

**Anais do  
II Workshop Brasileiro de GeoInformática**

**GeoInfo 2000**



12 e 13 de junho de 2000  
Centro Anhembi, São Paulo, SP, Brasil

Copyright © 2000 by the National Institute for Space Research – INPE.  
Printed in Brazil.

Permission to make digital or hard copies for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

Additional copies may be ordered prepaid from:

National Institute for Space Research – INPE  
Image Processing Division – DPI  
Caixa Postal 515  
12201-970 São José dos Campos, SP, Brazil  
Phone: +55 12 345-6444  
Fax: +55 12 345-6468  
E-mail: [geoinfo@dpi.inpe.br](mailto:geoinfo@dpi.inpe.br)

Electronic copies of the papers in these proceedings are available at  
<http://www.tecgraf.puc-rio.br/geoinfo2000/>.

# Prefácio

Este volume contém os 18 trabalhos apresentados no GeoInfo 2000, o II Workshop Brasileiro de GeoInformática, nos dias 12 e 13 de junho de 2000, no Centro Anhembi, em São Paulo, SP.

O GeoInfo 2000 continua a série iniciada com o GEO-INFO 1999 e visa reunir os interessados em pesquisa em informações espaciais, com ênfase nos aspectos computacionais. O principal objetivo desses workshops anuais é servir como fórum para a apresentação de novas idéias.

O GeoInfo 2000 foi realizado em paralelo com o GEOBrasil 2000 com o objetivo de aproximar a comunidade acadêmica da comunidade de usuários e da indústria.

Foram submetidos 25 trabalhos para o GeoInfo 2000—mais do que o dobro do que em 1999. Esses trabalhos foram avaliados pela Comissão de Programa, com auxílio de revisores adicionais. Cada trabalho foi avaliado por pelo menos dois especialistas no assunto do trabalho. Com base nessas avaliações, foram selecionados os 18 trabalhos que constam destes anais. Todo o processo de submissão e avaliação foi feito pela internet, o que permitiu que os curtos prazos pudessem ser cumpridos. Esse processo está completamente documentado no *site* do GeoInfo 2000, que também contém versões PDF dos artigos aceitos: <http://www.tecgraf.puc-rio.br/geoinfo2000/>.

Além das sessões de contribuições técnicas, o GeoInfo 2000 teve também palestras dos convidados internacionais Max Egenhofer (University of Maine) e Jayant Sharma (Oracle). Os resumos dessas palestras também constam destes anais.

Agradecemos a todos que contribuíram para o GeoInfo 2000: aos autores, pelo esforço no preparo dos seus trabalhos; aos avaliadores, por contribuírem com a sua *expertise*; e aos membros da Comissão de Programa e da Comissão Organizadora, pelo seu tempo e dedicação. Agradecemos também à Divisão de Processamento de Imagens do INPE e ao TeCGraf (Grupo de Tecnologia em Computação Gráfica da PUC-Rio), pelo apoio logístico e secretarial, fundamentais para a realização do GeoInfo 2000. A todos, muito obrigado!

O GeoInfo 2000 é um evento apoiado pela Sociedade Brasileira de Computação e foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), no âmbito da cooperação existente entre o CNPq e a National Science Foundation (NSF) sobre “Collaborative Research on Integrating Geographic Information”.

Clodoveu Davis, PRODABEL  
Luiz Henrique de Figueiredo, IMPA  
*Coordenadores da Comissão de Programa*

Gilberto Câmara, INPE  
*Coordenador da Comissão Executiva*

# Comissão de Programa

Clodoveu Davis (PRODABEL) (coordenador)  
Luiz Henrique de Figueiredo (LNCC) (coordenador)  
Ana Carolina Salgado (UFPE)  
Carla Madureira (UFRJ)  
Claudia Bauzer Medeiros (UNICAMP)  
Flavio Torres (PETROBRAS)  
Frederico Fonseca (University of Maine)  
Geovane Magalhães (CPqD)  
João Argemiro Paiva (INPE)  
Jugurta Lisboa Filho (UFV)  
Luiz Paulo Vieira Braga (UFRJ)  
Marcelo Tilio Monteiro de Carvalho (PUC-Rio)  
Marco Antonio Grivet Mattoso Maia (PUC-Rio)  
Margareth Simões (EMBRAPA)

# Comissão Executiva

Gilberto Câmara (INPE) (coordenador)  
Antônio Miguel Monteiro (INPE)  
Claudia Bauzer Medeiros (UNICAMP)  
Marcelo Gattass (PUC-Rio)

# Avaliadores Adicionais

Aluir Dal Poz  
Anselmo Cardoso de Paiva  
Claudio Esperança  
Luiz Fernando Martha  
Marcelo Gattass  
Marco Antonio Casanova  
Max Egenhofer  
Paulo Cezar Carvalho  
Roberto de Beauclair Seixas  
Simone Diniz Junqueira Barbosa  
Waldemar Celes

# Conteúdo

## Palestras convidadas

- Computational Models for Spatio-Temporal Movement ..... 1  
*Max Egenhofer*
- Oracle Spatial: Experiences with Extensible Databases ..... 2  
*Jayant Sharma*

## Contribuições

- Extração Semi-Automática de Rodovias Usando Programação Dinâmica ..... 3  
*Aluir Porfírio Dal Poz, Peggy Agouris*
- Segmentação e Classificação Supervisionada para Identificar Pastagens Degradadas ..... 10  
*Lucimar Moreira, Eduardo Delgado Assad*
- On the Use of Indicator Simulation for Numerical Predictions with Uncertainty Assessment ..... 16  
*Carlos Alberto Felgueiras, Suzana Druck Fuks, Antônio Miguel Vieira Monteiro, Eduardo Celso Gerbi Camargo*
- Extração de Dados em Sistemas de Informações Ambientais: Arquitetura e Esquema de Metadados ..... 22  
*Hélio Alvaro de Mello Perez, Ana Maria de Carvalho Moura, Astério Kiyoshi Tanaka*
- Extensões ao Modelo OMT-G para Produção de Esquemas Dinâmicos e de Apresentação ..... 29  
*Clodoveu A. Davis Jr., Alberto H. F. Laender*
- Towards a Unified Framework for Geographical Data Models ..... 37  
*Gilberto Câmara, Antônio Miguel Vieira Monteiro, João Argemiro Paiva, Jonas Gomes, Luiz Velho*
- Ontologias e Interoperabilidade Semântica entre SIGs ..... 45  
*Frederico Fonseca, Max Egenhofer, Karla A. V. Borges*
- Trading e Agentes em Sistemas de Informações Geográficas Distribuídos ..... 53  
*Sônia Virgínia Alves França, Carlos André Guimarães Ferraz*
- Arquiteturas Cliente-Servidor para Bibliotecas Geográficas Digitais ..... 60  
*José Roberto Osses, João Argemiro de Carvalho Paiva, Gilberto Câmara*
- Modelagem Geométrica 3D e Análise Espacial de Sistemas Cársticos com o uso de OpenGL e SQL ..... 68  
*José Antonio Ferrari, Alain Mangin*

Servlets e COM para Visualização de Dados Geográficos na Web .....	76
<i>Arlindo Cardarett Vianna, Anselmo Cardoso de Paiva, Marcelo Gattass</i>	
Uma Metodologia para Geração de Modelos de Elevação a partir de Curvas de Nível .....	82
<i>Roberto de Beauclair Seixas, Luiz Henrique de Figueiredo, Claudio Antonio da Silva, Paulo Cezar Pinto Carvalho</i>	
Cartografia e Sistemas de Informação Geográfica como Sistemas Semióticos: Uma Análise Comparativa .....	88
<i>Alysson Bolognesi Prado, M. Cecília C. Baranauskas, Cláudia M. Bauzer Medeiros</i>	
Aplicação de Raciocínio Baseado em Casos a Sistemas de Apoio à Decisão Ambiental .....	96
<i>Daniel dos Santos Kaster, Claudia Bauzer Medeiros, Heloísa Vieira da Rocha</i>	
Exchange Mechanism for Global Mapping .....	104
<i>Teresa Cristina Veiga</i>	
Fatores Determinantes de Desempenho de Métodos de Acesso Multidimensionais .....	112
<i>Ricardo Rodrigues Ciferri, Ana Carolina Salgado</i>	
A Comparison among Different Synchronized Tree Traversal Algorithms for Spatial Joins .....	120
<i>Sandro Danilo Gatti, Geovane Cayres Magalhães</i>	
TerraLib: Technology in Support of GIS Innovation .....	126
<i>Gilberto Câmara, Ricardo Cartaxo Modesto de Souza, Bianca Maria Pedrosa, Lúbia Vinhas, Antônio Miguel Vieira Monteiro, João Argemiro Paiva, Marcelo Tilio de Carvalho, Marcelo Gattass</i>	

# Índice de autores

Alain Mangin .....	68
Alberto H. F. Laender .....	29
Aluir Porfírio Dal Poz .....	3
Alysson Bolognesi Prado .....	88
Ana Carolina Salgado .....	112
Ana Maria de Carvalho Moura .....	22
Anselmo Cardoso de Paiva .....	76
Antônio Miguel Vieira Monteiro .....	16, 37, 126
Arlindo Cardarett Vianna .....	76
Astério Kiyoshi Tanaka .....	22
Bianca Maria Pedrosa .....	126
Carlos Alberto Felgueiras .....	16
Carlos André Guimarães Ferraz .....	53
Claudia Bauzer Medeiros .....	88, 96
Claudio Antonio da Silva .....	82
Clodoveu A. Davis Jr. ....	29
Daniel dos Santos Kaster .....	96
Eduardo Celso Gerbi Camargo .....	16
Eduardo Delgado Assad .....	10
Frederico Fonseca .....	45
Geovane Cayres Magalhães .....	120
Gilberto Câmara .....	37, 60, 126
Heloísa Vieira da Rocha .....	96
Hélio Alvaro de Mello Perez .....	22
Jayant Sharma .....	2
Jonas Gomes .....	37
José Antonio Ferrari .....	68
José Roberto Osses .....	60
João Argemiro Paiva .....	37, 60, 126
Karla A. V. Borges .....	45
Lucimar Moreira .....	10
Luiz Henrique de Figueiredo .....	82
Luiz Velho .....	37
Lúbia Vinhas .....	126
M. Cecília C. Baranauskas .....	88
Marcelo Gattass .....	76, 126
Marcelo Tilio de Carvalho .....	126
Max Egenhofer .....	1, 45
Paulo Cezar Pinto Carvalho .....	82

Peggy Agouris .....	3
Ricardo Cartaxo Modesto de Souza .....	126
Ricardo Rodrigues Ciferri .....	112
Roberto de Beauclair Seixas .....	82
Sandro Danilo Gatti .....	120
Suzana Druck Fuks .....	16
Sônia Virgínia Alves França .....	53
Teresa Cristina Veiga .....	104



# Computational Models for Spatio-Temporal Movement

MAX EGENHOFER

Department of Spatial Information Science and Engineering, University of Maine  
max@spatial.maine.edu

**Abstract.** Movement is change in location over time. A geospatial lifeline model movement as a time-stamped record of locations that an individual has occupied over a period of time. This paper distinguishes three types of geospatial lifelines. Lifeline steps capture discrete movement associated, for example, with modeling residential histories. Lifeline threads model the likely locations that an object might pass through while continuously moving between two locations. Lifeline necklaces offer different semantics as they describe the set of all possible locations that an individual might pass through when moving between A and B. We demonstrate the rich set of methods to query lifeline necklaces and their elementary building blocks, lifeline beads, through analytical geometry operations upon cones.

## Oracle Spatial: Experiences with Extensible Databases

JAYANT SHARMA

Oracle

jsharma@us.oracle.com

**Abstract.** Object relational databases have become the new standard for addressing the growing data management and analysis needs of non-traditional database applications such as Spatial Information Systems. Two critical issues must be resolved in order to effectively meet the requirements of these applications. They are: (i) representation and (ii) content based search of spatial data. Oracle Spatial addresses these issues by providing a spatial data type (SDO\_GEOMETRY), an indexing mechanism, and functions/operators on the spatial data type. It enables spatial data to be stored, accessed, indexed, and nalyzed quickly and efficiently in an Oracle8i database. This gives application developers the facility to store all locational information within an ndustry standard database server without having to resort to custom-built external indexes and functions to get the performance they need. Users of spatial data also gain access to standard Oracle8i features such as bigger database size limits, faster backup and recovery, and Java in the database.

# Extração Semi-Automática de Rodovias Usando Programação Dinâmica

ALUIR PORFÍRIO DAL POZ<sup>1</sup>

PEGGY AGOURIS<sup>2</sup>

<sup>1</sup>UNESP- Universidade Estadual Paulista, Rua Roberto Simonsem, 305, 19060-9000 Presidente Prudente, SP, Brasil

[aluir@prudente.unesp.br](mailto:aluir@prudente.unesp.br)

<sup>2</sup>Dept. of Spatial Information Science and Engineering - University of Maine  
5711 Boardman Hall, Orono, Maine 04469-5711, USA

[peggy@spatial.maine.edu](mailto:peggy@spatial.maine.edu)

**Resumo.** Este artigo apresenta um método de extração semi-automática de rodovias em imagens de satélite ou em fotografias aéreas digitalizadas, visando a captura e atualização de base de dados de SIG (Sistema de Informação Espacial). Basicamente, um modelo genérico de rodovia é formulado usando 6 propriedades básicas de rodovia, sendo resolvido através do algoritmo de programação dinâmica. Alguns pontos sementes descrevendo grosseiramente a rodovia devem ser fornecidos pelo operador. Os fundamentos matemáticos do método e os resultados experimentais são apresentados e discutidos neste artigo.

**Abstract.** This paper presents a semi-automatic road extraction approach from aerial and satellite images for GIS (Geographic Information System) road network updating and capture. Basically, a generic road model is formulated using six basic road properties, which is solved sequentially by a dynamic programming algorithm. A few seed points describing coarsely the road need to be provided by the operator. The mathematical fundamentals of the methods and experimental results are presented and discussed in this paper.

## 1 Introdução

A coleta e a atualização de dados para SIG são geralmente realizadas através de fotografias aéreas e imagens de satélite. Diversas pesquisas sobre extração automática e semi-automática de rodovias em imagens digitais têm sido realizadas por especialistas das áreas de Visão Computacional e Fotogrametria Digital, desde a década de 70. Trabalhos pioneiros nesta área são encontrados em Bajcsy e Tavakoli (1976) e Quam (1978). O critério mais comum usado para classificar os métodos de extração de rodovias está relacionado com a necessidade ou não de interferência do operador humano com o sistema de extração, dando origem aos denominados métodos automáticos e semi-automáticos. É possível identificar dois esquemas básicos de extração semi-automática de rodovias. Num primeiro, o operador necessita fornecer a posição e a direção da rodovia em um ponto inicial, a partir do qual o restante da rodovia é extraída. Normalmente se empregam algoritmos baseados na análise de consistência de perfis da superfície intensidade extraídos transversalmente ao eixo da rodovia (Quam, 1978; Vosselman e de Knecht, 1995) ou na análise de bordas anti-paralelas (Nevatia and Babu, 1980, Dal Poz et al., 2000<sup>a</sup>) ou na combinação entre ambos princípios

(McKeown e Denlinger, 1988). Num segundo, o operador necessita fornecer alguns pontos sementes descrevendo grosseiramente a forma e a posição da rodovia. O polígono inicial pode então ser refinado através de algoritmos baseados na otimização global por programação dinâmica ou no princípio do contorno ativo ou *snakes* (Gruen e Li, 1997, Li, 1997, Dal Poz et al., 2000<sup>a</sup>). Se estes métodos forem combinados com métodos de detecção automática de pontos sementes, como um encontrado em Zlotnick e Carnine (1993), a metodologia resultante poderia ser considerada como automática. Entretanto, uma metodologia originalmente desenvolvida para ser totalmente automática é apresentada em Baumgartner et al. (1997, 1999). Neste método, contexto (isto é, relações entre as rodovias e outros objetos) e análise de bordas ao longo do espaço-escala são combinados para extrair automaticamente as rodovias. Outros exemplos de métodos automáticos são encontrados em Barzohar e Cooper (1996) e Ruskoné et al. (1994).

A ausência de uma teoria unificada de visão computacional tem levado ao surgimento de uma grande variedade de técnicas. Como, até então, nenhuma solução automática provou ser competitiva frente a habilidade natural do operador humano, soluções semi-automáticas têm sido propostas combinando a habilidade de

interpretação do operador humano com a capacidade de medida do computador. A metodologia a ser apresentada neste artigo baseia-se numa solução de um modelo matemático geral de rodovia através do algoritmo de programação dinâmica proposta em Li (1997).

Na seção 2 são apresentados os fundamentos teóricos do método para extração semi-automática de rodovias através do algoritmo de programação dinâmica. Os resultados experimentais obtidos são apresentados na seção 3. Finalmente, a seção 4 apresenta as principais conclusões.

## 2 Extração de rodovias usando programação dinâmica

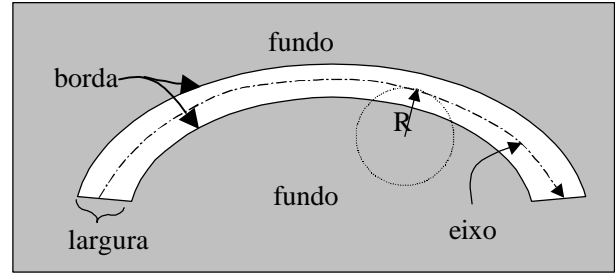
A metodologia de extração a ser apresentada consiste basicamente em resolver um modelo genérico de rodovia através de uma estratégia de otimização global usando programação dinâmica. Para inicializar o processo de extração é necessário que um operador forneça alguns pontos sementes descrevendo grosseiramente a rodovia. Este procedimento dá origem a um polígono inicial, a ser refinado através de uma estratégia iterativa. Em cada iteração novos vértices aproximados são interpolados linearmente entre os pares de vértices preexistentes. O polígono resultante é então refinado resolvendo o modelo geral de rodovia usando a estratégia de otimização global através de programação dinâmica. O processo converge quando os novos vértices inseridos não contribuem mais com o refinamento do polígono descrevendo a rodovia.

Nas próximas subseções serão apresentados o modelo geral de rodovia e alguns detalhes acerca de sua solução usando o algoritmo de programação dinâmica. Devido à limitação de espaço, o algoritmo de programação dinâmica não é apresentado neste artigo. Para uma compreensiva leitura sobre esta técnica, recomenda-se consultar Ballard e Brown (1982) e Li (1997).

### 2.1 Modelagem matemática de uma rodovia

A tarefa de extração de rodovias em uma imagem digital fica mais viável à medida que mais conhecimentos sobre rodovias forem incorporados num modelo genérico.

A modelagem matemática (Li, 1997; Grün; Li, 1997) da entidade rodoviária (figura 1) é desenvolvida com base em seis propriedades básicas, sendo três fotométricas e três geométricas. Primeiro, as propriedades fundamentais são apresentadas, juntamente com as respectivas formulações matemáticas. Posteriormente, o modelo geral é apresentado.



**Figura 1** Elementos básicos de uma rodovia

A primeira propriedade é fotométrica e estatuí que os pixels de uma rodovia são mais claros que os de fundo. Isto significa que uma rodovia em uma imagem digital pode ser considerada como uma faixa estreita e contínua de alta intensidade de brilho, margeada por regiões (fundos) de baixa intensidade. Portanto, a soma dos quadrados dos tons de cinza da imagem ( $G(x,y)$ ) ao longo da rodovia é máxima, isto é,

$$E_{p1} = \sum_i G^2(x_i, y_i) \rightarrow \text{Máx} \quad (1)$$

A segunda propriedade é também fotométrica e fundamenta-se no conhecimento a priori de que o material de rolagem, geralmente asfalto ou concreto, não varia muito em curtas distâncias. Isso implica em respostas espectrais similares em segmentos curtos. Portanto, uma expressão matemática baseada no conceito de homogeneidade pode ser escrita, como segue,

$$E_{p2} = \sum_i \sum_j [G(x_j, y_j) - G_{d^m}^{DS_i}]^2 \rightarrow \text{Mín} \quad (2)$$

onde,  $G_{d^m}^{DS_i}$  é dada pela expressão 3.

$$G_{d^m}^{DS_i} = \frac{\sum_j G(x_j, y_j)}{DS_i} \quad (3)$$

Nas equações 2 e 3, o índice  $j$  é usado para indexar pontos ao longo de cada vetor ( $i, i+1$ ). A distância  $DS_i$  é dada por,

$$DS_i = \sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2} \quad (4)$$

A terceira propriedade é uma generalização das 2 primeiras. Visto que uma rodovia é uma feição linear de alta intensidade de brilho, e que os pontos mais distantes dessa curva têm menor influência na definição da mesma, pode-se escrever a seguinte expressão matemática para representar esta propriedade,

$$E_{p3} = \sum_i e^{-d_i^2} G^2(x_i, y_i) \rightarrow \text{Máx} \quad (5)$$

onde,

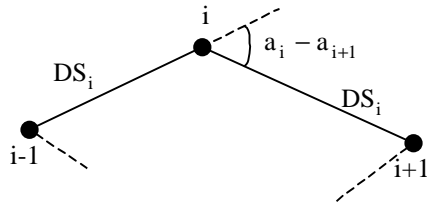
- $d_i$  é a distância entre um ponto  $(x_i, y_i)$  e a curva discreta representando a rodovia; e
- $e^{-d_i^2}$  é uma função gaussiana, inversamente proporcional ao quadrado de  $d_i$ .

A quarta propriedade reflete uma característica geométrica fundamental de uma rodovia, isto é, a suavidade. Por questões práticas, uma rodovia é normalmente composta por segmentos retos e curvas suaves, geralmente circulares. Li (1997) apresenta a seguinte fórmula (figura 2),

$$E_g = \sum_i [2 - 2\cos(a_i - a_{i+1})] / DS_i \rightarrow \text{Mín} \quad (6)$$

onde,

- $a_i - a_{i+1}$  é o ângulo de deflexão entre dois vetores sucessivos da entidade discreta representando a rodovia ; e
- $DS_i$  é a distancia entre o vértice  $i$  e  $i-1$  ou  $i+1$ .



**Figura 2** Curvatura no vértice  $i$

A quinta propriedade estatui que a curvatura local de uma rodovia possui um limite máximo, isto é,

$$C_{gi} = |a_i - a_{i+1}| < T \quad (7)$$

onde,  $T$  é um limiar.

A sexta propriedade está relacionada com o fato da largura da rodovia não variar significativamente. Entretanto, esta propriedade já está implícita na 3ª propriedade.

Considerando agora que uma rodovia pode ser representada por um polígono de  $n$  vértices, o seguinte par de equações pode ser usado para modelar o objeto rodovia (Li, 1997),

$$E = \sum_{i=1}^{n-1} ((Ep_1 - bEp_2 + cEp_3)[1 + \cos(a_i - a_{i+1})]) / DS_i = \sum_{i=1}^{n-1} E_i(p_{i-1}, p_i, p_{i+1}) \quad (8)$$

$$C_i = |a_i - a_{i+1}| < T, \quad i = 1, \dots, n-1 \quad (9)$$

onde,  $b$  e  $c$  são duas constantes positivas.

Portanto, o modelo matemático de rodovia é composto por 2 funções:

- $E$ : corresponde à função objetivo (eq. 8), podendo ser decomposta em funções  $E_i$  dependendo apenas de três pontos vizinhos  $(p_{i-1}, p_i, p_{i+1})$  do polígono que representa a rodovia; e
- $C_i$ : devido a sua definição (eq. 9), também é denominada de injeção de desigualdade e permite limitar o espaço de busca do problema, pois a função  $E$  é apenas avaliada para pontos que satisfaçam  $C_i < T$ .

O modelo matemático 8 é resolvido sequencialmente através do algoritmo a ser apresentado na próxima seção.

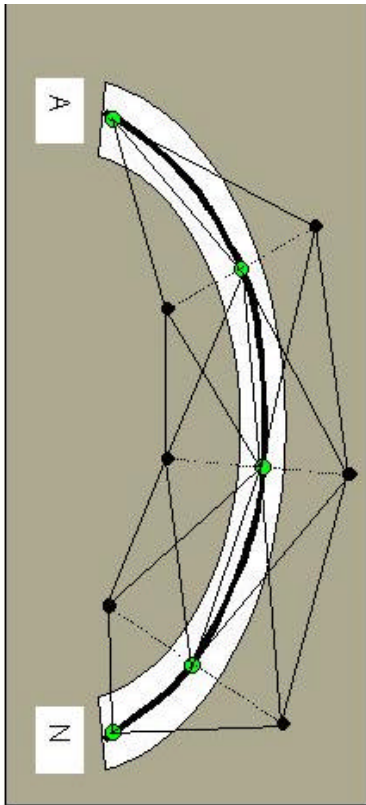
## 2.2 Solução para o modelo matemático geral

O princípio geral da extração de rodovias através da programação dinâmica é mostrada na figura 3. Como mostra esta figura, nem todos os vértices estão inter-relacionados simultaneamente. Nesse caso, a programação dinâmica é um algoritmo eficiente para resolver o problema de otimização (Ballard e Brown, 1982). O problema consiste então em encontrar o melhor caminho entre A e N. Pode ainda ser observado na figura 3 que:

- a entidade a ser extraída é descrita por um polígono de  $n$  vértices;
- os vértices que definem o caminho ótimo, isto é, a rodovia extraída, são calculados a partir das respectivas posições aproximadas. Cada vértice move em torno de sua respectiva posição aproximada, sendo que a área de busca é

normalmente tomada como uma pequena janela. As várias possibilidades que os vértices aproximados podem assumir implicam na geração de uma série de polígonos; e

- a seqüência de vértices que define o caminho ótimo é tal que maximiza a função objetivo  $E$  e satisfaz a injunção  $C_i < T$ .



**Figura 3** Princípio de método

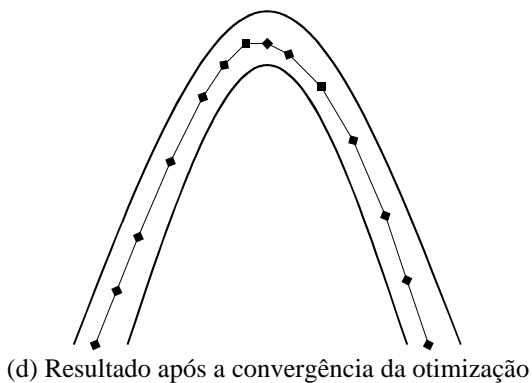
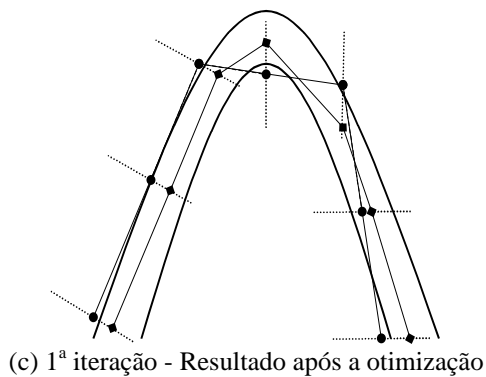
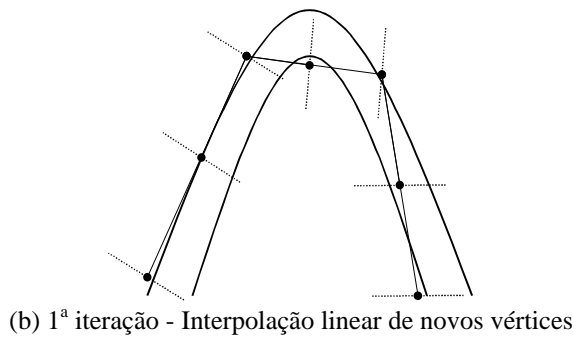
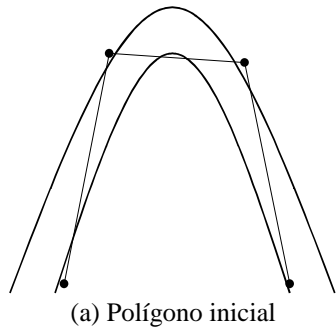
Como já foi mencionado, geralmente a janela de busca é bidimensional. Sendo  $n$  o número de vértices e  $m$  o número de possibilidades que cada vértice pode assumir dentro da janela de busca, o esforço computacional será da ordem de  $O(nm^3)$ . Portanto, o uso de janelas maiores para aumentar o raio de convergência aumentaria significativamente o esforço computacional. Duas estratégias podem ser exploradas para diminuir o número de elementos ( $m$ ) na janela de busca (Li, 1997):

- **uso de uma janela de busca unidimensional:** nesse caso, os elementos da janela de busca são tomados numa direção ortogonal ao polígono inicial no ponto  $P_i$ . Esse procedimento possibilita manter o raio de convergência, mas o esforço computacional é bem menor; e
- **uso de janelas de resolução variada:** esta estratégia está relacionada à teoria da pirâmide de imagem (Ballard e Brown, 1982), onde a base corresponde à imagem original e as outras camadas, na direção do topo, correspondem à mesma imagem, mas com resoluções cada vez menores. Uma solução alternativa, computacionalmente mais atrativa e que produz bom resultado, baseia-se na computação dos elementos da janela de busca em intervalos diferentes. Como será mostrado, a seguir, o processo de extração é iterativo, permitindo o refinamento sucessivo do polígono inicial. Portanto, intervalos com um maior número de pixels são usados no início, permitindo a obtenção de um grande raio de convergência. À medida que o processo de convergência vai sendo atingido, o intervalo entre pontos na janela de busca vai sendo diminuído também, até atingir o valor do sub-píxel, permitindo a obtenção de um resultado final com alta precisão.

À primeira vista, a descrição de uma rodovia através de um polígono pode ser realizada com vértices equidistantes. Como nesse caso as posições dos vértices não estão relacionadas com a forma da curva, a quantidade de pontos é geralmente bem maior que a necessária para descrevê-la. Uma solução mais adequada é apresentada em Li (1997). Entretanto, algumas modificações foram realizadas na versão original do algoritmo. A figura 4 ilustra o princípio do processo de otimização usando programação dinâmica. Inicialmente, são fornecidos vários pontos sementes, cujas localização e distribuição são grosseiras (figura 4(a)). Ligando-se estes pontos, obtém-se a primeira aproximação para a rodovia. O processo de refinamento do polígono inicial é iterativo, como segue:

- **1ª iteração:** entre cada par de pontos sementes é inserido um novo ponto através de interpolação linear (figura 4(b)). As linhas tracejadas (figuras 4(b) e 4(c)) representam as janelas de busca. O procedimento de otimização através da programação dinâmica gera uma nova aproximação para a rodovia (figura 4(c)); e
- **demais iterações:** procedimentos idênticos são repetidos nas demais iterações.

O processo converge quando numa determinada iteração todos os novos pontos inseridos são colineares aos pontos otimizados na iteração anterior. O resultado após a convergência é ilustrado na figura 4(d).

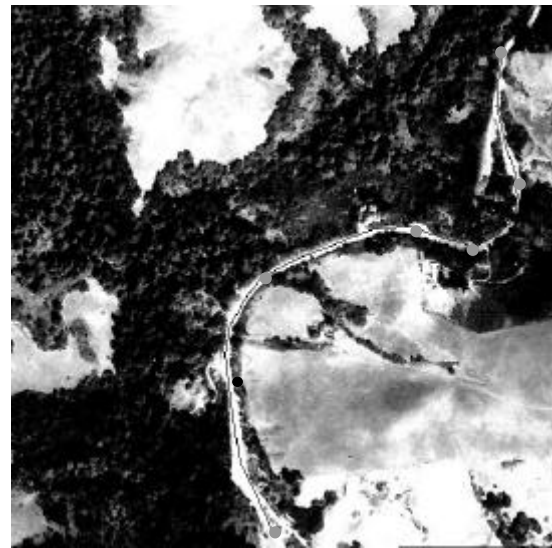


**Figura 4** Ilustração do processo de otimização

### 3 Resultados e discussões

O método apresentado na seção 2 foi implementado em linguagem Borland C++ para o ambiente Windows-PC.

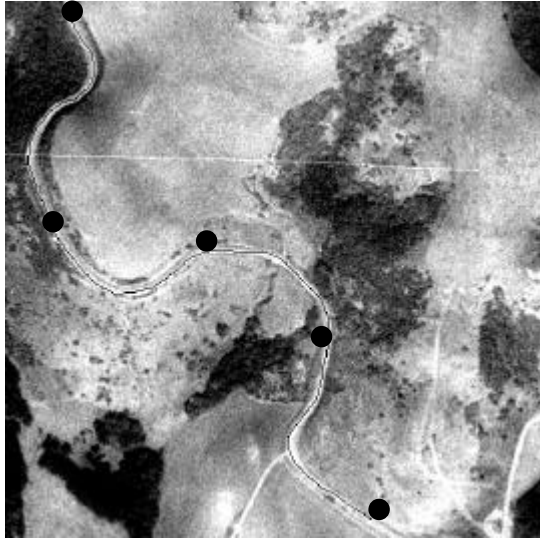
Foram utilizadas duas imagens reais (500x500 pixels) de média resolução, sendo que cada pixel corresponde a aproximadamente 2m no terreno. Essas imagens foram obtidas junto a *home page* do Sistema BADGER (<http://badger.parl.com/>), que é composta por um *software* para a distribuição via Internet de mapas e imagens da Baía de São Francisco, Califórnia, EUA.



**Figura 5** Resultados obtidos com a Imagem-1

O primeiro experimento foi realizado com a Imagem-1 (figura 5), cujo conteúdo é bastante desfavorável. A largura da rodovia é bastante irregular e várias obstruções encontram-se presentes numa região de curvatura acentuada. Os resultados obtidos e os pontos sementes usados foram sobrepostos sobre a imagem processada. Esses resultados podem ser considerados bastante satisfatórios, pois são poucos os trechos que provavelmente estariam em desacordo com a visão natural. Além disso, a rodovia em análise distancia-se consideravelmente do modelo ideal de rodovia - por exemplo, largura irregular, bordas irregulares, presença de obstáculos e sombras, etc. O fator inerente a este exemplo que beneficia o algoritmo de programação dinâmica é o bom contraste entre a superfície de rolagem e as adjacências. Nesse caso, as irregularidades geométricas (largura, borda) não são fatores críticos. A transposição de irregularidades na superfície de rolagem (por exemplo, árvores e sombras) é facilitada pela estratégia adotada envolvendo a inserção de novos vértices a cada iteração.

Embora todos os vértices já calculados, inclusive os que estão sendo inseridos, permaneçam livres para variar, maior restrição é imposta à medida que as iterações são realizadas. Dessa forma, a colocação de pontos sementes para evitar, no início, que novos vértices caiam sobre ou próximos aos obstáculos diminui a influência dos mesmos na otimização global.



**Figura 6** Resultados obtidos com a Imagem-2

A Imagem-2 (figura 6) apresenta uma rodovia principal com baixíssimo contraste em relação às adjacências. Os resultados obtidos e os pontos sementes usados foram sobrepostos sobre a imagem de entrada. Este resultado pode ser considerado excelente, podendo inclusive ser considerado bem razoável por um observador humano. Embora não haja obstáculos e sombras, em geral a rodovia é muito mal definida, existindo um trecho onde é difícil para um observador humano definir exatamente onde passa a rodovia. Vale também ressaltar a boa qualidade do eixo extraído. Apesar do baixo contraste da rodovia em relação às adjacências, existe um padrão não bem definido, cujo máximo da função objetivo ocorre nos pontos que definem a rodovia extraída. Nota-se também que em certos trechos existem regiões estreitas mais clara na faixa mais central da rodovia, fator que provavelmente deve ter contribuído para o bom resultado obtido.

#### 4 Conclusões

Neste artigo foi apresentado e avaliado experimentalmente um método para extração de rodovias em imagens de satélite ou fotografias aéreas digitalizadas. Este método pode ser usado na captura semi-automática

de dados para base de dados de SIG. Neste caso, o operador humano realiza a tarefa de reconhecimento da rodovia e identifica alguns pontos sementes descrevendo grosseiramente a forma e a posição da rodovia. A identificação de pontos sementes pode também ser realizada automaticamente, levando a um processo de extração de rodovias totalmente automatizado. Uma outra aplicação da metodologia apresentada é na verificação de rodovias preexistentes, numa base de dados de SIG, em imagens recentes (Dal Poz e Agouris, 2000<sup>b</sup>). Nesse caso, os pontos sementes necessários podem ser extraídos automaticamente da própria base de dados usando critérios topológicos e geométricos.

Para o tipo de imagem testada, i. e., imagens de média resolução, o método mostrou ser robusto frente à presença de obstruções, sombras, baixo contraste e outras irregularidades. A qualidade dos resultados foram compatíveis com os que seriam obtidos por um operador humano. Apesar de ser necessária uma pequena intervenção do operador para inicializar o processo, a tarefa cansativa e demorada de realizar medidas precisas da rodovia é deixada a cargo do algoritmo de programação dinâmica.

A continuidade desta pesquisa deverá abranger três direções. Primeiro, como as características do método apresentado são mais apropriadas para os casos envolvendo imagens de baixa e média resolução, algumas melhorias serão necessárias para os casos envolvendo imagens de alta resolução. Segundo, visando uma melhor avaliação do potencial do método, será necessária a realização de um maior número de experimentos envolvendo, basicamente, imagens de resolução e conteúdo diferentes. Finalmente, tendo em vista as aplicações do método, serão investigadas as possibilidades para a inicialização automática do processo de extração.

#### Agradecimentos

Este trabalho foi apoiado pela FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo, Brasil), Processo no. 1998/15553-3, e NSF (*National Science Foundation, USA*), Processo no. IRI-9702233.

#### Referências bibliográficas

- [1] R. Bajcsy, M. Tavakoli, "Computer recognition of roads from satellite picture," *IEEE Transactions on Systems, Man, and Cybernetics* 6 (1976), 76-84.
- [2] D. H. Ballard, C. M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [3] M. Barzohar, D. Cooper, "Automatic finding of main roads in aerial images by using geometric-stochastic



- models and estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (1996), 707--721.
- [4] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, H. Ebner, "Update of road in GIS from aerial imagery: verification and multi-resolution extraction," In: *Int. Arch. Photogramm. Remote Sensing*, 53--58, 1996.
- [5] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, "Multi-resolution, semantic objects, and context for road extraction," In: *Workshop on Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*. Birkhäuser Verlag, Basel (Schweiz), 140--156, 1997.
- [6] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, H. Ebner, "Automatic road extraction in rural areas," In: *Int. Arch. Photogramm. Remote Sensing*, 107--112, 1999.
- [7] A. P. Dal Poz, S. Gyftakis, P. Agouris, "Semi-automatic road extraction: Comparison of methodologies and experiments", In: *DC 2000 ASPRS Annual Conference*, Washington, DC, USA, May 2000<sup>a</sup> (no prelo).
- [8] A. P. Dal Poz, P. Agouris, "Georeferenced road extraction and formulation of hypothesis for new road segments ", In: *SPIE's 14<sup>th</sup> Annual International Symposium on Aerospace/Defence Sensing, Simulation, and Controls - Conference Automated Geo-Spatial Image and Data Exploitation*, Orlando, FL, USA, April 2000<sup>b</sup> (no prelo).
- [9] A. Gruen, H. Li, "Semi-automatic linear feature extraction by dynamic programming and LSB-Snakes," *Photogrammetric Engineering & Remote Sensing* 63 (1997), 985--995.
- [10] H. Li, *Semi-automatic road extraction from satellite and aerial images*. PhD thesis, Report No. 61, Institute of Geodesy and Photogrammetry, ETH-Zurich, Switzerland, 1997.
- [11] D. M. McKeown, J. L. Denlinger, "Cooperative methods for road tracking in aerial imagery," In: *Workshop of Computer Vision and Pattern Recognition*, 662--672, 1988.
- [12] R. Nevatia, K. R. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing* 13 (1980), 257--269.
- [13] L. H. Quam, "Road tracking and anomaly detection in aerial imagery," In: *Image Understanding Workshop*, 51--55, 1978.
- [14] R. Ruskoné, S. Airault, O. Jamet, "A road extraction system using the connectivity properties of the network," *Zeitschrift für Photogrammetrie und Fernerkundung* 5 (1994), 174--180.
- [15] G. Vosselman, J. de Knecht, "Road tracing by profile matching and Kalman filtering," In: *Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Gruen, A. O. Kuebler, and P. Agouris (eds), Birkhäuser, Basel-Boston-Berlin, 265--274, 1995.
- [16] A. Zlotnick, P. D. Carnine, "Finding roads seeds in aerial images," *CVGIP: Image Understanding* 57(1997), 243--260.

# Segmentação e classificação supervisionada para identificar pastagens degradadas.

Lucimar Moreira

Eduardo Delgado Assad.

Embrapa Cerrados, BR 020, Km 18. 73.301-970 - Planaltina – DF

E. mail: [lucimar@cpac.embrapa.br](mailto:lucimar@cpac.embrapa.br)

[Assad@cpac.embrapa.br](mailto:Assad@cpac.embrapa.br)

**Abstract-** One of the major environmental issues in the Brazilian Cerrado region is the identification of the levels of degradation in the cultivated pastures. About 50 million hectares of Cerrado is currently covered by cultivated pastures, with different degradation levels. This paper assesses the performance of supervised classification per region for identifying different degradation levels in cultivated pastures. The study area is located in the córrego Lamarão Watershed, Federal District. The satellite images were processed by using the SPRING/INPE Geographical Information System. We were able to identify five levels of cultivated pasture degradation.

## Introdução

Na década de 1970, houve uma grande expansão da produção pecuária devido, especialmente, ao baixo valor das terras, a oferta de crédito e o surgimento de espécies forrageiras com alta capacidade de adaptação ao clima e a baixa fertilidade dos solos. Atualmente, a atividade de pecuária bovina é responsável por 44% do rebanho bovino nacional. Este rebanho tem nas pastagens cultivadas sua principal fonte alimentar. Por outro lado, o uso indiscriminado das áreas de pastagem com elevada carga animal e a desatenção às necessidades do requerimento na correção e fertilização dos solos, impôs um processo de extrativismo, pela exportação de produto animal, condicionando a perda da capacidade produtiva das pastagens. Entende-se por degradação de pastagens o processo evolutivo de perda de vigor, de produtividade, de capacidade de recuperação natural das pastagens para sustentar os níveis de

produção e de qualidade exigidos pelos animais, assim como, a de superar os efeitos nocivos de pragas, doenças e invasoras, culminando com a degradação avançada dos recursos naturais, em razão de manejos inadequados (Macedo & Zimmer, 1993). Este trabalho teve como objetivo verificar o desempenho da segmentação e classificação supervisionada por regiões, implementadas no SPRING/INPE, para identificar áreas com diferentes fases de degradação de pastagem.

Área de estudo: A microbacia hidrográfica do córrego Lamarão localiza-se na parte Sudeste do Distrito Federal, entre as coordenadas geográficas 15° 54' 50" a 16° 01' 10" de latitude Sul e 47° 25' 30" a 47°36' 30" de longitude Oeste. Possui área aproximada de 9.630 ha.

## Materiais e métodos

Imagens TM Landsat 5 órbita 221/71, referentes à 10/10/1987, 31/08//1996 e 30/05/97;

Mapa planialtimétrico 1:100.000 da Diretoria do Serviço Geográfico;

Levantamento semi-detalhado dos solos da Bacia do Rio Jardim, na escala 1:50.000;

Software SPRING/INPE.

De acordo com Spain & Gualdrón (1988) e Barcellos (1997) o parâmetro cobertura vegetal representa um dos elementos de discriminação da fase de degradação das pastagens. A cobertura do solo, associada a outros componentes bióticos como revegetação por espécies nativas e exóticas e termiteiros constitui indicador do processo de perda da capacidade produtiva das pastagens. As fases de degradação de pastagem, segundo Spain & Gualdrón (1988) estão apresentadas na Tabela 1.

TABELA 1. Fases de degradação de pastagens segundo os parâmetros limitantes e seu nível de degradação.

Fases de degradação	Parâmetros limitantes	Deterioração
1	Vigor e qualidade	Leve
2	Fase 1 + baixa população	Moderado
3	Fase 1 + 2 + e invasoras	Forte
4	Fase 1 + 2 + 3+ formigas e cupins	Muito forte
5	Fase 1 + 2 + 3 + 4 e baixa cobertura do solo	Muito forte

FONTE: Adaptado de Spain & Gualdrón (1988).

Segmentação de imagem - Uma técnica de segmentação de imagens foi desenvolvida para o sistema SPRING e possui três funções principais: *Segmentação, Treinamento e Classificação*. A

execução do processo de segmentação de imagens exige a definição das bandas espectrais e dos limiares de similaridade e de área. Esses parâmetros são responsáveis pela qualidade da segmentação que servirá de base para a classificação temática posterior. Foram definidas as bandas espectrais 1 a 5 e 7 do Landsat TM 5. Os critérios de aceitação e rejeição dos limiares foram analisados em decorrência da separação coerente dos tons de cinza visualmente distintos. Como exemplo de trabalhos realizados utilizando limiares diferentes, citam-se Batista *et al.* (1994) que utilizaram os valores de 8 e 10 para os limiares de similaridade e 4 e 9 para os de área, para estimar a extensão do desflorestamento. Na avaliação do desempenho das redes neurais para o monitoramento do desflorestamento Machado *et al.* (1994) utilizaram o valor 10, tanto para o limiar de similaridade como para o de área. Para o estudo temporal da cobertura vegetal e uso da terra Watrin (1994) utilizou os limiares de similaridade 8 e 10 e o limiar 10 de área. Saiz & Valério Filho (1995) utilizaram a técnica de transformação IHS e Segmentação para caracterizar o uso da terra e a cobertura vegetal, na região de Tandil, Província de Buenos Aires, Argentina. Os valores de similaridade e de área usados foram de 8 e 10, respectivamente. Nascimento (1997) realizou vários testes de segmentação para mapear unidades de paisagem na Amazônia e o melhor resultado foi obtido com os valores de 12 para similaridade e 10 para o valor de área.

A avaliação do processo de segmentação foi realizada por meio da comparação visual entre a imagem segmentada e pela sobreposição da imagem realçada. Essa metodologia foi adotada, pois de acordo com Cross *et al.* (1988); essa é uma forma

qualitativa, mas eficiente para avaliar a segmentação.

Após a avaliação das imagens segmentadas e definido o par de limiares (similaridade e área) mais adequado, fez-se a classificação dessa imagem que requer que do analista a especificação do limiar de aceitação para a separação ou o agrupamento de classes distintas.

A principal vantagem de trabalhar com classificação por regiões é que essa técnica fornece informações de natureza espacial, que não podem ser extraídas individualmente de pixels e que são normalmente consideradas durante o processo de análise visual de imagens.

Classificação da imagem. Para a classificação da imagem segmentada, foram selecionadas algumas áreas de treinamento para cada classe e fornecidas ao classificador para caracterização individual das classes temáticas. Após adquirir amostras para todas as classes, as imagens foram classificadas, utilizando a distância de Bhattacharyya.

## Resultados e discussão

Para a segmentação o limiar mais adequado de área e similaridade foi 8/8. A exatidão do mapeamento foi realizada para a imagem de 1997, devido à existência de dados de campo. Em relação à imagem de 1987, o resultado foi verificado junto aos técnicos da EMATER-PAD-DF. Com a verificação de campo, observou-se que o resultado da classificação foi satisfatório. Isto pode ter ocorrido pela divisão da área em diversas classes e pelo número de regiões amostradas.

A partir da segunda metade da década de 70, a cobertura vegetal natural que predominava nos

anos anteriores foi substituída pela agricultura e pela pastagem cultivada. Essa modificação no uso da terra deu-se após a implantação do Programa de Assentamento Dirigido do Distrito Federal (PAD-DF). A interpretação das imagens analisadas neste trabalho identificaram diferentes classes de uso da terra e 5 fases de degradação de pastagens. A Tabela 2 apresenta as classes de uso da terra e cobertura vegetal natural para os anos de 1987 e 1997, em ha.

Classe de Uso da terra	1987	1997
Mata	826,03	1025,94
Cerradão	1,17	3,92
Cerrado	193,07	85,86
Campo	152,71	-
Agric. anual	4771,10	4389,24
Agric. irrigada	68,08	575,81
Past. deg. 1	3168,70	1108,87
Past. deg. 2	-	225,94
Past. deg. 3	-	420,70
Past. deg. 4	-	1256,20
Past. deg. 5	-	232,26
Solo exposto	310,28	291,45
Queimada	98,20	-
Desmatada	39,54	-
Barragem	2,63	9,59
Total	9631,51	9625,78

Tabela 2 – Classes de Uso da terra e cobertura vegetal natural, para os anos de 1987 e 1997.

Em 1987, a agricultura anual ocupava praticamente a metade da microbacia (49,5%), seguido pelas pastagens cultivadas (29,3%). A agricultura irrigada ainda não era significativa, havia apenas um pivô-central próximo ao córrego Pindaibal e uma área irrigada por aspersão, localizada nas margens do córrego Açude, totalizando 0,7%.

A mudança mais significativa observada em 1997 foi o aumento das áreas irrigadas por pivô-central (5,9%). As áreas com agricultura anual reduziram

para 45,6%, sendo que a soma dessas totalizou 51,5%.

Das áreas de cerrado restavam apenas 0,9% e em estado de degradação. As áreas com solo exposto tiveram uma pequena diminuição para 3,0%. As áreas com reservatórios d'água aumentaram para 0,1%, localizando-se nos córregos Poço Claro, Barreiro do Meio e Derradeiro Poço. As regiões com Mata de Galeria e de Cerradão aumentaram para 10,6% e 0,04%, respectivamente. Este aumento pode ser explicado pela data da imagem (maio), onde a vegetação próxima a estas regiões, ainda encontravam-se verdes, sendo então englobadas por essas classes. Observou-se também, que as áreas de Mata Seca, neste período encontravam-se verdes, sendo então classificadas como Mata de Galeria. As áreas de campo não foram observadas, provavelmente foram classificadas como Mata.

As áreas anteriormente classificadas como desmatadas foram abandonadas transformando-se em vegetação secundária.

Em 1987, as pastagens cultivadas encontravam-se no nível 1 (deterioração leve) e ocupavam 32,9% do total da área. Os outros níveis de degradação não foram observados. Em 1997 as áreas com pastagem cultivada em bom estado de manejo, nível de degradação 1 (deterioração leve) reduziram para 10,9%, demonstrando que do total plantado, a maior parte, encontrava-se degradada. Verificou-se 2,3% no nível 2 (deterioração moderada), e 2,4% na área degradada 3 (forte). O nível de degradação 4 (muito forte) foi o que totalizou a maior área com 13,0 % e o nível 5 (muito forte), com 4,4%.

O efeito da falta de cobertura do solo pode ser um dos fatores mais graves no tocante ao impacto ambiental. As perdas de solos e nutrientes, associadas a menor capacidade de produção de biomassa condiciona o assoreamento dos mananciais e cursos d'água e certamente descredenciam as pastagens degradadas a desempenhar seu papel captador de carbono.

Segundo Dedecek (1986) em ambiente de Cerrado os solos descobertos podem perder até 59 toneladas de solo/ha/ano, enquanto as pastagens, quando bem formadas, representam o sistema de maior proteção aos riscos de erosão.

As alternativas tecnológicas desenvolvidas para o restabelecimento da capacidade produtiva das pastagens passam, em sua grande maioria, pela correção e fertilização do solo, associado à sua movimentação com implementos agrícolas (Oliveira *et al.*, 1996). Este revolvimento do solo é necessário para a descompactação superficial ou para a implantação de um cultivo anual associado com a pastagem. É importante ressaltar que muitas vezes as áreas destinadas a implantação de pastagens são restritas ou inaptas para essa utilização, o que as tornam inadequadas ao desenvolvimento de agricultura convencional.

A partir dos resultados alcançados, alguns comentários podem ser feitos:

O processo de ocupação e uso da terra foi realizado sem a observação dos critérios básicos, como a aptidão agrícola das terras. A classe pastagem foi dividida em 5 classes, realçando as fases de degradação.

Estes aspectos provocam direta ou indiretamente impactos na região. O uso

inadequado dos solos, com outro tipo de atividades pode rapidamente provocar seu esgotamento, de caráter físico, (acentuando os processos de erosão) ou de caráter químico (reduzindo a porcentagem de matéria orgânica, reduzindo o estoque de nutrientes).

As pastagens, quando não degradadas promovem com alta eficiência:

- ciclagem de nutrientes;
- captação de carbono da atmosfera;
- redução dos processos de erosão.

Neste trabalho, com a identificação de cinco classes de degradação, torna-se possível inserir em cada polígono, quais das atividades listadas acima (ou se todas) estão sendo afetadas. No caso da microbacia do córrego Lamarão, não foram observados processos de erosão acentuados.

### **Conclusão**

De acordo com os resultados obtidos neste trabalho, a utilização das técnicas de segmentação de imagens e classificação supervisionada por regiões, implementadas no SPRING, mostraram-se eficiente na separabilidade das regiões.

Utilizando as imagens TM Landsat e a classificação supervisionada por regiões, foi possível identificar as pastagens com 5 níveis de degradação. É importante ressaltar que muitas vezes as áreas destinadas a implantação de pastagens são restritas ou inaptas para essa utilização, o que as tornam inadequadas ao desenvolvimento de agricultura convencional.

Trata-se portanto de um primeiro esforço no sentido de, além de identificar a classe de uso,

caracterizar e quantificar por nível de degradação. A partir desta identificação, torna-se fundamental estabelecer os procedimentos de radiometria destes alvos para posterior extrapolação espacial em outras áreas de cerrado.

Com os resultados alcançados, abrem-se novas perspectivas para trabalhos futuros, utilizando regiões e situações diferentes, contribuindo para o desenvolvimento do processamento digital de imagem.

### **Referências bibliográficas**

- BARCELLOS, A. O.; VIANA FILHO, A.; BALBINO, L. C.; OLIVEIRA, I. P.; YOKOYAMA, L. P. Produtividade animal em pastagens renovadas em solo arenoso de cerrado. In: REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA DE ZOOTECNIA, 34.,1997. Juiz de Fora, Anais. Juiz de Fora: SBZ, 1997, v.4, p.207-209.
- BATISTA, G. T., MEDEIROS, J. S.; MELLO, E. M. K.; MOREIRA, J. C.; BINS, L. S. A new approach for deforestation assessment. In: INTERNACIONAL SYMPOSIUM ON RESOURCE AND ENVIRONMENTAL MONITORING, 1994, Rio de Janeiro,1994. Proceedings. Rio de Janeiro: ISPRS, 1994. v. 30, p.170-174.
- CROSS, A. M.; MASON, D. C.; DURY, S. J. Segmentation of remotely-sensed images by a split-and-merge process. International Journal of Remote Sensing, v. 9, n.8, p. 1329-1345, 1988.
- DEDECEK, R. A.; RESCK, D. V. S.; FREITAS JÚNIOR, E. Perdas de solo, água e nutrientes

- por erosão em latossolo vermelho escuro dos cerrados e de manejo da palhada do milho. Revista Brasileira de Ciência do Solo, Campinas, v.10, p. 265-272, 1986.
- MACEDO, M. C. M. Pastagens no ecossistema cerrados: pesquisas para o desenvolvimento sustentável. In: REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA DE ZOOTECNIA, 32., 1995. Brasília, Anais. Brasília: SBZ, 1995. p. 28-62.
- MACHADO, R. J.; BARBOSA, V. C.; LIPORACE, F.S.; SANTOS, J. R.; VENTURIERI, A. Deforestation monitoring of the Amazon region using neural networks - a comparison between different photo-interpreters and networks. In: Resource and environmental monitoring: Rio de Janeiro, ISPRS, 1994, v. 30, Part 7b, p.49-55. Proceedings of an International Symposium organized by VII of the International Society of Photogrammetric and Remote Sensing, Rio de Janeiro, Sept.. 26-30, 1994.
- NASCIMENTO, P. S. R. Avaliação de técnicas de segmentação e classificação por regiões em imagens Landsat-TM visando o mapeamento de unidades de paisagem na Amazônia. São José dos Campos: INPE, 1997. 102p. Tese mestrado.
- OLIVEIRA, I. P. ; KLUTHCHCOUSKI, J.;YOKOYAMA, L. P.; DUTRA, L. G. Sistema Barreirão: recuperação/ renovação de pastagens degradadas em consórcio com culturas anuais. Goiânia: EMBRAPA-CNPAP, 1996. 90p (EMBRAPA-CNPAP, Documento, 64).
- SAIZ, C. C.; VALÉRIO FILHO, M. Técnicas de processamento de imagens digitais aplicadas à caracterização de classes de uso do solo e cobertura vegetal em região de clima temperado úmido. In: SIMPÓSIO LATINOAMERICANO DE PERCEPCIÓN REMOTA.7.;1995. Puerto Vallarta. SELPER-México. Latinoamérica Evaluada desde el Espacio. Puerto Vallarta. Memórias. Noviembre, 1995. p.593-600.
- SPAIN, J. M.; GUALDRÓN, R. Degradación y rehabilitación de pasturas. In: VI Reunión del Comité Asesor de la RIEPT. Memórias. Red Internacional de Evaluación de Pastos Tropicales. Veracruz, México 1988. CIAT. p. 269-283.
- WATRIN, O.S. Estudo da dinâmica na paisagem da Amazônia oriental através de técnicas de geoprocessamento. São José dos Campos: INPE, 1994. 151p. Tese mestrado.

# On the Use of Indicator Simulation for Numerical Predictions with Uncertainty Assessment

CARLOS ALBERTO FELGUEIRAS<sup>1</sup>  
SUZANA DRUCK FUKS<sup>2</sup>  
ANTÔNIO MIGUEL VIEIRA MONTEIRO<sup>1</sup>  
EDUARDO CELSO GERBI CAMARGO<sup>1</sup>

<sup>1</sup>INPE/DPI—Instituto Nacional de Pesquisas Espaciais, Caixa Postal 515, 12201 São José dos Campos, SP, Brasil  
{carlos,miguel,eduardo}@dpi.inpe.br

<sup>2</sup>EMBRAPA/CPAC – Empresa Brasileira de Agropecuária, Rodovia Brasília Fortaleza Br 020 Km 18, Planaltina  
Distrito Federal, Brasil  
drucks@ensam.inra.fr

**Abstract.** Raster representations of thematic and numerical spatial attributes are used very commonly in a GIS environment for computational simulation and analysis of spatial processes. This paper addresses the problem of predictions with uncertainty assessment for GIS raster representations created from a set of sample points of spatial attributes. The realizations of a stochastic simulation inference process, over numerical attribute samples, are used in order to infer the attribute values and related uncertainties at non-sampled spatial locations. A case study, with elevation sample data, is presented to illustrate the concepts used in this work.

## 1 Introduction

GIS environment allows one to simulate and analyze different scenarios that can be used to support decisions made about a specific real spatial process. The main idea is to integrate spatial data attribute representations in order to study the spatial process in a computational environment. The final scenarios depend on the data representations and also on the mathematical model used to integrate them. The attribute representations are derived from a set of attribute samples, commonly sample points, obtained in a spatial region of interest. Nonlinear stochastic procedures, based on the indicator kriging approach, can be used to create attribute representations along with uncertainty information related to a set of estimated attribute values. The uncertainty of each representation can be propagated to the resulting scenarios of a spatial process modeling. The resulting uncertainties will qualify the scenarios, or the objects presented in the scenarios, yielding a quantitative information of the risk assumed when a determined scenario is chosen. In this context, this work presents a methodology to create attribute representations, from a set of sample points, using a nonlinear stochastic approach called *indicator simulation*. Furthermore, this work shows how to obtain uncertainty values related to the attribute value inferences created by this methodology. Different uncertainty metrics, based on confidence intervals, will be addressed. A case study for an elevation sample set will be presented to show how the methodology can be applied to

real data and how to use the uncertainty metrics in order to qualify inferences of numerical attribute representations.

## 2 The geostatistical paradigm for attribute inferences with uncertainty assessment

From a geostatistical point of view, the distribution of a spatial attribute in a region  $A \subset \mathfrak{R}^2$  of the earth surface is represented as a random function  $Z(\mathbf{u})$ . For each position  $\mathbf{u} \in A$  the attribute is considered as a random variable (RV) that can assume different values depending on the model of the spatial distribution of  $z(\mathbf{u})$ , i. e., depending on its probability distribution function (pdf). The conditional cumulative distribution function (ccdf) of a continuous RV  $Z(\mathbf{u})$ , conditioned to  $(n)$  sample points  $z(\mathbf{u}_\alpha)$ ,  $\alpha = 1, 2, \dots, n$ , can be denoted as:

$$F(\mathbf{u}; z | (n)) = Prob\{Z(\mathbf{u}) \leq z | (n)\}$$

A random function (RF) is a set of RVs defined over some field of interest. A RF  $Z(\mathbf{u})$  is characterized by a set of all its  $K$ -variate ccdfs and its multivariate ccdf is defined as:

$$F(\mathbf{u}_1, \dots, \mathbf{u}_K; z_1, \dots, z_K) = Prob\{Z(\mathbf{u}_1) \leq z_1, \dots, Z(\mathbf{u}_K) \leq z_K\}$$

From the ccdf one can derive different optimal estimates for any unsampled value  $z(\mathbf{u})$  in addition to the ccdf mean, which is the least-squares error estimate (Deutsch, 1998). Also, the univariate ccdf of a RV is used to model



uncertainty about the value  $z(\mathbf{u})$  while the multivariate ccdf is used to model joint uncertainty about  $K$  values  $z(\mathbf{u}_1), \dots, z(\mathbf{u}_k)$ . Therefore, it is possible to derive various probability intervals that can be used as uncertainty metrics. These derivation processes will be addressed in the next sections.

### 3 The ccdf determination

The ccdf of a numerical RV, or of a numerical RF, can be obtained *parametrically* or *non-parametrically*. In the parametrical approach, the ccdf is determined by a limited set of statistical parameters. For example, the Gaussian ccdf is fully determined by two parameters, the mean and the variance of the distribution. Unfortunately it is a hard work to find out whether the distribution of a continuous attribute can be modeled by parametric ccdf or not. Non-parametrical distributions are more common for spatial attributes and can be estimated using the indicator kriging approach that will be explained in the next section.

### 4 The ccdf approximation using the indicator kriging approach

Instead of the variable  $Z(\mathbf{u})$ , consider its binary indicator transformation  $I(\mathbf{u}; z_k)$  defined as:

$$I(\mathbf{u}; z_k) = \begin{cases} 1, & \text{for } Z(\mathbf{u}) \leq z_k \\ 0, & \text{for } Z(\mathbf{u}) > z_k \end{cases}$$

The expectation  $E\{I(\mathbf{u}; z_k) | (n)\}$  yields an estimation  $F^*$  for the ccdf of  $Z(\mathbf{u})$  at the cutoff value  $z_k$  and conditioned to the  $n$  sample data, i. e.:

$$\begin{aligned} E\{I(\mathbf{u}; z_k) | (n)\} &= \\ 1 \cdot \text{Prob}\{I(\mathbf{u}; z_k) = 1 | (n)\} &+ 0 \cdot \text{Prob}\{I(\mathbf{u}; z_k) = 0 | (n)\} = \\ 1 \cdot \text{Prob}\{I(\mathbf{u}; z_k) = 1 | (n)\} &= F^*(\mathbf{u}; z_k | (n)) \end{aligned}$$

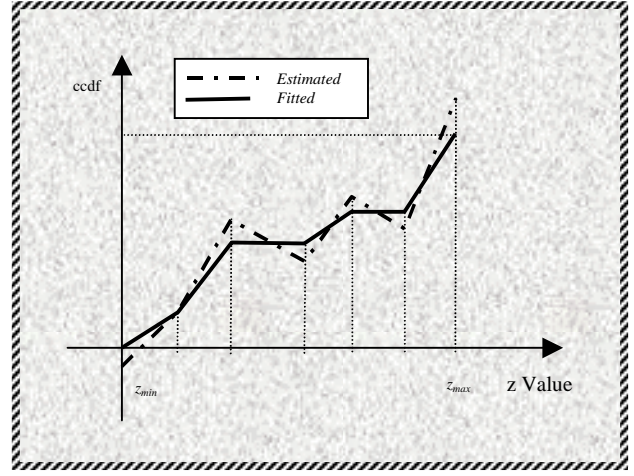
Using a linear kriging approach, as simple or ordinary kriging, to evaluate the expectation  $E$  defined in the above equation, the indicator kriging of a continuous variable aims to provide a least-squares estimate of the ccdf at cutoff  $z_k$ . A set of ccdf estimates in various cutoffs can lead to an approximation of the full ccdf of  $Z(\mathbf{u})$ . Some corrections for the follow order relation deviations:

$$0 \leq F^*(\mathbf{u}; z_k | (n)) \leq 1 \quad \forall z_k, k = 1, \dots, K$$

and

$$F^*(\mathbf{u}; z_j | (n)) \leq F^*(\mathbf{u}; z_k | (n)) \text{ se } z_j \leq z_k$$

must be performed to guarantee that the ccdf estimations are between 0 and 1 and increase monotonically. Figure 1 illustrates the fitting process of the ccdf estimation using 5 cutoff values.



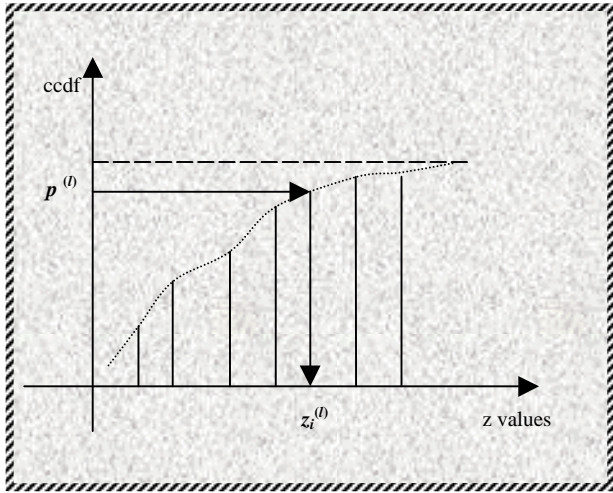
**Figure 1** The ccdf estimation using indicator kriging approach with order relation corrections

### 5 The indicator simulation approach

Stochastic simulation, hereafter called simulation for simplicity, is the process of drawing alternative, equally probable, joint realizations of the component RVs from an RF model (Deutsch, 1998). Each realization of  $Z(\mathbf{u})$  is denoted by  $z^{(l)}(\mathbf{u})$ ,  $\mathbf{u} \in A$ . A conditional simulation is the simulation conditioned to a set of  $n$  sample data. In this case the resulting realizations honor the sample data values at their location, i. e.,  $z^{(l)}(\mathbf{u}_\alpha) = z(\mathbf{u}_\alpha)$ ,  $\forall l$ .

Deutsch, 1998, presents a sequential indicator simulation approach that uses local ccdf approximation, determined by the indicator kriging approach, in order to obtain realizations of RVs  $Z(\mathbf{u})$ ,  $\mathbf{u} \in A$ . For creating a raster representation, one univariate ccdf is modeled at each node of the all grid nodes visited along a random sequence. To ensure reproduction of the  $z$ -covariance model, each univariate ccdf is made conditional not only to the sample data but also to all values simulated at previously visited locations (Goovaerts, 1997).

The realizations are drawn using probability values, obtained from an uniform random model, that are mapped to  $z$  values taking into account the estimate univariate ccdf for each node location (Felgueiras, 1999). Figure 2 illustrates this process.



**Figure 2** Process of obtaining a realization from an estimated univariate cdf

## 6 Evaluation of statistical parameters from the realizations

The set of realizations at a node location  $\mathbf{u}$  can be used to determine a cdf, along with its parameters, of a RV  $Z(\mathbf{u})$ .

The most popular predictive cdf parameter is the mean value  $\mu$ . From a set of realizations the mean value of a cdf is evaluated as the average of all the realizations. The variance and the standard deviation,  $\sigma$ , is easily evaluated using the realization values and the mean value.

The median value,  $q_{.5}$ , can be determined splitting the set of realization into 2 subsets, each with equal number of elements. Also, the set of realizations can be split in more equal subsets to derive different quantile values. When the median and the mean values are closer the distribution can be considered symmetric. The median is a more robust estimator for non-symmetrical distributions (Isaaks, 1989).

## 7 Uncertainty assessment for local estimates

As already emphasized, in section 2, the univariate cdf of a RV is used to model uncertainty about the value  $z(\mathbf{u})$  while the multivariate cdf is used to model joint uncertainty about  $K$  values  $z(\mathbf{u}_1), \dots, z(\mathbf{u}_k)$ . Therefore, given a cdf model it is possible to derive various probability intervals that can be used as uncertainty metrics.

For numerical attributes usually the uncertainties are expressed as confidence intervals. When the cdf of a RV  $Z(\mathbf{u})$  presents a high degree of symmetry and the normality of the distribution can be assumed, the estimated value  $z^*(\mathbf{u})$ , typically the mean value  $\mu$ , and the standard deviation  $\sigma$  are combined to derive a Gaussian-type confidence intervals, centered on  $z^*(\mathbf{u})$ , such as:

$$Prob\{Z(\mathbf{u}) \in [\bar{z}_z(\mathbf{u}) \pm \sigma(\mathbf{u})]\} \cong 0.68$$

or

$$Prob\{Z(\mathbf{u}) \in [\bar{z}_z(\mathbf{u}) \pm 2\sigma(\mathbf{u})]\} \cong 0.95$$

where  $\sigma^2(\mathbf{u}) = E\{(Z(\mathbf{u}) - E\{Z(\mathbf{u})\})^2\}$ .

For non-symmetrical distributions one can derive probability intervals based on quantiles of the cdf. For example, the 95% interval  $[q_{0.025}; q_{0.975}]$  is taken as:

$$Prob\{Z(\mathbf{u}) \in [q_{0.025}; q_{0.975}] / (n)\} = 0.95$$

with  $q_{0.025}$  and  $q_{0.975}$  being the 0.025 and 0.975 quantiles of the cdf, i. e.,  $F^*(\mathbf{u}; q_{0.025}|(n)) = 0.025$  and  $F^*(\mathbf{u}; q_{0.975}|(n)) = 0.975$

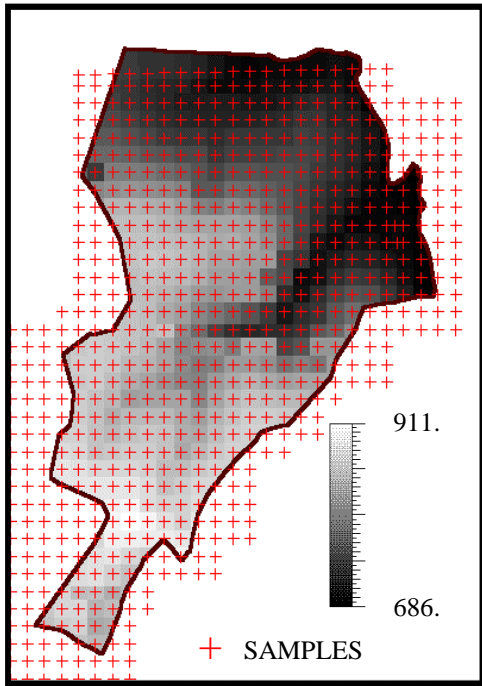
## 8 A case study for elevation data

In order to illustrate the concepts presented above, this case study uses a set of elevation data sampled in the region of an experimental farm called Canchim. The study region is located in the city of São Carlos, SP, Brazil, and cover an area of 2660 ha between the north-south coordinates from s 21°55'00" to s 21°59'00" and the east-west coordinates from w 47°48'00" to w 41°52'00".

The data set consists of 662 elevation samples distributed in the Canchim region as illustrated in the Figure 3. Some statistic values of this sample set is shown in the Table 1.

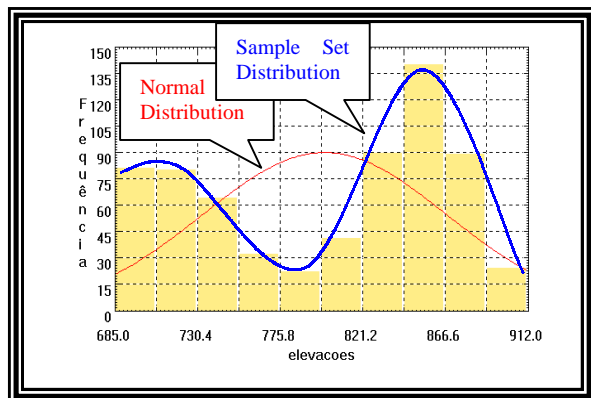
<i>Statistic</i>	<i>Value</i>
Number of Samples	662
Mean Value	800.596
Variance	4481.662
Standard Deviation	66.945
Coefficient of Variation	0.084
Coefficient of Skewness	-0.296
Coefficient of kurtosis	1.562
Minimum Value	687.000
Lower Quartile	732.500
Median	827.000
Upper Quartile	859.500
Maximum Value	911.000

**Table 1** Univariate statistics of the elevation sample set of the Canchim region



**Figure 3** Distribution of the elevation data set observed in the Canchim region.

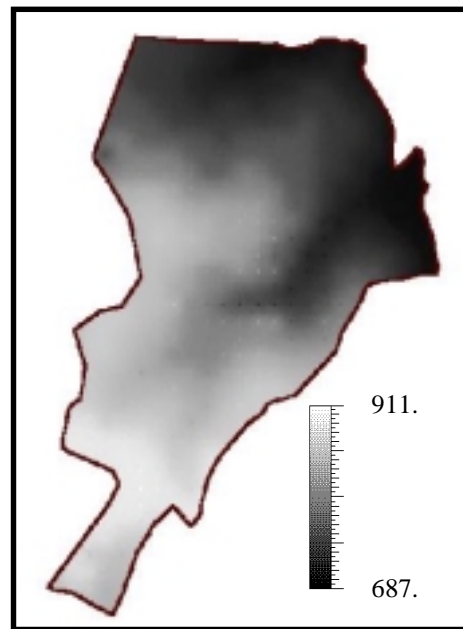
The histogram graph, presented in the Figure 4, shows the distribution of the elevation sample set compared with a normal curve distribution. It can be seen that the sample data distribution approximates a bimodal behavior and differs considerably from the Gaussian (normal) or symmetrical distribution.



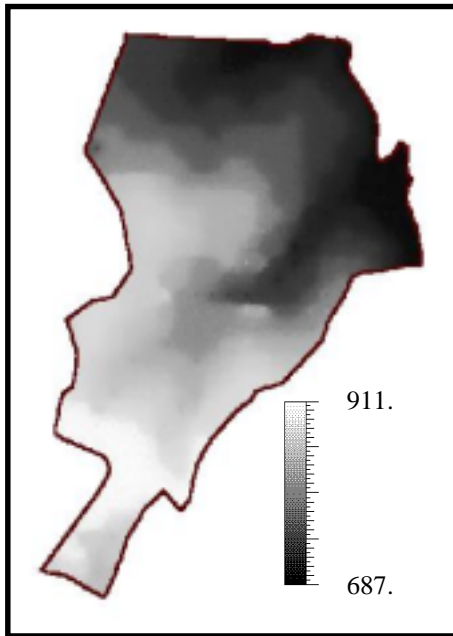
**Figure 4** Histogram of the elevation sample set emphasizing the non-normal and non-symmetrical behavior of the distribution.

The original sample set was split in 10 equal subsets (deciles) using 9 cutoff elevation values. Each cutoff value was considered in order to create indicator subsets using the indicator transformation explained in section 4. The variability of the indicator subsets are analyzed allowing the definition of an experimental and a theoretical variogram model for each subset. These tasks were performed using the geostatistical module of the SPRING GIS version 3.3 (SPRING (DPI/INPE), 1999).

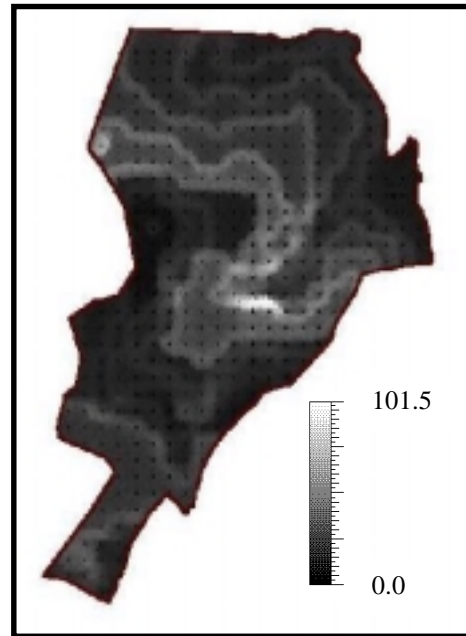
The variogram models, along with the original sample set, were used to set the parameter values of the gslib (Deutsch, 1998) sequential simulation program named sisim. This program was modified and used for estimating 400 realizations of 200 rows by 200 columns elevation grids (rectangular regular grids). Considering the 400 elevation realizations at any grid location  $u$  it was possible to render the mean and the median value maps using the methodologies defined in section 6. These maps are shown in the Figures 5 and 6. A qualitative (visual) comparative analysis of the two maps shows that they differs. This is explained by the non-symmetrical distribution of the elevation distribution model. Because of these, the median map can be considered more representative as central measure for this attribute in the region considered.



**Figure 5** Elevation grid map of local mean values estimated from the 400 grid realizations

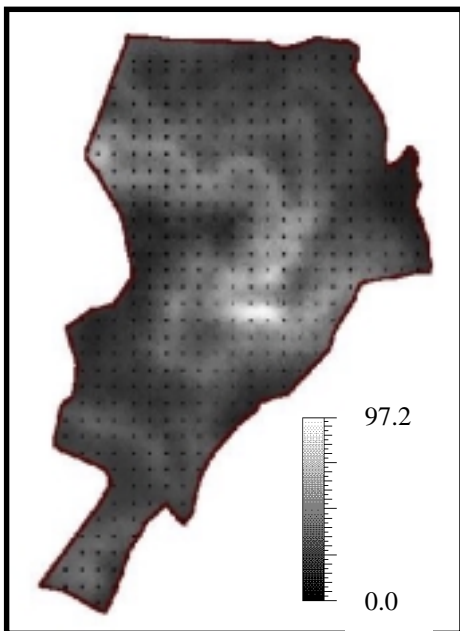


**Figure 6** Elevation grid map of local median values estimated from the 400 grid realizations

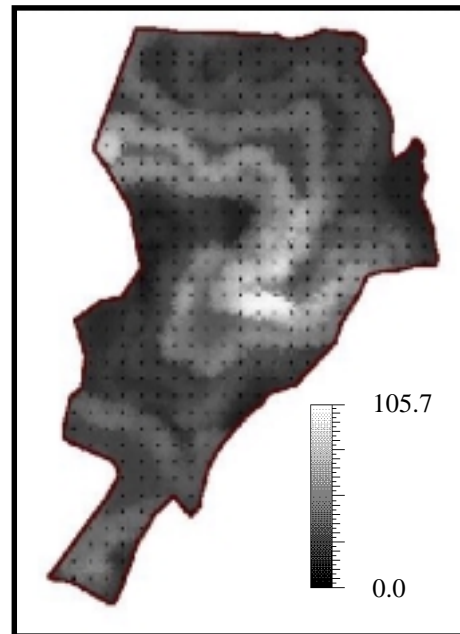


**Figure 8** Map of local uncertainties based in the interquartile confidence intervals ( $Prob\{Z(\mathbf{u}) \in [q_{0.25}; q_{0.75}]\} = 0.5$ )

The Figures 7, 8 and 9 show uncertainty maps rendered also using the 400 elevation realizations and the confidence interval methodologies explained in section 7.



**Figure 7** Map of local uncertainties based on Gaussian-type confidence intervals ( $Prob\{Z(\mathbf{u}) \in (\mu \pm \sigma)\} \approx 0.68$ )



**Figure 9** Map of local uncertainties based on interdecile confidence intervals ( $Prob\{Z(\mathbf{u}) \in [q_{0.10}; q_{0.90}]\} = 0.8$ )

It can be seen that all the above uncertainty map values are related to the attribute behavior. These uncertainty maps present maximum uncertainty values on regions (whiter regions) where the attribute values behave more erratically. Minimum uncertainty values (blackier regions) appear where the attribute values vary smoothly.

The map of Figure 7 shows uncertainty values based on Gaussian-type confidence intervals. This map was generated using one standard deviation centered in the mean value ( $Prob\{Z(\mathbf{u}) \in (\boldsymbol{\mu} \pm \boldsymbol{\sigma})\} \approx 0.68$ ). It is common to use this map as the uncertainty map related to the map estimated by mean values (Figure 5). A care has to be taken on using this type of uncertainty representation. It must be used only when the attribute variation can be modeled as RV with symmetric-distributions (normal one, for example).

The maps of Figures 8 and 9 represent uncertainties as confidence intervals based on quantiles. The quantile values are estimated directly from the 400 realizations as explained in the section 7.

The map of Figure 8 was obtained using interquartile confidence intervals ( $Prob\{Z(\mathbf{u}) \in [q_{0.25}; q_{0.75}]\} = 0.5$ ) while the map of Figure 9 was generated with interdecile confidence intervals ( $Prob\{Z(\mathbf{u}) \in [q_{0.10}; q_{0.90}]\} = 0.8$ ). As expected the map of Figure 9 contains larger uncertainty values than the one of Figure 8. The decision about which one to use depends on the accuracy demanded by an application. Finally the interquantile uncertainty maps are more appropriated to be used when the RV distributions can not be proven to have symmetrical behavior.

## 9 Conclusions

The concepts and results presented in this work show that the indicator simulation methodology is an interesting option to be considered when estimates with uncertainty assessments for numerical spatial attributes are required. The use of indicator simulation approach presents the following advantages:

- the indicator approach is non-parametric, so, it can be used independently of the attribute distribution model;
- the indicator approach allows assessment of uncertainties related to the attribute variability using an approximation of attribute distribution model;
- the sequential indicator algorithm determine the univariate cdfs taking into account the sample data set and all values simulated at previously locations. This ensure reproduction of the z-covariance model, representing better the attribute variability;
- the various equally probable outcome realizations of the indicator simulation can be used as input for complex spatial modeling (with multilayer analysis) performed by Monte Carlo simulation method. Also,

the outcomes of the spatial analysis results can be used to define their cdf's and, therefore, modeling their uncertainties.

Finally, it can be emphasized that the indicator simulation methodology can be applied, also, to thematic spatial attributes with minor modifications. This has been the subject of researchs that will be reported in the near future.

## References

- [1] P. A. Burrough and R. A. McDonnell, *Principles of Geographical Information Systems*, Oxford University Press, 1998
- [2] E. C. G. Camargo, *Desenvolvimento, implementação e teste de procedimentos geoestatísticos (krigeagem) no Sistema de Processamento de Informações Georeferenciadas (SPRING)*. Dissertação (Mestrado em Sensoriamento Remoto) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1997.
- [3] D. J. Cowen, “GIS versus CAD versus DBMS: what are the differences?”, *Photogrammetric Engineering and Remote Sensing*, 54, (1988), 1551-1554.
- [4] J. L. De Oliveira, F. Pires and C. B. Medeiros, “An environment for modeling and design of geographic applications”, *GeoInformatica*, 1, (1997), 29-58.
- [5] C. V. Deutsch and A. G. Journel, *GSLIB Geostatistical Software Library and User's Guide*. Oxford University Press, 1998.
- [6] C. A. Felgueiras, *Modelagem Ambiental com Tratamento de Incertezas em Sistemas de Informação Geográfica: O Paradigma Geoestatístico por Indicação*. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, Publicado em <http://www.dpi.inpe.br/teses/carlos/>, 1999.
- [7] C. A. Felgueiras, A. M. V. Monteiro, S. D. Fuks and E. C. G. Camargo, “Inferências e Estimativas de Incertezas Utilizando Técnicas de Krigeagem Não Linear” [CD-ROM]. In: *V Congresso e Feira para Usuários de Geoprocessamento da América Latina*, 7, Salvador, 1999. Anais. Bahia, gisbrasil'99. Seção de Palestras Técnico-Científicas.
- [8] G. B. M. Heuvelink, *Error Propagation in Environmental Modeling with GIS*, Bristol, Taylor and Francis Inc, 1998.
- [9] E. H. Isaaks and R. M. Srivastava, *An Introduction to Applied Geostatistics*, Oxford University Press, 1989.
- [10] SPRING (DPI/INPE) Sistema de Processamento de Informações Georeferenciadas – Divisão de Processamento de Imagens (DPI) do Instituto Nacional de Pesquisas Espaciais (INPE).. <http://www.dpi.inpe.br/spring/>, 1999.

# Extração de Dados em Sistemas de Informações Ambientais: Arquitetura e Esquema de Metadados

Hélio Alvaro de Mello Perez<sup>1</sup>  
Ana Maria de Carvalho Moura<sup>1</sup>  
Astério Kiyoshi Tanaka<sup>2</sup>

<sup>1</sup>Instituto Militar de Engenharia  
Depto de Engenharia de Sistemas  
e-mail : [helio.perez@usa.net](mailto:helio.perez@usa.net)  
[anamoura@ime.eb.br](mailto:anamoura@ime.eb.br)

<sup>2</sup>Universidade do Rio de Janeiro  
Depto de Informática Aplicada  
e-mail : [tanaka@uniriotec.br](mailto:tanaka@uniriotec.br)

**Abstract.** Environmental information is very important for the creation of environmental protection strategies, representing one of the greatest worries of most public organizations and societies in the world. Characterized by its heterogeneous and distributed nature, this kind of information needs to be integrated and changed into useful information, originating the so-called Environmental Information Systems (EIS). One of the greatest difficulties in constructing an EIS concerns, besides the huge amount of available data, the lack of a storage standard format, turning out its use and integration to be very difficult. Besides, the most specific and well-known metadata standards used for cataloguing environmental data still present a low power level for representing the structural part of data repositories. This paper presents a three-layer architecture to support the access and extraction processes of environmental data captured from heterogeneous and distributed repositories, and a metamodel based on metadata to enable the structure mapping of these data into an intermediate level, allowing users to build their programs to access the information directly, without changing the original storage location nor the data structure or data formats.

## 1. Introdução

Informações ambientais são imprescindíveis para a criação de estratégias de proteção ambiental. O problema é que, apesar da grande quantidade de dados disponíveis, não existe um formato padrão (planilhas, arquivos, banco de dados etc.) de armazenamento, dificultando sua integração e utilização pelas pessoas envolvidas no processo de tomada de decisão. Além disso, dados ambientais possuem características que dificultam o gerenciamento dos dados, tais como: grande volume de dados a ser processado; heterogeneidade de tipos de dados e de ambientes (software e hardware); estrutura complexa, com características espaço-temporal, incertos e altamente distribuídos (cada informação ou conjunto de dados pode estar localizada em locais diferentes, dificultando ainda mais sua recuperação e integração). Somam-se a isto ainda os principais problemas enfrentados pelos usuários na busca de informações ambientais [8]: os dados não existem ou são insuficientes; os dados não são referenciados pelos seus fornecedores ou são referenciados sob um determinado critério de classificação específico de um domínio; os dados são difíceis de serem

acessados (precisam ser integrados e transformados); os conjuntos de dados acessados são difíceis de serem utilizados pois são inconsistentes ou incompatíveis; a qualidade dos dados recuperados é difícil de ser medida já que é difícil se comparar dados produzidos por diferentes modelos científicos sem sua documentação.

Apesar do grande número de soluções sendo utilizadas e em desenvolvimento, ainda não existe uma estrutura conceitual de integração destas soluções. As abordagens existentes são superficiais em relação aos tipos de repositórios de dados utilizados por um Sistema de Informação Ambiental (SIA), sendo necessárias transformações entre formatos, aumentando a complexidade das soluções.

Dentre as abordagens existentes, a mais viável são os padrões de metadados. Existem padrões específicos para a catalogação de dados ambientais ou geográficos, a exemplo do FGDC [2] e UDK [9]. Porém, durante o desenvolvimento desse trabalho foi verificado que, apesar desses padrões serem muito utilizados para a definição de informações ambientais, ainda possuem um baixo poder de representação da parte estrutural dos repositórios de dados. Como exemplo pode-se citar a quinta seção do padrão de metadados FGDC (seção 3.2 de [6]), que só

<sup>1</sup> Hélio Perez é consultor na área de administração de dados e de banco de dados. Atualmente presta consultoria no portal financeiro InvestShop.com <http://www.investshop.com.br>

<sup>2</sup> Professor do Departamento de Informática Aplicada da UNIRIO (Universidade do Rio de Janeiro) e Pesquisador Colaborador do Departamento de Engenharia de Sistemas do IME. Parcialmente patrocinado pelos projetos CNPq processo n.350652/94-5 e PROTEM-CC/INRIA processo n. 68.0139/98.2

contempla tabelas relacionais. Esta seção é responsável pela descrição do conteúdo da informação do conjunto de dados a nível de entidades, atributos e domínios, o que é ainda pouco abrangente em relação aos diversos tipos de repositórios existentes nos SIAs.

Além disso, a complexidade de um SIA é tão grande que a tendência é não permitir a criação de novos mecanismos para a conversão e tratamento de cada repositório de dados para um novo formato, o que geraria alguns problemas, tais como: gasto de espaço físico de armazenamento desnecessário e até mesmo inviável, devido ao volume de informações; direito de acesso e uso da informação. O processo de geração de informações ambientais é caro, levando as instituições a restringirem seu acesso e/ou uso através de citações, pagamentos ou outras formas de segurança.

Todos esses aspectos, somados a falta do poder de representação da estrutura dos dados nos padrões de metadados para SIAs, geraram a necessidade da criação de uma estrutura adequada para o armazenamento da informação estrutural dos dados ambientais para cada repositório de dados, de forma clara, mantendo-os armazenados em seus locais de origem. Para atingir tal objetivo foi criado um meta-modelo capaz de mapear a estrutura dos dados ambientais em um nível intermediário, permitindo aos usuários desenvolverem programas que acessem a informação de forma direta, sem a necessidade da troca do local de armazenamento ou de mudanças na sua estrutura de armazenamento e padrão dos dados. Esse meta-modelo faz parte de um dos componentes da arquitetura proposta para a extração de dados em SIAs, também apresentada neste artigo. De forma a facilitar a criação e manutenção de instâncias da meta-modelagem desenvolvida foi criada uma ferramenta para ambiente Internet com o objetivo de facilitar o acesso a usuários em qualquer parte do mundo. Este requisito é de essencial importância num SIA, dada a sua característica natural de distribuição das informações [6].

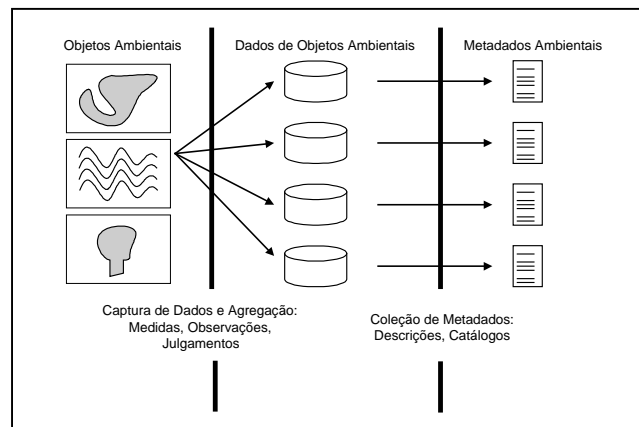
O restante desse artigo está organizado da seguinte forma: a seção 2 apresenta uma visão geral sobre SIAs e suas principais etapas de desenvolvimento; a seção 3 define os componentes básicos de uma arquitetura para a extração de dados em SIAs, onde o papel dos metadados ocupa um lugar de destaque; a seção 4 apresenta o meta-modelo de metadados criado, tecendo alguns comentários sobre a ferramenta de apoio desenvolvida para a gerência desses metadados; finalmente, a seção 5 expõe as conclusões gerais do trabalho, com suas contribuições e sugestões para trabalhos futuros.

## 2. Características dos SIAs

SIA é um sistema de gerência de informações ambientais, que manipula informações tais como solo, água, ar, clima, espécies existentes no mundo etc. [3]. A figura 2.1 apresenta três componentes importantes que

participam de um sistema de informação ambiental, visando um melhor entendimento de sua estrutura de processo:

- **Objetos Ambientais:** representam quaisquer objetos do mundo real de interesse, tais como entidades naturais (rios, lagos, oceanos, florestas, animais, etc.) e estruturas criadas pelo ser humano (casas, prédios, pontes, fábricas, etc.);
- **Dados de Objetos Ambientais:** são coleções de dados que agrupam objetos ambientais. Tal coleção é uma entidade abstrata que pode ser manipulada por computadores ou por tomadores de decisão;
- **Metadados Ambientais:** representam os metadados ambientais utilizados para referenciar um determinado dado de objeto ambiental. Cada dado de objeto ambiental possui um ou mais objetos de metadados que determinam seu conteúdo e formato.



**Figura 2.1** Modelo de Objetos de Três Modos de Sistemas de Informação Ambiental

Günther [3] divide, de acordo com o fluxo da informação ambiental, a estrutura de um SIA em quatro fases: *Captura de Dados*, *Armazenamento de Dados*, *Análise de Dados* e *Gerenciamento de Metadados*.

A fase de *Captura de Dados* tem por objetivo a coleta, processamento e agregação dos dados ambientais em sua forma bruta (séries de medida de tempo, fotografias aéreas etc.), armazenando-os em arquivos ou em bancos de dados.

O processo da captura de dados se dá através de três etapas: processamento dos dados brutos, classificação e validação, e interpretação dos dados.

Na fase de *Armazenamento de Dados* é necessário se ter um bom projeto de banco de dados e estruturas de armazenamento apropriadas, levando a um aumento da performance do sistema. Devido a complexidade e heterogeneidade dos dados ambientais é aconselhável dispor de bancos de dados que provejam extensões à tecnologia relacional.

Na fase de *Análise de Dados* a informação disponível é preparada para propósitos de apoio à tomada de decisão.

É necessário o acesso simultâneo aos dados que estão dispersos geograficamente, armazenados em diferentes *hardwares*, e organizados em uma grande variedade de modelos de dados.

Finalmente, a fase de *Gerenciamento de Metadados* é a fase de maior importância durante o processo de criação de um SIA. Metadados são coletados e agregados nas três fases descritas anteriormente. São armazenados em estruturas de dados apropriadas, fornecendo o apoio necessário às operações de pesquisa, navegação e transferência de dados durante a fase de Análise de Dados. A criação dos metadados deve ser feita, caso seja possível, em paralelo à coleta dos dados originais, o que nem sempre ocorre.

Geralmente a criação de metadados é manual, dependendo de alguém que conheça como obter e interpretar a informação ali contida. Existem duas soluções viáveis para o problema: a primeira seria a criação de uma arquitetura organizacional que funcionasse como guia para os fornecedores dos dados durante a captura dos metadados. A segunda seria a criação de um processo de extração automática dos metadados. A primeira opção é mais voltada para as empresas privadas ou públicas que criam seus processos de extração de metadados de forma padronizada, garantindo sua qualidade. A segunda opção tem conquistado maior espaço dentre as alternativas, apesar de ainda ser difícil de ser alcançada.

Uma das soluções que facilita o processo de geração de metadados é a criação de padrões de metadados, que fornecem uma maneira sistemática de coletar os metadados. Dentre os mais importantes padrões de metadados ambientais estão o padrão americano FGDC e o europeu UDK. Ambos documentam as características dos dados ambientais, conteúdo, qualidade e condição, da mesma forma que os rótulos auxiliam o consumidor, permitindo aos usuários de tais dados determinarem se eles servem ou não para seus propósitos. Em [6] é feito um estudo detalhado desses padrões, a partir do qual foi possível concluir que: o padrão FGDC possui um maior apoio (softwares e documentação) para seu desenvolvimento e utilização pelas entidades públicas ou particulares; já o padrão UDK peca pela falta de documentação existente no idioma inglês, o que muito dificulta sua aceitação. Conforme já mencionado, a principal deficiência do padrão FGDC é não permitir representar a estrutura das fontes de dados pela seção 5, visto que essa especificação é feita em texto livre, e portanto não estruturada. A idéia seria estender essa seção, estruturando-a, de forma a abranger diversos tipos de fontes de dados.

Metadados representam portanto um módulo de extrema relevância numa arquitetura de extração de dados em ambientes heterogêneos. A seguir é apresentada uma visão

geral da arquitetura proposta para a extração de dados em SIAs.

### 3. Arquitetura de Extração de Dados em SIAs

A construção de um SIA representa uma tarefa de extrema complexidade, devido a heterogeneidade da estrutura de armazenamento (planilhas, arquivos textos, banco de dados etc.) e da grande quantidade de dados disponíveis e distribuídos em diferentes locais. Tais sistemas devem permitir que os usuários consultem as informações de modo eficiente e preciso. Para isso, é necessária a união de diversas técnicas de diferentes áreas de pesquisas, tais como: metadados, ontologias, ferramentas de catalogação, classificação e pesquisa na Web, agentes etc. A integração dessas técnicas leva à definição de uma arquitetura para a extração de dados em SIAs.

A heterogeneidade dos dados ambientais e seu armazenamento em múltiplos repositórios distribuídos requer que a construção de um SIA tenha como base uma arquitetura distribuída, permitindo que seus componentes sejam integrados de forma incremental. A arquitetura proposta para a extração de dados em SIAs leva em conta essa importante característica, a partir de componentes bem determinados, permitindo que o sistema ambiental seja desenvolvido em várias camadas independentes. Além disso, pode-se utilizar uma abordagem *bottom-up*, onde o sistema é desenvolvido a partir da inclusão de segmentos independentes (fontes de dados), podendo ser integrados posteriormente, de forma distribuída, um após o outro.

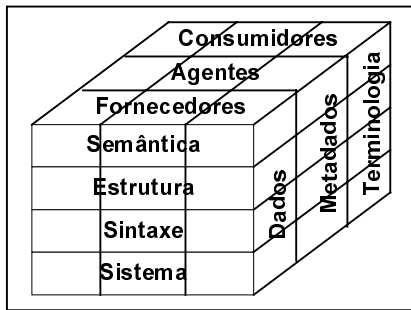
Tal arquitetura deve lidar com a informação em diversos níveis de heterogeneidade, ou seja, deve trabalhar com os quatro níveis de interoperabilidade existentes [7], onde cada um é responsável por:

- **Sistema:** diferenças de hardware e de sistemas operacionais;
- **Sintaxe:** diferenças na representação dos dados, formato e armazenamento;
- **Estrutura:** diferenças de modelos de dados e estruturas;
- **Semântica:** diferenças no significado são dependentes do vocabulário e da terminologia utilizados para expressar a informação e o conteúdo na qual está interpretada.

A figura 3.1 apresenta os tipos de usuários que interagem na arquitetura, o grau de interoperabilidade e os níveis de informação [7]. Consumidores são os usuários finais que buscam a informação; fornecedores são os responsáveis pela geração dos dados (informação); e os agentes, também conhecidos como *brokers*, representam os intermediários entre os consumidores e os fornecedores, isto é, são os responsáveis pela



disponibilização da informação gerada de forma organizada pelos fornecedores, permitindo o seu acesso pelos consumidores.



**Figura 3.1** Tipos de usuários, grau de interoperabilidade e níveis de informação

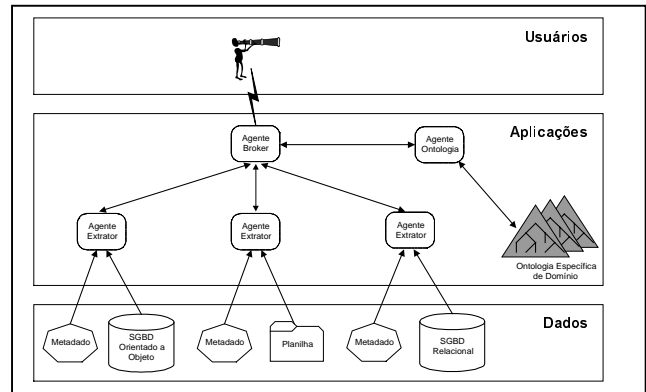
Em resumo, a figura 3.1 pode ser interpretada da seguinte forma: consumidores utilizam a terminologia (ontologia) para recuperar a informação; agentes utilizam os metadados para criar processos de extração dos dados; e os fornecedores geram os dados, fornecendo-os aos consumidores.

A partir dos aspectos apresentados anteriormente é possível apresentar a arquitetura proposta para um SIA, que é composta de seis componentes, a saber: Usuário, Agente Broker, Agente Ontologia / Ontologia Específica de Domínio, Agente Extrator, Metadado e Fonte de Dados.

A figura 3.2 apresenta uma abordagem de alto nível da arquitetura proposta para SIAs dividida em 3 camadas, de forma a permitir uma melhor representação dos componentes participantes envolvidos no processo.

A seguir, é apresentada uma visão geral de cada uma das camadas:

- A primeira camada (**Usuários**) representa os consumidores de informação, ou seja, os cientistas, governantes, entidades privadas etc.;
- A segunda camada (**Aplicações**) representa os "brokers" ou seja, as entidades responsáveis pela integração da informação e os agentes extratores, isto é, aqueles que se comunicam diretamente com cada repositório de dados para recuperarem a informação. Cada repositório possui um ou mais extratores ou tradutores (também denominado "wrappers") responsáveis pela recuperação da informação solicitada. Essa segunda camada é muito importante, visto que efetua a ligação entre os consumidores de informação (Usuários) e os produtores de dados (Dados). Nela se encontram as ontologias específicas de domínio, equivalentes a um dicionário de termos, utilizadas para resolução de conflitos semânticos [7];
- A terceira camada (**Dados**) representa os repositórios de informações geradas pelos produtores de dados.



**Figura 3.2** Arquitetura proposta para SIAs

A arquitetura descrita é por natureza distribuída, permitindo que se utilize a Internet como meio de transmissão de informações, levando, juntamente com a tecnologia de orientação a objetos, a um alto grau de escalabilidade e abrangência. Uma das soluções seria o uso de uma arquitetura distribuída, como a arquitetura CORBA [5] considerada atualmente como plataforma padrão para integração e distribuição de objetos. A arquitetura CORBA é uma arquitetura de objetos distribuídos, permitindo que aplicações façam solicitações aos objetos, de forma transparente, independentemente da linguagem, sistema operacional ou considerações de localização [1].

A heterogeneidade das estruturas das fontes de dados (arquivos textos, planilhas, bancos de dados relacionais e orientados a objeto etc.) foi um ponto marcante para a identificação da necessidade de armazenamento de suas estruturas sob forma de metadados, além dos metadados normalmente encontrados nos padrões de catalogação de dados ambientais. O "Agente Extrator" precisa conhecer a estrutura da fonte de dados, ou seja como ela está organizada, de modo a realizar adequadamente o processo de extração dos dados. Esse conjunto de metadados, o qual denominamos de metadado estrutural, é um ponto importante no escopo desse trabalho, cuja modelagem é apresentada na seção a seguir.

#### 4. Meta-Modelo para Suporte à Extração de Dados

Os estudos de padrões de metadados para SIAs e a definição de uma arquitetura de extração de dados em SIA levou à conclusão de que um padrão de metadados abrangente deve contemplar também metadados que representem a estrutura de armazenamento de todas as fontes de dados envolvidas no sistema. Esta seção apresenta um dos componentes necessários à construção de um SIA através de um meta-modelo denominado **Meta-Modelo para Sistemas de Informações Ambientais (MM-SIA)** [6], permitindo que os agentes extratores

recuperem a estrutura das fontes de dados a partir de um repositório central.

O MM-SIA foi desenvolvido tendo como base as seguintes fontes de dados: Arquivos Textos, com ou sem separador, Planilhas, Páginas Web (XML), Tabelas ou Visões oriundas de SGBDs Relacionais, SGBDs Relacionais-Objeto e Classes oriundas de SGBDs Orientados a Objetos. Estas fontes foram escolhidas por incluírem os tipos de fontes de dados mais comumente encontrados para o armazenamento de dados ambientais. O modelo é plenamente extensível, não impedindo que outros tipos de fontes sejam acomodados, de acordo com as necessidades dos usuários.

A idéia dessa modelagem é permitir a criação de um repositório de informações sobre a estrutura de cada fonte de dados participante de um SIA, permitindo que um agente extrator possa utilizar esse repositório para recuperar a estrutura de cada fonte de dados.

A figura 4.1 apresenta o diagrama de classes da MM-SIA modelado em UML [10], cujas propriedades são auto-explicativas.

O meta-modelo pode ser entendido da seguinte forma: um *sistema* ambiental é composto de uma ou mais fontes de dados com características estruturais diferentes (planilhas, arquivos texto, tabelas etc.). Cada fonte de dados possui um conjunto de atributos que a define, sendo que cada um desses pode ter ou não restrições. A classe correspondente à fonte de dados *arquivo* representa arquivos textos que podem ser com ou sem separador, ou seja, um arquivo texto sem separador significa que cada coluna do arquivo possui sua posição inicial e final determinada (por exemplo, nome vai da coluna 1 a 10). Já o arquivo texto com separador possui um caracter que indica a separação entre as colunas (por exemplo, o caracter \* separando as colunas nome e cpf).

A classe de atributos *atributo-arquivo* representa o local onde o atributo está localizado dentro do arquivo texto onde: C = atributo no cabeçalho; R = atributo no rodapé e E = atributo no interior do arquivo. Isso é devido a necessidade de se mapear arquivos com controles em sua primeira e/ou última linha, diferindo da estrutura do resto do arquivo.

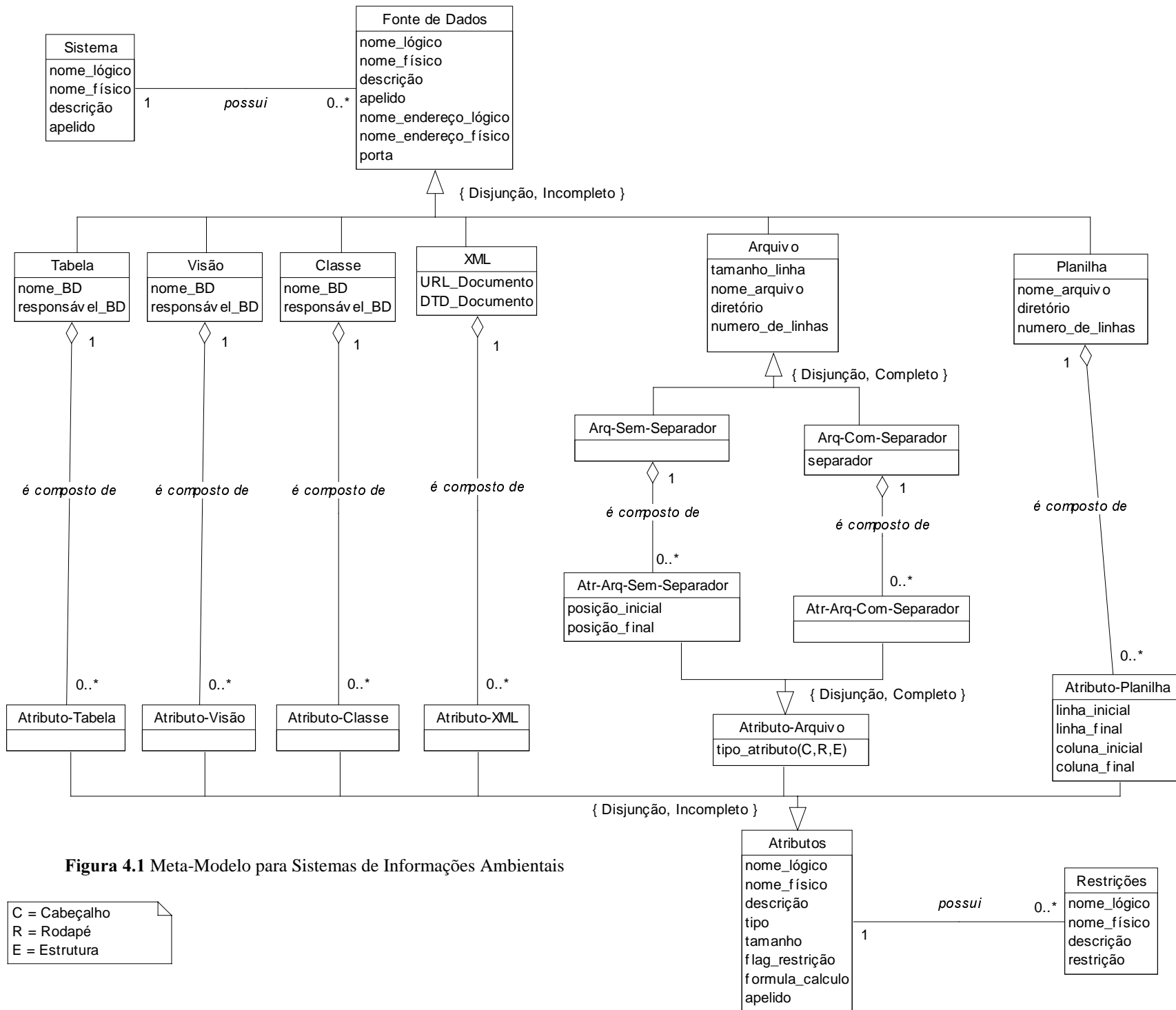
As demais fontes de dados correspondem às seguintes classes do meta-modelo:

- **tabela e visão**: estas classes permitem considerar respectivamente objetos e visões de sistemas gerenciadores de bancos de dados;
- **classe**: permite representar objetos de sistemas de bancos de dados orientados a objetos e relacionais-objeto.
- **planilha**: representa planilhas eletrônicas, a exemplo do MS Excel.
- **XML**: permite considerar páginas no formato XML.

De forma a validar o meta-modelo apresentado, foi desenvolvida uma ferramenta de gerencia de metadados para o ambiente Web, utilizando-se como estudo de caso aplicações reais desenvolvidas na Embrapa Solos - RJ [11] e no Instituto Agrônômico de Campinas - SP [12], onde grande parte dos dados encontram-se em planilhas e tabelas relacionais, tais como: solo, chuva, temperatura mínima e insolação.

Essa ferramenta foi construída com base na tecnologia do padrão Web através do uso de *applets* Java [4], possuindo uma navegação simples através de seus objetos como os estados no diagrama de transição de estados. Também permite ao usuário executar as operações básicas de gerenciamento e manipulação de cada objeto do MM-SIA.

Foram criados os metadados das informações de solo, chuva, temperatura mínima e insolação no padrão FGDC, além do mapeamento da estrutura dessas fontes de dados para o meta-modelo MM-SIA. Maiores detalhes podem ser encontrados em [6].



**Figura 4.1** Meta-Modelo para Sistemas de Informações Ambientais

C = Cabeçalho  
 R = Rodapé  
 E = Estrutura

## 5. Conclusão

Dentre as fases de desenvolvimento de um SIA a gerência de metadados é um ponto de grande relevância. Corresponde à fase de classificação e identificação da informação ambiental, onde são definidos todos os descritores dos dados, inclusive os que descrevem a sua estrutura. A definição dessa estrutura é imprescindível durante a criação de extratores para a recuperação dos dados distribuídos, já que esses precisam conhecer a estrutura de armazenamento dos dados para poder recuperá-los de forma correta. Esta estrutura não é contemplada, por exemplo, no padrão de metadados FGDC tão utilizado pelos SIAs. Este artigo apresentou uma arquitetura genérica para a extração de dados ambientais distribuídos e um meta-modelo para a definição da estrutura dessas fontes de dados, permitindo aos donos dos dados informarem a estrutura de suas fontes de dados. Com base nesse meta-modelo foi construída uma ferramenta para gerência de metadados ambientais, tendo sido aplicada a sistemas reais de meio-ambiente. A meta-modelagem desenvolvida é extensível, podendo ser estendida a outras estruturas de armazenamento diferentes das padrões.

Como continuação desse trabalho pretende-se incluir na seção 5 (Entidade e Atributos) do padrão FGDC o meta-modelo criado, utilizando-se como fonte as instâncias já armazenadas na ferramenta desenvolvida. Para isso é necessária a extensão dessa seção de modo a contemplar os novos tipos de fontes de dados.

## Referências

- [1] Cattel, R. G. G. The Object Databases Standard: ODMG 2.0, Morgan Kaufman, 1997.
- [2] Federal Geographic Data Committee (FGDC). <http://www.fgdc.gov> 1999
- [3] Gunther, O. Environment Informations Systems. Springer-Verlag Berlin Heidelberg, 1998.
- [4] JAVA, SUN Microsystems. <http://java.sun.com/>, 1999.
- [5] Object Management Group <http://www.omg.org>, 1999
- [6] Perez, H. A. M. Modelagem de Metadados para Suporte a Extração de Dados em Sistemas de Informações Ambientais. Tese de Mestrado, Instituto Militar de Engenharia, Rio de Janeiro, fev. 2000.
- [7] Sheth, A., Kashyap, V., Lima, T. Semantic Information brokering: How can a Multi-Agent Approach Help?. Proceedings of the Third International Workshop on Cooperative Information Agents, julho 1999.
- [8] Simon, E., Tomasic, A. Improving Access to Environment Data using Context Information, SIGMOD Record; 1997.
- [9] Especificação do Modelo UDK (UmwelDatemKatalog). <http://www.mu.niedersachsen.de/udk>, 1999.
- [10] Object Management Group <http://www.omg.org/uml/>, 1999
- [11] Empresa Brasileira de Pesquisa Agropecuária - Centro Nacional de Pesquisa de Solos, <http://www.cnps.embrapa.br>, 1999
- [12] Instituto Agrônômico, <http://www.iac.br>, 1999

# Extensões ao Modelo OMT-G para Produção de Esquemas Dinâmicos e de Apresentação

CLODOVEU A. DAVIS JR.<sup>1</sup>

ALBERTO H. F. LAENDER<sup>2</sup>

<sup>1</sup>PRODABEL - Empresa de Informática e Informação do Município de Belo Horizonte,

Av. Presidente Carlos Luz, 1275, 31230-000 Belo Horizonte, MG, Brasil

cdavis@uol.com.br

<sup>2</sup>DCC/UFMG - Departamento de Ciência da Computação - Universidade Federal de Minas Gerais,

Av. Presidente Antônio Carlos, 6627, 31270-901 Belo Horizonte, MG, Brasil

laender@dcc.ufmg.br

**Abstract.** A construção de um esquema conceitual é parte fundamental do processo de desenvolvimento de aplicações. No caso das aplicações geográficas, no entanto, é necessário levar em consideração outros fatores, especialmente relativos à representação dos objetos espaciais. Sem a definição da representação, torna-se difícil a especificação dos relacionamentos espaciais e topológicos entre os objetos. Reunindo elementos conceituais e de representação, aplicações geográficas podem ser modeladas estaticamente usando o modelo OMT-G. Este artigo propõe extensões ao modelo OMT-G, no sentido de permitir a especificação de transformações entre representações, por meio de um esquema dinâmico. Também são propostas extensões para a especificação de diferentes apresentações visuais para cada representação modelada, compondo um esquema de apresentações. As extensões propostas permitem que projetos de aplicações geográficas incorporem aspectos de múltiplas representações e múltiplas apresentações, viabilizando assim o compartilhamento do banco de dados geográfico entre diferentes grupos de usuários.

## 1. Introdução

A construção de um esquema conceitual é parte fundamental do processo de desenvolvimento de sistemas de informação. Durante a modelagem, é necessário identificar todos os objetos do mundo real que de alguma forma interfiram no sistema, por atuar na parcela do mundo real que se está procurando modelar. Em seguida, é preciso extrair um conjunto de características de cada objeto identificado, em um processo de *abstração*. A incorporação da geometria e da topologia dos objetos espaciais ao esquema conceitual consiste em escolher uma *representação* adequada para cada um deles, que seja capaz de incorporar suas características espaciais, como localização, topologia e forma geométrica. A existência de uma representação para um objeto espacial não determina completamente sua *aparência visual*, ou seja, a forma segundo a qual o objeto será apresentado ao usuário, na tela ou em papel. A cada representação podem corresponder uma ou mais *apresentações*, alternativas de visualização adequadas para comunicar o significado dos dados geográficos de acordo com as necessidades do usuário e da aplicação.

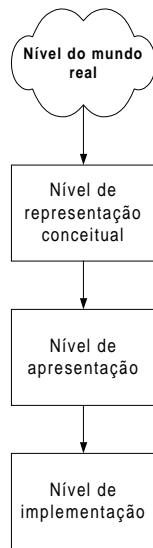
Ao considerar a necessidade de modelagem de representações e apresentações, percebe-se que os modelos de dados geográficos atualmente disponíveis não incorporam recursos para que transformações envolvendo representações e apresentações sejam concebidas. Este trabalho propõe um conjunto de extensões ao modelo OMT-G [2, 3, 4], de modo a incorporar primitivas para a construção de esquemas *dinâmicos* e de esquemas *de apresentação*, e

está organizado como se segue. A Seção 2 apresenta os níveis de abstração adotados para modelagem de dados geográficos. A Seção 3 apresenta brevemente o modelo OMT-G. A Seção 4 apresenta as extensões propostas. Finalmente, a Seção 5 apresenta as conclusões do trabalho e discute aspectos relativos à sua utilização prática.

## 2. Níveis de Abstração

Modelos de dados variam de acordo com o nível de abstração. Para aplicações geográficas, existem basicamente quatro níveis distintos de abstração (Figura 1). No *nível do mundo real* estão os fenômenos geográficos reais a serem representados. O *nível de representação conceitual* oferece um conjunto de conceitos formais com os quais as entidades geográficas podem ser modeladas da forma em que são percebidas pelo usuário, em um nível alto de abstração. Neste nível são definidas as classes básicas, contínuas ou discretas, que serão criadas no banco de dados, cada qual associada a uma classe de representação espacial, sendo que é possível que um elemento do mundo real possua várias representações. Em seguida, o *nível de apresentação* oferece ferramentas com as quais se pode especificar os diferentes aspectos visuais que as entidades geográficas têm de assumir ao longo de seu uso em aplicações. Neste nível, cada alternativa de representação definida no nível anterior é associada a uma ou mais apresentações. Finalmente, no *nível de implementação* são definidos padrões, mecanismos de armazenamento, estruturas de dados e funções de uso geral para implementar

fisicamente cada representação, conforme definida no nível de representação conceitual, e cada apresentação exigida, conforme definido no nível de apresentação.



**Figura 1 - Níveis de especificação de aplicações geográficas**

A definição de níveis de abstração apresentada difere da proposta em [6]. Existe uma certa discordância quanto à fusão, em um único nível, das definições conceituais e de representação. É nosso entendimento que a modelagem conceitual para aplicações geográficas não pode ser levada a termo sem que alguma forma de representação seja definida para os objetos espaciais, sob pena de não se conseguir conceber adequadamente os relacionamentos entre esses objetos. Grande parte da discussão ao redor deste aspecto decorre de uma confusão entre representação e apresentação, e portanto a inclusão de um nível separado para receber as especificações relativas a aspecto visual e gráfico contribui para resolver o problema. O nível de apresentação situa-se entre o nível de representação conceitual e o de implementação, uma vez que introduz detalhes de especificação parcialmente dependentes dos recursos disponíveis para a implementação, mas ao mesmo tempo define os parâmetros recomendáveis para o melhor uso da informação concebida para a aplicação.

### 3. Modelagem Usando o Modelo OMT-G

O modelo OMT-G propõe uma série de primitivas que permitem construir o esquema estático de aplicações geográficas, no qual são especificadas as classes envolvidas no problema, juntamente com suas representações básicas e seus relacionamentos. A partir do esquema estático, é possível produzir um conjunto de restrições de integridade espacial que precisam ser implementadas pela aplicação ou pelo banco de dados geográfico utilizado.

O modelo OMT-G parte das primitivas definidas para o diagrama de classes da *Universal Modeling Language* (UML) [12], introduzindo primitivas geográficas com o objetivo de aumentar a capacidade de representação semântica daquele modelo, e portanto reduzindo a distância entre o modelo mental do espaço a ser modelado e o modelo de representação usual. O modelo OMT-G é baseado em três conceitos principais: *classes*, *relacionamentos* e *restrições de integridade espaciais*. Classes e relacionamentos definem as primitivas básicas usadas para criar esquemas estáticos de aplicação. A identificação de restrições de integridade espacial é uma atividade importante no projeto de uma aplicação, e consiste na identificação de condições que se precisam ser garantidas para que o banco de dados esteja sempre íntegro. As restrições de integridade espaciais para o modelo OMT-G foram detalhadas em [3] e [4], e não serão apresentadas aqui. As primitivas de classes e relacionamentos serão apresentadas a seguir.

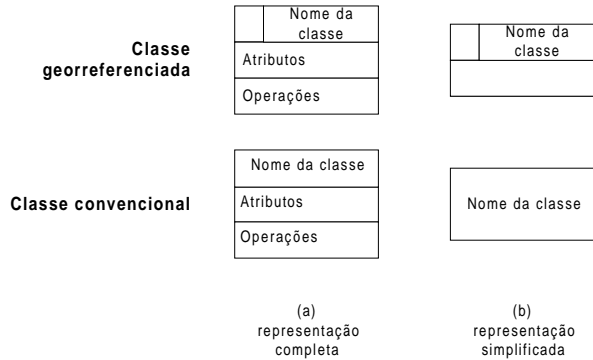
#### 3.1 Classes

As classes definidas pelo modelo OMT-G podem ser *georreferenciadas* ou *convencionais*. Uma *classe georreferenciada* descreve um conjunto de objetos que têm representação espacial e que são associados a elementos do mundo real localizáveis na Terra [5], assumindo a visão de campos e objetos proposta por Goodchild [9]. Uma *classe convencional* descreve um conjunto de objetos com propriedades, comportamento, relacionamentos e semântica semelhantes, o qual pode ter algum tipo de relacionamento com objetos espaciais, mas não possui propriedades geográficas ou geométricas.

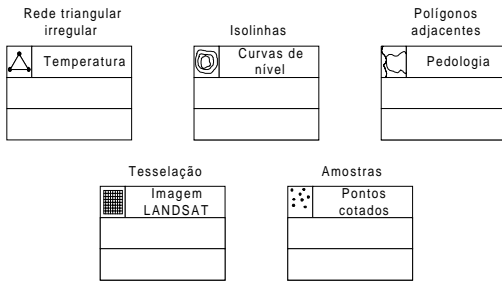
As classes georreferenciadas são especializadas em *geo-campos* e *geo-objetos*. Geo-campos representam objetos e fenômenos distribuídos continuamente no espaço, correspondendo a variáveis como tipo de solo, relevo e geologia [5]. Geo-objetos representam objetos geográficos particulares, individualizáveis, associados a elementos do mundo real, como edificações, rios e árvores. As classes convencionais são simbolizadas exatamente como na UML. As classes georreferenciadas são simbolizadas no modelo OMT-G de forma semelhante (Figura 2a), incluindo no canto superior esquerdo um retângulo que é usado para indicar a geometria da representação. Em ambos os casos, símbolos simplificados podem ser usados (Figura 2b). Os objetos podem ter ou não atributos não espaciais associados, listados na seção central da representação completa. Métodos ou operações associadas são especificadas na seção inferior.

O modelo OMT-G apresenta um conjunto fixo de alternativas de representação geométrica, usando uma simbologia que distingue geo-objetos (Figura 3) e geo-campos (Figura 4). São definidas cinco classes descendentes de

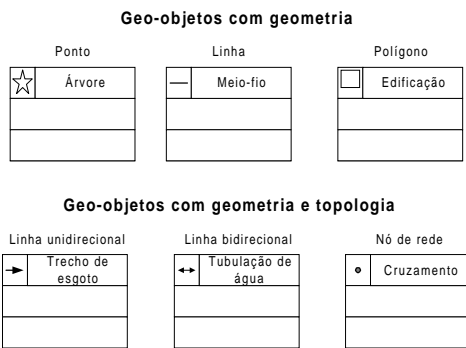
geo-campo (*isolinhas, polígonos adjacentes, tesselação, amostragem e rede triangular irregular*) e duas classes descendentes de geo-objeto, *geo-objeto com geometria e geo-objeto com geometria e topologia*, sendo que da primeira descendem as classes *ponto, linha e polígono*, e da segunda descendem as classes *nó de rede, arco unidirecional e arco bidirecional*.



**Figura 2 - Notação gráfica para as classes do modelo OMT-G**



**Figura 3 - Geo-campos**

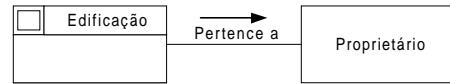


**Figura 4 - Geo-objetos**

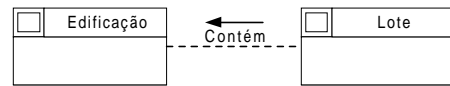
### 3.2 Relacionamentos

Considerando a importância dos relacionamentos espaciais e não espaciais para a compreensão do espaço modelado, o modelo OMT-G representa os três tipos de rela-

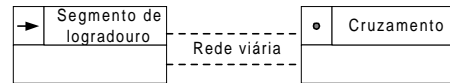
cionamento que podem ocorrer entre suas classes: *associações simples, relacionamentos topológicos em rede e relacionamentos espaciais*. *Associações simples* representam relacionamentos estruturais entre objetos de classes diferentes, convencionais ou georreferenciadas. *Relacionamentos espaciais* representam relações topológicas, métricas, ordinais e *fuzzy*. *Relacionamentos de rede* são relacionamentos entre objetos que estão conectados uns com os outros.



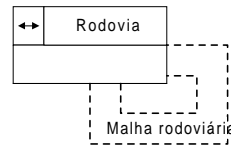
(a) Associação simples



(b) Relacionamento espacial



(c) Relacionamento de rede arco-nó



(d) Relacionamento de rede arco-arco

**Figura 5 - Relacionamentos**

No modelo OMT-G, associações simples são representadas graficamente com linhas contínuas, enquanto relacionamentos espaciais são representados com linhas pontilhadas (Figura 5a e b), o que facilita a distinção visual entre relacionamentos baseados em atributos alfanuméricos e baseados na localização e geometria dos objetos. O nome do relacionamento é anotado sobre a linha, e uma seta pode ser desenhada para deixar clara a direção de leitura (por exemplo, na Figura 5b, lê-se “lote contém edificação”). Relacionamentos de rede são indicados por duas linhas pontilhadas paralelas, entre as quais o nome do relacionamento é anotado (Figura 5c). Os relacionamentos são em geral especificados entre uma classe de nós e uma classe de arcos, mas estruturas de redes sem nós podem ser definidas, especificando um relacionamento recursivo sobre uma classe de arcos (Figura 5d). Os relacionamentos são ainda caracterizados por sua *cardinalidade*. A notação de cardinalidade adotada pelo modelo OMT-G é a mesma empregada pela UML, usando números para indicar quan-

tidades, reticências para denotar faixas e o asterisco para simbolizar “muitos”, como em 1 . . \*.

### 3.3 Generalização e especialização

*Generalização* é o processo de definição de classes que são mais genéricas (superclasses) do que classes com características semelhantes (subclasses) [8, 11]. A *especialização* é o processo inverso, no qual classes mais específicas são detalhadas a partir de uma classe mais genérica, adicionando novas propriedades na forma de atributos. Cada subclasse herda atributos, operações e associações de sua superclasse.

No modelo OMT-G, as abstrações de generalização e especialização se aplicam tanto a classes georreferenciadas quanto a classes convencionais, seguindo as definições e a notação propostas na UML, em que um triângulo conecta a superclasse a suas subclasses. Cada generalização pode ter um *discriminador* associado, que indica qual propriedade ou característica está sendo abstraída pelo relacionamento de generalização.

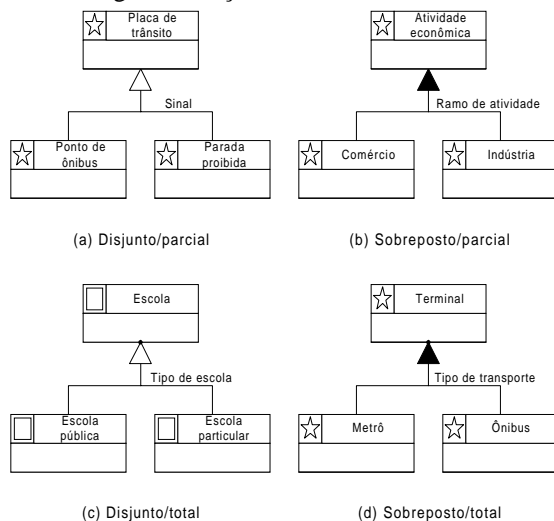


Figura 6 - Exemplos de generalização espacial

As generalizações (espaciais ou não) podem ser especificadas como *totais* ou *parciais* [11, 12]. Uma generalização é total quando a união de todas as instâncias das subclasses é equivalente ao conjunto completo de instâncias da superclasse. A UML representa a totalidade através do uso dos elementos de restrição predefinidos *completo* e *incompleto*, mas no modelo OMT-G foi adotada a notação introduzida em [11], na qual um ponto é colocado no ápice do triângulo para denotar a totalidade (Figura 6). Além disso, o modelo OMT-G também adota a notação OMT [13] para os elementos de restrição predefinidos *disjunto* e *sobreposto* da UML, ou seja, em uma generalização disjunta o triângulo é deixado em branco e em uma generalização sobreposta o triângulo é preenchido. Por-

tanto, a combinação de disjunção e totalidade gera quatro tipos de restrição aplicáveis a generalização e especialização. A Figura 6 apresenta exemplos de cada combinação.

### 3.4 Agregação

Agregação é uma forma especial de associação entre objetos, onde se considera que um deles é montado a partir de outros. A notação gráfica usada no modelo OMT-G segue a empregada na UML. Uma agregação pode ocorrer entre classes convencionais, entre classes georreferenciadas ou entre uma classe convencional e uma classe georreferenciada. Quando a agregação ocorre entre classes georreferenciadas, é necessário usar a *agregação espacial*.

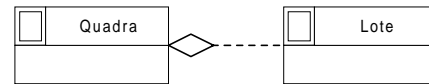


Figura 7 - Agregação espacial

A agregação espacial é um caso especial de agregação na qual relacionamentos topológicos “todo-parte” são explicitados [1, 10]. A agregação espacial indica que a geometria de cada parte deve estar totalmente contida na geometria do todo. Além disso, não é permitida a superposição entre a geometria das partes e a geometria do todo deve ser totalmente coberta pela geometria das partes. Um exemplo dessa primitiva está apresentado na Figura 7, que mostra uma situação em que quadras são compostas de lotes, ou seja, as quadras são geometricamente equivalentes à união dos lotes contidos nelas.

### 3.5 Generalização Cartográfica

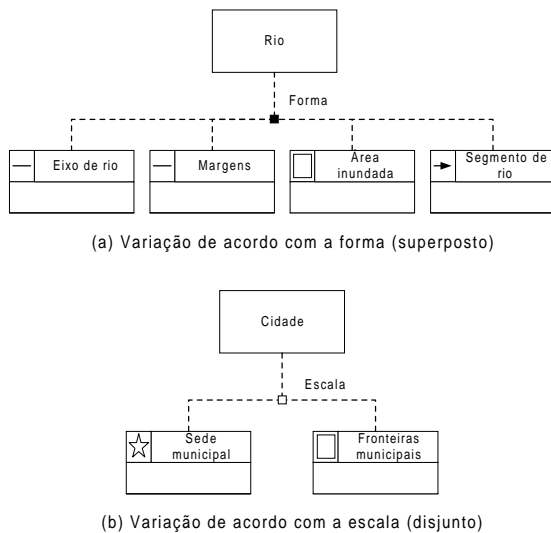
A *generalização*, no sentido cartográfico, pode ser definida como uma série de transformações que são realizadas sobre a representação da informação espacial, cujo objetivo é melhorar a legibilidade e a compreensão dos dados. Um objeto do mundo real pode ter diversas representações, de acordo com a escala de visualização: uma cidade pode ser representada como um ponto em um mapa de escala pequena, e como um polígono em um mapa de escala maior.

Definir se a representação deve ser mais simples ou mais elaborada depende da percepção que o usuário tem do objeto correspondente no mundo real, e de como essa representação afeta os relacionamentos espaciais que podem ser estabelecidos com outros objetos modelados. Considerando a necessidade de tais relacionamentos, pode haver a demanda para mais de uma representação para um dado objeto [7]. Esse é freqüentemente o caso quando a informação geográfica precisa ser compartilhada entre diversas aplicações.

A primitiva de *generalização cartográfica* foi inclu-



ída no modelo OMT-G para registrar a necessidade de representações diferentes para um mesmo objeto. Nessa primitiva, a superclasse não precisa ter uma representação específica, já que poderá ser percebida de maneiras diferentes, conforme especificado nas subclasses. Estas são representadas por formas geométricas distintas, podendo herdar os atributos alfanuméricos da superclasse e ainda possuir atributos próprios. O objetivo dessa primitiva é permitir a especificação de relacionamentos independentes envolvendo cada alternativa de representação considerada, em particular relacionamentos espaciais, que dependem da geometria da representação. É permitido que a mesma alternativa de representação ocorra em mais de uma classe, pois em cada uma o nível de detalhamento ou resolução pode variar.



**Figura 8 - Generalização cartográfica**

A generalização cartográfica pode ocorrer em duas variações: de acordo com a *forma geométrica* ou de acordo com a *escala*. A variação de acordo com a forma é usada para registrar a existência de múltiplas representações independentes de escala para uma classe. A variação de acordo com a escala é usada para especificar diferentes aspectos geométricos de uma classe dada, cada qual correspondendo a uma faixa de escalas. A Figura 8 apresenta um exemplo de cada variação.

A notação usada para generalização cartográfica utiliza um quadrado para conectar a superclasse a suas subclasses. As linhas utilizadas são pontilhadas. As palavras *Escala* e *Forma* são usadas como discriminadores. O quadrado é deixado em branco quando as subclasses são disjuntas, e preenchido se as subclasses podem se superpor (Figura 8).

## 4. Extensões ao Modelo OMT-G

Quando o esquema estático indica a necessidade de múltiplas representações de alguma classe, ou quando a aplicação envolve a derivação de alguma classe a partir de outras, propõe-se a criação de um *esquema dinâmico*. Nesse esquema, todos os processos de transformação de dados podem ser especificados, permitindo a identificação de quaisquer métodos que sejam necessários para a implementação. Além do esquema dinâmico, propõe-se a criação de um *esquema de apresentação*, onde são especificadas as diretrizes para a visualização de qualquer representação prevista no esquema estático, usando os recursos do SIG subjacente. Cada classe pode ter várias apresentações, cada qual adequada para uma determinada finalidade de uso – de maneira análoga à definição da cartografia para os mapas em papel.

### 4.1 Esquema dinâmico

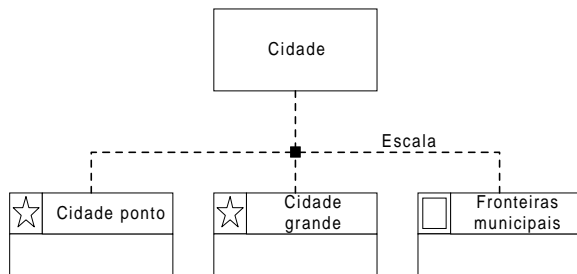
Os esquemas dinâmicos são baseados nas primitivas de classe, conforme definidas para os esquemas estáticos do modelo OMT-G. As classes que estão envolvidas em algum tipo de transformação são conectadas por meio de linhas contínuas, com setas que indicam a direção da transformação. Os operadores de transformação envolvidos e seus parâmetros, quando houver, são indicados por meio de texto sobre a linha que denota a transformação.

No esquema dinâmico, pode-se indicar se o resultado da transformação precisa ou não ser materializado. Classes resultantes muito simples, ou que são passos intermediários em uma transformação mais complexa, frequentemente não precisam ser materializadas, e podem ser armazenadas apenas temporariamente. Tais classes temporárias são indicadas usando linhas tracejadas em seu contorno. As classes que são resultantes de alguma transformação e que precisam ser materializadas (devido à complexidade do processo ou às necessidades específicas da aplicação) são denotadas com linhas contínuas, exatamente como no esquema estático.

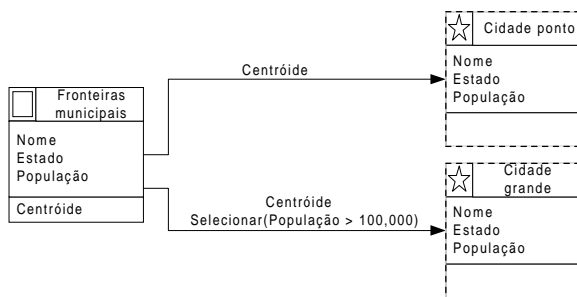
Observe-se que a especificação de transformações no esquema dinâmico é em geral exigida quando as primitivas de generalização cartográfica e de agregação espacial são usadas no esquema estático. Essas duas primitivas são indicativas da possibilidade de produzir uma representação a partir de outras. Quando o esquema estático contém uma primitiva de agregação espacial, existe frequentemente a necessidade de se especificar a operação que vai ser usada para construir o todo a partir das partes.

A Figura 9 apresenta um fragmento de esquema estático onde é usada a primitiva de generalização cartográfica. Ao interpretar esse esquema estático, não fica claro se e como as três representações alternativas para a classe

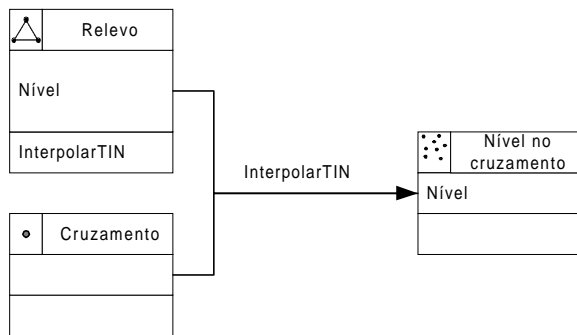
Cidade irão coexistir na aplicação. Embora algum relacionamento entre elas exista, conforme indicado pela primitiva de generalização cartográfica, o esquema estático não explora a semântica que conduz à possibilidade de gerar algumas representações a partir de outras. O esquema dinâmico (Figura 10) resolve esse problema, registrando o entendimento de que a classe *Fronteiras municipais* contém informação suficiente para produzir a geometria das outras duas classes.



**Figura 9 - Representações alternativas para a classe Cidade (esquema estático)**



**Figura 10 - Transformações entre representações da classe Cidade (esquema dinâmico)**

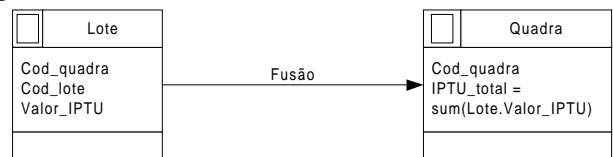


**Figura 11 - Exemplo de operação (esquema dinâmico)**

As transformações indicadas no esquema dinâmico podem relacionar qualquer número de classes originais, bem como qualquer número de classes resultantes, dependendo da natureza da operação de transformação. Cadeias de transformações também podem ser definidas, permitindo dessa forma a especificação de processos complexos de análise espacial. A Figura 11 apresenta um exemplo

desse tipo de operação. Na figura, um geo-campo da classe TIN, representando o relevo, é usado juntamente com uma classe de cruzamentos de vias (nós de rede) para produzir outro geo-campo, Nível no cruzamento, que é representado por um conjunto de amostras. Na classe resultante, a geometria (localização das amostras) é retirada da classe Cruzamento, e o valor do campo em cada posição é calculado através do método *InterpolarTIN*, desta maneira preenchendo o atributo Nível do geo-campo resultante.

A Figura 12 apresenta um exemplo de esquema dinâmico resultante de uma agregação espacial, correspondendo ao fragmento de esquema estático apresentado na Figura 7. Instâncias da classe *Lote* são unidas, usando um operador de generalização cartográfica denominado *Fusão* [7], para criar instâncias da classe *Quadra*. Observe que a transformação também indica que um atributo de *Quadra*, *IPTU\_total*, deve ser preenchido com o somatório dos atributos *Valor\_IPTU* dos lotes correspondentes. Algumas vezes, no entanto, não é possível especificar uma transformação automática como essa, especialmente nos casos em que o todo é obtido primeiro e as partes são então extraídas dele.



**Figura 12 - Agregação no esquema dinâmico**

Um estudo sobre a possível variação de operadores TR, considerando todas as combinações de tipos de representação para as classes original e resultante, foi apresentado em [7] como parte de um arcabouço conceitual sobre múltiplas representações. Os operadores empregados para tais transformações são baseados em algoritmos bem conhecidos nas áreas de geometria computacional, generalização cartográfica e análise espacial.

## 4.2 Esquema de apresentação

O esquema de apresentação, proposto a seguir, procura reunir os requisitos definidos pelo usuário quanto às alternativas de apresentação e saída para cada objeto geográfico. Essas alternativas podem incluir apresentações criadas especificamente para visualização em tela, para impressão na forma de mapas ou cartas, para interpretação visual em um processo de análise, e outras.

Cada apresentação é definida a partir de uma representação contida no esquemas estático ou no esquema dinâmico do nível de representação conceitual. Operações de transformação para apresentação (TA) são especificadas, permitindo obter o aspecto visual desejado a partir da

simples forma geométrica definida para a representação. Observe-se que operações TA não modificam a alternativa de representação definida previamente, nem mudam o detalhamento definido no nível de representação conceitual. Se isso for necessário, uma nova representação tem de ser criada a partir de uma representação existente, usando as ferramentas de especificação de múltiplas representações (como a primitiva de generalização cartográfica) e registrando essa alternativa nos esquemas estático e dinâmico.

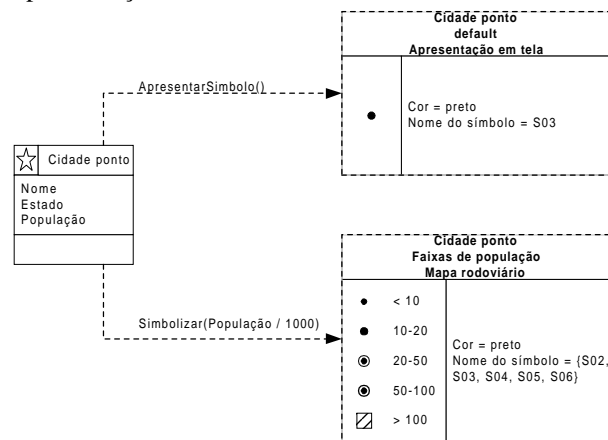
As operações TA mais comuns envolvem a simples definição de atributos gráficos. Como o esquema de apresentação está mais próximo da implementação que os esquemas estático e dinâmico, os recursos gráficos específicos do SIG subjacente podem ser levados em conta. Outros operadores mais sofisticados (classificação, simbolização, exagero, deslocamento, destaque) também podem ser empregados. Um estudo sobre a variedade de operadores para TA foi apresentado em [7].

O esquema de apresentação necessita de apenas três primitivas. A primeira é a própria primitiva de classes, definida para os esquemas estático e dinâmico. A segunda é usada para indicar a operação TA, de maneira semelhante à usada para denotar as transformações no esquema dinâmico. É composta de uma linha tracejada simples, com uma seta que indica o sentido da operação, sobre a qual é especificado o operador a ser usado. No processo de especificação dessa expressão de transformação, quaisquer características geométricas ou atributos alfanuméricos que foram definidos no nível de representação conceitual para a classe podem ser usadas como parâmetros. As linhas indicando operações TA são tracejadas para distingui-las visualmente das operações de transformação de representações, que são especificadas no esquema dinâmico com linhas contínuas. A terceira primitiva serve para especificar uma apresentação, e contém duas seções. A seção superior indica o nome da classe, o nome da apresentação, e a aplicação na qual é usada. A segunda é dividida em duas partes: à esquerda, um pictograma indica o aspecto visual dos objetos após a transformação, à semelhança de uma legenda cartográfica, e à direita são lançadas especificações mais precisas quanto aos atributos gráficos, incluindo cor da linha, tipo de linha, espessura da linha, padrão de preenchimento, cor de preenchimento, e nome do símbolo. A especificação dos atributos gráficos pode ser feita já considerando a codificação de símbolos usada pelo SIG subjacente. Pode existir qualquer número de pictogramas na seção esquerda da primitiva de especificação de apresentações, cada qual associado a um valor ou faixa de valores obtidos a partir das características de cada objeto. Nesse caso, a seção da direita deve detalhar os atributos gráficos de cada apresentação gerada. Atributos comuns podem ser especificados apenas uma vez,

enquanto atributos variáveis são especificados como listas de valores individuais. Como no caso do esquema dinâmico, as apresentações são indicadas com linhas tracejadas quando não precisam ser materializadas no banco de dados, e com linhas contínuas no caso contrário.

Cada classe georreferenciada especificada no esquema estático precisa ter pelo menos uma apresentação correspondente especificada no esquema de apresentação. Caso exista mais de uma apresentação para uma dada representação, uma delas deve ser identificada como a *default*. Alternativamente, cada usuário ou aplicação pode eleger sua apresentação *default*.

Observe-se que a definição de apresentações a partir de uma representação é um conceito que pode ser também aplicado no caso de objetos convencionais, ou seja, objetos que não possuem características geométricas ou geográficas. Isso significa que aspectos visuais alternativos para tabelas e campos podem também ser especificados no nível de apresentação, simplificando o trabalho final de implementação.



**Figura 13 - Esquema de apresentação para a classe Cidade ponto**

Como exemplo, considere-se a definição de apresentações para a classe *Cidade ponto*, apresentada anteriormente. Primeiro, uma apresentação *default* para visualização em tela é definida. Em seguida, uma apresentação diferente é especificada, na qual diferentes símbolos são atribuídos às instâncias dependendo do valor do atributo *População*, para um mapa rodoviário (Figura 13).

A Figura 14 introduz outro exemplo de esquema de apresentação. A classe básica, neste caso, representa setores censitários como polígonos. Três diferentes apresentações são derivadas desta representação: a primeira é a apresentação *default* para a tela, usando um padrão hachurado; a segunda mostra os resultados de uma classificação sobre o atributo de renda média do chefe do domicílio; a terceira apresenta apenas os setores onde a densi-

dade demográfica é considerada excessivamente alta, como parte de uma análise de risco para a saúde.

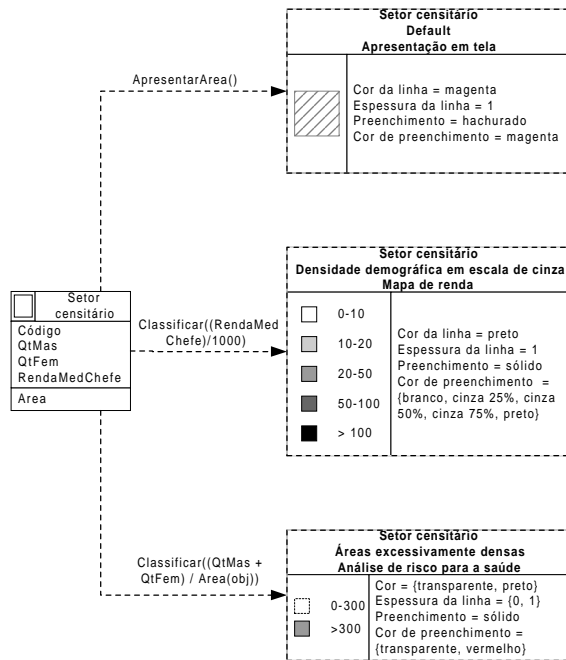


Figura 14 - Esquema de apresentação para a classe Setor censitário

## 5. Conclusão

Os recursos complementares ao modelo OMT-G apresentados neste trabalho são fundamentalmente importantes para a correta concepção e documentação de aplicações geográficas. A aproximação entre o modelo mental do usuário e o modelo de implementação passa a se dar com maior amplitude, uma vez que os aspectos dinâmicos e a aparência visual dos objetos são melhor detalhados.

Ao final do processo de modelagem, uma avaliação da funcionalidade exigida para as transformações pode ajudar a orientar o processo de escolha do SIG mais adequada para a implementação. Adicionalmente, o conjunto de esquemas proposto aqui pode facilitar uma eventual migração da aplicação para uma plataforma diferente da que foi utilizada para a implementação original. Nesse sentido, os esquemas dinâmico e de apresentação constituem também ferramentas de engenharia reversa para aplicações já desenvolvidas.

Este trabalho está prosseguindo, através do desenvolvimento de um conjunto de funções e extensões a um SIG comercial, visando demonstrar a validade dos conceitos aqui desenvolvidos no caso de uma aplicação real. Uma ferramenta de modelagem de dados geográficos baseada no modelo OMT-G, incluindo todas as primitivas apresentadas para os esquemas estático, dinâmico e de apre-

sentação e apoiada no software de diagramação Visio, está atualmente em fase de testes.

## 6. Referências

- [1] Abrantes, G., Carapuça, R. Explicit representation of data that depend on topological relationships and control over data consistency. In *Proc. Fifth European Conference and Exhibition on GIS*, 1:869-877, 1994.
- [2] Borges, K. A. V. *Modelagem de Dados Geográficos: Uma Extensão do Modelo OMT para Aplicações Geográficas*, Dissertação de Mestrado, Fundação João Pinheiro, Belo Horizonte, 1997.
- [3] Borges, K. A. V., Laender A. H. F., Davis Jr., C. A. Spatial data integrity constraints in object oriented geographic data modeling. In *Proceedings of the 7th International Symposium on Advances in Geographic Information Systems (ACM GIS'99)*, 1-6, 1999.
- [4] Borges, K. A. V., Laender, A. H. F., Davis Jr., C. A. OMT-G: a OMT-G: An Object Modeling Technique for Geographic Applications. Submetido à publicação em *GeoInformatica*, 2000.
- [5] Câmara, G. *Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos*. PhD thesis, INPE, 1995.
- [6] Câmara, G., Casanova, M. A., Hemerly, A., Magalhães, G. C. e Medeiros, C. M. B. *Anatomia de Sistemas de Informação Geográfica*. UNICAMP, 1996.
- [7] Davis Jr., C. A., Laender, A. H. F. Multiple representations in GIS: materialization through geometric, map generalization, and spatial analysis operations. In *Proc. 7th ACM Intl. Symp. Advances in GIS*, 60-65, 1999.
- [8] Elmasri, R., Navathe, S. *Fundamentals of database systems*. 2<sup>nd</sup> Edition. Addison-Wesley, 1994. 873p.
- [9] Goodchild, M. F. Geographical data modeling. *Computers & Geosciences*, 18(4):401-408, 1992.
- [10] Kösters, G., Pagel, B., Six, H. Object-oriented requirements engineering for GIS-applications. In *Proc. Intl. Workshop Advances in GIS*, 61-69, 1995.
- [11] Laender, A. H. F., Flynn, D. J. A semantic comparison of modelling capabilities of the ER and NIAM models. In Elmasri, R., Kouramajian, V., Thalheim, B. (editors) *Entity-Relationship approach – ER'93*, 242-256, Springer-Verlag, 1994.
- [12] Rational Software Corporation. *The Unified Modeling Language: notation guide*, version 1.1 July 1997. (<http://www.rational.com>).
- [13] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorenzen, W. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.

# Towards A Unified Framework For Geographical Data Models

GILBERTO CÂMARA<sup>1</sup>,  
ANTÔNIO MIGUEL VIEIRA MONTEIRO<sup>1</sup>,  
JOÃO ARGEMIRO PAIVA<sup>1</sup>,  
JONAS GOMES<sup>2</sup>,  
LUIZ VELHO<sup>2</sup>

<sup>1</sup>IMPA- Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina, 11, 22460 Rio de Janeiro, RJ, Brasil

<sup>2</sup>INPE—Instituto Nacional de Pesquisas Espaciais, Caixa Postal 515, 12201 São José dos Campos, SP, Brasil

**Abstract.** This paper describes a unified framework for the problems of modelling and processing geographical entities. We propose a general definition of *geographical objects*, and show that the different types of geographical data can be expressed as particular cases of this definition. Furthermore, we present a taxonomy for the various types of GIS operations, defined in terms of the properties of this definition. Our goal is to argue that GIS data types and operations can be defined based on a single formal notion, which encapsulates the GIS concepts of both fields and objects, with important consequences for system and interface design, interoperability issues and language proposal.

## 1 Introduction

The literature on GIS usually considers two broad classes of models of geographic information: *field-based* and *object-based* (Coucelis, 1992). The *field model* represents the geographical data as a set of spatial distributions over the geographical space. The *object model* represents the world as a surface occupied by discrete, identifiable entities, with a geometrical representation and descriptive attributes.

Some authors have already pointed out that the *field* and *object* views have their roots in different cognitive perceptions of the geographical space and that they are extreme, idealised notions from a more complex taxonomy (Burrough and Frank, 1995; Coucelis, 1996).

Notwithstanding the importance of the conceptual debate, it is important to consider the challenges of representing these concepts in a computer environment. We consider that the computer-based modelling of geographical reality requires a level of formalization and generalization (either explicit or implicit), which is necessary to define precisely the nature of operations to be performed in a GIS.

An additional important consideration is the recent emphasis on the GIScience literature on the definition of ontologies of the geographical space (Smith and Mark, 1998). As the authors have pointed out on a companion paper (Câmara et al, 2000), the idea of ontologies could form the basis for a general model for spatial information data. Therefore, in the application domains where such ontologies can be defined, it should be possible to define a

unified perspective for modelling and analysis of geographical entities can supersede the field-object dichotomy.

In this perspective, we propose a general formal definition of *geographical objects*, and show that the different types of geographical data can be expressed as particular cases of this definition. Furthermore, we present a taxonomy for the various types of GIS operations, in terms of properties of this definition. Our goal is to show that GIS data types and operations can be defined based on a single formal notion, with important consequences for system and interface design, interoperability issues and language proposal.

One of the most relevant aspects of this definition is its impact on the questions of interoperability. It has been recognized that interoperability in GIS requires a level of semantic modelling to account for the correspondence of concepts between different systems (Câmara et al, 1999). By establishing a formal notion and referring the concepts presented in GIS literature to this definition, we could provide a basis for a unified semantic framework for GIS.

The paper is organized as follows. Section 2 introduces the abstraction paradigm used as a basis for our concepts. Section 3 proposes a general definition for geographical objects, discusses the representation of these objects and gives a classification. Section 4 shows how traditional GIS concepts are related with the definition of a geographical object. Section 5 summarizes the typical operations on geographical data, as seen in the perspective of our framework. Section 6 outlines some consequences

of this approach, both from a theoretical and a practical point of view.

## 2 Abstraction Paradigms For Geographical Modelling

Geographical data modelling for GIS can be viewed as a special case of computational modelling of physical phenomena. In computational modelling, a very important and necessary step is to establish mathematical models, which create the abstract descriptions of the real-world

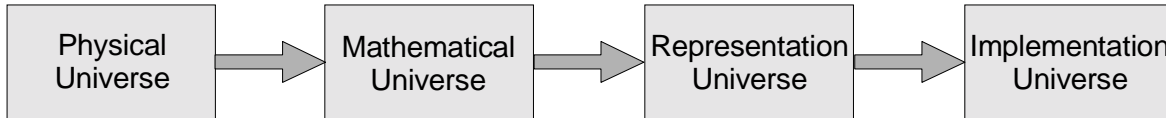


Figure 1 – Abstraction levels in Computer Modeling (from Gomes and Velho, 1995)

entities of interest. In a classical paper in the area of geometric modelling, Requicha (1980) established a conceptual framework which distinguishes between the physical, mathematical and representational levels of abstraction. His work was further extended by Gomes and Velho (1995) which propose the “four universes paradigm” as a general modeling mechanism for applied computational mathematics in general, and not only for geometric modelling. The four abstraction levels are described as (See Figure 1):

1. The *physical* universe, which comprises the real-world entities that will be modelled in the computer.
2. The *mathematical* universe, which includes a formal definition of the entities which are included in the model.
3. The *representation* universe, which defines how the various continuous models are discretized.
4. The *implementation* universe, where the data structures are associated to the discretized objects of the representation universe.

These abstraction levels have been successfully used in different areas of computer graphics. An extensive use of the paradigm is found in (Gomes et alii. 1998), where it has enabled the formulation of a robust conceptual approach both for theoretical issues and implementation, related with the subject of warping and morphing.

Our experience has shown that the use of the four universes paradigm for geographic data, is also particularly suited: At the physical universe level, we find elements such as *parcels, rivers and soil maps*; at the mathematical level, we define *geographical objects*, and

specialise this notion into classes of geographical data which encompasses traditional concepts from GIS such as *fields* and *objects*; at this level, we distinguish between *raster* and *vector* representations, which may be further specialised, such as *grids, TINs, image* structures for raster and *arc-node* and *arc-node-polygon* structures; at the implementation level the actual coding takes place and we find data structures such as *R-trees* and *quad-trees*.

One of the important consequences of the multiple levels of abstraction paradigm is to make an explicit

reference to design decisions which are very often implicitly and informally defined. For example, the use of raster data structures to store terrain models in a computer implies a mathematical formulation of what is a terrain. In this approach, we are compelled to define the elements of each universe and to establish the relation between the elements of each universe.

## 3 Geographical Objects

Using the abstraction paradigm introduced in the previous section, the most important issue is concerned in characterizing the mathematical entities which will describe the elements from the physical world. In the case of geographical data, this leads us to the concept of a *geographical object* (*geo-object*, or simply *go*) as the basic element of the mathematical universe. A geo-object is a triple  $go = (S, A, f)$  where:

1.  $S \subset \mathbb{R}^2$  is a subset of the Euclidean plane, it is the *geometrical support*.
2.  $A$  is a set of *attribute domains*  $A_1, \dots, A_n$ .
3.  $f: S \rightarrow A_1 \times A_2 \times \dots \times A_n$  is the *attribute function* of the geo-object, which associates, to each location in the support, a value on the set of attribute domains.

Note that this definition mimics that of a graphical object introduced in (Gomes et alii., 1996). It caters the different subtypes of geographical data. We should remark that this definition, in accordance to our paradigm, is completely generic and is not bound to any particular representation.

A geo-object is *geo-referenced* if there exists a parameterization  $g$  from the geometrical support  $S$  to the surface of the earth. Mathematically the map  $g$  can be described approximately by a parameterization from  $S$  to the surface of a sphere. Different parameterizations are

possible related with the various map projections of the sphere.

### 3.1 Representation of Geographical Objects

Once we have a geo-object the first step towards implementation is to obtain its representation. The representation of a geo-object consists in discretising both its geometric support and the attribute function. The representation of the geometric support consists in representing its topology and geometry. This topic is extensively studied in the area of geometric modelling. In GIS it is used mainly the representation techniques based on decomposition. These techniques employ a top-down methodology where the geometric support is decomposed into simpler geometrical objects which are easier to represent.

A very important geographical data is a terrain. As a geo-object a terrain is a real valued function  $f: S \rightarrow R$ , where  $S$  is a subset of the euclidean plane. Two commonly used representations for terrains are the *TIN* (Triangulated Irregular Network) and the *grid*. A TIN representation subdivides the geometrical support  $S$  of the terrain into triangles in such a way to form a triangulation (intersecting triangles should share a vertex or edge). A grid representation subdivides the geometrical support  $S$  into small rectangles so as to construct a lattice. Figures 2(a) and 2(b) show, respectively, a TIN and a grid representation of a terrain.

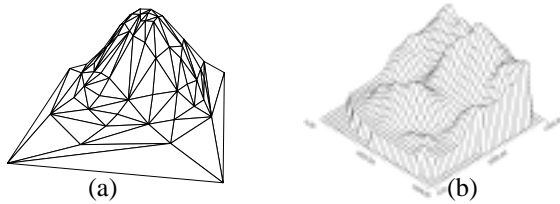


Figure 2 – TIN (a) and Grid representation (b) of terrain.

If we are interested only in the geometry and topology of the terrain it is enough to use a TIN or a grid representation. Nevertheless, in order to obtain a complete representation of the geo-object we must represent the attribute function on the TIN or on the grid. This is generally attained by sampling the function at each element of the representation. On a TIN samples are generally taken at the vertices of the triangles; on a grid samples can be taken either at the vertices of the grid or at the center of each rectangular cell.

Note that because the grid decomposition can be easily structured into a rectangular lattice, this representation is completely characterized by giving the

number  $m \times n$  of decomposition cells (rectangles) and the value of the attribute function in each cell. The structuring is given by the natural row/columns ordering of the lattice. The grid representation of a geo-object is called a *matrix* or *raster* representation. The order  $m \times n$  of the decomposition matrix is called the spatial resolution of the representation.

To conclude the representation of the geo-object the attribute function should assume only a finite number of attribute values (this corresponds to the use of a specified number of bits to represent the attribute values). The process of discretizing the attribute function is called *quantization* (this comes from the usage of the term in the area of image processing). Each of the values assumed by the function is called a *quantization level*. The number of quantization levels to be used depends on the nature of the geographic data being represented.

We should remark that if the attribute function is quantized to  $n$  levels,  $a_1, \dots, a_n$ , then it determines a

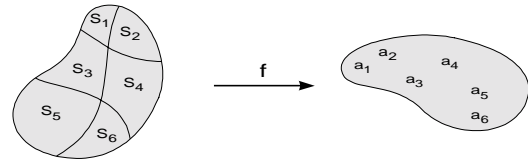


Figure 3 – Attribute function with  $n$  values and associated partition

partition of the geometric support of the geo-object into a collection of disjoint sets  $S_i, 1=1, \dots, n$ , such that the attribute  $f$  at each set  $S_i$  is constant and equal to  $a_i$ . This is illustrated in the Figure 3 below.

An important issue when representing geo-objects consists in reconstructing the object from its discrete representation. We will not discuss this topic in this article.

### 3.2 Classification of Geographical Objects

Geo-objects are classified according to the topology of the support, and the values of the attribute function. Four different classes are possible: simple, composite, homogeneous and non-homogeneous geo-objects.

A geo-object is called *simple* if its support  $S$  is a connected region in  $R^2$ . Otherwise it is called a *composite* object. A geo-object is called *homogeneous* if its attribute function assumes a constant value  $f(s) = (a_1, \dots, a_n), \forall s \in S$ . Otherwise, the geo-object is called *non-homogeneous*. Some examples will be given below.

### 3.2.1 Simple Geo-Objects

Figure 4 shows an example of a simple geographic object. The object describes a country named “Brazil”. The geometric support  $S$  is a connected region, and it has two constant attribute values: “name” and “population”.

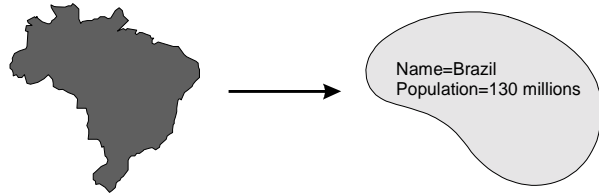
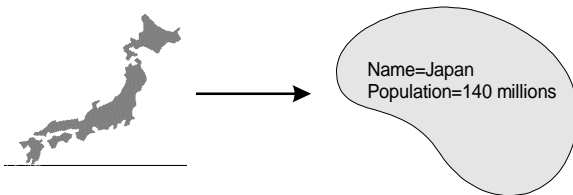


Figure 4 - A simple and homogeneous geo-object.

In practice, the notion of a *single, homogeneous geo-object* is too simplistic and we need to use the more complex classes of geo-objects to characterize the geographical entities.

### 3.2.2 Composite homogeneous Geo-object

In this case, we are dealing with a geographical object where the geometric support  $S$  has several connected components and the attribute function assumes a constant value. Figure 5 shows an example, which describes the country named “Japan”, where  $S$  is a set



with four connected components (representing the main islands of the Japanese archipelago), and the attribute domains  $A_1$  and  $A_2$  are “name” and “population”

Figure 5 - Example Of A Composite Homogeneous Object.

### 3.2.3 Simple Non-homogeneous Geo-object

For these object types, the support  $S$  is composed of a connected region, but the attribute function  $f$  varies for each point in the region. Depending on the number of levels used in the representation of the attribute function, the geometric support is partitioned into a finite number of

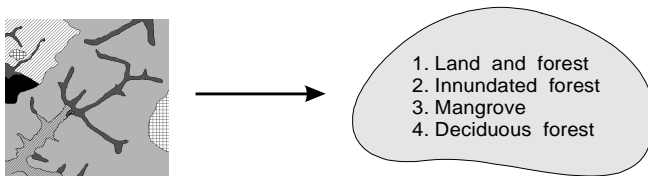


Figure 6 - Example of a simple and non-homogeneous object.

sets. This concept corresponds to the notion of fields (Goodchild, 1992), such as vegetation maps and topography. Figure 6 shows an example of a simple non-homogeneous object (a vegetation map). Note the partitioning of the geometric support as we described in Figure 3.

### 3.2.4 Composite non-homogeneous Geo-object

In this case, the geometrical support  $S$  is composed of several connected components, and the attribute function is not constant over  $S$ . This definition, in its generality, has no direct counterpart in the traditional GIS elements. However, there is a very important special case, namely:

- The geometric support is a disjoint collection of connected regions,  $S = \cup S_i$ ,  $S_i \cap S_j = \emptyset$  ( $i \neq j$ );
- The attribute function  $f$  is constant on each set  $S_i$ , and the constant values assumed are distinct, in general

This special type of composite non-homogeneous geo-object constitutes a formal definition of a *coverage* in the ARC/INFO™ system (ESRI, 1994).

Figure 7 shows an example of an ARC/INFO™ coverage, which corresponds to a map of the South Asia region, composed of different islands. The map is dealt by the system as a single object, whose components (the geometrical supports associated to each island) are mapped to different values of the attribute set.

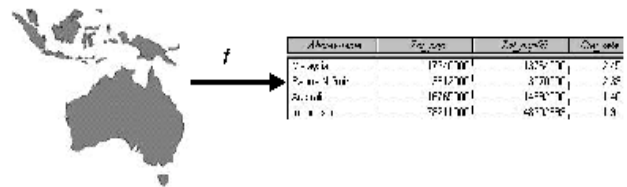


Figure 7 Example of a composite, non-homogeneous object.

## 4 Correspondence Of Definition To Gis Literature

In this section, we will attempt to relate the concepts introduced above with the traditional concepts in the GIS literature, especially those used by the OpenGIS® consortium (OpenGIS, 1998a; OpenGIS, 1998b). Since the latter definitions are mostly informal, an exact matching is not always possible, but the approximate correspondences are already illustrative of the hazards of semantic model conversion between systems.

The OpenGIS® model is based on an abstract class (*feature*) which has two specialisations: *feature with geometry* and *coverage*.



The definition of *feature with geometry* allows for complex geometrical representations to be associated to the same feature and for different features to share the same geometrical representation. The notion of *feature with geometry* in OpenGIS<sup>®</sup> (OpenGIS, 1998a) corresponds roughly to our definition of a composed, homogeneous object (Section 3.2.2) or a single non-homogeneous geo-object (Section 3.2.3).

An OpenGIS<sup>®</sup> *coverage* is an association between a geometric description of entities and a set of attributes; this association is defined by a coverage function (or *c\_function*)  $f:(geometry) \rightarrow (attribute\ set)$ . The OpenGIS<sup>®</sup> proponents chose not to use a representation-independent definition for *coverage*, and to define it in terms of specific geometrical representations, such as a *grid coverage*, *TIN coverage* or *geometry coverage* (OGC, 1998b).

Most of the OpenGIS<sup>®</sup> specialisations of coverages, such as *sample*, *grid*, *TIN* and *image* can be considered as computer representations of the concept of a *simple non-homogeneous object* (Section 3.2.3), using specific data structures. However, their definition of *geometry coverage* can be considered as a special case of a *composite non-homogeneous object* (section 3.2.4).

Thus, we can observe that the definitions of *coverage* on the OpenGIS are based on different formal concepts. In other words, a *grid coverage* in OpenGIS does not have the same formal basis as a *geometry coverage*. This situation may lead to potential problems in understanding and using this concept as a basis for interoperability. In fact, in a previous work (Câmara et al, 1999), we have argued against the choice of the OpenGIS consortium of using industry terminology, such as *feature* and *coverage*, which is already content-rich and are associated by the users with existing semantic concepts (OGC, 1998a; OGC, 1998b).

Table 1 provides a resume of the relation between our definitions and established industry notions.

**TABLE 1 - CORRESPONDENCE OF DEFINITIONS**

	Connected geometrical support	Non-Connected geometrical support
Homogeneous		<i>Simple feature (OpenGIS)</i>
Non-homogeneous	<i>Tin, Grid, Image and SurfaceCoverage (OpenGIS)</i>	<i>Coverage (ARC/INFO), Feature Collection Coverage (OpenGIS)</i>

## 5 OPERATIONS ON GEOGRAPHICAL DATA

One of the main purposes of a single formal definition for geographical objects is to provide a unified perspective on the semantics of GIS operations. In current practice, the use of different models for fields and object usually leads to implementation of different subsystems on a GIS: raster map operations implementing Tomlin's Map Algebra (Tomlin, 1990) and vector spatial queries with languages similar to Spatial SQL (Engenhofer, 1994). This situation is not always desirable. For example, "overlap" is a well-known concept for denoting a topological configuration between two geographical entities (Engenhofer et al., 1994). In many GIS, it is possible to inquire if two entities represented by vector polygons "overlap", but it is not straightforward to inquire if the same property holds for two digital terrain models.

### 5.1 Atomic Operations on Geographical Objects

A general GIS should have basic operations which are generally applicable to different types of geographical entities, including both *field* and *object* models. Therefore, we have chosen to characterize what we propose to be a minimal set of atomic operations, applicable to different types of geographical objects (described in section 3). Upon this minimal set, more complex operations can be applied, eventually leading to more specific operation classes such as Map Algebra. In this section, we propose such a minimal set and in the next one, indicate how these atomic operation can be used as building blocks of more complex ones.

We consider three classes of operations with geo-objects: Attribute based, Spatial based and Creation-Delete operations. We will describe these operations below.

In keeping to our paradigm, which distinguishes between the mathematical and the representation universes, our definition of operations on geographical objects does not consider representation issues. These issues are considered to be implementation-dependent. For example, the practical implementation of a mathematical operation between geo-objects may require their conversion from a vector to a raster representation. Ideally, this conversion should be performed automatically by the system (under certain rules). The fact that most commercial systems require the user to request an explicit vector-to-raster conversion only serves as indication that GIS technology would benefit from a formal basis, much as the database technology was improved by Codd's relational model (Codd, 1962).

Using a formal definition as a basis, we can distinguish which operations are essential part of the

properties of geographical objects and those which are constrained by representation-based issues.

### 5.1.1 Attribute Based Operations.

Two basic operations are proposed: VALUE ( $\lambda, A_i$ ), which returns the value of attribute  $A_i$  at location  $\lambda$  (this operation is actually the value of the attribute function for attribute  $A_i$ ), and ASSIGN ( $\lambda, A_i, a$ ), which assigns a value  $a$  for attribute  $A_i$  at location  $\lambda$ . For shorthand notation, we indicate these operations as  $f(\lambda, A_i)$  and  $\alpha(\lambda, A_i, a)$ , respectively.

### 5.1.2 Spatial based Operations

Spatial based operations can be further classified based on a spatial predicate denoted by  $\xi$ . As an example we have the following cases:

#### 5.1.2.1 Topological restrictions

Given a pair of geo-objects  $go_i$  and  $go_j$ , and a topological predicate  $\theta$ , these operations return a boolean value ( $\{true, false\}$ ) based on the result of the application of the topological predicate  $\theta(go_i, go_j)$ . The topological predicates proposed are EQUAL, DISJOINT, INTERSECT, TOUCH, CROSS, OVERLAP, CONTAINS, WITHIN, and RELATE, as defined by the OpenGIS consortium (OGIS, 1998).

#### 5.1.2.2 Distance-based operations

Given a pair of geo-objects  $go_i$  and  $go_j$ , the DISTANCE( $go_i, go_j$ ) operation returns the distance measure between these two objects. For shorthand notation, we indicate this operation as  $dist(go_i, go_j)$ . We can also consider a boolean predicate  $\delta(go_i, go_j, d)$  which returns true or false depending whether the distance between  $go_i$  and  $go_j$  is smaller than a given value  $d$ .

#### 5.1.2.3 Direction-based operations

Direction relationships between objects can be qualitatively described using the notion of cardinal directions (Frank, 1991). The object support is abstracted using its minimum bounding rectangle, and the direction relationships are expressed through a 3x3 matrix representing true or false values for NORTHWEST, NORTH, NORTHEAST, WEST, CENTER, EAST, SOUTHWEST, SOUTH, and SOUTHEAST relations. Given a pair of geo-objects  $go_i$  and  $go_j$ , the DIRECTION( $go_i, go_j$ ) operation returns the cardinal direction matrix between these two objects. We can also consider a boolean predicate PATH( $go_i, go_j, dir\_rel$ ) which returns true or false depending on the direction relation  $dir\_rel$  value between  $go_i$  and  $go_j$

### 5.1.3 Combined Spatial and Attribute Based Operations

Given a pair of geo-objects  $go_i$  and  $go_j$ , this class of operation return a value which is calculated on the attribute values of  $go_i$ , based on the geometrical support of  $go_j$ . We call this types of operation REGION, and consider specialisations such as REGION\_MAX ( $A_i, go_i, go_j$ ) and REGION\_AVE( $A_i, go_i, go_j$ ), which compute, respectively, the maximum and the average value of the attribute  $A_i$  of the geo-object  $go_i$ , constrained by the geometrical support of the geo-object  $go_j$ . Figure 8 shows an example of a "region" operation, where the attribute of first geo-object is a numerical value, and we compute the maximum value of the region indicated by  $S_2$ .

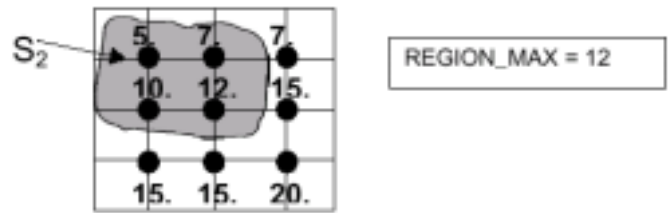


Figure 8 – Example of a region operation.

Again, it should be stressed that some operations need a conversion between representations of geo-objects (such as the conversion from a grid to a TIN). In this case the operation depends on the conversion technique used.

## 5.2 Composite Operations on Geographical Objects

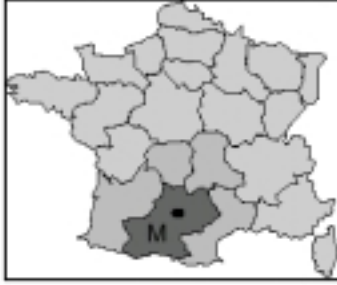
In this section, we indicate how the usual operations of spatial queries and map algebra can be expressed in terms of the atomic operations proposed. We will consider three such types of operations: spatial selection, spatial join and point operations (which usually are implemented in separate systems).

#### Definition 1. Spatial Selection.

The *spatial selection* operation can be defined as follows. Given a set of geo-objects  $GO = \{go_1, \dots, go_n\}$ , a reference geo-object  $go^*$ , the *spatial selection* operation  $\varphi: GO \rightarrow GO$ , given a spatial predicate  $\xi$  which relates the geo-objects  $go \in GO$  to  $go^*$  is defined by:

$$\varphi_{\xi}(GO) = \{ go \in GO \mid \xi((go, go^*)) \}.$$

The *spatial selection* operator is such that the output is a subset of the original set, composed of all geo-objects that satisfy the geometrical predicate, as in the example: “select all regions of France which are adjacent to the Midi-Pyrinees regions (which contains the city of Toulouse)”, illustrated in Figure 9.



**Figure 9 – Example of spatial selection operation**

*Definition 2. Spatial Join.*

Let  $GO_1$  and  $GO_2$  be two sets of geo-objects. Let  $\xi$  be a spatial predicate computable for every pair of objects  $(g_{o_1}, g_{o_2})$ , where  $g_{o_1} \in GO_1$  and  $g_{o_2} \in GO_2$ . The spatial join operation  $\theta: GO_1 \times GO_2 \rightarrow GO_1 \times GO_2$  is such that:

$$\theta_{\xi}(GO_1, GO_2) = \{ (g_{o_1}, g_{o_2}) \in (GO_1, GO_2) \mid \xi(g_{o_1}, g_{o_2}) \}.$$

Spatial join is an operation where a comparison between two sets of geo-objects  $GO_1$  and  $GO_2$  takes place, based on a spatial predicate which is computed over the geometrical support of these sets. The name “spatial join” is employed by analogy to the join operation in relational algebra. The result of the spatial join operation is a *set of object-pairs*, which satisfy the spatial restriction. One example of spatial join would be: “Find all native reservations located closer than 50 km to the main roads in Amazonia”. The answer is a set of pairs of geo-objects (reservation, road). We can also consider operations where an output is generated, based on one or more inputs.

*Definition 3. Point operations.*

Let  $g_{o_1}, g_{o_2}, \dots, g_{o_n}$  be geo-objects used as input,  $g_{o_{n+1}}$  be the output geo-object,  $A^1, A^2, \dots, A^{n+1}$  be the attribute sets associated to these geo-objects and  $S_o$  be the geometric support of the output. Let  $A_i^k$  be the  $i$ -th attribute of the  $k$ -th geo-object. The *point operation*  $\Pi: S_o \times A_1^1 \times A_1^2 \times \dots \times A_1^n \rightarrow A_i^{n+1}$  induces a function  $\pi$  such that:

$$f_{n+1}(p) = \pi(f_1(\lambda, A_1^1), \dots, f_n(\lambda, A_1^n)), \forall p \in S.$$

For point operations, the value of the output attribute at each location is a function only of the input attribute

values at the corresponding location. One example would be the boolean operations: “Calculate a soil aptitude map based on climate, soil, and slope maps, where the conditions are such that a soil is deemed “*good for agriculture*” if it rains more than 1000 m/year and the soil has a *ph* between 6.5 and 7.5, and the slope is less than 15%”. This operation can be easily defined, based on the proposed atomic operations VALUE and ASSIGN (defined in section 5.1.1).

Other operations on geo-objects can be defined in a similar fashion. Note that the above definitions make no explicit assumption on the specific type of geo-object considered. In current systems, the first two operations are which usually implemented in a separate sub-system than the third one. As we have argued, there is no compelling reason that this should be the case, and that the atomic operations proposed in section 5.1 should be general enough to serve as a basis for an interoperable language for geographical data.

## 6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a generic definition for geographical objects. From this definition, we obtained different types of geographical entities and formulated a minimal set of general GIS operations. We have shown that this minimal set can be used to build more complex operations, equivalent to usual definitions of spatial selections and map algebra, but which are not constrained to any particular model of geographical data (such as the field or object models).

We point out that a unified formal definition of geographical objects does not, in any respect, diminish the relevance of the conceptual debate at the semantic level. The field and object views are, in fact, based on deeply rooted notions in human perception of his environment. Our primary aim was to indicate that the computerized modelling of the geographical data is necessarily a reductionist view, leading to formal notions which provides a basis for unified semantics of GIS operations.

We should point out that the concept of a geographical object introduced allows for new categories of geo-objects whose geometric support are not defined as a subset of the plane. This makes it very suitable to define, for instance, a concept of a volumetric geographical object which seems to be quite adequate to bring volumetric visualization techniques into the realm of GIS systems. More generally, GIS systems will incorporate more and more multimedia techniques. The concept of geographical object is ready to absorb this tendency. We intend to discuss these issues in our future work.

## 7 ACKNOWLEDGEMENTS

This work has been partially motivated as part of the effort on the development of SPRING, a GIS built by INPE and its partners. In particular, the first author would like to thank his colleagues on the SPRING project, especially Ubirajara Freitas and Ricardo Cartaxo, for many a fruitful discussion on the topic of GIS modelling. This work has been partially financed by a joint NSF/CNPq project on "Interoperability on GIS" (process CNPq 480322/99).

## 8 REFERENCES

- Burrough, P.A.; Frank, A. Concepts and Paradigms in Spatial Information: Are current geographical information systems truly generic ?. *International Journal of Geographic Information Systems*, 9(2):101-116, 1995.
- Burrough, P.A.; Frank, A. (eds). *Geographic Objects with Indeterminate Boundaries*. London, Taylor and Francis, 1996.
- Couclelis, H. Towards an Operational Typology of Geographical Entities with Ill-defined Boundaries. In: Burrough, P.A.; Frank, A. (eds). *Geographic Objects with Indeterminate Boundaries*, chap3, pp. 45-56. London, Taylor and Francis, 1996.
- Câmara, G.; Souza, R.C.M.; Freitas, U.M.; Garrido, J.C.P (1996). "SPRING: Integrating Remote Sensing and GIS with Object-Oriented Data Modelling". *Computers and Graphics*, vol.15 , n.6, July 1996.
- Câmara, G.; Thomé, R.; Freitas, U.; Monteiro, A.M.V. "Interoperability In Practice: Problems in Semantic Conversion from Current Technology to OpenGIS". *2nd International Conference for Interoperability and Geographic Information Systems*. Zürich, Switzerland, March 10-12, 1999.
- Câmara, G.; Souza, R.C.M.; Monteiro, A.V.M.; Paiva, J.A.C. "Action-Driven Ontologies of the Geographical Space: Beyond the Field-Object Debate". Submitted to GIScience Conference, 2000.
- Clementini, E.; Di Felice, P.; Van Oosterom, P. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In: *Proc. of Third International Symposium on Spatial Data Handling* (Singapore, 1993), pp. 277-295.
- Egenhofer, M. Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, 6:86-95, 1994.
- Egenhofer, M.; Mark, D.; Herring, J. *The 9-Intersection: Formalism and its Use for Natural Language Spatial Predicates*. NCGIA Technical Report 94-1, May 1994.
- ESRI. *Arc/Info Data Management Concepts, data models, database design, and storage*. Redlands, CA, Environmental Systems Research Institute, 1994.
- Gomes, J.M.; Velho, L. Abstraction Paradigms for Computer Graphics. *The Visual Computer*, 11:227-239, 1995.
- Gomes, J.; Costa, B.; Darsa, L.; Velho, L. Graphical Objects. *The Visual Computer*, 12(6):269-282, 1996
- Gomes, J.; Costa, B.; Darsa, L.; Velho, L. *Warping and Morphing of Graphical Objects*. Morgan Kaufman Publishers, San Francisco, , 1998
- Güting, R.H. An Introduction to Spatial Database Systems. *VLDB Journal*, 3(4), October 1994.
- OGC - OpenGIS Consortium. *Topic 5: The OpenGIS Feature*, 1998a..
- OGC - OpenGIS Consortium. *Topic 6: The Coverage Type and Its Subtypes*, 1998b..
- Smith, B.; Mark,D., 1998. Ontology and Geographic Kinds. In: T.Peucker and N.Chrisman (eds), *International Symposium on Spatial Data Handling*. Vancouver, Canada, pg. 308-320.
- Tomlin, D. *Geographic information systems and Cartographic Modeling*. Prentice Hall, New York, 1990

# Ontologias e Interoperabilidade Semântica entre SIGs

FREDERICO FONSECA<sup>12</sup>

MAX EGENHOFER<sup>1</sup>

KARLA A. V. BORGES<sup>2</sup>

<sup>1</sup>National Center for Geographic Information and Analysis  
Department of Spatial Information Science and Engineering  
University of Maine, Orono, ME 04469-5711, USA

[{fred,max}@spatial.maine.edu](mailto:{fred,max}@spatial.maine.edu)

<sup>2</sup> Prodabel - Empresa de Informática e Informação do Município de Belo Horizonte

Av. Presidente Carlos Luz, 1275

31230-000 - Belo Horizonte - MG - Brasil

[karla@pbh.gov.br](mailto:karla@pbh.gov.br)

**Abstract.** The complexity and richness of geospatial data and the difficulty of their representation create specific problems for geographic information systems interoperability. This paper proposes the creation of software components from ontologies as a way to integrate geographic information. These software components are derived from ontologies using an object-oriented mapping. The translation of ontologies into active information system components leads to ontology-driven geographic information systems. The approach introduced by this work enables the use of the architecture by users developing new applications and by GIS database designers. The results presented here answer many questions presented in past works that dealt with the quest for a new GIS. The proposed architecture is up to date with requirements of modern information systems that should provide interoperability and move from managing data and information to managing knowledge.

**Resumo.** A complexidade e riqueza das informações geográficas e a dificuldade de sua representação cria problemas específicos na tentativa de sua integração. Este trabalho apresenta a estrutura de um Sistema de Informação Geográfico baseado em ontologias, que são o principal componente neste tipo de sistema. Um mapeamento baseado em orientação a objetos é usado para criar classes a partir das ontologias. Esta abordagem proporciona um alto grau de interoperabilidade, permitindo integração parcial de informações, quando a integração completa não é possível. Este sistema permite ainda o reaproveitamento de classes já desenvolvidas incorporando, assim, nos novos sistemas, o conhecimento existente em outros SIGs. Os resultados deste trabalho respondem a muitas das questões apresentadas em trabalhos relacionados a nova geração de SIGs. A estrutura apresentada permite a resolução da interoperabilidade semântica, uma das características dos modernos sistemas de informação.

## 1 Introdução

Hoje em dia existe uma enorme quantidade de dados relativos ao planeta Terra. Estes dados vêm, não só de novos sistemas de informação geográficos (SIG), mas também de satélites novos e cada vez mais sofisticados. Ao mesmo tempo, a expansão da rede mundial de computadores e o aparecimento de novos domínios de aplicações introduziram mudanças importantes no desenvolvimento de sistemas de computação. Os sistemas modernos são distribuídos e altamente heterogêneos. Isto tudo leva a novos desafios com relação à pesquisa em SIG. Um deles é a integração de diferentes tipos de informação, não apenas com relação ao conteúdo, mas também com relação a sua própria natureza como é o caso de

informações geográficas e informações alfanuméricas em geral.

Interoperabilidade é capacidade que um sistema possui de compartilhar e trocar informações e aplicações (Bishr 1997). De acordo com Sheth (1999), a nova geração de sistemas de informação deverá ser capaz de resolver a interoperabilidade semântica, na qual um fato pode ter mais que uma descrição, para poder fazer um bom uso das informações disponíveis com a chegada da Internet e da computação distribuída. Estes sistemas deverão ser capazes de entender o modelo que o usuário faz do mundo e seus significados e entender também os modelos por trás das fontes de informação. Sheth (1999) acredita que o uso de mediadores (Wiederhold 1998) é a solução para ligação entre o usuário e estas fontes de informação.

Ontologia é uma antiga disciplina que vem desde o estudo feito por Aristóteles sobre as categorias e a metafísica, é a ciência que estuda o ser e suas propriedades. Para a comunidade de Inteligência Artificial, ontologias são teorias que especificam um vocabulário relativo a um certo domínio. Este vocabulário define entidades, classes, propriedades, predicados e funções e as relações entre estes componentes. Uma análise detalhada do uso de ontologias como suporte à interoperabilidade pode ser encontrada em Wiederhold & Jannink (1999) e Mena *et al.* (1998). Ontologias têm um papel chave na resolução da interoperabilidade semântica entre SIGs (Wiederhold 1994). Sheth (1999) acrescenta que além de ontologias, o uso do contexto também é fundamental nesta solução. Ele afirma também que a pesquisa sobre a nova geração de sistemas de informação deve se concentrar, a princípio, em domínios específicos, como os SIGs, e depois então desenvolver arquiteturas mais gerais.

A pesquisa sobre a nova geração de SIGs apesar de frequentemente estar focada em interoperabilidade (Câmara 1995; Kottman 1999; Sondheim *et al.* 1999), tem também observado outras áreas. Couclelis (1992) e Egenhofer & Mark (1995) argumentam em favor de um novo SIG que supere as limitações da geometria euclidiana. Pissinou *et al.* (1993) acreditam, que entre os rumos que a nova geração de SIGs deve tomar, estão o uso de orientação a objetos e Inteligência Artificial. Worboys (1995) descreve áreas em que a ciência da computação tem uma participação fundamental na especificação de uma nova geração de sistemas de informação e considera que SIGs com capacidade de lidar com objetos em 3 dimensões e SIGs baseados em sistemas de conhecimento fazem parte desta geração. Egenhofer & Mark (1995) introduzem a “Geografia do Cotidiano”, uma disciplina que estuda a maneira como as pessoas raciocinam sobre o espaço geográfico e o tempo, considerando que os futuros SIGs terão como base esta geografia.

Este trabalho aborda a questão da integração de informações geográficas e apresenta uma proposta para uma arquitetura de sistemas de informação geográficos baseados em ontologias. A arquitetura é baseada no uso de ontologias representadas formalmente e na sua tradução para componentes de software. Esta tradução é feita através do uso de técnicas de orientação a objetos. O resultado deste trabalho responde diversas questões apresentadas em trabalhos relacionados à nova geração de SIGs. A estrutura apresentada permite a resolução da interoperabilidade semântica, uma das características dos modernos sistemas de informação.

O restante deste artigo é organizado da seguinte maneira: a seção 2 faz uma revisão do uso de ontologias em sistemas de informação. A seção 3 apresenta um SIG baseado em ontologias. A seção 4 mostra os tipos de classes disponíveis no sistema e como é feita a navegação através da hierarquia de classes. A seção 5 apresenta conclusões e direções futuras deste trabalho.

## 2 Ontologias e Sistemas de Informação

Os motivos para se usar ontologias em sistemas de informação geográficos são discutidos em Smith & Mark (1998) e Frank (1997). Smith & Mark (1998) consideram que uma ontologia do mundo geográfico pode:

- ajudar a entender como diferentes comunidades compartilham informações;
- ajudar a descobrir certas distorções presentes nos processos cognitivos de apreensão do mundo geográfico;
- fornecer padrões para o desenvolvimento de SIGs.

Além disto, o uso de ontologias no desenvolvimento de SIGs permite estabelecer correspondências e relações entre os diferentes domínios de entidades espaciais.

Frank (1997) acredita que o uso de ontologias no desenvolvimento de sistemas contribui para uma melhor qualidade do produto final, já que elas podem ajudar a evitar problemas como (1) inconsistências entre ontologias implícitas em SIGs, (2) conflitos entre conceitos ontológicos e implementação e (3) conflitos entre ontologia do senso comum e conceitos matemáticos embutidos no software.

Kuhn (1993) acredita que ontologias podem levar os SIGs a superar a metáfora de mapas, que apresenta o mundo como camadas superpostas e independentes. Os problemas do uso da metáfora de mapas na construção de SIGs foram apontados em Nunes (1991) e Frank (1984).

### 2.1 Sistemas de Informação baseados em Ontologias

Filósofos e engenheiros de software têm pontos de vista diferentes sobre ontologias. Guarino (1998) apresenta a diferença entre ontologia no sentido filosófico e na maneira como o termo é usado pela comunidade de Inteligência Artificial. Para a engenharia de software, ontologia descreve uma certa realidade com um vocabulário específico, usando um conjunto de premissas de acordo com o sentido intencional das palavras do vocabulário. No sentido filosófico, ontologia é um sistema específico de categorias que reflete uma visão específica do mundo. Gruber (1992) define uma ontologia como uma especificação explícita de uma conceituação. Guarino (1998) revê e amplia esta definição: uma ontologia é uma

teoria lógica que corresponde ao significado intencional de um vocabulário formal, ou seja, um comprometimento ontológico com uma conceituação específica do mundo. Os modelos intencionais de uma linguagem lógica usando este vocabulário são controlados por seu comprometimento ontológico. Este comprometimento e a conceituação subentendida são refletidas na ontologia pela aproximação desses modelos intencionais.

Desta forma, podemos dizer que para o engenheiro de software existem diversas ontologias, enquanto que para o filósofo existe apenas uma, a Ontologia, com letra maiúscula. Para resolver esta questão, Smith (1998) sugere uma distinção terminológica entre uma ontologia baseada na realidade (ontologia-R) e uma ontologia epistemológica (ontologia-E). Ontologia-R é uma teoria que explica como o universo é organizado e corresponde ao mundo dos filósofos. Uma ontologia-E serve aos propósitos dos engenheiros de software e pode ser definida como uma teoria que explica como um indivíduo, grupo, linguagem ou ciência entende um determinado domínio.

## 2.2 Classificação de Ontologias

Guarino (1997) classifica as ontologias de acordo com sua dependência em relação a uma tarefa específica ou a um ponto de vista:

- *Ontologias de Alto Nível*: descrevem conceitos bem gerais. Em relação aos SIGs, estas ontologias descreveriam os conceitos básicos do espaço. Uma teoria descrevendo partes e todos, e suas relação com a topologia, chamada mereologia (Smith 1995), estaria neste nível;
- *Ontologias de Domínio*: descrevem um vocabulário relacionado a um domínio genérico. No caso de SIGs, poderia ser, por exemplo, uma descrição dos fatos e entidades relacionados a sensoriamento remoto ou ao ambiente urbano;
- *Ontologias de Tarefas*: descrevem uma tarefa ou uma atividade, como interpretação de imagens ou avaliação de poluição sonora em ambientes urbanos;
- *Ontologias de Aplicação*: descrevem conceitos que dependem tanto de um domínio específico como de uma tarefa específica, e geralmente são uma especialização de ambos. Em nossa proposta, estas ontologias são criadas a partir da combinação de ontologias de níveis superiores. Elas representam as necessidades dos usuários com relação a uma aplicação específica, como, por exemplo, a avaliação da disponibilidade de camarões na costa do Espírito Santo.

Outra distinção que Guarino (1998) faz é entre ontologias refinadas e não-refinadas, ou ainda, *off-line* e

*on-line*. Uma ontologia não-refinada tem um número mínimo de axiomas e seu objetivo é ser compartilhada por usuários que concordem sobre uma determinada visão de mundo. Uma ontologia refinada precisa de uma linguagem de alta expressividade e tem um grande número de axiomas. Ontologias não-refinadas têm mais chance de ser compartilhadas e deveriam ser usadas *on-line* para dar suporte à funcionalidade de sistemas de informação. Já as ontologias refinadas deveriam ser usadas *off-line* e somente para referência. A solução proposta aqui apresenta um modelo de navegação embutido nas classes básicas do sistema. Este modelo permite a transformação de objetos de uma classe para outra, sejam de classes pertencentes a ontologias refinadas para não-refinadas ou vice-versa. Este sistema de navegação esta descrito em Fonseca (1999) e permite que a distância entre as duas soluções sugeridas em Guarino (1998) seja diminuída.

## 3 SIGs baseados em Ontologias

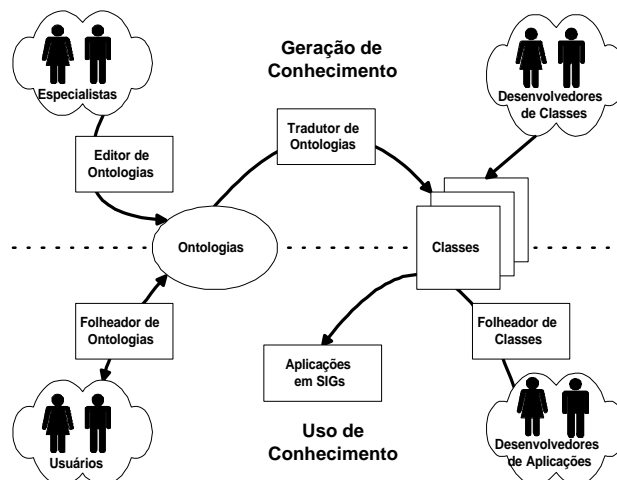
O uso de ontologias explícitas no desenvolvimento e uso de sistemas de informação leva ao que são chamados de Sistemas de Informação baseados em ontologias (Guarino 1998). Em nossa proposta de um SIG baseado em ontologias, as ontologias são representadas em uma estrutura formal baseada em Rodriguez (2000), que descreve ontologias através seus componentes: partes, funções e atributos. Esta estrutura considera mais a robustez da implementação (Guarino *et al.* 1999), já que não é direcionada a ser usada em sistemas dedução automática e sim para extração e manipulação de informações. Em Cranefield (1999) pode ser encontrada uma discussão completa sobre as formas de armazenamento de ontologias de acordo com o uso pretendido. As ontologias são organizadas em uma estrutura hierárquica, já que está é considerada uma das melhores formas de se representar o mundo geográfico (Smith 1995; Harvey *et al.* 1999). O mapeamento das entidades presentes nas ontologias para componentes de software é feito usando-se técnicas de orientação a objeto e herança múltipla. Este tipo de mapeamento permite integração parcial de informações quando uma integração completa não é possível. Este sistema permite ainda o reaproveitamento de classes já desenvolvidas, incorporando, assim, nos novos sistemas, o conhecimento existente em outros SIGs.

Um SIG baseado em ontologias é composto por um editor de ontologias, por um servidor de ontologias, por ontologias especificadas formalmente e por classes derivadas de ontologias. A especificação de ontologias é feita através do uso de um editor de ontologias. Um sistema para editar, folhear, traduzir e reutilizar ontologias é o Ontolíngua (Gruber 1992). O sistema Ontolíngua

permite que a edição de ontologias seja feita por grupos de especialistas trabalhando simultaneamente. As ontologias podem ser traduzidas para linguagens como CORBA Interface Definition Language (IDL) (OMG 1991), Prolog (Clocksin and Mellish 1981), Epikit e KIF (Genesereth 1990). Em nossa proposta será usado um novo editor de ontologias, já que optamos por traduzir e implementar as classes derivadas das ontologias em Java, opção que não está disponível no sistema Ontolíngua. O editor a ser implementado também será capaz de ler e escrever ontologias em XOL (Karp *et al.* 1999), uma linguagem que foi criada para facilitar o compartilhamento de ontologias e é baseada em *Extensible Markup Language* (XML) (Graham and Quin 1999).

Já que está sendo utilizada uma abordagem orientada a objetos na especificação das ontologias, é possível traduzir as classes para uma linguagem de computação orientada a objetos. Nós optamos por fazer a tradução para interfaces em Java. Uma interface em Java descreve um conjunto de métodos públicos que deverão ser implementados por classes em Java. Então, uma interface não é uma implementação. O mecanismo para implementação de herança múltipla em Java através do uso de Interfaces para um SIG baseado em ontologias está descrito em Fonseca *et al.* (2000b).

O resultado do processo inicial de uso de um SIG baseado em ontologias é um conjunto de ontologias especificadas formalmente e um conjunto de classes. Podemos chamar este processo inicial de geração de conhecimento (Figura 1). As ontologias são administradas por um servidor de ontologias. Este servidor permite o folheamento de ontologias, assim, colocando à disposição de usuários, os metadados referentes às informações disponíveis. O servidor de ontologias também contém apontadores para sistemas de informação geográficos. A comunicação entre o servidor de ontologias e os SIGs é feita através de mediadores. Estes mediadores são responsáveis por extrair as informações dos SIGs e criar as instâncias das classes. Estas classes são um ponto chave em nossa proposta. Elas vão conter as informações extraídas dos SIGs e o conhecimento extraído das ontologias.



**Figura 1** Esquema de um SIG baseado em Ontologias.

#### 4 Classes e Navegação

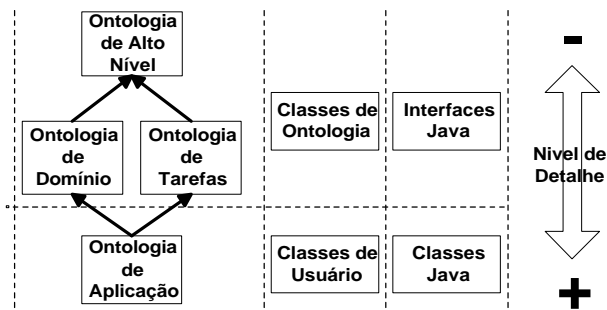
Sistemas de informação geográficos baseados em ontologias são criados através do uso de classes extraídas de diversas ontologias. Estas classes são usadas como base para o desenvolvimento de novas aplicações. Desta forma, o conhecimento acumulado por especialistas pode ser reutilizado por diversos usuários. Considere, por exemplo, o caso de um usuário que precisa modelar uma classe para lidar com um objeto geográfico que tenha limites imprecisos. Este é um problema difícil, mas que é encontrado em diversas aplicações de SIGs. Uma ontologia que tenha estabelecido os tipos de limites existentes em objetos geográficos, e como eles devem ser tratados em termos de operações e características, pode ser usada neste caso. O usuário pode reaproveitar o conhecimento de especialistas e implementar seu objeto geográfico baseado nesta classe. Na seção 4.1 mostramos os dois tipos de classe presentes num ambiente de sistemas de informação geográficos baseados em ontologias. Na seção 4.2 introduzimos o modelo de navegação proposto.

##### 4.1 Classes de Usuário e Classes de Ontologias

O desenvolvimento de aplicações usando-se classes derivadas de ontologias é feito criando-se o que chamamos de *Classes de Usuário* em oposição a *Classes de Ontologias* que são as classes logo após a tradução. *Classes de Usuário* são criadas através da combinação de *Classes de Ontologias* através do uso de herança múltipla. *Classes de Usuário* são muito mais específicas do que as *Classes de Ontologia* que são mais genéricas. *Classes de Usuário* pertencem ao nível *Ontologias de Aplicação* enquanto que *Classes de Ontologia* pertencem ao nível de *Ontologias de Alto Nível, Ontologias de Domínio* e



*Ontologias de Tarefas* (Guarino 1998). Na arquitetura proposta existem diferentes níveis de ontologias. O nível de detalhe da informação extraída é correspondente aos níveis das ontologias. *Ontologias de Alto Nível* são mais gerais e têm menos detalhes enquanto o contrário ocorre nas demais ontologias. As ontologias dos níveis mais baixos são criadas a partir de refinamentos das definições das ontologias dos níveis superiores. Assim cada novo nível de ontologias incorpora o conhecimento do nível superior (Figura 2).



**Figura 2** Níveis de Ontologias e Classes, estendido a partir de Guarino (1998).

#### 4.2 Navegação

Em geral, as *Classes de Usuário* são derivadas de mais de uma entidade das ontologias. Então é necessário ter um mecanismo para implementar herança múltipla. Nós escolhemos implementar as classes usando vários papéis. Cada classe pai corresponde a um papel que o objeto pode representar. O modelo para implementar objetos que representam diversos papéis foi sugerido em Pernici (1990). Por exemplo, uma instância da classe lote pode ser vista como um lote ou um polígono. Embora uma instância de uma classe possa ser vista através de múltiplas interfaces, múltiplos papéis, ela nunca perde sua identidade. A identidade do objeto é preservada e o que é colocado à disposição do usuário são suas diversas interfaces.

Nós propomos aqui dois tipos de navegação: generalização/especialização e extração de papéis. Para implementar a generalização, um objeto tem de ter a capacidade de gerar a partir de si próprio uma instância imediatamente superior deste objeto. Para implementar a especialização, um objeto tem de ser capaz de gerar uma instância de si próprio a partir de uma instância da classe imediatamente superior. Estes dois métodos têm como principal objetivo: (1) a reutilização de conhecimento e (2) aplicações de operações que de outra maneira não estariam disponíveis para certas classes. Estas operações também permitem a integração de informações. Isto é feito através da navegação até pontos em comum na árvore de de

hierarquia.

A extração de papéis é feita para se aplicar operações que estão disponíveis para certas classes. Ela também é usada para a geração de visões. Por exemplo, quando o usuário quer obter uma visão de rede de uma cidade, todos os objetos que têm as entidades nó ou trecho de rede, como um de seus papéis, fornecem estes papéis para serem exibidos e manipulados.

Um objeto pode ter de se adaptar a várias visões e relacionamentos através de mudanças de classes. Por exemplo, um objeto poligonal que tem de se fundir com um outro objeto que pertence a uma de suas superclasses, tem que se adaptar fazendo o papel desta superclasse. Assim a operação pode ser efetuada usando-se a classe superior comum a ambos os objetos. Informações nunca são perdidas em conversões porque elas ocorrem apenas nas representações dos objetos e não em sua forma original. A possibilidade de se usar operações de navegação em uma árvore hierárquica de ontologias como guia de operações de generalização de dados geográficos foi sugerida em Fonseca (2000a).

Os objetos geográficos têm de implementar, ou então herdar, todos os métodos das classes das quais eles são derivados. Já que estamos propondo uma arquitetura de objetos distribuídos, a implementação dos objetos pode se basear no uso de serviços externos para a execução de seus métodos. Por exemplo, quando da implementação de um objeto setor censitário, um programador tem de escrever um método para o cálculo de população de uma área que cobre diversos setores censitários. Em vez de desenvolver este método, o programador pode chamar um serviço que tenha uma implementação sofisticada deste cálculo. Do ponto de vista do usuário, o método é executado localmente, mas na realidade ele é processado usando métodos e serviços localizados em outro servidor.

Assim como em CORBA (OMG 1991) e Java (Gosling and McGilton 1995), onde todas as classes são derivadas de uma classe básica, nós propomos aqui que todas as classes sejam derivadas de uma classe única chamada Objeto. Esta classe tem dois métodos que são fundamentais para o sistema, os métodos `para cima()` e `a partir de()`. O método `para cima()`, quando aplicado a um determinado objeto, retorna um objeto da classe que deu origem a este objeto. O método `a partir de()` gera uma instância da própria classe a partir de uma instância de uma classe superior. Estes dois métodos proporcionam a navegação por toda a árvore de ontologias.

Consideremos o seguinte exemplo em que duas classes são derivadas da Classe `Setor_Censitário`: `Setor_Censitário_A` e `Setor_Censitário_B`

(Figura 3). O usuário quer usar a operação cálculo de população em Setor\_Censitário\_A. Mas a operação só está disponível em Setor\_Censitário\_B. Então é necessário que: (1) Setor\_Censitário\_A gere uma instância de Setor\_Censitário usando para cima(); (2) Setor\_Censitário\_B gere uma instância de si mesmo a partir da instância de Setor\_Censitário gerada na operação anterior; (3) a operação é aplicada na nova instância de Setor\_Censitário\_B e o resultado estará disponível para o usuário em uma instância de Setor\_Censitário\_B. O processo inverso poderia ser aplicado para se gerar uma instância de Setor\_Censitário\_A.

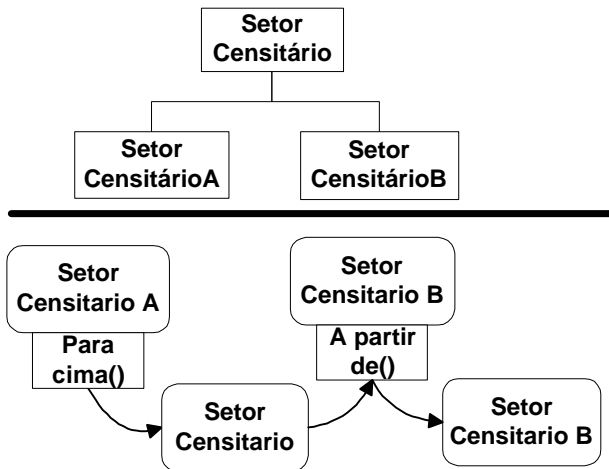


Figura 3: Navegação

## 5 Conclusões

Este trabalho apresentou uma proposta de um sistema de informação geográfico baseado em ontologias. Estas ontologias são especificadas formalmente e podem ser traduzidas em componentes de software. Estes componentes podem ser usados no desenvolvimento de aplicações geográficas e também na troca de informações. Foi proposta uma estrutura hierárquica para representação das ontologias. A tradução é feita através do uso de técnicas de orientação a objeto. A estrutura apresentada funciona como integradora de diversos sistemas de informação geográficos.

### 5.1 Resultados Principais

Devido a complexidade do mundo geográfico temos que conviver com diversas visões de mundo (Frank 1997). Assim, é preciso que os desenvolvedores de SIGs tenham ferramentas que permitam a combinação de várias ontologias. A solução apresentada aqui permitiu a formação de classes complexas oriundas de diversas

ontologias. Esta abordagem também possibilitou a integração de informações parciais, quando a integração total não foi possível. A interoperabilidade semântica foi resolvida através do uso de classes derivadas de ontologias. Toda a manipulação de informações foi feita baseada nas definições das entidades geográficas presentes nas ontologias.

O uso de orientação a objetos na tradução de ontologias em classes levou a uma estrutura hierárquica representando o mundo geográfico. O sistema de navegação fez uso desta estrutura e permitiu a geração de novas instâncias classes mais ou menos detalhadas. Estas instâncias permitiram o compartilhamento de informações geográficas e também a reutilização de conhecimento. O compartilhamento pôde ser parcial quando não foi possível um compartilhamento integral.

A possibilidade de se ter diversas visões do mesmo objeto geográfico através do destaque dos papéis que um objeto pode desempenhar atendeu a alguns dos requisitos básicos de um usuário de um sistema de informação geográfico (Laurini and Thompson 1992).

### 5.2 Direções Futuras

A estrutura comercial atual da Internet é bem diferente da do passado. Um dos modelos que está se estabelecendo é a oferta de serviços básicos gratuitos ao lado da cobrança por serviços mais sofisticados. Este interesse em captar novos consumidores pode ser o passo inicial para sistemas baseados em ontologias. Ontologia mais genéricas podem ser oferecidas gratuitamente enquanto ontologias mais detalhadas seriam cobradas. Outro papel que os provedores de serviços podem ter é o de integradores de informação. Um dos pontos mais importantes em sistemas de informação geográficos baseados em ontologias é a geração das instâncias das classes. É preciso conhecer a fonte das informações e, ao mesmo tempo, é preciso ter conhecimento das ontologias. Esta ligação é importante e delicada. Provedores de informação podem fazer este serviço para ter o direito de comercializar grandes bases de dados. Por exemplo, importantes bases de dados pertencentes a governos municipais poderiam estar disponíveis ao público em geral, e não estão apenas por falta de uma estrutura eficiente de acesso. Desta forma, uma das futuras direções desta pesquisa é analisar o papel destes provedores de serviços e informações como portais de entrada dos usuários do tipo de sistema apresentado aqui.

## 6 Agradecimentos

Este trabalho de Frederico Fonseca foi desenvolvido com financiamento de National Science Foundation, projetos

NSF SBR-9700465 e IIS-9970123, uma bolsa da NASA/EPSCoR número 99-58 e uma bolsa da ESRI. Além disto, a pesquisa de Max Egenhofer é financiada pelos projetos NSF IRI-9613646, BDI-9723873, e EIA-9876707; por National Imagery and Mapping Agency projeto número NMA202-97-1-1023; por National Institute of Environmental Health Sciences, NIH, projeto número 1 R 01 ES09816-01 e por um contrato com Lockheed Martin.

## Referências

Bishr, Y. (1997) *Semantic Aspect of Interoperable GIS*. Ph.D. Thesis, Wageningen Agricultural University, The Netherlands.

Câmara, G. (1995) *Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos*. Ph.D. Thesis, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP.

Clocksin, W. and Mellish, C. (1981) *Programming in Prolog*. Springer-Verlag, New York.

Couclelis, H. (1992) People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS. in: A. U. Frank, I. Campari, and U. Formentini, (Eds.), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science* 639, pp. 65-77, Springer-Verlag, Pisa.

Cranefield, S. and Purvis, M. (1999) UML as an Ontology Modelling Language. in: D. Fensel, (Ed.) *Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence*, Stockholm.

Egenhofer, M. and Mark, D. (1995) Naive Geography. in: A. Frank and W. Kuhn, (Eds.), *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '95, Semmering, Austria. Lecture Notes in Computer Science* 988, pp. 1-15, Springer-Verlag, Berlin.

Fonseca, F. and Egenhofer, M. (1999) Ontology-Driven Geographic Information Systems. in: C. B. Medeiros, (Ed.) *7th ACM Symposium on Advances in Geographic Information Systems*, Kansas City, MO, pp. 14-19.

Fonseca, F., Egenhofer, M. and Davis, C. (2000a) Ontology-Driven Information Integration. in: C. Bettini and A. Montanari, (Eds.), *The AAAI-2000 Workshop on Spatial and Temporal Granularity*, Austin, TX.

Fonseca, F., Egenhofer, M., Davis, C., and Borges, K. (2000b) Ontologies and Knowledge Sharing in Urban GIS. *CEUS - Computer, Environment and Urban Systems* 24(3): 232-251.

Frank, A. (1984) Computer Assisted Cartography—Graphics or Geometry. in: *Journal of Surveying Engineering*. 110, pp. 159-168,

Frank, A. (1997) Spatial Ontology: A Geographical Point of View. in: O. Stock, (Ed.) *Spatial and Temporal Reasoning*. pp. 135-153, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Genesereth, M. R. (1990) *The Epikit Manual*. Epistemics, Inc., Palo Alto, CA, Technical Report.

Gosling, J. and McGilton, H. (1995) *The Java Language Environment: a White Paper*. Sun Microsystems, Mountain View, CA, Technical Report.

Graham, I. and Quin, L. (1999) *XML Specification Guide*. John Wiley & Sons, New York.

Gruber, T. (1992) *A Translation Approach to Portable Ontology Specifications*. Knowledge Systems Laboratory - Stanford University, Stanford, CA, Technical Report KSL 92-71.

Guarino, N. (1997) Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. in: M. Pazienza, (Ed.) *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97*, Frascati, Italy, pp. 139-170.

Guarino, N. (1998) Formal Ontology and Information Systems. in: N. Guarino, (Ed.) *Formal Ontology in Information Systems*. pp. 3-15, IOS Press, Amsterdam, Netherlands.

Guarino, N., Masolo, C. and Vetere, G. (1999) OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems* 14(3): 70-80.

Harvey, F., Battenfield, B. and Lambert, S. (1999) Integrating Geodata Infrastructures from the Ground Up. *Photogrammetric Engineering & Remote Sensing* 65(11): 1287-1291.

Karp, P., Chauhdri, V. and Thomere, J. (1999) *XOL: an XML-based Ontology Exchange Language*. Artificial Intelligence Center SRI International, Menlo Park, CA, Technical Report.

Kottman, C. (1999) Introduction: System and Experiences. in: M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, (Eds.), *Interoperating Geographic Information Systems*. pp. 215-219, Kluwer Academic Publishers, Norwell, MA.

Kuhn, W. (1993) Metaphors Create Theories for Users. in: A. Frank and I. Campari, (Eds.), *Spatial Information Theory. Lecture Notes in Computer Science* 716, pp. 366-376, Springer-Verlag, Berlin.

Laurini, R. and Thompson, D. (1992) *Fundamentals of Spatial Information Systems*. Academic Press, London.

- Mena, E., Kashyap, V., Illarramendi, A., and Sheth, A. (1998) Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. in: N. Guarino, (Ed.) *Formal Ontology in Information Systems*. pp. 269-283, IOS Press, Amsterdam.
- Nunes, J. (1991) Geographic Space as a Set of Concrete Geographical Entities. in: D. Mark and A. Frank, (Eds.), *Cognitive and Linguistic Aspects of Geographic Space*. pp. 9-33, Kluwer Academic Publishers, Norwell, MA.
- OMG, Ed. (1991) *The Common Object Request Broker: Architecture and Specification, Revision 1.1*. OMG Document No. 91.12.1 Framingham, MA.
- Pernici, B. (1990) Objects with Roles. in: *IEEE/ACM Conference on Office Information Systems*, , pp. 205-215.
- Pissinou, N., Makki, K. and Park, E. (1993) Towards the Design and Development of a New Architecture for Geographic Information Systems. in: B. K. Bhargava, T. W. Finin, and Y. Yesha, (Eds.), *Second International Conference on Information and Knowledge Management*, Washington, DC, USA, pp. 565-573.
- Rodriguez, A. (2000) *Assessing Semantic Similarity among Spatial Entity Classes*. University of Maine, Orono.
- Sheth, A. (1999) Changing Focus on Interoperability in Information Systems: from System, Syntax, structure to Semantics. in: M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, (Eds.), *Interoperating Geographic Information Systems*. pp. 5-29, Kluwer Academic Publishers, Norwell, MA.
- Smith, B. (1995) On Drawing Lines on a Map. in: A. Frank and W. Kuhn, (Eds.), *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '95, Semmering, Austria. Lecture Notes in Computer Science 988*, pp. 475-484, Springer Verlag, Berlin.
- Smith, B. (1998) An Introduction to Ontology. in: D. Peuquet, B. Smith, and B. Brogaard, (Eds.), *The Ontology of Fields*. pp. 10-14, NCGIA, Bar Harbor, ME.
- Smith, B. and Mark, D. (1998) Ontology and Geographic Kinds. in: *International Symposium on Spatial Data Handling*, Vancouver, Canada, pp. 308-320.
- Sondheim, M., Gardels, K. and Buehler, K. (1999) GIS Interoperability. in: P. Longley, M. Goodchild, D. Maguire, and D. Rhind, (Eds.), *Geographical Information Systems. 1 Principles and Technical Issues*, John Wiley & Sons, INC, New York.
- Wiederhold, G. (1994) Interoperation, Mediation and Ontologies. in: *International Symposium on Fifth Generation Computer Systems (FGCS94)*, Tokyo, Japan, pp. 33-48.
- Wiederhold, G. (1998) *Value-added Middleware: Mediators*. Stanford University, Technical Report.
- Wiederhold, G. and Jannink, J. (1999) Composing Diverse Ontologies. in: *8th Working Conference on Database Semantics (DS-8)*, Rotorua, New Zealand.
- Worboys, M. F. (1995) *GIS - A Computing Perspective*. Taylor & Francis Inc., Bristol, PA.

# **Trading e Agentes em Sistemas de Informações Geográficas Distribuídos**

SÔNIA VIRGÍNIA ALVES FRANÇA<sup>1</sup>

CARLOS ANDRÉ GUIMARÃES FERRAZ<sup>2</sup>

<sup>1</sup> UNIT-Universidade Tiradentes - Av. Dr. José T. D. Nabuco, 300, 49030-270, Aracaju, SE  
soniafranca@yahoo.com

<sup>2</sup> UFPE/Centro de Informática - Caixa Postal 7851, 50732-970, Recife, PE  
cagf@cin.ufpe.br

**Resumo:** Este trabalho descreve um modelo que possibilita que Sistemas de Informações Geográficas heterogêneos, distribuídos em redes de longas distâncias se comuniquem, interoperando dados e serviços. O modelo baseia-se nas tecnologias de objetos distribuídos e agentes e no padrão OGIS, definindo os conceitos de GIS *Trader* e Geoagentes como componentes que tornam possível a interoperabilidade entre aplicações de geoprocessamento, bem como contribuem para um bom desempenho quando essas aplicações se encontram distribuídas em redes heterogêneas e de longa distância.

**Abstract:** This paper describes a model that allows heterogeneous Geographic Information Systems, distributed over wide area networks to communicate with each other, interoperating data and services. The model is based on distributed objects and agents technologies and on the OGIS standard, defining the concepts of GIS *Trader* and Geoagents as components that enable interoperability between geoprocessing applications, as well as contribute to good performance when such applications are distributed over heterogeneous and wide area networks.

## **1 Introdução**

Os Sistemas de Informações Geográficas ou SIGs, foram adotados extensamente nas duas últimas décadas para auxiliar no planejamento urbano, agricultura, transporte e muitos outros campos [5]. No entanto, A maioria dos SIGs são implementados para executar em ambientes fechados, possuindo seu próprio modelo de dados e funções para manipular os mesmos. A troca de dados entre SIGs heterogêneos é bastante complexa ou praticamente não existe [2, 9, 14]. Para resolver esse problema são adotados padrões, diminuindo custos e tempo de aquisição de dados geográficos. Um dos padrões para troca de dados geográficos mais difundido atualmente é o OGIS (*Open Geographic Interoperability Specification*) [3, 8].

Os SIGs são desenvolvidos, muitas vezes, para uma certa comunidade de usuários, assim eles são equipados com um conjunto de funções de interesse dessa comunidade [4]. No entanto, pode ser que o usuário necessite de uma função que o seu SIG não dispõe, assim, seria interessante que ele pudesse fazer uso de funções que estivessem disponíveis em outros SIGs.

Quando esses SIGs se encontram distribuídos em redes heterogêneas e de larga escala, serão necessários mecanismos que tornem possível a comunicação entre eles de forma eficaz e eficiente. É importante ressaltar que ambientes de geoprocessamento envolvem grande volumes de dados e um grande problema em ambientes de

larga escala é saber onde encontrar esses dados ou a função desejada.

O modelo proposto neste trabalho permite que sistemas de informações geográficas distribuídos em redes heterogêneas e de longas distâncias se comuniquem, interoperando dados e serviços, e fornecendo aos usuários desses SIGs, transparência de acesso e de localização, facilidades para buscar a informação através de interfaces bem definidas, simples e claras, bem como tempos de resposta razoáveis, considerando longa distância e dados geográficos de grande volume.

## **2 Modelo Baseado em GIS *Trader* e Agentes Geográficos**

O modelo reúne as tecnologias de objetos distribuídos [13], o padrão OGIS e agentes [7, 10, 12] com a finalidade de proporcionar a interoperabilidade entre aplicações de geoprocessamento, bem como garantir um bom desempenho quando essas aplicações se encontram distribuídas em redes heterogêneas e de longa distância.

O modelo funciona da seguinte forma: servidores cadastram seus serviços de geoprocessamento em um GIS *trader*. O GIS *trader* é responsável por fazer o anúncio desses serviços. O usuário, através da aplicação SIG, faz a requisição de um serviço que a sua aplicação não possui. A busca e invocação desse serviço é feita pelos

geoagentes que se comunicam com o GIS *trader* e com o servidor, obtendo assim o resultado para o usuário[6].

Uma visão macro do modelo é apresentada na figura 1, com os seguintes componentes: servidor, GIS *trader*, geoagentes e aplicação GIS.

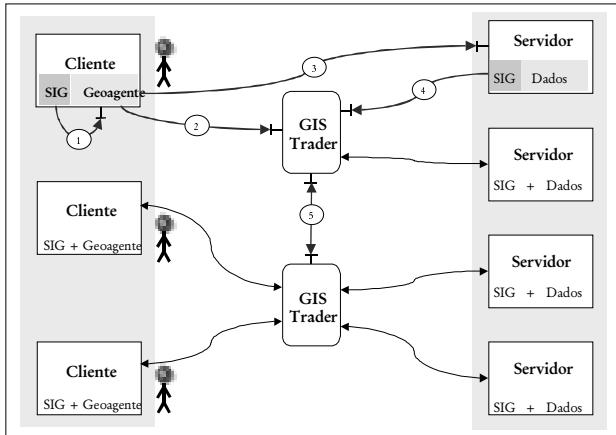


Figura 1: Visão do modelo baseado em GIS *trader* e agentes geográficos.

A comunicação entre os componentes do modelo é feita via interfaces. As setas numeradas (figura 1) mostram as diferentes interfaces necessárias para a comunicação entre os componentes. A interface 1 representa a comunicação da aplicação SIG com o geoagente; essa interface é necessária quando a aplicação deseja fazer a requisição de um serviço. A interface 2 representa a comunicação do geoagente com o GIS *trader*; o geoagente é responsável por ir buscar o serviço requisitado pela aplicação. A interface 3 representa a comunicação do geoagente com o servidor; neste momento o geoagente está fazendo a requisição do serviço. A interface 4 representa a comunicação do servidor com o GIS *trader*, usada no momento em que o servidor deseja fazer a oferta de um serviço de geoprocessamento. A interface 5 representa a comunicação entre os GIS *traders*, esta ocorre caso exista a necessidade de uma federação de *traders* [1, 17] em um sistema de larga escala.

Os componentes do modelo são objetos CORBA [13] que estão distribuídos nas redes, comunicando-se através do protocolo IIOP e sendo gerenciados por ORBs. A escolha por CORBA se dá pela independência de plataforma e de linguagem, e sua habilidade de tratar problemas de heterogeneidade. Essas facilidades trazem ganhos muito grandes, já que está se tratando de ambientes de geoprocessamento que são naturalmente heterogêneos. Outra vantagem é a facilidade de escalabilidade do sistema, ou seja, novos SIGs podem ser adicionados ao modelo a qualquer momento. O serviço *trader* [1, 13] oferecido pelo CORBA é adaptado para atender às necessidades dos usuários de SIGs, surgindo assim o conceito de GIS *trader*.

A eficiência das buscas em redes de longa distância pode se dar através dos geoagentes (ou agentes geográficos, agentes que fazem tarefas voltadas ao geoprocessamento) e do GIS *trader*. Juntos, irão localizar, buscar e invocar os serviços requisitados pelos usuários com mais rapidez, visto que no GIS *trader* são catalogados todos os serviços de geoprocessamento disponíveis para uma certa comunidade de informação. Os geoagentes se comunicam com o GIS *trader* para localizar o serviço requisitado. Se não houvesse o GIS *trader*, os geoagentes teriam que buscar o serviço requisitado entre todos os serviços disponíveis na rede.

Nos servidores encontram-se implementações de objetos CORBA. Quando um novo objeto, que faz um serviço de geoprocessamento, é instanciado, ele pode ser cadastrado no GIS *trader*, para que este faça o anúncio e, assim as aplicações podem fazer uso do serviço. O servidor é requisitado no momento em que o geoagente faz a invocação a seus objetos; ele faz o processamento do pedido e envia o resultado para o geoagente.

Nos clientes residem as aplicações/interfaces SIG e os agentes geográficos. As interfaces SIGs são as aplicações que usam os serviços de geoprocessamento disponíveis na rede. Eles têm o seu próprio modelo de dados e funções para processar esses dados, devem estar de acordo com as especificações do padrão OGIS, tornando possível a interoperabilidade dos dados geográficos. Dessa forma, não haverá problemas no formato dos dados que forem trocados entre os diferentes SIGs. Quando um usuário deseja um serviço que a sua aplicação SIG não dispõe, este interage com uma interface para fazer o pedido do serviço, informando o tipo do serviço desejado e dados de entrada. Também é através dessa interface que o usuário recebe a resposta do seu pedido ou a informação de que o pedido não pode ser atendido. A transparência de localização ao serviço é garantida por CORBA, portanto a aplicação não sabe onde é que o serviço solicitado foi encontrado.

Nos próximos tópicos, será detalhado o funcionamento do GIS *trader* e dos geoagentes, principais elementos do modelo.

## 2.1 GIS *Trader*

*Traders* atuam como um *broker*, possibilitando a clientes encontrar serviços que eles necessitam e anunciam serviços em um ambiente distribuído [1, 11, 13]. Um *trader* interage com os servidores (fornecedores de serviço), os clientes (usuários de serviço) e outros *traders*. O GIS *trader* será responsável por fazer o anúncio de serviços de geoprocessamento, ou seja, ele faz apenas o anúncio de serviços que são do interesse da comunidade de geoprocessamento à qual ele faz parte. Os exportadores (servidores) cadastram seus serviços de geoprocessamento no GIS *trader*, para que este faça o anúncio, e os importadores buscam no GIS *trader* os serviços ofertados.

Se uma entidade desejar usar um serviço oferecido dentro de uma outra comunidade, interações entre *traders* serão necessárias. Estas interações são conhecidas por federações de *traders* [17]. Uma federação de *GIS trader* é criada com o objetivo de aumentar o alcance do usuário, já que este terá acesso a mais serviços.

A figura 2 mostra uma visão detalhada do *GIS trader* com os demais componentes e interfaces. São identificados quatro componentes básicos: importador (geoagente) e exportador de serviços (servidor de geoprocessamento), o *GIS trader* e o administrador da federação.

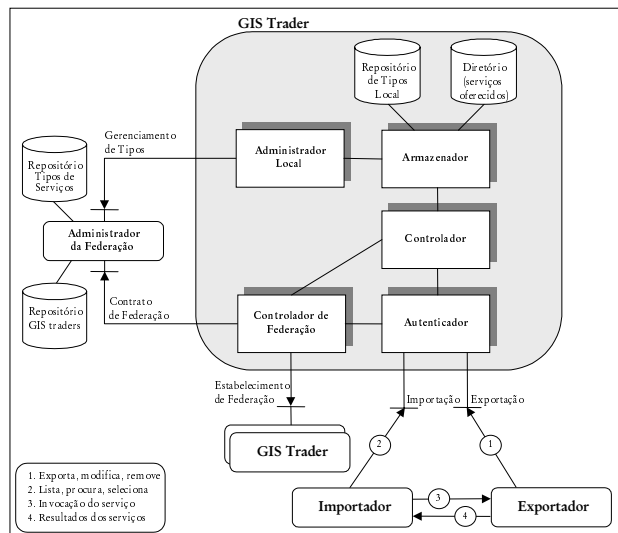


Figura 2: Detalhamento dos módulos do *GIS trader*.

### 2.1.1 Descrição dos Módulos

O *GIS trader* possui cinco módulos que serão descritos a seguir.

O módulo *Armazenador* armazena operações que manipulam a informação correspondente aos tipos de serviços (no *Repositório de Tipos Local*) e serviços oferecidos (no *Diretório*). Por se tratar de um *GIS trader*, os tipos de serviços ofertados são referentes a geoprocessamento, como mapeamento de uma estrada, topografia de uma área, etc. No *Diretório* são armazenados todos os serviços ofertados pelo *GIS trader*.

O módulo *Controlador* é responsável por executar as operações locais do *GIS trader*, tais como: inserir, remover e alterar ofertas de serviços. O módulo *Controlador de Federação* analisa os pedidos que chegam através da interface *Estabelecimento de Federação* considerando o contrato de federação e o envia para o módulo *Controlador*, que irá fazer a busca do pedido federado. O módulo *Controlador de Federação* também é usado para informar com quais *GIS traders* deve ser estabelecida a federação.

O *Autenticador* é o módulo que verifica a permissão do importador ou exportador para executar uma operação.

Se o exportador desejar fazer a oferta de um serviço que não seja do interesse do *GIS trader*, o *Autenticador* informa que o serviço não pode ser cadastrado. Esse é o módulo que traz a diferença entre um *trader* comum e o *GIS trader*. Nos *traders* comuns, podem ser criados e cadastrados qualquer tipo de serviço.

O *Administrador Local* gerencia a informação armazenada no *GIS trader*. É através dele que o *Administrador da Federação* informa qualquer mudança nos tipos de serviços, usando a interface *Gerenciamento de Tipos*.

### 2.1.2 Administrador da Federação

O *Administrador da Federação* é um módulo associado e externo ao *GIS trader* e tem como principais funções: a criação de novos tipos de serviços e o gerenciamento da federação de *GIS traders*. Vale destacar, novamente, que uma federação de *trader* integra grande número de serviços registrados e negociados, aliviando o trabalho de cada *trader* - como se fossem várias páginas amarelas integradas [13].

É no *Administrador da Federação* que são definidos os tipos de serviços de interesse dos usuários da federação. Os tipos de serviços, juntamente com sua descrição, estão armazenados no *Repositório de Tipos de Serviços* e devem ser especificados pelo *Administrador da Federação* para que não ocorram problemas de semântica nos nomes dos tipos de serviços. Ou seja, se cada *GIS trader* criasse os seus próprios tipos de serviços, como é feito normalmente em *traders* comuns, poderia ser que eles usassem nomes bastante diferentes para representar uma funcionalidade semelhante, dificultando que certos serviços fossem encontrados, por exemplo: o tipo de serviço “mapeamento” pode ter nomes bem diferentes em outros *GIS traders* da federação, como por exemplo: “desenho”, “cartograma”. Quando um tipo de serviço é criado, removido ou alterado, o *Administrador da Federação* avisa aos *GIS traders* da federação usando a interface *Gerenciamento de Tipos*.

Quando um *GIS trader* é instanciado, ele deve ser cadastrado no *Administrador da Federação*. As informações do *GIS trader* ficam armazenadas no *Repositório de GIS traders*. O *Administrador da Federação* informa quais os tipos de serviços suportados pela federação e um *GIS trader*, caso exista federação, para que ele se ligue no momento que seja necessário o estabelecimento de federação, usando a interface *Contrato de Federação*. Pode ocorrer que um *GIS trader* tenha problemas e os outros não consigam estabelecer a federação; com isso, o *GIS trader* que está tentando se integrar a federação, contata o *Administrador da Federação* para que este lhe informe um outro *GIS trader* para que a federação seja estabelecida.

### 2.1.3 Atividades de um GIS Trader

O GIS *trader* executa duas atividades principais: o cadastramento e o anúncio de serviços de geoprocessamento [6]. A seguir será detalhada cada uma dessas atividades.

#### Cadastramento de serviços

O cadastramento de um serviço é solicitado pelo exportador ao GIS *trader*. Assim, o exportador negocia com o GIS *trader*, através da interface Exportação. Primeiramente, o GIS *trader* informa os tipos de serviços suportados por ele; se o serviço for de um tipo de serviço existente, então pode ser cadastrado. O serviço é verificado no módulo Autenticador; se estiver tudo correto, isto é, o serviço é de um tipo existente no GIS *trader*, as informações do serviço são passadas para o módulo Execute, que faz o armazenamento no banco de dados de serviços oferecidos. A partir deste momento, o serviço está disponível para os importadores. O exportador também pode solicitar a remoção e atualização de um serviço.

#### Anúncio de Serviços

Quando um importador deseja encontrar um serviço de geoprocessamento, este procura um GIS *trader*, usando a interface Importação. O importador informa ao GIS *trader* o tipo de serviço desejado, o qual verifica se possui cadastrado algum serviço que seja do tipo procurado. Se o GIS *trader* possuir o serviço, ele retorna uma lista contendo todos os serviços encontrados para o importador. A especificação CORBA fornece duas opções de resultados: o *trader* pode retornar uma lista com todos os serviços encontrados ou escolhe um dos serviços desta lista. No GIS *trader* foi adotada a opção de retornar a lista para que a escolha do serviço fique a critério do geoagente. Caso o GIS *trader* não possua o serviço, ele verifica a existência do serviço na federação. O GIS *trader* de origem (ligado ao importador) inicia a negociação com outros GIS *traders* a procura do serviço desejado; para isso é usado o módulo Execute Federação através da interface Estabelecimento de Federação. O pedido vai passando *trader* a *trader*, na tentativa de encontrar o serviço. O *trader* que está sendo visitado no momento conhece a referência do GIS *trader* de origem; assim, quando o serviço for encontrado, o resultado é enviado diretamente para o GIS *trader* de origem, que passa o resultado - referência da interface do objeto prestador do serviço requisitado - para o importador. Se o serviço não for encontrado na federação ou o tempo de consulta for expirado (*timeout*), uma exceção é enviada para o importador.

### 2.2 Geoagentes - Agentes Geográficos

Os geoagentes agem como elemento integrador da aplicação GIS com o restante dos componentes da arquitetura. O que justifica o uso da tecnologia de agentes no presente modelo é que eles trabalham em nome dos usuários integrando-se às aplicações GIS, auxiliando nas buscas e requisições de serviços abertos de geoprocessamento [15]. A adoção dos agentes diminuirá a interferência do usuário até que toda a informação requisitada seja recuperada.

Cada aplicação deve possuir seus próprios geoagentes. A implementação deles está vinculada ao modelo de dados e organização da aplicação, por um lado, e ao modelo OGIS, por outro, para garantir a heterogeneidade e abertura do sistema.

No atual modelo existem dois tipos de geoagentes diferentes: geoagente de importação e geoagente de composição, que se comunicam entre si para responder às necessidades dos usuários [6]. Os geoagentes conhecem o modelo de dados da aplicação e atuam como intermediadores. Os geoagentes de importação se comunicam com o GIS *trader* a procura do serviço desejado, e assim que o encontra, repassam sua referência para os geoagentes de composição, que fazem a requisição do serviço. Na figura 3 são mostrados os dois tipos de geoagentes e a comunicação entre eles e os demais componentes do modelo.

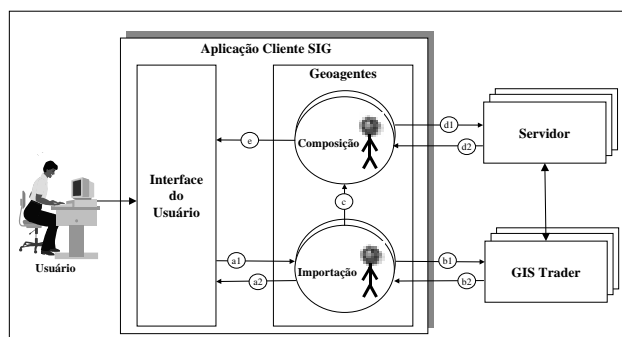


Figura 3: Modelo com detalhamento dos geoagentes.

#### 2.2.1 Geoagente de Importação

O geoagente de importação faz interface com o GIS *trader*, para localizar serviços. Através da interface do usuário, o geoagente de importação recebe as informações necessárias para que seja feita uma requisição (fluxo a1, figura 3). De posse das informações, o geoagente de importação começa a negociar com o GIS *trader*, para importar um serviço. O geoagente de importação requisita ao GIS *trader* uma lista de serviços disponíveis, do tipo de serviço solicitado pelo usuário (fluxo b1, figura 3). Se o GIS *trader* possuir algum serviço do tipo solicitado, ele envia uma lista dos serviços encontrados (fluxo b2, figura 3). Para encontrar o serviço, pode ser necessário que o GIS *trader* estabeleça federação com outros GIS *traders*,



mas isso é transparente para o geoagente de importação. Se nenhum serviço for encontrado, uma mensagem é enviada ao geoagente de importação, que retorna a aplicação do usuário (fluxo a2, figura 3). Podem ser encontrados diversos serviços de um mesmo tipo em um único GIS *trader*, então o GIS *trader* envia uma lista com todos os serviços encontrados. Ao receber a lista de serviços, o geoagente de importação escolhe, aleatoriamente, um serviço para fazer a invocação. A escolha do serviço a ser invocado poderia ser feita de várias outras maneiras: uma delas seria mandar a lista para que o usuário escolhesse; outra forma seria escolher o primeiro serviço da lista. A opção por escolher o serviço aleatoriamente é para evitar ao máximo a interferência do usuário. O critério de escolher o primeiro serviço da lista, poderia ficar sobrecarregando um único serviço. Em um futuro, o geoagente poderá ser inteligente para fazer a escolha de forma criteriosa.

Depois de escolher um serviço, o geoagente de importação passa as informações para o geoagente de composição (fluxo c, figura 3).

### 2.2.2 Geoagentes de Composição

O geoagente de composição é responsável por invocar o serviço no servidor de geoprocessamento e fazer a composição dos resultados, se necessário. Ele conhece o modelo de dados da aplicação e faz os ajustes necessários para que não ocorram problemas de semântica dos dados. Portanto, quando o geoagente de composição recebe do geoagente de importação a referência do objeto, ele faz a invocação do serviço no servidor (fluxo d1, figura 3), que envia os resultados (fluxo d2, figura 3). O geoagente de composição faz os ajustes necessários no resultado e o envia para a aplicação cliente (fluxo e, figura 3). Em alguns casos pode ser necessário que vários servidores sejam invocados, por exemplo: o usuário necessita da composição de dados que estão espalhados em vários servidores. Nesse momento será necessária a composição de todos os resultados obtidos.

## 3 Um Exemplo

Para validar o modelo, foi feito um estudo utilizando o caso do Projeto “Corredor Atlântico do Mercosul”, ou abreviadamente CAM. O consórcio Corredor Atlântico do Mercosul é uma organização privada, sem fins lucrativos, que abrange portos, companhias de navegação e sindicatos, com o objetivo de tornar as vias costeiras mais competitivas. A iniciativa visa agilizar e facilitar a integração entre os países do Mercosul (Brasil, Argentina, Paraguai e Uruguai), transferindo alguns milhões de toneladas de carga da rede rodoviária intra-costal para o sistema costeiro de navegação [16].

O exemplo trata do uso dos objetos básicos do modelo. Considera-se a seguinte consulta: uma carga de

milho situada em Buenos Aires - Argentina deverá ser enviada para Recife - Brasil. Neste exemplo o Administrador da Federação se localizará no Rio de Janeiro por ser um ponto central do Mercosul, otimizando o tráfego de mensagens na rede. Assim, se houver necessidade de cadastramento de novos serviços, os pedidos são enviados para este ponto, bem como, quando houver a instanciação de novos GIS *traders* no Corredor. Conforme novos pontos são adicionados ao Corredor, novos GIS *traders* são instanciados, já que a demanda e oferta de serviços aumenta.

Primeiramente, o ponto de Buenos Aires deve ter conhecimento de todas as vias pelas quais essa carga pode ser transportada. Dessa forma, são verificados quais dados são necessários para a execução da consulta. Se houver dados que o ponto não possui, por estarem distribuídos na rede, ou que necessitem ser atualizados, um pedido é enviado para o geoagente de importação para que encontre os mesmos. O geoagente de importação se comunica com o GIS *trader*, que informa uma lista de serviços de dados referentes a estradas, portos e ferrovias em que a carga poderá passar. Caso seja necessário, o GIS *trader* ao qual o ponto de Buenos Aires está ligado, se comunica com outros GIS *traders* da federação a procura dos serviços. Neste momento, podem acontecer três problemas:

1. o serviço não ser encontrado no tempo limitado: uma mensagem é retornada para o usuário avisando o *timeout*;
2. o tipo de serviço desejado ainda não foi cadastrado pelo Administrador da Federação. Neste caso, deve ser enviado um pedido para que o Administrador da Federação verifique a possibilidade da criação do novo tipo de serviço e assim, as aplicações comecem a disponibilizar esse serviço nos diversos GIS *traders* da federação;
3. algum GIS *trader* com o qual se deseja estabelecer federação está com problemas. Neste caso, pede-se ao Administrador da Federação a referência de um outro GIS *trader* para que se estabeleça uma nova federação. Dessa forma a busca de serviços não fica prejudicada pelo fato de se ter um GIS *trader* sem funcionar.

Não havendo problemas, o GIS *trader* fornece para o geoagente de importação uma lista dos serviços encontrados. O geoagente de importação escolhe, aleatoriamente, um serviço da lista e passa a referência do mesmo para o geoagente de composição que faz a invocação do serviço no devido servidor. Conforme os resultados da invocação vão sendo obtidos, o geoagente de composição vai fazendo os devidos ajustes para entregá-los ao usuário. Dessa forma, é transparente para o usuário quais servidores foram invocados para a obtenção dos dados. Estes dados se encontram distribuídos nos diversos dos pontos do Mercosul e são sendo sempre

atualizados, dessa forma as tomadas de decisões tornam-se mais precisas.

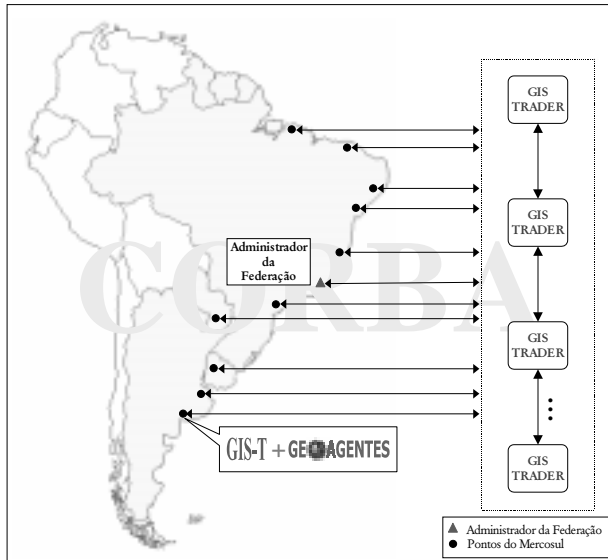


Figura 4: Exemplo de distribuição dos componentes no CAM.

Com os resultados, a aplicação SIG faz o mapeamento das estradas, ferrovias e portos, fazendo um estudo comparativo custo/benefício dos possíveis caminhos. Várias questões devem ser observadas, por exemplo:

- disponibilidade do porto;
- tempo de viagem entre as diversas cidades do percurso;
- distância entre as cidades;
- tempo que leva para descarregar a mercadoria.

Pode ser que a melhor rota necessite da utilização de vários meios de transporte, criando assim, uma rede intermodal.

#### 4 Conclusões

Este trabalho teve por objetivo propor um modelo onde diversos sistemas de informações geográficas distribuídos pudessem se comunicar, compartilhando dados e serviços, sem se preocupar com questões relacionadas a heterogeneidade, localização dos dados e serviços e distâncias entre eles. Como resultado, foi apresentado um modelo obtido através da combinação de tecnologias emergentes nas áreas de sistemas distribuídos, sistemas de informações geográficas e agentes. Foram utilizadas as especificações CORBA e OGIS, e agentes, surgindo assim os conceitos de GIS traders e geoagentes, que formam o modelo proposto.

A interoperabilidade é alcançada tanto em nível operacional, utilizando CORBA, que garante a localização e interoperabilidade de objetos em ambiente distribuído e heterogêneo, quanto em nível de aplicação através do uso

do padrão OGIS que garante o compartilhamento de dados geográficos. Os geoagentes e o GIS trader surgem para auxiliar os usuários nas buscas dos dados e serviços que se encontram distribuídos, para facilitação de aplicações de geoprocessamento heterogêneas em ambientes de larga escala.

O GIS trader, baseado nas especificações CORBA, foi criado para organizar o cadastramento de serviços de geoprocessamento distribuídos em redes heterogêneas. O que justifica a criação de um GIS trader específico é que os serviços de geoprocessamento apresentam grande complexidade, necessitando assim de uma forma eficiente para organizá-los, para facilitar as buscas e, conseqüentemente, aumentar o compartilhamento. A federação de GIS traders é necessária quando se trata de redes de longas distâncias, pois além de não sobrecarregar um único GIS trader, o anúncio e cadastramento passa a ser distribuído, permitindo que usuários e aplicações se comuniquem com GIS traders mais “próximos”, diminuindo o tráfego na rede. Outra vantagem é que se houver um GIS trader com problemas, existem outros para executar o pedido. O Administrador da Federação foi criado para fazer o gerenciamento dos GIS traders da federação, bem como para controlar a criação de tipos de serviços, amenizando problemas de ambigüidade semântica, muito comum em geoprocessamento.

Agentes vêm sendo aplicados em várias áreas para auxiliar na execução de tarefas. Com isso, foi observada a possibilidade de agentes auxiliarem em tarefas relacionadas a geoprocessamento, surgindo assim conceito de geoagente. Os geoagentes definidos no modelo irão se comunicar com o GIS trader e com servidores de geoprocessamento, auxiliando na busca e invocação de serviços, bem como executando tarefas de conversão e composição de dados geográficos. Dessa forma, a obtenção de dados se torna mais fácil e, principalmente, transparente para o usuário, que não irá se preocupar com localização e formatos dos dados.

Como sugestões para trabalhos futuros, propõe-se a implementação do modelo para que possam ser avaliadas as suas vantagens e também verificadas questões relacionadas ao desempenho do sistema, para eventuais ajustes na implementação, bem como no modelo. Uma outra sugestão é a evolução dos geoagentes através de um estudo mais detalhado sobre inteligência artificial, para que os agentes passem a raciocinar e tomar decisões mais precisas. Também deve ser estudada a possibilidade de agentes móveis, pois esses agentes podem procurar os dados com maior facilidade, bem como ajudar no compartilhamento de recursos distribuídos. Também é sugerido um estudo mais aprofundado com relação a GIS traders e federações de GIS traders, visando encontrar melhores formas de organização para obter mais precisão e eficiência nas buscas dos serviços. Sugere-se a verificação da possibilidade de dividir os GIS traders por

contextos, onde cada GIS *trader* faz o cadastro de um tipo de serviço específico, no sentido de aumentar ainda mais as possibilidades de escalabilidade e atendimento a demandas futuras. Esses trabalhos possibilitarão melhoras e ajustes no modelo proposto, trazendo benefícios para o compartilhamento de aplicações de geoprocessamento.

## 5 Referências Bibliográficas

[1] Bearman, M. *Trading in Open Distributed Environments*, **Proceedings of the International Conference on Open Distributed Processing**, Brisbane, Austrália, fevereiro 1995.

<http://www.dstc.edu.au/Publications/index.html>

[2] Behrens, C., Shklar, L., Basu, C., Yeager, N., Au., E. *The Geospatial Interoperability Problem: Lessons Learned from Building the Geolens Prototype*. **Proceedings of Interop97**, Califórnia, dezembro 1997.

<http://geolens.tipandring.com>

[3] Buehler, K. **The OpenGIS™ Guide – Introduction to Interoperable Geoprocessing**. Ed. K. Buehler and L. McKee, Massachusetts, 1998.

<http://opengis.org/guide/guide1.html>.

[4] Câmara, G., Casanova, M. A., Hemerley, A. S., Medeiros, C. M. B., Magalhães, G. **Anatomia de Sistemas de Informação Geográfica**. X Escola da Computação, Campinas: SBC, julho 1996.

[5] Filho, J., Iochpe, C. *Introdução a Sistemas de Informações Geográficas com Ênfase em Banco de Dados*. **XV Jornada de Atualização em Informática**, Recife, outubro 1996.

[6] França, S. V. A. **Um Modelo de Sistema de Informações Geográficas Distribuído Baseado em Traders e Agentes**. Dissertação de mestrado DI/UFPE, Recife, dezembro 1999.

[7] Franklin, S., Graesser, A. *Is it an Agent, or Just a Program?: a Taxonomy for Autonomous Agents*. **Proceeding of Third International Workshop on Agent Theories, Architectures, and Languages**. Verlag, setembro 1996.

<http://www.msci.memphis.edu/~franklin/AgentProg.html>

[8] Gardels, K. *The Open GIS Approach to Distributed Geodata and Geoprocessing*. **Third International Conference/Workshop on Integrating GIS and Environmental Modeling**, pp. 21-25 Santa Fé, janeiro 1996.

<http://www.regis.berkeley.edu/gardels/envmodel.html>

[9] Lopez, X. R. *Interoperability Through Organization: The Role of Digital Libraries in Distributed Knowledge Management*. **Proceedings of Interop97**, Califórnia, dezembro 1997.

<http://bbq.ncgia.ucsb.edu/conf/interop97/program/papers>

[10] Jennings, N. R. e Wooldridge, M. **Agent Technology Foundations, Applications, and Markets**, UNICOM, 1ª edição, 1998.

[11] Macêdo, R. C. *Internet Trader: Um Trader para localização de objetos em um ambiente distribuído na Internet*. Plano de Dissertação, DI/UFPE, março 1999.

<http://www.di.ufpe.br/~rcdm/mestrado/index.html>

[12] Nwana, H. S. *Software Agents: An Overview*. Knowledge Engineering Review. Vol. 11, No 3, pp.1-40, setembro 1996.

<http://www.sce.carleton.ca/docs/AgentsOverview/ao.html>

[13] Orfali, R., Harkey, D., Edwards, J. **The Essential Distributed Objects - Survival Guide**. John Wiley & Sons, Inc, 1.a edição, 1996.

[14] Perez, C., Salgado, A., Ferraz, C. *Processamento de Informações Geográficas Distribuídas: Arquiteturas para as Redes de Serviços Públicos*. **XXIII Conferência Latino Americana de Informática (CLEI'97)**, pp. 157-167, Valparaíso, Chile, novembro 1997.

[15] Rodrigues, A., Grueau, C., Raper, J. Neves, N. *Research on Spatial Agents*. **Proceedings of the Third Joint European Conference and Exhibition on Geographical Information (JEC-GI'97)**, Viena, Austria, abril 1997.

<http://helios.cnig.pt/~armanda/>

[16] Vivacqua, P. A., Stehling, S. M. F. *A Logistic Project for South America's Integration*. **Seminário de La OCDE sobre Redes de Transporte Intermodal y Logística**, México, junho 1997.

[17] Vogel, A., Bearman, M., Beitz, A. *Enabling Interworking of Traders*, **Proceedings of the International Conference on Open Distributed Processing**, Brisbane, Austrália, Fevereiro, 1995.

<http://www.dstc.edu.au/Publications/index.html>

## Arquiteturas Cliente-Servidor para Bibliotecas Geográficas Digitais

José Roberto Osses<sup>1</sup>  
João Argemiro de Carvalho Paiva<sup>2</sup>  
Gilberto Câmara<sup>2</sup>

<sup>1</sup>FUNCATE- Fundação de Ciência, Aplicações e Tecnologia Espaciais, Av. Dr. João Guilhermino - 429 - 11o Andar  
Edifício Saint James, Centro - São José dos Campos - SP - Brasil - CEP: 12.210-131

{osses} @geo.funcate.org.br

<sup>2</sup>INPE-Instituto Nacional de Pesquisas Espaciais, Caixa Postal 515, 12201 São José dos Campos, SP, Brasil  
{miro,gilberto}@dpi.inpe.br

**Abstract.** This work describes the development of a client-server application for dissemination of geographical data on the Internet. The authors describe a strategy that relies on use of the conceptual schema of the geographical database as a means for data selection and for establishing a compromise between client and server. The resulting system has acceptable performance and indicates a possible client-server solution for geographical digital libraries.

**Resumo:** Este trabalho descreve o desenvolvimento de uma aplicação cliente-servidor para a disseminação de dados geográficos na Internet. Os autores descrevem uma estratégia que se baseia no modelo conceitual de uma base de dados geográfica como forma de seleção de dados, e para o estabelecimento de um compromisso entre cliente e servidor. O sistema resultante apresenta performance aceitável e indica a possibilidade de uma arquitetura cliente-servidor para bibliotecas geográficas digitais.

**Keywords.** Geographical databases, Geographical digital libraries, GIS.

### 1 Introdução

Um dos desafios crescentes para as instituições que lidam com informações geográficas é a publicação de dados através da Internet. Por sua natureza gráfica e bidimensional, o ambiente WWW (“World Wide Web”) oferece uma mídia adequada para a difusão da geoinformação. A médio prazo, espera-se que a disponibilidade “on-line” de grandes bases de dados espaciais e de ferramentas eficientes de navegação torne a geoinformação acessível de forma ampla, sem a necessidade de aquisição de software específico. Muitos esforços tem sido realizados neste sentido procurando estender as técnicas já utilizadas para bibliotecas digitais convencionais, que já utilizam a Internet para difundir seu acervo, para Bibliotecas Geográficas Digitais (BGD) onde os objetos a serem armazenados são capazes de representar dados geográficos.

O principal objetivo de uma BGD é fornecer ferramentas para armazenar, descobrir e recuperar dados geográficos. Uma das mais importantes iniciativas tomadas com este objetivo é a “Alexandria Digital Library” (ADL) na Universidade de Santa Bárbara na Califórnia (UCSB,2000). O propósito da ADL é aprimorar o gerenciamento de acesso a informações

geograficamente referenciadas, tais como mapas, fotos aéreas e atlas. A sua interface permite ao usuário navegar no banco de dados usando uma seleção de regiões a partir de um mapa mestre e também selecionando funções no menu. A arquitetura da ADL envolve quatro componentes básicos:

- A interface com o usuário que suporta acesso de forma gráfica e textual aos outros componentes do sistema.
- Um catálogo distribuído que permite ao usuário identificar repositórios de interesse.
- Um componente de armazenamento distribuído contendo os repositórios digitais.
- Um componente de ingestão que permite armazenamento de novos repositórios, extração de metadados e adição dos metadados aos catálogos.

Diversas arquiteturas tem sido propostas para BGD e de maneira geral a solução adotada para facilitar a difusão de informação geográfica através da Internet é acoplamento de um servidor de dados geográficos. As tecnologias comerciais disponíveis podem ser enquadradas genericamente em duas grandes classes:

- Servidores de mapas, que, respondendo a pedidos remotos, enviam uma imagem (matriz) de tamanho fixo nos formatos GIF ou JPEG. Esta solução permite configurar o servidor para responder a diferentes tipos de consulta, sem requerer que todos os dados a ser transmitidos sejam pré-computados. Entretanto, o usuário consegue visualizar apenas as imagens enviadas; qualquer novo pedido é enviado de volta para o servidor, resultando em mais uma transferência pela Internet. Dependendo da velocidade de acesso, esta estratégia pode resultar em longos e sucessivos períodos de espera. Como por exemplo temos o “Internet Map Server” da ESRI.
  
- Clientes de Apresentação, que adotaram como solução a transmissão de todos os dados no formato vetorial para a máquina do cliente, com posterior visualização local. Estes servidores encapsulam a informação em formatos gráficos, que podem ser apresentados por meio de programas adicionais (“plug-ins”) acoplados a “browsers” como o Netscape ou o Explorer ou por meio de “applets” JAVA. Esta estratégia permite uma maior flexibilidade do lado do cliente, que pode realizar operações locais de visualização e consulta sob os dados transferidos. O tempo de acesso inicial para transferência é maior que no caso anterior, mas muitas das operações posteriores serão realizadas localmente, o que resulta usualmente em um tempo de resposta médio melhor. Exemplos são os produtos “Geomedia Web Map” da INTERGRAPH, “Map Guide” da AUTODESK e “SpringWeb” do INPE.

No entanto, as duas alternativas apresentam problemas. No primeiro caso, todos os dados ficam armazenados no servidor, e no segundo, todos precisam ser transferidos para o cliente. Seria mais conveniente dispor de configurações *cliente-servidor*, que pudessem balancear os pedidos de consulta, permitindo uma apresentação e navegação local em parte dos dados e realizando acessos remotos ao servidor apenas quando estritamente necessário.

Além dos problemas relativos a transmissão de dados muitos outros aspectos devem ser levados em conta para a construção de uma BGD. Ao se coletar dados geográficos para fazerem parte da

BGD já se depara com um grande problema que é a falta de padronização de formatos, modelos semânticos e metadados para os diversos sistemas de onde eles tiveram suas origens, dificultando assim sua integração. A tarefa de definir um modelo abrangente que possa ser utilizado nas diversas aplicações de Sistemas de Informações Geográficas não é trivial e até o momento não se chegou a um consenso. Estão surgindo diversas propostas de padronizações e um dos grupos que tem se preocupado com estes problemas é o Consórcio OpenGIS® (OGC,2000).

Neste contexto este trabalho apresenta uma proposta de arquitetura cliente-servidor para dados geográficos, abordando alguns aspectos relacionados a BGD tais como transmissão e apresentação de mapas, conversão de dados e buscas por atributos e regiões. Esta arquitetura está baseada em tecnologia JAVA, utilizando um “applet” no ambiente do cliente e um “servlet” no ambiente do servidor, e propõe uma forma de interface com o usuário que irá produzir consulta, que serão pré-processadas pelo “applet” e enviadas ao “servlet” se necessário.

O trabalho está organizado como segue. Na seção 2, discutimos as alternativas de arquitetura cliente-servidor para bibliotecas digitais. Na seção 3, descrevemos os componentes da solução proposta. Na seção 5, fazemos um balanço dos pontos positivos e negativos de nossa proposta.

## 2. Arquiteturas para Bibliotecas Geográficas Digitais

Gardels (1997) apresenta uma visão geral de arquitetura de uma biblioteca geográfica digital, operando no ambiente da Internet. Na sua visão, esta arquitetura apresenta os seguintes componentes, ilustrados na Figura 1:

- Módulo visualizador.
- Módulo Analisador.
- Módulo Fusor.
- Módulo Atomizador.

O módulo visualizador é constituído basicamente pela interface com o usuário. Esta interface deve permitir aos usuários descobrir, recuperar e apresentar dados geográficos e pode ser dividida em módulos de apresentação de atributos e outros módulos com o propósito de atender requisitos específicos da aplicação.

De maneira geral, o usuário não conhece previamente o conteúdo da BGD. Assim a interface deve fornecer meios para que ele busque por dados de sua área de interesse, como por exemplo através de uma ferramenta de interação com um mapa onde ele possa selecionar uma região geográfica. Uma outra forma pode ser através do uso de metadados que podem direcionar os usuários a formular comandos de busca que permitam localizar os dados desejados através de seleção de atributos que satisfaçam seus requisitos.

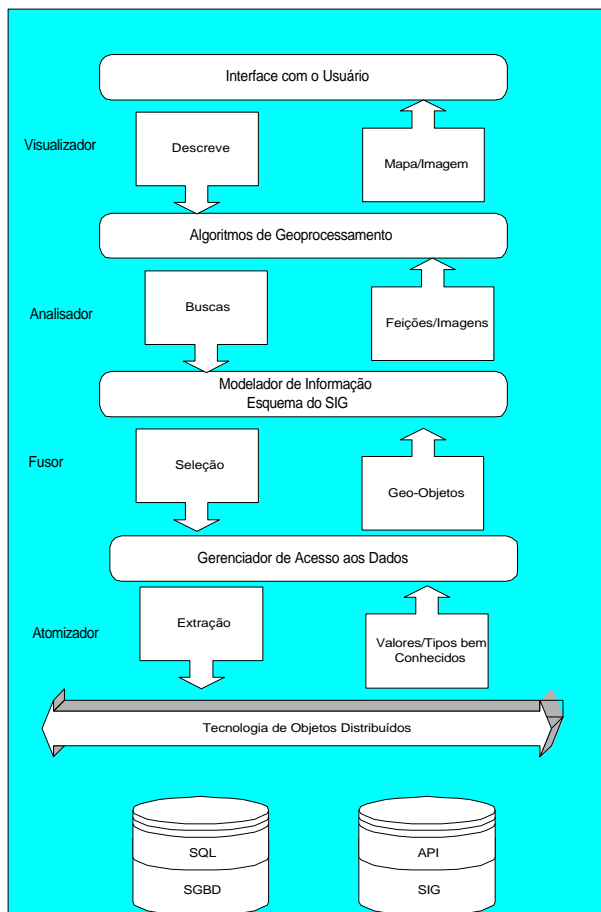


Figura 1 – Arquitetura proposta por Gardels.

O módulo analisador, de forma oposta ao visualizador, geralmente está localizado no servidor, totalmente ou em grande parte. Ele será responsável por: (a) receber descrições geradas na interface com o usuário interpretar e executar as instruções nelas contidas; (b) Montar as coleções de objetos geográficos que serão enviadas como resposta as descrições para a interface com o usuário. O analisador pode estar centralizado em

um servidor único ou diversos servidores atuando de forma federada (Laurini,1998). O módulo fusor trata dos problemas de integração de diversos esquemas utilizados para originar os dados que integrarão a BGD. Em cada sistema os dados geográficos possuem representações conceituais diferentes. Ainda não existe um consenso com relação a um esquema de representação conceitual para dados geográficos. Alguns esforços tem sido feitos neste sentido sendo os mais notáveis os efetuados pelo consórcio OpenGis®. O módulo atomizador trata das conversões estruturais dos dados armazenados em diferentes sistemas. Existem alguns padrões intermediários utilizados para transferência tais como DXF, SDTS, SAIF etc. Este mecanismo não garante que os dados sejam todos convertidos devido a dificuldade de representar nestes formatos detalhes internos de todos os sistemas utilizados.

### 3 Arquitetura Proposta

#### 3.1 Configuração

A arquitetura proposta está baseada no modelo de camadas múltiplas, sendo utilizadas três camadas neste trabalho. A primeira camada é um navegador para Internet que serve como um Cliente Universal. Acoplado ao navegador temos um “applet” JAVA. O Applet é responsável por funções ligadas ao cliente tais como: apresentação de mapas e objetos geográficos, controle de diálogos e interfaces, controle de cache de dados e controle de geração de expressões em linguagem de consulta.

A segunda camada é constituída de um servidor para protocolos HTTP com capacidade de executar “servlets” em JAVA. O “servlet” é um aplicativo que permanece em execução no servidor aguardando por solicitações dos clientes e tem a capacidade de atender diversas solicitações simultâneas. O “applet” pode estabelecer uma conexão com qualquer servidor onde exista um “servlet” preparado para receber suas requisições. A seleção do servidor pode ser feita de forma predefinida ou pode ser fornecida ao usuário uma interface para seleção através de seu nome ou endereço IP. Não existem restrições técnicas com relação ao número de servidores que podem ser acessados simultaneamente por um “applet”. O protocolo de comunicação entre estas duas camadas é o HTTP de onde são especialmente utilizados os métodos GET e POST para implementação de um sub-protocolo para troca de mensagens específicas entre as camadas.

A descrição do sub-protocolo utilizada neste trabalho é apresentada em tópico adiante que descreve a comunicação entre o “applet” e o “servlet”.

A terceira camada é composta por um Sistema Gerenciador de Banco de Dados Relacional que armazena todos os dados utilizados pelo sistema. O “servlet” acessa as informações do banco de dados relacional utilizando a API JDBC da linguagem JAVA que é um conjunto de especificações que define como um programa escrito em JAVA pode se comunicar e interagir com um banco de dados (Siple,1998).

O sistema desenvolvido para avaliar esta arquitetura foi denominado SIGTEIA formado pelo anacronismo entre Sistema de Informações Geográficas e uma variação da tradução de WEB (Rede, Teia) utilizado para designar a Internet.

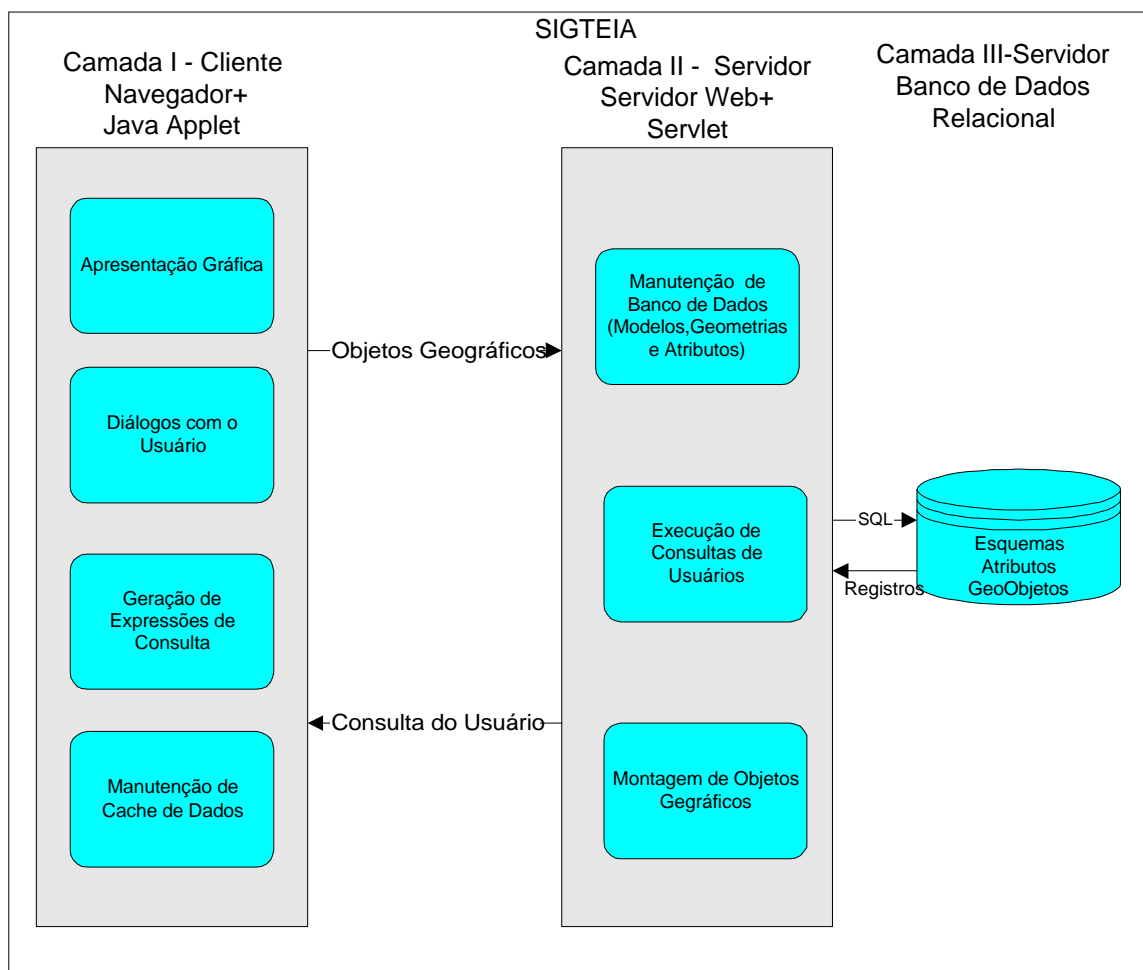


Figura 2 – Funções principais do SIGTEIA.

O “servlet” é encarregado de interpretar as solicitações dos clientes, acessar o banco de dados, executar os comandos necessários, preparar e enviar a resposta adequada de volta ao cliente.

A Figura 2 apresenta as funções principais de cada parte do sistema ligados ao cliente e ao servidor.

### 3.2 O Modelo Conceitual

O Modelo Conceitual adotado é baseado no modelo do SPRING (INPE/DPI, 2000). Neste trabalho estão sendo tratadas somente entidades geo-referenciadas que podem ser classificadas como Geo-Objetos. Além disso a única representação implementada para os Geo-Objetos é a vetorial. Outra diferença com relação ao

Modelo do SPRING é que cada Geo-Objeto possui uma única representação que é do tipo “WKG” (*Well Known Geometry*) definidos pelo padrão OpenGis®. A persistência dos dados é feita totalmente em bancos de dados relacionais utilizando campos binários longos para armazenamento de “WKG”. A Figura 3 apresenta o modelo orientado por objetos do SIGTEIA.

O Banco de Dados Geográfico é a classe principal neste modelo. Uma instância desta classe contém informações a respeito do esquema do banco de dados em uso ou ativo, ou seja, quais os projetos, projeções, categorias, planos de informação e geo-objetos para determinado banco de dados geográfico. Um servidor de dados geográficos pode possuir diversos bancos de dados geográficos e cada um deles com esquemas diferentes. No caso do SIGTEIA quando o “applet” faz uma requisição solicitando o esquema de determinado banco de dados geográfico o “servlet” fica responsável por gerar uma instância desta classe e enviar este objeto como resposta. O Projeto define uma região física normalmente retangular onde os dados estão contidos. Este Projeto está associado a uma projeção e possui um conjunto de planos de informação que por sua vez vão possuir uma coleção de Geo-Objetos. As categorias procuram descrever fenômenos que possuem características comuns e desta forma agrupá-los para facilitar seu entendimento. No SIGTEIA existem dois tipos básicos de categoria uma para determinar as características dos planos de informação, ou seja, que tipos de dados estão sendo agrupados em cada plano e outra que determina características comuns entre Geo-Objetos que é denominada categoria de objetos. Um plano de informação de determinada categoria pode receber objetos de diversas categorias de objetos diferentes.

No SIGTEIA o Visual esta relacionado ao Geo-Objeto e não a categoria de objeto a que ele esta associado. Desta forma o usuário poderá efetuar diferenciações de representação entre Geo-Objetos da mesma categoria que pode ser útil para operações tais como agrupamento por atributos semelhantes.

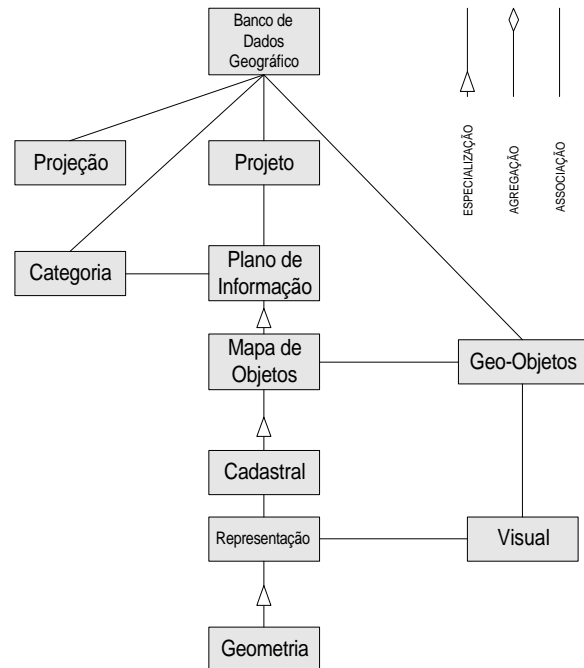


Figura 3 – Modelo Orientado por Objetos do SIGTEIA

### 3.3 Organização do Banco de Dados Geográfico

O Banco de Dados Geográfico é organizado no servidor e está totalmente contido em um Sistema de Banco de Dados Relacional. Um servidor pode possuir diversos bancos de dados geográficos e pode estar atendendo a demanda por informações referentes a bancos diferentes ao mesmo tempo. Cada banco de dados possui tabelas que armazenam informações com relação ao esquema dos dados, a geometria e atributos de Geo-Objetos.

### 3.4 Funções do “Servlet”

O “servlet” fica localizado no servidor junto com o Banco de Dados Geográfico, e é responsável por manter uma conexão com o Banco de Dados Relacional e com o “applet”. Na conexão com o “applet” se faz necessário estabelecer um protocolo para que o “servlet” seja capaz de interpretar as solicitações efetuar as atividades corretas e enviar as respostas adequadas. Por exemplo o “applet” pode solicitar que o “servlet”



envie o esquema do banco de dados geográficos, e a ação a ser tomada é conectar-se ao banco de dados ler as informações relativas ao esquema, montar objetos para as classes pertencentes ao esquema e enviar estes objetos de volta ao “applet”. O protocolo utilizado pelo “servlet” para comunicação com o banco de dados relacional é o padrão SQL. A conexão é efetuada através da chamada de rotinas pertencentes a API JDBC. O servidor carrega o “servlet” que pode então aceitar diversas requisições dos clientes e retornar dados a eles. O “servlet” possui métodos denominados “doGet” e “doPost” que são acionados toda vez que ele recebe uma requisição

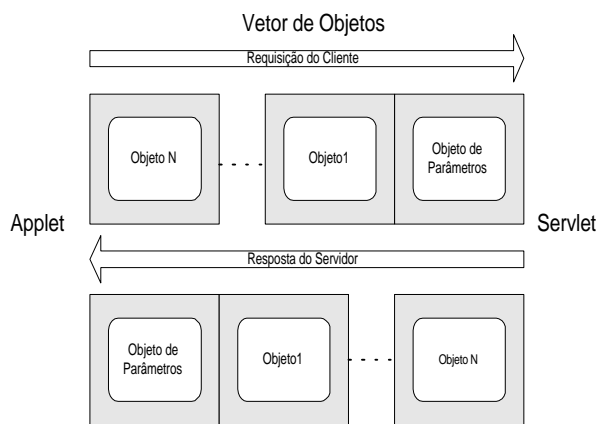


Figura 4 – Troca de informações entre Applet/Servlet

do cliente do tipo GET e POST do protocolo HTTP. Estes métodos efetivamente irão tratar da comunicação entre o “servlet” e os clientes. Para o SIGTEIA um protocolo foi definido para possibilitar esta comunicação. Toda vez que o cliente envia um comando de GET ou POST deve enviar também um vetor de parâmetros e objetos para que o “servlet” interprete execute o comando e devolva outro vetor contendo parâmetros e objetos como resposta. A Figura 4 exemplifica o fluxo de parâmetros e objetos entre “Applet/Servlet”.

### 3.5 Funções do Applet

O “applet” tem como funções básicas o controle de apresentação do modelo e dos dados geográficos e seus atributos aos usuários. Além disso o “applet” se preocupa em manter um cache de dados do lado do cliente de forma a minimizar a transferência de dados de forma desnecessária entre o cliente e o servidor. As interfaces com o

usuário estão organizadas em forma de painéis sobrepostos. O controle de seleção de ativação de painéis é feito por meio de tabuladores posicionados em uma de suas laterais com um nome identificador do painel. Os painéis contém os elementos de interface, tais como botões, listas, e textos editáveis, utilizados para interação com o usuário. A Figura 5 apresenta a interface inicial do “applet” do SigTeia. O sistema tem suas funções divididas em 8 painéis principais: Conexão, Esquema, Apresentação, Seleção por Planos, Seleção por Atributos, Metadados e Mensagens. Os painéis interagem entre si e comandos em um deles pode afetar o contexto de outro. Basicamente as funções de cada painel são:

- Conexão, controla a seleção de servidores e estado da conexão trocando mensagens entre o “applet” e o “servlet”.
- Esquema, apresenta o esquema através de uma interface hierárquica em forma de árvore.
- Seleção por planos, permite a seleção total de um plano de informação para apresenta. Possui dispositivos para atribuir cores e prioridades de apresentação dos geo-objetos pertencente ao plano selecionado.
- Seleção por atributos, permite a montagem de consultas simples em SQL que serão enviadas ao servidor para buscar os objetos desejados.
- Objetos, é uma interface em forma de tabela utilizado para apresentar os atributos que podem ser selecionados pelo usuário.
- Apresentação, possui uma área para desenho que permite a interação do usuário com a representação gráfica dos objetos selecionados.
- Metadados, é apresentada em forma textual algumas informações adicionais que podem ser importantes para facilitar o entendimento pelo usuário do conteúdo das informações no banco de dados. O texto apresentado esta em formato HTML.

## 4. Análise da Arquitetura Proposta

A grande vantagem da arquitetura desenvolvida no trabalho é sua relativa simplicidade. Partindo-se de um modelo de dados bem definido, foram definidos componentes cliente-servidor para dados geográficos que operam de forma semelhante aos bancos de dados convencionais. Os dados são trazidos para a máquina do cliente sob demanda.

Este trabalho mostra que o uso de um modelo de dados conceitual facilita muito o estabelecimento de um compromisso cliente-servidor. O uso de um modelo semântico forte aumenta a percepção cognitiva do usuário sobre o conteúdo do banco de dados residente num servidor remoto.

A principal limitação da arquitetura proposta é sua ênfase nos aspectos descritivos dos dados para a seleção e falta de filtragem espacial dos dados. Assim, caso o usuário queira visualizar uma área geográfica limitada e o dado desejado cubra uma área maior, ele é recuperado na sua totalidade e enviado para o cliente. Isto resulta numa transmissão de dados possivelmente desnecessária.

## 5. Conclusões

Este trabalho mostra a viabilidade do desenvolvimento de tecnologias JAVA (“applet”-“servlet”) para bibliotecas geográficas. Indica ainda que o uso de um esquema conceitual pode ser importante suporte neste tipo de aplicativo. Finalmente, mostramos que uma solução completa deveria envolver outros aspectos não considerados neste trabalho.

## Agradecimentos

Este trabalho faz parte do projeto de cooperação conjunta CNPq/NFS sobre Interooperabilidade em Sistemas de Informações Geográficas, sendo financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

## Referências Bibliográficas

Câmara, G.; Souza, R.C.M.; Freitas, U. M.; Garrido, J. “SPRING: Integrating Remote Sensing and Gis by Object-Oriented Data Modelling”. *Computer and Graphics*, vol. 20, n.3,1996.

Gardels, K. (1997). Open GIS and On-Line Environmental Libraries, *SIGMOD Record*, Association of Computing Machinery Special Interest Group on Management of Data, March 1997.

INPE/DPI (2000). SPRING: Sistema de Processamento de Informações Geo-referenciadas. <<http://www.dpi.inpe.br/spring>>

Laurini R. (1998). Spatial Multidatabase Topological Continuity and Indexing: a Step

towards Seamless GIS Data Interoperability. *International Journal of Geographical Information Sciences*. Vol. 12,4 June 1998, pp. 373-402.

OGC- OpenGis® Consortium, 2000. Technical Specifications.

<[www.opengis.org/techno/specs.htm](http://www.opengis.org/techno/specs.htm)>

Siple, Mathew D. *The complete guide to JAVA database programming with JDBC*, Computing McGraw-Hill, 1998.

UCSB, 2000- University of California at Santa Barbara, 2000 – <<http://alexandria.sdc.ucsb.edu>>

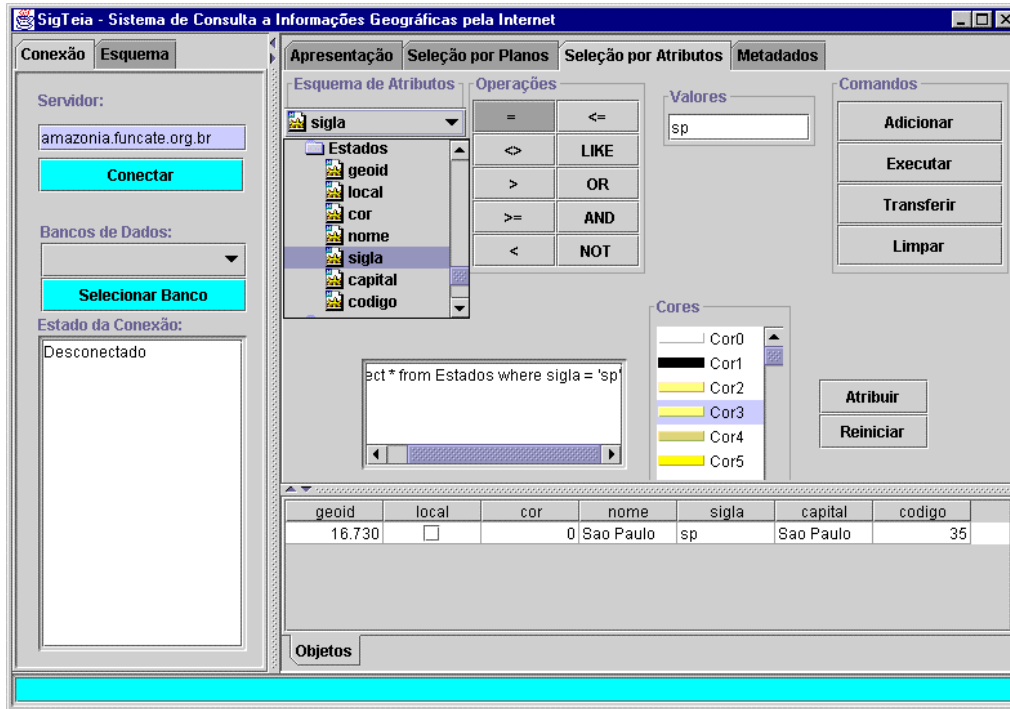


Figura 5 - Painel de Seleção por Atributos.

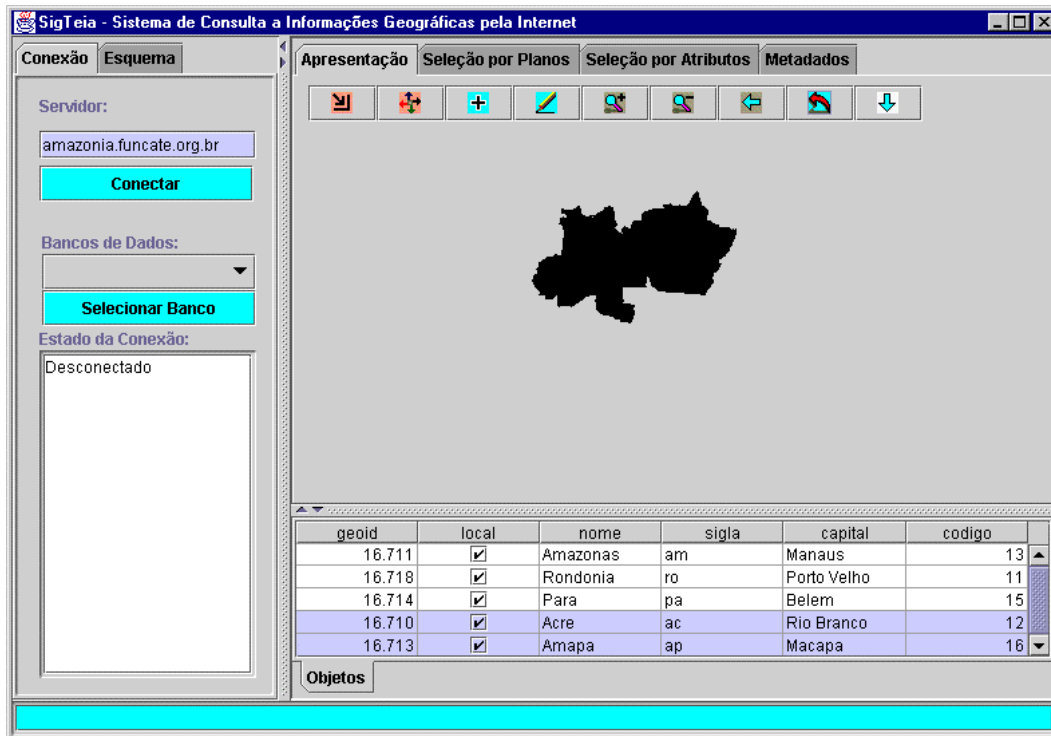


Figura 6 – Painéis de Apresentação, Conexão e Objetos.

# MODELAGEM GEOMÉTRICA 3D E ANÁLISE ESPACIAL DE SISTEMAS CÁRSTICOS COM O USO DE OPENGL E SQL

JOSÉ ANTONIO FERRARI<sup>1</sup>  
ALAIN MANGIN<sup>2</sup>

<sup>1</sup>IG Instituto Geológico Av. Miguel Stefano 3900, 04301-903 São Paulo - SP Brasil  
ferrari@igeologico.sp.gov.br

<sup>2</sup>Laboratoire Souterrain du Centre National de Recherche Scientifique - Moulis 09200 - Saint Girons France

**Abstract.** This paper describes a system developed for geometrical modeling and analysis of karst systems. The program calculates the closure of loops in cave surveys by least square method, and provides the means to obtain query derived maps of the underground system. The software integrates OpenGL and Microsoft Jet database engine technologies to produce 3D visualization of cave passages and digital elevation models linked with database. Some examples produced with the system are presented.

**Resumo.** Este trabalho descreve um programa para a modelagem geométrica e análise de sistemas cársticos. O programa utiliza o método dos mínimos quadrados para calcular o fechamento de poligonais obtidas em levantamentos topográficos subterrâneos, e produz mapas a partir de pesquisas à base de dados. As tecnologias OpenGL e Microsoft Jet database engine são integradas para produzir visualizações 3D de cavernas e modelos de elevação digital, conectados a uma base de dados. Alguns exemplos elaborados pelo programa são apresentados.

## 1. Introdução

Pesquisas geocientíficas em terrenos cársticos muitas vezes utilizam informações distribuídas por uma rede intrincada de passagens subterrâneas. Nos estudos da dinâmica e estrutura desta paisagem, a integração dos dados subterrâneos com informações superficiais é conveniente. Ao estudar a geometria das rotas de fluxo subterrâneas, a distribuição de registros paleoclimáticos, ou no desenvolvimento de estudos ambientais, a visualização tridimensional do conjunto da paisagem (superfície e sistema subterrâneo), associada a uma base de dados, amplia o poder de análise do pesquisador.

Este trabalho apresenta uma descrição do sistema SIS (Sistema de Informações Subterrâneas) que integra a biblioteca gráfica *OpenGL* com o gerenciador de base de dados *Microsoft Jet database engine*. O programa calcula o fechamento de poligonais da topografia subterrânea, gerencia informações relativas aos diferentes segmentos da caverna e gera visualizações tridimensionais do sistema subterrâneo e das superfícies a ele associadas.

## 2. Visão geral do programa

O programa foi desenvolvido para a plataforma Windows com a linguagem VisualBasic 5. Foi empregada uma interface do tipo MDI (*multiple-document interface*), que permite ao usuário exibir vários documentos ao mesmo

tempo (cada documento em sua própria janela). Graças a esta interface, o usuário pode trabalhar com diferentes sistemas subterrâneos ou manipular várias informações de um mesmo sistema.

As informações obtidas no levantamento de campo são armazenadas numa tabela da base de dados. Os dados são processados (cálculo do fechamento de poligonais) e o resultado fica armazenado em outra tabela. Após este processamento, saídas gráficas podem ser geradas. As consultas (padrão SQL) que são feitas à base de dados, resultam em relatórios ou saídas gráficas (modelos tridimensionais). Também é possível executar consultas diretamente sobre a saída gráfica, neste caso, ao clicar o mouse sobre o objeto visualizado, sua identificação será exibida sobre o gráfico e o registro associado será iluminado numa tabela. A figura 1 ilustra o funcionamento geral do sistema.

## 3. Entrada de dados e estrutura da informação

A modelagem geométrica da caverna é obtida a partir dos dados da topografia subterrânea. O levantamento topográfico resulta num conjunto de visadas (linhas) que unem bases topográficas (pontos). Os atributos intrínsecos de uma visada são: base de partida, base de chegada, comprimento, direção e inclinação. Estas informações são inseridas no banco de dados através de planilhas. As medidas de largura e altura dos condutos,

obtidas em cada base, também são registradas, elas são utilizadas para dar volume ao sistema subterrâneo. Trabalhos sobre levantamentos topográficos em cavernas [1,2] discutem a metodologia de campo e a elaboração de plantas e perfis com detalhes de feições subterrâneas, este assunto não será abordado neste trabalho.

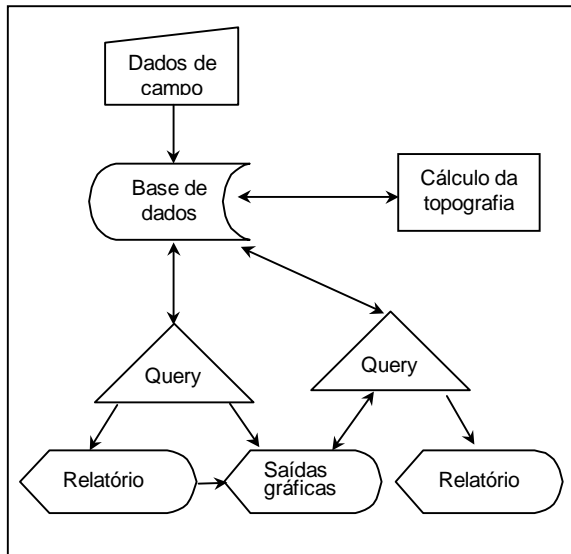


Figura 1. Visão geral do programa.

A evolução de uma caverna pode ser complexa, resultando em sistemas intrincados que apresentam vários andares ou níveis de desenvolvimento [3]. O SIS permite que os dados do mapeamento sejam distribuídos em diferentes setores, possibilitando o agrupamento de visadas em função de uma característica de interesse do pesquisador. A tabela para entrada de dados topográficos é a tabela padrão do programa. Nela estão armazenadas todas as visadas de todos os setores definidos pelo usuário. Para facilitar a manipulação dos dados, foram criados artifícios para a edição dos setores em tabelas independentes (uma janela para cada setor). O SIS possui todas as interfaces para a manipulação da base de dados.

Cada visada representa um segmento do sistema subterrâneo e seu comprimento depende da sinuosidade das passagens ou do interesse do topógrafo em destacar algum aspecto. O modelo relacional de banco de dados permite que além da tabela padrão (tabela de dados topográficos), várias tabelas de informação sejam associadas aos segmentos mapeados. A visada (base de partida e base de chegada) é a chave de ligação de um segmento da caverna com diferentes tabelas de informação. Os campos das tabelas podem ser definidos em função das necessidades do usuário, no entanto, os campos que estabelecem a ligação com o mapa (visada) são criados automaticamente e não podem ser excluídos.

As tabelas podem ser estruturadas com campos convencionais (numéricos, texto, booleanos e memo) e com campos (texto) que contêm o endereço de arquivos de imagem (bmp, jpg, gif, wmf ou emf) ou de texto (txt). Ao criar um campo que armazena o endereço de um arquivo, o usuário também define qual o aplicativo que será acionado para o abrir. O programa possui módulos nativos para exibir as imagens e o arquivo texto, mas é possível indicar um aplicativo externo para exibir os documentos. Neste caso, outros formatos de arquivo também podem ser escolhidos. Esta característica multimídia do sistema permite a interação de imagens, textos ou sons ao analisar segmentos da caverna. Por exemplo, pode-se visualizar fotos dos estratos e o croqui de um depósito sedimentar.

Esta capacidade do programa é útil em pesquisas que exploram registros sedimentares distribuídos ao longo de quilômetros de passagens subterrâneas, como os já executados no carste da Chapada Diamantina [4, 5], ou em estudos multidisciplinares voltados para o manejo turístico destes monumentos geológicos.

#### 4. Processamento dos dados topográficos

Em levantamentos topográficos, um problema bem conhecido é o do fechamento de poligonais. As imprecisões das medidas de campo levam a um erro durante o fechamento, impedindo a coincidência entre o ponto de partida e o ponto de chegada.

Quando se trabalha com apenas uma poligonal, não é difícil distribuir manualmente o erro entre as visadas. A correção manual torna-se impossível quando trabalhamos com sistemas complexos, onde um grande número de poligonais estão conectadas [6, 7]. Para eliminar a subjetividade na distribuição dos erros, utiliza-se o método dos mínimos quadrados, que pressupõe que os erros de medida são aleatórios e seguem uma distribuição gaussiana. Este método é adotado pelo SIS e segue o algoritmo de Taillard [7].

Durante o processamento, é gerada uma matriz de descrição do sistema, e são obtidas as componentes cartesianas das bases topográficas, ponderando os erros padrão que foram definidos para os instrumentos de medição (bússola, clinômetro e trena). As coordenadas x, y e z de uma base de origem (definida pelo usuário) são utilizadas para referenciar todo o sistema. Ao término do processo, as coordenadas de todas as bases topográficas são armazenadas numa tabela. Estas coordenadas são utilizadas na modelagem geométrica do sistema subterrâneo.

## 5. Modelagem geométrica das passagens subterrâneas e superfícies associadas

A visualização da modelagem geométrica das passagens subterrâneas e superfícies associadas é obtida com o uso do sistema gráfico OpenGL. Trata-se de um conjunto de funções gráficas 3D, desenvolvidas nos anos 80 para as estações de trabalho da Silicon Graphics. Tornou-se disponível para usuários de computadores pessoais quando a Microsoft incluiu uma implementação da API para o Windows NT 4.0 e, posteriormente para o Windows 95/98. Em essência, OpenGL renderiza polígonos e fornece algumas ferramentas para rotacionar, escalar e movimentar os objetos gráficos, no entanto, todo o trabalho para definir iluminação, criar e carregar modelos, desenvolver códigos para exibição e otimização depende do programador.

Para visualizar as passagens subterrâneas, as coordenadas cartesianas tridimensionais obtidas no processamento dos dados topográficos são transformadas para o plano 2D com o auxílio de uma projeção ortográfica. Os objetos 3D são construídos agrupando-se primitivas como linhas e polígonos. O volume do sistema é obtido considerando as medidas de largura e altura dos condutos. Neste caso, polígonos planares são utilizados para compor as laterais, o teto e o assoalho das passagens. As seções dos condutos são definidas nas bases topográficas e são representadas por retângulos. A direção da seção em cada base é obtida considerando a bissetriz do ângulo entre duas visadas. A figura 2 apresenta o método utilizado para dar volume ao sistema.

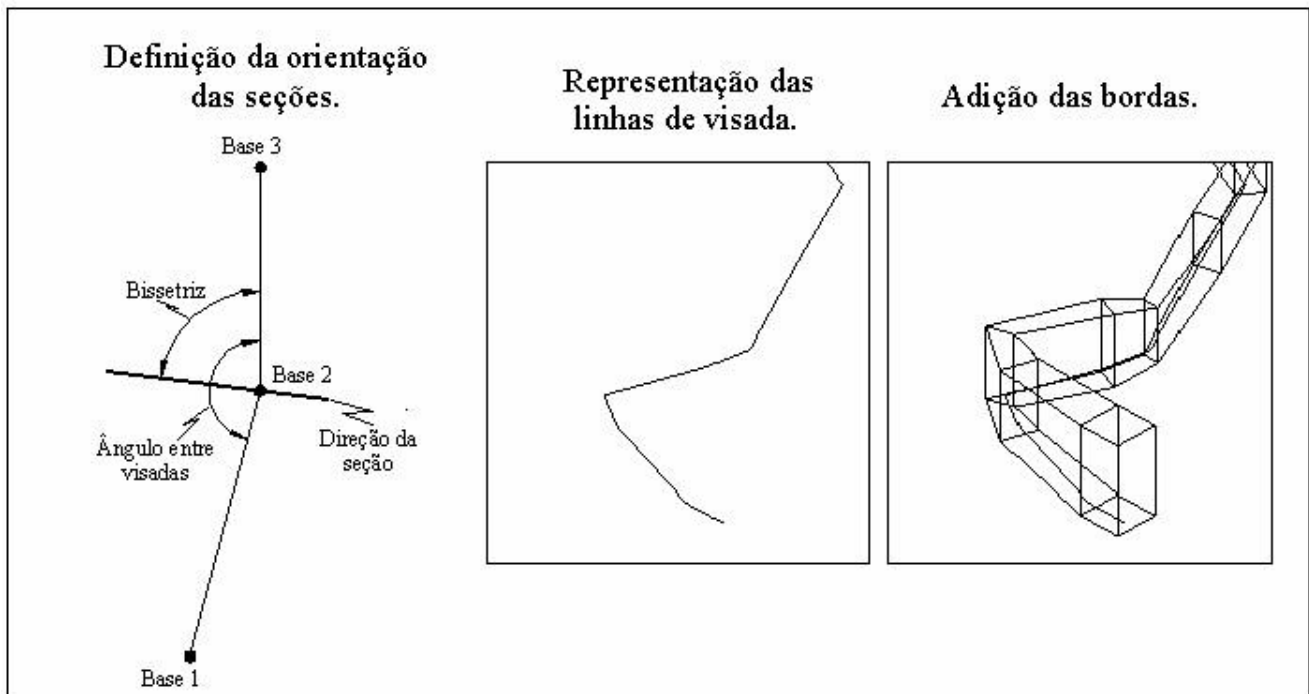


Figura 2. Modelagem das passagens subterrâneas

Superfícies geradas no programa Surfer [8] podem ser importadas e exibidas em conjunto com o sistema subterrâneo. A representação das superfícies é feita utilizando triângulos ou retângulos, cujos vértices são definidos pelas coordenadas dos nós do *grid* importado. A cor do polígono representa a média dos valores de *z* dos vértices.

Para iluminar e sombrear objetos 3D, o sistema gráfico necessita do vetor normal (vetor perpendicular ao

plano) de cada polígono desenhado. A representação da superfície com o uso de retângulos não permite que todas as possibilidades de renderização sejam utilizadas, pois raramente estes polígonos possuem um único plano no sistema cartesiano 3D. Neste caso, o objeto deve ser exibido em estrutura de arame. A representação triangular não apresenta este tipo de problema, pois os triângulos sempre são planares.

## 6. Manipulação das saídas gráficas

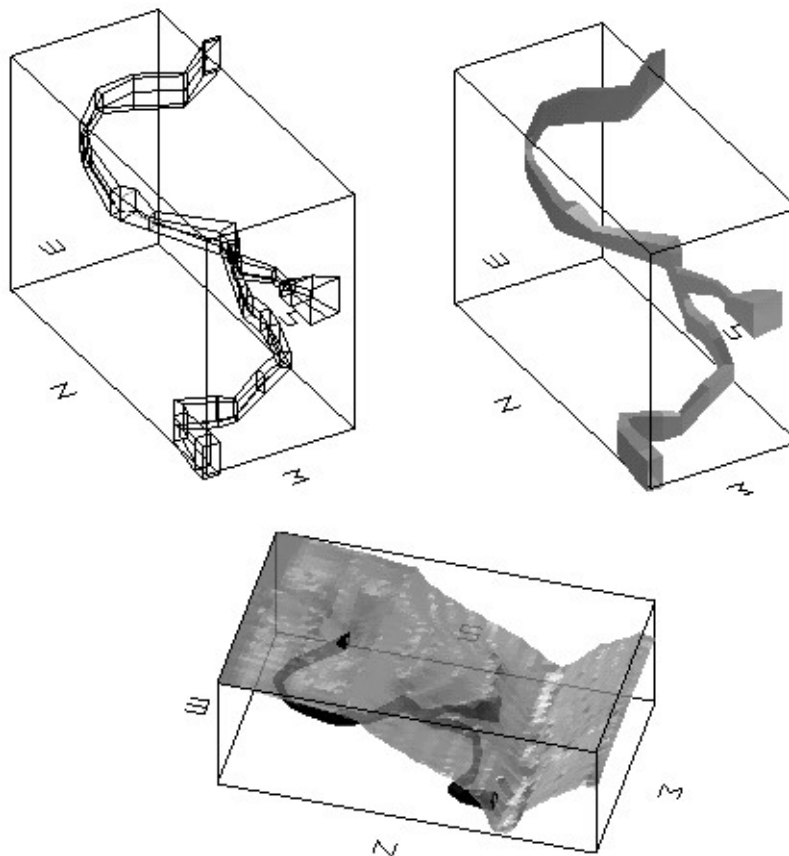
Foram desenvolvidas interfaces para explorar os recursos oferecidos pela API OpenGL. Ao gerar uma saída gráfica, alguns dos atributos das cenas são válidos para todos os objetos visualizados, no entanto, existem atributos que podem ser manipulados individualmente para cada objeto.

A posição do observador pode ser modificada interativamente. Uma interface com posições pré definidas e barras de rolagem permitem que os objetos sejam observados de qualquer ângulo. A cena pode ser rotacionada automaticamente nos eixos x, y ou z, com intervalos angulares variados.

As saídas gráficas geradas pelo programa contam com 3 pontos de iluminação, localizados na superfície de uma esfera imaginária que circunscreve os objetos 3D. Interfaces possibilitam a definição independente dos seus azimutes, inclinações e cores. Cada ponto possui 3 tipos

de luz (ambiente, difusa e especular). Os índices dos componentes RGB de cada tipo podem ser modificados. Estas propriedades são importantes para a composição das cores dos objetos. Neste ambiente gráfico, a cor é definida pela interação entre as propriedades da luz e as propriedades do material que compõem o objeto. No SIS, o usuário também pode alterar as propriedades do material.

As propriedades dos polígonos que compõem os objetos 3D de uma cena podem ser modificadas de forma independente. Numa mesma cena, pode-se observar uma superfície em estrutura de arame e uma caverna solidificada, ou uma superfície sombreada e transparente para permitir a visão do objeto subterrâneo. As paletas de cores associadas a cada objeto 3D também podem ser editadas. A figura 3 apresenta algumas possibilidades de exibição.



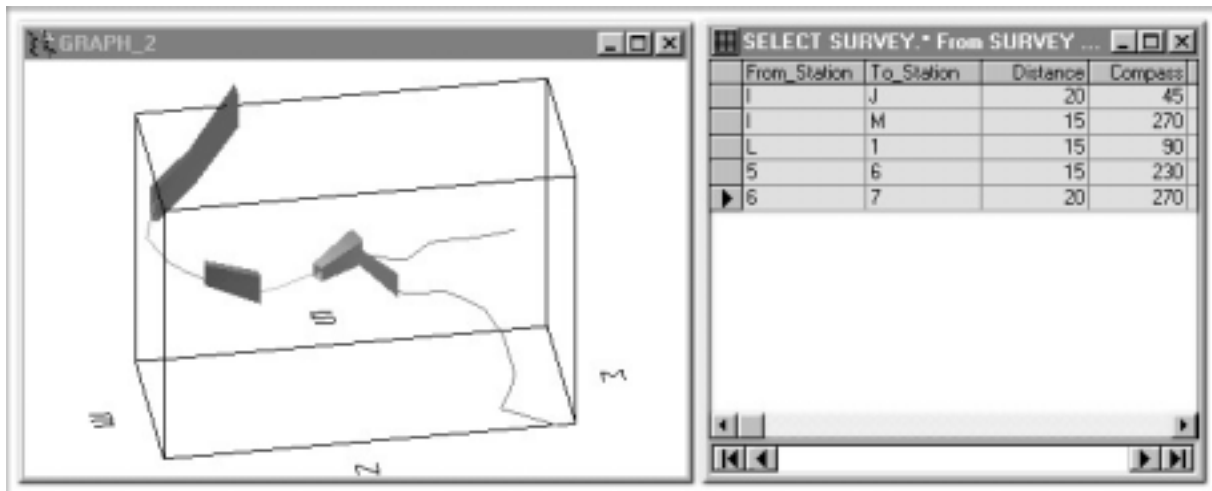
**Figura 3.** Sistema subterrâneo em estrutura de arame, com os polígonos preenchidos e sob uma superfície sombreada e transparente.

## 7. Análise espacial

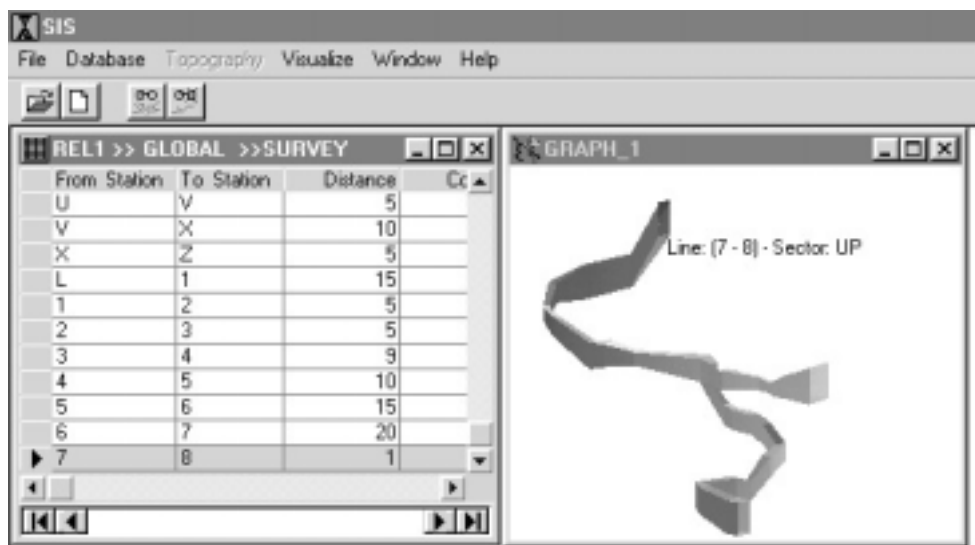
A interface MDI utilizada no SIS permite que um mesmo sistema subterrâneo seja observado simultaneamente em diferentes janelas, com atributos e pontos de vista diferenciados. Esta característica propicia ao pesquisador uma ferramenta de análise por inspeção visual da caverna e das superfícies associadas.

Aliada à interface MDI, a base de dados possibilita a realização de pesquisas a partir de critérios definidos pelo usuário. Um editor foi desenvolvido para a elaboração de

questões estruturadas. Pode-se perguntar ao sistema onde existem depósitos fossilíferos, ou onde existem colônias de morcegos no interior da caverna. Também é possível realizar pesquisas, cruzando várias informações. Existem muitas possibilidades de análise, entretanto, dependentes da estrutura da base de dados desenvolvida para o projeto. Ao executar uma pesquisa, a resposta vem em forma de tabela; o programa permite sua conversão em um mapa, destacando os segmentos que representam a resposta. Na figura 4, podemos observar um mapa resultante de uma pesquisa.



**Figura 4.** Os segmentos da caverna que apresentam volume, correspondem ao resultado da pesquisa à base de dados (segmentos com comprimento  $\geq 15$  metros). As linhas mostram as passagens da caverna que não respondem à indagação.



**Figura 5.** Ao clicar sobre mapa sobre a caverna, o setor é identificado na saída gráfica e na tabela. Caso a tabela contenha textos ou figuras associadas ao segmento, estes documentos podem ser exibidos.



É possível consultar informações relacionadas a determinado segmento da caverna, clicando sobre o objeto 3D. Foram desenvolvidas subrotinas para identificar cada primitiva (polígono ou linha) associada aos objetos 3D. Ao clicar sobre o mapa, o sistema OpenGL retorna o identificador (o número) da primitiva. Um arquivo de acesso aleatório, previamente criado, contém a chave de ligação do identificador com a base de dados. O arquivo é lido, e a partir daí o registro da base de dados é automaticamente posicionado, colocando a disposição do usuário as informações relativas ao segmento selecionado. Ao clicar sobre uma superfície, o valor médio da célula selecionada é encontrado no arquivo de acesso aleatório e exibido na tela. A figura 5 apresenta um exemplo de consulta sobre o mapa.

## 8. Exemplos de aplicação

As aplicações com o uso do SIS estão em seu estágio inicial. O programa vem sendo utilizado em estudos geocientíficos para compreender a evolução e dinâmica de paisagens cársticas.

Na figura 6, podemos observar resultados parciais da representação do sistema Coumo d'Hyouenedo, localizado nos Pirineus franceses. Este sistema possui 94.843 metros de desenvolvimento, e foi mapeado por exploradores franceses. O mapa apresentado resulta de uma pesquisa [9] que vem sendo desenvolvida pelo CNRS.

Um exemplo com a exibição de linhas de visada e superfície do terreno pode ser observado na figura 7. Este mapeamento [10] foi realizado no carste de Itaetê - BA, região da Chapada Diamantina. A superfície do terreno foi obtida a partir de levantamento topográfico, utilizando bússola, trena e clinômetro.

A figura 8 apresenta a distribuição das visadas do sistema Areias (Iporanga -SP). O mapa foi elaborado com o objetivo de compreender a gênese das cavernas [11]. Na mesma figura, podemos observar o modelo digital do terreno da superfície cárstica da região do sistema Areias [12]. O modelo foi elaborado a partir da digitalização de cartas topográficas na escala 1:10.000.

## 9. Considerações finais e perspectivas

Os resultados obtidos mostram que a combinação da API OpenGL com o gerenciador de base de dados *Microsoft Jet database engine*, dentro de uma interface tipo MDI, possibilita o desenvolvimento de sistemas para visualização científica e análise espacial de informações em sistemas subterrâneos.

O programa continua em evolução. Serão desenvolvidos módulos para a análise estatística das informações, módulos para a visualização 3D de

estruturas geológicas associadas ao sistema subterrâneo e o aperfeiçoamento da modelagem geométrica das passagens, visando uma melhor representação da complexidade morfológica dos condutos.

O sistema pode ser adquirido no site [http://www.geocities.com/karst\\_map](http://www.geocities.com/karst_map)

Este projeto foi desenvolvido com o apoio da FAPESP Proc.No 97/6329-0.

Agradeço as valiosas recomendações dos revisores.

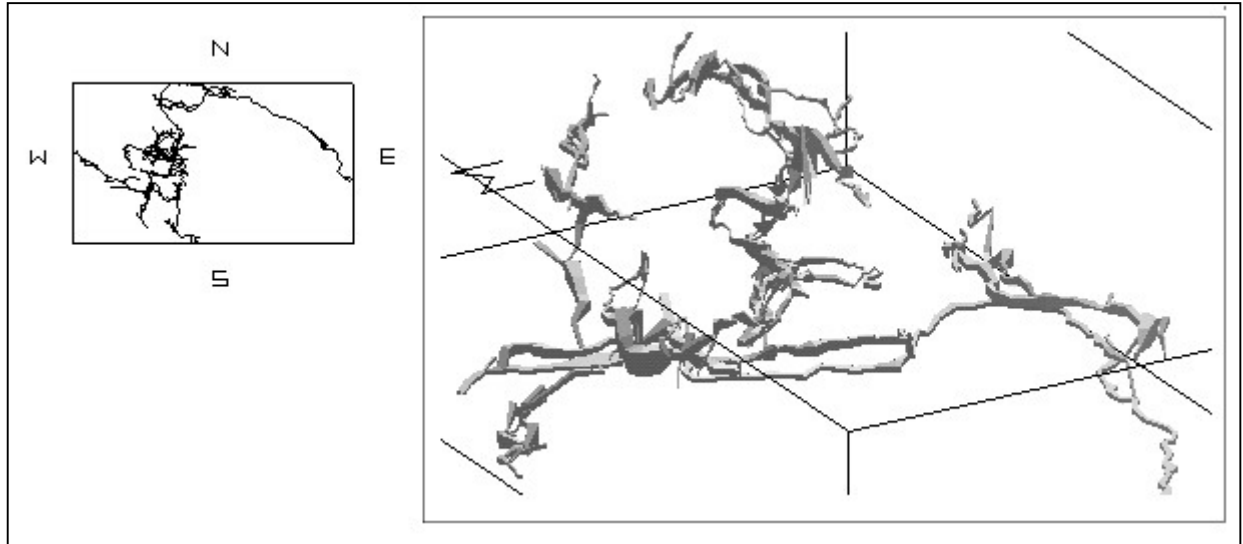
## Referências

- [1] Grossenbacher, Y. (1991) Topographie souterraine. Cours Société Suisse de Spéléologie, 4, 105 p.
- [2] Thomson, K.C & Taylor, R.L (1991) The Art of Cave Mapping. Journal of the Missouri Speleological Survey . Vol. 31 (1-4): 182 p.
- [3] Palmer, A. N. (1981) Cave levels and their interpretation. National Speleological Society Bulletin, 49(2): p. 50-66.
- [4] Ferrari, J.A (1992) Interpretação de feições cársticas na região de Iraquara - Bahia. Dissertação de Mestrado IG - UFBA. 93 p.
- [5] Laureano, F.V (1998) O registro sedimentar clástico associado aos sistemas de cavernas Lapa Doce e Torrinhã, Município de Iraquara, Chapada Diamantina (BA). Dissertação de Mestrado, IG-USP. 98 p.
- [6] Schmidt, V. A. & Schelleng, J. H. (1970) The application of the method of least squares to the closing of multiply connected loops in cave or geological surveys. Bulletin of the National Speleological Society, 32(3): 51-58
- [7] Taillard, E. (1991) Un calcul de boucles en topographie souterraine. Actes du 9º Congrès National de la Société Suisse de Spéléologie, 1991. p. 245 – 247.
- [8] Golden Software (1993) Surfer, Surface Map System, Golden Colorado. <http://www.goldensoftware.com/>
- [9] Viorel, H.(2000) L'influence de la geologie sur le karst. Etude comparative entre la zone d'Arbas (Haute-Garonne, France) et la zone Obarsia Closani - Piatra Mare (Mehedinti - Roumanie). Tese de doutoramento em desenvolvimento.Laboratoire Souterrain du Centre National de Recherche Scientifique
- [10] Pereira, R.G.F..A. (1998) .Caracterização geomorfológica e geoespeleológica do carste da bacia do rio Una, Borda leste da Chapada Diamantina, BA. Dissertação de mestrado, IG- USP, 95 p.
- [11] Souza, M.A. e Karmann, I. (1996) Caracterização morfológica e aspectos genéticos do sistema de cavernas

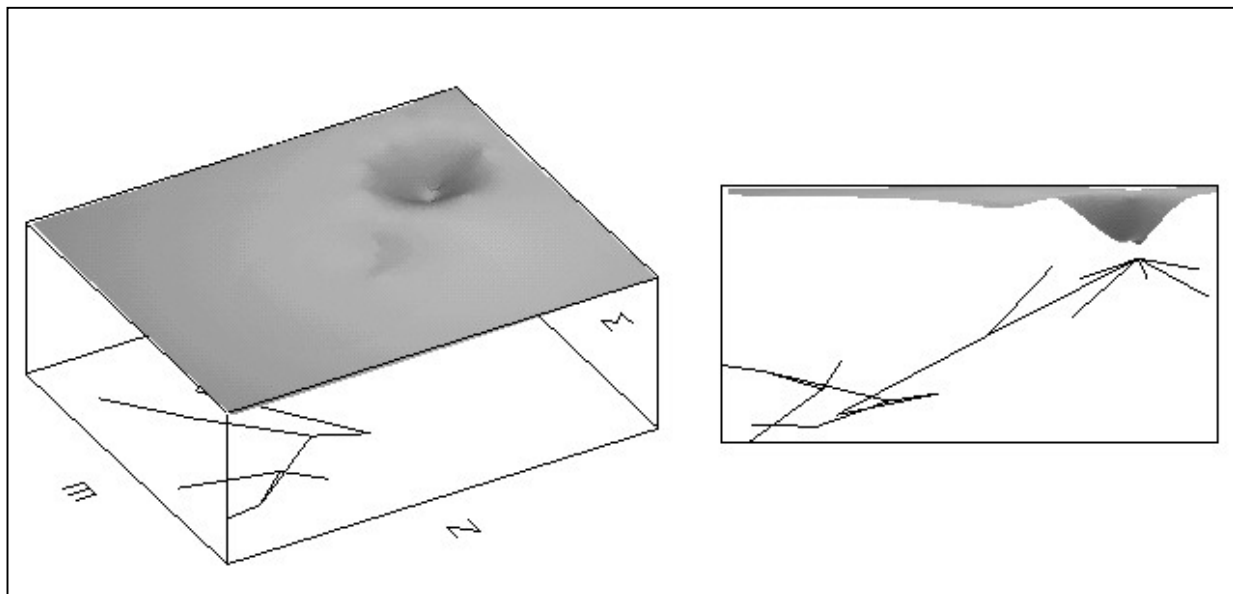
areias, Iporanga, SP. XXXIX Cong. Bras. de Geologia, Anais, vol. 4 pp 586-590.

[12] Genthner, C. (2000) Aplicação do traçador fluorescente Rhodamina-WT no estudo geohidrológico do

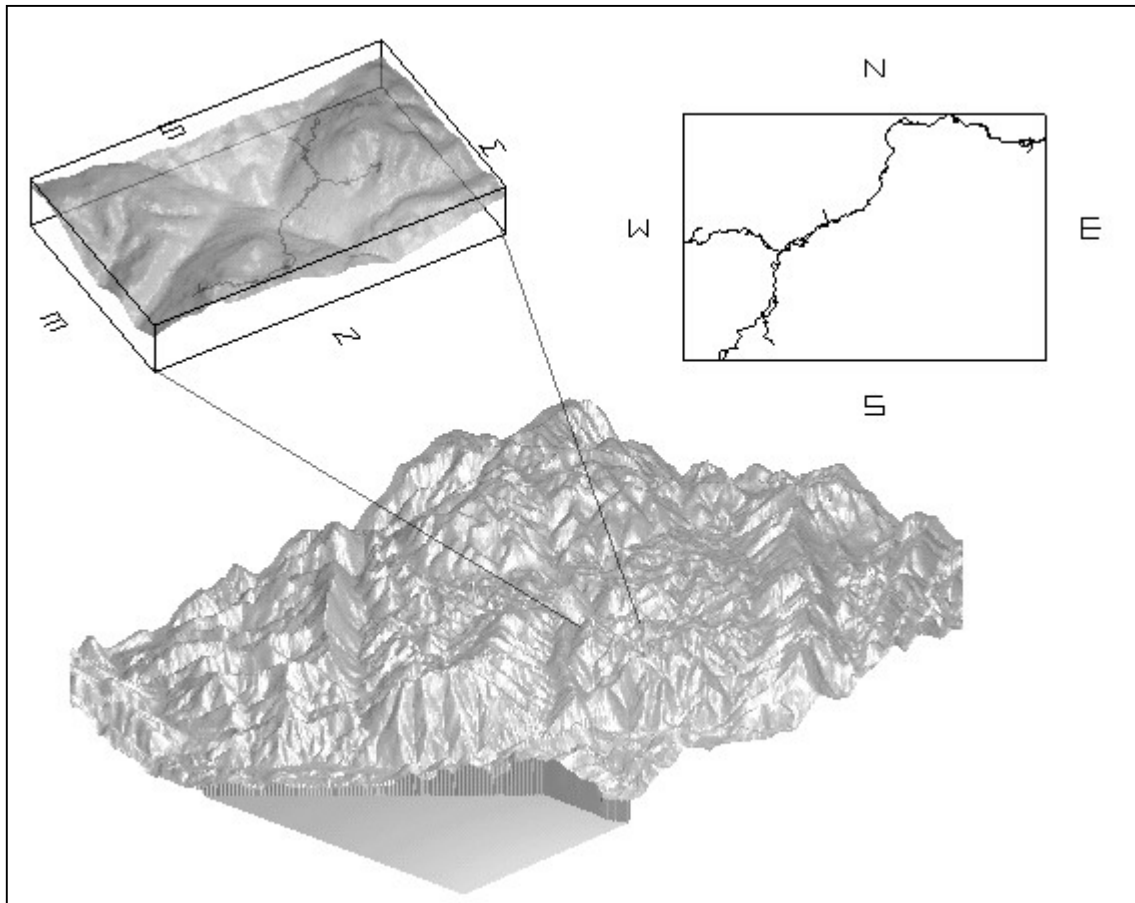
sistema cárstico do Lajeado - Vale do Betari, Iporanga - SP. Dissertação de mestrado em desenvolvimento Instituto de Geociências -USP.



**Figura 6.** Sistema Coumo d'Hyuenedo. Vista em planta e detalhe em 3D.



**Figura 7.** Linhas de visada e superfície do terreno, Gruta do Baixão, Itaetê - BA.



**Figura 8.** Planta das linhas de visada do Sistema Areias e superfícies topográficas.

# Servlets e COM para a Visualização de Dados Geográficos na Web

ARLINDO CARDARETT VIANNA<sup>1,2</sup>  
ANSELMO CARDOSO DE PAIVA<sup>2,3</sup>  
MARCELO GATTASS<sup>2</sup>

<sup>1</sup> Universidade Estácio de Sá - Campus Barra  
Av. Prefeito Dulcídio Cardoso, 2900,  
22631-021 Rio de Janeiro, RJ, Brasil  
[acv@estacio.br](mailto:acv@estacio.br)

<sup>2</sup> PUC-Rio – Pontifícia Universidade Católica do Rio de Janeiro  
TeCGraf – Grupo de Tecnologia em Computação Gráfica  
Rua Marquês de São Vicente, 225  
22453-900 – Rio de Janeiro, RJ  
[{acv,paiva,gattass}@tecgraf.puc-rio.br](mailto:{acv,paiva,gattass}@tecgraf.puc-rio.br)

<sup>3</sup> UFMA – Universidade Federal do Maranhão  
Centro-Tecnológico - Departamento de Informática  
Campus do Bacanga, S/N, 65000-000 – São Luís-MA  
[paiva@ufma.br](mailto:paiva@ufma.br)

**Abstract.** This paper presents an ongoing study on the viability of using Java servlets and COM components for developing dynamic web sites containing geographic data. The maps are presented using TWF (TeCGraf Web Format). The technologies employed are discussed and the context in which they are useful is described.

**Resumo.** Este artigo apresenta um estudo em andamento sobre a viabilidade do uso de Java *servlets* e objetos COM para a criação de páginas dinâmicas que envolvam dados geográficos. Os mapas apresentados utilizam o formato TWF (TeCGraf Web Format). Faz-se uma abordagem das tecnologias utilizadas e apresenta-se uma descrição do contexto no qual elas são úteis.

## 1 Introdução

A popularização da Internet, o advento da *Web* e o crescimento da importância dos Sistemas de Geoprocessamento em uma série de atividades apresentam novas perspectivas para as pessoas que necessitam utilizar dados geográficos. A união desses três fatores torna possível a interação visual com os dados geográficos e uma maior acessibilidade a eles.

O usuário pode solicitar a visualização de mapas e gerar gráficos a partir dos dados geográficos armazenados, os quais podem ser atualizados em tempo real junto com a base de dados. Em razão da expansão da Internet é possível trabalhar com os dados de praticamente qualquer lugar.

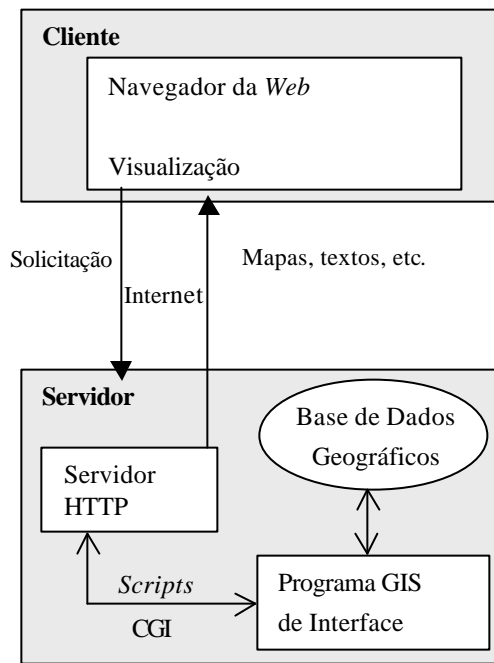
As diferentes maneiras utilizadas para publicar dados geográficos na Internet, em geral, recaem no modelo clássico da Internet (cliente/servidor). Essa arquitetura é composta de um cliente com capacidade de exibição de dados geográficos e um programa de servidor de

geoprocessamento sendo executado juntamente com o servidor *Web* para responder às solicitações sobre os dados geográficos. A Figura 1 representa essa arquitetura. Tipicamente o cliente é um navegador (como Netscape ou IExplorer) que realiza solicitações a um servidor *Web* utilizando o protocolo HTTP (*Hypertext Transfer Protocol*). Quando o cliente realiza a solicitação de um mapa ou outra operação sobre os dados geográficos, o servidor *Web* a repassa para o servidor de geoprocessamento, que atende à solicitação e envia a resposta para o cliente através do servidor *Web*. A comunicação entre os dois servidores é feita, em geral, através de interfaces como CGI (*Common Gateway Interface*), ISAPI (*Internet Server Application Program Interface* – Microsoft) e NSAPI (API do servidor Netscape).

Conforme descrito em Plewe[1], existem algumas maneiras básicas para publicar o dado geográfico na *Web*:

- copiando a base de dados geográficos completa ou parcialmente no cliente;

- exibindo mapas estáticos em páginas HTML (*Hypertext Markup Language*);
- possibilitando ao cliente navegar na base de metadados;
- permitindo a navegação em mapas dinâmicos;
- permitindo ao cliente a realização de consultas e análises sobre a base de dados no servidor, assim como a navegação em mapas dinâmicos.



**Figura 1** Arquitetura Típica para a Publicação de Dados Geográficos na Web.

Entre as opções mais utilizadas para a publicação de dados geográficos na Web podemos citar o SpringWeb [2] MapGuide (AutoDesk) [3], o GeoMedia WebMap (Intergraph) [4] e o MapObjects Internet Map (ESRI) [5].

O SpringWeb é um representante da primeira maneira para a publicação de dados geográficos na Web. Trata-se de um *applet* Java que roda no cliente e precisa da realização de uma cópia local da base de dados, apresentando então um nível razoável de interação com esses dados. Entre as funções disponibilizadas estão operações de visualização como *zoom*, *pan* e navegação sobre os mapas, além da possibilidade de consulta aos atributos de um objeto geográfico específico e a uma tabela com o conjunto de dados de uma determinada categoria de objetos.

O MapObjects Internet Map apresenta como desvantagem principal o fato de publicar os mapas como imagens, em geral nos formatos JPEG ou GIF, e suportar somente as bases de dados nos formatos proprietários da

ESRI. Embora o usuário possa realizar consultas sobre os dados, em razão dos mapas estarem em formato *raster* não é possível navegar através de entidades gráficas.

O MapGuide utiliza um formato proprietário para os mapas a serem visualizados chamado SDF. No entanto, existe uma ferramenta para importar dados de uma série de outras bases, como ArcView e MapInfo, que possibilita a navegação sobre os mapas, mas sem permitir gerá-los dinamicamente.

O Geomedia WebMap apresenta como principal vantagem a capacidade de publicar dados na Web de diferentes formatos, como MGE e ArcView Shapefiles, entre outros. Neste programa os mapas são exibidos no formato ActiveCGM para clientes MSWindows ou como JPEG para clientes em outros sistemas operacionais. Uma de suas principais desvantagens advém da considerável quantidade de programação em *scripts* CGI necessária para construir uma aplicação completa. Esses *scripts* acessam objetos COM desenvolvidos como uma parte da versão completa do Geomedia, os quais realizam as operações sobre os dados geográficos e geram os arquivos para a exibição dos mapas. Esse sistema permite a geração dinâmica de mapas e a navegação através deles.

Todos esses programas têm em comum o fato de trabalharem utilizando CGI, ISAPI ou NSAPI como interface de comunicação entre o servidor de dados geográficos e o servidor de HTTP. Essas interfaces apresentam uma série de desvantagens, que serão discutidas em seguida.

Neste trabalho será apresentado um estudo sobre a viabilidade da utilização de Java *servlets* e componentes COM para o desenvolvimento de *sites* exibindo dados geográficos. Nesse estudo pretende-se utilizar o formato TWF (*Tecgraf Web Format*), desenvolvido por Gattass *et al.*[6], para a exibição dos mapas. A implementação da arquitetura aqui proposta se encontra em desenvolvimento no TeCGraf/PUC-Rio. Inicialmente serão apresentadas as tecnologias utilizadas, destacando suas principais vantagens e a razão de sua adequação para o tratamento do problema. Em seguida é fornecido o contexto de utilização dessas tecnologias para a publicação de dados geográficos na Web.

## 2 Java Servlets

Um *servlet* é um módulo (programa escrito em Java) que é carregado dinamicamente para atender às solicitações de um servidor Web, ou seja, é uma extensão acrescentada ao servidor que aumenta a sua funcionalidade. Os servidores Web respondem às solicitações dos usuários, geralmente, usando o protocolo HTTP através do envio de documentos escritos em HTML.

Em linhas gerais, os *servlets* estão para os servidores *Web* assim como os *applets* (componente de um navegador gráfico e não uma aplicação) estão para os navegadores, com a diferença que os *servlets* não possuem interfaces gráficas.

Os *servlets* podem ser carregados em diversos servidores, pois a API utilizada para escrevê-los usa apenas o ambiente da Máquina Virtual do servidor. O protocolo HTTP é o mais utilizado para a comunicação com o servidor. Os *servlets* podem ser usados para o processamento de formulários (HTML), interagir com bancos de dados, ou de uma maneira geral atuar como uma camada intermediária em uma arquitetura de três camadas, como apresentado esquematicamente na Figura 2.

As principais alternativas para a montagem de páginas dinâmicas são as seguintes: CGI, APIs proprietárias, ASP e *servlets*.

CGI é um padrão de interface utilizado entre servidores HTTP e programas para se comunicarem. O servidor executa um programa (chamado *script*) cujo resultado é então transmitido ao cliente. Normalmente, um *script* CGI gera como resultado uma página HTML para ser exibida no navegador. Os programas CGI são executados na máquina onde está localizado o servidor *Web*. Eles recebem os dados através de variáveis de ambiente e “entrada padrão” e os transmitem de volta ao servidor através da “saída padrão”. Podem ser escritos em qualquer linguagem (C/C++, PERL, TCL, CGI Lua, entre outros). A principal desvantagem de se utilizar *scripts* CGI é que para atender a cada requisição do usuário o servidor *Web* precisa criar um novo processo (cada processo produz uma nova conexão com o banco de dados utilizado e o servidor *Web* tem que esperar até que os resultados lhe sejam enviados), o que resulta numa baixa eficiência. Outro ponto negativo é com relação à segurança, uma vez que os arquivos (*scripts*) não ficam inteiramente protegidos, pois devem ser armazenados nos sub-diretórios CGI-bin do servidor. Mais uma grande desvantagem está relacionada à baixa taxa de reutilização do código, em função das linguagens usadas para a programação. Para criar um novo módulo, a única possibilidade de reutilização é o “copiar-colar”.

Muitos servidores *Web* incluem APIs proprietárias que expandem a sua funcionalidade. Os exemplos mais difundidos são a NSAPI (API do Servidor da Netscape), na qual os programadores podem criar módulos binários que irão acrescentar e/ou substituir elementos para autenticação, autorização ou geração dinâmica de conteúdos, e ISAPI (*Internet Server Application Program Interface* – Microsoft), que tem como princípio básico criar uma DLL que é carregada no servidor quando o HTTP é inicializado e permanece instalada enquanto for necessária. Esta DLL irá gerenciar a conexão com a base de dados sem

a necessidade de se criarem novas conexões, como ocorre com o CGI. O grande problema em adotar tais soluções é que elas são proprietárias, perdendo-se a portabilidade. Outro problema é que geralmente estas aplicações são desenvolvidas utilizando-se linguagens de programação como C ou C++, que podem interferir no bom funcionamento do servidor *Web*.

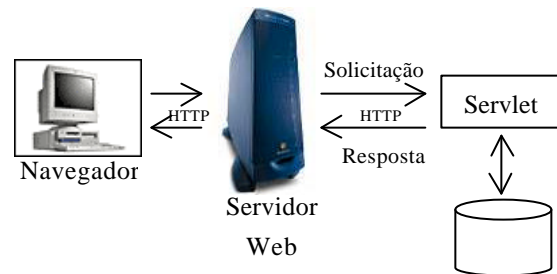


Figura 2 Arquitetura de execução de um *servlet*.

A solução da Microsoft para a criação de páginas dinâmicas é o ASP (Active Server Pages). Ele fica embutido nas páginas HTML, não sendo pré-compilado. O servidor *Web* processa um arquivo utilizando uma DLL (ASP.DLL) que irá interpretar os comandos ASP embutidos na página HTML. O grande problema desta solução é que ela está ligada a um tipo particular de servidor – o Internet Information Server (IIS) – ou a alguma ferramenta que consiga fazer a tradução do ASP antes de enviar a resposta para o cliente.

Segundo Goodwill[7], as principais razões que indicam o uso de Java *servlets* são as seguintes:

- Eficiência: a inicialização de um *servlet* é feita apenas uma vez, ou seja, após ele ser carregado pelo servidor *Web*, as novas solicitações são apenas chamadas do método de serviço.
- Persistência: os *servlets* podem manter o estado entre solicitações.
- Portabilidade: eles são escritos usando Java e, por este motivo, são portáteis, bastando utilizar o JVM (Java Virtual Machine) do servidor, sem fazer uma alteração sequer no código-fonte.
- Robustez: Java possui um método bem definido para o tratamento de erros que minimiza a perda de memória por alocações indevidas (*garbage collector*).
- Segurança: os *servlets* por si só herdam toda a segurança que é peculiar a um servidor *Web*, porém os Java *Servlets* também podem contar com o Java Security Manager.
- Reutilização: como os *servlets* são escritos em Java, eles possuem as vantagens da utilização de uma

linguagem orientada a objetos, como a possibilidade de reutilização.

### 3 TWF

TWF (*Tecgraf Web Format*) é um formato de armazenamento de dados criado para representar um desenho que contenha linhas, regiões, textos e imagens que deverá ser transmitido via *Web*.

O formato de arquivo TWF foi proposto por Ferreira *et al.*[8] com o objetivo de obter um arquivo vetorial que atendessem aos requisitos propostos pelo W3C (World Wide Web Consortium)[9]. Esses requisitos definem um arquivo vetorial com capacidade de organização dos elementos em camadas, exibição de dados *raster*, tratamento de sistemas de coordenadas e possibilidade de interação com o usuário para operações de *zoom*, *pan*, seleção de objetos, tratamento de eventos, entre outras. Essa proposta foi estendida por Gattass *et al.*[6] para suportar a exibição de mapas na *Web*.

Um arquivo TWF é composto de um cabeçalho e uma série de camadas. Cada camada é composta por funções e cada função possui um único identificador e um número variável de argumentos.

A representação de mapas segundo esse formato baseia-se em dois processos: codificação dos elementos geométricos do mapa em um arquivo e interpretação do arquivo.

O processo de codificação realiza as seguintes operações: quantização das coordenadas, simplificação das polilinhas, codificação das coordenadas e compressão. Por outro lado, o processo de interpretação do arquivo consiste de somente dois passos: descompressão e interpretação das primitivas.

Existem duas ferramentas para a geração de arquivos TWF: uma para programadores e outra para usuários finais. A ferramenta para programadores é uma biblioteca de funções escrita em ANSI-C. Para usuários finais foi desenvolvida uma extensão do Geomedia, baseada na tecnologia COM, que exporta dados no formato TWF, mapeando cada *feature* em uma camada do arquivo.

Estão disponíveis dois visualizadores para esse formato de arquivo: um *applet* baseado na plataforma Java 1.2[10] e um *plug-in* desenvolvido em C com OpenGL[11].

Entre as principais vantagens desse formato de arquivo podemos enumerar:

- formato vetorial de arquivo, permitindo operações de visualização no cliente sem acessos ao servidor;
- geração de arquivos menores do que a maioria dos outros formatos disponíveis, conforme demonstrado em Gattass *et al.*[6].

### 4 COM e DCOM

Os componentes de *software* são uma evolução natural no desenvolvimento de *software* orientado a objetos, possibilitando o isolamento de pedaços de uma aplicação em componentes separados, os quais podem ser reutilizados por outras aplicações.

O *Component Object Model* (COM), da Microsoft, (Rogerson[12]) define um padrão binário para a integração de componentes, permitindo que objetos escritos em uma linguagem possam ser acessados a partir de outra.

A motivação da Microsoft para criar o COM partiu da necessidade de se estabelecer um padrão que permitisse que objetos criados por diferentes desenvolvedores pudessem interagir uns com os outros, de forma efetiva.

Esse modelo utiliza todos os conceitos básicos de orientação a objetos: encapsulamento, herança e reutilização. De uma forma geral, um objeto COM responde a uma série de métodos que serão utilizados pelos clientes.

As aplicações interagem umas com as outras ou com o sistema através de um conjunto de chamadas de funções ou métodos, que são conhecidas como interface. Esta deve ser desenvolvida de forma que possa ser reutilizada em vários contextos.

O modelo DCOM (*Distributed Component Object Model*) é uma extensão do modelo COM para suportar a comunicação entre objetos em computadores distintos, seja através de uma rede ou mesmo da Internet. Com o DCOM, a aplicação pode ser distribuída em vários locais de acordo com a conveniência do desenvolvedor e/ou do usuário. É importante ressaltar que o DCOM cuida dos detalhes de protocolos de rede de mais baixo nível, liberando o desenvolvedor para que este centralize seus esforços na busca por soluções mais rápidas e eficientes.

### 5 Arquitetura Proposta

Na arquitetura proposta, esquematizada na Figura 3, temos um cliente (navegador) se comunicando com um servidor *Web* através de requisições no protocolo HTTP, as quais são respondidas pelo servidor com o envio para o cliente de um arquivo HTML com os dados solicitados.

Em um ambiente *Web* para geoprocessamento é desejável que as páginas HTML sejam geradas dinamicamente em uma aplicação (Servidor de Dados Geográficos) integrada ao servidor *Web*. Existem vários mecanismos para realizar a integração entre essa aplicação e o servidor, como já foi visto. Na arquitetura proposta está sendo utilizado o mecanismo de Java *Servlets*. Com ele escrevemos um programa Java que é chamado pelo servidor *Web* para tratar as solicitações enviadas pelo cliente. Entre outras vantagens, os *servlets* possibilitam o acesso às facilidades de interconexão oferecidas por Java e a bases de

dados relacionais através do JDBC, facilidades para reutilização e modularidade e a perspectiva de ganhos de desempenho em razão de não precisar criar um novo processo para atender cada uma das solicitações do cliente. Assim, o *servlet* funcionará como a aplicação que atenderá às solicitações dos clientes, através do servidor *Web*, e gerará dinamicamente as páginas HTML para a publicação dos dados geográficos, exercendo um papel de mediador entre o cliente e os dados geográficos.

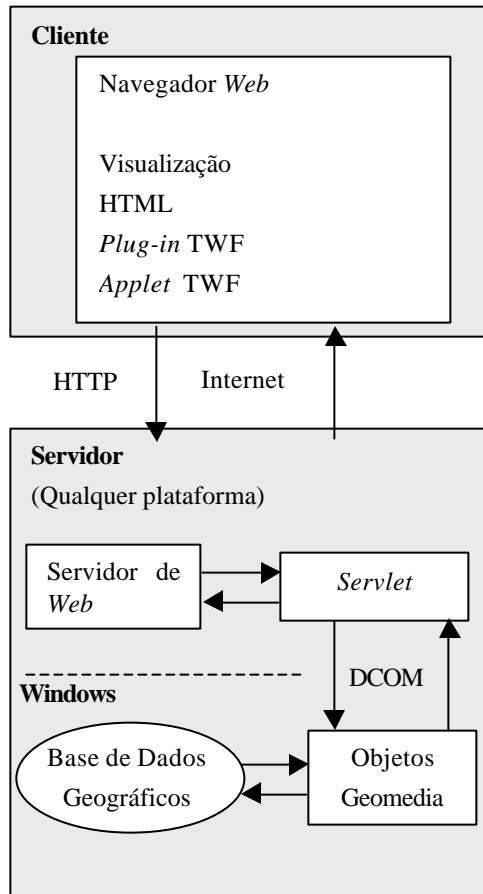


Figura 3 Arquitetura Proposta.

Para acessar os dados geográficos e responder a solicitações como a geração de um mapa ou a realização e exibição de uma consulta espacial ou de uma operação de interseção entre objetos geográficos, o *servlet* precisa de uma biblioteca de objetos que lhe permita realizar essas operações. Essa biblioteca é disponibilizada pelo Geomedia como um conjunto de objetos COM, os quais expõem uma parte de sua funcionalidade. Como principal vantagem da utilização desses objetos podemos citar o acesso a diferentes bases de dados, como MGE e ArcView.

Para a transmissão e exibição dos mapas no cliente foi escolhido o arquivo TWF, por se tratar de um formato vetorial, compactado e que disponibiliza operações de visualização (*pan, zoom, etc.*) no cliente. Além disso, esse formato vetorial é aberto, estando disponível na *Web* tanto a proposta do formato como *plug-ins* e *applets* para visualizá-lo, o que permite ter clientes em qualquer sistema operacional. A biblioteca para a geração desses arquivos, a partir de bases acessáveis pelos objetos Geomedia, está disponível como um objeto COM.

Um importante aspecto nesta arquitetura que está sendo proposta é a comunicação entre o *servlet* escrito em Java e os objetos COM, possivelmente rodando em máquinas distintas. A Microsoft estendeu sua implementação da Máquina Virtual Java para permitir uma integração automática entre Java e COM. Assim, um objeto COM pode ser chamado diretamente de Java e os objetos implementados em Java são disponibilizados como objetos COM. Essa solução possui a desvantagem de exigir que o *servlet* esteja rodando uma Máquina Virtual Java, que não é o padrão.

Na nossa proposta optamos por realizar essa comunicação através do J-Integra[13], uma ferramenta de integração desenvolvida pela Linar Ltda. Ela se comunica com os objetos COM utilizando o DCOM (*Distributed COM*), que se baseia em uma camada de RPC, a qual realiza chamadas TCP/IP para efetivamente completar a comunicação. Em resumo, no nível mais baixo o J-Integra usa as classes padrões de acesso à comunicação em rede oferecidas pela linguagem Java, o que garante a portabilidade da solução.

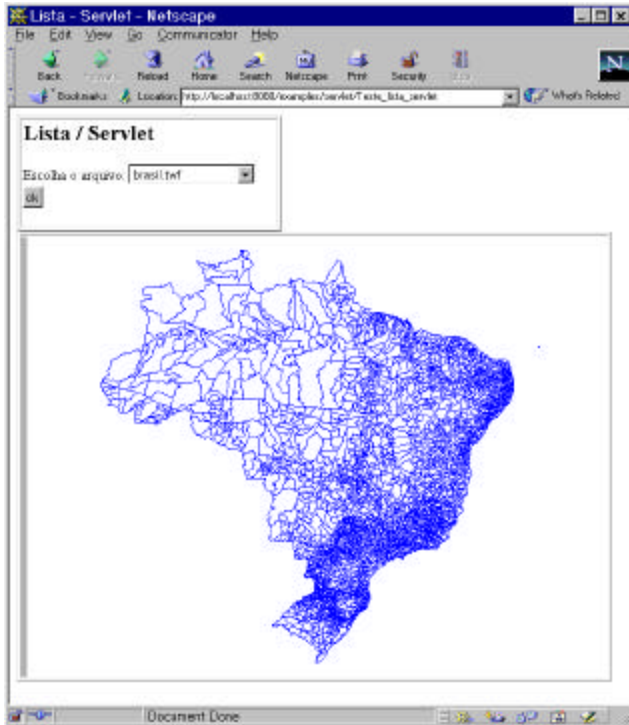
Para o programador do *servlet*, a solução com o J-Integra possibilita que os componentes COM sejam tratados como objetos Java puros, apresentando as propriedades, os métodos e os eventos do objeto COM como propriedades, métodos e eventos Java.

A partir da leitura de bibliotecas que contenham informações das classes e interfaces dos objetos COM, uma ferramenta distribuída com o J-Integra – com2java – gera os *proxies* necessários para a ligação entre COM e Java, ou seja, arquivos que contenham as classes em Java equivalentes aos componentes COM. Podemos entender estas classes geradas como um *wrapper* para os objetos COM. Essas bibliotecas podem estar explicitamente definidas em arquivos com extensão *olb* ou *tlb*, e algumas vezes as informações estão escondidas em arquivos executáveis ou em DLLs.

No desenvolvimento deste sistema já foi implementado um *servlet* que, sem a utilização de objetos COM, gera páginas HTML com mapas previamente gerados no formato TWF. A Figura 4 mostra uma página produzida com esse *servlet*, que possui capacidade de conexão direta com o



banco de dados associado ao mapa para a realização de consultas convencionais a bancos de dados. Atualmente está sendo desenvolvido o empacotamento dos objetos COM em classes Java, utilizando a ferramenta com2java do J-Integra. Isso é necessário para a geração dinâmica dos mapas a partir da base de dados acessável pelo Geomedia.



**Figura 4** Exibição de Mapa em TWF com *servlet*.

### Agradecimentos

Tecgraf é um laboratório financiado principalmente pela PETROBRÁS.

### Referências

- [1] Plewe, B. *GIS ONLINE: Information Retrieval, Mapping, and the Internet*. ONWORD Press, 1997.
- [2] SpringWeb.  
[www.dpi.inpe.br/springweb/springweb.html](http://www.dpi.inpe.br/springweb/springweb.html)
- [3] MapGuide. [www.mapguide.com](http://www.mapguide.com)
- [4] GeoMedia Web Map.  
[www.intergraph.com/software/geo\\_map/](http://www.intergraph.com/software/geo_map/)
- [5] MapObjects Internet Map Server.  
[www.esri.com/base/products/internetmaps/](http://www.esri.com/base/products/internetmaps/)
- [6] Gattass, M.; Ferreira, C. C. F.; Vilar, A. S.; Glasberg, M. "Efficient Map Visualization on the Web". Geoinfo'99, Campinas, São Paulo, 1999.

[7] Goodwill, J. *Developing Java Servlets – The Authoritative Solution*. Sams Publishing, 1999.

[8] Ferreira, C. C. F.; Gattass, M.; Figueiredo, L. H. "Um Estudo Sobre Arquivos Vetoriais para a Visualização de Mapas na Web". GIS Brasil 99 – V Congresso e Feira Para usuários de Geoprocessamento da América Latina, Salvador, Bahia, 1999.

[9] World Wide Web Consortium Requirements.

[www.w3.org/Graphics/ScalableReq](http://www.w3.org/Graphics/ScalableReq)

[10] TWF Applet. [www.tecgraf.puc-rio.br/twf/applet/](http://www.tecgraf.puc-rio.br/twf/applet/)

[11] TWF Netscape Plug-In.

[www.tecgraf.puc-rio.br/twf/plugin/](http://www.tecgraf.puc-rio.br/twf/plugin/)

[12] Rogerson, D. *Inside COM*. Microsoft Press, 1997

[13] J-Integra™. [www.linar.com](http://www.linar.com)

# Uma Metodologia para Geração de Modelos de Elevação a partir de Curvas de Nível

ROBERTO DE BEAUCLAIR SEIXAS

LUIZ HENRIQUE DE FIGUEIREDO  
PAULO CEZAR PINTO CARVALHO

CLAUDIO ANTONIO DA SILVA

IMPA–Instituto de Matemática Pura e Aplicada  
VISGRAF–Laboratório de Visualização e Computação Gráfica  
Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, RJ, Brasil  
lhf, tron, cads, pcezar@visgraf.impa.br

**Abstract.** Modelos digitais de terrenos baseados em elevação são muito utilizados e possuem inúmeras aplicações. Portanto, técnicas que possibilitem a conversão de outras representações para modelos de elevação são extremamente importantes. Neste trabalho, apresentamos uma metodologia em desenvolvimento para geração automática de modelos de elevação a partir de curvas de nível, com uma resolução refinada de acordo com a necessidade do usuário. Além disso, é apresentada uma breve descrição dos problemas envolvidos nesta metodologia.

## 1 Introdução

Há grande demanda por modelos digitais de terreno baseados em elevação (DEM, *Digital Elevation Models*) e portanto é importante a conversão de outros modelos de terreno para DEM. Um grande volume de dados de terreno está disponível na forma de curvas de nível, principalmente os obtidos manualmente a partir de fotos aéreas [2]. O objetivo desse trabalho é estudar algoritmos de conversão de dados de terreno em curva de nível para DEM.

Uma técnica muito usada para conversão para DEM é amostrar as curvas de nível, triangular os pontos amostrais, obtendo uma aproximação linear por partes do terreno, que é então amostrada numa grade regular, gerando um DEM [2, 6]. O principal problema com essa técnica é que as duas amostragens efetuadas introduzem erros. Como resultado, nem sempre é possível capturar todas as feições importantes do terreno. Um efeito disso seria a aparição de abruptas declividades no terreno.

Neste trabalho, propomos estudar o desempenho de algoritmos de interpolação de DEM diretamente a partir das curvas de nível. Em particular, pretendemos focar em algoritmos “naturais”, baseados em evolução geométrica (usando morfologia matemática [1]) ou pseudo-física (usando equações diferenciais parciais [6]), que são os mais adequados a esse tipo de problema, por produzirem superfícies mais suaves. Esse estudo vai usar dados reais de alta resolução, como subsídio para a implantação de um banco de dados de modelos de terrenos cobrindo todo o território nacional, um projeto do Ministério da Ciência e Tecnologia.

O objetivo deste trabalho é descrever um *pipeline* completo para a conversão de dados de curvas de nível, armazenadas em formato digital vetorial, para um modelo de elevação final, com resolução definida pelo usuário. Apesar da etapa de interpolação de alturas ser naturalmente a mais importante nesse *pipeline*, é também importante a



Figura 1: Exemplos de curvas de nível.

interpretação dos dados originais, cuja qualidade e processo de aquisição estão fora do nosso controle.

## 2 Descrição do problema

Discutiremos o problema numa forma mais abstrata de modo a identificar os principais pontos a serem atacados, ainda sem uma formulação rigorosa.

Feita a ressalva acima, os dados do problema podem ser descritos da seguinte forma: É dado um conjunto de curvas poligonais simples e fechadas, representando as curvas de nível de um terreno (Figura 1). Frequentemente, é dado também um conjunto de *pontos notáveis*, com cotas associadas a cada ponto. Algumas das curvas têm *cotas as-*

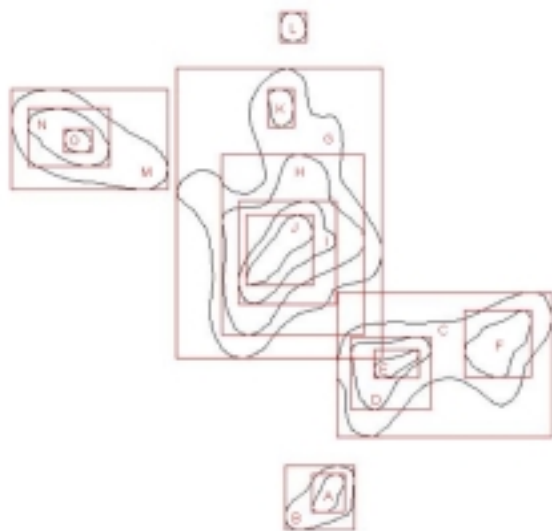


Figura 2: Envoltórias das curvas de nível da Figura 1.

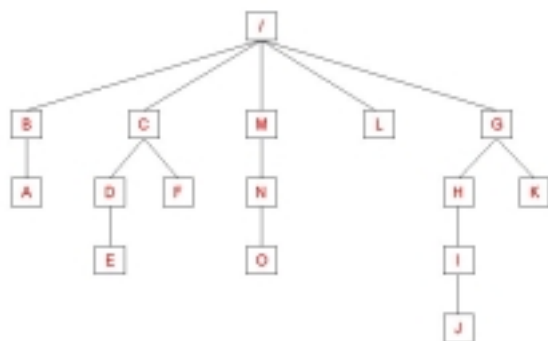


Figura 3: Árvore das curvas da Figura 1.

sociadas a elas, mas nem todas. Tipicamente, um mapa de nível contém curvas correspondentes a níveis de 20 em 20 metros, mas somente as curvas correspondentes aos níveis múltiplos de 100 são explicitamente cotadas.

O problema então é, a partir dos dados acima, calcular a altura de todos os pontos numa grade regular posta sobre o terreno. O modelo de elevação (DEM) que queremos é dado pelo conjunto dos valores dessas alturas, organizados em forma matricial.

Como nem todas as curvas de nível estão cotadas, o primeiro problema que aparece é determinar as cotas das curvas não cotadas. Naturalmente, a intenção de quem fez o mapa é que ele não seja ambíguo, isto é, que seja possível deduzir as cotas que faltam.

### 3 Árvore Topológica de Alturas

Como mencionado anteriormente, a maioria das curvas em mapas de curvas de nível (isolinhas) não possuem informação de altura associada [5]. Contudo, baseado na análise das que possuem tal informação, fornecida através de textos espaçados em intervalos regulares, e no conhecimento prévio do incremento adotado entre as curvas, por exemplo de 20 em 20 metros, pode-se deduzir a cota das curvas restantes. Porém, este procedimento “visual” de interpretar os mapas de curvas de nível não pode ser implementado diretamente no computador.

Para obter uma solução automática, desenvolvemos uma técnica de ordenação hierárquica das curvas baseada nas relações de inclusão das envoltórias do seu contorno (*bounding-box*), como ilustrado na Figura 2. A partir dessa árvore, e da classificação (mãe, filha ou irmã), podemos eficientemente definir as cotas das curvas, mesmo as que não possuem tal informação.

Desta forma, classificamos uma curva *A* como sendo filha, ou seja, interna e de nível superior, de outra curva *B*, quando a curva *A* possui sua envoltória completamente contida na envoltória da curva *B*. Neste caso, dizemos que *B* é mãe de *A* e que *A* é filha de *B*. Quando as curvas não possuem relação de mãe e filha (contido ou contém) entre suas envoltórias, dizemos que tais curvas são irmãs. Após essa classificação, temos as curvas organizadas hierarquicamente em uma árvore de alturas (Figura 3).

É importante observar que devido à estrutura utilizada na construção da árvore topológica, basta que a altura de qualquer nó seja identificada, para que todas as demais alturas sejam determinadas. Por exemplo, podemos dizer que a altura da raiz da árvore se encontra ao nível do mar.

Embora essa técnica pareça ser coerente e exata, o teste de envoltória não é suficiente para classificar corretamente uma curva como sendo filha de outra em todos os casos, como mostra a Figura 4. Assim, verificou-se a necessidade de acrescentar ao teste de filho por envoltória um teste adicional baseado em ponto-em-polígono para garantir uma correta classificação [4, 7].

### 4 Interpolação Morfológica

O principal problema na geração de mapas de altura a partir de curvas de nível é determinar os dados de altura *entre* as curvas. Este fato nos sugere imediatamente a necessidade de algum método de interpolação, de modo que a partir de duas curvas adjacentes possamos determinar a altura exata entre elas. Embora este raciocínio esteja correto, não é qualquer método de interpolação que pode ser utilizado.

Os métodos de interpolação aritméticos mais comuns não podem ser usados visto que não conseguem resolver casos que aparecem frequentemente nas representações por curvas de nível. Como exemplo, considere a Figura 5,

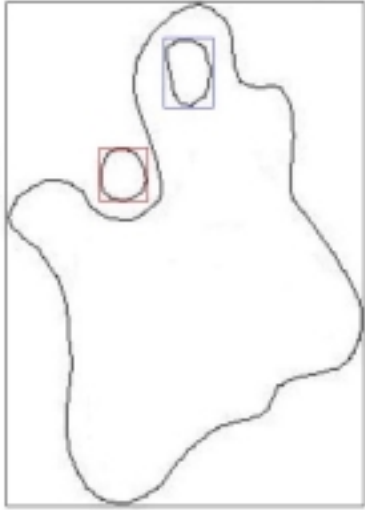


Figura 4: Exemplo de falha no teste de envoltórias.

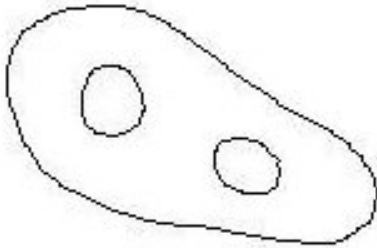


Figura 5: Interpolação aritmética para curvas adjacentes.

onde temos duas curvas adjacentes de mesma altura. A interpolação entre tais curvas seria dada pela média das alturas, ou seja, o resultado seria que o ponto médio entre as curvas teria a mesma altura, o que geraria uma incoerência.

Desta forma, focalizamos nossos estudos em um método baseado em evolução geométrica, usando morfologia matemática [1]. Neste método, as curvas existentes são forçadas a se expandirem naturalmente (usando o operador de dilatação) até colidirem umas com as outras. Na região da colisão, acaba-se por definir novas curvas que são o resultado da interpolação entre as curvas originais, com a altura determinada pela média aritmética das alturas destas curvas. Tal técnica mostrou-se natural e eficiente; os resultados podem ser vistos na Figura 9.

## 5 Detalhes de Implementação

A metodologia adotada para a interpolação de um modelo de elevação a partir das curvas de nível requer as seguintes etapas bem definidas ilustradas na Figura 6.

1. Leitura das curvas de nível;
2. Classificação das curvas lidas;
3. Fechamento das curvas abertas;
4. Inserção das curvas na árvore de alturas;
5. Atribuição de alturas a todas as curvas;
6. Rasterização das curvas;
7. Interpolação das alturas nos pontos intermediários.

A primeira etapa consiste na leitura dos dados do terreno, que no nosso caso estão em arquivos no formato DGN. O formato DGN é um formato proprietário da INTERGRAPH e bastante utilizado em geoprocessamento, visto que pode ser utilizado para armazenar diferentes tipos de dados geográficos. Essa etapa não é trivial pois o formato DGN possui uma estrutura complicada e não é completamente documentado. Apesar disso, a documentação encontrada foi suficiente para lermos os arquivos de teste.

A segunda etapa promove uma filtragem inicial dos dados de modo a classificar as curvas quanto ao aspecto topológico: fechadas, abertas ou ponto cotado. A importância desta etapa é que nem todas as curvas podem ser inseridas diretamente na árvore, o que somente deve ser feito para curvas cujas propriedades topológicas estejam completamente especificadas, o que é válido apenas para as curvas fechadas. As curvas classificadas como abertas são submetidas ao processamento da etapa de fechamento, onde são “completadas” (fechadas) e, em seguida, inseridas na árvore.

Para fins de otimização, a primeira e a segunda etapas ocorrem simultaneamente: os dados são classificados à medida que vão sendo lidos.

Uma vez tendo as curvas classificadas, podemos inserir diretamente na árvore as curvas fechadas, visto que estas possuem suas propriedades topológicas bem definidas, na etapa de ordenação.

As curvas classificadas como abertas, mas não de fronteira, são submetidas a uma etapa de fechamento utilizando a técnica clássica de interpolação de Hermite [3]. Dois exemplos desse fechamento estão ilustrados na Figura 8.

Uma vez fechadas, tais curvas fornecem informação suficiente para que todos os nós da árvore possam ter suas cotas corretamente atualizadas, pois a estrutura da árvore permite que, a partir de uma altura determinada em um nó, se chegue às alturas dos outros nós.

Desta forma, com a árvore completamente preenchida com as curvas, rasterizamos as curvas de modo a poder aplicar a técnica de interpolação de contornos usando morfologia



Figura 6: Principais etapas do processo.

gia matemática, diretamente no espaço da imagem. Com isso, completamos nosso *pipeline* com a etapa final, onde a imagem rasterizada (representação matricial do dado vetorial), constituída por todas as curvas presentes na árvore, é interpolada, gerando com isso um mapa de alturas completamente denso entre as curvas de nível ou com a densidade desejada.

A separação do *pipeline* em etapas bem definidas permite que o desenvolvimento possa ser feito separadamente e somente integrado quando todas as partes estiverem suficientemente testadas.

### 5.1 Problemas Encontrados

- Interpolação linear não é adequada para o problema, conforme discutido anteriormente.
- Curvas de fronteira: Como citado na Seção 3, problemas topológicos, a representação por curvas de nível deve permitir que a altura das curvas não cotadas possa ser deduzida a partir das curvas cotadas. Entretanto, algumas curvas que ultrapassam os limites do mapa podem nos levar a interpretações incorretas dado o seu caráter ambíguo, como mostra a Figura 7. Tais curvas foram denominadas de fronteira e, no momento, não estão sendo tratadas. Esperamos que elas possam ser resolvidas normalmente quando compostas em outro mapa adjacente que a complete.
- Dificuldade de Fechamento: As curvas originalmente cotadas vêm dividida em partes não contínuas, devi-



Figura 7: Problema de ambigüidade das curvas de fronteira.

do a restrições no formato DGN ou à forma como o arquivo foi gerado originalmente. Essa divisão é feita para plotar um *label* correspondente à altura, ao longo da curva. Isso dificulta o processo de identificação de quais partes formam cada curva. Porém, como já citado, as curvas cotadas têm cota múltipla de 100, o que nos permite assumir que elas se encontram entre curvas fechadas (espaçadas em saltos de 20 em 20 metros). Utilizando tal informação, podemos determinar as partes que estão entre determinados saltos e assumí-las como fazendo parte da mesma curva. Assim, basta ordená-las e aplicar a técnica de fechamento.

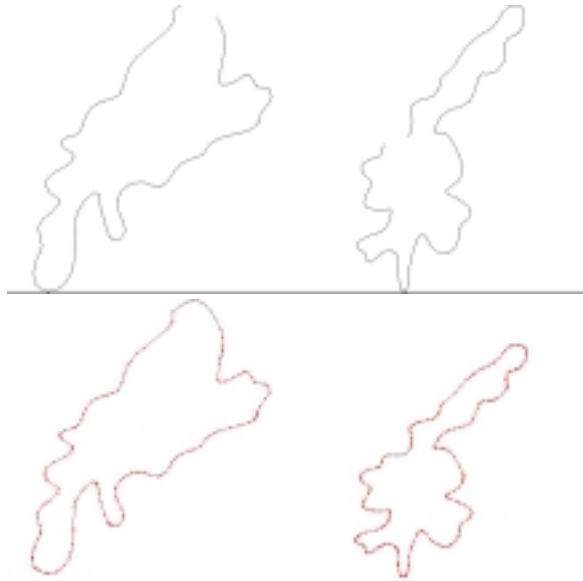


Figura 8: Exemplos de fechamento.

## 6 Conclusão

A técnica de interpolação de contornos baseada em morfologia matemática [1] mostrou-se bastante eficiente e precisa para os dados analisados. A Figura 9 mostra alguns passos no processo de geração de curvas intermediárias utilizando os operadores morfológicos.

Embora o problema em questão seja bem definido—isto é, converter representações de modelos de terrenos baseadas em curvas de nível para modelos de elevação—foi possível comprovar que nem todos os métodos de interpolação são aplicáveis a dados deste tipo, tal como foi discutido anteriormente para os métodos de interpolação aritmética na Seção 4.

Assim, como alternativa, testamos a solução baseada em operadores morfológicos que promove curvas intermediárias mais naturais e coerentes com os terrenos originais. Os resultados obtidos mostraram-se precisos o suficiente para comprovar que a técnica é eficiente e eficaz, como pode ser visto na Figura 9.

Embora os resultados apresentados até o presente momento sejam satisfatórios, é importante salientarmos que o trabalho encontra-se em desenvolvimento e que ainda contará com o tratamento do problema de ambigüidade no processamento das curvas de fronteira. Além disso, é necessário incluir um recurso de ajuste de resolução do mapa de elevação, de modo que o usuário possa escolher a resolução que melhor se adapte às suas necessidades.

Pretendemos também comparar o desempenho do método de interpolação morfológica com métodos de interpolação baseados em equações diferenciais parciais [6]. Para isso, vai ser necessário também desenvolver procedi-

mentos para comparação quantitativa dos resultados obtidos, além da comparação visual.

## Referências

- [1] W. Barrett, E. Mortensen, and D. Taylor. An Image Space Algorithm for Morphological Contour Interpolation. In *Proceedings of Graphics Interface'94*, pages 16–24, 1994.
- [2] G. Câmara, M. A. Casanova, A. S. Hamerly, G. C. Magalhães, and C. M. B. Medeiros. *Anatomia de Sistemas de Informações Geográfica*. X Escola de Computação, 1996.
- [3] G. Farin. *Curves and Surface for Computer Aided Geometric Design*. Academic Press, Inc, 1988.
- [4] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1990.
- [5] M. Garland and P. S. Heckbert. Fast Polygonal Approximation of Terrains and Height Fields. Technical Report CMU-CS-95-181, School of Computer Science, Carnegie Mellon University, 1995. <http://graphics.cs.uiuc.edu/~garland/CMU/scape/scape.pdf>.
- [6] M. B. Gousie and W. R. Franklin. Converting Elevation Contours to a Grid. In *SDH'98, Vancouver*, 1998. <http://www.ecse.rpi.edu/Homepages/wrf/research/p/contour.pdf>.
- [7] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

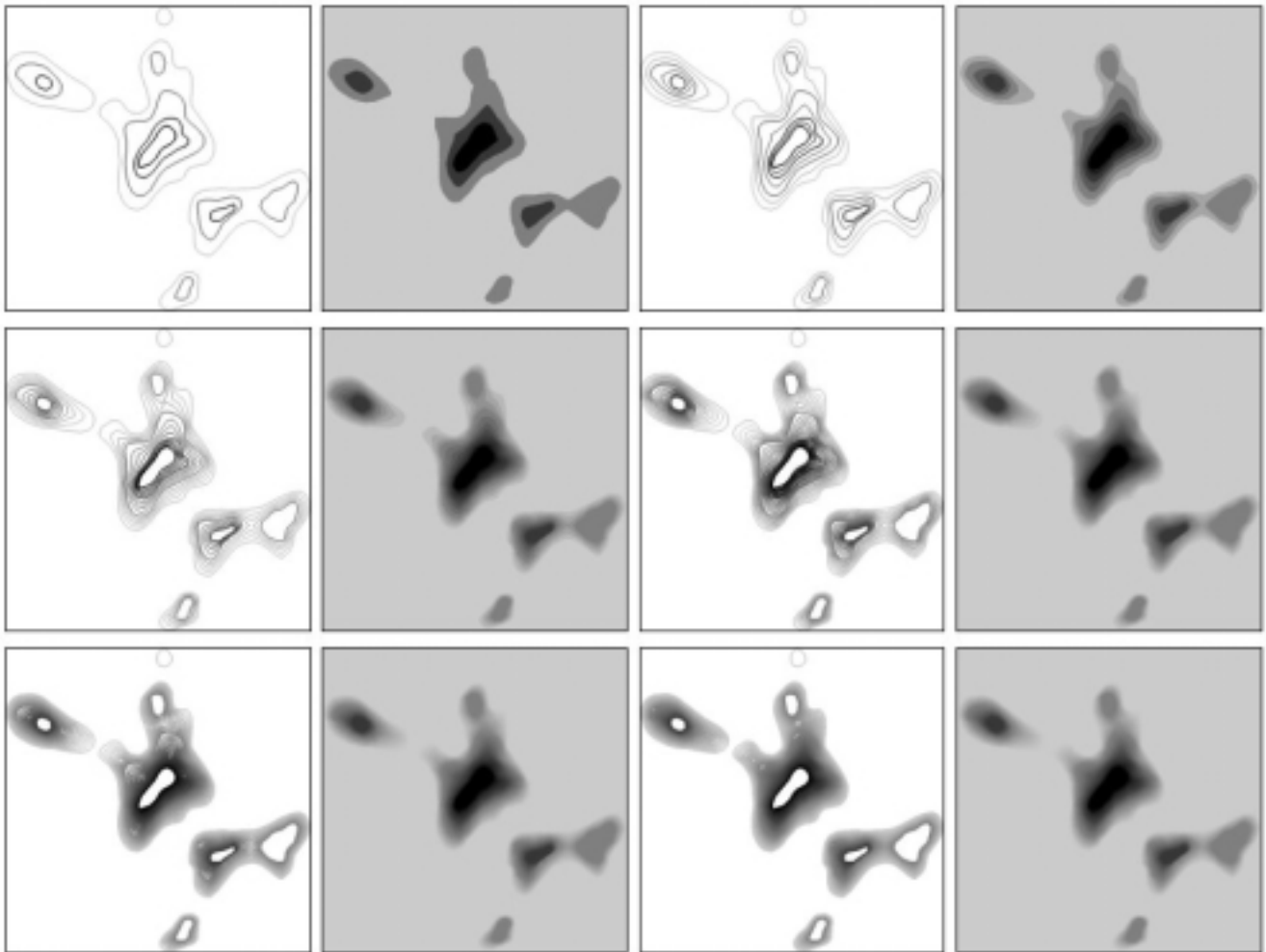


Figura 9: Etapas do resultados.

# Cartografia e Sistemas de Informação Geográfica como Sistemas Semióticos: Uma Análise Comparativa

ALYSSON BOLOGNESI PRADO<sup>1</sup>

M. CECÍLIA C. BARANAUSKAS

CLÁUDIA M. BAUZER MEDEIROS

Instituto de Computação – Unicamp

{alysson.prado, cecilia, cmbm}@dcc.unicamp.br

## Abstract

Several improvements are noticed on Geographic Information Systems, motivated by the general trends in information technologies, expanding their potential uses to domain experts. In that sense, GIS should consider the user's familiarity with Cartography and its traditional way of representing natural phenomena. Understanding the construction and interpretation of maps as communication activities, this paper evaluates the expression power of GIS relatively to their cartographic elements, based on a Semiotic approach. Results obtained from the comparison between Cartographic and GIS semiotic systems show a great difference in the potential for communication for each one of the domains.

## Sumário

Sistemas de Informação Geográfica têm experimentado avanços importantes motivados pelas novas tecnologias de informação, que também têm ampliado seu potencial de uso para além dos especialistas no domínio. Nesse sentido, os SIGs deveriam considerar a familiaridade do usuário com a Cartografia e sua forma tradicional de representação dos fenômenos naturais ou construídos pelo homem. Entendendo a construção e interpretação de mapas como atividades de comunicação, este trabalho visa avaliar o poder de expressão de SIGs no que se refere à representação de elementos cartográficos, a partir da abordagem Semiótica. Resultados obtidos da comparação dos sistemas semióticos da Cartografia e dos SIGs constataam uma grade diferença no potencial de comunicação em cada um dos domínios.

## 1.Introdução

O desenvolvimento da tecnologia de informação na última década tem oferecido oportunidades de melhorar dramaticamente o processo de tomada de decisões e resolução de problemas no domínio geoespacial. Segundo Egenhofer (1999), embora os Sistemas de Informação Geográfica (SIGs) tenham alcançado um crescimento e popularidade sem precedentes na última década, para torná-los uma contribuição a comunidades mais amplas de usuários, mudanças significativas de *design* devem ocorrer. Em especial, uma nova tendência é acoplar SIGs a dispositivos de telefonia móvel e GPS, criando os *Spatial Information Appliances* (SIA), assistentes pessoais do futuro para problemas que envolvem o espaço, movendo a tecnologia do desktop para o trabalho de campo e da mão dos cientistas e profissionais especializados para as dos cidadãos em geral.

Problemas sérios de usabilidade e *gaps* entre tarefas do usuário em implementações de SIG têm sido reportados em literatura recente. Câmara et al. (1999) argumentam que, apesar dos avanços ocorridos na modelagem de dados, é o *design* da interface de SIG que desempenha um papel crucial em determinar a aceitação do sistema. Davis & Laender (1999) discutem a utilidade dos SIGs entre diferentes grupos de usuários, cada qual com seu conjunto particular de aplicações e suas necessidades específicas no que diz respeito principalmente à representação dos objetos.

Ainda segundo Davis & Laender (1999, pág. 36) “a representação de um objeto espacial não determina completamente **sua aparência visual**, ou seja, a forma segundo a qual o objeto será apresentado ao usuário, na tela ou em papel. A cada representação correspondem uma ou mais **apresentações**, alternativas de visualização adequadas para comunicar o significado dos dados geográficos de acordo com as necessidades da aplicação”.

---

<sup>1</sup> Este trabalho tem o apoio da FAPESP (processo 98/15294-8) e PRONEX II MCT - SAI.



SIGs tentam preservar a familiaridade do usuário com a apresentação tradicional do mundo. Dominar o processo de representação da realidade através de mapas há muito tem sido objeto de estudo e pesquisa. A necessidade do homem de compreender o universo que o cerca exige que a cartografia, enquanto “arte de conceber, levantar, redigir e divulgar mapas” (Joly, 1990, pág. 7), seja cada vez mais eficiente.

Baseamos este trabalho na hipótese de que a construção e interpretação de mapas são atividades de comunicação sobre fenômenos e fatos do mundo real, tanto naturais quanto construídos pelo homem. Neste contexto, o objetivo do trabalho é avaliar o poder de expressão de Sistemas de Informação Geográfica no que tange a representação de elementos cartográficos, sob o ponto de vista da Semiótica.

A metodologia empregada para documentar e avaliar os elementos constituintes de mapas foi baseada em uma abordagem semiótica, tendo sido aplicada tanto a mapas produzidos a partir da Cartografia tradicional quanto aos produzidos por um SIG. Inicialmente, foram resgatados e classificados os elementos expressivos do domínio da cartografia tradicional. A seguir, foram avaliados diferentes SIG e a simbologia que estes disponibilizam para a geração de mapas, para verificar se as interfaces de tais sistemas disponibilizam elementos que possibilitem a comunicação cartográfica com eficiência comparável à dos mapas produzidos sem o uso de SIGs. Esta análise permitiu constatar uma grande diferença no potencial de comunicação em cada um dos domínios.

Este trabalho está organizado da seguinte forma: inicialmente apresentamos trabalhos relacionados à representação visual cartográfica, objetivando uma melhor compreensão dos elementos cartográficos. A seguir, apresentamos o referencial de semiótica utilizado para a análise proposta. Na seção 4, analisamos comparativamente os dados coletados dos domínios cartográfico e de Sistemas de Informação Geográfica. Concluindo, discutimos os resultados, mostrando uma possível interpretação e projeções de seu uso no *design* de interfaces de Sistemas de Informação Geográfica.

## 2. Representações visuais em Cartografia

Existe um grande número de trabalhos voltados ao estudo da Cartografia e dos elementos utilizados na produção de mapas, como por exemplo Barkowsky & Freksa (1997), Couclelis & Gottsegen (1997). A seguir, discutiremos trabalhos que se distinguem dos demais pela sua maior preocupação com o processo de percepção e interpretação das representações visuais.

### 2.1. O conceito de variáveis visuais

O trabalho de Bertin (1967) não se resume à Cartografia, mas trata das representações visuais de um modo geral. O autor coleta e analisa diversas simbologias gráficas e, para cada uma, associa propriedades perceptivas que se pode obter com sua utilização. Como resultado, apresenta uma classificação dos dados a serem representados de acordo com sua natureza geométrica: pontos, linhas e áreas, o que foi chamado de **modos de implantação**. Por estar interessado na expressão no plano do papel, Bertin (1967, pág. 42) limita seu estudo a representações bidimensionais, desconsiderando também fenômenos de natureza temporal.

O ponto inicial do seu trabalho é a afirmação geral de que a comunicação é feita por meio de marcas no papel. A apresentação de um dado ocorre através de variações ou modulações de características dessas marcas, como sua forma, posição ou cor. Deste raciocínio surge a lista das variáveis visuais: tamanho, valor (tons de uma mesma cor), granulação, cor (matiz), orientação e forma, além da posição no plano bidimensional. Forma-se, assim, um conjunto de transformações que aplicadas isoladamente ou em conjunto seriam capazes de transmitir visualmente qualquer tipo de informação, respeitando-se as limitações que o próprio Bertin se impôs – bidimensionalidade e atemporalidade. O quadro apresentado na Tabela 1 sintetiza as categorias de variáveis visuais estudadas por Bertin para os diversos modos de implantação.

Para cada uma das variáveis visuais, são determinadas as possibilidades de percepção da natureza que se deseja imprimir aos dados:

- associativa: uma variável visual é dita associativa quando os dados por ela representados podem ser agrupados de acordo com outras categorias, independentemente das variações provocadas por esta variável. Ex: ao representarmos em um mapa duas informações a respeito de cada cidade, como população através do tamanho da mancha e atividade econômica principal através da forma (círculos para agricultura, quadrados para indústria, triângulos para outros), seremos capazes de visualizar a distribuição populacional independentemente da atividade econômica, identificando regiões com manchas maiores ou menores. Portanto a forma é uma variável associativa, suas variações podem ser tratadas de forma associada quando se analisa a outra variável (tamanho). Uma variável não associativa é dita dissociativa.
- seletiva: permite isolar espontaneamente todos os elementos pertencentes a uma mesma categoria, dentro do conjunto total dos signos representados.

Ex: ao vermos um conjunto de círculos de diferentes cores somos capazes de separar visualmente todos os de uma só cor dos demais. A cor é portanto uma variável seletiva.

- ordenada: uma variável é ordenada quando se perceber uma seqüência natural nos dados apresentados. Ex: diversos tons de cinza (valor), indo do mais claro ao mais escuro, podem ser percebidos como uma seqüência. Logo, valor é ordenado.
- quantitativa: uma variável é quantitativa quando é possível atribuir um valor ao elemento representado a partir da sua representação. Ex.: se uma figura é duas vezes maior que outra (tamanho), podemos dizer que a primeira representa um dado que tem duas vezes o valor da segunda.

As variáveis visuais podem ou não conseguir transmitir todas as propriedades perceptivas. A Tabela 2 indica o resultado dos estudos de Bertin, atribuindo a cada variável visual as propriedades perceptivas que ela permite.

Certas formas de representação não permitem detectar determinadas características dos dados (itens em branco na Tabela 2). Por exemplo, o uso da cor (matiz) não é capaz de traduzir o caráter quantitativo de uma representação, pois não existe uma relação natural entre matizes e quantidades na mente do leitor de um mapa. Uma associação desse tipo só seria possível mediante a criação arbitrária de uma escala de cores ou legenda (Monmonier, 1991, pág. 150).

Variáveis Visuais	Modos de Implantação					
	Pontual		Linear		Zonal	
Posição no plano						
Tamanho						
Valor	(tom claro) 	(tom escuro) 	(claro) 	(escuro) 	(claro) 	(escuro) 
Granulação						
Cor	(azul) 	(vermelho) 	(azul) 	(vermelho) 	(azul) 	(vermelho) 
Orientação						
Forma						

Tabela 1: Classificação das variáveis visuais de Bertin

Variáveis Visuais	Propriedades Perceptivas			
	Quantitativa	Ordenada	Seletiva	Associativa
Posição no plano				
Tamanho				
Valor				
Granulação				
Cor				
Orientação				
Forma				

Tabela 2: Propriedades perceptivas das variáveis visuais

Embora exista um esforço no sentido de criar condições para que o leitor do mapa seja capaz de identificar os diferentes elementos ou perceber seqüências nos dados representados, a abordagem de Bertin não leva em consideração o **significado** que certas variáveis, como forma e granulação, podem sugerir ao leitor.

## 2.2. O conceito de alfabeto cartográfico

Ramirez (1993) coleta e analisa dados do domínio cartográfico visando identificar os elementos básicos com os quais mapas são construídos. Para isso, baseando-se na teoria de níveis lingüísticos de Chomsky (1972), decompõe sucessivamente e isola os componentes de mapas até obter um conjunto básico de elementos. A este conjunto (Figura 1), deu o nome de **alfabeto cartográfico**. A seguir, definiu um conjunto de regras de utilização deste alfabeto, o que chamou de gramática cartográfica.

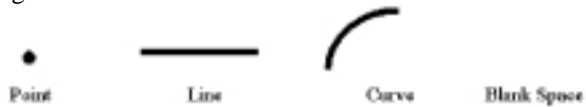


Figura 1: o alfabeto cartográfico de Ramirez

As regras da gramática levam a elementos semelhantes às variáveis visuais de Bertin, ou seja, são modulações a serem aplicadas sobre o conjunto básico de primitivas (alfabeto), através dos quais seriam obtidas as possibilidades de representação para a construção de mapas. Sobre um mesmo elemento original - digamos, uma linha - podemos combinar variações de cor, granulação (tracejado, por exemplo) ou tamanho (espessura) e com isso progressivamente construir mapas.

Ramirez considera os mapas como sendo **sentenças**, conjuntos de elementos relacionados de modo a conter um significado específico: a descrição de uma realidade geográfica. Embora esta abordagem sirva ao seu objetivo de permitir automatizar a construção de mapas ao isolar seus elementos mínimos, a granularidade da decomposição atingida é alta, gerando elementos por si desprovidos de significado, o que não dá margem ao estudo dos fenômenos de comunicação que existem quando se constrói ou se interpreta um mapa.

## 2.3. Semântica cartográfica

Analogamente à abordagem de Ramirez, Pratt (1993) afirma existir muito em comum entre mapas e outras formas de comunicação como linguagens naturais escritas. Ambas as formas de expressão utilizam marcas de tinta sobre o papel, através de um código preestabelecido. Entretanto, ressalta que o que diferencia os mapas é a sua característica geométrica e visual, o que

os aproxima dos objetos que se deseja representar no sentido de se utilizar das características de percepção destes para representá-los.

A análise feita por Pratt tem como objetivo a determinação da **semântica cartográfica** (Map Semantics), ou seja, obter um conjunto de regras de validação do conteúdo de um mapa. Para tanto, lança mão da definição de Schlichtmann (1985), em que cada elemento cartográfico pode ser decomposto em dois componentes: o “substantivo” e a “localização”. Ao primeiro cabe transmitir a natureza do objeto representado (citando os exemplos do próprio Pratt, uma casa, uma igreja ou um poço) através de suas propriedades visuais (ainda seguindo o exemplo,  $\square$ ,  $\dagger$  e  $\circ$ , respectivamente). Já o segundo transmite a localização do objeto no mundo real, com base na localização no mapa do elemento que o representa.

Para que o leitor de um mapa seja capaz de interpretar os pares substantivo e localização  $X+(x,y)$  presentes no mapa, é preciso decodificar a realidade geográfica expressa por tais pares. Para isso, Pratt considera necessárias duas funções: uma função de **interpretação simbólica**  $I(X)$ , que correlacione cada elemento  $X$  do domínio cartográfico com o conjunto de entidades da realidade geográfica que se quer representar e uma função de **interpretação espacial**  $\mu(x,y)$  que correlacione os pontos da superfície  $(x,y)$  do mapa com posições no espaço. Assim, a função  $I$  mapeia  $\square$  com o conjunto de casas,  $\dagger$  com o conjunto de igrejas, e assim por diante, enquanto  $\mu$  informa suas respectivas localizações (Pratt, 1993, pág. 80).

A determinação da validade de cada elemento  $X+(x,y)$  de um mapa ocorre ao verificarmos a existência no mundo real de seu correspondente  $I(X)+\mu(x,y)$ . Pratt prossegue no sentido de determinar as condições de validade do mapa como um todo, analisando as implicações da ausência de símbolos em determinada região do mapa, a ambigüidade figura/fundo (*figure/ground*). O próprio Pratt, entretanto, indica que não se aprofunda na descrição da função de interpretação  $I(X)$ , deixando de considerar formas de representação gráficas como cor e textura (como as variáveis visuais de Bertin), pois para seu objetivo é suficiente conhecer a existência de  $I(X)$ . No entanto, se pretendemos estudar o processo comunicativo de produção e interpretação de mapas, é justamente nesta função que devemos nos concentrar.


## 3. A Cartografia como um Sistema Semiótico

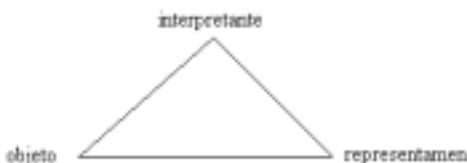
A ciência cartográfica sempre esteve atenta quanto à necessidade de utilização de elementos providos

de significado, como nos indica Oliveira (1988, pág. 109):

*Um símbolo cartográfico [...] não pode abdicar, inteiramente, do seu caráter figurativo associativo, em favor do símbolo geométrico puro.[...] Um mapa não é - não se pode permitir que seja - um diagrama meramente geométrico, em que as distâncias e as relações horizontais estejam corretas; deve, até certo ponto, sugerir a aparência do assunto.*

Nosso trabalho adota a teoria semiótica como referencial para análise de representações cartográficas, pela clareza com que trata as relações de significação entre os elementos representantes e representados. A Semiótica se ocupa do estudo dos signos. Um signo é “algo que está para alguma coisa para alguém” (Peirce, 1990, pág. 46; Santaella, 1996). Peirce, um dos teóricos cuja obra serve de embasamento para este trabalho, define o signo como sendo uma relação triádica entre um objeto, um *representamen* e um interpretante ( Figura 2). Objeto é a parte do signo a qual se quer referenciar (por exemplo, a existência de uma árvore em um determinado local). Já o *representamen* é a entidade que efetivamente é utilizada na tentativa de comunicação, com a intenção de representar o objeto (no mesmo exemplo, dado um mapa em papel do local, a presença de marcas de tinta na

forma  ou mesmo simplesmente um × podem ser *representamens* da existência e localização da árvore). O processo de significação ou **semiose** ocorre quando, ao se apresentar o *representamen* a uma mente pensante, nela é produzida uma idéia que a remete ao objeto, idéia que é chamada **interpretante**.



**Figura 2: O signo de Peirce**

Uma classificação geral dos signos proposta por Peirce os divide em ícones, índices e símbolos, de acordo com o tipo de relação existente entre o *representamen* e o objeto. A importância do estudo dessa relação, conforme salienta Santaella (1989, pág 58), está no fato de ela ser a

*base fundamental para se considerar o nível ou grau de interpretabilidade do signo, isto é , seu potencial interpretativo.* A classificação de Peirce é apresentada a seguir:

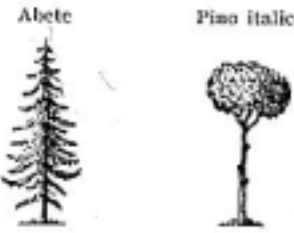
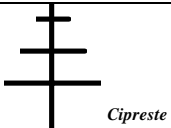

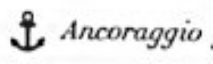
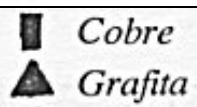
- ícone: a representação se dá de uma forma direta, por semelhanças das características perceptivas entre objeto e *representamen*;
- índice: o *representamen* se associa ao objeto por uma relação natural de pressuposição ou dependência;
- símbolo: a associação é arbitrária, estabelecida socialmente ou de forma imposta.

Além desta primeira classificação, mais divulgada na literatura, Peirce propôs uma sub-classificação para os ícones (Peirce, 1972), cujo *representamen* passa a ser denominado **hipo-ícone**, e é classificado em:

- imagem : reproduz as qualidades características do objeto;
- diagrama: tem sua representatividade baseada nas relações entre as partes do objeto referenciado;
- metáfora: expressa “o caráter representativo de um *representamen*” (Peirce, 1972, pág. 117) através de um paralelo com algo diverso.

A Tabela 3 ilustra esta taxonomia, através de exemplos que extraímos da própria Cartografia. A seta no eixo vertical indica o sentido em que tanto a capacidade de significação (ou seja, seu poder representativo) como a especificidade de um signo aumentam.

Imagens são signos com um grau maior de especificidade, ou seja, podem ser utilizados como *representamens* de um conjunto menor de objetos. Além disso, estes signos carregam em si um maior poder de significação, pelo fato de sua relação com o objeto representado ser mais perceptível e imediata - são “auto-referentes” (Santaella, 1989, pág. 57). Caminhando em direção ao extremo oposto, temos signos cada vez mais desprovidos de relação direta entre *representamen* e objeto, sugerindo progressivamente uma interpretação menos imediata. Por isso, são signos com um grau de especificidade menor, permitindo que a eles seja associada uma gama maior de significados.

↑ significação/especificidade	Ícone	imagem		Extraído de Agostini (1972), tabela VIII
		diagrama		Agostini (1972), tabela VIII
		metáfora		Oliveira (1988), pág. 110
	Índice			Agostini (1972), tabela XIV
	Símbolo			Martinelli (1991), pág. 58

**Tabela 3: Uma ilustração da taxonomia de signos de Peirce em elementos cartográficos**

Esta classificação peirceana de signos pode ser usada como forma de traçar um perfil de um **sistema semiótico**, ou seja, de um conjunto de signos em um contexto em que possuem uma certa inter-relação e assumem uma determinada gama de significados.

Definimos **compatibilidade entre sistemas semióticos** como uma relação associativa baseada na distribuição que os signos neles presentes apresentam no eixo acima citado. Dois sistemas semióticos seriam compatíveis, ou seja, possuiriam grau de interpretabilidade equivalente para um mesmo perfil de indivíduo interpretador, se os elementos expressivos que o constituem utilizam proporcionalmente as mesmas formas de significação.


#### 4. O estudo dos sistemas semióticos da Cartografia e de SIG

Com base na taxonomia de signos apresentada na seção anterior, procedeu-se a coleta e classificação dos dados referentes aos dois sistemas semióticos analisados. Foram utilizados livros e publicações voltadas para o ensino de Cartografia, principalmente aqueles editados antes do surgimento de SIG, procurando um maior grau de independência entre os dados dos dois sistemas semióticos aqui estudados.

Em paralelo, foi feito o mesmo trabalho de análise sobre interfaces de três Sistemas de Informação

Geográfica, para identificar os conjuntos de signos disponibilizados para o usuário durante o processo de confecção de mapas. Foram usados os SIG ArcView 3.0 (<http://www.esri.com>), Idrisi 2.01 (<http://www.idrisi.com>) e Spring 3.2 (<http://www.dpi.inpe.br/spring>).

O método adotado para classificação dos signos em cada um dos sistemas semióticos estudados baseou-se no seguinte procedimento: na Cartografia pré-computacional encontramos a relação *representamen-objeto* explícita em descrições didáticas de como construir mapas e representar acidentes geográficos, vegetação, construções humanas, etc., bastando classificar esta relação de acordo com a taxonomia já descrita. Já para o conjunto de signos dos SIG dispúnhamos apenas do *representamen* não existindo indicação, na interface dos softwares, dos objetos com os quais o *representamen* estaria relacionado. A determinação da relação *representamen-objeto* nesse caso considerou duas possibilidades: em primeiro lugar, nos casos em que existe elemento semelhante no domínio cartográfico, supomos para o signo do SIG a mesma classificação dada ao da Cartografia. No caso da inexistência de um elemento correspondente, procuramos identificar um conjunto de possíveis objetos com os quais o *representamen* poderia estar relacionado. A classificação foi feita com base na relação *representamen-objeto* com o maior grau de especificidade (mais próximo de imagens).

Por exemplo, o signo  do SIG ArcView poderia referenciar apenas a presença de um hidrante (ícone), como o prédio do Corpo de Bombeiros, ou algo para combate a incêndio (índice). Optou-se pela primeira interpretação, pois ícones são de interpretação mais direta que índices (Tabela 3).

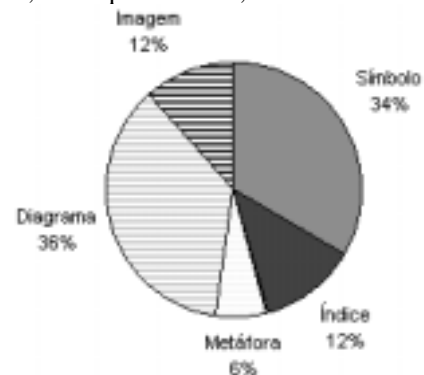
Em alguns dos SIG analisados é dada ao usuário a possibilidade de alterar e personalizar o conjunto inicial de signos, como os padrões de preenchimento de áreas no Spring, por exemplo. Isso não foi levado em consideração em nossa análise; restringimo-nos somente aos elementos que por *default* estão disponíveis ao usuário, pois a personalização não é disponível em todos os *softwares*, e mesmo em um mesmo software podemos não conseguir fazê-la em todos os elementos (ponto, linha e área).

Apesar da importância das cores na construção de mapas, já que são largamente utilizadas para significação, como nos informa Oliveira (1988, págs. 109-111) ao sugerir o uso de verde para vegetação, azul para hidrografia e castanho para relevo, seu uso não foi considerado neste trabalho. No passado, a produção de mapas era limitada quanto ao uso de cores por dificuldades técnicas na impressão (Monmonier, 1991). Já os *softwares* analisados disponibilizam igualmente todo o espectro visual (RGB), não se diferenciando da Cartografia tradicional. Quanto ao uso adequado das escalas de cores que são disponibilizadas pelos SIG, tal problema já foi abordado pelo trabalho de Bertin (1967).

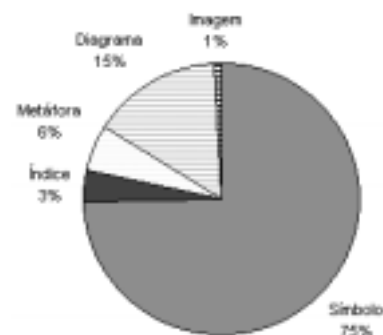
Foram identificados e classificados 161 signos da literatura cartográfica e 143 dos SIG, e a partir destes dados gerada uma série de gráficos. O primeiro (Figura 3) corresponde aos resultados da análise e classificação dos dados relativos à Cartografia tradicional. Podemos notar a predominância de diagramas (36%) e símbolos (34%), demonstrando uma distribuição equilibrada entre os extremos do eixo. Ainda se agruparmos todo o conjunto de ícones, obtemos mais da metade dos signos (54%) utilizando para representação as características primárias dos objetos representados, ou seja, neste sistema semiótico são frequentes as representações mais imediatas, por meio das próprias características dos objetos representados.

Já no conjunto de resultados da análise de SIGs (Figura 4) encontramos outra distribuição, com absoluta maioria de elementos classificados como símbolos (75%). Isso revela o caráter generalista deste sistema semiótico, por possuir signos de baixa especificidade. Tais signos são providos de pouco poder de significação, não carregando em si qualquer informação além daquela que lhe é atribuída por convenção. Com isso, torna-se menos

imediate a interpretação dos elementos que constituem o mapa e, conseqüentemente, a leitura deste como um todo.



**Figura 3: Gráfico da distribuição de signos da Cartografia**



**Figura 4: Distribuição dos signos coletados em SIG**

## 5. Discussão e conclusões

O presente trabalho contribui ao entendimento de representações visuais em Cartografia, como os apresentados na seção 2, ocupando-se do potencial de significação de representações cartográficas. A abordagem focaliza a **interpretação** que o leitor de um mapa pode fazer de seus elementos visuais.

É desejável que a aparência visual dos componentes do mapa seja adequada ao uso e à categoria de usuário da aplicação. Na produção cartográfica convencional, o processo é baseado no conhecimento empírico do cartógrafo, incluindo seu senso estético. A análise comparativa dos sistemas semióticos da Cartografia e de SIGs mostra que o segundo ainda é tecnocêntrico, o que pode ser explicado pela herança das restrições iniciais da tecnologia de interfaces gráficas (palhetas genéricas de símbolos ocupam menos espaço e servem a um número maior de objetivos na criação de mapas no computador).

Sistemas de Informação Geográfica podem ser usados para produzir infinitas interpretações do mundo

real e apresentá-las como mapas. Neste caso, uma parte da interface é dedicada a permitir aos usuários definir sua interpretação (ativando módulos de análise espacial) e a segunda parte à apresentação cartográfica. Poderíamos ser levados a pensar que limitações de escolha dos signos para construção de mapas deveriam ser impostas para não sobrecarregar a interface e confundir o usuário leigo. É fundamental termos em mente que o perfil dos usuários de SIG tem se deslocado do especialista em computação para o especialista em outros domínios do conhecimento que necessitam manipular dados espaciais. Assim, apesar de o usuário de SIG poder ser alguém leigo em Cartografia – tanto a tradicional quanto a digital – ele toma também a posição de usuário do mapa que será produzido. Ou seja, a análise, processamento e apresentação dos dados geográficos deixam de ter a intermediação do especialista em computação e em SIG. Portanto, torna-se desejável que esse SIG disponibilize ao usuário elementos que facilitem a construção de um mapa cuja futura leitura e interpretação para tomada de decisões seja o mais natural e imediata possível.

É importante destacar que, embora o sistema semiótico da Cartografia tenha apresentado resultados mais favoráveis à interpretabilidade dos signos constituintes dos mapas do que os SIGs, a informação geográfica no computador pode ser muito mais rica do que a informação cartográfica convencional, se exploradas as possibilidades da mídia do computador, e não permanecendo meramente uma duplicação digital de mapas já existentes. A produção de mapas via SIG é, assim, um passo adiante da cartografia eletrônica “padrão”, que se limita a reproduzir mapas previamente criados usando a cartografia tradicional. Estes sistemas permitem a construção de mapas a partir de dados brutos observados no mundo e armazenados em bancos de dados geográficos. Desta forma, a partir de um conjunto básico de dados, uma infinidade de mapas totalmente diferentes pode ser produzida, adequando-se à variedade de aplicações que necessitam de informações geográficas.

## Referências

- Agostini, Aminto (1972). *Disegno Topografico*, Ed. Ulrico Hoepli, Milão.
- Barkowsky, Thomas & Freksa, Christian (1997). “Cognitive Requirements on Making and Interpreting Maps”, *Lecture Notes in Computer Sciences 1329*, Springer-Verlag.
- Bertin, Jaques (1967). *Semiologie Graphique: les diagrammes, les reseaux, les cartes*, Gaurhiers-Villars, Paris.
- Câmara, Gilberto et al. (1999). “Handling Complexity in GIS Interface Design”, *anais do Geo-Info 1999 – I Workshop Brasileiro de Geoinformática*.
- Chomsky, N. (1972). *Syntatic Structures*, The Hague: Mouton & Co., Printers.
- Couclelis, & Gottsegen (1997). “What Maps Mean to People: Denotation, Connotation and Geographic Visualization in Land-Use Debates”, *Lecture Notes in Computer Sciences 1329*, Springer-Verlag.
- Davis, Clodoveu Jr. & Laender, Alberto H. F. (1999). “Múltiplas Representações em Aplicações Urbanas de Sistemas de Informação Geográfica”, *anais do Geo-Info 1999 – I Workshop Brasileiro de Geoinformática*.
- Egenhofer, Mark (1999). “Spatial Information Appliances: A Next Generation of Geographic Information Systems”, *anais do Geo-Info 1999 – I Workshop Brasileiro de Geoinformática*.
- Fonseca, Rômulo S. (1973). *Elementos de Desenho Topográfico*, Ed. McGrawHill do Brasil.
- Joly, Fernand (1990). *A Cartografia*, Editora Papirus.
- Martinelli, Marcello (1991). *Curso de Cartografia Temática*, Ed. Contexto.
- Monmonier, Mark (1991). *How to Lie with Maps*, The University of Chicago Press.
- Oliveira, Cêurio (1988). *Curso de Cartografia Moderna*, IBGE.
- Peirce, Charles S. (1972). *Semiótica e Filosofia*, Textos Seleccionados, Editora Cultrix.
- Peirce, Charles S. (1990). *Semiótica*, Tradução de “Collected Papers of Charles Sanders Peirce”, Ed. Perspectiva.
- Pratt, Ian (1993). “Map Semantics”, *Lecture Notes in Computer Sciences 716*, Springer-Verlag.
- Ramirez, Raul J. (1993). “Development of a Cartographic Language”, *Lecture Notes in Computer Sciences 716*, Springer-Verlag.
- Santaella, Lúcia (1989). “Por uma classificação da linguagem visual”, *Face, jan/jun 1989*, Editora da PUC-São Paulo.
- Santaella, Lúcia (1996). *O que é Semiótica*, Editora Brasiliense, 12ª edição.
- Schlichtmann, Hansgeorg (1985). “Characteristic traits of the semiotic system ‘Map Symbolism’”. *The Cartographic Journal*, 22.

# Aplicação de Raciocínio Baseado em Casos a Sistemas de Apoio à Decisão Ambiental

DANIEL DOS SANTOS KASTER<sup>1</sup>, CLAUDIA BAUZER MEDEIROS<sup>1</sup>, HELOÍSA VIEIRA DA ROCHA<sup>1</sup>

<sup>1</sup>IC-UNICAMP – CP 6176, 13081-970 – Campinas, SP, Brasil  
{daniel.kaster, cmbm, heloisa}@ic.unicamp.br

**Abstract.** One area which demands decision support applications is environmental planning. Because of the inherent complexity of environmental systems, standard Decision Support Systems (DSS) do not supply enough functionalities to deal with the related issues. Spatial DSS focused on environmental domains extend traditional DSS by providing adequate tools for environmental analysis and problem solving. This paper describes an environmental SDSS under development at IC-UNICAMP which combines scientific workflows, Case-Based Reasoning (CBR) and Geographical Information Systems (GIS). This system is based on two components: (1) an existing SDSS named WOODSS (WorkflOw-based spatial Decision Support System), developed at IC-UNICAMP, which interacts with a GIS and uses scientific workflows to document environmental models; and (2) an intelligent model retrieval mechanism based on CBR.

## 1 Introdução

Sistemas de apoio à decisão (*Decision Support Systems - DSS*) são sistemas que fornecem mecanismos para facilitar a interação do usuário, através de interfaces e pacotes específicos de software, com modelos de análise e dados de um determinado domínio para gerar e avaliar soluções alternativas.

Uma das áreas em que há grande demanda de soluções de apoio à decisão é o planejamento ambiental. O uso de técnicas de planejamento ambiental visa explorar racionalmente os recursos naturais [6]. No contexto da agricultura, por exemplo, envolve determinar o que plantar, onde e quando plantar, como preparar a terra, que técnicas de controle de pragas utilizar e como realizar a irrigação.

Em planejamento ambiental, os processos são usualmente expressados através de modelos, que aproximam e simplificam a realidade, fornecendo uma representação mais estruturada dos fenômenos em estudo. Modelos simulam, ou imitam, as operações e reações de processos do mundo real [16]. Modelagem e simulação computacional auxiliam cientistas a explorar hipóteses, analisar o funcionamento de processos ambientais e investigar a resposta de sistemas ambientais a possíveis mudanças.

Problemas ambientais, no entanto, são inerentemente complexos, pois tipicamente possuem características espaciais, são dependentes do tempo e de uma série de características geralmente inter-relacionadas [16]. Sendo assim, DSS convencionais são limitados para assistir à tomada de decisão no caráter ambiental.

Sistemas Ambientais de Apoio à Decisão (*Environmental Decision Support Systems - EDSS*) e Sistemas Espaciais de Apoio à Decisão (*Spatial Decision Support Systems - SDSS*) são extensões de DSS que oferecem aos usuários ambientes flexíveis para a tomada de decisões neste con-

texto. EDSS são extensões de DSS que possuem ferramentas especificamente desenvolvidas para assistir ao processo de tomada de decisão ambiental [13]. SDSS estendem DSS no sentido de que são capazes de manipular adequadamente dados espaciais, auxiliando eficientemente processos decisórios onde a dimensão espacial é fundamental para análise [4, 5]. Um SDSS aplicado ao domínio ambiental pode ser considerado um EDSS, e será tratado neste texto como um SDSS ambiental.

Geralmente, EDSS e SDSS utilizam Sistemas de Informação Geográfica (SIG) para executar tarefas de gerenciamento e análise de dados espaciais [13, 9]. No contexto deste artigo, SIG são entendidos como sistemas de informação que permitem coletar, armazenar e recuperar informações baseadas em sua localização espacial e explorar relações entre conjuntos de dados [12, 3].

Este artigo apresenta um SDSS ambiental, atualmente em desenvolvimento no IC-UNICAMP. Este sistema é baseado na combinação de conceitos de Inteligência Artificial e de Bancos de Dados Geográficos. O objetivo é oferecer um ambiente computacional que auxilie os usuários a realizar de forma eficiente as atividades necessárias à criação de cenários alternativos para planejamento ambiental.

O trabalho aqui apresentado é baseado em dois componentes: (1) um SDSS baseado em workflows científicos, que interage com um SIG, já desenvolvido na UNICAMP, denominado WOODSS (*WorkflOw-based spatial Decision Support System*) [14, 15]; e (2) um mecanismo de recuperação inteligente e de edição de modelos que utiliza técnicas de Raciocínio Baseado em Casos (*Case-Based Reasoning - CBR*) [10, 20] para auxiliar usuários a desenvolver novas estratégias de solução de problemas de planejamento ambiental a partir de estratégias já armazenadas no WOODSS.

As principais contribuições deste artigo são:



- Uma discussão sobre o uso de técnicas de raciocínio baseado em casos para apoio à decisão ambiental;
- A descrição da arquitetura de um sistema que combina workflows, bancos de dados e raciocínio baseado em casos para apoio à decisão na área de planejamento agro-ambiental.

Este artigo está organizado como segue: a seção 2 apresenta uma revisão dos conceitos necessários ao entendimento do trabalho, a saber, workflows científicos e CBR; a seção 3 descreve o sistema WOODSS; a seção 4 apresenta uma discussão sobre a utilização de CBR para apoio à decisão ambiental; a seção 5 descreve a aplicação de CBR no WOODSS; e a seção 6 mostra alguns trabalhos relacionados ao tema deste artigo; e a seção 7 traz as conclusões e o trabalho em andamento.

## 2 Conceitos básicos

Os principais conceitos para o entendimento deste artigo são associados a workflows científicos e raciocínio baseado em casos, descritos nas próximas seções.

### 2.1 Workflows Científicos

Um workflow pode ser definido como uma seqüência de passos necessários para atingir um determinado objetivo. Cada passo deste processo é chamado atividade ou tarefa, e pode ser executado por um ou mais agentes. Um agente ou executor é uma pessoa ou componente de software capaz de executar uma ou mais destas tarefas. Através de um papel é possível descrever um determinado (tipo de) agente de acordo com um conjunto pré-estabelecido de habilidades ou conhecimento de contexto necessários à execução de uma tarefa [2]. O objetivo de um sistema de workflows é auxiliar na especificação, execução, monitoramento e coordenação de um fluxo de trabalho.

A seqüência de passos de um processo decisório pode ser descrita através de um workflow, onde cada passo é representado por uma atividade. As informações de entrada e saída (dados, condições de execução, dispositivos a usar, entre outros) e os relacionamentos entre os diferentes passos do processo são expressados pelo fluxo do workflow.

Workflows científicos são extensões a sistemas de workflow tradicionais, especialmente definidos para documentar procedimentos e experimentos científicos [19, 18]. O trabalho científico é caracterizado por um grande grau de flexibilidade e incerteza e a ocorrência de exceções é maior do que o trabalho comercial. Desta forma, os mecanismos de workflow padrão são insuficientes para descrever este tipo de trabalho.

Workflows científicos estendem sistemas de workflow tradicionais nos seguintes aspectos:

- **Incompletude:** workflows científicos podem ser executados até mesmo quando incompletos, sendo assim construídos progressivamente durante sua execução. Workflows tradicionais, ao contrário, precisam ser totalmente definidos antes de serem executados;
- **Reutilização parcial:** workflows científicos diferem dos tradicionais pois são considerados como blocos em construção para especificação de experimentos. Assim, workflows parciais podem ser utilizados para a especificação de novos workflows;
- **Modificação dinâmica:** workflows científicos permitem re-executar atividades e também retroceder a uma atividade anterior, restabelecer seu contexto e continuar a execução por um novo curso de ação. Isto é, a especificação de um workflow pode ser modificada dinamicamente;
- **Execução de processos inválidos:** no domínio científico, processos de decisão são baseados no mecanismo de tentativa e erro. Workflows científicos são flexíveis o suficiente para suportar este tipo de abordagem;
- **Especificação a partir da instância:** workflows tradicionais são especificações para serem executadas frequentemente, onde cada execução é uma instância. Workflows científicos, por outro lado, podem ser executados uma única vez, como em tentativas sem sucesso. Uma vez que um workflow científico pode ser definido dinamicamente, sua especificação é realizada a partir da instância, ao invés da especificação definir a instância.

Segundo [15], o processo decisório no caráter ambiental pode ser naturalmente expressado como um experimento científico, onde o objetivo é produzir um mapa, representando um cenário alternativo, que irá indicar como resolver o problema. Este mapa é gerado a partir da execução de uma seqüência parcialmente ordenada de atividades, seguida de ajustes sucessivos, usando funções de um SIG. Assim, o processo de geração de mapas pode ser apropriadamente modelado como um workflow científico.

### 2.2 Raciocínio baseado em casos

Raciocínio Baseado em Casos (*Case-Based Reasoning - CBR*) é uma técnica para resolução de problemas que relembra e adapta soluções que foram usadas para resolver problemas similares anteriores [11, 20, 1]. Em CBR, um caso contém informações sobre uma experiência passada, como: a descrição da situação encontrada; em que esta difere de outras situações; e como o sistema reagiu à situação.

CBR é baseado no raciocínio natural humano, e existem muitas evidências de que pessoas usam CBR no seu

raciocínio cotidiano. Kolodner cita, em [10], pesquisas que durante o processo de aprendizado de uma nova habilidade as pessoas freqüentemente retornam a problemas anteriores para saber como realizar uma tarefa no novo contexto. Mecânicos, por exemplo, utilizam experiências próprias e de outros mecânicos para gerar hipóteses sobre problemas em um carro e como verificá-las.

As idéias do CBR foram desenvolvidas, em parte, como uma reação a algumas desvantagens de técnicas tradicionais de Inteligência Artificial, como o Raciocínio Baseado em Regras (*Rule-Based Reasoning - RBR*), que consiste em analisar uma situação através de um conjunto de regras pré-estabelecidas. Dentre estas desvantagens podemos citar:

- **Obtenção de conhecimento:** é geralmente mais fácil para especialistas articular soluções lembrando soluções para problemas previamente encontrados do que realizando inferências sobre um conjunto potencialmente grande de regras. Além disso, em sistemas baseados em regras é preciso reestruturar o código para acomodar expansões e tratamento de casos particulares. Em sistemas de CBR, por outro lado, o aprendizado é uma tarefa natural, através da retenção de novos casos. Desta maneira, torna-se mais simples adequar o sistema a novas situações e situações anômalas. Além disso, o conhecimento do domínio não precisa ser completamente definido no desenvolvimento do sistema, sendo expressado através do conjunto de casos armazenado;
- **Memória:** sistemas baseados em regras não possuem memória, o que obriga a construção de soluções sempre a partir do zero. Assim, erros anteriores tendem a acontecer novamente. Sistemas baseados em casos, por outro lado, possuem uma memória de casos anteriores, continuamente incrementada. Isto permite construir soluções a partir de outras, minimizando o tempo gasto e evitando cometer novamente os mesmos erros;
- **Robustez:** em sistemas baseados em regras, quando um problema não se encaixa em nenhuma regra, o sistema simplesmente não pode resolvê-lo. Sistemas baseados em casos, por sua vez, podem adaptar casos com características similares que podem ser úteis para resolver o problema atual.

Podemos descrever o ciclo básico de processamento do CBR como: dado um problema, obter as soluções anteriores relevantes, adaptá-las para o problema atual e armazenar o novo caso, juntamente com sua solução. Em [1] este processo é resumido nas seguintes atividades (figura 1):

1. **Recuperar:** analisar o caso de entrada extraindo os

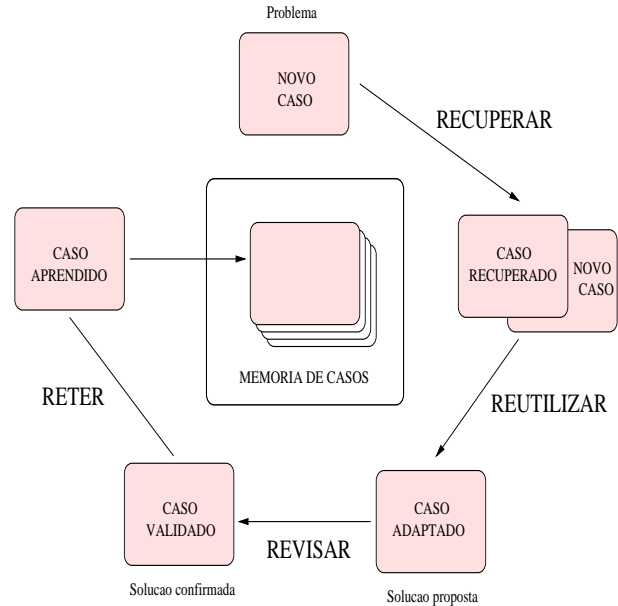


Figura 1: Ciclo básico de processamento em CBR [1].

descritores relevantes e recuperar, utilizando estes descritores, os casos mais similares ao caso de entrada. Esta atividade é fundamentada em técnicas como esquemas de "ranking" e métricas de similaridade;

2. **Reutilizar:** construir novas soluções a partir das soluções dos casos recuperados e/ou partes destas, através de ajustes e adaptações;
3. **Revisar:** avaliar e testar a solução construída para determinar sua correteude, utilidade e robustez;
4. **Reter:** acrescentar o novo caso à memória.

CBR pode ser aplicado aos mais variados domínios. Nesta diversidade, podemos distinguir dois tipos básicos de implementação de sistemas CBR: *sistemas totalmente automáticos* e *sistemas de recuperação de informação baseados em casos* [11]. Sistemas totalmente automáticos resolvem problemas de forma autônoma e têm mecanismos de interação com o mundo para avaliar os resultados de suas decisões. Sistemas de recuperação de informações baseados em casos, por sua vez, subsidiam pessoas que resolvem os problemas, como uma extensão da memória do usuário, a quem cabe realizar o raciocínio e tomar as decisões. É do segundo tipo de implementação que estamos tratando neste trabalho.

### 3 O sistema WOODSS

Como mencionado na introdução, este artigo propõe a aplicação de técnicas de CBR em um sistema espacial de

apoio à decisão, para auxiliar usuários em tarefas de planejamento ambiental. Esta seção apresenta o sistema WOODSS (*Workflow-based spatial Decision Support System*) [14, 15], desenvolvido no IC-UNICAMP, no qual estão sendo incorporados estes conceitos.

O WOODSS é baseado na noção de workflows científicos para descrição dos passos da construção de modelos. De fato, no contexto de sistemas geográficos, os mapas gerados são as soluções alternativas para um problema, e os modelos são os algoritmos utilizados para gerá-los.

O objetivo do WOODSS é auxiliar o usuário na etapa de construção de mapas, cabendo ao usuário interpretá-los e definir as medidas a serem tomadas. Isto é, o WOODSS visa oferecer mecanismos para manipulação e gerenciamento de modelos. Para alcançar este objetivo, ele interage com um SIG capturando as interações dos usuários em atividades de geração de mapas e cenários alternativos e documentando-as como workflows científicos (figura 2).

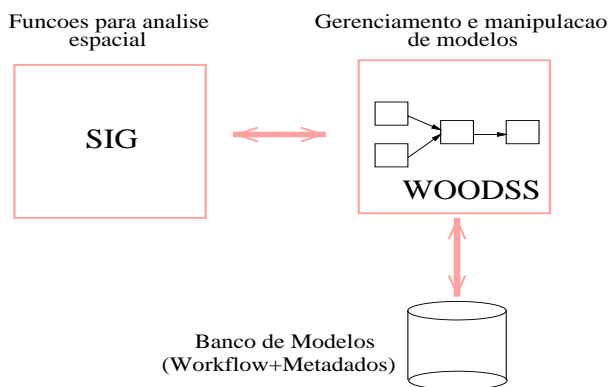


Figura 2: Interação do WOODSS com um SIG [14, 15].

Workflows científicos são usados no WOODSS com os seguintes objetivos:

- Para documentação do processo decisório;
- Como especificações em alto nível de modelos de simulação de processos ambientais;
- Como especificações parametrizadas de procedimentos de decisão, que podem ser reutilizados e adaptados para situações similares.

No WOODSS, um modelo é constituído por um workflow científico e um conjunto de metadados associados. O workflow científico descreve a seqüência de passos utilizada e o fluxo de dados entre estes passos. Os metadados contêm informações descritivas sobre o problema e sobre a

solução encontrada. Desta forma, contextualizam o algoritmo usado no domínio da aplicação, o que é essencial para reutilizações futuras.

Os modelos (workflows científicos e metadados associados) são armazenados em um banco de dados, também chamado banco de modelos. Sendo assim, os usuários podem consultar o banco de modelos, recuperar os modelos mais relevantes, e modificá-los para a geração de novas soluções. Os workflows podem ser modificados através da inserção, remoção ou refinamento de atividades, além da modificação de dependências e metadados.

A apresentação de processos como workflows oferece ao usuário uma visão clara e representativa. Permite ao usuário analisar o processo decisório como um todo, facilitando o controle do processo e o planejamento dos próximos passos. Além disso, o WOODSS oferece mecanismos que facilitam a execução, no sistema geográfico, das soluções geradas.

Mais detalhes sobre o WOODSS escapam a este artigo e podem ser encontrados em [14, 15].

#### 4 CBR e apoio à decisão ambiental

O raciocínio baseado em casos apresenta características que motivam sua aplicação em modelagem ambiental. Problemas ambientais são inerentemente complexos e em geral insuficientemente conhecidos e modelados. CBR é uma técnica onde o conhecimento é modelado a partir de instâncias, de forma que não se faz necessário modelar formalmente o conhecimento do domínio.

Conceitos que são partes do mundo real, como um pássaro, uma cadeira ou um fenômeno físico, são polimórficos. Isto é, suas instâncias podem ser caracterizadas de várias formas, e não é possível chegar a uma definição clássica completa em termos de um conjunto de características para tais conceitos. Uma alternativa para abordar este problema é representar estes conceitos extensivamente, ou seja, defini-los através de seus conjuntos de instâncias – ou casos [1].

Modelar sistemas ambientais através de conjuntos extensivos de instâncias aparece como uma alternativa perante a complexidade do domínio ambiental. Assim, é possível armazenar instâncias de processos aleatórios, bem como estados de processos em instantes diferentes e construir o conhecimento do domínio através destes conjuntos de instâncias. Este tipo de modelo é conhecido como modelo de instância específica [17].

A aquisição de conhecimento em raciocínio baseado em casos faz parte do processo natural de construção de soluções, uma vez que cada novo caso é armazenado na memória de casos. Desta maneira, o contínuo desenvolvimento do conhecimento de sistemas ambientais e das técnicas utilizadas pode ser acomodado sem maiores dificuldades.

des.

Além disso, representar o conhecimento através de instâncias facilita o tratamento de exceções e anomalias, não incomuns em processos ambientais. Afinal, cada caso encerra dentro de si uma situação acontecida e a forma como foi resolvida. Logo, exceções e anomalias podem ser tratadas de maneira análoga a situações comuns.

É possível extrair conhecimentos e generalizações a partir de uma memória de casos extensiva. Assim, é possível chegar a modelos mais genéricos e eficientes analisando o conjunto de instâncias armazenado. Além disso, uma memória de casos extensiva pode ser explorada, buscando padrões e tendências nos processos ambientais, contribuindo para o melhor entendimento destes processos.

Portanto, CBR possui um grande potencial de utilização em aplicações ambientais. Na seção 6 são apresentados algumas aplicações já desenvolvidas nesta área.

## 5 Raciocínio baseado em casos no WOODSS

A abordagem apresentada nesta seção descreve a inserção de CBR no WOODSS. O processo de recuperação de modelos é o primeiro passo na reutilização de soluções. Identificar o modelo mais similar possível ao modelo desejado para uma nova situação oferece ao usuário informações relevantes e diminui o trabalho de adaptação e adequação do modelo a uma nova situação. O objetivo da aplicação de CBR na edição e recuperação de modelos visa aprimorar a eficiência da reutilização de soluções.

### 5.1 Considerações sobre a recuperação de modelos

Um dos problemas do WOODSS reside na recuperação de modelos. Atualmente, a única forma de recuperação oferecida pelo sistema é através de palavras-chave. No entanto, esta forma de recuperação de modelos não é sempre adequada ao perfil dos usuários.

Usuários diferentes podem ter visões e descrições diferentes para os mesmos modelos, e mecanismos de busca tradicionais baseados em palavras-chave são limitados para lidar com este panorama. Além disso, estes mecanismos não tratam similaridades semânticas adequadamente.

Por exemplo, considere que um usuário gerou e armazenou no banco de modelos um modelo descrito como "avaliação da aptidão agrícola das terras da cidade de Campinas". Em um processo decisório posterior, outro usuário deseja realizar uma tarefa similar, só que para a cidade de Belo Horizonte. Se o segundo usuário fornecer palavras-chave do tipo "análise", "capacidade de uso" e "Belo Horizonte", sistemas convencionais baseados em consulta por matching exato de palavras-chave não retornariam qualquer modelo, embora o primeiro modelo pudesse ser útil na construção da sua solução para o segundo problema.

Considere, ainda, uma relação de similaridade mais

profunda. Por exemplo, a similaridade entre as cidades de São Paulo, Santos e Florianópolis depende de que aspectos estão sendo considerados. Se for considerado o aspecto de localização geográfica, São Paulo e Santos são mais similares; se forem aspectos ambientais, possivelmente Santos e Florianópolis tenham maior semelhança, por serem cidades litorâneas; e se for considerada a característica de ser capital de estado, a similaridade entre São Paulo e Florianópolis é maior.

Da mesma forma, em planejamento ambiental é preciso considerar o contexto do problema para que sejam recuperados modelos relevantes. A solução proposta consiste em embutir conhecimento do domínio da aplicação nos mecanismos de edição e recuperação de modelos para obter melhores resultados. Neste ponto, entra a inserção de CBR, que oferece naturalmente mecanismos para tratar *similaridade* entre casos (modelos, na aplicação em questão).

### 5.2 Arquitetura do WOODSS utilizando CBR

Sistemas de CBR utilizam uma estrutura para representação e organização dos casos, denominada memória de casos, e estruturas de índices para otimizar a recuperação de casos. Os algoritmos de recuperação devem considerar métricas de similaridade e "ranking" de casos, dependentes do conhecimento do domínio. A proposta de combinar bancos de dados e CBR visa especificar o mecanismo de CBR aproveitando o conhecimento já bastante maduro em bancos de dados sobre organização, indexação e recuperação de dados.

Um caso, nesta abordagem, corresponde a uma instância de utilização de um modelo. Desta maneira, um modelo pode ter vários casos associados. Por exemplo, um mesmo modelo (algoritmo no SIG) pode ser utilizado em mais de uma situação com dados de entrada diferentes. As diferentes situações são consideradas diferentes casos associados ao mesmo modelo. Armazenando todos os casos referentes a um modelo, o sistema automaticamente enriquece o conjunto de informações sobre o modelo, representando diferentes situações que levariam um modelo a ser lembrado, aumentando a flexibilidade da recuperação.

A figura 3 mostra de forma simplificada a arquitetura do WOODSS, já considerando a utilização de CBR. O módulo monitor/conversor captura as interações do usuário com o SIG e as codifica como um workflow científico. Além disso, é capaz de traduzir modelos em alto nível no WOODSS em códigos executáveis no SIG.

O banco de modelos contém os modelos (workflow científico + metadados) armazenados e a memória de casos os casos associados aos modelos. Para recuperar um modelo, o usuário fornece uma descrição do problema, que é comparada com os casos armazenados na memória de casos. O sistema retorna ao usuário o modelo referente ao caso mais

semelhante ao problema apresentado.

O módulo de edição permite que o usuário realize as adaptações e os ajustes necessários para reutilizar o modelo como solução para o novo problema. Uma vez validado pelo usuário, se o modelo gerado difere dos modelos já armazenados, este é retido no banco de modelos e o caso correspondente retido no banco de casos. Caso contrário, somente o novo caso de utilização é retido.

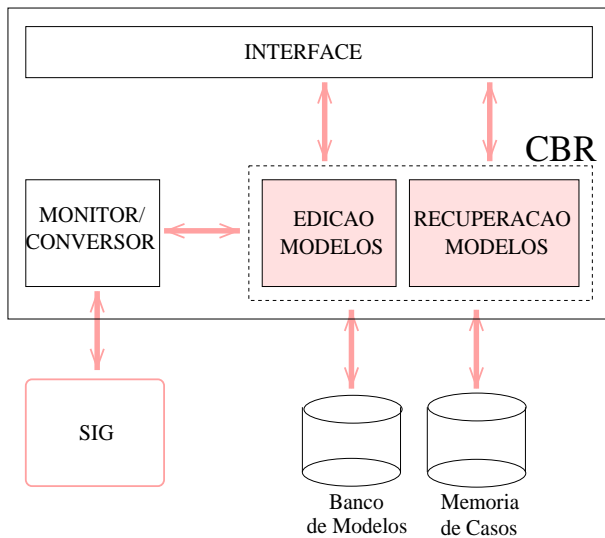


Figura 3: Arquitetura do WOODSS utilizando CBR na recuperação de modelos.

Rizzoli et al. Apresenta em [13] um conjunto de características genéricas desejáveis de um sistema de apoio à decisão ambiental (EDSS). A necessidade de cada uma destas características em um sistema particular irá depender da natureza do problema, da situação e dos usuários do sistema:

1. Capacidade de adquirir, representar e estruturar o conhecimento do domínio em estudo. Para isto, é preciso identificar o domínio da aplicação, estabelecer quais são as estruturas de dados necessárias para representar os componentes do domínio modelado e entender os relacionamentos entre eles;
2. O banco de modelos, onde o conhecimento estruturado do domínio é armazenado, deve permitir a separação entre dados e modelos para facilitar a reutilização de modelos e a prototipagem;
3. Capacidade de lidar com dados espaciais;
4. Oferecer ajuda especialista no domínio de interesse, através de técnicas de inteligência artificial. Conhecimentos do domínio são úteis para auxiliar o usuário a

recuperar informações, preparar dados, selecionar modelos e interpretar resultados;

5. Fornecer mecanismos para auxiliar o usuário durante a formulação do problema e na escolha dos métodos de solução.

Nota-se que o WOODSS oferece inicialmente as três primeiras características sugeridas:

- De fato, workflows no WOODSS permitem representar e estruturar o conhecimento do domínio em estudo, representado através dos modelos. Além disso, com o mecanismo de captura de interações dos usuários com o SIG, o WOODSS permite adquirir conhecimentos automaticamente. Os relacionamentos entre as tarefas e sub-processos são apresentados naturalmente através dos workflows (1);
- O banco de modelos é provido pelo WOODSS (2);
- O SIG acoplado ao WOODSS fornece as funções de manipulação e análise de dados espaciais (3).

Com a inserção de CBR, o WOODSS adquire características desejáveis de um EDSS até então não oferecidas:

- Com a utilização de CBR na edição e recuperação de modelos torna-se sensível ao domínio e auxilia o especialista na formulação do problema e na escolha dos métodos de solução (modelos) e pode ser mais efetivo nas tarefas de planejamento ambiental (4);
- Através do conhecimento do domínio embutido no conjunto de casos e do mecanismo de similaridade parcial entre casos, o sistema pode identificar semelhanças entre os questionamentos do usuário e os modelos armazenados, retornando os mais adequados, que, por sua vez, contêm tarefas computacionais de modelagem e simulação (5).

Sob esta ótica, o acoplamento de CBR ao WOODSS permite a construção de um *SDSS ambiental* adequado.

## 6 Trabalhos correlatos

Existem algumas abordagens que utilizam CBR em aplicações de apoio à decisão ambiental, como [8, 17, 7]. Hastings apresenta em [7] um sistema de notificação de alertas sobre infestações de gafanhotos combinando CBR e raciocínio baseado em modelos (*Model-Based Reasoning*). O processo decisório utilizado consiste em estimar a proporção de folhagem que será consumida pelos gafanhotos, avaliar se a infestação pode afetar as plantações e, caso isto ocorra, estimar os custos e benefícios das medidas a serem tomadas. O sistema, denominado CARMA (*CASE-based Range Management Adviser*), não utiliza SIG.

Verdenius e Broeze apresentam em [17] um sistema baseado em CBR aplicado ao domínio de tratamento de águas de esgoto utilizando plantas e microorganismos. Durante o processo de transformação de substâncias tóxicas, existe uma grande demanda de oxigênio. Para que a tarefa de purificação da água tenha um desempenho ótimo, uma técnica utilizada consiste em injetar doses de oxigênio na água. A idéia do sistema consiste em gerenciar o nível de oxigênio e decidir nas variadas situações que medidas tomar. Este sistema também não utiliza SIG.

Uma aplicação que combina CBR e SIG é apresentada por Holt e Benwell em [8]. Nesta referência, os autores descrevem um sistema de classificação de solos denominado ZONATION. A proposta deste sistema reside no fato de que, segundo os autores, não existe uma fórmula genérica que pode ser aplicada para classificação de séries de solos para um ambiente (ou área) genérico. O sistema apresentado permite aos especialistas realizar classificações baseadas em instâncias prévias, utilizando conhecimento específico do domínio.

Embora a abordagem de Holt e Benwell utilize SIG, a proposta descrita neste artigo é original, pois: (1) o domínio da aplicação é distinto (neste trabalho é o domínio de planejamento agro-ambiental); e (2) a abordagem apresentada neste artigo visa assistir ao usuário na tarefa de construção de modelos ambientais e não envolve a utilização de modelos definidos formalmente, como nas aplicações citadas.

## 7 Conclusões e extensões

Este artigo apresenta um sistema atualmente em desenvolvimento no IC-UNICAMP visando a construção de um SDSS ambiental. Este sistema está baseado em acoplar a um sistema atualmente existente, o WOODSS, um conjunto de ferramentas de edição e recuperação de modelos utilizando CBR. O trabalho está sendo centrado na área de planejamento agro-ambiental, contando para isto com apoio de especialistas da Faculdade de Engenharia Agrícola da UNICAMP.

A combinação de técnicas de CBR e bancos de dados geográficos para apoio à decisão é um tópico de pesquisa de ponta, onde existem ainda poucos trabalhos desenvolvidos. A proposta descrita é original em vários aspectos, destacando a possibilidade de auxiliar o usuário na criação e adaptação de modelos, recuperados com CBR, e pela aplicação estar focada no domínio de planejamento agro-ambiental.

Além disso, destacam-se os seguintes resultados:

- Com a inserção de CBR o sistema WOODSS torna-se mais efetivo para apoio à decisão no contexto ambiental, fornecendo mecanismos para gerenciamento e recuperação eficiente de modelos, tratando-os como casos;

- A definição de atributos e parâmetros adequados para aplicação de CBR em um domínio específico e novo. Estes atributos correspondem às especificações para representação, indexação, armazenamento e recuperação de casos. Levantando, desta forma, conhecimentos que podem ser aplicados futuramente em novos trabalhos de pesquisa.

O estágio atual deste trabalho é o levantamento de atributos e indicadores de similaridade no domínio agro-ambiental, realizado em paralelo com modificações ao WOODSS para permitir incorporar estruturas de CBR.

## Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq e pelos projetos PRONEX MCT/FINEP SAI (Sistemas Avançados de Informação) e CNPq/NSF Interoperabilidade em SIG. Os autores agradecem especialmente ao professor Jansle Vieira Rocha, da FEAGRI UNICAMP, pelas inúmeras reuniões, discussões, dados e casos para este trabalho.

## Referências

- [1] A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications*, 7(1):39–59, 1994.
- [2] P. Barthelmess. Sistemas de workflow: Análise da Área e proposta de modelo. Master's thesis, IMECC-UNICAMP, 1996.
- [3] G. Câmara, M. A. Casanova, A. S. Hemerly, G. C. Magalhães, and C. B. Medeiros. *Anatomia de Sistemas de Infomação Geográfica*. Instituto de Computação - UNICAMP, 1996.
- [4] M. D. Crossland, B. E. Wynne, and W. C. Perkins. Spatial decision support systems: An overview of technology and a test of efficacy. *Decision Support Systems*, 14(3):219–235, July 1995.
- [5] P. J. Densham. Spatial decision support systems. In D. J. Maguire, M. F. Goodchild, and D. W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, chapter 26, pages 403–412. Longman Scientific and Technical, New York, NY, 1991.
- [6] K. Fedra. GIS and environmental modeling. In M. F. Goodchild, B. O. Parks, and L. T. Steyaert, editors, *Environmental Modeling with GIS*, chapter 5, pages 35–50. Oxford Univesity Press, New York, NY, 1993.
- [7] J. D. Hastings. *A Mixed Paradigm Approach to Problem Solving in Incomplete Causal-Theory Domain*. PhD thesis, University of Wyoming, Laramie, Wyoming, 1996.

- [8] A. Holt and G. L. Benwell. Applying case-based reasoning techniques in GIS. *The International Journal of Geographical Information Science*, 13(1):9–25, 1999.
- [9] P. Keenan. Using GIS as a DSS generator. Working Paper MIS 95-9, Department of Management Information Systems, Faculty of Commerce, University College Dublin, Dublin, Ireland, April 1997.
- [10] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [11] J. L. Kolodner and M. Y. Jona. Case-based reasoning: An overview. Technical Report 15, Northwestern University, June 1991.
- [12] D. J. Maguire. An overview and definition of GIS. In D. J. Maguire, M. F. Goodchild, and D. W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, chapter 1, pages 9–20. Longman Scientific and Technical, New York, NY, 1991.
- [13] A. E. Rizzoli and W. J. Young. Delivering environmental decision support systems: Software tools and techniques. *Environmental Modelling and Software*, 12(2–3):237–249, 1997.
- [14] L. Seffino. WOODSS - a spatial decision support system based on scientific workflows. Master's thesis, UNICAMP, July 1998.
- [15] L. Seffino, C. B. Medeiros, J. V. Rocha, and Bei Yi. WOODSS - a spatial decision support system based on workflows. *Decision Support Systems*, 27(1–2):105–123, November 1999.
- [16] L. T. Steyaert. A perspective on the state of environmental simulation modeling. In M. F. Goodchild, B. O. Parks, and L. T. Steyaert, editors, *Environmental Modeling with GIS*, chapter 3, pages 16–30. Oxford University Press, New York, NY, 1993.
- [17] F. Verdenius and J. Broeze. Generalized and instance-specific modelling for biological systems. *Environmental Modelling and Software*, 14:339–348, 1999.
- [18] J. Wainer, G. Vossen, and C. B. Medeiros. Workflow management in geoprocessing applications. Technical Report 04/98-I, University of Münster, Germany, February 1998.
- [19] J. Wainer, M. Weske, G. Vossen, and C. B. Medeiros. Scientific workflow systems. In *Proceedings of the NSF Workshop on Workflow Process Automation: State-of-the-art and Future Directions*, 1996.
- [20] I. D. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, 1997.

# Exchange Mechanism for Global Mapping

*Teresa Cristina Veiga*

IBGE- Instituto Brasileiro de Geografia e Estatística, Av. Brasil, 15671, 21241-051 Rio de Janeiro, RJ, Brasil

[teresaveiga@ibge.gov.br](mailto:teresaveiga@ibge.gov.br)

**Abstract.** This paper describes findings and results of the research realized for the Global Mapping Project, with the cooperation of Geographic Survey Institute (GSI), Japan, in order to set the procedures for converting cartographic digital data in vector format, produced at IBGE, at 1 to 1million scale, to Global Map specifications.

## 1. Introduction

The Global Mapping Project, led by the International Steering Committee for Global Mapping (ISCGM) have its roots on the results of the United Nations Conference on Environment and Development (UNCED), held in Brazil in 1992, and on the report of the Special Session of the United Nations General Assembly on the Implementation of Agenda 21, held in June 1997, stressing the importance of public access to globally mapped information and of international cooperation in making it available.

A Global Map (GM), or a group of global digital geographic datasets, is the expected resultant product of the Global Mapping Project. According to the GM Specifications [3], the main objective of Global Mapping Project is to bring all nations and concerned organizations together to develop and provide easy and open access to global digital geographic information, at a scale of 1:1 million, to facilitate the implementation of global agreements and conventions for environmental protection, for monitoring environmental phenomena and for encouraging sustainable development.

The first phase of the Project, up to year 2000, involves gathering digital geographic information, at global map scale, organized into thematic layers, or coverages, either in vector form (baseline features) or raster form (elevation, land cover, land use and vegetation), through international cooperation of national mapping organizations.

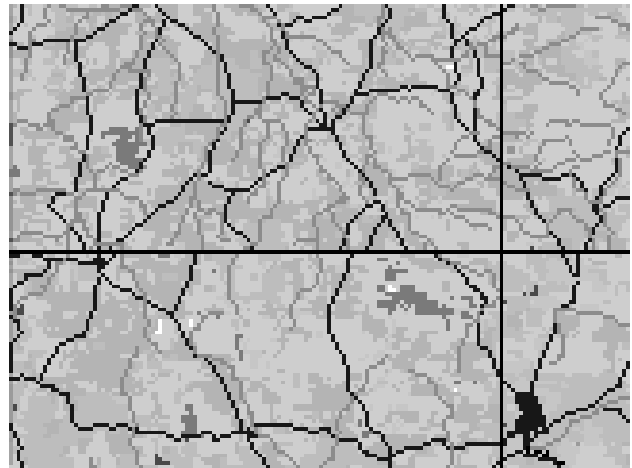
All countries are encouraged, by ISCGM, to participate according to their capacity of developing suitable data for other countries (level A) or for its own territory (level B), or even only by inputting data to be developed by a third party (level C).

Brazil, through its National Mapping Agency (Brazilian Institute of Geography and Statistics – IBGE) joined the Global Mapping Project in 1999, as a level B participant country, after being invited to integrate the project, since its participation at the VI United Nations Regional Cartographic Conference for the Americas, on

July 1997, at New York, where the Global Mapping Project was object of special attention. Before joining officially the Project, IBGE participated at the Global Mapping Meeting at Gifu, Japan, on November at the same year, where it was requested the support of National Mapping Agencies for the implementation of the project until the end of year 2000.

As a consequence, it was offered to IBGE the opportunity to send a participant for the Global Mapping (Environment) Course held in Japan, from May to August, 1998. During the Course it was tried to use the existent vector digital dataset from the Digital Chart of the World (DCW), referent to the CIM sheet of Goiânia (SE-22), but it was verified that data was out of date and tiles didn't mach properly, needing edition, as shown in Figure 1.

**Figure 1 - DCW file example**



## 2. Information in digital format for conversion

Many countries, like Brazil, which are participating of Global Mapping Project, are being encouraged by the International Steering Committee for Global Mapping (ISCGM) to convert their paper maps from many different digital formats, using a variety of systems, database structures and a different set of attributes for defining spatial entities, specially in terms of baseline features, to



the Global Map specifications or, at least, to a format compatible with VPF, which can be used by ISCGM to distribute the datasets. Some of these digital databases can be used directly in Geo Information Systems (GIS) others are ready only for Cartographic purposes or for map sheets printing.

Exchanging data related with Global Mapping Project implies on both importing from VPF to a particular system and exporting from any system to VPF format. A first step can be taken by using any existent digital data and convert it to VPF, directly or through other formats which can be readily used by participating countries. It doesn't exist an appropriate method which can fulfill all needs of all users. Each case must be seen as a particular case but there are some basic procedures that are common to the process as a whole.

### **3. Software systems for producing and converting digital information**

Different software systems are used for digital databases production and management, either for cartographic or GIS purposes. Among many of them, we can mention, for example:

- Intergraph MGE (based on ODBC/ACCESS, Oracle or Informix) [7];
- ESRI ArcInfo (ArcView) [6];
- Bentley Microstation (Geographics) [5];

Import and export from one software system to another, inside a GIS environment, is still a hard task. It is difficult to find a software which can fully read any other format. All GIS software have significant limitations in converting from one format to another, according to the complexity of their importing and exporting functions, despite the effort of software suppliers on overcome it.

### **4. CIM Project**

The International Chart of the World at 1 to 1 million (1:1,000,000) scale (CIM), of the Brazilian Territory, has been produced by the Brazilian Institute of Geography and Statistics (IBGE) according to the "Manual of Rules, Specifications and Technical Procedures" [1]. These rules were adopted at Bonn's United Nations Technical Conference about the International Chart of the World at 1 to 1 million scale, in 1962, after the revision of London (1909) and Paris (1913) resolutions.

The "Manual" [1] aimed at standardize the procedures for making cartographic originals at 1 to 1 million scale, representing the geographic space mainly by hydrography (drainage), hipsography (elevation), and relevant planimetric elements, necessary for the physical and social characterization of a territory.

The Brazilian territory is covered by a set of 46 CIM sheets (or tiles) which give support for the realization of strategic studies and territorial analysis in a national scale. These sheets are also used as a support for building other cartographic documents, at smaller scales, which will compose the base for representing thematic subjects like population distribution, soils, geology and vegetation, among others.

The preparation of the new version of 1:1,000,000 sheets, which started in 1993, was concluded in 1998 and printed in 1999. IBGE, through the Department of Cartography, made a revision of the necessary procedures and elaborated a new version for the "Manual" (1), attending to the rules of Bonn's agreement about the International Chart of the World at 1 to 1 million (1:1,000,000) scale (CIM), at the UN Technical Conference.

The project of converting the 1:1,000,000 paper sheets to digital format, named CIM Project, is meant to both cartographic purpose and for being used by GIS systems, mainly on Intergraph MGE environment, which is the main software system used at IBGE. The CIM database was structured to fit with this system and to fulfill other cartographic requests.

The first phase of the process for converting the 46 sheets at 1:1,000,000 scale to digital format, by IBGE, started in the beginning of 1998 and involved the conversion from paper maps to raster files by scanning all the cartographic originals used for composing the new edition. By the middle of 1999 the raster files, obtained through this process, started to be converted from raster to Intergraph .DGN vector format files. This work is being carried out by a third party contracted by the National Agency of Electric Energy (ANEEL) in urgent need of a reliable digital database to support energetic development plans. The next necessary steps to be taken are: to codify the vectorized files, to link them with the attributes database, to build topology and to produce new data at smaller scales for regional and national planning. Database is still based on ODBC/ACCESS environment, but will be converted to Oracle system in a near future.

The system selected to realize this work was Intergraph MGE system which comprises raster as well as vector modules and is already spread through several Departments at IBGE. By the end of March 2000 the vectorization of the 46 sheets was completed and ready to be structured topologically.

The cartographic originals (blue, red, yellow, green and black) where split in 9 different vector files, or layers, for each sheet, according to the categories specified at the Digital Topographic Map Library (MTD) structure: hipsography (contour line, height), limit (political-administrative limit), hydrography (river, canal, lake,

shoreline), vegetation (marshland, mangrove), locality (city, village, capital), transportation system (road, railroad, train station), works and buildings (breakwater, sea wall, lighthouse, plant, cemetery, rural building).

The 1:1.000.000 scale digital database, coming from the International Chart of the World (CIM) and covering all the territory will be used for building other databases at smaller scales and for the Global Map Project. This means IBGE will be able to produce its own data for the Global Map.

The Department of Cartography (DECAR) is in charge of producing vector data for the Global Map Project, which will be adapted from the CIM Project. The Department of Natural Resources (DERNA) will be in charge of producing data related with vegetation, land use and land cover. Elevation data can be obtained from the existent vector database through the contourlines and other elevation data. Political boundaries, at Municipal level, were added to CIM Project and to the Global Map Project from the existent Digital Municipal Boundary 1997 database. By the end of the year 2000, It will be incorporated the new Digital Municipal Boundary used for the Census 2000.

Digital data representing CIM sheets are grouped in classes of cartographic elements which are assembled in the categories shown at Table 1.

**Table 1 - CIM Categories**

CATEGORY NUMBER (*)	CIM CATEGORY	CATEGORY CODE
01	hypsography (elevation)	hp
02	limit (boundary)	lm
04	hydrography (drainage)	hd
05	vegetation	vg
06	locality (place, site)	lc
07	transportation system	st
09	work and building 02 (structures)	o2

(\*) The category code number 03 corresponds to the category referential point (survey point, landmark) which elements are not represented at CIM Project; elements of the categories 08 and 10, work and building 01 and 03, are incorporated at the work and building 02.

The structure of MTD database which has been applied to Global Map CIM Project has been built based on Intergraph MGE structure on ODBC/ACCESS system, and comprises: categories, features, attributes and domains for the attribute values. **Table 2**, shows an example of the structure for MTD category HIDROGRAPHY.

Data obtained through the vectorization process will be codified, linked with the attributes in the database and topologically structured. For being used by the Global Map Project, however, these files need to be compatible with Global Map specifications and to be converted to

VPF format. The alternatives affordable to accomplish this task, by the time of the research were:

- 1) Using SDTS module, inside Intergraph MGE System, to create an output which can be used by another System to create VPF files;
- 2) Using Bentley GEOEXCHANGE System, to create an ARC/INFO file or an ARCVIEW "shape" file to be used for further conversion to VPF;
- 3) Creating a new project using .DGN graphic files and ACCESS database tables, in an ARC/INFO/ARCVIEW environment for exporting to VPF format through the programs already created for this.

## 6. GM Project

According to GM Specifications [3] (pg. 29/30), the Global Map product will encompass the entire globe, at a scale of 1:1 million. For accomplish this task, data sets must be consistent with either the scale and the specifications. Spatial features are organized into thematic layers or "coverages" each containing logically related geographic information. For the purpose of this research, only vector form layers were considered. The features of the vector data model, on these layers, which will be comprised of three spatial objects: points, edges, and faces, can be seen at **Table 3**.

Still according to the Specifications [3], vector features will be selected on the basis of their suitability for 1:1 million scale mapping. As a result of this, some features may occur only in less densely areas where they are referentially more important.

Global Map thematic coverages share the same coordinate system and tiling structure. The tiling schema adopted for Brazil have the dimensions of 5 x 5 degrees.

## 7. Comparison

CIM Project and GM project databases are different and structured diversely. The correspondence between layers, features and attributes can be evaluated through the Tables 4, 5 and 6.

**Table 4 - Correspondence between CIM Project categories and GM Specifications categories**

CIM Category Name	GM Category Name
hypsography (elevation)	elevation (raster)
limit (boundary)	boundaries
hydrography (drainage)	drainage
vegetation	vegetation (raster)
locality (place, site)	population centers
transportation system	transportation
works and building (structure) 02	not on an specific layer

Features from CIM project are comprised on 7 categories. A full correspondence among layers of

categories was not encountered. In some categories like drainage and transportation many features are compatible, but in others like boundaries and population centers there was some difficulty for making the correspondence with Global Map mandatory features, like the coast line which doesn't exist as a separate feature on CIM Project. The works and building category has no correspondence at all.

While Global Map specifications foresee the features representing the vector data model comprising of three spatial objects: points, edges and faces, besides text, CIM project works with 6 different spatial objects, including text. (See Table 5a and 5b)

**Table 5a** - Spatial objects representing vector features at MTD (MGE)

CIM Object type	CIM object description
point	(*)object with no dimensional representation
line	(*)object which extension can be represented
area boundary	object with two dimensions represented
area centroid	(*) point inside the area boundary
closing line	imaginary line for closing an area boundary
text centroid	point to which is assigned a text

(\*) object connected to an attribute table

**Table 5b** - Spatial objects representing vector features according to GM Specifications

GM Object type	GM Object description [3]
point	0-dimensional geometric primitive
edge	1-dimensional topologic primitive
face	2-dimensional topologic primitive
text	text

The topological structure used inside Intergraph MGE environment doesn't configure an area or a polygon with attributes but, instead, an area boundary associated with a centroid which contains the attributes of the polygon represented by this area boundary. This centroid in some cases can be considered a point feature as well, which can cause some confusion.

Some attributes and features related to a Global Map coverage are similar to the ones of CIM Project; others are contained in a different layer of CIM Project or just don't exist. The distribution of spatial objects or features among categories is also not the same in both projects. "Dam/weir" included in the "drainage" coverage of Global map project, for example, must be extracted from the category "work and building 02 (structures)" from CIM Project. These incompatibilities, which can be seen at **Table 6**, require that some extra preparation work must be done in order to make features and attributes usable for the

Global Map Project.

Political boundaries are obtained from the Digital Municipal Boundary 1997 database, which has been developed for ARC/INFO environment and adapted to CIM Project prior of being used directly for Global Map.

## 8. Hypothesis

Two possibilities were figured out as being the most suitable to fulfill the task of building the Global Map CIM Project.

The first one - and apparently the easiest one - was to maintain the structure and codes from CIM Project, in Portuguese, and convert them to Global Map, in a way that feature names and attributes are converted to English and to Global Map specifications on the process; the second one, to structure a completely new project, based on MTD structure at MGE environment, in English, following GM features specifications, to be exported directly to Global Map.

## 9. Research

The main purpose of the research was to evaluate which was the best way to use existent digital data from IBGE's CIM Project, in order to produce Global Map tiles of Brazilian territory in VPF format.

Data used on the research tests came from the digital database, rasterized and vetorized with Intergraph MGE System, derived from 1:1,000,000 existent CIM cartographic originals.

It was selected a file representing a single CIM sheet from the digital database with graphic features connected to its attributes, codified and topologically structured, containing all categories and a representative number of features, in order to test the first possibility outlined in the hypothesis, and the behavior of different features (objects). A political boundary file, required by GM specifications, was composed with the coastline feature taken from the drainage database, once at the moment of the research, the category "political boundaries" file was not yet available for CIM Project.

The first alternative tested was to convert directly from Intergraph MGE system using SDTS module. Because the existent version of Intergraph MGE Modules, at IBGE, is working based on an ODBC/ACCESS database system, it was not possible to use the SDTS module, which needs a database schema system not applicable for the existent database.

Using Bentley GEOEXCHANGE System was possible to create ARC/INFO files and ARCVIEW "shape" files which could be used for conversion to VPF.

A brief explanation about the sequence for GeoExchange conversion process, starts by setting the

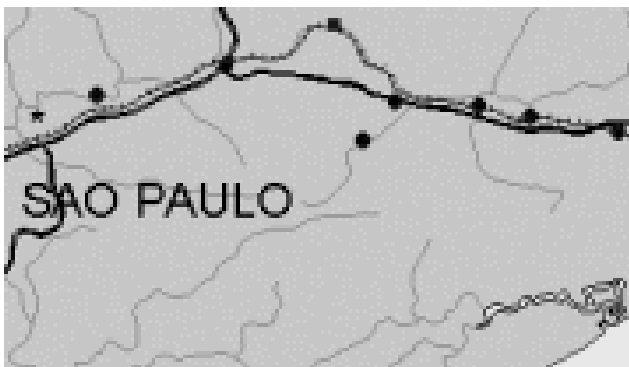
database contained in the directory of the Intergraph MGE project for ODBC system. It's not necessary to build a new project if a MGE project is already structured in a particular directory. It is necessary only to open the file to be converted on GeoExchange, to open the existent MGE project, to set the necessary variables and to export the file. GeoExchange exports, separately, each feature contained on the file.

Point and line features hardly presented any problem on the conversion, but polygons were object of some struggle before being properly exported and used on the next steps. For being exported from Intergraph MGE system, the area boundaries need to be "complexed" with the area centroids to generate a "face" type output required by Global Map specifications. GeoExchange, however, don't export completely these polygons when they have other polygons (holes) inside, like lakes on islands or islands on lakes. It's necessary to separate these features before exporting.

To convert CIM Project files to the tiles with GM specifications, ARCINFO software was used as an intermediary step. After exporting the selected features from CIM files, through Geoexchange, to the ArcInfo .E00 or shape file formats, it is necessary to import them into ArcInfo coverages. The conversion process to VPF, uses the "VPFkit for Global Map" built with the Production Line Tool Set (PLTS) from ESRI [6].

The final Global Map sample product, obtained from CIM Project files converted to VPF format, through the procedures executed during the research, can be seen with ARCVIEW software [6] as exemplified in Figure 4. Its features structure is shown in the Table 7.

**Figure 4 - Global Map vector file example**



As the second alternative tested, using Bentley's GeoExchange software and the programs already developed in ARCINFO environment by GSI, fulfilled the needs to convert data from Intergraph MGE to Global Map VPF, for the purpose of this research, it was not necessary to test the third one.

**Table 7 – Global Map structure at VPF**

GM VPF Layer	Cov. Name	GM VPF Feature	Feat. Type	Feat. Code	Global Map Feature Description
Hydrography	hydro	miscp	point	BA030	Island smaller then 1 km2
		watcrsl	edge	BH140	Water course, river, stream
		misc1	edge	BI020	Dam, Weir
		inwatera	face	BH000	Inland water area
				BH090	Land subject to inundation

## 10. Conclusions

In order to fulfill the requests of a final product which fits with Global Map specifications, some extra initial work must be done with CIM files to avoid future troubles. This work implies in some adaptation, separation and aggregation of features in different layers, as it has been done with the coast line feature created from CIM category "hydrography" to compose the Global Map "political boundaries" coverage.

The first problem faced is the way the feature name is written in the CIM database. As outlined on the first hypothesis, the database was maintained in the way it was structured for CIM Project, in Portuguese. Intergraf MGE and the Microsoft ACCESS internal structures allow the use of blank spaces but ArcInfo "import command" doesn't recognize it. GeoExchange, on the conversion process, maintain the original name and doesn't allow any change or edition. As soon as the features are exported it is necessary to edit their names, eliminating or substituting the blank spaces, allowing ArcInfo "import command" to work properly.

Point features and line features tested, hardly presented problems on the conversion process. Polygon features, meanwhile, had to be edited and tested many times, in different ways, to find the right way to use them in the sequence of conversion procedures from MGE to ArcInfo, through GeoExchange, and latter to VPF through "VPFKit". The perfect solution, however, couldn't be found. Polygon holes on complexed area boundary features aren't exported by GeoExchange directly and had to be worked separately, which complicates automation procedures.

Island point feature doesn't exist on CIM Project database as a separate entity. For Global Map project it will represent all islands with less then 1 square km. This is a subject to be considered on the further development of the project because all islands which appears on CIM database are composed by an area boundary with an area centroid containing the attributes, independently of their size.

The Global Map boundary feature "coast line" doesn't exists as a separate feature on CIM database but as

an attribute of a political administrative boundary line feature. In this way it will be possible to make a specific query to export it separately. The same problem occurs with ocean/sea boundary feature. At the CIM database “ocean” is an attribute of a “permanent water body” area centroid feature from the category Hydrography. By querying and “complexing” it with the area boundary feature it is possible to create a new feature to be exported to the Global Map coverage “political boundaries”.

Other problem encountered, and which had to be worked out during the conversion procedures, is related with projection, coordinates and working units. The digital database derived from CIM cartographic originals was built on Lambert Conformal Conic Projection, with Corrego Alegre Datum, using the unit “degrees, minutes and seconds” for longitude and latitude coordinates, and working unit of 10000 UOR per km (Units of Resolution or the smallest unit of accuracy at which a point can be located within a design area) for measurements.

All this had to be converted in ARCINFO before exporting to VPF. Degrees, minutes and seconds were converted to decimal degrees and the units of resolution from Km to m using the conversion factor 0,001 meters. Conversion parameters are necessary to adapt projections and working units to Global map specification.

A new database in English, built on Intergraph MGE structure, but using GM specification for features and attributes can be used in the future for testing the second hypothesis. It could facilitate the conversion process, eliminating some steps. The names of the features in the proposed database are proper to conversion without blank spaces. Global Map feature code was added to the project structure as an attribute, in this way there won't be problems of code repetition for different feature objects in Global Map structure once MGE code structure doesn't allow code repetition for different elements. The structure of MTD codes was maintained in order to keep the link with CIM Project, specially, because features for composing GM categories came from different CIM categories, for example, “dam/weir” from the category “work and building 02 (structures)” to the coverage “drainage” and “ocean/sea” from the “hydrography” to the “political boundaries”.

The model of the proposed structure for a database for further conversion from an MGE project to a Global Map Project can be seen at **Table 8**.

For the scope of CIM Project it will be useful to include in the Global Map CIM Project the feature “land subject to inundation”, at “Inland water” feature table, because it is a valuable information for environmental development projects. Some other features existent and codified in both databases, such as depth line, pier, danger points, rapids, waterfall, breakwater, sea wall, native

settlement and lighthouse, could also be included when available.

As a final remark, it will be very useful and helpful for Global Map CIM Project if a solution to convert digital data directly from Intergraph MGE to VPF format could be found.

## 11. References

[1] Fundação Instituto Brasileiro de Geografia e Estatística, Diretoria de Geociências, Departamento de Cartografia - IBGE/DGC/DECAR (Brazilian Institute of Geography and Statistics, Directory of Geosciences, Department of Cartography), “Manual de normas, especificações e procedimentos técnicos para a Carta Internacional do Mundo, ao Milionésimo - CIM 1:1 000 000 (Manual of technical standards, specifications and procedures for the International Chart of the World, at 1 to 1 million scale)”, *Série Manuais Técnicos em Geociências, número 2 (Technical Manuals on Geosciences Series, number 2)*, IBGE, Rio de Janeiro, 1993.

[2] Fundação Instituto Brasileiro de Geografia e Estatística, Diretoria de Geociências, Departamento de Cartografia - IBGE/DGC/DECAR (Brazilian Institute of Geography and Statistics, Directory of Geosciences, Department of Cartography), *Mapoteca Topográfica Digital (Digital Topographic Map Library including database structure and the specifications of standard features and its attributes)*, DOCUMENTAÇÃO INTERNA - VERSÃO 04 (Internal paper in development - version 04), Rio de Janeiro, 1999.

[3] Secretariat of International Steering Committee for Global Mapping - ISCGM, *Report of the Fifth Meeting of International Steering Committee for Global Mapping, Canberra, Australia*, GSI, 1998.

[4] Secretariat of International Steering Committee for Global Mapping - ISCGM, *Explanation for Vmap Level 0 in ArcInfo Export format*, 1999.

[5] Bentley Systems Brasil (GeoExchange, Geographics, MicroStation-SE), <http://www.bentley.com.br>.

[6] ESRI – Environmental System Research Institute, Inc. (ArcInfo/ArcView), <http://www.esri.com>.

[7] Sisgraph Ltda. (Intergraph MGE)  
<http://www.sisgraph.com.br/>  
<http://intergraph.com/federal/products/imis/vpfps.asp>

[8] . Secretariat of International Steering Committee for Global Mapping - ISCGM, *Report of the Sixty Meeting of International Steering Committee for Global Mapping, Cambridge, England*, GSI, 1999.

**Table 2 - Example of MTD (Intergraph MGE/ACCESS) structure (Categories, Features, Attributes and Domains) for the CIM Project category **HIDROGRAFIA** (hydrography, **drainage**)**

Feature Name	Feat. Code	Feat. Attribute Table	Feature Type	lv	co	s	w	Attribute Field	Do Nr	Domain Name	Attribute Value Table	Att. Val	Value Description
CONTORNO HD PERMANENTE LIM (permanent water body)	04302		area boundary	1	71	0	1						
CONTORNO HD PERMANENTE (permanent water body)	04402	tbhd_mas_agua	area centroid	1	0	0	0	mslink	0				
								mapid	0				
								nm_nome (name)	0				
								nm_rio_associado (river name)	0				
								md_ar_poli (area)	0				
								cd_classificacao (water body type)	14	HDDOM01	hdclassificacao	1	baia (bay)
												2	rio (river)
												3	furo (hole)
												4	parana (estuary)
												5	vala (ditch)
												6	oceano (ocean)
												7	canal (canal)
												8	lago (lake)
												9	lagoa (lagoon)
												10	represa (dam)
								cd_navegabilidade (navigation)	15	HDDOM02	hdnavegabilidade	1	navegavel (navigable)
												2	não navegavel (not navigable)
												3	nao identificado (unidentified)
								cd_fluxo (flow)	16	HDDOM03	hdfluxo	1	permanente (permanent)
												2	intermitente (intermittent)
												3	nao identificado (unidentified)
ILHA (island)	04403	tbhd_ilha	area centroid	54	117	0	0	mslink	0				
								mapid	0				
								nm_nome_ocor (name)	0				
								md_ar_poli (area)	0				
								cd_tipo_ilha (island type)	22	HDDOM09	hdtipoilha	1	fluvial (river island)
												2	maritima (sea island)
												3	outras (other)
												4	nao identificado (unidentified)
DELIMITADOR HD	04599		area boundary	62	23	5	0						

**Table 8 - Example of Global Map CIM Project (Intergraph MGE/ACCESS) structure (Categories, Features, Attributes and Domains) for the Global Map coverage **DRAINAGE****

Feature Name	Feature Code	Feature Attribute Table	Feature Type	lv	co	st	we	Attribute Field	Do Nr	Domain Name	Attribute Value Table	Attribute Value	Value Description
island	04403	tbdr_island	point	54	117	0	0	mslink	0				
								mapid	0				
								f_code	0			BA030	
								nm_nam_island	0				
inland water	04302	tbdr_inland_water	face	2	72	0	1	mslink	0				
								mapid	0				
								f_code	0			BH001	
								nm_nam_inland_wate	0				
								cd_hyc	23	DRDOM01	hydrologicalcateg	0	unknown
												6	non-perennial /intermittent/fluctuating
												8	perennial/permanent
water border	04599		edge	62	23	5	0						
water text	04601		text	61	61	0	0						



# Fatores Determinantes de Desempenho de Métodos de Acesso Multidimensionais

RICARDO RODRIGUES CIFERRI<sup>1</sup>  
ANA CAROLINA SALGADO<sup>2</sup>

<sup>1</sup>DIN/UEM – Departamento de Informática da Universidade Estadual de Maringá,  
Av. Colombo, 5790, Zona 07, 87020-900 Maringá, PR, Brasil

[rrc@din.uem.br](mailto:rrc@din.uem.br)

<sup>2</sup>CIn/UFPE – Centro de Informática da Universidade Federal de Pernambuco,  
Caixa Postal 7851, Cidade Universitária, 50732-970 Recife, PE, Brasil

[acs@cin.ufpe.br](mailto:acs@cin.ufpe.br)

**Abstract:** Spatial data base systems (SDBSs) deal with data that are special in nature and size. Thus, the technologies developed for conventional data base systems such as physical design, access methods, query optimizers and languages, have to be modified in order to satisfy the needs of a SDBS. These modifications, embedded in several SDBSs, or being proposed by research projects, need to be evaluated. Our research focuses specifically multidimensional access methods (MAMs), which are very important structures used to diminish the execution time of search operations in a SDBS. In this paper, we discuss two important factors that affect the performance of MAMs: degree of overlap between non-zero size data objects and data selectivity.

## 1 Introdução

A manipulação de dados espaciais, tais como pontos, linhas, polígonos e sólidos tridimensionais, é um requisito fundamental para inúmeras aplicações atuais, as quais incluem aplicações geográficas de gerenciamento de serviços de utilidade pública (telefonia, eletricidade, água e esgoto), aplicações de planejamento e administração cadastral urbano e rural, aplicações ambientais, aplicações CAD/CAM (projeto e fabricação auxiliados por computador), aplicações cartográficas e aplicações de biotecnologia.

O gerenciamento em memória secundária de dados que possuem extensão e/ou localização no espaço é tarefa específica de uma classe particular de sistemas de banco de dados (SBDs), denominada de SBDs espaciais (SBDEs). Sistemas de banco de dados espaciais estendem as funcionalidades típicas de gerenciamento de dados alfanuméricos em disco, oferecendo suporte integrado a tipos de dados espaciais em seu modelo de dados, linguagem de consulta e implementação física.

Dentre as várias estruturas utilizadas por um SBDE, métodos de acesso multidimensionais (MAMs) destacam-se por permitirem que o desempenho na recuperação de dados seja melhorado significativamente. Estes métodos, também conhecidos como mecanismos de indexação espacial, organizam o espaço multidimensional e os objetos contidos neste espaço de tal forma que somente algumas partes do espaço e um subconjunto dos objetos espaciais armazenados sejam considerados para responder uma dada consulta espacial. Esta organização é fundamental, devido ao rápido crescimento do volume de dados armazenado em SBDEs. Os bancos de dados do sistema *Earth Observing System* da NASA e do projeto SEQUOIA 2000,

por exemplo, armazenarão respectivamente, quando finalizada a fase de coleta de dados, 10 Pbytes ( $10^{16}$ ) e 100 Tbytes ( $10^{14}$ ) de dados.

Após duas décadas de pesquisa, diversos MAMs foram desenvolvidos, os quais são voltados principalmente para a indexação de pontos e retângulos. Gaede e Günther [8] realizaram uma extensiva revisão bibliográfica dos métodos de acesso propostos até 1996. Este levantamento revelou a existência de aproximadamente 50 métodos de acesso. Infelizmente, esforço compatível não tem sido verificado na comparação de desempenho de MAMs. Poucos trabalhos efetuaram testes de desempenho comparativos, sendo que estes trabalhos abrangeram apenas um pequeno subconjunto dos métodos de acesso, com ênfase na família R-tree. Ademais, alguns trabalhos mostram-se limitados na investigação de certos fatores determinantes de desempenho, isto é, fatores que influenciam diretamente o desempenho dos MAMs no suporte a consultas espaciais e no suporte a operações de inserção, remoção e modificação de dados.

A pesquisa sendo desenvolvida pelos autores deste artigo está direcionada para a análise de desempenho de métodos de acesso multidimensionais através do uso da técnica experimental de *benchmark* de banco de dados. Neste sentido, primeiramente foram identificados os principais fatores determinantes de desempenho (Tabela 1). Em seguida, foi feito um levantamento de trabalhos anteriores voltados à investigação e comparação da variação do desempenho de MAMs. Após este levantamento, foi realizada uma análise crítica e comparativa dos diversos trabalhos correlatos em função de cada um dos fatores determinantes de desempenho. Vale destacar que a referida análise crítica visou identificar problemas e limitações nas metodologias usadas para investigar a influência dos



fatores na eficiência dos MAMs. Tanto o levantamento de trabalhos correlatos, quanto a análise crítica e comparativa podem ser encontrados em Ciferri e Salgado [5].

A análise de desempenho pretende abranger diversos MAMs, tais como R-tree com algoritmos de particionamento de nó quadrático e linear, R-tree Greene, R<sup>+</sup>-tree, R<sup>\*</sup>-tree, Hilbert R-tree, cell tree with oversize shelves, PMR-quadtrees e M-tree. Um ambiente de teste deve incluir: (1) um conjunto representativo de arquivos de dados reais e sintéticos (isto é, gerados artificialmente); (2) uma interface funcional que permita a comunicação entre usuários finais e a ferramenta de *benchmark* e consequentemente torne possível a parametrização dos fatores determinantes de desempenho e (3) mecanismos para a visualização estatística dos resultados de desempenho coletados. Nosso ambiente de teste, além de atender a estes requisitos, medirá o desempenho segundo diferentes perspectivas, ou seja, através da utilização de diferentes tipos de medidas de desempenho (acessos a disco, operações geométricas, tempo de utilização da UCP, espaço de armazenamento, tempo total gasto, etc.).

No estágio atual de nossa pesquisa, somente o fator distribuição espacial dos dados foi investigado para um subconjunto dos métodos de acesso anteriormente citados (métodos de acesso da família R-tree). Para isto, foi proposta uma metodologia que permite a geração de um

conjunto de distribuições de dados com diferentes características, as quais tornam possível que a influência do fator distribuição espacial dos dados seja analisada sob perspectivas distintas, desde uma reduzida até uma acentuada influência no desempenho dos MAMs. Como resultado, confirmou-se que a distribuição espacial dos dados exerce grande influência no desempenho tanto absoluto quanto relativo de MAMs. Ademais, em especial, foram verificadas novas relações de desempenho, algumas das quais contrariam resultados de desempenho e conclusões obtidas em trabalhos anteriores, tais como em Cox Júnior [7] e em Carneiro [3]. Maiores detalhes sobre a investigação do fator distribuição espacial dos dados podem ser encontrados em [6].

Este artigo discute especificamente aspectos relacionados com os fatores determinantes de desempenho “grau de sobreposição entre objetos espaciais de dimensão não-zero” e “seletividade dos dados”. Estes dois fatores, apesar de importantes, tem recebido um tratamento superficial em investigações anteriores. A próxima seção discute aspectos relacionados com o grau de sobreposição entre objetos espaciais de dimensão não-zero. O controle da seletividade dos dados e as diversas aplicações deste controle são discutidos na seção 3. Já a seção 4 apresenta brevemente algumas características de trabalhos correlatos. Por fim, a seção 5 apresenta as conclusões.

classes de fatores	fatores determinantes de desempenho
relacionados aos dados	tipo de dado, distribuição espacial dos dados, volume (quantidade) e escalabilidade, tamanho e formato de objetos espaciais de dimensão não-zero, grau de sobreposição entre objetos espaciais de dimensão não-zero e seqüência de inserção dos dados.
relacionados à carga de trabalho	tipo de consulta (ex.: <i>nearest neighbour query</i> ), características associadas a parâmetros de consulta (tais como tamanho, formato e distribuição espacial dos retângulos relativos às janelas de consulta de <i>range queries</i> ou distribuição espacial dos pontos base de consultas do tipo <i>point query</i> ), dinâmica dos dados (operações de inserção, remoção e modificação) e seletividade dos dados.
relacionados ao gerenciamento de <i>buffer-pool</i>	política de gerenciamento, tamanho total do <i>buffer</i> e tamanho da página de disco. A variação deste último fator conduz, para um mesmo MAM, a diferentes agrupamentos de objetos. Já os demais fatores determinam quais e quantos objetos estarão presentes em memória principal, requisitos fundamentais para otimizar os acessos a disco.
relacionados a parâmetros da estrutura	a variação de alguns parâmetros, tal como o número mínimo de entradas por nó para os MAMs R-tree e R <sup>*</sup> -tree, permite um ajuste diferenciado na estrutura de dados e por conseguinte um melhor ou pior desempenho.

**Tabela 1 Fatores determinantes de desempenho de métodos de acesso multidimensionais**

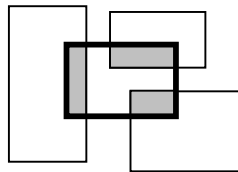
## 2 Grau de Sobreposição entre Objetos Espaciais de Dimensão não-zero

Diversos trabalhos têm destacado a influência da sobreposição entre objetos espaciais no desempenho de MAMs [8, 9, 14]. Greene [9], por exemplo, concluiu que o método R<sup>+</sup>-tree produz um melhor desempenho quando ocorrem

menos sobreposições entre objetos espaciais de dimensão não-zero. Já Kriegel *et al* [14], em suas considerações finais, destacam algumas limitações presentes na avaliação de desempenho realizada pelos próprios autores, dentre as quais encontra-se a ausência de testes de desempenho que investigassem os efeitos da variação do grau de sobreposição. A caracterização exata do termo “grau de sobreposi-

ção”, no entanto, não foi efetuada por Kriegel *et all*, nem pelos demais trabalhos correlatos estudados.

O grau de sobreposição, segundo o entendimento dos autores deste trabalho, pode ser definido a partir de duas medidas. A primeira consiste na média do número de objetos intersectados por cada objeto espacial armazenado em um arquivo de dados. Outra medida possível para contabilizar o grau de sobreposição é baseada na extensão espacial envolvida. Para retângulos no espaço bidimensional, por exemplo, esta medida deve ser calculada em dois passos. Inicialmente, para cada objeto espacial deve-se calcular a fração de área envolvida com sobreposições, sendo que esta fração deve ser quantificada através de um percentual de área relativo à área total do *extent*. Por fim, calcula-se a média das frações. Um alto grau de sobreposição corresponde, portanto, a um grande número de objetos intersectados e a uma vasta área de sobreposição. A Figura 1 ilustra um exemplo no qual um dado objeto espacial intersecta três outros objetos, sendo que a fração de área envolvida com sobreposições está em cinza.



**Figura 1 Exemplo de sobreposição**

Ao invés de controlar o grau de sobreposição segundo as medidas acima citadas, por exemplo na geração de dados sintéticos, alguns trabalhos correlatos apresentaram apenas uma estimativa do grau de sobreposição médio presente nos vários arquivos de dados através da medida "densidade de sobreposição" [1, 9, 11, 13]. Esta medida indica quantas vezes o *extent* é coberto pela *coverage* (área total) de todos os objetos espaciais, e pode ser calculada pela Equação 1, onde *n* corresponde ao número de objetos espaciais. A *coverage* de todos os objetos espaciais, de modo equivalente, também pode ser calculada através da expressão *n* \* tamanho médio dos objetos.

$$Densidade = \frac{\sum_{i=1}^n tamanho\_objeto(i)}{tamanho\_extent}$$

**Equação 1 Densidade de sobreposição**

A medida densidade de sobreposição, segundo alguns pesquisadores [10, 15], é um bom indicador do grau de sobreposição, sendo que valores crescentes de densidade tendem a indicar um maior grau de sobreposição. Neste sentido, Günther *et all* [11] apresenta alguns exemplos com densidades distintas, sendo que para uma densidade de sobreposição de 100% foi verificado que sobreposições entre objetos espaciais ocorriam, mas em um grau não

muito elevado, enquanto que uma densidade de 5% tornava praticamente inexistente a ocorrência de sobreposições entre objetos espaciais. A Tabela 2 apresenta as densidades consideradas em alguns trabalhos.

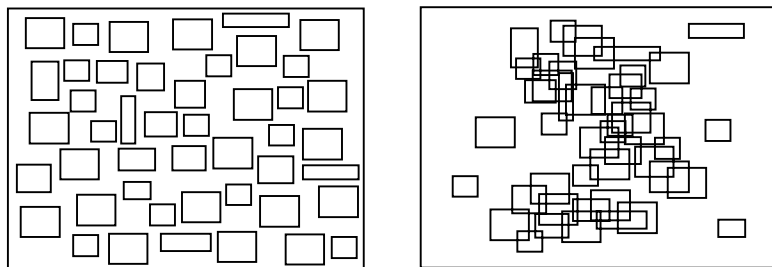
trabalhos correlatos voltados à análise de desempenho de MAMs	densidade de sobreposição
Greene [9]	0,0001; 0,01; 1 e 10
Beckmann <i>et all</i> [1]	2; 2,5; 8; 10 e 11,2
Kamel e Faloutsos [13]	entre 0,029 e 1
Günther <i>et all</i> [11]	0,05; 0,3 e 1

**Tabela 2 trabalhos correlatos**

A medida densidade de sobreposição, entretanto, possui como grande limitação o fato de não ser muito precisa, sendo que situações completamente antagônicas podem apresentar o mesmo valor de densidade. Como exemplo, a Figura 2 ilustra o conteúdo de dois arquivos de dados com a mesma densidade de sobreposição, mas claramente contendo diferentes graus de sobreposição. Vale destacar que, em um destes arquivos, propositadamente, não ocorre nenhuma sobreposição. Desta forma, pode-se concluir que o uso isolado da medida densidade de sobreposição na investigação da variação do desempenho de MAMs em função do grau de sobreposição não é apropriado, sendo necessário o uso de um conjunto alternativo de medidas que caracterize de uma forma mais precisa o grau de sobreposição.

Ainda há espaço, portanto, para novas investigações sobre a influência da sobreposição entre objetos espaciais no desempenho de MAMs, a qual deve ser conduzida de um modo controlado e exaustivo. A obtenção de diferentes valores para o grau de sobreposição pode ser alcançada através da variação de outros fatores determinantes de desempenho, os quais influenciam indiretamente no grau de sobreposição. Cox Júnior [7] observou que conjuntos de dados com objetos grandes, por exemplo, tendem a gerar sobreposições e provocam alterações de desempenho distintas por parte dos métodos de acesso. A distribuição espacial dos dados, por sua vez, também influencia no grau de sobreposição, sendo um bom exemplo ilustrado na Figura 2. Ooi [15] verificou que quando os objetos estão distribuídos de uma forma mais uniforme, o grau de sobreposição é menor.

Vale destacar, entretanto, que a realização de testes que investigam isoladamente a variação do desempenho em função da distribuição espacial dos dados ou em função do tamanho dos objetos espaciais não proporcionam qualquer conclusão independente com relação à sobreposição entre objetos espaciais, apesar destes testes também conduzirem à variação do grau de sobreposição.



**Figura 2 Exemplo de arquivos com a mesma densidade de sobreposição, mas contendo diferentes graus de sobreposição**

A variação da distribuição espacial dos dados para dois arquivos de um mesmo tipo<sup>1</sup>, por exemplo, em geral conduz à variação do grau de sobreposição, tal como um arquivo de dados com distribuição altamente correlacionada apresentando um grau de sobreposição cinquenta vezes maior do que um arquivo com distribuição uniforme. Neste sentido, pode-se concluir, incorretamente, que um método de acesso multidimensional *A* apresenta um desempenho superior ao método de acesso *B* quando o grau de sobreposição é alto, ao passo que para um baixo grau de sobreposição o método de acesso *A* apresenta os piores resultados. Neste caso específico, entretanto, o desempenho é influenciado prioritariamente pela distribuição espacial dos dados, sendo que a influência do grau de sobreposição pode ser considerada como secundária. Observação similar pode ser feita no caso da variação do tamanho dos objetos, com destaque para a manutenção da distribuição espacial dos dados.

Para verificar de fato qual a influência da sobreposição entre objetos espaciais no desempenho de métodos de acesso multidimensionais deve-se variar conjuntamente ambos os fatores distribuição espacial dos dados e tamanho dos objetos espaciais, os quais devem ser usados para controlar o grau de sobreposição. Uma possível estratégia consiste da geração de um conjunto de pares de arquivos, sendo que cada par possui um grau de sobreposição distinto, mas os arquivos de um dado par possuem o mesmo grau de sobreposição (ou muito próximos) e são formados por diferentes distribuições de dados e tamanhos de objetos. Como exemplo, pode-se citar a geração do par de arquivos *A* e *B*, com o arquivo *A* constituído de objetos pequenos sob distribuição altamente correlacionada e o arquivo *B* formado por objetos grandes distribuídos de modo uniforme no espaço, mas ambos arquivos apresentando o mesmo grau de sobreposição.

Caso o desempenho, para um determinado método de acesso, seja semelhante para os arquivos de dados de um mesmo par e significativamente diferente entre os

vários pares de arquivos<sup>2</sup>, comprova-se que o desempenho foi influenciado basicamente pela sobreposição dos dados (e independente, de certo modo, da distribuição espacial dos dados e do tamanho dos objetos espaciais). Por outro lado, a constatação freqüente, para diferentes métodos de acesso, de um desempenho muito diferenciado para arquivos de dados de um mesmo par ou de um desempenho similar para pares com graus de sobreposição muito distintos (ausência, baixo, médio e alto, por exemplo), pode indicar que o grau de sobreposição não constitui em um fator determinante de desempenho para MAMs.

### 3 Controle da Seletividade dos Dados

A seletividade dos dados corresponde ao percentual de dados recuperados por uma consulta com relação ao conjunto completo de possíveis respostas à consulta. No caso de consultas espaciais de seleção, este percentual é calculado, portanto, com base na quantidade total de objetos espaciais armazenados no arquivo de dados sobre o qual a consulta foi executada, enquanto para consultas do tipo junção espacial, este percentual é calculado com base no número de tuplas do produto cartesiano de dois arquivos de dados.

O controle da seletividade dos dados, isto é, a possibilidade de se recuperar somente uma certa percentagem dos dados, como exemplo 1%, 5%, 15% ou 30%, é muito importante, dado que uma consulta pode gerar um desempenho não proporcional ao percentual de dados recuperados [4]. Uma consulta, logicamente, pode gerar um desempenho distinto em função do percentual de dados recuperados, ocorrendo de modo geral uma diminuição no desempenho à medida que a seletividade dos dados aumenta, ou seja, à medida que cresce o volume de dados recuperados. Entretanto, apesar da diferença de desempenho obtida com a utilização de graus de seletividade distintos, espera-se que a razão entre os desempenhos seja proporcional à razão entre os graus de seletividade.

<sup>1</sup> arquivos com características equivalentes quanto a tipo de dado, volume, tamanho e formato dos objetos espaciais.

<sup>2</sup> deve-se usar graus de sobreposição com certa diferença, de modo que a variação não reflita uma variação contínua.

Como exemplo, se a recuperação de 10% dos objetos espaciais armazenados em um arquivo de dados gastar 5 segundos, a recuperação de 30% e 50% dos objetos espaciais deve gastar respectivamente em torno de 15 e 25 segundos. Quando a recuperação de 30% dos objetos espaciais consumir, por exemplo, 30 segundos, verifica-se um desempenho não proporcional ao percentual de dados recuperados. Em especial, Günther *et al* [11] destaca que a seletividade dos dados constitui em um excelente indicador de variações no desempenho relativo. Para sistemas gerenciadores de banco de dados relacionais, o estudo da variação do desempenho em função do percentual de dados recuperados por uma consulta tem sido largamente realizado, com destaque para o uso do *benchmark* de Wisconsin.

A investigação da variação do desempenho de métodos de acesso em função do percentual de dados recuperados por uma consulta é tanto viável quanto particularmente interessante. Viável no sentido que alguns trabalhos correlatos já constataram, de forma indireta, que o aumento do grau de seletividade dos dados pode alterar o desempenho relativo dos métodos de acesso, assim como pode alterar o desempenho absoluto de um MAM. Esta constatação foi feita a partir da variação do tamanho da janela de consulta de *intersection range queries*. O crescimento do tamanho da janela de consulta implica, indiretamente, em um aumento do grau de seletividade para a maioria absoluta das situações<sup>3</sup>. Como exemplo, Carneiro [3] verificou que a diferença de desempenho entre as variantes da  $R^*$ -tree e da R-tree tende a cair com o aumento do tamanho da janela de consulta. A simples variação do tamanho da janela de consulta não permite, entretanto, um controle preciso do grau de seletividade. Para isto, adicionalmente é necessário uma distribuição sistemática dos dados e das janelas de consulta.

Dentre os trabalhos correlatos revisados, apenas dois trabalhos consideraram de algum modo o fator seletividade dos dados nos testes de desempenho. O trabalho de Guttman [12] não investigou a variação do desempenho do método de acesso R-tree em função do percentual de dados recuperados por uma consulta, e por conseguinte não controlou a variação do grau de seletividade dos dados, limitando-se apenas a constatar que a execução das consultas, no caso *intersection range query*, conduzia a uma seletividade entre 3% e 6% do total de objetos espaciais armazenados nos arquivos de dados.

Já o trabalho de Günther *et al* [11] destaca-se por ter investigado a variação do desempenho de estratégias

---

<sup>3</sup> considerando que as janelas possuem o mesmo formato e distribuição espacial do centro, diferindo somente com relação ao tamanho. Assim, a seletividade dos dados para duas janelas com mesmo centro, na pior hipótese, será igual.

distintas de processamento de junção espacial em função do percentual de dados recuperados por vários subtipos de junção espacial, cada qual com um predicado específico, tais como *distance minor-equal*, *intersection* e *direction northwest*. Apesar do enfoque dado aos testes de desempenho não ter sido direcionado para métodos de acesso, a existência de estratégias fundamentadas no uso de MAMs, tais como *scan-and-index* e *synchronized tree traversal*, permitiu uma avaliação do método de acesso quadtree, escolhido para dar suporte a estas estratégias.

Em especial, verifica-se que ainda existe espaço para se realizar novas pesquisas com métodos de acesso multidimensionais envolvendo o fator seletividade dos dados. Neste sentido, pode-se realizar testes que:

- investiguem de forma precisa e controlada o comportamento de métodos de acesso específicos, tais como os MAMs da família R-tree, frente a diferentes graus de seletividade dos dados. Em particular, pode-se verificar se o desempenho do método de acesso em questão é proporcional ou não ao percentual de dados recuperados por uma consulta. Quanto ao tipo de consulta, pode-se destacar consultas do tipo *range query*, uma vez que este tipo de consulta tem sido amplamente investigado na literatura e possui grande popularidade, por exemplo, entre usuários de aplicações georeferenciadas;
- investiguem a variação do desempenho relativo de MAMs em função do percentual de dados recuperados por uma consulta. Para métodos de acesso com estrutura na forma de uma árvore de múltiplos caminhos (*multiway tree*), por exemplo, o agrupamento de objetos espaciais em nós folhas deve ser feito de tal maneira que um baixo grau de seletividade (tal como 1% do número total de objetos espaciais indexados) conduza ao acesso a um número reduzido de nós, especialmente de nós folhas. Caso contrário, pode ocorrer uma degradação excessiva no desempenho. Já para um alto grau de seletividade, para o qual ocorre a recuperação de grande parte dos objetos indexados (como exemplo, 75%), espera-se o acesso a uma grande quantidade de nós, sendo a eficiência do agrupamento de objetos espaciais menos importante neste caso. Assim, pode-se verificar o desempenho relativo de vários MAMs tanto para um baixo grau de seletividade dos dados, para o qual é esperado uma diferença significativa no desempenho, quanto para um alto grau de seletividade dos dados, para o qual espera-se desempenhos semelhantes;
- investiguem a influência que a distribuição espacial dos dados exerce na seletividade dos dados. Isto pode ser conseguido através da recuperação de um mesmo percentual de dados para regiões distintas do *extent*. Como resultado, poderá ocorrer duas possíveis situações. Na primeira situação, o desempenho é determinado predominantemente pelo grau de seletividade e portanto praticamente o mesmo

quando consideradas regiões distintas do *extent*. Já na segunda situação, verifica-se que o desempenho difere de forma significativa de uma região para outra, apesar do percentual de dados recuperados pela consulta ser o mesmo. Neste caso, em particular, pode-se dizer que o desempenho de um método de acesso, para um dado grau de seletividade, é influenciado pela distribuição espacial dos dados;

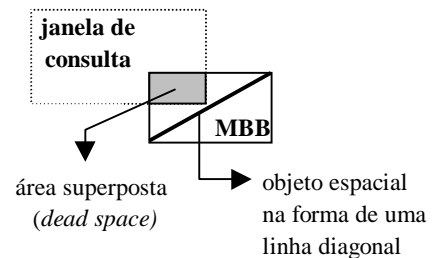
A seletividade dos dados também destaca-se pela função que exerce no suporte à escolha de um plano de execução por parte de um otimizador de consultas. Este fator é comumente estimado para determinar a escolha de planos de execução que usam índices convencionais ou métodos de acesso multidimensionais. Dependendo da seletividade estimada dos dados, o otimizador de consultas de um sistema gerenciador de banco de dados (SGBD) pode escolher, por exemplo, entre o uso de um índice ou a realização de uma busca seqüencial. Em geral, para graus de seletividade muito altos, o uso de índices não é recomendado, uma vez que a sobrecarga de execução causada por estas estruturas supera qualquer ganho de desempenho potencial. Neste caso, a realização de uma busca seqüencial gera um melhor desempenho.

A escolha de uma estratégia para processamento de consultas em um SGBD espacial deve considerar ambos os predicados espacial e convencional. Neste sentido, o otimizador de consultas deve primeiramente gerar diferentes planos de execução, os quais definem a ordem de processamento das diversas subconsultas associadas aos predicados, sendo que em alguns casos a ordem de processamento restringe o conjunto de dados no qual uma subconsulta posterior irá atuar. Com base em uma estimativa de custo associado a cada plano, o otimizador de consultas deve então escolher o plano com menor custo. Câmara, *et all* [2] apresenta e discute diferentes planos de execução para consultas com ambos os tipos de predicado espacial e convencional, com o predicado espacial abrangendo especificamente consultas do tipo *region query* e junção espacial.

Há casos, no entanto, que uma consulta é constituída apenas por um predicado espacial. Um típico exemplo é "selecione todos os municípios que estão localizados na zona da mata de Pernambuco" (*containment region query*). Nestes casos, a escolha do otimizador de consulta resume-se basicamente a dois planos: (1) o uso de um método de acesso multidimensional ou (2) a varredura completa do arquivo de dados.

No **primeiro plano**, assume-se, claro, que o respectivo arquivo de dados encontra-se indexado. Vale destacar que este plano não se restringe unicamente ao uso de um método de acesso, sendo adicionalmente necessário uma fase posterior de processamento. Tal necessidade é conse-

qüência direta do fato que métodos de acesso representam a geometria de objetos espaciais de dimensão não-zero, como exemplo uma linha ou um polígono, através de uma aproximação, visando diminuir tanto o custo de armazenamento quanto o custo de processamento para se determinar a satisfação dos relacionamentos espaciais. As aproximações, tal como MBB, garantem que nenhum dos objetos espaciais que satisfazem o relacionamento em questão seja desconsiderado na resposta da consulta, mas em contrapartida permitem a recuperação de objetos espaciais que não satisfazem o relacionamento espacial (chamados de falsos candidatos), uma vez que a determinação do relacionamento pode ser decidida com base em um espaço, contido dentro da aproximação, que não faz parte da geometria do objeto (*dead space* – Figura 3). Assim, pode-se dizer que MAMs são imprecisos, no sentido que estes não retornam a resposta final e exata das consultas, retornando ao invés disto um superconjunto de candidatos. Na literatura, esta fase é comumente denominada de filtragem, uma vez que um MAM apenas filtra (ou descarta) os objetos que certamente não satisfazem o relacionamento espacial em questão.



**Figura 3 Recuperação de um falso candidato**

Devido à presença de falsos candidatos no conjunto de objetos selecionados, é necessário uma fase posterior de processamento, chamada de refinamento, na qual é verificado, para cada objeto espacial candidato, se o relacionamento espacial é satisfeito com relação à geometria exata do objeto, sendo descartados quaisquer falsos candidatos. A fase de refinamento é altamente custosa, pois requer tanto o acesso à geometria exata dos objetos espaciais, e portanto a transferência para memória principal de uma quantidade relativamente grande de dados, quanto a realização de cálculos geométricos complexos para se determinar a satisfação do relacionamento espacial. O custo do primeiro plano deve ser calculado, portanto, a partir dos custos de ambas as fases de filtragem e refinamento.

Já o **segundo plano** resume-se à aplicação direta da fase de refinamento para todos os objetos espaciais armazenados no arquivo de dados (estratégia equivalente a uma busca seqüencial). Desta forma, para cada objeto espacial é necessário o acesso à sua geometria exata e a realização

subsequente de cálculos geométricos sobre esta geometria para se determinar a satisfação do relacionamento espacial.

A escolha por parte do otimizador de consultas entre os dois planos de execução anteriormente citados é altamente influenciada pela seletividade estimada dos dados, a qual exerce grande peso na determinação do custo dos planos. A escolha do primeiro plano, e portanto o uso de um MAM, é baseado nas seguintes premissas: (1) a fase de refinamento é altamente custosa e deve ser aplicada a uma quantidade extremamente reduzida de objetos espaciais; (2) a fase de filtragem, por sua vez, é várias vezes menos custosa do que a fase de refinamento e assim, deve ser usada para restringir o número de objetos (candidatos) a serem analisados na fase de refinamento. Desta forma, a fase de filtragem serve para diminuir o custo da fase de refinamento, a qual basicamente determina o desempenho de uma consulta.

A escolha do primeiro plano é válida, no entanto, somente se a fase de filtragem realmente reduzir drasticamente o número de objetos a serem analisados na fase de refinamento, ou seja, se o grau de seletividade for baixo. Neste caso, a soma dos custos de ambas as fases de filtragem e refinamento tende a ser menor do que o custo resultante da aplicação direta da fase de refinamento para todos os objetos espaciais armazenados no arquivo de dados (segundo plano), devido principalmente à diferença significativa nos custos da fase de refinamento de ambos os planos. Por outro lado, se o grau de seletividade for muito alto, esta diferença torna-se pequena, podendo até ser superada pelo custo associado à fase de filtragem do primeiro plano de execução, o que resultaria em um melhor desempenho para o segundo plano.

Visando proporcionar heurísticas confiáveis para otimizadores de consulta espaciais, surge então a necessidade de se determinar para quais graus de seletividade dos dados a utilização de um método de acesso multidimensional não torna-se desvantajosa. Para isto, deve-se investigar o desempenho de ambos os planos de execução segundo diferentes graus de seletividade dos dados, desde graus reduzidos (como exemplo 1%) até graus elevados de seletividade (tais como, 80% ou 90%). Vale destacar, entretanto, que o ponto de equilíbrio (ou seja, o grau de seletividade dos dados segundo o qual ambos os planos proporcionam o mesmo desempenho no processamento de um dado tipo de consulta espacial) varia em função do método de acesso escolhido para dar suporte ao primeiro plano, sendo a investigação voltada a um tipo específico de MAM. Neste sentido, a análise de diversos métodos de acesso é particularmente interessante.

De modo análogo à discussão realizada para consultas espaciais de seleção, pode-se também investigar sobre o ponto de equilíbrio no processamento de consultas do

tipo junção espacial, ou seja, identificar o grau de seletividade segundo o qual o uso de uma estratégia baseada em MAM, tal como *scan-and-index*, produz o mesmo desempenho que o processamento completo do produto cartesiano de dois conjuntos de dados espaciais.

#### 4 Trabalhos Correlatos

Guttman [12] realizou testes de desempenho exclusivamente voltados para a validação e ajuste da estrutura R-tree, sendo considerados diferentes valores para os parâmetros  $M$  e  $m$  (número máximo e mínimo de entradas por nó, respectivamente) e diferentes algoritmos de particionamento de nó (exaustivo, quadrático e linear).

Kriegel *et al* [14] realizaram testes específicos tanto para métodos de acesso a pontos (2-level grid file, BANG file, hB-tree e BUDDY hash tree), quanto para métodos de acesso espaciais (R-tree e PLOP hashing). Esse trabalho possui como característica principal o uso de diversos tipos de distribuição de dados, com destaque para distribuições altamente correlacionadas.

O principal diferencial dos testes realizados por Greene [9] está relacionado com o fato de que os MAMs analisados (R-tree,  $R^+$ -tree, K-D-B-tree e 2D-ISAM) foram implementados no topo do SGDB POSTGRES através do uso de procedimentos especiais com acesso as funções da camada de abstração de métodos de acesso deste SGBD.

Beckmann *et al* [1] realizaram testes para ajustar, validar e comparar a estrutura proposta  $R^*$ -tree com outros métodos de acesso, a saber: R-tree com algoritmos de particionamento de nó quadrático e linear, R-tree Greene e 2-level grid file. A execução dos testes foi efetuada em três etapas distintas. A primeira foi voltada à análise de desempenho dos MAMs da família R-tree no suporte a retângulos, mas considerando apenas consultas espaciais de seleção. Já a segunda gerou configurações de teste para consultas do tipo junção espacial. Por fim, a última etapa foi voltada à análise de desempenho no suporte a pontos.

Günther e Bilmes [10] realizaram testes de desempenho para comparar um tipo alternativo de MAM, chamado cell tree, com os métodos de acesso R-tree Greene e  $R^+$ -tree. A principal característica da cell tree é que esta estrutura provê suporte direto a objetos espaciais de dimensão não-zero com geometria arbitrária, em oposição ao suporte restrito e único de retângulos.

A estratégia utilizada nos testes realizados por Cox Júnior [7] é baseada na geração sintética de arquivos de dados e de consultas a partir de um conjunto de fatores determinantes de desempenho (tipo de dado, tipo de consulta, tamanho dos objetos, distribuição espacial e dinâmica dos dados). Em particular, foram analisados os métodos R-tree, R-tree Greene,  $R^+$ -tree,  $R^*$ -tree e uma variante desta

sem a rotina de reinserção de entradas. Carneiro [3] adaptou e estendeu a estratégia proposta em [7] visando o uso de um conjunto de dados reais representativos para aplicações de gerenciamento de serviços de utilidade pública.

Ooi [15] realizou testes de desempenho voltados para a validação, o ajuste e a comparação de desempenho da estrutura spatial kd-tree (skd-tree), para ambas versões estática e dinâmica. Na comparação, foram investigados os métodos de acesso multidimensionais R-tree, packed R-tree, 4d-tree e mkd-tree.

Kamel e Faloutsos [13] realizaram experimentos para comparar o método de acesso proposto Hilbert R-tree com relação aos métodos de acesso R-tree e R\*-tree. Na comparação dos métodos foi utilizada apenas uma única medida, número de acessos a disco, a qual foi usada principalmente na coleta de resultados de desempenho relativos à consultas do tipo *intersection range query*.

Por fim, Günther *et al* [11] investigaram o desempenho das estratégias *nested loop*, *synchronized tree traversal* e *scan-and-index* no suporte a consultas do tipo *intersection spatial join*, *containment spatial join*, *distance minor-equal spatial join*, *direction northwest spatial join* e *enclosure spatial join*.

## 5 Conclusão

Este artigo descreveu o atual estágio da pesquisa sendo desenvolvida pelos autores com relação a métodos de acesso multidimensionais. Ademais, também foram discutidos aspectos relacionados com os fatores determinantes de desempenho “grau de sobreposição entre objetos espaciais de dimensão não-zero” e “seletividade dos dados”.

Para os fatores acima citados, verificou-se a necessidade de novos testes de desempenho, os quais devem incorporar novas metodologias para investigar a influência destes fatores na eficiência dos MAMs. Para o fator “grau de sobreposição entre objetos espaciais de dimensão não-zero”, uma possível estratégia foi apresentada na seção 2. Já o controle da seletividade dos dados e as diversas aplicações deste controle foram discutidos na seção 3. Vale destacar que a pesquisa descrita neste artigo representa um primeiro passo para a definição de um *benchmark* de banco de dados voltado à análise de desempenho de MAMs.

## Referências Bibliográficas

[1] Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B. The R\*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proc. 1990 ACM SIGMOD Conference*, pp. 322-331, 1990.

[2] Câmara, G., Casanova, M.A., Hemerly, A.S., Magalhães, G.C., Medeiros, C.M.B. *Anatomia de Sistemas de Informação Geográfica*. 1996. 193pp.

[3] Carneiro, A.P. Análise de Desempenho de Métodos de Acesso Espaciais Baseada em um Banco de Dados Real. Master's thesis, IC, Unicamp, 1998.

[4] Ciferri, R.R. Um Benchmark voltado à Análise de Desempenho de Sistemas de Informações Geográficas. Master's thesis, Unicamp, 1995.

[5] Ciferri, R.R, Salgado, A.C. Análise de Desempenho de Métodos de Acesso Multidimensionais: Estado da Arte e Direções Futuras. Technical report, CIN/UFPE, 2000. 105 pp.

[6] Ciferri, R.R, Salgado, A.C., Cortez, S.S. Investigando a Variação do Desempenho de Métodos de Acesso Multidimensionais em Função da Distribuição Espacial dos Dados. Submetido para julgamento no SBBD'2000.

[7] Cox Junior, F.S. Análise de Métodos de Acesso a Dados Espaciais Aplicados a Sistemas Gerenciadores de Banco de Dados. Master's thesis, DCC, Unicamp, 1991.

[8] Gaede, V., Günther, O. Multidimensional Access Methods. *ACM Computing Surveys*, 30(2):170-231,1998.

[9] Greene, D. An Implementation and Performance Analysis of Spatial Data Access Methods. In *Proc. 5<sup>th</sup> IEEE ICDE*, pp. 606-615, USA, 1989.

[10] Günther, O., Bilmes, J. Tree-based Access Methods for Spatial Databases: Implementation and Performance Evaluation. *IEEE TKDE*, 3(3):342-356, September 1991.

[11] Günther, O., Oria, V., Picouet, P., Saglio, J.-M., Scholl, M. Benchmarking Spatial Joins À La Carte. In *Proc. 10<sup>th</sup> Conference on Scientific and Statistical Database*, pp. 32-41, Italy, 1998.

[12] Guttman, A. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proc. 1984 ACM SIGMOD Conference*, pp. 47-57, USA, June 1984.

[13] Kamel, I., Faloutsos, C. Hilbert R-tree: An Improved R-tree using Fractals. In *Proc. 20<sup>th</sup> VLDB Conference*, pp. 500-509, Chile, 1994.

[14] Kriegel, H.-P., Schiwietz, M., Schneider, R., Seeger, B. Performance Comparison of Point and Spatial Access Methods. In *Design and Implementation of Large Spatial Databases*, v. 409 of LNCS, pp. 89-114. 1989.

[15] Ooi, B.C. Efficient Query Processing in Geographic Information Systems. In Goos, G., Hartmanis, J., editors, volume 471 of LNCS, chapter 4, pp. 116-135. 1990.

# A comparison among different synchronized tree traversal algorithms for spatial joins

SANDRO DANILO GATTI<sup>1</sup>, GEOVANE CAYRES MAGALHÃES<sup>1</sup>,

<sup>1</sup>Institute of Computing, Univ. of Campinas - Caixa Postal 6176, 13083-970, Campinas, SP, Brasil  
{gatti, geovane}@dcc.unicamp.br

**Abstract.** Spatial join techniques for spatial access methods were analysed, considering factor such as buffer pool size, page size and intermediate join indexes ordering criteria. This analysis was based on real data, taken from GIS applications for utilities. Results of this work assess the way those factors affect spatial join performance and can be used for tuning the algorithms.

## 1 Introduction

We all have seen the great advances of informatics. New and different applications arise every day, followed by the development of faster machines and sophisticated software. In this context, we have geographic applications, applied in areas such as precision agriculture, urban planning and facilities (telecommunications), and so on. These areas are assisted by Geographic Information Systems (GIS), softwares that deal with spatial data.

Spatial data have characteristics that make them different from conventional data. That urges the development and tuning of algorithms that manage and use these data efficiently. Among these algorithms and operations, spatial joins can be included.

This work aims to evaluate some spatial join proposals based on buffer pool and page sizes, as well as intermediate join indexes ordering criteria, which affect enormously spatial joins performance. It takes an R\*-tree as the index on which the operation will be run.

This paper is organized as follows: section 2 presents works that influence this investigation; section 3 presents a description of the system used in the tests, the analysis criteria and the data used; the section 4 presents the results obtained in this work, as well as some discussions; and finally, the section 5 presents the conclusions of this work.

## 2 Related works

### 2.1 Spatial access methods

Many spatial access methods have been proposed and can be classified on different ways [1, 2]. This work takes as its base the R\*-tree [3], an evolution of Guttman's R-trees [4].

The R-tree based methods are structures derived from B-trees. By these, methods the space is hierarchically divided into regions, with the lower levels enclosed by higher ones, grouping the minimum bounding rectangles (MBR) of objects. Basically, R-trees have two kinds of nodes:

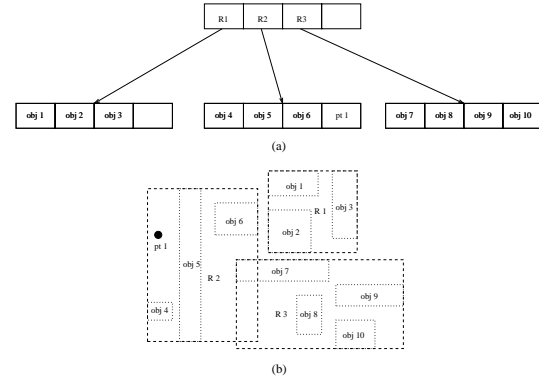


Figure 1: An R-tree (a) and the indexed points and rectangles (b).

- leaf nodes, which store information about the indexed objects in the form  $(id, MBR)$ , where the first field is the object identifier and the second one is the object MBR;
- non-leaf nodes, which contains information as  $(pointer, MBR)$ , where *pointer* is a pointer to a node of a lower level and *MBR* is an MBR that encloses the MBR of the children node.

Every node is stored in a disk page, which maximum capacity is  $M$  entries like  $(pointer, MBR)$  or  $(id, MBR)$ . The minimum number of entries is  $m \leq \frac{M}{2}$ .

Figure 1 shows the points and rectangles indexed by an R-tree. Note that the MBRs that belong to the same level can intersect. This structure is followed by conventional R-tree and by the R\*-tree. The difference between R-tree and R\*-tree remains in the overflow and insertion treatment algorithms, which, in the R\*-tree implementation, try to decrease the MBRs overlapping and coverage areas.



## 2.2 Spatial join methods

After the development of spatial access methods, research efforts turned to the development of more efficient algorithms of insertion, deletion and querying. But the most important for users are queries and, among queries, spatial joins deserve special attention.

Spatial joins are an important class of spatial query. This kind of query is used to answer questions like “Which public buildings are close to squares?” or “Which roads crosses river X?”.

These operations are analogous to relational joins, that use conventional data and also demand lots of computational efforts. But, instead of dealing with conventional data, spatial joins deal with spatial data and spatial predicates, such as “intersection”, “containment”, among others.

On this fact resides another difference between spatial join and relational joins: the last one frequently uses the equality criteria as comparison predicate, while the first one hardly uses equality, using more frequently intersection among objects. The predicates used on spatial joins hinders the use of algorithms for conventional joins.

This is initially due to the fact that spatial data do not have a total natural ordering which preserves spatial locality. Besides, conventional joins algorithms are optimized for equality. But spatial joins hardly ever use equality, which makes it difficult the use of conventional algorithms.

The researches in this area resulted in some methods, optimized for use when there are not indexes available on the entries, when only one entry is indexed, and when both entries are indexed.

When there are not indexes available, it is possible to apply a technique that uses partitioning, assigning tuples to buckets and using a plane-sweeping technique. Methods that use this approach are the PBSM (*Partition Based Spatial-Merge Join*) [5], the SSSJ (*Scalable Sweeping-Based Spatial Join*) [6], the *Spatial Hash-Join* [7] and another variant of this one, introduced by Zimbrão in [8].

If only one of the sets are indexed, it's possible to use an scan-and-index approach, or build a run-time index on the non-indexed set, and then run the join. Proposals which attack this problem are the *seeded-tree*, presented in [9], and the *SISJ* (*slot index spatial join*), presented in [10].

Finally, if both sets are indexed, a general join method can be applied, tuning it to the chosen access method. A grid-based access method could use some hashing technique and a tree-based access method could use a synchronized tree traversal (STT) technique [11, 12, 13].

Spatial joins, as well as other spatial queries, can be run in two steps: *filtering*, when candidates to the answer set are chosen and *refinement*, when the real shape of the object is retrieved and evaluated against the predicate. The approaches that should be paid attention to, when dealing with two indexed sets, are the STT methods.

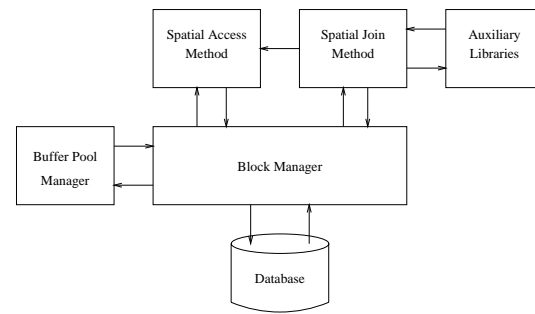


Figure 2: The system architecture.

## 3 Implementation

This work emphasizes time requirements, given in terms of I/O operations. The join methods chosen for implementation are the ones based on STT, presented in the following works [11], [12] and [13].

### 3.1 System architecture

A modularized system was implemented in order to ease methods evaluation, which gives total control of the operation to the researcher. It is formed by the disk abstraction module (DAM), the spatial join module (SJM), the index module (IM) and auxiliary libraries modules (ALM). These modules were implemented in the C language and the tests were run on a SUN SparcStation 20, with 128 megabytes of memory.

The index and disk abstraction modules were reused, with some modifications, from previous works [2, 14]. The system architecture is shown in Figure 2.

The DAM is the responsible for the interface between the system and the SJM, the IM and the operating system. It also controls a buffer pool and parameters such as buffer pool size, page size, amount of pages to be pinned in the buffer pool and page replacement policies. The IM keeps the trees indexing and data handling. By now, our work uses  $R^*$ -trees, but can also incorporate other indexing methods.

The SJM includes three spatial join methods: the *nested loop* (NL) [11], a *depth-first* (DF) [12], and a *breadth-first* (BF) [13]. The ALM keeps auxiliary functions, such as sorting, list manipulation, and other operations.

### 3.2 The data

The data used in our work is a real set, obtained from SAGRE Project (Sistema Automatizado de Gerência de Rede Externa) [15, 16, 17], developed by Centro de Pesquisa e Desenvolvimento (CPqD). This system automatizes processes related to management, planning, designing, expansion, among other operations, of an outside telecommunica-



Figure 3: MBRs referring to *city* set.

tion plant. They are representative in applications such as power distribution facilities, telephony, water distribution facilities, and others.

These data are constituted basically by a sequence of double float elements. Every four doubles form a MBR in the form  $(x_{min}, y_{min}, x_{max}, y_{max})$ . These data are kept in three different sets: the first contains points, named *poles*, the second contains rectangles, named *blocks* and the third contains points and rectangles, named *city*, shown in Figure 3.

The *poles* set contains 13,813 MBRs, the *blocks* set contains 2,473 MBRs and the *city* contains 66.837 MBRs, which includes the *poles* and *blocks* sets and also other elements, such as manholes, cables, pipelines, and others.

The data sets have non-uniform distribution and sizes, mainly the *city* set, which also have a high density. These sets were indexed into three distinct  $R^*$ -trees, one for the *poles* set, other for the *blocks* set and another for the *city* set. For each set, we have trees with the following page sizes: *1k*, *2k*, *4k* e *8 kbytes*.

### 3.3 Tested queries

All the queries were set to run a join between the *poles* and the *blocks* sets, which page sizes varying among *1k*, *2k*, *4k* e *8 kbytes*. Buffer pool sized varied among 2, 4, 8, 16, 24 e 32 pages. Different ordering criteria were used to sort the intermediate join indexes (IJI): an ordering by  $z$  curve (SZ), and the ordering obtained by the plane-sweeping technique for the DF method; the ordering obtained by the plane-sweeping technique (NS), sorting by  $x_{min}$  of one of the entries (SO), sorting by the sum of the  $x$  centers of the intersection MBRs (SS), sorting by the  $x$  center of the MBR that encloses the intersecting MBRs (SC), and an ordering

by a Hilbert curve (SH).

Every query was run 10 times, computing the average CPU time, from the same initial conditions. The buffer replacement policy was LRU (least recently used).

Other tests were also done, running a join between *blocks*  $\times$  *blocks* and *city*  $\times$  *city*. For this last queries, the buffer pool size varied from 2 to 256 disk pages.

## 4 Discussion and results

The results of this work are presented now, divided into CPU results, memory usage, and I/O results. Except when explicitly mentioned, all the results were obtained from *poles*  $\times$  *blocks* join.

### 4.1 CPU time and main memory usage

CPU time is not our main interest. But some measures were done in order to check the influence of the IJI ordering criteria and disk page size in spatial join performance. Figure 4 (a) shows a comparison between the three methods, not considering ordering times, while Figures 4 (b) and (c) shows the results for the DF and BF methods with sorting, respectively.

By the experiments related to CPU, we can see that all the methods are influenced by page size. This is due to the fact that, with smaller pages, the tree is higher and the nodes MBRs enclose a smaller area, containing less elements. This allows a greater capacity of filtering interior levels of the trees, and have less comparisons.

It's also possible to see that *plane-sweeping* techniques contribute to reduce de CPU time spent by the BF and DF methods, comparing to the NL method. On the other hand, the use of sorting increased the CPU time, mainly in the BF method, which has to deal with larges IJI.

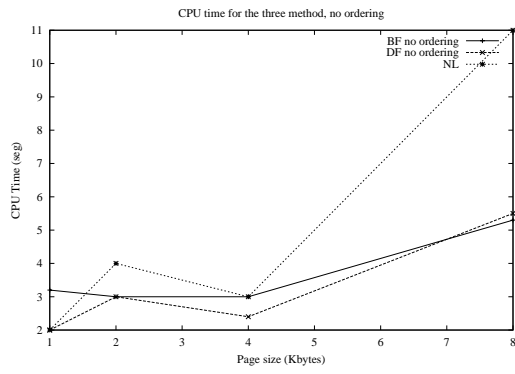
It is important to say that the BF method used much more memory than the other methods in order to store the IJIs, mainly using 1 kbyte and 4 kbytes page. This also justifies the peaks existing for the BF curves for 1 kbyte and 4 kbytes pages, the page sizes for which the IJIs had the largest sizes.

### 4.2 I/O operations

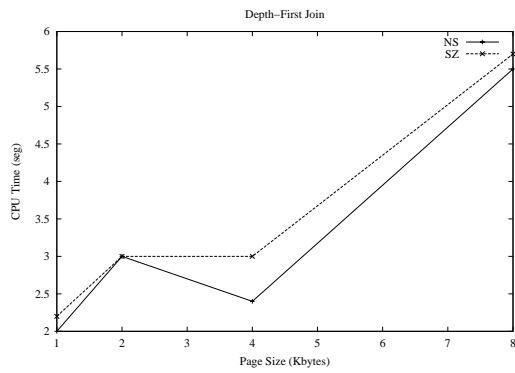
We present now the results for the I/O operations obtained by the NL, DF and BF methods. The NL method was the worst one when running *poles*  $\times$  *blocks*. But it's useful in order to compare the gains we may have with some policies.

#### 4.2.1 Results for DF method

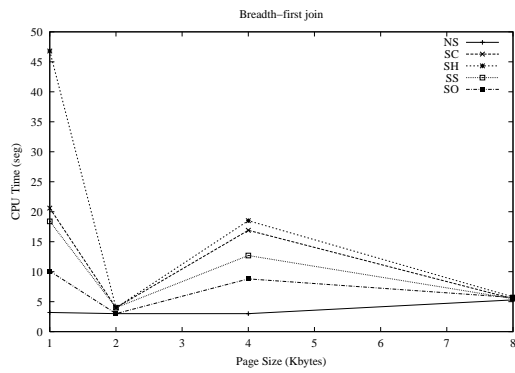
The Figure 5 shows the results for the DF method, comparing with the results of NL method. Although not shown here, the results for 1 kbyte and 2 kbytes pages reflect these



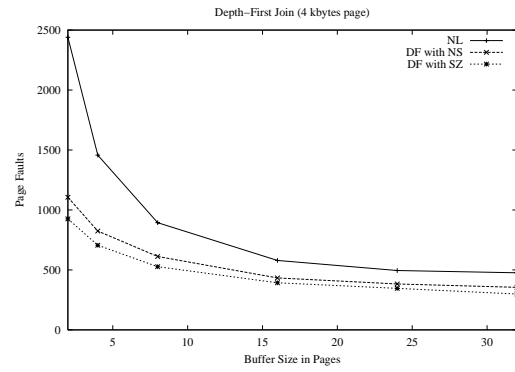
(a) CPU time not considering ordering time



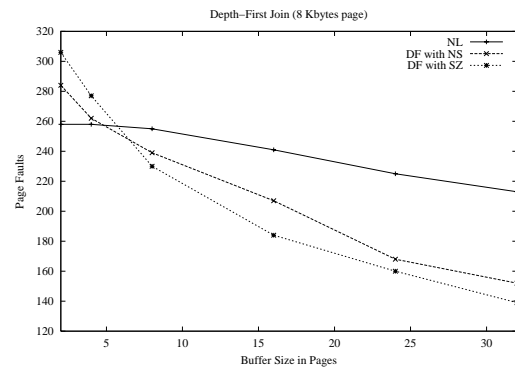
(b) CPU time for DF method



(c) CPU time for BF method



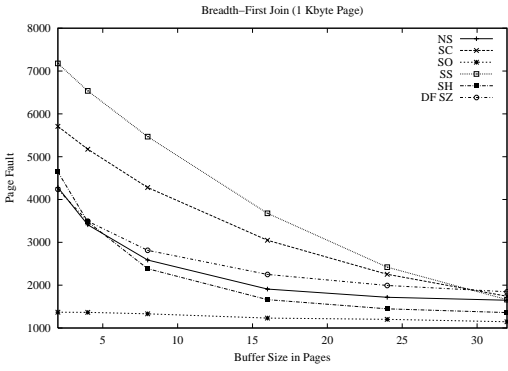
(a) 4 kbyte page



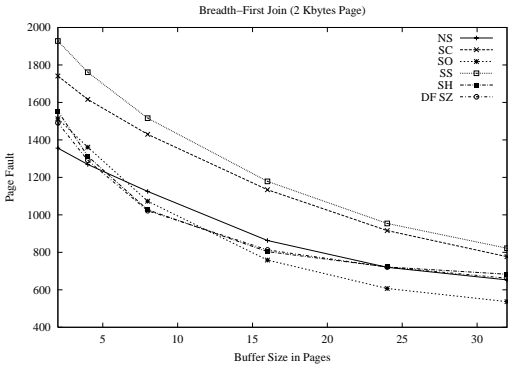
(b) 8 kbyte page

Figure 5: Results for the DF method, using indexes on 4 kbytes (a) e 8 kbytes (b).

Figure 4: CPU time spent by the different methods and a comparison among them.



(a) 1 kbyte page



(b) 2 kbyte page

Figure 6: Results BF method 1 kbytes (a) e 2 kbytes (b) pages.

results. It's possible to see that the improvements introduced resulted into better performance.

But, opposite to Brinkhoff *et al.* [12], we believe that it's possible the use of the  $z$  curve in order to sort the IJI. In their work, Brinkhoff *et al.* said that the gain in terms of I/O operations did not paid the extra CPU expenses when sorting by  $z$  curve. But today we have faster machines and capable of handling the extra workload of  $z$  curve sorting by, so that the gap between the expenses of NS and SZ would be very small or vanish.

#### 4.2.2 Results BF method

The BF method, more than the DF method, is highly influenced by the order the pairs are place in the IJI. The results of the BF tests for 1 kbyte and 2 kbytes can be seen in Figure 6, comparing to DF method with SZ.

The best ordering criteria for the BF method was SO,

mainly when we have indexes with different heights. In opposite to DF method, the BF method was able to read all the needed pages only once. This is due to this method nature: it's possible to know all the pages that are necessary to the join in advance, allowing a better ordering of the IJI. On the other hand, this method demanded much more memory than the DF method. And, according to the size of this IJI, it would be necessary to store parts of it in the disk, what causes more accesses to disk and could nullify the benefits of this method. Just to have an idea of how big the IJI can be, some experiments generated IJIs with 70.7% of the index size.

#### 4.2.3 Other results

In addition to the experiments shown, where were joined the *poles* and *blocks* set, other tests were run. These tests were *blocks*  $\times$  *blocks* (BB) and *city*  $\times$  *city* (CC). The results for BB tests are according to the ones already presented here. On the other hand, the CC for DF tests diverged from these ones.

On CC tests, the NL method was superior to the DF method, considering I/O results. The BF method was still the best concerning this criteria, but when the buffer pool was relatively small, NL was better.

These results can be caused by the difference of sizes and distribution of the *city* set. Some analysis were done and they show that there are some MBRs that cover great part of data space (Figure 3). These large MBRs results into large intermediate MBRs, which spread to higher level, increasing the overlapping among internal node. This is directly related to the data insertion routine in the index: if the insertion routine generates internal levels with high overlap, it will affect performance of all the queries. But, in order to know, without doubts, the real cause of this anomaly, other experiments should be performed, with data of several configurations, what has been the target of new tests.

### 5 Conclusions

By the results of these test we can conclude that:

- the increase of page size results the increase of CPU workload. So, the increase of a page size should be followed by improvement of the plane-sweeping algorithms;
- a good ordering of the IJI is very important to the good performance of the join. In our tests, the best were SZ for DF and SO for BF;
- a good performance of spatial joins also depends on a buffer pool with a good proportion of pages. In our tests, a buffer with good proportion was about 10% or more of the size of the indexes in pages;

- the use of the BF algorithm can, depending on the data density, turn into prohibitive IJI sizes;
- there are data sets where the use of the DF method results into worse performance, in number of I/O operations, than the NL method;
- insertion and manipulation routines have fundamental importance on all kinds of queries.

In short, spatial join methods could and should be improved, taking into account the factor analysed here. The gain the tuning of these factors could bring justify completely this work and new researches.

### Acknowledgements

We would like to thank CPqD for the data sets used in this work. Thanks also to CNPq for the financial support of this work, through Pronex/SAI project.

### References

- [1] B. C. Ooi. *Efficient query processing in geographic information systems*, volume 471 of *Lecture Notes in Computer Science*. Springer-Verlag Inc., New York, NY, USA, 1990.
- [2] F. S. Cox and G. C. Magalhães. Implementação e análise de métodos de acesso adados espaciais. In *VII Simpósio Brasileiro de Banco de Dados*, Porto Alegre, Maio 1992.
- [3] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R\*-tree: an efficient and robust access method for points and rectangles. In *Proceeding of the 1990 ACM SIGMOD International Conference on Management of Data*, Junho 1990.
- [4] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1984.
- [5] J. Patel and D. DeWitt. Partition based spatial-merge join. In *SIGMOD*, pages 259 – 270, Montreal, Canadá, Junho 1996.
- [6] L. Arge, O. Procopiuc, S. Ramaswamy, T. Suel, and J. Vitter. Scalable sweeping-based spatial join. In *Proc. of the 24th VLDB Conference*, New York, USA, 1998.
- [7] M. Lo and C. V. Ravishankar. Spatial hash-joins. In *Proceedings of the 1996 ACM SIGMOD, SIGMOD RECORD*, pages 247–258, Montreal, Canadá, Junho 1996. ACM Press.
- [8] G. Z. da Silva. *Avaliação de Junções em Bancos de Dados Espaciais*. PhD thesis, COPPE/UFRJ, Rio de Janeiro, Junho 1999.
- [9] M. Lo and C. V. Ravishankar. Spatial joins using seeded trees. In *Proceedings of the 1994 ACM SIGMOD*, volume 23 of *SIGMOD RECORD*, pages 209–220. ACM Press, Junho 1994.
- [10] N. Mamoulis and D. Papadias. Integration of spatial join algorithms for processing multiple inputs. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, Filadéfia – Pensilvânia, Junho 1999.
- [11] O. Günther. Efficient computation of spatial joins. In *International Conference on Data Engineering*, pages 50–60, Los Alamitos, Ca., USA, Abril 1993. IEEE Computer Society Press.
- [12] T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient processing of spatial joins using R-trees. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 22(2):237–246, junho 1993.
- [13] Y.-W. Huang, N. J., and E. A. Rundensteiner. Spatial joins using R-trees: Breadth-first traversal with global optimizations. In *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases*, pages 396–405, 1997.
- [14] A. P. Carneiro. Análise de desempenho de métodos de acesso espaciais baseada em um banco de dados real. Master's thesis, Universidade Estadual de Campinas - UNICAMP, 1998.
- [15] G. C. Magalhães. Telecommunications outside plant managements throughout Brazil. In *Conference XX Proceedings*, Nashville, 1997.
- [16] G. C. Magalhães. Projeto SAGRE. *Fator GIS*, Out./Nov./Dez. 1993.
- [17] G. C. Magalhães. The development of open systems for engineering applications. In *Proc. of XVII Intl. Conference on AM/FM*, Denver - Colorado, Março 1994.

# TerraLib: Technology in Support of GIS Innovation

GILBERTO CÂMARA<sup>1</sup>, RICARDO CARTAXO MODESTO DE SOUZA<sup>1</sup>

BIANCA MARIA PEDROSA<sup>1</sup>, LÚBIA VINHAS<sup>1</sup>

ANTÔNIO MIGUEL VIEIRA MONTEIRO<sup>1</sup>, JOÃO ARGEMIRO PAIVA<sup>1</sup>

MARCELO TILIO DE CARVALHO<sup>2</sup>, MARCELO GATTASS<sup>2</sup>

<sup>1</sup> Instituto Nacional de Pesquisas Espaciais - INPE

Av. dos Astronautas, 1758, São José dos Campos (SP), Brazil 12227-001

{gilberto, cartaxo, bianca, lubia, miguel, miro}@dpi.inpe.br

<sup>2</sup> Catholic University of Rio de Janeiro

Rua Marquês de São Vicente 225, Rio de Janeiro, Brazil 22.453-900

{gattass, tilio}@tecgraf.puc-rio.br

**Abstract.** This work describes the development of a new GIS library (called TerraLib), that is aimed at providing a rich and powerful environment for the development of GIScience research. The motivation for this proposal is the current lack of either public or commercial GIS libraries that cater for the diversity of GIS data and algorithms, especially when viewed upon the latest advances in geographical information science. TerraLib is open source software, allowing a collaborative environment and its use for the development of multiple GIS tools.

**Keywords.** GIS, Spatial Analysis, Software Libraries.

## 1. Introduction

The last 20 years have seen dramatic developments in GIS technology and geographical information science. GIS software is now ubiquitous, and there are systems in different types and sizes, varying from the desktop to the corporate user and different solutions for Internet data access and distribution. In the vast majority of cases, such developments have been industry-driven, with few exceptions (such as the IDRISI and SPRING systems). Fierce competition and growing user demand has resulted in a number of high-quality solutions, which are largely responsible for the vast increase in the GIS marketplace.

However, the vast majority of the industry solutions is aimed at supporting basic needs of capture, archival and visualisation of spatial data. Recent technological advances have concentrated in issues such as user-friendly interfaces, interoperability across data repositories and spatial extensions of database technology. These developments have largely ignored recent advances in GIScience, which include research areas such as geostatistics

(Goodvaerts, 1997), global and local spatial statistics (Getis and Ord, 1998), dynamic modelling and cellular automata (Couclelis, 1997; White and Engelen, 1997), heuristic search (Oppenshaw, 1998), environmental modelling (Burrough, 1998), point pattern analysis (Bailey and Gattrel, 1995), uncertainty assessment and modelling (Heuvelink, 1998; Felgueiras, 1999), spatial econometrics (Anselin, 1988) and neural networks for spatial data (Medeiros, 1999).

The authors posit that the geographical information community would benefit from the availability of a general, open source GIS library. This resource would make a positive impact by allowing researchers and solution developers access to wider range of tools than what is currently offered by the commercial companies. In a similar approach as the GNU and Linux efforts, such development does not happen by spontaneous growth: there has to be a core set of technologies in which further development can take place. Our proposal for the development of the **TerraLib** spatial library aims precisely at offering the GIS community a basis for further development.

The work is divided as follows. Section 2 indicates the general principles and of **TerraLib**. In Section 3, we present the main components of the library. In Section 4, we illustrate the programming environment of **TerraLib**, from a simple set of programs.

## 2. TerraLib Design Rationale

### 2.1 Generic Software Requirements

What sort of environment should a GIS library cater for? It should provide, first of all, support for the basic components of spatial data sets: *data translators, map representation, geometrical data structures and algorithms*. It also needs to support the establishment of *data models* for GIS data, without imposing strong constraints on their use. It also needs to provide *user interface* tools, which may be used for simple applications.

The need for perform efficient *data translation* is the single most important design consideration in TerraLib. Most users already have their data in an existing system and will want to perform analysis and exploration using techniques not available in their original system. There should be support for inclusion of tabular data, which has been assigned a geographical reference (as a census tract number or a zipcode).

*Map representation* and *cartographic projection* tools form a basic core of a GIS library. The library should support a basic core of cartographic projections and a set of techniques for reprojection and integration of spatial data. It should be very easy to add support for new projections and datum.

*Algorithms* form a basic core of most successful research efforts. In many GIS libraries, the misuse of object-oriented principles has resulted in classes that contain both the underlying data structure and the corresponding set of algorithms. In this case, the algorithms would be unnecessarily linked to a particular type of data structure. Therefore, TerraLib algorithms are to designed as independent entities, which do not belong to a particular class.

The traditional *geometrical data structures* used for geographical data include vector data structures (points, lines, polygons, triangular meshes), raster structures (matrices) and relational structures (tables). Such structures

also need underlying support for indexing structures such as R-trees.

In the case of *data models*, one of the important advances in GIScience in the 90's has been the widespread acceptance of a general conceptual data model for geographical data. In this model, geographical reality is represented as either fully definable entities (*features*) or smooth, continuous spatial variation (*fields*). Although this simple dichotomy has been subject to objective criticism (Couclelis, 1992; Burrough and Frank, 1996), it has proven a useful frame of reference. It has adopted, with some variations, in the design of the latest generation of GIS technologies, such as SPRING (Cámara et al, 1996), Arc/Info-8 and OpenGIS.

We have designed TerraLib to allow different alternatives of *data models* to be implemented from the same software basis. Therefore, we emphasized a very loose coupling between the data model classes (on one side) and the algorithms and data structures (on the other side). Although this de-coupling may lead to some duplication, it should be useful to be able to derive different applications based on TerraLib that might use the data model of different systems. For example, one developer may want to integrate TerraLib to existing systems such as GeoMedia or ARC/INFO-8. Another may develop OpenGIS-compliant applications.

### 2.2 Support for Innovative Research

The preceding discussion was centred in the support of conventional GIS applications. However, in order to be useful as a support for innovative research, **TerraLib** needs, from its conceptual base, to consider the needs of emerging GIS research areas, such as:

- Uncertainty modelling, where each field is coupled with information about spatial imprecision, which is propagated in map algebra operations (Heuvelink, 1998).
- Cellular automata, whose application to GIS requires that each cell has its own inherent set of attributes (as distinct from a single state) which represent its relevant physical, environmental, social, economic or institutional characteristics. (Couclelis, 1997; White and Engelen, 1997).
- Dynamic modelling (Burrough, 1998) requires the support for timers and interactive procedures.

- Applications such as Spatial Statistics and Spatial Interaction Models require the use of spatial proximity matrices for supporting spatial relations among elements.

### 2.3 Development Strategy

Since **TerraLib** was designed to support an open, collaborative development environment, some basic principles are in order:

1. The interface for each class should be kept as *minimal* as possible. The introduction of new algorithms and tools should not affect already-existing code, include keeping *include* files intact.
2. The implementation of the geometrical data structures is kept completely separate from the programming interface, using the “pimpl” idiom (Sutter, 2000).
3. There should be a maximal degree of *ortogonality* between the components of the library, and they are designed to be used *independently*.

We are not committed to a single programming style, but use the ideas of multi-paradigm programming, which advocates the combination of different techniques such as object-oriented, algorithmic and generic programming [Coplien, 1999]. This technique is particularly suited to the C++ language [Stroustrup, 1997].

Reuse considerations have also led the authors to choose C++ as the development language. We hope to adapt algorithms that have been developed for **SPRING** (Câmara et al, 1996).

## 3 Software Structure

### 3.1 Main Components

**TerraLib** is developed as a multi-tier library, and its first version includes:

- At the higher level, it provides abstract data classes to manage and represent geographical information, based on a field/object paradigm, viz.: *Feature*, *Network*, *Surface*, *CellArray*.
- At the intermediate level, it provides support for *geometrical data structures*, *data formats*, *map representation* and *algorithms*.
- At its lowest level, it provides spatial indexing structures for efficient handling of large data sets.

### 3.2 Geometrical and Attribute Structures

The *data structures* include:

- **TeCoord2D**<sup>1</sup>: a 2D coordinate.
- **TeLine**: a vector of 2D coordinates, that can be associated to a height.
- **TeLineSet**: a set of lines.
- **TePolygon**: composed of Lines and of other Polygons (its children).
- **TePolygonSet**: a set of polygons.
- **TePointSet**: a set of 2D samples.
- **TeTIN**: a triangular mesh.
- **TeGrid**: a raster data structure (used for images and grids).
- **TeTable**: an attribute table used for linking to relational DBMSs

Each instance of a data structure has a unique identifier, to enable its linking and storage in a DBMS. We illustrate some of these geometrical structures and its relationships in Figure 1.

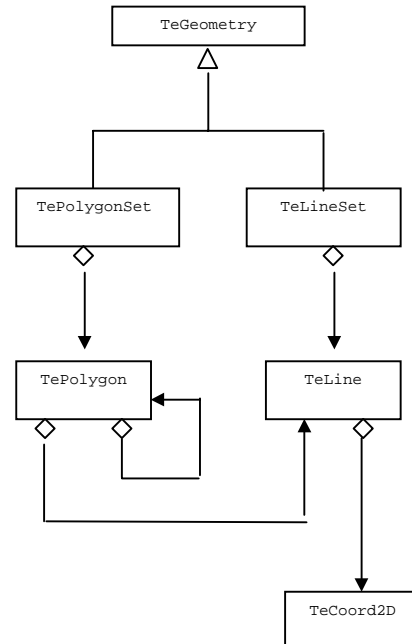


Figure 1 – Polygon and Line vector representation

<sup>1</sup> Throughout the text, TerraLib classes and functions are indicated with a Te- prefix.



As can be seen in the Figure 1, the TePolygonSet class is a composite of TePolygon. A TePolygon is a composition of TeLine and/or other TePolygons (its children). A TeLineSet is a composition of TeLines that are a composition of TeCoord2D.

A basic pattern for use in GIS geometrical structures is the Composite pattern (Gamma et al, 1996). As the Composite pattern can be used in another components of a GIS we propose a parameterised implementation of it (shown partially below):

```

template <class T>
class TeGeomComposite
{
public:

typedef vector<T> Components;
typedef Iterator<T> TeGeomIterator;

// -- Constructors
TeGeomComposite(){}

// -- Destructor
virtual ~TeGeomComposite();

// -- Methods
void Add ( const T& component );
int Size();
T& Next() const ;
T* First();
T* Last();

protected:
Components components_;
};

```

When considering the implementation of the geometry classes, we should allow for different alternatives to be tried and tested for efficiency and convenience. For example, a TePolygonSet can be represented as a graphical file or might be integrated into a relational database, with one polygon per tuple, in a similar fashion as the Simple Feature definition of the OpenGIS consortium (Open GIS, 1999).

To allow for different implementations of the Polygon class to be defined, we have used the so-called “pimpl” idiom (Sutter, 2000). This programming idiom proposes a separation between a class and its implementation, by using an opaque pointer to hide the implementation details.

In the case of Polygons, a special case of the Composite pattern arises: a Polygon is composed of Lines and of other Polygons (its children), as it can be seen below.

```

class TePolygon
{
public:
    // public members ...
private:
    struct TePolygonImpl *pimpl;
};
// sample implementation
// there could be alternatives
struct TePolygonImpl
{
class TeLine2DSet: public
    TeGeomComposite<TeLine2D> {}
class TeChildSet: public
    TeGeomComposite<TePolygon> {}

    TeLine2DSet lines_;
    TeChildSet children_;
};

```

Therefore, the implementation of the geometrical data structures has required a combination of Design Patterns and Generic Programming paradigms.

### 3.3 Algorithms and Iterators

For Terralib algorithms, we have adopted the principles of *generic programming*: “decide which algorithms you want; parametrize them so they work for a variety of suitable types and data structures” (Stroustrup, 1997). Following the example of the STL library, which is now part of the C++ standard, we propose the use of the *iterator* concept, as a basis for . Iterators are a generalisation of the idea of pointers, and are used in the STL to separate the containers from the algorithm [Austern, 1998].

The algorithms are defined in terms of different types of iterators of spatial data types, such as: TePolygonIterators, TeLineIterators, TePointIterators, and TeTableIterators.

In order to illustrate the concept, we present the case of a line simplification algorithm. Such an algorithm can be applied in the case of a set of lines, as well as in the case of a set of polygons. We would like to design such a function as independent entity, which does not belong to a particular class, as shown below.

```

void TeSimplifyLines (
    TeLineIterator begin,
    TeLineIterator end ) ;
int
main()
{
    TeLineSet ls;
    TeImportE00(ls, "lines.e00");
    TeSimplifyLines (ls.FirstLine(),
                    ls.LastLine());

    TePolygonSet ps;
    TeImportE00(ps, "polygons.e00");
    TeSimplifyLines (ps.FirstLine(),
                    ps.LastLine());
//....
return 0;
}

```

The parameters for the `SimplifyLines` function are `TeLineIterators`. The `SimplifyLines` algorithm doesn't need to assume anything beyond minimal functionality guaranteed by the `TeLineIterator`, which include functions to have access to a `TeLineSet` by traversing it in some order.

Note that both `LineSet` and `PolygonSet` classes need to provide method that return `TeLineIterators`. The geometric structures are thus responsible for providing the `Iterators` that are meaningful for the use of the algorithms.

### 3.4 User Interfaces and Visualisation

Strictly speaking, user interfaces and visualisation are not an essential part of `TerraLib`. Our emphasis is an efficient and flexible set of data structures and algorithms, allowing different applications to produce their own interfaces or the linking of `TerraLib` programs to existing environments.

However, in many cases, users may want a simple visualisation and user interface environment that allows for rapid prototyping of ideas and new concepts. For those purposes, we provide a set of GUI and visualisation classes, which are based on the **Qt** public-domain software library (Troll Tech, 2000).

The `TeApplication` class provides a simple interface for visualisation of GIS data. It provides a method `Run()` for creating an event loop, much similar to the operation of the `Motif` and `MFC` toolkits. The method `Show()` instructs the user interface to display a data, which may be a simple geometrical structure or a more complex layer.

## 3.5 Support for Innovative Research

The applications described in Section 2.2 have motivated some design decisions:

- In `TerraLib`, each field of `Surface` type is associated with an `Uncertainty` information. This field is part of the representation of the surface and access to it is available for error propagation functions.
- A new type of field, a `CellArray` type, is introduced to handle multiple attributes for a single cell, and make it easier to define transitional rules.
- Support for a `Graph` geometrical data structure is included, to allow for both spatial interaction algorithms and different alternatives for proximity measures (O'Sullivan, 1999).

## 4. Programming in TerraLib

### 4.1 Hello, GIS World !

We consider that the best description of GIS library is achieved by showing how it should be used in practice. Thus, we present a set of programs, starting from simple examples, which illustrate the principles and practice of programming in **TerraLib**. This description assumes a familiarity with the basics of GIS data structures and algorithms, as well as with the general programming in C++. Let us start by considering a problem: what is the simplest GIS program that can be written?

```

#include <teapplication.h>
#include <tegeometry.h>
#include <tedataconversion.h>
int main ()
{
    TeApplication app;
    TePolygonSet ps;
    TeImportShape (ps, "BR.shp" );
    app.Show (ps);
    app.Run();
}

```

This program reads a file containing a data set, in "shapefiles" format, and displays it. The first line of this program creates an instance of the `TeApplication` class, which is responsible for providing a simple interface for visualisation of GIS data. Since `TerraLib` is designed to be standalone, the use of this class is optional. The visualisation procedures

associated to TerraLib are described in the “User Interface and Visualisation” section.

The next line indicates a data conversion procedure (in this case, an ARC/View *shapefile* containing the co-ordinates of the counties of Brazil, or *municípios* in Portuguese). TerraLib provides efficient data translation tools for formats such as SHP, MIF and E00. Note that we have defined `TeImportShape` as a function rather than a method for a `TeGeometry` class in keeping with *principle 1* (minimal interfaces).

The next line simply indicates the existence of a method `Show()`, associated to the `TeApplication` class, which requests to the user interface the display of data. The last line indicates that the “main loop” of the `TeApplication` class is called. This command creates a window interface.

#### 4.2 Simple Geometrical Algorithms

The next step is to apply some simple geometrical algorithms. For example, let us suppose we want to read a data set containing the counties of Brazil as a set of polygons, and we would like to generate a Voronoi diagram from the centroids of these polygons. This is done as follows:

```
#include <tegeometry.h>
#include <tedataconvert.h>
#include <tealgorithm.h>
int main ()
{
    TePolygonSet ps;
    TeImportShape( ps, "Br.shp");

    TePointSet pt;

    TeGenerateCentroids
    ( ps.FirstPoint(),
      ps.LastPoint(),
      pt.FirstPoint() );

    TePolygonSet voron;
    TeVoronoi (
    pt.FirstPoint(),
    pt.LastPoint,
    voron.FirstPoint() );

    TeExportGBR(voron, "Vor.gbr");
}
```

This program creates a `TePointSet` from an existing `TePolygonSet` (read from a SHP file), and generates a Voronoi diagram with the `TeVoronoi` function. The result is

exported to a file in the GEOBR data format. Note the use of iterators as interfaces to the algorithms.

#### 4.3 Data Management

In the preceding examples, there is no abstract modelling of this data set, no control organisation is in place, and there are no attributes associated to the geometrical structures. If we want data management and to associate attributes to the geometrical structures, we need to introduce the concept of a geographical database (or `Geodatabase` for short). The `Geodatabase` class supports archival of both geometrical and descriptive parts of GIS data set. Additionally, it uses existing relational data base technological solutions (such as MySQL and ODBC) as basis for management of tables.

```
#include <teapplication.h>
#include <tegeodatabase.h>
#include <telayer.h>
#include <tegeometry.h>
int main ()
{
    TeApplication app;
    TeGeoDataBase db;
    db.Open ("World");
    TePolygonSet ps;
    TeImport (ps, SHP, "BR.shp" );

    TeTable t;
    db.ImportTable(t, "CA", "Br.dbf");

    TeLayer br;
    db.NewFeature(br, "Br", ps);
    db.Associate (br, t, "IBGE_ID");

    app.Show ( br );
    app.Run();
}
```

The above program requires the instantiation of a `GeoDatabase`, which is responsible for data management and handling attribute information. In this program, a higher-level structure (a `Layer` consisting of `Features`) is created from lower-level structures (a `PolygonSet` and a `Table`), in four steps:

- a geometry is created from existing data;
- a table is created from an external source;
- a new layer is instantiated, from an existing geometry;
- the attributes and the geometry are associated, by means of one attribute which indicates a spatial index.

This procedure allows for the association of many attribute sets to the same geometry (and vice-versa).

These examples show that even the simplest GIS application needs some procedures for data management and abstract modelling. Otherwise, the library user is left with a substantial burden to construct higher-level structures. They also show the application of our design rationale in some simple, but significant examples.

## 5. Conclusion

This paper outlines the rationale for the initial version of TerraLib, an open source software library aimed at offering the GIScience community a basis for shared development. This proposal is based on more than 15 years experience in GIS software development by the groups of the of Brazil's National Institute for Space Research (INPE) and the Catholic University of Rio de Janeiro. By making a version of the proposal available for public discussion, we hope to attract partners for long-term co-operative partnerships.

## Acknowledgements

This work has been partially financed by a joint NSF/CNPq project on "Interoperability on GIS" (process CNPq 480322/99).

## References

- ANSELIN, L. 1988. *Spatial Econometrics, Methods And Models*. Dordrecht: Kluwer Academic.
- ANSELIN, L. 1998. Interactive techniques and exploratory spatial data analysis. In: P. Longley, M. Goodchild, D. Maguire and D. Rhind (eds.), *Geographical Information Systems: principles, techniques, management and applications*, pp. 251-264. New York: Wiley.
- AUSTERN, M.H., 1999. *Generic Programming and the STL: Using and Extending the C++ Standard Template Library*. Reading, Addison-Wesley, 1999.
- BAILEY, T.; GATTRELL, A., 1995. *Spatial Data Analysis by Example*. London, Longman.
- BURROUGH, P.; FRANK, A., 1996. (eds) *Geographic Objects with Indeterminate Boundaries*. London, Taylor and Francis.
- BURROUGH, P., 1998. "Dynamic Modelling and GIS". In: Longley, P., Brooks, S., McDonnell, R., Macmillan, B. (eds), *Geocomputation: A Primer*. New York, John Wiley & Sons.
- CÂMARA, G.; SOUZA, R.C.M.; FREITAS, U.M.; GARRIDO, J.C.P., 1996 "SPRING: Integrating Remote Sensing and GIS with Object-Oriented Data Modelling". *Computers and Graphics*, vol.15, n.6, pp.13-22.
- CÂMARA, G.; SOUZA, R.C.M.; MONTEIRO, M.V.; PAIVA, J.; GARRIDO, J., 1999. "Handling Complexity in GIS Interface Design". In: *Proceedings of the 1 Brazilian Workshop on GeoInformatics*, Campinas, São Paulo. [www.dpi.inpe.br/geoinfo99](http://www.dpi.inpe.br/geoinfo99).
- COPLIEN, J. *Multi-Paradigm Design for C++*. Reading, Addison-Wesley, 1999.
- COUCLELIS, H., 1992. "People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS". In: Frank, A.; Campari, I. and Fomentini, U. (eds) *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pp. 65-77. Berlin, Springer.
- COUCLELIS, H., 1997, "From cellular automata to urban models: new principles for model development and implementation", *Environment and Planning B: Planning & Design*, 24, 165-174.
- DEUTSCH, C.; JOURNAL, A. 1998. *GSLIB: Geostatistical Software Library and User's Guide*. New York, Oxford University Press.
- FELGUEIRAS, C. 1999. "Modelagem Ambiental com Tratamento de Incertezas em Sistemas de Informação Geográfica: O Paradigma Geoestatístico por Indicação". PhD Thesis in Computer Science, INPE (in Portuguese).
- GETIS, A., ORD J. K., 1996. "Local spatial statistics: an overview". In: *Spatial Analysis: Modelling in a GIS Environment*. (LONGLEY, P.; BATTY, M., eds), pp. 261-277. New York, John Wiley.
- GOOVAERTS, P. 1997. *Geostatistics for Natural Resources Evaluation*. New York, Oxford University Press.
- HEUVELINK, G. 1998. *Error Propagation in Environmental Modelling with GIS*. London, Taylor and Francis.
- LEONDES, C. (ed), 1997. *Image Processing and Pattern Recognition (Neural Network Systems Techniques and Applications Series, Vol 5)*. New York, Academic Press.

LONGLEY, P., 1998. (ed) *Geocomputation: A Primer*. New York, John Wiley and Sons, 1998.

MEDEIROS, J.S., 1999. *Geographical Databases and Artificial Neural Networks: Technologies in Support of Land Management*. PhD Thesis in Geography, University of São Paulo (in Portuguese).

OPENSHAW, S., 1998. "Building automated Geographical Analysis and Exploration Machines". In: *Geocomputation: A primer* (Longley, P. A., Brooks, S. M. and McDonnell, B. (eds)), p. 95-115. Chichester, Macmillan Wiley.

O'SULLIVAN, D. 1999. Exploring the structure of space: towards geo-computational theory. In: *Proc. IV International Conference on GeoComputation*. Mary Washington College, USA.

<[www.geovista.psu.edu/geocomp/geocomp99](http://www.geovista.psu.edu/geocomp/geocomp99)>.

STROUSTRUP, B. 1997. *The C++ Programming Language*. Reading, Addison-Wesley.

SUTTER, H., 2000. *Exceptional C++*. Reading, Addison-Wesley.

TROLLTECH, 2000. *The Qt Interface Library*. [www.trolltech.com](http://www.trolltech.com).

WHITE, R.; ENGELEN, G., 1997 'Cellular Automata as the Basis of Integrated Dynamic Regional Modelling' *Environment and Planning B*, Vol.24, pp.235-246.