

Um Ambiente Computacional de Modelagem para Aplicações Ambientais

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Maria de Fátima R. O. Pires da Silva e aprovada pela Banca Examinadora.

Campinas, 9 de dezembro de 1997.


Prof. Dra. Claudia Bauzer Medeiros
(Orientadora)

Tese apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.



UNIDADE	BC
N.º CHAMADA:	Unicamp
V.	Si38a
TOMBO	34882
PROG.	395198
PREÇO	R\$ 11,00
DATA	02/09/98
N.º CPD	

CM-00115514-6

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Silva, Maria de Fátima R.O. Pires da

Si38a Um ambiente computacional de modelagem para aplicações ambientais / Maria de Fátima R.O. Pires da Silva -- Campinas, [S.P. :s.n.], 1997.

Orientador : Claudia Bauzer Medeiros

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas de informações geográficas. 2. Banco de dados orientado a objetos. 3. Engenharia de software. 4. Projeto de sistemas.

I. Medeiros, Claudia Bauzer. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

**Um Ambiente Computacional de Modelagem
para Aplicações Ambientais**

Maria de Fátima R. O. Pires da Silva

Tese de Doutorado

Um Ambiente Computacional de Modelagem para Aplicações Ambientais

Maria de Fátima R. O. Pires da Silva¹

Dezembro de 1997

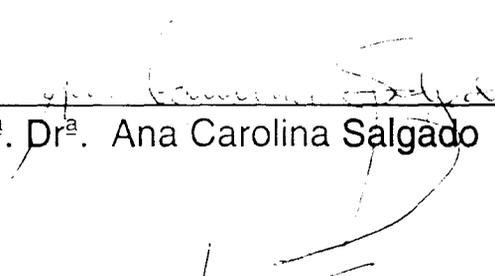
Banca Examinadora:

- Prof. Dra. Claudia Bauzer Medeiros (Orientadora)
- Prof. Dra. Ana Carolina Salgado
Departamento de Informática – UFPE
- Prof. Dr. José Carlos Maldonado
ICMSC – USP
- Prof. Dr. Geovane Magalhães
Instituto de Computação – UNICAMP
- Prof. Dra. Eliane Martins
Instituto de Computação – UNICAMP
- Prof. Dra. Rozely F. dos Santos
Faculdade de Engenharia Civil – UNICAMP (Suplente)
- Prof. Dra. Maria Cecilia Baranauskas
Instituto de Computação – sc Unicamp (Suplente)

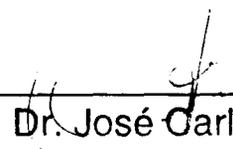
¹Este trabalho foi realizado com suporte dos projetos Protem GEOTEC do CNPq e ITDC GEOTOOLS da Comunidade Européia.

© Maria de Fátima R. O. Pires da Silva, 1998.
Todos os direitos reservados.

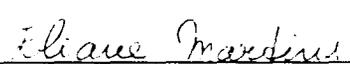
Tese de Doutorado defendida e aprovada em 09 de dezembro de 1997 pela Banca Examinadora composta pelos Professores Doutores



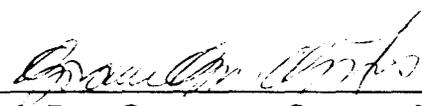
Prof^a. Dr^a. Ana Carolina Salgado



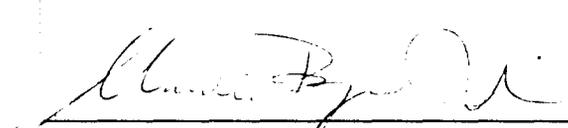
Prof. Dr. José Carlos Maldonado



Prof^a. Dr^a. Eliane Martins



Prof. Dr. Geovane Cayres Magalhães



Prof^a. Dr^a. Claudia Maria Bauzer Medeiros

Dedicatória

Dedico este trabalho a minha mãe, meu marido e meus filhos.

Sumário

Aplicações geográficas são inerentemente complexas devido a natureza dos dados que manipulam e dos processos que atuam sobre esses dados.

Atualmente, muitas dessas aplicações estão construídas sobre SIG (Sistemas de Informações Geográficas), que provêem mecanismos de armazenamento, análise e apresentação de dados geográficos. No entanto, SIG apresentam limitações ao uso de toda a sua potencialidade. Estas limitações referem-se à sua interface e facilidade de modelagem e ao perfil do usuário que são especialistas na sua área de domínio e não têm formação adequada em Engenharia de Software e projeto de banco de dados.

A tese é uma contribuição à solução dessas duas limitações, oferecendo ao usuário um ambiente de modelagem de aplicações geográficas, o UAPE. Utilizando o ambiente, os usuários serão capazes de projetar aplicações de acordo com suas necessidades, sem ter que aprender os conceitos de implementação do SIG subjacente.

As principais contribuições deste trabalho são:

- um *modelo* de dados geográficos, GMOD, que modela aspectos estáticos e dinâmicos do mundo real,
- uma *metodologia* de especificação e projeto de aplicações geográficas para planejamento ambiental,
- o *ambiente* UAPE, que integra modelo e metodologia de forma a assistir o usuário na tarefa de projetar suas aplicações.

Abstract

Geographic applications are intrinsically complex due to the nature of the data manipulated and also due to the processes acting over these data.

Today, many of these applications are built on top of a GIS (Geographic Information System), a software that provides efficient storage, analysis and presentation tools for spatial data. Nevertheless, GISs present limitations that prevent users from taking full advantage of available GIS tools. These limitations are mainly related to their interface and modeling features and also to the fact that end-users are experts in their application domain but do not have the adequate background in Software Engineering or database design.

This thesis is a contribution to solve these two limitations, presenting UAPE - an environment for modeling and designing geographic applications. With the environment, users will be able to design applications according to their needs, abstracting the implementation details related to the underlying GIS.

The major contributions of this thesis are:

- an object oriented *model*, GMOD, which supports both data and process modeling,
- a *methodology* for environmental application design,
- an *environment*, UAPE, that integrates model and methodology in order to help users in environmental application modeling and design.

Agradecimentos

Aos meus colegas do grupo de banco de dados do IC pelo apoio e colaboração na troca de idéias, em especial ao Rodrigo Macedo de Azevedo e Marcos Antônio Vaz Salles pela implementação do protótipo.

Aos meus professores do IC por despertar e desenvolver a minha capacidade de pesquisa, em especial ao Prof. Dr. Nelson Castro Machado, Prof. Dr. Tomasz Kowaltowski, Prof. Dr. Claudio Lucchesi e em especial à minha orientadora Prof. Dra. Claudia Bauzer Medeiros pela confiança e incentivo constantes.

Aos meus colegas "usuários" de aplicações geográficas, em especial Profa. Dra. Rozely dos Santos, Prof. Dr. Jansle Vieira Rocha, Prof. Gianpaolo Pellegrino, Dr. Jurandir Zullo Jr. e Prof. Dr. Carlos Alberto Vettorazzi pela valiosa colaboração na fundamentação das idéias.

Ao meu marido Reynaldo e meus filhos Isabela e Reinho pela paciência e compreensão ao longo do desenvolvimento do trabalho.

À minha mãe, grande amiga e companheira.

Conteúdo

Dedicatória	v
Sumário	vi
Abstract	vii
Agradecimentos	viii
1 Introdução e Motivação	1
2 Terminologia e Conceituação Básica	5
2.1 Desenvolvimento de Software	5
2.2 Modelagem Estática, Dinâmica e de Processos	7
2.3 Aplicações Geográficas	8
2.4 Sistemas de Informações Geográficas	10
2.5 Conceitos Básicos de Orientação a Objetos	12
2.6 Resumo	12
3 Modelagem de Dados Geográficos	13
3.1 Modelagem de Fenômenos Geográficos	13
3.1.1 Modelagem espaço-temporal integrada	15
3.1.2 Modelos com representações múltiplas	15
3.1.3 Modelo para SIG usando sistemas especialistas	17
3.1.4 GeO2	17
3.2 Ambientes de Modelagem	18
3.3 Resumo	20
4 Modelo de Dados Geográficos - GMOD	21
4.1 Introdução	21
4.2 Modelagem Estática	24
4.2.1 Classes Básicas	24

4.2.2	Relacionamentos no GMOD	31
4.3	Modelagem Dinâmica	36
4.4	Modelagem de Processos	40
4.5	Operações no GMOD	41
4.6	Resumo	43
5	Metodologia de Desenvolvimento de Aplicações Ambientais	44
5.1	Contexto	44
5.2	Concepção de Aplicações Geográficas	46
5.3	Exemplo de uso da metodologia	51
5.4	Resumo	54
6	Ambiente UAPÉ	55
6.1	Visão Geral	55
6.2	Concepção de Aplicações Geográficas sob o UAPÉ	59
6.3	Arquitetura Conceitual do Ambiente	64
6.3.1	Interface do Usuário	66
6.3.2	Módulo de Modelagem e Projeto	66
6.3.3	Módulo de Consulta e Manipulação	68
6.3.4	Módulo da Metodologia	69
6.3.5	Módulo de Exportação	69
6.3.6	Módulo de Importação	70
6.4	O BDUAPE	70
6.5	Resumo	71
7	Estudo de Caso - Controle de Riscos de Incêndio	74
7.1	Definição do Projeto	75
7.2	Modelagem	75
7.2.1	Classes	75
7.2.2	Relacionamentos	75
7.2.3	Planos de Informação	76
7.2.4	Modelagem Dinâmica	76
7.2.5	Modelagem de Processos	76
7.2.6	Diagrama de Classes	77
7.2.7	Esquema O_2 da Aplicação	78
7.3	Implementação	81
7.4	Avaliação	81
7.5	Resumo	81

8	Considerações Finais	82
8.1	Conclusões e Contribuições	82
8.2	Algumas Limitações da Proposta	83
8.3	Extensões	84
A	Apêndice A	86
A.1	Modelo de Objetos	86
A.2	Esquema do BDUAPE	87
	Bibliografia	96

Lista de Figuras

2.1	Fases Genéricas do Desenvolvimento de Software	6
3.1	Quadro Comparativo entre os Modelos	18
4.1	Modelo de Dados - GMOD	25
4.2	Exemplo de Geo-Campo Temático	27
4.3	Exemplo de Geo-Campo MNT	27
4.4	Exemplo de Geo-Campo Sensoriamento Remoto	28
4.5	Classes de Representação	29
4.6	Hierarquia de Classes Temporais	31
4.7	Generalização na Visão de Objetos	33
4.8	Generalização na Visão de Campos	34
4.9	Agregação em Aplicações Geográficas	34
4.10	Associações Simples entre Geo-Classes	35
4.11	Associação - Versão	36
4.12	Relacionamento Causal	38
4.13	Exemplo de Diagrama de Estados	39
4.14	Exemplo de Processo - Cálculo da Umidade do Solo	42
5.1	Metodologia para Aplicações Ambientais	47
6.1	Arquitetura de Banco de Dados do UAPÉ	56
6.2	GMOD com as Classes de Suporte ao Projeto	58
6.3	Metodologia para Aplicações Ambientais sob o UAPÉ	62
6.4	Relacionamento entre as Fases das Metodologias	64
6.5	Arquitetura do UAPÉ	65
6.6	Esquema de Operação do Módulo de Exportação	69
6.7	Esquema de Operação do Módulo de Importação	70
7.1	Derivação do GMOD para o Estudo de Caso	77
7.2	Derivação do GMOD - Geo-campo	77

7.3	Derivação do GMOD - Geo-objeto	78
7.4	Derivação do GMOD para uma aplicação	79
7.5	Consulta ao BDImp para reutilização de dados	80
A.1	Modelo de Objetos do UAPÉ	87

Capítulo 1

Introdução e Motivação

Esta tese apresenta um ambiente de apoio à modelagem e ao projeto de aplicações ambientais, visando à auxiliar usuários não especialistas em computação a projetar e manipular bancos de dados geo-referenciados. Este ambiente, denominado UAPÉ, engloba um modelo orientado a objetos de dados e processos para criação do banco de dados e uma metodologia especialmente desenvolvida para o projeto dessas aplicações. O ambiente foi projetado de forma a ser aberto e independente do SIG a ser utilizado.

SIG (Sistemas de Informações Geográficas) são sistemas de captação, armazenamento, manipulação, análise e apresentação de dados geo-referenciados [Aro89]. O número e variedade de domínios de aplicações que utilizam SIG têm crescido exponencialmente devido aos avanços das tecnologias de suporte e diminuição dos custos de hardware. A motivação para este trabalho é que, apesar desta crescente demanda, SIG ainda não provêem um ambiente adequado para os usuários finais. Mais especificamente, os SIG atuais disponibilizam, por exemplo, operações eficientes de armazenamento e consulta sobre dados geográficos, mas têm sérias limitações no suporte a atividades básicas como modelagem e especificação do ponto de vista do usuário. Assim, apesar de seu potencial, SIG ainda são utilizados de forma incipiente por diversas razões [NCG89]:

- *ausência de metodologia de planejamento*: não existem métodos padrão que auxiliem o usuário na determinação precisa de quais dados devem ser coletados para a obtenção das informações desejadas. Os dados a serem coletados devem ser avaliados quanto ao volume e à qualidade que determinarão decisões quanto a modelos de análise, grau de amostragem, interpolações e estruturas de dados a serem utilizadas
- *custo de coleta*: a entrada de dados é uma tarefa dispendiosa, correspondendo a 60% de todo o investimento de implantação de uma aplicação geográfica [LMS90]
- *ausência de gerenciamento apropriado de dados*: muitos SIG disponíveis hoje comercialmente não têm o suporte de um SGBD (Sistema de Gerenciamento de Banco

de Dados) que realmente possa ser considerado como tal. Isto dificulta operações de segurança, compartilhamento de dados, controle de integridade e concorrência, facilidades fornecidas tipicamente por SGBD [EAT92, MGR91, Rom90]

- *limitação de funções disponíveis*: as funções de análise fornecidas pelos SIG atuais ainda são insuficientes para atender às necessidades dos usuários, sendo em geral voltadas para um domínio específico
- *impedância funcional entre sistema e usuário*: cada SIG implementa um modelo de dados próprio, geralmente distante da terminologia utilizada e do entendimento do mundo real pelos seus usuários.

Pesquisas e desenvolvimentos envolvendo SIG são basicamente de dois tipos: aquelas da comunidade de computação e aquelas dos usuários finais (e.g. cartógrafos, geógrafos, cientistas sociais, engenheiros). Recentemente, tem havido esforços no sentido de diminuir a distância entre estes dois mundos. Esta constatação baseia-se no grande número de conferências e relatórios de projeto em conjunto, como também no fato de que muitos projetos envolvendo SIG são altamente multidisciplinares. No entanto, é verdade que existem diferentes opiniões e que estas influenciam a abordagem do problema de processamento de dados geo-referenciados.

De fato, enquanto cientistas da computação muitas vezes não entendem perfeitamente os requisitos dos usuários finais, estes últimos raramente têm o conhecimento necessário para uma utilização correta das facilidades oferecidas por SIG. Como observado em [EM95], "muitos SIG exigem um treinamento extensivo, não apenas para familiarizar os usuários com a terminologia do software, mas também para educá-los em formalizações utilizadas para representar dados geográficos e gerar informações a partir deles". Constata-se que muitos dos problemas dos usuários finais têm origem em duas questões principais:

- *Limitações do SIG quanto à interface e facilidades de modelagem*

Geralmente existe uma incompatibilidade entre o vocabulário e o entendimento do mundo por parte do usuário e as facilidades de modelagem e desenvolvimento oferecidos por SIG. Assim para desenvolver aplicações ou consultar dados, os usuários são forçados a distorcer, de alguma maneira, a sua visão do mundo para se acomodar ao ambiente do software no que diz respeito à sua linguagem e estrutura de dados.

- *Limitações dos usuários devido à falta de treinamento adequado em Computação*

Os usuários são especialistas no seu campo de trabalho, e raramente têm um treinamento apropriado seja em Engenharia de Software ou em técnicas de modelagem e

projeto de banco de dados. O conhecimento destas técnicas possibilita a construção de aplicações mais eficientes e favorece a reutilização de dados e software.

A tese é uma contribuição à solução destas duas limitações, oferecendo ao usuário um *ambiente de modelagem de aplicações geográficas*, o UAPÉ - geo-User Analysis and Project Environment. UAPÉ é um ambiente amigável e extensível construído de forma prototipal no Instituto de Computação, Universidade Estadual de Campinas e que se propõe a auxiliar o usuário final a trabalhar com diferentes SIG através de um arcabouço conceitual coerente. Um aspecto inovador neste trabalho é que o UAPÉ integra as modelagens estática, dinâmica e de processos, utilizando as tecnologias de Engenharia de Software e banco de dados.

O UAPÉ disponibiliza para o usuário facilidades de modelagem e projeto de aplicações na área ambiental. O ambiente é aberto e foi projetado para ser acoplado a diferentes SIG como uma camada entre o SIG e o usuário final. Atualmente o UAPÉ não fornece uma interface para um SIG comercial, mas está baseado num protótipo de banco de dados geográfico desenvolvido em O_2 , um sistema de gerenciamento de banco de dados orientado a objetos [SV92]. A facilidade de se acoplar o UAPÉ a um SIG depende do grau de abertura do próprio SIG.

Utilizando o ambiente, os usuários serão capazes de projetar aplicações de acordo com suas necessidades, sem a necessidade de aprender os conceitos de implementação do SIG subjacente.

O UAPÉ prevê que o usuário seja conduzido através de uma série de etapas que asseguram uma documentação apropriada das decisões de projeto, permitindo o reuso e integração de fontes de dados já existentes. Estas etapas são impostas por uma *metodologia geral de concepção de aplicações geográficas* que foi desenvolvida na UNICAMP e validada por usuários finais de diferentes domínios de aplicação. As atividades de modelagem são suportadas por um *modelo semântico de dados*, GMOD, orientado a objetos, também utilizado pela metodologia, que apresenta ao usuário uma visão do mundo mais próxima da sua realidade. Este modelo, ao contrário da maioria das propostas existentes, considera a dimensão espaço-temporal integrada a restrições de integridade. O GMOD também representa uma contribuição significativa do ponto de vista de modelagem pois contempla não apenas a modelagem de dados, mas também a modelagem dinâmica e de processos.

O nome do ambiente reflete suas funcionalidades. Uapé é uma palavra indígena para a vitória régia - uma "interface" agradável - cujas folhas abrigam sob si um complexo ecossistema. Analogamente, UAPÉ apresenta ao usuário uma interface amigável para manipulação de estruturas complexas de dados geográficos.

As principais contribuições deste trabalho estão, portanto, voltadas à diminuição da distância entre usuário e SIG comentada em [PM97, MP94]. Estas contribuições são:

- um *modelo* de dados geográficos, GMOD, que modela aspectos estáticos e dinâmicos do mundo real, cuja primeira versão foi descrita em [PMB93, PM93],
- uma *metodologia* de especificação e projeto de aplicações geográficas para planejamento ambiental, voltada às necessidades do usuário, descrita em [MP95] e ampliada em [PMS96], e
- o *ambiente* UAPÉ, que integra modelo e metodologia de forma a assistir o usuário na tarefa de projetar aplicações geográficas e navegar no banco de dados espacial. Os principais aspectos deste ambiente foram descritos em [OPM97].

Os capítulos que se seguem estão organizados da seguinte forma. Inicialmente o Capítulo 2 define e discute conceitos fundamentais que são tratados nesta tese. O Capítulo 3 apresenta alguns dos trabalhos mais relevantes relacionados com modelos de dados geográficos e ambientes computacionais de modelagem de aplicações geográficas. O Capítulo 4 descreve o modelo de dados geográficos GMOD. O Capítulo 5 apresenta a Metodologia de Desenvolvimento de Aplicações Geográficas proposta na tese. O Capítulo 6 descreve a arquitetura básica do ambiente computacional UAPÉ. O Capítulo 7 descreve um exemplo de uso do UAPÉ como uma ferramenta para o projeto de uma aplicação geográfica específica e do seu banco de dados e, finalmente, no Capítulo 8 são apresentadas algumas considerações finais na forma de uma análise crítica do ambiente, conclusões e possíveis extensões do trabalho.

Vários módulos do UAPÉ foram especificados ou implementados como parte de trabalhos de iniciação científica ou dissertação de mestrado, mencionados ao longo do texto.

O Apêndice A ilustra a especificação do UAPÉ através do seu modelo de objetos, segundo a metodologia OMT [RBea91], e o esquema do seu banco de dados (BDUAPE), utilizando a linguagem de definição de dados do sistema O_2 .

Capítulo 2

Terminologia e Conceituação Básica

Este capítulo define e discute conceitos fundamentais que são tratados nesta tese. A Seção 2.1 introduz os principais conceitos utilizados em metodologias de desenvolvimento de aplicações em geral. A Seção 2.2 conceitua modelagem estática, dinâmica e de processos introduzindo noções sobre níveis de abstração comumente consideradas no processo de modelagem. A Seção 2.3 apresenta alguns aspectos referentes a aplicações geográficas. A Seção 2.4 caracteriza Sistemas de Informações Geográficas (SIG) em geral. A Seção 2.5 apresenta os conceitos básicos associados ao paradigma de orientação a objetos que são utilizados na tese.

2.1 Desenvolvimento de Software

O UAPÉ propõe-se a auxiliar o usuário a projetar aplicações geográficas a partir de alguns conceitos específicos, definidos a seguir.

A definição do modelo do **ciclo de vida** é a primeira consideração importante no planejamento do processo de desenvolvimento de software [Pre94]. O ciclo de vida de software determina todas as atividades necessárias para definir, desenvolver, testar, operar e manter um produto de software. O modelo de ciclo de vida estabelece uma base para o controle das várias atividades envolvidas no desenvolvimento e manutenção de um software.

Existem diversas divisões propostas para o ciclo de vida convencional de um software: modelo "waterfall", prototipação e modelo espiral. A descrição destes modelos foge ao escopo deste trabalho.

Pressman [Pre94] considera que o processo de desenvolvimento de software contém três fases genéricas independente do modelo escolhido. As três fases, definição, desenvolvimento e manutenção são encontradas em todo desenvolvimento de software, independente da área de aplicação, porte do projeto ou complexidade. A figura 2.1 ilustra estas

tres fases genéricas, detalhando os passos que compõem cada uma, segundo o modelo "waterfall".

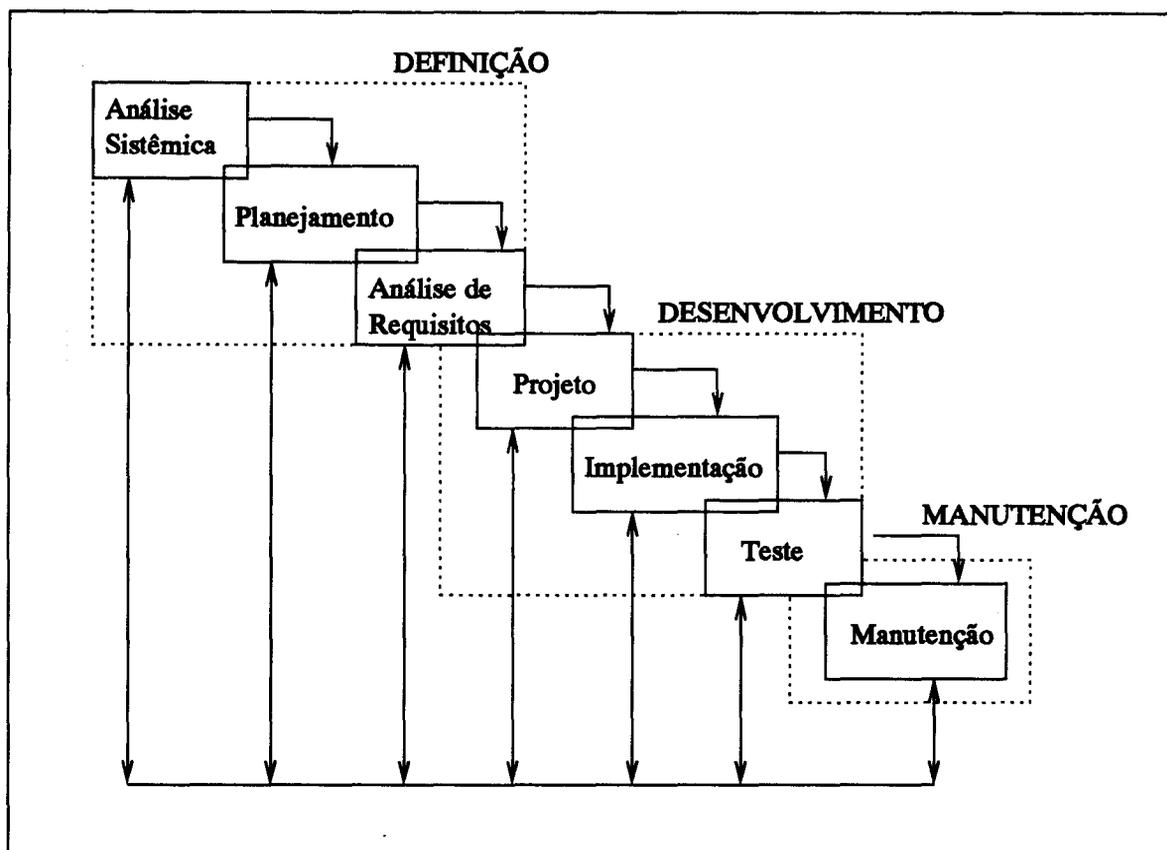


Figura 2.1: Fases Genéricas do Desenvolvimento de Software

Neste texto são utilizados termos como método e metodologia, cujos significados são encontrados na bibliografia de formas diversas, conforme o autor e muitas vezes de forma ambígua. Os termos que se seguem utilizam as definições de [Tol89] aplicadas a desenvolvimento de software. Ressalte-se que vários termos usados são particulares à área de banco de dados, sendo os termos correspondentes na Engenharia de Software muitas vezes diferentes.

- **Instrumento:** Instrumento é uma convenção notacional aliada às respectivas operações, normalmente automatizáveis, que suportam atividades específicas do processo de desenvolvimento ou de manutenção de software (por exemplo, linguagens, diagramas, tabelas).
- **Ferramenta:** Ferramenta é a implementação computacional de um instrumento ou

de um método, em um determinado ambiente.

- **Técnica:** Técnica é um conjunto de princípios para a execução de uma tarefa específica do processo de desenvolvimento ou de manutenção de software. Em geral, as técnicas são suportadas por conjuntos de ferramentas.
- **Método:** Método é um conjunto de diretivas e regras com vistas à seleção e aplicação de técnicas e instrumentos no desenvolvimento ou manutenção de software.
- **Metodologia:** Metodologia é um conjunto de métodos coerente, consistente e completo, juntamente com princípios e orientações de como, onde, quando e porque devem ser empregados no desenvolvimento e manutenção de um sistema de software. Uma *metodologia* deve ser uma filosofia geral aplicável a todo o *ciclo de vida* do software. É importante observar que em Engenharia de Software, o termo atualmente mais utilizado é *processo*, mas este uso não será feito na tese em função do entendimento do usuário do que seja um processo (vide abaixo).
- **Ambiente:** Ambiente computacional, de agora em diante simplesmente denominado ambiente, é um conjunto integrado de ferramentas que auxiliam o processo de criação e evolução de software, segundo um determinado método ou metodologia.

Para a concepção de aplicações convencionais existem atualmente diversas metodologias que orientam os desenvolvedores na definição, desenvolvimento e manutenção das aplicações. Independentemente da abordagem utilizada, todas elas partem do entendimento das necessidades dos usuários e das regras do negócio (*o que fazer*) para a elaboração das fases de projeto e implementação que se concentram em *como* atender tais necessidades.

2.2 Modelagem Estática, Dinâmica e de Processos

O modelo proposto na tese visa a permitir uma descrição do mundo real sob o ponto de vista de aplicações geográficas. Neste contexto, modelagem do mundo real compreende a *modelagem estática*, a *modelagem dinâmica* e a *modelagem de processos*, envolvendo a seleção, abstração e generalização de entidades de interesse do usuário assim como também a especificação dos aspectos dinâmicos e funcionais. Os resultados da modelagem direcionam a definição do banco de dados geográfico assim como também especificam a biblioteca de funções e parâmetros que devem ser utilizados junto com os dados no banco de dados.

A modelagem estática está voltada à determinação das principais entidades a serem armazenadas no banco de dados, seus componentes (atributos) e interrelacionamentos. A modelagem dinâmica e de processos preocupam-se com a construção de modelos matemáticos que descrevem operações envolvendo a representação dos dados armazenados e inclui a simulação de fenômenos naturais no tempo [BX94].

Técnicas tradicionais de modelagem não são suficientes para tratar de informações geográficas. As dificuldades devem-se principalmente ao fato de que dados geográficos precisam ser considerados com relação à sua localidade, tempo de observação e precisão [Chr90, Cou92, FT93, Goo91b, KT92, RM92] e os modelos atuais não levam isto em consideração. Criar um modelo de dados geográfico é uma tarefa complexa porque envolve a representação, de forma discreta, de uma realidade contínua e espacial e deve conciliar as diversas visões dos usuários da aplicação.

Em geral, modelos propostos na literatura para aplicações geográficas concentram-se ora nos dados, para criação do banco de dados, ora nos processos, para definição matemática dos relacionamentos entre as entidades espaciais. Na maior parte das vezes, estes modelos são utilizados de forma independente, impedindo que haja uma integração adequada entre os dados armazenados e as funções que serão aplicadas sobre eles para simular condições do mundo real.

Modelos de dados geográficos vêm sendo propostos por especialistas em bancos de dados, geralmente utilizando noções de orientação a objetos [TS94]. Modelos dinâmicos e de processos são usualmente analisados por especialistas da área-alvo (biólogos, ecologistas, engenheiros) e estão voltados à especificação matemática do mundo.

O UAPÉ, como se verá posteriormente, atende ambas as necessidades. O Capítulo 3 contém uma descrição detalhada de diferentes modelos propostos na literatura.

2.3 Aplicações Geográficas

O conjunto de atividades e tecnologias envolvidas no processamento de dados geográficos recebe o nome genérico de *geoprocessamento*. Dados espaciais que estão ligados a uma localização na superfície da Terra são chamados de *dados geo-referenciados* [Car89].

Aplicações geográficas ou de geoprocessamento tratam de fenômenos que ocorrem no mundo real considerando seus aspectos espaciais. Mais precisamente, esses fenômenos geográficos são geo-referenciados. Aplicações geográficas envolvem vários domínios do conhecimento. Exemplos são [Aro89]: planejamento urbano, otimização de rotas, gerenciamento de redes públicas, demografia, cartografia, agricultura, administração de recursos naturais, monitoramento costeiro, controle de incêndios e epidemias. Cada tipo de aplicação trata com diferentes facetas, escalas e características espaço-temporais.

No contexto deste trabalho, uma aplicação geográfica é entendida como um conjunto de

processos executados numa seqüência definida segundo um modelo pré-estabelecido, sobre uma base de dados geo-referenciados e convencionais. Alguns desses processos podem ser automatizados (por exemplo, *overlays* de mapas), enquanto em outros a automatização nem sempre é cabível (por exemplo, definição de estratégias). O conjunto de processos automatizados constitui um *software* cujos componentes são criados através de uma série de traduções que mapeiam os requisitos do usuário em código executável em computador.

Em [MGR91], aplicações geográficas são divididas em três grandes categorias: aplicações *sócio-econômicas* (manipulação e análise de dados espaciais sobre a terra, o povo e infraestrutura); aplicações *ambientais* e aplicações de *gerenciamento* sobre as duas categorias acima. Uma outra divisão canônica (usualmente adotada por fornecedores de SIG) está baseada na escala geográfica tratada pela aplicação. Aplicações AM/FM (Automated Mapping/Facilities Management - aplicações de cadastro) consideram dados em escalas que variam no intervalo de 1:500 a 1:20.000. Aplicações ambientais trabalham com escalas menores (e.g. abaixo de 1:5.000.000). Estas taxonomias não são absolutas, uma vez que planejadores ambientais podem utilizar também dados cadastrais e aplicações AM/FM podem requerer estudos ambientais.

Devido às características espaço-temporais de seus dados, muitas aplicações geográficas ainda não têm suporte computacional adequado. Os principais desafios encontrados são:

- *diversidade de perfil* dos usuários (engenheiros, antropólogos, geólogos, dentre outros), com diferentes visões dos mesmos fenômenos geográficos.
- *peculiaridades de modelagem*: assim como qualquer aplicação padrão, a concepção de uma aplicação geográfica deve passar pelos estágios de definição, desenvolvimento e manutenção. A fase de definição concentra-se principalmente na modelagem conceitual, para a qual não há padrão.
- *metodologia de concepção*: não existem estudos publicados de como as metodologias de desenvolvimento convencionais podem ser adaptadas para apoiar as especificidades das aplicações geográficas.
- *gerenciamento de dados*: dados geográficos são uma forma de dados espaciais. Neste trabalho, dados espaciais referem-se a dados geo-referenciados e portanto os termos, espacial, geográfico e geo-referenciado, serão utilizados indistintamente.

Dados geográficos são comumente caracterizados como tendo dois componentes fundamentais [Aro89]: o fenômeno sendo estudado e a localização espacial do fenômeno. Um outro componente crítico para a informação geográfica é o tempo [Peu93, HW90, Lan89, Vra89, BFAT91]. Dados geográficos descrevem um fenômeno em uma localização em um momento específico. A coleta de dados ao longo do tempo (por

exemplo, séries históricas) é fundamental para aplicações voltadas ao planejamento e tomada de decisão.

Os dados são obtidos em diferentes formatos, de diferentes fontes e localizações geográficas e são capturados por diversos tipos de equipamentos; os dados ocupam grande espaço de armazenamento e necessitam de operações especializadas de análise e apresentação de resultados não disponíveis em sistemas de gerenciamento de banco de dados comerciais.

Entidades geo-referenciadas são caracterizadas por suas propriedades específicas (como densidade) e relacionamentos espaciais com outras entidades (e.g. distância). Tais propriedades fazem com que as facilidades disponíveis nos atuais SGBD precisem ser estendidas para um suporte adequado às aplicações geográficas.

- *consultas e análise espacial:* consultas a dados geográficos requerem freqüentemente funções de análise, envolvendo problemas de processamento de operações métricas e topológicas.
- *apresentação dos resultados:* o resultado das consultas necessita de algoritmos de apresentação, envolvendo procedimentos sofisticados que requerem conhecimentos de computação gráfica e processamento de imagens.

2.4 Sistemas de Informações Geográficas

Os problemas envolvendo SIG começam pela definição do termo [MGR91, Cow90, Aro89, SMSE87]. Há inúmeras definições para SIG, cada uma baseada no tipo de usuário e no domínio da aplicação. A definição mais geral [Goo91a] considera SIG como um sistema de informações cujos dados são de alguma forma geo-referenciados. A abordagem da comunidade técnica de banco de dados define SIG como um *banco de dados não-convencional que suporta o gerenciamento de dados espaciais*. A visão de Engenharia de Software considera SIG como um conjunto de ferramentas e algoritmos para manipular dados geográficos.

Dependendo da definição, diferentes aspectos são considerados refletindo a multiplicidade de uso da tecnologia SIG. Neste trabalho combinamos as visões de bancos de dados e processos com as seguintes propriedades: um *SIG é um software que gerencia e executa operações de consulta e atualização em dados geo-referenciados armazenados em um banco de dados geográfico*. Outras dimensões importantes para o usuário são a sua capacidade de visualização e apresentação dos dados e os tipos de análise espacial permitidos. Vale a pena ressaltar que nem todo SIG utiliza realmente um SGBD, mas sim um conjunto de bases de dados.

Na comunidade de banco de dados, SIG são associados a *banco de dados espaciais*, havendo uma grande quantidade de pesquisas relacionadas a estruturas de dados e métodos de acesso [Sam89, GB90]. Em muitos casos, o gerenciador de dados usa um SGBD relacional para armazenar dados convencionais, com apontadores para os dados espaciais que por sua vez são gerenciados por módulos de software específicos.

A pesquisa em SIG envolve tanto necessidades dos usuários como tópicos em Computação. Estes últimos abrangem uma enorme gama de assuntos, que vão desde os aspectos cognitivos até o armazenamento de dados.

Esta tese concentra-se em aspectos relativos a bancos de dados, mais especificamente na área de modelagem e projeto de aplicação. Alguns outros tópicos de pesquisa em bancos de dados e SIG são [MP94]:

- *linguagens de consulta e otimização*: as interfaces de SIG normalmente suportam dois tipos de interação: linguagens de consulta textuais e manipulação interativa dos elementos geográficos [OM96]. Dois tópicos precisam ser considerados: a validade destes mecanismos para consultas espaço-temporais e a forma de apresentação dos resultados. Há várias propostas de extensões à linguagem SQL a fim de torná-la uma linguagem espacial. A validade desta abordagem é discutida em [Ege92]. Vários outros mecanismos de consulta textual são descritos na literatura de SIG [WCY89, SV92, LL93].
- *captura de dados*: durante a entrada de dados, os componentes espaciais e não-espaciais das entidades geo-referenciadas precisam ser lidos e relacionados corretamente. Um das questões mais difíceis envolvendo aplicações SIG, e especialmente aquelas baseadas em dados cartográficos (mapas), é a atividade de preparação dos dados, isto é, problemas associados com interpretação de mapas, incompatibilidades de sistemas de coordenadas ou análise de imagens [Mul91].
- *extensão dos atuais SGBD para suportar dados geo-referenciados*: SGBD ainda não são utilizados de forma conveniente como suporte a SIG. Para que tal aconteça, muitas de suas facilidades precisam ser revistas e adaptadas. O modelo de dados relacional é atualmente o modelo mais utilizado para o tratamento de dados não-espaciais. Embora conveniente para o tratamento de dados convencionais, o uso do modelo relacional padrão para dados espaciais ainda apresenta grandes desafios pelas características inerentes dos dados. Exemplos desses desafios são o tratamento de dados complexos, a adaptação dos mecanismos de tratamento de versões e visões para dados geográficos [Cer96, Cur97, DGM95], o suporte a recursos não convencionais de apresentação de resultados e o suporte a múltiplas representações [NM95, MBJ96].

Outros desafios da área de pesquisa em bancos de dados irão contribuir diretamente para a evolução de sistemas de informações geográficas. Dentre eles podemos citar: sistemas de gerenciamento de dados espaço-temporais [Lan89, Sno92, BFAT91, Lan93, Pri89], "data mining" para a determinação de correlações entre os dados armazenados [AIS93], sistemas ativos para suporte a tomada de decisões em aplicações de planejamento [Web90, YS91, PMP93], gerenciamento de transações longas [BK91], gerenciamento de bancos de dados heterogêneos e distribuídos para integração de dados [EP90, Chr90, She91, CGM⁺93].

2.5 Conceitos Básicos de Orientação a Objetos

O modelo GMOD, uma das contribuições da tese, é orientado a objetos. Neste texto, utilizamos como base para modelagem, o modelo baseado em classes de [Bee89]. Um objeto é uma instância de uma classe e é caracterizado por sua identidade, *estado* (conjunto de valores de atributos) e *comportamento* (conjunto de operações ou métodos que podem ser aplicados ao objeto). Um objeto o pode ser composto de outros objetos o_1, \dots, o_n , caso em que o é chamado *complexo* e o_1, \dots, o_n são os *componentes* de o . Este processo de composição é realizado através da aplicação de *construtores* – por exemplo, construtores de conjunto – que permitem especificação progressiva de objetos cada vez mais complexos a partir de componentes previamente definidos. Objetos não complexos são denominados *simples*. As classes são estruturadas em hierarquias de herança; os ancestrais de uma classe C na hierarquia são as *superclasses* de C e seus descendentes, as *subclasses* de C . Os descendentes de uma classe *herdam* sua estrutura e comportamento.

Denominamos de *meta-classe* uma classe cujas instâncias são classes. Uma *classe abstrata* é uma classe que não é instanciada em objetos. Uma classe é *concreta* quando ela é instanciável em objetos.

2.6 Resumo

Este capítulo apresentou alguns conceitos e terminologia básica que serão utilizados no texto da tese. Em especial, foram definidos termos de Engenharia de Software que serão empregados no UAPÉ e terminologia para descrição do modelo orientado a objetos.

Capítulo 3

Modelagem de Dados Geográficos

Este capítulo tem como objetivo apresentar alguns dos trabalhos mais relevantes relacionados com modelos de dados geográficos (seção 3.1) e ambientes computacionais de modelagem de aplicações geográficas (seção 3.2).

3.1 Modelagem de Fenômenos Geográficos

A modelagem de dados espaciais constitui ainda um desafio [Abe89, HW90, Goo92, RM92]. Num nível de abstração mais alto, os usuários finais entendem a realidade geográfica segundo dois modelos básicos: a *visão ou modelo de campos* e a *visão ou modelo de objetos* [FG90, NCG89, Goo91a]. Essas visões são implementadas em diferentes estruturas de dados no SIG, utilizando operações específicas para cada tipo de representação. Essas implementações têm confundido as atividades de modelagem e implementação [Cou92].

A visão de *campos* entende o mundo como uma superfície contínua (camada) sobre a qual as propriedades variam em uma distribuição contínua (por exemplo, pressão atmosférica). Entidades individuais são criadas no processo de modelagem e não existem independentemente [goo93]. A ênfase está no conteúdo destas áreas, ao invés de nos seus limites.

A visão de *objetos* trata o mundo como uma superfície povoada de objetos identificáveis, que existem independentemente de qualquer definição (por exemplo, um determinado rio). Nesta visão, dois objetos podem ocupar o mesmo lugar no espaço. As entidades do banco de dados correspondem aos objetos identificáveis, que são definidos a priori.

Tanto objetos quanto campos podem ser organizados por *tema* (vegetação, solo).

Estes modelos são mapeados em diferentes estruturas: tesserar e vetorial (vector). Existem discussões a respeito de qual a mais apropriada, o que gerou o debate conhecido como "raster-vector debate" [Cou92].

Os dados na *visão de campos* são usualmente processados em formato de malha (*tesseral*) (entidades espaciais descritas como unidades poligonais do espaço - células - em uma matriz). Cada célula contém um valor temático. Células podem ter diferentes formatos; células quadradas são chamadas de *pixels*. O formato raster (que é o nome mais comumente usado para dados no formato de malha) é um tipo especial de malha com células de formato retangular, organizadas em ordem linear de varredura. Neste caso, as coordenadas não são armazenadas, mas sim derivadas pela posição das células na ordem de varredura.

Os dados na *visão de objetos* são processados como pontos, linhas e polígonos (formato *vetorial*), usando listas de pares de coordenadas. Os limites de regiões são armazenados de forma precisa, e vários atributos podem ser associados a um único elemento. *Redes* são um caso especial de dados vetoriais, onde os elementos são conjuntos de arcos e nós. As redes são usadas principalmente em gerenciamento de recursos e análise de rotas (por exemplo, transporte ou hidrografia).

Os trabalhos iniciais em modelagem de dados para aplicações geográficas se preocuparam basicamente com as estruturas para suportar a geometria e a espacialidade dos dados (incluindo a noção de topologia). Os primeiros modelos refletiam diretamente a geometria e estavam fortemente relacionados com estruturas de dados. [RL95] ressalta que os desenvolvedores que utilizam SIG são forçados a definir as entidades que serão tratadas pela aplicação em termos das estruturas internas do SIG subjacente. Este fato causa uma impedância entre a visão do usuário da aplicação e as necessidades de implementação no SIG. Por exemplo, o mesmo conceito do mundo real é denominado, dependendo do SIG, como tema, categoria, camada, plano de informação, mapa. Esta situação está muito evidente em um grande número de sistemas disponíveis atualmente, onde o usuário se refere diretamente a estruturas de arco-nó, no caso de sistemas vetoriais, ou a grids e quadrees, em sistemas raster. Como consequência, os procedimentos de modelagem misturam necessidades da aplicação com as restrições impostas pelas estruturas internas do SIG.

Alguns modelos de dados propostos visam a abstrair questões inerentes a uma determinada área de aplicação [Her92, Kub84, KE92, KT92, FR93, RL95]. Modelos de dados orientados a objetos vêm sendo recomendados para aplicações geográficas [MMS93, WHM90, SV92, SA93, DRSM93, AA93, PMP93, Wor94b, RL95]. As principais vantagens se referem ao fato de que tais modelos permitem ao usuário uma especificação incremental e reutilizável, devido às propriedades de herança, composição e encapsulamento. Além do mais, diferentemente de outros modelos, o projetista também pode descrever o comportamento das entidades do mundo real, permitindo uma melhor modelagem da dinâmica, inerente aos fenômenos do mundo real.

Segue uma revisão de alguns modelos genéricos de dados geográficos recentemente pro-

postos na literatura, todos eles orientados a objetos. Esta revisão não é exaustiva. Esses modelos foram selecionados para representar, cada um, características específicas a serem comentadas que os distinguem dos demais. Os principais problemas (que são comuns a todas as propostas de modelagem existentes) são a ausência de relacionamentos e restrições e a falta de menção à modelagem dinâmica. Apenas o modelo de Worboys considera a variação temporal. A figura 3.1 ilustra a comparação entre os modelos considerando estas características.

3.1.1 Modelagem espaço-temporal integrada

Worboys [Wor94a] propõe um *modelo de representação* que integra as dimensões espacial e temporal. A proposta adota a visão de objetos no sentido de que o "espaço de informação" é visto como sendo composto de objetos independentes, encapsulando estado e comportamento.

A modelagem espacial estende o modelo apresentado em [Wor92] baseado em topologia combinatorial, onde objetos espaciais são apresentados como *complexos simpliciais*. Um *simplicial* é um ponto único (0-simplicial), um segmento finito de reta (1-simplicial) ou uma área triangular (2-simplicial). Um *complexo simplicial* é uma coleção de *simpliciais* que não se sobrepõem, tal que se um *simplicial* pertence a um complexo, então todos os seus componentes também pertencem. Um *complexo simplicial* é determinado pelo seu componente simplicial maximal.

A modelagem temporal considera duas dimensões ortogonais: o tempo de transação e o tempo válido [Sno92]. As referências bitemporais são expressas por elementos denominados *BTE* (*bitemporal elements*). Um *BTE* é a união de um conjunto finito de produtos cartesianos de intervalos da forma $I_D \times I_E$, onde I_D é um intervalo de tempo de transação e I_E é um intervalo de tempo válido. Um *BTE* T é tal que $(t_D, t_E) \in T$ se, e somente se, no tempo t_D há informação no banco de dados que o objeto bitemporalmente referenciado por T existe como tempo válido em t_E .

Um objeto espaço-temporal *ST-simplicial* é um objeto único que possui as extensões espacial e bitemporal, descrito por um par ordenado $\langle S, T \rangle$, onde S é um *simplicial* e T é um *BTE*. Um *ST-complexo* é um conjunto finito de *ST-simplicial*.

Complementando o modelo de representação, o trabalho define diversas operações sobre *ST-complexos* classificadas em igualdade e conjunto, topológicas, projeções, composição e seleção (espacial e temporal).

3.1.2 Modelos com representações múltiplas

Dois modelos consideram várias representações: [TS94, Pim95] e [CCH⁺96]. O trabalho de Times e Salgado [TS94] e Pimentel [Pim95] apresenta uma proposta de modelagem

de dados geográficos em mais alto nível, o MGEO+, com o objetivo de permitir que os usuários raciocinem sobre os objetos em qualquer domínio de problema. O modelo é descrito utilizando os conceitos de orientação a objetos onde as entidades do mundo real de interesse da aplicação são organizadas em uma hierarquia de classes. Os objetos podem ter mais de uma representação geométrica e simbólica, ou seja, "um mesmo objeto que pode ser representado em diferentes escalas e projeções, terá associado a ele um nível de informação descritiva, uma geometria e uma simbologia".

Além de modelar os dados, o MGEO+ se preocupa com a apresentação. No nível mais alto de abstração, a principal classe é o *Modelo de Representação* (classe de modelos de aplicações). A aplicação é modelada através de Planos de Informação, que são compostos por Entidades Geográficas associadas entre si. Existem diferentes representações permitidas: Representação Simbólica (relativa à apresentação de uma Entidade), Representação Espacial (relativa a Elementos Contínuos- matricial ou Discretos-vetorial) e Representação Descritiva. Estas representações são determinadas como Propriedades de Entidades Geográficas.

Para a classe Modelo de Representação são definidos os operadores conversão de escala e conversão de projeção. Para as classes de objetos não espaciais são definidos alguns operadores convencionais como cálculo de soma, máximo e mínimo. Para as classes de objetos espaciais, os operadores são agrupados em orientados a conjuntos (igual, pertence, subconjunto, dentre outros), topológicos (interior, fronteira, adjacente, dentre outros) e métricos (distância, comprimento, área, perímetro, direção). Representações matriciais e vetoriais são contempladas respectivamente, nas hierarquias de classes com raiz nas classes Elementos Contínuos e Elementos Discretos. Recentemente, este modelo foi implementado sobre o SGBD orientado a objetos O_2 [Bat97]. A implementação simplifica algumas classes e define vários métodos para processamento de consultas espaciais.

O modelo apresentado em [CCH⁺96] está organizado em níveis de abstração, distinguindo as abstrações conceituais de um fenômeno do mundo real de suas representações, além de adotar uma abordagem unificada das visões de campo e objeto. Assim, um campo ou objeto geográfico é especificado formalmente pelo usuário no nível conceitual e associado a diferentes representações (nível de representação), cada uma das quais, por sua vez, armazenada em uma variedade de estruturas físicas (nível de implementação).

No nível conceitual, o mundo real é modelado como uma coleção de classes, classificadas em classes convencionais e classes geo-referenciadas ou geo-classes. As instâncias das geo-classes modelam campos e objetos geográficos, enquanto as instâncias das classes convencionais correspondem ao conceito padrão de classes em SGBD orientado a objetos. Os objetos das classes convencionais são denominados objetos não-espaciais.

No nível de representação, o modelo suporta duas hierarquias básicas de classes de representação, cujas raízes são as classes *REP_GEOCAMPO* e *REP_GEOOBJETO*. As

instâncias destas classes são chamadas de *representações de geo-campos e de geo-objetos*, respectivamente e naturalmente descrevem representações de geo-campos e geo-objetos.

A associação entre objetos dos dois níveis é feita através da classe *REPRESENTA*. Cada instância desta classe é um objeto complexo associando um geo-campo ou geo-objeto *o* a uma representação *r* e vice-versa.

3.1.3 Modelo para SIG usando sistemas especialistas

Existem vários exemplos do uso de sistemas especialistas em SIG. Aqui destacamos o modelo desenvolvido por Subramaniam e Adam [SA93] que aproveita algumas das idéias apresentadas em [OM88] e integra a abordagem orientada a objetos com conceitos de regras de produção.

O modelo suporta somente a visão de objetos e as entidades do mundo real são agrupadas em classes de objetos simples denominadas *Non_Reference_Spatial_Feature* (por exemplo, estradas, rios) e classes de objetos compostos denominadas *Reference_Spatial_Feature* (por exemplo, estados, condados). As entidades espaciais são organizadas em hierarquias de composição onde um objeto complexo é composto por objetos mais simples.

O modelo implementa a classe Regra para que se possa incorporar ao sistema conhecimento sobre contextos e perspectivas do usuário. Nesta classe são instanciadas e mantidas regras necessárias para o processamento de consultas imprecisas. As regras não são associadas explicitamente a objetos no banco de dados, podendo, portanto, serem adicionadas ou removidas sem interferir em qualquer um deles.

Os relacionamentos explorados no modelo são basicamente os de agregação e composição.

3.1.4 GeO2

O modelo de dados GeO2 apresentado em [DRSM93] separa o nível conceitual do nível físico ou interno. A localização das entidades geográficas é definida na classe *Geometry* que garante a independência entre os dois níveis. A hierarquia *Geometry* tem várias subclasses básicas, que modelam estrutura e comportamento de dados vetoriais.

O modelo conceitual é uma extensão do modelo Entidade-Relacionamento com suporte a mecanismos de herança e propagação. Uma entidade pode ter diversos atributos de localização permitindo a representação de objetos complexos.

O modelo de localização, intermediário entre o conceitual e o físico, permite o processamento da localização das entidades.

O modelo de dados geométrico e topológico suporta as estruturas de dados internas e influi na funcionalidade e desempenho da aplicação. O modelo suporta três níveis de

estrutura de dados: o nível "spaghetti" no qual as primitivas são definidas independentemente umas das outras; o nível de *Rede* corresponde aos grafos planares e não-planares apropriados para redes onde é importante o aspecto de navegação; por último, o nível de *Mapa* ou *Topológico* incorpora faces.

O modelo semântico construído pelo usuário é traduzido em classes no banco de dados orientado a objetos (O_2). O modelo não implementa classes temporais e não explora a especificação de relacionamentos.

	Dimensão Temporal	Relac.	Represent. Múltipla	Visão Campo	Visão Objeto	Modelagem Dinâmica	Modelagem Funcional
Worboys	x	-	-	-	x	-	-
MGEO+	-	-	x	x	x	-	-
Camara	-	-	x	x	x	-	-
Subramanian	?	-	?	-	x	-	-
GEO2	-	-	?	?	x	-	-
GMOD	x	x	x	x	x	x	x

Nota: O caracter "?" indica que a característica correspondente não é comentada na referência utilizada na tese

O caracter "-" indica que a característica correspondente não é considerada no modelo

O caracter "x" indica que a característica correspondente é tratada no modelo

Figura 3.1: Quadro Comparativo entre os Modelos

3.2 Ambientes de Modelagem

Os SIG atuais não suportam modelagem de dados e de processos de forma adequada. Do ponto de vista da Computação, as soluções para estes problemas têm sido apontadas pelas comunidades de banco de dados e engenharia de software; a distância entre o projeto da aplicação e seu desenvolvimento pode ser diminuída através de metodologias adequadas de modelagem e projeto, onde a visão do mundo real seja diretamente suportada por ferramentas apropriadas que mapeiem a visão do usuário em estruturas do banco de dados. As mesmas facilidades de modelagem ajudam a integração de diferentes fontes de dados, contribuindo para o controle de padrões de qualidade.

Existem alguns trabalhos voltados para o desenvolvimento de ferramentas e ambientes

que possibilitem a aproximação entre o usuário e SIG. Esta aproximação favorece a reutilização de dados e tomada de decisões [CHCS95]. Os trabalhos podem ser classificados genericamente como: ferramentas para resolverem problemas específicos; nova geração de SIG; e diferentes tipos de ambientes.

Novas gerações de SIG têm sido desenvolvidas de forma prototipal, geralmente sobre sistemas de gerenciamento de banco de dados orientados a objetos ou extensíveis. Estes bancos de dados servem de base para solução dos problemas de modelagem e manipulação de dados [VvO91, DRSM93, SV92, MMS93, GR93, Gut89, SFGM93]. Muitos desses protótipos baseiam-se nas facilidades fornecidas por esses bancos de dados para ajudar na especificação dos dados. Embora tais sistemas demonstrem a preocupação em ajudar o usuário na modelagem de dados, não se vê a mesma intenção com relação à modelagem de processos.

Exemplos de ambientes que consideram modelagem de processo são [SAA⁺96, SSA95]. Estes, no entanto, não consideram a integração com a modelagem de dados.

O trabalho de [SSA95] é dirigido especificamente à modelagem de processos, descrevendo um ambiente computacional para métodos de modelagem científica, para suportar representação, manipulação e avaliação de conceitos científicos. A modelagem de processos é construída em termos de estruturas-R (estruturas de Representação), que são representações abstratas de um conceito (semelhante ao conceito de tipos abstratos de dados). Estruturas-R podem ser combinadas através de construtores (e.g. agregação) e restrições. A dinâmica dos processos do mundo real é modelada através de seqüências de transformações de instâncias de estruturas-R. Como estas transformações levam dados de um estado (instância de estrutura-R) para outro, a dinâmica dos fenômenos geográficos é modelada utilizando grafos dirigidos de estrutura-R onde os nós são as instâncias e as arestas representam transformações.

Estas idéias foram aplicadas na construção de um ambiente computacional de modelagem - Amazonia - que suporta pesquisa em hidrologia de larga escala. O conceitos de estruturas-R foram implementados com a ajuda de um sistema de banco de dados orientado a objetos. O ambiente foi ainda utilizado na construção de um outro sistema, com o mesmo propósito, porém sobre um banco de dados distribuído [SAA⁺96]. Este tipo de ambiente auxilia os usuários de SIG na modelagem de processos de forma análoga à construção de algoritmos, sendo por isso um importante passo para aproximar os usuários das ferramentas de SIG. No entanto, não auxilia o usuário no projeto ou utilização de conjuntos de dados, ou mesmo na pesquisa por dados existentes, que são características necessárias ao ambiente de aplicações geográficas.

Uma abordagem diferente é proposta em Hermes [AAT94], que é um ambiente que auxilia o gerenciamento de dados para a tomada de decisões em planejamento de áreas urbanas. Hermes está baseado em uma metodologia de planejamento urbano, que dirige o

usuário na seleção de dados apropriados e desenvolvimento de cenários, para a tomada de decisões. A metodologia considera três passos de planejamento: planejamento estratégico, que define regras para a classificação de áreas urbanas com relação a restrições desejadas; planejamento tático, que procura áreas potencialmente adequadas aos objetivos do projeto; e o planejamento operacional. Hermes permite que o usuário armazene diferentes algoritmos para combinação de dados, cada um destes correspondendo a um modelo. O ambiente está construído sobre um sistema de banco de dados geográfico relacional. Os dados são modelados de acordo com uma extensão do modelo relacional, dirigido para dados geo-referenciados. Hermes não considera a visão de campos da modelagem, devido ao fato que está voltado a aplicações de planejamento urbano.

Um último ambiente de projeto de aplicações em SIG que merece ser mencionado é discutido em [SW95]. Trata-se de uma ferramenta CASE com suporte a dados temporais para auxiliar usuários de SIG na prototipação rápida de aplicações que serão implementadas no SIG subjacente. O modelo estende as noções de orientação a objetos com os conceitos de modelagem temporal. Não há, no entanto, nenhuma preocupação com os padrões de trabalho do usuário.

3.3 Resumo

Este capítulo apresentou alguns trabalhos relevantes de modelagem de dados para aplicações geográficas e alguns ambientes de modelagem de processos.

Foram mencionadas propostas de modelos geográficos que vão desde os modelos de estrutura e representação de dados até os que abstraem a realidade geográfica num nível mais semântico, portanto, mais próximo do usuário.

Não há integração, em um único ambiente, de modelagem de dados e de processos. Na maioria dos casos, a dimensão temporal é ignorada e nenhum modelo ou ambiente se preocupa com relacionamentos ou restrições entre as entidades modeladas.

O próximo capítulo apresenta a primeira contribuição desta tese, o modelo GMOD, que integra modelagem estática de dados à modelagem dinâmica e de processos.

Capítulo 4

Modelo de Dados Geográficos - GMOD

Este capítulo descreve o modelo de dados geográficos GMOD. O GMOD é orientado a objetos e difere da maioria dos modelos recentemente propostos pelas seguintes razões:

- é independente de qualquer modelo de dados utilizado por SIG comerciais, permitindo representar dados convencionais e geo-referenciados em um nível mais alto de abstração,
- é extensível, permitindo ao usuário derivar as classes de interesse da sua aplicação a partir de classes pré-definidas,
- permite a especificação de processos, e conseqüentemente, a especificação de aplicações,
- permite representar relacionamentos temporais entre fenômenos geográficos, e
- permite a especificação de regras para a implementação de restrições e funções de monitoramento e simulação, possibilitando retratar a dinâmica do mundo real

4.1 Introdução

O GMOD é orientado a objetos e é a base de comunicação entre o usuário e o ambiente UAPÉ. O GMOD utiliza como base a proposta de [Cam95], adicionando três importantes características: modelagem da dimensão temporal, modelagem dinâmica e modelagem de processos (ou funcional), com descrição dos relacionamentos entre as entidades.

A utilização do modelo GMOD nas atividades de modelagem e projeto de aplicações ambientais permite que o usuário considere, assim como em [CCH⁺96], quatro níveis de abstração para reconhecimento, classificação e descrição dos dados e seus relacionamentos.

- O nível do *mundo real* visualiza a realidade geográfica como composta de uma coleção de objetos e entidades bem definidas (modelo de objetos), ou como uma coleção de entidades espaciais que, pela sua natureza, são incompletas ou imprecisas (modelo de campos). Neste último caso, as entidades não podem ser vistas em toda a sua extensão, mas somente descritas através de amostragens e reconstruídas por interpolações. No contexto de aplicações geográficas, a identificação das entidades de interesse pode variar com a disciplina, escala espacial e opiniões pessoais.
- O nível *conceitual* modela as entidades discretizadas no nível acima, segundo as visões e critérios de agrupamento de interesse da aplicação (por exemplo, escala, tempo).
- O nível da *representação* introduz detalhes sobre a representação geométrica e topológica das propriedades espaciais das entidades geográficas definidas no nível acima.
- O nível de *implementação* corresponde às estruturas internas e implementação das operações. Este nível corresponde efetivamente à organização interna de armazenamento dos dados e métodos de acesso.

No caso de bancos de dados geográficos, este nível especifica a forma como as entidades geográficas serão implementadas. A escolha do modelo de implementação está estreitamente ligada a critérios de uso e volume de armazenamento. Por exemplo, para controle de fluxo de tráfego é mais conveniente modelar a extensão espacial de estradas em estruturas de dados do tipo arco-nó do que num conjunto de células adjacentes.

GMOD está baseado em três conceitos principais: *classes* (no sentido de orientação a objetos); *relacionamentos* que permitem conectar estas classes de várias maneiras; e *restrições* que são impostas sobre as classes, relacionamentos e suas instâncias. Classes e relacionamentos podem ser temporais ou atemporais, de acordo com a possibilidade de suas instâncias poderem ou não variar com o tempo. Restrições permitem realizar a modelagem dinâmica.

A Figura 4.1 mostra a estrutura de classes do GMOD no nível conceitual através de um diagrama que utiliza as convenções gráficas do modelo OMT [RBea91]. A notação gráfica do OMT foi estendida para facilitar a indicação de componentes peculiares do GMOD, incluindo-se a notação para relacionamentos de derivação, causais e versões e classes temporais (vide Capítulo 7). A representação de classes foi simplificada para facilitar a visualização, excluindo-se do gráfico a descrição dos seus *atributos* e *métodos*. Os atributos serão discutidos quando necessário. Os métodos básicos são aqueles pré-definidos em

qualquer sistema orientado a objetos, a saber, *create*, *destroy*, *update*, *display*. Diferentes implementações de representação têm, além disso, métodos específicos para sua manipulação (por exemplo, métodos de relacionamentos topológicos no caso de representação vetorial). A maneira como as classes são manipuladas pelo UAPÉ está descrita no Capítulo 6.

Uma aplicação geográfica compreende uma seqüência de execução de atividades de coleta, tratamento, processamento e análise de dados, segundo um conjunto de algoritmos e processos definidos a priori. O usuário denomina este conjunto de "modelo".

O "modelo" estuda ou interpreta os relacionamentos quantitativos e qualitativos existentes entre as entidades geográficas de interesse da aplicação, selecionadas à luz do "modelo". Este pode ser estático ou dinâmico. Neste último caso, o "modelo" considera o comportamento dos relacionamentos entre as entidades no tempo, ou seja, considera a dimensão temporal, com o tempo fazendo parte da caracterização das entidades geográficas de interesse. Assim um "modelo" relativamente complexo, considera os aspectos estáticos, dinâmicos e funcionais das entidades geográficas e seus relacionamentos.

O GMOD contém basicamente três grupos de classes que são utilizadas para a modelagem estática (Convencional, Geo-Classe, Geo-Região, Representação, Plano de Informação), modelagem dinâmica (Regra) e a modelagem funcional (Processo). Além destas, outras classes, descritas no Capítulo 6, foram adicionadas para dar suporte à utilização do GMOD no ambiente UAPÉ.

O *modelo de objetos* abstrai a estrutura de dados das entidades geográficas sendo modeladas. Ele descreve as classes de objetos, seus relacionamentos e restrições estáticas e dinâmicas válidas para as classes e relacionamentos. As classes de suporte à modelagem de dados estão descritas na Seção 4.2.

O *modelo dinâmico* abstrai o comportamento temporal do sistema com relação a seus objetos. O componente principal do modelo são os *eventos* que provocam mudanças nos *estados* dos objetos, através de *ações* pré-definidas, dependendo de determinadas *condições*. As classes de suporte à modelagem dinâmica estão descritas na Seção 4.3.

O *modelo de processos* abstrai como os valores são computados independentemente de ordem, decisões ou estrutura de dados. O modelo de processos abstrai as funções que relacionam processos aos objetos. As classes de suporte à modelagem de processos estão descritas na Seção 4.4.

A Figura 4.1 mostra todas as classes básicas, sendo que as classes Processo e Regra são analisadas respectivamente nas seções de modelagem de processos e dinâmica e as demais na modelagem estática.

4.2 Modelagem Estática

4.2.1 Classes Básicas

GMOD classifica os dados geográficos em duas classes principais: Geo-Classe e Convencional, que levam em consideração a natureza descritiva e espaço-geométrica da informação e os relacionamentos entre estes dois aspectos. Geo-Classes são, por sua vez, subdivididas em duas classes: Geo-Objeto e Geo-Campo. As instâncias de uma Geo-Classe são objetos que têm algum componente espacial.

Uma classe Convencional descreve entidades que não são normalmente geo-referenciadas. Por exemplo, uma aplicação de planejamento de agricultura pode necessitar de informações sobre espécies de cana-de-açúcar, ou propriedades de fertilizantes. Esta informação é normalmente armazenada em classes convencionais. Eventualmente, instâncias dessas classes podem ser associadas a instâncias de Geo-Classe (por exemplo, uma plantação de cana-de-açúcar será descrita por uma instância de Geo-Classe que tem um componente convencional que descreve dados de colheita). Esta distinção entre classes convencionais e geo-classes permite que diferentes aplicações compartilhem dados não espaciais e favorece a reutilização de projeto e dados.

Tanto Geo-Classe como classe Convencional podem ser especializadas, usando o conceito de herança.

Em mais detalhes, as classes do modelo GMOD de suporte à modelagem de dados são:

- **CONVENCIONAL:** a classe Convencional modela objetos não geo-referenciados, ou seja, trata-se do conceito normal de modelagem de dados. Ela é a abstração de todas as classes cujas instâncias são elementos do mundo real que não possuem características geo-referenciáveis, mas que no contexto de uma determinada aplicação geográfica podem estar associados a entidades geográficas. Como exemplo, temos famílias associadas a regiões de cultura e produtos químicos associados a pontos de coleta de água subterrânea. Chamaremos tais elementos de objetos *convencionais*.

Um objeto convencional pode eventualmente ser espacializado associando-o a uma região (por exemplo, colocação de um poste em uma rede elétrica).

Em uma notação informal, a classe Convencional pode ser descrita como:

Convencional = $\langle \{x : \textit{Convencional}\} \rangle$

onde x é nome de atributo de um tipo que não contenha componentes espaciais.

- **GEO-REGIÃO:** a classe Geo-Região abstrai a extensão espacial de uma entidade geográfica. Esta classe é omitida em algumas propostas de modelos de dados geo-referenciados, sendo em muitos casos denominada Objeto Geométrico. No entanto,

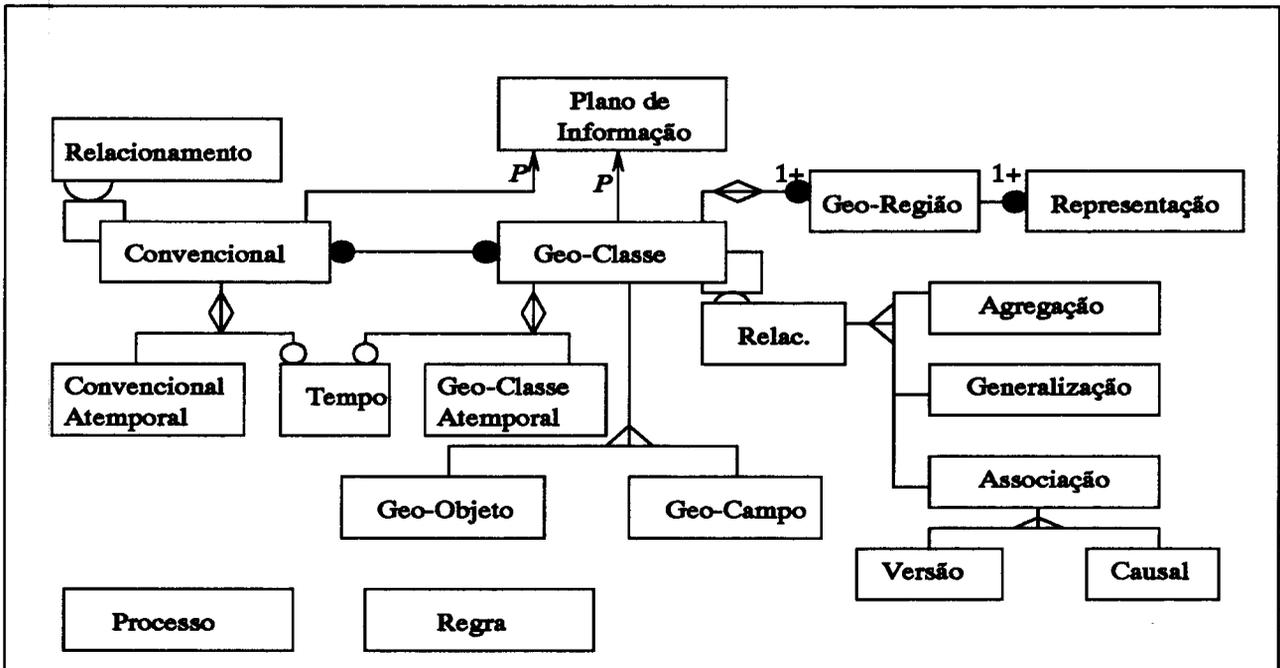


Figura 4.1: Modelo de Dados - GMOD

os conceitos são diferentes neste nível de abstração. Uma geo-região é descrita através de um conjunto de coordenadas interpretadas segundo uma referência espacial [OGI97].

Em uma notação informal, a classe Geo-Região pode ser descrita como:

Geo-Regiao= $\langle rep : Representacao, ref : Georeferencia \rangle$

onde *Representacao* é uma representação da entidade geográfica e *Georeferencia* é a informação sobre o sistema de coordenadas correspondente.

O GMOD permite que uma entidade geográfica seja associada a vários objetos da classe Geo-Região. Esta facilidade permite, por exemplo, que uma entidade possa estar associada a diferentes geometrias de região, dependendo da escala de trabalho, projeção ou precisão da coleta de dados. Esta associação determina o tratamento dos operadores espaciais no nível de representação. Como exemplo, o operador intersecção atua diferentemente entre polígonos e linhas e entre linhas.

- **GEO-CLASSE:** é uma classe abstrata que é especializada em duas outras classes - classe Geo-Campo e classe Geo-Objeto. A divisão entre geo-campos e geo-objetos é necessária para que o usuário possa representar visões de campos e objetos. Os

objetos de uma Geo-Classe sempre têm o atributo *local* cujo valor é uma região que descreve a sua extensão geográfica. Este atributo é um objeto da classe Geo-Região.

Usando notação informal,

Geo-Classe = $\langle \{a : \textit{Convencional}\}, local : \textit{Geo} - \textit{Regiao} \rangle$

onde *a* e *local* são nomes de atributos.

- **GEO-CAMPO:** a classe Geo-Campo é a abstração de todas as classes do modelo de campos, cujas instâncias são expressas como funções sobre uma variável. Obviamente, cada classe agrupa valores que se referem a uma mesma variável. Algumas classes podem se referir a variáveis que são distribuídas sobre a superfície de forma contínua (por exemplo, Altitude, Declividade, Uso do Solo); outras, a variáveis que são distribuídas de forma discreta (por exemplo, População, Habitação).

Matematicamente, um geo-campo pode ser definido por um mapeamento que abstrai a distribuição espacial de uma variável geográfica *V* sobre alguma região da superfície terrestre *A* [CCH⁺96]. Um geo-campo sempre tem um atributo *domínio* cujo valor é uma sub-região geográfica de *A*; um atributo *contra-domínio*, cujo valor pertence ao conjunto *V* e um atributo *geo-mapeamento* cujo valor é um mapeamento *f* de *A* para *V*. O conjunto *V* determina, portanto, a forma de classificação do espaço geográfico *A*. Os geo-campos se diferenciam conforme a definição de *A*, *V* e *f*.

Usando notação informal:

Geo-Campo = $\langle \{a : \textit{Convencional}\}, dominio : \textit{Geo} - \textit{Regiao}, contra - dominio : V, \langle local : \textit{Geo} - \textit{Regiao}, valor : V \rangle \rangle$

onde *a*, *dominio*, *contra - dominio*, *local* e *valor* são nomes de atributos e a tupla $\langle local, valor \rangle$ indica o mapeamento de sub-regiões do domínio em valores de *V*. Os componentes *a* e *dominio* são os componentes *a* e *local* herdados de Geo-Classe.

De acordo com [CCH⁺96] um geo-campo pode ser classificado em *Temático*, *MNT* (*Modelo Numérico de Terreno*) e *Sensoriamento Remoto*.

- *Geo-Campo Temático:* é definido por um geo-mapeamento $f: A \rightarrow V$ tal que *V* é um conjunto finito. Suponha que $V = \{v_1, v_2, \dots, v_n\}$. Então, como *f* é uma função, $\{f^{-1}(v_1), f^{-1}(v_2), \dots, f^{-1}(v_n)\}$ é um conjunto finito de sub-regiões disjuntas de *A* (Figura 4.2).
- *Geo-Campo MNT:* é definido por um geo-mapeamento $f: A \rightarrow V$, tal que *f* é definida apenas para um subconjunto de *A*. Intuitivamente, uma instância MNT é especificada por uma aproximação da distribuição espacial da variável *V* (por exemplo, topografia ou dados geofísicos) (Figura 4.3).

- *Geo-Campo Sensoriamento Remoto (SR)*: é definido por um geo-mapeamento $h: A \rightarrow V$ tal que h é a função composta das funções $g: A \rightarrow C$ e $f: C \rightarrow V$ onde C é um conjunto de células c que representam cada uma das partições de tamanho uniforme de A . Cada célula c_i representa uma partição ou subregião de A .

A função f associa a cada uma das células um valor V . Cada uma das partições de A define o domínio de uma instância SR (Figura 4.4).

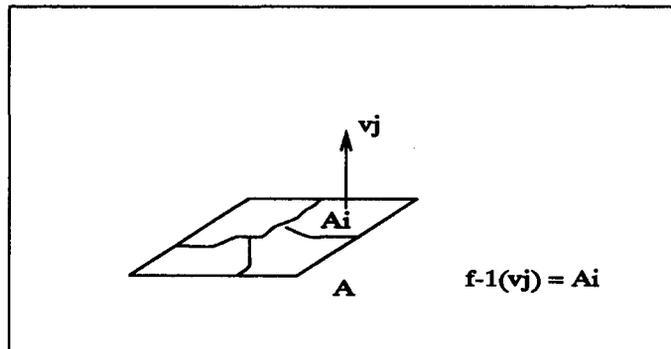


Figura 4.2: Exemplo de Geo-Campo Temático

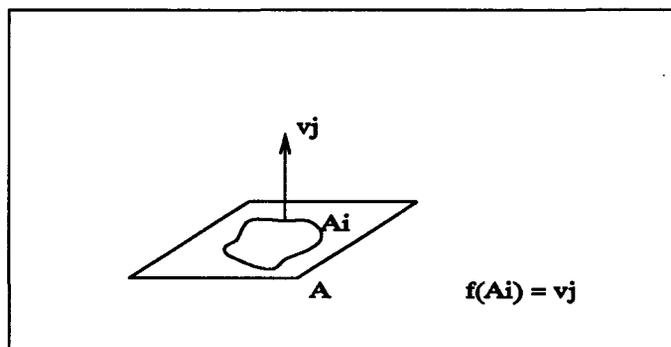


Figura 4.3: Exemplo de Geo-Campo MNT

- **GEO-OBJETO**: a classe Geo-Objeto é a abstração de todas as classes cujas instâncias são objetos identificáveis, correspondendo a uma entidade geográfica individualizável na realidade geográfica. Cada classe agrupa objetos de natureza homogênea do ponto de vista descritivo.

Usando notação informal,

Geo-Objeto = $\langle \{a : \text{Convencional}\}, \{b : \text{Geo-Classe}\}, \text{local} : \text{Geo-Regiao} \rangle$

onde a , b e $local$ são nomes de atributos.

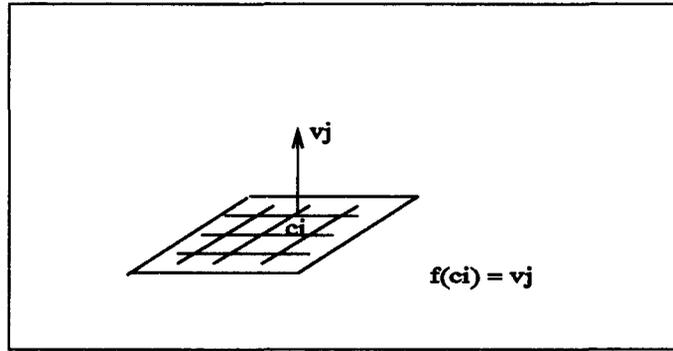


Figura 4.4: Exemplo de Geo-Campo Sensoriamento Remoto

- **REPRESENTAÇÃO:** esta classe abstrai todas as possíveis formas de representação de uma geo-região. O GMOD usa, no nível conceitual, as classes de representação propostas em [CCH⁺96], que pertencem a duas hierarquias principais, com origem nas classes REP_GEOOBJETO e REP_GEOCAMPO (figura 4.5).

A classe REP_GEOOBJETO, tem duas subclasses: REP_ELEMENTAR e REP_COMPLEXA. Uma instância de REP_ELEMENTAR define um elemento geométrico simples - ponto, linha e região com e sem buracos - em um sistema de coordenadas bem definido, representando a localização de apenas um geo-objeto. Uma instância de REP_COMPLEXA define um conjunto C de elementos geométricos simples ou complexos, em um sistema de coordenadas bem definido. O modelo permite associar elementos ou mesmo sub-conjuntos de C a geo-objetos distintos e interpretá-los como representações das localizações destes geo-objetos.

Para a classe REP_GEOCAMPO, o modelo oferece as seguintes subclasses principais: REP_TESSELACAO, REP_GRADE2D, GRADETRIANGULAR, SUBDIVISAOPLANAR, REP_ISOLINHAS e REP_PONTOS correspondendo aos mais freqüentes modelos de representação de campos [Goo93].

Note-se que a escolha do *modelo de representação* está estreitamente ligada a critérios de uso e volume de armazenamento como já mencionado anteriormente. Por exemplo, para controle de fluxo de tráfego é mais conveniente associar a extensão espacial de vias de acesso a classes de representação do tipo REP_GEOOBJETO do que a um conjunto de células adjacentes (correspondendo a representações do tipo REP_GEOCAMPO), embora as duas opções representem as mesmas entidades geográficas.

A manipulação direta das classes de representação se dá na fase de *implementação* da aplicação quando é gerado o projeto da aplicação que será convertido posteriormente para a linguagem do SIG subjacente.

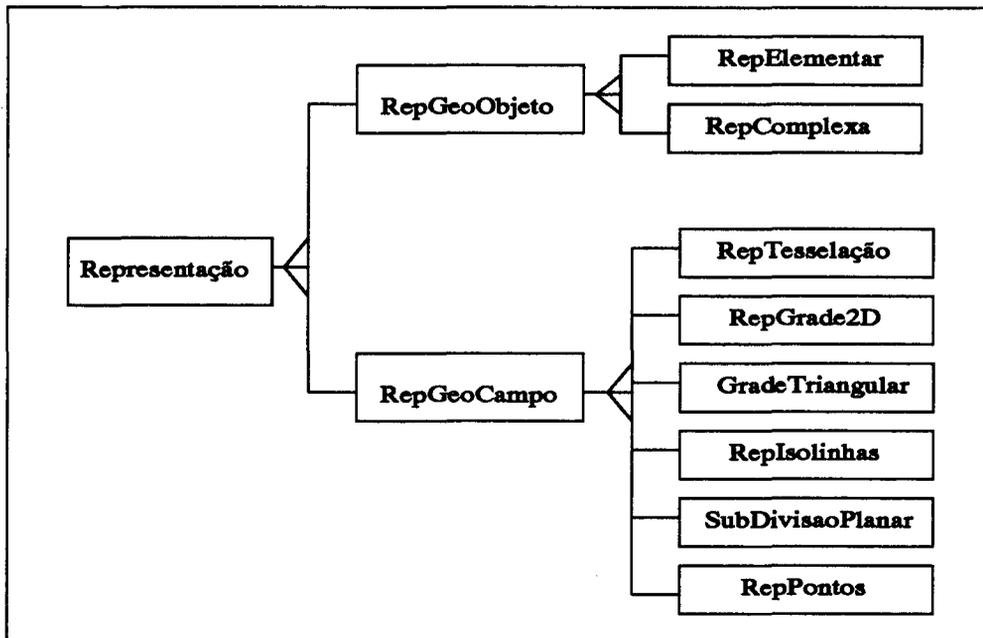


Figura 4.5: Classes de Representação

- **PLANO DE INFORMAÇÃO:** uma área de estudo em uma aplicação está associada a vários planos de informação (PI). Cada PI corresponde a uma determinada visão da área de estudo segundo um aspecto (tema ou conjunto de temas). Assim, se a área de estudo é uma cidade, podemos modelá-la segundo vários aspectos, como por exemplo, suas construções, sua rede de vias de acesso, entre outras. Planos de Informação podem conter tanto elementos modelados por campos quanto por objetos.

Cada instância da classe Plano de Informação corresponde à definição de uma *visão* sobre a área de estudo. A noção de visão aqui é semelhante ao conceito de visão em SGBD, ou seja, uma parte do banco de dados que é gerada a partir de uma consulta, tendo um esquema próprio.

Um plano de informação é definido através de seu esquema (nome, geo-classes e/ou classes convencionais envolvidas) e uma regra de formação (correspondente à consulta ou conjunto de consultas que produzem a extensão da visão). Esta regra de formação envolve a especificação de operações espaciais e/ou convencionais sobre as instâncias das classes envolvidas.

Esta especificação corresponde à definição de um processo que irá gerar o PI. A

Figura 4.1 modifica a notação do OMT para indicar que o relacionamento entre PI e geo-classes é um relacionamento de derivação (P). Processos e a geração de PI são comentados na seção referente à modelagem de processos (seção 4.4).

Cada instância da classe PI será especificada a partir do par $\langle Esq, c : Consulta \rangle$ onde Esq é a definição do esquema da visão correspondente ao PI e C é um conjunto de consultas sobre o banco de dados subjacente. Estas consultas podem combinar dados efetivamente armazenados e dados de outros PI.

- **TEMPO:** tempo é uma variável muito importante em qualquer aplicação geográfica e várias soluções têm sido propostas para sua inclusão na modelagem [FCF92, Lan93, Peu94]. Os resultados dos inúmeros trabalhos da área de modelos de banco de dados temporais estão descritos em [CT95].

A dimensão temporal pode ser representada de várias maneiras e descrita por diversos tipos de funções. Dois tipos de dimensão temporal são normalmente considerados: *tempo de transação* (descreve quando o dado é realmente armazenado no banco de dados) e *tempo válido* (descreve como os valores variam no mundo real).

O GMOD adota a especificação de tempo do SAIF [OGI97]. Esta especificação corresponde a uma hierarquia de classes com raiz na classe Tempo, permitindo ao usuário escolher as características temporais de suas entidades. A Figura 4.6 mostra as principais classes temporais do SAIF.

Resumidamente, elas têm a seguinte semântica:

- *Tempo*: é um objeto temporal que representa uma porção específica da linha do tempo.
- *ObjTempo*: é a posição do objeto temporal na linha do tempo, isto é, o horário, ou a data ou a duração indicando a existência do objeto.
- *RefTemporal*: se refere à base dos valores de tempo: UTC - Universal Coordinated Time, GPS - Global Positioning System ou referência local (por exemplo, data do sistema).

Uma classe do GMOD pode ser definida pelo usuário como *temporal* (que significa que ao usuário interessa a evolução do fenômeno) ou *atemporal*. No primeiro caso, à definição da classe é incorporado um componente que é um objeto do tipo Tempo. Por exemplo, seja $C = \langle Tipo \rangle$ a definição do tipo da classe não temporal C . Sua temporalização corresponde à geração de

$CT = \langle Tipo, t : Tempo \rangle$ onde,

t é um objeto temporal da hierarquia de classes com raiz em *Tempo*.

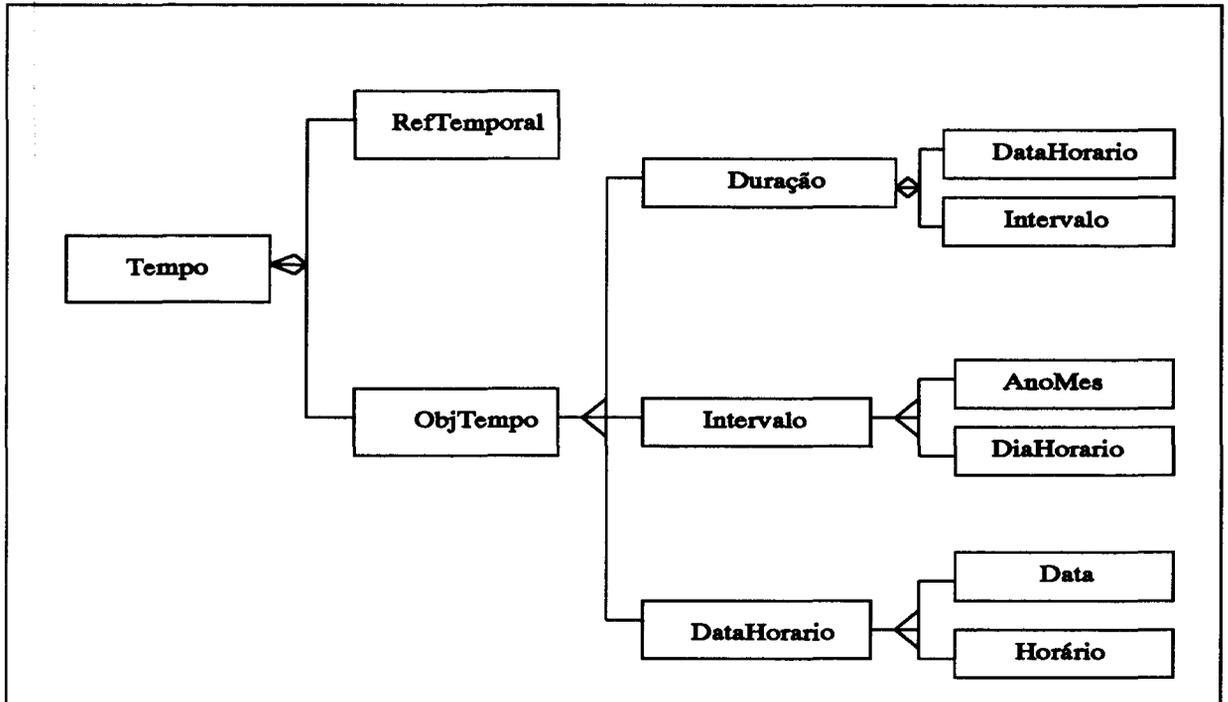


Figura 4.6: Hierarquia de Classes Temporais

Esta proposta é similar à proposta de [SW95] mas, diferentemente desta, permite a variação temporal não apenas para dados espaciais, mas também para os dados convencionais.

4.2.2 Relacionamentos no GMOD

A modelagem no GMOD considera dois tipos de relacionamentos: explícitos e implícitos. O primeiro precisa ser especificado pelo usuário enquanto que um relacionamento implícito é todo aquele que pode ser derivado de dados existentes (por exemplo, relacionamentos topológicos) [CSE94, MC95].

A maioria dos modelos de dados geográficos ignora a possibilidade da modelagem de relacionamentos entre fenômenos do mundo real. Embora muitas propostas sejam baseadas no paradigma da modelagem Entidade-Relacionamento (ER) (por exemplo, [FR93]), este modelo não oferece recursos semânticos suficientes para descrever entidades geográficas. Com o aparecimento do modelo orientado a objetos, algumas restrições semânticas desapareceram, ficando mais natural a representação de relacionamentos complexos que aparecem tipicamente em aplicações geográficas.

Os relacionamentos encontrados nos modelos ambientais podem ser classificados em três grandes categorias [Goo93]:

- **escalares** - relacionamentos escalares associam entidades em uma razão de medida

definida (por exemplo, usando uma escala determinada). Exemplos são os mapas topográficos.

- **conceituais ou qualitativos** - são os relacionamentos realizados no nível conceitual
- **matemáticos ou quantitativos** - são os relacionamentos entre instâncias de classes, definidos como fórmulas ou funções. Relacionamentos matemáticos podem ser divididos ainda como:
 - *estatísticos* - envolvem pelo menos um processo estocástico representado por uma ou mais variáveis randômicas; podem ser divididos em "steady-state" ou dinâmicos (função do tempo)
 - *determinísticos* - não têm uma variável randômica; podem ser divididos em "steady-state" (os relacionamentos podem ser representados por expressões algébricas para estudo de diagnósticos) ou dinâmicos (função do tempo - representados por equações diferenciais). Relacionamentos determinísticos podem ser classificados em [AEB95]:
 - * *determinação própria*: uma condição é determinada unicamente por uma condição anterior,
 - * *determinação causal*: ocorre quando existe uma conexão produtiva ou genética entre duas condições,
 - * *interação*: é definida como uma forma de causa mútua ou recíproca,
 - * *determinação mecânica*: é o mesmo que determinação própria quantitativa, exceto que o conseqüente é determinado por um antecedente em combinação com outros fatores, como causas externas,
 - * *determinação estatística*: ocorre quando um conseqüente é determinado pela ação conjunta de vários antecedentes independentes,
 - * *determinação estrutural*: refere-se à determinação de partes pelo todo.

Neste trabalho estamos interessados em explorar os relacionamentos conceituais e matemáticos existentes entre os diversos fenômenos geográficos. O entendimento desses relacionamentos permite uma modelagem mais rica dos fenômenos estudados em aplicações geográficas. A seguir discutimos os relacionamentos conceituais. Relacionamentos matemáticos são detalhados nos modelos dinâmico e de processos.

O GMOD considera três tipos principais de relacionamentos explícitos conceituais: *generalização/especialização, agregação, associação*.

Generalização/Especialização

O relacionamento de *Generalização/Especialização* permite que o usuário generalize/derive novas entidades a partir de outras já especificadas. Este relacionamento é inerente à modelagem orientada a objetos através do mecanismo de "herança" (is-a).

Em aplicações convencionais, a generalização se aplica em diversas situações onde se deseja, por exemplo, abstrair entidades ou classes mais genéricas de tal forma a compartilhar semelhanças entre classes, preservando suas diferenças. Inversamente, a especialização permite derivar classes a partir de outras já definidas reutilizando estruturas de dados e comportamento, ou mesmo redefinindo-os.

Exemplos de utilização deste tipo de relacionamento nas visões de objetos e de campos estão ilustrados nas Figuras 4.7 e 4.8 respectivamente. Uma classe Construção pode ser especializada em classes específicas dependendo do tipo de sua atividade: Residencial, Comercial ou Serviço. Uma estrada pode ser especializada em estradas principais e estradas secundárias.

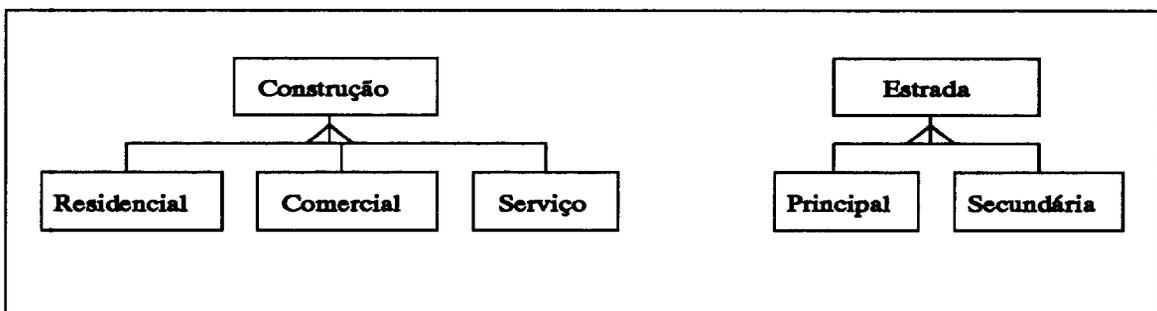


Figura 4.7: Generalização na Visão de Objetos

No exemplo de relacionamento de generalização entre geo-campos a partir do geo-campo Solo (Figura 4.8) deseja-se abstrair somente as regiões de solo do tipo apropriado para o cultivo de arroz. Note-se que toda a especificação da classe Solo se aplica à classe Solo_Cultivo_Arroz, com a diferença que os valores do domínio e o contra-domínio foram ajustados para os valores válidos.

Agregação

O relacionamento de *Agregação* permite construir objetos complexos e é suportado pelo mecanismo de "composição" do modelo orientado a objetos (part-of). Por exemplo, um arquipélago pode ser descrito como uma agregação de ilhas, e uma bacia hidrográfica pode ser descrita como o conjunto de seus rios. Ainda em outro exemplo, quando não

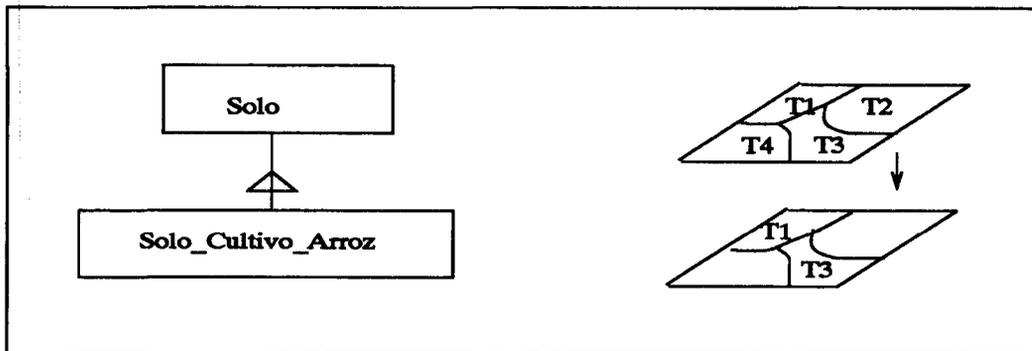


Figura 4.8: Generalização na Visão de Campos

explicitado, o atributo *local* de uma entidade geográfica agregada é computado a partir do atributo *local* dos objetos componentes.

O conceito de agregação em aplicações convencionais tem sido utilizado intensivamente e na prática tem se traduzido na composição de atributos para a especificação de relações do modelo relacional. No modelo orientado a objetos o conceito de agregação foi estendido para a criação de objetos complexos a partir da composição de objetos já existentes (Figura 4.9).

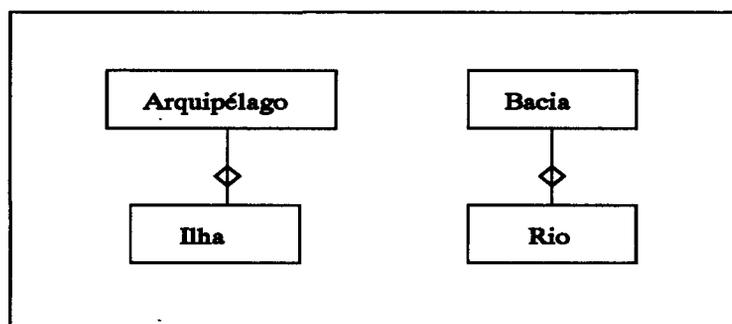


Figura 4.9: Agregação em Aplicações Geográficas

Associação

Associações correspondem aos relacionamentos genéricos da modelagem ER, permitindo que usuários estabeleçam diferentes tipos de conexões entre entidades. No GMOD algumas associações são pré-definidas como as associações entre geo-classes e geo-regiões, geo-classes e classes convencionais e geo-regiões e classes de representação. Estas associações já estão definidas previamente e não podem ser alteradas pelo usuário.

Ao definir classes, o usuário estabelece associações adicionais que são dependentes da aplicação e por isto, usualmente, são denominadas *relacionamentos ou associações conceituais*. Estes relacionamentos estão representados de forma genérica pela classe Relacionamento associada a Geo-Classe e Classe Convencional. A implementação de relacionamentos como classes tem diversas vantagens como mencionado em [KT92]: relacionamentos podem ter métodos associados a eles, relacionamentos podem ter atributos associados e inclusive relacionar-se entre si.

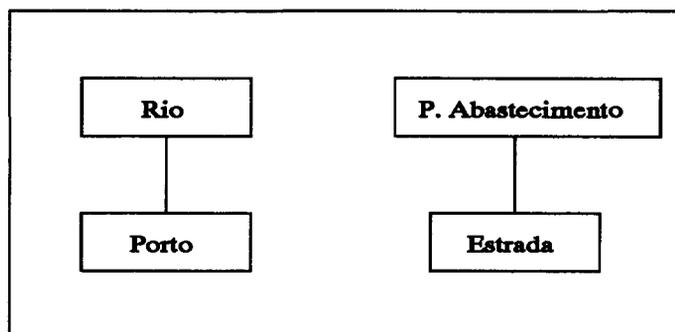


Figura 4.10: Associações Simples entre Geo-Classes

Alguns exemplos de associações entre geo-classes estão na Figura 4.10: rios e os seus portos; e estradas e postos de abastecimento.

Todos os tipos de relacionamentos não diretamente suportados pela modelagem orientada a objetos podem ser vistos como um tipo de associação, mas distinguimos duas variações semânticas em especial - *versionamento* e *relacionamento causal* - dada a sua importância do ponto de vista do usuário. Relacionamentos causais modelam a dinâmica do mundo real, sendo por isto discutidos na próxima seção.

Relacionamento de Versão

Relacionamentos de versão permitem que usuários estabeleçam conexões entre versões de um único conceito. Quando o usuário estabelece relacionamentos de versão para uma classe do modelo, o banco de dados geográfico da aplicação é especificado de forma a permitir o armazenamento de múltiplas versões para os objetos ou instâncias da classe. Com isto, o usuário pode conectar cenários distintos derivados para a mesma área de estudo, para um projeto de planejamento ambiental, por exemplo. Cada cenário pode ser o resultado de diferentes simulações, e os relacionamentos de versão auxiliam a navegação entre estes cenários. O gerenciamento de versões de dados geográficos e sua aplicação ao conceito de cenários alternativos é discutido em [CM97].

Por exemplo, seja $C = \langle \text{Tipo} \rangle$ a definição do tipo da classe não versionada C . Seu versionamento corresponde à geração de

$CV = \langle \text{Tipo}, VS : \text{long} - \text{integer} \rangle$ onde,

long - integer é um inteiro que controla a versão dos objetos da classe *CV* (classe *C* versionada). Há vários algoritmos para criação deste valor, dependendo do tipo de gerenciamento de versões utilizado. O GMOD se baseia na proposta de versões de [Cur97], onde a geração dos valores de *VS* segue o mecanismo de *database versions* de [CJ90].

Os exemplos da Figura 4.11 indicam que as instâncias das classes Indústria e Uso do Solo devem ser estruturadas de forma a poderem ser versionadas.

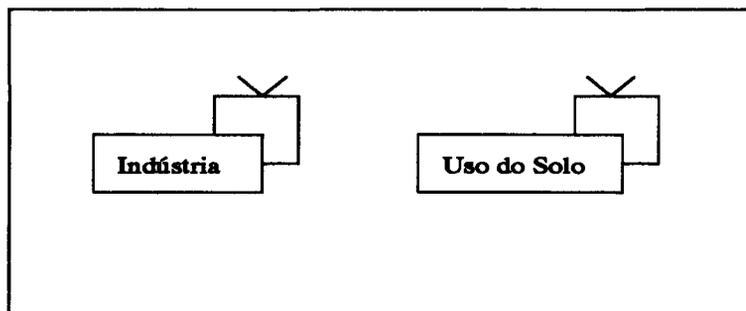


Figura 4.11: Associação - Versão

4.3 Modelagem Dinâmica

O "modelo" do usuário considera basicamente, como já ressaltamos, as entidades geográficas de interesse da aplicação (o que), os processos que operam sobre os atributos das entidades (como) e os eventos que disparam a execução dos processos, sob determinadas condições (quando). A modelagem dinâmica é o ponto chave das simulações realizadas por usuários em aplicações ambientais. No entanto, propostas de modelos de dados voltados a aplicações deste tipo ignoram esta faceta. A introdução explícita da modelagem dinâmica é uma das contribuições do GMOD.

Em aplicações geográficas a modelagem dinâmica é o ponto central do "modelo" do usuário. Simplesmente identificar ou abstrair um relacionamento entre as entidades (por exemplo, Uso do Solo e Precipitação) é muito pobre no contexto destas aplicações. Este relacionamento é quantitativo e cumulativo no tempo, ou seja, o estado atual de uma entidade é ditado pelos seus estados anteriores.

A modelagem dinâmica abstrai o comportamento temporal da aplicação com relação a seus objetos. O componente principal do modelo são os *eventos* que provocam mudanças nos *estados* dos objetos, através de *ações* pré-definidas, dependendo de determinadas *condições*.

A tripla $\langle \text{evento}, \text{condicao}, \text{acao} \rangle$ pode ser mapeada para o conceito de *regra de*

produção de bancos de dados ativos [Buc94, CAM93]. Estes são sistemas que permitem a execução dinâmica de ações a partir de eventos externos ou internos, se verificada a condição. Esta tese propõe assim, que a modelagem dinâmica seja apoiada por um conjunto de regras incorporadas ao modelo da aplicação. Para a modelagem dinâmica, o GMOD implementa a classe Regra.

- **REGRA:** uma regra no GMOD não é associada explicitamente às demais classes do banco de dados, mas criada separadamente, podendo ser adicionada ou removida sem interferir na definição destas classes. Uma regra corresponde a uma tupla $\langle E, C, A \rangle$, onde E especifica um evento simples ou composto, C especifica uma consulta a ser realizada sobre o banco de dados e A é um método ou função que deve ser executada dependendo de E e C . A semântica correspondente é "On E , if C , then execute A ". Eventos simples são especificados como pares $\langle m, c \rangle$ denotando que um método m aplicado à classe c sinaliza um evento; c pode ser uma Geo-Classe ou uma classe Convencional. Eventos compostos são obtidos a partir de eventos simples, usando os conectores *AND*, *OR* e *NOT*. Várias linguagens de especificação de eventos já foram propostas na literatura. Nesta tese, adota-se o modelo de [Sam94] para a especificação destes eventos.

A semântica dessa classe estende a proposta de [MC95]. Nesta última, regras são usadas apenas para a manutenção de integridade, ao passo que aqui a noção de eventos, condições e ações é mais ampla.

De fato, no GMOD, regras podem ser utilizadas não apenas para a especificação de restrições de integridade, mas também para a especificação geral da dinâmica das aplicações. Um caso especial do uso de regras, como se verá a seguir, é o de especificação de relacionamentos causais. Um exemplo de restrição de integridade em aplicação ambiental é "Toda rocha precambriana tem ao menos 4 milhões de anos". Esta restrição se aplica a datações de objetos da classe Convencional Rocha.

As restrições mais comumente encontradas na literatura para dados geográficos são as restrições topológicas [CSE94, CDvO93], que definem relacionamentos binários topológicos entre representações vetoriais de geo-objetos. Um exemplo é "As fronteiras de duas fazendas vizinhas são adjacentes".

Relacionamento Causal

Tão importantes como os relacionamentos de versão são os relacionamentos *causais* que estabelecem conexões de causa-efeito entre os fenômenos modelados. Por exemplo, o tipo de solo de uma área afeta diretamente as características da vegetação. Relacionamentos causais são importantes no estudo de impactos ambientais onde a influência das atividades humanas é estudada de acordo com o seu impacto na natureza. Independentemente das atividades humanas, existem muitos relacionamentos causais entre elementos da natureza

que também são amplamente estudados, considerando o tempo (escala espaço-temporal) e a abrangência (local, regional ou global).

Normalmente, relacionamentos de causa-efeito são modelados através de *relacionamentos funcionais* entre variáveis e requerem uma abordagem quantitativa para a modelagem, embora sejam utilizados algumas vezes para inferência qualitativa sobre o comportamento do sistema. Assim, o usuário deve especificar não somente as variáveis que se afetam mutuamente, mas também formular quantitativamente o relacionamento funcional exato entre elas. Nestes modelos, conexões causais entre variáveis são modeladas explicitamente através de *fórmulas matemáticas*.

Relacionamentos causais também indicam a necessidade da modelagem dinâmica das variáveis envolvidas. Por exemplo, a Figura 4.12 indica que, no contexto da aplicação geográfica sendo modelada, interessa estudar os efeitos sobre os objetos da classe Uso do Solo, quando ocorrem mudanças nos estados dos objetos da classe Vegetação. Estes relacionamentos são importantes porque implicam no estabelecimento de *restrições de integridade referencial* não apenas conceituais, mas também espaço-temporais. No exemplo, alterações na extensão da cobertura vegetal (propriedade espacial) afetam ao longo do tempo as características do solo subjacente (propriedade convencional).

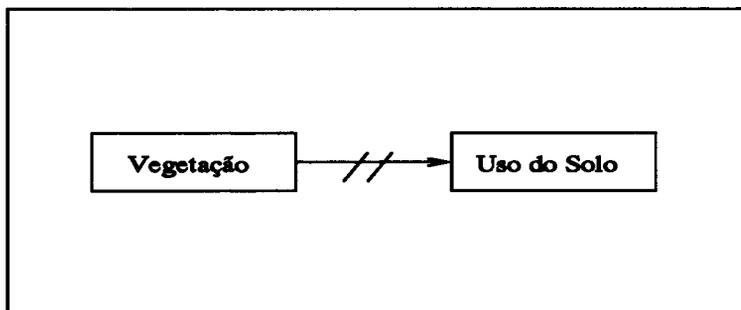


Figura 4.12: Relacionamento Causal

Na metodologia OMT, o modelo dinâmico de uma aplicação é representado graficamente por um conjunto de diagramas de estados, como mostrado na Figura 4.13. A figura ilustra que a condição do solo em termos de umidade se altera em função da quantidade de chuva, da própria condição do solo e de sua vegetação subjacente.

Modelando a dinâmica da aplicação através de diagrama de estados, estabelecemos as regras implícitas do "modelo" do usuário com a identificação dos eventos que provocam alterações nos estados dos objetos e da influência que esta alteração de estado tem sobre o estado de outros objetos relacionados.

Para entendermos como podemos extrair o conjunto de regras da aplicação a partir do modelo dinâmico segundo a metodologia OMT, vamos definir os componentes abstraídos

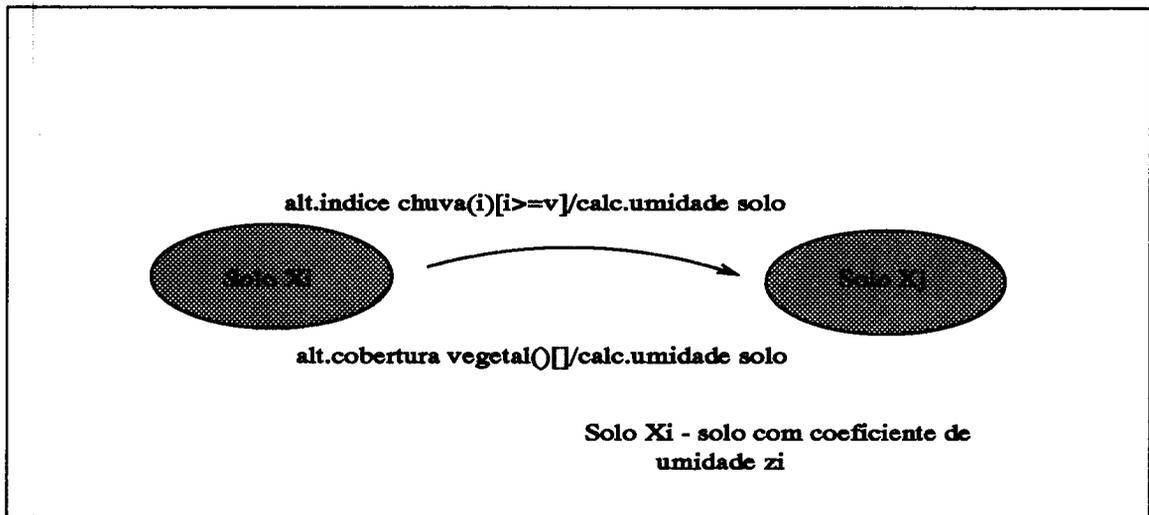


Figura 4.13: Exemplo de Diagrama de Estados

por um diagrama de estados.

Os valores dos atributos e relacionamentos de um objeto caracterizam o seu *estado*. Através do tempo, os objetos estimulam-se mutuamente resultando numa série de mudanças em seus estados. Um estímulo individual de um objeto para o outro é um *evento*. A resposta a um evento depende (*condição*) do estado do objeto que o recebe e pode causar uma mudança de estado ou o envio de um outro evento.

Assim, um evento é um sinal que alguma coisa aconteceu. Eventos são interpretados no contexto de estados. Uma transição entre estados representa uma resposta a um evento, incluindo o próximo estado e possíveis *ações* e eventos enviados a outros objetos. Uma condição é uma função booleana que controla se uma transição pode ocorrer ou não. Um diagrama de estado é um grafo de estados e transições rotuladas por eventos. Uma ação é uma operação ativada em resposta a um evento. O nome do estado da entidade geográfica é escrito dentro da figura ovalada. O nome de um evento é escrito sobre a indicação de transição (flecha) e pode opcionalmente ser seguido por um ou mais atributos entre parêntesis. A condição pode ser listada dentro da chave depois do nome do evento. A ação é indicada na transição precedida pelo caracter /.

Assim, a partir de um diagrama de estados podemos extrair uma série de regras que serão incorporadas à especificação da aplicação.

No exemplo da figura 4.13 podemos deduzir a seguinte regra:

eventos: índice de chuva atinge valor igual a i mm e alteração na área de cobertura vegetal

condição: solo com índice de umidade igual a z_i (estado X_i do solo) e índice de chuva maior ou igual a v , sendo v um valor pré-definido

ação: executar o processo que calcula novo índice de umidade do solo (alteração do estado do solo para X_j)

Usando este paradigma, cada mudança de estado é descrita por uma *regra de produção*. No exemplo da restrição de integridade "As fronteiras de duas fazendas vizinhas são adjacentes", esta pode ser traduzida para:

evento: *update*(Fazenda1)

condição:

cross(*boundary*(Fazenda1), *boundary*(Fazenda2)) *or disjoint*(Fazenda1, Fazenda2)

ação: *abort*

Esta regra mostra de forma simplificada predicados espaciais topológicos. *Boundary* é o nome dado na literatura à fronteira de um objeto vetorial. *Cross* e *disjoint* são relacionamentos topológicos tradicionais. A regra pode ser generalizada para conjuntos de fazendas substituindo-se a condição por uma consulta sobre o conjunto, que retorna "true" se a atualização viola o relacionamento "adjacent". O trabalho [Cil96] implementou um banco de dados orientado a objetos ativo para gerenciar integridade topológica utilizando o modelo GMOD.

4.4 Modelagem de Processos

Os processos ambientais no mundo real são tipicamente tri-dimensionais, temporais e complexos. Exemplos de processos ambientais do mundo real são transferências de calor e radiação, hidrodinâmica dos fluidos e desenvolvimentos ecológicos. Tal complexidade pode incluir comportamentos não-lineares, componentes estocásticos e ciclos de *causa-efeito* (feedback) através de múltiplas escalas de espaço e tempo (resposta rápida - trocas biofísicas; escalas intermediárias - processos biológicos e biogeoquímicos; escala longa - mudanças na comunidade, composição e estrutura do ecossistema).

Pode existir um entendimento *qualitativo* razoável de um processo em particular, mas o entendimento *quantitativo* ainda é limitado. Como exemplo, um dos maiores desafios atualmente é determinar os relacionamentos quantitativos entre a biosfera e os componentes físicos e químicos do sistema climático global.

Como vimos, o "modelo" do usuário considera basicamente as entidades geográficas de interesse da aplicação (o que), os processos que operam sobre os atributos das entidades (como) e os eventos que disparam a execução dos processos, sob determinadas condições (quando).

Os processos são definidos por um conjunto de operações que desempenham diversas funções no modelo. Por exemplo, geram informações a partir de cálculos sobre os valores

de atributos das entidades envolvidas ou atualizam atributos dessas entidades.

Esses processos estão embutidos principalmente nos relacionamentos matemáticos ou quantitativos do modelo de objetos. Como exemplo, um relacionamento causal indica que a alteração de estado de uma entidade geográfica implica em alteração de estado das entidades geográficas relacionadas no modelo (semelhante ao conceito de integridade referencial). Suponha que estamos trabalhando com fluxo de escoamento de água na superfície e distribuição de chuva numa região. Após uma tempestade, os parâmetros que descrevem o fluxo atual devem ser revistos e alterados. Isso determina a execução de modelos matemáticos que levam em consideração o estado de escoamento da água na superfície antes e depois da chuva (temporal), além de considerar outros fatores como declividade e potencial de evaporação.

A modelagem de Planos de Informação também supõe a execução de um processo sobre as entidades geográficas envolvidas. Neste sentido, PI constituem entidades derivadas a partir de outras entidades primitivas do modelo. Como exemplo, a geração de um PI de Declividade a partir de um geo-campo MNT envolve a execução de operadores estatísticos e espaciais.

Para suportar a modelagem de processos, o GMOD implementa a classe Processo que suporta a especificação dos processos da aplicação. As ações especificadas na construção de regras são referências a processos especificados em instâncias da classe Processo.

- **Processo:** Usando uma notação informal podemos definir um processo como,

Processo= $\langle \textit{nome} : \textit{String}, \{\textit{entradas}\}, \{\textit{saidas}\}, \textit{funcao} : \textit{String}, \{\textit{Regras}\} \rangle$

onde *nome* é o nome do processo, *entradas* e *saidas* definem os parâmetros de entrada e saída respectivamente, *função* define o efeito do processo e pode ser descrita diretamente em linguagem natural, equações matemáticas ou alguma outra linguagem e *Regras* representam restrições de processamento que dizem respeito às condições que devem ser estabelecidas para que o processo seja executado.

Na metodologia OMT, o modelo de processos é representado graficamente pelo diagrama de fluxo de dados, como mostrado na Figura 4.14. A figura ilustra o processo de cálculo do novo coeficiente de umidade do solo em função do índice de chuva e da área de cobertura vegetal (vide Figura 4.13).

4.5 Operações no GMOD

No Capítulo 3, *objeto* foi definido como uma instância de uma classe e caracterizado por seu *estado* (conjunto de valores de atributos) e *comportamento* (conjunto de operações ou métodos que podem ser aplicados ao objeto). De acordo com o modelo de objetos

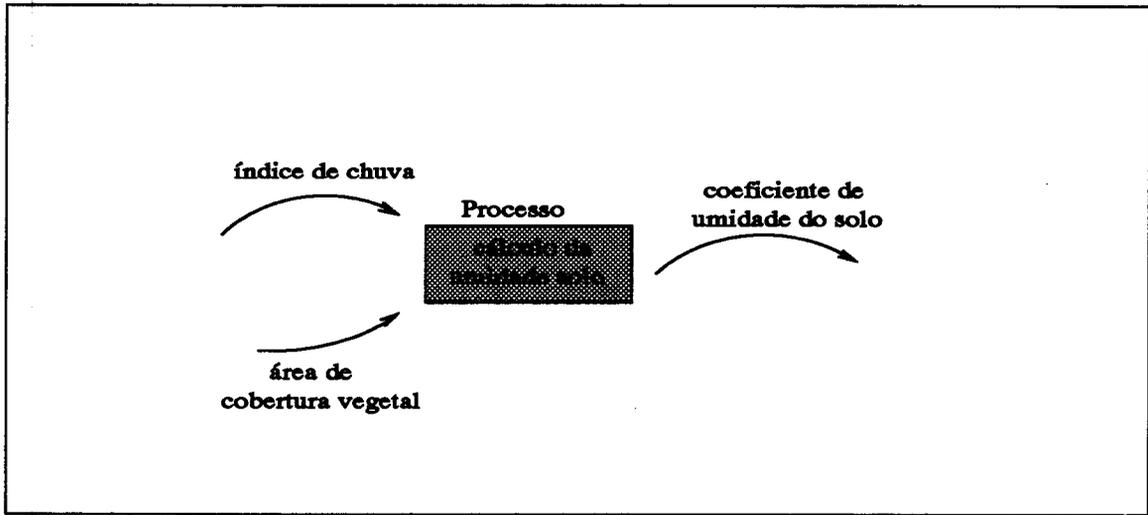


Figura 4.14: Exemplo de Processo - Cálculo da Umidade do Solo

da OMG [Kim95], uma operação tem uma *assinatura*. A assinatura inclui o nome da operação, uma lista de parâmetros e uma lista de valores-resultado, se houver algum. Mais formalmente, uma operação Ω tem a assinatura:

$$\Omega: (x_1 : \sigma_1, x_2 : \sigma_2, \dots, x_n : \sigma_n) \rightarrow (y_1 : \rho_1, y_2 : \rho_2, \dots, y_m : \rho_m)$$

Ω é o nome da operação. A assinatura especifica $n \geq 1$ parâmetros com nomes x_i e tipos σ_i e $m \geq 0$ resultados y_i de tipos ρ_i .

Cada classe de objetos \mathcal{C} tem um conjunto de operações $\text{Ops}(\mathcal{C}) = \Omega^{\mathcal{C}}_1, \Omega^{\mathcal{C}}_2, \dots$

Normalmente, em aplicações geográficas, as operações associadas às entidades são operações que atuam sobre as características espaciais do objeto. Exemplos destes operadores são [TS94, CSE94, Cil96]:

- *adjacent* - testa se uma região tem fronteira comum com outra região
- *contains* - testa se uma região contém outra região
- *area* - calcula a área de uma dada região
- *perimeter* - calcula o perímetro de uma dada região

Operadores espaciais são plenamente suportados por SIG e por isso não serão especificados na instanciação do GMOD para uma determinada aplicação geográfica. Um exemplo de operador voltado a um domínio específico é o cálculo do PH médio de um rio, num determinado período. Os processos especificados na modelagem de processos são

convertidos em métodos e funções no GMOD. O cálculo de umidade do solo, por exemplo, é transformado em uma função que é embutida em um método que atualiza o atributo "índice de umidade" do geo-objeto "Solo".

Deve-se ressaltar que a especificação de operações é uma tarefa muito complexa no contexto do usuário desenvolvedor de aplicações geográficas, já que envolve o conhecimento de linguagem e lógica de programação. Além disso, a linguagem utilizada deve ser suportada pelo SIG subjacente. Devido à complexidade de implantação, muitos trabalhos enfatizam somente a estrutura dos objetos, não tratando explicitamente dos métodos associados a classes. Estes métodos são, no entanto, mencionados quando se discute a implementação de modelos [SV92, Bat97].

4.6 Resumo

Este capítulo descreveu em detalhes o modelo de dados geográficos GMOD. O GMOD se diferencia dos demais modelos por considerar, além da modelagem de dados propriamente dita, a modelagem de relacionamentos, a modelagem dinâmica e a modelagem de processos. A modelagem dinâmica é suportada pela instanciação da classe Regra. A modelagem de processos é suportada pela especificação de métodos pré-definidos no GMOD, pela especificação de processos associados às ações do modelo dinâmico e pela especificação de Planos de Informação. Os processos identificados são descritos na classe Processo. A modelagem temporal é suportada pela classe Tempo adotada a partir do modelo SAIF. Modelagem dinâmica e de processos exigem suporte de bancos de dados ativos.

Capítulo 5

Metodologia de Desenvolvimento de Aplicações Ambientais

Uma das contribuições deste trabalho é desenvolver ferramentas que aproximem o SIG do usuário. Para isto foi necessário, antes de mais nada, entender as formas de trabalho de vários perfis de usuários de aplicações ambientais e mapear estes procedimentos para alguma seqüência padrão de etapas. Esta seqüência, apresentada neste capítulo como "metodologia do usuário", representa, com alto nível de abstração, os passos executados por uma grande variedade de usuários ao desenvolver suas aplicações. Esta "metodologia do usuário" é uma das bases para o desenvolvimento do UAPÉ, apresentado no Capítulo 6. O entendimento desta seqüência de etapas facilitou muito a especificação do ambiente e garantiu que esta especificação satisfaça a um conjunto mínimo de requisitos de usabilidade por parte do usuário.

Este capítulo apresenta desta forma, a Metodologia de Desenvolvimento de Aplicações Ambientais proposta nesta tese. Esta metodologia foi desenvolvida a partir da experiência em especificação e manuseio de aplicações de planejamento ambiental. A Seção 5.1 apresenta o contexto onde se inserem metodologias de desenvolvimento de software. A Seção 5.2 descreve em alto nível a metodologia proposta. A Seção 5.3 apresenta um exemplo de uso da metodologia no desenvolvimento de uma aplicação geográfica.

5.1 Contexto

O processo de desenvolvimento de aplicações geográficas ainda está extremamente defasado em relação ao encontrado no campo de aplicações convencionais.

A ausência de uma disciplina de Engenharia de Software compromete o processo de concepção de uma aplicação geográfica na medida em que a falta de formalismo dificulta o controle de qualidade, a reutilização de dados, a documentação das decisões e produ-

tos intermediários, entre outros. Além do mais, mesmo abordagens de desenvolvimento de sistemas, como as metodologias orientadas a objetos, quando utilizadas no domínio das aplicações geográficas, não produzem resultados satisfatórios. Isto acontece principalmente devido às propriedades espaciais, topológicas e temporais que não podem ser tratadas de forma adequada.

Em geral, não há a menor preocupação com a modelagem de dados e a modelagem dinâmica e de processos são realizadas de forma isolada através da execução de modelos de comportamento de determinados fenômenos estudados pela aplicação.

A modelagem dos dados do mundo real é, hoje em dia, totalmente influenciada pela forma como os dados são representados dentro do SIG (basicamente raster e vetor). A visão cartográfica do mundo real influencia a decisão de quais dados devem ser coletados e a forma como serão organizados e integrados. Nesta visão, os fenômenos geográficos do mundo real são representados por "modelos", por exemplo, modelos matemáticos. Um "bom" modelo é o mais simples que prevê o comportamento do mundo real para o fenômeno de interesse, de forma correta e consistente. Os "modelos" são criados para prever o comportamento de certos aspectos do mundo real através da descrição dos relacionamentos entre os dados de interesse.

Atualmente, as metodologias utilizadas são específicas para cada tipo de domínio como planejamento ambiental, planejamento urbano, agricultura, entre outros. A decisão de quais métodos empregar considera principalmente as características da aplicação, os seus objetivos, a área de estudo e o perfil da equipe técnica envolvida no projeto.

Toda a concepção de uma aplicação geográfica é motivada pela necessidade de uma avaliação mais sistêmica e menos empírica sobre uma determinada extensão geográfica, em função dos objetivos a serem atingidos. Os objetivos estão ligados a alguma linha de trabalho, como por exemplo, planejamento ambiental, estudos de impacto ambiental, programas racionais de utilização e manejo de recursos naturais, planejamento urbano, planos diretores, dentre outros.

De forma geral, as fases tradicionais do ciclo de vida de software (análise, projeto, implementação e manutenção) são perfeitamente válidas para uma metodologia de desenvolvimento de aplicações geográficas. No entanto, a característica básica desejável nesta metodologia é a flexibilidade na escolha de diferentes estratégias metodológicas, específicas em cada uma das fases previstas. Chamamos de *estratégia metodológica* um conjunto de métodos que devem ser utilizados de forma sequencial ou paralela, escolhidos de acordo com os objetivos da aplicação e características geográficas da área de estudo. Cada área geográfica representa um estudo de caso específico e pode necessitar de uma estratégia metodológica diferenciada para a concepção de um mesmo tipo de aplicação geográfica.

A diversidade de estratégias metodológicas utilizadas em aplicações geográficas aumenta dia a dia. [Mor92], por exemplo, inventariou cerca de 100 estratégias para aplicações

de planejamento ambiental. Muitas delas são heurísticas limitadas pelo domínio da aplicação, fontes de dados disponíveis e pela abordagem do problema pelo usuário, influenciando o modelo de processo utilizado, sem contar com as restrições impostas pelo próprio SIG subjacente.

O projetista de uma aplicação precisa interpretar as necessidades do usuário e mapear a realidade geográfica em dados de entrada, que são manipulados segundo o "modelo" implementado no SIG, ou mesmo integrados e analisados manualmente. Uma metodologia deste tipo é basicamente orientada ao tratamento dos dados geográficos de interesse da aplicação utilizando-se métodos e técnicas específicas [San94]. Em muitos casos, não é possível vislumbrar uma verdadeira metodologia (em termos de Engenharia de Software); algumas vezes, não existe nenhum "modelo"; além do mais, em geral, usuários não seguem nenhum conjunto definido de passos, mas alternam simulações com resultados de análise. A analogia mais adequada seria a de prototipagem rápida, que, no entanto, nunca é precedida por atividades de especificação ou análise de requisitos.

5.2 Concepção de Aplicações Geográficas

A metodologia proposta, retratada na Figura 5.1, é resultado de um estudo de dois anos com diferentes usuários de SIG na área de planejamento ambiental.

As fases genéricas previstas pela Engenharia de Software para a concepção e desenvolvimento de aplicações, foram adaptadas a aplicações geográficas sendo consideradas quatro etapas básicas: planejamento, inventário, desenvolvimento e avaliação. Esta adaptação visa a aproximar a realidade do usuário desenvolvedor dos conceitos de Engenharia de Software sem perda do formalismo desejado. Ao mesmo tempo, destaca a importância das atividades relacionadas ao tratamento e pré-processamento dos dados. Segundo [Aro89] o sucesso de uma aplicação é determinado pelos seguintes fatores: o conjunto de dados utilizado (estratégia metodológica), a forma de organização destes dados (inventário), o "modelo" utilizado (estratégia metodológica) e os critérios de avaliação dos resultados (avaliação). O tratamento das questões relacionadas com estes fatores está embutido nas quatro etapas básicas da metodologia proposta.

Ressaltamos alguns problemas de terminologia encontrados:

- o conceito de *análise* em Engenharia de Software (ou seja, análise dos requisitos do software) é diferente do conceito de análise em aplicações geográficas (que corresponde à execução de funções de análise espacial);
- o usuário utiliza o termo *modelo* para se referir a uma determinada estratégia metodológica, ou seja, um conjunto de métodos que serão utilizados e definirão o modelo de dados, o modelo dinâmico e o modelo de processo, termos técnicos da Engenharia

de Software; portanto, para o usuário, modelo e metodologia são termos utilizados indistintamente;

- o usuário desenvolvedor não tem perfeitamente definido os conceitos de *o que* desenvolver (visão lógica) e *como* desenvolver (visão física); nas quatro etapas mencionadas acima os usuários utilizam métodos que confundem as duas visões; a metodologia (e sua implantação no UAPÉ, como se verá no próximo capítulo) direciona o usuário a separar estas duas visões;
- a fase de manutenção corresponde, para o usuário, a uma seqüência de atividades que envolve monitoramento, simulações, correção de modelos, entre outras. Desta fase, a última etapa foi chamada de *avaliação* pois corresponde à avaliação constante não só do software como da realidade sendo estudada.

As fases são descritas através do detalhamento de suas atividades, sem, no entanto, mencionar os métodos e técnicas específicas utilizadas pelo usuário em cada uma delas. Uma descrição detalhada se encontra em [San94].

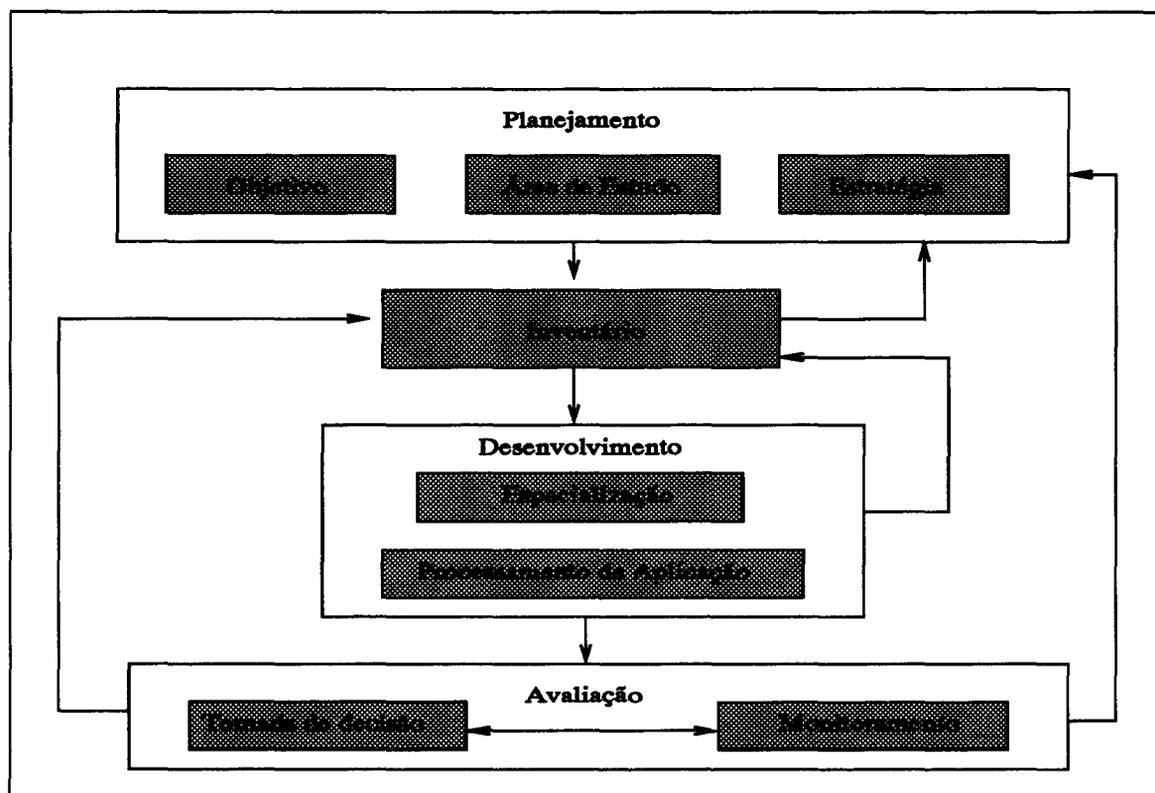


Figura 5.1: Metodologia para Aplicações Ambientais

1. Planejamento

O planejamento compreende as fases de planejamento propriamente dita e análise de requisitos do ciclo de vida do software. Ele envolve três atividades: definição dos objetivos, definição da área de estudo e definição da estratégia metodológica.

Os *objetivos* determinam os dados e processos considerados nas etapas subsequentes com a descrição dos problemas a serem resolvidos.

A definição da *área de estudo* consiste na determinação da extensão geográfica sobre a qual atuará a aplicação. A área de estudo é caracterizada sob os aspectos político, econômico, social e geográfico. No caso de aplicações de planejamento ambiental são identificadas as potencialidades, fragilidades, acertos e conflitos existentes na região.

A *estratégia metodológica* está baseada em métodos específicos do domínio da aplicação e a sua escolha está fortemente apoiada nos objetivos e na área de estudo. A estratégia direciona: 1) a determinação dos fatores ou variáveis físicas, biológicas e antrópicas a considerar; 2) a escolha da escala ou precisão dos dados a coletar; 3) a forma de classificação em ordem de importância dos fatores e como ponderá-los em função dos atributos de análise; 4) a forma de estudo das interações entre os vários fatores (modelagem de processo); 5) o intervalo de tempo de coleta dos dados.

Algumas metodologias de planejamento ambiental são estrategicamente encaminhadas para analisar os fatores de maneira que possam, em síntese final, identificar áreas homogêneas, definidas espacialmente, denominadas *unidades de paisagem*. Em outras palavras, uma metodologia de planejamento ambiental visa apresentar e relacionar os fatores do meio que compõem a área de estudo de forma espacializada, objetivando agrupar as características semelhantes dos fatores selecionados em unidades homogêneas, pressupondo-se que tenham uma auto-organização interna.

No caso de aplicações de planejamento ambiental, a identificação das unidades de paisagem é utilizada na definição de diretrizes para uma região: para cada unidade (ou conjunto de unidades semelhantes) é definido um conjunto de diretrizes, já que se pressupõe que cada unidade apresenta uma dinâmica particular.

Outras atividades típicas como avaliação de riscos, gerenciamento de projetos e análise de requisitos, são perfeitamente cabíveis em aplicações geográficas utilizando técnicas e métodos formais da Engenharia de Software.

2. Inventário

O inventário se incumbe da definição e coleta de dados para a aplicação, correspondendo em termos de Engenharia de Software, a uma análise mais detalhada dos dados envolvidos na aplicação. Ele consiste em três atividades básicas:

- *seleção* das variáveis ou fatores a serem analisados; os fatores são analisados em função da estratégia metodológica adotada, da área de estudo e da fonte dos dados, levando em consideração sua qualidade e forma de obtenção.

- *definição* da escala, precisão e nível de detalhe.

Para essas duas atividades, as técnicas e métodos de Engenharia de Software, para modelagem de dados, podem ser empregados, embora não se consiga abstrair toda a semântica das entidades geográficas, relacionamentos e restrições. Conforme visto no capítulo anterior, o GMOD estende a modelagem convencional para justamente suprir estas deficiências.

- *coleta* de dados através da utilização de mapas topográficos, interpretação de fotografias aéreas, trabalhos de campo ou teledetecção (satélites ou aviões).

Em aplicações convencionais a coleta de dados não é tratada de maneira especial, com a utilização de técnicas e métodos formais específicos. Em aplicações geográficas, as atividades relacionadas com a coleta, integração e organização dos dados merecem um tratamento diferenciado pelas características espaço-temporais dos dados, incluindo outros aspectos inerentes como escala, precisão e ponderação.

Dependendo da qualidade dos dados coletados, pode-se voltar à etapa anterior para revisão dos tipos de variáveis a serem utilizadas e respectivas ponderações.

Nesta fase, as diferentes fontes de dados são homogeneizadas em termos de escala, tipo de dados e incompatibilidades temporais (exemplo, datas de coleta). Além disso, se decide quais fatores serão espacializados (geo-classes) e quais serão tratados como dados convencionais durante a fase de modelagem e integração. O usuário considera a forma de organização dos dados pelo SIG, as características de armazenamento e recursos de análise disponíveis.

Para aplicações ambientais, [San94] sugere técnicas e métodos formais para as atividades de seleção, definição e coleta de dados, mencionadas acima.

3. Desenvolvimento

O desenvolvimento corresponde às fases de projeto, codificação e testes da Engenharia de Software. Dependendo do que seja esta aplicação, o processamento pode corresponder à execução de um conjunto de consultas (e obtenção de mapas) até a construção de código complexo. O processo de desenvolvimento é direcionado para as seguintes etapas:

- definição e geração dos dados - estabelecimento de todos os arquivos/bancos de dados a serem usados

- especificação do modelo de processo (a partir da estratégia metodológica da fase 1)
- implementação do modelo de processo em um SIG
- validação - em geral a partir da análise visual dos resultados por especialistas no domínio

A fase de desenvolvimento compreende as seguintes atividades:

- *Espacialização dos dados:*

Esta operação corresponde a geo-referenciar os dados e armazená-los no banco de dados do SIG subjacente ou confeccionar os mapas manualmente, após conversão de formatos quando necessário. Este procedimento, geralmente, introduz novos erros no conjunto de dados.

- *Processamento da Aplicação*

Esta atividade corresponde, na prática, a uma série de passos estreitamente dependentes da estratégia metodológica adotada na fase inicial. Algumas destas atividades são:

- transformação dos elementos definidos nas etapas anteriores para o modelo de dados do SIG subjacente; isto exige, do ponto de vista de banco de dados, a definição de arquivos, relações, classes de dados, especificação de formato e definição e implementação de operações. Do ponto de vista da aplicação exige também definição de fatores como parâmetros de visualização e visões de usuários;
- derivação de Planos de Informação (PI) com: atribuição de pesos a cada um dos PI conforme sua importância na integração das informações, e atribuição de pesos a cada um dos elementos determinado no PI (ou seja, a cada uma das categorias de classificação destes elementos);
- simulações diversas do modelo adotado com verificação dos resultados em campo; as simulações correspondem a variações nos pesos atribuídos, reclassificações nos PI e outras medidas [San94].

Esta etapa de processamento é extremamente dependente do modelo adotado na Definição da Estratégia Metodológica e é executada com mais precisão e flexibilidade quando se dispõe de um SIG para a execução de diversas funções de análise espacial e apresentação de resultados. Em muitos casos, dependendo do volume dos dados coletados, a atividade de teste é realizada através de um projeto chamado "projeto piloto". Este projeto corresponde à implementação de um subconjunto das funções previstas visando avaliar, dentre

outros, a qualidade dos dados armazenados e a validade do modelo especificado. Os testes, neste caso, servem para evidenciar não apenas erros de implementação de algoritmos, mas principalmente erros na coleta de dados e na concepção/especificação da aplicação.

4. Avaliação

A avaliação corresponde à fase de manutenção da Engenharia de Software e inclui a análise, pelo especialista, dos resultados do desenvolvimento. As etapas da fase de avaliação são:

- *Tomada de decisão e especificação de políticas*

Os usuários estudam os diferentes resultados obtidos pela aplicação e escolhem um ou mais deles para indicar possíveis soluções. Estas soluções são traduzidas em sugestões e diretivas a serem tomadas. A especificação de políticas é adotada, em geral, em caso de planejamento e controle ambiental.

- *Monitoramento*

A partir de agora, os especialistas acompanham a implementação das diretivas e os resultados são comparados com os previstos pela aplicação.

5.3 Exemplo de uso da metodologia

Para ilustrar a metodologia, vamos aplicá-la no desenvolvimento de uma aplicação geográfica de análise de riscos de incêndios florestais utilizada como estudo de caso no nosso trabalho [FV96].

1. Planejamento

O *objetivo* da aplicação é avaliar uma estratégia metodológica de análise de riscos de incêndios florestais, englobando vários fatores ambientais (naturais e antrópicos) e utilizando os recursos de um SIG.

A *área de estudo* é a Estação Experimental de Tupi pertencente ao Instituto Florestal do Estado de São Paulo, localizada no município de Piracicaba, com coordenadas geográficas aproximadas $22^{\circ}44'15''$ e $47^{\circ}32'$.

A *estratégia* utilizada nesta aplicação visa o mapeamento de riscos de incêndio da região para identificar os locais com maiores ou menores riscos de incêndio. A estratégia justifica os fatores a serem considerados, a escala, a ordem de importância e o modelo de processo.



Várias são as estratégias metodológicas utilizadas em *mapeamento de riscos de incêndio* tais como: estatísticas de ocorrências de incêndios; correlações com informações como o número de habitantes e índices econômicos da população; e combinação de vários dados demográficos e climáticos com classificação de regiões através de regressões múltiplas.

2. Inventário

As *variáveis* selecionadas para a aplicação foram: principais acessos a E.E.Tupi (estradas e carreadores), cursos d'água e represas, ocupação do solo, relevo e ocupação das áreas vizinhas.

Os *dados* para esta aplicação foram obtidos através de uma carta planialtimétrica 1:10000 e fotografias aéreas verticais pancromáticas do local.

3. Desenvolvimento

Para esta aplicação foi utilizado o SIG IDRISI, onde foram digitalizados cinco mapas (*dados fonte*) na escala 1:10000 a partir da carta e fotografias aéreas mencionadas acima:

- principais acessos a E.E.Tupi, com a seguinte classificação: rodovia, estradas, carreadores e linha de força
- cursos d'água e represas, com a seguinte classificação: represas e cursos d'água
- ocupação do solo com a seguinte classificação: talhões, serviço, mata, capoeira/bosque e represas
- relevo com as curvas de nível com equidistância vertical de 5 metros; com isso foi possível construir o MDT (Modelo Digital de Terreno) da área. Utilizando MDT foi possível o estudo de duas variáveis importantes para a propagação de incêndios florestais: declividade do terreno e aspecto. Foram gerados a seguir o mapa de classes de declividade da área e o mapa das principais faces de exposição do terreno
- ocupação das áreas vizinhas com a seguinte classificação: horto, cana, mata, capoeira, pasto, colônia, área urbana, estrada

O *modelo de processo* foi especificado de forma a poder ser executado no SIG subjacente. Na aplicação, analisando-se as condições específicas da área, considerou-se a importância de cinco variáveis principais para o estudo de risco de incêndios florestais na E.E.Tupi:

- estradas e carreadores - gerada a partir do mapa de acessos, estradas e carreadores
- vizinhança - gerada a partir do mapa de ocupação das áreas vizinhas
- uso da terra - gerada a partir do mapa de ocupação do solo e cursos d'água e represas
- declividade - gerada a partir do mapa de relevo
- exposição - gerada a partir do mapa de relevo

Cada uma das variáveis mencionadas acima foi introduzida no estudo a partir de Planos de Informação (PI). Para cada PI adotou-se um peso específico. Por exemplo, o peso de menor valor (1,0) foi atribuído ao plano "Exposição", enquanto que o plano "Vizinhança" recebeu o maior peso (1,6).

Dentro de cada PI, as informações correspondentes foram ainda classificadas, recebendo pesos variando de 1 a 20, de acordo com sua maior ou menor influência para o risco de incêndio.

Após a obtenção dos cinco PI, estes foram integrados em um mapa síntese, através de uma operação de adição realizada com o auxílio do SIG. Desse modo, cada ponto do mapa resultante representou a soma dos valores do mesmo ponto em cada PI, agrupando todas as informações com seus respectivos pesos. Como resultado, foi obtido um mapa que foi decomposto em regiões de acordo com o risco de incêndio.

Ressalte-se que este tipo de especificação "algorítmica" (da forma, "dado/mapa A gerado a partir da aplicação da função X ao dado/mapa B") é a forma mais comum de especificação de aplicação em SIG no domínio ambiental. Não só os usuários desconhecem programação e especificação de programas, mas o próprio SIG induz tal tipo de especificação. Um programa é, na verdade, uma seqüência de processamento de funções do SIG sobre dados geo-referenciados, podendo, no máximo, ser definido em uma linguagem macro. Já aplicações urbanas, em especial as que exigem roteamento ou alocação de recursos, costumam exigir mais sofisticação de programação (e maior especialização no domínio).

4. Avaliação

As medidas tomadas foram as de planejamento e prevenção nas regiões com maiores risco de incêndio em uma área de floresta, nativa ou implantada.

A partir de então, os especialistas acompanham a implementação das diretivas e os resultados são comparados com os previstos na aplicação.

5.4 Resumo

Este capítulo apresentou a metodologia de desenvolvimento de aplicações geográficas proposta neste trabalho. O contexto atual de desenvolvimento destas aplicações foi comparado ao estágio atual de desenvolvimento de aplicações convencionais apoiado em técnicas de Engenharia de Software. A metodologia foi descrita através do detalhamento das suas quatro fases: planejamento, inventário, desenvolvimento e avaliação. Por último, como exemplo, a metodologia foi aplicada no desenvolvimento de uma aplicação geográfica real.

A metodologia foi influenciada pela forma de trabalho do usuário e pelo perfil das ferramentas computacionais disponíveis em SIG, que induzem o estilo "programação por menu", sendo os algoritmos embutidos no modelo de processos. Neste domínio, a principal dificuldade reside em determinar os dados adequados (modelagem de dados) e em atribuir-lhes os pesos convenientes (modelagem de processos) para combiná-los adequadamente.

Este entendimento dos procedimentos e padrões utilizados pelo usuário é importante como embasamento para a proposta de construção de um ambiente de modelagem e projeto de aplicações. O próximo capítulo descreve o UAPÉ, cujo objetivo é exatamente este.

Capítulo 6

Ambiente UAPÉ

Este capítulo descreve a arquitetura básica do ambiente computacional UAPÉ, uma ferramenta do tipo CASE para auxiliar o usuário na análise e projeto de aplicações geográficas. Algumas das tarefas que qualquer ferramenta CASE precisa suprir (pelo menos, parcialmente) são: garantir que os modelos de dados, dinâmico e de processos estejam corretos de acordo com a semântica da metodologia embutida, manter as suas especificações textuais, documentar os modelos e fornecer recursos gráficos de modelagem.

A seção 6.1 introduz uma visão geral do ambiente, salientando as principais decisões para sua concepção. A seção 6.2 mostra a metodologia descrita no capítulo anterior adaptada para ser utilizada sob o UAPÉ. A seção 6.3 descreve os módulos do ambiente que dão suporte às diversas atividades previstas na metodologia.

6.1 Visão Geral

O UAPÉ deve ser entendido como uma camada de software a ser acoplada a algum SIG. Esta camada de software utiliza um SGBD orientado a objetos para gerenciar os dados do usuário e os dados necessários ao próprio ambiente.

O UAPÉ permite que o usuário desenvolvedor especifique uma aplicação geográfica. Para a especificação, pode-se reutilizar dados já existentes importados do SIG subjacente ou modelar novas entidades geográficas e seus relacionamentos. O UAPÉ está calcado em três princípios:

- *modelo de dados* - o usuário modela os fenômenos geográficos segundo o modelo orientado a objetos GMOD (Capítulo 4);
- *metodologia de especificação* - o usuário é auxiliado no desenvolvimento da aplicação a partir de uma metodologia desenvolvida para esse fim (Capítulo 5);

- *independência de SIG* - o ambiente pode ser utilizado por diversos SIG graças a duas características: sua implementação em um SGBDOO (Sistema de Gerenciamento de Banco de Dados Orientado a Objetos) e uso de conceitos adotados por alguns SIG comerciais.

A Figura 6.1 mostra uma visão em alto nível do UAPÉ do ponto de vista de acoplamento a SIG.

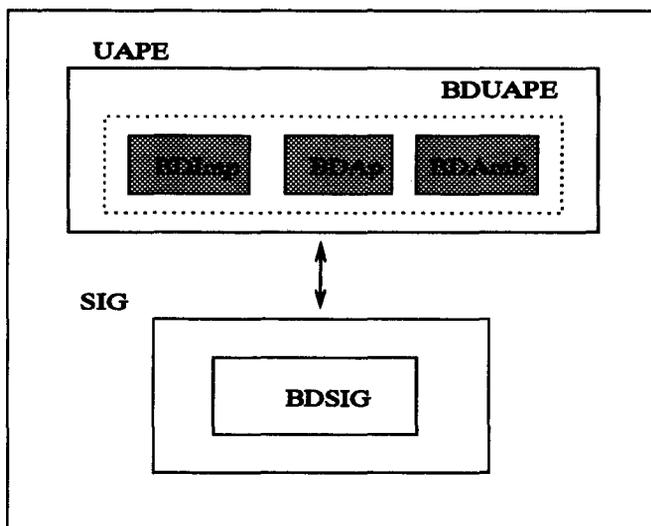


Figura 6.1: Arquitetura de Banco de Dados do UAPÉ

O projeto de uma nova aplicação é realizado sobre o banco de dados do UAPÉ (BDUAPE) que é baseado no GMOD. Este banco de dados pode ser visto como composto conceitualmente por três blocos:

- **BDImp**: dados já disponíveis no banco de dados do SIG subjacente e que são importados para o UAPÉ para reutilização eventual pela nova aplicação.
- **BDAP**: novos dados definidos pela aplicação sendo projetada, usando o GMOD, que posteriormente serão exportados para o SIG.
- **BDAMB**: dados e classes específicos ao UAPÉ, acrescentados ao GMOD para gerenciamento de aplicações existentes e documentação da nova aplicação.
- **BDSIG**: é o banco de dados geográfico do SIG subjacente. Como já foi mencionado, nem todo SIG utiliza um SGBD. No entanto, todo SIG utiliza algum conjunto de arquivos gerenciado por algum sistema operacional (por exemplo, Unix). Entende-se nesta tese que BDSIG é na verdade o conjunto de dados do SIG.

O BDUAPE foi desenvolvido utilizando o sistema O_2 . A exportação e importação do SIG subjacente exigem desenvolvimento de módulos de conversão de esquema e dados.

Esta arquitetura de comunicação parte de duas premissas:

- Se o banco de dados do SIG subjacente não suportar as facilidades do UAPÉ (por exemplo, não permitir uso de gatilhos) várias das propriedades das aplicações modeladas não poderão ser implementadas, sendo o usuário advertido deste problema.
- É permitido ao usuário realizar a carga dos novos dados diretamente sobre o SIG subjacente. Neste caso, o UAPÉ apenas é utilizado como ferramenta de projeto para o SGBD subjacente, sendo que apenas o esquema de BDAP é exportado. Este esquema é informado ao usuário que assim tem conhecimento das modificações no SGBD do SIG introduzidas pela nova aplicação.

O BDAMB é um repositório com a descrição das classes que serão instanciadas pelo usuário e outros dados necessários para dar suporte ao usuário na reutilização de classes de outros projetos como documentação e relacionamentos. O BDAMB gerencia, através de classes acrescentadas no GMOD, as aplicações que são projetadas a partir do UAPÉ.

O BDAMB utiliza o GMOD acrescentando-lhe três novas classes: Projeto, Área de Estudo e Documento. Estas classes definem o contexto de uma aplicação geográfica caracterizado por dados inerentes ao projeto da aplicação que serão convertidos posteriormente para a linguagem do SIG subjacente. A descrição das classes, seus atributos e métodos é dada utilizando a linguagem de definição de dados do sistema O_2 . A Figura 6.2 mostra o acoplamento destas classes ao GMOD, onde a caixa pontilhada limita o GMOD.

- **PROJETO:** a descrição do problema e objetivos são registrados em uma instância da classe Projeto.

```
class Projeto
    type tuple (projid:    string,
               descricao: string,
               dataref:   Date,
               estrategia: string)
end;
```

- **ÁREA DE ESTUDO:** é a extensão espacial sobre a qual a aplicação geográfica atuará. Um projeto está associado a uma única área de estudo, partindo-se do pressuposto que um projeto é caracterizado essencialmente pelos seus objetivos aplicados a uma determinada região. Mais especificamente, a definição da área de estudo do projeto está em um objeto da classe Área de Estudo associado a um objeto da classe Geo-Região.

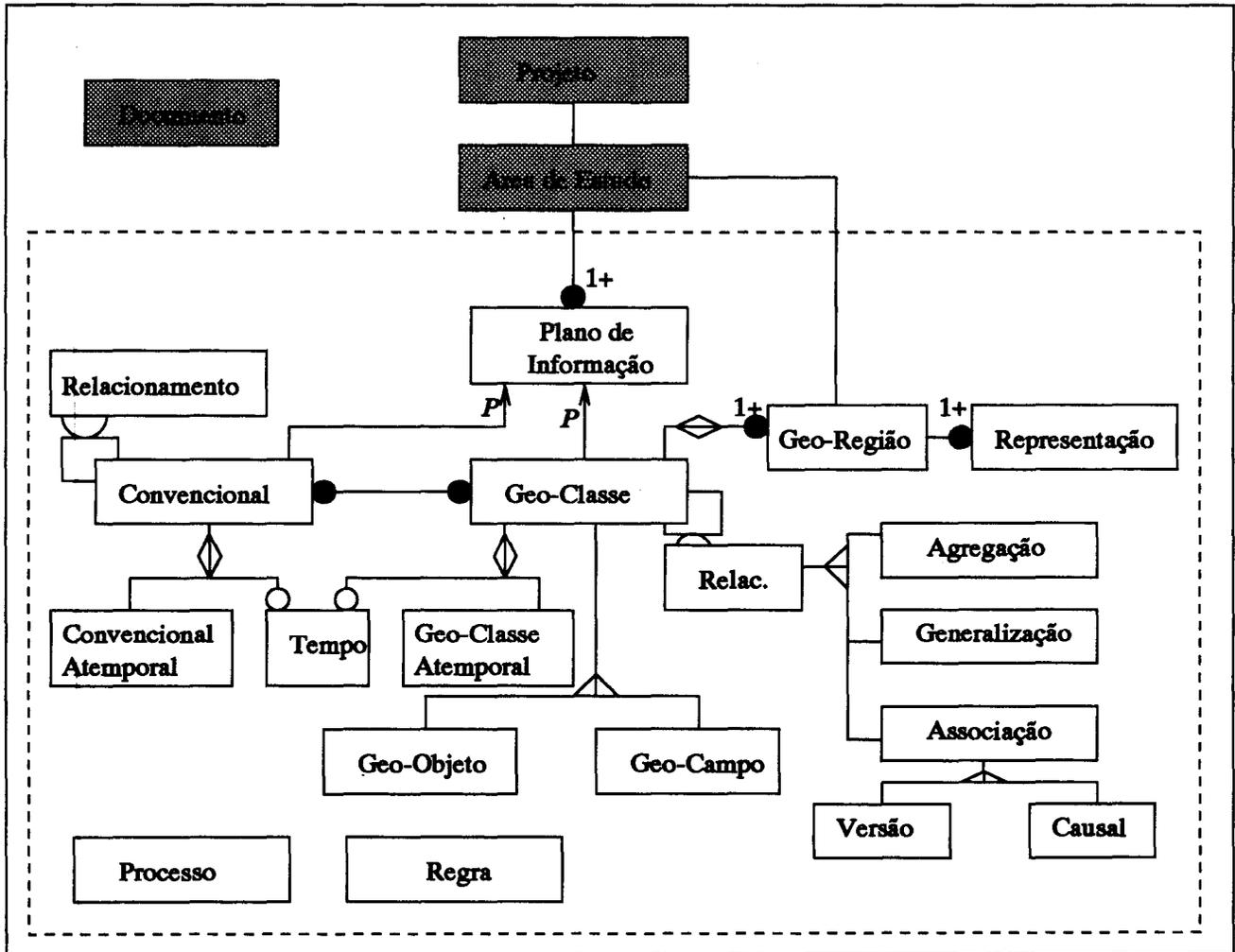


Figura 6.2: GMOD com as Classes de Suporte ao Projeto

```

class Area-Estudo
    type tuple (projeto: Projeto,
               descricao: string,
               dataref: Date,
               regioao: Geo-Regiao)
end;

```

- **DOCUMENTO:** o usuário pode registrar as decisões tomadas ao longo do projeto instanciando a classe Documento. Esta classe pode ser especializada em classes que abstraem os diversos tipos de documentos (e mídias) que podem ser utilizados para registrar diferentes tipos de informação como: decisões de projeto, prazos, métodos empregados, fluxo de operações executadas, entre outros. Estes documentos podem ser utilizados, dentre outras coisas, como metadados sobre os dados armazenados. Um exemplo comum de uso de metadados é a definição do que os usuários chamam de "linhagem" dos dados, como por exemplo, o tipo de dispositivo que o produziu (sensoriamento remoto, foto) e algoritmos e procedimentos de processamento digital do dado (filtros aplicados, procedimentos de geo-referenciamento). Meta-dados são fundamentais em bibliotecas digitais geográficas [GM97]

```

class Documento
    type tuple (projeto: Projeto,
               conteudo: string,
               autor: string,
               dataref: Date)
end;

```

A adoção do modelo orientado a objetos do sistema O_2 como modelo intermediário entre o GMOD e o SIG subjacente permite obter um *modelo genérico* [CCH⁺96] para o qual possam ser mapeados e integrados dados de diversos bancos de dados geográficos. Por outro lado, o usuário tem acesso a informações de diversas fontes sem precisar conhecer a linguagem de manipulação dos vários sistemas de armazenamento. Na abordagem adotada pelo UAPÉ, através do GMOD, o usuário abstrai a realidade geográfica de interesse sem preocupar-se com os detalhes de modelos formais como o O_2 , ou com os detalhes de implementação de diversos sistemas proprietários.

6.2 Concepção de Aplicações Geográficas sob o UAPÉ

A metodologia descrita no capítulo anterior foi adaptada para ser utilizada sob o UAPÉ com as seguintes premissas:

- deve suportar tanto a modelagem estática como a dinâmica e a de processos;
- deve levar em consideração o fato de que a modelagem e simulação do mundo real é uma atividade interativa. Assim deve suportar um ciclo de vida de desenvolvimento incremental, onde usuários alternam entre atividades de projeto/teste/validação;
- deve suportar os padrões de trabalho do usuário e ao mesmo tempo induzi-lo a seguir procedimentos de Engenharia de Software;
- deve ajudar o usuário a documentar suas atividades e levar em consideração o fato que muitas delas não podem ser automatizadas.

As principais adaptações feitas são:

- os *objetivos* são registrados como atributo da classe Projeto;
- a *área de estudo* constitui uma instanciação da classe Area de Estudo;
- a *estratégia metodológica* direciona toda a concepção da aplicação e determina como será a instanciação do modelo GMOD pelo usuário. A estratégia adotada é documentada no GMOD através de instanciação da classe Documento, complementada pela especificação das restrições dos relacionamentos do modelo;
- o *inventário* é tipicamente uma atividade manual. O UAPÉ auxilia o usuário na exploração dos dados já cadastrados no ambiente que foram importados dos SIG subjacentes (BDImp). Isto propicia a reutilização dos dados, facilitando o inventário. Além disto, o usuário pode consultar a documentação já cadastrada no ambiente para avaliar estratégias adotadas em projetos correlatos, respeitando os esquemas de segurança e manutenção de integridade;
- a *especialização* dos dados corresponde a associar ao dado o seu geo-referenciamento. O usuário não irá executar esta atividade no UAPÉ, mas somente indicar o tipo de representação que será utilizada para o armazenamento no SIG das entidades modeladas;
- os resultados da *tomada de decisão*, podem ser documentados no UAPÉ, como objetos da classe Documento. Neste caso, podem ser retomados sempre que necessário;
- a atividade de *monitoramento* não é suportada pelo UAPÉ. O monitoramento corresponde a uma reavaliação sistemática da solução adotada e sua implementação, em função do observado no mundo real, num confronto permanente com os objetivos da aplicação. Do ponto de vista do UAPÉ, isto pode resultar em atualizações nos modelos estático, dinâmico e de processos;

- incorporação de alguns métodos e técnicas empregados na metodologia OMT [RBea91].

Na metodologia OMT, a fase de análise compreende a construção dos modelos de objetos, dinâmico e funcional (ou de processos). Na fase de projeto elabora-se o projeto de sistema e o projeto de classes de objetos. É dada pouca ênfase à fase de implementação que se preocupa com a codificação e criação do banco de dados. O foco da metodologia proposta é o projeto de classes que integra os três modelos construídos na fase de análise, projeta algoritmos para a implementação de métodos, projeta os relacionamentos, ajustando a estrutura de classes para a otimização de acesso.

Na metodologia sob o UAPÉ, o usuário especifica os três modelos de análise do OMT, através da instanciação das classes do GMOD, e o ambiente gera automaticamente o projeto de classes da aplicação na linguagem do sistema O_2 ou exporta a especificação para o SIG subjacente, na linguagem apropriada. O UAPÉ não dá suporte ao projeto de sistema, visto que a manipulação das entidades geográficas da aplicação será feita pelo SIG subjacente.

Portanto, a partir da especificação de uma aplicação geográfica construída sob o UAPÉ, é permitido gerar um esquema de classes em O_2 com suporte a versões alternativas e temporais, múltiplas representações espaciais, especificação de regras para serem utilizadas em bancos de dados geográficos ativos e especificação de processos para serem gerenciados por exemplo, por mecanismos de workflow incorporados a SIG.

A seguir descrevemos as fases da metodologia (Figura 6.3) sem os detalhes dos métodos e técnicas orientadas a objetos utilizados, já que estes são descritos em [RBea91]. Estes métodos são aplicados de forma transparente para o usuário desenvolvedor. O Capítulo 7 traz um exemplo simples de uso do ambiente e ilustra a execução das diversas fases.

1. Definição do Projeto

Inicialmente, o usuário deve definir o projeto dentro do ambiente UAPÉ. Para isso é criada nova instância da classe Projeto e outra da classe Area de Estudo. Na operação de instanciação, o usuário fornece os valores dos atributos dos respectivos objetos. O usuário pode documentar a estratégia metodológica adotada instanciando a classe Documento. Esta fase objetiva inicializar a aplicação dentro do UAPÉ e corresponde à parte da etapa de Planejamento da metodologia de desenvolvimento do capítulo 5.

2. Modelagem

Esta fase visa construir os três modelos que abstraem os principais aspectos de um sistema: modelo estático (ou de objetos), modelo dinâmico e o modelo de processos e compreende parte da fase de Inventário e parte da fase de desenvolvimento (projeto) da metodologia do Capítulo 5.

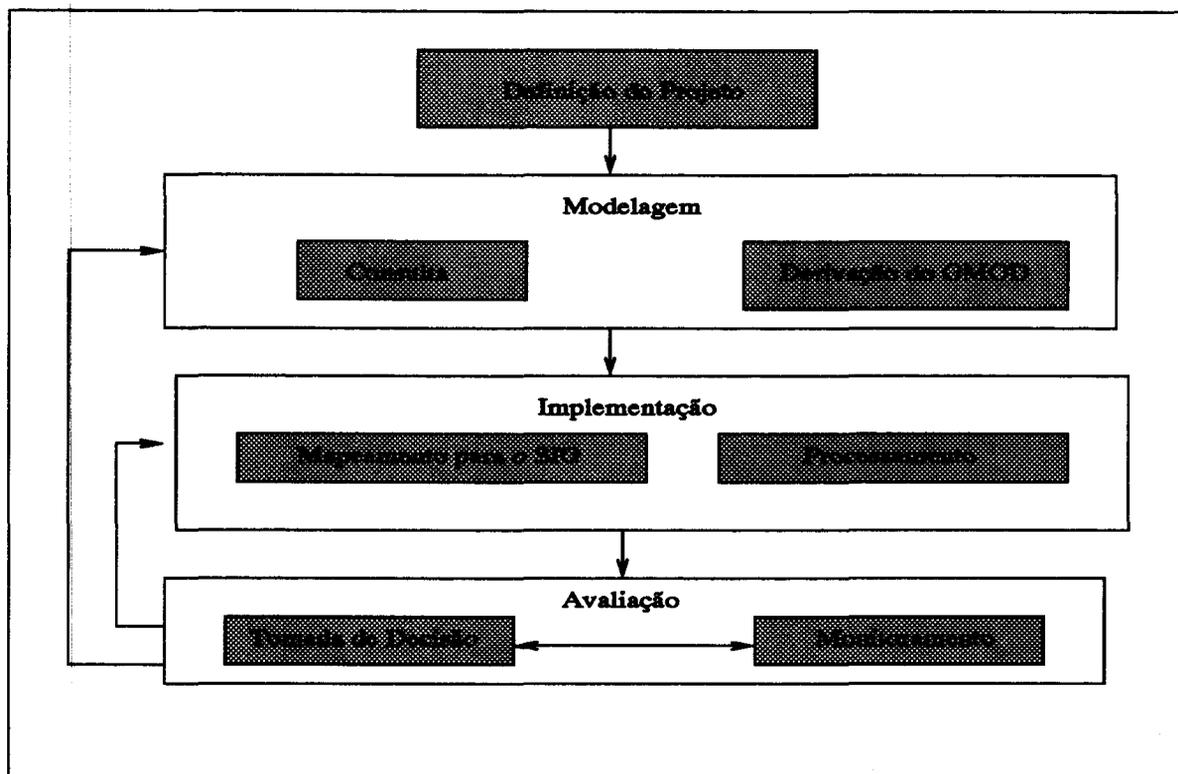


Figura 6.3: Metodologia para Aplicações Ambientais sob o UAPÉ

O primeiro passo é a análise dos requisitos da aplicação para a construção do modelo de objetos. O modelo de objetos precede o modelo dinâmico e o modelo de processos porque a estrutura de dados é normalmente melhor definida, menos dependente dos detalhes da aplicação e mais estável à medida que o sistema se desenvolve. As funções do modelo de processos podem ser expressas em linguagem natural ou equações matemáticas.

Os três modelos não têm a mesma importância em todos os tipos de aplicação. Pelo contrário, em algumas delas, como zoneamento, o modelo dinâmico e de processos podem ser dispensáveis, enquanto que em aplicações de monitoramento ambiental, estes modelos são imprescindíveis.

A especificação do modelo dinâmico é importante se o SIG subjacente suporta o gerenciamento de regras, ou seja, estiver suportado por um banco de dados ativo.

De forma resumida, o usuário constrói o modelo de objetos da sua aplicação identificando classes, seus atributos e relacionamentos, a partir dos objetivos do projeto, da área de estudo e da estratégia metodológica escolhida. As necessidades de versionamento e temporalização são identificadas neste momento. Nesta fase o usuário alterna pesquisa ao BDImp e especificação de novas classes no BDAP. A pesquisa permite ao usuário verificar quais classes de interesse estão disponíveis e neste caso

seleccioná-las para o BDAP.

Para a construção do modelo dinâmico o usuário identifica classes onde as mudanças de estado dos seus objetos, no tempo, são de interesse da aplicação. Outras mudanças de interesse são aquelas que afetam o estado de objetos de outras classes. As classes envolvidas, neste caso, participam de um relacionamento causal identificado no modelo de objetos. Os eventos, as condições que devem ser satisfeitas para que o evento seja considerado e as ações que devem ser acionadas nestas condições compõem as regras que fazem parte do modelo dinâmico da aplicação. Uma ação acionada por um evento pode envolver uma seqüência de processos a serem executados. Estes processos serão modelados durante a modelagem de processos.

A geração de objetos da classe Regra a partir da especificação dos componentes do modelo dinâmico é mostrada em [Cil96].

O ponto de partida para a identificação dos processos da aplicação é, segundo a metodologia OMT, as ações do diagrama de estado das classes. Para a modelagem o usuário deve identificar os processos ou funções envolvidos na aplicação (sem se preocupar com o nível de complexidade e abstração) e descrevê-los fornecendo:

- o nome do processo,
- os objetos ou valores de atributos de entrada e saída (parâmetros) e a
- descrição do processo (o que ele faz) em linguagem natural, equação matemática, pseudo-código ou outra forma apropriada.
- restrições de processamento

Estas informações serão utilizadas na instanciação da classe Processo.

O usuário identifica restrições entre classes de objetos ou entre objetos da mesma classe, incluindo restrições temporais. Outras restrições são aquelas sobre os parâmetros de entrada e saída. Estas restrições são registradas em instâncias da classe Regra do GMOD.

O estágio atual de implementação do UAPÉ não dá suporte à construção dos diagramas dos modelos dinâmico (diagrama de estados) e de processos (diagrama de fluxo de dados), mas permite que o usuário descreva textualmente os seus componentes, como descrito no Capítulo 4.

3. Implementação

Terminada a modelagem da aplicação, estão disponíveis no BDUAPE as informações necessárias para o mapeamento da especificação da aplicação para a linguagem do

SIG subjacente através de *drivers* específicos. Em seguida, inicia-se o desenvolvimento da aplicação no que diz respeito a espacialização dos dados e processamento da aplicação (ambos fora do UAPÉ).

4. Avaliação

A tomada de decisões e os resultados do monitoramento podem ser documentados no UAPÉ associando-se instâncias da classe Documento à instância correspondente da classe Projeto.

A Figura 6.4 relaciona as fases clássicas da Engenharia de Software com as fases da Metodologia proposta no Capítulo 5 e a adaptação desta sob o UAPÉ, descrita acima.

Engenharia de Software	Análise Sistemática	Planejamento	Análise Requisitos	Projeto	Implement.	Teste	Manutenção
	Definição (o que)			Desenvolvimento (como)			Manutenção
Metodologia do Usuário	Planejamento			Inventário	Desenvolvimento		Avaliação
	Definição do Projeto		Modelagem		Implementação	Avaliação	
Metodologia do UAPÉ	Definição do Projeto		Modelagem		Implementação		Avaliação

Figura 6.4: Relacionamento entre as Fases das Metodologias

6.3 Arquitetura Conceitual do Ambiente

Do ponto de vista do usuário, o UAPÉ pode ser visto como composto de um conjunto de módulos integrados como apresentado na Figura 6.5.

Os módulos dão suporte à execução das fases da metodologia descrita na seção anterior. A coordenação da execução das fases é ditada pelo Módulo da Metodologia. A Interface permite que o usuário tenha uma visão uniforme do ambiente interagindo com todos os outros módulos. Os acessos e manipulações dos repositórios do ambiente são gerenciados pelo Módulo de Consulta. O Módulo de Modelagem e Projeto suporta a

fase de Modelagem da metodologia permitindo que o usuário especifique os três modelos: de objetos, dinâmico e de processos. Os Módulos de Importação e Exportação são responsáveis por alimentar o BDImp e migrar o conteúdo do BDAP para o SIG subjacente, respectivamente.

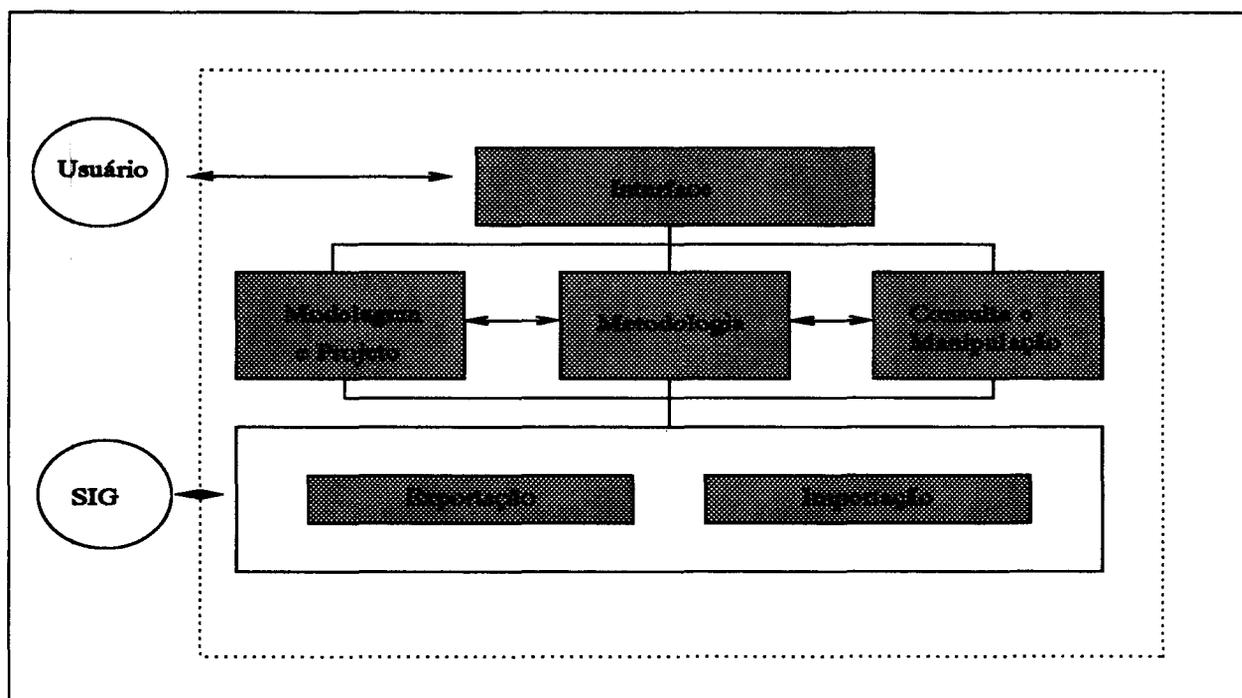


Figura 6.5: Arquitetura do UAPÉ

A interação do usuário com o UAPÉ tem tipicamente o seguinte perfil. O usuário inicia uma sessão para realizar o projeto de uma aplicação. Toda interação se faz pelo Módulo de Interface. Inicialmente, o usuário determina que classes de dados deseja reutilizar do banco de dados subjacente. Para isto, aciona o Módulo de Consulta e Manipulação que, por sua vez, aciona o Módulo de Importação, que traz para o BDImp o esquema do banco de dados subjacente e seleciona as extensões de classe que deseja importar para o BDImp.

Uma vez realizada esta tarefa, é iniciado o processo de modelagem, com interação Interface/Módulo de Modelagem e Projeto. Terminada a modelagem, as novas classes são exportadas para o SIG e sua definição é armazenada no BDAMB.

Finalmente, o usuário pode carregar os dados diretamente no SIG.

A interação com os Módulos de Consulta e de Modelagem é mediada pelo Módulo de Metodologia, que dirige o usuário interativamente a especificar suas classes e processos segundo a metodologia do Capítulo 5.

No estágio atual do UAPÉ, os módulos de Importação e Exportação não foram implementados e todo o ambiente roda em cima do BDUAPE, implementado no O_2 .

A seguir, cada um destes módulos é descrito de forma mais detalhada.

6.3.1 Interface do Usuário

O UAPÉ tem uma interface de manipulação direta. O principal objetivo da interface é oferecer as facilidades de modelagem e projeto da aplicação a partir da metodologia embutida. Estas facilidades incluem um editor gráfico para o modelo da aplicação e mecanismos de exploração (*browsing*). Através da Interface o usuário pode visualizar, criar e modificar o modelo da sua aplicação. O Capítulo 7 contém cópias de telas mostrando as características gráficas da interface implementada no protótipo. A arquitetura desta interface foi baseada em tese de doutorado em desenvolvimento no IC-UNICAMP [Oli97]. A parte relativa ao projeto de aplicação utiliza um módulo desenvolvido como parte de iniciação científica. A interação com os diversos bancos de dados armazenados no O_2 utiliza uma nova versão do sistema de interface GOODIES [OA93].

6.3.2 Módulo de Modelagem e Projeto

O Módulo de Modelagem e Projeto é responsável por permitir ao usuário efetuar modelagem estática, dinâmica e de processos. No GMOD o usuário modela a aplicação geográfica sem se preocupar com detalhes de implementação. Sob este aspecto, apenas indica diretrizes que serão utilizadas pelo ambiente UAPÉ em tempo de geração da especificação do banco de dados geográfico correspondente. A utilização do GMOD como base de comunicação com o usuário provê uma ferramenta para a especificação correta de operações e esquemas de dados.

O funcionamento do Módulo de Modelagem e Projeto está baseado nas seguintes premissas:

- o conjunto de dados que o usuário tem disponível inicialmente é consistente em termos de extensão espacial, escala e precisão. Esta consistência depende da seleção do usuário, que solicita dados para preencher o BDImp;
- as operações permitidas não consideram as possíveis restrições do SIG subjacente (por exemplo, sobre versões); estas devem ser tratadas pelos Módulos de Importação e Exportação;
- nenhuma operação de projeto pode alterar a definição das classes do BDImp, importadas do SIG subjacente e que devem permanecer íntegras;

- durante a modelagem o usuário deve garantir a consistência do BDAp na execução de todas as operações permitidas.

As operações básicas permitidas no Módulo de Modelagem e Projeto são as seguintes:

- *criação de classes* a partir de especialização das classes do GMOD para as modelagens de dados, dinâmica e de processos. O ambiente conduz o usuário na especificação de processos a partir de relacionamentos causais e instanciação de Planos de Informação.
- *criação de Planos de Informação*: corresponde à instanciação da classe Plano de Informação. Na definição de um PI o usuário especifica os critérios a serem utilizados para a determinação dos geo-objetos e geo-campos que comporão o PI. Estes já devem ter sido previamente definidos no BDAp.
- *versionamento de classes do BDAp*: o versionamento de uma entidade geográfica é indicado através de um auto-relacionamento. Este auto-relacionamento significa que o ambiente deve associar à classe mecanismos que permitam o armazenamento de versões.
- *temporalização de classes do BDAp*: a temporalização de uma entidade geográfica é abstraída através da utilização da classe Tempo do modelo GMOD. No UAPÉ o usuário indica que a classe modelada deve ser temporalizada associando a classe em questão à classe Tempo e escolhendo uma das formas de representação suportada pelo GMOD.
- *criação de relacionamentos*: sob o UAPÉ o usuário pode criar explicitamente os seguintes tipos de relacionamentos, conforme definidos no capítulo 4: associação (causal e versionamento), generalização/especialização, e agregação. Os relacionamentos entre versões de entidades geográficas e os relacionamentos temporais são estabelecidos implicitamente pelas operações de temporalização e versionamento, descritas acima. Os relacionamentos espaciais são estabelecidos e tratados pelo SIG subjacente através dos módulos de entrada, tratamento/análise e apresentação dos dados.

A documentação da aplicação é efetivada através do estabelecimento de relacionamentos de classes do modelo com a classe Documento.

Restrições de integridade e modelagem dinâmica em geral são implementadas através da instanciação da classe Regra.

6.3.3 Módulo de Consulta e Manipulação

O Módulo de Consulta é utilizado para recuperar e manipular dados geográficos e esquemas expressos em GMOD. Supõe-se que estes dados foram originalmente especificados através do Módulo de Modelagem e Projeto. O módulo de consulta realiza operações sobre o BDUAPE.

A consulta ao BDImp permite a reutilização de dados e meta-dados existentes no SIG subjacente. São disponibilizados dois tipos básicos de consultas:

- *consultas espaciais*

O usuário informa a extensão espacial de interesse e o UAPÉ devolve os nomes de todas as classes que tenham instâncias contidas nesta área. O resultado da consulta é uma lista de classes como por exemplo, Vegetação, Solo e Estrada.

Para que o usuário possa avaliar se a classe atende às suas necessidades, ele precisa visualizar as geo-classes selecionadas, localizadas dentro da extensão espacial determinada pelo usuário.

O usuário escolhe classes da lista apresentada e o UAPÉ apresenta as geo-classes e suas instâncias dentro da área de estudo.

O usuário seleciona as classes de seu interesse e estas são transportadas para o BDAP. Para que o BDAP seja íntegro, algumas premissas são observadas pelo ambiente: 1) as classes escolhidas devem conter instâncias consistentes quanto à escala e precisão; 2) a cada projeto deve corresponder uma única extensão espacial (mesmo que descontínua).

- *consultas exploratórias*

Estas envolvem tanto consultas a esquema quanto a instâncias do BDAMB. O usuário consulta o BDUAPE sem a restrição espacial da área de estudo. As classes podem ser igualmente visualizadas e aquelas selecionadas pelo usuário passam a fazer parte do BDAP, observando-se as mesmas premissas mencionadas no item anterior.

O Módulo de Consulta permite que o usuário consulte o BDAMB que inclui informações sobre aplicações existentes desenvolvidas sob o UAPÉ. Esta facilidade provê maior suporte na avaliação de decisões tomadas em outros projetos como documentação e modelos associados a relacionamentos.

6.3.4 Módulo da Metodologia

Este módulo foi projetado para assistir o usuário do UAPÉ nas atividades de modelagem e projeto, segundo a metodologia específica desenvolvida como parte da tese, descrita no Capítulo 5.

O Módulo da Metodologia oferece ao usuário dois tipos básicos de assistência. Primeiro, ele direciona a modelagem de dados e processos, seguindo as fases da metodologia. Segundo, ele valida as ações do usuário, apresentando alternativas no caso de ações inválidas.

6.3.5 Módulo de Exportação

O Módulo de Exportação mapeia o BDAP especificado na linguagem do SGBD O_2 , para a linguagem de definição de dados do SIG subjacente. Este módulo decide, de acordo com o conhecimento que o *driver* correspondente possui, que características do GMOD serão traduzidas e de que forma elas serão mapeadas para o SIG (por exemplo, versões, aspectos temporais, relacionamentos causais). Este módulo contém regras de transformação específicas a cada SIG. O Módulo de Exportação pode eventualmente utilizar as facilidades oferecidas pelo SIG para realizar a tradução entre os modelos. A Figura 6.6 apresenta o esquema de operação deste módulo.

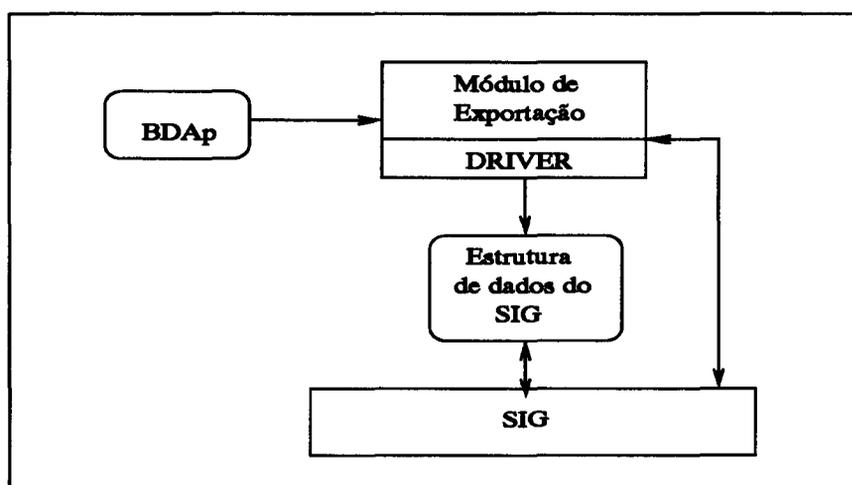


Figura 6.6: Esquema de Operação do Módulo de Exportação

6.3.6 Módulo de Importação

O Módulo de Importação traduz o modelo de dados do SIG subjacente (entidades geográficas e sua definição) para a linguagem de definição de dados do SGBD O_2 . As informações importadas são incorporadas ao BDImp para permitir eventual reuso de dados e esquema por parte do usuário. A Figura 6.7 apresenta o esquema de operação deste módulo.

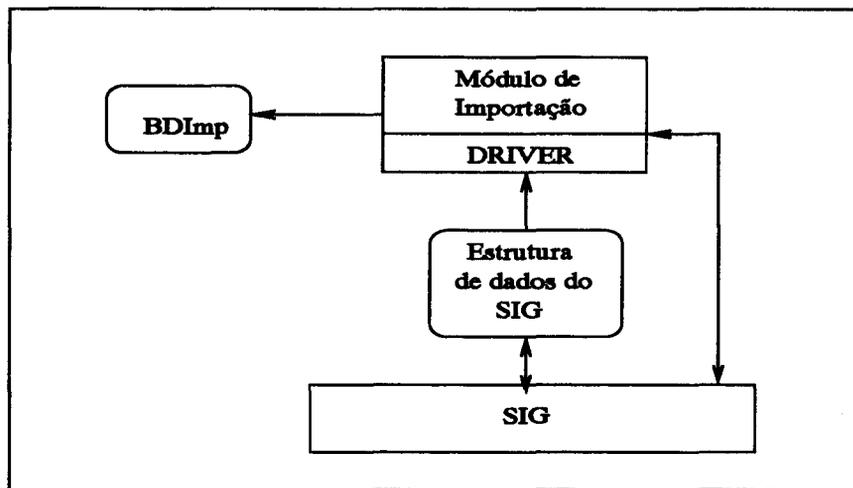


Figura 6.7: Esquema de Operação do Módulo de Importação

O Módulo de Importação não resolve os conflitos que possam resultar da importação das novas entidades, em termos de representação (mesmas entidades com representações diferentes), estrutura de dados, extensão espacial, projeção, escala, conflitos semânticos, entre outros.

Por outro lado, o *driver* de importação deve se preocupar em tratar dos aspectos temporais e de versões, caso o SIG suporte estas características.

6.4 O BDUAPE

O BDUAPE contém as seguintes classes:

- classes do GMOD (em BDImp e BDAmb)
- classes contendo instâncias que permitem geração automática do banco de dados que será exportado para o BDSIG.

As seções anteriores descreveram o primeiro conjunto de classes. As demais classes, descritas no Apêndice A como *meta-classes*, são criadas para permitir aos módulos de importação e exportação mapear dados e métodos entre o BDUAPE e o BDSIG.

Trata-se, na verdade, de um dicionário de dados orientado a objetos onde cada instância de uma (meta)classe especifica uma classe do GMOD.

Seja, por exemplo, a geo-classe Rio, especificada como

```
class Rio inherits Geo_Objeto
  type tuple (nome:      string,
             extensao:   real,
             profundidade: real,
             local:      Linha,
             ...
  method m1 (...)
end;
```

No BDUAPE esta especificação de geo-classe é armazenada como instância da classe Meta-GeoObjeto, com o seguinte conteúdo:

<"Rio", <lista de *Atributos*>, <conjunto de *Métodos*>, <conjunto de *Regras*>, onde *Atributos* são instâncias da classe Atributo do BDUAPE (Apêndice A), como por exemplo, <"nome", "string">, <"extensao", "real">, e assim por diante (vide exemplo acima).

Métodos e *Regras* são instâncias da classe Processo e Regra definidas conforme descrito no Apêndice A.

O apêndice A especifica o BDUAPE segundo esta filosofia.

6.5 Resumo

Este capítulo descreveu o ambiente de modelagem e projeto de aplicações geográficas UAPÉ. O ambiente incorpora a metodologia e o modelo GMOD propostos nesta tese. Os Módulos Interface, Modelagem, Consulta e Metodologia foram implementados como parte de uma tese de doutorado [Oli97] e trabalhos de iniciação científica. O tratamento de versões está especificado em uma dissertação de mestrado [Cur97] e parte da modelagem dinâmica (apenas restrições topológicas) foi implementada em outra dissertação de mestrado [Cil96]. O próximo capítulo mostra como o UAPÉ pode ser usado de forma efetiva em uma aplicação real.

Capítulo 7

Estudo de Caso - Controle de Riscos de Incêndio

Este capítulo descreve brevemente um exemplo de uso do UAPÉ como uma ferramenta para o projeto de uma aplicação geográfica específica e do seu banco de dados.

Os usuários consultam o BDUAPE com o objetivo de identificar dados que possam ser reutilizados. A partir disto, inicia-se o processo de instanciação ou derivação do GMOD com a criação de classes específicas da aplicação. Este processo envolve basicamente a especialização das classes *Convencional* e *Geo-Classe*, além da especificação dos relacionamentos entre elas através da instanciação da classe *Relacionamento*. A partir das classes iniciais, o usuário pode definir instâncias de *Plano de Informação* especificando o processo de geração do PI. Estes processos são armazenados na classe *Processo*. As restrições identificadas e a dinâmica da aplicação são modeladas como um conjunto de regras armazenadas na classe *Regra*. As ações definidas em cada regra referenciam instâncias da classe *Processo*. Também, a cada passo, o usuário pode optar por documentar suas ações criando objetos da classe *Documento* que podem ser associados às instâncias específicas através do relacionamento de associação.

O objetivo da aplicação é desenvolver uma estratégia metodológica de análise de riscos de incêndios florestais, englobando vários fatores ambientais (naturais e antrópicos) e utilizando os recursos de um SIG. A aplicação foi desenvolvida por pesquisadores da Escola Superior de Agricultura "Luiz de Queiroz", da Universidade de São Paulo [FV96]. A área de estudo escolhida foi a Estação Experimental de Tupi pertencente ao Instituto Florestal do Estado de São Paulo. Este mesmo exemplo já foi utilizado no Capítulo 5 durante a exemplificação de uso da metodologia.

7.1 Definição do Projeto

O usuário define o projeto instanciando a classe Projeto e a classe Area de Estudo do modelo GMOD. Os valores dos atributos das instâncias são solicitados pelo UAPÉ que inicializa o BDAP para o projeto "Avaliação de Riscos de Incêndios Florestais" para a Estação Experimental de Tupi.

A classe Documento associa estas instâncias às instâncias do Projeto e de Area de Estudo para documentar a estratégia adotada.

7.2 Modelagem

7.2.1 Classes

- Convencionais - foram utilizadas as classes *Familia* (temporal) e *Indices econômicos* da região
- Geo-Classes
 - Geo-Objetos - foram utilizadas as classes de *Rios e Represas* (vetor-linha), *Construções* (vetor-ponto), *Linhas Elétricas* (vetor-linha), *Estradas e Carreadores* (vetor-linha) e *Rodovias* (vetor-linha)
 - Geo-Campos - foram utilizadas as classes de *Relevo* (RepIsoLinhas), *Uso Solo* (SubDivisaoPlanar), *Temperatura do Ar* (RepPontos) e *Umidade do Ar* (RepPontos)

7.2.2 Relacionamentos

Foram abstraídos os seguintes relacionamentos entre:

- Linhas de Alta Tensão e Carreadores - causal com a seguinte restrição associada: se existe linha então existe carreador (integridade referencial). Um carreador é uma trilha que acompanha toda a extensão de uma linha de alta tensão para possibilitar manutenções na rede elétrica.
- Estrada e Uso Solo - com a seguinte restrição associada: não pode haver construção e plantio dentro de uma determinada faixa de terra dos dois lados da estrada ("buffer").
- Umidade do Ar e Temperatura do Ar - relacionamento quantitativo associado a um processo descrito no item de Modelagem de Processos.

7.2.3 Planos de Informação

Os seguintes PI devem ser especificados para a execução do "modelo" do usuário na fase de processamento:

- Uso da Terra - a partir da aplicação do operador de *adição* de Rios e Represas e Uso Solo
- Vizinhança - a partir da aplicação do operador de *vizinhança* de 1km da Area de Estudo
- Estradas e Carreadores - a partir da *adição* de Rodovias e Estradas e Carreadores
- Declividade - a partir de operações sobre Relevô
- Exposição - a partir de operações sobre Relevô

7.2.4 Modelagem Dinâmica

A modelagem dinâmica abstrai as seguintes restrições que serão posteriormente convertidas na linguagem de regras:

- Se existe linha de alta tensão então existe carreador
- Se *dia sem chuva* (evento) *sempre* (condição) *calcule índice de risco de incêndio* (ação)

7.2.5 Modelagem de Processos

A modelagem de processos identifica os seguintes processos da aplicação:

- geração do PI Uso da Terra
- geração do PI Vizinhança
- geração do PI Estradas e Carreadores (*overlay*)
- geração do PI Declividade
- geração do PI Exposição
- cálculo do índice de risco de incêndio a partir da aplicação de operadores que combinam os PI

7.2.6 Diagrama de Classes

As figuras 7.1, 7.2, 7.3 mostram os diagramas do resultado da modelagem, que correspondem a uma instância do modelo GMOD.

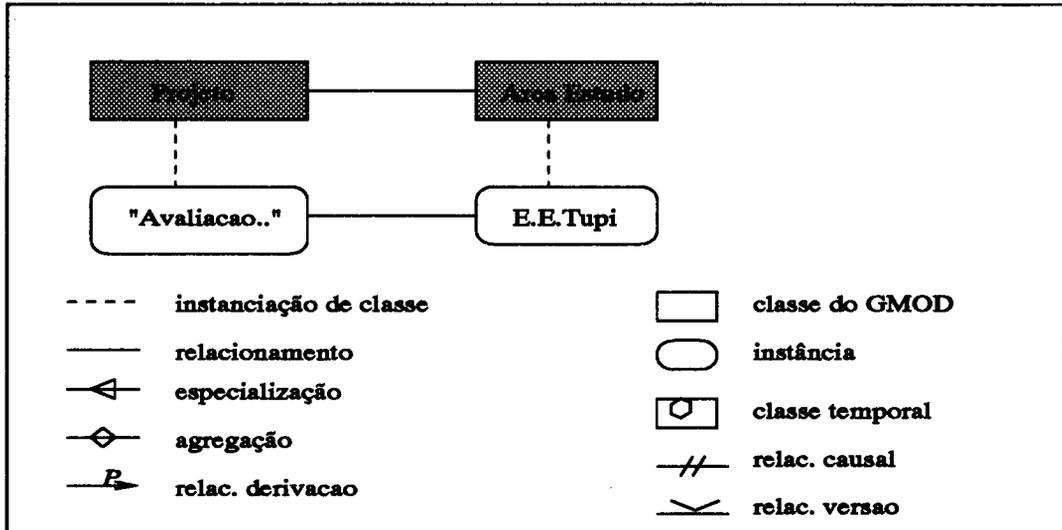


Figura 7.1: Derivação do GMOD para o Estudo de Caso

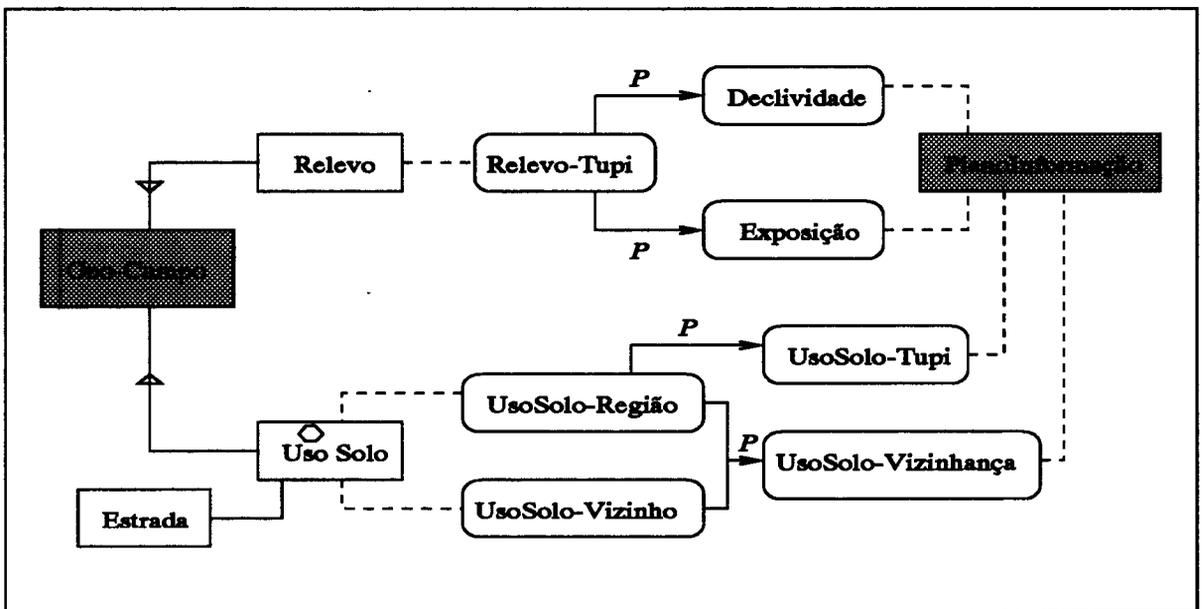


Figura 7.2: Derivação do GMOD - Geo-campo

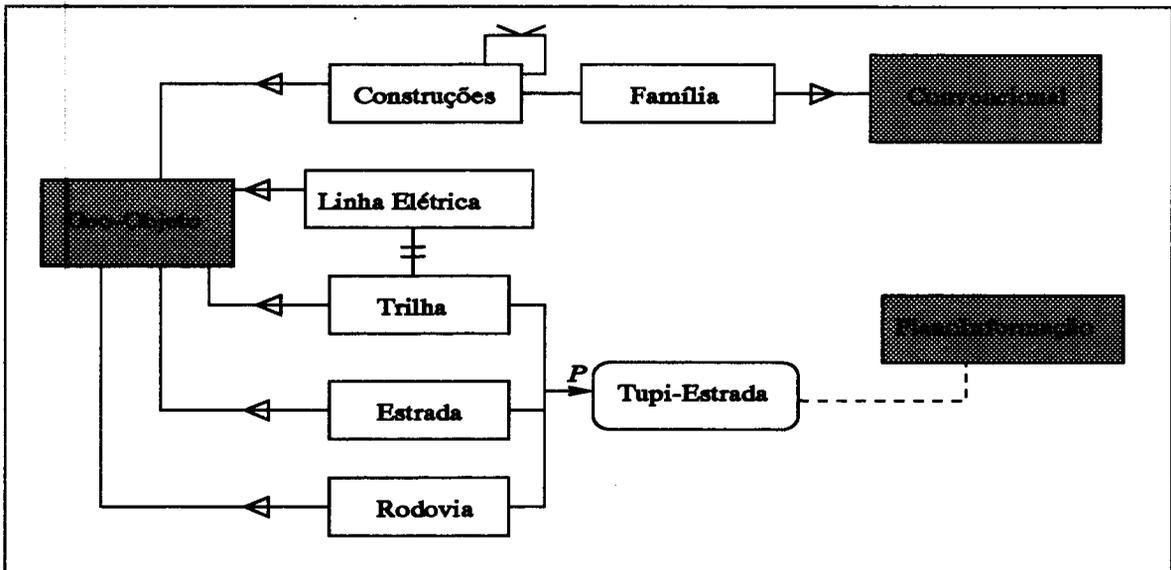


Figura 7.3: Derivação do GMOD - Geo-objeto

7.2.7 Esquema O_2 da Aplicação

O esquema completo das classes não será mostrado para não cansar o leitor, já que a maioria das classes segue o modelo padrão.

A classe Rodovia, por exemplo, dá origem à seguinte definição em O_2 :

```
class Rodovia inherits Geo-Objeto
  type tuple (nome:    string,
              extensao: real,
              local:   Geo-Regiao,
              ...)
  /* todas as operacoes/processos associados a esta classe
  method...
end;
```

A Figura 7.4 mostra o diagrama da modelagem de uma aplicação utilizando o protótipo do UAPÉ.

A Figura 7.5 mostra a tela de interação do Módulo de Consulta e Manipulação, em desenvolvimento, estendendo o sistema GOODIES [OA93].

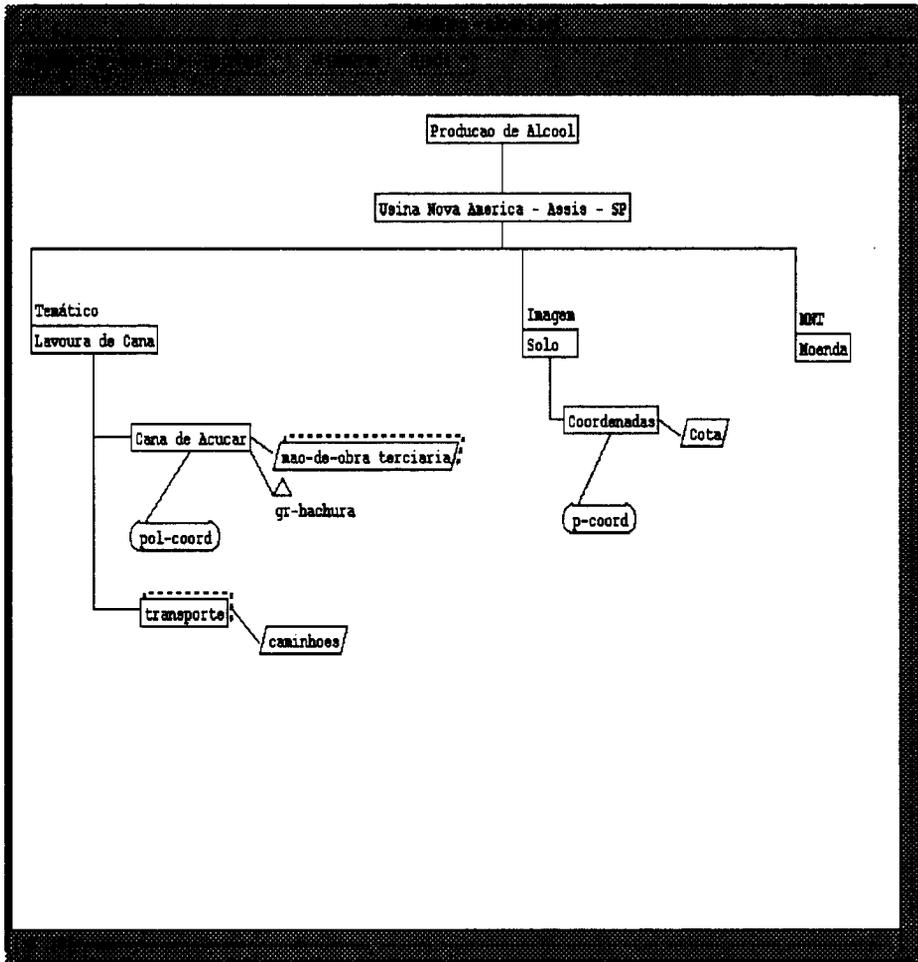


Figura 7.4: Derivação do GMOD para uma Aplicação

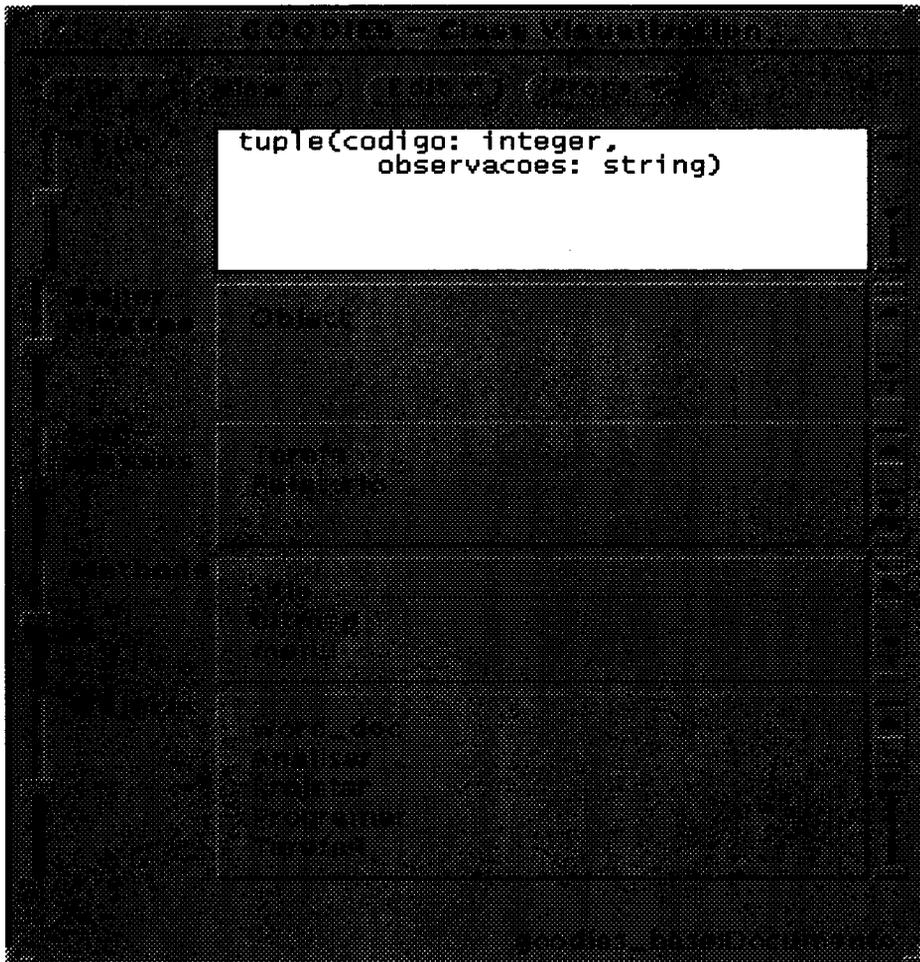


Figura 7.5: Consulta ao BDImp para Reutilização de Dados

7.3 Implementação

Terminada a modelagem da aplicação, ou seja, a instanciação do modelo GMOD, o BDUAPE contém a especificação de todas as classes definidas pelo usuário. A aplicação original foi desenvolvida no SIG IDRISI. No caso do UAPÉ, seria necessário um conversor de estruturas de O_2 para arquivos do SIG IDRISI.

Algumas entidades geográficas foram digitalizadas e armazenadas no banco de dados do SIG. Estas entidades correspondem basicamente às novas instâncias de classes que foram especificadas em função desta aplicação e que não pertenciam ao BDImp, ou a classes que não continham todas as instâncias necessárias para atender esta nova aplicação.

Durante o processamento no SIG, novos PI podem ser criados para ajustes no modelo, assim como também novas classes. Neste caso, o usuário utiliza a linguagem de definição do SIG.

Estas alterações no modelo original gerado pelo UAPÉ, feitas através do SIG, só serão refletidas no BDUAPE se forem migradas pelo Módulo de Importação.

7.4 Avaliação

Nesta aplicação, o mapa-base de risco de incêndio obtido foi avaliado através de entrevistas com os administradores da E.E.Tupi e os resultados foram confirmados preliminarmente. As decisões tomadas influenciaram o planejamento do controle de incêndios, a permissão para queimas controladas, o estabelecimento de zonas de perigo, a previsão do comportamento do fogo, entre outras. A tomada de decisões e os resultados da fase de monitoramento podem ser documentados no UAPÉ associando-se instâncias da classe Documento à instância da classe Projeto para esta aplicação.

7.5 Resumo

Este capítulo mostrou como o UAPÉ pode ser usado de forma efetiva em uma aplicação real. O objetivo da aplicação é desenvolver uma estratégia metodológica de análise de riscos de incêndios florestais, englobando vários fatores ambientais (naturais e antrópicos) e utilizando recursos de um SIG.

Em cada fase da metodologia sob o UAPÉ foram identificados os componentes da aplicação: projeto, área de estudo, classes, relacionamentos e restrições, planos de informação, regras e processos. A descrição destes elementos é armazenada nas classes do BDUAPE para posterior conversão para o modelo de dados do SIG subjacente. Após a conversão seguem-se as atividades de processamento no ambiente do SIG e avaliação dos resultados obtidos.

Capítulo 8

Considerações Finais

8.1 Conclusões e Contribuições

Esta tese apresentou um ambiente de apoio à modelagem e projeto de aplicações ambientais, visando a auxiliar usuários não especialistas em computação a projetar e manipular bancos de dados geo-referenciados. Este ambiente, denominado UAPÉ, engloba um modelo orientado a objetos de dados e processos para criação do banco de dados e uma metodologia especialmente desenvolvida para o projeto dessas aplicações. O ambiente foi projetado de forma a ser aberto e independente do SIG a ser utilizado.

As principais contribuições são:

- um modelo de dados geográficos, GMOD, que modela aspectos estáticos e dinâmicos do mundo real,
- uma metodologia de especificação e projeto de aplicações geográficas para planejamento ambiental, voltada às necessidades do usuário,
- o ambiente UAPÉ, que integra modelo e metodologia de forma a assistir o usuário na tarefa de projetar aplicações ambientais e navegar no banco de dados espacial.

Além disto, vale mencionar que muitos trabalhos publicados preocupam-se em analisar isoladamente cada um dos aspectos ou desafios envolvidos no domínio de aplicações ambientais. Em especial, não há modelo que integre aspectos estáticos e dinâmicos. Este trabalho é resultado de uma análise extensiva sobre o domínio deste tipo de aplicações geográficas. Esta análise permitiu identificar padrões de procedimentos e dados empregados na sua concepção. A compilação destes padrões se traduz no modelo GMOD e na metodologia descrita no Capítulo 5.

O ambiente computacional UAPÉ permite que o usuário desenvolvedor utilize o modelo e a metodologia de forma transparente e amigável. Adicionalmente, o BDUAPE

possibilita o armazenamento e (re)utilização do *conhecimento* das aplicações modeladas sob o ambiente. Este conhecimento é representado pela descrição das entidades, dos seus inter-relacionamentos, dos processos que atuam sobre elas e das regras que orientam a execução destes processos.

Esta característica de *reutilização* do ambiente atende a requisitos fundamentais envolvendo reuso de especificação e de dados de diferentes aplicações geográficas.

A reutilização de padrões de projeto tem merecido destaque na pesquisa em desenvolvimento de software e modelagem de banco de dados com a conscientização de que as etapas de modelagem e projeto determinam a qualidade da aplicação desenvolvida. Os conceitos do paradigma de orientação a objetos, usados no GMOD, provêm mecanismos poderosos de reutilização de estrutura e comportamento das entidades modeladas.

A reutilização de dados é facilitada pelo UAPÉ, que permite ao usuário consultar esquema e dados já existentes visando reaproveitá-los. Esta possibilidade de reutilização de dados entre aplicações geográficas é fundamental para a disseminação e popularização do uso de SIG.

Assim, a descrição em termos de conceitos de bancos de dados e implementação de software, do processo de concepção de aplicações geográficas, no que diz respeito a modelagem, especificação e implementação, é uma das grandes contribuições deste trabalho.

Uma outra contribuição foi a utilização da metodologia OMT e os conceitos relacionados, na modelagem e projeto de aplicações geográficas e na própria concepção do ambiente UAPÉ. Esta experiência demonstrou algumas deficiências do OMT no tratamento e especificação de algumas características típicas de aplicações ambientais. As extensões propostas para a metodologia merecem ser exploradas mais profundamente no sentido de ratificá-las dentro deste contexto.

8.2 Algumas Limitações da Proposta

O UAPÉ como ambiente de modelagem e especificação de aplicações ambientais apresenta algumas limitações que são conseqüências do próprio estágio de desenvolvimento do usuário e abertura dos SIG existentes, como mencionado no Capítulo 1.

A complexidade da dinâmica do comportamento deste tipo de aplicação exige a utilização de métodos e técnicas relativamente complexos de modelagem. Para apoiar a especificação desta dinâmica, o UAPÉ disponibiliza recursos gráficos relativamente simples, basicamente blocos de construção. No entanto, a semântica subjacente pressupõe o entendimento de conceitos básicos relativos a *regras de produção*, *classificação* das entidades geográficas em geo-campos e geo-objetos, *linguagem* de especificação de processos, dentre outros. Esta abordagem pode, portanto, comprometer de alguma forma, a facilidade de uso do ambiente pelo usuário desenvolvedor.

A reutilização de dados em aplicações geográficas, como já mencionado, é um aspecto importante em geo-processamento. O UAPÉ fornece este tipo de facilidade permitindo a importação de dados a partir do SIG subjacente e possibilitando o armazenamento no BDUAPE do *conhecimento* embutido nas diversas aplicações modeladas sob o ambiente. Este *conhecimento* está representado pela especificação de todas as classes de objetos que compõem o modelo da aplicação derivado a partir do GMOD. No entanto, a reusabilidade deste conhecimento está condicionada à abertura do SIG subjacente (importação e exportação de esquemas e dados) e na colaboração e integração entre as operações realizadas sob o ambiente e as realizadas sob o SIG (por exemplo, alterações na especificação original sob o SIG).

Outro aspecto se refere à independência do SIG subjacente. O modelo GMOD reflete as necessidades de modelagem de aplicações ambientais num nível de abstração que protege o usuário de detalhes de implementação das estruturas de dados do SIG. No entanto, para que o Módulo de Exportação consiga mapear a especificação gerada pelo UAPÉ para o ambiente do SIG subjacente, o usuário precisa fornecer informações sobre o tipo de representação espacial das entidades geográficas sendo modeladas. Estas representações são fortemente dependentes das estruturas de dados suportadas pelo SIG. Desta forma, a biblioteca de *drivers* dos Módulos de Importação e Exportação do UAPÉ deve permitir o mapeamento dos tipos de representação suportados pelo GMOD, sem que o usuário precise fornecer detalhes de especificação considerando algum SIG específico.

Finalmente, o modelo GMOD e a metodologia proposta estão voltados ao domínio de aplicações ambientais. Este tipo de aplicação permite o tratamento de diversos aspectos complexos de modelagem estática e dinâmica, mas não exige a exploração de entidades, por exemplo *redes*, típicas de aplicações urbanas. Esta limitação justifica portanto, a extensão proposta na próxima seção.

8.3 Extensões

Há várias extensões possíveis para o trabalho apresentado, tanto práticas quanto teóricas. Estas extensões são descritas a seguir.

Um primeiro conjunto de tarefas diz respeito a tarefas de implementação. Há ainda vários módulos do UAPÉ que precisam ser implementados, em especial o Módulo de Interface com suporte gráfico à modelagem dinâmica e de processos. A arquitetura modular descrita no Capítulo 6 não foi traduzida, na prática, por um conjunto de módulos equivalentes. Seria necessário também, recodificar partes do UAPÉ, principalmente no que se refere à parte de suporte metodológico, atualmente embutida no Módulo de Modelagem e Consulta.

Outro conjunto de ferramentas importantes são os Módulos de Importação e Ex-

portação, que deveriam ser implementados para testar a efetiva independência do UAPÉ com relação ao SIG subjacente. Já foi feita uma especificação destes módulos para o IDRISI, sem implementação, a partir do BDUAPE implementado no O_2 . Uma primeira tentativa poderia ser a implementação de módulos para o GeO_2 , tendo em vista que este SIG foi desenvolvido sobre o próprio O_2 , sendo assim mais fácil a tarefa de conversão entre os ambientes.

A modelagem dinâmica e de processos pressupõe que o sistema subjacente seja um banco de dados ativo, o que limita a implementação sobre SIG comerciais. Uma outra possibilidade seria o tratamento dos processos como uma especificação de uma seqüência de operações na linguagem de especificação de aplicação do SIG subjacente. Mesmo assim, o gerenciamento de restrições seria prejudicado.

Do ponto de vista de modelagem, valeria a pena verificar a aplicabilidade dos conceitos de *frameworks* e *design patterns* [GHJV95] na especificação do GMOD e das classes do UAPÉ. De fato, várias das classes propostas formam um conjunto básico a ser reutilizado e adaptado conforme as necessidades de uma aplicação, de forma semelhante ao que ocorre em *design patterns*. Na verdade, o GMOD pode ser caracterizado como um *framework orientado a objetos* e esta extensão exploraria as conseqüências da apresentação formal do modelo dentro desta perspectiva.

Finalmente, o UAPÉ está voltado a aplicações ambientais, no que tange a metodologia e modelagem de processos. Uma outra extensão é verificar como (e se é possível) estender o ambiente para projeto de aplicações urbanas, por exemplo, AM/FM.

Apêndice A

Apêndice A

Este Apêndice ilustra parte da especificação do UAPÉ através do seu modelo de objetos, resultado das fases de análise e projeto segundo a metodologia OMT, e o esquema do BDUAPE utilizando a linguagem de definição de dados do sistema O_2 .

A.1 Modelo de Objetos

Como destacado no Capítulo 6, o objetivo do UAPÉ é auxiliar o usuário na instanciação do GMOD para compor a especificação da aplicação corrente. O diagrama de objetos (Figura A.1) ilustra as principais classes manipuladas pelo ambiente. Para ficar mais clara a descrição do modelo, no texto vamos fazer referência à classe sendo descrita colocando em destaque o seu nome no modelo.

O ambiente trata principalmente das classes (*Classe*) abstraídas pelo modelo GMOD. Estas classes podem ser importadas pelo ambiente (*Importada*) ou definidas em função da aplicação corrente (*Aplicação*). Para cada uma destas classes podem ser definidos atributos (*Atributo*) e metodos (*Processo*). Os relacionamentos entre as classes da aplicação (importadas e específicas) estão descritos em *Relacionamento* que podem ser associação simples, versão, causal, agregação ou generalização. As classes importadas trazem consigo as suas instâncias (*Instância*).

As regras e processos especificados na modelagem dinâmica e de processos são descritos pelas instâncias das classes *Regra* e *Processo* respectivamente.

A especificação temporal das instâncias é apoiada pela hierarquia de classes temporais especializadas a partir da classe *Tempo*.

O ambiente manipula classes cujas instâncias representam descrições de classes da aplicação (*Meta-Classe*) ou classes cujas instâncias são objetos que descrevem aspectos da aplicação (*Concreta*). As meta-classes descrevem classes de geo-campos (*Geo-Campo*), geo-objetos (*Geo-Objeto*) e entidades convencionais (*Convencional*). Às geo-classes são as-

sociados objetos de tipo de representação (*Representação*) que podem ser pontos (*Ponto*), linhas (*Linha*), polígonos (*Poligono*) ou estruturas celulares (*Grid*).

As classes concretas são: *Projeto* que descreve o projeto que compreende a aplicação sendo modelada, *Area de Estudo* que descreve a área de estudo da aplicação, *Plano de Informação* que descreve cada um dos planos de informação e *Documento* que descreve todos os documentos especificados pelo usuário durante a modelagem.

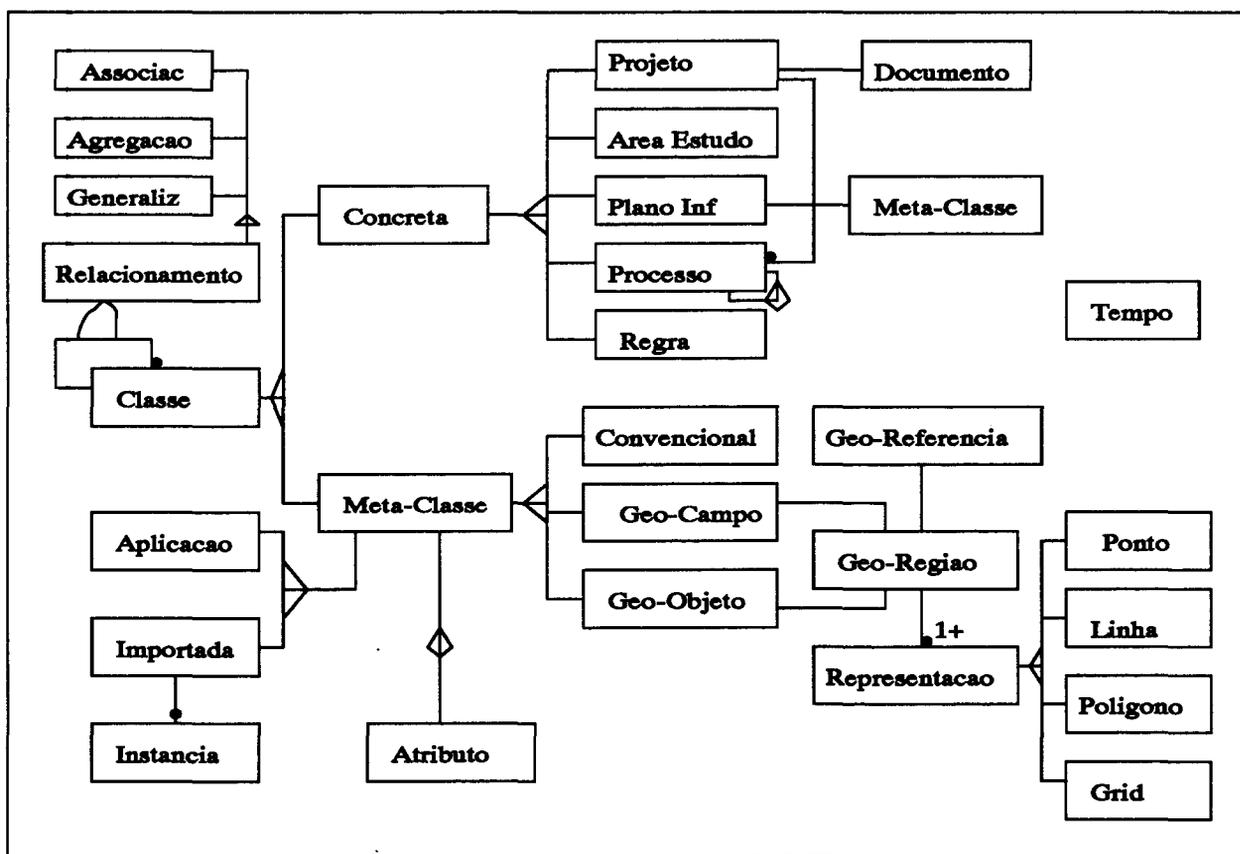


Figura A.1: Modelo de Objetos do UAPÉ

A.2 Esquema do BDUAPE

As instâncias das classes do UAPÉ descrevem todas as entidades geográficas utilizadas por aplicações geográficas instanciadas sob o ambiente. Em função desta característica, o BDUAPE se constitui numa crescente base de conhecimento no domínio de aplicações ambientais.

Para não estender em demasia a descrição das classes, foram omitidas as definições de alguns métodos e atributos implementados, irrelevantes para o entendimento geral do esquema.

```
/* classe basica do BDUAPE
```

```
class Classe inherits Object
  /* informacoes comuns a todas as classes do BDUAPE
end;
```

```
class Concreta inherits Classe
  /* informacoes comuns a todas as classes concretas do BDUAPE
end;
```

```
/* a estrategia metodologica do projeto esta descrita atraves de uma instancia
/* de Documento e
/* list(Processo) determina a sequencia de processos a serem executados
```

```
class Projeto inherits Concreta
  type tuple( projid:    string,
              descricao: string,
              dataref:  Date,
              processo: set(Processo),
              estrategia: set(Documento))
end;
```

```
class Area-Estudo inherits Concreta
  type tuple( projeto:  Projeto,
              descricao: string,
              dataref:  Date,
              regiao:   Geo-Regiao)
end;
```

```
class Documento inherits Concreta
  type tuple( projeto:  Projeto,
              conteudo: string,
              autor:    string,
```

```
                                dataref:  Date)
end;

/* as classes seguintes contem a descricao das entidades de interesse
/* de uma aplicacao modelada/especificada sob o UAPE

class Meta-Classe inherits Classe
    type tuple( atributos:  set(Atributo),
               metodos:    set(Processo),
               restricoes: set(Regra),
               flagver:    ... /* indica se a classe deve ser versionada
               flagtemp:  ...)//* indica se a classe e temporal

method exporta(..) /* este metodo traduz a descricao da entidade
                  /* para uma outra estrutura de dados especifica;
                  /* acionado pelo modulo de exportacao do UAPE
method importa(..) /* importa dados do bd geografico subjacente
                  /* acionado pelo modulo de importacao

end;

class Meta-Convencional inherits Meta-Classe
    type tuple( nome-classe:  string)
end;

class Meta-GeoClasse inherits Meta-Classe
    type tuple( nome-geoclasse: string,
               local:          set(Geo-Regiao))
end;

/* a classe Meta-GeoCampo descreve todas as entidades geograficas
/* da aplicacao, na visao de campos do usuario
/* se o usuario reutiliza os geo-campos de BDImp, a descricao destes
/* esta no atributo inst-geocampo

class Meta-GeoCampo inherits Meta-GeoClasse
    type tuple( inst-geocampo: ... )
```

```
end;
```

```
/* a classe Meta-GeoObjeto descreve todas as entidades geograficas
/* da aplicacao, na visao de objetos do usuario
/* se o usuario importou os geo-objetos de BDImp, a descricao destes
/* vem acompanhado de suas instancias tambem importadas do SIG subjacente
```

```
class Meta-GeoObjeto inherits Meta-GeoClasse
  type tuple( inst-geobjeto:  set(Inst-GeoObjeto))
end;
```

```
class Inst-GeoObjeto inherits Classe
  type tuple( id-instancia:  string,
             val-instancia:  set(Valor-Atributo),
             local-instancia: Geo-Regiao)
end;
```

```
class Valor-Atributo inherits Classe
  type tuple( nome-atributo:  string,
             val-atributo:    string)
end;
```

```
/* as classes seguintes implementam a descricao da extensao
/* espacial das entidades geograficas e como estas extensoes
/* estao representadas
```

```
class Geo_Regiao inherits Classe
  type tuple( local:          set(Representacao),
             sis-referencia:  Geo-Referencia)
end;
```

```
class Representacao inherits Classe

end;
```

```
class Ponto inherits Representacao
```

```
        type tuple( c1: real,
                    c2: real)
end;

class Poligono inherits Representacao
    type tuple( contorno: set(Linha),
               pt-interno: Ponto)
end;

class Linha inherits Representacao

end;

class Isolinha inherits Linha
    type tuple( valor: real,
               arco: Arco)
end;

class Arco inherits Linha
    type tuple( trajeto: list(Ponto))
end;

class Grid inherits Representacao
    type tuple( ponto-origem: Ponto,
               geom-celula: .. )) /* ponto, retangulo, triangulo
end;

class Geo-Referencia
    /* sistema de coordenadas
end;

/* esta classe contem a descricao dos atributos especificados pelo
/* usuario na descricao das entidades geograficas da aplicacao

class Atributo inherits Object
    type tuple( nome-atributo: string,
               tipo-atributo: TipoPrimitivo,
```

```
        comp:          integer,  
        restricao:     Regra)  
end;
```

```
class TipoPrimitivo  
    type set("integer", "string", "real")  
end;
```

```
/* as classes seguintes descrevem os diversos tipos de relacionamentos  
/* entre as entidades geograficas da aplicacao
```

```
class Relacionamento inherits Meta-Classe  
    type tuple( classes:      set(Meta-Classe),  
               nome-relac:  string,  
               processo:    Processo,  
               restricoes:  set(Regra))  
end;
```

```
class R-Agregacao inherits Relacionamento  
  
end;
```

```
class R-Especializacao inherits Relacionamento  
  
end;
```

```
class R-Causal inherits Relacioanamento  
  
end;
```

```
class R-Versao inherits Relacioanamento  
  
end;
```

```
/* esta classe contem a especificacao de todos os PI identificados  
/* na aplicacao
```

```
class PI inherits Concreta
  type tuple( nome:      string,
              definicao: set(Meta-Classe),
              proc-geracao: Processo,
              projid:   Projeto)
end;
```

```
/* as classes que se seguem implementam a hierarquia de classes
/* com raiz na classe Tempo
```

```
class Tempo
  type tuple( extensao:  ObjTempo,
              referencia: RefTemporal)
end;
```

```
class RefTemporal inherits Classe
  type set("utc", "gps", "local")
end;
```

```
class ObjTempo inherits Classe

end;
```

```
class Intervalo inherits ObjTempo

end;
```

```
class AnoMes inherits Intervalo
  type tuple( qanos:  integer,
              qmes:   integer)
end;
```

```
class DiaHorario inherits Intervalo
  type tuple( qdias:   integer,
              qhoras:  integer,
              qminutos: integer,
```

```
                qsegundos: integer)
end;

class DataHorario inherits ObjTempo

end;

class Data inherits DataHorario
    type tuple( ano:    integer,
                mes:   integer,
                dia:   integer)
end;

class Horario inherits DataHorario
    type tuple( hora:    integer,
                minuto: integer,
                segundo: integer)
end;

class Duracao inherits ObjTempo
    type tuple( inicio:  DataHorario,
                extensao: Intervalo)
end;

/* regras identificadas na modelagem dinamica da aplicacao;
/* implementam restricoes de integridade (rest. de cardinalidade,
/* dominio) e referencial de relacionamentos causais e abstraem o
/* comportamento dinamico das entidades da aplicacao

class Regra inherits Concreta
    type tuple( nome:    string,
                evento:  Evento,
                condicao: string,
                acao:    set(Processo))
end;

class Evento inherits Concreta
```

```
end;
```

```
/* processos identificados na modelagem de processos da aplicacao;  
/* podem estar associados a geracao de PI, acoes disparadas por regras  
/* e relacionamentos quantitativos do modelo
```

```
class Processo inherits Concreta  
  type tuple( nome:      string,  
              entradas: set(Atributo),  
              saidas:   set(Atributo),  
              funcao:   string,  
              restricoes: set(Regra),  
              componentes: list(Processo))  
end;
```

Bibliografia

- [AA93] G. Alonso e A. El Abbadi. GOOSE: Geographic Object Oriented Support Environment. In *Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems*, pp. 38–43, 1993.
- [AAT94] E. Apolloni, F. Arcieri, e M. Talamo. Environments for land application development: Issues and design guidelines. In *Proc. International Workshops on Advanced Research in Geographic Information Systems (IGIS)*, Springer Verlag Lecture Notes in CS 884, pp. 1–14, 1994.
- [Abe89] D. Abel. SIRO-DBMS: a Database Toolkit for Geographical Information Systems. *International Journal of Geographical Information Systems*, 3(2):103–116, 1989.
- [AEB95] E. Allen, G. Edwards, e Y. Bedard. Qualitative Causal Modeling in Temporal GIS. In *Proc. COSIT'95*, Springer Verlag Lecture Notes in CS 988, pp. 397–412, 1995.
- [AIS93] R. Agrawal, T. Imielinski, e A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings ACM SIGMOD Conference*, pp. 207–218, 1993.
- [Aro89] S. Aronoff. *Geographic Information Systems*. WDL Publications, Canada, 1989.
- [Bat97] D. F. Batista. Incorporando Características Espaciais em um SGBDOO. Master's thesis, UFPE, outubro de 1997.
- [Bee89] C. Beeri. Formal Models for Object-Oriented Databases. In *Proc 1st International Conference on Deductive and Object-Oriented Databases*, pp. 370–395, 1989.
- [BFAT91] R. Barrera, A. Frank, e K. Al-Taha. Temporal Relations in Geographic Information Systems. *ACM Sigmod Record*, 20(3):85–93, 1991.

- [BK91] N. Barghouti e G. Kaiser. Concurrency Control in Advanced Database Applications. *ACM Computing Surveys*, 23(3):319–344, 1991.
- [Buc94] A. Buchmann. Active Object Systems. In A. Biliris A. Dogac, M. Oszu e T. Sellis, editores, *Advances in Object oriented Database Systems*, pp. 201–224. Springer Verlag, 1994.
- [BX94] M. Batty e Y. Xie. Modelling inside GIS: Part I. Model Structures, Exploratory Spatial Data Analysis and Aggregation. *International Journal of Geographical Information Systems*, 8(3):291–308, 1994.
- [CAM93] S. Chakravathy, E. Anwar, e L. Maugis. Design and Implementation of an Active Capability for an Object Oriented Database. Technical Report CIS-TR-93-001, University of Florida, 1993.
- [Cam95] G. Camara. *Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos*. PhD thesis, INPE, dezembro de 1995.
- [Car89] J. Carter. *Fundamentals of Geographic Information Systems: A Compendium*, capítulo On Defining the Geographic Information System, pp. 3–8. American Society for Photogrammetry and Remote Sensing, 1989.
- [CCH⁺96] G. Camara, M. Casanova, A. Hemerly, G. Magalhães, e C. Medeiros. *Anatomia de Sistemas de Informação Geográfica*. 10 Escola de Computação, 1996.
- [CDvO93] E. Clementini, P. DiFelice, e P. van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-user Interaction. In *Proc Third International Symposium Spatial Databases - SSD*, pp. 277–295, 1993.
- [Cer96] N. Cereja. *Visões em Sistemas de Informações Geográficas - modelo e mecanismos*. Master's thesis, UNICAMP, dezembro de 1996.
- [CGM⁺93] D. Cowan, T. Grove, C. Mayfield, R. Newkirk, e D. Swayne. Managing Environmental Data - an Extension to the GIS Architecture. In *Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems*, pp. 22–27, 1993.
- [CHCS95] S. Carver, I. Heywood, S. Cornelius, e D. Sear. Evaluating Field-based GIS for Environmental Characterization, Modelling and Decision Support. *International Journal of Geographical Information Systems*, 9(4):475–486, 1995.
- [Chr90] N. Chrisman. Deficiencies of Sheets and Tiles: Building Sheetless Databases. *International Journal of Geographical Information Systems*, 4(2):157–168, 1990.

- [Cil96] M. Cilia. Bancos de Dados Ativos com Suporte a Restrições Topológicas em Sistemas de Informação Geográfica. Master's thesis, UNICAMP, março de 1996.
- [CJ90] W. Cellary e G. Jomier. Consistency of Versions in Object-Oriented Databases. In *Proc. 16th VLDB*, pp. 432–441, 1990.
- [CM97] L. M. Cura e C. B. Medeiros. Versões em Bancos de Dados para SIGs. In SBC, editor, *Anais, XII Simpósio Brasileiro de Bancos de Dados*, 1997.
- [Cou92] H. Couclelis. People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in CS 639, pp. 65–77, 1992.
- [Cow90] D. Cowen. GIS versus CAD versus DBMS: what are the Differences? In *Introductory Readings in Geographical Information Systems*, pp. 52–61. Taylor and Francis, 1990.
- [CSE94] E. Clementini, J. Sharma, e M. Egenhofer. Modelling Topological Spatial Relations: Strategies for Query Processing. *Computer & Graphics*, 18(6):815–822, 1994.
- [CT95] J. Clifford e A. Tuzhilin. *Recent Advances in Temporal Databases*. Springer Verlag, 1995.
- [Cur97] L. M. Delval Cura. Tratamento de Versões em Bancos de dados para Sistemas de Informações Geográficas. Master's thesis, UNICAMP, Março de 1997.
- [DGM95] E. V. Dias, S. F. Granado, e G. C. Magalhaes. Uso de versões na garantia de consistência em ambientes mistos de projeto e operação. In *Anais, X SBBD*, pp. 321–333, 1995.
- [DRSM93] B. David, L. Raynal, G. Schorter, e V. Mansart. GEO2: Un Modele de Donnees Geographiques avec O2. In *Proc AFCET Conf.*, pp. 93–103, 1993.
- [EAT92] M. Egenhofer e K. Al-Taha. Reasoning about gradual changes of topological relationships. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in CS 639, pp. 196–219, 1992.
- [Ege92] M. Egenhofer. Why not SQL! *International Journal of Geographical Information Systems*, 6(2):71–86, 1992.

- [EM95] M. Egenhofer e D. Mark. Naive Geography. In *Proc. COSIT'95*, Springer Verlag Lecture Notes in CS 988, pp. 1–15, 1995.
- [EP90] A. Elmagarmid e C. Pu. Introduction to the Special Issue on Heterogeneous Databases. *ACM Computing Surveys*, 22(3), 1990.
- [FCF92] A. Frank, I. Campari, e U. Formentini, editores. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Lecture Notes in CS 639. Springer Verlag, 1992.
- [FG90] A. Frank e M. Goodchild. Two Perspectives on Geographical Data Modeling. Technical Report 90-11, National Center for Geographic Information and Analysis, 1990.
- [FR93] R. Fernandez e M. Rusinkiewicz. A Conceptual Design of a Soil Database for a GIS. *International Journal of Geographical Information Systems*, 7(6):525–540, 1993.
- [FT93] P. Franciosa e M. Talamo. ESPRIT Project EP6881 - AMUSING. *IEEE Data Engineering Bulletin*, pp. 46–50, 1993.
- [FV96] S. F. B. Faray e C. A. Vettorazzi. Evaluation of Fire Risks in Forest Areas using a GIS. Technical report, ESALQ-USP, Department of Rural Engineering, 1996. In Portuguese.
- [GB90] O. Gunther e A. Buchman. Research Issues in Spatial Databases. *ACM Sigmod Record*, 19(4):61–68, 1990.
- [GHJV95] E. Gamma, R. Helm, R. Johnson, e J. Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [GM97] M. A. Gonçalves e C. B. Medeiros. Bibliotecas Digitais para Dados Geográficos. In *Anais, GIS Brasil 97*, 1997. Anais em CD-ROM, 9 páginas.
- [Goo91a] M. Goodchild. Integrating GIS and Environmental Modeling at Global Scales. In *Proc GIS/LIS'91*, volume 1, pp. 117–127, 1991.
- [Goo91b] M. Goodchild. Spatial Analysis with GIS: Problems and Prospects. In *Proc GIS/LIS'91*, volume 1, pp. 40–48, 1991.
- [Goo92] M. F. Goodchild. Geographical Data Modelling. *Computers and Geosciences*, 18(4):401–408, 1992.

- [Goo93] M. Goodchild. *Environmental Modelling with GIS*, capítulo The State of GIS for Environmental Problem-Solving, pp. 7–16. Oxford University Press, 1993.
- [GR93] O. Gunther e W-F Riekert. The Design of GODOT: an Object-oriented Geographic Information System. *IEEE Data Engineering Bulletin*, pp. 4–9, september de 1993.
- [Gut89] R. Gutting. Gral: An Extensible Relational Database System for Geometric Applications. In *Proceedings 15th VLDB Conference*, pp. 33–44, 1989.
- [Her92] J. Herring. TIGRIS: A Data Model for an Object-oriented Geographic Information System. *Computers and Geosciences: An international journal*, 18(4):443–451, 1992.
- [HW90] G. Hunter e I. Williamson. The Development of a Historical Digital Cadastral Database. *International Journal of Geographical Information Systems*, 4(2):169–180, 1990.
- [KE92] Z. Kemp e O. Elizabeth. An Object Model for Distributed Multimedia Geographic Data. In *Proc. EGIS Conference*, 1992.
- [Kim95] W. Kim, editor. *Modern Database Systems: The Object Model, Interoperability, and Beyond*. ACM PRESS, 1995.
- [KT92] Z. Kemp e R. Thearle. Modelling Relationships in Spatial Databases . In *Proc 5th International Symposium on Spatial Data Handling*, pp. 313–322, 1992. Volume 1.
- [Kub84] S. Kubo. Alis: A Geographical Information System for Urban Research. *IEEE Transactions on Knowledge and Data Engineering*, 1984.
- [Lan89] G. Langran. A Review of Temporal Database Research and its Use in GIS Applications. *International Journal of Geographical Information Systems*, 3(3):215–232, 1989.
- [Lan93] G. Langran. *Time in Geographical Information Systems*. Taylor and Francis, 1993.
- [LL93] Y. Leung e K. S. Leung. An Intelligent Expert System Shell for Knowledge-based Geographical Information Systems: 1 - the tools. *International Journal of Geographical Information Systems*, 7(1):189–200, 1993.

- [LMS90] W. Lodwick, W. Monson, e L. Svoboda. Attribute Error and Sensitivity Analysis of Map Operations in Geographical Information Systems: Suitability Analysis. *International Journal of Geographical Information Systems*, 4(4):413–428, 1990.
- [MBJ96] C. B. Medeiros, M. Bellosta, e G. Jomier. Managing Multiple Representations of Georeferenced Elements. In IEEE, editor, *Proc. 7th DEXA96 Workshop*, pp. 364–371, 1996.
- [MC95] C. B. Medeiros e M. Cilia. Maintenance of Binary Topological Constraints through Active Databases. In *Proc. 3rd ACM Workshop on Advances in GIS*, pp. 127–134, 1995.
- [MGR91] D. J. Maguire, M. F. Goodchild, e D. W. Rhind, editores. *Geographical Information Systems - volume I*. John Wiley and Sons, 1991.
- [MMS93] P. Milne, S. Milton, e J. Smith. Geographical Object-oriented Databases: a Case Study. *International Journal of Geographical Information Systems*, 7:39–56, 1993.
- [Mor92] I. Moreira. *Monitoramento Ambiental - Estudos*, capítulo Origem e Síntese dos Principais Métodos de Avaliação de Impacto Ambiental. PIAB, 1992.
- [MP94] C. B. Medeiros e F. Pires. Databases for GIS. *ACM Sigmod Record*, 23(1):107–115, 1994.
- [MP95] C. B. Medeiros e F. Pires. Um Ambiente Computacional de Modelagem de Aplicações Geográficas. *CIG - Cadernos de Informações Geo-Referenciadas*, 1(1), 1995. Revista Eletrônica, editada UNICAMP.
- [Mul91] J. Muller. *Geographical Information Systems - volume I*, capítulo Generalization of Spatial Databases, pp. 457–475. John Wiley and Sons, 1991.
- [NCG89] NCGIA. The Research Plan of the National Center for Geographic Information and Analysis. *International Journal of Geographical Information Systems*, 3(2):117–136, 1989.
- [NM95] H. Naja e N. Mouaddib. The Multiple Representation in an Architectural Application. In *Proc. DEXA Conference*, pp. 237–246, 1995.
- [OA93] J. L. Oliveira e R. O. Anido. Browsing and Querying in Object Oriented Databases. In *Proc. 2nd International Conference on Information and Knowledge Management*, pp. 364–373, 1993.

- [OGI97] OGIS. SAIF. <http://www.wimsey.com/infosafe/saif/saifHome.html>, 1997.
- [Oli97] J. Oliveira. *Interfaces para Sistemas Geográficos*. PhD thesis, IC-UNICAMP, dezembro de 1997. Orientação C. B. Medeiros.
- [OM88] J. Orenstein e F. Manola. PROBE: Spatial Data Modeling and Query Processing in an Image Database Application. *IEEE Transactions on Software Engineering*, 14(5):611–629, 1988.
- [OM96] J. Oliveira e C. B. Medeiros. User Interface Architectures, Languages and Models in Geographic Databases. In *Proc XI SBDD*, pp. 20–42, 1996.
- [OPM97] J. Oliveira, F. Pires, e C. B. Medeiros. An Environment for Modelling and Design of Geographic Applications. *GeoInformatica*, 1(1):29–58, 1997.
- [Peu93] D. Peuquet. What, Where and When - a Conceptual Basis for Design of Spatiotemporal GIS Databases. In *Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems*, pp. 117–122, 1993.
- [Peu94] D. Peuquet. It's about time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. In *Annals of the Association of American Geographers*, 1994.
- [Pim95] F. L. Pimentel. Uma Proposta de Modelagem Conceitual para Dados Geográficos – O Modelo MGeo+. Master's thesis, UFPE, outubro de 1995.
- [PM93] F. Pires e C. B. Medeiros. Uma Metodologia para Projeto de Sistemas de Informações Geográficas. In *Anais, VII Simpósio Brasileiro de Engenharia de Software*, pp. 351–364, 1993.
- [PM97] F. Pires e C. B. Medeiros. *Sistema de Informações Geográficas - Aplicações na Agricultura*, capítulo Bancos de Dados e Sistemas de Informação Geográfica. Ministério da Agricultura - EMBRAPA, 1997.
- [PMB93] F. Pires, C. B. Medeiros, e A. Barros. Modelling Geographic Information Systems using an Object Oriented Framework. In *Proc XIII International Conference of the Chilean Computer Science Society*, pp. 217–232, 1993.
- [PMP93] N. Pissinou, K. Makki, e E. Park. Towards the Design and Development of a New Architecture for Geographic Information Systems. In *Proc. 2nd International Conference on Information and Knowledge Management- CIKM*, pp. 565–573, 1993.

- [PMS96] F. Pires, C. B. Medeiros, e R. F. Santos. Um Ambiente Computacional de Apoio à Concepção de Aplicações Geográficas. In *Anais GIS Brasil*, pp. 545–553, 1996.
- [Pre94] R. S. Pressman. *Software Engineering - A Practitioners Approach*. Mc Graw-Hill Book Company Europe, 1994.
- [Pri89] S. Price. Modelling the Temporal Element in Land Information Systems. *International Journal of Geographical Information Systems*, 3(3):233–244, 1989.
- [RBea91] J. Rumbaugh, M. Blaha, e W. Premerlani et al. *Object Oriented Modeling and Design*. Prentice Hall, New Jersey, 1991.
- [RL95] J. Raper e D. Livingstone. Development of a Geomorphological Spatial Model using Object-Oriented Design. *International Journal of Geographical Information Systems*, 9(4):359–383, 1995.
- [RM92] J. F. Raper e D. J. Maguire. Design Models and Functionality in GIS. *Computers and Geosciences*, 18(4):387–400, 1992.
- [Rom90] G-C. Roman. Formal Specification of Geographic Data Processing Requirements. *IEEE Transactions on Knowledge and Data Engineering*, 2(12):370–380, 1990.
- [SA93] R. Subramanian e N. Adam. The Design and Implementation of an Expert Object-Oriented Geographic Information System. In *Proc. 2nd International Conference on Information and Knowledge Management- CIKM*, pp. 537–546, 1993.
- [SAA+96] A. Saran, D. Agrawal, A. El Abbadi, T. R. Smith, e J. Su. Scientific Modeling using Distributed Resources. In *Proc. ACM GIS'96*, pp. 177–189, 1996.
- [Sam89] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1989.
- [Sam94] P. R. F. Sampaio. Restrições Dinâmicas em Bancos de Dados Ativos Orientados a Objetos. Master's thesis, IMECC-UNICAMP, novembro de 1994.
- [San94] R. Santos. *Planejamento Ambiental - Curso Apostilado*. Faculdade de Engenharia Civil, UNICAMP, 1994.
- [SFGM93] M. Stonebraker, J. Frew, K. Gardels, e J. Meredith. The Sequoia 2000 Benchmark. In *Proc ACM SIGMOD Conference*, pp. 2–11, 1993.

- [She91] I. Shepherd. *Geographical Information Systems - volume I*, capítulo Information Integration and GIS, pp. 337–360. John Wiley and Sons, 1991.
- [SMSE87] T. Smith, S. Menon, J. Star, e J. Estes. Requirements and Principles for the Implementation and Construction of Large-scale Geographic Information Systems. *International Journal of Geographical Information Systems*, 1(1):13–31, 1987.
- [Sno92] R. Snodgrass. Temporal Databases. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in CS 639, pp. 22–63, 1992.
- [SSA95] T. R. Smith, J. Su, e A. E. Abbadi. Computational Modelling Systems. *Information Systems*, 20(2):127–153, 1995.
- [SV92] M. Scholl e A. Voisard. *Building an Object-oriented System – the Story of O2*, capítulo Geographic Applications – an Experience with O2. Morgan Kaufmann, California, 1992.
- [SW95] P. A. Story e M. F. Worboys. A Design Support Environment for Spatial-Temporal Database Applications. In *Proc. COSIT'95*, Springer Verlag Lecture Notes in CS 988, pp. 413–430, 1995.
- [Tol89] C. M. T. Toledo. ANA-RE: um Método para Análise e Especificação de Requisitos. Master's thesis, UNICAMP, junho de 1989.
- [TS94] V. C. Times e A. C. Salgado. Uma Modelagem Orientada a Objetos para Aplicacoes Geográficas. In *Proc. 9o. Simpósio Brasileiro de Banco de Dados - SBBD*, pp. 293–309, 1994.
- [Vra89] R. Vrana. Historical Data as an Explicit Component of Land Information Systems. *International Journal of Geographical Information Systems*, 3(1):33–50, 1989.
- [VvO91] T. Vijlbrief e P. van Osterom. The Geo++ System: An Extensible GIS. In *Proc. European GIS Conference*, pp. 177–189, 1991.
- [WCY89] J. Wu, T. Chen, e L. Yang. QPF: a Versatile Query Language for a Knowledge-based Geographical Information System. *International Journal of Geographical Information Systems*, 3(1):51–58, 1989.
- [Web90] C. Webster. Rule-based Spatial Search. *International Journal of Geographical Information Systems*, 4(3):241–260, 1990.

- [WHM90] M. Worboys, H. Hearnshaw, e D. Maguire. Object-oriented Data Modelling for Spatial Databases. *International Journal of Geographical Information Systems*, 4(4):369–384, 1990.
- [Wor92] M. Worboys. A Generic Model for Planar Geographical Objects. *International Journal of Geographical Information Systems*, 6(5):353–372, 1992.
- [Wor94a] M. Worboys. A Unified Model for Spatial and Temporal Information . *The Computer Journal*, 37(1):26–33, 1994.
- [Wor94b] M. Worboys. Object oriented approaches to georeferenced information. *International Journal of Geographical Information Systems*, 8(4):385–400, 1994.
- [YS91] Z. Yang e D. Sharpe. Design of Buffer Zones for Conservation Areas and a Prototype Spatial Decision Support System (SDSS). In *Proc GIS/LIS'91*, volume 1, pp. 60–70, 1991.