

251841-3

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Uma Abordagem Unificada para
Análise Exploratória e
Simulação Interativa Visual**

por

Carla Maria Dal Sasso Freitas

Tese submetida como requisito parcial
para a obtenção do grau de
Doutor em Ciência da Computação

Prof. Dr. Flávio Rech Wagner
Orientador



Porto Alegre, Agosto de 1994

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Freitas, Carla Maria Dal Sasso

Uma Abordagem Unificada para Análise Exploratória e Simulação Interativa Visual / Carla Maria Dal Sasso Freitas. - Porto Alegre: CPGCC da UFRGS, 1994.

146 p.: il.

Tese (doutorado) - Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1994. Orientador: Wagner, Flávio Rech.

Tese: Computação Gráfica, Simulação, Visualização Científica

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Biblioteca da UFRGS

30425

681.327.16(043)
F866A

INF
1995/251841-3
1995/03/13

Para Thales, Renata e Raquel.

Agradecimentos

Muitas pessoas contribuíram das mais diversas formas para que este trabalho atingisse este estágio. A todas agradeço. Algumas, entretanto, tiveram uma participação mais direta na minha formação, neste trabalho em si e na disponibilidade de recursos que permitiram sua elaboração. A estas quero expressar minha gratidão.

Ao colega e orientador, Flávio Rech Wagner, por acolher a idéia deste trabalho, dar-me a liberdade necessária ao seu desenvolvimento e amadurecimento e, principalmente, pela ajuda indispensável no seu aprimoramento.

Aos professores Dalcídio Moraes Cláudio, Ana Maria de Alencar Price e Léo Pini Magalhães, pelas sugestões por ocasião da apresentação da proposta da tese. Em especial, ao Dalcídio, pelo apoio e interesse constantes. Aos professores Roberto Tom Price e José Palazzo Moreira de Oliveira, pelas críticas e sugestões ao trabalho abordando linguagens visuais e pelo incentivo durante este período.

Ao Instituto de Informática e ao Curso de Pós-Graduação em Ciência da Computação da UFRGS, pela oportunidade e pelos diversos recursos que tornaram disponíveis para a realização de todo o doutorado. Aos professores Ricardo Reis e Clésio Saraiva dos Santos devo um agradecimento especial pelo constante incentivo à área de Computação Gráfica no CPGCC e no Instituto de Informática, respectivamente.

Ao pessoal administrativo do CPGCC e do Instituto, pela simpatia e prestatividade. Ao pessoal dos Laboratórios, em especial ao Luis Otávio, pela paciência e disponibilidade.

Ao CNPq e FAPERGS, pelo apoio financeiro aos projetos que motivaram e sustentaram esta tese. Novamente ao CNPq, pela manutenção da bolsa de pesquisa no período.

Aos alunos Ana Elisa Ferreira Schmidt, Isabel Harb Manssour, Ernesto Lindstaedt e Daniel Francisco Sachett, por darem vida a algumas ferramentas de exploração visual e simulação interativa e pela cessão de imagens e figuras que ilustram parte deste texto. Ao professor Adroaldo Raizer, do GRUCAD/UFSC, por acolher nossas idéias elaboradas através do trabalho de Ana Elisa.

À amiga Lia Goldstein Golendziner, pelas idéias desenvolvidas no projeto AMPLO, as quais me acompanham até hoje, e pelo exemplo de organização e paciência. Ao primeiro mestre, Anatólio Laschuk, por tudo que aprendi, mesmo indiretamente, ao longo destes anos.

À amiga e colega Silvia Delgado Olabariaga, pela motivação e movimentação constantes, pelo muito que aprendi nestes anos de convivência diária, pelo apoio e confiança e... pelas muitas canecas de café!

Ao Thales, pelo amor, incentivo e paciência. Às minhas queridas filhas, Renata e Raquel, por entenderem a importância pessoal deste trabalho. A meus pais, irmãos, cunhados e sogros, pela disponibilidade em "distrair" as meninas em muitas ocasiões ...

Sumário

Lista de figuras	9
Lista de tabelas	11
Resumo	12
Abstract	13
1 Introdução	14
1.1 A análise exploratória de dados e a visualização científica	14
1.2 Simulação e modelagem interativa visual	16
1.3 Linguagens visuais de programação e visualização de programas	19
1.4 Classes de visualização	22
1.5 Objetivos e delimitação do trabalho	23
1.6 Organização do texto	25
2 Casos selecionados	26
2.1 Análise exploratória de dados multivariados populacionais	26
2.2 Análise de dados multivariados espaciais.	27
2.3 Análise exploratória de dados univariados	29
2.4 Exploração de resultados de análise de elementos finitos	30
2.5 Simulação interativa visual de objetos animados dinamicamente	31
2.6 Simulação interativa visual de sistemas discretos	32
2.7 Similaridades e diferenças entre os casos estudados	34
3 Bases para a análise exploratória e a simulação interativa visual	36
3.1 Classificação dos objetos de interesse e seus atributos	36
3.2 Objetivos e tarefas em análise exploratória e simulação interativa visual	40
3.3 Classificação de representações visuais	45
3.4 Metodologia para seleção de representações visuais	47
3.5 Considerações finais	53

4 Recursos visuais e interativos para análise exploratória e simulação	54
4.1 Classes de ferramentas	54
4.2 Representação visual de entidades e atributos	55
4.3 Navegação	60
4.4 Seleção de entidades e atributos	62
4.5 Consulta e comparação de entidades e atributos	65
4.6 Preparação, processamento e controle	66
4.7 Anotação e armazenamento de imagens	69
4.8 Resumo para referência	69
5 VISTA - Plataforma para análise exploratória e simulação interativa visual	74
5.1 Características gerais e abordagens adotadas	74
5.2 A linguagem visual	77
5.2.1 Base de ferramentas e base de entidades	77
5.2.2 Mapeamento e navegação	78
5.2.3 Seleção e consulta	80
5.2.4 Avaliação	84
5.2.5 Preparação do processamento, controle e registro	86
5.3 Modelagem de <i>VISTA</i>	87
5.3.1 Comentários iniciais	87
5.3.2 Modelagem das entidades	89
5.3.3 Modelagem de representações visuais e mapeamento	93
5.3.4 Modelagem das ferramentas	96
5.3.5 Modelo dinâmico	99
6 Discussão	102
6.1 Comparação com outras propostas	102
6.1.1 Critérios de comparação	102
6.1.2 Ponto de vista do usuário-projetista	103
6.1.3 Ponto de vista do usuário-final	108
6.1.4 Ponto de vista de projetistas e usuários-finais	110
6.2 Considerações sobre a implementação das ferramentas	112

7 Conclusões e perspectivas	115
7.1 Resumo	115
7.2 Avaliação e perspectivas	117
Anexo 1: Modelagem das entidades de casos selecionados	122
Anexo 2: Exemplo de uso da metodologia	126
Anexo 3: Classes de <i>VISTA</i>	129
Referências bibliográficas	138

Lista de figuras

Figura 1.1	- Classes de visualização quanto ao momento da interação	22
Figura 3.1	- Representações visuais adequadas em função de {objeto, objetivo}	49
Figura 3.2	- Representações visuais adequadas em função da natureza e da dimensão do atributo	51
Figura 4.1	- Ícones na forma <i>bitmap</i> , subconjunto das faces de Chernoff e ícones 3D	56
Figura 4.2	- Diagramas de modelos na linguagem <i>SLAM II</i> e na linguagem REDES	57
Figura 4.3	- Gráfico de dispersão com dados de medidas de crânio de roedores	57
Figura 4.4	- Modelo geométrico de um contator, com mapeamento para isolinhas	58
Figura 4.5	- Mapa de cores falsas sobre geometria	59
Figura 4.6	- Mapeamento de variáveis escalares no <i>EFVis</i>	60
Figura 4.7	- Recurso de navegação câmera sintética para um objeto	61
Figura 4.8	- Seletor espacial tipo plano de corte	63
Figura 4.9	- Seletor lógico definindo níveis de densidade mapeados para grau de opacidade em mapa volumétrico	64
Figura 4.10	- Seletor lógico definindo valor escalar mapeado para cor e grau de transparência da superfície	64
Figura 4.11	- Ferramenta de avaliação no <i>EFVis</i>	66
Figura 4.12	- Mecanismos de controle de experimentos de <i>ASVI</i>	68
Figura 5.1	- Estrutura geral de <i>VISTA</i>	77
Figura 5.2	- Representação sugerida para a base de ferramentas e base de entidades	78
Figura 5.3	- Ferramenta de navegação câmera , associada a objeto da classe objeto manufaturável	79
Figura 5.4	- Ferramenta focalizador usada sobre a representação por diagrama	80
Figura 5.5	- Uso da ferramenta leitor	81
Figura 5.6	- Uso de seletor individual e leitor sobre entidade da classe entidade natural	81
Figura 5.7	- Mapeamento linhas de contorno aplicado a atributo "potência" na região selecionada pelo seletor janela	82
Figura 5.8	- Ferramenta medidor escalar aplicada ao atributo "permeabilidade" no ponto selecionado pelo seletor ponto	82
Figura 5.9	- Ferramenta de seleção espacial plano de corte , aplicada sucessivas vezes sobre entidade mapeada para modelo geométrico	83

Figura 5.10 - Seletor lógico aplicado a entidade natural, entrada da expressão e aplicação de mapeamento para isosuperfície	84
Figura 5.11 - Mapeamento para cores falsas aplicado ao resultado de avaliador da grandeza "indução magnética" na região selecionada através do seletor janela	85
Figura 5.12 - Avaliador aplicado a atributos de entidade exibidos em com mapeamento para gráfico ; resultados numéricos exibidos à parte	85
Figura 5.13 - Ferramenta de atribuição aplicada a atributo de entidade a ser simulada	86
Figura 5.14 - Facilidades de controle propostas com a ferramenta experimento	87
Figura 5.15 - Modelo da classe ENTIDADE	90
Figura 5.16 - Modelo da classe ELEMENTO-GEOMÉTRICO	91
Figura 5.17 - Modelo da classe ELEMENTO-ESPACIAL	92
Figura 5.18 - Modelo da classe REPRESENTAÇÃO-VISUAL	93
Figura 5.19 - Modelo da classe GRÁFICO	94
Figura 5.20 - Modelo da classe MAPA	95
Figura 5.21 - Modelo da classe MAPEAMENTO	96
Figura 5.22 - Modelo da classe NAVEGADOR	97
Figura 5.23 - Modelo da classe SELETOR	98
Figura 5.24 - Modelo da classe CONSULTA	98
Figura 5.25 - Diagramas de estado para as classes SELETOR LÓGICO e MAP-GRÁFICO	100

Lista de tabelas

Tabela 3.1 - Classificação dos atributos de objetos	38
Tabela 3.2 - Relação das tarefas com os objetivos	43
Tabela 3.3 - Classes de representações visuais identificadas	47

Resumo

Análise exploratória visual e simulação interativa visual são duas áreas afins, porém tratadas separadamente até o momento. O trabalho relatado neste texto objetivou a abordagem unificada destas duas classes de problemas, a partir da identificação de ambas com as questões de linguagens visuais, visualização de dados e de programas.

A partir de estudos de casos considerados representativos destas duas classes (análise de dados multivariados populacionais, análise de dados multivariados espaciais, análise de dados univariados, estudo de campos eletromagnéticos a partir de resultados da análise de elementos finitos, simulação interativa visual de objetos animados dinamicamente e simulação interativa de sistemas discretos) é estabelecida uma classificação de entidades e atributos. Entidades são consideradas como pertencendo a três grandes classes: sistemas compostos por múltiplas entidades, objetos manufaturáveis e entidades ou fenômenos naturais, e são caracterizadas por atributos que podem ser analisados de acordo com sua natureza, o tipo dos valores que podem assumir, a natureza e dimensão do domínio no qual estão definidos. A seguir, são identificados os objetivos e tarefas no tratamento das entidades e atributos. Como objetivos considera-se observação de objetos num contexto, análise da estrutura dos objetos, análise das suas propriedades estáticas ou dinâmicas, análise comparativa de vários objetos e comunicação de informações. Tarefas básicas que servem à realização dos diversos objetivos são navegação, seleção, consulta, preparação e controle de processamento, anotação de textos e armazenamento de resultados. Esta classificação de tarefas permite a determinação das ferramentas básicas necessárias à realização tanto de análise exploratória como de simulação interativa, ambas visuais. É estabelecido um conjunto coordenado de ferramentas que permitem a realização das tarefas mencionadas com as diferentes classes de entidades.

Essencial a estas classes de problemas é a representação visual a ser adotada como forma de transmissão de uma informação a respeito de uma entidade. Este aspecto é enfatizado neste trabalho; é estabelecida uma taxonomia para representações visuais, a saber, ícones, gráficos e tabelas, diagramas e redes, modelos geométricos, mapas e seqüências, de acordo com a informação fornecida e com o atendimento dos objetivos identificados anteriormente. Na seqüência, é proposta uma metodologia para a escolha de representações visuais baseada na classe das entidades sob estudo e nos objetivos do estudo (ou de uma fase do estudo), numa primeira etapa, e na natureza dos atributos e dimensão de seu domínio, numa segunda etapa.

A tese apresenta a abordagem baseada em ferramentas como paradigma básico de linguagem visual para análise exploratória e simulação interativa visual e estende o uso da abordagem declarativa de visualização de programas à questão de visualização de dados. As classes de entidades reconhecidas ao longo do estudo são modeladas de acordo com a abordagem orientada a objetos; da mesma forma, o conjunto de ferramentas é organizado numa plataforma orientada a objetos. Esta plataforma é comparada, do ponto de vista funcional e considerando diferentes classes de usuários, com sistemas existentes. A estrutura proposta é, ainda, discutida como um modelo de referência para análise exploratória e simulação interativa visual.

Palavras-chave: visualização científica, análise exploratória visual, simulação interativa visual, computação gráfica.

Abstract

TITLE: An Unified Approach to Visual Exploratory Analysis and Interactive Simulation.

Two related areas, visual exploratory analysis and visual interactive simulation, have been handled separately until now. This thesis gives an integrated approach to these two classes of problems, based on their relationships to visual languages, data visualization and program visualization. Case studies representing both areas like multivariate analysis of populational data, multivariate analysis of spatial data, analysis of univariate data, study of electromagnetic fields based on finite element analysis, dynamic graphical simulation and interactive visual discrete-event simulation, allow the construction of a taxonomy for entities and attributes. Entities are considered to belong to three classes: multiple-entity systems, manufacturable objects, and natural entities or phenomena. They are characterized by their attributes, which in turn are classified by nature, value types and nature and dimension of their definition domain.

The goals and tasks in the study of entities and attributes are identified. Goals are observing an object in a context, analyzing an object structure, analyzing their static and dynamic properties, comparing objects or different states of objects and communicating information. Tasks are elementary procedures employed to achieve certain goals: navigation, selection, query, modification, processing control, text annotation and data storage. This classification of tasks determine the basic tools that a scientific user needs to visually explore data or conduct visual interactive simulation studies. These tools are organized as a set of interactive procedures that can be applied to visual representations of entities or their attributes.

Visual representations are, of course, essential to visual exploratory analysis and visual interactive simulation, in order to precisely and adequately represent information about entities. This question is emphasized in this work. Based on the information conveyed by the several existing graphical representations as well as the goals established earlier, six classes of visual representations are considered: icons, graphs and tables, diagrams and networks, geometric models, maps, and sequences of these representations. A methodology for selecting the appropriate visual representation is then proposed, based firstly on the goals of the study and the classes of entities and attributes, and secondly on the nature and dimension of the attributes domain.

This thesis presents a tool-centered approach as the paradigm for both visual exploratory analysis and visual interactive simulation. The thesis also extends the concept of declarative visualization employed for program visualization to scientific data visualization. The classes of entities identified in the case studies are modeled using the object-oriented approach; the set of tools defined here is also organized in an object-oriented framework. Our proposal is discussed as a reference model for visual exploratory analysis and visual interactive simulation.

Keywords: scientific visualization, visual exploratory analysis, visual interactive simulation, computer graphics.

1 Introdução

As atividades de análise exploratória visual, visualização científica, modelagem e simulação interativa visual são caracterizadas. É estabelecida a relação destas áreas com a de linguagens visuais, tanto nos aspectos concernentes à construção visual interativa de programas como nos relativos à visualização de programas. Os objetivos deste trabalho são delimitados e a organização do texto é descrita.

1.1 Análise exploratória de dados e visualização científica

Em variadas áreas de atividade - social, científica ou tecnológica - dados são coletados ou produzidos por instrumentos. Na realização de um censo, por exemplo, uma infinidade de dados são levantados acerca da população da região em estudo: número de habitantes por domicílio, idade, grau de escolaridade, renda familiar, profissão, bens móveis e imóveis, etc. No estudo de uma determinada região natural são levantadas informações sobre a flora e a fauna existentes; dependendo do nível de detalhe do estudo, estas informações devem ser sobrepostas, associadas, por exemplo, a um mapa geológico da região. No projeto de um motor, o campo eletromagnético induzido deve ser estudado e, para tanto, após a modelagem geométrica deste objeto pode ser aplicado o método de análise de elementos finitos. A partir dos valores assim obtidos, grandezas físicas podem ser calculadas de forma que o campo eletromagnético possa ser analisado. Radares, estações e satélites meteorológicos coletam uma quantidade bastante grande de dados referentes a temperatura, umidade, precipitação pluviométrica, ventos e pressão atmosférica ao longo do tempo. Tais dados podem ser utilizados tanto para informação através de boletins meteorológicos como para análises posteriores dos fenômenos. Instrumentos como tomógrafos e de ressonância magnética produzem imagens de estruturas internas para possibilitar métodos de análise não-invasiva. Nestes casos, as necessidades giram em torno da extração de características da massa de dados e reconstrução de estruturas internas.

Todos estes exemplos têm em comum a necessidade de exploração dos dados para fins de compreensão dos fenômenos que eles representam. Este é o objetivo principal do que é chamado de análise exploratória de dados [TUKEY 1977], aplicação principal da

visualização científica. "Observar os dados para ver o que eles dizem" requer interagir com os dados, selecionar subconjuntos, procurar características, enfim, realizar *guided tours* [HURLEY e BUJA 1990; YOUNG e RHEINGANS 1991] pela massa de dados. As possibilidades de interação caracterizam, portanto, a análise exploratória de dados.

Conforme mencionado acima, a visualização científica tem por objetivo facilitar a análise exploratória de dados, provendo representações gráficas para os objetos em estudo. Isto não quer dizer que a visualização científica não se preocupe com a apresentação dos dados de forma a facilitar sua comunicação; de fato, muitas aplicações ditas de visualização são apresentação de dados de forma mais compreensível ao usuário leigo, sem a preocupação com os aspectos interativos de exploração. A diferença entre visualização científica e apresentação visual de dados é que nesta última a preocupação é a comunicação de resultados já previamente entendidos; já a visualização científica preocupa-se em possibilitar o entendimento destes dados [EARNSHAW e WISEMAN 1992].

Visualização científica congrega um conjunto de técnicas e ferramentas destinadas a promover novas maneiras de solucionar problemas nas áreas científicas e de engenharia, com tecnologia corrente. Neste sentido, é um processo análogo à análise numérica, ou seja, um conjunto de métodos de solução de problemas, sendo em geral referido como análise visual de dados, devido ao seu objetivo principal, já exposto. Como exemplo de possibilidades de exploração visual de dados pode-se citar a exibição seletiva de informações de censo sobre o mapa de uma determinada região: o número de habitantes pode ser indicado como cota de um gráfico 3D; cores diferentes podem ser utilizadas para indicar a atividade sócio-econômica predominante. Embora estas mesmas imagens possam ser usadas para comunicação de resultados, a associação de informações sobre o mapa, numa primeira etapa, pode ser utilizada como preparação para verificação de correlação entre diferentes variáveis. Esta investigação prévia é facilitada pela possibilidade de selecionar variáveis e formas de apresentação em gráficos freqüentemente simples. Na etapa de exploração de dados não é necessário o uso de imagens sofisticadas e carregadas de informação. Há que se cuidar a quantidade e importância da informação a ser colocada na imagem; freqüentemente os especialistas em visualização tendem a criar imagens esteticamente interessantes, com muitos detalhes, esquecendo que a informação a ser transmitida é o mais importante. *How to lie and confuse with visualization* (sessão do *Visualization'92*) e *The Dangers of Glitziness and Other Visualization Faux Pas* (vídeo apresentado no *Electronic Theater* do

desse modelo a experimentos com parâmetros diferentes, dependendo das hipóteses a serem testadas. A proposição da simulação interativa visual é a condução dos experimentos de simulação pelo usuário, com visualização de dados intermediários e controle completo do experimento e de parâmetros do modelo durante todo o processo. Esta classe de simulação contrapõe-se ao processo tradicional onde o simulacionista recebe os resultados após o experimento e deve analisá-los para testar novas hipóteses ou dar por concluído o trabalho. Simulação interativa visual requer, portanto, exibição e interação durante o processamento, concentrando-se nas necessidades do usuário do modelo. Apenas no final da década de 80 [BELL e O'KEEFE 1987] começou a surgir uma sistematização dos conceitos de VIS e propagou-se a potencialidade dos processos interativos de simulação.

Modelagem interativa visual (VIM - *Visual Interactive Modeling*) abrange os aspectos de construção do modelo com a ajuda de um ambiente gráfico-interativo, ou seja, concentra-se nas necessidades do construtor do modelo de simulação (programador). Surpreendentemente, as fronteiras entre VIS e VIM parecem ter ficado nebulosas desde as primeiras experiências na década de 70 até o final da década de 80 [ROOKS 1991]. A comparação inevitável com as demais aplicações de Computação Gráfica mostra que a distinção entre modelagem e análise já era bem clara em áreas como projeto mecânico, de estruturas e de circuitos em geral [ENCARNAÇÃO et al. 1981; WAGNER et al. 1985]. Aparentemente, a comunidade de simulação tradicional demorou a perceber, de modo mais genérico, as vantagens e a aplicabilidade de técnicas interativas na modelagem de sistemas em geral.

A referência ao termo simulação implica imediatamente o tratamento de sistemas dinâmicos, ou seja, o tratamento de sistemas/objetos cujo comportamento é descrito num modelo, onde a evolução do estado deste é dependente do tempo. Quando a simulação é interativa visual, é possível acompanhar e controlar esta evolução. A simulação das caixas de um supermercado, por exemplo, pode ser feita com a exibição de um diagrama de ícones representando o modelo, incluindo as eventuais filas de clientes à espera de atendimento. O controle interativo da simulação possibilita abertura e fechamento de caixas, alteração da taxa de chegada de clientes às caixas, etc. Estas alterações devem se refletir imediatamente na representação gráfica, mesmo que haja um tempo de "acomodação" do modelo após as alterações. Neste sentido, ROOKS [1991] determina quatro requisitos básicos para um sistema completo de VIS, em termos de possibilidades: intervenção (interação com o modelo), inspeção (acesso aos dados durante

experimentos), especificação (determinação de parâmetros do modelo) e visualização (representação gráfica do comportamento do modelo e relacionamentos de interesse). Em relação à fase de modelagem, entretanto, não existem considerações maiores a respeito. Em geral, é provida uma linguagem gráfica que permite a construção do modelo como uma rede de ícones ou um diagrama de blocos, sendo isto traduzido para código executável ou interpretável. A possibilidade de representação por ícones tornou a área de sistemas de manufatura uma das primeiras a receber ferramentas de modelagem interativa visual com vistas à simulação [BELL e O'KEEFE 1987], excetuando-se aquelas com tradição em projeto auxiliado por computador, como projeto de circuitos elétricos ou eletrônicos.

Uma situação interessante ocorre quando VIM e VIS são disponíveis num único ambiente. Isto significa ter ao alcance do usuário, durante o processo de experimentação, facilidades de alteração do modelo. BELL e O'KEEFE [1987] já alertavam para o risco de modificação do modelo, fugindo da realidade em direção ao sistema ideal. Outros autores, particularmente aqueles preocupados com a análise de resultados de experimentos através de técnicas estatísticas, não são favoráveis à modelagem interativa visual durante a experimentação, pois as alterações nas estatísticas coletadas impediriam a análise global. Seria necessário encerrar a coleta de estatísticas ou estabelecer precisamente o término do novo regime transitório, para reiniciar a coleta de dados.

Do ponto de vista histórico, a utilização de recursos visuais e interativos em simulação de sistemas em geral seguiu com um pouco de retardo o caminho de outras áreas de aplicação. Em particular, isto se deve a dois fatos: (1) a comunidade de simulação de sistemas discretos trabalha com linguagens estáveis e que apenas mais recentemente passaram a incorporar recursos para a melhoria da interface com o usuário; por exemplo, *SIMAN*, *SIMSCRIPT* e *GPSS*; (2) a comunidade de simulação de sistemas contínuos tem trabalhado ao longo dos anos com linguagem Fortran, em computadores de grande porte, desprovidos do estilo de interação orientado a eventos proporcionado pelas modernas estações de trabalho dotadas de periféricos gráficos. A "popularização" das estações de trabalho e a possibilidade de ligação em rede vieram alterar esta situação, a partir do final da década de 80.

Surgiram recursos para a criação de apresentações (em pós-processamento) de resultados de experimentos com modelos de simulação escritos em linguagens convencionais como, por exemplo, *CINEMA* [POORTE e DAVIS 1990] para modelos

SIMAN, e *PROOF* [EARLE et al. 1990] para modelos em GPSS. Surgiram recursos para a construção de interfaces gráficas que permitissem o acompanhamento dos experimentos através de cenários icônicos e gráficos diversos; um exemplo é *SIMGRAPHICS* [CACI 1991] para *SIMSCRIPT II.5*. Sistemas completos de simulação interativa visual foram descritos mais recentemente [ROOKS 1991], sendo caracterizados por agregar aspectos de modelagem, construção de cenários, controle de experimentos e análise de resultados. O sistema descrito por ROOKS [1993] é específico provendo por si próprio as ferramentas necessárias para todas as tarefas vinculadas à simulação: ambiente de pré-processamento para análise de dados brutos, ambiente de modelagem para descrição do modelo, ambiente de preparação de cenários para descrição de experimentos e ambiente de controle de experimentos. Já *OpenSim* [MUTKA e MCKINLEY 1993] é um ambiente aberto onde modelos de simulação (textuais) podem ser agregados e experimentos podem ser definidos. A partir dos nomes das variáveis do modelo, é possível selecionar aquelas que devem ser monitoradas e coletadas em arquivo para posterior análise em pós-processamento (geração de gráficos), dentro do próprio *OpenSim*. Os recursos providos por este sistema assemelham-se aos identificados por FREITAS [1992b] e aos implementados por LINDSTAEDT [em and.] sobre a linguagem *SIMSCRIPT*.

1.3 Linguagens visuais de programação e visualização de programas

A maioria dos modelos de simulação são programas escritos em linguagens de simulação (de propósitos gerais ou específicos) ou de programação. Assim, a noção de linguagem visual de simulação leva ao exame das abordagens adotadas para linguagens visuais de programação (*visual programming languages*). Na realidade, isto faz parte de uma tendência mais ampla, de uso de recursos visuais na área de desenvolvimento de programas e algoritmos. Por linguagens visuais de programação (ou linguagens gráficas de programação) entende-se a classe de linguagens que se vale de símbolos gráficos tanto para representar como para construir programas. Por outro lado, a visualização de programas trata tanto da exibição da estrutura dos programas como da exibição da sua execução, mesmo que estes não tenham sido construídos graficamente. Neste último caso estão a visualização de estruturas de dados e dos algoritmos implementados pelos programas [MYERS 1990; FREITAS 1992a]. Dentro de visualização de programas, MYERS [1990] distingue: (a) visualização de código, como sendo a exibição do código

textual com anotações ou do diagrama da estrutura do programa; (b) visualização de estruturas de dados, como a apresentação explícita das estruturas declaradas no programa; e (c) visualização de algoritmos, como a apresentação de uma imagem abstrata representando o comportamento do programa.

O presente trabalho envolve aspectos tanto de visualização de dados como de visualização de programas, já que muitos modelos de simulação são implementados como programas escritos em determinada linguagem. No entanto, a preocupação maior em simulação interativa visual é a observação do estado interno do modelo (programa de simulação) durante a realização de experimentos. Os resultados finais são encarados como o estado do modelo após determinado tempo. Como este estado interno é descrito através de estruturas de dados, passa-se a considerar a visualização de programas e algoritmos do ponto de vista da observação das estruturas de dados. Duas abordagens são encontradas [ROMAN e COX 1989]: abordagem **imperativa** e abordagem **declarativa**. Na primeira, a exibição das estruturas internas é obtida com a inclusão de código extra no programa para que sejam mostradas variáveis e estruturas de interesse, quando ocorrem alterações dignas de serem reportadas ao observador. Na segunda, a exibição é considerada uma abstração, um mapeamento de estados do programa para estados de imagens. Neste caso, declarações devem estabelecer esta associação e o sistema deve realizar este mapeamento automaticamente. Estas declarações não devem estar presentes no código do programa; são um conjunto de regras que ao serem aplicadas às estruturas de dados devem resultar na alteração dos atributos das imagens.

Considerando o modelo subjacente à linguagem visual, ou seja, a construção visual base da linguagem, classificamos as linguagens [FREITAS 1992a] em:

1. icônicas, quando as construções visualizadas e/ou manipuladas pelo usuário são ícones. O processo de construção de uma sentença nessas linguagens é basicamente a manipulação de ícones. Exemplos são *Pict/D* [GLINERT e TANIMOTO 1984] e *Tinkertoy* [EDEL 1988];

2. diagramáticas, quando a linguagem utilizada é baseada em diagramas. É a abordagem utilizada em *Pecan* [REISS 1984] e *PegaSys* [MORICONI e HARE 1985];

3. tabulares, quando o processo de construção de sentenças nessas linguagens corresponde ao preenchimento de tabelas ou formulários. Um exemplo é *Formal* [SHU 1985];

4. demonstracionais, quando as construções são exemplificadas pelo usuário para o sistema. Exemplos dos resultados são fornecidos ou o funcionamento de uma operação é demonstrado pelo usuário, através de técnicas interativas. Esta abordagem foi introduzida por *Pygmalion* [SMITH 1975] e reforçada pelo sistema de Myers [MYERS e BUXTON 1987].

A abordagem diagramática é, sem dúvida, a mais amplamente utilizada, tanto em termos de programação como em construção de aplicativos; basta observar os sistemas de visualização AVS [UPSON et al. 1989], apE [DYER 1990] e Iris Explorer [SILICON GRAPHICS 1991], que utilizam o paradigma *data flow* para a construção de diagramas de módulos representando um aplicativo em construção. As demais abordagens têm mais desvantagens que vantagens em termos de generalidade de uso. A abordagem icônica é vantajosa em linguagens cujas construções têm uma uniformidade intrínseca, como *Lisp*. O paradigma de preenchimento de formulários, apesar de originar interfaces fáceis de usar, não é tão difundido quanto a abordagem diagramática.

Assim, não se pode afirmar que uma abordagem reúne mais condições que as outras de ser adotada mais amplamente. Em se tratando de linguagens de programação, o uso de uma única abordagem tende a restringir a generalidade de sua aplicação. Isto quer dizer que apenas para situações específicas é interessante adotar uma dentre as quatro abordagens citadas (icônica, diagramática, tabular e demonstracional). Para a obtenção de um ambiente de programação de propósitos gerais, uma alternativa viável parece ser o uso simultâneo de várias abordagens como em REISS [1987] e BORGES e JOHNSON [1990]. Tais ambientes (denominados "multiparadigma") requerem uma semântica interna consistente de modo que representações diferentes do mesmo objeto possam ser mapeadas entre si.

Considerando estas mesmas abordagens, mas para o caso de visualização de dados e programas, verifica-se que para visualização de estruturas de dados, incluindo animação de algoritmos, a abordagem icônica é largamente empregada [BROWN 1988], enquanto que, para a visualização da estrutura de um programa em execução, o mais usual é o

acompanhamento da passagem do controle de execução pelos diversos módulos, em um diagrama [REISS 1984].

Relacionando as necessidades de visualização em simulação com linguagens visuais vemos que a especificação do modelo de simulação (VIM) através de construções gráficas enquadra-se como linguagem visual de programação enquanto a exibição de dados intermediários (VIS) utiliza as técnicas de visualização de dados e de programas, em geral na forma imperativa.

1.4 Classes de visualização

Das considerações feitas nas seções anteriores, a respeito de visualização de dados e interação durante este processo, podem ser estabelecidas classes de visualização. MARSHALL et al. [1990] estabelecem três momentos para interação e visualização: pós-processamento, monitoramento (*tracking*) e controle (*steering*).

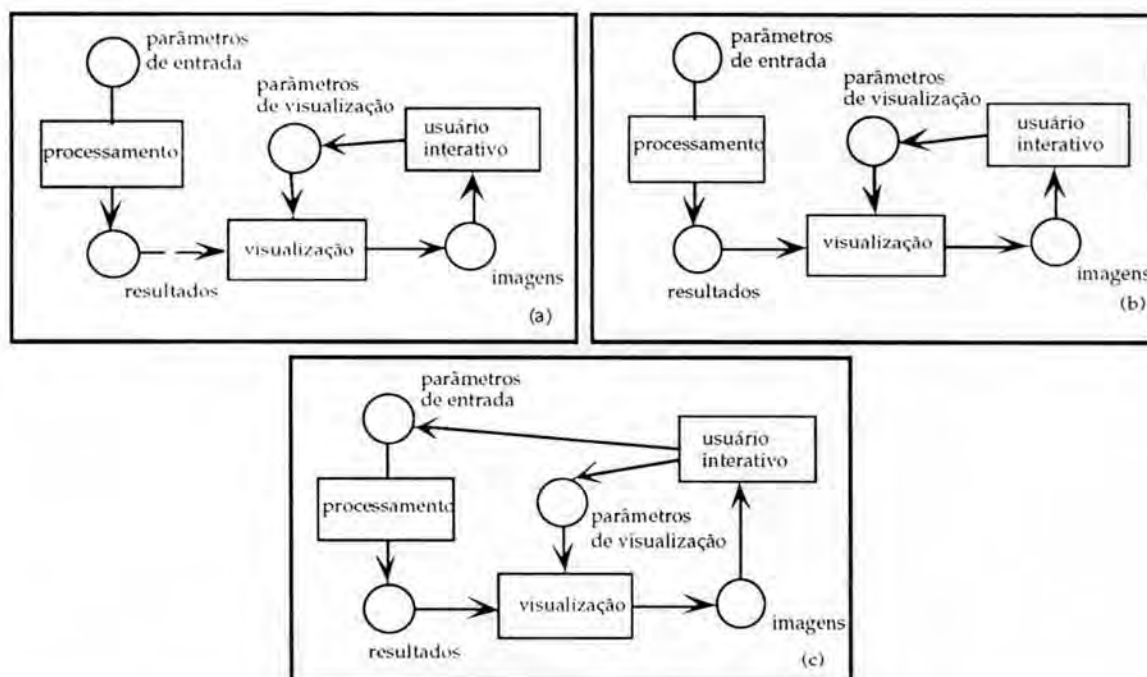


Figura 1.1 - (a) Visualização em pós-processamento; a seta tracejada indica que os resultados passam à visualização após o término do processamento. (b) Visualização para monitoramento do processamento. (c) Visualização para controle do processamento.

Na figura 1 identifica-se como visualização em *pós-processamento* (a) aquela categoria onde apenas os resultados finais de uma computação são visualizados, ou seja, não há possibilidade de visualização de resultados intermediários, como é o caso da categoria *monitoramento* (b). Já na classe visualização para *controle* (c), o usuário pode interativamente alterar parâmetros da computação em andamento. Esta última classe engloba as aplicações de simulação interativa visual.

No caso de visualização em pós-processamento, as possibilidades de interação são variadas: seleção de entidades e atributos a serem observados, seleção das representações associadas a estes elementos, observação quadro a quadro de um fenômeno, etc. O monitoramento de uma computação através de visualização deve prever ativação e desativação do acompanhamento da evolução das variáveis e alteração do passo do monitoramento, além de todas as outras facilidades já citadas. O controle do processamento corresponde à classe mais completa em termos de visualização e interação; além do monitoramento da execução de experimentos, estes podem ser interrompidos e o modelo pode ser alterado através de ações como ajuste de coeficientes de equações, alteração da taxa de geração de determinadas entidades do modelo, dentre outras possibilidades.

1.5 Objetivos e delimitação do trabalho

O trabalho foi definido tomando dois objetivos. O primeiro, "estabelecer bases para o desenvolvimento de sistemas de análise exploratória visual e simulação interativa visual", é derivado do fato destas áreas serem relativamente recentes e, por isso, carecerem de sistematização. Esta sistematização envolve aspectos metodológicos e funcionais, isto é, como focar problemas de análise exploratória e simulação interativa visual e que funções são necessárias para conduzir tais problemas a uma solução. O segundo objetivo é o de desenvolver uma plataforma de software onde tais aspectos sejam considerados, isto é, onde os requisitos determinados no estudo dos aspectos funcionais sejam concretizados e reunidos de forma a suportar o enfoque estabelecido para a resolução de problemas nestas duas áreas.

Quanto ao primeiro objetivo, cabe ressaltar a diferença entre aspectos metodológicos e funcionais. Por aspecto metodológico entende-se uma metodologia para a seleção e uso de modelos visuais, em diferentes níveis de envolvimento com estas

Na figura 1 identifica-se como visualização em *pós-processamento* (a) aquela categoria onde apenas os resultados finais de uma computação são visualizados, ou seja, não há possibilidade de visualização de resultados intermediários, como é o caso da categoria *monitoramento* (b). Já na classe visualização para *controle* (c), o usuário pode interativamente alterar parâmetros da computação em andamento. Esta última classe engloba as aplicações de simulação interativa visual.

No caso de visualização em pós-processamento, as possibilidades de interação são variadas: seleção de entidades e atributos a serem observados, seleção das representações associadas a estes elementos, observação quadro a quadro de um fenômeno, etc. O monitoramento de uma computação através de visualização deve prever ativação e desativação do acompanhamento da evolução das variáveis e alteração do passo do monitoramento, além de todas as outras facilidades já citadas. O controle do processamento corresponde à classe mais completa em termos de visualização e interação; além do monitoramento da execução de experimentos, estes podem ser interrompidos e o modelo pode ser alterado através de ações como ajuste de coeficientes de equações, alteração da taxa de geração de determinadas entidades do modelo, dentre outras possibilidades.

1.5 Objetivos e delimitação do trabalho

O trabalho foi definido tomando dois objetivos. O primeiro, "estabelecer bases para o desenvolvimento de sistemas de análise exploratória visual e simulação interativa visual", é derivado do fato destas áreas serem relativamente recentes e, por isso, carecerem de sistematização. Esta sistematização envolve aspectos metodológicos e funcionais, isto é, como enfocar problemas de análise exploratória e simulação interativa visual e que funções são necessárias para conduzir tais problemas a uma solução. O segundo objetivo é o de desenvolver uma plataforma de software onde tais aspectos sejam considerados, isto é, onde os requisitos determinados no estudo dos aspectos funcionais sejam concretizados e reunidos de forma a suportar o enfoque estabelecido para a resolução de problemas nestas duas áreas.

Quanto ao primeiro objetivo, cabe ressaltar a diferença entre aspectos metodológicos e funcionais. Por aspecto metodológico entende-se uma metodologia para a seleção e uso de modelos visuais, em diferentes níveis de envolvimento com estas

representações: o projetista de um modelo de simulação, o projetista de um aplicativo específico de visualização ou um usuário final, interagindo com seus modelos. Por aspecto funcional entende-se um conjunto de recursos visuais e interativos para a exploração dos dados e entidades sob estudo e que suporte a metodologia proposta. Considerando os trabalhos realizados até o momento em visualização científica, a preocupação maior tem sido com a produção de ferramentas de visualização, tanto aplicações específicas para determinadas áreas (modelagem de moléculas, simulação da dinâmica de fluidos, por exemplo) como desenvolvimento de técnicas de visualização dependentes dos tipos dos dados a serem visualizados (algoritmos para exibição de grandezas vetoriais baseados no rastreamento de partículas, por exemplo). Existe uma grande variedade de sistemas para visualização, desde pacotes simples de apresentação de dados até construtores de aplicativos baseados em bibliotecas de módulos.

Na junção dos aspectos metodológico e funcional concentra-se o segundo objetivo, que é o de especificar uma plataforma de software dentro de uma abordagem que contemple tarefas de análise exploratória e simulação interativa visual ao invés de apenas apresentação gráfica de resultados. No caso dos pacotes de apresentação de dados, existe um repertório fixo de representações visuais que podem ser escolhidas pelo usuário; por exemplo, gráficos diversos, contornos por linhas de isovalor, contornos por superfícies de isovalor, linhas de trajetória de fluxo, etc [BRODLIE et al. 1992]. No caso de aplicações específicas desenvolvidas sobre um determinado sistema de visualização, apenas as representações previstas pelo programador da aplicação podem ser escolhidas pelo usuário. Este é também o caso do usuário de um modelo de simulação desenvolvido dentro do contexto de simulação interativa visual: as representações visuais são escolhidas pelo projetista do modelo e o usuário final, em geral, não tem possibilidade de alterá-las. Assim, em relação ao aspecto metodológico, estes sistemas nada oferecem para o usuário final; ele deve se ater às facilidades de representação visual oferecidas pela plataforma escolhida. Em alguns casos, os sistemas oferecem uma abordagem para o desenvolvimento de aplicativos como, por exemplo, o diagrama de fluxo de dados do AVS [UPSON et al. 1989] e do *Iris Explorer* [SILICON GRAPHICS 1991]. Em relação ao aspecto funcional, são oferecidas técnicas de visualização variadas, mas sem o devido suporte para exploração dos dados ali representados. O aspecto interativo deve ser programado sob medida pelo projetista do aplicativo ou do modelo de simulação. Igualmente limitada é a extensão dos sistemas com novas técnicas de visualização; os sistemas baseados em bibliotecas de módulos são os mais flexíveis neste aspecto porque novos módulos podem ser desenvolvidos e incorporados à biblioteca.

O presente trabalho parte, pois, do desenvolvimento de uma metodologia para a seleção de representações visuais, tanto por parte do projetista de um modelo ou aplicativo como por parte de um usuário final, estabelece as funcionalidades necessárias em termos de técnicas de visualização e interação, de forma a suportar esta metodologia, e, finalmente, sistematiza um conjunto de recursos visuais e interativos através de uma plataforma orientada a ferramentas de análise e simulação. Por ser uma plataforma, este ambiente deve ser aberto e extensível, por natureza, o que motivou a escolha da abordagem orientada a objetos. É proposto um paradigma para linguagens visuais de exploração, contemplando as necessidades de análise exploratória de dados em geral e do controle interativo visual sobre modelos de simulação. Ambas as situações são tratáveis na plataforma cuja arquitetura é proposta.

1.6 Organização do texto

O texto apresenta, no capítulo 2, algumas situações selecionadas onde se apresentam problemas de análise exploratória e simulação interativa visual, envolvendo visualização e interação. Estas servem de motivação para o capítulo 3, onde são apresentadas as questões básicas sobre análise exploratória e simulação interativa visual, culminando com a apresentação de uma metodologia para a seleção de representações visuais de entidades e seus atributos. Os requisitos de interação e visualização necessários do ponto de vista funcional são apresentados no capítulo 4, enquanto o capítulo 5 propõe a arquitetura de uma plataforma que unifica os recursos para análise exploratória e simulação interativa visual. Esta plataforma dá suporte à metodologia e implementa os recursos interativos visuais explicitados no capítulo 4, reunidos sob a forma de uma linguagem visual de exploração e controle. O capítulo 6 discute a plataforma em relação a outros sistemas. O capítulo 7 resume o trabalho, avalia-o de acordo com a sua contribuição às áreas abordadas, localizando-o em relação ao que seria um modelo de referência para sistemas de análise exploratória e simulação interativa visual, e delinea as perspectivas de sua continuidade.

2 Casos selecionados

Alguns problemas representativos de diversas áreas foram estudados e são aqui descritos os seus objetivos e as conseqüentes necessidades em termos de tratamento dos dados. Os problemas envolvem classes diferentes de entidades e foram escolhidos com base tanto na observação das aplicações descritas na literatura como no acompanhamento e desenvolvimento de aplicações específicas. Os aspectos aqui enumerados apóiam a identificação das bases metodológica e funcional para análise exploratória e simulação interativa visual.

2.1 Análise exploratória de dados multivariados populacionais

Tomemos como problema desta classe o estudo de populações de determinado gênero dentre os mamíferos, particularmente as espécies que se distribuem numa determinada área zoogeográfica [FREITAS 1990]. Pode-se assumir que um dos objetivos (aquele que utilizaremos como exemplo) é verificar as diferenças entre as espécies em termos morfológicos.

Para tanto, são coletadas informações a respeito de cada exemplar: nome da espécie, sexo, ponto de coleta (latitude, longitude) e 16 medidas de crânio. A análise destes dados exige a plotagem de gráficos diversos, o tratamento dos dados quantitativos em termos de normalização, a obtenção de medidas estatísticas, a realização de testes e a utilização de procedimentos de análise multivariada [JOHNSON e WICHERN 1982], em especial, análise de componentes principais. Do ponto de vista de modelagem cada entidade pode ser interpretada como um ponto no espaço multidimensional de atributos.

Do ponto de vista de necessidades em termos de visualização, as representações visuais são, na maioria das vezes, gráficos 1D, 2D ou 3D, com ícones ou cores para distinguir diferentes classes (espécies e sexo, neste caso). Deve ser possível construir gráficos para observação da relação entre diferentes medidas e características, como realizado experimentalmente por OLIVEIRA [1993].

Os gráficos são utilizados tanto para representar dados "brutos" como para exibir o resultado de cálculos e análises. Deve ser possível, por exemplo, exibir informações de

média, frequência, desvio padrão, etc das diferentes espécies e/ou por sexo, em gráficos de barras. Ou, na técnica de análise de componentes principais, as matrizes de correlação e de covariância das 16 medidas de crânio são utilizadas: os 16 autovetores obtidos formam um sistema de coordenadas multidimensional onde cada exemplar é localizado como um ponto. A observação do agrupamento dos pontos num gráfico com os três eixos de maior autovalor permite analisar diferenças entre as classes de animais.

Um aspecto importante é a possibilidade de comparação de resultados de diferentes sessões de análise, seja em papel, seja numa sessão posterior. Para tanto, é necessário dispor de facilidades de salvamento de resultados em arquivos, *log* das sessões e de reprodução dos resultados (visuais ou numéricos) em papel.

A leitura dos dados brutos de determinados exemplares exibidos nos gráficos é igualmente necessária, para a construção de hipóteses; estes dados devem poder ser exibidos na forma de tabelas, seletivamente, ou seja, deve poder ser feita uma operação tipo *browsing*.

2.2 Análise de dados multivariados espaciais

Dados multivariados espaciais, como o próprio nome representa, são dados associados a posições no espaço. Genericamente, este domínio pode ter dimensão maior ou igual a dois. Exemplos clássicos vêm da análise de dados meteorológicos, dinâmica de fluidos, análise de dados obtidos a partir de sensores médicos, etc.

Tomaremos como primeiro caso para estudo, a análise de dados médicos [MANSSOUR e JUNGBLUT 1992; SACHET 1993] a partir de um conjunto de imagens tomográficas. As informações presentes no modelo correspondem a um conjunto de matrizes de densidade (fatias) que, justapostas, formam um volume eventualmente não-uniforme em função da diferença entre o número de fatias e as dimensões das matrizes bidimensionais. O objetivo neste caso é explorar visualmente este volume e, além disso, obter modelos geométricos de regiões com característica expressa por um determinado intervalo de valores de densidade. Análise quantitativa em termos de área e/ou volume desses modelos geométricos deve ser igualmente possível.

Do ponto de vista de representação visual, cada fatia pode ser exibida isoladamente; a exibição do volume é útil com transparência para a visualização do interior [MANSSOUR 1994]. Informações derivadas das fatias originais são necessárias e devem poder ser representadas graficamente. Histogramas de frequência, cortes do volume em outras orientações e contornos de regiões em cada fatia são exemplos típicos. A extração de modelos geométricos de regiões internas selecionadas [SACHET em and.] pode ser seguida de uma visualização deste modelo sozinho ou no contexto do volume.

A leitura de valores da densidade de pontos selecionados só é necessária para guiar a escolha de intervalos de densidade para a delimitação de estruturas a serem visualizadas ou a terem seus modelos geométricos extraídos. Em ambos os casos é necessária a vinculação de parâmetros de cor e nível de opacidade aos intervalos de densidade selecionados.

No caso acima, as informações são obtidas através de fatias regularmente espaçadas e, nestas, em pontos igualmente distribuídos de forma regular. Entretanto isto não pode ser assumido como ocorrendo em todas as situações [TREINISH 1992]. Muitas vezes, dados são coletados e associados explicitamente a coordenadas de um espaço, caracterizando dados esparsos. Por exemplo, concentrações químicas amostradas em diferentes profundidades do solo ou temperaturas obtidas em diferentes pontos da atmosfera. O objetivo é, na maioria das vezes, representar a variável medida como uma função das coordenadas do ponto [NIELSON et al. 1991] de modo que esta função possa ser avaliada numa malha regular e, então, exibida e analisada quantitativamente.

A análise de dados espaciais esparsos requer, portanto, além do uso de técnicas de representação visual adequadas à dimensão do espaço, o emprego de técnicas de interpolação ou aproximação. É importante, para efeito de análise, a observação conjunta do resultado de duas ou mais técnicas. Na leitura de valores, o resultado da avaliação da função no ponto desejado deve poder ser comparado com o dado bruto, para observação do erro associado à técnica de interpolação ou aproximação.

2.3 Análise exploratória de dados univariados

Dados univariados são caracterizados como uma coleção de valores de uma única variável, medida num determinado domínio unidimensional. Tomemos como exemplo a medida da emissão luminosa de uma estrela ao longo do tempo, que pode ser utilizada para deduzir a estrutura interna da mesma; alterações nesta medida denotam oscilações estelares (contrações que alteram a emissão luminosa) as quais permitem deduzir massa, taxa de rotação e de revolução, etc [OLIVEIRA et al. 1991; OLABARRIAGA 1993].

O estudo não é realizado sobre os dados brutos e sim sobre dados pré-processados pela aplicação de uma transformada de Fourier, que produz uma série de valores de intensidade vs. frequência, os quais podem ser plotados num gráfico onde os picos designam as frequências principais do sinal medido. O problema reside na exploração destes picos, já que existe uma infinidade de picos secundários e ruídos que precisam ser descartados. O processo de identificação de picos reais corresponde a um ciclo - detecção de pico, análise da vizinhança, determinação da classe do pico - até que todos os picos reais tenham sido identificados [OLABARRIAGA 1993]. Neste ponto, uma análise de outro tipo é feita: comparação de picos e verificação de padrões de repetição. A dificuldade encontrada diz respeito ao tamanho do conjunto de dados: anos de observação diária resultam em milhões de valores (=pontos num gráfico 2D), o que é praticamente impossível de representar visualmente.

Do ponto de vista de visualização, portanto, há necessidade de mapear os dados pré-processados para uma representação visual que permita procura de picos reais e de padrões entre picos; a representação visual proposta por OLABARRIAGA [1993] é um mapeamento da curva intensidade vs. frequência para uma espiral, onde ícones e cor representam, respectivamente, quantidade de picos e intensidade na frequência correspondente ao ponto sobre a espiral. A curva original correspondente a pelo menos parte do trecho mapeado para a espiral também deve poder ser recuperada e exibida.

Para uma análise completa, é necessário poder alterar as faixas de frequência que são mapeadas para a espiral, alterar o número de pontos a serem considerados na determinação de picos reais, realizar consultas aos valores de intensidade e frequência, assim como derivar outros valores a partir destes. Outras facilidades também necessárias são o descarte de dados irrelevantes, a possibilidade de guardar um registro das sessões

de análise já feitas e a possibilidade de anotação (marcação textual) nas imagens produzidas.

2.4 Exploração de resultados de análise de elementos finitos

Na área de projeto, a fase de análise da eficiência e usabilidade do objeto que está sendo projetado é extremamente importante. Nesta análise, o comportamento de variáveis importantes frente a determinadas condições de contorno é verificado de modo que seja possível detectar eventuais falhas de projeto. Em muitos casos, esta análise é realizada pelo método dos elementos finitos: o espaço-problema é discretizado de forma a tornar possível o cálculo de grandezas a serem analisadas. A análise de estruturas na engenharia civil, de comportamento de sistemas mecânicos e de sistemas elétricos são alguns exemplos onde é empregado este método de análise.

Tomaremos aqui como exemplo a aplicação do método de elementos finitos para o estudo de campos eletromagnéticos [SCHMIDT 1994]. Neste estudo, peças são modeladas geometricamente no espaço tridimensional, é gerada uma malha de elementos finitos tridimensionais e são calculados os valores nodais dos elementos. A partir dos valores nodais podem ser obtidos os valores de grandezas físicas como campo magnético, permeabilidade, indução magnética, etc. Tipicamente, a visualização é realizada em pós-processamento, ou seja, após a obtenção dos valores nodais, funções específicas são utilizadas para a obtenção dos valores das variáveis desejadas e para a representação visual dos resultados através de isolinhas na superfície ou em cortes do objeto. Entretanto, técnicas de visualização volumétrica podem ser utilizadas para a exibição de isosuperfícies internas, propiciando uma maior compreensão das grandezas, compreensão esta dificilmente obtida com uma série de cortes.

Em termos de visualização, esta classe de aplicações requer o mapeamento de grandezas escalares e vetoriais sobre a representação geométrica do objeto, sobre cortes selecionados ou volumetricamente. Entretanto, não se deve descartar a possibilidade de realizar gráficos mais simples representando a relação entre duas ou três variáveis calculadas.

É nas facilidades de exploração quantitativa, entretanto, que reside a maior contribuição para o trabalho de análise. SCHMIDT [1994] define ferramentas de exploração para obtenção de resultados numéricos: interativamente deve ser possível especificar um ponto, uma linha, uma face ou todo um elemento onde deve ser calculado o valor de uma variável a partir dos valores nodais. Obviamente, este cálculo é dependente da variável desejada e requer interpolação.

Uma outra necessidade detectada é a possibilidade de indicar a repetição do cálculo de elementos finitos para uma determinada região do objeto, numa resolução (número de elementos) maior. Isto representa realizar um processamento ainda pesado computacionalmente para uma estação de trabalho atual. Se forem propiciadas, entretanto, todas as facilidades de indicação da região do objeto na qual se quer realizar este "zoom" de cálculo, esta capacidade pode ser agregada facilmente.

2.5 Simulação interativa visual de objetos animados dinamicamente

Por simulação de objetos animados dinamicamente entende-se a produção de seqüências animadas onde o movimento dos objetos é determinado segundo leis da Dinâmica. O objetivo desta classe de modelagem é a produção de seqüências com o maior realismo possível. Objetos rígidos ou flexíveis [NEDEL 1993], articulados [MUSSE 1994] ou não-articulados [LEMOS 1993] são modelados geometricamente e lhes são atribuídos parâmetros como coeficiente de elasticidade, graus de liberdade das articulações, etc, conforme a classe.

A realização da simulação do movimento requer a especificação das forças que atuam sobre os objetos e do ponto onde elas atuam, além de eventuais restrições que determinam, por exemplo, pontos do objeto que permanecem imóveis até um determinado limiar de força aplicada. Em termos de controle dos experimentos, deve ser possível, ainda, especificar o tempo de simulação requerido (quantos segundos) e a forma como devem ser exibidos os resultados intermediários: só no final ou passo a passo. A determinação do movimento dos objetos pode ser realizada segundo dinâmica inversa, ou seja, são especificadas as posições destino e as forças são calculadas.

Em termos de visualização, os requisitos de tal classe de aplicação se resumem à exibição da seqüência animada de forma simplificada ou com técnicas de acabamento das imagens, durante a geração ou em pós-processamento. Esta seqüência deve poder ser reexibida, quadro a quadro, se necessário, característica esta que já pode ser encarada como uma forma de exploração dos resultados do experimento. Dependendo da classe de usuários da aplicação pode ser requerida a exibição de gráficos mostrando a evolução de parâmetros como velocidade e aceleração com o tempo. Eventualmente as informações de forças e velocidades poderiam ser representadas com ícones de vetores sobre a própria geometria do objeto.

2.6 Simulação interativa visual de sistemas discretos

Para exemplificar esta classe de sistemas, tomaremos o ambiente de simulação do sistema AMPLO [WAGNER 1991], que permite a simulação de sistemas digitais modelados através de linguagens de descrição de hardware. Neste ambiente, sistemas digitais são modelados como redes de blocos (denominados "agências"), que se comunicam através de sinais, os quais são variáveis tipificadas de acordo com o nível de descrição do sistema digital (nível lógico, nível de transferência entre registradores e nível de sistema). Numa rede, as agências podem estar descritas em níveis de abstração diferentes, isto é, agências podem representar outras redes e, assim, constituir uma hierarquia. O sistema, na sua representação mais exterior, é uma agência, com sinais (variáveis) de entrada e saída. O detalhamento dessa agência numa rede constitui a descrição simulável. Como a rede pode conter agências em níveis hierárquicos diferentes, no momento da simulação o sistema deve ser transformado numa rede de agências primitivas, sem hierarquia. Esta etapa é denominada construção do modelo de simulação, que contém, além das agências a serem efetivamente simuladas, uma relação de todas as variáveis que podem ser monitoradas pelo projetista (usuário).

A simulação de tal classe de sistemas requer [WAGNER 1991] a especificação das variáveis a serem monitoradas, seja através da sua exibição concomitante ao processamento, ou através da coleta de valores para análise em pós-processamento, a determinação do tempo de simulação, a indicação de condições de parada e a criação e vinculação de estímulos ao modelo (seqüência de eventos associados às variáveis de

entrada e, eventualmente, às variáveis internas às agências). Em termos de representação visual, tanto os estímulos ao modelo de simulação como os resultados da monitoração das variáveis podem ser exibidos em gráficos bidimensionais valor vs. tempo, demonstrando as formas de onda dos sinais. Consultas a valores de variáveis específicas também podem ser realizadas, mesmo que seu monitoramento não tenha sido solicitado.

Tomando-se outro exemplo de situação onde o objeto de estudo é um sistema composto por múltiplas entidades, por exemplo, um posto de pedágio, o objetivo é verificar o número ótimo de cabines em funcionamento nos diversos dias da semana, para evitar engarrafamentos. Deve-se estabelecer a distribuição de probabilidade que regula o número de carros a chegar no posto e a variação que ocorre tanto ao longo do dia como ao longo da semana. Conhecendo-se o tempo de atendimento, pode-se verificar o tempo de passagem dos carros em função do número de cabines funcionando. Este é o caso típico de problema onde é fundamental o registro de dados para uma análise estatística posterior. Representações visuais que podem ser empregadas neste exemplo são gráficos mostrando a relação tempo de permanência vs. número de cabines, em função da hora e do dia da semana. O tamanho das filas é também monitorado, a fim de ser minimizado. Nesta classe de problema, ainda, pode-se empregar uma abordagem hierárquica de animação como proposta por JOHNSON e POORTE [1988] e implementada por LINDSTAEDT [1993]. Nesta abordagem, conforme a fase do estudo de simulação, a representação visual do modelo pode ser mais ou menos simplificada. Na fase de depuração do programa de simulação, pode-se utilizar uma animação simples apenas pela mudança de cores numa representação esquemática do modelo. Para a apresentação final de resultados de experimentos, entretanto, ícones podem representar postos de atendimento e gráficos de barra podem representar as filas associadas aos postos, de modo que possa ser observada sua variação ao longo do tempo, numa forma mais compreensível.

Quanto aos aspectos de interação, é necessário prover facilidades de alteração de parâmetros do experimento, representações visuais das entidades e variáveis e consulta a variáveis eventualmente medidas mas não exibidas.

2.7 Similaridades e diferenças entre os casos

Dados multivariados populacionais representam atributos dos indivíduos das populações (ou genericamente dos conjuntos em estudo). Da mesma forma, entidades componentes de um sistema a ser simulado também são descritas por atributos. Estas duas classes não tem necessariamente uma representação gráfica associada às entidades em si, mas geralmente os atributos e as relações entre eles são exibidos na forma de gráficos diversos. O processamento aplicado nos dois casos é diferente: enquanto dados multivariados populacionais são estudados através de métodos estatísticos, sistemas de entidades são simulados utilizando técnicas de simulação discreta. Os resultados da simulação passam posteriormente por análise estatística. É importante, nestas duas situações-problema, a manutenção do histórico do processamento para comparação entre diferentes sessões de análise.

Dados multivariados espaciais e dados associados a objetos tridimensionais passam geralmente por algum processamento relativo à obtenção de informações derivadas, importantes para a análise. Enquanto os dados associados a objetos tridimensionais aqui estudados, passavam por um processo de análise de elementos finitos, os dados multivariados espaciais não envolviam este tipo de tratamento, mas isto é dependente do fenômeno amostrado. Esta última classe passa, eventualmente, por tratamento dos dados de entrada para a correção eventual de distorções do processo de amostragem. Nos estudos aqui relatados, dados tomográficos são consultados num caso e, no outro, campos eletromagnéticos sobre objetos tridimensionais são analisados. Em ambas as situações, a consulta a valores e a representação gráfica de dados selecionados constitui a atividade mais freqüente.

O estudo de caso envolvendo dados univariados revela também a necessidade de realização de pré-processamento e, posteriormente, representação gráfica e processo interativo de consulta. A seleção de dados a serem analisados na situação observada mostrou-se de fundamental importância, devido ao volume de dados originalmente coletados e processados. A comparação entre diferentes sessões de análise e entre diferentes subconjuntos de dados foi outra necessidade constatada, também em virtude do volume de dados.

As duas classes de simulação estudadas - simulação de sistemas discretos e simulação de objetos animados dinamicamente - revela outra classe de necessidades que é a de formas de controle da simulação, incluindo a especificação de parâmetros iniciais.

Em ambas, também, a coleta de informações na medida do avanço da simulação é importante para análise posterior. No caso de simulação de sistemas discretos esta coleta pode servir tanto à análise estatística como à demonstração do funcionamento do sistema. Na simulação de objetos animados dinamicamente, o caso estudado se restringia à geração de animações e, assim, a coleta de informações era, por si só, a atividade principal, pois permitia cumprir o objetivo estabelecido. Em termos de representação gráfica, a simulação por si só não tem grandes necessidades além de uma representação simplificada para o controle, porque a aplicação estudada não gerava as imagens finais.

3 Bases para a análise exploratória e a simulação interativa visual

Os aspectos relevantes ao embasamento da análise exploratória e da simulação interativa visual são explorados. São introduzidas classificações de objetos e atributos e de representações visuais; são distingüidos os objetivos e tarefas em ambos os casos. É proposta uma metodologia para a seleção de representações visuais com base em classes de objetos e atributos e nas tarefas a serem realizadas pelo usuário final.

3.1 Classificação dos objetos de interesse e seus atributos

A análise exploratória de dados e a simulação interativa visual são conjuntos de ferramentas ou procedimentos utilizados para aumentar a compreensão da estrutura ou comportamento de entidades, fenômenos ou sistemas diversos. Utilizar-se-á aqui o nome objeto para referir as entidades ou sistemas sob estudo, como forma de generalizar a apresentação das idéias. Os objetos podem ser classificados de acordo com a sua natureza, a qual, de certa forma, determina a classe principal de ferramentas de análise que são utilizadas no seu estudo.

Sistemas compostos por múltiplas entidades são conjuntos organizados de entidades inter-relacionadas como, por exemplo, redes de computadores e sistemas de manufatura. Bancos, supermercados, lojas, sistemas diversos, enfim, podem ser modelados desta forma, sendo a simulação discreta a ferramenta de análise natural neste caso.

Objetos manufaturáveis são aqueles freqüentemente representados por um modelo geométrico e informação analítica associada. Exemplos típicos são peças mecânicas e estruturas, objetos freqüentemente submetidos à análise pelo método de elementos finitos.

Entidades naturais e fenômenos são comumente representados por dados multivariados, sendo objeto de análise multivariada ou simulação contínua. Populações animais, vegetais ou humanas, dados meteorológicos, dados coletados de sensores em

Medicina são exemplos desta classe de entidades. Dependendo dos propósitos, eventualmente são extraídas informações geométricas destes dados.

Ambos os processos de análise exploratória e de simulação visual interativa são baseados em características ou propriedades dos objetos. Assim, torna-se imprescindível completar a classificação de objetos com a classificação de suas propriedades ou características. Utiliza-se o termo **atributo** para designar uma propriedade do objeto, seja esta uma característica de estrutura ou de comportamento. Considerando um sistema composto por múltiplas entidades, como um supermercado, por exemplo, o número de pontos de venda, o tempo médio entre chegadas de clientes e o tempo médio de atendimento por ponto de venda são atributos do sistema, fazendo parte do modelo. No caso de objetos manufaturáveis, um de seus atributos designa sua estrutura, através da descrição geométrica de suas partes e conexões; o tipo do material é outro atributo, possibilitando a análise/simulação de seu comportamento. Para entidades naturais ou fenômenos descritos por dados multivariados, cada variável amostrada é um atributo do objeto.

A classificação de objetos é variável de autor para autor, assim como o é a distinção de atributos dos objetos; ver, por exemplo, ROBERTSON [1991], BRODLIE et al. [1992] e TREINISH [1992]. A classificação apresentada na Tabela 3.1 foi desenvolvida com base nas idéias dos autores citados, tendo sido refinada a partir da apresentada em FREITAS e WAGNER [1993]. Atributos são classificados de acordo com a **natureza** ou **tipo das características** que representam, **natureza** ou **tipo dos valores** que eles podem assumir, **natureza** e **dimensão do domínio** onde estão definidos.

A **natureza de um atributo** indica a classe de informação que ele representa. Um atributo tipo **característica** corresponde, como o próprio nome diz, a uma característica isolada do objeto; não há uma função associada, a priori. Entretanto, eventualmente, a análise exploratória dos dados, através dos atributos deste tipo pode concluir a existência de relações entre eles. Num conjunto de dados sobre indivíduos de uma população animal de uma dada região, podemos ter diversos atributos do tipo **característica**: identificação da espécie, identificação do ponto de coleta, sexo, número de cromossomos, medidas de crânio. Um atributo de natureza **escalar** denota um conjunto de valores amostrados de uma relação definida no espaço do objeto; por exemplo, a temperatura e a pressão de um gás num recipiente.

Tabela 3.1 - Classificação dos atributos de objetos.

Critério	Classes	Significado
Natureza ou tipo do atributo	característica agregação escalar vetor	característica isolada, independente coleção de atributos grandeza escalar, amostrada de uma função grandeza vetorial, idem
Natureza ou tipo dos valores	alfanumérico numérico símbolo	valores de identificação valores ordinais, discretos ou contínuos sub-atributo
Natureza do domínio	discreto contínuo contínuo-discretizado	enumeração, conjunto finito ou infinito todos os pontos no espaço 1D, 2D, 3D, n D regiões no espaço 1D, 2D, 3D, n D
Dimensão do domínio	1D 2D 3D n D	dado definido no espaço 1D valor associado a espaço 2D valor associado à posição no espaço 3D valores no espaço n -dimensional

Um atributo do tipo **vetor** permite a representação de grandezas vetoriais, isto é, grandezas dotadas de magnitude, direção e sentido, como é o caso da velocidade de fluidos. Assim, enquanto um **escalar** é representado por um único valor, um **vetor** é representado por uma tripla de valores. Um atributo do tipo **agregação** representa uma coleção de **atributos**, ou seja, é um atributo composto por sub-atributos, devendo ser tratado desta forma. Por exemplo, a cor de um pixel é um atributo do tipo **agregação**, formado por três sub-atributos, cada um designando a quantidade de vermelho, verde e azul, considerando-se o modelo RGB de cor. Uma **agregação** pode, portanto, conter organizações de dados de outros tipos. Para um objeto da classe **sistemas de múltiplas entidades**, um atributo deste tipo pode conter a descrição da hierarquia ou rede de entidades componentes e suas conexões. Para um **objeto manufaturável**, o valor desta classe de atributo pode ser a descrição geométrica do objeto (polígonos descrevendo uma peça mecânica, por exemplo). Para **entidades** ou **fenômenos naturais**, este atributo pode conter a descrição da malha utilizada para especificar como os valores de atributos escalares ou vetoriais distribuem-se no domínio do objeto. Este é o caso de dados meteorológicos descrevendo temperatura, pressão, umidade, velocidade de vento sobre uma determinada região. Para representar uma série de dados no tempo, o objeto poderia ter um atributo agregação constituído por n outras agregações, cada uma correspondendo a uma observação num determinado instante de tempo.

A **natureza** ou **tipo dos valores** corresponde ao conceito de tipo de dado encontrado em linguagens de programação tradicionais, ou seja, se o atributo assume valores alfanuméricos ou numéricos, dentre uma enumeração finita ou infinita ou dentro dos reais ou, ainda, se o atributo for composto por valores simbólicos, representando sub-atributos. O **tipo dos valores** está intimamente relacionado com a **natureza do atributo**. No exemplo da população animal citado acima, o atributo espécie pode assumir valores que representam os nomes das espécies, dentre um conjunto de espécies conhecidas. A identificação do ponto de coleta pode ser um valor numérico, também dentro de uma enumeração, ou uma agregação {latitude, longitude}. Já atributos **escalares** ou **vetoriais** são representados por valores inteiros ou reais. No caso de um sistema de múltiplas entidades, um atributo "composição" será de natureza **agregação** e conterá valores simbólicos, indicando outras entidades, componentes do sistema.

A **natureza** e a **dimensão do domínio** identificam o espaço de definição dos atributos, ou seja, o espaço de definição do próprio objeto em estudo. Os atributos podem estar definidos num domínio discreto, restrito a um conjunto finito ou infinito de elementos, ou num domínio contínuo. Num domínio contínuo, os valores podem estar definidos para todos os pontos (continuamente) ou para regiões. Considerando uma região geográfica como o objeto em estudo, o atributo que designa a altura do terreno é definido para todos os pontos do domínio contínuo, enquanto outro que indica a densidade populacional em sub-áreas é definido por regiões, sobre o mesmo domínio contínuo. Outro exemplo corresponde ao número de indivíduos de cada espécie, que é definido numa enumeração (nomes das espécies); por outro lado, o número total de indivíduos é definido para toda a região. A **dimensão** de um atributo indica, na realidade, a dimensão do espaço onde o objeto está definido. Um atributo pode estar definido no espaço unidimensional (1D), bidimensional (2D), tridimensional (3D) ou n -dimensional (n D). O tamanho de uma fila é um atributo de dimensão 1D, pois uma fila, neste caso, é um objeto definido no espaço temporal 1D. Dados a respeito de uma região geográfica (2D) são de dimensão 2, enquanto dados associados a pontos no espaço 3D são de dimensão 3. Exemplos de dados n -dimensionais são obtidos de aplicações que geram dados multivariados, como, por exemplo, sensoriamento remoto e dados estatísticos sobre populações. Conforme a interpretação, a dimensão de um atributo permite representar dados espaciais, temporais, espectrais ou multidimensionais.

No anexo 1 encontram-se modelos dos objetos e atributos tomados dos casos mencionados no capítulo 2 e representados segundo a metodologia OMT [RUMBAUGH et al. 1991]. A notação utilizada nesta metodologia está descrita brevemente no capítulo 5.

3.2 Objetivos e tarefas em análise exploratória e simulação interativa visual

Tanto em análise exploratória visual como em simulação interativa visual, características e propriedades dos objetos ou dados em estudo são representados como atributos visuais de representações gráficas diversas. Desta forma, o usuário pode construir um modelo mental da situação, a partir do modelo visual, constatando relações, testando hipóteses, questionando os dados, ou seja, realizando todo um processo de investigação, onde a exploração visual e o questionamento são predominantes. A escolha do atributo visual que deve representar uma certa propriedade do objeto é dependente da informação que o usuário deseja extrair naquele contexto. Alguns autores distinguem necessidades de usuários de visualizações em termos de consultas que podem ser feitas, enquanto outros distinguem problemas num nível de abstração mais baixo. Considerando um conjunto de dados arbitrário, ROBERTSON [1991] distingue três tipos de consultas: qual o valor de um atributo num ponto, como é a distribuição local dos valores de um atributo e como é a distribuição global dos valores de um atributo. Já WEHREND e LEWIS [1990] distinguem classes de operações realizadas, considerando mais **ações**, do que propriamente **objetivos**: identificar, localizar, distinguir, categorizar, agrupar, distribuir, comparar, relacionar, associar e correlacionar.

Com base na observação de aplicações descritas na literatura, e em algumas aqui desenvolvidas, estabeleceu-se um conjunto de objetivos na análise exploratória e/ou simulação interativa visual num primeiro nível de abstração:

- análise de objetos num contexto;
- análise da estrutura de um objeto;
- análise das propriedades estáticas de um objeto;
- análise das propriedades dinâmicas de um objeto;
- análise comparativa de dois ou mais objetos;
- comunicação de informações.

A **análise** de objetos num contexto corresponde ao objetivo de um usuário que analisa o comportamento de um sistema de múltiplas entidades, onde componentes precisam ser distingüidos para a observação do comportamento global; por exemplo, num sistema de manufatura onde diferentes máquinas estão representadas na realização de suas tarefas interdependentes. Um exemplo diverso seria um cirurgião que analisa um órgão e precisa identificar uma região a ser retirada, a qual não é necessariamente um componente estrutural do órgão. A visualização da **estrutura** de um objeto complexo é comum em aplicações de análise de sistemas mecânicos, ou de órgãos reconstruídos a partir de dados obtidos com sensores médicos. A **análise de propriedades estáticas** de dados obtidos por instrumentos ou gerados por ferramentas de simulação ou análise é o objetivo de um usuário interessado em procurar padrões na massa de dados como, por exemplo, regiões de densidade ou temperatura constante. Cabe ressaltar que uma propriedade estática pode, dependendo da aplicação, identificar aspectos estruturais de objetos (p. ex., densidade de tecidos). No caso da **análise de propriedades dinâmicas** o objetivo é estudar a evolução de quaisquer propriedades que variem de acordo com algum parâmetro, inclusive tempo. Exemplos tradicionais são velocidade do vento numa região ou comportamento de filas num banco. A análise requer muitas vezes, o controle das propriedades estáticas e/ou dinâmicas de um objeto, o que inclui, além da exibição dinâmica, a possibilidade de alteração de parâmetros do objeto. A **análise comparativa** de dois ou mais objetos é característica da fase final ou decisória da análise ou simulação, quando são selecionados e comparados dois ou mais objetos para melhor compreensão tanto de estrutura e propriedades estáticas como de comportamento. Esta mesma análise se aplica à comparação de diferentes situações do mesmo objeto. A **comunicação de informações** é objetivo final da maioria dos casos pois tipicamente, dados coletados ou resultados de análise ou simulação devem ser comunicados numa forma compreensível a pessoas não necessariamente envolvidas com os processos de aquisição das informações e análise das mesmas.

Estes objetivos são gerais e para atingí-los é necessária a realização de **operações** ou **tarefas**, algumas gerais, outras específicas para cada caso. A par das operações identificadas por WEHREND e LEWIS [1990] e listadas anteriormente, outros autores referem-se a ações destinadas a atender os objetivos da análise exploratória visual e/ou da simulação interativa visual. SPRINGMEYER et al. [1992], por exemplo, caracterizam o processo de análise de dados científicos e estabelecem ações ou tarefas que são desempenhadas por cientistas. Sob a denominação de **interação com**

representações, são identificadas as ações de (1) geração de representações visuais; (2) exame das representações; (3) orientação da representação visual; (4) consulta a valores numéricos; (5) comparação; e (6) classificação. Já numa outra fase do processo de análise, a fase de **processamento**, há três outras tarefas: (1) estimativas e transformações; (2) derivação de novas condições; e (3) geração de estatísticas. Em outras duas fases do processo de análise, **manipulação de dados e expressão de idéias**, outras ações específicas são realizadas, a saber: (1) navegação; (2) gerência de dados; e (3) seleção de subconjuntos de dados, a título de **manipulação**; (1) registro e (2) descrição, como ações de **expressão de idéias**. BESHERS e FEINER [1993], tratando exclusivamente da exploração de dados multidimensionais, resumem todas as tarefas de visualização a uma combinação de três ações básicas: explorar (quando a relação é desconhecida), pesquisar (quando o objetivo é buscar uma característica particular no conjunto de dados) e comparar (quando a meta é analisar de forma comparativa dois subconjuntos da mesma relação, ou de relações distintas). Em se tratando de simulação interativa visual, ROOKS [1991] lista quatro tarefas essenciais a um sistema de VIS: (1) intervenção, que corresponde a técnicas de iniciar a interação com um modelo; (2) inspeção de dados relevantes; (3) especificação de parâmetros do modelo; (4) visualização de dados do modelo.

Todos os autores citados acima distinguem análise qualitativa de análise quantitativa; além disso, as tarefas associadas a cada uma também são distintas, assim como o são, em parte, as ferramentas interativas. Citando ROOKS [1991], "sistemas VIS devem auxiliar a análise qualitativa facilitando a compreensão da dinâmica do sistema e devem suportar a verificação quantitativa dessa compreensão". Na mesma linha de pensamento, SPRINGMEYER et al. [1992] classificam como "qualitativo" o meio de identificar características nos dados; "quantitativo" é utilizado para designar meios de avaliar simbólica ou numericamente características previamente identificadas.

Partindo dos resultados obtidos pelos autores mencionados e confrontando-os com os objetivos estabelecidos, identificamos as seguintes tarefas básicas que, combinadas, permitem o atendimento das necessidades dos usuários nos processos de análise exploratória visual e simulação interativa visual (vide Tabela 3.2):

- navegação pelo universo de objetos;
- seleção de um conjunto de objetos ou de parte de um objeto;
- consulta a atributos de objetos;

- preparação de processamento (modificação de atributos; inicialização de parâmetros);
- processamento (simulação, cálculos, monitoramento, controle);
- anotação de textos e armazenamento de dados.

Tabela 3.2 - Relação das tarefas com os objetivos de análise exploratória e simulação interativa visual

Objetivos	Tarefas
Análise de objetos num contexto	Navegação, Seleção, Consulta Anotação, Armazenamento
Análise da estrutura	Navegação, Seleção, Consulta Anotação, Armazenamento
Análise das propriedades estáticas	Navegação, Seleção, Consulta Preparação, Processamento Anotação, Armazenamento
Análise das propriedades dinâmicas	Navegação, Seleção, Consulta Preparação, Processamento Anotação, Armazenamento
Análise comparativa	Navegação, Seleção, Consulta Preparação, Processamento Anotação, Armazenamento
Comunicação	Anotação, Armazenamento

A primeira tarefa corresponde a ações de navegação, seja por um volume de dados, coletado através de sensores ou gerado por algum processamento, seja pelo domínio de um relação descrita formalmente ou, ainda, por um modelo representado em alguma linguagem visual de simulação. A tarefa de navegação é diferente para cada classe de objetos, dentre as descritas na seção 3.1, sendo, portanto, efetivada através de diferentes recursos interativos (vide capítulo 4). O requerimento básico neste tipo de tarefa é a possibilidade de observação dos objetos, para efeitos de localização e determinação de operações posteriores. Esta tarefa é realizada praticamente na maioria dos momentos pois serve de posicionamento do usuário para os procedimentos específicos de análise.

A tarefa de seleção de um conjunto de entidades ou de parte da massa de dados, conhecida como *data culling* [SPRINGMEYER et al. 1992], visa diminuir o espaço de

exploração e/ou de consulta, através da redução do número de objetos em análise direta. Isto quer dizer que o conjunto total de objetos é submetido a um critério seletivo e dele são extraídos apenas aqueles objetos que satisfazem as restrições impostas pelo usuário, para uma dada situação particular de análise. Este critério seletivo pode ser dado por uma condição ou pode ser a indicação direta por parte do usuário. Novamente neste caso, os recursos interativos para esta tarefa diferem de acordo com a natureza das entidades envolvidas. A tarefa de seleção é utilizada para propiciar o atendimento de todos os objetivos listados para qualquer classe de objetos mencionada: a especificação de um determinado intervalo de densidade para a visualização de determinada região num volume de dados tomográficos ou a indicação de atributos a serem monitorados numa simulação são exemplos de critérios de seleção.

A tarefa de consulta a atributos de objetos enfatiza os aspectos quantitativos da exploração visual, isto é, engloba as ações de questionamento por valores de atributos específicos dos objetos. Esta tarefa requer recursos interativos para indicação do objeto, especificação do atributo a ser consultado e possibilidade de indicação da forma de representação dessa informação: gráfica ou numérica. Eventualmente, o atributo que está sendo solicitado não existe explicitamente no modelo e deve ser derivado a partir daqueles existentes; facilidades de cálculo para esta situação são essenciais. A tarefa de consulta é utilizada para realizar todos os objetivos listados.

Já a tarefa de preparação de processamento corresponde a ações de inicialização de parâmetros diversos para a realização de procedimentos de análise ou simulação das propriedades dos objetos. A título de exemplo, pode-se citar a alteração da distribuição de probabilidade utilizada para gerar entidades de uma determinada classe num sistema sendo simulado ou, no caso de simulação de movimento, por exemplo, a alteração de parâmetros iniciais (posição, velocidade inicial, trajetória, forças, etc. dependendo do modelo físico sendo utilizado).

A tarefa de processamento corresponde à ativação de experimentos de simulação ou procedimentos específicos de cálculo, tais como análise de elementos finitos, análise estatística, etc. O usuário pode desejar monitorar ou controlar o processamento; isto é particularmente empregado na análise de propriedades dinâmicas. Neste caso são requeridos recursos para a determinação de pontos de observação (pontos de parada do processamento), a exibição dinâmica do estado dos objetos e a interrupção momentânea do processamento para o ajuste de algum parâmetro.

Finalmente, as tarefas de anotação e armazenamento dizem respeito à preparação e ao armazenamento de material para análise posterior ou comunicação. A tarefa de anotação permite a composição de textos sobre as representações visuais, permitindo a explanação de resultados ali apresentados. Já a tarefa de armazenamento permite a coleta de dados ou gravação de representações visuais para posterior recuperação ou produção de *hardcopy*.

3.3 Classificação de representações visuais

Tanto na análise exploratória de dados como na simulação interativa visual o principal canal de comunicação utilizado é a visão. A eficiência das representações visuais, portanto, influencia a realização deste processo de exploração, já que os aspectos de percepção devem ser considerados. Com o intuito de sistematizar a escolha de abstrações gráficas, foi elaborada uma classificação de representações visuais, baseada em idéias reunidas em estudos anteriores [FREITAS 1990; FREITAS 1992a; FREITAS 1992b]. Tais estudos abordaram temas como as técnicas de Computação Gráfica utilizadas em aplicações de simulação, linguagens visuais de programação, visualização de estruturas de dados e animação de algoritmos, além de linguagens e ambientes de simulação visual. Todas estas ferramentas e aplicações se utilizam de abstrações visuais pré-definidas como forma de comunicação com o usuário.

A maioria dos modelos de simulação são programas escritos em linguagens de programação convencionais (*Fortran*, por exemplo) ou em linguagens de simulação de propósitos gerais (*SIMAN*, *GPSS*, etc.) e existe, portanto, uma relação entre modelos visuais de simulação e linguagens visuais de programação. Ainda, os resultados de experimentos realizados com modelos de simulação ou de análises realizadas sobre dados científicos coletados são representados em estruturas de dados, advindo daí uma relação entre visualização de programas e de estruturas de dados e a visualização de experimentos e de dados em geral.

Programas convencionais podem ser visualmente representados por diferentes tipos de diagramas de blocos, redes de ícones e formulários (tabelas). Já em modelos de simulação as duas primeiras classes são freqüentemente utilizadas: diagramas de blocos

em *GPSS/PC* e em *SIMAN*, e redes de ícones em *Slamsystem*, *Slam II* e *Witness*. Formulários são usados para especificar experimentos em *SIMAN*. Na visualização de programas as abordagens diagramática e icônica são utilizadas para demonstrar o fluxo de execução; ícones são utilizados na representação de estruturas de dados. Sistemas de simulação comerciais que incorporam apresentação de resultados de experimentos tanto em tempo real como em pós-processamento utilizam gráficos para os dados medidos e diagramas de blocos ou redes de ícones para o modelo propriamente dito. É o caso de *Witness*, *SIMAN/CINEMA*, *Simscrip II.5* e *Slam/Tess*.

Modelos de fenômenos complexos e dados científicos em geral não seguem, entretanto, os paradigmas mencionados acima. Para estes casos, há sugestão de alguns autores tanto no uso de um paradigma único como no uso de representações variadas cujo critério de escolha muitas vezes não é mencionado. ROBERTSON [1991] sugere o uso de uma cena natural como abstração básica para a exibição de dados científicos em geral. A abordagem mapeia propriedades dos objetos (dados) para atributos físicos de superfícies, dependendo dos objetivos do cientista. Já ELLSON e COX [1988] utilizam a abordagem icônica, onde atributos dos dados são mapeados para características visuais dos ícones. Nestes dois trabalhos não existe, portanto, uma classificação de abstrações gráficas; duas classes específicas foram escolhidas como paradigmas de visualização: mapa sobre representação de superfície e ícones 3D. Uma classificação aparece em LOHSE et al. [1990], mas não inclui abstrações como imagens de fenômenos complexos como, por exemplo, o traçado de partículas que representa a trajetória de um fluxo sendo simulado. Estes autores identificaram cinco classes de representações visuais, a partir do agrupamento de representações escolhidas de livros populares e submetidas à classificação por diferentes pessoas; as cinco classes são gráficos e tabelas, mapas, diagramas, redes e ícones. A classe "mapa" reúne representações visuais baseadas em linhas, padrões, cores, superfícies, partículas e ícones. Exemplos típicos destas representações são linhas e superfícies de isovalores, trajetória de partículas, ícones sobre malha de pontos, linhas de trajetória, enfim, as representações visuais obtidas diretamente sobre o espaço-domínio dos dados.

A classificação proposta aqui (Tabela 3.3) baseia-se na observação de várias aplicações e na análise das vantagens e desvantagens das propostas acima citadas. A classificação contempla a informação que é transmitida pela representação no que se refere às necessidades típicas em análise exploratória de dados e simulação interativa visual.

Convém salientar que a última representação constante nesta tabela corresponde a uma seqüência de qualquer uma das anteriores. Comumente a apresentação de uma seqüência de imagens estáticas ou de uma seqüência animada visa a percepção da evolução do comportamento de um objeto ao longo do tempo. Entretanto, a animação pode ser usada para salientar características presentes nos objetos [GERSHON 1992], se animação for tomada como qualquer alteração visual na imagem. Nesse trabalho, GERSHON [1992] experimenta animação para melhorar a compreensão de detalhes em conjuntos de dados; na comparação de dois atributos de uma região, por exemplo, as representações visuais de ambos são mapas de cores e sua exibição é alternada para salientar a relação existente.

Tabela 3.3 - Classes de representações visuais identificadas

Classe de representação visual	Informação possibilitada
Ícones	observação de um objeto no contexto
Gráficos e tabelas	propriedades estáticas de objetos propriedades dinâmicas de objetos
Diagramas e redes	estrutura de objetos comportamento de um objeto num instante t
Modelos geométricos	estrutura de objetos
Mapas	propriedades estáticas de objetos comportamento de um objeto num instante t
Seqüências	evolução de propriedades dinâmicas de objeto realce de características de um objeto

3.4 Metodologia para seleção de representações visuais

A apresentação de objetos e atributos destes objetos para efeito de comunicação de dados e resultados só é eficiente se as representações visuais escolhidas assim o forem. Isto significa que as representações visuais devem ser escolhidas (1) com base nas informações a serem transmitidas e/ou realçadas e (2) com base nos objetivos do usuário

e nas tarefas que ele executará (sobre os objetos) no decorrer do processo de análise ou simulação.

A metodologia proposta aqui não se baseia num único paradigma escolhido como representação visual de base, como é o caso de ROBERTSON [1991]. É, por outro lado, baseada nos três aspectos discutidos acima: objetos e atributos, objetivos e tarefas, classes de representações visuais. A metodologia consiste de cinco fases expostas a seguir.

- **Adquirir conhecimento a respeito dos objetos a serem analisados ou simulados.**

De acordo com o discutido na seção 3.1, os objetos devem ser identificados, assim como seus atributos. Deve ser feito um enquadramento nas classes estabelecidas (sistemas de múltiplas entidades, objetos manufaturáveis e entidades e fenômenos naturais).

- **Determinar os objetivos e tarefas na análise ou simulação.**

Aqui consideram-se os aspectos colocados na seção 3.2, ou seja, identificação de um objeto no contexto; análise da estrutura de um objeto; análise das propriedades estáticas de um objeto; análise das propriedades dinâmicas de um objeto; acompanhamento do comportamento de um objeto; controle do comportamento de um objeto; e comparação de dois ou mais objetos. Devem ser levantadas e respondidas questões como: quais objetos necessitam de identificação; quais os critérios para a identificação de um objeto dentre vários; para que objetos serão efetuadas questões (e quais são elas) que envolvem sua estrutura, suas propriedades estáticas, suas propriedades dinâmicas; quais as questões de ordem qualitativa e quantitativa, envolvendo atributos de objetos; quando será necessário acompanhar o comportamento de um objeto; quais objetos necessitam ter seu comportamento controlado; em que situação é necessária e quais os critérios para a comparação de dois ou mais objetos.

- **Refinar as questões levantadas.**

Para cada tarefa ou questão enumerada na fase anterior, determinar se o resultado deve ser apresentado visual ou numericamente. Isto é necessário porque, freqüentemente, a análise quantitativa sucede a análise qualitativa para testar hipóteses levantadas com base na observação de uma representação visual. Para as questões que implicam processamento, determinar que atributos são utilizados e que outros parâmetros são necessários para sua execução.

- **Mapear {objeto, questão} ou {atributo, questão} para uma (ou mais de uma) representação visual.**

A Figura 3.1 mapeia classes de objetos e objetivos para representações visuais sugeridas. Tomando como exemplos de cada classe de objetos: uma rede de computadores, um sistema mecânico, uma população humana e um volume de dados meteorológicos, pode-se interpretar mais claramente o mapeamento apresentado. Assim, tomando os objetivos listados anteriormente:

Objetivo	Objeto		
	sistema de múltiplas-entidades	objeto manufaturável	entidade ou fenômeno natural
observação de objetos	<i>ícone</i>	<i>modelo geométrico</i>	<i>ícone, gráfico, modelo geométrico</i>
análise da estrutura	<i>diagrama, rede</i>	<i>modelo geométrico</i>	<i>ícone, mapa, tabela</i>
análise de propr. estáticas	<i>características de ícone, gráfico, tabela</i>	<i>mapa, tabela, gráfico</i>	<i>mapa, tabela, gráfico</i>
análise de propr. dinâmicas	<i>seqüência de ícones, gráficos, tabelas</i>	<i>seqüência de mapas, gráficos, tabelas</i>	<i>seqüência de mapas, gráficos, tabelas</i>
controle de propriedades	<i>características de ícone, gráfico, tabela</i>	<i>mapa, tabela, gráfico</i>	<i>mapa, tabela, gráfico</i>
análise comparativa de objetos	<i>representações escolhidas</i>	<i>representações escolhidas</i>	<i>representações escolhidas</i>
comunicação	<i>representações escolhidas</i>	<i>representações escolhidas</i>	<i>representações escolhidas</i>

Figura 3.1 - Representações visuais adequadas em função de {objeto, objetivo}

- 1 Para a observação de objetos num contexto:
 - componentes num rede de computadores são representados através de um ícone para cada componente;
 - peças num sistema mecânico são descritas através dos modelos geométricos das peças;
 - um fenômeno meteorológico pode ser identificado através de padrões no gráfico de evolução do fenômeno ou, ainda, ícones sobre o domínio do fenômeno.

- 2 Para a análise da estrutura de um objeto:
 - uma rede de computadores é representada através de um diagrama ou rede de ícones;
 - peças num sistema mecânico são distingüidas pela exibição dos modelos geométricos correspondentes.

- 3 Para a análise de propriedades estáticas de um objeto:
 - informações coletadas a respeito da rede de computadores são mapeadas para gráficos diversos (de linha, de barras, histogramas, tridimensionais) ou características dos ícones;
 - informações associadas a uma peça mecânica podem ser representadas por gráficos ou tabelas, ou como um mapa de cores sobre a geometria da peça;
 - regiões de um fenômeno meteorológico são representadas por mapa de cores, mapa de linhas, mapa com ícones ou de superfícies.

- 4 Para a análise das propriedades dinâmicas de um objeto, nos exemplos acima, podem ser usadas seqüências das representações citadas para o caso estático.

- 5 Para o controle das propriedades de um objeto, são usadas as mesmas representações:
 - parâmetros da rede de computadores podem ser modificados;
 - características (material, por exemplo) da peça mecânica podem ser alteradas;
 - no caso dos dados meteorológicos, no entanto, não há sentido em alterar valores, a menos que haja um modelo do fenômeno cujas condições devem ser inicializadas a partir das observadas na natureza.

- 6 A análise comparativa de dois ou mais objetos (ou de dois ou mais estados do mesmo objeto) é, obviamente, baseada na comparação de propriedades dos objetos envolvidos. Neste caso, representações escolhidas de acordo com as propriedades devem ser exibidas conjuntamente.
- 7 A comunicação de dados coletados ou resultados de processamento é realizada a partir de representações escolhidas por ocasião das análises efetuadas nos diversos atributos dos objetos.

Considerando a classificação dos atributos exposta anteriormente e os dois objetivos relacionados com atributos (análise de estrutura, propriedades estáticas e dinâmicas), devemos também considerar as representações visuais cabíveis ou possíveis em relação às classes de atributos. A Figura 3.2 apresenta as representações visuais adequadas para os casos possíveis de natureza e dimensão do domínio do atributo. Quando for necessário, representações visuais devem ser compostas a partir destas "básicas".

Natureza do atributo	Dimensão do domínio			
	1D	2D	3D	nD
característica	-----diagrama de dispersão ou característica de ícone-----			
escalar	gráfico de linhas, diagrama de barras, histograma	mapa de isolinhas, mapa de ícones, gráfico 3D	isosuperfície, mapa volumétrico	mapa de ícones, composição
vetorial		mapa de ícones, linhas, partículas	mapa de ícones, linhas, partículas	
agregação	--- diagrama, mapa de ícones, dependendo da natureza dos componentes-----			

Figura 3.2 - Representações visuais adequadas em função da natureza e da dimensão do atributo.

- **Estabelecer a composição (combinação) válida em relação às representações visuais escolhidas.**

A composição de representações visuais é utilizada quando devem ser representados vários atributos de uma entidade ou, eventualmente, atributos análogos de entidades diferentes. A composição de representações visuais deve ser determinada para evitar ambigüidades; algumas combinações podem não fazer sentido e, mesmo, podem prejudicar o entendimento da situação. Como exemplo de combinação possível pode-se citar uma grandeza escalar (temperatura) e uma vetorial (vento) sobre uma região. A temperatura pode ser representada por cores, com transparência, e o vento pode ser representado por ícones de setas, onde a espessura pode significar a magnitude da velocidade e a própria seta indica a direção.

O estabelecimento desta metodologia justifica-se tanto do ponto de vista do usuário (cientista) de um sistema de análise exploratória ou simulação interativa visual, como das necessidades de um projetista de uma aplicação específica enquadrada nessas classes. O usuário-cientista encontra nesta metodologia uma diretriz para o uso conjugado de ferramentas de análise e representação visual. Cabe lembrar que as fases 1, 2 e 3 correspondem a passos necessários a qualquer estudo. Nas fases 4 e 5, este usuário escolhe (sob orientação da metodologia) formas alternativas de representar visualmente seu modelo. O usuário-cientista está interessado em interpretar seus dados ou observar seu modelo segundo diferentes critérios e pode, então, utilizar a metodologia em conjunto com um sistema que lhe forneça possibilidades de selecionar dinamicamente a representação visual para seu problema e realizar as diferentes tarefas de análise ou simulação sobre as representações visuais escolhidas. Já o usuário-projetista encontra nesta metodologia aspectos essenciais para a fase de análise do desenvolvimento de sua aplicação. Determinadas representações visuais poderão ser eleitas como base da aplicação e o usuário desta realizará seu trabalho de forma mais dirigida e, eventualmente, mais eficiente.

O anexo 2 apresenta a aplicação da metodologia em duas situações: para dados multivariados populacionais e para o desenvolvimento do EFVis [SCHMIDT 1994]. Reproduz, desta forma, os enfoques do usuário de análise exploratória visual e do projetista de uma aplicação específica.

3.5 Considerações finais

A classificação das entidades (objetos) em três classes distintas e das propriedades (atributos) em consonância com critérios que contemplem natureza das propriedades, tipo dos valores, natureza e dimensão do domínio de definição permitem o tratamento unificado de problemas de diversas áreas. Tanto objetos como atributos podem ser representados visualmente e estas representações devem ser escolhidas de acordo com as informações que é necessário extrair dos dados a cada passo do processo de análise de dados científicos. A metodologia proposta é uma primeira abordagem ao assunto, tendo sido definida com base na observação tanto do descrito na literatura como das aplicações utilizadas nos estudos de caso.

Representações visuais são utilizadas nas tarefas realizadas para atingir os objetivos do processo de análise de dados científicos. Os objetivos foram claramente identificados, assim como as tarefas, que correspondem a ações utilizando ferramentas adequadas. Cada ferramenta reúne recursos computacionais de interação e de processamento para atuar sobre as diferentes classes de entidades e de atributos.

4 Recursos visuais e interativos para análise exploratória e simulação

Os processos de análise exploratória e simulação interativa visual são baseados em recursos (visuais interativos) reunidos em ferramentas. Este capítulo apresenta estas ferramentas de forma classificada e de acordo com as tarefas que suportam, as quais foram previamente identificadas no capítulo anterior.

4.1 Classes de ferramentas

Assim como tarefas são agrupadas para suportar a realização dos objetivos numa sessão de análise ou de simulação, as ferramentas são coleções de recursos visuais interativos que suportam a realização de tarefas. Ferramentas podem ser classificadas como de **mapeamento, exploração, preparação e controle de processamento e registro**.

As ferramentas de **mapeamento** correspondem às facilidades de indicação de representações visuais para objetos e seus atributos. As ferramentas de **exploração** correspondem a facilidades de suporte à navegação, seleção e consulta dos objetos em estudo. As ferramentas de **preparação e controle de processamento** contemplam as facilidades de análise das informações existentes no modelo, através de métodos numéricos, estatísticos e de simulação. Além da geração de resultados a partir de um processamento específico, englobam a preparação do processamento, sua monitoração e controle. A preparação do processamento diz respeito ao estabelecimento das condições iniciais dos parâmetros; enquanto monitoração e controle do processamento implicam no acompanhamento do processamento, através da observação de atributos das entidades e, eventualmente, intervenção no mesmo. Já as ferramentas da classe **registro** correspondem às facilidades de anotação de textos em representações visuais e armazenamento de resultados numéricos e gráficos, para posterior recuperação ou reprodução em outros meios.

As diversas ferramentas são definidas, em sua maioria, como um **procedimento interativo sobre uma representação visual**. Para uma melhor compreensão da constituição das diversas ferramentas, elas são identificadas através da apresentação dos recursos visuais interativos necessários às diversas tarefas da análise exploratória visual e da simulação visual interativa. Ferramentas compartilham recursos; por exemplo, um cursor é a base da navegação mas também o é da seleção, como veremos adiante. São apresentados inicialmente os recursos em termos de representações visuais uma vez que estes são utilizados por todas as ferramentas.

4.2 Representação visual de entidades e atributos

Além da classificação de representações visuais, o capítulo 3 apresentou um mapeamento {objeto - representação} dependente dos objetivos do usuário da metodologia. Assim, os recursos necessários para que se possa efetuar este mapeamento correspondem a técnicas de visualização que implementem estas representações visuais. Observando a Figura 3.1, extraem-se dela diversas representações visuais, que são discutidas a seguir.

Ícones visam identificar um elemento (objeto) num determinado contexto e, se necessário, explicitar, através de propriedades visuais do ícone, atributos do objeto sob análise. A utilização de ícones na forma *bimap* é limitada do ponto de vista dos atributos que podem ser associados a suas propriedades (fig. 4.1a). Mais poderosos são os ícones descritos no plano ou no espaço, como entidades geométricas com propriedades diversas para os quais possam ser mapeados os atributos dos objetos do modelo (fig. 4.1b e 4.1c). Para que ícones possam ser construídos a partir de outros, de forma a representar objetos mais complexos, ou mais informações de um mesmo objeto, é necessário o uso de representação hierárquica na sua estruturação.

As técnicas para exibição de ícones são técnicas clássicas comumente utilizadas para a representação de objetos 2D e 3D. Do ponto de vista de modelagem dos ícones, eles são objetos polimórficos (ponto, linha, polígono, esfera, cubo, poliedro, etc.) com métodos próprios de exibição e atributos como posição, dimensões, cor, textura e, eventualmente, texto associado. Forma, dimensões, cor, textura e texto servem de base para o mapeamento de atributos.

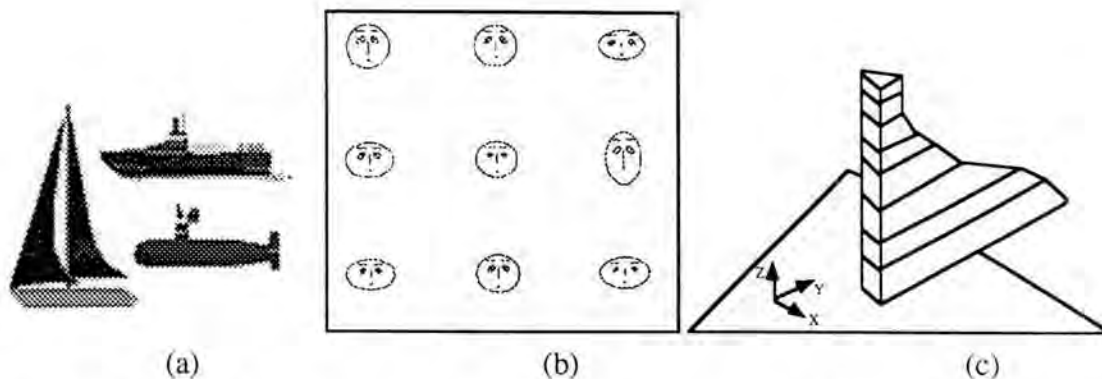


Figura 4.1 - (a) Ícones na forma *bitmap* obtidos com *SIMGRAPHICS*; (b) subconjunto das faces de Chernoff, utilizadas para representar dados multivariados; (c) ícone 3D cujo tamanho e forma representam magnitude de velocidade e direção de fluxo; cores no ícone e na base representam temperatura e pressão de um fluido [ELLSON e COX 1988].

Os **diagramas** (fig. 4.2) são construídos a partir de símbolos (ícones) conectados, ou seja, um diagrama é modelado como um objeto composto de outros objetos primitivos e, portanto, utiliza métodos de exibição herdados destes. **Diagramas de blocos** e **redes** são também modelados da mesma forma, considerando ícones para blocos e para componentes primitivos da rede. Os atributos das entidades do modelo são, então, mapeados para propriedades dos ícones. Como um diagrama é basicamente utilizado na representação da estrutura do sistema modelado, atributos globais deste sistema devem ser representados visualmente de outra forma (gráfico ou ícone especial).

Gráficos são as formas visuais mais utilizadas para representar e permitir a análise da variação de um atributo no tempo, no espaço, ou em relação a outro(s) atributo(s). Os tipos mais comuns são gráficos de pontos (de dispersão, *scatter plots*), gráficos de linhas, gráficos de barras, histogramas de frequência, todos bi- ou tridimensionais (fig. 4.3).

Além do tipo, cada gráfico tem propriedades visuais como apresentação dos eixos, texto associado ao eixo, apresentação da entidade geométrica básica do gráfico (ícone associado ao ponto, tipo e cor da linha, tamanho e cor da barra, etc). Neste sentido, os gráficos correspondem a uma classe de objetos, com propriedades alteráveis dependendo dos atributos dos dados que se queira representar. **Tabelas** são utilizadas quando há necessidade de relacionar os dados na sua forma original; as tabelas podem apresentar todos ou parte dos atributos dos objetos.

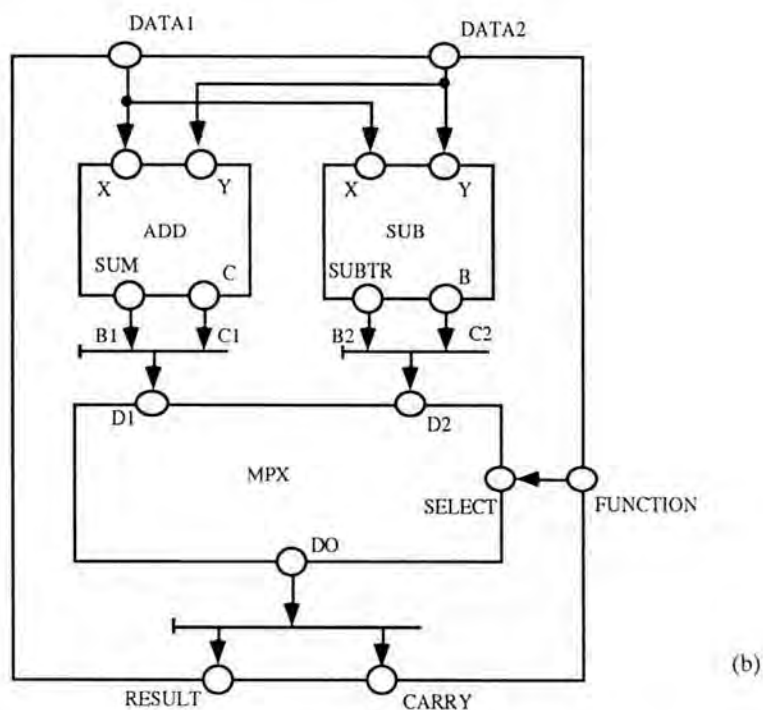
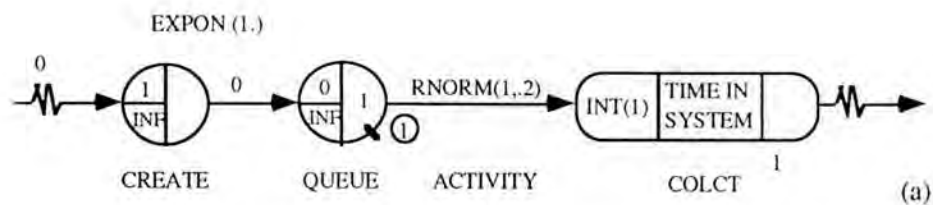


Figura 4.2 - (a) Diagrama representando um modelo de fila na linguagem *SLAM II* [O'REILLY 1991]; (b) diagrama de uma unidade aritmética simples na linguagem *REDES* [GOLENDZINER et al. 1990].

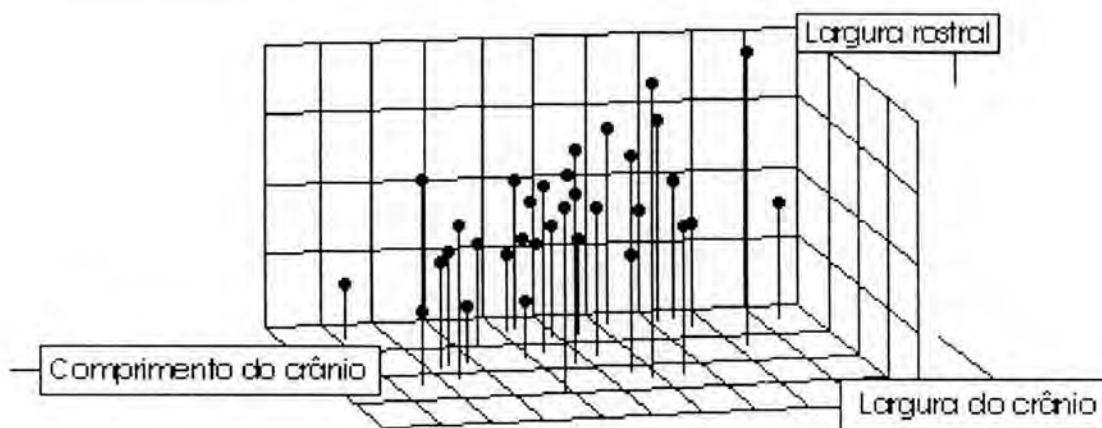


Figura 4.3 - Gráfico tipo diagrama de dispersão com dados de medidas de crânio de roedores. Cada ponto representa as coordenadas de um espécimen no espaço tridimensional definido pelas variáveis.

Os **modelos geométricos** visam representar mais realisticamente os objetos em estudo, mostrando forma, tamanho e estrutura. A exibição se dá em função da técnica de modelagem utilizada: linhas, faces poligonais, sólidos componentes, superfícies, etc. Visualmente, o modelo geométrico de um objeto pode variar desde a forma aramada (fig. 4.4) até formas fotorealistas. Além de representar a estrutura dos objetos, os modelos geométricos servem de suporte para o mapeamento de diferentes atributos, com a utilização de outras formas de representação visual .

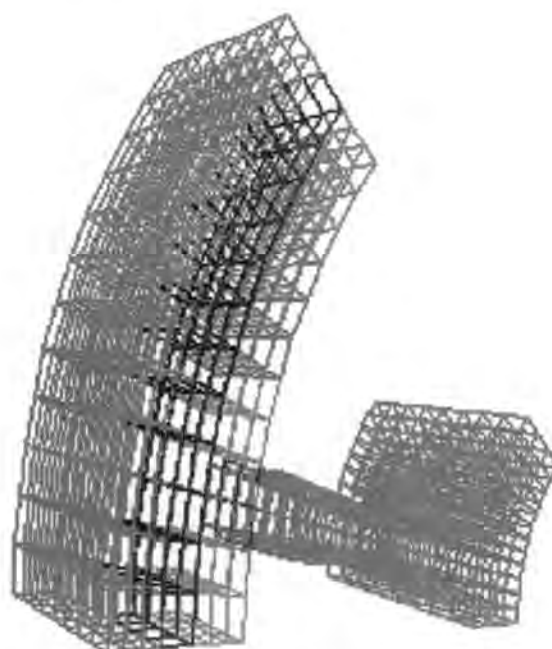


Figura 4.4 - Modelo geométrico de uma peça (imagem cedida por SCHMIDT [1994]).

Mapas designam todas as representações visuais resultantes do mapeamento dos atributos dos objetos para atributos visuais de um plano, de um volume ou da geometria dos objetos. Linhas, cores, texturas e ícones podem ser utilizados sobre a representação geométrica para indicar valores diversos, associados a vértices ou faces (fig. 4.5).

No caso de objetos ou fenômenos sem descrição geométrica (volumes de dados estruturados ou não), o mapeamento dos valores em questão para um plano ou volume exige interpolação ou exibição direta (volumétrica). Os mapeamentos de valores de atributos podem ser feitos com: preenchimento de cores (as chamadas cores falsas), padrões ou texturas; traçado de linhas de contorno (isovalores) que delimitam regiões numa superfície; exibição de superfícies de isovalor, que limitam regiões no espaço com

determinado valor de atributo; traçado de linhas de trajetória, em se tratando de grandezas vetoriais; e mapeamento de ícones sobre o plano ou espaço, com diferentes atributos representados como características dos ícones.

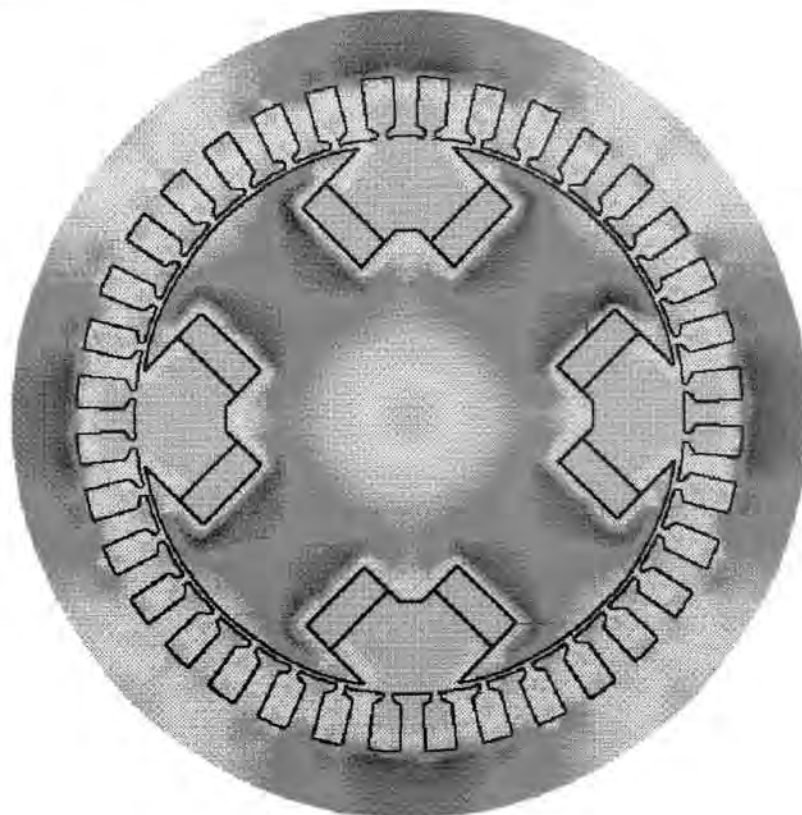


Figura 4.5 - Mapa de cores falsas sobre geometria, representando a grandeza "indução" em um **motor** (imagem cedida pelo GRUCAD, UFSC, para SCHMIDT [1994]).

Embora classicamente a dinâmica de objetos possa ser representada através de gráficos que mostram a evolução de valores de atributos ao longo do tempo, quando se tem estes valores mapeados para um tipo de representação visual, mostrar a representação visual variando ao longo do tempo pode ser facilmente realizado com a exibição de uma **seqüência** de mapeamentos. Esta seqüência pode ser apresentada de forma estática, isto é, várias "fotos" uma ao lado da outra, ou, então, de forma dinâmica, utilizando animação. Na realidade, a seqüência, em si, não é uma representação visual, mas uma forma de apresentação de representações visuais. Entretanto, optamos por classificá-la assim dada a representatividade que tem.

Um aspecto importante é a forma de associação de entidades e atributos a representações visuais. Isto corresponde à classe de ferramentas denominada **mapeamento**. As representações visuais são, na realidade, resultado de mapeamentos do

espaço dos objetos para o espaço das representações. Estes mapeamentos devem ser declarados explicitamente, de forma interativa. A alternativa utilizada por SCHMIDT [1994] e LINDSTAEDT [em and.] é a associação entre uma lista de objetos/variáveis e uma lista de representações (fig. 4.6).

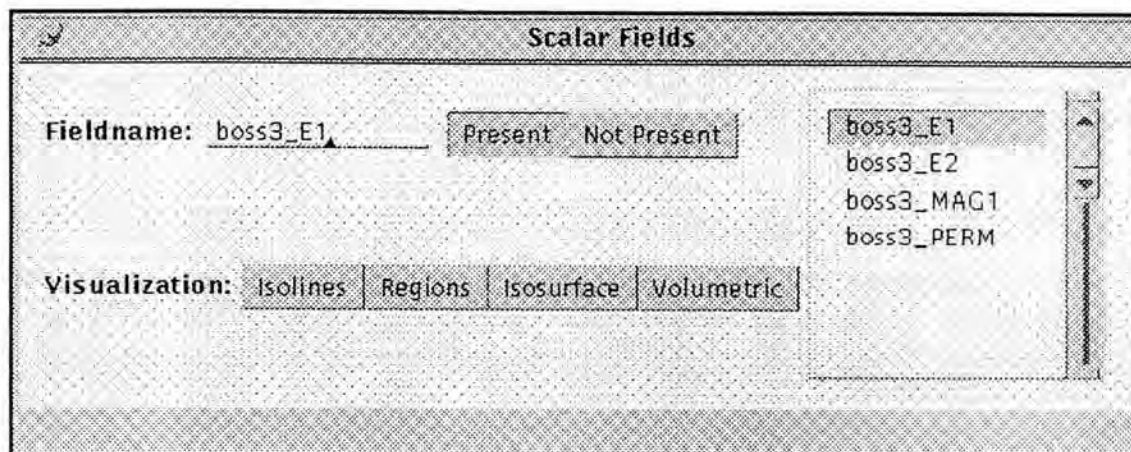


Figura 4.6 - Mapeamento de variáveis escalares no *EFVis* [SCHMIDT 1994]: os campos de escalares são selecionados e a representação gráfica é escolhida. A lista exibe os atributos selecionados, possibilitando também a composição de representações.

Mapeamentos são invocados automaticamente após declarados e podem ser alterados por ocasião de uma consulta ou do processamento.

4.3 Navegação

Navegar pelo objeto em estudo permite observá-lo de todas as posições possíveis, em função da natureza do objeto, mas, principalmente, da dimensão do seu domínio. A navegação corresponde ao posicionamento do usuário em relação ao objeto, requerendo uma ferramenta básica denominada **cursor**. Normalmente o cursor indica o que se costuma chamar de posição corrente (**cursor espacial**, navegação no espaço) ou objeto corrente (**cursor individual**, navegação num conjunto de objetos). A navegação, portanto, corresponde à alteração da posição deste cursor, conforme o interesse do usuário, propiciando uma visão do objeto segundo o cursor, sendo uma tarefa preparatória para as outras atividades como seleção de objetos e atributos, consulta, comparação, processamento e modificação de valores, conforme o caso. A abordagem

mais comum de cursor é um ícone que pode ser levado a diferentes posições do espaço-objeto representado na tela, indicando o próximo ponto de interesse.

Considerando-se objetos da classe sistemas compostos por múltiplas entidades, tipicamente representados por diagramas ou redes (rever fig. 4.2) bidimensionais, os recursos visuais necessários à navegação são fornecidos por ferramentas relativamente simples que permitem "passear" sobre o diagrama ou rede. Isto é comumente feito com ferramentas deslocadoras do foco de interesse (**focalizadores**). Uma implementação comum deste tipo de ferramenta é a barra de *scroll* em janelas. O mesmo tipo de navegação é utilizado em outras classes de objetos, representados por modelos 2D como, por exemplo, entidades naturais ou fenômenos representados visualmente como mapas (imagens).

No caso de modelos de domínio tridimensional, sistemas representados por diagramas ou redes de ícones 3D, objetos manufaturáveis, entidades ou fenômenos naturais amostrados no espaço, a navegação mais usual é a da **câmera sintética**. Duas situações surgem aqui: (1) a navegação pode ser apenas ao redor do objeto; (2) a navegação pode ser livre, tanto ao redor como no interior do objeto. O primeiro caso pode ser tratado como uma ferramenta tal qual a mostrada na figura 4.7, onde os controles da câmera permitem girar ao redor de um objeto - que é o ponto de interesse na cena - e, eventualmente aproximar-se e distanciar-se do mesmo.

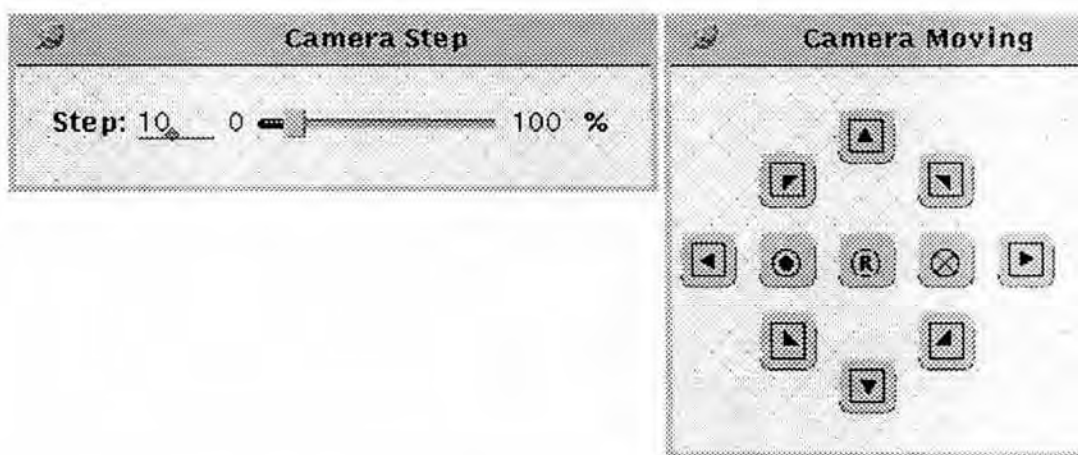


Figura 4.7 - Recurso de navegação **câmera sintética** para um objeto: os botões com setas permitem a movimentação em torno do objeto; os três botões centrais permitem aproximação, afastamento e *reset*.

O segundo caso deve ser tratado com controles tais que, além das funções do controle anterior, pode ser alterado o ponto de interesse, ou seja, a câmera pode ser direcionada para outra posição/orientação, além do centro do objeto.

4.4 Seleção de entidades e atributos

Conforme foi mencionado no capítulo 3, a tarefa de seleção de um conjunto de entidades ou de parte da massa de dados visa diminuir o volume de dados a ser navegado, consultado ou operado. Assim como a base da navegação é um cursor, a seleção é efetuada através de um **seletor**. Funcionalmente, os seletores determinam um ou mais "objetos correntes". Seletores podem ser de três classes:

- **seletor individual**, onde a seleção é feita por indicação de entidade(s) exibida(s) na tela;
- **seletor espacial**, pela delimitação física de uma parte do conjunto de entidades;
- **seletor lógico**, pela determinação de um critério com base nas propriedades das entidades. Esta classe de recursos é especialmente útil para casos onde o volume de dados é muito grande.

Um seletor individual (de indicação) é um cursor individual associado a uma confirmação, usado para a escolha de entidades designadas por atributos tipo característica ou para entidades modeladas como sistemas compostos por múltiplas entidades. É encontrado usualmente em qualquer aplicação interativa. Um seletor espacial é uma entidade geométrica, dependente da dimensão do domínio da entidade e da dimensão do sub-domínio desejado, como enumerado a seguir:

- uma janela, quando se quer limitar uma área a ser observada sobre uma região 2D ou sobre uma linha;
- um volume, quando o objeto está definido no espaço 3D e se quer limitar um sub-volume;
- um plano de corte aplicado a um volume quando se quer obter um plano ou um subvolume;

- uma linha quando se quer limitar a consulta a esta categoria de entidade geométrica;
- um cursor espacial de ponto, quando se quer seleccionar um ponto apenas do espaço-domínio.

Uma forma simplificada de seletor espacial tipo plano de corte especifica os parâmetros do mesmo (fig. 4.8), sem manipular sua representação geométrica. Uma forma mais realista, entretanto, implica no controle interativo do plano de corte através de sua representação visual, o que pode ser obtido através de mecanismos análogos aos propostos para o controle da câmara sintética. A forma da fig. 4.8 tem a vantagem de ser de fácil manipulação pois apresenta as possibilidades considerando o domínio da entidade a ser explorada.

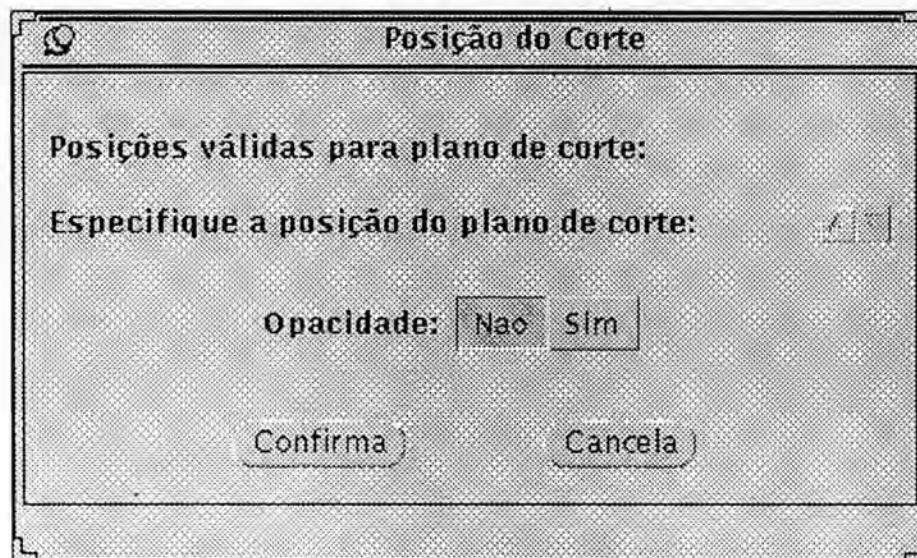


Figura 4.8 - Seletor espacial plano de corte, implementado por MANSSOUR [1994].

Um seletor lógico é um critério especificado com base nos valores de atributos das entidades. É necessário que uma ou mais entidades tenham sido previamente representadas visualmente ou selecionadas. São especificadas, então, restrições sobre valores das propriedades destas entidades. Estas restrições são, na realidade, expressões lógicas envolvendo os seus atributos:

- um valor de um determinado atributo tipo característica seleciona um subconjunto da relação original;
- um intervalo de valores de um atributo escalar numa área plana seleciona uma região;

- diversos valores fixos de uma atributo escalar numa área plana selecionam sub-áreas;
- um ou mais intervalos de valores de um atributo escalar numa região do espaço selecionam uma ou mais sub-regiões (fig. 4.9);
- um ou mais valores de um atributo escalar numa região do espaço selecionam sub-regiões (fig. 4.10).

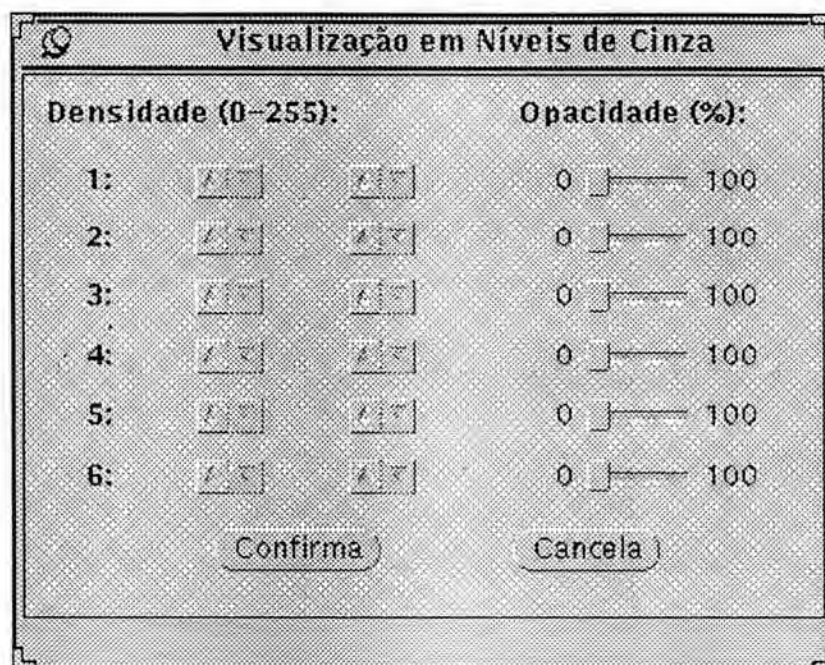


Figura 4.9 - Seletor lógico definindo intervalos de densidade a serem extraídos de um volume de dados tomográficos, como isosuperfícies, combinado a **mapeamento** para grau de opacidade em mapa volumétrico, tal como implementado por MANSSOUR [1994].

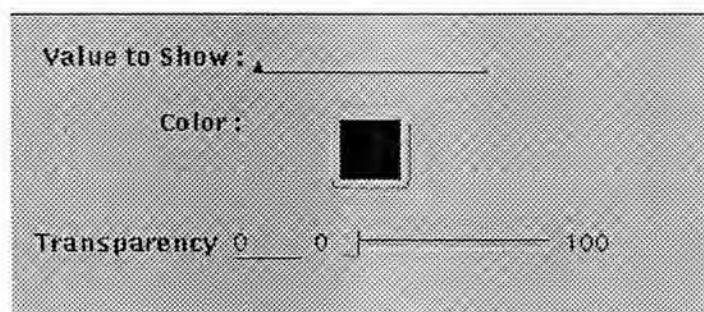


Figura 4.10 - Seletor lógico definindo um valor escalar para região a ser exibida; a seleção lógica está associada a um **mapeamento** para cor e grau de transparência da superfície desejada, considerando mapa volumétrico de um objeto 3D [SCHMIDT 1994].

4.5 Consulta e comparação de entidades e atributos

Representações visuais de uma entidade ou atributo permitem, em primeira instância, uma análise qualitativa do objeto. Uma verificação quantitativa é possível através da observação do valor numérico do atributo em questão. **Consultas** estão associadas ao tipo do dado sob análise, resultando uma representação gráfica ou valores numéricos, dependentes do domínio do atributo. As ferramentas de Consulta podem ser dos seguintes tipos:

- **leitor**, que serve a atributos tipo característica, sendo apenas a exibição dos valores associados à entidade indicada, ou seja, presentes no modelo;
- **medidor escalar**, para o caso de obtenção de valores escalares. Os valores escalares em geral são calculados por interpolação ou aproximação, com base naqueles presentes no modelo;
- **medidor vetorial**, análogo ao medidor escalar, para grandezas vetoriais.

Um exemplo de leitura é a obtenção de valores de outras características de um determinado espécimen, representado num gráfico como o da fig. 4.3. Um medidor escalar pode ser utilizado para retornar a temperatura de um ponto na massa de ar, quando este atributo está sendo exibido como uma mapa de cores no espaço. O valor do ponto em questão é calculado por interpolação dos valores presentes no modelo e que foram coletados por um radar meteorológico. Estas ferramentas estão, portanto, associadas aos atributos coletados ou calculados previamente e atuam sobre um ponto selecionado, uma entidade ou um grupo de entidades, conforme estejam definidos os atributos em questão.

Há, entretanto, grandezas que são derivadas de atributos escalares ou vetoriais. Neste caso, é necessário um conjunto de **avaliadores**, que façam uso de funções de cálculo dependentes da aplicação. Estes avaliadores produzem novos dados a partir dos existentes e atuam sobre um sub-espaço ou subconjunto de entidades/atributos selecionados. Os avaliadores se enquadram melhor como ferramentas de processamento e, portanto, serão tratados na seção 4.6.

Tanto a verificação qualitativa como a quantitativa de atributos de entidades devem poder ser comparativas, no sentido de que entidades diferentes, atributos diferentes da mesma entidade ou o mesmo atributo processado com estímulos diferentes possam ser

analisados conjuntamente. Isto significa compor ou colocar lado a lado representações visuais das diferentes situações e permitir sua manipulação como elementos de um espaço de representações. As representações, sendo, então, entidades (visuais), devem poder ser manipuladas geometricamente (transladadas) para melhor observação. A composição de representações visuais numa mesma imagem é automática no caso de serem escolhidos mapeamentos de atributos diferentes do mesmo objeto.

4.6 Preparação, processamento e controle

O estudo das entidades representadas num modelo envolve, em geral, algum tipo de processamento, o qual pode variar do cálculo de estatísticas simples como média e desvio padrão até a simulação do comportamento das entidades sob certas condições de contorno. O cálculo de novos dados a partir dos existentes corresponde aos avaliadores mencionados anteriormente. Esta classe de recursos é, portanto, dependente da natureza das entidades, da natureza e dimensão do domínio dos atributos. Por exemplo, a realização de uma análise de componentes principais requer a seleção das características a serem correlacionadas (uso prévio de um seletor lógico), assim como a seleção da amostra (seleção espacial). No caso de entidades da classe objetos manufaturáveis, pode ser requerida a avaliação num ponto, numa linha, numa área ou num volume, dependendo da grandeza a ser calculada [SCHMIDT 1994] (vide fig. 4.11).

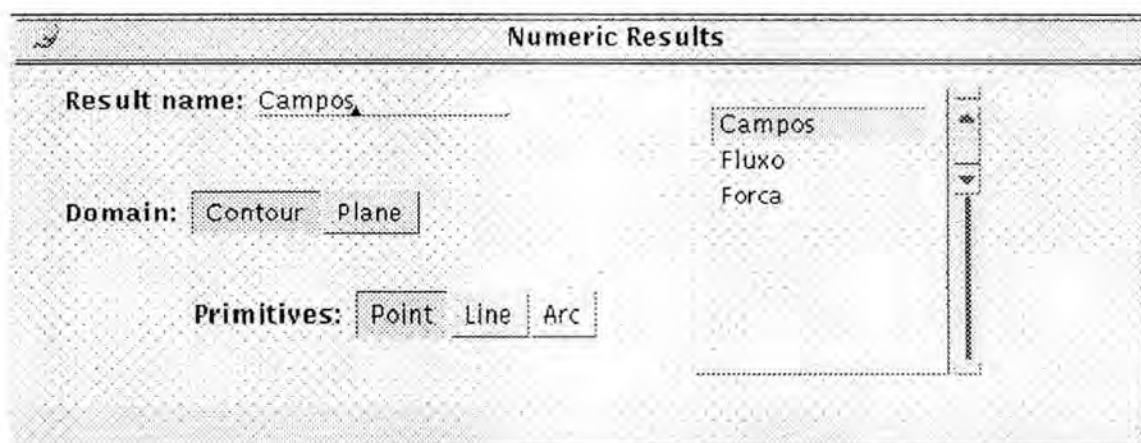


Figura 4.11 - Ferramenta de avaliação, indicando domínio da variável e região de avaliação (ponto, linha, arco), tal como especificado no *EFVis*. Neste caso, o sistema calcula a grandeza automaticamente, ou seja, o nome da grandeza aciona a função que a calcula.

Já em sistemas de múltiplas entidades submetidos à simulação é usual a coleta de estatísticas, que devem ser analisadas: o conjunto de avaliadores é, portanto, diferente, adequado à análise de resultados de experimentos. Em se tratando de entidades naturais, como populações, a forma de avaliação usual é o cálculo de grandezas estatísticas. No caso de fenômenos naturais, os processos de avaliação estão mais relacionados com o mapeamento dos valores obtidos em coleta ou simulação para o domínio do fenômeno, sendo usual uma avaliação semelhante a dos modelos geométricos.

Em aplicações de simulação interativa visual é pressuposto básico a possibilidade de execução monitorada de experimentos. Portanto, as possibilidades de interação não se tratam apenas de consultar dados já gerados ou coletados, mas de verificar durante o processamento, o que se passa com entidades do modelo. Este caso só se dá quando o modelo deve ser experimentado ou processado segundo outros parâmetros.

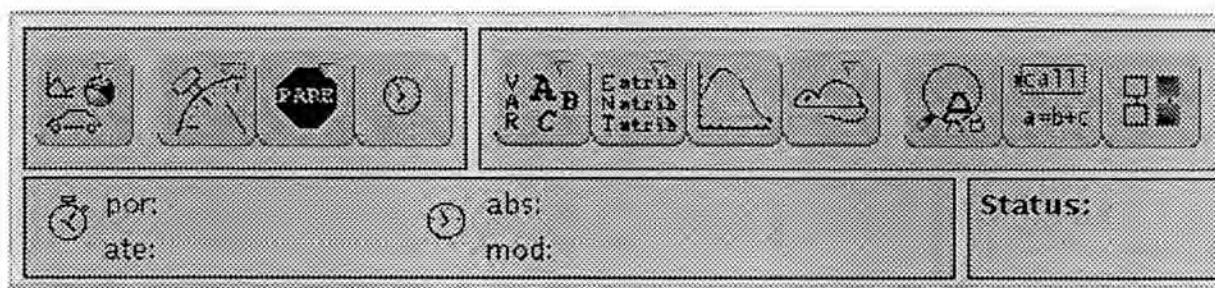
O conjunto dos recursos necessários a um ambiente que permita a modificação de valores de parâmetros do modelo é cumulativo em relação às situações anteriores. Assim, após a seleção da entidade ou atributo e a consulta a ser valor atual, deve-se proceder a modificação através de um mecanismo de **atribuição**, cuja essência é a entrada de dados compatível com o atributo em questão. O mecanismo é representado por duas ferramentas de **atribuição**. A primeira permite atribuição de valores quaisquer aos atributos; a segunda apresenta a particularidade de fixar um valor constante para o atributo. Isto é especialmente útil para a depuração e validação de modelos de simulação e teste de hipóteses.

Os recursos de representação visual (seção 4.2), navegação (seção 4.3), seleção (seção 4.4), consulta e comparação (seção 4.5) são também recursos destinados à monitoração do processamento. Para que o mapeamento seja realizado durante a execução de experimentos ou processamento, os mapeamentos devem ter sido previamente indicados e o mecanismo de disparo do processamento deve ativá-los. Para controlar a execução dos experimentos propriamente ditos são necessários os seguintes mecanismos [LINDSTAEDT em and.]:

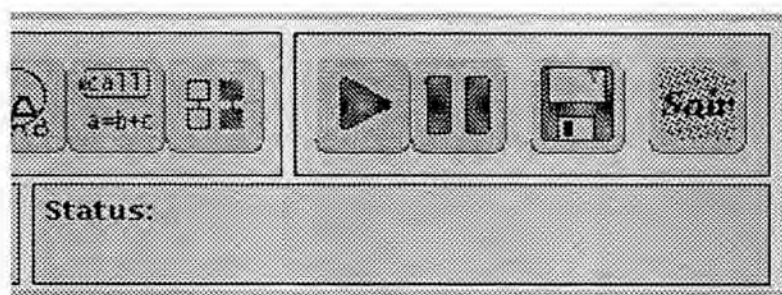
- **cronômetro**, que permite indicar qual deve ser a evolução do processamento no tempo, ou seja, passo a passo, por Δt tempo ou até o tempo t ;

- **indicador de pontos de parada**, que permite estabelecer as condições sobre as variáveis no modelo nas quais o processamento deve se interromper para que o usuário tome alguma decisão;
- **histórico do processamento**, que corresponde à facilidade de indicação de arquivos onde devem ser armazenados valores de variáveis em monitoração;
- **execução** do experimento, que é um recurso específico para indicar o início de um experimento de simulação sobre um modelo previamente preparado.

LINDSTAEDT [em and.] implementa recursos deste tipo sob a forma de ícones numa *toolbar* (fig. 4.12) os quais acionados ativam procedimentos específicos de requisição de parâmetros através de janelas. Estes mecanismos são utilizados para monitorar e controlar simulações usando o ambiente *Simscrip II.5*.



(a)



(b)

Figura 4.12 - Mecanismos de controle de experimentos: (a) ícones para escolha de representações visuais, cronômetro, indicador de pontos de parada, especificação de unidade de tempo, monitoramento de variáveis e entidades do modelo, especificação de parâmetros para experimentos, navegação em representações visuais, ativação de *trace* e observação da lista de eventos; (b) ícones para disparo e parada do experimento.

4.7 Anotação e armazenamento de imagens

As representações visuais geradas pelas ferramentas de mapeamento devem poder ser manipuladas graficamente e armazenadas para análise posterior ou, simplesmente, para apresentação. As ferramentas que reúnem os recursos necessários para tal são duas: ferramentas para anotação e para armazenamento de imagens.

- **anotação**, permite apor um texto a uma representação visual de forma semelhante a uma legenda. Esta é uma ferramenta que complementa os recursos fornecidos pelos mapeamentos já que, naqueles, são colocados textos simples, em geral relacionados com a identificação das entidades e atributos representados visualmente. A ferramenta de anotação é baseada num cursor simples que permite indicar, no espaço da representação visual, a posição onde deverá ser inserido o texto entrado via uma caixa de diálogo simples;
- **armazenamento de imagens**, permite indicar que representações visuais correntemente exibidas devem ser armazenadas em arquivo para posterior recuperação. Esta facilidade é análoga à captura de imagens através do utilitário *Snapshot* disponível em ambiente *OpenWindows*. Cabe ressaltar que esta recuperação é só para exibição; para que seja possível interagir novamente com o objeto, este deve ser recuperado e um mapeamento deve ser aplicado.

4.8 Resumo para referência

Ao longo deste capítulo foram introduzidas várias ferramentas como coleções de recursos interativos visuais para o suporte às tarefas em análise exploratória e simulação interativa visual. Na medida do possível as ferramentas foram exemplificadas. A seguir sua composição e objetivos são estabelecidos, de forma sintética e direta, propiciando um quadro de referência ao próximo capítulo onde estas ferramentas aparecem na arquitetura da plataforma.

A. Ferramenta de Mapeamento

Representação visual: ícone específico e caixa de parâmetros;

Procedimento interativo de base: escolha da representação visual para entidade ou atributo; especificação de parâmetros de mapeamento.

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;
Tarefa que suporta: todas.

B. Ferramentas de exploração

B.1 Cursor individual

Representação visual: ícone;
Procedimento interativo de base: deslocamento do ícone no domínio dos dados;
Atua sobre: qualquer representação visual de entidades;
Tarefa que suporta: navegação e seleção individual.

B.2 Cursor espacial

Representação visual: ícone;
Procedimento interativo de base: deslocamento do ícone no domínio dos dados;
Atua sobre: qualquer representação visual de entidades;
Tarefa que suporta: navegação e seleção espacial.

B.3 Focalizador

Representação visual: janela de tamanho ajustável e com barras de *scroll* horizontal e vertical;
Procedimento interativo de base: tratamento de *scroll* e de redimensionamento: mapeamento de porção do domínio 2D para a área da janela;
Atua sobre: representação visual de entidades 2D;
Tarefa que suporta: navegação.

B.4 Câmera sintética

Representação visual: quadro de controle;
Procedimento interativo de base: tratamento dos parâmetros de foco; mapeamento de porção do domínio 3D para a área da janela;
Atua sobre: qualquer representação visual de entidades 3D;
Tarefa que suporta: navegação.

B.5 Seletor individual

Representação visual: ícone do cursor individual;
Procedimento interativo de base: apontamento utilizando o cursor individual;
Atua sobre: qualquer representação visual de entidades;

Tarefa que suporta: seleção de entidades.

B.6 Seletores espaciais

Representações visuais e procedimentos interativos de base:

- a) Janela sobre domínio 2D ou 1D; ajuste de janela em torno do cursor espacial ou individual;
- b) Volume sobre domínio 3D; ajuste do volume em torno do cursor espacial;
- c) Plano de corte sobre domínio 3D; ajuste do plano de corte no ponto do cursor espacial;
- d) Linha sobre domínio 2D ou 3D; definição da linha a partir do cursor espacial;
- e) Ponto em domínio 2D ou 3D; definição do ponto indicado pelo cursor espacial.

Atuam sobre: qualquer representação visual de entidades

Tarefa que suportam: seleção de entidades e atributos.

B.7 Seletor lógico

Representação visual: quadro de atributos e expressão associada;

Procedimento interativo de base: expressão com os atributos ;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual ou espacialmente;

Tarefa que suporta: seleção de entidades e atributos.

C. Ferramentas de consulta

C.1 Leitor

Representação visual: mostrador de valores de atributos;

Procedimento interativo de base: exibição dos valores dos atributos da entidade selecionada;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;

Tarefa que suporta: consulta e comparação

C.2 Medidor Escalar

Representação visual: mostrador de valores de atributos;

Procedimento interativo de base: exibição dos valores calculados dos atributos da entidade selecionada;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;

Tarefa que suporta: consulta e comparação.

C.3 Medidor Vetorial

Representação visual: mostrador de valor escalar derivado;

Procedimento interativo de base: exibição dos valores calculados;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;

Tarefa que suporta: consulta e comparação.

D. Ferramentas de preparação, processamento e controle

D.1 Avaliador

Representação visual: mostrador de valor;

Procedimento interativo de base: entrada da expressão envolvendo atributos ou chamada de função; exibição do valor;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;

Tarefa que suporta: consulta, comparação, processamento.

D.2 Atribuição

Representação visual: mostrador de valor;

Procedimento interativo de base: entrada do valor do atributo;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;

Tarefa que suporta: preparação do processamento.

D.3 Atribuição de valor fixo

Representação visual: mostrador de valor;

Procedimento interativo de base: entrada do valor do atributo;

Atua sobre: qualquer representação visual de entidades e atributos previamente selecionados individual, espacial ou logicamente;

Tarefa que suporta: preparação do processamento.

D.4 Cronômetro

Representação visual: relógio com seleção de opções de execução

Procedimento interativo de base: seleção da opção de execução dentre as oferecidas:
passo a passo, por um intervalo de tempo Δt , até o tempo de processamento t .
Atua sobre: qualquer representação visual de entidades e atributos previamente
selecionados individual, espacial ou logicamente;
Tarefa que suporta: processamento (simulação).

D.5 Indicador de pontos de parada

Representação visual: quadro de atributos e de expressão;
Procedimento interativo de base: entrada de expressão envolvendo atributos;
Atua sobre: qualquer representação visual de entidades e atributos previamente
selecionados individual ou espacialmente;
Tarefa que suporta: processamento.

D.6 Histórico do processamento

Representação visual: quadro de atributos e indicação de arquivo de histórico;
Procedimento interativo de base: associação de atributos a arquivos;
Atua sobre: qualquer representação visual de entidades ou atributos sendo
processados;
Tarefa que suporta: processamento e armazenamento de dados.

E Ferramentas de anotação e armazenamento de imagens

E.1 Anotação

Representação visual: caixa de texto
Procedimento interativo de base: indicação, com cursor, da posição do texto;
entrada do texto;
Atua sobre: qualquer representação visual de entidades e atributos;
Tarefa que suporta: armazenamento de imagens

E.2 Armazenamento de imagens

Representação visual: ícone;
Procedimento interativo de base: indicação da representação visual a ser
armazenada;
Atua sobre: qualquer representação visual de entidades ou atributos;
Tarefa que suporta: armazenamento de imagens.

5 VISTA - Plataforma para análise exploratória e simulação interativa visual

A coleção de ferramentas identificada no capítulo 4 é reunida de tal forma a garantir as características essenciais a ambientes de análise exploratória e simulação interativa visual. São apresentadas e justificadas as abordagens adotadas tanto para a linguagem visual de interface com o usuário como para a estruturação da plataforma, cuja modelagem é aqui apresentada.

5.1 Características gerais e abordagens adotadas

Ponderando os diversos requisitos ou princípios de projeto enumerados por vários autores [BRITTAİN et al. 1990; DYER 1990; ELVINS 1990; HULTQUIST e RAIBLE 1992; ROOKS 1991; SCHROEDER et al. 1992; TURNER et al. 1990; UPSON et al. 1989], em relação às necessidades em diferentes situações, comentadas no capítulo 2, estabeleceu-se como características imprescindíveis a um ambiente que deve reunir facilidades de análise exploratória visual e simulação interativa visual: adequação e facilidade de uso, flexibilidade, consistência, extensibilidade e robustez.

Adequação e facilidade de uso estão associadas ao estilo de interação utilizado para implementar a linguagem visual. Interfaces do tipo manipulação direta que colocam o conjunto de facilidades do ambiente disponível no estilo *point-and-click* são muito mais fáceis de usar do que uma linguagem de comandos textual ou, mesmo, por menus. Por adequação do ambiente entende-se a transposição para a linguagem visual das expressões utilizadas cotidianamente pelo usuário, ressaltando-se ferramentas novas que venham a ser oferecidas. Os métodos usuais de análise dos objetos em estudo devem estar disponíveis de forma direta.

O ambiente deve ser **flexível** no sentido de que exigências variadas sejam atendidas com um mínimo de facilidades; isto deve ser provido pela linguagem visual, que dá acesso ao conjunto de ferramentas da plataforma. Problemas diversos devem poder ser investigados e isto significa que dados de natureza diversa devem poder ser representados e diferentes necessidades de exploração e processamento devem ser

suportadas. Pela identificação das necessidades de aplicações diferentes, foi determinado o conjunto de ferramentas apresentado no capítulo 4. As ferramentas podem ser combinadas de forma elegante e flexível, propiciando seu emprego de forma diversificada.

Consistência também diz respeito à linguagem visual pois deve-se garantir o comportamento das ferramentas de forma previsível. Por exemplo, dado um objeto definido no domínio 3D com um atributo escalar, uma consulta aos valores deste atributo realizada diretamente no espaço 3D deve fornecer os mesmos resultados que uma consulta operada sobre um corte previamente realizado.

A característica de **extensibilidade** é aplicável tanto à linguagem visual como à arquitetura do ambiente propriamente dita. O ambiente deve ser aberto, apto a receber novas formas de modelagem de dados e novas ferramentas de análise ou simulação. Projetistas podem construir novas ferramentas utilizando aquelas existentes e incorporando-as ao sistema e à linguagem visual.

Robustez diz respeito à prevenção de erros por parte do usuário. A experimentação é a essência da operação do sistema que, como tal, deve suportar ações do usuário na combinação de ferramentas e no seu uso sobre diferentes tipos de entidades e atributos.

Partindo do pressuposto que o usuário típico de aplicações de análise exploratória e simulação interativa visual é um cientista ou engenheiro buscando experimentar seus modelos e dados (frequentemente volumosos) de forma a descobrir fatos ali representados e resolver seus problemas mais rapidamente, sem escrever novos programas, a linguagem de trabalho neste ambiente é orientada para o uso de ferramentas visuais interativas que suportam as tarefas de análise e simulação. A linguagem visual aqui proposta sistematiza o uso das ferramentas num contexto genérico, sem atrelar-se a uma aplicação específica. Desta forma, aspectos de modelagem são tratados na medida do necessário às questões de análise e experimentação. A modelagem interativa dos objetos em estudo não é por ora tratada mas, garantindo-se a extensibilidade do ambiente, garante-se a possibilidade de incluir estas facilidades.

A adoção da **abordagem orientada a ferramentas** atende os requisitos de adequação, facilidade de uso e consistência da linguagem, em relação ao usuário típico.

De estudos experimentais [SPRINGMEYER 1992] sabe-se que o processo de análise de dados corresponde ao uso sistematizado e metodológico de ferramentas, tanto manuais como automatizadas. Assim, a adequação da linguagem é obtida através da transposição destas ferramentas para o ambiente: a facilidade de uso é garantida pela escolha do estilo de interface do tipo manipulação direta de ícones representando ferramentas e objetos. O aspecto de consistência é garantido pela uniformidade com que as ferramentas estão definidas, todas com comportamento pré-determinado, mas com flexibilidade de aplicação. Ferramentas podem ser aplicadas em ordem variada e sobre entidades de natureza diversa, facilitando tanto o aprendizado do uso do sistema como sua aplicação.

Os sistemas de visualização divulgados mais recentemente [HULTQUIST e RAIBLE 1992; SCHROEDER et al. 1992] têm seguido a abordagem orientada a objetos em função da reusabilidade e extensibilidade possíveis. Estes sistemas, entretanto, foram desenvolvidos fundamentando-se nos aspectos de visualização e não nos requisitos de análise exploratória. Na área de simulação, linguagens orientadas a objetos, como *MODSIM II* [BELANGER 1990; BELANGER 1991] e *Sim++* [LOMOW e BAEZNER 1990] também não são respostas às necessidades em termos de simulação interativa uma vez que os aspectos de representação visual e interação devem ser programados dentro do modelo de simulação. As características citadas acima são mais facilmente obtidas quando se constrói totalmente um sistema. Entretanto, a concepção de um sistema aberto baseado na abordagem orientada a ferramentas, onde cada ferramenta corresponde a uma classe, é garantia da possibilidade de adaptação de facilidades existentes, pois elas podem facilmente ser encapsuladas e agregadas ao ambiente como novas classes.

Assim, o paradigma de orientação a objetos foi escolhido para a estruturação do ambiente pois permite extensibilidade e robustez. Um aspecto importante é o fato de que uma arquitetura robusta e extensível, aliada a uma linguagem visual flexível, baseada numa abordagem coerente, permite a incorporação de todas as ferramentas de modelagem necessárias a um ambiente integrado completo.

A plataforma proposta é denominada *VISTA* e é estruturada de modo a refletir a abordagem adotada como paradigma de linguagem visual: uma **base de ferramentas** é uma coleção de objetos; as ferramentas são instanciadas pelo usuário ao serem aplicadas a elementos da **base de entidades**, as quais são também modeladas como objetos (fig. 5.1). Tem-se, então, duas grandes classes de objetos no sistema: objetos-ferramenta e objetos-entidade, os quais são manipulados através de uma linguagem visual. Uma

característica importante e única do sistema aqui proposto é que ferramentas aplicadas sobre entidades podem produzir outras entidades, que passam a fazer parte do ambiente e como tal podem ser exploradas e operadas.

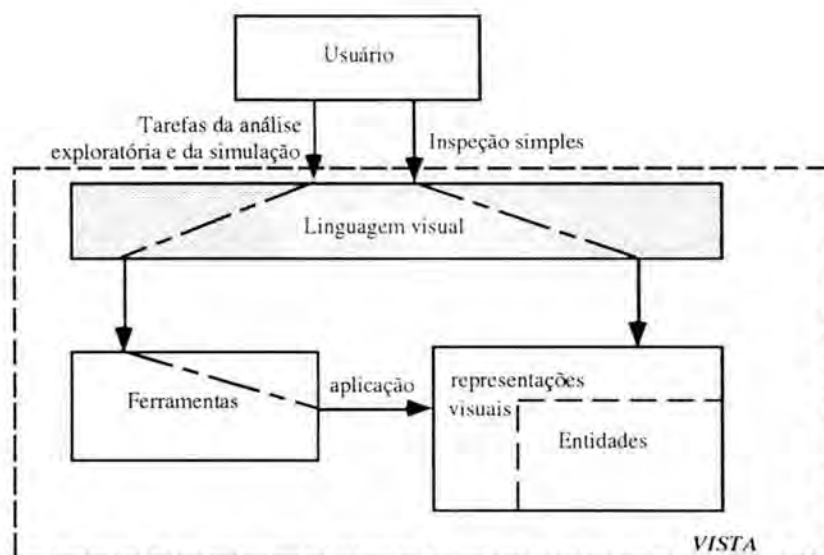


Figura 5.1 - Estrutura geral de VISTA.

5.2 A linguagem visual

5.2.1 Base de ferramentas e base de entidades

A linguagem de interface com o usuário é interativa, baseada na indicação da ferramenta a ser aplicada e na indicação do objeto sobre o qual ela deve atuar. Na sua forma ideal, deve haver uma interface textual, correspondente à gráfica, de modo que tarefas que não requerem interação, como a geração de imagens para apresentações, por exemplo, possam ser executadas isoladamente produzindo um arquivo. A descrição da linguagem, neste trabalho, é feita a partir de exemplos¹, para que o resultado de aplicação das ferramentas possa ser mais diretamente compreendido. A interface textual não é aqui abordada, embora seja viável.

¹ As figuras foram elaboradas "manualmente", não correspondendo a protótipos de modo que não houvesse atrelamento antecipado a um padrão de interface. A representação visual das ferramentas poderá ser alterada durante a implementação de modo a explorar as facilidades oferecidas pelo ambiente de desenvolvimento.

As ferramentas disponíveis estão "à mão" na **base de ferramentas** de *VISTA*, de onde podem ser retiradas para uso (fig. 5.2).

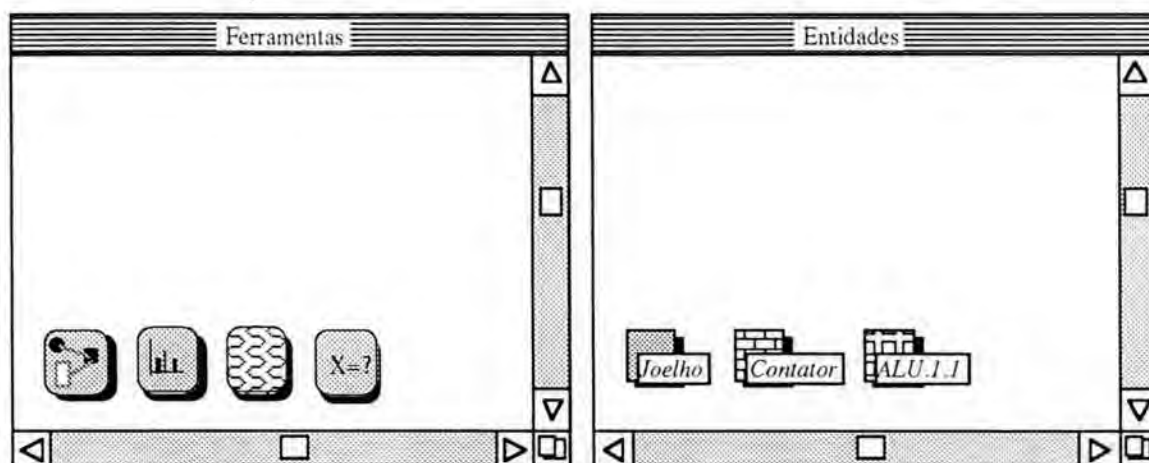


Figura 5.2 - Representação sugerida para a base de ferramentas e base de entidades.

Os objetos estão igualmente acessíveis ao usuário, através da **base de entidades**, que nada mais é do que o conjunto de objetos ou modelos sob investigação pelo usuário. Tanto a base de ferramentas como a de entidades podem ser inspecionadas visualmente através de uma ferramenta **focalizador**, representada pelas barras de *scroll*.

A base de ferramentas contém somente uma ferramenta de cada tipo mas a seleção de uma delas equivale a criar uma instância da mesma. Optou-se por uma base de ferramentas e não uma *toolbar* para possibilitar diferentes configurações e o acréscimo de novas ferramentas de forma transparente. Ferramentas são representadas por ícones; a utilização delas sobre entidades é realizada pela manipulação de seus ícones.

5.2.2 Mapeamento e navegação

As ferramentas atuam sempre sobre uma representação visual dos objetos a serem investigados; a determinação dessa representação visual é realizada através da aplicação de uma ferramenta tipo **mapeamento**. A vinculação de uma representação visual a um objeto é realizada buscando o mapeamento correspondente à representação visual desejada e levando-o até o ícone que representa o objeto. Esta vinculação é simples mas pode-se obter resultados poderosos como será visto adiante. Num primeiro momento, se

o objeto estiver selecionado como um todo há certas restrições na aplicação de mapeamentos, dependendo da classe do objeto e da natureza dos atributos que são utilizados para modelá-los. Assim, se o objeto for um sistema de múltiplas entidades descrito como uma agregação de outras entidades representadas por ícones, a representação visual imediata é um **diagrama**. Se for um objeto manufaturável, em geral há uma descrição geométrica como atributo, e a representação visual mais natural é o **modelo geométrico**.

A figura 5.3 mostra o resultado da aplicação de uma ferramenta **modelo geométrico** sobre um objeto manufaturável e a fig. 5.4 o de uma ferramenta **diagrama** sobre um modelo de simulação de um sistema digital. Os mapeamentos ativam automaticamente as ferramentas de navegação de acordo com o domínio da entidade: **câmera sintética** (o objeto está definido no espaço 3D) e **focalizador** (o diagrama do sistema é descrito no plano). Assim, na escolha de uma entidade ou na aplicação de um mapeamento, é tornada disponível uma ferramenta de navegação.

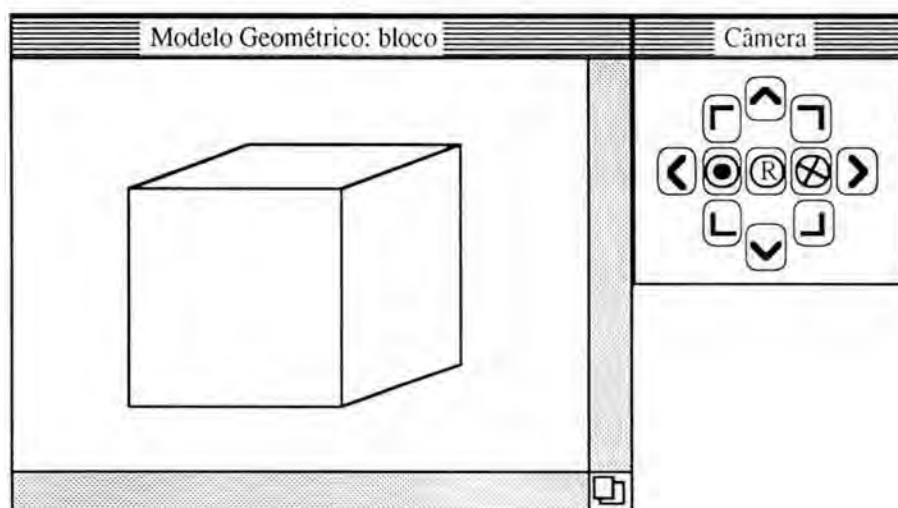


Figura 5.3 - Ferramenta de navegação **câmera sintética** associada a um objeto da classe objeto manufaturável, representado visualmente pelo mapeamento **modelo geométrico**.

A ferramenta **câmera sintética**, utilizada para navegação em domínios 3D, ao ser ativada, assume a forma de um quadro de controle, associado à janela onde é exibido o objeto sob análise. Através destes controles é possível orientar-se ao redor do objeto, assim como afastar-se e aproximar-se. Na figura 5.4, as barras de *scroll* permitem a navegação no diagrama e o botão de redimensionamento permite o enquadramento de maiores ou menores porções do objeto.

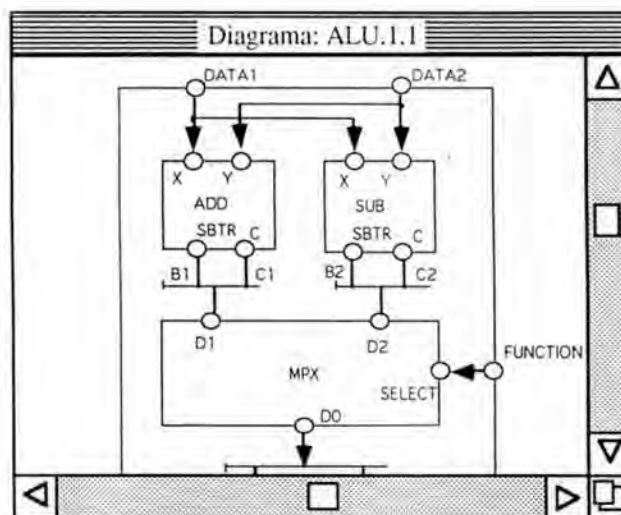


Figura 5.4 - Ferramenta **focalizador** (barras de *scroll*) usada sobre a representação por **diagrama** de uma entidade da classe sistema de múltiplas entidades.

5.2.3 Seleção e consulta

A exploração dos objetos se dá, na continuidade, com o uso de ferramentas de seleção. O **seletor individual** é buscado da base de ferramentas e levado até a entidade a ser indicada. Após a "aplicação" deste seletor, a entidade fica selecionada e seus atributos ficam disponíveis, aparecendo nominalmente numa janela à parte. Uma ferramenta de consulta tipo **leitor**, por exemplo, pode ser levada até o atributo desejado e seu valor específico é mostrado. A entidade selecionada pode ser uma entidade num diagrama (ver exemplo na figura 5.5, referente a um modelo de sistema digital) ou uma entidade num gráfico (figura 5.6, referente a um exemplar de uma população representado por um ícone num gráfico semelhante ao da figura 4.3). Eventualmente pode ser selecionada mais de uma entidade, bastando que outros seletores individuais sejam aplicados.

Seletores espaciais permitem indicar e extrair partes das entidades em estudo; os resultados são dependentes das classes das entidades. Estas ferramentas são selecionadas da base e aplicadas à representação visual de um objeto. Considerando um objeto manufaturável ou um fenômeno amostrado espacialmente, na aplicação de um seletor espacial tipo **janela**, a área delimitada fica marcada na representação visual e os atributos associados aparecem numa janela à parte. Na continuidade, o usuário pode buscar uma

ferramenta **mapeamento** e aplicar a um atributo (figura 5.7) ou, com o mesmo procedimento, aplicar uma ferramenta de consulta tipo **medidor escalar** (figura 5.8).

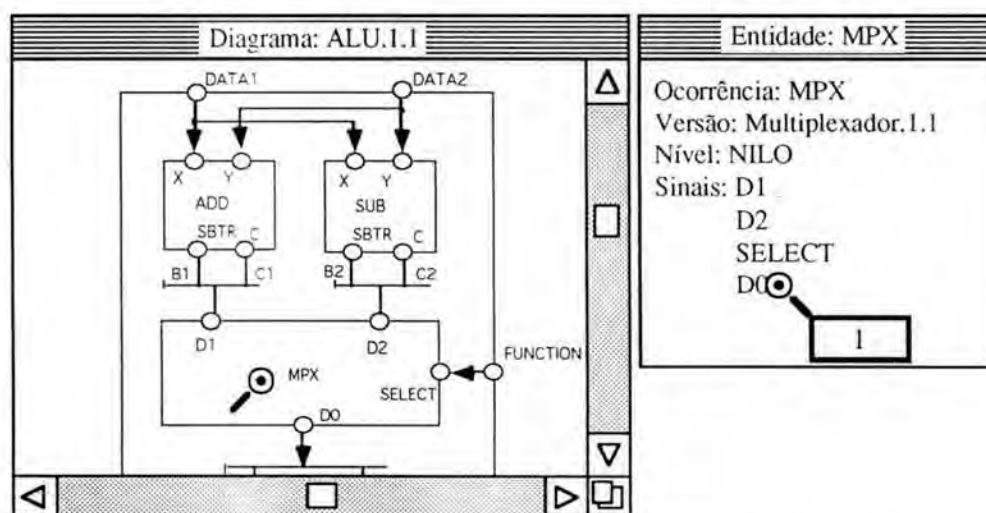


Figura 5.5 - Uso da ferramenta **leitor**: após a seleção da entidade MPX, seus atributos são mostrados e um **leitor** é utilizado para obter o valor do atributo D0.

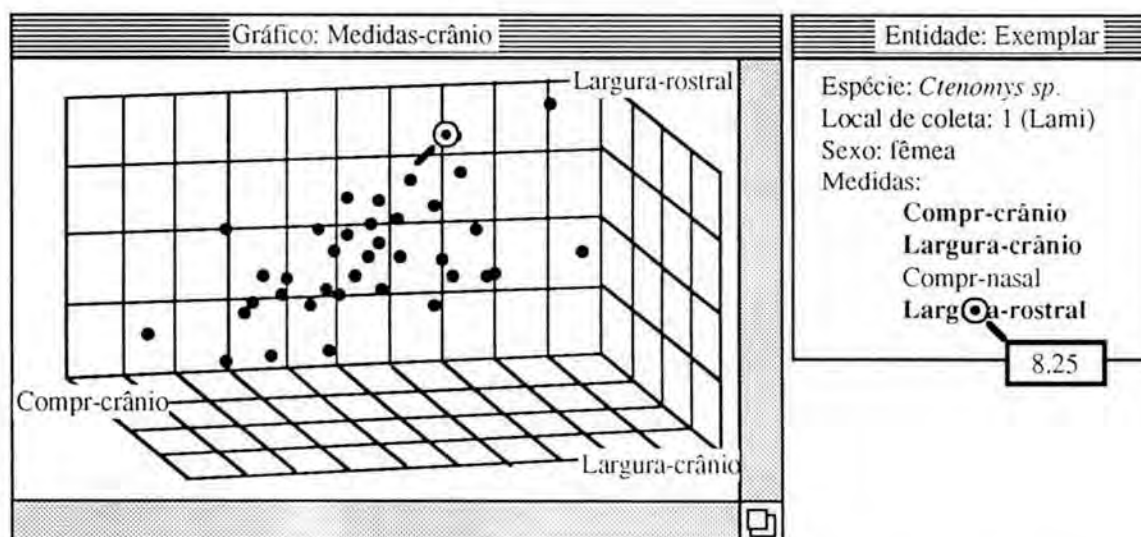


Figura 5.6 - Uso de seletor **individual** e **leitor** sobre entidade da classe entidade natural.

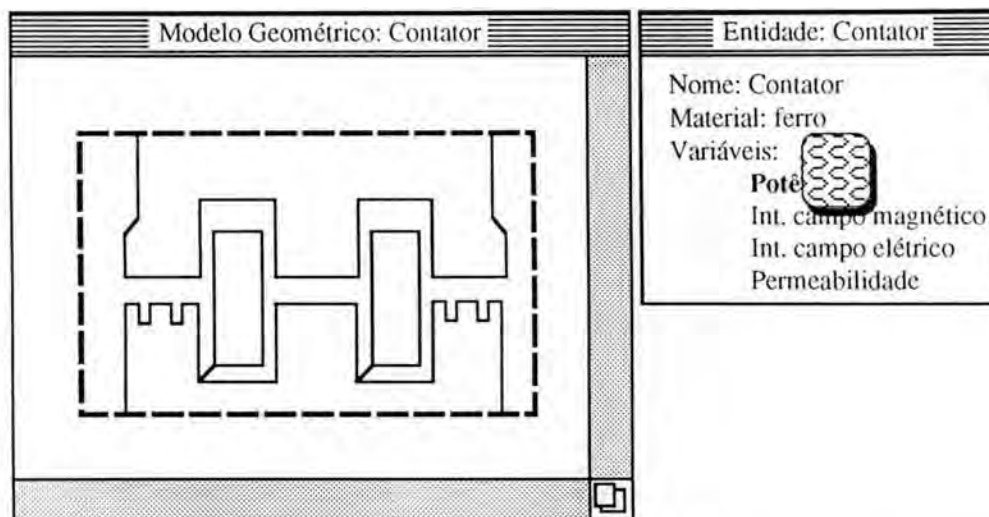


Figura 5.7 - Mapeamento **linhas de contorno** aplicado ao atributo "potência" da entidade (previamente exibida com **modelo geométrico**) na região selecionada pelo **seletor janela**.

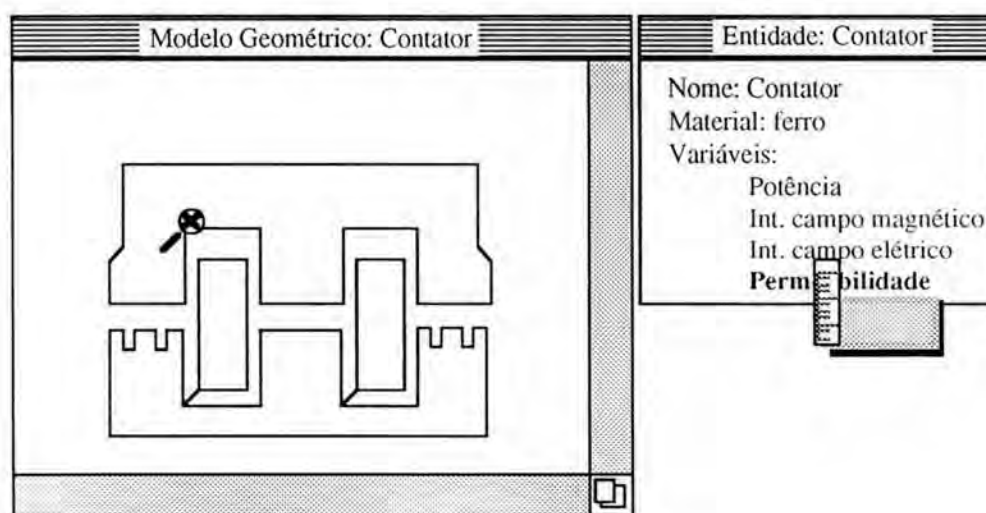


Figura 5.8 - Ferramenta **medidor escalar** aplicada ao atributo "permeabilidade" no ponto selecionado pelo **seletor ponto**.

Procedimentos análogos são efetuados para a seleção com volume, plano de corte, linha e ponto. Em qualquer um dos casos pode-se selecionar mais de uma instância da mesma ferramenta a selecionar diversas sub-regiões do domínio para consulta e comparação. Por exemplo, a observação de um atributo em diversos planos de corte de um objeto 3D pode ser feita aplicando várias instâncias deste seletor ao mesmo domínio (fig. 5.9). Observe-se os controles de posicionamento semelhantes ao da **câmera sintética** associados a um plano. Uma ferramenta de mapeamento tipo **linhas de**

contorno aplicada a estas instâncias produziria tantas representações visuais (do mesmo tipo) quantos fossem os seletores ativados. No caso de um sistema de múltiplas entidades, a aplicação de um **seletor janela** resulta um subconjunto das entidades do sistema.

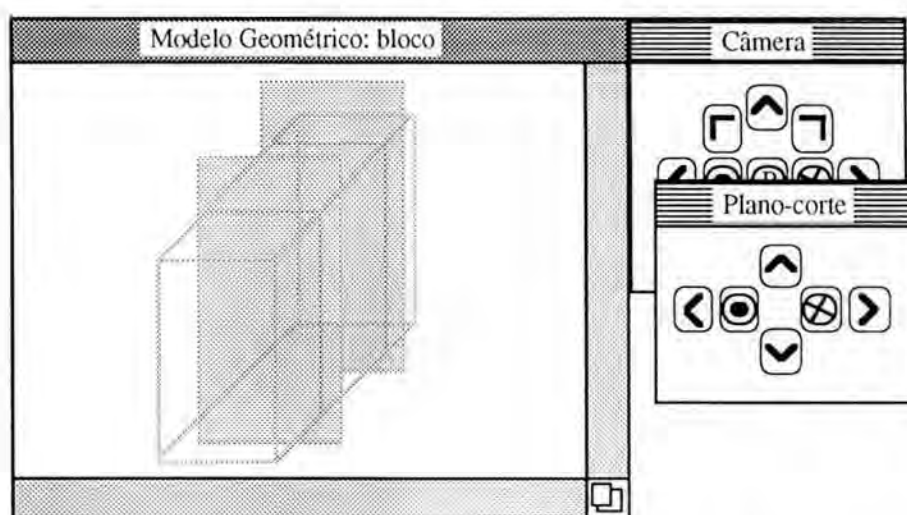


Figura 5.9 - Ferramenta de seleção espacial **plano de corte**, aplicada sucessivas vezes sobre uma entidade representada através de **modelo geométrico**.

Os **seletores lógicos** são aplicados a entidade(s) previamente selecionada(s) da base e representadas visualmente, correspondendo ao estabelecimento de critérios de seleção sobre os atributos. Assim, selecionada uma entidade ou grupo de entidades, os seus atributos são exibidos em janela à parte, como descrito anteriormente, e devem ser utilizados na construção de uma expressão lógica que especifique o critério de seleção desejado. O sub-domínio assim selecionado é passível de representação visual, bastando que seja aplicada uma ferramenta de mapeamento sobre o seletor.

A figura 5.10a mostra a aplicação de um **seletor lógico** sobre uma entidade; a expressão lógica é definida e um mapeamento é aplicado sobre o seletor (fig. 5.10b). Neste exemplo, um **seletor lógico** envolvendo um atributo "densidade" foi utilizado para indicar uma sub-região de um volume de dados tomográficos; o mapeamento escolhido foi o de **isosuperfície**; o resultado é a representação da isosuperfície correspondente à região com aquele valor de densidade, sobre o volume original (fig. 5.10c). Observe-se que a aplicação de um seletor lógico por si só não tem efeito visual imediato; fica a critério do usuário a especificação de um mapeamento.

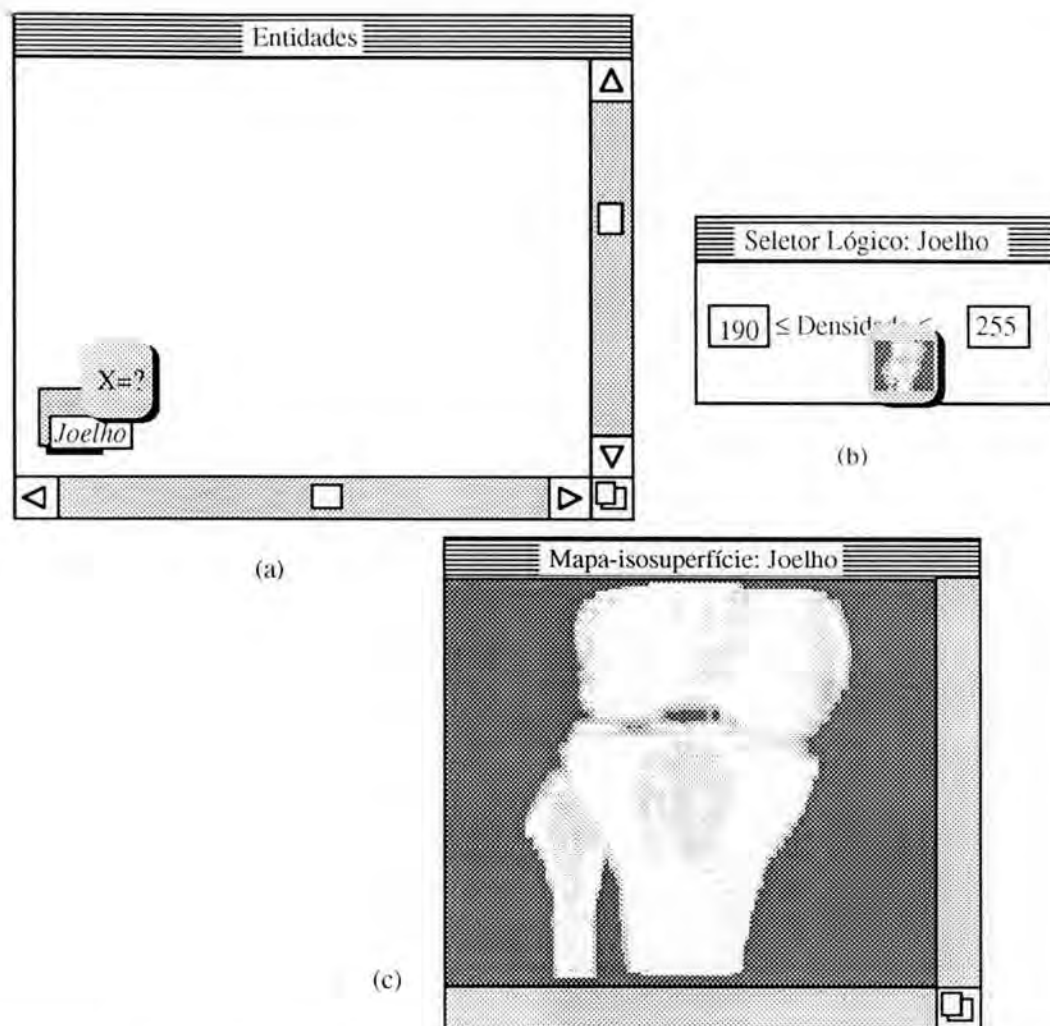


Figura 5.10 - (a) **Seletor lógico** aplicado a uma entidade natural (volume de dados tomográficos); (b) entrada da expressão e mapeamento para **isosuperfície** (c).

5.2.4 Avaliação

Sobre resultados de uma seleção também podem ser aplicadas ferramentas da classe **avaliador**, que correspondem a cálculo de uma ou mais grandezas em função de outras. O uso desta ferramenta requer a especificação, pelo usuário, de uma função dependente da aplicação, que gere a grandeza desejada. Por exemplo, a grandeza "indução magnética" pode ser obtida dos atributos "campo magnético" e "permeabilidade" na região delimitada por um seletor espacial. Se o domínio selecionado é um ponto, o resultado é exibido num mostrador, associado à janela onde está a representação visual da entidade. Se a seleção espacial indicar um domínio 2D, numa região, o resultado pode ser

exibido através de um mapeamento indicado da forma apresentada na fig. 5.11 (**mapa de cores falsas** sobre geometria).

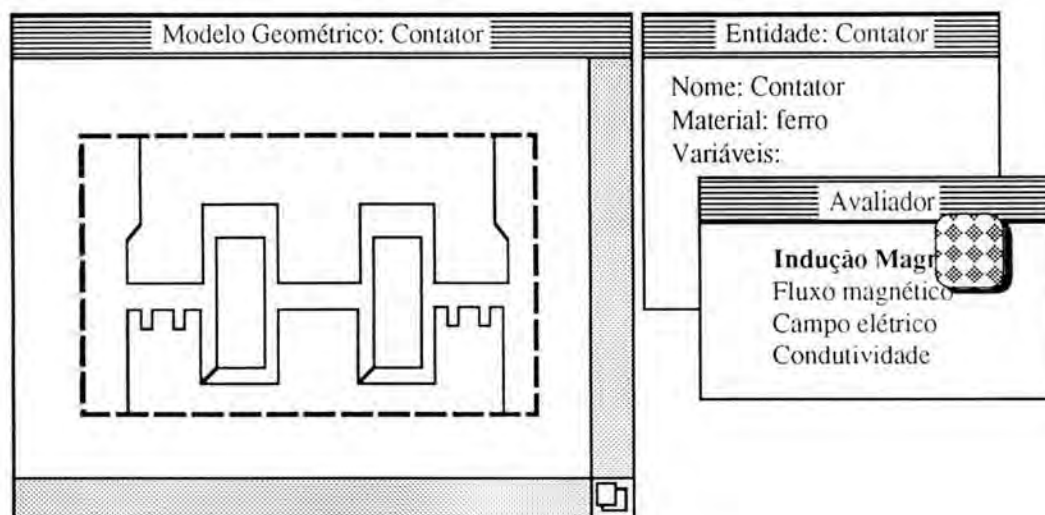


Figura 5.11 - Mapeamento para **cores falsas** aplicado ao resultado de **avaliador** da grandeza "indução magnética" na região selecionada através do **seletor janela**.

Uma ferramenta **avaliador** para entidades naturais é usada para obter valores de média e desvio padrão das medidas representadas num gráfico (semelhante ao da fig. 4.3); estes valores são exibidos em mostrador separado (fig. 5.12).

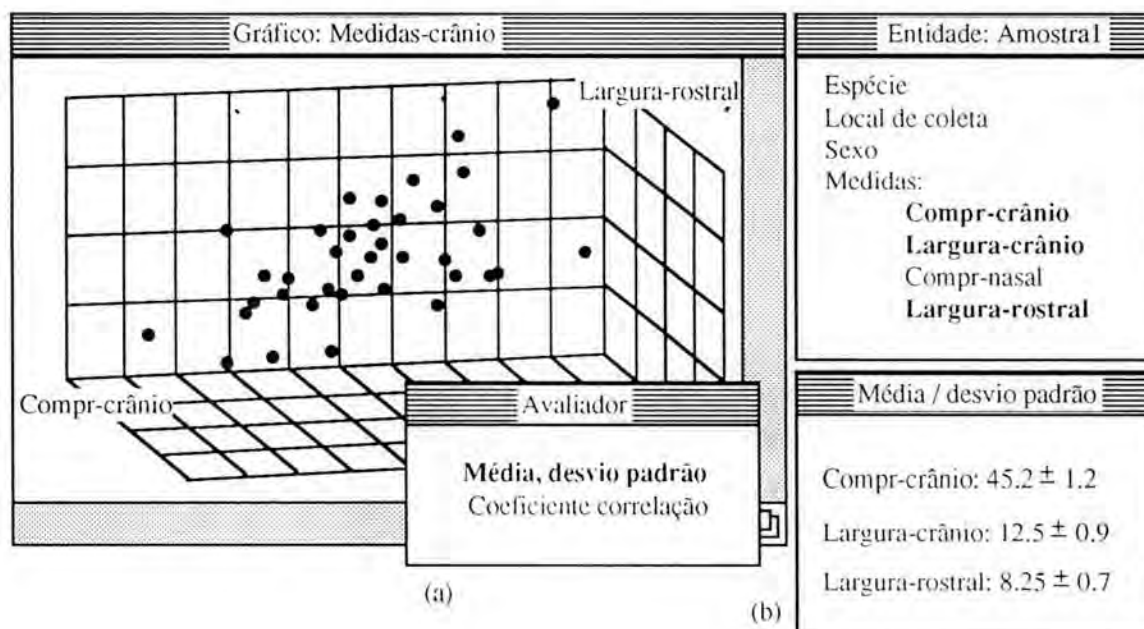


Figura 5.12 - (a) **Avaliador** aplicado a atributos de entidade exibidos com mapeamento para **gráfico**; (b) resultados numéricos exibidos à parte.

5.2.4 Preparação do processamento, controle e registro

Para a realização de outros tipos de processamento, entretanto, pode ser necessário estabelecer condições iniciais para parâmetros ou atributos. Neste caso, uma ferramenta **atribuidor** é utilizada para indicar estímulos a modelos de simulação que devem ser experimentados. A ferramenta deve ser buscada e aplicada sobre o atributo em questão e o novo valor deve ser especificado. Uma ferramenta de **atribuição fixa** permite indicar que um valor é constante para um atributo. Um exemplo de **atribuidor** aparece na figura 5.13, associado a um modelo a ser simulado.

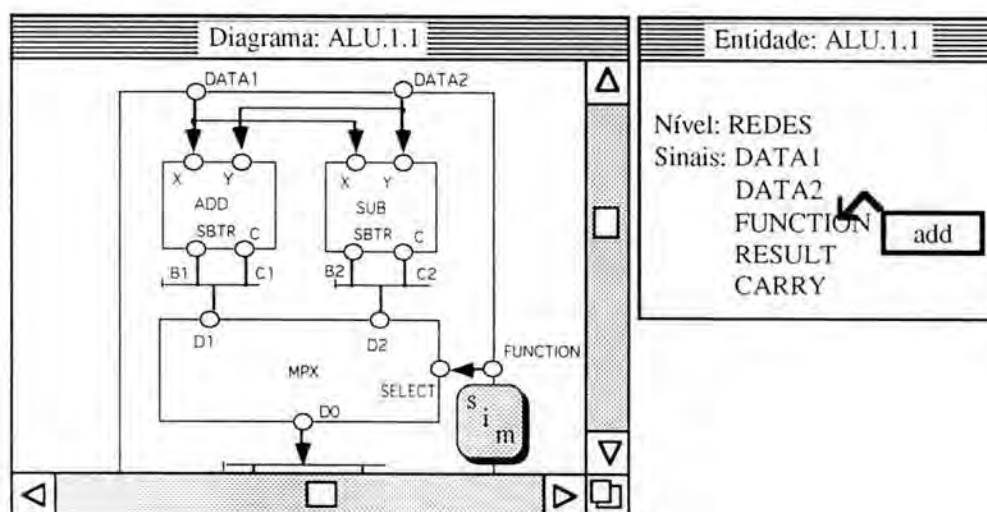


Figura 5.13 - Ferramenta de **atribuição** aplicada a atributo FUNCTION de entidade a ser simulada. Observa-se um ícone representando a ferramenta **experimento**, para preparação de experimentos de simulação.

As ferramentas **cronômetro** e **indicador de pontos de parada** são utilizadas na fase anterior ao experimento, preparando seu controle. Cronômetro permite indicar o comportamento da execução em relação ao tempo - passo a passo, por Δt tempo ou até o tempo t - e Indicador-de-Pontos-de-Parada permite estabelecer condições de parada da execução, dependentes de valores de variáveis. Caso o usuário queira armazenar a evolução do processamento em termos de valores de variáveis, há a ferramenta **histórico do processamento**, onde pode-se especificar o nome de um arquivo de histórico para o salvamento de valores de atributos. Todas estas ferramentas estão associadas a uma de mais alto nível que é a **experimento**, ativada através da aplicação de seu ícone à entidade a ser simulada (fig. 5.13). Neste caso, as informações requisitadas pelas ferramentas são indicadas em janelas específicas como as mostradas na figura 5.14, onde se vê o indicador Cronômetro, a facilidade de definição de arquivo de

log e o painel para ativação e desativação de monitoramento de atributos. A escolha da representação visual para monitoramento é feita como nas formas anteriores: tomando o ícone do mapeamento e levando-o até o nome do atributo na lista.

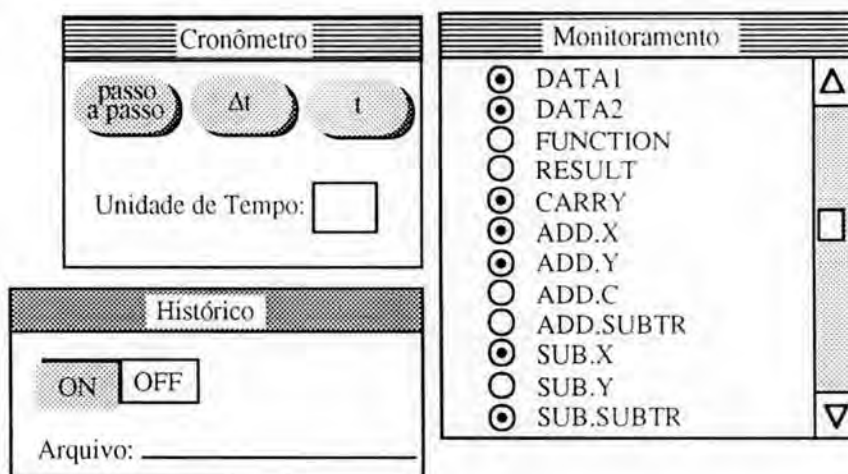


Figura 5.14 - Facilidades de controle propostas com a ferramenta **experimento**.

Eventualmente pode ser necessário armazenar representações visuais ao invés de valores numéricos de variáveis; isto é possível com duas ferramentas: **anotação**, para apor textos (legendas) em imagens, e **registro de imagens**, para o armazenamento em arquivo de imagens selecionadas espacialmente, direto na tela. A ferramenta de **anotação** poderia ser utilizada na imagem apresentada na figura 5.10c, para colocar um texto indicativo de que a densidade selecionada corresponde ao tecido ósseo no volume de dados tomográficos. A ferramenta de **registro de imagens** poderia ser empregada a seguir, para "salvar" esta imagem num arquivo de modo a permitir *hardcopy* posterior.

5.3 Modelagem de VISTA

5.3.1 Comentários iniciais

Tanto as ferramentas como as entidades sob investigação são objetos. Desta forma a aplicação de uma ferramenta a uma entidade desencadeia a execução de métodos próprios de cada entidade. Conforme a ferramenta, informações são requisitadas para a entidade que as devolve e a ferramenta realiza a função associada. Conforme já mostrado (fig. 5.1), VISTA é estruturada a partir de duas grandes classes: ferramentas e entidades.

Para o projeto de VISTA foi escolhida a metodologia OMT [RUMBAUGH et al. 1991], porque produz uma descrição que integra os aspectos de estrutura estática e dinâmica do sistema sendo projetado. Esta metodologia utiliza três tipos de modelos para descrever completamente um sistema: o **modelo de objetos**, o **modelo dinâmico** e o **modelo funcional**. O modelo de objetos descreve os objetos no sistema e suas relações; o modelo dinâmico descreve as interações entre os objetos, em função do tempo, e o modelo funcional descreve as transformações nos dados.

Na notação OMT para o **modelo de objetos**, classes são representadas por retângulos divididos em três partes: a superior contém o nome da classe; a intermediária, o nome dos atributos, e a inferior, as operações suportadas pelas classes. Subclasses são representadas por uma divisão em níveis, a partir de um triângulo que representa a generalização. Eventualmente, existe um atributo que discrimina os membros das subclasses; este atributo tem seu nome escrito ao lado da representação gráfica de generalização. Associações entre membros das classes têm seus nomes identificados sobre a representação gráfica da associação: os papéis que cada classe representam na associação são identificados ao lado de cada uma. A cardinalidade das associações é representada de diferentes formas: uma linha simples indica uma associação 1-para-1 ou 1-para-n, com n colocado ao lado da classe; um círculo vazio representa uma associação 1-para-1 opcional; um círculo cheio indica uma associação 1-para-n opcional. Numa associação 1-para-n, os membros podem estar ordenados. Uma associação do tipo agregação tem uma simbologia especial, um losango. Uma associação pode ser modelada como uma classe, com atributos e, eventualmente, operações associadas. Neste caso, a linha que normalmente representa uma associação é acrescida da representação gráfica da classe; a classe correspondente ao mapeamento é conectada à associação através de uma linha curva, conforme utilizado na figura 5.21.

O **modelo dinâmico** é representado por **diagramas de transição de estados**, capturando o aspecto de controle que descreve a seqüência de operações sem preocupar-se com a sua implementação. **Estados** são representados por elipses e **transições** por linhas rotuladas com os nomes dos eventos. Operações podem estar associadas a estados, quando são realizadas em decorrência da entrada do objeto nos estados, ou a transições, quando estão relacionadas a eventos. Operações podem corresponder a **atividades** ou a **ações**; atividades são operações com duração considerável, enquanto ações, são instantâneas para efeito de modelagem. Cada diagrama

de estados representa o comportamento de uma classe de objetos. As atividades aparecem dentro da notação gráfica de estado, sendo precedidas pela palavra *do*. As ações, sendo associadas a eventos, aparecem após os respectivos eventos, nas transições. Os estados num diagrama representam uma dada classe de equivalência de valores de atributos e de associações dos objetos cujo comportamento está sendo modelado. Os eventos, correspondendo a mudanças de estado, podem ser representados como operações no modelo de objetos, embora sejam mais expressivos que estas.

O **modelo funcional** corresponde a diagramas de fluxo de dados (DFDs) que indicam como os valores dos atributos são calculados, ou seja, mostra a dependência funcional entre os valores dos atributos dos objetos representados no modelo de objetos. Diagramas de fluxo de dados contêm **processos, fluxos de dados e repositórios de dados**. Processos transformam dados, fluxos indicam a passagem de dados entre processos ou entre processos e repositórios e estes últimos armazenam dados passivamente. Os processos neste modelo correspondem às atividades e ações do modelo dinâmico; fluxos correspondem a objetos ou atributos do modelo de objetos.

As seções a seguir apresentam o modelo de objetos da plataforma já que tanto entidades como ferramentas são modeladas como classes. É apresentado o modelo dinâmico de apenas duas ferramentas, já que o comportamento da plataforma é padronizado. Este modelo é suficiente para compreender genericamente o funcionamento de *VISTA*. Para simplificação deste texto, o modelo funcional é dispensado. A descrição das classes apresentadas nas próximas seções é mais detalhada no anexo 3.

5.3.2 Modelagem das entidades

Tomando os objetos da classe ENTIDADE, que reúne as três grandes classes identificadas no capítulo 3, e observando os modelos das entidades dos casos selecionados (vide anexo 1), a classe ENTIDADE foi modelada como segue (figura 5.15). Neste modelo, são listadas apenas as operações mais relevantes. Nos demais modelos apresentados, para evitar sobrecarga de informações, optou-se por não incluir as operações, que estão, entretanto, listadas no anexo 3.

Sistemas de múltiplas-entidades, como o próprio nome diz, são compostos por objetos da classe ENTIDADE-COMPONENTE, que é uma generalização de sub-sistemas

e entidades primitivas. Sub-sistemas são modelados com a classe SUB-SISTEMA, que é uma agregação de ENTIDADE-COMPONENTE, numa modelagem recursiva. Uma ENTIDADE-COMPONENTE pode ser uma ENTIDADE-PRIMITIVA, suportando, dessa forma, a modelagem de sistemas com um nível hierárquico apenas. As entidades componentes mantêm uma relação de comunicação de eventos. ENTIDADE-COMPONENTE também pode ter atributos globais (Dados-característicos). Não estão aqui representados os atributos de sub-sistemas e entidades primitivas.

Objetos manufaturáveis são agregações de objetos componentes. Cada OBJETO-COMPONENTE é descrito geométrica e analiticamente, num espaço-domínio cuja dimensão é atributo do objeto manufaturável. A descrição geométrica é modelada através de uma classe ELEMENTO-GEOMÉTRICO; a descrição analítica corresponde a atributos globais (Dados-característicos) e a atributos dos elementos deste espaço-domínio.

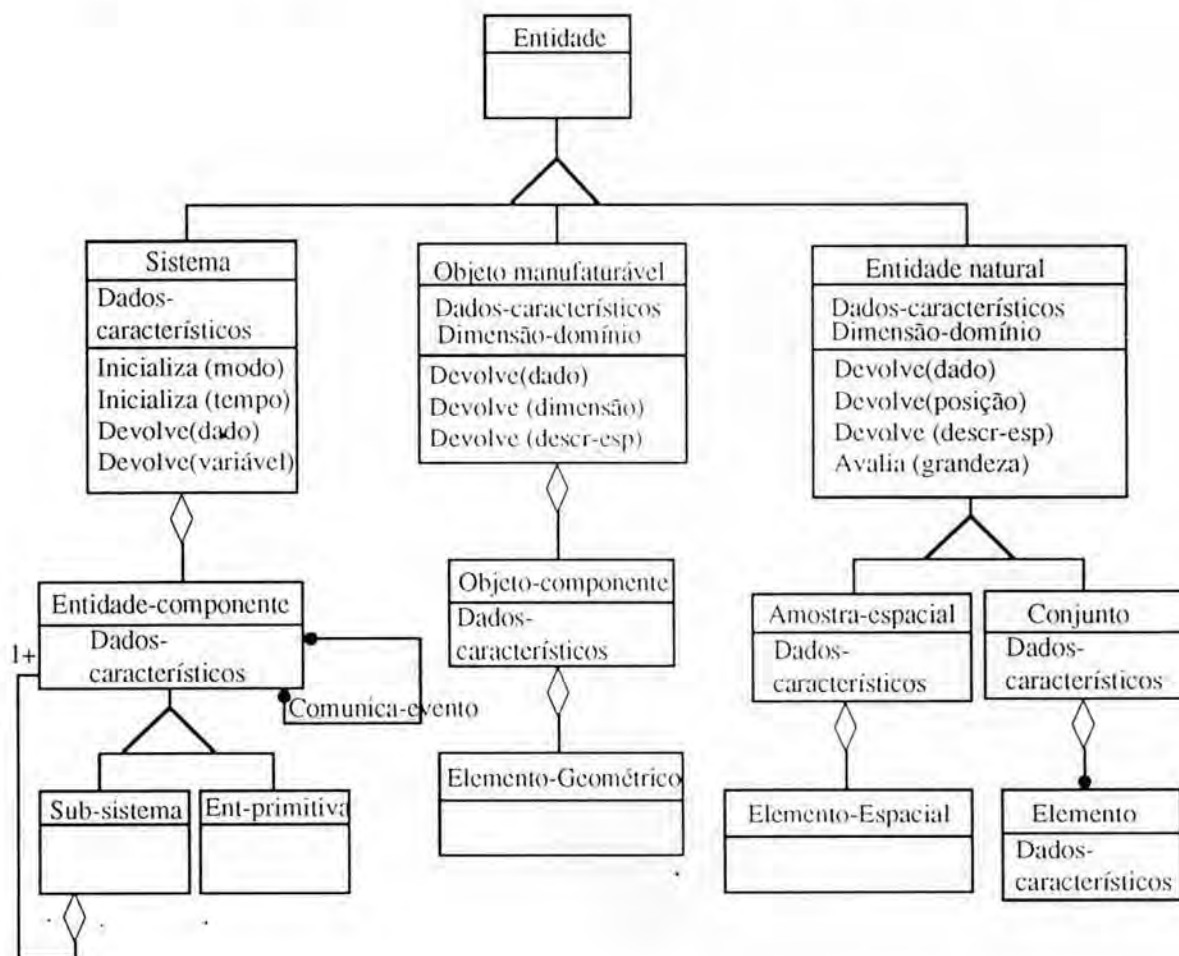


Figura 5.15 - Modelo da classe ENTIDADE.

A classe ENTIDADE/FENÔMENO NATURAL é generalização de duas classes: AMOSTRA-ESPACIAL e CONJUNTO, que representam, respectivamente, dados associados a pontos num espaço qualquer e dados associados a entes de algum conjunto. AMOSTRA-ESPACIAL tem um atributo global que representa a dimensão do espaço-domínio e é uma agregação de objetos da classe ELEMENTO-ESPACIAL. CONJUNTO não tem dimensão de domínio associado explicitamente, pois esta é implícita no número de características de cada ELEMENTO do conjunto. Entretanto, uma entidade CONJUNTO pode ter Dados-característicos que representam um atributo global de todos os elementos.

As classes ELEMENTO-GEOMÉTRICO (fig. 5.16) e ELEMENTO-ESPACIAL (fig. 5.17) representam o espaço-domínio de definição dos atributos das entidades OBJETO-COMPONENTE e AMOSTRA-ESPACIAL (fig. 5.15), respectivamente.

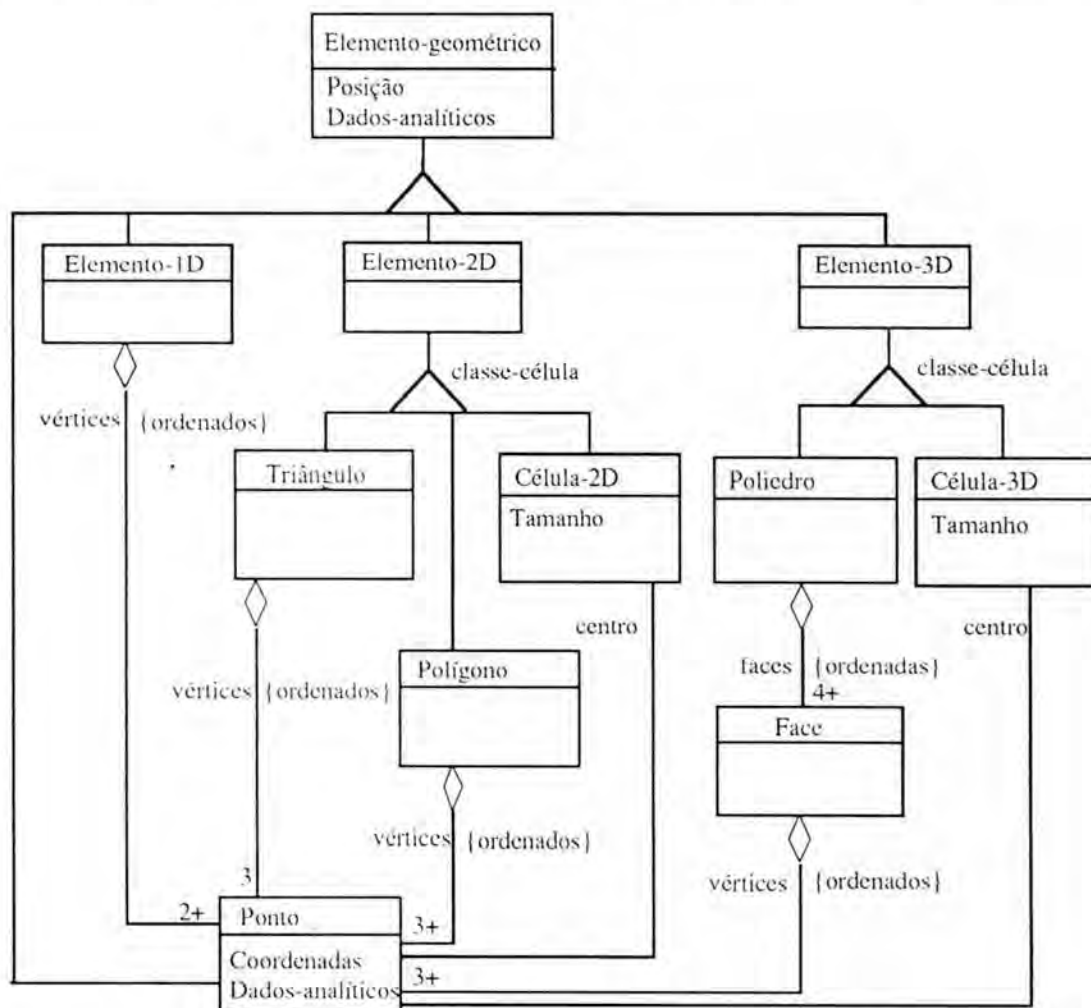


Figura 5.16 - Modelo da classe ELEMENTO-GEOMÉTRICO.

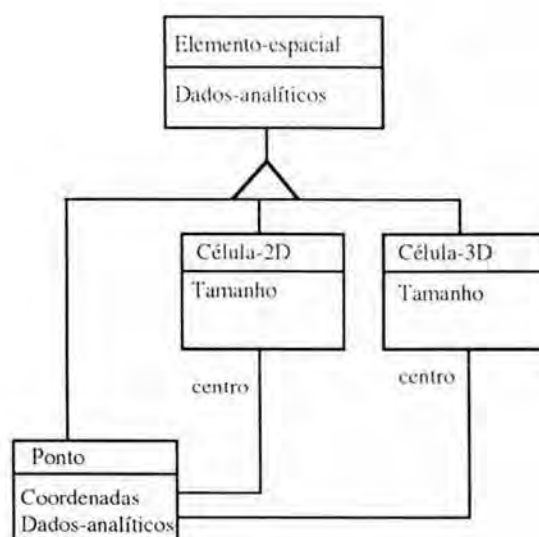


Figura 5.17 - Modelo da classe ELEMENTO-ESPACIAL.

Um objeto ELEMENTO-GEOMÉTRICO é uma generalização de 4 sub-classes: PONTO, ELEMENTO-1D, ELEMENTO-2D e ELEMENTO-3D. PONTO é o objeto mais primitivo desta descrição; um OBJETO-COMPONENTE pode ser uma agregação de pontos, simplesmente, permitindo modelar o espaço-domínio constituído por pontos discretizados. A classe ELEMENTO-1D modela entidades definidas espacialmente como elementos lineares. Já a classe ELEMENTO-2D é uma generalização de três sub-classes, destinadas a modelar células triangulares (TRIÂNGULO), células quadriláteras (CÉLULA-2D) e polígonos (POLÍGONO). A classe ELEMENTO-3D modela sólidos poliédricos (POLIEDROS); células hexaédricas (CÉLULA-3D) definidas por uma posição central e um tamanho são consideradas à parte em função de sua larga aplicação. Cabe observar que todas as classes, em seu último nível, têm a classe PONTO como componente primitivo.

Cada ELEMENTO-GEOMÉTRICO possui atributos (Dados-analíticos) que, neste caso, devem ser considerados como estando associados ao baricentro do elemento. Os Dados-analíticos correspondem aos diferentes atributos já mencionados, características, dados escalares e dados vetoriais. Eventualmente, estes atributos podem estar associados aos pontos do espaço-domínio, numa granularidade mais fina.

A classe ELEMENTO-ESPACIAL representa a descrição espacial (geométrica, temporal, espectral) de dados amostrados de forma regular ou irregular no espaço-

domínio de definição da entidade em estudo. É, portanto, uma generalização de PONTO, CÉLULA-2D e CÉLULA-3D. Os atributos Dados-analíticos reúnem variáveis dos tipos de dados escalar, vetorial e característica. Deve-se observar que não há necessidade de que os três tipos de atributos estejam presentes para a descrição analítica de um objeto.

Considerando a informações modeladas na classe AMOSTRA-ESPACIAL, a partir de elementos espaciais pode-se derivar elementos geométricos, quando estes se fizerem necessários para uma avaliação ou para mapeamento para um determinada representação visual.

5.3.3 Modelagem de representações visuais e mapeamento

A classe REPRESENTAÇÃO-VISUAL é uma generalização de seis sub-classes: ICONE, DIAGRAMA, GRÁFICO, TABELA, OBJETO-GEOMÉTRICO e MAPA (fig. 5.18).

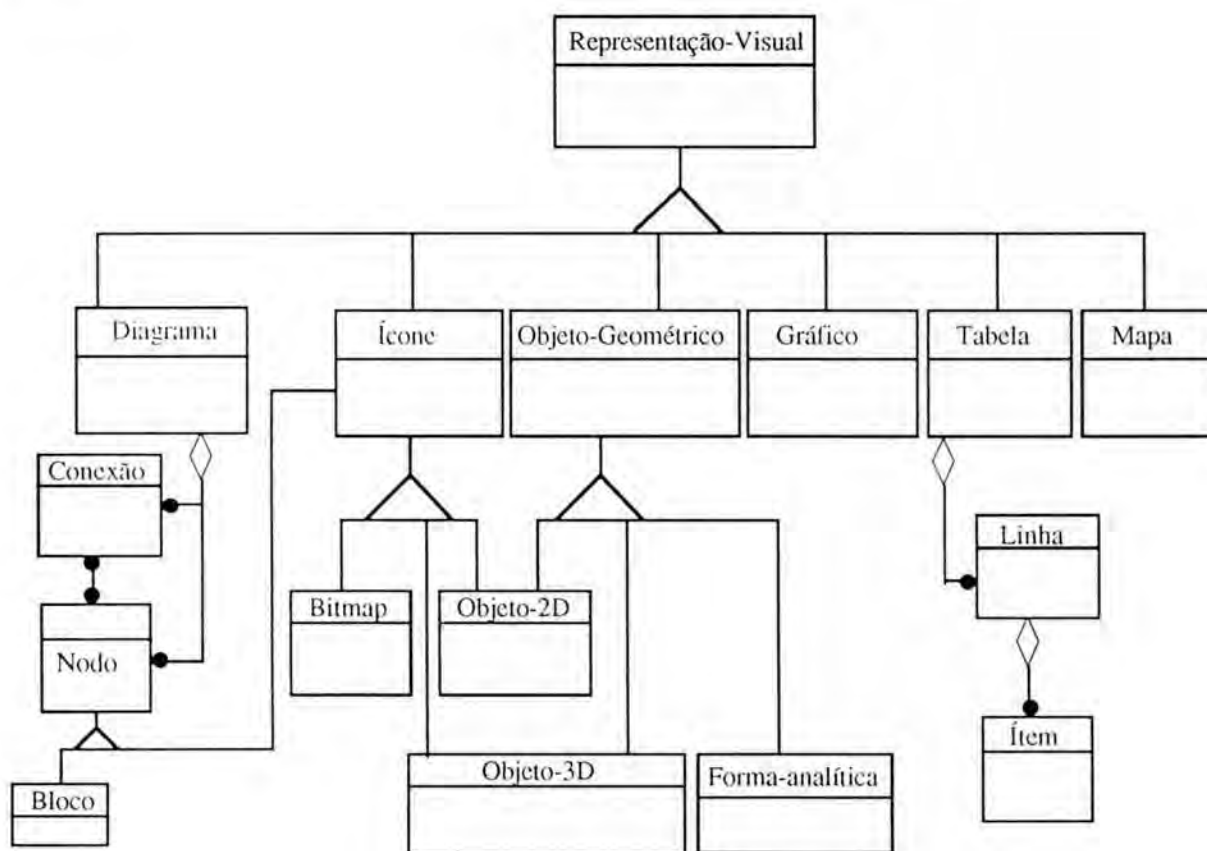


Figura 5.18 - Modelo da classe REPRESENTAÇÃO-VISUAL.

A classe ÍCONE é também uma generalização de sub-classes que correspondem a recursos para a exibição de objetos na forma "bitmap", objetos geométricos 2D e objetos geométricos 3D. A classe DIAGRAMA reúne os recursos necessários à construção de diagramas de nodos e conexões, onde os nodos podem ser blocos simples ou ícones. A classe GRÁFICO (ver figura 5.19) modela os tipos mais usuais de gráficos: GRÁFICO-2D, GRÁFICO-3D, GRÁFICO-SETORES, GRÁFICO-BARRAS e HISTOGRAMA. TABELA é uma classe para a representação visual de relações na forma tabular clássica, com uma composição básica de objetos das classes LINHA e ÍTEM. A classe OBJETO-GEOMÉTRICO corresponde aos recursos clássicos de exibição de modelos geométricos 2D e 3D, com técnicas de acabamento de faces, para o caso 3D, e incluindo formas analíticas. A classe MAPA (ver figura 5.20) é bastante complexa, reunindo as técnicas de visualização de dados de características, campos escalares e vetoriais sobre diferentes domínios geométricos.

As classes GRÁFICO e MAPA representadas separadamente congregam os recursos para a exibição de gráficos diversos e para a exibição de dados com as técnicas de visualização consagradas para dados dos diversos tipos. GRÁFICO é uma generalização de classes para a geração de gráficos bidimensionais, tridimensionais, de setores, de barras e histogramas. Já a classe MAPA representa os recursos para a geração de mapas de cores falsas, mapas de ícones, linhas de contorno, isosuperfícies, para grandezas escalares, e linhas de trajetória e trajetória de ícones, para representação de campos vetoriais.

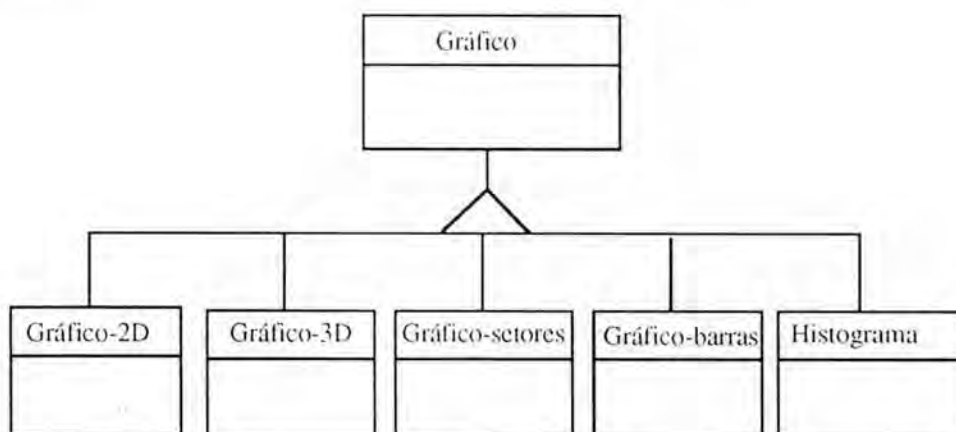


Figura 5.19 - Modelo da classe GRÁFICO.

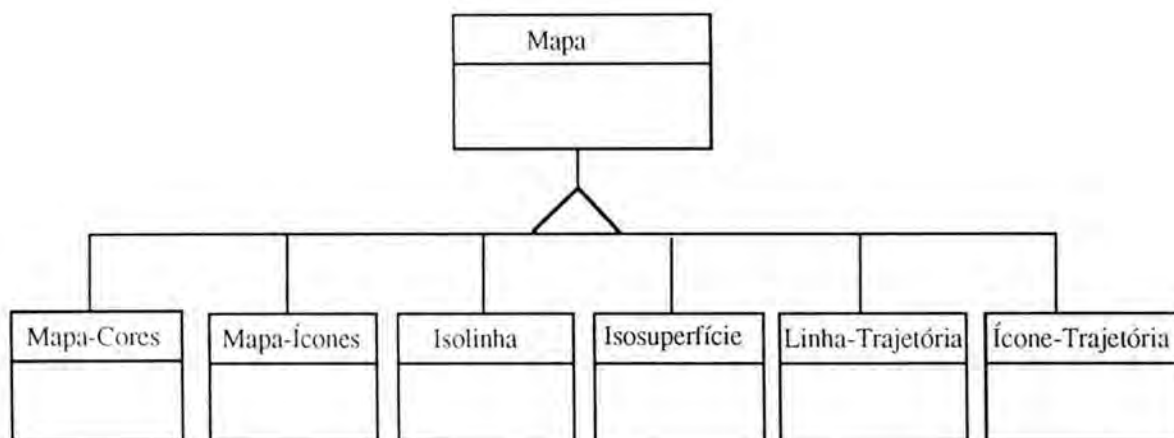


Figura 5.20 - Modelo da classe MAPA.

Relacionando entidades (e, por conseguinte, seus atributos) e representações visuais está definida a ferramenta **mapeamento**. Esta ferramenta permite associar uma instância de uma representação visual a uma entidade ou atributo de entidade. Dado que esta ferramenta é, na realidade, um relacionamento dinâmico, ou seja, estabelecido e eliminado, requerendo parâmetros específicos, optou-se por modelá-la como uma classe (fig. 5.21). As várias representações visuais são obtidas a partir de sub-classes da classe MAPEAMENTO, pois há atributos diferentes, particulares para cada tipo de representação. Observe-se que a classe GRÁFICO-2D (fig. 5.19), por exemplo, é responsável pela geração de gráficos bidimensionais. A ela devem ser informadas variáveis (atributos ou grandezas) para cada um dos eixos, legendas e outros atributos visuais. A sub-classe MAP-GRÁFICO é responsável pela obtenção dessas informações do usuário e pela manutenção do relacionamento entre a entidade selecionada no modelo e um objeto GRÁFICO-2D. O mapeamento para uma representação visual baseada em ícones requer a especificação de quais atributos são mapeados para quais características dos ícones. No mapeamento de uma entidade da classe CONJUNTO para um gráfico, há que se determinar a sub-classe do gráfico, a associação atributo-eixo, etc. Desta forma, as sub-classes da classe MAPEAMENTO realizam toda a parametrização das representações visuais, sendo responsáveis pelo acesso aos atributos das entidades e acionamento dos métodos das representações visuais.

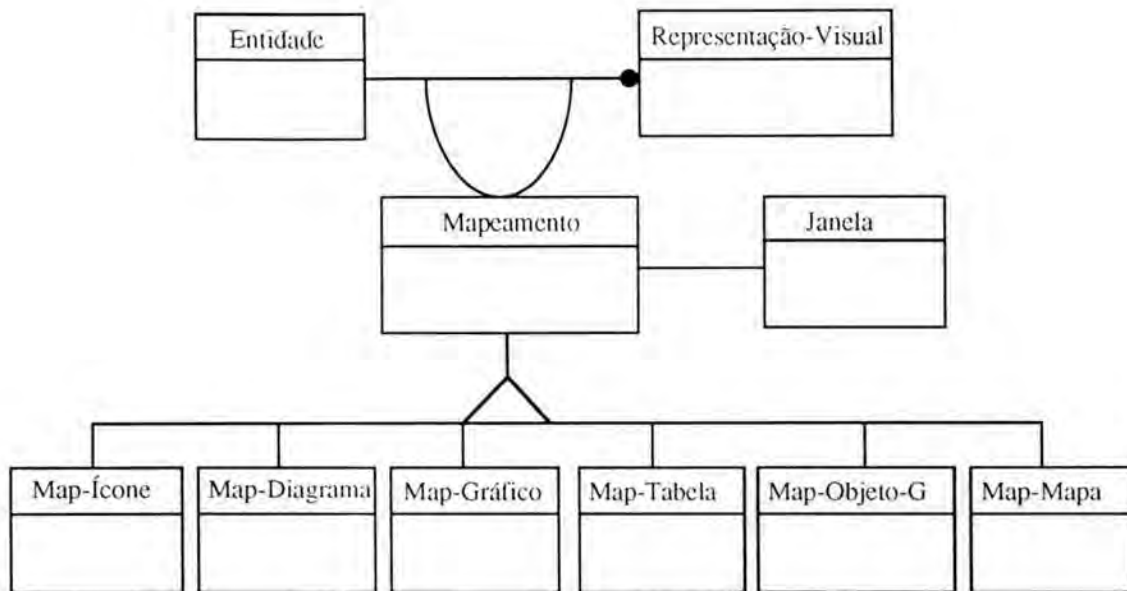


Figura 5.21 - Modelo da classe MAPEAMENTO.

5.3.4 Modelagem das ferramentas

As ferramentas necessárias à análise exploratória e simulação interativa visual já foram identificadas no capítulo 4. A ferramenta de mapeamento, por sua relação com representações visuais, foi descrita na seção anterior. As demais ferramentas também são modeladas como classes de objetos que são instanciadas ao serem aplicadas sobre entidades.

As ferramentas de navegação são modeladas como sub-classes de uma classe NAVEGADOR (fig. 5.22). CURSOR provê movimentação e armazena posição corrente ou objeto corrente, dependendo da natureza do domínio da entidade representada visualmente. FOCALIZADOR permite ajustar a janela sobre entidades descritas no espaço 2D, enquanto CÂMERA corresponde à movimentação de uma câmera sintética sobre objetos 3D. Como as ferramentas de navegação alteram o centro de interesse do observador, a alteração de seus parâmetros implica na atualização da imagem exibida na janela base da interação.

Da mesma forma, as ferramentas de seleção são sub-classes de uma classe SELETOR (fig. 5.23). A classe SELETOR-INDIVIDUAL provê os recursos de indicação de uma ou mais entidades dentre um conjunto, através do CURSOR. Os

diversos seletores espaciais são também modelados como classes: SELETOR-PONTO permite selecionar um ponto num domínio qualquer; SELETOR-LINHA permite indicar uma linha num espaço-domínio qualquer, a partir de dois pontos; SELETOR-JANELA atua sobre domínios 1D ou 2D, selecionando uma sub-região; SELETOR-VOLUME permite indicar um sub-volume através de um envelope volumétrico ajustável; SELETOR-CORTE atua sobre domínio 3D, para selecionar um plano. O SELETOR-LÓGICO utiliza uma expressão (EXPRESSÃO) lógica sobre os atributos das entidades para determinar o conjunto ou região de interesse.

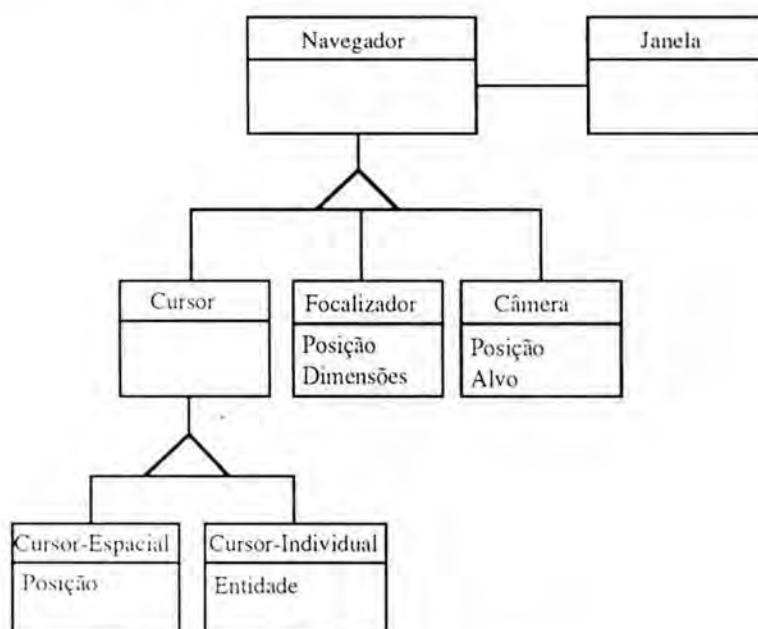


Figura 5.22 - Modelo da classe NAVEGADOR.

A classe CONSULTA (figura 5.24) representa as ferramentas de leitura e medição de atributos. LEITOR é uma classe que provê acesso imediato aos valores de atributos tipo característica presentes no modelo. Está associada tanto à classe SELETOR-INDIVIDUAL como à SELETOR-ESPACIAL, pois a entidade é selecionada e seus atributos se tornam acessíveis. MEDIDOR-ESCALAR e MEDIDOR-VETORIAL são classes destinadas à consulta a valores de atributos escalares e vetoriais no sub-espaço onde estão definidos. Esta consulta pode envolver, portanto, interpolação dos valores conhecidos e representados no modelo como parte dos objetos ELEMENTO-GEOMÉTRICO e ELEMENTO-ESPACIAL. Estas duas classes estão associadas a SELETOR-ESPACIAL, dependendo do espaço-domínio de definição dos atributos.

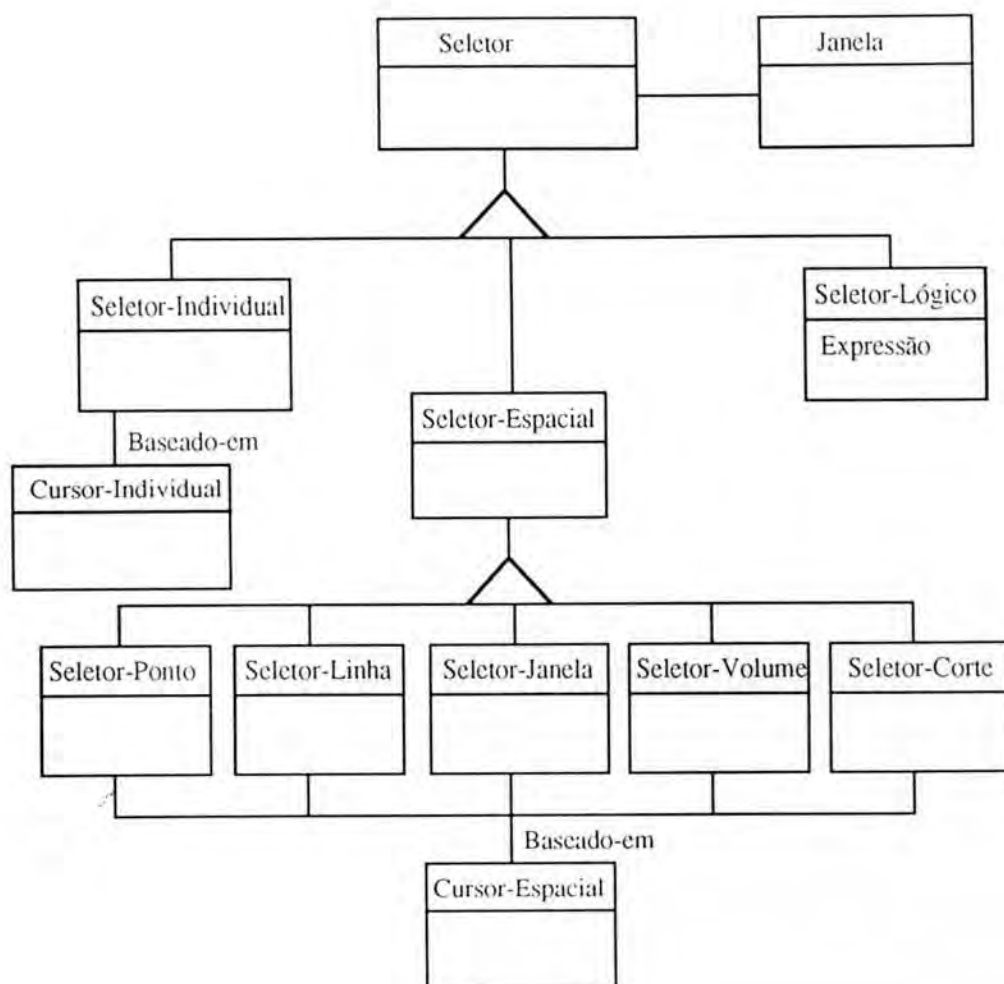


Figura 5.23 - Modelo da classe SELETOR.

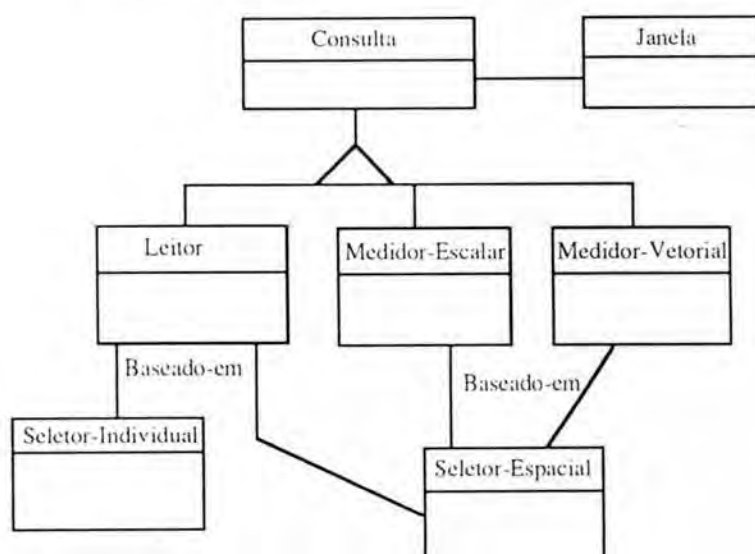


Figura 5.24 - Modelo da classe CONSULTA.

As ferramentas de processamento e controle são modeladas como classes independentes. A classe AVALIADOR permite derivar variáveis a partir de atributos existentes, através de uma expressão ou função. Da mesma forma que a classe CONSULTA, a classe AVALIADOR atua sobre o resultado de uma seleção espacial. A classe ATRIBUIÇÃO permite inicializar variáveis com valores determinados, com vistas à realização de algum processamento ou, ainda, inicializar variáveis com constantes. A classe EXPERIMENTO provê as facilidades de especificação de pontos de parada, especificação de opções de execução, indicação de atributos a serem registrados em arquivos e disparo do experimento propriamente dito. Finalmente, a classe REGISTRO implementa as facilidades de legendas em imagens já exibidas como resultado de mapeamentos e funções de gravação e recuperação de imagens de arquivos em disco. Como as classes descritas na seção anterior, estas estão detalhadas no anexo 3.

5.3.5 Modelo dinâmico

Embora os aspectos interativos da plataforma tenham sido abordados na seção 5.2, por ocasião da apresentação da linguagem visual, o modelo dinâmico deve ser construído para formalizá-los. Para evitar detalhes desnecessários, não serão abordados os estados e eventos usuais de um ambiente com interface do estilo manipulação direta, nem todos os estados e eventos considerando a operação de todas as ferramentas definidas. Exemplifica-se o modelo dinâmico através dos diagramas de estados da classe SELETOR-LÓGICO (fig. 5.25a) e da classe MAP-GRÁFICO (fig. 5.25b). Considere-se, além disso, que estas ferramentas estão sendo aplicadas a uma entidade da classe ENTIDADE NATURAL.

Na figura 5.25a pode-se observar que as mensagens enviadas por esta classe (SELETOR-LÓGICO) à entidade sobre a qual foi aplicado o **seletor**, correspondem a eventos. Ao arrastar-se o ícone do **seletor lógico** sobre uma entidade, a instância de SELETOR-LÓGICO assim criada recebe o nome da entidade a ser operada. SELETOR-LÓGICO requisita o nome dos atributos da entidade através da chamada ao método apropriado, aguarda que o usuário indique os atributos considerados como critérios de seleção e aguarda que a expressão seja construída. De posse da expressão, esta é analisada e, estando corretamente especificada, constrói o subconjunto de entidades que satisfazem à condição. Durante esta última atividade (representada no último estado do

diagrama), SELETOR-LÓGICO aciona novamente a entidade, buscando os elementos do conjunto e verificando quais devem constar no resultado da seleção.

No exemplo da figura 5.25b, o procedimento inicial é análogo. Após a obtenção dos nomes dos atributos da entidade, a ferramenta MAP-GRÁFICO aguarda os parâmetros para o mapeamento, ou seja, dimensão do gráfico, tamanho de janela, correspondência eixo-atributo, legendas, etc. Após a parametrização da representação visual, MAP-GRÁFICO declara explicitamente a vinculação entre a entidade e a representação GRÁFICO, mantendo os parâmetros e chamando os métodos apropriados para a busca de valores de atributos das entidades e geração do gráfico propriamente dito.

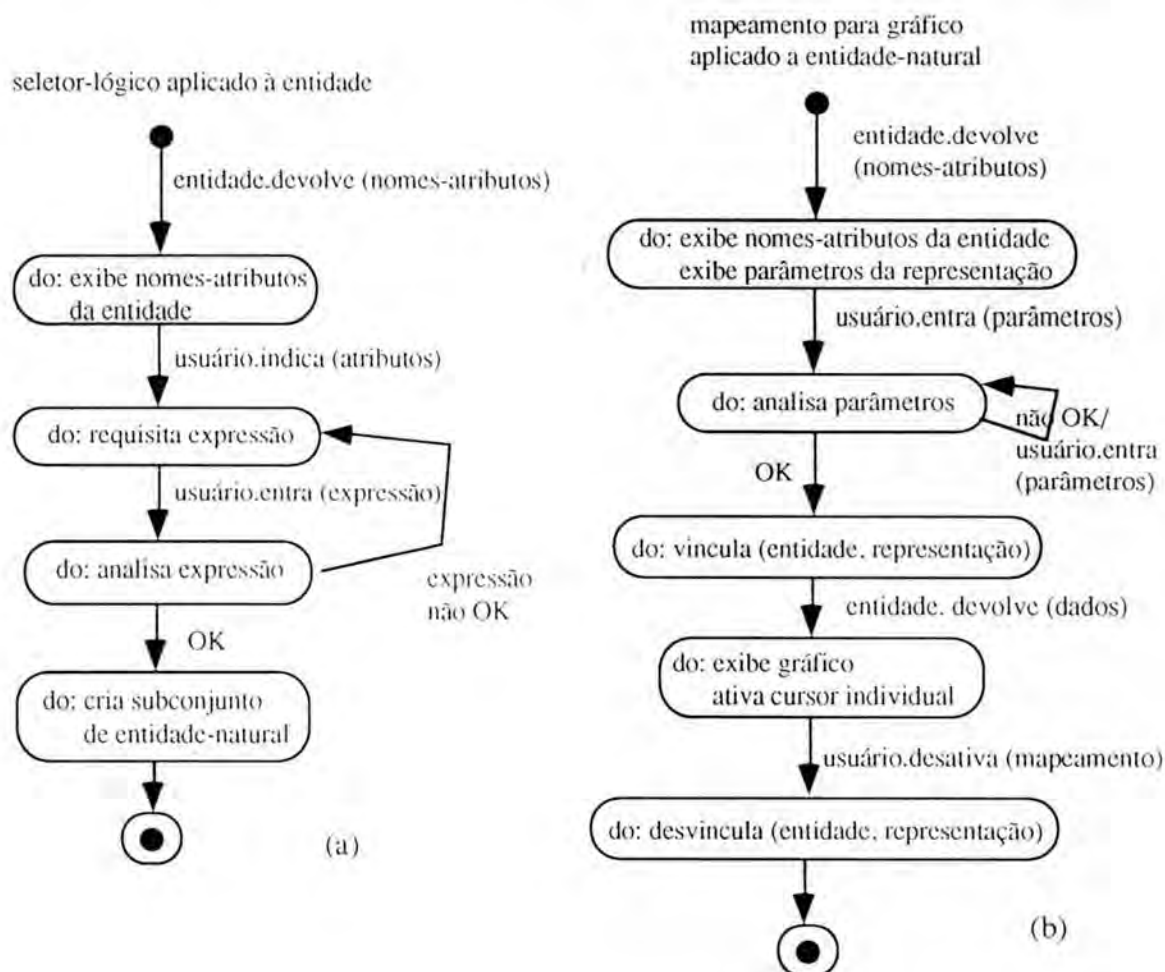


Figura 5.25 - Diagramas de estado para as classes (a) SELETOR-LÓGICO e (b) MAP-GRÁFICO.

Os diagramas acima não mostram a aplicação das ferramentas a entidades de outras classes. A generalização destes diagramas é realizada a partir do modelo dinâmico da classe genérica SELETOR, que verifica a classe da entidade a qual está sendo aplicada a ferramenta em questão e passa a se comportar de acordo ela.

6 Discussão

VISTA é comparada com outras soluções descritas na literatura em relação aos critérios de adequação e facilidade de uso, flexibilidade, consistência, extensibilidade e robustez. São feitas considerações sobre a implementação das ferramentas utilizando como base *software* de diferentes categorias.

6.1 Comparação com outras soluções

6.1.1 Critérios de comparação

A plataforma proposta nesta tese é destinada a atender duas classes de usuários: usuários-projetistas, que necessitam um conjunto de recursos de alto nível a partir do qual possam desenvolver aplicações específicas, e usuários-finais que necessitam um sistema no qual possam estudar e solucionar seus problemas. Os usuários-projetistas terão acesso aos objetos que compõem a plataforma de forma a poder criar novos tipos de objetos para suas aplicações, quando for o caso. Os usuários-finais podem operar o sistema, sem modificações, a partir da linguagem visual oferecida. A questão mais importante para esta classe de usuários é como modelar seus dados no ambiente e se este dispõe das ferramentas de análise adequadas.

Do ponto de vista de usuários-projetistas, portanto, a comparação deve ser feita com bibliotecas gráficas, construtores de aplicativos e com linguagens de programação. Considerando usuários-finais, a comparação deve ser feita com sistemas *turn-key*.

Para comparação foram escolhidos os sistemas disponíveis na UFRGS e os sistemas descritos na literatura mais recentemente, incluindo sistemas orientados a objetos. Os sistemas escolhidos são representativos das classes de software mencionadas acima.

6.1.2 Ponto de vista do usuário-projetista

O sistema *Doré* (*Dynamic Object Rendering Environment*) [KAPLAN 1991], por exemplo, é uma interface gráfica comparável ao *GKS*, no sentido de que constitui um núcleo de funções gráficas 3D, destinado ao desenvolvimento de aplicações interativas, com geração de imagens dinâmicas e de alta qualidade. *Doré* é orientado a objetos, tanto em arquitetura como em implementação. Consiste de três camadas, sendo a mais interna o *núcleo de objetos*, a intermediária o *núcleo funcional* e a mais externa a *camada de ligação*. O *núcleo de objetos* é responsável pela alocação, execução e destruição de objetos, de forma que um usuário pode criar novos tipos e instalá-los dinamicamente, sem acesso ao código fonte do sistema. O *núcleo funcional* contém todos os tipos de objetos instalados no sistema e a parte comum de código usada pelos métodos dos objetos. A última camada, a *de ligação*, permite que múltiplas interfaces sejam definidas para o sistema, tanto em termos de linguagem de programação, como em funcionalidades. Isto significa que aplicações diferentes podem usar subconjuntos diferentes de funcionalidades *Doré*.

O objetivo básico do uso de *Doré* é a obtenção da imagem de uma cena. Nesta cena estão os objetos primitivos, câmera e luzes; esta coleção de informações - um conjunto de objetos - compõe o banco de dados da cena. Objetos primitivos são totalmente definidos por atributos, os quais podem ser *locais*, ou seja, fazerem parte do próprio objeto primitivo, ou *globais* ou *herdados*, que são dependentes do estado do ambiente. Estes atributos globais são alteráveis por objetos *atributos*.

A forma como o sistema pode ser utilizado por programadores é bastante diversificada uma vez que a camada de ligação é modular e, além disso, o sistema oferece interfaces de mais baixo nível, permitindo a conexão de procedimentos específicos de geração de imagens (*renderers*), criados pelo programador, e de dispositivos de exibição diferentes.

Comparativamente, tanto *Doré* como *VISTA* têm arquitetura orientada a objetos. Considerando as facilidades oferecidas, os objetos definidos em *VISTA* são de mais alto nível e apresentam uma semântica mais próxima das possíveis aplicações, englobando tanto atributos geométricos como analíticos. Sua aplicabilidade é geral, com esforço menor do que *Doré* exige pois neste os objetos são atores cujas imagens devem ser geradas. Por fim, a diferença fundamental reside no amplo suporte que *VISTA* provê às

tarefas de exploração, análise e processamento, não existentes em *Doré*. Considerando adequação e facilidade de uso por parte de um usuário-projetista os dois sistemas são bastante diferentes pois *Doré* é uma biblioteca, enquanto *VISTA* é um ambiente que pode ser especializado (ou configurado) para uma aplicação. Do ponto de vista de extensibilidade ambos permitem a especificação de novos tipos de objetos, mas *VISTA* permite que funções sejam incorporadas ao ambiente existente, enquanto *Doré* requer a programação de uma aplicação para abrigar a nova função. *Doré* é flexível no sentido de que fornece praticamente todos os recursos necessários à geração de imagens, o que permite atender exigências diversificadas de programadores. As necessidades relativas à interação, entretanto, não são providas diretamente pela biblioteca, carecendo de programação. Em relação à consistência, no caso de uma biblioteca, esta característica poderia ser analisada através do comportamento das primitivas; neste sentido cabe mencionar que há uma definição clara do papel das diferentes classes de objetos.

Considerando a classe dos construtores de aplicativos, analisaremos o sistema *Iris Explorer* [SILICON 1991], que é dotado de uma linguagem de programação visual para a construção dos aplicativos. A abordagem seguida nesta linguagem é a *fluxo de dados*, ou diagramática (vide classificação na seção 3.3); as aplicações são construídas utilizando um editor de diagramas (*map editor*), onde módulos são interligados e se comunicam através de *portas* de entrada e saída. A arquitetura do *Iris Explorer* é também em camadas, com as duas mais inferiores invisíveis ao programador: a *camada de sistema* é constituída por bibliotecas gráficas e sistema de janelas; a *camada de suporte à programação* contém bibliotecas para criação de estruturas geométricas. Três camadas superiores - o *nível de módulos*, o *nível de mapas* e o *nível de aplicação* - fornecem os recursos para a construção de módulos e das aplicações. Um módulo é uma função que recebe dados através de portas de entrada e parâmetros através de um painel de controle, realiza seu processamento e passa os dados para outros módulos, através de suas portas de saída. Módulos são conectados formando mapas; eventualmente, se os mapas se tornam grandes, alguns módulos podem ser agrupados num único painel de controle, formando grupos. Assim, cada grupo passa a corresponder a um ícone de módulo no mapa. Um mapa e seu painel de controle podem ser salvos como uma aplicação; neste caso, a aplicação é executada sem o editor de mapas associado, ou seja, como uma aplicação convencional.

Módulos são construídos por uma ferramenta do *Iris Explorer* (*Module Builder*), que é um gerador interativo de código. Através de janelas específicas são informados

nome do módulo a ser criado, nome da função, *includes*, bibliotecas, portas de entrada, portas de saída, argumentos da função, associação entre os dados de entrada e os argumentos da função, enfim, todas as informações necessárias para a incorporação de uma função ao sistema como um módulo. Para que os dados do usuário possam ser processados pelos módulos, uma ferramenta denominada *DataScribe* é utilizada para gerar módulos que convertam os dados do formato do usuário para os tipos que o *Iris Explorer* suporta os quais são bastante abstratos e de alto nível; por exemplo, o tipo *lattice* é uma matriz multidimensional regularmente estruturada que pode armazenar dados escalares e vetoriais. O *DataScribe* é uma ferramenta interativa, baseada num *template*, onde é descrito o formato de entrada e o formato de saída.

O editor de mapas (*Map Editor*) apresenta a área básica de trabalho do usuário do *Iris Explorer*. Módulos previamente preparados são escolhidos de uma lista e posicionados na janela do editor de mapas. Módulos são conectados, respeitando os tipos de dados associados às portas. Em geral, o mapa contém um módulo *Render* que apresenta a imagem final. Os painéis de controle e janelas associadas aos módulos podem ser minimizados e maximizados em tamanho de modo a melhor aproveitar a área física. A execução de uma mapa é acompanhada na janela do editor de mapas pela mudança de cor dos módulos à medida em que são executados.

Comparando *VISTA* e *Iris Explorer*, ambos se propõem a prover um ambiente onde o usuário-projetista possa desenvolver novas técnicas e trabalhar com novos tipos de dados. *Iris Explorer* provê de antemão mais facilidades para construção de módulos e conversores de formatos. Entretanto, não tem a gama de recursos de exploração aqui previstos. A linguagem de programação visual é bastante simples e poderosa; no entanto, da mesma forma que em *VISTA*, as funções devem ter sido previamente preparadas para serem ali utilizadas. No caso desta plataforma, as ferramentas implementadas são simplesmente acrescentadas à base de ferramentas e o usuário passa a usá-las, não estando amarrado a uma seqüência pré-estabelecida de execução.

Embora sua estrutura seja modular, *Iris Explorer* não é orientado a objetos. Neste caso, a reusabilidade de módulos que implementam alguma função é limitada ao uso do código fonte como num ambiente convencional. Já *VISTA*, sendo orientada a objetos, permite que o usuário-projetista utilize as classes de ferramentas para implementar novas. Quanto à forma como os dados podem ser introduzidos no sistema, *Iris Explorer* provê uma ferramenta específica que permite converter dados brutos em dados descritos em um

de seus formatos pré-definidos. *VISTA* exige que os dados, ou seja, as entidades, já estejam descritas de forma orientada a objetos, ou seja, de acordo com a modelagem apresentada na seção 5.3.2. A especificação de formas de modelagem adequadas a cada classe de entidade deve ser objeto de diferentes trabalhos futuros. Entretanto, considerando que estas técnicas de modelagem serão ferramentas, como as de exploração e análise, *VISTA* provê todo o arcabouço necessário para tê-las incorporadas.

Observando os quesitos de adequação e facilidade de uso, pode-se dizer que no que tange a facilidades de desenvolvimento de aplicativos, atualmente, o *Iris Explorer* é superior à *VISTA* porque fornece as ferramentas para geração de conversores de dados e incorporação de módulos ao sistema. Esta última, entretanto, é superior no conjunto coerente, consistente e, principalmente, independente, de facilidades de exploração que oferece. No *Iris Explorer* estas facilidades devem ser programadas em módulos específicos e acionadas dentro de um fluxo de execução fixo. Do ponto de vista de extensibilidade, ambos os sistemas têm o mesmo potencial, embora como já tenha sido dito, a reusabilidade inerente à abordagem OO represente um ganho substancial no tempo de desenvolvimento por parte do usuário-projetista. Flexibilidade do ponto de vista de um usuário-projetista está relacionada com a facilidade de desenvolvimento de aplicações no ambiente. *Iris Explorer* é flexível, porque permite o desenvolvimento de aplicações variadas; é também consistente, em função de sua estrutura baseada em módulos pré-definidos, cujo comportamento é previsível.

Outras linguagens específicas para desenvolvimento de aplicações de análise exploratória e simulação interativa visual ficam restritas às linguagens de programação convencionais com bibliotecas gráficas, às linguagens de simulação também com bibliotecas gráficas e às linguagens de comandos que ativam funções gráficas interativamente. Da classe de linguagens de simulação com bibliotecas gráficas selecionamos *MODSIM II* [BELANGER 1990; BELANGER 1991] que é orientada a objetos e cujos programas podem conter chamadas às funções de *SIMGRAPHICS* [CACI 1991], pacote de funções que permite a geração de ícones, gráficos e interfaces baseadas em menus, para aplicações de *Simscrip II.5* [CACI 1983]. Da classe de linguagens de comando selecionamos *IDL* [RESEARCH SYSTEMS 1992], que é tanto uma linguagem interativa, interpretada, que pode ser usada da linha de comandos, como uma linguagem de programação estruturada, que pode ser usada para a construção de aplicações, envolvendo processamento, interação com dados, visualização e processamento de imagens.

MODSIM II, linguagem de programação de propósitos gerais, orientada a objetos, baseada em *Modula-2*, é uma linguagem compilada, gerando código C, o que garante sua portabilidade para diferentes plataformas. As construções orientadas a objetos incluem herança simples e múltipla, ligação dinâmica de objetos, polimorfismo e encapsulamento de dados. As construções opcionais que suportam simulação são baseadas em *Simscrip* *II.5*. É uma linguagem fortemente tipificada, suportando tipos definidos pelo usuário. Exibição e interação são garantidas com o uso de *SIMGRAPHICS*. Ícones, gráficos e cardápios podem ser criados neste ambiente e associados a objetos e variáveis do programa. Ícones e gráficos são atualizados dinamicamente durante a execução do programa de simulação. A interação do usuário com o modelo de simulação deve ser programada, o que equivale a prever, no modelo de simulação, todas as interações possíveis que o usuário pode necessitar ou desejar. Esta é a maior limitação das linguagens de simulação que incorporam alguma biblioteca gráfica; a exploração deve ser pré-programada no próprio modelo de simulação. Desta forma *MODSIM II* e *VISTA* oferecem facilidades de nível bastante diferenciado a usuários-projetistas. Enquanto em *MODSIM II* aplicações devem ser construídas a partir de recursos básicos de uma linguagem, utilizando recursos típicos de interfaces gráficas (cardápios, caixas de diálogo, botões) acrescidos de facilidades para geração de gráficos, em *VISTA* o usuário-projetista já trabalha a partir de recursos de análise de mais alto nível, modelados como classes básicas do ambiente. Um aspecto importante, já mencionado, que deve ser ressaltado é o fato de em *MODSIM II*, tanto a interação como as representações visuais devem estar programadas no modelo de simulação. Em *VISTA*, estes aspectos estão desvinculados do modelo e associados à fase de experimentação, cujos recursos de controle residem na plataforma.

IDL (Interactive Data Language) é comercializada como um pacote para análise e visualização interativa de dados científicos e de engenharia, integrando funções diversas de análise numérica, processamento de imagens, estatística e visualização. É uma linguagem de programação interpretada e orientada a *arrays*. Variáveis podem ser declaradas e tipificadas, mas podem ter seu tipo redeclarado a qualquer momento o que é inconveniente em termos de consistência. Com sentenças simples, funções podem ser chamadas para gerar imagens diversas a partir de variáveis tipo *array*. A linguagem apresenta operações *built-in* sobre este tipo de variável, o que facilita o trabalho sobre dados multidimensionais; entretanto, toda a sua interface é baseada na linha de comandos. Para a programação de aplicações, oferece um conjunto de *widjets*, ao estilo do *Guide* [SUN MICROSYSTEMS 1989]. Para um usuário-projetista, são atraentes as funções de

visualização, processamento de imagens e, mais recentemente, funções de análise numérica bastante difundidas [PRESS et al. 1988]. Entretanto, o suporte conceitual oferecido é pobre, não há estrutura propiciando reusabilidade e todos os aspectos de modularidade e consistência são responsabilidade direta do usuário-projetista. A linguagem é, sem dúvida, flexível, pois tanto é possível realizar tarefas simples, que envolvem visualização pouco sofisticada, como tarefas mais complexas, que envolvem o processamento de dados multidimensionais. A questão do suporte à exploração é, entretanto, semelhante a outros sistemas já mencionados; além de uma câmera sintética, não existem funções pré-definidas, concebidas como suporte ao processo de análise visual dos dados. Cabe ressaltar, entretanto, que pelo fato das operações com matrizes serem um dos pontos fortes desta linguagem, o processamento deste tipo de dado fica facilitado.

6.1.3 Ponto de vista do usuário-final

Tomando, agora, uma perspectiva de usuário-final, a plataforma deve ser comparada a sistemas da classe *turn-key*: IDL, comentada acima, pode ser usada como sistema *turn-key*: entretanto, não é um sistema para usuários leigos, pois exige conhecimento de linguagens de programação. O sistema *MPGS* [CRAY RESEARCH 1991] é exemplo típico de sistema desta classe, com a particularidade de adotar uma abordagem distribuída entre o supercomputador Cray e estações gráficas Silicon Graphics. Está, portanto, associado a este tipo de plataforma, pois algumas funções que requerem alta capacidade de processamento são executadas no supercomputador. É um sistema de visualização em pós-processamento, com ênfase em aplicações de engenharia. Objetos modelados computacionalmente e submetidos a algum processamento que gere dados escalares e vetoriais associados à geometria são exibidos utilizando métodos de contorno por isolinhas e isosuperfícies, vetores e traçado de partículas. Câmera sintética e planos de corte são oferecidos como funções de navegação e seleção respectivamente, embora estes conceitos não estejam explícitos. É um sistema flexível dentro das limitações impostas pelas funções implementadas; é consistente pois dados são informados através de formatos padronizados e as funções têm seu espectro de aplicação bem determinado. Comparativamente a *VISTA*, tem a vantagem utilizar as facilidades de um supercomputador para agilizar o processamento de certas funções; entretanto, em *VISTA* esta questão é perfeitamente tratável: a abordagem orientada a objetos dá amplo suporte conceitual à execução de diferentes ferramentas de forma distribuída,

principalmente sendo a comunicação entre objetos tratada como troca de mensagens. A principal diferença entre os dois ambientes é a flexibilidade e a extensibilidade de *VISTA* que *MPGS*, por ser *turnkey*, não apresenta. *VISTA* pode ser configurada para diversas aplicações e suporta uma variedade de tipos de dados, enquanto *MPGS* requer a conversão de quaisquer dados para seus formatos fixos.

BRITAIN et al. [1990] descrevem um sistema que se aproxima de *VISTA* em termos de objetivos voltados para usuários finais. Este sistema é destinado à análise interativa de dados escalares e vetoriais 3D, seguindo uma abordagem semelhante à proposta neste trabalho. O sistema provê vários métodos de visualização implementados de forma a serem executados em plataformas de hardware heterogêneas. Isto é possível pela utilização de uma interface gráfica portátil e de um sistema de gerência de interfaces com usuário (*UIMS*). O sistema é destinado ao usuário-final e é também baseado na abordagem orientada a ferramentas. Ferramentas são objetos e são instanciadas quantas vezes for necessário para a realização de uma análise. Em termos de arquitetura, o sistema é organizado em camadas: camada de interface, implementada através do *UIMS*, e camada funcional, constituída pelo banco de dados e sua interface de acesso e pelo conjunto de ferramentas e seu gerenciador. Como ferramentas, o sistema oferece planos de corte, isosuperfície, emissão de partículas, consulta a dados pontuais, colocação de legendas em imagens e visualização primária de dados volumétricos. Comparando com *VISTA*, não há uma distinção entre ferramentas de navegação, exploração e visualização. Isosuperfície, por exemplo, é ao mesmo tempo seleção e visualização. Há dois modos de operação: o sistema provê uma câmera sintética, que pode ser manipulada geometricamente, para reposicionamento e reorientação ("modo câmera"). Quando uma ferramenta geométrica como plano de corte deve ser posicionada sobre o objeto em análise, há um chaveamento através de um botão, para o "modo ferramenta": todas as ações do usuário passam a se refletir sobre a ferramenta e não sobre a câmera. Em relação aos dados manipulados, só há previsão para dados 3D sobre *grids* e geometrias. O sistema é portátil e, pela descrição, fácil de usar e flexível, havendo a possibilidade de agrupamento de ferramentas. Neste último caso, agrupamento quer dizer poder manipular geometricamente as ferramentas em conjunto (por exemplo, vários planos de corte). Em *VISTA*, não há agrupamento de ferramentas no sentido acima; ferramentas são aplicadas em seqüência e a criação de novas só é possível por um programador, pela reutilização das existentes. *VISTA* e o sistema proposto por BRITAIN et al. [1990] diferem conceitualmente, pois neste último, representações visuais são encaradas como sinônimo de técnicas de visualização e estas são consideradas ferramentas. Em *VISTA*, ferramentas

são definidas conceitualmente e representações visuais de entidades são obtidas por mapeamento do estado interno dos objetos para atributos de representações gráficas.

6.1.4 Ponto de vista de projetistas e usuários finais

Considerando o oferecimento de dois níveis de interface, tanto para usuários programadores como para usuários-finais, encontramos sistemas construtores de aplicativos, já exemplificados através do *Iris Explorer*, representativo dos sistemas baseados na abordagem *data flow* como *AVS* [UPSON et al. 1989], *apE* [DYER 1990] e *Data Explorer* [LUCAS et al. 1992], e sistemas experimentais diversos baseados numa biblioteca de métodos que pode ser usada por um usuário-final, como sistema *turnkey*, e por programadores, como uma plataforma de desenvolvimento. Os representantes desta última classe de sistemas adotam soluções diferentes para as diversas questões que devem ser tratadas. LANG et al. [1991] concentram-se na integração de funções de processamento e de visualização num único sistema, baseando-se no uso de um banco de dados único. O sistema interpreta uma linguagem de comandos RSYST (que dá nome ao software) a qual interliga módulos de processamento e de visualização. *SuperGlue* [HULTQUIST e RAIBLE 1992] é um ambiente de programação baseado na linguagem Scheme, como linguagem de controle e programação de interfaces, e na linguagem C, como linguagem de processamento numérico e gráfico. Embora apresente primitivas específicas para a área de dinâmica de fluidos, a proposição inicial e principal do sistema é extensibilidade e reusabilidade de código, através da abordagem orientada a objetos. TURNER et al. [1990] apresentam o *Fifth Dimension Toolkit* que é um conjunto de classes destinadas a suportar a construção de aplicações gráficas (animação baseada em leis mecânicas, animação facial, deformação, etc.) 2D e 3D. As entidades modeladas, entretanto, são apenas objetos geométricos e a maioria das classes apresentadas representam objetos de interface com usuário e gerência de janelas. Embora tenha sido veiculado que o sistema suporta aplicações de visualização científica, ele está fortemente centrado em animação de atores sintéticos. SCHROEDER et al. [1992] descrevem *VISAGE*, sistema que mais se aproxima do pretendido pela plataforma. É orientado a objetos e baseado na abordagem *data flow*, apresentando duas grandes classes de objetos: dados e processos (que implementam as técnicas de visualização). Os dados são modelados como uma descrição geométrica e dados escalares e vetoriais associados. Os processos tomam dados, processam e geram geometrias destinadas à exibição por um

objeto tipo *renderer*. É, portanto, substancialmente diferente de *VISTA*, que modela as diversas ferramentas de acordo com os objetivos a que servem, chegando semanticamente mais próxima das necessidades de usuários cientistas.

Os sistemas mencionados acima são, em sua maioria, voltados para visualização de dados em geral. Considerando simulação interativa visual, os desenvolvimentos mais recentes são descritos por ROOKS [1993] e MUTKA e MCKINLEY [1993].

ROOKS [1993] desenvolve um ambiente de simulação interativa, segundo um paradigma "centrado no usuário", com interfaces consistentes entre usuário, modelo e sistema de simulação. A idéia central é retirar do modelo de simulação todos os aspectos que são, na realidade, relativos ao controle de experimentos. O projetista do modelo de simulação deve se preocupar somente com a lógica e estruturas de dados do modelo; cabe ao analista da simulação interagir com o modelo e com o sistema de simulação para realizar os experimentos adequados. O sistema proposto por ROOKS [1993] é composto de quatro subsistemas: ambiente de pré-processamento, ambiente de desenvolvimento do modelo, ambiente de análise do modelo e sistema de cenário. O ambiente de análise do modelo é o responsável por implementar as funções de simulação interativa visual através de um processador de interações e de um controlador de execução; estes utilizam os recursos preparados no sistema de cenário para permitir interação entre usuário e modelo (inspeção de dados, alteração de parâmetros, etc). ROOKS [1993] propõe, na realidade, um ambiente para criação de aplicações de simulação, possibilitando a configuração com diferentes linguagens, isto é, pode ser considerando um ambiente de desenvolvimento de simulação genérico do ponto de vista da linguagem a ser utilizada. Não há, entretanto, a possibilidade de serem analisados dados resultantes de simulações como ocorre em *VISTA*. Esta, entretanto, não oferece ferramentas de modelagem que permitam incorporar imediatamente linguagens de simulação diversas; antes, oferece um arcabouço conceitual para a modelagem.

O outro sistema analisado, *OpenSim* [MUTKA e MCKINLEY 1993], atua mais como um integrador de ferramentas para gerência e análise de projetos de simulação. Compreende quatro componentes: uma interface gráfica que permite criar arquivos de entrada para programas de simulação; uma ferramenta de análise de resultados, que permite extrair dados gerados por programas de simulação e exibí-los em gráficos, normalmente 2D; uma ferramenta de execução de experimentos de simulação em ambiente distribuído e uma ferramenta de monitoramento de execução, que permite definir

mecanismos de acompanhamento de valores de variáveis através de sua exibição dinâmica. Neste sistema, o usuário é o construtor do modelo de simulação pois toda a interação para monitoramento é realizada através dos nomes das variáveis existentes no programa de simulação. Não há previsão para interação entre sistema e um usuário de nível de abstração mais alto. *OpenSim* é substancialmente diferente de *VISTA* porque é dedicado apenas a fornecer facilidades para a criação de entradas para programas de simulação e para a análise de resultados. O tratamento dado a programas de simulação é vantajoso em relação à *VISTA* pois não há necessidade de modelagem dos objetos na forma aqui especificada. *OpenSim* perde, entretanto, nos recursos de análise de resultados de experimentos, já que permite apenas a produção de gráficos simples.

Considerando todos os sistemas comentados pode-se dizer que *VISTA*: a) foi concebida considerando a integração de ferramentas de análise exploratória e simulação interativa visual, dado que as duas áreas têm muito em comum; b) apresenta uma abordagem consistente de modelagem tanto de entidades como de ferramentas; c) é tanto voltada para usuários finais como para usuários programadores, uma vez que sua extensibilidade é garantida tanto pela abordagem orientada a ferramentas da interface como pela abordagem orientada a objetos de seu projeto (eventuais extensões não afetam a interface com o usuário); d) não é afetada pela incorporação de modeladores interativos ou não, bastando incluí-los na base de ferramentas.

6.2 Considerações sobre implementação

Tomando a concepção básica de *VISTA* como uma coleção de ferramentas de mapeamento, exploração, processamento e registro de dados científicos e abstraindo-se uma implementação completa numa linguagem de programação orientada a objetos, pode-se analisar as possibilidades de sua implementação tanto sobre um sistema construtor de aplicativos de visualização como *Iris Explorer* como sobre um ambiente de simulação com recursos gráficos como *Simsript II.5* com *Simgraphics*.

Já foi observado que o sistema *Iris Explorer* oferece a abordagem *dataflow* em sua interface com o usuário, a qual é diferente daquela utilizada como linguagem visual de exploração em *VISTA*: o usuário-final precisaria interagir com um mapa, cujo fluxo de execução é fixo, contrapondo-se à flexibilidade apresentada pela abordagem orientada a

ferramentas. Uma alternativa seria implementar *VISTA* como um aplicativo *Iris Explorer*, o que é discutido abaixo.

Cada ferramenta pode ser implementada como um módulo, com um painel de controle individual, para a entrada de seus próprios parâmetros. A interface orientada a ferramentas, com possibilidade de inspeção visual das mesmas e das entidades, entretanto, pode não ser implementada com facilidade, pois depende dos recursos disponíveis no ambiente e, eventualmente, de recursos disponíveis em outros níveis (por exemplo, bibliotecas utilizadas pelo próprio sistema). O conceito de "aplicação" de uma ferramenta a uma entidade pode não ter mapeamento direto para a interface já que isto pode envolver, por exemplo, a especificação do nome do arquivo que contém os dados relativos à entidade em questão. Igualmente, a aplicação de ferramentas diversas, sucessivamente, deveria estar prevista no diagrama que descreve *VISTA*, pois o fluxo de dados entre os diferentes módulos deve ser indicado explicitamente. Considerando a modelagem dos dados, *Iris Explorer* tem formatos fixos definidos para os arquivos de dados. Os modelos das entidades previstos por *VISTA* deveriam ser representados nos formatos padrões de *Iris Explorer*.

O fato de *Iris Explorer* ser destinado mais à visualização do que à análise exploratória torna menos imediata a implementação das facilidades necessárias para o desempenho de tais tarefas. Por exemplo, cada possibilidade de consulta, combinada com uma forma de visualização, deveria ser implementada como um módulo, que recebe entidades de outros módulos os quais realizam as formas de seleção definidas. A nível conceitual, portanto, perde-se a manipulação da ferramenta de consulta como um objeto, passando-se à abordagem de processo. Por outro lado, uma variada gama de técnicas de visualização está disponível, bastando apenas a conversão dos dados para os formatos aceitos pelo *Iris Explorer*.

Considerando a implementação do conjunto de ferramentas sobre um ambiente de simulação como *Simsript II.5* com *Simgraphics*, verifica-se que suas facilidades só podem ser utilizadas no que diz respeito à representação do modelo a ser simulado. Todas as ferramentas devem ser implementadas numa linguagem de programação convencional. LINDSTAEDT [em and.] implementa as ferramentas de controle e monitoração de experimentos de simulação utilizando uma interface *OpenLook* gerada com *Guide*. Esse trabalho não segue a abordagem orientada a ferramentas, na interface com o usuário; é uma aplicação específica destinada a exercitar facilidades da simulação interativa visual.

através de um pré-processamento de modelos descritos em *Simsript*. Entretanto, as ferramentas de mapeamento de variáveis e de controle de experimentos foram implementadas e provam ser eficientes. Por conseguinte, a implementação de *VISTA* sobre um ambiente de simulação como este constitui, na realidade, uma implementação sobre uma linguagem de programação, utilizando as facilidades para simulação deste ambiente. Pode-se, ainda, para melhor garantir as características de extensibilidade, flexibilidade e robustez, optar por uma implementação numa linguagem orientada a objetos como *MODSIM II*, que agrega tanto facilidades de simulação próprias de *Simsript* como os recursos gráficos de *Simgraphics*.

Considerando todas as ferramentas definidas, suas respectivas implementações tornam necessária uma gama diversificada de software. Para o caso de simulação de sistemas em geral (discretos ou contínuos) é necessária uma linguagem de simulação que permita a descrição das entidades a serem simuladas. Para o estudo de campos escalares ou vetoriais, é necessário um conjunto de rotinas de análise numérica que podem ser utilizadas para a implementação das ferramentas de consulta, avaliação e mapeamentos para representações visuais. Para o tratamento de entidades descritas por dados multivariados do tipo característica, são necessárias ferramentas de análise multivariada como análise de discriminante, componentes principais, etc, além das ferramentas estatísticas convencionais. Assim, a garantia de uniformidade na implementação de todas estas configurações de *VISTA* só pode ser obtida através de uma implementação sobre uma linguagem orientada a objetos, utilizando uma biblioteca gráfica e recursos de interface com o usuário. Desta forma, todas as ferramentas de análise e de simulação próprias de cada classe de aplicações, ou seja, de cada classe de entidades, podem ser incorporadas ao ambiente com mínimo esforço. Por sua abordagem, *VISTA* permite esta incorporação sem afetar quaisquer outras classes de ferramentas já presentes numa dada configuração.

7 Conclusões e perspectivas

A contribuição do trabalho é aqui resumida e são avaliados os resultados obtidos. Na seqüência, as possibilidades de continuidade são comentadas.

7.1 Resumo

A principal contribuição deste trabalho é a especificação e sua análise de uma abordagem unificada para análise exploratória e simulação interativa visual. Esta abordagem traduz-se para uma metodologia de seleção de representações visuais, um conjunto de ferramentas de suporte às tarefas executadas nestas duas classes de estudos e um paradigma de linguagem visual de análise exploratória e controle de simulação. Foram realizados estudos de casos considerados representativos destas duas classes: análise de dados multivariados populacionais, análise de dados multivariados espaciais, análise de dados univariados, estudo de campos eletromagnéticos através de resultados de análise de elementos finitos, simulação interativa visual de objetos animados dinamicamente e simulação interativa de sistemas discretos.

A partir destes estudos foi definida uma classificação de entidades e atributos. Entidades podem pertencer a três grandes classes, a saber, sistemas compostos por múltiplas entidades, objetos manufaturáveis e entidades ou fenômenos naturais, e são caracterizadas por atributos que podem ser analisados de acordo com sua natureza, o tipo dos valores que podem assumir, a natureza e dimensão do domínio no qual estão definidos.

Ainda, a partir dos casos estudados, foram distingüidos objetivos e tarefas rotineiras de análise exploratória e simulação interativa. Como objetivos considera-se observação de objetos num contexto, análise da estrutura dos objetos, análise das suas propriedades estáticas e dinâmicas, análise comparativa de vários objetos e comunicação de informações. Tarefas básicas que servem à realização dos diversos objetivos são navegação, seleção, consulta, preparação, monitoramento e controle de processamento, anotação de textos e armazenamento de imagens. Esta classificação de tarefas permitiu a

definição de um conjunto básico de ferramentas necessário à realização tanto de análise exploratória visual como de simulação interativa visual. Estas ferramentas permitem que as tarefas identificadas sejam realizadas sobre as diferentes classes de entidades.

Ferramentas são procedimentos interativos sobre representações visuais de entidades e atributos. Toda representação visual é adotada como forma de transmissão de uma informação a respeito de uma entidade. Este aspecto recebeu especial atenção nesta tese; as representações visuais foram classificadas em ícones, gráficos e tabelas, diagramas e redes, modelos geométricos, mapas e seqüências, de acordo com a informação fornecida e com o atendimento dos objetivos identificados. Foi proposta uma metodologia para a escolha de representações visuais tendo como base, numa primeira etapa, a classe das entidades sob estudo e os objetivos do estudo (ou de uma fase dele) e, numa segunda etapa, a natureza dos atributos e a dimensão de seu domínio.

Como a metodologia parte dos objetivos do estudo e levanta as questões a respeito das entidades, de modo a sugerir representações visuais adequadas a cada caso, pode-se afirmar que ela serve tanto ao usuário final de um ambiente de análise exploratória ou simulação interativa visual como a um projetista de ferramentas novas destinadas a estas duas classes de aplicações. A par disto, a metodologia serviu tanto à conceituação da visualização de dados como mapeamento do estado interno das entidades para atributos de uma representação visual como à definição de ferramentas que auxiliassem a realização das tarefas de análise e simulação.

VISTA, a plataforma proposta nesta tese, reúne todos os elementos de suporte à metodologia e, por conseguinte, às tarefas de análise exploratória e simulação interativa visual. Traz a abordagem orientada a ferramentas como paradigma de linguagem visual, transpondo para o ambiente computacional o método de trabalho de um usuário científico no processo de análise de dados, tal como destacado por SPRINGMEYER et al. [1992]. A linguagem visual aqui proposta é conceitualmente diferente daquela empregada nos sistemas de visualização conhecidos: é baseada na "aplicação" de ferramentas às entidades em estudo, suportando a realização das tarefas necessárias à obtenção de respostas às questões levantadas pelo usuário. Entidades e ferramentas são modeladas segundo a abordagem orientada a objetos, de modo que há uma estruturação consistente em *VISTA*.

Vale ressaltar que a metodologia proposta sugere várias representações visuais, dependendo dos objetivos do estudo. *VISTA* suporta a escolha de representações alternativas, através da declaração de um mapeamento entre entidades-atributos e representações visuais. As diferentes ferramentas podem ser aplicadas indistintamente sobre as várias representações visuais. Neste sentido, *VISTA* difere dos conhecidos sistemas de visualização centrados no paradigma *dataflow* pois nestes as possibilidades de representação visual e exploração estão programadas, mesmo que numa linguagem de alto nível de conexão de módulos.

7.2 Avaliação e perspectivas

Considerando os objetivos e tarefas da análise exploratória visual e da simulação interativa visual, tal como foram identificados e expostos no capítulo 3, pode-se afirmar que o conjunto de ferramentas definido atende as necessidades destas classes de aplicações. A metodologia proposta é uma solução inicial para a questão da escolha de representações visuais perceptualmente efetivas. Neste aspecto reside a primeira extensão possível deste trabalho qual seja, uma análise da classificação de representações visuais apresentada e da própria metodologia, à luz de estudos sobre a percepção humana.

Cabe observar que *VISTA* é definida como a reunião destas ferramentas, mas não apresenta necessariamente todas numa mesma configuração. Este é um aspecto ainda não mencionado neste trabalho: apesar de terem sido definidos recursos para o atendimento de aplicações variadas, dificilmente todos vão estar presentes ao mesmo tempo. Por exemplo, um usuário cujos problemas sejam tratados com análise de dados multivariados populacionais muito provavelmente jamais trabalhará com problemas cuja solução envolva análise de elementos finitos. Há, entretanto, um conjunto significativamente grande de recursos que são comuns a todas as aplicações de análise exploratória e simulação interativa visual. A diferença de uma aplicação para outra é, em geral, o subconjunto das ferramentas da classe de avaliação de atributos e de processamento, que implementam métodos dependentes da aplicação.

Avaliando o presente trabalho em termos de funcionalidades observa-se que não é contemplada a fase de construção dos modelos, tanto considerando entidades a serem analisadas a partir de dados coletados, como entidades a serem simuladas. Os sistemas de

visualização e análise existentes contam com mecanismos de conversão de formatos de arquivos e isto é encarado como parte do processo de modelagem. Já os ambientes de simulação em geral são construídos contemplando uma linguagem específica. As exceções são *OpenSim* [MUTKA e MCKINLEY 1993] e o sistema proposto por ROOKS [1993]. O primeiro, entretanto, não traz todos os mecanismos de controle aqui propostos; o segundo necessita ser configurado para uma linguagem de simulação específica para ser utilizado. A extensibilidade inerente à abordagem adotada tanto na linguagem visual como na estruturação de *VISTA* garante, entretanto, que os aspectos de modelagem possam ser tratados de forma consistente no futuro.

Considerando o ponto de vista de "ambientes de visualização", pode-se avaliar o presente trabalho à luz da discussão apresentada por BUTLER e HANSEN [1992], a qual se resume a seguir. O citado trabalho relata uma discussão sobre ambientes de visualização partindo da noção de uma máquina abstrata. Um ambiente de visualização é uma máquina de alto nível, de propósito especial, adequada à visualização científica. Em se tratando de uma máquina, são considerados aspectos como dados (tipos de dados), conjunto de instruções (operações) e estruturas de controle (interfaces de programação).

Quanto a tipos de dados, BUTLER e HANSEN [1992] indicam a importância de discutí-los conceitualmente e não como formatos de dados. Há uma considerável confusão entre tipos de dados e formatos de dados; freqüentemente os sistemas apresentam formatos de malhas (topologia) como tipos de dados diferentes. A noção de **função de múltiplas variáveis** é discutida como tipo de dado abstrato científico principal. Esta noção é a de um **mapeamento n-m**, com **n** variáveis independentes e **m** variáveis dependentes. Os formatos existentes nos sistemas conhecidos correspondem a escolhas específicas de domínio e imagem ou diferentes representações computacionais deste esquema. É apresentada uma taxonomia baseada em relacionamentos de especialização e agregação, levando à noção mais genérica de funções de múltiplas variáveis; o trabalho reconhece, entretanto, a necessidade de aprofundar o estudo antes de assumir esta noção realmente como tipo de dado abstrato principal. Sob este aspecto *VISTA* apresenta tipos de dados bem determinados e genéricos o suficiente para abranger a variada gama de aplicações de análise exploratória e simulação.

Voltando à noção de máquina abstrata, cabe ressaltar que BUTLER e HANSEN [1992] não discutem os aspectos relativos às operações. Aparentemente, como a discussão diz respeito a ambientes de visualização, as operações neste caso são gerações

de representações visuais e são, portanto, assumidas as técnicas convencionais. Já nesta tese, a preocupação com ferramentas de exploração e processamento reflete-se na concepção da plataforma. As operações da máquina abstrata (ferramentas no nosso caso) estão previstas dentro de um contexto e um *modus operandi* coerente.

Na questão estruturas de controle em ambientes de visualização, a discussão identificou a existência de três níveis diferenciados de controle nos sistemas atuais, a saber, controle interativo, tipicamente dirigido por cardápios, programação em alto nível, representada pelo paradigma (visual) *dataflow* de construção de aplicativos, e programação de propósito geral, provida por bibliotecas de funções usualmente em C. Todos os três níveis são considerados importantes mas a discussão se concentra no quão completa deve ser a interface de programação de alto nível, pois a colocação de estruturas de controle complexas numa linguagem visual é possível mas pode tornar a interface complexa demais tornando a tarefa de construir aplicativos de visualização não trivial, mesmo para um usuário projetista. A arquitetura dos sistemas é igualmente discutida e o trabalho aponta duas classes: os aplicativos organizados como um único processo, com ênfase no controle interativo e interface tipo manipulação direta, e aqueles organizados como vários processos, enfatizando interfaces baseadas em *dataflow*. Aqui o trabalho se concentra na discussão dos aspectos de monitoramento e controle do processamento: em interfaces baseadas em *dataflow* a granularidade do monitoramento e do controle está associada à granularidade do aplicativo. O acompanhamento pode requerer "entrar" num determinado módulo, configurando-se como um problema de visualização de programas e de dados. A distinção entre visualização de programa, visualização de dados e linguagem visual de programação é enfatizada. *VISTA* considera apenas o controle interativo através da abordagem orientada a ferramentas. Quando o usuário requer um resultado que só pode ser obtido com a aplicação em seqüência de um conjunto grande de ferramentas, o controle interativo pode ser desvantajoso sobre um ambiente orientado a *dataflow*, onde o mesmo resultado seria obtido com a construção de um diagrama onde os módulos realizam cada processo primitivo. Isto pode ser contornado em *VISTA*, pela sua característica de extensibilidade: não há restrições quanto à composição de ferramentas numa de mais alto nível. Entretanto, no presente estágio, esta facilidade de combinação não está presente na linguagem visual definida.

BUTLER e HANSEN [1992] introduzem, ainda, a discussão sobre um modelo de referência para visualização científica, ou seja, um modelo para análise e avaliação de ambientes de visualização. Retomando a noção de função de múltiplas variáveis, o

embrião do que seria um modelo de referência constitui-se da taxonomia de dados científicos mencionada anteriormente e de um diagrama representando o fluxo de objetos entre operadores. Operadores foram definidos como pertencendo a duas categorias: mapeamentos e exibidores. Mapeamentos tomam funções de múltiplas variáveis e geram outras funções; exibidores produzem representações visuais. Exemplos de mapeamentos são extração de fatias e cálculo de gradiente.

Em relação ao exposto acima, é interessante observar que o trabalho aqui desenvolvido abordou estes aspectos de forma independente, ou seja, sem o conhecimento prévio desses resultados. A necessidade de sistematizar os conceitos de programação visual, visualização de programas e visualização de dados já havia sido identificada [FREITAS 1992b]; da mesma forma, já havia sido reconhecida a importância de uma taxonomia para entidades e atributos [FREITAS 1992a]. A utilização de uma metáfora alternativa à abordagem *dataflow* (a abordagem baseada em ferramentas) veio como decorrência da ênfase tanto em análise exploratória visual como em simulação interativa visual, visto que traduz mais fielmente o processo de análise de dados científicos. *VISTA*, a plataforma aqui especificada, contempla vários tipos de dados e oferece dois níveis de controle, o controle interativo por manipulação direta de ícones e a programação de propósito geral. Aqui cabe observar que a abordagem adotada assim como a modelagem orientada a objetos permitiu unificar o tratamento dado a entidades e ferramentas. Ambas são hierarquias de classes. Instâncias desses objetos são manipuladas diretamente pelo usuário. O controle interativo e a programação de propósito geral estão mais próximos em *VISTA*, já que ambos os níveis são definidos sobre classes de ferramentas. A questão da interface de programação em alto nível é parcialmente substituída pela configuração do ambiente, quando são inicializados apenas os objetos necessários a uma dada classe de aplicações. Resta, entretanto, definir um mecanismo de agrupamento de ferramentas em "macros" pelo usuário final, quando estas devem ser executadas em seqüência, com pouca ou nenhuma intervenção externa. A realização deste mecanismo como mais uma ferramenta de controle prova, por outro lado, a extensibilidade de *VISTA* obtida com a abordagem adotada.

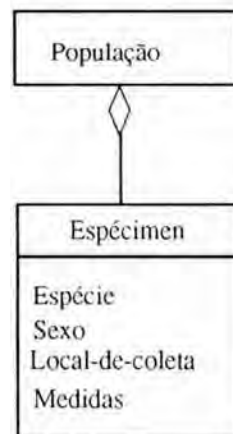
Embora a questão do modelo de referência não tenha sido explicitamente tratada, a taxonomia de entidades e atributos apresentada e a sistematização do conjunto de ferramentas para análise exploratória e simulação interativa visual, conjunto este baseado em procedimentos interativos sobre representações visuais, é, sem dúvida, uma alternativa ao modelo discutido por BUTLER e HANSEN [1992].

Uma importante contribuição à área de análise exploratória visual é, pois, a formalização de um modelo de referência baseado na abordagem orientada a ferramentas a partir dos resultados aqui obtidos. A abordagem orientada a ferramentas é, também, adequada para a distribuição do processamento numa rede de estações de trabalho ou entre estações e supercomputador. É sabido que um dos principais gargalos em aplicações científicas é o volume de dados. Neste caso, a execução de ferramentas de seleção ou análise em supercomputador e a interação a partir de estações de trabalho é plenamente viável dentro da abordagem conceitual proposta, não alterando a linguagem visual oferecida ao usuário de *VISTA*, revelando-se como outra contribuição possível originária deste trabalho.

ANEXO 1: Modelagem das entidades dos casos selecionados

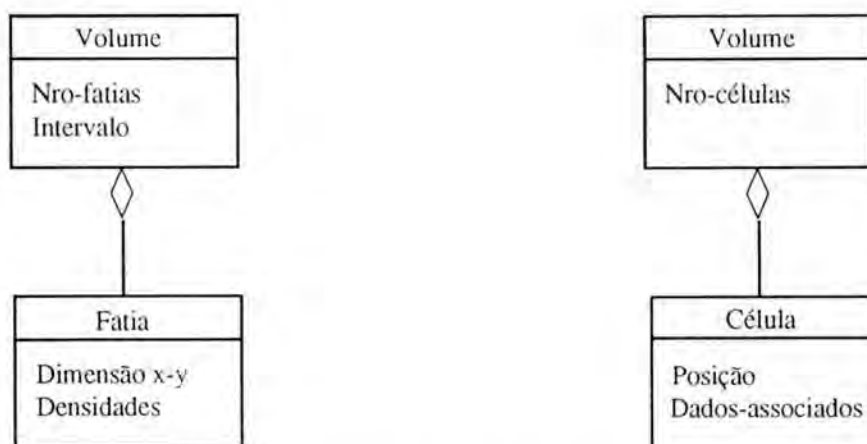
A1.1 Dados multivariados populacionais

O diagrama abaixo representa a fauna de uma determinada região, da qual foram tomadas medidas morfológicas. Espécie e sexo são atributos tipo característica, definidos dentre uma enumeração. Local-de-coleta é uma agregação {latitude, longitude}, um par de escalares, definida neste domínio contínuo, porém discretizado. As medidas são uma agregação de características, tomadas isoladamente, com valores numéricos definidos nos reais.



A1.2 Dados multivariados espaciais

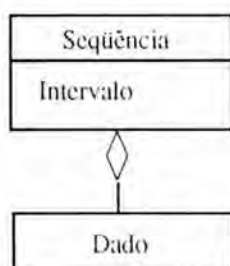
Dados tomográficos correspondem a estruturas regulares (fatias), tomadas espacialmente, com um intervalo fixo. Dados coletados espacialmente de forma esparsa são representados volumetricamente indicando-se explicitamente a posição de cada amostragem. Nos dois casos abaixo, os volumes são representados por agregações, de fatias ou de células. Cada fatia é tomada com determinada regularidade, possui uma dimensão e uma matriz de densidades de tecido (níveis de cinza). Por sua vez, cada célula tem implícita sua localização no espaço e o conjunto de dados associados.



Os atributos Nro-fatias, Intervalo, Nro-células e Dimensão x-y são do tipo característica, assumindo valores numéricos inteiros; o atributo Posição é um par $\{x,y\}$ que representa a função Escalar contínua de posição da amostragem no espaço tridimensional. O atributo Densidade é um Escalar, enquanto Dados-associados pode ser Escalar ou Vetorial.

A1.3 Dados univariados

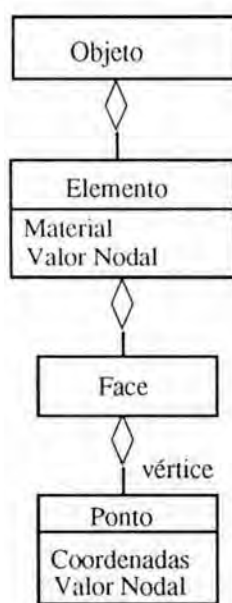
Esta classe de dados corresponde a uma seqüência de valores tomados ao longo de uma dimensão (tempo, por exemplo), em intervalos regulares, conforme o diagrama abaixo. O atributo Intervalo é do tipo Característica, inteiro ou real. Já o Dado pode ser Escalar, Vetorial ou Característica.



A1.4 Resultados de análise de elementos finitos

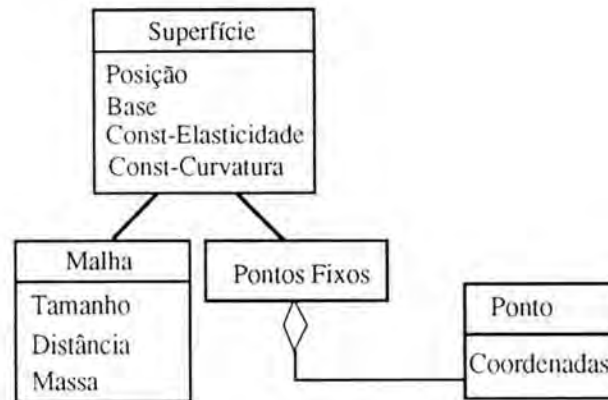
A análise de elementos finitos é aplicada sobre entidades cuja descrição espacial foi discretizada previamente. A modelagem abaixo considera que o processamento já foi efetuado, ou seja, que faz parte do conjunto de dados o valor nodal de cada elemento. Um objeto é discretizado em elementos compostos de um determinado Material, atributo

tipo Característica, definido numa enumeração. Cada elemento pode ter um atributo Valor Nodal calculado para o baricentro do elemento. Cada Elemento está definido geometricamente em termos de Faces e estas em termos de seus vértices, que são objetos Ponto, cujos atributos são Coordenadas e Valor Nodal.



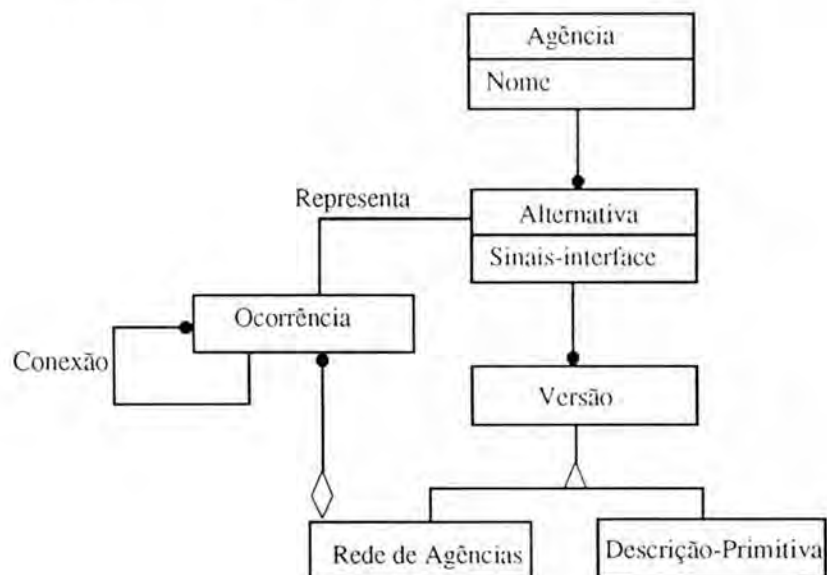
A1.5 Atores sintéticos deformáveis modelados geometricamente

O diagrama a seguir representa uma superfície deformável descrita a partir de uma malha de pontos regularmente espaçados. A superfície está posicionada num sistema de coordenadas, tem atributos de elasticidade e curvatura e são especificadas restrições sobre alguns pontos da malha, que são fixos. Atributos globais da superfície são Posição (função Escalar de posição), Base (vetores da base do sistema de coordenadas local) e as constantes que são tipo Característica. A superfície está descrita geometricamente por uma Malha, cujos atributos são do tipo Característica (Tamanho, Distância e Massa). Os pontos fixos são uma agregação de Pontos (definidos por suas Coordenadas).



A1.6 Modelo de redes de agências

O modelo de redes de agências é utilizado para representar sistemas digitais descritos modular e hierarquicamente. Cada sistema digital corresponde a uma Agência que pode ser descrita em termos de interface com o meio externo e em termos de sua estrutura interna. Diferentes interfaces com o meio externo caracterizam diferentes Alternativas de projeto deste sistema. Cada Alternativa, por sua vez, pode ter várias formas (Versões) de descrição interna, dependendo da linguagem de descrição de hardware escolhida. Algumas versões são primitivas, diretamente simuláveis; outras são hierárquicas, representando, na verdade, redes de sub-agências.



ANEXO 2: Exemplo de uso da metodologia

Serão utilizados como exemplos os casos de tratamento de dados multivariados populacionais (capítulo 2, seção 2.1) e de exploração de resultados de análise de elementos finitos (capítulo 2, seção 2.4).

A2.1 Dados multivariados populacionais

Fase 1: modelagem das entidades; vide anexo 1, seção 1.1.

Fase 2: determinação dos objetivos e tarefas do estudo.

Objetivos:

- determinar a distribuição de espécies geograficamente;
- verificar a distribuição de animais geograficamente, identificando medidas individualmente;
- determinar as diferenças morfológicas entre as espécies;
- determinar possíveis correlações entre medidas dentro da mesma espécie.

Tarefas e questões associadas:

- consulta a atributos de animais específicos
- comparação entre animais diferentes, considerando medidas e local de coleta;
- análise quantitativa das medidas pela realização de estatísticas como:
 - obtenção de média e desvio padrão de cada medida, por espécie;
 - classificação dos espécimens segundo as medidas morfológicas, utilizando análise de componentes principais;
 - quais medidas morfológicas separam as espécies e grupos dentro da mesma espécie.
- comunicação dos resultados.

Fase 3: refinamento das questões

- a distribuição geográfica dos exemplares requer mapeamento visual;
- a distribuição geográfica dos exemplares, segundo medidas, requer mapeamento visual e identificação textual;
- a consulta a atributos de exemplares exibidos geograficamente tem resposta numérica;
- a diferenciação morfológica entre os exemplares requer análise de componentes principais e apresentação por mapeamento visual.

- as correlações entre medidas requerem: a) representação visual por gráficos e b) resultado numérico.

Fase 4: escolha de representações visuais

- distribuição geográfica dos exemplares: mapa da região com ícones plotados;
- distribuição geográfica com medidas: mapa da região com ícones e medidas anotadas;
- consulta a atributos de exemplares: lista dos atributos e valores / tabela;
- diferenciação morfológica: gráfico com os eixos representando os autovetores da análise de componentes principais;
- correlações entre medidas: gráfico medida x medida | x medida | e tabela com matriz de correlação.

Fase 5: composição de representações

- distribuições geográficas;
- gráficos entre medidas.

A2.2 Análise de resultados de elementos finitos

Fase 1: modelagem das entidades; vide seção A1.4.

Fase 2: determinação dos objetivos e tarefas do estudo.

Objetivos:

- observação do modelo geométrico do objeto, incluindo material por elemento;
- observação de grandezas escalares associadas aos nós ou baricentros dos elementos;
- observação de grandezas vetoriais associadas aos nós dos elementos;
- observação de grandezas derivadas das fornecidas, sobre elementos geométricos do objeto.

Tarefas e questões associadas:

- observação do objeto a partir de diferentes pontos de vista;
- seleção e visualização de grandezas escalares e vetoriais;
- cálculo de grandezas derivadas sobre entes geométricos do objeto.

Fase 3: refinamento das questões:

- a representação geométrica dos objetos utiliza a descrição básica dos elementos como hexaedros;
- materiais e grandezas escalares e vetoriais são mapeados sobre a geometria;
- resultados numéricos são calculados sobre o contorno do objeto ou sobre um plano qualquer;
- resultados numéricos são avaliados com base em primitivas geométricas (ponto, linha ou arco).

Fase 4: escolha de representações visuais

- em sua representação mais simples, o modelo geométrico do objeto é apresentado na forma aramada;
- numa aproximação mais realista, os materiais dos elementos são representados com cores diferentes;
- a representação mais realista do objeto adota cores e graus de transparência diferentes por material;
- grandezas escalares podem ser representadas por isolinhas, no contorno do objeto, num plano de corte ou em todo o objeto, volumetricamente;
- grandezas escalares podem ser representadas como um mapa de cores no contorno do objeto;
- grandezas escalares podem ser representadas por isosuperfícies, volumetricamente, com transparência;
- resultados numéricos são exibidos textualmente ou em gráficos;

Fase 5: composição de representações

- diferentes grandezas escalares não podem ser exibidas sobre a geometria com o mesmo tipo de mapeamento;
- as demais representações podem ser combinadas, de acordo com especificação livre do usuário.

ANEXO 3: Classes de *VISTA*

A3.1 Classes relativas a entidades

Sistema de múltiplas entidades

Classe: Sistema

Atributos:

Tempo de simulação
 Modo de simulação
 Condições de Parada
 Dados Característicos
 Variáveis de Estado

Operações:

Simula
 Inclui (condição)
 Exclui (condição)
 Inializa (tempo)
 Inicializa (modo)
 Devolve (nomes-atributos)
 Devolve (nomes-variáveis)
 Devolve (nome-atributo, dado)
 Devolve(nome-variável, valor)
 Inicializa(nome-variável, valor)
 Inicializa (nome-atributo, dado)
 Fixa (nome-variável, valor)
 Fixa (nome-atributo, dado)

Relações:

Composta-por Entidade-Componente.

Classe: Entidade-componente

Atributos:

Dados Característicos
 Variáveis de Estado

Operações:

Altera-estado (evento)
 Devolve (nomes-atributos)
 Devolve (nomes-variáveis)
 Devolve (nome-atributo, dado)
 Devolve (nome-variável, valor)
 Inicializa(nome-variável, valor)
 Inicializa (nome-atributo, dado)
 Fixa (nome-variável, valor)
 Fixa (nome-atributo, dado)

Sub-classe: Sub-sistema

Relações:

Composta-por Entidade-Componente

Sub-classe: Ent-Primitiva

Atributos:

Procedimento

Objeto manufaturável

Classe: Objeto manufaturável

Atributos:

Dados Característicos
Dimensão do Domínio

Operações:

Devolve (dado)
Devolve (nomes-atributos)
Devolve (dimensão)
Devolve (descrição espacial)
Atribui (nome-atributo, valor)

Relações:

Composto por Objeto-Componente

Classe: Objeto-componente

Atributos:

Dados Característicos

Operações:

Devolve (nomes-atributos)
Devolve (nome-atributo, dado)
Devolve (descrição espacial)

Relações:

Composto-por Elemento-Geométrico

Classe: Elemento-Geométrico

Atributos:

Posição
Dados Analíticos

Operações:

Devolve (nomes-atributos)
Devolve (nome-atributo, dado)
Devolve (posição)
Devolve (descrição espacial)
Avalia (nome-variável, domínio, dado)

Relações:

Generalização-de { Ponto, Elemento-1D, Elemento-2D, Elemento-3D }

Entidades e fenômenos naturais

Classe: Entidade natural

Atributos:

Dados Característicos
Dimensão do Domínio
Posição

Operações:

Devolve (nomes-atributos)
Devolve (nome-atributo, dado)
Devolve (posição)
Devolve (descrição espacial)
Avalia (nome-variável, domínio, dado)

Relações:

Generalização-de { Amostra-Espacial, Conjunto }

Sub-classe: Amostra-Espacial

Atributos:

Dados Característicos

Operações:

Devolve (nomes-atributos)

Devolve (nome-atributo, dado)

Devolve (descrição espacial)

Avalia (nome-variável, domínio, dado)

Relações:

Composto-por Elemento-Espacial

Classe: Elemento-Espacial

Atributos:

Dados Analíticos

Operações:

Devolve (nomes-atributos)

Devolve (nome-atributo, dado)

Devolve (descrição espacial)

Avalia (nome-variável, domínio, dado)

Relações:

Generalização-de { Ponto, Célula-2D, Célula-3D }

Sub-classe: Conjunto

Atributos:

Dados Característicos

Operações:

Devolve (nomes-atributos)

Devolve (nome-atributo, dado)

Avalia (nome-variável, dado)

Relações:

Composta-por Elemento

Classe: Elemento

Atributos:

Dados Característicos

Operações:

Devolve (nomes-atributos)

Devolve (nome-atributo, dado)

Avalia(nome-variável, dado)

Classes referenciadas acima:

Classe: Elemento-2D

Atributos:

Base

Operações:

Devolve limites: lista de linhas

Devolve valor escalar (nome-atributo,dado)

Devolve valor vetorial (nome-atributo,dado)

Devolve valor característica (nome-atributo, dado)

Devolve valor no elemento (nome-atributo,dado)

Relações:

Generalização-de Triângulo, Polígono, Célula-2D

Classe: Elemento-3D

Atributos:

Base

Operações:

Devolve elementos-planares: lista de elementos-planares

Devolve valor escalar (nome-atributo, dado)

Devolve valor vetorial (nome-atributo, dado)

Devolve valor característica (nome-atributo, dado)

Devolve valor no elemento (nome-atributo, dado)

Relações:

Generalização-de Poliedro, Célula-3D

Classe: Elemento-1D

Atributos:

Operações:

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo, dado)

Relações:

Composto-por Pontos

Classe: Triângulo

Atributos:

Operações:

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo, dado)

Relações:

Composto-por Pontos

Classe: Polígono

Atributos:

Operações:

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo, dado)

Relações:

Composto-por Pontos

Classe: Célula-2D

Atributos:

Tamanho

Centro

Operações:

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo, dado)

Classe: Poliedro

Atributos:

Operações:

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo, dado)

Relações:

Composto-por Face

Classe: Face**Atributos:****Operações:**

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo,dado)

Relações:

Composto-por Pontos

Classe: Célula-3D**Atributos:**

Tamanho

Centro

Operações:

Devolve-coordenadas (pontos)

Devolve valor analítico (nome-atributo, dado)

Classe: Ponto**Atributos:**

Coordenadas

Operações:

Devolve-coordenadas (coordenadas)

A3.2 Classes relativas às representações visuais

Classe: Representação visual**Operações:**

Exibe

Relações:

Generalização-de Diagrama, Ícone, Objeto-Geométrico, Gráfico, Tabela, Mapa

Classe: Diagrama**Atributos:**

Lista-de-nodos

Lista-de-conexões

Lista-de-texto

Operações:

Inicializa-nodos (lista-de-nodos)

Inicializa-conexões (lista-de-conexões)

Inicializa-texto (lista-de-texto)

Exibe

Classe: Ícone**Atributos:**

Cor

Tamanho

Tipo

Descrição

Operações:

Inicializa-cor (cor)

Inicializa-tamanho (tamanho)
 Inicializa-tipo (tipo)
 Inicializa-descrição (descrição)
 Exibe

Relações:

Generalização de Bitmap, Objeto-2D, Objeto-3D.

Classe: Objeto-geométrico

Atributos:

Descrição-espacial

Operações:

Inicializa-descrição (descrição-espacial)
 Exibe

Classe: Gráfico

Relações:

Generalização de Gráfico-2D, Gráfico-3D, Gráfico-setores, Gráfico-barras,
 Histograma

Classe: Gráfico-2D

Atributos:

Legendas
 Tipos-ícones
 Lista-de-pontos

Operações:

Inicializa-legendas (legendas)
 Inicializa-ícones (lista-de-tipos)
 Inicializa-dados (lista-de-pontos)
 Exibe

Observação: as demais sub-classes de Gráfico são análogas.

Classe: Tabela

Atributos:

Descrição-colunas
 Lista-de-valores

Operações:

Inicializa-descrição (nro-colunas, nome-colunas)
 Inicializa-dados (lista-de-valores)

Classe: Mapa

Relações:

Generalização de Mapa-Cores, Mapa-Ícones, Isolinha, Isosuperfície, Linha-
 Trajetória, Ícone-Trajetória.

Classe: Mapa-Cores

Atributos:

Lista-de-{posição-valor}
 Lista-de-cores

Operações:

Inicializa-mapa (lista-de-cores)
 Inicializa-dados (lista-posição-valor)
 Exibe

Observação: as demais sub-classes de Mapas são análogas, no sentido de que são inicializadas estruturas internas e os métodos da classe geram a representação visual.

Classe: Mapeamento

Atributos:

Entidade
Representação visual
Janela

Operações:

Vincula (entidade, representação, janela)

Relações:

Generalização-de Map-Ícone, Map-Diagrama, Map-Gráfico, Map-Objeto-G, Map-Mapa, Map-Tabela

A3.3 Classes relativas às ferramentas

Classe: Navegador

Relações:

Generalização de {Cursor, Focalizador, Câmera}
Associado a Janela

Classe: Cursor

Relações:

Generalização de {Cursor espacial, Cursor individual}

Classe: Cursor Espacial

Atributos:

Entidade associada
Ícone
Posição

Operações:

Movimenta
Devolve (posição)

Classe: Cursor Individual

Atributos:

Entidade associada
Ícone

Operações:

Movimenta
Devolve (entidade)

Classe: Focalizador

Atributos:

Dimensões-janela
Posição-janela

Operações:

Movimenta
Redimensiona
Devolve (posição)
Devolve (dimensões)

Classe: Câmera

Atributos:

Posição

Alvo

Luzes

Operações:

Movimenta (deslocamento)

Orienta

Ativa (luz, parâmetros)

Desliga (luz)

Classe: Seletor

Relações:

Generalização de { Seletor individual, Seletor espacial, Seletor lógico }

Associado a Janela

Classe: Seletor Individual

Operações:

Devolve (objeto)

Relações:

Vinculado-a Cursor Individual

Classe: Seletor Espacial

Operações:

Devolve (entidade)

Relações:

Generalização-de { Seletor-Ponto, Seletor-Linha, Seletor-Janela, Seletor-Volume, Seletor-Corte }

Vinculado-a Cursor Espacial

Classe: Seletor Lógico

Atributos:

Expressão

Operações:

Constrói (expressão)

Devolve (entidade)

Classe: Consulta

Operações:

Devolve (valor)

Relações:

Generalização-de { Leitor, Medidor-escalar, Medidor-vetorial }

Classe: Avaliador

Atributo:

Função

Operações:

Devolve (valor)

Relações:

Baseado-em Seletor

Classe: Atribuição

Operações:

Atribui (valor, atributo)
Atribui-constante (valor, atributo)

Relações:

Baseado-em Seletor

Classe: Experimento**Atributos:**

Lista de arquivos-log
Lista de variáveis monitoradas
Lista de expressões
Lista de estímulos

Operações:

Especifica (variável, arquivo-log)
Constrói (expressão)
Remove (expressão)
Remove (variável)
Cria (estímulo)
Remove (estímulo)
Ativa (entidade)
Especifica (opção, tempo de simulação)

Relações:

Baseado-em Seletor

Classe: Registro**Atributos:**

Janela associada
Arquivo de imagem

Operações:

Vincula (janela)
Legenda (janela)
Armazena (janela)

Relações:

Baseado-em Seletor

Referências bibliográficas

- BELANGER, R. MODSIM II - A modular, object-oriented language. In: WINTER SIMULATION CONFERENCE, 1990, New Orleans. **Proceedings...** New Orleans: IEE/ACM/SCS, 1990. p. 118-122.
- BELANGER, R. **MODSIM II - The high-level, object-oriented language**. La Jolla, CA: CACI Prod. Co., 1991. (20 p.)
- BELL, P.C.; O'KEEFE, R.M. Visual Interactive Simulation: History, Recent Developments, and Major Issues. **Simulation**, San Diego, v. 49, n.3, p.109-116. Sept. 1987.
- BESHERS, C.; FEINER, S. Autovisual: Rule-based design of interactive multivariate visualizations. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 13, n. 4, p. 41-49. July 1993.
- BORGES, J.A.; JOHNSON, R.E. Multiparadigm visual programming language. In: WORKSHOP ON VISUAL LANGUAGES, 1990, Skokie, Illinois. **Proceedings...** Los Alamitos: IEEE, 1990. p.233-240.
- BRITAIN, D.L.; ALLER, J.; WILSON, M.; WANG, S.C. Design of an end-user data visualization system. In: IEEE CONFERENCE ON VISUALIZATION, 1990, San Francisco. **Proceedings...** Los Alamitos: IEEE, 1990. p. 323-328.
- BRODLIE, K.W. ET AL. (Eds.) **Scientific Visualization - Techniques and Applications**. Berlin: Springer-Verlag, 1992.
- BROWN, M.H. Exploring algorithms using Balsa-II. **Computer**, Los Alamitos, v. 21, n. 5, p. 14-33, May 1988.

- BUTLER, D.M.; HANSEN, C. Visualization'91 Workshop Report: Scientific Visualization Environments. **Computer Graphics**, New York, vol 26, n.3 , p. 213-216, Aug. 1992.
- CACI PRODUCTS. **SIMSCRIPT II.5 Programming language manual**. La Jolla: CACI, 1983.
- CACI PRODUCTS. **SIMGRAPHICS User's guide and casebook**. La Jolla: CACI, 1991.
- CRAY RESEARCH. **Multipurpose Graphic System User Manual**. Eagan, 1991.
- DYER, D.S. A dataflow toolkit for visualization. **IEEE Computer Graphics and Applications**, v. 10, n. 4, p. 60-69, July 1990.
- EARLE, N.J.; BRUNNER, D.T.; HENRIKSEN, J.O. Proof: The general purpose animator. In: WINTER SIMULATION CONFERENCE, 1990, New Orleans. **Proceedings...** New Orleans: IEEE/ACM/SCS, 1990. p. 106-108.
- EARNSHAW, R.A.; WISEMAN, N. **An Introductory Guide to Scientific Visualization**. Berlin: Springer-Verlag, 1992.
- EDEL, M. The Tinkertoy graphical programming environment. **IEEE Trans. on Software Engineering**, Los Alamitos, v. 14, n. 8, p. 1110-1115, Aug. 1988.
- ELLSON, I.S.; COX, D. Visualization of Injection Molding. **Simulation**, San Diego, v. 51, n. 5, p. 184-188, Nov. 1988.
- ELVINS, T.T. A visualization computing environment for a widely dispersed scientific community. San Diego: SDSC, Aug. 1990. (GA-A20109).
- ENCARNAÇÃO, J.L.; TORRES, O.F.F.; WARMAN, E.A. (Eds) **CAD/CAM as a basis for the development of technology in developing nations**. In: IFIP WG 5.2 WORKING CONFERENCE, 1981, São Paulo. **Proceedings...** São Paulo: IFIP, 1981.

- FREITAS, C.M.D.S. **Técnicas de Visualização em Simulação**. Porto Alegre: PGCC/UFRGS, 1990. (TI 187).
- FREITAS, C.M.D.S. (a) **Um estudo sistemático sobre linguagens visuais**. Porto Alegre: PGCC/UFRGS, 1992. (RP 197).
- FREITAS, C.M.D.S. (b) **Um estudo sobre a aplicação dos conceitos de orientação a objetos e de recursos interativos visuais a linguagens e sistemas de simulação**. Porto Alegre: PGCC/UFRGS, 1992. (RP não publicado).
- FREITAS, C.M.D.S.; WAGNER, F.R. A methodology for selecting visual representations in scientific and simulation applications. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 1993, Recife. **Anais...** Rio de Janeiro: SBC, 1993. p. 89-97.
- GERSHON, N.D. Visualization of fuzzy data using generalized animation. In: CONFERENCE ON VISUALIZATION, 1992, Boston. **Proceedings...** Los Alamitos: IEEE, 1992. p.268-273.
- GLINERT, E.P.; TANIMOTO, S.L. Pict: an interactive graphical programming environment. **Computer**, Los Alamitos, v. 17, n. 11, p. 7-25, Nov. 1984.
- GOLENDZINER, L.G.; WAGNER, F.R.; FREITAS, C.M.D.S. Modeling digital systems in an integrated design environment. In: Darringer, J.A. and Rammig, F.J. (Eds.) **Computer Hardware Description Languages and Their Applications**. Amsterdam: North-Holland, 1990. p. 147-156.
- JOHNSON, M.E.; POORTE, J.P. A hierarchical approach to computer animation in simulation modeling. **Simulation**, San Diego, v. 50, n. 1, p. 30-36, Jan. 1988.
- HULTQUIST, J.P.M.; RAIBLE, E.L. SuperGlue: a programming environment for scientific visualization. In: IEEE CONFERENCE ON VISUALIZATION, 1992, Boston. **Proceedings...** Los Alamitos: IEEE, 1992. p. 243- 250.

- HURLEY, C.; BUJA, A. Analyzing high-dimensional data with motion graphics. **SIAM J. Sci.; Statist. Comput.**, New York, v. 11, p. 1193-1211, 1990.
- HURRION, R.D. **The Design, Use, and Required Facilities of an Interactive Visual Computer Simulation Language to Explore Production Planning Problem**. Londres: Univ. of London, 1976. (PhD Thesis).
- KAPLAN, M. The design of the Doré graphics system. In: Blake, E.H.; Wisskirchen, P. (Eds.). **Advances in object-oriented graphics**. Berlin: Springer-Verlag, 1991.
- LANG, U.; LANG, R.; RÜHLE, R. Integration of visualization and scientific calculation in a software system. In: IEEE CONFERENCE ON VISUALIZATION, 1991, San Diego. **Proceedings...** Los Alamitos: IEEE, 1991. p. 268-274.
- LEMOS, R.R. **Animação e tratamento de colisões de corpos rígidos utilizando análise dinâmica**. Porto Alegre: PGCC/UFRGS, Junho 1993. (Dissertação de Mestrado).
- LINDSTAEDT, E. **Implementação da abordagem hierárquica para a animação na modelagem de simulação usando SIMSCRIPT**. Porto Alegre: PGCC/UFRGS, 1993. (TI 339).
- LINDSTAEDT, E. **Ambiente de simulação interativa visual para SIMSCRIPT**. Porto Alegre: PGCC/UFRGS. (Dissertação de mestrado, em andamento).
- LOHSE, J. ET AL. Classifying visual knowledge representations.: a foundation for visualization reserach. In: IEEE CONFERENCE ON VISUALIZATION, 1990, San Francisco. **Proceedings...** Los Alamitos: IEEE, 1990. p. 131-138.
- LOMOW, G.; BAEZNER, D. A tutorial introduction to object-oriented simulation and Sim++. In: WINTER SIMULATION CONFERENCE, 1990, New Orleans. **Proceedings...** San Diego: IEEE/ACM/SCS, 1990. p. 149-153.
- LUCAS, B. ET AL. An architecture for a scientific visualization system. In: IEEE CONFERENCE ON VISUALIZATION, 1992, Boston. **Proceedings...** Los Alamitos: IEEE, 1992. p. 107-114.

- MARSHALL, R. ET AL. Visualization methods and simulation steering for a 3D turbulence model of Lake Erie. **Computer Graphics**, New York, v. 24, n. 2, p. 89-97, Mar. 1990.
- MANSSOUR, I; JUNGBLUT, V. **SIMTOMA - Um sistema para visualização volumétrica de imagens de tomografia computadorizada e ressonância magnética**. Porto Alegre: Bacharelado em Informática da PUC/RS, 1992. (Trabalho de conclusão).
- MANSSOUR, I.H. **Algoritmo de Ray-Casting para visualização volumétrica de dados obtidos por tomografia computadorizada**. Porto Alegre: PGCC/UFRGS, Fev. 1994. (TI 372).
- MCCORMICK, B.H.; DEFANTI, T.A.; BROWN, M. (Eds.). Visualization in Scientific Computing. **Computer Graphics**, New York, v. 21, n. 6, 1987.
- MORICONI, M.; HARE, D.F. Visualizing program designs through PegaSys. **Computer**, Los Alamitos, v. 18, n. 8, p. 72-85, Aug. 1985.
- MUTKA, M.W.; MCKINLEY, P.K. Supporting a simulation environment with OpenSim. **Simulation**, San Diego, v. 61, n. 4, p. 223-235, Oct. 1993.
- MUSSE, S. **Modelagem e animação de sistemas articulados baseadas na análise dinâmica**. Porto Alegre: PGCC/UFRGS, 1994. (Dissertação de mestrado).
- MYERS, B. Taxonomies of visual programming and program visualization. **Journal of Visual Languages and Computing**, New York, v. 1, n. 1, p. 97-123, 1990.
- MYERS, B.; BUXTON, W. Creating interaction techniques by demonstration. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 7, n. 9, p. 51-60, Sept. 1987.
- NEDEL, L.P. **Simulação de objetos deformáveis baseada na análise dinâmica**. Porto Alegre: PGCC/UFRGS, junho 1993. (Dissertação de Mestrado).

- NIELSON, G ET AL. Visualizing and modeling scattered multivariate data. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 11, n. 3, p. 47-55, May 1991.
- OLABARRIAGA, S.D. Visualization of oscillations data via spirals. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 1993, Recife. **Anais...** Rio de Janeiro: SBC, 1993. p. 99-105.
- OLIVEIRA, A.C. **Visualização de dados multidimensionais**. Porto Alegre: Depto. de Informática Aplicada/UFRGS, 1993. (Projeto de diplomação).
- OLIVEIRA, K. S. ET AL. A detection of the evolutionary time scale of the DA white dwarf G117-B15A with the whole earth telescope. **The Astrophysical Journal**, 1991, n. 9, p. L45-48.
- O'REILLY, J.J. SLAM II Tutorial. In: WINTER SIMULATION CONFERENCE. 1991, Phoenix. **Proceedings...** Los Alamitos: IEEE/SCS/ACM, 1991. p. 112-117.
- POORTE, J.P.; DAVIS, D.A. Computer animation with CINEMA. In: WINTER SIMULATION CONFERENCE, 1990, New Orleans. **Proceedings...** New Orleans: IEEE/ACM/SCS, 1990. p. 123-128.
- PRESS, W.H. ET AL. **Numerical recipes in C - The art of scientific computing**. New York: Cambridge University Press, 1988.
- REISS, S.P. Graphical program development with Pecan program development system. **Sigplan Notices**, New York, v. 19, n. 5, p. 30-41, May 1984.
- REISS, S.P. Working in the Garden environment for conceptual programming. **IEEE Software**, Los Alamitos, v. 4, n. 6, p. 16-27, Nov. 1987.
- RESEARCH SYSTEMS. **IDL Basics**. Boulder: Research Systems, 1992. p. 250.
- ROBERTSON, P.K. A methodology for choosing data representations. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 11, n. 3, p. 56-67, May 1991.

- ROMAN, G.C.; COX, K.C. A declarative approach to visualizing concurrent computations. **Computer**, Los Alamitos , v. 22, n. 10, p. 25-36, Oct. 1989.
- ROOKS, M. A unified framework for visual interactive simulation. In: WINTER SIMULATION CONFERENCE, 1991, Phoenix. **Proceedings...** Los Alamitos, IEEE/SCS/ACM, 1991. p. 1146-1155.
- ROOKS, M. A user-centered paradigm for interactive simulation. **Simulation**, San Diego, v. 60n. 3, p. 168-177, March 1993.
- RUMBAUGH, J. ET AL. **Object-oriented modeling and design**. Englewood Cliffs: Prentice-Hall, 1991.
- SACHET, D.F. **Um estudo de técnicas para obtenção de modelos geométricos a partir de dados médicos 3D**. Porto Alegre: PGCC/UFRGS, 1993. (TI 305).
- SACHET, D.F. **Extração de modelos geométricos a partir de dados volumétricos**. Porto Alegre: PGCC/UFRGS. (Dissertação de mestrado, em andamento).
- SCHMIDT, A.E.F. **Visualização e exploração de dados científicos associados a objetos tridimensionais**. Porto Alegre: PGCC/UFRGS, 1994. (Dissertação de mestrado).
- SCHROEDER, W.J. ET AL. VISAGE: an object-oriented scientific visualization system. In: VISUALIZATION'92. Boston, 19-23 Outubro, 1992. **Proceedings...** Los Alamitos: IEEE, 1992. p. 219-226.
- SHU, N.C. FORMAL: A forms-oriented visual directed application development system. **Computer**, Los Alamitos, v. 18, n. 8, p. 38-50, Aug. 1985.
- SMITH, D.C. **Pygmalion: a computer program to model and stimulate creative thought**. Stanford: Stanford University, 1975. (PhD thesis).
- SILICON GRAPHICS. **Iris Explorer User's Guide**. Mountain View: Silicon Graphics, 1991.

- SPRINGMEYER, R. **Designing for scientific data analysis: from practice to prototype**. Livermore: Lawrence Livermore National Laboratory, 1992. (PhD Thesis).
- SPRINGMEYER, R.R.; BLATTNER, M.M.; MAX, N.L. A characterization of the scientific data analysis process. In: IEEE CONFERENCE ON VISUALIZATION, 1992, Boston. **Proceedings...** LOS ALAMITOS: IEEE, 1992. p. 235-242.
- SUN MICROSYSTEMS. **GUIDE - Graphical User Interface Development Guide**. Mountain View: Sun Microsystems, 1989.
- TREINISH, L.A. **Introduction to data management methods for scientific visualization**. New York, ACM, 1992. p. 6.1 - 6.26 (SIGGRAPH'92 Course 1 Notes).
- TUKEY, J.W. **Exploratory Data Analysis**. Reading: Addison-Wesley, 1977.
- TUFTE, E. R. **Envisioning Information**. Cheshire: Graphics Press, 1990.
- TURNER, R. ET AL. An object-oriented methodology using dynamic variables for animation and scientific visualization. In: Chua, T.S.; Kunii, T.L. (Eds.) **CG INTERNATIONAL'90**. Tokyo: Springer-Verlag, 1990. p. 317-326.
- UPSON, C. ET AL. The Application Visualization System: a computational environment for scientific visualization. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 9, n. 4, p. 30-42, July 1989.
- WAGNER, F.R.; FREITAS, C.M.D.S.; GOLENDZINER, L.G. **O processo de projeto de sistemas digitais num ambiente integrado de CAD**. Porto Alegre: PGCC/UFRGS, 1985. (RT 024).
- WAGNER, P.R. **Um ambiente integrado de simulação de sistemas digitais**. Porto Alegre: PGCC da UFRGS, 1991. (Dissertação de mestrado).

- WEHREND, S.; LEWIS, C. A problem-oriented classification of visualization techniques.
In: IEEE CONFERENCE ON VISUALIZATION. 1990, San Francisco.
Proceedings... Los Alamitos: IEEE, 1990. p. 139-143.
- YOUNG, F.W.; RHEINGANS, P. Visualizing structure in high-dimensional multivariate data. **IBM Journal of Research and Development**, New York, v. 35, n. 1/2, p. 97-107, Jan/Mar. 1991.



Uma Abordagem Unificada para Análise Exploratória e Simulação Interativa Visual

Defesa de Tese apresentada aos Senhores:

Prof. Dr. Dalcídio Moraes Claudio

Prof. Dr. Léo Pini Magalhães (DCA/UNICAMP)

Prof. Dr. Luiz Alberto Vieira Dias (INPE)

Vista e permitida a impressão.
Porto Alegre, 26/10/94.

Prof. Dr. Flávio Rech Wagner,
Orientador.

Prof. Dr. José Palazzo Moreira de Oliveira,
Coordenador do Curso de Pós-Graduação
em Ciência da Computação.