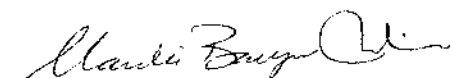


Incorporação de Facilidades Espaço-Temporais em Bancos de Dados Orientados a Objetos

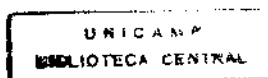
Este exemplar corresponde à redação final da tese devidamente corrigida e defendida pelo Sr. Marcio de Araújo Botelho e aprovada pela Comissão Julgadora.

Campinas, 18 de dezembro de 1995.



Profa. Dra. Claudia M. Bauzer Medeiros
(Orientadora)

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.



UNIDADE	79C
N.º CHAMADA:	
V.	Ex.
COMBO	26673
PROC.	66756
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,50
DATA	02/06
Q.º CPU	

CM-00082889-9

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Botelho, Márcio de Araújo


B657i Incorporação de facilidades espaço-temporais em bancos de dados orientados a objetos/Márcio de Araújo Botelho -- Campinas, [S.P. :s.n.], 1995.

Orientador : Claudia Bauzer Medeiros


Dissertação [mestrado] - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Ciência da Computação.

1. Bancos de dados orientados a objetos. 2. *SIG. 3.*Espaço temporal. 4.*SGBDOO. I. Medeiros, Cláudia Bauzer II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Ciência da Computação. III. Título.

Tese de Mestrado defendida e aprovada em 18 de dezembro de 199
pela Banca Examinadora composta pelos Profs. Drs.



Prof (a). Dr (a). Claudia Benzer Medeiros



Prof (a). Dr (a). Luiz Eduardo Benzer



Prof (a). Dr (a). Juander Zullo Jr

**Incorporação de Facilidades Espaço-Temporais
em Bancos de Dados Orientados a Objetos**

Marcio de Araújo Botelho

Incorporação de Facilidades Espaço-Temporais em Bancos de Dados Orientados a Objetos¹

Marcio de Araújo Botelho²

Orientadora:

Prof.Dra. Claudia Bauzer Medeiros

Departamento de Ciência da Computação

IMECC - UNICAMP

¹ Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação da UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

² O autor é Bacharel em Ciência da Computação pela Universidade Federal do Ceará.

**Think Globally,
Act Locally**
- environmentalist René Dubos.

Aos meus pais
José Waldez Botelho e Carmen de Araújo Botelho,
pela dedicação e empenho com que sempre lutaram pela nossa formação.

Agradecimentos

Ao longo desta jornada tivemos a sorte de poder contar com o apoio e o incentivo de pessoas muito especiais que não pouparam esforços em nos ajudar. Gostaria de agradecer a todos os colegas de mestrado, professores, funcionários, parentes e amigos que de alguma forma ajudaram para que este trabalho fosse realizado. Em especial, gostaria de expressar a minha profunda gratidão a aqueles com quem tive o prazer de conviver durante essa importante etapa de minha vida, e que foram fundamentais para a realização desta dissertação.

À minha orientadora Profa. Cláudia pela sua confiança, forte incentivo, preciosos ensinamentos transmitidos no decorrer do curso, e grande empenho para superar os obstáculos e vencer as incertezas.

Aos professores Ricardo Dahab, Célia Picinin e Pedro Sérgio que sempre estiveram presentes para incentivar e ajudar em momentos difíceis.

Aos funcionários do DCC, em especial a Roseli, a Solange e ao Luíz (em memória), pela solicitude e presteza com que sempre nos auxiliaram.

À minha família, sempre pronta para ajudar nos momentos mais difíceis.

Aos padrinhos Ronan e Zezé, presenças constantes nos momentos mais críticos da nossa vida, pelo apoio na superação de mais uma etapa.

Ao Celso e Cristina pelo grande apoio, incentivo e confiança proporcionados.

Aos amigos da república George Lima, Eduardo Camponogara, Fábio Póvoa e Ricardo Barros pelo espírito de companheirismo e agradáveis momentos compartilhados.

Aos amigos do grupo de Banco de Dados pelas sugestões e valiosa cooperação durante os nossos encontros. Em especial, Mariano, Juliano, Fátima, Cristina, Ricardo pelas inúmeras revisões e colaborações.

Aos amigos Horácio, Sandra, Ana Maria, Maria, Dinalva, Sueli, Raul César, Émerson, Aredis, Edmar, Edson e Maurício pelo apoio e incentivo.

À Cármen Sílvia Gonzales pelo apoio incondicional, enorme incentivo, e pelos momentos felizes vividos durante essa caminhada.

Aos amigos Licínio Corrêa, Wellington Borges e Ana Maria Freitas que sempre deram apoio apesar da distância.

Ao CNPQ, pelo suporte financeiro durante todo o trabalho de Mestrado.

Esta dissertação foi desenvolvida no contexto do projeto PROTEM-CNPQ-GEOTEC e do projeto GEOTOOLS ITDC 116 da Comunidade Européia.

Resumo

Esta dissertação descreve uma abordagem para estender um sistema gerenciador de banco de dados orientado a objetos de forma a permitir suporte a dados espaço-temporais.

As principais contribuições são:

- Descrição de um modelo de dados espaço-temporal orientado a objetos que permite representar a evolução de dados espaço-temporais, comuns em sistemas de informação geográfica.
- Definição de estruturas de dados para dar suporte ao modelo em um banco de dados orientado a objetos, armazenando dados espaciais na representação vetorial. Estas estruturas viabilizam o armazenamento da evolução temporal dos objetos, que encapsulam métodos de acesso aos seus estados temporais.
- Especificação de uma taxonomia de consultas espaço-temporais em SIG.

Esta proposta estende assim outros modelos para SIG, permitindo que essas facilidades sejam incluídas em sistemas futuros.

Abstract

This dissertation presents a framework to incorporate support for spatial-temporal data in object oriented database management systems. The main contributions are:

- Description of a spatial-temporal object oriented data model, allowing the representation of spatial-temporal data evolution, common in geographic information systems.
- Definition of data structures in a object oriented database to support the model, storing spatial data in the vector format. This structures make possible to store the temporal evolution of the objects, which encapsulate access methods to their temporal states.
- Specification of a taxonomy of spatial-temporal queries in geographic information systems.

This proposal extends other GIS models, bringing the possibility to incorporate new facilities in future systems.

Conteúdo

Agradecimentos	v
Resumo	vii
Abstract	viii
1 Introdução	1
1.1 Motivação e Objetivos da Dissertação	1
1.2 Terminologia	3
1.3 Alguns Problemas Existentes em SIG	4
1.4 Processos Espaço-Temporais	5
1.5 Consultas e Aplicações de SIG Temporais	6
1.6 Arquiteturas Existentes em SIG	9
1.7 Padrões de Modelagem	10
1.8 Organização da Dissertação	11
2 Conceitos de Bancos de Dados Temporais	13
2.1 Terminologia	14
2.1.2 Atributo Invariante no Tempo	14
2.1.3 Atributo Variante no Tempo	14
2.1.4 Chronon	15
2.1.5 Densidade no Tempo	15
2.1.6 Ordem no Tempo	15
2.1.7 Dimensões Temporais	16
2.1.8 Span e Tempo de Vida	18
2.1.9 Marca-de-Tempo	18
2.1.10 Aninhamento Temporal	19

2.1.11 Granularidade Temporal	21
2.1.12 Tempo-válido Relativo/Absoluto.....	21
2.1.13 Indeterminância e Modalidade Temporal	22
2.1.14 Expressão Temporal.....	23
2.1.15 Interpolação Temporal	23
2.2 Linguagens de Consulta Temporal	24
2.3 O Modelo TOODM (Temporal OO Data Model).....	25
2.3.1 Consultas que Retornam Intervalos de Tempo	26
2.3.2 Consultas que Retornam Objetos Temporais.....	27
2.4 A Camada de Gerenciamento Temporal.....	28
2.5 Resumo	31
 3 Modelagem Temporal de Dados Geográficos	 32
3.1 Introdução	32
3.2 Evolução do Armazenamento de Dados em SIG	33
3.3 O Modelo de Dados Proposto.....	35
3.3.1 Definição do Modelo.....	35
3.3.2 Nível Conceitual	36
3.3.3 Nível de Representação	38
3.4 Processos Geográficos Temporais.....	40
3.5 Fronteira Temporal	42
3.6 Modelagem de processos Geográficos Temporais	43
3.7 Dimensões Temporais.....	45
3.8 Evolução Temporal dos Geo-objetos.....	47
3.8.1 Temporalização da Lista de Geo-objetos Componentes.....	48
3.8.2 Temporalização de Atributos não-espaciais	49
3.8.3 Temporalização da Localização	50
3.9 Exemplos de Evolução da Localização	51
3.9.1 A Evolução de um Geo-objeto Linear	51
3.9.2 A Evolução de um Geo-objeto Poligonal.....	52
3.10 Resumo.....	53

4 Consultas em SIG	54
4.1 Introdução	54
4.2 Algumas Classificações de Consultas Geográficas.....	54
4.3 Classificação de Consultas Proposta.....	57
4.3.1 Definições	57
4.3.2 Dominância Dimensional.....	58
4.4 Componentes Básicos de uma Consulta Espacial Atemporal.....	59
4.4.1 Seleção Espacial	59
4.4.2 Projeção Espacial.....	62
4.4.3 Agrupamento Espacial	64
4.5 Componentes Básicos na Dimensão Temporal	65
4.5.1 Seleção Temporal.....	66
4.5.2 Projeção Temporal.....	67
4.5.3 Agrupamento Temporal.....	68
4.6 Componentes Básicos na Dimensão Espaço-Temporal	69
4.6.1 Seleção Espaço-Temporal	69
4.6.2 Projeção Espaço-Temporal.....	70
4.6.3 Agrupamento Espaço-Temporal.....	72
4.7 Descrição Global das Consultas	74
4.8 Operação entre Mapas	75
4.8.1 União de Mapas	75
4.8.2 Sobreposição de Mapas.....	75
4.8.3 Decomposição Espacial.....	77
4.8.4 Outros Tipos de Operação.....	77
4.9 Resumo	78
5 Estrutura dos Dados Espaço-Temporais	79
5.1 Introdução	79
5.2 Estrutura de Dados Espaciais Atemporais.....	80
5.3 A Formação dos Polígonos	82

5.4 Exemplos de Consultas.....	84
5.4.1 Consultas Espaciais.....	85
5.4.2 Consultas Temporais.....	86
5.4.3 Consultas Espaço-Temporais.....	88
5.5 Estrutura Interna das Classes Geométricas	88
5.6 Implementação das Consultas	91
5.7 Classes Localização e Geometria.....	95
5.8 Classes Geográficas.....	96
5.9 Estratégia para Armazenar e Acessar Dados Espaço-Temporais	98
5.10 Resumo.....	99
 6 Conclusões	 101
 Anexo	 104
1 Comprimento	104
2 Área	105
3 Interseção	107
4 Distância	108
5 Inclusão	109
6 Continência.....	110
 Referências	 111

Lista de Tabelas

2.1 SGBD x Dimensões Temporais.....	17
2.2 Modelos de Aninhamento Temporal.....	20
4.1 Consultas Geográficas.....	55

Lista de Figuras

1.1 Consultas Espaço-Temporais.....	8
3.1 Modelo no Nível Conceitual	37
3.2 Modelo no Nível de Representação	39
3.3 Evolução dos Geo-objetos no Tempo.....	40
3.4 Fronteira Temporal de um Processo	42
3.5 Relacionamentos entre Processos.....	44
3.6 Evolução de uma Localização bitemporal.....	45
3.7 Evolução de um Geo-objeto Linear.....	51
3.8 Evolução de um Geo-objeto Poligonal	52
4.1 Um Exemplo de Consulta Geográfica.....	56
4.2 Windowing	60
4.3 Clipping.....	61
4.4 Projeção Espacial.....	63
4.5 Agrupamento Espacial.....	65
4.6 Taxonomia de Consultas	74
4.7 União de Mapas	75
4.8 Sobreposição de Mapas	76
4.9 Decomposição Espacial.....	77
5.1 Representação Espacial de um Geo-objeto	81
5.2 Particionamento de um Polígono.....	82
5.3 Junção de um Polígono	83
5.4 Chamada ao Método Loc	92
5.5 Execução de uma Consulta.....	93
5.6 Intervalos de Interseção	93
5.7 Estratégias de Armazenamento	99

Capítulo

1 Introdução

1.1 Motivação e Objetivos da Dissertação

Os sistemas de informações geográficas (SIG) surgiram como uma forma de facilitar o trabalho de análise geográfica automatizando o processo de coleta e processamento de dados geográficos. Segundo [Aro89], os SIG são sistemas computacionais que armazenam e manipulam informações geográficas. [FL95] ressalta que esses sistemas tiveram um importante papel como tecnologia de integração entre várias disciplinas distintas: Geografia, Cartografia, Fotogrametria, Sensoriamento Remoto, Geodésia, Engenharia, Estatística, Ciência da Computação, Pesquisa Operacional, Inteligência Artificial, Demografia, e muitos outros ramos das ciências sociais, ciências naturais, ciências econômicas e ciências políticas. Os autores fazem três importantes observações sobre a sua definição:

1. Os SIG estão relacionados com outras aplicações de bancos de dados, mas com uma importante diferença. Todas as informações em um SIG estão ligadas a uma referência espacial. Outras aplicações manipulam dados que podem conter informações localizacionais (como endereços de ruas, ou códigos postais), mas os SIG usam geo-referências como o meio principal de armazenamento e acesso aos dados.
2. Os SIG integram tecnologia. Enquanto outras aplicações são usadas somente para analisar fotografias aéreas e imagens de satélite, para criar modelos estatísticos, ou para desenhar mapas, essas funções foram todas integradas pelos SIG.
3. Os SIG devem ser vistos como uma tecnologia, ao invés apenas de um software ou um hardware. Ou seja, um SIG é a forma como os dados são coletados, processados, analisados e apresentados, auxiliando o processo de tomada de decisão.

Os SIG têm deixado de ser soluções computacionais criadas para resolver problemas específicos e circunstanciais, e cada vez mais tornam-se uma tecnologia sólida e permanente nas instituições. Entretanto os SIG atuais são atemporais, ou seja, não provêm nenhum mecanismo para manipulação de versões antigas dos dados georeferenciados. Em contrapartida, os SIG temporais liberariam o usuário da difícil tarefa de gerenciar manualmente a crescente quantidade de dados históricos desses sistemas.

Para facilitar a criação e o desenvolvimento de SIG temporais, esta dissertação apresenta um estudo aprofundado das necessidades existentes nessa área e descreve um modelo geográfico para representar as informações espaço-temporais desses sistemas. Baseada nesse modelo, uma taxonomia foi elaborada enquadrando as principais consultas espaço-temporais. Além disso, identificamos uma possível estrutura de dados para armazenar essas informações analisando também vantagens e desvantagens de algumas pequenas variações alternativas que permitam otimizar o desempenho de consultas de acordo com o tipo de aplicação.

O modelo geográfico espaço-temporal descrito nessa dissertação tem algumas características que são descritas a seguir:

- É baseado nos conceitos de orientação a objetos de [O2t93, BDK92], e pode ser embutido em um Sistema Gerenciador de Banco de Dados Orientado a Objetos (SGBDOO) tornando-o capacitado a prover as classes básicas necessárias para a construção de um SIG temporal.
- Utiliza uma estrutura de abstração multi-nível como em [CCM+94], provendo o isolamento entre as classes geográficas da aplicação e o nível físico dos dados.

A estrutura de dados proposta para armazenar os dados do modelo geográfico tem como características:

- É baseada na estratégia de [Bra94], onde há agregação de atributos temporais (*timestamps*) ao esquema das classes que precisam ter a sua variação no tempo armazenada. Esta estratégia facilita a conversão de bases de dados atemporais em temporais.
- Provê os métodos espaço-temporais necessários para a efetivação de uma consulta espaço-temporal.

A taxonomia de consultas espaço-temporais apresentada tem as seguintes características:

- Está fundamentada no conceito de dominância dimensional de [Lan95].
- Divide uma consulta nos componentes básicos: seleção, projeção e agrupamento.

O restante deste capítulo apresenta alguns conceitos básicos e uma visão geral dos problemas a serem estudados.

1.2 Terminologia

No decorrer desta dissertação são usados alguns termos que têm significados distintos para cada autor. A seguir iremos explicitar os conceitos que estão associados a esses termos empregados daqui em diante.

[Gat91] diz que a geografia vista num banco de dados geográfico parece um grande reservatório ocupado por *objetos*. Os objetos são portanto uma representação no banco de dados das *entidades geográficas*. As entidades geográficas são os fenômenos geográficos do mundo real (por exemplo: florestas, rios, estradas, cidades, dentre outras).

Ainda segundo o autor, o *espaço geográfico* definido através dos *relacionamentos espaciais* entre as entidades é mais consistente com o papel da geografia em muitas formas de planejamento. O espaço geográfico é o meio físico onde as entidades geográficas coexistem. Um relacionamento espacial é uma informação derivada a partir do posicionamento de uma entidade em relação à outra entidade. Por exemplo, a distância entre duas cidades é um tipo de relacionamento entre duas entidades geográficas. Portanto é semanticamente mais rico perguntar a distância entre as duas cidades, do que perguntar a distância entre as coordenadas geográficas correspondentes das duas cidades. Apesar de terem grande importância numa análise geográfica, os relacionamentos espaciais não têm sido explorados satisfatoriamente nos sistemas geográficos atuais.

Os dados geográficos são comumente caracterizados como tendo três componentes fundamentais: *atributo*, *localização* e *tempo* [Aro89, Her94]. O componente *atributo*¹ descreve as propriedades temáticas de uma entidade geográfica. O componente *localização* contém a descrição das características espaciais de uma entidade geográfica. Uma *representação espacial* de uma entidade geográfica descreve a forma geométrica associada à posição geográfica dessa entidade. Uma entidade geográfica, armazenada como um *objeto* em um banco de dados orientado a objetos, pode ter diversas representações espaciais conforme a escala utilizada. Por exemplo, uma cidade pode ser representada por um ponto num mapa de escala pequena e por um polígono num mapa de escala grande. O componente *tempo* descreve os períodos em que os valores daqueles dados geográficos são válidos.

Para facilitar a análise geográfica, um mapa é virtualmente dividido em *camadas temáticas*, onde cada camada consiste num conjunto de características geográficas logicamente relacionadas. As características que são agrupadas em cada camada variam de

¹ Também chamados de atributos convencionais, ou atributos não-espaciais.

acordo com a conveniência do usuário. Por exemplo, estradas de ferro e estradas rodoviárias podem ser agrupadas em uma única camada de transporte, e lagos e rios podem ser agrupados como a camada de hidrografia. Entretanto um mesmo objeto do banco de dados pode pertencer a mais de uma camada. Por exemplo, os objetos do tipo *pais* poderiam pertencer às camadas sócio-política, religião, idioma, dentre outras, a partir de valores dos seus atributos convencionais.

Nem todas as análises requerem usar todas as camadas simultaneamente. Em alguns casos um pesquisador irá estudar os relacionamentos entre apenas algumas camadas específicas [FL95].

1.3 Alguns Problemas Existentes em SIG

Vários problemas em SIG permanecem ainda sem uma solução satisfatória. A seguir veremos alguns deles:

1. A geografia é infinitamente complexa e deve ser generalizada, aproximada e abstraída para que seja representada dentro de dimensões finitas de um sistema computacional discreto. Um grande percentual dos problemas em SIG advém da necessidade de se utilizar dados geográficos imprecisos provenientes de fontes diferentes de aquisição de dados com níveis distintos de generalização e escalas incompatíveis.
2. Muitos dos nossos conceitos de percepção espacial são ambíguos e imprecisos. Apesar dessa ambigüidade semântica, linguagens de consulta cada vez mais poderosas e interfaces mais amigáveis podem tornar esses sistemas mais fáceis de serem usados e portanto mais eficientes.
3. O uso de dados geográficos envolve problemas de ordem legal, como o direito de propriedade sobre os dados, privacidade sobre dados pessoais, possibilidade de cobrar tarifas de utilização da base de dados, dentre outros [NS93].
4. A cartografia desenvolveu várias técnicas de apresentação da informação espacial que podem ser aperfeiçoadas com os novos recursos de multimídia e técnicas tridimensionais de visualização. Algumas dessas novas técnicas talvez consigam representar melhor análises espaciais complexas, tornando os SIG mais úteis nas tomadas de decisão.
5. Um problema prioritário é encontrar um modelo espaço-temporal padrão para SIG. Em cada classe específica de aplicações existe um ou mais modelos que melhor se adaptam [FG90, Fra90]. Para cada modelo, existe um conjunto de estruturas de dados que melhor o implementa [FR93].

6. O número de possíveis funções para um SIG parece ser ilimitado, mas um certo conjunto de operações primitivas pode ser encontrado, como por exemplo: sobreposição de mapas, zonas de fronteira e o cálculo de áreas [MGR91]. Para isso deve-se criar uma taxonomia das funções e consultas dos SIG.

Essa dissertação propõe uma solução para os dois últimos itens.

1.4 Processos Espaço-Temporais

O estudo da dinâmica espaço-temporal não é novo e nem exclusivo da geografia, mas ainda não foi incorporado a SIG. Em qualquer processo geográfico sendo observado, é necessário incluir o componente temporal nos dados para representar as *mudanças* e derivar relacionamentos de causa e efeito. Esforços feitos para incorporar a dimensão temporal nos SIG têm servido para revelar que muitos problemas estão no nível conceitual.

Além da contínua investigação dos processos sociais, tais como padrões de atividade diária e crescimento urbano, a necessidade de avançar a compreensão dos efeitos das atividades humanas no meio ambiente é atualmente vista com grande interesse. Bancos de dados estão sendo usados para estudar alguns fenômenos ecológicos como os movimentos das massas de ar que indicam mudanças de clima, o movimento de correntes marítimas que estão relacionadas com a vida de todo o ecossistema marinho, as mudanças das coberturas vegetais que afetam a fauna terrestre, os efeitos da poluição dos rios sobre o abastecimento de água e a atividade de pesca de uma região, e finalmente os efeitos da poluição do ar sobre a camada de ozônio do planeta e o conseqüente derretimento das camadas polares. Pesquisadores em gerenciamento de recursos naturais estão tendo uma visão mais conservacionista e integrada procurando obter maior diversidade e produtividade a longo prazo, evitando assim uma exploração desordenada desses recursos.

As mudanças na geografia muitas vezes ocorrem em ciclos e têm uma importância fundamental para o entendimento dos processos ambientais. Comparando-se a configuração atual das entidades geográficas armazenadas no sistema com os estados antigos dessas entidades, podemos identificar o que foi modificado e, a partir daí, tentar descobrir as razões dessa mudança. Os processos ambientais são estudados através da observação empírica de seus efeitos sobre as entidades geográficas. Uma vez estabelecida uma hipótese baseada em acontecimentos passados para explicar alguns fenômenos atuais, essa hipótese é usada para prever as possibilidades de ocorrência desse ou de outros fenômenos no futuro. Quanto mais a hipótese é validada ao longo do tempo, maior é a sua credibilidade, e portanto é maior a sua eficiência na previsão de eventos futuros.

Os SIG têm evoluído de simples mapas automatizados para poderosas ferramentas de análise, planejamento e decisão, através da combinação e integração de todas as categorias de dados georeferenciados. Esses sistemas devem destacar a localização espacial dos

dados, facilitar o gerenciamento dos recursos que modelam, auxiliar em análises estatísticas das informações geográficas, e também ajudar no planejamento através de mecanismos de simulação e previsão de dados. Apesar desses sistemas serem capazes de analisar grande quantidade de dados de uma forma integrada e flexível, ainda são incapazes de representar as mudanças que ocorrem ao longo do tempo, ficando limitados a uma visão instantânea da realidade que modelam.

Muitas pesquisas foram feitas em bancos de dados espaciais e bancos de dados temporais isoladamente. Existem também alguns trabalhos sobre análises geográficas espaço-temporais [ASS93], mas a maioria falha conceitualmente por não considerar o processo de mudança na geografia. [Lan95] coloca o modelo conceitual de mudança temporal como um dos requisitos técnicos para o desenvolvimento de um SIG temporal. Além disso, nenhum desses trabalhos permite total liberdade de variação nas dimensões espaciais e temporais simultaneamente. A principal razão para isso é que dados verdadeiramente espaço-temporais são raros e os que existem são difíceis de tratar devido ao tamanho e assimetria.

A adição da dimensão temporal deve trazer a possibilidade de não apenas armazenar o histórico de estados passados das entidades geográficas, mas também de analisar o mundo como um grande reservatório de entidades dinâmicas interrelacionadas por processos (naturais ou não) que funcionam como catalisadores de mudanças nessas entidades. Sendo assim, a integração das dimensões espaciais e temporais nos SIG tem muito mais potencial para análise do que a soma das funcionalidades dessas dimensões independentes.

1.5 Consultas e Aplicações de SIG Temporais

Os SIG podem ser classificados quanto ao tipo de aplicação em duas grandes categorias:

- Aplicações de administração de recursos naturais e controle ambiental

Essas aplicações utilizam predominantemente o formato raster de dados adquiridos geralmente de imagens de satélite cobrindo grandes áreas. Normalmente esse tipo de sistema usa pequenas escalas. Alguns exemplos dessas aplicações são: gerenciamento florestal, planejamento urbano, controle ecológico de regiões e gerenciamento agrícola.

No contexto temporal, esse sistemas poderiam ajudar na prevenção de incêndio florestal identificando as áreas críticas, no controle do desmatamento, no gerenciamento de plantações agrícolas, análise da propagação de uma mancha de óleo no mar, controle de níveis de poluição, acompanhamento de migração de espécies animais, dentre outras.

- Aplicações de administração de instalações e redes de utilidade pública

Também conhecidas como AM/FM (automated mapping / facilities management), essas aplicações servem para gerenciar estruturas complexas com grande número de componentes interligados por uma grande malha. Um componente pode ser subdividido em componentes menores e composto de vários atributos que compõem a sua especificação técnica. Utilizam predominantemente o formato vetor de dados. Alguns exemplos desses sistemas são: rede de telefonia, rede de tráfego, sistema de geração de energia hidroelétrica e rede de distribuição de energia.

No contexto temporal, esses sistemas podem auxiliar no planejamento e simulação de situações, como por exemplo, na implantação de um novo gerador de energia, e ainda, sua localização. Além disso, esses sistemas podem otimizar o funcionamento da rede identificando os períodos de pico de utilização da rede e sugerindo caminhos alternativos de distribuição para desfazer os "gargalos". A resolução de problemas através da detecção de irregularidades como a identificação de áreas com um histórico de alto índice de falhas também pode trazer grandes economias.

Segundo [Lan95], SIG temporais armazenam os valores históricos dos dados geográficos e portanto podem responder a consultas do tipo: *"Onde e quando houve mudança ?"; "Que tipo de mudanças ocorreram ?"; "Com que frequência ocorreram as mudanças ?"; "Qual foi a periodicidade das mudanças ?"*. Diante dessas informações o sistema pode descobrir, ainda, se existe algum padrão temporal, que tendências são visíveis e qual processo é o responsável pela mudança. Os tipos de consultas espaço-temporais de acordo com o domínio de busca da consulta são:

1. Consulta Temporal Simples: *"Qual é o estado do objeto A no tempo T ?"*. (fig 1.1a)
2. Consulta Temporal de Intervalo: *"O que aconteceu com o objeto A durante um intervalo T1 a T2 ?"*. (fig 1.1b)
3. Consulta Espaço-Temporal Simples: *"Qual é o estado da região R no tempo T ?"*. (fig 1.1c)
4. Consulta Espaço-Temporal de Intervalo: *"O que aconteceu com a região R no intervalo T1 a T2 ?"*. (fig 1.1d)

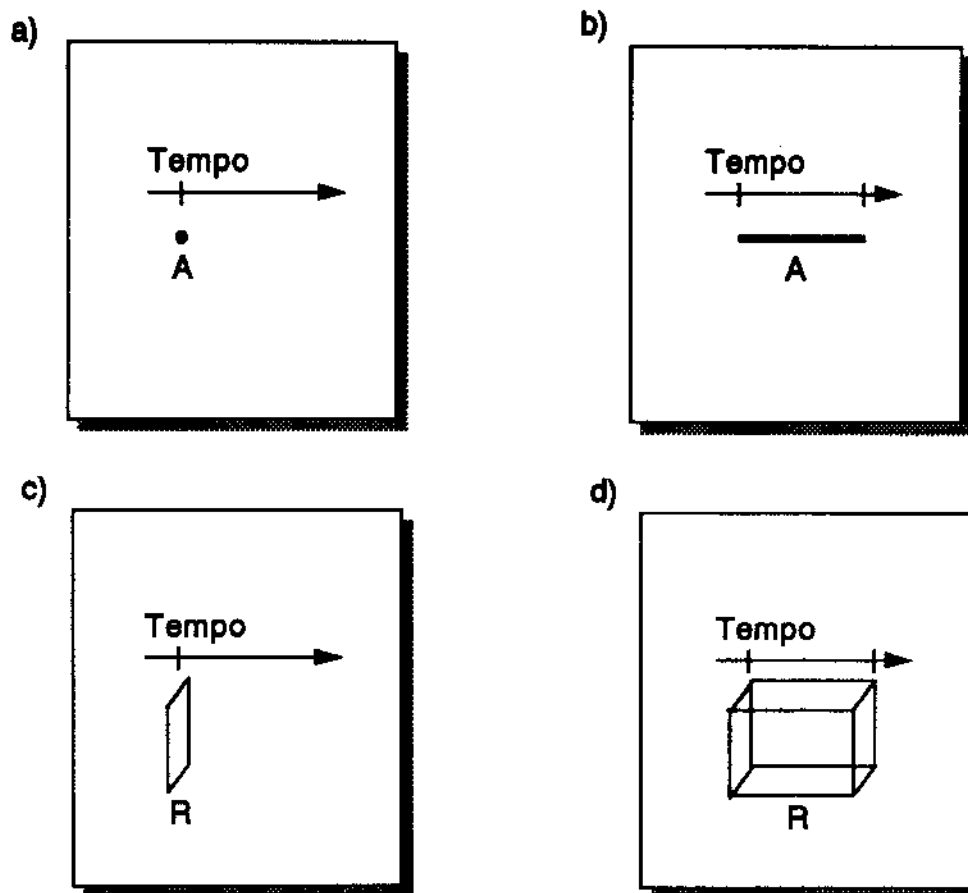


Fig 1.1 - Consultas Espaço-Temporais

[Peu94], reconhecendo a importância de que os SIG estejam preparados para lidar com a dimensão de tempo, propõe uma abordagem **tríade** unificando aspectos temporais, localizacionais, e os atributos das entidades geográficas. Segundo a autora, para cada informação existem as visões de localização daquela entidade (georeferência), da composição interna daquela entidade (objeto) e do período de existência daquela entidade (tempo). Essa abordagem permite que o usuário faça três tipos básicos de perguntas:

- Quando + Onde → O quê
- Quando + O quê → Onde
- O quê + Onde → Quando

[MJ93] classifica os tipos de consultas de um SIG temporal em:

- Apresentação dos dados armazenados.
- Determinação dos relacionamentos espaciais entre entidades diferentes.

- Simulação e comparação de cenários alternativos baseados na combinação de camadas de dados.
- Previsão do futuro através da análise de tendências.

[Lan95] classifica as principais funcionalidades trazidas por um SIG temporal como: histórico, análise, atualização, controle de qualidade, restrições temporais e apresentação de animações.

1.6 Arquiteturas Existentes para SIG

Os SIG estão implementados em diferentes arquiteturas: proprietária, em camadas ou bancos de dados estendidos [MP94]. Os sistemas proprietários (por exemplo, GRASS) armazenam todos os seus dados em arquivos comuns do sistema operacional, fornecendo uma biblioteca de funções para acessá-los, manipulá-los e apresentá-los. Esses sistemas são limitados sob a perspectiva de um SGBD por não terem funcionalidades como recuperação de falhas, controle de concorrência, otimização de consultas, suporte a transações e independência física dos dados.

Outros SIG usam uma abordagem híbrida na qual um SGBD é usado para armazenar os atributos não-espaciais enquanto que os dados espaciais ficam com um gerenciador de dados espaciais próprio que utiliza arquivos do sistema operacional (por exemplo, ARC/INFO [Esr92]). Enquanto essa abordagem tem tido sucesso, ela complica a otimização de consultas e execução, principalmente em um ambiente multiusuário. Uma terceira abordagem exemplificada por Montage, GEO e Paradise utiliza um sistema integrado onde todos os dados são armazenados no banco de dados [DKL+94].

Os bancos de dados utilizados podem ser relacionais, relacionais estendidos ou orientados a objetos. Os sistemas geográficos baseados em SGBD relacionais ou relacionais estendidos têm algumas limitações para incorporar os dados espaciais:

1. [Ege92] expõe várias razões por que uma extensão da linguagem SQL não é suficiente para atender à complexidade de consultas de um SIG, como por exemplo: a falta de interatividade dinâmica com o resultado de uma consulta, a falta de *identidade* para tuplas, a ausência de consultas de metadados e de consultas que expliquem o processo de raciocínio utilizado para alcançar o resultado, falta de respostas qualitativas e a impossibilidade de desassociar a recuperação das informações da apresentação das informações. Apesar disso existem esforços no sentido de estender a linguagem SQL, como por exemplo a extensão temporal ao padrão SQL-92 chamada de TSQL2 [SAA+94].

1. Introdução

2. O modelo relacional dos SGBD convencionais não é compatível com a complexidade dos dados geográficos. Essa diferença de modelo de dados entre as aplicações e o banco de dados é chamada de *impedance mismatch*.
3. SIG precisam gerenciar e processar dados com estruturas irregulares (por exemplo, combinações de tuplas, listas e conjuntos) e não apenas tuplas de tamanho regular como no modelo relacional. Além disso, as estruturas dos dados para modelagem de vários tipos de associação entre os dados devem ser suportadas também.
4. As aplicações SIG utilizam tipos de dados geográficos específicos (ou classes) que poderiam ser encapsulados e reutilizados na construção de tipos mais complexos. Por exemplo o tipo de dado *rio* é usado para a formação de uma *bacia hidrográfica*. Para isso, o modelo de dados do banco de dados deve ser extensível, ou seja, deve permitir a inclusão de novas classes através de *herança* (simples ou múltipla) e composição. A herança garante que a nova classe (classe filha) herde as propriedades e métodos de uma classe existente (classe pai).

Os SGBDOO têm sido apontados como a melhor solução para lidar com a realidade complexa e extremamente diversificada das entidades geográficas [WHM90, MMS93]. Esses bancos de dados têm características importantes para as aplicações não convencionais, as quais nunca tiveram muito suporte de um SGBD relacional, pois estes provaram ser pouco produtivos nessa área. Os SGBDOO tentam mesclar os conceitos trazidos pelo paradigma de orientação a objetos com os conceitos de bancos de dados tradicionais [Dit94]. Além de não ter as limitações citadas acima, algumas vantagens trazidas pelo paradigma de orientação a objetos são:

1. Objetos podem ser reutilizados para gerar outros objetos mais especializados ou generalizados. Se os objetos são bem escolhidos, o usuário pode facilmente formar um modelo mental do sistema [OvdB89].
2. As características de *ligação tardia* (*late binding*) e *polimorfismo* das linguagens orientadas a objeto desobrigam o usuário de ter que saber nomes de atributos ou seus tipos, preservando assim o encapsulamento dos objetos [MPB93].
3. Os tipos abstratos de dados (ADT) podem ser criados para prover ao programador um isolamento de detalhes de implementação [Pri89].

1.7 Padrões de Modelagem

Alguns padrões geográficos foram criados usando o paradigma de orientação a objetos. O Spatial Data Transfer Standard (SDTS) é, dentre estes, certamente o mais difundido especialmente nos EUA, devido a uma resolução do presidente americano

obrigando todos os órgãos governamentais aderirem ao padrão [Cli94]. A preocupação do governo americano com a padronização dos dados geográficos se deveu à constatação feita através de uma pesquisa realizada pelo OMB (Office of Management and Budget) de que mais de 4 bilhões de dólares eram gastos por ano com dados geográficos.

Um dos padrões geográficos que considera a dimensão temporal é o Spatial Archive and Interchange Format (SAIF) [Mur95]. O SAIF é um padrão de desenvolvimento de SIG criado na província de British Columbia no Canadá para facilitar o intercâmbio de dados espaciais e espaço-temporais entre os SIG. Este padrão foi aceito como um padrão nacional no Canadá e tem como características principais:

- Segue o paradigma de orientação a objetos com herança múltipla.
- A sua biblioteca de funções (*toolkit*) usa técnicas de paralelismo para agilizar o processamento e acesso aos dados.
- Incorpora o tempo semanticamente no modelo do banco de dados através de várias classes temporais ou espaço-temporais que podem ser agregadas formando o objeto geográfico temporal.
- Propõe modelar qualquer dado geográfico variante no tempo através da criação de classes geográficas combinando ou herdando de algumas das mais de 300 classes base fornecidas pelo sistema.
- Usa uma linguagem de definição de dados (CSN) e uma linguagem de notação sintática de objetos (OSN) proprietárias, ambas no formato ASCII.
- Se propõe a ser compatível com o novo padrão SQL ainda em desenvolvimento: SQL3.

O SAIF falha por não apresentar uma proposta para o processo de *mudança geográfica*. Além disso a incorporação semântica do tempo no padrão aumenta bastante o número de classes primitivas tornando-o mais complexo e pouco compreensível ao usuário.

1.8 Organização da Dissertação

Nesta dissertação estendemos a semântica das principais consultas geográficas incorporando a dimensão temporal e desenvolvemos uma taxonomia para essas consultas espaço-temporais em SIG. Além disso, caracterizamos os processos de mudanças geográficas e mostramos que eles podem ser vistos como entidades reais. O nosso objetivo é mostrar como um SGBD pode ser adaptado às necessidades dos SIG modernos. Para isso procuramos demonstrar a viabilidade de se criar uma camada de gerenciamento de dados espaço-temporais utilizando a estratégia da camada de gerenciamento temporal desenvolvida por [Bra94] sobre o SGBDOO O₂.

A dissertação está estruturada da seguinte forma:

- O capítulo 2 apresenta uma revisão bibliográfica das principais propostas de bancos de dados temporais, apresentando e descrevendo os conceitos usados na literatura sobre os respectivos modelos temporais. O modelo temporal TOODM de [Oli93] adotado nessa dissertação e a implementação desse modelo feita por [Bra94] são também brevemente apresentados.
- O capítulo 3 propõe um modelo geográfico temporal orientado a objetos baseado em extensões a [CCM+94]. Um modelo geométrico é apresentado para representar o componente localizacional dos objetos. Várias considerações são feitas sobre a representação dos processos de mudança geográfica e relacionamento entre processos.
- O capítulo 4 propõe uma taxonomia de consultas de um SIG temporal distinguindo as consultas feitas na dimensão espacial, temporal e espaço-temporal.
- O capítulo 5 descreve as estruturas de dados necessárias para implementar o modelo proposto e mostra como consultas da taxonomia do capítulo 4 podem ser implementadas nestas estruturas.
- Finalmente o capítulo 6 apresenta as conclusões e possíveis extensões.

Capítulo 2

Conceitos de Bancos de Dados Temporais

Os bancos de dados convencionais foram projetados para armazenar e processar os dados mais recentes, ou seja, os dados atuais. À medida que novos valores se tornam disponíveis através das atualizações, os valores que existiam antes são removidos do banco de dados. Assim, esses bancos de dados capturam apenas uma imagem instantânea da realidade (*snapshot*). Bancos de dados convencionais são insuficientes para aquelas aplicações que necessitam dos dados passados e/ou futuros. A inexistência de um sistema temporal força essas aplicações a criar seus próprios métodos para gerenciar suas informações temporais. Num sentido mais amplo, um banco de dados que mantém dados passados, presentes e futuros é chamado de banco de dados temporal [TCG+95].

O excessivo volume de dados que esses sistemas acumulam devido à política de manutenção dos dados antigos tem sido contra-balançado pelos custos decrescentes dos meios de armazenamento de dados, como por exemplo os drives óticos, e pelas técnicas de compressão de dados. A capacidade dos bancos de dados em interpretar a dimensão do tempo simplifica o desenvolvimento de aplicações que necessitam armazenar e processar versões antigas dos dados, e/ou informações que variam com o tempo.

Apesar de terem sido criados vários modelos de dados temporais, pouco tem sido feito para unificar as características comuns desses modelos e formar uma base de conceitos. A comunidade científica tem trabalhado no sentido de criar um vocabulário padrão para aumentar a eficiência de cooperação entre os pesquisadores, propiciando a definição de uma linguagem de consulta temporal consensual, seja estendendo a linguagem SQL, para suportar as dimensões temporais, ou desenvolvendo uma nova linguagem SQL orientada a objetos mais apta a incorporar o melhor modelo de dados temporal [PSE+94].

A busca pelo modelo que melhor incorpora a semântica temporal tem sido ofuscada pela preocupação com outras considerações como apresentação e implementação. O modelo de dados temporal deve ser criado puramente em termos das suas propriedades semânticas, incluindo possivelmente múltiplos modelos de representação e apresentação. A

ênfase é em flexibilidade, de forma que vários otimizadores de consultas, algoritmos de atualização de dados, técnicas de indexação e modelos de representação estejam disponíveis no SGBD. Essa abordagem é consistente com a tendência mundial de arquiteturas abertas. [PSE+94].

Em [PSE+94], uma infraestrutura básica é apresentada para a criação das próximas gerações de SGBD temporais, que deverão incorporar alguns conceitos, como por exemplo: modelo de tempo ramificado, indeterminância temporal, valores de tempos relativos, periódicos e aproximados, representação do tempo com intervalos disjuntos, compatibilidade com as aplicações antigas, dentre outras.

Este capítulo mostra os principais conceitos abordados, restritos a SGBD relacionais. Finaliza com um modelo de banco de dados temporal orientado a objetos, que será estendido na dissertação para modelagem espaço-temporal. Uma revisão bibliográfica extensa sobre bancos de dados temporais é apresentada em [Oli93].

2.1 Terminologia

Muitos conceitos definidos na literatura sobre bancos de dados temporais são novos ou ainda não estão muito bem consolidados. No decorrer desta dissertação utilizaremos os conceitos apresentados a seguir:

2.1.1 Atributo (ou entidade¹) Invariante-no-Tempo

Um atributo invariante no tempo² é um atributo cujo valor está condicionado a não mudar ao longo do tempo. Em termos funcionais, é uma função de valor constante ao longo do tempo.

2.1.2 Atributo (ou entidade) Variante-no-Tempo

Um atributo variante no tempo³ é um atributo cujo valor não está restrito a ser constante ao longo do tempo. Em outras palavras, pode ou não mudar com o tempo.

¹ Uma entidade do mundo real pode ser representada em um banco de dados na forma de uma tupla (no modelo relacional) ou objeto (no modelo orientado a objetos).

² Também chamado de dado estático.

³ Também chamado de dado temporal.

2.1.3 Chronon (Ponto de Tempo)

Um *chronon*⁴ é a menor duração de tempo suportada por um SGBD temporal, ou seja, é uma unidade indivisível de tempo. Um *chronon* específico é um subintervalo de duração fixa no eixo de tempo.

2.1.4 Densidade do Tempo

O tempo é contínuo por natureza. Entretanto existem três visões de tempo:

- Discreta - isomorfa aos números naturais, ou seja, existe a noção de que para cada momento de tempo existe um único sucessor.
- Densa - isomorfa aos números racionais, com a noção de que entre quaisquer dois pontos de tempo existe outro ponto de tempo.
- Contínua - isomorfa aos reais, ou seja, não existem “buracos” entre os pontos de tempo.

A interpretação discreta de tempo tem sido comumente adotada pela comunidade de pesquisa em bancos de dados temporais pela sua simplicidade e relativa facilidade de implementação, pois os instrumentos de medição e contagem de tempo são imprecisos (ou têm uma precisão limitada).

2.1.5 Ordem do Tempo

Existem três modelos *estruturais* de tempo:

- Linear - o tempo está linearmente ordenado: para dois pontos de tempo (*chronons*) distintos t e t' , ou t é anterior a t' , ou t' é anterior a t . Em outras palavras, o tempo é um conjunto de pontos de tempo ordenados e igualmente espaçados, e denotado por T , onde $T = \{0, 1, 2, \dots, \text{now}, \dots\}$. O símbolo 0 é o início relativo, e *now* é uma constante especial que representa o tempo atual. O valor de *now* muda à medida que o relógio avança. Qualquer ponto além de *now* é tempo futuro.
- Ramificado - o tempo é um grafo acíclico com ramificações em qualquer ponto do seu eixo principal: para dois pontos de tempo (*chronons*) distintos t e t' , t pode ser simultâneo a t' . Em qualquer ponto de T pode haver um outro conjunto de pontos

⁴ Outros termos às vezes são empregados ao invés de *chronon* ou ponto de tempo. Os termos *instante* e *momento* significam um ponto no modelo contínuo de tempo, ou uma unidade indivisível de tempo no modelo discreto (*chronon*). Os termos *quantum de tempo* e *unidade de tempo* podem dar uma idéia de intervalo de tempo, ou de uma unidade divisível de tempo.

linearmente ordenados T' , e assim sucessivamente. Neste caso, podemos considerar que os pontos de tempo estão em uma ordem parcial. Quando várias alternativas devem ser consideradas como possíveis evoluções dos dados temporais, como em sistemas de diagnósticos, investigações e planejamento, o modelo de tempo ramificado é usado, definindo uma ordem parcial aos pontos.

- Circular - pode ser usado para representar eventos recorrentes. Neste caso em que os eventos são periódicos, nenhum relacionamento de ordenação é definido. [CL93] define ainda os ciclos físicos (baseados em calendários e tabelas de tempo) e os ciclos semânticos ou circuitos (por exemplo, ciclo de uma plantação agrícola).

2.1.6 Dimensões Temporais

Uma taxonomia de tempo em bancos de dados e em modelos de dados temporais foi desenvolvida em [Sno86]. Os tempos apresentados a seguir são ortogonais e podem ser suportados separadamente ou conjuntamente em um banco de dados.

- Tempo-de-Transação⁵ (*Transaction Time*)

Um valor é armazenado em um banco de dados em algum ponto no tempo. Após ser armazenado, ele pode ser recuperado. O tempo de transação de um valor no banco de dados é o tempo em que aquele valor foi armazenado. Os tempos de transação são consistentes com a ordem de serialização das transações. Os valores de tempo-de-transação não podem ser maiores que o tempo atual. Além disso, por ser impossível mudar o passado, os tempos de transação não podem ser mudados. Os tempos de transação podem ser implementados usando os tempos de confirmação (*commit time*) das transações de atualização registrados pelo relógio do sistema.

- Tempo-Válido⁶ (*Valid Time*)

O tempo válido de um fato é o tempo quando aquele fato aconteceu na realidade modelada. O conceito de reencarnação exprime a capacidade de uma entidade no mundo real “morrer” (ser desativada) e posteriormente “renascer” (reativada). Portanto um fato pode ser associado com qualquer número de eventos e intervalos disjuntos combinados. Por exemplo, um estudante pode estudar por dois anos, trancar o curso por 6 meses e então reativar a sua matrícula para terminar o seu curso. O tempo-válido dos fatos é fornecido pelo usuário.

- Tempo-Definido-pelo-Usuário (*User-Defined Time*)

⁵ Outros termos usados para tempo de transação são: tempo de registro, tempo do banco de dados, tempo extrínseco e tempo físico.

⁶ Outros termos usados para tempo válido são: tempo do mundo real, tempo lógico, tempo intrínseco e tempo do usuário.

O tempo-definido-pelo-usuário é um atributo de domínio não interpretado pelo banco de dados, usado para armazenar data e hora de alguns fatos. Este domínio tem mais semelhança com domínios de atributos convencionais, do que com os domínios temporais, pois não existe um suporte especial na linguagem de consulta para tratá-los. O tempo-definido-pelo-usuário pode armazenar atributos como data de aniversário e data de contratação. Este tipo de tempo é gerenciado pelo usuário e é o mais fácil de ser implementado.

Esses tipos de tempo induzem diferentes tipos de bancos de dados [Tabela 2.1]. Um banco de dados convencional não suporta o tempo válido e nem o tempo de transação, e é chamado de **banco de dados instantâneo** (*snapshot database*), pois contém apenas uma imagem instantânea da realidade. Um **banco de dados tempo-válido** contém todo o histórico de uma empresa (*historic database*). Um **banco de dados de tempo-de-transação** suporta o tempo de transação e armazena todas as alterações realizadas sobre os dados. Este banco de dados permite voltar para qualquer estado passado no banco de dados (*rollback database*) e fazer consultas nesse estado, pois não há exclusão física dos dados. Portanto ficam registrados os erros, provendo todas as informações necessárias para uma auditoria e simplificando os mecanismos de correção de erros, contabilidade e recuperação de falhas. As alterações feitas podem ser consultadas para encontrar erros e evitar a inconsistência dos dados. Um **banco de dados bitemporal** registra tanto o tempo válido quanto o tempo de transação e combina as características dos anteriores. A sequência completa de atualizações e os valores atualizados são disponíveis para consulta.

SGBD	Dimensões Temporais	
	Tempo de Transação	Tempo Válido
Instantâneo		
Rollback	✓	
Histórico		✓
Bitemporal	✓	✓

Tab.2.1 - SGBD x Dimensões Temporais

Os bancos de dados de tempo-de-transação armazenam o histórico de atividades do banco de dados e os bancos de dados de tempo-válido armazenam o histórico de atividades do mundo real. Além disso, a alteração do valor de um dado em um banco de dados de tempo-válido implica na sua substituição pelo valor novo, enquanto que nos bancos de dados de tempo-de-transação o valor novo é adicionado ao valor antigo como um novo estado [Sno86].

Em um banco de dados de tempo-de-transação é possível fazer consultas como se estivesse num estado passado, ou seja, todas as alterações feitas desse estado em diante não são consideradas. Por exemplo, "Qual seria o salário do mês passado de Pedro se estivéssemos em outubro de 1993?". O sistema volta para o estado do banco de dados ativo em outubro de 1993 e consulta o valor do salário de Pedro no mês anterior (setembro de 1993) conforme se sabia naquela época.

Os SGBD temporais devem suportar todos os modelos temporais, mas nem todas as aplicações temporais precisam de suporte para ambas as dimensões temporais. O projetista de aplicação deve conhecer bem a semântica dos modelos temporais para não desperdiçar recursos, evitando definir atributos invariantes no tempo como temporais [Sno86].

Uma relação temporal que inclua o tempo válido é *retroativa* se cada dado armazenado de qualquer instância é sempre válido no passado. O conceito pode ser estendido para tuplas individuais e atualizações. Por exemplo, uma atualização retroativa seria: "Maria foi promovida a professora titular no mês passado".

Uma relação temporal que inclua o tempo válido é *proativa* se cada dado de qualquer instância da relação era válido no futuro quando estava sendo armazenado. O conceito pode ser estendido para tuplas individuais e atualizações [JCE+94]. Por exemplo, uma atualização proativa seria: "Pedro será contratado no próximo mês".

Um banco de dados temporal que suporta somente o tempo-de-transação não permite atualizações retroativas ou proativas, pois uma tupla se torna efetiva no momento da sua inclusão no banco de dados.

2.1.7 Span e Tempo-de-Vida (*Lifespan*)

O termo *span*⁷ significa uma duração direcionada de tempo. Uma duração de tempo é uma quantia de tempo com um comprimento determinado em número de *chronons*, mas sem *chronons* específicos de início e fim. Por exemplo, a duração "uma semana" tem o comprimento de 7 dias, mas pode se referir a qualquer bloco de 7 dias consecutivos. Um *span* pode ser positivo, denotando passagem de tempo para o futuro, ou negativo, denotando volta no tempo para o passado. Um *span* pode ter um comprimento fixo (por exemplo, semana) ou variável (por exemplo, mês).

O tempo-de-vida de um banco de dados é o tempo sobre o qual ele é definido. O tempo-de-vida tempo-válido de uma entidade armazenada no banco de dados se refere ao tempo de existência dessa entidade no mundo real, enquanto que o tempo-de-vida tempo-de-transação se refere ao tempo de existência dessa entidade no banco de dados.

2.1.8 Marca-de-Tempo (Timestamp)

Uma marca-de-tempo é um valor de tempo associado a um atributo, a uma tupla ou a uma relação em um banco de dados relacional, ou a um objeto em um banco de dados orientado a objetos. O conceito pode ser especializado quanto à dimensão temporal para

⁷ Alguns sinônimos de *span* são *duração*, *intervalo* e *distância de tempo*. Entretanto um intervalo normalmente tem os seus *chronons* de início e fim fixos, e uma duração é geralmente considerada ser não direcional, ou seja, sempre positiva.

marca-de-tempo de tempo válido ou marca-de-tempo de tempo de transação. Existem três formas de representação de uma marca-de-tempo:

- **Evento** - Um evento⁸ é um instante isolado no tempo. Um evento é dito ocorrer em um tempo t se ele ocorrer em qualquer tempo durante o chronon representado por t .
- **Intervalo** - Um intervalo de tempo⁹ representa a duração de tempo entre dois eventos e pode ser representado por um conjunto de pontos de tempo (chronons) consecutivos delimitados pelos seus pontos extremos. O intervalo fechado $[i, f]$ contém todos os pontos de tempo entre i e f (inclusive i e f), enquanto que o intervalo semi-aberto $[i, f)$ contém todos os pontos de tempo de $[i, f]$ exceto f . Um ponto de tempo t pode ser considerado um intervalo de tempo degenerado $[t, t]$.
- **Elemento Temporal** - Um elemento temporal é uma união finita de regiões n -dimensionais de tempo. Os elementos temporais são fechados sob as operações de união, interseção e complemento e podem ser especializados em elementos de tempo-válido, elementos de tempo-de-transação e elementos bitemporais. Nesses casos, eles são definidos como a união finita de intervalos de tempo-válido, intervalos de tempo-de-transação e retângulos bitemporais, respectivamente. Um elemento temporal pode ser considerado como uma união disjunta de intervalos de tempo. Qualquer intervalo ou elemento temporal que inclui a constante especial *now* aumenta a medida que o valor de *now* avança.

[TCG+95] argumenta que o elemento temporal é a representação de tempo mais compacta e intuitiva para bancos de dados temporais. No mundo real, as propriedades de uma entidade mudam, mas ainda a consideramos como sendo a mesma entidade. Seguindo este raciocínio, nós devemos pensar no valor de um atributo como uma função de tempo. Isso nos ajuda a prevenir a fragmentação do histórico de uma entidade no banco de dados. Se intervalos ou instantes são usados como marcas-de-tempo, o histórico de um atributo é distribuído em várias tuplas, aumentando a complexidade da linguagem de consulta do ponto de vista do usuário [GC93].

2.1.9 Aninhamento Temporal

Duas estratégias diferentes para a incorporação da dimensão do tempo no modelo relacional apareceram na literatura. Na primeira, o esquema de uma relação é expandido para incluir um ou mais atributos temporais distintos (por exemplo, *tempo_início* e *tempo_fim*) para armazenar o período de tempo sobre o qual uma entidade é considerada válida. Essa abordagem é chamada na literatura de **marcação-de-tempo na tupla** (*tuple*

⁸ Um evento também poderia ser chamado de instante ou momento, mas estes termos podem ser confundidos com o termo distinto chronon. Apesar de um evento ser pontual, instrumentos de medição de tempo marcam a ocorrência de eventos em termos de intervalos de tempo e não de pontos de tempo.

⁹ O termo *período de tempo* às vezes é usado como sinônimo de intervalo de tempo, mas período pode sugerir um ciclo ou um fenômeno recursivo.

timestamping) ou modelo na primeira-forma-normal (1NF). A outra abordagem, chamada de *marcação-de-tempo no atributo* (*attribute timestamping*) ou modelo na não-primeira-forma-normal (N1NF), adiciona atributos temporais aos atributos originais, estendendo o domínio de cada atributo para valores complexos que incorporam a dimensão temporal. [Lan95] apresenta várias estratégias de implementação de bancos de dados temporais usando o modelo relacional com *marcação-de-tempo na tupla* ou *marcação-de-tempo no atributo*.

Considere, por exemplo, uma relação que registre as mudanças de departamento e salário de empregados de uma organização. Enquanto ambas as relações parecem ter o mesmo conteúdo de informação, ou seja, os mesmos dados sobre dois empregados no mesmo período de tempo, os modelos representam essas informações de formas bem diferentes. Na abordagem 1NF (Tabela 2.2a), cada momento relevante de cada empregado é representado por uma tupla separada. Na abordagem N1NF (Tabela 2.2b), o histórico de cada empregado é representado com uma única tupla, na qual as marcas-de-tempo estão embutidas como componentes dos valores de cada atributo.

Nome	Departamento	Salário	tempo
João	Vendas	20K	0
João	Finanças	20K	1
João	Finanças	27K	3
Pedro	RH	30K	1
Pedro	RH	29K	2
Pedro	RH	32K	5

Tab.2.2a - Modelo de relação 1NF

Nome	Departamento	Salário	tempo
João	0 → Vendas	0 → 20K	{0-3}
	1 → Finanças	3 → 27K	
Pedro	0 → RH	1 → 30K	{1-5}
		2 → 29K	
		5 → 32K	

Tab.2.2b - Modelo de relação N1NF

Nos modelos N1NF, todas as informações sobre uma entidade do mundo real são armazenadas em uma única tupla. Nos modelos 1NF essa propriedade não se aplica. Ou seja, ao contrário dos modelos 1NF, os modelos N1NF aninham dados relacionados de uma mesma entidade em uma única tupla.

O termo N1NF é usado para referenciar vários outros tipos de relaxação da propriedade 1NF, incluindo modelos que permitem relações aninhadas (*nested relations*) e atributos multi-valorados. Portanto os termos *temporalmente-aninhado* e *temporalmente-desaninhado* normalmente são usados ao invés de N1NF e 1NF respectivamente.

2.1.10 Granularidade Temporal

A granularidade temporal de um dado é o nível de abstração temporal em que esse dado é definido. Em outras palavras, esta propriedade define a precisão com que o valor de tempo é representado. A granularidade simples é expressa por um componente (por exemplo, “segundo”). A granularidade composta é expressa por vários componentes (por exemplo, “dia/mês/ano”).

O suporte a múltiplas granularidades temporais é uma das funcionalidades básicas que os SGBD temporais devem prover ao usuário. Além disso, as operações de conversão entre as diferentes granularidades também devem ser fornecidas. A integração de diferentes bancos de dados (ou relações) definidos em granularidades diferentes [PSE+94] é feita através da conversão de granularidade.

As operações temporais efetuadas em uma consulta temporal devem utilizar a granularidade mais fina possível para obter máxima precisão. Entretanto o usuário pode preferir que as respostas sejam apresentadas em uma granularidade temporal mais grossa para aumentar a legibilidade. Em um SIG por exemplo, os dados de produção agrícola podem ser armazenados em valores mensais, mas podem ser apresentados em valores anuais. Isto se assemelha a uma mudança de escala num mapa cartográfico.

Os resultados de operações temporais entre entidades de diferentes granularidades temporais podem gerar incertezas e imprecisões. Neste caso, se a granularidade exigida pelo usuário for mais fina do que a maior granularidade dentre as entidades participantes da operação, a resposta deve ser aproximada, ou então a resposta deve fornecer um intervalo correspondente a uma margem de incerteza. Por exemplo, a diferença entre os tempos-de-vida de duas entidades, onde o tempo-de-vida da primeira entidade é 47 horas e o tempo-de-vida da segunda é 8 horas, seria 39 horas. Mas se a primeira entidade usa uma granularidade de dias, o sistema armazenaria o seu tempo de vida como 1 dia (24h). A diferença entre o tempo-de-vida das entidades seria então de 16 horas.

2.1.11 Tempo-Válido Relativo/Absoluto

O tempo-válido absoluto define a posição de uma entidade no eixo temporal em uma dada granularidade temporal. Essa posição não depende nem do tempo-válido de outra entidade e nem de variáveis de tempo como a variável especial *now*. Por exemplo, “O salário de Maria foi aumentado em 30 de março de 1993”, ou “Pedro foi assassinado em outubro de 1994”.

O tempo-válido relativo tem a sua posição relativa a alguma entidade, ou a variáveis de tempo como a variável especial *now*. O relacionamento temporal entre duas entidades pode ser qualitativo (por exemplo, antes/depois) ou quantitativo (por exemplo, 3 dias antes ou 4 anos depois). Alguns exemplos de tempo-válido relativo são: “O salário de Maria foi

aumentado ontem”, “O salário de Pedro foi aumentado um mês após o aumento de salário de Maria”. Observe que os tempos relativos envolvem tanto os fatos definidos cronologicamente quanto os fatos indefinidos cronologicamente [JCE+94].

Em aplicações de planejamento e programação de horários a utilização do tempo-válido relativo, que evidencia os relacionamentos temporais entre os eventos, é mais comum do que a utilização do tempo-válido absoluto, que fixa a posição desses eventos no eixo temporal. As **váriaveis de tempo** são usadas como pontos de referência móveis ao longo do eixo temporal para facilitar a definição de uma ordem temporal entre eventos relacionados [PSE+94]. Por exemplo, “A plantação das sementes deve ser feita 3 dias após a primeira chuva do mês”. Neste caso, a primeira chuva do mês é uma variável de tempo usada como referência que só recebe valor após acontecer a primeira chuva. As variáveis de tempo periódicas são necessárias para representar eventos recorrentes. Por exemplo, um fato pode ocorrer “toda segunda” ou “a cada dia 11 do mês”.

2.1.12 Indeterminância e Modalidade Temporal

Uma informação é temporalmente indeterminada quando não tem um valor de tempo-válido preciso. Essas informações temporalmente incertas podem ser caracterizadas com a expressão “não sei quando”, ou mais precisamente, “não sei exatamente quando”. Por exemplo, um fato pode ter ocorrido “entre 14:00 e 16:00 hs”, ou “na segunda ou terça”, “algum dia na semana passada”, ou “por volta do meio do mês” [JCE+94].

A **modalidade temporal** é a maneira que um fato, associado originalmente com um *chronon* ou intervalo numa dada granularidade, se distribui nos *chronons* correspondentes numa granularidade mais fina ou dentro do intervalo no mesmo nível de granularidade. Existem basicamente dois tipos de modalidades temporais:

- A modalidade temporal *algumas-vezes* (*sometimes*) - quando sabe-se que o fato aconteceu em pelo menos um dos *chronons* correspondentes numa granularidade mais fina, ou em pelo menos um dos *chronons* do intervalo, se um intervalo for dado. Os fatos temporalmente indeterminados têm modalidade temporal *algumas-vezes*, pois não se sabe precisamente em qual *chronon* este fato ocorreu. Por exemplo, “A luz estava acesa ontem”, significando que ela estava ligada por pelo menos 1 *chronon* ontem.
- A modalidade temporal *sempre* (*always*) - quando sabe-se que o fato aconteceu em todos os *chronons* correspondentes numa granularidade mais fina, ou em todos os *chronons* do intervalo, se um intervalo for dado. Por exemplo, “A loja ficou aberta durante todo o domingo”.

Note que um fato temporalmente indeterminado não implica que o fato tenha valor incerto. A incerteza de valor existe quando não sabemos se o fato ocorreu realmente. Por exemplo, devido a um distúrbio ecológico em um lago, supõe-se que houve contaminação

do lago no passado. Como não se pode ter certeza de que houve contaminação até que a análise da água seja feita, o valor que representa este fato no banco de dados é incerto [Her94].

2.1.13 Expressão Temporal

Uma expressão temporal é uma construção sintática usada numa consulta que calcula um valor temporal, isto é, um instante, um intervalo de tempo, ou um elemento temporal. Essas expressões podem ser usadas em conjunto com as expressões relacionais e expressões booleanas em construções mutuamente recursivas. Algumas considerações importantes são:

- Um elemento temporal constante é uma expressão temporal (por exemplo, $[0, now]$).
- Se A é um atributo, então $[[A]]$ é uma expressão temporal que extrai o valor temporal de A . Por exemplo, se o valor de SALARIO é $\langle [11, 49] 15K, [50, 54] 20K, [55, 60] 25K \rangle$, então $[[SALARIO]] = [11, 60]$.
- Se A e B são atributos e θ é um operador (como \leq ou $>$), então $[[A\theta B]]$ é uma expressão temporal. Esta expressão extrai o domínio temporal quando os valores dos atributos A e B tinham o relacionamento θ .
- Se c é uma constante temporal, então $[[A\theta c]]$ e $[[c\theta B]]$ também são expressões temporais. Por exemplo, $[[SALARIO \neq 20K]]$ é $[11, 49] \cup [55, 60]$.

2.1.14 Interpolação Temporal

A derivação do valor de um atributo do banco de dados num chronon para o qual este valor não foi explicitamente armazenado é chamado de interpolação temporal. Essa derivação é tipicamente expressa como uma função calculada a partir dos valores temporalmente precedentes e/ou sucessores no histórico desse atributo. Este conceito é importante para históricos longos (em particular, dados científicos contínuos) onde os dados são coletados por amostragem, ou quando todos os dados são coletados, e a interpolação é usada como uma forma de compressão.

2.2 Linguagens de Consulta Temporal

O projeto de uma linguagem de consulta temporal associada com um modelo temporal deve levar em consideração os aspectos relevantes ao projeto de uma linguagem convencional: facilidade de uso, expressividade e implementabilidade [PSE+94].

A maioria das propostas encontradas na literatura para linguagens de consulta temporal estenderam uma linguagem de consulta atemporal definindo cláusulas específicas para descrever o predicado temporal. Estas cláusulas contêm operadores temporais que retornam valores booleanos (comparadores) e/ou que retornam valores de tempo.

Os principais operadores temporais que retornam valores de tempo são a união(\cup), a interseção(\cap), a diferença ($-$) e a negação (\neg). Por exemplo, $[1990,1993] \cap [1992,1995]$ resulta em $[1992,1993]$. A semântica dos principais comparadores temporais é definida a seguir, onde as diferenças são medidas em *chronon*.

Sintaxe do Operador Temporal		Condição se Verdadeiro
$[a,b]$	BEFORE $[c,d]$	$b < c$
$[a,b]$	AFTER $[c,d]$	$a > d$
$[a,b]$	DURING $[c,d]$	$(a \geq c) \text{ and } (b \leq d)$
$[a,b]$	EQUIVALENT $[c,d]$	$(a = c) \text{ and } (b = d)$
$[a,b]$	ADJACENT $[c,d]$	$(c - b = 1) \text{ or } (a - d = 1)$
$[a,b]$	OVERLAP $[c,d]$	$(a \leq d) \text{ and } (c \leq b)$
$[a,b]$	FOLLOWS $[c,d]$	$(a - d = 1)$
$[a,b]$	PRECEDES $[c,d]$	$(c - b = 1)$

- Os valores de tempo $[a,b]$ e $[c,d]$ são intervalos de tempo. Entretanto pontos de tempo e elementos temporais também poderiam ser comparados.
- Um ponto de tempo (*chronon*) pode ser representado por um intervalo degenerado ($a=b$ ou $c=d$).
- Se f e g são comparadores temporais, então $f \wedge g$, $f \vee g$ e $\neg f$ também são.
- Se u e v são comparadores temporais, então $u \subseteq v$ é uma expressão booleana.

Dentre as linguagens de consulta propostas para o modelo relacional, a linguagem *Tquel* é a mais conhecida [TCG+95]. Esta linguagem, apresentada em [Sno92], é uma extensão mínima da linguagem Quel do SGBD relacional Ingres para incorporar a álgebra temporal. TQuel suporta as dimensões temporais tempo-válido, tempo-de-transação, e como a maioria das linguagens de consulta convencionais, também suporta o tempo-

definido-pelo-usuário. Dentre as novas cláusulas acrescentadas à linguagem Quel, as mais importantes são:

- **as of** - procura no eixo de Tempo-de-Transação o estado cujo tempo-de-transação é o especificado pela cláusula. O predicado atemporal da consulta é então aplicado nesse estado.

Exemplo: "Qual era a população das cidades de São Paulo se a pesquisa fosse feita em 1980 ?"

retrieve (C.Nome, C.População) where C.Estado = "São Paulo " as of Janeiro 1, 1980

- **when** - procura no estado atual aquelas tuplas nas quais pelo menos um dos seus intervalos de Tempo-Válido intercepta o tempo-válido especificado pela cláusula, restringindo a parte atemporal da consulta a essas tuplas do banco de dados.

Exemplo: "Qual era a população das cidades de São Paulo em 1980 ?"

retrieve (C.Nome, C.População) where C.Estado = "São Paulo " when C overlap Janeiro 1, 1980

Outras cláusulas da linguagem TQuel incluem os conceitos de agregados, indeterminação temporal e evolução de esquema [Sno92].

2.3 O Modelo TOODM (Temporal OO Data Model)

Muitas extensões aos modelos relacionais para incorporar o tempo foram propostos nos últimos 15 anos. A maioria suporta somente a dimensão de tempo-válido. Alguns poucos somente suportam o tempo de transação ou suportam os dois tipos de tempo. Esses modelos podem ser comparados em relação à dimensão de tempo-válido com três questões fundamentais:

"Como o tempo válido é representado ?",

"Como os fatos são associados com o tempo válido ?"

e "Como são representados os valores de atributos ?" [Sno95].

Entretanto, os modelos OO têm demonstrado serem mais aptos para armazenar dados temporais, pois facilitam o aninhamento temporal dos dados.

O modelo TOODM de [OM93] incorpora as dimensões tempo-válido e tempo-de-transação ao modelo orientado a objetos. Além disso, este modelo estendeu os modelos temporais anteriores para suportar a evolução temporal de todas as propriedades dos objetos e do esquema.

O modelo TOODM possibilita a existência de três tipos de objetos: objetos invariantes no tempo, objetos variantes no tempo e objetos do tipo TIME, usados para representar valores de tempo. O eixo de tempo é linear e discreto. Uma classe pode incorporar até duas dimensões temporais e deve ter pelo menos as mesmas dimensões temporais que as suas superclasses. Para consultar o TOODM, [OM93] propôs a linguagem de consulta TOOL (Temporal Object Oriented Language).

Segundo [OM93], o resultado de uma consulta temporal pode ser: valores de atributos atemporais (por exemplo, número real), objetos atemporais (ou objetos invariantes no tempo), valores de tempo (por exemplo, [1992,1993]), e objetos variantes no tempo.

2.3.1 Consultas que Retornam Intervalos de Tempo

Para fazer as consultas que retornam valores de tempo foram criadas duas cláusulas: **TWHEN** e **VWHEN**.

A cláusula **TWHEN** retorna o conjunto de tempos de transação em que o dado selecionado pela consulta foi armazenado ou atualizado. Se a cláusula **VALID** for especificada, a consulta primeiro restringe os objetos selecionados pelo predicado ao intervalo de tempo-válido dado. A sintaxe da consulta é a seguinte:

TWHEN (predicado atemporal)
FROM (repositório de classe)
VALID (expressão temporal)

O resultado dessa consulta é obtido através da execução dos seguintes passos:

1. Obtém os intervalos de tempo-válido que satisfazem a cláusula **VALID**.
2. Encontra os objetos que satisfazem ao predicado atemporal.
3. Retorna os tempos-de-transação nos quais os objetos encontrados em (2) eram válidos no tempo determinado em (1).

Por exemplo: “**TWHEN** predicado_X **FROM** repositório **VALID** [[Y.age=25]]” primeiro determina em qual tempo-válido Y tinha 25 anos, depois retorna os tempos-de-transação que satisfazem o predicado X nesse período.

A cláusula **VWHEN** retorna os intervalos de tempo-válido do dado selecionado pela consulta. Se a cláusula **INDB** for especificada, a consulta primeiro restringe os objetos

selecionados pelo predicado ao intervalo de tempo-de-transação dado. A sintaxe da consulta é a seguinte:

```
VWHEN (predicado atemporal)
FROM (repositório de classe)
INDB (expressão temporal)
```

O resultado dessa consulta é obtido através da execução dos seguintes passos:

1. Obtém os estados com os tempos-de-transação especificados a cláusula INDB.
2. Encontra os objetos que satisfazem ao predicado atemporal.
3. Retorna os tempos-válido dos objetos selecionados em (2) e armazenados nos estados determinados em (1).

Por exemplo, "VWHEN predicado_Z FROM repositório INDB [[1/11/95]]" primeiro recupera o estado dos dados armazenados no tempo tempo-de-transação 1/11/95, e então retorna os tempos-válido das tuplas que satisfazem o predicado Z e que foram armazenadas nesse estado.

2.3.2 Consultas que Retornam Objetos Temporais

As consultas que retornam objetos temporais são chamadas de consultas de fatiamento temporal (*timeslice*). Essas consultas retornam os estados (ou fatias) de um banco de dados que satisfazem certas condições durante um período de tempo-válido ou tempo-de-transação. Para esse tipo de consulta foram criadas duas cláusulas temporais: TSLICE e VSLICE. A sintaxe desse tipo de consulta é a seguinte:

```
SELECT (predicado atemporal)
FROM (repositório de classe)
TSLICE (expressão temporal)
VSLICE (expressão temporal)
```

O resultado dessa consulta é obtido através da execução dos seguintes passos:

1. Restringe o domínio da consulta para os estados do banco de dados correspondentes aos intervalos de tempo especificados em TSLICE e/ou VSLICE e seleciona os objetos ativos nesse domínio.
2. Executa a parte atemporal da consulta nos objetos determinados em (1).

Por exemplo, “SELECT f.salario FROM f in FUNCIONARIO WHERE f.nome==Jose VSLICE [[f.cargo==gerente]]” identifica quando o funcionário de nome José foi gerente, e retorna o salário desse funcionário nesses períodos.

2.4 A Camada de Gerenciamento Temporal

A camada de gerenciamento temporal (CGT) desenvolvida por Brayner adota o modelo TODM incorporando o conceito de tempo no banco de dados orientado a objetos (BDOO) O₂ [Bra94]. Através de uma interface de menus, a camada permite adicionar ao esquema de uma classe as estruturas necessárias para armazenar as marcas-de-tempo dos seus atributos. A CGT garante a consistência temporal entre as classes do banco de dados através das **restrições de integridade temporal**, que impedem que uma classe temporal tenha menos dimensões temporais que sua classe antecessora.

O módulo de consultas da CGT faz um mapeamento das operações temporais definidas no modelo temporal em comandos pertencentes à linguagem de consulta nativa do O₂. Os comandos mapeados interagem com dois componentes da arquitetura do O₂: o gerenciador de esquema e o gerenciador de objetos, e podem ser classificados em três conjuntos distintos:

- Comandos de manipulação de esquema - têm a função de definir esquemas, bases, classes, aplicações, programas, funções e métodos.
- Instruções imperativas - são utilizadas dentro de um corpo de programa de O₂C¹⁰ para alterar objetos e valores.
- Comandos de Consulta - têm a função de recuperar objetos e valores do BD.

Entre as características temporais adotadas na implementação do modelo, podemos destacar:

- O Tempo é *linear* e *discreto* nos dois eixos temporais.
- O eixo Tempo-Válido foi definido no formato de *intervalos*.
- O eixo Tempo-de-Transação foi definido no formato *eventos* e tem como origem o instante de criação da base de dados.
- A granularidade é fixa em “segundo” no eixo de Tempo-de-Transação.

¹⁰ Esta é a linguagem de programação nativa do banco de dados O₂.

- A granularidade é composta e variável no eixo de Tempo-Válido.

A CGT é baseada em estruturas que utilizam os construtores de tipos do O_2 que viabilizam a incorporação do contexto temporal em objetos.

Para o nível mais externo de um objeto complexo é incorporada apenas uma dimensão temporal, conforme o construtor utilizado. Se o objeto tiver o construtor *tuple* no seu nível mais externo, então a dimensão Tempo-Válido é incorporada ao objeto com a adição do atributo <histórico_objeto>. Se o objeto tiver o construtor *set* ou *list* no seu nível mais externo, então a dimensão Tempo-de-Transação é incorporada ao objeto com a adição do atributo <tt_objeto>.

Os objetos podem ter componentes complexos. Para componentes definidos pelo construtor *set* ou pelo construtor *list* somente a dimensão Tempo-de-Transação é incorporada através da adição do atributo <tt_componente>. Para componentes definidos pelo construtor *tuple* somente a dimensão Tempo-Válido é incorporada através da adição do atributo <histórico_componente>. A regra prossegue recursivamente nos componentes do objeto até ser encontrado um componente não composto, ou seja, um tipo primitivo do O_2 (integer, real, char, string ou boolean) ou uma referência a um objeto de uma classe (OID^{11}).

Neste caso, as duas dimensões temporais (tempo-válido e tempo-de-transação) são incorporadas ao componente pela adição dos atributos <tv_início_componente>, <tv_fim_componente> e <tt_componente>.

Por exemplo, suponha que exista uma classe Pessoa com o seguinte esquema:

```
class Pessoa
    type tuple (   nome : string ,
                  ender: string ,
                  data-nascimento : Date ,
                  cpf: integer ,
                  filhos: list ( Pessoa ) ,
                  );
```

Se incorporássemos os dois eixos temporais nessa classe utilizando a CGT, a classe Pessoa bitemporal ficaria da seguinte forma (tv é tempo-válido, tt é tempo-de-transação):

```
class Pessoa
type tuple (   histórico_objeto: list ( tuple ( início_tv: real , fim_tv: real ) ) ,
```

¹¹ Identificador do objeto.

```
nome: list ( tuple ( tt_nome: real;
                    histórico_nome: list ( tuple ( valor_nome: string ,
                                                    início_tv_nome: real ,
                                                    fim_tv_nome: real ) ) ) ) ,
ender: list ( tuple ( tt_ender: real;
                    histórico_ender: list ( tuple ( valor_ender: string ,
                                                    início_tv_ender: real ,
                                                    fim_tv_ender: real ) ) ) ) ,
data-nascimento : Date ,
cpf: integer ,
filhos: list ( tuple ( valor_filhos: set (
                    list ( tuple ( tt_Pessoa: real ,
                                    histórico_Pessoa: list ( tuple ( valor_Pessoa: Pessoa ,
                                                                    início_tv_Pessoa: real ,
                                                                    fim_tv_Pessoa: real ) ) ) ) ,
                                )
                    tt_filhos: real ) )
)
```

Os componentes *cpf* e *data-de-nascimento* são atributos invariantes no tempo. O atributo *data-de-nascimento* é um atributo temporal do tipo tempo-definido-pelo-usuário.

A marcação-de-tempo no atributo ao invés da marcação-de-tempo no objeto (ou marcação-de-tempo na tupla do modelo relacional) diminui a redundância de dados, evitando a replicação de todos os atributos de um objeto a cada alteração de um de seus atributos. A replicação de objetos traz o problema de múltiplos OID para um mesmo objeto do mundo real.

O módulo de consultas da CGT implementou algumas das cláusulas temporais definidas pela linguagem de consulta TOOL de [Oli93]: VSlice, TSlice, VWhen, TWhen. Para recuperar um estado de um objeto, procura-se os valores de atributos desse objeto que tenham um tempo-de-transação igual ou menor que o tempo-de-transação especificado.

Os comparadores temporais implementados foram: Before, After, Before_Overlap e After_Overlap. Os operadores temporais implementados que retornam valores de tempo foram: First_Instant, Last_Instant, Max_Interval, Min_Interval e All_Interval.

2.5 Resumo

Este capítulo apresenta uma breve revisão de alguns conceitos relevantes ao entendimento da dissertação, especialmente no que tange a linguagem de consulta a ser utilizada como base.

Capítulo 3

Modelagem Temporal de Dados Geográficos

3.1 Introdução

Um modelo é uma representação formal limitada da realidade, restringido pela natureza discreta dos mecanismos de manipulação e análise. A dificuldade de definir um modelo útil e compreensível reside em encontrar um sistema formal, cujo comportamento imita o melhor possível o funcionamento do mundo real. A modelagem de dados espaciais pode ser definida como o processo de discretizar a variação espacial das entidades do mundo real [FG90].

Um modelo de dados deve descrever as operações lógicas e definir seus resultados, e a estrutura de dados deve mostrar como eles podem ser armazenados e otimizar as operações para aumentar o desempenho. Uma estrutura de dados pode prover uma excelente representação física para vários modelos de dados, assim como um modelo de dados pode ser materializado em várias estruturas de dados diferentes. Na área de sistemas geográficos, o termo "modelo" é confundido com estrutura de dados e influenciado por programas já existentes, enquanto deveria procurar ser fiel ao máximo na sua representação [Fra90]. Em sistemas geográficos, a modelagem dos dados é tipicamente guiada pelos formatos de dados tradicionalmente empregados para representar os fenômenos geográficos [CCM+94]. O formato *raster* é originário de imagens e é caracterizado por uma matriz de células de tamanhos regulares, onde para cada célula é associado um conjunto de valores representando as características geográficas da região correspondente. O formato *vetorial* descreve dados espaciais com uma combinação de formas geométricas (pontos, linhas e polígonos).

Existem duas interpretações básicas do espaço: visão de *campo* e visão de *objetos* [Fra90]. Na visão de campos as entidades geográficas são definidas por valores que variam ao longo de todo o plano. Em outras palavras, cada região do plano tem um valor para cada característica geográfica que estiver sendo modelada. Na visão de objetos, o mundo está repleto de objetos identificáveis e cada posição (X,Y) no espaço está ou vazia ou ocupada por uma ou mais entidades geográficas. Cada entidade tem um conjunto de atributos associados que descreve as suas características geográficas e a diferencia das outras entidades.

Um modelo adequado deve comportar tanto a visão de objetos quanto a de campos. [Peu94] propõe um modelo híbrido contendo as duas visões de espaço. Nesse modelo haveria uma representação raster do espaço, onde cada célula teria um atributo apontando para todas as entidades geográficas que se localizassem sobre ela. Propõe igualmente uma estrutura de objetos em que há atributos apontando para células na representação raster para definir a sua posição geográfica no espaço, uma representação vetorial para descrever a forma de seu componente espacial e os relacionamentos espaciais com as outras entidades. Segundo a autora, o modelo uniria as duas visões que passariam a ser cooperativas e interdependentes.

A questão de quais dados armazenar é também crucial e depende do domínio da aplicação e do perfil do usuário. Não raro, usuários precisam se contentar com os dados disponíveis, em função do custo de coleta, processamento e armazenamento de dados espaciais. [Peu94] ressaltava a importância de sistemas geográficos armazenarem informação contextual, chamada por ela de "*conhecimento de alto nível*", que é derivado a partir dos dados "*observados*". Segundo a autora, a adição desse tipo de informação transforma o "*modelo histórico do mundo*" no que pode ser chamado de "*modelo de processos*", tornando possível tratar consultas explanatórias de causa-consequência e de planejamento. Ao se adotar um modelo orientado a objetos, este conhecimento ficaria incorporado na estrutura dos objetos geográficos sob a forma de métodos em alguns objetos especiais chamados de *eventos*, que seriam os agentes de mudança, e em outros objetos especiais chamados de *regras*, que teriam a função de garantir a consistência do modelo com a realidade.

3.2 A Evolução do Armazenamento de Dados em SIG

Os Sistemas Gerenciadores de Bancos de Dados (SGBD) têm como grande vantagem a noção de independência de dados, que permite às aplicações se absterem do armazenamento de seus dados e se concentrarem em sua atividade principal. Em sistemas geográficos é cada vez mais comum a arquitetura em várias camadas de software e a interoperabilidade dessas camadas através de protocolos padrão de comunicação. Assim, a preocupação passou a ser encontrar uma combinação de camadas ideal para cada aplicação

e para cada tipo de usuário. Essa tendência pode ser verificada analisando-se a evolução da arquitetura dos sistemas de informação geográfica.

Os primeiros programas a manipular dados geográficos começaram a aparecer no fim da década de 50. Eles utilizavam imagens digitais no formato *raster* provenientes principalmente de radares e *scanners*. A partir de 1972, com o lançamento satélite americano *Landsat 1*, esses programas começaram a fazer o processamento e análise de imagens de satélite e sensoriamento remoto. Eles executavam análises muito específicas e devido ao seu alto custo, eram limitados a grandes centros de pesquisa. Entretanto, na década de 70 eles foram gradativamente sendo acoplados como módulos independentes em sistemas geográficos de uso mais geral. O administrador do sistema era o responsável pelo gerenciamento dos dados armazenados em arquivos comuns, que ficavam cada vez maiores e mais numerosos [ABCK91].

[Arg95] classifica os SIG de acordo com o histórico de seu desenvolvimento em geral. Segundo o autor, a primeira geração dos SIG são os programas que apareceram no início da década de 70, com o simples objetivo de automatizar a confecção de mapas cartográficos, evitando ter que refazê-los a cada nova alteração. Todos os componentes geográficos e até os textos descritivos nesses antigos mapas eletrônicos são formados de vetores (pontos, linhas e polígonos) e não há nenhuma informação topológica que os relacione. Nesses sistemas, todos os dados são gravados num único arquivo sequencial no formato ASCII (por exemplo: DXF) ou binário, cuja estrutura é proprietária do fabricante.

A segunda geração trouxe a possibilidade de armazenar informações descritivas de cada objeto geográfico, e relacionamentos topológicos com os objetos vizinhos. Em alguns desses sistemas, as informações não-espaciais são armazenadas em sistemas gerenciadores de bancos de dados relacionais (SGBDR), mas os dados espaciais são armazenados em arquivos do sistema operacional numa estrutura de dados exclusiva do fabricante do SIG. Um exemplo típico desse caso é o ARC/INFO [Esr92]. Conseqüentemente surgiu uma diversidade de formatos de dados espaciais que tornou praticamente inviável o compartilhamento de dados entre aplicativos de diferentes fornecedores. Estruturas de índices espaciais contendo informações topológicas foram desenvolvidas em cada sistema para aumentar o desempenho das operações espaciais. Esses mecanismos de acesso tiveram que evoluir muito para atender a necessidade de incorporar operações espaciais cada vez mais complexas.

A terceira geração transferiu o repositório de dados espaciais para dentro do SGBD relacional trazendo vantagens como: compartilhamento de dados, rapidez de acesso aos dados, alta funcionalidade, alta integridade. Apesar dos fornecedores terem levado seus bancos espaciais para uma arquitetura relacional, em alguns casos, a estrutura das relações e o modelo de dados continuou de domínio exclusivo do fabricante do sistema. Ainda era necessário a utilização de um módulo de importação e exportação de dados para que outros programas geográficos pudessem acessar os dados. Nestes casos, mesmo tendo implementado seu banco de dados em uma arquitetura aberta, o usuário estava preso a um fornecedor único, que tinha todo o controle de quais dados e para qual formato poderia

exportar. No entanto, como é muito dispendioso para o desenvolvedor manter o sistema atualizado para todas as novas versões de SGBD que são lançadas, alguns deles realmente abriram a estrutura de seus dados. Apesar dos fornecedores disponibilizarem o formato de seus dados espaciais, o desenvolvimento de programas de manipulação desses dados por parte do usuário é bastante complexo e problemático, devido às deficiências dos SGBD convencionais em lidar com dados complexos[BVH95].

As próximas gerações de SIG deverão se abster de detalhes específicos dos SGBD, mas sem abrir mão das vantagens já alcançadas com a tecnologia de bancos de dados. Para que isso seja possível, os SGBD terão que evoluir para incorporar os novos conceitos trazidos pelas aplicações modernas, como representações espaciais, multimídia, dimensões temporais e versões. Objetos complexos como os geográficos necessitam de modelos que armazenem as suas estruturas hierárquicas e seus componentes [WHM90]. As técnicas de orientação a objetos trouxeram benefícios como reutilização de código através da herança e agregação, identidade constante de objeto e polimorfismo, e mostram-se como as mais adequadas para representar essas entidades [MPB93]. Os SGBD relacionais têm sérias limitações, pois exigem que os dados estejam na primeira forma normal e por isso o acesso aos dados torna-se bastante oneroso devido às junções das relações. Os sistemas gerenciadores de bancos de dados orientados a objeto (SGBDOO) aparecem como a opção ideal para armazenar esses dados complexos.

3.3 O Modelo de Dados Proposto

3.3.1 Definição do Modelo

O modelo de dados geográficos temporal que propomos é semelhante ao modelo de [CCM+94], aqui estendido para incorporar a dimensão temporal. Apesar de alguns esforços para criar um modelo orientado a objetos padrão, ainda não existe um consenso [MM94]. Na dissertação será adotada a abordagem do modelo orientado a objeto baseada em classes do sistema O₂ [O2t93]. Resumidamente, um objeto é uma instância de uma classe e é caracterizado pela sua **estrutura**, ou forma como os seus atributos estão organizados, seu **comportamento**, ou conjunto de operações ou métodos que podem ser aplicados a ele, e seu **estado**, ou conjunto de valores atribuídos a ele. Objetos dispõem de mecanismos de herança e composição para possibilitar a reutilização de objetos previamente definidos. Objetos simples (ou elementares) podem formar objetos compostos (ou complexos) utilizando-se os construtores disponíveis no modelo (como *set*, *list*, *tuple*). A hierarquia de classes é definida no esquema, assim como os atributos e a assinatura dos métodos dessas classes. As classes antecessoras de uma classe C na hierarquia de classes são chamadas de *superclasses* de C.

O modelo de [CCM+94] é orientado a objetos e segue a filosofia de abstração de dados multi-nível, que divide a modelagem em quatro níveis distintos:

- O nível do mundo real compõe as entidades reais modeladas, como plantações, colheita, bacia hidrográfica, rios, rede telefônica, dentre outras.
- O nível conceitual introduz ferramentas para formalizar a modelagem das entidades geográficas definidas no nível anterior. Esse nível define de forma conceitual quais informações deverão ser armazenadas de cada entidade geográfica.
- O nível de representação trata dos detalhes pertinentes às possíveis representações geométricas e topológicas das propriedades espaciais das entidades geográficas. Esse nível cuida de questões como escala, precisão, projeção cartográfica e outras.
- O nível de implementação (ou físico) cobre as estruturas de dados e operações usadas nas representações, incluindo as estruturas de índices para agilizar o acesso aos dados, como as R-trees e quadrees.

[CCM+94] ressalta que na modelagem de dados convencional o nível de representação é trivial e portanto não precisa existir. Em contrapartida, dados espaciais podem ter múltiplas representações, que mudam de acordo com a escala empregada e com o tipo de análise que se deseja fazer. Uma operação espacial pode ter diferentes implementações relativas à representação que estiver sendo usada. O cálculo da distância entre dois objetos, por exemplo, pode ser feita usando a fórmula Euclideana em mapas de escala grande; ou no caso de mapas de escala pequena, com a fórmula da distância esférica, onde a curvatura terrestre deve ser levada em consideração; ou ainda com a soma do comprimento dos segmentos percorridos, como no caso de uma rede de rodovias entre duas cidades[Gat91].

3.3.2 Nível Conceitual

Seguindo [CCM+94], existem no banco de dados duas classes básicas no nível conceitual refletindo as visões de campos e objetos: **GeoCampo** e **GeoObjeto**. Iremos nos limitar na dissertação ao tratamento de objetos geográficos representados pelos geo-objetos, que são instâncias de classes de uma hierarquia cuja raiz é a classe **GeoObjeto**. Geo-objetos têm atributos não-espaciais (também chamados de atributos convencionais) e um atributo espacial (denominado **Localização**) e podem ser *elementares*, *compostos* ou *fracos*.

O espaço geográfico será aqui modelado como um conjunto de geo-objetos que se relacionam entre si, cada um com seu estado (atributos) e comportamento (métodos). As restrições de integridade (ou regras) dos objetos geográficos são formalizadas como métodos ou como outros objetos.

- O geo-objeto **elementar** corresponde a uma entidade que não é composta por outros geo-objetos. O seu estado é constituído apenas de atributos não-espaciais e de um atributo espacial. A definição de quais são os geo-objetos elementares depende do domínio geográfico que está sendo modelado. O geo-objeto "fazenda" seria um geo-objeto elementar em um sistema de controle ambiental de grandes regiões. Entretanto, se o sistema fosse controlar o cultivo de diversas plantações de uma mesma fazenda, o geo-objeto "fazenda" seria composto.

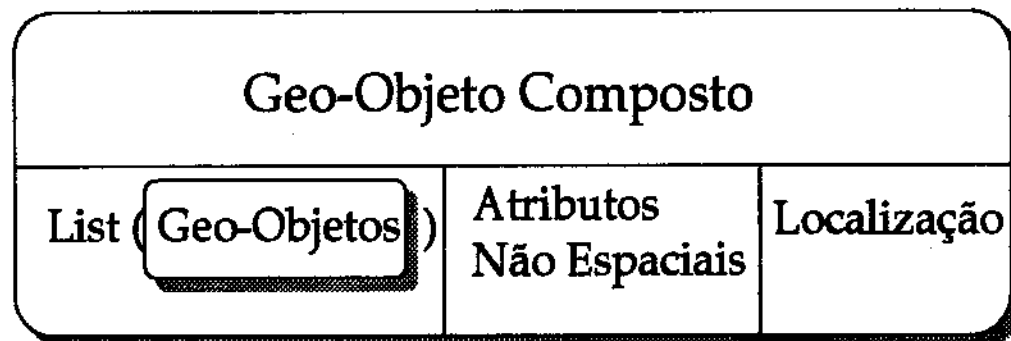


Fig 3.1 - Modelo no Nível Conceitual

- O geo-objeto **composto** é construído a partir da composição de outros geo-objetos, cujas localizações devem pertencer a uma mesma classe de referenciamento geográfico. A figura 3.1 mostra os elementos do estado de um objeto composto: ele é formado por um conjunto de atributos não-espaciais, um conjunto de geo-objetos e um atributo Localização. O geo-objeto composto "fazenda", por exemplo, é composto por uma coleção de prédios, canais de irrigação, estradas, plantações, e outros geo-objetos. Em alguns casos a Localização do geo-objeto composto pode ser computada a partir das localizações de seus componentes. A Localização do geo-objeto composto pode também delimitar uma área maior que a união das Localizações dos seus componentes. Por exemplo, o geo-objeto composto "fazenda" pode conter áreas improdutivas que não estão representadas por nenhum dos geo-objetos que a compõem.
- O geo-objeto **fraco** contém apenas o componente de Localização e existe somente enquanto faz parte de um geo-objeto composto. Por exemplo, um trecho de rua que liga uma rua a um estacionamento pode não ter tanta importância para ser considerado um objeto geográfico completo.

A definição proposta de um geo-objeto apresenta a seguinte sintaxe:

geo-objeto : tuple([atributos não-espaciais]¹, [list(geo-objeto)], Localização)

¹ O colchete indica opcionalidade.

Nosso modelo também considera a existência de objetos não geográficos, que são os objetos comuns encontrados em aplicações. Estes objetos pertencem a classes que não têm o componente Localização. Exemplos são Pessoas (que cultivam as terras da “fazenda”).

A independência de um geo-objeto de sua Localização (que é um outro objeto do banco de dados) permite que mais de um geo-objeto tenha o mesmo componente espacial. Ou seja, podem haver classes de entidades abstratas como Cooperativa agrícola e classes de entidades físicas como Fazenda que possuem uma instância com a mesma representação espacial. Dessa forma, se uma cooperativa X mudar de localização física, basta modificar seu componente espacial (por exemplo, mudando a sua associação para o componente espacial de outra instância de Fazenda).

3.3.3 Nível de Representação

Nesse nível de modelagem para a visão de objetos, a representação mais comum é a vetorial. Os modelos geográficos vetoriais propostos na literatura (por exemplo, [DRSM93], [dFM93] e [SV92]) adotam a geometria euclideana para representar o componente espacial dos objetos geográficos.

Em [CCM+94], os objetos geográficos são separados de sua representação espacial através de *hyperlinks*. Em nosso modelo a representação múltipla está restrita ao componente Localização de um geo-objeto.

Para o nível de representação, propomos um conjunto de classes que descrevem a geometria (vetorial) de um geo-objeto, cuja classe principal é denominada Geometria. Esta extensão propõe que as classes geométricas sejam providas pelo SGBD como classes primitivas. A componente Localização de um geo-objeto é definida por uma lista de objetos da classe Geometria.

Localização : list (Geometria)
Geometria : tuple ([atributos não-espaciais], list (Obj_Geom))

Cada instância da classe Geometria corresponde a uma representação espacial do geo-objeto e é composta por atributos não-espaciais para guardar informações específicas da representação (como tipo de projeção, escala, ponto de georeferência, dentre outras) e um atributo espacial para guardar objetos geométricos que definem a geometria da representação. A classe Geometria é usada para encapsular as representações vetoriais do componente espacial dos geo-objetos e serve como elo de comunicação com o nível de abstração superior.

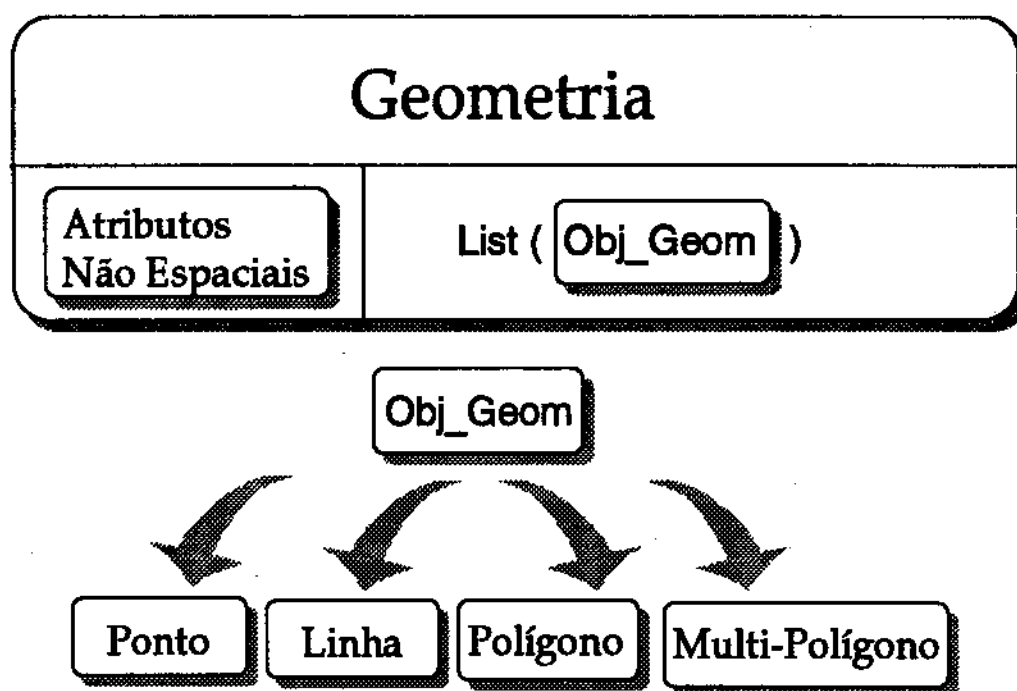


Fig 3.2 - Modelo no Nível de Representação

A classe Geometria tem como componente espacial uma lista de objetos da classe abstrata Obj_Geom que é uma generalização das classes geométricas básicas (fig3.2). O *polimorfismo*, um dos principais conceitos trazidos pelo modelo orientado a objetos, permite que uma lista de objetos de uma classe C contenha instâncias das subclasses de C [O2t93, BF94]. Assim, a lista de objetos de Geometria pode conter instâncias das subclasses de Obj_Geom que são as classes Ponto, Linha, Polígono e Multi-Polígono. Essas classes básicas estão descritas da seguinte forma:

```

MultiPolígono : tuple( list ( Polígono ), list ( Polígono ) )
Polígono      : list ( Linha )
Linha         : list ( Ponto )
Ponto         : tuple ( coord_x, coord_y )
  
```

A classe Multi-Polígono é composta por uma lista de polígonos para representar regiões desconexas, e uma lista de polígonos para representar buracos desconexos. A classe Polígono é composta por uma lista de linhas. A classe Linha é composta por uma lista de pontos.

As operações espaciais, como distância, interseção, continência, comprimento e área, podem ser definidas como métodos da classe Obj_Geom e redefinidas nas suas subclasses. O mecanismo de *late binding* garante que a versão correta da implementação dessas funções seja escolhida de acordo com a classe do objeto passado como parâmetro na consulta espacial [O2t93, BF94]. Por exemplo, ao se aplicar a operação de interseção entre

dois geo-objetos a implementação do método de interseção executada dependerá da classe geométrica que compõe a Localização desses geo-objetos.

Em resumo, qualquer área geográfica representada por um geo-objeto está delimitada por sua Localização que é uma lista de instâncias da classe Geometria compostas por objetos geométricos que têm como elementos fundamentais objetos da classe Ponto.

3.4 Processos Geográficos Temporais

Conceitualmente podemos enxergar o domínio de dados de um SIG temporal como um espaço tridimensional onde uma das dimensões é o tempo (fig 3.3). Qualquer entidade geográfica ocupa um espaço físico definido pela sua geometria e um intervalo temporal definido pela sua existência no mundo real. Uma fatia do banco de dados perpendicular ao eixo temporal e paralela ao plano X x Y do domínio espaço-temporal representa o estado do mundo modelado naquele instante de tempo. Uma entidade geográfica pode estar presente em vários estados de instantes temporais distintos.

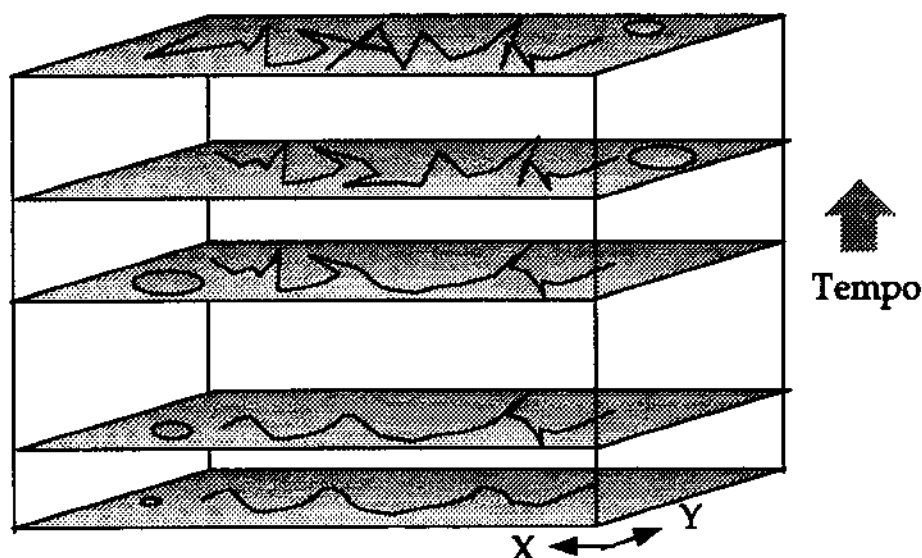


Fig 3.3 - Evolução dos Geo-objetos no Tempo

Apesar do tempo ser de natureza contínua, todos os modelos que representam o tempo usam valores discretos para popular o eixo temporal. Entre dois estados temporais consecutivos, o valor de um atributo de uma entidade geográfica pode ser considerado constante ou pode ser calculado com uma função de interpolação usando como argumentos os valores desse atributo nos dois estados temporais mais próximos e a distância temporal entre esses estados.

Um *processo geográfico* é definido como uma ação exercida ou um efeito sofrido por uma entidade geográfica, modificando-a. O processo começa quando a atividade ou o efeito começa a mudar o estado da entidade geográfica. O processo termina quando a atividade ou o efeito cessa e a entidade tem o seu estado completamente alterado. Exemplos de processos são a maturação de uma plantação, a erosão, a pulverização, o desmatamento, a queimada, a colheita, e outros. O processo geográfico é portanto, a materialização de uma mudança ou transição de estado de uma entidade geográfica.

A evolução temporal das células que discretizam um campo geográfico ou das propriedades de um objeto geográfico não é uniforme. As entidades geográficas normalmente sofrem mudanças lentas, graduais e irregulares no seu estado ao longo de sua existência. Isso se deve ao comportamento caótico característico da natureza. Por exemplo, uma região pode ter sofrido um processo de desmatamento que em poucos dias mudou o estado da sua cobertura vegetal, enquanto que as regiões não afetadas sofreram mudanças suaves ao longo dos anos. Ou seja, a velocidade com que as mudanças ocorrem pode variar significativamente entre regiões. Além disso, a velocidade de mudanças em uma mesma região também pode variar. Por exemplo, a região desmatada pode levar anos para recuperar as suas características originais.

Entre dois estados consecutivos na escala de tempo as regiões que não mudaram não precisam ter outro estado armazenado. Assim, cada região só precisa de tantas versões quantas forem as vezes que sofreu mudanças. Em muitas aplicações geográficas o usuário não precisa ter essa percepção, ou seja, supõe que cada estado é independente. Na visão de objetos, todos os geo-objetos têm um valor específico em cada instante de tempo. Nesses casos, em uma consulta temporal é possível extrair um estado temporal especificado e processar os dados desse estado como num banco de dados atemporal.

O modelo de versões de [MJ94] permite estudar a evolução temporal de dados geográficos. Se usada a visão de campo para modelar uma região, qualquer mudança de valor em algum ponto do campo gera um novo estado do campo inteiro. Analogamente se for usada a visão de objetos, qualquer alteração de valor em algum atributo do objeto gera um novo estado contendo tal objeto e todos os outros objetos presentes no estado anterior ainda que não tenham sofrido nenhuma alteração. Note que apesar da duplicação lógica de grande parte dos dados, não é obrigatória a duplicação física de dados.

Os estados temporais estão interrelacionados não só pelas versões dos objetos geográficos, mas também por *relacionamentos geográficos* entre os processos geográficos, ligando várias entidades geográficas numa rede espaço-temporal de causa-consequência. Um processo geográfico pode ter sido ocasionado por algum outro processo num estado passado, e que por sua vez poderá vir a trazer consequências futuras para outras entidades geográficas. Os relacionamentos geográficos são como linhas invisíveis que ligam processos geográficos dispersos no domínio espaço-temporal. Por exemplo, uma pulverização com agrotóxicos em uma plantação agrícola pode causar, a longo prazo, a poluição química de um rio que passa perto da fazenda. As queimadas de uma região estão relacionadas com o decréscimo de produtividade do solo nas colheitas subseqüentes e

também com o aumento de poluição do ar em regiões vizinhas. O desmatamento às margens de um rio causa a sua erosão e o seu assoreamento.

3.5 Fronteira Temporal

Assim como a fronteira espacial de regiões geográficas representadas no formato raster é nebulosa [Low94], a fronteira temporal de um processo geográfico também é nebulosa. Isso ocorre porque a maioria das entidades geográficas é formada por inúmeros componentes que mudam de estado em momentos distintos. Por exemplo, os valores de medições ambientais (como temperatura, umidade, pluviosidade, dentre outros) são amostras de processos naturais contínuos, que variam irregularmente no domínio espaço-temporal [Neu89]. Para diminuir tal problema, propomos representar a transição de estado de uma entidade geográfica como um intervalo temporal, ao invés de um instante temporal.

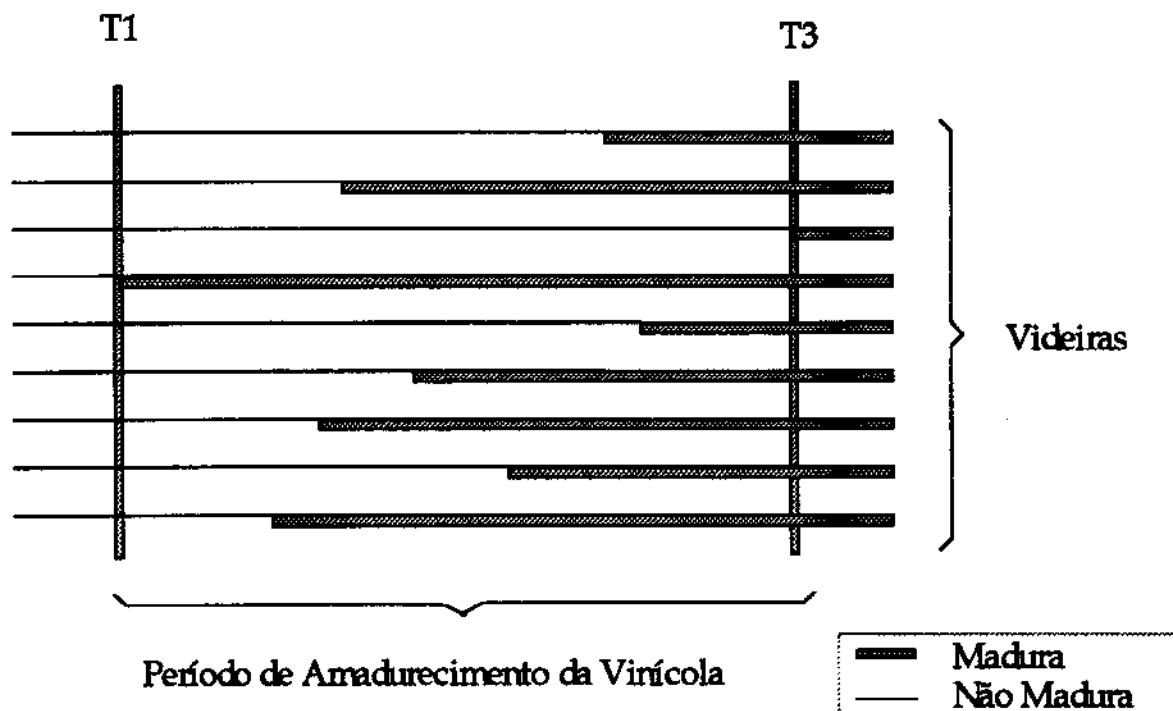


Fig 3.4 - Fronteira Temporal de um Processo

A figura 3.4 mostra uma transição de estado temático de um geo-objeto composto: uma vinícola. A transição de estado da vinícola representa o seu processo de maturação. Cada linha representa um componente do geo-objeto composto: uma videira. A largura da linha representa o valor de um atributo temático da videira: "nível de maturação". O instante de mudança de largura de uma linha representa uma transição de estado de uma

videira: o momento em que ela fica pronta para a colheita. O período T1 a T3 representa o período em que a vinícola inteira sofre o processo de maturação.

Observe que não há um instante único em que todas as videiras amadureceram, mas um intervalo de tempo (T1 a T3) em que isso ocorreu. Mas nem sempre todos os elementos de um geo-objeto composto mudam de estado, ou talvez alguns demoram exageradamente para sofrer alteração. Usando o exemplo da vinícola, é possível que algumas videiras não amadureçam. Portanto normalmente deseja-se saber apenas o período de tempo quando foi atingido um certo percentual de alteração de estado. Por exemplo, precisa-se saber quando 75% das videiras alcançaram um índice de hidratação maior que 5 a partir de um dado instante. O sistema deve retornar o período inteiro correspondente ao instante em que a primeira videira indicou esse índice (dentre as 75%) até o instante em que 75% delas alcançou esse índice. Assim, o período principal do processo, ou seja, o período de maior percentual de mudanças, é enfatizado.

3.6 Modelagem de Processos Geográficos Temporais

Um SIG temporal que se proponha a modelar processos deve focalizar principalmente as mudanças entre os estados e dar menor ênfase aos estados estáticos como visão histórica do mundo. O desafio para as ferramentas de análise espaço-temporal consiste em ajudar a compreender os processos e relacionamentos, e não simplesmente recuperar estados. Nem todos os processos geográficos relacionados se interceptam no tempo, e raramente coincidem num mesmo intervalo de tempo. Consequentemente não se pode entender os processos naturais apenas comparando-se estados instantâneos de duas épocas distintas. Corre-se o risco de ignorar fatos que ocorreram em estados intermediários e que não deixaram vestígios nos estados investigados.

Um processo geográfico pode envolver várias entidades geográficas, e representa a mudança de estado dessas entidades. Assim, um processo pode ser representado por um geo-objeto composto, onde a sua lista de geo-objetos contém as entidades geográficas participantes deste processo, e os seus atributos contém os valores (ou intervalos de valores) que caracterizam a transição de estado dessas entidades.

Por exemplo, o geo-objeto "processo de amadurecimento" é composto por todas as videiras, e os seus atributos contém os valores que caracterizam o processo de amadurecimento de uma videira. Uma videira está madura quando os valores dos atributos que indicam o seu estado de maturação, são compatíveis com os valores especificados no geo-objeto "processo de amadurecimento". Quando isto ocorre, o identificador de objeto da videira é colocado em uma lista, componente do processo. Dessa forma, é possível saber em qualquer momento o percentual já executado daquele processo, neste caso, o número de videiras já maduras.

O tempo de existência do geo-objeto "processo de amadurecimento" é o intervalo que inicia no instante de amadurecimento da primeira videira e termina no instante em que todas as videiras estão maduras. O próprio sistema poderia atualizar periodicamente o estado dos geo-objetos que fossem processos. A vantagem de se pensar em processos como geo-objetos consiste em facilitar a análise das variações espaço-temporais das entidades geográficas através da generalização das mudanças individuais de cada entidade.

Os fatores que influenciam um processo geográfico podem ser atributos temáticos de um único geo-objeto. Por exemplo, a análise de adequação do solo de uma fazenda para o cultivo de uma espécie vegetal envolve fatores como: umidade, composição orgânica, oxigenação, percentual de sais minerais, dentre outros. Cada fator tem valores próprios e impróprios para o plantio. Um solo é considerado adequado quando a medição desses fatores indica que todos alcançaram valores favoráveis para o cultivo daquele tipo de vegetal. O processo de adequação do solo é completado quando esses fatores (atributos do geo-objeto solo) atingiram um valor próprio ao cultivo da planta. O período de adequação do solo (tempo-válido do processo) equivale à fronteira temporal que separa o solo inadequado do solo adequado.

Um outro exemplo é a análise de risco de erosão de uma região, que envolve fatores como a umidade do solo, composição geológica, declividade, cobertura vegetal, entre outros. A observação do processo de crescimento de risco de erosão pode prevenir acidentes e aumentar o conhecimento sobre as suas causas e consequências através da identificação de outros processos temporalmente próximos (como por exemplo, o aumento de chuvas). Um grafo hierárquico de dependência entre esses processos geográficos pode também ser criado, onde cada arco do grafo representa um relacionamento geográfico.

Por exemplo, em um nível de abstração mais alto, o processo de maturação depende de outros processos, como irrigação, pulverização e mudanças climáticas (fig 3.5). Neste caso, os processos relacionados podem ocorrer em regiões vizinhas à região analisada. A composição, neste caso, corresponde à noção de causalidade e dependência entre fenômenos geográficos.

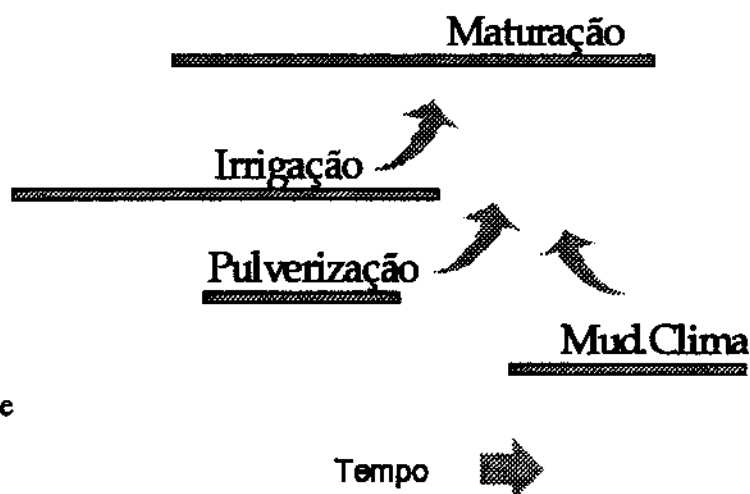


Fig 3.5 - Relacionamentos entre Processos

Um outro tipo de dependência entre processos ocorre quando a região é dividida, e o processo principal é composto pelo mesmo tipo de processo associado a cada subregião. Por exemplo, o processo de adequação do solo de uma região seria completado quando os processos de adequação das suas subregiões fossem completados.

3.7 Dimensões Temporais

Como já foi visto no capítulo 2, as dimensões temporais mais utilizadas são tempo-de-transação e tempo-válido. A primeira se refere ao tempo em que os dados são armazenados no sistema. A segunda se refere ao tempo em que os dados são considerados válidos no mundo real.

A componente espacial de um objeto geográfico pode variar nessas duas dimensões, assim como seus atributos não-espaciais. Chamaremos de *Localização bitemporal* a Localização que está associada a valores de tempo nesses dois eixos temporais. Uma Localização bitemporal de um objeto geográfico pode conter vários estados correspondentes aos tempos de transação das mudanças efetuadas. Cada estado desses pode conter diversos valores localizacionais, cada um associado a um período de validade. Cada alteração efetuada em uma Localização bitemporal de um objeto geográfico gera um novo estado dessa Localização referente ao novo tempo de transação.

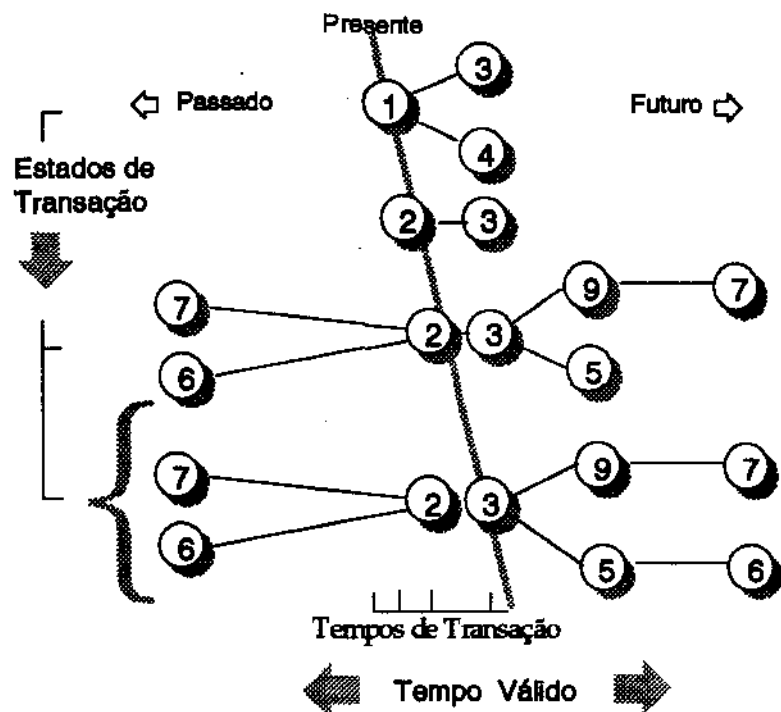


Fig 3.6 - Evolução de uma Localização Bitemporal

A figura 3.6 mostra que uma Localização pode ser válida para um tempo futuro, caso seja uma previsão. Além disso, podem existir diversas localizações para um mesmo período de validade no futuro, representando algumas possibilidades de mudança daquela informação. Um fato passado pode ser incerto, como é o caso das versões de fatos históricos das civilizações antigas, podendo haver diferentes possibilidades de localização para o passado. Não há diferentes valores para Localização no presente, pois um objeto espacial não pode estar em dois lugares ao mesmo tempo. Qualquer que seja o objeto geográfico modelado, ele tem uma única localização georeferenciada no presente. Portanto o eixo de tempo-válido é ramificado no passado e no futuro, ao contrário do eixo de tempo de transação que é linear em toda a sua extensão.

O tempo de transação não é considerado nesta tese, pois a análise da evolução dos dados nesse eixo temporal não adiciona nenhum conhecimento para o entendimento dos processos geográficos.

O tempo de validade é útil para objetos geográficos que têm suas propriedades (atributos não-espaciais ou Localização) variantes com o tempo. As imagens de satélite das condições meteorológicas de uma região são referentes ao período em que elas foram captadas. A área cultivada de uma fazenda muda conforme as condições de plantio. As rodovias têm sua geometria alterada a cada obra de expansão.

Algumas vezes, em casos de planejamento, é interessante guardar não apenas o tempo de validade de um objeto complexo como também armazenar o tempo de validade dos seus componentes relativo ao início do intervalo de validade do objeto complexo. Por exemplo, em uma construção de uma estrada podemos armazenar o tempo total da obra no formato absoluto (ex. de 05/10/1995 a 20/12/1995) conforme o contrato de execução. Os tempos parciais de construção dos trechos intermediários ao longo da estrada podem ser armazenados como eventos relativos ao início da obra ou à etapa anterior (por exemplo, 1 mês e 10 dias do início da obra).

Este recurso facilita atividades de planejamento. O usuário pode pensar primeiramente apenas em estimar a duração de eventos de maior magnitude e depois se preocupar com a distribuição de valores de tempos de suas sub-partes. Por exemplo, um agricultor calcula que terá seis meses de boas condições climáticas para plantar a sua safra. Dentro desse período ele pretende plantar soja, arroz e se possível trigo, nas diversas áreas da fazenda. Cada uma dessas plantações é executada em etapas: análise do solo, preparação de solo, plantio de sementes, irrigação, pulverização de pesticidas, dentre outras. Cada um desses processos agrícolas é aplicado em cada área com durações que variam conforme o tipo de plantação. Alguns desses processos podem ser feitos em paralelo nas diversas áreas. O resultado desse planejamento será um grafo hierárquico de dependência temporal² entre os processos (como por exemplo, um Grafo PERT de atividades). O planejamento deve atender certas restrições espaciais (por exemplo: cultura A não deve ser plantada ao lado da cultura B), e restrições temporais (por exemplo:

² É chamado de hierarquia de eventos em [Whi93].

cultura A não deve ser plantada após a cultura B). O valor de tempo-válido das etapas pode ser obtido indiretamente a partir dos tempos relativos de cada etapa. No entanto, à medida que estados previstos se tornem estados presentes, o valor de tempo-válido deve ser modificado para agilizar consultas.

3.8 Evolução Temporal dos Geo-objetos

A evolução temporal de geo-objetos será considerada apenas segundo o eixo de tempo-válido. A evolução da estrutura e do comportamento de um geo-objeto refere-se ao problema de evolução de esquemas, que não será abordado na dissertação. Assim sendo, iremos nos restringir ao problema da evolução de estado de um objeto geográfico, ou seja: os dados não-espaciais, os dados espaciais (Localização) e o seu conjunto de geo-objetos. Existem diversas propostas para modelar a evolução temporal de dados não-espaciais estendendo um banco de dados atemporal. A evolução temporal dos geo-objetos irá se basear no modelo [Oli93, OM93] descrito no capítulo 2.

Nessa dissertação, chamamos de *temporalização* o processo de alteração do esquema de uma classe ou de parte dela para que suas instâncias se tornem capazes de armazenar a evolução temporal das entidades que elas modelam. Basicamente a temporalização se resume em anexar à classe os atributos necessários para armazenar todos os valores válidos em alguma época do objeto e dos seus componentes. Cada atributo do objeto temporal tem associado um *atributo temporal* para armazenar os seus *tempos de validade*, ou seja, uma lista de intervalos de tempo em que aquele valor é válido. O atributo temporal é descrito da seguinte forma:

atributo temporal: list (tuple (tv_início, tv_fim))

Um geo-objeto pode ter períodos não consecutivos em que ele esteja "ativado" e períodos em que ele esteja "desativado". Por exemplo, um geo-objeto "plantação" do exemplo anterior pode ser desativado durante o período de entre-safra e reativado quando voltar a época de plantio.

O processo de temporalização do geo-objeto acrescenta um atributo temporal à definição do próprio objeto:

geo-objeto temporalizado: tuple(geo-objeto, atributo temporal)

O atributo temporal é inicializado com o valor do tempo no instante de início de validade do geo-objeto. Assim, um Lote criado no instante T1 seria definido por:

Lote: tuple (geo-objeto, lista_tempo(tuple (T1, ∞)))

Se num tempo T2 o Lote desaparecer (por exemplo, se for incorporado a outro Lote, a definição passa a ser:

Lote: tuple (geo-objeto, lista_tempo(tuple (T1, T2)))

Finalmente, se passado algum tempo o Lote é reconstituído em T3, temos:

Lote: tuple (geo-objeto, lista_tempo(tuple (T1, T2), tuple (T3, ∞)))

Naturalmente essa representação supõe que geo-objeto seja um objeto com identidade definida. No entanto, os componentes do geo-objeto podem também variar no tempo. O processo de temporalização também modifica a estrutura dos seus três componentes básicos: atributos não-espaciais, a lista de geo-objetos componentes e Localização. Portanto após o processo de temporalização um geo-objeto fica descrito da seguinte forma:

geo-objeto : tuple (tuple([atributos não-espaciais temporalizados],
[lista de geo-objetos componentes temporalizada],
Localização temporalizada)
, atributo temporal)

3.8.1 Temporalização da Lista de Geo-Objetos Componentes

A temporalização da lista de geo-objetos componentes é definida por:

list(tuple (list (geo-objeto temporalizado), atributo temporal)).

A quantidade de elementos da lista de geo-objetos componentes pode variar com o tempo. Suponha que uma Fazenda teve 3 tipos de Plantações (p1, p2 e p3) durante o período T1 a T6, 2 Plantações (p1 e p2) durante o período T7 a T8 e 4 Plantações (p1, p2, p3 e p4) durante o período T9 a T20. Dessa forma, o valor da lista de geo-objetos componentes dessa Fazenda fica descrito assim:

lista_versões (tupla (lista_obj (p1, p2 e p3), lista_tempo (tupla (T1, T6))),
tupla (lista_obj (p1 e p2), lista_tempo (tupla (T7, T8))),
tupla (lista_obj (p1, p2, p3 e p4), lista_tempo (tupla (T9, T20)))
)

Qualquer geo-objeto componente pode ser composto, elementar ou fraco. A temporalização é feita recursivamente até atingir um geo-objeto elementar ou fraco. A temporalização de um geo-objeto elementar ou fraco corresponde à temporalização de seus componentes não-espaciais e da sua Localização.

O geo-objeto composto e seus geo-objetos componentes podem evoluir independentemente em conteúdo. A mudança de conteúdo de um geo-objeto componente é

processada dentro do próprio geo-objeto componente. Por exemplo, um geo-objeto "divisão_agrícola" que compõe um geo-objeto "fazenda" pode mudar sua cultura agrícola. Da mesma forma, o conteúdo do geo-objeto composto pode mudar independentemente dos seus componentes. Por exemplo, o geo-objeto "fazenda" pode ter o seu atributo "proprietário" alterado.

3.8.2 Temporalização de Atributos Não-Espaciais

Cada atributo não-espacial é temporalizado em separado a partir da adição de um atributo temporal [Bra94]:

atributo não-espacial temporalizado:

`list(tuple(atributo não-espacial, atributo temporal))`

Por exemplo, suponha que uma divisão_agrícola tenha arroz no período T1 a T5 e feijão no período T8 a T9, não tendo nenhum cultivo entre T5 e T8. O valor do atributo cultura dessa divisão_agrícola seria:

```
lista_versões ( tupla ( arroz, lista_tempo ( tupla (T1, T5) ) ),
                tupla ( feijão, lista_tempo ( tupla (T8, T9) ) )
              )
```

Caso arroz tenha sido encontrado efetivamente durante T1 a T2 e T4 a T5, temos que o valor do atributo cultura dessa divisão_agrícola seria:

```
lista_versões ( tupla ( arroz, lista_tempo ( tupla (T1, T2), tupla (T4, T5) ) ),
                tupla ( feijão, lista_tempo ( tupla (T8, T9) ) )
              )
```

Se o atributo não-espacial é composto, ou seja, construído a partir de outros atributos utilizando os construtores do O₂ (*list*, *set*, *unique set* e *tuple*), a temporalização prossegue recursivamente até que o último componente temporalizado não seja composto. Cada atributo componente é temporalizado em separado.

Se o atributo composto é $\alpha(A)$, onde α pode ser o construtor *list*, *set* ou *unique set*, o processo de temporalização é assim:

1ª etapa: `list (tuple ($\alpha (A)$, atributo temporal)`

2ª etapa: `list (tuple ($\alpha (list (tuple (A, atributo temporal)))$, atributo temporal))`

Se A é um atributo não composto, o processo pára aqui. Senão o processo continua temporalizando os seus componentes mais internos.

Suponha que o atributo composto é **tuple (A,B,C)**. Após a temporalização esse atributo é descrito assim (supondo que A,B e C não são compostos):

```
list ( tuple ( tuple ( list ( tuple ( A, atributo temporal ) ) , list ( tuple ( B, atributo temporal ) ) ,  
list ( tuple ( C, atributo temporal ) ) ), atributo temporal ) )
```

3.8.3 Temporalização da Localização

A Localização de um geo-objeto também evolui no tempo. Por exemplo, o geo-objeto "divisão_agrícola" tem a sua dimensão espacial alterada a cada ano de acordo com as condições do solo, características específicas do vegetal plantado, fatores mercadológicos, mudanças de vias de acesso às colheitas, dentre outras. As alterações de uma Localização são registradas no nível de representação armazenando-se a variação dos pontos de suas linhas de contorno.

A evolução temporal da Localização de um objeto composto é independente da evolução temporal da Localização dos geo-objetos componentes. Ou seja, qualquer alteração na Localização de geo-objetos componentes não afeta a Localização do geo-objeto composto que os contém. Por exemplo, os limites de uma divisão_agrícola podem mudar sem alterar os limites da fazenda a qual ela pertence, assim como os limites da fazenda podem mudar sem alterar os limites das suas divisões³. Além disso, a área coberta pelas plantações pode não cobrir a área total da fazenda. Por isso a Localização do geo-objeto composto é armazenada separadamente da Localização dos seus componentes.

A Localização temporalizada de um geo-objeto fica assim:

```
Localização : list ( Geometria )  
Geometria   : tuple ( [atributos não-espaciais temporalizados],  
                    list ( tuple( list ( Obj_Geom ), atributo temporal ) ) )
```

Assim, os atributos não-espaciais e a lista de objetos geométricos da representação são temporalizados. Com isso, a quantidade de objetos geométricos de uma representação espacial pode mudar sem que esses objetos tenham sido modificados. Por exemplo, uma fazenda representada por uma lista de polígonos pode ter a sua representação alterada pela inclusão de mais um polígono.

Os objetos geométricos, instâncias das subclasses de **Obj_Geom** que compõem a geometria de qualquer Localização são temporalizados da seguinte forma:

```
MultiPolígono : list ( tuple ( tuple ( list ( Polígono ), list ( Polígono ) ), atributo temporal ) )  
Polígono       : list ( tuple ( list ( Linha ), atributo temporal ) )  
Linha          : list ( tuple ( list ( Ponto ), atributo temporal ) )
```

³ Obviamente para que isso seja verdade as alterações não poderiam ser numa fronteira compartilhada pelos geo-objetos fazenda e divisão_agrícola.

Ponto : tuple (tuple (coord_x, coord_y), atributo temporal)

Assim, um multi-polígono pode ter o seu conjunto de polígonos alterado, um polígono pode ter o seu conjunto de linhas alterado e uma linha pode ter o seu conjunto de pontos alterado sem que qualquer ponto tenha sido desativado.

3.9 Exemplos de Evolução da Localização

Os exemplos a seguir ilustram a evolução da Localização de um geo-objeto mostrando como os seus componentes geométricos se modificam.

3.9.1 A Evolução de um Geo-Objeto Linear

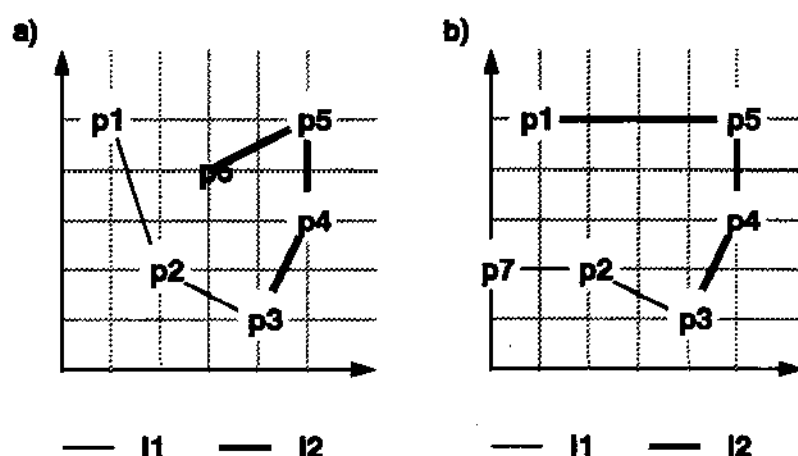


Fig 3.7 - Evolução de um Geo-objeto Linear

Suponha que duas estradas tenham num instante T1 uma representação espacial composta pelas linhas l1 e l2. A linha l1 é composta pelos pontos p1, p2 e p3. A linha l2 é composta pelos pontos p3, p4, p5 e p6 (fig 3.7a). No instante T2, as estradas sofreram modificações nas suas Localizações. Com isso, o ponto p6 foi desativado e o ponto p7 foi criado. A linha l1 passou a conter os pontos p7, p2 e p3. A linha l2 passou a conter os pontos p3, p4, p5 e p1 (fig 3.7b).

O valor das linhas temporalizadas que compõem a Localização das estradas pode ser descrito assim:

```
l1: lista_versões ( tupla ( lista_obj ( p1, p2, p3 ), lista_tempo ( tupla ( T1, T2 ) ) )
                  tupla ( lista_obj ( p7, p2, p3 ), lista_tempo ( tupla ( T2, ∞ ) ) ) )
```

l2: lista_versões (tupla (lista_obj (p3, p4, p5, p6), lista_tempo (tupla (T1, T2)))
 tupla (lista_obj (p3, p4, p5, p1), lista_tempo (tupla (T2, ∞))))

O valor dos pontos p1, p6 e p7 temporalizados pode ser descrito assim:

p1: tupla (tupla_coord (1,5), lista_tempo (tupla (T1, ∞)))

p6: tupla (tupla_coord (3,4), lista_tempo (tupla (T1, T2)))

p7: tupla (tupla_coord (0,2), lista_tempo (tupla (T2, ∞)))

3.9.2 A Evolução de um Geo-Objeto Poligonal

Suponha que uma fazenda tenha num instante T1 uma representação espacial composta por um único polígono Pg1 composto pelas linhas l1, l2 e l3 (fig 3.8a). No instante T2, a fazenda sofreu modificações na sua Localização e Pg1 teve a linha l1 desativada e incorporou uma nova linha l4 (fig 3.8b). A linha l3 é composta de apenas dois pontos: p1 e p2 (fig 3.8c). No instante T3, a fronteira é novamente modificada e a linha l3 passa a conter o ponto intermediário p3.

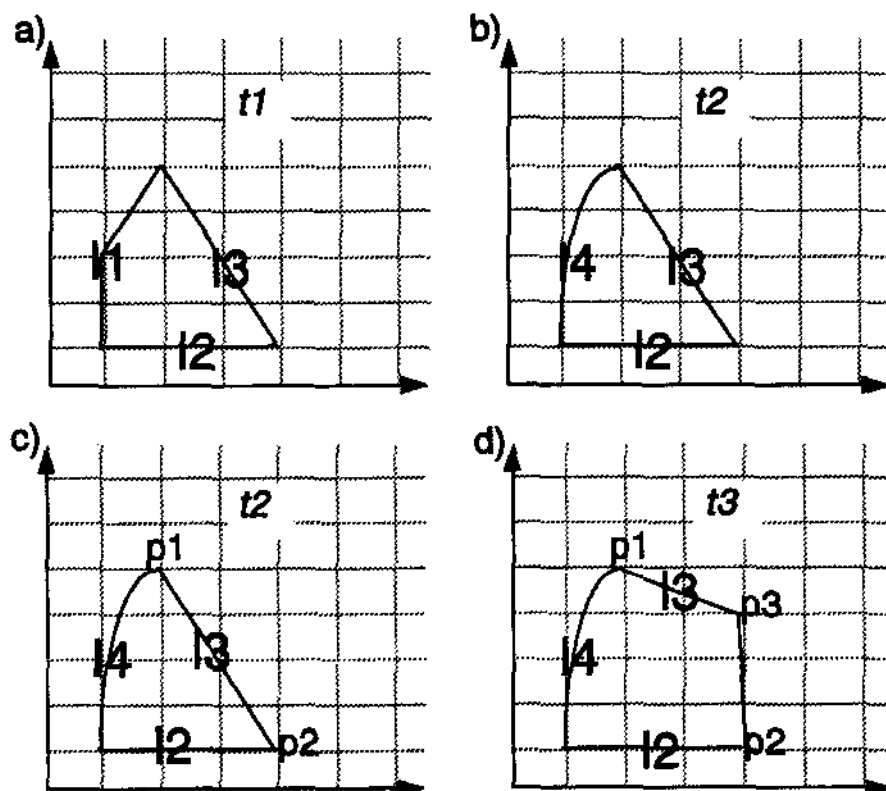


Fig 3.8 - Evolução de um Geo-objeto Poligonal

Observe que o polígono sofreu duas modificações:

1. Em T2, uma nova linha (14) foi criada substituindo a linha 11 do polígono. Neste caso, a linha 11 é compartilhada por alguma outra entidade que não foi modificada, portanto esta linha fica inalterada (fig 3.8b).
2. Em T3, a lista de pontos da linha 13 foi alterada. Nesse caso, todas as entidades que compartilham esta linha têm a sua Localização automaticamente atualizada (fig 3.8c).

O valor do polígono temporalizado que representa a Localização dessa fazenda pode ser descrito assim:

```
Pg1: lista_versões ( tupla ( lista_obj (11, 12 e 13 ), lista_tempo ( tupla (T1, T2) ) ),
                    tupla ( lista_obj ( 12, 13 e 14 ), lista_tempo ( tupla (T2, ∞) ) )
                    )
```

O valor das linhas temporalizadas que compõem o polígono Pg1 é:

```
11: lista_versões ( tupla ( lista_obj (p[11]1, ..., p[11]n), lista_tempo ( tupla (T1, ∞) ) )
                    )
```

```
12: lista_versões ( tupla ( lista_obj (p[12]1, ..., p[12]n), lista_tempo ( tupla (T1, ∞) ) )
                    )
```

```
13: lista_versões (
    tupla ( lista_obj (p1, p[12]2, ..., p[12]n-1, p2), lista_tempo ( tupla (T1, T3) ) )
    tupla ( lista_obj (p1, p[12]2, ..., p3, ..., p[12]n-1, p2), lista_tempo ( tupla (T3, ∞) ) )
    )
```

```
14: lista_versões ( tupla ( lista_obj (p[14]1, ..., p[14]n), lista_tempo ( tupla (T2, ∞) ) )
                    )
```

O valor dos pontos temporalizados que compõem a linha 13 pode ser descrito assim:

```
p1: tupla ( tupla_coord (2,5), lista_tempo ( tupla (T1, ∞) ) )
p2: tupla ( tupla_coord (4,1), lista_tempo ( tupla (T1, ∞) ) )
p3: tupla ( tupla_coord (4,4), lista_tempo ( tupla (T3, ∞) ) )
```

3.10 Resumo

Este capítulo apresentou o modelo espaço-temporal orientado a objetos que será utilizado como base no restante da dissertação.

A seguir, será desenvolvida uma taxonomia de consultas espaço-temporais em SIG, que será posteriormente exemplificada utilizando as estruturas de dados propostas no capítulo 5. Estas estruturas correspondem à materialização do modelo em um SGBDOO.

Capítulo 4

Consultas em SIG

4.1 Introdução

Não existe um consenso sobre uma definição que caracterize toda a diversidade de aplicações geográficas existentes. Muitos SIG são dirigidos para resolver problemas bem específicos. Cada vez mais são encontradas novas áreas que podem se beneficiar dessa tecnologia. No entanto, todos os SIG têm um conjunto de funções básicas em comum, como a capacidade de acessar objetos baseados na sua Localização ou atributos, calcular áreas, sobrepor mapas e construir áreas de contorno ao redor de objetos (*buffer*) [MGR91].

[Mul93] diz que existem dois tipos de aplicações em SIG: aquelas que usam SIG como uma ferramenta para gerenciar recursos para propósitos concretos, de forma a prover serviços específicos, e aquelas que apresentam SIG como um meio de aprender e analisar problemas científicos. O objetivo no segundo caso é descobrir leis gerais para explicar e prever eventos.

Este capítulo discute as classificações existentes de consultas em SIG e propõe uma nova classificação caracterizando consultas espaciais, temporais e espaço-temporais e exemplificando cada caso.

4.2 Algumas Classificações de Consultas Geográficas

Alguns trabalhos feitos na área de consultas geográficas têm procurado classificar um conjunto básico de consultas, que combinados entre si, possam representar as outras operações de análise espacial dos SIG. [Gat91] classifica as consultas em SIG em

descritivas: "Localize e mostre todas as cidades com população maior que 5000 habitantes", e analíticas: "Por que ... ?" ou "E se ... ?". [dFM93] as classifica em consultas topológicas (adjacência, borda e co-borda), de conjuntos (coincidente, elemento-de, continência e interseção) e métricas (mínimos/máximos e raios de abrangência).

Segundo [Lan95], em 1978, David Sinton publicou um artigo onde mostra que os dados geográficos têm três componentes básicos: atributo (ou tema), localização e tempo. Segundo ele, um desses componentes é fixo, outro é controlado e apenas um desses componentes é medido (veja tabela a seguir). [Lan95] demonstra que o ideal é não fixar nenhum desses três componentes, e observa que os SIG atuais ainda não têm essa capacidade. O acompanhamento de mudanças ambientais em uma região ou a análise dos dados do censo são aplicações que demandam que seja possível controlar o tempo e localização para medir os atributos, sem que haja portanto, algum componente fixo.

	Fixo	Controlado	Medido
Dados de solo	Tempo	Atributo	Localização
Mapa Topográfico	Tempo	Atributo	Localização
Mapas de papel	Tempo	Localização	Atributo
Imagem raster	Tempo	Localização	Atributo
Histórico do Clima	Localização	Tempo	Atributo
Histórico das Marés	Localização	Atributo	Tempo
Trajeto de um veículo	Atributo	Localização	Tempo

Tab. 4.1 - Consultas Geográficas

[Peu94] identifica três tipos básicos de perguntas combinando os três componentes interrelacionados da sua tríade: **onde** (visão localizacional), **o quê** (visão de atributo) e **quando** (visão temporal). Segundo a autora, se forem conhecidos os valores de dois desses componentes podemos descobrir o valor do terceiro. Mas ao contrário de Sinton, ela não fixa nenhum desses componentes. Quando deve poder representar não somente um instante de tempo, mas um intervalo de tempo ou um conjunto de intervalos de tempo. Onde deve poder representar não somente um polígono indivisível, mas também uma região composta por várias sub-regiões. O quê deve poder representar não somente uma instância, mas também um conjunto de instâncias. O quê deve ainda poder representar o estado do atributo (ou tema) de uma instância ou classe de objetos. Entretanto, a autora não esclarece que a resposta pode ser uma combinação de componentes. Veja os exemplos abaixo, de acordo com as três tipos de consultas básicas de Peuquet, a partir da figura 4.1.

- quando + onde → o quê

Resulta no objeto ou conjunto de objetos (o quê) que estão presentes em uma dada localização ou conjunto de localizações (onde) e em um dado intervalo de tempo ou conjunto de intervalos de tempo (quando). Portanto podem existir várias instâncias de o quê referentes a cada localização e em cada instante (ou intervalo) de tempo. Por exemplo, a consulta: "O quê foi plantado na fazenda X nos últimos 2 anos ?", poderia ter

como resposta: "Trigo na área 1 no ano 1 e 2, Cana na área 2 no ano 1, Trigo na área 2 no ano 2".

- quando + o quê → onde

Resulta na localização ou conjunto de localizações (onde) ocupados por um dado objeto ou conjunto de objetos (o quê) em um dado intervalo de tempo ou conjunto de intervalos de tempo (quando). Portanto existem várias localizações como resposta, onde cada uma pode ter um valor diferente para aquele atributo em cada instante (ou intervalo) de tempo. Por exemplo, a consulta: "Onde foi plantado trigo nos últimos dois anos ?", poderia ter como resposta: "Na área 1 da fazenda X no ano 1 e 2, na área 2 da fazenda X no ano 2, e na área 3 da fazenda Y no ano 2".

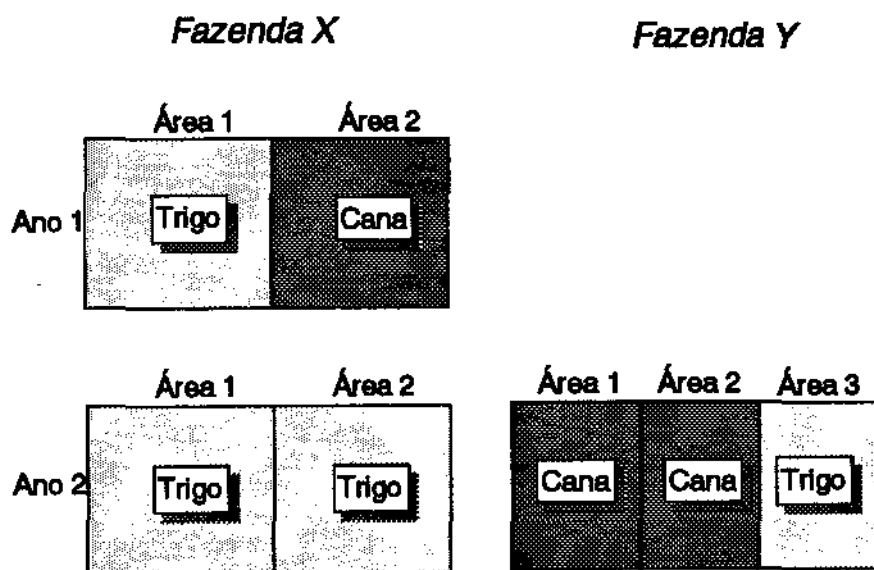


Fig. 4.1 - Um Exemplo de Consulta Geográfica

- o quê + onde → quando

Resulta no intervalo de tempo ou conjunto de intervalos de tempo (quando) em que um dado objeto ou conjunto de objetos (o quê) estavam em uma dada localização ou conjunto de localizações (onde). Portanto em cada intervalo de tempo do resultado, cada localização pode ter um valor diferente para aquele atributo. Por exemplo, a consulta: "Quando foi plantado trigo na fazenda X ?", poderia ter como resposta: "Nos anos 1 e 2 na área 1, e no ano 2 na área 2".

4.3 Classificação de Consultas Proposta

4.3.1 Definições

Pretendemos definir um conjunto de consultas geográficas espaço-temporais básicas, as quais poderemos combinar para formar as consultas mais complexas existentes na maioria dos sistemas geográficos. Para isto dividimos uma consulta em 3 componentes: predicado, domínio e resultado.

- **Predicado** é a parte interrogativa da consulta. A função do predicado é restringir o domínio da consulta e selecionar o resultado.
- **Domínio** da consulta é o conjunto de dados sobre o qual será aplicado o predicado para se extrair o resultado. O domínio da consulta pode ser um conjunto de objetos geográficos ou um esquema de uma classe geográfica e pode envolver o estado, restrições ou comportamento dessas informações. O estado de um objeto é composto por atributos convencionais, atributos espaciais, atributos temporais e relacionamentos espaço-temporais. As restrições de um objeto são compostas pelas restrições da instância, pelas restrições da classe e superclasses daquele objeto e pelas restrições implícitas dos outros objetos para com aquele. O comportamento de um objeto é determinado pelo comportamento definido em sua classe (próprio e herdado). O comportamento e os relacionamentos de um objeto são traduzidos em métodos em um modelo OO. Os métodos de relacionamento apenas retornam informações sobre um conjunto de objetos, ao contrário dos métodos de comportamento que alteram o estado dos objetos.
- **Resultado** é a resposta dada conforme a aplicação de um predicado sobre um domínio. Uma consulta pode resultar em valores de atributos convencionais, valores de atributos espaciais, valores de atributos temporais ou em um conjunto de objetos geográficos (temporais ou não).

As consultas aos esquemas das classes prevêm que o SGBD tenha disponível um nível de metaclasses acessível ao usuário. Assim este tipo de consulta é feito de forma semelhante às consultas comuns, exceto que o domínio nesse caso são as instâncias das metaclasses. Esta dissertação não irá discorrer sobre esse tipo consulta.

[DV93] e [Cif95] mostram exemplos de operações aplicáveis aos sistemas geográficos. Na dissertação, estendemos a semântica dessas operações para incorporar a dimensão temporal.

Ao longo dessa seção usaremos o conceito de mapa como sendo um conjunto de objetos geográficos interrelacionados. Exemplos de mapas são vegetação, hidrografia, solos, dentre outros.

Se um banco de dados temporal estiver disponível, qualquer operação geográfica atemporal pode ser estendida para permitir que o usuário efetue a mesma consulta sobre dados relativos a um estado passado ou futuro. Existem duas formas de se fazer essa consulta. A primeira é movendo-se tanto no eixo de tempo-de-transação quanto no eixo de tempo-válido. Por exemplo, pode-se perguntar o que a fazenda X plantava no ano passado (tempo-válido) para o estado armazenado no ano passado (tempo-de-transação). Esse caso é usado para identificar e corrigir erros de inconsistência ou alterações indevidas nos dados feitas após o último estado consistente do banco de dados. Não iremos tratar aqui desse tipo de operação temporal, uma vez que não consideramos a dimensão do tempo-de-transação no modelo proposto.

A segunda é movendo-se somente no eixo de tempo-válido, ou seja, permanecendo no estado atual do banco de dados. Por exemplo, pode-se perguntar o que a fazenda X plantava no ano passado, ou calcular a área prevista de colheita para o próximo ano. Esse caso de consulta temporal envolve apenas uma mudança para outro instante de tempo na escala de tempo-válido. De agora em diante apenas este tipo de consulta será considerado.

4.3.2 Dominância Dimensional

Seguindo a classificação de Sinton, Langran criou o conceito de *dominância dimensional* para designar a dimensão preponderante em uma aplicação geográfica [Lan95]. As aplicações com *dominância espacial* têm a maior parte de suas consultas feitas sobre os dados atuais (tempo fixo), portanto usam principalmente a dimensão espacial. As aplicações de *dominância temporal* têm a maior parte de suas consultas feitas sobre o histórico dos dados (localização fixa), portanto usam principalmente a dimensão temporal. Aplicações de *dominância espaço-temporal* são aquelas em que as consultas combinam dados de vários objetos de um mesmo estado com dados desses objetos em estados distintos. Uma consulta nessas aplicações pode percorrer a dimensão espacial e a dimensão temporal. Exemplos são: controle ambiental, modelagem de simulação, treinamentos militares, carta de navegação eletrônica, dentre outras.

Nesta tese, procuramos criar uma classificação de consultas para esse último tipo de dominância. As dominâncias espacial e temporal podem ser vistas como uma especialização da dominância espaço-temporal, portanto a nossa classificação pode ser aplicada também para aplicações que se enquadram nos outros tipos de dominância. As consultas espaciais buscam relações espaciais entre objetos em um tempo fixo. As consultas temporais varrem estados de um único objeto (ou classe) ao longo do tempo. As consultas espaço-temporais buscam relações espaciais entre objetos ao longo do tempo.

4.4 Componentes Básicos de uma Consulta Espacial Atemporal

Esta seção descreve os componentes básicos de uma consulta atemporal. Na seção 4.5, propomos a extensão para classificar consultas temporais.

As consultas em sistemas geográficos selecionam instâncias ou esquemas e projetam informações do que foi selecionado. O resultado pode ser agrupado e generalizado. Qualquer consulta envolve pelo menos um desses componentes básicos: *seleção*, *agrupamento* e *projeção*. A seleção restringe o domínio da pesquisa. O agrupamento sintetiza e generaliza o resultado. A projeção apresenta o resultado. Esses componentes de uma consulta são integrados e comumente parecem indissociáveis.

4.4.1 Seleção Espacial

A *seleção espacial* especifica quais geo-objetos de um mesmo estado temporal serão recuperados do BD a partir de um predicado. Uma seleção espacial pode conter predicados sobre atributos convencionais, espaciais, calculados (por exemplo, a soma de dois atributos numéricos) ou predicados de instância.



Os predicados filtram os geo-objetos do domínio espacial. O predicado *de instância* é aquele que especifica explicitamente quais instâncias de geo-objetos serão consultadas. O predicado é *convencional* se atua sobre um atributo convencional ou sobre um atributo calculado a partir de um atributo convencional. O predicado é *espacial* se atua sobre um atributo espacial ou sobre um atributo calculado a partir de um atributo espacial. Os predicados espaciais utilizam operações espaciais classificadas em *métricas*, *de orientação* e *topológicas*. Abaixo estão exemplificados alguns predicados classificados por tipo de operação espacial.

Operações Métricas

Exemplos de operações métricas são: distância, perímetro, comprimento e área[Cif95]. A operação de distância pode ser euclideana, esférica ou a soma de segmentos em uma rede. Em geral, ela mede a distância entre os centróides de dois geo-objetos. A operação de perímetro retorna o comprimento da fronteira de um polígono. A operação de comprimento retorna a extensão de um geo-objeto linear. A operação de área calcula a área de um geo-objeto. Os predicados espaciais métricos selecionam os geo-objetos cuja operação métrica resulta em um valor dentro de um intervalo especificado no predicado. Por exemplo, a consulta: "Selecione todas as fazendas com área < 10 ha".

Operações de Orientação

Exemplos de operações de orientação são: norte, sul, leste, oeste, esquerda, direita, abaixo e acima. Essas operações dependem de uma definição formal que contorne as ambigüidades da linguagem natural. Por exemplo, a consulta: "Selecione todas as fazendas ao norte da cidade de Campinas".

Operações Topológicas

Essas operações identificam relacionamentos topológicos entre objetos geográficos. Exemplos de operadores topológicos são: cruzamento, interseção, disjunção, adjacência, inclusão e continência [Cif95].

Essas operações são aplicadas em relação ao(s) geo-objeto(s) especificado(s) no predicado de instância, aqui chamado(s) de geo-objeto(s) fonte.

- **Interseção** - identifica os geo-objetos que intersectam o geo-objeto fonte e as regiões espaciais compartilhadas por eles. A interseção espacial também pode retornar geo-objetos fracos, ou seja, um ponto, uma linha, um polígono, um multi-polígono ou uma combinação desses elementos.

A seguir estão exemplificadas duas variações de uma seleção com predicados espaciais topológicos de interseção onde o usuário fornece um dos parâmetros espaciais da restrição.

- a) **Windowing** - Essa função aplicada sobre um mapa retorna os geo-objetos que intersectam uma janela (geralmente retangular) desenhada sobre o mapa (fig.4.2).

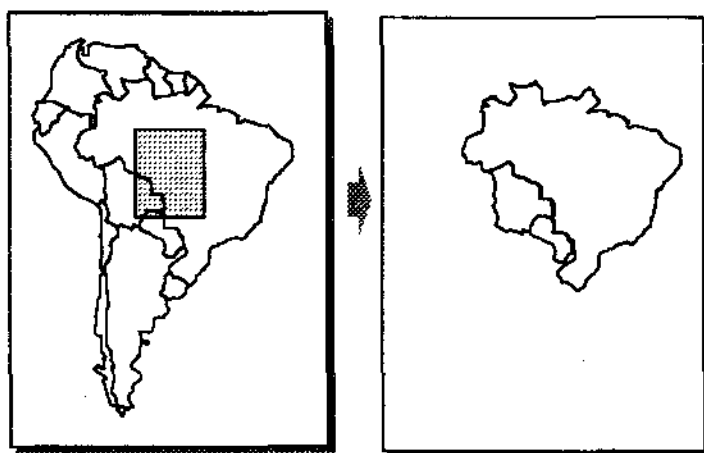


Fig. 4.2 - Windowing

- b) **Clipping** - Essa é a operação que extrai parte do mapa localizado numa dada área (fig.4.3). Ao contrário da operação anterior, o resultado é exatamente a intersecção da área retangular com a geometria dos geo-objetos do mapa.

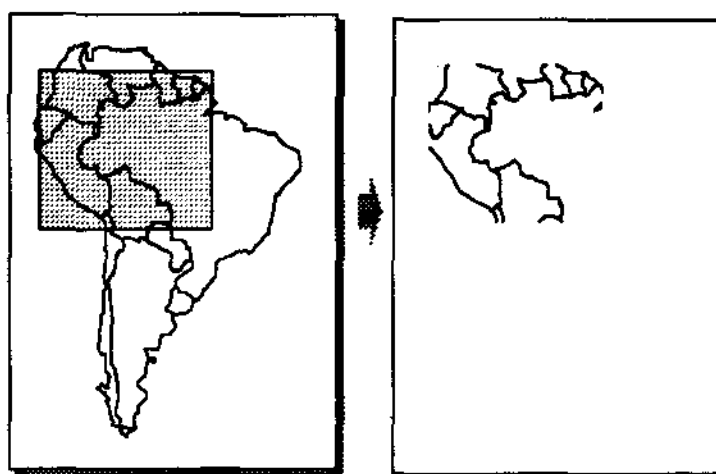


Fig. 4.3 - Clipping

- **Inclusão** - identifica quais geo-objetos estão contidos dentro do geo-objeto fonte.
- **Continência** - identifica quais geo-objetos contêm o geo-objeto fonte. É dual de inclusão.
- **Proximidade** - seleciona geo-objetos que intersectam uma área com uma dada distância ao redor do geo-objeto fonte. A análise de proximidade [Cif95] consiste em gerar um geo-objeto fraco cuja fronteira tem uma distância constante k ao redor do geo-objeto fonte. A região gerada é chamada de *buffer* ou *zona de buffer*. Uma única operação pode gerar um *buffer* ao redor de múltiplos geo-objetos ao mesmo tempo. Múltiplas zonas de *buffer* de raios distintos podem ser gerados ao redor de um geo-objeto formando anéis de proximidade. Um anel de proximidade é uma região delimitada por duas fronteiras distantes $c1$ e $c2$ das bordas do geo-objeto fonte, onde $c2 > c1$. As constantes $c1$ e $c2$ podem ser fixas, ou podem ser calculadas em função de algum atributo do geo-objeto. Por exemplo, anéis de proximidade podem ser gerados ao redor de centros comerciais para estimar a área de influência de cada centro, onde as constantes $c1$ e $c2$ do anel são calculadas em função da área total de cada centro em $m2$ [Ste94].
- **Adjacência** - seleciona geo-objetos adjacentes, ou que estejam vizinhos ao geo-objeto fonte. Não há a necessidade de que os geo-objetos compartilhem fronteiras. Por exemplo, em uma rede de tráfego ferroviário, uma estação de trem é adjacente à próxima estação ao longo da linha de trem.

Um predicado pode exigir que o atributo pesquisado tenha um *valor específico*, pertença a um *intervalo de valores* ou a uma *categoria* pré-definida. Em um sistema de controle de produção agrícola de uma região, uma consulta pode selecionar fazendas que atendam a um dos predicados abaixo, classificadas por tipo:

a) valor específico

"Selecione fazendas que plantam café" (predicado convencional)

"... tenham área igual a 10.000 m²" (predicado espacial métrico)

"... estão na posição geográfica lat a, lon b" (predicado espacial topológico)

b) intervalo de valores

"... produção de 10 a 70 toneladas grãos/ano" (predicado convencional)

"... com distância de um rio inferior a 2 km" (predicado espacial métrico)

c) categoria

"... plantam leguminosas" (predicado convencional)

"... contidos no vale do Paraíba" (predicado espacial topológico)

[Cif95] chama a seleção espacial com predicado espacial topológico de *busca topológica*.

4.4.2 Projeção¹ Espacial

A *projeção espacial* destaca informações dos geo-objetos selecionados por uma seleção espacial. As informações projetadas podem ser atributos não-espaciais, espaciais, calculados, resultados de agrupamentos, ou combinações desses. Por exemplo, as projeções espaciais podem retornar:

a) atributos espaciais (inclusive geo-objetos fracos com qualquer dimensão espacial)

- "Apresente a Localização das fazendas contidas numa região de preservação ambiental". (predicado topológico: "contidas numa região de preservação ambiental").
- "Apresente a Localização das cidades que estão a menos de 1000m da rodovia". (um predicado espacial métrico: "estão a menos de 1000m da rodovia").
- "Onde o rio X intersecta a fazenda Y ?" (projeção de atributos espaciais resultantes de um predicado espacial topológico; predicados de instância: "fazenda Y" e "Rio X").
- "Apresente a Localização das fazendas que estão na região do vale da Capivara" (um predicado espacial topológico de categoria: "estão na região do vale da Capivara")

b) atributos convencionais

- "Qual é o tipo de vegetação da região R ?" (predicado de instância: "região R").

¹ A dissertação utiliza o termo projeção por analogia a sistemas relacionais.

- "...a produção das fazendas que ficam à beira da estrada" (um predicado espacial topológico: "que ficam à beira da estrada").

c) atributos calculados

Um atributo pode ser calculado a partir dos atributos originais do geo-objeto (por exemplo, a soma de dois atributos inteiros ou o resultado de operações espaciais aplicada sobre a Localização).

- "Qual é a área da fazenda Y ?" (um predicado de instância: "fazenda Y"; resultado de uma operação espacial métrica: área).
- "O rio X intersecta a fazenda Y ?" (resultado lógico de uma operação espacial topológica: interseção; predicados de instância: "fazenda Y" e "rio X").

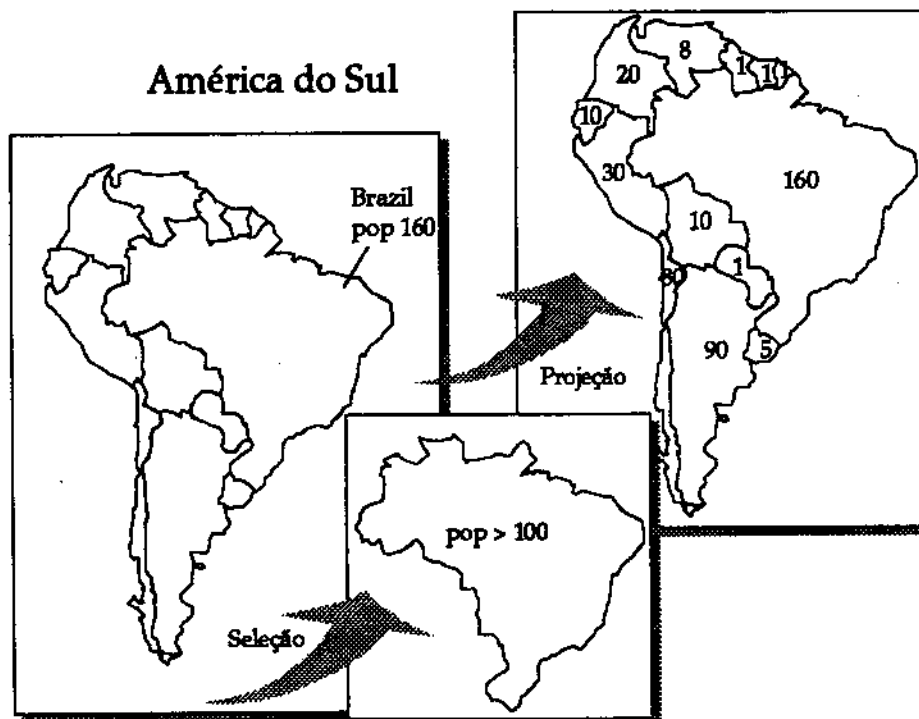


Fig. 4.4 - Projeção Espacial

A projeção pode apresentar dados não-espaciais, dados espaciais ou dados não-espaciais associados à representação espacial. Por exemplo, no mapa dos países da América do Sul (fig.4.4) cada país é um objeto geográfico com o nome do país e a população como atributos não-espaciais. Supondo o mapa à esquerda, temos dois resultados de projeção. No mapa superior, o atributo população é **projetado** associado ao seu país correspondente. No mapa inferior, a operação projeta a população sobre a representação espacial apenas dos países que atendem à restrição dada.

Se o resultado de uma projeção for um geo-objeto composto, pode ser preferível projetar os valores de cada um dos seus componentes. Assim, uma consulta do tipo: "O que está plantado na fazenda X hoje?" teria como resposta uma projeção espacial das divisões da fazenda com o tipo de cultura plantada em cada divisão.

Note que a apresentação dos resultados pode variar nos diferentes SIG, e se possível, a apresentação de um SIG deve ser customizada pelo usuário. No caso do exemplo, o mapa resultante poderia ter cada país apresentado com uma cor e/ou textura correspondente a uma faixa de uma escala de valores populacionais dividida em intervalos pelo usuário. Atributos textuais poderiam ser apresentados em uma outra janela não gráfica. O resultado da seleção poderia ser o mapa original completo com os países que satisfizessem a restrição realçados por uma largura maior da linha de contorno, ou preenchidos com uma textura ou cor específica. As operações de distorção e deslocamento de dados espaciais, ou alteração de características gráficas dos dados apresentados estão relacionadas com a interface da aplicação.

4.4.3 Agrupamento Espacial

O *agrupamento espacial* consiste em acumular os valores de um mesmo atributo de vários geo-objetos, resultantes de uma seleção espacial. Após o agrupamento ser feito, é aplicada uma função de síntese ao conjunto criado, como por exemplo: soma, média, mínimo, máximo, maior, menor, moda, união, interseção, dentre outras. O agrupamento de um atributo é diferente de um atributo calculado, pois este último é feito com os atributos de um mesmo geo-objeto. O agrupamento é geralmente usado em uma seleção com predicado de intervalo ou categoria (seleção múltipla). Os atributos a serem agrupados podem ser convencionais (não-espaciais), espaciais ou calculados. Os atributos calculados podem ser derivados de atributos convencionais ou espaciais. O resultado do agrupamento pode ou não ser armazenado como um novo objeto geográfico.

O agrupamento espacial é feito com valores de diferentes geo-objetos em um mesmo estado temporal. Exemplos de agrupamento de atributos convencionais, espaciais e calculados são respectivamente: "...a soma da produção de fazendas que ficam à beira da estrada" (predicado espacial), "...a união das Localizações das fazendas que estão na região do vale da Capivara" (predicado espacial de categoria), e "...a soma das áreas das fazendas que plantam café" (agrupamento de atributo calculado a partir do atributo espacial: Localização; e predicado convencional). Nesses exemplos, as funções aplicadas ao conjunto formado pelo agrupamento são respectivamente: soma aritmética, união espacial e soma aritmética.

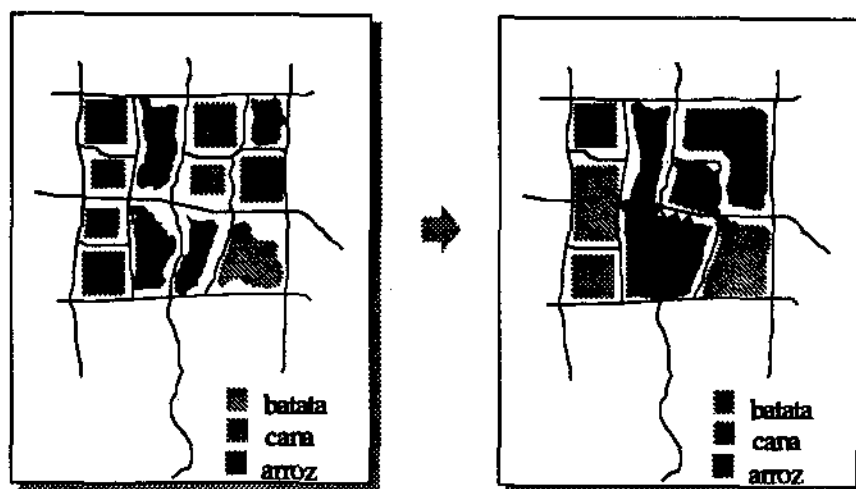


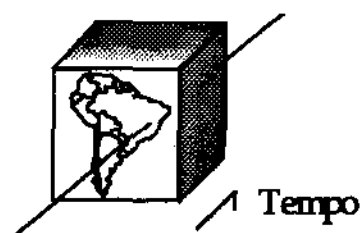
Fig. 4.5 - Agrupamento Espacial por Tipo de Clutura Agrícola

A operação de agrupamento espacial de Localização executa a fusão geométrica dos componentes espaciais dos objetos geográficos que pertencem ao mesmo estado, eliminando as fronteiras entre esses objetos. A figura 4.5 mostra o agrupamento espacial das divisões que cultivam o mesmo produto em um conjunto de fazendas. O mapa resultante passa a conter apenas 3 sub-divisões agrícolas de polígonos desconexos. [Cif95] chama essa operação de reclassificação.

[Cif95] chama o agrupamento espacial de um atributo convencional de superposição numérica. Essa operação projeta sobre a Localização de um objeto geográfico um atributo calculado pelo agrupamento de um ou mais dos seus atributos convencionais. Observe que no modelo proposto, o conteúdo dos temas está descrito nos valores dos atributos convencionais dos geo-objetos. Entretanto em [Cif95] existe um objeto geográfico distinto para cada tema. Nesse caso, polígonos "sliver" podem resultar da superposição de temas onde os objetos geográficos que se superpõem e representam a mesma região tenham a geometria da sua representação espacial um pouco diferente [HOM93].

4.5 Componentes Básicos na Dimensão Temporal

Esta seção descreve consultas com dominância temporal (não-espacial). Seus componentes básicos são também seleção, projeção e agrupamento.



4.5.1 Seleção Temporal

A **seleção temporal** não-espacial recupera valores de atributos de geo-objetos em estados temporais distintos, sem considerar os seus relacionamentos espaciais. Uma seleção temporal pode conter predicados de instância ou predicados sobre atributos convencionais, espaciais, temporais e calculados (por exemplo, a soma de dois valores de um atributo numérico em estados distintos). O predicado é convencional se atua sobre um atributo convencional ou sobre um atributo calculado a partir de um atributo convencional. O predicado é espacial se atua sobre um atributo espacial ou sobre um atributo calculado a partir de um atributo espacial. O predicado é temporal se atua sobre um atributo temporal ou sobre um atributo calculado a partir de um atributo temporal. Os predicados temporais analisados utilizam operações temporais definidas na seção 2.2, por exemplo: antes, depois e durante.

Um exemplo de predicado temporal do tipo valor específico é: "... existiam em 1/1/1964". Um exemplo de predicado temporal do tipo intervalo de valores é: "... fundada após o ano de 1964". E um exemplo de predicado temporal do tipo categoria é: "... plantada durante o governo X".

Em uma seleção temporal, os predicados espaciais utilizam algum ponto ou polígono de referência fornecido pelo usuário. Alguns exemplos de seleções temporais utilizando predicados espaciais são:

Operações Métricas

A operação de distância em uma seleção temporal pode ser usada para determinar em qual intervalo de tempo o geo-objeto alcançou uma certa distância de um ponto (ou polígono) de referência. Por exemplo, "Selecione o período em que a estrada X estava a menos de 1km da área retangular delimitada pelas coordenadas Lat1, Lon1, Lat2 e Lon2".

Operações Topológicas

- **Interseção** - identifica quando um geo-objeto intersectou um dado polígono de referência. A operação pode ainda indicar onde houve essa interseção. Por exemplo, "Selecione a Localização da estrada X no período em que intersectou a área retangular delimitada pelas coordenadas Lat1, Lon1, Lat2 e Lon2".

A operação de *windowing* estendida para suportar a dimensão temporal retorna geo-objetos que intersectam uma forma geométrica desenhada sobre o mapa, em algum estado temporal pertencente ao intervalo de tempo especificado. A operação de *clipping* no contexto temporal trunca o componente espacial e temporal dos geo-objetos seleccionados. Essas operações exigem que seja feita uma seleção temporal em todos os geo-objetos para identificar se o geo-objeto alguma vez intersectou o polígono desenhado sobre o mapa.

- **Inclusão** - identifica quando um geo-objeto esteve incluído em uma área definida pelo usuário. Por exemplo, "Selecione os períodos em que a fazenda X estava incluída na área retangular delimitada pelas coordenadas Lat1, Lon1, Lat2 e Lon2".
- **Continência** - identifica quando um geo-objeto conteve uma área definida pelo usuário. Por exemplo, "Selecione os períodos em que a fazenda X conteve a área retangular delimitada pelas coordenadas Lat1, Lon1, Lat2 e Lon2".

4.5.2 Projeção Temporal

A *projeção temporal* destaca informações dos geo-objetos selecionados por uma seleção temporal. As informações projetadas podem ser atributos não-espaciais, espaciais, temporais, calculados, de agrupamentos, ou combinações desses. Alguns exemplos de projeções temporais com predicados espaciais, e classificadas por tipo de resultado são:

a) atributos espaciais (podem ser geo-objetos fracos com qualquer dimensão espacial)

- "Quais áreas da fazenda X em algum ano intersectaram o retângulo delimitado pelas coordenadas X1,Y1,X2 e Y2 ?" (predicado espacial topológico: "intersectaram o retângulo delimitado...", um predicado de instância: "fazenda X").
- "...a Localização de uma fazenda X no ano que tinha isenção fiscal" (predicado convencional: "tinha isenção fiscal", um predicado de instância: "fazenda X").

b) atributos convencionais

- "Qual era o tipo de vegetação da região R no ano passado ?" (predicado temporal: "no ano passado", predicado de instância: "região R").
- "...o valor da produção de trigo do ano passado de uma fazenda X" (predicado temporal: "do ano passado", um predicado de instância: "fazenda X").

c) atributo temporal

- "Que ano a fazenda X esteve contida no retângulo delimitado pelas coordenadas X1,Y1,X2 e Y2 ?" (predicado espacial topológico: "contida no retângulo delimitado...", predicado de instância: "fazenda X").
- "...o período em que a fazenda X plantou cana" (predicado convencional: "plantou cana", predicado de instância: "fazenda X").

d) atributos calculados

- “Qual é a área da fazenda X no ano passado ?” (predicado temporal: “no ano passado”, predicado de instância: “fazenda X” e resultado de uma operação espacial métrica: área).
- “...a diferença entre o faturamento e as despesas em 1/8/95 da fazenda X” (predicado temporal: “em 1/8/95”, predicado de instância: “fazenda X”).

4.5.3 Agrupamento Temporal

O *agrupamento temporal* consiste em acumular os valores de um mesmo atributo de geo-objetos selecionados por uma seleção temporal. Após ser feito o agrupamento dos valores válidos ao longo do intervalo de tempo especificado, uma função de síntese é aplicada ao conjunto formado (como por exemplo, soma aritmética, máximo intervalo, união espacial e mínimo valor).

O agrupamento temporal de Localização permite que os componentes espaciais de objetos geográficos válidos em diferentes períodos sejam agrupados. Por exemplo, pode-se agrupar todas as representações espaciais antigas de uma fazenda, ou então agrupar apenas as regiões da fazenda que intersectaram um polígono de referência definido na consulta. No primeiro caso, a região resultante corresponderia à área total ocupada pela fazenda em todos os tempos.

A seguir estão relacionados exemplos de agrupamento temporal classificados pelo tipo de atributo agrupado:

a) atributo convencional

- “... a média da taxa de natalidade do Brasil nos últimos 10 anos”, exige um agrupamento da taxa de natalidade coletada anualmente (um predicado de instância: “ Brasil” e um predicado temporal: “ nos últimos 10 anos”).
- “...a soma dos valores de produção de cada ano da fazenda X”, exige um agrupamento dos valores de produção anuais (um predicado de instância: “ fazenda X”).

b) atributo calculado de um atributo convencional

- “...a menor diferença entre faturamento e despesa mensal da fazenda X nos últimos 2 anos”, exige um agrupamento da diferença de dois atributos convencionais (um predicado de instância: “fazenda X” e um predicado temporal: “nos últimos 2 anos”).

c) atributo espacial

- "...a união das Localizações da mata atlântica nos últimos 10 anos", exige um agrupamento da Localização (um predicado de instância: "mata atlântica" e um predicado temporal: "nos últimos 10 anos").

d) atributo calculado de um atributo espacial

- "... a variação média da área ocupada pela mata atlântica nos últimos 10 anos", exige um agrupamento do resultado da operação de cálculo de área da região em cada ano (um predicado de instância: "mata atlântica" e um predicado temporal: "nos últimos 10 anos").

e) atributo temporal

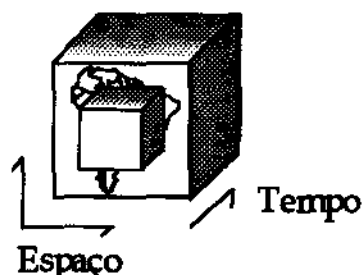
- "...o maior período de cultura da cana da fazenda X", exige um agrupamento dos períodos em que foi plantada cana na Fazenda X (um predicado de instância: "fazenda X" e um predicado convencional: "cultura de cana").

f) atributo calculado de um atributo temporal

- "... a média mensal de inatividade da linha de comunicação X durante o último ano", exige um agrupamento dos períodos em que a linha estava inativa, ou seja, do atributo temporal anexado ao atributo lógico atividade da linha de comunicação X (um predicado convencional: "inatividade", um predicado temporal: "durante o último ano" e um predicado de instância: "linha de comunicação X").

4.6 Componentes Básicos na Dimensão Espaço-Temporal

Nesta seção são discutidas as consultas com dominância espaço-temporal, que percorrem a dimensão espacial e a dimensão temporal de forma irregular e independente. Seus componentes básicos são também seleção, projeção e agrupamento.



4.6.1 Seleção Espaço-Temporal

A seleção espaço-temporal recupera valores de atributos de geo-objetos em estados temporais distintos, considerando os seus relacionamentos espaciais. A seleção espaço-temporal pode ser desmembrada em múltiplas seleções temporais e uma seleção espacial, ou em múltiplas seleções espaciais e uma seleção temporal. Uma seleção espaço-temporal

pode conter predicados de instância ou predicados sobre atributos convencionais, espaciais, temporais e calculados

Alguns exemplos de seleções espaço-temporais utilizando operações espaciais são:

Operações Métricas

A operação de distância em uma seleção espaço-temporal pode ser usada para determinar qual intervalo de tempo o geo-objeto ficou a uma certa distância de outro geo-objeto. Por exemplo, "Selecione o período em que a estrada X estava a menos de 1km do rio Y". Esta consulta efetua uma seleção temporal sobre os estados válidos da estrada X e do rio Y, sendo que uma seleção espacial é feita em cada período em que há interseção temporal desses geo-objetos para identificar se a estrada atende à restrição espacial.

Operações Topológicas

- **Interseção** - caracteriza-se por identificar os geo-objetos e as regiões espaciais que intersectaram o geo-objeto fonte em algum instante de tempo. Por exemplo, o usuário pode consultar se duas fazendas já compartilharam uma fronteira alguma vez.
- **Inclusão** - determina se um geo-objeto esteve contido alguma vez em outro geo-objeto. Por exemplo, o usuário pode desejar saber se em alguma época uma fazenda estava incluída em uma área de preservação.
- **Continência** - determina se um geo-objeto conteve alguma vez outro geo-objeto. Por exemplo, o usuário pode querer saber se em alguma época a fazenda conteve uma plantação de cana.
- **Proximidade** - determina quais geo-objetos estiveram alguma vez perto do geo-objeto fonte. Por exemplo, o usuário pode querer saber quando a fazenda esteve próxima de um lago.
- **Adjacência** - determina se algum geo-objeto esteve alguma vez em uma posição de vizinhança do geo-objeto fonte. Por exemplo, deseja-se saber se alguma vez uma plantação de trigo esteve adjacente a uma plantação de milho em uma fazenda.

4.6.2 Projeção Espaço-Temporal

A projeção espaço-temporal destaca informações dos geo-objetos selecionados por uma seleção espaço-temporal. As informações projetadas podem ser atributos não-espaciais, espaciais, temporais, calculados, de agrupamentos, ou combinações desses. Alguns exemplos de projeções espaço-temporais com predicados espaciais, classificadas por tipo de resultado são:

a) atributos espaciais (podem ser geo-objetos fracos com qualquer dimensão espacial)

- "Quais áreas da fazenda X em algum ano intersectaram uma área de preservação?". (um predicado espacial topológico: " intersectaram uma área ...", e um predicado de instância: " fazenda X").

Uma seleção temporal é feita sobre a fazenda X, sendo que uma seleção espacial é feita em cada estado para identificar se houve interseção com uma área de preservação.

- "... as Localizações das fazendas que intersectam alguma rodovia e que em alguma época tiveram isenção fiscal" (um predicado convencional: "tiveram isenção fiscal", e um predicado espacial: "intersectam alguma rodovia").

Uma seleção espacial é feita para identificar as fazendas que atualmente intersectam alguma rodovia, e depois é feita uma seleção temporal em cada fazenda selecionada para identificar quais as que tiveram isenção fiscal.

b) atributos convencionais

- "Qual era o tipo de vegetação de todas as regiões que estavam contidas numa área de preservação nos dois últimos anos ?" (um predicado temporal: " nos dois últimos anos" e um predicado espacial: " que estavam contidas ...").

Uma seleção temporal de 2 anos de intervalo é feita sobre todas as regiões, sendo que uma seleção espacial é feita em cada estado temporal para identificar se houve alguma região contida em uma área de preservação.

- "... os valores de produção acima de 500 nos últimos 10 anos de todas as fazendas que ficam à beira da estrada" (um predicado espacial: "ficam à beira da estrada", um predicado temporal: "nos últimos 10 anos" e um predicado convencional: "produção acima de 500").

Pode-se considerar que essa consulta realiza uma seleção espacial das fazendas que ficam atualmente à beira da estrada, para então ser feita uma seleção temporal das fazendas selecionadas para identificar quais as que alcançaram produção acima de 500 ton/ha nos últimos 10 anos.

c) atributo temporal

- "Que ano uma fazenda ficou contida em uma área de preservação ?". (um predicado espacial topológico: " ficou contida ...").

Uma seleção temporal é feita sobre todas as fazendas, sendo que uma seleção espacial é feita em cada estado de interseção temporal entre a fazenda e cada instância de área de preservação para identificar se a fazenda esteve contida em uma área de preservação.

- "... os períodos em que fazendas adjacentes produziram acima de 500 ton/ha" (um predicado convencional: "produção acima de 500 ton/ha", um predicado espacial: "adjacente a outra fazenda").

Pode-se considerar que essa consulta realiza uma seleção temporal em todas as fazendas para identificar as que alcançaram produção acima de 500 ton/ha. A seguir é feita uma seleção espacial em cada estado de interseção temporal entre a fazenda e cada instância dentre as outras fazendas para identificar quais são adjacentes.

d) atributos calculados

- "Qual é a maior área alcançada pelas fazendas que alguma vez intersectaram um rio?" (um predicado espacial: "que alguma vez intersectaram um rio").

Pode-se considerar que essa consulta realiza uma seleção temporal em todas as fazendas, sendo que uma seleção espacial é feita em cada estado de interseção temporal entre a fazenda e cada instância de rio para identificar se a fazenda intersectou um rio. A seguir é feito um agrupamento espaço-temporal das áreas para identificar qual foi a maior.

- "... área de fazendas adjacentes que alguma vez tiveram produção de cana do tipo 1 ou cana do tipo 2 acima de 100" (um predicado espacial: "fazendas adjacentes", um predicado convencional: "produção de cana do tipo 1 ou cana do tipo 2 acima de 100")

Pode-se considerar que uma seleção temporal é feita em todas as fazendas para identificar quais tiveram produção de cana do tipo 1 ou cana do tipo 2 acima de 100, sendo que uma seleção espacial é feita em cada estado de interseção temporal entre a fazenda e cada instância dentre as outras selecionadas para identificar aquelas que são adjacentes.

4.6.3 Agrupamento Espaço-Temporal

O agrupamento espaço-temporal agrupa valores de um mesmo atributo de geo-objetos selecionados por uma seleção espaço-temporal. Ou seja, o valor de um atributo de um geo-objeto de uma época pode ser agrupado com o valor desse atributo de outro geo-objeto em outra época. O agrupamento espaço-temporal pode ser desmembrado em múltiplos agrupamentos temporais e um agrupamento espacial, ou em múltiplos

agrupamentos espaciais e um agrupamento temporal. Exemplos de agrupamentos espaço-temporais classificados por tipo de atributo são:

a) atributo convencional

- "... a soma da produção dos últimos 10 anos das fazendas que ficam à beira da estrada". (um predicado espacial: "ficam à beira da estrada" e um predicado temporal: "dos últimos 10 anos")

A consulta faz inicialmente uma seleção espacial das fazendas que ficam à beira da estrada. Então a consulta faz um agrupamento temporal da produção de cada fazenda selecionada. Daí é feito um agrupamento espacial das produções agrupadas das fazendas.

b) atributo temporal

- "... o maior período de produção de uma fazenda contida em uma área de preservação" (predicado espacial: "fazenda contida em uma área de preservação").

Uma seleção espacial é feita para selecionar as fazendas que estão em uma área de preservação. Então um agrupamento espaço-temporal dos períodos de produção das fazendas selecionadas é feito para identificar o maior período de produção agrícola.

c) atributo espacial

- "... o agrupamento (união) das Localizações das fazendas que intersectam um rio nos períodos em que tiveram isenção fiscal" (um predicado convencional: "tiveram isenção fiscal" e um predicado espacial: "intersectam um rio").

Uma seleção espacial é feita para selecionar as fazendas que intersectam um rio. Então é feita uma seleção temporal em cada fazenda selecionada para identificar aquelas que tiveram isenção fiscal. Daí é feito um agrupamento espaço-temporal da Localização das fazendas selecionadas nos períodos selecionados.

d) atributo calculado

- "... maior produção de cana do tipo 1 ou cana do tipo 2 de todas as fazendas que alguma vez intersectaram alguma estrada" (um predicado espacial: "intersectaram alguma estrada")

Uma seleção temporal é feita em cada fazenda para identificar quais as que produziram cana do tipo 1 ou 2, sendo que uma seleção espacial é feita em cada estado de interseção temporal para identificar quais delas intersectaram uma estrada alguma vez. Então é feito um agrupamento espaço-temporal da produção das fazendas selecionadas.

4.7 Descrição Global das Consultas

Em resumo, podemos dizer que uma consulta pode ser aplicada nas dimensões espacial, temporal e espaço-temporal. A consulta consiste na aplicação de um predicado sobre um domínio para obter um resultado. A consulta é traduzida nos componentes seleção, agrupamento e projeção em uma linguagem de consulta. A estrutura dos principais conceitos apresentados é a seguinte:

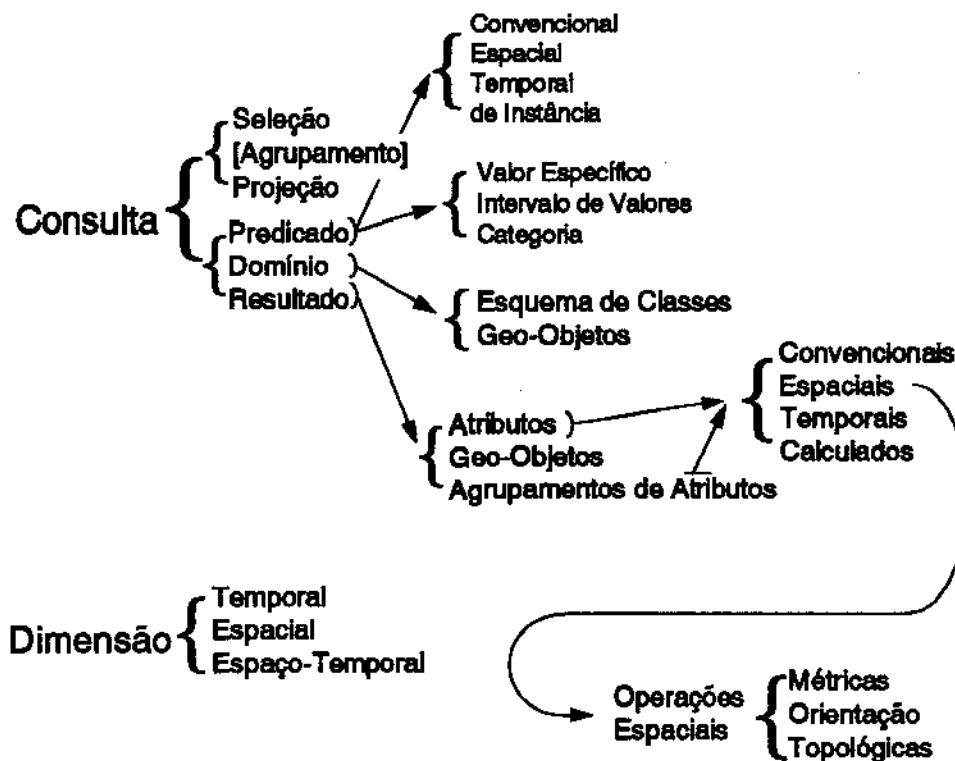


Fig. 4.6 - Taxonomia das Consultas

4.8 Operações entre Mapas

4.8.1 União de Mapas²

A operação de união de mapas consiste em projetar a Localização de todos geo-objetos de um ou mais temas em um único mapa. Esta operação gera uma visão de uma combinação de temas sem modificar os geo-objetos desses temas. Os temas são projetados seguindo uma ordem de relevância, onde os temas mais úteis para o usuário naquele momento são projetados por último, sobrepondo os primeiros.

A união de mapas espaço-temporal ocorre quando há uma união de temas de estados temporais distintos. Um exemplo para este caso seria a união do mapa das fazendas há dez anos com o mapa hidrográfico atual (fig 4.7).

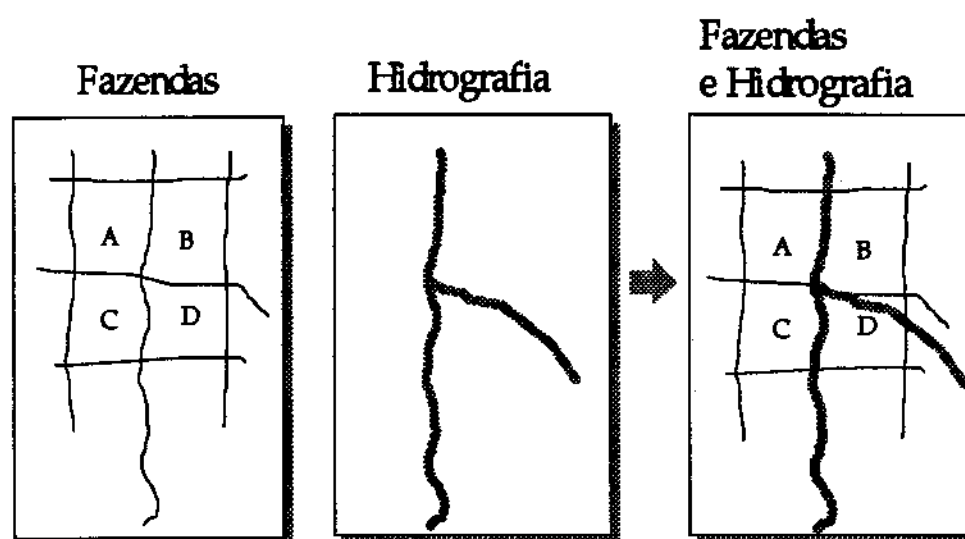


Fig. 4.7 - União de Mapas

4.8.2 Sobreposição³ de Mapas

A sobreposição de mapas combina vários temas dos geo-objetos originais e deriva um novo mapa contendo apenas geo-objetos fracos. Esta operação costuma ser aplicada em campos. O valor do único atributo não-espacial desses geo-objetos fracos gerados contém a união dos valores dos atributos originais correspondentes aos temas sobrepostos.

² Essa operação também é chamada de superposição *cookie cutter* [Cif95].

³ Esta operação é chamada de *Superposição* em [Cif95].

A análise de ponderação [Cif95] pode ser feita através dessa soma de atributos temáticos não-espaciais gerada pela sobreposição de mapas.

[Cif95] especifica uma variação dessa operação chamada de *superposição booleana*, onde operadores booleanos são aplicados aos temas originais. Neste caso, as regiões que apresentam um tema específico têm valor 1 e as que não o apresentam têm valor 0. A região resultante será formada pelas regiões que apresentam valor 1 em todos os temas envolvidos.

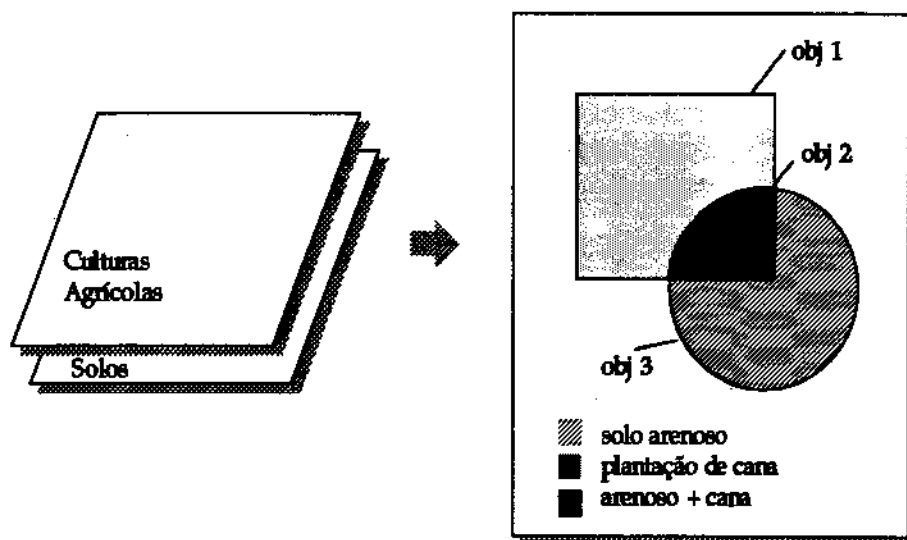


Fig. 4.8 - Sobreposição de Mapas

Na figura 4.8, o geo-objeto fraco 1 (plantação de cana) oriundo do tema cultura agrícola, tem uma região em comum com o geo-objeto fraco 3 (solo arenoso) proveniente do tema de solos. Após a sobreposição surgirá o geo-objeto 2 (plantação de cana em solo arenoso) correspondendo à combinação desses dois temas. A operação de sobreposição cria tantos geo-objetos fracos quantos forem os polígonos resultantes da sobreposição.

Em um sistema temporal, as representações espaciais dos geo-objetos originais podem ser de estados temporais distintos, mas os geo-objetos fracos gerados não têm atributo temporal. Um exemplo seria a sobreposição do mapa atual de solos com a distribuição de culturas do ano anterior. A operação de sobreposição não permite a utilização de intervalos temporais, pois isso exigiria que polígonos diferentes pudessem ocupar um mesmo lugar. Por exemplo, sobrepor o mapa de solos sobre o mapa de culturas agrícolas durante o período T1 a T5. Neste caso, uma mesma região poderia ter várias configurações em instantes distintos.

4.8.3 Decomposição Espacial

A decomposição espacial cria objetos geográficos a partir dos polígonos não-conectados do geo-objeto fonte. O exemplo da figura 4.9 mostra que foram criados 3 novos objetos geográficos a partir das regiões desconexas das plantações de cana. Nesse exemplo as plantações de batata são decompostas em duas regiões.

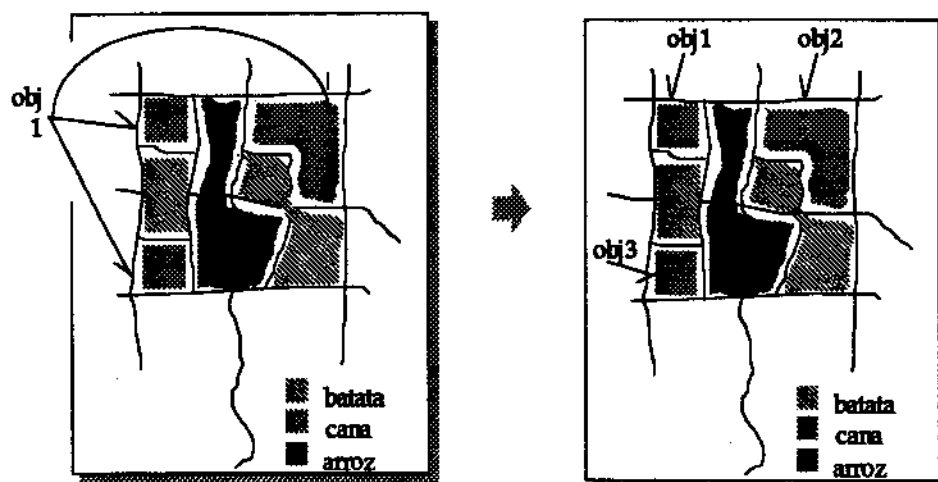


Fig. 4.9 - Decomposição Espacial por Tipo de Cultura Agrícola

A decomposição temporal cria objetos geográficos a partir dos intervalos temporais distintos do geo-objeto fonte. Por exemplo, a decomposição temporal da Localização de uma fazenda cria um geo-objeto independente para cada instância temporal de Localização da fazenda válida em um intervalo de sua existência.

[Cif95] define essa operação como a separação da fronteira de uma região da sua área interior.

4.8.4 Outros Tipos de Operação

Algumas operações não se enquadram em nenhuma das operações citadas anteriormente. Exemplos são: controle de fluxo, roteamento de frota, caminho mínimo e outras. Essas operações são específicas de alguns grupos de aplicações geográficas, e por isso não serão consideradas nessa dissertação.

4.9 Resumo

Este capítulo descreveu e classificou as principais categorias de consultas espaço-temporais. Estas consultas são caracterizadas de acordo com a dominância nos eixos espaço-temporais. O próximo capítulo descreve estruturas de dados adequadas para permitir a execução de todas as categorias descritas. Estas estruturas permitem armazenar dados de acordo com o modelo do capítulo 3.

Capítulo 5

Estrutura dos Dados Espaço-Temporais

5.1 Introdução

Muito se tem pesquisado na área de armazenamento da evolução temporal dos dados. Como resultado desse esforço, foram implementados vários bancos de dados temporais [TCG+95]. Entretanto, são poucas as publicações que tratam do armazenamento da evolução temporal de dados espaciais em sistemas geográficos.

As aplicações que manipulam dados espaço-temporais exigem sistemas computacionais com grande capacidade de armazenamento e alta velocidade de processamento. De fato, as poucas experiências feitas no sentido de desenvolver bancos de dados espaço-temporais fracassaram devido ao baixo desempenho alcançado. A principal conclusão obtida é que são necessárias novas técnicas de armazenamento e acesso a dados espaço-temporais.

Ainda não há um consenso sobre como saber quais dados geográficos devem ser temporais e que tipo de informação temporal deve ser armazenada. [Lan93] aponta várias questões que devem ser decididas antes da implementação de um sistema geográfico temporal:

- Para decidir quais dados devem ser temporalizados, o usuário deve analisar fatores como a volatilidade do dado, a importância do dado no sistema e a responsabilidade organizacional do dado.
- Em alguns casos, é importante armazenar não somente quando o dado é alterado, mas também quando é validado.
- A atualização de dados espaciais pode ser feita incrementalmente ou totalmente.

- Pode ser mais importante saber o que foi mais comum durante um certo intervalo de tempo, do que saber precisamente a realidade em um instante específico de tempo, ou seja, a precisão de dados isolados pode não ser tão importante para a aplicação. Neste caso, o SIG deve prover suporte para operações de generalizações temporais.

O armazenamento da evolução temporal dos dados convencionais descrito em [Bra94] foi utilizado no modelo proposto nesta dissertação. Neste capítulo, propomos uma estrutura de dados para armazenar a evolução temporal da Localização dos geo-objetos. Além disso, mostramos como aplicar as consultas apresentadas no capítulo 3 usando essa estrutura.

5.2 Estrutura de Dados Espaciais Atemporais

[Ci95] descreve diversas estruturas de dados propostas na literatura para armazenar dados espaciais usando o modelo relacional, como por exemplo: *total* (Whole Polygon Structure), *spaghetti*, *topológico*, *DIME* (Dual Independent Map Encoding), *Arc-Node* e o *modelo de objetos relacional*. Os dados espaciais armazenados nessas estruturas estão no formato vetorial.

A estrutura proposta aqui baseia-se no modelo definido no capítulo 3 onde um geo-objeto é composto de: atributos não-espaciais, uma lista de geo-objetos, e a sua Localização, que é uma lista de instâncias de Geometria.

Uma instância de Geometria contém uma lista de objetos geométricos que podem ser: pontos, linhas, polígonos ou multi-polígonos. Uma linha contém uma lista de pontos, um polígono contém uma lista de linhas, e um multi-polígono contém uma lista de polígonos. Essa estrutura é mais compacta do que as baseadas no modelo relacional, devido principalmente à possibilidade de estruturas orientadas a objetos terem atributos de tamanho variável. Esta forma de definição é semelhante ao modelo *spaghetti*, exceto que linhas e polígonos são armazenados como objetos, enquanto que no modelo *spaghetti* a visão é unificada para o ponto.

O exemplo abaixo mostra o geo-objeto fazenda e suas divisões agrícolas modelados usando a definição do capítulo 3. Os geo-objetos são representados por instâncias das classes geográficas *Fazenda* (geo-objeto-composto) e *Divisão_agrícola* (geo-objeto fraco). As instâncias de *Divisão_agrícola* *div1* e *div2* representam as divisões da fazenda. A instância de *Fazenda* *faz1* representa a fazenda e a sua lista de geo-objetos contém *div1* e *div2*. As Localizações de *faz1*, *div1* e *div2* contêm apenas uma representação espacial cada uma: *G1*, *G2* e *G3* respectivamente. A figura 5.1 mostra a Localização de *faz1*. As tabelas abaixo mostram o conteúdo dos objetos desse exemplo.

Fazendas

Fazenda Id	Nome	Proprietário
faz1	São João	Francisco Pinto

Divisão agrícola

Divisão agrícola Id	Nome	Cultura	Produção
div1	Divisão Norte	Trigo	100
div2	Divisão Sul	Cana	200

Localização

Localização Id	Lista de Geometria
loc_faz1	G1
loc_div1	G2
loc_div2	G3

Geometria

Geometria Id	Lista de Objetos Geométricos	Atributos não-Espaciais		
		Projeção	Escala	...
G1	Pol1, Pol2	ProjeçãoG1	EscalaG1	...
G2	Pol1	ProjeçãoG2	EscalaG2	...
G3	Pol2	ProjeçãoG3	EscalaG3	...

Polígono

Polígono Id	Lista de Linhas
Pol1	l1, l2
Pol2	l1, l3

Linha

Linha Id	Lista de Pontos
l1	p1,...p[l1]n ..., p2
l2	p1, ...p[l2]n ..., p2
l3	p1, ...p[l3]n ..., p2

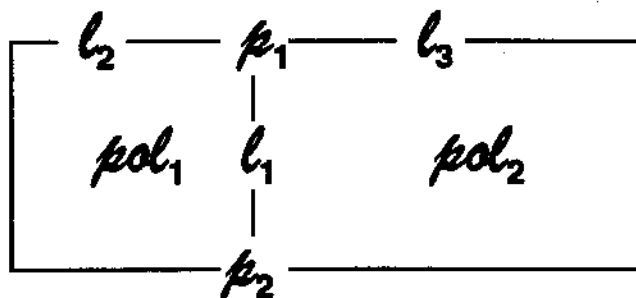


Fig 5.1 - Representação Espacial de um Geo-objeto

5.3 A Formação dos Polígonos

No mundo real, duas regiões geográficas vizinhas compartilham uma mesma fronteira, portanto os polígonos que as representam no SIG devem compartilhar as linhas que os separam, evitando a redundância de dados espaciais e diminuindo o risco de haver inconsistências [OvdB89].

Todos os polígonos do modelo são supostos convexos e simples. No banco de dados, segundo a estrutura descrita, um polígono pode ser criado de três formas:

- A partir da delimitação de uma **nova região**. Neste caso, novas linhas são criadas para representar a fronteira do novo polígono.
- A partir do **particionamento de um polígono existente**. Neste caso, as linhas de particionamento devem ser criadas. Além disso, cada linha que conecta uma das extremidades de uma linha de particionamento criada, pode ser substituída por duas novas linhas. Observe que em alguns casos, pode ser preferível que os polígonos remanescentes sejam novos e que o polígono original deixe de existir.

Segundo esta proposta, o particionamento de um polígono exige a eliminação de algumas linhas e a criação de novas linhas. O exemplo mostra o particionamento de Pol2 em Pol4 e Pol5 num instante t2, transformando a linha l1 em l1a e l1b e a linha l3 em l3a e l3b (fig 5.2).

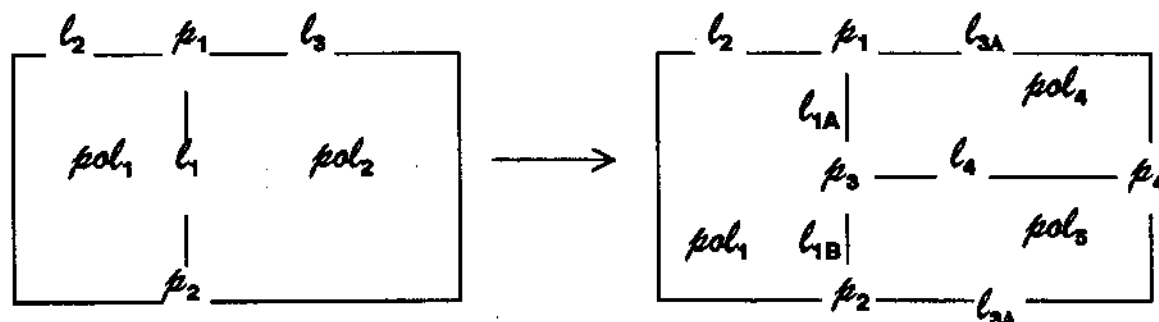


Fig 5.2 - Particionamento de um Polígono

As tabelas de linhas e polígonos ficam da seguinte forma:

Polígono

Polígono Id	Lista de Linhas	Tempo Válido
Pol1	l1, l2	[0-t2]
Pol1	l1a, l1b, l2	[t2-∞]
Pol2	l1, l3	[0-t2]
Pol4	l1a, l3a, l4	[t2-∞]
Pol5	l1b, l4, l3b	[t2-∞]

Linha

Linha Id	Lista de Pontos	Tempo Válido
l1	p1,...p[l1]n ..., p2	[0-t2]
l1a	p1,...p[l1a]n ..., p3	[t2-∞]
l1b	p3,...p[l1b]n ..., p2	[t2-∞]
l2	p1, ...p[l2]n ..., p2	[0-∞]
l3	p1, ...p[l3]n ..., p2	[0-t2]
l3a	p1, ...p[l3a]n ..., p4	[t2-∞]
l3b	p4, ...p[l3b]n ..., p2	[t2-∞]
l4	p3, ...p[l3]n ..., p4	[t2-∞]

- A partir da **junção de dois polígonos vizinhos**. Neste caso, nenhuma linha é criada. As linhas que separavam os dois polígonos vizinhos deixam de existir e as outras linhas são incorporadas ao polígono remanescente.

Segundo esta proposta, a junção de um polígono exige apenas a eliminação das linhas de fronteira entre os dois polígonos. O exemplo abaixo mostra a junção de Pol4 e Pol5 num instante t2, eliminando a linha l4 (fig 5.3).

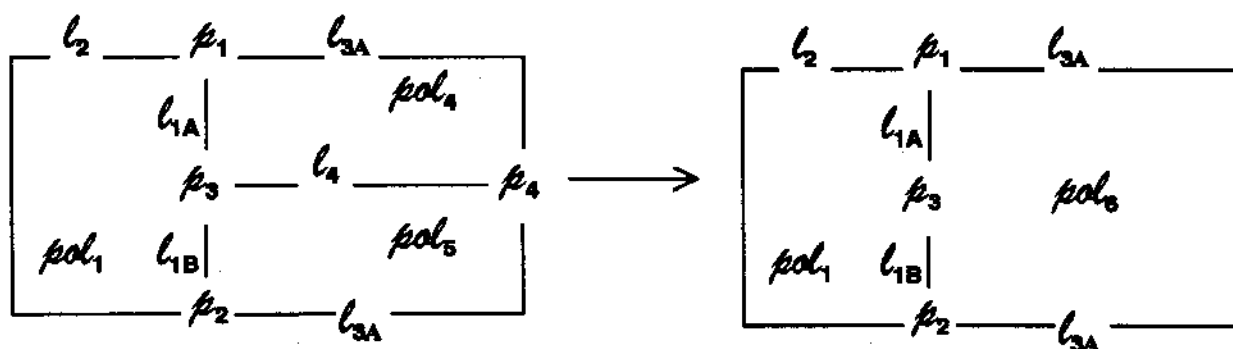


Fig 5.3 - Junção de um Polígono

As tabelas de linhas e polígonos ficam da seguinte forma:

Polígono

Polígono Id	Lista de Linhas	Tempo Válido
Pol1	l1a, l1b, l2	[0-∞]
Pol4	l1a, l3a, l4	[0-t2]
Pol5	l1b, l4, l3b	[0-t2]
Pol6	l1a, l1b, l3b, l3a	[t2-∞]

Linha

Linha Id	Lista de Pontos	Tempo Válido
l1a	p1,...p[l1a]n ..., p3	[0-∞]
l1b	p3,...p[l1b]n ..., p2	[0-∞]
l2	p1, ...p[l2]n ..., p2	[0-∞]
l3a	p1, ...p[l3a]n ..., p4	[0-∞]
l3b	p4, ...p[l3b]n ..., p2	[0-∞]
l4	p3, ...p[l3]n ..., p4	[0-t2]

5.4 Exemplos de Consultas

Seja um banco de dados com classes Fazenda, Divisão_agrícola, Rio e Reserva. Esta seção apresenta exemplos de consultas espaciais, temporais e espaço-temporais usando a linguagem TOOL [OM93] descrita na seção 2.3. Para exemplificar tais consultas são propostas algumas operações espaço-temporais (algoritmos no anexo).

Algumas operações espaço-temporais envolvem apenas um geo-objeto (por exemplo, área, perímetro e comprimento). Neste caso, a operação recebe como parâmetro um geo-objeto e um intervalo temporal sobre o qual esta operação deve ser aplicada. Por exemplo, a operação espaço-temporal φ aplicada sobre o geo-objeto Fazenda1 para o período t1 a t2 seria descrita da seguinte forma: " φ (Fazenda1, t1, t2)". A operação φ retorna uma lista de valores numéricos associados aos respectivos intervalos em que a Localização de Fazenda1 é constante durante o período t1 a t2. Na ausência do parâmetro t2 é assumido o intervalo degenerado t1.

Outras operações espaço-temporais envolvem mais de um geo-objeto (por exemplo, distância, interseção, inclusão e continência). Neste caso, com exceção da operação de distância que retorna valores numéricos, as demais operações retornam valores booleanos. Para simplificar os exemplos, as operações apresentadas utilizam apenas dois geo-objetos. Portanto, a operação recebe como parâmetro dois geo-objetos e um intervalo temporal sobre o qual esta operação deve ser aplicada. Por exemplo, a operação espaço-temporal λ aplicada sobre o geo-objeto Fazenda1 e Rio1 para o período t1 a t2 seria descrita da seguinte forma: " λ (Fazenda1, Rio1, t1, t2)". A operação λ retorna uma lista de valores

associados aos respectivos intervalos em que a Localização de Fazenda1 e a Localização de Rio1 são constantes durante o período t1 a t2. Na ausência do parâmetro t2 é assumido o intervalo degenerado t1. Essas operações podem ser sobrecarregadas para poderem receber uma lista de geo-objetos, como no caso das consultas aninhadas.

Os detalhes de implementação estão no fim deste capítulo. As consultas utilizam a cláusula VSLICE da linguagem TOOL para limitar o escopo da pesquisa no eixo de tempo válido.

Uma consulta pode exigir a aplicação de uma operação espacial várias vezes. Por exemplo, a consulta: “Selecione todas as fazendas que intersectam um rio” é traduzida em múltiplas operações de interseção (uma para cada estado de interseção temporal entre cada instância de rio e fazenda). Ou ainda, a consulta: “Selecione todas as fazendas contidas na reserva Juruá” necessitaria de uma operação de inclusão para cada estado de interseção temporal entre cada instância de fazenda e a reserva Juruá.

5.4.1 Consultas Espaciais

Nas consultas espaciais atemporais, a cláusula VSLICE do modelo TOODM [OM93] pode ser omitida. Na ausência de valor no parâmetro de tempo-válido, é considerado o estado atual.

Alguns exemplos de consultas com predicado espacial são:

- A consulta “Selecione todas as fazendas distantes até 10km do rio X” é traduzida em múltiplas operações de distância. A função *in_range* retorna um valor lógico indicando se existe pelo menos um valor dentro do intervalo numérico especificado. Usando a linguagem TOOL, esta consulta ficaria assim:

```
SELECT f
FROM f in Fazenda VSLICE [Now]
WHERE in_range( distância(f,
                        (SELECT r
                         FROM Rio VSLICE [Now]
                         WHERE r.nome == X),
                        now),
                0km,10km)
```

- A consulta “Selecione as divisões agrícolas que plantam cana e têm área maior que 1000ha” é mapeada em operações de área.

```
SELECT d
FROM d in Divisão_agrícola VSLICE [Now]
```

WHERE in_range(area(d, now), 1000ha, ∞) and d.cultura == cana

- A consulta “Selecione as fazendas contíguas à fazenda X” é mapeada em múltiplas operações de interseção de Localização (se existe interseção entre as linhas de dois polígonos, eles são contíguos).

```
SELECT f
FROM f in Fazenda VSLICE [Now]
WHERE f.nome = X and
      not empty( interseção(f,
                           (SELECT f2
                              FROM f2 in Fazenda VSLICE [Now]
                              WHERE f2.nome == X),
                           now ) )
```

Alguns exemplos de consultas espaciais que utilizam as operações espaciais para a projeção de um atributo calculado a partir da Localização de um geo-objeto são:

- A consulta: “Determine o valor da área da Fazenda X” usa a operação de área para calcular a área da Localização da fazenda X.

```
SELECT area(f, now)
FROM f in Fazenda VSLICE [Now]
WHERE f.nome == X
```

- A consulta: “Determine quais rios intersectam a Fazenda X” usa múltiplas operações de interseção entre a fazenda X e cada instância de rio.

```
SELECT interseção(f, r, now)
FROM f in Fazenda, r in Rio VSLICE [Now]
WHERE f.nome == X
```

5.4.2 Consultas Temporais

Alguns exemplos de consultas temporais são:

- A consulta: “Selecione as Localizações das divisões agrícolas nos períodos que plantaram cana entre 1980 a 1990” não utiliza nenhuma operação espacial, pois a consulta tem um predicado convencional (cana) e um predicado temporal. O método *loc*, definido para a classe Geo-objeto (veja a seção 5.8), pode ser aplicado a qualquer geo-objeto. Este método retorna a lista das geometrias de um geo-objeto válidas no período especificado. O operador $[[\xi]]$ retorna o elemento temporal em que a expressão ξ é válida (veja a seção 2.1.13).

```
SELECT d→loc([[f.cultura == cana]])  
FROM d in Divisão_agrícola VSLICE [1980-1990]  
WHERE d.cultura == cana
```

Esta consulta é processada da seguinte forma:

1. a cláusula VSLICE restringe o domínio da consulta para o período 1980 a 1990;
2. a cláusula WHERE seleciona as fazendas que plantaram cana nesse período;
3. a cláusula SELECT retorna a Localização dessas fazendas durante o(s) período(s) que plantou cana.

Note que uma fazenda pode ter várias versões temporais de sua Localização correspondentes aos períodos em que plantou cana entre 1980 e 1990. Neste caso, todas as versões temporais da Localização da fazenda são retornadas.

Em consultas temporais, a utilização das operações espaciais em predicados espaciais é limitada a comparar os geo-objetos com um objeto geométrico de referência passado como parâmetro.

Alguns exemplos de consultas com predicados espaciais são:

- A consulta: “Selecione o período em que a fazenda X esteve incluída na área [x1,y1,x2,y2]” é traduzida em uma operação de inclusão na janela fornecida para cada versão temporal da fazenda. A função *to_Obj* retorna um geo-objeto com uma geometria constante delimitada pelas as coordenadas indicadas.

```
SELECT [[inclusão(f, to_Obj (x1,y1,x2,y2) , 0, now)]]  
FROM f in Fazenda VSLICE [0-Now]  
WHERE inclusão(f, to_Obj (x1,y1,x2,y2) , 0, now)  
and f.nome == X
```

- A consulta: “Selecione o nome das fazendas que contiveram o ponto [x1,y1] alguma vez após 1980” é traduzida em uma operação de continência para todas as fazendas.

```
SELECT f.nome  
FROM f in Fazenda VSLICE [1980-Now]  
WHERE continência(f, to_Obj (x1,y1), 1980, now)
```


- A consulta: “Selecione as Localizações das fazendas que tinham área maior que 1000 ha em 1980” é mapeada para uma operação de área para cada fazenda.

```
SELECT f→loc(1980),  
FROM f in Fazenda VSLICE [1980]  
WHERE in_range( area(f, 1980), 1000ha, ∞ )
```

- A consulta: “Determine o valor da área da Fazenda X em 1970” é traduzida em uma operação de área.

```
SELECT area(f, 1970)  
FROM f in Fazenda VSLICE [1970]  
WHERE f.nome = X
```

5.4.3 Consultas Espaço-Temporais

Nesse tipo de consulta, as operações espaciais podem ser utilizadas em predicados espaço-temporais, os quais testam o relacionamento espaço-temporal entre geo-objetos.

Alguns exemplos de consultas com predicados espaço-temporais são:

- A consulta: “Selecione a Localização atual das Fazendas que intersectaram a reserva ambiental de Juruá nos anos 1970 a 1980” é traduzida em múltiplas operações de interseção entre cada fazenda e a reserva “Juruá”.

```
SELECT f→loc(now)  
FROM f in Fazenda VSLICE [1970-1980]  
WHERE interseção(f,  
                  (SELECT r  
                   FROM r in Reserva VSLICE [1970-1980]  
                   WHERE r.nome='Juruá'),  
                  1970, 1980)
```

- A consulta: “Selecione as interseções das Localizações de todas as fazendas com as Localizações de todas as reservas ambientais criadas nos anos 1970 a 1980”.

```
SELECT interseção(f, r, 1970, 1980)  
FROM f in Fazenda, r in Reserva VSLICE [1970-1980]
```

5.5 Estrutura Interna das Classes Geométricas

Esta seção descreve a especificação das classes geométricas Ponto, Linha, Polígono e MultiPolígono. As operações espaço-temporais acionam o método *vslice_s* de cada objeto geométrico de um geo-objeto para recuperar a sua geometria. O método *vslice_s* recupera

uma fatia temporal do objeto geométrico válida no intervalo de tempo passado como parâmetro. Este método retorna uma lista de pontos equivalente ao contorno do objeto geométrico no tempo-válido especificado. O método *vslice_s* tem duas partes distintas (exceto na classe Ponto). A primeira parte verifica quais versões temporais daquele objeto geométrico são válidas no intervalo de tempo especificado. Para cada versão válida, a segunda parte do método aciona o método *vslice_s* de todos os seus componentes.

O método *vslice_s* da classe Ponto verifica apenas se o domínio temporal do ponto intersecta o intervalo de tempo especificado, pois os pontos não têm versionamento temporal. Os identificadores dos pontos válidos naquele intervalo são retornados para os seus objetos superiores nas hierarquias de composição, até que a lista de pontos formada seja retornada para a consulta em execução. A fim de simplificar os algoritmos, não consideramos os multipolígonos com buracos. Para representar a descontinuidade que um objeto geométrico pode ter, a lista de pontos retornada pelo método *v_slice* possui as constantes do tipo ponto: *begin_LINHA*, *end_LINHA*, *begin_POLIGONO*, *end_POLIGONO*, *begin_MULTIPOLIGONO* e *end_MULTIPOLIGONO*, indicando o início e o fim de um objeto geométrico.

O esquema das classes geométricas fica assim:

```
class Obj_Geom
    type dummy: integer;
```

```
class Ponto inherit Obj_Geom
type objeto:tuple( coord: tuple ( x: Real, y: Real ), t:list ( tuple (inicio: integer, fim: integer) ) )

method vslice_s( inicio_s: integer, fim_s: integer)
    resultado : list(tuple ( x: Real, y: Real ));
    tmp: tuple (inicio: integer, fim: integer );
    t_overlap : boolean;
begin
    t_overlap = FALSE;
    resultado = ( );
    for each tmp in self.objeto.t do {
        if ((tmp.inicio <= fim_s) & (tmp.inicio >= inicio_s)) ||
           ((tmp.fim <= fim_s) & (tmp.fim >= inicio_s))
            t_overlap = TRUE; break; }
    if t_overlap /* Se há overlap temporal */
        resultado += self.objeto.coord;
return resultado;
end
```

```

class Linha inherit Obj_Geom
type objeto:list( tuple( pts: list ( Ponto ), t:list ( tuple (inicio: integer, fim: integer ) ) ) )

method vslice_s(inicio_s: integer, fim_s: integer)
  resultado : list(tuple ( x: Real, y: Real ));
  estado: tuple( pts: list ( Ponto ), t:list ( tuple (inicio: integer, fim: integer ) ) );
  ponto1 : ponto;
  tmp: tuple (inicio: integer, fim: integer );
  t_overlap : boolean;
begin
  t_overlap = FALSE;
  for each estado in self.objeto do {
    for each tmp in estado.t do {
      if ((tmp.inicio <= fim_s) & (tmp.inicio >= inicio_s)) ||
        ((tmp.fim <= fim_s) & (tmp.fim >= inicio_s))
        t_overlap = TRUE; break; }
      if t_overlap
        resultado += begin_LINHA /* separador */
        for each ponto1 in estado.pts do
          resultado += ponto1→vslice_s(inicio_s, fim_s);
        resultado += end_LINHA;
      }
  }
return resultado;
end

```

```

class Poligono inherit Obj_Geom
type objeto:list( tuple( lns: list ( Linha ), t:list ( tuple (inicio: integer, fim: integer ) ) ) )

method vslice_s(inicio_s: integer, fim_s: integer)
  resultado : list(tuple ( x: Real, y: Real ));
  estado: tuple( lns: list ( Linha ), t:list ( tuple (inicio: integer, fim: integer ) ) );
  linha1: linha;
  tmp: tuple (inicio: integer, fim: integer );
  t_overlap : boolean;
begin
  t_overlap = FALSE;
  for each estado in self.objeto do {
    for each tmp in estado.t do {
      if ((tmp.inicio <= fim_s) & (tmp.inicio >= inicio_s)) ||
        ((tmp.fim <= fim_s) & (tmp.fim >= inicio_s))
        t_overlap = TRUE; break;}

    if t_overlap

```

```

        resultado += begin_POLIGONO /* separador */
        for each linha1 in estado.lns do
            resultado += linha1→vslice_s(inicio_s, fim_s);
        resultado += end_POLIGONO
    }
    return resultado;
end

```

```

class MultiPoligono inherit Obj_Geom
type objeto: list( tuple( mp: tuple( pols: list (Poligono), pols_b: list (Poligono) ), t: list ( tuple
(inicio: integer, fim: integer) ) ) )

method vslice_s(inicio_s: integer, fim_s: integer)
    resultado : list(tuple ( x: Real, y: Real ));
    estado: tuple( mp: tuple( pols: list (Poligono), pols_b: list (Poligono) ), t: list ( tuple
(inicio: integer, fim: integer) ) );
    poligono1: poligono;
    t_overlap : boolean;
    tmp: tuple (inicio: integer, fim: integer );
begin
    t_overlap = FALSE;
    for each estado in self.objeto do {
        for each tmp in estado.t do {
            if ((tmp.inicio <= fim_s) & (tmp.inicio >= inicio_s)) ||
                ((tmp.fim <= fim_s) & (tmp.fim >= inicio_s))
                t_overlap = TRUE; break; }
            if t_overlap
                resultado += begin_MULTIPOLIGONO
                for each poligono1 in estado.mp.pols do
                    resultado += poligono1→vslice_s(inicio_s, fim_s);
                resultado += end_MULTIPOLIGONO
            }
        }
    return resultado;
end

```

5.6 Implementação de Consultas

O método *loc* da classe Geo-objeto retorna uma lista de objetos geométricos da Localização de um geo-objeto, válidos no intervalo de tempo fornecido como parâmetro. Este método é composto de duas partes. A primeira determina se o geo-objeto foi válido durante o intervalo especificado. Em caso afirmativo, a segunda parte envia uma mensagem para o objeto Localização desse geo-objeto acionando o seu método *select_geometry*. O método *select_geometry* seleciona dentre as representações espaciais

da Localização, o objeto geometria adequado à escala e projeção do mapa atual, usando a função *right_scale*. Após isto, o método envia uma mensagem para o objeto geometria acionando o seu método *select_lst_objs*, que retorna uma lista de objetos geométricos válidos no intervalo de tempo especificado. Essa lista é devolvida para os métodos *select_geometry* e *loc*, que por sua vez a devolve para a consulta em execução (fig 5.4).

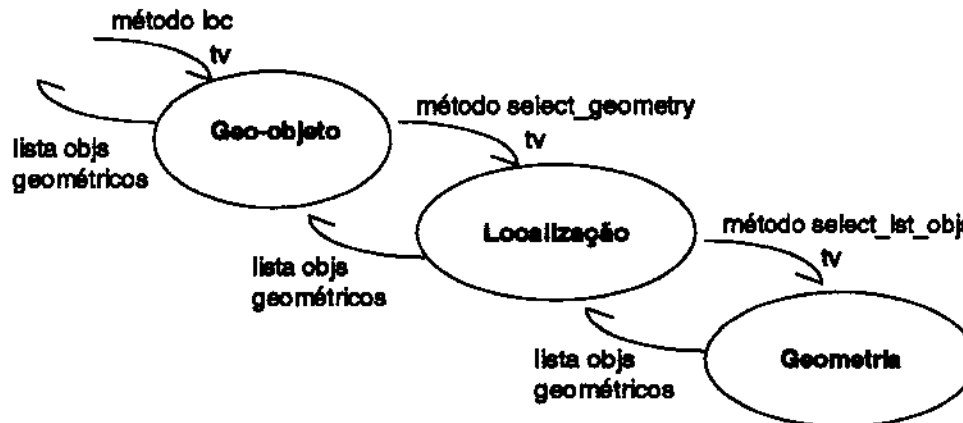


Fig 5.4 - Chamada ao Método Loc

A lista de objetos geométricos retornada pelo método *loc*, tem a seguinte estrutura:

```
list( tuple( lst_objs: list (Obj_Geom),
            t: list ( tuple (inicio: integer, fim: integer) ) ) )
```

Cada elemento da lista tem dois componentes: um atributo temporal (*t*) e um atributo espacial (*lst_objs*) que é a representação espacial da Localização do geo-objeto válida nos intervalos de tempo contidos em *t*. Note que o componente *lst_objs* é representado por uma lista de objetos geométricos, indicando que num instante de tempo um geo-objeto pode estar espacialmente representado por uma combinação de objetos geométricos.

As operações espaço-temporais que envolvem *apenas um geo-objeto* são processadas da seguinte forma (fig 5.5):

1. Os objetos geométricos de um geo-objeto válidos no intervalo especificado são recuperados e colocados em uma lista, através do método *loc* do geo-objeto.
2. A lista de objetos geométricos é processada pela operação espaço-temporal, calculando um resultado para cada elemento individual.
3. Uma lista com os resultados é retornada.

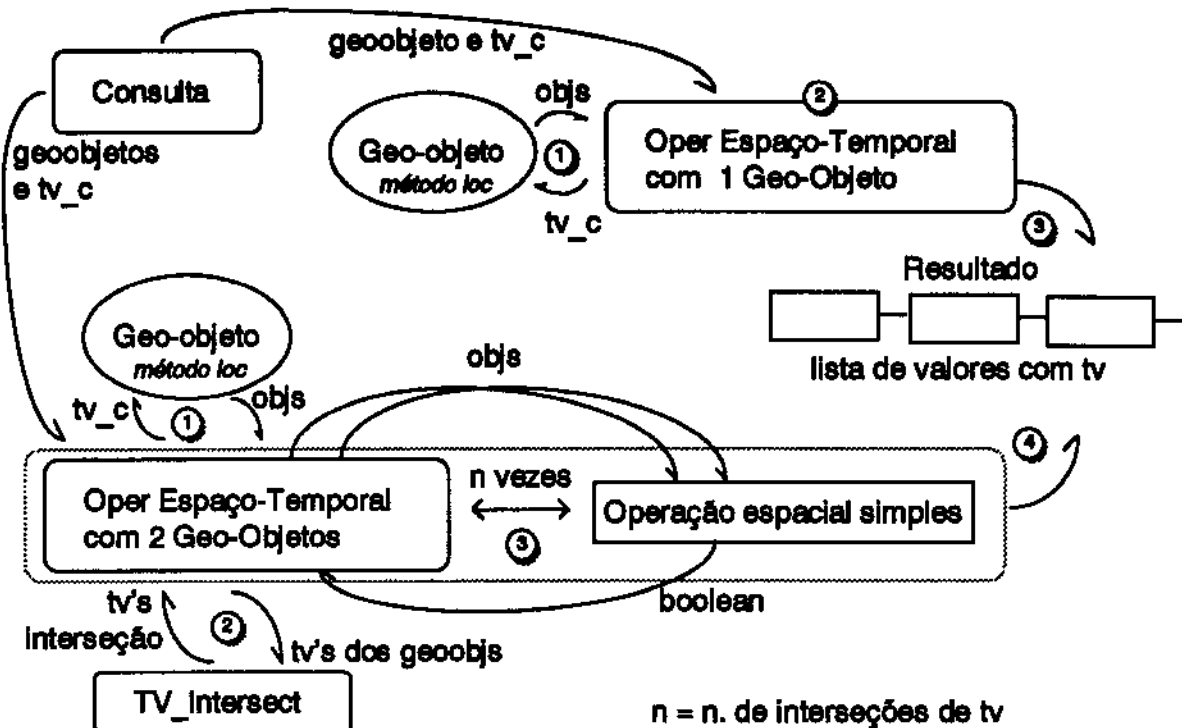


Fig 5.4 - Execução de uma Consulta

As operações espaço-temporais envolvendo *mais de um geo-objeto* exigem que se determine previamente os intervalos de tempo em que há interseção temporal desses geo-objetos. Para cada intervalo encontrado, a operação espaço-temporal é processada usando as n listas de objetos geométricos (correspondentes aos n geo-objetos envolvidos na operação) válidas nesse intervalo.

Seja, por exemplo, a seguinte variação de estados da Localização de dois geo-objetos ao longo do tempo:

loc.geo1 = lgeo1[1] [t1-t2), lgeo1[2] [t2-t5), lgeo1[3] [t5-t8)
 loc.geo2 = lgeo2[1] [t0-t1), lgeo2[2] [t1-t3), lgeo2[3] [t3-t6), lgeo2[4] [t6-t10)

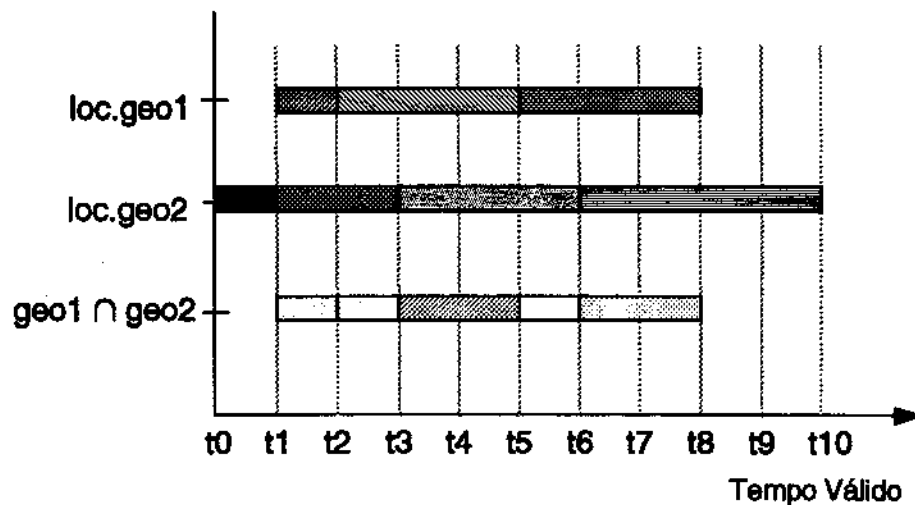


Fig 5.6 - Intervalos de Interseção

As fatias temporais sobre as quais se deve aplicar a operação espacial são (fig 5.6):

$[t1-t2): lgeo1[1] \text{ e } lgeo2[2], [t2-t3): lgeo1[2] \text{ e } lgeo2[2],$
 $[t3-t5): lgeo1[2] \text{ e } lgeo2[3], [t5-t6): lgeo1[3] \text{ e } lgeo2[3] \text{ e}$
 $[t6-t8): lgeo1[3] \text{ e } lgeo2[4].$

Para resolver este problema propomos o seguinte algoritmo para encontrar os intervalos de interseção temporal:

1. Coloca-se todos os valores (instantes de tempo) de início e fim de cada intervalo de tempo dos geo-objetos em uma lista única (os valores podem ser repetidos).
2. Ordena-se a lista em ordem crescente, onde os valores de fim antecedem os valores de início se forem iguais.
3. Percorre-se a lista sequencialmente contando o número de valores de início de intervalo (i) e também o número de valores de fim de intervalo (f).
4. Sempre que $i-f = n$, onde n é o número de geo-objetos envolvidos, coloca-se o intervalo do elemento atual até o próximo elemento da lista em uma outra lista.
5. A lista resultante contém os intervalos de interseção.

Por exemplo, a lista de intervalos ordenada (passo 2) do exemplo seria:

($t0_{2f}, t1_{2f}, t1_{1f}, t1_{2f}, t2_{1f}, t2_{1f}, t3_{2f}, t3_{2f}, t5_{1f}, t5_{1f}, t6_{2f}, t6_{2f}, t8_{1f}, t10_{2f}$)

Os arcos superiores representam os intervalos temporais do geo-objeto 2. Os arcos inferiores representam os intervalos temporais do geo-objeto 1. A primeira vez que $i-f = n$ ocorre no quarto elemento da lista. Portanto o primeiro intervalo de interseção é $[t1-t2]$. Continua-se a execução dos passos 3 e 4 até o fim da lista. A lista resultante contém todos os intervalos de interseção.

Portanto nas consultas espaço-temporais, as operações espaciais que envolvem mais de um geo-objeto são processadas da seguinte forma (fig 5.5):

1. Os objetos geométricos de cada geo-objeto válidos para o intervalo de tempo especificado são recuperados e colocados em uma lista através do método *loc*.
2. Os tempos-válidos dessas listas são processados pela função *t_intersect* que retorna os intervalos de interseção temporal da Localização desses geo-objetos.
3. Para cada intervalo de interseção, os objetos geométricos válidos nesse intervalo são processados pela operação espacial, calculando um resultado.
4. Uma lista dos resultados é retornada.

Se a consulta envolve mais de dois tipos de geo-objetos, e deseja-se verificar se existe um relacionamento espacial entre alguns deles (não entre todos), a complexidade aumenta ainda mais. Por exemplo, o usuário pode querer saber quando os rios intersectam fazendas ou regiões. Nessa dissertação não abordamos esses casos.

5.7 Classes Localização e Geometria

```
class Localização
    type localiz: list ( Geometria );

method select_geometry( tv_inicio: integer, tv_fim: integer );
    /* seleciona a representação adequada */
    resultado: list( tuple( lst_objs: list (Obj_Geom),
                           t: list ( tuple (inicio: integer, fim: integer) ) ) )
    geol: Geometria;
begin
    resultado = ( );
    for each geol in self.localiz do
        if right_scale(geol.atribs)
            resultado = geol → select_lst_objs( tv_inicio, tv_fim );
    return (resultado);
end
```

```
class Geometria
    type geom: tuple(
        atribs: [atributos não-espaciais temporalizados],
        objs: list( tuple( lst_objs: list (Obj_Geom),
                           t: list ( tuple (inicio: integer, fim: integer) ) ) )
    );
```



```

method select_lst_objs( tv_inicio: integer, tv_fim: integer );
    resultado: list( tuple( lst_objs: list (Obj_Geom),
                           t:list ( tuple (inicio: integer, fim: integer ) ) );
    lista_obj: tuple( lst_objs: list (Obj_Geom),
                     t:list ( tuple (inicio: integer, fim: integer ) ) );
    tempo: tuple (inicio: integer, fim: integer );
    overlap : boolean;
begin
    resultado = list ( );
    /* seleciona as versões da geometria que estão em tv */
    for each lista_obj in self.geom.objs do {
        overlap = FALSE;
        for each tempo in lista_obj.t do {
            if ((tv_inicio <= tempo.inicio) & (tempo.inicio <= tv_fim)) ||
                ((tv_inicio <= tempo.fim) & (tempo.fim <= tv_fim))
                overlap = TRUE; break; /* há overlap */
        }
        if overlap
            resultado += get_temporal_versions( lista_obj ); }
    return resultado;
end

```

A função `get_temporal_versions` deve retornar uma lista contendo todas as versões da geometria do geo-objeto armazenadas entre os componentes da lista de objetos geométricos passada como parâmetro.

5.8 Classes Geográficas

```

class GeoObjeto
type objeto: tuple(
    l: Localização,
    lst_geoobjs: list( tuple( lst_objs: list (GeoObjeto),
                           t:list ( tuple (inicio: integer, fim: integer ) ) ),
    t: list ( tuple (inicio: integer, fim: integer ) )
);

method loc(tv_inicio: integer, tv_fim: integer);
    resultado: list( tuple( lst_objs: list (Obj_Geom),
                           t:list ( tuple (inicio: integer, fim: integer ) ) );
    tmp: tuple (inicio: integer, fim: integer );
    geoobj_intersect: boolean;
begin
    resultado: list( );
    if fim == null      fim = inicio

```

```

for each tmp in self.objeto.t {
    if ((tv_início <= tmp.inicio) & (tmp.inicio <= tv_fim)) ||
        ((tv_início <= tmp.fim) & (tmp.fim <= tv_fim))
        geoobj_intersect = TRUE; break; }
    if geoobj_intersect
        resultado = self.objeto.l → select_geometry( tv_início, tv_fim ); }
    return (resultado);
end

```

```

class Divisão_agrícola inherit GeoObjeto;
type objetog: tuple(

    nome: list( tuple( lst_nome: string,
                      t:list ( tuple (inicio: integer, fim: integer) ) ) ),
    cultura: list( tuple( lst_cultura: string,
                        t:list ( tuple (inicio: integer, fim: integer) ) ) ),
    produção: list( tuple( lst_produção: real,
                          t:list ( tuple (inicio: integer, fim: integer) ) ) )
);

```

```

class Fazenda inherit GeoObjeto;
type objetog: tuple(

    nome: list( tuple( lst_nome: string,
                      t:list ( tuple (inicio: integer, fim: integer) ) ) ),
    proprietário: list( tuple( lst_proprietário: string,
                              t:list ( tuple (inicio: integer, fim: integer) ) ) ),
);

```

```

class Rio inherit GeoObjeto;
type objetog: tuple(

    nome: list( tuple( lst_nome: string,
                      t:list ( tuple (inicio: integer, fim: integer) ) ) ),
    vazão: list( tuple( lst_proprietário: real,
                      t:list ( tuple (inicio: integer, fim: integer) ) ) ),
);

```

```
class Reserva inherit GeoObjeto;
type objetog: tuple(

    nome: list( tuple( lst_nome: string,
                      t: list ( tuple (inicio: integer, fim: integer) ) ) ),
    espécie_existente: list( tuple( lst_espécie_existente: string,
                                   t: list ( tuple (inicio: integer, fim: integer) ) ) ),
    data_lei: Date,
    );
```

5.9 Estratégia para Armazenar e Acessar Dados Espaço-Temporais

Para [Lan95], a dominância dimensional da aplicação geográfica indica como os dados devem ser agrupados:

- As aplicações de *dominância temporal* devem agrupar o histórico de valores de cada dado (dados temporalmente agrupados) (fig 5.7a).
- As aplicações de *dominância espacial* devem agrupar dados de um mesmo estado temporal (dados temporalmente desagrupados) (fig 5.7b).
- As aplicações de *dominância espaço-temporal* devem agrupar os dados espacialmente próximos, com as suas versões temporais mais recentes (dados temporalmente semi-agrupados).

[Lan95] associa a dominância dimensional de uma aplicação geográfica usando o modelo relacional com o tipo de atualização empregada:

- A atualização *total* baseada em replicação de relação, favorece as aplicações geográficas com *dominância espacial*. Neste caso, cada tabela representa um estado temporal. Essa abordagem agiliza as consultas espaciais, mas é inviável para aplicações com muitos estados temporais, pois geraria um alto volume de dados com uma grande redundância de informações (fig 5.7a).
- A atualização *incremental* baseada na replicação dos atributos, favorece as aplicações geográficas com *dominância temporal*. Neste caso, cada atributo agrupa o seu histórico. Essa abordagem agiliza consultas temporais evitando as pesadas operações de junção de relações. [RY94] afirma que a replicação baseada em atributos é compacta em relação à replicação de tuplas ou tabelas, mas exige que os atributos tenham tamanho variável (N1FN). O modelo orientado a objetos permite a existência de

atributos multivalorados ou atributos com tamanho variável, sendo portanto mais adequado para armazenar dados temporais [Mul93, BVH95] (fig 5.7b).

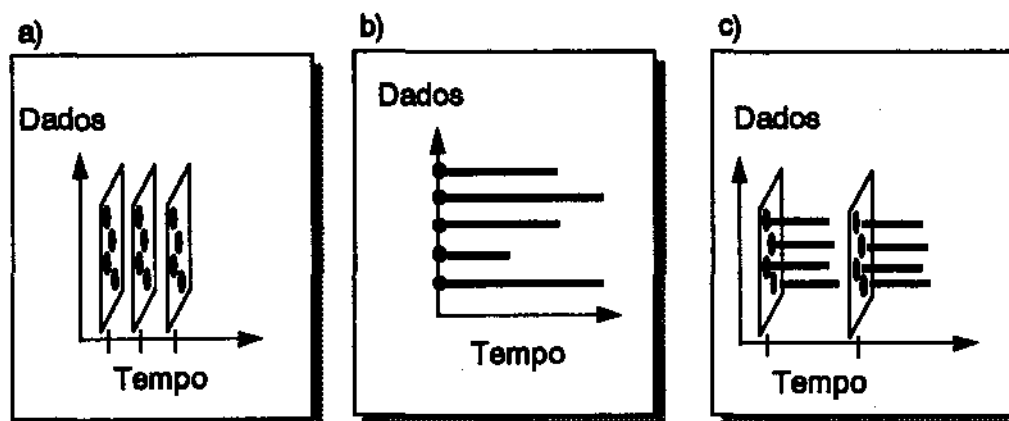


Fig 5.7 - Estratégias de Armazenamento

- A atualização *híbrida* (ou composta), ou seja, a atualização incremental periodicamente intercalada com a atualização total, favorece as aplicações geográficas com dominância *espaço-temporal*. Essa abordagem incorpora as vantagens das duas anteriores (fig 5.7c).

No nosso modelo, um mapa é representado por um geo-objeto temporal composto que contém uma lista de geo-objetos interrelacionados (por exemplo, rios, vegetação, países, dentre outros). Assim como em qualquer geo-objeto temporal composto, a inclusão ou exclusão de geo-objetos do mapa gera uma nova versão da sua lista de geo-objetos. O último estado temporal do mapa inteiro corresponde à versão mais recente da sua lista de geo-objetos e agrupa todos os geo-objetos contidos nele no intervalo de tempo $[x, \text{now}]$, onde x é o ponto de tempo em que houve a última mudança na sua lista de geo-objetos. Cada estado temporal do mapa contém todas as informações referentes aos seus componentes participantes daquele estado. Por outro lado, os geo-objetos contidos em um estado temporal do mapa agrupam o seu histórico de valores.

Pode-se abstrair este exemplo para qualquer geo-objeto temporal composto. A lista de geo-objetos é atualizada totalmente, e o conteúdo dos geo-objetos elementares é atualizado incrementalmente. Portanto o modelo desta dissertação assemelha-se bastante com a estratégia de atualização híbrida usando o paradigma de orientação a objetos.

5.10 Resumo

Este capítulo apresenta as estruturas de dados necessárias ao modelo de dados espaço-temporal descrito no capítulo 3. A partir destas estruturas, o capítulo mostrou como as consultas descritas pela taxonomia do capítulo 4 podem ser executadas. As

5. Estrutura dos Dados Espaço-Temporais

estruturas, portanto, além de corresponderem às necessidades do modelo, permitem a execução de qualquer consulta espaço-temporal.

Capítulo 6

Conclusões

Esta dissertação descreveu uma abordagem para a incorporar o suporte a dados espaço-temporais em sistemas gerenciadores de bancos de dados orientados a objetos para dados no formato vetorial. O modelo proposto foi baseado no modelo geográfico multi-nível proposto em [CCM+94] e no modelo temporal TOODM proposto em [Oli93], estendidos para suportar dados espaço-temporais na dimensão de tempo-válido. O modelo de dados proposto considera não só a visão histórica do mundo, como também a visão de processos. Ou seja, não somente os fatos geográficos são modelados, mas também os processos de *mudança* geográfica.

A dimensão tempo-válido é incorporada utilizando os construtores *tuple* e *list* de qualquer SGBDOO, através do processo de *temporalização* definido no capítulo 3. Este processo anexa atributos temporais ao esquema de uma classe geográfica tornando os seus geo-objetos aptos a armazenarem a sua evolução histórica. O modelo temporal proposto utiliza o *elemento temporal* como marca-de-tempo, por ser a mais compacta e intuitiva [TCG+95]. A temporalização é feita estendendo a proposta de [Bra94] (específica para dados tradicionais) à localização espacial de geo-objetos. Esta extensão é descrita conceitualmente e sob a forma de estrutura de classes em um BDOO.

Uma taxonomia de consultas geográficas espaço-temporais foi criada utilizando o conceito de dominância dimensional de [Lan95]. As consultas são divididas em consultas espaciais, temporais e espaço-temporais. Algumas operações atemporais de sistemas geográficos descritas em [Cif95] e [DV93] são mapeadas para a taxonomia proposta.

Além disso, vários exemplos de consultas geográficas são apresentados utilizando a linguagem de consulta TOOL de [Oli93]. As consultas utilizam várias operações espaço-temporais descritas em pseudo-código no anexo. O processamento interno dessas consultas é apresentado, mostrando todo o interrelacionamento entre os vários objetos geométricos e os objetos geográficos envolvidos durante a execução da consulta.

As principais contribuições apresentadas por este trabalho são:

- Um modelo de dados para armazenar a evolução temporal dos objetos geográficos foi apresentado. Além disso, uma proposta de representação para um processo geográfico também foi apresentada.
- Uma taxonomia de consultas geográficas foi proposta, abrangendo de uma forma simples e flexível as possíveis consultas geográficas espaciais, temporais e espaço-temporais.
- A estrutura de dados proposta utiliza construtores comuns a todos os bancos de dados orientados a objetos e a todas as linguagens orientadas a objetos. Isto possibilita a sua implementação em praticamente qualquer SGBDOO.
- Foi demonstrado como as estruturas de dados são utilizadas durante a execução de uma consulta espaço-temporal, através de operações espaço-temporais e dos métodos dos objetos envolvidos.

Um SIG que utilize um SGBDOO contendo as facilidades espaço-temporais apresentadas nesta dissertação não precisa gerenciar as estruturas de dados espaço-temporais, e nem implementar operações espaço-temporais básicas (como interseção, inclusão, distância, dentre outras), que estariam disponíveis a partir do SGBDOO.

As extensões a este trabalho podem ser tanto práticas quanto teóricas. Dentre as extensões teóricas, podemos citar:

- Estender o modelo geométrico criando novas classes geométricas para permitir a utilização de partes de uma linha como fronteira de um polígono (como por exemplo, *uma classe conexão* ou uma classe *linha direcionada*).
- Adicionar a coordenada de profundidade Z nos pontos de um geo-objeto para visualizar a Localização dos geo-objetos em suas formas tridimensionais, usando técnicas de interpolação para gerar inclinações de superfície. Assim, a Localização de um geo-objeto pode ter representações espaciais correspondentes a outros modelos topográficos de dados espaciais como o DEM (Data Elevation Model), DTM (Digital Terrain Model) e o TIN (Triangulated Irregular Network).
- Estender o modelo para suportar dados no formato *raster* (Geocampo).
- Incorporar ao sistema a possibilidade de evolução do esquema das classes geográficas.

As extensões práticas são voltadas à implementação do modelo e estruturas de dados propostas, bem como das operações espaço-temporais definidas na taxonomia. Além disso, outras extensões podem ser consideradas, visando otimização das estruturas e operações:

- Prover a possibilidade de armazenar a lista de geo-objetos de um geo-objeto composto implicitamente na forma de uma consulta. Por exemplo, seja um geo-objeto composto de todas as divisões agrícolas que plantam trigo. A lista de geo-objetos desse geo-objeto composto seria definida como: " `Divisão_agrícola.cultura=='Trigo'` ". Neste caso, a atualização da lista seria dinâmica. Este recurso seria útil para representar geo-objetos tipicamente compostos (como por exemplo um Estado, País, dentre outros).
- Utilizar estruturas topológicas para agilizar a execução de consultas espaciais múltiplas do tipo: "Selecione todos os rios que intersectam estradas"[dFM93]. Ao invés de processar individualmente cada par de instâncias, um algoritmo pode ser usado para procurar o resultado de uma forma mais global. Para isso, pode ser armazenado o *Minimum Bounding Rectangle* (MBR) [Cox91] do objeto geográfico. Sempre que é selecionado um novo mapa, ativando outros geo-objetos na tela, a matriz ordenada de MBR dos objetos deve ser atualizada na memória. Com isso, as consultas que envolvem proximidade espacial como interseção, inclusão e continência são bastante otimizadas.
- Implementar uma interface de consulta interativa que possibilitasse a reutilização de resultados de consultas parciais. O processo cognitivo humano é inerentemente interativo: tendemos a dividir grandes problemas em pequenas etapas de mais simples compreensão e portanto mais fáceis de serem resolvidas.

Anexo

As consultas espaço-temporais utilizam as operações espaço-temporais internas ao BD apresentadas a seguir:

1 Comprimento

Esta operação recebe como parâmetro um geo-objeto e um intervalo de tempo, e retorna o comprimento do perímetro de cada versão da geometria associada ao seu tempo válido. A função *dist* utilizada recebe dois pontos cartesianos como parâmetro e retorna a distância euclidiana entre eles. Os objetos do tipo ponto têm comprimento nulo.

```
function comprimento (o:GeoObjeto, inicio_s: integer, fim_s: integer );  
  
  lista_pontos: list (tuple ( x: Real, y: Real ));  
  p, p_antecessor: tuple ( x: Real, y: Real );  
  objeto: Obj_Geom;  
  geometria: list( tuple( lst_objs: list (Obj_Geom),  
                        t:list ( tuple ( inicio: integer, fim: integer ) ) ) );  
  resultado: list( tuple( comprimento: real, inicio: integer, fim: integer ) );  
  compr: real;  
  elem_geom = tuple( lst_objs: list (Obj_Geom),  
                    t:list ( tuple ( inicio: integer, fim: integer ) ) )  
  semaforo: boolean;
```

begin

```

if fim_s == 0
    fim_s = inicio_s
geometria = o→loc(inicio_s, fim_s);
for each elem_geom in geometria do {
    lista_pontos = ( );
    for each objeto in elem_geom.lst_objs do
        lista_pontos += objeto→vslice_s(inicio_s, fim_s);

    compr = 0;
    semaforo = FALSE;
    for each p in lista_pontos do {

        if (p == end_LINHA)
            semaforo = FALSE;
        if semaforo {
            if p_antecessor = 0 /* Se 1º ponto da linha */
                p_antecessor = p;
            else {
                compr = compr + dist( p, p_antecessor );
                p_antecessor = p;
            }
        }

        if (p == begin_LINHA) {
            semaforo = TRUE;
            p_antecessor = 0 }
        resultado += list( tuple( compr, elem_geom.t ) );
    }
    retorne resultado;
end

```

2 Área

Esta operação recebe como parâmetro um geo-objeto e um intervalo de tempo, e retorna a área de cada versão da geometria do geo-objeto associada ao seu tempo válido. Os objetos do tipo linha e ponto têm área nula.

```

function area (o:GeoObjeto, inicio_s: integer, fim_s: integer);

    lista_pontos: list (tuple ( x: Real, y: Real ));
    p, p_antecessor, p_primeiro: tuple ( x: Real, y: Real );
    geometria: list( tuple( lst_objs: list (Obj_Geom),
        t:list ( tuple ( inicio: integer, fim: integer ) ) ) );

```

```

resultado: list( tuple( area_geometria: real, inicio: integer, fim: integer ) );
area: real;
semaforo, semaforo2 : boolean;
elem_geom = tuple( lst_objs: list (Obj_Geom),
                    t:list ( tuple ( inicio: integer, fim: integer ) ) )
objeto: Obj_Geom;

begin
  if fim_s == 0
    fim_s = inicio_s
  geometria = o→loc(inicio_s, fim_s);
  for each elem_geom in geometria do {
    area = 0;
    lista_pontos = ( );
    for each objeto in elem_geom.lst_objs do
      lista_pontos += objeto→vslice_s(inicio_s, fim_s);
      semaforo = FALSE;
      semaforo2 = FALSE;
      for each p in lista_pontos do {
        if (p == end_POLIGONO)
          semaforo = FALSE;
        if semaforo {
          if (p == end_LINHA)
            semaforo2 = FALSE;
          if semaforo2 {
            if p_antecessor = 0      /* Se 1º ponto */
              p_antecessor = p;
            else { area += ( p.y * p_antecessor.x ) -
                      ( p.x * p_antecessor.y );
              p_antecessor = p;
            }
          }
          if (p == begin_LINHA) {
            semaforo2 = TRUE;
            p_antecessor = 0 }
        }
        if (p == begin_POLIGONO)
          semaforo = TRUE;
      }
      area = 1/2 * area;
      resultado += list( tuple( area, elem_geom.f ) );
    }
  retorne resultado;
end

```

3 Interseção

Esta operação recebe como parâmetro dois geo-objetos e um intervalo de tempo, e retorna uma lista contendo um valor lógico para cada intervalo de interseção temporal. O algoritmo utiliza a função *intersect* que a partir de duas listas de pontos, retorna um valor lógico indicando se existe interseção.

function *interseção* (*o1*:GeoObjeto, *o2*:GeoObjeto, *inicio_s*: integer, *fim_s*: integer);

```

    elem_geom = tuple( lst_objs: list (Obj_Geom),
                      t:list ( tuple (inicio: integer, fim: integer ) ) )
    objeto: Obj_Geom;
    lista_pontos_1, lista_pontos_2: list (tuple ( x: Real, y: Real ));
    geometria1, geometria2: list( tuple( lst_objs: list (Obj_Geom),
                                         t:list ( tuple (inicio: integer, fim: integer ) ) ));
    resultado: list( tuple( intersecao_1: boolean, inicio: integer, fim: integer ) );
    intervalos_intersecao: list( tuple(inicio_t: integer, fim_t: integer ) );
    intervalo: tuple(inicio_t: integer, fim_t: integer);
begin
    if fim_s == 0
        fim_s = inicio_s
    geometria1 = o1→loc(inicio_s, fim_s);
    geometria2 = o2→loc(inicio_s, fim_s);
    intervalos_intersecao = tv_intersect( geometria1, geometria2, inicio_s, fim_s );
    for each intervalo in intervalos_intersecao do {
        for each elem_geom in geometria1 do
            for each objeto in elem_geom.lst_objs do {
                lista_pontos_1 += objeto→vslice_s(intervalo.inicio_t, intervalo.fim_t);
            }
        for each elem_geom in geometria2 do
            for each objeto in elem_geom.lst_objs do {
                lista_pontos_2 += objeto→vslice_s(intervalo.inicio_t, intervalo.fim_t);
            }
        resultado += list( tuple(intersect(lista_pontos_1, lista_pontos_2),
                                   intervalo.inicio_t, intervalo.fim_t ) );
    }
    retorne resultado;
end
```

4 Distância

Esta operação recebe como parâmetro dois geo-objetos e um intervalo de tempo, e retorna uma lista contendo a distância entre as geometrias dos geo-objetos para cada intervalo de interseção temporal. A função *centroid* retorna o centro geográfico (ou o ponto centróide) a partir de uma lista de pontos que representa uma geometria.

```
function distância (o1:GeoObjeto, o2:GeoObjeto, inicio_s: integer, fim_s: integer);
    elem_geom = tuple( lst_objs: list (Obj_Geom),
                      t:list ( tuple (inicio: integer, fim: integer ) ) )
    objeto: Obj_Geom;
    distancia: real;
    lista_pontos_1, lista_pontos_2: list (tuple ( x: Real, y: Real ));
    geometria1, geometria2: list( tuple( lst_objs: list (Obj_Geom),
                                         t:list ( tuple (inicio: integer, fim: integer ) ) ));
    resultado: list( tuple( d: real, inicio: integer, fim: integer ) );
    intervalos_intersecao: list( tuple(inicio_t: integer, fim_t: integer) );
    intervalo: tuple(inicio_t: integer, fim_t: integer);
begin
    if fim_s == 0
        fim_s = inicio_s
    geometria1 = o1→loc(inicio_s, fim_s);
    geometria2 = o2→loc(inicio_s, fim_s);
    intervalos_intersecao = tv_intersect( geometria1, geometria2, inicio_s, fim_s );
    for each intervalo in intervalos_intersecao do {
        for each elem_geom in geometria1 do
            for each objeto in elem_geom.lst_objs do {
                lista_pontos_1 += objeto→vslice_s(intervalo.inicio_t, intervalo.fim_t);
            }
        for each elem_geom in geometria2 do
            for each objeto in elem_geom.lst_objs do {
                lista_pontos_2 += objeto→vslice_s(intervalo.inicio_t, intervalo.fim_t);
            }

        distancia = dist(centroid(lista_pontos_1), centroid(lista_pontos_2 ) );
        resultado += list( tuple(distancia, intervalo.inicio_t, intervalo.fim_t ) );
    }
    retorne resultado;
end
```

5 Inclusão

Uma geometria está incluída num polígono se não há interseção entre eles, e se existe um ponto de cada objeto geométrico da geometria que está incluído no polígono. A operação inclusão recebe como parâmetro dois geo-objetos e um intervalo de tempo, e retorna uma lista contendo um valor lógico para cada intervalo de interseção temporal. A função *p_in* retorna TRUE somente se o ponto está incluído no polígono. A função *mid* retorna um ponto central de uma lista de pontos. A função *all_false* retorna TRUE somente se todos os valores lógicos do resultado de uma operação espacial forem FALSE.

```
function inclusão ( o1:GeoObjeto, o2:GeoObjeto, inicio_s: integer, fim_s: integer );
    elem_geom = tuple( lst_objs: list (Obj_Geom),
        t:list ( tuple (inicio: integer, fim: integer ) ) )
    geometria1, geometria2: list( tuple( lst_objs: list (Obj_Geom),
        t:list ( tuple (inicio: integer, fim: integer ) ) ) );
    objeto: Obj_Geom;
    intervalos_intersecao: list( tuple(inicio_t: integer, fim_t: integer) );
    intervalo: tuple(inicio_t: integer, fim_t: integer);
    resultado: list( tuple( inclusao_l: boolean, inicio: integer, fim: integer) );
begin
    inclusão_l = FALSE;
    if fim_s == 0
        fim_s = inicio_s
    geometria1 = o1→loc(inicio_s, fim_s);
    geometria2 = o2→loc(inicio_s, fim_s);
    intervalos_intersecao = tv_intersect( geometria1, geometria2, inicio_s, fim_s );

    for each intervalo in intervalos_intersecao do {
        if all_false( interseção(o1, o2, intervalo.inicio_t, intervalo.fim_t) ){
            inclusão_l = TRUE;
            lista_pontos = ( );
            for each elem_geom in geometria1 do
                for each objeto in elem_geom.lst_objs do {
                    lista_pontos = objeto→vslice_s(intervalo.inicio_t, intervalo.fim_t);
                    if not p_in(mid(lista_pontos),geometria2, intervalo.inicio_t, intervalo.fim_t)
                        inclusão_l = FALSE;
                }
            }
        }
        resultado += list( tuple( inclusao_l, intervalo.inicio_t, intervalo.fim_t ) );
    }
    retorne resultado;
end
```

6 Continência

Um polígono contém uma geometria se não há interseção entre eles e se o polígono contém um ponto de cada objeto geométrico da geometria. Esta função é dual da função de inclusão, e portanto a sua implementação consiste em chamar a função de inclusão com os parâmetros invertidos. A função recebe como parâmetro dois geo-objetos e um intervalo de tempo, e retorna uma lista contendo um valor lógico para cada intervalo de interseção temporal das suas geometrias.

```
function continencia ( o1: GeoObjeto, o2: GeoObjeto, inicio_s: integer, fim_s: integer );
```

```
resultado: list( tuple( continencia_l: boolean, inicio: integer, fim: integer ) );
```

```
begin
```

```
    resultado = inclusao(o2, o1, inicio_s, fim_s);
```

```
retorne resultado;
```

```
end
```

Referências

- [ABCK91] J. Antenucci, K. Brown, P. L. Croswell, M. J. Kevany. *Geographic Information Systems: a guide to technology*. Van Nostrand Reinhold. New York, 1991.
- [Arg95] Argus. The evolution of geographic information systems. Technical Report, Argus Technologies Corporation, 1995.
- [Aro89] S. Aronoff. *Geographic information systems: a management perspective*. WDL Publications. Ottawa, Ontario, Canada, 1989.
- [ASS93] K. K. Al-Taha, R. T. Snodgrass and M. D. Soo. Bibliography on Spatiotemporal Databases. *SIGMOD RECORD*, 22(1):59-67, March 1993.
- [BDK92] F. Bancilhon, C. Delobel, and P. Kanellakis. *Building an object-oriented database system: the story of O2*. Morgan Kaufman Pub, 1992. 625 pages.
- [BF94] F. Bancilhon and G. Ferran. ODMG-93: The object database standard. in *Data Engineering*, IEEE Computer Society, 17(4):3-14, December 1994.
- [Bra94] A. R. Brayner. Implementação de um sistema temporal em um banco de dados orientado a objetos. Master's thesis, Universidade Estadual de Campinas / UNICAMP, 1994.
- [BVH95] L. Becker, A. Voigtman, and K. Hinrichs, Towards an object-oriented geographic data model. Universidade de Münster, 1995.
- [CCM+94] G. Camara, M. A. Casanova, C. B. Medeiros, M. Souza and U. Freitas. A model to cultivate objects and manipulate fields. In *Proc Proceedings of the Second ACM Workshop on Advances in GIS*, pages 20-28, 1994.
- [CL93] J. Cheylan and S. Lardon. Towards a conceptual data model for the analysis of spatio-temporal processes: the example of the search for optimal grazing strategies. In *Advances in Spatial Databases*, number 716 in Lecture Notes in Computer Science, pages 113-138. Springer Verlag, September 1993.

- [Cli94] W. Clinton. Coordinating geographic data acquisition and access: the national spatial data infrastructure (NSDI). In *The Federal Register*, 59(71):17671-17674. Executive Order 12906, April 13th 1994.
- [Cif95] R. R. Ciferri. Um Benchmark voltado para a análise de desempenho de um SIG. Master's thesis, Universidade Estadual de Campinas / UNICAMP, 1995.
- [Cox91] F. S. Cox. Análise de métodos de acesso a dados espaciais aplicados a sistemas gerenciadores de bancos de dados. Master's thesis, Universidade Estadual de Campinas / UNICAMP, 1991.
- [dFM93] L. de Floriani and P. Marzano. Spatial queries and data models. In *Advances in Spatial Databases*, number 716 in Lecture Notes in Computer Science, pages 113-138. Springer Verlag, September 1993.
- [Dit94] K. R. Dittrich. Object-oriented database management: unfinished variations on a popular theme. PUC-Rio DB Workshop, Rio de Janeiro, September, 1994.
- [DKL+94] D. J. DeWitt, N. Kabra, J. Luo, J. M. Patel, and J. Yu. Client-Server Paradise. In *Proc of the 20th VLDB Conference*, Santiago, Chile, 1994.
- [DRSM93] B. David, L. Raynal, G. Schorter, and V. Mansart. Geo2: Why objects in a geographical DBMS. In *Advances in Spatial Databases*, number 692 in Lecture Notes in Computer Science, pages 264-276. Springer Verlag, June 1993.
- [DV93] B. David and A. Voisard. A unified approach to geographic data modelling. Technical Report, Ludwig-Maximilians-Universität München, Institut für Informatik, September 1993.
- [Ege92] M. Egenhofer. Why not SQL. *International Journal of Geographical Information Systems*, 6(2):71-86, March - April 1992.
- [Esr92] ESRI. ARC/INFO: Gis today and tomorrow. *ESRI White Paper Series*, ESRI Inc., Redlands, California, March 1992.
- [FG90] A. U. Frank and M. F. Goodchild. Two perspectives on geographical data modelling. Technical Report 90-11, National Center for Geographic Information & Analysis (NCGIA), November 1990.
- [FL95] K. E. Foote and M. Lynch. Geographic information systems as an integrating technology: context, concepts, and definitions. *The Geographer's Craft Project*, Department of Geography, University of Texas, Austin, USA, 1995.
- [FR93] A. S. Fotheringham and P. A. Rogerson. GIS and spatial analytical problems. *International Journal of Geographical Information Systems*, 7(1):3-19, January - February 1993.

- [Fra90] A. U. Frank. Spatial concepts, geometric data models and data structures. Technical Report 90-11, National Center for Geographic Information & Analysis (NCGIA), November 1990.
- [Gat91] A. C. Gatrell. *Concepts of space and geographical data*, chapter 9, pages 119-134. Volume 1 of Maguire et al. [MGR91], Longman Scientific & Technical, 1991.
- [GC93] S. K. Gadia and V. Chopra. A relational model and SQL-like query language for spatial databases. In *Advances in Spatial Databases*, number 759 in Lecture Notes in Computer Science, pages 213-225. Springer Verlag, November 1993.
- [Her94] L. H. Hermosilla. A unified approach for developing a temporal GIS with database and reasoning capabilities. In *Proc of the EGIS'94*. Paris, France, March 1994.
- [HOM93] S. Hoop, P. Oosterom, and Martien Molenaar. Topological Querying of Multiple Map Layers. In *Advances in Spatial Databases*, number 716 in Lecture Notes in Computer Science, pages 113-138. Springer Verlag, September 1993.
- [JCE+94] C. S. Jensen, J. Clifford, T. Elmasri, S. K. Gadia, P. Hayes, and S. Jajodia. A consensus glossary of temporal database concepts. *SIGMOD RECORD*, 23(1):52-64, March 1994.
- [Lan93] G. Langran. Issues of implementing a spatiotemporal system. *International Journal of Geographical Information Systems*, 7(4):305-314, October - December 1993.
- [Lan95] G. Langran. *Time in geographical information systems*. Taylor Francis. Bristol, Pensilvania, USA, 1995.
- [Low94] K. E. Lowell. Probabilistic temporal GIS modelling involving more than two map classes. *International Journal of Geographical Information Systems*, 8(2):73-93, March - April 1994.
- [MGR91] D. J. Maguire, M. F. Goodchild, D. W. Rhind. *Geographical Information Systems, Principles and Applications*. Longman Scientific & Technical, 1991.
- [MMS93] P. Milne, S. Milton, and J. L. Smith. Geographical object-oriented databases - a case study. *International Journal of Geographical Information Systems*, 7(1):39-55, January-February 1993.
- [MM94] F. Manola and G. Mitchell. A comparison of object models in odbms-related standards. in *Data Engineering*, IEEE Computer Society, 17(4):3-14, December 1994.

- [MJ93] C. B. Medeiros and G. Jomier. Managing alternatives and data evolution in GIS. in *ACM/ISCA Workshop on advances in geographic information systems*, Baltimore, USA, November 1993.
- [MJ94] C. B. Medeiros and G. Jomier. Using versions in GIS. In *Proc ACM SIGMOD Record*, 1(23):107-115, March 1994.
- [MPB93] C. B. Medeiros, F. Pires, and A. Barros. Modelling geographic information systems using an object oriented framework. In *Proc XIII International Conference of the Chilean Computer Science Society*, pages 217-232, 1993.
- [MP94] C. B. Medeiros and F. Pires. Databases for GIS. In *Proc ACM SIGMOD Record*, 1(23):107-115, March 1994.
- [Mul93] J. C. Muller, Latest developments in GIS/LIS, *International Journal of Geographical Information Systems*, 7(4):293-304, October - December 1993.
- [Mur95] D. Murray. SAIF: an object oriented archival and interchange format for the future. Safe Software Inc., Surrey, BC, Canada, 1995.
- [Neu89] L. Neugebauer. Extending a database to support the handling of enviromental measurement data. In *Design and Implementation of Large Spacial Databases*, number 409 in Lecture Notes in Computer Science, pages 147-166. Springer Verlag, June 1989.
- [NS93] P. Nijkamp and H. J. Scholten. Spatial information systems: design, modelling, and use in planning. *International Journal of Geographical Information Systems*, 7(1):3-19, January-February 1993.
- [O2t93] O2Technology. The O2 user manual, version 4.3. Technical Report, O2Technology, Versailles, France, 1993.
- [OM93] L. M. Oliveira and C. B. Medeiros. Managing time in object-oriented databases. Technical Report DCC 14/93, Universidade Estadual de Campinas / UNICAMP, July 1993.
- [Oli93] L. C. Medina Oliveira. Incorporação da dimensão temporal em bancos de dados orientados a objetos. Master's thesis, Universidade Estadual de Campinas / UNICAMP, 1993.
- [OvdB89] P. V. Oosterom and J. van den Bos. An object-oriented approach to the design of geographic information systems. In *Design and Implementation of Large Spacial Databases*, number 409 in Lecture Notes in Computer Science, pages 255-270. Springer Verlag, June 1989.

- [PSE+94] N. Pissinou, T. R. Snodgrass, R. Elmasri, I. S. Mumick, M. T. Ozsu, B. Pernici, S. Arie, B. Theodoulidis, and U. Dayal. Towards an infrastructure for temporal databases: Report of an invitation arpa/nsf workshop. *SIGMOD RECORD*, 23(1):35-51, March 1994.
- [Peu94] D. J. Peuquet. A conceptual framework for the representation of temporal dynamics in GIS. *Annals of the Association of American Geographers*, September, 1994.
- [Pri89] S. Price. Modelling the temporal element in land information systems. *International Journal of Geographic Information Systems*, 3(3):233-243, July - September 1989.
- [RY94] H. Raafat and Z. Yang. Relational spatial topologies for historical geographical information, *International Journal of Geographical Information Systems*, 8(2):163-173, March-April 1994.
- [SAA+94] T. R. Snodgrass, I. Ahn, G. Ariav, D. Batory, J. Clifford, C. E. Dyreson, R. Elmasri, F. Grandi, C. S. Jensen, W. Kafer, N. Kline, K. Kulkarni, T. Y. C. Leung, N. Lorentzos, J. F. Roddick, A. Segev, M. D. Soo, and S. M. Stripada. TSQL2 language specification. *SIGMOD RECORD*, 23(1):65-86, March 1994.
- [SV92] M. Scholl and A. Voisard. *Geographic Applications: an experience with O2*, chapter 28, pages 585-618. Bancelhon et al. [BDK92], Morgan Kaufman Pub, 1992.
- [Sno86] T. R. Snodgrass. Temporal databases. University of North Carolina at Chapel Hill Illsoo Ahn, AT&T Bell Laboratories, 1986.
- [Sno92] T. R. Snodgrass. Temporal databases. Department of Computer Science, University of Arizona, Tucson, AZ, 1992.
- [Sno95] T. R. Snodgrass. Temporal object-oriented databases: a critical comparison. chapter 19, pages 386-408. In *Next Generation Database Technology / Part 1*. 1995.
- [Ste94] J. Steinberg. Desktop mapping in Windows era. *Business Geographic*, 2(6):8-11, *Gis World*, 1994.
- [TCG+95] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass. *Temporal databases: theory, design, and implementation*, The Benjamin/Cummings Publishing Company Inc., 1995.
- [WHM90] M. F. Worboys, H. M. Hearnshaw, and D. J. Maguire. Object oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, 4(4):369-383, October - December 1990.

- [Whi93] P. A. Whigham. Hierarchies of Space and Time. In *Advances in Spatial Databases*, number 716 in Lecture Notes in Computer Science, pages 190-199. Springer Verlag, September 1993.