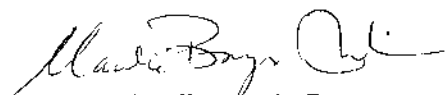


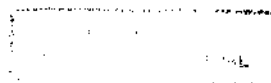
# Integração de Sistemas de Banco de Dados Heterogêneos em Aplicações de Planejamento Urbano

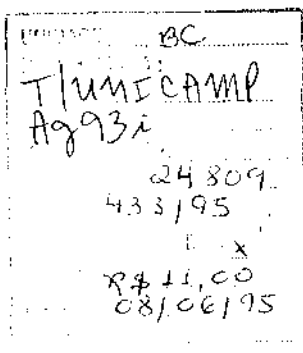
Este exemplar corresponde à redação final da  
tese devidamente corrigida e defendida pela  
Sra. Cristina Dutra de Aguiar e aprovada  
pela Comissão Julgadora. †

Campinas, 15 de março de 1995.

  
Profa. Dra. Claudia Maria Bauzer Medeiros  
X

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





CM-00071873-2

FICHA CATALOGRAFICA ELABORADA PELA

BIBLIOTECA DO INECC DA UNICAMP

Aguiar, Cristina Dutra de  
Ag93i      Integracao de sistemas de banco de dados heterogeneos em  
aplicacoes de planejamento urbano / Cristina Dutra de Aguiar. --  
Campinas, [SP : s.n.], 1995.

Orientadora : Claudia Maria Bauzer Medeiros.  
Dissertacao (mestrado) - Universidade Estadual de Campinas,  
Instituto de Matematica, Estatistica e Ciencia da Computacao.

1. Banco de dados - Gerencia. 2. Planejamento urbano.  
I. Medeiros, Claudia Maria Bauzer. II. Universidade  
Estadual de Campinas. Instituto de Matematica, Estatistica  
e Ciencia da Computacao. III. Titulo.

# Integração de Sistemas de Banco de Dados Heterogêneos em Aplicações de Planejamento Urbano<sup>1</sup>

Cristina Dutra de Aguiar<sup>2</sup>

Departamento de Ciência da Computação  
IMECC – UNICAMP

Banca Examinadora:

- Claudia Maria Bauzer Medeiros (Orientadora)<sup>3</sup>
- Geovane Cayres Magalhães<sup>4</sup>
- Ana Carolina Salgado de Aguiar<sup>5</sup>
- Luiz Eduardo Buzato (Suplente)<sup>6</sup>

---

<sup>1</sup> Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação da UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

<sup>2</sup> A autora é formada em Ciência da Computação pela Universidade Federal de São Carlos.

<sup>3</sup> Professora do Departamento de Ciência da Computação – IMECC – UNICAMP.

<sup>4</sup> Pesquisador em Telecomunicações do Centro de Pesquisas e Desenvolvimento da TELEBRÁS. Professor em tempo parcial no Departamento de Ciência da Computação – IMECC – UNICAMP.

<sup>5</sup> Professora do Departamento de Informática – CCEN – UFPe.

<sup>6</sup> Professor do Departamento de Ciência da Computação – IMECC – UNICAMP.

Dedico

Aos meus pais

Ivor e Maria Helena

Ao meu noivo

Ricardo

Ao meu irmão

Juliano

# Agradecimentos

À professora e amiga Claudia pela orientação geral do trabalho.

Aos meus pais pelo apoio e estímulo constantes.

Ao Ricardo por seu amor e companheirismo.

A Telma pelo incentivo.

Ao prof. Geovane Cayres Magalhães e a Carlos Alberto Previdelli (Telebrás) pela colaboração durante a fase de modelagem da aplicação do CPqD Telebrás.

A Marco Augusto Weber (Eletropaulo) e a Otto Almeida (Carta Consultoria) pelas críticas e sugestões durante a fase de modelagem da aplicação da Eletropaulo.

A Fátima Pires e a Juliano Lopes de Oliveira pela contribuição na realização deste trabalho.

Ao CNPq pela concessão da bolsa de mestrado.

A FAEP pela concessão do auxílio ponte para finalização da escrita da dissertação.

Ao funcionário Luiz Cláudio Rosa da secretaria da pós-graduação da UNICAMP.

A todos aqueles que direta ou indiretamente contribuíram para a realização deste trabalho, em especial aos colegas do DCC – UNICAMP.

# Sumário

O nome Sistemas de Informações Geográficas (SIGs) denota todo software que manipula dados georeferenciados – dados conectados espacialmente à superfície terrestre. Os SIGs modernos são baseados em bancos de dados relacionais, estendidos de modo a suportar eficientemente as aplicações georeferenciadas. Estudos recentes indicam que o paradigma de orientação a objetos é mais indicado para este tipo de sistema. Entretanto, a migração de sistemas relacionais para orientados a objetos envolve altos custos.

Esta dissertação propõe uma solução para o problema que consiste na definição de mecanismos que permitam integrar os atuais sistemas (baseados em relações) e os novos sistemas (baseados em objetos), com ênfase em aplicações urbanas. A arquitetura proposta integra SGBDs orientados a objetos e relacionais projetados, respectivamente, a partir dos modelos OMT e ECR. Para permitir tal integração, foram desenvolvidas primitivas de mapeamento entre os conceitos destes dois modelos de dados, além de primitivas para a conversão de esquemas OMT em definições de classes do modelo do sistema orientado a objetos O<sub>2</sub>. O mecanismo proposto foi validado através da integração de duas aplicações reais que utilizam os elementos básicos do planejamento urbano. Estas aplicações são utilizadas pela Telebrás para o gerenciamento da rede de telefonia e pela Eletropaulo para o gerenciamento da rede elétrica.

# Abstract

The name Geographic Information Systems (GIS) denotes software that handles georeferenced data – data connected spatially to the earth surface. Modern GIS are based in relational database systems, extended to efficiently support georeferenced applications. Recent studies indicate that the object oriented paradigm is more adequate for this type of system. However, the migration of relational to object oriented systems is costly.

This dissertation presents a solution to this problem, which consists in defining mechanisms that allow the integration of the present (relation based) systems and the new (object based) systems, with emphasis in urban applications. The architecture proposed integrates object oriented and relational DBMS designed, respectively, using the OMT and ECR models. In order to allow this integration the dissertation developed primitive operations for mapping between both data models, as well as primitives for converting OMT schemas into schemas of the O<sub>2</sub> object oriented DBMS. This proposal was validated through the integration of two real life applications which use the basic elements of urban planning: Telebrás' telephone network management system and Eletropaulo's electrical network management system.

# Conteúdo

<b>1. Motivação e Estruturação da Dissertação</b>	<b>01</b>
<b>2. Conceitos Básicos</b>	<b>04</b>
2.1 Sistemas de Banco de Dados .....	04
2.2 Sistemas de Banco de Dados Heterogêneos .....	06
2.2.1 Problemas Apresentados pelo Ambiente SBDH .....	08
2.2.2 Identificação de Conflitos no Processo de Integração de Esquemas .....	09
2.3 Sistemas de Informações Geográficas .....	10
2.3.1 Funcionalidades de um SIG .....	15
2.3.2 Estrutura de um SIG .....	16
2.3.3 Formatos de Dados .....	17
2.3.4 O SIG e o Planejamento Urbano .....	19
<b>3. Arquitetura Proposta</b>	<b>21</b>
3.1 Descrição do Modelo .....	21
3.1.1 Escolha da Abordagem Utilizada para o Desenvolvimento do Esquema Conceitual .....	21
3.1.2 Escolha da Forma de Mapeamento Entre Visões dos Usuários e Esquemas Locais .....	22
3.1.3 Escolha do Modelo de Dados Global .....	24
3.2 A Metodologia OMT .....	26
3.2.1 O Modelo de Objetos .....	27
3.2.1.1 Objeto, Classe, Atributo e Operação .....	27
3.2.1.2 Associação, Multiplicidade, Qualificação e Agregação.....	29
3.2.1.3 Generalização .....	35
3.2.1.4 Restrição .....	37
3.3 O Modelo ECR .....	39
3.3.1 Entidade e Atributo .....	39
3.3.2 Categoria .....	41
3.3.3 Relacionamento e Cardinalidade .....	45
3.4 Um Modelo Orientado a Objetos para Aplicações Geográficas .....	49



<b>4. Algoritmos para Transformação de Modelos</b>	<b>53</b>
4.1 Mapeamentos Existentes	53
4.1.1 Modelo Orientado a Objetos → Modelo Relacional	53
4.1.2 Modelo Relacional → Modelo Orientado a Objetos	56
4.1.3 Modelo ECR → Modelo Relacional	58
4.2 Mapeamentos Propostos	60
4.2.1 Modelo ECR → Modelo OMT	60
4.2.1.1 Tipo-entidade e Categoria	61
4.2.1.2 Atributo de Tipo-entidade ou de Categoria	62
4.2.1.3 Relacionamento Binário	65
4.2.1.4 Atributo de Relacionamento	67
4.2.1.5 Relacionamento Ternário e de Ordem Superior	69
4.2.2 Modelo OMT → Modelo ECR	74
4.2.2.1 Classe, Atributo e Operação	74
4.2.2.2 Generalização	75
4.2.2.3 Associação Binária e Atributo de Ligação	76
4.2.2.4 Associação como uma Classe	79
4.2.2.5 Agregação	80
4.2.2.6 Associação Qualificada	82
4.2.2.7 Associação Ternária e de Ordem Superior	82
4.2.2.8 Instanciação	84
4.2.3 Modelo OMT → Modelo de Dados do Sistema O <sub>2</sub>	84
4.2.3.1 Classe, Atributo e Operação	88
4.2.3.2 Generalização	89
4.2.3.3 Associação Binária e Atributo de Ligação	91
4.2.3.4 Associação como uma Classe	95
4.2.3.5 Agregação	97
4.2.3.6 Associação Qualificada	98
4.2.3.7 Associação Ternária e de Ordem Superior	102
4.2.3.8 Instanciação	103
<b>5. Descrição das Aplicações e Visão de Alto Nível</b>	<b>109</b>
5.1 Aplicação CPqD Telebrás	109
5.1.1 O Modelo de Dados do Mapeamento Urbano Básico	111
5.2 Aplicação Eletropaulo	113
5.2.1 O Modelo de Dados do SIGRADE	115
5.3 Elementos Comuns e Proposta de Integração	116
<b>6. Modelagem das Aplicações e Integração</b>	<b>120</b>
6.1 Processo de Integração – Observações Iniciais	120
6.2 Modelagem de Mapeamento Urbano – Telebrás	121
6.3 Modelagem de Mapeamento Urbano – Eletropaulo	122

6.4 Integração .....	125
6.4.1 Tradução do Modelo Eletropaulo .....	125
6.4.2 Integração dos Esquemas Conceituais .....	125
6.4.3 Refinamento do Esquema Conceitual Integrado .....	128
6.4.4 Visão Geral da Integração .....	130
<b>7. Conclusões</b>	<b>133</b>
<b>Referências Bibliográficas</b>	<b>135</b>
<b>A Descrição Semântica dos Elementos do Modelo de Dados Telebrás</b>	<b>143</b>
<b>B Descrição Semântica dos Elementos do Modelo de Dados Eletropaulo</b>	<b>149</b>
<b>C Descrição Semântica dos Elementos do Modelo de Dados Integrado Refinado</b>	<b>157</b>

# Lista de Figuras

2.1	Representação do objeto espacial praça .....	13
2.2	Representação da fotografia aérea da praça .....	13
2.3	Representação dos objetos espaciais ponto, nó, segmento de linha, “string”, arco, cadeia e polígono .....	14
3.1	Arquitetura proposta .....	25
3.2	Representação gráfica da notação das classes .....	29
3.3	Associações binária e ternária .....	32
3.4	Multiplicidade de associações .....	32
3.5	Agregação .....	33
3.6	Árvore de agregação .....	33
3.7	Associação qualificada .....	34
3.8	Atributo de ligação .....	34
3.9	Uso de nomes de papéis .....	35
3.10	Hierarquia de generalização com subclasses disjuntas .....	36
3.11	Restrição ordenação .....	38
3.12	Restrição chave candidata .....	38
3.13	Restrição instanciação .....	38
3.14	Representação dos conceitos tipo-entidade e atributo .....	41
3.15	Representação de atributos básicos e adquiridos .....	41
3.16	Representação gráfica de categoria .....	42
3.17	Categoria de generalização .....	43
3.18	Exemplo de categoria de generalização .....	43
3.19	Categoria de subconjunto (categoria ISA).....	44
3.20	Categoria de subconjunto disjunta .....	44
3.21	Representação do conceito relacionamento .....	47
3.22	Representação gráfica da classificação dos relacionamentos .....	47
3.23	Exemplos de valores de $(i_1, i_2)$ .....	48
3.24	Representação do conceito nomes de conexão .....	49
3.25	O modelo de dados MODGEO <sub>2</sub> .....	50

4.1	Mapeamento ECR → OMT (parte 1)	70
4.2	Mapeamento ECR → OMT (parte 2)	71
4.3	Mapeamento ECR → OMT (parte 3)	72
4.4	Mapeamento ECR → OMT (parte 4)	73
4.5	Mapeamento OMT → ECR (parte 1)	85
4.6	Mapeamento OMT → ECR (parte 2)	86
4.7	Mapeamento OMT → ECR (parte 3)	87
4.8	Mapeamento OMT → O <sub>2</sub> (parte 1)	105
4.9	Mapeamento OMT → O <sub>2</sub> (parte 2)	106
4.10	Mapeamento OMT → O <sub>2</sub> (parte 3)	107
4.11	Mapeamento OMT → O <sub>2</sub> (parte 4)	108
5.1	Representação gráfica dos conceitos logradouro e linha central	111
5.2	Representação gráfica do conceito testada	112
5.3	Esquema simplificado do modelo de dados do MUB	113
5.4	Estrutura funcional do sistema SIGRADE	114
5.5	Regras de integração para casos mais comuns	119
6.1	Correspondência entre os elementos do modelo Telebrás e as categorias do modelo MODGEO <sub>2</sub>	122
6.2	Esquema conceitual do MUB Telebrás	123
6.3	Esquema conceitual do mapeamento urbano da Eletropaulo	124
6.4	Tradução do esquema conceitual do mapeamento urbano da Eletropaulo	126
6.5	Esquema integrado	127
6.6	Refinamento do esquema integrado	129

# Capítulo 1

## Motivação e Estruturação da Dissertação

Esta dissertação tem por objetivo propor um modelo para a integração de bancos de dados heterogêneos em sistemas de informações geográficas visando aplicações que envolvam elementos de planejamento urbano.

Sistemas de informações geográficas (SIGs) manipulam enormes volumes de dados, armazenados em arquivos especiais, ou mais recentemente, em bancos de dados relacionais. A utilização de bancos de dados relacionais exige o acoplamento do SIG a módulos externos para tratamento de relacionamentos espaciais [Sou+93]. Estes sistemas são complexos, pois exigem dos profissionais de aplicação o conhecimento do banco de dados e dos módulos, cada qual com características diferentes.

Estudos recentes indicam que o uso de bancos de dados orientados a objetos ajudam a simplificar tais sistemas, além de permitir aos usuários (geógrafos, geólogos, engenheiros, arquitetos e outros) maior participação no desenvolvimento da aplicação.

Com isso, a tendência futura para aplicações que utilizam SIGs é a orientação a objetos. Neste sentido, algumas experiências já estão sendo realizadas, tanto sob a forma de protótipos (em universidades e centros de pesquisas) quanto sob a forma de alguns produtos. Entretanto, o investimento atual em SIGs que utilizam sistemas relacionais impede a simples migração destes para novos sistemas (orientados a objetos). Assim, surge o problema de integrar sistemas existentes aos sistemas mais recentes, oferecendo uma interface única ao usuário. A dissertação se propõe a solucionar este problema, com ênfase em aplicações urbanas.

O planejamento urbano representa uma das principais áreas de utilização dos SIGs, uma vez que é responsável pelo controle e organização das cidades. O crescimento populacional, geográfico e econômico das cidades exige um sistema capaz de prever e direcionar a expansão da malha urbana, além de gerenciar a utilização dos seus recursos públicos.

A base de todos os procedimentos da implantação do georeferenciamento no planejamento urbano é o mapeamento urbano básico, o qual consiste no conjunto de informações gráficas e alfanuméricas referentes à base cartográfica das plantas cadastrais. Atuando como uma camada subjacente, é sobre este mapeamento que são desenvolvidas as mais variadas aplicações. Desta forma, a integração realizada nesta dissertação é feita unicamente em relação ao mapeamento urbano básico.

Com base nessas considerações, foi proposta uma arquitetura para a integração de SGBDs orientado a objetos e relacional, com o objetivo de oferecer aos usuários uma camada virtual através da qual estes podem acessar os dados localizados nos BDs de maneira transparente. Estes SGBDs componentes considerados no processo de integração são representados, respectivamente, por esquemas que utilizam os modelos OMT [Rum+91] e ECR [EWH85]. Para tanto, foram desenvolvidas primitivas de mapeamento entre os conceitos destes dois modelos de dados, além de primitivas para a conversão de esquemas OMT em definições de classes do modelo do sistema orientado a objetos O<sub>2</sub> [Deu+92].

A validação do mecanismo proposto foi realizada através da integração de duas aplicações reais, de grande porte, que utilizam os elementos básicos do planejamento urbano, tais como logradouro, lote e linha central, entre outros. A primeira aplicação é utilizada pela Telebrás para o gerenciamento da rede de telefonia e corresponde, neste trabalho, ao esquema conceitual local projetado no modelo OMT. A segunda aplicação é utilizada pela Eletropaulo para o gerenciamento da rede elétrica e corresponde, neste trabalho, ao esquema conceitual local projetado no modelo ECR. Na integração destas aplicações foram utilizadas tanto as primitivas de mapeamento oferecidas quanto o processo de integração descrito em [SP94].

Finalmente, para mostrar o efeito da integração, algumas consultas foram formuladas tanto no esquema original (ambiente dos SGBDs componentes) quanto no esquema global (ambiente do SGBD heterogêneo). Através da formulação destas consultas, foi constatado que não houve qualquer perda de informações devido ao processo de integração.

Além deste capítulo introdutório, a dissertação é composta por mais seis capítulos.

O capítulo 2 descreve uma visão geral de bancos de dados heterogêneos, destacando os principais problemas apresentados por este ambiente. Este capítulo também discute conceitos de sistemas de informações geográficas, destacando as suas funcionalidades e a importância do planejamento urbano.

A arquitetura proposta para resolver o problema de integração dos BDs orientado a objetos e relacional é apresentada no capítulo 3. Neste capítulo também são sumarizados os modelos OMT e ECR, os quais são utilizados nos projetos dos esquemas conceituais dos SGBDs componentes. Em adição, é descrito o modelo orientado a objetos para aplicações geográficas MODGEO<sub>2</sub>.

O capítulo 4 descreve os mapeamentos propostos entre os conceitos do modelo ECR e os conceitos do modelo OMT. Além disso, tendo em vista a completude do trabalho realizado, são descritos os mapeamentos entre o modelo OMT e o modelo ECR e entre o modelo OMT e o modelo utilizado pelo sistema O<sub>2</sub>.

As aplicações geográficas envolvendo planejamento urbano são descritas no capítulo 5. Neste capítulo também são destacados os elementos comuns a ambas aplicações.

A integração das aplicações descritas no capítulo 5 utilizando o mecanismo proposto é realizada no capítulo 6, que também destaca algumas consultas.

A dissertação é concluída no capítulo 7, com as contribuições esperadas e sugestões para extensões. Informações adicionais sobre os modelos utilizados no processo de integração são encontradas nos apêndices A, B e C.

# Capítulo 2

## Conceitos Básicos

### 2.1 Sistemas de Banco de Dados

Um sistema de banco de dados (SBD) é composto por um programa de software chamado sistema gerenciador de banco de dados (SGBD) e por um conjunto de dados, chamado banco de dados (BD) [EP90]. Um dos objetivos dos BDs é oferecer algum nível de abstração, escondendo os detalhes de armazenamento dos dados dos usuários dos BDs. Para se obter esta abstração, é utilizado o conceito de **modelo de dados**, que é uma coleção de ferramentas conceituais utilizadas para descrever a estrutura do BD. A estrutura de um BD engloba a descrição dos seus dados, os relacionamentos entre eles, a semântica e as restrições que atuam sobre estes dados. A maioria dos modelos de dados também inclui um conjunto de operações utilizadas para especificar consultas e atualizações nos BDs [EN89].

Os SGBDs apresentam gerenciamento automático de erros, capacidade de consultas “ad-hoc”, consistência global de dados (ou seja, a ausência de dados inconsistentes), autorização para acesso e eficiente manipulação das informações armazenadas. É através da **linguagem de consulta** oferecida pelo SGBD que os usuários podem submeter consultas ao sistema. A linguagem de consulta oferecida é altamente dependente do modelo de dados utilizado.

Um SBD pode ser tanto centralizado quanto distribuído [SL90]. Um SBD centralizado consiste em um único SGBD centralizado que gerencia o BD localizado em um único sistema de computação. Já um SBD distribuído consiste em um único SGBD distribuído que gerencia múltiplos BDs, garantindo aos usuários transparência de distribuição. Os BDs podem residir em um único sistema de computação ou em vários sistemas de computação, os quais podem diferir em hardware, software ou suporte de comunicação. O SGBD distribuído suporta apenas um único modelo de dados e uma única linguagem de



consulta, além de um único mecanismo de gerenciamento de transação e de otimização de consultas, entre outros.

O uso de vários SBDs em uma organização (por exemplo, o caso da UNICAMP) cria a necessidade de compartilhamento de dados. Uma primeira solução para este problema é a conversão de todos os sistemas para um único modelo de dados com um único método de acesso. Porém, esta conversão não é aceitável, porque além de requerer alto custo, há também a necessidade de modelos de dados específicos para representar as diferentes aplicações do mundo real.

Desta forma, a solução encontrada é a **integração** ou a **interoperabilidade** dos diversos SBDs. Há uma diferença sutil entre estes termos. Integração significa combinação de BDs e sistemas como um todo, ao passo que interoperabilidade significa o compartilhamento dos dados entre os BDs, sem uma integração global.

[Hsi92] destaca cinco requisitos necessários ao compartilhamento global de dados:

- **transparência de acesso aos diversos BDs:** os usuários devem acessar os diferentes BDs como se estivessem acessando os SGBDs a que estão acostumados.
- **autonomia local de cada um dos BDs:** os diversos BDs existentes devem compartilhar os seus dados sem comprometer suas restrições de integridade, aplicações específicas ou mecanismos de segurança.
- **característica multimodelo e multilinguagem:** os BDs que participam do compartilhamento de dados global devem possuir diferentes modelos de dados e, conseqüentemente, diferentes linguagens de manipulação de dados.
- **existência de uma arquitetura paralela e de propósito geral para suportar os diversos BDs:** os BDs e os seus respectivos SGBDs devem ser suportados por computadores e dispositivos de armazenamento secundário dedicados, organizados de uma maneira paralela.
- **mecanismos de controle de concorrência eficientes:** os diversos acessos concorrentes aos BDs devem ser controlados eficientemente para que as consultas realizadas pelos usuários sejam executadas corretamente.

Surge, então, o conceito de **sistemas de banco de dados heterogêneos**.

## 2.2 Sistemas de Banco de Dados Heterogêneos

Um sistema de banco de dados **heterogêneo** (SBDH) é um pacote de software construído sobre vários SGBDs heterogêneos preexistentes, os quais são integrados em um único SBDH de uma maneira cooperativa. Estes SGBDs são preexistentes no sentido que foram criados independentemente, de uma forma não coordenada e sem considerar que um dia seriam integrados. Eles são heterogêneos no sentido que operam em diferentes ambientes (hardware, protocolos de comunicação, sistemas operacionais) e podem utilizar diferentes SBDs. No ambiente SBDH, cada SGBD preexistente recebe o nome de SGBD componente. Cada SGBD componente pode ser, por sua vez, um SBD centralizado, um SBD distribuído ou um outro SBDH. Além disso, um SGBD componente pode participar de um ou mais SBDHs distintos.

O SBDH cria uma integração lógica dos diversos SGBDs componentes, oferecendo aos usuários acesso uniforme aos dados contidos nos vários BDs, sem migrar os dados e sem requerer que os usuários conheçam tanto a localização quanto as características dos diferentes SGBDs. Além disso, o uso da abordagem SBDH permite que aplicações preexistentes permaneçam operacionais, e que novas aplicações possam acessar os dados localizados nos vários SGBDs componentes. Em outras palavras, um usuário pode acessar os dados dos SGBDs que participam do SBDH tanto indiretamente através da interface do SBDH quanto diretamente através da interface dos diversos componentes.

Além da **heterogeneidade** e **distribuição**, uma terceira característica inerente aos SBDHs é a **autonomia** dos seus componentes ([SL90, Man+92]). Existem quatro tipos diferentes de autonomia: de projeto, de comunicação, de execução e de associação.

De acordo com a autonomia de projeto ou decisão, os SGBDs são independentes na escolha dos seus modelos de dados e linguagem de consulta, na forma de gerenciamento dos dados e na definição das restrições que atuam sobre este gerenciamento, na interpretação semântica dos seus modelos, na determinação das funcionalidades oferecidas ao sistema global e na implementação de suas estruturas e algoritmos. Este tipo de autonomia é um dos principais causadores da heterogeneidade entre SGBDs componentes.

Já a autonomia de comunicação permite que os SGBDs sejam independentes para determinar se irão ou não se comunicar com os outros SGBDs componentes, além de decidir quando e como isso será realizado.

A autonomia de execução, por sua vez, determina que os SGBDs são independentes para executar consultas locais e externas, definindo sua ordem de execução. Uma vez que os SGBDs componentes têm total controle sobre a execução das suas transações, eles podem abortá-las a qualquer momento durante as suas execuções, desde que estas não satisfaçam as restrições locais. Adicionalmente, as operações locais não são afetadas logicamente pela participação do SGBD componente no ambiente SBDH.

Finalmente, de acordo com a autonomia de associação, os SGBDs são independentes para determinar qual informação compartilharão com o sistema global, quais consultas globais irão atender, quando irão iniciar a sua participação com o sistema global e quando irão finalizá-la.

A preservação da autonomia é um ponto chave no ambiente SBDH, uma vez que a integração dos SGBDs em um SBDH não deve infringir o direito destes gerenciarem os seus recursos sem a interferência do SBDH.

Um SBDF (SBD federado) é sinônimo de um SBDH [EP90]. Um SBDF pode ser classificado como fracamente ou fortemente acoplado ([SL90, Cle+93, Oli93]), dependendo de quem gerencia a federação e de como os componentes são integrados.

A abordagem SBDF fracamente acoplado parte do princípio que os diversos SBDs devem ser interoperáveis. Desta forma, neste ambiente, é responsabilidade dos usuários criar e manter a federação, uma vez que o SBDF não exerce nenhum controle sobre esta. Para tanto, os usuários devem conhecer tanto a localização quanto os problemas de heterogeneidade dos SGBDs componentes. As vantagens apresentadas por esta abordagem são que nenhum esforço é exigido ao administrador do BD (ABD) para resolver a heterogeneidade semântica existente entre os SGBDs componentes e que, além disso, não é necessário que o ABD antecipe as necessidades dos usuários da federação, uma vez que estes conseguem acessar os dados desejados quando necessário. Por outro lado, esta abordagem possui algumas desvantagens, tais como a ausência da transparência de localização dos dados dos diversos BDs e a dificuldade da realização das operações de atualização. Outros termos utilizados para SBDF fracamente acoplados são SBDMs (SBD Múltiplos) e sistemas interoperáveis.

Já a abordagem SBDF fortemente acoplado parte do princípio que os esquemas que representam os vários SGBDs componentes devem ser integrados. Desta forma, nesta abordagem é responsabilidade da federação e dos seus administradores criar, controlar e manter a federação. A vantagem apresentada advém do fato de que o SBDF fortemente acoplado esconde dos usuários da federação a localização e a heterogeneidade semântica dos BDs locais, através de esquemas globais. Por outro lado, a integração de esquemas é uma tarefa difícil de ser realizada.

Em suma, todos os termos acima citados descrevem um sistema distribuído que inclui um componente global para acessar as informações globalmente compartilhadas, e múltiplos componentes locais que gerenciam as suas informações. As diferenças estão na estrutura do componente global e como ele interage com os componentes locais [BHP92].

## 2.2.1 Problemas Apresentados pelo Ambiente SBDH

A integração de diversos SGBDs em um ambiente SBDH origina vários problemas, devido às características de heterogeneidade, autonomia e distribuição destes componentes.

Os principais problemas do ambiente SBDH são: o **gerenciamento de transações** ([BK91, Geo91, GRS91, BGS92, BST92, JS92, Raz92, Cle+93, Dre+93, HHS93, ST93, Woe+93]), a **autonomia local**, o **gerenciamento de consultas** ([HB91, DKS92, Cle+93, HK93, LS93a, RCD93]), a **transparência de localização** e a **manutenção do esquema** ([BHP92]), o **gerenciamento de autorização**, a **integração de esquemas** ([BLN86, Tho+90, BHP92, Oli93]) e a **diferença na representação dos dados** ([AP93, Cle+93, Dre+93, Lim+93, LS93b, LSS94]).

O problema de integração de esquemas está relacionado à maneira segundo a qual o usuário pode ver logicamente os dados localizados nos múltiplos BDs. Cada BD tem um esquema, o qual descreve a estrutura dos seus dados, e cada usuário tem uma visão, a qual descreve que porção do dado é de seu interesse. A visão particular do usuário não pode ser afetada pela participação do SGBD no ambiente SBDH.

Resumidamente, este problema pode ser definido como o processo de desenvolvimento de um esquema conceitual global que engloba uma coleção de esquemas locais. Existem duas abordagens básicas para este problema: abordagem de esquema global e abordagem de esquema federado.

Na abordagem de esquema global, os diversos esquemas locais que participam do sistema são integrados em um único esquema global. Em outras palavras, os esquemas locais desenvolvidos independentemente são integrados, suas diferenças semânticas são resolvidas e um esquema global final é oferecido ao usuário. Este esquema global é, portanto, independente da heterogeneidade dos SGBDs e das representações de dados. As visões que os diferentes grupos de usuários e suas respectivas aplicações podem possuir podem ser definidas sobre o esquema global. Quando esta abordagem é utilizada, o problema de integração de esquemas também é conhecido como problema de integração de visões.

Já na abordagem de esquema federado, não há a criação de um esquema global. A tarefa de integrar os dados localizados nos múltiplos BDs é deixada a cargo dos usuários. Para tanto, estes devem conhecer tanto as diferenças de representação dos dados quanto as suas localizações. O sistema global apenas oferece diversos esquemas locais, os quais descrevem os dados dos diversos SGBDs componentes que podem ser compartilhados. Geralmente, a linguagem oferecida por estes tipos de esquemas possui uma grande variedade de funções e é poderosa o suficiente para oferecer maior controle sobre a informação sendo manipulada.

Uma lista abrangente de projetos que utilizam as abordagens acima citadas pode ser encontrada em [BHP92].

## 2.2.2 Identificação de Conflitos no Processo de Integração de Esquemas

A identificação de conflitos é de essencial importância para o projeto de BDs heterogêneos. Segundo [BLN86], um conflito entre duas representações  $R_1$  e  $R_2$  do mesmo conceito, pertencentes a esquemas distintos, surge quando estas representações não são idênticas. Duas representações são idênticas quando são exatamente as mesmas, ou seja, os mesmos construtores foram utilizados e as mesmas restrições foram aplicadas.

A existência de conflitos entre duas ou mais representações está relacionada ao fato de que diferentes usuários modelam o mesmo pedaço do mundo real de maneiras distintas, de acordo com as suas percepções. A esta multiplicidade de representações de um determinado problema do mundo real dá-se o nome de relativismo semântico [SP94].

Desta forma, o processo de integração depende da identificação de similaridades e diferenças existentes entre os elementos dos diferentes esquemas, além da identificação de conjuntos de elementos distintos que são relacionados entre si por alguma propriedade semântica. De acordo com [SP94], os conflitos podem ser divididos em três grupos: conflitos de nome, conflitos semânticos e conflitos estruturais.

O primeiro tipo de conflito, **conflito de nome**, refere-se aos nomes utilizados para representar os diferentes elementos existentes nos esquemas a serem integrados. Diferentes nomes podem ser aplicados ao mesmo elemento (problema dos sinônimos) ou o mesmo nome pode ser aplicado a diferentes elementos (problema dos homônimos). Conflitos de nome não estão relacionados apenas a elementos do mesmo tipo, ou seja, podem ocorrer, por exemplo, entre um objeto e um atributo. Um exemplo de sinônimo ocorre quando o nome *cliente* é utilizado para representar, em um esquema, todos os clientes atendidos por uma loja, enquanto que o nome *comprador* é utilizado em outro esquema para representar a mesma situação.

Após a determinação dos conflitos de nome, devem ser identificados os **conflitos semânticos**. Este tipo de conflito surge quando o mesmo elemento é modelado nos diferentes esquemas, porém representando conjuntos que se sobrepõem. Em outras palavras, o conjunto de instâncias do elemento de um esquema é mais abrangente do que o conjunto de instâncias do elemento do outro esquema. Por exemplo, em uma visão, uma classe *estudante* representa todos os alunos de uma universidade, ou seja, alunos de graduação, pós-graduação e doutorado. Já em uma segunda visão, uma classe *estudante\_grad* representa apenas os alunos de graduação da universidade.

Finalmente, **conflitos estruturais** surgem sempre que diferentes construtores estruturais são utilizados para modelar o mesmo conceito representado em diferentes visões. Como exemplo, o mesmo conjunto de objetos do mundo real pode ser representado como um tipo-entidade em uma visão e como um atributo de um tipo-entidade em outra visão. A diversidade de conflitos estruturais que podem aparecer em um problema de integração depende da semântica do modelo de dados utilizado.

Em geral, as discrepâncias existentes entre os esquemas apresentam mais do que um tipo de conflito.

## 2.3 Sistemas de Informações Geográficas

Sistemas de informações geográficas (SIGs) são sistemas de banco de dados que permitem a captação, o processamento, o armazenamento, a recuperação e a análise dos dados geográficos, além da visualização destes [Sou+93].

Exemplos de aplicações que utilizam SIGs são: planejamento urbano, planejamento regional, gerenciamento e mapeamento de recursos naturais, tais como gerenciamento agrícola, ecológico e estudos biológicos, mapeamentos estatísticos e determinação de censo populacional, determinação de áreas suscetíveis a terremotos, inundações ou secas, planejamento do uso da terra, entre outras [Dan90].

Gráficos, fotos e mapas são saídas padrão de consultas SIG. Um **mapa** é uma representação gráfica de uma particular área, como por exemplo, uma cidade, um país, um continente. Um mapa tem por finalidade mostrar as principais características de uma área e a maneira na qual estas características se relacionam umas com as outras, podendo resultar de combinação de outros mapas (sobreposição). Dependendo do objetivo de utilização do mapa, diferentes características podem ser representadas. Por exemplo, para um mapa de uma cidade, seria interessante representar escolas, praças e hospitais. Já para o mapa de um continente, seria interessante representar os países e as suas capitais.

O mapa serve como uma fonte de armazenamento de informações a partir das quais decisões envolvendo distância, direção, adjacência, localização relativa e outras operações mais complexas podem ser tomadas [PM90]. A sua utilidade varia desde a solução de problemas simples envolvendo a rotina diária, tal como a determinação do caminho que leva uma pessoa de sua casa até o supermercado mais próximo, ou a simples determinação das principais características de uma cidade, até a solução de problemas mais complexos envolvendo propósitos governamentais e comerciais, tal como o gerenciamento de recursos minerais ou estratégias militares. O principal aspecto a ser considerado é que, independentemente da sua utilização, o mapa tem um papel de destaque dentro da sociedade moderna.

Elementos de dados que podem ser representados em um mapa geralmente são referenciados como **objetos espaciais** ou **georeferenciados**. Estes objetos espaciais são armazenados com base em um sistema de coordenadas padrão (por exemplo, latitude, longitude e altitude, esta última com relação ao nível do mar). A representação visual destes objetos espaciais é determinada através da utilização de técnicas de simbolização cartográfica. De maneira simplificada, estas técnicas associam diferentes figuras geométricas em diferentes cores aos vários objetos espaciais existentes.

A utilização de mapas para o armazenamento, recuperação e consulta dos objetos espaciais possui algumas desvantagens ([Mar90, PM90]):

- **limitação de velocidade e de capacidade de armazenamento de informações.** O conjunto de informações manipulado por um mapa é geralmente grande e complexo. A utilização de técnicas manuais para o projeto e uso de mapas afeta tanto a eficiência do armazenamento e da recuperação das informações quanto o desenvolvimento de técnicas quantitativas e analíticas que envolvem estes dados. Neste caso, eficiência diz respeito à velocidade de obtenção ou armazenamento das informações e à precisão das mesmas. Um problema de eficiência se refere à implementação de técnicas quantitativas e analíticas tendo em vista o volume e a variação dos dados. Um exemplo é a determinação do número de escolas de um dado estado, considerando que existem apenas mapas dos municípios deste estado, ou da distância em quilômetros de uma determinada cidade a um rio importante.
- **dificuldade na integração dos mapas.** A maior parte das consultas realizadas em mapas requer a junção dos objetos espaciais espalhados entre os distintos mapas existentes (por exemplo, mapa hidrográfico, mapa de uso de solos). Para tanto, dois problemas devem ser solucionados: a diferença de escalas e a utilização de símbolos cartográficos iguais que representam diferentes objetos espaciais. No primeiro caso, os vários mapas sendo pesquisados devem ser alterados de forma a possuírem a mesma escala de representação. Em seguida, estes mapas devem ser sobrepostos, para que as incompatibilidades existentes possam ser eliminadas. Finalmente, a sobreposição deve mostrar os locais nos quais ocorrem a justaposição espacial dos diferentes objetos desejados. No segundo caso, especial atenção deve ser dada aos símbolos cartográficos que representam os diferentes objetos espaciais. Pode ser que, em um mapa hidrográfico, os rios sejam representados por segmentos de linha e que, em um mapa territorial, estes mesmos segmentos de linha sirvam para representar divisas entre dois territórios. Quando a justaposição destes mapas for realizada, estas incompatibilidades devem ser resolvidas.
- **dificuldade na atualização dos mapas.** O processo de realizar a alteração de um mapa existente é caro, demorado e pode gerar mapas inconsistentes. A alteração de um mapa requer que as mudanças realizadas em qualquer um dos seus objetos espaciais sejam refletidas nos outros mapas que possuam estes mesmos objetos ou que possuam objetos relacionados a estes. Por exemplo, pode-se considerar que a mesma cidade seja representada tanto em um mapa político quanto em um mapa

territorial. Qualquer alteração nesta cidade deveria ser realizada em ambos os mapas. Não existe, porém, nenhuma garantia que isto irá ocorrer.

SIGs, por serem automatizados ([Dan90, Mar90, PMS93]), devem atender aos seguintes requisitos [Smi+89]:

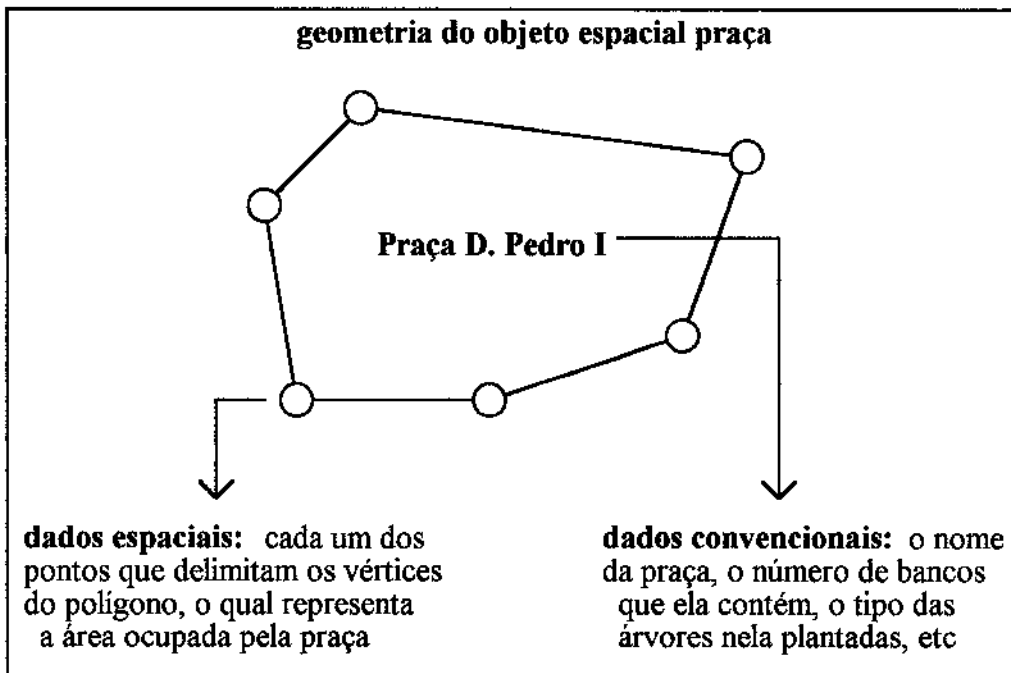
- habilidade de manipular grandes quantidades de objetos espaciais.
- habilidade de realizar consultas rápidas sobre a existência, localização e propriedades dos objetos espaciais.
- eficiência na manipulação de consultas mais complexas envolvendo sobreposição de objetos espaciais.
- flexibilidade para expansão, de acordo com o número de usuários do sistema.
- capacidade de aprender as características dos objetos espaciais, permitindo, desta forma, determinar acontecimentos futuros (como, por exemplo, determinar áreas suscetíveis a terremotos).

A eficiência de um ambiente SIG está diretamente relacionada à sua habilidade de manipular diferentes tipos de dados. Dados espaciais, dados convencionais e dados pictóricos são os três tipos de dados básicos que modelam os objetos georeferenciados:

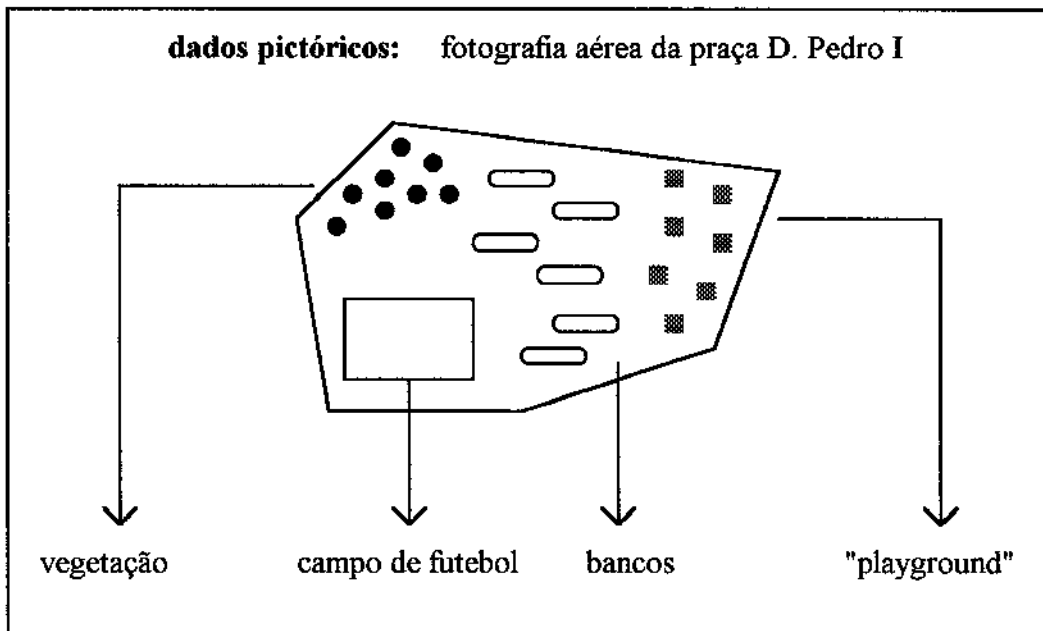
- **dados espaciais:** são os dados que indicam a descrição da geometria do objeto espacial (por exemplo, o limite geográfico de uma praça). Estes dados espaciais descrevem a geometria do objeto com relação à sua localização em referência à superfície terrestre. SIGs utilizam dois tipos básicos de formatos de dados espaciais: vetorial e varredura.
- **dados convencionais:** são os dados comumente encontrados em bancos de dados convencionais. Isto é, são as características não espaciais que descrevem o fenômeno geográfico independente de sua localização (por exemplo, o nome da praça). Estes dados convencionais são alfanuméricos.
- **dados pictóricos:** são os dados originados por sensoriamento remoto (por exemplo, imagens de satélite) ou por fotografia aérea de uma determinada região geográfica (uma fotografia da praça).

O objeto espacial praça e os seus três tipos de dados básicos são mostrados nas figuras 2.1 e 2.2.





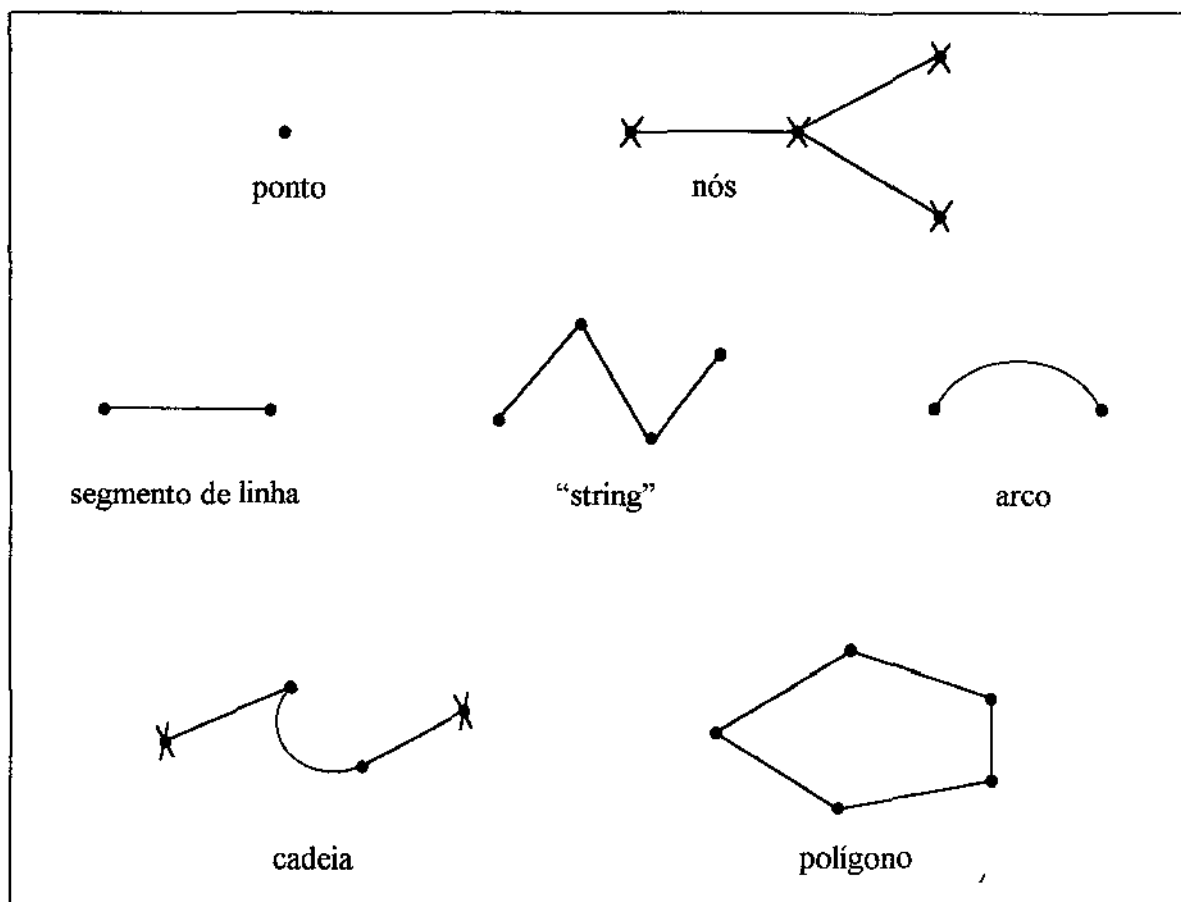
**Figura 2.1** Representação do objeto espacial praça.



**Figura 2.2** Representação da fotografia aérea da praça.

Os dados espaciais de um objeto espacial permitem tanto a determinação de alguns atributos adicionais para este objeto quanto a realização de operações espaciais. Um exemplo para o primeiro caso seria a determinação do atributo área. Já para o segundo caso, um exemplo seria a determinação dos hotéis mais próximos à praça.

No formato vetorial, existem três tipos básicos para representar a localização espacial de um objeto espacial: ponto, linha e polígono [Dan90, SE90, WHM90, Ant+91]. A seguir são descritas as definições destes tipos, além de algumas notações derivadas. A figura 2.3 representa estes objetos.



**Figura 2.3** Representação dos objetos espaciais ponto, nó, segmento de linha, "string", arco, cadeia e polígono.

- **ponto:** é um objeto espacial de dimensão zero que especifica a localização geográfica através de um conjunto de coordenadas, geralmente representadas como um par de números. Um ponto não possui área. Pontos são utilizados também para a definição de objetos espaciais mais complexos, como linhas e polígonos.
- **nó:** é um tipo especial de ponto, portanto também um objeto zero dimensional e sem área. Um nó representa uma junção topológica ou um ponto final e pode especificar uma localização geográfica.
- **linha:** é um objeto espacial unidimensional. Um segmento de linha é uma linha entre dois pontos. Algumas formas especiais de linhas incluem "string" (uma série de segmentos de linha), arco (seqüência de pontos em forma de curva) e cadeia

(seqüência direcionada não interceptada de segmentos de linhas ou arcos, delimitada por um nó inicial e um nó final).

- **polígono:** é um objeto espacial bidimensional que representa uma área fechada. Polígonos simples são áreas não divisíveis, enquanto que polígonos complexos são divididos em áreas de diferentes características.

Um mesmo objeto pode ter diferentes representações em função da escala ou do nível de detalhe. Assim, um lago pode ser representado por um ponto denotando a sua centróide ou por um polígono, dependendo da escala utilizada.

Além dos atributos, localização espacial e representação pictórica dos objetos espaciais, uma quarta característica que é particularmente relevante aos sistemas SIGs é o tempo [Dan90, MM93, PMS93]. Esta característica é importante uma vez que, dependendo da data na qual a foto do satélite foi tirada ou na qual a representação dos objetos foi realizada, estes podem ter sofrido alterações. Utilizando o exemplo da praça acima mencionado, pode ser que em dois anos a quadra de futebol não exista mais, e que em seu lugar exista um coreto. Assim, consultas devem levar o tempo em consideração.

### 2.3.1 Funcionalidades de um SIG

De acordo com [Ant+91, Mag91], os SIGs devem oferecer três funcionalidades distintas: cartográfica, de banco de dados e analítica.

A primeira delas enfoca os aspectos cartográficos de SIGs, com relação ao processamento e geração de mapas de uma maneira precisa e confiável. Os objetivos da funcionalidade cartográfica englobam a obtenção dos dados dos mapas, a manipulação gráfica destes (como exemplo a adição, a remoção e a alteração de elementos), a geração de “displays” gráficos e o processo de traçado.

Já a funcionalidade de banco de dados está relacionada com todas as vantagens apresentadas pelos SGBDs convencionais, tais como formatação padronizada dos dados, a possibilidade de evitar dados inconsistentes, a utilização de modelos de dados e linguagens de consultas específicas e a existência de rotinas de otimização e mecanismos de acesso eficientes, além das capacidades de gerenciamento de transação e recuperação de falhas. No entanto, os SGBDs voltados para SIGs devem realizar o armazenamento e o gerenciamento de objetos espaciais, oferecendo capacidades para suportar interação eficiente entre dados espaciais e dados convencionais. Devido à natureza dos dados espaciais, uma linguagem de consulta para SIGs é mais complexa do que as linguagens de consultas dos SGBDs convencionais. Além das facilidades encontradas nas linguagens tradicionais, a linguagem de consulta para SIG deve também se preocupar com relacionamentos geométricos e topológicos. A manipulação de dados não convencionais implica em uma extensão nos níveis mais internos, tais como otimizador de consultas,

métodos de acesso e armazenamento físico dos dados [Sou+93]. Desta forma, SGBDs devem ser estendidos para permitir tratamento de aplicações SIG.

Finalmente, a funcionalidade analítica tem por finalidade oferecer operações de cálculo. Estas operações englobam, por exemplo, a determinação da distância entre dois objetos espaciais, a determinação do comprimento de uma linha ou da área ocupada por uma região. Outras operações de cálculo estão relacionadas com cálculos estatísticos e previsões de acontecimentos futuros baseados nos dados atuais. No primeiro caso, um exemplo seria a determinação do número de escolas de um município, ou a determinação da porcentagem de eleitores de um determinado partido político. Para o segundo caso, um exemplo seria a determinação das áreas suscetíveis a terremotos ou a previsão de chuvas fortes em um período futuro. Finalmente, operações de “overlay” estão relacionadas com a sobreposição, junção, união e intersecção de dois ou mais mapas (ou objetos destes mapas) de modo a originar um mapa final. Uma característica comum à maior parte das aplicações SIGs é a dependência destas aplicações às camadas temáticas, as quais consistem de coleções de dados geográficos específicos para uma determinada região [PMS93]. Por exemplo, pode-se citar a camada de hidrografia, de relevo e de solo. Um SIG deve ter a capacidade de combinar duas ou mais camadas a fim de oferecer diferentes visões da região sendo analisada.

### 2.3.2 Estrutura de um SIG

Segundo [Cow90], um SIG constitui um tipo particular de sistema de informação. Como tal, este deve fornecer procedimentos para captação (aquisição e entrada de dados), gerenciamento, manipulação, análise e saída de dados, de maneira integrada.

O processo de captação dos dados é tipicamente a função mais cara dos ambientes SIGs [Dan89], tanto em custo monetário quanto em tempo gasto. Este processo pode ser dividido em duas fases: a fase de aquisição dos dados espaciais propriamente ditos e a fase de pré-processamento. A primeira fase envolve a identificação e a coleta de dados necessários a uma aplicação e o posterior armazenamento destes dados em uma base de dados qualquer. Os dados espaciais que servem de entrada para esta fase encontram-se em uma variedade de formas, as quais podem ser pedaços de papel, mapas produzidos manualmente, fotografias ou materiais digitais. Algumas técnicas utilizadas nesta fase são a digitalização manual ou semi-automática dos dados, o “escaneamento” de fotos ou mapas, a utilização de procedimentos de coordenadas geométricas para o cálculo e a entrada das coordenadas dos objetos espaciais e a tradução de arquivos digitais existentes. Um estudo detalhado destas técnicas pode ser encontrado em [Dan89, Dan90, SE90, Ant+91]. Já a fase de pré-processamento tem por finalidade converter a base de dados gerada na fase de captura para uma forma suportada pelo banco de dados oferecido pelo SIG. Isto é, os dados capturados inicialmente pelo sistema devem ser convertidos para tipos de dados, sistemas de coordenadas e estruturas de dados compatíveis ao BD do SIG. Alguns procedimentos de pré-processamento são: conversão de formatos, redução dos

dados e generalização, detecção de erros e interpolação, entre outros. Mais detalhes sobre este assunto podem ser encontrados em [SE90].

O processo de gerenciamento de dados tem por finalidade gerenciar os objetos espaciais armazenados no BD do SIG, tornando-os disponíveis aos usuários. Algumas operações típicas são a criação do banco de dados, o acesso, a entrada, a alteração, a eliminação e a recuperação dos dados armazenados. Todas estas operações devem ser realizadas eficientemente, de maneira transparente aos usuários. Outra função importante do processo de gerenciamento dos dados é a segurança do sistema, permitindo que apenas usuários autorizados possam acessar os dados a eles permitidos. Mais detalhes podem ser encontrados em [SE90, Ant+91].

Os processos de manipulação e análise, por sua vez, são os mais importantes do ponto de vista dos usuários do sistema. Eles estão relacionados com as rotinas de manipulação dos dados do objetos espaciais que realizam funções analíticas e funções de processamento. Mesmo oferecendo as mais variadas funções analíticas e de processamento, o ambiente SIG deve ser modular para suportar novas funções, uma vez que os usuários podem desejar realizar operações não existentes. Descrições mais completas sobre este assunto podem ser encontradas em [Dan90, SE90].

Finalizando, o processo de saída está relacionado com os produtos finais do processamento, através dos quais o SIG interage com os usuários. Exemplos são listagens, mapas, “displays” interativos de dados espaciais e convencionais e geração de arquivos de dados, entre outros. Como um dos principais objetivos do SIG é manipular dados espaciais, especial atenção deve ser dada à visualização destes. Os usuários devem ser capazes tanto de visualizar a geometria dos dados finais gerados quanto de conhecer o relacionamento entre os mesmos (vizinhança). Desta forma, algumas operações interessantes a serem oferecidas são rotação e translação de figuras, “zooming”<sup>1</sup> e “shifting”<sup>2</sup>. Um estudo deste tópico pode ser encontrado em [SE90].

### 2.3.3 Formatos de Dados

SIGs utilizam dois tipos básicos de formatos de dados: **vetorial** e **varredura**. O primeiro tipo é referente a representação utilizando pontos, linhas e polígonos. O segundo armazena os dados sob a forma de matrizes, ou grades de valores. Em outras palavras, no primeiro caso, a unidade fundamental de representação é a linha, enquanto que no segundo caso a unidade fundamental de representação é a célula.

O formato varredura considera que o principal componente para a representação de um objeto é a sua posição. Desta forma, ele é conhecido como baseado em posição, uma vez

---

<sup>1</sup> O usuário é capaz de alterar a escala de representação de um determinado mapa ou de uma região deste, tornando-o maior ou menor.

<sup>2</sup> O usuário é capaz de deslocar o mapa para a direita ou esquerda, para cima ou para baixo.

que os objetos são armazenados de acordo com as suas posições. Neste formato, cada posição é caracterizada por possuir um conjunto de propriedades do objeto que está representando. Já o formato vetorial considera que o principal componente para a representação de um objeto é sua característica geográfica/cartográfica. Desta forma, ele é conhecido como baseado em característica, uma vez que os objetos são armazenados relativamente às suas características geográficas/cartográficas. Neste formato, as demais propriedades dos objetos, incluindo posição, são armazenadas como propriedades adicionais.

No formato varredura, cada célula da matriz contém apenas um único valor temático. Caso dois ou mais objetos ocupem a mesma célula, apenas o valor correspondente ao objeto mais abrangente é representado. Desta forma, o tamanho de cada célula deve ser cuidadosamente escolhido e deve ser menor ou igual ao tamanho do menor objeto a ser representado. Além disso, neste formato, os limites dos objetos não são bem definidos, contraditoriamente ao formato vetorial, no qual limites de regiões são armazenados precisamente.

Devido às diferentes percepções existentes do mundo real e devido às diferentes necessidades de consulta dos usuários dos sistemas, é interessante que uma aplicação SIG utilize tanto o formato varredura quanto o vetorial para a representação dos dados espaciais. Além disso, a aplicação também deve oferecer rotinas eficientes para a conversão de um formato para outro. Uma importante consideração a ser feita é que ambas as representações podem implementar modelos do mundo real equivalentes, mas não idênticos.

Consultas dos usuários são geralmente divididas em dois grupos [HB92]. O primeiro grupo de consultas requisita a posição de alguma classe de objetos dentro de uma determinada janela espacial. Já o segundo grupo de consultas requisita a identificação de objetos encontrados dentro de uma determinada janela espacial e pertencente a alguma subclasse de objetos. O formato de dados varredura responde melhor à primeira classe de consultas, enquanto que o vetorial à segunda. Desta forma, ambos modelos são necessários para um ambiente SIG para que consultas baseadas tanto em posição quanto em característica sejam manipuladas eficientemente.

Uma variedade de estruturas que implementam ambos os formatos (varredura e vetorial) têm sido implementadas, cada uma com diferentes desempenhos e características de armazenamento. Para o formato varredura, as principais estruturas existentes são: “array” regular (bidimensional, tridimensional, 8-dimensional), “array” irregular e estrutura hierárquica (como exemplo “quadtree”). Já para o formato vetorial, as principais estruturas existentes são: “spaghetti” (não estruturado), hierárquico, topológico direcional (como exemplo, arquivos DIME), topológico complexo (como exemplo, POLYVRT) e baseado em objetos. Adicionalmente, também podem existir formatos híbridos, os quais combinam características dos formatos varredura e vetorial. Uma implementação para este formato é o modelo “vaster”. Informações mais detalhadas sobre este tópico, incluindo

características de cada implementação, além de suas vantagens e desvantagens podem ser encontradas em [Peu90, SE90, HB92, RM92].

Os formatos acima descritos e as respectivas estruturas que os implementam são utilizados em um ambiente SIG para armazenar os dados espaciais dos objetos espaciais que fazem parte do sistema. Porém, estas estruturas são ineficientes para o armazenamento dos dados convencionais destes objetos. Aplicações SIGs sofisticadas devem garantir complexo inter-relacionamento entre dados espaciais e convencionais dos objetos espaciais com os quais estão trabalhando. Desta forma, cinco principais projetos de banco de dados, baseados nos modelos de dados já existentes, têm sido utilizados para organizar os dados convencionais em um SIG [RM92]. Estes modelos são: de arquivo “flat” (sem estrutura), hierárquico, de rede, relacional e orientado a objetos. Além das vantagens apresentadas por cada um dos modelos acima citados, a utilização de um destes modelos para o armazenamento de dados convencionais de uma aplicação SIG implica em menor redundância, uma vez que as estruturas que implementam os formatos varredura e vetorial podem apenas possuir apontadores para as informações em comum (como exemplo, para tabelas, no caso de banco de dados relacionais).

### 2.3.4 O SIG e o Planejamento Urbano

O planejamento urbano representa uma das principais áreas de utilização dos SIGs, uma vez que é responsável pelo controle e organização de cidades. O intenso crescimento populacional, geográfico e econômico das cidades exige um sistema capaz de prever e direcionar a expansão da malha urbana, além de gerenciar a utilização dos seus recursos públicos. Desta forma, a implantação do geoprocessamento no planejamento urbano permite que a população possua maior acesso aos serviços de saúde, alimentação, educação e infra-estrutura básica, entre outros.

A utilização do SIG no planejamento urbano apresenta diversas vantagens, entre as quais pode-se citar [Lem94, San94, TLS94]:

- a criação de um BD georeferenciado para todos os organismos da municipalidade, estabelecendo uma fonte integrada que relaciona todos os aspectos urbanos técnicos, sociais e econômicos. Desta forma, é garantida maior eficiência na comunicação entre os diversos setores da administração municipal.
- a manutenção automática da base cartográfica através de atualizações via levantamentos topográficos que são direcionados conforme os cadastros de loteamentos, alvarás de construção e outras fontes.
- o apoio à administração municipal no acompanhamento e avaliação do desenvolvimento urbano.

- a reprodução imediata das plantas cartográficas, cadastrais e de mapas temáticos a todos os usuários, com mesmo nível de atualização e confiabilidade.
- a obtenção de informações a nível de lote, quadra, bairro ou de qualquer área desejada, de forma imediata e precisa.
- a automatização das tarefas que se apóiam nas bases cartográfica e cadastral, possibilitando maior rapidez e confiabilidade das informações obtidas.
- a redução dos gastos na produção e manutenção das informações vinculadas à representação física do espaço urbano.

A base de todos os procedimentos da implantação do geoprocessamento é o mapeamento urbano. Este mapeamento urbano deve ser entendido como uma camada sobre a qual se apresentam os diversos elementos que condicionam a ocupação do território, onde se desenvolvem as atividades humanas e ocorrem os relacionamentos de ordem social e até mesmo ambiental.

O mapeamento urbano constitui o conjunto de informações gráficas e alfanuméricas referentes à base cartográfica das plantas cadastrais [Tel92]. Em outras palavras, as informações obtidas de plantas cadastrais imobiliárias, tais como o posicionamento das propriedades, dos logradouros e suas respectivas codificações, são armazenadas com base na cartografia, originando uma base digital georeferenciada.

Atuando como uma camada subjacente, é sobre o mapeamento urbano que são desenvolvidas as mais variadas aplicações. Exemplos são as aplicações de gerenciamento da rede de telefonia, como é o caso da Telebrás [Mag93], as aplicações de gerenciamento da rede de energia elétrica, como é o caso da CELPE [CSV94] e da Eletropaulo [Web94a] e as aplicações de gerenciamento da rede de distribuição de água, como é o caso da Sanepar [Sie94].

De maneira geral, os principais elementos que compõem o mapeamento urbano são os logradouros, os lotes, as quadras e os setores. Aplicações mais completas também consideram os elementos de hidrografia, as edificações de destaque, a altimetria, os elementos de utilidade urbana e os marcos geodésicos, entre outros. No entanto, aplicações SIG em geral envolvem grande volume de investimento financeiro e, em adição, a manipulação da informação espacial é mais complexa, mais demorada e mais sujeita a erros do que a manipulação de informações alfanuméricas. Desta forma, deve-se escolher um conjunto mínimo de informações a serem armazenadas de modo que elas atendam satisfatoriamente aos requisitos básicos da aplicação em questão.



# Capítulo 3

## Arquitetura Proposta

### 3.1 Descrição do Modelo

A tese aqui desenvolvida visa a integração de bancos de dados orientados a objetos e relacionais. Antes da definição da arquitetura utilizada para esta integração, vale ressaltar alguns pontos que foram levados em consideração e que contribuíram para a proposta desta arquitetura. Estes pontos são:

- a abordagem utilizada para o desenvolvimento do esquema conceitual.
- a forma de mapeamento entre visões dos usuários e esquemas locais.
- o modelo de dados global.

Abaixo são descritos cada um destes pontos, justificando as diversas decisões tomadas. Em seguida, a arquitetura proposta é explanada.

#### 3.1.1 Escolha da Abordagem Utilizada para o Desenvolvimento do Esquema Conceitual

Existem duas abordagens utilizadas para o desenvolvimento do esquema conceitual durante o processo de integração de esquemas [Tho+90, Oli93]:

- **abordagem de esquema global.** Esta abordagem consiste na integração de diversos esquemas locais (correspondentes aos SGBDs componentes) em um único esquema global. Em outras palavras, os esquemas locais desenvolvidos independentemente

são integrados, suas diferenças semânticas e estruturais são resolvidas e um esquema global final é oferecido ao usuário. Este esquema global é, portanto, independente da heterogeneidade dos SGBDs componentes. As visões dos diferentes grupos de usuários e suas respectivas aplicações podem ser definidas sobre o esquema global. A vantagem da utilização desta abordagem está relacionada ao fato de que os usuários não precisam se preocupar com a localização dos dados e com a heterogeneidade semântica dos BDs locais. A desvantagem é que o processo de integração de esquemas é uma tarefa difícil de ser realizada.

- **abordagem de esquema federado.** Nesta abordagem não há a criação de um esquema global: apenas são oferecidos aos usuários diversos esquemas locais, os quais descrevem os dados dos diversos SGBDs componentes que podem ser compartilhados. Desta forma, a tarefa de integrar os dados localizados nos múltiplos BDs é deixada a cargo dos usuários, os quais devem conhecer tanto as diferenças de representação dos dados quanto as suas localizações. As vantagens apresentadas por esta abordagem são que nenhum esforço é exigido do administrador do banco de dados (ABD) para resolver a heterogeneidade semântica existente entre os SGBDs componentes e que, além disso, não é necessário que o ABD antecipe as necessidades dos usuários, uma vez que estes conseguem acessar os dados desejados quando necessário. Por outro lado, esta abordagem possui algumas desvantagens, tais como a não transparência de localização dos dados dos diversos BDs e a dificuldade de realização das operações de atualização.

Dentre as abordagens acima descritas, a escolhida para o desenvolvimento deste trabalho foi a de **esquema global**. Esta abordagem oferece maior facilidade aos usuários do sistema global, uma vez que garante tanto a transparência de localização quanto a ausência de heterogeneidade semântica. Desta forma, os usuários podem acessar os dados dos diversos SGBDs componentes de uma forma transparente.

### 3.1.2 Escolha da Forma de Mapeamento Entre Visões dos Usuários e Esquemas Locais

As formas de mapeamento entre visões dos usuários e esquemas locais são caracterizadas como transformação de modelos de dados, tradução de linguagem de consulta e acesso cruzado de modelo [HK89]. A transformação de modelo de dados é a transformação automática de dados em um BD com um determinado modelo de dados (modelo de dados fonte) em dados em um BD em outro modelo de dados (modelo de dados destino). A tradução de linguagens é a tradução automática de uma transação em uma linguagem de manipulação de dados de alto nível para outra linguagem de alto nível. Finalmente, o acesso cruzado permite que os usuários acessem o BD utilizando uma linguagem diferente da que este possui. Desta forma, os usuários podem utilizar uma linguagem de consulta e um modelo de dados que lhes são familiares.

Existem quatro formas de mapeamento entre visões dos usuários e esquemas locais ([HK89, OM93]). Como o enfoque da integração desenvolvida neste trabalho é o nível de esquema conceitual, apenas será considerada a transformação de modelos.

- **mapeamento de um único modelo para outro modelo.** Neste mapeamento ocorre uma transformação entre cada par de modelos de dados. Para o mapeamento ser mais completo, ele deve ser realizado em ambos os sentidos. Em outras palavras, um primeiro SGBD deve suportar aplicações baseadas no modelo de dados de um segundo SGBD e vice versa. A desvantagem apresentada por esta abordagem está relacionada ao fato de que o número de mapeamentos aumenta consideravelmente à medida que cresce o número de modelos diferentes no ambiente heterogêneo.
- **mapeamento de um único modelo para vários outros modelos.** Neste mapeamento é oferecido um modelo de dados global, para o qual todos os esquemas dos SGBDs componentes são traduzidos. Desta forma, são realizadas transformações de modelos de dados de um único modelo global para múltiplos modelos alvo dos respectivos SGBDs existentes.
- **mapeamento de vários modelos para um único modelo.** Neste mapeamento todos os dados do ambiente heterogêneo são armazenados utilizando-se um único modelo de dados de mais baixo nível, sendo oferecidos aos usuários esquemas virtuais nos diferentes modelos de dados dos SGBDs componentes. A desvantagem apresentada por esta forma de mapeamento está relacionada ao fato de que todos os dados do ambiente heterogêneo devem ser armazenados utilizando-se o mesmo modelo de dados, o que implica na necessidade de uma migração dos dados, a qual nem sempre é desejável ou possível.
- **mapeamento de vários modelos para vários outros modelos.** Neste mapeamento é oferecido um modelo de dados intermediário, e ocorre a combinação de dois estágios de mapeamento. No primeiro deles, BDs em um modelo de dados fonte são mapeados no modelo de dados intermediário. No segundo estágio, o modelo de dados intermediário é transformado para os modelos de dados alvo. As visões finais dos usuários podem ser representadas utilizando-se qualquer um dos modelos de dados dos SGBDs componentes. A desvantagem apresentada por esta forma de mapeamento está relacionada ao número de mapeamentos necessários.

Dentre estas formas de mapeamento descritas, a escolhida para o desenvolvimento deste trabalho foi a segunda (**mapeamento de um único modelo para vários outros modelos**). Esta forma de mapeamento oferece facilidade de incorporação de novos SGBDs ao ambiente global e garantia de acesso uniforme a todos os sistemas. Por outro lado, para utilizar o ambiente heterogêneo, qualquer usuário deveria aprender o novo modelo de dados global. Esta desvantagem não apresenta problemas, uma vez que pode ser oferecida uma interface gráfica aos usuários, através da qual consultas podem ser realizadas de maneira simples e rápida.

### 3.1.3 Escolha do Modelo de Dados Global

A identificação de um modelo de dados global poderoso e flexível é fundamental para o processo de integração. [OM93] destaca alguns requisitos básicos que tal modelo deve apresentar:

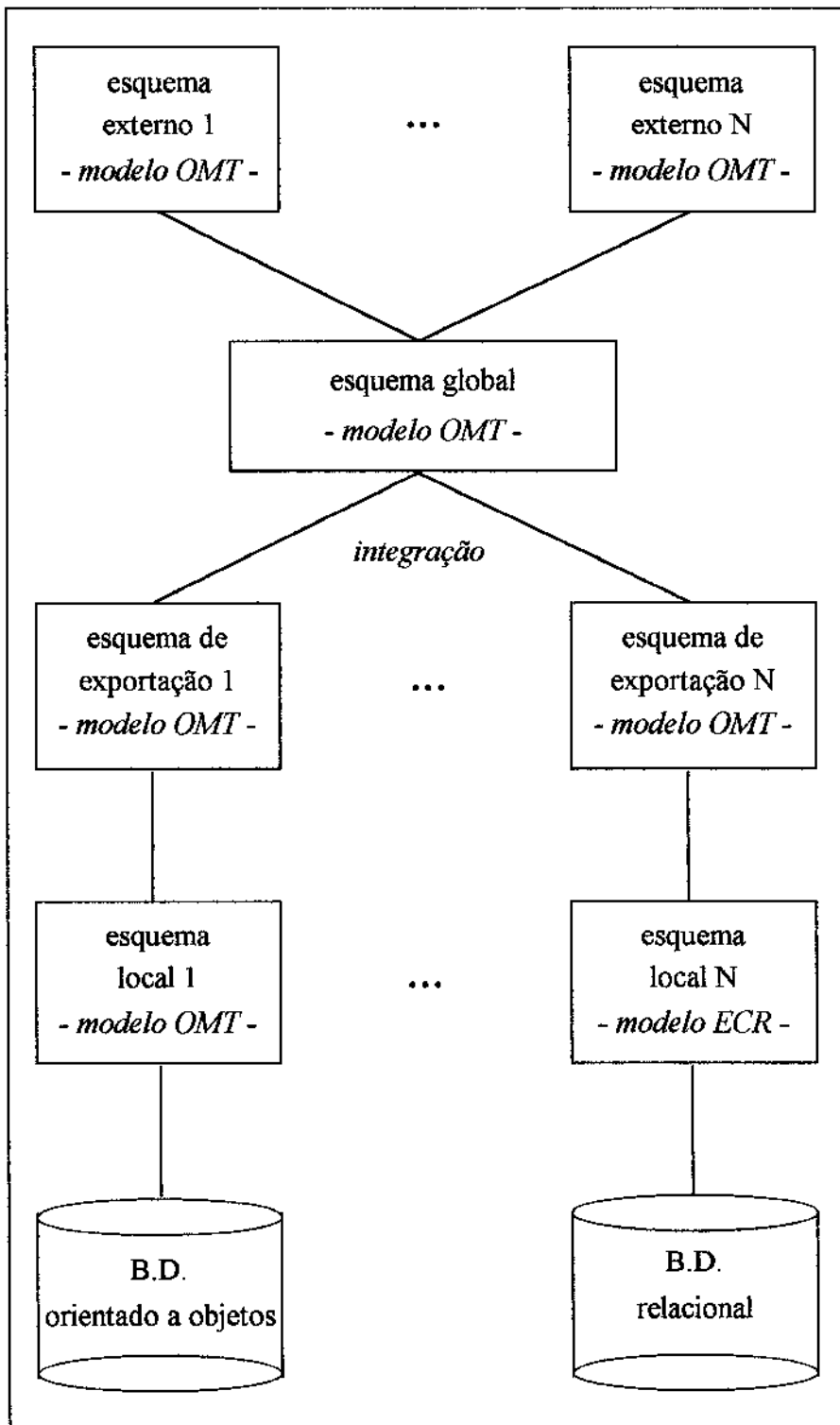
- facilidade para representar informações contidas em outros modelos, de forma simples e clara.
- capacidade de suportar diferentes níveis de abstração usados nos projetos dos diversos SGBDs componentes.
- possibilidade de acrescentar informações relevantes não contidas nos esquemas dos SGBDs componentes como, por exemplo, restrições de integridade implícitas ou mantidas pelas aplicações.
- disponibilidade de uma linguagem de definição e manipulação de dados abrangente.

De acordo com os requisitos acima citados, o modelo de dados global escolhido para o desenvolvimento deste trabalho foi o **modelo orientado a objetos**. Os modelos orientados a objetos são modelos expressivos e flexíveis, e utilizam conceitos claros e naturais. Estes modelos foram projetados com a finalidade de representar estruturas complexas de maneira coerente e uniforme [SL94]. Para um SIG, a criação de modelos complexos é uma necessidade, uma vez que dados espaciais e não espaciais devem ser manipulados conjuntamente. Além disso, modelos orientados a objetos permitem que as características espaciais e não espaciais de uma classe sejam herdadas por outras classes [RM92].

A modelagem orientada a objetos aumenta a integração entre os dados e a aplicação, permitindo que os dados e as operações sejam projetadas ao mesmo tempo. Este fato é importante para aplicações geográficas, uma vez que as operações encapsuladas pelas classes devem ser aplicadas aos seus atributos espaciais e não espaciais. Adicionalmente, relações topológicas também podem ser armazenadas para cada instância da classe.

O modelo de dados utilizado neste trabalho é o modelo do sistema  $O_2$  [Deu+92]. Com a finalidade de facilitar a visualização e a compreensão do esquema  $O_2$ , é utilizada a metodologia OMT para o projeto lógico deste BD orientado a objetos [Rum+91]. Uma vantagem adicional do OMT é a sua ampla aceitação para projetos de aplicações no contexto de engenharia de software. Desta forma, a integração será realizada considerando-se esta metodologia.

A figura 3.1 representa a arquitetura proposta, de acordo com as decisões acima tomadas.



**Figura 3.1** Arquitetura proposta.

Os SGBDs orientado a objetos e relacional considerados são representados, respectivamente, por esquemas projetados a partir dos modelos OMT [Rum+91] e ECR [EWH85]. Como citado anteriormente, a utilização de modelos gráficos para a representação dos BDs permite uma melhor visualização e compreensão dos seus esquemas. Neste trabalho, é necessário fazer diferença entre esquema local e esquema de exportação. Cada esquema local representa o esquema de cada SGBD que participa do ambiente global, e é expresso no modelo de dados deste componente. Por outro lado, o esquema de exportação representa o esquema local expresso no modelo de dados lógico global (no caso em questão, o modelo OMT).

Para tanto, devem ser oferecidas primitivas para o mapeamento (processo de tradução) entre os conceitos dos modelos ECR e OMT. Estes esquemas de exportação são, finalmente, integrados em um único esquema global, o qual utiliza o modelo OMT. As visões dos grupos de usuários e das aplicações devem ser derivadas deste esquema global. Estas visões são representadas pelos esquemas externos.

De acordo com esta arquitetura, o processo de integração dos BDs orientado a objetos e relacional inicia com a definição dos esquemas. Em paralelo, os esquemas utilizando o modelo ECR são traduzidos em esquemas equivalentes utilizando o modelo OMT. Finalmente, os diferentes esquemas OMT são integrados em um esquema OMT global, utilizado pelos usuários para acessar o BD heterogêneo.

## 3.2 A Metodologia OMT

A metodologia OMT (Object Modeling Technique) [Rum+91] é uma metodologia indicada para o desenvolvimento de sistemas baseados no paradigma orientação a objetos. Três tipos de modelos inter-relacionados são utilizados nesta metodologia para a descrição de um sistema: modelo de objetos, modelo funcional e modelo dinâmico. Cada modelo oferece uma notação gráfica específica para a representação dos conceitos orientados a objetos.

O **modelo de objetos** descreve a estrutura estática dos objetos no sistema, mostrando seus atributos, suas identidades, seus relacionamentos com outros objetos e as operações que caracterizam cada classe de objetos. Este modelo é o principal dos três modelos desta metodologia, uma vez que descreve os objetos que compõem o sistema, ou seja, descreve o **que** compõe o sistema, ao invés de quando ou como. A representação gráfica utilizada por este modelo é chamada **diagrama de objetos**. Um diagrama de objetos é um grafo cujos nós são classes de objetos e cujos arcos são relacionamentos entre classes.

O **modelo dinâmico** descreve os aspectos do sistema que variam com o tempo. Este modelo tem por finalidade especificar e implementar aspectos de controle do sistema, descrevendo as seqüências de operações que ocorrem em resposta a estímulos externos,

sem levar em consideração o que as operações fazem, sobre quem elas operam ou como são implementadas. A representação gráfica utilizada por este modelo é chamada **diagrama de estados**. Um diagrama de estados é um grafo cujos nós são estados e cujos arcos são transições entre estados causadas por eventos. Estados representam valores de objetos, enquanto que eventos representam estímulos externos.

Finalmente, o **modelo funcional** descreve as transformações dos valores dos dados do sistema. Este modelo tem por finalidade mostrar como valores de saída em uma computação são derivados de valores de entrada, sem se preocupar como ou quando a computação é realizada. A representação gráfica utilizada por este modelo é chamada **diagrama de fluxo de dados**. Um diagrama de fluxo de dados representa uma computação, e é um grafo cujos nós são processos e cujos arcos são fluxos de dados.

Cada modelo descreve um aspecto do sistema, mas contém referências aos outros modelos. Desta forma, os três modelos são partes ortogonais inter-relacionadas da descrição de um sistema completo.

A dissertação visa a integração de bancos de dados heterogêneos que utilizam os paradigmas relacional e orientado a objetos, em particular, sistemas desenvolvidos utilizando-se o modelo ECR (Entidade Categoria Relacionamento) e a metodologia OMT, respectivamente. O modelo ECR é um modelo semântico utilizado para descrever os dados de uma aplicação. Desta forma, para se realizar a integração, apenas o modelo de objetos do OMT será considerado.

### 3.2.1 O Modelo de Objetos

A modelagem de objetos OMT combina conceitos do paradigma orientação a objetos (tais como classe e herança) com conceitos de modelagem de dados (entidades e associações). Desta forma, pode-se considerar que os conceitos OMT empregados são similares aos da modelagem entidade-relacionamento estendida ([Rum+91, PB94]).

#### 3.2.1.1 Objeto, Classe, Atributo e Operação

O propósito da modelagem de objetos é descrever **objetos**, que são abstrações utilizadas para a modelagem do mundo real, sendo distinguíveis e possuindo identidade. O termo objeto está relacionado ao fato de que os objetos são distinguíveis por sua própria existência, e não pelas propriedades descritivas que possam ter. Desta forma, é possível a existência de diferentes objetos com os mesmos valores de atributos e relacionamentos. Objetos possuem identidade, significando que suas existências independem dos valores de seus atributos e relacionamentos.

Uma **classe** descreve um grupo de objetos que possuem os mesmos componentes (atributos), comportamento comum (operações), mesmos relacionamentos com outros

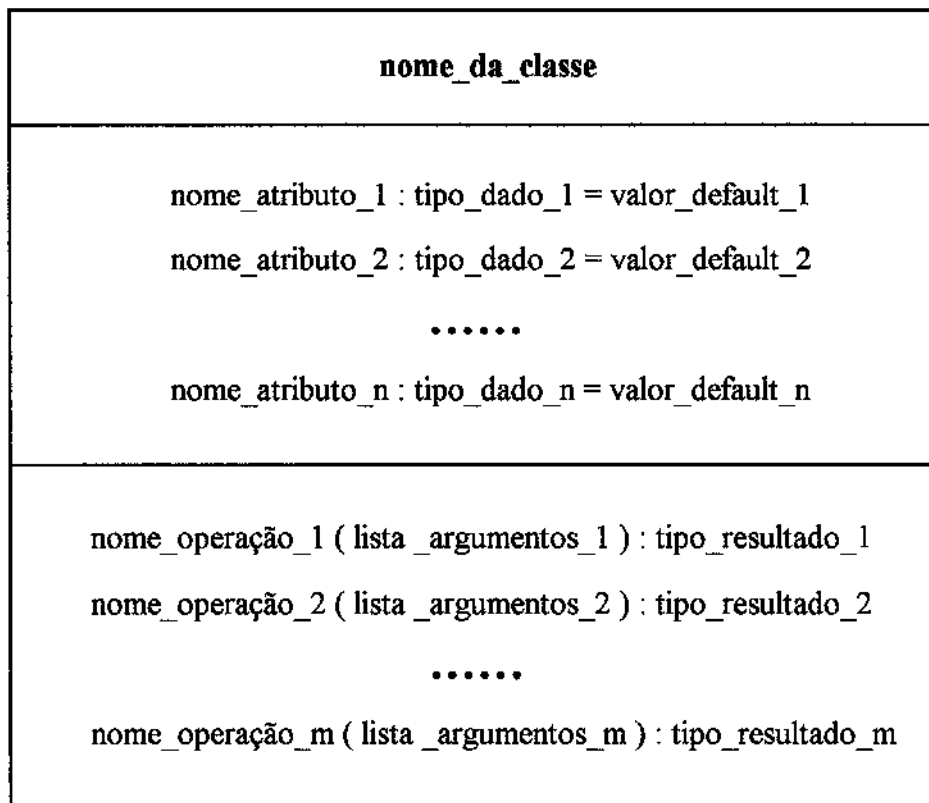
objetos do sistema e finalmente que expressam a mesma semântica. A modelagem OMT suporta tanto classes concretas quanto abstratas. Classes concretas são aquelas que possuem instâncias, ao passo que as classes abstratas não possuem instâncias diretas. A classe de um objeto é uma propriedade implícita deste objeto.

Os valores de dados armazenados pelos objetos recebem o nome de **atributos**. Um **atributo** é uma propriedade dos objetos de uma classe. Cada nome de atributo é único dentro de uma classe, o que não é verdadeiro quando todas as classes são consideradas. A modelagem OMT suporta a especificação de detalhes adicionais e opcionais para os atributos, tais como tipo ou valores “default”.

Uma **operação** é uma função ou transformação que pode ser aplicada a objetos de uma classe por outros objetos. Um **método** é a implementação de uma operação para uma classe. Uma operação possui um objeto “alvo” como argumento implícito, e pode possuir, em adição, uma lista de argumentos e um valor de retorno. A metodologia OMT suporta o conceito de operação polimórfica, permitindo que a mesma operação seja aplicada a várias classes diferentes. Desta forma, a mesma operação pode possuir diferentes formas em diferentes classes, implicando que o comportamento da operação depende da classe de seu objeto alvo. No entanto, algumas restrições são impostas. A primeira delas está relacionada ao fato de que todos os métodos que implementam as operações polimórficas devem realizar, logicamente, a mesma tarefa (embora possuam diferentes implementações!). Outra restrição está relacionada ao fato de que todos os métodos devem possuir a mesma assinatura, ou seja, o mesmo número e tipo dos argumentos e o mesmo tipo de resultado. Todos os objetos de uma classe compartilham as mesmas operações.

A notação utilizada pela metodologia OMT para a representação das classes é um retângulo com o nome da classe em negrito. Este retângulo é dividido em até três regiões. A primeira região contém o nome da classe. A segunda região contém uma lista de atributos. Cada nome de atributo pode ser seguido por detalhes opcionais, tais como tipo e valor “default”. O tipo é precedido por dois pontos (:), enquanto que o valor “default” é precedido por um sinal de igual (=). Finalmente, a terceira região contém uma lista de operações da classe. Cada nome de operação pode ser seguido por detalhes opcionais, tais como lista de argumentos e tipo resultante. A lista de argumentos é escrita entre parênteses logo após o nome da operação e os argumentos são separados por vírgulas. O nome e o tipo de cada argumento devem ser especificados. O tipo resultante não deve ser omitido caso exista, uma vez que é importante distinguir operações que retornam valores das operações que não retornam. Caso uma operação não possua argumentos, isto deve ser especificado através da declaração de uma lista de argumentos vazia entre parênteses. Tanto os atributos quanto as operações podem não ser especificados durante a modelagem de objetos do sistema. A figura 3.2 sumariza a notação gráfica para a representação das classes.





**Figura 3.2** Representação gráfica da notação das classes.

### 3.2.1.2 Associação, Multiplicidade, e Agregação e Qualificação

Os relacionamentos ou conexões existentes entre classes são modelados através de associações. Uma **associação** descreve um grupo de ligações com estrutura e semântica comuns. Uma **ligação** é uma conexão conceitual ou física entre instâncias de objetos. Matematicamente, uma ligação é definida como uma tupla, que é uma lista ordenada de instâncias de objetos. As associações podem ser binárias (relacionar duas classes), ternárias (relacionar três classes) ou de ordem superior. A maioria das associações é, entretanto, binária.

A **multiplicidade** restringe o número de objetos que podem ser associados a outros objetos, especificando quantas instâncias de uma classe podem ser relacionadas a uma única instância da classe associada. A multiplicidade é freqüentemente descrita como sendo “um” ou “muitos”. Geralmente o valor da multiplicidade é um intervalo simples, porém ela pode assumir valores não contínuos. Um atributo especial, chamado **qualificador**, pode ser especificado com a finalidade de reduzir a multiplicidade efetiva de uma associação. O qualificador geralmente é utilizado para reduzir a multiplicidade das associações de “muitos” para “um”. A utilização do qualificador garante maior semântica à modelagem, uma vez que detalha a descrição da associação.

A **agregação** é uma forma fortemente acoplada de associação que possui semântica adicional, tal como cláusula transitiva e propagação do valor do atributo. A agregação especifica o relacionamento “é parte de”, “é composto por”, no qual os objetos que representam os componentes de algo são associados ao objeto que representa este algo. A cláusula transitiva é a propriedade mais significativa da agregação, e especifica que se a classe A é parte da classe B e a classe B é parte da classe C, então a classe A é parte da classe C. A propriedade de propagação do valor do atributo está relacionada ao fato de que algumas propriedades da classe completa são propagadas para os seus componentes, provavelmente com algumas modificações locais. Uma última propriedade da agregação é a anti-simetria, que especifica que se uma classe A é parte da classe B, a classe B não é parte da classe A.

Com a finalidade de ressaltar as similaridades da agregação e da associação, cada relacionamento de agregação existente entre um agregado e seus componentes é tratado como uma agregação separada. Desta forma, é possível especificar também a multiplicidade de cada componente dentro da estrutura. Em termos semânticos, o agregado é um objeto estendido tratado como uma unidade em muitas operações, embora fisicamente ele seja composto por objetos menores. A agregação pode ser utilizada em um número arbitrário de níveis. Adicionalmente, os componentes em um agregado podem constar em muitos agregados.

A metodologia OMT exige a especificação das operações e dos atributos que são propagados da classe que representa o agregado para as classes componentes. O comportamento da propagação está relacionado a uma agregação e possui uma direção (classe agregada → classe componente), além do nome da operação ou do atributo sendo propagado.

Uma **associação qualificada** relaciona dois objetos e um qualificador. Seja, por exemplo, a associação existente entre *arquivos* e *diretórios*. De maneira geral, um arquivo pode pertencer a vários diretórios, porém, dentro de um diretório, cada arquivo é único, identificado pelo seu nome. Desta forma, um diretório e um nome de arquivo conduzem a um único arquivo. Neste caso, os objetos relacionados pertencem às classes *arquivo* e *diretório*, e o qualificador é o *nome do arquivo*. A sintaxe da qualificação também indica que cada nome de arquivo é único dentro do seu diretório.

Construtores adicionais utilizados na modelagem de associações incluem atributos de ligação e nomes de papéis. Um **atributo de ligação** é uma propriedade das ligações de uma associação, da mesma forma que um atributo é uma propriedade dos objetos em uma classe. Cada atributo da ligação tem um valor para cada ligação. Este construtor deve ser utilizado principalmente em associações “muitos-para-muitos”, nas quais os atributos são propriedades da ligação, e não podem ser acoplados a nenhum dos objetos. Atributos de ligação podem aparecer também tanto em associações “um-para-um” quanto em associações “um-para-muitos”. Muitas vezes, torna-se interessante modelar uma **associação como uma classe**, principalmente quando as ligações desta associação podem participar de uma associação com outros objetos ou quando estas estão sujeitas a

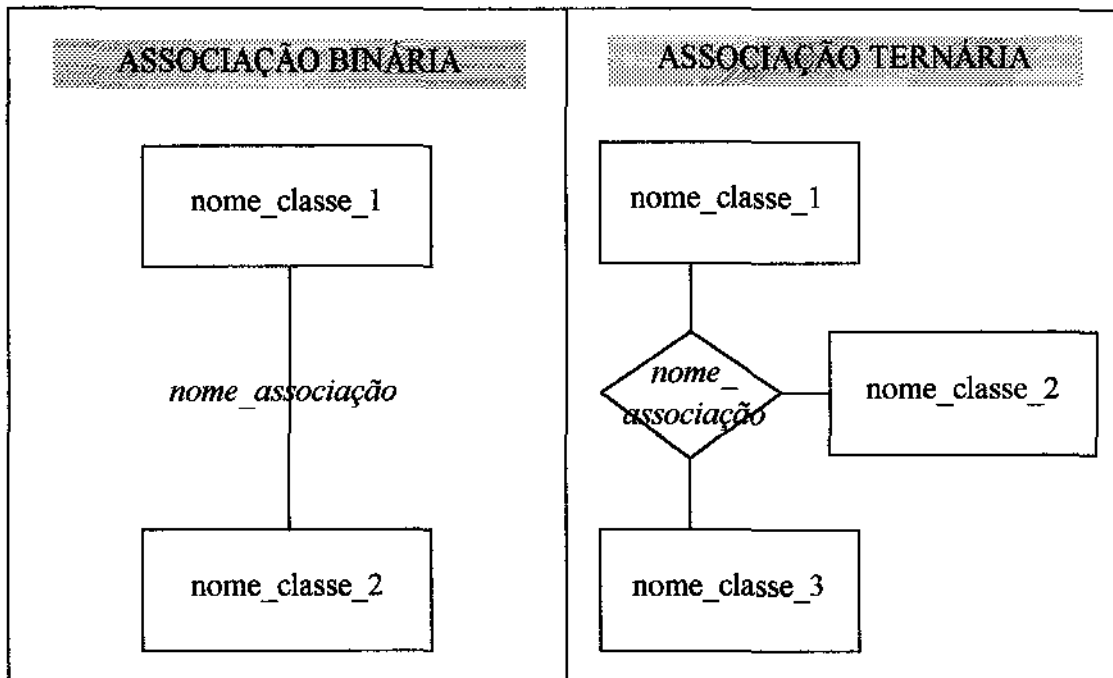
operações. Neste caso, cada ligação da associação torna-se uma instância da classe, e pode possuir, além dos seus atributos, nome e operações.

Um **papel** é uma extremidade de uma associação. Um **nome de papel** é um nome que identifica univocamente uma das pontas de uma associação. De maneira geral, nomes de papéis são necessários tanto para associações entre dois objetos da mesma classe quanto para distinguir entre duas ou mais associações existentes entre os mesmos pares de classes. Quando a associação entre classes é auto-explicativa, os nomes de papéis podem ser omitidos do diagrama de objetos.

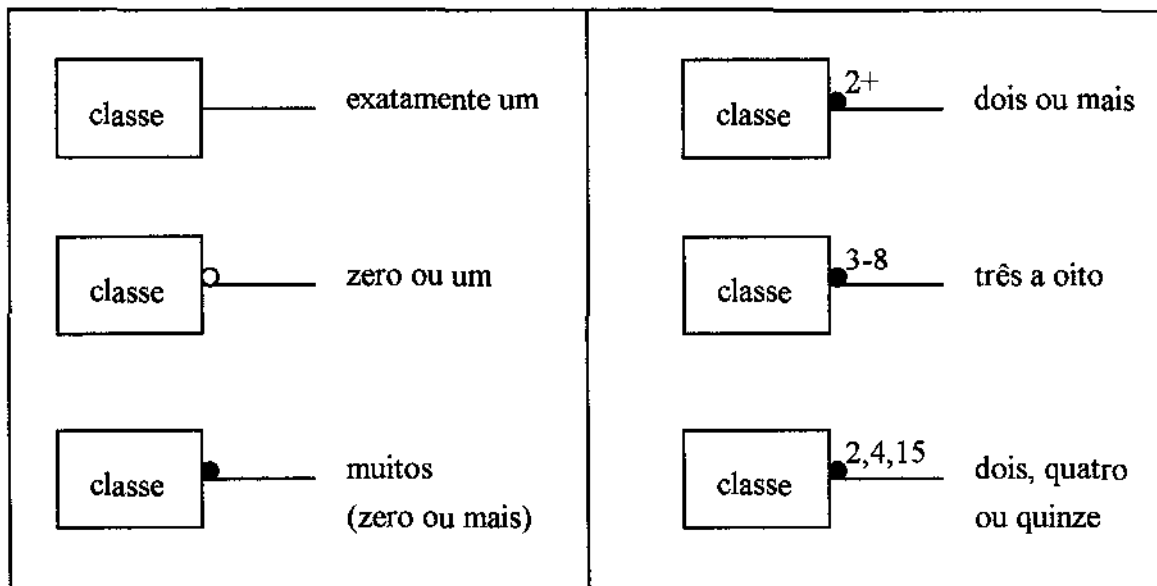
Em uma associação binária, o nome do papel é considerado um atributo derivado, uma vez que o seu valor é um conjunto de objetos inter-relacionados. Em outras palavras, cada papel de uma associação binária identifica um objeto ou conjunto de objetos associado a um objeto na outra extremidade. Já em associações ternárias ou de ordem superior, os nomes de papéis associados a cada uma das extremidades não representam atributos derivados das classes participantes, uma vez que estas associações não podem ser percorridas de uma extremidade à outra sem perda de semântica. Uma vez que os nomes de papéis servem para distinguir entre objetos diretamente conectados a um determinado objeto, todos os nomes de papéis devem ser únicos. Adicionalmente, o fato do nome de papel ser considerado um atributo derivado da classe implica que este deve ser único também dentro dela.

A notação gráfica utilizada pela metodologia OMT para a representação das associações binárias é uma linha entre classes, com o nome da associação em itálico. O nome da associação pode ser omitido se o par de classes tem apenas uma única associação cujo significado é óbvio. As associações ternárias ou de ordem superior são representadas por um losango com linhas conectando as classes associadas. O nome da associação é escrito próximo ao losango. A figura 3.3 mostra a notação gráfica para associações binárias e ternárias.

O diagrama de objetos representa a multiplicidade através de símbolos localizados no final das linhas que representam as associações. De maneira geral, a multiplicidade é especificada com números ou conjuntos de intervalos, tal como 1 (exatamente um), 2+ (dois ou mais), 3-8 (três a oito) e 2, 4, 15 (dois, quatro ou quinze). As multiplicidades mais frequentemente utilizadas são “exatamente um”, “zero ou um” e “muitos” (representando zero ou mais). Estas últimas multiplicidades, por serem mais utilizadas, recebem representações especiais. A multiplicidade “exatamente um” é representada por uma linha sem símbolos em seu final. Já a multiplicidade “zero ou um” é representada por um pequeno círculo (vazio) desenhado próximo à classe “zero ou um”. Finalmente, a multiplicidade “muitos” é representada por um pequeno círculo pintado (sólido) desenhado próximo à classe “muitos”. A figura 3.4 exemplifica estas representações.



**Figura 3.3** Associações binária e ternária.



**Figura 3.4** Multiplicidade de associações.

A agregação é representada de maneira semelhante à associação, diferindo desta pela existência de um pequeno losango desenhado próximo a uma das classes da associação (próximo à classe agregada), indicando qual é a classe “composta por” e qual é a classe “parte de”. Quando existem várias classes que são parte de uma classe mais geral, as linhas da associação podem ser combinadas originando uma árvore de agregação no diagrama de objetos. Este conceito é esboçado nas figuras 3.5 e 3.6. Caso haja propagação de uma ou mais operações ou atributos da classe agregada para as classes

componentes, estas operações e atributos devem ser explicitados com uma pequena seta direcionada e com os seus nomes próximos à associação existente.

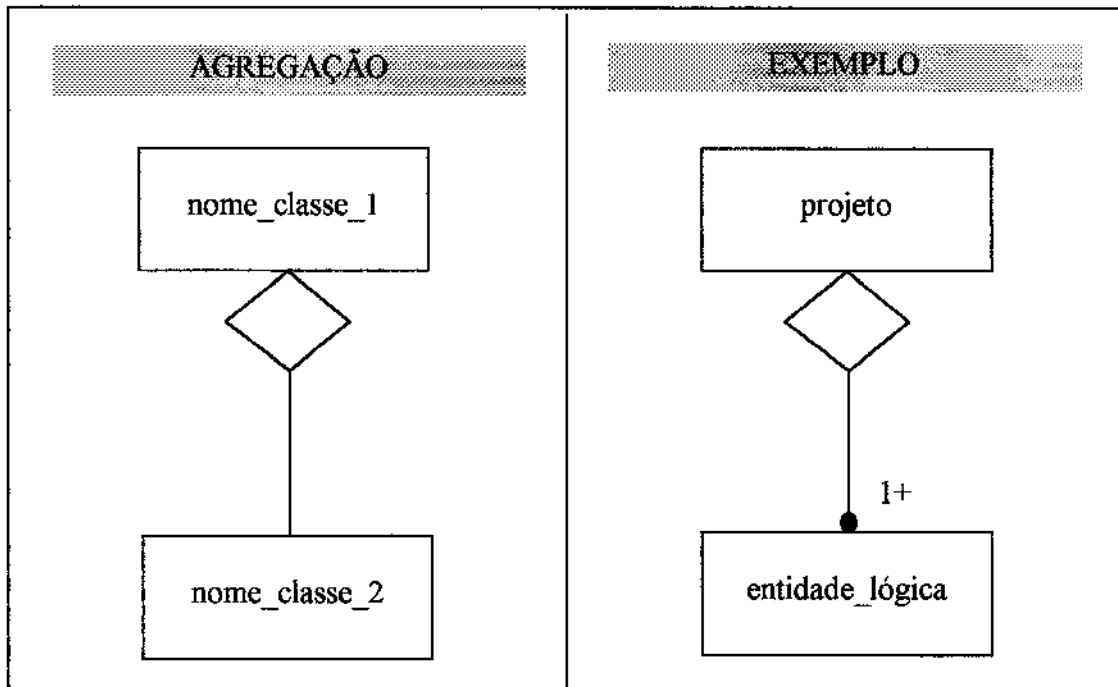


Figura 3.5 Agregação.

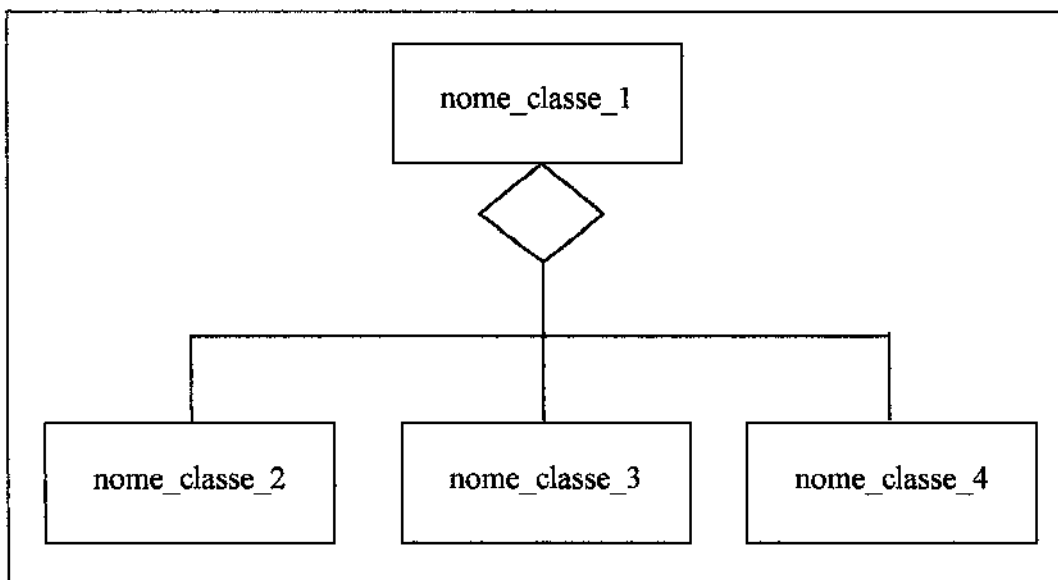


Figura 3.6 Árvore de agregação.

Um qualificador é desenhado como uma pequena caixa no final de uma linha que representa uma associação, próximo à classe que ele qualifica. A figura 3.7 esboça este conceito. Nesta figura também é apresentada, como exemplo, a modelagem da associação

qualificada entre *diretório* e *arquivo* previamente discutida. A notação gráfica utilizada pela metodologia OMT para representação de atributos de ligações tem por finalidade enfatizar as similaridades existentes entre estes e os atributos dos objetos. Ambos são representados na segunda região de um retângulo. Os atributos de ligações são acoplados às associações através de um laço. A figura 3.8 sumariza este construtor. Finalmente, os nomes dos papéis que as classes associadas desempenham em um relacionamento são representados próximos às respectivas classes, como ilustrado na figura 3.9.

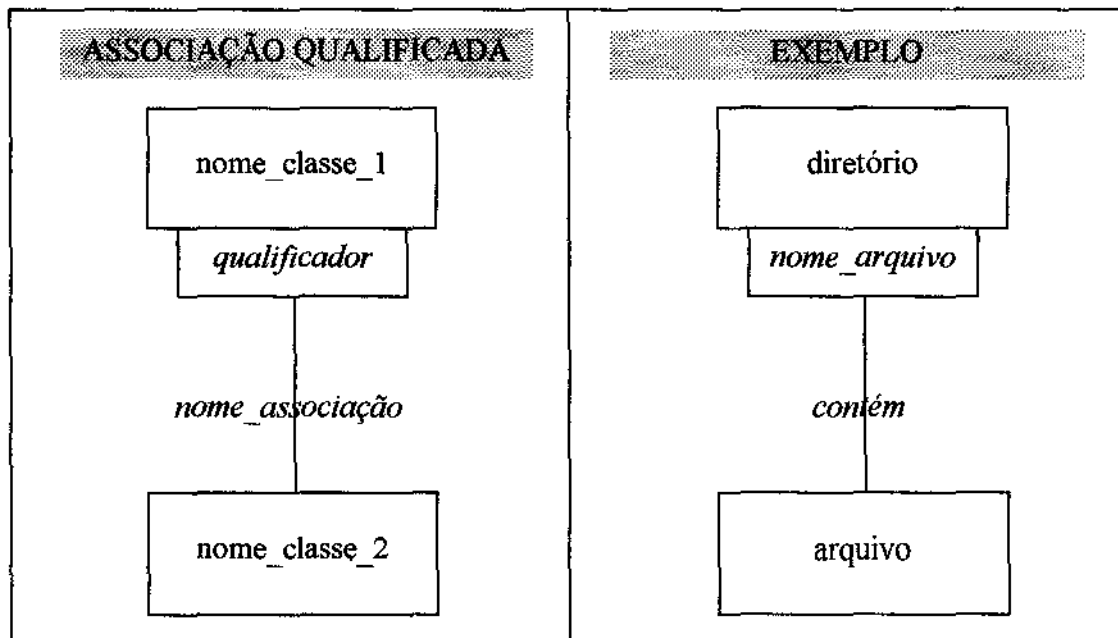


Figura 3.7 Associação qualificada.

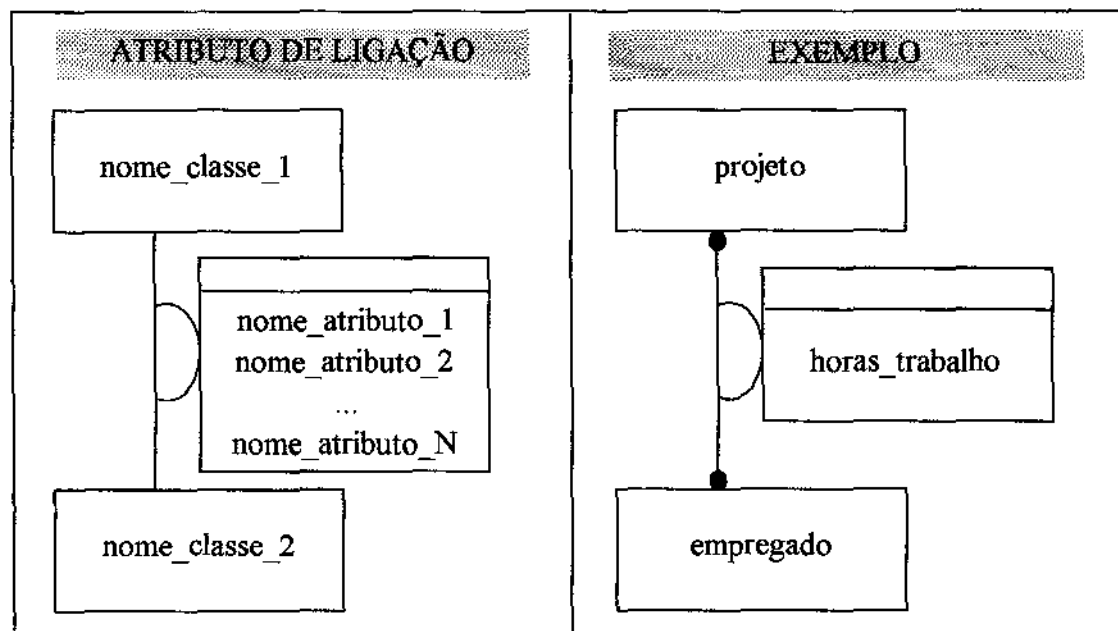
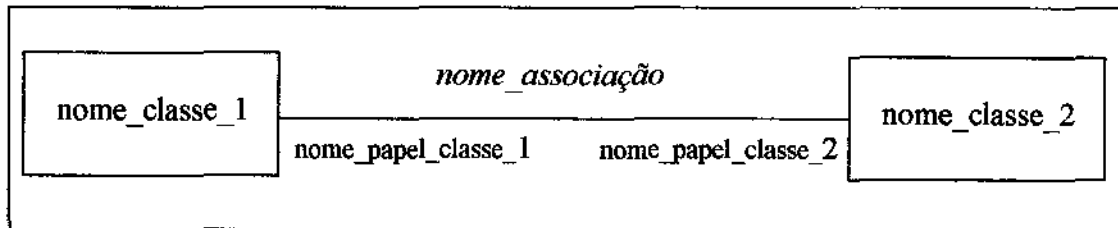


Figura 3.8 Atributo de ligação.



**Figura 3.9** Uso de nomes de papéis.

### 3.2.1.3 Generalização

Os modelos de objetos são construídos utilizando-se três construtores básicos: classes, associações e generalizações. A **generalização** é uma abstração que tem por finalidade compartilhar as semelhanças e preservar as diferenças existentes entre as classes. Em um relacionamento de generalização, uma classe mais genérica (chamada superclasse) é refinada em uma ou mais subclasses.

O mecanismo de compartilhamento de atributos e operações que ocorre em uma generalização é conhecido como **herança**. Todos os atributos e operações comuns a um grupo de subclasses são acoplados à superclasse, e compartilhados pelas subclasses. Cada subclasse herda todas as características de sua superclasse e, em adição, também acrescenta seus atributos e suas operações específicas.

O termo **especialização** refere-se aos aspectos da mesma idéia que o termo generalização, porém utilizando um ponto de vista diferente do mesmo relacionamento. A palavra generalização está relacionada ao fato de que uma superclasse generaliza as suas subclasses, isto é, uma superclasse é mais genérica que as subclasses, e possui os atributos e operações comuns. Já a palavra especialização está relacionada ao fato de que as subclasses refinam ou especializam a superclasse, isto é, herdam as características da superclasse e ainda adicionam novas características.

A generalização pode ser utilizada em um número arbitrário de níveis e, neste caso, são utilizados os conceitos ancestral e descendente. Uma instância de uma subclasse é simultaneamente uma instância de todas as suas classes ancestrais. Como cada instância de uma subclasse é também uma instância da superclasse, o relacionamento generalização também é conhecido como relacionamento “é-um”.

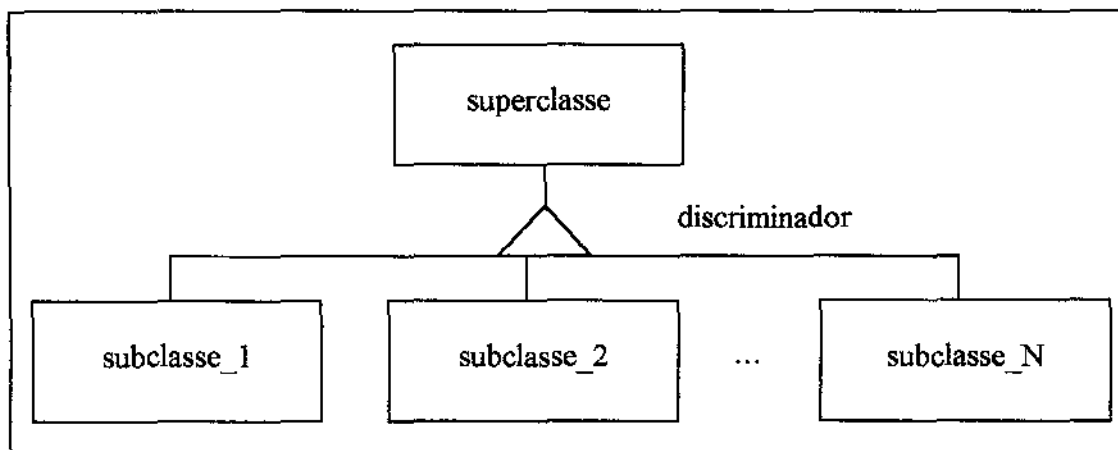
A correspondência entre objetos da superclasse e objetos das subclasses é especificada através da utilização de um discriminador opcional. Um **discriminador** é um atributo de um tipo enumerado que indica qual propriedade do objeto está sendo abstraída em um particular relacionamento de generalização. Somente uma propriedade por vez deve ser discriminada. Os valores do discriminador são herdados em correspondências “um-para-um” com as subclasses da generalização.

Foi visto anteriormente que uma subclasse, além de herdar os atributos e operações da superclasse, também pode conter atributos e operações específicas. No entanto, a subclasse pode sobrepor (“override”) uma característica da superclasse, definindo uma nova característica com o mesmo nome. Neste caso, a característica da subclasse refina e substitui a característica da superclasse. A metodologia OMT suporta a sobreposição de atributos e métodos de operações. No primeiro caso, deve ser preservado o tipo dos atributos, enquanto que os valores “default” podem ser sobrepostos. Já no segundo caso deve ser preservada a assinatura da operação, ou seja, o número e o tipo dos argumentos e o tipo de retorno da operação.

O conceito **herança múltipla** permite que uma classe herde atributos e comportamento de mais do que uma superclasse. Em outras palavras, informações de duas ou mais superclasses são “misturadas” em uma subclasse. A utilização deste conceito oferece vantagens e desvantagens. De um lado, a herança múltipla garante maior poder para a especialização das classes. Por outro lado, ocorre a perda da simplicidade conceitual da modelagem sendo realizada.

Alguns conflitos entre definições podem ocorrer quando uma classe herda características de duas ou mais superclasses. Um primeiro conflito refere-se à característica de uma classe ancestral que é encontrada mais do que uma vez em diferentes caminhos da hierarquia de generalização. Neste caso, como se trata da mesma característica, ela é herdada apenas uma única vez. Outro conflito refere-se à existência de definições paralelas, as quais criam ambigüidades (como exemplo, pode-se citar a definição de mesmas operações que são implementadas utilizando-se métodos distintos).

A notação OMT para a representação da hierarquia de generalização é um triângulo conectando uma superclasse a suas subclasses. Um triângulo pintado indica que as subclasses se sobrepõem. Por outro lado, um triângulo não pintado indica que as subclasses são disjuntas. Finalmente, caso o discriminador seja especificado, ele deve ser desenhado próximo ao triângulo da generalização (figura 3.10).



**Figura 3.10** Hierarquia de generalização com subclasses disjuntas.



### 3.2.1.4 Restrição

O modelo de objetos permite que certas restrições sejam expressas próximas aos elementos que estão restringindo. De maneira geral, estes elementos podem ser objetos, classes, atributos, ligações e associações. Algumas restrições são implicitamente incorporadas na própria estrutura do modelo de objetos, como o fato das subclasses serem mutuamente exclusivas em uma hierarquia de generalização disjunta.

Um tipo especial de restrição é a **ordenação**. A ordenação é uma parte inerente à associação e é utilizada para indicar que os elementos da extremidade “muitos” de uma associação têm uma classificação que deve ser preservada. A **instanciação** é um outro tipo de restrição, e estabelece um relacionamento entre uma classe e suas instâncias. A exibição explícita do relacionamento da instanciação é útil quando ambas as instâncias e as classes devem ser manipuladas como objetos.

A multiplicidade das associações é um exemplo de restrição que atua na ligação. Como visto anteriormente, a metodologia OMT oferece uma notação gráfica especial para a especificação da multiplicidade em associações binárias. No entanto, esta notação gráfica não é aconselhável para associações ternárias ou de ordem superior, uma vez que pode gerar ambigüidade na interpretação destas associações. Nestes casos, é indicada a utilização do conceito **chave candidata**. Uma chave candidata é um conjunto mínimo de atributos que identifica de maneira única um objeto ou ligação. A identificação de um objeto é sempre uma chave candidata para uma classe, enquanto que uma ou mais combinações de objetos inter-relacionados são chaves candidatas para associações.

Cada chave candidata restringe as instâncias de uma classe ou a multiplicidade de uma associação. Uma associação “muitos-para-muitos” possui como chave candidata os dois objetos relacionados. Já uma associação “um-para-muitos” possui como chave candidata apenas o objeto da extremidade “muitos”. Finalmente, uma associação “um-para-um” possui duas chaves candidatas, representadas por qualquer um dos objetos associados. A especificação de uma chave candidata pode ser realizada mesmo quando as classes que participam da associação são opcionais.

Na metodologia OMT as restrições devem ser representadas próximas aos elementos (objeto, classe, atributo, ligação ou associação) que restringem. Estas restrições devem ser especificadas entre chaves. As figuras 3.11 e 3.12 exemplificam, respectivamente, a utilização das restrições **ordenação** e **chave candidata**. Na figura 3.12, apenas as classes *classe\_1* e *classe\_2* especificam a multiplicidade “muitos” na associação ternária. Já a restrição **instanciação** é representada através de setas pontilhadas que interligam as instâncias às suas respectivas classes (figura 3.13).

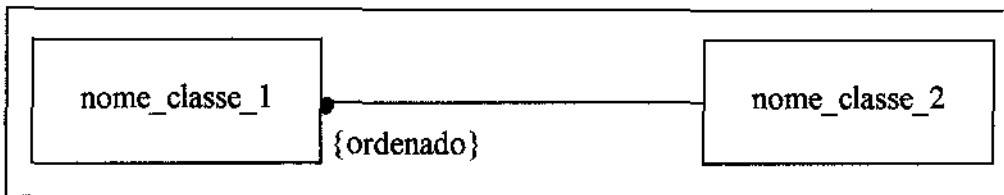


Figura 3.11 Restrição ordenação.

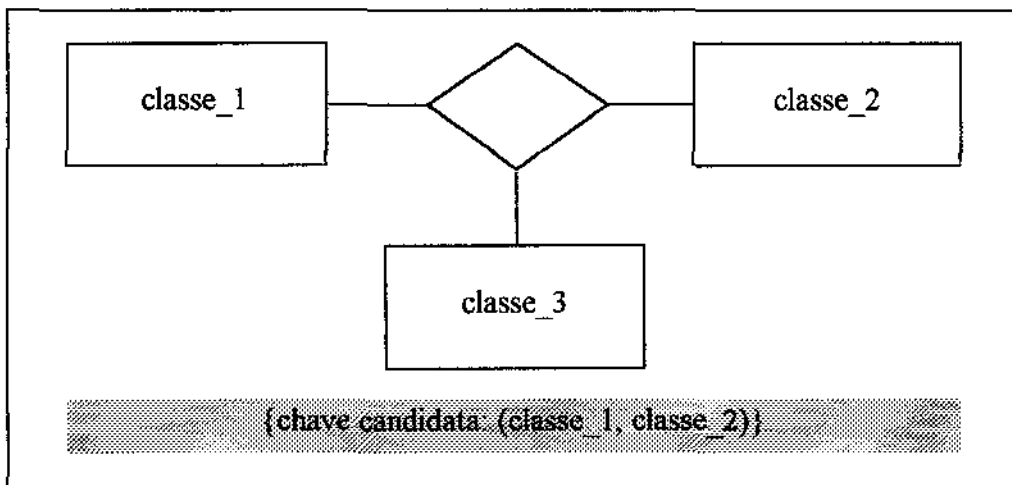


Figura 3.12 Restrição chave candidata.

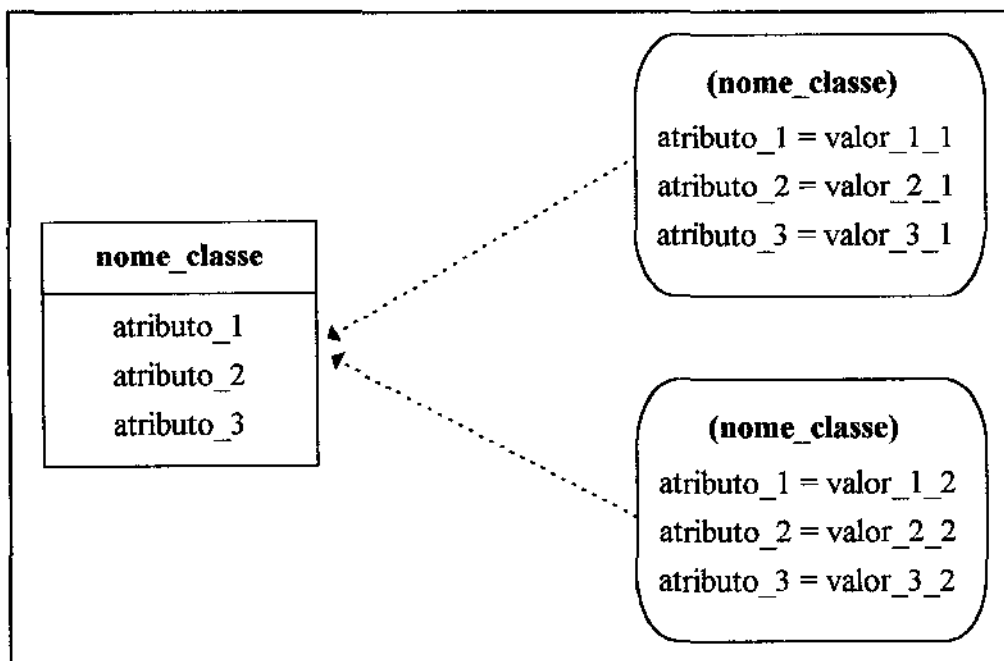


Figura 3.13 Restrição instanciação.

## 3.3 O Modelo ECR

O modelo ECR (Entity-Category-Relationship) [EWH85] é uma versão estendida do modelo entidade-relacionamento (MER) proposto por [Che76]. O modelo ECR apresenta diversas vantagens quando comparado ao MER. Como exemplo, pode-se citar a introdução do conceito de categoria, a definição mais geral do conceito de atributo, a especificação mais completa das propriedades de cardinalidade e dependência de existência dos relacionamentos e, finalmente, a não necessidade de identificação de chaves primárias.

### 3.3.1 Entidade e Atributo

O modelo ECR é baseado em uma percepção do mundo real que consiste em uma coleção de objetos básicos chamados entidades. Uma **entidade** é um objeto que existe e que é distinguível dos outros objetos por um conjunto específico de propriedades. Uma entidade pode ser concreta, como uma pessoa, um carro, um livro, ou pode ser abstrata, como um conceito, um emprego. Um **tipo-entidade** é um conjunto de entidades do mesmo tipo, ou seja, entidades que possuem propriedades similares. Tipos-entidade são disjuntos, uma vez que uma entidade somente pode ser membro de um único tipo-entidade. A notação utilizada para representar entidades individuais é  $e_1, e_2, \dots, e_i$ , enquanto que a notação utilizada para representar tipos-entidade é  $T_1, T_2, \dots, T_i$ .

Os **atributos** são propriedades que descrevem entidades e relacionamentos. Para cada atributo existe um conjunto de valores permitidos, chamado domínio daquele atributo. Formalmente, um atributo  $a$  é definido como uma função com domínio definido em um tipo-entidade  $T$ , uma categoria  $C$  ou um relacionamento  $R$ , e com imagem definida em um conjunto de valores  $V$  ou no produto cartesiano de  $n$  conjuntos de valores:

$$a : T \rightarrow P(V) \quad \text{ou} \quad a : C \rightarrow P(V) \quad \text{ou} \quad a : R \rightarrow P(V)$$

onde  $P(V)$  é um conjunto potencial de valores.

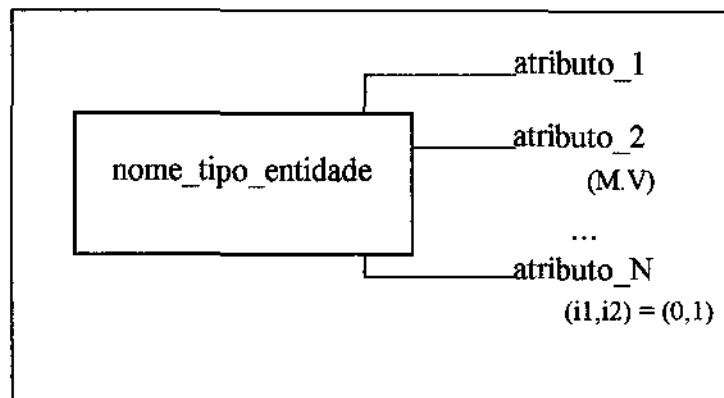
Os atributos de uma entidade ou de um relacionamento são de dois tipos: básicos e adquiridos. **Atributos básicos** são atributos inerentes à entidade, independente dos relacionamentos de que esta participa. Já os **atributos adquiridos** são atributos não fundamentais à entidade, os quais foram associados a ela através de um processo de abstração no qual o relacionamento que envolvia a entidade foi omitido do projeto do banco de dados. Como exemplo, um atributo básico (inerente) de um tipo-entidade *pessoa* é o seu *nome*, ou seja, pessoas possuem nome independentemente do contexto do problema. Por outro lado, um atributo adquirido (não fundamental) deste tipo-entidade *pessoa* é o seu *número empregado*, ou seja, nem todas as pessoas possuem um número de empregado como atributo em qualquer contexto. Neste caso, o atributo *número empregado* é representado como um atributo de *pessoa* porque, devido a um

processo de abstração, o relacionamento que associava o tipo-entidade *pessoa* ao tipo-entidade *companhia* e que tinha *número\_empregado* como atributo foi omitido do projeto do banco de dados.

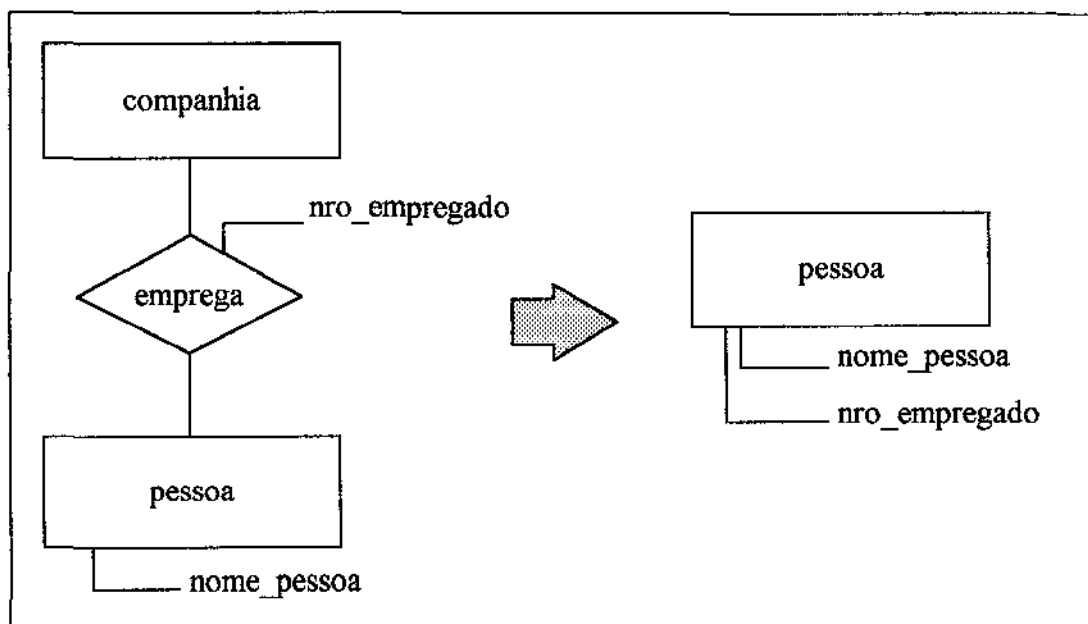
O modelo ECR permite uma definição mais geral do conceito de atributo, quando comparado ao MER. Os atributos podem ser classificados como **simples** ou **compostos** e, de acordo com as suas cardinalidades, como **monovalorados** ou **multivalorados**. Um atributo é simples quando possui apenas valores simples, ou seja,  $V$  é um conjunto de valores simples, e é composto quando é formado por vários outros atributos, ou seja,  $V$  é o produto cartesiano de diversos conjuntos  $V_1, V_2, \dots, V_n$ , onde  $V = V_1 \times V_2 \times \dots \times V_n$ . A cardinalidade de um atributo determina o número de valores, obtidos de um conjunto de valores  $V$ , que podem ser associados a esse atributo. Desta forma, de acordo com a cardinalidade, os atributos podem assumir apenas um único valor (monovalorado) ou vários valores (multivalorado). A restrição de cardinalidade para um determinado atributo é especificada por um par de números  $(i_1, i_2)$ , onde  $0 \leq i_1 \leq i_2$  e  $i_2 > 0$ . Estes números correspondem, respectivamente, ao número mínimo e máximo de valores que o atributo pode assumir dentro de um determinado domínio. No caso em questão, o atributo será monovalorado se  $i_2 = 1$  e multivalorado se  $i_2 > 1$ . Adicionalmente, um atributo pode ser total ou parcial. No primeiro caso, não é permitido que o atributo tenha valor nulo, ou seja,  $i_1 \geq 1$ . Já no segundo caso, o atributo pode assumir valor nulo, ou seja,  $i_1 = 0$ .

O modelo ECR também oferece o conceito de restrição de unicidade, através do qual pode ser determinado um atributo ou um conjunto de atributos que identificam univocamente uma entidade. Este conceito é similar ao de chaves primárias no MER, diferindo deste apenas pela não necessidade de sua especificação.

A notação gráfica utilizada pelo modelo ECR para a representação dos tipos-entidade é um retângulo com o nome do tipo-entidade escrito em seu interior. Já os atributos são representados por formas ovais conectadas através de segmentos de linhas simples aos tipos-entidade que eles descrevem. Os nomes dos atributos também são escritos no interior da figura. Por simplicidade, as formas ovais que representam os atributos não serão explicitadas. Desta forma, os seus nomes dos atributos serão conectados aos respectivos tipos-entidade através de segmentos de linhas simples. O par de números  $(i_1, i_2)$  que especifica a restrição de cardinalidade deve ser representado próximo ao atributo que está restringindo. Caso a especificação não seja realizada, os valores "default" assumidos são  $i_1 = i_2 = 1$  (atributo monovalorado e total). A cardinalidade "default" para um atributo multivalorado é  $i_1 = 1, i_2 = \infty$ . Neste caso, o fato do atributo ser multivalorado deve ser representado pelo símbolo M.V. próximo ao seu nome. A figura 3.14 sumariza a notação gráfica para a representação dos tipos-entidade e dos atributos. Na figura 3.15 estão representados o atributo básico *nome\_pessoa* e o atributo adquirido *número\_empregado* para o tipo-entidade *pessoa*. Para melhor compreensão da discussão anteriormente realizada, o processo de abstração que associa o atributo *número\_empregado* ao tipo-entidade *pessoa* é indicado.



**Figura 3.14** Representação dos conceitos tipo-entidade e atributo.



**Figura 3.15** Representação de atributos básicos e adquiridos.

### 3.3.2 Categoria

Um importante conceito adicionado pelo modelo ECR ao MER é o de **categoria**. Uma categoria é uma abstração utilizada para o agrupamento de entidades pertencentes a um ou mais tipos-entidade, de acordo com os papéis que estas entidades desempenham nos relacionamentos. Em outras palavras, categorias de entidade representam tanto um subconjunto de entidades de um tipo-entidade quanto um agrupamento de entidades de diversos tipos-entidade. A utilização deste conceito é fundamental, uma vez que através dele é possível representar categorias de generalização (superclasse) e categorias ISA (subconjunto). Vale destacar que as abstrações categoria de generalização e categoria ISA

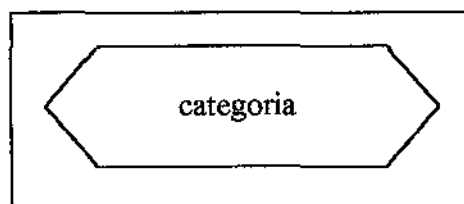
são importantes no projeto de SBDHs, uma vez que permitem conciliar diferentes modelagens utilizadas pelos diferentes SGBDs componentes [BL84].

Em uma **categoria de generalização**, múltiplas entidades que pertencem a tipos-entidade distintos que desempenham papéis similares em um ou mais relacionamentos são agrupadas em uma categoria (superclasse). Sejam os tipos-entidade *pessoa*, *automóvel* e *caminhão*. *Pessoa* e *automóvel* são ligadas entre si através de um relacionamento de posse. O mesmo é válido para os tipos-entidade *pessoa* e *caminhão*. Desta forma, como os tipos-entidade *automóvel* e *caminhão* desempenham papéis similares, estes podem ser agrupados na categoria *veículo*.

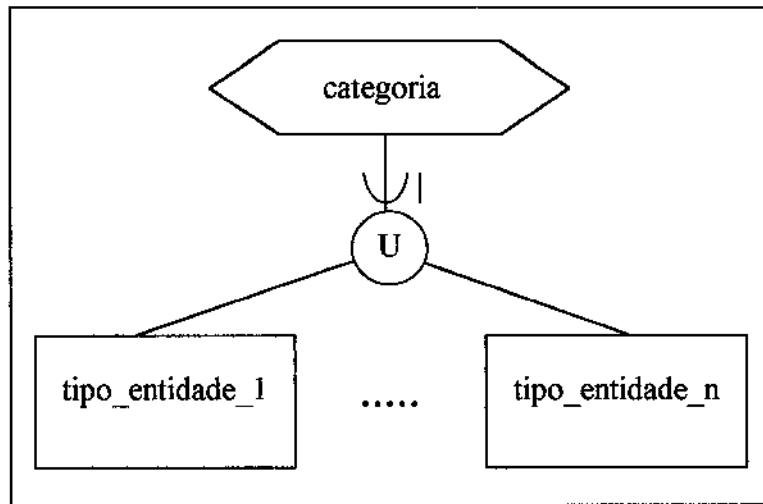
Já em uma **categoria de subconjunto** (categoria ISA), membros de um mesmo tipo-entidade são especializados em uma ou mais categorias. Considere o tipo-entidade *empregado*. Em uma aplicação de uma biblioteca, este poderia ser especializado, por exemplo, nas categorias *bibliotecário* e *faxineiro*.

Categorias não são necessariamente disjuntas, uma vez que uma determinada entidade pode ser membro de diversas categorias. Categorias podem possuir um conjunto de atributos associados a elas. Este conjunto é formado pela união de todos os atributos básicos de todos os tipos-entidade que formam a categoria, com todos os atributos adquiridos que foram especialmente definidos para a categoria. Vale ressaltar que estes atributos não são duplicações dos atributos dos tipos-entidade nos quais a categoria é definida. Na verdade, estes atributos são herdados destes tipos-entidade básicos. Desta forma, a utilização do conceito de categoria não promove redundância de dados.

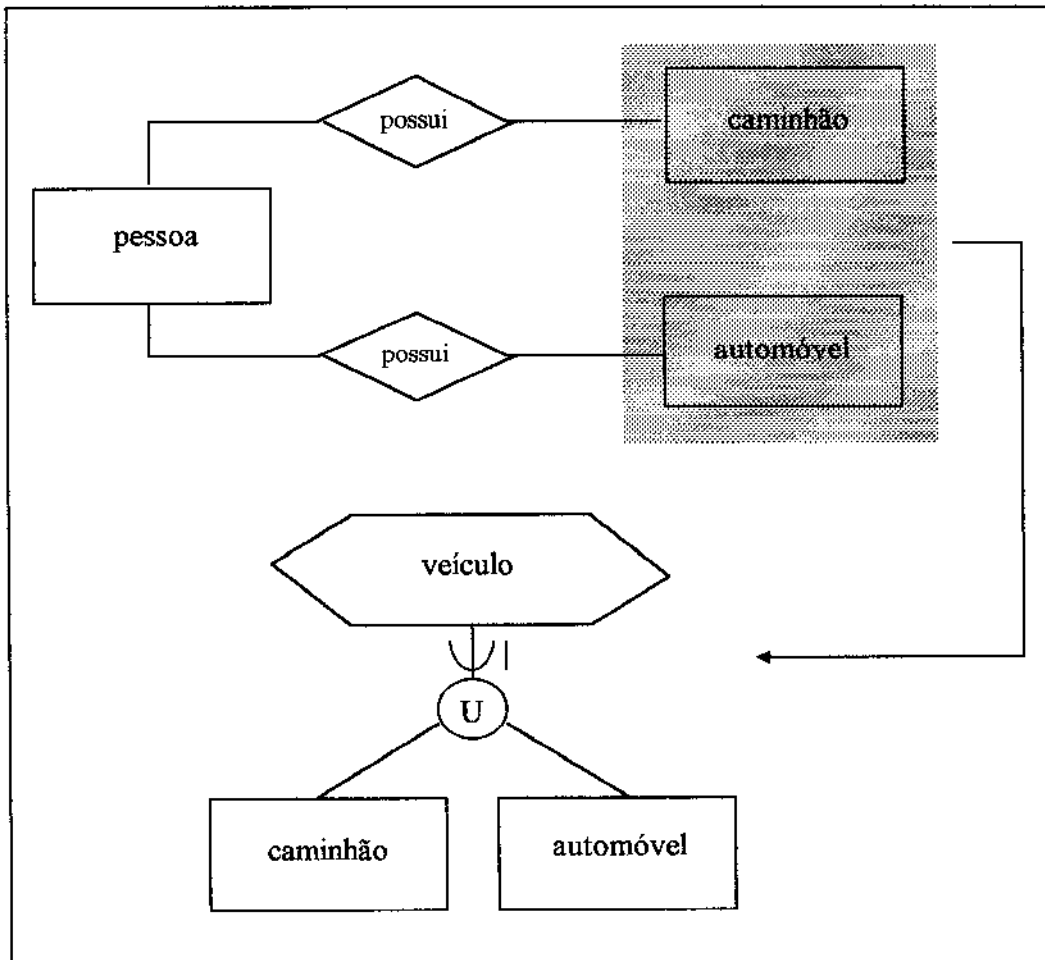
A notação gráfica utilizada para a representação da categoria é uma caixa hexagonal, com o nome da categoria escrito em seu interior (figura 3.16). A figura 3.17 ilustra a categoria de generalização. A letra U denota o fato da categoria ser um subconjunto da união de diversos tipos-entidade. A figura 3.18 considera o exemplo discutido anteriormente sobre os tipos-entidade *pessoa*, *automóvel* e *caminhão*.



**Figura 3.16** Representação gráfica de categoria.

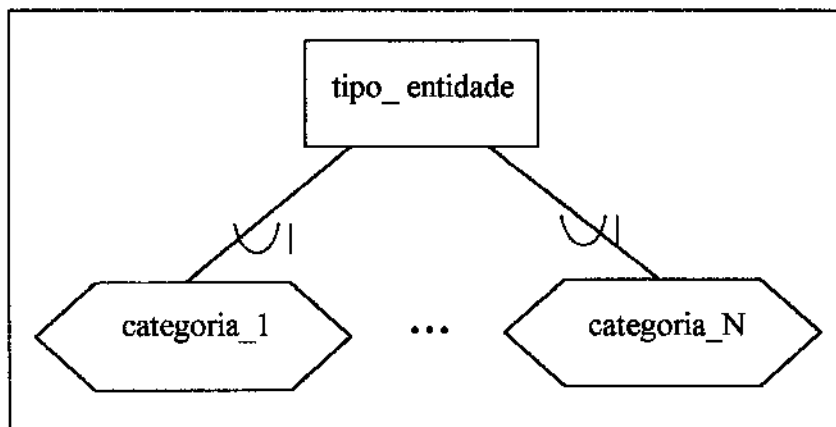


**Figura 3.17** Categoria de generalização.

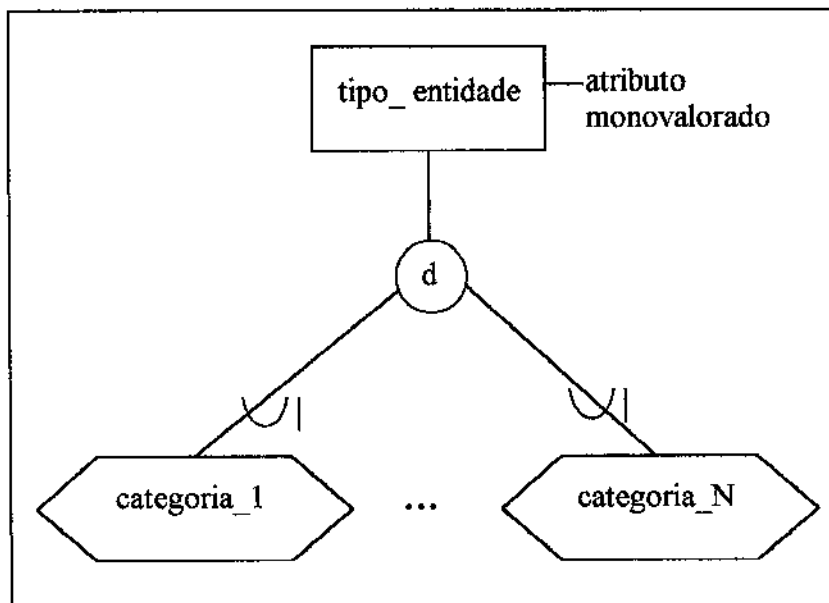


**Figura 3.18** Exemplo de categoria de generalização.

A categoria de subconjunto é ilustrada na figura 3.19 com um símbolo de subconjunto no arco que conecta a categoria ao tipo-entidade. Em alguns casos, restrições adicionais podem ser especificadas nas categorias. Pode-se, por exemplo, desejar que duas ou mais categorias que participam de uma categoria ISA sejam disjuntas. Para representar graficamente esta situação, é necessário que se especifique tanto o símbolo de disjunção quanto um atributo monovalorado no tipo-entidade no qual as categorias são especializadas. Esta situação é ilustrada na figura 3.20.



**Figura 3.19** Categoria de subconjunto (categoria ISA).



**Figura 3.20** Categoria de subconjunto disjunta.



A definição formal do conceito de categoria é:

$$C_i = T_1[S_1] \cup T_2[S_2] \cup \dots \cup T_n[S_n]$$

onde: para  $i \leq j \leq n$

- $C_i$  representa a categoria
- $T_j$  representa o tipo-entidade
- $U$  representa a operação união de conjuntos
- $S_j$  é um predicado opcional que restringe a participação das entidades do tipo-entidade  $T_j$  na categoria, ou seja, determina a condição na qual um membro de um tipo-entidade é também um membro da categoria  $C_i$ .

Através desta definição, pode-se observar que uma categoria pode ser idêntica a um tipo-entidade, quando  $j = 1$ . A representação gráfica desta situação é especificada sobrepondo-se a representação gráfica do conceito de categoria (caixa hexagonal) na representação gráfica do conceito tipo-entidade (retângulo). Vale destacar que ambos tipo-entidade e categoria são conjuntos de entidades.

### 3.3.3 Relacionamento e Cardinalidade

Além dos conceitos de entidade e categoria, o modelo ECR também usa o conceito de relacionamento para a modelagem dos problemas do mundo real. Um **relacionamento** é uma associação entre várias categorias (ou tipos-entidade). Uma **instância do relacionamento** é, por sua vez, um elemento (tupla) do relacionamento. Formalmente, os relacionamentos são relações matemáticas sobre categorias de entidade. A notação utilizada para representar os relacionamentos é:

$$R \subseteq X ( C_1, C_2, \dots, C_n ) = \{ ( e_1, e_2, \dots, e_n ) : e_j \in C_j \text{ para } 1 \leq j \leq n \}$$

onde:  $R$  : relacionamento  
 $C_i$  : categoria  
 $e_i$  : entidade  
 $X$  : produto cartesiano

Como visto anteriormente, o modelo ECR permite uma especificação mais completa das propriedades de cardinalidade e dependência dos relacionamentos, quando comparado ao MER. Similarmente à técnica utilizada para especificar restrições de cardinalidade nos atributos, a participação de uma categoria em um relacionamento é indicada por um par de números  $(i_1, i_2)$ , onde  $0 \leq i_1 \leq i_2$  e  $i_2 > 0$ . Estes números correspondem, respectivamente, ao número mínimo e máximo de membros de uma categoria que podem participar de

instâncias do relacionamento. De acordo com este par de números, os relacionamentos podem ser classificados como total, parcial, funcional ou específico.

A **participação total** de uma categoria em um relacionamento implica que uma entidade desta categoria deve, necessariamente, estar participando de uma instância do relacionamento em qualquer estado válido do banco de dados. Em outras palavras, a participação total de uma categoria em um relacionamento implica em uma dependência existencial das entidades daquela categoria relacionadas a entidades da outra categoria participante. Para tanto, o valor de  $i_1$  deve ser maior ou igual a 1. Já a **participação parcial** de uma categoria em um relacionamento implica na possibilidade de existir uma entidade pertencente a esta categoria que não esteja participando de nenhuma instância do relacionamento. Neste caso, o valor de  $i_1$  deve ser igual a 0.

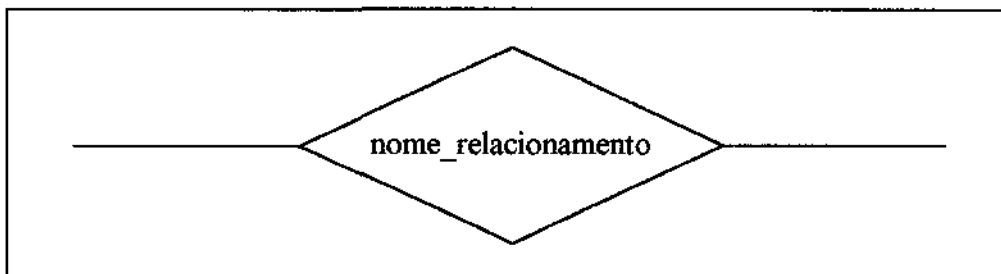
Um relacionamento é **funcional** quando no máximo uma entidade membro de uma categoria existir em um relacionamento em um determinado momento. Assim, o valor a ser especificado para  $i_2$  deve ser 1. Finalmente, um relacionamento é **específico** se ele é total e se uma entidade, uma vez ligada a uma instância do relacionamento, não puder ser removida desta instância a menos que a própria entidade seja eliminada.

Quando nenhuma especificação for realizada, os valores “default” assumidos pelos números  $i_1$  e  $i_2$  são, respectivamente, 0 e  $\infty$ . Isto corresponde a uma situação na qual não são especificadas restrições na participação das entidades no relacionamento.

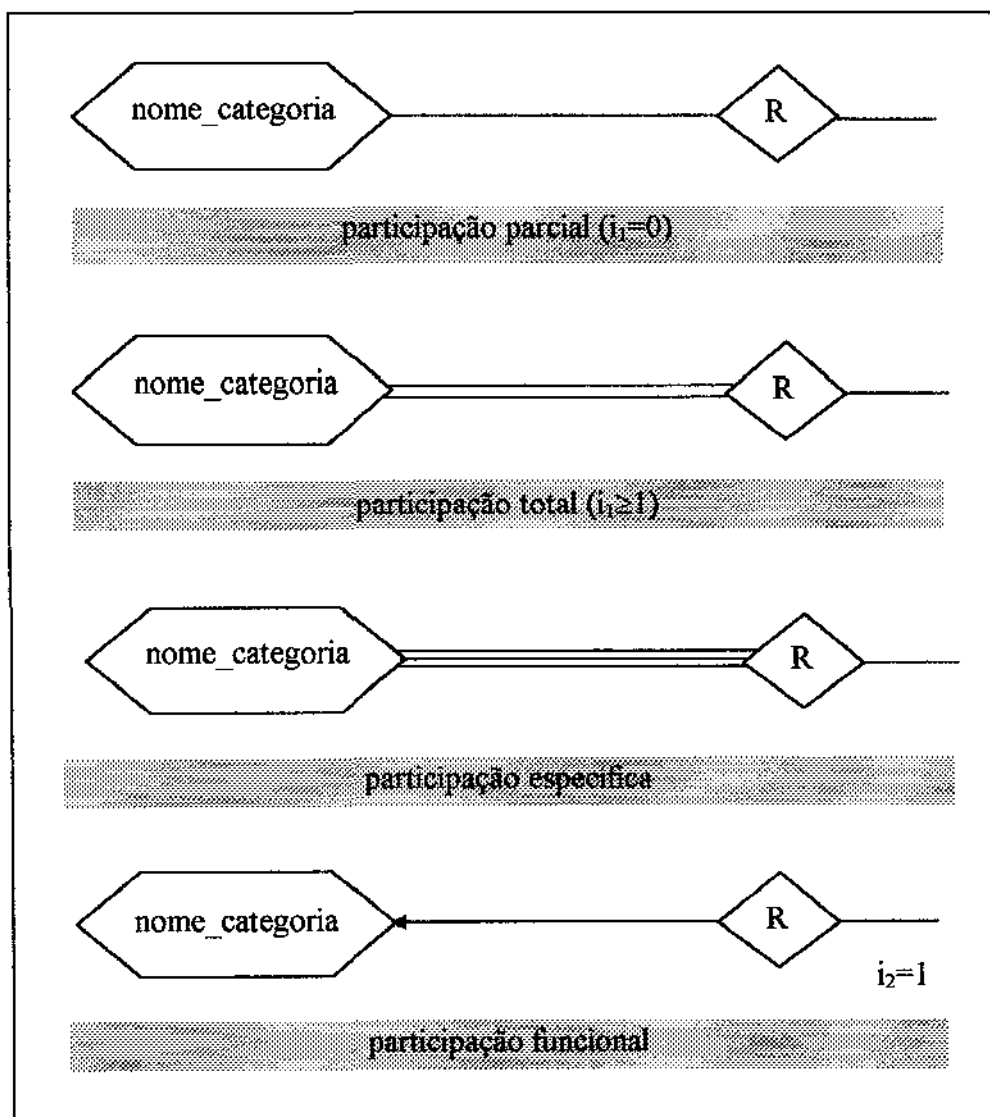
Três observações adicionais podem ser feitas com relação ao conceito de relacionamento. A primeira delas refere-se ao fato de que os relacionamentos, similarmente às entidades, podem ser descritos por atributos. As mesmas regras destacadas anteriormente para os atributos de entidades são válidas também para atributos de relacionamentos. A segunda observação diz respeito à possibilidade de uma categoria exercer múltiplos papéis dentro de um relacionamento, indicando que categorias que participam de um relacionamento não são necessariamente distintas. Em outras palavras, uma categoria pode ser associada a ela mesma através de um relacionamento unário (auto relacionamento). Finalmente, a terceira e última observação está relacionada à existência de nomes de conexão, os quais são associados às linhas que representam a participação das categorias nos relacionamentos. Estes nomes de conexão estabelecem ligações tanto no sentido entidade  $\rightarrow$  relacionamento quanto no sentido relacionamento  $\rightarrow$  entidade e são, desta forma, usados para referenciar entidades em uma categoria que são relacionadas a uma entidade de outra categoria por um relacionamento específico.

A notação gráfica utilizada pelo modelo ECR para a representação de relacionamentos é um losango com o nome do relacionamento escrito em seu interior (figura 3.21). O losango do relacionamento é ligado às categorias que dele participam através de linhas. Quando a linha que une uma categoria a um relacionamento é simples, a participação é parcial. Caso esta linha seja dupla, a participação é total. Finalmente, caso a linha seja tripla, a participação da categoria no relacionamento é específica. Já a participação

funcional é representada através de uma flecha, além da restrição adicional que  $i_2 = 1$ . A figura 3.22 exemplifica as notações acima citadas.



**Figura 3.21** Representação do conceito relacionamento.



**Figura 3.22** Representação gráfica da classificação dos relacionamentos.

Sempre que valores alternativos para  $i_1$  e  $i_2$  forem utilizados, eles devem ser especificados próximos às linhas de participação. De maneira geral, o valor  $i_1$  é representado pela própria participação (total, parcial, funcional ou específica) da categoria no relacionamento. Já o valor  $i_2$  é representado da mesma forma que no modelo ER. Alguns exemplos especificando os valores que podem ser assumidos pelo par de números  $(i_1, i_2)$  pode ser encontrada na figura 3.23.

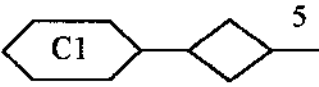
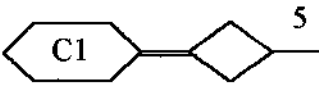
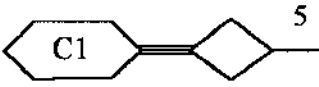
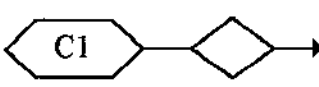
relacionamento	valores de $i_1$ e $i_2$	exemplo
participação parcial	$(i_1, i_2) = (0, i_2)$ $i_2$ : cardinalidade do relacionamento	 $(i_1, i_2) = (0, 5)$
participação total	$(i_1, i_2) = (1, i_2)$ $i_2$ : cardinalidade do relacionamento	 $(i_1, i_2) = (1, 5)$
participação específica	$(i_1, i_2) = (1, i_2)$ $i_2$ : cardinalidade do relacionamento	 $(i_1, i_2) = (1, 5)$
participação funcional	$(i_1, i_2) = (i_1, 1)$ $i_1$ : participação (parcial, total, específica)	 $(i_1, i_2) = (0, 1)$

Figura 3.23 Exemplos de valores de  $(i_1, i_2)$ .

Para finalizar a descrição dos conceitos empregados pelo modelo ECR é importante destacar a representação utilizada por este modelo para os nomes de conexão. Como foi visto anteriormente, os nomes de conexão são escritos próximos aos arcos que representam a participação das categorias nos relacionamentos. Considere o diagrama ECR esboçado na figura 3.24, o qual representa o relacionamento existente entre os tipos-

entidade *poligono* e *polilinha*. Por simplicidade, os nomes de conexão foram representados em itálico. Desta forma, o nome de conexão *polilinhas* associado ao relacionamento *é\_construído* permite uma referência do tipo *polilinhas of poligono*. Similarmente, o nome de conexão *poligonos* permite referenciar entidades pertencentes ao tipo-entidade *poligono* através de entidades do tipo *polilinha*, utilizando a expressão *poligonos of polilinha*. Os nomes de conexão podem ser considerados nomes invertidos dos papéis que uma categoria desempenha em um relacionamento.

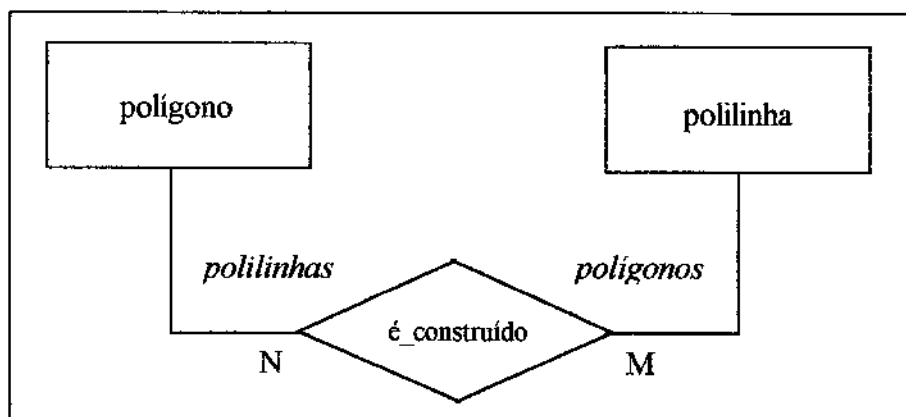


Figura 3.24 Representação do conceito nomes de conexão.

### 3.4 Um Modelo Orientado a Objetos para Aplicações Geográficas

As aplicações geográficas são aplicações que manipulam tanto atributos espaciais quanto atributos convencionais. A maioria das técnicas de modelagem tradicionais existentes para estas aplicações apresentam um tratamento diferenciado entre os dados espaciais e os dados alfanuméricos, utilizando convenções diferentes para sua modelagem e manuseio [TS94].

Desta forma, existe a necessidade de um modelo de dados genérico que permita que o entendimento conceitual de um problema ou aplicação seja o mais completo possível, ou seja, englobe todos os aspectos envolvidos na modelagem de uma aplicação geográfica. Em outras palavras, um modelo de dados geográfico consiste na formulação de um conjunto adequado de abstrações para a representação da realidade geográfica no BD, e na definição de critérios de manipulação e regras de integridade.

O modelo de dados geográfico aqui utilizado é um modelo orientado a objetos proposto por [Pir95]. A figura 3.25 esboça a estrutura deste modelo genérico, o modelo MODGEO<sub>2</sub> (Modelo de Organização de Dados Geográficos Orientados a Objetos). Este modelo serve como uma interface entre o usuário e o SIG, uma vez que o usuário possui

uma visão global abstrata dos dados a serem manipulados e os SIGs comerciais atualmente disponíveis oferecem apenas modelos diretamente relacionados às suas estruturas de dados internas. Em seguida é realizada uma breve descrição das classes que compõem o diagrama de objetos.

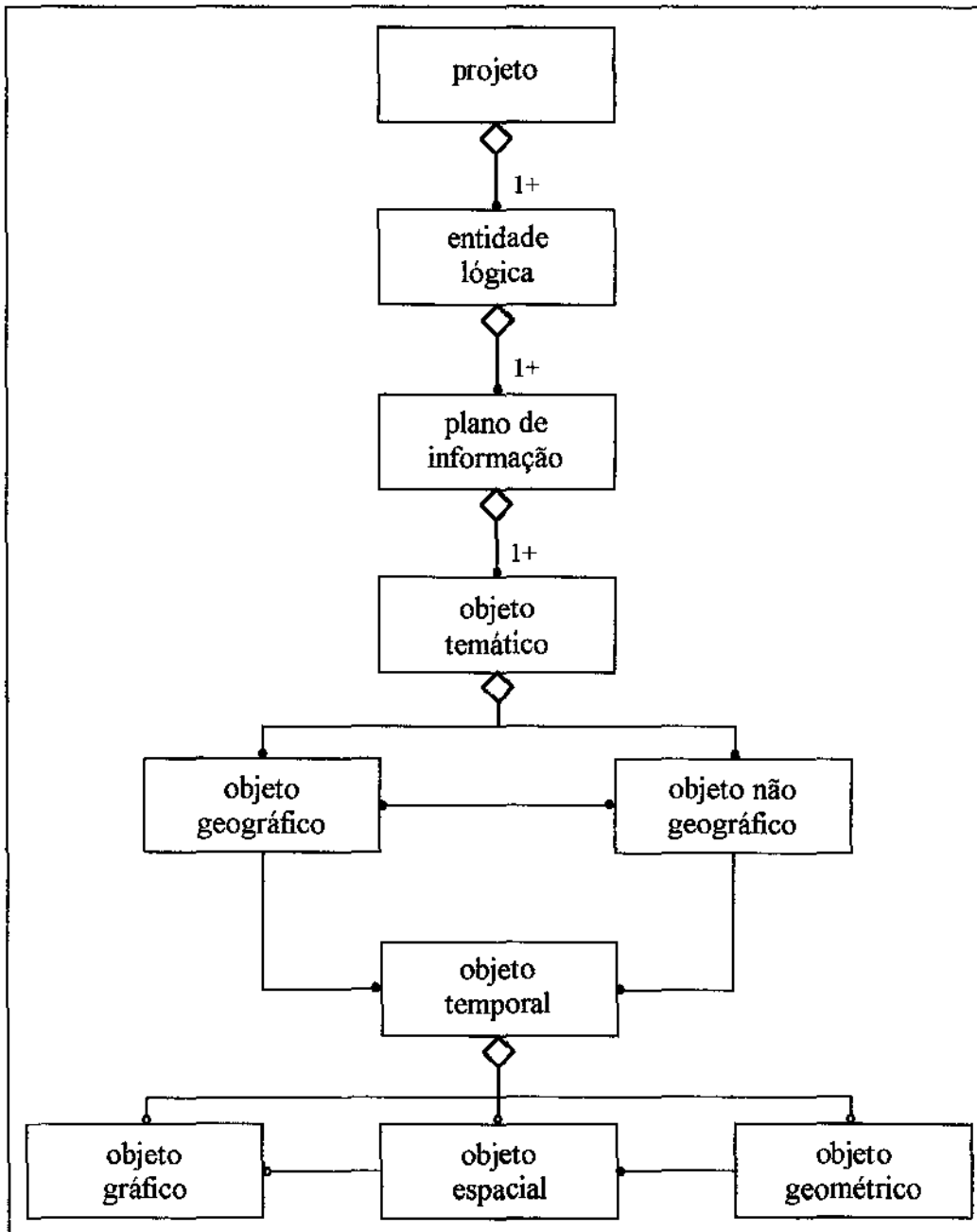


Figura 3.25 O modelo de dados MODGEO<sub>2</sub>.

- **projeto.** Um projeto é a unidade mais geral de informação, e representa o problema ou a aplicação geográfica a ser implementada. Esta classe possui os atributos *dataref*, *nome*, *objetivo*, *coordenador*. Além destes atributos, deve ser armazenado o

conjunto de *entidades lógicas* que compõem o projeto, uma vez que cada instância desta classe é composta por uma ou mais entidades lógicas. Exemplos de projeto são: planejamento urbano, gerenciamento de rede, gerenciamento de recursos energéticos e cadastramento rural, entre outros.

- **entidade lógica.** Uma entidade lógica ou região alvo é qualquer objeto que possua significado para o problema. Esta classe possui os atributos *dataref*, *descrição*, *projeção*, *ceinf* e *cdsup*. Estes dois últimos atributos referem-se ao retângulo envolvente, do qual deve ser armazenado apenas o canto inferior esquerdo (atributo *ceinf*) e o canto superior direito (atributo *cdsup*). Além destes atributos, deve ser armazenado o conjunto de *planos de informação* que compõem a entidade lógica, uma vez que cada instância desta classe é composta por um ou mais planos de informação. Exemplos de entidades lógicas são: estrada, construção, cidade e extensão geográfica, entre outros.
- **plano de informação.** Um plano de informação representa um conjunto de informações referente a uma superfície terrestre. Esta classe possui os atributos *dataref*, *descrição*, *formato* (raster ou vector) e *escala*. Além destes atributos, deve ser armazenado o conjunto de *objetos temáticos* que compõem o plano de informação, uma vez que cada instância desta classe é formada por um ou mais objetos temáticos. Exemplos de plano de informação são mapas de solo, mapas de vegetação, mapas cadastrais de lotes e rede elétrica de uma cidade, entre outros.
- **objeto temático.** Um objeto temático é composto por objetos geográficos e objetos não geográficos. Esta classe representa uma agregação de objetos pertencentes ao plano de informação acima descrito. Esta classe possui os atributos *t1* e *t2* (intervalo de tempo no qual os dados são válidos). Além destes atributos, deve ser armazenado o conjunto de *objetos geográficos* e *não geográficos* que compõem o objeto temático.
- **objeto geográfico.** Um objeto geográfico é um objeto que possui atributos espaciais, ou seja, possui localização geográfica. Esta classe possui os atributos *t1* e *t2*. Além destes atributos, deve ser armazenado uma lista de *objetos temporais* aos quais o objeto geográfico em questão está associado. Um objeto geográfico pode estar associado a objetos não geográficos.
- **objeto não geográfico.** Um objeto não geográfico é um objeto que não possui atributos espaciais. Esta classe possui os atributos *t1* e *t2*. Além destes atributos, deve ser armazenado uma lista de *objetos temporais* aos quais o objeto não geográfico em questão está associado. Um objeto não geográfico pode estar associado a objetos geográficos.
- **objeto temporal.** Um objeto temporal é uma agregação de objetos gráficos, objetos espaciais e objetos geométricos, ao longo do tempo. Com o decorrer do tempo, um

objeto temporal pode ser gerado a partir da variação de qualquer uma das características do objeto temático correspondente. Estas características podem ser tanto convencionais quanto espaciais, correspondendo, respectivamente, aos objetos não geográficos e geográficos. Esta classe possui os atributos *t1* e *t2*. Além destes atributos, deve ser armazenado o conjunto de *objetos gráficos, espaciais e geométricos* que compõem o objeto temporal.

- **objeto gráfico.** Um objeto gráfico representa a visualização gráfica do objeto temático. Alguns atributos desta classe são *cor, símbolo e hachurado*. Além destes atributos, devem ser armazenados os *objetos temporais* aos quais o objeto gráfico em questão pode estar associado. Exemplos de objetos gráficos são desenhos contendo várias árvores para representar florestas e o desenho de um avião para representar aeroportos, entre outros.
- **objeto espacial.** Um objeto espacial é a informação espacial do objeto geográfico no tempo. Esta classe possui como atributos um conjunto de *pontos* correspondentes às coordenadas do objeto geográfico em questão. Além deste atributo, deve ser armazenado os objetos gráficos e os objetos geométricos associados.
- **objeto geométrico.** Um objeto geométrico é utilizado para descrever características geométricas de um objeto. Esta classe deve conter como atributos as informações comuns a todas as classes geométricas suportadas pelo sistema. Uma melhor definição destas classes pode ser encontrada em [SV90]. Além destes atributos, devem ser armazenados os *objetos temporais* aos quais o objeto geométrico em questão pode estar associado. Como exemplo, pode-se indicar que as coordenadas do objeto temático correspondem a uma linha, a um polígono irregular ou a um círculo.



# Capítulo 4

## Algoritmos para Transformação de Modelos

Como visto no capítulo 3, antes de se realizar a integração propriamente dita dos esquemas dos bancos de dados relacional e orientado a objetos, é necessário efetuar o mapeamento entre os conceitos do modelo ECR e os conceitos do modelo OMT. Este capítulo descreve os mapeamentos propostos para este fim. Além disso, tendo em vista a completude do trabalho realizado, são descritos os mapeamentos entre o modelo OMT e o modelo ECR e entre o modelo OMT e o modelo utilizado pelo sistema  $O_2$ . Vale ressaltar que as regras para a tradução do modelo ECR para o modelo relacional são apresentadas em [EWH85, Oli93].

O capítulo está organizado da seguinte forma. Inicialmente, são apresentadas algumas propostas de mapeamento entre modelos encontradas na literatura. A seguir, são descritos os mapeamentos desenvolvidos na dissertação.

### 4.1 Mapeamentos Existentes

#### 4.1.1 Modelo Orientado a Objetos $\rightarrow$ Modelo Relacional

##### **Sacramento e Laender**

O modelo orientado a objetos utilizado na metodologia proposta por [SL94] é o modelo de dados do sistema  $O_2$ . Duas simplificações foram realizadas neste modelo com o objetivo de facilitar o mapeamento para o modelo relacional canônico: a imposição de que todas as entidades são objetos e a introdução do conceito de identificador. Um

identificador é um atributo da classe que permite distinguir um objeto de outro por suas propriedades descritivas.

Além destas simplificações, foi realizada também uma adaptação da propriedade *relacionamento* do sistema ObjectStore para a modelagem de objetos complexos. De acordo com esta propriedade, um relacionamento é um par de ponteiros inversos, ou seja, se um objeto possui um ponteiro para outro, então este segundo objeto também possui um ponteiro para o primeiro. Desta forma, é possível determinar relacionamentos de cardinalidade “um-para-um”, “um-para-muitos” e “muitos-para-muitos”, dependendo se o objeto possui um ponteiro para uma única instância de outra classe ou para um conjunto de instâncias de outra classe.

No processo de mapeamento proposto neste trabalho a parte estática é convertida em um esquema relacional, em termos de tabelas e restrições de integridade, enquanto que a parte dinâmica é representada pelos cabeçalhos dos procedimentos que implementarão as operações definidas para as classes. São considerados dois tipos de restrições de integridade: restrições de chave e restrições de integridade referencial. Além disso, uma opção de remoção é associada à restrição de integridade referencial. A remoção pode ser por bloqueio, por propagação ou por substituição por nulos.

As principais regras de mapeamento propostas são:

- cada classe pode ser mapeada para uma ou mais relações, dependendo da cardinalidade dos seus relacionamentos. Os atributos definidos para a classe em questão correspondem diretamente aos atributos das relações.
- cada relacionamento de cardinalidade “um-para-um” ou “um-para-muitos” é representado inserindo-se o atributo que representa o relacionamento como chave estrangeira em uma das relações. No caso de relacionamento “um-para-muitos”, este atributo deve permanecer na relação do lado “muitos”.
- cada relacionamento “muitos-para-muitos” é mapeado em uma nova relação, a qual possui como atributos as chaves primárias de ambas as classes relacionadas.
- cada hierarquia de generalização é mapeada em várias relações. A relação que representa a superclasse possui somente os atributos desta. Já as relações que representam as subclasses possuem, além dos seus próprios atributos, a chave primária da superclasse.

Para cada uma das regras de mapeamento acima definidas são determinadas também as restrições de integridade que atuam sobre as chaves primárias e estrangeiras das relações resultantes.

## Meng et al

A metodologia proposta por [Men+93] consiste em um conjunto de regras para a transformação da parte estrutural de um esquema orientado a objetos em um esquema relacional equivalente. Em outras palavras, os métodos não são considerados.

Nesta metodologia, dois tipos de relações são geradas: relações correspondentes às classes do esquema e relações correspondentes aos atributos das classes cujas definições de tipo utilizam o construtor *set*. Desta forma, a seguir são destacadas as principais regras de mapeamento propostas:

- cada classe é mapeada em uma relação. A chave primária desta relação é determinada utilizando-se o OID da classe em questão. Cada atributo atômico da classe corresponde diretamente a um atributo da relação.
- cada atributo da classe cuja definição de tipo não utiliza o construtor *set* e que referencie uma outra classe (atributo complexo) é renomeado com o mesmo nome da chave primária da classe referenciada. Este atributo permanece na relação correspondente à classe em questão e tem o papel de chave estrangeira.
- cada atributo da classe cuja definição de tipo utiliza o construtor *set* implica na criação de uma nova relação. Caso o atributo não referencie uma outra classe, a nova relação possuirá como atributos a chave primária da classe em questão e o próprio atributo. Caso contrário, ou seja, caso o atributo referencie uma outra classe, os atributos da relação serão as chaves primárias da classe em questão e da classe referenciada. Em ambos os casos, a chave primária da nova relação será a concatenação dos seus dois atributos.
- cada hierarquia de generalização “é-um” é mapeada em várias relações. A relação que representa a superclasse possui somente os atributos desta. Já as relações que representam as subclasses possuem, além dos seus próprios atributos, todos os atributos da superclasse.

Como os nomes dos atributos complexos ou dos identificadores dos objetos (OIDs) podem ter sido alterados durante o processo de transformação de esquemas, é criada uma tabela que armazene as alterações necessárias. Esta tabela tem por finalidade mostrar a correspondência entre os elementos do esquema orientado a objetos inicial e os elementos do esquema relacional obtido.

## 4.1.2 Modelo Relacional → Modelo Orientado a Objetos

### Keim, Kriegel e Miethsam

A metodologia proposta por [KKM93] utiliza informação semântica adicional oferecida pela modelagem entidade-relacionamento durante o processo de tradução do modelo relacional para o modelo orientado a objetos. A justificativa dos autores está relacionada ao fato de que o modelo orientado a objetos é mais estruturado e semanticamente mais rico do que o modelo relacional. Exemplos de conhecimento semântico adicional são: tabelas que representam os relacionamentos, tipos dos relacionamentos, atributos ou grupos de atributos que representam as chaves estrangeiras, entre outros. Esta informação semântica adicional é fundamental para que o processo de transformação seja capaz de substituir as chaves estrangeiras e os relacionamentos por referências diretas orientadas a objetos. Caso não seja possível obter estas informações automaticamente, o administrador do banco de dados deverá oferecer o suporte necessário.

Desta forma, o algoritmo proposto pode ser dividido em duas fases. Na primeira delas o esquema relacional inicial é transformado em um esquema entidade-relacionamento equivalente. São determinadas as entidades, os atributos e os relacionamentos. No segundo passo, este esquema intermediário é transformado em um esquema orientado a objetos final. As principais regras de mapeamento propostas são:

- cada entidade  $E$  com os atributos  $A_i$  de domínio  $D_i$  ( $i=1, \dots, n$ ) e chave  $K(E)$  é mapeada em uma classe de mesmo nome e que possui os mesmos atributos. Também são identificadas a(s) chave(s) primária(s) da nova classe.
- cada relacionamento funcional  $R: E \rightarrow F$  é mapeado em dois métodos: um para a classe  $E$  e um para a classe  $F$ . O método da classe  $E$  deve ser do tipo  $R: \rightarrow F$ , enquanto que o método da classe  $F$  deve ser do tipo  $R: \rightarrow \text{set}(E)$ .
- cada relacionamento não funcional  $R$  entre as entidades  $E_1, E_2, \dots, E_n$ , com possíveis atributos  $A_k$  de domínio  $D_k$  ( $k=1, \dots, q$ ) é mapeado em  $q+1$  métodos para a classe sendo mapeada. O primeiro método corresponde ao relacionamento e deve ser do tipo  $R: \rightarrow \text{set}(E_1 \times E_2 \times \dots \times E_{i-1} \times E_{i+1} \times \dots \times E_n)$  onde  $E_i$  é a entidade em questão. Os demais métodos correspondem aos atributos e devem ser do tipo  $A: E_1 \times E_2 \times \dots \times E_{i-1} \times E_{i+1} \times \dots \times E_n \rightarrow D$ .

Juntamente com a criação do esquema orientado a objetos, o mapeamento da transformação para o modelo entidade-relacionamento é armazenado em uma base de conhecimento. Esta base de conhecimento relaciona classes com entidades e relações, métodos com relacionamentos e atributos de relacionamentos, ou seja, todas as alterações realizadas.

## **Premerlani e Blaha**

A metodologia conceitual utilizada em [PB94] para representar o banco de dados orientado a objetos é a metodologia OMT. Desta forma, este trabalho oferece um algoritmo para a transformação do esquema relacional inicial em um esquema OMT equivalente, sendo que nenhuma regra de transformação é oferecida para mapear este esquema obtido em um modelo de dados orientado a objetos. Durante o processo de transformação dos esquemas é utilizada semântica adicional obtida de especificações de projeto, dos dicionários de dados, do conhecimento semântico do problema sendo analisado e de consultas realizadas sobre as informações armazenadas no banco de dados.

A seguir são destacados os passos propostos para a realização do mapeamento. Embora estes passos estejam implicitamente ordenados, eles podem ser executados em qualquer ordem, dependendo da aplicação.

- preparação de um modelo de objetos inicial. Este modelo de objetos possui uma classe para cada relação existente. Os atributos da classe correspondem aos atributos da relação. Este modelo de objetos inicial será refinado durante os demais passos.
- determinação das chaves candidatas e estrangeiras através de especificações existentes ou conhecimento semântico adicional.
- determinação de generalizações através de uma análise detalhada das chaves primárias das relações. Em muitos casos uma relação é mapeada como uma subclasse quando a sua chave primária é composta de um ou mais atributos que possuem o papel de chave estrangeira.
- determinação de associações e de suas multiplicidades. Cada classe do modelo de objetos inicial que possui como chave primária a concatenação de duas ou mais chaves estrangeiras é mapeada em uma associação. As demais associações são representadas através de chaves estrangeiras nas demais classes. As multiplicidades devem ser determinadas a partir de especificações existentes ou conhecimento semântico adicional.

Algumas transformações podem ser realizadas no esquema OMT gerado com a finalidade de torná-lo semanticamente mais rico. Um exemplo é a substituição de algumas associações pelo relacionamento de agregação. Esta substituição deve ser realizada baseada no conhecimento semântico da aplicação. Outro exemplo é a utilização do qualificador na tentativa de restringir a multiplicidade de uma associação.

## 4.1.3 Modelo ECR → Modelo Relacional

### Elmasri, Weeldreyer e Hevner

Juntamente com a proposta do modelo entidade categoria relacionamento, o trabalho descrito em [EWH85] apresenta as regras de mapeamento deste modelo para o modelo relacional. Uma característica deste trabalho é a utilização do conceito “surrogate” como um atributo identificador das entidades pertencentes a um tipo-entidade ou a uma categoria. O “surrogate” é definido automaticamente pelo sistema e é invisível aos usuários finais. A necessidade de sua existência está relacionada ao fato de que o modelo ECR não exige a declaração de chave primária.

As principais regras de mapeamento propostas são:

- cada tipo-entidade é mapeado em uma relação com um único atributo identificador (“surrogate”). Os atributos monovalorados do tipo-entidade correspondem diretamente aos atributos da relação.
- cada categoria é mapeada em uma relação, a qual possui, além dos atributos monovalorados da categoria, o “surrogate” correspondente. Em adição, de acordo com o tipo de categoria (generalização ou ISA), é especificada a restrição de integridade referencial que relaciona este “surrogate” aos “surrogates” dos tipos-entidade básicos.
- cada atributo multivalorado é mapeado em uma nova relação, a qual possui como atributos o próprio atributo e o “surrogate” da relação que representa o tipo-entidade, a categoria ou o relacionamento. A chave primária desta nova relação será a combinação do “surrogate” com o atributo multivalorado.
- cada relacionamento funcional entre as categorias (tipos-entidade)  $E_1$  e  $E_2$  é representado inserindo-se o “surrogate” da categoria (tipo-entidade)  $E_1$  na relação correspondente à categoria (tipo-entidade)  $E_2$ . Neste caso,  $E_2$  representa a categoria (tipo-entidade) cujo valor de  $i_2$  é 1.
- cada relacionamento não funcional é mapeado em uma nova relação, a qual possui como atributos os “surrogates” de ambas as categorias relacionadas e os próprios atributos do relacionamento. A chave primária desta nova relação será a concatenação dos “surrogates”.

## Oliveira

[Oli93] propõe uma metodologia que examina os construtores tipo-entidade, categoria e relacionamento seqüencialmente, uma vez que as categorias são definidas sobre os tipos-entidade e os relacionamentos são definidos sobre os tipos-entidade e as categorias. Outra característica da metodologia é a utilização do conceito “surrogate” nas relações em que não são definidos atributos que identifiquem univocamente uma entidade. O “surrogate” é definido automaticamente pelo sistema, é visível aos usuários e pode ser utilizado na construção das operações de junção das relações.

As principais regras de mapeamento propostas são:

- cada tipo-entidade é mapeado em uma relação de mesmo nome. Os atributos monovalorados do tipo-entidade correspondem diretamente aos atributos da relação. A chave primária desta nova relação é a chave do tipo-entidade que a originou, ou, caso esta não exista, um “surrogate” formado pela concatenação do nome da relação com o sufixo ID.
- cada atributo composto é decomposto nos atributos simples que o compõem. Estes atributos simples são, por sua vez, incluídos na relação correspondente ao tipo-entidade, categoria ou relacionamento sendo mapeado.
- cada atributo multivalorado é mapeado em uma nova relação, a qual possui como atributos o próprio atributo e o identificador (ou “surrogate”) da relação que representa o tipo-entidade, a categoria ou o relacionamento. A chave primária desta nova relação será a combinação do identificador (ou “surrogate”) com o atributo multivalorado. Adicionalmente, é especificada a dependência de inclusão entre esta nova relação e a correspondente ao tipo-entidade, categoria ou relacionamento.
- cada categoria ISA é mapeada em uma relação cuja chave é o seu atributo identificador ou o “surrogate” criado para ela. Os atributos monovalorados da categoria são incluídos na relação gerada.
- cada tipo-entidade que compõe uma categoria de generalização é mapeado em uma relação, a qual possui todos os atributos monovalorados da categoria mais os atributos monovalorados específicos do tipo-entidade em questão. A chave primária desta nova relação é o atributo identificador (ou o “surrogate”) da categoria.
- cada relacionamento “muitos-para-muitos” é mapeado em uma nova relação, a qual possui como chave primária os atributos identificadores (ou “surrogates”) de ambos os tipos-entidade (ou categorias) associados. Em adição, esta nova relação possui os atributos monovalorados específicos do relacionamento.

- cada relacionamento “um-para-um” ou “um-para-muitos” que possui atributos próprios é mapeado em uma nova relação, a qual possui como chave primária os atributos identificadores (ou “surrogates”) de ambos os tipos-entidade (ou categorias) associados. Em adição, esta nova relação possui os atributos monovalorados específicos do relacionamento.
- cada relacionamento “um-para-um” ou “um-para-muitos” que não possui atributos próprios é representado inserindo-se o identificador (ou “surrogate”) do tipo-entidade (ou categoria) do lado “muitos” como chave estrangeira da relação que representa o lado “um”.

Tanto as correspondências entre as relações geradas e os construtores do modelo ECR quanto as restrições que atuam sobre os valores dos atributos são armazenadas no dicionário de dados durante o processo de mapeamento.

## 4.2 Mapeamentos Propostos

Esta seção descreve os mapeamentos propostos entre os modelos ECR e OMT, além do mapeamento  $OMT \rightarrow O_2$ .

### 4.2.1 Modelo ECR $\rightarrow$ Modelo OMT<sup>1</sup>

O processo de mapeamento do modelo ECR para o modelo OMT proposto neste trabalho tem início com a análise dos tipos-entidade, das categorias e dos seus respectivos atributos. Em seguida, são estudados os relacionamentos binários e seus respectivos atributos. Finalmente, alguns conceitos menos utilizados, tal como relacionamentos ternários ou de ordem superior são examinados. Os conceitos do modelo ECR são convertidos em uma série de classes, atributos e associações entre estas classes, além da especificação de restrições de integridade que atuam sobre os objetos. Estas restrições de integridade devem ser especificadas entre chaves.

Desta forma, a seguir é apresentada a metodologia para a tradução de esquemas ECR em esquemas OMT equivalentes. No final desta seção são apresentadas várias figuras que exemplificam as regras de mapeamento propostas (figuras 4.1 a 4.4).

---

<sup>1</sup> Qualquer conceito seguido por apóstrofe (') representa, nesta seção, um conceito do modelo OMT.



### 4.2.1.1 Tipo-entidade e Categoria

Um tipo-entidade ou uma categoria do modelo ECR corresponde a uma classe no modelo OMT. O nome da classe é o mesmo nome do tipo-entidade ou da categoria.

Uma categoria ISA  $C\_ISA$  entre o tipo-entidade (ou categoria)  $E$  e as categorias  $C_1, C_2, \dots, C_n$  é mapeada em uma hierarquia de generalização  $G$  entre a superclasse  $E'$  e as subclasses  $C_1', C_2', \dots, C_n'$  no modelo OMT. A superclasse de  $G$  corresponde diretamente a  $E$ , enquanto que as subclasses correspondem respectivamente a  $C_1, C_2, \dots, C_n$ , de acordo com a regra anteriormente especificada.

$C\_ISA$  pode ser disjunta ou não. No primeiro caso, o triângulo que representa  $G$  não deve ser pintado. Além disso, deve ser especificado o discriminador para  $G$ . Este discriminador corresponde a um atributo monovalorado declarado no tipo-entidade (ou categoria)  $E$ . O tipo de dado do discriminador deve ser obtido a partir de informações do dicionário de dados. Desta forma, este atributo de  $E$  não deve ser considerado posteriormente no mapeamento dos atributos dos tipos-entidade (ou categoria), uma vez que ele já foi mapeado no modelo OMT em um discriminador. Já no segundo caso, ou seja,  $C\_ISA$  não é disjunta, o triângulo que representa  $G$  deve ser pintado.

Uma categoria de generalização entre a categoria  $C$  e os tipos-entidade (ou categorias)  $E_1, E_2, \dots, E_n$  é mapeada em  $n$  classes no modelo OMT, denominadas  $E_1', E_2', \dots, E_n'$ . Neste caso, cada classe  $E_i'$  corresponde à junção do tipo-entidade  $E_i$  com a categoria  $C$ . Em outras palavras, cada classe  $E_i'$  do modelo OMT possuirá, além de seus próprios atributos, os atributos adquiridos e os relacionamentos de  $C$ .

As regras a seguir formalizam as observações anteriores:

para cada tipo-entidade  $E$  ou categoria  $C$  do modelo ECR  
crie uma classe de mesmo nome no modelo OMT

para cada categoria ISA do modelo ECR  
entre o tipo-entidade  $E$   
e as categorias  $C_1, C_2, \dots, C_n$

crie uma hierarquia de generalização  $G$  no modelo OMT da seguinte forma:  
para o tipo-entidade, crie uma superclasse de mesmo nome  
para cada uma das categorias, crie uma subclasse de mesmo nome  
se a categoria ISA não é disjunta  
então pinte o triângulo que representa  $G$

*continua*

senão não pinte o triângulo que representa  $G$   
 determine o discriminador e o seu tipo de dado  
 elimine o atributo monovalorado correspondente ao discriminador de  $E$

para cada categoria de generalização do modelo ECR  
 entre a categoria  $C$   
 e os tipos-entidade  $E_1, E_2, \dots, E_n$

crie  $n$  classes no modelo OMT da seguinte forma:

para cada tipo-entidade  $E$

gere uma classe de mesmo nome que corresponde à junção de  $E$  com  $C$   
 para cada relacionamento de  $C$  com outro tipo-entidade (ou categoria)  $X$   
 crie uma associação entre as classes  $E$  e  $X$  de acordo com as regras  
 definidas para as associações

#### 4.2.1.2 Atributo de Tipo-entidade ou de Categoria

Um atributo pode ser classificado como simples ou composto e, de acordo com a sua cardinalidade, como monovalorado ou multivalorado. Adicionalmente, um atributo pode ser parcial ou total, dependendo se ele pode assumir valor nulo ou não. Desta forma, existem várias combinações possíveis para os atributos e cada uma destas combinações é mapeada diferentemente para o modelo OMT.

Um atributo simples monovalorado (parcial ou total)  $AM_o$  do tipo-entidade (ou categoria)  $E$  corresponde diretamente a um atributo de mesmo nome na classe  $E'$ , denominado  $AM_o'$ . Neste caso,  $E'$  corresponde ao mapeamento de  $E$ . O tipo de dado de  $AM_o'$  é o mesmo tipo de dado de  $AM_o$  e deve ser determinado através de informações obtidas do dicionário de dados. Se  $AM_o'$  pode assumir valores nulos, ou seja, se  $AM_o$  é um atributo parcial, então esta restrição externa deve ser especificada próxima ao atributo em questão.

O mapeamento de um atributo simples multivalorado (parcial ou total)  $AM_u$  do tipo-entidade (ou categoria)  $E$  implica em duas alterações no esquema OMT:

- a criação de uma nova classe, denominada  $AM_u'$ , que possui o mesmo nome do atributo em questão. Esta nova classe possui um único atributo, cujo nome é formado pela concatenação do string "a\_" com o nome de  $AM_u$ . O tipo de dado deste atributo é o mesmo tipo de dado de  $AM_u$  e deve ser determinado através de informações obtidas do dicionário de dados.

- a inserção de uma associação entre  $AMu'$  e a classe  $E'$ . O nome desta associação é determinado pela concatenação do string "ref\_" com o nome de  $AMu$ . A finalidade desta nova associação é, além de associar o atributo à sua respectiva classe, representar a sua cardinalidade. A multiplicidade de  $AMu'$  na associação será "zero ou mais" ou "um ou mais" – "muitos" – dependendo se  $AMu$  é um atributo parcial ou total, respectivamente. A multiplicidade de  $E'$  deve ser especificada externamente, com a ajuda do projetista ou do administrador do banco de dados. Seu valor "default" será "exatamente um".

Similarmente ao mapeamento do atributo simples multivalorado, o mapeamento de um atributo composto  $AC$  do tipo-entidade (ou categoria)  $E$  implica em duas alterações no esquema OMT:

- a criação de uma nova classe, denominada  $AC'$ , a qual possui o mesmo nome de  $AC$ . Os atributos de  $AC'$  correspondem aos atributos que compõem  $AC$  e possuem os mesmos nomes e os mesmos tipos de dados que estes.
- a inserção de uma associação entre  $AC'$  e a classe  $E'$ . O nome desta associação é determinado pela concatenação do string "ref\_" com o nome de  $AC$ . A multiplicidade de  $AC'$  na associação depende do fato do atributo em questão ser monovalorado, multivalorado, parcial ou total. Desta forma, se  $AC$  é um atributo monovalorado parcial, então a multiplicidade será "zero ou um". No caso de  $AC$  ser um atributo monovalorado total, então a multiplicidade será "exatamente um". Por outro lado, se  $AC$  for um atributo multivalorado parcial, a multiplicidade será "zero ou mais". Finalmente, caso  $AC$  seja um atributo multivalorado total, a multiplicidade será "um ou mais". Já a multiplicidade de  $E'$  na associação deve ser especificada externamente, com a ajuda do projetista ou do administrador do banco de dados. Seu valor "default" será "exatamente um".

Qualquer outro valor que o par de números  $(i_1, i_2)$  de um atributo assuma deve ser especificado no modelo OMT próximo à multiplicidade da classe que representa o atributo.

As regras a seguir formalizam as observações anteriores:

para cada atributo simples monovalorado  $AMo$  do modelo ECR  
pertencente ao tipo-entidade  $E$   
que possui tipo de dado  $TD$

estenda a declaração da classe  $E$  do modelo OMT da seguinte forma:

*continua*

crie um atributo de nome  $AM_o$  para esta classe de tipo de dado  $TD$

se o atributo em questão é parcial  
então especifique esta restrição externa

para cada atributo simples multivalorado  $AM_u$  do modelo ECR  
pertencente ao tipo-entidade  $E$   
que possui tipo de dado  $TD$

crie uma nova classe no modelo OMT da seguinte forma:  
atribua o nome  $AM_u$  à classe  
crie um atributo  $a_{AM_u}$  para esta classe de tipo de dado  $TD$

insira uma associação no modelo OMT  
entre as classes  $AM_u$  e  $E$   
chamada  $ref_{AM_u}$

se o atributo em questão é parcial  
então a multiplicidade da classe  $AM_u$  na associação é “zero ou mais”  
senão a multiplicidade da classe  $AM_u$  na associação é “um ou mais”

determine a multiplicidade da classe  $E$  na associação (“*default*”: “*um*”)

para cada atributo composto  $AC$  do modelo ECR  
pertencente ao tipo-entidade  $E$   
composto dos atributos  $A_i$  de tipos de dado  $TD_i$  ( $i=1, \dots, n$ )

crie uma nova classe no modelo OMT da seguinte forma:  
atribua o nome  $AC$  à classe  
para cada atributo  $A_i$   
crie um atributo de mesmo nome para esta classe de tipo de dado  $TD_i$

insira uma associação no modelo OMT  
entre as classes  $AC$  e  $E$   
chamada  $ref_{AC}$

se o atributo em questão é monovalorado parcial  
então a multiplicidade da classe  $AC$  na associação é “zero ou um”

se o atributo em questão é monovalorado total  
então a multiplicidade da classe  $AC$  na associação é “exatamente um”

*continua*

se o atributo em questão é multivalorado parcial  
então a multiplicidade da classe AC na associação é “zero ou mais”

se o atributo em questão é multivalorado total  
então a multiplicidade da classe AC na associação é “um ou mais”

determine a multiplicidade da classe E na associação (“*default*”: “*um*”)

### 4.2.1.3 Relacionamento Binário

Um relacionamento  $\mathcal{R}$  entre os tipos-entidade (ou categorias)  $E_1$  e  $E_2$  é mapeado em uma associação  $\mathcal{R}'$  entre as classes  $E_1'$  e  $E_2'$  no modelo OMT. Neste caso, as classes  $E_i'$  correspondem respectivamente aos tipos-entidade  $E_i$ . A associação tem o mesmo nome do relacionamento.

Um nome de conexão no modelo ECR corresponde diretamente a um nome de papel no modelo OMT. Os nomes assumidos por ambos os conceitos são os mesmos, porém as localizações são invertidas. Considere o relacionamento  $\mathcal{R}$  acima descrito. Seja  $ConE1$  o nome de conexão representado próximo ao tipo-entidade  $E_2$  e  $ConE2$  o nome de conexão representado próximo ao tipo-entidade  $E_1$ . No mapeamento para o modelo OMT,  $ConE1$  passa a ser representado próximo à classe  $E_1'$ , enquanto  $ConE2$  passa a ser representado próximo à classe  $E_2'$ .

A determinação da multiplicidade da associação  $\mathcal{R}'$  do modelo OMT é realizada a partir do par de números  $(i_1, i_2)$ , o qual representa respectivamente o número mínimo e máximo de membros de uma categoria (ou tipo-entidade) que podem participar de instâncias do relacionamento. Desta forma, o mapeamento da cardinalidade de  $\mathcal{R}$  na multiplicidade de  $\mathcal{R}'$  é realizado em duas etapas:

- a especificação dos valores  $(i_1, i_2)$  para cada tipo-entidade (ou categoria) que participa de  $\mathcal{R}$ . Estes valores são determinados a partir das cardinalidades do relacionamento. De maneira geral, se o relacionamento é total, então  $i_1 \geq 1$ . Por outro lado, se o relacionamento é parcial, então  $i_1 = 0$ . No caso do relacionamento ser funcional, então  $i_2 = 1$ . Finalmente, se o relacionamento é específico, então  $i_1 \geq 1$  e, em adição, deve ser especificada a restrição de integridade referencial que determina que uma entidade, uma vez ligada a uma instância do relacionamento, não pode ser removida desta instância a menos que a própria entidade seja deletada. A figura 3.23 apresenta alguns exemplos.
- a associação de  $(i_1, i_2)$  às suas respectivas classes no modelo OMT. Os valores determinados para a participação do tipo-entidade (ou categoria)  $E_1$  devem ser

especificados na associação  $R'$  próximos à classe  $E_2$ . Similarmente, os valores determinados para a participação do tipo-entidade (ou categoria)  $E_2$  devem ser especificados na associação  $R'$  próximos à classe  $E_1$ .

As regras a seguir formalizam as observações anteriores:

para cada relacionamento binário  $R$  do modelo ECR  
entre os tipos-entidade  $E_1$  e  $E_2$

crie uma associação no modelo OMT  
entre as classes  $E_1$  e  $E_2$   
com o mesmo nome do relacionamento

se existem nomes de conexão representados em  $R$   
então represente-os na associação, invertendo suas localizações e mantendo os  
seus nomes

determine a multiplicidade da associação:

para cada tipo-entidade (ou categoria) que participa do relacionamento

especifique os valores de  $(i_1, i_2)$

se  $i_1 = 0$  e  $i_2 = 1$   
então represente esta multiplicidade por um pequeno círculo vazio

se  $i_1 = i_2 = 1$   
então represente esta multiplicidade por uma linha sem símbolos

se  $i_1 = 0$  e  $i_2 = N$   
então represente esta multiplicidade por um pequeno círculo pintado

se  $i_1 = 1$  e  $i_2 = N$   
então represente esta multiplicidade por um pequeno círculo pintado e  
especifique 1+

se qualquer outro valor de  $i_1$  e  $i_2$  foi especificado  
então represente-o de acordo com os conceitos do modelo OMT

se a participação do tipo-entidade no relacionamento é específica  
então especifique esta restrição de integridade referencial

#### 4.2.1.4 Atributo de Relacionamento

As regras de mapeamento para os atributos de relacionamentos são similares às regras oferecidas para o mapeamento dos atributos de tipos-entidade e categorias. Desta forma, existem várias combinações possíveis para os atributos de relacionamento (por exemplo, simples monovalorado parcial, simples multivalorado total, composto monovalorado parcial e composto multivalorado total, entre outras) e cada uma destas combinações é mapeada diferentemente para o modelo OMT.

Um atributo simples monovalorado (parcial ou total)  $ASMo$  do relacionamento  $R$  corresponde diretamente a um atributo de mesmo nome na associação  $R'$ , denominado  $ASMo'$ . Neste caso, a associação  $R'$  corresponde ao mapeamento do relacionamento  $R$ . O tipo de dado de  $ASMo'$  é o mesmo tipo de dado de  $ASMo$  e deve ser determinado através de informações obtidas do dicionário de dados. Se  $ASMo'$  pode assumir valores nulos, ou seja, se  $ASMo$  é um atributo parcial, então esta restrição externa deve ser especificada próxima ao atributo em questão.

Um atributo composto monovalorado (parcial ou total)  $ACMo$  do relacionamento  $R$  é mapeado como uma associação como uma classe no modelo OMT, denominada  $ACMo'$ , a qual deve ser declarada na associação correspondente a  $R$ . O nome de  $ACMo'$  é o mesmo nome do atributo. Os atributos de  $ACMo'$  correspondem diretamente aos atributos que compõem  $ACMo$  e possuem os mesmos nomes e os mesmos tipos de dados que estes. Se  $ACMo'$  pode assumir valores nulos, ou seja, se  $ACMo$  é um atributo parcial, então esta restrição externa deve ser especificada próxima a  $ACMo'$ .

Um atributo multivalorado (parcial ou total)  $AMu$  do relacionamento  $R$  é mapeado como uma classe no modelo OMT, denominada  $AMu'$ . O nome da classe é o mesmo nome do atributo. Dependendo do fato do atributo ser simples ou composto, duas diferentes situações podem ocorrer:

- se o atributo é simples, então  $AMu'$  possui um único atributo, cujo nome é formado pela concatenação do string "a\_" com o nome de  $AMu$ . O tipo de dado deste atributo é o mesmo tipo de dado de  $AMu$  e deve ser determinado através de informações obtidas do dicionário de dados.
- se o atributo é composto, então os atributos de  $AMu'$  correspondem diretamente aos atributos que compõem  $AMu$  e possuem os mesmos nomes e os mesmos tipos de dados que estes.

Em ambas as situações, a classe  $AMu'$  deve estar relacionada à associação  $R'$ , a qual representa o mapeamento de  $R$ . Desta forma, se  $R$  é um relacionamento binário,  $R'$  será uma associação ternária. Em adição,  $AMu'$  será um dos atributos que compõem a chave

candidata para  $R'$ . Se  $AMu'$  pode assumir valores nulos, ou seja, se  $AMu$  é um atributo parcial, então esta restrição deve ser garantida através de métodos.

O mapeamento de um atributo multivalorado pertencente a um relacionamento de cardinalidade 1:1 ou 1:n pode ser realizado através de uma solução alternativa, a qual é dividida em dois passos. No primeiro passo, o atributo multivalorado é eliminado do relacionamento e inserido no tipo-entidade (ou categoria) do lado n. Caso a cardinalidade do relacionamento seja 1:1, então deve-se escolher um dos tipos-entidade (ou categorias). No segundo passo, o mapeamento deste “novo” atributo deve ser realizado de acordo com as regras definidas para os atributos de tipo-entidade ou categoria.

As regras a seguir formalizam as observações anteriores:

para cada atributo simples  $AS$  do modelo ECR  
pertencente ao relacionamento  $R$   
de tipo de dado  $TD$

se  $AS$  é um atributo monovalorado

então crie um atributo  $AS$  de tipo de dado  $TD$  na associação  $R$

se  $AS$  é um atributo parcial

então especifique esta restrição

se  $AS$  é um atributo multivalorado

então crie uma classe  $AS$  no esquema OMT

crie um atributo  $a\_AS$  para esta classe de tipo de dado  $TD$

relacione  $AS$  à associação  $R$

acrescente esta nova classe como chave candidata para a associação

para cada atributo composto  $AC$  do modelo ECR

pertencente ao relacionamento  $R$

composto dos atributos  $At_i$  de tipos de dados  $TD_i$

se  $AC$  é um atributo monovalorado

então crie uma associação como uma classe  $AC$  na associação  $R$

para cada atributo  $At_i$  de tipo de dado  $TD_i$

crie um atributo  $At_i$  de tipo de dado  $TD_i$  para a associação como uma  
classe

*continua*



se  $AC$  é um atributo multivalorado  
 então crie uma classe  $AC$  no esquema OMT  
 para cada atributo  $At_i$  de tipo de dado  $TD_i$   
 crie um atributo  $AT_i$  de tipo de dado  $TD_i$  para esta classe  
 relacione  $AC$  à associação  $R$   
 acrescente esta nova classe como chave candidata para a associação

#### 4.2.1.5 Relacionamento Ternário e de Ordem Superior

O mapeamento de relacionamentos ternários ou de ordem superior para o modelo OMT é realizado de maneira análoga ao mapeamento dos relacionamentos binários. Desta forma, generalizando a discussão realizada previamente, o mapeamento de qualquer relacionamento  $R$  entre os tipos-entidade (ou categorias)  $E_i$  ( $i \geq 1$ ) gera uma associação  $R'$  entre as classes  $E_i'$ ,  $E_i'$  correspondendo ao mapeamento do tipo-entidade (ou categoria)  $E_i$ . O nome da associação é o mesmo que o do relacionamento.

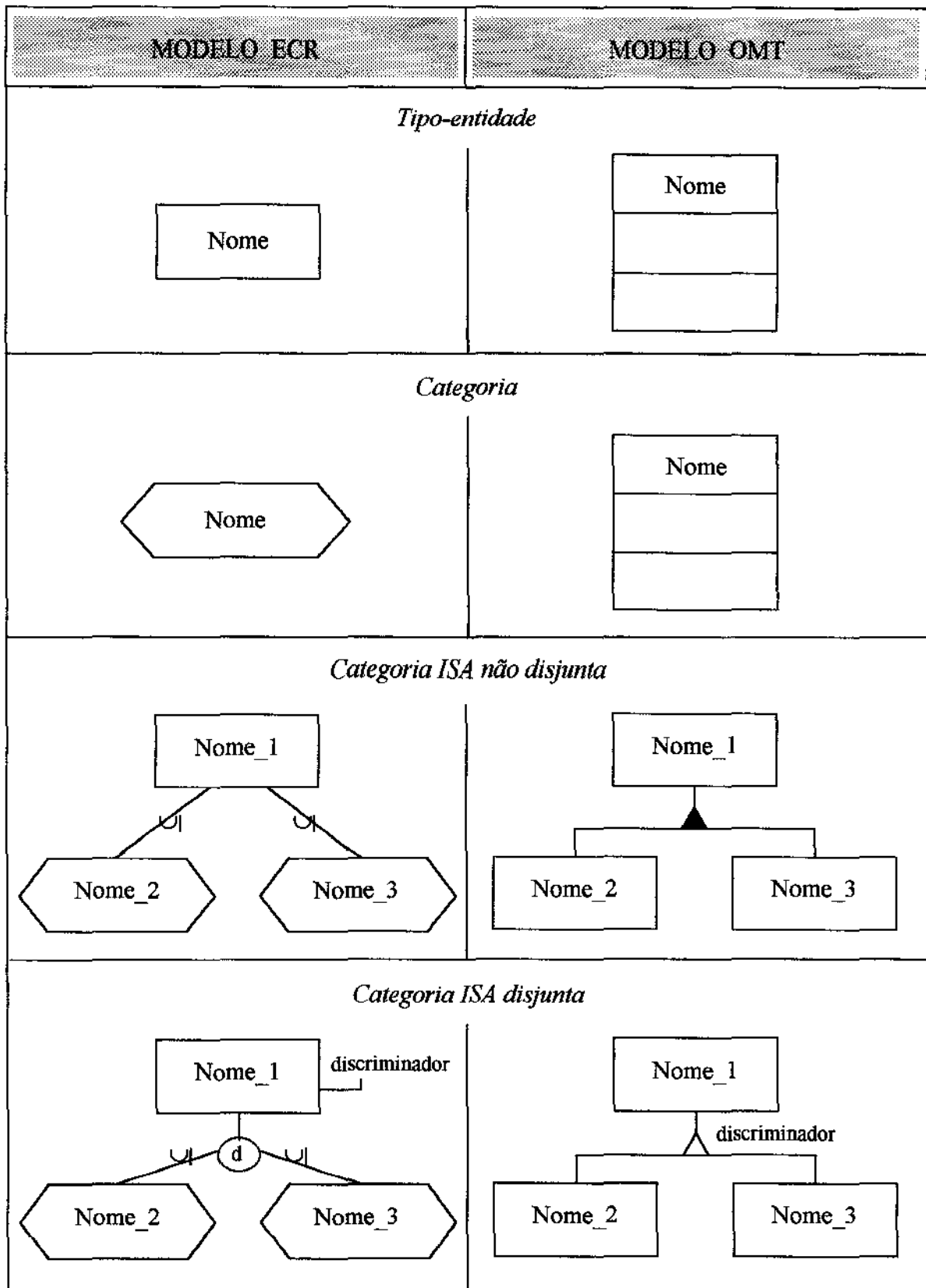
Ao invés de representar a multiplicidade das classes que participam de uma associação ternária ou de ordem superior através de símbolos, a metodologia OMT utiliza o conceito de chaves candidatas. Neste conceito, são escolhidas para chaves candidatas de uma associação as classes do lado “muitos”. Desta forma, o mapeamento das cardinalidades de  $R$  deve obedecer esta restrição. A chave candidata pode ser especificada mesmo quando a participação da classe na associação for opcional.

As regras a seguir formalizam as observações anteriores:

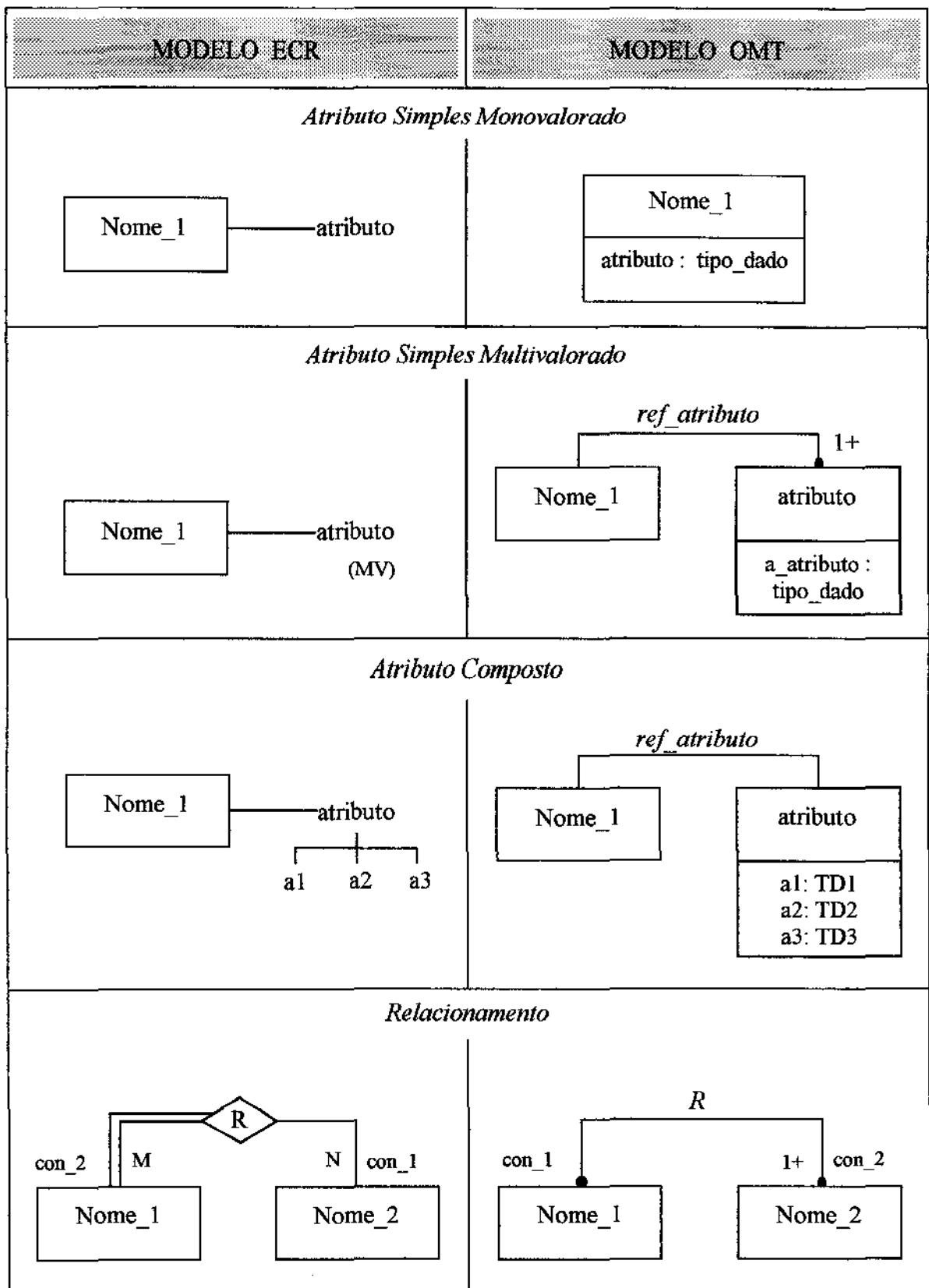
para cada relacionamento ternário ou de ordem superior  $RT$  do modelo ECR  
 que relaciona os tipos-entidade (ou categorias)  $E_1, E_2, \dots, E_n$  ( $n \geq 3$ )

crie uma associação  $RT$  no modelo OMT  
 entre as classes  $E_1, E_2, \dots, E_n$   
 com o mesmo nome do relacionamento

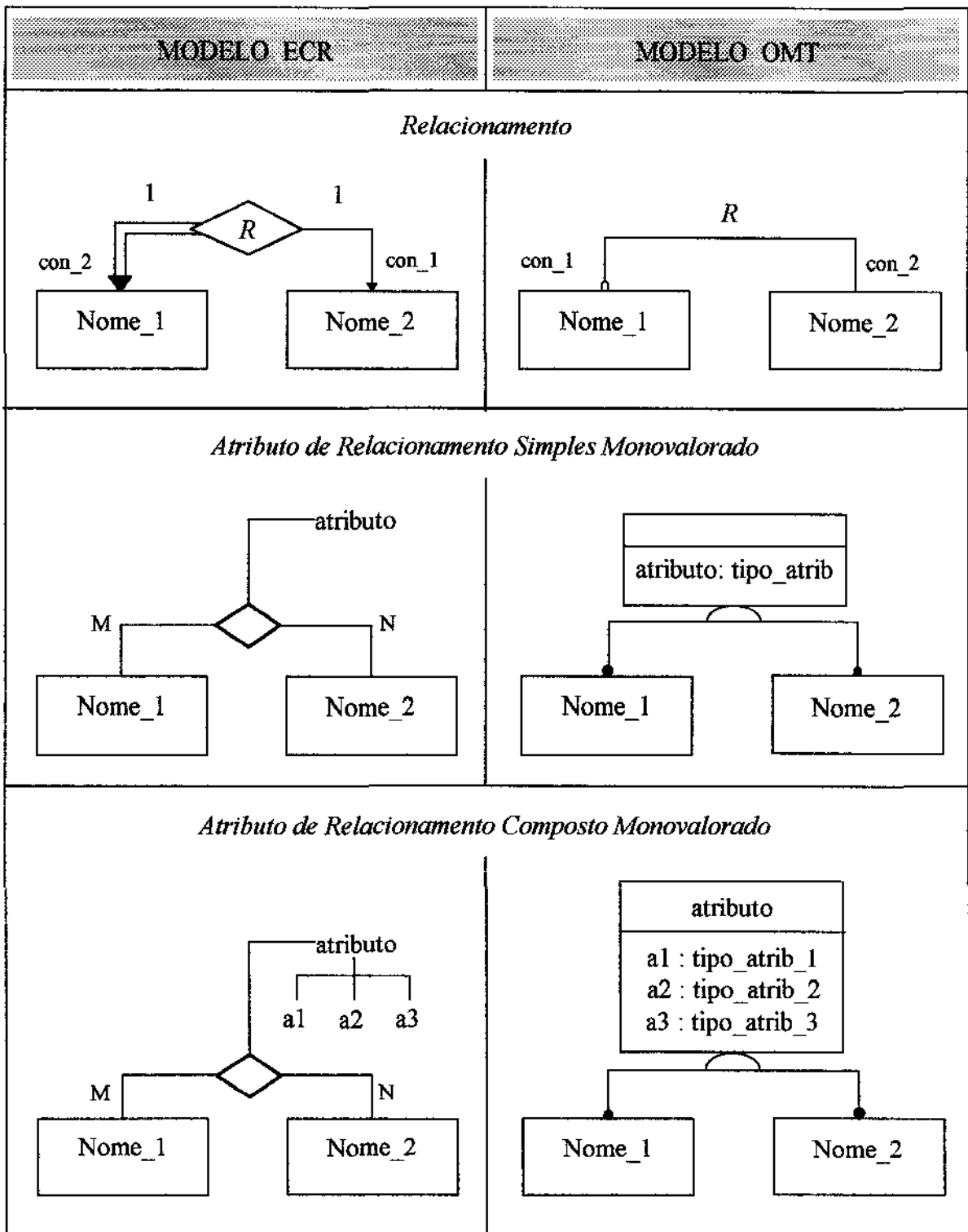
para cada tipo-entidade  $E$  que participa do relacionamento  
 se  $E$  representa um lado “muitos”  
 então insira a classe correspondente como chave candidata da associação



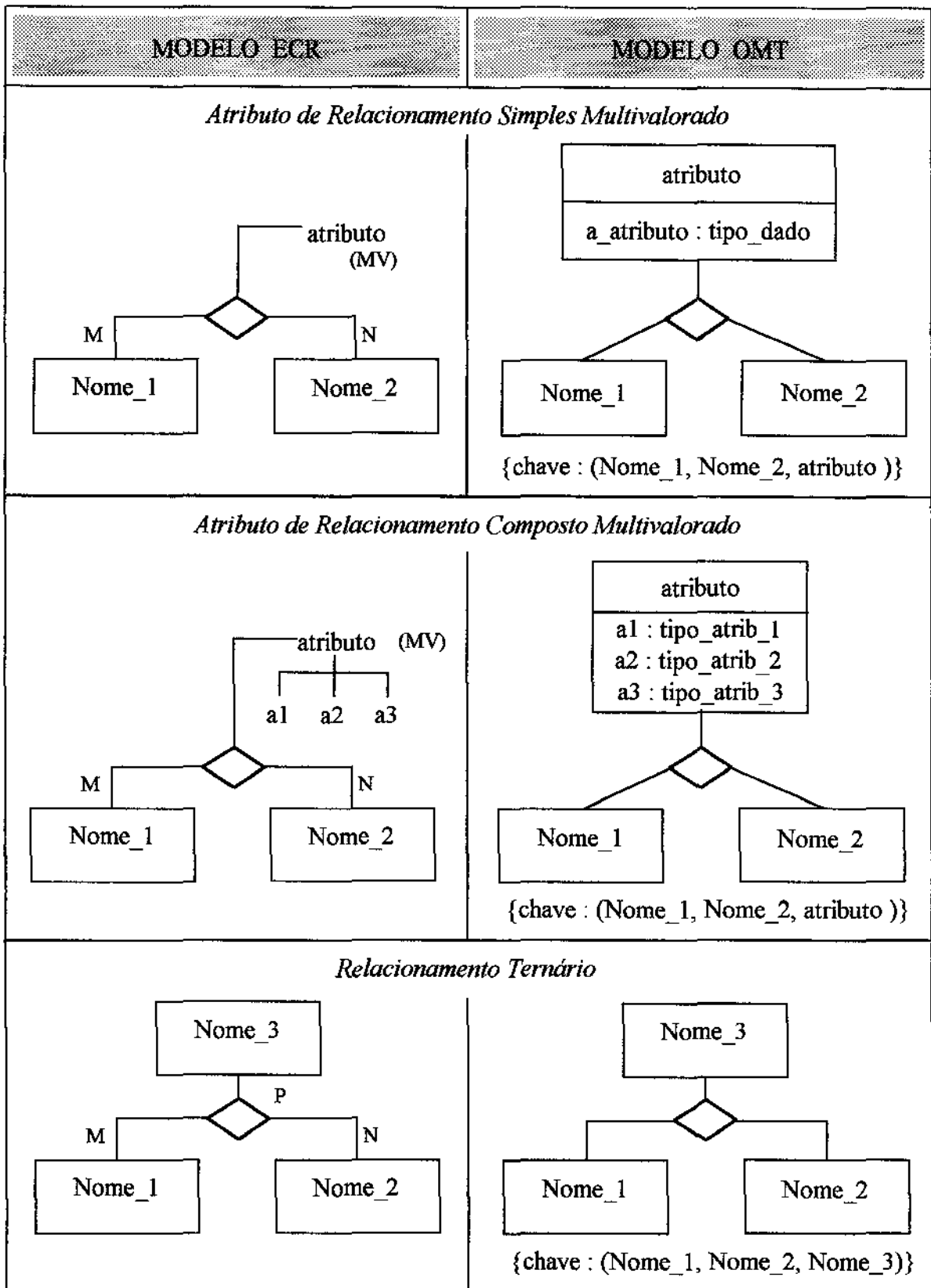
**Figura 4.1** Mapeamento ECR → OMT (parte 1)



**Figura 4.2** Mapeamento ECR → OMT (parte 2)



**Figura 4.3** Mapeamento ECR → OMT (parte 3)



**Figura 4.4** Mapeamento ECR → OMT (parte 4)

## 4.2.2 Modelo OMT → Modelo ECR<sup>2</sup>

O modelo OMT modela tanto a parte estrutural (ou estática) quanto a parte dinâmica (ou comportamental) de uma aplicação. Por outro lado, o modelo ECR apenas considera o aspecto estrutural. Desta forma, no processo de mapeamento proposto neste trabalho, a parte estática é convertida em uma série de tipos-entidade (ou categorias), atributos e relacionamentos entre estes tipos-entidade (ou categoria). Já a parte dinâmica é representada pelos cabeçalhos dos procedimentos que implementarão as operações definidas para as classes e será apenas indicada. Este tipo de procedimento é semelhante ao realizado por [SL94] no mapeamento objeto → relacional.

A seguir é apresentada a metodologia para a tradução de esquemas OMT em esquemas ECR equivalentes. A seqüência de apresentação da metodologia sugere uma ordem de execução dos mapeamentos. No final desta seção são apresentadas várias figuras que exemplificam as regras de mapeamento propostas (figuras 4.5 a 4.7).

### 4.2.2.1 Classe, Atributo e Operação

Uma classe  $C$  do modelo OMT gera um tipo-entidade ou uma categoria de mesmo nome no modelo ECR, denominado  $C'$ . A escolha entre tipo-entidade ou categoria deve ser realizada pelo projetista ou pelo administrador do banco de dados durante o processo de mapeamento.

Um atributo  $A$  da classe  $C$  corresponde diretamente a um atributo de mesmo nome no tipo-entidade (ou categoria)  $C'$ , denominado  $A'$ . O tipo de dado do atributo também será o mesmo e deve ser inserido no dicionário de dados. Este atributo  $A'$  é definido como simples, monovalorado e total. No entanto, caso tenha sido especificada no modelo OMT alguma restrição indicando que o atributo  $A$  pode assumir valores nulos, então a cardinalidade de  $A'$  será parcial ao invés de total. Já o valor "default" definido para  $A$  deve ser associado a  $A'$  todas as vezes que uma entidade for inserida em  $C'$ . Em outras palavras, o procedimento que implementa a inserção de uma entidade no banco de dados é responsável por esta associação.

Uma operação  $O$  da classe  $C$  é mapeada em um cabeçalho de procedimento de mesmo nome no modelo ECR. Em adição, a assinatura do procedimento possuirá os mesmos argumentos (nome e tipo) da lista de argumentos e o mesmo tipo resultante que a operação  $O$ . Já o método que implementa  $O$  corresponde à implementação do procedimento em questão, com as devidas alterações realizadas.

As regras a seguir formalizam as observações anteriores:

---

<sup>2</sup>Qualquer conceito seguido por apóstrofe (') representa, nesta seção, um conceito do modelo ECR.

para cada classe  $C$  do modelo OMT

com atributos  $A_i$ , tipos de dado  $TD_i$  e valores "default"  $VD_i$  ( $i=1, \dots, n$ )  
com operações  $O_j$ , lista de argumentos  $LA_j$  e tipo resultante  $TR_j$  ( $j=1, \dots, m$ )  
e com métodos  $M_j$ ,  $M_j$  correspondendo à implementação da operação  $O_j$

crie um tipo-entidade ou uma categoria de mesmo nome no modelo ECR

para cada atributo  $A_i$

crie um atributo de mesmo nome no tipo-entidade ou categoria  
insira o tipo de dado  $TD_i$  no dicionário de dados  
se não existe restrição de que  $A_i$  pode assumir valores nulos  
então declare-o como simples, monovalorado e total  
senão declare-o como simples, monovalorado e parcial

para cada operação  $O_j$

defina um cabeçalho de procedimento  
com lista de argumentos  $LA_j$   
de tipo resultante  $TR_j$   
implementado de acordo com o método  $M_j$

#### 4.2.2.2 Generalização

Uma hierarquia de generalização do modelo OMT corresponde a uma categoria ISA no modelo ECR. As regras que determinam este mapeamento são similares às definidas anteriormente para a tradução de categorias ISA em hierarquias de generalização. A diferença apresentada está relacionada ao fato de que as regras devem ser realizadas em sentido contrário.

Uma hierarquia de generalização  $G$  entre a superclasse  $C$  e as subclasses  $C_1, C_2, \dots, C_n$  é mapeada em uma categoria ISA  $C\_ISA$  entre o tipo-entidade (ou categoria)  $C'$  e as categorias  $C'_1, C'_2, \dots, C'_n$ . O tipo-entidade (ou categoria)  $C'$  corresponde diretamente a  $C$ , enquanto que as categorias  $C'_i$  correspondem às subclasses  $C_i$ , de acordo com as regras definidas para classes, atributos e operações.

Se as subclasses de  $G$  se sobrepõem, ou seja, o triângulo que representa  $G$  é pintado, então  $C\_ISA$  é não disjunta. Por outro lado, se as subclasses de  $G$  não se sobrepõem, ou seja, o triângulo que representa  $G$  não é pintado,  $C\_ISA$  é disjunta. Neste caso, o discriminador de  $G$  deve ser declarado como um atributo adicional de mesmo nome no tipo-entidade (ou categoria)  $C'$ . Este atributo adicional é definido como simples,

monovalorado e total. Já o seu tipo de dado é o mesmo do discriminador e deve ser inserido no dicionário de dados.

As regras a seguir formalizam as observações anteriores:

para cada hierarquia de generalização  $G$  do modelo OMT

entre a superclasse  $C$

e as subclasses  $C_1, C_2, \dots, C_n$

crie uma categoria ISA no modelo ECR da seguinte forma:

para a superclasse  $C$

crie um tipo-entidade de acordo com as regras  
de mapeamento de classe, atributo e operação

para cada uma das subclasses  $C_i$

crie uma categoria de acordo com as regras  
de mapeamento de classe, atributo e operação

se o triângulo que representa  $G$  é pintado

então a categoria ISA é não disjunta

senão a categoria ISA é disjunta

se o discriminador  $D$  de tipo de dado TDD foi especificado em  $G$

então declare  $D$  como um atributo do tipo-entidade (ou categoria)  $C$

{ simples, monovalorado e parcial }

insira o tipo de dado TDD no dicionário de dados

#### 4.2.2.3 Associação Binária<sup>3</sup> e Atributo de Ligação

As regras que determinam o mapeamento de associações binárias são similares às definidas anteriormente para a tradução de relacionamentos em associações. Em outras palavras, uma associação binária é mapeada em um relacionamento binário, um nome de papel corresponde a um nome de conexão com sua posição invertida e a multiplicidade da associação determina a cardinalidade do relacionamento. As diferenças apresentadas estão relacionadas ao fato de que as regras devem ser realizadas em sentido contrário e de que os atributos de ligação serão mapeados juntamente com as associações nas quais estão localizados. Em adição, também será considerado a tradução da restrição ordenação.

<sup>3</sup>Desconsiderar associações com qualificadores e com associações como uma classe, as quais possuirão regras específicas.



Sejam  $C_1$  e  $C_2$  duas classes no modelo OMT. Sejam  $C_1'$  e  $C_2'$  os dois tipos-entidade (ou categorias) respectivamente correspondentes no modelo ECR. Uma associação  $A$  entre  $C_1$  e  $C_2$  é mapeada em um relacionamento  $A'$  de mesmo nome entre  $C_1'$  e  $C_2'$ .

A cardinalidade de  $A'$  é determinada a partir da multiplicidade de  $A$ . No modelo OMT, a multiplicidade é especificada através de símbolos localizados no final das linhas que representam as associações. Estes símbolos indicam, implicitamente, o número mínimo e máximo de instâncias de uma classe que podem estar relacionadas a uma única instância da classe associada, denominados respectivamente  $N_{\min}$  e  $N_{\max}$ . Por exemplo, se um pequeno círculo vazio é desenhado próximo à classe  $C_2$ , então cada instância da classe  $C_1$  pode estar relacionada a no mínimo zero e no máximo uma instância da classe  $C_2$ . Em outras palavras,  $N_{\min} = 0$  e  $N_{\max} = 1$  para a classe  $C_1$ . A mesma consideração é válida para quando a multiplicidade é especificada com números ou conjuntos de intervalos.

Uma vez determinados os valores de  $N_{\min}$  e  $N_{\max}$  para uma classe  $C_i$ , estes podem ser associados respectivamente ao par de números  $(i_1, i_2)$  que representa a participação do tipo-entidade (ou categoria)  $C_i'$  no relacionamento  $A'$ . Por exemplo, se o valor assumido por  $i_1$  for 0, então a participação de  $C_1'$  no relacionamento é parcial. Por outro lado, se o valor assumido por  $i_2$  for 1, então a participação de  $C_1'$  no relacionamento é funcional.

Em adição, a modelagem OMT permite a especificação da restrição ordenação próxima à classe  $C_i$ . No mapeamento para o ECR, este conceito deve ser traduzido em uma restrição externa de que todos os elementos da extremidade muitos (relativos ao tipo-entidade ou categoria  $C_i'$ ) do relacionamento têm uma classificação explícita que deve ser preservada.

Considere agora que a associação  $A$  possui um atributo de ligação, denominado  $AL$ . Este atributo de ligação é mapeado em um atributo de mesmo nome no relacionamento  $A'$ , denominado  $AL'$ . O tipo de dado de  $AL'$  é o mesmo de  $AL$  e deve ser inserido no dicionário de dados. Se alguma restrição foi especificada no modelo OMT indicando que  $AL$  pode assumir valores nulos, então  $AL'$  é definido como simples, monovalorado e parcial. Caso contrário, o atributo  $AL'$  é definido como simples, monovalorado e total. O valor "default" de  $AL$  deve ser associado a  $AL'$  todas as vezes que uma instância do relacionamento  $A'$  for inserida no banco de dados.

Finalmente, um nome de papel no modelo OMT corresponde diretamente a um nome de conexão no modelo ECR. Os nomes assumidos por ambos os conceitos são os mesmos, porém as suas localizações são invertidas. Por exemplo, se o nome de papel  $papel\_1$  foi representado próximo à classe  $C_1$ , então o respectivo nome de conexão deve ser representado próximo ao tipo entidade (ou categoria)  $C_2'$ . Similarmente, se o nome de papel  $papel\_2$  foi representado próximo à classe  $C_2$ , então o respectivo nome de conexão deve ser representado próximo a  $C_1'$ .

As regras a seguir formalizam as observações anteriores:

para cada associação  $A$  do modelo OMT

que relaciona as classes  $C_1$  e  $C_2$

com possíveis atributos  $At_i$  de tipos de dado  $TD_i$  e valores "default"  $VD_i$  ( $i=1,\dots,n$ )

crie um relacionamento no modelo ECR

entre os tipos-entidade (ou categorias)  $C_1$  e  $C_2$

com o mesmo nome da associação

determine os valores de  $(i_1, i_2)$  para  $C_1$

se  $C_1$  está associada a zero ou uma instância de  $C_2$

então  $i_1 = 0$  e  $i_2 = 1$

se  $C_1$  está associada a exatamente uma instância de  $C_2$

então  $i_1 = 1$  e  $i_2 = 1$

se  $C_1$  está associada a zero ou mais instâncias de  $C_2$

então  $i_1 = 0$  e  $i_2 = N$

se  $C_1$  está associada a uma ou mais instâncias de  $C_2$

então  $i_1 = 1$  e  $i_2 = N$

se a multiplicidade foi especificada com números ou conjuntos de intervalos

então determine  $i_1$  e  $i_2$  de acordo com estes valores

se a restrição de ordenação foi indicada

então especifique restrição externa para os elementos da extremidade muitos

determine os valores de  $(i_1, i_2)$  para  $C_2$ , realizando o mesmo processo acima

se a associação possui atributos

então para cada atributo  $At_i$

crie um atributo de mesmo nome no relacionamento

insira o tipo de dado  $TD_i$  no dicionário de dados

se não existe restrição de que  $At_i$  pode assumir valores nulos

então declare-o como simples, monovalorado e total

senão declare-o como simples, monovalorado e parcial

se existem nomes de papéis representados na associação  $A$

então represente-os no relacionamento, invertendo suas localizações e mantendo os seus nomes

#### 4.2.2.4 Associação como uma Classe

Uma associação como uma classe pode ser mapeada em dois conceitos diferentes no modelo ECR, dependendo se este elemento possui ou não uma ligação com outra classe. Se a associação como uma classe não possui ligação com outra classe, ela será mapeada como um atributo composto. Por outro lado, se a associação como uma classe possui ligação com uma outra classe, ela será mapeada em um tipo-entidade (ou categoria).

Uma associação como uma classe  $AC$  do modelo OMT que não possui ligação com outra classe é mapeada em um atributo composto de mesmo nome no modelo ECR, denominado  $AC'$ . Este atributo  $AC'$  é definido como monovalorado e total. No entanto, caso tenha sido especificada no modelo OMT alguma restrição indicando que  $AC$  pode assumir valores nulos, então a cardinalidade de  $AC'$  será parcial ao invés de total.

Os atributos que compõem  $AC'$  correspondem diretamente aos atributos de  $AC$  e possuem os mesmos nomes e os mesmos tipos de dados que estes. Estes tipos de dados devem ser inseridos no dicionário de dados. Os valores "default" atribuídos aos atributos de  $AC$  devem ser associados aos atributos de  $AC'$  todas as vezes que uma instância do relacionamento for inserida no banco de dados.

Uma associação como uma classe  $ACL$  do modelo OMT que possui ligação com outra classe implica na criação de um novo tipo-entidade (ou categoria) de mesmo nome no modelo ECR, denominada  $ACL'$ . Considere que  $ACL$  está declarada na associação  $A$ , a qual relaciona as classes  $C_1$  e  $C_2$ . Considere também que o mapeamento de  $A$  já foi realizado, originando um relacionamento  $A'$  entre os tipos-entidade (ou categorias)  $C_1'$  e  $C_2'$ . O novo tipo-entidade,  $ACL'$ , deve estar associado ao relacionamento  $A'$ . Desta forma, se  $A$  é uma associação binária, então  $A'$  será um relacionamento ternário. A cardinalidade de  $ACL'$  será 1.

Os atributos de  $ACL$  correspondem diretamente aos atributos do tipo-entidade  $ACL'$ , os quais possuem os mesmos nomes e os mesmos tipos de dados que estes. Estes tipos de dados devem ser inseridos no dicionário de dados. Os valores "default" atribuídos aos atributos de  $ACL$  devem ser associados aos atributos de  $ACL'$  todas as vezes que uma entidade for inserida em  $ACL'$ .

Em ambas as situações, qualquer operação  $O$  definida para a associação como uma classe deve ser mapeada em um cabeçalho de procedimento que possui o mesmo nome, a mesma lista de argumentos e o mesmo tipo resultante que esta operação. O método que implementa  $O$  corresponde à implementação do procedimento em questão, com as devidas alterações realizadas.

As regras a seguir formalizam as observações anteriores:

para cada associação como uma classe AC do modelo OMT  
com atributos  $A_i$ , tipos de dado  $TD_i$  e valores “default”  $VD_i$  ( $i=1, \dots, n$ )  
com operações  $O_j$ , lista de argumentos  $LA_j$ , tipo resultante  $TR_j$  e métodos  $M_j$ ,  $M_j$   
correspondendo à implementação de  $O_j$  ( $j=1, \dots, m$ )  
declarada na associação AS entre as classes  $C_1$  e  $C_2$

remova AC de AS e execute as regras de mapeamento para associações binárias

se AC não possui ligação com outra classe

então crie um atributo composto de mesmo nome no relacionamento AS

se não existe restrição de que AC pode assumir valores nulos

então declare o atributo composto como monovalorado e total

senão declare o atributo composto como monovalorado e parcial

para cada atributo  $A_i$

crie um atributo de mesmo nome para o atributo composto

insira o tipo de dado  $TD_i$  no dicionário de dados

se AC possui ligação com outra classe

então crie um tipo-entidade (ou categoria) de mesmo nome

associe este tipo-entidade (ou categoria) ao relacionamento AS

especifique sua cardinalidade como 1

para cada atributo  $A_i$

crie um atributo de mesmo nome para o tipo-entidade

insira o tipo de dado  $TD_i$  no dicionário de dados

para cada operação  $O_j$

defina um cabeçalho de procedimento

com lista de argumentos  $LA_j$

de tipo resultante  $TR_j$

implementado de acordo com o método  $M_j$

#### 4.2.2.5 Agregação

Uma agregação entre as classes  $C_1$  e  $C_2$  ( $C_1$  “é composta por”  $C_2$ ) no modelo OMT é mapeada para o modelo ECR da mesma forma que uma associação binária. Assim, as regras de mapeamento das agregações são as mesmas regras definidas anteriormente para as associações binárias. A multiplicidade de  $C_2$  poderá ser “um”, “zero ou um” ou “muitos”, entre outras. Já a multiplicidade de  $C_1$  será “exatamente um”.

Considere  $C_1'$  e  $C_2'$  os tipos-entidade (ou categorias) do esquema ECR correspondentes ao mapeamento das classes  $C_1$  e  $C_2$ . Pode ser que em um relacionamento de agregação ocorra a duplicação de métodos e atributos da classe  $C_1$  para a classe  $C_2$ . Neste caso, deve haver a duplicação dos respectivos procedimentos e atributos do tipo-entidade (ou categoria)  $C_1'$  na declaração de  $C_2'$ . Desta forma,  $C_2'$  possuirá, além dos seus próprios atributos e procedimentos, os atributos e procedimentos que foram duplicados de  $C_1'$ , com as devidas alterações realizadas.

As regras a seguir formalizam as observações anteriores:

para cada agregação  $AG$  do modelo OMT  
entre as classes  $C_1$  e  $C_2$  ( $C_1$  "é composta por"  $C_2$ )  
com possíveis atributos  $A_i$ , tipos de dado  $TD_i$  e valores "default"  $VD_i$  ( $i=1, \dots, n$ )  
a serem duplicados  
com possíveis operações  $O_j$ , lista de argumentos  $LA_j$ , tipo resultante  $TR_j$  e  
métodos  $M_j$ ,  $M_j$  correspondendo à implementação de  $O_j$  ( $j=1, \dots, m$ )  
a serem duplicados

remova a agregação, substituindo-a por uma associação

especifique a multiplicidade de  $C_1$  como "exatamente um"

realize as regras de mapeamento para associações binárias

se  $AG$  possui atributos a serem duplicados  
então estenda a declaração do tipo-entidade (ou categoria)  $C_2$  da seguinte forma:  
para cada atributo  $A_i$   
crie um atributo de mesmo nome para o tipo-entidade  
insira o tipo de dado  $TD_i$  no dicionário de dados

se  $AG$  possui operações a serem duplicadas  
então defina procedimentos para o tipo-entidade  $C_2$  da seguinte forma:  
para cada operação  $O_j$   
defina um cabeçalho de procedimento  
com lista de argumentos  $LA_j$   
de tipo resultante  $TR_j$   
implementado de acordo com o método  $M_j$

#### 4.2.2.6 Associação Qualificada

Sejam  $C_1$  e  $C_2$  duas classes no modelo OMT relacionadas entre si através de uma associação qualificada  $AQ$ . Seja  $Q$  o qualificador de  $AQ$  acoplado à classe  $C_1$ . Sejam  $C_1'$  e  $C_2'$  os tipos-entidade (ou categorias) respectivamente correspondentes no modelo ECR. O mapeamento de uma associação qualificada implica em duas alterações:

- a declaração do qualificador  $Q$  como um atributo do tipo-entidade (ou categoria)  $C_2'$ . Em outras palavras, este novo atributo possuirá o mesmo nome e o mesmo tipo de dado que  $Q$ . A sua cardinalidade será simples, monovalorado e total.
- o aumento da multiplicidade da classe  $C_2$ . Em outras palavras, caso a multiplicidade de  $C_2$  seja “um”, ela deve ser alterada para “muitos”.

Em seguida, deve ser realizado o mapeamento para  $AQ$ , o qual segue as regras anteriormente definidas para as associações binárias.

As regras a seguir formalizam as observações anteriores:

para cada associação qualificada  $AQ$  do modelo OMT  
que relaciona as classes  $C_1$  e  $C_2$   
e que possui o qualificador  $Q$  de tipo de dado  $TDQ$  acoplado a  $C_1$

estenda a declaração do tipo-entidade (ou categoria)  $C_2$  do modelo ECR  
declare  $Q$  como um atributo simples, monovalorado e total de  $C_2$   
insira o seu tipo de dado  $TDQ$  no dicionário de dados

se a multiplicidade de  $C_2$  é “um”  
então aumente-a para “muitos”

remova  $Q$  e execute a regra de mapeamento de associações binárias para  $AQ$

#### 4.2.2.7 Associação Ternária e de Ordem Superior

O mapeamento de associações ternárias e de ordem superior para o modelo ECR é realizado de maneira análoga ao mapeamento dos relacionamentos binários. Desta forma, generalizando a discussão previamente apresentada, o mapeamento de qualquer associação  $A$  entre as classes  $C_i$  ( $i \geq 1$ ) gera um relacionamento  $A'$  entre os tipos-entidade (ou

categorias)  $C_i'$ ,  $C_i'$  correspondendo ao mapeamento da classe  $C_i$ . O nome do relacionamento é o mesmo nome da associação.

Ao invés de representar a multiplicidade das classes que participam de uma associação ternária ou de ordem superior através de símbolos, a metodologia OMT utiliza o conceito de chaves candidatas. Neste conceito, são escolhidas para chaves candidatas de uma associação as classes do lado “muitos”. Desta forma, se  $C_i$  representa um dos atributos que compõem a chave candidata para  $A$ , então sua cardinalidade será  $N$ . Caso contrário, ou seja,  $C_i$  não representa um dos atributos que compõem a chave candidata para  $A$ , então sua cardinalidade será 1.

Em adição, a associação  $A$  pode possuir um ou mais atributos de ligação. As regras de mapeamento para a tradução de um atributo de ligação em um atributo de relacionamento foram explicadas detalhadamente na seção 4.2.2.3.

As regras a seguir formalizam as observações anteriores:

para cada associação ternária ou de ordem superior  $A$  do modelo OMT  
que relaciona as classes  $C_1, C_2, \dots, C_n$  ( $n \geq 3$ )  
com possíveis atributos  $At_i$  de tipos de dado  $TD_i$  e valores “default”  $VD_i$  ( $i=1, \dots, n$ )

crie um relacionamento no modelo ECR  
entre os tipos-entidade (ou categorias)  $C_1, C_2, \dots, C_n$   
com o mesmo nome da associação

para cada classe  $C_i$  que compõe a chave candidata de  $A$   
especifique a cardinalidade do tipo-entidade (ou categoria)  $C_i$  como  $N$

para cada classe  $C_i$  que não compõe a chave candidata de  $A$   
especifique a cardinalidade do tipo-entidade (ou categoria)  $C_i$  como 1

se a associação possui atributos  
então para cada atributo  $At_i$   
crie um atributo de mesmo nome no relacionamento  
insira o tipo de dado  $TD_i$  no dicionário de dados  
se não existe restrição de que  $At_i$  pode assumir valores nulos  
então declare-o como simples, monovalorado e total  
senão declare-o como simples, monovalorado e parcial

### 4.2.2.8 Instanciação

Uma instanciação estabelece um relacionamento entre uma classe e suas instâncias. Seja  $C$  a classe do modelo OMT relacionada às instâncias  $l_1, l_2, \dots, l_n$ . Seja  $C'$  o tipo-entidade (ou categoria) correspondente ao mapeamento da classe  $C$ . Uma instância  $l$  é mapeada em uma entidade  $E$  membro do tipo-entidade (ou categoria)  $C'$ . Os valores assumidos pelos atributos de  $E$  devem ser os valores determinados na instanciação. Em adição, a entidade  $E$  deve ser inserida no banco de dados.

As regras a seguir formalizam as observações anteriores:

para cada instância  $l$  do modelo OMT  
relacionada à classe  $C$   
que determina o valor  $V_i$  para o atributo  $A_i$  de  $C$  ( $i=1, \dots, n$ )

crie uma entidade de mesmo nome no modelo ECR  
membro do tipo-entidade ou categoria  $C$

para cada valor  $V_i$   
associe este valor ao atributo  $A_i$  da entidade

insira a entidade no banco de dados

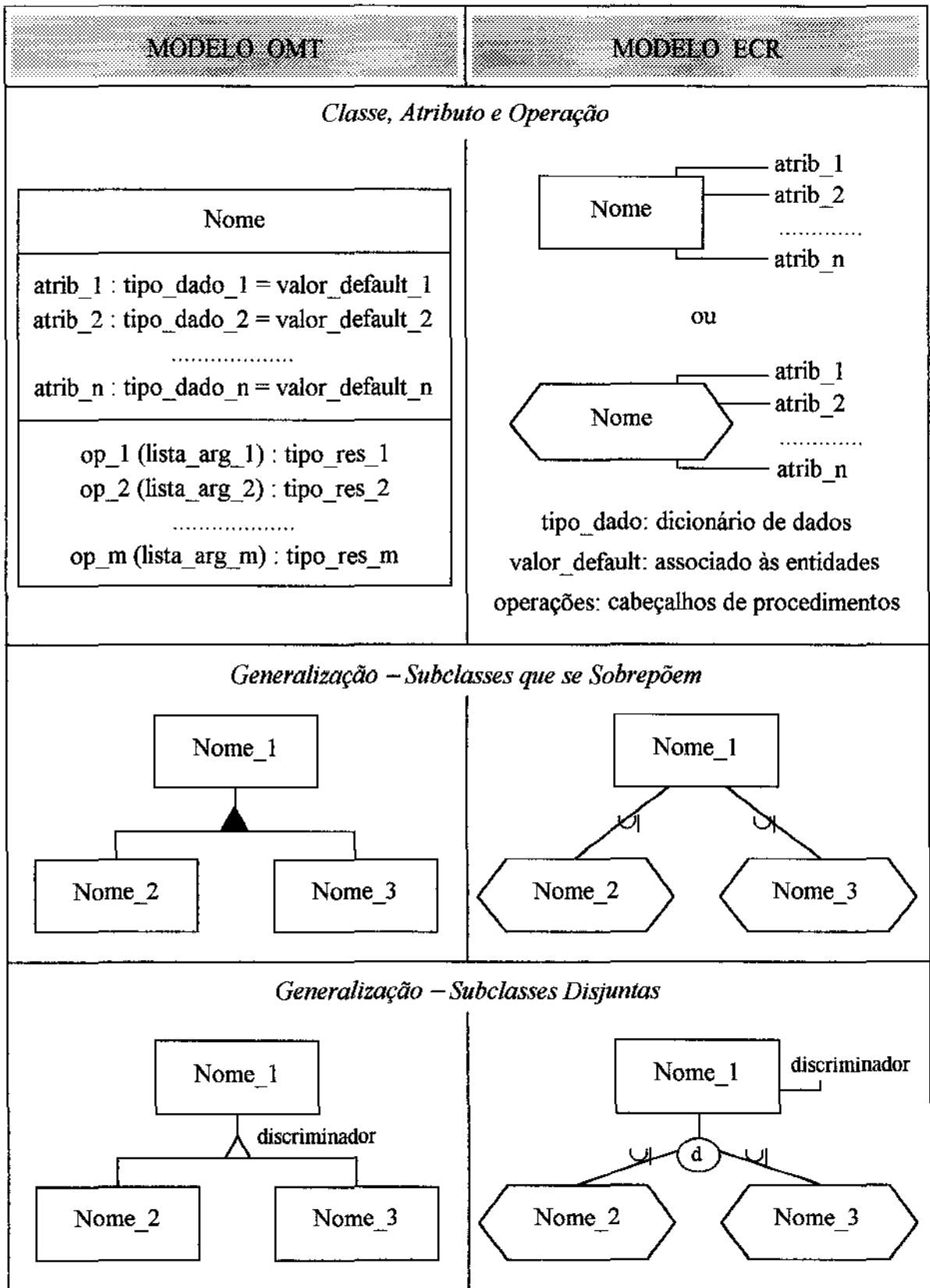
### 4.2.3 Modelo OMT $\rightarrow$ Modelo de Dados do Sistema $O_2$ <sup>4</sup>

O projeto de um BD utilizando o sistema  $O_2$  [Deu+92] pode ser dividido em duas fases: a fase de definição do esquema e a fase de implementação dos métodos. Durante a primeira fase, várias definições de classes são criadas. Estas definições de classes descrevem a estrutura interna (tipo) e o comportamento (método) dos seus objetos. Já na segunda fase são implementados os métodos correspondentes às declarações previamente realizadas.

No processo de mapeamento proposto neste trabalho, os conceitos do modelo OMT são convertidos em uma série de declarações de classes. A implementação dos métodos apenas é indicada. Desta forma, a seguir é apresentada a metodologia para a tradução de esquemas OMT em esquemas  $O_2$  equivalentes. A seqüência de apresentação da metodologia sugere uma ordem de execução dos mapeamentos. No final desta seção, são apresentadas várias figuras que exemplificam as regras de mapeamento propostas (figuras 4.8 a 4.11).

<sup>4</sup>Qualquer conceito seguido por apóstrofe (') representa, nesta seção, um conceito do modelo  $O_2$ .





**Figura 4.5** Mapeamento OMT → ECR (parte 1)

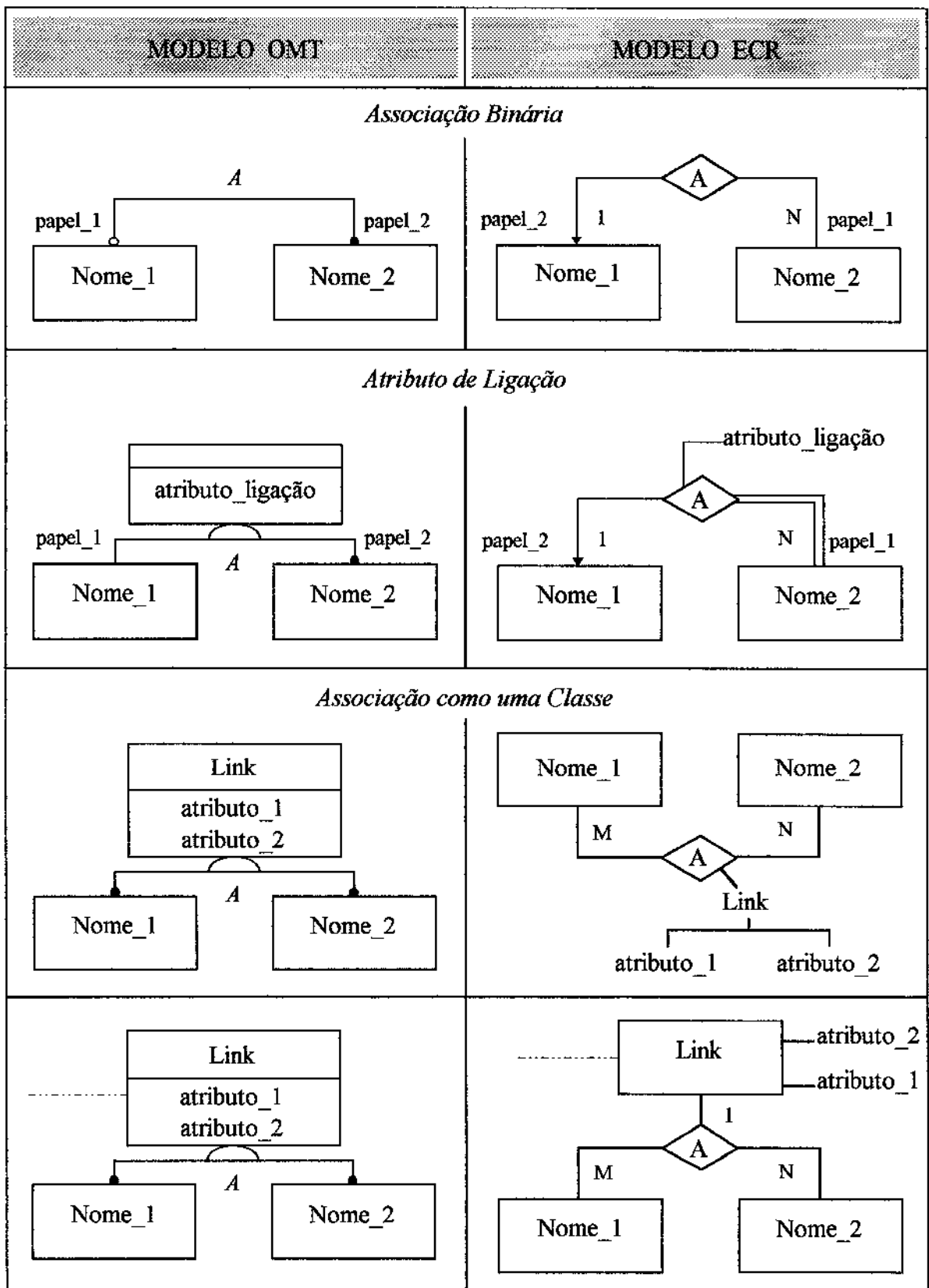
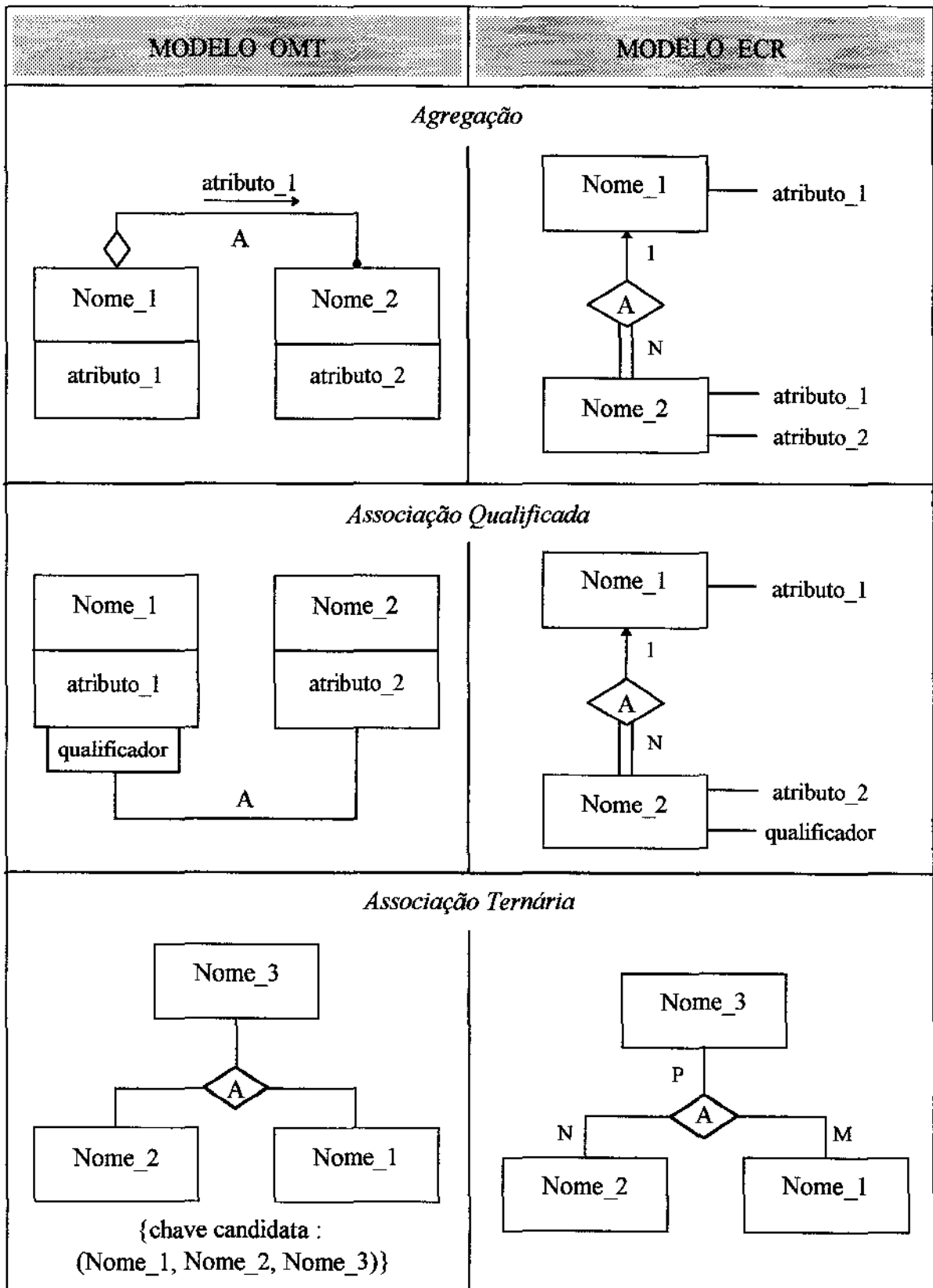


Figura 4.6 Mapeamento OMT → ECR (parte 2)



**Figura 4.7** Mapeamento OMT → ECR (parte 3)

### 4.2.3.1 Classe, Atributo e Operação

Uma classe  $C$  do modelo OMT gera uma definição de classe de mesmo nome no modelo  $O_2$ , denominada  $C'$ . Caso os objetos pertencentes a esta nova classe sejam persistentes, ela deve ser declarada utilizando-se a cláusula “with extension”. É tarefa do projetista ou do administrador do banco de dados determinar se os objetos são persistentes ou não.

Um atributo  $A$  da classe  $C$  corresponde diretamente a um atributo de mesmo nome na classe  $C'$ , denominado  $A'$ . O tipo de dado do atributo também será o mesmo. Já o valor “default” definido para  $A$  deve ser associado ao atributo  $A'$  através de um método. Caso o valor “default” de  $A$  não tenha sido especificado, um valor “default” definido pelo sistema  $O_2$  pode ser associado a  $A'$ .

Uma operação  $O$  da classe  $C$  gera uma declaração de método de mesmo nome na classe  $C'$ , denominado  $O'$ . A declaração do método  $O'$  possui a mesma lista de argumentos (nome e tipo dos argumentos) e o mesmo tipo resultante que a operação  $O$ . Existem duas formas de se declarar um método em  $O_2$ : junto com a declaração de classe ou individualmente após a declaração de todas as classes. No segundo caso, deve ser especificada a cláusula “in class nome da classe”, para que o método seja associado à sua respectiva classe. Este trabalho adota a primeira alternativa.

Um método  $M$  da classe  $C$  corresponde a uma implementação de método da classe  $C'$ , com as devidas alterações realizadas. A implementação de um método no modelo  $O_2$  pode ser feita logo após a sua declaração ou após a declaração de todas as classes. No segundo caso, tanto a assinatura do método quanto a cláusula “in class nome da classe” devem ser especificadas. Em ambos os casos, deve ser especificada também a cláusula “body”. Este trabalho adota a primeira alternativa.

As regras a seguir formalizam as observações anteriores:

para cada classe  $C$  do modelo OMT  
com atributos  $A_i$ , tipos de dado  $TD_i$  e valores “default”  $VD_i$  ( $i=1, \dots, n$ )  
com operações  $O_j$ , lista de argumentos  $LA_j$  e tipo resultante  $TR_j$  ( $j=1, \dots, m$ )  
e com métodos  $M_j$ ,  $M_j$  correspondendo à implementação da operação  $O_j$

crie uma classe de mesmo nome no modelo  $O_2$  da seguinte forma:

```
add class C {with extension}
  type tuple ( A1 : TD1,
              A2 : TD2,
              .....
              An : TDn )
```

*continua*

*continuação*

```
method  $O_1 ( LA_1 ) : TR_1$   
    body (  $M_1$  )  
method  $O_2 ( LA_2 ) : TR_2$   
    body (  $M_2$  )  
.....  
method  $O_m ( LA_m ) : TR_m$   
    body (  $M_m$  )  
method assign-default  
    /* associa aos atributos seus valores "default" */  
end;
```

### 4.2.3.2 Generalização

Uma generalização  $G$  entre a superclasse  $C$  e as subclasses  $C_1, C_2, \dots, C_n$  é mapeada em  $n+1$  definições de classe no modelo  $O_2$ , representadas respectivamente nesta seção por  $C', C'_1, C'_2, \dots, C'_n$ .

O mapeamento da superclasse  $C$  na classe  $C'$  segue as mesmas regras definidas anteriormente para as classes. Em adição, pode ser que um discriminador do tipo enumerado tenha sido especificado no modelo OMT. Neste caso, o discriminador será um atributo adicional da classe  $C'$ . Este atributo adicional possuirá o mesmo nome e o mesmo tipo de dado que o discriminador. Desta forma, sempre que um objeto pertencente a uma subclasse for criado, o valor do discriminador deve ser automaticamente atribuído.

O mapeamento de cada subclasse  $C_i$  na classe  $C'_i$  segue as mesmas regras definidas anteriormente para as classes. Em adição, deve ser especificada a cláusula "inherits nome da superclasse", para estabelecer que a classe  $C'_i$  herda as propriedades (atributos e métodos) da superclasse em questão.

Uma hierarquia de generalização no modelo OMT pode ser disjunta ou não, dependendo se as suas subclasses são disjuntas ou se sobrepõem, respectivamente. A única maneira de se garantir esta restrição no modelo  $O_2$  é através de métodos.

As regras a seguir formalizam as observações anteriores:

para cada hierarquia de generalização  $G$  do modelo OMT

entre a superclasse  $C$

e as subclasses  $C_1, C_2, \dots, C_n$

com discriminador  $D$  de tipo de dado TDD

crie  $n + 1$  classes de mesmo nome no modelo  $O_2$  da seguinte forma:

para a superclasse  $C$

com atributos  $A_l$ , tipos de dado  $TD_l$  e valores "default"  $VD_l$  ( $l=1, \dots, k$ )

com operações  $O_j$ , lista de argumentos  $LA_j$  e tipo resultante  $TR_j$  ( $j=1, \dots, m$ )

e com métodos  $M_j, M_j$  correspondendo à implementação da operação  $O_j$

crie uma classe de mesmo nome no modelo  $O_2$  da seguinte forma:

```
add class  $C$  {with extension}
```

```
type tuple (  $A_1 : TD_1,$   
               $A_2 : TD_2,$   
              .....,  
               $A_k : TD_k,$   
               $D : TDD$  )
```

```
method  $O_1$  ( $LA_1$ ) :  $TR_1$   
      body ( $M_1$ )
```

```
.....  
method  $O_m$  ( $LA_m$ ) :  $TR_m$   
      body ( $M_m$ )
```

```
method assign-default  
      /* associa aos atributos seus valores "default" */
```

```
end;
```

para cada subclasse  $C_i$

com atributos  $A_x$ , tipos de dado  $TD_x$  e valores "default"  $VD_x$  ( $x=1, \dots, p$ )

com operações  $O_y$ , lista de argumentos  $LA_y$  e tipo resultante  $TR_y$  ( $y=1, \dots, q$ )

e com métodos  $M_y, M_y$  correspondendo à implementação da operação  $O_y$

crie uma classe de mesmo nome no modelo  $O_2$  da seguinte forma:

```
add class  $C_i$  inherits  $C$ 
```

```
type tuple (  $A_1 : TD_1,$   
               $A_2 : TD_2,$   
              .....,  
               $A_p : TD_p$  )
```

```
method  $O_1$  ( $LA_1$ ) :  $TR_1$   
      body ( $M_1$ )
```

*continua*

```

method  $O_2 ( LA_2 ) : TR_2$ 
  body (  $M_2$  )
.....
method  $O_q ( LA_q ) : TR_q$ 
  body (  $M_q$  )
method assign-default
  /* associa aos atributos seus valores "default" */

end;

```

#### 4.2.3.3 Associação Binária<sup>5</sup> e Atributo de Ligação

Em  $O_2$ , as associações entre as classes são representadas através de relacionamentos de herança e/ou composição (agrupamento de objetos menores dentro de maiores através de construtores – set, list e tuple). O construtor a ser utilizado será determinado pela multiplicidade da associação no modelo OMT.

Sejam  $C_1$  e  $C_2$  duas classes no modelo OMT relacionadas entre si através de uma associação  $AS$ . Sejam  $C_1'$  e  $C_2'$  as duas classes respectivamente correspondentes no modelo  $O_2$ . O mapeamento da associação  $AS$  implica na inserção de um atributo adicional na classe  $C_1'$ , denominado  $AA'$ . Dependendo da multiplicidade de  $AS$ , do fato de  $AS$  possuir atributos e da especificação da restrição ordenação, diferentes situações podem ocorrer.

Nas três situações descritas a seguir,  $AS$  não possui atributos. O nome do atributo  $AA'$  será formado pela concatenação do string "ref\_" com o nome da classe  $C_2$ , uma vez que este será constituído por um atributo simples que referencia a classe  $C_2'$ . (**caso 1**)

- $AS$  representa uma associação na qual a classe  $C_1$  está associada a uma única instância de  $C_2$  ou a zero ou uma instância de  $C_2$ . Neste caso,  $AA'$  constitui em uma referência a esta classe.
- $AS$  representa uma associação na qual a classe  $C_1$  está associada a várias instâncias de  $C_2$  e a restrição de ordenação não foi especificada. Neste caso,  $AA'$  constitui em uma referência a um conjunto de instâncias desta classe. Em outras palavras, deve ser utilizado o construtor *set*.

<sup>5</sup> Desconsiderar associações com qualificadores e com associações como uma classe, as quais possuirão regras específicas.

- AS representa uma associação na qual a classe  $C_1$  está associada a várias instâncias de  $C_2$  e a restrição de ordenação foi especificada. Neste caso, AA' constitui em uma referência a um conjunto ordenado de instâncias desta classe. Em outras palavras, deve ser utilizado o construtor `list`.

Nas três situações descritas a seguir, AS possui atributos. O atributo AA' será constituído a partir de uma tupla. O nome desta tupla será formado pela concatenação do string "tupla\_" com o nome da classe  $C_2$ . Já o atributo que referencia a classe  $C_2'$  terá como nome a concatenação do string "ref\_" com o nome da classe  $C_2$ . Considere  $A_i$  um conjunto de atributos de AS que possuem tipos de dado  $TD_i$ . **(caso 2)**

- AS representa uma associação na qual a classe  $C_1$  está associada a uma única instância de  $C_2$  ou a zero ou uma instância de  $C_2$ . Neste caso, AA' é declarado como uma tupla de  $i+1$  outros atributos. Os primeiros  $i$  atributos correspondem aos atributos de AS e possuem o mesmo nome e o mesmo tipo de dado que estes. Já o último é um atributo adicional que referencia a classe  $C_2'$ .
- AS representa uma associação na qual a classe  $C_1$  está associada a várias instâncias de  $C_2$  e a restrição de ordenação não foi especificada. Neste caso, AA' é declarado como um conjunto de tuplas de  $i+1$  outros atributos. Os primeiros  $i$  atributos correspondem aos atributos de AS e possuem o mesmo nome e o mesmo tipo de dado que estes. Já o último é um atributo adicional que referencia a classe  $C_2'$ . Em outras palavras, deve ser utilizado o construtor `set`.
- AS representa uma associação na qual a classe  $C_1$  está associada a várias instâncias de  $C_2$  e a restrição de ordenação foi especificada. Neste caso, AA' é declarado como um conjunto ordenado de tuplas de  $i+1$  outros atributos. Os primeiros  $i$  atributos correspondem aos atributos de AS e possuem o mesmo nome e o mesmo tipo de dado que estes. Já o último é um atributo adicional que referencia a classe  $C_2'$ . Em outras palavras, deve ser utilizado o construtor `list`.

Em ambas as situações, caso seja permitido que AA' possua valor nulo, isto deve ser garantido através dos métodos de  $C_1'$ . Similarmente, qualquer especificação de que uma instância de  $C_1'$  está associada a um número restrito de instâncias de  $C_2'$  também deve ser garantida através de métodos.

A regra de determinação do nome do atributo AA' pode ser alterada se os nomes dos papéis que as classes  $C_1$  e  $C_2$  desempenham na associação AS foram especificados no modelo OMT. Neste caso, o atributo que referencia a classe  $C_2'$  terá o nome do papel desempenhado por esta classe na associação.



As associações binárias no modelo OMT serão mapeadas para o modelo O<sub>2</sub> considerando seus dois sentidos. Em outras palavras, da mesma maneira como foi efetuado o mapeamento de AS considerando a classe C<sub>1</sub>, deve ser feito o mapeamento de AS considerando a classe C<sub>2</sub>. Esta abordagem apresenta como vantagem uma busca mais rápida. Por outro lado, quando a associação possui atributos, estes são mapeados em apenas uma das duas classes. Esta abordagem evita a redundância de dados. Se o projetista ou o administrador do banco de dados desejar manter os atributos da associação em ambas as classes, a integridade destes atributos deve ser garantida.

As regras a seguir formalizam as observações anteriores:

*/\* as declarações previamente realizadas são apresentadas em **itálico***  
*/\* as novas declarações são apresentadas em **negrito***

para cada associação AS do modelo OMT  
 que relaciona as classes C<sub>1</sub> e C<sub>2</sub>  
 e que não possui atributos

estenda a declaração da classe C<sub>1</sub> do modelo O<sub>2</sub> da seguinte forma:

se C<sub>1</sub> está associada a uma única instância de C<sub>2</sub> ou a zero ou uma instância de C<sub>2</sub>

então *add class* C<sub>1</sub>

*type tuple ( /\* atributos já declarados de C<sub>1</sub> \*/,  
                   **ref\_c2 : C<sub>2</sub>** )*

*/\* métodos já declarados de C<sub>1</sub> \*/*

*end;*

se C<sub>1</sub> está associada a várias instâncias de C<sub>2</sub> e não há restrição de ordenação

então *add class* C<sub>1</sub>

*type tuple ( /\* atributos já declarados de C<sub>1</sub> \*/,  
                   **ref\_c2 : set ( C<sub>2</sub> )** )*

*/\* métodos já declarados de C<sub>1</sub> \*/*

*end;*

se C<sub>1</sub> está associada a várias instâncias de C<sub>2</sub> e há restrição de ordenação

então *add class* C<sub>1</sub>

*type tuple ( /\* atributos já declarados de C<sub>1</sub> \*/,  
                   **ref\_c2 : list ( C<sub>2</sub> )** )*

*/\* métodos já declarados de C<sub>1</sub> \*/*

*end;*

estenda a declaração da classe C<sub>2</sub> do modelo O<sub>2</sub> utilizando as regras acima definidas  
 (regra válida apenas para associações binárias comuns)

*continua*

para cada associação AS do modelo OMT  
 que relaciona as classes  $C_1$  e  $C_2$   
 e que possui atributos  $A_i$  de tipos de dado  $TD_i$  ( $i=1, \dots, n$ )

estenda a declaração da classe  $C_1$  do modelo  $O_2$  da seguinte forma:

se  $C_1$  está associada a uma única instância de  $C_2$  ou a zero ou uma instância de  $C_2$   
 então add class  $C_1$

```

type tuple ( /* atributos já declarados de  $C_1$  */
            tupla_c2 : tuple (  $A_1$  :  $TD_1$ ,
                               .....,
                                $A_n$  :  $TD_n$ ,
                               ref_c2 :  $C_2$  ))
    
```

*/\* métodos já declarados de  $C_1$  \*/*

end;

se  $C_1$  está associada a várias instâncias de  $C_2$  e não há restrição de ordenação  
 então add class  $C_1$

```

type tuple ( /* atributos já declarados de  $C_1$  */
            tupla_c2 : set ( tuple (  $A_1$  :  $TD_1$ ,
                                    .....,
                                     $A_n$  :  $TD_n$ ,
                                    ref_c2 :  $C_2$  )) )
    
```

*/\* métodos já declarados de  $C_1$  \*/*

end;

se  $C_1$  está associada a várias instâncias de  $C_2$  e há restrição de ordenação  
 então add class  $C_1$

```

type tuple ( /* atributos já declarados de  $C_1$  */
            tupla_c2 : list ( tuple (  $A_1$  :  $TD_1$ ,
                                      .....,
                                       $A_n$  :  $TD_n$ ,
                                      ref_c2 :  $C_2$  )) )
    
```

*/\* métodos já declarados de  $C_1$  \*/*

end;

remova os atributos de AS e estenda a declaração da classe  $C_2$  do modelo  $O_2$   
 utilizando as regras definidas para associações sem atributos  
 (regra válida apenas para associações binárias comuns)

#### 4.2.3.4 Associação como uma Classe

O mapeamento de uma associação como uma classe implica em duas alterações no modelo  $O_2$ : a criação de uma nova classe e a inserção de um atributo adicional em uma das classes associadas, o qual referencia esta nova classe criada. A primeira alteração segue basicamente as mesmas regras definidas anteriormente para as classes. Já a segunda alteração é mapeada de maneira semelhante ao mapeamento dos atributos das associações.

Uma associação como uma classe  $AC$  do modelo OMT gera uma definição de classe de mesmo nome no modelo  $O_2$ , denominada  $AC'$ . Os atributos de  $AC$  correspondem diretamente aos atributos da classe  $AC'$ , os quais possuem os mesmos nomes e os mesmos tipos de dados que estes. Similarmente, as operações de  $AC$  correspondem diretamente às operações da classe  $AC'$ , as quais possuem as mesmas listas de argumentos e os mesmos tipos resultantes que estas.

Sejam  $C_1$  e  $C_2$  duas classes no modelo OMT relacionadas entre si através de uma associação  $AS$ . Sejam  $C_1'$  e  $C_2'$  as duas classes respectivamente correspondentes no modelo  $O_2$ . Considere também que o mapeamento das associações já foi realizado. Este mapeamento foi executado de acordo com o caso 1 ou o caso 2, conforme descrito anteriormente (seção 4.2.3.3). Desta forma, existem duas maneiras na qual a classe  $C_1'$  pode referenciar a nova classe  $AC'$ :

- se o mapeamento de  $AS$  foi realizado de acordo com o caso 1, remova temporariamente a declaração da associação e acrescente à classe  $C_1'$  um novo atributo, cujo nome é formado pela concatenação do string "tupla\_" com o nome da classe  $C_2$ . Este novo atributo pode ser declarado como uma tupla, um conjunto de tuplas ou um conjunto ordenado de tuplas, dependendo do atributo removido. O primeiro atributo da tupla referencia a classe  $C_2'$ . Já o segundo atributo possui como nome a concatenação do string "ref\_" com o nome de  $AC$  e constitui uma referência a esta classe.
- se o mapeamento de  $AS$  foi realizado de acordo com o caso 2, então a classe  $C_1'$  já possui um atributo adicional declarado como uma tupla. Neste caso, deve ser adicionado a esta tupla um outro atributo, o qual possui como nome a concatenação do string "ref\_" com o nome de  $AC$  e constitui uma referência a esta classe.

As regras a seguir formalizam as observações anteriores:

*/\* as declarações previamente realizadas são apresentadas em **itálico***

*/\* as novas declarações são apresentadas em **negrito***

para cada associação como uma classe AC do modelo OMT

com atributos  $A_i$ , tipos de dado  $TD_i$  e valores "default"  $VD_i$  ( $i=1, \dots, n$ )

com operações  $O_j$ , lista de argumentos  $LA_j$ , tipo resultante  $TR_j$  e métodos  $M_j, M_j$

correspondendo à implementação de  $O_j$  ( $j=1, \dots, m$ )

declarada na associação AS entre as classes  $C_1$  e  $C_2$

crie uma classe de nome AC utilizando a regra de mapeamento para classes

remova AC de AS e execute as regras de mapeamento para associações binárias  
(tanto para a classe  $C_1$  quanto para a classe  $C_2$ )

se AS não possui atributos

então remova a declaração da associação da classe  $C_1$  do modelo  $O_2$

se  $C_1$  está associada a uma ou a zero ou uma instância de  $C_2$

então add class  $C_1$

type tuple (*/\* atributos já declarados de  $C_2$  \*/*,

**tupla\_c2 : tuple ( ref\_c2 :  $C_2$ ,  
ref\_ac : AC ) )**

*/\* métodos já declarados de  $C_2$  \*/*

end;

se  $C_1$  está associada a várias instâncias de  $C_2$  e não há restrição ordenação

então add class  $C_1$

type tuple (*/\* atributos já declarados de  $C_2$  \*/*,

**tupla\_c2 : set ( tuple ( ref\_c2 :  $C_2$ ,  
ref\_ac : AC ) ) )**

*/\* métodos já declarados de  $C_2$  \*/*

end;

se  $C_1$  está associada a várias instâncias de  $C_2$  e há restrição ordenação

então add class  $C_1$

type tuple (*/\* atributos já declarados de  $C_2$  \*/*,

**tupla\_c2 : list ( tuple ( ref\_c2 :  $C_2$ ,  
ref\_ac : AC ) ) )**

*/\* métodos já declarados de  $C_2$  \*/*

end;

se AS possui atributos

então acrescente a declaração **ref\_ac : AC** na tupla do atributo **tupla\_c2**

### 4.2.3.5 Agregação

Uma agregação entre as classes  $C_1$  e  $C_2$  ( $C_1$  “é composta por”  $C_2$ ) no modelo OMT é mapeada para o modelo  $O_2$  da mesma forma que uma associação. Assim, as regras de mapeamento das agregações são as mesmas regras definidas anteriormente para as associações binárias. A multiplicidade de  $C_2$  poderá ser “um”, “zero ou um” ou “muitos”, entre outras. Uma diferença entre as regras de mapeamento das associações binárias e das agregações é que esta última apenas modela a associação em um sentido: do agregado para os seus componentes.

Considere  $C_1'$  e  $C_2'$  as classes do esquema  $O_2$  correspondentes ao mapeamento das classes  $C_1$  e  $C_2$ . Pode ser que em um relacionamento de agregação ocorra a duplicação de métodos e atributos da classe  $C_1$  para a classe  $C_2$ . Neste caso, deve haver a duplicação dos respectivos métodos e atributos da classe  $C_1'$  na declaração da classe  $C_2'$ . Desta forma,  $C_2'$  possuirá, além dos seus próprios atributos e métodos, os atributos e métodos que foram duplicados de  $C_1'$ , com as devidas alterações realizadas. Em adição, o método *assign-default* de  $C_2'$  deve ser estendido para associar os novos valores “default” aos novos atributos da classe.

As regras a seguir formalizam as observações anteriores:

*/\* as declarações previamente realizadas são apresentadas em **itálico***

*/\* as novas declarações são apresentadas em **negrito***

para cada agregação  $AG$  do modelo OMT

entre as classes  $C_1$  e  $C_2$  ( $C_1$  “é composta por”  $C_2$ )

com possíveis atributos  $A_i$ , tipos de dado  $TD_i$  e valores “default”  $VD_i$  ( $i=1, \dots, n$ )  
a serem duplicados

com possíveis operações  $O_j$ , lista de argumentos  $LA_j$ , tipo resultante  $TR_j$  e  
métodos  $M_j$ ,  $M_j$  correspondendo à implementação de  $O_j$  ( $j=1, \dots, m$ )  
a serem duplicados

remova a agregação, substituindo-a por uma associação

realize as regras de mapeamento para associações binárias (apenas para  $C_1$ )

se  $AG$  possui atributos e/ou métodos a serem duplicados

então estenda a declaração da classe  $C_2$  do modelo  $O_2$  da seguinte forma:

***add class**  $C_2$*

***type tuple** ( /\* atributos já declarados de  $C_2$  \*/,*

***$A_1$  :  $TD_1$ ,***

*continua*

```

A2 : TD2,
.....,
An : TDn )
/* métodos já declarados de C2 */
method O1 ( LA1 ) : TR1
    body ( M1 )
.....
method Om ( LAm ) : TRm
    body ( Mm )
method assign-default
    /* associa aos atributos seus valores "default" */
    /* associa aos novos atributos seus valores "default" */
end;

```

#### 4.2.3.6 Associação Qualificada

Existem duas diferentes alternativas de mapeamento para uma associação qualificada. A vantagem apresentada por uma alternativa é a desvantagem da outra, e vice versa. A primeira delas desconsidera a função da associação qualificada e restabelece a multiplicidade das classes associadas. A vantagem apresentada é um projeto mais limpo. A segunda, por sua vez, realiza o mapeamento propriamente dito. A vantagem apresentada é a possibilidade de utilização deste conceito no esquema O<sub>2</sub>, enquanto que a desvantagem está relacionada à duplicação de informações.

Sejam C<sub>1</sub> e C<sub>2</sub> duas classes no modelo OMT relacionadas entre si através de uma associação qualificada AQ. Seja Q o qualificador de AQ acoplado à classe C<sub>1</sub>. Sejam C<sub>1</sub>' e C<sub>2</sub>' as duas classes respectivamente correspondentes no modelo O<sub>2</sub>. De acordo com a decisão do projetista ou do administrador do banco de dados, duas situações podem ocorrer:

- se a primeira alternativa foi adotada, então o qualificador Q deve ser declarado como um atributo da classe C<sub>2</sub>'. Em adição, caso a multiplicidade de C<sub>2</sub>' seja "um", ela deve ser alterada para "muitos". O mapeamento de AQ segue as regras anteriormente definidas para as associações binárias.
- se a segunda alternativa foi adotada, então o qualificador Q deve ser declarado tanto como atributo da classe C<sub>1</sub>' quanto como atributo da classe C<sub>2</sub>'. Semelhantemente ao mapeamento das associações, a declaração de Q em C<sub>1</sub>' depende da multiplicidade de AQ, do fato de AQ possuir atributos e da especificação da restrição ordenação:

- se AQ não possui atributos, então um atributo adicional cujo nome é formado pela concatenação do string “tupla\_” com o nome da classe  $C_2$  é criado. Este atributo adicional é declarado como um conjunto de tuplas de dois outros atributos. O primeiro corresponde ao qualificador Q e possui o mesmo nome e o mesmo tipo de dado que este. Já o segundo atributo, cujo nome é formado pela concatenação do string “ref\_” com o nome da classe  $C_2$  (ref\_c2), referencia a classe associada. Se  $C_1$  está associada a uma ou a zero ou uma instância de  $C_2$ , então ref\_c2 apenas constitui uma referência a esta classe. Se  $C_1$  está associada a várias instâncias de  $C_2$  e a restrição ordenação não foi especificada, então ref\_c2 constitui uma referência a um conjunto de instâncias desta classe. Finalmente, se  $C_1$  está associada a várias instâncias de  $C_2$  e a restrição ordenação foi especificada, então ref\_c2 constitui uma referência a um conjunto ordenado de instâncias desta classe.
- se AQ possui atributos e  $C_1$  está associada a uma ou a zero ou uma instância de  $C_2$ , então um atributo adicional cujo nome é formado pela concatenação do string “tupla\_” com o nome da classe  $C_2$  é criado. Este atributo adicional é declarado como um conjunto de tuplas de vários outros atributos. Um deles corresponde ao qualificador Q e possui o mesmo nome e o mesmo tipo de dado que este. Outro corresponde à classe associada. Os demais representam os atributos da associação.
- Nos demais casos, um atributo adicional cujo nome é formado pela concatenação do string “tupla\_” com o nome da classe  $C_2$  também é criado. Este atributo é declarado como um conjunto de tuplas de dois outros atributos. O primeiro deles corresponde ao qualificador Q e possui o mesmo nome e o mesmo tipo de dado que este. O segundo é um outro atributo adicional, o qual é formado por um conjunto de tuplas dos atributos da associação e do atributo que referencia a classe associada.

As regras a seguir formalizam as observações anteriores:

#### **alternativa 1**

*/\* as declarações previamente realizadas são apresentadas em itálico*  
*/\* as novas declarações são apresentadas em **negrito***

para cada associação qualificada AQ do modelo OMT  
 que relaciona as classes  $C_1$  e  $C_2$   
 e que possui o qualificador Q de tipo de dados TDQ acoplado a  $C_1$

declare Q como um atributo de  $C_2$  do modelo O<sub>2</sub> da seguinte forma:

*add class  $C_2$*   
*type tuple (*/\* atributos já declarados de  $C_2$  \*/*)*

*continua*

```

Q : TDQ )
  /* métodos já declarados de C2 */
  end;

```

se a multiplicidade de  $C_2$  é “um”  
então aumente-a para “muitos”

remova  $Q$  e execute a regra de mapeamento de associações binárias para  $AQ$   
(tanto para a classe  $C_1$  quanto para a classe  $C_2$ )

## alternativa 2

*/\* as declarações previamente realizadas são apresentadas em itálico*  
*/\* as novas declarações são apresentadas em negrito*

para cada associação qualificada  $AQ$  do modelo OMT  
que relaciona as classes  $C_1$  e  $C_2$   
com o qualificador  $Q$  de tipo de dado TDQ acoplado a  $C_1$   
e que não possui atributos

declare  $Q$  como um atributo de  $C_2$  do modelo OMT da seguinte forma:

```

add class C2
  type tuple (/* atributos já declarados de C2 */
    Q : TDQ )
  /* métodos já declarados de C2 */
  end;

```

estenda a declaração da classe  $C_1$  do modelo OMT da seguinte forma:

se  $C_1$  está associada a uma única instância de  $C_2$  ou a zero ou uma instância de  $C_2$

```

então add class C1
  type tuple (/* atributos já declarados de C1 */,
    tupla_c2 : set ( tuple ( Q : TDQ,
      ref_c2 : C2 ) ) )
  /* métodos já declarados de C1 */
  end;

```

continua



continuação

se  $C_1$  está associada a várias instâncias de  $C_2$  e não há restrição de ordenação

então add class  $C_1$

```
type tuple ( /* atributos já declarados de  $C_1$  */,  
            tupla_c2 : set ( tuple ( Q : TDQ,  
                                    ref_c2 : set (  $C_2$  ) ) ) )
```

```
/* métodos já declarados de  $C_1$  */
```

end;

se  $C_1$  está associada a várias instâncias de  $C_2$  e há restrição de ordenação

então add class  $C_1$

```
type tuple ( /* atributos já declarados de  $C_1$  */,  
            tupla_c2 : set ( tuple ( Q : TDQ,  
                                    ref_c2 : list (  $C_2$  ) ) ) )
```

```
/* métodos já declarados de  $C_1$  */
```

end;

remova  $Q$  e execute a regra de mapeamento de associações binárias para a classe  $C_2$

para cada associação qualificada  $AQ$  do modelo OMT

que relaciona as classes  $C_1$  e  $C_2$

com o qualificador  $Q$  de tipo de dado TDQ acoplado a  $C_1$

e que possui atributos  $A_i$  de tipos de dado  $TD_i$  ( $i=1, \dots, n$ )

declare  $Q$  como um atributo de  $C_2$  do modelo OMT da seguinte forma:

add class  $C_2$

```
type tuple ( /* atributos já declarados de  $C_2$  */
```

```
            Q : TDQ )
```

```
/* métodos já declarados de  $C_2$  */
```

end;

estenda a declaração da classe  $C_1$  do modelo OMT da seguinte forma:

se  $C_1$  está associada a uma única instância de  $C_2$  ou a zero ou uma instância de  $C_2$

então add class  $C_1$

```
type tuple ( /* atributos já declarados de  $C_1$  */,  
            tupla_c2 : set ( tuple ( Q : TDQ,  
                                    ref_c2:  $C_2$ ,  
                                     $A_1$  :  $TD_1$ ,  
                                     $A_2$  :  $TD_2$ ,  
                                    .....  
                                     $A_n$  :  $TD_n$  ) ) )
```

continua

```

/* métodos já declarados de C1 */
end;

```

se C<sub>1</sub> está associada a várias instâncias de C<sub>2</sub> e não há restrição de ordenação  
então add class C<sub>1</sub>

```

type tuple ( /* atributos já declarados de C1 */
  tupla_c2 : set ( tuple ( Q : TDQ,
                        adicional : set ( tuple ( A1 : TD1,
                                                A2 : TD2,
                                                .....
                                                An : TDn,
                                                ref_c2: C2))))))

```

```

/* métodos já declarados de C1 */
end;

```

se C<sub>1</sub> está associada a várias instâncias de C<sub>2</sub> e há restrição de ordenação  
então add class C<sub>1</sub>

```

type tuple ( /* atributos já declarados de C1 */
  tupla_c2 : set ( tuple ( Q : TDQ,
                        adicional : list ( tuple ( A1 : TD1,
                                                  A2 : TD2,
                                                  .....
                                                  An : TDn,
                                                  ref_c2: C2))))))

```

```

/* métodos já declarados de C1 */
end;

```

remova Q e execute a regra de mapeamento de associações binárias para a classe C<sub>2</sub>

#### 4.2.3.7 Associação Ternária e de Ordem Superior

O mapeamento de uma associação ternária ou de ordem superior AOS do modelo OMT implica na criação de uma classe de mesmo nome no modelo O<sub>2</sub>, denominada AOS'. Considere que a associação em questão relaciona n classes e possui m atributos. A classe AOS' possui apenas um único atributo, o qual é declarado como um conjunto de tuplas de n+m outros atributos. Cada um dos n primeiros atributos constituem uma referência a uma classe que participa da associação. Já os m últimos atributos correspondem diretamente aos atributos da associação, e possuem os mesmos nomes e os mesmos tipos de dados que estes.

Todas as vezes que um determinado objeto for associado a alguma classe que participa de uma associação ternária ou de ordem superior, deve ser associado também o respectivo objeto na classe AOS. Em adição, também deve ser respeitada a multiplicidade de participação de cada classe na associação (representada tanto através de símbolos quanto através da identificação das chaves candidatas). Ambas situações devem ser garantidas através de métodos.

As regras a seguir formalizam as observações anteriores:

para cada associação ternária ou de ordem superior AOS do modelo OMT  
que relaciona as classes  $C_1, C_2, \dots, C_n$  ( $n \geq 3$ )  
e que possui os atributos  $A_1, A_2, \dots, A_m$  ( $m \geq 0$ )

crie uma classe de nome AOS no modelo  $O_2$  da seguinte forma:

```
add class AOS
  type atrib_adic: set ( tuple ( ref_c1 : C1,
                               ref_c2 : C2,
                               .....,
                               ref_cn : Cn,
                               A1 : TD1,
                               A2 : TD2,
                               .....,
                               Am : TDm ) )
```

#### 4.2.3.8 Instanciação

Uma instanciação no modelo OMT é mapeada em um objeto no modelo  $O_2$ . Existem duas diferentes formas de se realizar este mapeamento:

- através da execução dos métodos new e assign-default. O primeiro método tem por finalidade criar o objeto e associá-lo a sua respectiva classe. Já o segundo tem por finalidade associar ao objeto os seus valores "default".
- através da execução do método new e posterior declaração do objeto. Neste caso, durante a declaração do objeto, já são associados a ele os seus valores "default".

As regras a seguir formalizam as observações anteriores:

### alternativa 1

para cada instanciação  $I$  do modelo OMT  
que especifica a classe  $C$   
o objeto  $O$   
e os valores de seus atributos  $V_i$  ( $i=1, \dots, n$ )

crie um objeto  $O$  no modelo  $O_2$  da seguinte forma:

execute o método new:  $O = \text{new} ( C )$

execute o método assign-default de  $C$  para associar os valores  $V_i$  ao objeto  $O$

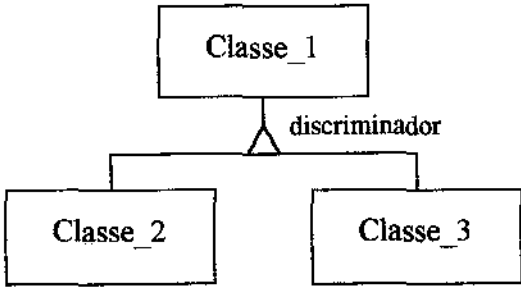
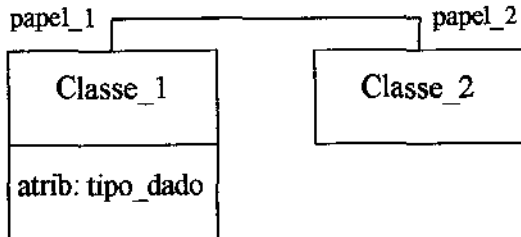
### alternativa 2

para cada instanciação  $I$  do modelo OMT  
que especifica a classe  $C$   
o objeto  $O$   
e os valores de seus atributos  $V_i$  ( $i=1, \dots, n$ )

crie um objeto  $O$  no modelo  $O_2$  da seguinte forma:

execute o método new:  $O = \text{new} ( C )$

declare o objeto:  $O : \text{tuple} ($   
     $\text{atrib}_1 : V_1,$   
     $\text{atrib}_2 : V_2,$   
    .....  
     $\text{atrib}_n : V_n )$

MODELO OMT	MODELO O <sub>2</sub>
<i>Classe, Atributo e Operação</i>	
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Nome_Classe</p> <hr/> <p>atrib_1 : tipo_dado_1 = valor_default_1  atrib_2 : tipo_dado_2 = valor_default_2  .....  atrib_n : tipo_dado_n = valor_default_n</p> <hr/> <p>op_1 (lista_arg_1) : tipo_res_1  op_2 (lista_arg_2) : tipo_res_2  .....  op_m (lista_arg_m) : tipo_res_m</p> </div>	<pre> add class Nome_Classe {with extension}   type tuple ( atrib_1 : tipo_dado_1,                atrib_2 : tipo_dado_2,                .....                atrib_n : tipo_dado_n )   method op_1 (lista_arg_1) : tipo_res_1   method op_2 (lista_arg_2) : tipo_res_2   .....   method op_m (lista_arg_m) : tipo_res_m   method assign-default end; </pre>
<i>Generalização</i>	
 <pre> classDiagram     Classe_1 &lt; -- Classe_2     Classe_1 &lt; -- Classe_3     </pre>	<pre> add class Classe_1 {with extension}   type tuple ( /* atributos                discriminador : ..... ) end;  add class Classe_2 inherits Classe_1 add class Classe_3 inherits Classe_1 </pre>
<i>Associação Binária</i> <i>(representado apenas o mapeamento referente a Classe_1)</i>	
 <pre> classDiagram     Classe_1 -- Classe_2 : papel_1, papel_2     class Classe_1 {         atrib: tipo_dado     }     </pre>	<pre> add class Classe_1 {with extension}   type tuple ( atrib: tipo_dado,                papel_2 : Classe_2 )   ..... end; </pre>

**Figura 4.8** Mapeamento OMT → O<sub>2</sub> (parte 1)

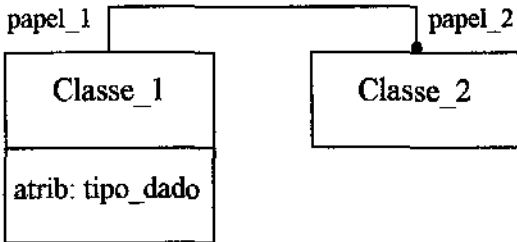
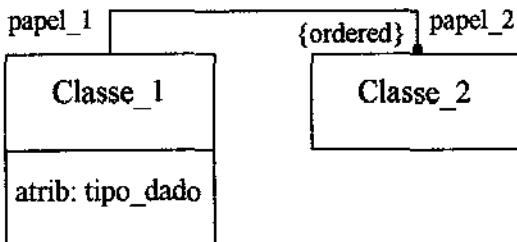
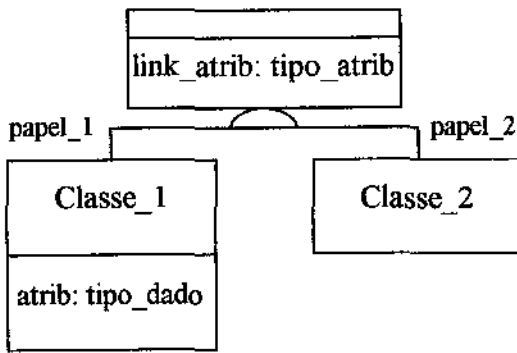
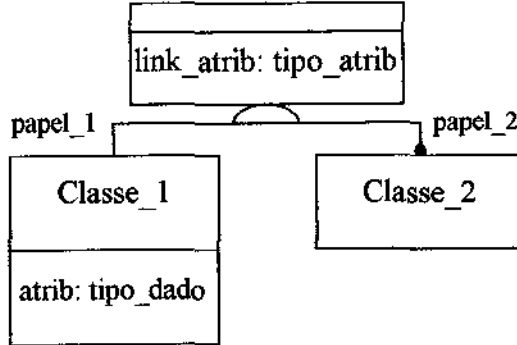
MODELO OMT	MODELO O <sub>2</sub>
<p><i>Associação Binária</i> (representado apenas o mapeamento referente a Classe_1)</p>	
	<pre> add class Classe_1 {with extension}   type tuple ( atrib : tipo_dado,                papel_2 : set ( Classe_2 ) )   ..... end; </pre>
	<pre> add class Classe_1 {with extension}   type tuple ( atrib : tipo_dado,                papel_2 : list ( Classe_2 ) )   ..... end; </pre>
	<pre> add class Classe_1 {with extension}   type tuple (     atrib : tipo_dado,     tupla_Classe_2 :       tuple ( link_atrib : tipo_atrib,               papel_2 : Classe_2 ) )   ..... end; </pre>
	<pre> add class Classe_1 {with extension}   type tuple (     atrib : tipo_dado,     tupla_Classe_2 : set (       tuple ( link_atrib : tipo_atrib,               papel_2 : Classe_2 )))   ..... end; </pre>

Figura 4.9 Mapeamento OMT → O<sub>2</sub> (parte 2)

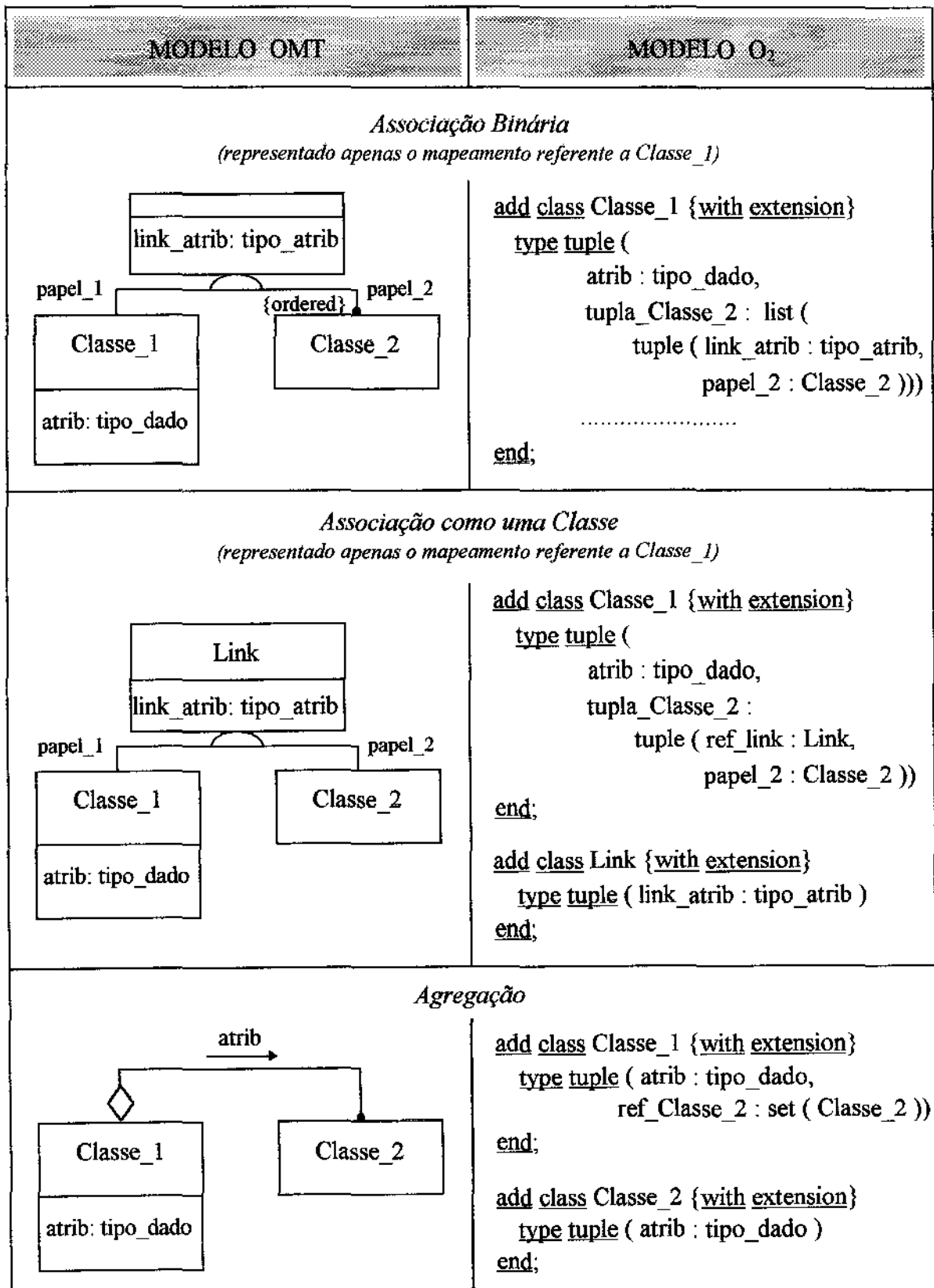


Figura 4.10 Mapeamento OMT → O<sub>2</sub> (parte 3)

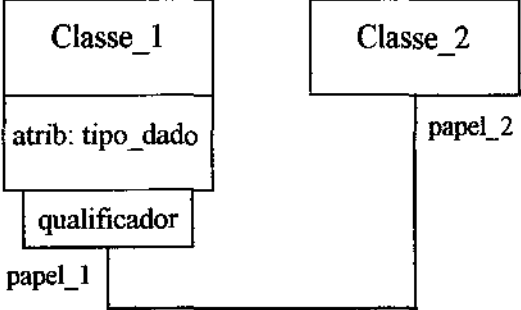
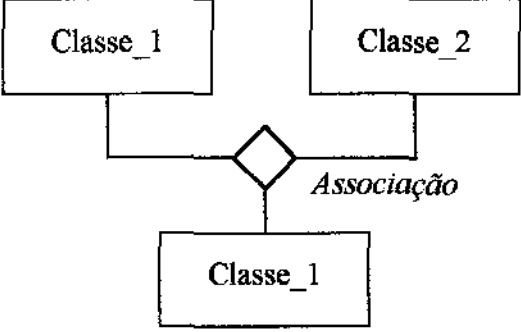
MODELO OMT	MODELO O <sub>2</sub>
<p style="text-align: center;"><i>Associação Qualificada</i></p> 	<pre> add class Classe_1 {with extension}   type tuple (     atrib : tipo_dado,     tupla_classe_2 : set (       tuple ( qualificador : ..... ,               papel_2 : Classe_2 ))) end;  add class Classe_2 {with extension}   type tuple ( qualificador: ..... )   ..... end; </pre>
<p style="text-align: center;"><i>Associação Ternária</i></p> 	<pre> add class Classe_1 {with extension} add class Classe_2 {with extension} add class Classe_3 {with extension} add class Associação {with extension}   type tuple ( atrib_adic :     set ( tuple (       ref_Classe_1 : Classe_1,       ref_Classe_2 : Classe_2,       ref_Classe_3 : Classe_3 ))) end; </pre>
<p style="text-align: center;"><i>Instanciação</i></p> <div style="border: 1px solid black; border-radius: 15px; padding: 10px; width: fit-content; margin: auto;"> <p><b>( nome_classe )</b>  nome_objeto  atrib_1 = valor_1  atrib_2 = valor_2</p> </div>	<pre> nome_objeto = new ( nome_classe )  /* execução do método assign-default de nome_classe para atribuir os valores aos atributos do objeto */ </pre>

Figura 4.11 Mapeamento OMT → O<sub>2</sub> (parte 4)



## Capítulo 5

# Descrição das Aplicações e Visão de Alto Nível

### 5.1 Aplicação CPqD Telebrás

O centro de pesquisas e desenvolvimento (CPqD) do sistema Telebrás, em cooperação com suas empresas operadoras, está desenvolvendo o sistema SAGRE – Sistema Automatizado de Gerenciamento de Rede Externa [Arg+93, Mag93, Mag+94]. O SAGRE é um projeto de mapeamento automático / gerência de facilidades que manipula tanto as informações alfanuméricas quanto as informações topológicas dos elementos da rede externa de telecomunicações. Por rede externa entende-se o conjunto de cabos, canalizações subterrâneas, postes, equipamentos de sustentação e de proteção a esses cabos, além de dispositivos eletrônicos complementares. O sistema SAGRE utiliza um GIS comercial como software subjacente de geoprocessamento e os dados estão armazenados em um SGBD relacional comercial. A concepção lógica do software desenvolvido obedece a conceitos de orientação a objetos.

O sistema SAGRE tem por finalidade automatizar os processos de planejamento, projeto, cadastro, implantação, operação e manutenção da rede de telecomunicações, visando, por exemplo, a redução do tempo de implantação das redes telefônicas, a redução do custo do terminal instalado e a melhoria da qualidade dos serviços prestados à comunidade. Este sistema está atualmente organizado em seis módulos (cadastro, conversão, administração, planejamento, projeto e operação), porém existe a possibilidade do desenvolvimento de novos módulos.

O banco de dados do SAGRE é formado por um conjunto de estruturas topológicas denominadas redes, as quais permitem a modelagem de várias redes existentes, tais como rede de cabos e de canalização. Cada um dos elementos que compõem estas redes representa graficamente um item (como exemplo cabo, canalização) e pode possuir

atributos alfanuméricos associados (como exemplo tipo, capacidade, comprimento). Estes elementos do cadastro da rede externa são, em geral, posicionados geograficamente sobre o mapeamento urbano das localidades onde a rede é mantida. Desta forma, o mapeamento urbano é fundamental para o SAGRE, uma vez que é sobre ele que serão localizados os equipamentos de rede, desenhados os percursos das redes de canalização e aérea e esquemático de detalhes.

O mapeamento urbano básico (MUB), também chamado de base cartográfica, é um modelo básico de representação do mapeamento urbano no SAGRE e contém mapas das áreas urbana e rural da cidade que são representados no banco de dados. Em outras palavras, o MUB é o conjunto de informações gráficas e alfanuméricas referentes à base cartográfica das plantas cadastrais [Tel92].

O modelo de dados do MUB contém apenas um conjunto mínimo de informações necessárias para se relacionar com o modelo de rede e suportar os aplicativos que serão desenvolvidos no projeto SAGRE. As informações manipuladas por este modelo são estruturadas por representação gráfica e visam, além de suportar as aplicações previstas, minimizar a tarefa de coleta de dados e facilitar a interação com as prefeituras e as concessionárias de serviços públicos. A vantagem apresentada pela utilização deste modelo é a obtenção de uma padronização mínima necessária para que a informação existente possa ser processada pelo SAGRE sem perda ou redundância de dados.

De acordo com o modelo gráfico do MUB, os elementos básicos de mapeamento urbano são organizados como uma série de níveis. Estes níveis gráficos são:

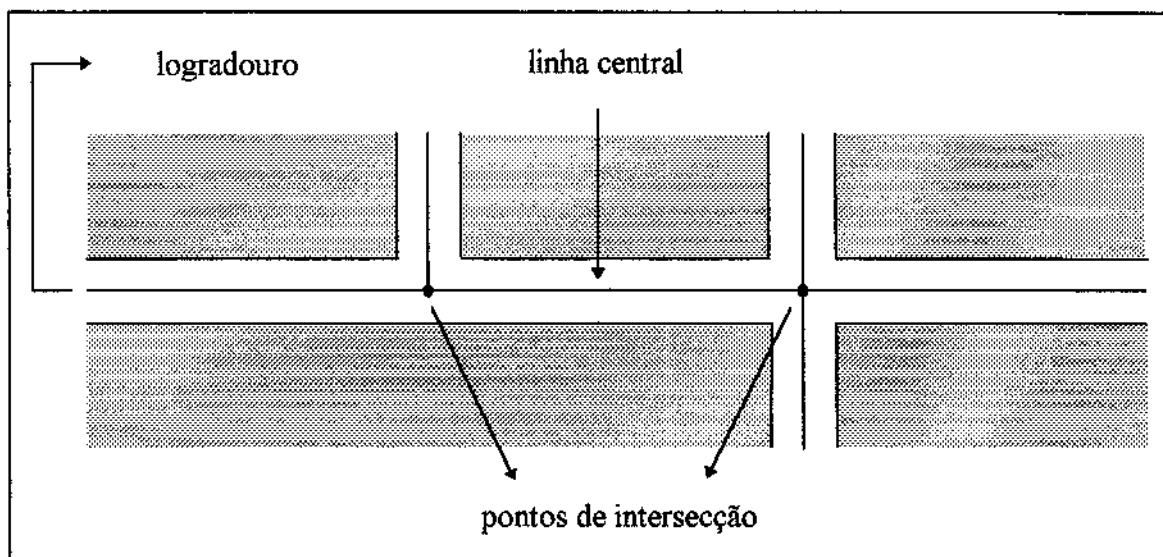
- linha central (trecho de logradouro)
- lotes ou divisas de lotes
- numeração predial
- nomenclatura dos logradouros
- arruamento, face de quadra ou quadra
- meio fio
- edificações de destaque e nomenclatura
- hidrografia e nomenclatura
- obras de arte e nomenclatura

- acidentes geográficos e nomenclatura
- passeios, calçadas e calçadas
- limites e nomenclatura
- marcos geográficos e grades de coordenadas

Apenas os níveis de linha central e de lote são imprescindíveis, e devem constar em todas as plantas convertidas, uma vez que a estes elementos gráficos estão relacionados, respectivamente, os atributos alfanuméricos de nomenclatura de logradouros e numeração predial, os quais serão utilizados para o processamento do SAGRE [Tel92].

### 5.1.1 O Modelo de Dados do Mapeamento Urbano Básico

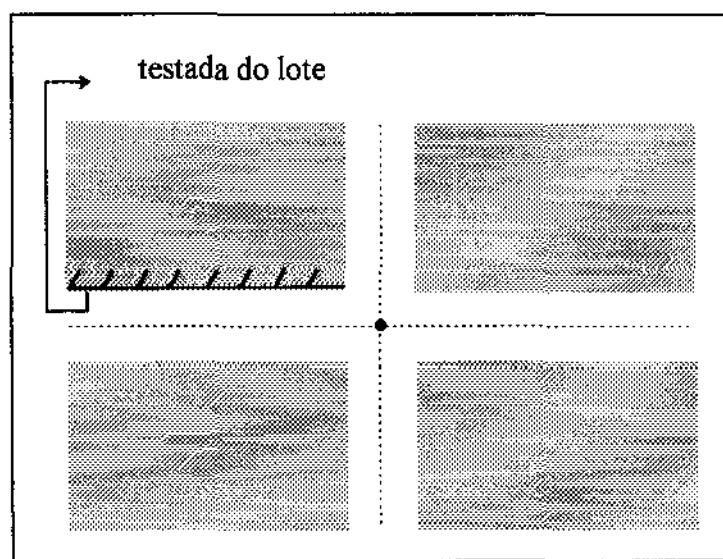
Um **logradouro** é qualquer rua, travessa, beco ou avenida, entre outros, constante no traçado urbanístico da cidade. No MUB, um logradouro é representado por um ou vários segmentos de linha central. Os segmentos de **linha central**, também chamados de **trecho de logradouro** ou “**center line**”, são linhas eqüidistantes de duas faces de quadra opostas, que passam ao longo do eixo dos logradouros e são delimitadas por pontos de intersecção (cruzamento de ruas e esquinas) com outros logradouros. Nos pontos de intersecção, os segmentos de linha central coincidem-se graficamente nas extremidades, formando uma rede de conectividade física. A figura 5.1 representa graficamente os conceitos logradouro e linha central.



**Figura 5.1** Representação gráfica dos conceitos logradouro e linha central.

No banco de dados do sistema, cada linha central está associada ao nome e ao código do logradouro a que pertence. Por outro lado, um logradouro é representado por um ou vários segmentos de linha central. Estas linhas centrais, além de representarem os logradouros, também servem de referência geográfica para a localização destes.

Outro elemento importante do modelo de dados do MUB é o **lote**. Um lote é simbolizado por uma linha que representa a sua face voltada para o arruamento. O significado semântico deste elemento é o mesmo que o de **testada de lote**. Já as divisas dos lotes das quadras do mapeamento urbano são simbolizadas por linhas que formam um ângulo de aproximadamente 45° com as linhas que representam os lotes. A figura 5.2 ilustra este conceito.



**Figura 5.2** Representação gráfica do conceito testada.

No banco de dados do sistema, a cada lote estão associadas a numeração predial, dados de terminais telefônicos existentes naquele lote, dados de inscrição em planos de expansão e dados de demanda. A numeração predial também está associada à linha central.

Uma vez definidos os elementos logradouro, linha central e lote (testada de lote), a figura 5.3 mostra um esquema simplificado do modelo de dados do MUB.

Dos elementos que compõem o modelo gráfico do MUB, apenas os presentes na figura 5.3 são representados no modelo de dados. Os demais elementos gráficos anteriormente enumerados possuem localização, mas servem apenas para visualização, não apresentando informações associadas no banco de dados do sistema [Arg+93].

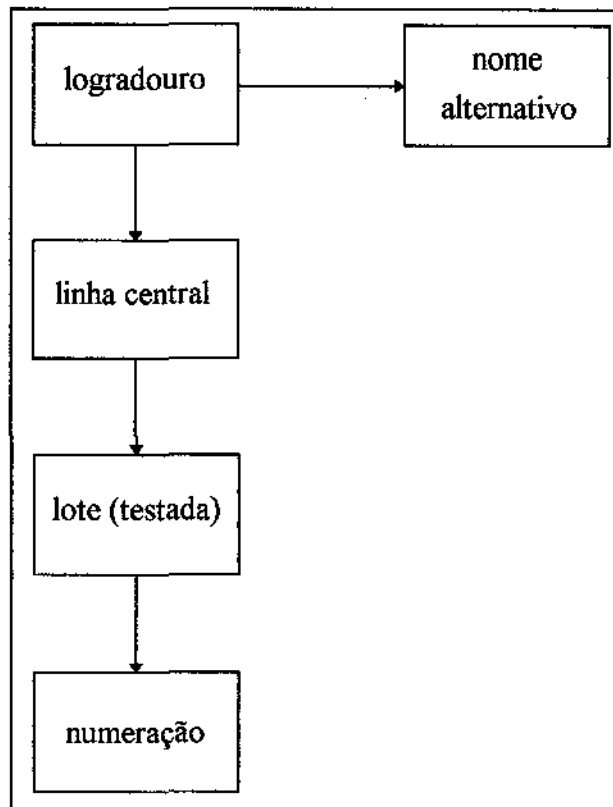


Figura 5.3 Esquema simplificado do modelo de dados do MUB.

## 5.2 Aplicação Eletropaulo

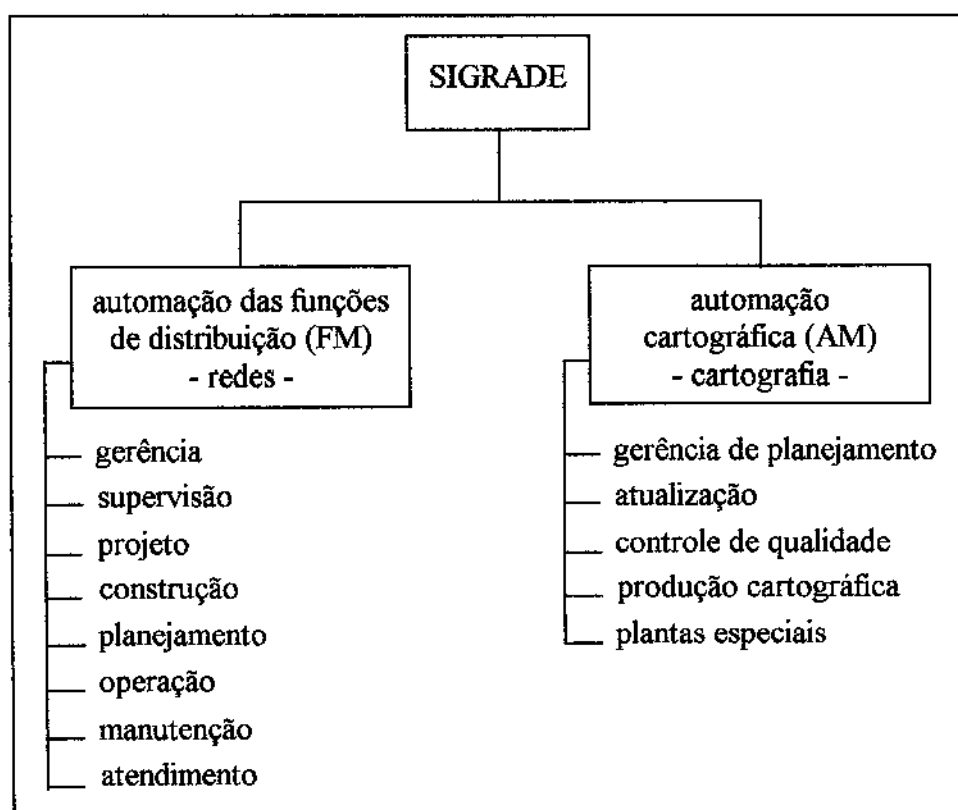
A empresa Eletropaulo (Eleticidade de São Paulo S/A) está desenvolvendo o sistema SIGRADE – Sistema de Informações Geográficas para Gestão de Redes de Distribuição de Energia [Web94a, Web94b]. O SIGRADE é um sistema AM/FM (“automated mapping” / “facilities management”) baseado em software SIG. Este sistema utiliza o um GIS comercial como software de geoprocessamento para a manipulação dos dados georeferenciados e um SGBD relacional para a manipulação dos dados alfanuméricos.

O sistema SIGRADE tem como objetivo automatizar as funções de distribuição elétrica tais como projeto, planejamento, construção, operação e manutenção. Além de garantir uma melhoria no atendimento aos clientes através do oferecimento de redes de distribuição mais eficientes e confiáveis, a empresa também tem obtido uma redução nos seus custos operacionais.

O projeto SIGRADE surgiu da necessidade da integração das bases de rede e cartográfica já existentes na Eletropaulo. Na base de dados de redes eram armazenados os

atributos e a localização dos elementos das redes primárias e secundárias. Já a base cartográfica era formada por um conjunto de mapas, materiais fotográficos, foto-índices e mosaicos que continham o mapeamento e o cadastro das redes, entre outros. No entanto, a base de redes estava armazenada em meio magnético e a base cartográfica em papel. O fato destas bases estarem armazenadas em mídias diferentes acarretava uma série de problemas e contribuiu, conseqüentemente, para o surgimento de um sistema automatizado para a manipulação de ambas as bases de maneira integrada.

Como citado anteriormente, o sistema SIGRADE é um sistema AM/FM. Como tal, este sistema oferece tanto funções relacionadas com o mapeamento automático da base cartográfica (AM) quanto facilidades relacionadas com a gerência da rede (FM). Esta estrutura funcional é esquematizada na figura 5.4.



**Figura 5.4** Estrutura funcional do sistema SIGRADE.

As facilidades voltadas à automação cartográfica estão incorporadas em um único sistema denominado AUTOCART – Automação Cartográfica [ACS94]. O AUTOCART é um dos sistemas mais importantes que compõem o SIGRADE, uma vez que fornece subsídios para a criação e a manutenção do banco de dados cartográfico digital utilizado para o gerenciamento e automação das várias funções típicas de distribuição de energia. Já as facilidades voltadas à automação das funções de distribuição estão implementadas em

diferentes módulos, cada um utilizando a base de dados georeferenciada principal [Mor+94].

## 5.2.1 O Modelo de Dados do SIGRADE

O ambiente SIGRADE mantém elementos elétricos e cartográficos em um único modelo de dados. Este modelo lógico integra tanto entidades espaciais quanto entidades alfanuméricas, respeitando os relacionamentos topológicos. Entidades espaciais são entidades que possuem como características localização, conectividade, dimensão espacial e forma. Tais entidades podem estar associadas entre si através de relacionamentos topológicos (como exemplo conecta, contém, adjacente a).

De acordo com as necessidades do ambiente e os recursos oferecidos pelo SIG utilizado, as entidades que compõem o modelo de dados do SIGRADE são classificadas em camadas de informação, as quais podem ser separadas em dois grupos [ACS94, Web94a]:

- **camadas referentes à cartografia:** quadras, detalhes construtivos, edificações, fronteiras geográficas, área de concessão, divisa de lote, elementos de hidrografia, elementos de superfície, elementos de utilidade urbana, município, limites de área, eixos de logradouro, vias de transporte, entre outros.
- **camadas referentes à rede elétrica:** subestações de energia elétrica, circuitos e trechos primários, secundários e de iluminação pública, chaves, fusíveis, disjuntores, transformadores, barramentos, pontos de entrega e postes, entre outros.

Todas as camadas de rede elétrica podem ser visualizadas tendo como referência os níveis de informação relacionados à base cartográfica.

Como pode ser observado, o modelo de dados do ambiente SIGRADE engloba elementos do mapeamento urbano básico (como exemplo quadra, lote, eixo de logradouro), elementos do mapeamento urbano mais específicos (como exemplo detalhes construtivos, edificações, elementos de superfície) e elementos do sistema de redes (linhas de transmissão, redes primárias e secundárias de distribuição de energia). No entanto, somente as entidades espaciais e alfanuméricas referentes ao modelo urbano básico e algumas referentes ao mapeamento urbano mais específico serão consideradas durante a integração deste modelo com o modelo de dados previamente apresentado da Telebrás. Em outras palavras, apenas os elementos de cartografia serão considerados.

Desta forma, a seguir é realizada uma breve especificação do modelo de dados do SIGRADE, na qual são destacados os principais elementos de mapeamento urbano deste modelo.

Uma **área de concessão** corresponde a uma área geográfica de atuação da Eletropaulo. A estrutura organizacional interna da empresa divide a área de concessão em uma ou mais áreas de regionais, baseada em alguns requisitos tais como região de atuação, departamento, divisão e seção em questão. Adicionalmente, é de grande interesse para a empresa o armazenamento de informações relativas às áreas geográficas eletricamente isoladas entre si, as chamadas **regiões elétricas**.

Um **município** é uma área geográfica correspondente à circunscrição administrativa autônoma do Estado. No banco de dados do sistema, para cada município são armazenadas informações tais como o nome, o código e a área total de ocupação. Em um município, existem diversas **localidades técnicas**, que são áreas geográficas definidas com o objetivo de atendimento técnico da rede elétrica e dos consumidores.

Outro elemento importante do modelo de dados do SIGRADE é a entidade divisa de lote. Uma **divisa de lote** representa a testada de um determinado lote em uma determinada **quadra**. Cada instância desta entidade possui informações relativas ao número do lote na rua, ao tipo de divisa e à quadra à qual o lote está associado. Para cada divisa de lote existe apenas um único número associado.

As divisas de lote são adjacentes aos eixos de logradouro. Um **eixo de logradouro** é uma porção do logradouro definida entre dois cruzamentos consecutivos. Desta forma, um **logradouro** é formado por um ou mais eixos de logradouro e representa uma via pública. No banco de dados do sistema, para cada logradouro são armazenadas informações como o código, o nome e o seu título, entre outras.

## 5.3 Elementos Comuns e Proposta de Integração

A integração de diferentes esquemas conceituais locais correspondentes aos diferentes SGBDs componentes em um único esquema conceitual global é de essencial importância para o projeto de bancos de dados heterogêneos. A dificuldade de sua realização está relacionada ao fato de que diferentes usuários modelam o mesmo pedaço do mundo real de maneiras distintas, de acordo com as suas percepções. A esta multiplicidade de representações de um determinado problema do mundo real dá-se o nome de relativismo semântico [SP94].

O relativismo semântico é, portanto, responsável pela existência de conflitos entre os elementos dos diversos esquemas locais. Tais conflitos devem ser identificados e solucionados durante o processo de integração. Para tanto, é necessário estabelecer uma correspondência entre os objetos dos diferentes esquemas através da identificação de suas similaridades e diferenças, além da identificação de conjuntos de objetos distintos que são relacionados entre si por alguma propriedade semântica.



Várias metodologias têm sido propostas para suportar o processo de integração de esquemas: [BOT86, Car87, Tem+87, Nye89, SPD92, AP93, MIR93, SP94]. [BLN86] faz uma comparação entre doze metodologias para a integração de visões ou bancos de dados, ou ambas. [Cle+93, Lim+93, LS93b, LSS94] são trabalhos referentes ao projeto do protótipo Myriad e discutem soluções para o problema de incompatibilidade a nível de instância.

A metodologia de integração de esquemas adotada neste trabalho é a descrita em [SP94]. Esta metodologia inicia o processo de integração com uma entrada manual do administrador do banco de dados. Esta entrada define a descrição formal dos inter-relacionamentos entre as visões. Baseada nestas descrições, as quais são conhecidas como asserções de correspondência, a metodologia automaticamente produz um esquema integrado e os mapeamentos entre este esquema final e cada uma das visões iniciais. Uma das vantagens apresentadas por esta metodologia é que ela é independente do modelo, uma vez que suas regras basicamente suportam integração de objetos e ligações, dois conceitos centrais da modelagem de dados conceitual.

De maneira simplificada, o algoritmo de integração de esquemas inicia com a integração de elementos correspondentes que não são atributos. Em seguida, caminhos correspondentes são integrados. Finalmente, atributos correspondentes são processados. A figura 5.5 apresenta alguns diagramas que sumarizam, para os casos mais comuns, as regras de integração propostas. Estes exemplos foram adaptados nesta dissertação à metodologia OMT. Após a integração, pode ser necessário refinar o esquema conceitual global resultante. O refinamento deste esquema pode ser realizado tanto automaticamente quanto com a ajuda do administrador do banco de dados.

Desta forma, a seguir são destacadas as asserções de correspondência entre os elementos do modelo Telebrás e os elementos do modelo Eletropaulo. Uma descrição detalhada destes elementos é realizada, respectivamente, nos apêndices A e B.

- *localidade*  $\equiv$  *município*

*com os atributos correspondentes*

*código\_localidade* = *código\_município*

*nome\_localidade* = *nome\_município*

- *linha central*  $\equiv$  *eixo de logradouro*

*com os atributos correspondentes*

*identificador* = *identif\_eixo*

- logradouro  $\equiv$  logradouro

*com os atributos correspondentes*

código\_logradouro = código\_lograd

nome\_logradouro = nome\_lograd

título\_logradouro = tit\_lograd

- lote  $\equiv$  divisa de lote

*com os atributos correspondentes*

descrição\_lote = toponímia\_divisa

num\_lote = número\_divisa

- logradouro – tipo\_logradouro  $\equiv$  logradouro – a<sub>1</sub> – eixo de logradouro – tipo\_lograd

*onde a<sub>1</sub> representa a associação entre as classes logradouro e eixo de logradouro*

- lote – a<sub>2</sub> – numeração – num  $\equiv$  divisa de lote – número\_divisa

*onde a<sub>2</sub> representa a associação entre as classes lote e numeração*

- a<sub>3</sub>  $\equiv$  a<sub>4</sub>

*onde a<sub>3</sub> representa a associação entre as classes lote e linha central*

*a<sub>4</sub> representa a associação entre as classes divisa de lote e eixo de logradouro*

- a<sub>5</sub>  $\equiv$  a<sub>1</sub>

*onde a<sub>5</sub> representa a associação entre as classes logradouro e linha central*

*a<sub>1</sub> representa a associação entre as classes logradouro e eixo de logradouro*

As asserções de correspondência acima foram especificadas sem uma comparação dos tipos dos atributos (cláusula *com os atributos correspondentes*), uma vez que uma especificação completa destes não foi possível. Desta forma, as equivalências foram determinadas baseadas somente no significado semântico destes atributos. Caso dois atributos correspondentes possuam tipos distintos, a eles deve ser aplicada uma função transformadora, a fim de que sejam integrados em um só atributo no esquema final.

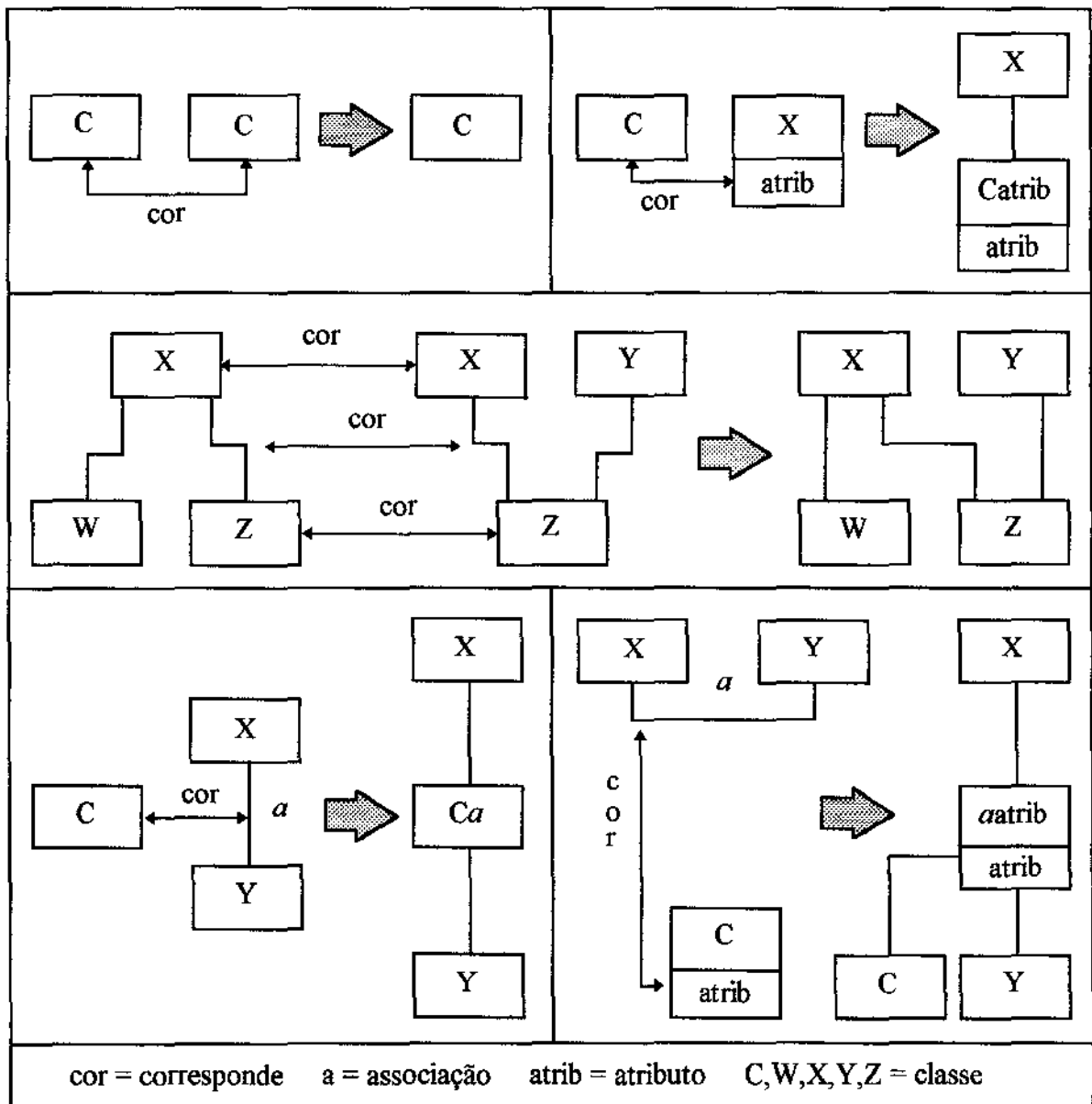


Figura 5.5 Regras de integração para casos mais comuns.

## Capítulo 6

# Modelagem das Aplicações e Integração

### 6.1 Processo de Integração – Observações Iniciais

Este capítulo descreve a integração dos modelos das aplicações do SAGRE e do SIGRADE, segundo a metodologia e algoritmos de transformação propostos nesta dissertação.

A análise e os objetivos das duas aplicações mostraram que, embora se tratando de um mesmo domínio (utilitários para redes públicas), existem necessidades e pontos de vista distintos. Tais diferenças são traduzidas, na modelagem de cada aplicação, por especificações de entidades próprias a cada domínio. Por exemplo, a aplicação da Eletropaulo tem necessidade de definição de *região elétrica*, mas este tipo de objeto geográfico não é necessário na aplicação do CPqD Telebrás. Outro exemplo são as definições relativas a *áreas e linhas de concessão*, que têm significados distintos (por exemplo, uma linha férrea precisa ser modelada na aplicação Eletropaulo, enquanto que na aplicação Telebrás aparece no máximo como mais um elemento urbano, sem associação de serviços telefônicos correspondentes).

Desta forma, após a análise dos dois domínios e modelos, chegou-se a conclusão que a integração deveria ser feita unicamente em relação ao MUB. Outros detalhes de cada domínio devem ser tratados de forma independente. É, inclusive, natural que a integração seja feita através do MUB, pois o contato entre os dois tipos de rede (elétrica e telefônica), do ponto de vista de sistema georeferenciado, é realizado exatamente através da associação de cada elemento à sua localização urbana.

Outro detalhe corresponde ao processo de integração em si. A proposta da dissertação é a integração de sistemas baseados em modelo de objetos e relacional. Para validar tal proposta, seria necessário encontrar aplicações já modeladas segundo estes dois paradigmas. Não existem, no entanto, sistemas urbanos reais que utilizem bancos de dados orientados a objetos. Assim sendo, e a partir das aplicações estudadas, decidiu-se remodelar a aplicação do CPqD usando orientação a objetos e em seguida aplicar os algoritmos propostos na dissertação às duas aplicações. A opção pelo SAGRE se deu principalmente pelo fato de que a modelagem existente no CPqD se adapta mais facilmente a uma transformação orientada a objetos, devido à forma de organização do projeto e à modelagem do banco de dados subjacente. No caso da Eletropaulo, já existia um modelo entidade-relacionamento da aplicação, que foi então integrado ao resto do sistema, segundo a metodologia proposta.

Em resumo, a integração das duas aplicações, descrita nas seções a seguir, utilizou as seguintes características:

- como se trata de aplicações georeferenciadas, a integração é feita a partir das propriedades de georeferenciamento e por conseguinte através de integração via MUB.
- a validação dos algoritmos da dissertação exigiu transformação preliminar do modelo SAGRE em um modelo geográfico orientado a objetos.

## 6.2 Modelagem de Mapeamento Urbano – Telebrás

A modelagem da aplicação de mapeamento urbano básico da Telebrás corresponde ao primeiro esquema componente a ser utilizado no processo de integração entre os bancos de dados orientado a objetos e relacional. Este esquema conceitual é modelado de acordo com o modelo orientado a objetos para aplicações geográficas MODGEO<sub>2</sub> descrito na seção 3.4. Este modelo utiliza o diagrama de classes da metodologia OMT [Rum+91].

O modelo MODGEO<sub>2</sub> é um modelo de organização que oferece um conjunto de classes genéricas ou metaclasses que devem ser adaptadas a cada aplicação geográfica específica. A figura 6.1 sumariza a correspondência existente entre os elementos do modelo de dados da Telebrás descritos na seção 5.1.1 e as categorias do modelo MODGEO<sub>2</sub>. Além dos elementos *logradouro*, *nome alternativo*, *linha central*, *lote* e *numeração*, também são especificados novos elementos, tais como *planejamento urbano*, *localidade*, *mapa cadastral de lote*, *lote\_f*, *linha central\_f*, *coordenada*, *polilinha* e *linha*. Estas novas classes foram adicionadas com a finalidade de oferecer um modelo mais claro e genérico.

<b>Projeto:</b>	planejamento urbano	
<b>Entidade Lógica:</b>	localidade	
<b>Plano de Informação:</b>	mapa cadastral de lote	
<b>Objeto Temático:</b>	lote	logradouro
<b>Objeto Geográfico:</b>	lote	linha central
<b>Objeto Temporal:</b>	X	X
<b>Objeto Gráfico:</b>	lote_f	linha central_f
<b>Objeto Espacial:</b>	coordenada	coordenada
<b>Objeto Geométrico:</b>	polilinha	linha

**Figura 6.1** Correspondência entre os elementos do modelo Telebrás e as categorias do modelo MODGEO<sub>2</sub>.

Uma vez especificadas as correspondências existentes entre os elementos do modelo de dados e as metaclasses do modelo MODGEO<sub>2</sub>, a figura 6.2 esboça o esquema conceitual do mapeamento urbano básico do CPqD Telebrás. As associações representadas nesta figura foram obtidas de especificações da Telebrás [Tel94] e do modelo de organização MODGEO<sub>2</sub>. A classe *coordenada* é representada duas vezes por motivos de clareza. Uma descrição do significado semântico dos elementos (classes, atributos e associações) deste modelo é realizada no apêndice A.

## 6.3 Modelagem de Mapeamento Urbano – Eletropaulo

A modelagem da aplicação de mapeamento urbano da Eletropaulo corresponde ao segundo esquema componente a ser utilizado no processo de integração entre os bancos de dados orientados a objetos e relacional. Este esquema conceitual é modelado utilizando-se o modelo semântico ECR [EWH85].

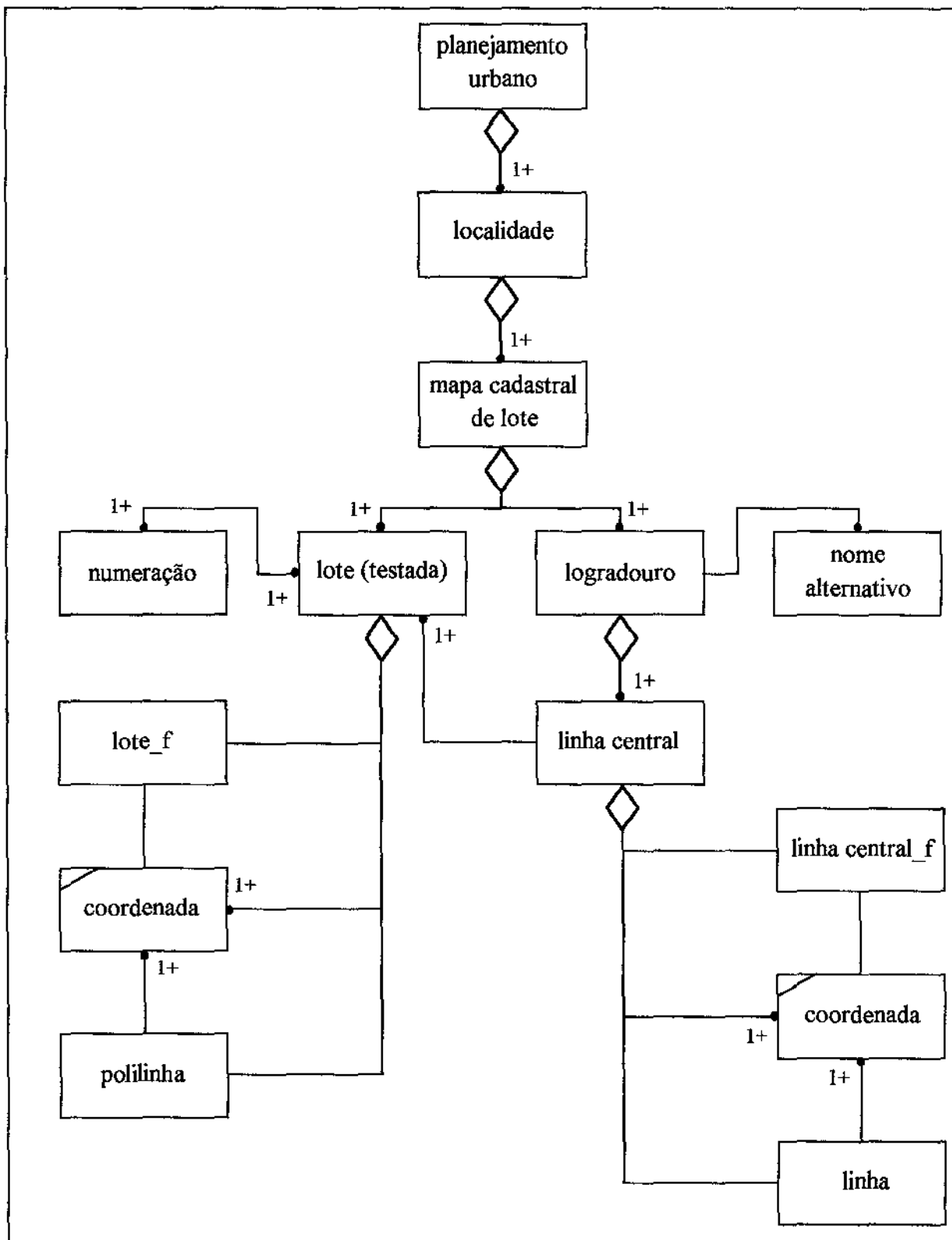
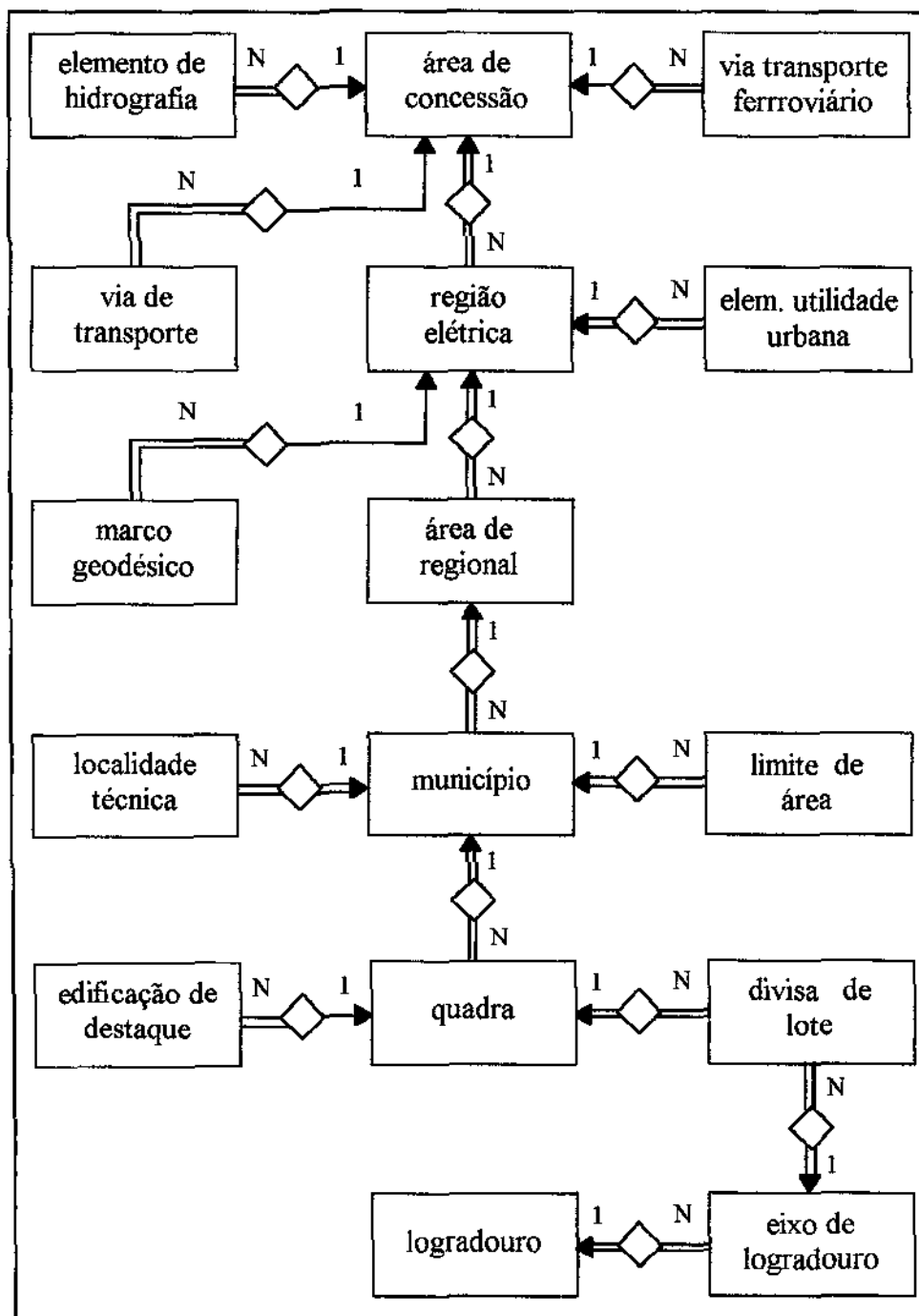


Figura 6.2 Esquema conceitual do MUB Telebrás.

A figura 6.3 esboça o esquema conceitual do mapeamento urbano da Eletropaulo. Os relacionamentos representados nesta figura foram obtidos de especificações da Eletropaulo [Ele93, Ele94]. Uma descrição do significado semântico dos elementos (entidades, atributos e relacionamentos) deste modelo é realizada no apêndice B.



**Figura 6.3** Esquema conceitual do mapeamento urbano da Eletropaulo.



## 6.4 Integração

### 6.4.1 Tradução do modelo Eletropaulo

Antes de iniciar a integração propriamente dita dos elementos dos modelos acima descritos, é necessário traduzir o esquema conceitual do mapeamento urbano da Eletropaulo, o qual foi realizado utilizando o modelo ECR, para um esquema conceitual equivalente utilizando o modelo OMT (modelo comum). Esta tradução é realizada com base nas regras de mapeamento descritas na seção 4.2.1. A figura 6.4 mostra este novo esquema.

### 6.4.2 Integração dos Esquemas Conceituais

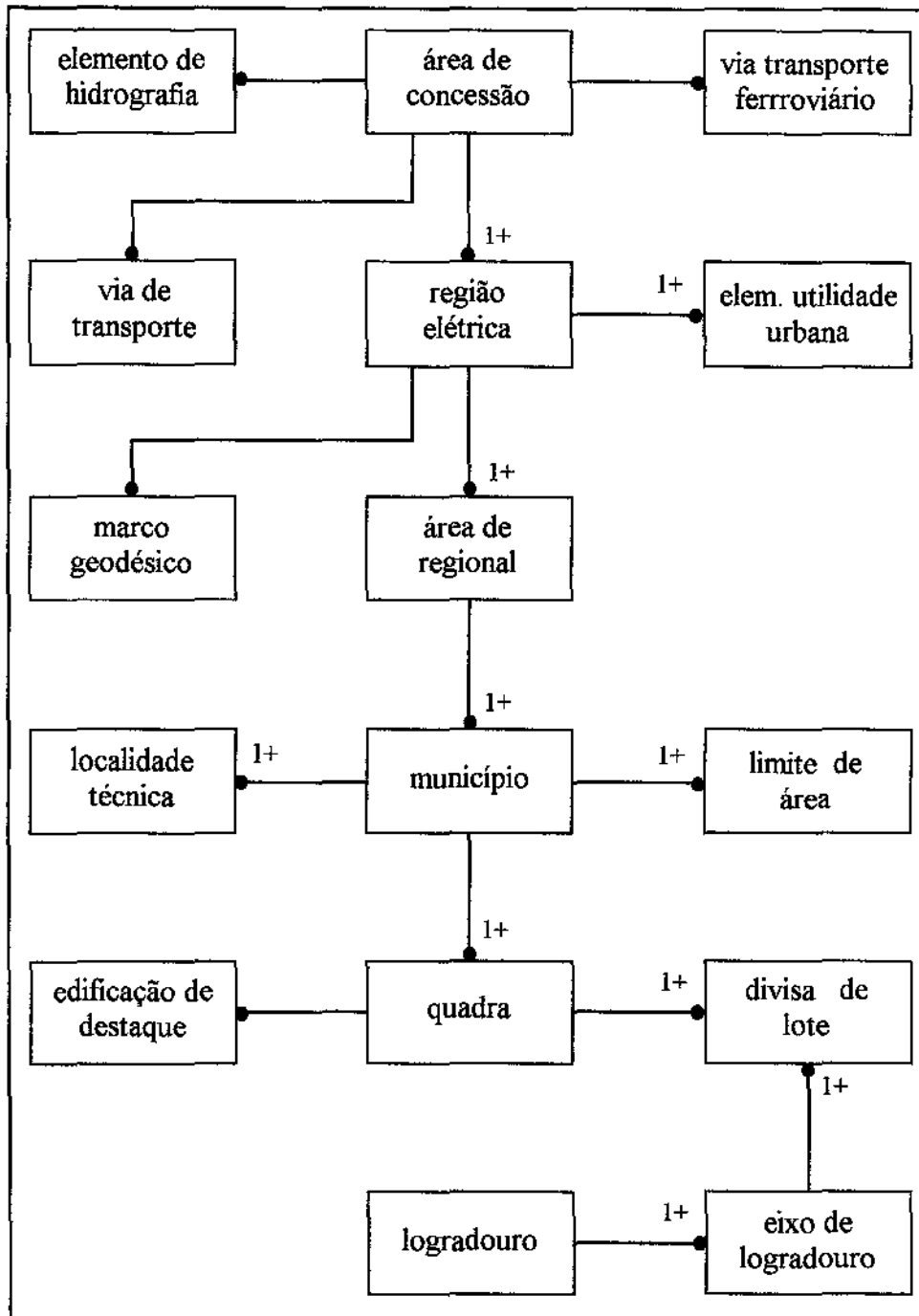
Conforme já mencionado anteriormente, a integração será feita unicamente em função do MUB. A partir desta experiência, esta dissertação propõe que, para aplicações urbanas georeferenciadas, qualquer integração de esquemas seja de fato realizada em dois passos:

- inicialmente, integração dos componentes georeferenciados das aplicações através de exame e unificação dos MUBs respectivos.
- a seguir, integração dos demais componentes (se houver).

Com isto, existe a garantia da integridade semântica do ponto de vista de localização, imprescindível em aplicações georeferenciadas. Os demais passos de integração são semelhantes aos realizados em aplicações não georeferenciadas.

A integração via MUB tem por objetivo evitar que os passos posteriores não acarretem inconsistências do ponto de vista semântico. Por exemplo, considere o problema bastante comum de existência de nomes que denotam o mesmo conceito em duas aplicações distintas – por exemplo, as diferentes formas de designar lotes nas duas aplicações analisadas aqui. O fato destes conceitos estarem associados a regiões urbanas correspondentes (o que pode ser visto através de integração via MUB) facilita identificar que se trata do mesmo elemento no segundo passo da integração.

Após a tradução do modelo da Eletropaulo para um modelo equivalente utilizando a metodologia OMT, foi realizada a integração dos esquemas correspondentes ao modelo da Telebrás e ao modelo da Eletropaulo. A figura 6.5 mostra o esquema integrado. Esta integração foi realizada respeitando as observações da seção 5.3. Nesta figura, por simplicidade, os atributos das classes não foram representados.



**Figura 6.4** Tradução do esquema conceitual do mapeamento urbano da Eletropaulo.

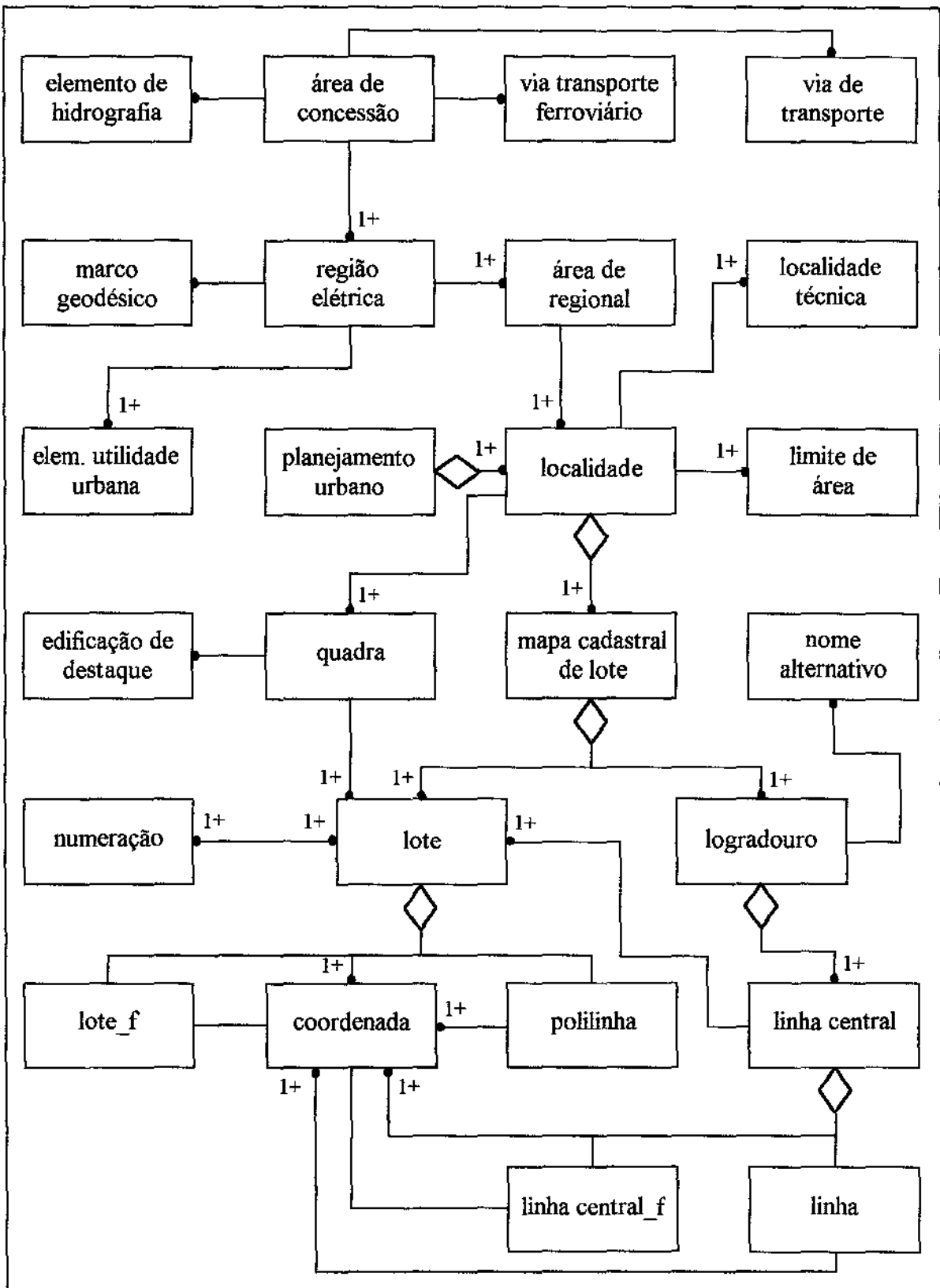


Figura 6.5 Esquema integrado.

### 6.4.3 Refinamento do Esquema Conceitual Integrado

O refinamento do esquema conceitual global resultante é realizado após o processo de integração, e tem como finalidade a produção de um modelo integrado mais claro e conciso. Segundo [SP94], este refinamento pode ser realizado tanto automaticamente quanto com a ajuda do administrador do banco de dados. Adotando-se a segunda abordagem, o administrador do banco de dados é capaz de identificar várias alterações que podem ser realizadas no esquema conceitual final do mapeamento urbano com o objetivo de torná-lo mais adequado às situações do mundo real. Estas alterações são listadas a seguir:

- pode ser adicionada uma agregação entre as classes *quadra* e *mapa cadastral de lote*, uma vez que um mapa cadastral de lote pode ser composto por uma ou mais quadras.
- a associação entre as classes *localidade* e *quadra* poderia ser substituída por uma agregação, uma vez que um município é composto por uma ou mais quadras. No entanto, baseado nesta possível alteração e na cláusula transitiva da agregação, a associação em questão pode ser removida, uma vez que se uma quadra é parte de um mapa cadastral de lote e um mapa cadastral de lote é parte de uma localidade, então uma quadra é parte de uma localidade. Desta forma, as agregações entre *localidade* e *mapa cadastral de lote* e *mapa cadastral de lote* e *quadra* representam implicitamente a associação entre *localidade* e *quadra*.
- a associação entre as classes *quadra* e *lote* pode ser substituída por uma agregação, uma vez que uma quadra é composta por um ou mais lotes.
- a agregação entre as classes *mapa cadastral de lote* e *lote* pode ser removida, uma vez que se um lote é parte de uma quadra e uma quadra é parte de um mapa cadastral de lote, então um lote é parte de um mapa cadastral de lote (cláusula transitiva). Desta forma, as agregações entre as classes *mapa cadastral de lote* e *quadra* e *quadra* e *lote* representam implicitamente a associação entre mapa cadastral de lote e lote.

Estas alterações são apresentadas na figura 6.6, a qual mostra o esquema conceitual final do processo de integração. A classe *coordenada* é representada duas vezes por motivos de clareza. Uma descrição completa dos elementos que compõem este esquema conceitual é realizada no apêndice C. Neste apêndice também estão destacadas quais classes, atributos e associações foram integrados.

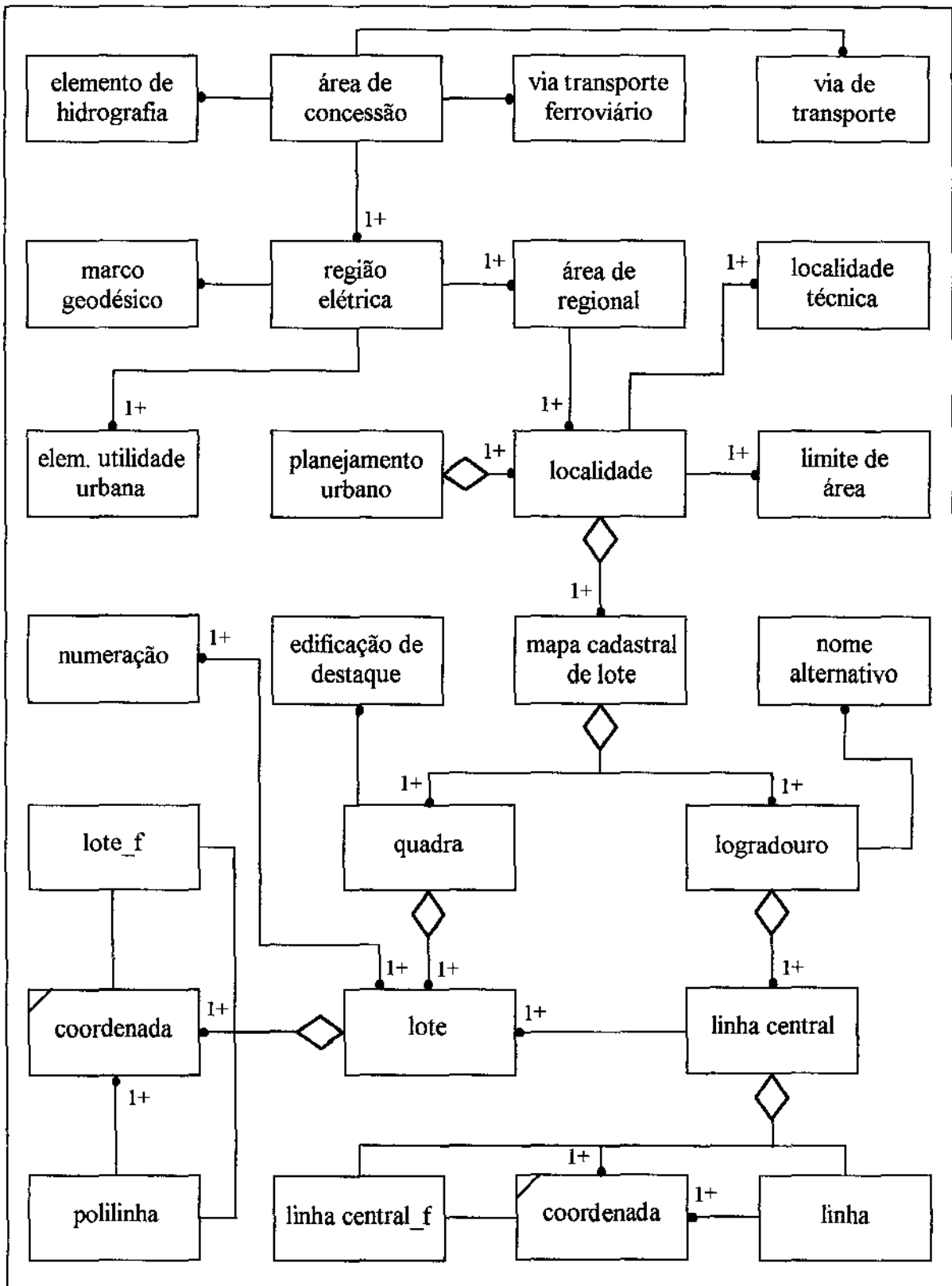


Figura 6.6 Refinamento do esquema integrado.

## 6.4.4 Visão Geral da Integração

Para mostrar o efeito da integração, a seguir são descritos alguns exemplos de consultas padrão nos três ambientes – ambiente relativo à aplicação CPqD Telebrás, ambiente relativo à aplicação Eletropaulo e ambiente global refinado. Tais consultas são caracterizadas pela identificação das características de um determinado elemento a partir da identificação do local onde este se encontra. Através da formulação destas consultas, foi constatado que não houve qualquer perda de informações devido ao processo de integração.

Para cada uma das três consultas a seguir, considere que qualquer usuário pode acessar qualquer dado do esquema global. Ou seja, nestes exemplos, não há preocupação para o fato de que o usuário é autorizado ou não a acessar os dados compartilhados.

**Consulta 1:** Liste as informações dos eixos de logradouro que compõem o logradouro de nome “X”.

- esquema conceitual do MUB Telebrás

passo 1 Pesquisar a classe logradouro identificando o objeto cujo valor do atributo *nome\_logradouro* é “X”.

passo 2 Pesquisar a classe linha central a partir da agregação *linha central – logradouro*.

resultado Para cada linha central que compõe o logradouro “X”, são listadas todas as suas informações – *identificador*, *status*, *pavimentado*, *t1* e *t2* (descritas detalhadamente no apêndice A).

- esquema conceitual do mapeamento urbano da Eletropaulo

passo 1 Pesquisar o tipo-entidade *logradouro* identificando o objeto cujo valor do atributo *nome\_lograd* é “X”.

passo 2 Pesquisar o tipo-entidade *eixo de logradouro* a partir do relacionamento *eixo de logradouro – logradouro*.

resultado Para cada eixo de logradouro que compõe o logradouro “X”, são listadas todas as suas informações – *identif\_eixo*, *length*, *tipo\_lograd* (descritas detalhadamente no apêndice B).

- esquema conceitual global refinado

passo 1 Pesquisar a classe *logradouro* identificando o objeto cujo valor do atributo *nome\_logradouro* é “X”.

passo 2 Pesquisar a classe *linha central* a partir da agregação *linha central – logradouro*.

resultado Para cada linha central que compõe o logradouro “X”, são listadas todas as suas informações – *identificador, status, pavimentado, t1, t2, length* e *tipo\_lograd* (descritas detalhadamente no apêndice C). Caso os usuários desejem selecionar apenas as mesmas informações que as obtidas em suas aplicações originais, estes devem fazer uma projeção apenas dos campos correspondentes.

**Consulta 2:** Determine o tipo do logradouro cujo código é C.

- esquema conceitual do MUB Telebrás

passo 1 Pesquisar a classe *logradouro* identificando o objeto cujo valor do atributo *código\_logradouro* é C.

resultado Valor do atributo *tipo\_logradouro* para o logradouro C.

- esquema conceitual do mapeamento urbano da Eletropaulo

passo 1 Pesquisar o tipo-entidade *logradouro* identificando o objeto cujo valor do atributo *código\_lograd* é C.

passo 2 Pesquisar o tipo-entidade *eixo de logradouro* a partir do relacionamento *eixo de logradouro – logradouro*.

passo 3 Determinar o tipo de logradouro predominante de acordo com o atributo *tipo\_lograd* das entidades obtidas no passo anterior.

resultado Valor do atributo *tipo\_lograd* para o logradouro C.

- esquema conceitual global refinado

passo 1 Pesquisar a classe *logradouro* identificando o objeto cujo valor do atributo *código\_logradouro* é C.

passo 2 Pesquisar a classe *linha central* a partir da agregação *linha central – logradouro*.

passo 3 Determinar o tipo de logradouro predominante de acordo com o atributo *tipo\_lograd* dos objetos obtidos no passo anterior.

resultado Valor do atributo *tipo\_lograd* para o logradouro C.

**Consulta 3:** Recupere a área das quadras que possuem mais do que 10 lotes.

- esquema conceitual do MUB Telebrás

passo 1 Executar uma função analítica para determinar, a partir dos lotes (localização espacial destes), quais lotes originam uma quadra.

passo 2 Executar uma função analítica para verificar, em cada quadra formada, se ela é composta por mais de 10 lotes.

passo 3 Executar uma função analítica para calcular a área para cada quadra obtida no passo anterior.

resultado As áreas das quadras compostas por mais de 10 lotes.

- esquema conceitual do mapeamento urbano da Eletropaulo

passo 1 Pesquisar o relacionamento *divisa de lote – quadra*, determinando, para cada membro do tipo-entidade *quadra*, quantas instâncias do tipo-entidade *divisa de lote* estão a ele associadas.

resultado O valor do atributo *área* para as quadras que são compostas por mais de 10 lotes.

- esquema conceitual global refinado

passo 1 Pesquisar a agregação *lote – quadra*, determinando, para cada objeto da classe *quadra*, quantas instâncias da classe *lote* estão a ele associadas.

resultado O valor do atributo *área* para as quadras que são compostas por mais de 10 lotes.



# Capítulo 7

## Conclusões

A dissertação propõe um modelo para a integração de sistemas de bancos de dados heterogêneos em aplicações de planejamento urbano.

Aplicações SIGs atuais utilizam bancos de dados relacionais para o gerenciamento dos seus dados. No entanto, devido à natureza espacial dos dados, o SGBD relacional deve ser estendido de modo a suportar eficientemente estas aplicações. Por outro lado, modelos orientados a objetos são mais estruturados e semanticamente mais ricos que o modelo relacional. Desta forma, a tendência futura para aplicações que utilizam SIGs é a orientação a objetos. Neste contexto, o processo de integração surge como uma solução viável para o compartilhamento dos dados armazenados em ambos bancos de dados.

Desta forma, os SGBDs orientado a objetos e relacional, representados respectivamente por esquemas que utilizam os modelos OMT [Rum+91] e ECR [EWH85] foram integrados com o objetivo de oferecer aos usuários uma camada virtual através da qual estes podem acessar os dados localizados nos BDs de maneira transparente, sem se preocupar com a localização dos dados ou com a heterogeneidade semântica. Para tanto, foram oferecidas primitivas de mapeamento entre os conceitos destes dois modelos de dados, além de primitivas para a conversão de esquemas OMT em definições de classes do modelo O<sub>2</sub>. Os mapeamentos para a tradução de esquemas ECR para o modelo relacional são encontrados em [EWH85, Oli93].

A validação da integração proposta foi realizada através da modelagem de duas aplicações reais de mapeamento urbano: uma oriunda da Telebrás [Mag93] e outra da Eletropaulo [Web94a]. Estas duas aplicações foram integradas considerando as entidades do mapeamento urbano básico, tais como logradouro, lote e linha central, entre outras. Nesta integração, foram utilizadas tanto as primitivas de mapeamento oferecidas quanto o processo de integração descrito em [SP94]. Finalmente, após a integração e o posterior refinamento do esquema global obtido, algumas consultas foram formuladas tanto no esquema original (ambiente do SGBD componente) quanto no esquema global (ambiente do SGBD heterogêneo), comprovando a manutenção das propriedades semânticas dos esquemas originais no esquema global.

Desta forma, as principais contribuições decorrentes da realização deste trabalho são:

- proposta de um mecanismo de integração de aplicações georeferenciadas em um ambiente de banco de dados heterogêneos.
- análise e algoritmos para mapeamentos entre os modelos ECR e OMT.
- análise e algoritmos para mapeamentos entre o modelo OMT e o modelo do sistema O<sub>2</sub>.
- análise de duas aplicações reais, de grande porte, com identificação dos principais aspectos necessários à sua integração, segundo a metodologia proposta.
- validação através de integração destas aplicações, ressaltando a importância de utilização do mapeamento urbano básico como base para tal integração.

Várias extensões podem ser realizadas neste trabalho, tanto práticas quanto teóricas. O principal conjunto de extensões do ponto de vista prático se refere, naturalmente, à implementação dos vários mapeamentos propostos na dissertação entre os distintos modelos de dados. Outra extensão, de maior porte, seria o desenvolvimento de uma camada que permita integrar bancos de dados relacionais e orientados a objetos a partir de uma interface baseada nos modelos ECR e OMT. Ainda outra extensão seria incorporar a modelagem espacial propriamente dita de forma automática.

Do ponto de vista teórico, uma extensão seria a complementação da proposta por uma linguagem de manipulação de objetos geográficos, a ser utilizada pelo usuário e mapeada em construções dos modelos subjacentes. A proposta de mapeamentos do modelo OMT para o modelo de dados de outros SGBDs orientados a objetos também poderia ser efetuada. Outra extensão seria comparar a solução proposta de integração a outras soluções, por exemplo, em termos de tradução de consultas entre modelos. Embora a proposta tenha sido motivada por aplicações de planejamento urbano, o modelo orientado a objetos georeferenciado é de propósito geral. Assim, uma outra extensão seria a verificação da aplicabilidade da proposta de integração a aplicações sobre outros domínios georeferenciados, em especial ambientais. Neste caso, um possível problema seria a integração de dados no formato raster, cuja modelagem exige a utilização de definições funcionais, não encontradas no sistema relacional, e sem suporte direto no modelo OMT. Finalmente, uma última extensão do modelo seria o acoplamento de um módulo baseado em regras com a finalidade de auxiliar o planejador urbano a tomar decisões.

# Referências Bibliográficas

- [ACS94] AUDI, R., CURADO, L. C. L., SAKAMOTO, F. H. AUTOCART – Sistema de Informação da Cartografia. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Painéis, p.58-66.
- [Ant+91] ANTENUCCI, J. C., *et al.* *Geographic Information Systems - A Guide to the Technology*. Van Nostrand Reinhold, 1991. 301p.
- [AP93] AZARBOD, C., PERRIZO, W. Building Concept Hierarchies for Schema Integration in HDDBS using Incremental Concept Formation. In: 2nd International Conference on Information and Knowledge Management, CIKM, Washington, DC, USA. *Proceedings*, 1993. p.732-734.
- [Arg+93] ARGONDIZIO, E. L., *et al.* Conversão de Mapeamento Urbano - Uma Metodologia para o Projeto SAGRE. In: IV Conferência Latino Americana sobre Sistemas de Informação Geográfica e II Simpósio Brasileiro de Geoprocessamento, São Paulo, SP, Brasil. *Anais*, 1993. p.207-219.
- [BGS92] BREITBART, Y., GARCIA-MOLINA, H., SILBERSCHATZ, A. Overview of Multidatabase Transaction Management. *The VLDB Journal*, v.1, n.2, p.181-238, October 1992.
- [BHP92] BRIGHT, M. W., HURSON, A. R., PAKZAD, S. H. A Taxonomy and Current Issues in Multidatabase Systems. *Computer*, v.25, n.3, p.50-59, March 1992.
- [BK91] BARGHOUTI, N. S., KAISER, G. E. Concurrency Control in Advanced Database Applications. *ACM Computing Surveys*, v.23, n.3, p.269-317, September 1991.
- [BL84] BATINI, C., LENZERINI, M. A Methodology for Data Schema Integration in Entity-Relationship Model. *IEEE Transactions on Software Engineering*, v.10, n.6, p.650-664, 1984.
- [BLN86] BATINI, C., LENZERINI, M., NAVATHE, S. B. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, v.18, n.4, p.323-364, December 1986.

- [BOT86] BREITBART, Y., OLSEN, P. L., THOMPSON, G. R. Database Integration in a Distributed Heterogeneous Database System. *IEEE Conference on Data Engineering*, p.301-310, February 1986.
- [BST92] BREITBART, Y., SILBERSCHATZ, A., THOMPSON, G. R. Transaction Management Issues in a Failure-Prone Multidatabase System Environment. *The VLDB Journal*, v.1, n.1, p.1-39, July 1992.
- [Car87] CARDENAS, A. F. Heterogeneous Distributed Database Management: The HD-DBMS. In: *IEEE. Proceedings*, v.75, n.5, 1987. p.588-600.
- [Che76] CHEN, P. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, v.1, n.1, p.9-36, 1976.
- [Cle+93] CLEMENTS, D., *et al.* *Myriad: Design and Implementation of a Federated Database Prototype*. CS Dept, Univ. of Minnesota, 1993. 19p. (Technical Report)
- [Cow90] COWEN, D. J. GIS versus CAD versus DBMS: What are the Differences?. In: D. J. Peuquet and D. F. Marble, editors, *Introductory Readings in Geographic Information Systems*, p.52-61. Taylor & Francis Ltd, 1990.
- [CSV94] CAVALCANTI, A. E. C., SILVA, I. C., VIEIRA FILHO, M. M. Experiência da CELPE no Processo de Automação de Geoprocessamento para o Controle da Distribuição de Energia no Estado de Pernambuco. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Concessionárias, p.18-26.
- [Dan89] DANGERMOND, J. A Review of Digital Data Commonly Available and Some of the Practical Problems of Entering Them into a GIS. In: W. J. Ripple, editor, *Fundamentals of Geographic Information Systems: A Compendium*, p.41-58. The American Society for Photogrammetry and Remote Sensing and the American Congress on Surveying and Mapping, 1989.
- [Dan90] DANGERMOND, J. A Classification of Software Components Commonly Used in Geographic Information Systems. In: D. J. Peuquet and D. F. Marble, editors, *Introductory Readings in Geographic Information Systems*, p.30-51. Taylor & Francis Ltd, 1990.
- [Deu+92] DEUX, O., *et al.* The Story of O<sub>2</sub>. In: F. Bancilhon, C. Delobel and P. Kanellakis, editors, *Building an Object-Oriented Database System: The Story of O<sub>2</sub>*, p.21-57. Morgan Kaufmann Publishers, Inc., 1992.
- [DKS92] DU, W., KRISHNAMURTHY, R., SHAN, M. C. Query Optimization in Heterogeneous DBMS. In: 18th VLDB CONFERENCE, Vancouver, Canada. *Proceedings*, 1992. p.277-291.

- [Dre+93] DREW, P., *et al.* Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems. *SIGMOD RECORD*, v.22, n.3, p.47-56, September 1993.
- [Ele93] ELETROPAULO. *Modelo Lógico para Implementação da Rede de Distribuição em Ambiente AM/FM*. São Paulo: Eletropaulo, 1993. 24p. (Relatório Técnico).
- [Ele94] ELETROPAULO. *Especificação Funcional – SIMAC – AUTOCART*. São Paulo: Área da Diretoria de Distribuição, Eletropaulo, 1994. 50p. (Relatório Técnico).
- [EN89] ELMASRI, R., NAVATHE, S. B. *Fundamentals of Database Systems*. The Benjamin/Cummings Publishing Company, Inc, 1989. 802p.
- [EP90] ELMAGARMID, A. K., PU C. Guest Editors' Introduction to the Special Issue on Heterogeneous Databases. *ACM Computing Surveys*, v.22, n.3, p.175-178, September 1990.
- [EWH85] ELMASRI, R., WEELDREYER, J., HEVNER, A. The Category Concept: An Extension to the Entity-Relationship Model. In: H. J. Schneider, editor, *Data and Knowledge Engineering*, p.75-116. North Holland, 1985.
- [Geo91] GEORGAKOPOULOS, D. Multidatabase Recoverability and Recovery. In: 1st International Workshop on Interoperability in Multidatabase Systems, Japan. *Proceedings*, 1991. California, IEEE Computer Society Press, 1991. p.348-355.
- [GRS91] GEORGAKOPOULOS, D., RUSINKIEWICZ, M., SHETH, A. On Serializability of Multidatabase Transactions Through Forced Local Conflicts. In: 7th International Conference on Data Engineering, Kobe, Japan. *Proceedings*, 1991. Califórnia, IEEE Computer Society Press, 1991. p.314-323.
- [HB91] HURSON, A. R., BRIGHT, M. W. Multidatabase Systems: An Advanced Concept in Handling Distributed Data. In: M. Yovits, editor, *Advances in Computers*, v.32, p.149-200, 1991.
- [HB92] HOLROYD, F., BELL, B. M. Raster GIS: Models of Raster Encoding. In: D. J. Maguire and J. F. Raper, editors, *Computers & Geosciences*, v.18, n.4, p.419-426. Pergamon Press, 1992.
- [HHS93] HWANG, S., HUANG, J. SRIVASTAVA, J. Concurrency Control in Federated Databases: A Dynamic Approach. In: 2nd International Conference on Information and Knowledge Management, CIKM, Washington, DC, USA. *Proceedings*, 1993. p.694-703.

- [HK89] HSIAO, D. K., KAMEL, M. N. Heterogeneous Databases: Proliferations, Issues, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, v.1, n.1, p.45-62, 1989.
- [HK93] HSU, C. N., KNOBLOCK, C. A. Reformulating Query Plans For Multidatabase Systems. In: 2nd International Conference on Information and Knowledge Management, CIKM, Washington, DC, USA. *Proceedings*, 1993. p.423-432.
- [Hsi92] HSIAO, D. K. Federated Databases and Systems: Part I - A Tutorial on Their Data Sharing. *The VLDB Journal*, v.1, n.1, p.127-179, July 1992.
- [JS92] JAGADISH, H. V., SHMUELI, O. A Proclamation-Based Model for Cooperating Transactions. In: 18th VLDB CONFERENCE, Vancouver, Canada. *Proceedings*, 1992. p.265-276.
- [KKM93] KEIM, D. A., KRIEGEL, H. P., MIETHSAM, A. Object-Oriented Querying of Existing Relational Databases. In: 4th International DEXA Conference. *Proceedings*, 1993. p.365-372.
- [Lem94] LEMOS, I. M. Projeto UNIBASE – Unificação das Bases Cadastrais da Região Metropolitana do Recife. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Regional, p.10-15.
- [Lim+93] LIM, E. P., *et al.* Entity Identification in Database Integration. In: 9th IEEE International Conference on Data Engineering, Vienna, Austria. *Proceedings*, 1993.
- [LS93a] LIM, E. P., SRIVASTAVA, J. Query Optimization and Processing in Federated Database Systems. In: 2nd International Conference on Information and Knowledge Management, CIKM, Washington, DC, USA. *Proceedings*, 1993. p.720-722.
- [LS93b] LIM, E. P., SRIVASTAVA, J. Entity Identification in Database Integration: An Evidential Reasoning Approach. International Symposium on Next Generation Database Systems and Their Applications, Fukuoka, Japan, 1993.
- [LSS94] LIM, E. P., SRIVASTAVA, J., SHEKHAR, S. Resolving Attribute Incompatibility in Database Integration: An Evidential Reasoning Approach. In: 10th IEEE International Conference on Data Engineering, Houston, USA. *Proceedings*, 1994.
- [Mag91] MAGUIRE, D. J. An Overview and Definition of GIS. In: D. J. Maguire, M. F. Goodchild and D. W. Rhind, editors, *Geographical Information Systems, Principles and Applications*, p.9-20. Longman Group UK Limited, 1991.

- [Mag93] MAGALHÃES, G. C. Projeto SAGRE - Sistema Automatizado de Gerência de Rede Externa - Telebrás - Telecomunicações Brasileiras S/A. *Fator GIS*, n.3, p.26-28, 1993.
- [Mag+94] MAGALHÃES, G. C. Especificação Técnica de Conversão de Dados - Proposta da TELEBRÁS – Projeto SAGRE. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Concessionárias, p.43-52.
- [Man+92] MANOLA, F., *et al.* Distributed Object Management. In: M. P. Papazoglou and T. K. Sellis, editors, *International Journal of Intelligent and Cooperative Information Systems*, v.1, n.1, p.5-42. World Scientific, 1992.
- [Mar90] MARBLE, D. F. Geographic Information Systems: An Overview. In: D. J. Peuquet and D. F. Marble, editors, *Introductory Readings in Geographic Information Systems*, p.8-17. Taylor & Francis Ltd, 1990.
- [Men+93] MENG, W., *et al.* Construction of a Relational Front-end for Object-Oriented Database Systems. *IEEE Data Engineering Conference*, 1993. p.476-483.
- [MIR93] MILLER, R. J. IOANNIDIS, Y. E., RAMAKRISHMAN, R. The Use of Information Capacity in Schema Integration and Translation. In: 9th Very Large Data Bases Conference, Dublin, Ireland. *Proceedings*, 1993. p.120-133.
- [MM93] MEDEIROS, C. B., MAGALHÃES, G. C. *Rule Application in GIS - a Case Study*. Campinas: DCC UNICAMP, 1993. 19p. (Relatório Técnico)
- [Mor+94] MORETTI, R., *et al.* VISUAL CADRE e VISUAL PROJAC – Sistemas de Cadastro e Projeto de Redes Elétricas de Distribuição. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Painéis, p.26-34.
- [Nye89] NYERGES, T. L. Schema Integration Analysis for the Development of GIS Databases. *International Journal of Geographical Information Systems*, v.3, n.2, p.153-183, 1989.
- [Oli93] OLIVEIRA, R. L. *Transparência de Modelos em Sistemas de Bancos de Dados Heterogêneos*. Campinas: DCC UNICAMP, 1993. (Tese de Mestrado)
- [OM93] OLIVEIRA, R., MAGALHÃES, G.C. *Metodologias para Conversão de Esquemas em Sistemas de Bancos de Dados Heterogêneos*. Campinas: DCC UNICAMP, 1993. 53p. (Relatório Técnico)
- [PB94] PREMIERLANI, W. J., BLAHA, M. R. An Approach for Reverse Engineering of Relational Databases. *Communications of the ACM*, v.37, n.5, p.42-49, 1994.

- [Peu90] PEUQUET, D. J. A Conceptual Framework and Comparison of Spatial Data Models. In: D. J. Peuquet and D. F. Marble, editors, *Introductory Readings in Geographic Information Systems*, p.250-285. Taylor & Francis Ltd, 1990.
- [PM90] PEUQUET, D. J., MARBLE, D. F. *Introductory Readings in Geographic Information Systems*. Taylor & Francis Ltd, 1990. 371p.
- [Pir95] PIRES, F. Um Ambiente de Modelagem de Dados Geográficos. To be published, 1995.
- [PMS93] PIRES, F., MEDEIROS, C. B., SILVA, A. B. Modelling Geographic Information Systems using an Object Oriented Framework. In: 13th International Conference of the Chilean Computer Society. *Proceedings*, 1993. p.217-232.
- [Raz92] RAZ, Y. The Principle of Commitment Ordering, or Guaranteeing Serializability in a Heterogeneous Environment of Multiple Autonomous Resource Managers Using Atomic Commitment. In: 18th VLDB CONFERENCE, Vancouver. *Proceedings*, 1992. p.292-312.
- [RCD93] RASCHID, L., CHANG, Y., DORR, B. J. Interoperable Query Processing With Multiple Heterogeneous Knowledge Servers. In: 2nd International Conference on Information and Knowledge Management, CIKM, Washington, DC, USA. *Proceedings*, 1993. p.461-470.
- [RM92] RARPER, J. R., MAGUIRE, D. J. Design Models and Functionality in GIS. In: D. J. Maguire and J. F. Rarper, editors, *Computers & Geosciences*, v.18, n.4, p.387-394. Pergamon Press, 1992.
- [Rum+91] RUMBAUGH, J., *et al.* *Object-Oriented Modeling and Design*. Prentice Hall, Inc., 1991. 500p.
- [San94] SANTANA, N. A. Geoprocessamento em Joinville. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Municipal, p.21-30.
- [SE90] STAR, J., ESTES, J. *Geographic Information Systems – An Introduction*. Prentice Hall, Inc, 1990. 303p.
- [Sie94] SIEBERT, U. SIG na Sanepar: Os Primeiros Passos. *Fator GIS*, n.7, p.29-31, 1994.
- [SL90] SHETH, A. P., LARSON, J. A. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, v.22, n.3, p.183-235, September 1990.



- [SL94] SACRAMENTO, E. R., LAENDER, A. H. R. Uma Abordagem Orientada a Objetos para o Projeto Lógico de Bancos de Dados Relacionais. In: 9º Simpósio Brasileiro de Banco de Dados, São Carlos, São Paulo, Brasil. *Anais*, 1994. p.110-126.
- [Smi+89] SMITH, T. R., *et al.* Requirements and Principles for the Implementation and Construction of Large-Scale Geographic Information Systems. In: W. J. Ripple, editor, *Fundamentals of Geographic Information Systems: A Compendium*, p.19-37. The American Society for Photogrammetry and Remote Sensing and the American Congress on Surveying and Mapping, 1989.
- [Sou+93] SOUZA, J. M., *et al.* Uma Arquitetura Organizacional para Sistemas de Informação Geográfica Orientados a Objetos. In: IV Conferência Latinoamericana sobre Sistemas de Informação Geográfica e II Simpósio Brasileiro de Geoprocessamento, São Paulo, SP, Brasil. *Anais*, 1993. p.187-204.
- [SP94] SPACCAPIETRA, S., PARENT, C. View Integration: A Step Forward in Solving Structural Conflicts. *IEEE Transactions on Knowledge and Data Engineering*, v.6, n.2, p.258-274, 1994.
- [SPD92] SPACCAPIETRA, S. PARENT, C. DUPONT, Y. Model Independent Assertions for Integration of Heterogeneous Schemas. *The VLDB Journal*, v.1, n.1, 1992. p.81-126.
- [ST93] SCHEUERMANN, P., TUNG, H. L. A Recovery Scheme for Multidatabase Systems. In: 2nd International Conference on Information and Knowledge Management, CIKM, Washington, DC, USA. *Proceedings*, 1993. p.665-673.
- [SV90] SCHOLL, M., VOISARD, A. Object-Oriented Database Systems for Geographic Applications: an Experiment with O<sub>2</sub>. Work in progress, unpublished draft, 1990.
- [Tel92] TELEBRÁS. *Aquisição de Dados de Mapeamento Urbano Básico - Recomendações para o Projeto SAGRE*. Campinas: CPqD Telebrás, 1992. 16p. (Relatório Técnico).
- [Tel94] TELEBRÁS. *Projeto Físico - Cadastro*. Campinas: CPqD Telebrás, 1994. 8p. (Relatório Técnico).
- [Tem+87] TEMPLETON, M. Mermaid – A Front End to Distributed Heterogeneous Databases. In: IEEE. *Proceedings*, v.75, n.5, 1987. p.695-708.
- [Tho+90] THOMAS, G., *et al.* Heterogeneous Distributed Database Systems for Production Use. *ACM Computing Surveys*, v.22, n.3, p.237-266, September 1990.

- [TLS94] TARACIEVICZ, M. C. S., LASS, M. C., SIKORSKI, S. R. Geoprocessamento Aplicado à Área do Município de Curitiba. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Municipal, p.1-10.
- [TS94] TIMES, V. C., SALGADO, A. C. Uma Modelagem Orientada a Objetos para Aplicações Geográficas. In: 9º Simpósio Brasileiro de Banco de Dados, São Carlos, São Paulo, Brasil. *Anais*, 1994. p.293-309.
- [Web94a] WEBER, M. A. A. Mais Energia para São Paulo. *Fator GIS*, n.5, p.46, 1994.
- [Web94b] WEBER, M. A. A. Geoprocessamento em Redes de Distribuição – Experiência Eletropaulo. In: GIS Brasil 94 – Congresso e Feira para Usuários de Geoprocessamento, Curitiba, PR, Brasil. *Anais*, 1994. Concessionárias, p.27-34.
- [WHM90] WORKBOYS, M. F., HEARNshaw, H. M., MAGUIRE, D. J. Object-Oriented Data Modelling for Spatial Databases. *International Journal of Geographical Information Systems*, v.4, n.4, p.369-383, 1990.
- [Woe+93] WOELK, D., *et al.* Task Scheduling Using Intertask Dependencies in Carnot. In: ACM SIGMOD Conference, Washington, DC, USA. *Proceedings*, 1993. p.491-494.

# Apêndice A

## Descrição Semântica dos Elementos do Modelo de Dados Telebrás

### Classes e Atributos

- **coordenada:** conjunto de pontos correspondentes às coordenadas dos objetos das classes *lote* e *linha central*.
- **linha:** indica que as coordenadas dos objetos da classe *linha central* correspondem a uma linha.
- **linha central:** trecho de logradouro delimitado por pontos de intersecção com outros logradouros.

**identificador:** identificador da linha central.

**status:** se a linha central está livre (1), se é imprópria para cabo (2), se é imprópria para duto (3) ou se está impedida (4).

**pavimentado:** se a linha central é pavimentada (0) ou não (1).

**t1:** data inicial de validade dos dados.

**t2:** data final de validade dos dados.

- **linha\_central\_f**: visualização gráfica do objeto da classe *linha\_central*, ou seja, figura que indica como a linha central será representada na tela.

**cor**: cor do desenho a ser exibido na tela.

**símbolo**: desenho propriamente dito.

**hachurado**: se a figura for hachurada, qual o grau de inclinação das linhas.

- **localidade**: cidade cujo mapeamento urbano está sendo realizado.

**código\_localidade**: código da localidade (campo numérico de cinco posições).

**cnl\_localidade**: código nacional da localidade (campo alfanumérico de cinco posições).

**nome\_localidade**: nome da localidade.

**data\_ref**: data de validade dos dados.

**descrição**: descrição alfanumérica da localidade.

**projeção**: projeção utilizada na inserção da localidade no banco de dados.

**ceinf**: canto esquerdo inferior do retângulo envolvente.

**cdsup**: canto direito superior do retângulo envolvente.

- **logradouro**: conjunto de linhas centrais que representam uma rua.

**código\_logradouro**: código real associado ao logradouro.

**nome\_logradouro**: nome do logradouro.

**tipo\_logradouro**: tipo do logradouro, por exemplo rua, avenida, beco, ...

**título\_logradouro**: título associado ao nome do logradouro, tal como comendador, general.

**t1**: data inicial de validade dos dados.

**t2**: data final de validade dos dados.

- **lote:** fachada (testada) de um lote voltada para o arruamento.
  - num\_lote:** menor número associado a um lote.
  - qtidade\_lote:** quantidade de números associados ao lote em questão.
  - descrição\_lote:** descrição alfanumérica do lote.
  - tipo\_lote:** se o lote existe (0) ou está somente projetado (1).
  - t1:** data inicial de validade dos dados.
  - t2:** data final de validade dos dados.
  
- **lote\_f:** visualização gráfica do objeto da classe *lote*, ou seja, figura que indica como o lote será representado na tela.
  - cor:** cor do desenho a ser exibido na tela.
  - símbolo:** desenho propriamente dito.
  - hachurado:** se a figura for hachurada, qual o grau de inclinação das linhas.
  
- **mapa cadastral de lote:** plano de informação da localidade utilizado na aplicação.
  - data\_ref:** data de validade dos dados.
  - descrição:** descrição alfanumérica do mapa cadastral de lote.
  - formato:** formato utilizado na inserção do mapa cadastral de lote no banco de dados (raster ou vector).
  - escala:** escala utilizada na inserção do mapa cadastral de lote no banco de dados.

- **nome alternativo:** nome alternativo associado a um logradouro. Em outras palavras, um logradouro pode possuir um nome e ser conhecido localmente por outro.

**nome\_alternativo:** nome alternativo do logradouro.

**tipo\_alternativo:** tipo do logradouro.

**título\_alternativo:** título associado ao nome do logradouro.

- **numeração:** representa os números que podem estar associados a um lote.

**num:** número do lote.

**descrição\_num:** descrição alfanumérica do número em questão.

- **planejamento urbano:** aplicação geográfica sendo implementada.

**data\_ref:** data de validade dos dados.

**nome:** nome do projeto.

**objetivo:** objetivo do projeto.

**coordenador:** coordenador do projeto.

- **polilinha:** indica que as coordenadas do objeto da classe lote correspondem a um polígono irregular.

# Associações e Agregações

- **coordenada – linha**: uma linha possui uma ou mais coordenadas; uma coordenada é referente a uma linha.
- **coordenada – linha central**: uma linha central é composta por uma ou mais coordenadas que indicam a sua localização espacial.
- **coordenada – linha central\_f**: uma figura gráfica da linha central possui uma coordenada indicando sua posição na tela; uma coordenada é referente a uma figura gráfica da linha central.
- **coordenada – lote**: um lote é composto por uma ou mais coordenadas que indicam a sua localização espacial.
- **coordenada – lote\_f**: uma figura gráfica do lote possui uma coordenada indicando sua posição na tela; uma coordenada é referente a uma figura gráfica do lote.
- **coordenada – polilinha**: uma polilinha possui uma ou mais coordenadas; um coordenada é referente a uma polilinha.
- **linha – linha central**: indica que as coordenadas de uma linha central correspondem a uma linha.
- **linha central – linha central\_f**: uma linha central é composta por uma figura ilustrativa que a representa.
- **linha central – logradouro**: um logradouro é composto por uma ou mais linhas centrais.
- **linha central – lote**: um lote é adjacente a uma linha central. Uma linha central possui um ou mais lotes adjacentes a ela.
- **localidade – mapa cadastral de lote**: uma localidade é composta por um ou mais mapas cadastrais de lote.
- **localidade – planejamento urbano**: um projeto de planejamento urbano é composto por uma ou mais localidades.
- **logradouro – mapa cadastral de lote**: um mapa cadastral de lote é composto por um ou mais logradouros.

- **logradouro – nome alternativo:** um logradouro pode possuir zero ou mais nomes alternativos; um nome alternativo somente está associado a um logradouro.
- **lote – mapa cadastral de lote:** um mapa cadastral de lote é composto por um ou mais lotes.
- **lote – numeração:** um lote possui um ou mais números associados a ele; um número pode estar associado a um ou mais lotes.
- **lote – lote\_f:** um lote é composto por uma figura ilustrativa que o representa.
- **lote – polilinha:** indica que as coordenadas do lote correspondem a um polígono irregular.



## Apêndice B

# Descrição Semântica dos Elementos do Modelo de Dados Eletropaulo

## Entidades e Atributos

- **área de concessão**: área geográfica de atuação da Eletropaulo.
  - identif\_concessão**: identificador da área de concessão.
  - área**: área ocupada pela área de concessão (área do polígono).
  - perímetro**: perímetro da área de concessão (perímetro do polígono).
  - length**: comprimento do arco em unidades de cobertura.
- **área de regional**: subdivisão da área de concessão da empresa de acordo com a estrutura organizacional da diretoria de administração.
  - identif\_área**: identificador da superintendência regional.
  - descrição\_área**: descrição alfanumérica da área de regional.

- **divisa de lote:** elemento representativo da testada de um lote em uma quadra.

**identif\_divisa:** identificador da divisa de lote.

**tipo\_divisa:** tipo da divisa de lote: geminada, divisa, subdivisa.

**número\_divisa:** número do lote na rua.

**toponímia\_divisa:** designação do lote, caso exista.

**length:** comprimento do arco em unidades de cobertura.

- **edificação de destaque:** elemento cuja área é definida pelos limites de edificações significativas, de referência ou qualquer outra de interesse para a Eletropaulo.

**identif\_dest:** identificador da edificação de destaque.

**toponímia\_dest:** designação da edificação de destaque, caso exista.

**tipo\_edificação:** tipo da edificação de destaque: igreja, cemitério, escola pública, ...

**área:** área ocupada pela edificação de destaque (área do polígono).

**perímetro:** perímetro da identificação de destaque (perímetro do polígono).

- **eixo de logradouro:** porção do logradouro definida entre dois cruzamentos consecutivos.

**identif\_eixo:** identificador do eixo de logradouro.

**length:** comprimento do arco em unidades de cobertura.

**tipo\_lograd:** classificação dos logradouros segundo seu sentido mais amplo: rua, avenida, viela, alameda, viaduto, ponte, travessa ...

- **elemento de hidrografia:** elemento representativo de rio, córrego, ribeirão, canal, lago, lagoa, reservatório, entre outros. Corresponde ao conjunto de águas correntes ou estáveis de uma região.

**identif\_hidrog:** identificador do elemento de hidrografia.

**tipo\_hidrog:** tipo do elemento de hidrografia: rio, ribeirão, córrego, canal ...

**toponímia\_hidrog:** designação do elemento, caso exista.

**área:** área ocupada pelo elemento de hidrografia (área do polígono).

**perímetro:** perímetro do elemento de hidrografia (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **elemento de utilidade urbana:** elemento representativo de torres de alta tensão.

**identif\_util:** identificador do elemento de utilidade urbana.

**tipo\_utilidade\_urbana:** tipo do elemento de utilidade urbana: torre\_ondas, torre\_at, ...

- **limite de área:** contorno de certos elementos que são representados por áreas tais como favela, faixa de domínio, bairro, entre outros.

**identif\_la:** identificador do limite de área.

**tipo\_limite:** tipo do limite de área: favela, faixa-domínio, subestação, ...

**toponímia\_la:** designação do limite de área, caso exista.

**área:** área ocupada pelo limite de área (área do polígono).

**perímetro:** perímetro do limite de área (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **localidade técnica:** área geográfica definida para fins de atendimento técnico da rede elétrica e dos consumidores.

**identif\_lt:** identificador da localidade técnica.

**nome\_lt:** nome da localidade.

- **logradouro:** elemento representativo de via pública.

**código\_lograd:** código do logradouro.

**nome\_abreviado\_lograd:** nome abreviado do logradouro.

**nome\_lograd:** nome do logradouro.

**tit\_lograd:** identificador do título do logradouro, como exemplo marechal, comendador, doutor.

**nome\_anterior\_lograd:** nome anterior do logradouro.

**nome\_anterior\_abreviado\_lograd:** nome anterior abreviado do logradouro.

**tit\_anterior\_lograd:** identificador do título anterior do logradouro.

**data\_alteração\_lograd:** data de alteração do nome do logradouro (anterior → atual).

- **marco geodésico:** elemento que representa curvas de nível e pontos de referência de nível (ou seja, elemento indicador da altimetria de uma região).

**identif\_mg:** identificador do elemento altimétrico.

**área:** área ocupada pelo marco geodésico (área do polígono).

**perímetro:** perímetro do marco geodésico (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **município**: circunscrição administrativa autônoma do Estado.

**código\_município**: código do município.

**nome\_município**: nome completo do município.

**nome\_abreviado\_mun**: nome abreviado do município.

**sigla\_estado\_mun**: sigla do estado no qual o município se localiza.

**cod\_elpa\_mun**: código do município para a Eletropaulo.

**área**: área ocupada pelo município (área do polígono).

**perímetro**: perímetro do município (perímetro do polígono).

**length**: comprimento do arco em unidades de cobertura.

- **quadra**: área limitada pelas fachadas dos lotes.

**identif\_quadra**: identificador da quadra.

**tipo\_quadra**: tipo de linha que compõe o elemento quadra: definida, indefinida, indeterminada, fictícia.

**área**: área ocupada pela quadra (área do polígono).

**perímetro**: perímetro da quadra (perímetro do polígono).

**length**: comprimento do arco em unidades de cobertura.

- **região elétrica**: área geográfica que contém um conjunto de elementos do sistema de distribuição que podem ser considerados eletricamente isolados dos elementos de outra região elétrica.

**identif\_reg**: identificador da região elétrica.

**nome\_reg**: nome da região elétrica.

**área**: área ocupada pela região elétrica (área do polígono).

**perímetro:** perímetro da região elétrica (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **via de transporte:** elemento representativo das vias de transporte dos tipos ponte, viaduto, rodovia e túnel.

**identif\_trans:** identificador da via de transporte.

**tipo\_transporte:** tipo de via de transporte: rodovia simples asfaltada, rodovia dupla asfaltada, estrada de terra simples, auto-estrada, ...

**toponímia\_trans:** designação do elemento, caso exista.

**área:** área ocupada pela via de transporte (área do polígono).

**perímetro:** perímetro da via de transporte (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **via transporte ferroviário:** elemento representativo de via de transporte ferroviário, tal como estrada de ferro e metrô de superfície.

**identif\_ferro:** identificador do elemento de transporte não viário.

**tipo\_ferro:** tipo do elemento de transporte ferroviário: linha de metrô subterrâneo / superfície, estrada de ferro.

**length:** comprimento do arco em unidades de cobertura.

# Relacionamentos

- **área de concessão – elemento de hidrografia:** uma área de concessão contém zero ou mais elementos de hidrografia; um elemento de hidrografia está contido em uma e somente uma área de concessão.
- **área de concessão – região elétrica:** uma área de concessão contém uma ou mais áreas eletricamente isoladas; uma região elétrica pertence a uma e somente uma área de concessão.
- **área de concessão – via de transporte:** uma via de transporte pertence a uma e somente uma área de concessão; uma área de concessão pode conter zero ou mais vias de transporte.
- **área de concessão – via transporte ferroviário:** uma área de concessão contém zero ou mais vias de transporte ferroviário; uma via de transporte ferroviário está contida em uma e somente uma área de concessão.
- **área de regional – município:** uma área de regional abrange um ou mais municípios; um município pertence a uma e somente uma área de regional.
- **área de regional – região elétrica:** de acordo com a estrutura organizacional da Eletropaulo, uma região elétrica é dividida em uma ou mais áreas de regional; cada área de regional pertence a uma e somente uma região elétrica.
- **divisa de lote – quadra:** em uma quadra, existem uma ou mais divisas de lote; cada divisa de lote é localizada em uma e somente uma quadra.
- **divisa de lote – eixo de logradouro:** uma divisa de lote é adjacente a um e somente um eixo de logradouro; um eixo de logradouro possui uma ou mais divisas de lote adjacentes a ele.
- **edificação de destaque – quadra:** em uma quadra, podem existir várias edificações de destaque; cada edificação de destaque é localizada em uma e somente uma quadra.
- **eixo de logradouro – logradouro:** um logradouro é formado por um ou mais eixos de logradouro; um eixo de logradouro forma um e somente um logradouro.
- **elemento utilidade urbana – região elétrica:** uma região elétrica possui uma ou mais torres de alta tensão; uma torre de alta tensão está localizada em uma e somente uma região elétrica.

- **limite de área – município:** um município pode ser dividido em um ou mais limites de área; cada limite de área indica os limites de uma região pertencente a um e somente um município.
- **localidade técnica – município:** em um município, existem uma ou mais localidades técnicas; uma determinada localidade técnica pertence a um e somente um município.
- **marco geodésico – região elétrica:** uma região elétrica contém zero ou mais marcos geodésicos; um marco geodésico está contido em uma e somente uma região elétrica.
- **município – quadra:** um município é formado por uma ou mais quadras; uma quadra pertence a um e somente um município.



# Apêndice C

## Descrição Semântica dos Elementos do Modelo de Dados Integrado Refinado

### Classes e Atributos

- **área de concessão:** área geográfica de atuação da Eletropaulo. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_concessão:** identificador da área de concessão.

**área:** área ocupada pela área de concessão (área do polígono).

**perímetro:** perímetro da área de concessão (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **área de regional:** subdivisão da área de concessão da empresa de acordo com a estrutura organizacional da diretoria de administração. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_área:** identificador da superintendência regional.

**descrição\_área:** descrição alfanumérica da área de regional.

- **coordenada:** conjunto de pontos correspondentes às coordenadas dos objetos das classes *lote* e *linha central*. Corresponde à classe de mesmo nome no modelo da Telebrás.
- **edificação de destaque:** elemento cuja área é definida pelos limites de edificações significativas, de referência ou qualquer outra de interesse para a Eletropaulo. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_dest:** identificador da edificação de destaque.

**toponímia\_dest:** designação da edificação de destaque.

**tipo\_edificação:** tipo da edificação de destaque: igreja, cemitério, escola pública, ...

**área:** área ocupada pela edificação de destaque (área do polígono).

**perímetro:** perímetro da identificação de destaque (perímetro do polígono).

- **elemento de hidrografia:** elemento representativo de rio, córrego, ribeirão, canal, lago, lagoa, reservatório, entre outros. Corresponde ao conjunto de águas correntes ou estáveis de uma região. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_hidrog:** identificador do elemento de hidrografia.

**tipo\_hidrog:** tipo do elemento de hidrografia: rio, ribeirão, córrego, canal ...

**toponímia\_hidrog:** designação do elemento, caso exista.

**área:** área ocupada pelo elemento de hidrografia (área do polígono).

**perímetro:** perímetro do elemento de hidrografia (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **elemento de utilidade urbana:** elemento representativo de torres de alta tensão. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_util:** identificador do elemento de utilidade urbana.

**tipo\_utilidade\_urbana:** tipo do elemento de utilidade urbana: torre\_ondas, torre\_at, ...

- **limite de área:** contorno de certos elementos que são representados por áreas tais como favela, faixa de domínio, bairro, entre outros. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_la:** identificador do limite de área.

**tipo\_limite:** tipo do limite de área: favela, faixa-domínio, subestação, ...

**toponímia\_la:** designação do limite de área, caso exista.

**área:** área ocupada pelo limite de área (área do polígono).

**perímetro:** perímetro do limite de área (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **linha:** indica que as coordenadas do objeto da classe *linha central* correspondem a uma linha. Corresponde à classe de mesmo nome no modelo da Telebrás.

- **linha central:** trecho de logradouro delimitado por pontos de intersecção com outros logradouros. Esta classe corresponde à integração das classes *linha central* (modelo Telebrás) e *eixo de logradouro* (modelo Eletropaulo).

**identificador:** identificador da linha central; corresponde à integração dos atributos *identificador* (modelo Telebrás) e *identif\_eixo* (modelo Eletropaulo).

**status:** se a linha central está livre (1), se é imprópria para cabo (2), se é imprópria para duto (3) ou se está impedida (4); corresponde ao atributo de mesmo nome no modelo Telebrás.

- pavimentado:** se a linha central é pavimentada (0) ou não (1); corresponde ao atributo de mesmo nome no modelo Telebrás.
- t1:** data inicial de validade dos dados; corresponde ao atributo de mesmo nome no modelo Telebrás.
- t2:** data final de validade dos dados; corresponde ao atributo de mesmo nome no modelo Telebrás.
- length:** comprimento do arco em unidades de cobertura; corresponde ao atributo de mesmo nome no modelo Eletropaulo.
- tipo\_lograd:** classificação dos logradouros segundo seu sentido mais amplo: rua, avenida, viela, alameda, viaduto, ponte, travessa ... Corresponde à integração do atributo *tipo\_lograd* do modelo Eletropaulo e do atributo *tipo\_logradouro* da classe *logradouro* no modelo da Telebrás.
- **linha\_central\_f:** visualização gráfica do objeto da classe *linha\_central*, ou seja, figura que indica como a linha central será representada na tela. Corresponde à classe de mesmo nome no modelo da Telebrás. Os seus atributos continuam os mesmos.
    - cor:** cor do desenho a ser exibido na tela.
    - símbolo:** desenho propriamente dito.
    - hachurado:** se a figura for hachurada, qual o grau de inclinação das linhas.
  - **localidade:** circunscrição administrativa autônoma do Estado, ou seja, cidade cujo mapeamento urbano está sendo realizado. Esta classe corresponde à integração das classes *localidade* (modelo Telebrás) e *município* (modelo Eletropaulo).
    - código\_localidade:** código do município; corresponde à integração dos atributos *código\_localidade* (modelo Telebrás) e *código\_município* (modelo Eletropaulo).
    - nome\_localidade:** nome completo do município; corresponde à integração dos atributos *nome\_localidade* (modelo Telebrás) e *nome\_município* (modelo Eletropaulo).
    - nome\_abreviado\_mun:** nome abreviado do município; corresponde ao atributo de mesmo nome no modelo da Eletropaulo.

**sigla\_estado\_mun:** sigla do estado no qual o município se localiza; corresponde ao atributo de mesmo nome no modelo da Eletropaulo.

**cod\_elpa\_mun:** código do município para a Eletropaulo; corresponde ao atributo de mesmo nome no modelo da Eletropaulo.

**área:** área ocupada pelo município (área do polígono); corresponde ao atributo de mesmo nome no modelo da Eletropaulo.

**perímetro:** perímetro do município (perímetro do polígono); corresponde ao de mesmo nome no modelo da Eletropaulo.

**length:** comprimento do arco em unidades de cobertura; corresponde ao atributo de mesmo nome no modelo da Eletropaulo.

**cnl\_localidade:** código nacional da localidade (campo alfanumérico de cinco posições); corresponde ao atributo de mesmo nome no modelo da Telebrás.

**data\_ref:** data de validade dos dados; corresponde ao atributo de mesmo nome no modelo da Telebrás.

**descrição:** descrição alfanumérica da localidade; corresponde ao atributo de mesmo nome no modelo da Telebrás.

**projeção:** projeção utilizada na inserção da localidade no banco de dados; corresponde ao atributo de mesmo nome no modelo Telebrás.

**ceinf:** canto esquerdo inferior do retângulo envolvente; corresponde ao atributo de mesmo nome no modelo da Telebrás.

**cdsup:** canto direito superior do retângulo envolvente; corresponde ao atributo de mesmo nome no modelo da Telebrás.

- **localidade técnica:** área geográfica definida para fins de atendimento técnico da rede elétrica e dos consumidores. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_lt:** identificador da localidade técnica.

**nome\_lt:** nome da localidade.

- **logradouro**: conjunto de linhas centrais que representam uma rua. Esta classe corresponde à integração das classes *logradouro* (modelo Telebrás) e *logradouro* (modelo Eletropaulo).

**código\_logradouro**: código real associado ao logradouro; corresponde à integração dos atributos *código\_logradouro* (modelo Telebrás) e *código\_lograd* (modelo Eletropaulo)..

**nome\_logradouro**: nome do logradouro; corresponde à integração dos atributos *nome\_logradouro* (modelo Telebrás) e *nome\_lograd* (modelo Eletropaulo).

**título\_logradouro**: título associado ao nome do logradouro, tal como comendador, general; corresponde à integração dos atributos *título\_logradouro* (modelo Telebrás) e *tit\_lograd* (modelo Eletropaulo).

**t1**: data inicial de validade dos dados; corresponde ao atributo de mesmo nome do modelo Telebrás.

**t2**: data final de validade dos dados; corresponde ao atributo de mesmo nome do modelo Telebrás.

**nome\_abreviado\_lograd**: nome abreviado do logradouro; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

**nome\_anterior\_lograd**: nome anterior do logradouro; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

**nome\_anterior\_abreviado\_lograd**: nome anterior abreviado do logradouro; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

**tit\_anterior\_lograd**: identificador do título anterior do logradouro; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

**data\_alteração\_lograd**: data de alteração do nome do logradouro (anterior → atual) ; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

- **lote**: fachada (testada) de um lote voltada para o arruamento. Esta classe corresponde à integração das classes *lote* (modelo Telebrás) e *divisa de lote* (modelo Eletropaulo).

**num\_lote**: menor número associado a um lote; corresponde à integração dos atributos *num\_lote* (modelo Telebrás) e *número\_divisa* (modelo Eletropaulo).

**qtidade\_lote:** quantidade de números associados ao lote em questão; corresponde ao atributo de mesmo nome do modelo Telebrás.

**descrição\_lote:** descrição alfanumérica do lote; corresponde à integração dos atributos *descrição\_lote* (modelo Telebrás) e *toponímia\_divisa* (modelo Eletropaulo).

**tipo\_lote:** se o lote existe (0) ou está somente projetado (1); corresponde ao atributo de mesmo nome do modelo Telebrás.

**t1:** data inicial de validade dos dados; corresponde ao atributo de mesmo nome do modelo Telebrás.

**t2:** data final de validade dos dados; corresponde ao atributo de mesmo nome do modelo Telebrás.

**identif\_divisa:** identificador da divisa de lote; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

**tipo\_divisa:** tipo da divisa de lote: geminada, divisa, subdivisa lote; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

**length:** comprimento do arco em unidades de cobertura lote; corresponde ao atributo de mesmo nome do modelo Eletropaulo.

- **lote\_f:** visualização gráfica do objeto da classe *lote*, ou seja, figura que indica como o lote será representado na tela. Corresponde à classe de mesmo nome no modelo da Telebrás. Os seus atributos continuam os mesmos.

**cor:** cor do desenho a ser exibido na tela.

**símbolo:** desenho propriamente dito.

**hachurado:** se a figura for hachurada, qual o grau de inclinação das linhas.

- **mapa cadastral de lote:** plano de informação da localidade utilizado na aplicação. Corresponde à classe de mesmo nome no modelo da Telebrás. Os seus atributos continuam os mesmos.

**data\_ref:** data de validade dos dados.

**descrição:** descrição alfanumérica do mapa cadastral de lote.

**formato:** formato utilizado na inserção do mapa cadastral de lote no banco de dados (raster ou vector).

**escala:** escala utilizada na inserção do mapa cadastral de lote no banco de dados.

- **marco geodésico:** elemento que representa curvas de nível e pontos de referência de nível (ou seja, elemento indicador da altimetria de uma região). Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_mg:** identificador do elemento altimétrico.

**área:** área ocupada pelo marco geodésico (área do polígono).

**perímetro:** perímetro do marco geodésico (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **nome alternativo:** nome alternativo associado a um logradouro. Em outras palavras, um logradouro pode possuir um nome e ser conhecido localmente por outro. Corresponde à classe de mesmo nome no modelo da Telebrás. Os seus atributos continuam os mesmos.

**nome\_alternativo:** nome alternativo do logradouro.

**tipo\_alternativo:** tipo do logradouro.

**título\_alternativo:** título associado ao nome do logradouro.

- **numeração:** representa os números que podem estar associados a um lote. Corresponde à classe de mesmo nome no modelo da Telebrás. Os seus atributos continuam os mesmos.

**num:** número do lote.

**descrição\_num:** descrição alfanumérica do número em questão.



- **planejamento urbano**: aplicação geográfica sendo implementada. Corresponde à classe de mesmo nome no modelo da Telebrás. Os seus atributos continuam os mesmos.

**data\_ref**: data de validade dos dados.

**nome**: nome do projeto.

**objetivo**: objetivo do projeto.

**coordenador**: coordenador do projeto.

- **polilinha**: indica que as coordenadas do objeto da classe *lote* correspondem a um polígono irregular. Corresponde à classe de mesmo nome no modelo Telebrás.

- **quadra**: área limitada pelas fachadas dos lotes. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_quadra**: identificador da quadra.

**tipo\_quadra**: tipo de linha que compõe o elemento quadra: definida, indefinida, indeterminada, fictícia.

**área**: área ocupada pela quadra (área do polígono).

**perímetro**: perímetro da quadra (perímetro do polígono).

**length**: comprimento do arco em unidades de cobertura.

- **região elétrica**: área geográfica que contém um conjunto de elementos do sistema de distribuição que podem ser considerados eletricamente isolados dos elementos de outra região elétrica. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_reg**: identificador da região elétrica.

**nome\_reg**: nome da região elétrica.

**área**: área ocupada pela região elétrica (área do polígono).

**perímetro:** perímetro da região elétrica (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **via de transporte:** elemento representativo das vias de transporte dos tipos ponte, viaduto, rodovia e túnel. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_trans:** identificador da via de transporte.

**tipo\_transporte:** tipo de via de transporte: rodovia simples asfaltada, rodovia dupla asfaltada, estrada de terra simples, auto-estrada, ...

**toponímia\_trans:** designação do elemento, caso exista.

**área:** área ocupada pela via de transporte (área do polígono).

**perímetro:** perímetro da via de transporte (perímetro do polígono).

**length:** comprimento do arco em unidades de cobertura.

- **via transporte ferroviário:** elemento representativo de via de transporte ferroviário, tal como estrada de ferro e metrô de superfície. Corresponde à classe de mesmo nome no modelo da Eletropaulo. Os seus atributos continuam os mesmos.

**identif\_ferro:** identificador do elemento de transporte não viário.

**tipo\_ferro:** tipo do elemento de transporte ferroviário: linha de metrô subterrâneo / superfície, estrada de ferro.

**length:** comprimento do arco em unidades de cobertura.

# Associações e Agregações

- **área de concessão – elemento de hidrografia:** uma área de concessão contém zero ou mais elementos de hidrografia; um elemento de hidrografia está contido em uma e somente uma área de concessão.
- **área de concessão – região elétrica:** uma área de concessão contém uma ou mais áreas eletricamente isoladas; uma região elétrica pertence a uma e somente uma área de concessão.
- **área de concessão – via de transporte:** uma via de transporte pertence a uma e somente uma área de concessão; uma área de concessão pode conter zero ou mais vias de transporte.
- **área de concessão – via transporte ferroviário:** uma área de concessão contém zero ou mais vias de transporte ferroviário; uma via de transporte ferroviário está contida em uma e somente uma área de concessão.
- **área de regional – localidade:** uma área de regional abrange um ou mais municípios (localidades); um município pertence a uma e somente uma área de regional (localidade).
- **área de regional – região elétrica:** de acordo com a estrutura organizacional da Eletropaulo, uma região elétrica é dividida em uma ou mais áreas de regional; cada área de regional pertence a uma e somente uma região elétrica.
- **coordenada – linha:** uma linha possui uma ou mais coordenadas; uma coordenada é referente a uma linha.
- **coordenada – linha central:** uma linha central é composta por uma ou mais coordenadas que indicam a sua localização espacial.
- **coordenada – linha central\_f:** uma figura gráfica da linha central possui uma coordenada indicando sua posição na tela; uma coordenada é referente a uma figura gráfica da linha central.
- **coordenada – lote:** um lote é composto por uma ou mais coordenadas que indicam a sua localização espacial.
- **coordenada – lote\_f:** uma figura gráfica do lote possui uma coordenada indicando sua posição na tela; uma coordenada é referente a uma figura gráfica do lote.

- **coordenada – polilinha:** uma polilinha possui uma ou mais coordenadas; um coordenada é referente a uma polilinha.
- **edificação de destaque – quadra:** em uma quadra, podem existir várias edificações de destaque; cada edificação de destaque é localizada em uma e somente uma quadra.
- **elemento utilidade urbana – região elétrica:** uma região elétrica possui uma ou mais torres de alta tensão; uma torre de alta tensão está localizada em uma e somente uma região elétrica.
- **limite de área – localidade:** um município (localidade) pode ser dividido em um ou mais limites de área; cada limite de área indica os limites de uma região pertencente a um e somente um município (localidade).
- **linha central – linha:** indica que as coordenadas de uma linha central correspondem a uma linha.
- **linha central – linha central\_f:** uma linha central é composta por uma figura ilustrativa que a representa.
- **linha central – logradouro:** um logradouro é composto por uma ou mais linhas centrais. Esta associação representa a integração das associações entre as classes *logradouro* e *linha central* do modelo Telebrás e *logradouro* e *eixo de logradouro* do modelo Eletropaulo.
- **linha central – lote:** um lote é adjacente a uma linha central. Uma linha central possui um ou mais lotes adjacentes a ela. Esta associação corresponde à integração das associações entre *lote* e *linha central* do modelo Telebrás e *divisa de lote* e *eixo de logradouro* do modelo Eletropaulo.
- **localidade – localidade técnica:** em um município (localidade), existem uma ou mais localidades técnicas; uma determinada localidade técnica pertence a um e somente um município (localidade).
- **localidade – mapa cadastral de lote:** uma localidade é composta por um ou mais mapas cadastrais de lote.
- **localidade – planejamento urbano:** o projeto de planejamento urbano é composto por uma ou mais localidades.
- **logradouro – nome alternativo:** um logradouro pode possuir zero ou mais nomes alternativos; um nome alternativo somente está associado a um logradouro.
- **logradouro – mapa cadastral de lote:** um mapa cadastral de lote é composto por um ou mais logradouros.

- **lote – numeração**: um lote possui um ou mais números associados a ele; um número pode estar associado a um ou mais lotes.
- **lote – lote\_f**: um lote é composto por uma figura ilustrativa que o representa.
- **lote – polilinha**: indica que as coordenadas do lote correspondem a um polígono irregular.
- **lote – quadra**: uma quadra é composta por um ou mais lotes.
- **mapa cadastral de lote – quadra**: um mapa cadastral de lote é composto por um ou mais quadras.
- **marco geodésico – região elétrica**: uma região elétrica contém zero ou mais marcos geodésicos; um marco geodésico está contido em uma e somente uma região elétrica.