulta

Exclusão do botão para Localização que dá acesso ao visualizador de mapas e gazetteer. Estas aplicações servem para determinar as regiões geográficas de busca, não cabíveis em Classes Convencionais.

Inclusão de botão para a classe de navegação Banco de Dados para chamada ao programa gateway, responsável pela consulta remota no banco de dados em foco.

ADV N_Metadados O Diagrama de Configuração da classe de navegação N_Metadados é apresentado na figura 5.14. Este diagrama pode ser interpretado de forma semelhante aos anteriores.

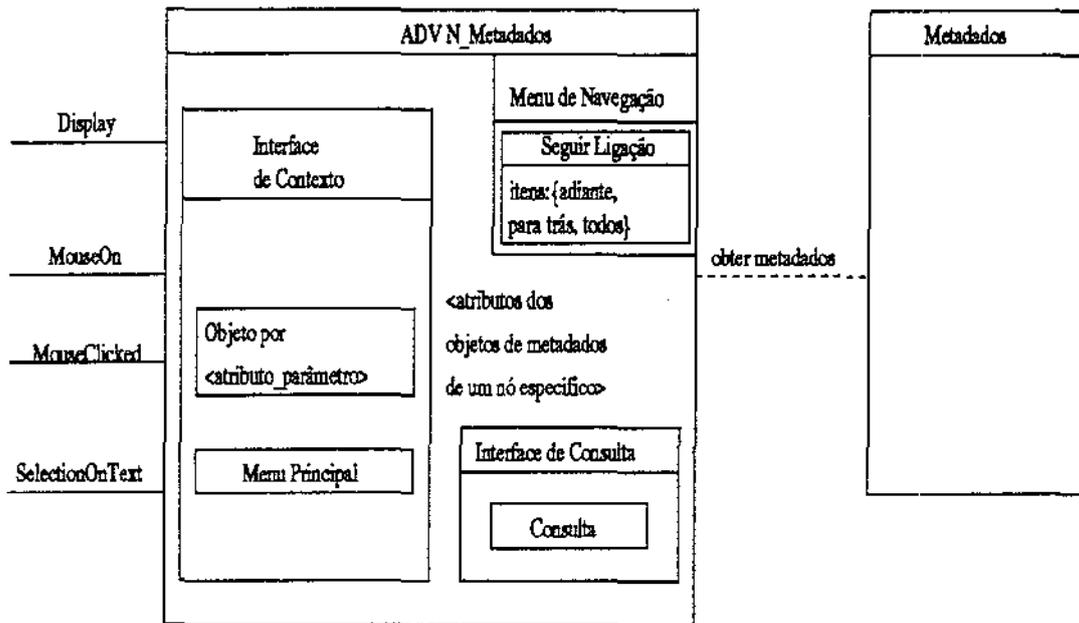


Figura 5.14: Diagrama de Configuração para os nós da Classe N_Metadados

ADVCharts O comportamento de uma aplicação é definido especificando-se a forma como os eventos externos afetam a navegação e a aparência da interface. Esta especificação é realizada através do conceito de *ADVCharts*. *ADVCharts* são uma generalização de *StateCharts* que suportam aninhamento de estados conjuntamente com ADVs. Eles permitem expressar a associação entre eventos externos e ADVs, apresentando os efeitos de cada evento externo em termos de transformações que ocorrem na interface. A utilização de *ADVCharts* envolve a descrição de estados aninhados (aninhamento comportamental) e a descrição do aninhamento das ADVs (aninhamento estrutural). Eles são usados para representar possíveis estados e transições em cada ADV.

A figura 5.15 apresenta genericamente os *ADVCharts* de uma Geo-Classe. As linhas partindo das ADVs aninhadas especificam os eventos e as ações decorrentes. As setas especificam transições de estado. Os retângulos com bordas arredondadas representam estados específicos. A elipse indica o estado em que a ADV está desativada.

Por exemplo, a ADV de uma Geo-Classe pode estar no estado *DisplayOff* (representado na figura pela *elipse Off*) ou no *DisplayOn* (retângulo com bordas arredondadas *On*), de acordo com os eventos *MouseClicked* e *Display*. Note que todos os eventos *MouseClicked*

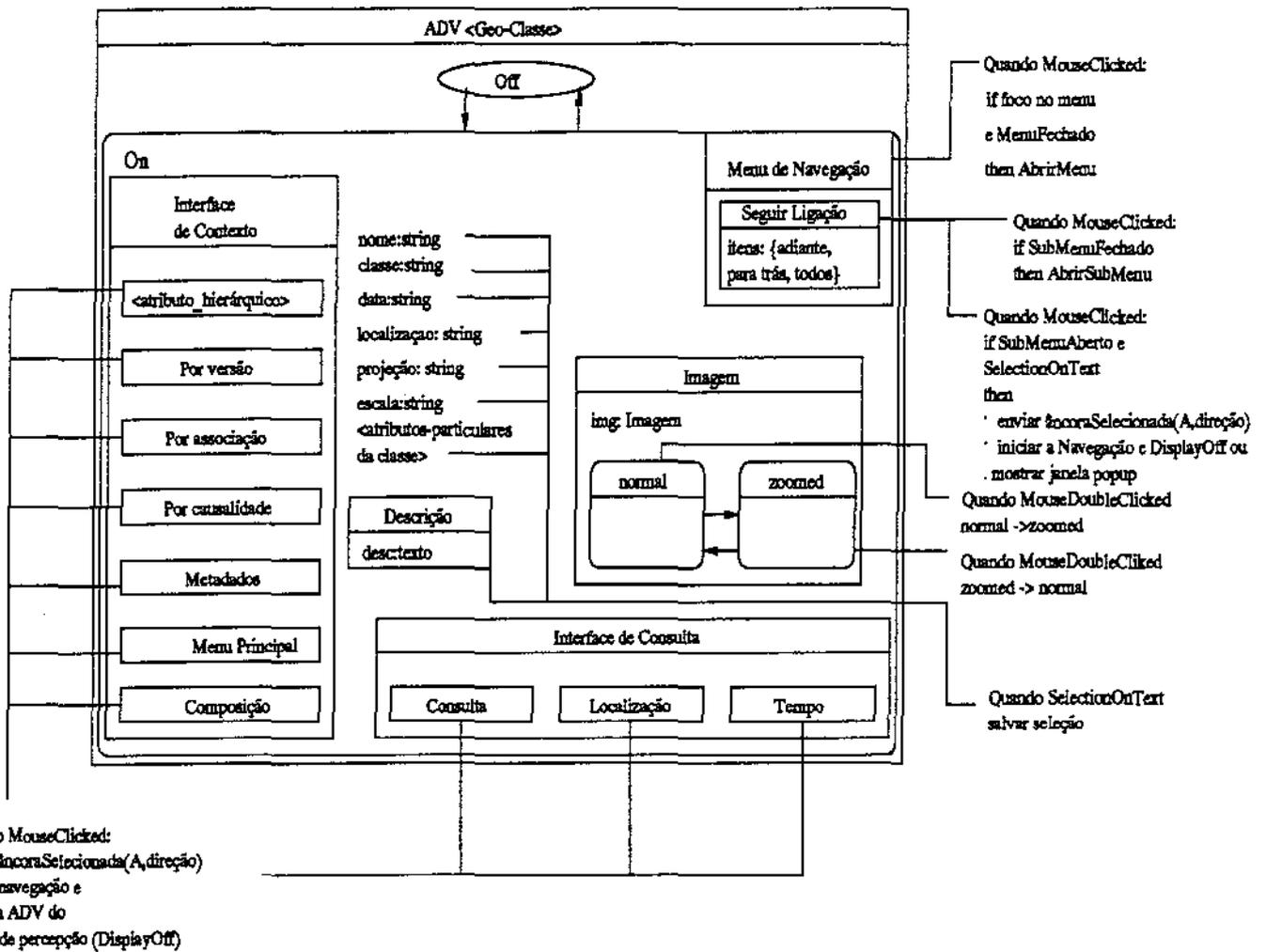


Figura 5.15: ADVChart para a ADV genérica Geo-Classe

que ocorrem sobre a Interface de Contexto e sobre a Interface de Consulta produzem o evento DisplayOff da ADV Geo-Classe.

O ADVChart para Classes Convencionais é construído de forma semelhante.

A figura 5.16 mostra o ADVChart para N_Metadados, que pode ser interpretado analogamente.

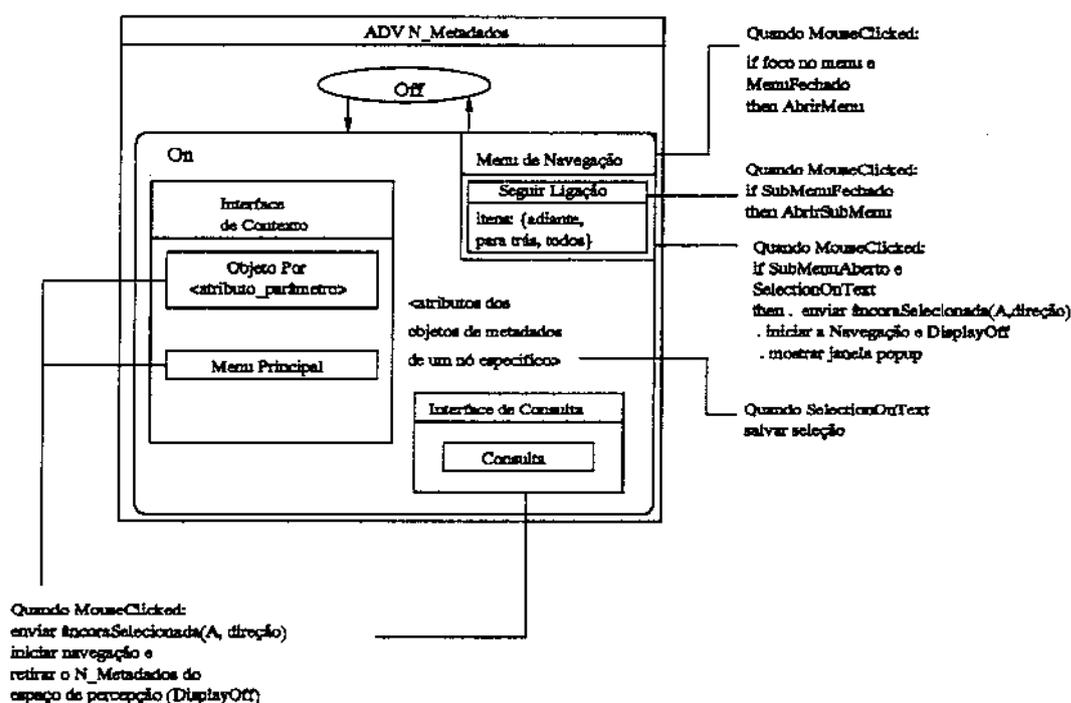


Figura 5.16: ADVChart para a ADV da Classe N_Metadados

5.3 Mapeamento para o Esquema do SGBD Hipermedia

O mapeamento da aplicação para o SGBD Hipermedia é efetuado com base nos diagramas construídos nas fases de Projeto de Navegação e de Projeto de Interface, usando o modelo hipermedia definido no capítulo anterior.

5.3.1 Estrutura das Classes de Navegação

A figura 5.17 apresenta a hierarquia das classes de navegação do SGBDH. Os nós das classes de navegação Classe Convencional e Geo-Classe são armazenados em coleções persistentes do banco de dados OO a fim de que possam ser utilizados como raízes de persistência para a construção de objetos virtuais no momento da criação dos contextos de navegação. Os nós da classe de navegação N_Metadados são virtuais e são criados

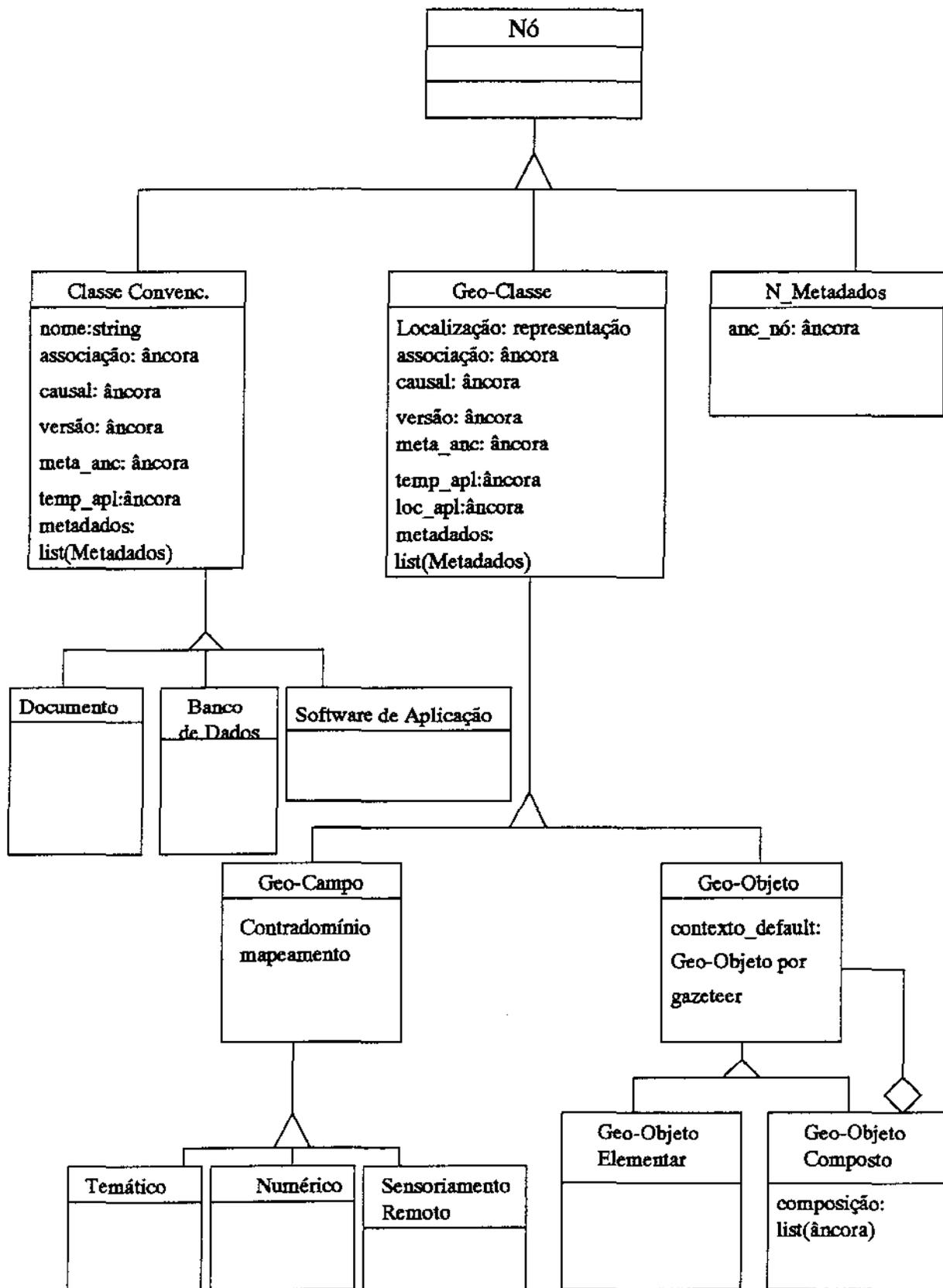


Figura 5.17: A Estrutura das Classes de Navegação mapeadas para o Esquema do SGBDH

durante a visita a cada contexto de navegação Metadados por Nó. Os nós são definidos através de consultas ao SGDBH que recuperam a lista dos objetos de Metadados de um nó. Os objetos de metadados são definidos no próximo capítulo.

5.3.2 Estrutura das Ligações e Ligações de Contexto

No Projeto Navegacional foram definidos cinco tipos de ligações entre objetos digitais: versionamento, associação, causalidade, agregação e metainformação.

A abordagem do modelo OOHD de Milet et al. mapeia cada possível relacionamento para um conjunto de instâncias de ligações contendo o nó origem e uma enumeração explícita dos possíveis nós destino. Uma consulta sobre um nó e suas ligações, que são armazenadas em coleções persistentes do BD, recupera todos os destinos de cada ligação que tem o nó como origem; o conjunto destes destinos forma o contexto *derivado de ligação* para o nó.

Esta forma de definir ligações e contextos derivados de ligações é demasiadamente rígida para suportar a dinâmica de uma biblioteca digital. A inclusão ou remoção de nós da biblioteca implica na definição ou na remoção manual de ligações, bem como em alterações explícitas dos destinos de várias ligações.

Para resolver estas limitações, propomos a definição de *tipos de ligações* no Projeto Navegacional. Um tipo de ligação determina um relacionamento semântico entre conjuntos de nós cujos elementos podem ser definidos dinamicamente através de computações ou de consultas ao BD. Os tipos são declarados em nossa biblioteca através de ligações de contexto dinâmicas, que especificam o comportamento navegacional de todas as instâncias daquele tipo.

Desta forma, um tipo de ligação da biblioteca é modelado por uma ligação com dois extremos dinâmicos: o extremo origem casa com a estrutura da âncora definida para aquele tipo de ligação; o extremo destino armazena uma chamada a um contexto de navegação ou o OID de um nó em algum contexto específico. Mais ainda, a natureza parametrizada dos contextos de navegação destinos destas ligações permite definir os destinos dinamicamente. As ligações de contexto para a navegação intracontexto, definidas pelas âncoras próximo e anterior das classes Informação Contextual, são construídas de forma semelhante, como explicitado no capítulo anterior.

5.3.3 Estrutura dos Contextos de Navegação

O mapeamento dos contextos de navegação da aplicação para o modelo do SGDBH utiliza as informações dos cartões definidos durante o Projeto de Navegação, que permitem especificar diretamente as visões correspondentes.

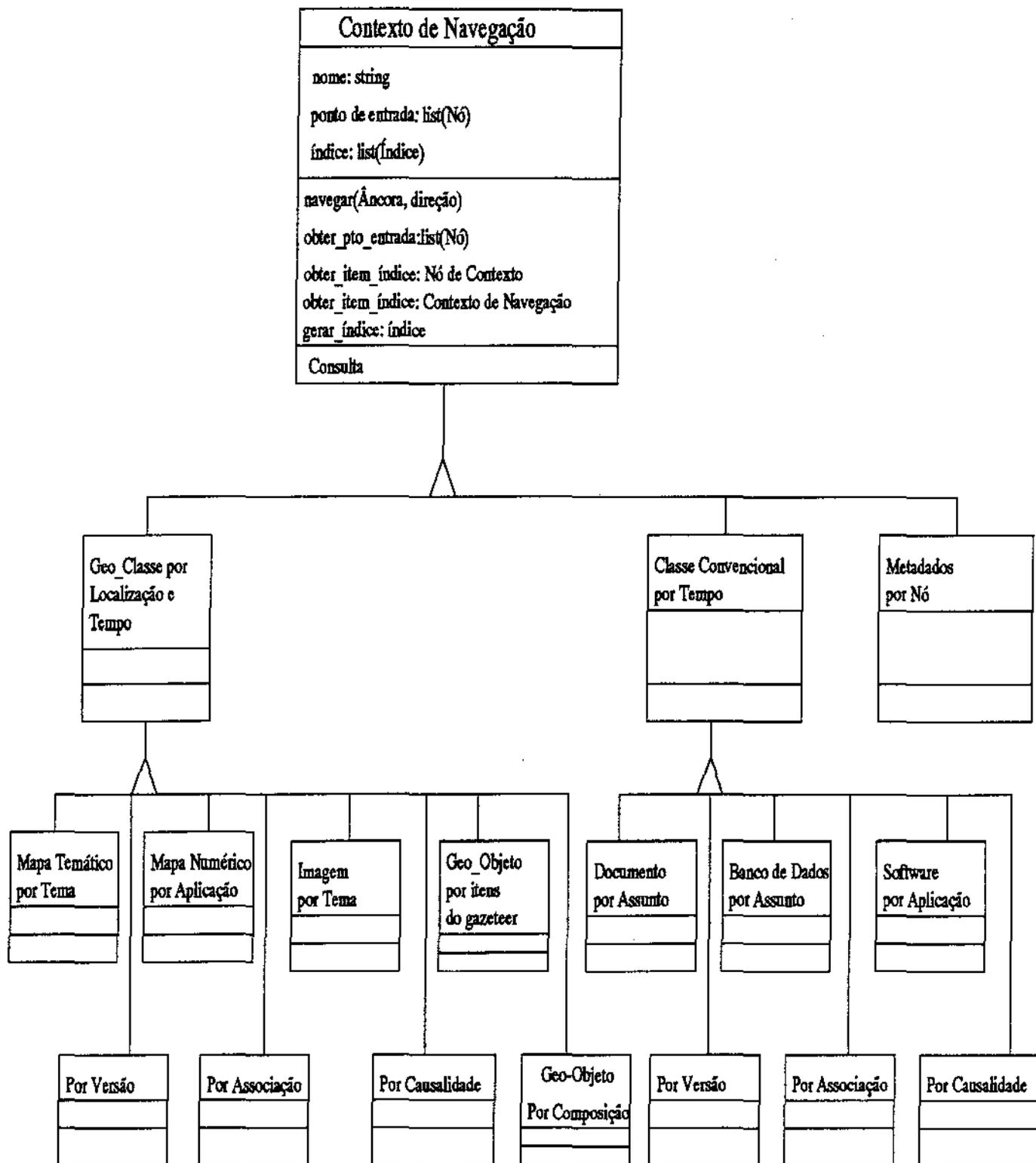


Figura 5.18: Hierarquia dos Contextos de Navegação da Biblioteca Digital

A figura 5.18 mostra a hierarquia dos contextos de navegação (visões parametrizadas), que reflete a organização do Esquema dos Contextos de Navegação no banco de dados. Cada classe nesta hierarquia encapsula um contexto de navegação definido por uma visão parametrizada. Os contextos estão organizados numa hierarquia definida a partir do contexto genérico *Contexto de Navegação*, particionando os nós da biblioteca em termos de Geo-Classes, de Classes Convencionais e de Metadados.

Cada contexto da classe *Contexto Metadados por Nó* contém um único objeto de navegação de *N_Metadados* referente a um dado nó da biblioteca. A hierarquia de especialização dos contextos *Geo-Classe por Localização e Tempo* e *Classe Convencional por Tempo* contempla não apenas contextos derivados de classes (por exemplo, *Geo-Objetos por itens do Gazetteer*), mas também contextos derivados de ligações e de agregado - que incluem, respectivamente, os contextos *Por versão*, *Por Associação* e *Por Causalidade*, e o contexto *Geo-Objetos Por Composição* para *Geo-Objetos*. Os primeiros são definidos tanto para *Geo-Classes*, quanto para *Classes Convencionais*.

A seguir, definimos a estrutura e as consultas para cada classe *Contexto* que aparece na figura.

1. Geo-Classe por Localização e Tempo

As extensões das subclasses do contexto *Geo-Classe por Localização e Tempo* estão limitadas aos nós da biblioteca que estão nos intervalos espaciais e temporais de busca e navegação. Estes intervalos são especificados pelo usuário e passados como parâmetros para a consulta da visão na raiz desta hierarquia. Esta consulta recebe ainda, como parâmetros, as funções de seleção espacial e temporal a serem aplicadas no predicado da consulta da visão.

A visão relativa ao contexto *Geo-Classe por Localização e Tempo* pode ser expressa na linguagem orientada a objetos de [AB91] como:

```
view Geo-Classe por Localização e Tempo(loc, temp, sel_esp(),sel_temp()) includes
select g
from g in Geo-Classe
where g.metadados.localização sel_esp() in loc and
g.metadados.tempo sel_temp() in temp
```

onde *loc* e *temp* representam os valores das janelas espaciais e temporais de busca definidas pelo usuário, *sel_esp()* e *sel_temp()* são funções de seleção espacial e temporal. Estas funções são implementadas em uma biblioteca de funções espaço-temporais

conectada ao SGBDH. Quando um dos valores não é definido pelo usuário, são utilizados valores default. Estes valores correspondem aos valores máximo de latitude e de longitude para o atributo de localização (loc) e ao maior intervalo temporal possível em relação ao parâmetro de tempo (temp).

2. Classe Convencional por Tempo

Os contextos de *Classe Convencional por Tempo* têm sua extensão restrita à janela temporal de busca definida para esta superclasse.

```
view Classe Convencional por Tempo(temp,sel_temp()) includes
select c
from c in Classe Convencional
where c.metadados tempo sel_temp() in temp
```

3. Metadados por Nó

A visão referente à classe Contexto *Metadados por Nó* é expressa como:

```
view Metadados por Nó(nó) herda_de N_Metadados includes
obj.metadados
```

onde o parâmetro *nó* especifica o nó da biblioteca ao qual os objetos de metadados se referem. Esta consulta recupera a lista dos objetos de metadados associada ao nó que foi passado como parâmetro.

4. Contextos Derivados de Classe

Cada um dos contextos derivados de classe referentes às Geo-Classes pode ser definido genericamente pela seguinte visão parametrizada:

```
view < Geo - Classe > por <Parâmetro> (classe, p) includes
select g
from g in Geo-Classe por Localização e Tempo
where g.metadados.Classe=classe and g.metadados.<atributo-parâmetro>=p
ordered by g.metadados.tempo
```

Em se tratando de Classes Convencionais, esses contextos são expressos da seguinte maneira:

```
visão < ClasseConvencional > por <Parâmetro> (classe, p) includes
select c
from c in Classe Convencional por Tempo
where c.metadados.Classe=classe and c.metadados.<atributo-parâmetro>=p
ordered by c.metadados.tempo
```

onde <Geo-Classe> e <Classe Convencional> representam genericamente a classe referente ao contexto (e.g., Mapa Temático ou Documento) e <Parâmetro> representa o atributo de particionamento do contexto (e.g., tema ou assunto). O parâmetro *classe* explicita a classe de particionamento e o parâmetro *p* explicita o atributo de particionamento dentro da classe concreta. Por exemplo, dois objetos dos Contexto Mapa Temático por Tema seriam definidos como MapaTemáticoPorTema (Mapa Temático, vegetação) e MapaTemáticoPorTema (Mapa Temático, hidrografia).

5. Contextos Derivados de Ligação

As consultas dos contextos *Por associação* e *Por causalidade* recuperam os objetos da biblioteca que têm relacionamentos de associação ou de causalidade com o objeto origem da navegação.

Para dar suporte às consultas dos contextos derivados de ligação, os metadados devem ser capazes de armazenar a informação sobre os relacionamentos. Como se verá no próximo capítulo, as classes de metadados capturam informações sobre relacionamentos em um atributo *Relacionamento*. Este atributo identifica os relacionamentos de um objeto com outros objetos. O atributo *Relacionamento* é constituído de dois subatributos: *Tipo*, que descreve qual a natureza do relacionamento (podendo assumir os valores *versionamento*, *causalidade* e *associação*), e um *Identificador* do objeto, que se refere ao relacionamento.

Os contextos derivados de ligação *Por associação* e *Por causalidade*, para Geo-Classes e para Classes Convencionais, podem ser expressos genericamente pela seguinte declaração de visão:

```
view <Classe> por <Relacionamento> (rel, nó_origem) includes
select g,c
from g in Geo-Classe por Localização e Tempo,
c in Classe Convencional por Tempo
where g.metadados.Relacionamento.tipo=rel and
```

```

g.metadados.Relacionamento.Identificador= no_origem and
c.metadados.Relacionamento.tipo=rel and
c.metadados.Relacionamento.Identificador= no_origem
ordered by c.metadado.tempo

```

onde <Classe> pode ser qualquer classe de navegação de Geo-Classe ou Classe Convencional, <Relacionamento> indica o tipo de relacionamento envolvido, os parâmetros *rel* e *nó_origem* indicam, respectivamente, o tipo de ligação que foi selecionada referente ao relacionamento envolvido, e o objeto que foi selecionado sobre o qual é definido o contexto.

A navegação através do contexto *por Versão* representa uma navegação através da história do nó. O ponto de entrada do contexto *por Versão* é definido como a versão mais atual (temporalmente) do objeto. Objetos de contextos *Por versão* são declaradas como visões parametrizadas da seguinte forma:

```

view <Geo-Classe Concreta> por Versão (nó_origem) includes
select g
from g in Geo-Classe por Localização e Tempo
where c.metadados.Relacionamento.tipo=versionamento and
c.metadados.Relacionamento.Identificador= nó_origem
ordered by c.metadado.tempo

```

```

view <Classe Convencional Concreta> por Versão (nó_origem) includes
select c
from c in Classe Convencional por Tempo
where c.metadados.Relacionamento.tipo=versionamento and
c.metadados.Relacionamento.Identificador= no_origem
ordered by c.metadado.tempo

```

5.3.4 Estrutura das Classes Informação Contextual

A figura 5.19 mostra a hierarquia das classes Informação Contextual construída a partir das especificações dos cartões correspondentes.

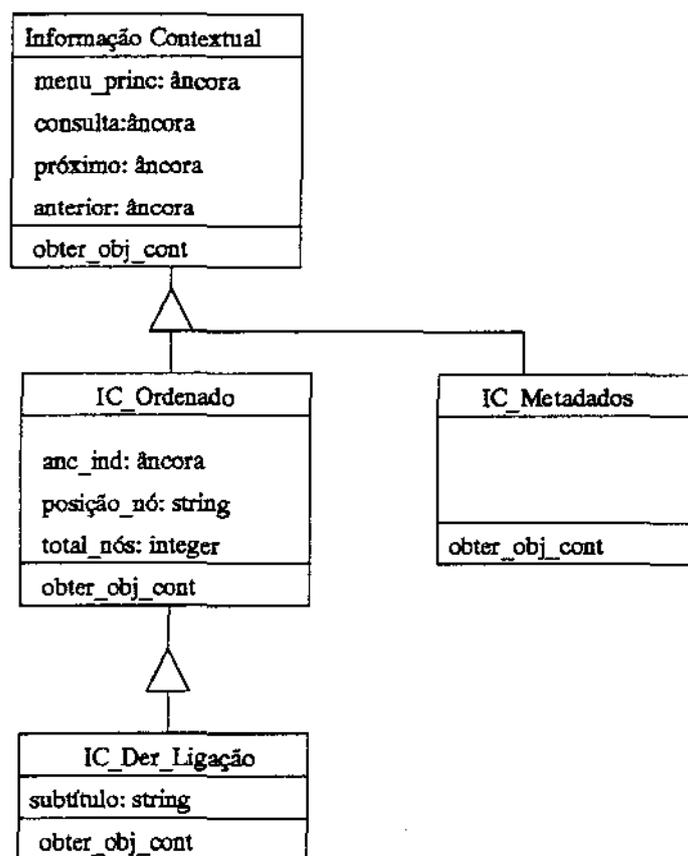


Figura 5.19: Hierarquia de Classes Informação Contextual da Biblioteca Digital Geográfica

5.3.5 Estrutura das ADVs

As ADVs são especificadas no SGBD Hipermídia a partir dos Diagramas de Configuração e dos ADVCharts. Elas são implementadas como especializações da classe abstrata ADV do nosso modelo, como apresentado na figura 5.20.

Cada uma das subclasses de ADV deve ser implementada com base nas hierarquias de composição e de especialização definidas nos respectivos Diagramas de Configuração. Os eventos, mensagens e ações especificados no projeto da Interface Abstrata são implementados através de métodos. O método *montar_tela* é definido e especializado para todas as

ADV e se encarrega de gerenciar os eventos sobre a ADV particular, trocando mensagens com a ADV genérica que utiliza o método *apresentar* para definir a apresentação do nó.

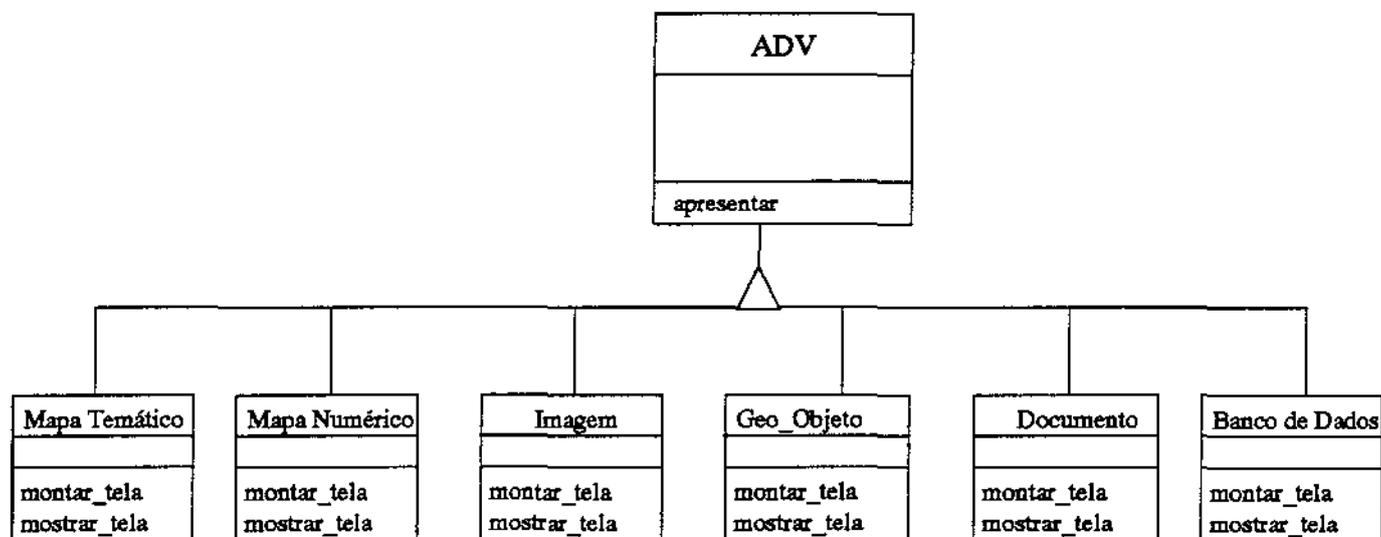


Figura 5.20: A Estrutura das ADVs da Biblioteca Digital Geográfica

A figura 5.21 apresenta a definição da classe *ADV_Imagem*. As demais ADVs podem ser definidas de maneira similar. A definição da *ADV_Imagem* é baseada no Diagrama de Configuração e no *ADVChart* genérico definido para as Geo-Classes, e utiliza diversas classes de interface tais como *Label*, *Image*, *Text* e *Button*. Estas classes já estão geralmente disponíveis em bibliotecas de interfaces de um SGBDOO.

A figura 5.21 apresenta a composição da *ADV_Imagem* com as ADVs Interface de Contexto, Interface de Consulta, Menu de Navegação, Imagem, e Descrição. As três primeiras ADVs são modeladas como conjuntos de botões referentes às âncoras de navegação. As duas últimas ADVs são relativas aos atributos Imagem e Descrição definidas na especificação da ADV. Os outros atributos da ADV capturados dos metadados são apresentados através de *Labels* para strings.

A figura 5.21 também mostra que a classe *ADV_Imagem* generaliza um conjunto de ADVs e especializa seu método *montar_tela* para as subclasses. As subclasses *ADV_Tema*, *ADV_Versão*, *ADV_Associação* e *ADV_Causal* são responsáveis por customizar a interface através de seu método especializado *montar_tela*, de acordo com os respectivos objetos de contexto a elas associados. Esta customização é efetuada através da inclusão de ADVs para a apresentação das informações do objeto de contexto, tais como as âncoras próximo e anterior, e através da exclusão de ADVs específicas, tais como o botão para o contexto em que já se está navegando.

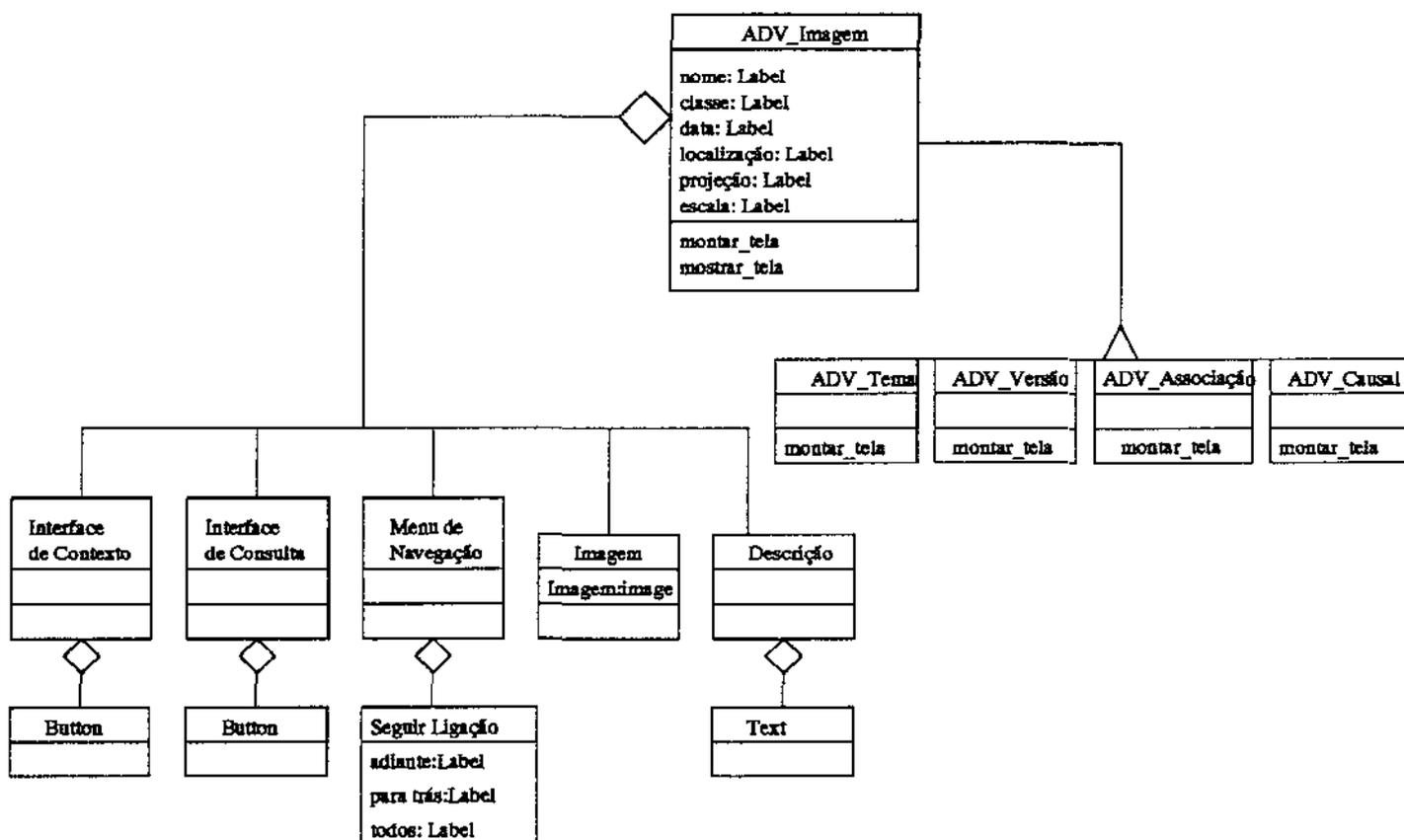


Figura 5.21: A Definição da classe ADV_Imagem

5.4 Resumo

Este capítulo apresentou a modelagem de uma aplicação hipermídia específica, utilizando o OOHDHDM modificado com as extensões propostas. Também mostrou o mapeamento do modelo da aplicação para o SGBD Hipermídia subjacente. A aplicação hipermídia modelada foi uma biblioteca geográfica. Como os contextos e hierarquias são definidos sobre a estrutura conceitual dos dados da biblioteca, as ligações incorporam a maior parte dos requisitos de usuários baseados na semântica da aplicação.

O fato de que um mesmo dado pode ser encontrado em vários contextos (reuso) evita o problema da partição dos domínios dos dados em partições estritas. Estas partições tornariam o acesso aos dados bem mais restrito. A possibilidade de mudar de contexto ou de hierarquia quando navegando, corresponde a uma alteração da perspectiva do usuário.

O uso de visões para definir contextos permite que estes contextos sejam computados dinamicamente e que definam partições do conjuntos de objetos da biblioteca. A geração da visão via consulta permite também a redefinição automática das janelas espaciais e temporais dos contextos com base em valores de área geográfica e de limites de tempo definidos pelo usuário. Com isto, o usuário pode reduzir o conjunto de dados para busca e pode restringir o espaço de navegação, de forma flexível. A redução de dados é suportada pelas características de catalogamento e de classificação dos metadados e pelas características de segmentação e de particionamento oferecidos pelos contextos. A redução dos dados torna consultas e buscas na biblioteca mais eficientes, bem como torna possível análises prévias dos dados armazenados.

Capítulo 6

O Ambiente de Catálogo da Biblioteca Digital Geográfica

6.1 Introdução

Uma biblioteca digital, sob uma perspectiva de ciência bibliotecária, deve ser encarada como uma extensão digital de uma biblioteca tradicional. Neste contexto, a biblioteca digital deve oferecer os mesmos serviços providos pela biblioteca tradicional, ao mesmo tempo que deve ampliá-los.

Levy e Marshall [LM95], por exemplo, definem o arcabouço de uma biblioteca tradicional através de um tripé constituído de *documentos*, *tecnologia* e *trabalho*. A partir desta caracterização, eles examinam as características de bibliotecas digitais em cada um dos três eixos, criticando idéias que norteiam o desenvolvimento dos atuais projetos de bibliotecas digitais. Na questão dos documentos, é condenada a idéia de que bibliotecas digitais contêm documentos fixos e permanentes. Em relação à segunda característica, os autores advogam que a tecnologia de futuras bibliotecas digitais comporte tanto material digital quanto não-digital, englobando assim uma larga faixa de materiais heterogêneos. Os autores censuram, desta forma, a idealização da “*morte do papel*”. Por fim, eles refutam o conceito de que bibliotecas digitais serão usadas apenas por indivíduos trabalhando isoladamente e defendem a idéia de que o trabalho de pesquisa em um ambiente de biblioteca é inerentemente *colaborativo*.

Já Nürnberg et al. [NFL⁺95] classificam os elementos de uma biblioteca tradicional em termos de *dados*, *metadados* e *processos*. A partir dessa taxonomia, analisam as características de bibliotecas digitais, que são definidas como sendo constituídas por *traduções digitais* dos elementos de uma biblioteca tradicional e de novos elementos para os quais não há uma analogia apropriada.

Ainda outra comparação, bastante completa, é encontrada em [Smi96]. Este trabalho

define que uma biblioteca tradicional é composta por três características organizacionais: 1) a organização da informação em objetos físicos tais como livros; 2) a organização física das coleções de acordo com vários atributos, tais como *assunto principal* e *autor*; e 3) um ambiente organizado que facilita o acesso direto a objetos físicos baseado nos atributos, bem como em um grau limitado de acesso indireto aos objetos. Este acesso indireto envolve múltiplas fontes de informação, tais como os bibliotecários, os catálogos e a própria forma como as coleções estão organizadas. O ambiente de uma biblioteca digital, em contrapartida, modifica esses três aspectos organizacionais. Em primeiro lugar, a organização da informação em objetos físicos é substituída por uma organização em *objetos lógicos*. Em segundo lugar, a organização física é substituída por múltiplas organizações lógicas (e.g., diversos contextos de navegação compartilhando um nó). Em terceiro lugar, uma biblioteca digital permite o uso de tecnologia digital na extração e na organização da informação contida nos objetos. A informação extraída deve satisfazer as necessidades de informação do usuário ou pode ser empregada por “bibliotecários digitais” na caracterização dos objetos. Esta metainformação sobre os objetos é utilizada para prover acesso à informação armazenada. Dentre as desvantagens do ambiente digital, cita-se a perda de interações existentes entre bibliotecários e usuários que ocorrem no ambiente de uma biblioteca tradicional.

Todos estes estudos ressaltam o papel do catálogo, que é ainda mais relevante em uma biblioteca digital. O capítulo anterior mostrou como definir uma aplicação de biblioteca digital usando o modelo hipermídia proposto. Este capítulo descreve o ambiente do catálogo eletrônico no contexto de nossa biblioteca digital geográfica. A seção 6.2 discute os papéis de um catálogo em uma biblioteca digital. A seção 6.3 define a arquitetura do catálogo. A seção 6.4 apresenta a hierarquia de metadados para o catálogo da biblioteca digital geográfica. A seção 6.5 descreve o processo de consultas sobre a biblioteca digital geográfica usando a arquitetura de metadados descrita nas seções 6.3 e 6.4. A seção 6.5 faz comparação com trabalhos correlatos. A seção 6.6 resume o capítulo.

6.2 O Catálogo de uma Biblioteca Digital

Dividiremos o ambiente do catálogo de uma biblioteca digital em termos de seus aspectos estáticos e dinâmicos. Os aspectos estáticos dizem respeito à capacidade de modelar os documentos. Os aspectos dinâmicos estão relacionados aos serviços que o catálogo deve prover.

Modelagem de Documentos

A modelagem dos documentos do catálogo de uma biblioteca digital está geralmente associada ao termo *metadados* ou *metainformação*. Contudo, estes termos possuem uma abrangência muito grande, sendo utilizados nos mais diversos contextos e com uma variedade considerável de significados. Outro problema é que uma partição estrita e estática do que são considerados dados e metadados é geralmente errônea [LLRD96]. Diversas definições genéricas são fornecidas a dados e metadados. Segundo Smith [SGG96], a pesquisa em metadados tem por objetivo descobrir métodos apropriados para a modelagem de várias classes de documentos. O autor utiliza o termo “*metainformação*” para definir o modelo de um objeto em uma biblioteca digital. Uma definição precisa, no contexto de bibliotecas digitais, é dada também por Smith em [SGG96], quando define metadados como “*informação armazenada em catálogos eletrônicos de bibliotecas digitais que modelam documento e consultas de usuários e cujo objetivo é suportar o acesso eficiente do usuário aos documentos de uma biblioteca digital*”.

Serviços

O catálogo eletrônico provê um conjunto de serviços à biblioteca digital. Estes serviços têm por objetivo primário permitir aos usuários acessarem a informação armazenada na biblioteca digital com eficiência. Desta forma, o catálogo digital estende o sentido comum de catálogos de bibliotecas tradicionais, absorvendo os papéis do bibliotecário e da organização física dos livros da biblioteca. São serviços que devem ser oferecidos pelo catálogo de uma biblioteca digital [Smi96]: 1) coordenação das interações do usuário com o ambiente da biblioteca; 2) construção do modelo cognitivo do usuário, de sua consulta e de seu espaço de trabalho; 3) provisão de acesso aos documentos; 4) realização de casamentos entre as consultas do usuário e as propriedades dos documentos armazenados; 5) criação de modelos de documentos em termos de metadados.

6.3 Arquitetura do Catálogo

[SGG96] apresenta uma arquitetura geral de quatro componentes para o catálogo de uma biblioteca digital. Esta arquitetura genérica inclui:

1. um componente de *modelagem de documentos* no qual modelos de documentos da biblioteca são construídos e mantidos pelos “bibliotecários” ou pelo sistema;
2. um componente de *modelagem de consulta* no qual o sistema interage com o usuário para construir um modelo de sua consulta;

3. um componente de *casamento de consulta* no qual o sistema interativamente casa o modelo da consulta do usuário com os modelos de documentos, selecionando a informação apropriada como resposta para o usuário;
4. um componente de *interoperabilidade de catálogo* no qual o sistema transforma consultas para a comunicação de/para outros catálogos.

Este modelo genérico será utilizado para definir a arquitetura de nosso catálogo, apresentada na figura 6.1.

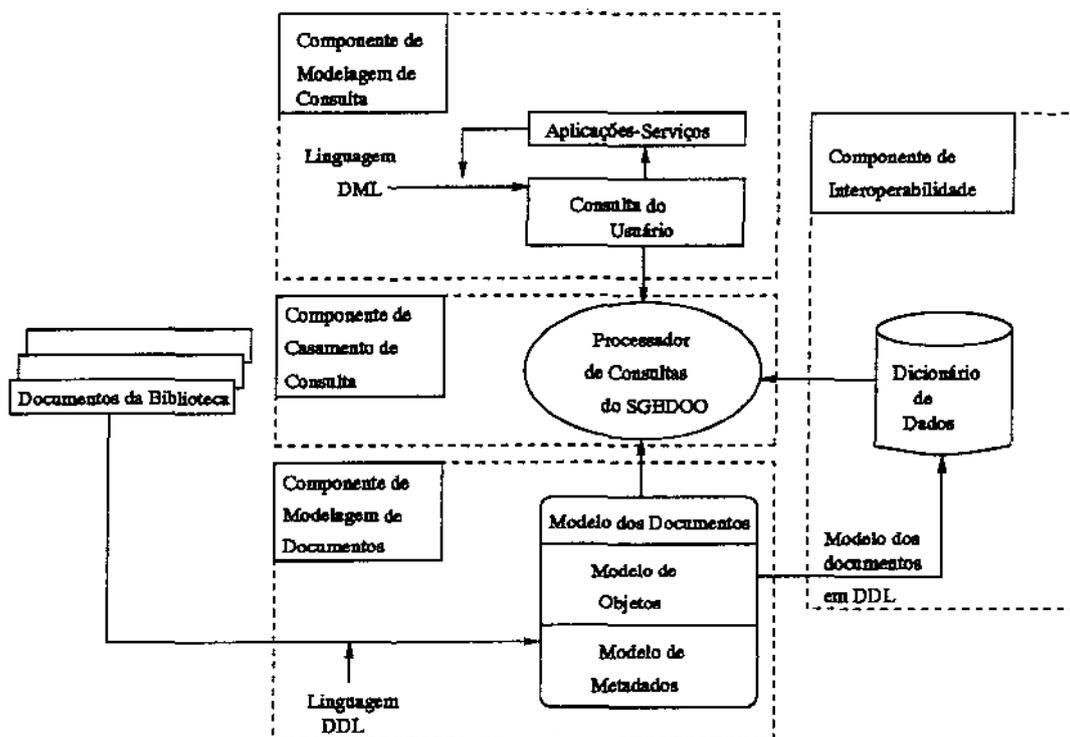


Figura 6.1: A Arquitetura do Catálogo Digital

O componente de Modelagem de Documentos inclui a Linguagem de Definição de Dados (DDL) do SGBDOO, a qual é utilizada para descrever o esquema das classes de documentos da biblioteca. O modelo de documentos pode ser dividido logicamente em dois componentes: a) o modelo dos objetos e b) o modelo de metadados. O primeiro foi descrito no capítulo 4. Este capítulo descreve o segundo modelo.

O componente de Modelagem de Consulta é formado pelas aplicações-serviço da biblioteca que são: o *Visualizador de Mapas* e o *Gazetteer*, para consultas espaciais; o *Formulário de Tempo*, para consultas temporais; e o *Formulário de Consulta*, para consultas sobre os modelos dos documentos. As aplicações-serviço interagem com o usuário através de interfaces baseadas em mapas e em formulários. As entradas do usuário são

traduzidas para a Linguagem de Manipulação de Dados (DML) do SGBDH. Os dados de resposta são retornados e apresentados pelas aplicações-serviço.

O componente de Casamento de Consulta é provido pelo processador de consultas do SGBDH.

O componente de Interoperabilidade é constituído pelo dicionário de dados do SGBDH, que descreve o esquema do banco de dados. Esta abordagem permite ao usuário obter documentação e informações sobre os valores de tipos e sobre as semânticas das classes de objetos e atributos específicos. Esta característica facilita o acesso do usuário e simplifica a tarefa de bibliotecários novatos na inclusão e no catalogamento dos dados. Uma das principais funções do componente de interoperabilidade é facilitar a construção de serviços que traduzem atributos entre diferentes modelos [BCGP97b, BCGP97a]. Estes serviços permitem que aplicações clientes que não entendem o modelo nem a linguagem da biblioteca possam comunicar-se com ela.

Os serviços providos pelo catálogo (via SGBDH e via aplicações-serviço) são mostrados na figura 6.2. Estes serviços são:

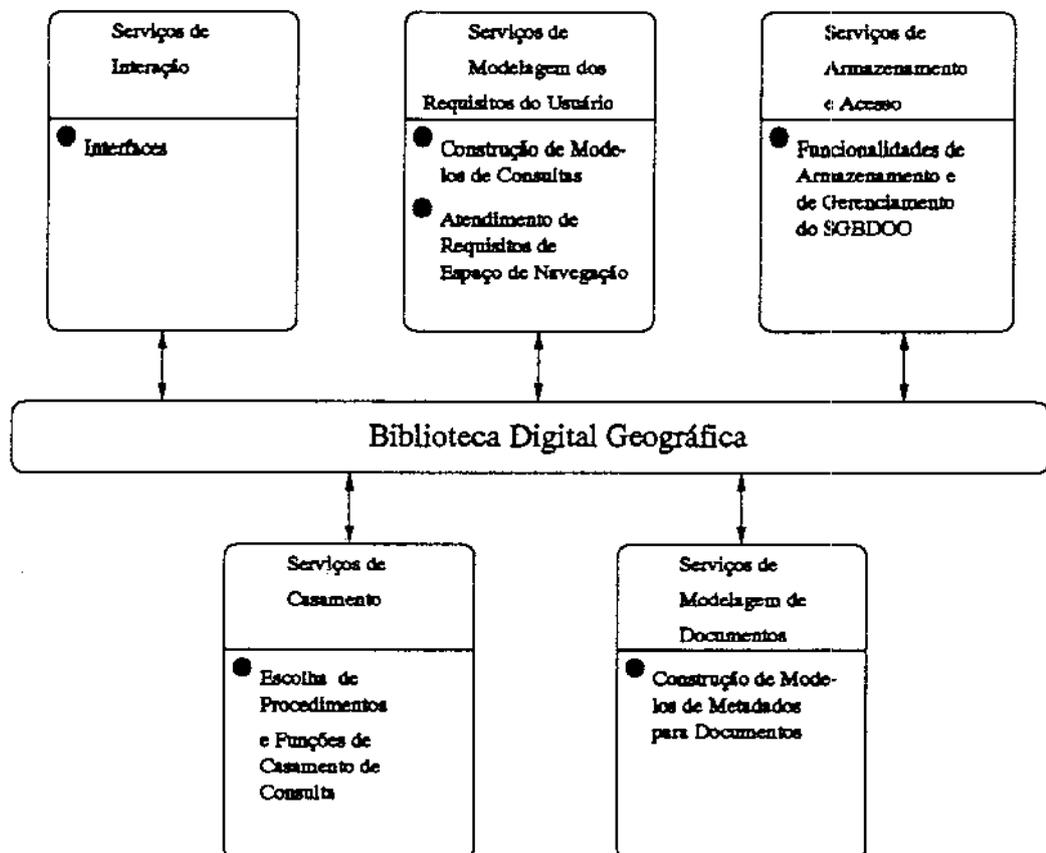


Figura 6.2: Os Serviços do Catálogo

- Serviço de Interação

Este serviço coordena a interação entre o usuário e o catálogo. O serviço inclui os aspectos de interface das aplicações-serviço.

- Serviço de Modelagem dos Requisitos do Usuário

Este serviço é empregado na modelagem das consultas feitas pelo usuário e no atendimento dos requisitos de espaço de navegação deste usuário. O serviço é suportado pelas aplicações-serviço da biblioteca que modelam e traduzem as consultas e que determinam o espaço de navegação do usuário através destas consultas.

- Serviço de Armazenamento e Acesso

Este serviço sustenta o armazenamento e o acesso aos objetos disponíveis nas coleções da biblioteca, explorando as funcionalidades de armazenamento e o gerenciamento do SGBDOO subjacente à biblioteca.

- Serviços de Casamento

Este serviço suporta a escolha e a aplicação de procedimentos de casamento apropriados nas consultas do usuário. O serviço é acomodado em nossa biblioteca pela possibilidade do usuário escolher, dentre várias funções espaciais e temporais disponíveis, aquela que mais se adequa às necessidades de sua consulta.

- Serviço de Modelagem de Documentos

Este serviço provê suporte aos bibliotecários para a criação de modelos dos objetos da biblioteca em termos de metadados. Este serviço é acomodado pelas primitivas de modelagem fornecidas pelo modelo hipermídia genérico descrito no capítulo 4.

6.4 A Hierarquia de Classes de Metadados da Biblioteca Digital Geográfica

Como dito anteriormente, o modelo de um documento da biblioteca é dividido conceitualmente no modelo do objeto e no modelo de metadados. O modelo de objetos inclui os aspectos conceituais, de navegação e de interface dos objetos e foi discutido em detalhes no capítulo 4. Esta seção trata da hierarquia de classes de metadados da Biblioteca Digital Geográfica, que permite descrever o modelo de metadados dos documentos da biblioteca.

Os metadados são modelados como tuplas que abstraem termos, definições e valores, os quais descrevem os dados da biblioteca. Estes metadados permitem aos usuários encontrarem e acessarem os dados, possibilitam determinar a adequação e o melhor uso para

estes dados. Além disso, os metadados beneficiam a organização e a produção dos dados e modelam a informação necessária para assegurar o manuseio correto destes dados, sendo assim de fundamental importância para as agências produtoras de dados.

Propomos os seguintes tipos de metadados:

- Metadados Descritivos

Descrevem propriedades convencionais dos objetos da biblioteca que auxiliam nas buscas e facilitam o acesso ao objeto.

- Metadados Espaciais

Descrevem propriedades espaciais do objeto, tais como seu MBR e sua escala, e são utilizados nas consultas espaciais da biblioteca.

- Metadados Temporais

Descrevem propriedades temporais dos objetos como os intervalos de validade.

- Metadados Administrativos

Descrevem propriedades relacionadas ao gerenciamento de um objeto na biblioteca. Exemplos incluem a data da última revisão e a identidade do administrador.

- Metadados Particulares

Descrevem propriedades próprias e particulares para certas classes de objetos não cobertas pelos conjuntos anteriores. Exemplos incluem a descrição do esquema para BancodeDados e o número e tipos de bandas para Imagens de Satélite [AS94].

- Metadados de Referência Indiretas

Descrevem referências a dados relacionados.

A hierarquia de classes de metadados atende o seguinte conjunto de requisitos:

- Acomodação de vários padrões existentes de metadados

A utilização de padrões promove interoperabilidade; permite ao provedor de informação se beneficiar de padrões existentes para descrição de dados e recursos; facilita a obtenção de consenso nas descrições dos dados usando apenas atributos, palavras-chave e valores [HB96]; auxilia na tarefa de catalogamento do dado.

É consenso na comunidade de pesquisa que não existe um único conjunto ou padrão de metadados capaz de cobrir todos os aspectos necessários, principalmente quando considerando campos de conhecimentos específicos. Assim, as classes de nossa hierarquia são definidas com base em padrões conhecidos. Cada classe da hierarquia

acomoda um único padrão - quer existente, quer estendido ou modificado - para atender necessidades específicas. Objetos das classes de metadados são integrados nas listas de metadados dos nós.

- Extensibilidade

A extensibilidade é essencial para qualquer modelo de metadados, pois novos conjuntos de metadados aparecerão com o desenvolvimento de infra-estruturas de redes [LLRD96]. Mais ainda, diferentes comunidades irão propor e projetar novos e diferentes tipos de metadados que representarão os interesses e o domínio de conhecimento da comunidade. Cada comunidade deve ser capaz de independentemente criar e manter os metadados que recaem dentro de sua esfera de influência.

A extensibilidade é alcançada graças ao paradigma OO, que possibilita a inclusão de novas classes de metadados na hierarquia.

- Novos papéis para metadados

Em nossa biblioteca, os metadados possuem o papel tradicional de prover organização e classificação do conteúdo da biblioteca. Adicionalmente, alguns dos atributos das classes de metadados são responsáveis pelo papel de dar suporte à navegação hipermídia e ao folheamento da biblioteca de maneira inovadora. Esta característica se faz presente na definição das consultas dos contextos de navegação, nas consultas das ligações dinâmicas que utilizam informações dos metadados, bem como na definição de itens de índices hierárquicos.

- Integração com o ambiente da Biblioteca Digital Geográfica

A integração é alcançada definindo-se as classes de metadados dentro do modelo hipermídia, utilizando a classe Metadados como a raiz da hierarquia das classes de metadados.

A hierarquia de classes de metadados apresentada nesta seção é focada na descrição de objetos da biblioteca geográfica. A hierarquia tem um escopo bem definido, visando uso em um ambiente de biblioteca digital de “alta integridade”. Neste ambiente, os administradores e “bibliotecários” são os responsáveis por adicionar as classes de metadados cabíveis e por gerenciar dados e respectivos metadados. A hierarquia de classes de metadados é apresentada na figura 6.3.

A raiz da hierarquia de classes é a classe Metadados do modelo hipermídia genérico. As subclasses de Metadados são todas constituídas por conjuntos de atributos organizados segundo a taxonomia apresentada. Os domínios destes atributos serão explicitados mais adiante.

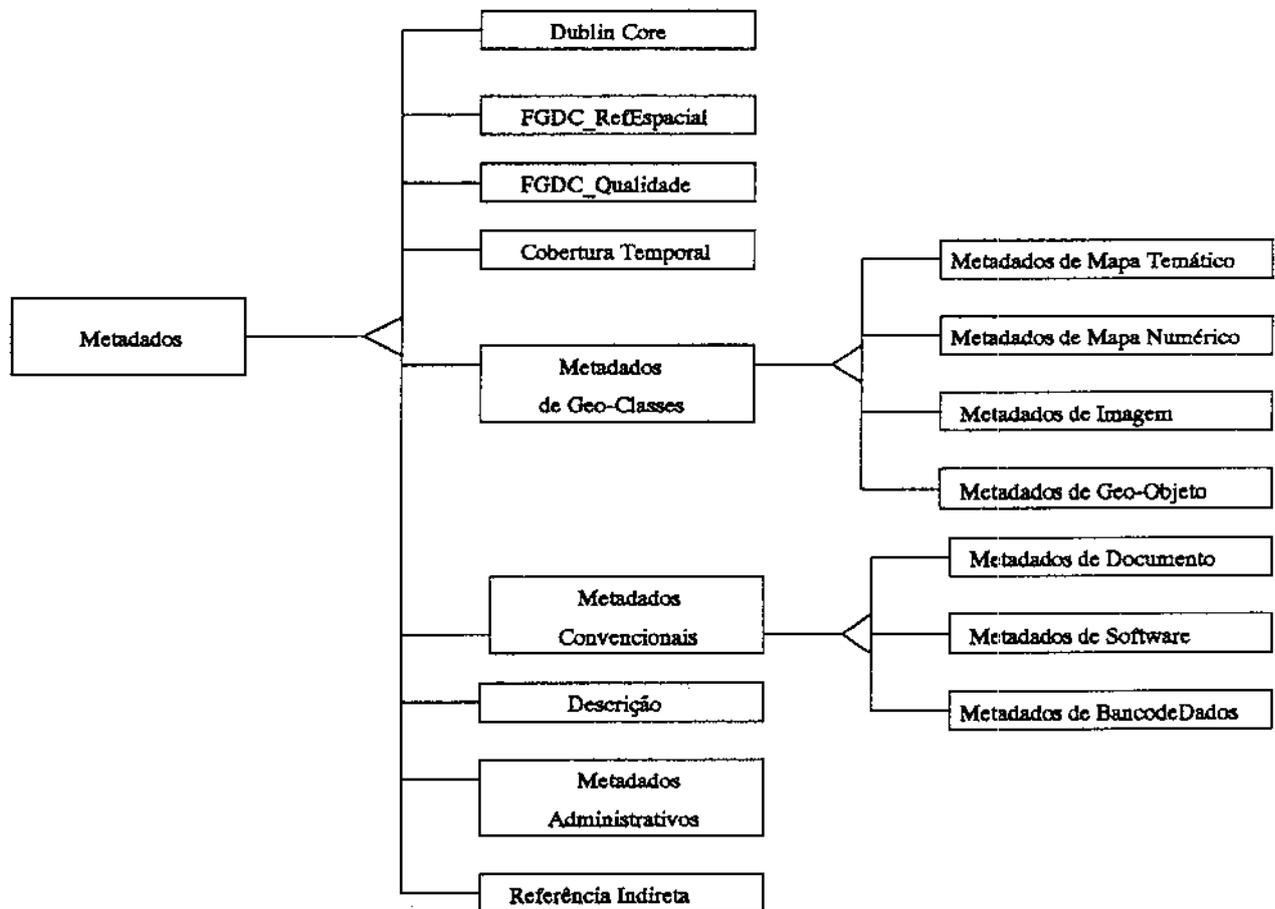


Figura 6.3: A Hierarquia de Classes de Metadados

Lembramos que cada nó da biblioteca tem uma lista de metadados. A única classe de metadados obrigatória dentro desta lista é a classe *Dublin Core*. As demais, opcionais, são definidas com três objetivos básicos: (1) descrever as características particulares das classes de dados da biblioteca; (2) suportar o leque de consultas projetado para a biblioteca; (3) suportar as operações de folheamento e de particionamento lógico das instâncias da biblioteca. Dublin Core e FGDC, por exemplo, são padrões internacionais para, respectivamente, dados convencionais em rede e dados espaciais. Como a hierarquia é implementada sobre o SGBDH conjuntamente com o modelo de objetos, o esquema dos metadados é armazenado no dicionário de dados do BD, permitindo assim consultas sobre informações relativas às classes e aos atributos de metadados.

Descreveremos, a seguir, cada uma das classes da hierarquia com maiores detalhes:

6.4.1 Classe Dublin Core

O Dublin Metadata Core Element Set é resultado de um *workshop* em metadados, visando definir padrões para metadados sobre a rede. O conjunto de metadados proposto tem por objetivo prover um conjunto minimal de atributos descritivos que facilitam a descrição, a descoberta e a indexação automática de documentos eletrônicos em rede.

A tabela 6.1 mostra os atributos do Dublin Core e seu significado *default*. A semântica destes atributos pode ser modificada de duas formas: 1) pelo uso de qualificadores emprestados por outros esquemas de metadados, os quais permitem designar significados mais detalhados e precisos para os atributos do Core; ou 2) por especializações e extensões *ad-hoc* que refinam os atributos.

O Dublin Core possui diversas limitações e defeitos. Em primeiro lugar, o Dublin Core propositadamente não impõe quaisquer amarrações sintáticas ou definições formais aos seus atributos, limitando-se apenas a definir sua semântica. A interoperabilidade entre sistemas requer uma descrição mais concreta de sintaxe e de semântica do que a provida pela definição inicial do Dublin Core. Em segundo lugar, a especificação não oferece nenhuma orientação para projetistas e implementadores de serviços de busca na rede. Mais ainda, com semântica muito vaga, sem vocabulário controlado, e sem definição de uma sintaxe apropriada para os atributos, é improvável que a informação provida possa ser processada algoritmicamente.

Em terceiro lugar, o Dublin Core representa uma predominância de catalogamento descritivo [LLRD96]. Apesar da importância deste tipo de metadado, ele é apenas um dos conjuntos de metadados que podem ser associados a objetos em rede. Outros tipos de metadados incluem: (1) metadados para *termos e condições*, que descrevem regras para o uso e acesso ao objeto; (2) *metadados administrativos*, relacionados ao gerenciamento de um objeto; (3) metadados para *classificação de conteúdo*, que provêem descrições dos atributos de um objeto dentro de um esquema de classificação multidimensional; (4)

Atributo	Descrição
Assunto	o tópico endereçado pelo objeto
Título	o nome do objeto
Autor	a pessoa que é a principal responsável pelo conteúdo intelectual do objeto
Publicador	o agente, a agência ou a instituição responsável por tornar o objeto disponível
Outro agente	as pessoas tais como editores e transcritores que têm alguma contribuição intelectual significativa no objeto
Data	a data de publicação
Tipo do Objeto	o gênero do objeto
Forma	a representação do objeto tal como arquivo <i>postscript</i> ou arquivo executável Windows
Identificador	o string ou o número usado para identificar unicamente o objeto
Relação	os relacionamentos com outros objetos
Origem	os objetos, digitais ou não, a partir do qual o objeto é derivado
Linguagem	a linguagem do conteúdo intelectual
Cobertura	a característica de localização espacial ou de duração temporal do objeto.

Tabela 6.1: Os atributos do Dublin Core

metadados de *proveniência*, que definem a fonte ou a origem do objeto como, por exemplo, a descrição do dispositivo usado para gerar uma imagem; (5) metadados *estruturais*, que definem componentes lógicos de objetos complexos ou compostos e que indicam como acessar estes componentes.

Mesmo se concentrando em atributos genéricos para catalogamento descritivo, o Dublin Core inclui alguns elementos específicos de domínio como, por exemplo, o elemento *cobertura*, que é específico para dados espaciais e temporais; o elemento *origem*, que é significativo apenas para objetos que não se originaram em uma forma digital; e o elemento *relacionamento*, que é inerentemente dependente de domínio, apresentando semântica confusa.

Em nossa biblioteca, o escopo bem definido da hierarquia de classes de metadados e as características do ambiente da biblioteca permitem utilizar um vocabulário controlado e uma sintaxe regular para os elementos de metadados do Dublin Core e demais atributos. Estas vantagens não podem ser alcançadas com o uso do Dublin Core em ambientes não-estruturados, devido à sua fraca especificação e à ausência de definição de sintaxe. Esta seção descreve as características de nossa classe Dublin Core, apresentada na figura 6.4.

1. Assunto

O *Assunto* é o campo do conhecimento ao qual o objeto pertence. O conjunto de possíveis valores deste atributo envolve os nomes das classes da biblioteca e uma lista restrita de nomes de subdomínios.

Dublin Core
Assunto
Título
Autor
Publicador
Data
Tipo
Forma
Identificador
Relação:
list(tuple(Tipo, Identificador))

Figura 6.4: A Classe Dublin Core

2. Título

É o nome do objeto. Este atributo tem um significado mais claro nas Classes Convencionais. Em Geo-Classes, este atributo é geralmente definido através de uma frase descritiva. O Título pode ser utilizado em consultas por palavra-chave, suportadas em máquinas de busca que indexam strings de texto (e.g., WAIS).

3. Autor

É a pessoa, o agente ou o dispositivo primariamente responsável pelo conteúdo intelectual do trabalho.

4. Publicador

É definido como o agente ou a agência responsável por tornar o objeto digital disponível.

5. Data

A data de publicação. Reflete a data em que o objeto tornou-se disponível na biblioteca. Corresponde ao tempo de transação do objeto digital e pode ser preenchido diretamente pelo sistema.

6. Tipo

É definido como a categoria abstrata ou o gênero do objeto, tal como novela, poema, dicionário, software executável, código fonte, arquivo de dados, áudio, vídeo ou qualquer categoria que se julgue útil para a descoberta e recuperação dos recursos sendo descritos.

7. Forma

É definida como a representação do objeto e tem por objetivo prover informação ao usuário sobre os recursos de hardware e de software necessários para mostrar ou operar o objeto. Por exemplo, consultas à biblioteca podem utilizar este atributo como um filtro para que o sistema retorne apenas dados que o usuário possa utilizar em seu ambiente local. Exemplos de valores para este atributo podem incluir documentos Postscript (.PS) ou Adobe Portable Document Format (.PDF), arquivos executável Windows95 ou UNIX (.EXE), arquivos HTML (.HTML).

8. Identificador

O identificador é utilizado para identificar unicamente o objeto. Os diferentes tipos de dados da biblioteca podem possuir diferentes tipos de identificadores, tais como ISBN, URL ou identificadores não-públicos, tais como número de relatórios técnico ou o oid interno no BD.

9. Relação

Este atributo identifica relacionamentos de um objeto com outros objetos da biblioteca. O atributo Relação é uma lista de registros formados de dois atributos: *Tipo*, que descreve a semântica do relacionamento; e *Identificador*, que especifica o item relacionado. Em nossa biblioteca, o atributo tipo pode assumir o seguinte conjunto de valores correspondentes aos relacionamentos da biblioteca: *associação*, *causalidade*, *versionamento*. Os valores para o atributo Identificador obedecem às mesmas regras do metadado Identificador.

Este metadado é a principal fonte de informação para a geração dos *contextos derivados de ligação* da biblioteca digital. As consultas destes contextos particionam os objetos da biblioteca em objetos que compartilham um mesmo relacionamento.

Os atributos *Outro Agente* e *Origem* do Dublin Core não foram considerados para a biblioteca geográfica. O atributo *Cobertura* é específico para dados espaciais e temporais. Devido à sua especificidade e devido à ausência de sintaxe bem definida no Dublin Core, este atributo foi retirado do conjunto central e passou a ser modelado separadamente por outras classes.

6.4.2 FGDC_RefEspacial

Esta classe é inspirada no padrão *FGDC Contents Standards for Digital GeoSpatial Metadata* [Com], que tem por objetivo estabelecer uma maneira uniforme de documentar e de descrever dados geográficos [GV97]. Neste sentido, esta classe é especialmente útil para modelagem de aspectos dos objetos da biblioteca derivados de Geo-Classe.

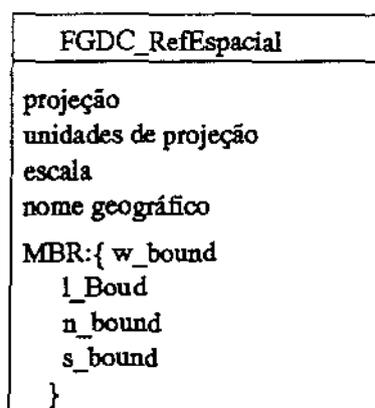


Figura 6.5: Classe FGDC_RefEspacial

Esta classe é responsável por determinar a localização geográfica do dado sendo descrito. Os atributos de FGDC_RefEspacial, descritos na figura 6.5, respondem consultas de usuários relativas à cobertura espacial dos dados. Estas consultas especificam a região de busca para consultas “*O que está aqui*” e recuperam a informação destes metadados como resposta em consultas “*Onde está isso?*” (vide capítulo 2). A Referência Espacial é também utilizada na consulta do Contexto *Geo-Classe por Localização e Tempo* para determinação do espaço de navegação da biblioteca.

Os domínios dos atributos são definidos segundo as especificações do FGDC. As coordenadas do MBR são descritas em coordenadas cartesianas com frações decimais de graus, usando valores negativos para longitudes no Hemisfério Oeste e para latitudes na Hemisfério Sul. As unidades de projeção explicitam os atributos calculados da projeção. O nome geográfico é um string que pode ser usado pelo usuário como uma referência implícita à localização do dado [FFLS96]. Para maiores detalhes, veja [Com].

6.4.3 FGDC_Qualidade

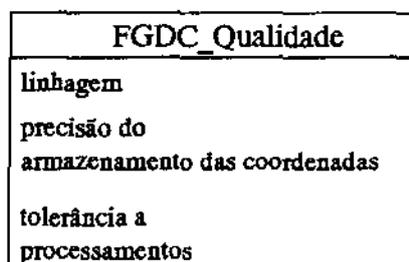


Figura 6.6: Classe FGDC_Qualidade

A classe *FGDC_Qualidade* é também baseada no Padrão FGDC e provê a base para assegurar a qualidade de um dado para uma aplicação. A *linhagem* descreve as origens dos dados espaciais, as transformações de coordenadas e métodos de derivação utilizados sobre os dados. A *precisão do armazenamento da coordenadas* se preocupa com quão precisamente as coordenadas do MBR representam as localizações verdadeiras. As *tolerâncias a processamento* incluem tolerância a incertezas armazenadas para os dados.

Estes metadados podem servir como filtro de qualidade para os dados retornados ao usuário, por consultas que especificam tais requisitos de qualidade.

6.4.4 Cobertura Temporal

Esta subclasse representa *os tempos de validade* para os dados da biblioteca e é opcional para todas as classes. Sua definição é inspirada no FGDC. A estrutura da classe Cobertura Temporal é mostrada na figura 6.7. A especificação do período de tempo de uma Cobertura Temporal pode ser efetuada através de um dos dois atributos: *Tempo/Dia*, e *Intervalo_Tempo/Dia*.

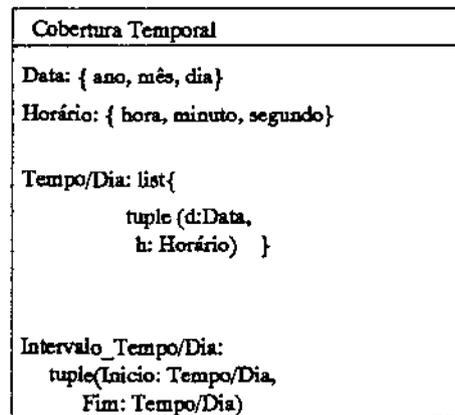


Figura 6.7: Classe Cobertura Temporal

Isto é feito especificando a *Data* e o *Horário*. As convenções para Datas e Horários são explicitadas do documento do Padrão FGDC.

Esta é a classe que dá suporte às consultas temporais dos usuários em relação aos tempos relacionados ao conteúdo dos objetos da biblioteca. Cobertura Temporal é também utilizada pelas consultas dos contextos *Geo-Classe por Localização e Tempo* e *Classe Convencional por Tempo* na determinação e restrição do espaço de navegação do usuário. Este espaço de navegação é definido como os objetos de navegação retornados pelas consultas temporais dos dois contextos.

6.4.5 Metadados de Geo-Classes

A classe metadados de Geo-Classes modela informações específicas e particulares para cada tipo de Geo-Classe. Ela armazena informações tais como número e tipos de bandas, número de linhas e *samples* de cada banda, para Imagens; descrição dos intervalos de valores para o contradomínio de Mapas Temáticos e Numéricos, e decisões relativas ao mapeamento dos intervalos.

6.4.6 Metadados Convencionais

A classe Metadados Convencionais modela informações que são específicas para as subclasses de Classe Convencional. Ela é especializada em *Metadados de Documento*, em *Metadados de Software* e em *Metadados de BancodeDados*. Dentre as informações específicas modeladas para cada uma destas subclasses, cita-se: plataforma, recursos de hardware mínimos, versão, para Metadados de Software; definição do esquema, tipo de acesso (consulta, inserção, alteração e remoção) e tipo de login, para Metadados De BancodeDados.

6.4.7 Metadado Descrição

A Descrição é uma descrição textual sobre o dado. Pode incluir, por exemplo, o propósito ou o uso pretendido para o dado. A descrição contém também um conjunto de *palavras-chave* para serem utilizadas por ferramentas de indexação de texto.

6.4.8 Metadados Administrativos

Estes metadados estão relacionados ao gerenciamento do objeto. Um objeto da classe Metadado Administrativo contém a *identidade do administrador*; um *ponto de contato*, que identifica um indivíduo que pode ser contatado sobre o dado; uma *instrução de contato*, que determina a forma de contato, por exemplo, e-mail ou número de telefone; e a *data (e horário) da última modificação* no dado, descrita nos mesmos moldes da Cobertura Temporal.

6.4.9 Referência Indireta

A Referência Indireta amarra referências indiretas a outros objetos que podem prover informação adicional sobre o objeto que está sendo descrito. A Referência Indireta pode referenciar outros repositórios de metadados relacionados ou pode conter identificadores persistentes únicos para outros recursos e objetos.

6.5 Modelo de Consulta da Biblioteca Digital Geográfica

O modelo de consulta da Biblioteca Digital Geográfica é caracterizado por uma integração simbiótica entre navegação hipermídia - baseada em folheamento e em estruturas hierárquicas - com consultas focadas em conteúdo e baseadas em metadados.

Para descrever o modelo de consultas, retomaremos o Esquema de Contextos de Navegação da biblioteca digital, apresentado na figura 5.4. Utilizaremos a figura para especificar o aspecto dinâmico da navegação.

A navegação é o paradigma básico de interação e de consulta em nossa biblioteca. Uma sessão de consulta de um usuário pode começar de três formas: (1) O usuário acessa diretamente o Formulário de Consulta e submete uma consulta à biblioteca; (2) O usuário inicia o folheamento da biblioteca através dos índices hierárquicos que dão acesso aos contextos; (3) O usuário especifica uma consulta inicial através das aplicações “*Visualizador de Mapas*”, “*Gazetteer*” ou “*Formulário de Tempo*”, de forma a restringir as janelas espaciais e temporais de busca e a delimitar o seu espaço de navegação.

Usuários não familiarizados com o esquema da biblioteca, provavelmente utilizarão a segunda forma de interação. Desta forma, consideraremos primeiramente o aspecto de folheamento da biblioteca. O aspecto de consultas ao catálogo será coberto logo a seguir.

6.5.1 Folheamento da Biblioteca

Para o processo de folheamento da biblioteca, o *Menu Principal* oferece uma visão geral dos dados fornecidos. Os botões do Menu Principal estão associados às classes de dados disponibilizadas pela biblioteca. Eles dão ao usuário acesso a um conjunto predefinido de índices hierárquicos para o alcance dos contextos de navegação.

Os contextos de navegação apresentam diversas propriedades relevantes, sob o ponto de vista de navegação do usuário, de customização e de evolução da biblioteca. Primeiramente, os contextos são computados e os parâmetros para as consultas de alguns destes contextos podem ser fornecidos pelo usuário. Em segundo lugar, deve ser possível adicionar novos contextos e estender a biblioteca. A utilização dos *contextos parametrizados* de [AB91] em nossa biblioteca cobre satisfatoriamente este aspecto, oferecendo flexibilidade na extensão do esquema dos contextos.

Os índices hierárquicos que dão acesso aos contextos refletem uma estrutura baseada em características intrínsecas e em categorias temáticas dos dados. Em nossa biblioteca, estes índices estão organizados *por Tema*, *por Aplicação*, *por Itens de Gazetteer*, e *por Assunto*. Por exemplo, índices por Tema provêem entradas que refletem temas geográficos comuns tais como vegetação, transporte e hidrografia, dando acesso a objetos de Contextos

de Mapas Temáticos e de Imagens. Índices por Aplicação para Softwares de Aplicação definem itens para áreas de aplicação a que se destina o software. Índices por Assunto contêm entradas para assuntos principais concebidos para Bancos de Dados e Documentos

Além da navegação *top-down* através dos índices hierárquicos que permitem alcançar contextos específicos, mais três estilos de navegação são providos ao usuário da biblioteca: 1) navegações intracontexto; 2) navegação intercontextos e 3) navegações baseadas em seleção. A navegação intracontexto permite folhear dentro de um contexto específico, de acordo com o caminho de navegação especificado.

A navegação intercontextos (representada na figura 5.4 pelas setas saindo e entrando nos contextos) permite mudar o contexto no qual se está acessando um nó. Esta característica permite uma maior introspecção no conteúdo da biblioteca e fornece uma idéia bem mais clara do esquema, facilitando o encontro dos dados desejados.

As navegações baseadas em seleção são efetuadas através de seleções em porções dos dados e permitem atender demandas não cobertas pelo projeto inicial da biblioteca, bem como necessidades de informação de usuários bastante específicas. Este estilo de navegação ocorre com o auxílio do *Menu de Navegação*.

6.5.2 Consultas

Uma vez que o usuário ganha familiaridade com o conteúdo da biblioteca, é esperado que consultas sejam combinadas com navegação. Existem basicamente dois tipos de consultas: (1) aquelas que restringem o espaço de navegação do usuário e que reduzem a quantidade de dados a ser folheada e (2) aquelas que especificam buscas na biblioteca.

O primeiro tipo de consulta é contemplado pelos serviços oferecidos pelas aplicações-serviço ligadas à biblioteca. Estas aplicações são o "*Visualizador de Mapas*", o "*Gazetteer*" e o "*Formulário de Tempo*", acessadas via *Interface de Consulta*. Esta interface é constituída dos botões *Localização*, *Tempo* e *Consulta*. O item *Localização* aponta para um subíndice cujos itens direcionam para o Visualizador de Mapas ou para o Gazetteer. O botão *Tempo* dá acesso ao Formulário de Tempo e o botão *Consultas*, ao Formulário de Consultas. Esta Interface de Consulta está presente no Menu principal e em todas as ADVs dos objetos de navegação da biblioteca, com algumas alterações dos itens incluídos para certas classes particulares. Por exemplo, a interface de Consulta para Classes Convencionais não possui o item *Localização*.

As aplicações-serviço são responsáveis pela definição das janelas geográficas e temporais de navegação da sessão de consulta do usuário. Por *default*, a região geográfica de busca é a terra inteira e a janela temporal *default* corresponde a todos os valores de tempo válidos para o sistema. Caso o usuário deseje restringir as janelas espaciais ou temporais de sua busca, deve utilizar os serviços oferecidos pelas aplicações. O Visualizador de

Mapas e o Gazetteer definem a região geográfica de interesse do usuário para busca e navegação na biblioteca. O Formulário de Tempo é responsável pela determinação das datas de coleta e/ou de validade dos nós que serão percorridos durante a navegação. As escolhas feitas pelo usuário são passadas como parâmetros para as consultas que populacionam os contextos *Geo-Classe por Localização e Tempo* e *Classe Convencional por Tempo*.

O segundo tipo de consulta é suportado pelo *Formulário de Consultas*, que promove as consultas à biblioteca. O Formulário de Consultas é acessado via botão Consulta. As consultas do formulário são restritas, em seu escopo, ao contexto a partir do qual o formulário foi ativado. Quando esta navegação é requisitada, a informação sobre o contexto de navegação em que a ação ocorreu é passada ao formulário. Esta informação permite à aplicação-serviço relativa ao formulário customizar sua interface, apresentando campos de metadados a serem preenchidos de acordo com a classe associada ao contexto. A aplicação retorna resultados de consultas apresentando-os em uma interface de manipulação de conjuntos, além de um índice para o acesso a um nó específico. Cada entrada do índice contém informações parciais de metadados, recuperados na consulta, e uma ligação para acessar o nó. A interface de conjunto desta aplicação-serviço permite ao usuário: selecionar resultados de consultas e adicioná-los ao conjunto dos dados desejados; remover objetos do conjunto; salvar o conteúdo do conjunto; recuperar itens de dentro do conjunto e fazer “download” destes itens.

Visualizador de Mapas

O objetivo primordial do *Visualizador de Mapas* é permitir ao usuário definir uma região geográfica de interesse para consultas e navegações na biblioteca digital geográfica. A região geográfica especificada é passada como parâmetro para visões que irão ser populacionadas com dados cuja referência espacial casa com as especificações da consulta da visão. Estes dados recuperados constituirão o espaço de navegação - referente às classes espaciais da biblioteca - sobre o qual o usuário procederá sua navegação. Estes dados também determinarão o escopo de todas as demais consultas sobre a biblioteca.

Os objetivos desta aplicação incluem:

1. Visualização da região definida

Um Mapa Base contido no Visualizador de Mapas permite visualizar a região de interesse selecionada.

2. Apresentação espacial dos resultados de consultas

As extensões espaciais do resultado das consultas são mostradas no Mapa Base relativas à região de busca definida, permitindo aos usuários visualizarem a extensão espacial dos dados recuperados em relação a área de interesse.

3. Composição de consultas espaciais

Além da determinação da região de interesse, o usuário tem a opção de escolher o método de casamento espacial que é mais apropriado para a sua consulta. Uma lista dos métodos disponíveis - definidos em uma biblioteca de funções espaço-temporais - é apresentada. Esta lista permite ao usuário selecionar a função que mais lhe convém. O método de casamento e a região são encapsulados e enviados ao contexto Geo-Classe por Localização e Tempo como parâmetros de sua consulta.

Existem, na literatura, diversas abstrações que podem ser utilizadas em nosso Visualizador de Mapas. Estas abstrações definem métodos de determinação da região geográfica de interesse. Algumas destas abstrações são discutidas abaixo:

1. Formulários para preenchimento das Coordenadas do Retângulo Envolvente

O usuário especifica, em uma interface baseada em formulários, a região de interesse pelo preenchimento das coordenadas de latitude e de longitude do retângulo envolvente relativo à região de interesse. O Mapa Base será redimensionado para mostrar a região especificada.

2. Seleção sobre o Mapa Base

Esta é a forma mais natural de procedimento de especificação. O usuário desenha uma região retangular de interesse com o mouse. A região selecionada é então aumentada para preencher a janela do Mapa Base. À medida que a resolução aumenta, detalhes adicionais (e.g., limites de estados e cidades, rios) são apresentados no Mapa Base para permitir uma melhor orientação ao usuário. Esta seqüência pode ser repetida, e quando o usuário fica satisfeito com a região de interesse, suas coordenadas são copiadas para o Formulário de Coordenadas do Retângulo Envolvente, onde podem ser manualmente editadas se necessário. Esta é a abordagem utilizada pelo projeto Alexandria [ALD⁺95].

3. Navegação hipermídia sobre mapas

Neste caso, o Visualizador de Mapas compreende um Atlas Eletrônico, que provê os meios para navegação na superfície da Terra. O Atlas trabalha a partir de uma hierarquia de mapas, que é definida conectando mapas digitalizados com hiperligações de tipos dedicados. Quando o usuário clica dentro de um mapa que está sendo apresentado, todas as áreas definidas para este mapa são pesquisadas. Um menu *popup* é criado contendo como opções textos descritivos relacionados à todas as áreas que possuem o ponto selecionado. O menu *popup* é então mostrado. Se o usuário seleciona algum item no menu, um mapa associado ao item é apresentado. Botões

na interface conectam o mapa sendo apresentado a possíveis vizinhos. Esta abordagem é utilizada pelo sistema SpacePicture [Kir93], um sistema hipermídia para arquivamento de imagens de satélite.

O Visualizador de mapas de nossa biblioteca pode escolher dentre os métodos acima o que mais lhe convém, ou pode empregar uma combinação dos três métodos. Como exemplo, a figura 6.8 mostra o Folheador de Mapas da Biblioteca Digital Alexandria

Gazetteer

Uma outra forma de determinar a região de interesse na Biblioteca Digital Geográfica é selecionar nomes de lugares ou fenômenos geográficos de um gazetteer. Um gazetteer funciona como uma lista de consultas pré-computadas, contendo coordenadas de latitude e de longitude que localizam um lugar específico na superfície da Terra. Em adição à localização, o Gazetteer agrupa locais e fenômenos individuais em tipos e classes tais como hidrografia, transporte, vegetação e lugares populacionados.

O gazetteer é pesquisado por: 1) um nome específico; 2) uma cidade, estado, país ou região específicos; ou 3) uma classe ou tipo de fenômeno geográfico específico. Quando diversos critérios são usados para encontrar os nomes do lugar ou do fenômeno, o resultado é uma lista de nomes que satisfazem a todos os critérios.

Uma vez que um lugar ou fenômeno de interesse tenha sido selecionado, sua localização geográfica pode ser mostrada no Mapa Base do Visualizador de Mapas. Os conjuntos de dados que as consultas ao catálogo recuperam recaem dentro da região do retângulo envolvente mostrada no Mapa Base.

O único projeto de gazetteer digital que se tem notícia é o da Biblioteca Digital Alexandria (ADL) [ALD⁺95].

Formulário de Tempo

O Formulário de Tempo fornece meios para o usuário determinar a janela temporal de interesse. A interface do Formulário de Tempo oferece alternativas para que o usuário escolha o tipo de tempo a ser consultado (Data de publicação ou período de tempo do conteúdo); o tipo de data ou de tempo a ser preenchido (único, múltiplos ou intervalo); e os operadores temporais (durante, antes, depois, etc.). Os valores de entrada fornecidos pelo usuário através do Formulário de Tempo são capturados e enviados como parâmetros às consultas dos contextos Geo-Classe por Localização e Tempo e Classe Convencional por Tempo. A figura 6.9 mostra o exemplo de um Formulário de Tempo retirado do projeto *Clearinghouse FGDC* [Neb].

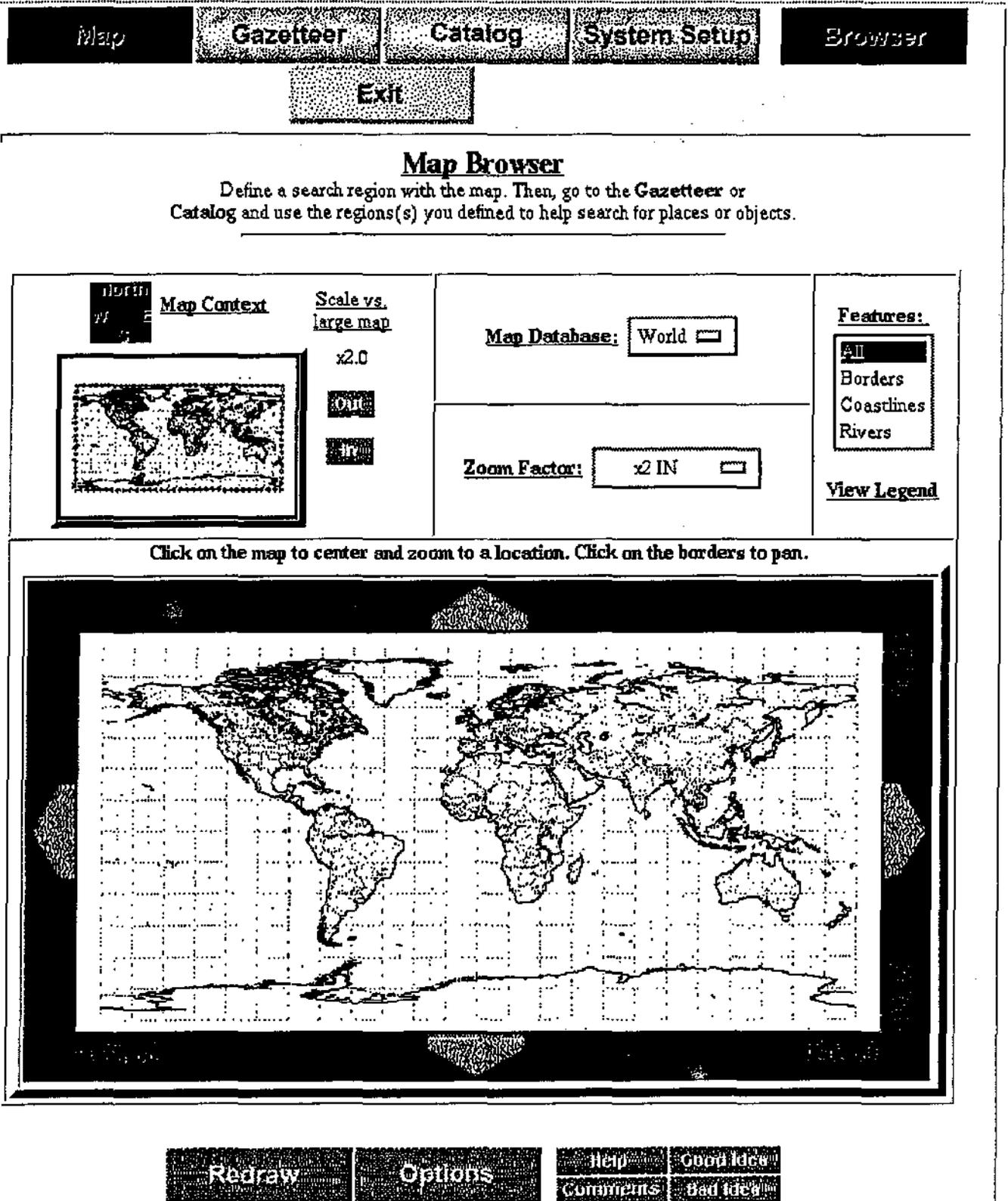


Figura 6.8: Exemplo de Visualizador de Mapas

Publication Date is the single date equal to

Publication Date is the range during
 through

Publication Date is within day(s) of the present

No Date Search

Figura 6.9: Exemplo de Formulário de Tempo

Formulário de Consulta

O Formulário de Consulta apresenta ao usuário uma interface baseada em formulários na qual o usuário preenche valores de atributos e metadados descrevendo as propriedades desejadas dos objetos. Estes formulários são customizados de acordo com as classes dos objetos que se quer acessar.

Uma vez que a consulta é submetida, o SGBDH se encarrega de processar a consulta traduzida e de retornar os resultados de volta ao formulário que os apresenta como um índice de resultados. As entradas do índice para cada item do resultado contêm um subconjunto de atributos e metadados do objeto, imagens reduzidas (optativas) do objeto, e uma âncora para o respectivo nó no SGBDH. As referências espaciais dos resultados de busca podem ser vistas no Mapa Base.

Os resultados das buscas do Formulário de Consultas e do Gazetteer são apresentados em uma interface orientada a conjuntos que permite marcar e desmarcar os itens recuperados; adicionar e salvar itens marcados no conjunto de resultados e recuperar itens previamente armazenados. A interface também permite mover-se para o Mapa Base de forma a visualizar as extensões espaciais dos itens de resultados marcados.

6.6 Considerações sobre as Classes de Metadados

6.6.1 Concordância com o Arcabouço Warwick

O *Arcabouço Warwick* [LLRD96] é uma arquitetura que provê ao Dublin Core uma formulação mais útil e concreta, de maneira a promover maior interoperabilidade entre provedores e catalogadores de conteúdo, indexadores, e sistemas automáticos de descrição e de descoberta de recursos.

O Arcabouço Warwick agrega múltiplos conjuntos de metadados e tem dois componentes principais: um *container* e *pacotes*. O container é uma unidade que agrega os conjuntos de metadados conhecidos como pacotes. No nível de container, cada pacote é um *stream* de bits, no qual o tamanho do pacote deve ser autodescrito. O conteúdo de pacotes individuais pode ser criptografado, permitindo o transporte dos metadados através de sistemas que não precisam ter acesso a conjuntos específicos.

Cada pacote é um objeto com um tipo que pode ser um *conjunto de metadados*, uma *indireção* ou um *container*. Conjuntos de metadados são pacotes que contém realmente metadados. Uma questão aberta é como aplicações clientes irão reconhecer e processar a semântica de múltiplos conjuntos de metadados. As aplicações que manuseiam conjuntos de metadados que podem ser estendidos irão requerer, obrigatoriamente, um esquema de registros de tipos. Indireção é um pacote que é uma referência indireta a outro objeto de informação. Este outro objeto pode ser um objeto de primeira classe, um pacote indiretamente referenciado, ou um repositório diferente. Um pacote *container* é ele próprio um container. Não existe limite para essa recursão.

O projeto de nossa hierarquia de classes de metadados, dentro do escopo da Biblioteca Digital Geográfica, entra em consonância com a maioria dos requisitos e das características do Arcabouço Warwick. *Containers* são representados pelas listas de metadados dos objetos; pacotes são instâncias de classes de metadados. A indireção é modelada diretamente pela classe Referência Indireta. A recursão pode ser alcançada no nível de definição das classes da própria hierarquia através de relacionamentos de agregação e especialização.

Em adição, o nosso trabalho provê descrições de tipos de classes e atributos de metadados e as armazena no dicionário de dados do SGBDOO. Além disso, são propostos mecanismos para registrar e introduzir novos modelos de metadados e agregá-los aos objetos, uma característica ainda aberta no Arcabouço Warwick.

6.6.2 Orientação a Objetos

A utilização do paradigma orientado a objetos é ideal para a definição de um modelo de metadados. Em primeiro lugar, diferentes classes de conjuntos de metadados recaem naturalmente em uma hierarquia de herança [LLRD96], espelhando diferentes taxonomias

de metadados. Em segundo lugar, as abstrações orientadas a objetos, tais como agregação e especialização, permitem modelar complexos modelos de metadados. Em terceiro lugar, sistemas orientados a objetos facilitam a navegação seguindo ponteiros (oid), enquanto sistemas relacionais exigem junções [FFLS96].

Por último, apesar de não ser uma característica ainda explorada em nossa hierarquia, o modelo OO permite modelar o comportamento dos objetos e definir atributos que são computáveis. Esta característica é bastante útil quando desenvolvendo classes de metadados para *termos e condições* de acesso aos objetos ou para permitir computações sobre os objetos da biblioteca, por exemplo, procedimentos de processamentos de imagens.

6.6.3 Trabalhos Relacionados

Diversos trabalhos na área de metadados têm se concentrado na especificação de conjuntos de atributos para descrição de objetos em ambientes digitais. Como exemplos citamos o Padrão DublinCore [WGMD95] e o Padrão de Conteúdo FGDC [Com]. Em particular, grande parte dos trabalhos na área de descrição de dados espaciais tem utilizado subconjuntos do padrão FGDC como base de seus sistemas (e.g., [BE96, HS96, ALD⁺95]). Estas abordagens se adaptam com facilidade à arquitetura de nosso catálogo.

Outros esforços significantes de pesquisa se preocupam em projetar arquiteturas que suportam a integração e a interoperação de vários conjuntos de metadados. Uma destas propostas é o arcabouço Warwick, cuja consonância com nosso trabalho já foi analisada. Outros esforços neste sentido são a arquitetura de metadados da Biblioteca Digital Stanford [BCGP97b, BCGP97a] e da Biblioteca Digital Alexandria [ALD⁺95].

A arquitetura de metadados da Biblioteca Digital Stanford provê uma infra-estrutura para fornecer interoperabilidade entre serviços autônomos e heterogêneos de bibliotecas digitais. A arquitetura tem quatro componentes principais: *Procuração de Modelos de Atributos*, *Serviços de Tradução de Modelos de Atributos*, *Procuração de Serviços de Busca* e *Repositório de Metadados*. Cada um destes componentes é implementado por um ou mais objetos CORBA distribuídos. O componente Procuração de Modelos de Atributos descreve modelos de atributos como coleções autocontidas de atributos que são, por sua vez, definidos como objetos de primeira classe. Este componente modela padrões de atributos conhecidos (e.g., Dublin Core) e permite expressar relacionamentos entre atributos. Os Tradutores de Modelos de Atributos servem para mediar diferentes convenções de metadados que são representadas pelas Procurações de Modelos de Atributos. Estes tradutores traduzem atributos e valores de atributos de um modelo para atributos e valores de um outro modelo. Procuração de Serviços de Busca é responsável por exportar metadados sobre o próprio serviço de busca, tais como tipos de truncamento utilizados e operadores booleanos suportados, bem como metadados sobre as coleções às quais o serviço dá acesso. Repositórios de Metadados são bancos de dados locais, possivelmente

replicados, que armazenam informações selecionadas de modelos de atributos, de tradutores de modelos de atributos, de procurações de busca, e de outros serviços providos pela arquitetura. O Repositório fornece informações úteis para uma "primeira busca" eficiente sobre metadados localmente relevantes.

Outro arcabouço é proposto pela Biblioteca Digital Alexandria [SGG96]. ADL modela a metainformação de documentos com pares valor/atributo no quais múltiplas linguagens podem ser utilizadas. Em particular, a Biblioteca Alexandria emprega uma combinação dos padrões USMARC e FGDC na descrição de seus dados. O repositório de metadados de Alexandria é definido em um modelo Relacional e implementado sobre um SGBD Relacional.

A arquitetura de nosso catálogo absorve e integra a maior parte das características das propostas acima relacionadas. Ela suporta múltiplos conjuntos de metadados na modelagem dos documentos; torna modelos de atributos públicos para usuários e aplicações através do dicionário de dados do SGBD; suporta indireção e recursão como no arcabouço Warwick; provê mecanismos para especificar e registrar novos conjuntos de metadados. As características diferenciadoras de nosso trabalho são a provisão de uma hierarquia de classes de metadados especializada, flexível e extensível, que é definida conjuntamente com o modelo hipermídia abstrato para permitir projetar aplicações de biblioteca. Todos estes mecanismos são integrados em um SGBDOO.

Outra característica inovadora de nosso trabalho é o uso de metadados em novos papéis tais como suporte à navegação hipermídia no nível de extremos de ligações e de contextos de navegação e como informação base para o particionamento e organização das coleções da biblioteca, utilizando-se visões com parâmetros de metadados.

Ainda outros exemplos de sistemas que provêem funcionalidades hipermídia baseadas em metadados são o Sistema de Dados Planetário da NASA [HB95] e o sistema DARE do Jet Propulsion laboratory da Nasa [HB96].

6.7 Resumo

Catálogos, em bibliotecas tradicionais, são essenciais para a recuperação de informação. Este papel é ainda mais importante no caso de bibliotecas digitais, onde servem para a base de navegação e de consulta hipermídia.

O capítulo 4 propôs um modelo para uma biblioteca digital, que foi instanciado no capítulo 5 para uma Biblioteca Digital Geográfica. Este capítulo definiu o catálogo desta biblioteca digital do ponto de vista de metadados, e mostrou como este catálogo serve de base para consultas em bibliotecas digitais.

A hierarquia de classes de metadados apresentada é um projeto especializado para descrição de dados espaciais e convencionais, representando conjuntos de metadados como

classes da hierarquia. Estas classes provêm uma base racional para implementação estruturada destes conjuntos e padrões de metadados sobre um SGBDOO, que se encarrega do gerenciamento e do acesso aos metadados. As listas de metadados dos nós são um mecanismo flexível para integração das informações contidas nestas classes.

As características do modelo de consulta da biblioteca permitem uma integração simbiótica de folheamento e de consulta além de uma flexibilidade raramente encontrada em sistemas de bibliotecas digitais. O espaço de navegação é definido por consultas do usuário e do sistema. A navegação, por sua vez, permite a avaliação da adequação e do escopo das consultas, e sua reavaliação se necessário.

Capítulo 7

Conclusões e Extensões

7.1 Conclusões

Esta dissertação propôs um modelo e uma metodologia para a construção de aplicações de bibliotecas digitais. Aqui, uma biblioteca digital foi considerada como uma aplicação hipermídia, com características e requisitos especiais, rodando sobre um Sistema Gerenciador de Banco de Dados Hipermídia Orientado a Objetos. Modelo e metodologia foram empregados em uma aplicação particular: uma Biblioteca Digital Geográfica que tem por objetivo colecionar e prover acesso a uma grande quantidade de dados geográficos e convencionais em rede.

A construção desta biblioteca exigiu a definição de um conjunto específico de metadados, que agrupa diversas propostas de padrão. A biblioteca geográfica suporta dois modos de interação: folheamento e consulta (a partir do SGBD subjacente). Ambos os modos são fortemente integrados no ambiente da biblioteca.

O modelo proposto integra o modelo do SGBD OOHDM de Milet et al. [MSL96] com conceitos do modelo de Dexter Estendido de [GT96], de forma a estender ambas as abordagens. A metodologia proposta é resultado de extensões à metodologia de projeto de aplicações hipermídia escolhida (OOHDM), adequando-a à modelagem de uma biblioteca digital.

Dentre as principais contribuições desta dissertação, citamos:

- estudo detalhado dos principais conceitos de bibliotecas digitais; apresentação de uma taxonomia genérica para os componentes destas bibliotecas; avaliação do uso de dados geográficos em bibliotecas digitais (Capítulo 2);
- estudo detalhado de modelos hipermídia, estabelecendo um conjunto de requisitos de um modelo hipermídia para aplicações de bibliotecas digitais (Capítulo 3);

- proposta de um modelo hipermídia orientado a objetos para estas bibliotecas (Capítulo 4);
- propostas de extensões à metodologia do OOHDMM para acomodar a modelagem de aplicações de bibliotecas digitais (Capítulo 5);
- especialização do modelo e utilização da metodologia estendida na modelagem de uma Biblioteca Digital Geográfica e posterior mapeamento para um SGBDOO (capítulo 5);
- definição do ambiente de catálogo eletrônico da biblioteca digital no contexto do SGBD hipermídia e especificação do catálogo para a biblioteca digital geográfica (Capítulo 6).

7.2 Extensões

Diversas extensões podem ser propostas a esta dissertação. A principal extensão seria a implementação do modelo em um SGBDOO. Esta implementação envolve diversas considerações, a saber:

1. Escolha do SGBDOO

Aqui, deve-se levar em consideração se o SGBDOO incorpora um mecanismo de visões, se este mecanismo possui os componentes necessários à biblioteca (e.g., um Gerenciador de Atualizações) e se o mecanismo é embutido ou acoplado ao SGBD, permitindo sua extensão e alteração.

2. Especificação dos algoritmos e das operações de navegação

Envolve a definição da política de casamento entre LocConts e EspLocs, além da definição de estruturas de dados para as operações de navegação que otimizem este casamento.

Nas subseções a seguir, explicitaremos outras extensões ao trabalho, divididas em extensões ao modelo, extensões ao ambiente e extensões ao catálogo.

7.2.1 Extensões ao Modelo

Extensão do Papel de Visões

No trabalho de Rossi [Ros96], que definiu o OOHDMM, uma aplicação hipermídia é construída como uma visão navegacional do esquema conceitual da aplicação, na qual nós e

ligações são gerados a partir de classes e de relacionamentos com o uso de mecanismos de visões. No trabalho, Rossi define uma linguagem de visões orientadas a objetos para este propósito especial. Sendo assim, uma extensão interessante seria a integração dos paradigmas de visões de nossos contextos com o trabalho de Rossi, e o estudo e avaliação de extensões nas possibilidades do uso de visões para definição de aplicações hipermídia.

Outro uso interessante do mecanismo de visões no âmbito de bibliotecas digitais seria na geração de *documentos virtuais ou dinâmicos*, documentos que realmente não existem mas que podem ser computados sob demanda usando dados do banco de dados. Questões desta pesquisa envolvem: Quais as características de documentos virtuais? Quais metadados são necessários para descrever documentos virtuais? Como construir catálogos para documentos virtuais? Como interligar e navegar entre documentos virtuais e reais?

Ainda outra extensão do uso de visões em sistemas hipermídia seria em relação aos aspectos de autorização, de segurança e de controle de acesso em sistemas hipermídia. Questões de pesquisa envolvem considerações sobre dados não estruturados presentes em hipermídias; restrição de navegação que envolve esconder/apresentar âncoras; problemas com o reuso dos dados em diversos contextos, através de diferentes caminhos de navegação; questões relativas à disseminação de cópias.

Melhoria do suporte à Navegação baseada em Conteúdo

Apesar da utilização de ligações dinâmicas e genéricas abrir a possibilidade de navegação através de mídias não textuais, utilizando-se o conteúdo destas mídias como a base da navegação, a utilização efetiva desta facilidade requer alguns refinamentos.

Em primeiro lugar, é preciso extrair a informação desejada a partir da seleção na mídia. Esta extração deve ser baseada em alguma propriedade invariante específica da mídia como, por exemplo, distribuição de cores, textura e forma, para imagens; padrões de onda, para voz.

Em segundo lugar, como consequência, devem ser definidos algoritmos e métodos específicos para o tratamento desta informação e para o casamento baseado em conteúdo entre a seleção e extremos de ligações, tendo como finalidade a navegação. Isto envolve áreas como processamentos de imagens e reconhecimentos de padrões. Uma característica deste casamento é que ele é por definição “*fuzzy*”, pois é geralmente impossível efetuar casamento exato em aplicações não textuais. Estes métodos multimídia devem ser responsáveis pela extração das propriedades desejadas da mídia, pelo armazenamento e indexação de extremos de ligações e pela recuperação e casamento das porções de mídia indexadas.

Em terceiro lugar, sob a perspectiva do usuário, a interface deve prover meios para este usuário definir o tipo de casamento requerido (e.g., cor ou textura, no caso de imagens) e para definir parâmetros para estes casamentos.

7.2.2 Extensões ao Ambiente da Biblioteca

Integração com Sistemas de Recuperação de Informações e com On-Line Analytical Processing (OLAP)

A utilização de um banco de dados como plataforma de desenvolvimento para uma biblioteca digital permite beneficiar-nos do poder da linguagem de consultas do BD subjacente. Consultas ao BD permitem recuperar objetos da biblioteca que preencham condições especificadas na consulta.

O poder das consultas da biblioteca pode, contudo, ser acrescido se provermos sua integração com Sistemas de Recuperação de Informação (SIR) e com técnicas da área de *On-Line Analytical Processing* (OLAP) [For]. SIRs podem aumentar o poder de consultas à biblioteca na medida em que eles oferecem características tais como indexação pelo conteúdo dos dados, uso de cálculos de probabilidade e de métodos de inferência estatística na recuperação de informação, utilização de linguagem natural e de conceitos de incerteza e de imprecisão nas consultas, apresentação de respostas de consultas segundo ordenações graduais. Já técnicas de OLAP provêem métodos para agrupar subconjuntos de dados, para apresentar agregados e características comuns aos dados em cada grupo, e para sumarizar respostas de consultas. Estas características podem ser utilizadas para fornecer informação ao usuário de forma a permitir-lhe refinar suas consultas mais adequadamente.

Aspectos de pesquisa aqui envolvem como integrar as características relacionadas de SIR e de OLAP à biblioteca de forma efetiva e eficiente e como estender técnicas tradicionais de ambas as áreas - normalmente associadas a processamento de textos e a dados alfa-numéricos - para o tratamento de dados multimídia.

Utilização de Mecanismos Ativos

Esta extensão contempla a investigação de como mecanismos ativos podem ser acoplados ao SGBD hipermídia, de forma a oferecer funcionalidades tais como:

1. Customização ativa de ADVs, permitindo a construção de interfaces de biblioteca personalizadas que incorporam as preferências do usuário e provêem diferentes comportamentos em diferentes contextos e situações. Isto envolve a investigação de extensões de regras ativas para manusear eventos externos tais como eventos de navegação na interface;
2. *Caching*, ocasionando a movimentação de objetos freqüentemente acessados - segundo regras estatísticas - para meios de armazenamento mais rápidos;
3. Auxílio ao mecanismo do Gerenciador de Atualizações, inclusive para repopulamentos de contextos;

4. Auxílio na Manutenção de Consistência, como por exemplo, remoção de “referências perdidas” explícitas frente a remoção de nós ou âncoras.

Melhoria do suporte e interação do usuário com a biblioteca

Em nossa biblioteca, existe uma distinção clara entre o usuário e o “bibliotecário”, que é responsável pela organização da biblioteca. Esta organização é expressa em termos da disposição e da interligação dos contextos de navegação. Apesar da abordagem proposta oferecer uma flexibilidade ao usuário no que diz respeito às possibilidades de navegação e à definição de seu espaço de navegação e apesar de fornecer uma integração simbiótica entre folheamento e consulta, a interação do usuário com a biblioteca pode ainda ser estendida se eliminarmos a distinção inicial.

Extensões, neste sentido, aparecem como consequência da necessidade de oferecer suporte para usuários livremente criarem suas próprias ligações, seus próprios contextos de navegação e ADVs customizadas.

Os aspectos de pesquisa pertinentes a estas extensões incluem a definição de ambientes de trabalho para cada usuário e extensões nas visões parametrizadas, no estilo de navegação e na apresentação dos nós de contexto. Características que devem ser consideradas incluem buscas e consultas sobre ambientes de trabalhos específicos, como integrar os contextos de usuários na hierarquia de contextos e no Esquema dos Contextos de Navegação, como restringir a navegação a espaços delimitados.

Integração com a WWW

A partir do momento em que consideramos que um dos objetivos principais da biblioteca digital é prover acesso à informação armazenada ao maior número possível de usuários, a integração de nosso arcabouço com a World-Wide Web (WWW) aparece como uma extensão natural. Apesar de alguma provisão neste sentido já ter sido oferecida no modelo, através da incorporação dos EspLocs que são capazes de modelar referências na WWW - como ligações embutidas em páginas HTML e como URLs de “flutuação livre” - problemas ainda existem. A WWW é uma tecnologia de redes que não foi projetada para trabalhar diretamente com bancos de dados, que é a tecnologia básica de nossa biblioteca. É necessário que os SGBDs sejam **integrados** ao ambiente WWW [Str96].

O nosso SGBD hipermídia pode ser encarado sob duas óticas distintas: como sendo um SGBDOO e como sendo um sistema hipermídia. Esta característica permite beneficiar-nos de trabalhos prévios que tratam da integração de SGBDOOs a WWW (e.g., [HAI97, Sjo94, JK96, O2W]) e de trabalhos que propõem abordagens para a integração de sistemas hipermídia a WWW (e.g., [AKM95, And97, GBS97]), aproveitando e adaptando as melhores soluções de cada área.

A integração do SGBD HiperMídia à WWW envolve a análise das possibilidades, das conseqüências, das alterações impostas na arquitetura e no modelo, além da análise das vantagens e desvantagens de utilizarmos a WWW como um *front-end* para a biblioteca digital. Entre as desvantagens, por exemplo, cita-se a inabilidade de *browsers*, mesmo utilizando a tecnologia Java [CH96], em manusear eventos relativos a seleções em porções de textos, impossibilitando assim a navegação baseada em seleção [GBS97].

Duas soluções iniciais merecem investigação. A primeira utiliza *gateways* CGI acoplados a arquivos especiais de configuração, que podem ser construídos com base nas informações do dicionário de dados do SGBDH. A segunda é inspirada na proposta de [O2W]. A idéia seria sobrecarregar os métodos montar_tela e apresentar das ADV dos nós para gerarem produções HTML. Estes métodos gerenciariam métodos “web_display” definidos em cada classe de ADV aninhada. Estes métodos produziriam código HTML para toda ADV aninhada na estrutura de uma ADV principal segundo um mapeamento básico entre primitivas do BD e códigos HTML.

Questões adicionais de pesquisa, neste sentido, incluem aspectos relacionados à performance do sistema, que podem incluir propostas de divisão da carga de trabalho entre o cliente e o servidor WWW, segmentação da interface do cliente em *frames* independentes, e utilização de agentes de banco de dados como processos *daemon*, que diminuiriam a sobrecarga de inicializações sucessivas do *gateway* CGI.

7.2.3 Extensões ao Catálogo

Aumento do Conjunto de Metadados

Uma extensão natural, neste caso, é o aumento do conjunto de metadados coberto pela hierarquia de classes de metadados. Esta extensão far-se-ia presente através da inclusão de novas classes de metadados como, por exemplo, *Metadados para Classificação de Conteúdo* e *Termos e Condições*, que descrevem regras para o acesso aos dados. Em particular, o primeiro tipo de metadados inclui a extração de informação do conteúdo dos objetos e a classificação dinâmica do conteúdo extraído; o segundo tipo envolve serviços de negociação, interação, pagamento, autenticação, dentre outros. Estes tipos de metadados interativos e adaptativos são melhor expressos via computações. Isto envolve uma segunda extensão: a definição de atributos e metadados *computáveis*, o que é facilmente acomodado em nosso projeto definindo estes atributos como chamadas a métodos das classes de metadados. Por exemplo, tais atributos podem acomodar métodos para processamentos de imagens ou outros procedimentos úteis que podem ser aplicados pelo usuário para o atendimento de seus requisitos de informação.

Um outro tipo de metadados útil seriam metadados sobre as coleções e sobre os contextos. Estes metadados incluiriam descrições das coleções e dos contextos como, por

exemplo, consulta e parâmetros do contexto, informações sobre facilidades de consultas às coleções, e informações estatísticas necessárias para que metabuscas prognostiquem a relevância da coleção para consultas particulares.

Extensões aos Serviços do Catálogo

Outra possível extensão é a inclusão de novos serviços à biblioteca como, por exemplo, serviços de traduções que traduzem entre modelos de metadados para facilitar a interoperabilidade no sentido biblioteca → mundo externo, e serviços de busca mais inteligentes que poderiam utilizar informações sobre relacionamentos entre atributos e metadados de coleções e de contextos.

Bibliografia

- [AAB⁺95] D. Andersen, P. Avery, P. Bloniarz, S. Dawes, S. Hyde, K. Kelly, A. Miller, and E. Rich. Balacing environmental quality and economic vitality in the adirondack park. Technical report, Center for Technology in Government, December 1995.
- [AB91] S. Abiteboul and A. Bonner. Object and views. In *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data*, pages 238–247, 29-31, may 1991.
- [AC93] H. Ashman and G. Chase. Link management within a geographic hypermedia information system. In *Proceedings of the 4th Autralian Conference on Information Systems*, Brisbane, Australia, September 1993.
- [AFY94] N. R. Adam, B. S. Fordham, and Y. Yesha. Some key issues in database systems in a digital library setting. In *Lecture Notes in Computer Science*, volume 916, pages 9–20. Springer Berlin, 1994.
- [AHN94] N. R. Adam, M. Halem, and S. Naqvi. Promissing research directions in digital libraries. In *Lecture Notes in Computer Science*, volume 916, pages 21–30. Springer Berlin, 1994.
- [AKM95] K. Andrews, F. Kappe, and H. Maurer. Serving information to the web with hyper-g. *Computer Networks and ISDN Systems*, 27(6), April 1995.
- [ALD⁺95] D. Andresen, L. Carver, R. Dolin, C. Fischer, J. Frew, M. Goodchild, O. Ibarra, R. Kothuri, M. Larsgaard, B. Manjunath, D. Nebert, J. Simpson, T. Smith, T. Yang, and Q. Zheng. The www prototype of the alexandria digital library. In *Proceedings of the International Symposium on Digital Libraries*, Tsukuba, Japan, 1995.
- [And97] K. M. Anderson. Integrating open hypermedia systems with the world wide web. In *Proceedings of the ACM Hypertext 97*, Southampton UK, 1997.

- [AS94] J. T. Anderson and M. Stonebraker. Sequoia 2000 metadata schema for satellite images. *SIGMOD RECORD*, 23(4):42–48, December 1994.
- [AV94] H. Ashman and J. Verbyla. Dynamic link management via the functional model of the link. In *Proceedings of the Basque International Workshop on Information Technology*, France, February 1994.
- [AVC94] H. Ashman, J. Verbyla, and T. Cawley. Hypermedia management in large-scale information systems using the functional model of the link. In *Proceedings of the 5th Australasian Database Conference'94*, Christchurch, New Zealand, January 1994.
- [Bal93] V. Balasubramanian. Hypermedia issues and applications: a state-of-art review. <http://www.netspot.unisa.edu.au/tutorial/hypertext.html>, November 1993.
- [BAN94] K. Böhm, K. Aberer, and E. Neuhold. Administering structured documents in digital libraries. In *Lecture Notes in Computer Science*, volume 916, pages 91–118. Springer Berlin, 1994.
- [BCGP97a] M. Baldonado, C. K. Chang, L. Gravano, and A. Paepcke. Metadata for digital libraries: Architecture and design rationale. Technical report, Stanford University, 1997.
- [BCGP97b] M. Baldonado, C. K. Chang, L. Gravano, and A. Paepcke. The Stanford digital library metadata architecture. *International Journal of Digital Libraries*, 1(2), February 1997.
- [BE96] B. Bicking and R. East. Dynamically integrating spatial data and metadata. In *Proceedings of the First IEEE Metadata Conference*, Silver Spring, Maryland, April 1996. IEEE.
- [BI95] M. Bieber and T. Isakowitz. Designing hypermedia applications. *Communications of the ACM*, 38(8):26–29, August 1995. special issue in Hypermedia Application Design.
- [Boo92] J. Boone. A survey of data models for hypermedia. Technical report, IBM Corporation and Department of Computer Science University of North Carolina, April 1992.
- [BR94] K. Böhm and T. C. Rakow. Metadata for multimedia documents. *SIGMOD RECORD*, 23(4):21–26, December 1994.

- [BWAH96] A. Bapat, J. Wasch, K. Aberer, and J. M. Haake. Hyperstorm: An extensible object-oriented hypermedia engine. In *Proceedings of the 7th ACM Conference on Hypertext*, pages 203–214, march 1996.
- [CCH⁺96] G. Câmara, M. A. Casanova, A. S. Hemerly, G. C. Magalhães, and C. M. B. Medeiros. *Anatomia de Sistemas de Informação Geográfica*. Instituto de Computação - UNICAMP, 1996. X Escola de Computação, Campinas, 8 a 13 de Julho de 1996.
- [Cer96] N. Cereja. Visões em sistemas de informações geográficas - modelo e mecanismos. Master's thesis, Instituto de Computação - Universidade Estadual de Campinas, 1996.
- [CG88] B. Campbell and J. M. Goodman. Ham: A general purpose hypertext abstract machine. *Communications of the ACM*, 31(7):856–861, July 1988.
- [CH96] G. Cornell and C. S. Horstmann. *Core Java*. Sunsoft Press, 1996.
- [CMK⁺94] F. Chen, M. Hearst, J. Kupiec, J. Pederson, and L. Wilcox. Metadata for mixed-media acess. *SIGMOD RECORD*, 23(4):64–71, December 1994.
- [Com] Federal Geographic Data Committee. Content standards for digital geospatial metadata. <http://geochange.er.usgs.gov/pub/tools/metadata/standard/metadata.html>.
- [CT95] A. Cornelio and G. Thomas. Geoscope requirements. Technical report, Bellcore - Bell Communications Research, June 1995.
- [DA97] P. Duguid and D. Atkins. Digital libraries- report of the santa fe planning workshop on distribued knowledge work environments. Technical report, University of Michigan School of Information, March 9-11 1997.
- [dCG97] S. A. B. da Cruz and L. Gonçalves. Cooperativa de banco de dados EM-BRAPA. In *Anais do GISBRASIL 97 - III Congresso e Feira para Usuários de Geoprocessamento*, Curitiba - Paraná, 12 - 16 maio 1997.
- [DHH⁺92] H. Davis, W. Hall, I. Heath, G. Hill, and R. Wilkins. Toward an integrated information environment with open hypermedia systems. In *ECHT'92 European Conference on Hypertext*, pages 181–190, Milano, Italy, 1992.
- [dS95] C. S. dos Santos. Design and implementation of object-oriented views. *Lecture Notes in Computer Science*, 978:91–102, 1995.

- [EMM94] M. Elkington, R. Meyer, and G. McConaughy. Defining the architectural development of eosdis to facilitate extension to a wider data information system. Technical Report 194-00131, EOSDIS Core System Project, April 1994.
- [EW94] A. Endal and S. Wharton. Ecs science requirements summary. Technical Report 194-00131, Hughes Applied Information Systems, Inc., April 1994. EOSDIS Core System Project.
- [FCF⁺95] J. Frew, L. Carver, C. Fischer, M. Goodchild, M. Larsgaard, T. Smith, and Q. Zheng. The alexandria rapid prototype: building a digital library for spatial information. In *1995 ESRI User Conference Proceedings*. Environmental Systems Reserach Institute, 1995.
- [FFLS96] C. Fischer, J. Frew, M. Larsgaard, and T. R. Smith. Alexandria digital library: Rapid prototype and metadata schema. *Lecture Notes In Computer Science*, 1082:221-241, 1996.
- [FHN97] P. Frohlich, N. Henze, and W. Nejd. Meta modeling for hypermedia design. In *Proceedings of the Second IEEE Metadata Conference*, Silver Spring, Maryland, September 16-17 1997.
- [For] S. Forsman. Olap Council white paper. <http://www.olapcouncil.org/research/whtpaply.htm>.
- [Fuh96] N. Fuhr. Models for integrated information retrieval and databases. *IEEE Computer Society - Bulletin of The Technical Committe on Data Engineering*, 19(1):3-13, March 1996.
- [GBA⁺94] H. M. Gladney, N. J. Belkin, Z. Ahmed, E. A. Fox, R. Ashany, and M. Zemanakova. Digital library: Gross structure and requirements. In *Proceedings of Digital Libraries '94*, 1994.
- [GBS97] K. Gronbaek, N. O. Bouvin, and L. Sloth. Designing dexter-based hypermedia services for the world wide web. In *Proceedings of the ACM Hypertext'97 Conference*, Southampton, England, 1997.
- [GMP94a] F. Garzotto, L. Mainetti, and P. Paolini. Adding multimedia collections to the dexter model. In *Procceedings of ECHT'94 - European Conference on Hypermedia Technology*, pages 70-80, Edinburgh, UK, September 1994.

- [GMP94b] F. Garzotto, L. Mainetti, and P. Paolini. Hdm2: Extending the e-r approach to hypermedia application design. In *Lecture Notes in Computer Science*, volume 823, pages 178–189, 1994. 12th International Conference on the Entity-Relationship Approach.
- [GMP95] F. Garzotto, L. Mainetti, and P. Paolini. Hypermedia design, analysis, and evaluation issues. *Communications of the ACM*, 38(8):74–85, August 1995.
- [GPS93] F. Garzotto, P. Paolini, and D. Schwabe. Hdm - a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, January 1993.
- [GT94] K. Gronbaek and R. Trigg. For a dexter-based hypermedia system. *Communications of the ACM*, 37(2):51–49, February 1994.
- [GT96] K. Gronbaek and R. Trigg. Toward a dexter-based model for open hypermedia: Unifying embedded references and link objects. In *Proceedings of Hypertext '96*, Washington, D.C., March 16-20 1996.
- [GV97] O. Gunther and A. Voisard. Metadata in geographic and environmental data management. In W. Klas and A. Shet, editors, *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data*. McGraw Hill, 1997.
- [HAI97] J. Hammerand, R. Aranha, and K. Ireland. Browsing object databases through the web. Technical report, Computer Science Department - Stanford University, 1997.
- [HB95] J. S. Hughes and A. Bernath. The Planetary Data System Web catalog interface - another use of the planetary data system data model. In *Proceedings of the 14th IEEE Symposium on Mass Storage Systems*, pages 263–273, Monterey, Ca, USA, 1995.
- [HB96] J. J. Hyon and R. B. Borgen. Data archival and retrieval enhancement (DARE) metadata modeling and its user interface. In *Proceedings of the First IEEE Metadata Conference*, Silver Spring, Maryland, April 1996. IEEE.
- [HS94] F. Halasz and M. Schwartz. The dexter hypertext. *Communications of the ACM*, 37(2):30–39, February 1994.
- [HS96] D. Hansen and M. Sebat. Reporting metadata for access in arcview and across internet within the arc-info data model. In *Proceedings of the ESRI User Conference*, Palm Springs, California, 1996.

- [ISB95] T. Isakowitz, E. A. Stohr, and P. Balasubramanian. Rmm: A methodology for structured hypermedia design. *Communications of the ACM*, 38(8):34-44, August 1995.
- [JH94] R. Jain and A. Hampapur. Metadata in video databases. *SIGMOD RECORD*, 23(4):27-33, December 1994.
- [JK96] J. Jingshuang and G. E. Kaiser. An architecture for integrating oodbs with www. In *Proceedings of the International World Wide Web Conference*, Paris, França, May 6-10 1996.
- [Kir93] T. Kirste. Spacepicture - an interactive hypermedia system satellite image archival system. *Computer & Graphics*, 17(3):251-260, 1993.
- [KJSK95] C. Kacmar, D. Jue, D. Stage, and C. Koontz. Automatic creation and maintenance of an organizational spatial metadata and document digital library. In *Proceedings of the DL'95*, 1995.
- [KK96] P. Kirstein and G. M. Kohsari. The c-oda project: Online access to electronic journals. *Communications of the ACM*, 39(6):88-99, June 1996.
- [KKH94] Y. Kyoski, T. Kitagawa, and T. Hayama. A metadatabase system for semantic image search by a mathematical model of meaning. *SIGMOD RECORD*, 23(4):34-41, December 1994.
- [KPD⁺95] K. Kelly, T. Pardo, S. Dawes, A. DiCarerino, and W. Hérad. Sharing the cost, sharing the benefits: The nys gis cooperative project. Technical report, New York Department of Environmental Conservation Center for Technology in Government, December 1995.
- [KS94] W. Klas and A. Shet. Metadata for digital media: Introduction to the special issue. *SIGMOD RECORD*, 23(4):19-20, December 1994.
- [Lan94] D. B. Lange. An object-oriented design method for hypermedia information systems. In *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, pages 366-375, 1994.
- [Las96] R. Lasher. Survey of the six dli projects. Documento disponível via <http://www-diglib.stanford.edu/diglib/WP/PUBLIC/DOC46.html>, January 1996.

- [LDG+96] P. H. Lewis, H. C. Davis, S. R. Griffiths, W. Hall, and R. J. Wilkins. Media-based navigation with generic links. In *Proceedings of the 7th ACM Conference on Hypertext*, pages 215–223, New York, 16-20, march 1996. ACM Press.
- [LDH92] Z. Li, H. Davis, and W. Hall. Hypermedia links and information retrieval. In *British Computer Society 14th Information Retrieval Colloquium*, Lancaster University, April 1-14 1992.
- [LGM95] C. Lynch and H. Garcia-Molina. IITA digital libraries workshop report - interoperability, scaling, and the digital libraries research agenda. A Report on the May 18-19, 1995 IITA Digital Libraries Workshop, August 1995.
- [LLRD96] C. Lagoze, C. A. Lynch, and Jr. R. Daniel. The warwivk framework: A container architecture for aggregating sets of metadata. Technical report, Cornell University, June 1996.
- [LM95] D. M. Levy and C. C. Marshall. Going digital: a look at assumptions underlying digital libraries. *Communications of the ACM*, 38(8):77–84, April 1995.
- [LS94] J. J. Leggett and J. L. Schnase. Viewing dexter with open eyes. *Communications of the ACM*, 37(2):76–86, February 1994.
- [MG97] C. M. B. Medeiros and M. A. Gonçalves. Bibliotecas digitais para dados geográficos. In *Anais do GISBRASIL 97 - III Congresso e Feira para Usuários de Geoprocessamento*, Curitiba - Paraná, 12 - 16 maio 1997.
- [MP94] C. B. Medeiros and F. Pires. Databases for gis. *SIGMOD RECORD*, 23(1):107–115, March 1994.
- [MSL96] J. R. L. Milet, D. Schwabe, and R. S. G. Lanzelotte. Autoria de aplicações hipermídia utilizando um banco de dados orientado a objetos. In *Anais do XI SBBD - Simpósio Brasileiro de Banco de Dados*, 1996.
- [Neb] D. Nebert. Geospatial data clearinghouse activity. <http://www.fgdc.gov/clearinghouse>.
- [NFL+95] P. J. Nurnberg, R. Furuta, J. J. Leggett, C. C. Marshall, and F. M. Shipman III. Digital libraires: Issues and architectures. In *Proceedings of DL '95*, 1995.

- [NS91] J. Noll and W. Scacchi. Integrating diverse information repositories: A distributed hypertext approach. *IEEE Computer Magazine*, pages 38–45, December 1991.
- [O2W] O2Web. O2web- white paper. <http://www.o2tech.com/>.
- [OPM97] J. Oliveira, F. Pires, and C. B. Medeiros. An environment for modelling and design of geographic applications. *GeoInformatica*, 1(1), 1997.
- [Por] Projeto Portinari. <http://www.portinari.org.br/>.
- [QRS⁺95] D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. Querying semistructured heterogeneous information. In *Lecture Notes in Computer Science*, volume 1013. Springer, 1995. Proceedings of Fourth International Conference on Deductive and Object-Oriented Databases, DOOD'95.
- [Ros96] G. Rossi. *Um Método Orientado a Objetos para o Projeto de Aplicações Hipermídia*. PhD thesis, Departamento de Informática, PUC-Rio, 1996.
- [RSLC95] G. Rossi, D. Schwabe, C. J. P. Lucena, and D. D. Cowan. An object-oriented model for designing the human-computer interface of hypermedia applications. In *International Workshop on Hypermedia Design*, Montpellier, France, 1995.
- [Sam95] H. Samet. Spatial data structures. In Won Kim, editor, *Modern Database Systems*, pages 361–385. ACM Press, 1995.
- [SB94] D. Schwabe and S. D. J. Barbosa. Navigation modelling of hypermedia applications. Technical Report MCC 42/94, Departamento de Informática, PUC-Rio, 1994.
- [SCB95] L. Shklar, C. St. Charles, and C. Behrens. Geoscope detailed design. Technical report, Bellcore - Bell Communications Research, September 1995.
- [SGG96] T. R. Smith, S. Geffner, and J. Gottsegen. A framework for catalogs and metainformation in the context of digital libraries. In *Proceedings of First IEEE Workshop on Metadata*, 1996. The Alexandria Digital library, University of California, Santa Barbara, CA (documento disponível via <http://alexandria.sdc.ucsb.edu/public-documents/ieee/>).
- [Sjo94] M. Sjolín. A WWW front end to an oodbms. In *First International World-Wide Web Conference*, Geneva CH, May 1994.

- [Smi96] T. Smith. The meta-information environment of digital libraries. *D-Lib Magazine - The Magazine of Digital Library Research*, July/August 1996.
- [Son96] G. Sonnenberger. Exploiting the functionality of object-oriented database management systems for information retrieval. *IEEE Computer Society - Bulletin of The Technical Committee on Data Engineering*, 19(1):14-23, March 1996.
- [SR94] D. Schwabe and G. Rossi. From domain models to hypermedia applications: an object-oriented approach. In *Workshop on Hypermedia Design, ECHT'94*, 1994.
- [SR95] D. Schwabe and G. Rossi. Building hypermedia applications as navigational views of information models. In *Proceedings of the 28th Hawaii International Conference on System Sciences*, january 1995.
- [SRB96] D. Schwabe, G. Rossi, and S. D. J. Barbosa. Systematic hypermedia application design with oohdm. In *Proceedings of ACM Conference on Hypertext (Hypertext'96)*, Washington, DC, march 1996.
- [SRC95] L. F. G. Soares, N. L. R. Rodriguez, and M. A. Casanova. Nested composite nodes and version control in an open hyperemdia system. *Information Systems*, 20(6):501-519, 1995.
- [SRG97] D. Schwabe, G. Rossi, and A. Garrido. Reutilização de projetos em aplicações hipermídia. In *Anais do III Wokshop em Sistemas Multimídia e Hipermídia*, pages 29-40, 23-25, Maio 1997.
- [SS90] H. A. Schutt and N. A. Streitz. Hyperbase: A hypermedia engine based on a relational database management system. In *Proceedings of the European Conference on Hypertext (ECHT'90)*, pages 95-108, Versaille, France, November 1990.
- [SSKS] L. Shklar, A. Shet, V. Kashyap, and K. Shah. Inforharness: Use of automatically generated metadata for search and retrieval of heterogeneous information. Bell Communications Research Draft.
- [Str96] R. E. Streit. Acesso a bancos de dados relacionais via www: Estudo de alternativas e proposta de ferramentas. Master's thesis, Instituto de Informática - Universidade Federal do Rio Grande do Sul, 1996.
- [SW88] J. Smith and S. Weiss. An overview of hypertext. *Communications of the ACM*, July 1988.

- [Tea97] Alexandria Team. Alexandria digital library - 1997 annual report. The Alexandria Digital Library, University of California, Santa Barbara, CA (documento disponível via <http://alexandria.sdc.ucsb.edu/public-documents/annual-report/>), 1997.
- [VC94] A. J. Vanzyl and B. Cesnik. Open hypertext systems. Technical report, Department of Eletronics and Computer Science - University of Southampton, 1994. An Examination of Requirements and Analysis of Implementation Strategies, comparing Microcosm, HyperTED and the World Wide Web.
- [VCC96] S. R. Vasanthakumar, J. P. Callan, and W. B. Croft. Integrating inquiry with an rdbms to support text retrieval. *IEEE Computer Society - Bulletin of The Technical Committe on Data Engineering*, 19(1):24-33, March 1996.
- [WGMD95] S. Weibel, J. Godby, E. Miller, and R. Daniel. OCLC/NCSA metadata workshop report. http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_core_report.html, 1995.
- [Wil96] R. Wilensky. Toward work-centered digital information services. *IEEE Computer Magazine*, 29(5), May 1996.
- [WL92] U. K. Wiil and J. J. Leggett. Hyperform: Using extensibility to develop dynamic, open and distribued hypertext systems. In *Proceedings of the European Conference on Hypertext (ECHT'92)*, pages 251-261, Milano, Italy, 1992.
- [Yan94] L. Yang. A hypertext query language for images. *SIGMOD RECORD*, 23(1):16-20, March 1994.

