

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**A MODELAGEM DE DOCUMENTOS
ESTRUTURADOS MULTIMÍDIA
INTEGRANDO SISTEMAS DE
HIPERTEXTOS E ODA/ODIF**

Por
Celso Roberto Perez



SABi



Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. José Valdeni de Lima
Orientador

Porto Alegre, janeiro 1994

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Perez, Celso Roberto

A modelagem de Documentos Estruturados Multimídia integrando sistemas de hipertextos e ODA/ODIF / Celso Roberto Perez.- Porto Alegre: CPGCC da UFRGS, 1994.

177 p. il.

Dissertação (mestrado)-Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1994. Orientador: De Lima, José Valdeni

Dissertação: Documentos Estruturados, Multimídia, Hiperdocumentos, Hipertextos, Modelos de Dados.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Biblioteca da UFRGS

6340

PEREZ, CELSO ROBERTO

A MODELAGEM DE DOCUMENTOS
ESTRUTURADOS MULTIMIDIA
INTEGRANDO SISTEMAS DE
681.32.072(043)
P438M

INF
1994/60972-4
1994/04/04

UFRGS

MOD. 2.3.2
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

"A mis queridos abuelos,
el Tata (*in memoriam*)
y la Nona."

AGRADECIMENTOS

Não posso classificar sob nenhum critério a ordem de menção das pessoas a quem gostaria de agradecer, porém desejo expressar meu agradecimento em primeiro lugar à estrutura acadêmica brasileira, que me aceitou através do curso de Pós-Graduação em Ciência da Computação da UFRGS e pelo apoio financeiro do convênio PEC/PG da CAPES.

Ao professor José Valdeni de Lima pela orientação, compreensão, paciência e amizade.

À professora Lia G. Golendziner, pelas valiosas contribuições e sugestões.

Ao corpo docente e funcionários do CPGCC, pela eficiência e simpatia com que sempre me atenderam.

Aos meus colegas do CPGCC e amigos da Casa do Estudante, pela amizade e companheirismo, e que ajudaram a me sentir tão cómodo neste país.

Aos meus pais, por serem maravilhosos e pela paz e apoio moral.

À Teresa, minha grande inspiração, pelas horas e esforços dedicados, e pela constante companhia, sem a qual tudo teria sido mais difícil.

Ao Sandi e Flavio, pela amizade e ajuda na conclusão deste trabalho.

À Deus que me iluminou em todas as minhas decisões, angústias e momentos difíceis.

SUMÁRIO

LISTA DE ABREVIATURAS	10
LISTA DE FIGURAS	11
LISTA DE TABELAS	13
RESUMO	14
ABSTRACT	16
1 INTRODUÇÃO	18
1.1 Motivação	18
1.2 Objetivos	21
1.3 Organização do trabalho	22
2 A GERÊNCIA DE DOCUMENTOS ESTRUTURADOS	
MULTIMÍDIA	24
2.1 Introdução	24
2.2 Recuperação de Informação (<i>Information Retrieval (IR)</i>)	26
2.3 Gerência de Documentos Estruturados Multimídia	28
2.3.1 Principais Funções	30
2.3.1.1 Criação de Documentos Estrut. Multimídia	31
2.3.1.2 Armazenamento dos Documentos	33
2.3.1.3 Manipulação dos Documentos	34
2.3.1.4 Recuperação de Documentos	35
2.3.1.5 Transmissão de Documentos	37
2.4 A Modelagem de Documentos Estruturados Multimídia	38
3 ODA/ODIF E A MODELAGEM DE DOCUMENTOS	
ESTRUTURADOS MULTIMÍDIA	41
3.1 Introdução	41
3.2 Arquitetura do documento	42

3.2.1 Tipos e Classes de Objetos em ODA	44
3.2.2 Tipos e Classes de Documentos em ODA	48
3.2.3 Arquitetura total dos Documentos ODA/ODIF	49
3.3 Modelo de processamento de documentos	51
3.4 Arquiteturas de conteúdo dos documentos	53
3.4.1 Arquitetura de Conteúdo Caracter	54
3.4.2 Arquitetura de Conteúdo Gráfico <i>Raster</i>	54
3.4.3 Arquitetura de Conteúdo Gráfico Geométrico	54
3.5 Formato de intercâmbio de documentos (ODIF)	54
3.6 Conclusões	56
3.6.1 Porque ODA/ODIF para a Gerência de Documentos Estruturados Multimídia (Aspectos Positivos)	57
3.6.2 Que falta em ODA/ODIF? (Aspectos Negativos)	58
4 SISTEMAS DE HIPERTEXTOS E A GERÊNCIA DE DOCUMENTOS ESTRUTURADOS MULTIMÍDIA	61
4.1 Introdução	61
4.2 O quê é hipertexto?	62
4.2.1 Evolução dos sistemas de hipertextos	66
4.2.2 Alguns problemas nos sistemas de hipertextos	67
4.2.2.1 Desorientação	67
4.2.2.2 Treinamento	68
4.2.2.3 Falta de um padrão	69
4.3 O modelo de hipertextos	69
4.3.1 Abstrações de Baixo Nível	71
4.3.1.1 Nós	71
4.3.1.2 Botões ou Âncoras	72
4.3.1.3 Elos ou Ligações	72

4.3.2	Abstrações de Alto Nível	75
4.3.3	Operações	76
4.3.4	Relação com a arquitetura do sistema	76
4.3.5	Suporte de aspectos avançados	78
4.3.6	Alguns modelos existentes	79
4.3.6.1	Modelo HDM	79
4.3.6.2	HAM	80
4.3.6.3	Modelo de LANGE	81
4.3.6.4	O modelo de referência DEXTER	82
4.4	Conclusões	84
4.4.1	Modelos de hipertextos para a modelagem de Documentos Estrut. Multimídia (Aspectos Positivos)	85
4.4.2	Porque não só modelos de hipertextos (Aspectos Negativos)	85
5	O PARADIGMA DE ORIENTAÇÃO A OBJETOS COMO FERRAMENTA DE MODELAGEM	87
5.1	Introdução	87
5.2	Modelo de dados Orientado a Objetos	88
5.3	Características	90
5.3.1	Objetos Complexos	90
5.3.2	Identidade de Objetos	90
5.3.3	Encapsulamento	91
5.3.4	Classificação	91
5.3.5	Herança	91
5.4	Porque utilizá-lo na modelagem de Documentos Estruturados Multimídia?	92

6 OHypA: O META-MODELO DE DOCUMENTOS ESTRUTURADOS MULTIMÍDIA PROPOSTO	93
6.1 Introdução	93
6.2 OHypA e sua relação com a arquitetura do sistema de Gerência de Documentos Estruturados Multimídia	94
6.3 Estrutura lógica básica (Ponto de partida)	99
6.4 Modelagem de referências internas	101
6.5 Representação de referências externas	106
6.6 Representação de Anotações e/ou Comentários	110
7 OHypA, ASPECTOS AVANÇADOS	115
7.1 Introdução	115
7.2 Visões em Documentos Estruturados Multimídia	116
7.3 Tratamento de Versões	127
7.3.1 Relacionamento entre versões	133
7.3.2 O Objeto Versão	135
7.3.3 Operações sobre versões	139
7.3.4 Operações/Consultas sobre a estrutura de versões	140
7.3.5 Atualização automática de relações hipertextuais	141
7.3.6 Gerência de Configurações	149
7.3.7 Propagação de modificações	151
7.4 O meta-modelo final	153
7.5 Abstrações do meta-modelo	157
7.5.1 Abstrações de Baixo Nível	157
7.5.2 Abstrações de Alto Nível	158
7.5.3 Tipos de Ligações (Elos)	158
8 DISCUSSÃO E CONCLUSÕES	160
8.1 Discussão	160

8.2 Contribuições do presente trabalho	163
8.3 Trabalhos futuros	164
ANEXO A-1 CLASSES DE OBJETOS DO META-MODELO	
OHypA	166
BIBLIOGRAFIA	170

LISTA DE ABREVIATURAS

DO²CE^M	Documentos Orientados a Objetos Complexos Estruturados Multimídia.
HAM	<i>Hypertext Abstract Machine</i>
HDM	<i>Hypermedia Design Model</i>
IR	<i>Information Retrieval</i>
ISO	<i>International Standard Organization</i>
ODA/ODIF	<i>Office Document Architecture and Interchange Format</i>
OHypA	<i>Office HyperDocument Architecture</i>
POO	Paradigma de Orientação a Objetos
SGBD	Sistema Gerenciador de Banco de Dados
SGBDOO	Sistema Gerenciador de Banco de Dados Orientado a Objetos
SGML	<i>Standard Generalized Markup Language</i>
SQL	<i>Standard Query Language</i>

LISTA DE FIGURAS

Figura 1.1	Plano de Apresentação da dissertação	22
Figura 2.1	Modelo geral de um sistema de macrotexto	25
Figura 2.2	Modelo tradicional de recuperação de informação	27
Figura 3.1	Estruturas definidas em ODA/ODIF	44
Figura 3.2	Estrutura lógica de um documento segundo ODA/ODIF	46
Figura 3.3	Estrutura de apresentação dos documentos segundo ODA/ODIF	47
Figura 3.4	Hierarquia de classes de objetos em ODA/ODIF	48
Figura 3.5	Modelo da arquitetura total dos documentos ODA/ODIF	50
Figura 3.6	Exemplo da estrutura lógica genérica de um documento do tipo "Artigo"	51
Figura 3.7	Modelo do processamento de documentos em ODA/ODIF	53
Figura 4.1	Estrutura básica de hipertextos. Exemplo de um trecho de dicionário	62
Figura 4.2	Correspondência entre a tela e o banco de dados do hipertexto	65
Figura 4.3	Hipertextos com "ordem"	66
Figura 4.4	Hipertextos em estrutura de rede	66
Figura 4.5	Abstrações de baixo nível (nó, elo, âncora) e seu relacionamento	72
Figura 4.6	Elos "mover-a"	73
Figura 4.7	Elo de expansão (<i>Zoom</i>). Exemplo de uma anotação	74
Figura 4.8	Elos de índices	74
Figura 4.9	Elos estruturais	75
Figura 4.10	Arquitetura dos sistemas de hipertextos (segundo DEXTER)	77
Figura 4.11	Abstrações definidas em HDM	80
Figura 4.12	Exemplo de uso de "Especificações de Apresentação"	83

Figura 6.1	Arquitetura do sistema de Gerência de Documentos Estruturados	
	Multimídia	96
Figura 6.2	Níveis no meta-modelo OHypA	97
Figura 6.3	Níveis do meta-modelo OHypA	98
Figura 6.4	Estrutura lógica genérica de um documento segundo ODA/ODIF	100
Figura 6.5	Exemplo da estrutura lógica genérica de um documento	101
Figura 6.6	Referências Internas no meta-modelo OHypA	104
Figura 6.7	Representação das referências internas segundo OHypA	105
Figura 6.8	Meta-modelo OHypA com representação de referências externas	109
Figura 6.9	Referências Externas segundo o meta-modelo OHypA	110
Figura 6.10	Anotações e/ou Comentários no meta-modelo OHypA	113
Figura 6.11	Exemplo da representação de anotações segundo OHypA	113
Figura 7.1	Visões em documentos estruturados multimídia	118
Figura 7.2	Meta-modelo OHypA, incluindo representações de visões	124
Figura 7.3	Exemplo de representação de visões segundo OHypA	125
Figura 7.4	Versões em documentos estruturados multimídia	131
Figura 7.5	Estrutura formada pelos objetos Versão, mediante as relações de derivação e de correlação	134
Figura 7.6	Meta-modelo OHypA, incluindo tratamento de versões	139
Figura 7.7	Tratamento de versões segundo o meta-modelo OHypA	146
Figura 7.8	Configurações de objetos compostos em OHypA	151
Figura 7.9	O meta-modelo OHypA final	153
Figura 7.10	Classes de objetos definidas em OHypA e pelos usuários	154
Figura 7.11	Meta-modelo OHypA e objetos definidos pelos usuários segundo a metodologia de Rumbaugh	156
Figura 7.12	Tipos de ligações definidas em OHypA	159

LISTA DE TABELAS

Tabela 3.1 Possíveis valores do atributo "Gerador para Subordinados"	51
--	----

RESUMO

Há um grande número de aplicações que necessitam manipular documentos. Tal manipulação exige uma gerência dos mesmos nas tarefas de criação, armazenamento, recuperação e transmissão. Estas tarefas devem levar em conta características e aspectos inerentes aos documentos tais como estrutura lógica, estrutura de apresentação e hiperestrutura formada pelas referências internas e externas existentes nos documentos envolvidos.

A multimídia estabelece novos requisitos para os sistemas de gerência de documentos estruturados. Gráficos, sons, e imagens contêm informações que enriquecem o conteúdo textual tradicional dos documentos, podendo ser potencialmente explorados pelos usuários em um processo de formulação de consultas e busca de documentos.

A complexidade das aplicações que manipulam documentos estruturados e do tipo multimídia exige o apoio de modelos capazes de expressar características semanticamente mais ricas. Desta forma, tais modelos devem permitir modelar as seguintes estruturas: lógica, de apresentação e hiperestrutura.

A adoção de um modelo conceitual de documentos é fator determinante nas possibilidades oferecidas para consultas e recuperação dos referidos documentos.

Para a especificação e definição de tal modelo, foram consideradas duas possibilidades: i) utilização dos sistemas de hipertextos, nos quais a ênfase é fornecer um corpo estruturado de objetos com ligações conectando objetos relacionados. Aqui, a estrutura está projetada especificamente para auxiliar aos leitores a navegar através da informação, ii) uso do padrão de documentos eletrônicos ODA/ODIF, o qual dá ênfase à composição e controle da forma dos documentos, preocupando-se, também, com a divisão *layout-estrutura-conteúdo* dos mesmos, mas faltando nele um tratamento específico em relação às características hipertextuais.

Neste trabalho se considera que a integração destas duas filosofias permitirá, de uma maneira natural, a modelagem de Documentos Estruturados Multimídia. As pesquisas e propostas para combinar estas duas opções foram escassas no passado e se desconhecem trabalhos deste tipo visando a Gerência de Documentos Estruturados Multimídia.

Como resultado desta integração, no presente trabalho é definido e especificado o meta-modelo OHypA (*Office HyperDocument Architecture*), que pode ser

considerado como uma extensão do padrão ODA/ODIF. Tal meta-modelo terá uma aplicação real e prática pela combinação da tecnologia de hipermídia e a representação de documentos do tipo ODA.

Assim sendo, uma vez que o presente trabalho trata a modelagem de Documentos Estruturados Multimídia, envolvendo as áreas definidas, ele permitirá o estudo e delineamento de possíveis soluções para problemas comuns resultantes da integração das referidas áreas.

Finalmente, serão integrados dois enfoques em pleno desenvolvimento, resultando num meta-modelo orientado a objetos, que será facilmente integrado a um Sistema de Banco de Dados Orientado a Objetos.

PALAVRAS CHAVE: Documentos Estruturados, Multimídia, Hiperdocumentos, Hipertextos, Modelos de Dados.

TITLE: "THE MODELING OF MULTIMEDIA STRUCTURED DOCUMENTS INTEGRATING HYPERTEXTS SYSTEMS AND ODA/ODIF".

ABSTRACT

There is a great number of applications that need a document manipulation. Such manipulation demands a management of these documents on tasks like creation, storage, recovery and transmission. Such tasks should consider marks and aspects inherent on documents such as logical structure, presentation structure and hyperstructure, shaped by internal and external references existent in the documents involved.

Multimedia establishes new requirements for management systems of structured documents. Graphics, sounds and images contain informations which increase the traditional textual content from the documents that might be potentially explored by user in a process of formulation queries and search for documents.

The complexity of the applications that manipulate structured documents and the kind of multimedia, demands support from models which are able to express characteristics semantically richer. This way, such models must allow the modeling of the following structures: logical, presentation and hyperstructure.

The adoption of a respected model of documents is a main factor on the offered possibilities for queries and recovering of the reported documents.

For specification and definition of such model, two possibilities were considered: i) utilization of hypertexts systems, in which the emphasis is to provide a structured body of objects connecting other objects related. Here, the structure is designed specifically to help readers to navigate through the information, ii) application of the electronic documents model ODA/ODIF, which emphasizes the composition and control of the documents shape, worries also about its sharing of layout-structure-content, but lacks a specific treatment in relation to hypertextual characteristics.

This work considers that the integration of these two philosophies will allow, in a natural way, the modeling of multimedia structured documents. The researches and proposals to connect these two options used to be rare in the past. Besides, works that emphasizes the Management of Multimedia Structured Documents are unknown.

As a result of such integration, this work describes precisely the OHypA metamodel (Office HyperDocument Architecture) which might be considered as an extension of the ODA/ODIF model. This metamodel will have a real and practical application with the connection of hypermedia technology and the representation of ODA documents.

This way, since this work is about the modeling of Multimedia Structured Documents, covering the described areas, it'll allow the study and outline of possible solutions to common problems that come from the integration of such areas.

Finally, two topics in development will be integrated, resulting in an object-oriented metamodel, that should be easily integrated in an Object-Oriented Data Base System.

KEYWORDS: Structured Documents, Multimedia, HyperDocuments, Hypertexts, Data Models.

1 INTRODUÇÃO.

1.1 Motivação.

No marco do projeto DO²CE^M (Documentos Orientados a Objetos Complexos Estruturados Multimídia), entende-se como Gerência de Documentos Estruturados Multimídia o armazenamento, manipulação, recuperação e transmissão, por parte de uma comunidade de usuários, de documentos contidos num banco de documentos.

Tal sistema deverá oferecer tanto as capacidades dos formadores de documentos (definição do formato e composição dos documentos, auxílio na criação de novos documentos, facilidades de transmissão, etc.) quanto as capacidades dos sistemas de hipertextos (possibilidade de navegação, consultas, conteúdo multimídia, etc.).

O primeiro passo para tratar tal sistema é considerar e definir o seu modelo subjacente. Mas, neste ponto surge a seguinte pergunta: como deverá ser tal modelo, de forma a permitir capacidades totalmente heterogêneas?

Segundo a literatura consultada, os modelos de hipertextos propostos [CAM88, CON87, DEL87, FUR90, HAL90, LAN90, NEM93, RAD91] fornecem:

- Um corpo estruturado de objetos, com relações que ligam componentes.
- Uma estrutura projetada especificamente para auxiliar aos leitores a navegar pelo espaço de informações.
- Facilidades para definir **relações lógicas** entre partes de informação, mas não é fácil definir a **composição** das mesmas.

Ou seja, com estes modelos o problema original é só parcialmente solucionado.

Por outro lado, muitos trabalhos tem tratado o problema da **estruturação e composição** de documentos [BRO89, CAM91, FUR89a, FUR89b, HOR85, LIM90, LIM91, MUS89, QUI92], nos quais pode se identificar:

- Estruturas desenvolvidas para sistemas interativos de preparação de documentos, cujo objetivo principal é produzir um documento impresso.
- Estruturas também projetadas para fornecer ajuda inteligente na edição, ou para controlar que o documento cumpra um estilo requerido.
- Os documentos são considerados desde um ponto de vista **Estático**, não consideram o processamento dele dentro de um banco de documentos.

Todas estas características podem ser reunidas e tratadas considerando o padrão para documentos eletrônicos ODA/ODIF

Como pode-se concluir, este padrão também auxilia parcialmente no problema original, no que se refere à **Composição e Controle** da forma dos documentos.

De acordo com as características identificadas nos sistemas de hipertextos e no padrão ODA/ODIF, se considera neste trabalho que a integração destas duas filosofias permitirá, de uma maneira natural, a modelagem de Documentos Estruturados Multimídia. As pesquisas e propostas para combinar estas duas visões foram escassas no passado, sendo atualmente objeto de profundos estudos.

Considerada a integração destas duas filosofias como uma solução para a modelagem de Documentos Estruturados Multimídia, surge agora a seguinte pergunta: como levar a cabo tal integração?

Para respondê-la, será proposto o paradigma de orientação a objetos como ferramenta de modelagem dos conceitos considerados. Vários motivos justificam tal escolha:

- Porque trata os aspectos **Estáticos** (estrutura) e **Dinâmicos** (comportamento) de forma integrada.
- Porque é mais adequado para o tratamento de objetos complexos e dinâmicos, por possuir maior naturalidade conceitual.
- Porque são desejáveis conceitos tais como:
 - Classificação.
 - Herança.
 - Reusabilidade-Modularidade.
 - Encapsulamento.
 - Agregação-Generalização.
 - Associação.

Assim sendo, uma vez que este trabalho estará tratando a modelagem de documentos estruturados multimídia, que envolverá as áreas definidas anteriormente, ele permitirá o estudo e delineamento de possíveis soluções para problemas comuns resultantes da integração das referidas áreas.

1.2 Objetivos.

Como resultado das motivações apresentadas na seção anterior, o presente trabalho terá os seguintes objetivos:

-Fazer um estudo das características dos sistemas de Gerência de Documentos Estruturados Multimídia, dando principal atenção ao modelo subjacente.

-Fazer um estudo do padrão ODA/ODIF, como base para o modelo orientado a objetos que será integrado.

-Considerar o modelo de "nós-ligações" que o enfoque de hipertextos propõe e exportá-lo ao modelo a integrar.

-Considerar o uso da estrutura de objetos como uma interpretação funcional de hipertextos.

-Definir e especificar um meta-modelo, resultado da integração, e que possa ser considerado tanto uma extensão do padrão ODA/ODIF quanto uma proposta de modelo de hiperdocumentos.

Tal meta-modelo terá uma aplicação real e prática pela combinação da tecnologia de hipermídia e a representação de documentos do tipo ODA. O referido meta-modelo deverá permitir:

- O intercâmbio parcial ou total de um hiperdocumento.
- A integração dos documentos do tipo ODA em um banco de hiperdocumentos.
- A definição de ligações lógicas arbitrárias.

- A definição de classes e subclasses de documentos.
- A definição de visões de diferentes usuários sobre os documentos.
- O tratamento de versões de documentos.
- A manipulação de anotações e/ou comentários sobre os documentos.
- A definição de atributos de autoria e segurança.

-Finalmente, integrar dois enfoques em pleno desenvolvimento, obtendo um modelo orientado a objetos, que será facilmente integrado a um Sistema de Banco de Dados Orientado a Objetos.

1.3 Organização do trabalho.

A figura 1.1 resume a organização da dissertação, refletindo as sucessivas etapas da pesquisa.

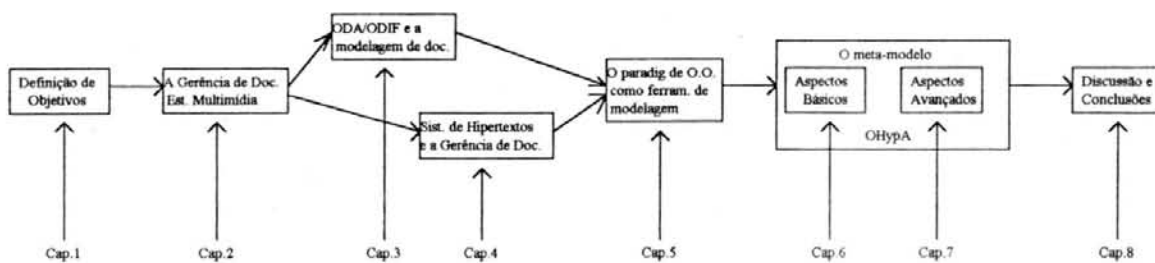


Figura 1.1: Plano de apresentação da dissertação.

No capítulo 2, apresenta-se a Gerência de Documentos Estruturados Multimídia, identificando-se as principais funções e características de tais sistemas. É dada uma especial atenção à modelagem dos documentos estruturados, definindo-se os aspectos considerados importantes nesta tarefa, que é o objeto de estudo neste trabalho.

O capítulo 3 introduz brevemente o padrão de documentos eletrônicos ODA/ODIF, o qual pode ser considerado como uma alternativa para a modelagem de documentos estruturados multimídia. Neste capítulo, também são identificados tanto os aspectos positivos (características desejáveis a ser adotadas) quanto os aspectos negativos do referido padrão.

O capítulo 4 trata dos sistemas de hipertextos e de sua possível relação com a Gerência de Documentos Estruturados Multimídia. Ou seja, neste capítulo eles são considerados e analisados de acordo com as características identificadas no capítulo 2. Aqui também são identificados os aspectos positivos e negativos que tais sistemas oferecem para a modelagem de documentos estruturados multimídia.

Analisadas as características tanto do padrão ODA/ODIF quanto dos sistemas de hipertextos, no capítulo 5 é considerado o paradigma de orientação a objetos, que será utilizado como ferramenta de modelagem para a integração das tais características, resultando no meta-modelo proposto OHypA.

Nos capítulos 6 e 7, especificamente, será feita a proposta do meta-modelo OHypA, sendo que enquanto no capítulo 6 são consideradas as características básicas, no capítulo 7 tratarão-se os aspectos avançados do mesmo.

Finalmente, no capítulo 8 são colocadas as conclusões e são analisadas tanto as contribuições quanto os futuros trabalhos resultantes da presente dissertação.

2 A GERÊNCIA DE DOCUMENTOS ESTRUTURADOS MULTIMÍDIA.

2.1 Introdução.

Há um grande número de aplicações que necessitam manipular documentos. Tal manipulação exige uma gerência dos mesmos nas tarefas de criação, armazenamento, recuperação e transmissão.

Estas tarefas devem levar em conta características e aspectos inerentes aos documentos tais como estrutura lógica, estrutura de apresentação e hiperestrutura formada pelas referências internas e externas existentes nos documentos envolvidos

A navegação baseada nesta organização parece ser apropriada como um meio de recuperação que se adapta perfeitamente com a filosofia dos Sistemas de Hipertextos.

Em outras palavras, a gerência dos bancos de documentos, levando em conta a filosofia dos sistemas de hipertextos pode ser considerado como um sistema de hipertexto de grande volume ou Macrotexto, segundo a classificação de Rada [RAD91].

Em tais sistemas, geralmente, diferentes pessoas criam documentos para incorporá-los ao banco de documentos, e uma instituição ou organismo servidor garante a manutenção do sistema de macrotexto, pois deve assegurar a disponibilidade de tais informações para diferentes usuários do sistema. Este processo é ilustrado na Figura 2.1.

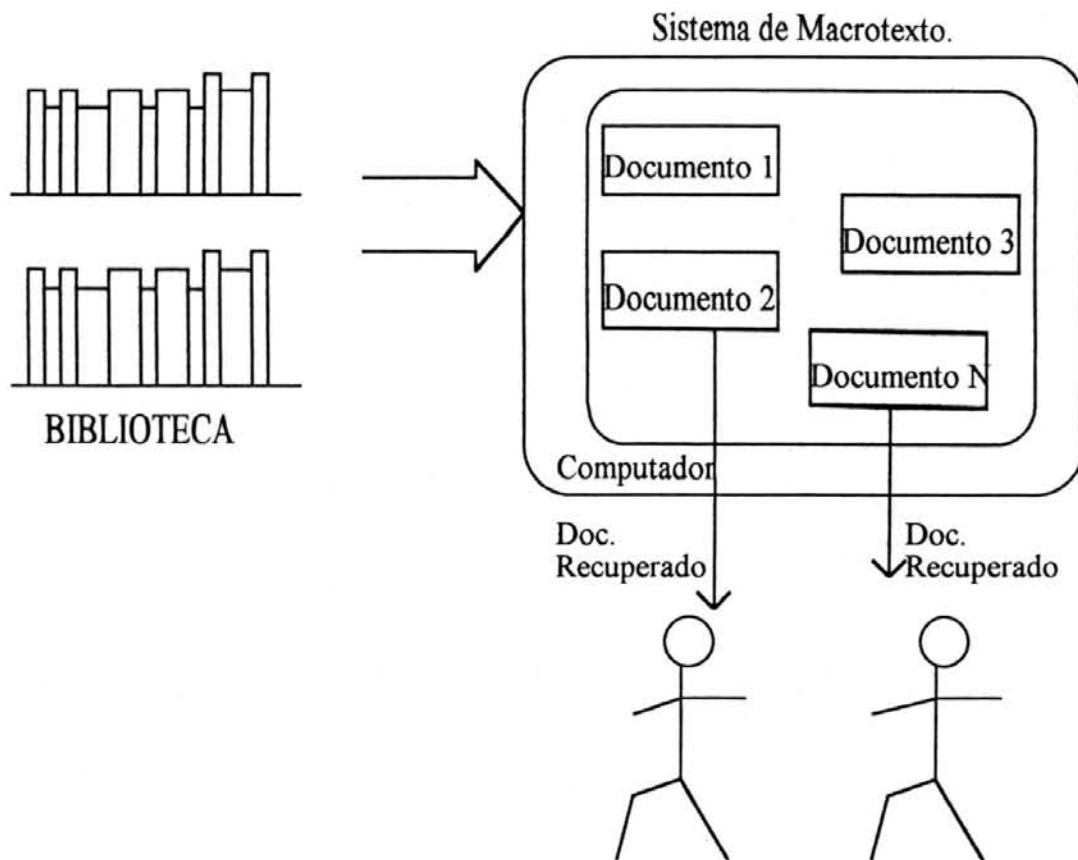


Figura 2.1: Modelo geral de um sistema de macrotexto. Os documentos de uma biblioteca são introduzidos no computador e usuários recuperam os mesmos do sistema.

Tradicionalmente as funções de armazenamento e recuperação de muitos documentos foram referidas com o termo "armazenamento e recuperação de informação".

A seguir neste capítulo, serão apresentados em forma resumida os princípios e características da recuperação de informação. Logo após, será tratada a Gerência de Documentos Estruturados Multimídia, como uma evolução dos sistemas de recuperação de informação, considerando as maiores diferenças entre estes sistemas.

Finalmente, será tratado o problema da modelagem dos Documentos Estruturados Multimídia manipulados por tais sistemas, considerando suas características e

aspectos. Este problema é o objetivo do presente trabalho, e no decorrer dos próximos capítulos, se apresentará e justificará uma proposta que considera tal modelagem.

2.2 Recuperação de Informação (*Information Retrieval (IR)*).

O surgimento dos computadores e o desenvolvimento de dispositivos de armazenamento eficientes e baratos originaram um desequilíbrio na Ciência da Informação.

As capacidades atuais disponíveis para armazenar tal informação excede amplamente a quantidade que pode ser organizada e recuperada eficientemente. Por isto, são necessários novos métodos para representar, organizar e recuperar a informação.

Como expressado anteriormente, pode-se coincidir com a definição que Salton [SAL83] fez da recuperação de informação. Ele diz que "A recuperação de informação está preocupada com a representação, armazenamento, organização e acesso de itens de informação".

Os sistemas de recuperação de informação geralmente consistem de um grande banco de dados textual, de um conjunto de palavras ou frases que descrevem o conteúdo dos documentos armazenados (termos indexados), os quais podem ser selecionados de duas formas: uma através de um vocabulário de termos *standards*, e a segunda opção pela livre escolha a partir dos títulos ou resumos dos documentos. Tais termos indexados servem como descritores ou apontadores aos documentos que eles representam.

Por outro lado, a informação requerida pelo usuário é representada como uma consulta, a qual consiste de uma série de termos combinados de acordo com a estrutura da linguagem de consultas, freqüentemente baseada na Álgebra de Boole.

Com isto, o processo de recuperação consiste de uma igualdade ("*matching*"), onde a relevância de um documento é julgada pela "similaridade" entre o termo que é descritor de tal documento e a consulta de busca. Este processo, que representa o modelo tradicional de recuperação de informação é representado esquematicamente na figura 2.2.

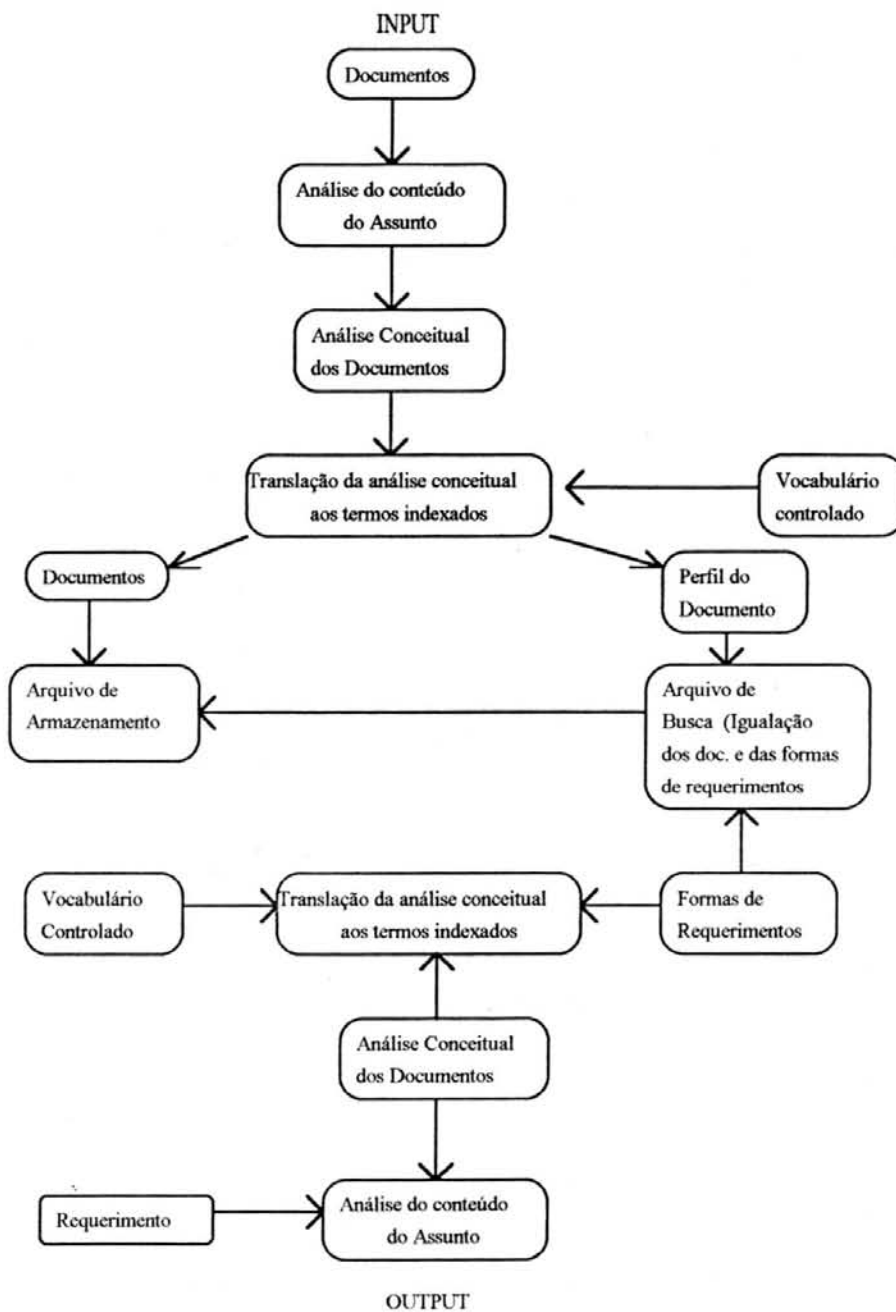


Figura 2.2 Modelo Tradicional de Recuperação de Informação.

Como foi mencionado anteriormente, uma das etapas do processo de recuperação de informação é a indexação, a qual refere-se à criação de uma estrutura de arquivo de índices, o qual contém termos que referenciam aos diferentes documentos do banco de dados. Tal processo de indexação pode ser feito tanto manualmente quanto automaticamente.

A indexação manual é considerada geralmente, muito difícil, pois as pessoas encarregadas desta tarefa não só devem ler e compreender todos os documentos, mas também devem lembrar o vocabulário de termos e as regras para a designação de termos.

Por outro lado, a indexação automática geralmente é baseada em análises estatísticas das palavras e frases usadas no texto do documento, utilizando métodos complexos.

A característica principal destes sistemas de recuperação de informação é sua complexidade, pois não só deve-se preocupar com um processo complicado e lento como é a indexação, mas também manter atualizada a estrutura de índices para poder permitir acesso eficiente, a que ocupa um espaço de armazenamento considerável.

2.3 Gerência de Documentos Estruturados Multimídia.

Texto é um tipo de dado completamente diferente de qualquer outro no que se refere a sua estrutura. Ele pode ser considerado, para o caso do armazenamento, como um grande "*string*"; mas para o caso da recuperação devem ser identificados "*substrings*" deste documento.

Além disto, um documento não só está formado por texto como pode conter outros tipos de informação (imagens, gráficos, som, etc.), que também têm uma estrutura

complexa para ser representada. Por isto, ainda que as capacidades de estruturação dos bancos de dados convencionais (relacionais, por exemplo) sejam satisfatórias para suportar estruturas complexas, elas não fornecem ferramentas apropriadas para a análises destes tipos de dados não convencionais.

Um sistema de Gerência de Documentos Estruturados Multimídia deve permitir ver um conjunto de documentos como um todo, um documento isoladamente e partes de um documento recuperadas como resultado de uma consulta.

Os bancos de dados sobre os quais aplicam-se estes sistemas devem ter capacidade de suportar definição de estruturas de textos, assim como de outros tipos de dados, recuperação de conjuntos de documentos, formatação de documentos recuperados, controle da terminologia dos mesmos e adição, modificação e remoção de documentos.

Aqui a palavra chave é "estruturação" da informação, pois, como foi visto na seção anterior, nos sistemas de recuperação tradicionais, a informação encontra-se no banco de dados em forma de ítems, os quais são relativamente não estruturados.

Por outro lado, a Gerência de Documentos Estruturados Multimídia visa considerar os documentos numa forma estrutural, tratando as relações entre componentes do mesmo documento (**INTRA**) e as relações entre componentes de diferentes documentos (**INTER**).

Pode-se dizer que a grande evolução que aconteceu desde os sistemas de recuperação de informação até os sistemas de Gerência de Documentos Estruturados foi devida à adição das características dos sistemas de hipertextos para melhorar as funções de criação, recuperação e navegação dos documentos contidos no banco de dados.

Outra melhoria notável será a modelagem e armazenamento dos referidos documentos num banco de dados orientado a objetos, o que permitirá solucionar problemas tais como a estruturação de tipos de dados não convencionais (textos, som, imagem, etc.), velocidade de recuperação e representação das ligações entre documentos.

2.3.1 Principais Funções.

Um sistema de Gerência de Documentos Estruturados Multimídia deve fornecer ferramentas que facilitem as atividades de manipulação de documentos num banco de documentos, algumas das quais são:

- a- Criar documentos novos ou como composição de outros já existentes.
- b- Criar, armazenar, exibir e manipular uniformemente texto, som, imagem e dados convencionais, sob a forma de objetos complexos.
- c- Criar ligações entre componentes de um mesmo documento (**intra**) ou entre vários documentos (**inter**) do banco.
- d- Possibilitar aos usuários criar diferentes visões de um mesmo documento.
- e- Identificar e manipular versões de documentos.
- f- Selecionar documentos por palavras-chaves armazenadas, por associação a outros documentos ou pelo conteúdo dos mesmos (por exemplo, reconhecimento de padrões nos casos de imagem ou som).
- g- Navegar pelas associações de documentos.
- h- Controlar o acesso aos documentos, mediante a definição de diferentes categorias de usuários, com seus correspondentes atributos.
- i- Auxiliar a monitoração da navegação através de diferentes ferramentas, tais como:
 - Estáticas: mapas, diagramas da estrutura, etc.

- Dinâmicas: caminhos percorridos, *backtracking*.

- Marcas: possibilidade do usuário ou o sistema marcar componentes e/ou relações para poder voltar a eles.

j- Possibilitar a adição de anotações, comentários e ligações pelos usuários, segundo sua categoria.

k- Fornecer capacidade de autoria cooperativa.

l- Definir atributos de segurança na autoria.

Para possibilitar tais capacidades, o sistema de Gerência de Documentos Estruturados Multimídia deve prover as seguintes funções necessárias:

1- Criação (autoria) de documentos estruturados.

2- Armazenamento dos mesmos no banco de documentos.

3- Manipulação dos documentos contidos no banco de documentos (versões, visões).

4- Recuperação de documentos levando em consideração requerimentos dos usuários (navegação, consultas).

5- Transmissão de documentos entre sistemas heterogêneos.

2.3.1.1 Criação de Documentos Estruturados.

A criação de documentos pode ser feita tanto por autores individuais quanto cooperativamente por várias pessoas. Tal autoria envolve as etapas de definição da estrutura lógica, definição da estrutura de apresentação e da edição do conteúdo de documento.

I - Definição da estrutura lógica do documento.

Através desta função, o usuário autorizado para tal pode definir a estrutura lógica de novas classes de documentos. Esta definição só precisa ser feita uma vez para cada classe; no momento da criação de um documento daquela classe, é criada uma instância da mesma, e a edição do conteúdo ficará subordinada às estruturas lógicas já definidas. Esta tarefa pode ser facilitada pela utilização de componentes básicos, como parágrafos, seções, capítulos, etc., para a definição da estrutura lógica das novas classes.

II - Definição da Estrutura de Apresentação.

Consiste em definir parâmetros para o documento, que irão definir a organização de seus elementos no momento da geração de seu formato final.

Formatação é a tarefa de gerar fisicamente um documento a partir da estrutura lógica do mesmo e de tais parâmetros.

Em outras palavras, formatar é gerar o "*layout*" dos objetos que pertencem ao documento em algum meio de apresentação gráfica.

Entre as funções de formatação pode-se citar: posicionamento de objetos em função da página física e de outros objetos exibidos; expansão ou redução de objetos; seleção das primitivas de exibição, como fontes de caracteres, tamanho, etc.

III - Edição do Conteúdo do Documento.

O sistema de Gerência de Documentos Estruturados Multimídia deve oferecer facilidades para a edição do conteúdo. Uma vez que o documento pode conter diferentes tipos de dados, um editor completo deve fornecer recursos para a edição de

qualquer desses tipos. Ou seja, deve-se dispor de um processador de textos simples (caracteres), um editor gráfico para a criação de figuras a partir de primitivas geométricas (segmentos, círculos, retângulos, etc.) e, ainda, a possibilidade de trabalhar com imagens digitalizadas.

Estes recursos devem estar integrados através de um bom sistema de *interface*. Esta tarefa é fortemente influenciada pela adoção de estruturas lógicas para os documentos, pois o sistema pode tanto facilitar quanto disciplinar a edição de elementos lógicos a partir de informações da estrutura.

2.3.1.2 Armazenamento dos Documentos.

Após o documento ter sido criado, o sistema deve fornecer meios eficientes de armazenamento permanente. Isto é feito levando em consideração a estrutura dos documentos, para logo após poder permitir as consultas e navegação dos mesmos. Neste ponto, é desejável modelar os documentos no banco de documentos como objetos, aproveitando todas as características e vantagens que oferecem estes sistemas.

O armazenamento de Documentos Estruturados Multimídia requer estruturas específicas que permitam:

- um rápido acesso.
- o compartilhamento numa rede de computadores.
- a associação com outros documentos.
- a evolução dinâmica do conteúdo e da estrutura.
- o sincronismo na recuperação de cada objeto.

A integração de tais documentos a um SGBD pode ser efetivada adotando-se uma das seguintes estratégias:

- Estender as funcionalidades internas de um SGBD já existente.
- Construir uma interface especializada em torno de um SGBD já existente.
- Construir um novo SGBD capaz de manipular objetos complexos, particularmente documentos estruturados.

É importante observar, entretanto, que em qualquer dos casos, deve-se dar atenção especial à definição da estrutura lógica dos documentos, pois esta exercerá influência significativa no grau de eficiência de armazenamento e das operações de consulta, assim como facilitar a associação de diretivas de apresentação.

2.3.1.3 Manipulação de Documentos.

A manipulação dos documentos estruturados multimídia armazenados no banco de documentos compreende, entre outras, as atividades de consulta e/ou modificação de tais documentos.

Estas atividades dependem de fatores tais como categoria de usuário, tipo de documento, nível de visibilidade associado ao documento, etc.

Este processo de modificação pode consistir de:

- Adição de **comentários e/ou anotações** a trechos de texto, os quais serão associados ao usuário que os realizou.

-Modificação do conteúdo do documento, seja eliminar, agregar ou modificar o texto original, em tal caso se estará gerando uma nova **versão** do documento. Esta operação só pode ser realizada pelo próprio autor do documento.

-Adição ou omissão de ligações entre componentes lógicos do mesmo ou de vários documentos, em cujo caso, se estará gerando novas **visões** do documento, as quais também são associadas ao usuário que as criou.

Para serem realizadas estas atividades, o sistema de Gerência de Documentos Estruturados Multimídia deverá ter a possibilidade de definir e tratar:

- Diferentes categorias de usuários, as quais terão associadas as funções que os mesmos podem fazer dentro do sistema.

- Aspectos de segurança, os quais estarão associados às diferentes categorias de usuários e às diferentes classes de documentos (segurança na autoria e modificação de documentos). Por exemplo, cada usuário pode definir o nível de visibilidade (pública, privada, grupal) de sua sessão de trabalho associada a um documento.

- A noção de tempo dentro do sistema, a qual cumpre um papel fundamental na atividade de modificação de documentos.

2.3.1.4 Recuperação de Documentos.

Esta função é a que diferencia os sistemas de Gerência de Documentos Estruturados Multimídia dos sistemas de recuperação de informação tradicionais (*IR*).

A forma de acessar e recuperar informação num sistema tradicional (*IR*) é através de consultas, utilizando a técnica de indexação, a qual é bastante limitada.

Por sua vez, a recuperação nos sistemas de Gerência de Documentos Estruturados Multimídia é uma atividade de extrema complexidade. Tal complexidade é gerada pela necessidade de se ter que atender, de forma transparente, aos requisitos individuais de consulta de cada tipo de dado multimídia (som, texto, imagem e dados tradicionais).

Um documento multimídia é formado por objetos que além de não estarem apresentados em formatos padronizados, estão armazenados, na maioria das vezes, em diferentes meios (magnéticos e óticos), organizados em distintas estruturas lógicas e físicas.

O problema principal da recuperação de documentos é de ordem conceitual e reside na dificuldade de defini-los semanticamente.

Tem se buscado nas associações entre dados formatados e não formatados, e nas ligações entre documentos (*hyperlinks*), soluções para a diminuição do "gap" semântico dos documentos multimídia.

Atualmente são usadas dois tipos de recuperação de documentos: **recuperação por conteúdo** e **recuperação por navegação** (*hyperlinks*) [BUM93].

A **recuperação por conteúdo** pode ser dada de diferentes formas: através da associação de palavras chaves que definam os documentos, pela busca da ocorrência de palavras no interior de textos, pelo reconhecimento de imagens, etc.

A outra forma de recuperação se fundamenta nos conceitos da hipermídia. Sistemas que fazem uso deste recurso devem também se preocupar com estruturas especiais de associação entre documentos e com recursos de auxílio à navegação no grafo de documentos [BUM93].

Esta navegação pode acontecer de três formas diferentes:

I- Por simples leitura ("reading"), a qual implica na leitura seqüencial tradicional, linha por linha do documento, desde a primeira até a última página.

II- Por busca ("searching"), que ocorre quando se sabe a identificação ou rótulo de alguma informação e se quer apenas tal informação específica.

III- Por folheio ("browsing"), na qual o usuário não está bem certo do que está querendo selecionar, ou seja, os limites do assunto pesquisado ainda não estão bem definidos. Neste caso, o usuário "pula" de um ponto a outro do documento, lendo alguns pequenos segmentos em cada trecho.

A navegação em sistemas de Gerência de Documentos Estruturados Multimídia possui mais de uma dimensão: a linear, usada para o movimento dentro de uma unidade lógica (parágrafo, seção, capítulo, etc.), e a não linear, usada para os "saltos de hipertextos". Por isso, estes sistemas de Gerência devem ajudar aos usuários a entender e localizar suas posições correntes, facilitando a navegação pelo hiperespaço.

2.3.1.5 Transmissão de Documentos.

Além da função de recuperação de documentos, segundo um determinado critério, o sistema de Gerência de Documentos Estruturados deve fornecer recursos de transmissão de documentos.

A transmissão de documentos é uma das maneiras comuns de transmissão de informação dentro de uma organização ou entre organizações.

Um documento pode ser transmitido tanto na forma de imagem quanto na forma de sua estrutura e conteúdo. No primeiro caso, na aplicação destino só é possível a impressão do documento, por exemplo sistemas de Fax. A segunda forma dá maior flexibilidade ao usuário, pois permite a reedição do documento. Assim, o receptor pode alterar o documento, se considerar necessário.

Para a transferência de documentos com sua estrutura lógica, é importante que tanto emissor quanto receptor adotem um padrão comum de estruturação dos documentos, como por exemplo ODA/ODIF ou SGML [ISO86, ISO89].

2.4 A modelagem de Documentos Estruturados Multimídia.

A multimídia estabelece novos requisitos para os sistemas de Gerência de Documentos Estruturados. Gráficos, sons e imagens contêm informações que enriquecem o conteúdo textual tradicional dos documentos, podendo ser potencialmente explorados pelos usuários num processo de formulação de consultas e busca de documentos.

Nos sistemas de recuperação de informação tradicionais, como foi visto, tal recuperação se dá, convencionalmente, por intermédio de palavras chaves ou por busca sequencial a palavras armazenadas em textos. Estes processos, aplicados na recuperação de documentos, não garantem a eficácia da recuperação, pois atendem às consultas de forma abrangente e não se preocupam com o contexto semântico das palavras.

Neste ponto, é preciso ressaltar a diferença básica entre um banco de dados e um banco de documentos. Um banco de dados é uma abstração do mundo real, constituindo-se de uma coleção de fatos extraídos da realidade e seguindo uma estrutura fixa e definida em algum esquema de banco de dados. Já um banco de documentos é uma

coleção de documentos que podem não seguir uma estrutura fixa definida no esquema, além de levar em conta os aspectos multimídia, estruturação e volume.

Um modelo considerado ideal para a representação dos documentos multimídia deve possuir, portanto, três estruturas conceituais básicas: uma, que defina a própria estrutura do documento, uma segunda que defina a forma como o documento é visualizado pelos usuários, e uma terceira que defina o contexto ao qual o documento está inserido, através do conteúdo armazenado.

A adoção de um modelo conceitual é fator determinante nas possibilidades oferecidas para consultas e recuperação dos referidos documentos estruturados multimídia.

O padrão ODA/ODIF é um exemplo de modelo para a arquitetura de documentos que se aproxima deste ideal, pois possui a preocupação em relação ao *layout* e a estrutura lógica, faltando um tratamento específico em relação às características hipertextuais.

Ou seja, o processo de modelagem tem que ser realizado levando em conta os seguintes aspectos:

a- Conteúdo Multimídia dos documentos. Ou seja, tais documentos não possuem só texto, outros tipos de informação (imagem, audio, gráficos, etc.) podem formar parte do conteúdo dos mesmos, os quais têm uma estrutura complexa para ser representada.

b- Estruturação dos documentos. Considerar os documentos em uma forma estrutural, tanto em nível de composição quanto em nível de relações lógicas.

c- Tratamento de Versões. Possibilidade de gerenciar tanto o histórico quanto diferentes configurações dos documentos.

d- Volume da informação. Por tratar-se de conteúdo multimídia, a quantidade de informação a ser manipulada é de tamanho razoável, o qual implica num tratamento especial.

e- O intercâmbio de documentos. Possibilidade de transmiti-los entre sistemas heterogêneos, segundo os padrões de intercâmbio existentes.

Esses aspectos conjuntamente com as áreas de Bancos de Dados, Formataadores de Documentos (ODA/ODIF) e Sistemas de Hipertextos, viabilizarão o estudo e delineamento de possíveis soluções para problemas comuns à modelagem de Documentos Estruturados Multimídia.

Como um primeiro passo para resolver os problemas mencionados acima, no presente trabalho será abordada tal modelagem dos documentos estruturados multimídia, segundo o paradigma de orientação a objetos, integrando conceitos do padrão de documentos ODA/ODIF e dos sistemas de hipertextos. Na referida abordagem será dada principal importância aos aspectos:

-Estruturação dos Documentos: - Composição

- Relações Hipertextuais.

-Autoria: - Visões de usuários dos documentos.

- Versões dos conteúdos dos documentos.

-Segurança: no sistema de Gerência de Documentos Estrut. Multimídia.

Nos próximos dois capítulos, serão tratados brevemente o padrão ODA/ODIF, e os sistemas de hipertextos, analisando-os de acordo com a sua adequação à modelagem de documentos estruturados multimídia. Depois disso, nos capítulos seguintes é proposto um meta-modelo resultado da integração dos mesmos.

3 ODA/ODIF E A MODELAGEM DE DOCUMENTOS ESTRUTURADOS MULTIMÍDIA.

3.1 Introdução.

Sistemas de escritório devem se integrar com outros sistemas de escritórios, em particular através de editores e sistemas de correio. Esta integração é mais fácil se todos os sistemas usam uma representação comum para seus documentos. Assim, para ser possível o intercâmbio de informações entre diferentes sistemas de escritórios, a *International Standard Organization (ISO)* desenvolveu o padrão ISO 8613 [ISO89] *Office Document Architecture and Interchange Format (ODA/ODIF)*, o qual define um formato padrão para documentos eletrônicos.

Neste padrão, os documentos de escritorio considerados são, por exemplo, memorandos, cartas, formulários e relatórios. Cada um deles são normalmente pequenos, ou seja não volumosos, e podem conter textos, figuras e tabelas. Este padrão se aplica na troca de documentos via comunicação de dados ou na simples troca dos meios de armazenamento.

O seu objetivo é permitir que a apresentação do documento seja feita da mesma forma em que ele foi criado e que ele possa ser editado ou reformatado. Além disso, ele permite que os documentos de troca possuam tipos diferentes de conteúdo (textos, gráficos, imagens e áudio).

ODA [HOR85 MUR89 MUS89 CAM91 e SHE91], adota o conceito de "classes" para agrupar os documentos que têm características similares.

O conjunto de classes de objetos no documento é, de fato, a estrutura genérica do documento. O conceito de classe de objeto simplifica a criação do documento,

melhora a eficiência de transmissão e preserva a integridade da estrutura do documento durante e depois da edição.

O padrão ODA não define regras para a estruturação dos documentos, mas fornece um esquema para que o criador faça isso. Ou seja, ODA não prescreve classes de objetos lógicos particulares (capítulos, seções, etc.) nem estruturas de documentos particulares. Em vez disso, ODA fornece um esquema dentro do qual se pode criar objetos, classes de objetos e estruturas; e uma vez criados, poder transmiti-los desde uma aplicação originadora a uma aplicação receptora.

Para obter tudo isto, ODA consiste de:

- 1 - Um modelo da Arquitetura do Documento.
- 2 - Um modelo do Processamento do Documento.
- 3 - Arquiteturas de Conteúdo.
- 4 - Um Formato de Intercâmbio de Documentos (ODIF).

Nas seções seguintes, serão apresentadas em maior detalhe cada uma destas partes.

3.2 Arquitetura do Documento.

Um documento ODA pode ser visto como uma árvore, cuja forma indica a estrutura do documento e cujas folhas contêm o seu conteúdo. A divisão entre estrutura e conteúdo facilita a criação e manuseio de documentos Multimídia.

A descrição de um documento em ODA é feita através de duas estruturas: estrutura lógica e estrutura de formatação ou apresentação.

Na primeira, o documento é estruturado com base no significado, ou seja, a estrutura lógica faz sentido para quem escreve e lê o documento. A estrutura lógica determina como os componentes lógicos, que formam uma estrutura de tipo árvore, se relacionam com o conteúdo do documento. Alguns exemplos de objetos lógicos são: capítulos, seções, figuras, parágrafos e diagramas.

Na segunda, a preocupação básica é com a aparência do documento, ou seja, como o documento é dividido e subdividido com base na apresentação. Esta estrutura associa o conteúdo do documento a uma hierarquia tipo árvore de objetos de apresentação ou disposição. Exemplos de objetos de apresentação são: páginas, quadros e blocos.

As duas estruturas, lógica e de apresentação, possuem uma dependência em nível de conteúdo, pois em ambas estruturas seus elementos são relacionados com as porções de conteúdo do documento.

Cada documento tem suas estruturas lógica e de apresentação específica (instância do documento). No entanto, existem estruturas genéricas (classe do documento), que controlam a criação daquelas estruturas, para cada estilo e classe de documento (Ver Fig. 3.1).

colocar importante

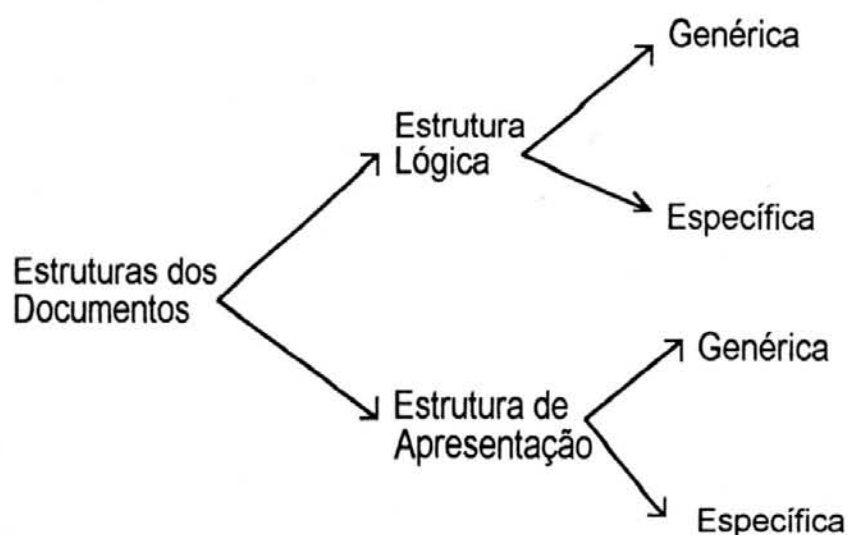


Fig. 3.1 Estruturas definidas em ODA/ODIF.

3.2.1 Tipos e Classes de Objetos em ODA.

Como já foi mencionado, ODA considera os documentos modelados como compostos por diferentes objetos. Tais objetos manipulados pelo padrão podem ser:

- **Básicos:** Pertencem ao nível mais baixo da hierarquia (folhas das árvores). Através deles é relacionada a estrutura ao conteúdo.

- **Compostos:** Consistem de componentes que podem ser outros objetos compostos e/ou objetos básicos.

Tanto um quanto o outro possuem um identificador único, um tipo associado e um conjunto de atributos designados, os quais podem ser propriedades (dimensão, posição, etc.) ou relações. Estas relações podem ser:

- **Hierárquicas:** as quais ligam ou relacionam os objetos para construir a estrutura tipo árvore, e são expressas pelos atributos "referência a objeto subordinado" (se o objeto é composto) ou "referência a porções de conteúdo" (se o objeto é básico). Estas

relações também podem ser chamadas "estruturais", pois representam a estrutura do documento.

-Lógicas-Lógicas: ligam objetos lógicos e estendem-se através da estrutura lógica.

-Disposição-Disposição: como as anteriores, mas estas ligam objetos de apresentação através da estrutura de apresentação.

-Lógica-Disposição: formam parte das chamadas "diretivas de apresentação", ou seja, como forma de referências a "*templates*" de apresentação, os quais definem cabeçalhos, colunas ou áreas de rodapé. Relacionam os objetos lógicos aos objetos de apresentação, definindo como será apresentado o objeto lógico.

Por outro lado, o tipo de um objeto designa atributos adicionais que podem ser aplicados a tais objetos. ODA distingue dois tipos de objetos: lógicos e de disposição ou apresentação.

I- Tipos de objetos lógicos:

-Documento. Representa a estrutura lógica total do documento.

-Objeto lógico composto. Representa objetos complexos, formados por outros objetos compostos e/ou básicos, por exemplo: capítulo, seção, tabela, figura, etc.

-Objeto lógico básico.Objetos folhas da hierarquia de composição, eles formam parte dos objetos compostos. Alguns deles são, por exemplo: parágrafo, item de tabela, nome, filiação, etc.

A Fig. 3.2 indica como estes tipos de objetos são combinados para formar a estrutura lógica.

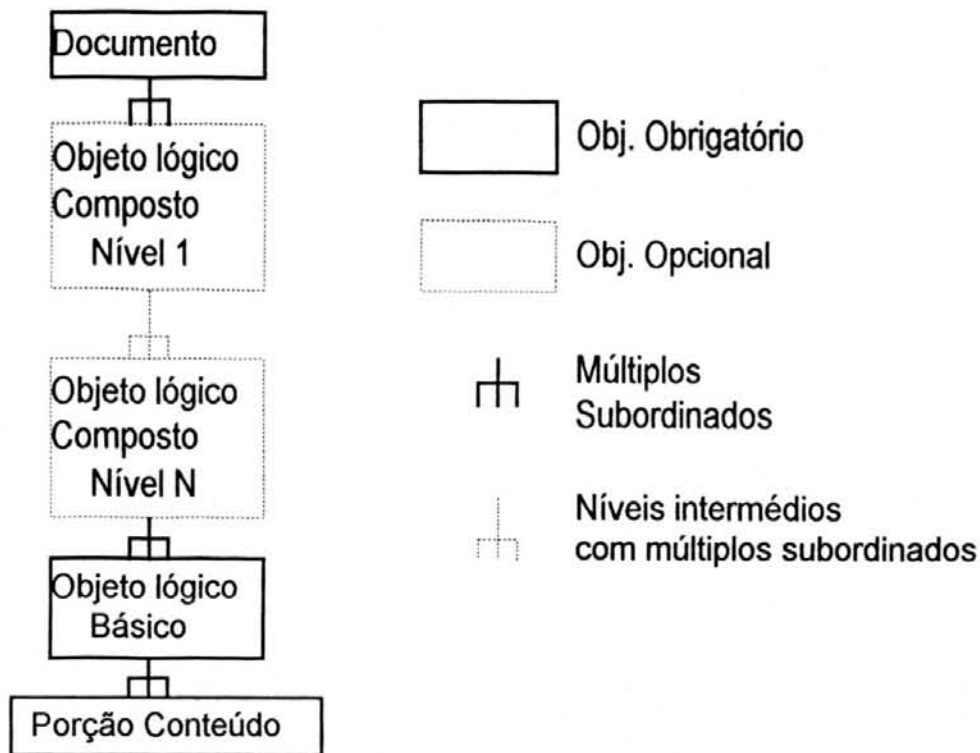


Fig. 3.2 Estrutura lógica de um Documento segundo ODA/ODIF.

II- Tipos de objetos de apresentação:

-**Documento**: Representa o conjunto total de páginas.

-**Conjunto de página**: abrange uma ou mais páginas e/ou um ou mais conjuntos de páginas subordinados, os quais necessitam ser identificados como um grupo.

-**Página composta**: quando o conteúdo dela é de diferentes arquiteturas de conteúdo.

-**Página básica**: quando o conteúdo é da mesma arquitetura de conteúdo.

-**Frame**: área de apresentação retangular dentro de uma página ou dentro de um *frame* de mais alto nível. Define limites dentro de uma página para a apresentação do conteúdo.

-**Bloco**: área retangular que contém uma ou mais porções de conteúdo da mesma arquitetura de conteúdo.

A Fig. 3.3 indica como estes objetos são combinados para construir a estrutura de apresentação.

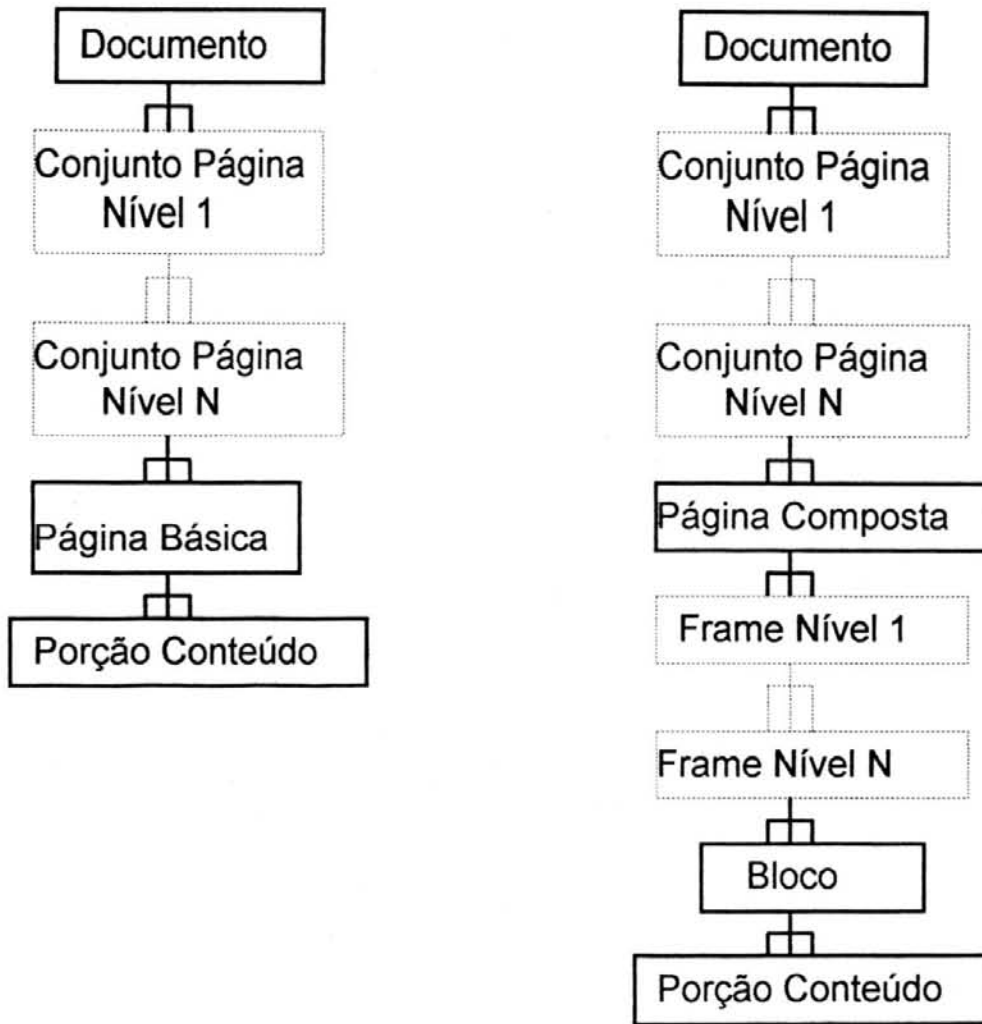


Fig. 3.3: Estrutura de Apresentação dos Documentos segundo ODA/ODIF.

Objetos do mesmo tipo com características adicionais comuns podem ser subagrupados em classes de objetos.

ODA originalmente define 9 classes de objetos, mas em [MUR89] define-se novas classes, formando a hierarquia representada na Fig. 3.4.

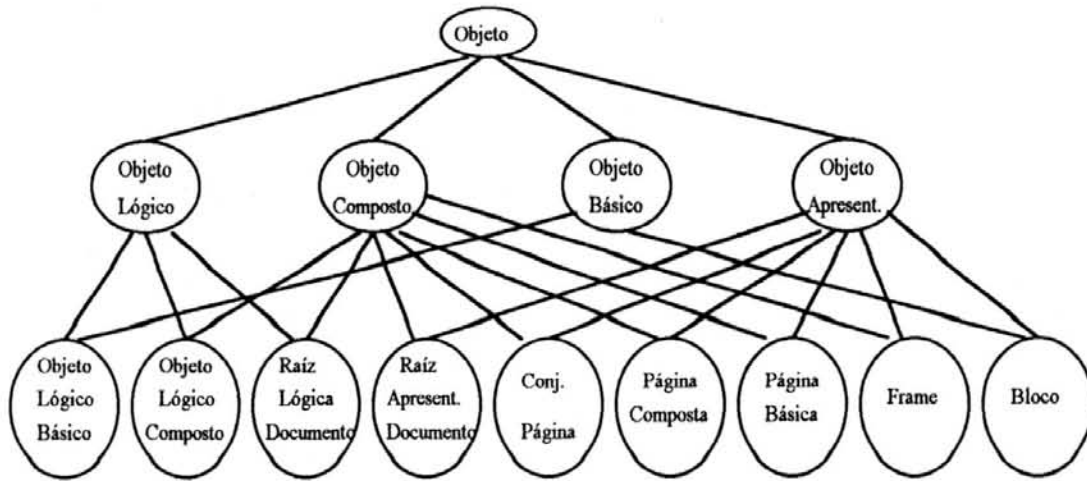


Fig. 3.4: Hierarquia de Classes de objetos em ODA/ODIF.

As classes de objetos também podem ser definidas pelo usuário, por meio das "Definições de Objetos", que são um conjunto de regras ou "*templates*" de acordo com os quais podem ser produzidos objetos das estruturas de documentos.

As classes definidas pelos usuários são subclasses de algumas das 9 classes definidas pelo ODA.

3.2.2 Tipos e Classes de Documentos em ODA.

Documentos similares podem ser agrupados em classes de documentos. Algumas podem ser, por exemplo:

- Cartas comerciais.
- Relatórios.
- Memorandos.
- Formulários.
- Artigos técnicos.

Da mesma maneira que ocorre nas classes de objetos, as classes de documentos não estão padronizadas. Elas podem ser definidas pela aplicação por meio de "Definições de Documentos". Tais definições tem que conter as definições de todos os objetos que podem ocorrer num documento da classe.

3.2.3 Arquitetura Total dos Documentos ODA/ODIF.

A representação total de um documento segundo ODA/ODIF deve conter:

- A **descrição do documento específico**, com :
 - a estrutura lógica.
 - a estrutura de apresentação.
 - o seu conteúdo.

-A **descrição da Classe do Documento**, com as definições dos objetos lógicos e/ou de apresentação (estruturas genéricas).

-O **Perfil do documento**, o qual contém informação para manipular o documento como um todo. Consiste de um conjunto de atributos no nível mais alto na estrutura do documento.

A Fig. 3.5 esquematiza o modelo da arquitetura total de um documento ODA/ODIF.

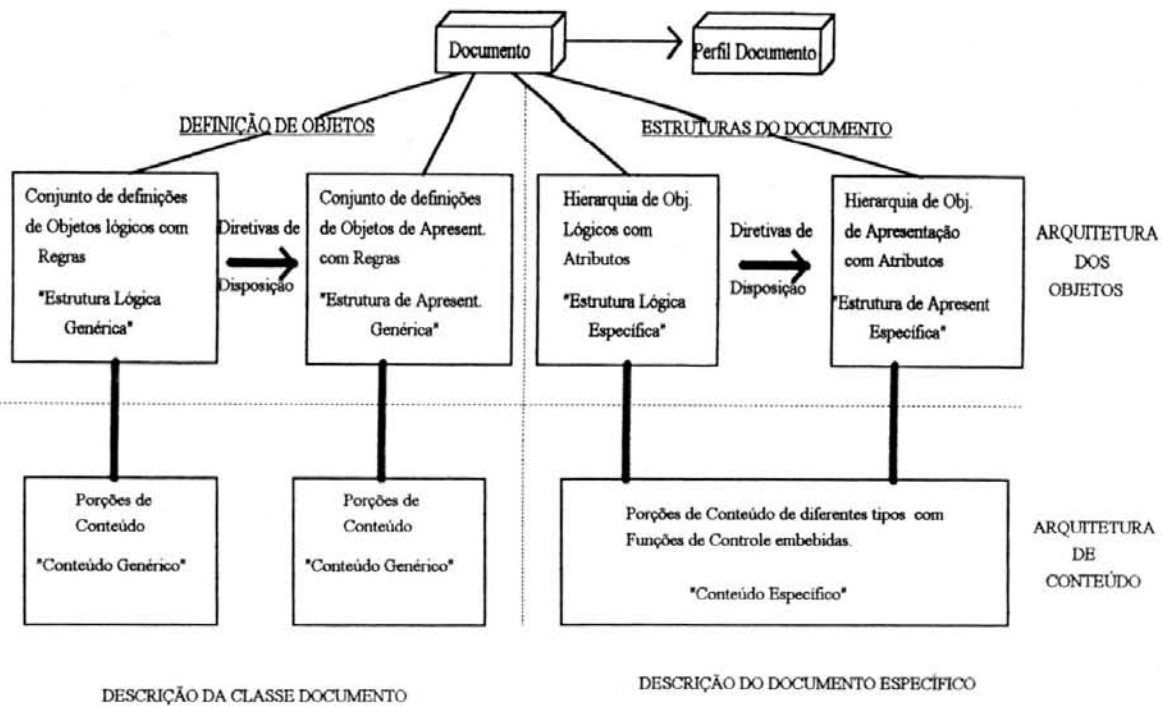


Fig. 3.5: Modelo da Arquitetura Total dos Documentos ODA/ODIF.

As regras para a estruturação de um documento são descritas pela associação de tais objetos com "descrições de classes".

As estruturas lógicas e de apresentação são hierarquias de objetos, onde cada objeto está descrito por um conjunto de atributos.

As estruturas genéricas consistem de dois conjuntos de descrições de classes de documentos (objetos lógicos e objetos de apresentação), que definem os tipos e combinações de objetos permitidas.

A descrição de classe de cada objeto não folha na estrutura inclui um atributo chamado "gerador para subordinados", o qual define como o objeto pode ser formado a partir dos objetos subordinados. Os possíveis valores que pode adotar tal atributo estão apresentados na tabela 3.1.

Tabela 3.1: Possíveis valores do atributo "Gerador para Subordinados".

OPT	Opcional	0 ou 1 Ocorrência
REQ	Requerido	1 Ocorrência
REP	Repetitivo	1 ou mais Ocorrências
OPT/REP	Opcional Repetitivo	0, 1 ou mais Ocorrências
SEQ	Seqüência	Ocorre na ordem especificada
AGG	Agregado	Ocorre em qualquer ordem
CHO	Opção	Só uma de um grupo de opções

Para concluir, apresenta-se na figura 3.6 um exemplo de definição de documento utilizando tais "Geradores para Subordinados".

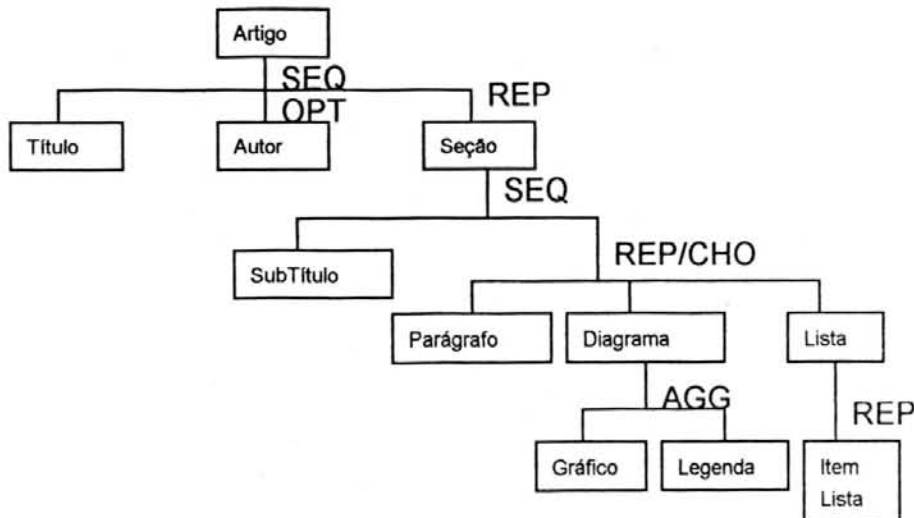


Fig. 3.6: Exemplo da Estrutura Lógica Genérica de um documento do tipo "Artigo".

3.3 Modelo de Processamento de Documentos.

As diferentes partes de um documento tipo ODA são envolvidas na manipulação de documentos através de três processos principais diferentes:

- **Processo de edição:** envolve a interação da edição do conteúdo e a edição da estrutura lógica. A edição da estrutura inclui a criação ou eliminação de objetos, mudanças em sua posição na estrutura, e modificações de valores de propriedades e atributos relacionais controlados pelas regras das definições lógicas. A orientação deste processo é feita usando-se a estrutura lógica genérica do documento.

- **Processo de disposição ou apresentação:** Envolve a interação da geração da estrutura de apresentação e a disposição do conteúdo. A geração da estrutura de apresentação cria *frames* e blocos aninhados ou superpostos sobre conjuntos de páginas e coloca o conteúdo dos objetos lógicos básicos nos blocos. Isto é controlado pelas diretivas de disposição dos objetos lógicos. O processo se utiliza das estruturas lógicas e de apresentação específicas e genéricas.

- **Processo de exibição:** diz respeito à apresentação do documento em uma forma adequada para o usuário. Este processo utiliza a estrutura de apresentação e mostra, em um meio adequado, uma versão visualizável do documento.

Como ODA é um padrão para armazenamento e intercâmbio de documentos, é importante a separação destes dois últimos processos, pois o armazenamento e a transmissão do documento devem ser independentes do dispositivo.

A Fig. 3.7 esquematiza o modelo de processamento de documentos, mostrando como os processos apresentados interagem uns com outros.

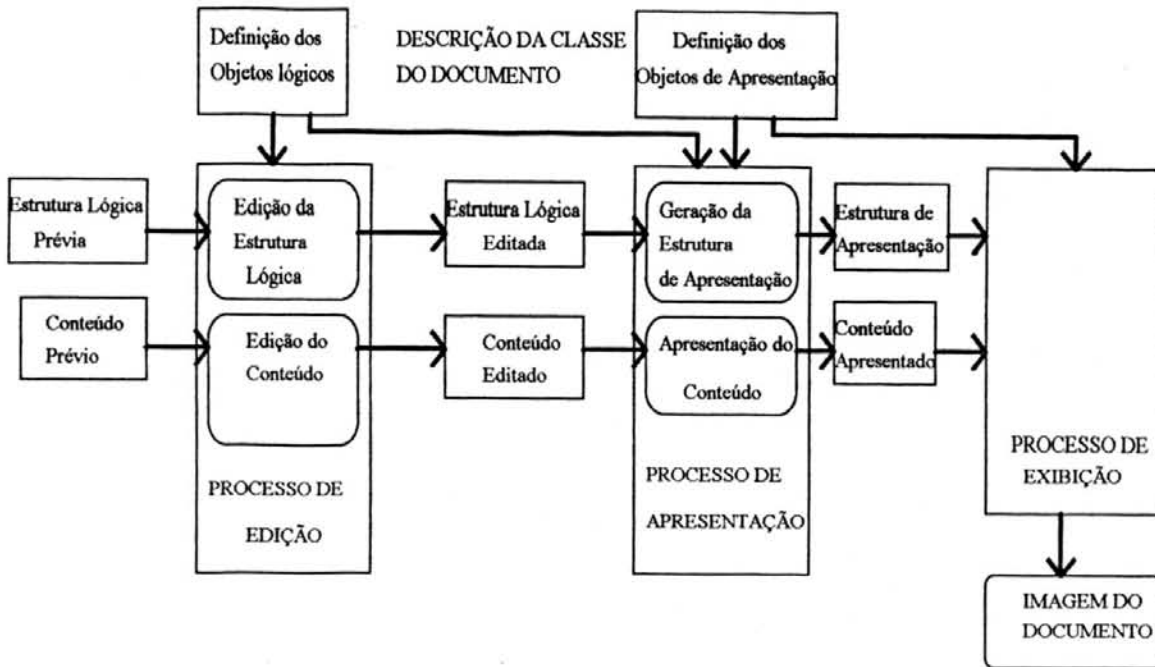


Fig. 3.7: Modelo de Processamento do Documento em ODA/ODIF.

3.4 Arquiteturas de Conteúdo dos Documentos.

O padrão ODA prevê três tipos diferentes de porções de conteúdo: caracter, gráficos *raster* e gráficos geométricos.

Cada uma delas é representada por um conjunto de atributos e um *string* de elementos gráficos com elementos de controle.

Dependendo da categoria de seus elementos gráficos, cada porção de conteúdo é estruturada e apresentada de acordo a uma certa **Arquitetura de Conteúdo**, a qual define atributos de apresentação dos objetos básicos assim como um conjunto de elementos gráficos e funções de controle com sua representação codificada.

3.4.1 Arquitetura de Conteúdo Caracter.

Está formada de caracteres gráficos e de caracteres de controle. A forma e posição dos caracteres é controlada pelos atributos e pelos caracteres de controle.

Contém facilidades para uma boa qualidade de formatação, porém, não fornece um modelo geral, tipo *PostScript*, onde o texto pode aparecer em qualquer orientação e pode ser escalado de diversas formas.

3.4.2 Arquitetura de Conteúdo Gráfico *Raster*.

Representa uma imagem gráfica de duas dimensões na forma de uma matriz retangular de elementos gráficos (*pels*).

Da mesma forma que no conteúdo caracter, mediante um conjunto de atributos controla-se a apresentação dos gráficos.

3.4.3 Arquitetura de Conteúdo Gráficos Geométricos.

Este conteúdo é, talvez, o mais simples, pois está baseado inteiramente no *Standard Computer Graphics Metafile (CGM) (ISO8632) [ISO87]*.

3.5 Formato de Intercâmbio de Documentos (ODIF).

O fluxo de dados (*data stream*) que representa o documento estruturado de acordo ao padrão ODA, consiste de uma seqüência de **descritores** e **unidades de texto**.

- **Descritores:** É uma combinação de ítems de dados consistindo de ítems de subdados e ítems de dados básicos representando os atributos do perfil do documento, uma definição de objeto ou um objeto.

- **Unidades de Texto:** Representam porções de conteúdo. É um ítem de dados que tem:

1- Ítem de subdados e ítems de dados básicos que representam os atributos da porção de conteúdo, e

2- Um conjunto de um ou mais ítems de dados básicos que representam os elementos gráficos de uma porção de conteúdo.

As unidades são distribuídas no fluxo de dados de duas formas, resultando em dois formatos possíveis do fluxo de dados.

I - Fluxo de Dados A: A ordem de transmissão é a seguinte:

- descritor do perfil do documento
- descritor das definições de apresentação
- descritor das definições lógicas
- unidades de textos para porções de conteúdo comuns
- descritores para objetos de apresentação
- descrição dos objetos lógicos
- unidades de texto para porções de conteúdo.

II - Fluxo de Dados B: Cada descritor de um objeto básico, ou de uma definição de objeto básico é seguida imediatamente pela unidade de texto de sua porção de

conteúdo. Este fluxo de dados pode ser usado quando consiste de descritores lógicos e unidades de texto ou descritores de apresentação e unidades de texto.

3.6 Conclusões.

Muitos trabalhos tem tratado o problema de estruturas para documentos [FUR82, BRO89, LIM88, LIM89, LIM89b, LIM89c, FUR89a, FUR89b, COR89, LIM90, LIM91, BOT92]. Tais estruturas tem sido desenvolvidas para sistemas interativos de preparação de documentos, cujo objetivo principal é produzir um documento impresso. Neste caso, **a estrutura do documento é projetada para fornecer ajuda inteligente na edição e atualização, ou para controlar que o documento cumpra um estilo requerido**.

Por outro lado nos sistemas de hipertextos, como será visto no capítulo seguinte, a ênfase está em fornecer um corpo estruturado de objetos, com ligações que unem objetos relacionados. Os usuários folheiam através da informação seguindo as ligações de um objeto a outro. Neste caso, **a estrutura esta projetada especificamente para auxiliar aos leitores a navegar através da informação**.

Como já foi visto, ODA fornece uma arquitetura de documento detalhada, a qual é hierárquica e orientada a objetos.

Ainda que ODA seja projetado principalmente para suportar documentos tipo impresso, ele poderia incluir um mecanismo de "ligação" que poderia ser usado para especificar ligações arbitrárias entre objetos.

Ele fornece as estruturas primárias e secundárias necessárias tanto para a edição interativa de documentos quanto para os hipertextos.

O ponto realmente importante está em estudar a possibilidade de usar ODA como um padrão tanto para documentos interativos quanto para hipertextos, propondo algumas extensões ao modelo ODA original, resultando na definição de um meta-modelo que permitirá a Gerência de Documentos Estruturados Multimídia.

A restrição atual a tudo isto é a forma limitada de especificar atributos e ligações nos objetos. Seus valores são normalmente derivados das definições genéricas dos objetos, ou herdados de atributos chamados "*Valores default*".

O caminho está em trabalhar na Estrutura Lógica Genérica, sobre a definição dos objetos, aumentando o poder de modelagem do padrão, para possibilitar a definição de métodos (não permite por enquanto) e novas e mais amplas definições de propriedades ("*valores default*", restrições, relações entre atributos, etc.).

3.6.1 Porque ODA/ODIF para a Gerência de Documentos Estruturados Multimídia (Aspectos Positivos).

Os aspectos que justificam a escolha de ODA/ODIF como ponto de partida para a definição de um meta-modelo que permita a Gerência de Documentos Estruturados Multimídia são:

- Por ser um padrão de intercâmbio de documentos, o qual oferece amplas vantagens para a transmissão de documentos entre sistemas.

- Pela divisão que faz entre Apresentação-Conteúdo-Estrutura Lógica dos documentos, que possibilita o tratamento de diferentes tipos de informação, favorecendo o aspecto multimídia dos documentos, independente da estrutura e da apresentação dos mesmos (ortogonalidade).

- A estrutura lógica possibilita, de forma eficiente, a definição de objetos compostos e de estruturas, o qual é desejável para a definição da composição dos documentos, mediante a reusabilidade das classes de objetos. ODA/ODIF define de forma eficiente a **composição dos objetos** (documentos ou partes de documentos).

- Pode incorporar facilmente novas arquiteturas de conteúdo.

- O perfil do documento e os índices podem ser usados em operações de armazenamento, navegação e recuperação.

3.6.2 Que falta em ODA/ODIF? (Aspectos Negativos).

Como foi visto na seção anterior, ODA/ODIF apresenta vários aspectos e características positivas e desejáveis que justificam a sua escolha como ponto de partida para a definição de um meta-modelo que permita a modelagem de Documentos Estruturados Multimídia.

Mas ao mesmo tempo, ele possui outros aspectos não desejáveis e carece de certas características necessárias à Gerência de Documentos Estruturados Multimídia.

Estes aspectos negativos do padrão serão resolvidos mediante a definição de extensões ao modelo original.

Alguns destes aspectos negativos são:

- Considera os documentos desde um ponto de vista **Estático**, ou seja, quando um documento é intercambiado entre dois sistemas, mas não considera o

processamento deste documento pelo sistema (armazenamento e recuperação desde um banco de documentos, navegação, etc.).

- Forma limitada de especificar atributos e ligações entre componentes da estrutura lógica.

- Não permite a definição de documentos Hiperímia (hiperdocumentos), resultado da combinação dos sistemas de hipertextos e a representação ODA/ODIF.

Para possibilitar isto, a futura extensão do padrão deverá permitir:

- O intercâmbio parcial ou total de hiperdocumentos entre sistemas, em ambientes heterogêneos.

- As características do modelo permitirão que documentos tipo ODA existentes sejam integrados num banco de hiperdocumentos, assim como permitirão diferentes visões para o mesmo conjunto de documentos.

- O suporte à criação de ligações ativas e referências cruzadas por parte de vários usuários (visões).

- O suporte para a confecção de referências externas a partir de documentos e a definição de classes de documentos já disponíveis. Também a possibilidade de definir "documentos parciais".

- A geração automática de tabelas de conteúdo e índices.

- Gerência de Anotações: Anotação é a função de anexar marcas a um documento, onde o conteúdo da marca pode consistir de textos,

desenhos, voz ou uma combinação deles. A facilidade de anotação deve satisfazer os seguintes requerimentos do usuário:

- a- Um usuário deve poder anotar todo ou parte do documento.
- b- Um usuário deve poder vincular múltiplas anotações a todo ou parte do documento.
- c- Um usuário deve poder anotar anotações (executar anotações consecutivas).
- d- Vários usuários devem poder anotar um documento ao mesmo tempo. Isto deveria incluir anotações simultâneas da mesma parte do documento (visões).
- e- Parâmetros da anotação tem que ser fornecidos, que especifiquem se a anotação é dependente ou independente de revisões.
- f- Suportar conjuntos múltiplos de anotações e permitir que elas sejam mostradas opcionalmente com o documento.

- Gerência de Versões: Oferecer mecanismos que permitam a manipulação de diferentes versões dos conteúdos dos documentos, possibilitando a navegação através do histórico do documento.

- Segurança: Versões e visões de documentos têm associado um requisito de segurança. Os usuários serão classificados de acordo com as funções de segurança disponíveis a eles. As versões e visões de documentos têm permissões associadas, as quais designarão aqueles usuários habilitados para o acesso à versão ou visão, e o tipo de acesso permitido. Também deverão ser definidos atributos de segurança no perfil do documento.

4 SISTEMAS DE HIPERTEXTOS E A GERÊNCIA DE DOCUMENTOS ESTRUTURADOS MULTIMÍDIA.

4.1 Introdução.

Atualmente o mundo está excedendo, se é que já não excedeu, o ponto de sobrecarga de informação. Isto não significa que a informação não esteja sendo usada efetivamente, mas o uso efetivo da informação é uma pequena fração do que potencialmente poderia ser, usando métodos apropriados de armazenamento e recuperação.

Uma das barreiras na exploração da informação é a falha em identificar as interconexões que permitem reconhecer ligações e similitudes entre partes de informações que estão normalmente armazenadas em lugares diferentes.

Com o objetivo de resolver este problema, várias propostas e alternativas foram analisadas e estudadas pela comunidade científica. Uma delas, a qual é de interesse deste trabalho, é o uso dos hipertextos para pesquisar e recuperar grandes volumes de informações.

Hipertexto é uma ferramenta para a construção e uso de estruturas associativas. Um documento normal é linear, e a tendência é lê-lo do começo ao fim, de forma seqüencial. Pelo contrário, a leitura mediante hipertextos é livre e pode-se pular de uma idéia a outra dentro do texto, dependendo somente do interesse do leitor.

Nas seguintes seções será dada uma definição de hipertextos, serão apresentadas as maiores diferenças com os sistemas de textos convencionais e se estudará a importância da filosofia de hipertextos na Gerência de Documentos Estruturados Multimídia.

4.2 O que é Hipertexto?

Para poder definir o que é hipertexto, necessita-se inicialmente definir o que é texto. Texto pode ser visto como "corpo de informação registrada", sendo esta idéia sinônima à de documento e, embora contenha predominantemente linguagem natural, também pode conter imagens [RAD91].

Livros, receitas, artigos, documentação de software, são exemplos de textos. Por outro lado, o prefixo hiper significa "posição superior", "além", "excesso".

Dessas definições, pode-se concluir que um hipertexto vai além do conceito de texto. Enquanto que um texto possui apenas uma dimensão (linear), um hipertexto possui mais de uma dimensão, que engloba as relações entre textos. Tais hipertextos podem ser vistos como um grafo onde cada nó contém um verbete, e os arcos representam ligações, ou referências, entre os verbetes. Quando se lê um verbete armazenado em um nó, referências explícitas para verbetes relacionados indicam onde se pode obter maiores informações sobre estes itens.

Um exemplo disto é apresentado na Fig. 4.1.

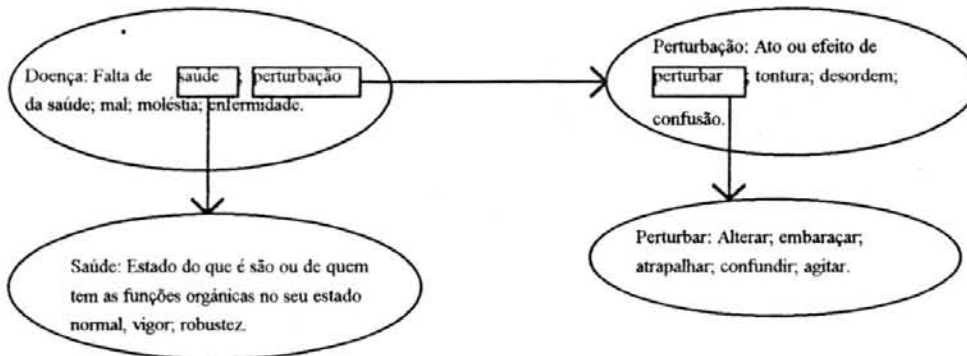


Fig. 4.1 Estrutura básica de hipertextos. Exemplo de um trecho de dicionário.

Os documentos que são criados, atualizados e manipulados com um sistema de hipertextos recebem o nome de "hiperdocumentos".

As maiores diferenças entre os hiperdocumentos e os documentos escritos convencionalmente são:

- Nos documentos convencionais as estruturas físicas e lógicas estão estritamente relacionadas. Fisicamente, o documento é uma seqüência linear de palavras, que tem sido dividida em linhas e páginas por conveniência.

Logicamente, o documento também é linear, as palavras estão combinadas para formar sentenças, sentenças formam parágrafos, parágrafos formam seções, etc.

Se o documento tem uma estrutura lógica hierárquica, esta é apresentada linearmente. Tais documentos induzem fortemente aos leitores a ler linearmente do começo ao fim, seguindo uma mesma seqüência.

- Por outro lado, os hiperdocumentos separam a estrutura lógica da física. Fisicamente, estes documentos são seqüências lineares de unidades independentes. Logicamente, eles são mais complexos. Os leitores raramente lêem tais hiperdocumentos do começo ao fim; ao contrário, eles localizam o artigo ou entrada de seu interesse (uma forma de acesso randômico) e então lêem a porção de informação seqüencialmente.

Entretanto, o leitor comumente encontrará referências cruzadas a outras porções de texto e, seguindo estes ponteiros, ele localizará o volume apropriado, encontrará a entrada apropriada e então a porção de texto relevante.

A estrutura lógica de referências dentro do próprio documento e a outros documentos é, por isto, mais complexa. O caminho lógico do leitor é uma rede através da qual pode percorrer o documento inteiro ou um conjunto deles.

Pelo expressado até agora, hipertextos são simultaneamente um método de armazenamento e recuperação de informações. Eles incorporam a noção de ligação de porções de informação, permitindo ao usuário navegar através de uma rede de partes de informações.

A informação está representada tanto por aquela que está armazenada em cada nó quanto pela forma com que os nós são ligados uns a outros. Além das ligações dentro de um documento (referências internas), ele permite criar ligações entre conjuntos de documentos (referências externas).

Quando estes sistemas de hipertextos permitem que se trabalhe com diversos tipos de meios, tais como imagens, vídeo, som, passam a denominar-se hipermídia (hipertextos+multimídia). No entanto, alguns autores usam os termos hipertextos e hipermídias como sinônimos. Esta convenção será adotada no presente trabalho.

Apesar de usar textos, que são componentes básicos dos hipertextos há muito tempo, os sistemas computadorizados recentemente passaram a ser usados com sistemas hipermídia. Isto aconteceu devido ao desenvolvimento de estações de trabalho com grande capacidade de processamento, telas gráficas de alta resolução, facilidades de comunicação através de redes e redução do custo de armazenamento de grandes volumes em sistemas "on line". Portanto, agora são considerados hipertextos apenas os sistemas onde a navegação pelo espaço hipertextual é controlada por computador.

Do ponto de vista estrutural, um hipertexto é um grafo direcionado e o sistema provê, para o usuário, mecanismos de navegação para percorrê-lo. Em um grande número de sistemas, há correspondência entre as janelas (mostradas na tela) e os nós armazenados em um banco de dados [CON87] quando se trata de informação "projetável em uma tela" (texto ou imagem). Esta correspondência é ilustrada pela Fig. 4.2, adaptada de [CON87].

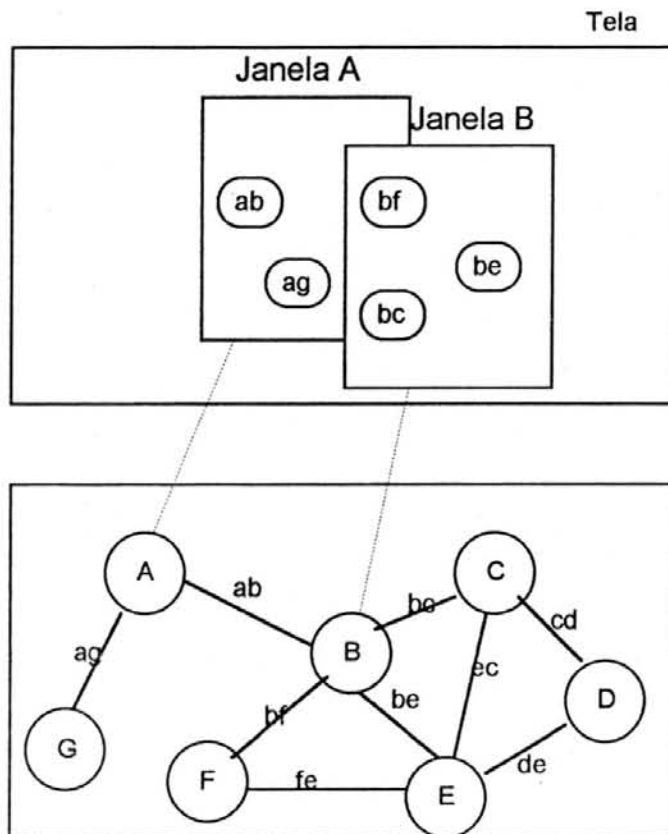


Fig. 4.2 Correspondência entre a tela e o banco de dados do hipertexto.

Alguns sistemas somente manipulam hiperdocumentos estruturados hierarquicamente ou como uma lista (ver Fig. 4.3). Outros, permitem qualquer organização dos componentes do hiperdocumento, possibilitando a criação de uma rede de informação (ver Fig. 4.4).

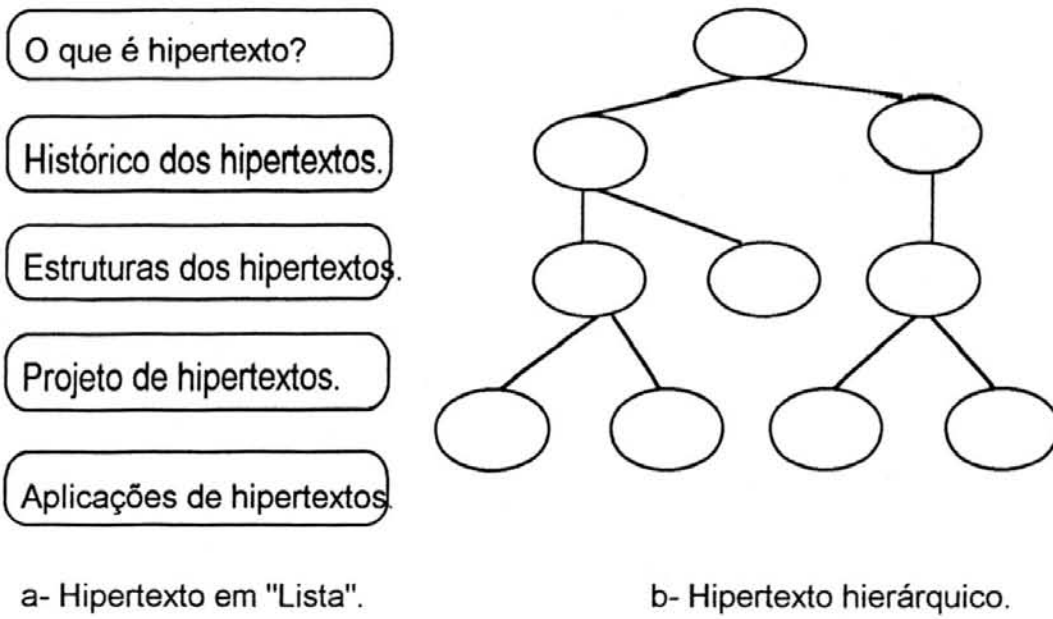


Fig. 4.3 Hipertextos com "ordem".

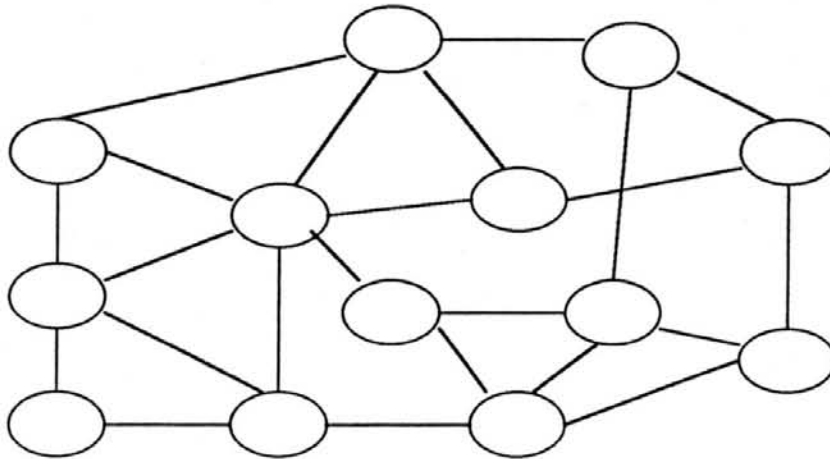


Fig. 4.4 Hipertextos em estrutura de rede.

4.2.1 Evolução dos Sistemas de Hipertextos.

Durante 1945, Vannevar Bush [BUS45, BUS67] propôs uma estruturação não linear de texto, que deveria corresponder com a natureza associativa do cérebro

humano. Em 1965, Ted Nelson batizou tal texto não linear de hipertexto e começou as tarefas de implementar hipertextos em computadores.

A idéia de textos não lineares tem sido usada por muitas pessoas quando elas tratam de colocar suas idéias em unidades de informação (nós) e fazem referência de uma unidade a outra.

Este modelo intuitivo formou a metáfora usada como base por vários sistemas e protótipos experimentais e comerciais de hipertextos. Entre os sistemas de hipertextos mais conhecidos e usados podem-se citar NLS/Augment [RHE85], XANADU [CON87], GUIDE [BRO87a, BRO87b], NoteCards [CON87, HAL88], NEPTUNE [CON87, DEL87], HyperTies [PLA91], HyperCard [NIE90] e ACQUA [MAR90, MAR91].

4.2.2 Alguns Problemas nos Sistemas de Hipertextos.

De alguma forma, a maioria dos problemas dos hipertextos são causados por sua grande flexibilidade, que é sua principal vantagem e justificção. Estes problemas podem ser agrupados em três grandes categorias: aqueles causados pela desorientação, pela falta de políticas firmes de treinamento e pela falta de um padrão.

4.2.2.1 Desorientação.

Os sistemas de hipertextos apresentam o problema de fazer sentir ao usuário "perdido" no espaço de informações, causado pela complexa estrutura associada a ele que é percorrida, mas que não é completamente visualizada. Ou seja, em qualquer momento, o usuário não sabe onde encontra-se nem como fazer para sair dessa situação.

As soluções a este problema de desorientação em hipertextos podem ser agrupadas em dois conjuntos.

No primeiro, pode-se criar marcas ou folheadores que permitem ao usuário determinar onde ele está em relação à rede total.

Na segunda opção, pode-se criar marcas, etiquetas ou "*milestones*" que representem localizações conhecidas. As ajudas de memória espacial são formas de melhorar a navegação através dos hipertextos. Estas técnicas melhoram a recuperação pela associação de itens não familiares a itens familiares.

4.2.2.2 Treinamento.

A novidade dos sistemas de hipertextos como ambientes de *software* tem como consequência problemas de treinamento, a maioria dos quais pela necessidade de adaptar os usuários a um meio não linear.

É possível que o desenvolvimento e uso dos hipertextos possam ser demorados pela falta de um adequado treinamento aos usuários. Parece contraditório falar que o treinamento deveria ser fundamental no desenvolvimento destes sistemas quando tal treinamento é uma das áreas de aplicação de hipertextos que tem maior futuro.

O sistema de ajuda em linha de HyperCard é um exemplo de como os hipertextos podem ser usados como meio de treinamento. Neste caso, o hipertexto é utilizado como uma ferramenta de treinamento do usuário para o uso de um sistema de hipertexto.

4.2.2.3 Falta de um padrão.

Os primeiros sistemas de hipertextos foram desenvolvidos por grupos de pesquisa e logo após tornaram-se produtos comerciais. Isto tem como consequência que não existe um padrão universal sobre o que é um hipertexto, podendo ficar como padrão, não o modelo mais correto, mas sim o mais popular ou conhecido.

Existe hoje em dia um entusiasmo geral sobre a idéia de substituir métodos lineares de armazenamento e recuperação de informação por métodos não lineares e associativos, mas existem muitas formas de realizar esta substituição.

Até agora, os dois conceitos que todo sistema de hipertexto inclui são os de nó e ligação, ainda que possam ser referenciados por diferentes nomes em diferentes sistemas. Por serem estes conceitos os componentes básicos de todo sistema de hipertexto, os padrões potenciais de hipertextos representarão métodos para restringir as formas nas quais eles podem ser definidos ou usados.

4.3 O Modelo de Hipertexto.

Como foi expresso na seção anterior, não existe um padrão que defina o que deve conter o modelo de um sistema de hipertexto, mas este modelo torna-se necessário para formalizar o processo de construção de aplicações de hipertextos.

Aqui surge uma pergunta importante, porque os sistemas de hipertextos precisam de um modelo de dados?

Segundo [NEM93], do ponto de vista da **arquitetura**, é importante isolar o modelo de dados como uma camada separada por diversos motivos:

-Para manter independência da implementação física; o modelo deve ser independente do sistema gerenciador de arquivos.

-Os modelos de dados fornecem um mecanismo de particionamento da informação de modo sistemático para ambientes distribuídos.

Do ponto de vista do **banco de dados**, os modelos tradicionais (hierárquico, rede, relacional) não mapeiam bem o ambiente hipermídia por pelo menos dois motivos:

- Semânticas restritivas de entidades e relacionamentos.
- Falta de suporte a abstrações estruturais.

Como os hiperdocumentos têm estruturas complexas, que tendem a diferir de um a outro, os modelos dos mesmos devem satisfazer dois critérios conflitivos:

- 1- Fornecer o maior conhecimento possível acerca da estrutura do hiperdocumento, para ajudar em sua criação, armazenamento e recuperação.
- 2- Fornecer flexibilidade, pois a estrutura do hiperdocumento é muito difícil de prever.

O modelo deveria oferecer um bom compromisso entre estes dois critérios. Como será visto a seguir, a maior parte dos modelos existentes emprega o paradigma básico de nós e elos (ligações). Serão analisados, portanto, os diferentes refinamentos feitos ao modelo básico.

Tais modelos possuem várias características comuns: definem objetos primitivos que, de algum modo, se relacionam e que possuem, cada um, um conjunto básico

de operações para exibir e editar objetos. Cada modelo de dados também define, ao menos, uma abstração de mais alto nível.

Por todo o expressado anteriormente, as propriedades dos modelos a serem apresentadas e levadas em consideração, segundo [NEM93] são:

- 1- Primitivas de representação de baixo nível.
- 2- Abstrações de alto nível para agregação e generalização.
- 3- Operações para suportar a definição e manipulação do modelo.
- 4- Relacionamentos entre o modelo de dados e a arquitetura do sistema.
- 5- Suporte de Aspectos Avançados.

4.3.1 Abstrações de Baixo Nível.

São entidades estruturais independentes de conteúdo, da aplicação e do ambiente do usuário. Como já foi expressado anteriormente, os principais componentes de um sistema de hipertexto são os nós e as ligações. A essência dos hipertextos está na existência de nós ligados. A maior parte dos modelos tratam eles como objetos de primeira classe, ou seja, são abstrações com valor semântico equivalentes aos nós.

4.3.1.1 Nós

São os átomos que compõem o hiperdocumento, representando trechos de informação. O grande problema na definição de um nó é o fato deste possuir uma fronteira bastante rígida, enquanto que em um texto comum, os parágrafos se integram com seus vizinhos. Por outro lado, os sistemas de hipertextos são mais flexíveis que os métodos tradicionais de estruturação de informação, pois eles permitem designar diferentes tipos de informação (texto, imagem, som, gráfico, animação, programas, etc.).

4.3.1.2 Botões ou Âncoras.

São regiões sensíveis ao *mouse* (ou ao dispositivo apontador que estiver sendo usado). A ativação deles pode ocasionar a execução de uma ação. Ele representa o início de um elo que, ao ser ativado, provoca o deslocamento desde o nó origem até o nó destino.

4.3.1.3 Elos ou Ligações.

A variedade de nós que podem ser definidos, como foi visto, fazem de hipertextos uma ferramenta extremamente flexível e poderosa para a representação de informação. Neste contexto, as ligações definem a estrutura do hipertexto e fornecem capacidades de folhear e pesquisar os nós da rede.

Os elos estabelecem conexões entre os nós do hipertexto, possibilitando a navegação do usuário pelo espaço de informações. Os elos tem início e fim, sendo assim, o nó que contém o início de um elo chama-se nó origem, e aquele que se encontra no fim do elo é chamado de nó destino. A fig. 4.5 mostra os componentes básicos e como se relacionam.

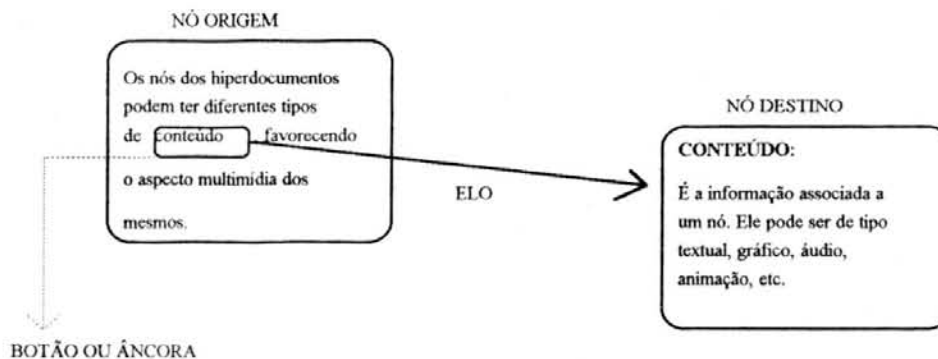


Fig. 4.5 Abstrações de baixo nível (nó, elo, botão) e seu relacionamento.

Segundo [PAR89], podem ser definidos diferentes tipos de elos ou ligações, dependendo da ação que executam e das informações que ligam.

-Elos "mover a" (*move-to*): Estas ligações simplesmente deslocam para um nó relacionado. Exemplos deste elo são as referências internas realizadas num hiperdocumento (referências a componentes lógicos da mesma estrutura), e as referências externas (ligam componentes lógicos de diferentes hiperdocumentos). A figura 4.6 exemplifica estes tipos de ligações.

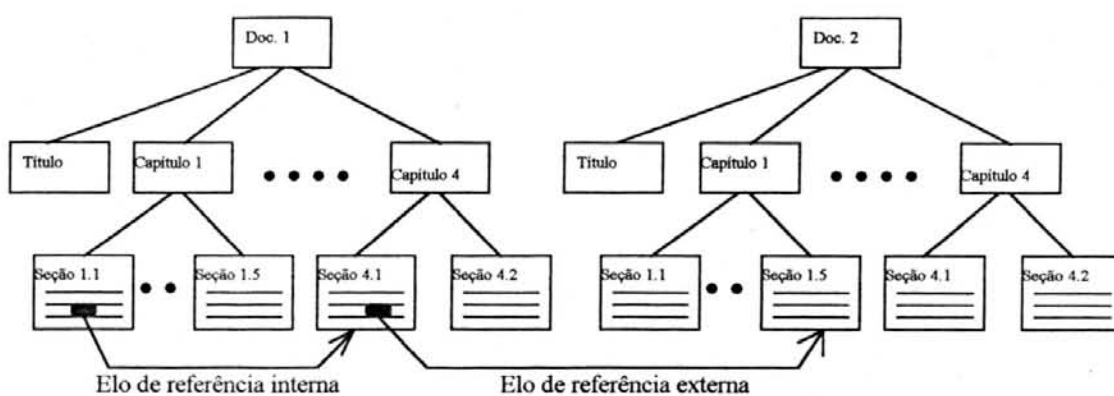


Fig. 4.6 Elos "mover a".

- Elos de expansão (*Zoom*): Expandem o nó atual a uma forma mais detalhada da informação contida nele. Este efeito é similar a mover-se de uma forma abstrata a uma forma mais completa do documento. Exemplos deste tipo de ligação são as anotações e comentários ou as notas de pé de página (ver figura 4.7).

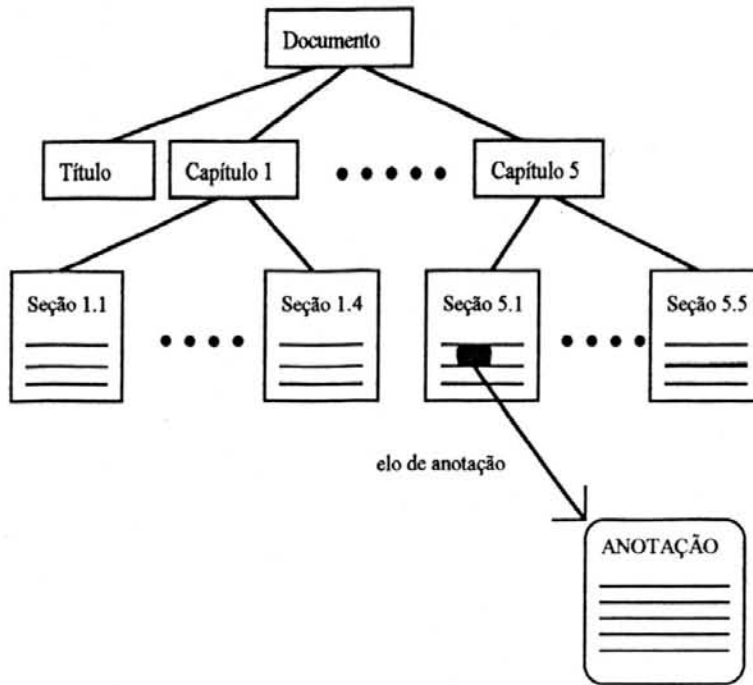


Fig. 4.7 Elo de expansão (*Zoom*). Exemplo de uma anotação.

- **Elos de visões:** A disponibilidade ou ativação destas ligações está condicionada pelo interesse ou propósito do usuário. Estas ligações estão ocultas, a não ser que elas sejam de interesse do usuário e ele tenha acesso ou tenha sido justificado para seu uso.

- **Elos de índices:** Levam ao leitor desde um nó indexado até a entrada indexada correspondente para aquele nó. O índice pode ser usado para busca de documentos que compartilham um termo indexado em particular. A figura 4.8 ilustra este tipo de ligação.

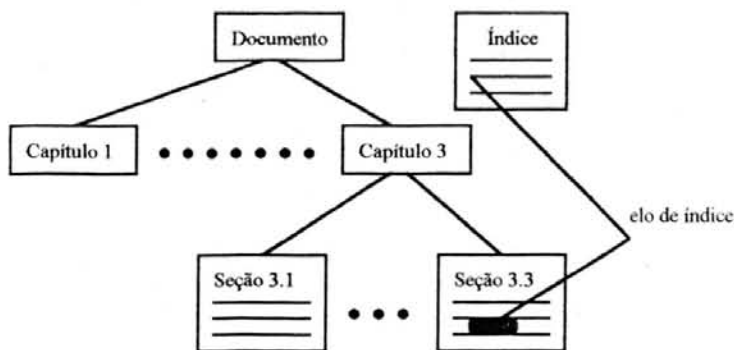


Fig. 4.8 Elos de índices.

-**Elos executáveis:** Permitem que o hipertexto seja considerado uma interface de programação de alto nível. Os botões ligados a eles causam ações ou a execução de algum tipo de código associado.

- **Elo estrutural:** Ligam as partes lógicas do hipertexto formando a estrutura lógica do mesmo. Permitem percorrer a estrutura, simulando a leitura seqüencial dos documentos impressos. A figura 4.9 apresenta um exemplo deste tipo de ligação.

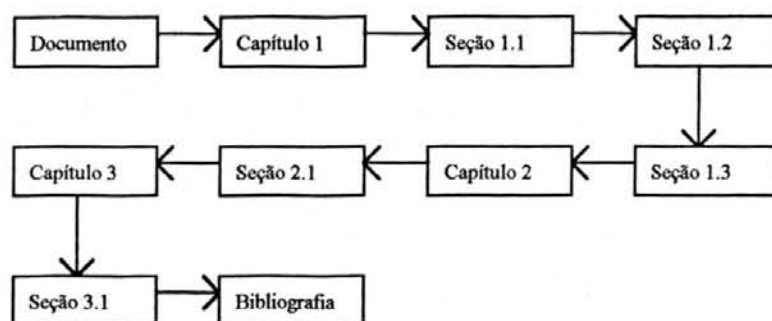


Fig. 4.9 Elos estruturais.

4.3.2. Abstrações de Alto Nível.

Usualmente são abstrações do tipo **agregação** ou **generalização**. Existem diferentes definições, variando de sistema para sistema, podendo-se identificar como mais importantes as seguintes:

No modelo HDM [GAR91, SCH93] definem-se as abstrações **entidade**, **perspectiva** e **esquema**, as quais serão tratadas na seção 4.3.6.1.

Por outro lado, no sistema NEPTUNE [CAM88] definem-se as abstrações **Contexto** e **Grafo**, consideradas na seção 4.3.6.3.

Também o modelo de LANGE [LAN90] define abstrações de alto nível, as quais são chamadas **Redes e Estruturas** (ver seção 4.3.6.4).

4.3.3 Operações.

Um modelo de hipertexto, além de definir os componentes estruturais do mesmo (abstrações de alto e baixo nível), também tem que especificar as operações e/ou métodos que tais componentes suportarão.

Tais operações incluem especificações para funções genéricas de visualização e edição dos componentes do modelo.

Estas operações podem ser divididas segundo sua função, sendo alguns exemplos os seguintes:

-Gerais da estrutura: CriaComponente, ApagaComponente, ModificaComponente, ValorAtributo, etc.

-Sobre Nós: VisualizaNó, DaValorAtributo, DaLigações, AutorNó, VersãoNó, etc.

-Sobre os elos: AdicionaElo, TemÂncora, TemDestino, VisualizaElo, AdicionaDestino, AdicionaÂncora, etc.

4.3.4 Relação com a Arquitetura do Sistema.

A arquitetura genérica que caracteriza a maior parte dos sistemas de hipertextos consiste de três camadas:

- Interface com o usuário final (Apresentação).
- Modelo do hipertexto (nós e elos).
- Interior dos componentes (armazenamento físico).

Esta abordagem é especialmente importante, pois permite utilizar a mesma estrutura para uma variedade de tipos de meios. Da mesma forma, os mesmos tipos de meios podem ser mapeados para múltiplas estruturas para suportar diferentes exibições aos usuários. Os diferentes sistemas utilizam vários métodos para manter a independência do modelo de dados com a interface do usuário e com o armazenamento físico dos componentes.

A figura 4.10 apresenta a arquitetura do modelo de referência DEXTER [HAL90], a qual reflete o discutido nesta seção.

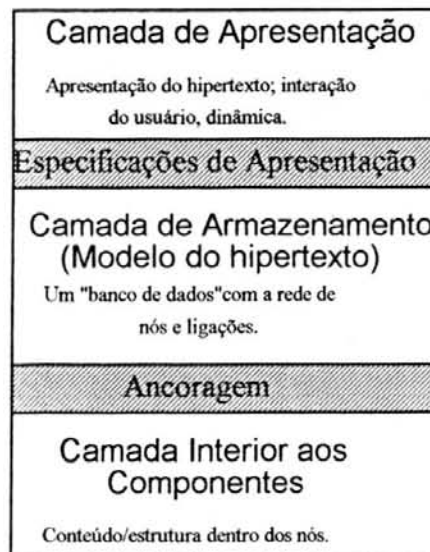


Fig. 4.10 Arquitetura dos sistemas de hipertextos (segundo DEXTER).

4.3.5 Suporte de Aspectos Avançados.

Estes aspectos não são específicos de sistemas de hipermídia, uma vez que se aplicam a qualquer ambiente de banco de dados, entretanto, são particularmente significativos para o desenvolvimento de sistemas de hipermídia distribuídos e colaborativos.

I- Versionamento: O suporte de versionamento fornece a capacidade de manipular mudanças da informação no tempo. Uma extensão é o conceito de configuração, onde uma coleção de versões de informações relacionadas é mantida.

O que deve ser discutido para isto é:

- 1- o tratamento deve fazer parte do modelo ou da aplicação?
- 2- como deve ser feita a propagação das versões?
- 3- quando um componente de um objeto composto é revisado, as informações de versões são propagadas para os componentes relacionados?
- 4- as novas versões devem ser representadas como cópias ou como operações que deram origem à mesma?
- 5- como verificar a consistência de uma configuração?

II- Colaboração: O suporte para ambientes de colaboração deve prever múltiplos usuários, controle de concorrência e funções distribuídas. As implicações no modelo de dados destes ambientes não foram ainda investigadas extensivamente.

III- Segurança: Restringir o acesso à informação é uma preocupação de qualquer ambiente multiusuário. Os tipos de restrições e a granularidade dos dados aos quais as restrições se aplicam podem variar muito.

4.3.6 Alguns Modelos Existentes.

Não é objetivo deste trabalho descrever em detalhe um grande número de modelos de hipertextos nem fornecer uma classificação dos mesmos, no entanto, os mesmos foram objeto de profundo estudo e análise por parte do autor para a realização da presente dissertação. A seguir será dada uma breve menção aos mais importantes. Maiores detalhes podem ser encontrados nas respectivas referências bibliográfica.

4.3.6.1 O modelo HDM.

O *Hypermedia Design Model* (HDM) [GAR91, SCH93] é um modelo para a especificação da hiperbase de uma aplicação hipermídia. O seu objetivo é permitir um processo de autoria "estruturado", separando as questões de autoria e permitindo o seu tratamento individualizado. Ele é um mecanismo de modelagem, que permite a descrição de maneira simples e concisa, e independente de sistemas, de hipertextos a serem projetados.

No modelo HDM definem-se dois níveis de autoria:

- "*Autoria in the large*": a qual define a topologia do hipertexto.

- "*Autoria in the small*": a qual trabalha no conteúdo dos nós, determinando sua aparência ao leitor.

HDM também define as abstrações:

-**Entidade**: Hierarquia de componentes que representam um objeto concreto ou conceitual da aplicação

-Perspectiva: Aparência de uma peça de informação. A descrição de um componente de acordo a uma perspectiva é chamada de **unidade**.

-Esquema: Conjunto de definições de tipos de entidades e tipos de ligações.

Também define diferentes tipos de ligações, sendo elas **estruturais, de aplicação e de perspectiva**. A figura 4.11 apresenta um exemplo que esclarece as abstrações definidas no modelo HDM.

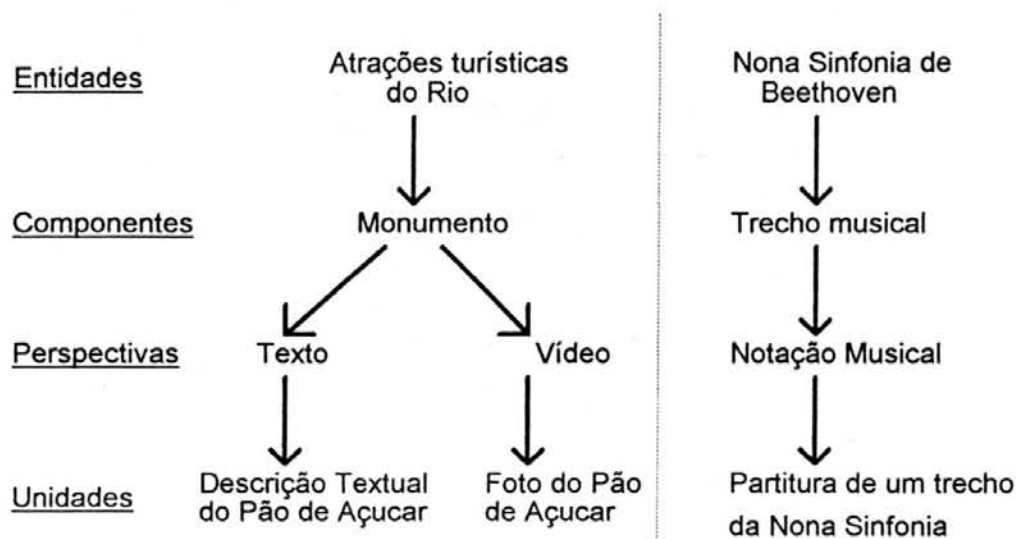


Figura 4.11: Abstrações definidas em HDM.

4.3.6.2 HAM.

HAM (Hypertext Abstract Machine) [CAM88], é um servidor baseado em transações, de propósito geral, para o armazenamento de hipertextos, que está baseado no sistema de armazenamento do Neptune [CON87],[DEL87].

Seu modelo de armazenamento é baseado em grafos e é orientado a objetos. Trabalha com cinco tipos de objetos: grafos, contextos, nós, elos e atributos.

-Grafo: É o objeto de mais alto nível dentro do sistema. Contém toda a informação referente à aplicação. Ele pode conter um ou mais contextos.

-Contextos: São partições de nós e elos de um grafo. Cada um tem um contexto pai, e zero ou mais contextos filhos.

-Nó: Contém dados arbitrários.

-Elo: Define uma relação entre dois nós do mesmo ou de diferentes contextos.

-Atributos: Podem estar associados aos nós, contextos ou elos. Ditos atributos e seus valores fornecem a semântica dos objetos.

Para exemplificar, suponha-se o seguinte caso: numa aplicação de autoria de livros, um grafo pode representar o conjunto de todos os livros escritos por um autor em particular e um contexto pode representar um livro em particular (um conjunto de nós, neste caso capítulos, e um conjunto de elos, neste caso as relações entre os capítulos).

4.3.6.3 O Modelo de Lange.

O modelo de Lange se concentra principalmente no modelo de dados; não se trata de uma proposta de arquitetura para sistemas de Hipertextos [LAN90]. Nós e elos são as unidades básicas do modelo que define duas abstrações compostas: **redes e estruturas**.

-Redes: Abstração que representa um conjunto de elos. Simboliza os elos referenciais que unem partes do hipertexto.

-Estruturas: Definem um tipo de organização para os nós e as redes, sendo possível reutilizar nós e redes em várias estruturas. Representam uma classe de elos organizacionais entre nós do hipertexto. Elas podem ser do tipo conjunto, seqüência, mapas e também definições recursivas, como por exemplo estrutura de tipo árvore.

Desta forma, as estruturas podem ser consideradas como um **filtro**, que possibilita representações lineares dos hipertextos. Por exemplo, a estrutura árvore de um livro deveria ser interpretada por um filtro da forma *top-down left-to-right*, de forma que o capítulo 1 e suas seções sejam apresentados antes que o capítulo 2, e assim por diante.

Não existe, ao contrário de outros modelos, uma clara separação entre conteúdo e estrutura. Em resumo, o modelo de Lange define hipertexto como uma coleção de nós e uma ou mais redes conectando os nós, e uma estrutura descrevendo a organização das partes que formam o hipertexto.

Uma das maiores decisões no desenvolvimento deste trabalho tem sido a separação dos aspectos de apresentação e da semântica de *browsing* do modelo, e seu deslocamento ao projeto da aplicação.

4.3.6.4 O Modelo de Referência DEXTER.

Como foi expressado nas seções anteriores, a falta de um modelo padrão levou à necessidade de tentar capturar as abstrações encontradas nos diversos sistemas de hipertextos.

O modelo DEXTER [HAL90] é o resultado de dois *workshops* sobre hipertextos. O primeiro ocorreu no Dexter Inc., em New Hampshire (EUA), e daí seu nome. Em tal modelo, um sistema de hipertexto é dividido em três camadas. (ver Figura 4.10)

- 1 - Camada Interna aos Componentes.
- 2 - Camada de Armazenamento.
- 3 - Camada de Apresentação.

Este modelo também define duas interfaces, a primeira entre a camada interna aos componentes e a camada de armazenamento é denominada ancoragem, e é quem proporciona o mecanismo para o endereçamento de itens dentro de um componente específico.

A interface entre a camada de execução e a camada de armazenamento é denominada especificação de apresentação. Esta é um mecanismo que permite o armazenamento de informação sobre como um componente deve ser apresentado ao usuário. Um exemplo do uso de Especificações de Apresentações pode ser visto na Figura 4.12.

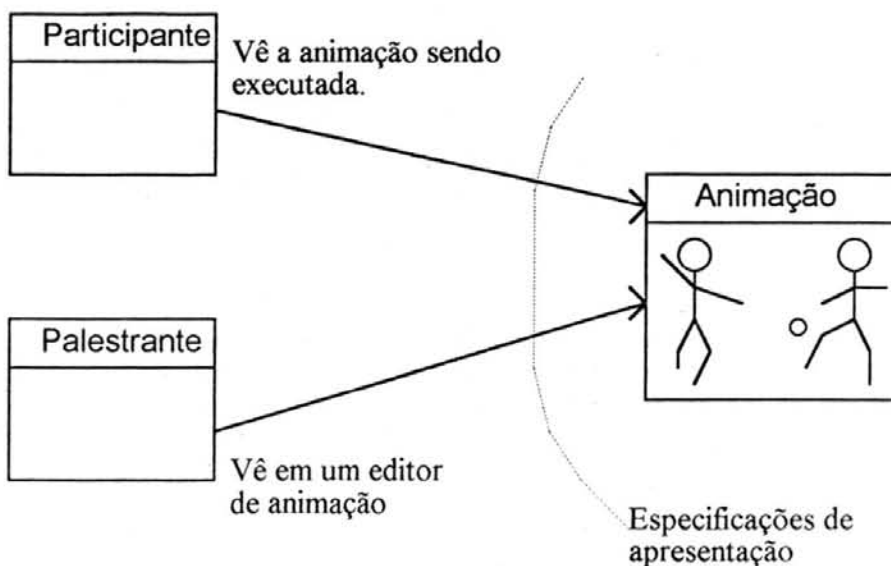


Fig. 4.12 Exemplo de uso de Especificações de Apresentação.

DEXTER ainda está em fase inicial de desenvolvimento, não tendo sido ainda utilizado na modelagem de um sistema completo de hipertextos. Nenhum sistema existente hoje em dia implementa todos os mecanismos previstos pelo modelo.

4.4 Conclusões.

Como já foi expressado em capítulos anteriores, o sistema de Gerência de Documentos Estruturados Multimídia deveria oferecer tanto as capacidades dos formataadores de documentos quanto as capacidades dos sistemas de hipertextos.

Como um primeiro passo para atingir o expressado anteriormente, foram identificadas as características positivas (desejáveis) do padrão de documentos ODA/ODIF que justificam a escolha de tal padrão como ponto de partida para a definição de um meta-modelo que permita tal gerência.

Mas também foram identificadas lacunas em tal padrão, que não satisfazem plenamente os requerimentos do problema a modelar, e que foram consideradas na seção 3.6.2.

Por tudo isso, o presente trabalho propõe uma integração destas duas filosofias (formatadores de documentos e sistemas de hipertextos), e para poder levar a cabo isto, foi necessário apresentar brevemente neste capítulo os sistemas de hipertextos, para depois, identificar as características desejáveis dos mesmos, que serão integradas no futuro meta-modelo. Na próximas subseções serão identificadas as características positivas e negativas dos sistemas de hipertextos em relação à Gerência de Documentos Estruturados Multimídia.

4.4.1 Modelos de Hipertextos para a modelagem de Documentos Estruturados Multimídia (Aspectos Positivos).

-A principal, sem dúvida, é fornecer um corpo estruturado de objetos. Neste caso, a estrutura está projetada especificamente para auxiliar aos leitores a navegar através do hiperespaço de informação.

-Tal estrutura possibilita definir, de forma eficiente, relacionamentos lógicos entre seus componentes.

- Alguns sistemas oferecem mecanismos para o tratamento de aspectos avançados (versões, visões, segurança), e seria interessante possuí-los no sistema de Gerência de Documentos Estruturados Multimídia.

4.4.2 Porque não só modelos de hipertextos? (Aspectos Negativos).

Assim como foram identificadas características positivas nos sistemas de hipertextos que auxiliariam na modelagem de documentos estruturados multimídia, eles também possuem outras que não são desejáveis ou carecem de certos aspectos essenciais para tal modelagem.

Estas características negativas e/ou falta de certos aspectos importantes são motivo pelo qual não é escolhida diretamente a filosofia de hipertextos como ferramenta única e total de modelagem. Mas o conhecimento e identificação de tais aspectos negativos permitirá uma melhor definição do meta-modelo a integrar. Os aspectos negativos identificados são:

I-O modelo de nós-ligações é uma representação de baixo nível, sendo orientada para a implementação e não para o projeto.

II-A ênfase dada aos sistemas de hipertextos em as aplicações correntes tem sido de ler ou navegar documentos já existentes. Pouco se considera o problema de usar um sistema de hipertextos para criar cooperativamente documentos.

III- Definição inadequada ou pouco clara da divisão entre estrutura-conteúdo-apresentação.

IV- Suporte inadequado às estruturas compartilhadas entre usuários, o que poderia ser resolvido pela definição de visões.

V- Falta de um padrão, o que complica a comunicação entre sistemas heterogêneos.

VI- Não existe uma clara definição dos objetos compostos, ou estruturas que possam ser manipuladas claramente. Depende muito da aplicação ou dos diferentes criadores.

VII- O já citado problema da desorientação do usuário na estrutura modelada.

5 O PARADIGMA DE ORIENTAÇÃO A OBJETOS COMO FERRAMENTA DE MODELAGEM.

5.1 Introdução.

Identificados os aspectos e características desejáveis do padrão de documentos ODA/ODIF (capítulo 3) e dos sistemas de hipertextos (capítulo 4) a ser integrados no sistema de Gerência de Documentos Estruturados Multimídia, agora torna-se necessário identificar uma ferramenta ou técnica que permita tal integração, de forma natural, e que possibilite a modelagem apropriada de tal sistema.

A multimídia, como foi visto, é a tentativa de se aumentar a capacidade de representação dos objetos do mundo real, que podem ser melhor expressos e definidos por intermédio da combinação de vários tipos de dados: som, texto, imagem (estática e animada), etc. Estes objetos requerem, para a sua manipulação, a ampliação do processamento tradicional de dados, de forma a suportar novos tipos. Não sendo a multimídia uma tecnologia independente, ela necessita o apoio de tecnologias de origens diversas que lhe dêem suporte, como por exemplo a orientação a objetos.

A orientação a objetos visa reunir o aspecto estrutural e o aspecto comportamental na definição do objeto, garantindo a representação das entidades do mundo real através de abstrações mais próximas da realidade. A versatilidade do modelo de dados orientado a objetos faz com que o mesmo se constitua num pré-requisito fundamental para o desenvolvimento de aplicações multimídia. Os Sistemas Gerenciadores de Banco de Dados Orientados a Objetos (SGBDOO), por serem instrumentos que combinam os recursos disponíveis nos SGBD's tradicionais com o paradigma de orientação a objetos num mesmo ambiente, despontam como a ferramenta ideal para o desenvolvimento de tais aplicações multimídia.

Além disto, a área de orientação a objetos está em franco desenvolvimento, com inúmeras propostas de metodologias e notações, sendo por isto de grande interesse para a criação de ambientes experimentais.

5.2 Modelos de Dados Orientados a Objetos.

A mutabilidade dos tipos de dados manipulados pelas novas aplicações e a necessidade de se representar a realidade de forma mais natural e fiel, foram fatores básicos para o surgimento de diferentes modelos de dados. Um grande desafio para estes modelos é obter estruturas contendo a semântica dos fenômenos representados. Atualmente existe uma grande tendência para se tratar os aspectos estáticos (estruturais) e dinâmicos (comportamentais) de forma integrada, onde a abordagem dinâmica aparece como o prolongamento da estrutural.

O enfoque integrado tem contribuído bastante para o avanço do processo de modelagem de dados, tendo como resultado recente os modelos orientados a objetos, que são mais completos e abrangentes do que as proposições de integração até então existentes.

Os aspectos estáticos exprimem as propriedades estruturais invariantes, definidas através do conjunto de associações entre os seus componentes; tais propriedades estão ligadas à noção de estado de um sistema. Os quatro conceitos básicos associados aos aspectos estáticos são os de objeto, relacionamentos, restrições de integridade estática e tempo. Todos os modelos de dados representam os aspectos estáticos.

Estes modelos têm um papel importante no futuro dos SGBD [DAT89] porque, em primeiro lugar, são mais adequados para o tratamento de objetos complexos (textos, gráficos, imagens, etc.) e dinâmicos (programas, simulações), e em segundo por

possuir maior naturalidade conceptual e por estarem em consonância com fortes tendências em linguagens de programação e Engenharia de *Software*.

Inicialmente pode-se dizer que orientação a objetos corresponde à organização de sistemas como uma coleção de objetos que integram estruturas de dados e comportamento. Além desta noção básica, a abordagem inclui um certo número de conceitos, princípios e mecanismos que a diferencia das demais.

Os modelos de dados orientados a objetos são atualmente o centro de várias pesquisas de uma grande parte da comunidade científica nas áreas de banco de dados e Engenharia de *Software*.

Apesar disso, a terminologia comumente usada nos SGBDOO não está ainda definida precisamente. Da mesma forma não há um consenso sobre quais devem ser as características mínimas requeridas para que um SGBD possa ser considerado orientado a objetos, e mais ainda, não existe um modelo de dados comum às diferentes visões. Outra deficiência é a inexistência de um modelo de dados orientado a objetos coerente, com uma base matemática sólida que facilite a definição de consultas e assegure sua consistência, como o cálculo ou a álgebra relacional nos modelos de dados relacionais. Também existe uma forte atividade experimental com protótipos construídos por indústrias e universidades, Galileo [ORS85], ORION [BAN87], IRIS [FIS87]; e os produtos já lançados no mercado, GEMSTONE [MAI84], O2 [VEL88] e VBASE [ONT88].

A ausência de um modelo de dados comum e uma teoria matemática de suporte implica no risco de que um destes protótipos surja e se converta no modelo padrão, não porque seja o melhor, mas o mais difundido.

Preocupados com este problema, dois grupos distintos de cientistas elaboraram respectivamente um manifesto [ATK89] e um contra-manifesto [LIN90], onde expressaram quais as características mínimas requeridas para que um SGBD possa ser considerado orientado a objetos.

A existência destes dois grupos é uma constatação de que não existe um consenso do que deveria ser um SGBDOO. No manifesto de [ATK89] existe uma clara defesa da criação de uma nova geração de SGBD, resultado da adição de capacidades de SGBD a linguagens de programação orientados a objetos. No contra-manifesto [LIN90], propõe-se uma extensão do modelo relacional tradicional.

5.3 Características.

5.3.1 Objetos Complexos.

São aqueles formados por construtores (conjunto, listas, tuplas, *arrays*) aplicados a objetos simples (inteiros, booleanos, *strings*). Segundo [ATK89] o conjunto mínimo de construtores requeridos para um SGBDOO é formado por: conjunto, lista e tupla.

5.3.2 Identidade de Objetos.

Num modelo com identidade de objetos, estes têm existência independente de seus valores correntes e dos endereços de armazenamento físico. A identidade de objeto persiste no tempo, independentemente das mudanças de estado.

5.3.3 Encapsulamento.

A ideia de encapsulamento possibilita a distinção entre a especificação e a implementação das operações de um objeto, além de fornecer a modularidade que permite uma melhor estruturação das aplicações ditas complexas.

5.3.4 Classificação (Classes).

A noção de classe significa um conjunto de objetos que possuem o mesmo tipo (atributos, relacionamentos, operações). A noção de classe é associada ao tempo de execução, podendo ser vista como uma representação por extensão, enquanto que o tipo é uma representação intencional. Cada classe tem um tipo associado, o qual especifica a estrutura e o comportamento de seus objetos.

As classes têm dois aspectos, sendo uma fábrica ou um depósito de objetos. A fábrica de objetos é usada para criar novos objetos do tipo associado à classe. O depósito da classe mostra que associada à classe está a sua extensão, ou seja, o conjunto de objetos que são suas instâncias.

5.3.5 Herança.

Herança é um mecanismo que permite ao usuário definir tipos de forma incremental, por refinamento de outros já existentes, permitindo composição de tipos em que as propriedades de um tipo são reutilizadas na definição de um novo tipo.

Na herança simples um certo tipo pode ter apenas um supertipo, da mesma forma uma subclasse herda de uma única classe.

5.4 Porque utilizá-lo na modelagem de Documentos Estruturados Multimídia?

As características que justificam a escolha do paradigma de orientação a objetos como ferramenta de modelagem, possibilitando a integração dos sistemas de hipertextos e do padrão ODA/ODIF (sistemas formatadores de documentos), visando a modelagem de Documentos Estruturados Multimídia, são apresentadas a seguir:

-Mais natural para o tratamento de objetos complexos multimídia (aspecto fundamental na aplicação tratada).

-As características de reusabilidade e modularidade são desejáveis para a definição, por exemplo, de diferentes partes e/ou tipos de documentos.

-Os conceitos de classe e herança são desejáveis pois permitem a definição de tipos de forma incremental, mediante o refinamento de outros já existentes, permitindo também a composição de tipos.

-Abstrações do tipo **generalização, associação e agregação** são adequadas para a modelagem das características dos Sistemas de Gerência de Documentos Estruturados Multimídia expressadas anteriormente.

-Será facilitada a integração do meta-modelo resultante a um SGBDOO, pois as características de volume de informação e complexidade dos documentos sugerem a utilização de um banco de dados.

6 OHypA, O META-MODELO DE DOCUMENTOS ESTRUTURADOS MULTIMÍDIA PROPOSTO.

6.1 Introdução.

Como já foi expressado, no âmbito do projeto DO²CEM entende-se como Gerência de Documentos Estruturados Multimídia a criação, armazenamento, recuperação e transmissão de documentos contidos em um banco de documentos.

Este sistema deverá oferecer as capacidades tanto dos sistemas de hipertextos quanto dos formatadores de documentos (compatíveis com o padrão ODA/ODIF), as quais foram também identificadas nos capítulos anteriores.

O primeiro passo para tratar tal sistema é considerar e definir um meta-modelo, o qual será consequência da integração dos conceitos dos sistemas de hipertextos e do padrão ODA/ODIF, utilizando como ferramenta de integração o paradigma de orientação a objetos.

Tal integração permitirá, de uma maneira natural, a Gerência de Documentos Estruturados Multimídia, possibilitando o estudo e delineamento de possíveis soluções para problemas comuns resultantes da integração das referidas áreas.

Como resultado da integração, a seguir será apresentado o meta-modelo OHypA (*Office HyperDocument Architecture*), o qual pode ser considerado tanto como uma extensão ao padrão ODA/ODIF quanto uma proposta de modelo de hiperdocumentos. Tal meta-modelo terá uma aplicação real e prática pela combinação da tecnologia de hipertextos e a representação de documentos segundo ODA/ODIF.

Para uma melhor organização do presente trabalho, a apresentação do meta-modelo OHypA será dividida em duas partes, sendo a primeira tratada no presente capítulo. Nela, serão considerados aspectos básicos do meta-modelo, que são:

- Relação com a arquitetura do sistema.
- Estrutura lógica básica do meta-modelo.
- A modelagem de Referências Internas (Relações Intra-Documentos).
- A modelagem de Referências Externas (Relações Inter-Documentos).
- A modelagem de Anotações e/ou Comentários.

Por sua vez, a segunda parte será tratada no próximo capítulo, e nela serão considerados aspectos avançados do meta-modelo, que são:

- Visões dos Usuários (Autoria/Segurança).
- Tratamento de versões dos documentos.
- Abstrações de Baixo Nível.
- Abstrações de Alto Nível.
- Tipos de ligações.
- Objetos do meta-modelo.
- Operações/Métodos que são suportados pelo meta-modelo.

6.2 OHypA e sua relação com a arquitetura do sistema de Gerência de Documentos Estruturados Multimídia.

O presente trabalho define um meta-modelo que auxilia na organização das porções de informação de um hiperdocumento multimídia. É possível que o projeto de um

hiperdocumento específico corresponda à organização estabelecida no meta-modelo, mas também permite-se que novas relações sejam acrescentadas em tal projeto.

Ou seja, o meta-modelo não pretende definir regras para a estruturação dos hiperdocumentos, mas fornece um esquema para que o criador (autor) faça isso. Nele, não são definidas classes de objetos lógicos particulares (capítulos, seções, figuras, etc.) nem estruturas de hiperdocumentos particulares (livros, artigos, roteiros turísticos, etc.). Em vez disso, o meta-modelo fornece um esquema, dentro do qual se pode criar objetos, classes de objetos, estruturas e relações entre eles, e uma vez criados, poder gerenciar e transmitir os mesmos de uma aplicação originadora para uma aplicação receptora.

Como um primeiro passo, será considerada e definida a relação entre o meta-modelo e a arquitetura do sistema de Gerência de Documentos Estruturados Multimídia.

Em capítulos anteriores, considerou-se que a maioria dos sistemas de computação, incluídos os sistemas de Gerência de Documentos Estruturados Multimídia, apresentam uma arquitetura dividida em três níveis: o **nível de apresentação** (*interface* com o usuário), o **nível da estrutura** (ou modelo do hiperdocumento, propriamente dito) e o **nível interior aos componentes** (ou armazenamento dos conteúdos). Tal arquitetura é mostrada na figura 6.1.

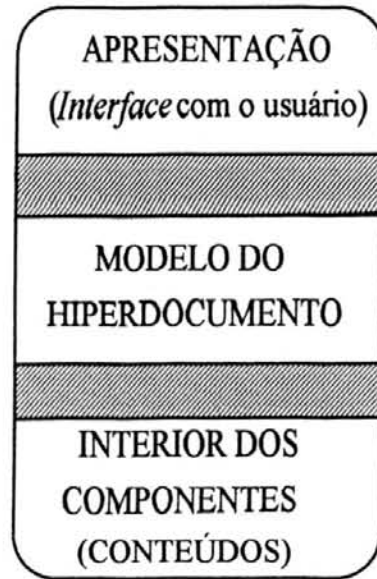


Fig. 6.1: Arquitetura do sistema de Gerência de Documentos Estruturados Multimídia.

O objetivo é trabalhar no nível do modelo do hiperdocumento, independente de questões de apresentação (como é mostrado o hiperdocumento aos usuários), pois como OHypA é resultado de uma extensão ao padrão ODA/ODIF, ele manterá a divisão e relações existentes entre as estruturas lógicas e de apresentação propostas no modelo original.

Também o meta-modelo independe das estruturas internas (conteúdos) de seus componentes, devido à divisão estrutura-conteúdo que faz o padrão ODA/ODIF, a qual será mantida no meta-modelo OHypA.

Finalmente o meta-modelo tem que ser independente da semântica de navegação (forma ou alternativas de percorrer a informação contida no documento), a qual será considerada no projeto das aplicações, mas ele deve oferecer as bases estruturais necessárias que possibilitem a definição e execução de tais semânticas de navegação.

Considerando o citado anteriormente, o nível do modelo de hiperdocumento, agora chamado meta-modelo OHypA, é subdividido em três níveis, como mostra a figura 6.2.

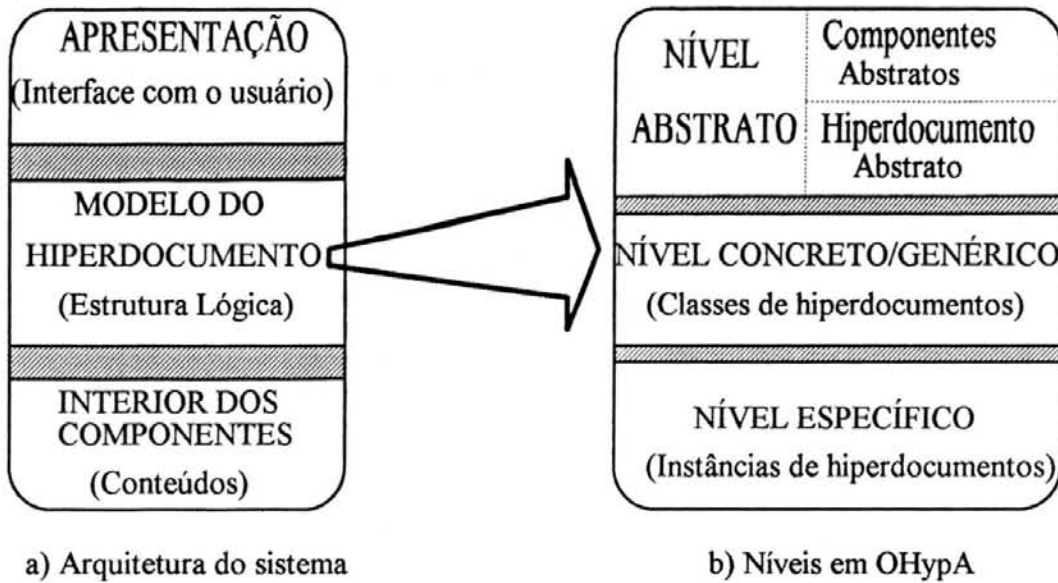


Figura 6.2: Níveis no meta-modelo OHypA.

-Nível Abstrato: Este nível está subdividido em: **Nível de Componentes Abstratos**, no qual são definidos as representações abstratas das peças (componentes) do hiperdocumento, e **Nível de Hiperdocumento Abstrato**, o qual define uma representação abstrata do hiperdocumento, ou seja, como são associados (associações válidas) os componentes abstratos.

Dentro do nível dos componentes abstratos, as representações são independentes umas de outras, e serão relacionadas no nível de hiperdocumento abstrato.

-Nível Concreto/Genérico: Neste nível definem-se classes de objetos e tipos de estruturas particulares a certos hiperdocumentos. Tais classes e estruturas são subclasses

das definidas no nível abstrato. Este nível corresponde às estruturas genéricas definidas no padrão ODA/ODIF.

-Nível Específico: Corresponde às instâncias das classes de hiperdocumentos definidos no nível anterior, ou seja, são os hiperdocumentos propriamente ditos, com suas relações às porções de conteúdo.

Estes níveis também podem ser exemplificados de uma outra maneira, que é apresentada na figura 6.3.

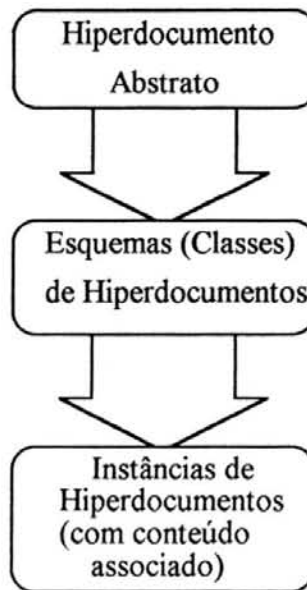


Figura 6.3: Níveis do meta-modelo OHypA.

No que segue do presente capítulo e no seguinte, conceitos serão gradativamente definidos, de maneira a ir formando o meta-modelo OHypA, resultando finalmente numa proposta de modelagem de Documentos Estruturados Multimídia.

6.3 Estrutura Lógica Básica (Ponto de Partida).

Uma das características (se não a mais importante) do padrão ODA/ODIF é a divisão Apresentação-Estrutura-Conteúdo que faz dos documentos, a qual é conveniente de adotar no meta-modelo OHypA pelos motivos já citados também no capítulo 3.

Outras características importantes de tal padrão de documentos são: facilidade de definir composição de objetos, estruturas projetadas para controlar que o documento cumpra um estilo ou tipo requerido e, principalmente, o fato de ser um padrão para o intercâmbio de documentos entre sistemas heterogêneos, o que facilitará a comunicação entre sistemas de Gerência de Documentos Estruturados Multimídia.

Por tudo isto, se tomará como base ou ponto de partida do meta-modelo OHypA, a estrutura lógica genérica definida em ODA/ODIF e, no decorrer do trabalho, serão acrescentados a ela conceitos e características, que transformarão o meta-modelo numa extensão ao padrão ODA/ODIF.

Tal padrão define três tipos de objetos lógicos: **Objeto Documento**, **Objeto Lógico Composto** e **Objeto Lógico Básico**.

A combinação de tais objetos resulta na estrutura lógica genérica apresentada na figura 6.4.

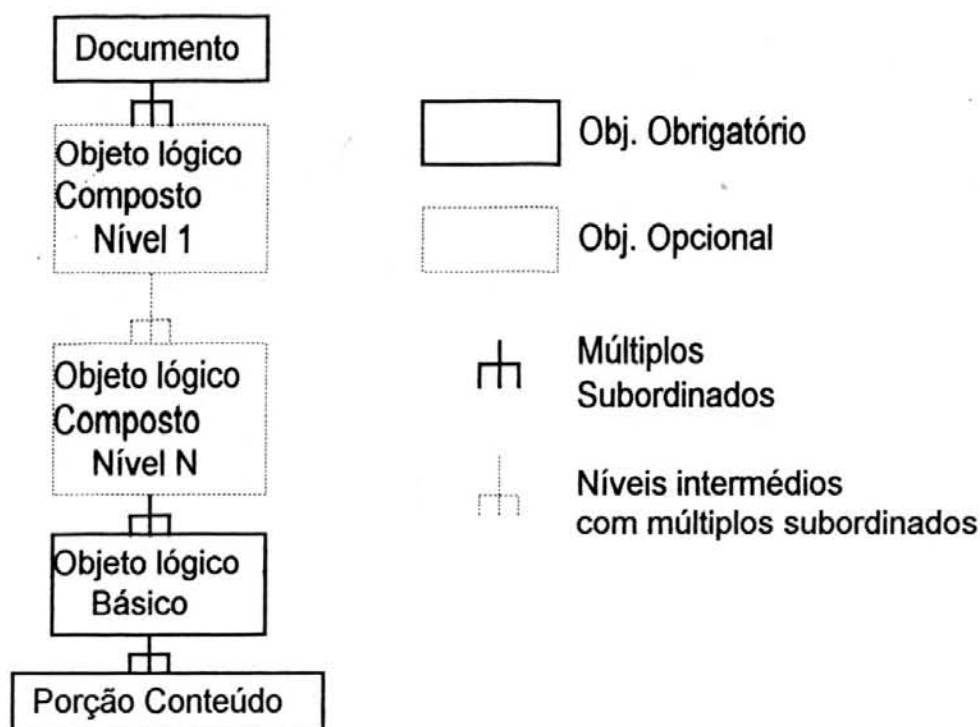


Figura 6.4: Estrutura Lógica Genérica de um Documento segundo ODA/ODIF.

Um exemplo de documento, segundo tal estrutura lógica genérica, é apresentado na figura 6.5, que somente considera a composição ou formação de um documento, ou seja um artigo está composto de um título, de um nome de autor e de um conjunto de seções, as quais, por sua vez, estão formadas por um conjunto de subseções, e assim por diante. Tal estrutura não trata qualquer aspecto hipertextual, de autoria, de segurança ou de versões.

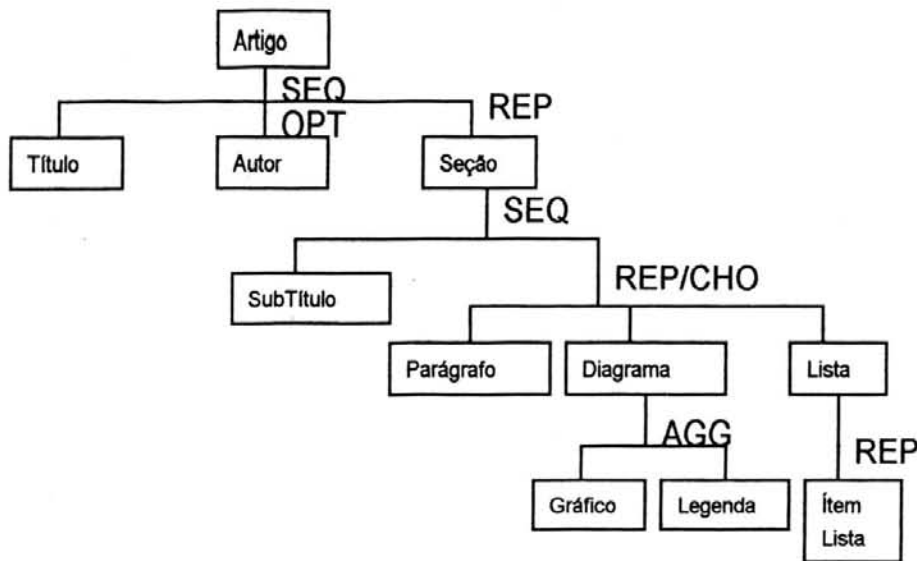


Fig. 6.5: Exemplo da Estrutura Lógica Genérica de um documento do tipo Artigo.

6.4 Modelagem de Referências Internas.

O primeiro problema detectado no modelo de documentos do padrão ODA/ODIF é quando se quer realizar relações lógicas entre porções ou componentes do documento. Por exemplo, o que acontece quando, em um parágrafo, o autor quer incluir uma referência bibliográfica ou uma referência a uma figura?

O modelo original, como foi apresentado, não permite ou não tem o poder de modelagem suficiente para representar estas situações.

Estas relações lógicas arbitrárias (criadas de acordo com o critério do autor do hiperdocumento) são chamadas Referências Internas, pois elas permitem, a partir de um componente lógico do hiperdocumento (parágrafo, figura, tabela, etc.) fazer referência a um outro componente lógico do mesmo hiperdocumento.

Pode-se deduzir que, dependendo dos componentes referenciados, existem diferentes tipos de referências internas, como por exemplo referências bibliográficas, referências a tabela, a figuras, a parágrafos, a fórmulas, etc., mas no nível do hiperdocumento abstrato do OHypA, elas são identificadas só como referências internas e, no nível concreto/genérico são especializadas segundo tais tipos.

Para possibilitar a representação de tais referências internas no meta-modelo, definem-se dois novos tipos de objetos lógicos:

- **Objeto Âncora.**
- **Objeto Referência Interna.**

A definição dos dois objetos foi motivada pela necessidade de manter a divisão ou limite entre a estrutura lógica do hiperdocumento e o conteúdo do mesmo.

Isto é conseguido pois, o objeto âncora é uma entidade de endereçamento indireto, ou seja, representa os botões ou pontos de partida localizados dentro do conteúdo de um componente lógico do hiperdocumento que conduzirão à referência citada.

O **objeto âncora** consta de uma lista de triplas do tipo:

- **Valor âncora**
- **Identificador referência.**
- **Direção.**

O primeiro componente é um valor arbitrário que especifica alguma localização, região, ítem ou *string* dentro da porção de conteúdo associada ao componente lógico (uma palavra, uma frase, uma parte de um gráfico, etc.).

O **identificador da referência** identifica univocamente o objeto Referência Interna associado a tal âncora, que forma parte da estrutura lógica do hiperdocumento, e não do conteúdo.

O último atributo do **objeto âncora**, a **direção**, indica se a âncora é origem ou destino da referência interna, permitindo desta forma, a modelagem de relações lógicas bidirecionais.

Por sua vez, o **objeto referência interna** forma parte da estrutura lógica do hiperdocumento, sendo independente dos conteúdos associados aos objetos lógicos ligados por ele.

O referido **objeto referência interna** está composto por:

- **Identificador de referência**, o qual representa univocamente o objeto. É aquele utilizado no **objeto âncora**.

- Um conjunto de identificadores de objetos lógicos básicos (parágrafos, figuras, tabelas, fórmula, etc.), que representam os objetos ligados ou relacionados pela referência interna. Este conjunto permite que as referências internas possam ser de cardinalidade variável (um para um, um para muitos, muitos para um, muitos para muitos).

- **Especificador de Apresentação**, o qual é um apontador ao objeto da estrutura de apresentação ligado à referência interna. Ou seja, mediante este apontador define-se a aparência ou apresentação que terá tal referência.

A definição das referências internas mediante a utilização destes dois objetos permite manter uma divisão entre a estrutura do hiperdocumento e seu conteúdo associado.

Por exemplo, o caso em que se modifica o valor ou posição de uma âncora ou botão localizado em um parágrafo. Isto simplesmente gerará uma modificação no objeto **âncora** (o qual está associado à porção de conteúdo correspondente), mas nenhuma mudança acontecerá no objeto **referência interna** associado.

O meta-modelo, resultante da adição destes dois novos objetos à estrutura lógica básica, está representado na figura 6.6.

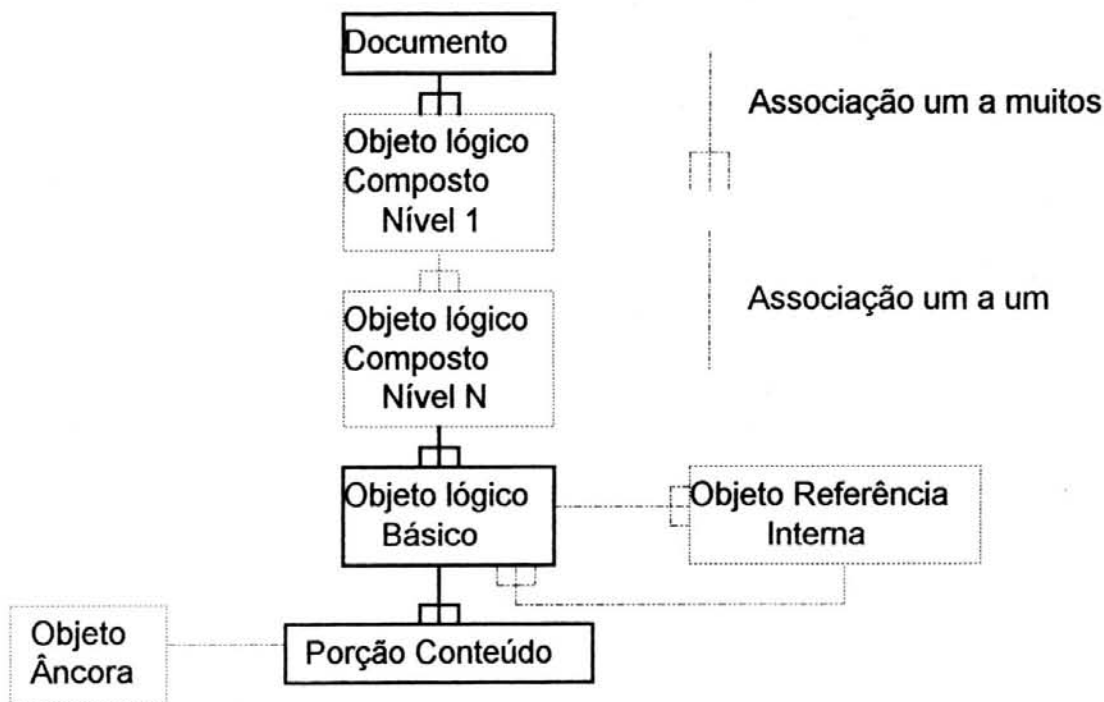


Figura 6.6: Referências Internas no meta-modelo OHypA.

Para exemplificar o referido meta-modelo, considere-se o seguinte caso: o autor de um hiperdocumento está escrevendo o parágrafo X (objeto lógico básico) e, no mesmo, inclui uma referência à figura 1, que pertence ao mesmo hiperdocumento. Logo, no mesmo parágrafo, incluirá uma referência à tabela A, que também pertence ao mesmo hiperdocumento. Mais adiante, o autor fará uma relação da figura 1 à tabela A.

Esta situação é representada na figura 6.7, na qual, por motivos de clareza, não são representados os objetos lógicos compostos que contêm aos objetos parágrafos, figura, tabela, os quais já foram representados na figura 6.5.

Também na figura 6.7 são ilustrados os atributos e valores correspondentes a cada objeto em questão.

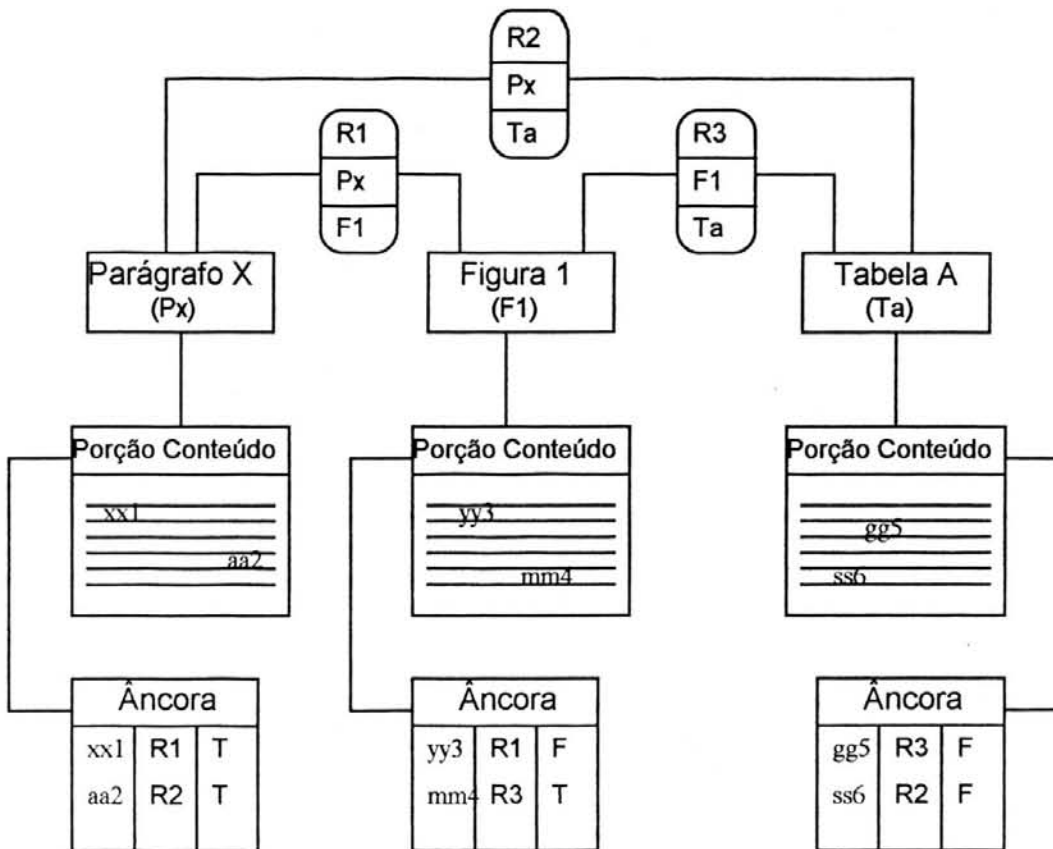


Figura 6.7: Representação das Referências Internas segundo OHypA.

Para finalizar, a definição dos dois novos objetos **Âncora** e **Referência Interna**, e a relação deles com a estrutura lógica e com as porções de conteúdo associadas do hiperdocumento, permitirá a modelagem eficiente de qualquer relação lógica arbitrária entre componentes de um mesmo hiperdocumento por parte do autor.

6.5 Representação de Referências Externas.

Assim como o autor de um hiperdocumento pode ter a necessidade de realizar ligações ou referências entre partes arbitrárias dele, ele também pode desejar, em qualquer momento da edição, referir-se à informações contidas em um outro hiperdocumento do banco de documentos.

Esta relação, a qual define uma ligação arbitrária entre um componente lógico de um hiperdocumento (parágrafo, figura, tabela, etc.) e um outro hiperdocumento, é denominada Referência Externa. Por exemplo, quando o autor de um documento inclui dentro de um parágrafo uma referência do tipo "*ver relatório técnico número 21, capítulo 2*".

O motivo da necessidade de contar com este recurso é oferecer ao autor a possibilidade de relacionar uma idéia, frase ou informação a um outro hiperdocumento que amplia ou que tem uma relação com a idéia expressada.

Estas referências externas podem ser divididas em duas classes:

- **Referências Externas Parciais:** Através delas se faz referência a um hiperdocumento como um todo, ou seja, identificam somente qual documento é referenciado, e não um componente específico dele. Por exemplo, numa figura pode-se colocar "*ver artigo 'Object Oriented Design'*".

- **Referências Externas Totais:** Neste caso, elas fazem referência a um hiperdocumento e a um componente dele em particular. Ou seja, é um par ordenado

Documento-Componente. Um exemplo seria: "ver Artigo 'Multimídia e Educação', capítulo 4".

A representação e modelagem destas referências externas é feita através de:

a- A definição do objeto **Referência Externa**, o qual está composto de:

- **Identificador de referência**, através do qual pode-se referir univocamente tal objeto.
- **Identificador do componente lógico origem**, ou seja, de onde parte a referência.
- **Identificação do destino**, que é o identificador do perfil do hiperdocumento, para o caso das referências externas parciais, ou um par ordenado identificador do perfil-identificador de objeto lógico, para o caso das referências externas totais.
- **Especificador de Apresentação**: Do mesmo modo que nas referências internas, este atributo relaciona o objeto lógico **referência externa** a um objeto de apresentação, o qual define a aparência ou apresentação da ligação externa.

b- A utilização do objeto **Âncora** já definido para as referências internas. Tal reutilização é justificada pela similaridade lógica entre os botões ou âncoras das citadas referências internas e as referências externas. A tal fim, acrescenta-se um atributo ao objeto **Âncora**, **Tipo Referência**, o qual especifica se a mesma é externa ou interna.

A divisão em duas partes da definição de referência externa, através do objeto âncora e o objeto referência externa, contribui para manter a independência entre conteúdo-estrutura já expressada em seções anteriores.

c- A utilização do objeto lógico **Perfil do Documento**, definido no padrão original ODA/ODIF. Como foi visto no capítulo 3, tal objeto **perfil do documento** contém informação que possibilita a manipulação do hiperdocumento como um todo. Ele consiste de um conjunto de atributos no nível mais alto da estrutura do hiperdocumento.

Para o tratamento das referências externas, adiciona-se um novo atributo, chamado **Citações** que é uma lista de pares do tipo:

- Identificador de referência externa.

- **Direção** (*to* para quando o documento é origem da referência, ou seja, ele cita um outro documento ou *from*, quando o documento é destino de uma referência externa, quer dizer, quando ele é citado por um outro documento).

Através deste novo atributo do objeto **Perfil do Documento**, se tem conhecimento de todas as referências externas contidas pelo hiperdocumento e as referências que fazem a ele. Esta informação é útil no caso, por exemplo, de eliminação de documentos citados por outros dentro do banco de documentos.

Neste caso, pode-se transformar automaticamente as referências externas em referências bibliográficas. Por exemplo, um documento X pode fazer uma citação no capítulo 2 ao documento Y e, no capítulo 4, aos documentos W e Z. O perfil de tal documento X, no atributo **citações**, terá três pares de valores:

-referência externa ao doc. Y - TO

-referência externa ao doc. W - TO

-referência externa ao doc. Z - TO

Da mesma forma, os atributos **citações** dos perfis dos documentos Y, W, e Z terão um par de valores:

-referência externa ao doc. X - FROM

A inclusão destes novos conceitos ao meta-modelo OHypA é mostrada na figura 6.8.

Por sua vez, a figura 6.9 dá um exemplo de modelagem de referência externas utilizando tais conceitos.

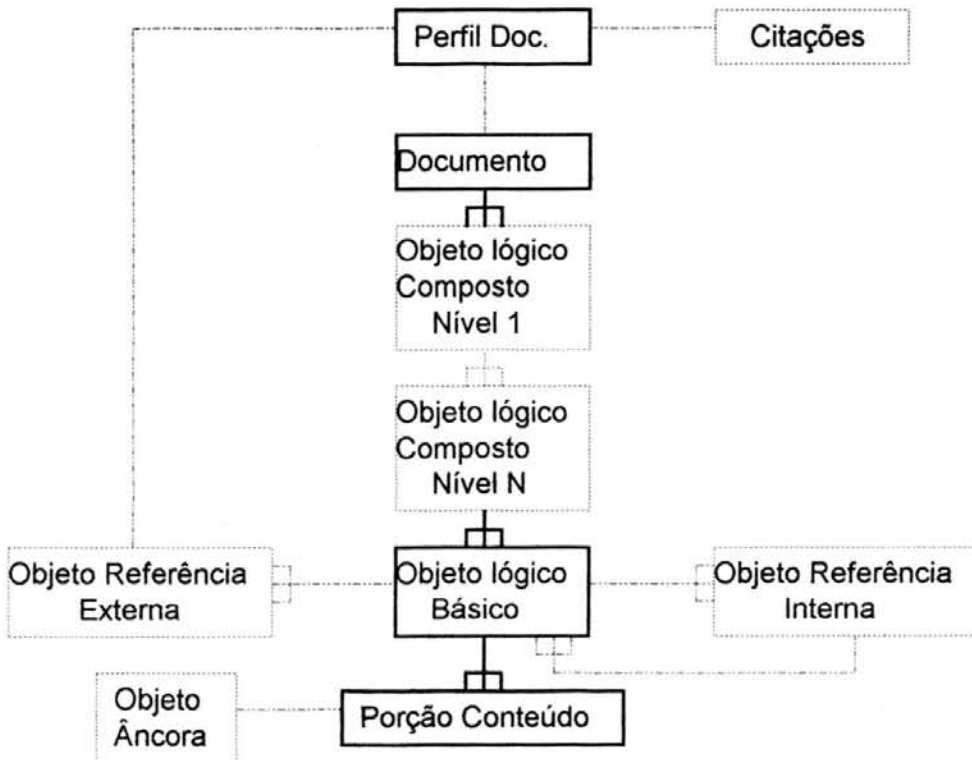


Figura 6.8: Meta-modelo OHypA com representação de Ref. Externas.

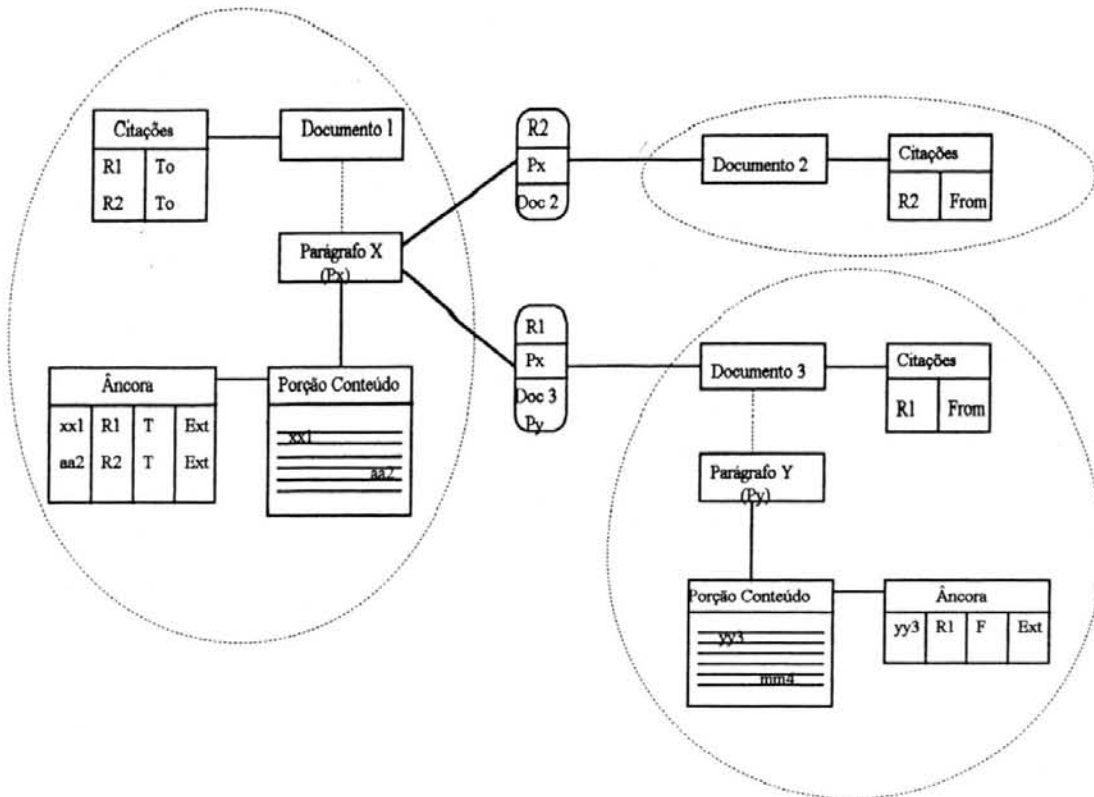


Figura 6.9: Referências Externas segundo o meta-modelo OHypA.

6.6 Representação de Anotações e/ou Comentários.

Outro aspecto não considerado no modelo original do padrão ODA/ODIF, mas muito importante na modelagem de Documentos Estruturados Multimídia, é a representação de anotações e/ou comentários feitos no hiperdocumento tanto pelo autor como por usuários dele.

Considere-se que o autor de um hiperdocumento, ao estar editando o conteúdo de uma seção, deseja incluir uma nota de rodapé de página, que aclara ou amplia alguma informação; ou deseja colocar algum comentário com respeito a um assunto tratado em tal seção. Outra situação poderia ser o caso que um leitor ocasional do hiperdocumento deseje colocar alguma anotação pessoal, que auxiliará em seu trabalho sobre tal hiperdocumento.

A diferença básica entre as referências internas, tratadas em seções anteriores, e este tipo de citação a anotações, é que enquanto nas referências internas são ligados ou relacionados dois ou mais componentes lógicos do hiperdocumento, nas citações a anotações adiciona-se, de alguma forma, informação que amplia a contida no hiperdocumento.

Para a modelagem das anotações e/ou comentários, definem-se dois novos tipos de objetos:

- **Objeto Anotação.**
- **Objeto Comentário.**

O objeto **Anotação**, basicamente cumpre a mesma função que o objeto âncora nas referências internas e externas, ou seja, representa os botões que ativam o objeto comentário associado a ele.

Este objeto está composto de uma lista de triplas de valores do seguinte tipo:

- **Valor Anotação.**
- **Destinatário.**
- **Identificador Comentário.**

O **valor da anotação** representa o botão e especifica alguma localização, região, item ou *string* dentro do conteúdo do componente lógico (pode ser uma palavra, uma frase, uma parte de uma figura, etc.).

O atributo **destinatário** é utilizado quando o autor da anotação designa uma pessoa em específico, a qual poderá ler tal anotação. Um exemplo que reflete esta situação

é quando um artigo está sendo escrito por dois autores e um deles, depois de trabalhar em alguma seção do mesmo, deixa algum comentário ou idéia, para que o colega leve em conta quando estiver trabalhando no hiperdocumento.

O último atributo, **identificador de comentário**, simplesmente liga o botão da anotação ao objeto **comentário** associado.

Por sua vez, tal objeto **comentário** trata-se na realidade de um objeto do tipo **porção de conteúdo**, definido no padrão ODA/ODIF original. Ao ser deste tipo, mantém-se a divisão estrutura-conteúdo e possibilita-se que o conteúdo do comentário seja de qualquer tipo (gráfico, texto, animação, voz, etc.), dando aspecto multimídia ao tratamento de anotações e/ou comentários. Um gráfico, por exemplo, pode conter uma anotação textual que auxilie à compreensão do mesmo, ou um parágrafo pode conter um comentário falado feito pelo autor do mesmo, ou ainda uma fórmula pode ter um comentário que seja uma imagem da resolução da mesma.

A adição destes objetos no meta-modelo OHypA é mostrada na figura 6.10, enquanto que a figura 6.11 apresenta um exemplo utilizando tais objetos do meta-modelo.

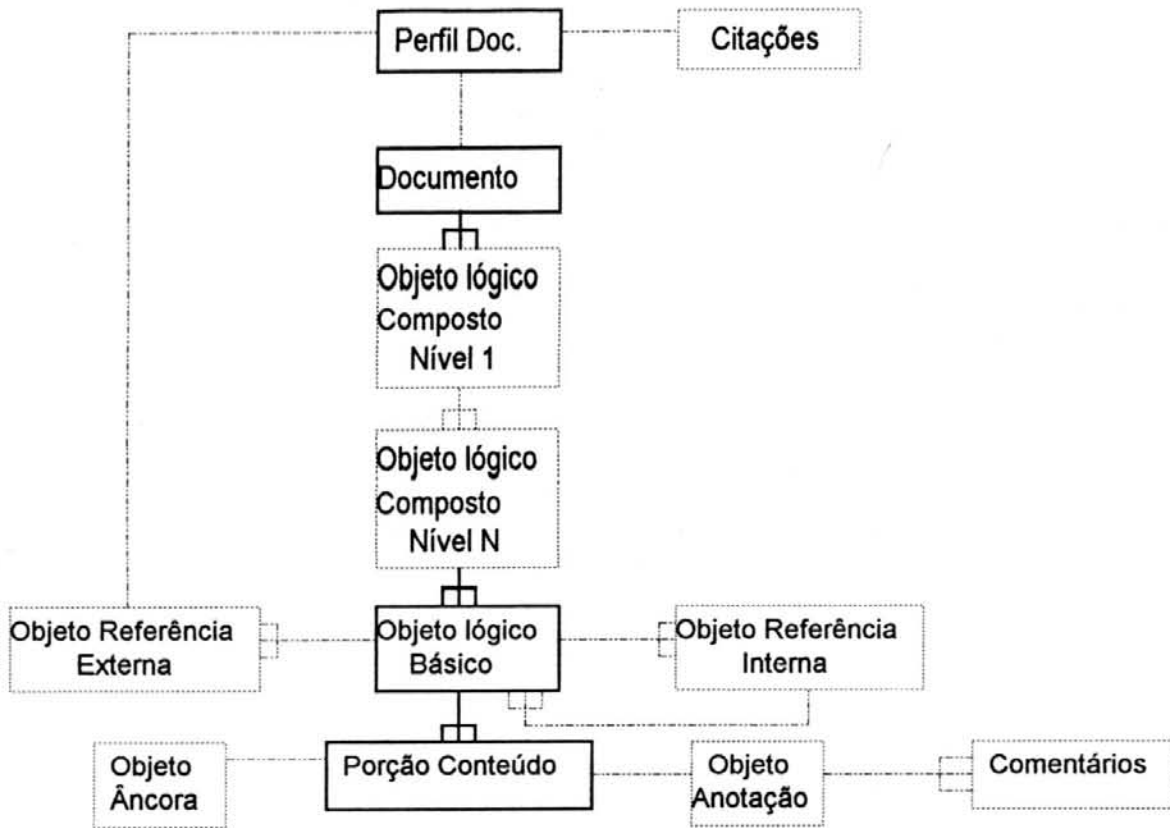


Figura 6.10: Anotações e/ou Comentários no meta-modelo OHypA.

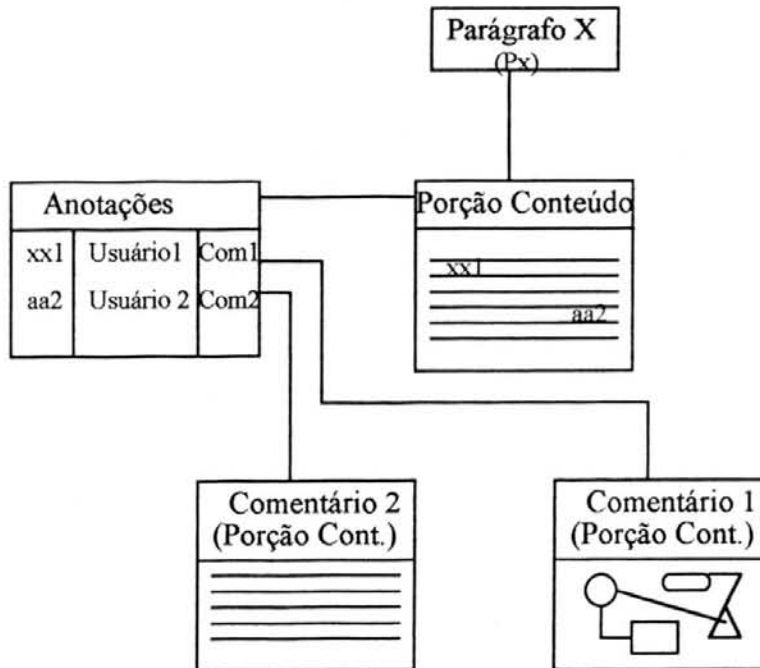


Figura 6.11: Exemplo da representação de anotações segundo OHypA.

No próximo capítulo, além de tratar aspectos avançados da modelagem de Documentos Estruturados Multimídia a ser incluídos no meta-modelo OHypA, serão especificados detalhadamente todos os objetos definidos no meta-modelo e apresentados nestes dois capítulos.

Tal especificação incluirá a definição precisa dos atributos e procedimentos de cada objeto, a identificação das abstrações de baixo e alto nível e a classificação das ligações ou relações presentes no meta-modelo.

7 OHypA, ASPECTOS AVANÇADOS.

7.1 Introdução.

O meta-modelo OHypA, como já foi apresentado no capítulo 6, só permite a modelagem e tratamento de Documentos Estruturados Multimídia desde uma visão simples, ou seja, criados e manipulados por um único usuário (o autor).

Na realidade, como também foi visto em capítulos anteriores, um sistema de Gerência de Documentos Estruturados Multimídia deve permitir o tratamento de um conjunto de documentos, contidos num banco de documentos, por um conjunto de usuários, que trabalham cooperativamente.

Em consequência, aqui surge a seguinte pergunta: "O que acontece quando o meta-modelo OHypA, que foi visto até agora, é inserido em um ambiente de múltiplos usuários, onde cada um deseja (e deveria poder) manipular as informações contidas nele?".

A resposta, evidentemente, será que no estágio atual de desenvolvimento, tal meta-modelo não suporta este tratamento. A solução, então, é a adição de novos conceitos e idéias, que permitirão a modelagem e tratamento de tais características.

No seguir deste capítulo, serão apresentados e justificados os conceitos de visões em Documentos Estruturados Multimídia, autoria/segurança, tratamento de versões, tipos de ligações, etc., os quais, adicionados aos conceitos já definidos e tratados no capítulo anterior, conformarão finalmente o meta-modelo OHypA completo.

7.2 Visões em Documentos Estruturados Multimídia.

Como foi visto no capítulo anterior, os Documentos Estruturados Multimídia são representados através de sua estrutura lógica, sua estrutura de apresentação e por seu conteúdo multimídia associado.

Por sua vez a estrutura lógica, a qual está representada através do meta-modelo OHypA, está formada por dois tipos de relações:

a- Relações de Composição: indicam como se relacionam os objetos lógicos do documento para formar outros objetos lógicos compostos. Estas relações formam uma estrutura hierárquica.

b- Relações lógicas hipertextuais: representam as ligações lógicas feitas pelo autor do documento, de acordo a suas necessidades de informação. Estas ligações, como já foi visto, podem ser:

-Referências Internas: relações entre objetos lógicos de um mesmo documento

-Referências Externas: relações entre objetos lógicos de diferentes documentos.

-Anotações e/ou Comentários: relações entre porções de conteúdo de um mesmo documento.

Estas ligações, por sua vez, formam geralmente uma estrutura de tipo **rede**.

Pode-se dizer, então, que a estrutura lógica dos Documentos Estruturados Multimídia resulta, na realidade, da combinação destas duas estruturas. Uma, a **hierarquia**

que representa a composição do documento, e uma outra de tipo **rede**, que representa as relações hipertextuais entre os objetos lógicos do documento.

Neste trabalho, define-se como **visão** o conjunto de relações hipertextuais presentes no documento. Ou seja, tal **visão** é, na realidade, uma sub-rede da estrutura lógica do documento, a qual está formada pelas referências internas, externas e anotações e/ou comentários feitos pelo autor no documento.

Definido isto, torna-se necessário voltar à pergunta formulada na seção 7.1: "O que acontece quando o meta-modelo OHypA é inserido num ambiente de múltiplos usuários, onde cada um deseja (e deveria poder) manipular as informações contidas nele?".

Como um primeiro passo para idealizar uma proposta que responde a esta questão, considere-se o seguinte exemplo:

O autor XX cria um novo documento multimídia NN, que contém referências internas, referências externas a outros documentos do banco de documentos, e anotações feitas por ele, as quais são consideradas apropriadas para seu trabalho.

Neste ponto, o documento multimídia NN possui uma **visão**, sobre sua estrutura lógica, do ponto de vista do autor XX (definida por ele).

No momento que este documento é integrado ao banco de documentos, ele pode ser acessado por um outro usuário YY. Tal usuário poderá percorrer a estrutura modelada pelo autor do documento, inclusive a **visão** definida por ele. Mas ele também deveria poder definir novas relações lógicas (referências internas, externas e anotações) independentes daquelas definidas pelo autor, as quais refletirão o seu próprio interesse no tratamento da informação contida no documento multimídia.

A forma de permitir a definição de novas relações lógicas no documento por parte de um outro usuário diferente do autor, é mediante a definição de diferentes **visões** sobre o documento, correspondendo cada uma a um usuário em particular.

Estas **visões** sobre o documento correspondem, na realidade, a sub-redes definidas pelos usuários sobre a estrutura lógica original. Isto significa que cada uma destas **visões** ou sub-redes é acrescentada a tal estrutura, possibilitando a livre definição, por parte dos diferentes usuários, de relações lógicas hipertextuais sem modificar a estrutura lógica original definida pelo autor do documento.

A figura 7.1 apresenta graficamente a idéia de visões em documentos estruturados multimídia.

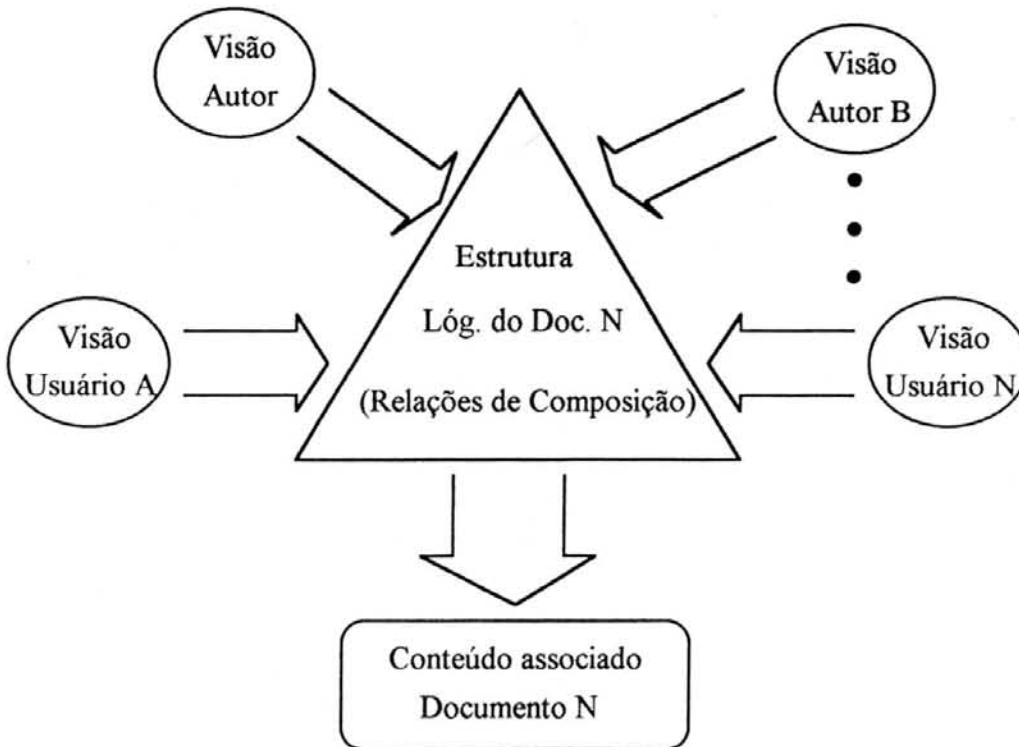


Figura 7.1: Visões em Documentos Estruturados Multimídia.

Para viabilizar o conceito de **visões**, é necessário considerar e definir os seguintes aspectos, que formam parte da autoria e segurança de documentos estruturados multimídia.

a- Nível de Usuário.

Como foi visto no exemplo apresentado, definem-se dois tipos de usuários:

1- **AUTOR**: é o criador do documento. Ele tem controle total sobre a estrutura lógica do documento (tanto a hierarquia de composição quanto as relações lógicas hipertextuais (visão) definidas por ele). Também é o único autorizado a modificar o conteúdo associado do documento (gerando novas versões, o que será tratado na próxima seção).

2- **LEITOR**: é aquele usuário que acessa e utiliza um documento que não é de sua autoria. Ele pode definir sua própria visão sobre o documento, tendo controle total sobre a mesma.

A fim de auxiliar no controle e segurança, o sistema de gerência de Documentos Estruturados Multimídia incluirá informações sobre os usuários do sistema.

b- Nível de visibilidade da visão.

Define para quem está habilitada e/ou disponível a visão em questão. Os três níveis identificados são:

1- **PÚBLICO**: a visão está disponível para qualquer usuário cadastrado no sistema.

2- PRIVADO: neste caso, a visão está disponível somente para o criador da mesma, sendo invisível para o resto dos usuários.

3- GRUPAL: a visão é visível para todos os usuários que pertencem ao mesmo grupo do autor da visão. Este nível permite, por exemplo, o trabalho e autoria cooperativa sobre documentos estruturados multimídia.

c- Operações permitidas.

As operações que podem ser realizadas sobre uma visão são: Criação, Modificação e Eliminação de referências internas, referências externas e anotações contidas na visão.

Como foi citado anteriormente, cada usuário possui controle total sobre sua visão, podendo realizar qualquer das operações mencionadas.

Aqueles usuários leitores de uma visão (habilitados pelo nível de visibilidade associado a ela) somente podem utilizar tal visão para navegar pela estrutura, sem modificar a mesma.

d- Elementos a manipular.

Como foi visto, dentro de sua visão o usuário pode fazer qualquer operação. Por outro lado, está proibido aos usuários leitores, modificar tanto o conteúdo do documento quanto a hierarquia de composição que define as relações entre os componentes do documento. Somente o autor pode trabalhar sobre o conteúdo (criando novas versões) e/ou sobre a hierarquia de composição (definindo novas organizações do documento).

e- Validade/Vigência.

Este aspecto se refere ao tempo de validade dos resultados das operações realizadas por um usuário em sua visão.

Identificam-se duas possibilidades:

1- TEMPORÁRIO: Os resultados das operações estarão vigentes enquanto o usuário estiver em sua sessão de trabalho sobre o documento. Uma vez concluída, os mesmos são eliminados.

2- PERMANENTES: Neste caso, os resultados das operações passam a formar parte, de forma permanente, da visão do usuário.

Considere-se o seguinte exemplo para aclarar o dito anteriormente: o usuário A, ao começar seu trabalho sobre o documento N, acha necessário criar temporariamente um comentário, depois, no decorrer do trabalho, cria permanentemente uma referência interna. Por último, modifica temporariamente uma referência externa presente em sua visão.

Logo que o usuário conclui e encerra sua sessão de trabalho, o sistema não considerará nem o comentário feito transitoriamente nem a modificação temporária da referência externa. O que terá que fazer é adicionar, à visão do usuário, a nova referência interna criada durante a sessão.

Identificados os aspectos considerados importantes para o tratamento de visões em Documentos Estruturados Multimídia, o próximo passo é definir como tais visões serão representadas no meta-modelo OHypA.

Para possibilitar a definição de uma visão por usuário, de forma que tais visões sejam independentes umas de outras, cada usuário que defina alguma relação hipertextual sobre um objeto lógico básico terá associado, a tal objeto, um objeto **Âncora** (que terá as referências internas e externas definidas por ele nesse objeto) e um objeto **Anotação** (que terá, por sua vez, os comentários feitos por ele)

Como consequência, na definição dos objetos **Âncora** e **Anotação** são adicionados os atributos:

Autor: Identificação do usuário proprietário da visão.

Data: da criação da visão.

Agora, cada objeto porção de conteúdo ligado a um objeto lógico básico terá associado um conjunto de objetos **Âncoras** e um conjunto de objetos **Anotação**. Estes conjuntos representam, na realidade, as visões associadas a tal objeto lógico básico.

Também é definido um novo tipo de objeto, denominado **Usuário**, que estará referenciado pelos atributos **Autor** e **Leitores**, pertencentes ao perfil do documento.

Este novo objeto **Usuário** terá os seguintes atributos:

- **Identificador:** atributo que caracteriza univocamente o usuário.
- **Grupo:** Identifica o grupo ao qual ele pertence. Este atributo auxilia na identificação dos usuários que formam parte do referido grupo.
- **Visibilidade:** Define o nível de visibilidade da visão associada ao usuário (pública, privada ou grupal).

- **Citações:** Este atributo, que foi definido no capítulo anterior como parte do objeto **Perfil do Documento**, agora é deslocado para este objeto, uma vez que as referências externas feitas por cada usuário formam parte de suas respectivas visões, e não do documento em geral.

Desta forma, passam a existir dois níveis de segurança:

1- Nível do documento, como um todo, no qual o autor define se o mesmo é público, privado ou grupal.

2- Nível das visões particulares de cada usuário. Neste caso, cada usuário que acessa ao documento terá conhecimento somente de aquelas visões definidas como públicas ou daquelas cujos criadores são integrantes do mesmo grupo ao qual ele pertence.

A figura 7.2 apresenta o meta-modelo OHypA incluindo as representações dos novos objetos e atributos que possibilitam a modelagem das diferentes visões dos usuários.

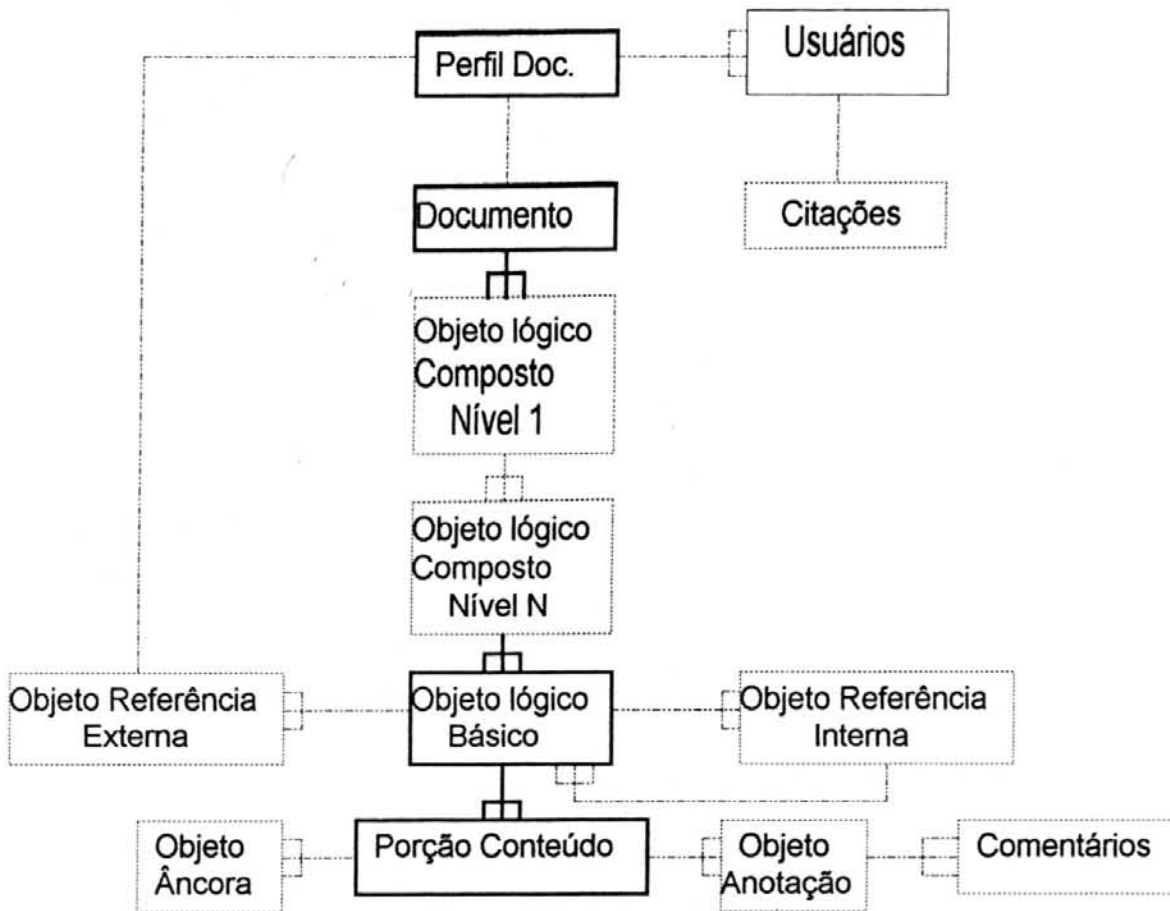


Figura 7.2: Meta-modelo OHypA, incluindo representação de visões.

Por outro lado, na figura 7.3 é apresentado um exemplo, utilizando o meta-modelo OHypA, para a representação de visões em documentos estruturados multimídia.

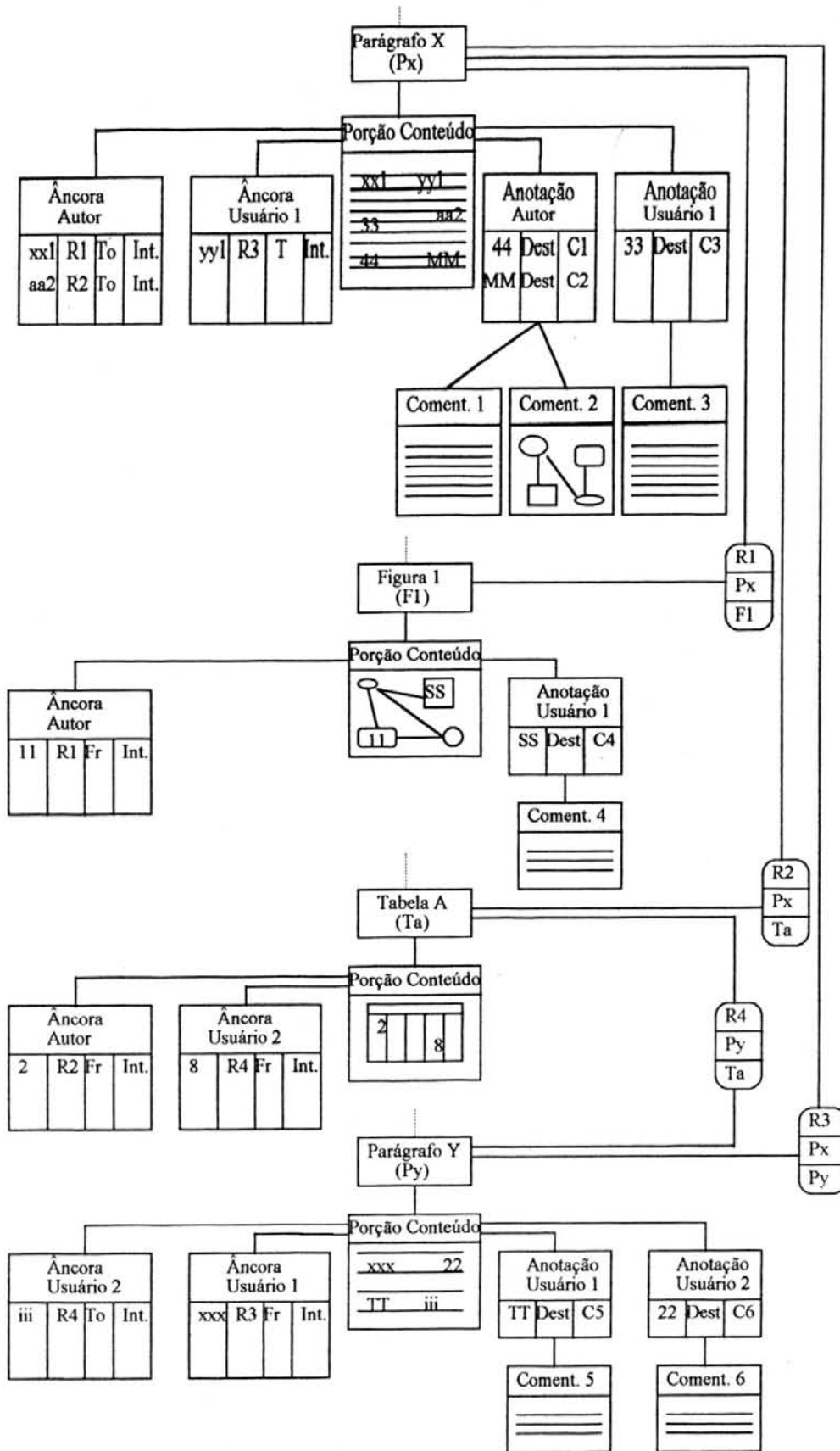


Figura 7.3: Exemplo de representação de visões segundo OHypA.

Como pode ser visto na figura 7.3, no parágrafo X o autor realiza uma referência à figura 1 (R1) e à tabela A (R2). Também o autor faz dois comentários, um C1 (textual) e um outro C2 (gráfico).

No mesmo parágrafo X, o usuário 1 cria sua própria visão, na qual define uma referência ao parágrafo Y (R3) e um comentário C3.

O usuário 1 também tem visões sobre a figura 1, na qual define o comentário C4, e sobre o parágrafo Y, na qual define o comentário C5.

Por último, no parágrafo Y um outro usuário 2 define sua visão, na qual cria uma referência (R4) à tabela A e um comentário C6, os quais são úteis para seu trabalho.

Resumindo o apresentado no exemplo, três usuários diferentes (o autor, e os leitores 1 e 2) têm suas próprias visões de um mesmo documento estruturado multimídia, sendo que além de ser independentes umas de outras, elas adicionam informação à estrutura lógica original.

Cada usuário poderá utilizar sua própria visão e aquelas nas quais ele está habilitado. Este controle será executado pelo sistema a partir das informações que possui o modelo em relação a cada usuário e ao nível de visibilidade de suas respectivas visões, informação que está contida no perfil do documento, nos atributos **autor** e **leitores**, como já foi citado.

7.3 Tratamento de versões.

Como foi considerado até agora, o meta-modelo OHypA associa cada objeto lógico básico de um documento multimídia (parágrafos, figuras, tabelas, fórmulas, etc.) a uma porção de conteúdo específica. Na verdade, esta situação não é realmente a mais apropriada para modelar o processo de autoria de documentos estruturados multimídia.

Por exemplo, no estado atual de desenvolvimento do meta-modelo, poderia acontecer a seguinte situação: se um autor, depois de ter criado o documento, deseja modificar parcial ou totalmente o conteúdo do mesmo (adicionar, mudar e/ou apagar trechos de informação), terá que optar por uma das duas opções seguintes:

- que a modificação criará um outro novo documento, ou seja, o documento sobre o qual se realiza a operação terá duas instâncias, uma representando o estado do mesmo antes de sofrer as modificações, e a outra representando o estado do documento depois de aplicar as operações sobre ele.

- as modificações serão realizadas sobre o documento, sem ter como resultado a criação de um outro novo. Neste caso, não se poderá ter nem o estado anterior do documento nem as informações e/ou ações que originaram tais modificações.

Como pode-se apreciar, estas duas opções são decididamente indesejáveis em um ambiente de autoria de documentos multimídia. A primeira implicará em uma multiplicação de documentos não necessária, pois por cada modificação simples realizada sobre um documento será criada uma cópia atualizada do mesmo.

Na segunda, o autor e os leitores em geral, só terão acesso à versão corrente do documento, não tendo a possibilidade de dispor dos estados anteriores do mesmo, nem

da história das modificações sofridas por ele, informação esta muito importante num processo de autoria.

Todas estas desvantagens poderiam ser solucionadas se o meta-modelo OHypA modelasse e suportasse o tratamento de versões em documentos estruturados multimídia. Ou seja, permitir a modelagem na própria estrutura do documento tanto da história das modificações do mesmo quanto os estados pelos quais foi passando durante o processo de autoria.

Devido a tudo que foi citado anteriormente, o objetivo agora é enriquecer o meta-modelo OHypA com conceitos e características referentes ao tratamento de versões, o qual será realizado no decorrer desta seção.

Algumas das primeiras conclusões que podem ser deduzidas até aqui, que contribuirão à idealização de uma proposta que resolva a modelagem e tratamento de versões em documentos estruturados multimídia são:

a- Os objetos lógicos básicos dos documentos terão que ter associados um conjunto de versões, usualmente ordenado pelo tempo, ao invés de um único objeto porção de conteúdo.

b- A criação de uma nova versão terá associada uma semântica definida, dentre uma série de possíveis. Por exemplo, uma que dirá que uma versão será criada a cada modificação realizada sobre o objeto em questão.

c- Uma versão do objeto porção de conteúdo representa um estado (instância) identificável, considerado semanticamente significativo pelo usuário.

Segundo [GOL93a, GOL93b], podem ser identificados dois níveis de gerência de versões:

1- Nível da Aplicação.

Este nível pode, frequentemente, utilizar a gerência de versões do nível do sistema (hierarquicamente inferior). Quando isto não é possível, deve ser construído um modelo de versões de mais alto nível, independente do sistema.

Neste nível, o objetivo é suportar a representação de informações dependentes do tempo, conforme foram definidas pelos usuários ou pela aplicação. O usuário deve poder editar a seqüência de versões, removendo e até mesmo alterando qualquer uma das versões.

2- Nível do Sistema.

Aqui, o histórico de versões de um objeto reflete a seqüência de modificações realizadas sobre o mesmo, mas da maneira como foram percebidas pelo sistema. Neste nível, o objeto versionado não necessariamente corresponde a um objeto da aplicação, sendo aqui chamado de unidade de versionamento.

O objetivo aqui é concentrar-se na **gerência de versões a nível da aplicação**, a fim de enriquecer o meta-modelo OHypA e mantê-lo independente do sistema de banco de dados que dará suporte a ele.

No seguir desta seção, será proposto um modelo de versões, o qual baseando-se no citado anteriormente deverá oferecer:

1- Exploração de Alternativas: Possibilitar que os usuários explorem várias configurações alternativas de um mesmo hiperdocumento multimídia.

2- Manutenção do Histórico de um Hiperdocumento: Permitir aos usuários manter e manipular a história das modificações dos diferentes componentes do hiperdocumento.

3- Atualização Automática de Referências: A referência desde um componente lógico do hiperdocumento para o objeto versionado associado pode indicar uma versão específica da entidade, a versão mais recente da entidade ou alguma versão que cumpra uma propriedade em particular. Neste sentido, o modelo de versões deve permitir a atualização automática destas referências quando uma nova versão é criada.

4- Gerência de Representações de uma mesma Informação: É importante que o modelo tenha condições de captar a noção de que diferentes representações (em caso particular, diferentes meios) podem ser usadas para descrever uma mesma informação (como, por exemplo, versão escrita e versão falada de um parágrafo). Representações diferentes de uma mesma informação devem ser tratadas como diferentes versões desta informação.

Para conseguir tudo isto, no contexto do meta-modelo OHypA define-se um novo tipo de objeto composto, denominado **Contexto Versionado**. Este objeto representa, na realidade, um conjunto de versões (objetos **Versão**) de um objeto lógico básico em particular. Cada uma dessas versões é uma representante de tal objeto lógico básico.

O objeto **Contexto Versionado** possui uma identificação própria, de forma a permitir sua referência e manipulação. Ele apresenta propriedades comuns a todas as versões

associadas, mantém as informações sobre o conjunto de versões e pode apresentar propriedades próprias.

A figura 7.4 esquematiza a modelagem de versões em documentos estruturados multimídia. É notável a diferença entre a figura 7.1 (apresentada na seção 7.2) e a presente. Na primeira, os objetos lógicos básicos (estrutura lógica do documento) têm associados objetos porções de conteúdo (conteúdo do documento).

Agora, cada objeto lógico básico está associado a um objeto **Contexto Versionado**, que representa todas as versões pelas quais passou tal objeto lógico.

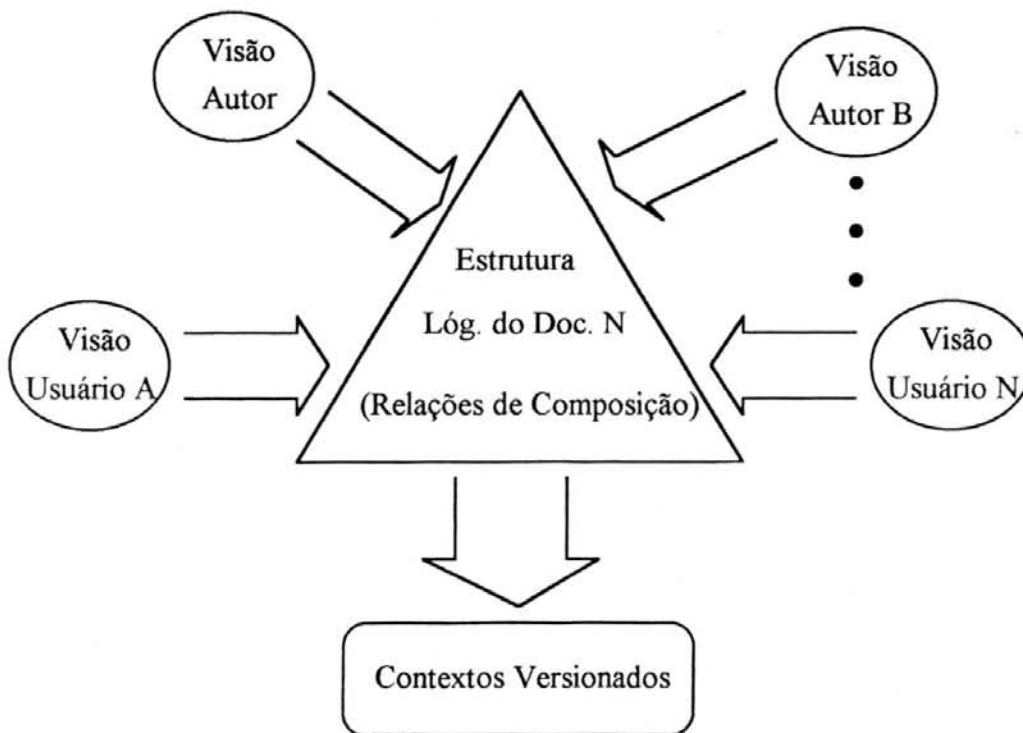


Figura 7.4: Versões em Documentos Estruturados Multimídia.

O tipo de referência definida neste trabalho, que relaciona cada objeto lógico básico (parágrafo, figura, tabela, fórmula, etc.) ao seu objeto Contexto Versionado, é denominada, segundo [GOL93a, GOL93b] **Ligação Dinâmica**.

Nela, só se faz referência ao objeto **Contexto Versionado**, sem saber, na realidade, qual é a versão atual. Ou seja, o próprio objeto **Contexto Versionado** possui informação que possibilita a identificação da versão atual correspondente. Tal informação é obtida mediante mecanismos que determinam qual a versão que deverá ser escolhida como atual.

Desta forma, um dos objetivos propostos é alcançado: a atualização automática de referências, pois a nível do meta-modelo, cada objeto lógico básico sempre fará referência a seu objeto Contexto Versionado, que por sua vez, estará sempre atualizado com a versão considerada atual.

Para isto, são definidos os seguintes atributos do objeto **Contexto Versionado**:

-**Identificador**: utilizado para referenciar o objeto.

-**Versão Atual**: Aponta a versão considerada corrente (última versão).

-**Primeira Versão**: Identifica a versão mais antiga do objeto em questão.

Esta informação é útil, uma vez que ela facilita a navegação através do histórico de versões.

-**No. versões**: Dá o número de versões associadas ao objeto contexto versionado.

Estes atributos são atualizados pelo sistema cada vez que o autor do hiperdocumento gera uma nova versão do componente.

7.3.1 Relacionamentos entre versões.

As relações existentes entre as diferentes versões de um objeto lógico básico do documento estruturado multimídia podem estar representando:

a- A evolução deste objeto através do tempo. Neste caso, a relação é chamada **Relação de Derivação**, pois cada nova versão é criada como sucessora da última e representa a evolução do objeto (parágrafo, figura, fórmula, etc.) no tempo.

Aqui, uma ligação $V1 \text{ -----} > V2$ significa que a versão $V2$ foi criada a partir da versão $V1$.

b- Diferentes alternativas de representação do mesmo objeto lógico composto. Ou seja, no contexto versionado existirão **versões correlatas**, as quais contêm a mesma informação, mas em diferentes meios (uma versão textual e uma versão gráfica, por exemplo).

Neste caso, a **Relação de Correlação** $V3 \text{ -----} > V4$ significa que a versão $V4$ representa, em outro meio, a informação contida na versão $V3$.

Com a definição destas relações, os objetos **versão** (que serão definidos a seguir) associados ao objeto **Contexto Versionado**, formam uma estrutura que reflete as relações existentes entre eles.

Tal estrutura é um grafo dirigido, no qual cada versão (objeto **versão**) pode possuir uma versão predecessora e/ou uma sucessora (segundo a relação de derivação), e uma série de versões correlatas (segundo a relação de correlação).

Com isto, pode-se navegar através do histórico de um componente em particular, e também do documento estruturado multimídia em geral, utilizando os mesmos mecanismos de navegação oferecidos na estrutura lógica. A figura 7.5 esquematiza as relações entre versões.

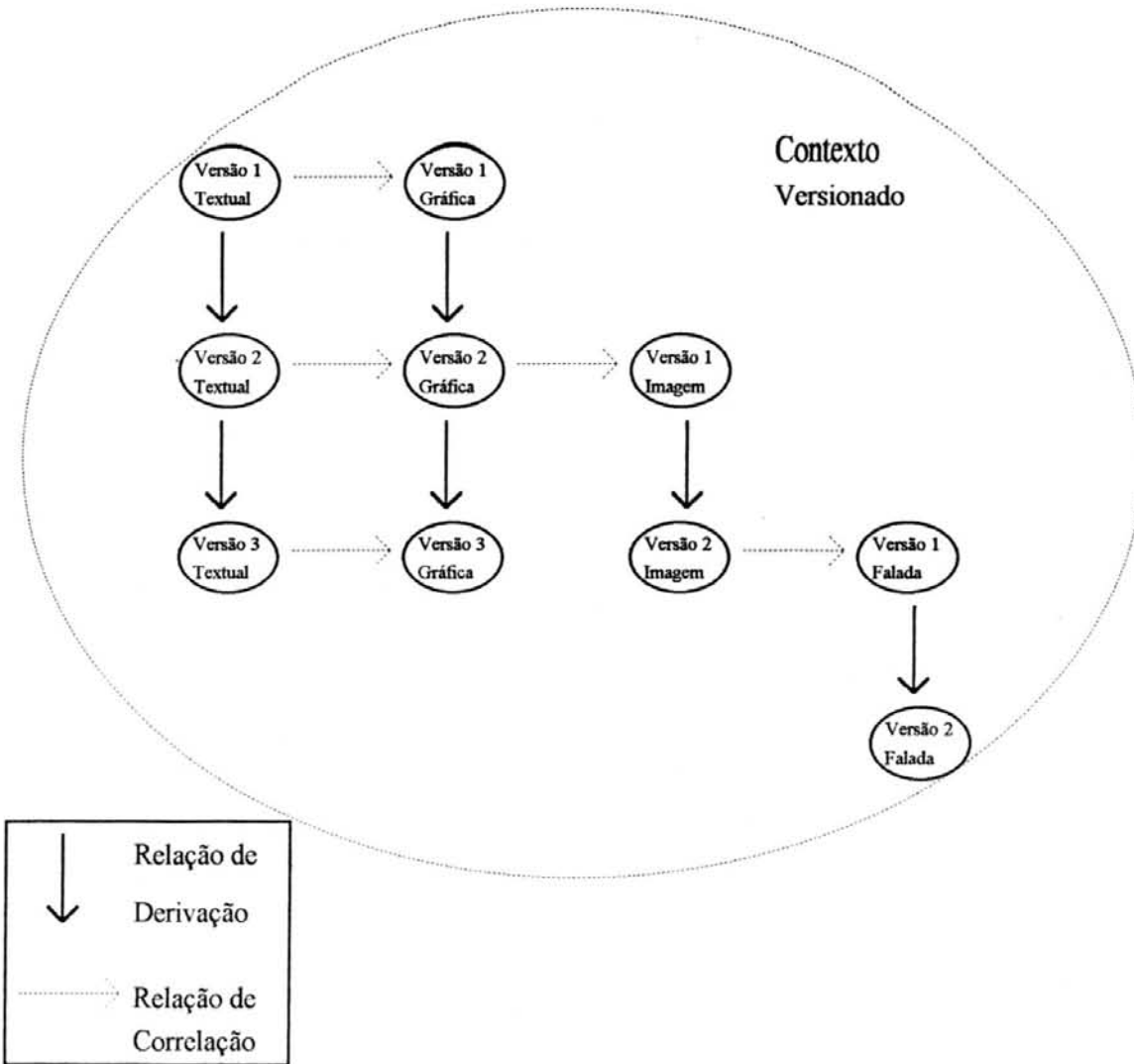


Figura 7.5: Estrutura formada pelos objetos Versão, mediante as relações de Derivação e de Correlação.

7.3.2 O Objeto Versão.

Como foi citado na subseção anterior, cada objeto lógico básico de um documento estruturado multimídia faz referência a um objeto composto denominado **Contexto Versionado**, que representa tanto a versão atual quanto a história de versões de tal objeto lógico.

Na realidade, para poder representar cada uma dessas versões associadas ao objeto **Contexto Versionado** é definido um novo objeto chamado **objeto Versão**.

Este objeto, substitui à porção de conteúdo associada a cada objeto lógico básico no meta-modelo OHypA definido no capítulo anterior, possibilitando agora a modelagem de versões no meta-modelo. Basicamente, este objeto está composto de:

- Um conjunto de objetos **Porções de conteúdo**, que representam o conteúdo propriamente dito associado ao objeto lógico (um parágrafo, por exemplo).
- Um conjunto de objetos **Âncoras**, que representam as referências internas e externas (visões) definidas pelos diferentes usuários sobre essa versão.
- Um conjunto de objetos **Anotação**, representando as anotações feitas por cada usuário sobre a versão.

Para obter todo o citado tanto nesta subseção quanto na anterior, são definidos os seguintes atributos no **objeto versão**:

- Identificador**: Identifica univocamente a versão em questão.
- Data**: Guarda informação do tempo de criação da versão.

-Estado: Atributo utilizado para controlar o tratamento da versão. Ele pode assumir os seguintes valores:

a- Temporária: a versão se encontra neste estado enquanto sofre modificações, ou seja, enquanto está sendo criada e/ou editada. As permissões e/ou restrições sobre as versões que estão neste estado são:

- Ela pode ser apagada.
- Não pode ser usada para derivar novas versões.
- Não pode ser usada pelos usuários para criar suas visões.

b- Permanente: a versão assume este valor quando se torna estável, isto quer dizer, quando o usuário termina o processo de criação da mesma. Neste estado, as versões:

- Não podem ser modificadas.
- Podem ser usadas para derivar novas versões.
- Podem ser usadas para definir versões correlatas a ela.
- Não pode ser destruída.
- Pode ser usada pelos usuários leitores para criar sua visões.

c- Obsoleta: é a forma de notificar aos objetos e/ou usuários que fazem referência a ela que pode ser apagada a qualquer momento.

-Porções Conteúdo: Referência ao conjunto de objetos porções de conteúdo associados à versão.

-**Âncoras:** Indica o conjunto de objetos âncoras associados aos objetos porções de conteúdo considerados no atributo anterior.

-**Anotações:** Faz referência aos objetos anotação associados às porções de conteúdo da versão.

-**Versão Anterior:** Indica a versão predecessora (segundo a relação de derivação).

-**Versão seguinte:** Indica a versão sucessora na estrutura de derivação do objeto versionado.

-**Correlata:** indica a versão correlata (diferente representação da mesma informação) que possui esta versão.

Complementando a definições acima mencionadas, define-se um novo atributo, **Última Versão**, nos objetos **Referência Interna**, **Referência Externa** e **Comentário**. O objetivo deste atributo é identificar a versão mais recente na qual aparece tal referência ou comentário, para evitar apagar relações hipertextuais de uma versão que também se encontram em outra versão mais nova.

Isto pode acontecer, pois os objetos **Referência Interna** e **Referência Externa** não estão associados a um objeto versão em particular, eles formam parte da estrutura lógica do documento multimídia, independente das versões.

Para esclarecer o uso deste atributo **última versão**, suponha-se o seguinte exemplo: o autor define, na primeira versão de um parágrafo X, três referências internas

R1, R2 e R3. Os três objetos **referência interna** têm, como valor do atributo **última versão** parágrafo X.

Após um certo tempo, o mesmo autor decide realizar uma nova versão do parágrafo X, identificada como parágrafo X', na qual mantêm as referências R1 e R2, mas decide apagar a referência R3 e adicionar uma nova R4. Agora, os objetos **referência interna** R1 e R2 terão no atributo **última versão** o valor Parágrafo X', ao igual que a nova referência R4. Por sua vez, a referência R3 continuará com o valor parágrafo X em seu atributo **última versão**.

Depois, o autor define a versão parágrafo X como obsoleta, e decide apagá-la. Este processo implica em apagar também todas as referências associadas a esta versão do parágrafo. Neste ponto, toma importância o uso do atributo **última versão**, pois o sistema, ao apagar o parágrafo X, também apagará todos aqueles objetos **referência interna** cujo atributo **última versão** seja igual a parágrafo X. No exemplo, ele pode apagar unicamente a referência R3, pois ela não forma parte de qualquer outra versão posterior do parágrafo em questão (a última versão na qual aparece é parágrafo X). Por outro lado, as referências R1, R2 não podem ser apagadas, pois ainda são válidas na nova versão parágrafo X'.

No exemplo apresentado, se considerou somente os objetos **referência interna**, mas a utilização do atributo **última versão** é igual nos objetos **referência externa** e **Comentário**.

O meta-modelo OHypA, com a adição dos objetos **Contexto versionado** e **Versão**, para a modelagem de versões em documentos estruturados multimídia, está esquematizado na figura 7.6.

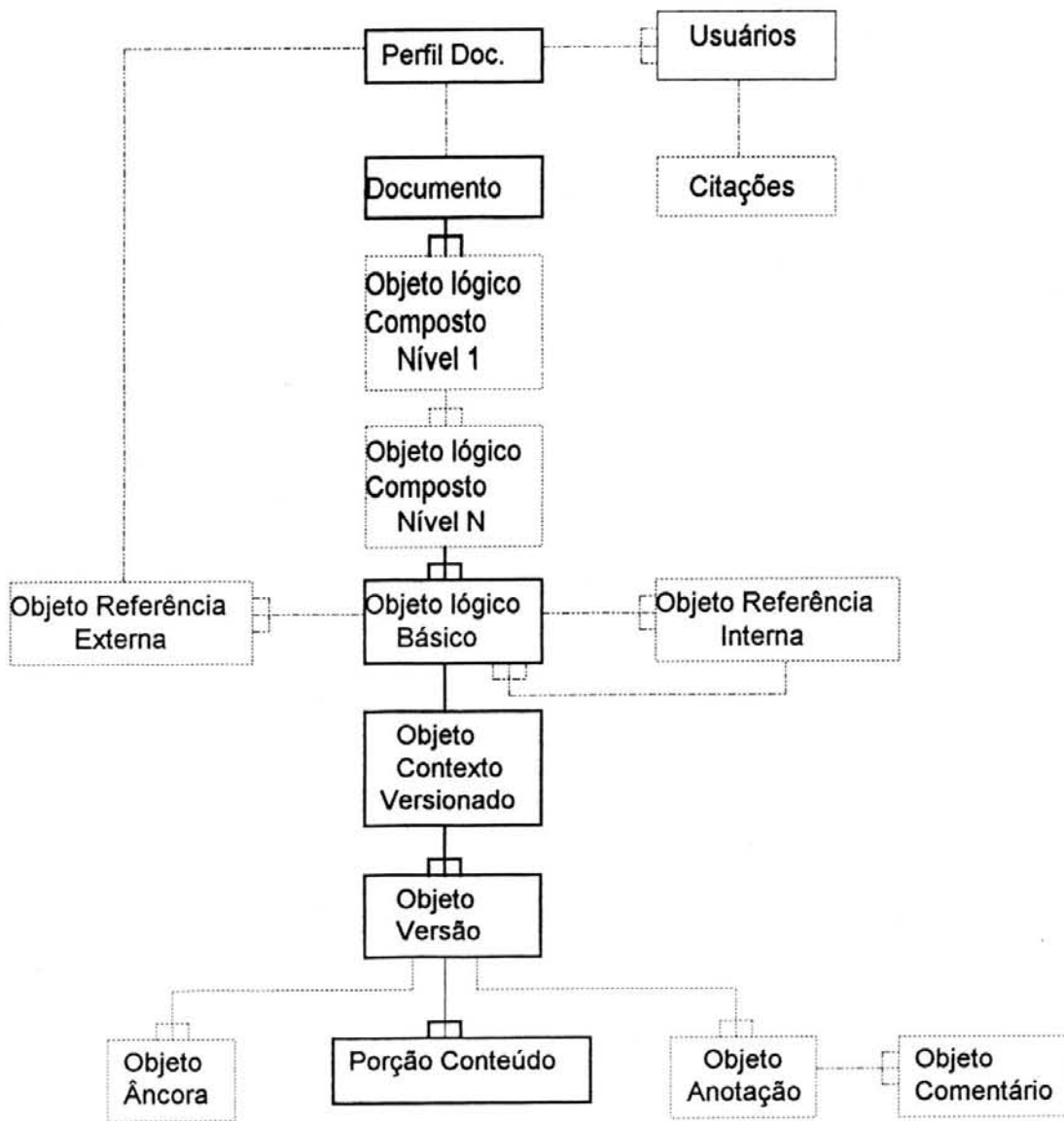


Figura 7.6: Meta-modelo OHypA, incluindo tratamento de versões.

7.3.3 Operações sobre versões.

As operações que envolvem o tratamento de versões em documentos estruturados multimídia são:

a- **Criação**: Esta operação abrange tanto a criação do objeto **contexto versionado** quanto a criação das versões propriamente ditas. O objeto **contexto versionado** é criado quando a primeira versão, correspondente ao objeto lógico básico, é criada.

A criação de uma nova versão tem as seguintes restrições:

- A versão da qual será derivada tem que estar em **estado permanente**.
- Esta criação pode ser realizada **unicamente** pelo autor do documento estruturado multimídia.
- O autor, explicitamente, expressa o estado da versão (temporária, enquanto está trabalhando nela; permanente, quando torna-se estável ou obsoleta, quando considera que pode ser apagada do sistema).

b- Modificação: Podem ser realizadas várias alterações sobre uma versão (no conteúdo, propriamente dito, ou nas relações hipertextuais associadas a ele), até que o usuário considere que a representação obtida é importante e deve ser registada como uma nova versão.

Estas alterações só podem ser feitas enquanto a versão é temporária. No caso de versões permanentes, as modificações gerarão uma nova versão.

c- Retirada de Versões: Significa apagar uma versão da história de versões do contexto versionado associado. Permite que versões obsoletas sejam eliminadas do histórico. Esta operação pode ser realizada sobre qualquer objeto versão que esteja em estado obsoleto, atualizando as relações de derivação e de correlação correspondentes na estrutura.

7.3.4 Operações/consultas sobre a estrutura de versões.

O modelo de tratamento de versões considerado em OHypA, além de suportar operações sobre as versões, como foi apresentado na subseção anterior, deve também possibilitar a realização de operações e/ou consultas sobre a estrutura formada pelo

conjunto de versões associadas a um objeto lógico básico em particular. Estas operações serão possíveis pela navegação em tal estrutura, através das relações de derivação e de correlação.

Algumas destas operações são:

a- Operações de busca, nas quais procura-se informação sobre uma versão específica (segundo data de criação, número de versão, etc.).

b- Operações sobre versões relativas a uma outra versão específica (sucessora, predecessora, correlata, etc.).

c- Operações sobre o objeto **contexto versionado**, para obter, por exemplo, a primeira versão, a versão atual, o número de versões, etc.

Todas as operações consideradas, tanto nesta subseção quanto na anterior, são perfeitamente suportadas pelo meta-modelo OHypA definido.

7.3.5. Atualização automática de relações hipertextuais.

Quando o autor de um documento estruturado multimídia decide criar uma nova versão de um objeto lógico básico, surge a seguinte pergunta:

O que acontece com as relações hipertextuais (referências internas, referências externas e anotações) definidas sobre a versão da qual será derivada a nova? Tais relações podem ter sido criadas tanto pelo autor do documento quanto por usuários leitores, segundo o conceito de visões tratado em este mesmo capítulo.

A proposta apresentada aqui, que responde a esta situação, pode ser dividida em duas partes, uma que considera as relações hipertextuais definidas pelo próprio autor do documentos (visão do autor), e outra que considera as relações hipertextuais definidas pelos diferentes usuários leitores do documento (e que formam as diferentes visões de usuário sobre o documento).

No primeiro caso, como a nova versão só pode ser criada pelo autor, a visão definida por ele (representada pelos objetos **âncora** e **anotação** associados à versão) é atualizada automaticamente no momento da criação da nova versão, da seguinte forma:

1- Se o botão (referência) continua igual, ou seja, não sofreu modificações, então:

-será modificado o valor do atributo **última versão** do objeto **referência interna**, **referência externa** ou **comentário** correspondente a tal botão, que passará a ter como valor o identificador da versão criada.

2- Se um botão foi apagado (referência), então será necessário:

-apagar a respectiva entrada no objeto **âncora** ou **anotação**, pois esta relação não existe mais nesta versão. Mas **não** será destruído o objeto **referência interna**, **referência externa** ou **comentário** associado a tal botão, pois ele continuará representando a relação hipertextual na versão anterior.

3- Se o valor do botão (trecho de texto) foi modificado, então:

-será modificado o atributo **valor âncora** do objeto **âncora**, se o botão representa uma referência interna ou externa, ou o atributo **valor anotação** do objeto **anotação**, se o botão representa um comentário.

-também se atualizará o valor do atributo **última versão**, no objeto **referência interna**, **referência externa** ou **comentário** associado a tal botão.

4- Se o destino da relação hipertextual foi modificado (mudou a referência interna ou a referência externa), será considerada como uma nova relação, o que implicará em:

- apagar no objeto **âncora** o correspondente **valor de âncora**.
- criar um novo objeto **referência interna** ou **referência externa**
- criar, também, a respectiva entrada no objeto **âncora**.

5- Se uma nova relação foi criada , implicará em:

- criar um novo objeto **referência interna** ou **referência externa**, segundo o tipo de relação criada.
- criar a respectiva entrada no objeto **âncora**.

No segundo caso, ou seja, com relação às visões definidas pelos usuários leitores sobre a versão, o mecanismo de tratamento de versões pode fazer duas coisas:

I- A nova versão só ficará com a visão atualizada do autor (como já foi considerado), enquanto que as outras visões ficarão vigentes na versão anterior. Neste caso, não existe uma atualização automática. Tal atualização de referências poderá ser realizada explicitamente a pedido, e sob controle do usuário proprietário da visão.

II- A nova versão atualizará as âncoras e anotações das visões de cada usuário, levando em conta o conteúdo da nova versão. Esta atualização consistirá em:

1- Se o botão (referência) continua igual na visão do usuário:

- só será modificado o atributo **última versão** do objeto **referência interna**, **referência externa** ou **comentário** associado.

2- Se foi eliminado um botão (referência) da visão do usuário:

- será apagada a respectiva entrada no objeto **âncora** ou **anotação** associado à visão do usuário, mas **não** será destruído o objeto **referência interna**, **referência externa** ou **comentário** associado a tal botão, pois ele continuará representando a relação hipertextual na versão anterior.

3- Se o valor do botão (trecho de texto) foi modificado :

-será modificado o atributo **valor âncora** do objeto **âncora**, se o botão representa uma referência interna ou externa, ou o atributo **valor anotação** do objeto **anotação**, se o botão representa um comentário, os quais pertencem à visão do usuário.

-também será atualizado o valor do atributo **última versão**, no objeto **referência interna**, **referência externa** ou **comentário** associado a tal botão.

Cabe mencionar que esta segunda opção pode gerar inconsistências entre versões-visões, portanto a mesma tem que ser objeto de mais profundos estudos e pesquisas. Sendo assim, o meta-modelo OHypA considerado suporta ambas políticas de atualização automática de relações hipertextuais.

Para fechar as idéias e conceitos considerados, e complementando a figura 7.6, na qual foi apresentada a modelagem e tratamento de versões segundo o meta-modelo OHypA, a seguir, na figura 7.7 é apresentado um exemplo da utilização deste modelo para o tratamento de versões de componentes lógicos em documentos estruturados multimídia.

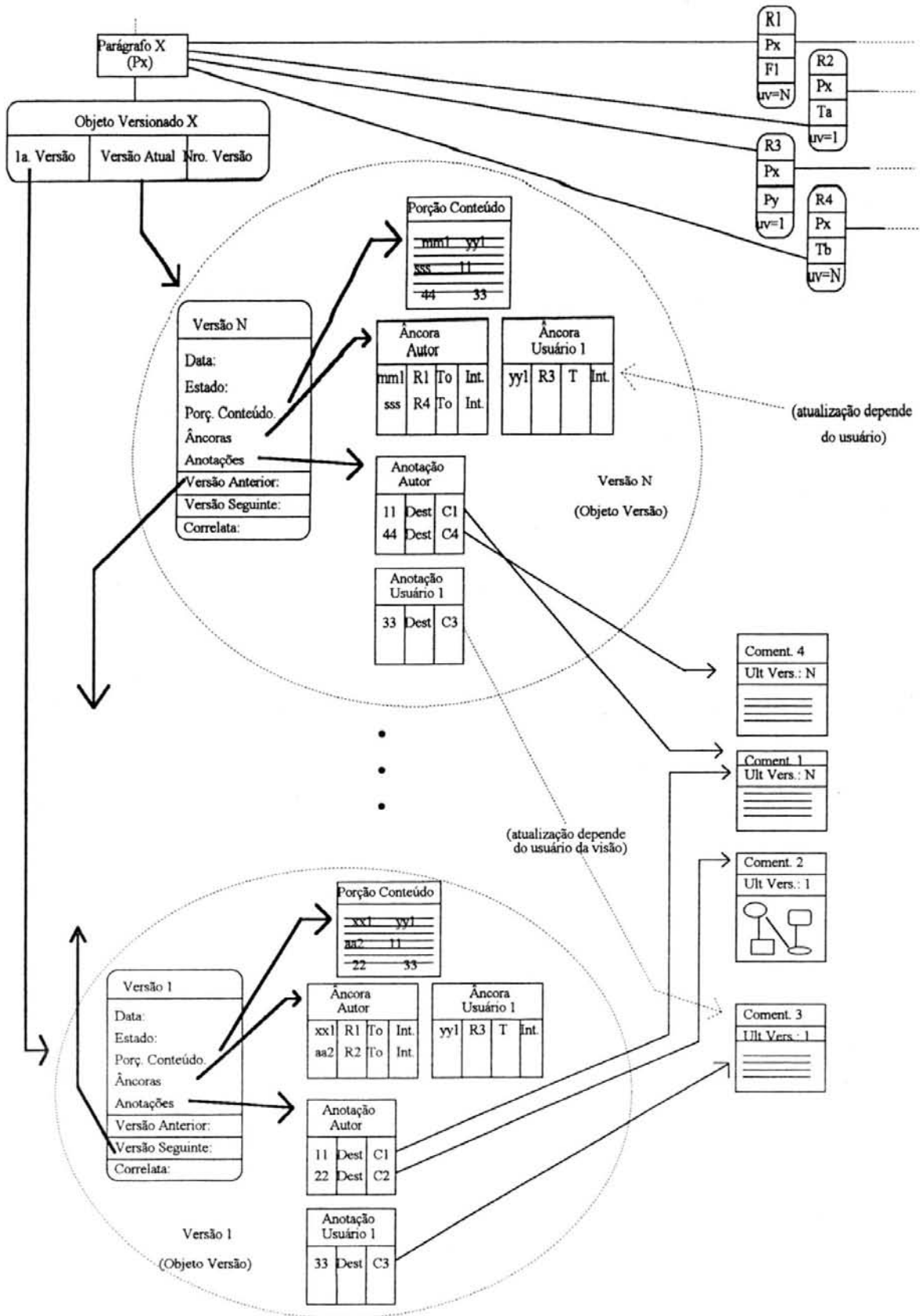


Figura 7.7 : Tratamento de Versões segundo o meta-modelo OHypA.

No exemplo da figura 7.7, na primeira versão (versão 1) do parágrafo X estão definidas duas visões sobre ela, uma que pertence ao próprio autor, e uma outra definida pelo usuário leitor 1.

A visão do autor está composta de:

a- Uma referência interna à figura 1, a qual tem como âncora o valor **xx1**.

Esta referência está representada com o objeto **R1**, cujo atributo última versão tem o valor **Versão 1**.

b- Uma outra referência interna à tabela A, que tem como âncora o valor **aa2**.

O objeto **R2** representa esta referência, e seu atributo última versão também tem o valor **Versão 1**.

c- Duas anotações auxiliares, cujas âncoras são **11** e **22**, respectivamente.

Tais anotações estão representadas pelos objetos **Coment1** e **Coment2**, cujos atributos última versão possuem o valor **Versão 1**.

Por sua vez, a visão do usuário 1 está composta de:

a- Uma referência interna ao parágrafo Y, cuja âncora associada possui o valor **yy1**. Esta referência está representada pelo objeto **R3**, com seu atributo última versão com valor **Versão 1**.

b- Uma anotação auxiliar, cuja âncora é **33**, representada pelo objeto **Coment3**, com valor **Versão 1** em seu atributo última versão.

Após um certo tempo, o autor do parágrafo decide criar uma nova versão do mesmo. Nesta nova versão (denominada **versão N**, para generalizar a história de versões), o autor realiza as seguintes modificações em sua visão:

a- Modifica o valor da âncora da referência **R1**, a qual passa de **xx1** para **mm1**. Como consequência, o sistema atualiza automaticamente o valor do atributo última versão do objeto **R1**, passando a ser agora **versão N**.

b- Elimina a referência à tabela A, ou seja a relação **R2**. Neste caso, o sistema simplesmente apaga a âncora **aa2** do objeto **Âncora**, mas continua mantendo o objeto **R2**, pois ele é significativo na versão 1.

c- Cria uma nova referência interna à tabela B cujo valor de âncora é **sss**. Em consequência, o sistema adiciona o valor de âncora ao objeto âncora da visão do autor, e cria um novo objeto **R4**, com valor **Versão N** no atributo última versão.

d- Mantém o comentário **C1**, e o sistema simplesmente atualiza o atributo última versão do objeto **Coment1**, atribuindo o valor **Versão N**.

e- Elimina o comentário **C2**. A atualização automática deve apagar do objeto anotação a âncora **22**, mas deve manter o objeto **Coment2**.

d- Por último, o autor cria uma nova anotação **C4**, com valor de âncora **44** e objeto comentário **Coment 4**.

No que se refere à visão do usuário 1 definida sobre a versão 1, e segundo o citado em subseções anteriores, poderá ser tanto atualizada automaticamente (o que pode gerar inconsistências) quanto pode ser atualizada sob controle do usuário proprietário da

visão (ou seja, a nova versão só terá atualizada a visão do autor, ficando a visão do usuário 1 ligada à versão 1).

7.3.6 Gerência de Configurações.

Segundo o meta-modelo OHypA, um documento estruturado multimídia está formado por uma hierarquia de objetos lógicos, sendo os níveis intermédios formados por objetos compostos, para chegar finalmente aos objetos lógicos básicos.

Pelo apresentado até aqui, as versões consideradas correspondem, na realidade, a estes objetos lógicos básicos, pois cada um deles terá associado um objeto contexto versionado.

Neste ponto surge a seguinte pergunta, o que acontece com os objetos compostos, como por exemplo, uma seção, que esta formada por três objetos básicos do tipo parágrafo, cada um dos quais com várias versões?

Para solucionar este problema utiliza-se o conceito de **Configuração**, o qual é explicado a seguir. Basicamente **configuração** é um conjunto de versões consideradas semanticamente válidas . No caso tratado aqui, a configuração de um objeto composto será um conjunto que inclui uma versão de cada um de seus objetos componentes. Ou seja, no caso tratado anteriormente, no qual um objeto composto do tipo seção tem como componentes três objetos básicos do tipo parágrafo, a configuração do objeto seção estará formada por uma versão de cada objeto parágrafo.

Assim como os objetos básicos podem ter associadas várias versões, formando a história de versões segundo a relação de derivação, os objetos compostos

também podem ter associadas várias configurações válidas, de acordo com aquelas versões, as quais também formarão um história de configurações válidas do objeto composto.

No âmbito do meta-modelo OHypA, define-se como configuração válida de um objeto composto o conjunto que inclui as versões atuais de cada objeto componente do mesmo. Ou seja, quando a versão de um componente deixa de ser válida (é criada uma outra nova), a configuração de todos os objetos compostos que fazem referência a ele também tem que ser atualizada. Tal atualização é denominada Propagação de Versões, e será considerada na próxima seção.

Outra configuração que inclua qualquer combinação de versões dos objetos componentes, poderá ser definida pelo autor do documentos, ficando sob sua responsabilidade a validade da mesma.

Para permitir a representação de configurações dos diferentes objetos compostos de um documento estruturado multimídia, no meta-modelo OHypA será associado, a cada objeto lógico composto, um conjunto de n-uplas. Cada uma destas n-uplas representará uma configuração válida, isto é, os identificadores das versões atuais dos objetos componentes. Por sua vez, as relações entre estas n-uplas representará a evolução das configurações do respectivo objeto. Na figura 7.8 é apresentado um exemplo que ilustra o conceito de configurações apresentado nesta subseção.

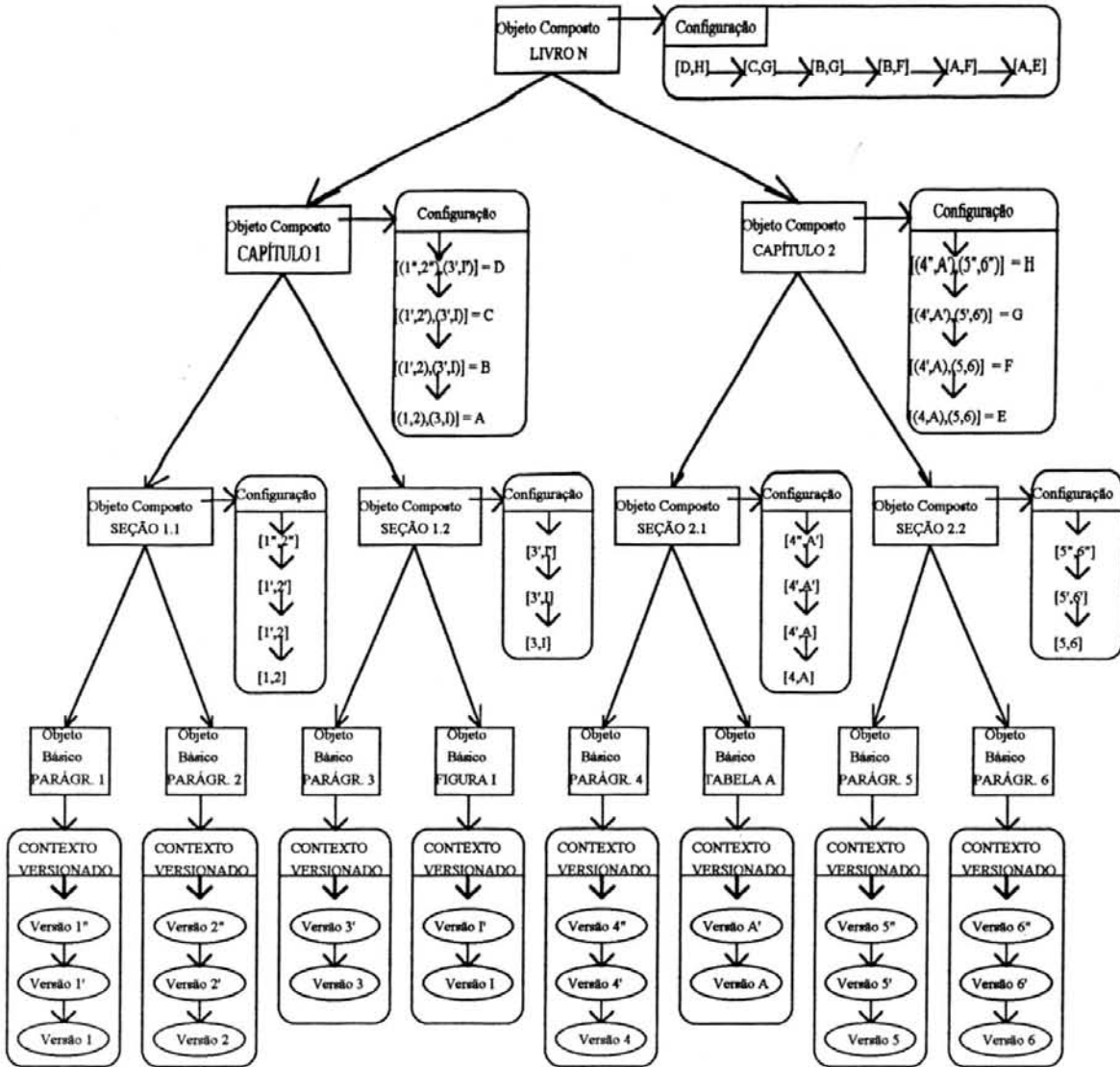


Figura 7.8: Configurações de Objetos compostos em OHypA.

7.3.7 Propagação de modificações.

Como foi considerado na subseção anterior, quando existem objetos compostos que possuem como componentes objetos que apresentam versões, as modificações nos componentes devem ser notificadas aos objetos que o utilizam. Estas modificações podem ser remoção e/ou criação de versões.

Segundo [GOL93a, GOL93b], as ações a serem tomadas frente a tais modificações nos componentes são:

1- Notificação: Aqui, podem ser utilizadas duas técnicas, uma **baseada em mensagens**, na qual o sistema envia mensagens para notificar os usuários das versões potencialmente afetadas. Na outra técnica, **baseada em sinais**, o sistema atualiza estruturas de dados, de modo que o usuário só terá conhecimento de uma mudança quando explicitamente fizer acesso a uma versão afetada.

Um problema a resolver é o escopo da notificação, ou seja, se são notificadas as versões que fazem referência direta ou indireta à versão modificada. Neste trabalho, não será considerada a notificação de modificações, realizando-se diretamente a propagação das mesmas, como será tratado a seguir.

2- Propagação: Apesar de útil em alguns casos, a propagação ilimitada de versões implica em uma proliferação de objetos muitas vezes indesejada. No modelo OHypA, a propagação é realizada ao longo da hierarquia dos componentes que incluem ao objeto lógico básico que sofreu a modificação, ou seja, o caminho utilizado para alcançar tal objeto lógico básico.

Também se estabelece como regra que a criação de uma nova versão não é propagada enquanto ela esteja no estado **temporário**. Desta forma, mantém-se compatibilidade com o tratamento de estados, definido em subseções anteriores.

Neste ponto é interessante observar que a solução descrita acima, onde a propagação de versões depende do estado das mesmas, oferece a noção de versão de documento e de versão de objetos lógicos compostos (ao invés de apenas versões de objetos lógicos básicos).

7.4 O meta-modelo final.

No decorrer, tanto do capítulo anterior quanto do presente, foram consideradas diferentes situações e/ou características referentes à modelagem de Documentos Estruturados Multimídia. Para cada uma delas, foi proposta uma solução, que gradativamente, foram conformando o meta-modelo OHypA.

Tal meta-modelo resulta, então, numa proposta de modelagem de Documentos Estruturados Multimídia, integrando conceitos dos sistemas de hipertextos e do padrão de documentos ODA/ODIF. Como resultado final, chegou-se ao meta-modelo OHypA apresentado na figura 7.9.

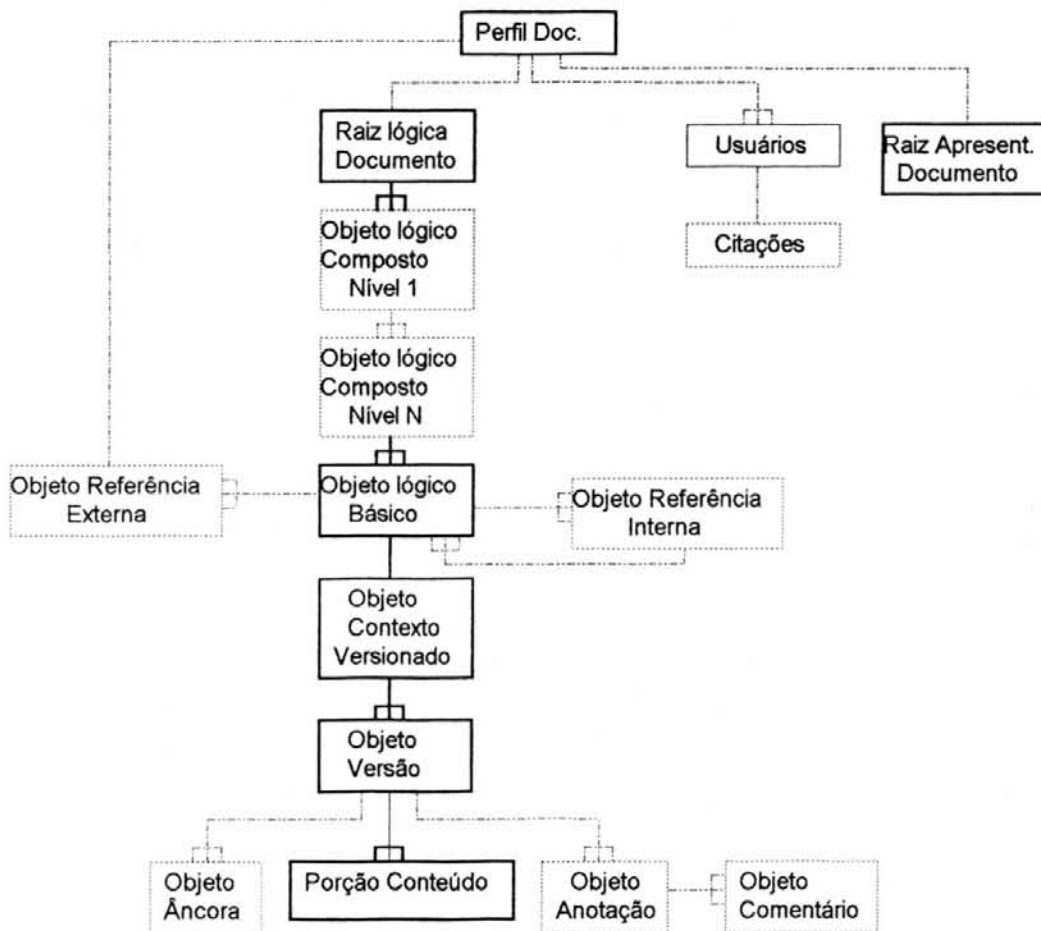


Figura 7.9: O Meta-modelo OHypA Final.

Lembrando o já citado no capítulo 6, o meta-modelo OHypA apresentado auxilia na organização das porções de informação de um hiperdocumento multimídia. É possível que o projeto de um hiperdocumento específico corresponda à organização estabelecida em tal meta-modelo, mas também permite-se que novas relações sejam acrescentadas em tal projeto.

Ou seja, o meta-modelo não pretende definir regras para a estruturação dos hiperdocumentos, mas fornece um esquema para que o criador (autor) faça isso. Nele, podem ser criados objetos, classes de objetos, estruturas e relações entre eles, e uma vez criados, os mesmos podem ser gerenciados e transmitidos de uma aplicação originadora para uma aplicação receptora.

Todas estas classes e objetos definidos pelos usuários serão, na realidade, subclasses daquelas definidas no meta-modelo OHypA (ver figura 7.10).

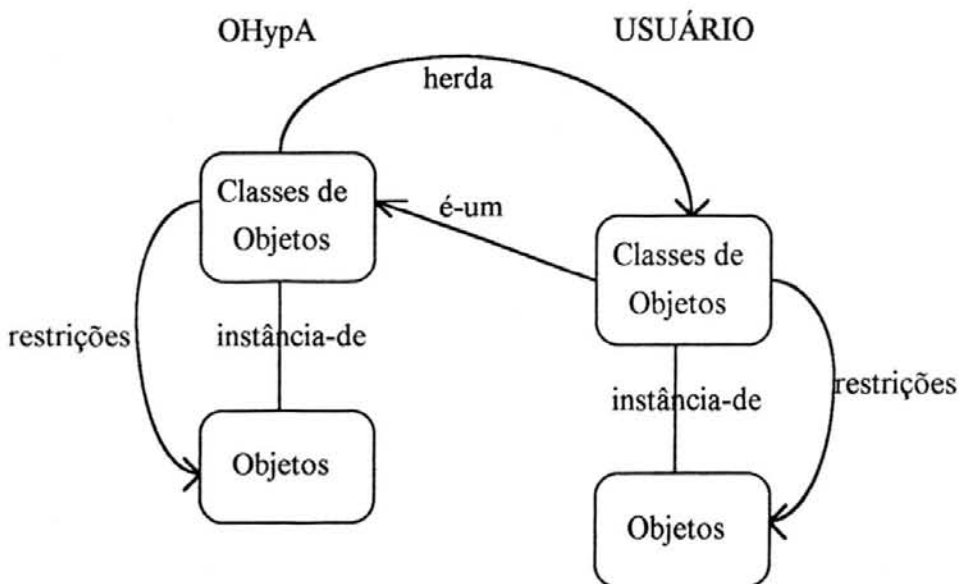


Figura 7.10: Classes de Objetos definidas em OHypA e pelos usuários.

Desta maneira, conforme foi visto no capítulo 6, o nível do modelo de hiperdocumento na arquitetura dos sistemas de Gerência de Documentos Estruturados Multimídia (ver figura 6.2) está subdividido em três níveis:

-Nível Abstrato: Classes, objetos e relações definidas no meta-modelo OHypA propriamente dito.

-Nível Concreto/Genérico: Classes e objetos definidos pelos usuários, segundo uma aplicação em particular.

-Nível Específico: Instâncias de hiperdocumentos específicos.

Esta relação entre o meta-modelo OHypA, apresentado na figura 7.9, e as classes e objetos definidos pelos usuários, segundo foi considerado na figura 7.10, é esquematizada a seguir na figura 7.11. Nela, utilizou-se uma outra representação, segundo a metodologia de Rumbaugh [RUM91].

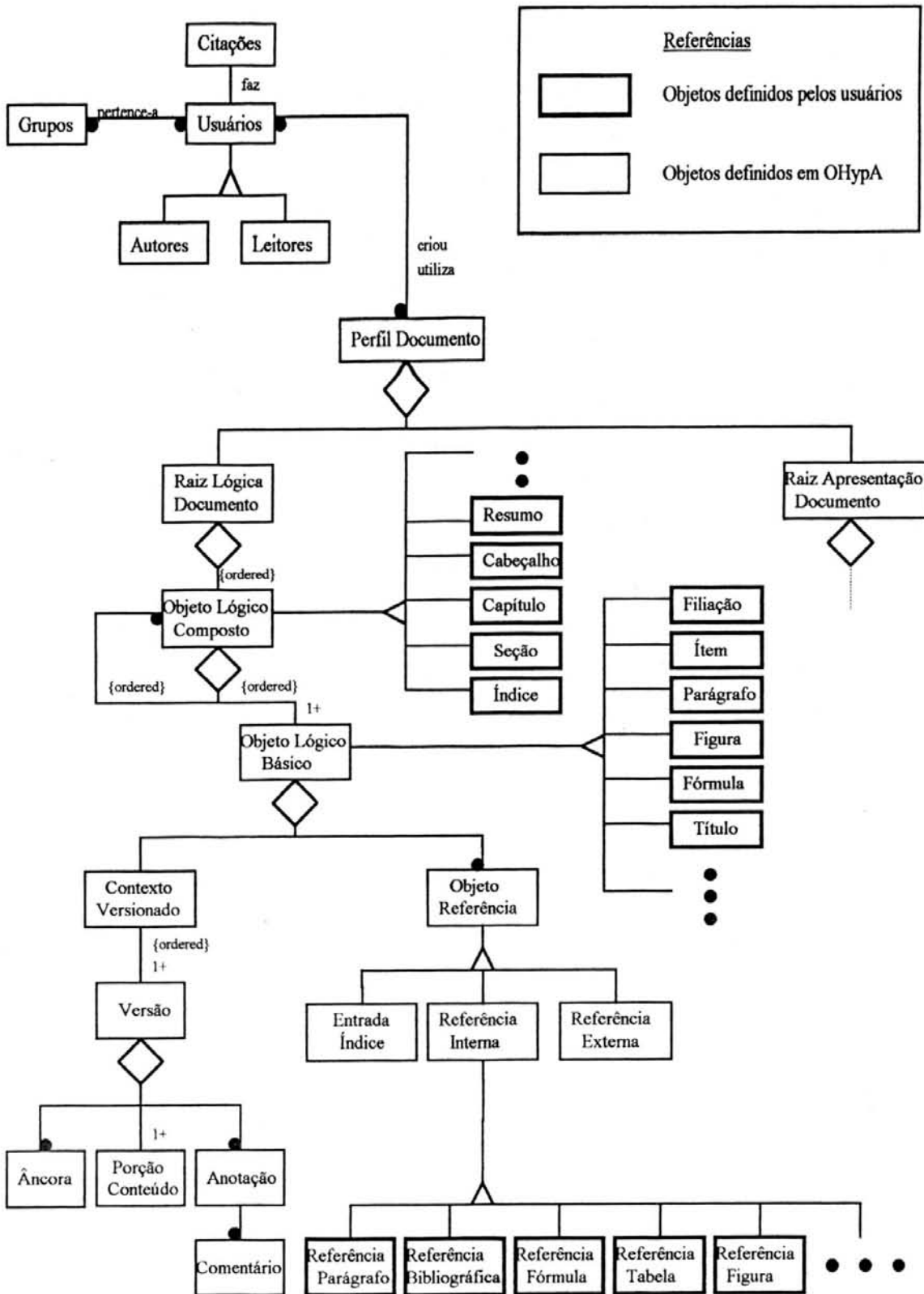


Figura 7.11: meta-modelo OHypA e objetos definidos pelos usuários segundo a metodologia de Rumbaugh.

Para concluir, no anexo A-1 desta dissertação são detalhados os objetos definidos em OHypA, identificando seus atributos e operações.

7.5 Abstrações do meta-modelo.

Como foi visto no capítulo 3, a maioria dos modelos de hipertextos existentes emprega o paradigma básico de nós, âncoras e elos. Cada um destes modelos também define, ao menos, uma abstração de mais alto nível.

O objetivo nesta seção é identificar tanto as abstrações de baixo nível quanto as abstrações de alto nível definidas no meta-modelo OHypA. Para tal fim será estabelecido um paralelo entre o paradigma básico de nós, elos e âncoras, utilizado pelos sistemas de hipertextos (o qual foi ponto de partida deste trabalho), e os objetos definidos em OHypA.

7.5.1 Abstrações de Baixo Nível.

Os componentes ou objetos básicos identificados são:

- <u>No paradigma básico</u>	- <u>Em OHypA</u>
- Nós	- Objeto lógico básico - Objeto Porção de Conteúdo - Objeto Comentário.
- Elos	- Objeto Referência (Interna, Externa)
- Âncoras	- Objeto Âncora - Objeto Anotação

7.5.2 Abstrações de Alto Nível.

Por sua vez, os objetos compostos ou esquemas identificados são:

- No paradigma básico

- Entidades ou Objetos

Compostos.

- Esquemas.

- Em OHypA

- Objeto Lógico Composto

- Objeto Contexto Versionado

- Objeto Versão

- Objeto Raiz Lógica Documento

7.5.3 Tipos de Ligações (Elos).

Um das abstrações mais importantes na modelagem de Documentos Estruturados Multimídia são os elos (ligações) existentes entre os diferentes componentes. Tais abstrações dão o sentido "hipertextual", definindo a estrutura do documento e possibilitando a navegação do usuário pelo espaço de informações.

Em OHypA, podem ser identificados diferentes tipos de ligações, dependendo da ação que executam e das informações que relacionam. Tais tipos estão apresentados na figura 7.12.

Tipos de Ligações (Elos).

Estruturais	→	- Referência a Objetos Subordinados. - Referência a Porções de Conteúdo.
Lógicas/Relacionais	→	- Referências Internas. - Referências Externas.
de Versões	→	- Relação de Derivação. - Relação de Correlação.
Ampliativas	→	- Anotações.
de Visões	→	- Entre diferentes visões, segundo o nível de visibilidade associado a elas (grupala, pública).

Figura 7.12: Tipos de ligações definidas em OHypA.

8 DISCUSSÃO E CONCLUSÕES.

8.1 Discussão.

Nos capítulos anteriores da presente dissertação, foi considerada a Gerência de Documentos Estruturados Multimídia, identificando-se as características mais importantes de tal aplicação. Focalizou-se como problema principal a modelagem de tais documentos estruturados multimídia. A estrutura lógica deles foi apontada como o elemento fundamental para auxiliar na gerência e busca de informações nele contidas.

Dentre as possíveis alternativas para tal modelagem, estudou-se tanto o padrão de documentos eletrônicos ODA/ODIF, como os diferentes modelos de hipertextos propostos na literatura. Como resultado deste estudo, chegou-se à conclusão que ambas alternativas não oferecem uma solução completa que consiga abranger todas as possibilidades identificadas no tratamento de documentos estruturados multimídia.

A partir desta conclusão e dos aspectos apontados anteriormente, foi proposto um meta-modelo de documentos estruturados multimídia (OHypA), resultado da integração do padrão ODA/ODIF e do paradigma básico de "nós-ligações" proposto pelos sistemas de hipertextos. Desta forma, o referido meta-modelo pode ser considerado tanto uma extensão ao padrão ODA/ODIF quanto uma proposta de modelo de hiperdocumentos.

Para a definição do meta-modelo OHypA foi utilizado o paradigma de orientação a objetos (POO), com o objetivo de usufruir das vantagens oferecidas por ele. Além disto, a área de orientação a objetos está em franco desenvolvimento, com inúmeras propostas de metodologias e notações, sendo por isto de grande interesse para a criação de ambientes experimentais.

As características mais importantes que identificam o meta-modelo OHypA são:

- Não define regras para a estruturação dos hiperdocumentos, mas fornece um esquema para que o autor o faça. Ou seja, não são prescritas nem classes de objetos lógicos nem estruturas de documentos particulares. Oferece um esquema, dentro do qual se pode criar objetos, classes e estruturas.

- É um mecanismo de modelagem que permite a descrição, de maneira simples e concisa, de hiperdocumentos a serem ainda projetados.

- Permite o projeto conceitual das aplicações, sem ter que preocupar-se com detalhes de implementação.

- Permite um processo de autoria estruturado, separando as questões de autoria e permitindo o seu tratamento individualizado. Nele, considera-se, fundamentalmente, a estrutura lógica dos documentos, mas sem perder de vista a ortogonalidade com os outros componentes do sistema. Estas considerações de ortogonalidade abrangem:

- a- Apresentação:** as relações entre os objetos lógicos e os objetos de apresentação, ou seja, entre as estruturas lógicas e de apresentação de um documento estão definidas pelos atributos **Diretivas de Disposição**, originais do padrão ODA/ODIF, as quais são mantidas no meta-modelo OHypA.

Tem que ser considerado que a apresentação dos documentos em um meio físico (tela, papel) depende da estrutura de apresentação e dos atributos de disposição embutidos na arquitetura do conteúdo associado.

b- Representação física: o projeto específico dos formatos de armazenamento está fora do alcance do meta-modelo. Porém, por ter sido ele projetado segundo o paradigma de orientação a objetos, sua integração ao modelo de um banco de dados orientado a objetos é muito simples.

c- Conteúdo: A divisão estrutura-conteúdo feita no padrão ODA/ODIF também foi preservada. O meta-modelo é totalmente independente das arquiteturas de conteúdos, podendo ser enriquecido cada vez que uma nova arquitetura é definida (documentos multimídia).

d- Semântica de Navegação: ela é especificada pelo código que implementa o sistema de Gerência de Documentos Estruturados Multimídia. Define a maneira de ser percorrida e apresentada a informação contida no documento. O meta-modelo, neste caso, oferece a base que permite tal navegação, através de: índices, perfis, ligações de composição, ligações lógicas, relações entre versões, etc. Em suma, ele possibilita:

I- Navegar pela hiperestrutura do documento.

II- Navegar através das diferentes visões de usuários, de acordo as permissões associadas.

III- Navegar pelo histórico de versões.

IV- Definir ferramentas de auxílio à navegação, tais como:

- **Estáticas:** mapas, diagramas da estrutura, etc.

- **Dinâmicas:** caminhos percorridos, *backtracking*.

- **Marcas:** possibilidade do usuário ou o sistema marcar componentes e/ou relações para poder voltar a eles.

8.2 Contribuições do presente trabalho.

Um resumo das principais contribuições do trabalho é apresentado a seguir:

- O estudo das características e propriedades da Gerência de Documentos Estruturados Multimídia, dando principal atenção ao modelo subjacente.

- O estudo do padrão de documentos eletrônicos ODA/ODIF desde o ponto de vista dos sistemas de Gerência de Documentos, identificando tanto as características desejáveis (aspectos positivos) quanto as lacunas ou omissões (aspectos negativos) para o tratamento do problema em questão.

- O estudo dos diferentes modelos de hipertextos propostos na literatura, considerando importante o paradigma básico de "nós-ligações" , mas não suficiente para o presente trabalho.

- A integração das estruturas hierárquicas de composição dos documentos (segundo ODA/ODIF) com as estruturas em rede dos hipertextos, resultando na proposta do meta-modelo de Documentos Estruturados Multimídia OHypA.

- A inclusão, dentro do meta-modelo OHypA, de um modelo de tratamento de versões, o qual permite:

- Exploração de alternativas.
- Manutenção do histórico dos componentes e do documento.
- Atualização automática de referências.

- Gerência de representações de uma mesma informação.

- A definição do conceito de visões de documentos estruturados multimídia, através da qual realiza-se o controle tanto do acesso aos documentos quanto da autoria dos mesmos. Este controle está baseado na definição de classes de usuários e níveis de visibilidade, o que proporciona restrições à execução de operações sobre as visões geradas em função da classe de usuário.

- A possibilidade de adicionar anotações, comentários, referências, etc. em um documento de outro autor sugere a noção de trabalho cooperativo, sendo esta uma área em franco desenvolvimento.

- A integração de dois enfoques em pleno desenvolvimento, obtendo um modelo orientado a objetos que é facilmente integrado a um Sistema de Banco de Dados Orientado a Objetos.

8.3 Trabalhos futuros.

Este trabalho é o ponto de partida para a implementação efetiva de um Sistema de Gerência de Documentos Estruturados Multimídia, que é o objetivo do projeto DO²CEM. Futuros trabalhos dentro do mesmo projeto, poderão continuar nesta linha, partindo dos resultados aqui obtidos e considerando em maior profundidade:

- 1- A relação existente entre as estruturas lógicas (meta-modelo OHypA) e as estruturas de apresentação dos documentos multimídia, através das **Diretivas de Disposição**, tomadas de ODA/ODIF e preservadas no meta-modelo (problema de **sincronismo** da apresentação).

2- O modelo de versões proposto, considerando:

- adicionar a noção de versões de estruturas.
- melhorar o tratamento de configurações dos documentos.
- otimizar a atualização automática das relações hipertextuais definidas nas visões dos usuários leitores, quando são criadas novas versões.

3- A semântica de navegação que será suportada pelo meta-modelo, ou seja, como será percorrida e apresentada a informação contida nos documentos modelados.

4- O processo de definição e edição das estruturas de composição de novos tipos de documentos, o qual terá que ser guiado e controlado pelo sistema, de acordo ao esquema definido segundo o meta-modelo OHypA.

5- A integração do Sistema de Gerência de Documentos Estruturados Multimídia a um Banco de Dados Orientado a Objetos (segundo o meta-modelo OHypA), pois as características de volume de informação e complexidade dos documentos sugerem a utilização de um banco de dados.

ANEXO A-1 CLASSES DE OBJETOS DO META-MODELO OHypA.

Perfil do Documento	
Título: ALFANUMÉRICO	
Data: INTEIRO	
Autor: USUÁRIO	
NívelSeguridade: (PUB,PRIV,GRUP)	
Leitores: LISTA[USUÁRIOS]	
RefEstDisposiç:RAIZ APRESENTAÇ	
RefEstLógica: RAIZ LOGICA DOC.	
DiretDisposição: DISPOSIÇÃO	
DáTítulo	DefineSeguridade
DefineTítulo	DáLeitores
DáAutor	AdicionaLeitor
DefineAutor	EliminaLeitor
DáSeguridade	DáEstruturaLóg.

Usuário
Nome: ALFANUMÉRICO
IdentifAcadêmica: ALFANUMÉRICO
Grupo: GRUPOS
Visibilidade: (PUB,PRIV,GRUP)
Citação: CITAÇÕES
DáNome
DefNome
DáIdentifAcadêmica
MudaIdentifAcadêmica
DáGrupo
DáVisibilidade
DefVisibilidade

Citações
ListaCitações: LISTA[(REF EXT, DIREÇÃO)]
AdicionaCitação
DáCitação
ApagaCitação

Raiz Lógica Documento	
Denominação: ALFANUMÉRICO	
Componentes: LISTA [OBJ. LÓGICO]	
GeradorSubordinados: LISTA [GERADOR]	
Configuração: LISTA[CONJ(VERSÕES)]	
DiretivaDisposição: DISPOSIÇÃO	
DáDenominação	DáGerador
DefDenominação	EliminaGerador
DáComponentes	DáConfig
AdicionaComponente	ConfigAnterior
EliminaComponente	ConfigSeguinte
ModificaComponente	MudaConfig
DefGerador	

Objeto Lógico Composto	
Denominação: ALFANUMÉRICO	
Componentes: LISTA [OBJ. LÓGICO]	
GeradorSubordinados: LISTA [GERADOR]	
Configuração: LISTA[CONJ(VERSÕES)]	
DiretivaDisposição: DISPOSIÇÃO	
DáDenominação	DáGerador
DefDenominação	EliminaGerador
DáComponentes	DáConfig
AdicionaComponente	ConfigAnterior
EliminaComponente	ConfigSeguinte
ModificaComponente	MudaConfig
DefGerador	

Objeto Lógico Básico
Denominação: ALFANUMÉRICO
RefInterna: REFERÊNCIA INTERNA
RefExterna: REFERÊNCIA EXTERNA
Conteúdo: CONTEXTO VERSIONADO
DiretivaDisposição: DISPOSIÇÃO
DáNome
DefNome
DáContextoVersionado

Referência Interna	
ÚltimaVersão: VERSÃO	
Origens: LISTA[OBJLÓGBÁSICOS]	
Destinos:LISTA[OBJLÓGBÁSICOS]	
DiretivasDisposição: DISPOSIÇÃO	
CriaRefInterna	AdicionaOrigem
ApagaRefInterna	ApagaDestino
DáDestinos	ApagaOrigem
DáOrigens	DáVersão
LêReferência	ModifVersão
AdicionaDestino	

Referência Externa	
ÚltimaVersão: VERSÃO	
Origens: LISTA[OBJLÓGBÁSICOS]	
Destinos:LISTA[(PERFIL DOC, OBJLOGBÁS)]	
DiretivasDisposição: DISPOSIÇÃO	
CriaRefExterna	AdicionaOrigem
ApagaRefExterna	ApagaDestino
DáDestinos	ApagaOrigem
DáOrigens	DáVersão
LêReferência	ModifVersão
AdicionaDestino	

Contexto Versionado
VersãoAtual: VERSÃO PrimeiraVersão: VERSÃO NoVersões: INTEIRO
DáVersãoAtual DáPrimeiraVersão DáNoVersões DerivaVersão DefineCorrelata EliminaVersão BuscaVersão ModificaVersãoAtual

Versão																
Data: INTEIRO Estado: (TEMP, PERM, OBSOL) DiretivaDisposição: DISPOSIÇÃO Anterior: VERSÃO Seguinte: VERSÃO Correlata: VERSÃO Conteúdo: LISTA[PORÇÃO CONT] Âncora: LISTA [ÂNCORAS] Anotação : LISTA [ANOTAÇÕES]																
<table border="0"> <tr> <td>AdicionaRefInt</td> <td>EliminaAnot</td> </tr> <tr> <td>AdicionaRefExt</td> <td>VersãoAnterior</td> </tr> <tr> <td>ModifRefInt</td> <td>VersãoSeguinte</td> </tr> <tr> <td>ModifRefExt</td> <td>VersãoCorrelata</td> </tr> <tr> <td>EliminaRefInt</td> <td>EstadoVisão</td> </tr> <tr> <td>EliminaRefExt</td> <td>EstadoVersão</td> </tr> <tr> <td>AdicionaAnot</td> <td>DefineEstVersão</td> </tr> <tr> <td>ModifAnot</td> <td></td> </tr> </table>	AdicionaRefInt	EliminaAnot	AdicionaRefExt	VersãoAnterior	ModifRefInt	VersãoSeguinte	ModifRefExt	VersãoCorrelata	EliminaRefInt	EstadoVisão	EliminaRefExt	EstadoVersão	AdicionaAnot	DefineEstVersão	ModifAnot	
AdicionaRefInt	EliminaAnot															
AdicionaRefExt	VersãoAnterior															
ModifRefInt	VersãoSeguinte															
ModifRefExt	VersãoCorrelata															
EliminaRefInt	EstadoVisão															
EliminaRefExt	EstadoVersão															
AdicionaAnot	DefineEstVersão															
ModifAnot																

Âncoras
Autor: USUÁRIO Data: INTEIRO DiretivaDisposição: DISPOSIÇÃO ListaÂncoras: LISTA[(Desloc: ALFANUMÉRICO, Ref: REFERÊNCIA, Direção: (TO, FROM), Tipo: (INT, EXT))]
AdicionaÂncora ApagaÂncora VerÂncora LêÂncora

Anotações
Autor: USUÁRIO Data: INTEIRO DiretivaDisposição: DISPOSIÇÃO ListaÂncoras: LISTA[(Desloc: ALFANUMÉRICO, Destinat: USUÁRIO, Coment: COMENTÁRIO))]
CriaAnotação ApagaAnotação VerAnotação LêAnotação

Comentários
Última Versão: VERSÃO Conteúdo: ARQUIT.CONTEÚDO Diretiva Disposição: DISPOSIÇÃO
Dá Versão Modifica Versão Dá Tipo Arquitetura

Porção Conteúdo
Conteúdo: ARQUIT.CONTEÚDO Diretiva Disposição: DISPOSIÇÃO
Dá Tipo Arquitetura

Grupos										
Projeto: ALFANUMÉRICO Instituição: ALFANUMÉRICO Responsável: USUÁRIO Integrantes: LISTA [USUÁRIOS]										
<table> <tr> <td>Dá Projeto</td> <td>Def Responsável</td> </tr> <tr> <td>Dá Instituição</td> <td>Dá Integrantes</td> </tr> <tr> <td>Define Projeto</td> <td>Adiciona Integrantes</td> </tr> <tr> <td>Define Instituição</td> <td>Elimina Integrantes</td> </tr> <tr> <td>Dá Responsável</td> <td></td> </tr> </table>	Dá Projeto	Def Responsável	Dá Instituição	Dá Integrantes	Define Projeto	Adiciona Integrantes	Define Instituição	Elimina Integrantes	Dá Responsável	
Dá Projeto	Def Responsável									
Dá Instituição	Dá Integrantes									
Define Projeto	Adiciona Integrantes									
Define Instituição	Elimina Integrantes									
Dá Responsável										

BIBLIOGRAFIA

- [ATK89] ATKINSON, Malcolm, et al. The object oriented Database System Manifesto. In: INTERNATIONAL WORKSHOP ON OBJECT-ORIENTED DATABASE SYSTEM, August 19, 1989. **Proceedings ...** [S.l. : s.n.], 1989.
- [BAN87] BANERJEE, J.; CHOU, H.; GARZA, J. Data Model issues for object oriented applications. **ACM Transactions on Information Systems**, New York, v.5, n.1, 1987.
- [BOT92] BOTAFOGO, R.; RIULIN, E. Structural Analysis of Hypertexts: Identifying hierarchies and useful metrics. **ACM Transactions on Information Systems**, New York, v.10 n.2, p. 142-179, Apr. 1992.
- [BRO89] BROWN, H. Standards for Structured Documents. **The Computer Journal**, * New York, v.32, n.6, p. 505-513, 1989.
- [BRO87a] BROWN, P.J. **Report on an application of GUIDE in decision support.** [S.l. : s.n.], June 1987. Personal Communication
- [BRO87b] BROWN, P.J. **The differences between UNIX GUIDE and OWL GUIDE.** [S.l. : s.n.], June 1987. Personal Communication.
- [BUS45] BUSH, V. As we may think. **Atlantic Monthly**, 1945. Também publicado em: GREIF, I (Ed.) **Computer-Supported Cooperative Work: A book of readings.** San Mateo, CA: Morgan/Kaufmann, 1988.
- [BUS67] BUSH, V. **MEMEX Revisited.** Evolution of an Information Society, 1967.

[BUM93] BUSTAMANTE, Marco Eduardo M. **Uma Interface Interativa e Navegacional para Hiperdocumentos**. Porto Alegre, Brasil : CPGCC da UFRGS, Ago. 1993. 147 p. (Dissertação de Mestrado)

[CAM88] CAMPBELL, B.; GOODMAN, J.M. HAM: A General Purpose Hypertext Abstract Machine. **Communications of the ACM**, v.31, n.7, p. 856-861, July 1988.

[CAM91] CAMPBELL-GRANT, I. Introducing ODA. **Computer Standard and Interfaces**, v.11, p. 149-157, 1991. *

[CON87] CONKLIN, J. Hypertext: An introduction and Survey. **IEEE Computer**, New York, v.20, n.9, p. 17-41, Sept. 1987.

[COR89] CORAY, G.; LEMONE, K. The Use of inheritance in Document Specifications. In: WORKSHOP ON OBJECT-ORIENTED DOCUMENT MANIPULATION, (WOODMAN), 1989, Rennes, France. **Preprints ... Paris : AFCET**, 1989. 410 p. p. 165-169. *

[DAT87] DATE, C.J. **A Guide to the SQL Standard**. [S.l.]:Addinson-Wesley, 1987.

[DAT89] DATE, C.J. **Introdução a Sistemas de bancos de dados**. [S.l.] : Editora Campus, 1989.

[DEL87] DELISLE, N.; SCHWARTZ, M.D. Context- A Partitioning Concept for Hypertext. **ACM Transactions on Information Systems**, New York, v.5 n.2, p. 168-186, Apr. 1987.

- [FIS87] FISHMAN, D.; BEECH, D.; CATE, H. IRIS: An object oriented database management system. **ACM Transactions on Information Systems**, New York, v.5 n.1, 1987.
- [FUR82] FURUTA, R.; SCOFIELD, J.; SHAW, A. Document Formatting Systems: Survey, Concepts and Issues. **Computing Surveys**. New York, v.14, n.3, p. 417-472, Sept. 1982.
- [FUR89a] FURUTA, R. An object-based taxonomy for abstract structure in document models. **The Computer Journal**, New York, v.32, n.6, p. 494-504, 1989.
- [FUR89b] FURUTA, R.; STOTTS, D. Object Structures in Paper Documents and Hypertexts. In: **WORKSHOP ON OBJECT-ORIENTED DOCUMENT MANIPULATION**, (WOODMAN), 1989, Rennes, France. **Preprints ... Paris : AFCET**, 1989. 410 p. p. 147-151.
- [FUR90] FURUTA, R.; STOTTS, P. The Trellis Hypertext Reference Model. In: **HYPertext STANDARIZATION WORKSHOP**. Gaithersburg. **Proceedings...** Washington : NIST, 1990. p. 83-93.
- [GAR91] GARZOTTO, F.; PAOLINI, P.; SCHWABE, D. HDM- A Model for the Design of Hypertext Applications. In: **ACM CONFERENCE ON HYPertext**, 3. (HYPertext'91), Dec. 15-18, 1991, San Antonio, Texas. **Proceedings...** New York:ACM, 1991. 485 p. p. 313-327.
- [GOL93a] GOLENDZINER, L.G. **Um estudo sobre versões em bancos de dados orientados a objetos**. Porto Alegre: CPGCC da UFRGS, 1993. 75 p. (Relatório de Pesquisa, 213).

- [GOL93b] GOLENDZINER, L.G; DOS SANTOS, C. S. Versões em Bancos de Dados Orientados a Objetos. In: SIMPÓSIO BRASILEIRO DE BANCOS DE DADOS, 8., (SBBD'93), 1993, Campina Grande, Paraíba. **Anais ...** Campina Grande: SBBD, 1993. p. 7-23.
- [HAL88] HALASZ, F.G. Reflections on NoteCards: Seven Issues for the Next Generation of hypermedia Systems. **Communications of the ACM**, New York, v.31, n.7, p.836-852, July 1988.
- [HAL90] HALASZ, F.G.; SCHWARTZ, M. The Dexter Hypertext Reference Model. In: **HYPERTEXT STANDARIZATION WORKSHOP**. Gaithersburg. **Proceedings...** Washington : NIST, 1990. p. 95-133.
- [HOR85] HORAK, W. Office Document Architecture and Office Document Interchange *
Format: Current Status of International Standardization. **IEEE Computer**, Los Alamitos, CA, p. 50-60, Oct. 1985.
- [ISO86] ISO. **ISO 8879 Information Processing- Text and Office Systems-**: Standard Generalized Markup Language. [S.l.]: ISO, 1986.
- [ISO87] ISO. **ISO 8632 Information Processing Systems -Computer Graphics-**: Metafile for the storage and transfer of picture description information, Parts 1-3. [S.l.]: ISO, 1987
- [ISO89] ISO. **ISO 8613 Information Processing- Text and Office Systems-**: Office Document Architecture (ODA) and Interchange Format. [S.l.]: ISO, 1989.

- [LAN90] LANGE, D. A Formal Model of Hypertext. In: HYPERTEXT STANDARIZATION WORKSHOP. Gaitherburg. **Proceedings...** Washington : NIST, 1990. p. 145-166.
- [LIM88] LIMA, José Valdeni de. Um estudo sobre o tratamento da estrutura lógica de documentos. In: CONFERENCIA LATINO AMERICANA DE INFORMATICA, 14., Buenos Aires. **Proceedings ...** Buenos Aires: CLAI, 1988. p. 402-416.
- [LIM89a] LIMA, José Valdeni de. Traitement de la Structure logique de Documents Multimédia. In: CONVOCATÓRIA BIANUAL DA CONVENÇÃO INFORMÁTICA LATINA, 7., Barcelona. **Anais...** Barcelona: CIL, 1989. p. 580-598. Março 1989.
- [LIM89b] LIMA, José Valdeni de; GALY, Henri; LOPEZ Mauricio. Integração da Estrutura lógica de documentos em SGBD. In: CONFERENCIA INTERNACIONAL DE LA SOCIEDAD CHILENA DE CIENCIA DE LA COMPUTACION, 9.; CONFERENCIA LATINO AMERICANA DE INFORMATICA, 15., Santiago, Chile. **Proceedings...** Santiago: CLAI, p. 190-202, Julio 1989.
- [LIM89c] LIMA, José Valdeni de, GALY Henri. Un exemple d'Intégration des documents multimédia aux SGBD: l'Intégration du modèle ODA á um modèle fonctionnel. In: WORKSHOP ON OBJECT-ORIENTED DOCUMENT MANIPULATION, (WOODMAN), 1989, Rennes, France. **Preprints ...** Paris : AFCET, 1989. 410 p. p. 101-112.

[LIM90] LIMA, José Valdeni de; GALY Henri. The integration of structure documents into DBMS. In: ELECTRONIC PUBLISHING, (EP90), 1990, Gaithersburg, MD-USA. **Proceedings...** [S.l.:s.n], 1990.

[LIM91] LIMA, José Valdeni de; CANDURO, Flávio Vinicius. Da concepção à divulgação de documentos: Integração da microedição e bancos de dados. In: CONGRESSO NACIONAL DE INFORMÁTICA, 24., set. 1991, São Paulo. **Anais ...** São Paulo:SUCESU, 1991, p. 142-151.

[LIN90] LINDSAY, B.; STONEBRAKER, M. The object oriented counter manifesto. **The Computer Journal**, New York, v.31, n.2, 1990. *

[MAI84] MAIER, D.; COPELAND, G. Making Smalltalk a database system. **SIGMOD RECORD**, New York, v.14, n.2, p.316-325, July 1984.

[MAR90] MARTINS, J.F.T.; ALBUQUERQUE, E.; MEIRA, S. Considerações sobre o Gerenciamento de objetos em Sistemas de Hipertextos (comunicação). In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 4., SBES, 1990, São Paulo. **Anais** Sao Paulo: SBES, 1990.

[MAR91] MARTINS, J.F.T. **Um gerenciador de objetos para sistemas de Hipertextos.** Recife, PE: UFPE, 1991 (Dissertação de Mestrado).

[MUR89] MURATA, M. An Object-Oriented interpretation of ODA. In: WORKSHOP ON OBJECT-ORIENTED DOCUMENT MANIPULATION, (WOODMAN), 1989, Rennes, France. **Preprints ...** Paris : AFCET, 1989. 410 p. p. 91-100. *

- [MUS89] MUSCATE, H. ODA Document Editing in Office Systems. In: WORKSHOP ON OBJECT-ORIENTED DOCUMENT MANIPULATION, (WOODMAN), 1989, Rennes, France. **Preprints ...** Paris : AFCET, 1989. 410 p. p. 68-78.
- [NEM93] NEMETZ, Fabio. Modelos de Dados para Hipermedia. CPGCC-UFRGS, Porto Alegre, Brasil: CPGCC da UFRGS, 1993. 42 p. (Trabalho Individual, 313)
- [NIE90] NIELSEN, J. Hipertext and Hypermedia. [S.l.]:Academic Press, 1990.
- [ONT88] ONTOLOGIC, Cambridge Vbase for object Applications. 1988.
- [ORS85] ORSINI, R.; CARDELL, L.; ALBANO A. Galileo: An strongly typed, interactive conceptual language. **ACM Transactions on Database Systems**, New York, v.10, n.2, 1985.
- [PAR89] PARSAYE, K.; CHIGNELL, M.; KHOSHAFIAN, S.; WONG H. **Intelligent Database**. [S.l.]:John Wiley & Sons, 1989. 474 p.
- [PER92] PERIM, M. **Um sistema de Gerenciamento de Hiperdocumentos para Ambientes de Desenvolvimento de Software**. Porto Alegre, Brasil: CPGCC da UFRGS, 1992. 198 p. (Dissertação de Mestrado).
- [PLA91] PLAISANT, C. An Overview of Hyperties, its User Interface and Data Model. In: BROWN, Heather (Ed.). **Hypermedia/Hypertext and Object oriented Database**, [S.l.]:Chapman and Hall, 1991. p.17-31
- [QUI92] QUINT, V.; VATTON I. Combining Hypertext and Structured Documents in Grif. In: ACM CONFERENCE ON HYPERTEXT, 4., Nov. 1992, Milano, Italy. **Proceedings...** [S.l.]:ACM, 1992.

- [RAD91] RADA, R. **Hypertext: From Text to Expertext**. [S.l.]:McGraw-Hill,1991. 238p.
- [RHE85] RHEINGOLD, H. **Tools for Thought**. [S.l.]:Computer Book Division/Simmon & Schuster, 1985.
- [RUM91] RUMBAUGH, et al. **Object Oriented Modeling and Design**. New Jersey:Englewood Cliffs, 1991. 500 p.
- [SAL83] SALTON, G.; MCGILL, M.J. **An Introduction to Modern Information Retrieval**. New York:Mc Graw-Hill. 1983
- [SCH93] SCHWABE, Daniel; ROSSI, Gustavo. Introdução aos Sistemas e à Autoria Hiperídia. Caderno de Textos. In: ESCOLA BRASILEIRA ARGENTINA DE INFORMÁTICA, 6. (EBAI), jul. 1993. Córdoba, Argentina. 71 p.
- [SHE91] SHERMAN, M.; ROSENBERG J. The need for style systems in ODA. **Computer Standards and Interfaces**, v.11, p. 177-182, 1991.
- [VEL88] VELEZ, F.; LECLUSE, C.; RICHARD P. O2- An object oriented data model. In: SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, June 1-3, 1988, Chicago, Illinois. **Proceedings...** New York:ACM, 1988. 447 p.
- [WEI93] WEISSHEIMER, E. **Um sistema de Gerenciamento de Hiperdocumentos Integrando Editores de texto e Sistemas de Gerência de Bases de Dados**. Porto Alegre, Brasil: CPGCC da UFRGS, 1993. 160 p. (Dissertação de Mestrado). 681.52.063(043) w4295 *



Informática
UFRGS

A Modelagem de Documentos Estruturados Multimídia Integrando Sistemas de Hipertextos e ODA/ODIF.

Dissertação apresentada aos Senhores:




Prof. Dr. Glesio Saraiva dos Santos



Prof. Dr. José Mauro Volkmer de Castilho




Prof. Dr. José Valdeni de Lima




Prof. Dr. Marcos Roberto da Silva Borges (NCE/UFRJ)

Vista e permitida a impressão.
Porto Alegre, 20/01/99.



Prof. Dr. José Valdeni de Lima,
Orientador.



Prof. Dr. Ricardo A. da L. Reis,
Coordenador do Curso de Pós-Graduação
em Ciência da Computação.