UNIVERSIDADE FEDERAL DA PARAÍBA

Centro de Ciências Exatas e da Natureza
Departamento de Informática

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

GEOMININGVISUALQL: UMA LINGUAGEM DE CONSULTA VISUAL PARA MINERAÇÃO DE DADOS GEOGRÁFICOS

Klebber de Araújo Pedrosa

João Pessoa – PB

Klebber de Araújo Pedrosa

GEOMININGVISUALQL: UMA LINGUAGEM DE CONSULTA VISUAL PARA MINERAÇÃO DE DADOS GEOGRÁFICOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática do Departamento de Informática da Universidade Federal da Paraíba como requisito parcial para obtenção do título de Mestre em Informática.

Orientadora: Dra. Valéria Gonçalves Soares.

João Pessoa – Paraíba
Agosto / 2010

P372g Pedrosa, Klebber de Araújo.
GeoMiningVisualQL: uma linguagem de consulta visual para mineração de dados geográficos / Klebber de Araújo Pedrosa. - - João Pessoa: [s.n.], 2010.
154 f.
Orientadora: Valéria Gonçalves Soares.
Dissertação (Mestrado) – UFPB/CCEN.

1.Informática. 2.Banco de dados geográficos. 3.Linguagem de consulta visual. 4. Mineração de dados.

UFPB/BC CDU: 004(043)

Ata da Sessão Pública de Defesa de Dissertação de Mestrado do KLEBBER DE ARAUJO PEDROSA, candidato ao Título de Mestre em Informática na Área de Sistemas de Computação, realizada em 10 de agosto de 2010.

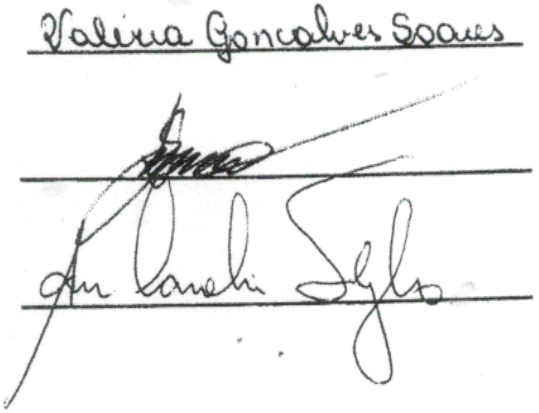
Aos dez dias do mês de agosto do ano dois mil e dez, às dez horas, na Sala de Reunião do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para examinar o candidato ao grau de Mestre em Informática, na área de "*Sistemas de Computação*", na linha de pesquisa "*Computação Distribuída*", o Sr. Klebber de Araujo Pedrosa. A comissão examinadora composta pelos professores doutores: Valéria Gonçalves Soares (DI - UFPB), Orientador e Presidente da Banca Examinadora, Lucídio dos Anjos Formiga Cabral, (DI-UFPB), como examinador interno e Ana Carolina Salgado (Cin/UFPE), como examinador externo. Dando início aos trabalhos, a Profª. Valéria Gonçalves Soares, cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse, oralmente, a exposição do trabalho de dissertação intitulado "*GeoMiningVisualQL:Uma Linguagem de Consulta Visual para mineração de Dados Geográficos*". Concluída a exposição, o candidato foi argüido pela Banca Examinadora que emitiu o seguinte parecer: "aprovado" Assim sendo, deve a Universidade Federal da Paraíba expedir o respectivo diploma de Mestre em Informática na forma da lei e, para constar, a professora Tatiana Aires Tavares, Sra. Coordenadora do PPGI, lavrou a presente ata, que vai assinada por ela, e pelos membros da Banca Examinadora. João Pessoa, 10 de agosto de 2010.

Tatiana Aires Tavares

Profª. Dra. Valéria Gonçalves Soares
Orientadora (DI-UFPB)

Prof. Dr. Lucídio dos Anjos Formiga Cabral
Examinador Interno (DI-UFPB)

Profª. Drª. Ana Carolina Salgado
Examinadora Externa (Cin/UFPE)



DEDICATÓRIA

Dedico este trabalho, antes de tudo, a Deus e a Nossa Senhora pelo dom da vida e aos meus familiares, especialmente ao meu pai Manoel, a minha mãe Alice, a minha noiva Lucinha, aos meus irmãos Clayton e Aline, ao meu tio e padrinho Alencar (in memorian) e a minha prima Glauciene.

AGRADECIMENTOS

A Deus, nosso Senhor e Pai eterno, pelas bênçãos e oportunidades concedidas, além das forças para que pudesse conquistar mais uma vitória em minha vida.

A Virgem Maria Santíssima, mãe eterna que roga por todos nós.

Aos meus pais, Manoel Pedrosa e Alice Araújo, por me apoiarem em todos os momentos da minha vida, muitas vezes diante de dificuldades e sacrifícios. Agradeço por acreditarem em mim e sempre estarem ao meu lado, dando-me forças e estímulos aos estudos.

Ao meu tio José de Alencar (in memorian) e a minha prima Glauciene, por terem me apoiado e acreditado na minha capacidade. Essa conquista também é de vocês.

A minha noiva Lúcia Chaves, pelo amor, carinho, apoio, incentivo e compreensão em todos os momentos, os quais foram fundamentais para o cumprimento desta etapa em minha vida.

Aos meus irmãos Clayton e Aline, pelo apoio, companheirismo, confiança, convivência e aprendizagem ao longo de nossas vidas.

A minha tia Alda e a minha cunhada Karla, pelas palavras de carinho, apoio e incentivo.

Aos meus sobrinhos Malu e Miguel, pelos sorrisos ingênuos através dos quais entendemos o verdadeiro sentido da vida.

A todos os meus familiares, especialmente ao **tio Severino** e todos aqueles, cujos nomes não descrevem neste, mas que contribuíram com o meu crescimento intelectual, profissional e cultural.

A minha orientadora, professora Valéria Gonçalves Soares, pela paciência, apoio, orientação, ensinamentos e, sobretudo, acreditar na minha capacidade durante todo este período da minha pós-graduação.

Aos meus amigos Thiego Marinho, Alan Bonifácio, Bruno Fernando, Saulo Oliveira, Kleyson Santos e Lamarck Ribeiro. Thiego, obrigado por essa forte amizade, pela paciência, aprendizado, incentivos e motivações. Alan, agradeço a sua paciência, amizade e força dada quase que todos os dias durante este tempo. Bruno, a você meu mestre e irmão, que Deus conserve essa nossa amizade de tantos anos. Saulo, obrigado pelo apoio, amizade e compreensão. Kleyson, agradeço ao seu incentivo, apoio e força dada. Lamarck, valeu pelos conselhos e companheirismo durante todo este período da nossa pós-graduação.

Aos meus amigos e colegas da UFPB, especialmente do Centro de Tecnologia, Jessé Miranda e Jadílson Paiva, pela amizade, paciência e apoio concedido.

Ao professor Hamilton Soares, pela amizade, apoio e compartilhamento da sua sabedoria.

A todos aqueles que contribuíram de alguma forma para a realização deste trabalho de mestrado.

Muito Obrigado!!!

RESUMO

Diversas áreas de domínio de conhecimento, tais como os sistemas de sensoriamente remoto, transportes, telecomunicações, cartografia digital, entre outras, fazem uso de uma grande quantidade de dados geográficos. Normalmente, esses dados são armazenados em Sistemas Gerenciadores de Banco de Dados Geográficos (SGBDGeo), através dos quais, muitas vezes, podem ser manipulados por Sistemas de Informações Geográficas (SIG). Entretanto, esses sistemas não são capazes de extrair novas informações, previamente desconhecidas pelos usuários, as quais podem estar embutidas dentro da base de dados do domínio analisado e que, de certo modo, representam algum conhecimento novo e de grande utilidade, por exemplo, para tomadas de decisões. Neste caso, é necessário fazer uso de técnicas específicas de Descoberta de Conhecimento em Banco de Dados (DCBD ou KDD, Knowledge Discovery in Database). Além disso, os dados geográficos apresentam características inerentemente visuais que, muitas vezes, podem ser associados a representações visuais geométricas ou pictográficas. Nesse contexto, existem algumas linguagens de consultas visuais para dados geográficos. Todavia, poucas delas tratam métodos de mineração espacial entre os dados. Desta forma, este trabalho propõe a construção de um ambiente para as tarefas de mineração de dados realizada sob certos domínios geográficos, além da especificação formal de uma linguagem de consulta visual a ser usada neste ambiente. Estas consultas são formuladas através de representações pictóricas de feições geográficas, operadores e relacionamentos espaciais existentes entre estes dados. Para tal, utilizam-se abstrações metafóricas sobre os metadados do ambiente geográfico, além da abordagem definida como "fluxo corrente" na qual o usuário foca a sua atenção em determinadas etapas do processo de mineração, facilitando a construção destas consultas por parte dos mesmos. Desta forma, o ambiente proposto tem como objetivo simplificar as consultas sobre tarefas de mineração de dados geográficos, tornando-as mais amigáveis aos usuários, concedendo mais eficiência e rapidez quando se comparado aos scripts textuais de consultas.

Palavras-Chave: Bancos de Dados Geográficos, Descoberta de Conhecimento, Linguagem de Consulta Visual.

ABSTRACT

Several areas of knowledge domain, such as remote sensing systems, transportation, telecommunication, digital mapping, among others, make use of large amounts of geographic data. Typically, these data are stored in Management Systems Geographic Database (SGBDGeo), through which can be often manipulated by Geographic Information Systems (GIS). However, these systems are not able to extract new information, previously unknown to users, which may be embedded within the database field analysed and that, somehow, represent new and userful knowledge, for example, for decision making. In this case, it is necessary to make use of specific techniques of Knowledge Discovery in Databases (KDD). Moreover, spatial data present inherently visual characteristics that, often, can be associated with geometric and pictographic visual representations. In this context, there are few visual query languages for spatial data. However, few of this treat mining methods among the spatial data. Thus, this paper proposes the construction of an environment for data mining tasks performed under certain geographical areas, beyond the formal specification of a visual query language to be used in this environment. These queries are formulated through pictorial representations of geographic features, operators, and spatial relationships between these data. To this end, we use metaphorical abstractions on the metadata of the geographical environment, and the approach defined as "flowing stream" in which the user focuses attention on certain stages of the mining process, facilitating the construction of these consultations a number of them. Thus, the proposed environment aims to simplify the tasks of consultations on mining spatial data, making them more user friendly, providing more efficiency and speed when compared to textual queries scripts.

Keywords: Geographic Databases, Knowledge Discovery, Visual Query Language.

SUMÁRIO

Lista de Figuras ..... **xii**

Lista de Tabelas ..... **xv**

Lista de Abreviações e Siglas ..... **xvii**

Capítulo 1: Introdução **18**

1.1 Motivação 18
1.2 Escopo do Trabalho 20
1.3 Objetivos 21
1.3.1 Objetivo Geral 21
1.3.2 Objetivos Específicos 21
1.4 Estrutura do Trabalho 22

Capítulo 2: Fundamentação Teórica **23**

2.1 Bancos de Dados Geográficos 23
2.1.1 Armazenamento de dados espaciais 27
2.1.2 Dependências Geográficas 29
2.1.3 Tipos de relacionamentos espaciais 30
2.2 Descoberta de Conhecimento em Bancos de Dados 33
2.2.1 Etapas do processo de descoberta de conhecimento em banco de dados 35
2.2.2 Mineração de Dados 36
2.2.3 Regra de Associação 37
2.3 Primitivas para a construção de uma Linguagem de Mineração de Dados 38
2.3.1 Dados relevantes na tarefa de mineração 39
2.3.2 Tipos de conhecimento a ser minerado: 40
2.3.3 Domínio do contexto a ser minerado 40
2.3.3.1 Sintaxe de definição de um esquema hierárquico 40
2.3.3.2 Sintaxe de definição de um conjunto hierárquico 41
2.3.4 Medidas interessantes 42
2.3.5 Apresentação (visualização) dos padrões descobertos 42
2.3.6 Exemplo de uma consulta geral feita através da linguagem DMQL 42
2.4 Linguagem GMQL 43
2.4.1 Visão Geral 43
2.4.2 Especificação dos dados relevantes 45
2.4.3 Tipos de descoberta de conhecimento 46
2.4.3.1 Characteristic Rules 46
2.4.3.2 Comparison Rules 47
2.4.3.3 Clustering Rules 48
2.4.3.4 Spatial Association Rules 48
2.4.3.5 Classification Rules 49
2.4.4 Definições de limiares 49
2.4.5 Conceitos de hieraquização 50
2.5 Ambientes de Consultas Visuais 50

2.5.1 Arquitetura de um Sistema de Consulta Visual 51
2.5.2 Importância de um SCV e seus principais aspectos 51
2.5.3 Classificações dos Sistemas de Consultas Visuais 52
2.6 Sistemas de Consultas Visuais para a Mineração de Dados Geográficos 53
2.7 Considerações Finais do Capítulo 53

Capítulo 3: Trabalhos Relacionados 55

3.1 Introdução 55
3.2 GeoMiner 56
3.2.1 Arquitetura 56
3.3 XQBE 58
3.4 GeoVisualQL 59
3.5 SVQL 60
3.6 VisMiner 61
3.7 Considerações Finais do Capítulo 63

Capítulo 4: GeoMiningVisualQL – Linguagem de Consulta para Mineração de Dados Geográficos 64

4.1 Introdução 64
4.2 Especificação da Linguagem 65
4.3 Definição Formal da Linguagem 66
4.4 Definiçã das Representações Pictóricas 68
4.4.1 Representação Geral 68
4.4.2 Ações a serem exercidas 68
4.4.3 Manipulação Hierárquica 69
4.4.4 Tipos de tarefas de mineração 69
4.4.5 Definição de termos espaciais ou não-espaciais no ambiente 71
4.4.6 Definição geral de uma feição geográfica 71
4.4.7 Definição geral de um atributo de uma feição geográfica 72
4.4.8 Definição genérica de funções não-espaciais 72
4.4.9 Definição de funções simples específicas 73
4.4.10 Definição genérica de funções espaciais 74
4.4.11 Definição de funções espaciais específicas 74
4.4.12 Definição de predicados espaciais 76
4.4.13 Definição Geral de um operador geográfico 77
4.4.14 Definição de operadores geográficos 78
4.4.15 Definição de operadores relacionais 80
4.4.16 Definição de operadores lógicos (booleanos) 81
4.5 Formalismo Gramatical da Linguagem 82
4.5.1 Etapas de construção da gramática da linguagem GeoMiningVisualQL 82
4.5.1.1 Definição da tarefa de mineração a ser realizada sobre determinados dados geográficos 82
4.5.1.2 Seleção de termos espaciais e não-espaciais a serem investigados na tarefa 84
4.5.1.3 Restrição condicional da consulta 96
4.5.1.4 Agrupamento de atributos envolvidos na consulta 103
4.5.1.5 Condições do reagrupamento da etapa IV 105
4.5.1.6 Definição de limiares na consulta 107

4.5.2 Gramática da Linguagem GeoMiningVisualQL 110
4.6 Considerações Finais do Capítulo 111

Capítulo 5: GeoMiningVisual – Ambiente de Consulta Visual sobre tarefas de Mineração de Dados Geográficos baseado em metáforas visuais 112

5.1 Introdução 112
5.2 Abordagem utilizada na formulação de uma consulta 114
5.3 Metáfora Visual do Ambiente 114
5.4 Metadados do Ambiente 115
5.5 Arquitetura do Ambiente 116
5.5.1 Camada de Interface Gráfica 118
5.5.2 Camada de Formulação da Consulta 119
5.5.3 Camada de Descrição 120
5.6 Implementação do Sistema GeoMiningVisual 120
5.6.1 Ambiente de Desenvolvimento 121
5.6.2 Funcionalidade do Sistema 122
5.6.3 Visão Geral da Formulação de uma Consulta no GeoMiningVisual 122
5.6.4 Configuração de uma consulta no GeoMiningVisual 123
5.6.4.1 Configuração dos dados relevantes na tarefa de mineração 123
5.6.4.2 Configuração do tipo de conhecimento a ser minerado 124
5.6.4.3 Configuração do conhecimento sobre o contexto do domínio a ser minerado 125
5.6.5 Sugestão de uma nova interface para o ambiente GeoMiningVisual 127
5.6.5.1 Consulta GMQL 128
5.6.5.2 Execução da consulta no ambiente GeoMiningVisual 129
5.7 Funcionamento do Analisador Sintático 133
5.8 Análise Comparativa 133
5.9 Considerações Finais do Capítulo 136

Capítulo 6: Conclusão e Trabalhos Futuros 138

6.1 Resumo 138
6.2 Dificuldades Encontradas 138
6.3 Contribuições deste Trabalho 139
6.4 Trabalhos Futuros 140

Referências Bibliográficas 141

Apêndice A 146

Apêndice B 151

LISTA DE FIGURAS

Figura 2.1: Exemplo de uma imagem de satélite. 23
Figura 2.2: Exemplo de um modelo numérico de terreno através de isolinhas[1]. 24
Figura 2.3: Camadas de Referências Geográficos. 25
Figura 2.4: Domínio Geográfico da cidade de João Pessoa. 25
Figura 2.5: Exemplos de representações matriciais. 28
Figura 2.6: Exemplo de tesselações. 28
Figura 2.7: Sobreposição de representações matricial e vetorial a partir de uma imagem real. 29
Figura 2.8: Exemplo de relacionamentos topológicos entre entidades espaciais. 32
Figura 2.9: Exemplo de relacionamentos direcionais entre entidades espaciais. 32
Figura 2.10: Exemplo de relacionamentos métricos entre entidades espaciais. 32
Figura 2.11: Relações de existência de um objeto baseado no tempo. 33
Figura 2.12: Etapas do processo de KDD. 35
Figura 2.13: Esquema hierárquico para a data de nascimento de um indivíduo. 41
Figura 2.14: Conceito hierárquico sobre o atributo "idade". 41
Figura 2.15: Exemplo de consulta feita em DMQL. 43
Figura 2.16: Sintaxe de uma consulta geral feitam em GMQL. 44
Figura 3.1: Arquitetura Geral do GeoMiner [Han, Koperski e Stefanvic 1997]. 56
Figura 3.2: Interface do GeoMiner. 57
Figura 3.3: Visualização da interface mostrando as janelas de ontologia e de consulta [Kade 2001]. 58
Figura 3.4: Janela Principal do GeoVisual. 59
Figura 3.5: Exemplo de uma consulta no GeoVisual pelos municípios que sobrepõem a Bacia de Curimataú. 60
Figura 3.6: Resultado da consulta feita na figura 3.5. 60
Figura 3.7: Exemplo de uma consulta visual feita no SVQL Composer. 61
Figura 3.8: Exemplo de uma representação metafórica "Rail/Road" [Bimonte et al. 2003]. 62
Figura 4.1: Primeira etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 83
Figura 4.2: Seleção de uma tarefa de regra de associação espacial. 83
Figura 4.3: Segunda etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 84
Figura 4.4: Decomposição do símbolo pictográfico de restrições espaciais. 85
Figura 4.5: Decomposição do símbolo pictográfico de funções simples. 85
Figura 4.6: Decomposição do símbolo pictográfico de operadores aritméticos. 86
Figura 4.7: Consulta visual e tradução da seleção de termos não-espaciais simples. 87
Figura 4.8: Exemplo de configuração da seleção de um atributo simples de uma entidade cidade. 87
Figura 4.9: Consulta visual e tradução da seleção de termos não-espaciais simples através da junção de atributos simples por um operador aritmético. 88
Figura 4.10: Exemplo de configuração da seleção de um termo não-espacial, baseando-se na representação da figura 4.9. 89

Figura 4.11: Consulta visual aplicando uma função sobre termos não-espaciais do ambiente geográfico. 90
Figura 4.12: Exemplo de configuração de uma consulta visual aplicando uma função para somar a população masculina das cidades do estado da Paraíba. 91
Figura 4.13: Consulta visual aplicando uma função espacial sobre feições geográficos de um ambiente. 92
Figura 4.14: Exemplo de configuração de uma consulta visual fazendo uso de funções espaciais. 92
Figura 4.15: Seleção de termos espaciais através de relacionamentos existentes entre feições geográficas do ambiente. 93
Figura 4.16: Configuração de um predicado espacial métrico. 94
Figura 4.17: Configuração de um predicado espacial direcional. 94
Figura 4.18: Configuração de um predicado espacial topológico. 95
Figura 4.19: Exemplo de uma consulta utilizando um predicado espacial métrico. 95
Figura 4.20: Terceira etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 96
Figura 4.21: Decomposição do símbolo pictográfico de operador geográfico. 97
Figura 4.22: Decomposição do símbolo pictográfico de operador lógico. 97
Figura 4.23: Decomposição do símbolo pictográfico de operador relacional. 97
Figura 4.24: Configuração visual da consulta na seleção condicional de um operador lógico aplicado sobre um atributo simples. 98
Figura 4.25: Configuração visual da consulta baseada na seleção condicional utilizando operadores lógicos e expressões relacionais. 98
Figura 4.26: Configuração visual da consulta baseada na seleção condicional utilizando operadores lógicos e um valor literal entre expressões relacionais. 99
Figura 4.27: Exemplo de parte de uma consulta fazendo restrição do registros do resultado através de expressões relacionais. 99
Figura 4.28: Exemplo de parte de uma consulta fazendo restrição do registros do resultado através de operadores lógicos e de expressões relacionais. 100
Figura 4.29: Restrição condicional baseada em operadores geográficos. 101
Figura 4.30: Restrição condicional baseada em predicados espaciais métricos. 101
Figura 4.31: Exemplo da utilização de um operador geográfico entre entidades geográficas do ambiente. 102
Figura 4.32: Exemplo de restrição condicional fazendo uso de um predicado espacial métrico. 103
Figura 4.33: Quarta etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 103
Figura 4.34: Exemplo de agrupamento de um termo não-espaciais na consulta. 104
Figura 4.35: Quinta etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 105
Figura 4.36: Tradução da seleção condicional sobre atributos simples na segunda e na quinta etapa de configuração de parte de uma consulta visual na GeoMiningVisualQL. . 106
Figura 4.37: Exemplo de consulta feita em GeoMiningVisualQL fazendo uso de restrição condicional sobre os atributos envolvidos no agrupamento feito na quarta etapa de configuração. 107
Figura 4.38: Sexta e última etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 108
Figura 4.39: Tradução dos parâmetros de suporte e confiança utilizados na sexta e última etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL. 108

Figura 4.40: Exemplo de uma consulta em GeoMiningVisualQL fazendo uso do parâmetro de suporte no qual controla o número de padrões retornados num processo de descoberta de conhecimento. 109
Figura 4.41: Formalismo Gramatical de parte da Linguagem GeoMiningVisualQL..... 110
Figura 5.1: Tela Inicial do Ambiente GeoMiningVisual. 113
Figura 5.2: Demonstração da metáfora e metadados do ambiente GeoMiningVisual. 116
Figura 5.3: Demonstração da arquitetura de um sistema típico de mineração de dados. .. 117
Figura 5.4: Arquitetura do sistema GeoMiningVisual. 118
Figura 5.5: Detalhamento do Gerenciador de Consultas. 120
Figura 5.6: Momento em que o usuário configura uma tarefa de regra de associação no ambiente. 124
Figura 5.7: Script gerado junto a definição de um valor literal para a tarefa de regra de associação. 125
Figura 5.8: Script padrão gerado junto a escolha de uma tarefa de regra de associação... 125
Figura 5.9: Etapa de seleção de atributos gerais do ambiente. 126
Figura 5.10: Proposta de nova interface para o ambiente GeoMiningVisual. 128
Figura 5.11: Exemplo de uma consulta em GMQL para uma tarefa de regra de associação espacial. 128
Figura 5.12: Definição de uma consulta a ser utilizada no ambiente. 130
Figura 5.13: Configuração da primeira etapa: seleção da tarefa de mineração a ser realizada no ambiente. 130
Figura 5.14: Configuração da segunda etapa: seleção de termos espaciais e não-espaciais a serem investigados na consulta. 131
Figura 5.15: Configuração da terceira etapa: aplicação de restrição condicionais sobre os resultado. 132
Figura 5.16: Configuração da quarta etapa: agrupamento de registros retornados. 132

LISTA DE TABELAS

Tabela 2.1: Entidade geográfica Avenida. 26
Tabela 2.2: Entidade geográfica Rio. 26
Tabela 2.3: Entidade geográfica Porto. 26
Tabela 2.4: Entidade geográfica Viaduto. 26
Tabela 4.1: Representação pictórica geral. 68
Tabela 4.2: Representação de ações a serem exercidas no ambiente. 68
Tabela 4.3: Representação do uso de níveis hierárquicos. 69
Tabela 4.4: Representação de tarefas de regras de associação entre os dados geográficos. 69
Tabela 4.5: Representação de tarefas de clusterização entre os dados geográficos. 69
Tabela 4.6: Representação de tarefas de classificação entre os dados geográficos. 70
Tabela 4.7: Representação de tarefas de caracterização entre os dados geográficos. 70
Tabela 4.8: Representação de tarefas de comparação entre os dados geográficos. 70
Tabela 4.9: Representação do uso de um termo não-espacial no ambiente. 71
Tabela 4.10: Representação do uso de um termo espacial no ambiente. 71
Tabela 4.11: Representação do uso de uma feição geográfica qualquer. 71
Tabela 4.12: Representação geral de um atributo simples da feição geográfica do uso de uma feição geográfica. 72
Tabela 4.13: Representação geral de um atributo simples da feição geográfica do uso de uma feição geográfica. 72
Tabela 4.14: Representação do uso de funções simples (não-espaciais) no ambiente. 72
Tabela 4.15: Representação do uso de uma função que calcula a média dos valores de um atributo específico. 73
Tabela 4.16: Representação do uso de uma função que calcula a soma dos valores de um atributo específico. 73
Tabela 4.17: Representação do uso de uma função que retorna o valor mínimo de um atributo específico. 73
Tabela 4.18: Representação do uso de uma função que retorna o valor máximo de um atributo específico. 73
Tabela 4.19: Representação do uso de uma função conta os registros retornados por uma consulta. 74
Tabela 4.20: Representação do uso de funções espaciais no ambiente. 74
Tabela 4.21: Representação do uso de uma função espacial delimitada externamente por uma fronteira simples ou contínua. 74
Tabela 4.22: Representação do uso de uma função espacial delimitada externamente por pontos contidos em uma fronteira simples ou contínua. 74
Tabela 4.23: Representação do uso de uma função espacial delimitada externamente por segmentos de uma fronteira simples ou contínua. 75
Tabela 4.24: Representação do uso de uma função espacial delimitada externamente por polígonos formados em torno de uma fronteira simples ou contínua. 75
Tabela 4.25: Representação do uso de uma função espacial delimitada internamente por uma fronteira simples ou contínua. 75
Tabela 4.26: Representação do uso de uma função espacial delimitada internamente por pontos contidos em uma fronteira simples ou contínua. 75

Tabela 4.27: Representação do uso de uma função espacial delimitada internamente por segmentos de uma fronteira simples ou contínua. 76
Tabela 4.28: Representação do uso de uma função espacial delimitada internamente por polígonos formados em torno de uma fronteira simples ou contínua. 76
Tabela 4.29: Representação de um predicado espacial métrico utilizado entre duas entidades geográficas no ambiente. 76
Tabela 4.30: Representação de um predicado espacial direcional utilizado entre duas entidades geográficas no ambiente. 77
Tabela 4.31: Representação de um predicado espacial topológico utilizado entre duas entidades geográficas no ambiente. 77
Tabela 4.32: Representação geral de um operador geográfico. 77
Tabela 4.33: Representação do operador geográfico CONTAINS. 78
Tabela 4.34: Representação do operador geográfico WITHIN. 78
Tabela 4.35: Representação do operador geográfico INTERSECT. 78
Tabela 4.36: Representação do operador geográfico NEIGHBOR. 78
Tabela 4.37: Representação do operador geográfico EQUAL. 79
Tabela 4.38: Representação do operador geográfico UNDER. 79
Tabela 4.39: Representação do operador geográfico OVER. 79
Tabela 4.40: Representação do operador geográfico WEST_FROM. 79
Tabela 4.41: Representação do operador geográfico EAST 80
Tabela 4.42: Representação do operador geográfico SOUTH_FROM. 80
Tabela 4.43: Representação do operador geográfico NORTH_FROM. 80
Tabela 4.44: Representação geral dos operadores relacionais. 80
Tabela 4.45: Tabela de operadores relacionais específicos. 81
Tabela 4.46: Representação geral dos operadores lógicos. 81
Tabela 4.47: Tabela de operadores lógicos específicos. 81
Tabela 5.1: Representação geral da feição geográfica correspondente a uma cidade. 129
Tabela 5.2: Representação geral da feição geográfica correspondente a um rio. 129
Tabela 5.3: Representação geral da feição geográfica correspondente a um lago. 129
Tabela 5.4: Principais características do Ambiente de Consulta GeoMiningVisual. 134
Tabela 5.5: Análise Comparativa do GeoMiningVisual em relação aos trabalhos vistos neste capítulo. 135

LISTA DE ABREVIAÇÕES E SIGLAS

BDGs	*Bancos de Dados Geográficos*
CDL	*Change Description Language*
DCBDGeo	*Descoberta de Conhecimento em Bancos de Dados Geográficos*
DMQL	*Data Mining Query Language*
DW	*Data Warehouse*
GeoMiningVisualQL	*Geographic Data Mining Visual Query Language*
GeoVisualQL	*Geographic Visual Query Language*
GMQL	*Geo Mining Query Language*
GUI	*Graphic User Interface*
KDD	*Knowledge Discovery in Databases*
LCV	*Linguagem de Consulta Visual*
OLAP	*On-line Analytical Processing*
SCV	*Sistemas de Consultas Visuais*
SDMOQL	*Spatial Data Mining Object Query Language*
SGBDGeo	*Sistemas Gerenciadores de Banco de Dados Geográficos*
SIGs	*Sistemas de Informações Geográficas*
SOLAP	*Spatial OLAP*
SQL	*Structured Query Language*
SVQL	*SOLAP Visual Query Language*
VQS	*Visual Query Systems*
XQBE	*XML Query By Example*

CAPÍTULO 1

INTRODUÇÃO

Esta dissertação apresenta uma Linguagem de Consulta Visual sobre Processos de Mineração de Dados Geográficos, além do projeto de construção de um ambiente sob o qual ela pode ser inserida. Neste capítulo, será apresentado o contexto em que este trabalho se insere, através de sua motivação, escopo e principais objetivos. Por fim, será apresentada a organização e estrutura do trabalho realizado neste estudo.

1.1 MOTIVAÇÃO

A mineração de dados (*Data Mining*) é uma área promissora que se encontra em amplo crescimento, uma vez que, através dela, podem-se extrair conhecimentos novos e úteis, que estão previamente desconhecidos para a maioria dos usuários perante a grande quantidade de dados presente em diversos tipos de Sistemas de Banco de Dados.

Conforme [Han et al. 1996], devido a sua importância no cenário atual, acredita-se que o sucesso dos sistemas de mineração de dados se assemelha ao sucesso obtido pelos sistemas relacionais que dominaram os sistemas de bancos de dados durante várias décadas. Apesar de existirem diferentes interfaces gráficas para usuários sobre os diversos sistemas comerciais de bancos de dados relacionais, a essência básica entre eles está na padronização de uma linguagem de consulta relacional.

Nesse sentido, o sucesso dos sistemas de banco de dados relacionais deve ser creditado à padronização da linguagem SQL ser feita em um estágio muito cedo de desenvolvimento. Assim, é de fundamental importância entender todas as primitivas que constituem uma linguagem de mineração de dados. Além disso, a mineração de dados cobre uma vasta área de tarefas, tais como: regras de associação, clusterização, classificação, entre outras. Portanto, existem diversas e diferentes interfaces gráficas entre essas tarefas e é de grande importância entender os mecanismos associados a cada método de mineração de dados ao se construir uma linguagem deste tipo.

Diversos algoritmos para mineração de dados têm sido propostos na literatura [Wu et al. 2008]. De forma geral, segundo [Elmasri e Navathe 2005], o principal problema

ao se aplicar os algoritmos associados às tarefas de regras de associação sobre *banco de dados tradicionais* é referente ao grande número de conjuntos freqüentes (ou grandes) que são gerados como resultado. Na verdade, esse número pode ser visto em uma ordem de complexidade exponencial em relação ao número de itens (dados) que alimentam o banco de dados, em virtude da maioria desses algoritmos tomarem como base o algoritmo APRIORI [Agrawal e Srikant 1994].

Ao se aplicar as técnicas de mineração de regras de associação sobre um *domínio geográfico*, os problemas para a geração de conjunto de itens freqüentes reduzem consideravelmente, uma vez que a base de dados aos quais são extraídos padrões é composta por *predicados espaciais* (relacionamentos espaciais) entre os objetos geográficos armazenados nos bancos de dados.

Por outro lado, alguns problemas se tornam inerentes, tais como: as *dependências geográficas* "bem conhecidas" (relacionamentos espaciais óbvios) e a redundância de relacionamentos existentes entre certas entidades geográficas, ou seja, relacionamentos que podem ser extraídos a partir de outros relacionamentos.

Alguns trabalhos têm sido feitos para otimizar os algoritmos utilizados em tarefas de regras de associação sobre dados geográficos [Dan et. al. 2003][Sherry et. al. 2002] [Dan et. al. 2003][Dan et. al. 2004]. Contudo, grande parte deles não trata as dependências geográficas como conhecimento prévio de entrada para o algoritmo. Em contrapartida, um trabalho feito neste contexto pode ser encontrado em [Bogorny 2006]. Neste caso, o uso prévio de conhecimentos semânticos pode eliminar estas informações consideradas redundantes ou desnecessárias, tornando todo o processo de mineração mais eficiente.

Por outro lado, existem muitos estudos relativos ao desenvolvimento de tecnologias, modelos e arquiteturas para o projeto de *Linguagens de Consultas Visuais* (*LCV*) sobre dados geográficos. A característica fundamental de uma LCV está relacionada ao fato dos usuários pensarem graficamente no momento em que estão formulando a sua consulta. Entretanto, apesar da grande maioria dos sistemas existentes possibilitarem gerar representações gráficas, apenas poucos deles suportam consultas visuais [Soares 2002]. Além disso, ao realizar uma busca sobre dados em um domínio geográfico, as cláusulas que compõem um script SQL através de consultas textuais muitas vezes são confusas e tornam-se bastante difíceis. Através de uma LCV, elementos pictográficos são utilizados para facilitar a busca pelos dados.

Sendo assim, este trabalho se propõe a estender a arquitetura de um ambiente de consultas visuais para bancos de dados geográficos conhecido como ***GeoVisual*** [Soares e Salgado 1999] [Soares 2002], o qual possui internamente a linguagem de consulta visual para dados geográficos conhecida como ***GeoVisualQL (Geographic Visual Query Language)***, incorporando técnicas e algoritmos de descoberta de conhecimento durante uma consulta realizada sob entidades geográficas.

Em extensões futuras deste trabalho, tem-se a pretensão de se fazer uso de conhecimentos semânticos prévios, a sua arquitetura foi construída de tal modo que se permita, em versões futuras, fazer uso de *ontologias* ou *esquemas* de banco de dados a fim de eliminar padrões e regras consideradas "*não interessantes*", tornando o processo de busca mais eficiente.

Assim, esta nova linguagem, a princípio responsável por realizar consultas visuais sobre tarefas de regras de associação existentes em processos de mineração de dados geográficos, foi denominada de ***GeoMiningVisualQL (Geopraphic Mining Visual Query Language)***. Através dela, é possível que usuários (de habilidades e competências distintas) executem, de forma transparente, a conversão entre uma *consulta visual* para um *script de consulta* feito na linguagem **GMQL** ***(Geo Mining Query Language)*** [Koperski 1999], a qual se encontra num processo de transição para utilização padrão sobre as linguagens de consultas realizadas em processos de mineração de dados geográficos.

Em sua filosofia, o ambiente sob o qual a linguagem ***GeoMiningVisualQL*** está inserida, apresenta metáforas visuais baseadas em "*árvores de consultas metafóricas"* e a abordagem de "*fluxo corrente"* na qual separa etapas envolvidas por todo um processo de mineração de dados e provê um caminho simples e intuitivo para configurar qualquer tarefa de mineração sobre dados geográficos.

1.2 ESCOPO DO TRABALHO

O escopo deste trabalho está voltado à especificação de uma linguagem de consulta visual para tarefas de mineração de dados geográficos e a definição de um ambiente visual no qual ela pode ser inserida de tal modo que venha automatizar o processo de busca por informações e facilitar a formulação de consultas a estes dados por diferentes usuários, além de realizar um resgate literário, melhorando, por conseguinte, o estado da arte na área de mineração de dados geográficos.

1.3 OBJETIVOS

Este seção apresenta os principais objetivos a serem alcançados por este trabalho.

1.3.1 Objetivo Geral

Este trabalho tem como objetivo analisar os principais problemas encontrados durante a execução de algoritmos de mineração de dados sob os relacionamentos existentes entre as entidades geográficas, propondo uma linguagem de consulta visual que automatiza as etapas do processo de descoberta de conhecimento em bancos de dados geográficos. Através desta linguagem, o usuário, utilizando-se de elementos (símbolos) pictográficos, poderá formular consultas visuais para encontrar informações entre os dados, além de buscar *padrões e relacionamentos espaciais (predicados espaciais) intrínsecos* e constituintes das entidades geográficas, utilizando técnicas de mineração de dados geográficos.

1.3.2 Objetivos Específicos

- Facilitar o uso de algoritmos de mineração de dados em bancos de dados geográficos através da definição de um ambiente de consultas visuais.
- Especificar formalmente uma Linguagem de Consulta Visual sob Processos de Extração de Conhecimentos em Bases de Dados Geográficas (***GeoMiningVisualQL).***
- Analisar a descrição dos ambientes de domínios geográficos a serem analisados.
- Analisar e propor uma metodologia de representação visual para cada domínio das aplicações.
- Identificar um conjunto mínimo comum de operadores de linguagens de mineração de dados, e propor representações pictóricas para cada um deles;
- Definir símbolos visuais para estes operadores e para as entidades geográficas do domínio estudado;
- Definir uma gramática para a linguagem ***GeoMiningVisualQL***.
- Descrever os processos de tradução entre consultas visuais feitas em ***GeoMiningVisualQL*** e consultas textuais ***GMQL***;

- Desenvolver um protótipo denominado ***GeoMiningVisual*** com as funcionalidades a serem oferecidas pela linguagem ***GeoMiningVisualQL***.
- Validar as funcionalidades da linguagem através de exemplos práticos de consultas.

1.4 ESTRUTURA DO TRABALHO

Esta dissertação está organizada em **seis capítulos** os quais levam em considerações diferentes aspectos, desde os conceitos, metodologias e técnicas envolvidas na pesquisa até o desenvolvimento e validação do trabalho proposto. Neste capítulo, foi apresentado o contexto no qual este trabalho se insere, através de sua motivação, escopo e objetivos. O restante da dissertação está estruturado da maneira descrita a seguir.

No **capítulo 2 (Fundamentação Teórica)** são apresentados os principais conceitos básicos a cerca dos ***Bancos de Dados Geográficos***, além de abordar os processos mais importantes para ***Descoberta de Conhecimento em Banco de Dados***. Em seguida, será apresentada uma visão geral sobre as *primitivas* que constituem as *linguagens de mineração de dados tradicionais e geográficos* junto aos conceitos e técnicas mais utilizadas no contexto dos ***Ambientes de Consultas Visuais***.

O **capítulo 3 (Trabalhos Relacionados)** apresenta alguns trabalhos existentes na literatura, envolvidos e relacionados com os conceitos básicos da definição do tema proposto nesta pesquisa, mostrando sua viabilidade.

No **capítulo 4 (A Linguagem GeoMiningVisualQL)**, é apresentado o projeto de construção da linguagem para mineração de dados geográficos, denominada, a princípio, **GeoMiningVisualQL**.

O **capítulo 5 (O Ambiente GeoMiningVisuaL)** apresenta o projeto de construção de um ambiente visual, através do qual se insere a *linguagem de consulta visual* proposta nesta dissertação, definindo sua arquitetura e principais características, além de mostrar suas funcionalidades e a aplicabilidade do mesmo através de alguns exemplos hipotéticos.

Por fim, o **capítulo 6 (Contribuições e Trabalhos Futuros)** apresenta a conclusão do trabalho desenvolvido, além de mostrar as principais dificuldades encontradas e as melhorias e relevâncias a serem utilizadas em versões futuras.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais conceitos teóricos e as diversas tecnologias a serem utilizadas nesta pesquisa, mostrando, por conseguinte, a relevância deste trabalho perante as técnicas atuais de descoberta de conhecimentos sobre dados geográficos.

2.1 BANCOS DE DADOS GEOGRÁFICOS

Os **Bancos de Dados Geográficos** (**BDGs**) armazenam dados que estão associados a uma determinada *localização geográfica* e possuem uma *dimensão espacial* que pode ser representada geometricamente através de um ponto, linha ou polígono (Figura 2.1). Muitas vezes esses dados são vistos como ***feições geográficas***, ou seja, abstrações de fenômenos reais, suportados por um sistema de coordenadas geográficas e um domínio temporal [Barros 2009]. Sua representação pode ser proveniente de imagens de satélite, modelos numéricos de terrenos (Figura 2.2), ambientes espaciais, levantamentos demográficos, sistemas de cartografia digital, etc. Nesse contexto, o conceito de entidade geográfica é visto como um fenômeno ou objeto do mundo real que, num determinado instante do tempo, possui atributos associados à sua localização na superfície terrestre.

Figura 2.1: Exemplo de uma imagem de satélite.

Fonte: *http://3.bp.blogspot.com/_r_UTib8CZWM/SBeFQ_qOYBI/AAAAAAAAA_M/5e4fYRmTZdE/s400/satelite%5B1%5D.jpg* (Acesso em 05/2009).

Figura 2.2: Exemplo de um modelo numérico de terreno através de isolinhas[1].

Fonte: *http://www.ciadaescola.com.br/zoom/imgs/342/image003.jpg* (Acesso em 05/2009).

Um dado geográfico apresenta ***atributos não-espaciais ou convencionais*** (nome, população, idade, cor, custo, entre outros) *e* ***atributos espaciais*** que são representados através de coordenadas geométricas espaciais as quais identificam a sua localização sobre a superfície terrestre. Conforme [Soares 2002], dados geográficos representam objetos ou fenômenos cuja localização geográfica é uma característica essencial à informação e indispensável a sua análise.

De maneira geral, podemos dizer que os BDGs são coleções de dados geo-referenciados que são manipulados por um **Sistema de Informação Geográfica** (**SIG**). Segundo [Worboys e Duckam 2004], os SIGs são sistemas computacionais capazes de capturar, modelar, armazenar, recuperar, manipular, analisar e apresentar dados geográficos. Em geral, os SIGs utilizam um conjunto de *camadas* para armazenar as informações sobre o mundo real. Essas camadas podem ser separadas baseadas na geometria de cada entidade geográfica de tal forma que a sobreposição de todas elas constitui a informação original (imagem real), conforme ilustra a Figura 2.3.

O *atributo espacial* de um dado geográfico pode ser constituído de uma referência explícita utilizando-se diretamente das suas coordenadas geográficas (latitude e longitude) ou por uma referência implícita, tal como um endereço ou um código postal. Um processo de geo-codificação, baseado no modelo de camadas utilizado pelos SIGs pode ser utilizado para criar referências geográficas explícitas a partir de referências geográficas implícitas [Sizo, Silva e Bittencourt 2002].

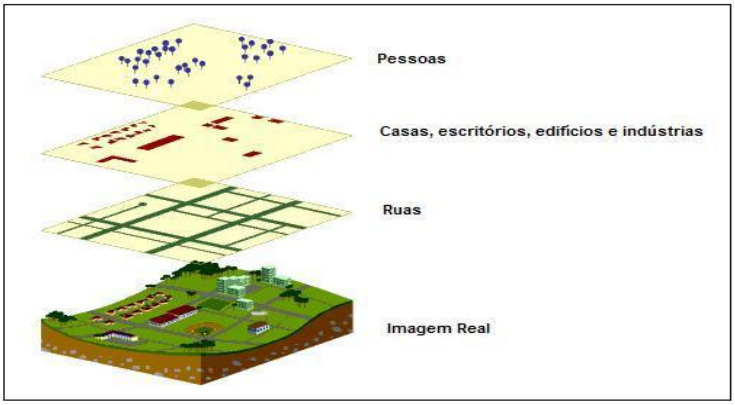


Figura 2.3: Camadas de Referências Geográficos.

Fonte: *(Adaptado de [Soares 2002]).*

Sendo assim, uma característica importante de um *Sistema de Informação Geográfica* é a sua capacidade de integrar informações geográficas de diversas e distintas fontes em uma mesma fonte (base) de dados.

Além disso, entidades espaciais (*feições espaciais*) em banco de dados geográficos são, geralmente, armazenadas em diversas relações, uma vez que muitos bancos de dados geográficos seguem abordagens *relacionais* ou *objeto-relacionais*. A Figura 2.4 ilustra um exemplo de um domínio geográfico relativo à cidade de João Pessoa, através do qual pode ter seus dados extraídos e armazenados em bancos de dados relacionais. Assim, têm-se *ruas*, *rios*, *portos* e *viadutos* representados sob **entidades geográficas** vistas em diferentes relações apresentando *atributos espaciais* e *não-espaciais*.

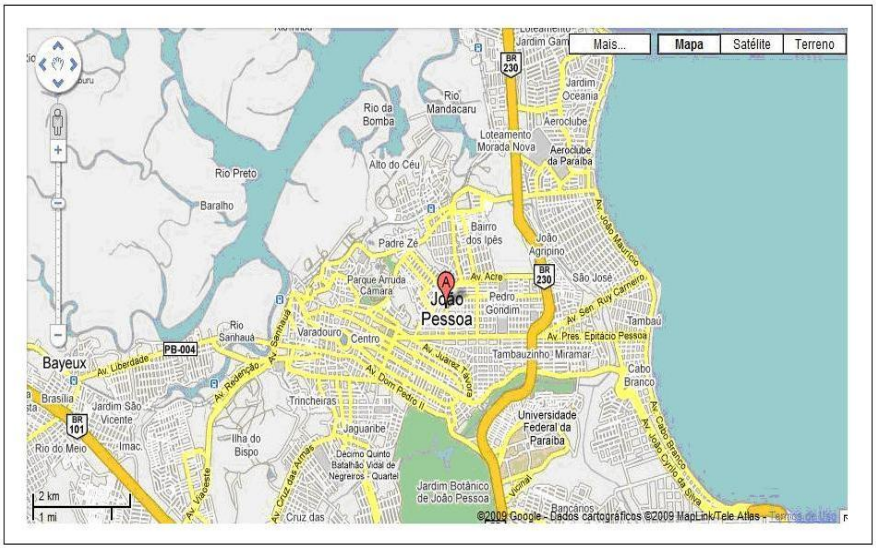


Figura 2.4: Domínio Geográfico da cidade de João Pessoa.

Fonte: *http://maps.google.com.br/maps?hl=pt-BR&tab=wl.* (Acesso em 05/2009).

Nas *Tabelas 2.1, 2.2, 2.3 e 2.4*, são apresentados alguns exemplos possíveis de relações que identificam algumas entidades geográficas que podem ser extraídas a partir da Figura 2.4 Observe que os *atributos espaciais* em cada entidade são visualizados no atributo "*forma/localização*". Além disso, tais entidades possuem relacionamentos espaciais intrínsecos entre si, tais como: *contém, cruza, toca, intersecta, etc*.

Estes relacionamentos são de fundamental importância visto que eles podem afetar o comportamento de outras entidades geográficas na vizinhança. Em conseqüência disso, as restrições existentes oriundas dos relacionamentos espaciais entre entidades geográficas, conhecidos como ***predicados espaciais***, tornam-se a *principal característica* dos dados geográficos a ser considerada durante a etapa de mineração ao longo de todo processo de descoberta de conhecimento sob bases de dados geográficas.

Tabela 2.1: Entidade geográfica Avenida.

Avenida_Rua		
id	***Nome***	***forma/localização***
0	Dom Pedro II	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
1	Epitácio Pessoa	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
2	Senador Ruy Carneiro	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
3	Cruz das Armas	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
4	João Machado	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
5	Cabo Branco	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)

Tabela 2.2: Entidade geográfica Rio.

Rio		
id	***Nome***	***forma/localização***
0	Sanhauá	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
1	Mandacaru	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
2	Preto	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
3	Jaguaribe	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)

Tabela 2.3: Entidade geográfica Porto.

Porto		
id	***Nome***	***forma/localização***
0	Porto de Cabedelo	ponto(x_1,y_1)

Tabela 2.4: Entidade geográfica Viaduto.

Viaduto			
id	***nome***	***ano_construcao***	***forma/localização***
0	Imperatriz Leopoldina	1974	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
1	Ayrton Senna	1978	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)
2	Tancredo Neves	1995	multi-linha($[x_1,y_1];[x_2,y_2];...;[x_n,y_n]$)

Nas tabelas vistas anteriormente, é fácil ver que *id, nome e ano_construcao* são **atributos não-espaciais** (também conhecidos como *convencionais* ou *textuais*) e relatam informações que descrevem ou caracterizam o dado geográfico. Além desses tipos de atributos, um dado geográfico pode possuir *características temporais* relativas à forma como os esses dados podem sofrer certas variações durante um período de tempo e *características gráficas* associadas a sua *representação pictográfica* ou *geométrica*. Sendo assim, pode-se dizer que um dado geo-referenciado é caracterizado através de sua descrição, sua posição geográfica, os relacionamentos espaciais existentes em relações a outras entidades geográficas e um intervalo de tempo correspondente à validação do fenômeno geográfico.

2.1.1 Armazenamento de dados espaciais

Diante das diversas formas sob a qual um dado geográfico pode ser estruturado, duas abordagens têm se destacado: a **representação matricial (***raster***)** e a **representação vetorial**.

A representação vetorial é mais usada para descrever aspectos discretos. Esta estrutura não é indicada para representar características continuamente variantes, como o *clima* de uma região. Neste caso, faz-se uso da representação matricial, utilizadas principalmente para descrever fenômenos que variam com o tempo.

Na estrutura vetorial, entidades geométricas, tais como ponto, linha e polígono ou objeto complexo (utilizados para extrair representações gráficas de entidades espaciais) são codificados através de coordenadas geográficas em relação a um sistema de coordenadas sob um plano cartesiano bidimensional.

Em contrapartida, a **estrutura matricial** ou *raster* (Figura 2.5) é vista como uma coleção de células de uma *malha regular* ou como um *mapa digitalizado* através de uma *matriz de pontos*. Neste caso, cada célula armazena um valor que corresponde a um tipo de entidade geográfica pertencente a um determinado domínio geográfico como, por exemplo, um *rio*, um *solo* ou uma *vegetação*.

Dá-se o nome de **tesselações** a uma estrutura matricial formada por uma superfície contínua representada através de um conjunto de unidades geométricas básicas (Figura 2.6).

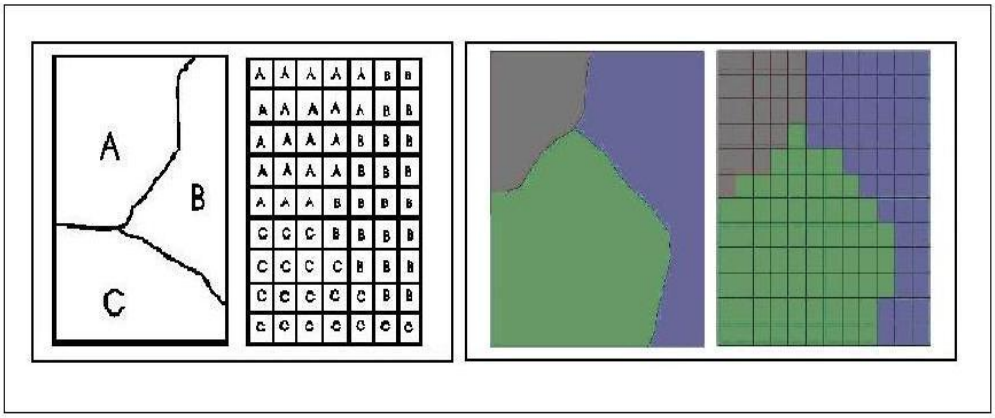


Figura 2.5: Exemplos de representações matriciais.

Fonte: *http://www.dpi.inpe.br/spring/teoria/introdu1/img00002.gif* (Acesso em 06/2009).

.

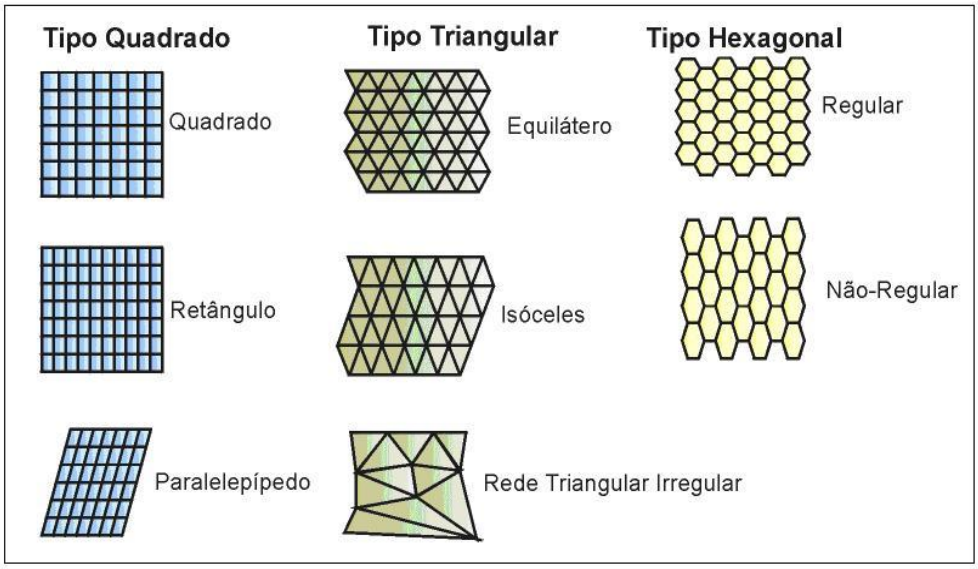


Figura 2.6: Exemplo de tesselações.

Fonte: *http://www.ufes.br/~geoufes/lgu/SIG/TransparenciasApostilaTeoricaSIG/Capitulo4.doc* (Acesso em 07/2009).

A diferença ente essas duas representações (matricial e vetorial) é visualizada através da sobreposição de uma imagem vetorial sobre uma imagem matricial. Vale ressaltar que quando uma determinada imagem é analisada em maior detalhe, a estrutura matricial apresenta uma maior perda de qualidade em relação a uma estrutura vetorial, conforme ilustra a Figura 2.7.

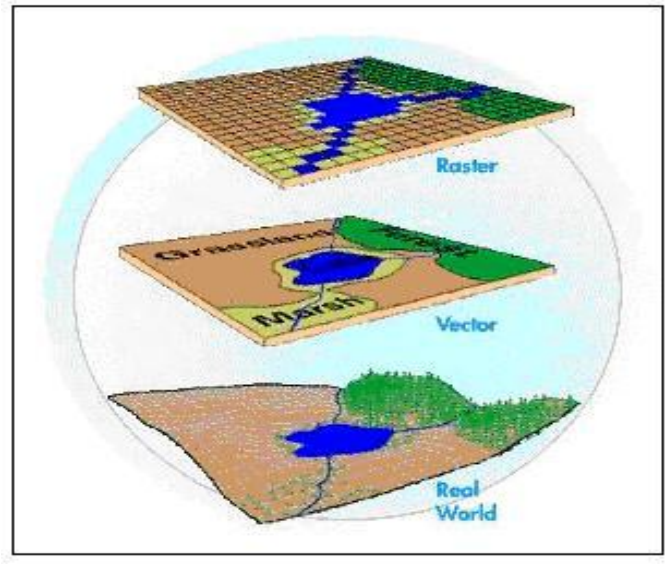


Figura 2.7: Sobreposição de representações matricial e vetorial a partir de uma imagem real.

Fonte: *[Soares 2002]*.

2.1.2 Dependências Geográficas

É importante destacar que ao se extrair *relacionamentos espaciais* sobre um domínio geográfico de estudo, durante um processo de descoberta de conhecimento, muitos desses relacionamentos podem ser vistos como "*interessantes",* enquanto que outros podem ser "*não-interessantes"*.

Um relacionamento é dito *interessante* quando o mesmo apresenta, ou traz consigo, um conhecimento novo ao usuário que até então era desconhecido do mesmo. Por exemplo, ao minerar uma região geográfica marcada pela presença de várias indústrias pode ser extraído um relacionamento (expressando um conhecimento inovador) que reflete a causa de doenças da população que vive ao redor dessa região. Por outro lado, um relacionamento *não-interessante* está relacionado a um conceito já conhecido e aceito pela maioria das pessoas. Por exemplo, a extração de localizações geográficas dos postos de gasolinas sob uma região, através da qual resultam intersectando rodovias.

Um aspecto importante a ser considerado sobre um dado geográfico é a sua *referência espacial*. Isto quer dizer que a sua análise e interpretação depende da sua localização em relação a outras entidades e, desta forma, eles não podem ser considerados isolados no espaço. Sendo assim, uma entidade geográfica possui três tipos básicos de relacionamentos: *obrigatórios*, *possíveis* e *proibitivos*.

Um **relacionamento obrigatório** entre duas entidades espaciais sobre uma superfície terrestre caracteriza uma **dependência geográfica**. Baseado nas Tabelas 2.1 *e*

2.4 e no contexto do domínio geográfico expresso na Figura 2.4, é fácil ver que a entidade geográfica ***Viaduto*** possui uma *dependência geográfica* em relação à entidade ***Avenida_Rua***, uma vez que existe um relacionamento espacial *obrigatório* intrínseco entre elas (***é_um****(Viaduto)* → ***cruza****(Avenida_Rua))*. Outro exemplo interessante de dependência geográfica, inserido no mesmo contexto, pode ser encontrado entre relacionamentos de entidades geográficas que referenciam **Pontes** e **Rios**, tais como ***é_uma****(Ponte)* → ***cruza****(Rio)*.

Um **relacionamento possível** é aquele caracterizado como principal objeto de estudo em um processo de descoberta de conhecimento em bancos de dados geográficos uma vez que eles são descritos por predicados espaciais ocultos e não-óbvios, sendo, neste caso, não conhecidos pelo usuário.

Um **relacionamento proibitivo**, assim como os obrigatórios, pode ser considerado uma associação espacial *não-interessante.* Um relacionamento proibitivo é caracterizado por ser óbvio e impossível, além de não representar nenhum conhecimento novo ou expressivo para o usuário. Um exemplo de um relacionamento impossível seria: ***é_um****(Mar)* → ***contém*** *(Rua).*

Uma característica importante e presente em poucos estudos na literatura consiste em utilizar as dependências geográficas entre entidades espaciais como um conhecimento prévio para entrada de algoritmos de mineração de dados. Neste sentido, [Bogorny 2006] utiliza conhecimentos de **ontologias** e **esquemas** de bancos de dados para ilustrar a dependência geográfica como representação semântica dos dados geográficos.

2.1.3 Tipos de relacionamentos espaciais

Um dos principais objetivos de estudos em *SIGs* tem sido voltado à tentativa de compatibilizar as diferentes estruturas de dados geográficos, tais como em [Soares 2002]. Neste caso, alguns trabalhos procuram estender o padrão SQL, adicionando operadores espaciais, porém mantendo toda a filosofia relacional do SQL.

Em paralelo, algumas soluções tentam adicionar operadores espaciais às novas linguagens propostas, permanecendo a estrutura e filosofia básica da linguagem SQL, uma vez que tal linguagem não apresenta, em sua natureza, operações métricas e topológicas sobre dados espaciais [Huang e Svensson 1993], [Câmara 1995] e [Wang et al. 2000]. Além disso, muitas extensões da linguagem SQL visam adicionar informações espaciais no projeto de banco de dados relacionais [Adam e Gangopadhyay 1997].

Existem na literatura, basicamente, três tipos de relacionamentos espaciais: *topológicos*, *direcionais* e de *distância (ou métricos).*

Os **relacionamentos topológicos** são caracterizados pelo tipo de intersecção existente entre duas entidades espaciais (Figura 2.8). Além disso, permanecem invariantes quando sofrem transformações de *rotação* e *translação*. Na literatura, há muitas abordagens que definem formalmente um conjunto de relacionamentos topológicos entre certas representações geométricas, tais como o *ponto*, a *linha* e o *polígono*.

Um **ponto** é definido como uma representação geométrica *adimensional*, tendo o seu interior definido como uma célula na representação matricial e o seu exterior como o complementar do seu interior em relação à matriz espacial.

Uma **linha** pode ser visualizada na matriz espacial como um conjunto de células seqüenciais e interconectada.

Um **polígono** é uma representação geométrica definida como células homogêneas interligadas em um espaço bi-dimensional da matriz espacial e, normalmente, representado através de uma região espacial.

Muitas destas abordagens citadas anteriormente têm sido baseadas nos modelos de interseções proposto por [Egenhofer e Hering 1991]. Nesse contexto, o **modelo de 4-interseções** descreve as relações topológicas binárias em termos das interseções dos seus *interiores* e *limites* entre dois objetos espaciais e o **modelo de 9-interseções** inclui também a interseção dos *exteriores* entre dois objetos espaciais. Tais interseções podem ser combinadas com operadores lógicos **and** (^) e **or** (**v**). [Hadzilacos e Tryfona 1992] estendeu a abordagem de [Egenhofer e Hering 1991] para a combinação de pontos, linhas e polígonos e este modelo de interseção provê oito relacionamentos topológicos binários possíveis: *cruza, contém, dentro, cobre, é coberto por, igual, disjunto* e *sobrepõe.*

Os **relacionamentos direcionais** são classificados de acordo com as localizações das entidades espaciais no espaço, segundo uma determinada ordem espacial entre elas (Figura 2.9). Um modelo particular dos relacionamentos direcionais são os relacionamentos cardinais que descrevem as relações direcionais através de um conjunto de símbolos. Na verdade, tal conjunto pode ser expresso como um conjunto de triplas **<*O, r, D*>**, onde r é visto como o relacionamento espacial entre O e D que são, respectivamente, o objeto espacial de origem e o objeto espacial de destino. Nesse contexto, r pode ser visto como um subconjunto do conjunto de pontos cardeais *(norte, sul, leste e oeste)* e do conjunto de pontos colaterais *(nordeste, noroeste, sudeste e sudoeste).*

Os ***relacionamentos métricos*** são baseados na distância euclidiana entre dois pontos geográficos no espaço (Figura 2.10). Neste caso, é possível ter-se uma noção exata e mais precisa sobre os seus posicionamentos no espaço.

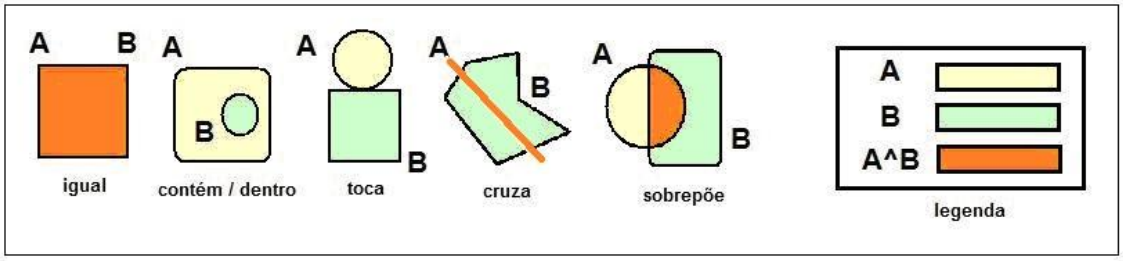


Figura 2.8: Exemplo de relacionamentos topológicos entre entidades espaciais.

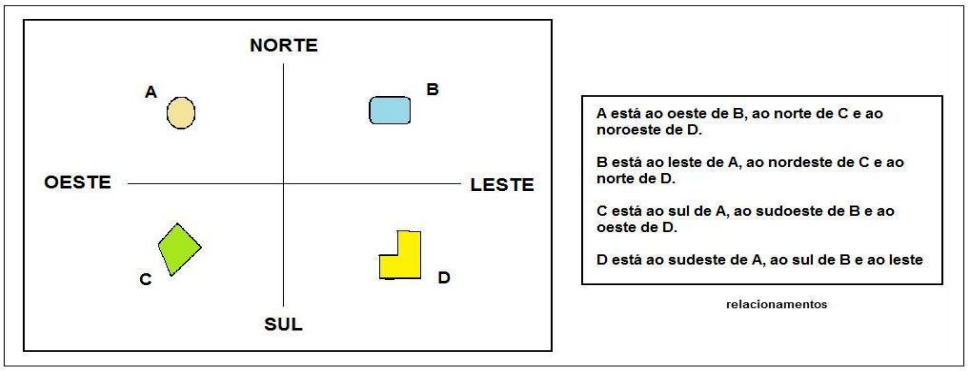


Figura 2.9: Exemplo de relacionamentos direcionais entre entidades espaciais.

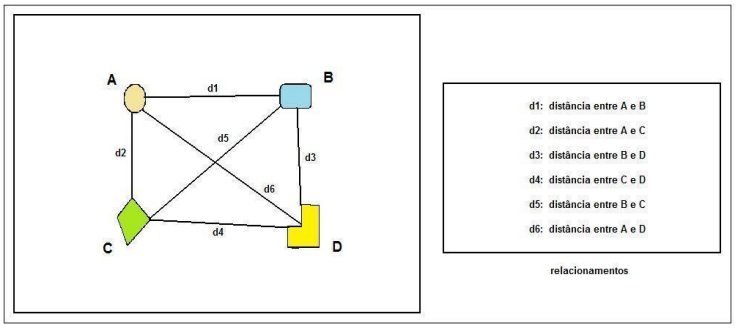


Figura 2.10: Exemplo de relacionamentos métricos entre entidades espaciais.

Devido ao fato de que os fenômenos geográficos podem sofrer modificações durante o decorrer do tempo, é possível classificar um tipo especial de *associação* chamado **de relacionamento temporal**. Neste contexto, em [Hornsby e Egenhofer 1999] é definida uma *linguagem de descrição de mudanças* chamada ***CDL (Change Description Language)*** baseada nas alterações sob a identidade de um objeto com o passar do tempo, caracterizando estados diferentes sobre um mesmo objeto, levando-se em consideração,

também, a fundamentação de noção de existência de tal objeto durante uma mudança de um fenômeno geográfico. É importante ressaltar a distinção entre os dois *estados de não-existência* de um objeto:

- *Não-existência sem história* - objetos que não existem e nunca existiram.
- *Não-existência com história* - objetos que não existem, mas já existiram anteriormente (no passado).

As mudanças entre os estados de um objeto no tempo são dadas através de transições entre estes estados. Por exemplo, um objeto que existe em um prezado momento (*item I* na *Figura 2.11*) após um determinado período de tempo pode passar a não existir (*item III* da *Figura 2.11*), caracterizando uma transição válida entre esses estados.

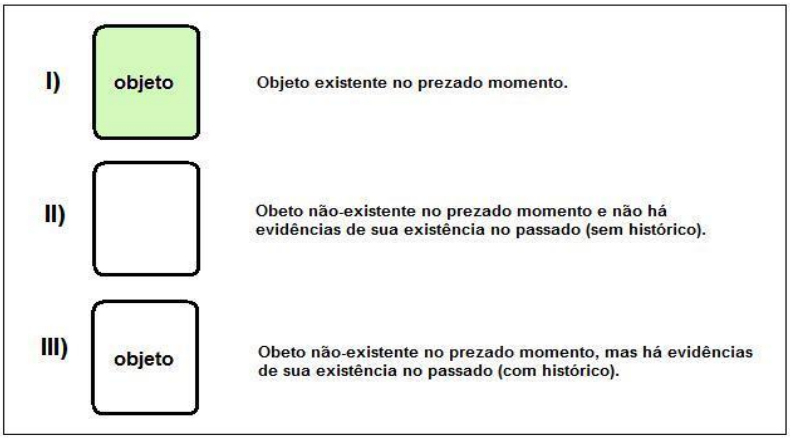


Figura 2.11: Relações de existência de um objeto baseado no tempo.

Existem também os chamados **relacionamentos *fuzzy*** [Bogorny 2003] que estão associados a relacionamentos entre entidades geográficas que não possuem limites (fronteiras) bem definido. Um rio, por exemplo, possui suas margens alteradas durante as estações de verão e inverno.

2.2 DESCOBERTA DE CONHECIMENTO EM BANCOS DE DADOS

As limitações tecnológicas encontradas no passado possibilitavam o armazenamento de um pequeno volume de dados, de tal forma que consultas simples eram suficientes para satisfazer as necessidades de um usuário.

Atualmente, com os grandes avanços da tecnologia, grandes quantidades de dados são armazenados em *Sistemas Gerenciadores de Banco de Dados* de modo que a necessidade de entender e compreender como esses dados estão relacionados se torna cada vez mais expressiva, uma vez que ultrapassam a capacidade humana de analisar e extrair conhecimentos a partir desses dados. Neste contexto, surgem técnicas e estudos da área de **Descoberta de Conhecimento em Banco de Dados** (*DCBD* ou *KDD, Knowledge Discovery Database*), baseadas nos princípios da *aprendizagem de máquina, inteligência artificial* e *estatística*. A DCBD é um processo iterativo e envolvido pelas técnicas eficientes e inteligentes de extração de conhecimento.

A **Descoberta de Conhecimento em Bases de Dados Geográficos** é um tipo particular de descoberta, pois está ligada à extração de características e padrões espaciais interessantes, à identificação de relacionamentos entre dados espaciais e não-espaciais, restrições entre objetos geográficos e outras características não explicitamente armazenadas nestes bancos de dados [Bogorny 2003].

Segundo [Koperski, Han e Adhikari 1997], os grandes progressos conseguidos na área de descoberta de conhecimentos se restringiam, quase que por completo, aos bancos de dados relacionais. Atualmente, em muitas bases de dados organizacionais, tais dados se encontram mais complexos, apresentando característica de dimensionalidade espacial no qual não podem ser minerados pelas técnicas de mineração de dados tradicionais. Neste caso, tornam-se imprescindíveis levar em consideração a semântica destes dados durante a aplicação dos algoritmos de mineração.

De uma maneira geral, a semântica dos dados geográficos pode ser expressa através dos *predicados espaciais* existentes entre os relacionamentos.

Em virtude de uma entidade geográfica (objeto) poder alterar ou afetar o comportamento de uma entidade vizinha (próxima a ela), os relacionamentos de vizinhança se caracterizam como de grande importância no processo de extração de conhecimento de dados geográficos. Sendo assim, os algoritmos de mineração utilizam-se deste tipo de informação para poder determinar as possíveis dependências geográficas entre estas entidades.

2.2.1 Etapas do processo de descoberta de conhecimento em banco de dados

O processo de extração de conhecimentos ou interpretação de padrões é dividido em várias etapas, envolvendo desde a *seleção dos dados, pré-processamento, limpeza, mineração de dados* e *sua interpretação final (apresentação).*

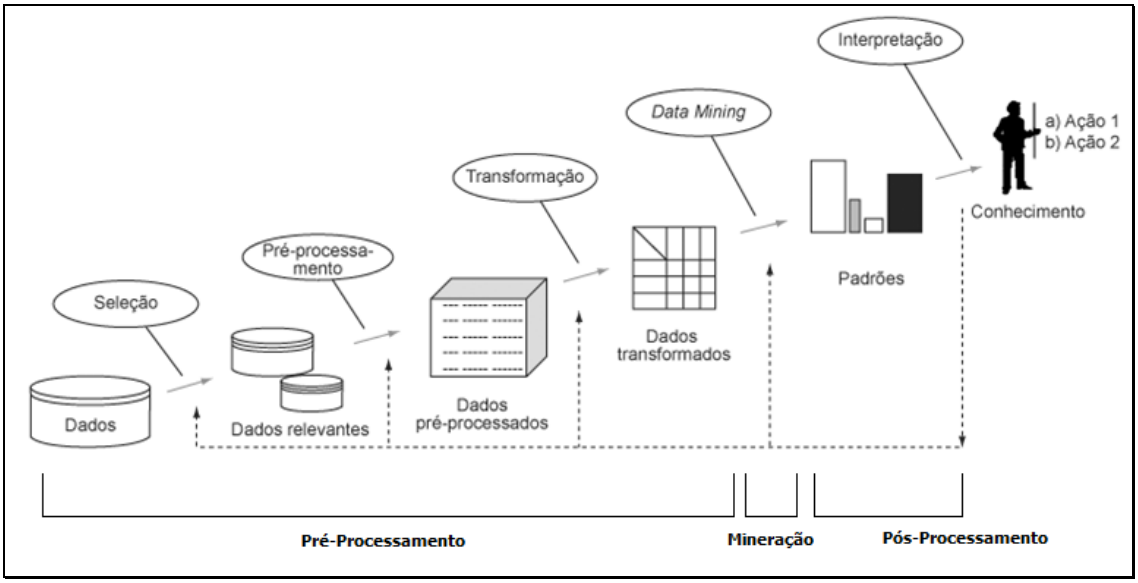


Figura 2.12: Etapas do processo de KDD.

Fonte: *Adaptado de* [Fayyad et al. 1996]

As etapas vistas na Figura 2.12 podem ser organizadas em três grandes fases: ***pré-processamento, mineração*** e ***pós-processamento***.

A fase de *pré-processamento* é a de maior duração durante todo o processo de descoberta de conhecimento e pode ser decomposta dos seguintes passos:

a) determinar metas e objetivos a serem alcançados;
b) realizar a limpeza dos dados, eliminando inconsistências e ruídos;
c) integrar os dados, unindo os dados de diversas fontes;
d) selecionar uma amostra dos dados que serão minerados;
e) converter ou transformar os dados selecionados para uma entrada válida para os algoritmos de mineração.

A *mineração de dados (data mining)* é a principal fase no processo de descoberta de conhecimento. Nesta etapa, são aplicados algoritmos sobre os dados convertidos em um formato válido, realizado na etapa anterior.

Por fim, tem-se a fase de *pós-processamento* no qual são apresentados os padrões extraídos no resultado da execução dos algoritmos de mineração.

2.2.2 Mineração de Dados

A **Mineração de Dados *(Data Mining)*** é uma etapa (passo) em que se aplicam diversos algoritmos sobre os dados de tal forma que se criem "*padrões*" sobre os mesmos. Existem várias tarefas utilizadas na mineração de dados, entre as quais podem ser classificadas em: *regras de associação, clusterização (agrupamento), classificação, outliers, etc.*

As tarefas de **regras de associação** obtêm papel de destaque nos processos de descoberta de conhecimento em banco de dados geográficos. Isto acontece porque os relacionamentos espaciais são as principais características a serem levadas em consideração na descoberta de conhecimentos em bases de dados geográficas. Assim, elas serão mais aprofundadas neste estudo e será vista com maiores detalhes mais adiante.

A técnica de **agrupamento** ou **clusterização** implica em identificar um conjunto de objetos semelhantes, baseadas em *funções métricas* (ou *de distância*), especificados em contextos distintos. Por exemplo, agrupar países de acordo com número da *população, clima* ou *região geográfica*. Segundo [Neves, Freitas e Câmara 2001], a idéia básica da clusterização é agrupar um conjunto de objetos em subconjuntos, de acordo com certos critérios tais como *homogeneidade* (objetos pertencentes a um mesmo cluster são os mais similares possíveis) e *separação* (objetos pertencentes a clusters diferentes devem ser os mais distintos possíveis).

A **classificação** é um método no qual a extração de conhecimento é feita sobre um conjunto de dados relevantes, abstraídos a partir da generalização de conceitos de baixo nível.

Os **outliers** fazem parte de uma metodologia aplicada a dados que se encontram em casos raros, normalmente especificados através de ruídos ou exceções sobre uma análise de um conjunto de dados. Em geral, esses dados confundem um processo de análise e sua influência pode tornar o resultado de um processo de descoberta "*pobre*". Entretanto,

em muitos casos, esses dados podem ser de extrema importância, tal como sob detecções em usos incomuns de cartões de crédito ou transferências bancárias.

2.2.3 Regra de Associação

As ***regras de associação*** consistem na técnica sob a qual itens de uma determinada base de dados aparecem freqüentemente relacionados. São muito utilizadas nas análises transacionais feitas em bancos de dados relacionais. Sendo assim, itens de uma transação implicam em outros itens na mesma transação.

Formalmente, uma regra de associação pode ser vista sob a forma $\boldsymbol{X \rightarrow Y}$, no qual *X* e *Y* são vistos como conjunto de itens que ocorrem em uma determinada transação. Outros autores definem estes conjuntos de itens como predicados no qual *X* representa o antecedente e *Y* o conseqüente.

O principal algoritmo que implementa e extrai regras de associação é o ***APRIORI*** [Agrawal e Srikant 1994]. Sua execução é fundamentada na probabilidade de duas medidas: o *suporte* e a *confiança*. Na literatura, existem outras medidas utilizadas na extração de regras de associação, tais como: ***ganho de entropia*** relacionada à distinção entre informações originais e distribuídas no banco [Morimoto et al. 1998], a ***convicção*** que indica o grau de implicação do antecedente sobre o conseqüente [Brin et al. 1997], o ***índice gini*** que é utilizado para medir o grau de concentração de qualquer distribuição estatística [Fukuda et al. 1996], entre outras.

O **suporte** é o percentual de transações que contém todos os itens na própria relação, ou seja, na própria base de dados inteira. A **confiança** esta relacionada à confiabilidade da regra perante a probabilidade de ocorrência do conseqüente dado que o seu antecedente é satisfeito.

De maneira geral, os problemas de extração das regras de associação podem ser decompostos em dois subproblemas:

a) encontrar todo o conjunto de itens freqüentes ou grandes;
b) gerar regras de alta confiança.

Um conjunto de itens é *freqüente* se o seu suporte (o percentual de transações que contem todos os itens analisados) é pelo menos igual ao suporte definido previamente na execução do algoritmo de mineração.

Uma regra de associação é *forte* ou de *alta confiança* se além de ter um conjunto de itens freqüentes, tiver, também, o grau de confiabilidade maior que certo grau de confiança definido previamente no algoritmo de mineração.

Uma observação importante a ser feita em relação ao algoritmo APRIORI é que descobrir o conjunto de itens freqüentes junto com o seu valor para suporte pode ser um problema significativo se a cardinalidade do conjunto de itens for muito alta. É fácil ver que tal algoritmo possui uma ordem de complexidade exponencial durante a sua execução. Um explicação mais detalhada da execução deste algoritmo pode ser encontrado no Apêndice A.

Alguns algoritmos têm sido propostos para reduzir o custo computacional, o número de candidatos freqüentes e o número das regras de associação. Nesse contexto, os algoritmos de mineração de regras de associação podem ser divididos em duas abordagens:

- Abordagem ***apriori-like*** na qual se utilizam diferentes *medidas* para reduzir o número de candidatos freqüentes [Morimoto et al. 1998] [Brin et al. 1997] [Fukuda et al. 1996] [Silberschatz e Tuzhilin 1996] [Padmanabhan e Tuzhilin 1998];
- Abordagem que geram ***conjuntos freqüentes fechados*** de modo a reduzir o número de candidatos freqüentes e regras que se encontram redundantes [Pasquier et al. 1999].

Vale ressaltar que uma regra de associação é dita redundante se o seu suporte é próximo ao valor pré-definido na análise

2.3 PRIMITIVAS PARA A CONSTRUÇÃO DE UMA LINGUAGEM DE MINERAÇÃO DE DADOS

Ao se realizar uma tarefa de mineração de dados, o usuário deve estar ciente da forma como os dados extraídos deverão ser apresentados. Sendo assim, uma tarefa de mineração pode ser especificada através de uma consulta. Conforme [Kade 2001], uma consulta de mineração de dados pode ser especificada através de certas primitivas (considerações filosóficas) que influenciam diretamente o projeto de construção de uma linguagem de mineração de dados, tais como:

- os *dados relevantes* na tarefa de mineração;
- o *tipo de conhecimento* a ser minerado;
- o conhecimento sobre o contexto do *domínio* a ser minerado;
- as *medidas* interessantes;
- a *apresentação* (*visualização*) dos padrões descobertos.

Baseando-se nestas primitivas, [Han et al. 1996] projetou uma linguagem de consulta para mineração de dados chamada **DMQL *(Data Mining Query Language).*** Uma característica importante de tal linguagem é que além de permitir mineração *ad-hoc* sobre os conhecimentos, a linguagem DMQL adota a sintaxe SQL-Like [Meo, Psaila e Ceri 1996], sendo, neste caso, possível ser integrada facilmente a uma linguagem de consulta relacional. Sua sintaxe é definida a partir de uma **gramática BNF** (também conhecida como ***formalismo de Backus-Naur***), a qual é um padrão bastante utilizado para a descrição sintática de linguagens e, em especial, as linguagens livres de contexto.

Todavia, conforme [Santos Silva 2002], a linguagem DMQL não possui recursos para pré-processamento de dados e possui um número limitado de algoritmos de mineração. Além disso, uma aplicação que implementa esta linguagem, a ***DBMiner*** [DBMiner Technology Inc. 2000], utiliza um recurso de descrição de tarefas através do qual o usuário define a tarefa de mineração a ser realizada e a aplicação, apresentando apenas cláusulas DMQL da tarefa a ser efetuada.

2.3.1 Dados relevantes na tarefa de mineração

Os **dados relevantes** na tarefa de mineração estão relacionados à parte do banco de dados que deve ser investigada. Em geral, o primeiro passo ao se definir uma tarefa de mineração é especificar um subconjunto de dados sobre o qual a mineração será efetuada. Esses dados podem ser associados, no processo de descoberta de conhecimento, ao subconjunto de dados gerados após a fase de pré-processamento, transformando-se, por conseguinte, em um formato válido de entrada para etapa de mineração. A DMQL provê um conjunto de cláusulas que podem ser empregadas para coletar os dados relevantes, tais como:

- ***use database****(database_name)* – escolher a base de dados na qual o processo será efetuado.

- ***use data warehouse****(data warehouse_name)* – especificar o Data Warehouse (DW) a ser utilizado no processo.
- ***from****(table(s)/cube(s))* – especificar a(s) tabela(s) ou cubo(s) (caso seja um DW) envolvidos na consulta.
- ***in relevance to****(attributes_list/dimension(s))* – selecionar a lista de atributos ou dimensões a serem exploradas.
- ***order by****(list)* – ordenar o resultado da consulta.
- ***group by****(list)* – especificar o critério de agrupamento a ser usado.
- ***having****(condition)* – utilizado para restringir condicionalmente uma consulta.

2.3.2 Tipos de conhecimento a ser minerado:

A segunda primitiva, a qual representa o tipo de descoberta a ser minerado, indica a tarefa de mineração a ser efetuada. Em outras palavras, refere-se aos possíveis tipos de mineração, tais como: a **mineração preditiva** (na qual se preocupa em como mostrar certas características dos dados irão se comportar no futuro), a **caracterização** (especifica ou argumenta a causa da tarefa ser realizada), a **classificação** (na qual particiona os dados em diferentes classes ou categorias, baseando-se em combinações de parâmetros), a **discriminação** (com o intuito de discriminar o conceito de uma classe em relação a outras) e a **associação** (com o objetivo de relacionar itens entre uma transação).

2.3.3 Domínio do contexto a ser minerado

A terceira primitiva está baseada na filosofia do contexto ao qual o domínio está inserido. Neste sentido, conceitos hierárquicos permitem o conhecimento em diferentes níveis de abstração.

Existem algumas cláusulas importantes relacionadas ao uso de hierarquias na linguagem DMQL. Dentre elas, duas merecem um grande destaque: a definição de um *esquema hierárquico* e a definição de um *conjunto hierárquico*.

2.3.3.1 Sintaxe de definição de um esquema hierárquico

Definição:

define hierarchy (name) **on** (schema) **as** [(level_list)]

Exemplo: Definir um esquema hierárquico para uma data.

define hierarchy (data_nascimento) **on** (data) **as** [(dia, mês, ano)]

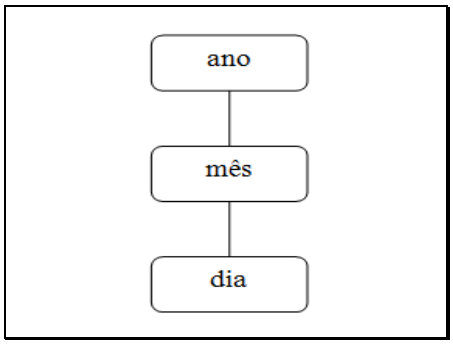


Figura 2.13: Esquema hierárquico para a data de nascimento de um indivíduo.

2.3.3.2 Sintaxe de definição de um conjunto hierárquico

Definição:

define hierarchy (name) **for** (attribute) **on** (object) **as** [(list1;list2;...;listn)]

Exemplo: Definir um conjunto hierárquico a partir da idade de um conjunto de pessoas.

define hierarchy (tempo) **for** (idade) **on** (cliente) **as** (list1;list2;...;listn)]
nível1: {criança, adolescente, adulto } < nível0: pessoa
nível2: {0, . . . , 12} < level1: criança
nível2: {13, . . . , 18} < level1: adolescente
nível2: {19, . . . , 60} < level1: adulto
nível2: {61, . . . } < level1: idoso

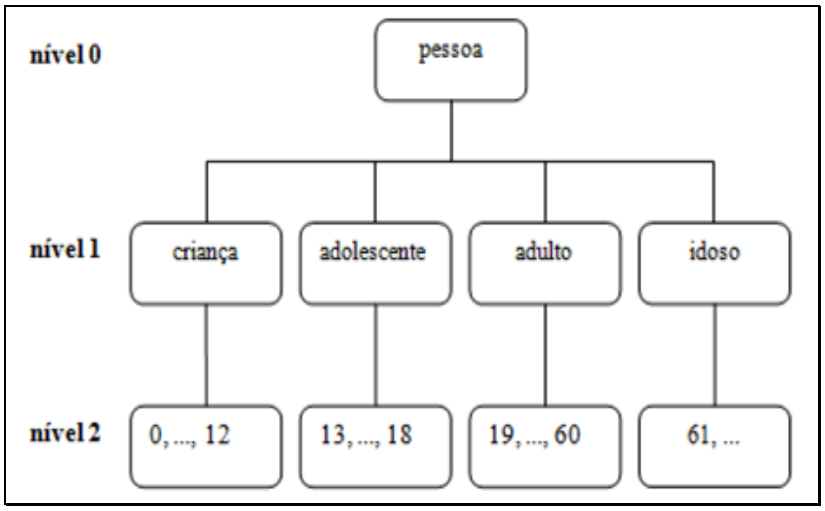


Figura 2.14: Conceito hierárquico sobre o atributo "idade".

2.3.4 Medidas interessantes

As *medidas interessantes* correspondem aos parâmetros que os usuários podem utilizar a fim de controlar o número de padrões retornados durante uma extração de conhecimento. Estas medidas incluem o *suporte*, a *confiança*, *ganho de entropia* [Morimoto et al. 1998], *convicção* [Brin et al. 1997], *índice de gini* [Fukuda et al. 1996], entre outros.

Sintaxe geral:

with (measure) **threshold** = (value)

Exemplos:

with (suporte) **threshold** = (0.65)
with (confiança) **threshold** = (0.55)

2.3.5 Apresentação (visualização) dos padrões descobertos

Nesse contexto, o usuário especifica a forma sob a qual os padrões descobertos através da consulta efetuada serão apresentados. Dentre as principais formas utilizadas para a visualização dos dados têm-se: tabelas, gráficos, árvores de decisão, cubos, etc. Na DMQL, a sentença padrão é representada a seguir:

display as (forma)

2.3.6 Exemplo de uma consulta geral feita através da linguagem DMQL

Na discussão anterior foram apresentadas algumas primitivas que constituem o padrão de projeto da linguagem DMQL. A seguir, tem-se um exemplo prático (Figura 2.15) de uma consulta generalizada.

Exemplo:

Suponha que um supervisor de vendas de uma certa loja de produtos eletrônicos queira caracterizar os hábitos de compras de seus clientes. Neste caso, considere a idade

desses clientes, o produto comprado e o lugar em que este produto foi pedido como parâmetros essenciais para esta análise e que esses hábitos estejam associados a compras que custam menos de 100 dólares. Além disso, é desejado buscar o percentual de clientes que possuem esta característica e que é de interesse, apenas, buscar compras feitas no Canadá e pagas com o cartão de crédito da American Express. Feito isso, os dados provenientes dessa mineração devem ser postos em uma tabela.

```
USE DATABASE AllElectronics db
USE HIERARCHY location hierarchy FOR B.address
MINE CHARACTERISTICS AS customerPurchasing
ANALYSE count%
IN RELEVANCE TO C.age, I.type, I.place made
FROM customer C, item I, purchases P, items sold S, works at W, branch B
WHERE I.item ID = S.item ID AND S.trans ID = P.trans ID AND P.cust ID =
C.cust ID AND P.method paid = "AmEx" AND P.empl ID = W.empl ID and
W.branch ID = B.branch ID AND B.address = "Canada" AND I.price _ 100
DISPLAY AS table
```

Figura 2.15: Exemplo de consulta feita em DMQL.

Fonte: [Han e Kamber 2001].

2.4 LINGUAGEM GMQL

Esta seção irá abordar, em mais detalhes, a linguagem base utilizada neste trabalho. Em sua essência, a linguagem GMQL é utilizada para realizar a extração de conhecimentos sobre áreas de domínio geográfico.

2.4.1 Visão Geral

Na seção anterior foi visto que uma das mais importantes partes no processo mineração de dados é a formulação e padronização de uma linguagem de consulta. Por outro lado, as tarefas de mineração podem ser feitas em dados mais complexos os quais apresentam, em sua característica, fortes evidências em dimensões espaciais.

Nesse contexto, é necessário haver uma extensão da linguagem DMQL sob a qual possam suportar dados geográficos. Esta extensão foi proposta em [Koperski 1999] através da construção da **linguagem GMQL** (Geo-Mining Query Language).

Em virtude do projeto da linguagem GMQL ser baseado no projeto da DMQL, alguns princípios são levados em consideração, tais como:

- Os dados sobre os quais o processo de mineração é efetuado estão armazenados tanto através de atributos não-espaciais quanto na forma de atributos espaciais;
- Existem conceitos hierárquicos tanto para atributos espaciais ou não-espaciais;
- Um conjunto de dados relevantes para a realização da tarefa de mineração deve ser especificado utilizando-se conceitos de alto nível;
- A linguagem deve ser capaz de dispor cláusulas que selecionem os dados usados no processo de descoberta;
- A linguagem deve ser capaz de minerar diferentes tipos de conhecimento;
- Os relacionamentos existentes entre os dados espaciais devem ser especificados de maneira similar às consultas ***Spatial SQL*** [Egenhofer 1994];
- Devem existir certos parâmetros os quais restrinjam os dados a serem selecionados em um processo de descoberta de conhecimento.

Por ser uma linguagem fundamentada na linguagem DMQL, a sintaxe da linguagem GMQL também pode ser apresentada através de uma extensão de uma gramática BNF, na qual ***"[ ]"*** representa 0 ou uma ocorrência, ***"{ }"*** representa 0 ou mais ocorrências e palavras em ***negrito*** representam palavras reservadas da linguagem.

MINE <*rule_specification*>
[**ANALYSE** <*aggregate_list*>]
WITH RESPECT TO <*attribute_list*>]
FROM <*relation(s)*>
[**WHERE** <*condition(s)*>]
[**GROUP BY** <*attribute_list*>]
[**HAVING** <*condition(s)*>]
{**SET** <*threshold_specification*>}
{<*hierarchy_specification*>}

Figura 2.16: Sintaxe de uma consulta geral feitam em GMQL.

A cláusula **MINE** <*rule_specification*> especifica o tipo ou o nome da regra a ser minerada. A cláusula **ANALYSE** <*aggregate_list*> especifica quais *atributos* e *tabelas* (quando as fontes de origem são ***Bancos de Dados***) ou quais os *atributos*, *predicados* ou *funções* (quando as fontes de origem são geradas a partir de um ***Data Warehouse***). A cláusula **WITH RESPECT TO** <*attribute_list*>] seleciona uma lista de atributos, predicados ou funções relevantes na tarefa de mineração. Além disso, esta lista pode incluir descrições espaciais de objetos que são representados como atributos geométricos. Caso seja realizada uma análise multidimensional, a lista poderá, também, incluir dimensões de um cubo de dados. As cláusulas **FROM** <*relation(s)*> [**WHERE** <*condition(s)*>] [**GROUP BY** <*attribute_list*>] [**HAVING** <*condition(s)*>] são utilizadas para recuperar um conjunto relevante de dados através uma consulta *SQL* comum. Por fim, a cláusula {**SET** <*threshold_specification*>} especifica um "*valor limiar*" em relação a uma determinada medida parametrizada que esteja sendo utilizada na tarefa de mineração e a cláusula {<*hierarchy_specification*>} especifica os conceitos hierárquicos utilizados no processo.

2.4.2 Especificação dos dados relevantes

Como visto anteriormente, um objeto espacial pode conter atributos espaciais ou não-espaciais. Além disso, os atributos não-espaciais podem se encontrar relacionados espacialmente através dos *predicados espaciais*, ou seja, através de relacionamentos *métricos*, *topológicos* ou *direcionais.*

Quando um *predicado espacial* está sendo usado em uma tarefa de mineração de dados ele deve ser especificado através da cláusula ***WITH RESPECT TO***. Em geral, esta cláusula pode ser usada junto com os seguintes templates:

- ***G_CLOSE_TO***(spatial_term, spatial_term, "distance_specification");
- ***DIRECTION***(spatial_term, spatial_term);
- ***TOPOLOGY***(spatial_term, spatial_term).

Conforme [Koperski 1999], um spatial_term é um atributo espacial ou um resultado de uma operação espacial executada sobre um objeto espacial.

Exemplo: *Recuperar, numa determinada base de dados geográficos específica, escolas e igrejas eqüidistantes 3 km.*

MINE SPATIAL ASSOCIATIONS DESCRIBING "School"
WITH RESPECT TO School.geo,
G_CLOSE_TO(School.geo, Church.geo, "3 km"),
Church.geo
FROM School, Church.

Observe que, no exemplo anterior, os dados a serem examinados restringem-se ao espaço geográfico existente (se houver) entre faculdades e igrejas que estão limitadas geograficamente a um raio igual a 3km de distância. Além disso, a cláusula ***FROM*** especifica os nomes das relações utilizadas no processo de mineração.

Vale ressaltar que se a mineração estiver sendo feita sobre um ***Data Warehouse*** (***DW***), então um cubo de dados será utilizado. Nesse contexto, a cláusula ***WITH RESPECT TO*** especifica as dimensões do cubo e a cláusula ***ANALYSE*** pode ser utilizada para representar as medidas armazenadas no cubo.

2.4.3 Tipos de descoberta de conhecimento

Existem diversas formas sob as quais as consultas podem ser utilizadas, de acordo com a tarefa de mineração que desempenham.

Os exemplos desta seção foram retirados e/ou adaptados de *[Koperski 1999]*.

2.4.3.1 Characteristic Rules

Especifica ou argumenta a causa da tarefa ser realizada através da generalização de um conceito satisfeito por todos ou muitos dos objetos selecionados. Neste caso, a visualização do conjunto relevante de dados pode ser apresentada através de diferentes níveis.

Exemplo: Caracterizar os estados de um país usando conceitos hierárquicos sobre os atributos espaciais e não-espaciais.

MINE CHARACTERISTICS DESCRIBING "USA_states"
ANALYSE states_census.geo
WITH RESPECT TO pop90, med_farm_income, with_bachelor, degp
FROM States_census

2.4.3.2 Comparison Rules

São regras que associam objetos de diferentes classes em um conjunto relevante de dados armazenados em um banco de dados. Em geral, comparam um conjunto de classes chamadas de ***target class*** com um outro conjunto(s) de classes chamadas de ***constrainting class***.

Sua sintaxe é muito semelhante à sintaxe vista anteriormente para os *characteristic rules,* exceto pelo fato de que ela permite incluir definições sobre a *target class* ou sobre a(s) *constrainting class(es).* Após a cláusula ***FOR*** especifica-se o nome da *target class* e os objetos aos quais pertencem a *target class* têm que ser satisfeito na cláusula ***WHERE***. Por outro lado, o nome de cada *constrainting class* deve ser especificado após a palavra reservada **VERSUS** seguidos da condição para satisfazer os objetos da *constrainting class.*

Exemplo: *Realizar uma consulta comparando as lojas com lucros elevados e lojas com baixos lucros.*

MINE COMPARISON DESCRIBING "Stores"
ANALYSE sum(sales)
WITH RESPECT TO states_census.geo, statename, type
FROM sates_census
WHERE *Stores.geo* ***INSIDE*** *states_census.geo*
FOR *"Hight profit stores"*
    WHERE *profit_rate > 30*
VERSUS *"Low profit stores"*
    WHERE *profit_rate < 10*

No exemplo anterior, as lojas são comparadas em relação aos estados onde estão localizadas e os tipos de lojas.

2.4.3.3 Clustering Rules

São regras que estão associadas à clusterização (aglomeração) de pontos sob a descrição de determinado contexto espacial no processo de descoberta.

Em consultas de clusterização, a cláusula ***WITH RESPECT TO*** contém atributos que representam propriedades dos objetos a serem clusterizados.

Exemplo: Consulta de mineração com descrição de *clusters* loja.

MINE CLUSTERS DESCRIBING "Stores"
ANALYSE sum(sales)
WITH RESPECT TO Stores.geo, type
FROM Stores

Neste caso, a descoberta de clusters loja irá ser analisada de acordo com o tipo de loja e o total de vendas.

2.4.3.4 Spatial Association Rules

Como visto no capítulo anterior, uma regra de associação espacial é uma regra da forma:

$$X_1, X_2, ..., X_m \rightarrow Y_1, Y_2, ..., Y_n \ (s\%, c\%)$$

onde que ao menos um dos predicados $X_1, X_2, ..., X_m, Y_1, Y_2, ..., Y_n$ é um ***predicado espacial*** e *s%* representa o **suporte** da regra e *c%* a **confiança** da regra.

Em consultas realizadas sob este tipo de tarefa, o primeiro *spatial_term* especificado na cláusula ***WITH RESPECT TO*** representa a propriedade espacial do objeto a ser descrito.

Exemplo: *Descrever as lojas de acordo com seus lucros e a relação espacial em relação aos shoppings para com os seus clientes.*

MINE SPATIAL ASSOCIATION DESCRIBING "Stores"
WITH RESPECTIVE TO Stores.geo, Stores.profit,
CLOSE_TO(Stores.geo, Shoppings_centers.geo, "2km")
FROM Stores, Shopping_centers

2.4.3.5 Classification Rules

A classificação é uma técnica utilizada na mineração de dados sob a qual dados são classificados em grupos similares. De forma geral, a distinção entre cada grupo de dados é feita através de um *class label attribute*.

Exemplo: *Classificar lojas de acordo com seus tipos.*

MINE CLASSIFICATION DESCRIBING "Stores"
ANALYSE type
WITH RESPECT TO sales, profit,
CLOSE_TO(Stores.geo, Shoppings_centers.geo, "2km")
FROM Stores, Shopping_centers

Neste caso, as lojas serão classificadas de acordo com o seu tipo, lucro ou seu relacionamento com os shoppings centers.

2.4.4 Definições de limiares

A cláusula "***SET*** <threshold_specification>" especifica os vários tipos de medidas que podem ser utilizadas em uma tarefa de mineração de dados. Entre as medidas mais comuns encontram-se: suporte , confiança, ganho de entropia, convicção e o índice de gini vistos anteriormente na *Seção 2.2*.3.

Exemplo:

SET *support* ***THRESHOLD*** 0.5

Neste exemplo, serão extraídos na mineração resultados com um suporte mínimo de 50%.

2.4.5 Conceitos de hieraquização

Por fim, a cláusula *"<hierarchy_specification>"* especifica o tipo de hierarquia utilizado no processo de mineração. Em ***GMQL***, um conceito hierárquico pode ser construído baseado em um conceito de agrupamento, através do qual mostram grupos de conceitos em níveis distintos.

Exemplo:

DEFINE *{MA, PI, CE, RN, PB, PE, AL, SE, BA}*
UNDER *{Nordeste}* ***IN*** *regiao*
DEFINE *{Norte, Nordeste, Sul, Sudeste, Centro-Oeste}*
UNDER *{Brasil}* ***IN*** *regiao*

Segundo [Koperski 1999], um conceito hierárquico espacial pode ser definido baseado em um atributo não-espacial que é chave primária de uma tabela contendo atributos espaciais. Assim, a hierarquia conceitual *Brasil* pode ser construída baseada no atributo *região*, tal como:

DEFINE SPATIAL HIERARCHY *Brasil* ***BASED ON*** *regiao*

2.5 AMBIENTES DE CONSULTAS VISUAIS

Em virtude das dificuldades encontradas, por parte dos usuários de um sistema, em relação à formulação dos scripts de consultas sobre determinadas bases de dados, houve a necessidade de se criar um ambiente amigável para facilitar o trabalho do usuário, oferecendo mecanismos que permitam construir tais consultas através da combinação de símbolos pictográficos. Nesse contexto, em [Batini et al. 1991] foi desenvolvido o ***Visual Query Systems (VQS)*** especificando um formato visual de consulta, além de inserir várias funcionalidades no intuito de facilitar a iteração do usuário com o sistema.

Conforme [Catarci et al. 1997], os VQS (*ou Sistemas de Consultas Visuais – SCV*) são definidos como sistemas de consulta que utilizam representações visuais e regras gráficas para denotar o domínio dos dados e expressar os pedidos solicitados. De maneira geral, o *SCV* é um sistema que pode ser utilizado por diferentes tipos de usuários, na

maioria das vezes inexperientes, de forma que possibilitem realizar pesquisas em determinadas bases de dados, melhorando, por conseguinte, a interação homem-máquina.

2.5.1 Arquitetura de um Sistema de Consulta Visual

Conforme [Appel 2003], a arquitetura de construção de um *SCV* deve apresentar duas partes fundamentais: *ambiente de iteração com o usuário* e o *ambiente de implementação*.

O ***ambiente de iteração com o usuário*** está relacionado à maneira sob a qual o usuário irá visualizar e manipular as informações. Através de um esquema definido, o usuário poderá dispor de um modelo de dados no qual possa fazer referência a um determinado contexto ou ambiente de domínio.

O ***ambiente de implementação*** está envolvido com a forma pelas quais os dados (informações) são persistidos e manipulados internamente. Nesse caso, tal ambiente está relacionado com banco de dados e a linguagem de consulta a serem utilizados.

Essa separação dos ambientes tem uma grande importância em virtude da possibilidade de utilizar diferentes modelos de dados sem levar em conta aspectos de implementação. Além disso, esta arquitetura permite estabelecer dois ***módulos de consulta***: um ***interno*** referente à *linguagem de consulta* (*linguagem base*) estabelecida no ambiente de implementação e um ***externo*** relacionado à *linguagem gráfica*.

É de fundamental importância que exista uma ***camada de tradução*** entre os módulos de consulta interno e externo a qual estabeleça um *mapeamento* entre as operações e representações de cada módulo.

2.5.2 Importância de um SCV e seus principais aspectos

Segundo [Catarci e Santucci 1994], os VQS são caracterizados através dos seguintes aspectos:

- utilizam metáforas visuais (ícones, diagramas, etc) fazendo com que o usuário se sinta atraído pelo sistema;
- o usuário não necessita ter um conhecimento prévio da estrutura interna do banco, realizando as suas consultas de forma *ad-hoc*;
- fazem uso de mecanismos iterativos de forma que facilitem a formulação de consultas através da manipulação direta de elementos naturais e intuitivos.

Além disso, conforme [Soares 2002], os fatores mais importantes que influenciam o uso efetivo de *VQS*, do ponto de vista do usuário, são:

- modelos adaptados para descrever os dados e as operações de consulta;
- representação escolhida para mostrar uma instância do modelo;
- estratégias de interação sugerindo como o sistema deve ser usado para formular uma determinada consulta.

Ao se formular uma consulta visual é imprescindível ter uma compreensão do domínio de interesse de forma que se venha a facilitar o entendimento dos diferentes tipos de conhecimento que podem estar armazenados em determinados bancos de dados [Catarci et al. 1997].

2.5.3 Classificações dos Sistemas de Consultas Visuais

Existem diversas classificações relativas aos *Sistemas de Consultas Visuais*, relacionados ao tipo de formalismo visual utilizado na elaboração e construção de uma consulta. Este formalismo constitui a definição de elementos visuais e as formas de interação entre eles e pode ser classificado em quatro paradigmas:

- ***Linguagem Baseada em Formulários*** - de maneira geral, utilizam protótipos visuais de tabelas para serem preenchidas pelos usuários ao formularem suas consultas [Zloof 1977].
- ***Linguagem Baseada em Diagramas*** – utiliza um conjunto limitado de símbolos geométricos associados a um conceito e de ligações que representam os relacionamentos entre eles [Catarci e Santucci 1994].
- ***Linguagem Baseada em Ícones*** – um sistema baseado em ícones utiliza um conjunto de símbolos que denotam entidades do mundo real e algumas funções disponíveis no sistema [Soares 2002].
- ***Linguagem Híbrida*** – tem o intuito de combinar qualquer tipo de formalismo visual visto anteriormente, possibilitando diversas alternativas para a representação do banco e da consulta.

2.6 SISTEMAS DE CONSULTAS VISUAIS PARA A MINERAÇÃO DE DADOS GEOGRÁFICOS

Em virtude das dificuldades encontradas para se formular consultas nas linguagens de mineração, em especial sobre processos de mineração de dados geográficos, e do fato desses dados apresentarem características inerentemente visuais, este trabalho desenvolve um Sistema de Consulta Visual o qual provê, internamente, uma Linguagem Gráfica para a Mineração de Dados Geográficos.

A arquitetura deste sistema foi construída e fundamentada nos padrões dos ambientes de consultas visuais vistos anteriormente. Além disso, regras gráficas e representações visuais sobre os metadados de domínios geográficos foram definidas a fim de facilitar a iteração do usuário com o sistema.

A linguagem gráfica utilizada neste ambiente é baseada em ícones gráficos de modo que um conjunto de símbolos é definido a fim de representar feições geográficas e funções espaciais.

2.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo teve como objetivo abordar e introduzir os principais conceitos e tecnologias utilizadas nos processos de mineração de dados geográficos.

Inicialmente, foram apresentadas as principais características a cerca dos bancos de dados geográficos, dando um enfoque importante sobre os relacionamentos espaciais existentes entre entidades geográficas, uma vez que tais relacionamentos constituem o principal foco dos processos de mineração sobre dados geográficos. Além disso, foram vistos algumas formas, através das quais, os dados geográficos podem ser estruturados, além de um aprofundamento sobre os conceitos de possíveis dependências geográficas que podem existir entre essas entidades e que assumem uma grande importância em estudos atuais, uma vez que elas podem ser utilizadas para acelerar todo o processo de mineração.

Em seguida, foi apresentado o conceito do processo de descoberta de conhecimento em banco de dados. Neste caso, foram abordadas as etapas (fases) que constituem todo um processo de descoberta, dando maior ênfase nas tarefas de regra de associação, em virtude de tais tarefas contribuírem com importantes extrações de

conhecimentos, quando comparado a outras tarefas, nos processos de mineração de dados geográficos.

Por outro lado, ao se definir ou projetar a construção de uma nova linguagem de consulta, é imprescindível que ela seja baseada em certas primitivas básicas já padronizadas. Com isso, foram apresentados os principais conceitos a cerca das primitivas básicas de construção de uma linguagem de consulta para mineração de dados. Em seguida, tais considerações foram abordadas especificamente para os dados geográficos.

Por fim, foram abordados as principais características que definem a construção de ambientes visuais de consultas, apresentando sua arquitetura básica, seus principais aspectos e importância perante o uso efetivo desses sistemas por parte do usuário, além as suas classificações.

No próximo capítulo, serão mostrados alguns trabalhos existentes na literatura os quais possuem alguma relação com o trabalho proposto nesta dissertação.

CAPÍTULO 3

TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos existentes na literatura que estão envolvidos e relacionados com os conceitos básicos da definição do tema proposto nesta pesquisa. Vale ressaltar que algumas dessas propostas são estritamente acadêmicas, não possuindo produtos comerciais no mercado.

3.1 INTRODUÇÃO

Existem, na literatura, diversos métodos propostos para a descoberta de conhecimento em bancos de dados. Muitos destes métodos são implementados em linguagens de consulta, tais como a ***DMQL*** [Han et al. 1996], ***RDM*** [Raedt 2000] e a ***ST-DMQL*** [Bogorny et. al. 2009].

Em relação aos domínios geográficos, especificamente, esses métodos de mineração podem ser utilizados por outras linguagens tais como a **GMQL** ***(Geo-Miner Query Language)*** [Han, Koperski e Stefanvic 1997], **LARECOS** [Biglin e Marsala 1998] e a **SDMOQL** ***(Spatial Data Mining Object Query Language)*** [Malerba, Appice and Vacca 2002].

Por outro lado, a maioria dessas linguagens possui apenas especificações formais. Com isso, poucas abordagens criaram alguns protótipos como é o caso do **GeoMiner** [Han, Koperski e Stefanvic 1997], o ***VisMiner*** [Bimonte et al. 2003] **Ares** [Appice et al. 2005] e o **IGENS** [Malerba et al. 2003].

Quanto ao processo de utilizar conhecimentos prévios como parte de execução dos algoritmos de mineração sobre dados geográficos, em [Bogonry 2006] é proposto um *framework* no qual utiliza *esquemas* de Bancos de Dados e *ontologias* como alternativas para expressar tais conhecimentos. Todavia, este trabalho não utiliza estes conceitos nas próprias formulações de consultas para mineração de dados.

Em relação a ambientes de consultas visuais existentes na literatura, alguns deles possuem certa relevância para a elaboração deste trabalho, entre eles: ***GeoMiner*** [Han,

Koperski e Stefanvic 1997], ***XQBE*** (XML Query By Example) [Kade 2001] ***GeoVisual*** [Soares 2002], ***SVQL*** [Souza 2008] e o ***VisMiner*** [Bimonte et al. 2003] .

3.2 GEOMINER

O ***GeoMiner*** é um protótipo para sistemas de mineração de dados espaciais desenvolvido por [Han, Koperski e Stenfavic 1997], sendo uma extensão do ***DBMiner*** (ferramenta para descoberta de conhecimento em bancos de dados convencionais) [DBMiner Technology Inc. (2000)].

A consulta as dados no *GeoMiner* é realizada através da linguagem *GMQL* vista anteriormente.

3.2.1 Arquitetura

A arquitetura do ***GeoMiner*** foi baseada na arquitetura do ***DBMiner*** e possui 3 grandes camadas *(*Figura 3.1*)*:

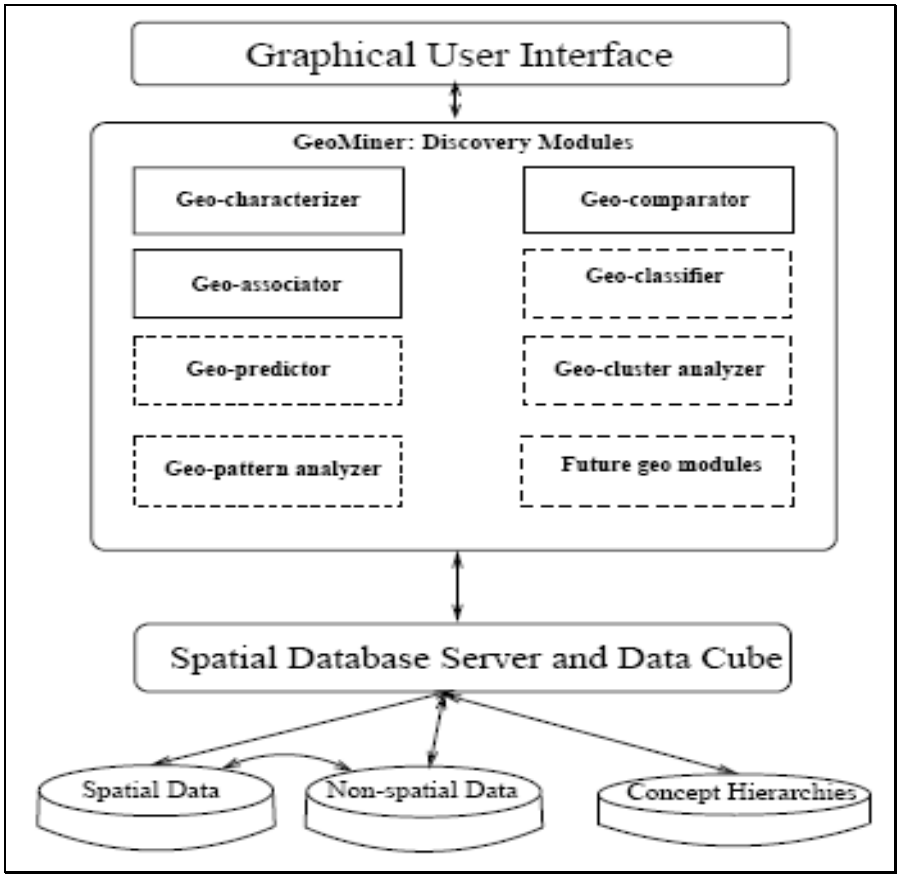


Figura 3.1: Arquitetura Geral do GeoMiner [Han, Koperski e Stefanvic 1997].

- ***Camada de Interface Gráfica*** – sob a qual permite o usuário realizar de forma amigável a manipulação com o sistema, além de possibilitar a exibição dos resultados de uma mineração;

- ***Camada de Descoberta de Conhecimento*** – através do qual é dividida em 5 módulos: *Geo-Characterizer,* o *Geo-Comparator,* o *Geo-Associator, o Geo-Classifier e o Geo-Cluster-Analyser.* Esses módulos são responsáveis por extrair conhecimento através da linguagem GMQL. É importante ressaltar que, cada tarefa de mineração, a GMQL oferece comandos específicos.

- ***Camada de Acesso e processamento aos dados espaciais*** – responsável pelo acesso interno às estruturas físicas de armazenamento.

De maneira geral, a arquitetura do ***GeoMiner*** foi desenvolvida sobre a base da arquitetura do ***DBMiner*** de tal modo que as funções de mineração são direcionadas para o *DBMiner*. O usuário, através da *interface do sistema*, pode realizar tarefas de mineração com o uso de *tabelas*, *mapas* ou *gráficos* (Figura 3.2).

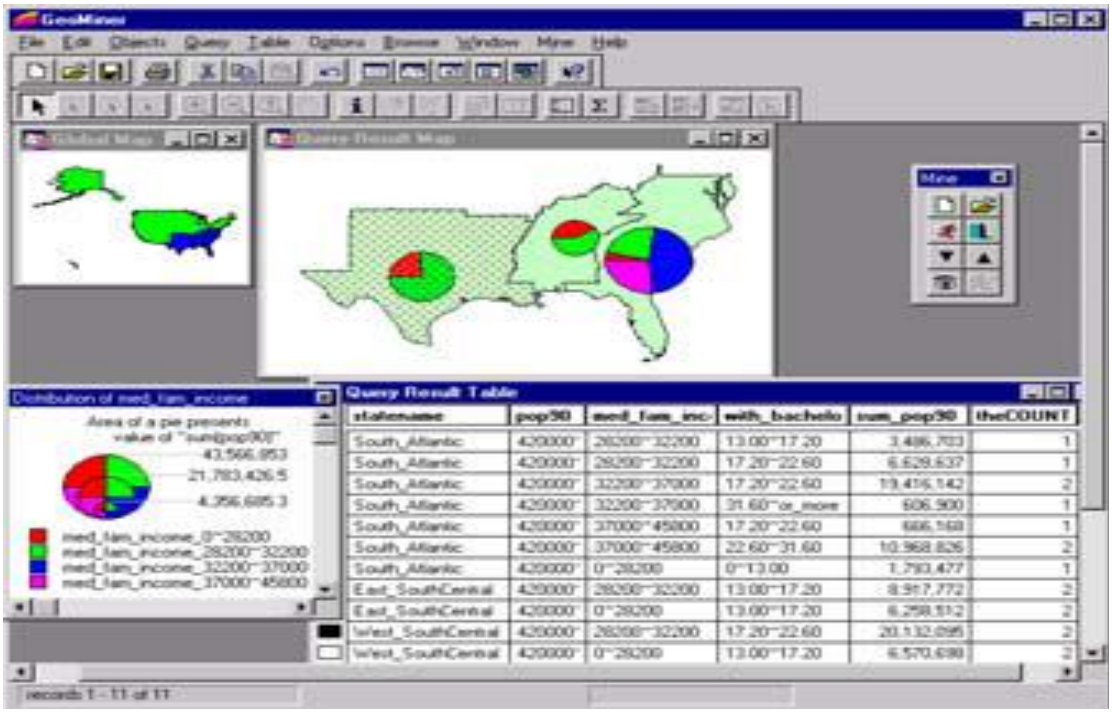


Figura 3.2: Interface do GeoMiner.

3.3 XQBE

A ***XQBE (XML Query By Example)*** é uma linguagem que envolve a utilização de um *modelo conceitual* para representar os *conceitos* e as *relações entre conceitos* que ocorrem em documentos XML pertencentes a um determinado domínio de problema. O modelo conceitual é representado por uma ontologia do domínio do problema [Kade 2001].

Neste trabalho, as consultas podem ser definidas tendo-se como base conceitos de ontologias. Além disso, existem regras de mapeamento no qual se permitem estabelecer compatibilidades entre documentos XML com determinadas ontologias.

Em resumo, segundo [Kade 2001], A ***XQBE*** é uma linguagem visual para consultas a documentos XML com base em ontologias, apresentando uma proposta de interface visual para esta linguagem. Esta interface (Figura 3.3) apresenta dois elementos principais: a *janela de ontologias* (a qual se baseia na representação gráfica de uma ontologia) e a *janela de consulta* (sob a qual se constrói um esquema XML relativo à visão de ontologia definida na *janela de ontologias*). Além disso, podem ser construídas diversas janelas de consultas para uma mesma ontologia.

De forma geral, conforme [Kade 2001], para realizar uma consulta o usuário deve:

1. Carregar uma ontologia. Com isso, o sistema mostra os elementos e associações entre os elementos da ontologia;

2. Criar uma ou mais janelas de consultas;

3. Arrastar elementos da ontologia para uma janela de consulta.

4. Realizar uma filtragem com os elementos de uma consulta.

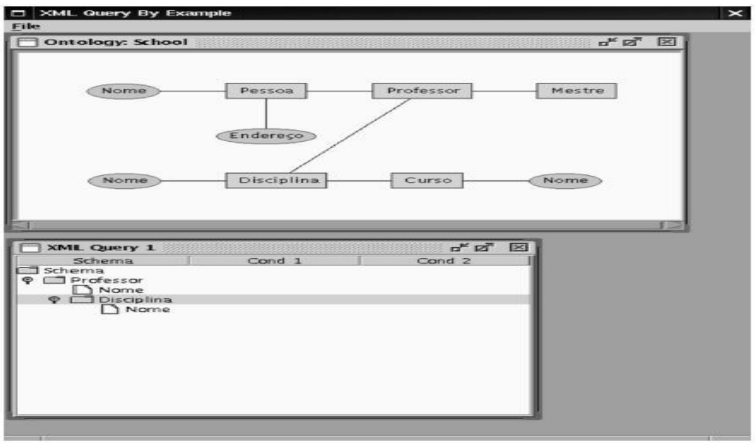


Figura 3.3: Visualização da interface mostrando as janelas de ontologia e de consulta [Kade 2001].

3.4 GEOVISUALQL

A ***GeoVisualQL*** [Soares 2002] é uma *Linguagem de Consulta Visual Geográfica* que, além de sua *especificação em SQL*, utiliza, em sua concepção, o modelo de dados do consórcio ***OpenGIS*** definido em [Open Gis Consortium 1999]. Tal linguagem surgiu em detrimento dos dados geográficos serem inerentemente visuais e poderem ser representados de forma gráfica (geométrica). Assim, a *GeoVisualQL* possibilita ao usuário formular sua consulta e receber o resultado da mesma através de símbolos visuais.

Em sua camada interna, a *GeoVisualQL* possui um *módulo de tradução* que realiza a conversão entre uma consulta feita por *símbolos gráficos* em uma consulta baseada na *especificação SQL do OpenGIS.*

Além disso, [Soares 02] desenvolveu um protótipo de um *Ambiente de Consultas Visuais* para *Sistemas de Informações Geográficas (SIGs),* denominado ***GeoVisual*** (Figura 3.4) baseado na sintaxe definida para a linguagem *GeoVisual*QL. Uma importante contribuição deste trabalho é que ele visa integrar duas diferentes áreas de pesquisa: o paradigma de consultas visuais e o uso de padrões de metadados espaciais.

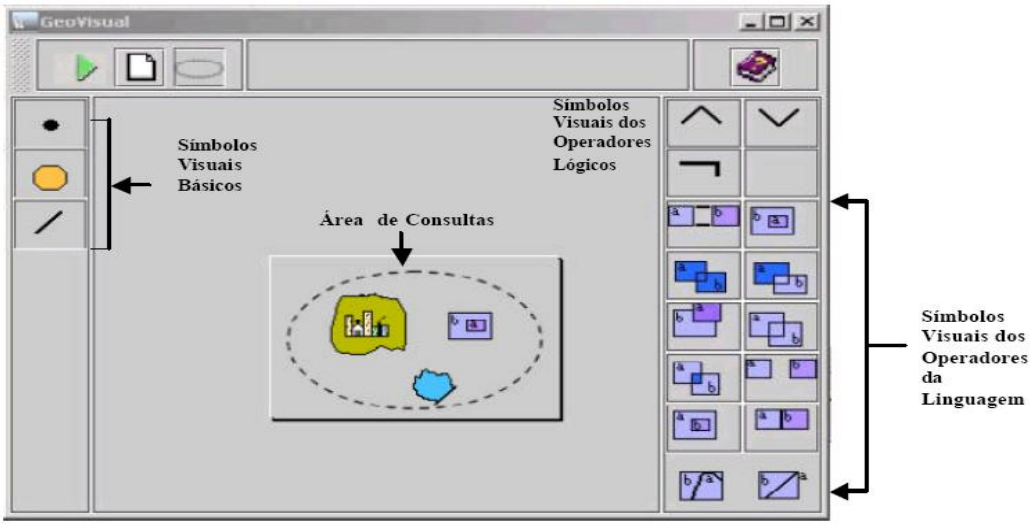


Figura 3.4: Janela Principal do GeoVisual.

No exemplo da Figura 3.5, o usuário deseja recuperar os municípios que sobrepõem a *Bacia de Curimataú*. Neste caso, o usuário seleciona os *símbolos visuais* da interface que melhor represente as entidades geográficas presentes na consulta. Com isso, os polígonos da consulta indicam *feições geométricas* com este tipo de geometria. Após definir uma instância dessas entidades como a *Bacia de Curimataú*, o usuário insere o

símbolo de *sobreposição (overlap)* representando a operação espacial a ser utilizada no relacionamento entre as entidades. Para isto, basta apenas o usuário definir a outra entidade como município. O resultado da consulta é visto na Figura 3.6.

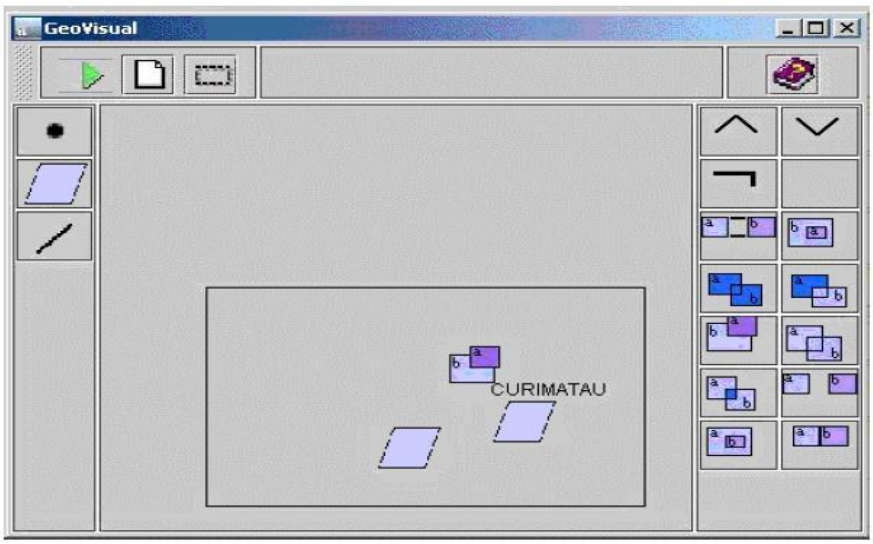


Figura 3.5: Exemplo de uma consulta no GeoVisual pelos municípios que sobrepõem a Bacia de Curimataú.

Figura 3.6: Resultado da consulta feita na figura 3.5.

3.5 SVQL

A ***SVQL*** *(SOLAP Visual Query Language)*, definida por [Souza 2008], é uma linguagem que visa auxiliar o processo de tomada de decisão através da integração entre os Sistemas de Informações Geográficas (SIGs) sob múltiplas perspectivas, através de ferramentas **OLAP** *(On-line Analytical Processing)*. Esta integração é mais conhecida como **SOLAP** *(Spatial OLAP)* [Bédard et al. 1997]. Em virtude da complexidade existente entre os operadores espaciais e analítico-multidimensionais, a união entre eles torna-se muito difícil, principalmente no contexto sintático e semântico. Assim, a linguagem *SVQL*

foi criada no intuito de facilitar as consultas sobre sistemas *SOLAP*, principalmente na recuperação de dados geográficos multidimensionais.

A validação e tradução das consultas são realizadas através do sistema ***SVQL Composer*** (Figura 3.7).

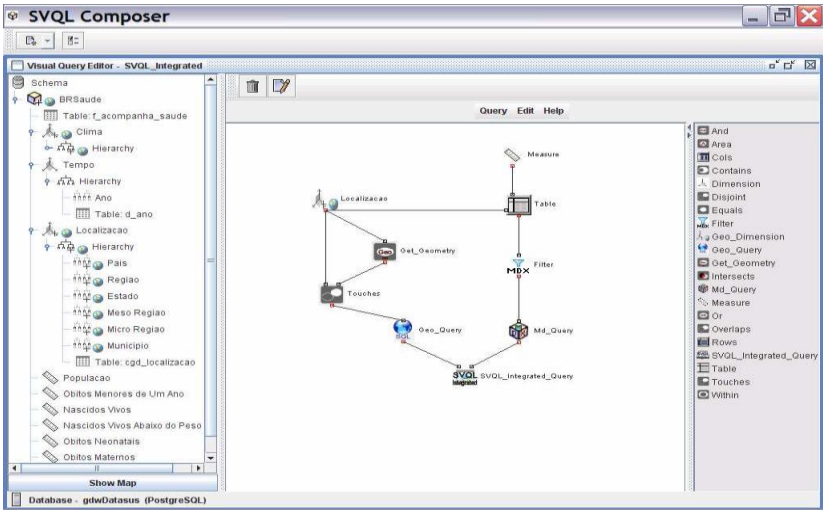


Figura 3.7: Exemplo de uma consulta visual feita no SVQL Composer.

De modo geral, segundo [Souza 2008], para se realizar uma consulta o usuário deverá tomar os seguintes passos:

1. *Determinar a base de dados e Escolher o tipo de consulta SVQL*: etapa responsável por manter a conexão com o banco e a configuração XML do esquema definido;
2. Conhecer o domínio de interesse através do *esquema visual*;
3. Instanciar as feições espaciais;
4. Conectar as instâncias com base na gramática da linguagem;
5. Parametrizar as instâncias;
6. Validar e traduzir a consulta.

3.6 VISMINER

O **VisMiner** é um protótipo desenvolvido por [Bimonte et al. 2003] de uma linguagem de mineração de dados espaciais, baseado em uma metáfora conhecida como "*Miner Trip*" na qual configura as tarefas de mineração através de ações e elementos utilizados pelo usuário (minerador) e as relações espaciais utilizadas nos algoritmos de

mineração, criando diferentes modelos de dados, além de explorar as ambigüidades topológicas existentes entre estes modelos.

A linguagem VisMiner utiliza conceitos hierárquicos uma vez que cada ícone em uma sentença visual, por ela construída, pode ser mais detalhada através de uma sentença visual definida pela própria linguagem ou de uma outra linguagem. Neste contexto, os níveis hierárquicos realizam todo o controle semântico, uma vez que organizam os elementos envolvidos em uma tarefa de mineração. Uma vez que ocorre um erro semântico, o VisMiner é capaz de detectá-lo propondo, por conseguinte, uma nova sentença visual . Neste caso, o controle semântico é executado por um algoritmo baseado no dado associado a cada elemento visual.

A metáfora "*Miner Trip"* organiza os elementos em duas diferentes categorias: os *elementos primitivos básicos* (que representam os elementos reais envolvidos na tarefa) e os *elementos primitivos* (usados para definir hierarquicamente a estrutura de uma sentença visual). Tal metáfora combina uma transcrição visual do tipo "arquivo/pasta", apresentadas sobre uma abordagem icônica, no qual se subdivide em 3 categorias: *SpatialMetaphor, DirNumMetaphor e NumMetaphor*. Estas categorias estão associadas às representações icônicas e os componentes espaciais envolvidos por algum domínio em análise.

As formulações de consultas feitas pelo protótipo que utiliza o VisMiner leva em consideração a representações de visualização. A representação da forma visual e a representação por redução icônica. A primeira dá uma visão geral de uma relação espacial, mostrando os relacionamentos que podem ser entendidos e interpretados por diferentes tipos de usuários. Já a segunda representação faz uma redução da visão geral para um elemento icônico que contém elementos ou relações utilizados pelo processo de mineração (Figura 3.8).

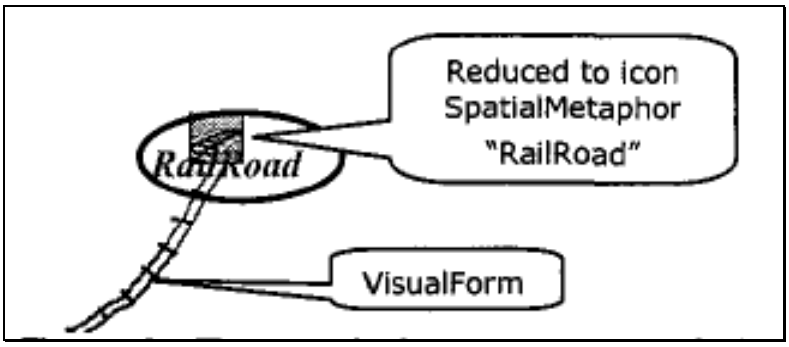


Figura 3.8: Exemplo de uma representação metafórica "Rail/Road" [Bimonte et al. 2003].

3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Os trabalhos relacionados apresentados neste capítulo ressaltam a importância deste estudo uma vez que procura unificar diversas áreas de pesquisas através de uma interface visual para descoberta de conhecimento em ambientes geográficos.

Nesse contexto, o ambiente ***GeoMiningVisual*** se destaca, principalmente, por apresentar, internamente, a unificação de linguagem de consulta visual, uma linguagem de mineração de dados e a abordagem de contextos geográficos. Além disso, utiliza informações de metadados em sua tradução, representando todo o esquema do ambiente geográfico e operadores da própria linguagem de forma visual.

No capítulo seguinte, será apresentada e definida a linguagem ***GeoMiningVisualQL***. Mais adiante, será visto como esta linguagem está vinculada ao Ambiente de Consulta Visual denominado ***GeoMiningVisual***.

CAPÍTULO 4

GeoMiningVisualQL – Linguagem de Consulta para Mineração de Dados Geográficos

Este capítulo apresenta a linguagem GeoMiningVisualQL, uma Linguagem de Consulta Visual sob Processos de Mineração de Dados Geográficos. A princípio, será feita a especificação e a definição formal da linguagem. Em seguida, será realizada toda a representação visual de ações e operadores presentes nas tarefas de mineração de dados geográficos, além da construção de todo formalismo gramatical da linguagem.

4.1 INTRODUÇÃO

Apesar de existirem na literatura algumas linguagens de consulta sobre mineração de dados (discutidas na Seção 3.1 deste trabalho), elas possuem a desvantagem de não apresentarem, em sua filosofia, nenhuma interface gráfica que facilite o uso e a manipulação das consultas por parte do usuário. Sendo assim, neste trabalho a *Linguagem Visual para Mineração de Dados Geográfico*, denominada ***GeoMiningVisualQL (Geographic Mining Visual Query Language)***, foi definida e especificada, concedendo aos usuários a possibilidade de realizarem suas consultas através de construções gráficas (pictóricas) sobre os elementos que constituem as primitivas da linguagem *GMQL* (*Geo Mining Query Language*) [Koperski 1999], fazendo uso de símbolos visuais.

A linguagem GMQL é uma extensão da linguagem DMQL (Data Mining Query Language) sendo utilizada, neste trabalho, como linguagem base em virtude da linguagem DMQL ter sido considerada, segundo [Kuijpers et al. 2008], uma transição para a utilização padrão sobre as linguagens de consultas realizadas em processos de mineração de dados. A DMQL também permite realizar a mineração *ad-hoc* sobre todos os tipos de conhecimentos que podem estar embutidos sobre determinados tipos de bancos de dados.

Outro fato importante em relação à adoção da GMQL como linguagem base, neste trabalho, está relacionado ao fato de tal linguagem poder fazer uso do conceito de

hierarquias durante a mineração de dados geográficos em diferentes níveis de granularidades, tornando o processo de mineração mais eficiente.

A linguagem *GeoMiningVisualQL* foi projetada tendo a sua definição formal associada à construção de uma gramática baseada nos padrões de linguagens de consultas visuais, além de ser definida a partir de especificações sintáticas e semânticas da linguagem GMQL, uma vez que as regras sintáticas (e também semânticas) da linguagem base devem ser obedecidas na linguagem visual (Apêndice B).

Por sua vez, a linguagem *GeoMiningVisualQL* está definida através de símbolos visuais que representam tarefas (ações) de mineração de dados, operadores existentes na linguagem GMQL, além das feições geográficas existentes em um determinado domínio geográfico.

Além disso, uma consulta formulada através da linguagem *GeoMiningVisualQL* é completamente visual uma vez que todos os seus elementos estão associados a uma representação gráfica.

4.2 ESPECIFICAÇÃO DA LINGUAGEM

Algumas considerações em relação aos operadores, representações pictóricas, e funções não-espaciais envolvidas serão essenciais na especificação ou definição formal da linguagem ***GeoMiningVisualQL***. Além disso, é importante ressaltar que a definição formal desta linguagem, a princípio, restringe-se a tarefas de regras de associação espaciais existentes entre feições geográficas.

Quanto aos operadores, eles podem ser divididos em espaciais e não-espaciais. Os operadores espaciais são classificados como predicados espaciais e os operadores não-espaciais são subdivididos em lógicos, relacionais e aritméticos.

Os operadores espaciais são utilizados no momento em que o usuário seleciona os termos espaciais que fazem parte da consulta e, por sua vez, são subdivididos em: *direcionais*, *topológicos* e de *distância*.

Quanto aos operadores direcionais, eles podem ser: ***west_from***, ***east_from***, ***south_from*** e ***north_from***;

A lista dos operadores topológicos disponíveis na linguagem são: ***contains***, ***within***, ***intersects***, ***neighbor***, ***equal***, ***under*** e ***over***.

As funções não-espaciais são subdividas em: ***sin***, ***cos***, ***tan***, ***log***, ***atn***, ***acos***, ***asin***, ***month***, ***year***, ***day***, ***area***, ***length***, ***min***, ***max***, ***centroid***, ***sum***, ***avg***, ***count(*)*** e ***wgavg***.

As funções espaciais são classificadas em duas categorias:

- ***boundary*** e suas subdivisões (***edges***, ***nodes*** e ***polygons***);
- ***interior*** e suas subdivisões (***edges***, ***nodes*** e ***polygons***).

Os *operadores booleanos* são classificados em: ***not***, ***and*** e ***or***.

Os *operadores relacionais* estão divididos em: =, <, <=, >, >=, <> e ***like***.

Já os *operadores aritméticos* são decompostos em: +, -, ***** e /.

Vale ressaltar que determinados elementos e operadores podem ser representados através de um ***símbolo visual (sv).***

Por outro lado, não há a preocupação em se definir representações geométricas dos resultados da consulta. Isto acontece em virtude do ambiente GeoMiningVisual, sob o qual a linguagem está incorporada, se preocupar apenas na formulação de uma sentença GMQL. Versões futuras deste sistema irão procurar realizar a vinculação com algum outro sistema, a fim de se ter uma execução completa de um algoritmo de mineração através da interface do ambiente ao qual a linguagem GeoMiningVisualQL está inserida. Neste caso, os usuários poderão obter diferentes perspectivas de visualização dos resultados colhidos.

Além disso, como relata [Bogorny e Alvares 2005], é difícil utilizar extensões da linguagem GMQL uma vez que existem apenas softwares com fins comerciais, tal como o *GeoMiner,* tendo, portanto, o seu código-fonte não aberto.

4.3 DEFINIÇÃO FORMAL DA LINGUAGEM

Definição 1

Seja a linguagem de consultas visuais sob tarefas executadas em processos de mineração de dados geográficos definida como ***GeoMiningVisualQL = (P, O, R)****, onde P é o conjunto de objetos pictóricos da linguagem, O é o conjunto de operadores permitidos e R é o conjunto de restrições sobre estes operadores.*

Definição 2

Uma feição geográfica em GeoMiningVisualQL é uma instância ***fg = (i, sv)*** *onde* ***i*** *é o valor da instância de* ***fg*** *e* ***sv*** *é o símbolo visual da feição.*

Definição 3

Um objeto pictórico em GeoMiningVisualQL é uma tupla ***p=(sv, fg, i)*** *onde* ***sv*** *é o símbolo visual do objeto,* ***fg*** *é a feição geográfica representada, e* ***i*** *(eventualmente vazia) é o valor da instância de* ***fg****.*

Definição 4

Um operador não-espacial em GeoMiningVisualQL é uma tupla ***op_ne =(as1, as2, sv)*** *onde* ***as1*** *e* ***as2*** *representam atributos simples (não-espaciais) de feições geográficas de um domínio e* ***sv*** *representa o símbolo visual do operador.. Caso o operador não-espacial seja aplicado a um único atributo, um dos outros atributos é nulo.*

Definição 5

Um operador espacial em GeoMiningVisualQL é uma tupla ***op_e =(ae1, ae2, sv)*** *onde* ***ae1*** *e* ***ae2*** *representam atributos espaciais de feições geográficas de um domínio e* ***sv*** *representa o símbolo visual do operador. Caso o operador espacial seja aplicado a um único atributo espacial, um dos outros atributos é nulo.*

Definição 6

Um operador em GeoMiningVisualQL é uma tupla ***o=(to, sv)*** *onde* ***to*** *representa o tipo do operador espacial, podendo ser um operador não-espacial* ***op_ne*** *ou um operador espacial* ***op_e****, e* ***sv*** *é o símbolo visual do operador.*

Definição 7

Um relacionamento espacial direcional em GeoMiningVisualQL é uma tripla ***rd = (op_d, fg1, fg2),*** *onde* ***op_d*** *um operador direcional e* ***fg1*** *e* ***fg2*** *são feições geográficas.*

Definição 8

Um relacionamento espacial topológico em GeoMiningVisualQL é uma tripla ***rt = (op_t, fg1, fg2)****, onde* ***op_t*** *um operador topológico e* ***fg1*** *e* ***fg2*** *são feições geográficas.*

Definição 9

Um relacionamento lógico em GeoMiningVisualQL é uma tripla ***rl = (op-l, rs1, rs2),*** *onde* ***op_l*** *é um operador lógico e* ***rs1*** *e* ***rs2*** *são relacionamentos espaciais topológicos ou relacionamentos lógicos. No caso do operador lógico* ***not*** *um dos relacionamentos da tripla* ***rl*** *pode ser vazio.*

Definição 10

*Uma **consulta visual cv** em GeoMiningVisualQL é representada por uma figura (árvore metafórica) que contém um agrupamento de símbolos visuais que ilustram uma tarefa de mineração sobre dados geográficos **t** envolvendo um relacionamento espacial topológico **rt**, um relacionamento espacial direcional **rd** ou um relacionamento lógico **rl**.*

Definição 11

*Um **função espacial fe** é uma tripla **fe = (op_e, fg1, fge2)**, onde **op_e** é um operador espacial, e, **fg1** e **fg2** são feições geográficas pertencentes a um domínio geográfico. No caso de **o** ser um operador lógico **not**, uma das feições geográficas da tripla **fe** pode ser não-existente.*

Definição 12

*Um **predicado espacial pe** é um subconjutno de uma **região geográfica**, sendo uma tripla **pe = (fe, fg1, fg2)** o qual resulta numa restrição espacial ocasionado por uma função espacial **fe** sobre feições geográficas **fg1** e **fg2**. Caso esta função seja unária, uma das feições geográficas da tripla **pe** pode ser não-existente.*

4.4 DEFINIÇÃO DAS REPRESENTAÇÕES PICTÓRICAS

4.4.1 Representação Geral

Tabela 4.1: Representação pictórica geral.

Conceito	*Descrição*
Símbolo Visual	*Tradução*
	Sentença traduzida

4.4.2 Ações a serem exercidas

Tabela 4.2: Representação de ações a serem exercidas no ambiente.

GMQL_Query	*Especifica a escolha por uma consulta sob tarefas de mineração de dados geográficos.*
	*Tradução*
	MINE

4.4.3 Manipulação Hierárquica

Tabela 4.3: Representação do uso de níveis hierárquicos.

Hierachy_Definition	*Especifica a escolha por uma definição hierárquica.*
	*Tradução*
	***DEFINE** '{' attribute {, attribute '{' attribute } '}' **UNDER** …*

4.4.4 Tipos de tarefas de mineração

Tabela 4.4: Representação de tarefas de regras de associação entre os dados geográficos.

Regra de Associação	*Especifica uma regra de associação como tipo particular de tarefa de mineração sobre os dados geográficos.*	
	*Tradução*	
	Sem literal	***SPATIAL ASSOCIATION WITH RESPECT TO***
	Com literal	***SPATIAL ASSOCIATION [DESCRIBING** literal **] WITH RESPECT TO***

Tabela 4.5: Representação de tarefas de clusterização entre os dados geográficos.

Clusterização	*Especifica uma clusterização como tipo particular de tarefa de mineração sobre os dados geográficos.*	
	*Tradução*	
	Sem literal	***CLUSTERS [ANALYSE …***
	Com literal	***CLUSTERS [DESCRIBING** literal **] [ANALYSE …***

Tabela 4.6: Representação de tarefas de classificação entre os dados geográficos.

Classificação	*Especifica uma classificação como tipo particular de tarefa de mineração sobre os dados geográficos.*	
	*Tradução*	
	Sem literal	***CLASSIFICATION [ANALYSE …***
	Com literal	***CLASSIFICATION [DESCRIBING*** *literal* ***] [ANALYSE …***

Tabela 4.7: Representação de tarefas de caracterização entre os dados geográficos.

Caracterização	*Especifica uma caracterização como tipo particular de tarefa de mineração sobre os dados geográficos.*	
	*Tradução*	
	Sem literal	***CHARACTERISTICS [ANALYSE …***
	Com literal	***CHARACTERISTICS [DESCRIBING*** *literal* ***] [ANALYSE …***

Tabela 4.8: Representação de tarefas de comparação entre os dados geográficos.

Outlier	*Especifica um outlier como tipo particular de tarefa de mineração sobre os dados geográficos.*	
	*Tradução*	
	Sem literal	***MINE COMPARISON [ANALYSE …***
	Com literal	***MINE COMPARISON [DESCRIBING*** *literal* ***] [ANALYSE …***

4.4.5 Definição de termos espaciais ou não-espaciais no ambiente

Tabela 4.9: Representação do uso de um termo não-espacial no ambiente.

Termo Não-Espacial	*Especifica um termo não-espacial (atrbutos ou funções simples) a ser usado no ambiente.*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

Tabela 4.10: Representação do uso de um termo espacial no ambiente.

Termo Espacial	*Especifica um termo espacial, isto é, um predicado espacial ou uma função geográfica a ser utilizada.*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

4.4.6 Definição geral de uma feição geográfica

Tabela 4.11: Representação do uso de uma feição geográfica qualquer.

Feição Geográfica	*Especifica uma feição geográfica qualquer pertencente a base de dados geográficas do ambiente analisado*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

4.4.7 Definição geral de um atributo de uma feição geográfica

Tabela 4.12: Representação geral de um atributo simples da feição geográfica do uso de uma feição geográfica.

Atributo Simples	*Especifica uma atributo simples de uma determinada feição geográfica pertencente a base de dados geográficas do ambiente analisado.*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

Tabela 4.13: Representação geral de um atributo simples da feição geográfica do uso de uma feição geográfica.

Atributo Geográfico	*Especifica uma atributo geográfico de uma determinada feição geográfica pertencente a base de dados geográficas do ambiente analisado.*
	*Tradução*
	atributoSelecionado.***geo***

4.4.8 Definição genérica de funções não-espaciais

Tabela 4.14: Representação do uso de funções simples (não-espaciais) no ambiente.

Definição genérica do uso de uma função simples (não-espacial)	*Especifica uma função não-espacial aplicada sobre atributos não-espaciais de entidades do ambiente analisado.*
	***Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.***

4.4.9 Definição de funções simples específicas

Tabela 4.15: Representação do uso de uma função que calcula a média dos valores de um atributo específico.

Função AVG	*Especifica uma função que realiza a média sobre os valores de um determinado atributo da entidade.*
	*Tradução*
	AVG(*relationName.nameAttributte****)***

Tabela 4.16: Representação do uso de uma função que calcula a soma dos valores de um atributo específico.

Função SOMA	*Especifica uma função que realiza a soma de valores contidos em um determinado atributo da entidade.*
	*Tradução*
	SUM(*relationName.nameAttributte****)***

Tabela 4.17: Representação do uso de uma função que retorna o valor mínimo de um atributo específico.

Função Mínimo	*Especifica uma função que retorna o valor mínimo de um determinado atributo da entidade.*
	*Tradução*
	MIN(*relationName.nameAttributte****)***

Tabela 4.18: Representação do uso de uma função que retorna o valor máximo de um atributo específico.

Função Máximo	*Especifica uma função que retorna o valor máximo de um determinado atributo da entidade.*
	*Tradução*
	MAX(*relationName.nameAttributte****)***

Tabela 4.19: Representação do uso de uma função conta os registros retornados por uma consulta.

Função Contador	*Especifica uma função que conta o número de registros retornados por uma consulta.*
	*Tradução*
	COUNT(*)

4.4.10 Definição genérica de funções espaciais

Tabela 4.20: Representação do uso de funções espaciais no ambiente.

Definição genérica do uso de uma função espacial	*Especifica um termo como sendo uma função espacial utilizado junto a uma feição geográfica.*
	***Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.***

4.4.11 Definição de funções espaciais específicas

Tabela 4.21: Representação do uso de uma função espacial delimitada externamente por uma fronteira simples ou contínua.

Boundary	*Especifica uma relação geográfica limitada ao exterior por uma fronteira contínua.*
	*Tradução*
	BOUNDARY(*atributoSelecionado****.geo)***

Tabela 4.22: Representação do uso de uma função espacial delimitada externamente por pontos contidos em uma fronteira simples ou contínua.

BoundaryNodes	*Especifica uma relação geográfica limitada ao exterior por pontos contidos em uma fronteira contínua.*
	*Tradução*
	BOUNDARY_NODES(*atributoSelecionado****.geo)***

Tabela 4.23: Representação do uso de uma função espacial delimitada externamente por segmentos de uma fronteira simples ou contínua.

BoundaryEdges	*Especifica uma relação geográfica limitada ao exterior por segmentos contidos em uma fronteira contínua.*
	*Tradução*
	BOUNDARY_EDGES(*atributoSelecionado****.geo)***

Tabela 4.24: Representação do uso de uma função espacial delimitada externamente por polígonos formados em torno de uma fronteira simples ou contínua.

BoundaryPolygons	*Especifica uma relação geográfica limitada por polígonos em torno de uma fronteira contínua.*
	*Tradução*
	BOUNDARY_POLYGONS(*atributoSelecionado****.geo)***

Tabela 4.25: Representação do uso de uma função espacial delimitada internamente por uma fronteira simples ou contínua.

Interior	*Especifica uma relação geográfica limitada pelo interior por uma fronteira contínua.*
	*Tradução*
	INTERIOR(*atributoSelecionado****.geo)***

Tabela 4.26: Representação do uso de uma função espacial delimitada internamente por pontos contidos em uma fronteira simples ou contínua.

InteriorNodes	*Especifica uma relação geográfica limitada pelo interior por pontos contidos em uma fronteira contínua.*
	*Tradução*
	BOUNDARY_NODES(*atributoSelecionado****.geo)***

Tabela 4.27: Representação do uso de uma função espacial delimitada internamente por segmentos de uma fronteira simples ou contínua.

InteriorEdges	*Especifica uma relação geográfica limitada pelo interior por segmentos contidos em uma fronteira contínua.*
	*Tradução*
	INTERIOR_EDGES(*atributoSelecionado****.geo)***

Tabela 4.28: Representação do uso de uma função espacial delimitada internamente por polígonos formados em torno de uma fronteira simples ou contínua.

InteriorPolygons	*Especifica uma relação geográfica limitada em seu interior por polígonos em torno de uma fronteira contínua.*
	*Tradução*
	INTERIOR_POLYGONS(*atributoSelecionado****.geo)***

4.4.12 Definição de predicados espaciais

Tabela 4.29: Representação de um predicado espacial métrico utilizado entre duas entidades geográficas no ambiente.

Relacionamento Métrico ou de Distância	*Especifica um predicado espacial de distância (métrico) a ser utilizado num relacionamento entre duas entidades geográficas do ambiente.*
	*Tradução*
	DISTANCE(*atributoSelecionado****.geo,*** *atributoSelecionado****.geo,*** *distancia****)***

Tabela 4.30: Representação de um predicado espacial direcional utilizado entre duas entidades geográficas no ambiente.

Relacionamento Direcional	*Especifica um predicado espacial direcional a ser utilizado num relacionamento entre duas entidades geográficas do ambiente.*
	*Tradução*
	DIRECTION(*atributoSelecionado* ***.geo,*** *atributoSelecionado* ***.geo)***

Tabela 4.31: Representação de um predicado espacial topológico utilizado entre duas entidades geográficas no ambiente.

Relacionamento Topológico	*Especifica um predicado espacial topológico a ser utilizado num relacionamento entre duas entidades geográficas do ambiente.*
	*Tradução*
	TOPOLOGY(*atributoSelecionado* ***.geo,*** *atributoSelecionado* ***.geo)***

Obs.: A tradução de cada um desses relacionamentos será detalhada na Seção 4.5.1.2.

4.4.13 Definição Geral de um operador geográfico

Tabela 4.32: Representação geral de um operador geográfico.

Operador Geográfico	*Especifica o uso de um operador geográfico por uma feição geográfica qualquer do ambiente.*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

4.4.14 Definição de operadores geográficos

Tabela 4.33: Representação do operador geográfico CONTAINS.

CONTAINS	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial contém outra.*
	*Tradução*
	A CONTAINS B

Tabela 4.34: Representação do operador geográfico WITHIN.

WITHIN	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está dentro de outra.*
	*Tradução*
	A WITHINB

Tabela 4.35: Representação do operador geográfico INTERSECT.

INTERSECT	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial intersepta outra.*
	*Tradução*
	A INTERSECT B

Tabela 4.36: Representação do operador geográfico NEIGHBOR.

NEIGHBOR	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial é vizinha de outra.*
	*Tradução*
	A NEIGHBOR B

Tabela 4.37: Representação do operador geográfico EQUAL.

EQUAL	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial é igual a outra.*
	*Tradução*
	A EQUAL B

Tabela 4.38: Representação do operador geográfico UNDER.

UNDER	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está abaixo de outra.*
	*Tradução*
	A UNDER B

Tabela 4.39: Representação do operador geográfico OVER.

OVER	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está acima de outra.*
	*Tradução*
	A OVER B

Tabela 4.40: Representação do operador geográfico WEST_FROM.

WEST_FROM	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está a oeste de outra.*
	*Tradução*
	A WEST_FROM B

Tabela 4.41: Representação do operador geográfico EAST.

EAST	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está a leste de outra.*
	*Tradução*
	A EAST B

Tabela 4.42: Representação do operador geográfico SOUTH_FROM.

SOUTH_FROM	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está ao sul de outra.*
	*Tradução*
	A SOUTH_FROM B

Tabela 4.43: Representação do operador geográfico NORTH_FROM.

NORTH_FROM	*Especifica um tipo específico de relacionamento espacial no qual uma entidade espacial está ao norte de outra.*
	*Tradução*
	A NORTH_FROM B

4.4.15 Definição de operadores relacionais

Tabela 4.44: Representação geral dos operadores relacionais.

Operador Relacional Geral	*Especifica um operador relacional geral a ser utilizado na consulta.*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

Tabela 4.45: Tabela de operadores relacionais específicos.

Nome do Operador	Símbolo	Tradução
Igual		=
Diferente		< >
Menor		<
Menor e Igual		< =
Maior		>
Maior e Igual		> =
LIKE		***LIKE***

4.4.16 Definição de operadores lógicos (booleanos)

Tabela 4.46: Representação geral dos operadores lógicos.

Operador Lógico Geral	*Especifica um operador lógico geral a ser utilizado na consulta.*
	*Tradução*
	Não gera script de consulta. Serve apenas para dar uma melhor legibilidade na consulta visual que está sendo formulada.

Tabela 4.47: Tabela de operadores lógicos específicos.

Nome do Operador	Símbolo	Tradução
NOT		***not***
AND		***and***
OR		***or***

Obs.: Devido à limitação do tempo, em versões futuras da linguagem serão definidas novas representações pictóricas relativas às especificações de outras tarefas de mineração sobre dados geográficos.

4.5 FORMALISMO GRAMATICAL DA LINGUAGEM

Ao se definir formalmente uma linguagem de consulta visual é de extrema importância especificar as representações pictóricas de seus principais elementos e operadores como visto anteriormente na Seção 4.3. Além disso, é necessário, também, especificarmos os símbolos terminais e não-terminais da linguagem.

Esta especificação pode ser simulada através dos símbolos não-terminais encontrados no ***Apêndice B*** deste trabalho, uma vez que há equivalência entre os símbolos terminais e não-terminais da linguagem GeoMiningVisualQL com os símbolos terminais e não-terminais da linguagem GMQL. Além disso, é importante ressaltarmos que apenas os símbolos não-terminais apresentam representações pictóricas na linguagem GeoMiningVisualQL, uma vez que os símbolos terminais expressam apenas dados brutos colhidos dos metadados do domínio geográfico que está sendo analisado.

4.5.1 Etapas de construção da gramática da linguagem GeoMiningVisualQL

A seguir, têm-se a construção de todo formalismo gramatical da linguagem *GeoMiningVisualQL,* baseado no formalismo gramatical da linguagem GMQL definido, também, no Apêndice B.

É importante ressaltar que este formalismo gramatical se restringe à parte das tarefas de regras de associação pertencentes à linguagem GMQL. Sendo assim, a árvore gerada por esse formalismo pode ser construídas a partir de partes específicas de todo um processo de mineração, tais como:

4.5.1.1 Definição da tarefa de mineração a ser realizada sobre determinados dados geográficos

Como visto anteriormente, a linguagem GMQL proporciona ao usuário scripts de criação de uma consulta qualquer ou scripts para a definição de hierarquias. Caso seja escolhida a construção de uma consulta, a primeira etapa de construção da gramática é

formada pelos símbolos pictográficos os quais irão determinar a tarefa de mineração a ser realizada sobre um determinado ambiente geográfico (Figura 4.1).

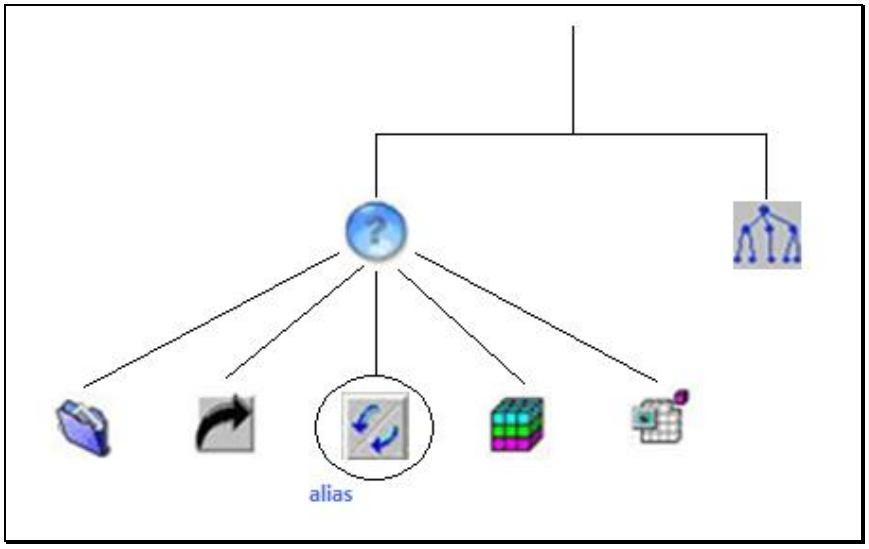


Figura 4.1: Primeira etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Vale lembrar que a constante *alias* é utilizada para a definição de um alias para a consulta.

Tradução da consulta visual responsável em definir a tarefa de mineração a ser utilizada:

A seguir, na Figura 4.2, será demonstrada a tradução desta primeira etapa de configuração de uma consulta na linguagem *GeoMiningVisualQL*. Para isto, será feita a restrição das possíveis tarefas de mineração para o tipo de regras de associação espacial existente entre entidades geográficas quaisquer de um ambiente geográfico específico.

Inicia uma consulta em GeoMiningVisualQL, restringindo a consulta para uma tarefa de regra de associação espacial.

Tradução

MINE SPATIAL ASSOCIATION WITH RESPECT TO [DESCRIBING alias]

Figura 4.2: Seleção de uma tarefa de regra de associação espacial.

Obs.: O script ***[DESCRIBING alias]*** é opcional uma vez que o usuário não possui a obrigação de definir um alias para a consulta.

4.5.1.2 Seleção de termos espaciais e não-espaciais a serem investigados na tarefa

Esta etapa é responsável em selecionar os termos espaciais e não-espaciais a serem investigados na tarefa de mineração que está acontecendo em um determinado processo de descoberta de conhecimento. Quanto aos termos não-espaciais, eles podem ser analisados através de operadores aritméticos e funções não-espaciais. Em relação aos termos espaciais, eles podem ser analisados através do uso de funções e/ou predicados espaciais. (Figura 4.3).

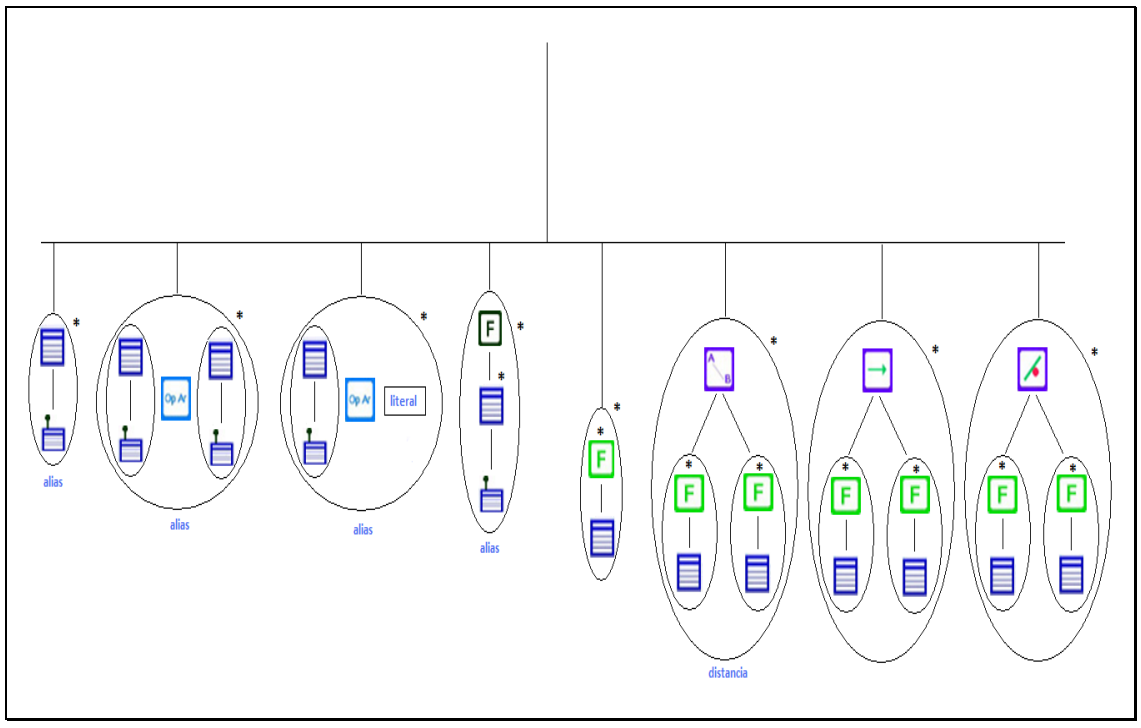


Figura 4.3: Segunda etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Analisando de forma geral, da esquerda para a direita, a etapa de seleção de um termo não-espacial pode ser decomposta na seleção de um atributo simples de uma entidade qualquer e na seleção da junção de dois ou mais atributos simples através de um operador aritmético qualquer ou de um atributo simples e um valor literal fixo. Por fim, pode ser utilizada a seleção de uma função simples aplicada a determinado atributo não-espacial da entidade.

Por outro lado, analisando a seleção de termos espaciais, pode-se realizar a escolha de uma função espacial aplicada sobre um atributo espacial de uma feição geográfica ou a seleção de relacionamentos espaciais existentes entre feições geográficas de um ambiente geográfico especifico.

Observe que, nesta etapa, também se pode definir um alias para cada termo não-espacial utilizado na consulta. Além disso, o valor ***** significa a possibilidade de uma seleção recursiva sobre esses termos.

Vale lembrar que alguns símbolos pictográficos relativos às restrições espaciais impostas pelos predicados espaciais presentes na gramática estão generalizados. Assim, eles podem ser decompostos, conforme mostra a Figura 4.4.

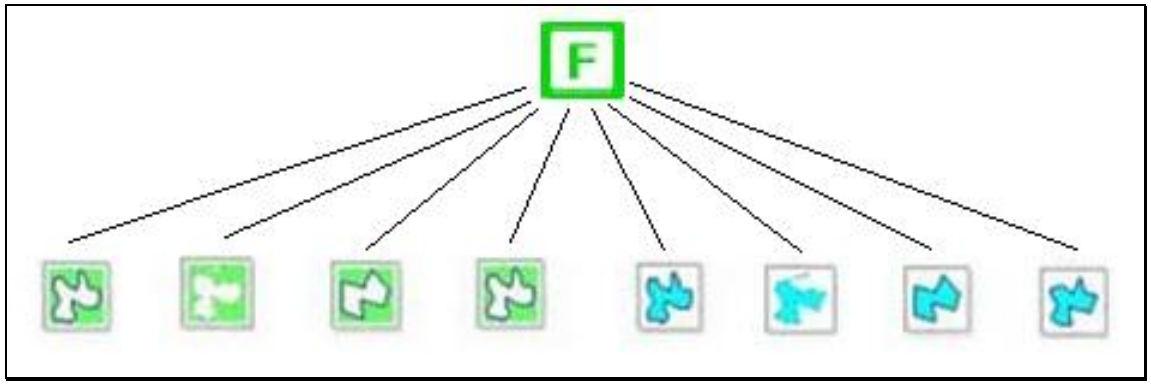


Figura 4.4: Decomposição do símbolo pictográfico de restrições espaciais.

Por outro lado, o símbolo de uma função não-espacial pode ser decomposto em símbolos de funções específicos, definidos na Seção 4.4.9, conforme ilustra a Figura 4.5.

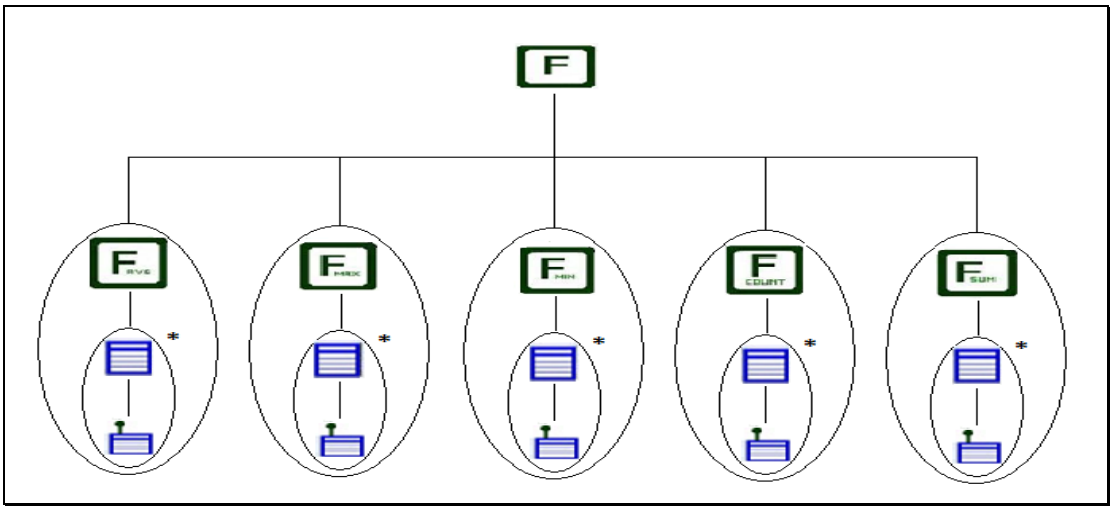


Figura 4.5: Decomposição do símbolo pictográfico de funções simples.

Já o símbolo geral de um operador aritmético pode ser decomposto em operadores específicos, tal como mostra a Figura 4.6.

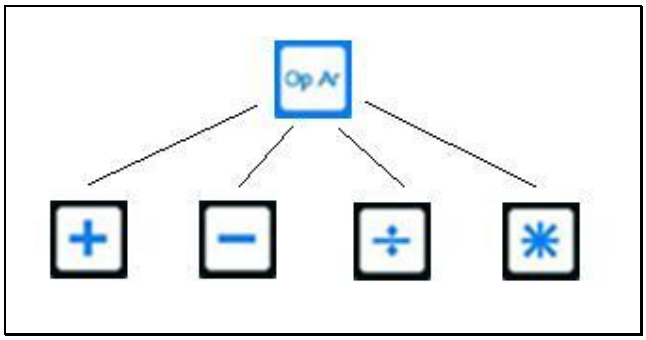
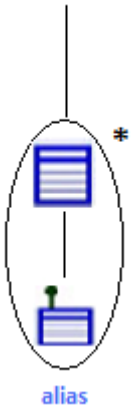


Figura 4.6: Decomposição do símbolo pictográfico de operadores aritméticos.

Tradução de parte da consulta visual responsável em selecionar termos não-espaciais:

Como visto anteriormente na Figura 4.3, a seleção de termos não-espaciais na consulta pode ser decomposta em quatro tipos. Em seguida, serão demonstradas as traduções da representação da consulta visual feita para cada um desses tipos.

Observe, na Figura 4.7, que os termos destacados em azul (tal como *Relacao.atributoSimples*) serão encontrados em praticamente todas as traduções de consultas visuais utilizadas nesta Seção 4.5 do capítulo, representando os dados oriundos dos metadados de uma ambiente geográfico analisado pelo sistema que utiliza, internamente, a linguagem GeoMiningVisualQL. Vale ressaltar que o termo *Relacao*, utilizado nas traduções, pode representar qualquer tipo de entidade presente na base de dados, podendo ser, inclusive, uma feição geográfica. Além disso, a parte referente à *Consulta Atual* ilustra a tradução da consulta de todas as etapas anteriores com a tradução da etapa atual.



Seleção de um termo não-espacial como sendo um atributo espacial simples de uma feição geografica .

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução desta etapa

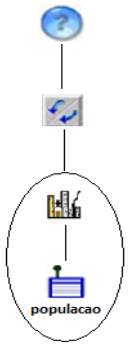
Relacao.atributoSimples ***[DESCRIBING alias]***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO
Relacao.atributoSimples ***[DESCRIBING alias] FROM*** *Relacao*

Figura 4.7: Consulta visual e tradução da seleção de termos não-espaciais simples.

Em um exemplo prático de consulta, o símbolo de uma feição geográfica simples  pode ser substituído por uma representação real contida nos metadados do ambiente geográfico em estudo. Neste caso, baseando-se nos metadados de alguns símbolos visuais definidos para um ambiente geográfico descrito por [Soares 2002], suponha que, por exemplo, o símbolo  represente a feição geográfica de uma cidade e que o usuário deseje analisar como termo não-espacial dela a população de cada cidade da base geográfica (existente como um atributo na base). Para isto, ele configuraria este atributo no sistema que implementa a linguagem GeoMiningVisualQL a partir da consulta visual ilustrada na Figura 4.8.



Exemplo de seleção de um termo não-espacial simples.

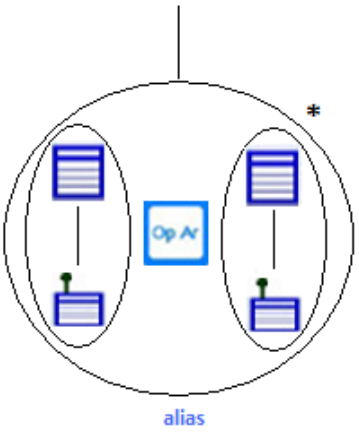
Tradução

MINE SPATIAL ASSOCIATION WITH RESPECT TO
Cidade.populacao ***FROM*** *Cidade*

Figura 4.8: Exemplo de configuração da seleção de um atributo simples de uma entidade cidade.

No caso da escolha do termo não-espacial ser baseada na junção de dois ou mais atributos simples através de um operador aritmético qualquer, temos a representação da Figura 4.9. Neste caso, a utilização deste operador serve para juntar resultados oriundos de atributos não-espaciais numéricos em uma espécie de campo virtual definido através do alias.

Assim, os elementos internos na consulta visual são atributos simples selecionados da mesma forma como demonstrado anteriormente na Figura 4.6. Sendo assim, a consulta visual é baseada numa operação aritmética aplicada sobre atributos simples internos à consulta.



Seleção de um termo não-espacial baseado na junção de dois tributos simples através de um operador aritmético.

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução desta etapa

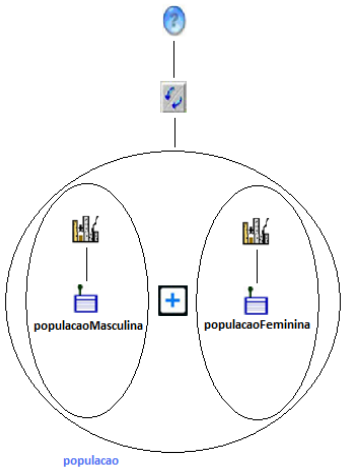
Relacao.atributoSimples ***OperAritmeico*** *Relacao.atributoSimples* ***[DESCRIBING*** *alias* ***]***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO *Relacao.atributoSimples* ***OperAritmetico*** *Relacao.atributoSimples* ***[DESCRIBING alias] FROM*** *Relação*

Figura 4.9: Consulta visual e tradução da seleção de termos não-espaciais simples através da junção de atributos simples por um operador aritmético.

Como exemplo de configuração da consulta nesta etapa, suponha que, desta vez, num determinado estudo, sobre as cidades do estado da Paraíba, o banco de dados possui dados separados em relação à população masculina e feminina de cada cidade deste estado. Todavia, se o usuário não estiver interessado nessa distinção e desejar obter informações a cerca da população de cada cidade como um todo, o mesmo poderia utilizar a união desses dados na própria consulta, tal como mostra a Figura 4.10. Vale lembrar que, neste caso, será criado um campo virtual *população* através de um alias.



Exemplo da seleção de um termo não-espacial (representando a população total das cidades da PB), através da junção de atributos simples (população masculina e população feminina) por um operador aritmético (o qual soma os valores dos atributos simples num único termo não-espacial).

Tradução

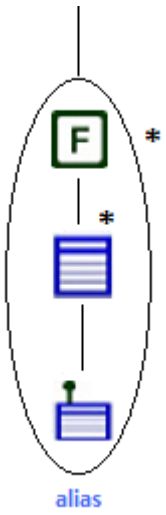
MINE SPATIAL ASSOCIATION WITH RESPECT TO *Cidade.populacaoMasculina + Cidade.populacaoFeminina* ***DESCRIBING*** *populacao* ***FROM*** *Cidade*

Figura 4.10: Exemplo de configuração da seleção de um termo não-espacial, baseando-se na representação da figura 4.9.

Vale ressaltar que a configuração do dado bruto é sempre feita através da definição de valores literais feitas no próprio sistema que utiliza a linguagem. Um exemplo simples e intuitivo seria o usuário dar um duplo clique no símbolo e lhe atribuir um valor.

De maneira semelhante à configuração de consulta vista na Figura 4.9, o usuário pode substituir a escolha da seleção de um atributo simples por um valor literal. Este caso, normalmente, é utilizado quando se define um campo virtual sobre os valores retornados por uma consulta.

Por outro lado, caso seja aplicada alguma função sobre os valores contidos sobre determinada entidade da base de dados do domínio analisado, utiliza-se a configuração de consulta vista na Figura 4.11.



Aplicação de uma função sobre atributos de uma feição geográfica qualquer.

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução desta etapa

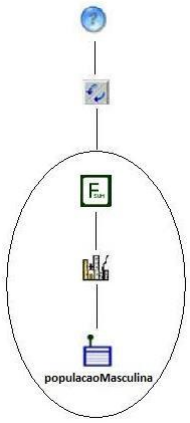
tipoFunção(*Relacao.atributoSimples****)[DESCRIBING alias]***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO *tipoFunção****(****Relacao.atributoSimples****)[DESCRIBING alias] FROM*** *Relacao*

Figura 4.11: Consulta visual aplicando uma função sobre termos não-espaciais do ambiente geográfico.

Como exemplo, suponha que se deseje somar toda a população do sexo masculino de cada cidade. Para isto, utilizaria a configuração da consulta feita na Figura 4.12. Vale ressaltar, mais uma vez, que a configuração do atributo *populacaoMasculina* seria definida no próprio sistema (ambiente visual) que implementa a linguagem *GeoMiningVisualQL*. Além disso, o próprio sistema é que oferece e se encarrega de utilizar determinadas bases de dados do interesse do usuário.

<table>
<tr>
<td rowspan="2"> 
</td>
<td>Aplicação de uma função que soma toda a população masculina das cidades do ambiente analisado. Neste caso, as cidades do Estado da Paraíba.</td>
</tr>
<tr>
<td>Tradução

MINE SPATIAL ASSOCIATION WITH RESPECT TO SUM(Cidade.populacaoMasculina) FROM Cidade</td>
</tr>
</table>

Figura 4.12: Exemplo de configuração de uma consulta visual aplicando uma função para somar a população masculina das cidades do estado da Paraíba.

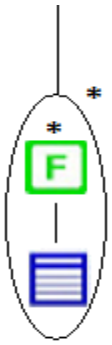
Tradução de parte da consulta visual responsável em selecionar termos espaciais:

Quanto à seleção de termos espaciais presentes em um determinado domínio geográfico, eles podem ser feitos envolvendo funções espaciais ou predicados espaciais. As funções espaciais são aplicadas sobre feições geográficas, delimitando os resultados de uma consulta sobre algumas áreas geográficas. A configuração deste tipo de consulta é generalizada como mostra a Figura 4.13.

Neste caso, o valor ***R*** de recursão, interno à consulta, é utilizado para que seja possível utilizar sucessivas funções espaciais no intuito de restringir áreas internas já restritas pela aplicação de funções espaciais anteriores.

Um exemplo comum deste tipo de consulta é quando se utiliza a função geográfica responsável em delimitar a análise do processo de descoberta de conhecimento sobre a região de fronteira de alguma feição geográfica. Para isto, substitui-se o símbolo geral de uma função espacial  pelo próprio símbolo de delimitação da região de fronteira  de uma feição geográfica.

Como exemplo prático de uma consulta fazendo-se uso de funções espaciais, suponha que um usuário esteja interessado em pesquisar os nomes de todas as cidades que são ribeirinhas aos rios do domínio geográfico analisado. Além disso, considere que o símbolo  represente os rios contidos nesse domínio. Neste caso, temos a construção da consulta vista na Figura 4.14



Delimitação de uma região geográfica fazendo-se uso de uma função espacial aplicada sobre termos espaciais.

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

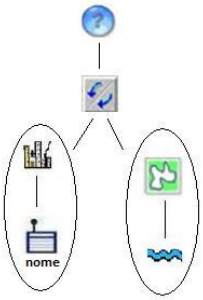
Tradução desta etapa

[tipoFuncao]*tipoFunção(*Relacao.geo****)***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO
[tipoFuncao]*tipoFunção(*Relacao.geo****) FROM*** *Relacao*

Figura 4.13: Consulta visual aplicando uma função espacial sobre feições geográficos de um ambiente.



Aplicação de uma função que soma toda a população masculina das cidades do ambiente analisado. Neste caso, as cidades do Estado da Paraíba.

Tradução

MINE SPATIAL ASSOCIATION WITH RESPECT TO
Cidade.nome , ***BOUNDARY(****Rio.geo****) FROM*** *Cidade, Rio*

Figura 4.14: Exemplo de configuração de uma consulta visual fazendo uso de funções espaciais.

Por outro lado, caso a escolha do termo espacial a ser utilizado na análise do processo de descoberta de conhecimento seja baseada num relacionamento espacial existente entre entidades geográficas do ambiente, ela pode ser configurada a partir de um relacionamento métrico, topológico ou direcional, conforme ilustra a Figura 4.15.

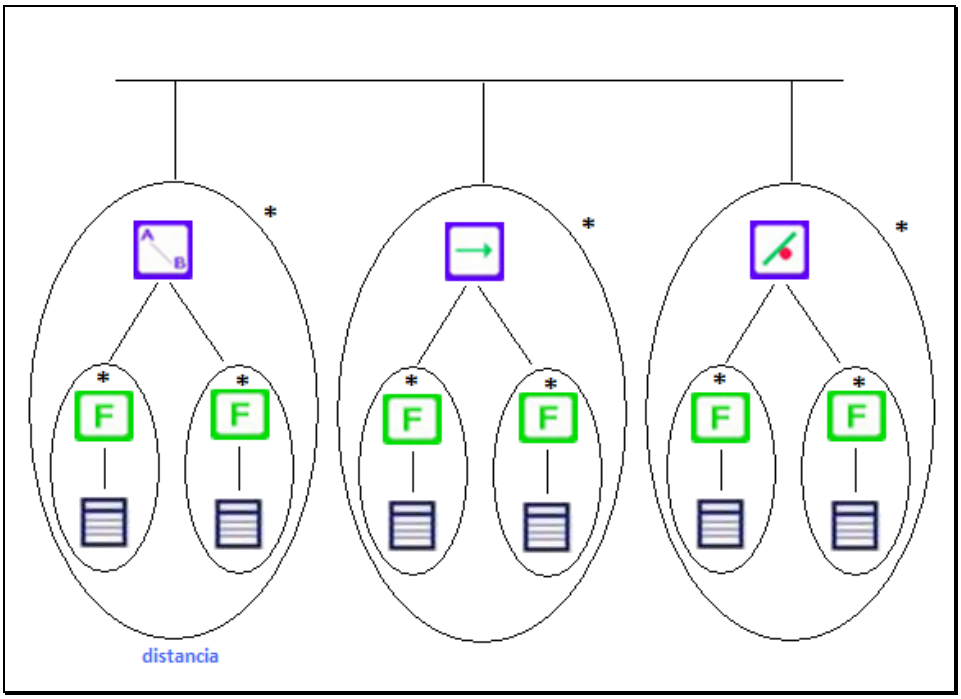
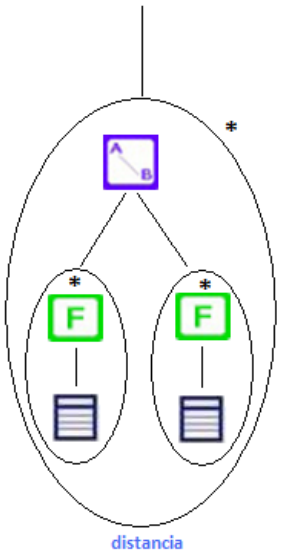


Figura 4.15: Seleção de termos espaciais através de relacionamentos existentes entre feições geográficas do ambiente.

É importante ressaltar que estes tipos de relacionamentos são classificados e conhecidos como predicados espaciais uma vez que restringem uma área geográfica. Como se pode observar na Figura 4.15, as configurações desses relacionamentos na linguagem GeoMiningVisualQL é bastante similares. Todavia, os relacionamentos métricos necessitam da entrada de um valor de distância a ser utilizado no mesmo. A seguir, nas Figuras 4.16, 4.17 e 4.18 são encontrados as traduções generalizadas de cada uma dessas partes da consulta visual.



Delimitação de uma região geográfica fazendo-se uso de um predicado espacial métrico aplicado sobre termos espaciais.

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

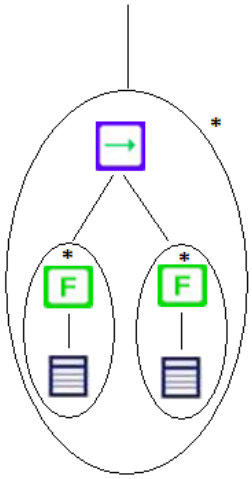
Tradução desta etapa

CLOSE_TO(*feicaoGeográfica1.geo****,*** *feicaoGeográfica2.geo****,*** *distancia****)***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO CLOSE_TO(*FeicaGeografica.geo1****,****FeicaoGeografica2.geo****,*** *distancia****)*** ***FROM*** *FeicaoGeografica*

Figura 4.16: Configuração de um predicado espacial métrico.



Delimitação de uma região geográfica fazendo-se uso de um predicado espacial direcional aplicado sobre termos espaciais.

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

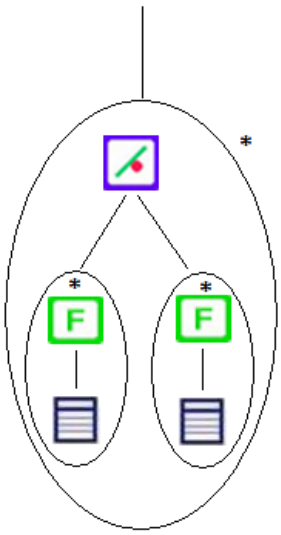
Tradução desta etapa

DIRECTION(*FeicaoGeografica1.geo****,*** *FeicaoGeografica2.geo****)***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO DIRECTION(*FeicaoGeografica1.geo****,*** *FeicaoGeografica2.geo****)*** ***FROM*** *FeicaoGeografica*

Figura 4.17: Configuração de um predicado espacial direcional.



Delimitação de uma região geográfica fazendo-se uso de um predicado espacial topológico aplicado sobre termos espaciais.

Traduções de etapas anteriores

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução desta etapa

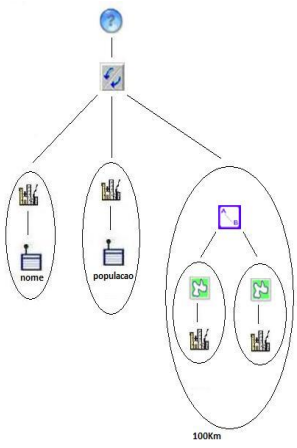
TOPOLOGY(*FeicaoGeografica1.geo****,*** *FeicaoGeografica2.geo****)***

Consulta Atual

MINE SPATIAL ASSOCIATION WITH RESPECT TO
TOPOLOGY(*FeicaoGeografica1.geo****,*** *FeicaoGeografica2.geo****)***
FROM *FeicaoGeografica*

Figura 4.18: Configuração de um predicado espacial topológico.

Considere o seguinte exemplo prático de consulta utilizando um predicado espacial métrico. Suponha que o usuário deseje recuperar os nomes e a população de todas as cidades que eqüidistam uma da outra, em relação as suas fronteiras, uma distância de 100km. A consulta visual formulada, neste caso, é vista na Figura 4.19.



Exemplo de utilização de um predicado espacial métrico na linguagem GeoMiningVisualQL

Tradução

MINE SPATIAL ASSOCIATION WITH RESPECT TO
Cidade.nome , Cidade.populacao ***CLOSE_TO(***
BOUNDARY(*Cidade.geo****),*** ***BOUNDARY(****Cidade.geo****),*** *100km****)***
FROM *Cidade*

Figura 4.19: Exemplo de uma consulta utilizando um predicado espacial métrico.

Da maneira semelhante a formulação de consulta feita para um predicado espacial métrico são feitas as formulações de consultas para predicados espaciais direcionais e topológicos.

4.5.1.3 Restrição condicional da consulta

Nesta etapa são utilizados símbolos pictográficos, conforme a Figura 4.20, responsáveis em realizar restrições condicionais na consulta que está sendo formulada. Essas restrições podem ser utilizadas de diversas maneiras, fazendo-se uso de operadores lógicos unários ou binários ou operadores relacionais para os termos não-espaciais. Por outro lado, para os termos espaciais, podem ser utilizados operadores geográficos, definidos na Seção 4.4.14. Além disso, é possível, também, aplicar um operador relacional oriundo do resultado de um *relacionamento métrico* entre entidades geográficas pertencentes ao domínio analisado.

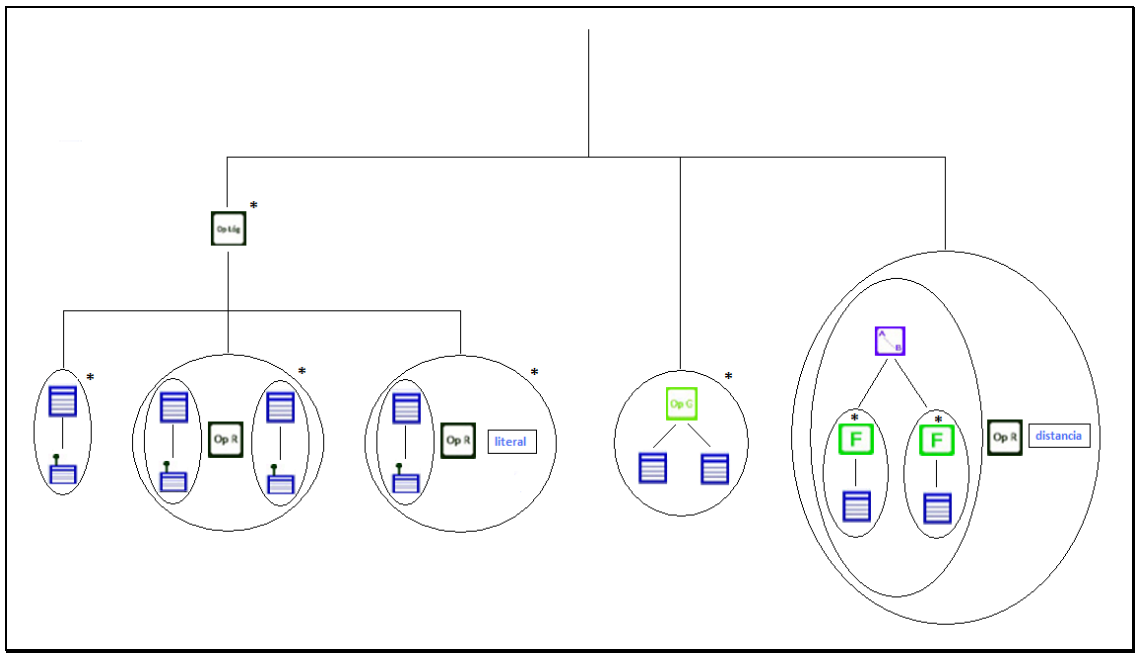


Figura 4.20: Terceira etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Da mesma forma como acontece com a generalização de símbolos relacionados a restrições espaciais sobre determinadas feições geográficas (Figura 4.4), também é possível decompor o símbolo geral de um operador geográfico em seus derivados, conforme ilustra a Figura 4.21.

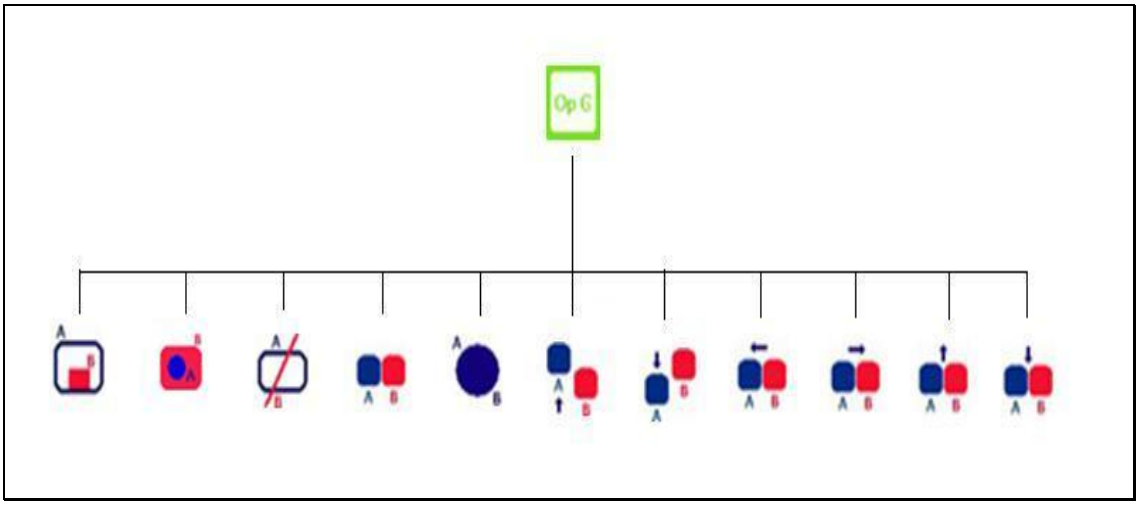


Figura 4.21: Decomposição do símbolo pictográfico de operador geográfico.

Por outro lado, o símbolo geral de operadores lógicos pode ser decomposto em símbolos específicos, tal como ilustra a Figura 4.22.

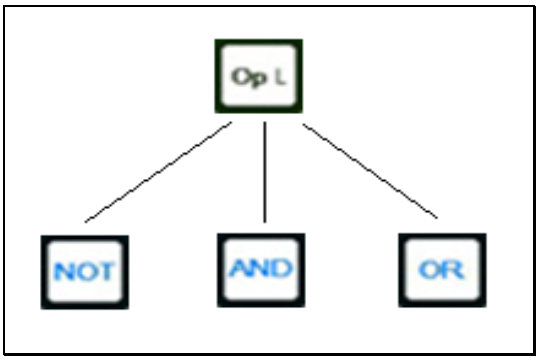


Figura 4.22: Decomposição do símbolo pictográfico de operador lógico.

Além disso, o símbolo geral de um operador relacional pode ser decomposto, conforme mostra a Figura 4.23.

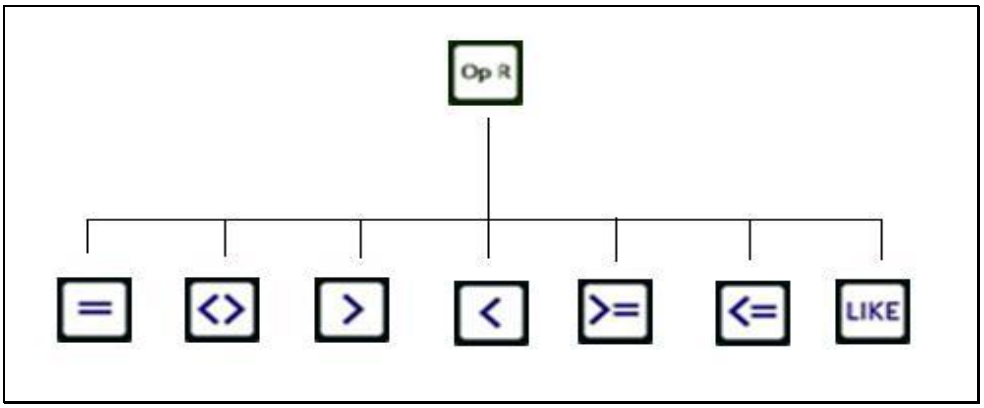
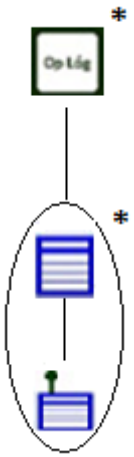


Figura 4.23: Decomposição do símbolo pictográfico de operador relacional.

Tradução de parte da consulta visual referente à utilização de restrições condicionais:

De maneira geral, esta etapa envolve duas partes: restrições condicionais sobre termos não-espaciais do ambiente e restrições condicionais sobre termos espaciais do ambiente geográfico analisado.

No caso da utilização de operadores lógicos unários sobre termos não-espaciais, estes são aplicados sobre um único atributo simples (Figura 4.24) ou sobre outros relacionamentos lógicos (Figuras 4.25 e 4.26). Todavia, operadores lógicos não-unários podem ser utilizados na junção de algumas dessas representações isoladamente. Além disso, a configuração de consulta vista na Figura 4.24, em particular, é utilizada sobre atributos booleanos pertencentes a entidades do domínio.

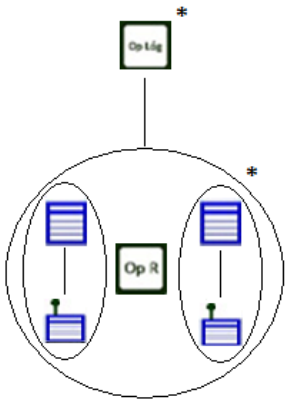


Seleção condicional de um operador lógico sobre atributos simples de entidades de um domínio.

Tradução

WHERE** [tipoOperadorLogico]**(**atributoSimples**)

Figura 4.24: Configuração visual da consulta na seleção condicional de um operador lógico aplicado sobre um atributo simples.

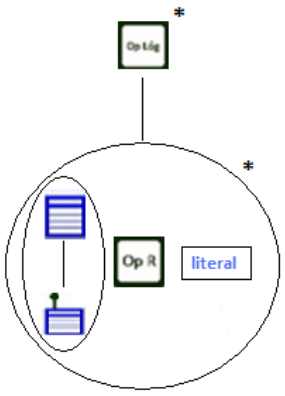


Seleção condicional utilizando um operador lógico com expressões relacionais sobre atributos simples das entidades de um domínio.

Tradução

WHERE** [tipoOperadorLogico]**(**atributoSimples ExpressaoRelacional atributoSimples**)

Figura 4.25: Configuração visual da consulta baseada na seleção condicional utilizando operadores lógicos e expressões relacionais.



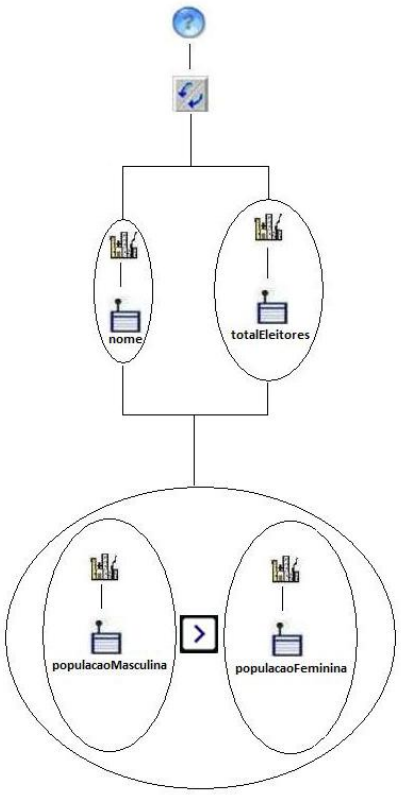
Seleção condicional utilizando um operador lógico com expressões relacionais entre um valor literal (definido pelo usuário) e um atributo simples de uma entidade do domínio.

Tradução

WHERE** [tipoOperadorLogico]**(**atributoSimples ExpressaoRelacional literal**)

Figura 4.26: Configuração visual da consulta baseada na seleção condicional utilizando operadores lógicos e um valor literal entre expressões relacionais.

Um exemplo de consulta, utilizando configurações visuais desta etapa pode ser visto na Figura 4.27. Neste caso, suponha que o usuário deseje recuperar todas as cidades junto com seus números de eleitores na qual a população masculina é maior que a população feminina.



Tradução da primeira etapa

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução da segunda etapa

*Cidade.nome , Cidade.totalEleitores **FROM** Cidade*

Tradução da etapa atual

WHERE (**populacaoMasculina > poluacaoFeminina**)

Consulta Completa

MINE SPATIAL ASSOCIATION WITH RESPECT TO
*Cidade.nome , Cidade.totalEleitores **FROM** Cidade*
WHERE (**populacaoMasculina > poluacaoFeminina**)

Figura 4.27: Exemplo de parte de uma consulta fazendo restrição do registros do resultado através de expressões relacionais.

Caso outra condição seja dada para a consulta feita na Figura 4.27, é fácil ver que a alteração é feita apenas na etapa responsável em realizar as restrições condicionais. Considere o atributo *segundoTurno*, do tipo booleano. Se este atributo estiver falso indica que o prefeito atual desta cidade foi eleito no primeiro turno das eleições deste município. Caso contrário, se ele for verdadeiro, indica que o atual prefeito foi eleito apenas no segundo turno das eleições. Assim, baseando-se na consulta feita anteriormente, suponha que o usuário esteja interessado em recuperar todas as cidades junto com seus números de eleitores na qual a população masculina é maior que a população feminina e obteve suas eleições atuais determinadas apenas no primeiro turno. A formulação desta consulta pode ser vista, a seguir, na Figura 4.28.

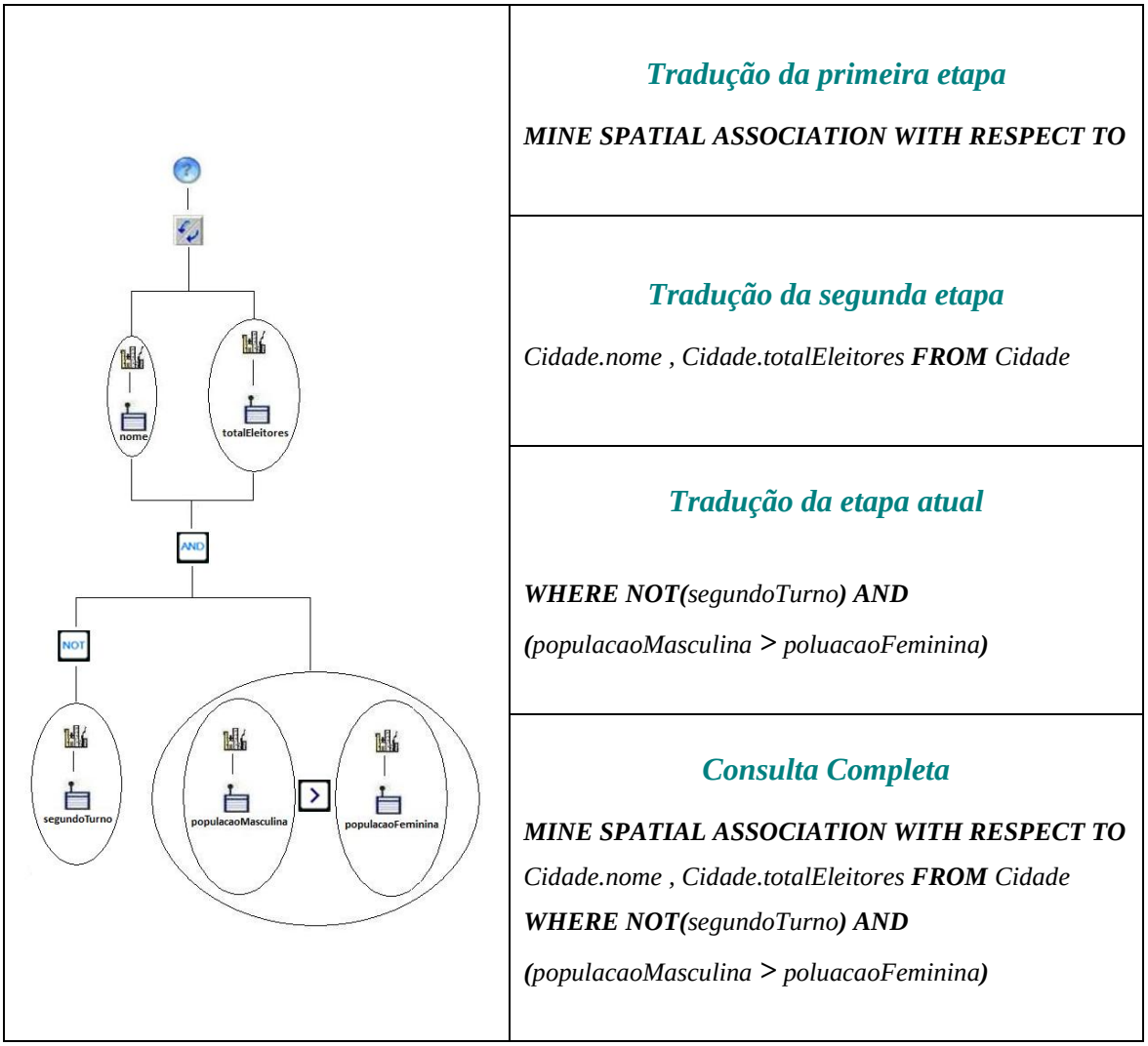
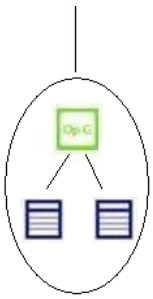


Figura 4.28: Exemplo de parte de uma consulta fazendo restrição do registros do resultado através de operadores lógicos e de expressões relacionais.

Por outro lado, quando a restrição de parte da consulta é referente aos termos espaciais, ela pode ser feita aplicando-se um operador geográfico entre duas entidades geográficas (Figura 4.29), caracterizando um dos relacionamentos encontrados na Figura 4.20, ou através da aplicação de um operador relacional aplicado sobre um predicado espacial métrico existente entre entidades geográficas do domínio analisado (Figura 4.30).

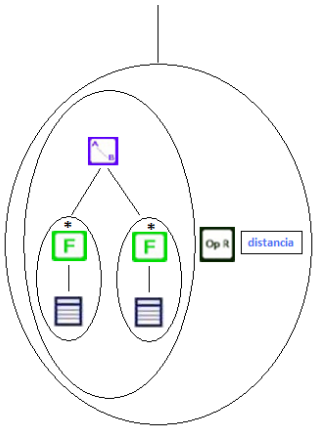


Restrição condicional utilizando um operador geográfico entre entidades do ambiente.

Tradução

entidadeGeografica1.geo ***operadorGeografico*** *entidadeGeografica2.geo*

Figura 4.29: Restrição condicional baseada em operadores geográficos.



Restrição condicional fazendo uso de um predicados espacial métrico.

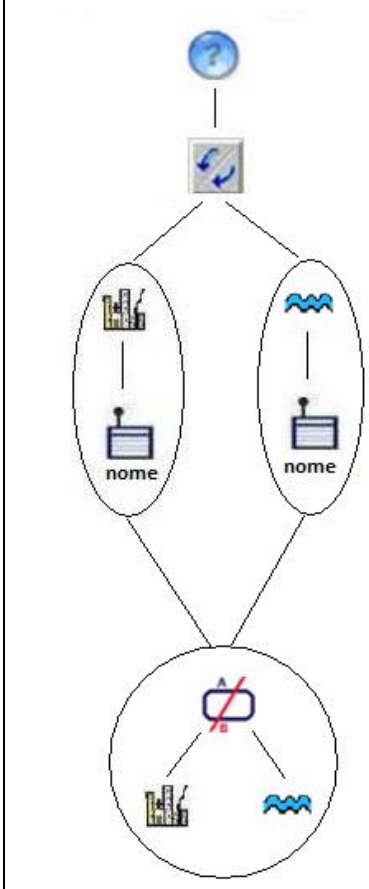
Tradução

G_CLOSE_TO(*feicaoGeográfica1.geo****,*** *feicaoGeográfica2.geo****)***
operadorRelacional distanciaEspecificada

Figura 4.30: Restrição condicional baseada em predicados espaciais métricos.

Um exemplo prático e bastante simples em relação ao uso de operadores geográficos entre entidades de um domínio geográfico pode ser visto da seguinte forma. Suponha que, em um ambiente geográfico específico, o símbolo  represente a feição geográfica de uma cidade e o símbolo  represente a feição geográfica de um rio qualquer. Assim, suponha que o usuário, ao procurar extrair novos conhecimentos da base de dados geográficos deste ambiente específico, esteja interessado em saber os nomes de

todas as cidades que são interceptadas por rios e os nomes desses rios que as interceptam. Para isto, ele configuraria a consulta visual ilustrada na Figura 4.31.



Tradução da primeira etapa

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução da segunda etapa

Cidade.nome , Rio.nome ***FROM*** *Cidade, Rio*

Tradução da etapa atual

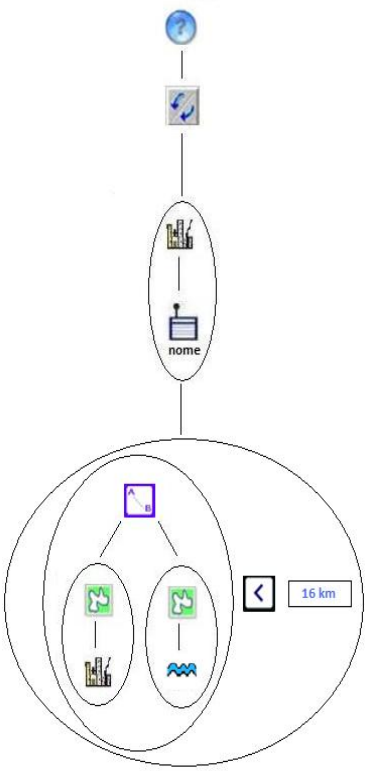
WHERE *Cidade.geo* ***INTERSECTS*** *Rio.geo*

Consulta Completa

MINE SPATIAL ASSOCIATION WITH RESPECT TO
Cidade.nome , Rio.nome ***FROM*** *Cidade, Rio* ***WHERE***
Cidade.geo ***INTERSECTS*** *Rio.geo*

Figura 4.31: Exemplo da utilização de um operador geográfico entre entidades geográficas do ambiente.

Já no caso da utilização de operadores relacionais sobre um predicado espacial métrico, considere que, numa consulta, o usuário procure, por exemplo, cidades cuja distância (em relação as suas fronteiras) para rios seja inferior a 16 km. Neste caso, tal condição pode ser configurada, conforme a Figura 4.32.



Tradução da primeira etapa

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução da segunda etapa

Cidade.nome ***FROM*** *Cidade, Rio*

Tradução da etapa atual

WHERE CLOSE_TO(BOUNDARY(*Cidade.geo****)***, ***BOUNDARY(*** *Rio.geo****))*** *< 16 km*

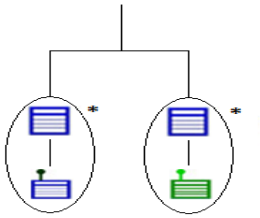
Consulta Completa

MINE SPATIAL ASSOCIATION WITH RESPECT TO *Cidade.nome* ***FROM*** *Cidade, Rio* ***WHERE CLOSE_TO(BOUNDARY(*** *Cidade.geo****)***, ***BOUNDARY(*** *Rio.geo****)) < 16 km***

Figura 4.32: Exemplo de restrição condicional fazendo uso de um predicado espacial métrico.

4.5.1.4 Agrupamento de atributos envolvidos na consulta

Nesta etapa, são selecionados atributos não-espaciais ou espaciais presentes em determinadas feições geográficas do ambiente de domínio analisado os quais podem ser utilizados para ordenar os dados provenientes do resultado de uma consulta a base de dados geográficos que contém esse domínio em grupos distintos (Figura 4.33).



Agrupamento de atributos envolvidos na consulta

Tradução

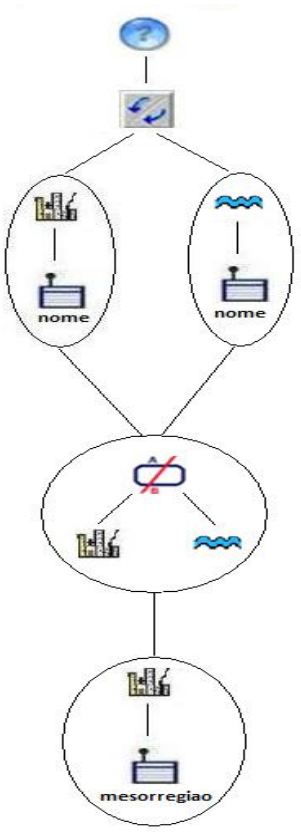
GROUP BY *(atributo)*$^{+}$

Figura 4.33: Quarta etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Obs.: O termo *(atributo)*$^{+}$ especifica que no mínimo um de qualquer tipo de atributo (espacial ou não) pode estar envolvido no agrupamento.

Tradução de parte da consulta visual referente condições de agrupamento de termos espaciais e não-espaciais:

Baseando-se na consulta realizada na Figura 4.31 através do qual um usuário estava interessado em recuperar todos os nomes das cidades que são interceptadas por rios e os nomes desses rios que as interceptam, suponha que o mesmo tem o interesse de agrupar este resultado através das mesorregiões (Agreste, Sertão, Cariri, etc.) que fazem parte do estado da Paraíba, uma vez que, a princípio, é o ambiente geográfico que está sendo levado em consideração neste estudo. Para isto, a consulta visual seria formulada como mostra a Figura 4.34.



Tradução da primeira etapa

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução da segunda etapa

Cidade.nome, Rio.nome ***FROM*** *Cidade, Rio*

Tradução da terceira etapa

WHERE *Cidade.geo* ***INTERSECST*** *Rio.geo*

Tradução da etapa atual

GROUP BY *Cidade.mesorregiao*

Consulta Completa

MINE SPATIAL ASSOCIATION WITH RESPECT TO
Cidade.nome, Rio.nome ***FROM*** *Cidade, Rio* ***WHERE***
WHERE *Cidade.geo* ***INTERSECST*** *Rio.geo*
GROUP BY *Cidade.mesorregiao*

Figura 4.34: Exemplo de agrupamento de um termo não-espaciais na consulta.

4.5.1.5 Condições do reagrupamento da etapa IV

Neste momento, são utilizados elementos responsáveis em restringir a consulta baseando-se em condições sobre o agrupamento feito na etapa anterior. Observe que os elementos que podem ser utilizados, nesta etapa, são os mesmos utilizados na terceira etapa de configuração de uma consulta em GeoMiningVisualQL, visto na Seção 4.5.1.3 (Figura 4.35).

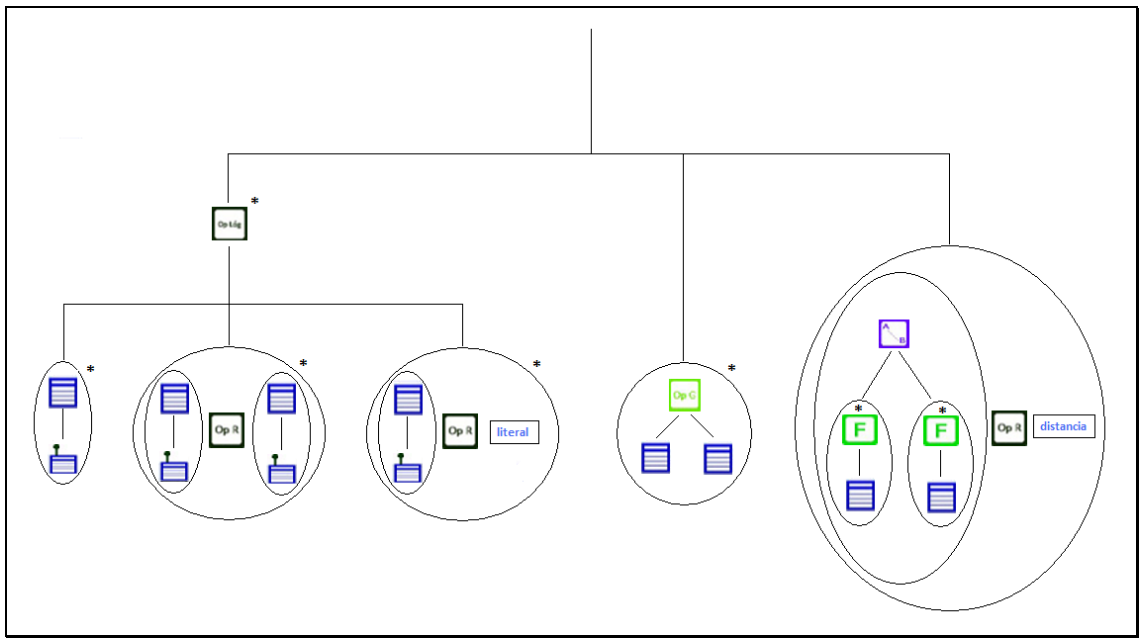


Figura 4.35: Quinta etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Todavia, de maneira geral, a distinção desta etapa em relação à terceira etapa de configuração vista na Seção 4.5.1.3 é que esta restrição de condição está relacionada com o agrupamento feito na etapa anterior, caracterizando condições para o agrupamento de termos espaciais ou não-espaciais envolvidos na consulta. Por outro lado, a restrição de condição vista na segunda etapa está relacionada aos campos a serem visualizados nos registros retornados por uma consulta.

Tradução de parte da consulta visual referente à utilização de restrições condicionais sobre termos agrupados na consulta:

As configurações de construção das partes de consultas visuais feitas nesta etapa são idênticas as configurações utilizadas na segunda etapa. Todavia, a tradução sofre uma

pequena alteração, trocando-se o script **WHERE** por **HAVING**, conforme ilustra a Figura 4.36.

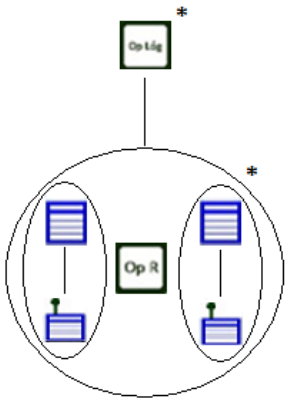
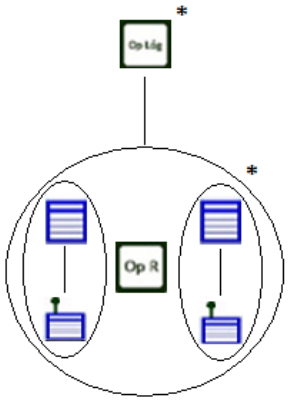
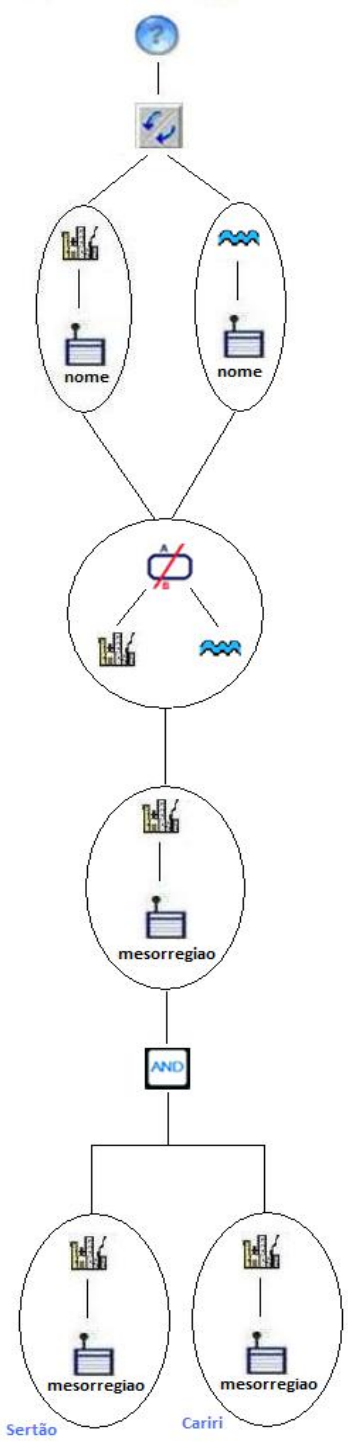
TERCEIRA ETAPA	
	*Tradução* ***WHERE*** *[tipoOperadorLogico]****(****atributoSimples ExpressaoRelacional atributoSimples****)***
QUINTA ETAPA	
	*Tradução* ***HAVING*** *[tipoOperadorLogico]****(****atributoSimples ExpressaoRelacional atributoSimples****)***

Figura 4.36: Tradução da seleção condicional sobre atributos simples na segunda e na quinta etapa de configuração de parte de uma consulta visual na GeoMiningVisualQL.

Tomando-se como base o exemplo feito na Figura 4.34, no qual a consulta feita por um usuário recupera e agrupa por mesorregiões paraibanas todas as cidades que são interceptadas por rios, suponha que esse agrupamento seja restrito, especificamente, para as regiões Sertão e Cariri. Tal consulta é formulada visualmente, conforme mostra a Figura 4.37.



Tradução da primeira etapa

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução da segunda etapa

Cidade.nome, Rio.nome ***FROM*** *Cidade, Rio*

Tradução da terceira etapa

WHERE *Cidade.geo* ***INTERSECT*** *Rio.geo*

Tradução da quarta etapa

GROUP BY *Cidade.mesorregiao*

Tradução da quinta etapa

HAVING Cidade.*mesorregiao = "Sertão" OR Cidade.mesorregiao = "Cariri"*

Consulta Completa

MINE SPATIAL ASSOCIATION WITH RESPECT TO *Cidade.nome, Rio.nome* ***FROM*** *Cidade, Rio* ***WHERE*** ***WHERE*** *Cidade.geo* ***INTERSECT*** *Rio.geo* ***GROUP BY*** *Cidade.mesorregiao* ***HAVING*** ***Cidade.****mesorregiao = "Sertão" OR Cidade.mesorregiao = "Cariri"*

Figura 4.37: Exemplo de consulta feita em GeoMiningVisualQL fazendo uso de restrição condicional sobre os atributos envolvidos no agrupamento feito na quarta etapa de configuração.

4.5.1.6 Definição de limiares na consulta

Por fim, a última etapa de configuração de uma consulta relativa a tarefas de regras de associação entre entidades geográficas é marcada pela utilização de determinados

parâmetros nos quais os usuários podem controlar o número de padrões retornados durante uma extração de conhecimento (Figura 4.38). Observe que a utilização desses parâmetros foi restrita à medidas mais utilizadas: suporte e confiança.

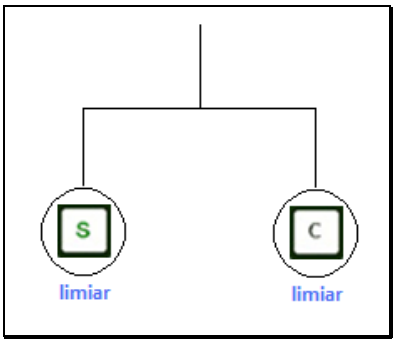


Figura 4.38: Sexta e última etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Neste caso, a constante *limiar* representa o valor de estimativa utilizado no parâmetro da consulta.

Tradução de parte da consulta visual referente a definição de medidas parametrizadas na consulta:

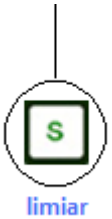
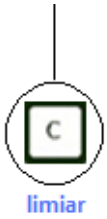
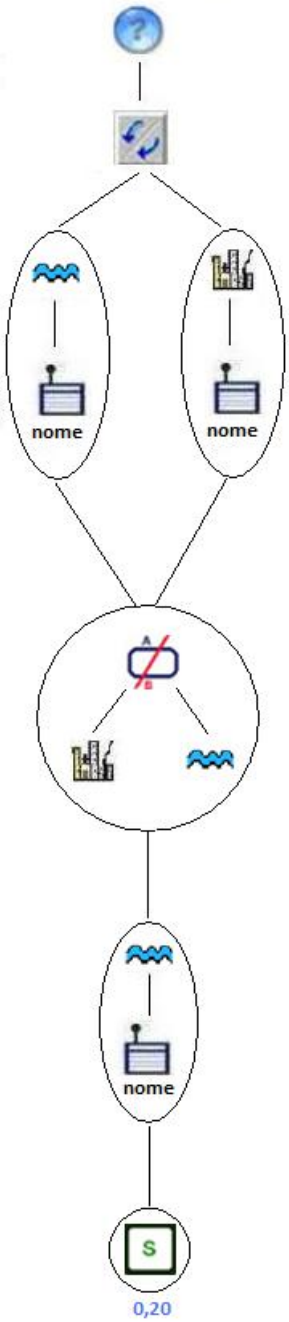
	*Tradução* ***SET SUPPORT THRESHOLD*** *limiar*
	*Tradução* ***SET CONFIDENCE THRESHOLD*** *limiar*

Figura 4.39: Tradução dos parâmetros de suporte e confiança utilizados na sexta e última etapa do formalismo gramatical de parte da linguagem GeoMiningVisualQL.

Como exemplo de utilização destes parâmetros numa consulta feita para a descoberta de conhecimento, considere que o usuário deseja obter todos os rios do estado da Paraíba que interceptam, pelo menos, 20% das cidades paraibanas. Além disso, a consulta deve mostrar todos os nomes dessas cidades, agrupando todo o resultado por cada rio (Figura 4.40).



Tradução da primeira etapa

MINE SPATIAL ASSOCIATION WITH RESPECT TO

Tradução da segunda etapa

Rio.nome, Cidade.nome ***FROM*** *Cidade, Rio*

Tradução da terceira etapa

WHERE *Cidade.geo* ***INTERSECST*** *Rio.geo*

Tradução da quarta etapa

GROUP BY *Rido.nome*

Tradução da quinta etapa

Não há elementos

Tradução da sexta etapa

SET SUPPORT THRESHOLD *0,20*

Consulta Completa

MINE SPATIAL ASSOCIATION WITH RESPECT TO *Rio.nome, Cidade.nome* ***FROM*** *Cidade, Rio* ***WHERE*** *Cidade.geo* ***INTERSECST*** *Rio.geo* ***GROUP BY*** *Rio.nome* ***SET SUPPORT THRESHOLD*** *0,20*

Figura 4.40: Exemplo de uma consulta em GeoMiningVisualQL fazendo uso do parâmetro de suporte no qual controla o número de padrões retornados num processo de descoberta de conhecimento.

Obs.: É fácil ver que numa determinada consulta, feita em *GeoMiningVisualQL*, nem sempre é necessário fazer uso de símbolos visuais contidos em todas as etapas de configuração visual.

4.5.2 Gramática da Linguagem GeoMiningVisualQL

A seguir (Figura 4.41), tem-se a gramática da linguagem GeoMiningVisualQL construída a partir da junção de todas as etapas vistas na seção anterior.

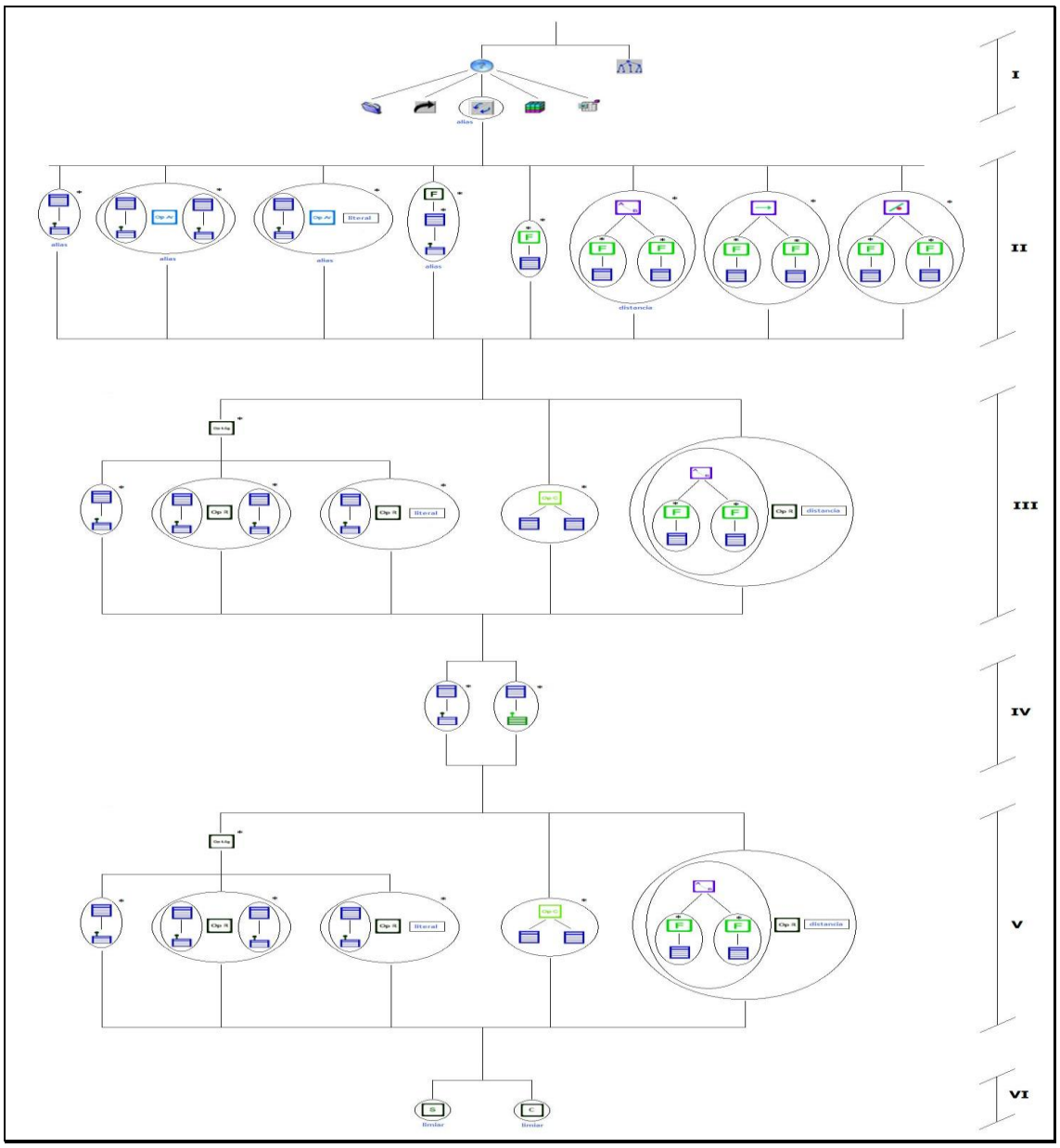


Figura 4.41: Formalismo Gramatical de parte da Linguagem GeoMiningVisualQL.

Observe que os símbolos pictográficos relativos às restrições espaciais impostas pelos predicados espaciais, os símbolos pictográficos relativos aos operadores relacionais e símbolos das funções espaciais estão generalizados, de forma a facilitar a leitura da gramática geral.

Além disso, ao se definir um determinado ambiente a ser analisado, é necessário que os elementos a ele pertencentes também sejam especificados através de símbolos visuais. No próximo capítulo, será mostrado um exemplo prático de formulação dessas consultas, fazendo-se uso de elementos (símbolos visuais) pertencentes aos metadados de um ambiente geográfico específico.

4.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Em grandes bases de dados geográficos, muitas vezes, é necessário utilizar técnicas específicas de descoberta de conhecimento, uma vez que os *Sistemas de Informações Geográficas (SIGs*) atuais apresentam certas limitações. Além disso, em sua essência, eles não possuem como característica própria, atuar frente à extração de conhecimentos.

Sendo assim, é necessário criar sistemas mais complexos, que tomem como base os *SIGs* e outras abordagens, tais como os *Ambientes Visuais de Consultas* e os *Sistemas de Mineração de Dados*.

Para isto, é necessário que se defina uma linguagem própria a qual aborde, em sua filosofia, características pertencentes a todas estas áreas, provendo meios para gerenciar qualquer tarefa de mineração sobre dados geográficos.

Assim, neste capítulo, foi definida a *Linguagem Visual de Consulta para Mineração de Dados Geográficos*, chamada ***GeoMiningVisualQL***. Inicialmente, foi feita a especificação da linguagem, baseando-se em algumas considerações com relação aos seus operadores e funções. Em seguida, foram realizadas definições formais da linguagem, restringindo-se ao uso de tarefas de regras de associação entre os dados geográficos. Além disso, foram demonstradas algumas representações pictográficas de alguns elementos pertencentes e envolvidos em um processo de mineração de dados geográfico, tais como ações, termos espaciais e não espaciais, além de funções espaciais. Por fim, foi mostrada toda a construção do formalismo gramatical da linguagem, relativo às tarefas de regras de associações entre entidades geográficas.

No próximo capítulo será apresentado o protótipo de construção de um ambiente, chamada ***GeoMiningVisual***, através do qual a linguagem *GeoMiningVisual*, apresentada anteriormente, está inserida.

CAPÍTULO 5

GeoMiningVisual – Ambiente de Consulta Visual sobre tarefas de Mineração de Dados Geográficos baseado em metáforas visuais

Neste capítulo será apresentado o projeto de construção do GeoMiningVisual, um protótipo de um Ambiente de Consulta Visual para as Tarefas de Mineração de Dados Geográficos, baseado em Metáforas Visuais. Tal sistema procura realizar, de forma transparente, a extração de conhecimento por usuários de diferentes habilidades e competências, através de uma consulta visual.

5.1 INTRODUÇÃO

Como mencionado no início deste trabalho, a mineração de dados geográficos espaciais é utilizada em diversas áreas, tais como: sistemas de sensoriamento remoto, transporte, telecomunicações, cartografia digital, entre outras. Além disso, diversos tipos de usuários podem estar envolvidos na realização de determinadas tarefas de mineração.

Em razão das formulações de consultas feitas através da linguagem GMQL apresentarem muitas dificuldades (devido ao grau de complexidade de sua sintaxe), o principal objetivo do ***GeoMiningVisual*** (Figura 5.1) é dar apoio visual aos usuários interessados em realizar processos de extração de conhecimentos sobre dados geográficos através de uma interface amigável apropriada. Neste caso, ele provê, internamente, a linguagem visual ***GeoMiningVisualQL***, a qual ajuda os usuários a configurarem determinadas tarefas de mineração sobre bases de dados geográficas, fazendo-se uso de seus elementos pictóricos na formulação de uma consulta.

Conforme [Bimonte et al. 2003], o gerenciamento das tarefas de mineração de dados espaciais se mostra bastante complexo, tendo-se, muitas vezes, a necessidade de integrar funcionalidades dos ***Sistemas de Informações Geográficas*** *(SIGs),* dos ***Sistemas de Informação de Visualizações Geográficas*** *(SIVGs)* e dos ***Sistemas Tradicionais de Mineração de Dados***.

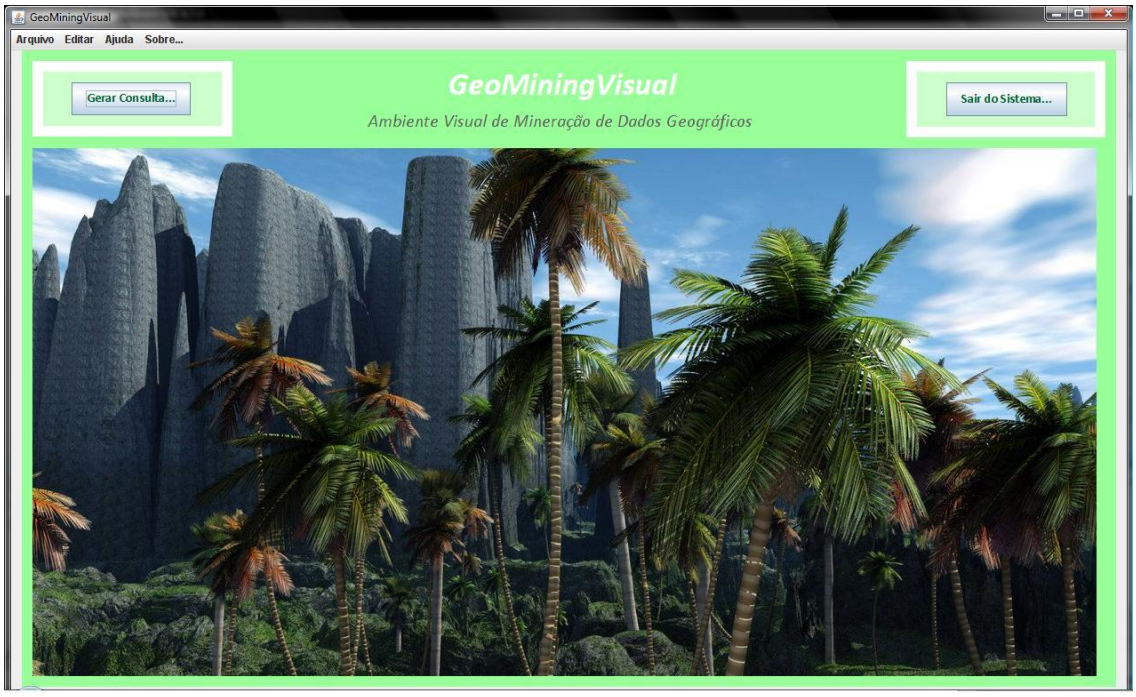


Figura 5.1: Tela Inicial do Ambiente GeoMiningVisual.
Fonte: Imagem retirada do site: http://www.ultrad.com.br (Acesso em 09/2009).

É importante ressaltar que essa integração é necessária uma vez que os *SIGs* e os *SIVGs*, de forma geral, estão associados às *características visuais dos dados*, enquanto que os *Sistemas de Mineração de Dados Tradicionais* estão vinculados a *extração de conhecimento* sobre determinadas bases de dados. Além disso, nos últimos anos, o estudo sobre o uso de linguagens visuais sobre *SIGs* tem obtido muita importância uma vez que apresentam eficiência no auxílio aos usuários perante as formulações de consultas sobre bases geográficas.

A princípio, em sua primeira versão, o *GeoMiningVisual* é um ambiente que trata apenas de parte das tarefas de regras de associação existentes entre determinadas feições geográficas que estão presentes em certos domínios geográficos. Entretanto, ele foi desenvolvido de forma a abranger, em versões futuras, diferentes outros conceitos de mineração de dados. Além disso, sua interface é baseada numa *árvore metafórica* a qual provê um caminho simples e intuitivo para a formulação de consultas associadas às tarefas de mineração sobre dados espaciais e não-espaciais, especificando suas relações e atributos envolvidos.

5.2 ABORDAGEM UTILIZADA NA FORMULAÇÃO DE UMA CONSULTA

Em relação ao processo de configuração ou formulação de uma consulta na linguagem GeoMiningVisualQL, foram adotadas algumas das estratégias definidas em [Soares 2002], tais como a ***navegação por esquema*** e a ***seleção***. A *navegação por esquema* faz uso de metadados na camada de interface, de tal modo que ajuda o usuário a selecionar elementos que farão parte da consulta. Já a estratégia de *seleção* está vinculada com a manipulação direta dos símbolos visuais da linguagem. Entretanto, foi feita uma integração destas duas estratégias unificando todo o processo de configuração através da abordagem chamada ***fluxo corrente***.

A abordagem do *fluxo corrente* está vinculada a etapas em que o usuário foca a sua atenção em configurações específicas da tarefa de mineração que está sendo submetida. Um exemplo de utilização desta abordagem será visto na Seção 5.6.5. Sendo assim, o usuário organiza os elementos envolvidos em todo o processo de mineração, mantendo uma dependência semântica e temporal entre os elementos de cada etapa deste processo.

5.3 METÁFORA VISUAL DO AMBIENTE

Uma metáfora é um processo cognitivo através do qual um determinado conceito ou termo pode ser representado ou interpretado através de outro(s) conceito(s) ou termo(s). Generalizando, é a interpretação de um conceito desconhecido ou incomum para um usuário por outros conceitos pertencentes a um determinado domínio de contexto conhecido por ele.

Sendo assim, o *GeoMiningVisual* foi desenvolvido adotando uma metáfora visual através de uma ***árvore icônica*** baseada nos termos e ações apresentados por um processo de mineração. Este conceito metafórico foi chamado de ***Árvore Icônica Metafórica***. Esta metáfora representa uma abstração dos conceitos utilizados nos processos de mineração de dados e permite a configuração de diferentes tarefas de mineração geográfica por diferentes tipos de usuários.

Na verdade, existem duas metáforas principais no ambiente. A primeira, denominada ***Árvore Icônica da Gramática***, abstrai toda a gramática da linguagem GMQL, enquanto a outra metáfora, conhecida como ***Árvore Icônica de Consulta***, abstrai

visualmente todos os passos de uma consulta que está sendo formulada pelo usuário e configurada para uma determinada tarefa de mineração. Além disso, existem metáforas visuais espalhadas pelo ambiente de tal forma que conduz o usuário a escolher relações e atributos pertencentes a determinados domínios geográficos em cada etapa do processo de mineração que está sendo construído.

Portanto, o conceito da *Árvore Icônica Metafórica* introduz no ambiente ***GeoMiningVisual*** conceitos e representações semânticas sobre todos os termos espaciais e não-espaciais envolvidos no contexto da linguagem.

5.4 METADADOS DO AMBIENTE

As informações de *metadados da linguagem* são utilizadas em um módulo de tradução com o objetivo de realizar, de forma transparente para o usuário, a conversão de uma consulta gráfica feita pelo mesmo para uma expressão equivalente sobre o script da linguagem GMQL. Além disso, os metadados também irão se referir aos domínios geográficos que podem ser especificados em documentos XML. Com isso, haverá a possibilidade, em versões futuras, de realizar o mapeamento das dependências geográficas de determinados domínios, permitindo que um conhecimento prévio pelo usuário possa ser utilizado, também de forma transparente, no momento da formalização de uma consulta, tornando, por conseguinte, a busca mais eficiente.

Na Figura 4.2, temos a ilustração de uma das etapas do processo de formulação de consultas no ambiente *GeoMiningVisual*. Neste caso, especificamente, tem-se o momento em que são escolhidos os termos espaciais e/ou não-espaciais a serem investigados numa consulta. Além disso, é possível observar alguns dos elementos citados anteriormente:

1. a ***árvore icônica metafórica*** da gramática da linguagem GMQL;

2. a ***árvore icônica metafórica*** da consulta que está sendo formulada no momento;

3. os ***metadados do ambiente*** que podem ser utilizados baseando-se nas *representações pictóricas* vistas na Seção 4.4.

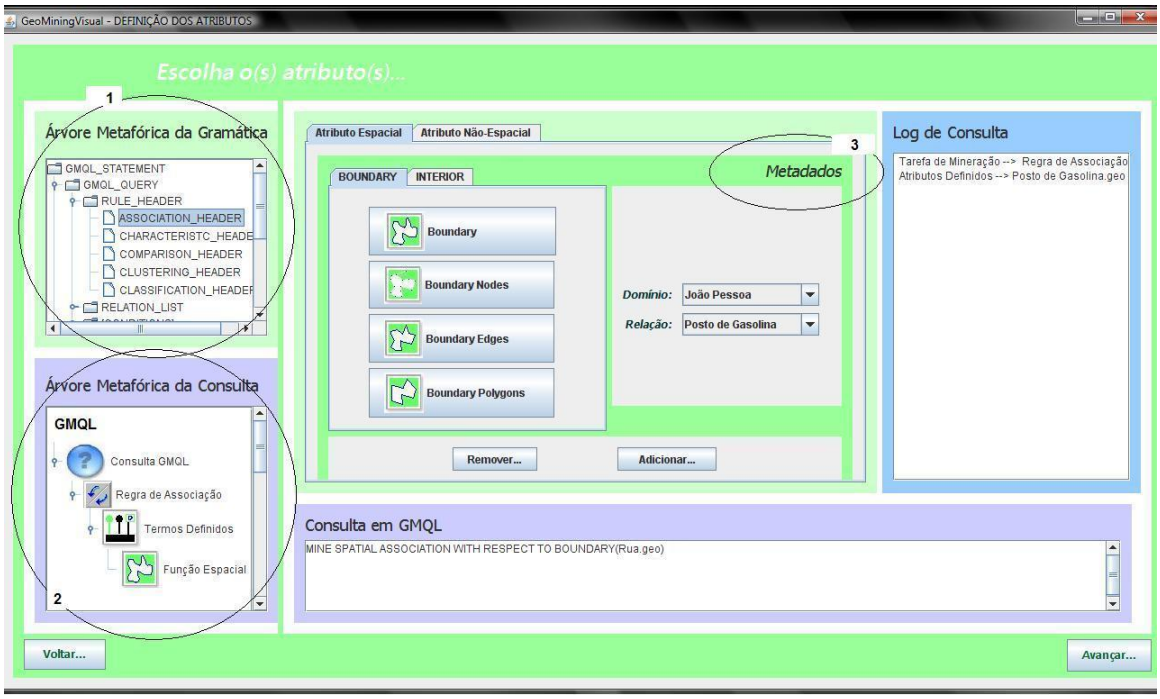


Figura 5.2: Demonstração da metáfora e metadados do ambiente GeoMiningVisual.

5.5 ARQUITETURA DO AMBIENTE

Uma vez que o objetivo principal do ***GeoMiningVisual*** é formular visualmente consultas GMQL, a concepção deste ambiente pode ser inserida dentro da filosofia da arquitetura típica de um sistema de mineração de dados, como mostra a Figura 5.3.

Após a realização de um pré-processamento sobre um conjunto de dados armazenados em algum banco, os dados são recolhidos e agrupados sob uma base de dados restrita. Associado a este processo, pode existir um conhecimento prévio armazenado em uma base de conhecimento de modo a otimizar a execução de algum algoritmo de mineração. A tarefa de mineração utilizada durante uma determinada consulta em ***GeoMiningVisuaQL*** feita por algum usuário, através da ***Camada de Interface Gráfica com Usuário (GUI)***, é que irá determinar o algoritmo de mineração a ser usado.

Esta base de conhecimento pode representar, por exemplo, certos conhecimentos até então conhecidos pelo usuário e expressos através do uso de *ontologias* ou *esquemas de banco*, como proposto em [Bogorny 2006]. Entretanto, em contribuições futuras deste trabalho, tem-se a pretensão de utilizar esta base de conhecimento na própria formulação da consulta (feita pelo usuário), tornando-a mais eficiente.

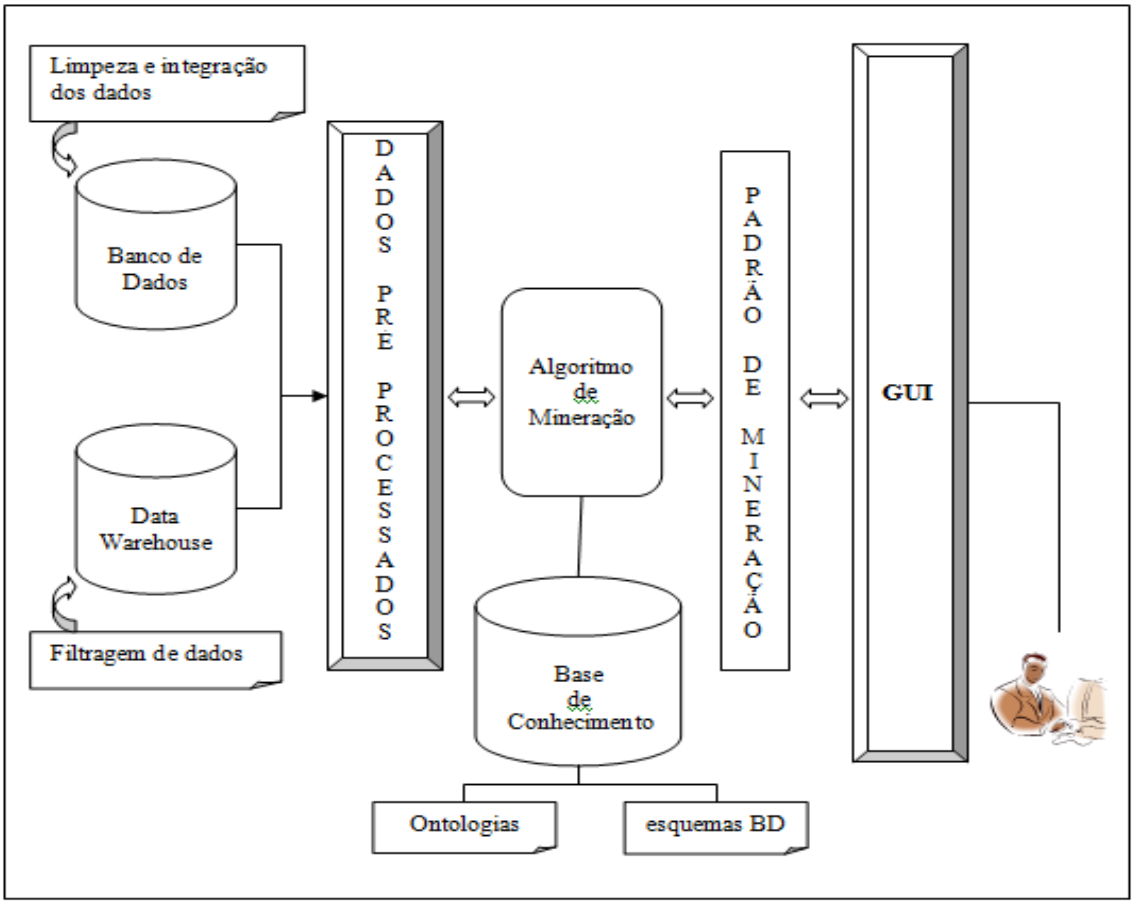


Figura 5.3: Demonstração da arquitetura de um sistema típico de mineração de dados.

Por outro lado, o ambiente definido neste trabalho também pode ser visto através de uma arquitetura que define as principais camadas de um ambiente de consultas. A Figura 5.4 ilustra as camadas existentes nesta arquitetura e os principais relacionamentos entre os módulos internos a cada uma dessas camadas.

É importante ressaltar que esta arquitetura foi definida seguindo os padrões de arquitetura que define um *Ambiente de Consulta Visual*, apresentados na *Seção 2.5.1*. Sendo assim, o projeto de construção do ambiente ***GeoMiningVisual*** está sendo desenvolvido, tendo-se como partes fundamentais: ***ambiente de interação com o usuário*** e o ***ambiente de implementação***.

O *ambiente de iteração com o usuário* foi implementado sobre a ***Camada de Interface Gráfica*** de forma a possibilitar ao usuário realizar suas consultas visuais através de elementos pictográficos.

O *ambiente de implementação* foi direcionado à definição dos módulos de tradução na ***Camada de Formulação da Consulta*** e dos metadados na ***Camada de Descrição***.

Sendo assim, tal sistema poderá utilizar diferentes modelos de dados em virtude da separação existente entre esses dois ambientes.

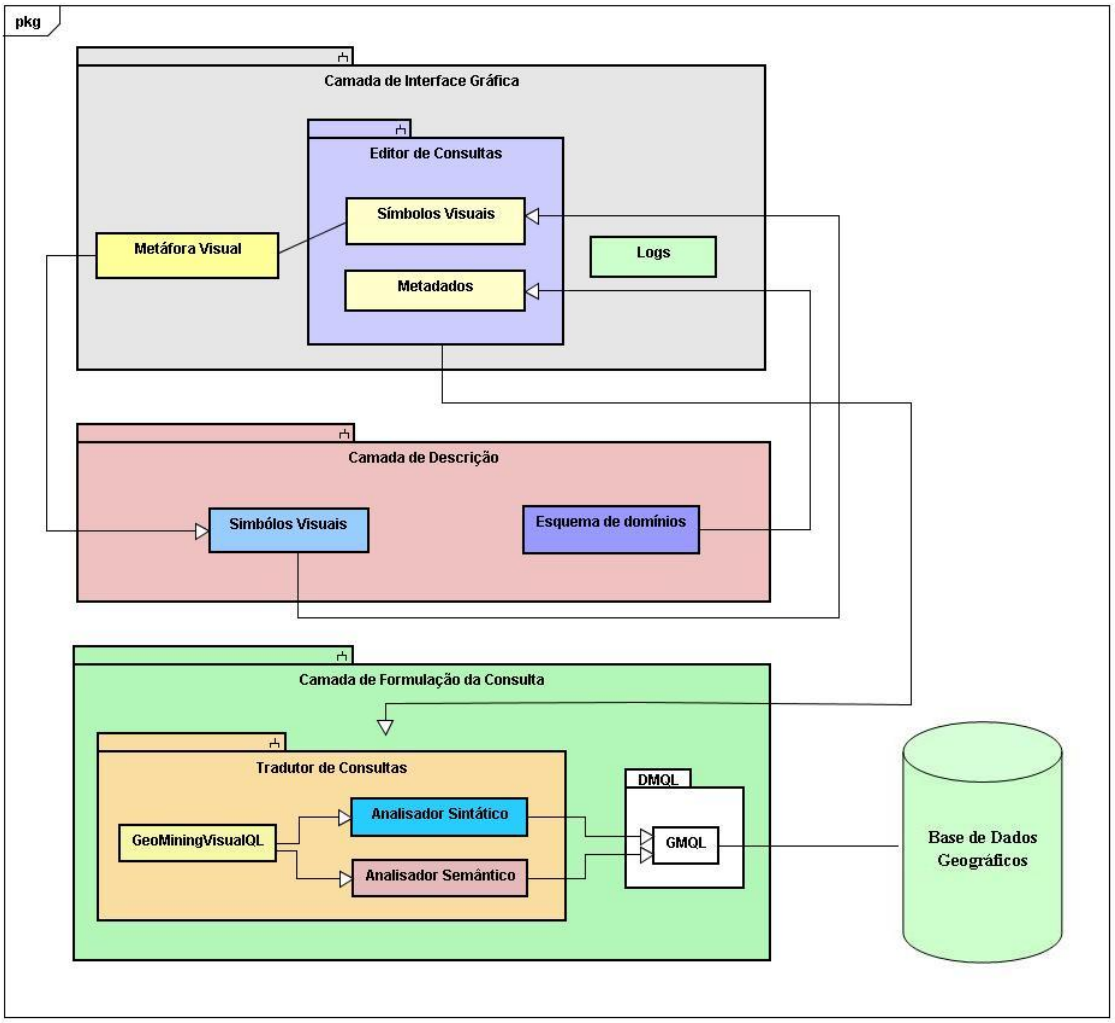


Figura 5.4: Arquitetura do sistema GeoMiningVisual.

5.5.1 Camada de Interface Gráfica

Representa a camada de mais alto nível na arquitetura do sistema ***GeoMiningVisual***. Nesse ambiente, os elementos gráficos são disponibilizados aos usuários que fazem uso do sistema.

Um conjunto de símbolos visuais está disponível para que os usuários possam formular suas consultas. Seu objetivo principal é facilitar a requisição de informações desconhecidas sobre um conjunto de dados geográficos de forma transparente ao usuário. Assim, não é necessário o usuário ter um conhecimento prévio específico da estrutura

sintática ou semântica da linguagem de mineração de dados geográficos ao formular sua consulta. Além disso, esta interface oferece mecanismos de interatividade de tal modo que ajude os usuários a compreender as informações extraídas.

Seus principais componentes são:

- ***árvore icônica metafórica da gramática GMQL*** – esta metáfora visual tem o intuito de orientar o usuário perante a formulação da consulta de mineração geográfica a ser construída.
- ***árvore icônica metafórica da consulta*** – esta metáfora visual possui o objetivo de mostrar ao usuário a representação visual corrente da consulta de mineração construída por ele.
- ***metadados*** – representações pictóricas dos elementos da linguagem *GeoMiningVisualQL* e das feições geográficas existentes nos domínio geográficos abordados pelo ambiente.
- ***log de consulta*** – processo de registro de todos os elementos envolvidos na construção da consulta: tipo de tarefa escolhida, atributos ou funções espaciais e não-espaciais, feições geográficas, relacionamentos, entre outros.

Com isso, este protótipo procura realizar a formulação de consultas visuais sob certos domínios geográficos especificados sobre os metadados da ***Camada de Descrição***.

5.5.2 Camada de Formulação da Consulta

De maneira geral, esta camada terá a definição da linguagem ***GeoMiningVisualQL*** através da especificação da sua *gramática* e a definição dos operadores espaciais que constituem a linguagem.

Além disso, esta camada contém um módulo de tradução responsável em realizar o mapeamento entre as consultas visuais feitas pelo usuário e a consulta textual feita em ***GMQL*** junto à definição de analisadores sintáticos e semânticos.

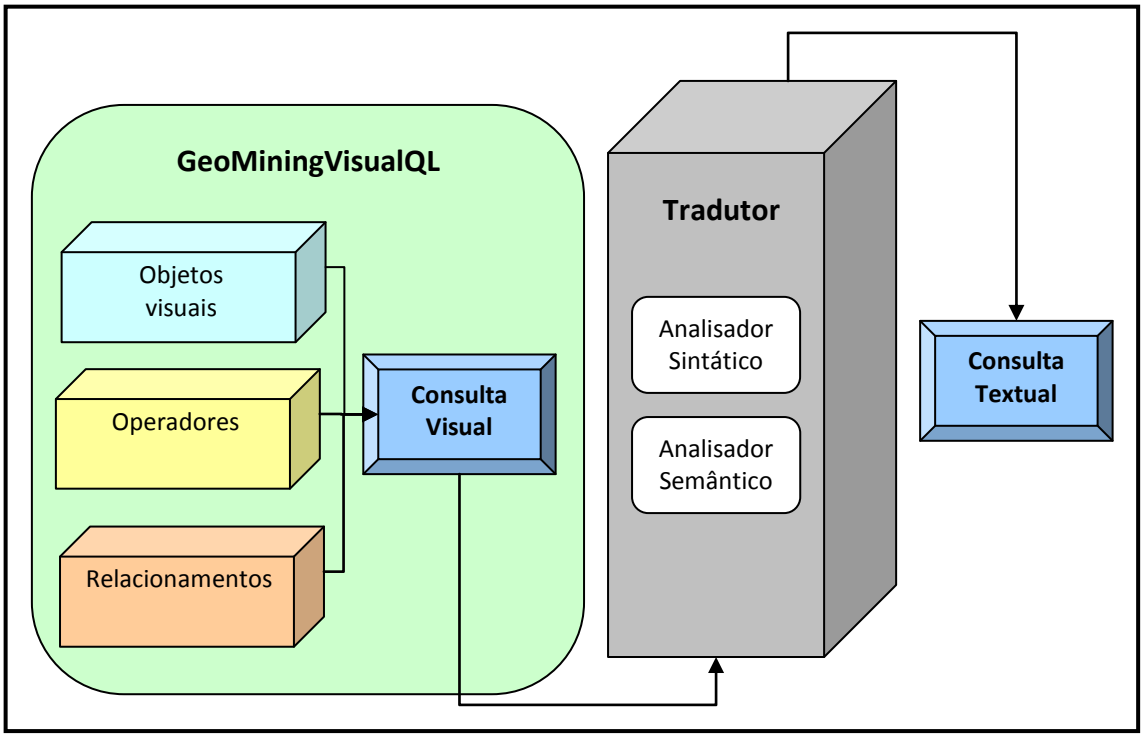


Figura 5.5: Detalhamento do Gerenciador de Consultas.

5.5.3 Camada de Descrição

Esta camada tem a importância de criar os símbolos visuais que são representações pictográficas associadas às entidades geográficas e aos operadores da linguagem definidos na Camada de Formulação da Consulta.

A Camada de Descrição é responsável pelas representações dos domínios dos ambientes geográficos especificados através de documentos XML. Além disso, estas especificações de domínio poderão ser utilizadas, em próximas versões, para registrar dependências geográficas existentes entre determinados ambientes que estão armazenados em Bases de Dados Geográficos, a fim de poder utilizá-las como conhecimentos prévios pré-existentes no momento da formulação da consulta.

5.6 IMPLEMENTAÇÃO DO SISTEMA GEOMININGVISUAL

Esta seção está voltada aos detalhes de implementação do sistema *GeoMiningVisual*. Sendo assim, serão abordadas as ferramentas e recursos utilizados pelo ambiente, além de mostrar parte de suas funcionalidades através da demonstração de um exemplo prático.

Todavia, o desenvolvimento do sistema *GeoMiningVisual* pode ser dividido em duas versões. A primeira versão foi uma tentativa de implementar a árvore da consulta baseando-se numa abordagem semelhante ao do ambiente VISMINER [Bimonte et al. 2003], utilizando-a de forma similar ao conceito de uma árvore de diretórios. Porém, com tal recurso fica inviável utilizar conceitos de recursão, uma vez que elementos de um diretório possuem a exigência de serem distintos.

Em virtude disto, alguns problemas foram detectados, sendo necessária a reformulação da consulta para a especificação atual (vista no capítulo anterior) e a projeção de uma segunda versão do ambiente. Assim, no final deste capítulo será sugerido um projeto de interface através do qual esta nova especificação da linguagem *GeoMiningVisualQL* possa ser inserida.

5.6.1 Ambiente de Desenvolvimento

A primeira versão do *GeoMiningVisual* foi desenvolvida usando a linguagem de programação JAVA. Para isto, foi utilizado o *Ambiente Integrado de Desenvolvimento* (***IDE, Integrated Develpoment Environment***) do **NetBeans**, versão 6.7.1.

A descrição dos ambientes de domínios geográficos utilizados foi projetado para que trabalhe com arquivos ***XML*** (*eXtensible Markup Language*) uma vez que, muitas vezes, são utilizados como base para a descrição de esquemas de Bancos de Dados. Com isso, a utilização, em versões futuras do sistema, de conhecimentos prévios em ambientes geográficos analisados pelo sistema se tornará mais fácil, uma vez que é possível estabelecer representações de dependências geográficas sobre esquemas de bancos de dados, conforme feito em [Bogorny 2006].

Além disso, é utilizada a API Stax para realizar todo o parser do sistema sobre os arquivos XML de descrição dos ambientes geográficos. Uma API (***Application Programming Interface)*** é um conjunto de rotinas e padrões para utilizar serviços de um software, sem saber detalhes da sua implementação, e neste caso particular, está envolvida com todo controle de leitura ou escrita de elementos pertencentes ao conteúdo de arquivos XML.

5.6.2 Funcionalidade do Sistema

Fazendo-se uso da interface deste protótipo, é possível o usuário selecionar um conjunto de símbolos pictográficos de elementos da linguagem *GeoMiningVisualQL* e feições geográficas presentes nos metadados de um determinado ambiente geográfico em estudo, formulando uma consulta gráfica, a partir de metáforas visuais, no intuito de serem submetidas para a realização de tarefas de mineração sobre dados geográficos.

Uma vez que cada elemento visual está associado à gramática da linguagem *GeoMiningVisualQL* e possui seu formalismo baseado na gramática da linguagem base, a GMQL, existe uma tradução do elemento visual para o script da linguagem base, conforme visto na definição de representações pictóricas da Seção 4.4.

Além disso, o analisador sintático da linguagem funciona realizando uma comparação entre o script de consulta gerado pela formulação da sentença visual no ambiente *GeoMiningVisual* perante um dos caminhos possíveis, gerados a partir da árvore principal da gramática da linguagem GMQL, apresentada no apêndice B deste trabalho.

5.6.3 Visão Geral da Formulação de uma Consulta no GeoMiningVisual

Analisando a Figura 5.2, é possível se ter uma visão geral de funcionamento da primeira versão do ambiente ***GeoMiningVisual*** perante a formulação de uma *sentença de consulta visual* feita por um usuário qualquer.

Inicialmente, o usuário, fazendo-se uso da *interface do ambiente*, utiliza recursos encontrados nos módulos da ***Camada de Interface Gráfica***, tal como o *Editor de Consultas*, o qual oferece símbolos visuais encontrados nos metadados do ambiente geográfico analisado e que estão contidos na ***Camada de Descrição***. Esses símbolos visuais podem ser reagrupados, junto aos símbolos dos elementos da linguagem *GeoMiningVisualQL*, através da metáfora visual da *Árvore Icônica da Consulta* que está sendo gerada. Paralelo a isso, o usuário poderá fazer uso da outra árvore metafórica a qual está vinculada com a representação da consulta feita diretamente na linguagem GMQL.

Uma vez que a formulação da consulta é feita baseada na abordagem do *fluxo corrente*, tendo o usuário focando sua atenção em diferentes etapas durante todo o processo de extração de conhecimento da tarefa de mineração que está sendo criada na consulta, a *Camada de Interface Gráfica* irá dispor de diferentes interfaces para cada etapa deste processo. Todavia, cada uma dessas etapas possui elementos distintos. Além disso, é

possível registrar todo o *controle semântico* encontrado nos metadados, através de registros de logs da consulta que está sendo formulada até o momento.

Por fim, a consulta visual gerada pelo editor de consultas da *Camada de Interface Gráfica* irá ser submetida à ***Camada de Formulação da Consulta*** a qual irá gerar o script textual da consulta que foi traduzido através de representações pictóricas contidas nos metadados. Em seguida, o *analisador sintático* irá validar o script da consulta com os possíveis scripts gerados pela árvore da gramática GMQL, a partir de caminhos válidos oriundos de um algoritmo de processamento sobre os terminais da árvore da gramática GMQL.

Sendo assim, a consulta final, totalmente escrita na linguagem GMQL, poderá ser utilizada para realizar o processamento de qualquer tarefa de mineração sobre bases de dados geográficas. Para isto, em versões futuras do sistema *GeoMiningVisual*, será realizada a vinculação do ambiente com algum sistema que realiza a busca, o processamento e as visualizações dos dados retornados por uma consulta feita GMQL, tal como o *GeoMiner* [Han, Koperski e Stenfavic 1997]. Todavia, será necessário que sistemas destes tipos forneçam o código fonte para os usuários, uma vez que eles são estritamente comerciais.

5.6.4 Configuração de uma consulta no GeoMiningVisual

A configuração de uma consulta a ser feita pelo ambiente *GeoMiningVisual* é realizada através das primitivas básicas de construção de uma linguagem de mineração de dados geral.

Com isso, cada etapa do processo de descoberta de conhecimento a ser definido no ambiente (o qual está associada às diferentes interfaces que a Camada de Interface Gráfica poderá dispor aos usuários) está vinculada a estas primitivas básicas. Em conseqüência disso, estas interfaces definem a abordagem do *fluxo corrente* utilizada no sistema.

5.6.4.1 Configuração dos dados relevantes na tarefa de mineração

Uma vez que é possível existir diversas bases de dados geográficas a serem escolhidas no ambiente, o usuário irá selecionar um determinado domínio a ser utilizado pelo sistema.

Esta escolha é baseada nos diversos esquemas de bancos de dados disponíveis pela aplicação, através de arquivos XML contidos nos metadados da *Camada de Descrição* do ambiente *GeoMiningVisual*.

5.6.4.2 Configuração do tipo de conhecimento a ser minerado

Ao iniciar a formulação de uma consulta no ambiente GeoMiningVisual, o usuário terá a disposição todas as tarefas de mineração associadas a ambientes geográficos. Todavia, por limitação de tempo, apenas parte da tarefa de regra de associação entre as entidades geográficas presentes em um domínio qualquer está implementada neste protótipo.

Uma vez que a linguagem GMQL permite definir alias para alguns termos ao longo da consulta, o usuário poderá definir um valor literal (neste exemplo: "minhaPrimeiraConsulta") para a tarefa de regra de associação que está sendo formulada. Uma visão geral desta primeira etapa de configuração da consulta pode ser vista na Figura 5.6.

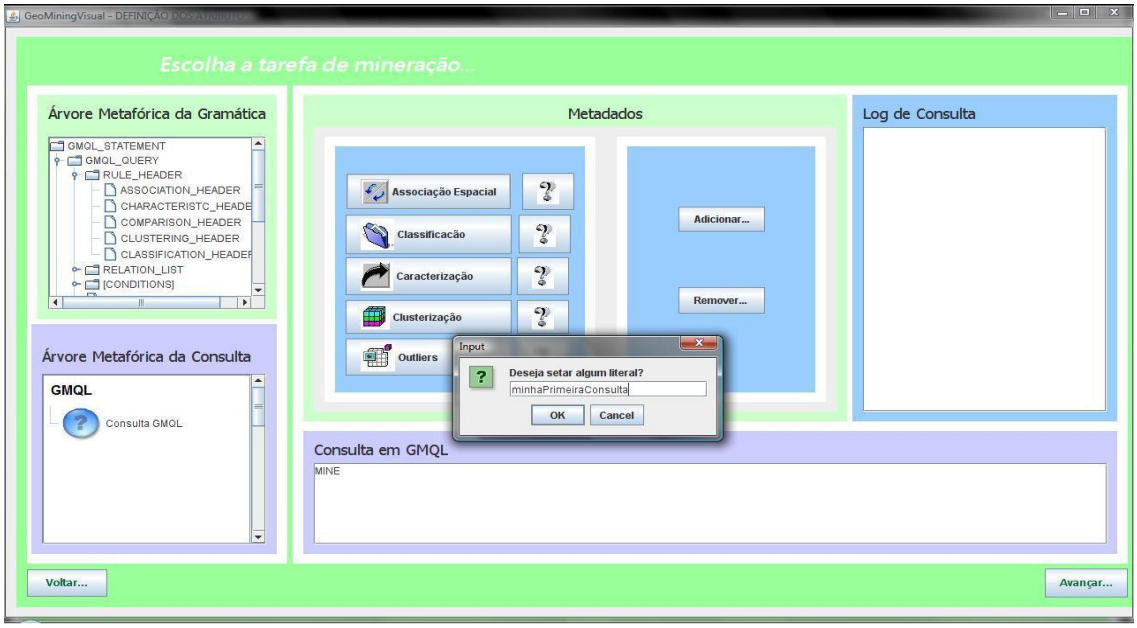


Figura 5.6: Momento em que o usuário configura uma tarefa de regra de associação no ambiente.

Vale lembrar que caso seja escolhido algum outro tipo de tarefa, o sistema irá reportar uma mensagem informado que apenas em versões posteriores irá ser possível realizar configurações de consulta com estas outras tarefas.

Por outro lado, uma vez selecionada a tarefa de regra de associação, será mostrado na parte inferior do sistema, "***Consulta em GMQL***", o script gerado pela tradução da escolha do símbolo visual associado a esta tarefa.

Assim, caso o usuário tenha adicionado a tarefa de regra de associação na consulta com o literal da figura 5.6, tem-se o script GMQL mostrado na Figura 5.7.

Figura 5.7: Script gerado junto a definição de um valor literal para a tarefa de regra de associação.

Caso não se tenha definido um literal para a tarefa, será criado o script default para a realização deste tipo de tarefa, como mostra a Figura 5.8.

Figura 5.8: Script padrão gerado junto a escolha de uma tarefa de regra de associação.

5.6.4.3 Configuração do conhecimento sobre o contexto do domínio a ser minerado

Esta etapa é uma das mais importantes na configuração ou formulação da consulta visual, uma vez que são selecionados atributos simples ou geográficos os quais podem estar envolvidos com diversos elementos, tais como os *relacionamentos* (topológicos, de distância ou métricos), *funções* (espaciais ou não-espaciais) e *operadores gerais* da linguagem *GeoMiningVisualQL*.

Partindo do pressuposto que esteja sendo analisado um ambiente geográfico do estado da Paraíba e dando seguimento ao exemplo da etapa de configuração da seção anterior, suponha que o usuário selecione, na etapa atual, uma função espacial que delimita a fronteira sobre um atributo geográfico no contexto do ambiente analisado, utilizando o atributo geográfico posto_gasolina na relação Joao_Pessoa, conforme mostra a Figura 4.9. É fácil ver que a *Árvore Metafórica da Gramática* serve apenas de base para o usuário percorrer os caminhos possíveis ao formular uma consulta na linguagem GMQL. Por outro

lado, a *Árvore Metafórica da Consulta* mostra ao usuário a consulta visual que está sendo formulada pelo ambiente.

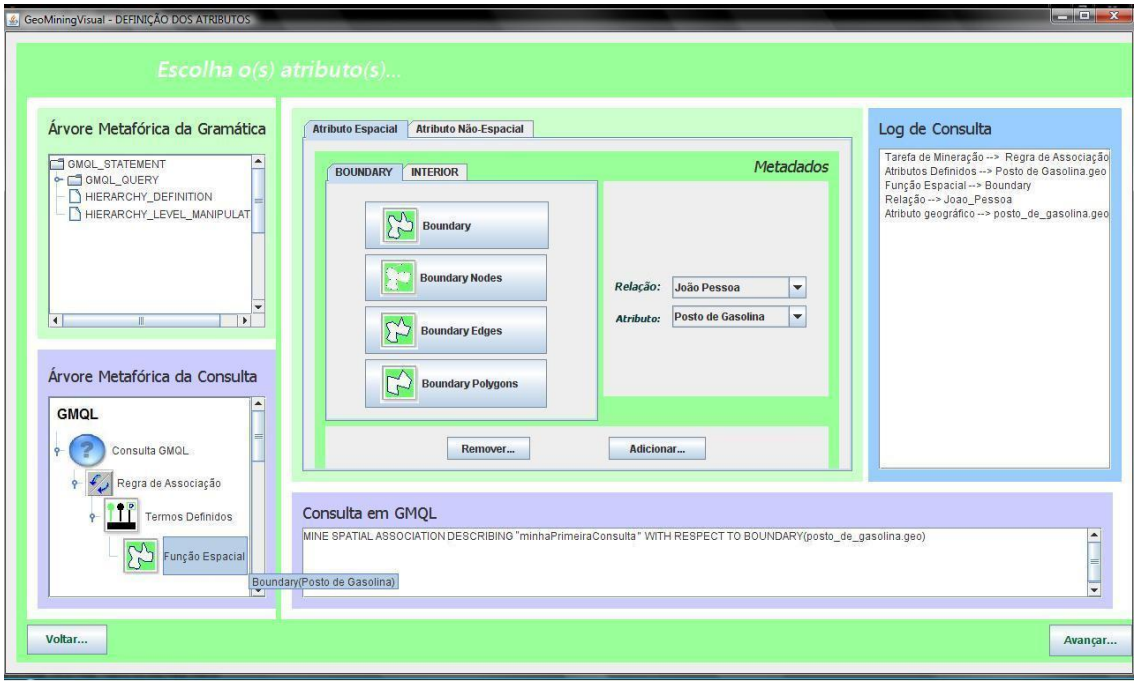


Figura 5.9: Etapa de seleção de atributos gerais do ambiente.

Observe que o símbolo visual  não é utilizado na especificação atual da linguagem. Ele foi utilizado apenas nesta primeira versão do sistema (versão antiga) no intuito de especificar a utilização de algum termo (espacial ou não) na consulta que está sendo formulada.

O log de consulta encontrado na Figura 5.9 tem a função de mostrar, ao usuário, informações úteis a respeito dos elementos até então envolvidos com a consulta visual que está sendo formulada, desde a sua primeira etapa de configuração. Neste exemplo, ele dispõe de algumas informações, tais como a tarefa de mineração selecionada pelo usuário na primeira etapa de formulação da consulta, além de atributos e funções espaciais.

Um elemento importante na interface da Figura 5.9 é a consulta visual formulada pelo usuário, a qual pode ser expressa através da *Árvore Metafórica da Consulta*. Por outro lado, vale ressaltar que esta consulta poderia ser estendida, adicionando outras primitivas básicas, em momentos distintos, as quais poderiam fazer uso de restrições condicionais através de operadores lógicos e relacionais, além da utilização de parâmetros que controlam os padrões retornados por uma extração de conhecimento.

No entanto, devido às limitações do conceito de uma árvore de diretórios através do qual esta versão do sistema foi implementada e ajustes feitos na nova especificação da linguagem, foi necessário elaborar um novo projeto de construção de uma interface ideal para o ambiente no qual todos os recursos da linguagem GeoMiningVisualQL (especificada no capítulo anterior) pudessem ser inseridos.

5.6.5 Sugestão de uma nova interface para o ambiente GeoMiningVisual

Uma vez que a filosofia de um ambiente de consulta visual, vinculado aos recursos oferecidos pela especificação da linguagem *GeoMiningVisualQL*, oferece vários recursos tais como metáforas visuais, metadados, abordagem de navegação sobre elementos visuais da consulta e etapas de configuração relacionadas às primitivas básicas de construção de uma linguagem de mineração de dados, é necessário projetar uma interface a ser usada pelo mesmo.

Neste caso, esta interface tem como objetivo permitir o usuário fazer uso de todos os recursos de forma mais simples e direta, provendo todas as funcionalidades da consulta, uma vez que algumas delas foram limitadas na versão anterior, tal como a recursão de elementos em uma mesma consulta.

Sendo assim, a Figura 5.10 ilustra um padrão de interface geral que pode ser utilizado pelo ambiente.

Neste caso, o usuário dispõe de vários símbolos visuais especificados na linguagem e dispostos por etapas de configurações específicas, além dos símbolos presentes nos metadados de descrição dos ambientes geográficos analisados. Com isso, o usuário “arrasta” os símbolos para dentro da área de construção da árvore metafórica, a qual gera os scripts textuais da linguagem base GMQL, através de um processo de tradução de cada ícone visual.

Além disso, a consulta visual pode ser navegada através de cada uma das etapas de navegação (I, II, III, IV, V e VI), tendo-se um destaque do *fluxo corrente* de configuração de uma etapa específica da consulta.

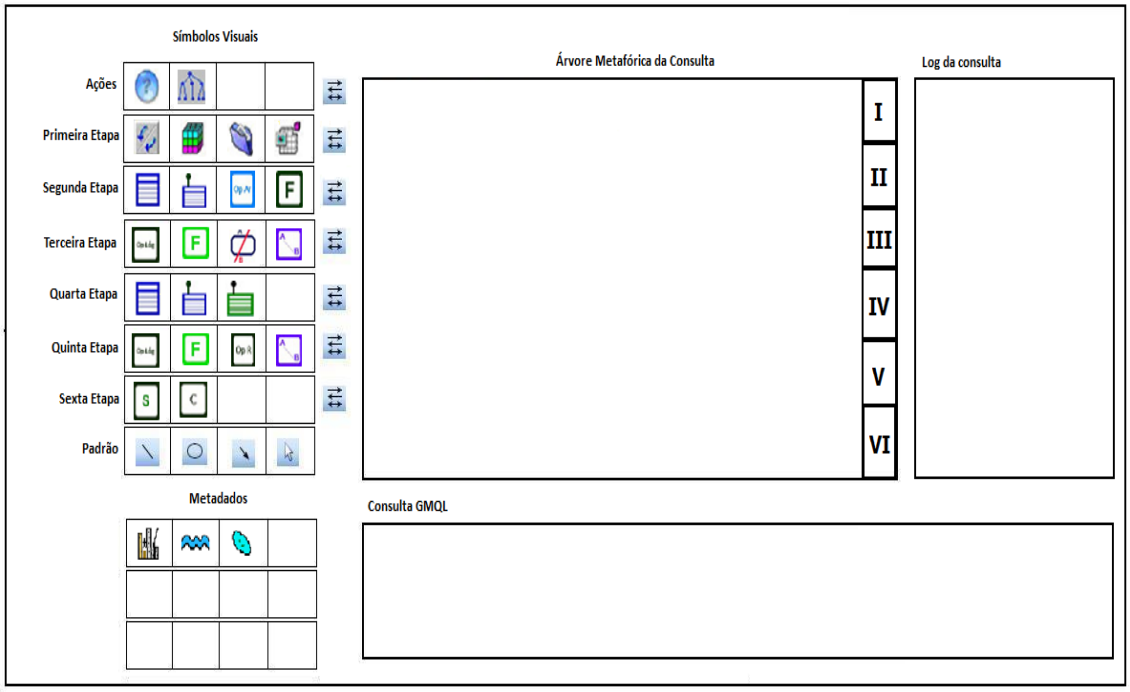


Figura 5.10: Proposta de nova interface para o ambiente GeoMiningVisual.

5.6.5.1 Consulta GMQL

A seguir, será demonstrado um exemplo hipotético de como pode ser feita a formulação de uma consulta visual no ambiente *GeoMiningVisual*, baseando-se no padrão de interface utilizado na Figura 5.10.

Neste caso, considere o fato de que um usuário, utilizando-se de uma base de dados específica, esteja interessado em saber os nomes de todas as cidades que são interceptadas por rios e possui estações ferroviárias. Além dos nomes destas cidades, devem ser retornados os nomes dos rios que as interceptam. Por fim, o resultado da consulta deve ser agrupado pela região que cada rio pertence. A Figura 5.11 ilustra o script de consulta GMQL referente a este exemplo.

```
MINE SPATIAL ASSOCIATION WITH RESPECT TO
Cidade.nome, Rio.nome FROM Cidade, Rio, Estacao
WHERE Cidade.geo INTERSECST Rio.geo AND
Cidade.geo CONTAINS Estacao.geo  GROUP BY
Rio.regiao
```

Figura 5.11: Exemplo de uma consulta em GMQL para uma tarefa de regra de associação espacial.

5.6.5.2 Execução da consulta no ambiente GeoMiningVisual

Como visto anteriormente, é possível construirmos a consulta no ambiente *GeoMiningVisual* através da separação, por etapas, de funções específicas existentes na execução do algoritmo de mineração.

Baseando-se nos metadados de alguns símbolos visuais definidos para um ambiente geográfico descrito por [Soares 2002], será formulada uma consulta visual correspondente à consulta em GMQL feita na Figura 5.11. A especificação de alguns destes símbolos pode ser vista nas Tabelas 5.1, 5.2 e 5.3.

Tabela 5.1: Representação geral da feição geográfica correspondente a uma cidade.

Cidade	*Especifica uma cidade qualquer que esteja presente no ambiente de domínio analisado.*
	*Tradução*
	O nome atribuído pelo usuário na consulta.

Tabela 5.2: Representação geral da feição geográfica correspondente a um rio.

Rio	*Especifica um rio qualquer que esteja presente no ambiente de domínio analisado.*
	*Tradução*
	O nome atribuído pelo usuário na consulta.

Tabela 5.3: Representação geral da feição geográfica correspondente a um lago.

Estação Ferroviária	*Especifica uma estação ferroviária qualquer que esteja presente no ambiente de domínio analisado.*
	*Tradução*
	O nome atribuído pelo usuário na consulta.

Assim, inicialmente, após definir o início de uma consulta através do símbolo  , utiliza-se o ambiente de domínio o qual se deseja analisar. Neste caso, a interface deverá dispor de uma seleção dos diversos ambientes de domínio presentes nos metadados. Com isso, ao selecionar uma determinada base de dados, o sistema faz uma varredura nos

metadados dessa base, carregando todos os elementos visuais a ele pertencentes (Figura 5.12).

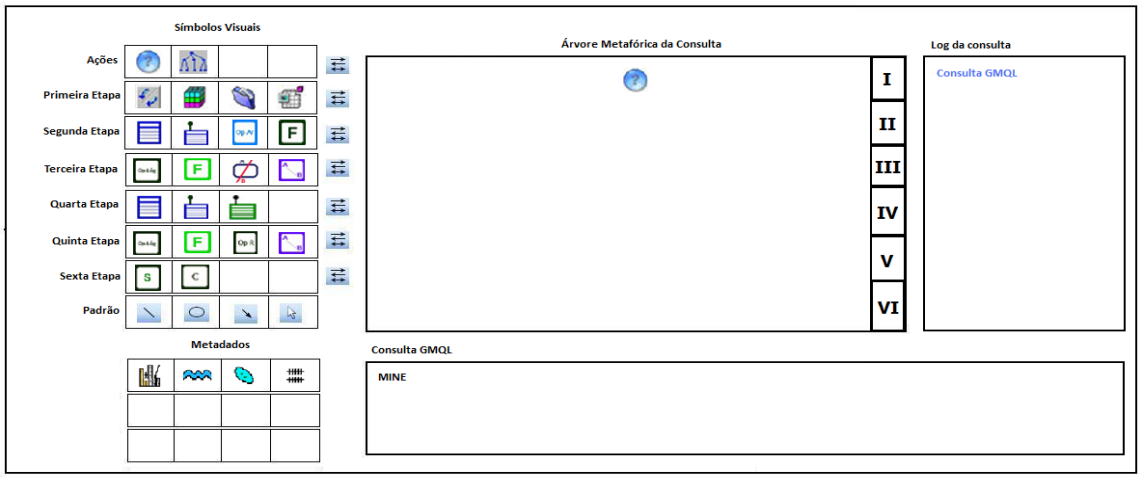


Figura 5.12: Definição de uma consulta a ser utilizada no ambiente.

Uma vez que o ambiente é definido, o usuário inicia sua consulta escolhendo um símbolo de uma tarefa de mineração qualquer. Para isto o usuário clica sobre a tarefa escolhida e, em seguida, arrasta a tarefa para a área responsável pela construção da árvore metafórica, interligando-o ao símbolo  , responsável pela definição de uma consulta visual da linguagem *GeoMiningVisualQL*. Uma vez que a consulta está sendo construída fazendo-se uso de um relacionamento espacial entre determinadas entidades geográficas (neste caso, cidades, rios e estações ferroviárias), o símbolo de uma tarefa de regra de associação  é escolhido, conforme ilustra a Figura 5.13. Observe que, neste caso, o fluxo corrente é demarcado através de uma tonalidade sobre o número correspondente à etapa atual de configuração.

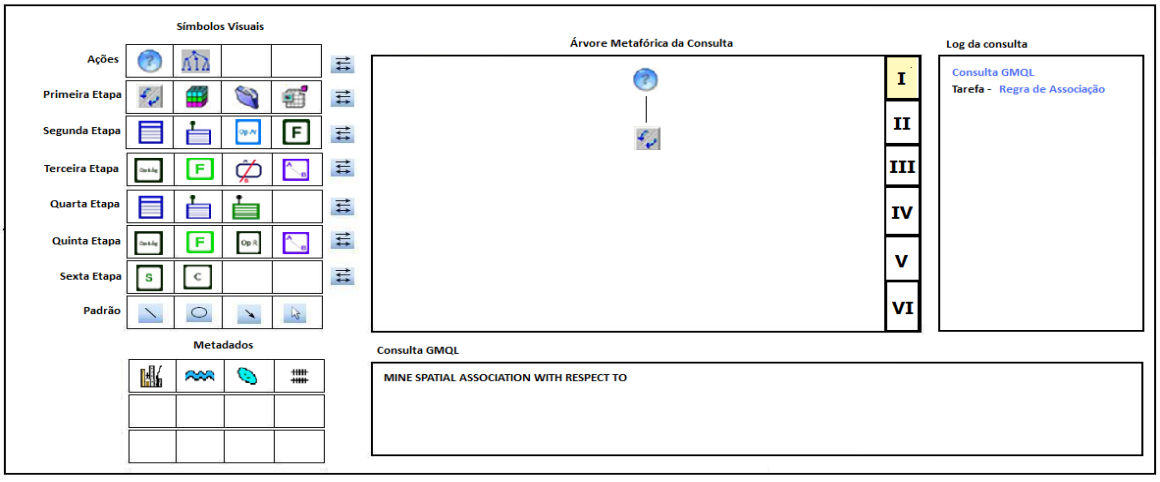


Figura 5.13: Configuração da primeira etapa: seleção da tarefa de mineração a ser realizada no ambiente.

O próximo passo será a seleção de termos espaciais ou não-espaciais a serem investigados na consulta. Neste caso, o usuário deseja recuperar nomes de rios e nomes de cidades. Como estes atributos são simples, o usuário faz uso de termos definidos na Figura 4.3 da Seção 4.5.1.2 do capítulo anterior. Neste caso, tomando-se o mesmo raciocínio da etapa anterior, o usuário seleciona os símbolos envolvidos nesta definição e os arrastam para a árvore metafórica que está sendo construída. Todavia, nesta configuração é necessário fazer uso do símbolo  o qual delimita um termo específico na consulta (Figura 5.14).

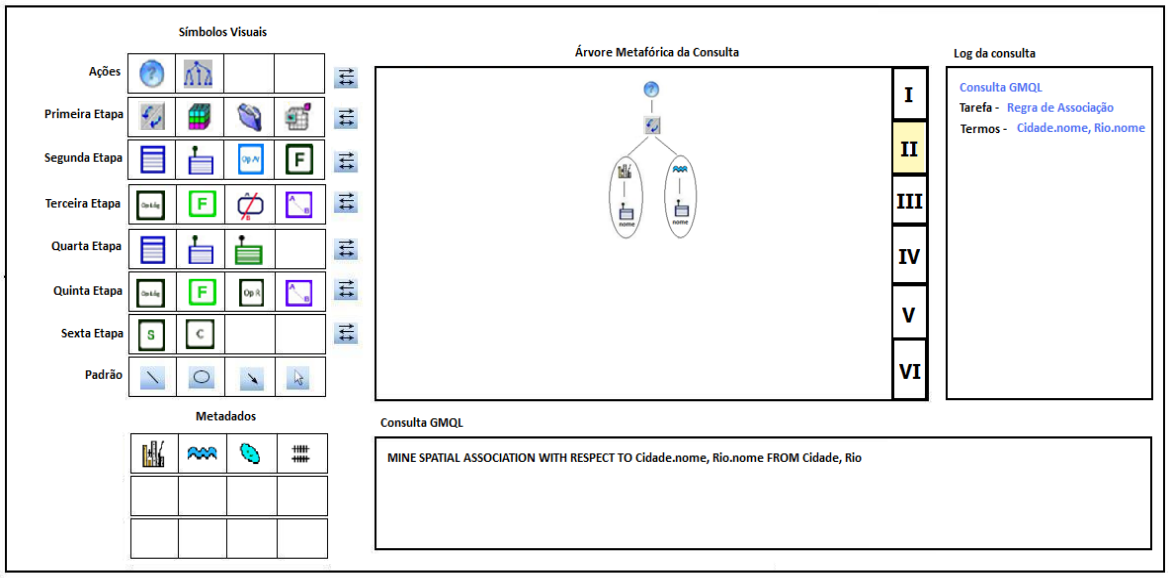


Figura 5.14: Configuração da segunda etapa: seleção de termos espaciais e não-espaciais a serem investigados na consulta.

Feito isso, a próxima etapa ou fluxo de configuração da consulta referente às restrições condicionais a consulta é iniciado, fazendo uso de relacionamentos espaciais existente entre as feições geográficas *Cidades* e *Rios* e as feições geográficas *Cidades* e *Estações Ferroviárias*. Neste caso, especificamente, tratam-se dos operadores geográficos que indicam operações de interseção  e inclusão  entre duas entidades geográficas, através do operador lógico  . Assim, a consulta visual é configurada da forma como ilustra a Figura 5.15.

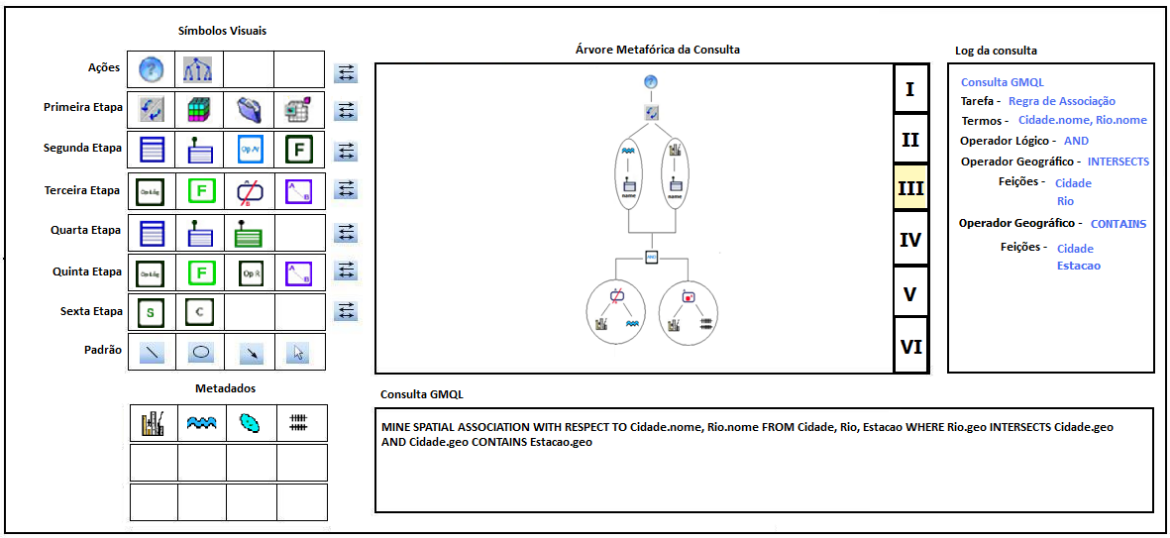


Figura 5.15: Configuração da terceira etapa: aplicação de restrições condicionais sobre o resultado de uma consulta.

Por fim, para completar a consulta requisitada pelo usuário, é necessário reagrupar os registros retornados pela consulta através de regiões aos quais as cidades são localizadas. Assim, o resultado é reordenado segundo o agrupamento feito nesta etapa. Com isso, a consulta é finalizada, conforme ilustra a Figura 5.16.

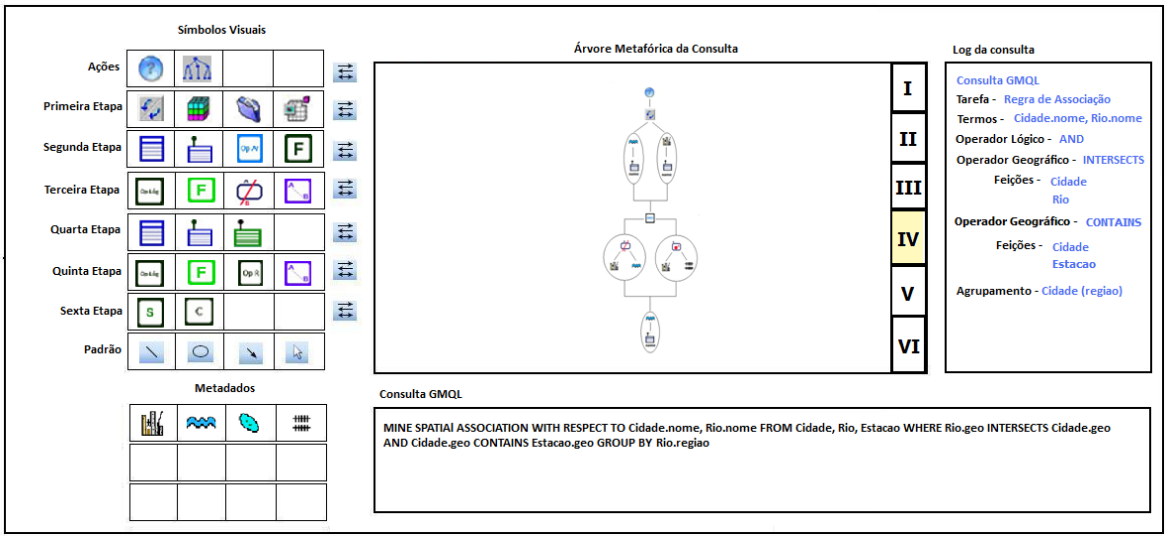


Figura 5.16: Configuração da quarta etapa: agrupamento de registros retornados.

É importante ressaltar que, caso seja necessário utilizar restrições condicionais sobre o agrupamento de uma consulta ou fazer uso de parâmetros para controlar o número de padrões retornados, o usuário poderia fazer uso das etapas de configuração V e VI. Todavia, neste exemplo não é necessário, sendo, portanto finalizada a consulta.

5.7 FUNCIONAMENTO DO ANALISADOR SINTÁTICO

Uma vez que o propósito do ambiente ***GeoMiningVisual*** é oferecer scripts textuais na linguagem GMQL gerados pelas consultas visuais, é necessário haver um mecanismo de validação da consulta. Em outras palavras, deve haver um módulo responsável por realizar uma varredura sintática sobre o script textual gerado pela consulta formulada pelo usuário.

Este módulo, conhecido com ***analisar sintático***, deve possuir uma função especifica, e neste caso, realizar um processamento sobre a árvore principal da gramática GMQL, de forma a se ter todos os caminhos possíveis para a formulação de uma consulta válida.

Com isso, basta apenas fazer uma comparação entre o script gerado pela consulta visual do ambiente com estes caminhos possíveis para saber se a consulta está formalmente válida. Todavia, vale ressaltar que a gramática da linguagem *GeoMiningVisualQL*, utilizada no ambiente, foi baseada na gramática da linguagem GMQL, tendo, por conseqüência, a tradução de um símbolo visual para um script válido da linguagem GMQL.

5.8 ANÁLISE COMPARATIVA

Os trabalhos relacionados apresentados no capítulo 3 ressaltam a importância deste estudo uma vez que ele procura unificar diversas áreas de pesquisas (*Bancos de Dados Geográficos*, *Processos de Descobertas de Conhecimento* e *Linguagens de Consultas Visuais)* através de uma interface visual para descoberta de conhecimento em ambientes geográficos.

A seguir, será apresentado um quadro comparativo do ambiente ***GeoMiningVisual***, o qual insere a linguagem ***GeoMiningVisualQL*** proposta neste trabalho, em relação aos principais trabalhos encontrados e discutidos na literatura. Esta análise será feita de acordo com as características, vantagens e limitações de cada um desses ambientes de consultas visuais.

A Tabela 5.4 apresentada as principais características pertencentes ao ***GeoMiningVisual***.

Tabela 5.4: Principais características do Ambiente de Consulta GeoMiningVisual.

GeoMiningVisual	
Escopo	Realização de consultas visuais sobre processos de mineração de dados geográficos.
Interface	Ícones e metadados que facilitam o entendimento do domínio analisado
Operadores	Operadores de predicados espaciais (topológicos, métricos e de distância), aritméticos, booleanos e relacionais.
Elementos	Tarefas ou Ações de Mineração, Feições Geográficas, operadores espaciais e não-espaciais, relacionamentos
Abordagem de Representação	Fluxo Corrente.
Metáfora	Árvore Icônica
Linguagem Base	GMQL
Resultado	Script de Consultas na Linguagem GMQL

A seguir, na Tabela 5.5, será feita uma comparação do ambiente ***GeoMiningVisual*** com alguns trabalhos existentes e relevantes na literatura. Para isto, são considerados alguns parâmetros os quais comparam a existência ou não de determinadas funcionalidades ou características que podem estar presentes nesses ambientes.

Apesar de ser um ambiente utilizado para a realização de tarefas de mineração de dados geográficos, não foi possível realizar certos tipos de estudos no ambiente ***GeoMiner,*** em virtude de ser um software estritamente comercial. Além disso, ele não apresenta características visuais, tendo as formulações de suas consultas feitas através de script textuais da linguagem *GMQL*.

Por outro lado, o ambiente ***GeoVisual*** possui muitas características semelhantes ao do ambiente ***GeoMiningVisual***. Na verdade, o *GeoMiningVisual* pode ser considerado

uma extensão do *GeoVisual*, uma vez que a filosofia do projeto de construção de sua arquitetura foi baseada na arquitetura do *GeoVisual*.

Todavia, o *GeoVisual* não apresenta recursos ou funcionalidades utilizadas em processos de descoberta de conhecimento. Sendo assim, foram inseridas, na linguagem interna utilizada no ambiente *GeoMiningVisual*, técnicas de descoberta de conhecimento durante uma consulta realizada sobre determinados domínios geográficos.

Tabela 5.5: Análise Comparativa do GeoMiningVisual em relação aos trabalhos vistos neste capítulo.

Ambientes de Consultas Visuais	*Representação Visual do Esquema*	*Linguagem de Consulta Visual*	*Linguagem de Mineração de Dados*	*Linguagem de Mineração de Dados Geográficos*	*Linguagem de Consulta Interna*	*Elementos pictográficos*	*Operadores Lógicos*	*Operadores Aritméticos e Relacionais*	*Operadores Espaciais*	*Linguagem Base - GMQL*	*Metáforas Visuais*
GeoMiner	*X*	*X*	*A*	*A*	*X*	*X*	*A*	*A*	*A*	*A*	*X*
GeoVisual	*A*	*A*	*X*	*X*	*A*	*A*	*A*	*A*	*A*	*X*	*A*
VisMiner	*A*	*A*	*A*	*A*	*N*	*N*	*N*	*N*	*N*	*A*	*A*
GeoMiningVisual	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*

Legenda { ***A**: apresenta a característica*; ***X**: não apresenta a característica*; ***N**: não foi possível avaliar*

No caso do ambiente **VisMiner**, tem-se um protótipo de um ambiente de mineração de dados espaciais, porém não existem trabalhos consolidados na literatura de tal modo que não se permita fazer uma analise mais detalhada de todas as funcionalidades e recursos envolvidos neste ambiente.

Sendo assim, após essa análise, o ambiente ***GeoMiningVisual*** se destaca, principalmente, por apresentar internamente a unificação de um linguagem de consulta visual, uma linguagem de mineração de dados e a abordagem de contextos geográficos.

Além disso, utiliza informações de metadados em sua tradução, representando todo o esquema do ambiente geográfico e elementos da linguagem através de consultas visuais.

5.9 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou o ***GeoMiningVisual***, um protótipo que possui, em sua essência, a ideologia básica de construção de um *ambiente visual* que trata a busca de novos conhecimentos sobre bases de dados geográficas. Em sua filosofia, o ambiente *GeoMiningVisual* possui, internamente, a linguagem de consulta visual ***GeoMininVisualQL***, definida no capítulo anterior deste trabalho.

Inicialmente foi mostrada a definição da abordagem de *fluxo corrente* ao qual é utilizada na configuração de uma consulta durante um processo de descoberta de conhecimento no ambiente. Com esta abordagem, a formulação da consulta pode ser decompostas em diferentes etapas com objetivos específicos, facilitando o entendimento da construção da consulta como um todo por parte do usuário.

Em seguida, foi apresentada a arquitetura sobre a qual o sistema *GeoMiningVisual* foi construído. Nesta arquitetura existem três camadas as quais possuem funções distintas e isoladas, de tal forma que permite ao sistema a inserção de recursos para automatizar o processo de descoberta, tal como o uso de *conhecimentos semânticos prévios* nos contextos de determinados domínios geográficos.

Posteriormente, foi feita uma breve discussão a respeito das tecnologias e ferramentas utilizadas para o projeto e construção da primeira versão deste *ambiente visual de consulta*, mostrando, também, a forma como este sistema foi projetado. Além disso, foram inseridas novas sugestões para a implementação de versões futuras do mesmo, principalmente em relação à interface de utilização do ambiente, o parser dos metadados contidos em arquivos XML e a utilização, em princípio, de conhecimentos semânticos prévios na própria consulta visual através de esquemas da base dados geográficos contidos nos metadados da sistema.

Por fim, foi mostrado, através de um exemplo prático, como pode ser feita a configuração das diferentes etapas do processo de descoberta de conhecimento utilizado pelo ambiente a fim de realizar a formulação de uma consulta visual. Vale ressaltar que este projeto é apenas um protótipo o qual ilustra a idéia a ser utilizada nas formulações de consultas visuais nas tarefas de mineração sobre bases de dados geográficas.

No próximo capítulo será apresentada a conclusão deste trabalho junto às suas contribuições para os processos de descoberta de conhecimento sobre bases de dados geográficos. Além disso, serão destacadas as principais dificuldades encontradas para a elaboração desta pesquisa, além de estudos e trabalhos futuros a serem realizados.

CAPÍTULO 6

CONCLUSÃO E TRABALHOS FUTUROS

6.1 RESUMO

Neste trabalho foi apresentada a construção de uma *Linguagem de Consulta Visual* para as tarefas de *Mineração de Dados Geográficos*, denominada ***GeoMiningVisualQL,*** em virtude de ser uma extensão da ***GeoVisualQL*** proposta por [Soares 2002].

Para isso, foram realizadas pesquisas sobre projetos de construções de *Linguagens de Mineração de Dados Geográficos* e *Linguagens de Consultas Visuais*, além de estudos aprofundados sobre as formas de representação de *feições geográficas* e sobre os padrões de arquitetura dos *Ambientes de Consultas Visuais*.

Em seguida, foi definida uma arquitetura de um ambiente visual no qual a linguagem ***GeoMiningVisualQL*** foi inserida, de acordo com os *padrões* de ambientes visuais estudados. Em sua filosofia, a linguagem **GeoMiningVisuaQL** procura realizar a formulação de consultas visuais em um ambiente de interface gráfica, facilitando a construção das consultas feitas pelos usuários através de símbolos visuais definidos na própria linguagem.

6.2 DIFICULDADES ENCONTRADAS

O protótipo desenvolvido para criar um ambiente para a realização de consultas visuais para mineração de dados geográficos não pôde ser finalizado pelas seguintes razões:

- A dificuldade na elaboração de consultas em GMQL foi maior do que a esperada, mesmo que apenas a regra de associação tenha sido utilizada;
- Houve uma grande dificuldade também em abranger todas as variações de uma possível consulta em GMQL, como pôde ser visto na solução apresentada no Capítulo anterior;

- O tempo demandado no aprendizado da linguagem base (GMQL) como também na criação de símbolos pictóricos para cada etapa da consulta também foi maior do que o esperado, e a solução encontrada foi a do fluxo corrente.
- A existência de poucos trabalhos na literatura que utilizam linguagens de consultas visuais sobre processos de mineração de dados geográficos, devido a sua complexidade.

6.3 CONTRIBUIÇÕES DESTE TRABALHO

Pode-se destacar como principal contribuição deste trabalho a idéia inovadora de enfrentar o desafio de especificar uma linguagem de consulta visual para mineração de dados geográficos. Além disso, destaca-se:

- Apresentação do estado da arte sobre bancos de dados geográficos, mineração de dados, linguagens de consultas visuais, linguagens de mineração de dados e os ambientes visuais de consultas.
- Especificação formal da gramática da linguagem, baseando-se na gramática da linguagem base (GMQL).
- Criação de símbolos pictóricos para cada atividade de associação da mineração de dados geográficos;
- Criação de símbolos pictóricos para todos os elementos constituintes da linguagem apresentada, tais como operadores e relacionamentos;
- Uso de *Árvores Metafóricas* como inovação da solução, e como facilitador do entendimento da construção da consulta.
- Descrição da arquitetura de um ambiente de consulta visual para realização de tarefas de mineração de dados, com a descrição funcional de seus módulos e camadas.
- Traduções de consultas visuais relativas a etapas distintas de um processo de descoberta de conhecimento para a linguagem base GMQL.
- Possibilidade de formular consultas sobre processos de descoberta de conhecimento em bases de dados geográficos de forma mais clara para usuários, uma vez que os símbolos criados são abstrações do mundo real.
- Utilização de exemplos práticos que validam a formulação das consultas feitas na linguagem definida.

6.4 TRABALHOS FUTUROS

Em relação aos trabalhos que podem ser feitos no futuro, de forma a estender a idéia proposta neste trabalho, sugere-se:

- Ampliação da gramática visual da linguagem de forma a abranger outros tipos de tarefas de mineração sobre dados geográficos.
- Utilização de *DataWarehouses* sobre qualquer tipo de tarefa a ser executada, inclusive nas tarefas de regras de associação as quais foram utilizadas neste estudo.
- Análise de erros semânticos, por parte de usuários distintos, uma vez que pode haver ambigüidade nas abstrações feitas.
- Utilização de conhecimentos semânticos prévios através de esquemas de bancos de dados ou ontologias, de forma a aperfeiçoar as consultas feitas no ambiente.
- Análise de como partes das consultas pode ser representada através de uma redução pictográfica.
- Implementação da segunda versão do ambiente ***GeoMiningVisual***, fazendo uso das especificações atuais.

REFERÊNCIAS BIBLIOGRÁFICAS

Adam, N. R. and Gangopadhyay, A. (1997) "**Database Issues in Geographic Information Systems**". Kluwer Academic Publishers.

Agrawal, R. and Srikant, R. (1994) "**Fast Algorithms for Mining Association Rules in Large Databases**". In: INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, VLDB, 20., San Francisco. Proceedings... California: Morgan Kaufmann. p.487-479.

Appel, A.P. (2003) "**Uma Linguagem Visual de Consulta a Banco de Dados utilizando o Paradigma de Fluxo de Dados**". Dissertação de Mestrado, Departamento de Ciências de Computação e Estatística. Universidade de São Paulo: São Carlos, SP.

Appice et al. (2005) "**Mining and Filtering Multi-level Spatial Association Rules with ARES**". In: INTERNATIONAL SYMPOSIUM ON METHODOLOGIES FOR INTELLIGENT SYSTEMS, ISMISm 15., 2005, New York. Proceedings...
[S.1.]: Springer, 2005.p.342-353 (Lecture Notes in Computer Science, 3488).

Barros, D. M. V. (2009) "**AGIS – Um Serviço para Processamento Geográfico e Analítico Multinível**". Dissertação de Mestrado. Recife, UFPE.

Batini, C.; Catarci, T.; Costabile, M. F. and Levialdi, S. (1991) "**Visual query systems.: A taxonomy**". In VDB, pages 153-168.

Bédard, Y.; Larrivée, S.; Proulx, M. J.; Létonurneau, F. and Caron, P.Y. (1997). "**Étude de l'état actuel et des besoins de R&D relativement aux architectures et technologies des data warehouses appliquées aux données spatiales**". Research Report for National Defence Canada Centre for Research in Geomatics, Laval University.

Bigolin, N. M. and Marsala, C. (1998) "**Fuzzy Spatial OQL for Fuzzy Knowledge Discovery in Databases**". 2nd European Symposium on Principles of Data Mining and Knowledge Discovery. J.M. Zytkow and M.Quafafou (Eds.) Springer Verlag Nantes, France September. pages 246-254. Lecture Notes in Computer Science.

Bimonti, S.; Ferrucci, F.; Laurini, R. and Polese, G. (2003) "**Prototype of a Visual Language for Spatial Data Mining Based on the 'Miner Trip' Metaphor: VisMiner**". Proceedings of the IEEE Symposium on Visual/Multimedia Languages, Auckland, New Zealand, October 28-31, 2003.

Bogorny, V. (2003) "**Algoritmos e Ferramentas de Descoberta de Conhecimento em Banco de Dados Geográficos**". Porto Alegre: PPGC da UFRGS.

Bogorny, V.; Alvares, L.O. (2005) "**Geographic Data Representation for Knowledge Discovery**". Technical Report. UFRGS. RP-349, Porto Alegre, Brazil, pp.32.

Bogorny, V. (2006) "**Enchancing spatial association rule mining in geographic databases**". Tese de Doutorado. Porto Alegre: PPGC da UFRGS.

Bogorny, V.; Alvares, L.O. and Kuijpers, B. (2009) "**ST-DMQL: A Semantic Trajectory Data Mining Query Language**". International Journal of Geographical Information Science, Volume 23, Issue 10 October 2009. UFRGS, Porto Alegre, Brazil, pages 1245 - 1276.

Brin, S.; Motwani, R.; Ulman, J. D. and Tsur, S. (1997) "**Dynamic Itemset counting and implication rules for market basket data**". In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 1997, Tucson, Arizona, USA. Proccedings … New York: AMC, p.255-264.

Câmara, G. (1995) "**Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos"**. Tese de Doutorado. INPE.

Catarci, T. and Santucci, G. (1994) "**Query by diagram: a graphical environment for querying databases**". ACM SIGMOD Record, v.23 n.2, p.515.

Dan Li, Jitender Deogun, Sherri Harmos. (2003) "**Interpolation techniques for geo-spatial association rule mining".** Proceedings of the 9th international conference on Rough sets, fuzzy sets, data mining and granular computing, Chongqing, China.

Dan Li , Jitender S. Deogun (2004) "**Interpolation models for spatiotemporal association mining**". Fundamenta Informaticae, v.59 n.2-3, p.153-172, February 2004

DBMiner Technology Inc. (2000) "**DBMiner Interprise 2.0**"
Disponível em: http://www.dbminer.com/ (último acesso em agosto de 2009)

Egenhofer, M. J. and Hering, J. (1991) "**Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases**". Technical Report, University of Maine, Orono.

Egenhofer, M. J. (1994) "**Spatial SQL: A Query and Presentation Language**". IEEE Transactions on Knowledge and Data Engineering, v.6 n.1, p.86-95.

Elmasri, R. and Navathe, S. B. (2005) "**Sistemas de Banco de Dados**". Editora Pearson, São Paulo, BRA.

Ester, M. et al. (2000) "**Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support**". Journal of Data Mining and Knowledge Discovery, [S.1.], v.4, n.2-3, p.193-216.

Ester, M.; Kriegel, H. and Sander, J. (2001) "**Algorithms and applications for spatial data mining**". Geographic Data Mining and Knowledge Discovery, Taylor and Francis.

Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. and Uthurusamy, R. (1996) "**Advances in Knowledge Discovery and Data Mining**" AAAI Press, Menlo Park, CA.

Fukuda, T. et al. (1996) "**Mining optimized associaton rules for numeric attributes**". In: ACM SIGMOD SYMPOSION ON PRINCIPLES OF DATABASE SYSTEMS, PODS, 15., Montreal. Proceedings... [S.1]: ACM Press, 1996. p. 182-191.

Hadzilacos, T. and Tryfona, N. (1992) "**A Model for Expressing Topological Integrity Constraints in Geographic Databases**". In: INTERNATIONAL CONFERENCE GIS - FROM SPACE TO TERRITORY: THEORIES AND METHODS OF SPATIO-TEMPORAL REASONING IN GEOGRAPHIC SPACE, GIS, Pisa. Proceedings... London: Springer, 1992. p.252-268

Han, J.; Cai, Y. and Cercone, N. (1993) "**Data-driven discovery of quantitative rules in relational databases**". IEEE Trans. Knowledge and Data Engineering, 5:29{40.

Han J. and Fu Y. (1995) "**Discovery of multiple-level association rules from large databases**". Proceedings... Int. Conference Very Large Data Bases, p. 420-431, Zurich, Switzerland.

Han, J.; Fu, Y.; Wang, W.; Kopersky, K. and Zaiane, O. A. (1996) "**DMQL: A DataMining Query Language for Relatinal Databases**" Simon Fraser University, B.C., Canada.

Han, J.; Koperski, K. and Stefanvic, N. (1997) "**GeoMiner: a system prototype for spatial data mining**" In: ACM-SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, SIGMOD, Tucson. Proceedings... [S.1.]: ACM Press, 1997. p.553-556

Han, J. and Kamber, M. (2001) "**Data Mining: Concepts and Techniques**" San Francisco: Morgan Kaufmann Publishers.

Hornsby, K. and Egenhofer, M. J. (1999) "**Shift in Detail Through Temporal Zoomig**". Tenth International Workshop on Database and Expert Systems Applications. Florence, Italy, IEEE Computer Society, pp. 487-491.

Huang, Z. and Svensson, P. (1993) "**Neighborhood Query and Analysis with GeoSAL, a Spatial Database Language**". Advances in Spatial Databases. In Proceedings of The 3rd International Symposium, SSD 93, Singapore.

Kade A. M. (2001) "**Uma linguagem visual de consulta a XML baseada em ontologias**". Dissertação de Mestrado - UFRGS.

Koperski, K. (1999) "**A progressive refinement approach to spatial data mining**". PhdThesis, the School of Computer Science, Simon Fraser University.

Koperski, K.; Han, J. and Adhikari, J. (1998) "**Mining knowledge in geographical data**". In COMM. ACM.

Lu, W.; Han, J. and Ooi, B. C. (1993) "**Discovery of general knowledge n large spatial databases**". In Proceedings… Far East Workshop on Geographic Information Systems, 275-289, Singapura.

Malerba, D.; Appice, A. and VACCA, N. (2002) "**SDMOQL: an OQL-based data mining query language for map interpretation tasks**". In: WORKSHOP ON DATABASE TECHNOLOGIES FOR DATA MINING, DTDM, Prague, 2002. Proceedings... [S.1.]: Springer.

Malerba, D. et al. (2003) "**Empowering a GIS with inductive learning capabilities: the case of INGENS**". Journal of Computers, Environment, and Urban Systems, [S.1.], v.27, p.265-281.

Massari, A. et al (1995). "**QBI: Query By Icons**". ACM SIGMOD International Conference on Management of Data, San Jose, California, ACM Press.

Meo, R.; Psaila, G. and Ceri, S. (1996) "**A new SQL-like operator for mining association rules**". In Proc. 1996 Int. Conf. Very Large Data Bases, pages 122{133, Bombay, India, Sept.

Meyer, B. (1992) "**Towards New Metaphors for Visual Query Languages for Spatial Information Systems**". Interfaces to Database Systems, R. Cooper (Ed.), Springer.

Morimoto et al. (1998) "**Algorithms for Mining Assosication Rules for Binary Segmentation of Huge Categorical Databases**", Proceedings of the 24th VLDB Conference, New York, USA.

Neves, M. C.; Freitas, C. C. and Camara, G. (2001) "**Mineração de Dados emGrandes Bancos de Dados Geográficos**". INPE. Relatório Técnico.

Padmanabhan, B. and Tuzhilin, A. (1998) "**A belief-driven method for discovering unexpected patterns**". In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, KDD, 4., 1998. Proceedings... New York: ACM Press, 1998. p94-100

PASQUIER, N. et al. (1999) "**Discovering frequent closed itemsets for association rules**". In: INTERNATIONAL CONFERENCE ON DATABASE THEORY, ICDT, 7., 1999, Jerusalem. Proceedings... [S.1]: Springer, 1999a. p398-416.

Raedt, S. (2000) "**A Logical Database Mining Query Language**". Proceedings of the 10th International Conference on Inductive Logic Programming, p.78-92, July 24-27, 2000

Santos, M. (2001) "**Padrão: um sistema de descoberta de conhecimento em bases de dados georreferenciadas**". Tese de doutorado - Universidade do Minho.

Santos Silva, M. P. (2002) "**SKDQL: Uma Linguagem Declarativa de Especificação de Consultas e Processos para Descoberta de Conhecimento em Bancos de Dados e sua Implementação**" Dissertação de Mestrado - UFPE.

Sherri Harms , Dan Li , Jitender Deogun , Tsegaye Tadesse (2002) "**Efficient rule discovery in a geo-spatial decision support system"**. Proceedings of the 2002 annual national conference on Digital government research, p.1-7, May 19-22, 2002, Los Angeles, California

Silberschatz, A. and Tuzhilin, A. (1996) "**What makes patterns interesting in knowledge discovery systems**" IEEE Transactions on Knowledge and Data Engineering, [S.1.], v.8, n.6, p.970-974.

Sizo, A. M.; Silva, L. V. L. and Bittencourt, P. O. (2002) "**Avaliação de Tráfego na Telefonia Móvel**". UNAMA, Belém – PA.

Soares, V.G. (2002) "**GeoVisual – Um ambiente de consultas visuais para bancos de dados geográficos**". Tese de Doutorado. Departamento de Informática – UFPE.

Soares, V. G. and Salgado, A. C. (1999) "**Consultas visuais em sistemas de informações geográficas baseadas em padrões de metadados espaciais**". In: I GeoInfo 1999, Campinas (SP).

Souza, C. F. (2008) "**SVQL: Uma Linguagem de Consulta Visual para SOLAP**". Dissertação (Mestrado em Ciencia da Computacao) - Universidade Federal de Pernambuco. Orientador: Valeria Cesário Times.

Wang, F.; Sha, J.; Chen, H. and Yang, S. (2000) "**GeoSQL: A spatial query language of object-oriented gis**". In Proceedings of the 2nd International Workshop on Computer Science and Informationn Technologies.

Worboys, M. and Duckam, M. (2004) "**GIS: A Computing Perspective**". Second Edition. Boca Raton, Florida, USA: CRC Press.

Wu, X.; Kumar, V.; Ross Quinlan, J.; Ghosh J.; Yang, Q.; Motoda, H.; McLachlan, G.; Ng, A.; Liu, B.; Yu, P.; Zhou, Z.; Steinbach, M.; Hand, M. and Steinberg, D. (2008) "**Top 10 algorithms in data mining**". Knowledge and Information Systems, 14(1):1{37.

ZLOOF, M. M. (1977). "**Query By Example: A Database Language**". IBM system Journal 4(16): 324-343.

APÊNDICE A

Exemplo de Utilização do Algoritmo APRIORI

Este apêndice tem o intuito de ilustrar o funcionamento do algoritmo ARIORI perante uma determinada base de dados específica.

Suponha que, por exemplo, dado um conjunto de 3 itens de supermercado (leite, pão e açúcar), as possíveis extrações de compras realizadas por um cliente qualquer poderiam ser resumidas, conforme ilustra a tabela 1:

Tabela 1: Exemplo de possibilidades de compras sobre um conjunto de 3 itens específicos.

Numero de itens levados por compra	**Possíveis compras**		
Nenhum item	não levar nada		
Apenas 1	Leite	pão	açúcar
2	leite e pão	leite e açúcar	pão e açúcar
3	leite, pão e açúcar (levar todos os itens)		

Observe que as possíveis extrações de compras feitas pelo cliente (tabela 1) podem ser obtidas através da equação exponencial 2^n, onde n representa cardinalidade do conjunto de itens. Logo, tem-se para o exemplo acima:

$2^n = 2^3 = 8$ possíveis compras pelo cliente, ou seja, *{ {não levar nada}, {leite}, {pão}, {açúcar}, {leite, pão}, {leite, açúcar}, {pão, açúcar}, {leite, pão, açúcar} }.*

Sendo assim, o tempo de execução do algoritmo cresce em uma ordem exponencial e isto acarreta um alto custo computacional.

Por outro lado, para reduzir o espaço de busca combinatória, os algoritmos para encontrar as regras de associação utilizam as seguintes propriedades:

a) O subconjunto de um conjunto de itens freqüentes (grande) precisa também ser grande. Esta propriedade é mais conhecida como **fechamento por baixo**, uma vez que é realizada através da inferência do superconjunto para o subconjunto.

b) O superconjunto de um conjunto de itens pequeno é também pequeno. Tal propriedade é mais conhecida como **antimonoticidade**.

É importante ressaltar que os algoritmos de mineração de regras de associação geram conjuntos "*candidatos*" a serem freqüentes no qual passam a ter a sua freqüência analisada sob certas circunstâncias. No APRIORI, por exemplo, esses *conjuntos candidatos* são freqüentes quando satisfazem as propriedades de *fechamento por baixo* e *antimonoticidade* vistas anteriormente.

Execução do algoritmo APRIORI

O processo de execução do algoritmo **APRIORI** envolve um conjunto de etapas. De maneira geral, os conjuntos candidatos são classificados em *k-itens*, onde k varia de 1 até a cardinalidade do conjunto de itens analisados. Sendo assim, em um primeiro momento, geram-se os conjuntos candidatos de cardinalidade 1 (CC-1) e verifica-se através das propriedades de ***fechamento por baixo*** e ***antimonoticidade*** a validade do conjunto ser freqüente perante o grau de ***suporte*** e ***confiança*** previamente estabelecido. Após serem validados, têm-se os conjuntos de itens freqüentes de tamanho 1 (CF-1). Em seguida, somam-se itens adicionais ao CF-1 resultante da etapa anterior, criando-se um conjunto de itens candidatos de tamanho 2 (CC-2) e, por conseguinte, todo o processo se repete. Tal execução prossegue até que nenhuma extensão do conjunto de itens em um determinado CF-k tenha todos os subconjuntos de itens k contidos em CF-k.

A figura 1 ilustra uma base de dados contendo 6 transações feitas por certo cliente nos quais estão presentes 5 itens: A, B, C, D, E.

id da transação	Itens presentes
001	A, B, C, D, E
002	B, C, E
003	A, C, D, E
004	A, B, C, E
005	A, B, C, D, E
006	B, C, D

Figura 1: Base de dados com 6 transações e 5 itens (A,B,C,D e E).

Em seguida, tem-se, a execução do algoritmo APIORI sobre a base de dados da figura 1. Suponha que se tenha definido para análise dos dados o valor do suporte igual a 50%.

Inicialmente, calcula-se o conjunto freqüentes de tamanho igual a um (1). Neste momento, são calculados os suportes de cada item individual, já que de início, cada item individual faz parte do conjunto de itens candidatos a serem freqüentes.

CC-1 = { {A}, {B}, {C}, {D}, {E} }

Tabela 2: Cálculo do suporte para os elementos do conjunto CC-1.

CÁLCULO DO SUPORTE PARA OS ELEMENTOS DO CONJUNTO CC-1	
suporte(A) = 4/6 = 0.67	**suporte(D)** = 4/6 = 0.67
suporte(B) = 5/6 = 0.83	**suporte(E)** = 5/6 = 0.83
suporte(C) = 6/6 = 1	

Como todos os suportes calculados acima são maiores que os 50% de suporte definido previamente para análise, todos os itens são classificados para o conjunto de itens freqüentes de tamanho igual a um (etapa k = 1 na *figura 2*).

Em seguida, somam-se itens adicionais ao CF-1 resultante da etapa anterior, criando-se um conjunto de itens candidatos de tamanho 2 (CC-2), respeitando, todavia, as propriedades de fechamento por baixo e antimonoticidade.

CC-2 = { {A,B}, {A,C}, {A,D}, {A,E}, {B,C}, {B,D}, {B,E}, {C,D}, {C,E}, {D,E} }

Tabela 3: Cálculo do suporte para os elementos do conjunto CC-2.

CÁLCULO DO SUPORTE PARA OS ELEMENTOS DO CONJUNTO CC-2	
suporte(A,B) = 3/6 = 0.5	**suporte(B,D)** = 3/6 = 0.5
suporte(A,C) = 4/6 = 0.67	**suporte(B,E)** = 4/6 = 0.67
suporte(A,D) = 3/6 = 0.5	**suporte(C,D)** = 4/6 = 0.67
suporte(A,E) = 4/6 = 0.67	**suporte(C,E)** = 6/6 = 1
suporte(B,C) = 5/6 = 0.83	**suporte(D,E)** = 3/6 = 0.5

Como todos os suportes calculados acima são maiores ou iguais aos 50% de suporte definido previamente para análise, todos os itens são classificados para o conjunto de itens freqüentes de tamanho igual a dois (etapa k = 2 na *figura 2*).

Em seguida, somam-se itens adicionais ao CF-2 resultante da etapa anterior, criando-se um conjunto de itens candidatos de tamanho três (CC-3), respeitando, novamente, as propriedades de fechamento por baixo e antimonoticidade.

CC-3 = {{A,B,C}, {A,B,D}, {A,B,E}, {A,C,D}, {A,C,E}, {A,D,E}, {B,C,D}, {B,C,E}, {B,D,E}, {C,D,E}}

Tabela 4: Cálculo do suporte para os elementos do conjunto CC-1.

CÁLCULO DO SUPORTE PARA OS ELEMENTOS DO CONJUNTO CC-3	
suporte(A,B,C) = 3/6 = 0.5	**suporte(A,D,E)** = 3/6 = 0.5
suporte(A,B,D) = 2/6 = 0.33	**suporte(B,C,D)** = 3/6 = 0.5
suporte(A,B,E) = 3/6 = 0.5	**suporte(B,C,E)** = 4/6 = 0.67
suporte(A,C,D) = 3/6 = 0.5	**suporte(B,D,E)** = 2/6 = 0.33
suporte(A,C,E) = 3/6 = 0.5	**suporte(C,D,E)** = 3/6 = 0.5

Nem todos os suportes calculados acima são maiores que os 50% de suporte definido previamente para análise. Sendo assim, dentre todos os conjunto de itens candidatos em CC-3, apenas o {A,B,D} e o {B,D,E} não são classificados para o conjunto de itens freqüentes de tamanho igual a três (etapa k = 3 na *figura 2*).

Em seguida, somam-se itens adicionais ao CF-3 resultante da etapa anterior, criando-se um conjunto de itens candidatos de tamanho quatro (CC-4), respeitando, mais uma vez, as propriedades de fechamento por baixo e antimonoticidade.

CC-4 = { {A,B,C,D}, {A,B,C,E}, {A,B,D,E}, {A,C,D,E}, {B,C,D,E} }

Tabela 5: Cálculo do suporte para os elementos do conjunto CC-4.

CÁLCULO DO SUPORTE PARA OS ELEMENTOS DO CONJUNTO CC-4	
suporte(A,B,C,D) = 2/6 = 0.33	**suporte(A,C,D,E)** = 3/6 = 0.5
suporte(A,B,C,E) = 3/6 = 0.5	**suporte(B,C,D,E)** = 2/6 = 0.33
suporte(A,B,D,E) = 2/6 = 0.33	

Nem todos os suportes calculados acima são maiores que os 50% de suporte definido previamente para análise. Sendo assim, dentre todos os conjunto de itens candidatos em CC-4, apenas o {A,B,C,E} e o {A,C,D,E} são classificados para o conjunto de itens freqüentes de tamanho igual a quatro (etapa k = 4 na *figura 2*).

Em seguida, somam-se itens adicionais ao CF-4 resultante da etapa anterior, criando-se um conjunto de itens candidatos de tamanho cinco (CC-5). Todavia, este conjunto de itens candidatos de tamanho igual a cinco (CC-5) não respeitam as propriedades de fechamento por baixo.

Observe que dentre os elementos do CC-5 = {A,B,C,D,E} nem todos os seus subconjuntos é também um conjunto de itens freqüente. Por exemplo, o subconjunto {A,B,D}. Logo, o algoritmo encerra-se com os resultados obtidos na *figura 2.14*.

Etapa	Conjuntos Freqüentes gerados
k = 1	{A}, {B}, {C}, {D}, {E}
k = 2	{A,B}, {A,C}, {A,D}, {A,E}, {B,C}, {B,D}, {B,E}, {C,D}, {C,E}, {D,E}
k = 3	{A,B,C}, {A,B,E}, {A,C,D}, {A,C,E}, {A,D,E}, {B,C,D}, {B,C,E}, {C,D,E}
k = 4	{A,B,C,E}, {A,C,D,E}

Figura 2: Resultado da execução do algoritmo APRIORI sobre a base de dados da figura 2.

APÊNDICE B

Definição da Gramática da Linguagem GeoMiningVisualQL a partir de adaptações da Linguagem GMQL.

Símbolos Terminais

%terminal ***MINE***
%terminal ***FROM***
%terminal ***WHERE***
%terminal ***GROUP BY***
%terminal ***HAVING***
%terminal ***SET***
%terminal ***SPATIAL ASSOCIATION***
%terminal ***DESCRIBING***
%terminal ***WITH RESPECT TO***
%terminal ***AS***
%terminal ***GEO***
%terminal ***GEOMETRY***
%terminal ***+***
%terminal ***-***
%terminal *******
%terminal ***/***
%terminal ***SUM***
%terminal ***AVG***
%terminal ***MIN***
%terminal ***MAX***
%terminal ***COUNT(*)***
%terminal ***G_CLOSE_TO***
%terminal ***DIRECTION***
%terminal ***TOPOLOGY***
%terminal ***NOT***
%terminal ***AND***
%terminal ***OR***
%terminal ***IN***
%terminal ***DISTANCE***
%terminal ***CONTAINS***
%terminal ***WITHIN***
%terminal ***INTERSECTS***
%terminal ***NEIGHBOR***
%terminal ***EQUAL***
%terminal ***UNDER***
%terminal ***OVER***
%terminal ***WEST_FROM***
%terminal ***EAST_FROM***

%terminal ***SOUTH_FROM***
%terminal ***NORTH_FROM***
%terminal ***ATTRIBUTE***
%terminal ***CONFIDENCE***
%terminal ***SUPPORT***
%terminal ***CLASSIFICATION***
%terminal ***EXCEPTION***
%terminal ***BOUNDARY***
%terminal ***BOUNDAR_EDGES***
%terminal ***BOUNDARY_NODES***
%terminal ***BOUNDARY_POLYGONS***
%terminal ***INTERIOR***
%terminal ***INTERIOR_EDGES***
%terminal ***INTERIOR_NODES***
%terminal ***INTERIOR_POLYGONS***
%terminal =
%terminal <
%terminal >
%terminal <=
%terminal >=
%terminal <>
%terminal ***LIKE***
%terminal ***literal***
%terminal ***relationName***
%terminal ***distanceSpecification***

Símbolos Não-Terminais:

%naoterminal ***GMQL_STATEMENT***
%naoterminal ***GMQL_QUERY***
%naoterminal ***RULE_HEADER***
%naoterminal ***RELATION_LIST***
%naoterminal ***CONDITIONS***
%naoterminal ***ATTRIBUTES***
%naoterminal ***THRESHOLD_SPECIFICATION***
%naoterminal ***SPATIAL_TERM***
%naoterminal ***NO_SPATIAL_TERM***
%naoterminal ***PRED_DIMENSION_LIST***
%naoterminal ***SPATIAL_ATTRIBUTE***
%naoterminal ***SPATIAL_OPERATION***
%naoterminal ***SPATAL_PREDICATE***
%naoterminal ***OPERATOR***
%naoterminal ***CONDITIONS***
%naoterminal ***COMPARE_CONDITIONS***
%naoterminal ***COMPARATOR***
%naoterminal ***GMQL_STATEMENT***
%naoterminal ***GMQL_STATEMENT***

Gramática GMQL (adaptada)

GMQL_STATEMENT → *gmql_Query*

(ver esta questao do attributes para diferenciar terminal de não terminal)

gmql_Query → MINE *ruleHeader* **FROM** *relationList* **[WHERE** *conditions***]** **[GROUPBY** *attributes* **{,***attributes***}] [HAVING** *conditions* **] [SET** *threshold_specification***]**

ruleHeader → *ASSOCIATION_HEADER*

ASSOCIATION_HEADER → SPATIAL ASSOCIATION [DESCRIBING ***literal*****]** **WITH RESPECT TO** *spatial_Term* **[AS** ***literal*****],** *pred_DimensionList*

spatial_Term → *spatial_Attribute* | *spatial_Operation* ***(****spatial_Attribute****)*** | *spatial_Operation****(****spatial_Term****)***

spatial_Attribute → *relationName****[.geo]*** | *relationName****[.geometry]***

spatial_Operation → BOUNDARY | ***BOUNDARY_NODES*** | ***BOUDARY_EDGES*** | ***BOUNDARY_POLYGONS*** | ***INTERIOR*** | ***INTERIOR_NODES*** | ***INTERIOR_EDGES*** | ***INTERIOR_POLYGONS***

pred_DimensionList → *nonSpatial_Term* ***[AS literal]***

{ *pred_DimensionList* ***}*** | *spatial_Predicate* ***{*** *pred_DimensionList* ***}***

nonSpatial_Term → *relationName* | *nonSpatial_Term operator nonSpatial_Term* | ***literal*** *operator* ***literal*** | ***literal*** *operator nonSpatial_Term* | *measure*

operator → **+** | **-** | ***** | **/**

measure →

SUM (*nonSpatial_Term****)*** | ***AVG (****nonSpatial_Term****)*** |

MIN (*nonSpatial_Term****) | MAX (****nonSpatial_Term****) |***

COUNT(*) | WGAVG(*nonSpatial_Term****,*** *nonSpatial_Term****)***

spatial_Predicate → G_CLOSE_TO (*spatial_Term****,*** *Spatial_Term****,*** *distanceSpecification****)***
| DIRECTION (*spatial_Term****,*** *Spatial_Term****) | TOPOLOGY (*** *spatial_Term****,***
*Spatial_Term****)|***

distance → number_distance

conditions → NOT (*conditions****) |*** *conditions* ***AND*** *conditions* **|** *conditions* ***OR*** *conditions* **|**
compare_Conditions

compare_Conditions →

nonSpatial_Term stand_Comparator nonSpatial_Term **|** *nonSpatial_Term*
stand_Comparator ***literal*** **|**
spatial_Term spatial_Comparator spatial_Term **|**
DIRECTION (*spatial_Term****,*** *spatial_Term* ***)*** *stand_Comparator nonSpatial_Term*
|
DIRECTION (*spatial_Term****,*** *spatial_Term* ***)*** *stand_Comparator* ***literal*** **|**
DISTANCE (*spatial_Term****,*** *spatial_Term* ***)*** *stand_Comparator nonSpatial_Term* **|**
DISTANCE (*spatial_Term****,*** *spatial_Term* ***)*** *stand_Comparator* ***literal***

stand_Comparator → = | < | > | <= | >= | < > | ***LIKE***

spatial_Comparator → CONTAINS | WITHIN | INTERSECTS | NEIGHBOT | EQUAL
| UNDER | OVER | WEST_FROM | EAST_FROM | SOUTH_FROM | NORTH_FROM

threshold_specification → **ATTRIBUTE** *attribute* **“*literal*” | CONFIDENCE “*literal*” |**
SUPPORT “*literal*” | CLASSIFICATION “*literal*” | EXCEPTION “*literal*” |