Representando Esquemas de Bancos de Dados Geográficos no Formato Atributo-Valor para a Inferência de Padrões de Análise

GISA HELENA MELO BASSALO[1]
CIRANO IOCHPE[2]
NARA BIGOLIN[2]
[1]UFPA- Universidade Federal do Pará, Rua Augusto Corrêa, 01, 66.000-00 Belém, PA, Brasil
gisa@amazon.com.br
[2]UFRGS—Universidade Federal do Rio Grande do Sul – Instituto de Informática, Caixa Postal 15.064, 91501-970
Porto Alegre, RS, Brasil
{ciochpe,bigolin}@inf.ufrgs.br

Abstract. Geographic Information Systems (GIS) have been so much used as planning and decision support tools in different knowledge areas. Its implementation is complex mainly due to the design of the Geographic Database (GDB). The construction of an analysis patterns catalogue would extremely help this task. An alternative to obtain analysis patterns candidates is the Data Mining (DM) algorithms application over GDB conceptual schemas. This solution requires conceptual schemas preparation for use them as input data in DM commercial tools, as such the application and model semantics preservation be guaranteed. Based on this context, this paper presents a proposal for conceptual schemas preparation process and an attribute-value format representation technic, with the objective of inferring GDB analysis patterns candidate to auxiliary GIS construction.

Resumo. Sistemas de Informação Geográfica (SIG) têm sido bastante utilizados como ferramentas de apoio ao planejamento e à decisão em diversas áreas do conhecimento. Sua implementação é complexa, sobretudo, em virtude da construção do Banco de Dados Geográficos (BDG). A montagem de um catálogo de padrões de análise auxiliaria sobremaneira a realização dessa tarefa. Uma alternativa para obter candidatos a padrão de análise é a aplicação de algoritmos de Mineração de Dados (MD) sobre esquemas conceituais de BDG. Essa solução requer a preparação dos esquemas conceituais para serem utilizados como dados de entrada por esses algoritmos, de tal modo que seja garantida a preservação da semântica da aplicação e do modelo empregado na simplificação da realidade. Com base no contexto descrito, este artigo apresenta uma alternativa para o processo de preparação dos esquemas conceituais de BDG e uma técnica de representação desses esquemas no formato atributo-valor com o propósito de inferir candidatos a padrão de análise de BDG para auxiliar a construção de sistemas de informação geográfica.

1 Introdução

A relevância de Sistemas de Informação Geográfica (SIG) tem levado um número cada vez maior de pesquisadores e de estudiosos a dedicar-se à descoberta de mecanismos que viabilizem, de forma rápida e consistente, sua implantação.

O Banco de Dados Geográficos (BDG) é um dos principais componentes do projeto de um SIG. Sua construção é complexa, grande parte devido à dificuldade de desenvolvimento da etapa de modelagem conceitual [1]. Essa etapa demanda muito tempo, pois requer que o projeto de um novo sistema seja iniciado do conhecimento do domínio e do levantamento dos requisitos da aplicação, mesmo que o problema a ser resolvido seja recorrente [2].

O crescente emprego de técnicas como, por exemplo, a reutilização de componentes, especificamente, a de definição e reutilização de padrões de análise tem contribuído para minimizar a complexidade da tarefa de modelagem, pois, além de agilizar a concepção do BDG e de possibilitar a disseminação de conhecimento e a troca de experiências entre projetistas, facilita a comunicação entre diferentes integrantes do projeto [1][3][4].

Um padrão de análise é uma combinação de elementos recorrentes de modelagem que apresenta a essência de uma solução para problemas análogos em contextos específicos e que descreve, em um nível arbitrário de abstração, um conjunto de objetos do mundo real, seus relacionamentos e as regras que definem seu comportamento e seu estado [5] [6] [2].

A maioria, senão a totalidade, dos padrões de análise, conhecidos da literatura, foi proposta por especialistas no domínio da aplicação a que se propõem, utilizando métodos empíricos [1][7]. Esse fato abre a possibilidade de tais padrões serem refutados por outros especialistas com experiências de projeto diferentes, ou até mesmo, conflitantes.

A semi-automatização do reconhecimento dos candidatos a padrão de análise através de métodos estatísticos ou mistos possibilitaria maior independência dos especialistas, permitindo que os padrões sejam aceitos sem restrições por toda a comunidade de projetistas de BDG.

A aplicação de algoritmos de MD da área de Descoberta de Conhecimento em Bases de Dados (DCBD) sobre esquemas de BDG é uma alternativa semi-automatizada para a inferência de candidatos a padrão de análise [8].

A DCBD ou *Knowledge Discovery in Databases (KDD)* é a área de estudo relacionada ao desenvolvimento de métodos e técnicas para descobrir algum conhecimento ou significado implícito em bases de dados e compreende, segundo Fayyad em [9], as etapas de seleção, pré-processamento, transformação (preparação dos dados), mineração de dados (processamento) e interpretação/avaliação (pós-processamento).

Essa proposta permite uma análise rápida de um grande volume de esquemas conceituais, visto que pode ter várias de suas etapas automatizadas e que possibilita trabalhar com a totalidade ou apenas uma amostra significativa do conjunto dos esquemas considerados.

Com a utilização de ferramentas de MD é possível validar o conhecimento extraído (candidato a padrão) com base em valores estatísticos[10] e, também, concluir a respeito dos resultados obtidos [11]. Isso porque esses resultados demonstram, no caso de esquemas de BDG, a probabilidade de ocorrência (grau de confiança) de um determinado componente ou associações destes nos esquemas considerados, tornando o processo mais preciso e menos dependente da participação do especialista.

O algoritmo de MD deve processar um número significativo de esquemas de BDG de domínios de aplicação afins. Esses esquemas, entretanto, são descritos com base em diferentes modelos de dados e se apresentam, geralmente, em formatos não compatíveis com as ferramentas de MD.

Uma possível estratégia para solucionar esse problema pode ser descrita da seguinte forma:

identificação de um *conjunto união* de conceitos que descrevam todos os aspectos da realidade representáveis pelos construtores (conceitos) e pelas regras dos principais modelos conceituais de SIG.

para cada conceito do *conjunto união* deve ser, então, definida uma representação em um formato compatível com ferramentas de MD, sendo necessário, para tanto, a elaboração e a aplicação de regras que garantam a conversão sem ambigüidades ou perdas semânticas.

Essa solução fundamenta-se no fato de que se um elemento de esquema se comporta como uma ocorrência de uso de um conceito do *conjunto união* e se para esse conceito existe pré-definida uma representação em um formato compatível com ferramentas comerciais de MD, então fica determinada a forma de representação deste elemento no formato apropriado para MD.

Os dados de entrada para ferramentas de MD, assumem a forma de conceitos, instâncias e atributos. O conhecimento a ser adquirido é chamado de expressão do conceito. As instâncias são os dados a serem minerados - são exemplos individuais e independentes do conceito a ser aprendido. Os atributos, por sua vez, apresentam valores que caracterizam as instâncias. Cada instância é composta de itens que são pares *atributo-valor* [12], ou seja, cada instância tem um único valor para cada um de seus atributos.

A representação dos esquemas conceituais no formato *atributo-valor*, sem permitir perda semântica, agrega boa parte do esforço para o caminho da descoberta dos padrões de análise, pois, conforme mencionado em [13], um dos maiores entraves à área de descoberta de conhecimento (*KDD*) é a etapa de preparação dos dados.

O objetivo deste artigo é, então, apresentar uma técnica de representação desses esquemas no formato *atributo-valor*, mantendo a integridade semântica dos modelos e da aplicação.

As outras seções deste artigo organizam-se da seguinte forma. A seção 2 apresenta o *conjunto união* de conceitos. A seção 3 trata da técnica de representação dos esquemas no formato atributo-valor. A seção 4 apresenta o algoritmo de representação e a seção 5 apresenta as conclusões e os futuros trabalhos complementares.

2 *Conjunto União* de Conceitos

Esquemas conceituais de bancos de dados geográficos são definidos com base em modelos conceituais de SIG.

Análises semânticas realizadas nos modelos MADS[14], GeoOOA/REGIS[15], SAIF [16], OMT-G [17][18] e o *framework* UML-GeoFrame [1][19] demonstram a existência concorrente de conceitos específicos de cada um e de conceitos genéricos que apresentam notória equivalência de significados.

A figura 2(i) apresenta alguns desses conceitos, de acordo com o modelo ao qual pertencem.

Modelos / Conceitos	GeoOOA	MADS	SAIF	OMT-G	Geoframe
Classe Espacial	ok	ok	-	ok	ok
Atributo	ok	ok	ok	ok	ok
Associação	ok	ok	ok	ok	ok
Agregação espacial	ok	-	-	ok	-
Gen/espec	ok	ok	ok	ok	ok
Pacote/tema	ok	-	-	-	ok

Figura 2(i): conceitos do *conjunto união*

O *conjunto união* de conceitos [20] é a somatória de todos os conceitos genéricos e específicos identificados nos modelos estudados. Dentre esses conceitos é possível citar: classes, atributos, associações, métodos, espacialidade, multiplicidade, metadados e pacote/tema.

O papel do *conjunto união* no processo de preparação dos esquemas de BDG é fundamentalmente a sumarização dos fenômenos da realidade para os quais existe representação nos modelos de SIG. Por intermédio dessa sumarização é possível prever que tipos de elementos podem ser esperados nas etapas do processo de representação e conseqüentemente os mecanismos necessários ao armazenamento e à decomposição dos esquemas em *subesquemas* significativos.

3 Técnica de Representação de Esquemas Conceituais de BDG no Formato Atributo-Valor

A representação dos esquemas no formato atributo-valor pode ser entendida como o desenvolvimento dos procedimentos de decomposição em *subesquemas* significativos e mapeamento para o formato atributo-valor.

3.1. Decomposição em *subesquemas* significativos

A decomposição objetiva identificar estruturas com possibilidade de configurar um candidato a padrão de análise. Essas estruturas, também chamadas de *subesquemas* conceituais *significativos* são formadas por e*lementos* de esquema que são, de fato, ocorrências de uso dos conceitos do *conjunto união*.

A decomposição, aqui apresentada, utiliza apenas alguns dos conceitos do *conjunto união* selecionados com base nos seguintes critérios: *conjunto interseção* de conceitos, ou seja, são considerados aqueles comuns a todos os modelos de SIG e; conceitos essenciais à preservação da semântica do modelo e da aplicação.

Esses conceitos são as construções genéricas do paradigma de OO para análise e projeto de BD: as classes, os atributos, as associações e a multiplicidade (cardinalidade), complementados pelo *pacote,* um mecanismo de organização semântica utilizado na linguagem de modelagem *UML* [21].

O nível de desagregação obtido na decomposição dos esquemas conceituais depende do tipo de conhecimento que se deseja extrair do conjunto de dados e da técnica de mineração utilizada. É possível minerar e obter candidatos a padrão de análise somente a partir das características explícitas dos esquemas, ou então, segmentar esses esquemas de modo a obter regras mais específicas e deduzir, também, aqueles implícitos nos esquemas considerados.

Dependendo da técnica empregada pelo algoritmo de MD, algumas composições mais desagregadas podem ser necessárias para viabilizar a compreensão dos resultados da mineração. Por outro lado, há circunstâncias em que relações explícitas nos esquemas tornam-se redundantes se configurarem *subesquemas*. Nesse caso, elas não devem ser registradas, porque durante a etapa de pós-processamento será possível inferi-las.

O procedimento de decomposição adota critérios que viabilizam a utilização dos resultados da aplicação da técnica de representação por algoritmos que implementem as tarefas de classificação, agrupamento e regras associativas e, também, facilitem a etapa de avaliação e interpretação dos padrões/regras gerados pela MD.

De um modo geral, as regras elaboradas para decompor os esquemas conceituais estão fundamentadas na identificação e na seleção dos elementos significativos do esquema. Em determinadas situações, entretanto, candidatos a padrão de análise são gerados por meio de operações matemáticas da teoria de conjuntos. O objetivo, nesse caso, é facilitar a etapa de pós-processamento e permitir a descoberta de algum conhecimento implícito nos esquemas.

O procedimento de decomposição é recursivo. Ocorre a partir do banco de dados que armazena os esquemas conceituais de BDG e obedece a regras formuladas conforme as especificidades do(s) conceito(s) que compõem os *subesquemas significativos*. A seguir são apresentadas regras genéricas correspondentes aos conceitos pacote, classe, atributo, associação e multiplicidade. Para cada um dos conceitos existem, ainda, regras mais específicas.

PACOTE

Todo pacote deve gerar um *subesquema* significativo. Os pacotes são considerados relacionados

a outros pacotes e individualmente. Os tipos de *subesquema* gerados com base neste conceito são:

a) PACOTE

Um conjunto *S* de *subesquemas* candidatos a padrão de análise, gerado com base neste tipo pode ser visto na figura 3.1(i).

S = {pacote P1, pacote P2, pacote P3}

Figura 3.1(i) – *subesquemas* do tipo PACOTE

b) PACOTE EM FUNÇÃO DE OUTRO PACOTE (aninhamento de pacotes)

Se um pacote P1 contém um pacote P2 que contém um pacote P3, então, o conjunto *S* de *subesquemas* válidos é composto conforme apresentado na figura 3.1(ii).

S = {Pacote P1 ⊃ Pacote P2 , Pacote P2 ⊃ P3 }

Figura 3.1(ii): *subesquemas* do Tipo PACOTE CONTÉM PACOTE.

CLASSE

Isoladamente e relacionada a outros conceitos do conjunto união, a classe é considerada um *subesquema* significativo à identificação de padrões de análise. São gerados a partir do conceito Classe os seguintes tipos de *subesquema*:

c) CLASSE.

O conjunto *S* mostrado na figura 3.1(iii) descreve exemplos de *subesquemas* desse tipo.

S = {Classe C1, Classe C2 }

Figura 3.1(iii): *subesquemas* do Tipo CLASSE.

d) CLASSE COM SEUS ATRIBUTOS.

Um exemplo de *subesquemas* deste tipo é descrito no conjunto *S* da figura 3.1(iv), na qual considera-se que a classe C1 tem os atributos a1 e a2 e a classe C2 tem o atributo b1.

S = { classe C1 com Atributo a1 e Atributo a2, classe C1 com Atributo a1, classe C1 com Atributo a2, classe C2 com Atributo b1 }

Figura 3.1(iv): *subesquemas* do Tipo CLASSE COM ATRIBUTOS.

e) CLASSE EM FUNÇÃO DO PACOTE que a contém.

Se a classe C1 está contida no pacote P2 e este está contido no pacote P1, então o *subesquema* resultante da aplicação desta regra é, somente, o observado na figura 3.1(v).

S = {pacote P2 ⊃ classe C1}

Figura 3.1(v): *subesquemas* do Tipo CLASSE EM FUNÇÃO DO PACOTE.

f) CLASSE COM SEUS ATRIBUTOS, EM FUNÇÃO DO PACOTE que os contém.

Os *subesquemas* são obtidos a partir da composição do pacote com a classe e a combinação de seus atributos de acordo com o procedimento descrito no item "c". A figura 3.1(vi) apresenta o conjunto *S* formado por exemplos de *subesquemas* gerados com base neste tipo. É considerado que o pacote P1 contém as classes C1 de atributos a1 e a2 e C2 de atributo b1.

S = {pacote P1 ⊃ classe C1 com Atributo a1 e Atributo A2, pacote P1 ⊃ classe C1 com Atributo a1, pacote P1 ⊃ classe C1 com Atributo a2, pacote P1 ⊃ classe C2 com Atributo b1}

Figura 3.1(vi): *subesquemas* do Tipo CLASSE COM SEUS ATRIBUTOS, EM FUNÇÃO DO PACOTE

ATRIBUTO

O atributo de uma classe ou de uma associação é considerado parte de um *subesquema*. Devem ser identificados e combinados as suas respectivas classes ou associações por entender-se que não há resultado semanticamente relevante, sob o ponto de vista da identificação de padrões de análise, para um atributo isoladamente.

Um *subesquema* válido contendo atributos deve ser, portanto, uma composição na qual a classe ou associação é mantida constante e os atributos são combinados segundo os fundamentos matemáticos da teoria de conjuntos [25], obedecendo à expressão:

combinação de "*n*" atributos "*m*" a "*m*", onde "*m*" deve variar de "*n*" até "*1*", formando, a cada valor assumido por "*m*", "$\frac{n!}{m!*(n-m)!}$" subesquemas.

Por exemplo, se uma determinada classe A foi concebida com os atributos a, b e c, o conjunto *S* dos *subesquemas* gerados é do mesmo tipo daquele resultante da aplicação da regra "d" e pode ser visto na figura 3.1(vii).

S = { classe A com os atributos a,b,c; classe A com atributos a,b; classe A com atributos a,c e classe A com atributos b,c; classe A com atributo a, classe A com atributo b; classe A com atributo c}

Figura 3.1(vii): *subesquemas* do Tipo CLASSES COM ATRIBUTOS.

ASSOCIAÇÃO

Cada associação entre duas classes é considerada um *subesquema* relevante à inferência de padrões de análise. Mesmo que existam, no esquema, associações entre mais de duas classes, múltiplas associações entre classes ou, ainda, associações n-ária, elas devem ser interpretadas como um conjunto de várias associações binárias. Isso é possível porque as técnicas de MD consideradas neste estudo permitem inferir candidatos a padrão de análise por intermédio da interpretação e avaliação, na etapa de pós-processamento [22].

Por outro lado, alguns *subesquemas*, supostamente redundantes, devem ser gerados pois permitem que a inferência dos candidatos seja feita apenas por meio da análise do *conseqüente* da regra gerada na etapa de pós-processamento. É o caso dos tipos "h" e "i" , descritos a seguir.

São considerados *subesquemas* válidos gerados a partir do conceito de associação entre duas classes, os seguintes tipos:

g) ASSOCIAÇÃO ENTRE CLASSES

O conjunto *S* mostrado na figura 3.1(vii) descreve exemplos de *subesquemas* desse tipo.

S = {classe A associada à classe B}

Figura 3.1(viii): *subesquemas* do Tipo ASSOCIAÇÃO ENTRE CLASSES.

h) ASSOCIAÇÃO ENTRE CLASSES EM FUNÇÃO DO(S) PACOTE(S) que contém(êm) as classes associadas;

Por exemplo, se a classe A pertencente ao pacote P1 está associada à classe B pertencente ao pacote P2, então os *subesquemas* gerados são os mesmos vistos na figura 3.1(ix).

S = {pacote P1 ⊃ classe A associada à classe B ⊂ pacote P2}

Figura 3.1(ix): *subesquemas* do Tipo ASSOCIAÇÃO ENTRE CLASSES EM FUNÇÃO DO(S) PACOTE(S).

i) ASSOCIAÇÃO ENTRE CLASSES CONSIDERANDO OS ATRIBUTOS DAS CLASSES

Um exemplo de *subesquemas* desse tipo é descrito no conjunto *S* da figura 3.1(x), a qual considera que a classe A com atributo a1 está associada à classe B com atributo b1.

S = {classe A com atributo a1 associada à classe B, classe A associada à classe B com atributo b1, classe A com atributo a1 associada à classe B com atributo b1}

Figura 3.1(x): *subesquemas* do Tipo ASSOCIAÇÃO ENTRE CLASSES CONSIDERANDO OS ATRIBUTOS DAS CLASSES.

Outros tipos de *subesquema* podem ser obtidos a partir do conceito *associação*. Eles não serão apresentados, neste artigo, ou por configurarem redundâncias semânticas geradas com o objetivo de facilitar a etapa de pós-processamento ou por incluírem outros conceitos do *conjunto união* aqui não tratados.

MULTIPLICIDADE OU CARDINALIDADE

Assim como o atributo, a multiplicidade não compõe isoladamente *subesquema* significativo para a inferência de padrões de análise. Sua ocorrência está subordinada à existência e conseqüente ligação a uma associação. De fato, a cardinalidade ou multiplicidade é considerada, neste trabalho, atributo de uma associação.

3.2. Mapeamento para o formato atributo-valor.

A representação adotada para a descoberta de candidatos a padrão de análise de BDG é aquela em que os dados de entrada estão organizados em uma única relação ou arquivo e para a qual os valores desses dados são apresentados em pares *atributo-valor*. Essa representação será denominada daqui por diante de *FMEC* - formato para mineração de esquemas conceituais.

O procedimento de mapeamento dos *subesquemas* resultantes da decomposição para o *FMEC* obedece a alguns pressupostos formulados com o intuito de garantir a representação dos *subesquemas* de acordo com exigências das ferramentas comerciais de MD as quais dizem respeito, principalmente, à organização e ao

tipo dos dados de entrada (categóricos, numéricos, etc.). Esses pressupostos são os seguintes:

a) cada esquema de BDG deve ser uma instância do conjunto dos dados de entrada;

b) cada item de uma instância deve ser um *subesquema* considerado na decomposição;

c) as instâncias possuem tamanho variado em função da ocorrência, ou não, de um determinado item (*subesquema*) [12];

d) as instâncias ou esquemas devem ser dispostos em uma coluna e identificados por um ID (identificador único) colocado em uma outra coluna - *Organização vertical,* na qual os valores dos atributos são dispostos em uma única coluna [23];

e) Cada item de uma instância será definido tomando-se por base a expressão genérica a seguir, descrita com base em notação própria desenvolvida para este fim.

$$\$P1 \,.\, \$P2 \,.\, \ldots\, \$Pn \,.\, A \,.\, a' + a'' + \ldots\, an \quad \& \quad \$P3 \,.\, \$P4 \,.\, \ldots\, \$Pn \,.\, B \,.\, b' + b'' + \ldots\, bn \quad : \quad R \,.\, n\text{-}m(1) + n\text{-}m(2)$$

Figura 3.2(i): Expressão genérica do formato *FMEC* para *subesquemas* conceituais, onde,

$P1, $P2,...,Pn são pacotes do esquema;

A e ***B*** são classes do esquema. É válido qualquer tipo de classe descrita no conjunto União;

a', a", ..., a" são os atributos da classe A . É válido qualquer tipo de atributo admitido pelo conjunto União;

b', b", ..., b" são os atributos da classe B. É válido qualquer tipo de atributo admitido pelo conjunto União;

R é o relacionamento entre classes. É válido todo tipo de relacionamento descrito no conjunto União;

n-m(1) e n-m(2) são as multiplicidades (cardinalidades) da associação.

"$" caractere de pacote. Marca o início da identificação do pacote;

"." caractere de sufixação. Indica que o sucessor mantém alguma relação de dependência com o antecessor;

"**&**" caractere de ligação. Indica que existe um relacionamento entre as classes apresentadas;

"**:**" caractere de relacionamento. Marca o início da identificação do tipo de relacionamento existente;

"**+**" caractere de adição. Indica que os elementos relacionados possuem o mesmo tipo.

3.3. Exemplo de Aplicação

O esquema da figura 3.3(i) representa uma simplificação da realidade urbana.

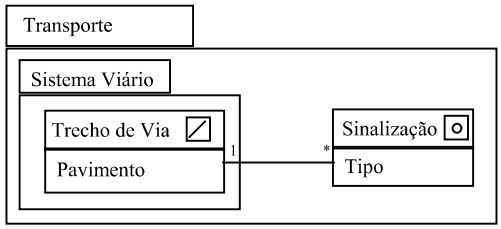

Figura 3.3(i): esquema conceitual de BDG – UML-Geoframe

Dado o esquema da figura 3.3(i), os *subesquemas* candidatos a padrão de análise gerados pelas regras relativas a cada tipo de conceito analisado, bem como sua organização no formato atributo-valor são apresentados na figura 3.3(ii):

ID	Subesquema (item)	Regra
1	$Transporte	a
1	$Sistema Viário	a
1	$Transporte.$Sistema Viário	b
1	Trecho de Via	c
1	Sinalização	c
1	Trecho de Via.pavimento	d
1	Sinalização.tipo	d
1	$Transporte.Sinalização	e
1	$Sistema Viário.Trecho de Via	e
1	$Transporte. Trecho de Via.pavimento	f
1	$Sistema Viário. Sinalização.tipo	f
1	Trecho de Via&Sinalização:AssocSimples. (1-1)+(0-n)	g
1	$Sistema Viário.Trecho de Via&Sinalização:AssocSimples. (1-1)+(0-n)	h
1	Trecho de Via&$Transporte.Sinalização:AssocSimples. (1-1)+(0-n)	h
1	$Sistema Viário.Trecho de Via&$Transporte.Sinalização:AssocSimples. (1-1)+(0-n)	h
1	Trecho de Via.pavimento&Sinalização.tipo:AssocSimples. (1-1)+(0-n)	i
1	Trecho de Via.pavimento&Sinalização:AssocSimples. (1-1)+(0-n)	i
1	Trecho de Via&Sinalização.tipo:AssocSimples. (1-1)+(0-n)	i

Figura 3.3(ii): Esquema conceitual de BDG no formato atributo-valor e em organização vertical.

4 Algoritmo de Representação de Esquemas de BDG no Formato Atributo-Valor

Nesta seção é apresentada de forma simplificada a seqüência de passos a serem executados durante a representação dos esquemas conceituais de BDG no formato atributo-valor.

O algoritmo de representação é recursivo. Até que todos os elementos de um mesmo esquema sejam analisados e mapeados para o formato atributo-valor, ele deve executar os seguintes passos:

(1) ler o banco de dados (relacional ou orientado a objetos) que armazena os esquemas conceituais de BDG;

(2) identificar/reconhecer marcador de esquemas (nome);

(3) encontrar os registros (a tabela) de pacotes;

(4) identificar os pacotes pertencentes a um mesmo esquema;

(5) mapear os pacotes de um mesmo esquema para o formato atributo-valor, da seguinte forma:

i. pacotes individualmente - armazenar *subesquemas* com a notação e a organização estabelecidas;

ii. pacotes combinados a outros pacotes na relação 1:1 - armazenar *subesquemas* com a notação e a organização estabelecidas;

(6) encontrar os registros (a tabela) de classes;

(7) mapear classes do mesmo esquema utilizado para os pacotes para o formato atributo-valor (notação e organização), da seguinte forma:

i. classes individualmente - armazenar *subesquemas* com a notação e a organização estabelecidas;

ii. classes associadas a pacotes - armazenar *subesquemas* com a notação e a organização estabelecidas;

iii. classes e seus atributos - armazenar *subesquemas* com a notação e a organização estabelecidas;

iv. classes e seus atributos em função do pacote que os contém - armazenar *subesquemas* com a notação e a organização estabelecidas;

(8) encontrar os registros (no BD) das associações;

(9) mapear as associações entre classes do mesmo esquema utilizado por pacotes e classes para o formato-atributo valor, da seguinte forma:

i. associações individualmente - armazenar *subesquemas* com a notação e a organização estabelecidas;

ii. associações e os pacotes relacionados às suas classes - armazenar *subesquemas* com a notação e a organização estabelecidas;

iii. associações e os atributos de suas classes - armazenar *subesquemas* com a notação e a organização estabelecidas;

iv. associações e os atributos de suas classes em função dos pacotes que contêm essas classes - armazenar *subesquemas* com a notação e a organização estabelecidas;

v. associações e seus atributos - armazenar *subesquemas* com a notação e a organização estabelecidas;

vi. associações e os pacotes relacionados às suas classes considerando os seus atributos - armazenar *subesquemas* com a notação e a organização estabelecidas;

vii. associações considerando os seus atributos e os atributos de suas classes - armazenar *subesquemas* com a notação e a organização estabelecidas;

viii. associações e os atributos de suas classes em função dos pacotes que contêm as classes, considerando os atributos das associações - armazenar *subesquemas* com a notação e a organização estabelecidas;

(10) voltar ao início e decompor próximo esquema.

5 Conclusões e Trabalhos Futuros

Se, é possível mapear qualquer um dos conceitos do *conjunto união* para o formato selecionado e se todo elemento de esquema é sempre uma ocorrência de uso desses conceitos, então é possível afirmar que qualquer esquema conceitual de bancos de dados geográficos pode ser representado no formato atributo-valor, compatível com ferramentas comerciais de mineração de dados.

A técnica desenvolvida pode ser utilizada para preparar não apenas esquemas de BDG, mas, também esquemas de outros tipos de BD.

Os *subesquemas* da decomposição foram gerados para utilização preferencialmente em algoritmos que implementem a técnica de indução de regras associativas, entretanto, eles podem ser considerados como dados de entrada, também, para algoritmos de classificação e agrupamento.

O estudo apresentado neste artigo ainda está em andamento. Trabalhos futuros podem, então: implementar o algoritmo de representação definido; desenvolver e implementar algoritmos interpretadores que mapeiem os elementos (conceitos) de cada modelo conceitual de SIG para os conceitos do conjunto união; descrever o conjunto união e o algoritmo de representação em XML com o intuito de compatibiliza-los a esquemas conceituais disponíveis na internet; evoluir o conjunto união de conceitos para um conjunto de ontologias que descrevam os diversos fenômenos da

realidade e permitam solucionar problemas, como, por exemplo, o de sinônimos, parônimos e homônimos.

Referências

[1] F. J. LISBOA, C. IOCHPE and K. A. V. BORGES, "Padrões de Análise para Reutilização de Esquemas de Dados de SIG em Aplicações de Gestão Urbana". CLEI. 2001.

[2] P. JOHANNESSON and P. WOHED, "The deontic Pattern – a framework for domain analysis in information systems design." Data & Knowledge Engineering. Vol.31. 1999.

[3] J. F. LISBOA and C. IOCHPE, "Mecanismos de Reutilização em Sistemas de Informação". In: Revista Acta Scientiae. Canoas, RS. 1999

[4] S. ROBERTSON and K. STRUNCH, "Reusing the Products of Analysis". Procs. of Int. Workshop on Software Reusability. Lucca, Italy. 1993. http://www.atlsysguild.com/GuildSite/SQR/reusingAnalysis.html.

[5] E. GAMMA et al. "Design Partterns: elements of reusable object-oriented software". Reading, MA: Addison Wesley. 1994.

[6] E. B. FERNANDEZ, "Building Systems Using Analysis Patterns". Procs. Of int. Software Architecture Workshop (ISAW3). 1998.

[7] J. F. LISBOA and C. IOCHPE, "Padrões de Análise para Banco de Dados Geográficos". PGCC da UFRGS, Porto Alegre,RS, 1999.

[8] C. MARSALA and N. M. BIGOLIN, "Spatial Data Mining with Fuzzy Decision Trees", Data Mining, p.235-248. WIT Press. 1998.

[9] U. FAYYAD, G. PIATETSKY-SHAPIRO and P. SMYTH, "From *data minig* to discovery knowledge in databases". AI Magazine, 3(17): 37-54. 1996.

[10] J. ELDER, H. P. KRIEGEL and X. XU, "Statistical perspective on knowledge discovery in databases". In proceedings of Advances in knowledge discovery and data mining. Pp83-115. 1996

[11] H. MANNILA. "Data mining: machine learning, statistics, and databases". In proc. of Eight Intern. Conference on Scientific and Statistical Database Management. Pp. 1-8. Stockholm. June 1996.

[12] I.H. WITTEN and E. FRANK, Data mining: practical learning tools and techniques with Java implementations. Morgan Kaufmann publishers. San Diego, CA, USA. 2000.

[13] K. KOPERSKI, J. ADHIKARI and J. HAN, "Spatial *data minig*: Progress and challenges". In SIGMOD'96 Workshop on Research Issues on *Data minig* and Knowledge Discovery (DMKD'96). 1996.

[14] C. PARENT, S. SPACCAPIETRA and E. ZIMANYI, MADS: Modeling of Application Data with Spatio-Temporal features. http://lbdwww.epfl.ch/e/research/mads. Consultado em 14/11/1999.

[15] G. KÖSTERS, B. PAGEL and H. SIX, GIS-application development with GeoOOA. International Journal of Geographical Information Science. London, v.11, n.4. 1997.

[16] SAIF 3.2. Ministry of Sustainable Resource Management. Province of British Columbia. Geographic Data BC. 1995-2001. Disponível em http://home.gdbc.gov.bc.ca/SAIF. . Consultado em setembro de 2001.

[17] K.V.A. BORGES, Modelagem de Dados Geográficos: uma Extensão do Modelo OMT para Aplicações Geográficas. Dissertação de Mestrado. Escola de Governo de MG/FJP. Belo Horizonte. 1997.

[18] C.A. DAVIS Jr. And A.H.F. LAENDER. Extensões ao modelo OMT-G para produção de esquemas dinâmicos e de apresentação. In Anais do II Workshop Brasileiro de GeoInformática (GeoInfo 2000), 29-36, 2000.

[19] L.V. ROCHA, C. IOCHPE and N. EDELWEISS. O *framework* Conceitual GeoFrame versão 2.0. Instituto de Informática da UFRGS. Porto Alegre. Junho, 2001.

[20] G.H.M. BASSALO. Integração de modelos conceituais para sistemas de informação geográfica voltada à preparação de esquemas de bancos de dados geográficos para utilização em ferramentas de descoberta de conhecimento. Porto Alegre: PPGC da UFRGS, 2001 (Trabalho Individual).

[21] G. BOOCH, Y. JACOBSON and J. RUMBAUGH, The Unified Modeling Language. Addison-Wesley, 1999.

[22] C. M.S. SILVA and C. IOCHPE. Applying the Process of Knowledge Discovery in Databases to Identify Analysis Patterns for Reuse in Geographic Database Design. SBIA 2002. Fortaleza, CE, Brazil.

[23] BRAND, E.; GERRITSEN, R. Association and Sequencing. DBMS Magazine. Miller Freeman Inc, 1998. Disponível em http://www.dbmsmag.com/9807m03.html. Acesso em 01nov.2001.