

144174-3

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

**Ambiente de Simulação Visual
Interativa**

por

Ernesto Lindstaedt

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação



Prof. Flávio Rech Wagner
Orientador

Porto Alegre, Agosto de 1995.

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Lindstaedt, Ernesto

Ambiente de Simulação Visual Interativa / Ernesto Lindstaedt.—Porto Alegre: CPGCC da UFRGS, 1995.

77 p.: il.

Dissertação (mestrado)—Universidade Federal do Rio Grande do Sul. Instituto de Informática. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1995. Wagner, Flávio Rech, orient.

1. Ambientes de Simulação. 2. Simulação. I. Wagner, Flávio Rech, orient. II. Título

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA			
N.º CHAMADA		N.º REG:	
681.32.001.57(043) L753A		32600	
ORIGEM: D		DATA: 21/08/96	
LATA: 02/08/96		PREÇO: R\$ 18,00	
FUNDO:	FORN.:		
II	II		

Simulação - SBU/II

Ambiente: Simulação

SIMSCRIPT

Interface gráfica

Visualização

ENP 1.03.04.00-2

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Hélgio Trindade

Pró-Reitor de Pesquisa e pós-Graduação: Prof. Cláudio Scherer

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. José Palazzo Moreira de Oliveira

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

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

32600

681.32.001.57(043)
L753A

INF
1996/144174-3
1996/08/21

MOD. 2.3.2

Aos meus pais, Heinz Peter e Iliana.
Aos meus irmãos, Roberto e Ricardo.
A minha esposa, Ângela.

AGRADECIMENTOS

Ao meu orientador, prof. Flávio Rech Wagner pelo contínuo incentivo na busca de um crescente aperfeiçoamento do trabalho realizado, assim como pelo apoio, valiosas sugestões e correções que tornaram esta dissertação não só possível como uma tarefa gratificante. Sua participação dedicada exerceu influência decisiva nos rumos tomados ao longa da elaboração do presente trabalho.

Ao meu amigo e colega Ricardo Corrêa Krug, pela contribuição prestada e pelo espírito sempre solidário e colaborador.

Aos meus pais, Peter e Iliana, por todo esforço, dedicação e orientação que tornaram possível que eu chegasse até aqui, por terem a dose necessária de paciência e por todo amor, carinho e compreensão em mim depositados.

Em especial aos queridos amigos Giancarlo e Denise, Alexandre Berg, Patrícia Noll, Denise Bandeira, Paulo Wagner, Cesar De Rose e Valéria, Ronaldo Mello, Rafael Bordini, Otília, Robson Lemos e Laurence Santi pelo convívio agradável e engrandecedor.

Aos demais amigos, colegas e funcionários do CPGCC.

Ao CNPQ pelo apoio financeiro.

SUMÁRIO

LISTA DE FIGURAS	8
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO E MOTIVAÇÃO	12
1.1 Objetivos da dissertação	12
1.2 O processo de simulação	13
1.3 Ambientes convencionais de simulação	14
1.4 Introdução à simulação visual interativa	15
1.5 A abordagem hierárquica	16
1.6 Abrangência do ambiente proposto	17
1.7 Estrutura da dissertação	18
2 O PACOTE DE SIMULAÇÃO SIMSCRIPT	19
2.1 A linguagem de simulação	19
2.2 O pacote SIMGRAPHICS	20
2.3 O ambiente de programação	22
3 ARQUITETURA DO AMBIENTE VISE	24
3.1 Recursos de interação, controle e visualização do ambiente	24
3.2 Utilização do ambiente	26
3.3 Implementação do ambiente	27
4 APRESENTAÇÃO DO AMBIENTE VISE	29
4.1 A interface principal do ambiente	29

4.2 Os recursos de visualização	30
4.2.1 Áreas de visualização do modelo	32
4.2.2 Área de visualização estatística	38
4.2.3 Controle das áreas de visualização ativas	43
4.3 Os recursos de interação	44
4.3.1 Manipulação de variáveis e atributos de entidades	44
4.3.2 Monitoração de variáveis e atributos de entidades	46
4.3.3 Manipulação das distribuições de probabilidade	49
4.3.4 Escolha da unidade de tempo da simulação	51
4.4 Os recursos de controle da execução	53
4.4.1 Escolha do modo de execução	53
4.4.2 Determinação dos pontos de parada	55
4.4.3 Controle da lista de eventos	57
4.4.4 Disparo e suspensão da execução	59
4.4.5 Status da simulação	60
4.4.6 Operações de carga e salvamento da simulação	61
4.4.7 Abandonando o ambiente	62
5 CONCLUSÕES E TRABALHOS FUTUROS	63
ANEXO A-1 PROGRAMA DE TESTE DO PROTÓTIPO	67
ANEXO A-2 LISTAGEM DO PROGRAMA DE TESTE DO PROTÓTIPO	72
ANEXO A-3 DESCRIÇÃO DOS FORMATOS DE ÍCONES EMPREGADOS PELO AMBIENTE	74
A-3.1 O formato OBT	74

A-3.2 O formato ico2D	74
BIBLIOGRAFIA	76

LISTA DE FIGURAS

Figura 4.1	Interface principal do ambiente VISE	29
Figura 4.2	Botão de visualização	30
Figura 4.3	Janela Visualization Scheme	31
Figura 4.4	Janela Model Visualization	33
Figura 4.5	Janela Associations	35
Figura 4.6	Janela Accessing Variables and Entity Attributes	36
Figura 4.7	Janela Statistical Visualization	38
Figura 4.8	Janelas XY, Histogram, Bar Chart e Clock	41
Figura 4.9	Janela Dial, Thermometer, XYZ e Pie Chart	42
Figura 4.10	Botão de controle das áreas	43
Figura 4.11	Janela Active Windows	43
Figura 4.12	Botão de acesso a variáveis e atributos de entidades	44
Figura 4.13	Janela Arrays	45
Figura 4.14	Janela Value	46
Figura 4.15	Janela Show Value	46
Figura 4.16	Botão de monitoração	47
Figura 4.17	Janela Monitoration on Screen	48
Figura 4.18	Janela Monitoration on Disk	48
Figura 4.19	Botão de distribuições de probabilidade	49
Figura 4.20	Janela Probability Distribution Calls	50
Figura 4.21	Janela Probability Distribution Classes	51
Figura 4.22	Botão de unidade de tempo	51
Figura 4.23	Janela Simulation Unit	52

Figura 4.24	Janela New Simulation Unit	52
Figura 4.25	Botão de modos de execução	53
Figura 4.26	Janela Event Classes	54
Figura 4.27	Janela Execution Step by Step	55
Figura 4.28	Botão de pontos de parada	55
Figura 4.29	Janela Conditional Breakpoints	56
Figura 4.30	Botão de controle da lista de eventos	57
Figura 4.31	Janela Event List	58
Figura 4.32	Janela Event Access	58
Figura 4.33	Botão de disparo da execução	59
Figura 4.34	Botão de pausa na execução	60
Figura 4.35	Janela Execution Status	60
Figura 4.36	Botão de Status	61
Figura 4.37	Botão de carga e salvamento	61
Figura 4.38	Botão de saída do ambiente	62
Figura 4.39	Janela Attention	62
Figura A-1.1	Execução do programa teste, nível 1	69
Figura A-1.2	Execução do programa teste, nível 2	70
Figura A-1.3	Execução do programa teste, nível 3	71

RESUMO

Ao longo dos anos houve um aumento considerável da complexidade das simulações o que tem motivado intensa pesquisa com o intuito de possibilitar uma análise mais adequada e efetiva dos dados gerados pelas mesmas.

A elaboração de modelos bem como sua verificação e validação também são alvos passíveis de aprimoramento, podendo levar ao desenvolvimento de uma série de recursos que podem revolucionar a atividade de simulação em pouco tempo.

Há também uma tendência de uso crescente de representações gráficas bem como de animações na simulação, respaldada pelo maior poder de processamento alcançado pelas estações de trabalho recentemente.

A presente dissertação consistiu no projeto e implementação de um ambiente de simulação visual interativa chamado VISE (de Visual Interactive Simulation Environment). Este ambiente utilizou a linguagem SIMSCRIPT como ferramenta de modelagem e oferece uma série de recursos de interação, controle e visualização com o intuito de facilitar e potencializar a atividade de simulação.

PALAVRAS-CHAVE: Simulação Visual Interativa, Ambientes de Visualização, Simulação, Animação, Interfaces Gráficas.

TITLE: "A VISUAL INTERACTIVE SIMULATION ENVIRONMENT"

ABSTRACT

Recent years have seen a considerable increase on the complexity of simulations. This growing complexity is the motivation for an intense research on more adequate and effective tools for the analysis of data generated by simulations.

The activities of building, verifying, and validating simulation models are also important targets of improvement, leading to the development of various facilities that may cause a revolution in the simulation field in very short time.

There is also a trend towards the increasing use of graphical representations and animations in simulation, motivated and supported by the higher processing capabilities of present workstations.

The present work consisted in the specification, design and implementation of a visual interactive simulation environment called VISE. This environment uses the SIMSCRIPT simulation language as modeling tool and offers various interaction, steering, and visualization facilities aiming at an easier and more powerful simulation activity.

KEYWORDS: Visual Interactive Simulation, Visualization Environments, Simulation, Animation, Graphical Interfaces.

1 INTRODUÇÃO E MOTIVAÇÃO

1.1 Objetivos da dissertação

O objetivo da presente dissertação é a construção de um ambiente de simulação visual interativa, isto é, um conjunto integrado de ferramentas que oferecem recursos voltados para a atividade de simulação de uma maneira visual e interativa. Para isto foi necessário:

- a escolha de um pacote de simulação adequado, em função da possibilidade de ligação deste a rotinas externas escritas na linguagem C;
- o estudo das características desejáveis em um ambiente de simulação visual interativa;
- o estudo dos recursos de interação, controle e visualização necessários e
- a implementação de um pacote integrado de recursos de interação, controle e visualização.

Existe atualmente uma tendência crescente no uso da animação no campo da simulação, objetivando uma melhor percepção por parte do simulacionista (aquele que realiza uma simulação) sobre a estrutura e o comportamento do modelo simulado, permitindo conseqüentemente uma maior compreensão em relação ao funcionamento do mesmo. Em resposta a esta tendência procurou-se observar no projeto do ambiente a inclusão de recursos que suportassem esse tipo de facilidade.

O termo animação é usado ao longo do texto com o significado de representação gráfica da dinâmica do comportamento das entidades e variáveis que compõem o modelo ao longo de uma sessão de simulação.

1.2 O processo de simulação

O processo de simulação pode ser dividido em diversas etapas e a divisão aqui apresentada baseia-se em [GOR 79] e em [WAG 92].

1. **Formulação do problema:** projeto de um novo sistema ou análise de deficiências de um já existente, estabelecendo-se neste caso as metas do estudo;
2. **Definição do Sistema:** identificação dos componentes e funções do sistema;
3. **Decisão sobre aplicabilidade do uso de simulação:** análise da relação custo-benefício das alternativas para o estudo em questão;
4. **Formulação do modelo:** especificação de componentes, variáveis, parâmetros, relações a serem incluídas no modelo;
5. **Preparação dos dados:** definição dos estímulos que serão fornecidos ao modelo durante a simulação;
6. **Translação do modelo:** codificação do modelo numa linguagem de simulação e verificação da correção da implementação do modelo em relação à sua especificação;
7. **Validação do modelo:** observação da correspondência do comportamento do modelo em relação ao sistema real;
8. **Planejamento estratégico:** planejamento do conjunto de experimentos;

9. **Planejamento tático:** planejamento de cada experimento;
10. **Experimentação:** realização do conjunto de experimentos;
11. **Interpretação:** análise dos resultados obtidos com a simulação;
12. **Documentação:** confecção dos manuais do sistema para sua manutenção e uso;
13. **Implementação dos resultados:** aplicação no sistema real das conclusões obtidas a partir da simulação.

1.3 Ambientes convencionais de simulação

Ambientes convencionais de simulação empregam normalmente um formato textual para a translação do modelo, isto é, a codificação do mesmo é feita na forma de uma linguagem. Os resultados da simulação são visualizados através de extensas tabelas ou listas numéricas, sem a utilização de gráficos ou animações. O processo convencional de condução da simulação implica em uma interatividade pequena entre o simulacionista e o modelo durante a simulação. Sucessivas modificações seguidas de recompilações do modelo são inerentes a este tipo de ambiente.

Outros ambientes de simulação possuem recursos diferentes: alguns oferecem a possibilidade de modelagem interativa visual, outros não permitem a construção de representações gráficas ou se permitem não dão a possibilidade do usuário interagir com o experimento durante uma simulação. Uma revisão bibliográfica bem detalhada sobre outros ambientes de simulação pode ser vista em [FRE 94] e [FRE 92].

1.4 Introdução à simulação visual interativa

O termo "Simulação visual interativa" (Visual Interactive Simulation ou VIS) foi introduzido por Bell e O'Keefe [BEL 87] e designa "a condução de uma simulação pelo usuário com a visualização de dados intermediários e a possibilidade de interação com o modelo, para modificação de parâmetros e variáveis, permitindo deste modo um controle completo do modelo durante este processo". Com o aumento do tamanho e da complexidade das simulações junto com o aumento no volume dos resultados gerados por elas, tornou-se interessante que os dados provenientes de simulações fossem exibidos de maneira gráfica para poderem ser convenientemente analisados, sendo que nos casos onde o volume de dados gerados é muito grande, esta é a única maneira de se proceder a uma análise sobre estes dados. Conforme Freitas [FRE 90] a visualização permite que se encontre estruturas (relações) existentes dentro dos dados e que não aparecem através de métodos convencionais de análise. Ainda segundo Freitas [FRE 94], é interessante que um ambiente de simulação visual interativa possua as seguintes características:

- Construção visual interativa do modelo de simulação, isto é, a especificação gráfica do modelo;
- Associação de representações gráficas a entidades e variáveis do modelo;
- Especificação da dinâmica das representações gráficas, o que corresponde à definição do comportamento das representações gráficas, ou seja, à especificação das alterações que as representações gráficas sofrem em função das modificações que ocorrem nas entidades ou variáveis com as quais estão associadas;
- Possibilidade de interação a nível de experimento (1º nível de interação), consistindo da implementação dos seguintes recursos: simulação passo a passo, simulação normal com interrupção, alteração de parâmetros entre duas execuções, ativação/desativação de vistas, ativação/desativação de monitoramento de variáveis, alteração das estatísticas exibidas;

- Possibilidade de interação a nível de modelo (2º nível de interação), onde seria possível interromper a execução da simulação e em seguida alterar o modelo para prosseguir a execução. Cabe salientar que isto causa um regime transitório logo após reiniciada a execução, onde as estatísticas não serão confiáveis;
- Possibilidade de coleta de resultados para pós-processamento;
- Possibilidade de interação a nível de pós-processamento.

1.5 A abordagem hierárquica

É desejável que os recursos de visualização sejam otimizados de forma diferenciada para o suporte às fases de verificação, validação, análise e apresentação dos resultados. Esta idéia foi apresentada por Johnson [JOH 88] onde ele descreve uma abordagem hierárquica para animação na modelagem de simulação, empregando diferentes classes de recursos de visualização para cada uma destas fases.

Nesta abordagem é proposta uma estrutura de 3 níveis crescentes de complexidade de animação e representação gráfica, sendo cada nível apropriado para determinada fase do processo de simulação.

O nível inicial (I) é usado na fase de depuração e verificação do modelo, sendo empregada uma representação gráfica com alto grau de abstração e baixo grau de detalhamento, utilizando-se na animação somente alteração de cores, visualização do conteúdo de variáveis, gráficos tipo termômetro, ícones abstratos (sem semelhança na aparência física com a entidade que é representada) e pouco ou nenhum movimento de ícones.

O segundo nível (II) é empregado nas fases de validação e análise do modelo, e sua representação gráfica possui um maior grau de detalhamento e um grau menor de abstração, caracterizando-se como uma representação 2D do sistema, com movimento, alteração de cores, display de variáveis e ícones com maior grau de correspondência em relação à aparência física da entidade que representam. Este nível também pode ser usado na fase de depuração e verificação.

O último nível (III) é destinado à fase de comunicação e apresentação do modelo e dos resultados, possuindo um baixo grau de abstração e um alto grau de detalhamento, tendo como característica uma representação 2D ou 3D do sistema incluindo todas as movimentações relevantes do sistema. Os ícones são detalhados e têm correspondência visual com a entidade que representam.

1.6 Abrangência do ambiente proposto

O ambiente proposto suporta amplamente as características de associação de representações gráficas, especificação da dinâmica destas representações e a possibilidade de interação a nível de experimento.

O conjunto de recursos oferecidos por VISE possibilita a adoção da abordagem hierárquica para animação na modelagem vista no capítulo 1.5. A construção visual interativa do modelo de simulação não foi implementada porque tinha-se em vista a modelagem em SIMSCRIPT criando-se um ambiente para dar suporte a partir da fase de depuração e verificação do modelo.

O ambiente não dá a possibilidade de interação a nível de modelo por razões técnicas que serão discutidas oportunamente. A coleta de resultados para pós-processamento e a interação neste nível também não são suportadas devido à natureza do próprio ambiente, que é de interatividade. Não se espera a princípio o uso do ambiente em simulações muito complexas e demoradas.

1.7 Estrutura da dissertação

Esta dissertação está dividida em cinco capítulos (incluindo o presente capítulo) e três apêndices.

O primeiro capítulo está voltado para uma introdução à simulação e ambientes de simulação, buscando explicar os objetivos da dissertação e os motivos que levaram a escolha do assunto bem como da maneira que o mesmo foi tratado.

O capítulo seguinte trata do pacote de simulação empregado sobre o qual se desenvolveu o presente trabalho.

A arquitetura do ambiente proposto é vista em seguida, onde são detalhadas as características do mesmo e apresentados seus recursos de interação, controle e visualização.

O funcionamento do ambiente e a forma de interação do usuário com o mesmo são discutidas no capítulo seguinte, onde são também descritas todas as janelas que compõem o sistema.

No último capítulo são apresentadas as conclusões sobre o trabalho realizado e é feita uma análise do sistema a partir de um exemplo-teste, sendo também discutidas as limitações do sistema bem como possíveis extensões futuras para o mesmo.

Os dois primeiros apêndices relacionam-se com o programa-exemplo usado no teste do protótipo e o último apêndice refere-se ao formato dos ícones usados pelo ambiente.

2 O PACOTE DE SIMULAÇÃO SIMSCRIPT

O pacote de simulação escolhido para a implementação do ambiente de simulação foi SIMSCRIPT II.5 . Os fatores que levaram a esta escolha são:

- SIMSCRIPT é uma linguagem de simulação bastante poderosa e completa;
- SIMSCRIPT possibilita a ligação a rotinas externas escritas nas linguagens C ou FORTRAN, o que pode ser visto em [CAC 89a];
- A versão UNIX da linguagem está disponível no Instituto de Informática, com boa documentação.

SIMSCRIPT é uma linguagem de simulação de propósito geral que suporta programação modular e estruturada.

2.1 A linguagem de simulação

SIMSCRIPT é uma linguagem de simulação orientada a eventos, embora suporte também a abordagem orientada a processos. A descrição completa da linguagem pode ser vista em [CAC 89].

A estrutura de um programa escrito em SIMSCRIPT pode ser dividida basicamente em 3 partes: preâmbulo, programa principal e rotinas de eventos (ou processos).

A primeira parte de um programa consiste no preâmbulo (*preamble*) onde é feita a declaração de entidades permanentes e temporárias bem como de seus atributos. Aqui também é feita a declaração de variáveis globais e de eventos e seus parâmetros.

No programa principal são inicializadas as variáveis globais, criadas as entidades permanentes, programados os primeiros eventos e por fim a simulação é disparada.

Cada evento está associado a uma rotina dentro da qual são descritas as ações que irão ocorrer quando este evento for disparado, isto é, quando o controle da execução passar para esta rotina.

Estas ações correspondem a alterações no estado do sistema tais como modificação nos valores de atributos das entidades, criação e destruição de entidades temporárias e programação de novos eventos.

A linguagem SIMSCRIPT não será detalhada no presente texto, sendo interessante que o leitor já possua um conhecimento razoável da linguagem. Maiores informações sobre SIMSCRIPT podem ser encontradas em [CAC 83].

2.2 O pacote SIMGRAPHICS

SIMGRAPHICS é uma biblioteca de rotinas e funções que possibilitam integrar gráficos às simulações feitas em SIMSCRIPT. Este pacote permite a elaboração de gráficos de apresentação, gráficos interativos e animação. Entre os gráficos de apresentação suportados estão histogramas, gráficos tipo torta, diagramas de barras e gráficos X-Y. Para criar gráficos de apresentação é necessário seguir os seguintes passos:

1. declarar a variável global que será observada como do tipo DISPLAY no preâmbulo do programa;
2. criar os ícones no *Graph Editor* e salvá-los como arquivos descritores de ícones;
3. adicionar ao código SIMSCRIPT comandos específicos para realizar a associação entre as variáveis tipo DISPLAY e os arquivos descritores de ícones;
4. inicializar a janela gráfica.

SIMGRAPHICS possui uma abordagem baseada em formulário que permite adicionar aos programas interfaces gráficas que permitem interatividade. Formulários são criados usando-se o *Graphics Editor* e podem ser acessados pelo programa através de um comando específico.

Este pacote ainda possibilita a criação de gráficos animados. Os ícones podem ser associados a entidades do modelo e mudam de posição ou de aparência em função da alteração que ocorrem nos atributos destas entidades ao longo da simulação. Para a elaboração de gráficos animados é preciso:

1. declarar as entidades que serão animadas como sendo do tipo GRAPHIC no preâmbulo do programa;
2. definir os ícones usando o *Icon Editor*;
3. definir um sistema de coordenadas;
4. adicionar o comando DISPLAY e comandos de controle de animação ao código SIMSCRIPT.

Os passos necessários para criação de gráficos de apresentação, formulários e gráficos animados estão descritos detalhadamente em [CAC 91].

O pacote SIMGRAPHICS não foi empregado na elaboração do ambiente VISE, uma vez que este tem como um dos seus objetivos tornar o controle dos gráficos associados ao modelo completamente independente deste, isto é, que não fosse necessário adicionar qualquer comando de controle de animação no código SIMSCRIPT, o que não é possível quando se emprega o pacote SIMGRAPHICS. Esta biblioteca restringe os gráficos a um contexto bidimensional, o que pode ser uma séria limitação caso se deseje empregar a simulação no 3º nível da abordagem hierárquica para animação.

Além disso, o uso de rotinas da biblioteca *XView* do sistema X Window na implementação do ambiente VISE possibilita uma uniformidade com as classes de objetos de interface empregadas no ambiente *OpenWindows* do sistema operacional *SunOs*, que segue a especificação *OPEN LOOK* de interfaces gráficas de usuário descrita em [SUN 90] e em [SUN 90a].

2.3 O ambiente de programação

A atividade de modelagem usando-se este pacote de simulação é completamente textual, implicando na codificação do modelo na linguagem SIMSCRIPT. Após feita a codificação, este é compilado e ligado gerando um código executável que quando disparado dá início então ao processo de simulação propriamente dito. A partir deste instante é que ocorrerão as animações definidas no momento da codificação do modelo.

O pacote SIMSCRIPT oferece algumas ferramentas para depuração do modelo, entre elas comandos da linguagem para monitoração de variáveis, o programa *prof*, útil quando se realiza a análise de performance de um programa, e opções de compilação para verificação de erros em tempo de execução.

É possível especificar no programa uma variável como sendo "monitorada". Cada vez que esta variável tiver seu conteúdo alterado, uma rotina especificada pelo usuário é chamada.

O utilitário *prof* gera um perfil de execução de um programa. Este perfil consiste numa relação das rotinas empregadas no programa, sendo que para cada rotina é exibido um conjunto de informações, a saber: porcentagem de tempo em relação ao tempo total de execução gasto pela rotina, uma soma do número de segundos gastos pela rotina e pelas listadas anteriormente, o número de vezes em que ela foi chamada, quantos milissegundos a rotina gastou cada vez que foi chamada e o nome da rotina.

Quando são usadas as opções de compilação com verificação de erros em tempo de execução é realizada uma validação a cada referência a elementos de um array ou referência a atributos. Em caso de erro, é gerado um relatório mais detalhado do estado da simulação no momento em que esta foi interrompida.

Como foi dito na seção anterior, todos aspectos de uma visualização devem estar explicitamente referenciados no código de um programa SIMSCRIPT. A biblioteca SIMGRAPHICS dá ao usuário um certo grau de interatividade caso se construa formulários, o que também exige a inclusão de comandos específicos no código do programa, além de obrigar o usuário a projetar a interface e construí-la através do *Graphics Editor*. Não é possível durante a execução alterar o aspecto das representações gráficas empregadas pelo modelo nem modificar a associação entre a representação e a variável. Por fim, o pacote SIMGRAPHICS está a princípio restrito a um contexto gráfico bidimensional.

O acesso e manipulação de variáveis é possível mas o usuário fica encarregado de escrever as rotinas de monitoração que tornam isto viável. É possível acessar a lista de eventos mas isto só pode ser feito com o uso de ponteiros específicos da linguagem.

3 ARQUITETURA DO AMBIENTE VISE

3.1 Recursos de interação, controle e visualização do ambiente

VISE provê 3 diferentes classes de recursos, a saber: recursos de interação, recursos de controle e recursos de visualização.

Com os recursos de interação oferecidos pelo ambiente é possível ao simulacionista interagir com as variáveis e atributos das entidades do modelo, podendo ler o valor atual de uma variável ou atribuir um valor a ela. O ambiente permite também que o usuário tenha acesso às distribuições de probabilidade usadas no modelo dando-lhe a oportunidade de direcionar a simulação ("steering").

Os recursos de controle permitem que a execução seja suspensa, reativada, que se tenha acesso à lista de eventos e se especifique pontos ou condições de parada na execução. Pode-se especificar também o modo de execução, que pode ser por um determinado número de unidades de tempo bem como até um certo tempo de simulação especificado pelo usuário. Além disso é possível disparar a execução até que haja a ocorrência de um determinado evento.

A classe de recursos de visualização permite que se especifique associações de variáveis e atributos de entidades a ícones a fim de que se possa visualizar de maneira gráfica o andamento da simulação. A visualização pode se dar de duas formas: a visualização estatística e a visualização do modelo.

No primeiro caso, trata-se de acompanhar através de gráficos estatísticos como histogramas, gráficos X-Y e X-Y-Z, diagramas de torta, diagramas de barras, relógios, diais e gráficos tipo termômetro a alteração do conteúdo de determinadas variáveis e atributos de entidades ao longo da simulação. Na visualização do modelo procurou-se implementar a associação de representações gráficas (ícones) a atributos de entidades e a variáveis e a especificação da dinâmica destas representações gráficas. Durante a simulação é possível ao usuário realizar dinamicamente a alteração dos parâmetros destas associações, conforme proposto por Freitas [FRE 92]

As cinco primeiras fases do processo de simulação (ver seção 1.2) não são alteradas com o uso do ambiente VISE. As principais diferenças no processo ocorrem nas fases de translação do modelo, validação, planejamento estratégico e tático, experimentação e interpretação dos resultados.

Na translação do modelo sem o uso do ambiente VISE é necessário incluir comandos específicos para monitoração de variáveis e para visualizações, o que não é necessário quando se realiza a simulação com o ambiente. O pacote SIMSCRIPT oferece poucas ferramentas para depuração úteis na verificação do modelo, ao contrário do ambiente que oferece funções como execução passo a passo, especificação de pontos de parada, acesso e monitoração de variáveis e atributos de entidades.

A validação do modelo é facilitada quando se usa o ambiente proposto, pois este oferece recursos que agilizam a especificação da visualização do comportamento do modelo. Além disso pode-se analisar a sensibilidade do modelo em relação à alteração de certos parâmetros, o que pode ser feito com os recursos de interação do ambiente.

As fases de planejamento estratégico, tático e a fase de experimentação também são beneficiadas em função do maior controle que o usuário ganha sobre a execução da simulação, podendo-se manipular o tempo de execução de cada experimento, inicializar o modelo através dos recursos de interação e alterar as distribuições de probabilidade do mesmo.

As facilidades de visualização do modelo possibilitam uma melhor interpretação dos resultados da simulação uma vez que tornam possível um acompanhamento mais efetivo de alterações nos estados de variáveis e atributos de entidades do modelo. Os recursos oferecidos pelo ambiente VISE se revelam também bastante úteis quando é adotada a abordagem hierárquica para animação na modelagem de simulação, introduzida na seção 1.5.

Nos níveis iniciais (I e II) da abordagem, pode-se especificar representações gráficas que empreguem ícones bidimensionais, usar os recursos de interação acessando-se o conteúdo de variáveis e atributos do sistema, modificar distribuições de probabilidade e utilizar gráficos estatísticos para realizar desde a depuração e verificação do modelo até a validação e análise do mesmo. No último nível (III) deve-se utilizar ícones com maior correspondência visual em relação à entidade que representam e para isto o ambiente oferece recursos de visualização com ícones tridimensionais e a possibilidade de associar os parâmetros destes a variáveis e atributos de entidades do modelo.

3.2 Utilização do ambiente

Os passos necessários para a utilização do ambiente VISE são os seguintes:

- Escrever o programa usando a linguagem SIMSCRIPT;
- Chamar o *script* de execução do ambiente;
- Interagir com o ambiente.

O *script* de execução do ambiente é responsável pela compilação do programa SIMSCRIPT, pela chamada de um pré-processador que vai analisar o código do usuário e gerar um segundo programa modificado com chamadas de rotinas do ambiente, pela ligação deste segundo programa com o código objeto do ambiente e pela chamada do programa executável resultante.

Assim que a interface principal do ambiente é exibida, o usuário pode começar a interagir com o ambiente, elaborando esquemas de visualização, inicializando variáveis e disparando a simulação, por exemplo.

3.3 Implementação do ambiente

O ambiente VISE foi implementado usando-se a linguagem C e rotinas da biblioteca XView. A interação com o código escrito em SIMSCRIPT só foi possível porque a linguagem SIMSCRIPT suporta chamadas a rotinas externas escritas em C ou FORTRAN. O código inicialmente escrito pelo usuário é modificado por um pré-processador que prepara o código para interagir com o ambiente. Este pré-processador gera um segundo código que possui as chamadas para as rotinas do ambiente, além de detectar as variáveis, atributos de entidades, eventos e rotinas definidas no programa e adicionar ao mesmo as rotinas de monitoração necessárias aos recursos de interação do ambiente. São também incluídos neste código eventos e rotinas de controle. Este segundo código resultante do processo de pré-processamento é ligado ao código objeto do ambiente gerando então um programa executável que é chamado pelo *script* de execução. Diversos mecanismos tiveram de ser projetados e implementados para:

- controle do tempo de execução;
- manipulação de variáveis e atributos de entidades;
- manipulação da lista de eventos;
- monitoração de variáveis e atributos de entidades;
- manipulação de distribuições de probabilidade;
- estabelecimento de pontos de parada;

- determinação das janelas de visualização ativas;
- disparo e suspensão da execução;
- criação e edição de esquemas de visualização;
- associação dos ícones com variáveis e atributos de entidades;
- visualização dos ícones bidimensionais e tridimensionais;
- carga e salvamento do esquema de visualização;
- carga e salvamento do estado atual da simulação;
- reconhecimento e alteração do código fonte inicial.

4 APRESENTAÇÃO DO AMBIENTE VISE

4.1 A interface principal do ambiente

A interface principal do ambiente VISE (figura 4.1) é composta de uma janela base retangular dividida em duas regiões, a saber: a região de controle e a região de status. A região de controle é composta por 14 botões dispostos na parte superior da interface e que são responsáveis pela ativação de determinadas tarefas relacionadas à interação, controle e visualização da simulação, sendo que cada tarefa é representada por um ícone no botão correspondente. Os botões estão agrupados nesta janela base de acordo com o tipo de tarefa que ativam. Na região de status, localizada na parte inferior da interface, são exibidas informações como o tempo atual de simulação, o modo de execução corrente, a unidade de simulação em uso e o status geral da simulação.

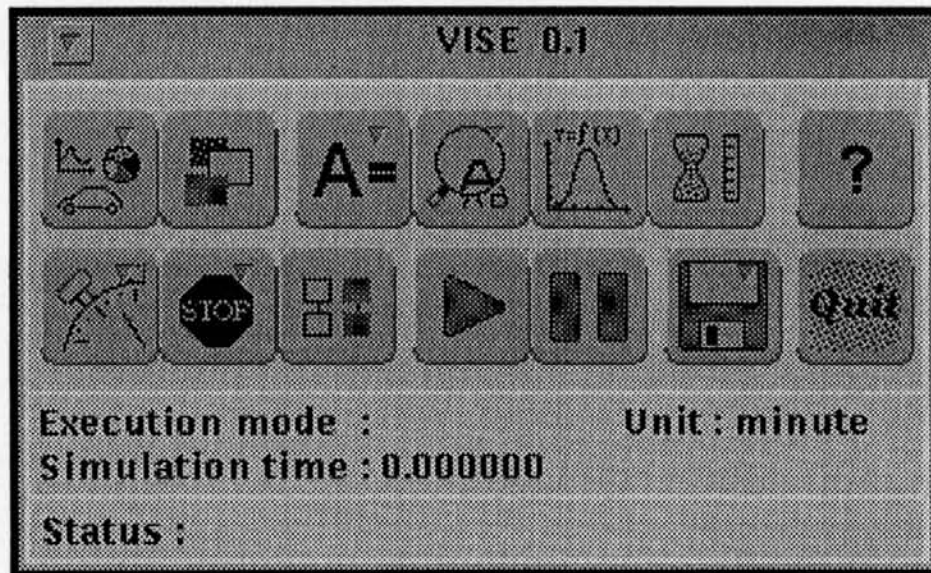


Figura 4.1- Interface principal do ambiente VISE

4.2 Os recursos de visualização

O ambiente VISE permite que se possa visualizar de maneira gráfica o comportamento de entidades e variáveis do modelo sendo simulado. O processo de visualização tem início na elaboração de um esquema de visualização que nada mais é do que a criação de uma série de áreas de visualização e a especificação das características de cada uma destas áreas definidas pelo simulacionista.

Uma área de visualização deve ser associada a uma das duas classes de visualização disponíveis. Uma vez definida a área é necessário definir os ítems que estarão associadas a ela, como gráficos estatísticos no caso de uma área de visualização estatística e de entidades visuais no caso de uma área de visualização do modelo. A definição de um esquema de visualização é feita ativando-se o botão de visualização (figura 4.2) na interface principal (figura 4.1).

A partir deste momento é possível criar um novo esquema, editar um esquema já existente, salvar o esquema em uso ou carregar para a memória um esquema de visualização previamente armazenado em disco.

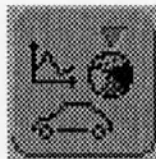


Figura 4.2- Botão de visualização

No momento de criar ou editar um esquema, é aberta uma janela responsável pela definição das características das áreas de visualização que compõem o esquema que está sendo criado ou editado. Esta janela é intitulada *Visualization Scheme* (figura 4.3) e possui na sua parte superior a lista das áreas já inseridas no esquema.

Cada área de visualização deve receber um nome que a identificará, além de ter definida sua altura, largura e cor de fundo. Depois disto a área deve ser inserida no esquema o que é feito através do botão *insert*.

Pode-se remover uma determinada área do esquema selecionando-a na lista e ativando o botão *delete*. É possível também editar uma área (botão *edit*) e duplicar o conteúdo de uma área, criando uma segunda área com as mesmas características da primeira, usando-se para isto o botão *copy*.

Assim que a área é inserida no esquema, pode-se proceder com a escolha da classe de visualização que vai ocorrer na área inserida. Esta escolha é realizada ativando-se uma das duas opções que aparecem na parte inferior da janela.

Pode-se optar pela classe de visualização estatística ou pela classe de visualização do modelo. Assim que uma das opções é ativada, uma janela correspondente é ativada, onde serão fornecidos parâmetros relativos à classe de visualização escolhida.

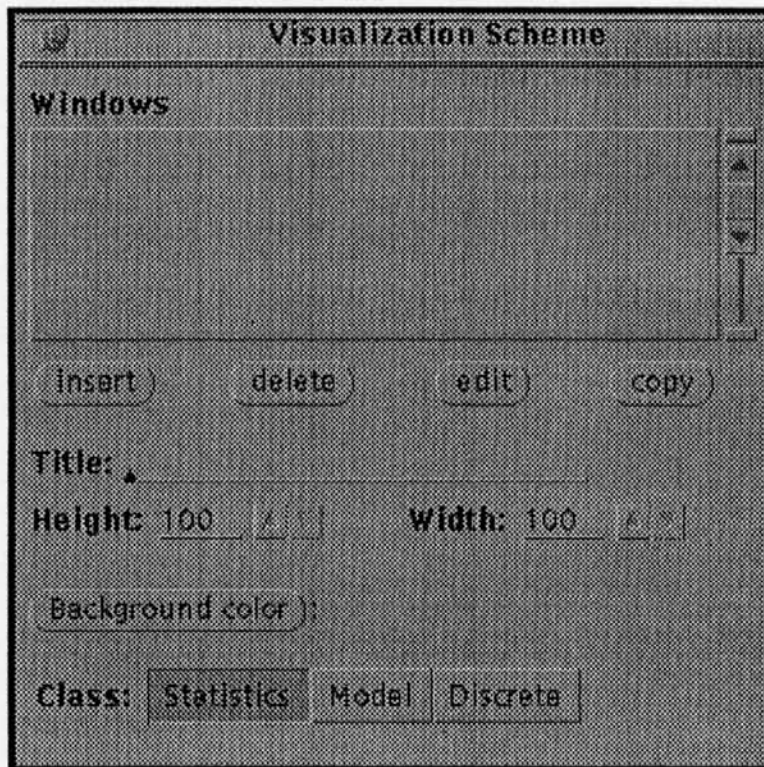


Figura 4.3- Janela Visualization Scheme

Na figura 4.3 relativa à janela *Visualization Scheme* aparece um terceiro botão chamado *Discrete* que corresponde a uma futura extensão do ambiente ainda não implementada e que é discutida no último capítulo da dissertação.

4.2.1 Áreas de visualização do modelo

Em uma área de visualização do modelo podem ser exibidas, durante a simulação, entidades visuais que são o produto da associação de uma representação gráfica (ícone) com atributos de entidades ou variáveis.

Dependendo da dimensão da área, podem ser exibidos ícones bidimensionais ou tridimensionais, além de cenários de fundo. Cada ícone pode ter seus parâmetros como localização ao longo dos eixos, tamanho, rotação em torno dos eixos e cor associados a variáveis e atributos de entidades. A visualização dos ícones que aparecem na área é realizada por uma câmera cujos parâmetros como posição do observador e posição do alvo também podem ser associados a variáveis e atributos de entidades.

A janela responsável pela definição de uma área de visualização do modelo é chamada *Model Visualization* (figura 4.4) e é composta por 3 listas: uma na parte superior, que contém as entidades visuais já criadas e inseridas na área, e duas listas na parte inferior da janela, sendo que uma contém os parâmetros das entidades visuais e a outra os parâmetros da câmera.

Quando se inicia a definição de uma área a primeira coisa a ser feita é a escolha da dimensão da visualização, que pode ser bi ou tridimensional. Isto é feito selecionando-se um dos botões colocados à direita do ítem *Dimension*.

Logo após pode-se proceder à escolha do cenário através do botão *Scenario*. Um cenário é a imagem de fundo que aparece na área de visualização do modelo durante a simulação. Será aberta em seguida uma janela na qual deve-se então fornecer o nome do arquivo que contém o cenário. Um cenário pode ser uma imagem no formato "rasterfile" da Sun, no caso de visualizações bidimensionais, ou um objeto composto no formato OBT no caso de visualizações tridimensionais. Caso não se especifique um cenário será usada como cor de fundo aquela escolhida no momento da criação da respectiva área no esquema de visualização.

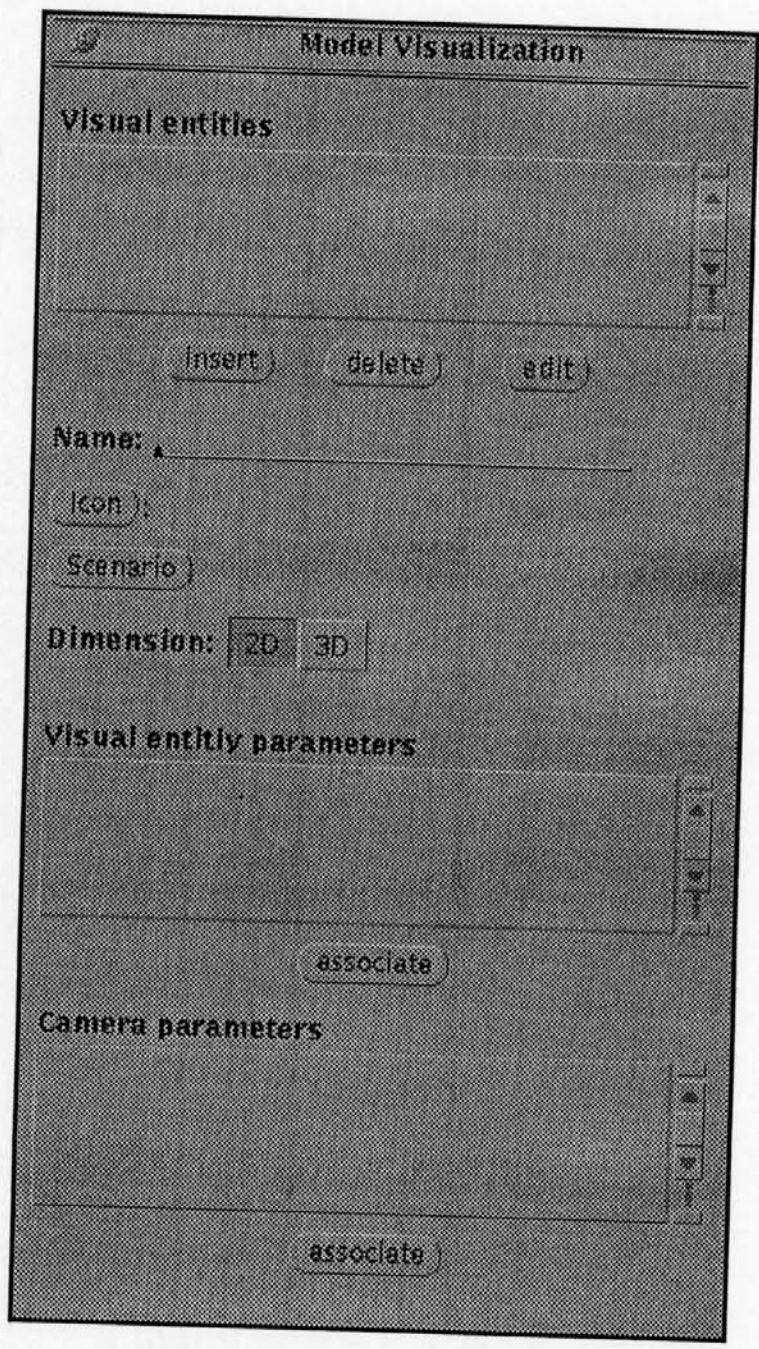


Figura 4.4- Janela Model Visualization

A seguir pode-se começar a criar as entidades visuais que irão compor a visualização nesta área. Para isto escolhe-se um ícone, o que é feito através do botão *Icon*. Novamente é aberta uma janela na qual deve-se introduzir o nome do arquivo que contém o ícone. O ícone pode estar no formato ico2d para representações gráficas bidimensionais ou no formato OBT para representações tridimensionais. Depois de carregado o ícone, procede-se à associação de parâmetros da entidade visual com atributos de entidades e variáveis do modelo.

Os parâmetros de uma entidade visual bidimensional são:

- Posição no eixo X;
- Posição no eixo Y;
- Rotação em torno do eixo X;
- Rotação em torno do eixo Y;
- Escala no eixo X;
- Escala no eixo Y;
- Cor do ícone.

Já os parâmetros de uma entidade visual tridimensional são:

- Posição no eixo X;
- Posição no eixo Y;
- Posição no eixo Z;
- Rotação em torno do eixo X;
- Rotação em torno do eixo Y;
- Rotação em torno do eixo Z;

- Escala no eixo X;
- Escala no eixo Y;
- Escala no eixo Z;
- Cor do ícone.

A associação é feita da seguinte forma: na lista de parâmetros da entidade visual aponta-se e seleciona-se com o mouse o parâmetro que irá ser associado. Em seguida seleciona-se o botão *associate* que fica imediatamente abaixo da lista de parâmetros da entidade visual. Com isto será ativada a janela *Associations* (figura 4.5) onde vai-se selecionar a classe de associação desejada. Existem 3 classes de associação, discutidas a seguir.

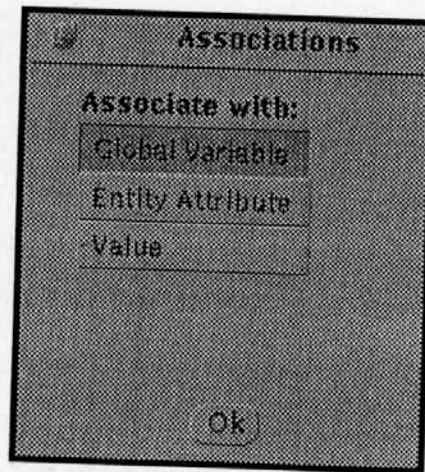


Figura 4.5- Janela Associations

Associando com variáveis globais Permite a associação com variáveis globais do modelo. Selecionando o botão *Global Variable* será ativada a janela responsável pelo acesso a variáveis do modelo. Esta janela chamada *Accessing Variables & Entity Attributes* (figura 4.6) contém uma lista na qual aparecem os nomes de todas variáveis globais do modelo. Para selecionar uma variável basta selecioná-la com o mouse na lista.

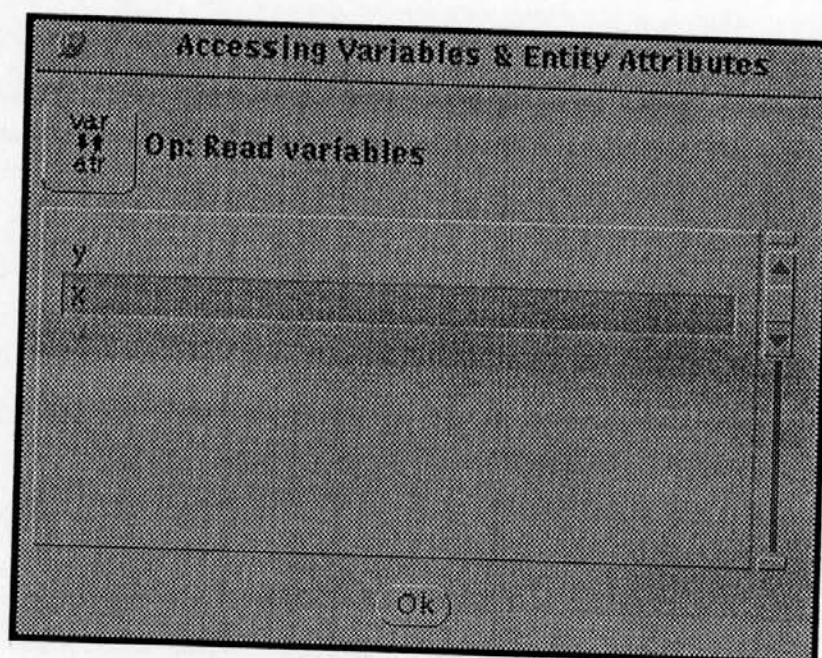


Figura 4.6- Janela Accessing Variables and Entity Attributes

Associando com atributos de entidades Possibilita a associação com o atributo de uma entidade do modelo. Selecionando o botão *Entity Attribute* será ativada a janela responsável pelo acesso a atributos de entidades do modelo. Esta janela chamada *Accessing Variables & Entity Attributes* contém uma lista na qual aparecem os nomes de todos atributos de todas entidades do modelo. Para selecionar um atributo basta selecioná-lo com o mouse na lista.

Associando com um valor Realiza a associação com um valor constante definido pelo usuário. Selecionando o botão *Value* será ativada uma janela intitulada *Value* na qual será fornecido pelo usuário o valor com o qual o parâmetro da entidade visual será associado. Uma extensão sugerida no último capítulo é a possibilidade de associação com expressões que incluem valores constantes e variáveis e atributos de entidades.

Depois do usuário ter estabelecido todos os vínculos que achar necessário entre os parâmetros da entidade visual que foi criada e determinadas variáveis ou atributos de entidades do modelo pode-se proceder à associação dos parâmetros da câmera com uma das 3 classes de associação existentes.

Os parâmetros de uma câmera num contexto tridimensional são:

- Posição do observador no eixo X;
- Posição do observador no eixo Y;
- Posição do observador no eixo Z;
- Posição do alvo no eixo X;
- Posição do alvo no eixo Y;
- Posição do alvo no eixo Z.

Já os parâmetros de uma câmera num contexto bidimensional são:

- Posição do observador no eixo X;
- Posição do observador no eixo Y;
- Altura do observador a partir do plano.

O processo de associação dos parâmetros da câmera é análogo ao processo de associação dos parâmetros das entidades visuais.

Finalmente, depois da entidade ter seus parâmetros definidos, deve ser identificada com um nome, o que é feito no espaço correspondente abaixo da lista de entidades visuais. A entidade visual pode então ser inserida na lista de entidades visuais da área de visualização corrente. Isto é feito através do botão *insert* que localiza-se abaixo desta lista (ver figura 4.3). Pode-se ainda remover uma entidade desta lista (botão *delete*) ou editar as especificações de uma entidade visual através do botão *edit*.

Cabe salientar que a primeira coisa a ser feita quando a janela *Model Visualization* é ativada é a escolha da dimensionalidade das representações gráficas da área em construção, do contrário um aviso de advertência será emitido pelo sistema.

A dimensão de uma área é única, isto é, não é possível, em uma mesma área de visualização, ter ícones bidimensionais e tridimensionais ao mesmo tempo. Para cada área só é possível especificar um cenário e uma câmera.

4.2.2 Área de visualização estatística

Nesta classe de área de visualização são oferecidos 8 diferentes tipos de gráficos a partir dos quais o usuário poderá ter uma visão do comportamento estatístico de variáveis e atributos de entidades do modelo. A janela responsável pela definição de uma área de visualização estatística é chamada *Statistical Visualization* (figura 4.7) e é composta por oito botões com ícones representativos dos tipos de gráficos, e de uma lista dos gráficos que pertencem à área de visualização estatística em construção.

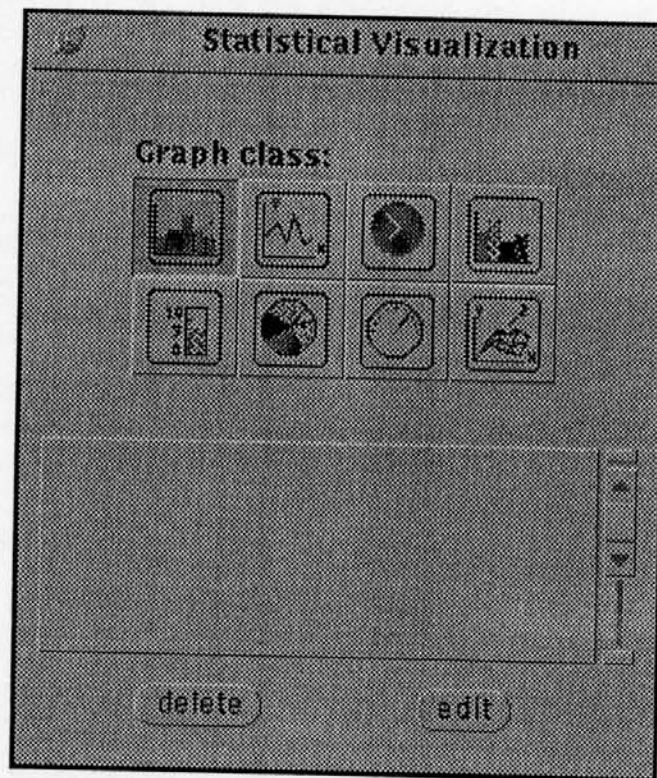


Figura 4.7- Janela Statistical Visualization

Para ativar o processo de especificação de um gráfico, basta selecionar com o mouse o botão correspondente ao gráfico que se deseja inserir na área. Quando é realizada a seleção, uma janela correspondente ao gráfico é ativada (figuras 4.8 a 4.9). Nesta nova janela, devem ser fornecidas as especificações relativas ao gráfico que se está elaborando, entre as quais as associações com variáveis e atributos de entidades. A seguir são descritas as principais características das diversas classes de gráficos suportados por VISE.

Histograma É usado para visualizar a distribuição de frequências em um determinado número de intervalos de ocorrência.

Gráfico X-Y Esta classe é útil para representar o comportamento de uma função no espaço 2D (uma variável independente e uma dependente).

Gráfico X-Y-Z Emprega-se este tipo de gráfico para representar o comportamento de uma função no espaço 3D (duas variáveis independentes e uma dependente).

Diagrama de Torta Quando for interessante saber o quanto uma determinada classe de dados representa em relação ao total da população, é indicado o uso de diagramas tipo torta.

Diagrama de Barras Este tipo é adequado para representar dados que podem ser agrupados em um certo número conhecido de classes, sendo que cada barra representa a frequência de ocorrência da classe correspondente.

Gráfico tipo Relógio O gráfico tipo relógio é a maneira gráfica para visualizar-se o tempo atual (corrente) de simulação.

Gráfico tipo Dial Um gráfico dial é útil para representar dados escalares que variam dentro de um intervalo fixo de valores.

Gráfico tipo Termômetro A classe de gráfico termômetro é usada para representar, assim como o gráfico dial, dados escalares que oscilem entre dois limites fixados.

Alguns parâmetros a serem fornecidos na especificação das características de um gráfico são comuns à maioria dos tipos, como por exemplo:

- **Título** do gráfico;
- **Nome dos eixos** (se houverem);
- **Cores** dos diversos itens que compõem um gráfico;
- **Posição** em X e Y na área de visualização;
- **Maior e menor** valor para cada eixo;
- **Escala** das marcações nos eixos;
- **Numeração** nas marcações (valor do intervalo);
- **Associação** com variável ou atributo de entidade.

Nos gráficos Diagrama de Torta, Dial e Relógio deve ser especificado o raio do gráfico a ser mostrado. Ainda no relógio pode-se optar pelo tipo de representação analógica ou digital e a exibição ou não do ano, mês e dia. É possível também determinar a exibição seletiva de horas, minutos e segundos. Em todos os tipos de gráficos pode-se determinar se a atualização dos mesmos se dará de forma dinâmica, isto é, ao longo da simulação, ou só no momento de pausa na execução. Isto pode ser feito através da opção *Dynamic*. No Histograma, Diagrama de Barras, gráficos X-Y e X-Y-Z pode-se também permitir a exibição do gráfico com grade, o que é especificado na opção *Grid*. Por fim, nos gráficos Diagrama de Barras e Diagrama de Torta é suportada a exibição de legendas referentes aos dados representados.

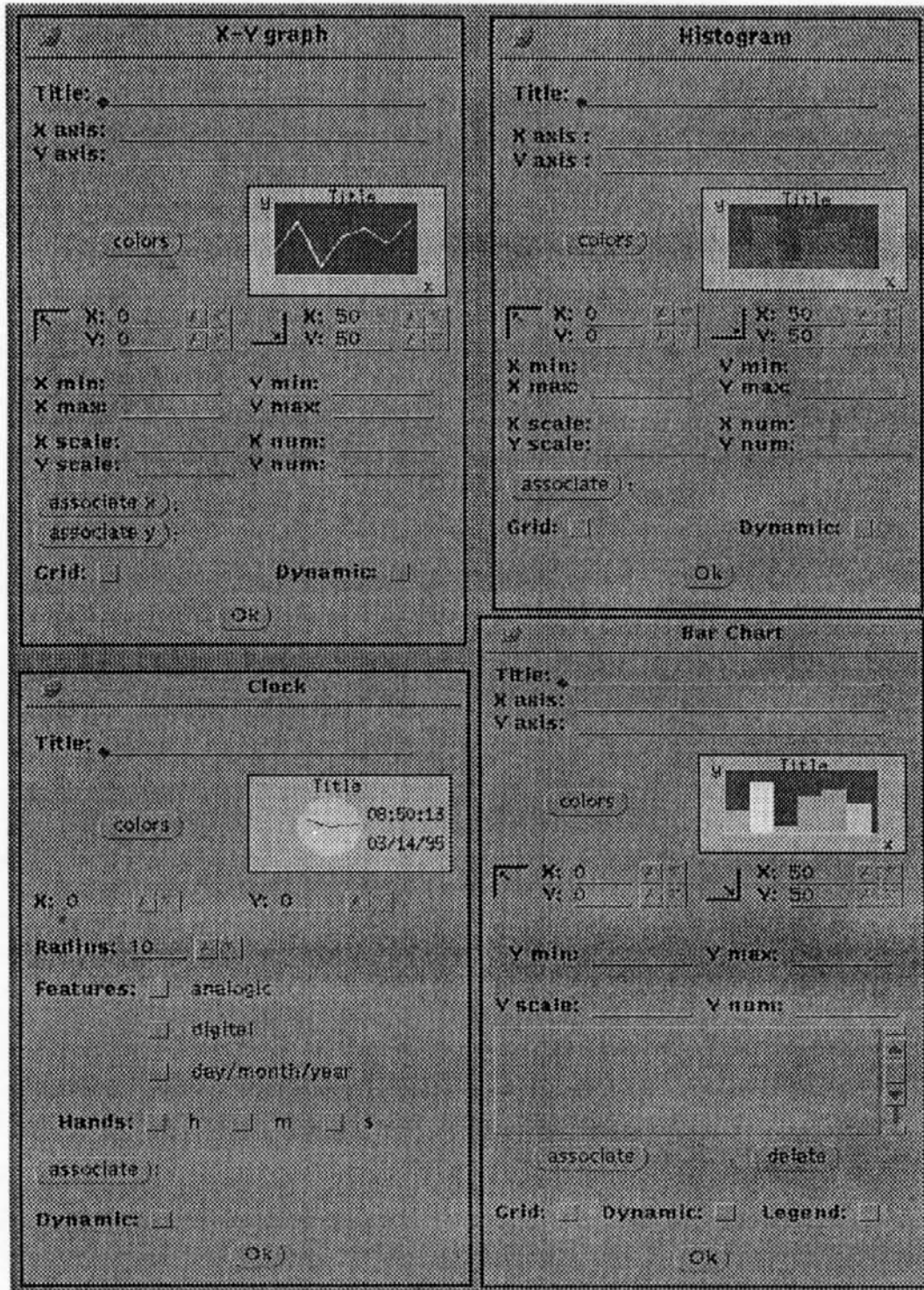


Figura 4.8- Janelas XY, Histogram, Bar Chart e Clock

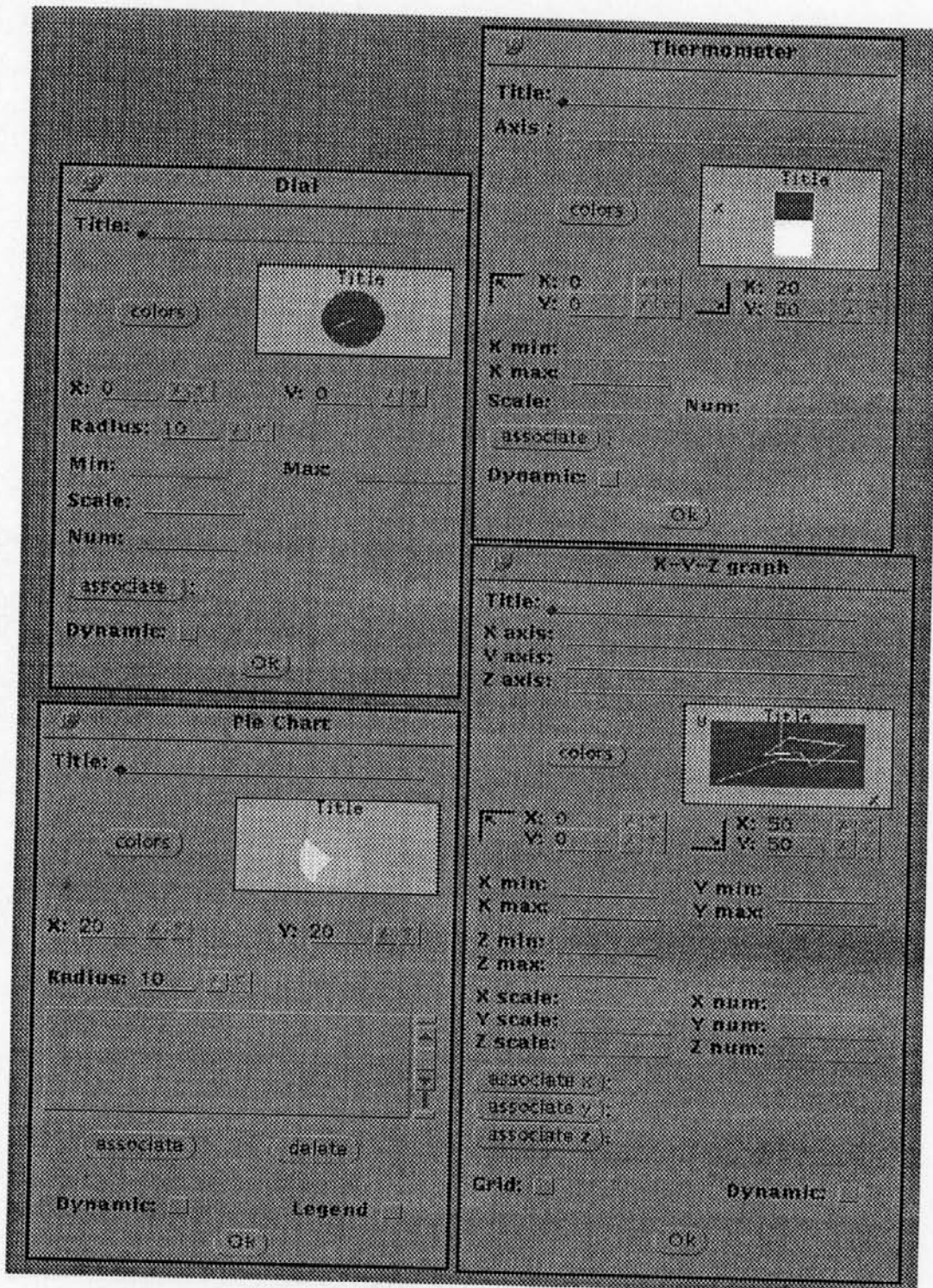


Figura 4.9- Janela Dial, Thermometer, XYZ e Pie Chart

4.2.3 Controle das áreas de visualização ativas

O usuário pode selecionar quais áreas de visualização devem permanecer ativas. Este controle é feito ativando-se o botão de controle das áreas ativas (figura 4.10) na interface principal.

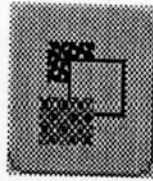


Figura 4.10- Botão de controle das áreas

Quando ativado o botão de controle das áreas, é aberta a janela *Active Windows* (figura 4.11) que possui uma lista das áreas já criadas. Selecionando-se uma área nesta lista esta será ativada ou desativada, pois este controle comporta-se como uma chave liga/desliga.

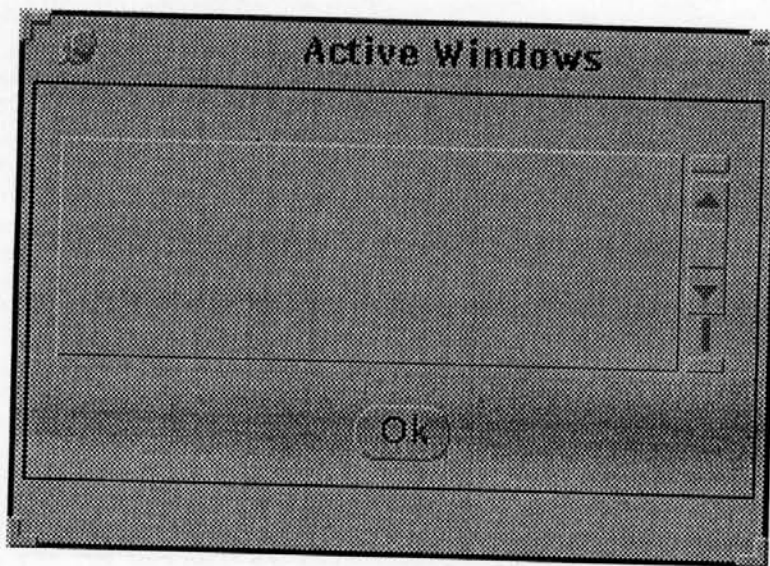


Figura 4.11- Janela Active Windows

4.3 Os recursos de interação

O ambiente VISE permite que o usuário mantenha as variáveis e atributos de entidades do modelo que está simulando sob seu controle, seja pela visualização de seus valores em qualquer momento da execução, seja pela alteração desses valores assim que o usuário julgar conveniente. O usuário pode também alterar os tipos de distribuições de probabilidade empregados no modelo bem como seus parâmetros.

4.3.1 Manipulação de variáveis e atributos de entidades

Para ter-se acesso às variáveis e atributos de entidades do modelo deve-se selecionar o botão correspondente (figura 4.12) na interface principal (figura 4.1).

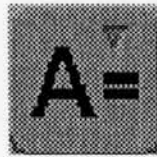


Figura 4.12- Botão de acesso a variáveis e atributos de entidades

A partir do momento que o botão de acesso a variáveis e atributos de entidades é selecionado é apresentado um cardápio com as diversas operações possíveis e cujas ações são descritas a seguir:

Read Visualizar o conteúdo atual de uma variável ou atributo de entidade.

Write Carregar uma determinada variável ou atributo com um valor a ser fornecido pelo usuário.

Force Fixar o valor de uma variável ou atributo não permitindo sua alteração posterior através do modelo, a não ser que o usuário assim o deseje.

Release Liberar uma variável ou atributo fixado pela operação *Force*.

Depois de realizada a seleção da operação é ativada uma janela intitulada *Accessing Variables & Entity Attributes* (figura 4.6). Nesta janela é apresentada uma lista que pode apresentar as variáveis globais ou os atributos das entidades do modelo, dependendo da seleção que é feita no botão de chaveamento que se encontra na parte superior desta janela. Ao lado deste botão é indicada a operação que está em curso. Para efetuar a escolha da variável ou atributo basta selecioná-la na lista com o mouse.

Para as variáveis que são matrizes é ativada em seguida a janela *Arrays* (figura 4.13) na qual deve-se especificar qual o índice da variável e o valor do mesmo, para que se possa determinar com qual elemento da matriz se está trabalhando.

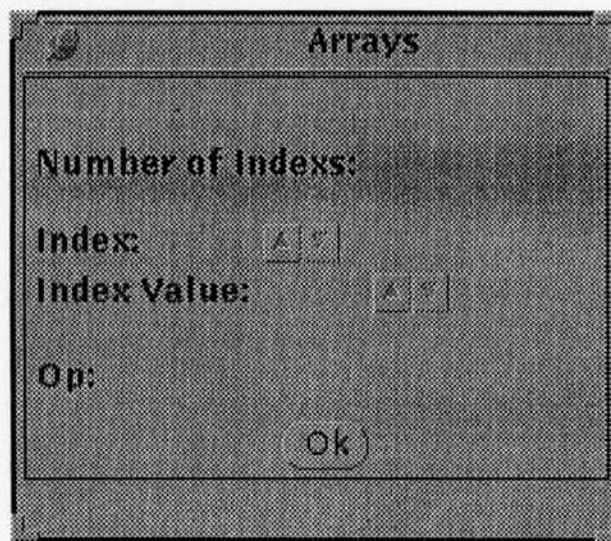


Figura 4.13- Janela Arrays

Caso as operações em andamento forem *Write* e *Force* será ativada então a janela *Value* (figura 4.14) na qual o usuário deverá entrar com o valor desejado.

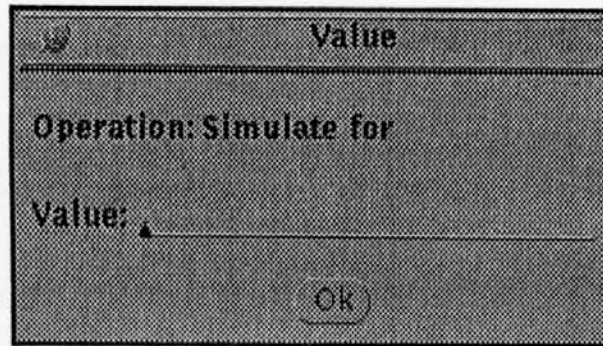


Figura 4.14- Janela Value

Se a operação for *Read* vai ser aberta a janela *Show Value* (figura 4.15) na qual será exibido o conteúdo corrente da variável.

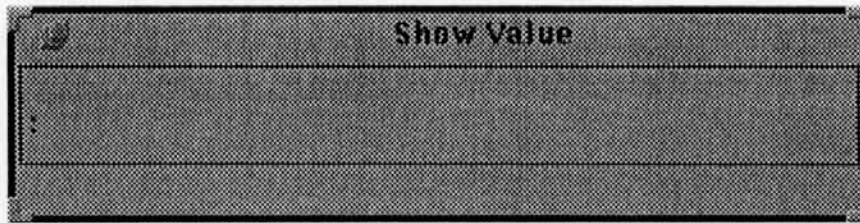


Figura 4.15- Janela Show Value

4.3.2 Monitoração de variáveis e atributos de entidades

O ambiente permite monitorar variáveis e atributos de entidades, o que é realizado através do controle de uma lista dos ítems que estão sendo monitorados. A monitoração consiste na apresentação para o usuário do conteúdo atual dos ítems monitorados, sendo possível também observar o comportamento de uma variável ao longo do tempo.

A monitoração é controlada a partir da seleção do botão de monitoração (figura 4.16) na interface principal.

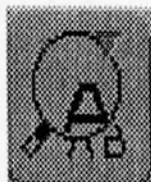


Figura 4.16- Botão de monitoração

Depois de selecionado este botão, é ativado um cardápio com as seguintes opções:

Monitoration Monitora uma variável ou atributo de entidade.

Screen list Exibe lista de variáveis e atributos que está sendo monitorada na tela.

Disk list Exibe lista de variáveis e atributos que está sendo monitorada em disco. Tendo sido escolhida a opção *Monitoration* é aberto um sub-cardápio com as opções para monitoração na tela ou em disco. Depois de selecionada a opção é ativada a janela *Accessing Variables & Entity Attributes* (figura 4.6) na qual deve ser selecionada a variável ou atributo de entidade que será monitorado.

Caso a opção de listagem na tela tenha sido selecionada, será ativada a janela *Monitoration on Screen* (figura 4.17) que exibirá as variáveis e atributos monitorados em tela com seus respectivos valores. Nesta janela é possível também liberar uma das variáveis ou atributos através da ativação do botão *Release*.

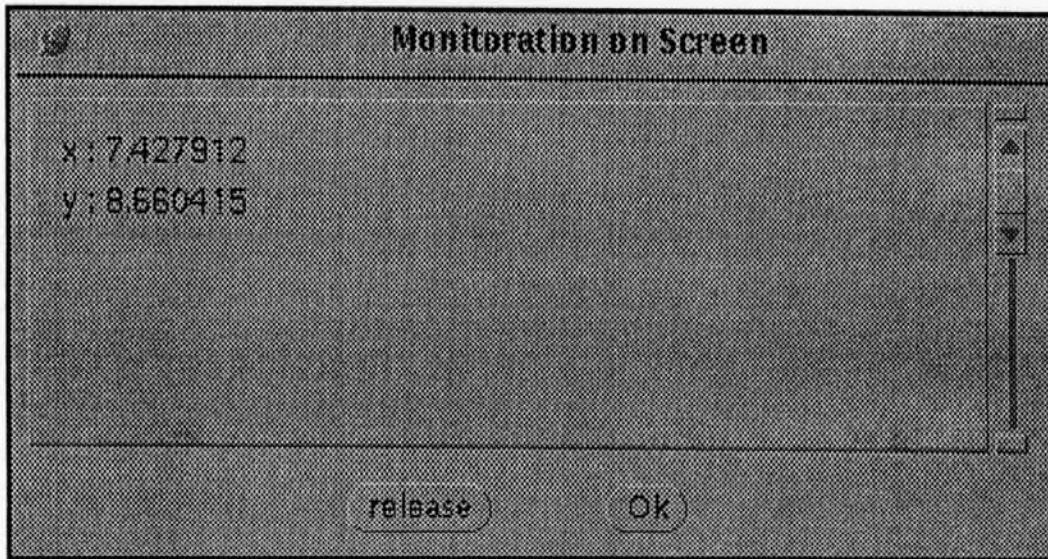


Figura 4.17- Janela Monitoration on Screen

Se a opção escolhida foi listagem em disco, será aberta a janela *Monitoration on Disk* (figura 4.18), que exibirá as informações referentes ao que está sendo monitorado em disco. As variáveis e atributos de entidades monitorados em disco são também monitorados em tela, portanto é possível liberar qualquer uma das variáveis ou atributos por meio da ativação do botão *Release* da janela *Monitoration on Screen*.

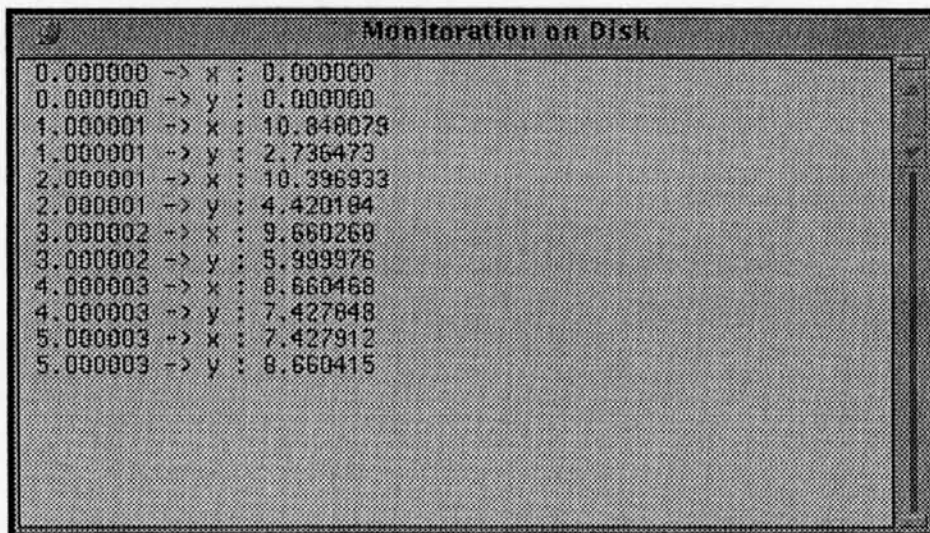


Figura 4.18- Janela Monitoration on Disk

A monitoração em disco permite que se tenha um histórico do conteúdo de uma variável ou atributo ao longo da simulação enquanto que a monitoração em tela só exibe o seu valor atual, isto é, possui um caráter instantâneo.

4.3.3 Manipulação das distribuições de probabilidade

É possível alterar o tipo e os parâmetros das distribuições de probabilidade usadas no modelo. Isto só é possível se for usada no código SIMSCRIPT a função *vise distribution* ao invés das funções de distribuição próprias da linguagem.

Esta função tem como primeiro parâmetro o tipo de distribuição seguido dos demais parâmetros, que vão depender do tipo de distribuição desejada. Esta classe de recurso é útil nas fases de verificação, validação e experimentação do modelo, pois permite o direcionamento ("steering") de uma sessão de simulação.

Para modificar uma distribuição de probabilidade basta selecionar o botão de distribuições de probabilidade (figura 4.19) na interface principal.

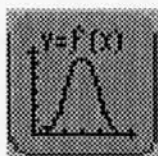


Figura 4.19- Botão de distribuições de probabilidade

Em seguida é ativada a janela *Probability Distributions Calls* (figura 4.20) na qual são exibidos trechos do código SIMSCRIPT onde estão as chamadas da função *vis distribution*. Pode-se selecionar a chamada que deseja-se alterar usando as setas indicativas de navegação na parte inferior da janela. Tendo-se selecionado a chamada, é possível alterar suas características acionando o botão *Edit*, localizado à direita das setas.

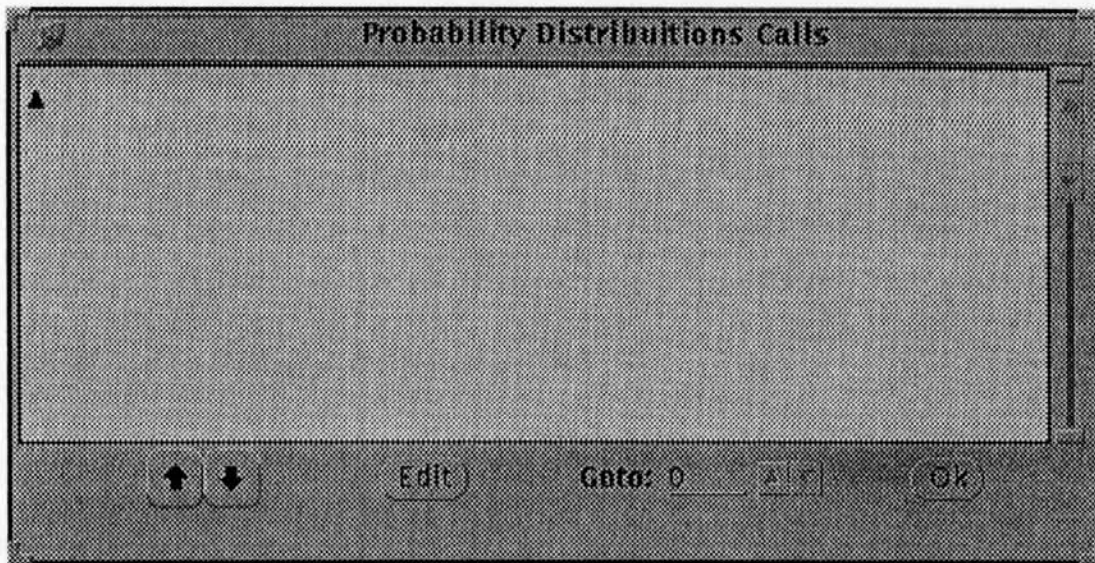


Figura 4.20- Janela Probability Distribution Calls

A partir deste instante será ativada a janela *Probability Distribution Classes* (figura 4.21), na qual será feita a edição dos parâmetros da chamada da função.

Na parte superior desta janela há uma lista de botões contendo as classes de distribuições implementadas que podem ser selecionadas através do apontamento e seleção com o mouse. Após isto, deve-se atribuir valores aos parâmetros, o que pode ser feito através da seleção de um parâmetro na lista de parâmetros que localiza-se na região central da janela. Depois de selecionado o parâmetro cabe atribuir-lhe um valor, havendo para este fim um campo logo abaixo da lista de parâmetros.

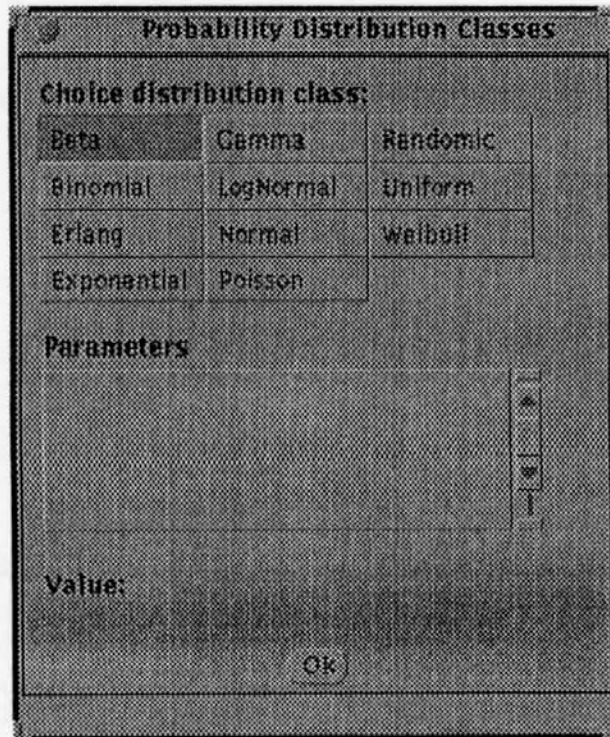


Figura 4.21- Janela Probability Distribution Classes

4.3.4 Escolha da unidade de tempo da simulação

Para alterar-se a unidade de tempo usa-se o botão de unidade de tempo (figura 4.22) na interface principal. Esta alteração se refere unicamente a unidade de tempo usada na interação com o usuário, não havendo modificação na unidade de tempo do modelo.

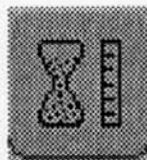


Figura 4.22- Botão de unidade de tempo

Uma vez ativado o botão, é aberta a janela *Simulation Unit* (figura 4.23) na qual pode-se escolher a nova unidade de simulação.

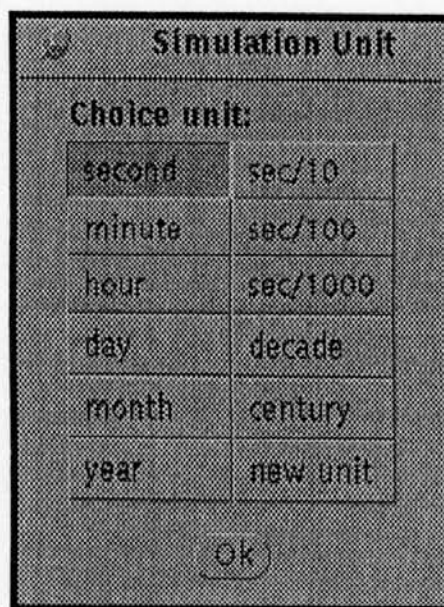


Figura 4.23- Janela Simulation Unit

Pode-se criar uma nova unidade de tempo usando-se a opção *new unit* desta janela. Neste caso será ativada uma nova janela chamada *New Simulation Unit* (figura 4.24) onde uma nova unidade será especificada. Para esta nova unidade deve ser especificado um nome pela qual ela será designada. O usuário pode determinar que a nova unidade seja um múltiplo de uma unidade já existente. Para isso ele deve selecionar uma das unidades que aparecem nesta janela e em seguida fornecer o fator de multiplicação.

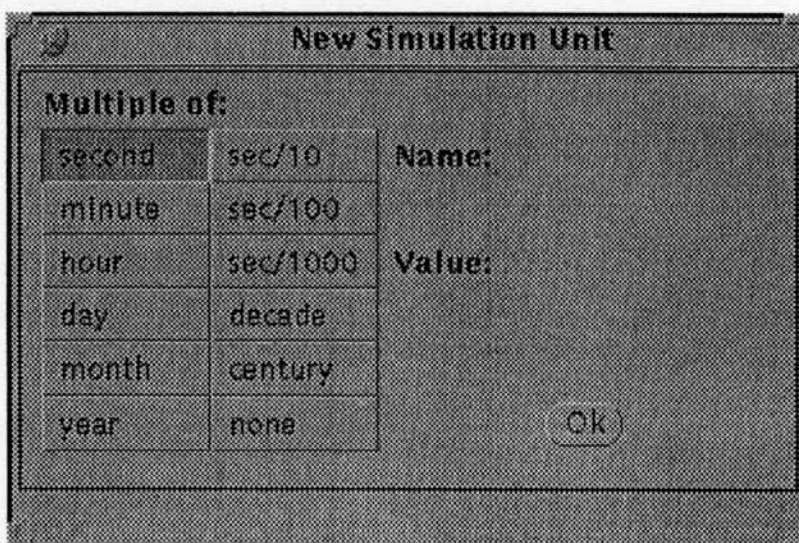


Figura 4.24- Janela New Simulation Unit

4.4 Os recursos de controle da execução

Estão disponíveis diversos mecanismos para o controle da execução, entre eles a escolha do modo de execução, controle de pontos de parada (*breakpoints*), controle da lista de eventos, disparo e suspensão da execução, exibição do status da simulação e operações de carga e salvamento da simulação.

4.4.1 Escolha do modo de execução

Para selecionar o modo de execução deve-se acionar o botão de modos de execução (figura 4.25) na interface principal.

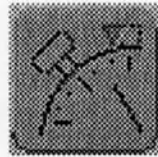


Figura 4.25- Botão de modos de execução

Depois de acionado este botão é exibido um cardápio contendo os modos existentes, a saber:

For Executa por um determinado número de unidades de tempo.

Until Executa até a unidade de tempo especificada.

Next event Executa até a ocorrência do evento especificado.

Step by step Executa o código linha a linha.

Se o modo escolhido foi *For* ou *Until* é ativada a janela *Value* onde será especificado por quantas unidades de tempo a execução prosseguirá (modo *For*) ou até qual unidade de tempo haverá execução (modo *Until*). Caso tenha sido selecionado o modo *Next event* será aberta a janela *Event Classes* (figura 4.26) que contém uma lista dos eventos do modelo.

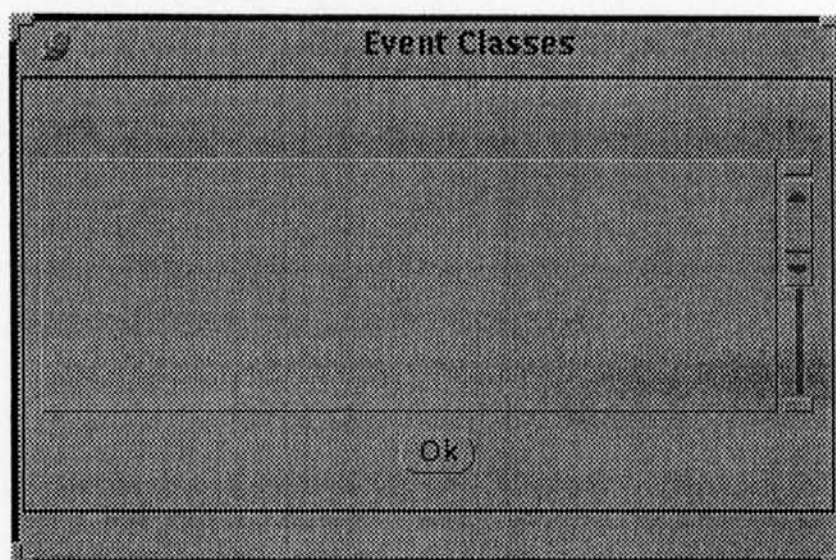


Figura 4.26- Janela Event Classes

Selecionando-se um dos eventos desta lista, a execução ocorrerá sem pausas (a não ser que hajam pontos de parada especificados) até a ocorrência do evento escolhido.

A opção de execução *Step by step* não ativa nenhuma janela imediatamente. A janela correspondente só é ativada no momento em que a execução da simulação é reiniciada, o que é feito com o botão de execução. Uma vez disparada a simulação, e estando-se no modo *Step by step*, é aberta a janela *Execution Step by Step* (figura 4.27) onde é exibido o trecho do código SIMSCRIPT com a linha a ser executada exibida em fundo reverso. O avanço da execução se dá linha a linha através do acionamento por mouse do botão *Next* localizado na parte inferior desta janela. Pode-se ainda disparar a simulação de maneira que seja exibida sempre a próxima linha a ser executada mas sem as paradas para confirmação de avanço. Isto é possível através do botão *Run* localizado à direita do botão *Next*.

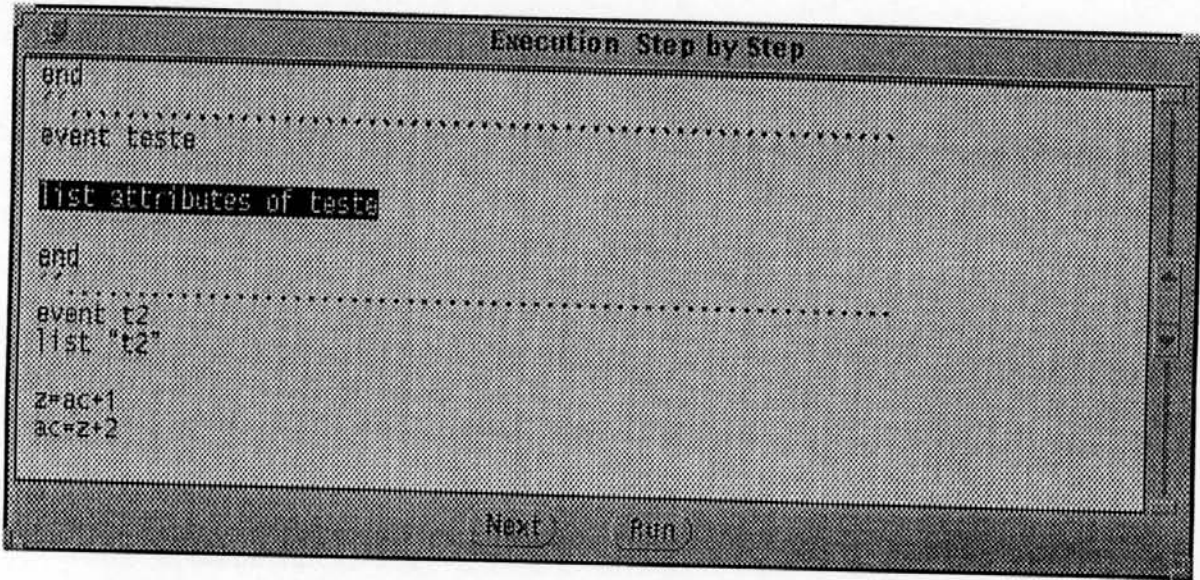


Figura 4.27- Janela Execution Step by Step

4.4.2 Determinação dos pontos de parada

O ambiente VISE possibilita estabelecer pontos de parada (*breakpoints*), recurso bastante útil na fase de depuração do código. Pode-se determinar pontos de parada acionando-se o botão de pontos de parada (figura 4.28) na interface principal.

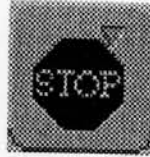


Figura 4.28- Botão de pontos de parada

A partir deste instante é acionado um cardápio apresentando os 2 tipos de parada existentes: parada condicional e parada por ocorrência de determinado evento.

Condition Neste tipo a execução é suspensa caso uma determinada condição especificada pelo usuário na forma de uma expressão torne-se verdadeira. Selecionando-se a opção de parada por condição passa-se para a janela *Conditional Breakpoints* (figura 4.29) onde pode-se montar uma condição de parada. Na parte superior da janela há uma lista que pode exibir variáveis globais ou atributos de entidades (isto depende da seleção feita com o botão de chaveamento que encontra-se acima desta lista). Abaixo desta lista há um conjunto de opções relativas às operações passíveis de serem usadas na montagem da expressão de parada. A montagem da expressão é feita selecionando-se variáveis ou atributos, operações e introduzindo valores no campo de montagem da expressão que encontra-se na parte central da janela. Concluída a montagem, a expressão deve ser inserida na lista de pontos de parada. Esta lista encontra-se logo abaixo do campo de expressão, e a inserção é efetivada através do botão *insert* que encontra-se abaixo desta lista. Pode-se remover expressões desta lista através do botão *delete* e ativar ou desativar determinada expressão, o que é realizado pelo botão *on/off*.

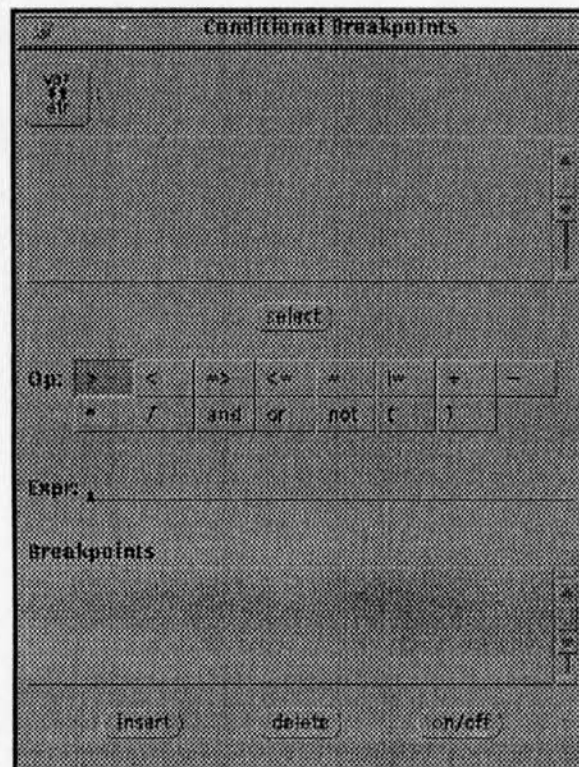


Figura 4.29- Janela Conditional Breakpoints

Event A simulação é interrompida se há a ocorrência de um evento especificado pelo usuário. Selecionando-se esta opção é ativada a janela *Event Classes* (figura 4.26) que contém a lista de eventos do modelo, bastando selecionar o evento desejado como condição de parada. Este recurso também é acessível a partir do botão de modos de execução.

4.4.3 Controle da lista de eventos

Este controle permite exibir e manipular a lista de eventos interna do SIMS-CRIPT, permitindo ao usuário alterar a ordem de programação (*scheduling*) dos eventos assim como outros atributos dos mesmos. Para ativar a janela responsável pelo controle da lista seleciona-se o botão de controle da lista de eventos (figura 4.30) na interface principal.

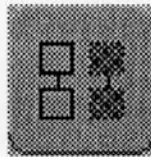


Figura 4.30- Botão de controle da lista de eventos

Esta janela é chamada *Event List* (figura 4.31) e exibe a lista de eventos.

Os eventos são representados por retângulos e estão organizados segundo sua classe; eventos da mesma classe possuem a mesma cor e encontram-se dispostos na mesma coluna, sendo que os eventos mais acima são os que ocorrerão primeiro na classe. Pode-se controlar o tamanho dos retângulos através dos botões de ampliação e redução localizados na região inferior esquerda da janela. A seleção de um evento se dá através do botão *select* que ativará a janela *Event Access* (figura 4.32), onde é possível visualizar e editar os atributos de um evento.

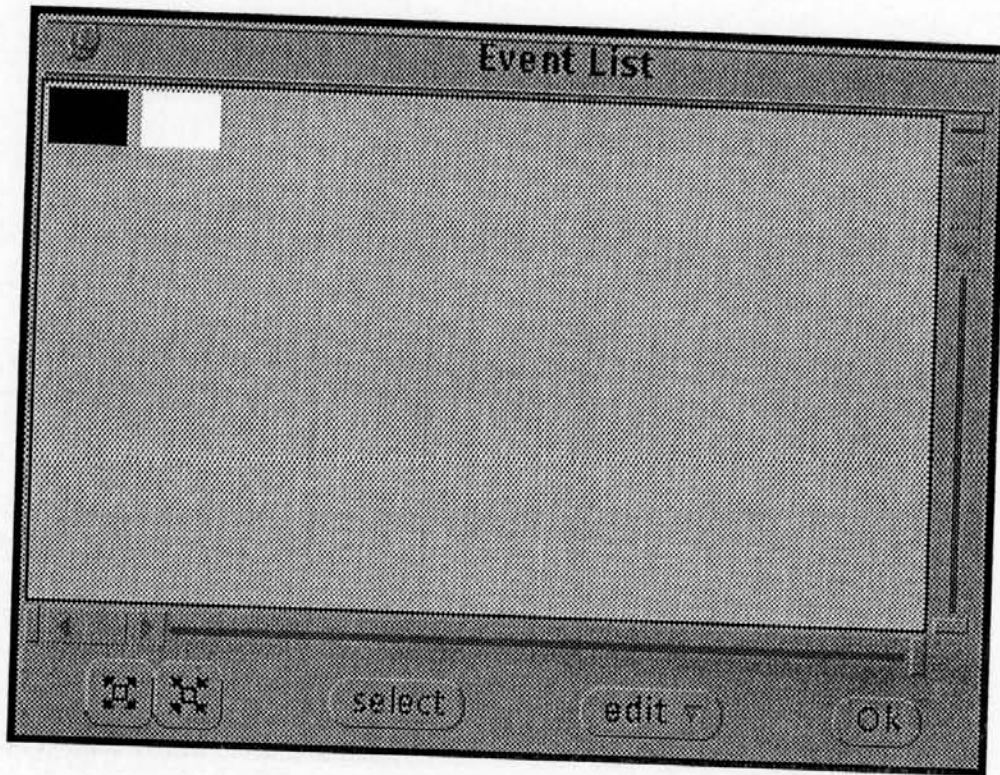


Figura 4.31- Janela Event List

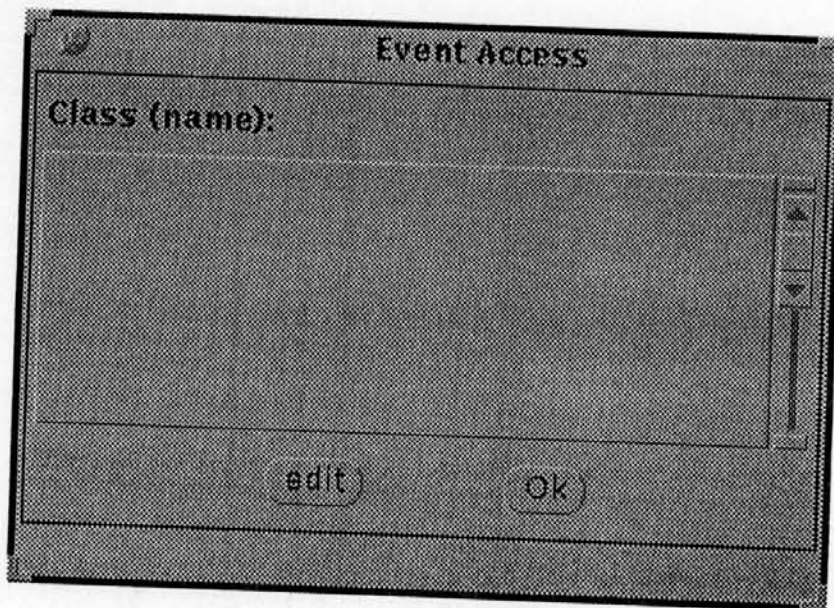


Figura 4.32- Janela Event Access

Os atributos de um evento são os seguintes:

- tempo de programação;
- ponteiro para o evento predecessor na lista;
- ponteiro para o evento sucessor na lista;
- os atributos definidos pelo usuário.

Ainda no controle da lista é permitido que haja remoção e inserção de eventos. Isto pode ser feito selecionando-se um evento usando-se o botão *edit* na janela *Event list* que apresentará um cardápio com estas opções assim que ativado.

Tendo sido escolhida a opção de inserção de eventos é exibido um sub-cardápio com as opções de inserir o evento na lista antes ou depois do evento selecionado. A seguir é aberta a janela *Event Access* onde o usuário entrará com os atributos do novo evento.

4.4.4 Disparo e suspensão da execução

Para disparar a execução da simulação usa-se o botão de disparo (figura 4.33) na interface principal. Ao lado deste botão há outro que por sua vez é responsável pela suspensão da execução (figura 4.34).

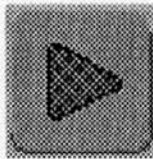


Figura 4.33- Botão de disparo da execução

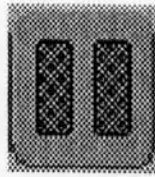


Figura 4.34- Botão de pausa na execução

4.4.5 Status da simulação

O recurso de status permite observar a seqüência das chamadas das funções, rotinas e eventos do modelo. Além disso traça um perfil estatístico em relação ao número de chamadas de cada função, rotina e evento. A janela de status é intitulada *Execution Status* (figura 4.35) e é ativada pelo botão de status (figura 4.36) localizado na interface principal.

Esta janela exibe na sua região superior a seqüência das chamadas enquanto que na sua parte inferior é exibido o perfil estatístico das funções, rotinas e eventos do modelo.

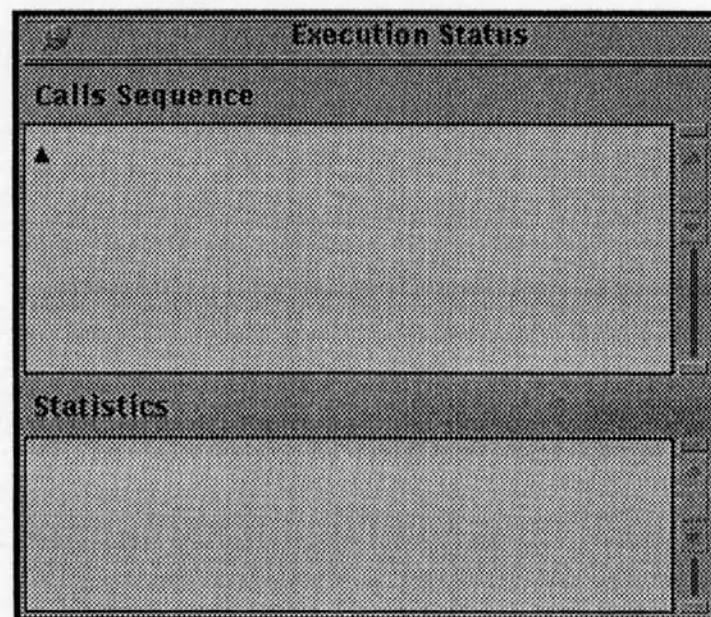


Figura 4.35- Janela Execution Status



Figura 4.36- Botão de Status

No perfil estatístico são exibidas as informações referentes ao número de vezes que cada função, rotina ou evento foi chamada e a porcentagem relativa ao número total de chamadas.

4.4.6 Operações de carga e salvamento da simulação

O ambiente dá suporte à carga, salvamento e salvamento periódico da simulação. Estas operações são ativadas pelo botão de carga e salvamento (figura 4.37) na interface principal.

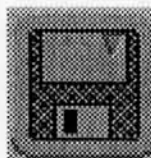


Figura 4.37- Botão de carga e salvamento

Depois de ativado este botão é apresentado um cardápio com as seguintes opções:

Save simulation Grava as informações referentes à simulação em curso, como o valor corrente das variáveis e atributos de entidades, informações sobre a lista de eventos, esquema de visualização em uso, as associações entre variáveis e representações gráficas, ícones, cenários, etc.

Load simulation Carrega para a memória as informações referentes a uma sessão de simulação previamente salva.

Save periodically Grava as informações referentes à simulação de tempos em tempos, sendo que se esta foi a opção escolhida, é ativada a janela *Value* para que o usuário informe o intervalo em unidades de tempo entre cada operação de salvamento.

Depois de selecionada uma opção é ativada a janela *Files* onde o usuário informará o nome do arquivo em uso.

4.4.7 Abandonando o ambiente

Para encerrar a sessão de simulação e abandonar o ambiente basta ativar o botão de saída do ambiente (figura 4.38) na interface principal.

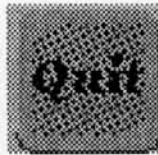


Figura 4.38- Botão de saída do ambiente

É emitido um aviso (figura 4.39) solicitando a confirmação da operação por parte do usuário. Dada a confirmação, todos os arquivos relativos à sessão são fechados e a sessão é abandonada.

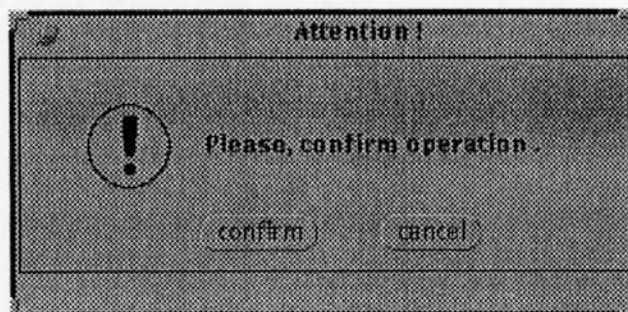


Figura 4.39- Janela Attention

5 CONCLUSÕES E TRABALHOS FUTUROS

O desenvolvimento do protótipo do ambiente mostrou a viabilidade de acrescentar recursos de interação, controle e visualização ao SIMSCRIPT, estando no momento implementados mais da metade dos recursos propostos.

SIMSCRIPT, apesar de ser uma poderosa linguagem de simulação, possui algumas limitações, tais como um ambiente textual para modelagem, poucos recursos para depuração do modelo, a necessidade de se declarar explicitamente no código comandos para visualização, elaboração pelo próprio usuário das rotinas de monitoração, baixo grau de interatividade com o simulacionista durante a execução, impossibilidade de se alterar o aspecto das representações gráficas empregadas no modelo, impossibilidade de modificação da associação entre a representação gráfica e a variável ou atributo de entidade durante a execução e visualização restrita a um contexto bidimensional.

O ambiente proposto oferece recursos de interação, controle e visualização. Os recursos de interação possibilitam ao usuário manipular o conteúdo de variáveis e atributos de entidades além de permitir modificar as distribuições de probabilidade usadas no modelo.

Diversos recursos de controle são oferecidos, entre eles a suspensão e reativação da execução, acesso à lista de eventos, especificação de pontos ou condições de parada e diferentes modos de execução.

A classe de recursos de visualização possibilita a especificação de associações de variáveis e atributos de entidades a ícones e gráficos estatísticos a fim de que se possa visualizar de maneira gráfica o andamento da simulação.

As funções de execução passo a passo, especificação de pontos de parada, acesso e monitoração de variáveis e atributos de entidades são bastante úteis na fase de verificação do modelo. Os recursos de visualização encontram aplicação na fase de validação e também na verificação.

As fases de planejamento e experimentação do processo de simulação também são beneficiadas tendo em vista o maior controle que o usuário tem sobre a execução da simulação.

Além disso a facilidade de visualização do modelo possibilita uma melhor interpretação dos resultados da simulação. Os recursos oferecidos pelo ambiente VISE também são aplicáveis quando adotada a abordagem hierárquica de animação na modelagem de simulação.

No anexo A-1 é apresentado um pequeno exemplo onde foram testados os recursos de visualização do modelo. A listagem do programa usado para o exemplo encontra-se no anexo A-2.

Uma das maiores restrições que este ambiente possui é não possibilitar a alteração do modelo em tempo de execução. Isto se deve ao fato de que uma modificação no código implicaria numa recompilação deste e em uma nova ligação com o código objeto do ambiente, o que só pode ser feito se o processo de execução for interrompido e um novo executável for gerado.

Esta é uma limitação bastante séria pois se o simulacionista ao longo da execução constatar que o seu modelo não é adequado ou for incompleto e desejar fazer modificações no mesmo será necessário abortar a simulação, realizar as alterações no código SIMSCRIPT, recompilá-lo e ligá-lo novamente com o ambiente para só então reiniciar a simulação. Entretanto esta foi uma opção de projeto pois tinha-se em vista a construção de um ambiente de simulação visual interativa que fizesse uso da linguagem SISMSCRIPT.

Outra limitação importante é o formato empregado para os ícones bidimensionais e tridimensionais, pois nenhum deles segue algum padrão de mercado ou padrão internacional conhecido em função da complexidade de implementação destes padrões e do tempo considerável que a tarefa consumiria.

Optou-se por usar no caso de ícones tridimensionais o formato OBT criado pelo grupo de computação gráfica do Instituto de Informática da UFRGS. Quanto aos ícones bidimensionais foi criado um formato próprio chamado ico2D. O formato dos ícones bidimensionais e tridimensionais é descrito no anexo A-3.

Uma extensão possível mas ainda não implementada é a especificação de uma linguagem de roteiro de animação. Um roteiro de animação é interessante de ser empregado quando é necessário que durante a simulação haja alguma animação completamente independente de valores de atributos ou variáveis do modelo, não estando formalmente associada a qualquer elemento do mesmo (exceto o tempo de simulação). Este roteiro seria associado a uma entidade visual (ícone bidimensional ou tridimensional) cujo comportamento seria guiado por este roteiro.

Outra extensão interessante seria o projeto de um esquema de visualização específico para variáveis discretas, com elementos visuais tais como ondas quadradas, ou entidades visuais que assumiriam uma entre um número determinado de aparências possíveis em função do estado da variável associada.

Vários aperfeiçoamentos poderiam ser feitos no que se refere às funções de associação, como por exemplo:

1. permitir a associação das representações gráficas a expressões que envolvam tanto valores constantes fornecidos pelo usuário como variáveis, atributos de entidades e operadores matemáticos básicos, elaborando desta forma um mapeamento entre o valor da variável ou atributo e a representação gráfica associada;
2. possibilitar associação de representações gráficas com roteiros de animação escritos pelo usuário;

3. viabilizar a associação com eventos do mouse, de modo que utilizando este dispositivo o usuário possa apontar e selecionar o ícone diretamente na área de visualização e conforme o movimento executado com o mouse o ícone seria afetado em um ou mais de seus parâmetros, como sua posição na área por exemplo.

Seria interessante também permitir que o simulacionista pudesse especificar como pontos de parada determinadas linhas no código do modelo. A execução seria interrompida cada vez que uma destas linhas fosse atingida.

A janela responsável pelo controle da lista de eventos poderia ser melhorada se fosse elaborada uma legenda que apresentasse a correspondência entre as cores das classes e os seus nomes. Também seria interessante que o tempo de programação dos eventos aparecesse dentro dos retângulos correspondentes a cada evento.

Finalmente, poderia ser implementado um recurso para pós-processamento, onde cada imagem gerada em uma determinada área de visualização seria gravada em um arquivo correspondente toda vez que a área sofresse uma alteração. Desta forma seria possível elaborar uma animação com todas as imagens geradas pela simulação em uma determinada área de visualização após o término da execução.

ANEXO A-1 PROGRAMA DE TESTE DO PROTÓTIPO

Para realizar uma análise dos resultados obtidos até a versão atual da implementação foi elaborado um programa escrito em SIMSCRIPT que foi usado como teste do protótipo do ambiente.

A listagem deste programa é apresentada no anexo A-2. O seu objetivo é analisar o desempenho dos postos de atendimento de um pedágio de uma estrada. Este pedágio possui dois postos, sendo que o tempo levado por cada um deles para atender um carro é de 30 segundos com um desvio-padrão de 1 segundo. Os carros chegam ao pedágio a uma taxa de 1 carro a cada 20 segundos com desvio-padrão de 5 segundos. O tempo total de simulação do exemplo é de uma hora.

Este programa possui 3 três rotinas de eventos:

- gera_carro
- libera_F1
- libera_F2

O evento `gera_carro` é responsável pela geração da entidade temporária `carro`. A medida que um carro surge este é colocado numa das duas filas do sistema (uma fila para cada posto de atendimento), sendo que o carro é sempre inserido na menor fila. Este evento se auto-programa para ocorrer num tempo futuro distante 20 segundos (com desvio-padrão de 5 segundos) do tempo atual de simulação, enquanto o valor do tempo futuro for inferior a uma hora. Os eventos `libera_F1` e `libera_F2` são responsáveis pela retirada de um carro da fila do respectivo posto de atendimento e são programados pelo evento `gera_carro`.

Este modelo foi elaborado usando-se a abordagem hierárquica para animação. No primeiro nível, que pode ser observado na figura A-1.1, foram empregados dois gráficos do tipo termômetro, cada um associado a uma variável que armazenava o número de carros em cada fila (fil1 e fil2). Os demais recursos que aparecem na figura são a execução passo a passo, a monitoração de variáveis na tela e em disco e a visualização da lista de eventos.

Para o nível seguinte (ver figura A-1.2) foram elaborados um gráfico XY da média de carros por minuto que chegavam no primeiro posto contra a variável que continha o tempo atual de simulação, um histograma da variável desloc, que contém o tempo em segundos da próxima chegada de um carro ao pedágio e dois gráficos do tipo dial associados às variáveis ac1 e ac2 que representam o número total de carros que já entraram em cada uma das filas.

Finalmente, para o último nível (figura A-1.3) foi usada uma representação bidimensional do pedágio, onde pode-se observar o mesmo visto de cima com um carro deslocando-se da direita para a esquerda ao longo da estrada em direção aos postos de atendimento. Além desta representação foi empregado um gráfico de torta associado a variáveis que contêm a média de carros por minuto que chegam a cada posto e um gráfico tipo relógio associado à variável que contém o tempo atual de simulação.

Caso não fosse utilizado o ambiente VISE neste programa teste, seria necessário realizar modificações no código a fim de serem inseridos comandos de animação da biblioteca SIMGRAPHICS.

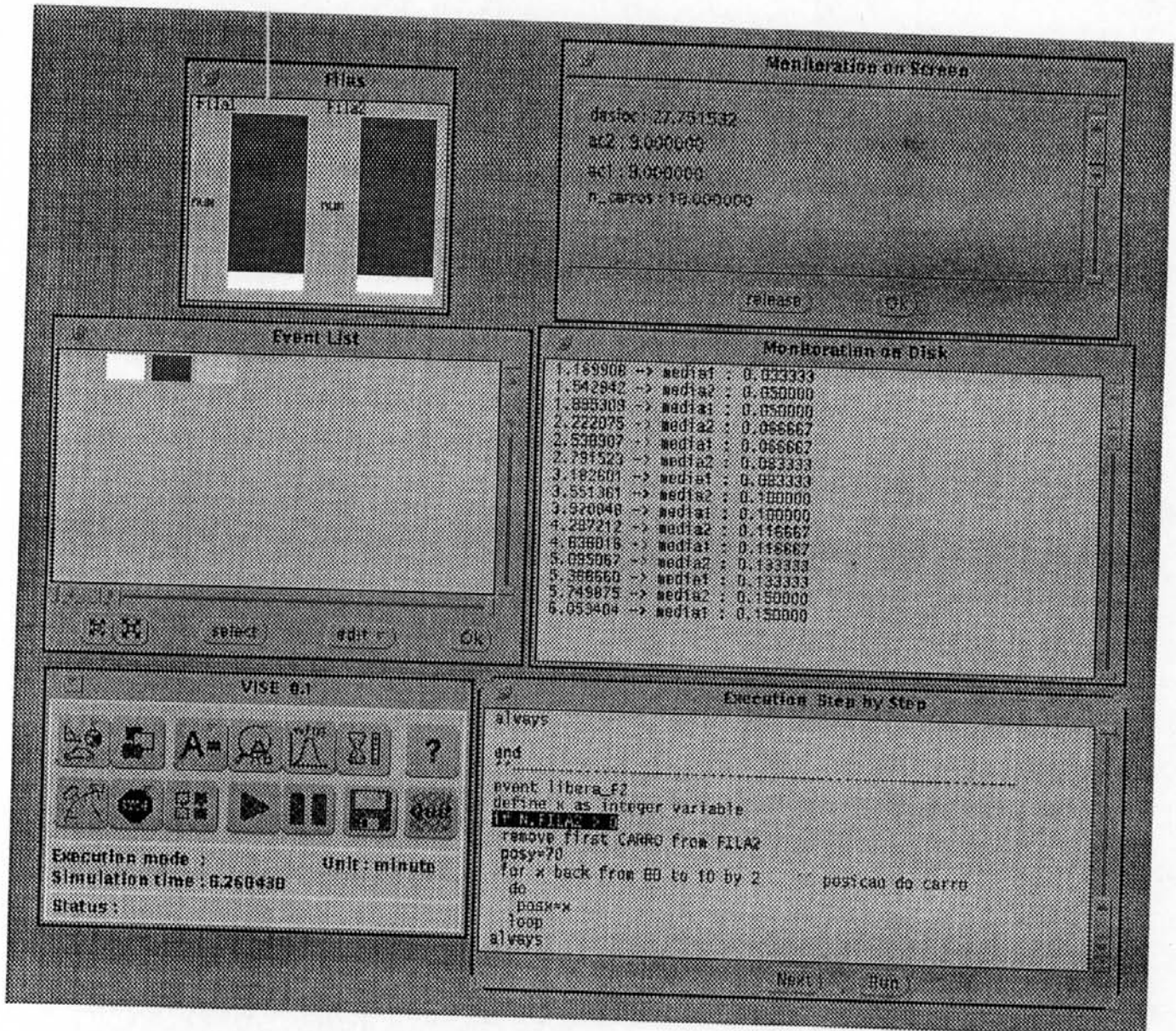


Figura A-1.1- Execução do programa teste, nível 1

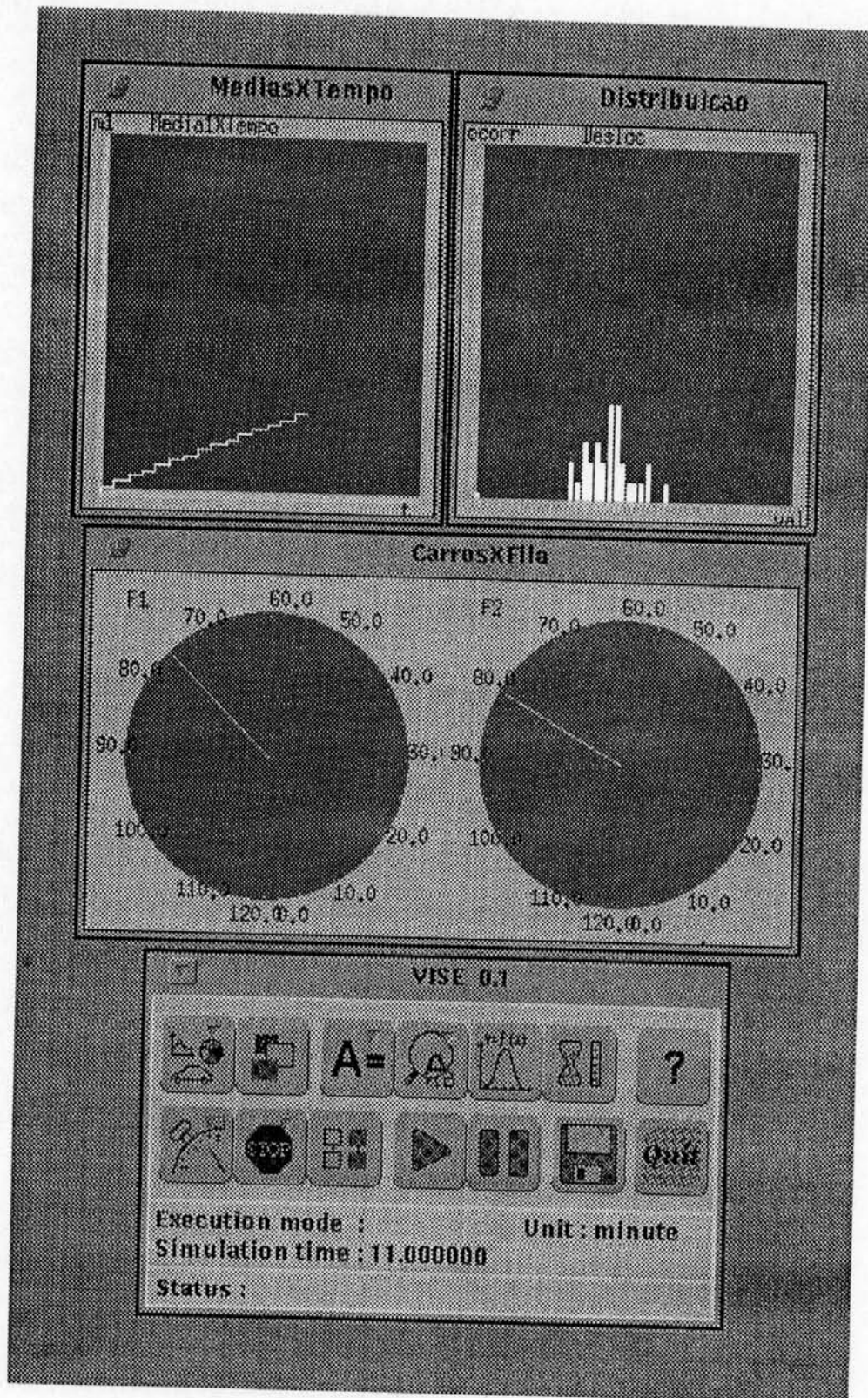


Figura A-1.2- Execução do programa teste, nível 2

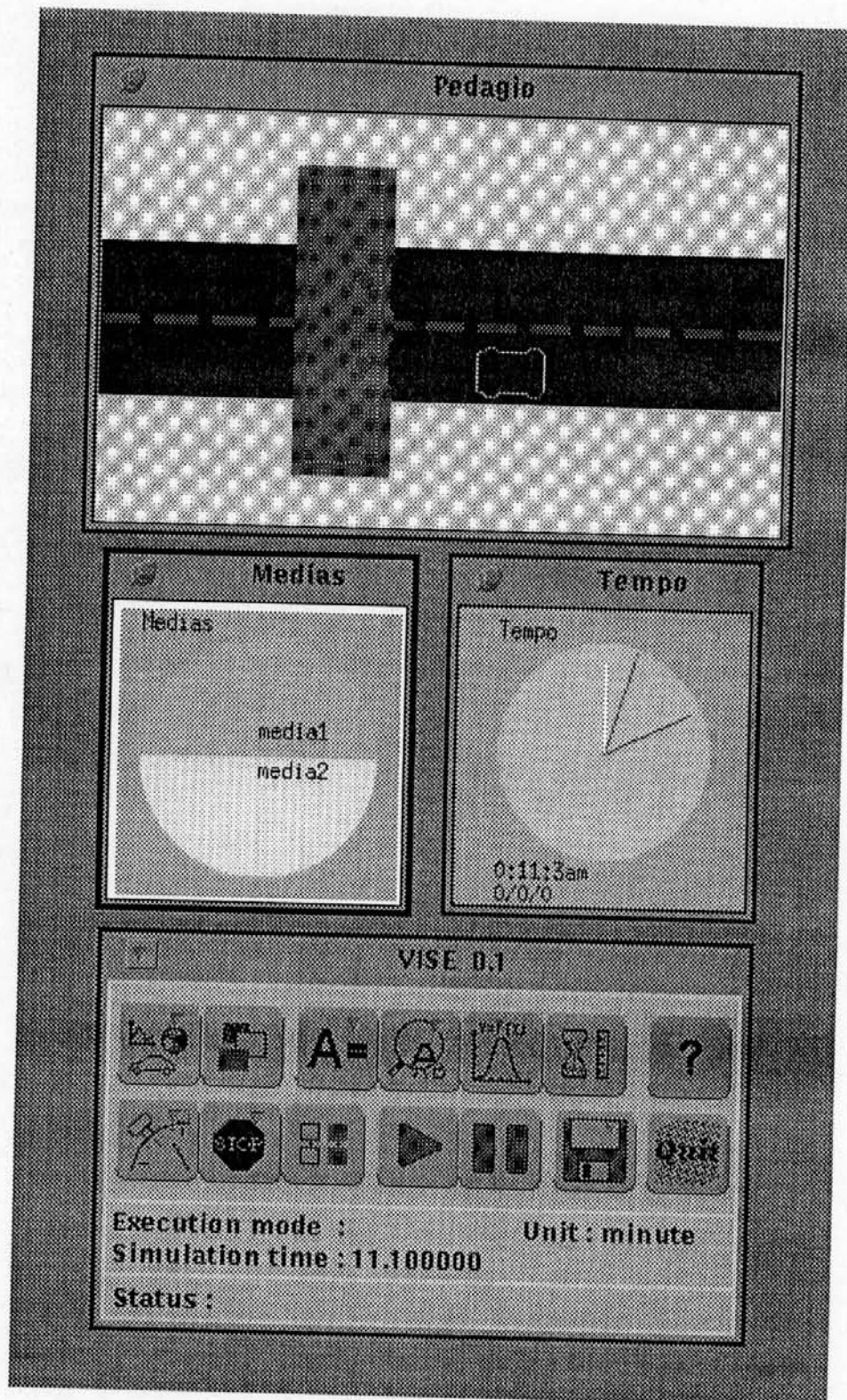


Figura A-1.3- Execução do programa teste, nível 3

ANEXO A-2 LISTAGEM DO PROGRAMA DE TESTE DO PROTÓTIPO

```

'' Modelo de Pedagio
'' Dissertacao de Mestrado - Ernesto Lindstaedt
'' Julho de 1995

preamble

normally , mode is integer

the system owns a FILA1 ,
    and owns a FILA2

event notices include gera_carro,libera_F1,libera_F2

Temporary entities

every CARRO has a tempo ,
    a numero ,
    and may belong to a FILA1 and may belong to a FILA2

define FILA1 as a set
define FILA2 as a set

define SEG to mean /60 minutes

define n_carros,ac1,ac2,posx as integer variables
define desloc as real variable
define proporcao1,proporcao2 as real variables
define media1,media2 as real variables
define tempo_simulacao,tempo_atual as real variable
define fil1,fil2 as real variables
end

''-----

main

desloc=0
n_carros=0
ac1=0
ac2=0
fil1=0
fil2=0
'' 0.0000115972 = 1 segundo
'' 0.000694444 = 60 segundos
'' 0.0417 = 60 minutos (1 hora)
tempo_simulacao=0.0417

N.FILA1=0
N.FILA2=0

schedule a gera_carro now

start simulation

tempo_atual=time.v * 86400 '' em segundos

proporcao1=ac1/n_carros
proporcao2=ac2/n_carros

media1=ac1/60 '' qtos carros por minuto
media2=ac2/60 '' qtos carros por minuto

end

```



```

event gera_carro
define x as integer variable

create CARRO

tempo(CARRO)=time.v * 86400 `` em segundos
n_carros=n_carros+1
numero(CARRO)=n_carros

for x back from 320 to 190 by 4 `` posicao do carro
do
  if N.FILA1 < N.FILA2
    posy=109
  else posy=70
  always
  posx=x
loop

if N.FILA1 < N.FILA2
  file CARRO in FILA1
  ac1=ac1+1
  media1=ac1/60
  schedule a libera_F1 in normal.f(30,1,1) SEG
else file CARRO in FILA2
  ac2=ac2+1
  media2=ac2/60
  schedule a libera_F2 in normal.f(30,1,1) SEG
always
fil1=N.FILA1
fil2=N.FILA2
tempo_atual=time.v * 86400
if time.v < tempo_simulacao `` 1 hora
  desloc= normal.f(20,5,1)
  if desloc+(time.v*86400) <= tempo_simulacao*86400
    schedule a gera_carro in desloc SEG
  always
always

end
``-----
event libera_F1
define x as integer variable
if N.FILA1 > 0
  remove first CARRO from FILA1
  posy=109
  for x back from 80 to 10 by 2 `` posicao do carro
  do
    posx=x
  loop
always

end
``-----
event libera_F2
define x as integer variable
if N.FILA2 > 0
  remove first CARRO from FILA2
  posy=70
  for x back from 80 to 10 by 2 `` posicao do carro
  do
    posx=x
  loop
always

end

```

ANEXO A-3 DESCRIÇÃO DOS FORMATOS DE ÍCONES EMPREGADOS PELO AMBIENTE

A-3.1 O formato OBT

A seguir é descrito o formato para ícones tridimensionais:

```
$OBJETO <nome do objeto> <x> <y> <z> <raio>
```

```
$FACES <numero de faces do objeto>
```

```
<numero de vertices face 1> <vertice i> ... <vertice j>
```

```
<numero de vertices face m> <vertice k> ... <vertice l>
```

```
$VERTICE <numero de vertices do objeto>
```

```
<coord. X vertice 1> <coord. Y vertice 1> <coord. Z vertice 1>
```

```
<coord. X vertice n> <coord. Y vertice n> <coord. Z vertice n>
```

```
$ENDOBJ
```

A-3.2 O formato ico2D

O formato ico2D foi projetado exclusivamente para a implementação do ambiente VISE, sendo bastante simples e conciso.

Cada ícone é descrito por uma série de registros, sendo que em cada registro estão armazenadas as informações relativas a coordenada X e Y do ponto e a cor da aresta.

A seguir é apresentado o formato utilizado para ícones bidimensionais:

<numero de vertices>

<coord. X vertice 1> <coord. Y vertice 1> <cor da aresta>

<coord. X vertice n> <coord. Y vertice n> <cor da aresta>

BIBLIOGRAFIA

- [BEL 87] BELL, P.C. ; O'KEEFE, R.M. Visual interactive simulation-history, recent developments, and major issues. **Simulation**, San Diego, CA, v.49, n.3 , p.109-116, Sept.1988.
- [FRE 94] FREITAS, Carla Maria Dal Sasso . **Uma abordagem unificada para análise exploratória e simulação interativa visual**. Porto Alegre: CPGCC da UFRGS, ago. de 1994. 146p. Tese de Doutorado.
- [FRE 92] FREITAS, Carla Maria Dal Sasso . **Metodologia e Técnicas para Visualização de Dados Multivariados**. Porto Alegre: CPGCC da UFRGS, 1992. 47p. Proposta de Tese de Doutorado.
- [FRE 90] FREITAS, Carla Maria Dal Sasso . **Técnicas de Visualização em Simulação**. Porto Alegre: CPGCC da UFRGS, 1990. 47p. Trabalho Individual n.187.
- [GOR 79] GORDON, Geoffrey. **System Simulation** . Englewood Cliffs : Prentice-Hall, 1979. 303p.
- [HEL 90] HELLER, Dan . **XView Programming Manual**. Sebastopol : O'Reilly & Associates, 1990. 642p.
- [JOH 88] JOHNSON, M.E.; POORTE, J.P. A Hierarchical approach to computer animation in simulation modeling. **Simulation**, San Diego, CA, v.50, n.1, p.30-36, Jan.1988.
- [NYE 90] NYE, Adrian . **XLib Reference Manual** : for Version 11 of the X Window System. Sebastopol : O'Reilly & Associates, 1990. 635p.
- [SUN 90] SUN MICROSYSTEMS. **Open Look** : Graphical User Interface Functional Specification. Reading : Addison-Wesley, 1990. 564p.

- [SUN 90a] SUN MICROSYSTEMS. **Open Look** : Graphical User Interface Application Style Guidelines. Reading : Addison-Wesley, 1990. 388p.
- [CAC 91] CACI PRODUCTS COMPANY.**SIMGRAPHICS** : User's Guide and Casebook. La Jolla, CA: CACI Products Company, 1991. 124p.
- [CAC 83] CACI PRODUCTS COMPANY.**SIMSCRIPT II.5** : Programming Language. La Jolla, CA : CACI Products Company, 1983. 402p.
- [CAC 89] CACI PRODUCTS COMPANY.**SIMSCRIPT II.5** : Reference Handbook. La Jolla, CA : CACI Products Company, 1989. 330p.
- [CAC 89a] CACI PRODUCTS COMPANY.**UNIX SIMSCRIPT II.5** : User's Manual. La Jolla, CA : CACI Products Company, 1989. 88p.
- [WAG 92] WAGNER, Flávio Rech. **Notas de Aula da Disciplina "Modelagem e Simulação de Sistemas Discretos"**. Porto Alegre:CPGCC da UFRGS, 1992.

Informática



UFRGS

Ambiente de Simulação Visual Interativa.

por

Ernesto Lindstaedt

Dissertação apresentada aos Senhores:

Profa. Dra. Carla Maria Dal Sasso Freitas

Prof. Dr. Carlos Eduardo Pereira (DELET/UFRGS)

Prof. Dr. Léo Pini Magalhães (UNICAMP)

Vista e permitida a impressão.

Porto Alegre, 08 / 03 / 96.

Prof. Dr. Flávio Rech Wagner,
Orientador.

Prof. Flávio Rech Wagner
Coordenador do Curso de Pós-Graduação
em Ciência da Computação - CPGCC
Instituto de Informática - UFRGS