Uma Metodologia para Projeto de Sistemas de Informações Geográficas

Fatima Pires
CCUEC - UNICAMP
fpires@dcc.unicamp.br

Claudia Bauzer Medeiros*
DCC - UNICAMP
cmbm@dcc.unicamp.br

Abstract

Este artigo apresenta uma metodologia para o projeto e desenvolvimento de aplicações de Sistemas de Informações Geográficas (SIG), baseada no modelo de orientação a objetos. Essa metodologia é o resultado de um trabalho conjunto de pesquisadores das áreas de Bancos de Dados e usuários de SIG, no contexto de aplicações de controle ambiental. Ela foi desenvolvida de forma a permitir ao usuário de tais sistemas participar ativamente do processo de projeto de suas aplicações.

Resumo

This paper presents a methodology for the design and development of Geographical Information Systems (GIS) applications, based on the object-oriented paradigm. The methodology is the result of joint work of database researchers and GIS users, in the domain of environmental control applications. It allows users to actively participate in the design process of their applications.

1 Introdução

Um SIG é um sistema para processamento de aplicações geográficas, como por exemplo cartografia ou planejamento ambiental. Os dados utilizados pertencem a uma classe especial, conhecidos como *dados georeferenciados*. Este termo se refere a dados que descrevem entidades do mundo real associadas à sua localização sobre o globo terrestre, como por exemplo, rede elétrica ou tipo de solo de uma determinada região. As informações sobre localização física de dados georeferenciados são armazenadas sob a forma de *dados espaciais*, nome genérico dado ao conjunto de estruturas de dados e técnicas de armazenamento para a representação de entidades multidimensionais e sua distribuição no espaço.

*Pesquisa parcialmente financiada pelos projetos FAPESP 91/2117-1 e CNPq 500869/91-0. As autoras agradecem as valiosas sugestões feitas pelos avaliadores do SBES, que permitiram a reavaliação da metodologia proposta, resultando em melhoria substancial na sua especificação. As autoras agradecem igualmente ao professor Ardemiris Barros Silva, da FEC-UNICAMP, pelas inúmeras horas de discussão e pelos esclarecimentos necessários à realização deste trabalho.

Técnicas tradicionais de modelagem de dados não são adequadas para tratamento de dados geográficos. Como ressaltado por [13], a dificuldade reside em que a maior parte destes dados tem seu processamento ligado à *localização* para o qual são válidos, ao *tempo* em que foram coletados e à sua *confiabilidade* (do ponto de vista de correção da coleta de dados. Este artigo apresenta uma metodologia para projeto e desenvolvimento de aplicações SIG, no contexto de planejamento ambiental. A metodologia é resultado de um trabalho conjunto de pesquisadores em computação e usuários destas aplicações.

A metodologia apresentada parte do pressuposto que existe um sistema de bancos de dados subjacente, orientado a objetos, que suporte a modelagem conceitual e que permita sua tradução para uma implementação adequada. Esta metodologia está sendo testada através da implementação de aplicações de controle ambiental usando o SGBD orientado a objetos O2 [5]. Os exemplos no decorrer do artigo usam a sintaxe desse sistema.

O resto do artigo está organizado da seguinte forma. A seção 2 apresenta uma visão geral dos diversos tipos de modelagem de sistemas SIG. A seção 3 apresenta o modelo de dados e a metodologia de projeto. A seção 4 dá um exemplo de modelagem a partir desta metodologia, para uma aplicação de controle ambiental. A seção 5 apresenta conclusões e direções futuras de pesquisa.

2 Características e Modelos de Dados para SIG

Os dados de um sistema geográfico podem ser classificados em três categorias [4]: *convencionais, espaciais e pictóricos*. Atributos *convencionais* são alfanuméricos e descrevem as propriedades tradicionais de uma entidade – por exemplo, nome de um acidente geográfico. Atributos *espaciais* descrevem características georeferenciadas (por exemplo, geometria de um lago). Em geral, são associados a coordenadas geográficas e descritos em termos de pontos, linhas e polígonos. Atributos *pictóricos* armazenam imagens (por exemplo, fotografias). Eles complementam a informação fornecida pelos atributos espaciais para permitir a caracterização de dados georeferenciados.

As consultas em um SIG podem ser classificadas em tres famílias diferentes [1]:

- apresentação dos dados armazenados - por exemplo, geração de um mapa;
- determinação de conjuntos de dados segundo padrões pré-determinados - por exemplo, divisão de uma região segundo sua cobertura vegetal
- previsão de situações futuras baseada em dados atuais - por exemplo, indicação de regiões ideais para o plantio de cana

Uma característica comum à maior parte das aplicações SIG é a sua dependência dos chamados *planos* ou *camadas temáticas*, que são conjuntos de dados sobre alguma entidade geográfica específica, como por exemplo, camada de hidrografia ou camada de geologia de uma dada região. Um SIG precisa prover aos usuários facilidades para seleção e combinação de várias camadas (as chamadas operações de *overlay*) a fim de lhes oferecer as várias visões sobre uma região sendo analisada.

Operações em SIG manipulam as três categorias de dados, tanto de forma isolada como combinada. O processamento combinado de dados dos três tipos apresenta problemas não apenas de implementação, mas até mesmo de especificação adequada por parte do usuário.

Um exemplo é uma operação de *overlay* que envolva duas camadas de natureza distinta, (por exemplo, densidade populacional e vegetação) armazenadas de formas diferentes e em escala diferente. Dados de população podem estar armazenados em registros com campos alfanuméricos (por exemplo, com nome de região e número de habitantes). Dados de vegetação são via de regra armazenados de forma mista: dados espaciais (coordenadas da região ocupada), textuais (descrição do tipo de vegetação) e pictóricos. Uma consulta envolvendo os dois temas, por exemplo, é: "quais as regiões de floresta tropical em que há núcleos habitacionais de mais de 30.000 habitantes". A formulação desta consulta em um sistema automatizado é extremamente complexa. Em sistemas comerciais, o usuário precisa conhecer inclusive detalhes do armazenamento de dados. Embora este problema diminua com o uso de SGBDs (Sistemas Gerenciadores de Bancos de Dados), o suporte por eles provido ainda não é adequado. As linguagens de consulta de bancos de dados tradicionais só recuperam dados textuais e não permitem suporte a consultas que envolvam este tipo de variação de dados armazenados.

A tendência atual em SIG é o desenvolvimento de aplicações usando bancos de dados. No entanto, há um sem-número de problemas abertos no que tange a introdução de aspectos espaciais em um SGBD e permitir a utilização de interfaces apropriadas (linguagens de consulta e interfaces gráficas).

Os dados espaciais podem estar no formato *raster* (*varredura*) ou *vector* (*vetorial*). Os primeiros geralmente são derivados de imagens coletadas por digitalização de mapas, ou levantamento aerofotogramétrico. Essas imagens são divididas em reticulados, armazenados como *células* de matrizes, onde cada célula está associada a um conjunto de valores para os pontos dentro da célula, ou um único valor médio para os pontos. No formato *vetorial*, dados são armazenados sob forma de coordenadas, representando *pontos, linhas e polígonos* aos quais são aplicadas funções geométricas. Para o processamento conjunto dos dois tipos de dados, é usual transformar dados do primeiro tipo no segundo (ou vice-versa): a cada célula é associado um ponto cujas coordenadas correspondem à posição da célula no espaço.

Classificações existentes dos modelos de dados SIG costumam dividi-los segundo o ponto de vista do usuário em tratamento em camadas (correspondente aos dados de formato varredura) e tratamento vetorial (por exemplo, [20] e [7]).

Este artigo propõe uma classificação diferente, baseada no modelo de dados utilizado e no suporte oferecido em termos de gerenciamento de espaço em disco:

- *modelo baseado em arquivos* – modelos que não dependem de nenhum SGBD. O projeto e modelagem de aplicações consiste em particionar os dados de acordo com os procedimentos que os usarão e as consultas são formadas por seqüências de procedimentos que acessam estas partições. Existe um arquivo para vegetação, outro para solo, e assim por diante.
- *modelo relacional* – estes modelos estão baseados em sistemas de bancos de dados relacionais e os dados são organizados em tabelas. O gerenciamento de espaço é geralmente

feito por dois sistemas: o SGBD suporta dados alfanuméricos e um outro sistema processa dados espaciais e pictóricos. Um exemplo comercial é o ARC-INFO [6]. Um sistema experimental é relatado em [14], onde a linguagem PSQL, para consultas a bancos de dados espaciais, foi construída sobre o banco de dados relacional ADMS. Nestes sistemas, cada tipo de dado é tratado por uma rotina específica (por exemplo, dados textuais pelo SGBD relacional, dados de imagem e espaciais por um conjunto de módulos de software que estão acoplados externamente ao SGBD).

- *modelo relacional estendido* – estes se apóiam em sistemas relacionais extensíveis e permitem criar dados compostos a partir de combinação de dados mais simples. Além disto, gatilhos ou procedimentos podem ser associados a campos de relações . Exemplos desse tipo de abordagem são [8], que descreve a modelagem e implementação de SIG usando o Starburst, e [18], que adiciona gerenciamento de árvores R e módulos de display gráfico ao sistema POSTGRES.

- *modelo orientado a objetos* - baseado em diferentes sistemas orientados a objetos, que usam todos dados vetoriais. Ainda não há nenhum produto na área, apenas descrição de protótipos experimentais. [10] analisa a representação de relacionamentos espaciais entre objetos no gerenciador de objetos Zenith. [21] propõe que aplicações SIG sejam tratadas a partir do acoplamento de um modelo de objetos a um sistema especialista. A implementação baseia-se em CLIPS/COOL, que é um ambiente de programação especialista. [2] modela aplicações cartográficas usando dados vetoriais no sistema de bancos de dados O2, em dois níveis de abstração: *Mapa* e *Geometria*. O nível de *Mapa* é um conjunto de tuplas, contendo componentes textuais e vetoriais (que pertencem ao nível de *Geometria*).

Embora os modelos subjacentes sejam diferentes, todas estas categorias apresentam características comuns:

- separação entre elementos textuais, pictóricos e espaciais;

- integração destes elementos através de mecanismos de composição. Esta composição é feita por camadas de software nos dois primeiros casos, por aninhamento de relações no caso do modelo relacional estendido, e por operadores de composição no modelo de objetos.

3 A metodologia de projeto

Conforme ressaltado em [13], cada família de aplicação tem necessidade de conjuntos próprios de regras para processar dados sobre tempo e espaço. A metodologia proposta por este artigo é geral o suficiente para ser empregada na modelagem de vários tipos de aplicação SIG. Isto é conseguido dividindo-se a atividade de modelagem em várias etapas de complexidade crescente, e deixando ao usuário a especificação das funções próprias à aplicação.

3.1 Princípios básicos

A metodologia é baseada no paradigma de orientação a objetos, e envolve dois tipos de atividades, realizadas de forma iterativa com a participação do usuário:

- determinação de tipos e operações de manipulação de dados (seções 3.2 e 3.3);

- identificação de entidades e características geográficas a serem manipuladas (seção 3.4);

Existe uma tendência entre pesquisadores em SIG para considerar sistemas orientados a objetos como os mais adequados para desenvolvimento de suas aplicações. Faltam, no entanto, experiências em dados reais. As principais razões citadas para justificar esta preferência são o suporte à reusabilidade e à dinâmica de objetos. Nossa opção por este tipo de modelo é justificada pelas necessidades constatadas na prática por usuários SIG:

- A orientação a objetos permite desenvolvimento interativo e incremental [12]. Muitos dos dados georeferenciados apresentam evolução temporal no seu comportamento (por exemplo, uma região de mata pode ter sua área modificada ao longo dos anos devido à devastação florestal). À medida em que esta evolução ocorre, aparecem novas entidades e novos relacionamentos entre entidades existentes, havendo portanto necessidade de modelagem incremental. Os demais modelos de nossa classificação não obedecem a esse requisito, sendo adequados apenas para um subconjunto de aplicações SIG em que haja menor comportamento evolutivo – por exemplo, sistemas cartográficos.

- O paradigma de objetos está baseado em encapsulamento e uso de métodos. Isso aumenta a possibilidade de reutilização de funções [19] e permite modelar o comportamento dinâmico de fenômenos naturais (usando métodos) e descrição de suas propriedades físicas. Todas as aplicações SIG que envolvem atividades de previsão e planejamento têm necessidade destas características.

- A maioria dos SIG atuais está concentrada operações de consulta (para efeitos práticos, não existem atualizações). Isto acontece porque os ambientes automatizados para SIG não provêm suporte adequado para modelar todos os interrelacionamentos de dados georeferenciados. Sistemas orientados a objetos preenchem esta lacuna, pois suportam a criação incremental de relacionamentos complexos entre objetos através de composição e herança.

O problema em se utilizar um modelo orientado a objetos é a ausência de padrão para tal tipo de modelo. Optamos por adotar o modelo baseado em classes formalizado por [3]. Neste modelo, classes são unidades para agrupamento de objetos que têm a mesma estrutura (*tipo*) e comportamento (*métodos*). Classes estão ligadas entre si por laços de herança e composição. Assim, um objeto é uma ocorrência de uma classe. Ele é caracterizado pelo seu estado (conteúdo), tipo e comportamento (dados pela classe a que pertence). Objetos complexos são criados a partir da composição de outros objetos usando construtores – como *set*, *list*, *tuple*. Além disto, o modelo comporta *funções*, que são aplicadas a valores (campos de objetos, por exemplo), independendo de onde estejam armazenados, e retornam valores.

Uma das vantagens em se utilizar um modelo baseado em classes é a existência de trabalhos que formalizam as operações neste tipo de modelo [15]. Com isto, poderemos, no futuro,

estabelecer um formalismo para a especificação de operações sobre objetos geográficos. A única tentativa neste sentido foi realizada por [13], no âmbito de regras formais de produção, usando lógica de primeira ordem e lógica temporal. Estas regras, no entanto, são voltadas para sistemas cartográficos, e de difícil utilização em outras famílias de SIG.

3.2 Tipos de dados

Os tipos básicos de dados são os seguintes:

- **Tipos que descrevem características não espaciais.** Atributos convencionais, que descrevem fenomenos independente de sua localização.
- **Tipo Imagem.** Dados pictóricos em formato bitmap.
- **Tipo Raster.** Dados em formato varredura
- **Tipos que descrevem características geométricas** Atributos em formato vetorial, utilizados como base para todas as operações espaciais:
  Point, Line, Polygon – como nos demais modelos vetoriais
  SpatialLobject – características espaciais dos objetos geográficos. Objetos do tipo *SpatialLobject* são construídos iterativamente a partir de herança e composição com outros objetos do tipo *SpatialLobject, Point, Line, Polygon, Raster* e *Image*.

A maioria dos demais modelos orientados a objetos para SIG cria objetos geográficos através da *composição* de objetos espaciais e não-espaciais. Nosso modelo, entretanto, especifica estes objetos a partir da propriedade de *herança*. Assim, ao invés da modelagem usual

"um objeto georeferenciado **contém** objetos espaciais"

nós modelamos o fato:

"um objeto georeferenciado **é um** objeto espacial"

A seguir, a especificação dos componentes geométricos e varredura usando a sintaxe do sistema de bancos de dados O2. A classe *Spatial_polygon*, por exemplo, descreve objetos de geometria poligonal. Outras classes podem ser criadas para manusear geometrias específicas.

Características espaciais

class *Image* **type** *Bitmap*
class *Raster* **type** array_2dim (tuple (cell: *Point*, valor: *real*))
class *Point* **type** tuple (x: *real* ,y: *real*);
class *Line* **type** tuple (name: *string*, coord: list(*Point*));
class *Polygon* **type** tuple (name: *string*, segment: list(*Line*));
class *Spatial_object* **type** tuple (name: *string*, geo_image: *Image*, geo_field: *Raster*);
class *Spatial_polygon* **inherits** *Spatial_object* **type** tuple (geometry: list(*Polygon*))

Note que *Line, Polygon* são definidas como *listas*, o que permite a ordenação de coordenadas e segmentos no espaço. A manutenção dos dados em uma estrutura ordenada facilita a aplicação de funções geométricas, cuja execução depende desta ordenação.

3.3 Operações de manuseio de dados

Os tipos básicos de dados incluem formatos vetorial e de varredura. Dados do tipo varredura são armazenados como matrizes, a fim de que possam ser manuseados pelos métodos gráficos associados. No entanto, para operações que têm necessidade de combinar os dois tipos de dados, convertemos todos para o formato vetorial, usando algoritmos padrão de tratamento destes dados. A escolha do formato vetorial se deve à existência de uma vasta experiência de implementação de algoritmos de manipulação de dados vetoriais, no domínio da geometria computacional. Por outro lado, operações de overlay devem ser preferencialmente executadas sobre dados em formato varredura. Neste caso, utilizam-se os dados varredura armazenados, ou é feita a conversão de vetorial para varredura.

Existem basicamente quatro tipos de operações para manuseio de dados:

- manipulação de características espaciais;
- manipulação de características convencionais – funções tradicionais que operam sobre caracteres alfanuméricos e não apresentam nenhuma novidade do ponto de vista de modelagem;
- manipulação combinando os dois tipos de características (incluindo conversão entre formatos vetorial e varredura); e
- manipulação de imagens.

Nos demais modelos SIG, ([10, 17]) os objetos das classes *Point, Line* e *Polygon* são a base para as operações espaciais. Em nosso modelo, tais operações podem ser feitas sobre dados vetoriais – no caso, por exemplo, de funções de conectividade – ou varredura (no caso de overlay). A vantagem em manter classes separadas para essas entidades é óbvia: permite criar bibliotecas de rotinas de processamento espacial e de manipulação de temas, que podem ser usadas por qualquer sistema SIG, independente da natureza da aplicação .

As classes que modelam características geométricas devem suportar métodos que implementem as operações definidas por [1] como necessárias a um SIG:

- correção e transformação de coordenadas - transformações geométricas, ajuste de arestas e coordenadas;
- mensuração - comprimento, área, distância;
- vizinhança - funções topográficas, intersecção , contorno, interpolação ;
- conectividade - contiguidade, proximidade, funções sobre grafos em geral.

Exemplos de métodos a serem definidos para a class *Line* são os vários tipos de intersecção:

class *Line* ...
  method *Inter_line* (l1,l2: *Line*): *boolean*;
(retorna verdadeiro se duas retas se interceptam)
  method *Inter_point* (l1,l2: *Line*): *Point*;
(retorna o ponto de intersecção)
  method *Inter_list* (l1,l2: *list(Line)*): *boolean*;
(retorna verdadeiro se ao menos uma reta de uma lista intercepta uma reta de outra lista)

3.4 Especificação de entidades geográficas

As seções anteriores descreveram a modelagem SIG do ponto de vista de estruturas de dados e operações, independente do usuário e do tipo de aplicação. Esta seção define os passos a serem seguidos para projeto de uma dada família de aplicações do ponto de vista de fenômenos geográficos. O projeto de aplicações SIG envolve basicamente quatro tipos de entidades para as quais é necessário envolvimento do usuário:

- aspectos básicos – são os tipos que descrevem as características geográficas não espaciais (seção 3.2);
- aspectos temáticos primários
- aspectos temáticos derivados
- granularidade

Aspectos temáticos primários

Denominamos *aspectos temáticos* qualquer conjunto de dados descrevendo uma camada temática. Um SIG utiliza vários aspectos temáticos – por exemplo, solo, hidrografia, sistema viário e outros. Cada família de aplicações se concentra em um subconjunto destes aspectos. Denominamos *aspecto primário* todo aquele pode ser definido a partir de um único tipo de fonte de informação. Aspectos primários constituem, do ponto de vista do usuário, os dados *coletados diretamente no mundo externo.*

Aspectos temáticos derivados

Aspectos temáticos primários não são suficientes para modelar uma região geográfica. Na maioria das vezes, é necessário combinar dados de diferentes aspectos a fim de criar outros temas. Denominamos o resultado desta combinação de de *aspecto temático derivado*, para indicar que retrata características que não podem ser inferidas a partir de um único aspecto temático (ao contrário do que acontece com aspectos primários). Aspectos primários correspondem a dados coletados externamente, enquanto aspectos derivados são gerados a partir da aplicação de funções que combinam dados de diversos aspectos primários e derivados. *Overlays* são um exemplo de operação de geração de aspectos derivados.

Um exemplo é o tema *solo para plantio* (derivado), que precisa combinar dados de *solo, curvas de nível, hidrografia precipitação pluviométrica, insolação*, oriundos de diferentes aspectos primários. A geração de aspectos derivados pode ser realizada através da operação usual de overlay (que sobrepõe dados varredura aplicando funções aritméticas) ou combinando dados de varredura e processamento geométrico sobre dados vetoriais.

Granularidade

Chamamos de *granularidade* a escala e o nível de detalhe dos dados. Diferentes tipos de usuários querem analisar o mesmo aspecto em um nível de detalhe maior ou menor. Normalmente, este tipo de consideração só se aplica a aspectos primários.

Por exemplo, o planejamento de tráfego de uma região urbana requer seu exame em um nível bem baixo – medidos em metros. Estudos de zoneamento, no entanto, trabalham com uma visão mais abstrata e estão somente interessados em dividir a região em áreas segundo as características demográficas e econômicas. Para isso, na maioria dos casos a cidade é dividida em zonas onde detalhes internos não interessam.

Aplicações de controle ambiental devem atender a todos esses níveis de usuários e portanto devem suportar diversos níveis de detalhamento. Geralmente, isso é feito *bottom-up* (ou seja, começando de um nível mais detalhado para abstrações sucessivas).

3.5 Passos da Metodologia

Qualquer atividade de projeto baseada no paradigma de objetos parte da determinação de classes e métodos [12]. O primeiro passo é **definir as classes espaciais** – *Point, Line, Polygon* e *Spatial_object* e derivadas. Estas hierarquias de classes formam uma biblioteca básica, fornecida como ponto de partida para a interação posterior com o usuário, que é realizada nos passos subsequentes. Ressalte-se que determinadas aplicações podem dispensar dados em formato varredura, ou dados em formato vetorial. Por exemplo, aplicações de roteamento utilizam quase que exclusivamente dados em formato vetorial. A metodologia parte do pressuposto de que ambos os tipos de formato são usados.

A seguir, o usuário deve **especificar operações adicionais** sobre essas classes necessárias a seu domínio de aplicações. Exemplos são a incorporação de noções vagas e qualitativas – proximidade ("perto", "longe"), tamanho ("grande", "pequeno"). Outro exemplo é o conjunto de funções necessárias à correção dos dados captados e unificação de coordenadas.

Para evitar adicionar métodos às classes geométricas básicas, a criação destas operações é realizada através da especificação de subclasses, que herdam o comportamento padrão e acrescentam novos serviços.

A partir deste instante, pode-se iniciar a **determinação das funções e aspectos** que devem ser providos pela aplicação. Inicialmente, devem ser definidos os *Aspectos geográficos básicos*, que são modelados como composição de valores atômicos e imagens.

Finalmente, são definidas as classes contendo **objetos geográficos** propriamente ditos (aspectos primários e derivados, com várias granularidades). Estes combinam aspectos básicos e outros objetos geográficos. Todos objetos geográficos fazem parte da hierarquia de classes cuja raiz é *Spatial_object*.

class *Obj_geográfico* **inherits** *Spatial_object* **type**
tuple (componentes atômicos, aspectos básicos, *Obj_geográficos, Image, Raster*)

Determinadas características podem ser modeladas por métodos ou por atributos. A opção depende dos requisitos da aplicação. Por exemplo, o usuário pode desejar classificar o objeto geográfico *Solo* como sendo do tipo argiloso, arenoso, etc., em função de sua composição orgânica. Este dado pode ser armazenado como um campo (Tipo: *string*) da classe *Solo*, ou ser obtido através da aplicação de um método (*Determinar_tipo*), que realiza esta classificação dinamicamente. As vantagens e desvantagens de cada tipo de modelagem são as mesmas encontradas no projeto de qualquer tipo de aplicação, qualquer que seja o modelo, e a escolha do caminho a seguir está sujeita ao mesmo tipo de heurística – vide, por exemplo, [9].

Objetos geográficos são agregados por composição em *Regiões*, derivadas, que passam a ser a unidade de trabalho básico do usuário final.

Os passos a serem seguidos na metodologia podem ser resumidos no seguinte:

1. Definir as características geométricas e suas operações. Este passo pode ser feito independentemente da aplicação. Os demais dependem de interação com o usuário.

2. Definir operações complementares nas classes definidas em (1).

3. Analisar o domínio da aplicação para determinar quais as funções e serviços básicos devem ser providos. Determinar aspectos a serem modelados.

4. Definir classes relativas a **Aspectos geográficos básicos**

5. Definir classes de objetos geográficos relativas a **Aspectos temáticos primários**, para a granularidade mais fina desejada, através de composição dos objetos definidos no passo (4).

6. Repetir enquanto houver fenômenos geográficos a definir:

   - Definir classes de objetos geográficos relativas a **Aspectos primários** de granularidade mais grossa, a partir de classes de granularidade mais fina
   - Definir classes de objetos geográficos relativas a **Aspectos derivados** a partir de operações sobre classes previamente definidas

7. Definir classes *Região*, que conterão dados sobre as regiões geográficas de interesse, usando composição e herança de classes definidas anteriormente.

Nos passos 4 a 7, a definição de cada classe é acompanhada pela especificação das operações respectivas.

4 Exemplo – projeto de sistema de controle ambiental

A especificação final das classes para uma aplicação de controle ambiental é apresentada em [11], onde é dada ênfase aos aspectos de bancos de dados. Parte desta especificação está indicada na figura a seguir. Nesta seção, detalhamos a especificação de alguns tipos de classes e operações, para ilustrar a metodologia e a participação do usuário.

Escolhemos como exemplo a criação das classes que descrevem o *Solo* de uma área, para determinação de erosão. A descrição dos fatores para cálculo do problema está em [16]. Dado um tipo de solo, cada região é analisada segundo os fatores K (potencial de erosão) e perda de solo (LS – loss of soil). O resultado é o zoneamento da região segundo suscetibilidade de erosão. Fórmulas básicas são então utilizadas para gerar o potencial de erosão de cada zona, combinando-se o fator KLS com um fator R (rain) de erosividade pluviométrica. Para um mesmo tipo de solo (valor único de KLS) podem ser geradas várias zonas de erosão distintas, dependendo da chuva. Neste caso, não há necessidade de criar operações específicas ou subclasses geométricas novas.

O **aspecto básico** solo é definido como

class *Solo* **type** tuple
(tipo: *string*, textura: *string*, permeabilidade: *string*)
method *Kfactor* **in class** *Solo*
/* se tipo = "alfisol" e textura = "clay" e permeabilidade = "slow" então Kfactor = 0.1 */
(outros valores de fator são indicados de forma semelhante)

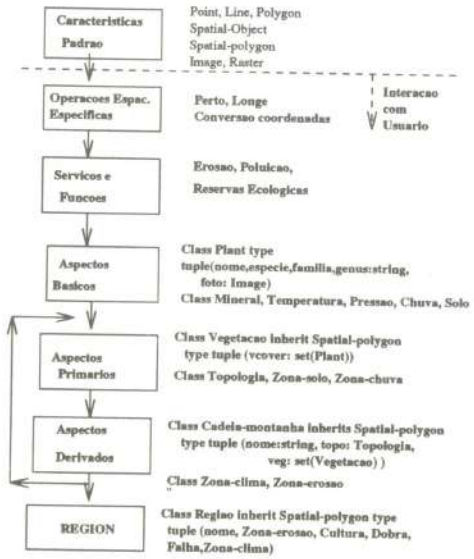

Figure 1: Exemplo - controle ambiental

Nosso trabalho trata do desenvolvimento de aplicações para controle ambiental em áreas não assentadas. Assim, o usuário necessita de dados dos seguintes **aspectos primários**: estrutura geológica, relevo, hidrografia, vegetação, zona-solo e fauna. O aspecto vegetação é construído a partir de informações sobre a flora de uma região e independe de dados de outras fontes. O aspecto *Zona_solo* cria polígonos segundo o tipo de solo e é especificado como:

class *Zona_solo* **inherit** *Spatial_polygon* **type**
tuple (s: *Solo*)

As regiões de erosão precisam combinar fatores de chuva e solo. O **aspecto derivado** *Zona_erosão* é definido como:

class *Zona_erosao* **inherit** *Spatial_polygon* **type**
tuple (zs: *Zona_solo*, zr: *Zona_chuva*)
method *RKLS* **in class** *Zona_erosão*
/* Para cada solo em *Zona_solo*, aplicar método *KLS*. Para cada polígono em *Zona_chuva*, aplicar método *R*. Fazer sobreposição do resultado, determinando novos polígonos. */

Passamos finalmente à criação de **Regiões**. Estas são tuplas contendo todas as características desejadas pelo usuário. Este define as principais zonas de interesse, para as quais existem limites predefinidos. A definição de novas regiões pode ser realizada por consulta, ou por aplicações que executam métodos combinando diferentes aspectos de regiões já existentes.

5 Conclusões

Este artigo apresentou uma metodologia para projeto de aplicações geográficas usando o paradigma de orientação a objetos. Esta metodologia está sendo testada no projeto de um conjunto de aplicações para controle ambiental que utilizam dados da Região da Cantareira, no Estado de São Paulo - aproximadamente 2000 km2 de área. A principal vantagem deste enfoque é a de permitir a participação do usuário na especificação de operações e classes para processamento de aplicações geográficas.

Uma atividade em andamento é a implementação de funções de interface apropriadas, para permitir que o usuário interaja diretamente com os objetos geográficos. Como parte das atividades a serem desenvolvidas ressaltamos a transformação de algoritmos de geometria computacional (por exemplo, funções de conectividade) para trabalharem com dados espaciais armazenados em disco, a fim de atender a consultas mais complexas sobre dados georeferenciados.

References

[1] S. Aronoff. *Geographic Information Systems.* WDL Publications, Canada, 1989.

[2] F. Baucilhon, C. Delobel, and P. Kanellakis, editors. *Building and Object-oriented System – the Story of O2*, chapter Geographic Applications – an Experience with O2. Morgan Kaufmann, California, 1992.

[3] C. Beeri. Formal Models for Object-oriented Databases. In *Proc. 1st International Conference on Deductive and Object-oriented Databases*, pages 370–395, 1989.

[4] M. Casanova, A. Hemerly, and A. Furtado. Cooperative Environments for Geographic Databases: a Prescriptive Analysis. In *Anais, 8 Simpósio Brasileiro de Bancos de Dados*, pages 266–279, 1993.

[5] O. Deux and al. The Story of O_2. *IEEE Transactions on Knowledge Bases and Data Engineering*, 2(1), 1990.

[6] Environmental Systems Research Institute, Inc. *ARC-INFO: GIS Today and Tomorrow*, 1992.

[7] A. Frank and M. Goodchild. Two Perspectives on Geographical Data Modelling. Technical Report 90-11, National Center for Geographic Information and Analysis, 1990.

[8] L. Haas and W. Cody. Exploiting Extensible DBMS in Integrated Geographic Information Systems . In *Proc. 2nd Symposium Spatial Database Systems*, pages 423–449. Springer Verlag Lecture Notes in Computer Science 525, 1991.

[9] R. Hull and R. King. Semantic Database Modeling: Survey, Applications and Research Issues. *ACM Computing Surveys*, 19(3):201–260, 1987.

[10] Z. Kemp and R. Thearle. Modelling Relationships in Spatial Databases . In *Proc 5th International Symposium on Spatial Data Handling*, pages 313–322, 1992. Volume 1.

[11] F. Pires, C. B. Medeiros, and A. Barros. Modelling Geographic Information Systems using an Object Oriented Framework. In *Proc XIII International Conference of the Chilean Computer Science Society*, 1993.

[12] M. Pittmann. Lessons Learned in Managing Object-oriented Development. *IEEE Software*, (1):43–53, January 1993.

[13] G-C. Roman. Formal Specification of Geographic Data Processing Requirements. *IEEE Transactions on Knowledge and Data Engineering*, 2(12):370–380, 1990.

[14] N. Roussoupolos, C. Faloutsos, and T. Sellis. An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering*, SE-14(5):639–650, 1988.

[15] E. Rundensteiner and L. Bic. Set Operations in Object-based Data Models. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):382–398, 1992.

[16] C. R. Valenzuela. The Digital Geographic Information System as a Management Tool. *Impact of Science on Society*, (156):314–324, 1990.

[17] A. Voisard. *Bases de Donnees Geographiques: du Modele de Donnees a l'INterface Utilisateur*. PhD thesis, Université de Paris-Sud Orsay, 1992.

[18] P. von Oosterom and T. Vijlbrief. Building a GIS on Top of the Open DBMS POSTGRES. In *Proc European GIS Conference*, 1991.

[19] N. Wilde, P. Matthews, and R. Huitt. Maintaining Object-oriented Software. *IEEE Software*, (1):75–80, January 1993.

[20] M. Worboys and S. Deen. Semantic Heterogeneity in Distributed Geographic Databases. *ACM Sigmod Record*, 20(4):30–34, 1991.

[21] F. Zhan and D. Mark. Object-Oriented Spatial Knowledge Representation and Processing: Formalization of Core Classes and their Relationships. In *Proc 5th International Symposium on Spatial Data Handling*, pages 662–671, 1992. Volume 2.