

THIAGO BICALHO FERREIRA

**O USO DO PERFIL UML GEOPROFILE COMO BASE PARA A
INTEROPERABILIDADE ENTRE MODELOS CONCEITUAIS DE
BANCO DE DADOS GEOGRÁFICOS**

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
2016

Ficha catalográfica preparada pela Biblioteca Central da Universidade
Federal de Viçosa - Câmpus Viçosa

T

F383u
2016
Ferreira, Tiago Bicalho, 1990-
O uso do perfil UML Geoprofile como base para a
interoperabilidade entre modelos conceituais de banco de dados
geográficos / Tiago Bicalho Ferreira. – Viçosa, MG, 2016.
xiii, 120f. : il. ; 29 cm.

Inclui apêndices.

Orientador: Jugurta Lisboa Filho.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.85-87.

1. Sistemas de informação geográfica - Banco de dados.
2. UML (Computação). I. Universidade Federal de Viçosa.
Departamento de Informática. Programa de Pós-graduação em
Ciência da Computação. II. Título.

CDD 22. ed. 005.74

THIAGO BICALHO FERREIRA

**O USO DO PERFIL UML GEOPROFILE COMO BASE PARA A
INTEROPERABILIDADE ENTRE MODELOS CONCEITUAIS DE
BANCO DE DADOS GEOGRÁFICOS**

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

APROVADA: .

Clodoveu Augusto Davis Junior

Sabrina de Azevedo Silveira

Jurgurta Lisboa Filho
(Orientador)

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter permitido a realização de mais um sonho. Aos meus familiares, José Ferreira dos Santos, Maria Aparecida Bicalho Ferreira e Gabriel Bicalho Ferreira pelas orações e interseções. A minha esposa Lídia Ferreira agradeço pela atenção, compreensão, companheirismo e orações feitas a Jesus Cristo em meu favor. Estas foram as pessoas que mesmo sem conhecer os desafios a serem enfrentados por mim, me deram força na caminhada através de palavras revigorantes e pedidos a Deus.

No âmbito acadêmico, agradeço ao meu orientador, Prof. Jugurta Lisboa Filho, pelo apoio e pelos momentos de aprendizado proporcionados no decorrer do mestrado. A todos os colegas e professores do mestrado, pela ajuda, companheirismo e momentos divididos nos anos de trabalho. Aos amigos Sérgio Stempluc, Ítalo Lopes Oliveira, Marcos Vinicius Montanari e Felipe Nalon pelas colaborações neste trabalho.

A todas as pessoas que contribuíram para que hoje este trabalho se tornasse realidade, seja nas aulas, em orações, palavras motivadoras ou contribuições de qualquer outro tipo, a estes dedico este trabalho e agradeço por tudo.

Muito obrigado!

SUMÁRIO

LISTA DE FIGURAS	vi
LISTA DE TABELAS	x
LISTA DE SIGLAS	xi
RESUMO.....	xii
ABSTRACT.....	xiii
1 INTRODUÇÃO.....	1
1.1 O Problema e sua Importância	2
1.2 Hipótese.....	4
1.3 Objetivos	4
1.4 Estrutura da Dissertação	5
2 REFERENCIAL TEÓRICO	6
2.1 Modelos Conceituais e Requisitos Espaciais	6
2.1.1 Modelo Conceitual GeoOOA.....	6
2.1.2 Modelo Conceitual OMT-G	8
2.1.3 Modelo Conceitual UML-GeoFrame	9
2.1.4 Modelo Conceitual PVL.....	11
2.2 Uso de Perfil UML para modelagem de dados geográficos.....	12
2.2.1 Perfil UML GeoProfile.....	13
2.3 Integração da abordagem MDA com o GeoProfile	19
2.4 Interoperabilidade de esquemas de dados	20
2.5 Uso de ferramentas de transformação para interoperabilidade de esquemas	21
2.5.1 Ferramenta CASE Enterprise Architect	22
3 MÉTODO PARA INTEROPERABILIDADE DE ESQUEMAS DE DADOS GEOGRÁFICOS	24
3.1 Etapa 1: Seleção de Modelos e Requisitos	26
3.2 Etapa 2: Especificação de Metamodelos	26

3.2.1	Especificação de metamodelos para representação de tipos básicos de objetos espaciais	26
3.2.2	Especificação de metamodelos para representação de fenômenos na visão de campos contínuos.....	29
3.2.3	Especificação de metamodelos para representação de redes.....	30
3.3	Etapa 3: Definição do Perfil UML e Restrições OCL.....	32
3.3.1	Perfil GeoOOA.....	32
3.3.2	Perfil OMT-G.....	34
3.3.3	Perfil UML-GeoFrame	36
3.3.4	Perfil PVL.....	38
3.4	Etapa 4: Aplicação e Configuração da Tecnologia MDG aos perfis UML.....	39
3.5	Etapa 5: Especificação de Regras de Transformação Horizontal.....	39
3.5.1	Mapeamento entre MDG_GeoOOA e MDG_GeoProfile.....	40
3.5.2	Mapeamento entre MDG_OMT-G e MDG_GeoProfile	42
3.5.3	Mapeamento entre MDG_UML-GeoFrame e MDG_GeoProfile	45
3.5.4	Mapeamento entre MDG_PVL e MDG_GeoProfile.....	47
3.6	Etapa 6: Proposta de extensão do GeoProfile	48
3.6.1	Mapeamento entre MDG_PVL e MDG-GeoProfile Estendido	50
3.7	Etapa 7: Especificação de Regras de Transformações Verticais	51
3.7.1	Mapeamento vertical MDG_GeoOOA	51
3.7.2	Especificação do mapeamento vertical MDG_OMTG	52
3.7.3	Especificação do mapeamento vertical MDG_UML-GeoFrame.....	55
3.7.4	Especificação do mapeamento vertical MDG_PVL	58
4	EXPERIMENTAÇÃO DO MÉTODO DE INTEROPERABILIDADE.....	61
4.1	Esquema 1: Mineração de Carvão.....	61
4.1.1	Esquema mineração de Carvão mapeado para Tecnologia MDG_GeoOOA ...	64
4.1.2	Esquema mineração de Carvão mapeado para Tecnologia MDG_OMTG.....	66

4.1.3 Mineração de Carvão mapeado para Tecnologia MDG_PVL	68
4.2 Esquema 2: Meio Ambiente	70
4.2.1 Esquema Meio Ambiente mapeado para Tecnologia MDG_OMTG.....	72
4.3 Esquema 3: Rede de Distribuição de Energia	73
4.3.1 Esquema Rede de Distribuição de Energia mapeado para Tecnologia MDG_UML-GeoFrame	77
4.3.2 Esquema Rede de Distribuição de Energia mapeado para Tecnologia MDG_OMTG	80
5 CONCLUSÕES E TRABALHOS FUTUROS	83
REFERÊNCIAS BIBLIOGRÁFICAS	85
APÊNDICE A- Customização da Ferramenta Enterprise Architect	88
APÊNDICE B- scripts que devem ser colocados para alterar o formato da classe.....	101
APÊNDICE C- templates modificados para transformação horizontal	106
APÊNDICE D- templates modificados para interoperabilidade vertical	112

LISTA DE FIGURAS

Figura 1.1: Interoperabilidade horizontal de esquema conceitual entre modelos de BDG que compartilham o requisito representação dos tipos básicos de objetos espaciais	3
Figura 1.2: Interoperabilidade horizontal de esquemas conceituais de BDG usando o GeoProfile.....	4
Figura 2.1: Ícones do GeoOOA para representação de fenômenos na visão de objetos e rede. .	6
Figura 2.2: Exemplo Distribuição de Energia.. ..	8
Figura 2.3: Ícones do OMT-G para representação de fenômenos na visão de campo, objetos e rede.	9
Figura 2.4: Exemplos de relacionamentos simples e espaciais do modelo	9
Figura 2.5: Ícones do UML-GeoFrame para representação de fenômenos na visão de campo, objetos, rede e domínios geográfico.	10
Figura 2.6: Modelagem conceitual de um sistema utilizando o modelo UML-Geoframe.	11
Figura 2.7: Pictogramas para representação de objetos geográficos no PVL	11
Figura 2.8: Esquema conceitual desenvolvido a partir dos construtores do PVL.....	12
Figura 2.9: Metamodelo do GeoProfile.	14
Figura 2.10: Estereótipos do Perfil GeoProfile.	16
Figura 2.11: Ícones Usados no GeoProfile.	17
Figura 2.12: Aplicando o GeoProfile na visão de Objetos.	18
Figura 2.13: Aplicando o GeoProfile na visão de Campo.	18
Figura 2.14: Aplicando o GeoProfile na visão de Rede.	19
Figura 2.15: Níveis de abstração MDA integrados ao Perfil GeoProfile	20
Figura 2.16: Interoperabilidade de esquemas conceituais	21
Figura 2.17: Utilização de ferramentas de transformação na abordagem MDA.....	22
Figura 2.18: Recursos que podem ser incluídos na tecnologia MDG da ferramenta EA.....	23
Figura 3.1: Metamodelo GeoOOA para representação dos tipos básicos de objetos espaciais.	27
Figura 3.2: Metamodelo OMT-G para representação dos tipos básicos de objetos espaciais..	27
Figura 3.3: Metamodelo UML-GeoFrame para representação dos tipos básicos de objetos espaciais.....	28
Figura 3.4: Metamodelo PVL representação dos tipos básicos de objetos espaciais.	29
Figura 3.5: Metamodelo OMT-G para representação de campos contínuos.	30
Figura 3.6: Metamodelo UML-GeoFrame para representação de campos contínuos.	30

Figura 3.7: Metamodelo GeoOOA para representação de redes.	31
Figura 3.8: Metamodelo OMT-G para representação de Redes.	31
Figura 3.9: Metamodelo UML_GeoFrame para representação de Redes.	32
Figura 3.10: Perfil GeoOOA	33
Figura 3.11: perfil OMT-G.....	35
Figura 3.12: Perfil UML-GeoFrame.....	37
Figura 3.13: Perfil PVL.	38
Figura 3.14: Mapeamento de Tecnologias.	40
Figura 3.15: Especificação de mapeamento da representação de objetos básicos entre MDG_GeoOOA e MDG_GeoProfile.....	41
Figura 3.16: Especificação de mapeamento da representação de rede entre MDG_GeoOOA e MDG_GeoProfile.	42
Figura 3.17: Especificação de mapeamento da representação de objetos básicos entre MDG_OMT-G e MDG_GeoProfile.	43
Figura 3.18: Especificação do mapeamento da representação de campos contínuos entre MDG_OMT-G e MDG_GeoProfile.	43
Figura 3.19: Especificação do mapeamento da representação de redes entre Tec MDG_OMT-G e Tec MDG_GeoProfile.....	44
Figura 3.20: Especificação do mapeamento de rede entre MDG_OMT-G e MDG_GeoProfile.	45
Figura 3.21: Especificação de mapeamento da representação de objetos básicos entre MDG_UML_GeoFrame e MDG_GeoProfile.	46
Figura 3.22: Especificação do mapeamento da representação de campos contínuos entre MDG_UML-GeoFrame e MDG_GeoProfile.	46
Figura 3.23: Especificação do mapeamento da representação de redes entre MDG_UML_GeoFrame e MDG_GeoProfile.	47
Figura 3.24: Especificação de mapeamento da representação de objetos básicos entre MDG_PVL e Tec MDG_GeoProfile.....	48
Figura 3.25: Extensão do metamodelo do GeoProfile.....	49
Figura 3.26: Extensão do metamodelo do GeoProfile.....	49
Figura 3.27: Especificação de mapeamento da representação de objetos básicos entre MDG_PVL e MDG_GeoProfile_estendido	50

Figura 3.28: Especificação de mapeamento vertical da representação de objetos básicos da tecnologia MDG_GeoOOA.....	51
Figura 3.29: Especificação de mapeamento vertical da representação de redes da tecnologia MDG_GeoOOA.	52
Figura 3.30: Especificação de mapeamento vertical da representação de objetos básicos da tecnologia MDG_OMTG.	53
Figura 3.31: Especificação de mapeamento vertical da representação de campos contínuos da tecnologia MDG_OMTG.	54
Figura 3.32: Especificação de mapeamento vertical da representação de redes da tecnologia MDG_OMTG.	55
Figura 3.33: Especificação de mapeamento vertical da representação de objetos básicos da tecnologia MDG_UML-GeoFrame.	56
Figura 3.34: Especificação de mapeamento vertical da representação de campos contínuos da tecnologia MDG_UML-GeoFrame.	57
Figura 3.35: Especificação de mapeamento vertical da representação redes da tecnologia MDG_UML-GeoFrame.....	58
Figura 3.36: Especificação de mapeamento vertical dos construtores point, Line e Polygon da tecnologia MDG_PVL.....	59
Figura 3.37: Especificação de mapeamento vertical dos construtores point, Line e Polygon da tecnologia MDG_PVL.....	60
Figura 4.1: Esquema conceitual Mineração de Carvão- UML-GeoFrame.....	61
Figura 4.2: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_UML-GeoFrame (PIM) mapeado para o nível PSM.....	62
Figura 4.3: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS.....	63
Figura 4.4: Esquema conceitual Mineração de Carvão ilustrado a partir dos construtores da Tecnologia MDG_GeoProfile.	64
Figura 4.5: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_GeoOOA (PIM) mapeado para o nível PSM.	65
Figura 4.6: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS.....	66
Figura 4.7: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_OMTG (PIM) mapeado para o nível PSM.	67

Figura 4.8: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS.....	68
Figura 4.9: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_PVL (PIM) mapeado para o nível PSM.	69
Figura 4.10: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS.....	70
Figura 4.11: Esquema conceitual Meio ambiente	71
Figura 4.12: Esquema conceitual Meio Ambiente mostrado a partir da Tecnologia MDG_UML-GeoFrame (PIM) mapeado para o nível PSM.....	71
Figura 4.13: Código DDL para criação do banco de dados Uso do Solo no SGBD PostGIS.	72
Figura 4.14: Esquema conceitual Meio Ambiente ilustrado a partir dos construtores da Tecnologia MDG_GeoProfile.	72
Figura 4.15: Esquema conceitual Uso do Solo mostrado a partir da Tecnologia MDG_UML-OTG (PIM) mapeado para o nível PSM.....	73
Figura 4.16: Código DDL para criação do banco de dados Uso do Solo no SGBD PostGIS..	73
Figura 4.17: Esquema conceitual Distribuição de Energia- GeoOOA.....	74
Figura 4.18: Esquema conceitual Distribuição de Energia mostrado a partir da Tecnologia MDG_GeoOOA (PIM).....	74
Figura 4.19. Esquema PSM rede distribuição de energia.....	75
Figura 4.20: Código DDL para criação do banco de dados de uma rede de distribuição de energia no SGBD PostGIS.	76
Figura 4.21: Esquema Distribuição de Energia ilustrado a partir dos construtores da Tecnologia MDG_GeoProfile.	77
Figura 4.22: Esquema Distribuição de Energia ilustrado a partir dos construtores da Tecnologia MDG_UML-GeoFrame.....	77
Figura 4.23. Esquema PSM Distribuição de Energia- UML-GeoFrame.....	78
Figura 4.24: Código DDL para criação do banco de dados Distribuição de Energia no SGBD PostGIS.....	79
Figura 4.25: Esquema Distribuição de Energia ilustrado a partir dos construtores da Tecnologia MDG_OMTG.	80
Figura 4.26: Esquema PSM Distribuição de Energia (OMTG).....	81
Figura 4.27: Código DDL para criação do banco de dados Distribuição de Energia no SGBD PostGIS.....	82

LISTA DE TABELAS

Tabela 2.1: Requisitos atendidos por modelos conceituais de BDG	7
Tabela 3.1: Relação entre modelos conceituais e requisitos selecionados.....	26
Tabela 3.2: Restrições contra o uso de mais de um estereótipo por classe no perfil GeoOOA.	33
Tabela 3.3: Restrições para evitar inconsistências na modelagem de redes do perfil GeOOA	34
Tabela 3.4: Restrições contra o uso de mais de um estereótipo por classe no perfil OMT-G	35
Tabela 3.5: Restrições para evitar inconsistências na modelagem de redes no perfil OMT-G	36
Tabela 3.6: Restrições contra o uso estereótipos incompatíveis na mesma classe do perfil UML_GeoFrame.	37
Tabela 3.7: Restrições para evitar inconsistências em redes do perfil UML_GeoFrame.....	38
Tabela 3.8: Restrições contra o uso estereótipos incompatíveis na mesma classe do GeoProfile estendido.....	50

LISTA DE SIGLAS

BDG - Banco de Dados Geográficos
CASE - Computer-aided software engineering
CIM - Computation Independent Model
DDL - Data Definition Language
EA – Enterprise Architect
ER – Entidade Relacionamento
MDA - Model Driven Architecture
MDG - Model Driven Generation
OCL - Object Constraint Language
OGC - Open Geospatial Consortium
OMG - Object Management Group
OMT - Object Modeling Technique
OOA - Object-Oriented Analysis
PIM - Platform Independent Model
PSM - Platform Specific Model
SGBD – Sistema Gerenciador de Banco de Dados
UML - Unified Modeling Language
XML - eXtensible Markup Language

RESUMO

FERREIRA, Thiago Bicalho. M. Sc., Universidade Federal de Viçosa, maio de 2016. **O uso do perfil UML Geoprofile como base para a interoperabilidade entre modelos conceituais de banco de dados geográficos.** Orientador: Jugurta Lisboa Filho.

As pesquisas na área de modelagem conceitual de banco de dados geográficos produziram diversos modelos conceituais e muitas ferramentas Computer-aided software engineering (CASE) específicas. Uma consequência de não se ter um modelo comum é a falta de interoperabilidade entre os esquemas de banco de dados geográficos. A interoperabilidade pode acontecer de forma vertical e horizontal. A interoperabilidade vertical diz respeito ao nível de abstração Model Driven Architecture (MDA) que um esquema pode assumir. Assim, um esquema conceitual Platform Independent Model (PIM) é transformado em um esquema dependente de plataforma Platform Specific Model (PSM) que é então transformado em código fonte. A interoperabilidade horizontal está ligada a possibilidade de um esquema de dados conceitual poder ser visualizado com construtores de diferentes modelos conceituais. A interoperabilidade de esquemas é importante, pois facilita a comunicação entre projetistas de diferentes equipes, possibilita a integração de diferentes projetos e o reuso de soluções já testadas e validadas. Para mostrar a interoperabilidade horizontal e vertical de esquemas é necessário utilizar um ambiente onde as características dos construtores do modelo possam ser replicadas. Características como restrições, ícones gráficos e formato da classe devem ser mantidas. O ambiente também deve oferecer recursos para mostrar de forma automatizada a interoperabilidade horizontal e vertical de esquemas. A ferramenta Enterprise Architect, através da tecnologia Model Driven Generation oferece estes recursos. Este trabalho descreve um método que mostra ser possível alcançar interoperabilidade horizontal e vertical de esquemas de dados geográficos, utilizando o perfil UML GeoProfile como modelo conceitual intermediário. Exemplos de esquemas conceituais de dados, extraídos da literatura, são transformados horizontalmente entre diferentes modelos conceituais. Para mostrar a equivalência dos esquemas, transformações verticais são realizadas, produzindo esquemas de banco de dados equivalentes no nível de código fonte.

ABSTRACT

FERREIRA, Thiago Bicalho. M. Sc., Universidade Federal de Viçosa, May 2016. **The Use of UML Geoprofile as Basis for the Interoperability among Conceptual Geographic Database Models.** Adviser: Jugurta Lisboa Filho.

Researches on geographic database conceptual modeling have produced several conceptual models and many specific Computer-aided software engineering (CASE) tools. One consequence of there not being a common model is the lack of interoperability among geographic database schemas. Interoperability may occur vertically or horizontally. Vertical interoperability concerns the level of Model Driven Architecture (MDA) abstraction a schema may have. Thus, a PIM (Platform-Independent Model) conceptual schema is transformed into a dependent PSM (Platform-Specific Model) schema, which is then transformed into source code. Horizontal interoperability is linked to the possibility of a conceptual data schema being visualized with constructs of different conceptual models. Schema interoperability is important since it facilitates the communication among designers from different teams and enables integration of different projects and the reuse of previously tested and validated solutions. In order to show the horizontal and vertical schema interoperability, one must use an environment where the characteristics of the model's constructs can be replicated. Characteristics such as restrictions, graphical icons, and class format must be kept. The environment must also provide resources to automatically show the horizontal and vertical schema interoperability. The Enterprise Architect tool, through the Model Driven Generation technology, offers such resources. This study describes a method that shows it is possible to reach horizontal and vertical interoperability of geographic data schemas using UML GeoProfile as an intermediate conceptual model. Examples of conceptual data schemas, extracted from the literature, are horizontally transformed among different conceptual models. In order to show the schemas are equivalent, vertical transformations are carried out, which produce database schemas equivalent to the source code.

1 INTRODUÇÃO

Há mais de 20 anos, pesquisadores da área de Banco de Dados Geográficos (BDG) vêm propondo novos modelos conceituais, que são extensões dos formalismos Entidade Relacionamento (ER) (CHEN, 1976) e modelos Orientados a Objetos (OO), para representar as especificidades dos dados geográficos (PINET, 2012). Diversas propostas foram apresentadas para solucionar problemas que envolviam, por exemplo, a falta de construtores espaço/temporais e a falta de ferramentas Computer-Aided Software Engineering (CASE) para auxiliar na tarefa de modelagem conceitual dos dados espaço/temporais (PINET, 2012; LISBOA FILHO et al., 2004).

Os modelos conceituais de dados geográficos são constituídos de um conjunto de construtores que auxiliam na abstração da realidade geográfica a ser modelada. Ferramentas CASE podem ser usadas para auxiliar e automatizar a elaboração dos esquemas conceituais de banco de dados. Com base em um levantamento cronológico na área de modelagem conceitual de BDG, Pinet (2012) propôs um esquema de classificação para os modelos conceituais a partir de seis requisitos espaciais e um requisito temporal. Os requisitos espaciais são: representação dos tipos básicos de objetos espaciais; especificação de relacionamentos espaciais; múltipla representação espacial; descrição de objetos com limites ou posições incertas; representação de campos contínuos; e representação de rede. O requisito temporal consiste na descrição da evolução dos objetos espaciais ao longo do tempo. A Tabela 2.1, da subseção 2.1, relaciona os modelos conceituais aos requisitos descritos em Pinet (2012). Cada requisito é representado por um conjunto de construtores espaciais ou temporais.

Devido ao grande número de modelos conceituais propostos e ferramentas CASE desenvolvidas, surgiram problemas como a falta de interoperabilidade entre os esquemas de BDG. Segundo Staub (2007), a interoperabilidade de esquemas desenvolvidos a partir de modelos conceituais pode ocorrer na direção vertical ou horizontal.

A interoperabilidade vertical acontece entre diferentes níveis de abstração da abordagem MDA (Model Driven Architecture). Na abordagem MDA, os níveis de abstração considerados são: Computation Independent Model (CIM) - um esquema é elaborado independente do ambiente computacional no qual será desenvolvido; Platform Independent Model (PIM) - o esquema assume o nível de abstração conceitual e é elaborado utilizando os construtores de um modelo conceitual; Platform Specific Model (PSM) - o esquema assume o

nível de abstração lógico, possuindo características de uma plataforma específica; e, por último, o código fonte (KLEPPE et al., 2003).

A interoperabilidade horizontal ocorre entre diferentes modelos no mesmo nível. Por exemplo, a interoperabilidade no nível PIM ocorre entre diferentes modelos conceituais (STAUB, 2007; STAUB et al., 2008). Dois modelos conceituais são interoperáveis quando cada construtor em um modelo conceitual X possui um ou mais construtores correspondentes em outro modelo conceitual Y e vice-versa. Desta forma, um esquema conceitual X_1 elaborado a partir do conjunto de construtores do modelo X pode ser transformado em um esquema conceitual de dados Y_1 e vice-versa.

1.1 O Problema e sua Importância

Diversas propostas foram apresentadas para solucionar problemas que envolviam a falta de construtores espaço/temporais, a falta de modelos conceituais para BDG e a falta de ferramentas CASE para auxiliar na tarefa de modelagem conceitual dos dados espaço/temporais (PINET, 2012; LISBOA FILHO et al., 2004). Entretanto, as propostas apresentadas acarretaram na falta de um modelo padrão na área de modelagem conceitual de BDG.

Segundo Sampaio (2009), a falta de um modelo conceitual padrão traz problemas para os projetistas de BDG e suas equipes, podendo-se citar a dificuldade de integração entre diferentes esquemas conceituais e a dificuldade de se reutilizar projetos já validados em sistemas anteriores.

Ainda não há consenso na definição de um modelo conceitual padrão para área de modelagem conceitual de BDG. Com base em Staub (2007), estes problemas podem ser classificados como problemas de interoperabilidade horizontal.

A interoperabilidade horizontal de um esquema conceitual de BDG pode acontecer entre modelos conceituais que apresentam o mesmo requisito. Por exemplo, um esquema conceitual de BDG, produzido a partir dos construtores do modelo conceitual OMT-G e cujas características referem-se apenas ao requisito de representação dos tipos básicos de objetos espaciais, pode ser convertido para os demais modelos apresentados na Tabela 2.1, com exceção do modelo conceitual Multiple Representation Schema Language.

Uma forma de mostrar a equivalência de esquemas conceituais entre modelos, é através da transformação vertical. Ao transformar os esquemas para o nível PSM, os mesmos devem ser iguais ou semelhantes.

A Figura 1.1 ilustra a interoperabilidade horizontal entre alguns modelos conceituais da Tabela 2.1. Cada aresta representa possíveis transformações horizontais, por exemplo: (OMT-G, PVL) e (OMT-G, T-Omega).

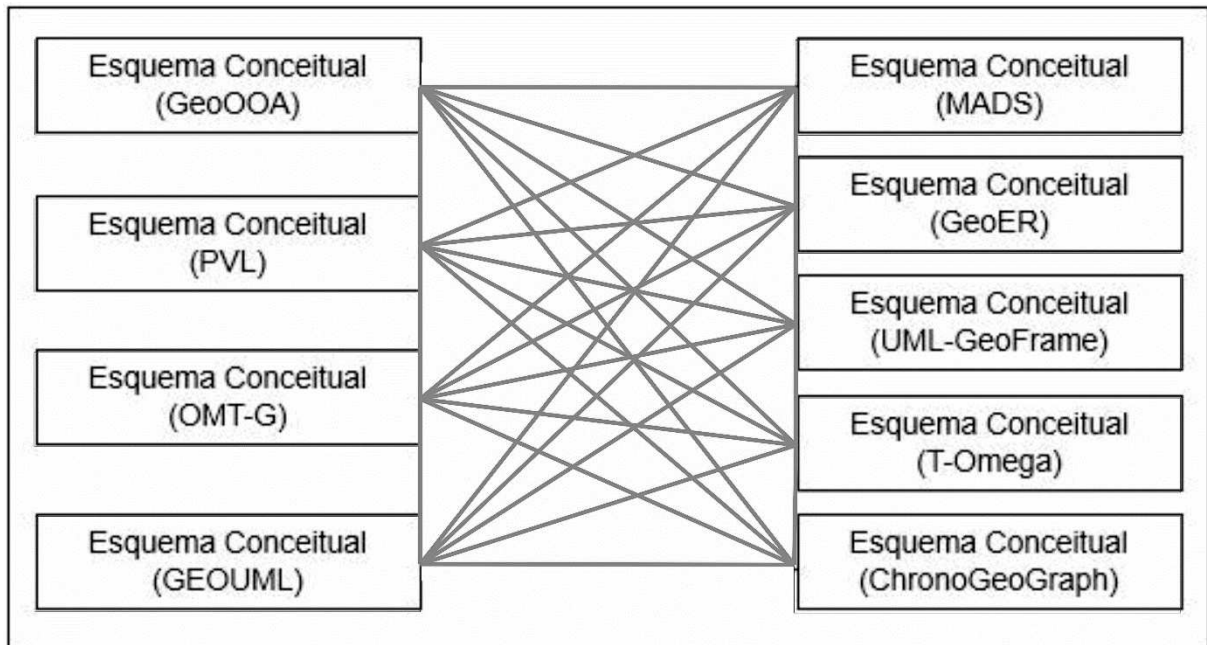


Figura 1.1: Interoperabilidade horizontal de esquema conceitual entre modelos de BDG que compartilham o requisito representação dos tipos básicos de objetos espaciais

Cada modelo conceitual de BDG possui um conjunto de construtores espaço/temporal específico. Desta forma, para mostrar a interoperabilidade horizontal de um esquema conceitual entre modelos que compartilham o mesmo requisito, é necessário desenvolver regras de transformação. Estas regras são responsáveis pelo mapeamento dos esquemas conceituais entre modelos.

Na Figura 1.1, cada modelo conceitual pode se relacionar com outros 8 modelos. Desta forma, para cada modelo devem ser desenvolvidos 8 conjuntos de regras de transformação. Logo, para mostrar a interoperabilidade de esquemas conceituais entre os modelos conceituais da Figura 1.1, devem ser desenvolvidos 72 conjuntos de regras de transformação.

Uma opção para reduzir o número de transformações horizontais é a utilização de um modelo conceitual base. Assim, o esquema conceitual de um modelo origem é transformado em um esquema conceitual do modelo base, que por seguinte é transformado em um esquema conceitual do modelo destino.

Analisando o trabalho de Pinet (2012), é possível perceber-se que nenhum modelo conceitual da Tabela 2.1 pode ser usado como base de interoperabilidade. Estes modelos não

possuem construtores espaço/temporais para todos os requisitos especificados por Pinet (2012).

O GeoProfile é um Perfil UML criado para modelagem conceitual de BDG (SAMPAIO 2009) (NALON, 2010). Por ser um perfil UML pode ser estendido e fazer uso de toda infraestrutura da Unified Modeling Language (UML), incluindo ferramentas CASE e restrições em Object Constraint Language (OCL). Uma possibilidade é utilizar o GeoProfile como base de interoperabilidade horizontal.

1.2 Hipótese

O conjunto de construtores do GeoProfile compreende o conjunto de construtores dos modelos conceituais de BDG. Portanto, o GeoProfile pode ser usado como base para alcançar interoperabilidade entre os esquemas conceituais produzidos pelos demais modelos.

A Figura 1.2 apresenta o GeoProfile como modelo base para a interoperabilidade. Ao utilizar o GeoProfile como modelo base o número de conjuntos de regras de transformação será reduzido, passando a ser necessário apenas regras de transformação entre cada modelo e o GeoProfile. Cada seta na Figura 1.2 representa um conjunto de regras de transformação.

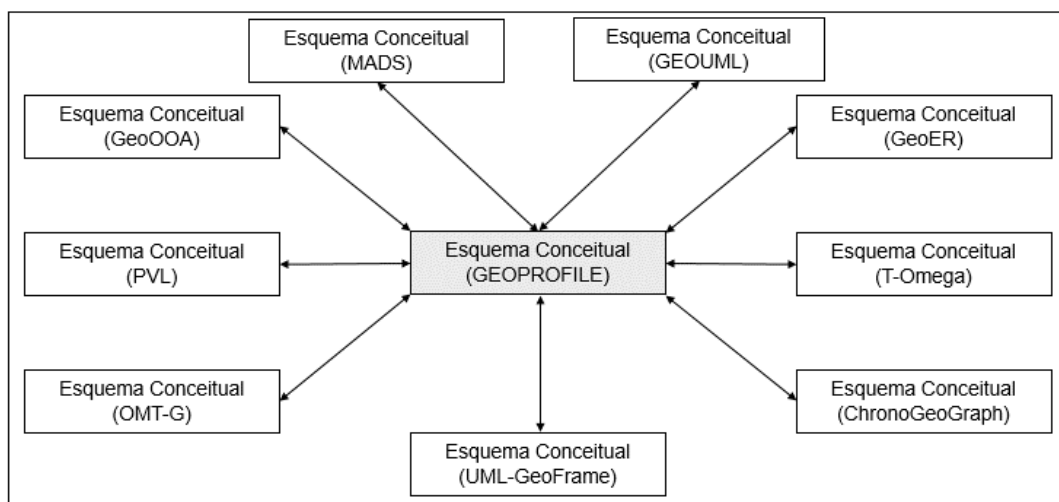


Figura 1.2: Interoperabilidade horizontal de esquemas conceituais de BDG usando o GeoProfile.

1.3 Objetivos

O objetivo deste trabalho é analisar, com base no metamodelo do Perfil GeoProfile, se é possível alcançar interoperabilidade horizontal de esquemas conceituais elaborados a partir de um conjunto de modelos conceituais existentes na literatura. Verificar se o metamodelo do Perfil GeoProfile atende aos requisitos específicos de BDG, descritos por Pinet (2012), e propor possíveis extensões ao Perfil GeoProfile para que este sirva de base para realizar a interoperabilidade desejada.

Especificamente, pretende-se:

- Selecionar um conjunto de modelos conceituais de BDG, para o desenvolvimento do trabalho;
- Utilizar a UML para especificar o metamodelo de cada modelo conceitual selecionado;
- Com base nos metamodelos desenvolvidos, especificar regras de transformação horizontal para os modelos conceituais analisados, buscando alcançar a interoperabilidade entre esquemas conceituais destes modelos;
- Especificar regras de transformação vertical para cada modelo conceitual selecionado;
- A partir de exemplos de esquemas conceituais de BDG, obtidos na literatura, mostrar se é possível alcançar a interoperabilidade horizontal (de esquemas PIM entre os metamodelos selecionados) e vertical (entre os diferentes níveis do MDA de esquemas desenvolvidos a partir de cada metamodelo selecionado).

1.4 Estrutura da Dissertação

O restante da dissertação está organizado como segue. O capítulo 2 apresenta o referencial teórico, onde são discutidos o Geoprofile, os modelos conceituais e a ferramenta CASE utilizada no trabalho.

No capítulo 3 são apresentadas as sete etapas para mostrar a interoperabilidade vertical e horizontal. Estas etapas incluem, seleção de modelos conceituais, especificação de metamodelos, especificação de perfis UML e tecnologias Model Driven Generation (MDG), extensão do perfil UML GeoProfile e mapeamentos verticais e horizontais.

No capítulo 4, com auxílio da ferramenta CASE Enterprise Architect (EA), e através de exemplos da literatura é mostrada a interoperabilidade horizontal e vertical de esquemas.

Finalmente, no capítulo 5 encontram-se as conclusões e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico da dissertação, e está organizado como segue. A seção 2.1 descreve alguns modelos conceituais de BDG. A seção 2.2 descreve a utilização de Perfis UML para modelagem conceitual de BDG, bem como alguns dos estereótipos do GeoProfile. Na seção 2.3 é discutida a integração do GeoProfile com a abordagem MDA. A seção 2.4 descreve a interoperabilidade de esquema de dados a partir da proposta de Staub (2007). E, por último, é feita uma descrição a respeito de ferramentas que auxiliam na interoperabilidade de esquemas.

2.1 Modelos Conceituais e Requisitos Espaciais

Modelos conceituais de BDG vêm sendo propostos com a finalidade de atender as especificações de modelagem conceitual em aplicações com informação geográfica. Estes permitem a abstração de fenômenos geográficos de acordo com suas dimensões temporais e espaciais. Em Pinet (2012) foi feito um levantamento cronológico dos modelos conceituais de BDG e os requisitos que eles suportam. A Tabela 2.1 apresenta os modelos conceituais levantados por Pinet (2012). Nesta tabela também são apresentadas as relações entre os requisitos e os modelos conceituais que suportam a representação desses requisitos.

Nas subseções a seguir são detalhados os modelos conceituais de BDG utilizados neste trabalho.

2.1.1 Modelo Conceitual GeoOOA

O Modelo GeoOOA (KÖSTERS, PAGEL e SIX, 1997) estende o modelo orientado a objetos Object-Oriented Analysis (OOA) (COAD e YOURDON, 1991). Neste modelo conceitual, classes convencionais e espaciais são diferenciadas por meio de símbolos gráficos.

A Figura 2.1 apresenta os ícones gráficos do GeoOOA utilizados para modelagem conceitual de fenômenos na visão de objetos e fenômenos na visão de rede.

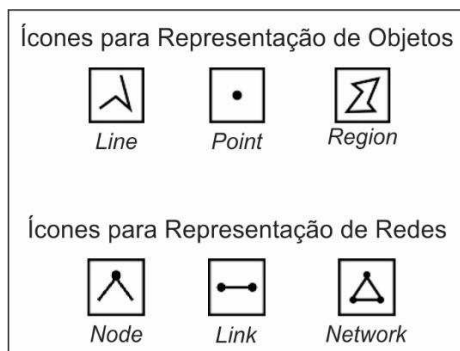


Figura 2.1: Ícones do GeoOOA para representação de fenômenos na visão de objetos e rede.

Tabela 2.1: Requisitos atendidos por modelos conceituais de BDG - Adaptado de Pinet(2012).

MODELOS CONCEITUAIS DE BDG X REQUISITOS	Modul-R (ER)	Cangoo (OOA)	GeoOOA (OOA)	GISER (ER)	GeoER (ER)	MADS (ER/OO)	Omega (UML)	GeoOM (OMT)	STER (ER)	PVL (UML)	Extended Spatio-Temporal (UML)	UML-GeoFrame (UML)	OMT-G (OMT)	Icons for GIS (ER/UML)	Semantics data modelo spatio-temporal database (UML)	Multiple Representation Schema Language (UML)	T-Omega (UML)	Conceptual framework for Spatio-temporal data modeling (OO)	ST USM (ER)	GEOUML (UML)	STGL Profile (UML)	ChronoGeoGraph (ER)
Representação dos tipos básicos de Objetos espaciais	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X
Especificação de relacionamentos espaciais		X	X		X	X	X	X				X	X				X	X		X		X
Representação de aspectos temporais	X		X			X			X	X	X	X						X	X			X
Múltipla representação espacial						X				X		X	X			X						X
Descrições de objetos com posições ou limites incertos						X									X						X	
Representação de campos contínuos						X		X				X	X									
Representação de rede			X									X	X							X		

A Figura 2.2 ilustra o uso dos requisitos de representação de redes e representação dos tipos básicos de objetos. A Figura 2.2 mostra um esquema de distribuição de energia elaborado com os construtores do modelo GeoOOA. Este esquema faz uso dos construtores de rede e de construtores de objetos espaciais básicos. A usina é representada pelo tipo geométrico polígono e a linha de transmissão é representada por um objeto do tipo linha. Os demais símbolos gráficos são classificados como nós e arestas e são utilizados para especificar o número de arestas e nós que compõem a rede Alta Voltagem, o número de arestas incidentes em um nó, e a quantas redes um nó ou uma aresta pode pertencer (STEMPLIUC, 2008).

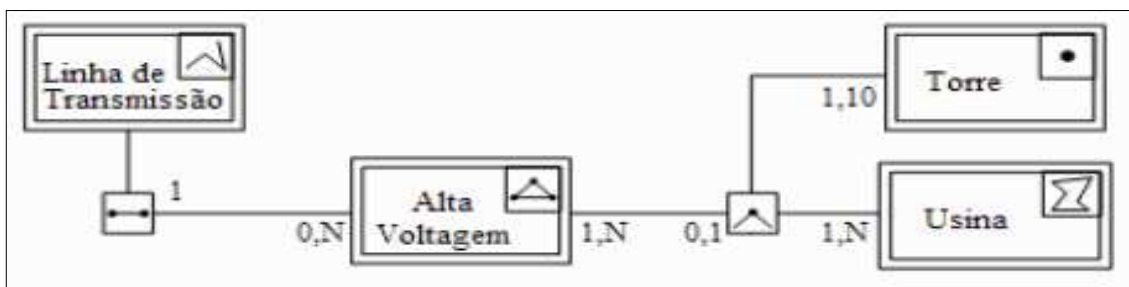


Figura 2.2: Exemplo Distribuição de Energia. Fonte: (KÖSTERS, PAGEL E SIX 1997).

2.1.2 Modelo Conceitual OMT-G

O modelo OMT-G (DAVIS, BORGES e LAENDER, 2002) é baseado no diagrama de classes do modelo Object Modeling Technique (OMT) (RUMBAUGH et al., 1991) e implementa primitivas geográficas, aumentando a capacidade semântica. Após a Object Management Group (OMG) ter adotado a UML como a linguagem de modelagem padrão, o modelo OMT-G passou a adotar os conceitos e notações da UML (STEMPLIUC, 2008). É característica deste modelo, a incorporação da representação de objetos espaciais na visão de campo e de objetos, permitindo a modelagem topológica e associações, podendo ser entre as classes espaciais e sobre classes espaciais e convencionais, estruturas de rede e múltiplas representações de um mesmo fenômeno geográfico (STEMPLIUC, 2008).

A Figura 2.3 apresenta os ícones do modelo conceitual OMT-G para modelagem conceitual de fenômenos na visão de campo, de objetos e de redes.



Figura 2.3: Ícones do OMT-G para representação de fenômenos na visão de campo, objetos e rede.

Como exemplo de uso do modelo OMT-G, a Figura 2.4 (a) apresenta um relacionamento simples onde uma edificação (objeto espacial) pertence a um proprietário. A Figura 2.4 (b) mostra o relacionamento entre dois objetos espaciais, onde um lote contém uma edificação.

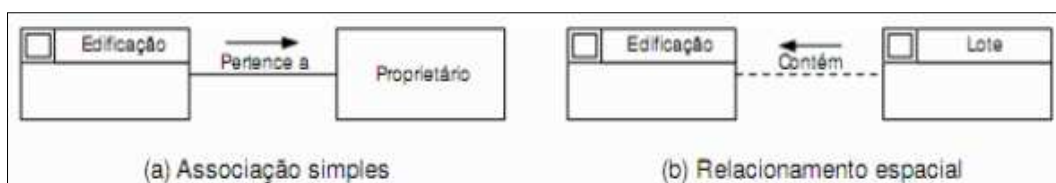


Figura 2.4: Exemplos de relacionamentos simples e espacial do modelo OMT-G (BORGES, DAVIS E LAENDER 2005).

2.1.3 Modelo Conceitual UML-GeoFrame

O UML-GeoFrame é uma extensão do modelo UML com a estrutura hierárquica de classes do framework GeoFrame (LISBOA FILHO E IOCHPE, 2008).

As características disponibilizadas neste modelo estão organizadas hierarquicamente em níveis de abstrações diferentes, sendo: nível de planejamento, nível de metamodelo e nível de representação espacial. A representação dos fenômenos geográficos é dada por símbolos gráficos (estereótipos) que podem ser colocados uma ou mais vezes na classe a ser representada. Com isso, o modelo dá suporte à múltipla representação espacial. O UML-GeoFrame inicialmente não possuía recursos suficientes para a modelagem de rede, sendo proposta uma extensão por Stempliuc (2008).

A Figura 2.5 apresenta os ícones utilizados no UML-GeoFrame para a representação de fenômenos espaciais na visão de Campos, de Objetos e de Rede. Esta tabela também apresenta os ícones utilizados na representação das generalizações.



Figura 2.5: Ícones do UML-GeoFrame para representação de fenômenos na visão de campo, objetos, rede e domínios geográfico.

No UML-GeoFrame os elementos geográficos podem ser representados por ícones de representação espacial e/ou por ícones de generalização, podendo ter mais estereótipos para a representação de escala (LISBOA FILHO e IOCHPE, 2008).

A Figura 2.6 ilustra um esquema conceitual de malha viária modelado a partir dos construtores dos tipos básicos de objetos espaciais do modelo UML-GeoFrame. Além da representação gráfica dos estereótipos espaciais básicos, observa-se o estereótipo de generalização. A Figura 2.6 também apresenta os relacionamentos entre objetos espaciais, como trecho de logradouro e cruzamento, onde um trecho de logradouro toca em dois cruzamentos e um cruzamento toca em n trechos de logradouro (n sendo 0 ou mais).

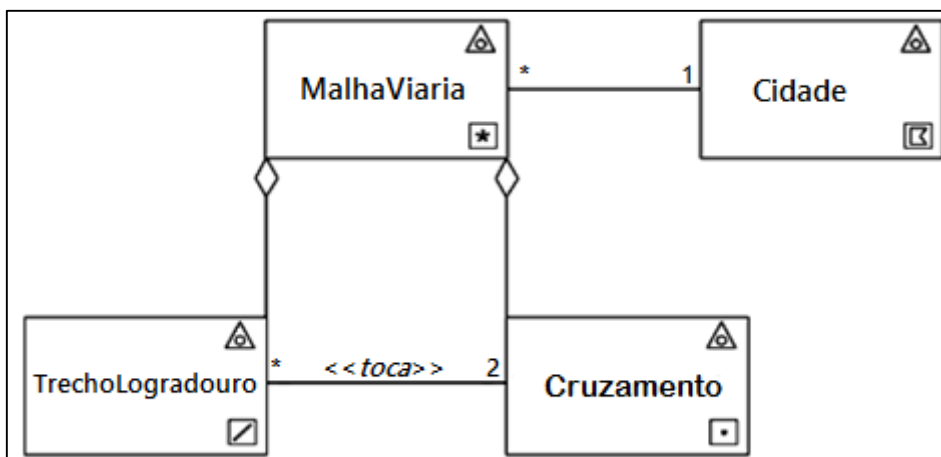


Figura 2.6: Modelagem conceitual de um sistema utilizando o modelo UML-Geoframe.

2.1.4 Modelo Conceitual PVL

O PVL é um modelo implementado na ferramenta CASE Perceptory (BÉDARD e LARRIVÉE, 2008). Este modelo consiste na união das extensões espaço-temporais com a UML (SAMPAIO, 2009).

O PVL não oferece construtores para a modelagem de campos geográficos e de redes, no entanto, com os construtores do PVL é possível a representação dos tipos básicos de objetos espaciais em espaços bidimensionais e tridimensionais. Com os construtores temporais do PVL é possível reproduzir instantes e intervalos de tempo (SAMPAIO, 2009).

A Figura 2.7 apresenta os ícones gráficos do PVL no espaço 2D. Os ícones apresentados são utilizados para representação de fenômenos na visão de objetos.

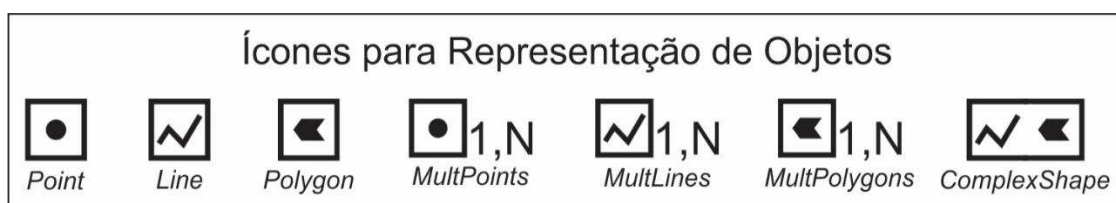


Figura 2.7: Pictogramas para representação de objetos geográficos no PVL

Como exemplo de uso do PVL, a Figura 2.8 apresenta um esquema conceitual do segmento de uma rua. Parte deste segmento está conectada por uma estrutura. A estrutura pode ser uma ponte, um viaduto, um túnel ou uma barragem. A barragem pode ser representada por um ponto ou por um segmento enquanto que o túnel, a ponte e o viaduto só podem ser representados por segmentos. A estrutura possui um nome, mas não há nenhum tipo de pictograma ligado a ela, por ser uma generalização das demais subclasses. As demais classes estão ligadas a um dos pictogramas apresentados na Figura 2.7.

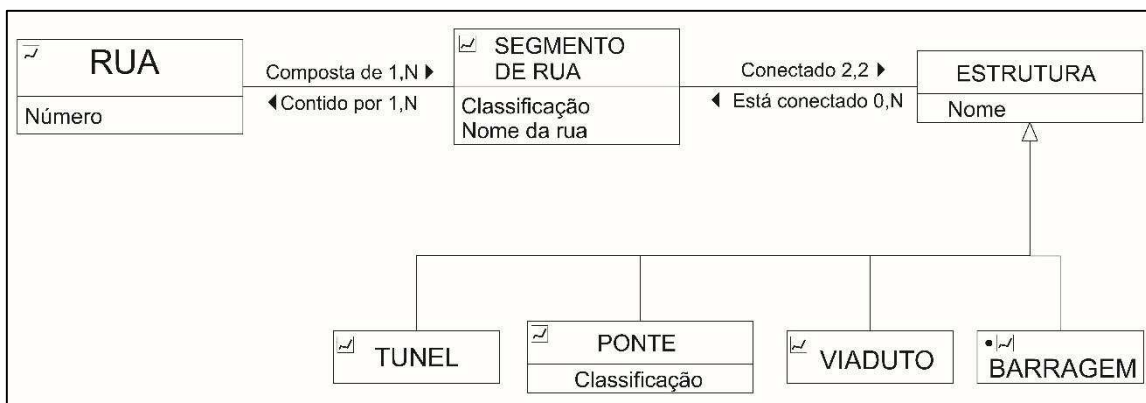



Figura 2.8: Esquema conceitual desenvolvido a partir dos construtores do PVL – Adaptado de Perceptory (s.d.).

2.2 Uso de Perfil UML para modelagem de dados geográficos

Para que a UML atenda detalhes disponíveis em linguagens mais específicas sem que se torne excessivamente complexa, seus criadores desenvolveram um recurso denominado Perfil (ERIKSSON et al., 2004). Eriksson et al. (2004) definem o perfil como “conjunto de mecanismos de extensão da UML (estereótipos, tagged values e constraints) agrupados em um pacote UML estereotipado como <<profile>>”.

O mecanismo de extensão da UML nomeado de estereótipo é caracterizado pela OMG (2007) como um dos principais veículos para customização da UML. O estereótipo pode estender uma metaclassa ou outro estereótipo, permitindo assim a criação de novos elementos de modelagem. Não há limites quanto ao número de vezes que uma metaclassa possa ser estendida (ERIKSSON et al., 2004). Além do nome dado ao estereótipo como, por exemplo, <<Point>>, um ícone gráfico pode ser adicionado (ex.: )

Segundo Nalon (2010), o Tagged Value é um meta-atributo adicional com nome e tipo vinculado a um estereótipo. Além de adicionar informações como autor, hora e a data aos elementos do modelo, os Tagged Values podem ser usados por seres humanos com finalidades administrativas e por máquinas com finalidade de geração de código (ERIKSSON et al., 2004; OMG, 2007).

As constraints podem ser expressas usando linguagens de restrição, por exemplo, a OCL. A OCL é um add-on da UML adotada pela OMG para expressar restrições e propriedades de elementos de modelos (FUENTES e VALLECILLO, 2004; WARMER e KLEPPE, 2003).

Esta tecnologia de extensão da UML pode ser usada para modelagem de dados geográficos, como o Perfil GeoProfile, descrito a seguir.

2.2.1 Perfil UML GeoProfile

O GeoProfile é um perfil UML que une características de maior destaque de diversos modelos conceituais específicos para modelagem de BDG, como os modelos OMT-G, MADS, GeoOOA, UML-GeoFrame e o PVL. O GeoProfile foi especificado por Sampaio (2009) e integrado à abordagem MDA por Nalon (2010).

Por ser um perfil UML, o GeoProfile possibilita o nível de abstração mais alto garantido pela abordagem MDA, auxiliando os projetistas nos primeiros passos do projeto (NALON, 2010). Outras vantagens incluem: ser naturalmente suportado por ferramentas CASE já consolidadas pela infraestrutura UML; e fazer uso de restrições OCL em esquemas de BDG (LISBOA FILHO et al., 2010).

2.2.1.1 Especificação do GeoProfile

Para especificação do GeoProfile, Sampaio (2009) inicialmente construiu um metamodelo do domínio com base nos principais modelos conceituais de BDG. São eles: OMT-G (BORGES; DAVIS; LAENDER, 2001), MADS (PARENT; SPACCAPIETRA; ZIMÁNUI, 2008), GeoOOA (KÖSTERS; PAGEL; SIX, 1997), UML-GeoFrame (LISBOA FILHO; IOCHPE, 2008) e o modelo PVL, implementado na ferramenta Perceptory (BÉDARD, 1999). A Figura 2.9 apresenta o metamodelo do GeoProfile. As regiões geográficas são caracterizadas por temas. Os temas podem ser representados por agregação de outros temas ou por fenômenos geográficos (GeoPhenomenon) e ou objetos convencionais (ConventionalObj) (SAMPAIO, 2009). Objetos espaciais podem ser representados na visão de campo (GeoField) ou na visão de objetos (GeoObject). O GeoProfile também oferece construtores para representação de rede. A metaclassa Network do GeoProfile é utilizada para armazenar informações alfanuméricas da rede, por isto ela herda característica da metaclassa ConventionalObj. As redes são formadas por objetos de redes (NetworkObj) que podem ser nodos (Node), arcos unidirecionais (Unidirectiona) ou arcos bidirecionais (Bidirectional).

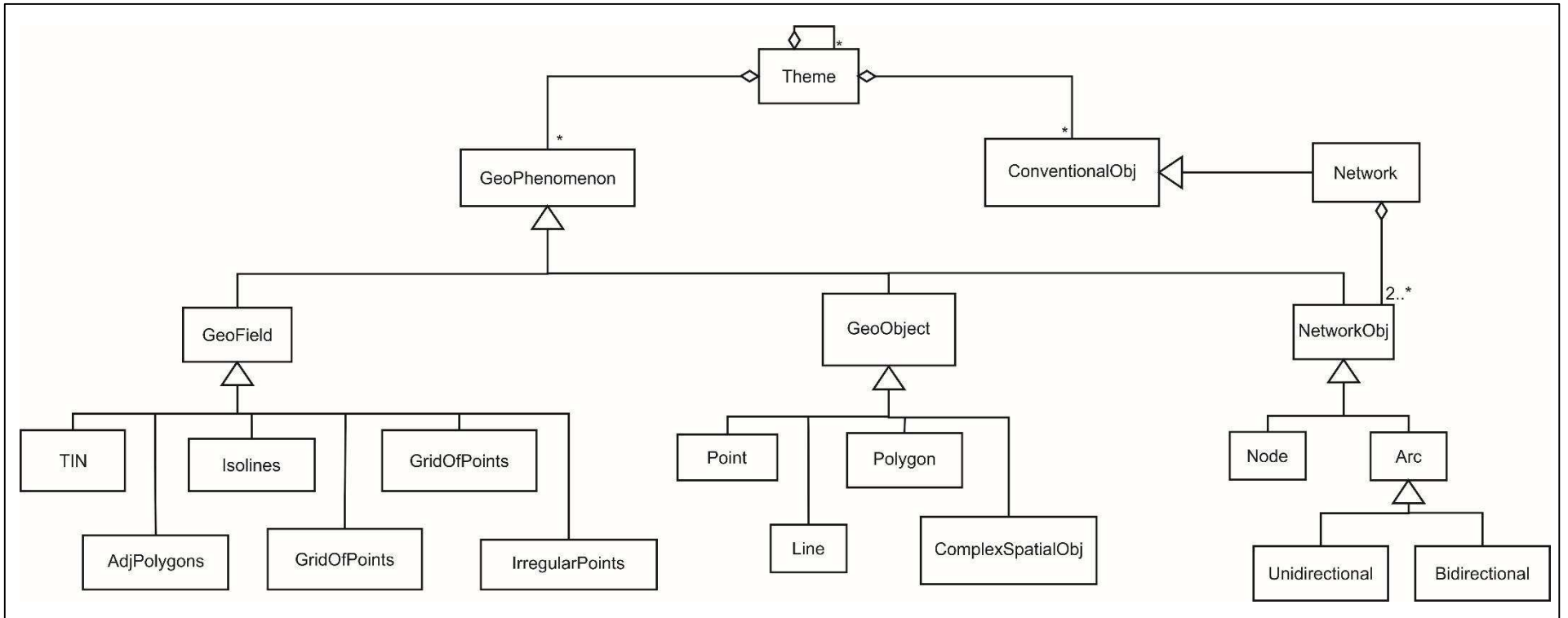


Figura 2.9: Metamodelo do GeoProfile.

A partir do metamodelo da Figura 2.9, Sampaio (2009) definiu estereótipos que estendem a metaclassa class. Sampaio (2009) também definiu estereótipos para representação dos relacionamentos e representação temporal. Os estereótipos de relacionamentos estendem a metaclassa association. Sampaio também especificou restrições OCL e criou ícones gráficos para o GeoProfile.

A Figura 2.10 mostra o metamodelo do GeoProfile especificado a partir da tecnologia de Perfil UML. Existe um pacote estereotipado <<profile>>, nomeado de GeoProfile. Dentro deste pacote existem três tipos de elementos da UML: <<metaclassa>>; <<estereótipo>>; e <<enumeração>>. Os estereótipos que estendem a metaclassa class são usados para a representação de fenômenos na visão de rede, fenômenos na visão de objetos, fenômenos na visão de campos e o tempo de duração dos fenômenos. Os estereótipos que estendem a metaclassa association são usados para representar os relacionamentos espaciais e/ou temporais entre fenômenos na visão de objetos. As enumerações são usadas no GeoProfile para a definição de temporalidade de um fenômeno geográfico.

Os estereótipos da Figura 2.10 que possuem seus nomes em itálico são considerados abstratos e não devem ser usados como elementos na construção de esquemas conceituais. Sampaio (2009) definiu estereótipos abstratos para organizar em grupos os demais estereótipos e para conter restrições comuns a todos os estereótipos do grupo.

As restrições que compõem o GeoProfile foram especificadas a partir da linguagem OCL. Estas restrições foram utilizadas no GeoProfile para evitar três tipos de erros na construção de esquemas conceituais: o primeiro refere-se a má construção de redes, o segundo refere-se a adição de relacionamentos topológicos impossíveis e o terceiro refere-se a adição de estereótipos incompatíveis no mesmo elemento (SAMPALIO, 2009). Neste trabalho as restrições OCL não serão apresentadas.

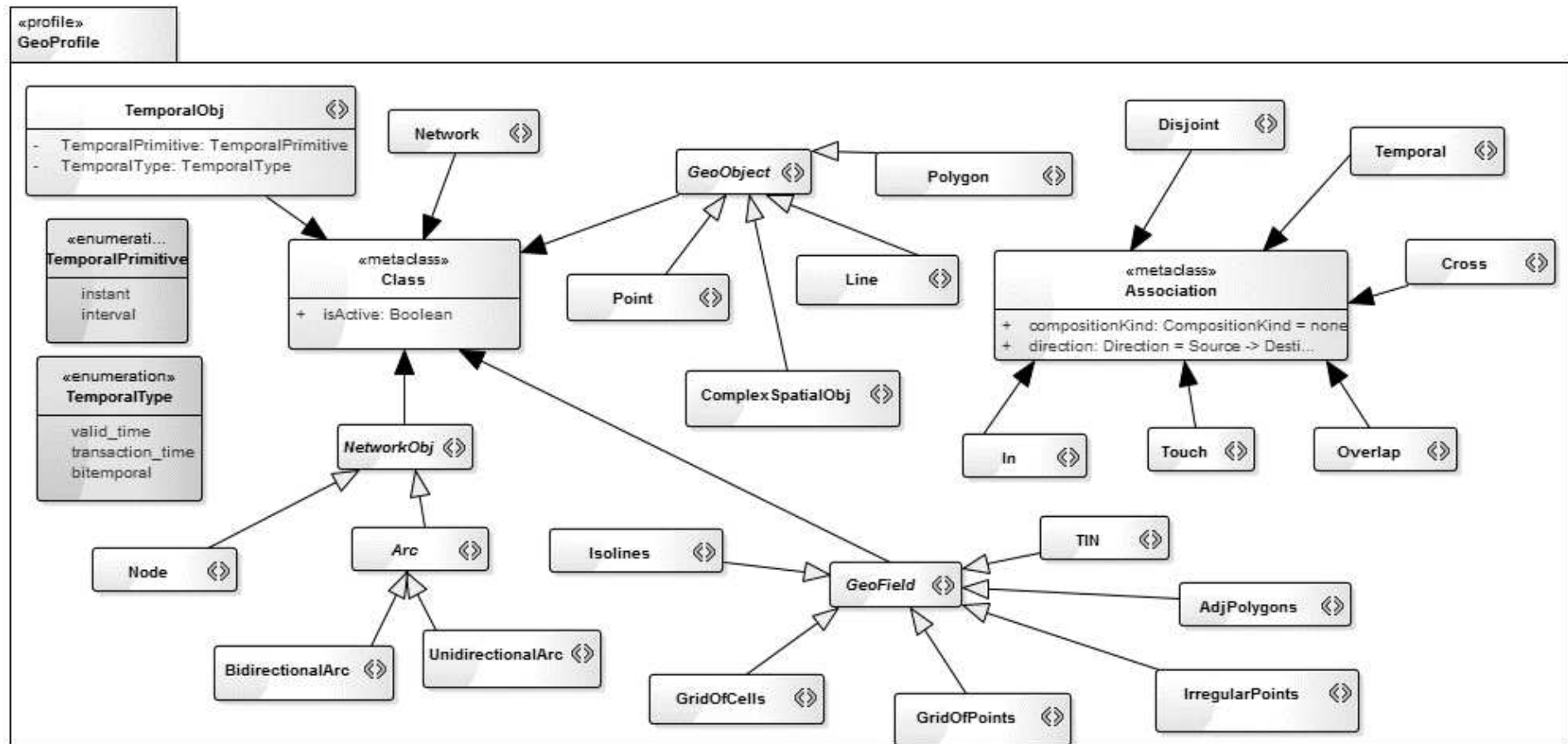


Figura 2.10: Estereótipos do Perfil GeoProfile.

Para cada estereótipo não abstrato do GeoProfile, Sampaio (2009) criou um ícone gráfico. A Figura 2.11 apresenta os ícones gráficos dos estereótipos do GeoProfile.

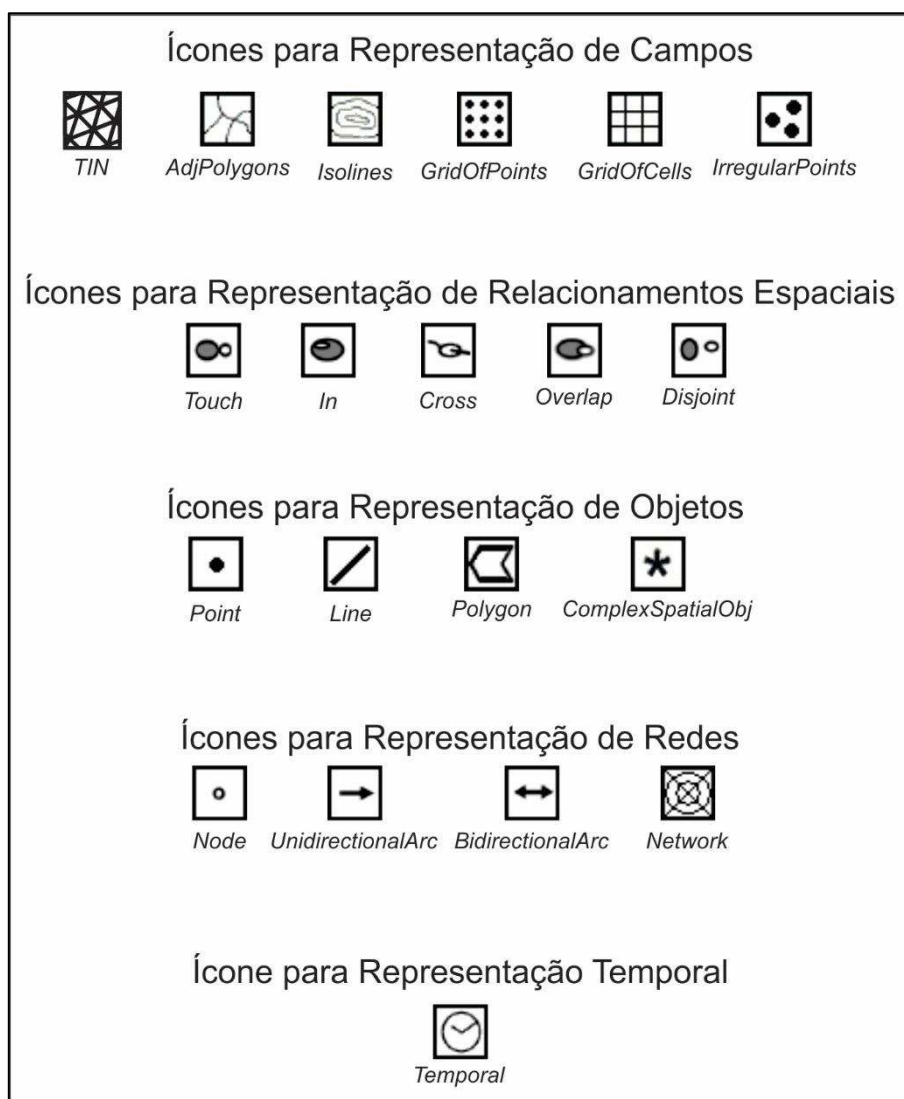


Figura 2.11: Ícones usados no GeoProfile – Adaptado de Sampaio (2009).

As subseções seguintes descrevem os conjuntos de estereótipos do GeoProfile que são utilizados no desenvolvimento deste trabalho.

2.2.1.2 Estereótipos para a modelagem de fenômenos na visão de objetos

De acordo com Câmara (2005), existe um paradigma de quatro universos que um determinado fenômeno espaço/temporal deve ser submetido para que possa ser representado computacionalmente. Um dos universos é o universo estrutural, dividido em duas grandes estruturas, sendo a estrutura vetorial e a estrutura matricial (CÂMARA, 2005).

As estruturas de dados vetoriais são utilizadas para representar as fronteiras de cada entidade geográfica através de formas geométricas, por exemplo, ponto, linha e polígono (CÂMARA, 2005).

Para a representação formal desta particularidade do universo estrutural no metamodelo do GeoProfile foram especificados quatro estereótipos: point, line, polygon e ComplexSpatialObj. A Figura 2.12 ilustra um esquema conceitual de pontos estratégicos para colocar hidrantes em uma cidade. A classe Hidrante é representada pelo estereótipo point e a classe cidade pelo estereótipo polygon.

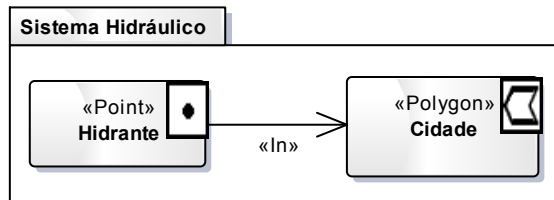


Figura 2.12: Aplicando o GeoProfile na visão de Objetos.

2.2.1.3 Estereótipos para a modelagem de fenômenos na visão de campo

O GeoProfile especifica seis estereótipos para tratar este tipo de representação: TIN; AdjPolygons; Isonlines; GridOfPoints; GridOfCells; e IrregularPoints. Os estereótipos GridOfCells e GridOfPoints são representados no universo estrutural pela estrutura de matriz, e os demais estereótipos citados anteriormente são representados pela estrutura vetorial do universo estrutural. A Figura 2.13 ilustra o uso destes estereótipos para modelagem conceitual de algumas condições abióticas do meio ambiente.

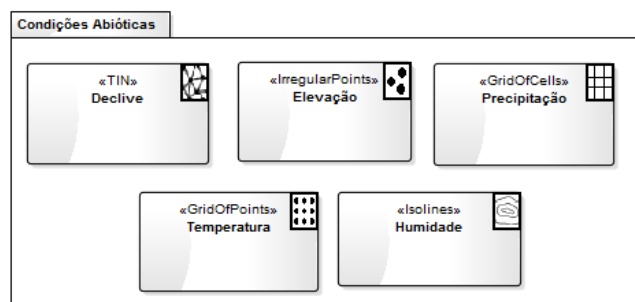


Figura 2.13: Aplicando o GeoProfile na visão de campo.

2.2.1.2 Estereótipos para a modelagem de rede

Um outro universo discutido por Câmara (2005) é o universo formal. Neste universo são discutidos conceitos de espaço relativo e espaço absoluto. No espaço absoluto a localização exata dos objetos geográficos deve ser representada no computador, já no espaço relativo, é o posicionamento relativo entre os objetos espaciais que devem ser representados. O modelo de representação em rede pertence ao espaço relativo (CÂMARA, 2005).

Segundo Stempliuć (2008), a representação de redes pode ser concebida através de pontos e linhas definidas no universo estrutural. No entanto, o GeoProfile possui uma estrutura própria para modelagem de rede. Ao invés de utilizar os estereótipos <<point>> e

<<line>>, Sampaio (2009) especificou estereótipos próprios: <<Network>>; <<NetworkObj>>; <<Node>>; e <<Arc>>. O estereótipo <<NetworkObj>> é especializado em <<Node>> e <<Arc>>. O estereótipo <<Arc>> é especializado em <<UnidirectionalArc>> e <<BidirectionalArc>>.

A Figura 2.14 ilustra a utilização dos estereótipos de rede do GeoProfile. Uma Rede Elétrica (Network) de alta voltagem é constituída de usinas (Node), torres (Node) e Linhas de transmissão (BidirectionalArc). Esta rede Alta Voltagem (Conventional Class) pode possuir mais de um transformador (Node).

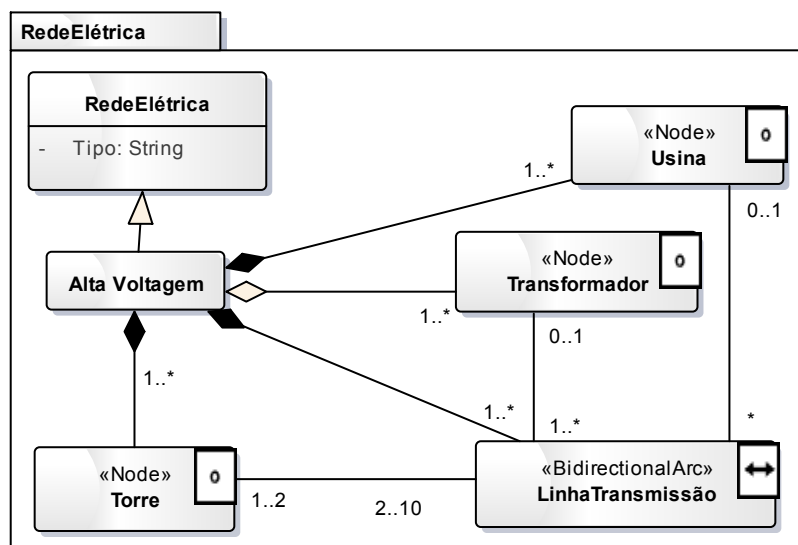


Figura 2.14: Aplicando o GeoProfile na visão de rede – Adaptado de: Sampaio (2009).

2.3 Integração da abordagem MDA com o GeoProfile

MDA é uma abordagem proposta pelo OMG (OMG, 2003), que permite o desenvolvimento de sistemas em diferentes níveis de abstração. Segundo Kleppe et al (2003), a MDA possui três níveis de abstração: CIM, PIM e PSM. O CIM não mostra detalhes da estrutura do sistema, mas o ambiente em que o sistema vai operar. Em relação ao GeoProfile, os estereótipos são aplicados na classe, podendo ser, por exemplo, representação do tipo ponto, linha ou polígono. O PIM é um modelo independente de qualquer tecnologia de implementação, contendo os requisitos de software necessários. No GeoProfile, os estereótipos são apresentados como atributos da classe. O PSM especifica detalhes a respeito da plataforma na qual será implementado.

A Figura 2.15 ilustra a transformação do esquema conceitual do GeoProfile nos diversos níveis do MDA. O esquema conceitual refere-se ao relacionamento entre cidade (City) e Distrito (District). O tipo de dado GM_Polygon no nível de abstração PIM e PSM

refere-se ao tipo de dado geográfico adotado pela ISO 19107. Este tipo de dado substitui o tipo polygon do GeoProfile.

A partir do nível PSM é possível gerar código fonte em Data Definition Language (DDL) para o desenvolvimento de um BDG.

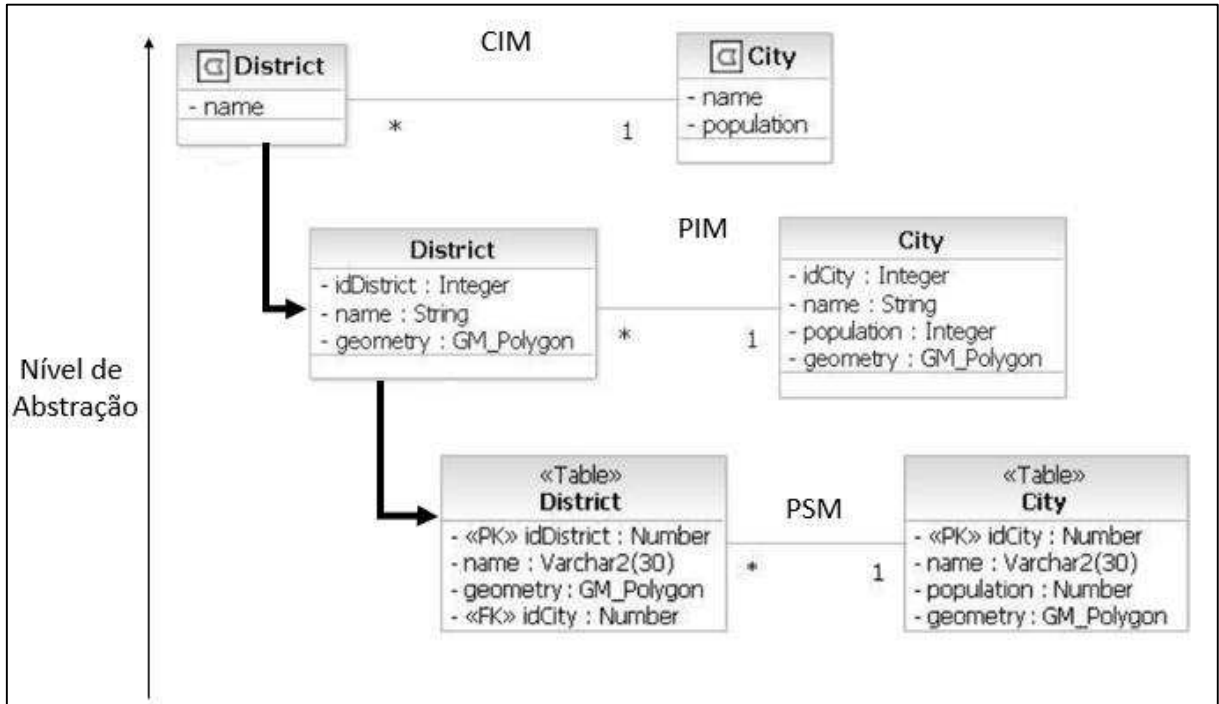


Figura 2.15: Níveis de abstração MDA integrados ao Perfil GeoProfile – Fonte: (Nalon 2010).

2.4 Interoperabilidade de esquemas de dados

Interoperabilidade é a comunicação transparente entre ambientes distintos que acontece sem esforço especial por parte de usuários (HUFNAGEL, 2009). Segundo Staub (2007), a interoperabilidade de esquemas de dados pode acontecer em duas direções, vertical e horizontal (Figura 2.16).

A interoperabilidade vertical acontece entre os diferentes níveis da abordagem MDA. Um domínio da aplicação A do mundo real (CIM) é modelado conceitualmente a partir dos construtores do metamodelo X. O resultado desta modelagem é um esquema conceitual (PIM). O esquema conceitual pode ser transformado em um esquema lógico (PSM), que por sua vez pode ser transformado em código fonte, no caso da Figura 2.16, uma instância de banco de dados. O mapeamento entre estes esquemas promove a interoperabilidade vertical entre níveis de abstração do MDA. O mesmo acontece para o domínio da aplicação B.

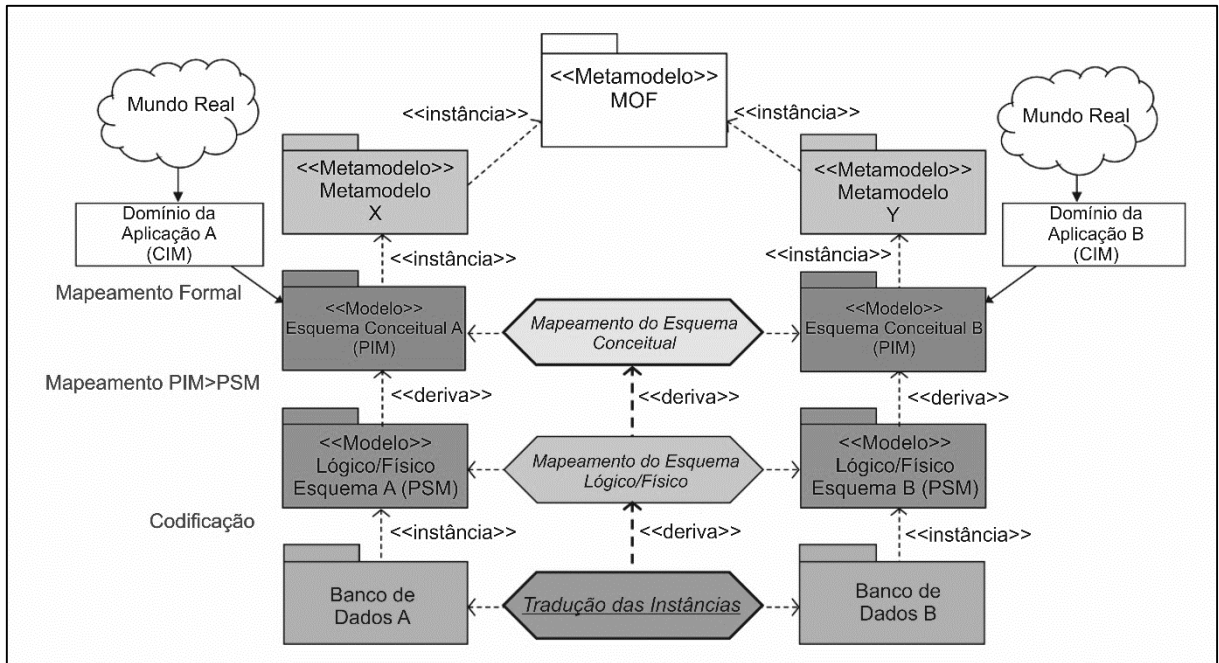


Figura 2.16: Interoperabilidade de esquemas conceituais – Fonte: (Staub 2007).

O conceito de interoperabilidade horizontal, defendido por Staub (2007) é que mapeamentos de esquemas podem ser feitos em cada nível da abordagem MDA. Por exemplo, no nível PIM é feito o mapeamento do esquema conceitual. Assim, um esquema conceitual, desenvolvido a partir dos construtores do metamodelo X, também pode ser construído pelos construtores do metamodelo Y e vice-versa. Isto pode acontecer entre os diferentes níveis do MDA, como ilustrado na Figura 2.16.

Tanto o mapeamento horizontal quanto o mapeamento vertical são constituídos de regras de transformação. Estas regras de transformação podem ser executadas manualmente ou automaticamente (por ferramentas de transformação).

2.5 Uso de ferramentas de transformação para interoperabilidade de esquemas

Segundo Staub (2007), as transformações de esquemas podem ser conseguidas de forma manual ou automática. Na transformação manual, as regras de transformação são executadas pelo projetista. Na transformação automática, as regras de transformação são executadas por ferramentas de transformação. A Figura 2.17 mostra a utilização de ferramentas de transformação para alcançar a interoperabilidade vertical entre níveis da abordagem MDA. Cada seta bidirecional desta figura representa códigos de transformação.

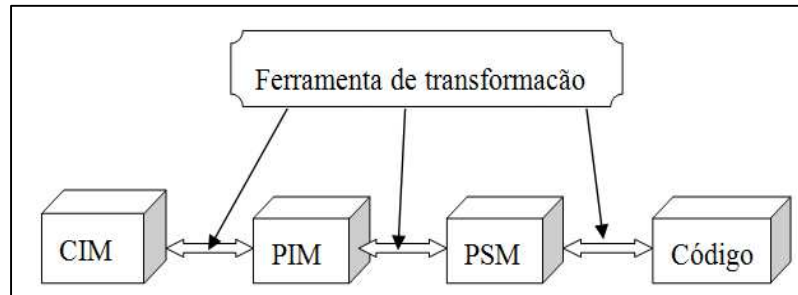


Figura 2.17: Utilização de ferramentas de transformação na abordagem MDA – Ferreira (2014).

As ferramentas de transformação também podem ser usadas para alcançar a interoperabilidade horizontal (FERREIRA et al. 2014).

2.5.1 Ferramenta CASE Enterprise Architect

A Enterprise Architect (EA) é uma ferramenta CASE comercial, licenciada pela Sparx Systems, que permite a criação de perfis UML de forma visual e a inserção e validação sintática de expressões OCL. A EA não oferece recursos para validação semântica de expressões OCL.

Além de ser uma ferramenta para modelagem, a EA atua como uma ferramenta de transformação MDA, apresentando uma linguagem própria para transformação entre os esquemas. Esta linguagem pode ser modificada para que seus usuários consigam chegar até o último nível da abordagem MDA, o código fonte (FERREIRA et al. 2014). Como a modelagem neste contexto refere-se a BDG, a última etapa MDA de interesse é o código fonte em DDL, sendo que a EA oferece recursos para a geração de códigos DDL.

Nesta ferramenta os estereótipos podem ser representados graficamente ou na forma textual. A ferramenta também oferece recursos para a múltipla representação de estereótipos. Além destes recursos, a ferramenta EA oferece uma tecnologia chamada MDG.

A tecnologia MDG oferece recursos para que a capacidade de modelagem da ferramenta EA seja estendida sem a necessidade de alteração do código fonte (EA, 2015). Ao usar esta tecnologia, todo ambiente da EA é configurado para atender a um domínio de modelagem específico.

Esta tecnologia consiste em um arquivo eXtensible Markup Language (XML) que pode incluir os recursos Profile toolboxes, MDA Transforms, images, Scripts, profiles, entre outros. A Figura 2.18 mostra uma tela de configuração desta tecnologia, onde estão listados os recursos que a tecnologia MDG pode oferecer.

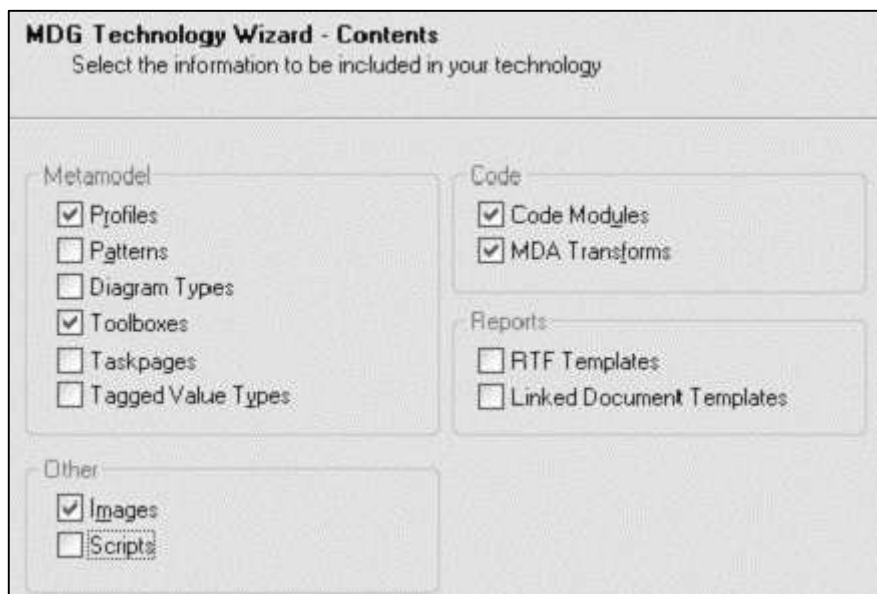


Figura 2.18: Recursos que podem ser incluídos na tecnologia MDG da ferramenta EA.

Na ferramenta EA a representação gráfica de estereótipos do perfil em esquemas conceituais também é realizada por meio da tecnologia MDG. Cada estereótipo do Perfil UML deve conter um atributo `_image`, com um script indicando o nome e a extensão da imagem anteriormente carregada na ferramenta. No script também são indicados a posição e o tamanho da imagem dentro do elemento da UML.

Toolboxes é um recurso oferecido pela ferramenta EA para criar uma caixa de ferramenta específica do perfil. Desta forma, os estereótipos do perfil podem ser organizados em diferentes grupos na caixa de ferramenta. Por exemplo, os estereótipos para modelagem de relacionamentos em BDG podem ser organizados no grupo Relacionamentos Espaciais. Os estereótipos na caixa de ferramenta podem ser representados por ícones.

MDA transforms é um recurso que permite o desenvolvimento de novas transformações ou a customização de transformações existentes.

O Apêndice A apresenta maiores detalhes sobre como construir uma tecnologia MDG para o domínio geográfico.

3 MÉTODO PARA INTEROPERABILIDADE DE ESQUEMAS DE DADOS GEOGRÁFICOS

Este capítulo descreve o método proposto para alcançar interoperabilidade horizontal e vertical de esquemas de dados geográficos, utilizando o perfil UML GeoProfile (SAMPAIO, 2009) como modelo conceitual intermediário. O método foi testado utilizando a ferramenta CASE EA. O método é composto das seguintes etapas:

- (1) seleção de um conjunto de modelos e requisitos a serem estudados;
- (2) especificação de parte do metamodelo para cada modelo selecionado na etapa 1;
- (3) especificação do Perfil UML e restrições OCL para que as restrições impostas em cada modelo sejam verificadas pela ferramenta CASE;
- (4) aplicar a tecnologia MDG da ferramenta EA para cada Perfil UML elaborado na etapa (3);
- (5) mostrar, por meio de exemplos, a interoperabilidade horizontal entre modelos conceituais;
- (6) estender o Perfil UML GeoProfile e repetir as etapas 2, 3, 4 e 5 para a extensão do GeoProfile;
- (7) mostrar, por meio de exemplos, a interoperabilidade vertical usando transformações entre níveis da abordagem MDA.

Na Tabela 2.1, do capítulo 2, foram relacionados 22 modelos conceituais de BDG com 7 requisitos espaço/temporais. Mostrar a interoperabilidade horizontal e vertical entre todos os modelos e para todos requisitos torna-se inviável no presente trabalho.

Para a etapa (1) foi feita uma revisão de literatura, onde constatou-se que a maioria dos modelos conceituais utilizam soluções parecidas para modelagem conceitual de alguns requisitos como, por exemplo, a utilização de ícones gráficos (pictogramas) em classes para representação de geometrias espaciais. No entanto, existem algumas diferenças entre os modelos conceituais que tornam difícil o processo de interoperabilidade. Por exemplo, alguns modelos associam elementos de redes com formas geométricas (ex.: UML-GeoFrame), desta forma um poste que normalmente é representado por uma geometria espacial do tipo ponto pode ser também representado como nodo da rede. Em outros modelos, como o OMT-G, a representação de elementos de rede é feita de forma separada da representação espacial. A etapa (1) é apresentada na seção 3.1.

A etapa (2) consiste na especificação de parte do metamodelo para cada modelo conceitual selecionado na etapa (1). A especificação do metamodelo tem como foco apenas os

requisitos relacionados com o modelo, como mostra a Tabela 3.1. Por exemplo, o metamodelo do modelo PVL foi especificado para atender apenas ao requisito Representação dos tipos básicos de objetos espaciais. A etapa de especificação dos metamodelos é apresentada na seção 3.2.

A partir dos metamodelos especificados na etapa (2) foram especificados perfis UML e restrições OCL para que as restrições dos modelos conceituais continuem sendo cumpridas na tecnologia de Perfil UML. A especificação de perfis UML e as restrições OCL (etapa 3) estão descritas na seção 3.3.

Na etapa (4) os Perfis UML especificados na etapa (3) foram submetidos ao processo de tecnologia MDG na ferramenta EA. Os recursos utilizados desta tecnologia são: perfis, Toolboxes, imagens com scripts e MDA Transforms. Estes recursos da tecnologia MDG auxiliam na construção de ambientes de modelagem conceitual específicos, aproximando-se do ambiente de modelagem do modelo conceitual original. Estes recursos também foram utilizados no GeoProfile. Neste trabalho, cada metamodelo submetido à tecnologia MDG da ferramenta EA recebe o nome “MDG_” + “nome do metamodelo”, por exemplo, MDG_PVL. O desenvolvimento desta tecnologia para os metamodelos está detalhado na seção 3.4.

A tecnologia MDG_GeoProfile será utilizada como base de interoperabilidade entre as tecnologias MDG_UML-GeoFrame, MDG_OMT-G, MDG_PVL e MDG_GeoOOA.

Para mostrar a interoperabilidade horizontal entre modelos que suportam o mesmo requisito (etapa 5), foram especificados mapeamentos horizontais entre as tecnologias MDG_ (PVL, UML-GeoFrame, OMT-G e GeoOOA) com a tecnologia MDG_GeoProfile. Neste trabalho também foram propostas adaptações ao Perfil UML GeoProfile, quando o mesmo não apresentou construtores correspondentes aos dos demais modelos conceituais. A execução da etapa (5) está descrita na seção 3.5.

A interoperabilidade vertical (etapa 6) é utilizada para mostrar a equivalência de esquemas nos diferentes níveis MDA. Desta forma, um esquema conceitual desenvolvido a partir de um modelo conceitual qualquer é mapeado para os demais níveis MDA. O mesmo esquema conceitual também é mapeado para outro modelo conceitual (interoperabilidade horizontal), e em seguida mapeado para os demais níveis MDA. Se o PSM e código fonte forem equivalentes é possível mostrar que a interoperabilidade horizontal também acontece. Na seção 3.6 são apresentados os estudos de interoperabilidade vertical.

3.1 Etapa 1: Seleção de Modelos e Requisitos

Para mostrar a possibilidade de interoperabilidade com base no GeoProfile foram selecionados alguns modelos que suportam a modelagem de alguns requisitos.

Neste trabalho não serão utilizados todos os modelos e requisitos da Tabela 2.1. Desta forma, os modelos escolhidos para serem usados neste trabalho foram: UML-GeoFrame, PVL, OMT-G e GeoOOA. Os requisitos escolhidos para serem usados neste trabalho foram: Representação dos tipos básicos de objetos espaciais, Representação de campos contínuos e Representação de redes. A Tabela 3.1 apresenta a relação entre os requisitos e modelos conceituais de BDG escolhidos.

Tabela 3.1: Relação entre modelos conceituais e requisitos selecionados.

Requisitos X Modelos	GeoOOA	PVL	UML-GeoFrame	OMT-G
Representação dos tipos básicos de Objetos espaciais	X	X	X	X
Representação de Campos Contínuos			X	X
Representação de Redes	X		X	X

3.2 Etapa 2: Especificação de Metamodelos

Nesta seção são especificados os metamodelos dos modelos selecionados. A especificação do metamodelo foi feita para descrever os construtores que modelam cada requisito.

3.2.1 Especificação de metamodelos para representação de tipos básicos de objetos espaciais

Segundo Pinet (2012), os objetos espaciais armazenados em sistemas de informação geográfica podem ser de diferentes tipos:

- Linha, ponto e polígono;
- Múltiplos pontos, múltiplas linhas e múltiplos polígonos;
- Conjunto de objetos de vários tipos.

Os modelos UML-GeoFrame, OMT-G, PVL e GeoOOA oferecem construtores para modelagem de objetos espaciais do tipo linha, ponto e polígono. Nos quatro modelos apresentados, este tipo de representação é feito por meio de ícones gráficos, ou estereótipos, que são adicionados à representação gráfica de uma classe. No PVL, os tipos básicos de objetos espaciais podem ser representados tanto no espaço bidimensional quanto no espaço tridimensional (BÉDARD e LARRIVÉE, 2008).

Neste trabalho, o espaço tridimensional não foi abordado. Esta dimensão é uma particularidade do modelo conceitual PVL.

Os modelos conceituais GeoOOA e OMT-G não tratam a representação de objetos complexos. As Figuras 3.1 e 3.2 ilustram partes dos metamodelos GeoOOA e OMT-G, respectivamente. Os metamodelos apresentados nestas figuras atendem unicamente ao requisito representação dos tipos básicos de objetos espaciais. A metaclasses GeoObj é usada para agrupar as geometrias pertencentes ao requisito em questão.

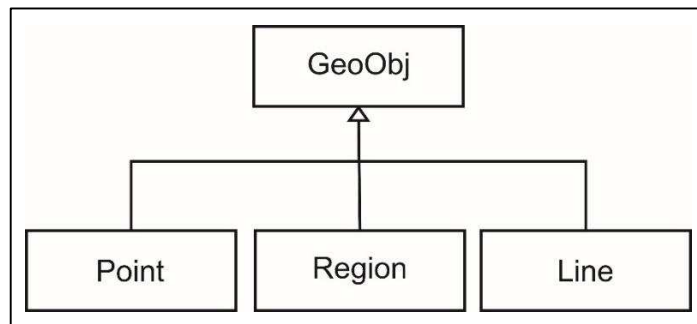


Figura 3.1: Metamodelo GeoOOA para representação dos tipos básicos de objetos espaciais.

O modelo OMT-G possui duas classes do tipo GeoObjeto: GeoObjeto com geometria e GeoObjeto com topologia (BORGES, 1997). A Figura 3.2 ilustra a classe dos Geo-objetos com geometria, chamada metaclasses Geometry.

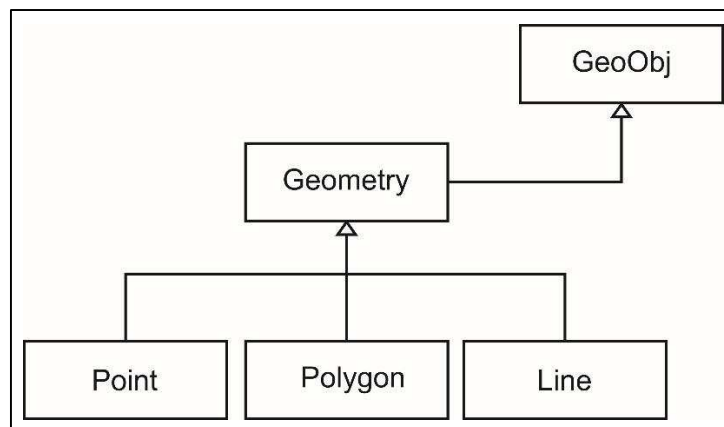


Figura 3.2: Metamodelo OMT-G para representação dos tipos básicos de objetos espaciais.

O modelo conceitual UML-GeoFrame apresenta uma solução para modelagem de geometrias complexas por meio do pictograma/estereótipo (⊠) colocado em uma classe. Com esta solução não é possível fazer a distinção de quais aglomerados de geometrias estão sendo representados no esquema conceitual. Não se sabe, por exemplo, se é um conjunto de pontos, conjunto de linhas, conjunto de polígonos ou um conjunto com geometrias mistas.

A Figura 3.3 ilustra parte do metamodelo do UML-GeoFrame. Esta parte do metamodelo destina-se a representação do requisito “fenômeno geográfico na visão de objetos”. A metaclassa GeoObj é utilizada para agrupar as geometrias deste requisito.

Um fenômeno geográfico na visão de objetos (GeoObj) pode ser representado por objetos espaciais (SpatialObject), e objetos espaciais podem ser utilizados para representação de um fenômeno complexo na visão de objetos. Um objeto espacial pode ser representado por uma geometria do tipo point, line ou polygon. O objeto espacial também pode ser representado por um conjunto de dois ou mais SpatialObject (ComplexSpatialObj).

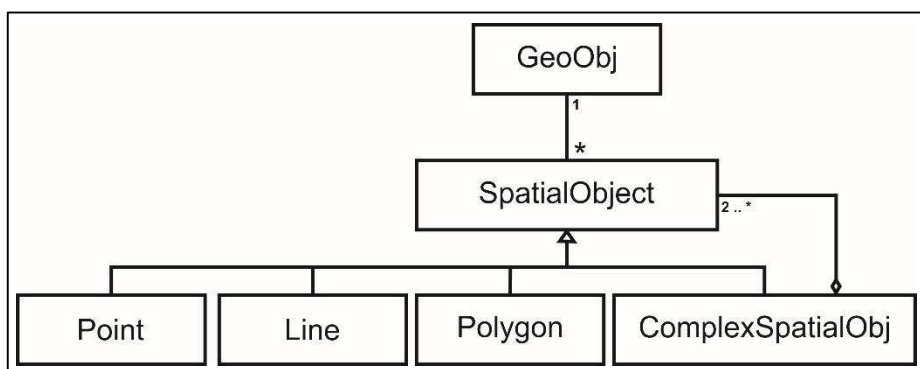


Figura 3.3: Metamodelo UML-GeoFrame para representação dos tipos básicos de objetos espaciais.

O modelo conceitual PVL oferece recursos para modelagem conceitual de geometrias complexas. Existem cardinalidades que são colocadas em conjunto com pictogramas para representar múltiplos pontos, múltiplas linhas e múltiplos polígonos ($\boxed{\bullet}1,N$)¹, este tipo de representação é chamado de Agregação Simples (BÉDARD e LARRIVÉE, 2008). Para geometrias distintas, chamados de Agregação Complexa, a solução utilizada no PVL é a união de diferentes símbolos gráficos no pictograma da classe. Por exemplo, objetos complexos que agregam linhas e polígonos são modelados com o pictograma ($\boxed{\text{---}\bullet}$) (BÉDARD e LARRIVÉE, 2008).

É importante distinguir a representação da agregação complexa, com a representação de múltipla representação espacial no modelo PVL. Na múltipla representação espacial símbolos gráficos estão envolvidos por retângulos diferentes (ex.: $\boxed{\bullet}\boxed{\text{---}}$). Na agregação complexa os símbolos gráficos estão envolvidos pelo mesmo retângulo (ex.: $\boxed{\text{---}\bullet}$).

A proposta de um metamodelo que atenda parte das necessidades do PVL é apresentada na Figura 3.4. Foi acrescentada a metaclassa GeoObj para agrupar as possíveis representações dos tipos básicos de objetos. A metaclassa AgregationGeometry foi especificada para organizar as possibilidades de representações complexas do PVL.

¹ O número 1, e N definem o mínimo e o máximo de geometrias.

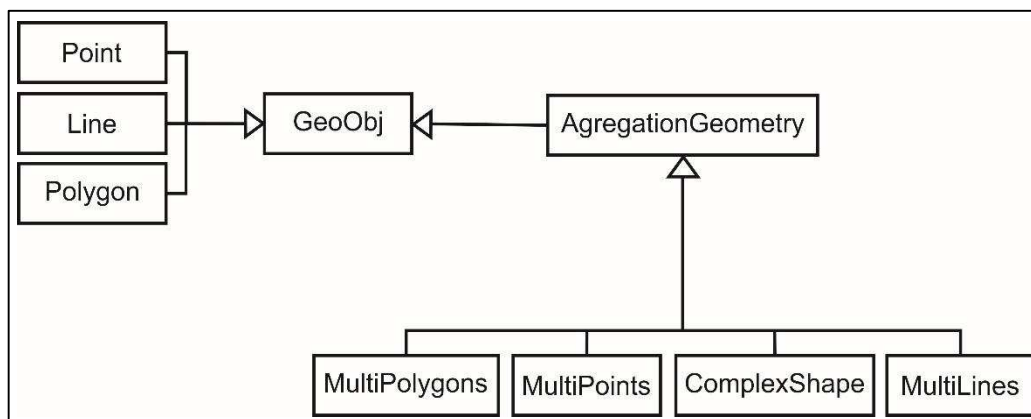


Figura 3.4: Metamodelo PVL representação dos tipos básicos de objetos espaciais.

Neste trabalho optou-se usar apenas uma metaclassa (*ComplexShape*) para representar a agregação de geometrias distintas. Desta forma, por exemplo, tanto o agregado de geometrias pontos e polígonos quanto o agregado de geometrias pontos e linhas serão representados pela metaclassa *ComplexShape*. Quando o aglomerado de geometrias também for desconhecido, *ComplexShape* também será usado para representação. Esta decisão foi tomada a partir de duas análises. A primeira refere-se ao Sistema Gerenciador de Banco de Dados (SGBD) PostGIS, que foi utilizado neste trabalho. Neste SGBD as possíveis formas de armazenamento de objetos complexos são: múltiplos pontos, múltiplos polígonos, múltiplas linhas e múltiplas geometrias. Desta forma, o armazenamento dos agregados de geometrias distintas são tratados do mesmo modo neste SGBD.

Dentre os modelos estudados só os modelos PVL e UML-GeoFrame oferecem construtores espaciais para representação de objetos complexos. Desta forma, a segunda análise refere-se à interoperabilidade com o modelo UML-GeoFrame. Um aglomerado de polígonos e linhas (☒) do PVL pode ser mapeado para um objeto complexo (☒) do UML-GeoFrame. O inverso também é possível, pois o PVL cobre todas as possibilidades de agregados de geometrias, no entanto não se sabe qual é o agregado de geometrias específico que está sendo representado em (☒) do UML-GeoFrame. Assim, um objeto complexo do UML-GeoFrame será mapeado para *ComplexShape* do PVL.

3.2.2 Especificação de metamodelos para representação de fenômenos na visão de campos contínuos

Segundo Pinet(2012) campos contínuos podem ser modelados de duas formas:

- Usando o conceito de atributos variantes no espaço. Uma função é usada para calcular a variável de interesse (ex.: f: ponto -> real) (GOMEZ, et al., 2010);
- Por meio de modelos de representação de campo como, por exemplo, TIN e Isolinhas.

Os modelos conceituais selecionados neste trabalho, que fornecem construtores para modelagem conceitual de campos contínuos são: UML-GeoFrame e OMT-G. Os dois modelos conceituais representam campos contínuos por meio de modelos de representação de campo.

A Figura 3.5 apresenta uma proposta do metamodelo do OMT-G para o requisito representação de campos contínuos. A metaclasse GeoField é utilizada para agrupar as possíveis representações de campos contínuos.

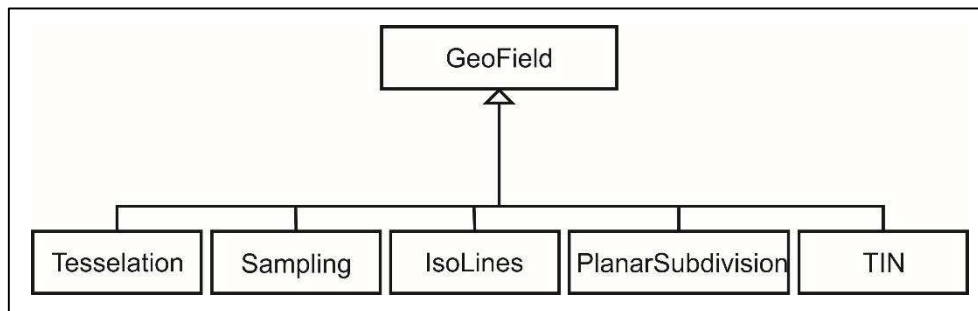


Figura 3.5: Metamodelo OMT-G para representação de campos contínuos.

A Figura 3.6 apresenta a proposta do metamodelo do UML-GeoFrame para o requisito representação de campos contínuos. Nesta figura um fenômeno na visão de campo (GeoField) pode ser representado por inúmeras representações de campos (FieldRepresentation). Um FieldRepresentation é um agregado de inúmeros objetos espaciais (SpatialObject). Um FieldRepresentation é especializado nos seguintes modelos de representação de campo: GridOfCells, AdjPolygons, Isolines, GridOfPoints, TIN e IrregularPoints.

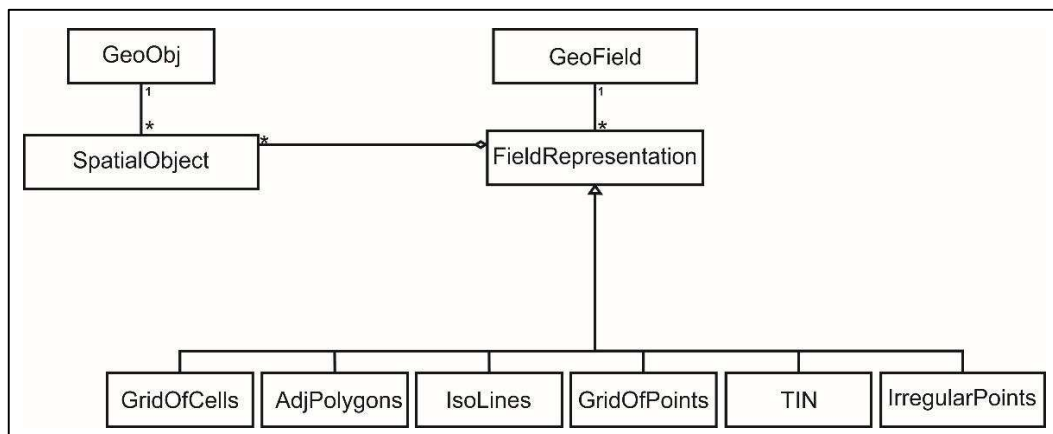





Figura 3.6: Metamodelo UML-GeoFrame para representação de campos contínuos.

3.2.3 Especificação de metamodelos para representação de redes

Dentre os quatro modelos conceituais selecionados neste trabalho, apenas o PVL não apresenta construtores para modelagem de fenômenos na visão de rede.

De acordo com Pinet (2012) existem propostas para formalizar estruturas de redes em diagramas. A proposta utilizada pelos modelos conceituais GeoOOA, UML-GeoFrame e OMT-G é a utilização de classes com ícones gráficos, para representação de nós e arcos de uma rede.

O modelo GeoOOA utiliza o pictograma  para representar a rede,  para representar os nós e  para representar os arcos. Cada classe nodo está ligada a fenômenos espaciais que podem ser representados por point ou Region. Cada classe link está ligada a fenômenos que podem ser representados por Line.

A Figura 3.7 mostra outra parte do metamodelo do GeoOOA. Este metamodelo pode ser utilizado para representação de fenômenos na visão de rede. A metaclass Network é utilizada para organizar o metamodelo. Network está ligada com Node e Link. Desta forma, ao modelar uma aplicação de rede esta estrutura deve ser mantida. Os tipos que Link e Node podem representar foram apresentados no metamodelo da Figura 3.1.

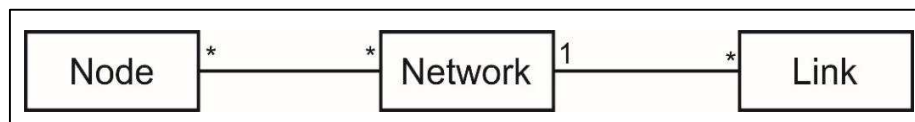


Figura 3.7: Metamodelo GeoOOA para representação de redes.

No modelo conceitual OMT-G, a representação de fenômenos de redes é feita através de objetos da classe Geo-Objeto. No entanto, essa representação é feita por objetos geométricos e topológicos: Node, UnidirectionalLine e BidirectionalLine. A Figura 3.8 mostra a parte do metamodelo do OMT-G para modelagem de rede. A metaclass Geometry and Topology foi definida para organizar os Geo-Objetos que apresentam geometrias e topologias.

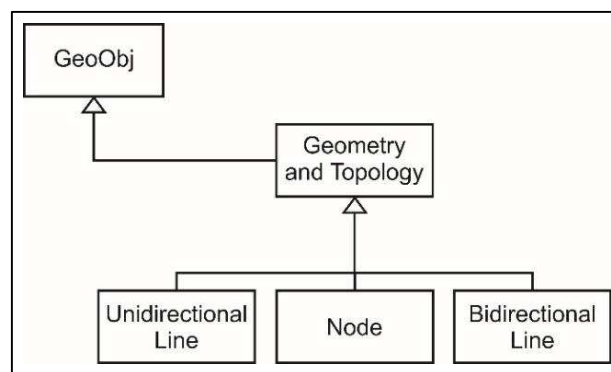


Figura 3.8: Metamodelo OMT-G para representação de Redes.

O Modelo UML-GeoFrame em sua proposta inicial não apresentava construtores para modelagem conceitual de fenômenos na visão de rede. Posteriormente, Stempliuć (2008)

propôs construtores para que o modelo UML-GeoFrame permitisse a modelagem conceitual desse tipo de fenômeno. A Figura 3.9 ilustra parte do metamodelo do UML-GeoFrame que atende o requisito de rede.

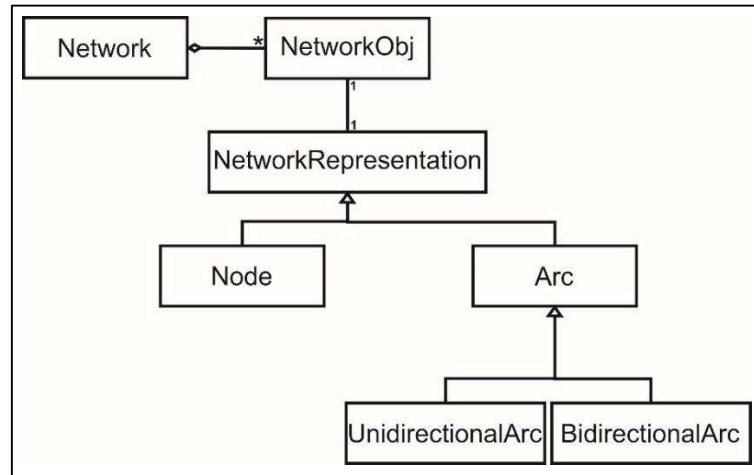


Figura 3.9: Metamodelo UML_GeoFrame para representação de Redes.

A proposta do metamodelo da Figura 3.9 está de acordo com a apresentada por Stempliu (2008). Uma rede (Network) é composta de diversos objetos de rede (NetworkObj). Cada objeto de rede oferece um tipo específico de representação, podendo ser um node ou um arco (Arc). Os arcos podem ser bi ou unidirecionais.

3.3 Etapa 3: Definição do Perfil UML e Restrições OCL

Nesta seção são especificados os Perfis UML e restrições em OCL. Os perfis UML em conjunto com as restrições em OCL oferecem aos projetistas recursos como construtores espaço/temporais e restrições de modelagem, semelhantes aos oferecidos pelos modelos conceituais. A especificação é feita com base nos metamodelos apresentados na seção 3.2.

3.3.1 Perfil GeoOOA

O perfil UML ilustrado na Figura 3.10 foi especificado de acordo com o metamodelo apresentado nas Figuras 3.1 e 3.7. Todos os estereótipos apresentados na Figura 3.10 são estendidos da metaclassa Class. O estereótipo GeoObj é abstrato, por isto é apresentado em itálico. O atributo `_image` colocado nos estereótipos não abstratos é utilizado para associar ícones gráficos aos estereótipos. O estereótipo Conventional foi acrescentado e é utilizado para modelar os objetos convencionais.

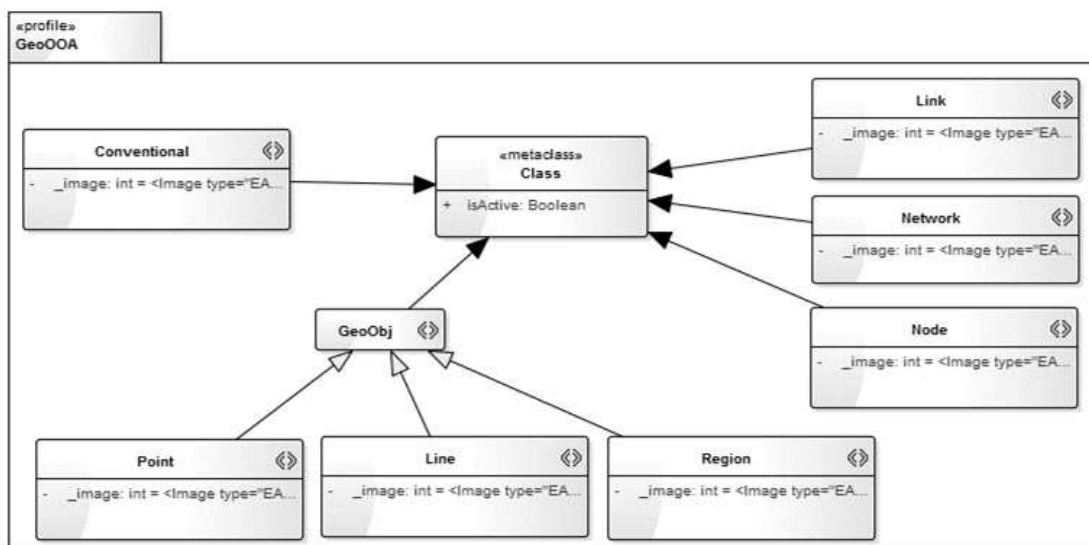


Figura 3.10: Perfil GeoOOA

Tabela 3.2: Restrições contra o uso de mais de um estereótipo por classe no perfil GeoOOA

Código	Restrição em OCL
01	context GeoObj inv: self.getAppliedStereotypes() -> select(s s.name = "Link" or s.name = "Network" or s.name = "Node" or s.name = "Conventional") -> isEmpty()
02	context Point inv: self.getAppliedStereotypes() -> select(s s.name = "Region" or s.name = "Line") -> isEmpty()
03	context Line inv: self.getAppliedStereotypes() -> select(s s.name = "Region" or s.name = "Point") -> isEmpty()
04	context Region inv: self.getAppliedStereotypes() -> select(s s.name = "Point" or s.name = "Line") -> isEmpty()
05	context Link inv: self.getAppliedStereotypes() -> select(s s.name = "Line" or s.name = "Point" or s.name = "Region" or s.name = "Conventional" or s.name = "Network" or s.name = "Node") -> isEmpty()
06	context Network inv: self.getAppliedStereotypes() -> select(s s.name = "Line" or s.name = "Point" or s.name = "Region" or s.name = "Conventional" or s.name = "Link" or s.name = "Node") -> isEmpty()
07	context Node inv: self.getAppliedStereotypes() -> select(s s.name = "Line" or s.name = "Point" or s.name = "Region" or s.name = "Conventional" or s.name = "Link" or s.name = "Network") -> isEmpty()

As restrições impostas ao metamodelo do GeoOOA podem ser impostas no Perfil GeoOOA utilizando comandos da linguagem OCL.

Ao utilizar a tecnologia de perfil da UML é possível associar diversos estereótipos a um elemento. Este tipo de representação não é adequado para o GeoOOA. Para evitar que o

projetista use este recurso em esquemas do GeoOOA profile, as restrições OCL apresentadas na Tabela 3.2 devem estar associadas aos estereótipos do GeoOOA profile.

Como especificado no metamodelo do GeoOOA, ilustrado na Figura 3.7, algumas precauções devem ser tomadas para modelagem conceitual de redes. Toda rede (Network) deve possuir pelo menos um Node e um Link. Todo nodo obrigatoriamente deve estar associado a uma rede. O nodo pode estar associado a mais de uma rede. Todo Link obrigatoriamente deve ser associado a uma única rede. Um nodo pode ser associado a objetos do tipo Point ou Region. Um link pode ser associado a objetos do tipo Line. Nodos e links não podem ser associados diretamente.

A Tabela 3.3 apresenta as restrições OCL a serem colocadas nos estereótipos do GeoOOA profile para evitar inconsistências na modelagem de redes.

Tabela 3.3: Restrições para evitar inconsistências na modelagem de redes do perfil GeOOA

Código	Restrição em OCL
01	context Network inv: not self.ownedAttribute.association.memberEnd.class.getAppliedStereotypes() -> select(s s.name = 'Node') -> isEmpty()
02	context Network inv: not self.ownedAttribute.association.memberEnd.class.getAppliedStereotypes() -> select(s s.name = "Link") -> isEmpty()
03	context Node inv: not self.ownedAttribute.association.memberEnd.class.getAppliedStereotypes() -> select(s s.name = "Network") -> isEmpty()
04	context Node inv: self.ownedAttribute.association.memberEnd.class.getAppliedStereotypes() -> select(s s.name = "Link") -> isEmpty()
05	context Link inv: not self.ownedAttribute.association.memberEnd.class.getAppliedStereotypes() -> select(s s.name = "Network") -> isEmpty()

3.3.2 Perfil OMT-G

O perfil UML ilustrado na Figura 3.11 foi especificado de acordo com o metamodelo apresentado nas Figuras 3.2, 3.5 e 3.8. Os estereótipos GeoObj, Geometry, Geometry and Topology e GeoField são abstratos, por isto são apresentados em itálico. O atributo *_image* colocado nos estereótipos não abstratos é utilizado para associar ícones gráficos aos estereótipos. Os estereótipos que estendem a metaclassa class foram definidos e explicados na especificação dos metamodelos.

Foi adicionado ao Perfil OMT-G o estereótipo NetworkAssociation. Este estereótipo estende da metaclassa association. Ele é utilizado no relacionamento dos estereótipos BidirectionalLine e/ou UnidirectionalLine com o estereótipo Node.

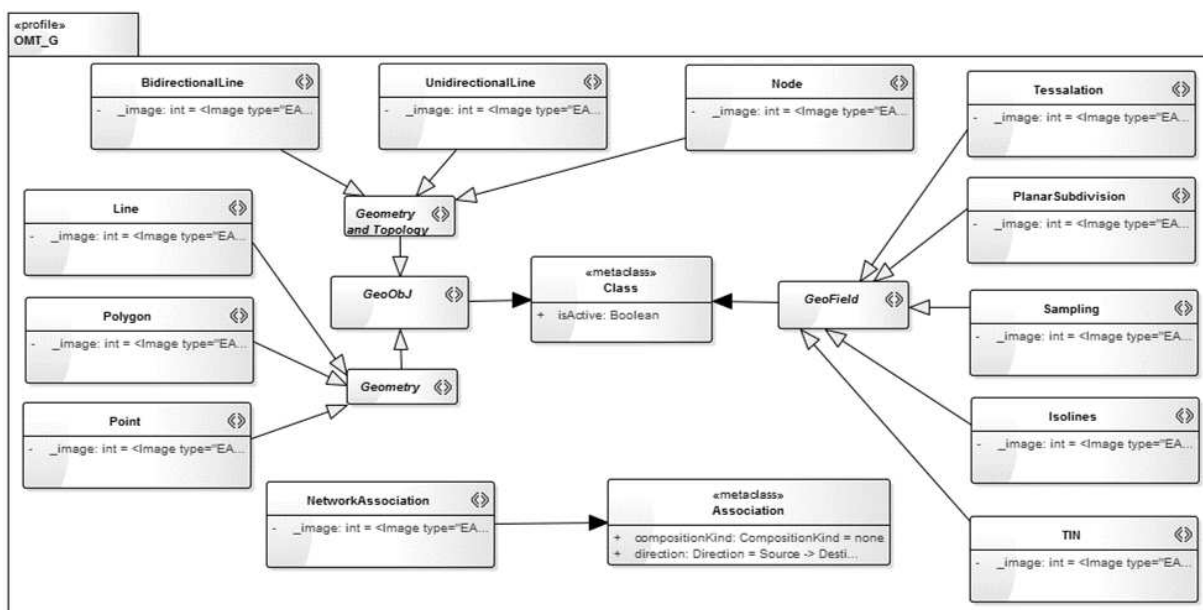


Figura 3.11: Perfil OMT-G

A tecnologia de Perfil UML permite associar diversos estereótipos a um elemento. Este tipo de representação não é adequado para o OMT-G. Desta forma, para que o projetista não venha utilizar este recurso em diagramas do OMT_G, restrições OCL devem ser definidas. A tabela 3.4 apresenta as restrições OCL a serem associadas aos estereótipos do profile OMT-G.

Tabela 3.4: Restrições contra o uso de mais de um estereótipo por classe no perfil OMT-G

Código	Restrições em OCL
01	context GeoObj inv: self.getAppliedStereotypes()-> select(s s.name = "Tesselation" or s.name = "PlanarSubdivision" or s.name = "Sampling" or s.name = "Isolines" or s.name = "TIN") -> isEmpty()
02	context Geometry inv: self.getAppliedStereotypes() -> select(s s.name = "BidirectionalLine" or s.name = "UnidirectionalLine" or s.name = "Node") -> isEmpty()
03	context Geometry and Topology inv: self.getAppliedStereotypes() -> select(s s.name = "Line" or s.name = "Polygon" or s.name = "Point") -> isEmpty()
04	context Line inv: self.getAppliedStereotypes() -> select(s s.name = "Point" or s.name = "Polygon") -> isEmpty()
05	context Point inv: self.getAppliedStereotypes() -> select(s s.name = "Line" or s.name = "Polygon") -> isEmpty()
06	context Polygon inv: self.getAppliedStereotypes() -> select(s s.name = "Line" or s.name = "Point") -> isEmpty()
07	context Node inv: self.getAppliedStereotypes() -> select(s s.name = "UnidirectionalLine" or s.name = "BidirectionalLine") -> isEmpty()

08	context UnidirectionalLine inv: self.getAppliedStereotypes() -> select(s s.name = "Node" or s.name = "BidirectionalLine") -> isEmpty()
09	context BidirectionalLine inv: self.getAppliedStereotypes() -> select(s s.name = "Node" or s.name = "UnidirectionalLine") -> isEmpty()
10	context Tesselation inv: self.getAppliedStereotypes() -> select(s s.name = "PlanarSubdivision" or s.name = "Sampling" or s.name = "Isolines" or s.name = "TIN") -> isEmpty()
11	context PlanarSubdivision inv: self.getAppliedStereotypes() -> select(s s.name = "Tesselation" or s.name = "Sampling" or s.name = "Isolines" or s.name = "TIN") -> isEmpty()
12	context Sampling inv: self.getAppliedStereotypes() -> select(s s.name = "Tesselation" or s.name = "PlanarSubdivision" or s.name = "Isolines" or s.name = "TIN") -> isEmpty()
13	context Isolines inv: self.getAppliedStereotypes() -> select(s s.name = "Tesselation" or s.name = "PlanarSubdivision" or s.name = "Sampling" or s.name = "TIN") -> isEmpty()
14	context TIN inv: self.getAppliedStereotypes() -> select(s s.name = "Tesselation" or s.name = "PlanarSubdivision" or s.name = "Sampling" or s.name = "Isolines") -> isEmpty()

Para a modelagem conceitual de redes com o Perfil OMT-G, algumas restrições devem ser impostas. O estereótipo NetworkAssociation deve ser usado como associação apenas entre os estereótipos de UnidirectionalLine e Node ou BidirectionalLine e Node. A Tabela 3.5 apresenta uma restrição que deve ser adicionada aos estereótipos do Perfil OMT-G.

Tabela 3.5: Restrições para evitar inconsistências na modelagem de redes no perfil OMT-G

Código	Restrição em OCL
01	context NetworkAssociation inv: let classes : OrderedSet(Class) = self.memberEnd.class -> asOrderedSet() in ((classes -> at(1)).getAppliedStereotypes() -> select(s s.name = "Node") -> notEmpty() and (classes -> at(2)).getAppliedStereotypes() -> select(s s.name = "BidirectionalLine") -> notEmpty()) or ((classes -> at(1)).getAppliedStereotypes() -> select(s s.name = Node) -> notEmpty() and (classes -> at(2)).getAppliedStereotypes() -> select(s s.name = UnidirectionalLine) -> notEmpty())

3.3.3 Perfil UML-GeoFrame

O perfil UML ilustrado na Figura 3.12 foi especificado de acordo com o metamodelo apresentado nas Figuras 3.3 e 3.6 e 3.9. Todos os estereótipos apresentados na Figura 3.12 são estendidos da metaclassa Class. O estereótipo GeographicObj, GeoField, GeoNetwork, SpatialObject e Arc são abstratos, por isto são apresentados em *itálico*. O atributo *_image*

colocado nos estereótipos não abstratos é utilizado para associar ícones gráficos aos estereótipos. O estereótipo ConventionalClass é utilizado para modelagem conceitual de objetos convencionais. Em esquemas no modelo conceitual UML-GeoFrame os objetos convencionais são representados por classes com o estereótipo gráfico representativo de objeto convencional.

Em esquemas no modelo UML-GeoFrame a múltipla representação é alcançada com a associação de vários estereótipos gráficos em uma classe. Todavia, existem estereótipos que não podem ser associados na mesma classe. Na tecnologia de Perfil esse processo é parecido, no entanto, o que são associados são os estereótipos. Para evitar que o projetista associe estereótipos incompatíveis na mesma classe, a Tabela 3.6 apresenta restrições OCL que devem ser associadas aos estereótipos do Perfil UML_GeoFrame.

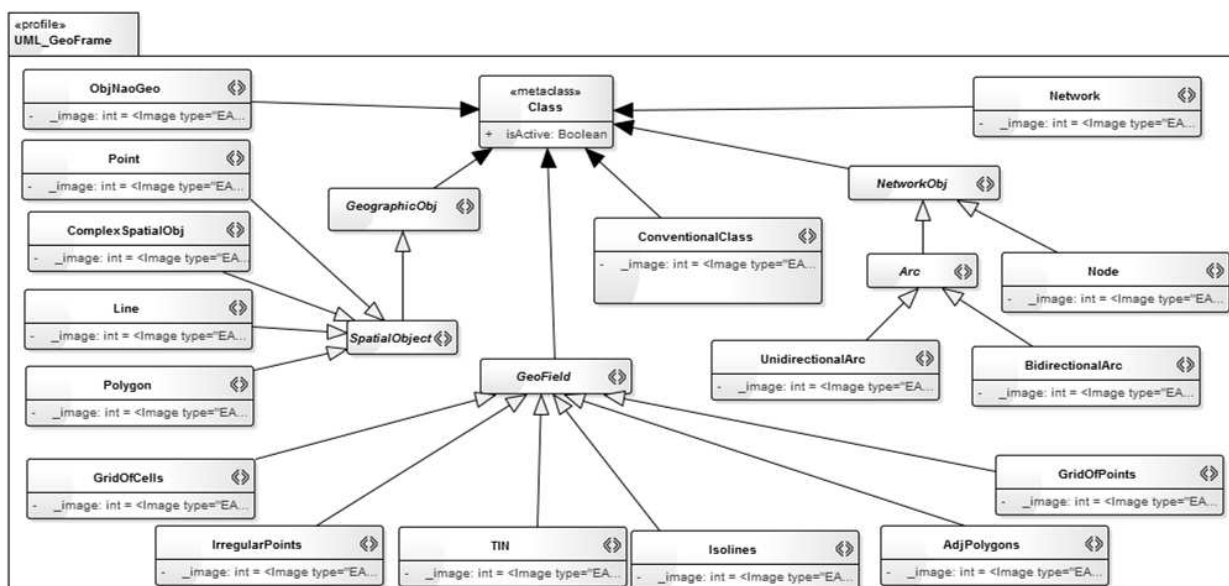


Figura 3.12: Perfil UML-GeoFrame.

Tabela 3.6: Restrições contra o uso estereótipos incompatíveis na mesma classe do perfil UML_GeoFrame.

Código	Restrições em OCL
01	context GeoField inv: self.getAppliedStereotypes() -> select(s s.name = "Point" or s.name = "Line" or s.name = "Polygon" or s.name = "ComplexSpatialObj" or s.name = "ConventionalObj") -> isEmpty()
02	context Network inv: self.getAppliedStereotypes() -> select(s s.name = "Point" or s.name = "Line" or s.name = "Polygon" or s.name = "ComplexSpatialObj" or s.name = "TIN" or s.name = "Isolines" or s.name = "GridOfCells" or s.name = "AdjPolygons" or s.name = "GridOfPoints" or s.name = "IrregularPoints" or s.name = "Node" or s.name = "UnidirectionalArc" or s.name = "BidirectionalArc" or s.name = "ConventionalObj") -> isEmpty()
03	context NetworkObj inv: self.getAppliedStereotypes() -> select(s s.name = "TIN" or s.name = "Isolines" or s.name = "GridOfCells" or s.name = "AdjPolygons" or s.name = "GridOfPoints" or s.name = "IrregularPoints" or s.name = "ConventionalObj") -> isEmpty()
04	context Node

	inv: self.getAppliedStereotypes() -> select(s s.name = "UnidirectionalArc" or s.name = "BidirectionalArc" or s.name = "ConventionalClass") -> isEmpty()
05	context UnidirectionalArc inv: self.getAppliedStereotypes() -> select(s s.name = "BidirectionalArc" or s.name = "ConventionalClass") -> isEmpty()

Para modelagem conceitual de redes com o UML-GeoFrame profile, algumas restrições devem ser impostas. Toda classe estereotipada por <<Network>> precisa estar associada a pelo menos um nodo <<Node>> e um arco <<BidirecionalArc>> ou <<UnidirectionalArc>>. Além disso, também é definido que todo nodo deve estar conectado a pelo menos um arco, e vice-versa.

A tabela 3.7 apresenta algumas restrições que devem ser adicionadas aos estereótipos do profile.

Tabela 3.7: Restrições para evitar inconsistências em redes do perfil UML_GeoFrame

Código	Restrição em OCL
01	context Network inv: not self.ownedAttribute.association.memberEnd.class .getAppliedStereotypes() -> select(s s.name = 'Node') -> isEmpty()
02	context Network inv: not self.ownedAttribute.association.memberEnd.class .getAppliedStereotypes() -> select(s s.name = 'UnidirectionalArc' or s.name = 'BidirectionalArc') -> isEmpty()
03	context Arc inv: not self.ownedAttribute.association.memberEnd.class .getAppliedStereotypes() -> select(s s.name = 'Node') -> isEmpty()
04	context Node inv: not self.ownedAttribute.association.memberEnd.class .getAppliedStereotypes() -> select(s s.name = 'UnidirectionalArc' or s.name = 'BidirectionalArc') -> isEmpty()

3.3.4 Perfil PVL

O perfil UML ilustrado na Figura 3.13 foi especificado de acordo com o metamodelo apresentado na Figura 3.4.

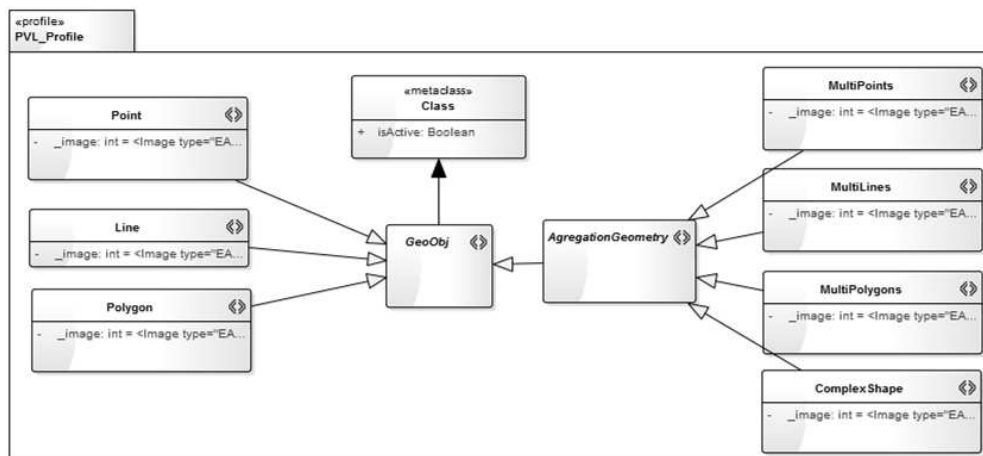


Figura 3.13: Perfil PVL.

Todos os estereótipos apresentados na Figura 3.13 são estendidos da metaclassa Class. O estereótipo <<GeoObj>> e <<AgregationGeometry>> são abstratos. O atributo `_image` colocado nos estereótipos não abstratos é utilizado para associar ícones gráficos aos estereótipos. Não é necessário criar um estereótipo para representação de objetos convencionais, para isso podem ser utilizadas as classes da UML.

Para o Perfil PVL, apresentado na Figura 3.13, não foram identificadas restrições.

3.4 Etapa 4: Aplicação e Configuração da Tecnologia MDG aos perfis UML

A tecnologia MDG na ferramenta EA promove a união de perfis UML, perfis toolboxes, scripts, imagens e outros componentes. O resultado após a utilização da tecnologia MDG é armazenado em um arquivo no formato XML.

Para especificação de perfis UML e restrições OCL foi feito um estudo exploratório em cada modelo conceitual. As seções anteriores apresentaram este estudo. Após a especificação de cada perfil UML, os próximos passos são: especificar os profiles toolboxes; a adição de imagens para representação gráfica; e a adição de scripts para modificação da forma da classe.

Os passos descritos no parágrafo anterior são de cunho técnico. Desta forma, foi feito um tutorial para mostrar como executar estes passos no perfil GeoProfile. A mesma ideia deve ser seguida para os perfis GeoOOA, OMT-G, UML-GeoFrame e PVL. Este tutorial encontra-se no Apêndice A.

Os scripts específicos de cada tecnologia para modificação da aparência e adição de imagens aos estereótipos são apresentados no Apêndice B. Para o desenvolvimento destes scripts o material de ajuda da EA foi consultado (EA, 2015).

Após passarem pelo processo da tecnologia MDG da ferramenta EA, os perfis UML GeoOOA, OMTG, UML_GeoFrame, PVL e GeoProfile foram nomeados por MDG_GeoOOA, MDG_OMTG, MDG-UML_GeoFrame, MDG-PVL e MDG_GeoProfile, respectivamente.

3.5 Etapa 5: Especificação de Regras de Transformação Horizontal

A tecnologia MDG do GeoProfile é usada como base de interoperabilidade entre as tecnologias MDG_GeoOOA, MDG_OMT-G, MDG_UML-GeoFrame e MDG_PVL.

As setas apresentadas na Figura 3.14 mostram o modo como os mapeamentos foram especificados. Todo esquema produzido pelas tecnologias MDG são inicialmente mapeados

para MDG-GeoProfile. Logo, todo esquema da MDG-GeoProfile pode ser mapeado para as outras tecnologias que possuem requisitos correspondentes.

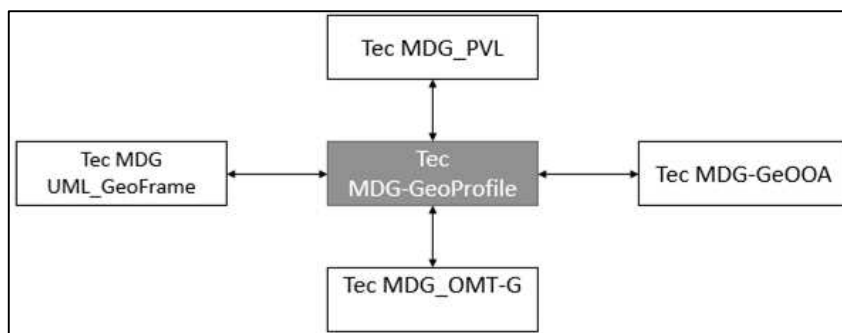


Figura 3.14: Mapeamento de Tecnologias.

A especificação de mapeamentos horizontais nesta seção é descrita por meio de pequenos esquemas conceituais de dados, ou esquemas conceituais incompletos (esquemas que não satisfazem toda exigência do domínio). Optou-se por propor novos construtores para o perfil GeoProfile quando constatado que os construtores da MDG_GeoProfile foram incapazes de representar os dos demais modelos.

3.5.1 Mapeamento entre MDG_GeoOOA e MDG_GeoProfile

Os requisitos de representação de tipos básicos de objetos espaciais na MDG_GeoOOA são modelados a partir de um retângulo de linha dupla, com ícones gráficos na parte superior direita dos retângulos. As geometrias espaciais consideradas pelo GeoOOA são Point, Line e Region.

Os requisitos de representação de tipos básicos de objetos espaciais na MDG_GeoProfile são apresentados a partir de uma classe com um estereótipo no formato de texto na parte superior e/ou pictogramas no canto direito superior.

A Figura 3.15 ilustra exemplos de classes de fenômenos na visão de objetos e os mapeamentos entre a MDG-GeoOOA e MDG_GeoProfile. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

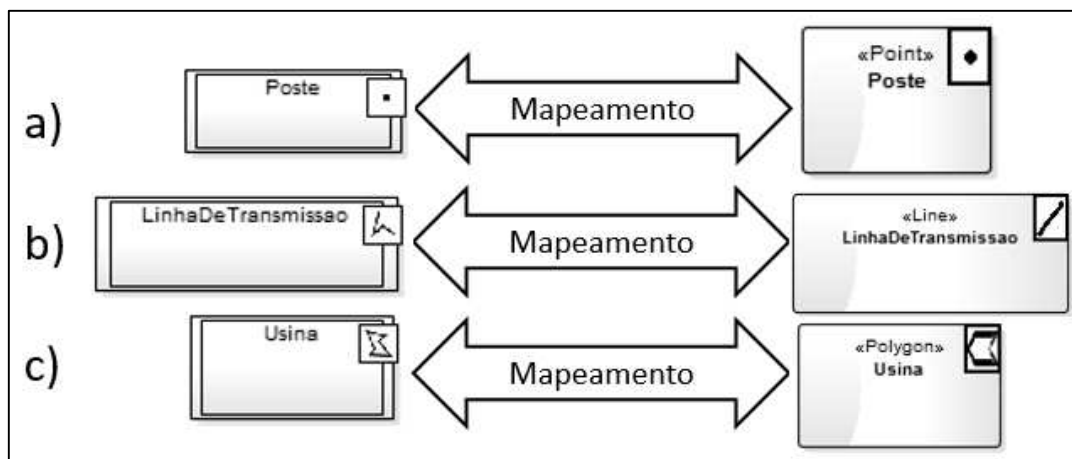


Figura 3.15: Especificação de mapeamento da representação de objetos básicos entre MDG_GeoOOA e MDG_GeoProfile.

Diferentemente da representação dos tipos de geometrias básicas da MDG_GeoOOA, os construtores de rede Node e Link são representados apenas por pictogramas no diagrama, o seja, não são colocados em retângulos. A Figura 3.16 mostra o mapeamento entre MDG_GeoOOA e MDG_GeoProfile contendo elementos de uma aplicação de rede de energia elétrica.

Na Figura 3.16-a o construtor Node está ligado a uma geometria do tipo Region. Desta forma o mapeamento deste esquema para a MDG_GeoProfile gera uma classe estereotipada com os estereótipos Node e Polygon. Se o construtor Node da MDG_GeoOOA estiver ligado com o construtor point ou com os construtores point e Region, é gerado respectivamente em MDG_GeoProfile uma classe estereotipada com <<node>> e <<point>> ou uma classe estereotipada com os estereótipos <<node>>, <<point>> e <<polygon>>. Se o construtor node não estiver ligado a nenhuma geometria é gerado em MDG_Geoprofile uma classe com apenas o estereótipo <<node>>.

A Figura 3.16-b ilustra um construtor Link ligado a um construtor Line. Este construtor pode ser mapeado para MDG_GeoProfile como uma classe com os estereótipos <<UnidirectionalArc>> e <<Line>>, ou para uma classe com os estereótipos <<BidirectionalArc>> e <<Line>>. Se o construtor Link não estiver ligado a nenhuma geometria é gerado em MDG_GeoProfile uma classe com o estereótipo <<UnidirectionalArc>> ou com o estereótipo <<BidirectionalArc>>. Para decidir se será utilizado o estereótipo <<UnidirectionalArc>> ou <<BidirectionalArc>> é necessária uma análise semântica por parte do projetista.

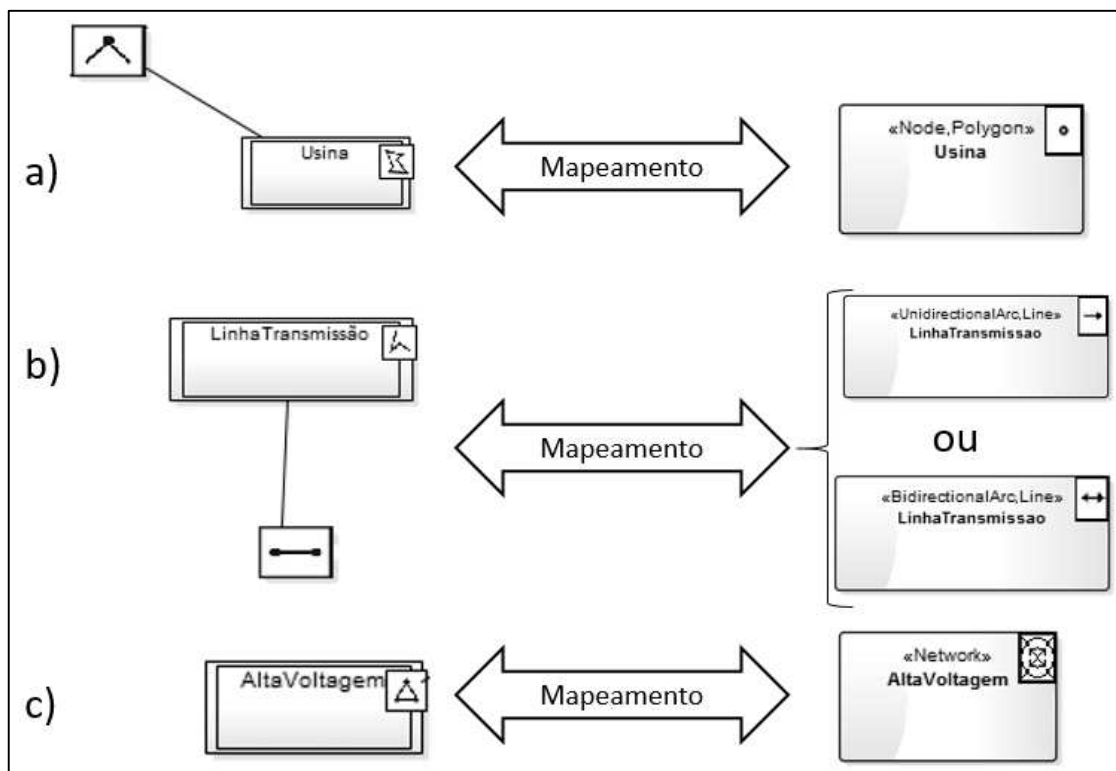


Figura 3.16: Especificação de mapeamento da representação de rede entre MDG_GeoOOA e MDG_GeoProfile.

A Figura 3.16-c apresenta um construtor de rede da MDG_GeoOOA. O mapeamento deste construtor para a MDG_GeoProfile é feito da mesma forma que o mapeamento dos tipos básicos de objetos espaciais apresentados na Figura 3.15.

3.5.2 Mapeamento entre MDG_OMT-G e MDG_GeoProfile

Os requisitos de representação de tipos básicos de objetos espaciais na MDG_OMT-G são modelados por classes, contendo ícones gráficos na parte superior direita. As geometrias espaciais consideradas pelo OMT-G são Point, Line e Polygon. Na parte esquerda da Figura 3.17 são apresentados os construtores da MDG_OMT-G. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

A ferramenta EA oferece recurso para ocultar os estereótipos textuais, no entanto, para melhor entendimento, os mesmos foram deixados na Figura 3.17.

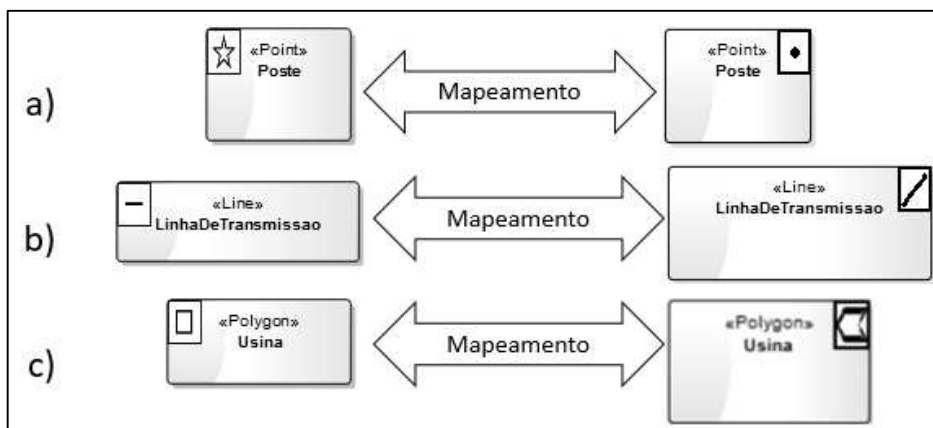


Figura 3.17: Especificação de mapeamento da representação de objetos básicos entre MDG_OMT-G e MDG_GeoProfile.

Os requisitos de representação de campos contínuos, da mesma forma que o requisito de representação de objetos básicos, na MDG_OMT-G são apresentados a partir de classes, contendo estereótipos gráficos na parte superior direita. As possíveis representações consideradas pelo OMT-G são: Isolines, PlanarSubdivision, Sampling, Tessellation e TIN. Estes construtores são mostrados no lado esquerdo da Figura 3.18. Alguns dos estereótipos da MDG_GeoProfile para representação de campos contínuos são apresentados no lado direito da Figura 3.18. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

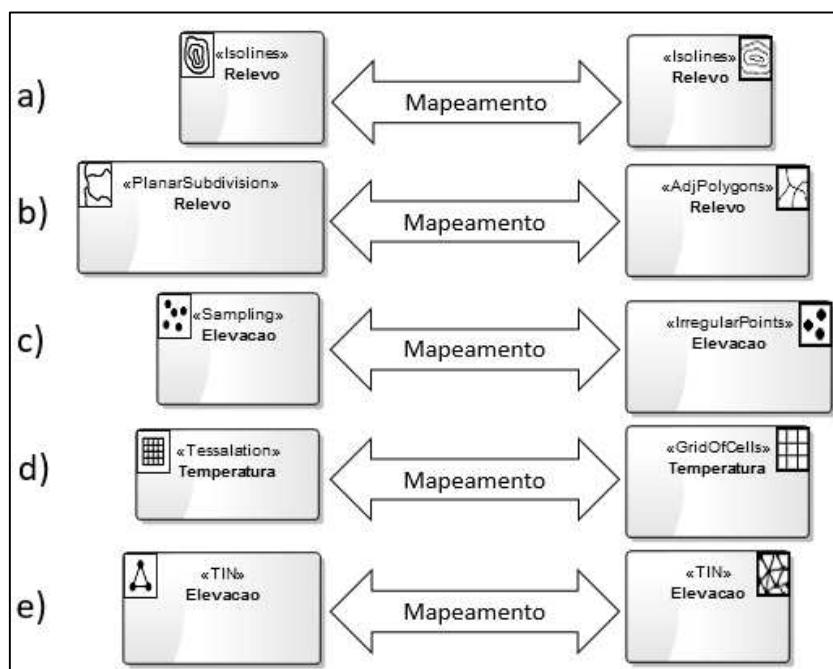


Figura 3.18: Especificação do mapeamento da representação de campos contínuos entre MDG_OMT-G e MDG_GeoProfile.

Para modelagem conceitual de esquemas que utilizam o requisito representação de rede, na MDG_OMT-G foi definido que os estereótipos gráficos seriam colocados no lado superior esquerdo da classe. No OMT-G os construtores utilizados para representação de rede são: BidirectionalLine, UnidirectionalLine e Node. Na MDG_GeoProfile foi definido que os estereótipos gráficos para representação do requisito de redes seriam colocados na parte superior direita da classe. No GeoProfile os construtores utilizados para representação de rede são Network, UnidirectionalArc, BidirectionalArc e Node. A Figura 3.19 ilustra a especificação do mapeamento dos construtores de rede entre MDG_OMT-G e MDG_GeoProfile.

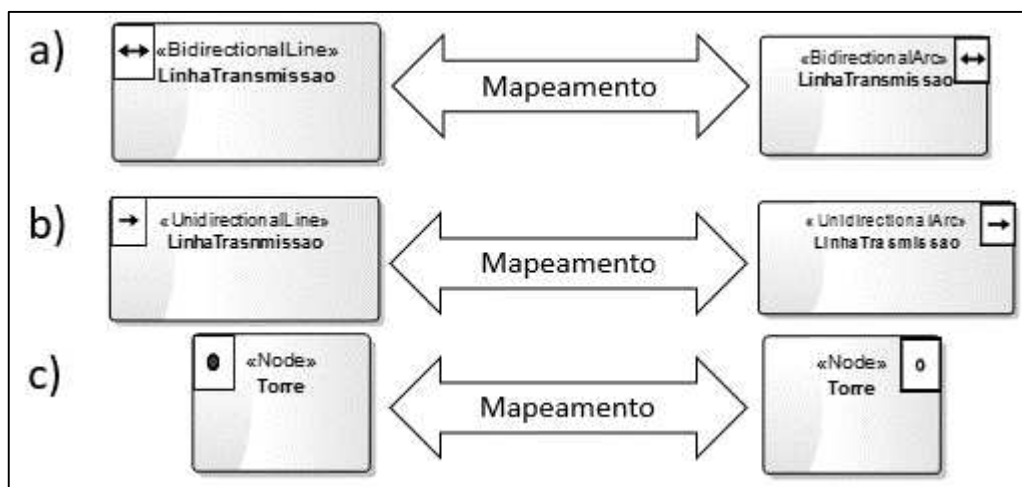


Figura 3.19: Especificação do mapeamento da representação de redes entre Tec MDG_OMT-G e Tec MDG_GeoProfile.

Na MDG_OMT-G o nodo da rede e as linhas bidirecionais ou unidirecionais são ligadas por uma associação denominada de networkAssociation. A restrição OCL para NetworkAssociation foi definida na seção 3.3.2. Na MDG_GeoProfile o nodo da rede e as linhas bidirecionais ou unidirecionais são ligadas por uma classe, nomeada de network, com um ícone gráfico no canto superior direito. A Figura 3.20 apresenta o mapeamento do requisito de rede entre MDG_OMT-G e MDG_GeoProfile.

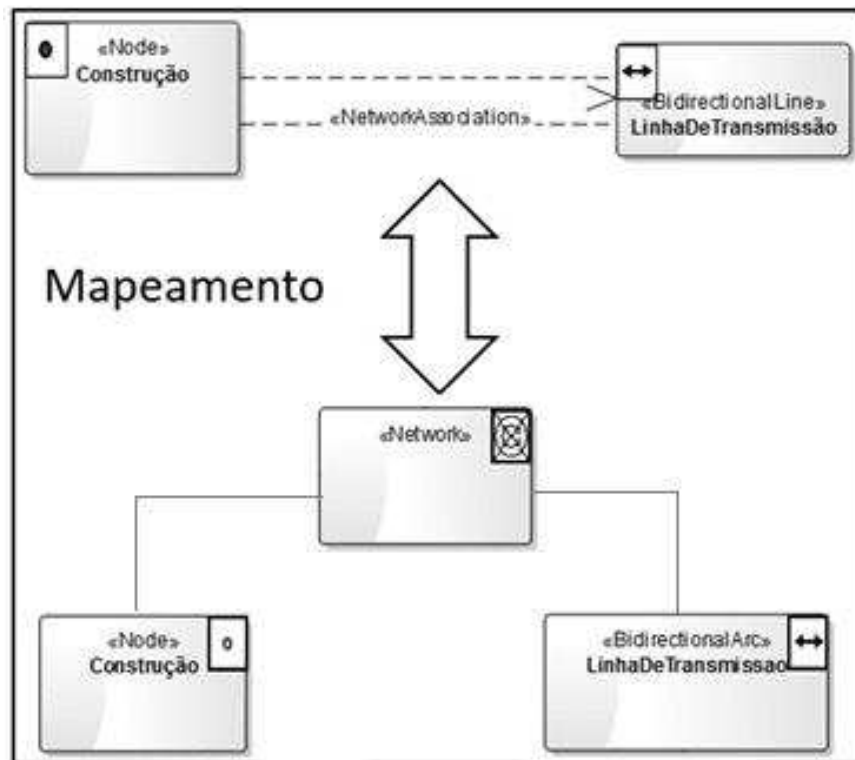


Figura 3.20: Especificação do mapeamento de rede entre MDG_OMT-G e MDG_GeoProfile.

3.5.3 Mapeamento entre MDG_UML-GeoFrame e MDG_GeoProfile

Os requisitos de representação de tipos básicos de objetos espaciais na MDG_UML-GeoFrame são modelados por classes, contendo ícones gráficos na parte superior direita e esquerda. O ícone gráfico da parte superior esquerda indica que a classe representa um objeto espacial. O ícone gráfico na parte superior direita indica o tipo de objeto espacial. As geometrias espaciais consideradas pela MDG-UML-GeoFrame são Point, Line, Polygon e ComplexSpatialObj. A Figura 3.21 ilustra o mapeamento entre as MDG_UML-GeoFrame e MDG-GeoProfile. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

Os requisitos de representação de campos contínuos, de igual modo ao requisito de representação de objetos básicos, na MDG_UML-GeoFrame são apresentados a partir classes, seguidas de estereótipos gráficos na parte superior direita e esquerda. O ícone gráfico colocado na parte superior esquerda indica que a classe representa um campo contínuo. Os ícones da parte superior direita indicam o tipo de representação de campo contínuo que está sendo utilizada. As possíveis representações consideradas pelo UML-GeoFrame são: Isolines, AdjPolygons, IrregularPoints, GridOfCells, TIN e GridOfPoints. A Figura 3.22 apresenta a especificação do mapeamento do requisito de campos contínuos entre

MDG_UML_GeoFrame e MDG_GeoProfile. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

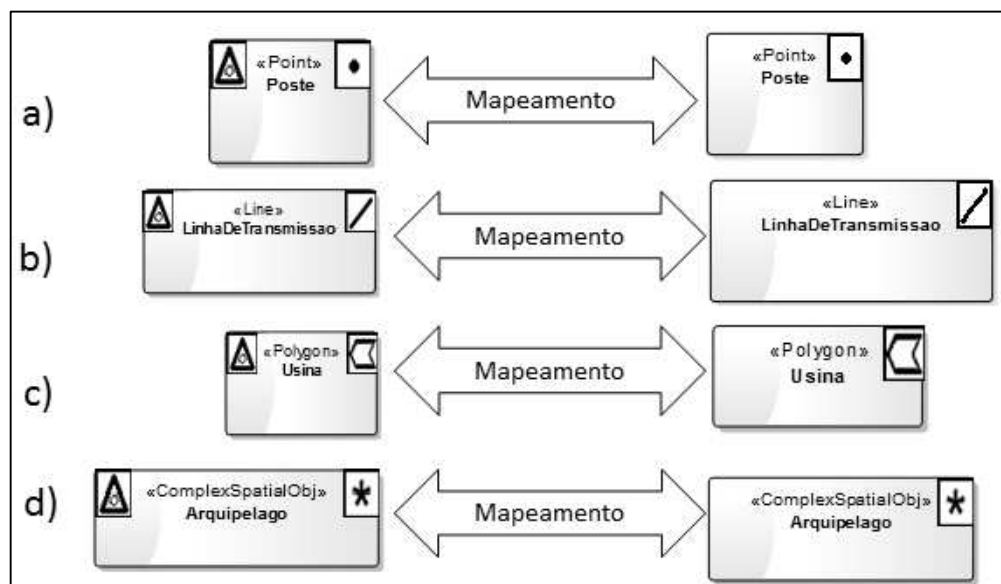


Figura 3.21: Especificação de mapeamento da representação de objetos básicos entre MDG_UML_GeoFrame e MDG_GeoProfile.

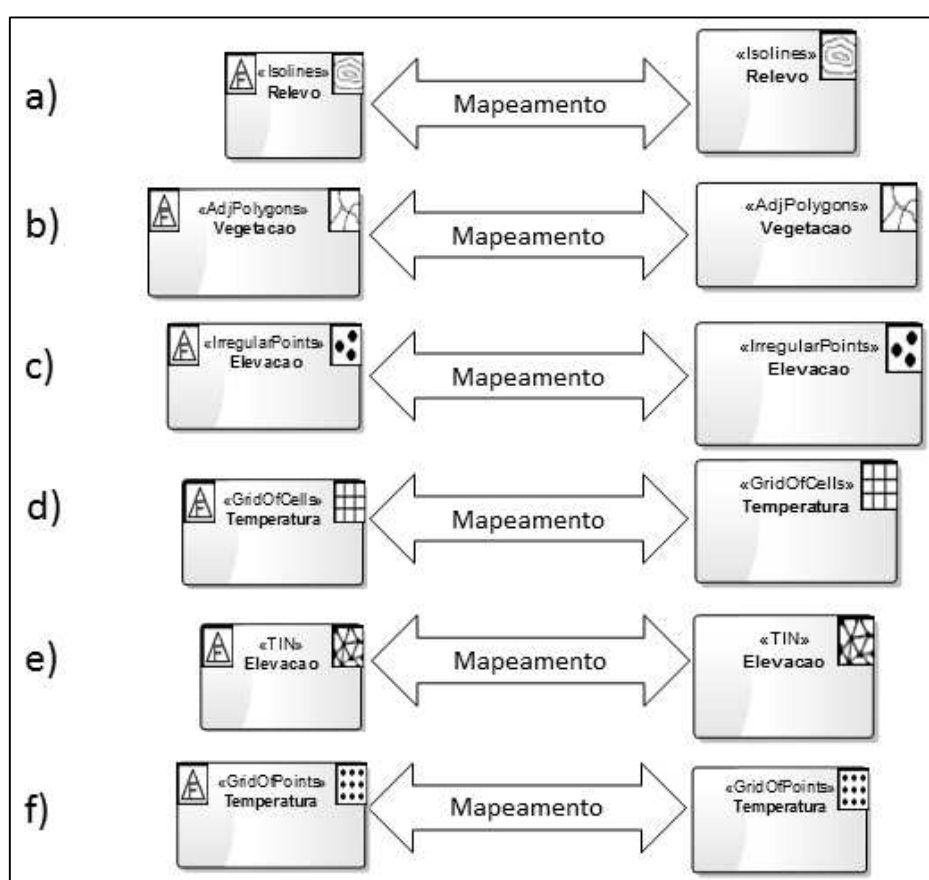


Figura 3.22: Especificação do mapeamento da representação de campos contínuos entre MDG_UML-GeoFrame e MDG_GeoProfile.

Para a modelagem conceitual de esquemas conceituais que utilizam o requisito representação de rede, na MDG_UML-GeoFrame foi definido que os estereótipos gráficos para representação de Node, BidirectionalArc e UnidirectionalArc sejam colocados no lado superior direito da classe. No lado superior esquerdo da classe é colocado um ícone gráfico para identificação de rede.

Na MDG_UML_GeoFrame o nodo da rede e as linhas bidirecionais ou unidirecionais são associadas a uma classe com o estereótipo Network. Esta classe pode conter dados alfanuméricos sobre a rede.

A Figura 3.23 apresenta o mapeamento do requisito de rede entre MDG_UML_GeoFrame e MDG_GeoProfile.

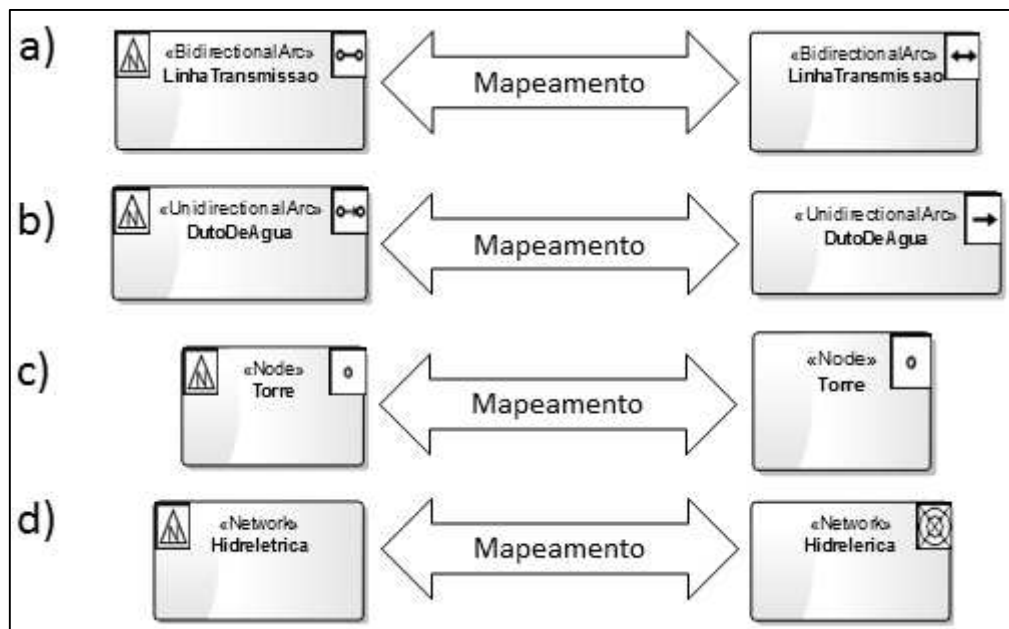


Figura 3.23: Especificação do mapeamento da representação de redes entre MDG_UML_GeoFrame e MDG_GeoProfile.

3.5.4 Mapeamento entre MDG_PVL e MDG_GeoProfile

Os requisitos de representação de tipos básicos de objetos espaciais na MDG_PVL são modelados por classes, contendo ícones gráficos na parte superior direita. Este ícone indica o tipo básico de objeto espacial que está sendo representado na classe. As geometrias espaciais consideradas pela MDG-PVL são Point, Line, Polygon, MultiPolygons, MultiPoints, MultiLines e ComplexShape.

Para as geometrias espaciais Point, Line, Polygon e ComplexShape o GeoProfile possui estereótipos correspondentes. A Figura 3.24 apresenta o mapeamento destas geometrias espaciais entre MDG_PVL e MDG_GeoProfile. Este mapeamento é feito de

forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

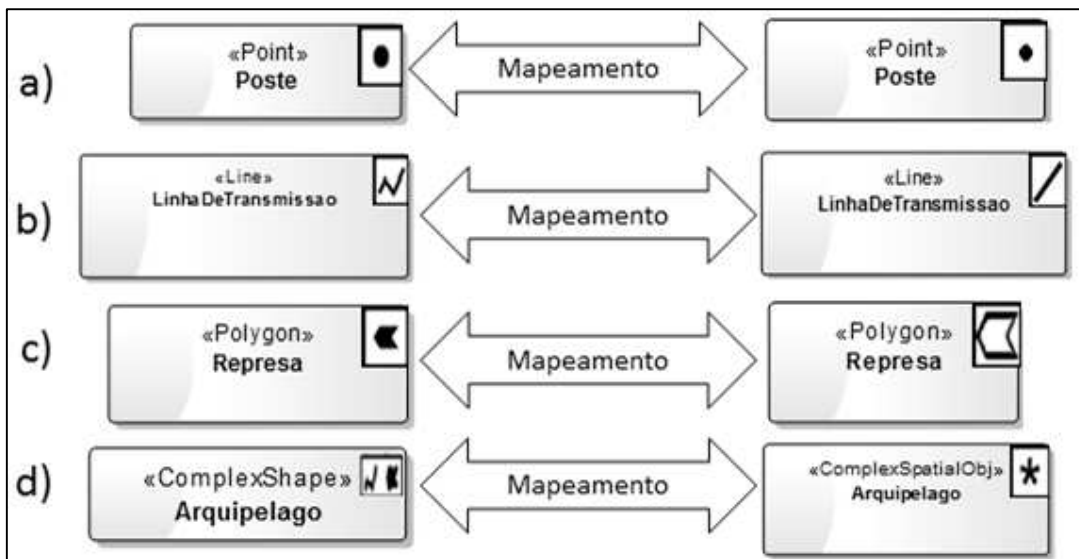


Figura 3.24: Especificação de mapeamento da representação de objetos básicos entre MDG_PVL e Tec MDG_GeoProfile.

O GeoProfile não apresenta construtores capazes de representar os tipos geométricos MultiPolygons, MultiPoints, MultiLines do modelo PVL. Desta forma, a seção 3.6 apresenta uma proposta de extensão do GeoProfile.

3.6 Etapa 6: Proposta de extensão do GeoProfile

A partir dos mapeamentos apresentados na seção 3.5 deste trabalho, foi constatado que o GeoProfile não possui construtores correspondentes para os tipos básicos de objetos MultiPolygons, MultiPoints e MultiLines do modelo conceitual PVL. Desta forma, esta seção descreve uma proposta de extensão do GeoProfile. Também é apresentada a especificação dos mapeamentos horizontais entre MDG-PVL e MDG_GeoProfile.

A Figura 3.25 ilustra parte do metamodelo do GeoProfile contemplando a proposta de extensão. Os tipos básicos MultiPoints, MultiLines e MultiPolygons que foram incluídos no metamodelo do GeoProfile estão de acordo com os padrões Open Geospatial Consortium (OGC). Estes tipos básicos são considerados agregações simples.

O tipo ComplexSpatialObj, que já existia no GeoProfile para modelagem de todos os tipos de agregados de geometria, a partir desta especificação deve ser utilizado para representação de agregados de geometrias complexas, ou seja agregado de geometrias com tipos diferentes.

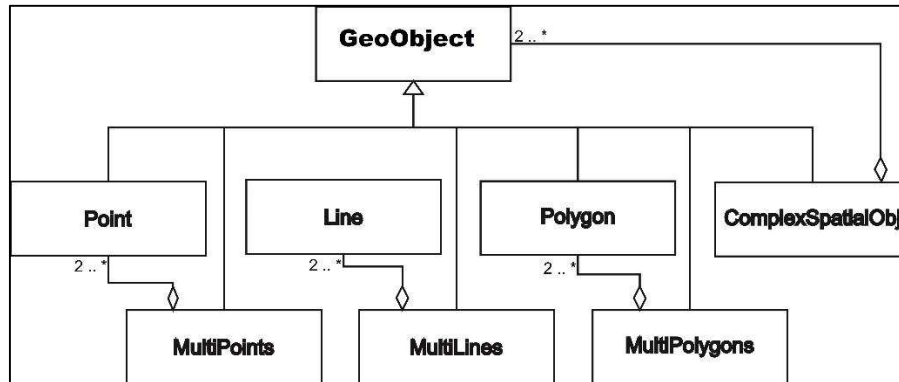


Figura 3.25: Extensão do metamodelo do GeoProfile.

A partir da Figura 3.25 foi estendido o Perfil UML para esta proposta. A Figura 3.26 apresenta parte do Perfil UML GeoProfile. Os novos elementos inclusos no metamodelo do GeoProfile foram considerados estereótipos no Perfil UML GeoProfile.

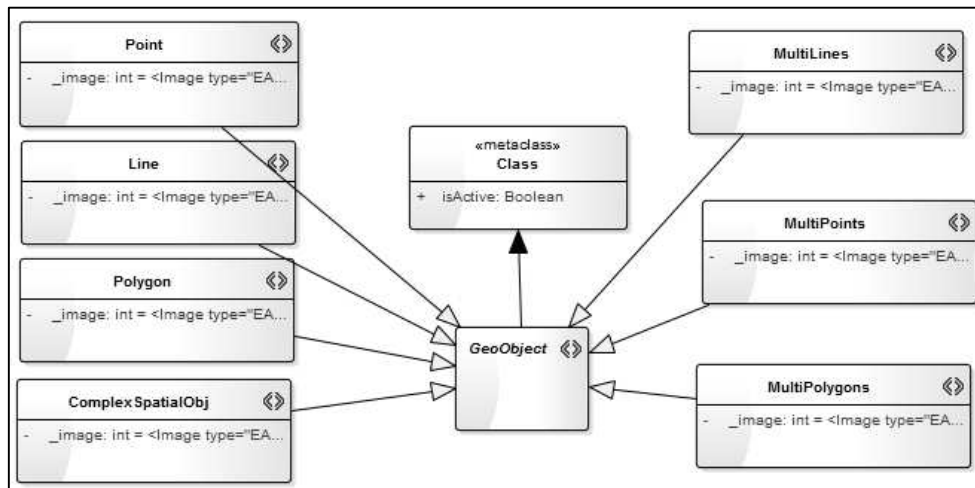


Figura 3.26: Extensão do metamodelo do GeoProfile.

Os ícones gráficos , e foram usados respectivamente para representação dos estereótipos MultiLines, MultiPoints e MultiPolygons. O simbolo de * dentro de cada pictograma é o mesmo utilizado para representação de objetos complexos (conjunto de pontos, linhas e polígonos) do GeoProfile, desta forma, ao associar o asterisco com um tipo básico de objeto espacial (linha, ponto ou polígono) e colocar no canto superior direito da classe, entende-se que a classe representa um conjunto de linhas, pontos ou polígonos.

Tratando-se de estereótipos que representam agregados de formas geométricas, não faz sentido permitir a multirepresentação entre os esterótipos ComplexShapeObj, Multilines, MultiPolygons e MultiPoints. Desta forma, restrições em OCL foram especificadas e colocadas nas classes espaciais. Além das restrições OCL herdadas do estereótipo GeoObject (SAMPAIO, 2009), as restrições apresentadas na Tabela 3.8 foram acrescentadas.

Tabela 3.8: Restrições contra o uso estereótipos incompatíveis na mesma classe do GeoProfile estendido.

Código	Restrição em OCL
01	context ComplexSpatialObj inv: self.getAppliedStereotypes() -> select(s s.name = "Multilines" or s.name = "MultiPolygons" or s.name = "MultiPoints" or s.name = "Point" or s.name = "Polygon" or s.name = "Line")) -> isEmpty()
02	context MultiPoints inv: self.getAppliedStereotypes() -> select(s s.name = "Multilines" or s.name = "MultiPolygons" or s.name = "ComplexSpatialObj" or s.name = "Point" or s.name = "Polygon" or s.name = "Line")) -> isEmpty()
03	context MultiLines inv: self.getAppliedStereotypes() -> select(s s.name = "MultiPoints" or s.name = "MultiPolygons" or s.name = "ComplexSpatialObj" or s.name = "Point" or s.name = "Polygon" or s.name = "Line")) -> isEmpty()
04	context MultiPolygons inv: self.getAppliedStereotypes() -> select(s s.name = "MultiPoints" or s.name = "MultiLines" or s.name = "ComplexSpatialObj" or s.name = "Point" or s.name = "Polygon" or s.name = "Line")) -> isEmpty()
05	context Point inv: self.getAppliedStereotypes() -> select(s s.name = "MultiPoints" or s.name = "MultiLines" or s.name = "MultiPolygons" or s.name = "ComplexSpatialObj")) -> isEmpty()
06	context Polygon inv: self.getAppliedStereotypes() -> select(s s.name = "MultiPoints" or s.name = "MultiLines" or s.name = "MultiPolygons" or s.name = "ComplexSpatialObj")) -> isEmpty()
07	context Line inv: self.getAppliedStereotypes() -> select(s s.name = "MultiPoints" or s.name = "MultiLines" or s.name = "MultiPolygons" or s.name = "ComplexSpatialObj")) -> isEmpty()

3.6.1 Mapeamento entre MDG_PVL e MDG-GeoProfile Estendido

Com a extensão do Geoprofile tornou-se possível o mapeamento horizontal entre os modelos PVL e GeoProfile. A Figura 3.27 apresenta a especificação de mapeamento entre as duas tecnologias. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice C.

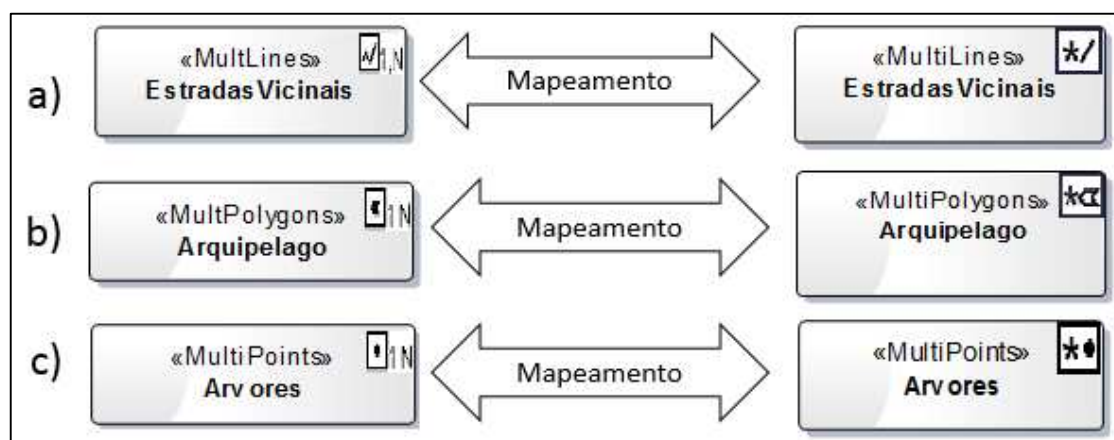


Figura 3.27: Especificação de mapeamento da representação de objetos básicos entre MDG_PVL e MDG_GeoProfile estendido.

3.7 Etapa 7: Especificação de Regras de Transformações Verticais

Esta seção mostra, por meio de exemplos, a interoperabilidade vertical usando transformações entre níveis da abordagem MDA. Neste trabalho os níveis considerados são PIM, PSM e código fonte. Os scripts de código fonte gerados neste trabalho podem ser utilizados para criação de BDG no SGBD PostGIS.

As subseções a seguir apresentam respectivamente o mapeamento vertical das tecnologias MDG_GeoOOA, MDG_OMTG, MDG_UML_GeoFrame e MDG_PVL.

3.7.1 Mapeamento vertical MDG_GeoOOA

No mapeamento vertical dos requisitos de representação de tipos básicos de objetos espaciais na MDG_GeoOOA (PIM) os estereótipos Point, Line e Region são transformados em colunas de tabelas com tipos de dados geográficos específicos do SGBD com suporte a objetos espaciais, no caso o PostGIS. Todos os atributos das classes (PIM) são convertidos em colunas de tabelas. No ato da transformação essas tabelas também recebem uma coluna para chave primária com nome da tabela + ID e com o tipo de dado integer. Desta forma, um esquema PSM é gerado.

A ferramenta EA também apresenta recursos para a transformação de esquemas PSM em códigos DDL. Este tipo de transformação não pode ser modificado na ferramenta.

A Figura 3.28 ilustra a transformação dos construtores Point, Line e Region da tecnologia MDG_GeoOOA. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice D.

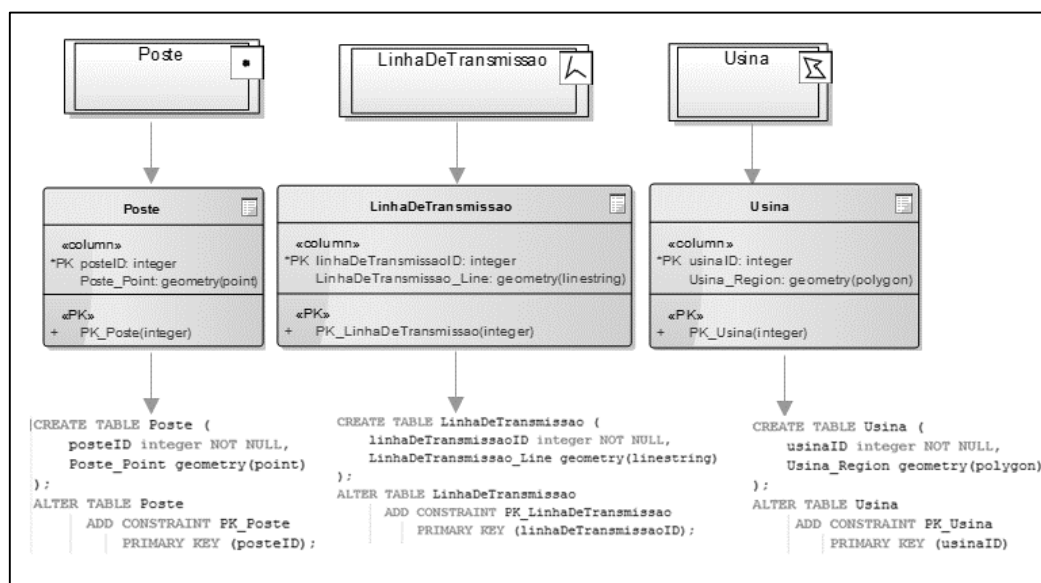


Figura 3.28: Especificação de mapeamento vertical da representação de objetos básicos da tecnologia MDG_GeoOOA.

A MDG_GeoOOA foi especificada neste trabalho para ser utilizada tanto na produção de esquemas conceituais que utilizem construtores de objetos básicos, quanto esquemas conceituais que utilizem construtores de rede.

Os construtores de rede Link e Node da tecnologia MDG_GeoOOA são convertidos em colunas de tabelas no nível PSM. No PostGIS essas colunas recebem o tipo de dado Geometry. Neste trabalho esse tipo de dado está sendo usado para armazenar dados sobre topologias. Para implementação, os algoritmos de rede discutidos em (STEMPLIUC2008) devem ser levados em consideração.

A Figura 3.29 ilustra a transformação vertical dos construtores de rede. Para os construtores de Node e Link foram feitas associações com os construtores básicos Region Line. O mapeamento vertical do construtor Network é o mesmo utilizado para o mapeamento de classes convencionais.

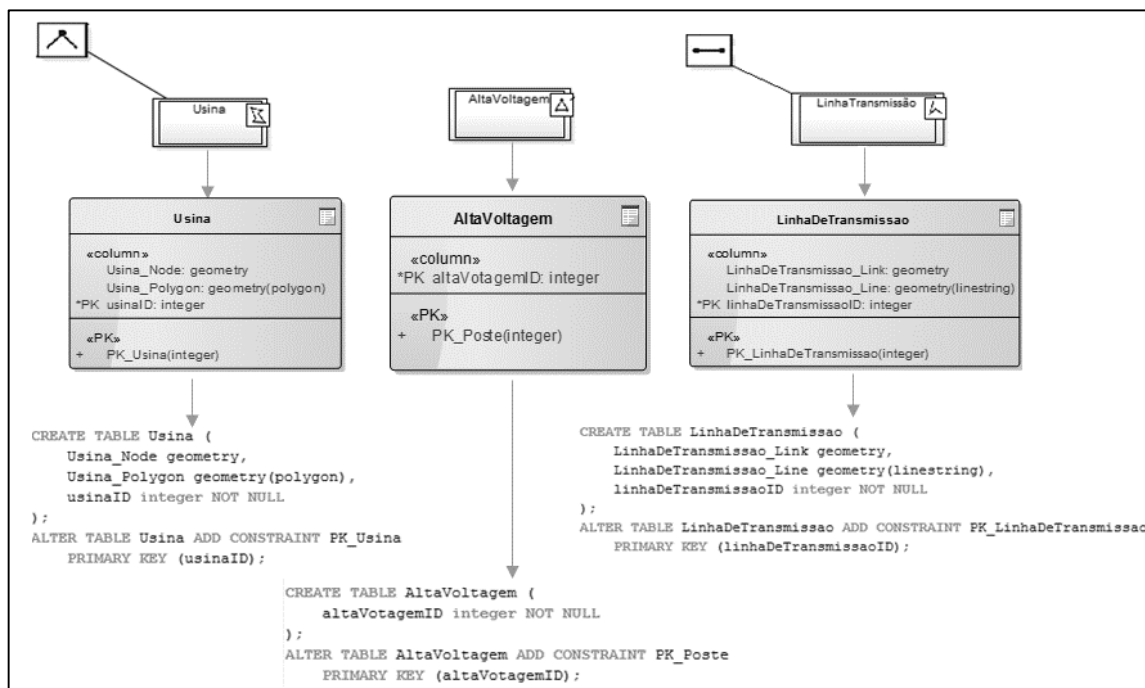


Figura 3.29: Especificação de mapeamento vertical da representação de redes da tecnologia MDG_GeoOOA.

3.7.2 Especificação do mapeamento vertical MDG_OMTG

No mapeamento vertical dos requisitos de representação de tipos básicos de objetos espaciais na MDG_OMTG (PIM) os estereótipos Point, Line e Polygon são transformados em colunas de tabelas com tipos de dados geográficos específicos do SGBD PostGIS. Todos atributos das classes (PIM) são convertidos em colunas de tabelas. A Figura 3.30 apresenta o mapeamento vertical dos construtores Point, Line e Polygon. Este mapeamento é feito de

forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice D.

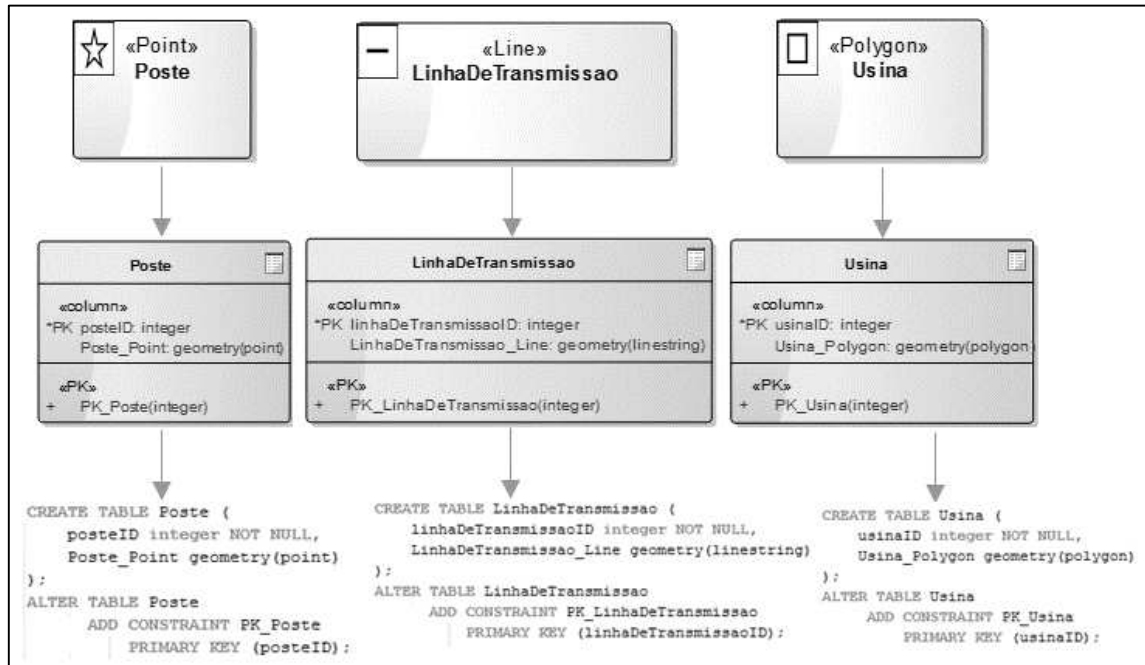


Figura 3.30: Especificação de mapeamento vertical da representação de objetos básicos da tecnologia MDG_OMTG.

No mapeamento vertical dos requisitos de representação de campos contínuos na MDG_OMTG (PIM) os estereótipos Isolines, PlanarSubdivision, Sampling, Tesselation, e TIN, são transformados em colunas de tabelas seguidas do tipo de dados específico (Martínez, 2014). Todos os atributos das classes (PIM) são convertidos em colunas de tabelas (PSM). A Figura 3.31 apresenta o mapeamento vertical dos construtores Isolines, PlanarSubdivision, Sampling, Tesselation e TIN. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice D.

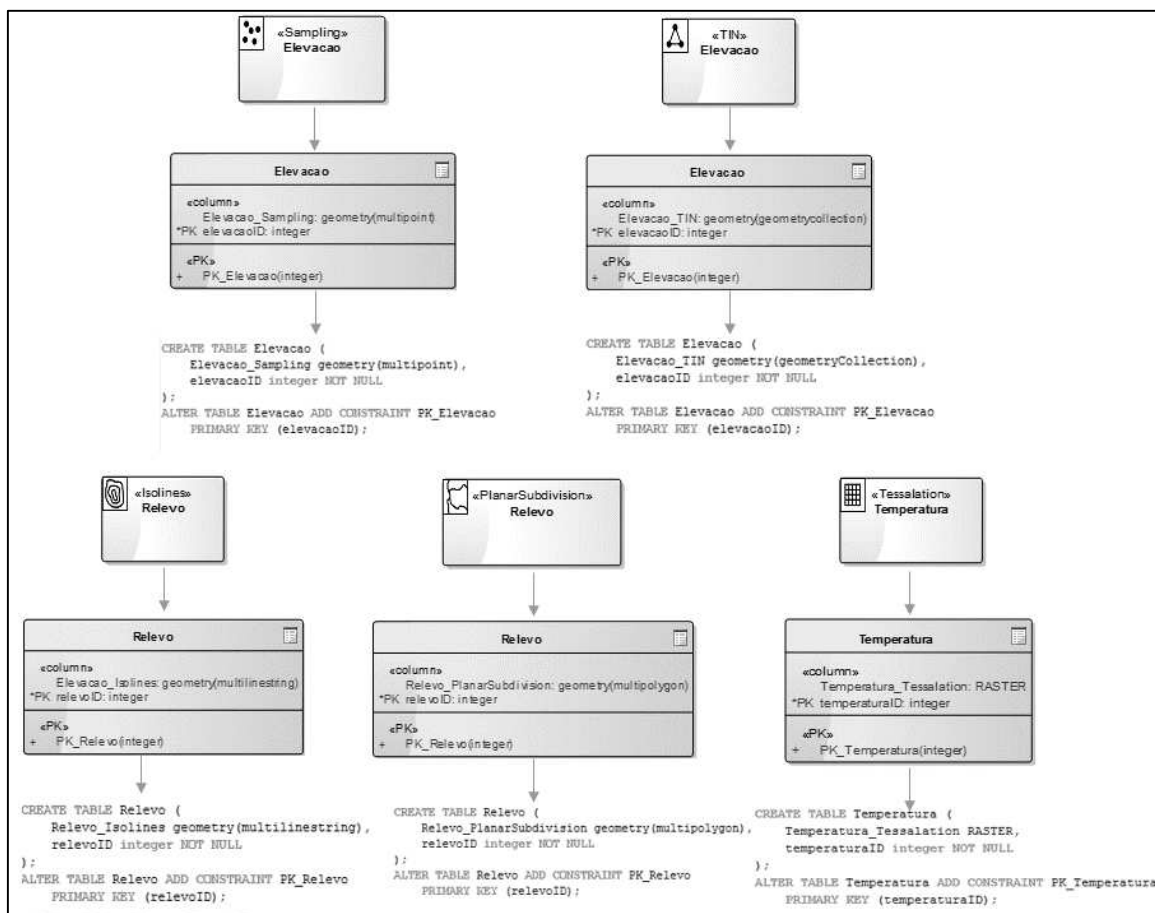


Figura 3.31: Especificação de mapeamento vertical da representação de campos contínuos da tecnologia MDG_OMTG.

A MDG_OMTG foi especificada neste trabalho para ser utilizada na produção de esquemas conceituais que utilizem construtores de objetos básicos, construtores de campos contínuos e construtores de rede. Os construtores de rede Node, UnidirectionalLine e BidirectionalLine da tecnologia MDG_OMTG são convertidos em colunas de tabelas no nível PSM.

A Figura 3.32 apresenta o mapeamento vertical dos construtores Node, UnidirectionalLine e BidirectionalLine utilizados para representação de esquemas conceituais na visão de rede da tecnologia MDG_OMTG.

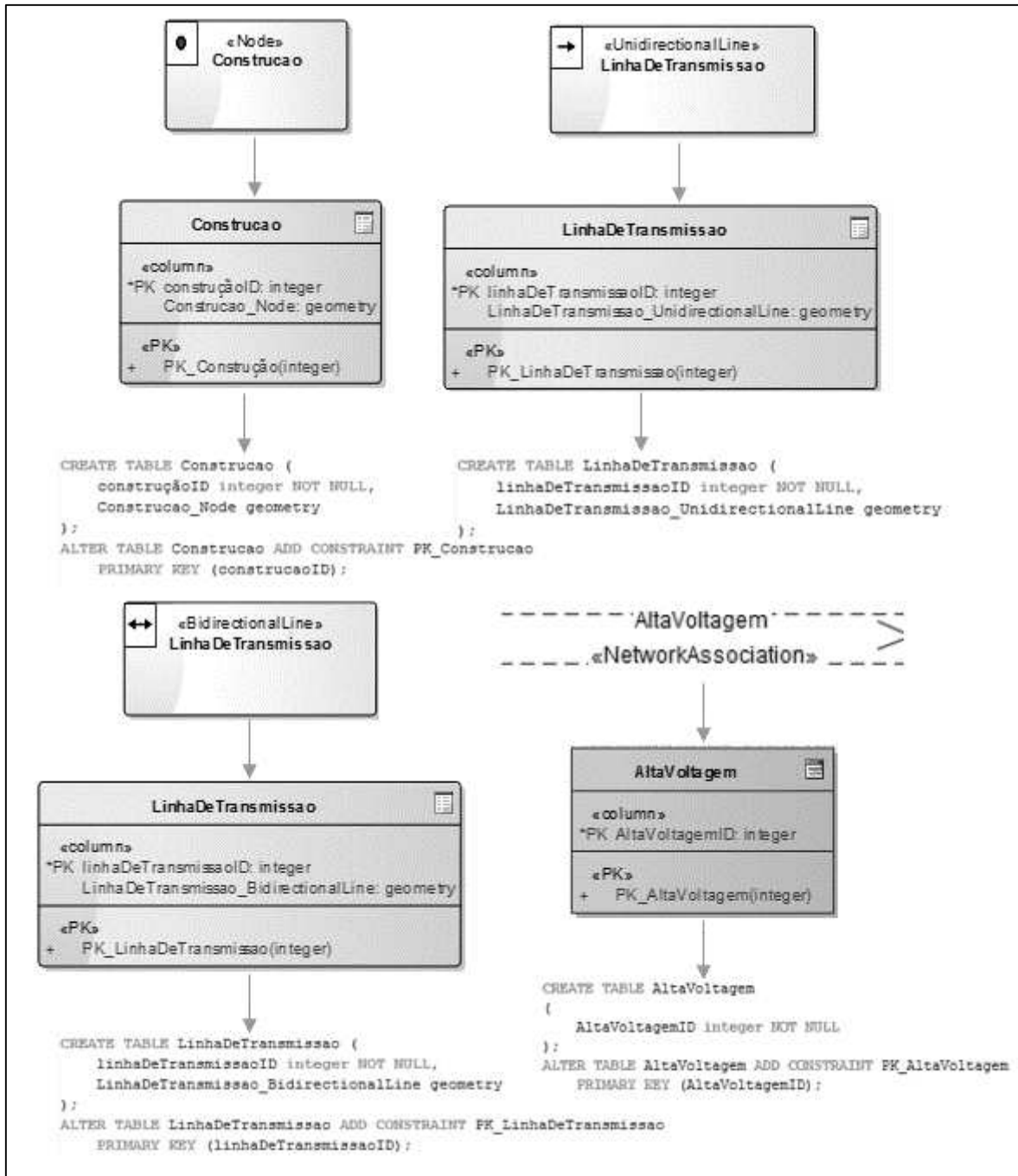


Figura 3.32: Especificação de mapeamento vertical da representação de redes da tecnologia MDG_OMTG.

3.7.3 Especificação do mapeamento vertical MDG_UML-GeoFrame

No mapeamento vertical dos requisitos de representação de tipos básicos de objetos espaciais na MDG_UML-GeoFrame (PIM) os estereótipos Point, Line, Polygon e ComplexSpatialObj são transformados em colunas de tabelas com tipos de dados geográficos específicos do SGBD PostGIS. Todos atributos das classes (PIM) são convertidos em colunas de tabelas. No ato da transformação estas tabelas também recebem uma coluna com o tipo de

dado integer nomeada com o nome da tabela + ID, para armazenar a chave primária. Desta forma, um esquema PSM é gerado.

A Figura 3.33 apresenta a especificação de mapeamento vertical dos construtores point, line, polygon e complexSpatialObj da tecnologia MDG_UML-GeoFrame. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice D.

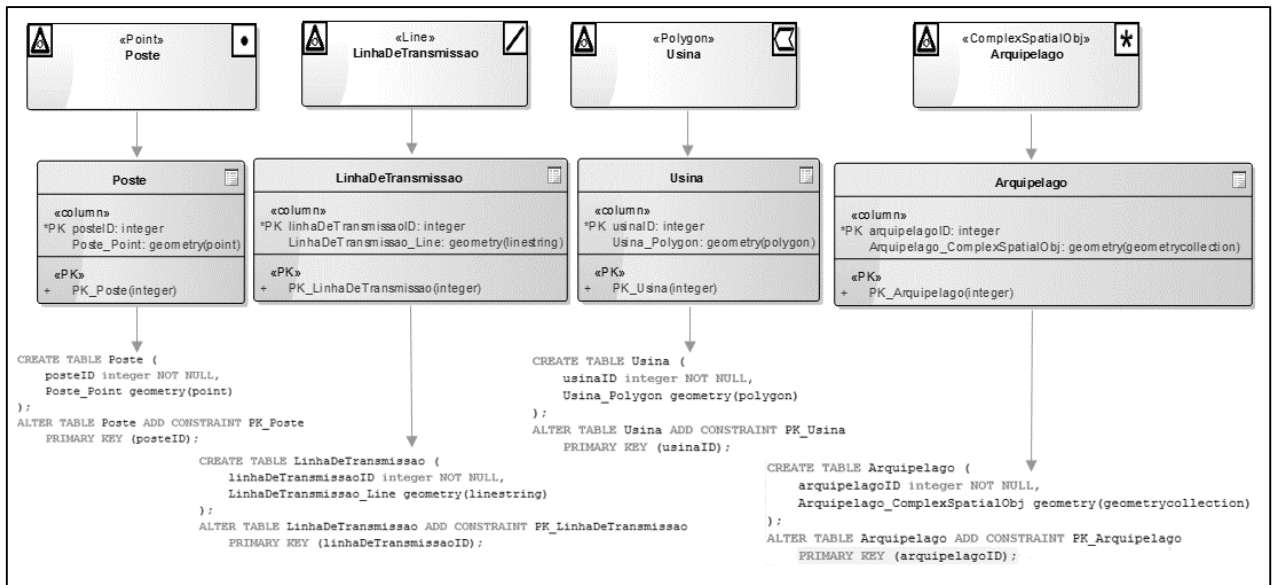


Figura 3.33: Especificação de mapeamento vertical da representação de objetos básicos da tecnologia MDG_UML-GeoFrame.

No mapeamento vertical dos requisitos de representação de campos contínuos na MDG_UML-GeoFrame (PIM) os estereótipos Isolines, AdjPolygon, IrregularPoints, GridOfCells, TIN e GridOfPoints, são transformados em colunas de tabelas seguidas do tipo de dados específico (Martínez, 2014). Todos os atributos das classes (PIM) são convertidos em colunas de tabelas PSM. A Figura 3.34 apresenta o mapeamento vertical dos construtores Isolines, AdjPolygons, IrregularPoints, GridOfCells, TIN e GridOfPoints. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice D.

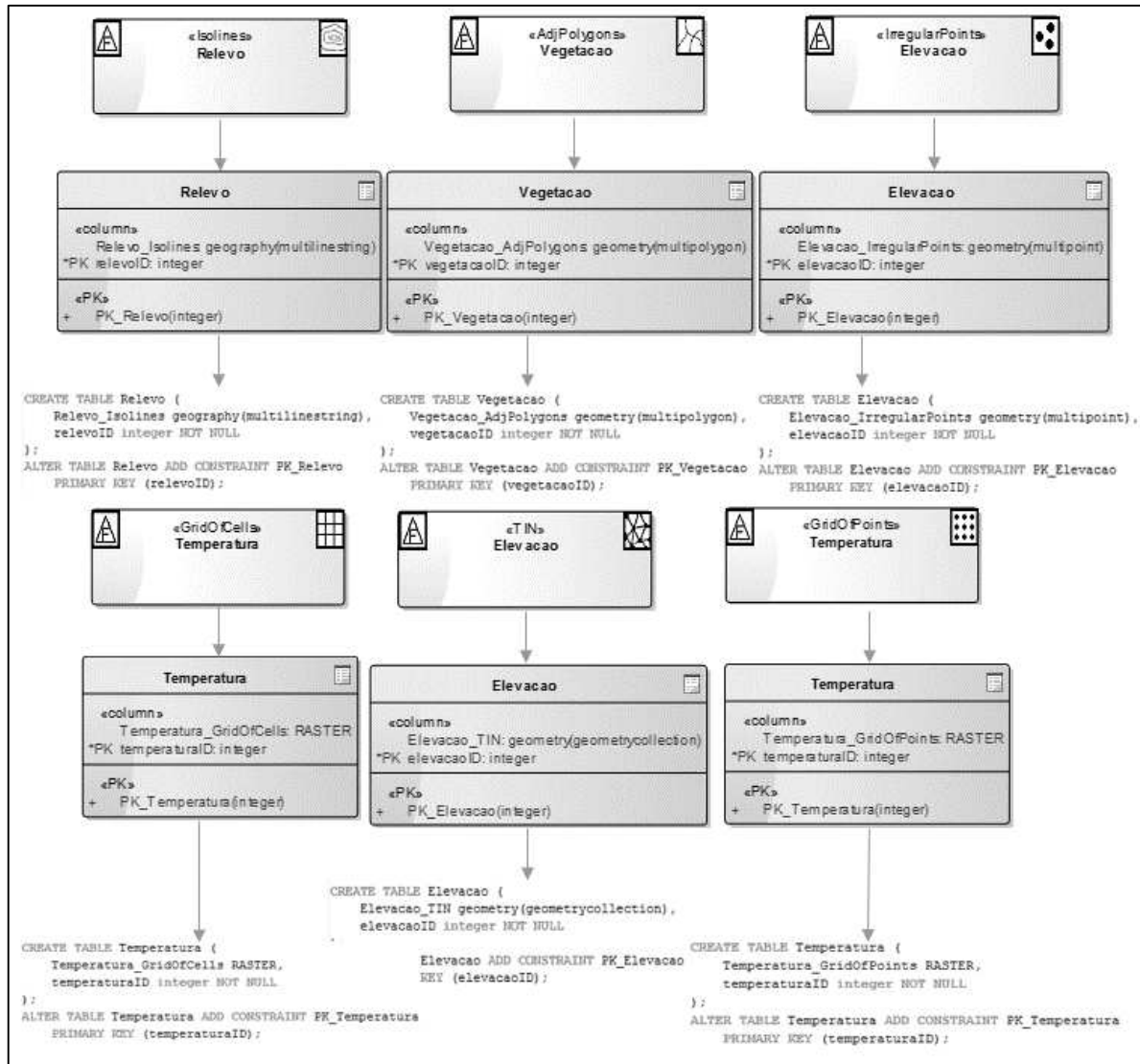


Figura 3.34: Especificação de mapeamento vertical da representação de campos contínuos da tecnologia MDG_UML-GeoFrame.

A tecnologia MDG_UML-GeoFrame foi especificada neste trabalho para ser utilizada na produção de esquemas conceituais desenvolvidos a partir de construtores de objetos básicos, construtores de campos contínuos e construtores de rede.

Os construtores de rede Node, UnidirectionalArc e BidirectionalArc da tecnologia MDG_UML-GeoFrame são convertidos em colunas de tabelas no nível PSM. No PostGIS essas colunas recebem o tipo de dado Geometry. O mapeamento vertical do construtor Network é o mesmo utilizado para o mapeamento de classes convencionais. A Figura 3.35 ilustra a transformação vertical dos construtores de rede da tecnologia MDG_UML-GeoFrame.

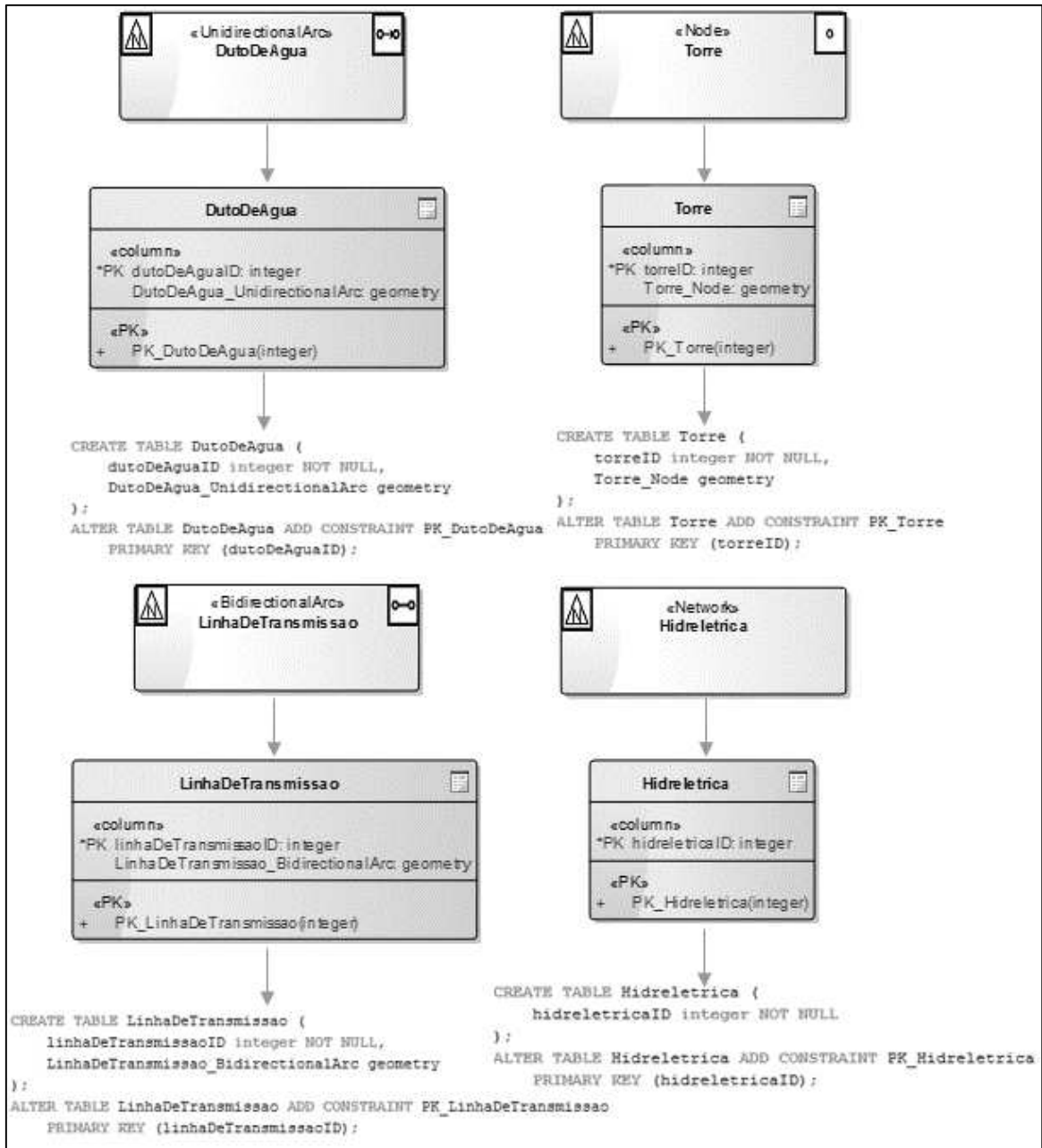


Figura 3.35: Especificação de mapeamento vertical da representação redes da tecnologia MDG_UML-GeoFrame.

3.7.4 Especificação do mapeamento vertical MDG_PVL

No mapeamento vertical dos requisitos de representação de tipos básicos de objetos espaciais na MDG_PVL (PIM) os estereótipos Point, Line, Polygon, MultiPoints, MultiLines, MultiPolygons e ComplexShape são transformados em colunas de tabelas com tipos de dados geográficos específicos do SGBD PostGIS. Todos atributos das classes (PIM) são convertidos em colunas de tabelas. A Figura 3.36 apresenta o mapeamento vertical dos construtores, Point, Line e Polygon. Este mapeamento é feito de forma automática por meio de código de transformação feito na ferramenta EA. Este código pode ser encontrado no Apêndice D.

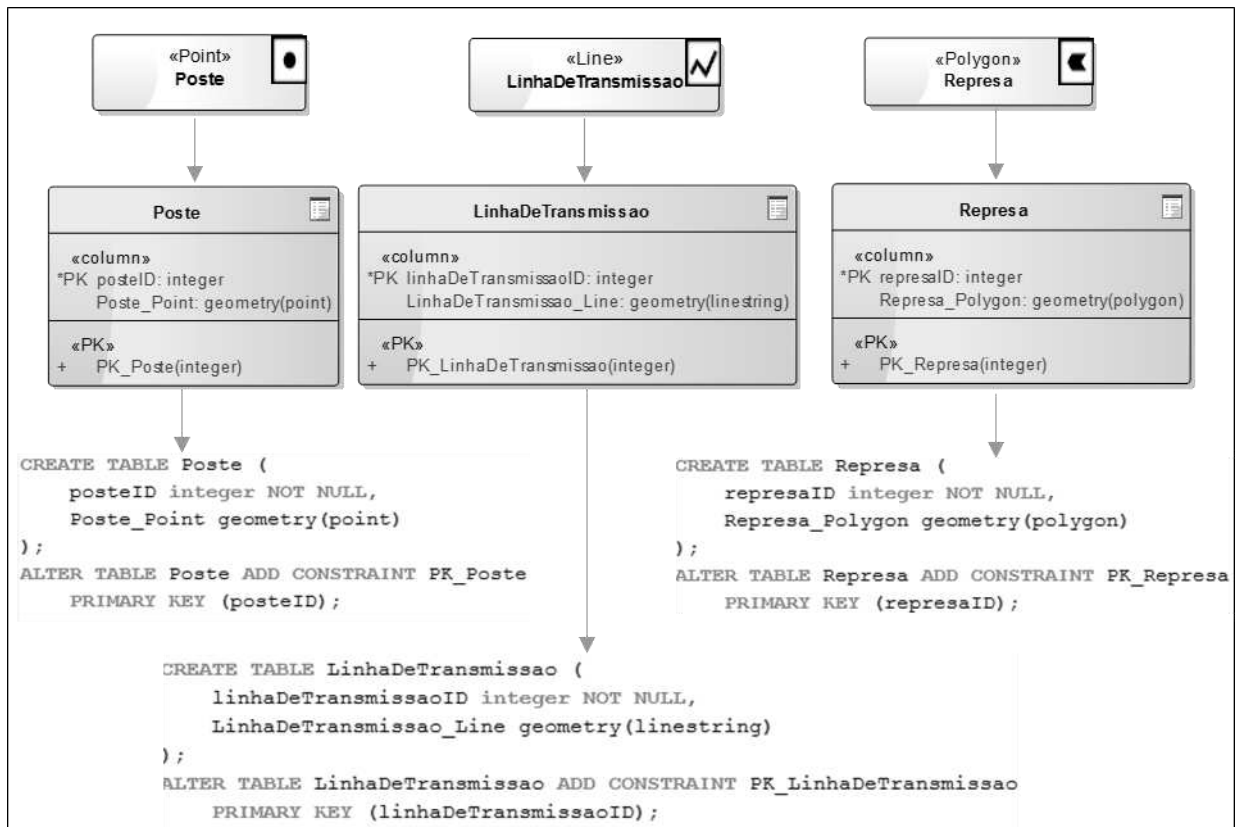


Figura 3.36: Especificação de mapeamento vertical dos construtores point, Line e Polygon da tecnologia MDG_PVL.

A Figura 3.37 apresenta o mapeamento dos demais construtores especificados neste trabalho. Sendo MultiPoints, MultiLines, MultiPolygons e ComplexShape. O código, do mapeamento vertical apresentado na Figura 3.37, pode ser encontrado no apêndice D.

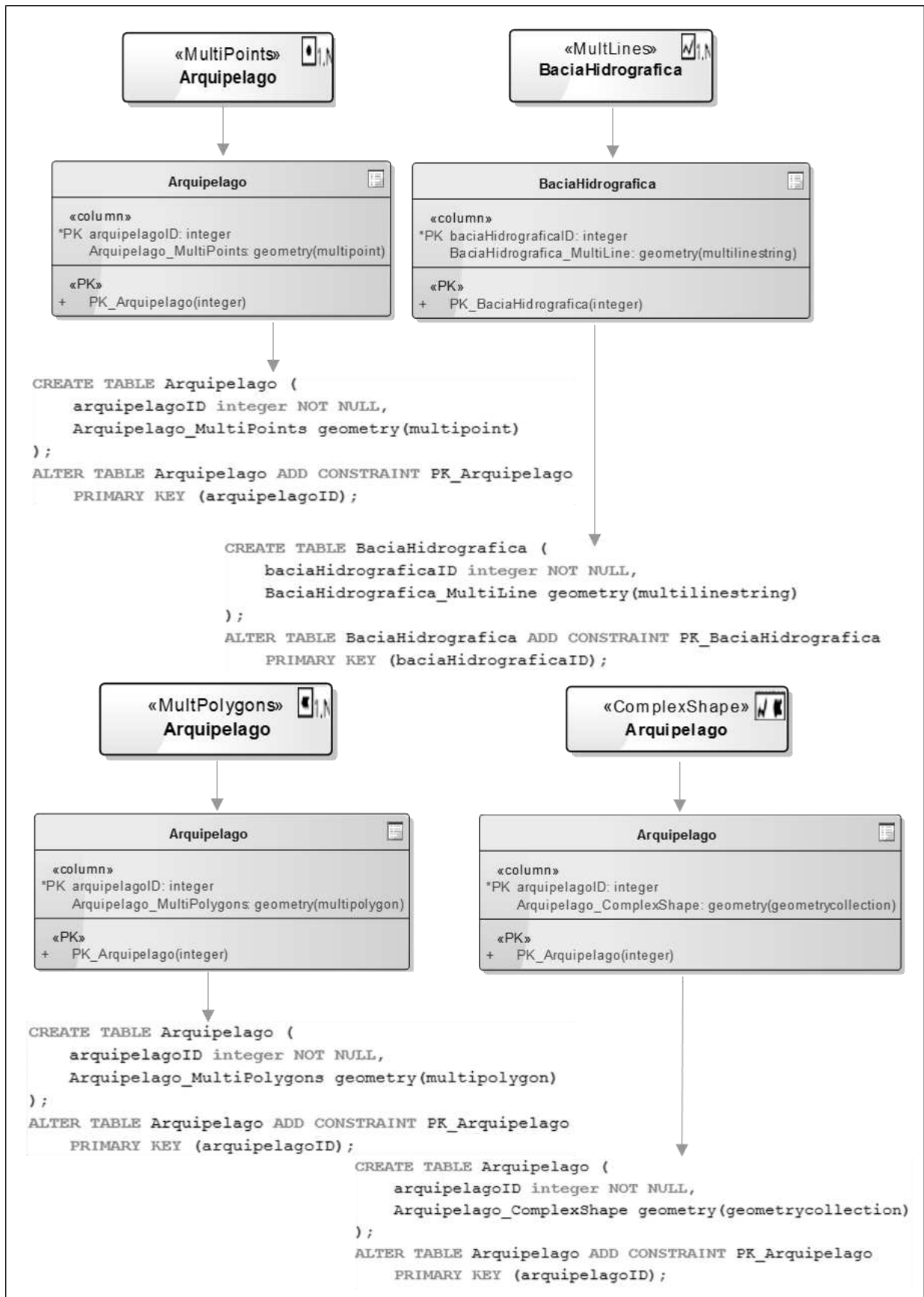


Figura 3.37: Especificação de mapeamento vertical dos construtores point, Line e Polygon da tecnologia MDG_PVL.

4 EXPERIMENTAÇÃO DO MÉTODO DE INTEROPERABILIDADE

Neste capítulo, uma análise da proposta foi executada com base em transformações de esquemas conceituais encontrados na literatura. Não foram trabalhadas todas as possibilidades de transformações. Na subseção 4.1 é apresentado um esquema conceitual referente à exploração de carvão. Esse esquema faz uso dos construtores do requisito representação de tipos básicos de objetos espaciais. Na subseção 4.2 é utilizado um esquema conceitual de uso do solo. Este esquema conceitual faz uso dos construtores do requisito representação de campos contínuos. Na subseção 4.3 é utilizado um esquema conceitual de distribuição de energia. Este esquema conceitual faz uso dos construtores utilizados para representação de redes.

4.1 Esquema 1: Mineração de Carvão

A Figura 4.1 apresenta um esquema conceitual de BDG sobre atividades voltadas para mineração de carvão em cidades, extraído de (LISBOA ET AL., 2002). Este esquema conceitual foi modelado a partir do modelo UML-GeoFrame.

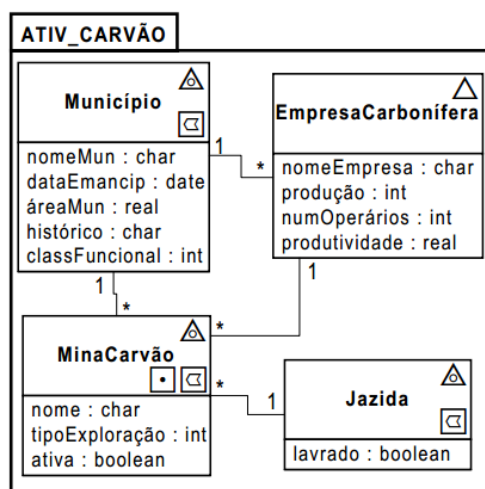


Figura 4.1: Esquema conceitual Mineração de Carvão- UML-GeoFrame. Fonte: Lisboa et al., 2002

O Pacote da Figura 4.1 apresenta um exemplo de atividade carvoeira onde um município representado por um objeto geográfico do tipo polígono (polygon) pode estar relacionada a nenhuma ou muitas empresas carboníferas. A classe Empresa Carbonífera foi modelada através de um objeto convencional. O município também pode possuir nenhuma ou muitas minas de carvão. A classe Mina Carvão pode ser representada por um objeto geográfico do tipo ponto (point), quando a mina de carvão é subterrânea, e por um objeto geográfico do tipo polígono (polygon), quando a mina de carvão se encontra em céu aberto. Uma Empresa

Carbonífera pode estar trabalhando em nenhuma ou muitas minas de carvão. Os carvões são retirados de jazidas. A Jazida é representada por um objeto geográfico do tipo Polygon.

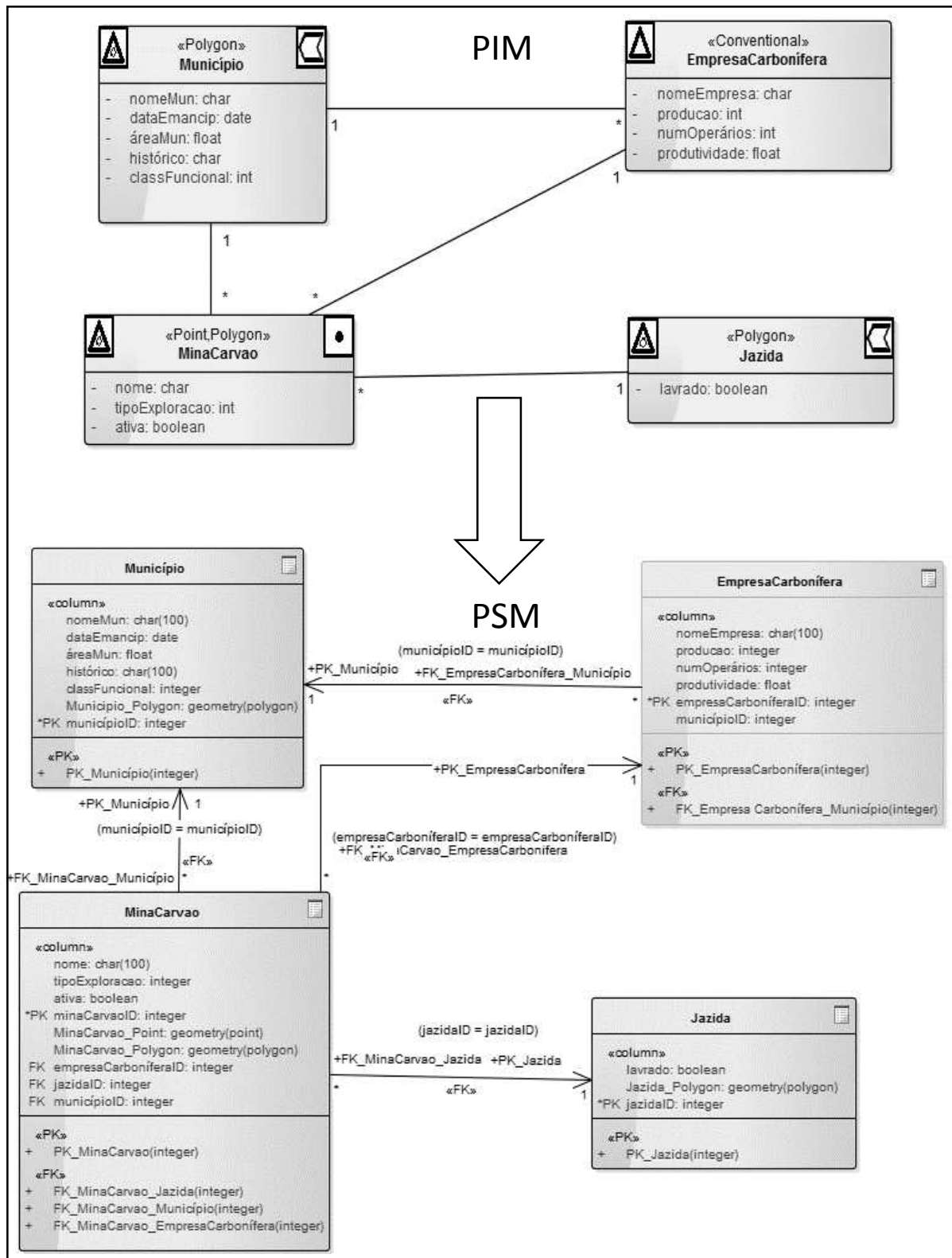


Figura 4.2: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_UML-GeoFrame (PIM) mapeado para o nível PSM.

A Figura 4.2 ilustra o esquema conceitual da Figura 4.1 modelado a partir da Tecnologia MDG_UML-GeoFrame e o esquema no nível PSM específico do SGBD PostGIS.

A partir do esquema PSM apresentado na Figura 4.2 e com a ferramenta EA foi possível gerar o código DDL automaticamente para o SGBD PostGIS. A Figura 4.3 apresenta o código gerado.

```

USE DATABASE PostGis;
DROP TABLE MinaCarvao CASCADE;
DROP TABLE Jazida CASCADE;
DROP TABLE EmpresaCarbonífera CASCADE;
DROP TABLE Município CASCADE;

/*Create Tables*/

CREATE TABLE MinaCarvao (
    nome char(100),
    tipoExploracao integer,
    ativa boolean,
    minaCarvaoID integer NOT NULL,
    MinaCarvao_Point geometry(point),
    MinaCarvao_Polygon geometry(polygon),
    empresaCarboníferaID integer,
    jazidaID integer,
    municípioID integer
);
CREATE TABLE Jazida (
    lavrado boolean,
    Jazida_Polygon geometry(polygon),
    jazidaID integer NOT NULL
);
CREATE TABLE EmpresaCarbonífera (
    nomeEmpresa char(100),
    producao integer,
    numOperários integer,
    produtividade float,
    empresaCarboníferaID integer NOT NULL,
    municípioID integer
);
CREATE TABLE Município (
    nomeMun char(100),
    dataEmancip date,
    áreaMun float,
    histórico char(100),
    classFuncional integer,
    Município_Polygon geometry(polygon),
    municípioID integer NOT NULL
);

/* Create Primary Keys, Indexes, Uniques, Checks*/

ALTER TABLE MinaCarvao ADD CONSTRAINT PK_MinaCarvao
    PRIMARY KEY (minaCarvaoID);
ALTER TABLE Jazida ADD CONSTRAINT PK_Jazida
    PRIMARY KEY (jazidaID);
ALTER TABLE EmpresaCarbonífera ADD CONSTRAINT PK_EmpresaCarbonífera
    PRIMARY KEY (empresaCarboníferaID);
ALTER TABLE Município ADD CONSTRAINT PK_Município
    PRIMARY KEY (municípioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Jazida
    FOREIGN KEY (jazidaID) REFERENCES Jazida (jazidaID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Município
    FOREIGN KEY (municípioID) REFERENCES Município (municípioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_EmpresaCarbonífera
    FOREIGN KEY (empresaCarboníferaID) REFERENCES EmpresaCarbonífera (empresaCarboníferaID);

```

Figura 4.3: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS

Aplicando as regras de mapeamentos horizontais desenvolvidos neste trabalho, o esquema conceitual (PIM) da Figura 4.2 pode ser mapeado (interoperabilidade horizontal) para um esquema conceitual (PIM) interpretado pelos construtores da tecnologia MDG_GeoProfile. A Figura 4.4 apresenta o esquema conceitual MineracaoCarvao a partir dos construtores da Tecnologia MDG_GeoProfile.

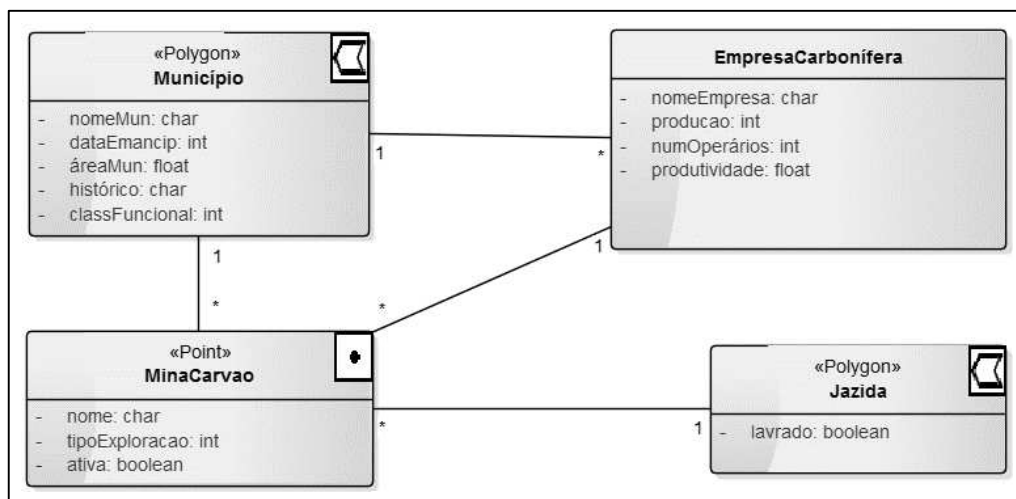


Figura 4.4: Esquema conceitual Mineração de Carvão ilustrado a partir dos construtores da Tecnologia MDG_GeoProfile.

Neste trabalho a tecnologia MDG_GeoProfile é utilizada como base de interoperabilidade. Desta forma, a partir do esquema conceitual apresentado na Figura 4.4 foram executados os mapeamentos horizontais (especificados neste trabalho) para que este esquema conceitual possa ser interpretado pelos construtores das tecnologias MDG_GeoOOA, MDG_OMTG e MDG_PVL.

Para cada novo esquema, interpretado por cada uma das Tecnologias MDG citadas anteriormente, foi feito o mapeamento vertical para o nível PSM seguido do mapeamento para código fonte DDL. A partir deste mapeamento foram verificadas as similaridades e diferenças entre as tecnologias em cada nível MDA para o requisito de representação de tipos básicos de objetos espaciais.

4.1.1 Esquema mineração de Carvão mapeado para Tecnologia MDG_GeoOOA

A partir do mapeamento horizontal do esquema conceitual apresentado na Figura 4.4 o PIM da Figura 4.5 pode ser gerado. Como neste trabalho não está sendo tratado o mapeamento horizontal do requisito múltipla representação geográfica, e o Modelo GeoOOA juntamente com a Tecnologia MDG não oferecem suporte para múltipla representação, a classe MinaCarvao está sendo representada unicamente pelo construtor point.

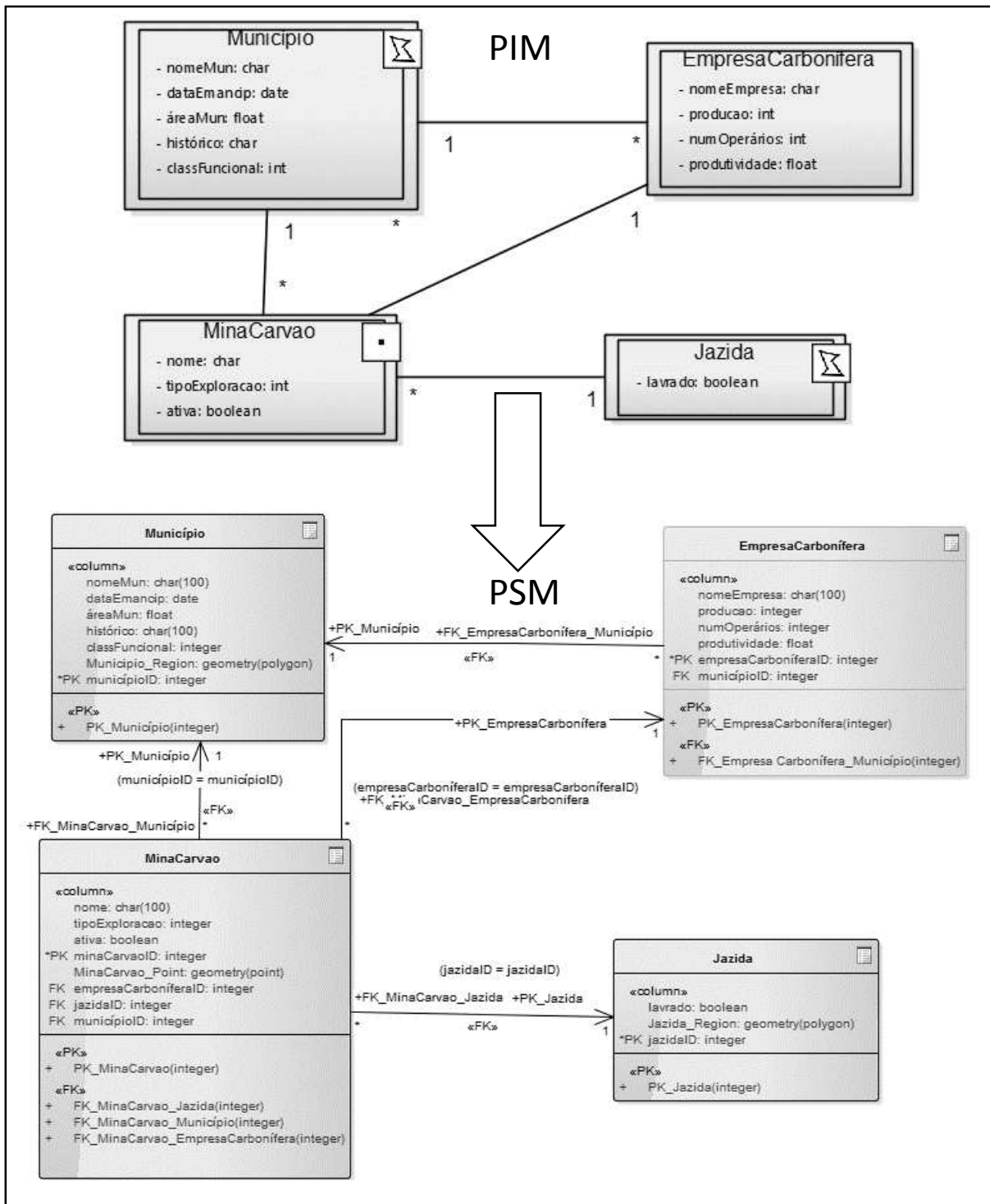


Figura 4.5: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_GeoOOA (PIM) mapeado para o nível PSM.

Ao comparar o PSM da Figura 4.5 com o PSM da Figura 4.2 percebe-se as diferenças nos nomes de algumas colunas da tabela como, por exemplo, Jazida_Polygon (PSM Figura 4.2) e Jazida_Região (PSM Figura 4.4). Outra diferença que pode ser percebida é a falta da coluna MinaCarvao_Polygon na tabela MinaCarvao do PSM da Figura 4.4. Esta coluna não existe devido a múltipla representação geográfica.

A Figura 4.6 apresenta o código DDL gerado automaticamente a partir do PSM da Figura 4.5.

```

USE DATABASE PostGis;
DROP TABLE MinaCarvao CASCADE;
DROP TABLE Jazida CASCADE;
DROP TABLE EmpresaCarbonifera CASCADE;
DROP TABLE Município CASCADE;

        /*Create Tables*/

CREATE TABLE MinaCarvao (
    nome char(100),
    tipoExploracao integer,
    ativa boolean,
    minaCarvaoID integer NOT NULL,
    MinaCarvao_Point geometry(point),
    empresaCarboniferalID integer,
    jazidaID integer,
    municípioID integer
);
CREATE TABLE Jazida (
    lavrado boolean,
    Jazida_Region geometry(polygon),
    jazidaID integer NOT NULL
);
CREATE TABLE EmpresaCarbonifera (
    nomeEmpresa char(100),
    producao integer,
    numOperários integer,
    produtividade float,
    empresaCarboniferalID integer NOT NULL,
    municípioID integer
);
CREATE TABLE Município (
    nomeMun char(100),
    dataEmancip date,
    áreaMun float,
    histórico char(100),
    classFuncional integer,
    Municipio_Region geometry(polygon),
    municípioID integer NOT NULL
);

        /* Create Primary Keys, Indexes, Uniques, Checks*/

ALTER TABLE MinaCarvao ADD CONSTRAINT PK_MinaCarvao
    PRIMARY KEY (minaCarvaoID);
ALTER TABLE Jazida ADD CONSTRAINT PK_Jazida
    PRIMARY KEY (jazidaID);
ALTER TABLE EmpresaCarbonifera ADD CONSTRAINT PK_EmpresaCarbonifera
    PRIMARY KEY (empresaCarboniferalID);
ALTER TABLE Município ADD CONSTRAINT PK_Município
    PRIMARY KEY (municípioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Jazida
    FOREIGN KEY (jazidaID) REFERENCES Jazida (jazidaID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Município
    FOREIGN KEY (municípioID) REFERENCES Município (municípioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_EmpresaCarbonifera
    FOREIGN KEY (empresaCarboniferalID) REFERENCES EmpresaCarbonifera (empresaCarboniferalID);
    
```

Figura 4.6: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS

Ao comparar as Figuras 4.3 e 4.5 é possível perceber que os códigos DDL apresentados diferem apenas nos nomes de algumas colunas das tabelas. Esta diferença também foi percebida no nível PSM.

4.1.2 Esquema mineração de Carvão mapeado para Tecnologia MDG_OMTG

A partir do mapeamento horizontal do esquema conceitual apresentado na Figura 4.4 o PIM da Figura 4.7 pode ser gerado.

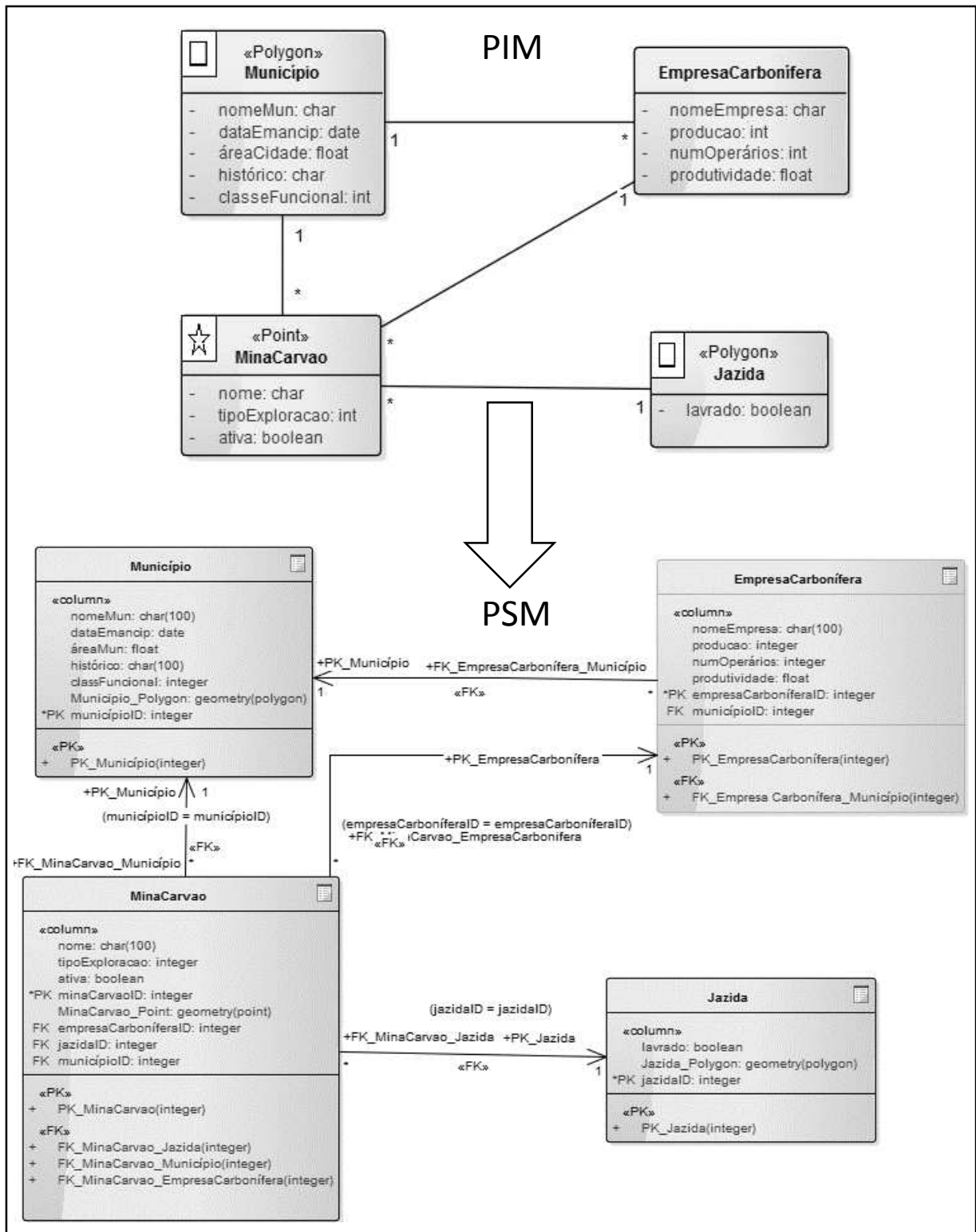


Figura 4.7: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_OMTG (PIM) mapeado para o nível PSM.

Novamente, como neste trabalho não está sendo tratado o mapeamento horizontal do requisito múltipla representação geográfica, a classe MinaCarvao está sendo representada unicamente pelo construtor point da Tecnologia MDG_OMTG.

Ao comparar o PSM da Figura 4.6 com o PSM da Figura 4.2 o que pode ser percebido é a falta da coluna MinaCarvao_Polygon na tabela MinaCarvao do PSM da Figura 4.6. Esta coluna não existe no PSM da Figura 4.6 devido a múltipla representação geográfica.

A Figura 4.8 apresenta o código DDL gerado automaticamente a partir do PSM da Figura 4.6.

```

USE DATABASE PostGis;
DROP TABLE MinaCarvao CASCADE;
DROP TABLE Jazida CASCADE;
DROP TABLE EmpresaCarbonífera CASCADE;
DROP TABLE Município CASCADE;

/*Create Tables*/

CREATE TABLE MinaCarvao (
    nome char(100),
    tipoExploracao integer,
    ativa boolean,
    minaCarvaoID integer NOT NULL,
    MinaCarvao_Point geometry(point),
    empresaCarboníferaID integer,
    jazidaID integer,
    municípioID integer
);
CREATE TABLE Jazida (
    lavrado boolean,
    Jazida_Polygon geometry(polygon),
    jazidaID integer NOT NULL
);
CREATE TABLE EmpresaCarbonífera (
    nomeEmpresa char(100),
    producao integer,
    numOperários integer,
    produtividade float,
    empresaCarboníferaID integer NOT NULL,
    municípioID integer
);
CREATE TABLE Município (
    nomeMun char(100),
    dataEmancip date,
    áreaMun float,
    histórico char(100),
    classFuncional integer,
    Município_Polygon geometry(polygon),
    municípioID integer NOT NULL
);

/* Create Primary Keys, Indexes, Uniques, Checks*/

ALTER TABLE MinaCarvao ADD CONSTRAINT PK_MinaCarvao
    PRIMARY KEY (minaCarvaoID);
ALTER TABLE Jazida ADD CONSTRAINT PK_Jazida
    PRIMARY KEY (jazidaID);
ALTER TABLE EmpresaCarbonífera ADD CONSTRAINT PK_EmpresaCarbonífera
    PRIMARY KEY (empresaCarboníferaID);
ALTER TABLE Município ADD CONSTRAINT PK_Município
    PRIMARY KEY (municípioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Jazida
    FOREIGN KEY (jazidaID) REFERENCES Jazida (jazidaID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Município
    FOREIGN KEY (municípioID) REFERENCES Município (municípioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_EmpresaCarbonífera
    FOREIGN KEY (empresaCarboníferaID) REFERENCES EmpresaCarbonífera (empresaCarboníferaID);

```

Figura 4.8: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS

4.1.3 Mineração de Carvão mapeado para Tecnologia MDG_PVL

A partir do mapeamento horizontal do esquema conceitual apresentado na Figura 4.4 o PIM da Figura 4.9 pode ser gerado.

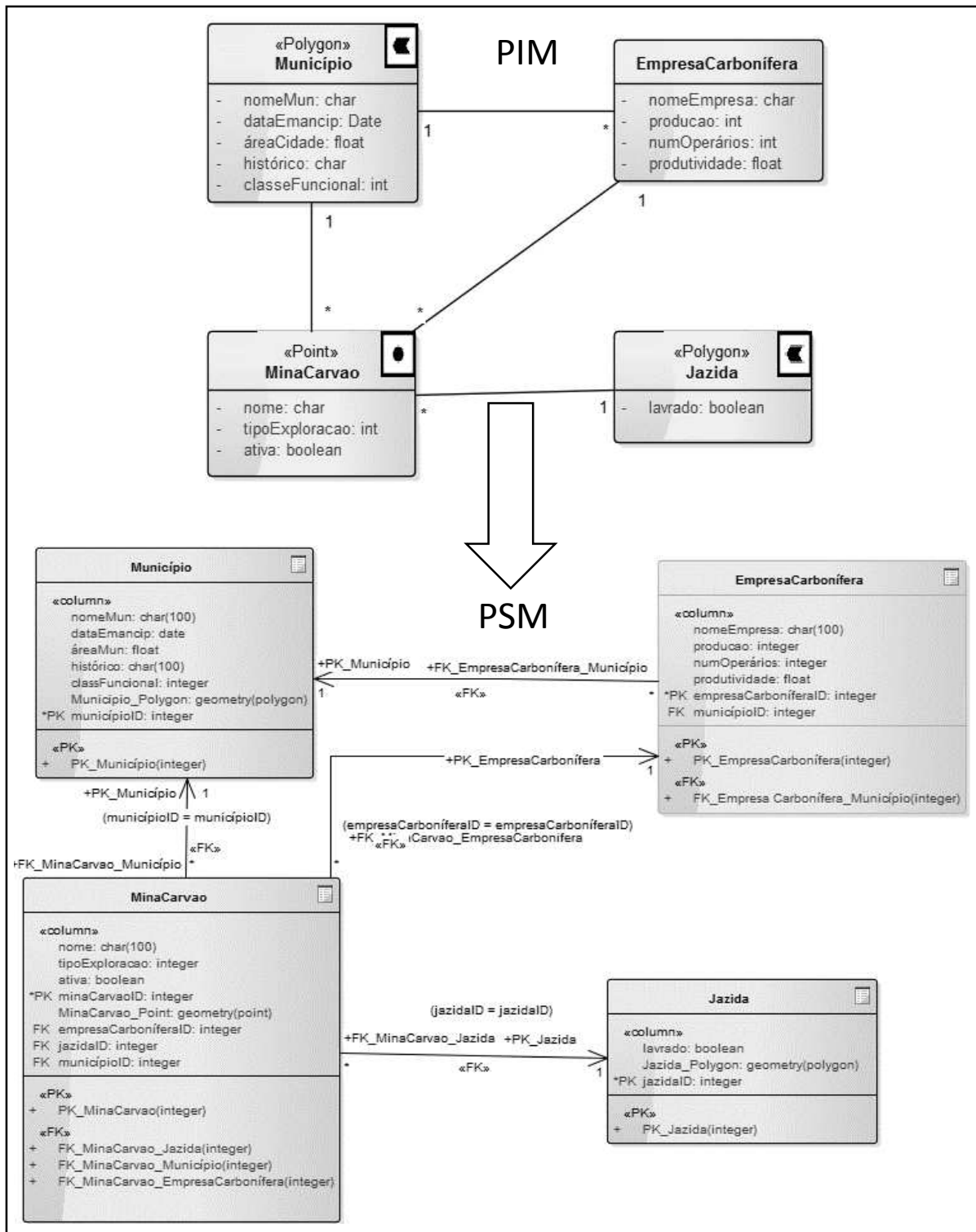


Figura 4.9: Esquema conceitual Mineração Carvão mostrado a partir da Tecnologia MDG_PVL (PIM) mapeado para o nível PSM.

A partir do PSM apresentado na Figura 4.9 o código DDL apresentado na Figura 4.10 pode ser gerado automaticamente pela ferramenta CASE EA.

```

USE DATABASE PostGis;
DROP TABLE MinaCarvao CASCADE;
DROP TABLE Jazida CASCADE;
DROP TABLE EmpresaCarbonifera CASCADE;
DROP TABLE Municipio CASCADE;

        /*Create Tables*/

CREATE TABLE MinaCarvao (
    nome char(100),
    tipoExploracao integer,
    ativa boolean,
    minaCarvaoID integer NOT NULL,
    MinaCarvao_Point geometry(point),
    empresaCarboniferalID integer,
    jazidaID integer,
    municipioID integer
);
CREATE TABLE Jazida (
    lavrado boolean,
    Jazida_Polygon geometry(polygon),
    jazidaID integer NOT NULL
);
CREATE TABLE EmpresaCarbonifera (
    nomeEmpresa char(100),
    producao integer,
    numOperários integer,
    produtividade float,
    empresaCarboniferalID integer NOT NULL,
    municipioID integer
);
CREATE TABLE Municipio (
    nomeMun char(100),
    dataEmancip date,
    áreaMun float,
    histórico char(100),
    classFuncional integer,
    Municipio_Polygon geometry(polygon),
    municipioID integer NOT NULL
);

        /* Create Primary Keys, Indexes, Uniques, Checks*/

ALTER TABLE MinaCarvao ADD CONSTRAINT PK_MinaCarvao
    PRIMARY KEY (minaCarvaoID);
ALTER TABLE Jazida ADD CONSTRAINT PK_Jazida
    PRIMARY KEY (jazidaID);
ALTER TABLE EmpresaCarbonifera ADD CONSTRAINT PK_EmpresaCarbonifera
    PRIMARY KEY (empresaCarboniferalID);
ALTER TABLE Municipio ADD CONSTRAINT PK_Municipio
    PRIMARY KEY (municipioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Jazida
    FOREIGN KEY (jazidaID) REFERENCES Jazida (jazidaID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_Municipio
    FOREIGN KEY (municipioID) REFERENCES Municipio (municipioID);
ALTER TABLE MinaCarvao ADD CONSTRAINT FK_MinaCarvao_EmpresaCarbonifera
    FOREIGN KEY (empresaCarboniferalID) REFERENCES EmpresaCarbonifera (empresaCarboniferalID);

```

Figura 4.10: Código DDL para criação do banco de dados Atividade_Carvoeira no SGBD PostGIS

4.2 Esquema 2: Meio Ambiente

A Figura 4.11 apresenta um pacote (Environment- Meio Ambiente) contendo um esquema conceitual BDG do Meio ambiente, extraído de (LISBOA FILHO E IOCHPE, 2008). Este esquema conceitual foi modelado a partir do modelo UML-GeoFrame. A classe Vegetation (Vegetação) é modelada com o construtor AdjPolygons (polígonos adjacentes), a classe Relief (Relevo) é modelada com o construtor Isolines (Isolinhas) e a classe Temperature (Temperatura) é modelada com o construtor GridOfPoints (Grade de Pontos).

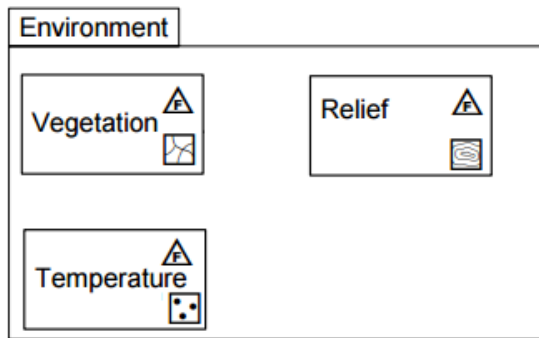


Figura 4.11: Esquema conceitual Meio ambiente. Fonte: Adaptado de Lisboa Filho e Iochpe, 2008.

A Figura 4.12 ilustra o esquema conceitual da Figura 4.11 modelado a partir da Tecnologia MDG_UML-GeoFrame e o esquema no nível PSM específico do SGBD PostGIS. Nesta Figura os termos em inglês da Figura 4.11 foram traduzidos para o Português.

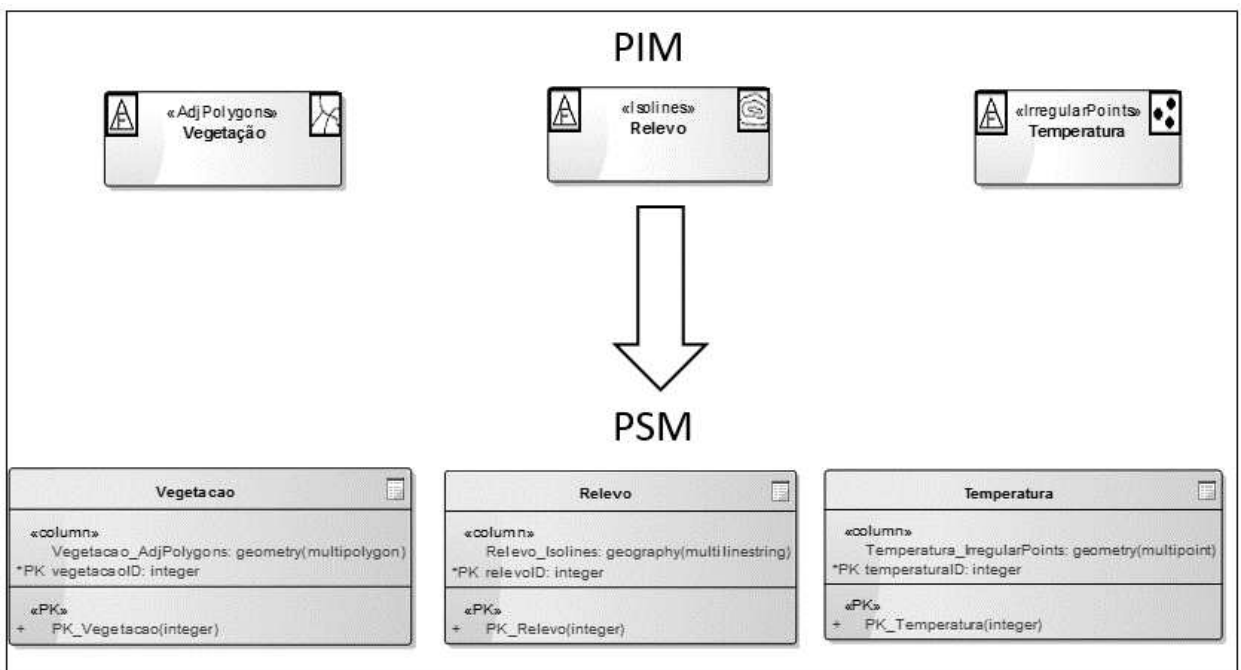


Figura 4.12: Esquema conceitual Meio Ambiente mostrado a partir da Tecnologia MDG_UML-GeoFrame (PIM) mapeado para o nível PSM.

A partir do esquema PSM apresentado na Figura 4.12 e com a ferramenta EA foi possível gerar o código em DDL automaticamente, para o SGBD PostGIS. A Figura 4.13 apresenta o código DDL gerado.

```

/*Drop Tables*/
DROP TABLE Vegetacao CASCADE;
DROP TABLE Temperatura CASCADE;
DROP TABLE Relevos CASCADE;
/*Create Tables*/
CREATE TABLE Vegetacao (
    Vegetacao_AdjPolygons geometry(multipolygon),
    vegetacaoID integer NOT NULL
);
CREATE TABLE Temperatura (
    Temperatura_IrregularPoints geometry(multipoint),
    temperaturaid integer NOT NULL
);
CREATE TABLE Relevos (
    Relevos_Isolines geography(multilinestring),
    relevosID integer NOT NULL
);
/*Create Primary keys*/
ALTER TABLE Vegetacao ADD CONSTRAINT PK_Vegetacao
    PRIMARY KEY (vegetacaoID);
ALTER TABLE Temperatura ADD CONSTRAINT PK_Temperatura
    PRIMARY KEY (temperaturaid);
ALTER TABLE Relevos ADD CONSTRAINT PK_Relevos
    PRIMARY KEY (relevosID);

```

Figura 4.13: Código DDL para criação do banco de dados Uso do Solo no SGBD PostGIS.

Ao aplicar as regras de mapeamentos horizontais desenvolvidos neste trabalho, o esquema conceitual (PIM) apresentado na Figura 4.12 pode ser mapeado (interoperabilidade horizontal) para um esquema conceitual (PIM) interpretado pelos construtores da tecnologia MDG_GeoProfile. A Figura 4.14 apresenta o esquema conceitual Meio Ambiente com os construtores da Tecnologia MDG_GeoProfile.



Figura 4.14: Esquema conceitual Meio Ambiente ilustrado a partir dos construtores da Tecnologia MDG_GeoProfile.

4.2.1 Esquema Meio Ambiente mapeado para Tecnologia MDG_OMTG

A partir do mapeamento horizontal do esquema conceitual apresentado na Figura 4.14 o PIM da Figura 4.15 pode ser gerado. A Figura 4.15 também apresenta o PSM gerado a partir das transformações verticais especificadas neste trabalho.

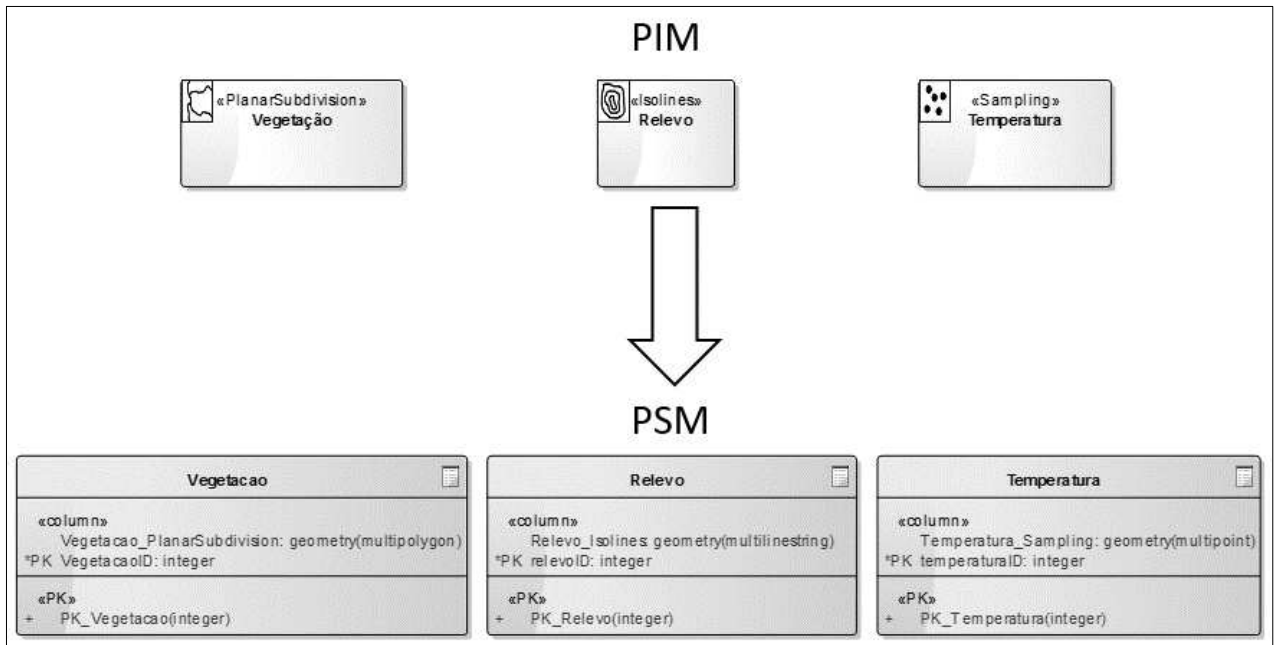


Figura 4.15: Esquema conceitual Uso do Solo mostrado a partir da Tecnologia MDG_UML-OTG (PIM) mapeado para o nível PSM.

A partir do PSM gerado na Figura 4.15 é possível gerar o código fonte DDL apresentado na Figura 4.16. Este código foi gerado automaticamente na ferramenta EA.

```

/* DROP TABLE*/
DROP TABLE Temperatura CASCADE;
DROP TABLE Relevo CASCADE;
DROP TABLE Vegetacao CASCADE;
/* CREATE TABLE*/
CREATE TABLE Temperatura (
    Temperatura_Sampling
    geometry(multipoint),
    temperaturaID integer NOT NULL
);
/* CREATE PRIMARY KEYS*/
ALTER TABLE Temperatura ADD CONSTRAINT PK_Temperatura
    PRIMARY KEY (temperaturaID);
ALTER TABLE Relevo ADD CONSTRAINT PK_Relevo
    PRIMARY KEY (relevoID);
ALTER TABLE Vegetacao ADD CONSTRAINT PK_Vegetacao
    PRIMARY KEY (VegetacaoID);
CREATE TABLE Relevo (
    Relevo_Isolines geometry(multilinestring),
    relevoID integer NOT NULL
);
CREATE TABLE Vegetacao (
    Vegetacao_PlanarSubdivision
    geometry(multipolygon),
    VegetacaoID integer NOT NULL
);

```

Figura 4.16: Código DDL para criação do banco de dados Uso do Solo no SGBD PostGIS.

4.3 Esquema 3: Rede de Distribuição de Energia

Neste trabalho, foram especificados os mapeamentos verticais e horizontais para os requisitos de rede, no entanto, estas especificações não foram codificadas com os recursos da

ferramenta EA. Entende-se que são necessários maiores esforços, para que a interoperabilidade vertical e horizontal dos requisitos de redes, possam ser desenvolvidas na ferramenta. Desta forma, os esquemas apresentados nesta seção não foram gerados automaticamente.

A Figura 4.17 apresenta um esquema conceitual de BDG de uma rede de distribuição de energia extraído de (KÖSTERS, PAGEL E SIX, 1996). Este esquema conceitual foi modelado a partir do modelo GeoOOA. A classe Transmissionline (Linha de Transmissão) é modelada a partir do construtor Line (Linha). Transmissionline é um link da rede HighVoltage (Alta Voltagem), um Transformer (Transformador) representado por um objeto do tipo Region (Região), é um nodo da rede HighVoltage e um nodo da rede LowVoltage (Baixa Voltagem). A rede LowVoltage também está ligada a um link Cable Line (Linha de cabo) que é representada por um objeto do tipo line.

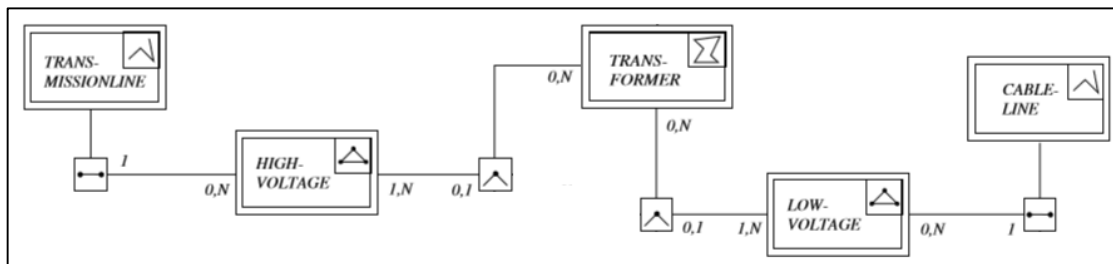


Figura 4.17: Esquema conceitual Rede de Distribuição de Energia- GeoOOA. Fonte: Adaptado de (KÖSTERS, PAGEL E SIX, 1996).

A Figura 4.18 ilustra o esquema conceitual da Figura 4.17 modelado a partir da Tecnologia MDG_GeoOOA. As cardinalidades foram invertidas para adequá-las a UML.

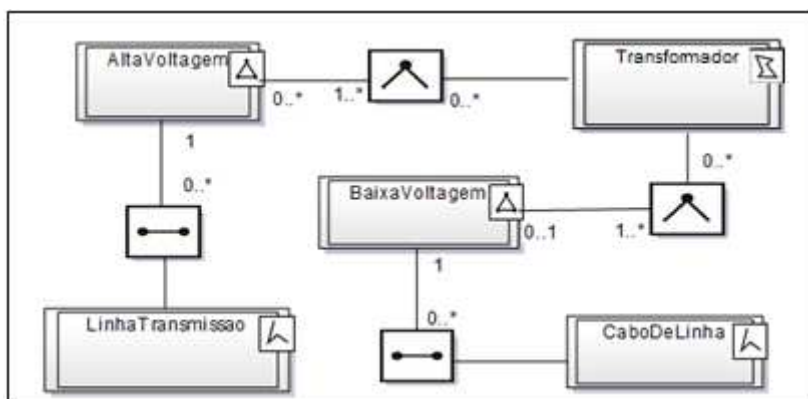


Figura 4.18: Esquema conceitual Rede de Distribuição de Energia mostrado a partir da Tecnologia MDG_GeoOOA (PIM).

De acordo com as especificações apresentadas neste trabalho, a Figura 4.19 apresenta o esquema PSM a partir do mapeamento vertical do esquema conceitual apresentado na Figura 4.18.

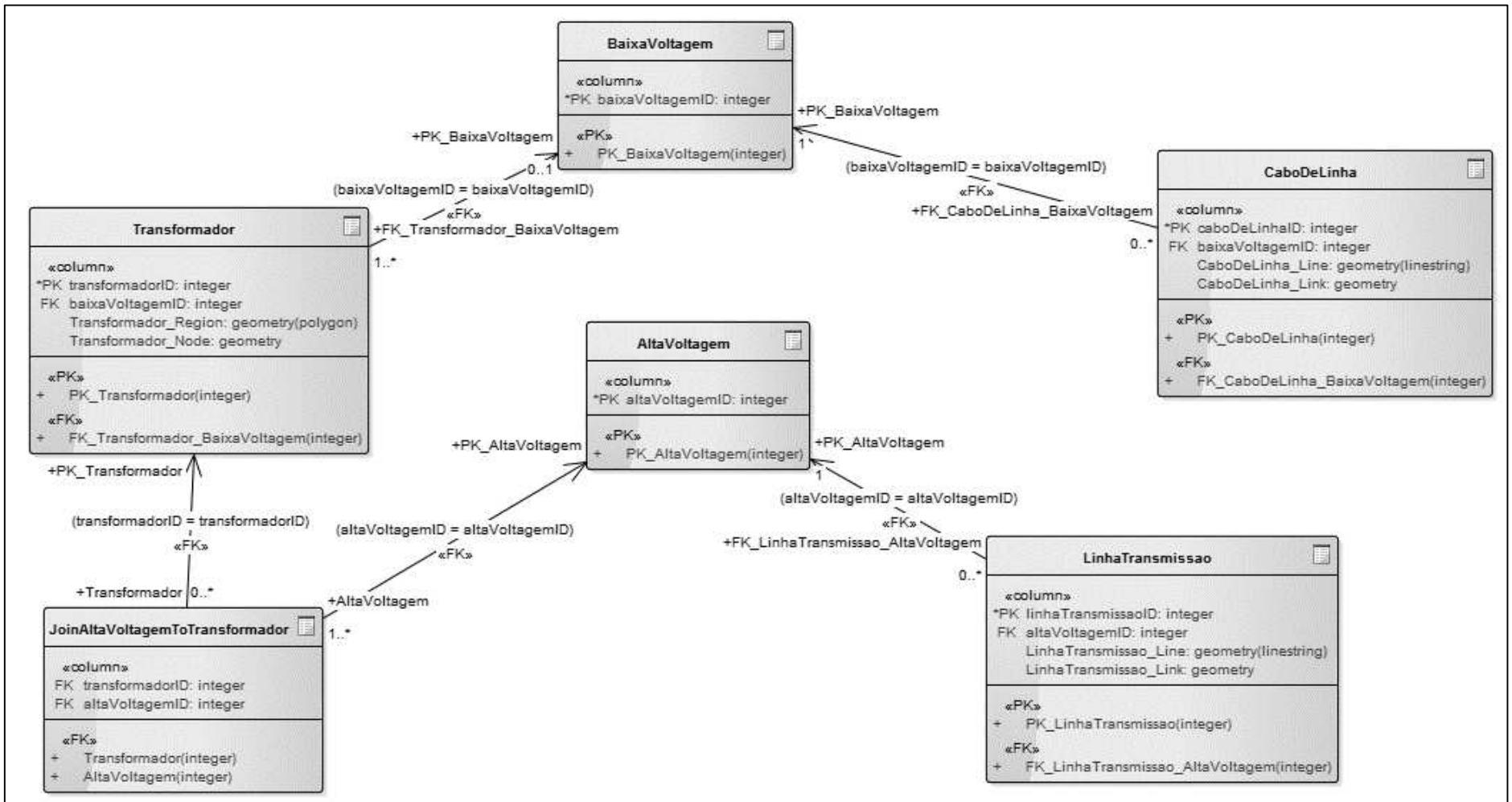


Figura 4.19. Esquema PSM Rede de Distribuição de Energia.

A tabela `JoinAltaVoltagemToTransformador` foi criada porque existe um relacionamento com cardinalidade N:N entre as tabelas `Transformador` e `AltaVoltagem`. As transformações destas cardinalidades foram baseadas no código de transformação DDL da ferramenta EA. A Figura 4.20 apresenta o código fonte DDL gerado a partir da Figura 4.19.

```

DROP TABLE Transformador CASCADE;
DROP TABLE LinhaTransmissao CASCADE;
DROP TABLE CaboDeLinha CASCADE;
DROP TABLE BaixaVoltagem CASCADE;
DROP TABLE JoinAltaVoltagemToTransformador CASCADE;
DROP TABLE AltaVoltagem CASCADE;
CREATE TABLE Transformador (
    transformadorID integer NOT NULL,
    baixaVoltagemID integer,
    Transformador_Region geometry(polygon),
    Transformador_Node geometry
);
CREATE TABLE LinhaTransmissao (
    linhaTransmissaoID integer NOT NULL,
    altaVoltagemID integer,
    LinhaTransmissao_Line geometry(linestring),
    LinhaTransmissao_Link geometry
);
CREATE TABLE CaboDeLinha (
    caboDeLinhaID integer NOT NULL,
    baixaVoltagemID integer,
    CaboDeLinha_Line geometry(linestring),
    CaboDeLinha_Link geometry
);
CREATE TABLE BaixaVoltagem (
    baixaVoltagemID integer NOT NULL
);
CREATE TABLE JoinAltaVoltagemToTransformador (
    transformadorID integer,
    altaVoltagemID integer
);
CREATE TABLE AltaVoltagem (
    altaVoltagemID integer NOT NULL
);
ALTER TABLE Transformador ADD CONSTRAINT PK_Transformador
    PRIMARY KEY (transformadorID);
ALTER TABLE LinhaTransmissao ADD CONSTRAINT PK_LinhaTransmissao
    PRIMARY KEY (linhaTransmissaoID);
ALTER TABLE CaboDeLinha ADD CONSTRAINT PK_CaboDeLinha
    PRIMARY KEY (caboDeLinhaID);
ALTER TABLE BaixaVoltagem ADD CONSTRAINT PK_BaixaVoltagem
    PRIMARY KEY (baixaVoltagemID);
ALTER TABLE AltaVoltagem ADD CONSTRAINT PK_AltaVoltagem
    PRIMARY KEY (altaVoltagemID);
ALTER TABLE Transformador ADD CONSTRAINT FK_Transformador_BaixaVoltagem
    FOREIGN KEY (baixaVoltagemID) REFERENCES BaixaVoltagem (baixaVoltagemID);
ALTER TABLE LinhaTransmissao ADD CONSTRAINT FK_LinhaTransmissao_AltaVoltagem
    FOREIGN KEY (altaVoltagemID) REFERENCES AltaVoltagem (altaVoltagemID);
ALTER TABLE CaboDeLinha ADD CONSTRAINT FK_CaboDeLinha_BaixaVoltagem
    FOREIGN KEY (baixaVoltagemID) REFERENCES BaixaVoltagem (baixaVoltagemID);
ALTER TABLE JoinAltaVoltagemToTransformador ADD CONSTRAINT Transformador
    FOREIGN KEY (transformadorID) REFERENCES Transformador (transformadorID);
ALTER TABLE JoinAltaVoltagemToTransformador ADD CONSTRAINT AltaVoltagem
    FOREIGN KEY (altaVoltagemID) REFERENCES AltaVoltagem (altaVoltagemID);

```

Figura 4.20: Código DDL para criação do banco de dados de uma rede de distribuição de energia no SGBD PostGIS.

Ao aplicar as regras de mapeamentos horizontais desenvolvidos neste trabalho, o esquema conceitual (PIM) apresentado na Figura 4.7 pode ser mapeado (interoperabilidade horizontal) para um esquema conceitual (PIM) utilizando os construtores da tecnologia MDG_GeoProfile. A Figura 4.21 apresenta o esquema conceitual da Rede de Distribuição de Energia com os construtores da Tecnologia MDG_GeoProfile.

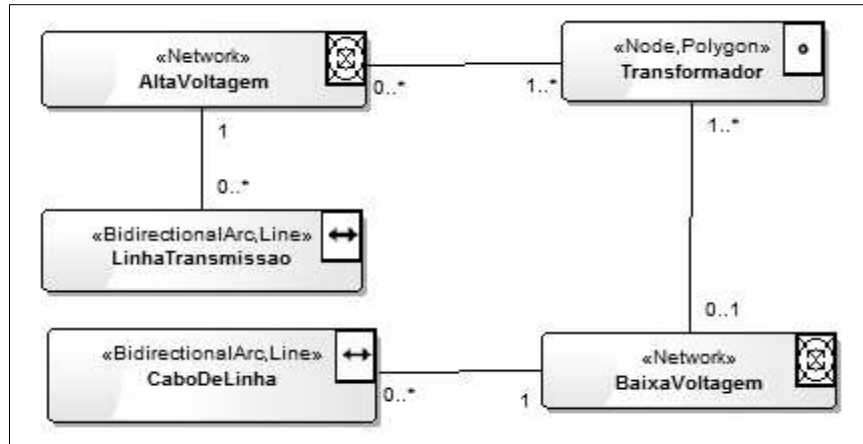


Figura 4.21: Esquema Distribuição de Energia ilustrado a partir dos construtores da Tecnologia MDG_GeoProfile.

4.3.1 Esquema Rede de Distribuição de Energia mapeado para Tecnologia MDG_UML-GeoFrame

A partir do mapeamento horizontal do esquema conceitual apresentado na Figura 4.21 o PIM da Figura 4.22 pode ser gerado. A Figura 4.22 apresenta o PIM de uma rede de distribuição de energia, mostrado a partir dos construtores da tecnologia MDG_UML_GeoFrame. Este esquema conceitual foi mapeado para um PSM. A Figura 4.23 apresenta o PSM. O PSM foi mapeado para código fonte DDL. A Figura 4.24 apresenta o código fonte DDL gerado automaticamente pela ferramenta EA.

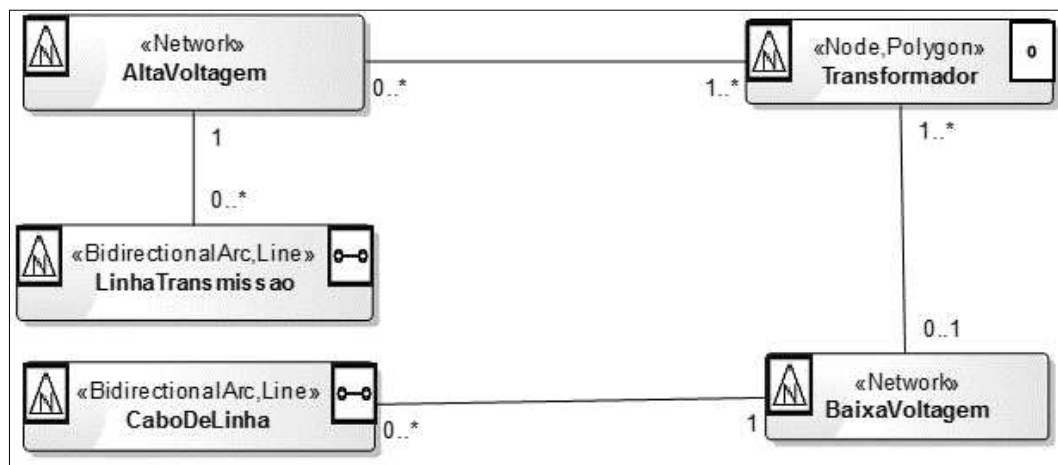


Figura 4.22: Esquema Rede de Distribuição de Energia elaborado com os construtores da Tecnologia MDG_UML-GeoFrame.

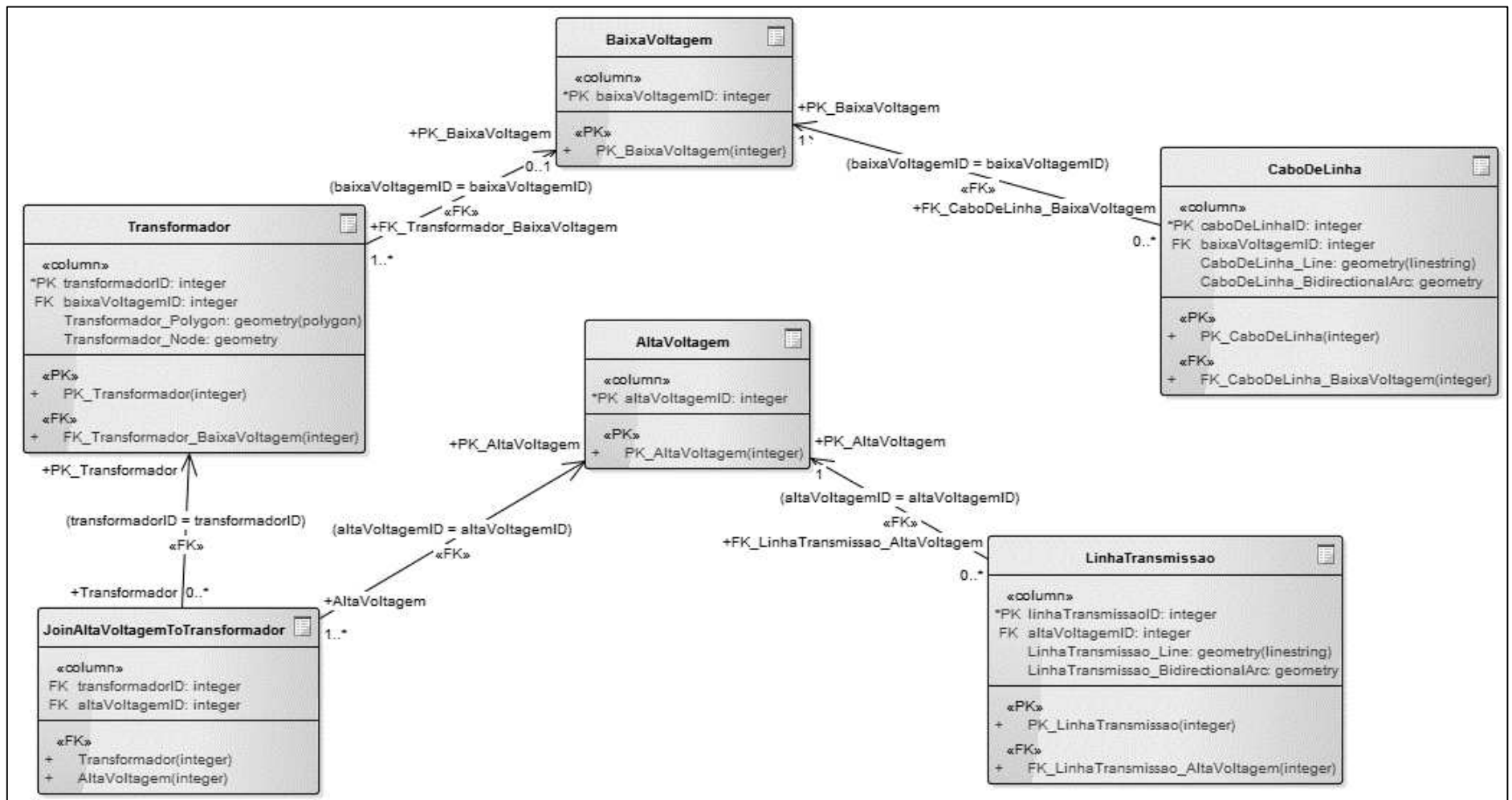


Figura 4.23. Esquema PSM Rede de Distribuição de Energia- UML-GeoFrame.

```

DROP TABLE Transformador CASCADE;
DROP TABLE LinhaTransmissao CASCADE;
DROP TABLE CaboDeLinha CASCADE;
DROP TABLE BaixaVoltagem CASCADE;
DROP TABLE JoinAltaVoltagemToTransformador CASCADE;
DROP TABLE AltaVoltagem CASCADE;
CREATE TABLE Transformador (
    transformadorID integer NOT NULL,
    baixaVoltagemID integer,
    Transformador_Polygon geometry(polygon),
    Transformador_Node geometry
);
CREATE TABLE LinhaTransmissao (
    linhaTransmissaoID integer NOT NULL,
    altaVoltagemID integer,
    LinhaTransmissao_Line geometry(linestring),
    LinhaTransmissao_BidirectionalArc geometry
);
ALTER TABLE Transformador ADD CONSTRAINT PK_Transformador
    PRIMARY KEY (transformadorID);
ALTER TABLE LinhaTransmissao ADD CONSTRAINT PK_LinhaTransmissao
    PRIMARY KEY (linhaTransmissaoID);
ALTER TABLE CaboDeLinha ADD CONSTRAINT PK_CaboDeLinha
    PRIMARY KEY (caboDeLinhaID);
ALTER TABLE BaixaVoltagem ADD CONSTRAINT PK_BaixaVoltagem
    PRIMARY KEY (baixaVoltagemID);
ALTER TABLE AltaVoltagem ADD CONSTRAINT PK_AltaVoltagem
    PRIMARY KEY (altaVoltagemID);
ALTER TABLE Transformador ADD CONSTRAINT FK_Transformador_BaixaVoltagem
    FOREIGN KEY (baixaVoltagemID) REFERENCES BaixaVoltagem (baixaVoltagemID);
ALTER TABLE LinhaTransmissao ADD CONSTRAINT FK_LinhaTransmissao_AltaVoltagem
    FOREIGN KEY (altaVoltagemID) REFERENCES AltaVoltagem (altaVoltagemID);
ALTER TABLE CaboDeLinha ADD CONSTRAINT FK_CaboDeLinha_BaixaVoltagem
    FOREIGN KEY (baixaVoltagemID) REFERENCES BaixaVoltagem (baixaVoltagemID);
ALTER TABLE JoinAltaVoltagemToTransformador ADD CONSTRAINT Transformador
    FOREIGN KEY (transformadorID) REFERENCES Transformador (transformadorID);
ALTER TABLE JoinAltaVoltagemToTransformador ADD CONSTRAINT AltaVoltagem
    FOREIGN KEY (altaVoltagemID) REFERENCES AltaVoltagem (altaVoltagemID);
CREATE TABLE CaboDeLinha (
    caboDeLinhaID integer NOT NULL,
    baixaVoltagemID integer,
    CaboDeLinha_Line geometry(linestring),
    CaboDeLinha_BidirectionalArc geometry
);
CREATE TABLE BaixaVoltagem (
    baixaVoltagemID integer NOT NULL
);
CREATE TABLE JoinAltaVoltagemToTransformador (
    transformadorID integer,
    altaVoltagemID integer
);
CREATE TABLE AltaVoltagem (
    altaVoltagemID integer NOT NULL
);

```

Figura 4.24: Código DDL para criação do banco de dados Distribuição de Energia no SGBD PostGIS.

4.3.2 Esquema Rede de Distribuição de Energia mapeado para Tecnologia MDG_OMTG

A partir do mapeamento horizontal do esquema conceitual apresentado na Figura 4.21 o PIM da Figura 4.25 pode ser gerado. A Figura 4.25 apresenta o PIM de uma rede de distribuição de energia, elaborado com os construtores da tecnologia MDG_OMTG. Observe que a quantidade de classes entre os esquemas das tecnologias MDG_GeoOOA, ilustrado na 4.18 e o esquema MDG_OMTG da Figura 4.25 diferem-se em número. Desta forma, as cardinalidades não foram levadas em consideração nesta transformação, onde o nível de abstração MDA utilizado é o PIM.

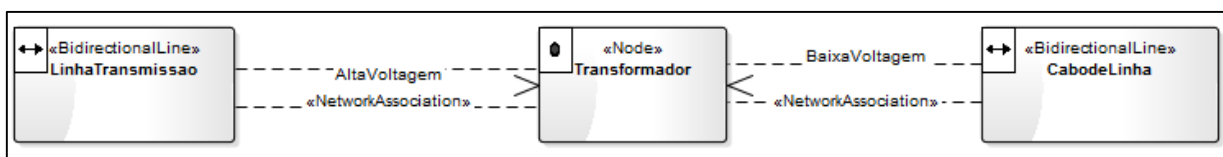


Figura 4.25: Esquema Rede de Distribuição de Energia elaborado com os construtores da Tecnologia MDG_OMTG.

Para resolver o problema com a cardinalidade, deve-se fazer alterações no esquema PSM. Se o projetista tiver interesse em colocar informações alfanuméricas sobre a rede, estas alterações também devem ser colocadas no nível PSM. De acordo com o especificado neste trabalho, estas associações específicas do OMTG devem virar tabelas no PSM, e são nestas tabelas que informações sobre a rede podem ser adicionadas.

Para gerar o PSM as seguintes cardinalidades devem ser consideradas: Uma linha de transmissão está associada a no mínimo 1 e no máximo 1 rede de alta voltagem; Uma rede de alta voltagem pode estar associada a 0 ou muitas linhas de transmissão; uma rede de alta voltagem pode estar associada a 1 ou n transformadores; um transformador pode estar associado a 0 ou n redes de alta voltagem; Um transformador pode estar associado a 0 ou 1 rede de baixa voltagem; Uma rede de baixa voltagem pode estar associada a 1 ou n transformadores; Um cabo de linha deve estar associado a 1 rede de baixa voltagem; Uma rede de baixa voltagem pode estar associado a 0 ou n cabos de linha.

A partir do PIM apresentado na 4.25 e da descrição das cardinalidades a serem utilizadas apresentada no parágrafo anterior, o PSM apresentado na Figura 4.26 pode ser gerado. Logo o código DDL apresentado na Figura 4.27 também pode ser gerado.

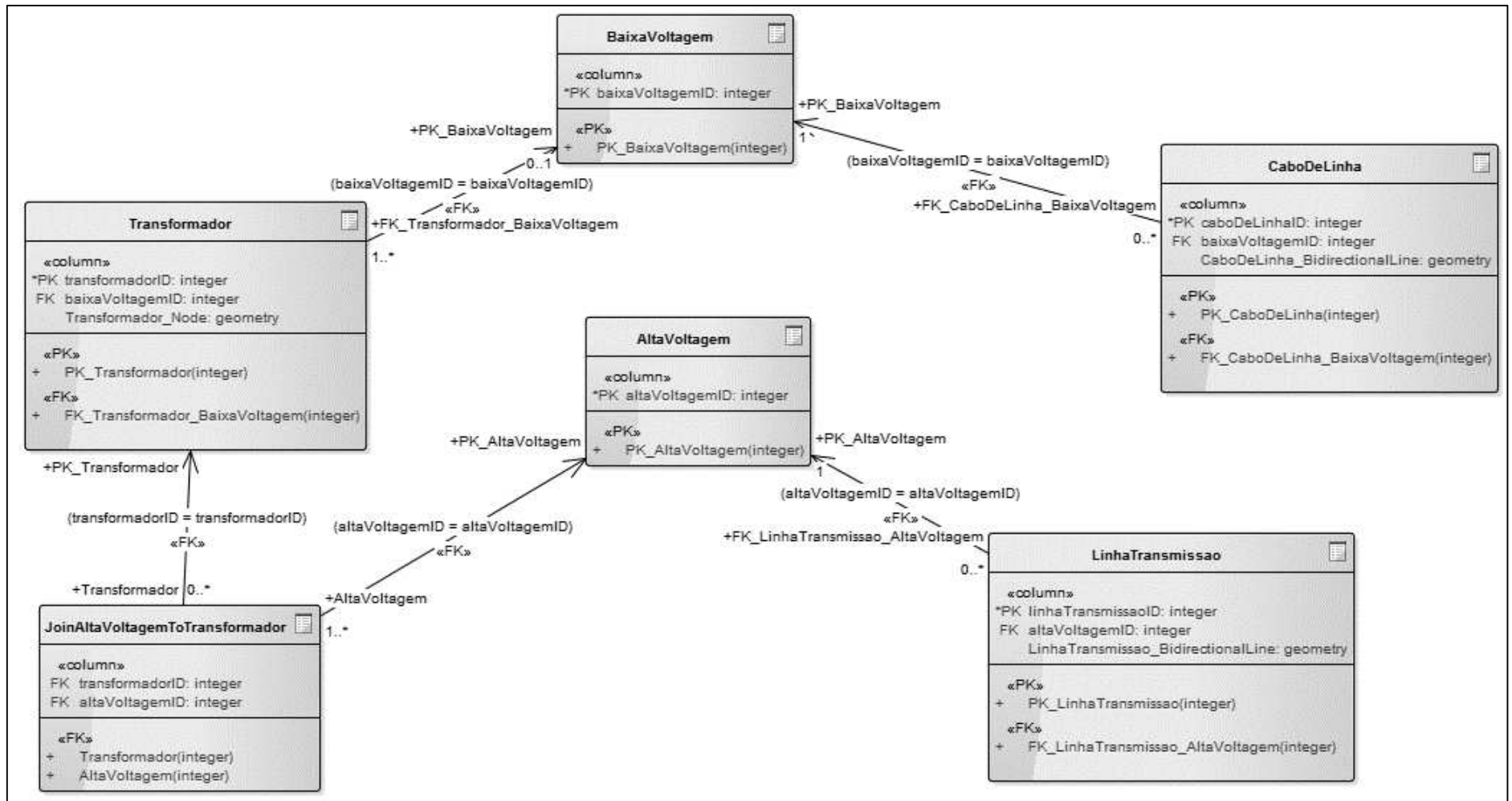


Figura 4.26: Esquema PSM Rede de Distribuição de Energia (OMTG).

```

DROP TABLE Transformador CASCADE;
DROP TABLE LinhaTransmissao CASCADE;
DROP TABLE CaboDeLinha CASCADE;
DROP TABLE BaixaVoltagem CASCADE;
DROP TABLE JoinAltaVoltagemToTransformador CASCADE;
DROP TABLE AltaVoltagem CASCADE;
CREATE TABLE Transformador (
    transformadorID integer NOT NULL,
    baixaVoltagemID integer,
    Transformador_Node geometry
);
CREATE TABLE LinhaTransmissao (
    linhaTransmissaoID integer NOT NULL,
    altaVoltagemID integer,
    LinhaTransmissao_BidirectionalLine geometry
);
ALTER TABLE Transformador ADD CONSTRAINT PK_Transformador
    PRIMARY KEY (transformadorID);
ALTER TABLE LinhaTransmissao ADD CONSTRAINT PK_LinhaTransmissao
    PRIMARY KEY (linhaTransmissaoID);
ALTER TABLE CaboDeLinha ADD CONSTRAINT PK_CaboDeLinha
    PRIMARY KEY (caboDeLinhaID);
ALTER TABLE BaixaVoltagem ADD CONSTRAINT PK_BaixaVoltagem
    PRIMARY KEY (baixaVoltagemID);
ALTER TABLE AltaVoltagem ADD CONSTRAINT PK_AltaVoltagem
    PRIMARY KEY (altaVoltagemID);
ALTER TABLE Transformador ADD CONSTRAINT FK_Transformador_BaixaVoltagem
    FOREIGN KEY (baixaVoltagemID) REFERENCES BaixaVoltagem (baixaVoltagemID);
ALTER TABLE LinhaTransmissao ADD CONSTRAINT FK_LinhaTransmissao_AltaVoltagem
    FOREIGN KEY (altaVoltagemID) REFERENCES AltaVoltagem (altaVoltagemID);
ALTER TABLE CaboDeLinha ADD CONSTRAINT FK_CaboDeLinha_BaixaVoltagem
    FOREIGN KEY (baixaVoltagemID) REFERENCES BaixaVoltagem (baixaVoltagemID);
ALTER TABLE JoinAltaVoltagemToTransformador ADD CONSTRAINT Transformador
    FOREIGN KEY (transformadorID) REFERENCES Transformador (transformadorID);
ALTER TABLE JoinAltaVoltagemToTransformador ADD CONSTRAINT AltaVoltagem
    FOREIGN KEY (altaVoltagemID) REFERENCES AltaVoltagem (altaVoltagemID);
CREATE TABLE CaboDeLinha (
    caboDeLinhaID integer NOT NULL,
    baixaVoltagemID integer,
    CaboDeLinha_BidirectionalLine geometry
);
CREATE TABLE BaixaVoltagem (
    baixaVoltagemID integer NOT NULL
);
CREATE TABLE JoinAltaVoltagemToTransformador (
    transformadorID integer,
    altaVoltagemID integer
);
CREATE TABLE AltaVoltagem (
    altaVoltagemID integer NOT NULL
);

```

Figura 4.27: Código DDL para criação do banco de dados Rede de Distribuição de Energia no SGBD PostGIS.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho mostrou que é possível alcançar interoperabilidade entre diferentes esquemas conceituais de banco de dados geográficos, utilizando, como base, o perfil UML GeoProfile. A ferramenta CASE Enterprise Architect foi utilizada na implementação do método proposto. Os modelos conceituais UML-GeoFrame, OMTG, GeoOOA e PVL e os requisitos representação de objetos básicos, representação de campos contínuos e representação de redes foram utilizados nesta dissertação para mostrar interoperabilidade de esquemas, na direção horizontal e na direção vertical.

A interoperabilidade horizontal foi exemplificada no nível PIM, da abordagem MDA, ou seja, um esquema conceitual foi transformado em um esquema equivalente, nos demais modelos, sendo antes transformado para um esquema no perfil GeoProfile.

Para mostrar que os esquemas são equivalentes, foi feita a transformação vertical de cada esquema conceitual, chegando em dois esquemas PSM também equivalentes. As transformações envolvendo os requisitos de representação de objetos básicos e representação de campos contínuos, além de serem especificadas foram implementadas na ferramenta EA.

A partir das tentativas de interoperabilidade entre o modelo PVL e o perfil GeoProfile, foi identificada a falta de construtores do GeoProfile para representação de múltiplos pontos, múltiplas linhas e múltiplos polígonos. Desta forma, neste trabalho foi apresentada uma proposta de extensão do GeoProfile.

A proposta apresentada neste trabalho possibilita que projetistas de BDG utilizem as ferramentas CASE da UML no projeto de esquemas conceituais de BDG. O projeto conceitual pode ser feito com os construtores e estereótipos específicos de cada modelo conceitual existente na literatura, e usando o perfil UML GeoProfile como base de interoperabilidade, esquemas conceituais podem ser mapeados para outros modelos conceituais. Espera-se que a partir da contribuição deste trabalho as pesquisas na área de BDG voltem-se para a tecnologia de perfil UML, evitando assim, o desenvolvimento de ferramentas CASE específicas e contribuindo para a padronização na área de modelagem.

Como trabalhos futuros, é importante desenvolver o metamodelo completo para um conjunto de modelos conceituais mais utilizados. Verificar se o perfil GeoProfile possui construtores suficientes para representar todos os requisitos de BDG, descritos em Pinet (2012). Estender, se necessário, o metamodelo do perfil GeoProfile, de forma que este possa servir de base para qualquer transformação entre esquemas conceituais de BDG. Implementar transformações automáticas na ferramenta EA para o requisito representação de rede.

Ainda como trabalhos futuros, implementar o método de transformação proposto, utilizando outras ferramentas CASE, principalmente as de código livre. Estas ferramentas devem possuir suporte a Perfil UML, restrições OCL e transformações MDA.

REFERÊNCIAS BIBLIOGRÁFICAS

- BÉDARD, Y.; LARRIVÉE, S. Modeling with Pictogrammic Languages. In: Shekhar, S.; Xiong, H. (Eds.). **Encyclopedia of GIS**. Berlin: Springer-Verlag, 2008. p. 716-725.
- BÉDARD, Y. Visual modeling of spatial databases: towards spatial PVL and UML. **Geomatica**, v. 53, n. 2, p. 169-185, 1999.
- BORGES, K. A. V. **Modelagem de dados Geográficos: Uma extensão do modelo OMT para aplicações Geográficas**. Dissertação(Mestrado)-Escola de Governo- Fundação João Pinheiro. Belo Horizonte, MG, Brasil, p. 139. 1997.
- 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.
- BORGES, K. A. V.; DAVIS Jr., C. A.; LAENDER, A. H. F. OMT-G: An Object Oriented Data Model for Geographic Applications. **GeoInformatica**, v.5, n.3, p. 221-260, set. 2001.
- CÂMARA, G. Representação computacional 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, cap. 1, p. 1-44, 2005.
- CHEN, P. P.-S. The entity-relationship model: toward a unified view of data. **ACM Transactions on Database Systems (TODS)**, 1, n. 1, 1976. 9-36.
- COAD, P.; YOURDON, E. **Object-Oriented Analysis**. 2nd ed. Englewoods Cliff: Prentice-Hall, 1991.
- 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.
- EA-ENTERPRISE ARCHITECT. Disponível em: <<http://www.sparxsystems.com>>. Acesso em: 19 Ago. 2015.
- ERIKSSON, H. et al. **UML 2 Toolkit**. Indianapolis: Wiley Publishing, 2004. 552p.
- FERREIRA, Thiago B.; STEPLIUC, Sergio M.; LISBOA-FILHO, Jugurta. Modelagem de Dados com o Perfil UML GeoProfile e Transformações MDA na Ferramenta Enterprise Architect. In: **9ª Conferencia Ibérica de Sistemas y Tecnologías de Informacion**. 2014. p. 603-608.2014.

- FUENTES, L.; VALLECILLO, A. An Introduction to UML Profiles. **UPGRADE, The European Journal for the Informatics Professional**, v. 5, n. 4, p. 5-13, abr. 2004.
- GÓMEZ, Leticia; VAISMAN, Alejandro; ZIMÁNYI, Esteban. Physical design and implementation of spatial data warehouses supporting continuous fields. In: **Data Warehousing and Knowledge Discovery**. Springer Berlin Heidelberg, 2010. p. 25-39.
- HUFNAGEL, Stephen P. Interoperability. **Military medicine**, v. 174, n. 5S, p. 43-50, 2009.
- KLEPPE, A. G.; WARMER, J. B.; BAST, W. **MDA explained: the model driven architecture: practice and promise**. 2ª. ed. Boston: Addison-Wesley Professional, 2003. 192p.
- KOSTERS, Georg; PAGEL, Bernd-Uwe; SIX, Hans-Werner. GIS-application development with GeoOOA. **International Journal of Geographical Information Science**, v. 11, n. 4, p. 307-335, 1997.
- LISBOA FILHO, Jugurta et al. A CASE tool for geographic database design supporting analysis patterns. In: **Conceptual Modeling for Advanced Application Domains**. Springer Berlin Heidelberg, 2004. p. 43-54.
- LISBOA FILHO, J.; IOCHPE, C. Modeling with a UML Profile. In: Shekhar, S.; Xiong, H. (Eds.). **Encyclopedia of GIS**. Berlin: Springer-Verlag, 2008. p. 691-700.
- LISBOA-FILHO, Jugurta et al. A uml profile for conceptual modeling in gis domain. In: **Proceedings of the International Workshop on Domain Engineering at CAISE. Hammamet, Tunisia**. 2010. p. 18-31.
- MARTINEZ, G. J. Z., **Modeling Environmental Niches and Potential Geographic Distributions Using a Formalism for Geospatial Database Design**. Universidade Federal de Viçosa. Viçosa, MG, Brasil p. 66. 2014.
- NALON, F. R. **Adequação de um Perfil UML para modelagem conceitual de bancos de dados geográficos aos padrões ISSO e OGC usando MDA**. Dissertação (Mestrado) - Universidade Federal de Viçosa, Viçosa-MG, p. 89. 2010.
- OBJECT MANAGEMENT GROUP. **MDA Guide**, v.1.0.1, OMG Document formal/2003-06-01 edition. Needham, MA, USA, 2003.
- OBJECT MANAGEMENT GROUP. **Unified Modeling Language: Superstructure**, v. 2.1.2, OMG Document formal/2007-11-02 edition. Needham, MA, USA, 2007.
- PARENT, C.; SPACCAPIETRA, S.; ZIMÁNYI, E. Modeling and Multiple Perceptions. In: Shear, S.; Xiong, H. (Eds.). **Encyclopedia of GIS**. Berlin: Springer-Verlag, 2008. p. 682-690.

- PERCEPTORY. Disponível em: <http://perceptory.scg.ulaval.ca/?page_id=53>. Acesso em: 14 mar. 2013.
- PINET, F. Entity-relationship and object-oriented formalisms for modeling spatial environmental data. **Environmental Modelling & Software**, 33, 2012. 80-91.
- RUMBAUGH, J. R.; BLAHA, M. R.; PREMERLANI, W.; EDDY, F.; LORENSEN, W. **Object-Oriented Modeling and Design**. UpperSaddle River: Prentice-Hall, 1991.
- SAMPAIO, G. B. **GeoProfile: Um Perfil UML para Modelagem Conceitual de Bancos de Dados Geográficos**. Dissertação (Mestrado) - Universidade Federal de Viçosa, Viçosa-MG, p. 65. 2009.
- STAUB, Peter. A model-driven web feature service for enhanced semantic interoperability. **OSGeo Journal**, v. 3, n. 1, 2007.
- STAUB, Peter; GNÄGI, Hans Rudolf; MORF, Andreas. Semantic interoperability through the definition of conceptual model transformations. **Transactions in GIS**, v. 12, n. 2, p. 193-207, 2008.
- STEMPLIUC, S. M. **Modelagem de restrições de integridade espaciais em aplicações de rede através do modelo UML-GeoFrame**. Dissertação (Mestrado) - Universidade Federal de Viçosa, Viçosa-MG. 2008.
- WARMER, J.; KLEPPE, A. **The Object Constraint Language: Getting Your Models Ready for MDA**. 2. ed. Boston: Addison Wesley, 2003.

APÊNDICE A- Customização da Ferramenta Enterprise Architect

A Enterprise Architect (EA) é uma ferramenta Computer-Aided Software Engineering (CASE) comercial licenciada pela Sparx System. No ano de 2014 foi feito um estudo por Ferreira et al., (2014) a respeito desta ferramenta CASE. Neste estudo foi constatado que a EA suporta perfis UML e pode ser usada para modelagem conceitual de Banco de Dados Geográficos (BDG).

Existem duas formas de se configurar um Perfil UML na EA. A primeira forma é comum a outras ferramentas CASE como por exemplo, Star UML, Visual Paradigm, Papyrus UML e Rational Software Architect. Neste tipo de configuração o perfil UML é usado diretamente na ferramenta. Os recursos oferecidos são, estereótipos, tag values e restrições. A Figura 1 apresenta um esquema conceitual de BDG modelado a partir dos construtores do GeoProfile na ferramenta EA. A configuração utilizada para modelar este esquema foi a descrita anteriormente. Maiores detalhes sobre este tipo de configuração podem ser conferidos no site do projeto GeoProfile²

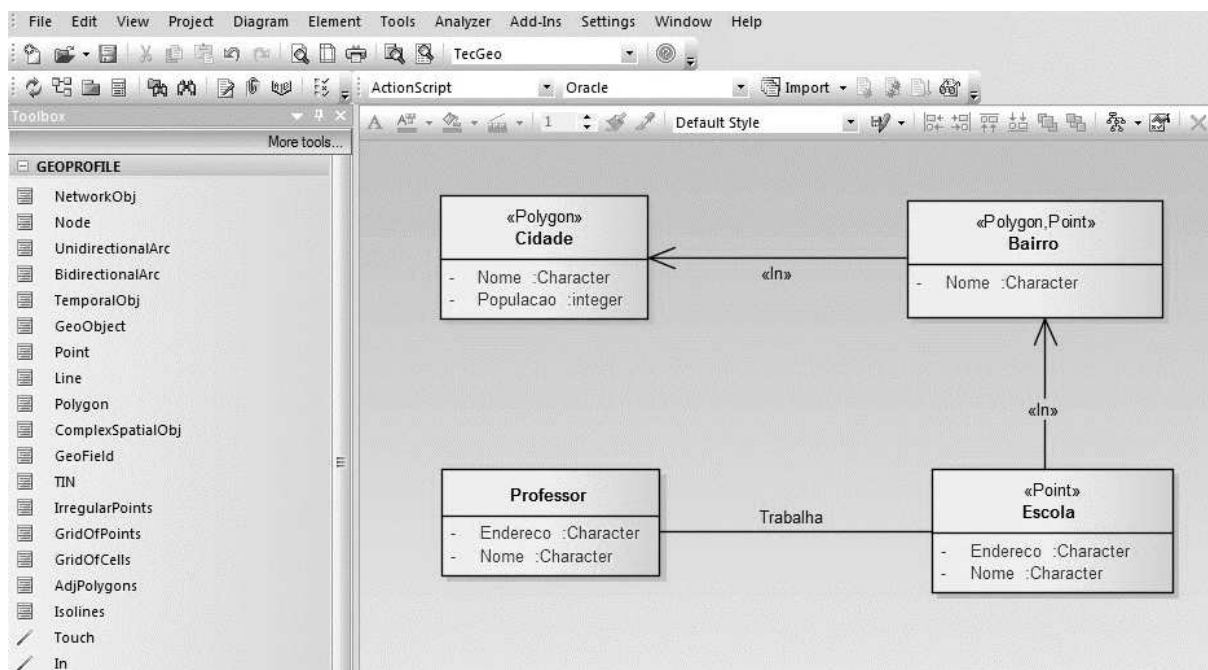


Figura 1- Utilização do Perfil GeoProfile na ferramenta EA – Primeira forma.

A segunda forma de configuração é utilizando a tecnologia Model Driven Generation (MDG). Neste tipo de configuração o perfil UML também é criado na ferramenta, no entanto seu uso não é feito diretamente como na primeira forma. O perfil UML é submetido a

² <http://www.dpi.ufv.br/projetos/geoprofile/>

tecnologia MDG, onde pode ser adicionado scripts que alteram o formato dos construtores da UML, estereótipos gráficos, profiles toolboxes para organização da caixa de ferramentas, códigos de transformação e etc. Utilizando a segunda forma de configuração da ferramenta EA a Figura 2 ilustra o mesmo esquema conceitual da Figura 1.

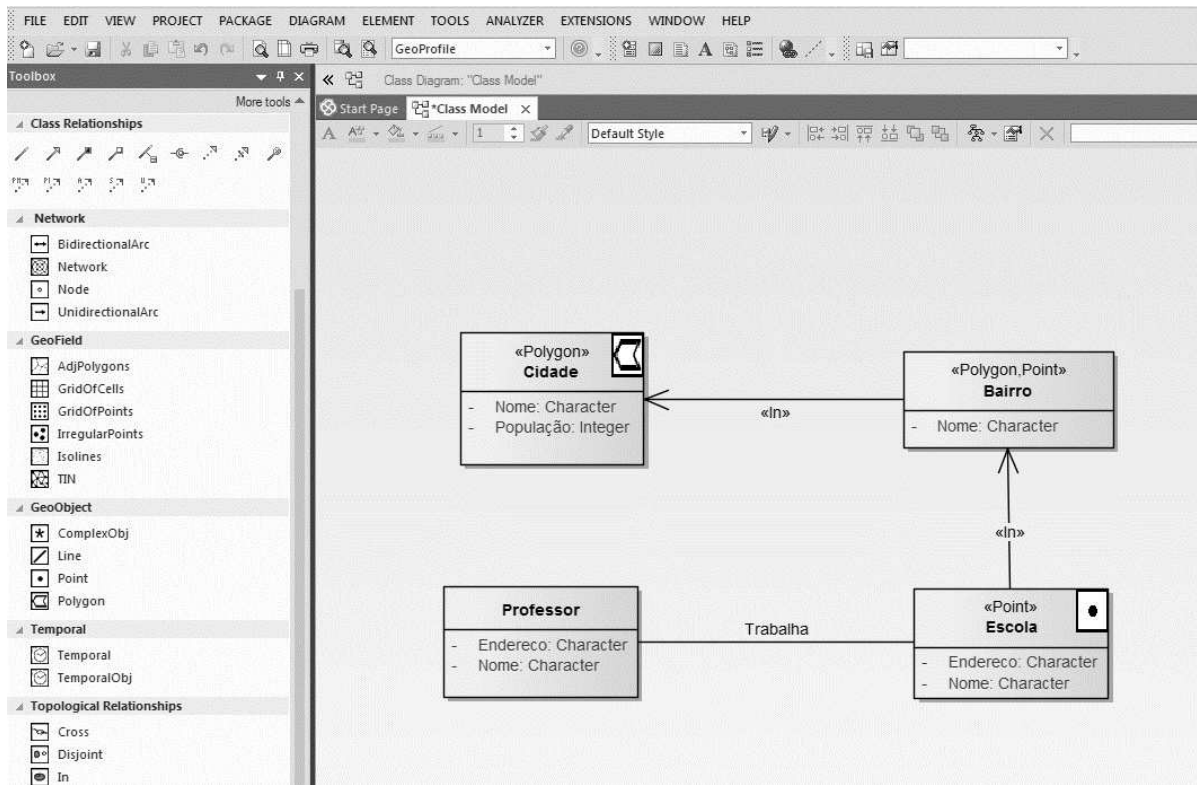


Figura 2- Utilização do Perfil GeoProfile na ferramenta EA – Segunda forma.

Além das diferenças explicitas na entre a Figura 1 e 2 existem outros benefícios em se usar a tecnologia MDG, por exemplo, regras de transformações.

A tecnologia MDG une todos os recursos em um arquivo no formato XML, desta forma, tanto o perfil UML quanto as características agregadas ao perfil UML podem ser facilmente transmitidos para outros usuários da EA.

Neste tutorial será apresentado os passos para o desenvolvimento da tecnologia MDG para o Perfil UML GeoProfile.

O GeoProfile

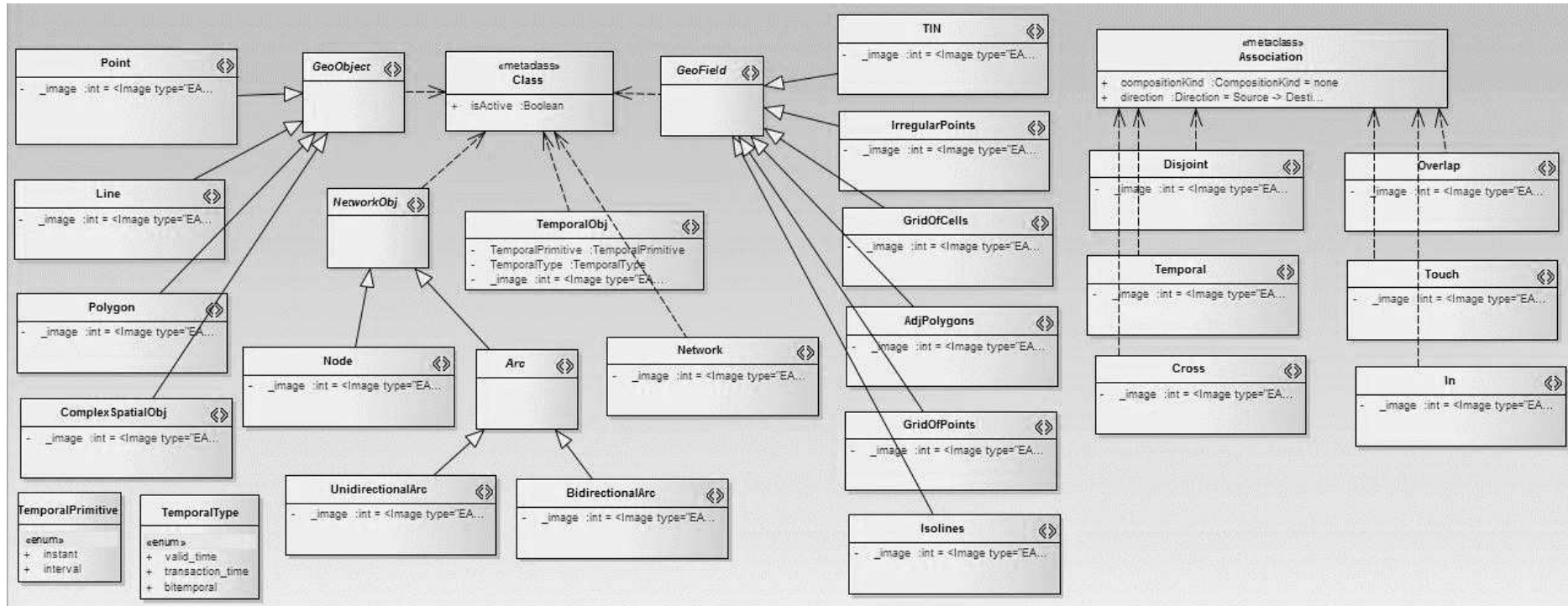


Figura 3- Perfil UML GeoProfile configurado na ferramenta Enterprise Architect.

O GeoProfile é um perfil UML usado para modelagem conceitual BDG. Este perfil pode ser configurado em ferramentas CASE da UML. A Figura 3 apresenta o GeoProfile configurado na ferramenta CASE EA.

Configuração do GeoProfile na ferramenta Enterprise Architect

No ambiente de modelagem da EA, por padrão é configurado uma caixa de ferramenta na lateral esquerda da tela, a ToolBox. Nesta caixa de ferramentas são encontrados os construtores para modelagem de Perfis UML. A Figura 4 apresenta a toolbox com os construtores de perfis carregados. Para encontrar os construtores de Profile vá em **More tools...**



Figura 4- Caixa de ferramentas com os construtores de perfis.

Para configurar o GeoProfile na EA os seguintes passos devem ser seguidos:

1. Crie um pacote Profile e dê o nome de GeoProfile;
2. Dentro do pacote GeoProfile, devem ser colocados as metaclasses, os estereótipos e enumerações. A Figura 3 apresenta como esses elementos devem ser associados dentro do pacote GeoProfile. As setas tracejadas são extensões e são usadas em metaclasses. As setas contínuas são generalizações e são usadas em estereótipos.
3. Os atributos `_image` que estão nos estereótipos serão explicados na próxima seção.

4. Para salvar o profile é preciso clicar com o botão direito do mouse sobre uma parte vazia da tela. No caso da Figura 3 na parte azul. Selecionar a opção **Advanced->Save as Profile**. Ao executar esse passo um arquivo XML será criado.

Maiores detalhes sobre como criar o perfil UML GeoProfile, e preparação do ambiente da EA podem ser encontrados nos tutoriais do site do GeoProfile³.

Preparando o ambiente para uso da tecnologia MDG para o GeoProfile Adicionando ícones gráficos aos estereótipos do GeoProfile

Antes de salvar o Perfil UML como mostrado no passo 4, é importante colocar o atributo `_image` nos estereótipos do GeoProfile. Para isso os seguintes passos devem ser seguidos:

1. Os ícones do GeoProfile devem ser carregados para a ferramenta EA. Vá em **Settings>images**. A Figura 5 apresenta a janela onde os ícones do GeoProfile podem ser adicionados. Os ícones do GeoProfile podem ser encontrados no site do projeto GeoProfile.

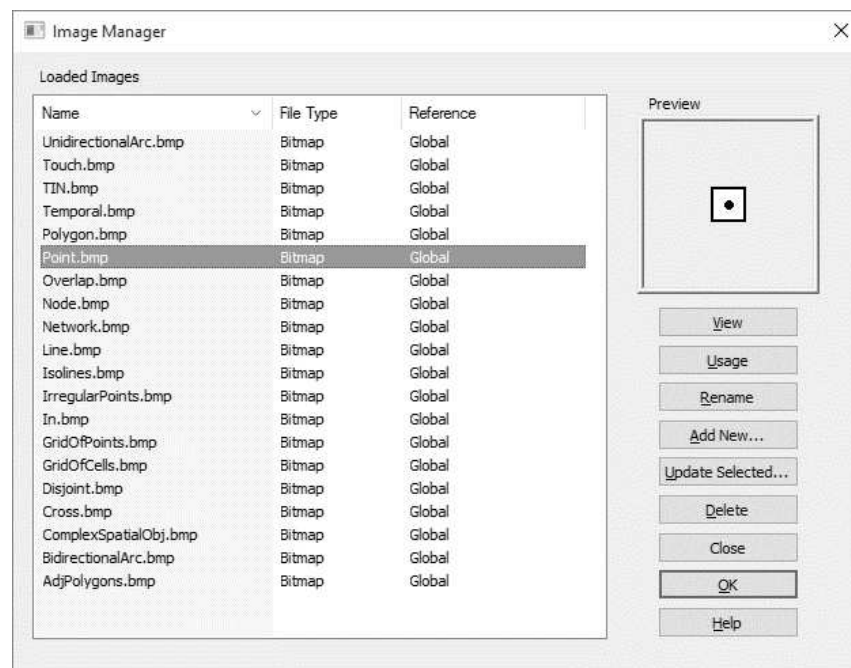


Figura 5- Adicionando imagens na Enterprise Architect.

³ http://www.dpi.ufv.br/projetos/geoprofile/tutorial_Architect.html

2. Selecione individualmente cada estereótipo do GeoProfile que necessite de ícones gráficos. Aperte a tecla F9 ou pressione o botão direito do mouse e vá em **Features & Properties** -> **Attributes**. Preencha os campos como na Figura 6. E clique na reticências do campo **Initial Value**.

Name:	<input type="text" value="_image"/>	
Type:	<input type="text" value="int"/>	...
Initial Value:	<input type="text" value=""/>	...
Stereotype:	<input type="text"/>	...
Alias:	<input type="text"/>	

Figura 6- Campos de atributos a serem preenchidos.

3. Após ter clicado nas reticências do campo **Initial Value** a janela da Figura 7 deve ser exibida. Nesta Janela o script da Figura 7 deve ser adicionado a cada estereótipo. A única alteração a ser feita é o nome da imagem a ser acessada e o nome da classe decoration. O exemplo desta Figura refere-se ao estereótipo point.



Figura 7- Script para adição de ícones aos estereótipos do GeoProfile.

Especificando Profile Toolbox

Apesar de ter salvo o GeoProfile em um arquivo no formato XML é importante que o mesmo continue aberto na EA.

Para especificar o profile Toolbox é preciso criar um novo pacote estereotipado <<Profile>>. Uma sugestão de nome é GeoProfile_ToolBox. Depois de ter criado o pacote estereotipado, os seguintes passos devem ser seguidos:

1. Na aba **Profile Helpers**, apresentada na Figura 4, clicar em **Create Custom Toolbox** e clicar em um espaço vazio. Ao clicar a Janela da Figura 8 será exibida. Selecione o pacote criado anteriormente, e confirme a opção.



Figura 8- Janela para configuração Create Custom Toolbox

2. Com a opção confirmada, aparecerá uma Janela como a da Figura 9 onde, os campos nome do Toolbox e descrição deverão ser preenchidos. Ao clicar em **Ok** a metaclasses `ToolboxPage` será exibida na tela.

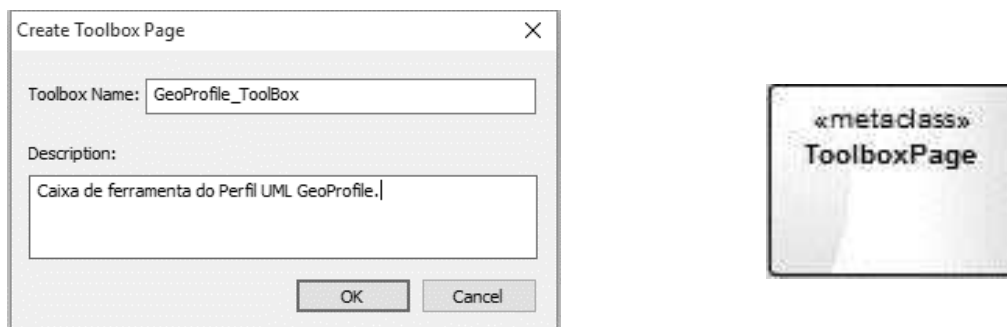


Figura 9- Criando a metaclasses `ToolboxPage`.

3. Clicar em **Add Tolbox Page** na aba **Profile Helpers** da janela apresentada na Figura 4. As Janelas apresentadas na Figura 10 demonstra como configurar cada paleta de estereótipos do GeoProfile na ferramenta. No exemplo desta Figura está sendo criado a Paleta de Relacionamentos espaciais. Observe que todos os estereótipos de relacionamentos espaciais do pacote GeoProfile devem ser adicionados a esta paleta. Toolbox icon é um recurso para adicionar ícones na caixa de ferramenta da EA. Estes ícones precisam ser de no máximo 16x16 pixels.
4. Os procedimentos apresentados no Passo 3, devem ser repetidos para estereótipos de GeoObjetos, GeoCampos, Redes e Temporal. A Figura 11 apresenta a configuração final do profile Toolbox. A metaclasses `ToolboxImage` e os estereótipos ligados a ela são gerados automaticamente. Da mesma forma que o profile GeoProfile o profileToolbox do GeoProfile deve ser salvo em um arquivo XML.

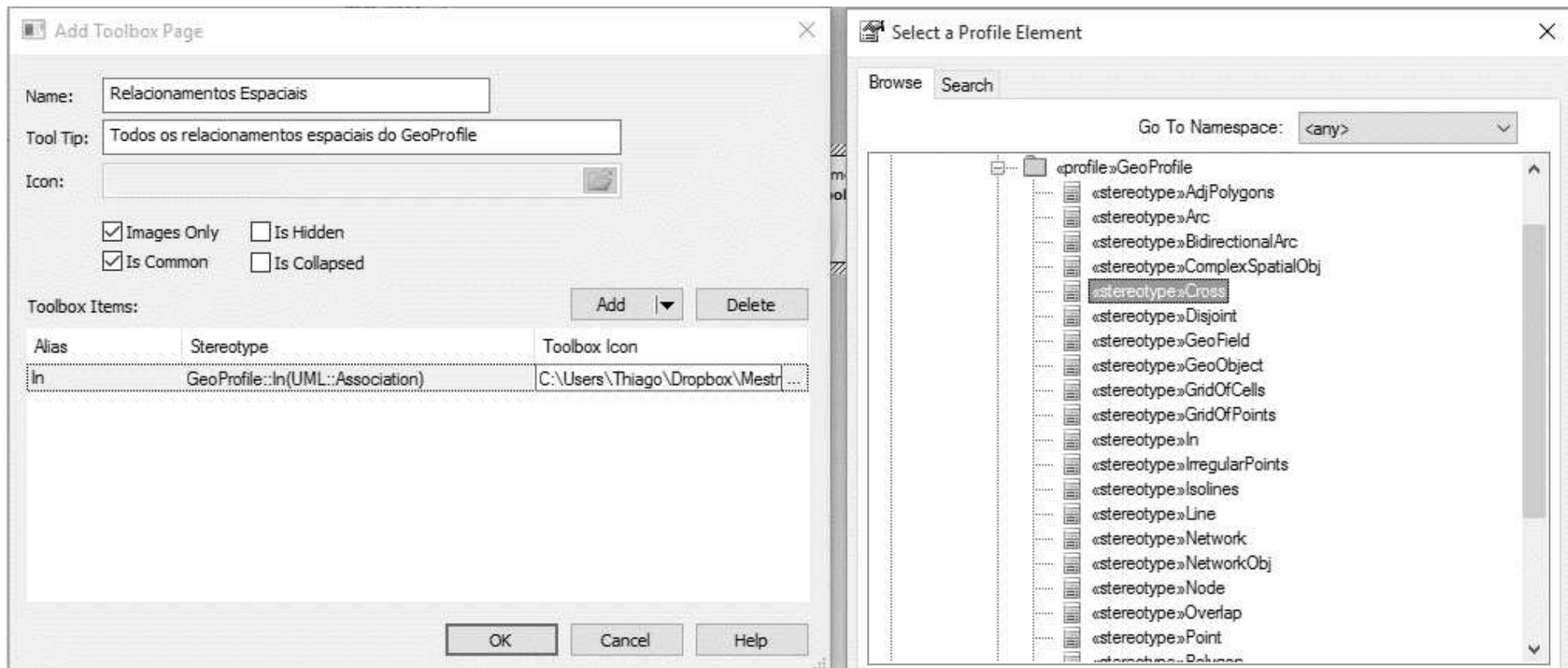


Figura 10- Configuração do Toolbox Pages do GeoProfile.

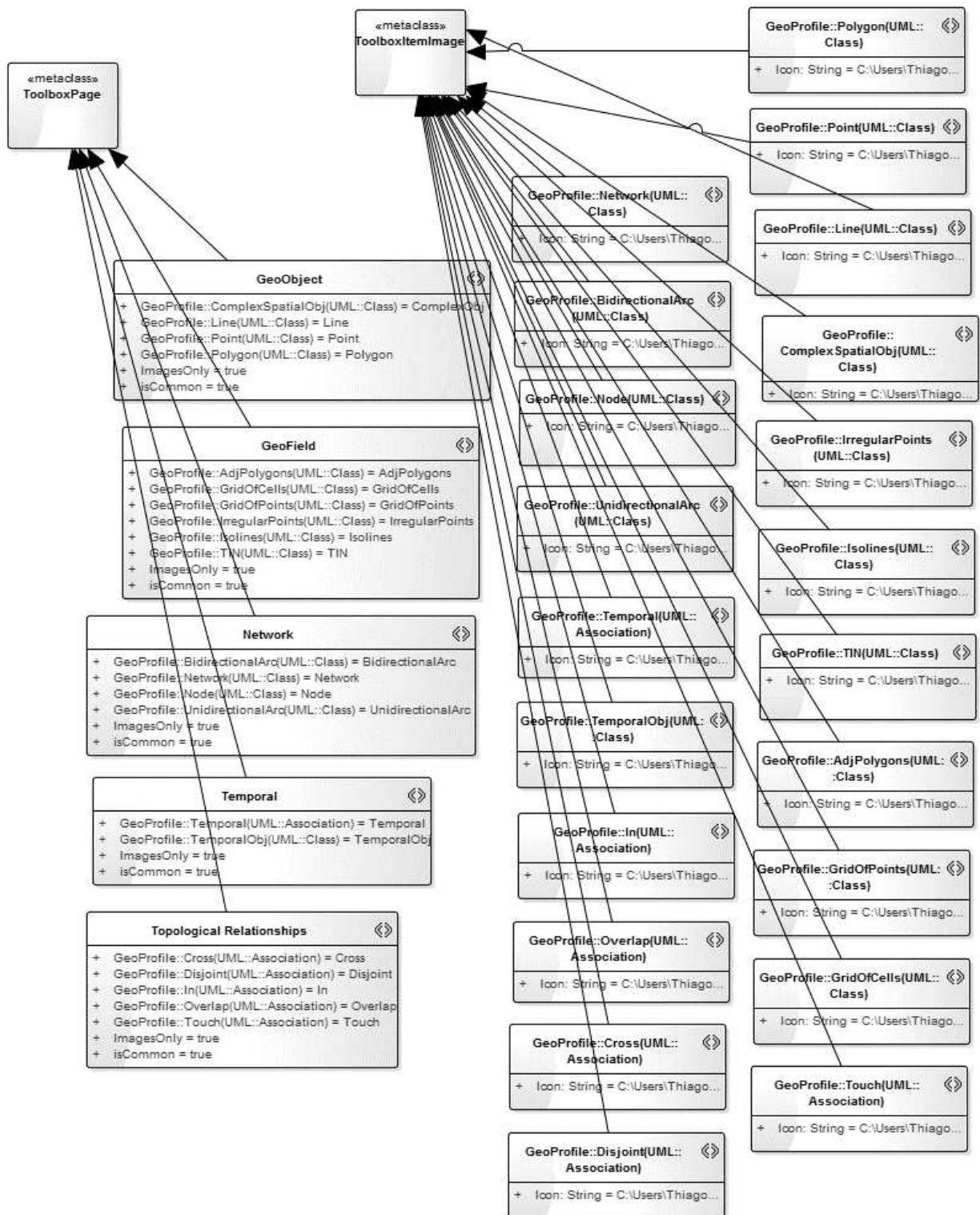


Figura 10- Configuração Final do Profile ToolBox do GeoProfile.

Especificando Regras de Transformações

A ferramenta EA oferece uma linguagem para especificação de regras de transformações. As transformações podem ser verticais ou horizontais.

A Figura 11 apresenta o ambiente de transformação de Templates da EA. No exemplo desta Figura, a transformação é horizontal do GeoProfile para o GeoOOA, como pode ser observado no campo Language. Para acessar esta Janela clique em **Tools->MDA Transformation Templates..** Detalhes sobre a sintaxe da linguagem de transformação, podem ser encontrados no **help** da ferramenta. Os códigos de transformações horizontais e verticais desenvolvidos neste trabalho podem ser encontrados no Apêndice B.



Figura 11- Janela de Transformação de Templates.

Gerando o XML da Tecnologia MDG

A tecnologia MDG permite criar um arquivo no formato XML que une todos os recursos desenvolvidos na seção anterior. Por exemplo, Profile, toolboxes customizados, transformações customizadas e arquivos de imagens.

Para ter acesso a esta tecnologia clique em **Tools->Generation Technology File ..** Após a execução deste passo, uma Janela será exibida. Nesta Janela uma descrição sobre a tecnologia MDG será apresentada, clique em **Avançar**. A próxima janela a exibida refere-se a prévia customização (MTS File), este recurso não será usado no GeoProfile. Deixe tudo da forma como está e clique em **Avançar**. A Figura 12 apresenta a Janela onde será dado o nome da tecnologia MDG, o ícone, a descrição e diretório onde o arquivo XML será salvo. Preencha como ilustrado na Figura, e clique em **Avançar**.

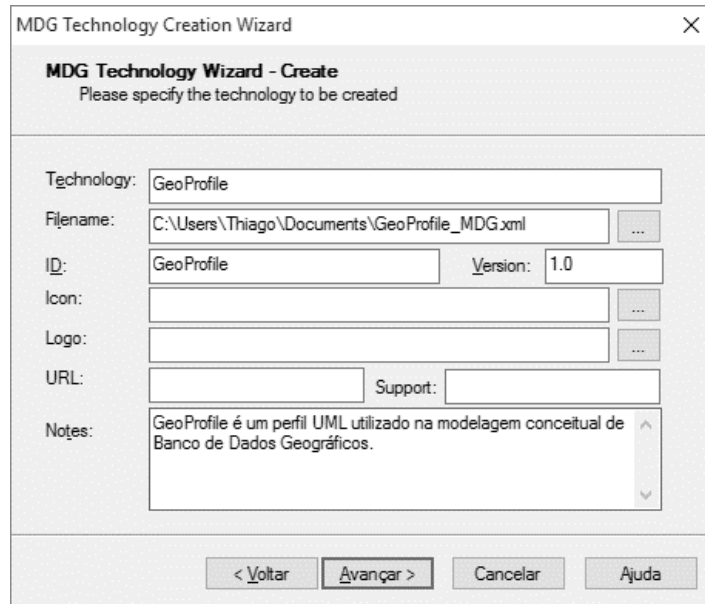


Figura 12- Janela para especificação da tecnologia que está sendo criada.

Na janela apresentada pela Figura 13, deve-se selecionar os recursos que deseja-se incluir na tecnologia MDG do GeoProfile. Neste caso, Profiles, Toolboxes, imagens e MDA Transforms. Em seguida clicar em **Avançar**.

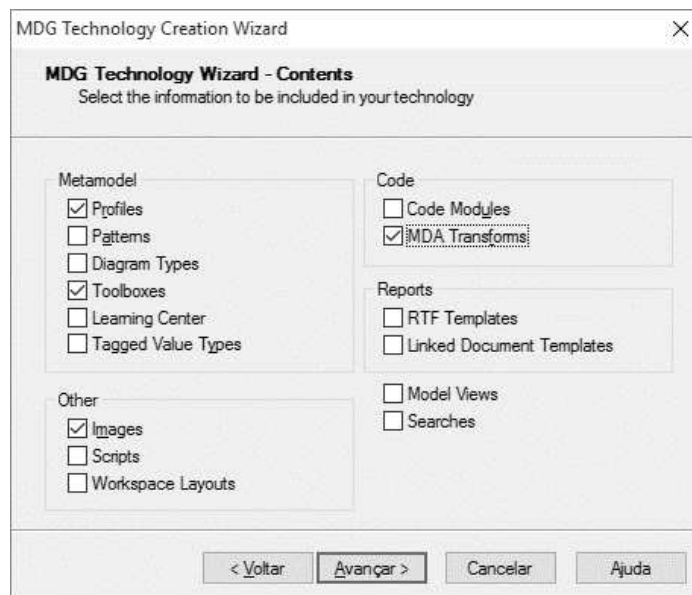


Figura 13- Recursos desejados na Tecnologia MDG.

As duas próximas Janelas, apresentadas na Figura 14 são utilizadas para adicionar respectivamente o profile, e o profile toolboxes. Para adiciona-los é preciso indicar o diretório dos arquivos XML. Selecionar o arquivo XML desejado e clicar na seta em azul da Figura 14. Logo em seguida clicar em **avançar**.

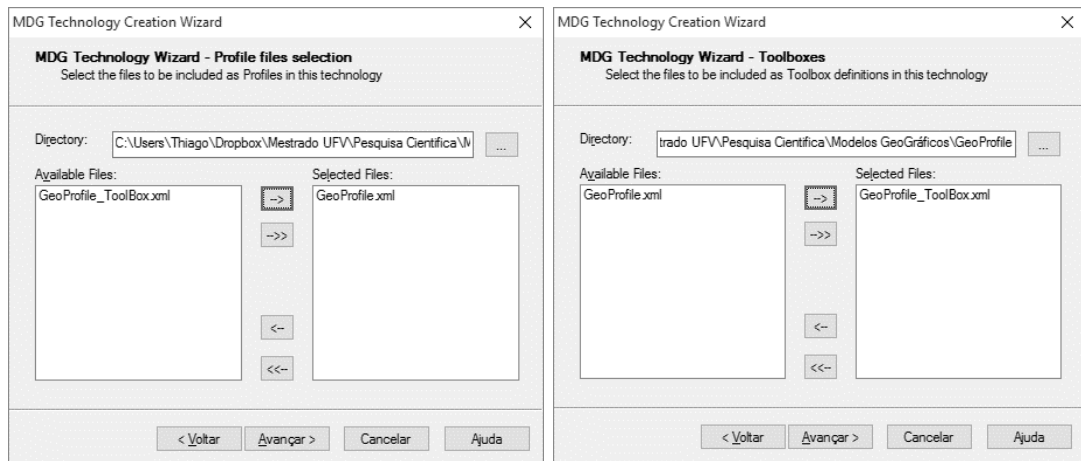


Figura 14- Adicionando o profile e o profile ToolBoxes.

As janelas exibidas na Figura 15 apresentam respectivamente os módulos de códigos de transformações que podem ser selecionados e os ícones gráficos carregados na ferramenta. Após selecionar os módulos de transformações desejados clique em **Avançar**. Após selecionar os ícones gráficos desejados clique em **Avançar**.

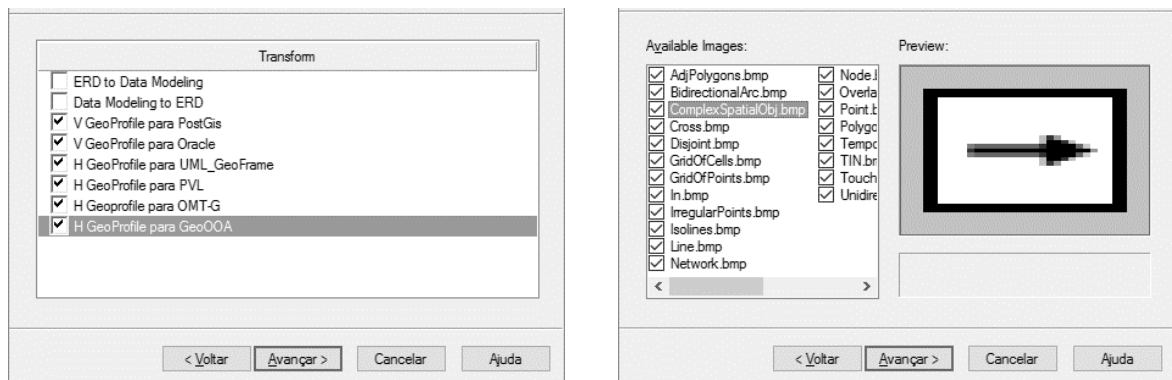


Figura 15- Adicionando os códigos transformações e ícones gráficos à tecnologia MDG.

Após a execução dos passos anteriores a Janela apresentada na Figura 16 será exibida. Bastando clicar em concluir para que o arquivo XML MDG seja gerado.

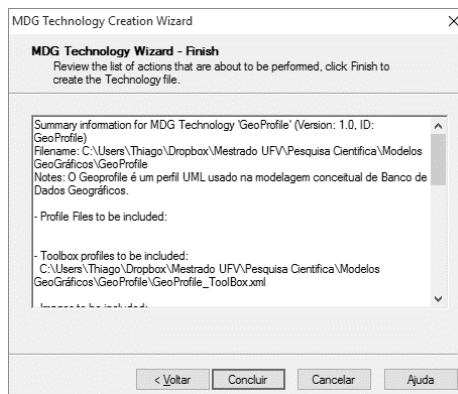


Figura 16- Janela exibida para geração de arquivos XML.

Usando a Tecnologia MDG do GeoProfile

Para utilizar a tecnologia MDG do GeoProfile na ferramenta basta, criar um novo projeto na ferramenta EA e clicar em **Tools-> MDG Technology import** e importar o arquivo XML. Após ter importado, o próximo passo é mudar o ambiente para o GeoProfile. O campo apontado pela seta na Figura 17 mostra como mudar o ambiente para GeoProfile. Nesta Figura também são apresentados os Toolbox Customizado, como pode ser observado no canto esquerdo da tela.

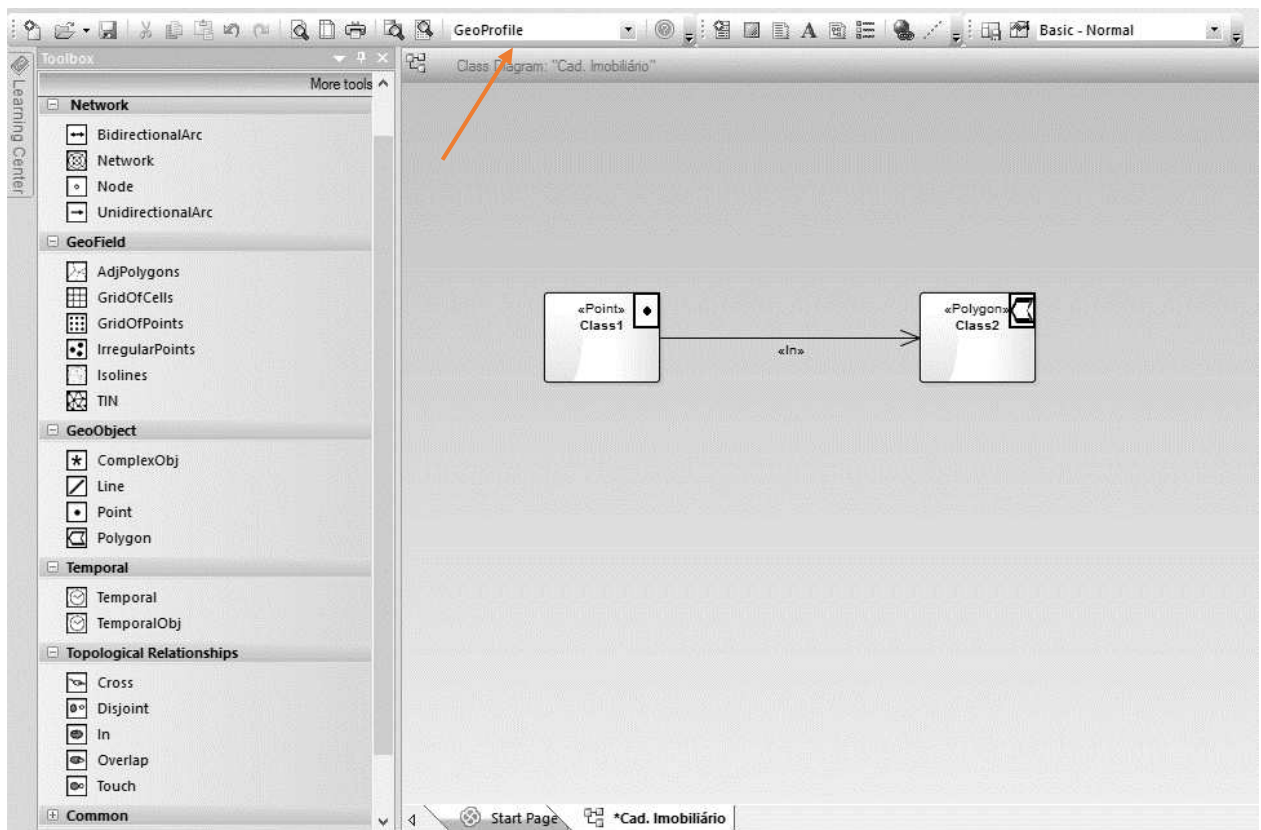


Figura 17- Ambiente do GeoProfile na ferramenta EA.

Para efetuar as transformações horizontais e verticais clique em **Tools -> Model Transformation -> Transform Current Package** escolha a transformação desejada e execute a transformação.

APÊNDICE B- scripts que devem ser colocados para alterar o formato da classe

Este apêndice apresenta os scripts que devem ser colocados para alterar o formato da classe. Com alteração do formato da classe um ambiente de modelagem conceitual semelhante ao do modelo conceitual que esta sendo replicado na tecnologia MDG.

MDG_GEOPROFILE	
<pre>//Point decoration Point { orientation = "NE"; image("Point.bmp", -10, -15, 120, 160); }</pre>	<pre>//Node decoration Node { orientation = "NE"; image("Node.bmp", -10, -15, 120, 160); }</pre>
<pre>//Line decoration Line { orientation = "NE"; image("Line.bmp", -10, -15, 120, 160); }</pre>	<pre>//BidirectionalArc decoration BidirectionalArc { orientation = "NE"; image("BidirectionalArc.bmp", -10, -15, 120, 160); }</pre>
<pre>//Polygon decoration Polygon { orientation = "NE"; image("Polygon.bmp", -10, -15, 120, 160); }</pre>	<pre>//UnidirectionalArc decoration UnidirectionalArc { orientation = "NE"; image("UnidirectionalArc.bmp", -10, -15, 120, 160); }</pre>
<pre>//ComplexSpatialObj decoration ComplexSpatialObj { orientation = "NE"; image("ComplexSpatialObj.bmp", -10, -15, 120, 160); }</pre>	<pre>//Isolines decoration Isolines { orientation = "NE"; image("Isolines.bmp", -10, -15, 120, 160); }</pre>
<pre>//MultiPoints decoration MultiPoints { orientation = "NE"; image("MultiPoints.bmp", -5, -10, 105, 110); }</pre>	<pre>//AdjPolygons decoration AdjPolygons { orientation = "NE"; image("AdjPolygons.bmp", -10, -15, 120, 160); }</pre>
<pre>//MultiLines decoration MultiLines { orientation = "NE"; image("MultiLines.bmp", -5, -10, 105, 110); }</pre>	<pre>//IrregularPoints decoration IrregularPoints { orientation = "NE"; image("IrregularPoints.bmp", -10, -15, 120, 160); }</pre>
<pre>//MultiPolygons decoration MultiPolygons { orientation = "NE"; image("MultiPolygons.bmp", -5, -10, 105, 110); }</pre>	<pre>//GridOfCells decoration GridOfCells { orientation = "NE"; image("GridOfCells.bmp", -10, -15, 120, 160); }</pre>
<pre>//Network decoration Network { orientation = "NE"; image("Network.bmp", -10, -15, 120, 160); }</pre>	<pre>//GridOfPoints decoration GridOfPoints { orientation = "NE"; image("GridOfPoints.bmp", -10, -15, 120, 160); }</pre>
<pre>//BidirectionalArc decoration BidirectionalArc { orientation = "NE"; image("BidirectionalArc.bmp", -10, -15, 120, 160); }</pre>	<pre>//TIN decoration TIN { orientation = "NE"; image("TIN.bmp", -10, -15, 120, 160); }</pre>

MDG_PVL	
<pre>//Point decoration Point { orientation = "NE"; image("Point.bmp", -10, -15, 120, 160); }</pre>	<pre>//MultiPoints decoration MultiPoints { orientation = "NE"; image("MultiPoints.bmp", -5, 0, 105, 105); }</pre>
<pre>//Line decoration Line { orientation = "NE"; image("Line.bmp", -10, -15, 120, 160); }</pre>	<pre>//MultiPolygons decoration MultiPolygons { orientation = "NE"; image("MultiPolygons.bmp", -5, 0, 105, 105); }</pre>
<pre>//Polygon decoration Polygon { orientation = "NE"; image("Polygon.bmp", -10, -15, 120, 160); }</pre>	<pre>//MutLines decoration MutLines { orientation = "NE"; image("MutLines.bmp", -5, 0, 105, 105); }</pre>
<pre>//ComplexShape decoration ComplexShape { orientation = "NE"; image("ComplexShape.bmp", -5, 0, 105, 105); }</pre>	
MDG_GeoOOA	
<pre>//Point shape main { rectangle(0,0,100,100); // (x1,y1,x2,y2) rectangle(5,5,95,95); // (x1,y1,x2,y2) addsubshape("namecompartment", 100, -50); shape namecompartment { h_align = "center"; editablefield = "name"; println("#name#"); } } decoration Point { orientation = "NE"; image("Point.jpg", 130, 10, 10,130); }</pre>	<pre>//Line shape main { rectangle(0,0,100,100); // (x1,y1,x2,y2) rectangle(5,5,95,95); // (x1,y1,x2,y2) addsubshape("namecompartment", 100, -50); shape namecompartment { h_align = "center"; editablefield = "name"; println("#name#"); } } decoration Line { orientation = "NE"; image("Line.jpg", 130, 10, 10,130); }</pre>
<pre>//Region shape main { rectangle(0,0,100,100); // (x1,y1,x2,y2) rectangle(5,5,95,95); // (x1,y1,x2,y2) addsubshape("namecompartment", 100, -50); shape namecompartment { h_align = "center"; editablefield = "name"; println("#name#"); } } decoration Region { orientation = "NE"; image("Region.jpg", 130, 10, 10,130); }</pre>	<pre>//Network shape main { rectangle(0,0,100,100); // (x1,y1,x2,y2) rectangle(5,5,95,95); // (x1,y1,x2,y2) addsubshape("namecompartment", 100, -50); shape namecompartment { h_align = "center"; editablefield = "name"; println("#name#"); } } decoration Network { orientation = "NE"; image("Network.jpg", 130, 10, 10,130); }</pre>

<pre> } //convencional shape main { rectangle(0,0,100,100); rectangle(5,5,95,95); addsubshape("namecompartment", 100, -50); shape namecompartment { h_align = "center"; editablefield = "name"; println("#name#"); shape ChildElement { if(HasProperty("stereotype", "property")) { SetCompartmentName("Properties"); } if(HasProperty("scope", "public")) { AppendCompartmentText("+"); } AppendCompartmentText("#NAME#"); } } } </pre>	<pre> } //Node shape main { image("Node.jpg", 0,0,100,100); } //Link shape main { image("Link.jpg", 0,0,100,100); } </pre>
MDG_OMTG	
<pre> //Point decoration Point { orientation = "NW"; image("Ponto.bmp", -10, -15, 120, 160); } </pre>	<pre> //BidirectionalLine decoration BidirectionalLine { orientation = "NW"; image("LinhaBi.bmp", -10, -15, 120, 160); } </pre>
<pre> //Line decoration Line { orientation = "NW"; image("Linha.bmp", -10, -15, 120, 160); } </pre>	<pre> //UnidirectionalLine decoration UnidirectionalLine { orientation = "NW"; image("LinhaUni.bmp", -10, -15, 120, 160); } </pre>
<pre> //Polygon decoration Polygon { orientation = "NW"; image("Poligono.bmp", -10, -15, 120, 160); } </pre>	<pre> //NetworkAssociation shape main{ // draw a dashed line noshadow = "true"; setlinestyle ("dash"); moveto(0,-60); lineto(100,-60); moveto(0,80); lineto(100,80); } </pre>
<pre> //TIN decoration TIN { orientation = "NW"; image("TIN.bmp", -10, -15, 120, 160); } </pre>	<pre> //Isolines decoration Isolines { orientation = "NW"; image("Isolinha.bmp", -10, -15, 120, 160); } </pre>
<pre> //PlanarSubdivision decoration PlanarSubdivision { orientation = "NW"; image("PoligonoAdj.bmp", -10, -15, 120, 160); } </pre>	<pre> //Tessalation decoration Tessalation { orientation = "NW"; image("Tesselacao.bmp", -10, -15, 120, 160); } </pre>
<pre> //Sampling decoration Sampling { } </pre>	

<pre> orientation = "NW"; image("Amostragem.bmp", -10, -15, 120, 160); } </pre>	
MDG_UML-GeoFrame	
<pre> //Point decoration ObjGeo { orientation = "NW"; image("ObjGeo.bmp", -10, -15, 120, 160); } decoration Point { orientation = "NE"; image("Point.bmp", -10, -15, 120, 160); } </pre>	<pre> //Line decoration ObjGeo { orientation = "NW"; image("ObjGeo.bmp", -10, -15, 120, 160); } decoration Line { orientation = "NE"; image("Line.bmp", -10, -15, 120, 160); } </pre>
<pre> //Polygon decoration ObjGeo { orientation = "NW"; image("ObjGeo.bmp", -10, -15, 120, 160); } decoration Polygon { orientation = "NE"; image("Polygon.bmp", -10, -15, 120, 160); } </pre>	<pre> //ComplexSpatialObj decoration ObjGeo { orientation = "NW"; image("ObjGeo.bmp", -10, -15, 120, 160); } decoration ComplexSpatialObj { orientation = "NE"; image("ComplexSpatialObj.bmp", -10, -15, 120, 160); } </pre>
<pre> //Node decoration Network { orientation = "NW"; image("Network.bmp", -10, -15, 120, 160); } decoration Node { orientation = "NE"; image("Node.bmp", -10, -15, 120, 160); } </pre>	<pre> //Network decoration Network { orientation = "NW"; image("Network.bmp", -10, -15, 120, 160); } </pre>
	<pre> //Conventional decoration Conventional { orientation = "NW"; image("ObjNaoGeo.bmp", -10, -15, 120, 160); } </pre>
<pre> //UnidirectionalArc decoration Network { orientation = "NW"; image("Network.bmp", -10, -15, 120, 160); } decoration UnidirectionalArc { orientation = "NE"; image("UnidirectionalArc.bmp", -10, -15, 120, 160); } </pre>	<pre> //BidirectionalArc decoration Network { orientation = "NW"; image("Network.bmp", -10, -15, 120, 160); } decoration BidirectionalArc { orientation = "NE"; image("BidirectionalArc.bmp", -10, -15, 120, 160); } </pre>
<pre> //GridOfPoints decoration GeoField { orientation = "NW"; image("GeoField.bmp", -10, -15, 120, 160); } decoration GridOfPoints { orientation = "NE"; image("GridOfPoints.bmp", -10, -15, 120, 160); } </pre>	<pre> //GridOfCells decoration GeoField { orientation = "NW"; image("GeoField.bmp", -10, -15, 120, 160); } decoration GridOfCells { orientation = "NE"; image("GridOfCells.bmp", -10, -15, 120, 160); } </pre>
<pre> //TIN decoration GeoField { orientation = "NW"; image("GeoField.bmp", -10, -15, 120, 160); } decoration TIN { </pre>	<pre> //Isolines decoration GeoField { orientation = "NW"; image("GeoField.bmp", -10, -15, 120, 160); } decoration Isolines { </pre>

<pre> orientation = "NE"; image("TIN.bmp", -10, -15, 120, 160); } </pre>	<pre> orientation = "NE"; image("Isolines.bmp", -10, -15, 120, 160); } </pre>
<pre> //IrregularPoints decoration GeoField { orientation = "NW"; image("GeoField.bmp", -10, -15, 120, 160); } decoration IrregularPoints { orientation = "NE"; image("IrregularPoints.bmp", -10, -15, 120, 160); } </pre>	<pre> //AdjPolygons decoration GeoField { orientation = "NW"; image("GeoField.bmp", -10, -15, 120, 160); } decoration AdjPolygons { orientation = "NE"; image("AdjPolygons.bmp", -10, -15, 120, 160); } </pre>

APÊNDICE C- templates modificados para transformação horizontal

Este apêndice apresenta os templates que foram modificados para que fosse possível a transformação horizontal dos esquemas conceituais de Banco de Dados Geográficos. Ao replicar os testes, os templates que não foram apresentados neste apêndice podem ser deixados com o código default.

MDG_GeoProfile para MDG-GeoOOA	
Template	Código de Transformação
<i>File</i>	<pre>Package { name="GeoProfile(PIM) para GeoOOA (PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ }</pre>
<i>Class</i>	<pre>/ if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class")/ / TRANSFORM_CURRENT("language", "stereotype")/ language=/ qt// genOptDefaultDatabase// qt/ / list="Attribute" @separator="\n" @indent=" "/ / if classStereotype=="Polygon"/ Stereotype = "Region" / endif/ / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype==""/ Stereotype = "Conventional" / endif/ } / list="Connector" @separator="\n"/</pre>
MDG_GeoProfile para MDG_PVL	
Template	Código de Transformação
<i>File</i>	<pre>Package { name="GeoProfile(PIM) para PVL (PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ }</pre>
<i>Class</i>	<pre>/ if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class")/ / TRANSFORM_CURRENT("language", "stereotype")/ language=/ qt// genOptDefaultDatabase// qt/ / list="Attribute" @separator="\n" @indent=" "/</pre>

	<pre> / if classStereotype=="Polygon"/ Stereotype = "Polygon" / endif/ / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype==""/ Stereotype = "" / endif/ / if classStereotype=="ComplexShape"/ Stereotype = "ComplexSpatialObj" / endif/ / if classStereotype=="MultiPoints"/ Stereotype = "MultPoints" / endif/ / if classStereotype=="MultiLines"/ Stereotype = "MultLines" / endif/ / if classStereotype=="MultiPolygons"/ Stereotype = "MultPolygons" / endif/ } / list="Connector" @separator="\n"/ </pre>
MDG_GeoProfile para MDG_OMTG	
Template	Código de Transformação
File	<pre> Package { name="GeoProfile(PIM) para OMTG (PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
Class	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class"/ / TRANSFORM_CURRENT("language", "stereotype"/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if classStereotype=="Polygon"/ Stereotype = "Polygon" / endif/ / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype==""/ Stereotype = "" / endif/ / if classStereotype=="Isolines"/ Stereotype = "Isolines" / endif/ / if classStereotype=="AdjPolygons"/ Stereotype = "PlanarSubdivision" / endif/ / if classStereotype=="IrregularPoints"/ Stereotype = "Sampling" / endif/ / if classStereotype=="GridOfCells"/ </pre>

	<pre> Stereotype = "Tessalation" / endif/ / if classStereotype=="TIN"/ Stereotype = "TIN" / endif/ } / list="Connector" @separator="\n"/ </pre>
MDG_GeoProfile para MDG_UML-GeoFrame	
Template	Código de Transformação
<i>File</i>	<pre> Package { name="GeoProfile(PIM) para UML-GeoFrame (PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
<i>Class</i>	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class"/ / TRANSFORM_CURRENT("language", "stereotype"/ language=/ qt// genOptDefaultDatabase// qt/ / list="Attribute" @separator="\n" @indent=" "/ / if classStereotype=="Polygon"/ Stereotype = "Polygon" / endif/ / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype=="ComplexSpatialObj"/ Stereotype = "ComplexSpatialObj" / endif/ / if classStereotype==""/ Stereotype = "Conventional" / endif/ / if classStereotype=="Isolines"/ Stereotype = "Isolines" / endif/ / if classStereotype=="AdjPolygons"/ Stereotype = "AdjPolygons" / endif/ / if classStereotype=="IrregularPoints"/ Stereotype = "IrregularPoints" / endif/ / if classStereotype=="GridOfCells"/ Stereotype = "GridOfCells" / endif/ / if classStereotype=="GridOfPoints"/ Stereotype = "GridOfPoints" / endif/ / if classStereotype=="TIN"/ Stereotype = "TIN" / endif/ } / list="Connector" @separator="\n"/ </pre>
MDG_GeoOOA para MDG_GeoProfile	
Template	Código de Transformação
<i>File</i>	<pre> Package { name="GeoOOA(PIM) para GeoProfile (PIM)" </pre>

	<pre> namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
Class	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class"/ / TRANSFORM_CURRENT("language", "stereotype"/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if classStereotype=="Region"/ Stereotype = "Polygon" / endif/ / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype=="Conventional"/ Stereotype = "" / endif/ } </pre>
MDG_PVL para MDG_GeoProfile	
Template	Código de Transformação
File	<pre> Package { name="PVL(PIM) para GeoProfile(PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
Class	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class"/ / TRANSFORM_CURRENT("language", "stereotype"/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if classStereotype=="Polygon"/ Stereotype = "Polygon" / endif/ e / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype==""/ Stereotype = "" / endif/ / if classStereotype=="ComplexSpatialObj"/ Stereotype = "ComplexShape" / endif/ / if classStereotype=="MultiPoints"/ Stereotype = "MultiPoints" / endif/ / if classStereotype=="MultiLines"/ Stereotype = "MultiLines" </pre>

	<pre> / endif/ / if classStereotype=="MultiPolygons"/ Stereotype = "MultiPolygons" / endif/ } / list="Connector" @separator="\n"/ </pre>
MDG_OMTG para MDG_GeoProfile	
Template	Código de Transformação
<i>File</i>	<pre> Package { name="OMTG (PIM) para GeoProfile (PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
<i>Class</i>	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Class { / TRANSFORM_REFERENCE("Class"/ / TRANSFORM_CURRENT("language", "stereotype"/ language=/ qt// genOptDefaultDatabase// qt/ / list="Attribute" @separator="\n" @indent=" "/ / if classStereotype=="Polygon"/ Stereotype = "Polygon" / endif/ / if classStereotype=="Line"/ Stereotype = "Line" / endif/ / if classStereotype=="Point"/ Stereotype = "Point" / endif/ / if classStereotype==""/ Stereotype = "" / endif/ / if classStereotype=="Isolines"/ Stereotype = "Isolines" / endif/ / if classStereotype=="PlanarSubdivision"/ Stereotype = "AdjPolygons" / endif/ / if classStereotype=="Sampling"/ Stereotype = "IrregularPoints" / endif/ / if classStereotype=="Tessalation"/ Stereotype = "GridOfCells" / endif/ / if classStereotype=="TIN"/ Stereotype = "TIN" / endif/ } / list="Connector" @separator="\n"/ </pre>
MDG_UML-GeoFrame para MDG_GeoProfile	
Template	Código de Transformação
<i>Files</i>	<pre> Package { name="UML-GeoFrame(PIM) para GeoProfile(PIM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
<i>Class</i>	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Class </pre>

```

{
/ TRANSFORM_REFERENCE("Class")/
/ TRANSFORM_CURRENT("language", "stereotype")/
language=/ qt/ / genOptDefaultDatabase/ / qt/
/ list="Attribute" @separator="\n" @indent=" "/

/ if classStereotype=="Polygon"/
Stereotype = "Polygon"
/ endif/
/ if classStereotype=="Line"/
Stereotype = "Line"
/ endif/
/ if classStereotype=="Point"/
Stereotype = "Point"
/ endif/
/ if classStereotype=="ComplexSpatialObj"/
Stereotype = "ComplexSpatialObj"
/ endif/
/ if classStereotype=="Conventional"/
Stereotype = ""
/ endif/
/ if classStereotype=="Isolines"/
Stereotype = "Isolines"
/ endif/
/ if classStereotype=="AdjPolygons"/
Stereotype = "AdjPolygons"
/ endif/
/ if classStereotype=="IrregularPoints"/
Stereotype = "IrregularPoints"
/ endif/
/ if classStereotype=="GridOfCells"/
Stereotype = "GridOfCells"
/ endif/
/ if classStereotype=="GridOfPoints"/
Stereotype = "GridOfPoints"
/ endif/
/ if classStereotype=="TIN"/
Stereotype = "TIN"
/ endif/
}
/ list="Connector" @separator="\n"/

```

APÊNDICE D- templates modificados para interoperabilidade vertical

Este apêndice apresenta os templates que foram modificados para que fosse possível a transformação Vertical dos esquemas conceituais de Banco de Dados Geográficos. A ferramenta EA por default contém códigos de transformações Data Definition Language (DDL) que possibilita o MDA para banco de dados convencionais. Os códigos DDL foram aproveitados para o MDA de banco de dados geográficos (Códigos como por exemplo: Criação de chaves primárias e estrangeiras, criação de relacionamentos e criação de tabelas). Ao replicar os testes, os templates que não foram apresentados neste apêndice podem ser deixados com o código default.

GeoOOA(PIM) para PostGIS (PSM)	
Template	Código de Transformação
File	<pre> Package { name="GeoOOA(PIM) para PostGIS(PSM)" namespace="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
Class	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Table { / TRANSFORM_REFERENCE("Table")/ / TRANSFORM_CURRENT("language", "stereotype")/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if elemType != "Association"/ PrimaryKey { Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ ID/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "NUMBER")/ / qt/ } } / endif/ / if classStereotype == "Point"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Point/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(POINT)")/ / qt/ } / endif/ / if classStereotype == "Region"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Region/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(POLYGON)")/ / qt/ } / endif/ / if classStereotype == "Line"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Line/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(LINESTRING)")/ / qt/ } / endif/ </pre>

	<pre> } / list="Connector" @separator="\n"/ </pre>
<i>attribute</i>	<pre> Column { / TRANSFORM_CURRENT("type", "stereotype", "collection", "constant", "containment", "ordered", "static", "volatile")/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,attType)/ / qt/ / if attType == "int" / type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"integer")/ / qt/ / endif/ / if attType == "char" / type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"varchar(50)")/ / qt/ / endif/ / if attType == "boolean" / type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"boolean")/ / qt/ / endif/ } </pre>
<i>Connector</i>	<pre> / if connectorType == "Generalization"/ ForeignKey { / TRANSFORM_REFERENCE("General",connectorGUID)/ Source { / TRANSFORM_REFERENCE("Table",connectorSourceElemGUID)/ name=/ qt/ / connectorName/ / qt/ multiplicity="0..1" Column { name=/ qt/ / CONVERT_NAME(connectorSourceElemName, "Pascal Case","Camel Case")/ ID/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"NUMBER")/ / qt/ } } Target { / TRANSFORM_REFERENCE("Table",connectorDestElemGUID)/ multiplicity="1" Column { name=/ qt/ / CONVERT_NAME(connectorDestElemName, "Pascal Case","Camel Case")/ ID/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"NUMBER")/ / qt/ } } } / endTemplate/ / if connectorType != "Association" and connectorType!="Aggregation"/ / endTemplate/ / if connectorDestElemType=="Association"/ \$dstMultiple = "T" \$srcMult = / connectorSourceMultiplicity/ / elseif connectorSourceElemType=="Association"/ \$sourceMultiple = "T" \$dstMult = / connectorDestMultiplicity/ / else/ \$srcMult = / connectorSourceMultiplicity/ \$dstMult = / connectorDestMultiplicity/ / if \$srcMult != "" and \$srcMult != "0" and \$srcMult != "0..1" and \$srcMult != "1"/ \$sourceMultiple = "T" / endif/ / if \$dstMult != "" and \$dstMult != "0" and \$dstMult != "0..1" and \$dstMult != "1"/ \$dstMultiple = "T" / endif/ / endif/ / if \$sourceMultiple == "T" and \$dstMultiple == "T"/ Table { </pre>

```

/ TRANSFORM_REFERENCE("LinkTable",connectorGUID)/
/ if connectorName != ""/
name=/ qt/ / connectorName/ / qt/
/ else/
name=/ qt/ Join/ connectorSourceElemName/ To/ connectorDestElemName/ / qt/
/ endif/
language=/ qt/ / genOptDefaultDatabase/ / qt/
}
ForeignKey
{
/ TRANSFORM_REFERENCE("FK1",connectorGUID)/
Source
{
/ TRANSFORM_REFERENCE("LinkTable",connectorGUID)/
name=/ qt/ / connectorDestElemName/ / qt/
/ if $srcMult != ""/
multiplicity=/ qt/ $srcMult/ qt/
/ endif/
Column
{
name=/ qt/ / CONVERT_NAME(connectorDestElemName, "Pascal Case","Camel Case")/ ID/ qt/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"NUMBER")/ / qt/
}
}
Target
{
/ TRANSFORM_REFERENCE("Table",connectorDestElemGUID)/
Column
{
name=/ qt/ / CONVERT_NAME(connectorDestElemName, "Pascal Case","Camel Case")/ ID/ qt/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"NUMBER")/ / qt/
}
}
}
ForeignKey
{
/ TRANSFORM_REFERENCE("FK2",connectorGUID)/
Source
{
/ TRANSFORM_REFERENCE("LinkTable",connectorGUID)/
name=/ qt/ / connectorSourceElemName/ / qt/
/ if $dstMult != ""/
multiplicity=/ qt/ $dstMult/ qt/
/ endif/
Column
{
name=/ qt/ / CONVERT_NAME(connectorSourceElemName, "Pascal Case","Camel Case")/ ID/ qt/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"NUMBER")/ / qt/
}
}
Target
{
/ TRANSFORM_REFERENCE("Table",connectorSourceElemGUID)/
Column
{
name=/ qt/ / CONVERT_NAME(connectorSourceElemName, "Pascal Case","Camel Case")/ ID/ qt/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"NUMBER")/ / qt/
}
}
}
/ elseif $sourceMultiple == "T"/
ForeignKey
{
/ TRANSFORM_REFERENCE("FK1",connectorGUID)/
Source
{
/ TRANSFORM_REFERENCE("Table",connectorSourceElemGUID)/
/ if connectorSourceElemType=="Association"/
name=/ qt/ / connectorDestElemName/ / qt/
/ elseif connectorName != ""/

```

```

name=/ qt/ / connectorName/ / qt/
/ elseif connectorDestRole != ""/
name=/ qt/ FK_/ connectorDestRole/ / qt/
/ endif/
/ if $SrcMult != ""/
multiplicity=/ qt/ $SrcMult/ qt/
/ endif/
Column
{
/ if connectorDestRole != ""/
name=/ qt/ / connectorDestRole/ / qt/
/ else/
name=/ qt/ / CONVERT_NAME(connectorDestElemName, "Pascal Case", "Camel Case")/ ID/ qt/
/ endif/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "NUMBER")/ / qt/
}
}
Target
{
/ TRANSFORM_REFERENCE("Table",connectorDestElemGUID)/
/ if $dstMult != ""/
multiplicity=/ qt/ $dstMult/ qt/
/ endif/
Column
{
name=/ qt/ / CONVERT_NAME(connectorDestElemName, "Pascal Case", "Camel Case")/ ID/ qt/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "NUMBER")/ / qt/
}
}
/ else/
ForeignKey
{
/ TRANSFORM_REFERENCE("FK1",connectorGUID)/
Source
{
/ TRANSFORM_REFERENCE("Table",connectorDestElemGUID)/
/ if connectorDestElemType=="Association"/
name=/ qt/ / connectorSourceElemName/ / qt/
/ elseif connectorName != ""/
name=/ qt/ / connectorName/ / qt/
/ elseif connectorSourceRole != ""/
name=/ qt/ FK_/ connectorSourceRole/ / qt/
/ endif/
/ if $dstMult != ""/
multiplicity=/ qt/ $dstMult/ qt/
/ endif/
Column
{
/ if connectorSourceRole != ""/
name=/ qt/ / connectorSourceRole/ / qt/
/ else/
name=/ qt/ / CONVERT_NAME(connectorSourceElemName, "Pascal Case", "Camel Case")/ ID/ qt/
/ endif/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "NUMBER")/ / qt/
}
}
Target
{
/ TRANSFORM_REFERENCE("Table",connectorSourceElemGUID)/
/ if $srcMult != ""/
multiplicity=/ qt/ $srcMult/ qt/
/ endif/
Column
{
name=/ qt/ / CONVERT_NAME(connectorSourceElemName, "Pascal Case", "Camel Case")/ ID/ qt/
type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "NUMBER")/ / qt/
}
}
}
/ endif/

```

PVL(PIM) para PostGIS (PSM)	
Template	Código de Transformação
File	<pre> Package { name="PVL(PIM) para PostGIS(PSM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
Class	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Table { / TRANSFORM_REFERENCE("Table"/ / TRANSFORM_CURRENT("language", "stereotype"/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if elemType != "Association"/ PrimaryKey { Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case"/ ID/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "Integer"/ / qt/ } } / endif/ / if classStereotype == "Point"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case"/ Point/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(POINT)"/ / qt/ } } / endif/ / if classStereotype == "Polygon"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case"/ Polygon/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(POLYGON)"/ / qt/ } } / endif/ / if classStereotype == "Line"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case"/ Line/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(LINESTRING)"/ / qt/ } } / endif/ / if classStereotype == "MultPoints"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case"/ MultPoints/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(multipoint)"/ / qt/ } } / endif/ / if classStereotype == "MultLines"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case"/ MultLines/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(multilinestring)"/ / qt/ } } / endif/ / if classStereotype == "MultPolygons"/ Column </pre>

	<pre> { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Multipolygons/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(multipolygon)")/ / qt/ } / endif/ / if classStereotype == "ComplexShape"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ ComplexShape/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(geometrycollection)")/ / qt/ } } / endif/ } / list="Connector" @separator="\n"/ </pre>
<i>attribute</i>	O mesmo código apresentado no template <i>attribute</i> da transformação GeoOOA(PIM) para PostGIS PSM.
<i>Connector</i>	O mesmo código apresentado no template <i>Connector</i> da transformação GeoOOA(PIM) para PostGIS PSM.
OMTG(PIM) para PostGIS (PSM)	
Template	Código de Transformação
<i>File</i>	<pre> Package { name="OMTG(PIM) para PostGIS(PSM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
<i>Class</i>	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Table { / TRANSFORM_REFERENCE("Table")/ / TRANSFORM_CURRENT("language", "stereotype")/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if elemType != "Association"/ PrimaryKey { Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ ID/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"Integer")/ / qt/ } } } / endif/ / if classStereotype == "Point"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Point/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(POINT)")/ / qt/ } } / endif/ / if classStereotype == "Polygon"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Polygon/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(POLYGON)")/ / qt/ } } / endif/ / if classStereotype == "Line"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Line/ qt/ </pre>

	<pre> type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(LINESTRING)")/ qt/ } / endif/ / if classStereotype == "Sampling"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Sampling/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(multipoint)")/ qt/ } / endif/ / if classStereotype == "TIN"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ TIN/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(geometrycollection)")/ qt/ } / endif/ / if classStereotype == "Isolines"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Isolines/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(multilinestring)")/ qt/ } / endif/ / if classStereotype == "PlanarSubdivision"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ PlanarSubdivision/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"GEOMETRY(multipolygon)")/ qt/ } / endif/ / if classStereotype == "Tessalation"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case","Camel Case")/ Tessalation/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase,"RASTER")/ qt/ } / endif/ } / list="Connector" @separator="\n"/ </pre>
attribute	O mesmo código apresentado no template attribute da transformação GeoOOA(PIM) para PostGIS PSM.
Connector	O mesmo código apresentado no template Connector da transformação GeoOOA(PIM) para PostGIS PSM.
UML-GeoFrame(PIM) para PostGIS (PSM)	
Template	Código de Transformação
File	<pre> Package { name="UML-GeoFrame(PIM) para PostGIS(PSM)" namespaceroot="true" / list="Namespace" @separator="\n\n" @indent=" "/ } </pre>
Class	<pre> / if classStereotype=="enumeration"/ / endTemplate/ Table { / TRANSFORM_REFERENCE("Table")/ / TRANSFORM_CURRENT("language", "stereotype")/ language=/ qt/ / genOptDefaultDatabase/ / qt/ / list="Attribute" @separator="\n" @indent=" "/ / if elemType != "Association"/ PrimaryKey </pre>

```

{
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ ID/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "Integer")/ / qt/
  }
}
/ endif/

/ if classStereotype == "Point"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Point/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(POINT)")/ / qt/
  }
/ endif/

/ if classStereotype == "Polygon"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Polygon/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(POLYGON)")/ / qt/
  }

/ endif/

/ if classStereotype == "Line"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Line/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(LINESTRING)")/ / qt/
  }

/ endif/

/ if classStereotype == "ComplexSpatialObj"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ ComplexSpatialObj/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(geometrycollection)")/ / qt/
  }

/ endif/

/ if classStereotype == "IrregularPoints"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ IrregularPoints/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(multipoint)")/ / qt/
  }

/ endif/

/ if classStereotype == "TIN"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ TIN/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(geometrycollection)")/ / qt/
  }

/ endif/

/ if classStereotype == "Isolines"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ Isolines/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(multilinestring)")/ / qt/
  }

/ endif/

/ if classStereotype == "AdjPolygons"/
  Column
  {
    name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ AdjPolygons/ qt/
    type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "GEOMETRY(multipolygon)")/ / qt/
  }

```

	<pre> } / endif/ / if classStereotype == "GridOfCells"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ GridOfCells/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "RASTER")/ qt/ } / endif/ / if classStereotype == "GridOfPoints"/ Column { name=/ qt/ / CONVERT_NAME(className, "Pascal Case", "Camel Case")/ GridOfPoints/ qt/ type=/ qt/ / CONVERT_TYPE(genOptDefaultDatabase, "RASTER")/ qt/ } / endif/ } / list="Connector" @separator="\n"/ </pre>
<i>Attribute</i>	O mesmo código apresentado no template attribute da transformação GeoOOA(PIM) para PostGIS PSM.
<i>Connector</i>	O mesmo código apresentado no template Connector da transformação GeoOOA(PIM) para PostGIS PSM.