

SERGIO MURILO STEMPLIUC

**MODELAGEM DE RESTRIÇÕES DE INTEGRIDADE
ESPACIAIS EM APLICAÇÕES DE REDE
ATRAVÉS DO MODELO UML-GEOFRAME**

Dissertação apresentada à
Universidade Federal de Viçosa,
como parte das exigências do
Programa de Pós-Graduação em
Ciência da Computação, para
obtenção do título de *Magister
Scientiae*.

VIÇOSA
MINAS GERAIS - BRASIL
2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

SERGIO MURILO STEMPLIUC

**MODELAGEM DE RESTRIÇÕES DE INTEGRIDADE
ESPACIAIS EM APLICAÇÕES DE REDE
ATRAVÉS DO MODELO UML-GEOFRAME**

Dissertação apresentada à
Universidade Federal de Viçosa,
como parte das exigências do
Programa de Pós-Graduação em
Ciência da Computação, para
obtenção do título de *Magister
Scientiae*.

APROVADA: 01 de agosto de 2008

Karla Albuquerque de Vasconcelos Borges
(Co-Orientadora)

Vladimir Oliveira Di Iorio
(Co-Orientador)

Clodoveu Augusto Davis Junior

Mauro Nacif Rocha

Jugurta Lisboa Filho
(Orientador)

Dedico essa dissertação à minha família e minha namorada, por sempre dizerem que era possível mesmo quando não acreditava mais

AGRADECIMENTOS

Não poderia deixar de agradecer em primeiro lugar a Deus. Em momentos de alegria e dificuldade, Ele sempre estava ao meu lado para me guiar e colocar as pessoas certas em meu caminho.

Aos meus pais, pelo carinho, pelas lições, pelo sacrifício para garantir sempre os melhores estudos e por compreenderem perfeitamente minhas dificuldades durante esse período de mestrado, dizendo sempre as palavras certas nos momentos certos. Ao meu irmão, meu melhor amigo, por toda ajuda durante minha graduação e pós-graduação, adiando seus projetos para que os meus se tornassem realidade.

À minha namorada, que mesmo a distância enfrentou comigo dia após dia todos os obstáculos desse período. Lado a lado superamos todas as dificuldades, amadurecemos e hoje temos a certeza de que estamos preparados para que finalmente possamos ficar juntos.

Ao meu orientador Jugurta Lisboa Filho deixo um agradecimento especial, por ter se tornado um amigo de confiança em tão pouco tempo de convivência. Agradeço seu equilíbrio entre os elogios, que sempre me motivaram, e as críticas, feitas sempre com muito respeito. Em raros momentos tive a honra de conhecer um profissional tão capacitado. Sem você não seria possível desenvolver este trabalho.

Ao meu amigo e orientador na graduação, Manoel Palhares, pela confiança, por tornar possível meu ingresso no mestrado e pelo conhecimento e experiência transmitidos de forma única. Aos novos amigos, pela ótima convivência, pelos inúmeros favores atendidos e por me ajudarem durante períodos difíceis.

À minha co-orientadora Karla Borges, por suas contribuições que ajudaram na identificação e correção de falhas durante o desenvolvimento deste trabalho.

Sou grato também à UFV e aos professores e técnicos do Departamento de Informática, em especial ao Prof. José Luis, pelo extraordinário ensino que trouxe profundas mudanças, e ao Altino, sempre muito educado, prestativo e organizado.

À CAPES pelo fundamental apoio financeiro que tornou possível minha moradia em Viçosa e o convívio próximo com meu orientador e amigos, refletindo positivamente em todo trabalho desenvolvido. Aos funcionários do SAAE de Viçosa que forneceram informações importantes em um dos estudos de caso utilizados.

Muito obrigado a todos!

BIOGRAFIA DO AUTOR

Bacharel em Ciência da Computação pela Pontifícia Universidade Católica de Minas Gerais (PUC-Minas) no ano de 2005 e mestrando em Ciência da Computação na área de Bancos de Dados Geográficos pela Universidade Federal de Viçosa (UFV) com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Possui experiência em desenvolvimento de sistemas para internet, pesquisas em Ciência da Computação e Ciência da Informação, modelagem de sistemas de informação e bancos de dados geográficos, além de atuar como professor na disciplina de introdução à programação durante o mestrado e como monitor na disciplina de banco de dados durante a graduação.

SUMÁRIO

AGRADECIMENTOS	iii
BIOGRAFIA DO AUTOR	iv
SUMÁRIO	v
LISTA DE FIGURAS	vii
LISTA DE TABELAS	viii
LISTA DE ABREVIATURAS	ix
RESUMO	x
ABSTRACT	xi
1 Introdução	1
1.1 Motivação.....	1
1.2 Objetivo geral.....	3
1.3 Objetivos específicos	3
1.4 Estrutura da dissertação.....	3
2 Modelagem conceitual de restrições de integridade espaciais	5
2.1 Representação de dados espaciais.....	5
2.2 Modelagem conceitual de bancos de dados geográficos.....	8
2.3 Modelos conceituais para bancos de dados geográficos.....	10
2.3.1 <i>Modelo GeoOOA</i>	11
2.3.2 <i>Modelo OMT-G</i>	12
2.3.3 <i>Modelo MADS</i>	14
2.3.4 <i>Modelo UML-GeoFrame</i>	16
2.4 Taxonomia de restrições de integridade espaciais	21
2.5 Object constraint language.....	25
2.5.1 <i>Contexto</i>	28
2.5.2 <i>Instância de contexto</i>	29
2.5.3 <i>Valores iniciais e regras de derivação</i>	29
2.5.4 <i>Operações de consulta</i>	30
2.5.5 <i>Invariáveis</i>	30
2.5.6 <i>Coleções de objetos</i>	31
2.5.7 <i>Operações de iteração</i>	32
2.5.8 <i>Pré-condições e pós-condições</i>	34
2.5.9 <i>Comentários</i>	35
2.6 Extensão da OCL para especificação de restrições de integridade espaciais	35

3	Extensão do modelo UML-GeoFrame para modelagem de rede .	39
3.1	Inclusão de propriedades de rede em classes geométricas.....	39
3.2	Especialização a partir de classes geométricas	40
3.3	Hierarquia de classes para modelagem de rede.....	41
3.4	Reformulação da hierarquia de classes para modelagem de rede	43
3.5	Novas classes para redes bidirecionais e unidirecionais	45
3.6	Estereótipos de rede	46
3.7	Definição dos elementos de representação de rede.....	46
3.8	Estrutura de dados para grafos	47
	3.8.1 Restrições de integridade para grafos bidirecionais	48
	3.8.2 Restrições de integridade para grafos unidirecionais	50
4	Estudos de caso para modelagem de aplicações de rede.....	51
4.1	Estudo de caso: Sistema de distribuição de água.....	51
4.2	Comparação entre os modelos UML-GeoFrame e OMT-G	58
4.3	Estudo de caso: Sistema de distribuição de energia elétrica.....	61
4.4	Comparação entre os modelos UML-GeoFrame e GeoOOA	63
5	Conclusões e trabalhos futuros.....	67
5.1	Contribuições deste trabalho	69
5.2	Trabalhos futuros	70
	Referências bibliográficas	71

LISTA DE FIGURAS

Figura 2.1 – Objetos unidimensionais.....	7
Figura 2.2 – Objetos bidimensionais.....	8
Figura 2.3 – Exemplos de relacionamentos de rede do modelo GeoOOA	12
Figura 2.4 – Exemplos de relacionamentos de rede do modelo OMT-G.....	13
Figura 2.5 – Exemplos de relacionamentos simples e espacial do modelo OMT-G .	13
Figura 2.6 – Exemplos de relacionamentos de transição e geração do modelo MADS	15
Figura 2.7 – Framework GeoFrame	16
Figura 2.8 – Estereótipos de generalização do modelo UML-GeoFrame.....	18
Figura 2.9 – Estereótipos de representação espacial do modelo UML-GeoFrame....	19
Figura 2.10 – Estereótipos de relacionamento espacial do UML-GeoFrame	20
Figura 2.11 – Exemplos de uso dos estereótipos espaço-temporais	21
Figura 2.12 – Classificação hierárquica de restrições de integridade espaciais.....	21
Figura 2.13 – Taxonomia de restrições de integridade espaciais.....	24
Figura 2.14 – Esquema conceitual UML para um banco de dados de companhia	27
Figura 2.15 – Contexto de expressões OCL no diagrama de classes da UML	28
Figura 2.16 – Composição espacial entre Lote e Quadra.....	37
Figura 3.1 – Especialização de Nó e Arco a partir de Ponto e Linha	40
Figura 3.2 – Hierarquia de classes para modelagem de redes.....	41
Figura 3.3 – Reformulação da hierarquia de classes para modelagem de redes	43
Figura 3.4 – Interpretação da representação de classes com uso de agregações espaciais.....	44
Figura 3.5 – Proposta final de extensão do GeoFrame para modelagem de redes....	45
Figura 3.6 – Estereótipos de generalização e representação de rede	46
Figura 3.7 – Diagramas de grafos bidirecionais e unidirecionais	48
Figura 4.1 – Ilustração de um sistema de distribuição de água.....	53
Figura 4.2 – Diagrama de classes de uma companhia de distribuição de água elaborado utilizando os construtores de rede do UML-GeoFrame	54
Figura 4.3 – Diagrama de classes de uma companhia de distribuição de água sem os estereótipos de generalização do UML-GeoFrame	56
Figura 4.4 – Diagrama de classes de uma companhia de distribuição de água elaborado através dos construtores de rede do OMT-G	59
Figura 4.5 – Ilustração de um sistema de distribuição de energia elétrica.....	62
Figura 4.6 – Diagrama de classes para distribuição de energia elétrica elaborado através dos construtores de rede do UML-GeoFrame.....	62
Figura 4.7 – Modelagem de companhia de energia elétrica através do GeoOOA.....	64

LISTA DE TABELAS

Tabela 4.1 - Comparação entre os modelos OMT-G, GeoOOA e UML-GeoFrame. 66

LISTA DE ABREVIATURAS

- GeoOOA – Geographic Object-Oriented Analysis
- MADS - Modeling for Application Data with Spatio-temporal features
- OCL - Object Constraint Language
- OMG - Object Management Group
- OMT-G - Object Modeling Technique for Geographic Applications
- OOA - Object-Oriented Analysis
- SGBD - Sistema Gerenciador de Banco de Dados
- SGBDG - Sistema Gerenciador de Banco de Dados Geográficos
- SGBD-OO - Sistema Gerenciador de Banco de Dados Orientados a Objetos
- SGBD-OR - Sistema Gerenciador de Banco de Dados Objeto-Relacional
- SIG - Sistema de Informação Geográfica
- STER - Spatio-Temporal Entity-Relationship
- UML - Unified Modeling Language

RESUMO

STEMPLIUC, Sergio Murilo, M.Sc., Universidade Federal de Viçosa, agosto de 2008. **Modelagem de restrições de integridade espaciais em aplicações de rede através do modelo UML-GeoFrame.** Orientador: Jugurta Lisboa Filho. Co-Orientadores: Karla Albuquerque de Vasconcelos Borges e Vladimir Oliveira Di Iorio.

Este trabalho está situado na área de sistemas de informação geográfica, especificamente abordando a modelagem de bancos de dados geográficos em domínios em que são fundamentais as informações referentes à conectividade de seus elementos. É tratada também a especificação de restrições de integridade sobre os elementos de uma rede, pois acredita-se que dessa forma seja possível alcançar uma maior qualidade dos dados geográficos. Como a ausência de construtores de rede no modelo UML-GeoFrame impossibilitava representar adequadamente os relacionamentos de conectividade entre os elementos da aplicação, este trabalho teve como objetivo propor uma extensão ao modelo para que fossem disponibilizados os construtores necessários para a elaboração de diagramas de classe envolvendo elementos de rede. Foi proposto também o uso da *Object Constraint Language* (OCL) para especificação formal de restrições de integridade. A fundamentação teórica possibilitou que fossem compreendidos os tipos de representações de dados espaciais, a modelagem conceitual de bancos de dados geográficos, a importância de uma linguagem formal para inclusão de informações complementares ao esquema conceitual, e a classificação dos diferentes tipos de restrições de integridade espaciais. A proposta foi organizada apresentando-se as diversas etapas da evolução do *framework* GeoFrame, apontando as vantagens e desvantagens encontradas em suas versões preliminares. A viabilidade da proposta foi evidenciada através de dois estudos de caso, com a elaboração de esquemas conceituais para os sistemas de distribuição de água e energia elétrica. Para complementar, o primeiro estudo de caso foi acompanhado de uma comparação entre os construtores de rede dos modelos UML-GeoFrame e OMT-G, e o segundo estudo de caso comparou os modelos UML-GeoFrame e GeoOOA. O objetivo foi apresentar as semelhanças, vantagens e desvantagens identificadas entre os modelos.

ABSTRACT

STEMPLIUC, Sergio Murilo, M.Sc., Universidade Federal de Viçosa, August of 2008. **Modeling spatial integrity constraints for network applications using the UML-GeoFrame data model.** Adviser: Jugurta Lisboa Filho. Co-Advisers: Karla Albuquerque de Vasconcelos Borges and Vladimir Oliveira Di Iorio.

This work is in the field of geographical information systems, specifically approaching geographic database modeling in domains in which information referring to element connectivity is fundamental. It also deals with specification of integrity constraints on elements of a network, since we believe that it is possible to obtain better quality of geographic data in this way. Because of the absence of network constructors in the UML-GeoFrame, data modeling lacked the suitable representation tools for connectivity relationships between the elements of the application. Therefore, the objective of this work was to propose an extension to the model to include the necessary constructors to design class diagrams involving network elements. Object Constraint Language (OCL) was also proposed for formal specification of integrity constraints. The theoretical foundation made it possible to understand types of representations of spatial data, conceptual modeling of geographic databases, the importance of a formal language for inclusion of complementary information to the conceptual scheme, and the classification of the different types of spatial integrity constraints. The proposal presents the several evolution stages of the GeoFrame framework, pointing out advantages and disadvantages in its first versions. The feasibility of the proposal was evaluated using two case studies; designing conceptual schemes for water and electric power distribution systems. As complementary information, the first study case was tied in with a comparison between the network constructors of the UML-GeoFrame and OMT-G data models, and the second case study between the UML-GeoFrame and GeoOOA data models. The aim was to present the similarities, advantages and disadvantages identified between the models.

1 INTRODUÇÃO

Este trabalho aborda os problemas encontrados durante a modelagem conceitual de bancos de dados geográficos para domínios que necessitem de informações referentes à conectividade de seus elementos. É considerada também a especificação de restrições de integridade em esquema conceitual, como mecanismo de melhoria da qualidade dos dados geográficos quando estes são inseridos ou modificados em um Sistema de Informação Geográfica (SIG).

Neste capítulo introdutório é apresentada a motivação para o desenvolvimento desta pesquisa através dos conceitos e da importância da especificação de restrições de integridade em SIG, seguida pela hipótese deste trabalho, seus objetivos geral e específicos, e a estrutura da dissertação.

1.1 Motivação

Para os usuários de SIG, a qualidade dos dados espaciais se tornou um assunto de grande importância diante de sua enorme quantidade (COCKCROFT, 1997). “Os usuários de dados espaciais normalmente desconhecem a precisão dos dados espaciais utilizados. Eles baseiam suas análises utilizando esses dados na suposição de que estes estão livres de erros ou que os erros são mantidos em um nível aceitável” (MARBLE, 1990, apud COCKCROFT, 1997, p. 2). Segundo Cockcroft (1997), isso levanta a questão se algum nível de erro é aceitável e, caso seja, em qual critério esse nível deveria se basear.

Em muitas aplicações, a qualidade dos dados da fonte é de grande importância para proporcionar resultados precisos para as consultas. Segundo Elmasri e Navathe (2005), esse problema é particularmente significativo no contexto de SIG devido à variedade de dados, fontes e técnicas de medida envolvidas, e da precisão esperada pelos usuários das aplicações.

A especificação de restrições de integridade possui então o objetivo de melhorar a qualidade dos dados armazenados. Em qualquer estado do banco de dados, todas as restrições devem ser satisfeitas para que os dados estejam de acordo com a qualidade desejada. As restrições de integridade comumente suportadas e utilizadas em bancos de dados convencionais são as de domínio, integridade de

entidade, estrutura de atributo e integridade referencial (ELMASRI e NAVATHE, 2005).

Uma técnica utilizada para se obter o aumento da qualidade dos dados armazenados em bancos de dados consiste em impor restrições de integridade durante a entrada de dados. Outra é realizar a “limpeza” de registros inseridos e que não atendam às restrições estabelecidas. Em qualquer uma, os dados contidos em uma base não devem ser contraditórios às regras definidas dentro do Sistema Gerenciador de Banco de Dados (SGBD) (comuns a qualquer aplicação) e àquelas definidas pelo usuário (específicas de um aplicativo).

O problema de restrições de integridade está bem consolidado na área de bancos de dados convencionais. Porém, como os dados utilizados em SIG possuem características geográficas em relação a sua localização e relacionamentos espaciais, um novo desafio foi estabelecido para o campo de restrições de integridade. É necessário agora estabelecer regras capazes de tratar as peculiaridades dos dados espaciais, em especial os relacionamentos entre os objetos espaciais.

Atualmente, as pesquisas no campo de restrições de integridade espaciais têm sido realizadas principalmente no desenvolvimento de modelos conceituais específicos para SIG (KÖSTERS, PAGEL e SIX, 1997), (BORGES, DAVIS e LAENDER, 2001, 2005), (LISBOA FILHO e IOCHPE, 2008) e (SPACCAPIETRA, PARENT e ZIMÁNYI, 2006). O modelo OMT-G (BORGES, DAVIS e LAENDER, 2001, 2005), por exemplo, suporta a modelagem de dados espaciais, possuindo mecanismos para modelar estrutural, geométrica e semanticamente as restrições de integridade espaciais em nível conceitual. Outro modelo, UML-GeoFrame (LISBOA FILHO e IOCHPE, 2008), também fornece suporte à modelagem de dados espaciais, no qual as associações entre entidades representando elementos da realidade podem ser expressas através do uso de estereótipos textuais para representar as restrições espaciais que se deseja impor, ou associações sem estereótipos para representar associações semânticas.

Observa-se, no entanto, a necessidade de construtores que possibilitem especificar restrições de integridade em esquemas conceituais de forma diferente ao uso de expressões textuais pré-definidas. Além disso, a ausência de construtores de rede no UML-GeoFrame impossibilita evidenciar a importância da conectividade entre os elementos da aplicação, dificultando a elaboração de esquemas conceituais de dados para essas aplicações.

1.2 Objetivo geral

Estender o modelo UML-GeoFrame para que este forneça os construtores necessários para a elaboração de diagramas de classe envolvendo elementos de rede. Através dos novos construtores deve ser possível realizar a modelagem de bancos de dados geográficos para domínios como malha hidrográfica de uma região, rede elétrica de uma cidade, malha de circulação viária urbana ou qualquer outra aplicação que tenha a necessidade de mapear a conectividade de seus elementos.

Incluir no UML-GeoFrame o uso de uma linguagem formal para especificação de restrições de integridade em situações nas quais seja impossível alcançar os objetivos desejados através apenas do uso dos construtores de classes e associações provenientes do diagrama de classes da Linguagem de Modelagem Unificada (*UML - Unified Modeling Language*) (OMG, 2007).

1.3 Objetivos específicos

Estender o modelo UML-GeoFrame inclui o estudo de modelos conceituais específicos para modelagem de bancos de dados geográficos e principalmente o estudo de trabalhos que abordam a modelagem de rede. A extensão deve ser feita através da inclusão de classes e relacionamentos no modelo UML-GeoFrame, mantendo seus princípios e sua compatibilidade com a especificação do metamodelo da UML.

Este trabalho inclui também o estudo de uma linguagem formal compatível com a UML e sua aplicação na modelagem de bancos de dados geográficos. O trabalho deve apresentar as vantagens dessa linguagem sobre a linguagem natural e sobre as expressões textuais pré-definidas do UML-GeoFrame.

Pretende-se também mostrar a viabilidade do modelo UML-GeoFrame estendido por meio do desenvolvimento de estudos de caso que envolvam aplicações de rede de infra-estrutura.

1.4 Estrutura da dissertação

O primeiro capítulo deste trabalho compreende a introdução à pesquisa realizada. O segundo capítulo apresenta o referencial teórico utilizado para

compreensão das estruturas de dados geográficos, os modelos conceituais existentes específicos para a modelagem de aplicações de SIG e o modo como as restrições de integridade espaciais são especificadas. O terceiro capítulo apresenta as propostas para que o modelo UML-GeoFrame suporte tanto a modelagem quanto a especificação de restrições de integridade envolvendo elementos de rede. O quarto capítulo apresenta dois estudos de caso para avaliar o modelo estendido durante a elaboração de esquemas conceituais para aplicações de rede. Por fim, o quinto capítulo apresenta as conclusões e trabalhos futuros.

2 MODELAGEM CONCEITUAL DE RESTRIÇÕES DE INTEGRIDADE ESPACIAIS

O objetivo deste capítulo é apresentar um embasamento teórico para que possa ser compreendido o contexto e os conceitos utilizados pelo UML-GeoFrame e a extensão proposta neste trabalho. A seção 2.1 apresenta os tipos de representação de dados espaciais utilizados durante a interpretação dos fenômenos geográficos. A seção 2.2 faz uma breve descrição sobre modelagem conceitual. A seção 2.3 descreve alguns modelos conceituais para dados geográficos considerados importantes para o desenvolvimento deste trabalho. A seção 2.4 apresenta taxonomias de restrições de integridade espaciais. A seção 2.5 faz um resumo sobre a linguagem de restrições de objetos (OCL). A seção 2.6 descreve uma extensão à OCL, encontrada na literatura.

2.1 Representação de dados espaciais

Ao se utilizar um banco de dados geográfico deve-se considerar o modo como os fenômenos geográficos de uma região podem ser percebidos. Segundo Laurini e Thompson (1992) e Rigaux, Scholl e Voisard (2001), esses dados podem ser classificados de quatro maneiras:

- Dados convencionais: São aqueles que não possuem propriedades geográficas. Por exemplo, o nome de uma rua ou a população de um país.
- Dados geográficos na visão de objetos ou entidades (geo-objetos): A utilização desse tipo de dado implica na divisão da região geográfica em entidades que devem ser disjuntas e identificadas de maneira única. Exemplos são as ruas, lotes, bairros, rios e rodovias.
- Dados geográficos na visão de campos (geo-campos): Através desse tipo de dado é possível identificar um valor para cada posição da região geográfica. Exemplos são os valores indicativos de temperatura, pressão, umidade do ar, altitude e vegetação.
- Dados geográficos na visão de redes: Esse tipo de dado utiliza apenas pontos e linhas para representar a conectividade entre elementos geográficos. Pode-se representar, por exemplo, a adjacência entre unidades administrativas e as redes de malha viária, de eletricidade e de telefonia.

Para que se possam representar fenômenos geográficos na visão de objetos, são utilizados os elementos geométricos ponto, objeto linear e objeto de superfície. Esses elementos, por sua vez, podem ser agregados para representar fenômenos observados na visão de campos, sendo definidos por Rigaux, Scholl e Voisard (2001) como:

- Ponto: Objeto zero-dimensional, utilizado tanto na identificação precisa de fenômenos ou elementos dentro de uma região quanto para representar uma região cujo formato ou área não são considerados importantes. Exemplos: edificações, pontos turísticos, cidades, cruzamentos, postes e antenas.
- Objeto linear: Objeto unidimensional, utilizado na representação de segmentos de linha e polilinhas. Uma polilinha é um conjunto finito de segmentos de linha, tal que cada ponto final de segmento (conhecido como vértice) é compartilhado por exatamente dois segmentos, exceto para os dois pontos finais (conhecidos como pontos extremos), os quais pertencem apenas a um único segmento.

A Figura 2.1 ilustra diversos tipos de polilinhas: (a) Segmento de reta ou aresta, contendo pontos extremos; (b) Polilinha simples, contendo vértices e pontos extremos; (c) Polilinha não-simples, contendo vértices, pontos extremos e interseção entre arestas não consecutivas; (d) Polilinha fechada simples, contendo somente vértices; (e) Polilinha monotônica, na qual, segundo a linha γ , cada linha γ' ortogonal a γ deve interceptar a polilinha em um único ponto; (f) Polilinha não-monotônica, definida através da impossibilidade de se traçar linhas ortogonais a um eixo tal que estas interceptem a polilinha em um único ponto.

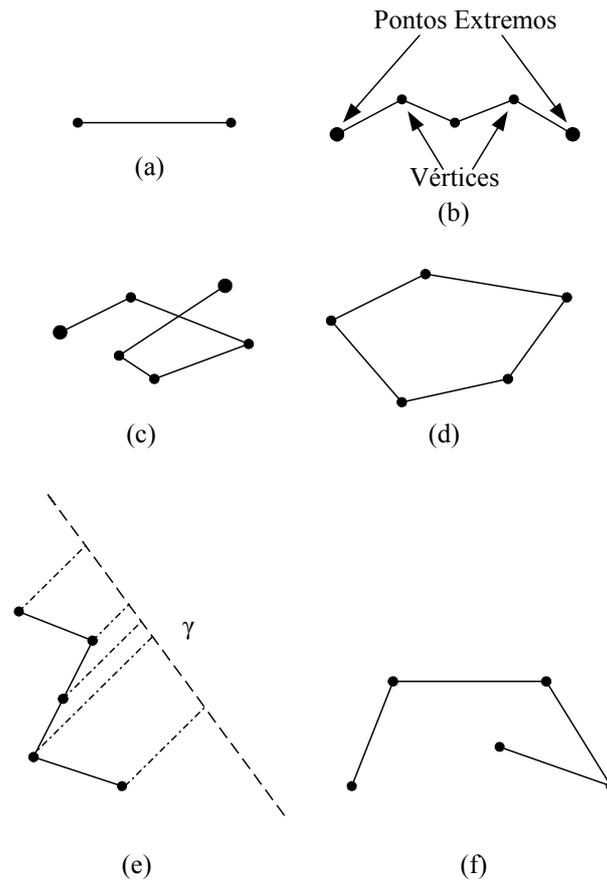


Figura 2.1 – Objetos unidimensionais
 Fonte: Rigaux, Scholl e Voisard (2001)

- Objeto de superfície: Objeto bidimensional, utilizado para representar uma região cujo formato ou área são considerados importantes. A principal representação geométrica é o polígono, definido por Rigaux, Scholl e Voisard (2001) como uma região do plano contornada por uma polilinha fechada, conhecida como fronteira.

A Figura 2.2 ilustra os seguintes tipos de objetos bidimensionais: (a) Polígono simples, cuja fronteira é uma polilinha simples; (b) Polígono não-simples, cuja fronteira é uma polilinha não-simples; (c) Polígono convexo, no qual qualquer par de pontos dentro do polígono deve possibilitar se traçar um segmento de reta que fique totalmente incluso no polígono; (d) Polígono monotônico, cuja fronteira pode ser dividida em duas polilinhas monotônicas de acordo com um eixo; (e) Polígono com buraco, definido como um polígono no qual existe um segundo polígono interno com

propriedades distintas; (f) Região, definida como um conjunto de polígonos que não se interceptam.

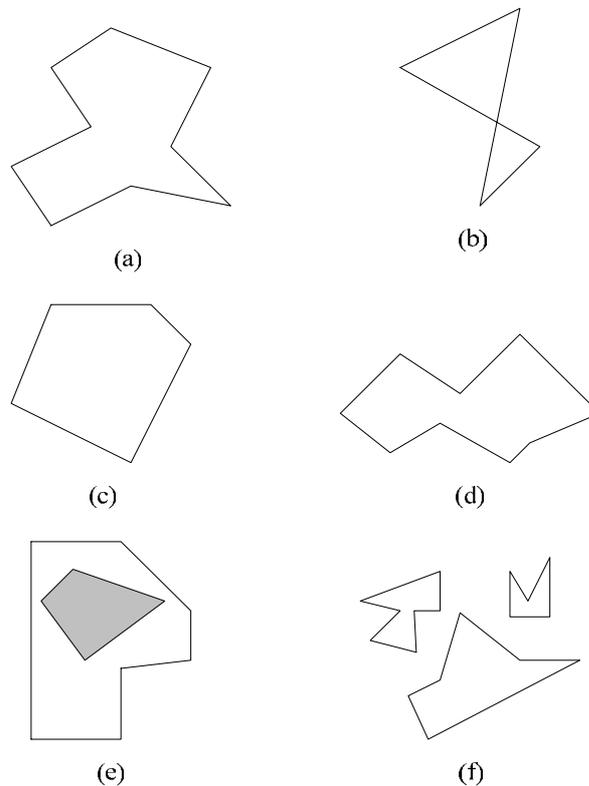


Figura 2.2 – Objetos bidimensionais
Fonte: Rigaux, Scholl e Voisard (2001)

Este trabalho possui foco em nível conceitual e aborda somente a visão de objetos e de redes, limitando-se, portanto, aos conceitos de representação espacial apresentados acima. Essas representações são conceitos abstratos e estão relacionadas à forma como a realidade pode ser interpretada e representada pelo projetista e usuários, mas internamente em um sistema computacional, o nível de detalhes e o número de representações possíveis são muito maiores. As representações de fenômenos na visão de campo e em sistemas computacionais são apresentadas em Laurini e Thompson (1992) e Rigaux, Scholl e Voisard (2001).

2.2 Modelagem conceitual de bancos de dados geográficos

“Um esquema conceitual é uma representação da visão que o usuário tem das informações gerenciadas pelo sistema” (WAZLAWICK, 2004).

Quando se deseja criar um sistema de informação de qualquer espécie, computacional ou não, deve-se inicialmente restringir a realidade àquela

correspondente ao problema. A modelagem conceitual auxilia os projetistas justamente neste ponto, abordando somente os elementos pertinentes ao problema.

Um modelo conceitual é, portanto, um mecanismo para que seja estudado o problema e não deve ser confundido com a arquitetura de software ou com o modelo de representação de dados, pois estes pertencem ao domínio da solução. A solução será derivada do modelo conceitual, mas não deve ser considerada durante esta etapa (WAZLAWICK, 2004). Segundo Laurini e Thompson (1992), embora seja uma abstração do mundo real, o resultado da modelagem conceitual é supostamente bem concreto em sua natureza, consistindo de representações esquemáticas de fenômenos e como eles se relacionam. A modelagem conceitual de banco de dados não foge à regra, ou seja, não deve considerar se a solução será expressa por sistemas computacionais ou manuais, apenas aborda os elementos de informação essenciais ao sistema.

A modelagem de bancos de dados geográficos enfrenta um desafio maior quando comparada à modelagem de bancos de dados convencionais. Como visto na seção anterior, dados geográficos são mais complexos e não podem ser representados utilizando-se apenas dados alfanuméricos, o que torna a modelagem também mais complexa. Além disso, os dados geográficos aumentaram o número de formas com as quais os usuários podem interpretar a realidade, podendo até mesmo interpretar um mesmo fenômeno de múltiplas formas dependendo, por exemplo, da escala utilizada.

Um modelo conceitual de dados geográficos tem então, como objetivo, fornecer construtores capazes de expressar as necessidades dos usuários. Antes da existência deste tipo de modelo, os usuários eram obrigados a transformar a forma como a realidade era percebida em estruturas internas dos SIG, o que diminuía a abstração e aumentava a dificuldade em se criar bons esquemas conceituais (BORGES, DAVIS e LAENDER, 2005). Mesmo o modelo entidade-relacionamento estendido e os diagramas de classe da UML não foram capazes de suprir todas as necessidades dos projetistas. Entretanto, desde o final dos anos 70, vêm sendo desenvolvidos modelos conceituais para transmissão de propriedades e restrições desses tipos de dados de maneira mais abstrata (ELMASRI e NAVATHE, 2005).

Elmasri e Navathe (2005) destacam ainda a necessidade de uma constante evolução por parte dos bancos de dados geográficos para que estes suportem as funcionalidades de extensibilidade e controle de qualidade de dados. Como as

aplicações de SIG estão em constante evolução, surgem novos tipos de dados e conseqüentemente a necessidade de representá-los conceitualmente e armazená-los em bancos de dados. Para isso é necessário que os modelos conceituais formalizem novos conceitos abstratos e que os SGBD disponibilizem, além de um conjunto de tipos geométricos básicos, a possibilidade de se definir novos tipos e métodos associados. O controle de qualidade de dados é também um tópico importante e já foi discutido anteriormente.

A próxima seção apresenta alguns modelos conceituais para dados geográficos que exercem influência sobre este trabalho.

2.3 Modelos conceituais para bancos de dados geográficos

Modelos conceituais vêm sendo refinados para atender as especificidades encontradas durante a modelagem conceitual de dados em aplicações de SIG. Com isto, permitem a abstração de fenômenos geográficos de acordo com suas dimensões espaciais e temporais, além de possibilitar a representação de associações entre entidades geográficas e não-geográficas, bem como a especificação de restrições de integridade espaciais, temporais e semânticas. Uma discussão completa sobre os detalhes de cada modelo está além do escopo deste trabalho, mas é apresentada uma visão geral sobre os avanços em alguns modelos. Uma discussão mais detalhada sobre requisitos e modelos conceituais é apresentada em Friis-Christensen, Tryfona e Jensen (2001).

Entre os vários modelos existentes específicos para SIG é importante se destacar o equilíbrio entre a abrangência e a facilidade de uso de sua notação (FRIIS-CHRISTENSEN, TRYFONA e JENSEN, 2001). Segundo Kösters, Pagel e Six (1997) e Spaccapietra, Parent e Zimányi (2006), alguns se destacam por serem abrangentes ao abordarem a maioria dos elementos necessários para a modelagem em diversos domínios, mas são complexos demais e de difícil utilização por usuários e desenvolvedores. Outros modelos são mais simples e facilitam a compreensão da realidade modelada, porém nem sempre são capazes de representar situações relevantes e muitas vezes utilizam informações textuais complementares para descrever um domínio específico.

Em geral, duas abordagens para a modelagem de dados geográficos foram adotadas: a abordagem entidade-relacionamento e a orientada a objetos

(WORBOYS, 1995, apud FRIIS-CHRISTENSEN, TRYFONA e JENSEN, 2001, p. 3). A extensão à abordagem entidade-relacionamento apresentada por Tryfona e Jensen (1999), *Spatio-Temporal Entity-Relationship (STER)*, é um importante passo para tornar possível a modelagem conceitual geográfica, sendo realizada através de pictogramas (pequenos símbolos gráficos) para representar conceitos, além de atributos e associações espaciais, temporais e espaço-temporais bem definidos para representar as propriedades de uma entidade e seus relacionamentos. Sem uma extensão como esta, as entidades convencionais e os atributos alfanuméricos seriam utilizados para representar os dados espaciais e temporais, o que resultaria em um modelo conceitual complexo, dificultando a compreensão e a comunicação entre os envolvidos no desenvolvimento da aplicação (SHEKHAR e CHAWLA, 2003). No entanto, o modelo STER, mesmo utilizando pictogramas, não é capaz de representar intuitivamente qual o papel desempenhado pela entidade e não incorpora mecanismos para especificação de restrições de integridade e qualidade dos dados (FRIIS-CHRISTENSEN, TRYFONA e JENSEN, 2001).

Os demais modelos apresentados nesta seção utilizam a abordagem orientada a objetos. Esta abordagem vem sendo amplamente adotada devido ao aumento de SGBD orientados a objetos (SGBD-OO) e objeto-relacionais (SGBD-OR), estimulado pela utilização crescente de linguagens orientadas a objetos como C++ e Java (SHEKHAR e CHAWLA, 2003). A vantagem em se utilizar o conceito de objetos em bancos de dados está na transformação direta entre os modelos utilizados para o banco de dados e para a linguagem de programação. A seguir, estão descritos de forma sucinta os modelos GeoOOA, OMT-G, MADS e UML-GeoFrame.

2.3.1 Modelo GeoOOA

O modelo GeoOOA (KÖSTERS, PAGEL e SIX, 1997) estende o modelo orientado a objetos OOA (*Object-Oriented Analysis*) (COAD e YOURDON, 1991). Possui como características principais, a capacidade de diferenciar entre classes espaciais e convencionais, entre diferentes tipos de objetos espaciais (pontos, linhas e regiões), e suportar relacionamentos topológicos do tipo todo-parte, estruturas de rede e classes temporais.

Uma característica interessante na diferenciação entre classes espaciais e convencionais é o uso de símbolos gráficos para representar as classes do tipo ponto,

linha e região, sendo uma característica utilizada hoje, por exemplo, pelos modelos UML-GeoFrame (LISBOA FILHO e IOCHPE, 2008) e OMT-G (BORGES, DAVIS e LAENDER, 2001). No trabalho descrito em Kösters, Pagel e Six (1997), são apresentados alguns exemplos de modelagem, dentre eles para uma empresa de energia elétrica. Este exemplo faz uso de símbolos para representação dos papéis de rede, nó e aresta, sendo os dois últimos utilizados para especificar o papel que uma classe com representação de objeto espacial exerce na rede. Através do relacionamento entre as classes pode-se especificar o número de arestas incidentes em um nó, o número de arestas e nós que compõem a rede e a quantas redes um nó ou aresta pode pertencer. A Figura 2.3 apresenta uma parte do diagrama proposto para uma companhia de energia elétrica.

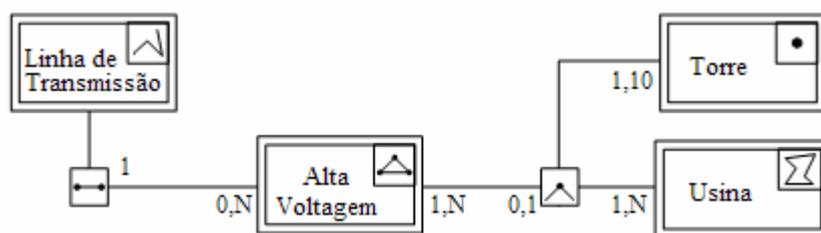


Figura 2.3 – Exemplos de relacionamentos de rede do modelo GeoOOA
Fonte: Kösters, Pagel e Six (1997)

No entanto, o modelo GeoOOA não fornece alguns construtores conceituais importantes, como a especificação de tipos de dados na visão de campos, mecanismos para especificação de restrições de integridade de modo explícito e formal, e a especificação da qualidade dos dados em relação às necessidades dos usuários, requisitos estes avaliados por Friis-Christensen, Tryfona e Jensen (2001).

2.3.2 Modelo OMT-G

O modelo OMT-G (*Object Modeling Technique for Geographic Applications*) (BORGES, DAVIS e LAENDER, 2001), inicialmente baseado apenas no diagrama de classe do modelo OMT (RUMBAUGH et al., 1991), adotou mais tarde também os conceitos e notações da UML. Isso se deve ao OMG (*Object Management Group*) ter considerado a UML uma evolução ao modelo e tê-la adotado como linguagem de modelagem padrão. O modelo OMT-G se caracteriza por incorporar a representação de elementos geográficos na visão de campos e

objetos através do uso de pictogramas, permitir a modelagem geométrica e topológica, associação entre classes espaciais, associação entre classes espaciais e convencionais, relacionamentos do tipo todo-parte, estruturas de rede e múltiplas representações de um mesmo fenômeno geográfico. A Figura 2.4-a apresenta classes de estrutura de rede com relacionamento do tipo arco-nó e nome da rede localizado entre as linhas pontilhadas; a Figura 2.4-b apresenta um relacionamento do tipo arco-arco.

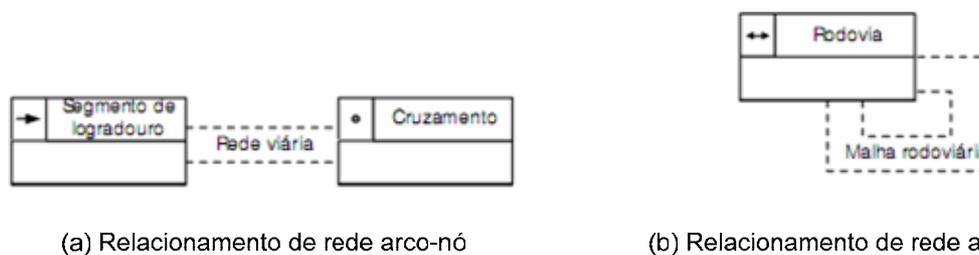


Figura 2.4 – Exemplos de relacionamentos de rede do modelo OMT-G
 Fonte: Borges, Davis e Laender (2005)

O modelo fornece também mecanismos para que sejam representadas as restrições de integridade baseadas na taxonomia apresentada por (COCKCROFT, 1997). Segundo Borges, Davis e Laender (2001), as restrições topológicas são especificadas através de agregação espacial, relacionamentos espaciais, conectividade e regras de integridade na visão de campo; da mesma forma, restrições de integridade semânticas são especificadas através de regras de integridade dos relacionamentos espaciais; e restrições definidas pelo usuário são especificadas através de métodos associados às classes. A Figura 2.5-a exemplifica uma associação simples (linha contínua) na qual *Edificação* pertence a *Proprietário*; a Figura 2.5-b exemplifica um relacionamento espacial (linha pontilhada) no qual *Lote* contém *Edificação*.



Figura 2.5 – Exemplos de relacionamentos simples e espacial do modelo OMT-G
 Fonte: Borges, Davis e Laender (2005)

Além do diagrama de classes, o modelo fornece também os diagramas de transformação e apresentação. O primeiro é responsável pela transformação das classes originadas do diagrama de classe, podendo modificar seus atributos, operações e representação geográfica. O diagrama pode também possuir qualquer número de classes fonte e resultante. O diagrama de apresentação tem como objetivo especificar o aspecto visual das classes identificadas nos dois diagramas anteriores, sem alterar suas representações. Pode ser utilizado, por exemplo, para especificar formas alternativas de apresentação na tela do computador e/ou durante a impressão de mapas ou gráficos.

Segundo análise feita em Friis-Christensen, Tryfona e Jensen (2001), embora nenhum modelo avaliado consiga atender por completo as necessidades da comunidade de SIG, o modelo OMT-G atende de modo parcial grande parte dessas necessidades. Destaca-se a especificação de restrições de integridade de forma explícita, embora através de textos pré-definidos e sem a utilização de uma linguagem formal para que novas restrições possam ser especificadas.

2.3.3 Modelo MADS

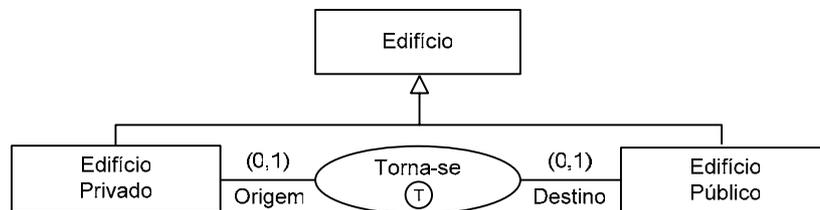
Ao contrário dos dois últimos modelos, o MADS (*Modeling for Application Data with Spatio-temporal features*) (SPACCAPIETRA, PARENT e ZIMÁNYI, 2006) não é uma extensão a um modelo existente para aplicações convencionais. O modelo utiliza a abordagem orientada a objetos, a ortogonalidade para a independência entre as estruturas de dados, espaço e tempo, atributos (mono/multivalorados, simples/complexos, derivados), métodos, restrições de integridade, relacionamentos n-ários e especialização/generalização. Características temporais e espaciais podem estar associadas a objetos, atributos e relacionamentos.

São definidas hierarquias para os tipos abstratos de dados espaciais e temporais, podendo-se também estendê-las segundo os tipos definidos pelo usuário. A primeira fornece construtores espaciais para representar o formato e localização dos fenômenos geográficos através de tipos simples ou complexos, utilizando pontos, linhas e áreas, tornando possível tanto a representação contínua (campo) quanto

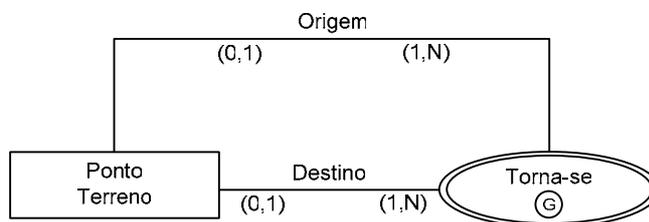
discreta (objeto). A segunda fornece construtores temporais para descrever o ciclo de vida dos fenômenos geográficos, incluindo a representação de instantes e intervalos.

O modelo MADS suporta a especificação de relacionamentos convencionais, de agregação, restrição e dinâmicos. Os relacionamentos de restrição englobam os relacionamentos topológicos, métricos e de direção (ex.: a nordeste de, a leste de), além de adicionar elementos temporais para sincronizar o ciclo de vida dos elementos envolvidos. Relacionamentos dinâmicos descrevem fenômenos nos quais tempo e/ou espaço desempenham um papel significativo. O modelo classifica esse tipo de relacionamento como de transição, para caracterizar a migração de objetos de um tipo para outro, ou de geração, para representar o surgimento de novos objetos.

A Figura 2.6-a apresenta um relacionamento de transição, em que são armazenados dados referentes à aquisição de edifícios privados por organizações públicas. O símbolo \textcircled{T} no relacionamento *Torna-se* o identifica como de transição. A Figura 2.6-b apresenta um relacionamento de geração para o processo de reestruturação de terrenos, no qual um conjunto de pontos de terreno é rearranjado para formar um novo conjunto de pontos de terreno. O símbolo \textcircled{G} no relacionamento de multi-associação *Torna-se* o identifica como de geração.



(a) Relacionamento de transição



(b) Relacionamento de geração

Figura 2.6 – Exemplos de relacionamentos de transição e geração do modelo MADS
 Fonte: Spaccapietra, Parent e Zimányi (2006)

2.3.4 Modelo UML-GeoFrame

O modelo UML-GeoFrame (LISBOA FILHO e IOCHPE, 2008), para o qual são propostos novos construtores neste trabalho, é apresentado em maiores detalhes nesta seção. Esse modelo propõe uma abordagem simples, utilizando o paradigma de orientação a objetos. Conforme pode ser visto na Figura 2.7, as classes que compõem o modelo são organizadas hierarquicamente no *framework* GeoFrame (LISBOA FILHO e IOCHPE, 1999), fornecendo os elementos fundamentais para a modelagem de bancos de dados geográficos.

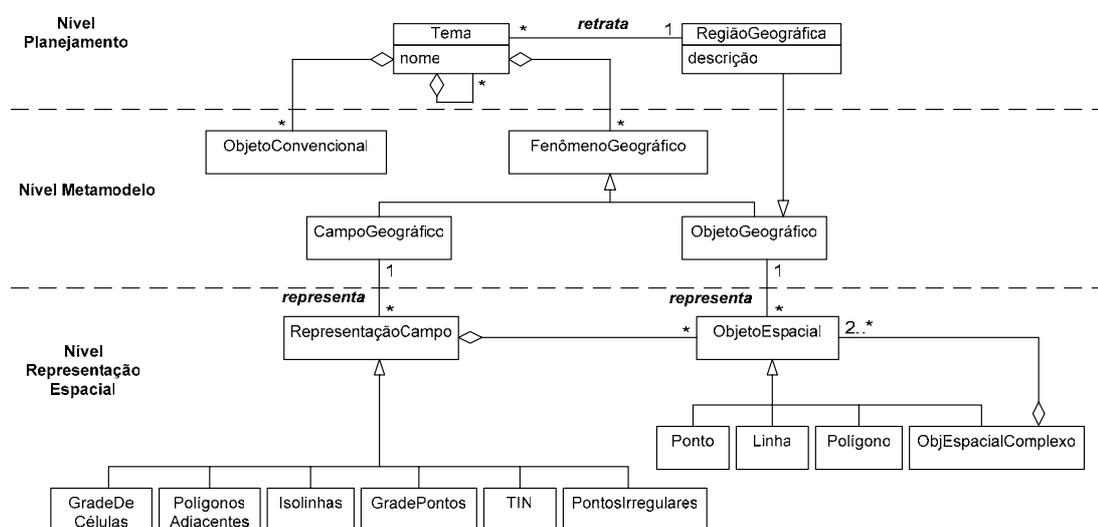


Figura 2.7 – Framework GeoFrame
Fonte: Lisboa Filho e Iochpe (2008)

O nível de planejamento é utilizado para a identificação das regiões geográficas correspondentes às áreas de interesse (classe *RegiãoGeográfica*) e seus temas associados (classe *Tema*), podendo-se subdividir os temas em uma hierarquia de sub-temas. Neste nível, não se tem a preocupação de como a realidade é interpretada, muito menos como deve ser representada.

O nível de metamodelo é composto por classes que refletem o modo como a realidade é interpretada, sendo objetos convencionais (classe *ObjetoConvencional*) ou georeferenciados (classe *FenômenoGeográfico*). Esta última classe é ainda especializada nas classes para a abordagem espacial contínua (classe *CampoGeográfico*) e discreta (classe *ObjetoGeográfico*).

O nível de representação espacial tem como objetivo refletir o modo como a realidade é representada por projetistas e usuários em um nível mais abstrato em

relação à sua representação dentro do banco de dados. A classe *ObjetoEspacial* generaliza as classes de representação espacial observadas na visão de objeto, sendo estas *Ponto*, *Linha*, *Polígono* e *ObjEspacialComplexo*. Esta última representa um objeto espacial composto de dois ou mais objetos espaciais. A classe *RepresentaçãoCampo* generaliza as classes da visão de campo, sendo estas *GradeDeCélulas*, *PolígonosAdjacentes*, *Isolinhas*, *GradePontos*, *TIN (Triangular Irregular Network)* e *PontosIrregulares*.

Um fenômeno geográfico do tipo *CampoGeográfico* pode ser representado por diversas classes de *RepresentaçãoCampo*, da mesma forma como um fenômeno geográfico do tipo *ObjetoGeográfico* pode ser representado por diversas classes de *ObjetoEspacial*. Esta propriedade do GeoFrame é conhecida como múltipla representação. Para exemplificar esta situação, supondo que a interpretação atribuída a uma cidade seja através da visão de objetos e que sua representação em uma determinada escala seja a de um polígono, ao se alterar a escala para uma muito menor, a representação mais adequada seria a de ponto. Outra propriedade importante do GeoFrame, devido à constante evolução da área, é a capacidade de se adicionar novas classes em qualquer dos três níveis para atender a novos requisitos.

O método para modelagem de bancos de dados geográficos através do UML-GeoFrame apresentado em Lisboa Filho e Iochpe (2008) é composto de cinco passos, através dos quais é possível se identificar o modo como os elementos do GeoFrame são integrados aos construtores do diagrama de classes da UML.

O primeiro passo concentra-se no nível de planejamento, sendo identificada(s) inicialmente a(s) região(ões) geográfica(s) considerada(s) na aplicação de SIG. São identificados também os temas relacionados à(s) região(ões) geográfica(s), os quais são modelados utilizando os construtores de pacotes da UML e permitindo uma abordagem *top-down* do problema. Através desta abordagem, é possível se dividir o problema em problemas menores, facilitando seu entendimento tanto por projetistas quanto por usuários.

O segundo passo concentra-se no nível de metamodelo, sendo os elementos essenciais de cada pacote representados segundo o diagrama de classes da UML e classificados em convencional e fenômeno geográfico nas visões de campo e de objetos. As associações entre as classes, até mesmo de diferentes pacotes, também são modeladas durante esta etapa sem se considerar as restrições espaciais, discutidas no quarto passo.

Outra característica importante no segundo passo é a utilização de estereótipos, mecanismos de extensão da UML para permitir a modelagem conceitual em domínios específicos. Sem este recurso, todas as classes da aplicação deveriam ser subclasses de uma das três classes do nível de metamodelo do GeoFrame, sobrecarregando e dificultando a leitura do esquema. A Figura 2.8 apresenta exemplos de classes com os estereótipos das metaclasses (a) *ObjetoConvencional*, (b) *ObjetoGeográfico* e (c) *CampoGeográfico*.

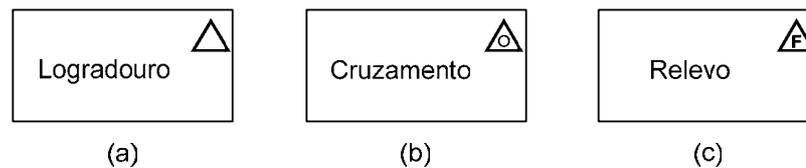


Figura 2.8 – Estereótipos de generalização do modelo UML-GeoFrame

O terceiro passo concentra-se no nível de representação espacial e, semelhante ao uso de estereótipos do segundo passo, é proposto o uso de estereótipos para os diferentes modos de se abstrair e representar a forma espacial dos fenômenos geográficos. Porém, é importante entender a diferença entre a definição dos estereótipos no segundo e terceiro passos. Se o mesmo raciocínio do segundo passo fosse utilizado, a classe *Rua*, por exemplo, seria modelada como subclasse de *Linha* e a classe *Poste* seria subclasse de *Ponto*. No entanto, existe uma incoerência neste raciocínio, pois o conceito de generalização/especialização não recomenda que coisas distintas se relacionem desta forma. Assim, embora uma linha possa ser utilizada para representar o aspecto geométrico de uma rua, isso não implica que uma rua seja uma linha.

Com base nas associações entre as classes *CampoGeográfico* e *RepresentaçãoCampo*, e *ObjetoGeográfico* e *ObjetoEspacial*, ambas identificadas como **representa** na Figura 2.7, é possível modelar a característica espacial de um fenômeno geográfico. Sobre essas associações foram definidos os estereótipos utilizados para a representação espacial, sendo a motivação a mesma apresentada anteriormente, não sobrecarregar e facilitar a leitura do esquema. Os estereótipos referentes a esta etapa podem ser observados na Figura 2.9. Dessa forma, são utilizados dois estereótipos em cada classe, um para refletir o tipo de metaclassa empregado e outro para refletir o tipo de representação espacial.

ObjetoEspacial		RepresentaçãoCampo	
 Ponto	 Polígono	 Isolinhas	 GradeDeCélulas
 Linha	 ObjEspacialComplexo	 TIN	 PontosIrregulares
		 GradePontos	 PolígonosAdjacentes

Figura 2.9 – Estereótipos de representação espacial do modelo UML-GeoFrame
Fonte: Lisboa Filho e Iochpe (2008)

O quarto passo aborda os relacionamentos entre as classes identificadas, classificando-os em semânticos ou espaciais. Um SIG implementa os relacionamentos semânticos da mesma maneira como é feito em aplicações convencionais, no qual uma entidade possui um campo que referencia outra entidade. No modelo relacional, este campo recebe o nome de chave estrangeira. Por outro lado, os relacionamentos espaciais são implementados através de procedimentos no SIG ou SGBD Geográfico (SGBDG) de acordo com o tipo de relacionamento que se deseja estabelecer, garantindo assim o cumprimento da restrição de integridade espacial quando os dados são inseridos ou modificados. No UML-GeoFrame, os tipos de relacionamentos espaciais pré-definidos correspondem a um subconjunto da combinação de relacionamentos possíveis entre as geometrias de ponto, linha e polígono, e foram propostos em Egenhofer e Franzosa (1991).

No modelo UML-GeoFrame, os relacionamentos semânticos são representados da mesma maneira como em uma modelagem de aplicação convencional, através de associação entre as classes contendo a multiplicidade em ambos os lados. Os relacionamentos espaciais, por sua vez, incluem estereótipos textuais com o texto entre “<<...>>”. Esse texto se refere ao tipo de restrição que se deseja impor. A Figura 2.10 apresenta exemplos de como as associações podem ser representadas no modelo UML-GeoFrame: (a) associação semântica; (b) associação espacial com restrição de integridade do tipo *inside* (dentro); (c) agregação espacial. É importante observar que o modelo torna possível o relacionamento entre classes convencionais, classes espaciais e classes convencionais e espaciais. É importante observar também a possibilidade de se representar a classe *Escola* como ponto ou polígono (múltipla representação) e a capacidade de se realizar relacionamentos espaciais do tipo todo-parte através do uso dos construtores de agregação e composição da UML.

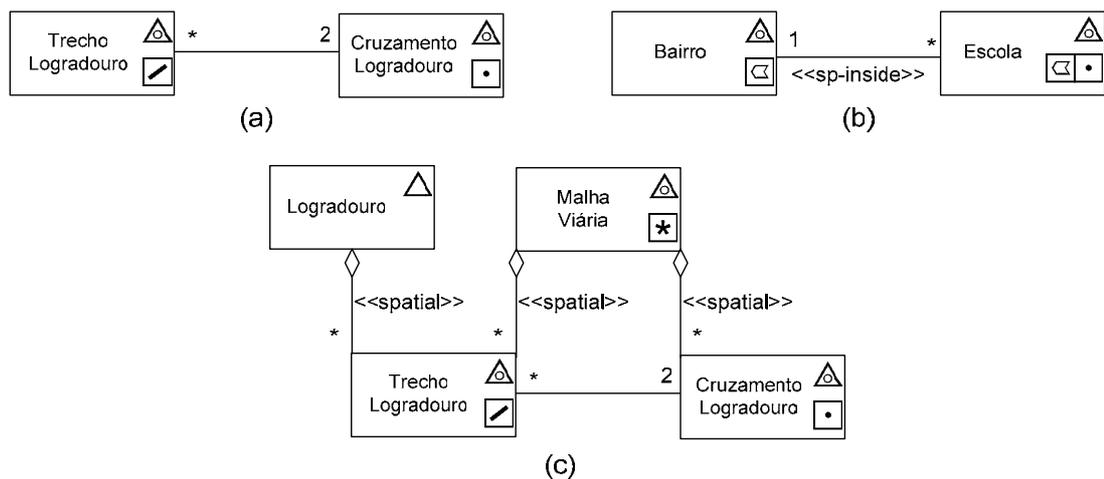


Figura 2.10 – Estereótipos de relacionamento espacial do UML-GeoFrame

No quinto e último passo, são representados os aspectos temporais relacionados às classes convencionais e espaciais. O conceito de temporalidade, já existente em aplicações convencionais, tem como objetivo expressar as mudanças que ocorrem ao longo do tempo nos dados armazenados no banco de dados.

Os dados descritivos, espaciais e temporais são ortogonais, o que torna possível a especificação de características temporais para objetos convencionais e objetos geográficos contínuos e discretos. Embora um SGBDG suporte a especificação de aspectos temporais considerando o tempo válido e o tempo de transação, o modelo UML-GeoFrame fornece construtores apenas para o primeiro caso. Segundo Lisboa Filho e Iochpe (2008), o tempo de validade é um instante ou intervalo de tempo em que um objeto do mundo real é considerado válido. O uso de intervalo de tempo implica no armazenamento das mudanças que foram realizadas tanto nas propriedades espaciais quanto descritivas de um objeto, através da criação de uma nova versão com atributos temporais que indicam o tempo válido inicial e final. O uso de instante de tempo considera classes com atributos temporais sem a necessidade de se criar novas versões dos objetos.

A Figura 2.11 apresenta exemplos de fenômenos espaço-temporais que podem ocorrer ao longo dos trechos de um logradouro. Assim como na Figura 2.10, a classe *Logradouro*, do tipo convencional, é uma agregação de instâncias da classe *TrechoLogradouro*, do tipo objeto geográfico com representação espacial de linha. Em cada trecho de logradouro, podem estar associados dois tipos de fenômenos com características temporais, representados pelas classes *Acidente* e *Obra*. A classe *Acidente* possui representação espacial pontual com característica temporal

representada por instante de tempo, resultando em uma nova instância contendo o momento de cada ocorrência. A classe *Obra*, por outro lado, possui atributos descritivos com característica temporal representada por intervalo de tempo, resultando em uma nova versão do objeto acrescentada de atributos temporais para representar o intervalo de tempo válido de cada ocorrência.

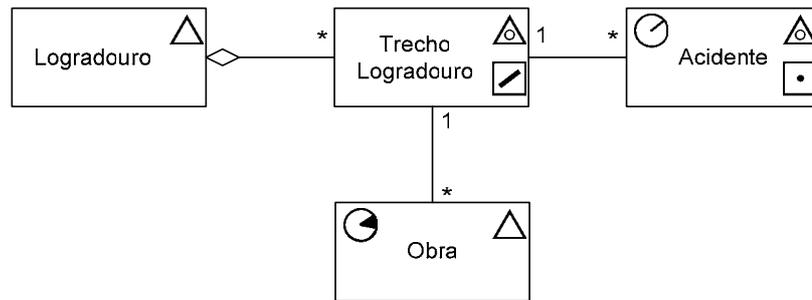


Figura 2.11 – Exemplos de uso dos estereótipos espaço-temporais

2.4 Taxonomia de restrições de integridade espaciais

Muitos pesquisadores utilizam uma classificação dos diferentes tipos de restrições de integridade para que seja possível dividir todo o conjunto em partes menores com o objetivo de entender melhor a área e direcionar seus estudos. Uma classificação desenvolvida por Cockcroft (1997) e amplamente aceita pela comunidade de SIG é apresentada na Figura 2.12.

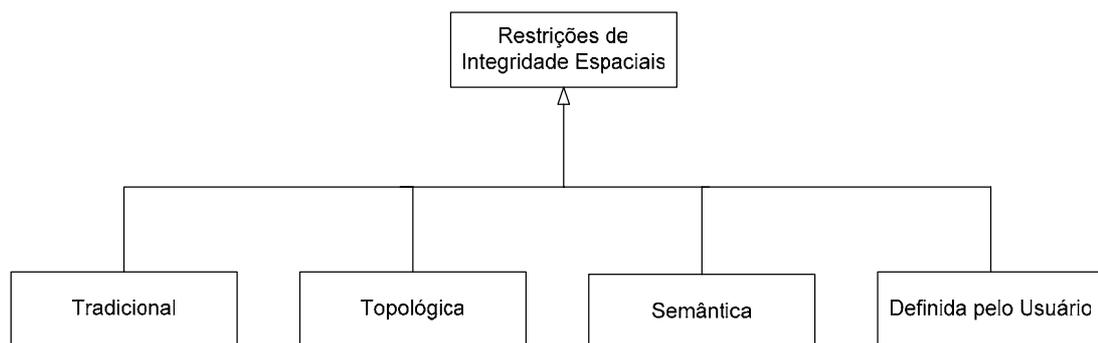


Figura 2.12 – Classificação hierárquica de restrições de integridade espaciais

Fonte: Cockcroft (1997)

As restrições classificadas como tradicionais são as restrições dominantes e capazes de serem especificadas em SGBD comerciais. São classificadas segundo Elmasri e Navathe (2005) em:

- **Restrição de domínio:** Restrição que impõe valores válidos para os atributos de acordo com determinado tipo. Exemplos são: Inteiro, Real, String, Data, Booleano etc.;
- **Restrição de integridade de entidade:** Cada instância de um tipo de entidade deve possuir um identificador único sem valor nulo;
- **Restrição de estrutura do atributo:** Especifica se um atributo possui um único valor ou é multivalorado e se o valor nulo é permitido;
- **Restrições de integridade referencial:** Um banco de dados não deve possuir nenhuma chave estrangeira que não tenha o seu valor equivalente na chave primária da entidade referenciada;
- **Restrições de superclasse/subclasse:** Impõe regras de disjunção/sobreposição e totalidade/parcialidade através da especialização/generalização.

As restrições de integridade que de fato se relacionam com aspectos espaciais são classificadas por Cockcroft (1997) em topológicas, semânticas e definidas pelo usuário.

“Topologia é o procedimento matemático utilizado para definir relações espaciais entre pontos, linhas e polígonos” (COCKCROFT, 1997). Egenhofer e Franzosa (1991) formalizaram nove possíveis relacionamentos que podem existir entre estes elementos geométricos. Clementini, Di Felice e Oosterom (1993) por outro lado identificaram um conjunto mínimo de cinco relacionamentos: disjunção, toca, dentro, cruza e sobreposição. No entanto, segundo Davis, Borges e Laender (2002), muitas vezes um conjunto maior de relacionamentos derivados desses cinco relacionamentos básicos é necessário devido a conceitos culturais ou semânticos familiares aos usuários. Os novos relacionamentos propostos foram: adjacência, coincide, contém e proximidade. O conceito de restrição de integridade espacial topológica é, portanto, a garantia de que os relacionamentos espaciais identificados não serão violados.

“Restrições semânticas diferem das topológicas devido ao interesse no significado das características geográficas” (COCKCROFT, 1997). Embora seja possível se dividir a área de um município em, por exemplo, áreas residenciais, comerciais, industriais e de reserva ambiental, e se especificar que edificações devem ser construídas dentro destes limites, alguns casos devem ser impedidos não simplesmente por um relacionamento topológico, mas pelo significado das entidades envolvidas. Um exemplo de violação de restrição de integridade espacial semântica é a tentativa de se demarcar uma edificação em uma área de reserva ambiental. É importante destacar, no entanto, que nem os modelos conceituais nem os SGBDG possuem construtores que possibilitam a especificação e implementação, respectivamente, desse tipo de restrição.

“As restrições definidas pelo usuário permitem a um SGBDG manter a consistência de forma semelhante às regras de negócio estabelecidas em SGBD convencionais” (COCKCROFT, 1997). Um exemplo desse tipo de restrição é a especificação, por parte do usuário, da distância mínima a ser respeitada entre um depósito de resíduos urbanos e áreas residenciais.

Para Cockcroft (1997), como nem todas as restrições de integridade são do tipo estático, satisfeito em cada estado do banco de dados, utilizou-se então um segundo tipo de restrição, restrição de transição, no qual são restringidas as possíveis transições entre estados do banco de dados. A taxonomia completa de Cockcroft (1997) é composta então por dois eixos. Um eixo para classificá-las em estáticas ou de transição, e outro para classificá-las em topológicas, semânticas ou definidas pelo usuário.

A partir dessa taxonomia surgiram então algumas variações, como a descrita por Lin et al. (2005) e Louwsma et al. (2006). A Figura 2.13 ilustra a taxonomia apresentada por Lin et al. (2005).

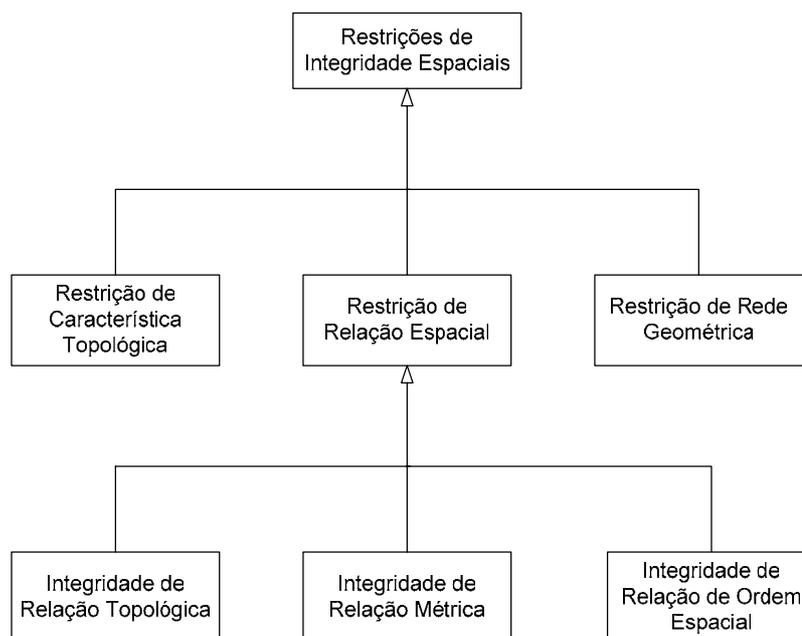


Figura 2.13 – Taxonomia de restrições de integridade espaciais
 Fonte: Lin et al. (2005)

“As restrições de característica topológica representam as propriedades topológicas de uma única geometria e devem ser satisfeitas dentro de bases de dados espaciais. Exemplos são: uma linha não deve se auto-interceptar ou um polígono deve ser fechado” (LIN et al., 2005).

“A restrição de relação espacial trata de restrições em termos da relação espacial entre duas ou mais geometrias que estão conectadas semanticamente. Por exemplo, rotas de ônibus devem estar sobre ruas” (LIN et al., 2005). Restrições de integridade de relação espacial são subdivididas em: relação topológica, relação métrica e relação de ordem espacial, de acordo com a classificação de relacionamentos espaciais proposta por Egenhofer e Franzosa (1991).

A definição de relação topológica é a mesma apresentada por Cockcroft (1997). A restrição métrica pode ser compreendida através do exemplo da distância mínima entre um depósito de resíduos urbanos e áreas residenciais apresentado durante a definição de restrição definida pelo usuário. A restrição de relação de ordem espacial define relacionamentos de proximidade, podendo-se expressar os relacionamentos com base em uma linguagem informal, como ao norte de, ao lado de, perto de, em frente a etc.

A restrição de rede geométrica se relaciona com a teoria de grafos. Um grafo é uma estrutura composta por nós (vértices) e arcos (arestas), no qual cada arco deve possuir um nó inicial e final, não necessariamente diferentes. A teoria de grafos é o estudo dos tipos de redes que podem ser formadas através dos relacionamentos entre nós e arcos e os métodos que podem ser aplicados, constituindo-se em uma importante área tanto da matemática quanto da ciência da computação. Apesar de serem representados por pontos e linhas, a informação contida, respectivamente, nos nós e arcos difere da informação geométrica. Segundo Laurini e Thompson (1992), devem ser descartadas as informações sobre o formato da linha, orientação e comprimento, concentrando-se apenas nos componentes estruturais, as junções e conexões. A informação principal é a conectividade, mas o sentido, que estabelece o ponto inicial e final, também é importante. Para Lin et al. (2005), a restrição de rede geométrica é utilizada para descrever as restrições de conectividade e fluxos em um modelo de redes.

Pode-se perceber que, apesar de certo grau de correspondência entre a classificação apresentada por Cockcroft (1997) e Lin et al. (2005), a primeira não aborda algumas restrições quando comparadas à segunda, como a restrição de característica topológica e do modelo de redes. Por outro lado, através da restrição de integridade espacial definida pelo usuário, Cockcroft (1997) engloba, mas não se restringe à restrição de integridade de relação métrica estabelecida por Lin et al. (2005). Tem-se também uma semelhança entre a restrição semântica de Cockcroft (1997) e a restrição de relação espacial de Lin et al. (2005), pois embora no último seja realizada a subdivisão desse tipo de restrição, os autores a definem como a relação entre duas ou mais geometrias conectadas semanticamente.

2.5 Object constraint language

A utilização de diagramas para modelagem de sistemas é uma técnica amplamente difundida entre projetistas, devido à sua simplicidade e capacidade de transmitir as características e comportamento do sistema. No entanto, existem alguns aspectos que não podem ser descritos através do uso de diagramas, como unicidade, derivação e limites dos valores de um atributo e restrições durante a entrada ou modificação de dados. Para suprir essas necessidades, podem ser utilizadas linguagens formais para uma descrição precisa e sem ambigüidades.

Entre essas linguagens está a linguagem de restrição de objetos (*Object Constraint Language - OCL*) (WARMER e KLEPPE, 2003; OMG, 2006), utilizada para descrever expressões nos modelos da UML. Para Anders Ivner (WARMER e KLEPPE, 2003), a OCL possui três características que a tornam uma linguagem de sucesso: ela é sucinta, o que implica em poucos elementos simples de serem compreendidos; é compacta, mas ainda assim poderosa, tornando possível serem escritas expressões curtas, precisas e capazes de expressar diversas ações; e assemelha-se ao uso de linguagens de programação orientadas a objetos, sendo familiar àqueles com alguma experiência em desenvolvimento de *software*.

Segundo Steve Cook (WARMER e KLEPPE, 2003), a OCL foi desenvolvida inicialmente em 1995 durante um projeto de modelagem de negócios dentro da IBM, onde estiveram Jos Warmer e Steve Cook, envolvidos também em 1996 com os esforços do OMG para padronização de uma linguagem para análise e projeto orientada a objetos. A principal proposta na época era a UML, cujo foco eram elementos diagramáticos com significados expressos por textos. A contribuição do trabalho realizado na IBM e a proposta de uma linguagem simples e precisa levou à combinação dessas propostas, resultando na utilização da OCL para que fossem adicionados aspectos que não podiam ser expressos pelos construtores conceituais da UML, tornando a OCL parte do padrão proposto pela OMG. Atualmente, a OCL se encontra na versão 2.0.

Apesar de seu nome refletir apenas a especificação de restrições, a OCL vai além. Através de seu uso é possível, por exemplo, se especificar consultas, definir regras de derivação, valores iniciais, novos atributos e operações. Além disso, como a OCL atua de modo a suprir as necessidades da UML, cada expressão escrita em OCL depende dos tipos definidos nos diagramas da UML.

Em resumo, a OCL possui quatro características principais: ser uma linguagem tanto para especificação de consultas quanto de restrições; possuir como base a teoria de conjuntos da matemática e a lógica de predicados, mas sem a utilização de símbolos que possam dificultar seu aprendizado; ser fortemente tipada, sendo a validade de suas expressões verificada durante a modelagem, antes da execução, resultando na remoção de erros em um estágio inicial do desenvolvimento da aplicação; e por último, ao contrário de linguagens procedurais, a OCL se caracteriza por ser uma linguagem declarativa, especificando o que deve ser feito, ao invés de como deve ser feito.

Considerando-se os benefícios da OCL e a proposta de extensão do modelo UML-GeoFrame, o objetivo desta seção é apresentar um embasamento teórico sobre alguns construtores que compõem esta linguagem, especificamente aqueles utilizados no diagrama de classes. Como a OCL não possui construtores para expressões envolvendo elementos geográficos, será utilizado como exemplo um esquema conceitual baseado no clássico esquema de banco de dados de *companhia* proposto por Elmasri e Navathe (2005), representado na Figura 2.14. Uma extensão da OCL para especificação de relacionamentos topológicos está descrita na seção 2.6.

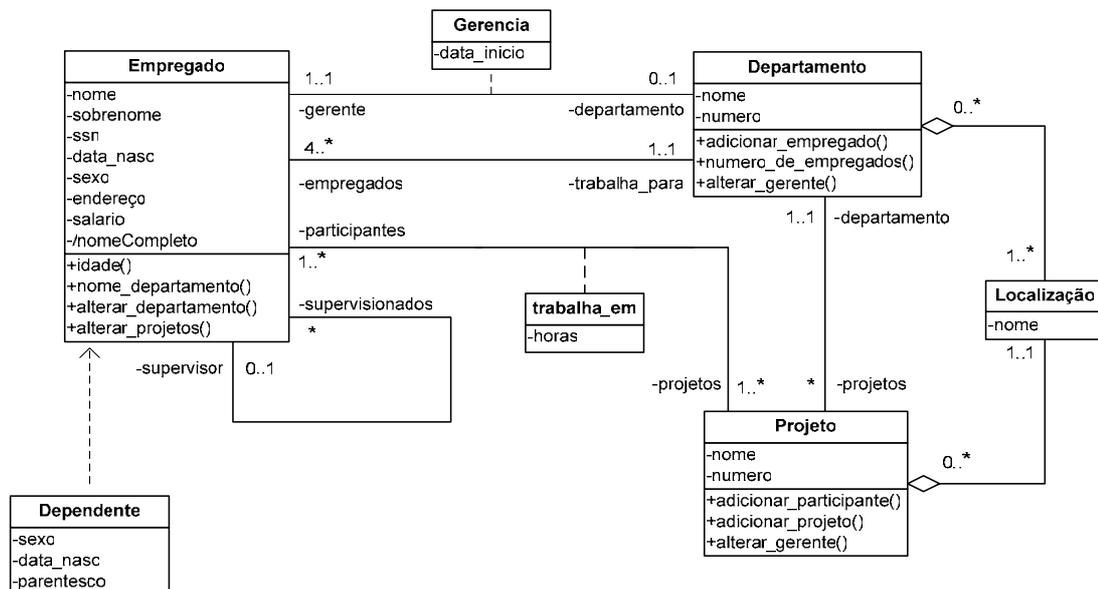


Figura 2.14 – Esquema conceitual UML para um banco de dados de companhia
 Fonte: Elmasri e Navathe (2005)

Apesar de abordar grande parte das informações pertinentes ao sistema, o esquema conceitual na Figura 2.14 não é capaz de expressar todas as informações necessárias. Dessa forma, utilizando os conceitos apresentados em Warmer e Kleppe (2003) e OMG (2006), cada uma das seções a seguir irá ilustrar o funcionamento da OCL e complementar a especificação da aplicação utilizada como exemplo.

2.5.1 Contexto

A definição de contexto é o passo inicial na formulação de qualquer expressão da OCL, pois se trata da ligação entre uma entidade do esquema da UML e a expressão OCL. A entidade pode ser uma classe, interface, tipo de dados definido pelo usuário, componente ou até mesmo uma operação ou instância.

As expressões podem ser definidas diretamente dentro do esquema conceitual ou em um arquivo texto separado. Quando especificadas em arquivo, a entidade contexto deve seguir a palavra-chave *context*, conforme o exemplo a seguir:

```
context Empregado  
inv: nome = "Sergio"
```

Quando dentro do esquema conceitual, o contexto deve ser definido conforme a Figura 2.15.

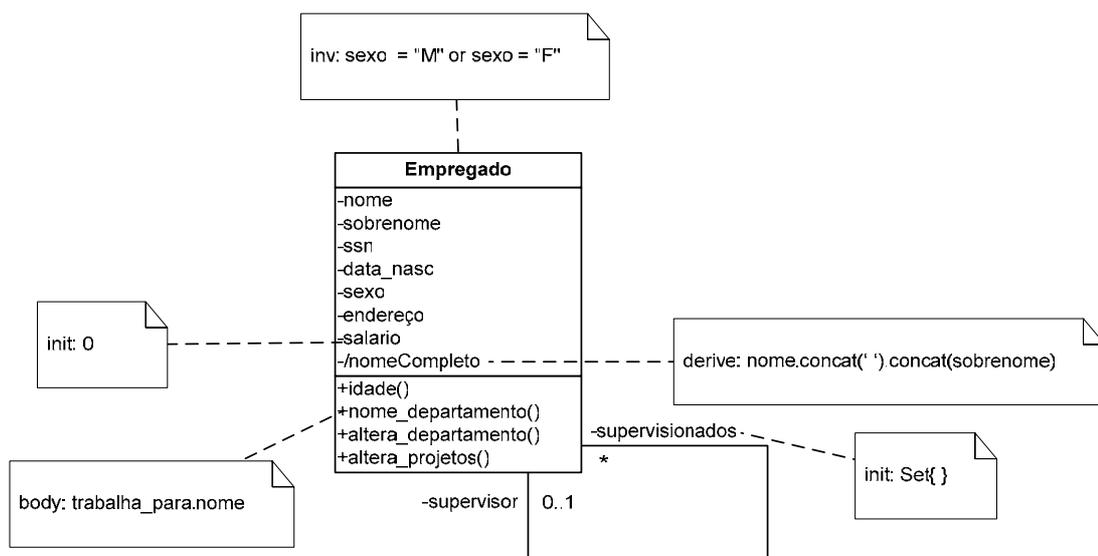


Figura 2.15 – Contexto de expressões OCL no diagrama de classes da UML

É importante se destacar que a definição do contexto também determina as funções que uma expressão OCL pode representar. Por exemplo, se o contexto for um atributo, a expressão pode representar um valor inicial ou uma regra de derivação, mas quando o contexto for uma classe, o mesmo não pode ser feito.

2.5.2 Instância de contexto

As expressões da OCL são avaliadas para cada instância da entidade contexto, assim como para as instâncias do conjunto definido pelas associações ou cadeias de associações iniciados no contexto.

Algumas vezes é necessário fazer referência explícita à instância de contexto. Quando necessário basta ser utilizada a palavra-chave *self*. Seu uso, no entanto, é opcional quando a referência à instância for muito óbvia. O exemplo a seguir ilustra sua utilização:

```
context Departamento  
inv: self.empregados -> size() > 3
```

2.5.3 Valores iniciais e regras de derivação

A OCL possui construtores específicos para que os valores iniciais de atributos e associações e as regras de derivação possam ser inclusos na modelagem da aplicação. O valor inicial de um atributo ou associação é estabelecido indicando-se o contexto e a palavra-chave *init*:

```
context Empregado::salario  
init: 0  
  
context Departamento::projetos: Set(Projeto)  
init: Set {}
```

As regras de derivação, similares às regras de valores iniciais, determinam o valor derivado para atributos ou associações. O valor derivado é estabelecido indicando-se o contexto e a palavra-chave *derive*, como no exemplo a seguir para o valor do atributo *nomeCompleto* da classe *Empregado*, composto pelo nome e sobrenome do empregado:

```
context Empregado::NomeCompleto  
derive: nome.concat(' ').concat(sobrenome)
```

2.5.4 Operações de consulta

Uma característica importante da OCL em relação às operações de consulta é a não alteração do estado atual do sistema, sendo definido apenas o valor ou conjunto de valores a serem retornados pela operação. O nome da operação, seus parâmetros e seu tipo de retorno representam o contexto da expressão OCL e a expressão após a palavra-chave *body* representa o que deve ser retornado pela operação. No exemplo a seguir, são retornados todos os empregados com idade superior ao valor atual do parâmetro da operação, participantes do projeto onde houve a chamada da operação *getParticipantesIdade*:

context

```
Projeto::getParticipantesIdade(pIdade:Number):Set(Empregado)  
body: participantes -> select(idade > pIdade) -> asSet()
```

2.5.5 Invariáveis

Invariáveis são restrições que devem ser verdadeiras durante todo o ciclo de vida de uma instância. Podem ser especificadas em atributos, classes e associações, com expressões que seguem a palavra-chave *inv*.

No modelo utilizado, pode-se estabelecer que o sexo de um empregado deva ser masculino ou feminino através da seguinte expressão OCL intitulada *avaliaSexo*:

context Empregado

inv: avaliaSexo: sexo = "M" or sexo = "F"

Invariáveis podem definir também regras para objetos associados. O exemplo a seguir define que todos os empregados de um departamento devem possuir sua maioridade civil:

context Departamento

inv: maioridade: empregados.idade >= 21

Esse tipo de invariável é definido através do *papel* utilizado na associação para se referir ao objeto ou coleção de objetos associados. Caso o *papel* não seja especificado, o nome da classe deve ser utilizado.

2.5.6 Coleções de objetos

A OCL possui ainda operações para que seja possível se trabalhar com associações de multiplicidade superior a 1. O uso dessas operações segue o operador flecha (\rightarrow). Operações definidas pelo usuário dentro das classes do diagrama seguem o operador ponto (\cdot).

Na OCL são definidos cinco tipos de coleções de objetos: *Set*, *OrderedSet*, *Bag*, *Sequence* e *Collection*, sendo o último um tipo abstrato para os quatro primeiros. O tipo *Set* contém instâncias de tipos válidos na OCL desde que estes não estejam duplicados no conjunto. *OrderedSet* possui as mesmas características de *Set*, porém com os elementos ordenados. *Bag* se diferencia de *Set* por permitir elementos duplicados e *Sequence* possui as características de *Bag*, porém com os elementos ordenados. É importante ressaltar que os elementos de *OrderedSet* e *Sequence* são ordenados segundo um número de seqüência e não pelos valores dos elementos.

O exemplo a seguir ilustra a sintaxe das operações definidas pela OCL sobre coleções de objetos:

```
context Departamento  
inv: empregados  $\rightarrow$  size()  $\geq$  4
```

Neste exemplo, é definida uma restrição na qual cada departamento deve possuir no mínimo quatro empregados.

Algumas operações da OCL para se manipular coleções e relevantes a este trabalho são destacadas a seguir. Uma característica importante dessas operações é a impossibilidade de se alterar o valor de uma coleção. Quando há necessidade de alteração, o resultado da operação será uma nova coleção.

- `count(objeto)`: Retorna o número de ocorrências do objeto na coleção;
- `excludes(objeto)`: Retorna verdadeiro se o objeto não é um elemento da coleção;
- `excludesAll(coleção)`: Retorna verdadeiro se todos os elementos da coleção parâmetro não estão presentes na coleção à qual se aplica a chamada de operação;
- `excluding(objeto)`: Resulta em uma nova coleção com o objeto removido da coleção original;

- `includes(objeto)`: Retorna verdadeiro se o objeto é um elemento da coleção;
- `includesAll(coleção)`: Retorna verdadeiro se todos os elementos da coleção parâmetro estão presentes na coleção à qual se aplica a chamada de operação;
- `including(objeto)`: Resulta em uma nova coleção com o objeto adicionado à coleção original;
- `isEmpty()`: Retorna verdadeiro se a coleção não contém nenhum elemento;
- `notEmpty()`: Retorna verdadeiro se a coleção contém algum elemento;
- `size()`: Retorna o número de elementos na coleção;
- `sum()`: Soma todos os valores dos elementos da coleção, sendo necessário que estes sejam de tipos que suportem a operação de adição.

2.5.7 Operações de iteração

Através da OCL, é possível também se construir expressões de iteração sobre os elementos de uma coleção. As operações definidas na OCL permitem que, a cada iteração, uma expressão passada como parâmetro seja avaliada sobre o elemento. Esse tipo de operação permite também a declaração de variáveis de iteração para se evitar ambigüidades.

O exemplo a seguir ilustra a sintaxe das operações de iteração definidas pela OCL sobre coleções de objetos:

```
context Projeto
inv:participantes -> exists( part | part.salario > 3000)
```

Neste exemplo, é definida uma restrição na qual, para cada projeto, deve existir ao menos um participante com salário superior a 3000.

As variáveis de iteração, exemplificadas acima pela variável *part*, são utilizadas quando o tipo dos elementos na coleção possui atributos com os mesmos nomes do tipo da instância de contexto. No exemplo acima, se por qualquer motivo existisse um atributo salário na classe *Projeto* e não fosse utilizada a variável de iteração, seria difícil compreender a qual salário a expressão OCL estaria se referindo. Como esta situação é possível devido à declaração da variável de iteração ser opcional, deve-se seguir a especificação da OCL, que define o escopo do tipo dos

elementos de forma aninhada. Dessa forma, a busca pelo tipo se inicia pela coleção mais interna, o que no exemplo resultaria em uma referência ao salário da classe *Empregado*.

Embora a declaração da variável de iteração seja opcional, existem casos em que sua declaração se torna inevitável, como no exemplo a seguir:

```
context Empregados
inv: projetos.participantes -> select( part:Empregado |
                                         part <> self )
```

O exemplo resulta em uma coleção de todos os empregados participantes dos mesmos projetos que a instância de contexto. Este exemplo também descreve a declaração opcional do tipo da variável de iteração, no caso, do tipo *Empregado*. *Self* representa a instância de contexto e *<>* é um operador de diferença.

Ainda resta uma observação importante a ser feita sobre as operações de iteração. Sempre que uma restrição é especificada para uma classe, isso implica que a condição se mantém para todas as suas instâncias. Operações de iteração, por outro lado, devem ser utilizadas quando se deseja definir restrições sobre um subconjunto de todas as instâncias de uma classe. No exemplo anterior, o subconjunto é dado pelos participantes dos mesmos projetos que a instância de contexto.

Algumas operações da OCL para se manipular coleções através da iteração são apresentadas a seguir:

- `any(expr)`: Retorna um elemento randômico da coleção para o qual a expressão passada como parâmetro seja verdadeira;
- `exists(expr)`: Retorna verdadeiro se existe ao menos um elemento na coleção para o qual a expressão seja verdadeira;
- `forall(expr)`: Retorna verdadeiro se a expressão é verdadeira para todos os elementos da coleção;
- `isUnique(expr)`: Retorna verdadeiro se a expressão possui um valor único para todos os elementos da coleção;
- `one(expr)`: Retorna verdadeiro se existe exatamente um elemento na coleção para o qual a expressão seja verdadeira;
- `reject(expr)`: Retorna um subconjunto contendo todos os elementos para os quais a expressão seja falsa;

- `select(expr)`: Retorna um subconjunto contendo todos os elementos para os quais a expressão seja verdadeira;
- `sortedBy(expr)`: Retorna uma coleção contendo todos os elementos da coleção original ordenados segundo a expressão passada como parâmetro.

2.5.8 Pré-condições e pós-condições

Aplicadas em operações definidas pelo usuário no esquema conceitual, as pré-condições e pós-condições são utilizadas para se definir as interfaces no sistema, uma vez que estas não descrevem como uma operação deve ser implementada. Uma pré-condição é uma expressão booleana que deve ser verdadeira no momento em que a operação inicia sua execução. Uma pós-condição é uma expressão booleana que deve ser verdadeira no momento em que a operação termina sua execução. A utilização de pré-condição não implica na utilização de pós-condição e vice-versa.

O exemplo a seguir ilustra a sintaxe de uso de pré-condições e pós-condições:

```
context Projeto::adicionar_participantes( p: Empregado )
pre: self.departamento.empregados -> includes( p )
pos: participantes = participantes@pre -> including( p )
```

No exemplo, a pré-condição restringe a execução da operação caso o empregado a ser adicionado ao projeto não trabalhe no departamento ao qual pertence o projeto. A pós-condição certifica-se de que após a execução da operação, o novo participante esteja incluso na coleção de participantes do projeto.

Em uma pós-condição, a expressão OCL pode se referir aos valores de um atributo, associação ou operação em dois momentos distintos: quando no início da operação e ao se concluir a operação. O uso convencional se refere ao valor no final da operação. Para fazer referência ao valor no início da operação, o nome da propriedade deve vir sucedido da palavra-chave *@pre*, como no exemplo anterior, no qual foi utilizado o valor da coleção de todos os participantes do projeto antes da inclusão do novo participante.

Outra palavra-chave importante ao se utilizar operações é *result*. Através dela, pode-se fazer referência ao valor a ser retornado pela operação. Modificando-se o exemplo anterior para ilustrar seu uso, tem-se uma nova função cujo retorno, indicado por *result*, é o resultado da expressão booleana *participantes = participantes@pre -> including(p)*:

context

```

Projeto::adicionar_participantes( p: Empregado ):Boolean
pre: self.departamento.empregados -> includes( p )
pos: result = ( participantes = participantes@pre ->
including(p) )

```

2.5.9 Comentários

Assim como em linguagens de programação e diagramas UML, a OCL também permite que sejam acrescentados comentários para facilitar a compreensão do que está sendo expresso. Um comentário de linha é representado por dois sinais de menos (--), sendo considerado como comentário tudo deste ponto até o fim da linha. Um comentário de uma ou mais linhas deve ficar entre os marcadores /* e */.

O exemplo a seguir ilustra a sintaxe para utilização de comentários:

```

/*
  as seguintes expressões ilustram a interface da operação
  adicionar_participantes da classe Projeto.
  Para se adicionar um novo participante ao projeto o
  empregado passado como parâmetro deve trabalhar no mesmo
  departamento ao qual pertence o projeto.
  Ao final da execução, o novo participante deve estar
  incluso na coleção de participantes do projeto.
*/

```

```

context Projeto::adicionar_participantes( p: Empregado )
pre: self.departamento -- departamento do projeto
      .empregados -- empregados do departamento
      -> includes( p ) -- empregado trabalha no departamento
pos: participantes = -- participantes ao final da execução
      -- com expressão booleana
      participantes@pre -- participantes no início da execução
      -> including( p )-- incluir empregado

```

2.6 Extensão da OCL para especificação de restrições de integridade espaciais

Apesar de toda simplicidade e expressividade da OCL, esta não é utilizada na modelagem de SIG devido a sua versão atual não possuir construtores para expressões envolvendo elementos geográficos. Friis-Christensen, Tryfona e Jensen (2001) destacam a importância em se adicionar elementos geográficos na OCL para modelagem desse tipo de aplicação.

Duboisset et al. (2005) propõem a extensão da OCL através da inserção de operações para se especificar relacionamentos topológicos entre polígonos segundo o modelo de 9-interseções de Egenhofer e Franzosa (1991). Através desta extensão, é

possível se especificar os seguintes tipos de relacionamentos: disjunção, contém, dentro, igual, toca, cobre, coberto por e sobreposição. A extensão segue as seis definições a seguir:

1. A primeira acompanha a definição de polígono e região apresentada na seção 2.1, sendo considerados apenas os polígonos sem buraco;
2. Os modelos de dados espaciais devem possuir no mínimo os conceitos de classes com atributos e relacionamentos, sendo a unicidade estabelecida através do menor conjunto de atributos que identifiquem unicamente uma instância. Todos os atributos devem possuir um tipo, incluindo os tipos polígono e região para dados espaciais. As classes que incluam esses dois atributos serão consideradas classes espaciais. O tipo região será identificado segundo a notação de conjunto da OCL, sendo um conjunto de polígonos: *Set(Polígonos)*.
3. São adicionadas oito operações à OCL correspondentes aos relacionamentos topológicos definidos em Egenhofer e Franzosa (1991). A sintaxe de uso dessas operações é a seguinte:

```
A -> relação_topologica( B )
```

Os parâmetros A e B são do tipo Polígono e o retorno da função é do tipo booleano com seu valor definido segundo a avaliação do relacionamento topológico.

O seguinte exemplo ilustra uma restrição de disjunção entre duas classes associadas, *Classe_1* e *Classe_2*. O atributo *geometry* representa o atributo espacial do tipo polígono:

```
context Classe_1  
inv: self.geometry -> disjoint(self.Class_2.geometry)
```

4. Como as novas operações da OCL estendida aceitam argumentos apenas do tipo polígono, operações de conjunto da OCL padrão podem ser utilizadas em atributos do tipo região para se acessar individualmente cada elemento e então serem aplicadas as operações necessárias. A sintaxe de uso das operações de conjunto é a seguinte:

atributo_geometrico -> operação de conjunto (...)

O exemplo a seguir tem como objetivo ilustrar o uso de tipos geométricos em operações de conjunto aliado aos conceitos do UML-GeoFrame. Segundo o diagrama da Figura 2.16, instâncias da classe *Lote* são representadas por um polígono e compõem uma instância da classe *Quadra*. Através da OCL estendida, essa composição pode ser formalizada e melhor especificada por uma restrição em que lotes devem possuir somente relacionamentos espaciais do tipo *coberto por* com a quadra associada. Sem essa restrição, ficaria subtendido também o relacionamento do tipo *dentro*:

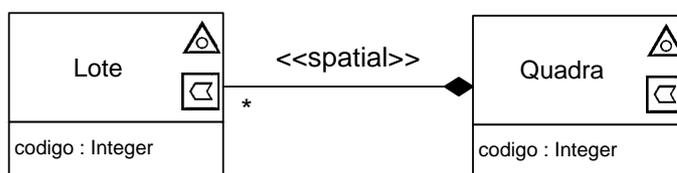


Figura 2.16 – Composição espacial entre Lote e Quadra

```
context Quadra
inv: self.geometry -> forAll( l:Lote |
    l.geometry -> coveredBy(self.geometry) )
```

5. Para que não seja necessário o uso de operações de conjunto em atributos do tipo Região, foram acrescentadas novas operações sobre a OCL estendida até a quarta definição. Essas operações têm como objetivo facilitar a especificação de restrições entre duas regiões de acordo com o modelo de advérbio de Claramunt (2000), sendo possível se adicionar sete advérbios aos oito tipos de relacionamentos definidos em Egenhofer e Franzosa (1991). Os advérbios são: em sua maioria, em sua maioria reverso, completamente, parcialmente, ocasionalmente, inteiramente e nunca. A definição da semântica de cada advérbio foge ao escopo deste trabalho, sendo apresentadas em Claramunt (2000). A sintaxe de uso dessas operações é a seguinte:

```
A -> relação_topologica( advérbio, B )
```

Os parâmetros A e B são do tipo Região e o retorno da função é do tipo booleano, com seu valor definido segundo a avaliação do relacionamento topológico. Se A ou B forem conjuntos vazios, a operação retornará falso.

6. Em um dado relacionamento entre A e B, é possível se determinar o número de partes de A que se relacionam com B através da seguinte sintaxe:

```
A -> select( parte_de_A | B -> exists( parte_de_B |
      parte_de_A -> relação_topologica( parte_de_B )))
      -> size() = s
```

Os parâmetros A e B são do tipo Região, *relação_topológica* representa um dos oito tipos de relacionamentos definidos em Egenhofer e Franzosa (1991) e *s* é o número correspondente de partes no relacionamento.

Além da extensão da OCL, Duboisset et al. (2005) propõem a extensão da ferramenta OCL2SQL (DEMUTH e HUSSMANN, 1999; DEMUTH, HUSSMANN e LOECHER, 2001), aplicação que permite a geração de código SQL a partir de expressões da OCL padrão. Com a extensão, o projetista deverá ser capaz de especificar restrições topológicas através da OCL proposta e gerar mecanismos de integridade em diferentes bancos de dados espaciais, embora inicialmente a extensão se limite aos mecanismos do SGBD Oracle.

3 EXTENSÃO DO MODELO UML-GEOFRAME PARA MODELAGEM DE REDE

Entre os modelos conceituais específicos para SIG, um problema importante que tem recebido pouca atenção diz respeito à modelagem de restrições de integridade envolvendo elementos de rede, como elementos de uma rede de distribuição de água, viária, hidrográfica, de telefonia, de energia etc. Um dos modelos que aborda este problema em maior profundidade é o modelo GeoOOA (KÖSTERS, PAGEL e SIX, 1997). O modelo OMT-G (BORGES, DAVIS e LAENDER, 2001), apesar de mais recente, fornece construtores simples para modelagem de rede, mas não aborda o problema com o mesmo nível de detalhamento do modelo GeoOOA.

Com isso, o objetivo deste capítulo é estender o modelo UML-GeoFrame para que este forneça construtores específicos para modelagem de elementos de rede e mostrar como a OCL pode ser empregada como linguagem de especificação formal de restrições de integridade neste domínio.

As seções a seguir apresentam as diversas etapas da evolução do GeoFrame, apontando as vantagens e desvantagens encontradas em suas versões preliminares. A versão final proposta é descrita na seção 3.5. As restrições de integridade propostas para o modelo de redes são apresentadas na seção 3.8.

3.1 Inclusão de propriedades de rede em classes geométricas

A primeira idéia de extensão do GeoFrame se restringia à inclusão de propriedades de rede nas classes espaciais *Ponto* e *Linha*. A vantagem em se utilizar esta abordagem é aproveitar as características geométricas de ponto e linha considerando-se respectivamente a representação geométrica para os elementos nó e arco, embora estes não dependam de informações como formato, comprimento, posição, orientação e transformação. Outra vantagem é a não inclusão de novos estereótipos para se representar estes elementos, apenas a inclusão de atributos referentes à conectividade seriam necessários para se especificar o seu tipo.

Demirel (2002) cita como vantagens desta abordagem a redução dos requisitos de armazenamento dos dados e o aumento de desempenho do

processamento de consultas, mas destaca as desvantagens em relação à manutenção dos dados. Embora os dados topológicos não dependam das propriedades geométricas, alterações na geometria dos objetos necessariamente ocasionariam a manutenção tanto de dados geométricos quanto topológicos.

3.2 Especialização a partir de classes geométricas

A segunda alternativa para extensão do GeoFrame teve como objetivo evitar os problemas encontrados na primeira proposta através da especialização das classes *Nó* e *Arco* a partir de *Ponto* e *Linha*, respectivamente. A Figura 3.1 ilustra essa alternativa.

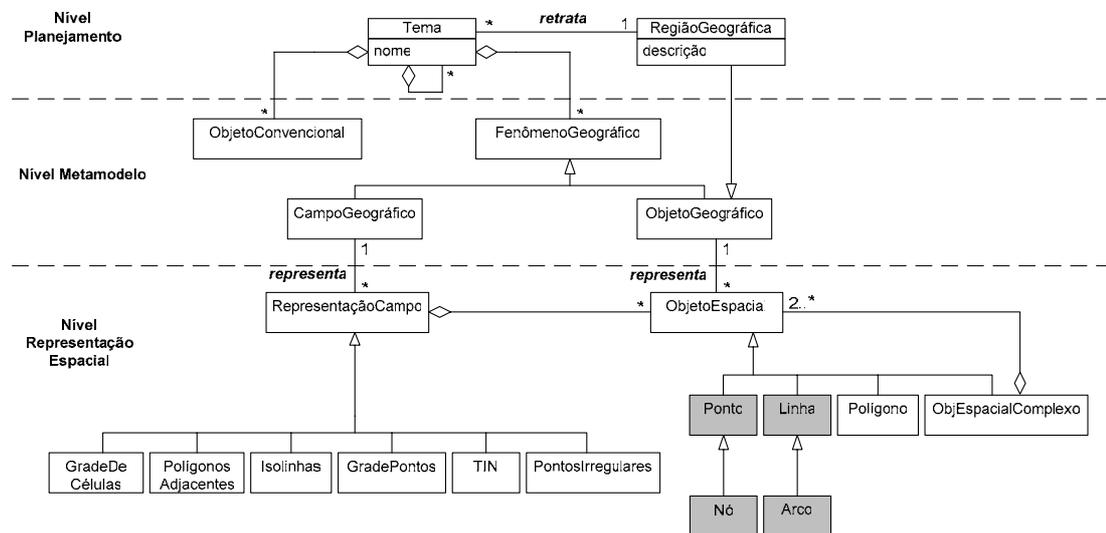


Figura 3.1 – Especialização de Nó e Arco a partir de Ponto e Linha

Esta solução possui as mesmas vantagens da proposta anterior, com exceção da inclusão de novos estereótipos, e resolve o problema relativo à manutenção dos dados topológicos quando houver alteração nos dados geométricos. No entanto, da mesma forma como destacado na seção 2.3.4, existe uma falha em raciocínios deste tipo, pois o conceito de generalização/especialização não recomenda que coisas distintas se relacionem desta forma.

3.3 Hierarquia de classes para modelagem de rede

Como a solução empregando o conceito de generalização/especialização se mostrou inadequada, optou-se pela inclusão de uma nova estrutura de classes hierárquica para fornecer os elementos fundamentais para a modelagem de rede. A Figura 3.2 apresenta esta abordagem.

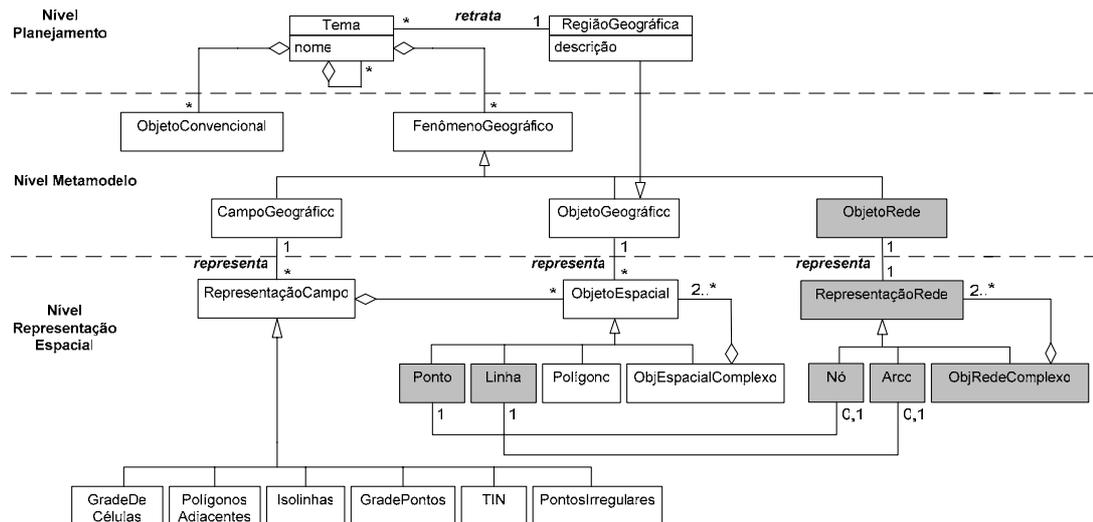


Figura 3.2 – Hierarquia de classes para modelagem de redes

No nível de metamodelo, foi acrescentada uma nova especialização a partir da classe *FenômenoGeográfico*, sendo possível através da classe *ObjetoRede* se interpretar a realidade através de elementos de rede. No nível de representação espacial, a classe *RepresentaçãoRede* generaliza as classes de representação de rede, sendo estas *Nó*, *Arco* e *ObjRedeComplexo*. Esta última representa um objeto de rede composto de dois ou mais objetos de rede.

Apesar da similaridade hierárquica com a visão de campo e objetos, a modelagem de redes possui algumas particularidades. A primeira delas refere-se à multiplicidade na associação entre as classes *ObjetoRede* e *RepresentaçãoRede*, identificada como *representa*. Como essa associação possui multiplicidade um para um (1:1), uma instância da classe *ObjetoRede* deve ser representada por uma única instância da classe *RepresentaçãoRede*, ou seja, não há múltiplas representações.

A segunda particularidade se refere à associação entre *Nó* e *Ponto* assim como *Arco* e *Linha* para representar o aspecto geométrico das classes de rede. Através dessa característica, é imposta a restrição de que em qualquer diagrama que

utilize o GeoFrame para modelagem de rede deverá existir uma classe do tipo *Ponto* associada a uma classe do tipo *Nó*, da mesma maneira que deverá existir uma classe do tipo *Linha* associada a uma classe do tipo *Arco*. Esses relacionamentos se tornam opcionais quando a aplicação envolver apenas fenômenos geográficos na visão de campos e/ou objetos.

A terceira característica se refere à classe *ObjRedeComplexo*. Apesar da aparente similaridade com a classe *ObjEspacialComplexo*, esta classe possui um significado diferente. Toda aplicação de representação de rede sempre fará uso da classe *ObjRedeComplexo*, uma vez que uma rede é uma agregação de nós e arcos. Sendo uma agregação, um mesmo nó ou arco pode pertencer a diversas redes. A classe *ObjEspacialComplexo*, por sua vez, nem sempre está presente na modelagem de uma aplicação de representação segundo a visão de objetos.

Esta solução, apesar de atender ao menos os requisitos básicos para modelagem de aplicações de rede, ainda possui alguns problemas a serem resolvidos. O primeiro deles diz respeito à associação entre os objetos de representação geométrica e topológica. Como muitas características geométricas não são pertinentes às classes *Nó* e *Arco*, as associações destas classes com *Ponto* e *Linha* tornam possível a especificação de consultas a partir das classes topológicas com acesso a características que não sejam de conectividade. Exemplo disso é a consulta a partir de uma classe do tipo *Arco* a valores de atributos como comprimento, curvatura ou espessura de uma linha.

Outro problema encontrado refere-se à representação de uma rede através da classe *ObjRedeComplexo*. Uma vez que este objeto é obrigatório na modelagem de rede e não acrescenta quaisquer atributos espaciais aos objetos agregados, uma nova solução tornou-se necessária para que esta classe fosse substituída por um conceito mais abstrato.

3.4 Reformulação da hierarquia de classes para modelagem de rede

Para solucionar os problemas citados, tornou-se necessária uma modificação na hierarquia de classes anterior, conforme ilustra a Figura 3.3.

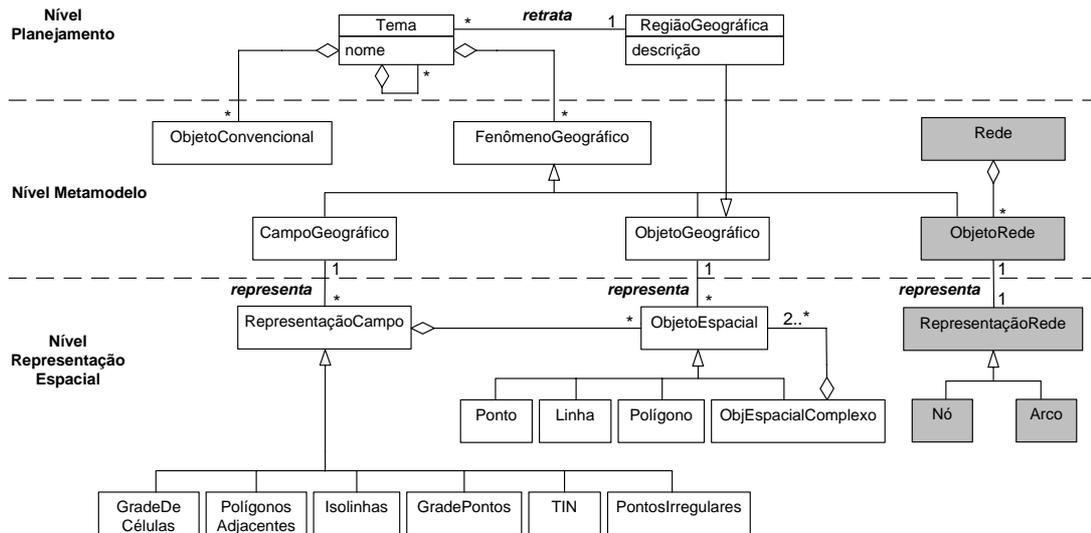


Figura 3.3 – Reformulação da hierarquia de classes para modelagem de redes

Foi removida a associação entre *Nó* e *Ponto*, assim como entre *Arco* e *Linha*, para que apenas as características geométricas essenciais sejam utilizadas pelos elementos de rede. Além disso, os elementos de uma rede podem estar associados a diferentes tipos de objetos espaciais. Exemplo disso é que, tanto uma usina representada como polígono, quanto um poste representado como ponto, podem exercer os papéis de nós em uma rede de energia elétrica.

No entanto, a modificação de maior importância foi a substituição da classe *ObjRedeComplexo* no nível de representação espacial pela classe *Rede* no nível de metamodelo. Como na versão anterior (Figura 3.2) apenas informações alfanuméricas eram adicionadas aos objetos de rede agregados, a interpretação de uma rede passou a ser a mesma de uma classe convencional, sendo mantida, porém, a agregação de objetos de rede. Essa agregação entre *Rede* e *ObjetoRede* implica que, no diagrama da aplicação, uma instância de *Rede* será uma agregação de instâncias de *ObjetoRede*, possibilitando a manipulação da rede como um objeto composto.

Através dessa agregação, é possível também que uma instância de *ObjetoRede* pertença a mais de uma rede.

Neste ponto, duas interpretações são possíveis. Na primeira, toda classe composta por classes espaciais é também uma classe espacial. No caso da visão de objetos, essa classe é representada por um *ObjEspacialComplexo*. Na segunda, esse tipo de relacionamento não necessariamente implica que a classe que sofre a composição seja também espacial, podendo ser uma classe do tipo *ObjetoConvencional*. A Figura 3.4, embora não utilize os construtores de rede propostos, exemplifica ambas as interpretações apresentadas.

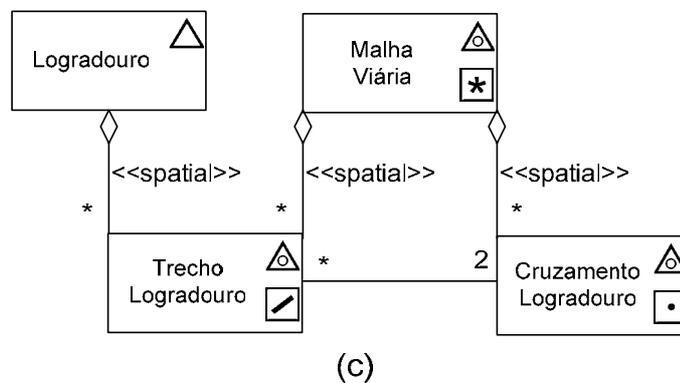


Figura 3.4 – Interpretação da representação de classes com uso de agregações espaciais

A classe *MalhaViária*, do tipo *ObjEspacialComplexo*, é utilizada para representar a agregação de todos os objetos espaciais que compõem a rede viária de uma cidade, representados pelos trechos de logradouro e cruzamentos com representações espaciais linear e pontual, respectivamente. A classe *Logradouro*, por outro lado, é utilizada para associação de dados alfanuméricos comuns aos trechos de logradouro agregados.

Como pode ser observado no exemplo acima, a agregação espacial torna a representação de uma classe subjetiva à interpretação do projetista. Dessa forma, semelhante ao que ocorre com a classe *Logradouro*, optou-se por representar a classe *Rede* no nível de metamodelo do UML-GeoFrame como uma classe convencional com agregação de objetos de rede.

3.5 Novas classes para redes bidirecionais e unidirecionais

Uma última modificação foi feita na hierarquia proposta para que seja possível se especificar redes bidirecionais e unidirecionais. Essa nova hierarquia, apresentada na Figura 3.5, é a proposta final deste trabalho para a extensão do GeoFrame.

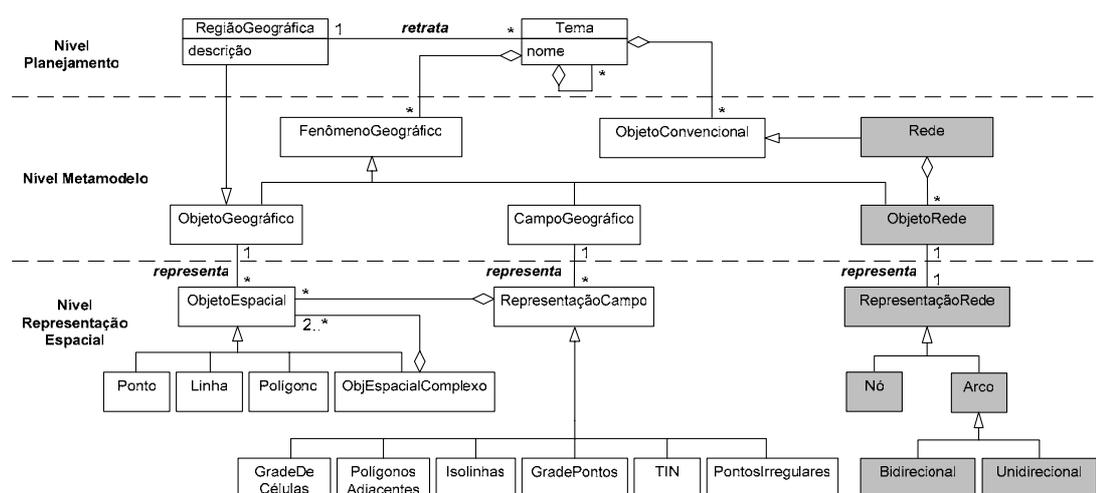


Figura 3.5 – Proposta final de extensão do GeoFrame para modelagem de redes

Apesar das estruturas de dados tanto para nós quanto para arcos serem diferentes nestes dois tipos de redes, a representação, por outro lado, difere apenas pelo formato do arco. Para o GeoFrame, as necessidades de se especificar o tipo da rede e manter a simplicidade do modelo baseiam-se na representação, sendo, portanto, proposta apenas a especialização da classe *Arco* nas classes *Bidirecional* e *Unidirecional*.

Outra modificação foi feita para se especializar a classe *Rede* a partir da classe *ObjetoConvencional*. O objetivo desta alteração é tornar mais clara a interpretação da classe *Rede* como sendo uma classe convencional e mostrar que toda instância da classe *Rede* deve pertencer a um pacote (classe *Tema*).

3.6 Estereótipos de rede

Com as novas classes adicionadas ao *framework*, foram definidos também novos estereótipos. Estes seguem os mesmos princípios do UML-GeoFrame, com um estereótipo definido para a especialização do fenômeno geográfico e outro para a representação espacial. A classe *Rede* utiliza o estereótipo de objeto convencional. A Figura 3.6 apresenta exemplos de classes com o estereótipo de generalização da metaclassa *ObjetoRede* (), e os estereótipos de representação das metaclassas *Nó* (), arco *Bidirecional* () e arco *Unidirecional* (), respectivamente.

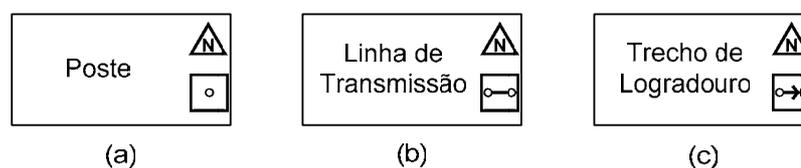


Figura 3.6 – Estereótipos de generalização e representação de rede

3.7 Definição dos elementos de representação de rede

Simplicidade e expressividade são características do modelo UML-GeoFrame que facilitam a compreensão da realidade modelada. Por outro lado, implicam em informações textuais adicionais para se construir um diagrama completo a respeito do problema. Modelos como o UML-GeoFrame, que disponibilizam construtores simples para o desenvolvimento de diagramas simples, possuem também informações textuais para facilitar a compreensão dos seus conceitos.

Informações textuais incorporadas ao modelo são úteis para se definir cada uma das classes ou mesmo estabelecer restrições comuns a diversos domínios. A classe *Linha*, por exemplo, é definida no UML-GeoFrame como uma representação espacial na visão de objetos utilizada pelo projetista na interpretação de um fenômeno geográfico segundo uma forma geométrica linear, como, por exemplo, um segmento de linha, uma polilinha ou mesmo uma curva.

Da mesma forma, a classe *Nó* é uma representação espacial na visão de objeto de rede utilizada pelo projetista quando o extremo das linhas for representado através

de um ponto com propriedades de conectividade, independente da posição real ou transformação aplicada ao elemento da realidade retratado.

A classe *Arco* é uma representação espacial na visão de objeto de rede utilizada pelo projetista para expressar a conectividade de um ou dois nós através de uma linha. É também independente da posição real ou transformação aplicada ao elemento da realidade retratado.

3.8 Estrutura de dados para grafos

Até o momento, os relacionamentos entre as classes de rede foram abordados de modo superficial. A partir deste ponto, são detalhados esses relacionamentos em redes bidirecionais e unidirecionais considerando-se as estruturas de dados das classes envolvidas. Apesar do UML-GeoFrame não abordar o nível de estrutura de dados, este nível é fundamental para que restrições de integridade sejam especificadas para todas as aplicações que utilizem os construtores propostos. Este tipo de restrição deve ser implícito aos diagramas e considerado somente na fase de implementação. O objetivo é poupar o projetista de detalhes pertinentes à estrutura de dados, possibilitando que apenas diagramas e informações pertinentes à aplicação sejam elaborados. Ao mesmo tempo, a especificação da aplicação deve ser a mais completa possível para que sejam evitadas ambigüidades.

Como através das classes *Nó* e *Arco* é possível se criar grafos bidirecionais e unidirecionais, existem na realidade dois tipos de relacionamentos entre estas classes. Baseando-se nos diagramas encontrados em Laurini e Thompson (1992) para esses dois tipos de grafos e adequando-os para as necessidades do modelo UML-GeoFrame, são propostos então dois diagramas de estrutura de dados para representação lógica desses relacionamentos (Figura 3.7).

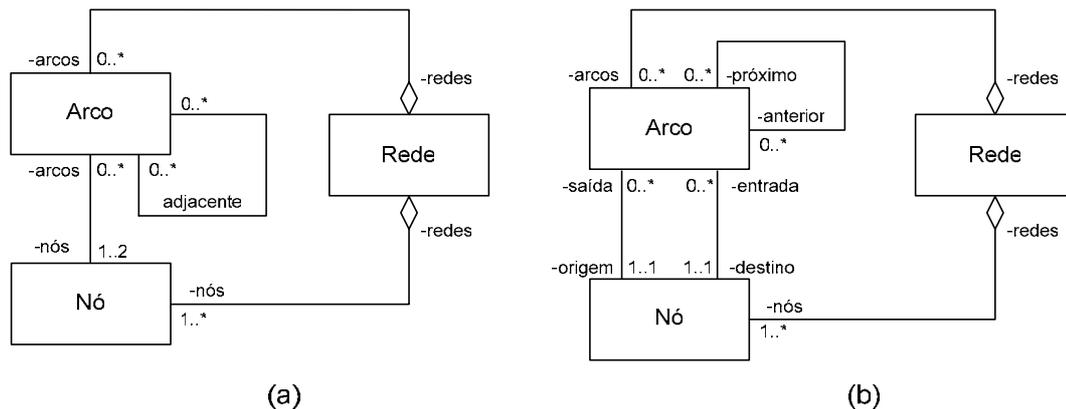


Figura 3.7 – Diagramas de grafos bidirecionais e unidirecionais

A Figura 3.7a apresenta um diagrama de grafo bidirecional no qual cada arco está associado a dois nós distintos ou a um único nó em ambas as extremidades, e cada nó pode estar associado a zero ou muitos arcos. A associação recursiva na classe *Arco* pode ser utilizada para se criar cadeias ou caminhos de arcos. Um arco pode estar associado a diversos outros ou ser o único arco do grafo. Uma rede deve ser uma agregação de um ou mais nós e arcos. Nós e arcos, por sua vez, podem ser partes de mais de uma rede.

Segundo o diagrama de grafo unidirecional apresentado na Figura 3.7b, cada arco deve estar associado a exatamente dois nós, não necessariamente diferentes, de origem e destino. A associação recursiva da classe *Arco* torna possível a criação de uma seqüência de arcos. Um arco pode estar associado seqüencialmente com vários arcos ou ser o único arco do grafo. Uma rede deve ser uma agregação de um ou mais nós e arcos. Nós e arcos, por sua vez, podem ser partes de mais de uma rede.

3.8.1 Restrições de integridade para grafos bidirecionais

Embora as associações entre as classes *Nó*, *Arco* e *Rede* possam ser representadas pelos diagramas da Figura 3.7, nem toda informação necessária é transmitida de forma completa e precisa. Por este motivo, é utilizada a OCL para complementar formalmente a especificação dos relacionamentos entre estas classes. A seguir, são apresentadas oito invariáveis OCL que especificam formalmente as restrições de integridade identificadas.

As restrições de integridade de (i) à (iv) referem-se ao diagrama de grafo bidirecional. A restrição (i) aborda o relacionamento de *Rede* com *Arco* e *Nó*, na qual todos os nós devem pertencer às mesmas redes de seus arcos associados:

```
(i) context Rede
inv: self.arcos.nós -> forAll( n | n.redes -> includes(self))
```

O contrário não pode ser afirmado, ou seja, um arco não necessariamente pertence à mesma rede de seus nós associados. Por exemplo, um nó que estabelece a conexão entre duas redes pertence a ambas. Os arcos associados a este nó, por sua vez, continuam participando cada um de sua rede.

As restrições de (ii) à (iv) abordam o relacionamento recursivo da classe *Arco*, uma vez que a cardinalidade determina esse relacionamento como sendo opcional e não se pode deduzir com facilidade quando um arco deve se relacionar com outro para que seja formado um caminho.

A restrição (ii) garante que um arco não será adjacente a si mesmo. Essa regra evita uma ambigüidade no caso de um arco possuir um mesmo nó como seus extremos:

```
(ii) context Arco
inv: self.adjacente -> forAll( a | a <> self )
```

A restrição (iii) impõe que todos os arcos com nó(s) em comum devem formar um caminho:

```
(iii) context Nó
inv: self.arcos -> forAll( a | a.adjacente ->
    includesAll( self.arcos -> excluding(a)))
```

A restrição (iv) impõe que todos os arcos adjacentes devem possuir ao menos um nó em comum:

```
(iv) context Arco
inv: self.adjacente -> forAll( a | a.nós ->
    forAll( nL1,nL2 | nL1 <> nL2 and self.nós ->
    forAll( nS1,nS2 | nS1 <> nS2 and ( nS1 = nL1 or nS1 = nL2
    or nS2 = nL1 or nS2 = nL2 ))))
```

É importante se observar que, devido à agregação com várias redes, nada se pode afirmar sobre a conectividade entre os arcos e as redes às quais estes pertencem. Um arco pode ser adjacente a outro, mas os dois não necessariamente pertencem às mesmas redes.

3.8.2 Restrições de integridade para grafos unidirecionais

As restrições de integridade de (v) à (viii) referem-se ao diagrama de grafo unidirecional.

A restrição (v) garante que todos os nós devem pertencer às mesmas redes de seus arcos associados:

```
(v) context Rede
inv: self.arcos.origem.redes -> includes( self )
      and
      self.arcos.destino.redes -> includes( self )
```

A restrição (vi) garante que um arco não será adjacente a si mesmo:

```
(vi) context Arco
inv: self.próximo -> forAll( a | a <> self )
      and
      self.anterior -> forAll( a | a <> self )
```

A restrição (vii) impõe que os arcos que possuam nó(s) em comum e que alternem entre a origem e destino, devem formar uma seqüência:

```
(vii) context Nó
inv: self.saída -> forAll( a | a.anterior ->
      includes( self.entrada -> excluding(a)))
      and
      self.entrada -> forAll( a | a.próximo ->
      includes( self.saída -> excluding(a)))
```

A restrição (viii) impõe que todos os arcos adjacentes devem possuir ao menos um nó em comum de forma que seja alternada entre a origem e destino:

```
(viii) context Arco
inv: self.anterior -> forAll( a | a.destino = self.origem)
      and
      self.próximo -> forAll( a | a.origem = self.destino)
```

No próximo capítulo, estudos de caso são utilizados para apresentar a importância da proposta de extensão do modelo UML-GeoFrame para elaboração de esquemas conceituais de bancos de dados geográficos em aplicações que manipulam elementos cujos relacionamentos formam uma rede. Os diagramas são acompanhados de expressões OCL para que informações complementares à aplicação sejam transmitidas de forma completa e precisa.

4 ESTUDOS DE CASO PARA MODELAGEM DE APLICAÇÕES DE REDE

Para exemplificar o uso dos construtores de rede propostos ao modelo UML-GeoFrame, são utilizados dois estudos de caso. O primeiro aborda um sistema de distribuição de água em área urbana. O segundo aborda um sistema de distribuição de energia elétrica, proposto por Kösters, Pagel e Six (1997) como forma de exemplificar o uso dos elementos de rede do modelo GeoOOA. O primeiro estudo de caso faz uma comparação entre os construtores de rede dos modelos UML-GeoFrame e OMT-G. No segundo estudo de caso foi feita a mesma comparação entre os modelos UML-GeoFrame e GeoOOA.

As comparações mostram como esses modelos atendem ao conjunto de requisitos identificados por Kösters, Pagel e Six (1997): (a) deixar claro quais classes são do tipo arco, nó e rede, sem que esse tipo seja identificado pelos nomes das classes; (b) mostrar a quais redes pertencem os arcos e nós; (c) mostrar quais arcos são incidentes a quais nós; (d) identificar as classes que possuem propriedades em comum (uso de superclasse); (e) deixar claro qual classe representa o elo entre duas ou mais redes. Outros requisitos identificados durante a extensão do modelo UML-GeoFrame são considerados também nas comparações.

4.1 Estudo de caso: Sistema de distribuição de água

O primeiro estudo de caso descreve a modelagem de um sistema real e contou com a colaboração de funcionários do Serviço Autônomo de Água e Esgoto (SAAE) do município de Viçosa (SAAE, 2008). Os termos empregados na descrição do problema baseiam-se nas definições de Barros et al. (1995). São apresentados apenas os elementos fundamentais à compreensão do sistema, desde a distribuição a partir da Estação de Tratamento de Água (ETA) até o hidrômetro do consumidor.

Foram identificadas duas redes para a distribuição de água dentro de um município: a rede principal, responsável pelo transporte da água desde a ETA até os pontos de conexão com condutos secundários, passando por reservatórios; e a zona de medição e controle, composta da maior parte da malha de distribuição e responsável pelo abastecimento dos consumidores.

Os reservatórios têm como objetivo atender as variações de consumo, as demandas de emergência e manter a pressão mínima ou constante na rede. Quanto à localização na rede, os reservatórios podem ser de dois tipos: reservatório de montante, situado a montante da rede de distribuição, para o suprimento normal; reservatório de jusante, responsável por armazenar as sobras da rede nos horários de menor demanda e abastecer a rede nas horas de maior consumo (Barros et al., 1995).

A adutora de água tratada é uma tubulação de maior diâmetro e responsável pelo transporte desde a ETA até o reservatório de montante, que por sua vez pode ser alimentado por diversas adutoras. Junto à adutora, pode existir também um macro medidor, equipamento utilizado para medir a vazão e a pressão da água. Portanto, caso esse equipamento esteja instalado, um trecho de adutora poderá ter como extremidades uma ETA e um macro medidor ou um macro medidor e um reservatório de montante.

Os condutos principais são os responsáveis pelo transporte da água desde os reservatórios até os diversos pontos de conexão com os ramais. Cada trecho do conduto principal pode ter como extremidades um reservatório, uma conexão com um ramal ou uma conexão entre trechos do conduto principal.

A conexão principal é o elo entre os dois tipos de rede, sendo responsável pela conexão entre um conduto principal (de maior diâmetro) e os ramais da rede (de menor diâmetro). Cada trecho do ramal possui ainda mais três tipos de conexões: conexão entre ramais; conexão entre ramais e ramais prediais (dispositivo de tomada); ponto final. Além disso, semelhante a uma adutora, um ramal pode ter entre seus trechos um macro medidor (mecanismo para medição), assim como um registro, uma válvula reguladora de pressão ou uma bomba de elevação (mecanismos para controle).

O registro tem como função permitir a interrupção do fluxo de água a partir do trecho de ramal onde está instalado. A escolha correta de qual registro fechar quando houver necessidade de realizar manutenção na rede é importante, pois evita que um número excessivo de pessoas seja atingido pela queda no abastecimento de água durante o processo. A válvula reguladora de pressão é utilizada para se determinar e manter a pressão constante da água em um setor. Uma bomba de elevação permite à rede de distribuição transpor os obstáculos impostos pelo relevo, bombeando a água de um nível mais baixo para outro mais alto.

O ramal predial é o conduto que se conecta ao dispositivo de tomada e permite que a água seja transportada da rede pública ao micro medidor (hidrômetro) do consumidor.

A Figura 4.1 ilustra alguns elementos de uma rede de distribuição de água. A Figura 4.2 apresenta um diagrama de classes para este sistema, elaborado utilizando os construtores de rede do UML-GeoFrame.

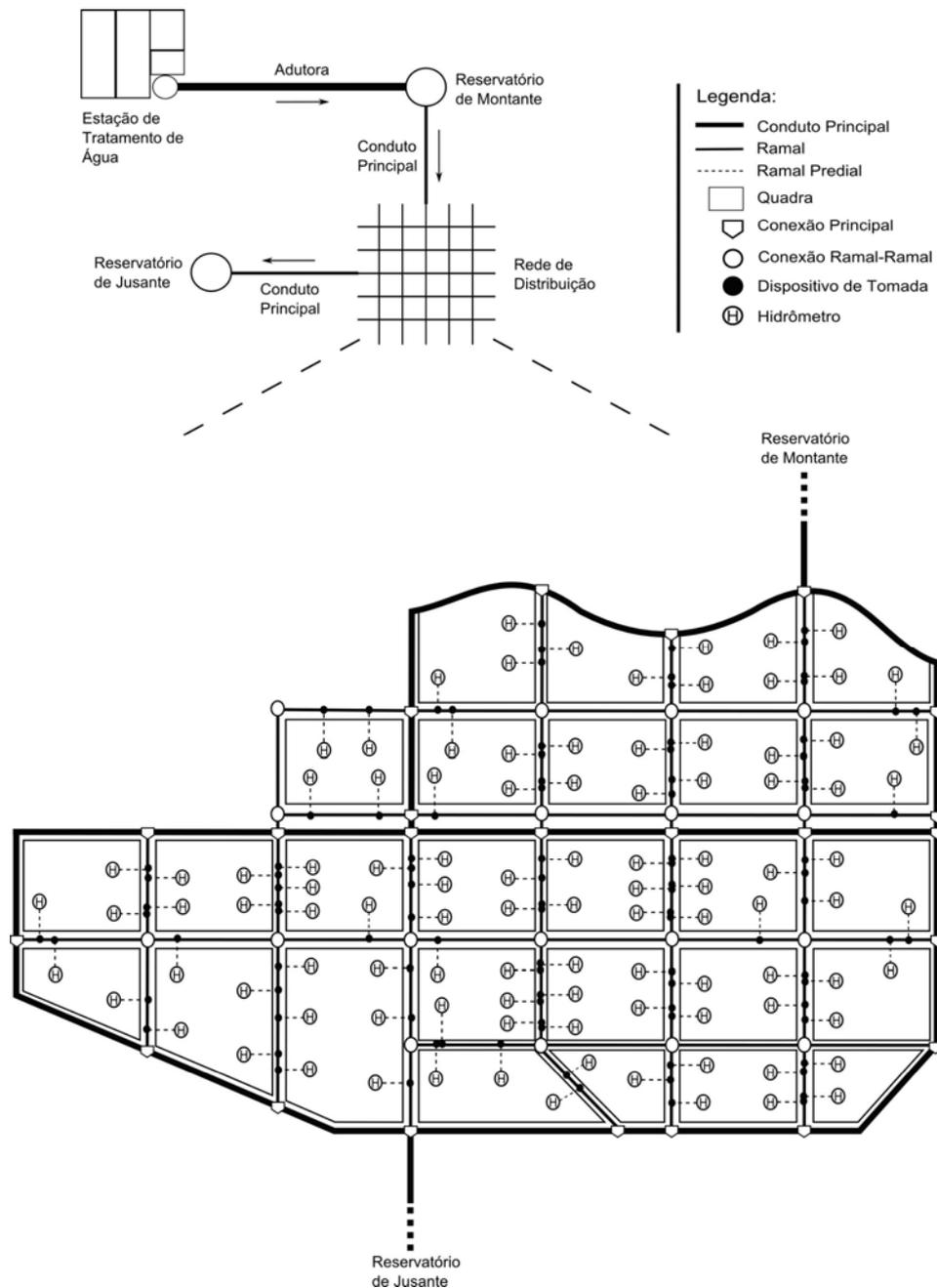


Figura 4.1 – Ilustração de um sistema de distribuição de água
Adaptado de Barros et al. (1995)

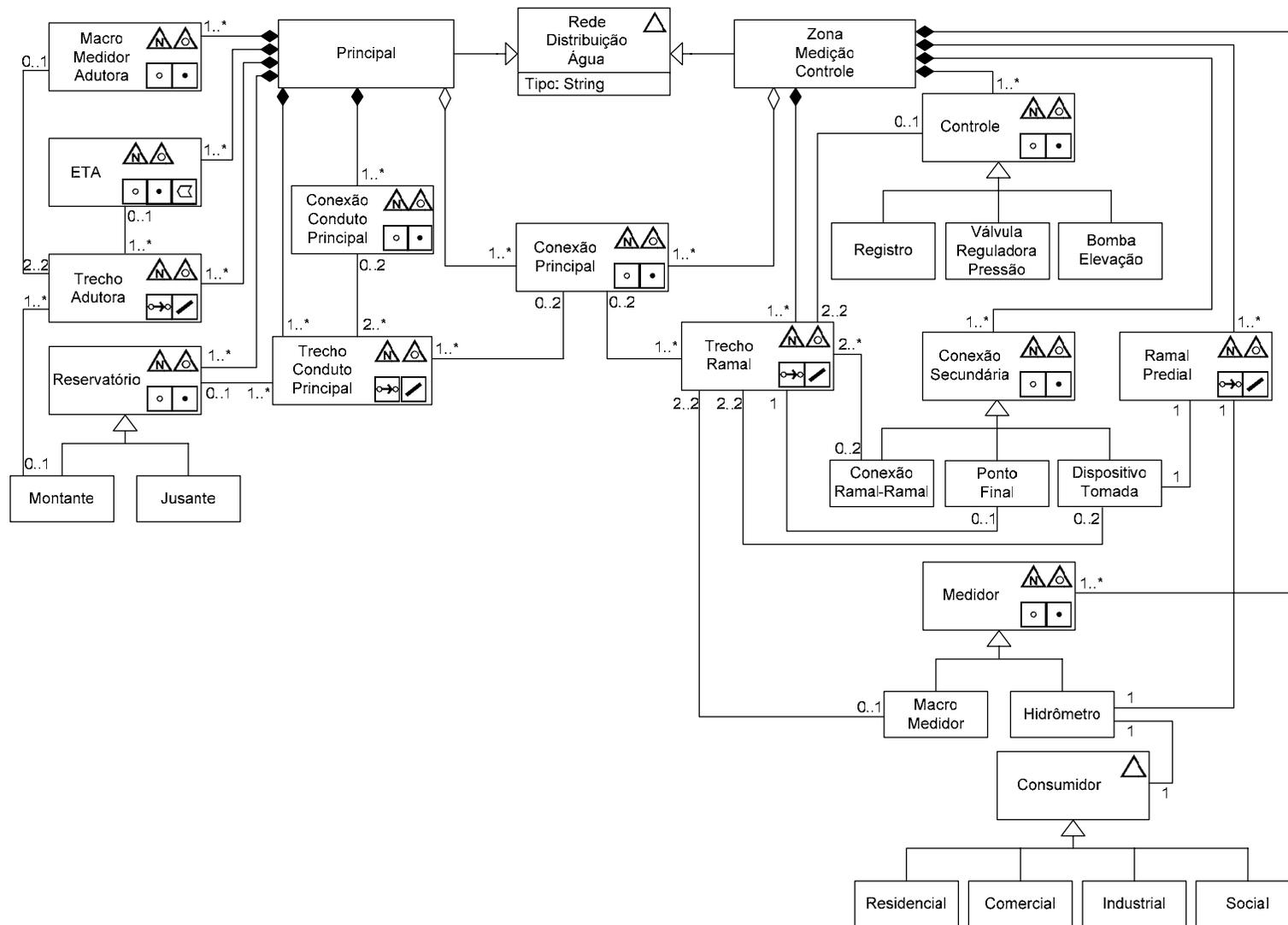


Figura 4.2 – Diagrama de classes de uma companhia de distribuição de água elaborado utilizando os construtores de rede do UML-GeoFrame

Quando nós e arcos correspondem a objetos espaciais, a rede é chamada de “especialmente embutida”. Exemplos desse tipo de rede são: malha viária, energia elétrica, abastecimento de água e oleodutos (Kösters, Pagel e Six, 1997). Um contra-exemplo é a construção de uma rede sobre os relacionamentos de adjacência entre os bairros de um município.

Para que fique claro esse relacionamento entre os elementos de uma rede e a representação de objetos espaciais, optou-se por utilizar o conceito de herança múltipla nas classes do diagrama. Para que isso ocorra, uma classe deve ser uma especialização das classes *ObjetoGeográfico* e *ObjetoRede*, e deve se associar às possíveis representações espaciais de ambas as visões. A substituição dessas especializações e associações é realizada através do uso dos estereótipos de generalização e representação. Para a visão de objetos, continua sendo possível o uso de múltiplas representações espaciais.

Na Figura 4.2, é possível se perceber que a maioria das classes apresenta quatro estereótipos. Esse número pode ser ainda maior caso seja utilizada a múltipla representação espacial, como pode ser observado na classe *ETA*, que desempenha o papel de nó na rede principal e ao mesmo tempo possui representação pontual ou poligonal como objeto geográfico. A posição dos estereótipos dentro das classes é uma sugestão, o projetista pode posicioná-los da forma que lhe convir. O uso de estereótipos espaciais, apesar de simplificar a visualização do diagrama, ainda assim não evitou que este ficasse sobrecarregado.

Para evitar esta situação, destaca-se como característica do UML-GeoFrame a opcionalidade de se exibir os estereótipos de generalização, uma vez que os estereótipos de representação espacial para as visões de campo, objetos e rede são conjuntos disjuntos. Dessa forma, uma classe pode exibir apenas os estereótipos de representação espacial. A Figura 4.3 reproduz o diagrama de classes da Figura 4.2 sem os estereótipos de generalização.

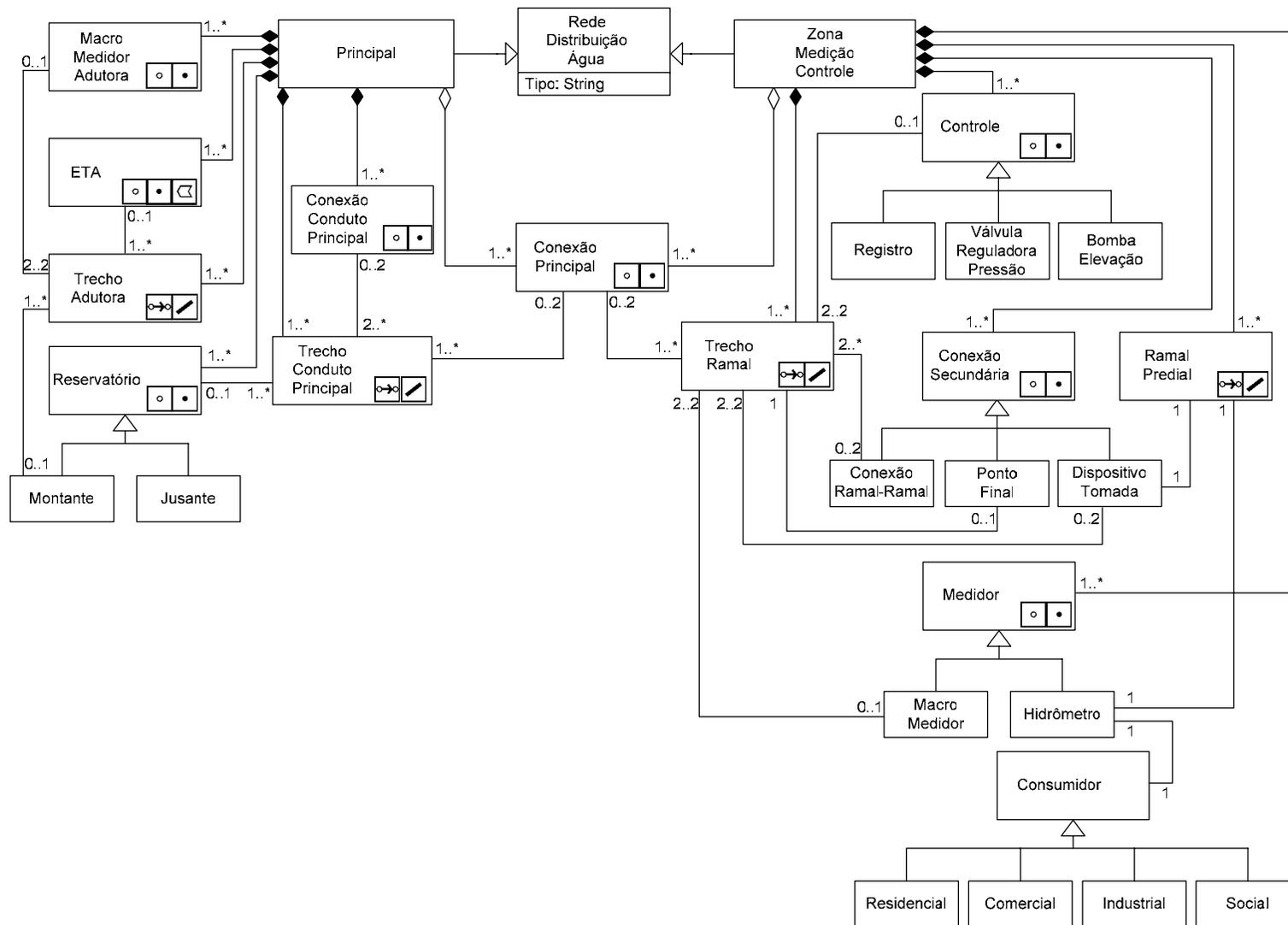


Figura 4.3 – Diagrama de classes de uma companhia de distribuição de água sem os estereótipos de generalização do UML-GeoFrame

Apesar da redução do número de estereótipos ser uma alternativa para diminuir a sobrecarga dos diagramas, existe uma dificuldade na rápida distinção entre os tipos das classes quando classes convencionais são utilizadas. Através dos estereótipos de generalização, uma classe poderá ser representada sem nenhum estereótipo somente se for uma subclasse. Ao serem omitidos os estereótipos de generalização, esse tipo de classe pode facilmente ser confundido com classes convencionais, como ocorre entre as classes *Hidrômetro* e *Consumidor*. A identificação da classe *Hidrômetro* como sendo do tipo espacial só pode ser concluída após a observação de sua especialização a partir da classe *Medidor*, sendo a classe *Consumidor* identificada como convencional após não ser observada nenhuma especialização. Por este motivo, é estimulado o uso dos estereótipos de generalização.

Ainda sobre as características presentes no diagrama dessa aplicação, observa-se também que através dos construtores de agregação e composição é possível se definir a qual(is) rede(s) pertence(m) as classes do tipo nó e arco. Além disso, os relacionamentos entre arcos e nós reduzem a ambigüidade durante a interpretação do diagrama. A classe *ConexãoPrincipal*, por exemplo, ao se agregar às redes *Principal* e *ZonaMediçãoControle*, e se relacionar as classes *TrechoRamal* e *TrechoCondutoPrincipal*, conecta estas duas redes.

Destaca-se também o uso de estereótipos para representação de arcos unidirecionais, determinando que em cada trecho exista apenas uma única direção para o fluxo da água. No entanto, não é necessária a apresentação de detalhes como nós iniciais e finais ou relacionamentos recursivos para formação de seqüências. Estas são questões consideradas apenas na fase de implementação, pois o importante durante a modelagem deste tipo de aplicação é determinar o tipo da rede e os nós e arcos que a compõe.

Uma última característica que pode ser observada no diagrama é o uso da classe *Consumidor*, do tipo convencional, sem participação direta na rede, uma vez que não possui estereótipos de rede e não é composta de nós e arcos. No entanto, sua participação na rede é feita de forma indireta, através de sua associação semântica com a classe *Hidrômetro*, uma vez que este faz a medição da água fornecida a um consumidor ou grupo de consumidores. O consumidor pode ser do tipo *Residencial*, *Comercial*, *Industrial* ou *Social* (hospitais, escolas, creches, prefeituras etc.), sendo diferenciadas as taxas de tarifa básica operacional e tarifa de água por metro cúbico.

Apesar do diagrama e das restrições de integridade estabelecidas na seção 3.8.2 transmitirem de forma abstrata grande parte dos requisitos necessários ao armazenamento dos dados da aplicação, especificações formais podem ainda ser estabelecidas através do uso da OCL para que sejam evitadas ambigüidades durante sua interpretação.

A restrição (i) determina que para todo macro medidor de adutora, deve existir um trecho de adutora que conecta uma ETA a esse macro medidor, assim como um trecho para conexão entre o macro medidor e um reservatório de montante.

```
(i) context MacroMedidorAdutora
inv: self.TrechoAdutora -> exists( t | t.ETA -> size() = 1)
and
self.TrechoAdutora -> exists(t | t.Montante -> size() = 1)
```

A restrição (ii) determina que para toda zona de medição e controle deve existir pelo menos um registro. A operação *oclIsTypeOf* resulta em verdadeiro somente se o tipo do objeto for idêntico ao argumento.

```
(ii) context ZonaMediçãoControle
inv: self.Controle -> exists( c | c.oclIsTypeOf ( Registro ))
```

4.2 Comparação entre os modelos UML-GeoFrame e OMT-G

O objetivo desta seção é realizar uma comparação entre os construtores de redes dos modelos UML-GeoFrame e OMT-G, apresentando as semelhanças, vantagens e desvantagens encontradas, mesmo reconhecendo que algumas características são subjetivas à percepção do usuário.

A Figura 4.4 apresenta um diagrama de classes para o sistema de distribuição de água elaborado com os construtores do modelo OMT-G.

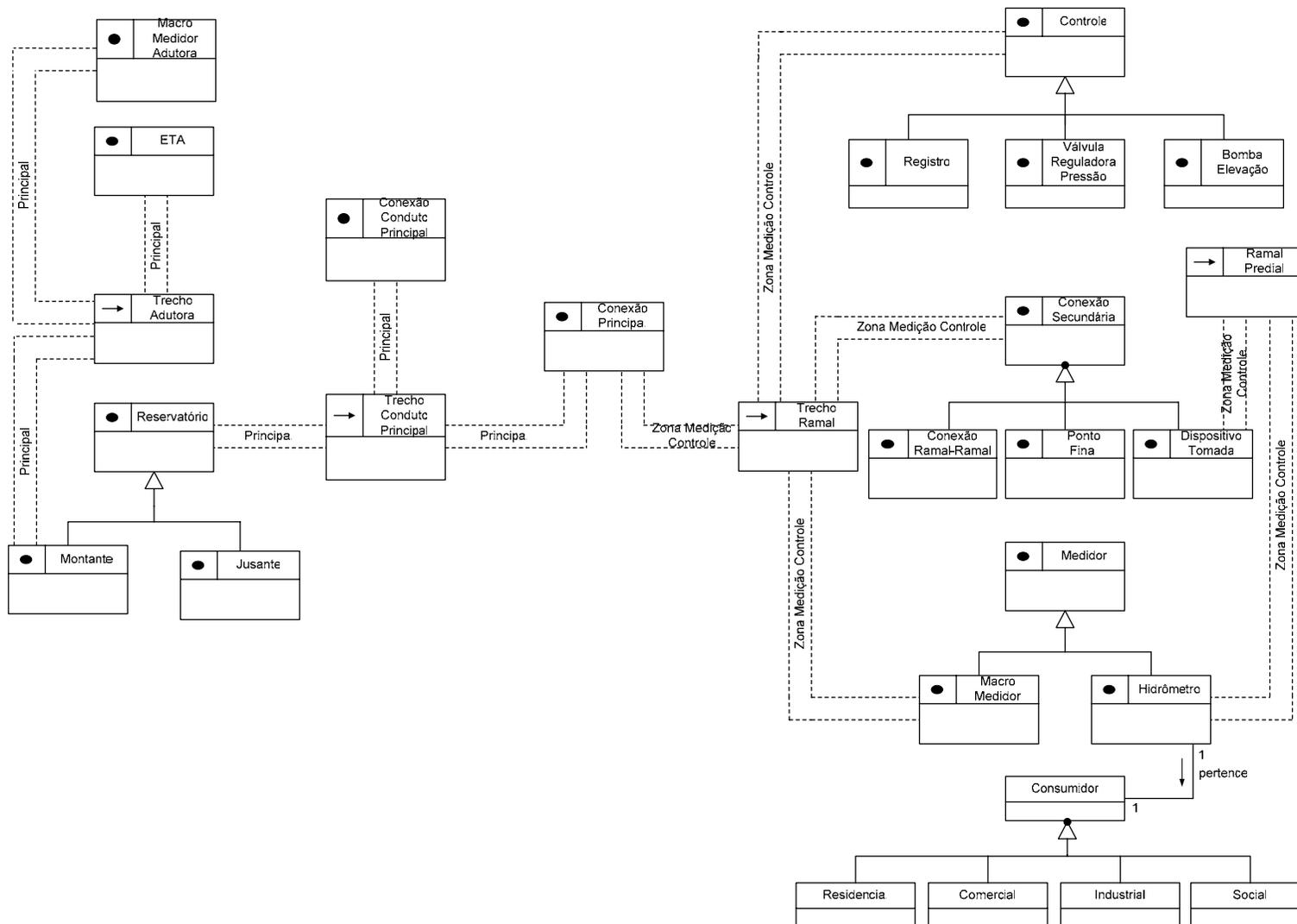


Figura 4.4 – Diagrama de classes de uma companhia de distribuição de água elaborado através dos construtores de rede do OMT-G

A solução apresentada através do modelo OMT-G utiliza os mesmos construtores de representação espacial na visão de redes do modelo UML-GeoFrame, constituindo-se de nós e arcos bidirecionais e unidirecionais, embora a representação dos estereótipos seja diferente. Quanto à identificação do tipo das classes, a abordagem utilizada pelo OMT-G assemelha-se à Figura 4.3, sendo identificadas classes convencionais, e classes georeferenciadas unicamente através da representação espacial. Assim como no UML-GeoFrame, a capacidade de associar classes convencionais e georeferenciadas tornou possível o relacionamento semântico entre as classes *Consumidor* e *Hidrômetro*. Outra semelhança entre os dois modelos é a fácil identificação dos nós de conexão entre redes.

Foram identificadas duas vantagens ao se utilizar o modelo OMT-G. A primeira delas foi a elaboração de um diagrama menos sobrecarregado, com um número muito menor de associações entre as classes para identificação dos relacionamentos de rede. A segunda foi o uso de construtores de generalização/especialização com a combinação dos tipos total/parcial e disjuncto/sobreposto. Como Lisboa Filho e Iochpe (2008) propõem o uso da generalização/especialização, mas não fazem a classificação desses tipos, assume-se que no OMT-G este conceito é utilizado com maior precisão. Porém, a opção de não incluir estas restrições no UML-GeoFrame visa manter a compatibilidade com a especificação padrão da UML.

O modelo OMT-G, apesar de possibilitar a identificação de redes, não torna possível o uso de atributos para suas descrições, uma vez que as redes não são classes no diagrama. Outra característica não encontrada no modelo OMT-G foi a capacidade de relacionar elementos de uma rede com a representação de objetos espaciais, característica presente no modelo UML-GeoFrame através da herança múltipla. Dessa forma, no OMT-G um fenômeno geográfico pode desempenhar o papel de nó ou arco em uma ou mais redes, mas não é possível ao mesmo tempo determinar sua representação como objeto geográfico.

Apesar do modelo OMT-G permitir o relacionamento direto entre nós e arcos, ele não permite expressar a cardinalidade desse tipo de relacionamento. Com isso, comparado ao diagrama elaborado com o modelo UML-GeoFrame, a classe *TrechoRamal* se relaciona de forma diferente com as conexões secundárias, associando-se diretamente com a classe *ConexãoSecundária*. Apesar de simplificar o

diagrama, essa característica na verdade diminui a precisão com que os requisitos da aplicação são representados no esquema conceitual.

Restrições de integridade representam um tópico tratado em detalhes por Davis, Borges e Laender (2002). As classes que compõem o modelo, relacionamentos do tipo todo-parte e expressões textuais pré-definidas utilizadas nos relacionamentos topológicos são especificados formalmente através de notações matemáticas. O modelo UML-GeoFrame, apesar de também possuir esses elementos de modelagem, não define formalmente suas restrições de integridade. No UML-GeoFrame, os estereótipos textuais utilizados para se determinar o tipo de restrição de integridade que se deseja impor aos elementos envolvidos em relacionamentos topológicos e o relacionamento do tipo todo-parte são os construtores desse modelo que mais se aproximam da especificação de restrições de integridade.

Baseado no modelo OMT-G, a proposta de extensão do modelo UML-GeoFrame inclui a definição formal dos relacionamentos entre os novos elementos de rede. No entanto, ao invés da notação matemática utilizada pelo OMT-G, optou-se pelo uso da OCL considerando-se sua facilidade de compreensão e o significado preciso de suas expressões. Um segundo motivo foi introduzir o uso da linguagem no modelo UML-GeoFrame para que restrições de integridade possam ser especificadas de modo formal nos casos em que o diagrama de classes não seja capaz de representar todos os requisitos da aplicação.

4.3 Estudo de caso: Sistema de distribuição de energia elétrica

O segundo estudo de caso, baseado em Kösters, Pagel e Six (1997), aborda um sistema hipotético de distribuição de energia elétrica formado por duas redes distintas, de alta e baixa voltagem. A rede de alta voltagem é responsável pela transmissão da energia elétrica desde a usina até os transformadores. A energia produzida na usina é transmitida por linhas de transmissão, cujos trechos podem ter como extremidades uma usina, uma torre ou um transformador.

O transformador pertence a ambas as redes, sendo responsável pela mudança das correntes de alta para baixa voltagem. Na rede de baixa voltagem, a energia que passa pelo transformador é transmitida por linhas de força, que podem ser dos tipos aérea ou subterrânea, passando por postes, até chegar ao consumidor. Dessa forma,

cada trecho de uma linha de força pode ter como extremidades um transformador, um poste ou um consumidor.

A Figura 4.5 ilustra os elementos da rede de distribuição de energia elétrica.

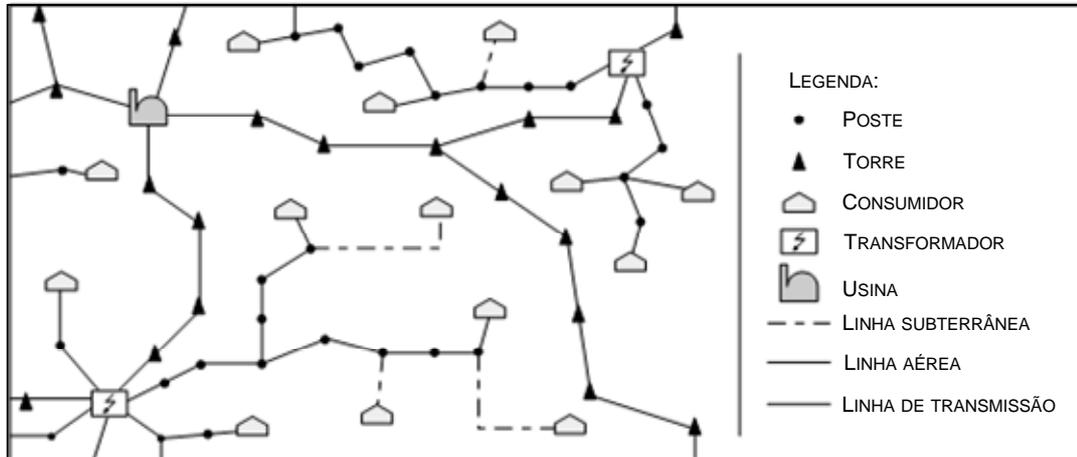


Figura 4.5 – Ilustração de um sistema de distribuição de energia elétrica
Fonte: Kösters, Pagel e Six (1997)

A Figura 4.6 apresenta o diagrama de classes para esse sistema elaborado através dos construtores do modelo UML-GeoFrame.

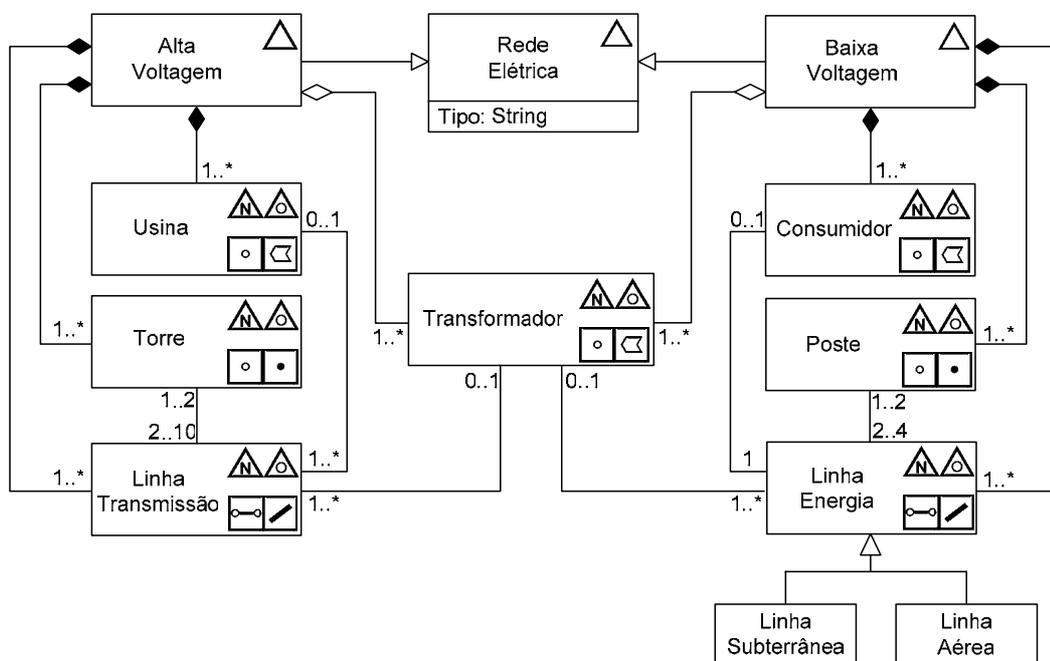


Figura 4.6 – Diagrama de classes para distribuição de energia elétrica elaborado através dos construtores de rede do UML-GeoFrame

Com exceção de sua natureza, a maior diferença entre o sistema de distribuição de água e o sistema de energia elétrica é o tipo de fluxo identificado nas redes. No primeiro caso foi necessário o uso de estereótipos de representação de arcos unidirecionais. Neste segundo, como o fluxo nas linhas de transmissão e linhas de força ocorre em ambas as direções, foi utilizado o estereótipo de arco bidirecional.

Assim como no primeiro estudo de caso, especificações formais podem ainda ser estabelecidas através do uso da OCL para que sejam evitadas ambigüidades durante a interpretação do esquema conceitual.

A restrição (i) determina que uma instância da classe *LinhaEnergia* não deve se associar a uma mesma instância da classe *Poste*. Para isso, basta determinar o valor inicial da associação entre *LinhaEnergia* e *Poste* como sendo um tipo *Set* e vazio, uma vez que este tipo não permite que elementos estejam duplicados no conjunto:

```
(i) context LinhaEnergia::Poste : Set(Poste)
    init: Set{}
```

De maneira semelhante, a restrição (ii) determina que uma instância da classe *LinhaTransmissão* não deve se associar a uma mesma instância da classe *Torre*:

```
(ii) context LinhaTransmissão::Torre : Set(Torre)
    init: Set{}
```

4.4 Comparação entre os modelos UML-GeoFrame e GeoOOA

Da mesma forma como na seção 4.2, o objetivo dessa seção é realizar uma comparação entre os construtores de redes de dois modelos. Nesse caso, entre os modelos UML-GeoFrame e GeoOOA.

A Figura 4.7 apresenta um diagrama de classes para o sistema de distribuição de energia elétrica elaborado através dos construtores do modelo GeoOOA.

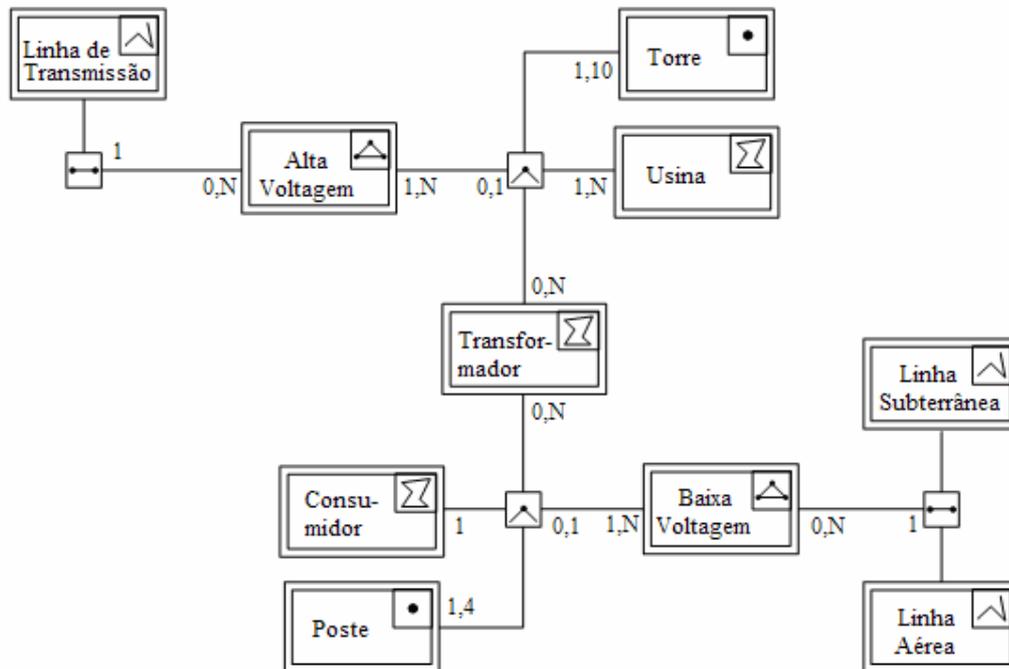


Figura 4.7 – Modelagem de companhia de energia elétrica através do GeoOOA
 Fonte: Kösters, Pagel e Six (1997)

Apesar da diferença entre os diagramas dos modelos UML-GeoFrame e GeoOOA, algumas semelhanças foram identificadas. A primeira delas refere-se à representação espacial na visão de redes, constituindo-se dos construtores de nós, arcos e redes. Em ambos os modelos, nós e arcos podem também fazer parte de mais de uma rede, como é o caso da classe *Transformador*, que faz o papel de nó nas redes de alta e baixa voltagem, estabelecendo a conexão entre estas redes.

Uma característica do modelo GeoOOA mantida no modelo UML-GeoFrame é a capacidade de relacionar elementos de rede com a representação de objetos espaciais. No GeoOOA, o papel desempenhado na rede por uma classe do tipo objeto espacial é representado através de um estereótipo de nó ou arco junto a sua associação com a classe do tipo rede. No UML-GeoFrame, o mesmo resultado é alcançado através da herança múltipla.

Outra influência foi a importância da identificação de redes como classes no esquema conceitual, deixando claro que novos atributos podem ser utilizados para identificação e descrição de uma rede, uma vez que os nós e arcos que a compõem são identificados através de associações.

Uma vantagem do modelo GeoOOA em relação ao modelo UML-GeoFrame é a capacidade de representar os relacionamentos de rede com um número menor de associações entre as classes que compõem o esquema conceitual. A associação entre nós e arcos é feita de modo a transmitir os números mínimo e máximo de arcos que podem incidir sobre um nó. A cardinalidade dos arcos não é apresentada, pois todo arco possui um nó em cada extremidade, podendo ser qualquer um desde que pertença à mesma rede.

Apesar dos diagramas desse modelo serem menos sobrecarregados que os diagramas do UML-GeoFrame, não são representadas as associações diretas entre nós e arcos, o que dificulta a representação de alguns requisitos da aplicação. Para exemplificar essa situação, supõe-se que no sistema de distribuição de energia elétrica um dos requisitos seja restringir o uso de linhas subterrâneas para conexão apenas entre um poste e um consumidor. Com isso, não seria possível representar a participação da classe *LinhaSubterrânea* na rede *BaixaVoltagem* sem que fosse possível a conexão entre dois postes ou entre um poste e um transformador através desse tipo de linha.

Uma situação em que a falta de associação direta entre nós e arcos dificultou a leitura do diagrama está representada na Figura 4.7 pelos elementos que compõem a rede de alta voltagem. Pela forma como está representada a associação entre essas classes, seria possível realizar a conexão entre duas usinas sem nenhuma torre atuando como intermediária para sustentação da linha de força, apesar de aparentemente esta ser uma situação improvável. Seria necessário um novo modo de relacionar esses elementos caso um dos requisitos da aplicação resultasse na existência de ao menos uma torre para conexão entre usinas. Situações como essas fizeram com que existissem diferenças na cardinalidade dos esquemas conceituais dos modelos UML-GeoFrame e GeoOOA.

A vantagem do modelo UML-GeoFrame em representar o relacionamento direto entre nós e arcos possibilita que o esquema conceitual seja lido e interpretado sem ambigüidades mesmo por usuários não especialistas na área de software, e ao mesmo tempo proporciona uma maior liberdade ao projetista para especificar com precisão os requisitos da aplicação.

Além disso, é importante destacar que no modelo UML-GeoFrame é possível se representar os fluxos existentes na rede através de arcos bidirecionais e

unidirecionais e que as redes são representadas por classes convencionais associadas aos tipos nó e arco através da agregação e composição.

Outra vantagem do modelo UML-GeoFrame é a preocupação para que toda informação complementar ao esquema conceitual seja expressa através de uma linguagem formal, enquanto no modelo GeoOOA esse tipo de informação é expressa através de uma linguagem natural.

Além disso, o número de informações complementares no UML-GeoFrame é muito menor devido à forma abrangente e precisa com que são tratados os requisitos da aplicação. A Tabela 4.1 apresenta o resumo da comparação entre os modelos OMT-G, GeoOOA e UML-GeoFrame com base nos diversos requisitos identificados para modelagem de redes.

Tabela 4.1 - Comparação entre os modelos OMT-G, GeoOOA e UML-GeoFrame

Requisitos	OMT-G	GeoOOA	UML-GeoFrame
Identificação dos tipos nó, arco e rede	✓	✓	✓
Identificação de arcos bidirecionais e unidirecionais	✓		✓
Identificação dos nós e arcos que compõem uma rede	✓	✓	✓
Rede representada como classe		✓	✓
Associação direta entre nós e arcos	✓		✓
Cardinalidade na associação direta entre nós e arcos			✓
Associação entre classes convencionais e georeferenciadas	✓	✓	✓
Classes abstratas para propriedades comuns às diversas classes	✓	✓	✓
Generalização/especialização combinando os tipos total/parcial e disjunto/sobreposto	✓		
Classes para conexão entre redes	✓	✓	✓
Classes com representação espacial nas visões de rede e objetos		✓	✓
Uso de linguagem formal para especificação de restrições de integridade			✓
Restrições de integridade implícitas ao modelo	✓		✓
Compatibilidade com a especificação da UML			✓
Diagramas menos sobrecarregados	✓	✓	

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho abordou a modelagem conceitual de bancos de dados geográficos para domínios em que são fundamentais as informações referentes à conectividade de seus elementos. Foram consideradas também as técnicas envolvidas na especificação de restrições de integridade em esquemas conceituais, com objetivo de melhorar a qualidade dos dados quando estes são inseridos ou modificados no banco de dados.

Como a ausência de construtores de rede no modelo UML-GeoFrame impossibilitava representar a conectividade entre os elementos da aplicação, o objetivo deste trabalho foi propor uma extensão ao modelo para que este fornecesse os construtores necessários para a elaboração de diagramas de classe envolvendo elementos de rede. Este trabalho se propôs também a incluir novos construtores que possibilitassem a especificação formal de restrições de integridade em situações nas quais seja impossível se alcançar os objetivos desejados através apenas do uso dos construtores de classes e associações provenientes do diagrama de classes da UML ou através do uso de expressões textuais pré-definidas.

O referencial teórico foi elaborado com o objetivo de fornecer uma fundamentação para compreensão do contexto em que se encontra este trabalho, os conceitos utilizados pelo UML-GeoFrame e a extensão proposta. Os tópicos abordados foram: tipos de representação de dados espaciais utilizados durante a interpretação dos fenômenos geográficos; descrição sobre modelagem conceitual e modelos conceituais para dados geográficos; taxonomias de restrições de integridade espaciais; resumo sobre a linguagem de restrições de objetos (OCL); extensão à OCL através da inserção de operações para se especificar relacionamentos topológicos entre polígonos.

A extensão do modelo UML-GeoFrame foi realizada de forma a fornecer construtores específicos para modelagem de elementos de rede mostrando como a OCL pode ser empregada como linguagem de especificação formal de restrições de integridade neste domínio. A proposta foi organizada apresentando-se as diversas etapas da evolução do GeoFrame, apontando as vantagens e desvantagens encontradas em suas versões preliminares. Com as novas classes adicionadas ao *framework*, foram incluídos também novos estereótipos. Estes seguem os mesmos

princípios do UML-GeoFrame, com um estereótipo definido para a especialização do fenômeno geográfico e outro para a representação espacial.

Ao final, foram detalhados os relacionamentos em redes bidirecionais e unidirecionais considerando-se as estruturas de dados das classes envolvidas e apresentadas restrições de integridade propostas para a modelagem de redes. Este tipo de restrição deve ser implícito aos diagramas e considerado somente na fase de implementação.

A viabilidade da proposta foi evidenciada através de dois estudos de caso, com a elaboração de esquemas conceituais para os sistemas de distribuição de água e energia elétrica, sendo possível constatar que os construtores de rede adicionados ao modelo UML-GeoFrame atendem ao conjunto de requisitos identificados por Kösters, Pagel e Six (1997) acrescidos de outros definidos nesta dissertação.

O primeiro estudo de caso foi acompanhado de uma comparação entre os construtores de rede dos modelos UML-GeoFrame e OMT-G, e no segundo estudo de caso a comparação foi realizada entre os modelos UML-GeoFrame e GeoOOA. O objetivo foi apresentar as semelhanças entre esses modelos, assim como as vantagens e desvantagens identificadas. Não foi realizada uma comparação com o modelo MADS devido à ausência de construtores de rede nesse modelo.

Durante esse processo de comparação foi possível se constatar que o modelo UML-GeoFrame apresenta importantes vantagens e algumas desvantagens em ambas as comparações. A principal desvantagem encontrada foi a elaboração de esquemas conceituais com um número maior de associações entre as classes para identificação dos relacionamentos de rede. Para exemplos pequenos, como os apresentados, esta situação é aceitável, mas em situações reais um diagrama sobrecarregado não permite uma rápida compreensão em alto nível de como o sistema funciona e esconde informações fundamentais como a simples identificação dos pontos de conexão entre redes.

No entanto, apesar dessa desvantagem, pode-se constatar através da Tabela 4.1, que o modelo UML-GeoFrame atende todos os outros requisitos identificados, com exceção da generalização/especialização combinando os tipos total/parcial e disjunto/sobreposto. Apesar de não atender esse requisito, o modelo UML-GeoFrame é o único totalmente compatível com a especificação do metamodelo da Linguagem de Modelagem Unificada (UML).

Através da identificação desses requisitos e do esforço realizado para atendê-los com a extensão do modelo UML-GeoFrame, pode-se concluir que os objetivos apresentados no início deste trabalho foram alcançados. É possível, então, a elaboração de diagramas de classe envolvendo elementos de rede, assim como o uso de estereótipos e de uma linguagem formal para especificação de restrições de integridade, mantendo os princípios do UML-GeoFrame e sua compatibilidade com a especificação do metamodelo da UML.

5.1 Contribuições deste trabalho

Este trabalho contribuiu para evolução do modelo UML-GeoFrame através da inclusão de novos construtores de rede fundamentados na teoria de grafos. As novas classes do modelo adicionaram uma nova visão sobre os dados geográficos, que a partir deste trabalho podem ser interpretados, além das visões de campo e objetos já existentes, também segundo a visão de rede. A importância dessa nova visão é possibilitar que informações como formato, orientação e comprimento da linha presentes na visão de objetos sejam descartadas para que somente os componentes estruturais, as junções e conexões sejam consideradas.

Todas as classes e relacionamentos propostos estão fundamentados na estrutura hierárquica de classes do modelo UML-GeoFrame, sendo apresentados também os estereótipos de generalização e representação espacial para os tipos rede, nó e arcos bidirecionais e unidirecionais. Além disso, os construtores incorporados ao modelo não o tornaram incompatível com a especificação do metamodelo da UML.

Outra contribuição deste trabalho foi possibilitar que classes do diagrama sejam interpretadas ao mesmo tempo nas visões de rede e objetos, formando uma rede espacialmente embutida. Dessa forma, uma classe com representação espacial na visão de objetos pode exercer o papel de nó ou arco em uma rede.

Esse relacionamento entre os elementos de uma rede e a representação de objetos espaciais utiliza o conceito de herança múltipla, com uma classe do diagrama sendo uma especialização das classes *ObjetoRede* e *ObjetoGeográfico*, e se associando as possíveis representações espaciais de ambas as visões. A substituição dessas especializações e associações foi realizada através do uso dos estereótipos de generalização e representação espacial.

Este trabalho também contribuiu através da proposta de uso da linguagem de restrições de objetos (OCL) para complementar a especificação dos diagramas elaborados através do modelo UML-GeoFrame. A OCL foi proposta devido às suas expressões serem curtas, precisas e compatíveis com a especificação do metamodelo da UML, além da possibilidade de extensão desta linguagem através de novas operações para modelagem de bancos de dados geográficos. Como este trabalho se concentrou na modelagem de redes e não houve a necessidade de estender a OCL, pode-se concluir que a proposta consiste na introdução desta linguagem junto ao modelo UML-GeoFrame.

5.2 Trabalhos futuros

O primeiro desafio identificado consiste em reduzir o número de associações para representação dos relacionamentos de rede. O foco inicial é diminuir o número de agregações e composições utilizadas para identificar a quais redes pertencem as classes do tipo nó e arco. Se considerado o primeiro estudo de caso, existem treze associações desse tipo e no segundo estudo de caso existem oito. O requisito do novo mecanismo para redução desse tipo de associação é trazer esse número a um nível aceitável, sem prejudicar as demais características do modelo. Em outras palavras, é facilitar a leitura e compreensão do esquema conceitual reduzindo o mínimo possível sua abrangência e precisão.

O segundo trabalho futuro tem como objetivo a continuidade da proposta de uso da OCL junto ao UML-GeoFrame. Uma vez que o trabalho apresentado somente iniciou esse processo e não houve a necessidade de estender essa linguagem para especificação de expressões no domínio de redes, sua continuidade nas visões de campos e objetos é importante e exigirá um trabalho de inclusão de novas operações.

Por fim, após estudar os principais modelos conceituais de dados específicos para aplicações de SIG, e de ter comparado o poder de expressão de seus construtores, percebe-se que, apesar de algumas vantagens de um modelo sobre outro, existe uma grande semelhança entre eles. No entanto, após mais de uma década de pesquisas, nenhum dos modelos alcançou posição de grande destaque em relação aos outros. Desta forma, um trabalho de grande importância para esta área é a definição formal, em consenso, de um perfil UML que aproveite o que cada modelo apresenta de melhor.

REFERÊNCIAS BIBLIOGRÁFICAS

BARROS, R. T. V. et al. **Manual de saneamento e proteção ambiental para os municípios**, v. 2, 3. reimpr. Belo Horizonte: Escola de Engenharia da UFMG, 1995.

BORGES, K. A. V.; DAVIS Jr., C. A.; LAENDER, A. H. F. OMT-G: An Object-Oriented Data Model for Geographic Applications. **GeoInformatica**, [S.L.], v.5, n.3, p. 221-260, set. 2001.

BORGES, K. A. V.; DAVIS Jr., C. A.; LAENDER, A. H. F. Modelagem Conceitual de Dados Geográficos. In: CASANOVA, M. A.; CÂMARA, G.; DAVIS Jr., C. A.; VINHAS, L.; QUEIROZ, G. R. (Org.). **Bancos de Dados Geográficos**. Curitiba: EspaçoGeo, 2005. cap. 3, p. 93-146.

CLARAMUNT, C. Extending Ladkin's Algebra on Non-convex Intervals towards an Algebra on Union-of Regions. In: Symposium on Advances in Geographic Information Systems, 8, 2000, Washington. **Proceedings...** New York: ACM, 2000. 198 p. 9-14.

CLEMENTINI, E.; Di FELICE, P.; OOSTEROM, P. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In: International Symposium on Advances in Spatial Databases, 3, 1993, Singapore. **Proceedings...** London: Springer-Verlag, 1993. 530 p. 277-295.

COAD, P.; YOURDON, E. **Object-Oriented Analysis**. 2nd ed. Englewoods Cliff: Prentice-Hall, 1991.

COCKCROFT, S. A taxonomy of spatial data integrity constraints. **GeoInformática**, [S.L.], v. 1, n. 4, p. 327-343, dec. 1997.

DAVIS Jr., C. A.; BORGES, K. A. V.; LAENDER, A. H. F. Integrity Constraints in Spatial Databases. In: DOORN, J. H.; RIVERO, L. C. (Org.). **Database Integrity: Challenges and Solutions**. Hershey: Idea Group, 2002. cap. 5, p. 144-171.

DEMIREL, H. **An Integrated Approach to the Conceptual Data Modeling of an Entire Highway Agency Geographic Information System (GIS)**. Berlin: Library of Berlin Technical University, 2002.

DEMUTH, B.; HUSSMANN, H. Using UML/OCL Constraints for Relational Database Design. In: International Conference on the Unified Modeling Language, 2, 1999, Fort Collins. **Proceedings...** Berlin: Springer, 1999. 754 p. 598-613.

DEMUTH, B.; HUSSMANN, H.; LOECHER, S. OCL as a Specification Language for Business Rules in Database Applications. In: International Conference on the Unified Modeling Language, 4, 2001, Toronto. **Proceedings...** Berlin: Springer, 2001. 507 p. 104-117.

DUBOISSET, M.; PINET, F.; KANG, M.; SCHNEIDER, M. Precise Modeling and Verification of Topological Integrity Constraints in Spatial Databases: From an Expressive Power Study to Code Generation Principles. In: International Conference on Conceptual Modeling, 24, 2005, Klagenfurt. **Proceedings...** Berlin: Springer, 2005. 496 p. 465-482.

EGENHOFER, M. J.; FRANZOSA, R. D. Point-set topological spatial relations. **International Journal of Geographic Information Systems**, v. 5, n. 2, p. 161-174, 1991.

ELMASRI, R.; NAVATHE, S. B. **Fundamentos de Bancos de Dados**. 4. ed. São Paulo: Addison Wesley, 2005.

FRIIS-CHRISTENSEN, A.; TRYFONA, N.; JENSEN, C. S. Requirements and Research Issues in Geographic Data Modeling. In: International Symposium on Advances in Geographic Information Systems, 9, 2001, Atlanta. **Proceedings...** New York: ACM, 2001. 176 p. 2-8.

KÖSTERS, G; PAGEL, B.; SIX, H. GIS-Application Development with GeoOOA. **International Journal of Geographical Information Science**, v. 11, n. 4, p. 307-335, 1997.

LAURINI, R.; THOMPSON, D. **Fundamentals of Spatial Information Systems**. San Diego: Academic Press, 1992.

LIN, X.; ZHANG, Y.; LIU, Y.; GAO, Y. Spatial data integrity ensuring mechanism in SDBMS. In: International Geoscience and Remote Sensing Symposium, 8, 2005, Seoul. **Proceedings...** Piscataway: IEEE, 2005. 684 p. 653-656.

LISBOA FILHO, J.; IOCHPE, C. Modeling with a UML Profile. In: SHEKHAR, S.; XIONG, H. (Eds.). **Encyclopedia of GIS**. Germany: Springer-Verlag, 2008. p.691-700.

LISBOA FILHO, J.; IOCHPE, C. Specifying analysis patterns for geographic databases on the basis of a conceptual framework. In: International Symposium on Advances in Geographic Information Systems, 7, 1999, Kansas City. **Proceedings...** New York: ACM, 1999. 166 p. 7-13.

LOUWSMA, J.; ZLATANOVA, S.; LAMMEREN, R.; OOSTEROM, P. Specifying and Implementing Constraints in GIS - with Examples from a Geo-Virtual Reality System. **GeoInformatica**, [S.L.], v.10, n. 4, p. 531-550, dez. 2006.

MARBLE, D. F. The extended data dictionary: A critical element in building viable spatial databases. In: Annual ESRI user conference, 11, 1991, Palm Springs. **Proceedings...** [S.L.]: ESRI, p. 245-261.

OMG – Object Management Group. **Object Constraint Language**. Needham: 2006. Disponível em: < <http://www.omg.org/docs/formal/06-05-01.pdf> >. Acesso em: 28 abr. 2008.

OMG – Object Management Group. **Unified Modeling Language**. Needham: 2007. Disponível em: < <http://www.omg.org/spec/UML/2.1.2/> >. Acesso em: 28 abr. 2008.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. **Spatial Databases With Application to GIS**. Los Altos: Morgan Kaufmann, 2001.

RUMBAUGH, J. R.; BLAHA, M. R.; PREMERLANI, W.; EDDY, F.; LORENSEN, W. **Object-Oriented Modeling and Design**. Upper Saddle River: Prentice-Hall, 1991.

SAAE - Serviço Autônomo de Água e Esgoto. Viçosa: 2008. Disponível em: < <http://www.saaevicosa.com.br> >. Acesso em: 10 maio 2008

SHEKHAR, S.; CHAWLA, S. **Spatial Databases: A Tour**. Upper Saddle River: Prentice Hall, 2003.

SPACCAPIETRA, S.; PARENT, C.; ZIMÁNYI, E. The MADS Data Model - Concepts to Understand the Structure of your Spatial and temporal Data. **International Workshop on Informative Modeling for the Architectural Heritage**, França, jul. 2006.

TRYFONA, N.; JENSEN, C. S. Conceptual Data Modeling for Spatiotemporal Applications. **GeoInformatica**, [S.L.], v. 3, n. 3, p. 245-268, set. 1999.

WARMER, J.; KLEPPE, A. **The Object Constraint Language: Getting Your Models Ready for MDA**. 2. ed. Boston: Addison Wesley, 2003.

WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. Rio de Janeiro: Elsevier, 2004.

WORBOYS, M. **GIS: A Computing Perspective**. London: Taylor and Francis, 1995.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)