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

Um ambiente integrado de simulação de sistemas digitais

por

Paulo Rech Wagner

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

Prof. Flávio Rech Wagner
Orientador

Porto Alegre, abril de 1991.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Wagner, Paulo Rech

Um ambiente integrado de simulação de sistemas digitais / Paulo Rech Wagner.—Porto Alegre: CPGCC da UFRGS, 1991.

121 p.: il.

Dissertação (mestrado)—Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1991. Orientador: Wagner, Flávio Rech.

Dissertação: Sistemas Digitais, Simulação, Ambientes de Projeto

Dedico este trabalho a
todas as pessoas que ajudaram
na sua conclusão.

SUMÁRIO

LISTA DE ABREVIATURAS 8

LISTA DE FIGURAS 10

RESUMO 12

ABSTRACT 14

1 INTRODUÇÃO 16

1.1 Introdução ao sistema AMPLO 16

1.2 Introdução ao processo de simulação de sistemas discretos . . 19

1.3 Objetivo do trabalho 20

1.4 Apresentação da arquitetura do ambiente proposto 21

1.5 Apresentação do trabalho 23

2 ASPECTOS DE IMPLEMENTAÇÃO 25

2.1 Recursos utilizados 25

2.2 Organização das informações na tela 26

2.3 Mecanismos de navegação 29

3 COMPOSIÇÃO DO AMBIENTE DE SIMULAÇÃO 33

3.1 Objetos do ambiente 33

3.2 Módulos que compõem o ambiente 37

3.3 Interface principal do ambiente 38

4 MÓDULO DO CONSTRUTOR DE MODELOS 41

4.1 Introdução 41

4.2 Método interativo 44

4.3 Método automático 45

4.4 Inicialização do modelo 46

5 MÓDULO DE DESCRIÇÃO DE ESTÍMULOS 51

5.1 Introdução 51

5.2 Descrição Textual 52

5.2.1 Construções da linguagem 53

5.3 Descrição Gráfica 58

6 MÓDULO DE VINCULAÇÃO DE ESTÍMULOS 63

6.1 Introdução 63

6.2 Vinculação Textual 65

6.3 Vinculação Gráfica 68

7 MÓDULO DA LINGUAGEM DE COMANDOS 72

7.1 Introdução 72

7.2 Recursos de visualização 73

7.3 Comandos de simulação 75

7.3.1 Inject 75

7.3.2 Force 77

7.3.3 Short 78

7.3.4 Put 79

7.3.5 Read 80

7.3.6 Trace 81

7.3.7 Run 83

7.3.8 Break 84

7.3.9 Batch 86

7.3.10 Save 87

7.3.11 Back 88

7.3.12 Time 88

7.3.13 Exit 89

8 MÓDULO DE ANÁLISE DE RESULTADOS 90

8.1 Introdução 90

8.2 Análise da monitoração 92

8.3 Análise do histórico 93

9 BANCO DE DADOS 94

9.1 Interface orientada a objetos 94

9.2 Primitivas do ambiente de simulação 95

9.3 Restrições de integridade do ambiente de simulação 98

10 INTERAÇÃO COM OS SIMULADORES 100

11 CONCLUSÕES 103

ANEXO A-1 SINTAXE DA LINGUAGEM DE DESCRIÇÃO TEMPORAL DE ESTÍMULOS 106

ANEXO A-2 DESCRIÇÃO DAS PRIMITIVAS DA BASE DE DADOS 111

BIBLIOGRAFIA 118

LISTA DE ABREVIATURAS

AG	Árvore Geradora
BAT	Batch
CD	Comandos
EV	Estímulo
EVMS	Vinculação (EV vinculado a um MS)
LDTE	Linguagem de Descrição Temporal de Estímulos
LGDE	Linguagem Gráfica de Descrição de Estímulos
LOG	Histórico
MAR	Módulo de Análise de Resultados
MCM	Módulo do Construtor de Modelos
MDE	Módulo de Descrição de Estímulos
MLC	Módulo da Linguagem de Comandos
MS	Modelo de Simulação
MVE	Módulo de Vinculação de Estímulos
RA	Rede de Agências
RE	Resultados

SS	Sessão de Simulação
ST_{inic}	Estado Inicial
ST_{int}	Estado intermediário
TS	Tabela de Símbolos

LISTA DE FIGURAS

Figura 1.1 Arquitetura do sistema AMPLO. 16

Figura 1.2 Objetos do banco de dados. 18

Figura 1.3 Tarefas para a condução do processo de simulação do sistema AMPLO. 21

Figura 2.1 Áreas fixas do ambiente de simulação. 26

Figura 2.2 Áreas temporárias do ambiente de simulação. 27

Figura 2.3 Representação gráfica da agência REG. 29

Figura 2.4 Representação de retângulos da agência REG. 30

Figura 3.1 Relações entre os objetos do ambiente de simulação. 34

Figura 3.2 Módulos que compõem o ambiente de simulação. 37

Figura 3.3 Interface principal do ambiente de simulação. 39

Figura 4.1 Integração do construtor de modelos. 41

Figura 4.2 Rede de versões compostas. 42

Figura 4.3 Rede de versões primitivas. 43

Figura 4.4 Processo de inicialização de um modelo de simulação. 49

Figura 5.1 Forma de onda para a descrição do estímulo BOOL. 55

Figura 5.2 Forma de onda para a descrição do estímulo SIG2. 57

Figura 5.3 Interface da descrição gráfica de estímulos. 59

Figura 5.4 Descrição gráfica de estímulos. 61

Figura 6.1 Interface para a vinculação textual. 66

Figura 6.2 Formato da tabela de vinculações. 67

Figura 6.3 Interface para a vinculação gráfica. 69

Figura 7.1 Interface da linguagem de comandos. 73

Figura 8.1 Interface da análise de resultados. 91

Figura 9.1 Estratégia de implementação da base de dados. 94

RESUMO

O trabalho apresenta os recursos oferecidos ao usuário do ambiente de projeto AMPLO para o controle e gerência do processo de simulação de sistemas digitais.

O ambiente de simulação proposto é constituído por diversas ferramentas baseadas em recursos gráficos- interativos. As ferramentas do ambiente permitem executar funções como : construir modelos de simulação a partir das descrições de sistemas armazenadas na base de dados, gerar estados iniciais para os modelos de simulação através de estratégias de inicialização pré-definidas, criar estímulos a serem aplicados aos modelos de simulação utilizando linguagens gráficas e textuais dedicadas, vincular estímulos às entradas primárias dos modelos de simulação, controlar a simulação através dos comandos de simulação disponíveis na sessão de simulação, analisar os resultados das simulações já realizadas através de recursos gráficos de visualização e criar uma seqüência de comandos que devem ser executados dentro de uma sessão de simulação.

O ambiente de simulação integra todos os dados gerados durante o processo de simulação em uma base de dados única. Para isto, os objetos manipulados pelas diversas ferramentas do ambiente e as relações existentes entre eles foram definidos de acordo com um modelo de dados uniforme que é a base para a implementação de uma base de dados íntegra e não redundante. A interface de acesso a esta base de dados é constituída por funções primitivas que realizam o acesso a cada um dos objetos. Estas primitivas de acesso à base de dados permitem a criação, alteração e remoção dos objetos mantendo a consistência geral dos mesmos, bem como vários tipos de consultas.

O processo de simulação propriamente dito é controlado por um conjunto de funções próprias para a simulação disponíveis na sessão de simulação. A

sessão de simulação apresenta uma linguagem de comandos que através de recursos de visualização gráfico-interativos permite ao usuário, entre outros recursos, alterar e monitorar valores de sinais do modelo de simulação e controlar o avanço do tempo de simulação. A sessão de simulação realiza a comunicação com os simuladores através de um sistema de troca de mensagens onde para cada comando fornecido durante a sessão de simulação, uma mensagem é acrescentada ao conjunto de mensagens enviadas ao simulador.

PALAVRAS CHAVE : ambientes de projeto de sistemas digitais, simulação de sistemas digitais, ambientes de simulação.

ABSTRACT

This work describes the facilities that are available to the user of the AMPLO design environment for controlling and managing the process of digital systems simulation.

The proposed simulation environment is composed by several tools that are of graphical-interactive nature. These tools support tasks like: building simulation models from system descriptions stored in the data base, generating initial states for the models according to various initialization strategies, creating stimuli to be applied to the models by using dedicated graphical and textual languages, associating stimuli to the primary inputs of the models, controlling the simulation run through a specialized command language, and analyzing results of already executed simulation runs.

The environment integrates all data that is generated during the simulation process in a unique data base. Therefore, objects that are manipulated by the several tools of the environment, as well as relationships between them, have been defined according to a uniform data model which is the basis for the implementation of a consistent and non-redundant data base. The access interface to this data base is composed by primitive functions that implement the access to the objects. These functions allow the creation, modification, and removal of objects, while maintaining their overall consistency, as well as several queries.

The process of simulation itself is controlled by a command language. These commands are available during the simulation session, which integrates the environment with the AMPLO simulators through a message system. The command language, through graphical-interactive visualization facilities, allow the user to modify and monitor signals values of the model and to control the simulation

time advancement. Each command issue adds a new message to a message queue to be sent to the simulator.

KEYWORDS : digital systems design environments, digital systems simulation, simulation environments.

1 INTRODUÇÃO

1.1 Introdução ao sistema AMPLO

O sistema AMPLO (AMbiente integrado para o Projeto LOgico de sistemas digitais) [WAG 88, FRE 88] é um ambiente integrado para o projeto de sistemas digitais complexos em desenvolvimento no CPGCC da UFRGS. Este ambiente possui ferramentas que permitem a descrição e validação de sistemas digitais em três níveis de projeto : sistema, transferência entre registradores (RT) e portas lógicas. Para cada um destes níveis existe uma linguagem de descrição de hardware, respectivamente, LAÇO [SIL 88], KAPA [WAG 87c] e NILO [WAG 87b].

A figura 1.1 apresenta a arquitetura global do sistema AMPLO.

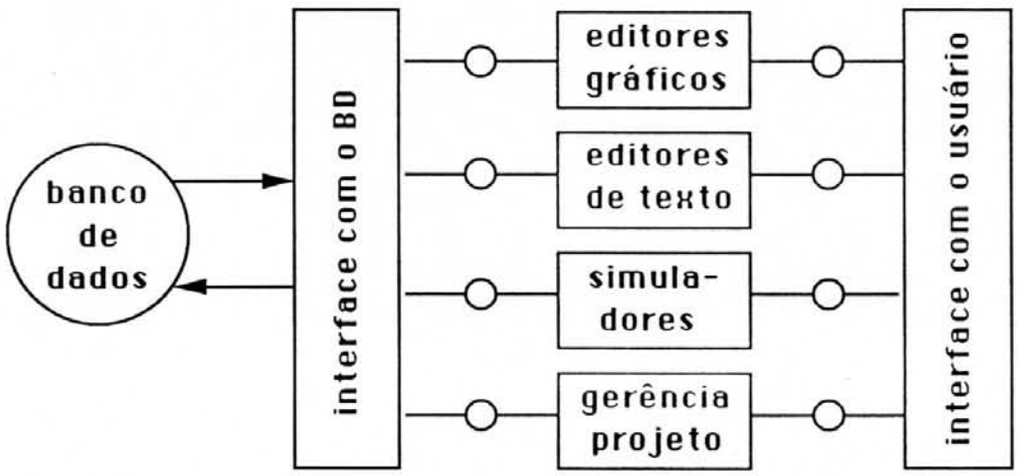

Figura 1.1: Arquitetura do sistema AMPLO.

Sistemas são descritos de forma modular e hierarquizada como redes de agências, através da linguagem REDES [WAG 87a]. Cada agência na rede pode ser descrita em uma de duas formas :

• como uma agência composta, constituída por uma rede de ocorrências de outras agências, utilizando-se da linguagem REDES;

• como uma agência primitiva, especificada através de uma de três linguagens de descrição de hardware : LAÇO, uma linguagem baseada em redes de Petri; KAPA, uma linguagem para descrições estruturais no nível de transferência entre registradores; e NILO, uma linguagem para descrições estruturais no nível de portas lógicas elementares e chaves bidirecionais.

As quatro linguagens de descrição suportadas pelo ambiente AMPLO possuem formas gráfica e textual que são completamente equivalentes entre si [WAG 86a]. Isto permite que a descrição de uma agência possa ser feita textualmente, com o auxílio de um editor de textos convencional e um compilador para a linguagem escolhida, ou graficamente, via um editor gráfico especializado para a linguagem.

As seguintes propriedades garantem uma efetiva integração do processo de projeto através de todos os níveis de descrição :

• integração de todos os dados de projeto numa base de dados unificada de forma íntegra e não redundante;

• todas as ferramentas de projeto são acessadas através de uma linguagem de comandos unificada de natureza gráfico-interativa;

• separação explícita entre descrições compostas representadas como redes de agências através da linguagem REDES, e descrições primitivas, representadas nas linguagens LAÇO, KAPA e NILO.

A estrutura dos sistemas digitais projetados em AMPLO é descrita como uma rede de agências. Uma agência pode apresentar várias alternativas e versões de projeto, conforme pode ser visto na figura 1.2. Alternativas de uma

mesma agência diferem entre si por possuírem diferentes definições para a interface, por exemplo, número diferente de sinais de entrada e saída, nomes, tipos de dados ou dimensão diferente para os sinais de entrada e saída ou, ainda, diferentes atributos geométricos da interface. Uma determinada alternativa de uma agência pode estar definida em vários níveis de descrição. Entretanto, os tipos de dados de um mesmo sinal da interface nas diversas definições devem ser compatíveis entre si. A cada alternativa, o projetista pode associar várias versões, mantendo, para isso, a mesma definição da interface e fornecendo diferentes descrições internas, sejam elas compostas ou primitivas.

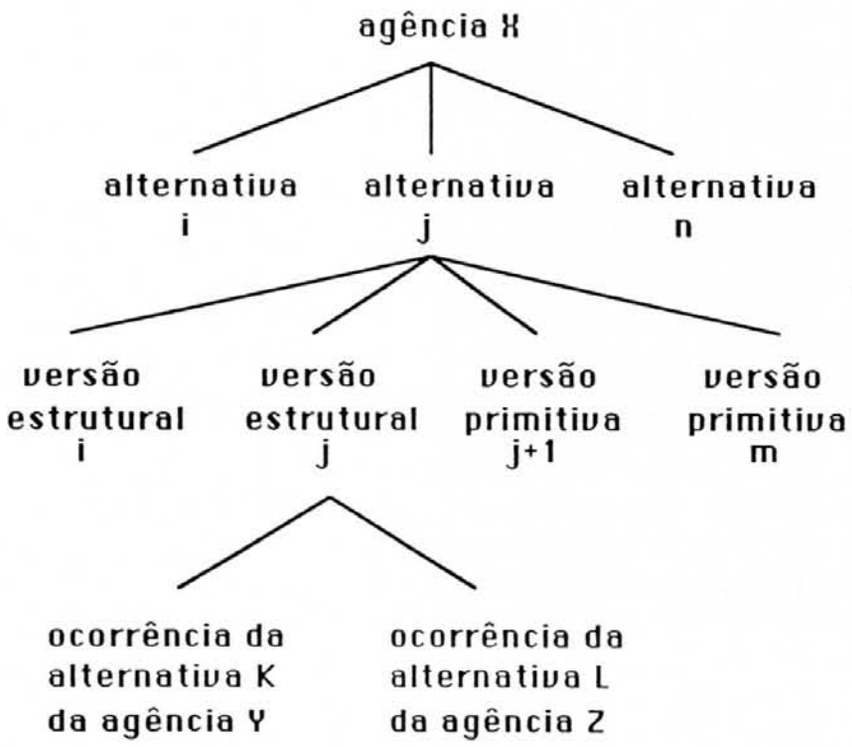

Figura 1.2: Objetos do banco de dados.

Pode-se descrever primitivamente uma agência em um dos três níveis mencionados. A especificação de uma agência é hierárquica. Deste modo, o projetista pode especificar a agência de forma modular até que alcance um nível de

refinamento no qual apenas lhe interessa descrever as agências numa das linguagens LAÇO, KAPA ou NILO.

A especificação de uma alternativa de agência cria, de fato, um tipo de agência na base de dados. Ocorrências destes tipos podem ser posteriormente usadas em versões compostas de outras agências. Ocorrências de um mesmo tipo têm em comum a mesma descrição para a interface.

Na construção de uma agência composta podem ser utilizadas ocorrências de alternativas (representam configurações dinâmicas) [KAT 87] ou ocorrências de versões (representam configurações estáticas). Uma agência composta por ocorrências de versões está pronta para ser simulada. Já uma agência composta por ocorrências de alternativas necessita a escolha das versões, para a simulação.

1.2 Introdução ao processo de simulação de sistemas discretos

Para conduzir um determinado processo de simulação [SHA 75], devem ser realizadas diversas tarefas que podem ser resumidas da seguinte forma :

- preparar o modelo de simulação, especificando os dados necessários ao modelo e definindo quais são os elementos que o compõem;

- preparar o cenário da simulação, isto é, escolher as condições iniciais necessárias para que o processo de simulação possa ser iniciado. Estas condições incluem a definição dos estímulos (sequência de eventos externos) que serão aplicados às variáveis de entrada primária do modelo de simulação e a indicação do estado inicial (valores) dos sinais existentes no modelo de simulação;

• realizar o experimento propriamente dito, através do controle dos comandos fornecidos interativamente ou provenientes de arquivos previamente preparados;

• analisar os resultados provenientes de um processo de simulação através dos recursos fornecidos ao usuário.

1.3 Objetivo do trabalho

O objetivo do trabalho é a especificação, projeto e implementação de um ambiente integrado de simulação para o sistema AMPLO. Este ambiente foi especificado de modo a oferecer ao usuário uma série de recursos que visam aumentar a produtividade do processo de simulação. O ambiente de simulação, assim como as diversas ferramentas do sistema AMPLO (linguagens, compiladores, editores gráficos), deverá possuir uma interface homogênea e essencialmente gráfica e, realizar a integração de todos os dados em uma base de dados única.

O ambiente de simulação proposto é composto ao menos pelas seguintes ferramentas essenciais :

• um construtor de modelos para obter as estruturas de dados necessárias ao processo de simulação;

• uma linguagem de descrição de estímulos a serem aplicados aos modelos durante a simulação;

• uma linguagem de comandos que ofereça facilidades para controlar e visualizar o processo de simulação.

Estas ferramentas possuem interfaces próprias, que têm recursos de visualização e auxílio ao usuário, de modo a facilitar a sua interação. Além destas

ferramentas indispensáveis ao ambiente de simulação, também foram acrescidas outras que auxiliam no processo de simulação, tais como : uma forma de analisar as simulações já realizadas, um procedimento para vincular os estímulos aos modelos de simulação, mecanismos para visualizar os objetos gerados e manipulados pelo ambiente, etc.

Foi necessário definir os objetos manipulados por estas ferramentas e as relações existentes entre eles, de modo a integrar estas ferramentas com a base de dados já existente. Deve ser levado em conta, nesta definição, que alguns destes objetos são complexos, contendo estrutura e hierarquia. A interface de acesso à base de dados foi estendida pela inclusão de primitivas de acesso a estes objetos, que devem garantir a consistência dos dados.

1.4 Apresentação da arquitetura do ambiente proposto

Nesta seção é apresentada, em linhas gerais, a proposta do processo de simulação a ser suportado pelo ambiente, objeto deste estudo. Na condução do processo de simulação, o usuário deve realizar as seguintes tarefas (como apresentado na figura 1.3) : preparar o modelo de simulação, preparar o cenário de simulação, controlar a sessão de simulação e analisar os resultados da simulação.

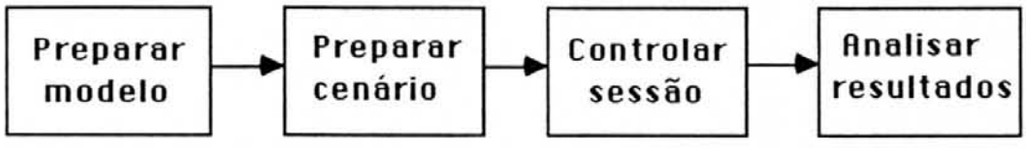

Figura 1.3: Tarefas para a condução do processo de simulação do sistema AMPLO.

Na preparação do modelo de simulação, o usuário deve criar o modelo de simulação a partir das descrições de agências contidas na base de dados e inicializar

o modelo de simulação. A construção do modelo de simulação exige a execução de dois processos : configurar a descrição e resolver a hierarquia (expansão em uma rede de versões primitivas). Para que a sessão de simulação possa ser iniciada, o usuário deve preparar um cenário de simulação para o modelo de simulação escolhido. Este cenário é composto por dois objetos armazenados na base de dados : estímulos e estado inicial. Os estímulos representam o comportamento temporal dos sinais de entrada do modelo de simulação, podendo ser especificados tanto de forma textual como gráfica, através de duas linguagens especializadas e orientadas às linguagens de descrição de hardware do ambiente. Nestas linguagens existem construções que permitem a descrição de sequências periódicas e não periódicas de valores ao longo do tempo. O cenário é completado com a seleção de um estado inicial para o modelo de simulação. Este estado inicial contém os valores iniciais dos sinais existentes no modelo de simulação. Estes objetos (estímulos e estado inicial) devem estar associados ao modelo de simulação escolhido.

Após a definição do cenário de simulação (escolha de um modelo de simulação previamente construído, dos estímulos de entrada e de um estado inicial para o modelo de simulação), o usuário tem à sua disposição uma linguagem de comandos para controlar a sessão de simulação propriamente dita. Essa sessão pode ser controlada interativamente ou através de objetos, que contêm comandos de simulação, previamente armazenados na base de dados. Pode-se salvar estados intermediários da simulação para serem utilizados novamente na mesma sessão ou serem utilizados como estado inicial em outras sessões, desde que estejam relacionadas com o mesmo modelo de simulação.

Finalmente, após encerrada uma sessão de simulação, pode-se realizar uma análise posterior do funcionamento do modelo de simulação durante a sessão. Esta análise é baseada em objetos criados durante a sessão, contendo o comportamento temporal de sinais escolhidos pelo usuário. Durante a sessão também é criado o objeto "LOG" que contém todos os comandos fornecidos pelo usuário ao longo da sessão. Este "LOG" também pode ser acessado na fase de análise de resultados.

1.5 Apresentação do trabalho

Após esta introdução geral versando sobre o sistema AMPLO e o ambiente de simulação proposto, os demais capítulos deste trabalho detalham a proposta e sua implementação tratando dos assuntos descritos a seguir.

No capítulo 2, são apresentados os recursos utilizados na implementação do ambiente de simulação, as definições de cada uma das áreas que compõem as interfaces do ambiente e, os mecanismos utilizados para navegar sobre a hierarquia dos objetos armazenados na base de dados.

O capítulo 3 define os objetos manipulados pelo ambiente assim como as relações existentes entre eles. Os objetivos dos módulos que compõem o ambiente e a interface principal do ambiente com suas opções e menus "pull-down" também são tratados.

O capítulo 4 apresenta o módulo do construtor de modelos. Este capítulo mostra como construir um modelo de simulação a partir de descrições armazenadas na base de dados. Também são descritos os dois métodos de construção do modelo de simulação, assim como as estratégias de inicialização para gerar um estado inicial para este modelo.

O capítulo 5 apresenta o módulo de descrição de estímulos, que permite a geração dos estímulos a serem aplicados aos modelos durante a simulação. Neste capítulo são introduzidas as duas linguagens de descrição de estímulos (textual e gráfica) com suas principais características e formas de interação com o usuário.

O capítulo 6 apresenta o módulo de vinculação de estímulos onde é realizada a ligação entre os estímulos e as entradas primárias de um modelo de simulação. As duas formas (textual e gráfica) de se realizar uma vinculação, com suas características e interfaces com usuário, também são apresentadas neste capítulo.

O capítulo 7 apresenta o módulo da linguagem de comandos que fornece os recursos para o controle do processo de simulação. São apresentados os recursos de visualização disponíveis e uma descrição detalhada de todos os comandos de simulação existentes na interface da linguagem de comandos, incluindo exemplos e sintaxe.

O capítulo 8 apresenta o módulo de análise de resultados. Neste capítulo são descritos os recursos de visualização disponíveis ao usuário para analisar os resultados de simulações já realizadas. Esta análise é realizada através da exibição das formas de onda dos sinais monitorados na simulação e, através do histórico dos comandos realizados durante todo o processo de simulação.

O capítulo 9 apresenta a integração do ambiente de simulação com a base de dados. Também são apresentadas as primitivas de acesso à base de dados referentes ao ambiente, assim como as restrições de integridade relativas à composição dos objetos.

O capítulo 10 apresenta a integração do ambiente de simulação com os simuladores do sistema AMPLO. Também é apresentado neste capítulo como é realizado o sistema de troca de mensagens entre ambiente e simuladores.

O capítulo 11 conclui este trabalho fazendo considerações sobre as principais características do ambiente, o tamanho de memória utilizada para armazenar os objetos na base de dados e o atual estágio de implementação do ambiente.

2 ASPECTOS DE IMPLEMENTAÇÃO

2.1 Recursos utilizados

O ambiente de simulação, assim como as demais ferramentas do projeto AMPLO, está sendo desenvolvido na linguagem C e ambiente PC visando a portabilidade para um ambiente UNIX.

As interfaces gráficas do ambiente (principal e a da linguagem de comandos) estão sendo desenvolvidas com o Pacote de controle da Interface com o Usuário (PIU), de modo que possa oferecer todas as facilidades para o usuário [GRU 89]. O PIU [OLA 89] é uma ferramenta que auxilia no projeto de interfaces amigáveis com o usuário [ROS 88], permitindo a definição do formato de telas e a organização de diálogos baseados em cardápios, sem que haja a necessidade de alterar o código de programação. Atualmente o PIU está sendo expandido para permitir novos recursos gráficos-interativos definidos no ambiente de simulação [FIG 91]. A interface com o usuário é descrita numa linguagem (LINUS [OLA 89]), compilada, e a interação é controlada via chamada das subrotinas do PIU.

Todos os outros recursos gráficos do ambiente de simulação são implementados com o Pacote Gráfico (PG), também totalmente desenvolvido no projeto AMPLO. O PG [PIN 88] é um conjunto de funções que auxiliam o projetista na manipulação de entes gráficos. Todas as operações gráficas e visualizações realizadas na área de trabalho do ambiente são executadas através do PG.

Ambos os ambientes gráficos (PIU e PG) também foram desenvolvidos na linguagem C e estão fundamentados em recursos gráficos [FOL 82].

2.2 Organização das informações na tela

A interface do ambiente de simulação apresenta-se dividida funcionalmente nas seguintes áreas de visualização: área de trabalho, área de menu e orientação, área de lembretes, área de mensagens de erro, área de eco de entrada, área de menus horizontais e área de help.

As três áreas apresentadas na figura 2.1 são áreas fixas do ambiente de simulação, isto é, estão sempre presentes durante a execução de qualquer operação do ambiente.

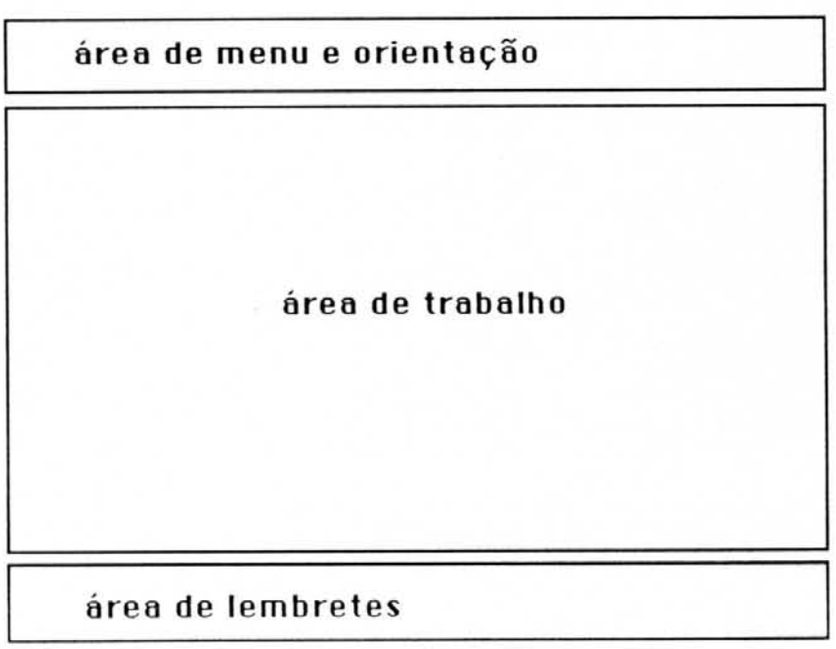

Figura 2.1: Áreas fixas do ambiente de simulação.

As áreas temporárias, como apresentado na figura 2.2, são áreas que permanecem ativas até o momento em que o usuário desejar. Estas áreas são sempre exibidas sobre a área de trabalho.

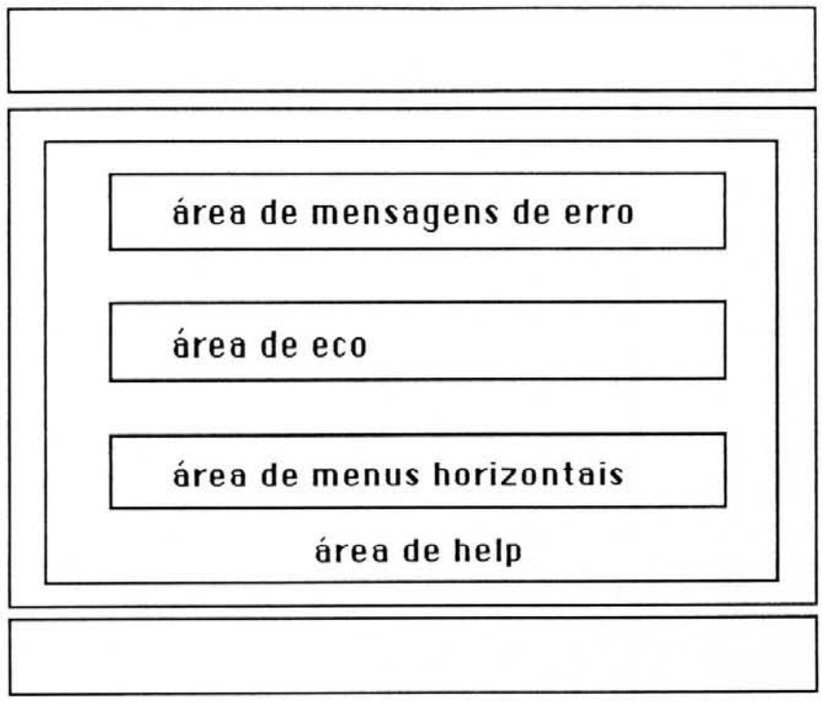

Figura 2.2: Áreas temporárias do ambiente de simulação.

A área de menu e orientação exibe informações diferenciadas de acordo com o estado em que se encontra o ambiente de simulação. Quando se está realizando operações sobre a área de trabalho, são exibidas nesta área informações a respeito dos objetos que estão associados com a operação que está se processando, como : nome do modelo de simulação, nome da sessão de simulação, nome do estímulo, nome da agência, alternativa ou versão.

Quando o usuário se encontra nas interfaces de interação do ambiente, interface principal e da linguagem de comandos, a área de menu e orientação apresenta os respectivos menus das interfaces. Estes menus são do tipo "pull-down" onde suas opções podem ser selecionadas de diversas maneiras : através da letra em destaque em cada nome da opção; através de teclas especiais específicas para determinadas opções; utilizando o teclado (setas de movimentação) para movimentar a barra em reverso que se sobrepõe às opções ou ainda através do "mouse" se este estiver presente. Algumas opções dos menus quando selecionadas exibem novos menus "pull-down" que são manipulados da mesma maneira. Deste modo, tem-se uma

hierarquia de menus onde através de uma tecla especial pode- se voltar ao menu de um nível mais acima.

A área de lembretes serve para orientar o usuário sobre o que ele deve fazer para realizar a operação que está sendo processada no momento.

A área de trabalho é utilizada para a exibição de todos os recursos gráficos e textuais necessários à realização das operações contidas no ambiente. É a área onde o usuário opera com os objetos armazenados na base de dados.

A área de mensagens de erros serve para o ambiente avisar ao usuário que ele cometeu algum erro. Quando um erro é cometido, o ambiente apresenta nesta área uma mensagem explicando qual foi o erro. Após tomar conhecimento do erro cometido, o usuário deve ativar uma tecla especial para apagar a mensagem e retornar ao estado em que se encontrava.

A área de entrada de eco é utilizada em caso de entrada de informações via teclado. Esta área permanece ativa enquanto o usuário não efetivar a informação ou cancelar a entrada de eco.

A área de menus horizontais permite a seleção de uma operação ou de uma opção para realizar determinada operação. A barra em reverso desloca-se horizontalmente a fim de realizar a seleção. Neste tipo de menu não existe a seleção por intermédio de uma letra ou uma tecla especial como acontece nas interfaces de interação (interface principal e da linguagem de comandos). Esta área permanece ativa enquanto o usuário não efetivar a sua escolha ou cancelar o menu horizontal.

A área de help é utilizada para prestar informações ao usuário a respeito dos objetos armazenados na base de dados através de consultas a mesma e sobre a função de uma determinada operação. As informações apresentadas são relativas ao contexto em que o usuário se encontra no ambiente de simulação [CAR 88].

2.3 Mecanismos de navegação

Ao se realizar uma operação onde é necessária a função de consulta e seleção a qualquer nível da base de dados, o "browser" é ativado. O "browser" é uma ferramenta que permite que se navegue sobre a hierarquia dos objetos armazenados na base de dados.

A navegação pode ser realizada através de dois mecanismos de visualização dos objetos armazenados na base de dados. No primeiro é utilizada a representação gráfica dos objetos (realizado na linguagem de comandos pela função de apontamento) como apresentado na figura 2.3. Neste tipo de representação somente são visualizados dois níveis da árvore hieráquica de cada vez. No segundo mecanismo, os objetos armazenados na base de dados são representados através de retângulos (utilizado em consultas, na construção do modelo de simulação e na seleção de objetos). Este tipo de representação é apresentado na figura 2.4.

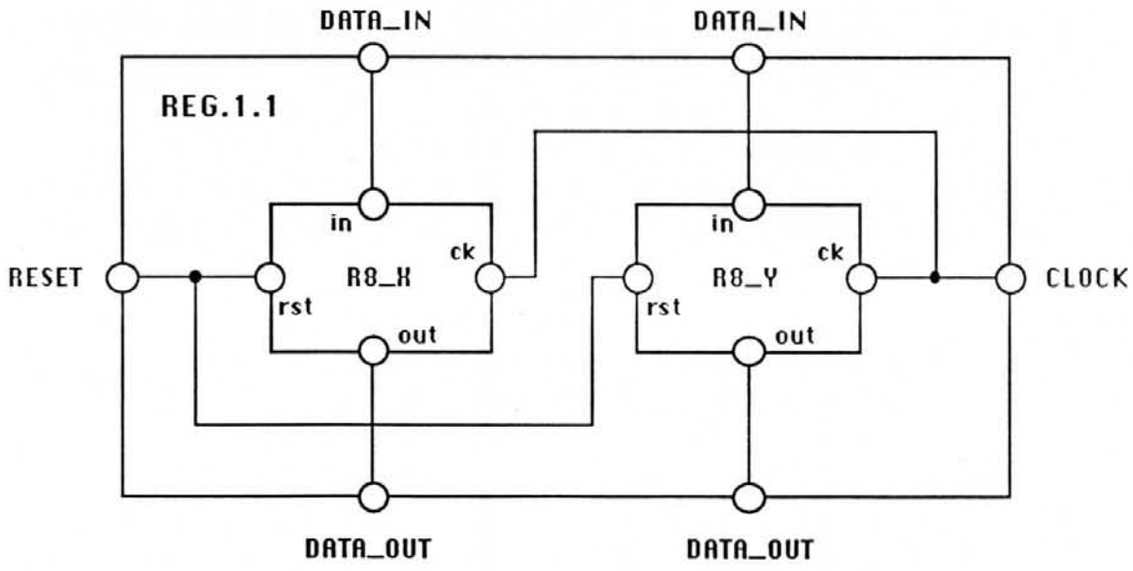

Figura 2.3: Representação gráfica da agência REG.

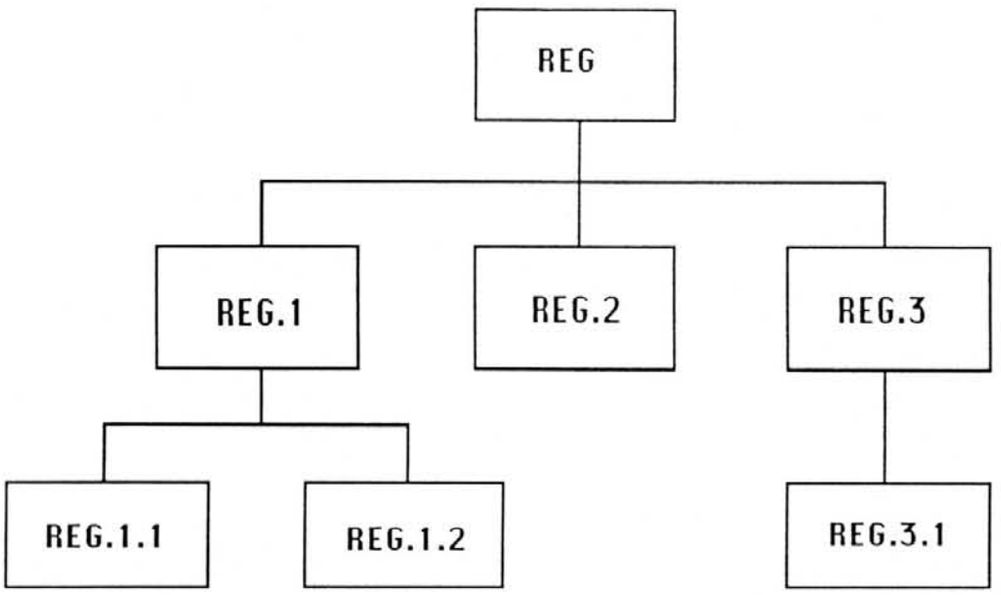

Figura 2.4: Representação de retângulos da agência REG.

Na representação gráfica dos objetos, existe a possibilidade de se realizar um "scroll", uma vez que esta representação pode ser muito grande de modo a não ser totalmente visível na área de trabalho. Na representação de retângulos, como o número de objetos armazenados na base de dados pode ser muito grande e o número de relações entre eles também, existem opções de seleção ("prunning" e "poda") [KAT 87] na hierarquia dos objetos de modo a facilitar a navegação e a visualização.

A navegação sobre os objetos é realizada na área de trabalho sendo que para cada objeto apontado durante a navegação um dos dois procedimentos seguintes pode ser executado :

• apresentar uma nova exibição da árvore hierárquica de acordo com critérios pré-estabelecidos. Deste modo, são exibidos os objetos que estão relacionados com o objeto apontado. Isto representa uma mudança no nível da árvore hierárquica dos objetos (em direção às folhas). Durante a navegação sobre a árvore hierárquica existe a possibilidade de se voltar um nível na hierarquia (em direção à raiz). Isto permite que se mude o caminho a ser percorrido na árvore hierárquica;

• selecionar este objeto apontado para executar uma operação que se encontra em andamento. Por exemplo, quando for solicitado um estado inicial para um modelo de simulação, o usuário pode informar textualmente toda a hierarquia na qual este estado inicial está relacionado ou então, utilizar o "browser" para selecioná-lo. Neste segundo caso, o estado inicial será uma folha da árvore hierárquica.

O mecanismo de navegação do "browser" pode ser realizado sobre os objetos gerais do AMPLO (agências, alternativas e versões) ou sobre os objetos específicos do ambiente de simulação (modelos de simulação, estímulos, estados, sessões de simulação, vinculações e "batchs"). Para ambos conjuntos de objetos armazenados na base de dados, a navegação pode ser realizada através da representação de retângulos. Já a representação gráfica é utilizada somente sobre os objetos gerais do AMPLO. Ao se iniciar a navegação sobre os objetos gerais do AMPLO, utilizando-se da representação de retângulos, são exibidas todas as agências armazenadas na base de dados. Através de mecanismos interativos simples, o usuário tem a possibilidade de realizar diversas consultas à base de dados [LUZ 91].

Quando a navegação é realizada sobre os objetos do ambiente de simulação, todos os modelos de simulação relacionados com a agência.alternativa.versão indicada são exibidos sob a forma de retângulos. A partir deste ponto, o usuário tem a possibilidade de realizar as seguintes consultas :

• indicando um modelo de simulação, obter os objetos que estão relacionados com ele (estados inicial e intermediário , vinculações e sessões de simulação);

• indicando um estado inicial, obter quais as sessões de simulação em que ele é utilizado;

• indicando um estado intermediário, obter qual a sessão de simulação que o gerou e quais as sessões de simulação em que ele é utilizado como um estado inicial;

- indicando uma vinculação, obter a exibição de sua tabela de vinculações, todos os estímulos que estão relacionados com ela e em quais as sessões de simulação que ela é utilizada;

- indicando uma sessão de simulação, obter os objetos contidos nela (resultado, log, estado inicial, estados intermediários e vinculação);

- indicando a agência ou a alternativa da agência na qual o modelo de simulação está relacionado, obter os estímulos que estão relacionados com ela, assim como os estímulos que não possuem relacionamento com nenhum objeto;

- indicando um estímulo, obter a sua descrição e quais as vinculações que o utilizam.

3 COMPOSIÇÃO DO AMBIENTE DE SIMULAÇÃO

3.1 Objetos do ambiente

AMPLO suporta a gerência do processo de simulação armazenando modelos de simulação, estímulos, vinculações, estados do sistema, sessões de simulação, "batchs" e resultados como objetos da base de dados. A figura 3.1 apresenta as relações existentes entre os objetos do ambiente de simulação. Estas relações são descritas a seguir.

Os objetos do ambiente de simulação guardam entre si determinadas relações que garantem a consistência entre eles. O ambiente de simulação é inteiramente consistente com os demais objetos de projeto armazenados na base de dados (agências, alternativas e versões) e com os demais recursos de gerência de LAGO [LUZ 90]. Os mecanismos de consulta de LAGO estão permanentemente à disposição durante todo o processo de simulação.

O objeto MS (Modelo de Simulação) é uma rede de versões primitivas de agências construído a partir das descrições de agências contidas na base de dados. Deste modo, o objeto MS deve estar relacionado com a agência.alternativa.versão raiz que lhe deu origem. Cada objeto MS é composto por três sub-objetos : RA (rede de agências), AG (árvore geradora) e TS (tabela de símbolos). Todos os objetos (agências, alternativas e versões) utilizados na construção do objeto MS não podem ser removidos ou alterados na base de dados enquanto existir algum objeto MS relacionado com eles.

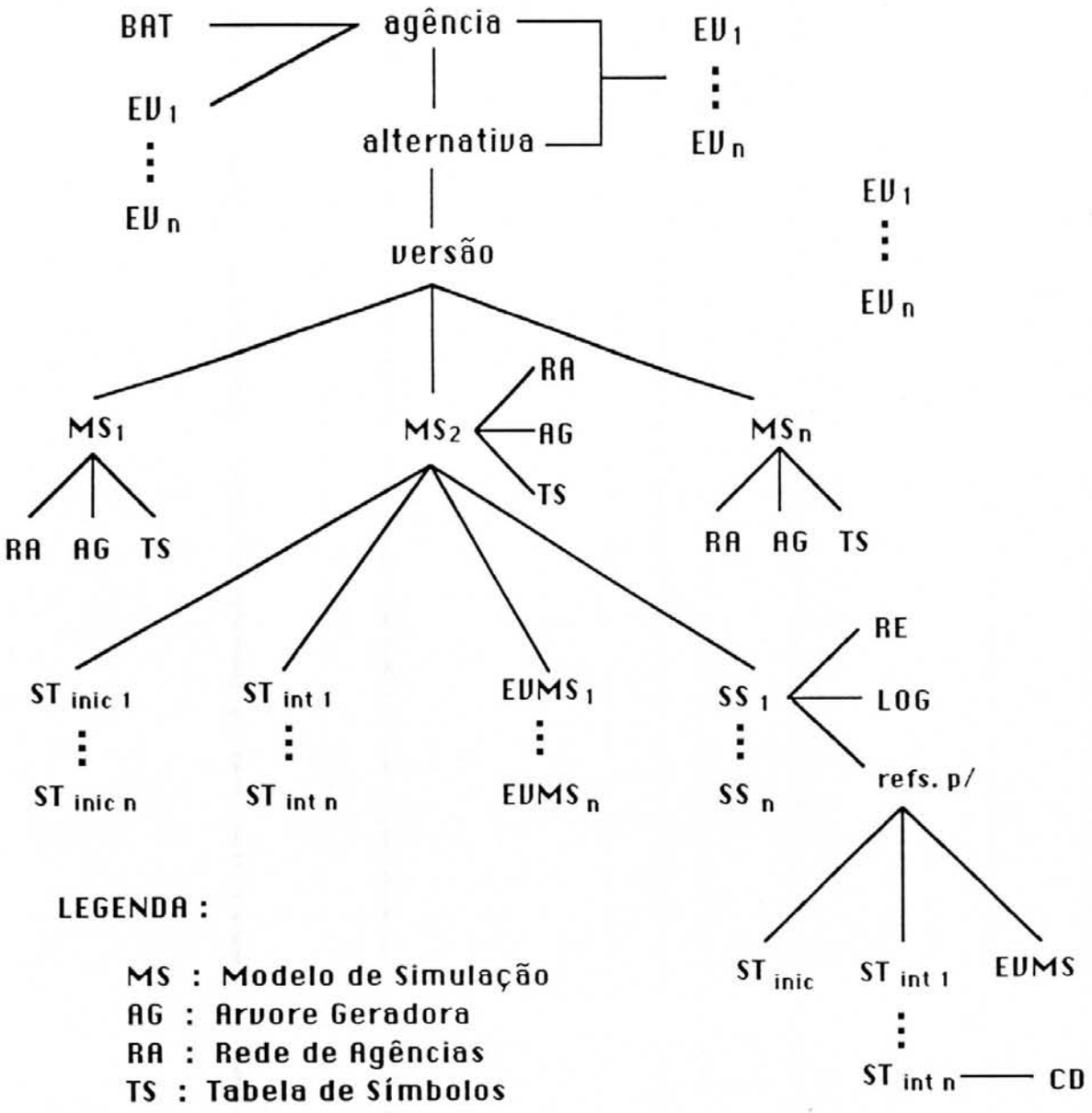

Figura 3.1: Relações entre os objetos do ambiente de simulação.

Os objetos EV (estímulos) representam eventos externos para os sinais de entrada dos modelos de simulação. Um objeto EV pode estar relacionado com os demais objetos da base de dados de uma de três maneiras :

• sem relacionamento, pode-se pensar como se fosse uma biblioteca de objetos EV que podem ser relacionados com qualquer modelo de simulação existente na base de dados;

• relacionados com a agência. Uma vez que as diferentes alternativas de uma agência não possuem uma diferença tão significativa na sua interface de modo que possa representar em estímulos muito diferentes entre si;

• relacionados com a agência.alternativa. Representa um conjunto de objetos EV que podem ser relacionados com um número mais reduzido de objetos MS (aqueles que estão relacionados com a mesma agência.alternativa).

Uma consistência deve ser realizada para verificar que na operação de remoção de agências e alternativas da base de dados, as mesmas não estão relacionadas com algum objeto EV existente.

Para cada objeto MS, podem estar relacionados um ou vários objetos ST_{inic} (estado inicial) assim como um ou vários objetos ST_{int} (estado intermediário). Estes objetos contêm o valor de todos os sinais do modelo de simulação. O objeto ST_{inic} é utilizado com estado inicial de uma sessão de simulação relacionada com o mesmo objeto MS. Já o objeto ST_{int}, criado em uma sessão de simulação, pode ser utilizado como estado inicial ou como estado intermediário de uma sessão de simulação. Ainda relacionado com o objeto MS existem : o objeto EVMS (vinculação) e o objeto SS (sessão de simulação).

O objeto EVMS contém a associação de descrições de estímulos com os sinais de entrada de um modelo de simulação. Deste modo, o objeto EVMS representa o relacionamento de um objeto MS com um ou vários objetos EV. Um

objeto EV não pode ser removido ou alterado na base de dados se ele for utilizado por um objeto EVMS. Os objetos ST_{inic}, ST_{int}, e EVMS só podem ser removidos da base de dados se não estiverem sendo utilizados em uma sessão de simulação. Quando um objeto MS é removido da base de dados, todos os objetos (ST_{inic}, ST_{int}, EVMS e SS) que estão relacionados com ele também são removidos.

Cada objeto SS possui relacionado um objeto RE (resultados) que contém o comportamento temporal dos sinais monitorados durante a sessão de simulação, um objeto LOG (histórico) que contém todos os comandos fornecidos pelo usuário ao longo da sessão de simulação, o nome do objeto ST utilizado como estado inicial da sessão de simulação, o nome do objeto EVMS e o nome dos objetos ST_{int} com o seu respectivo objeto CD (comandos) que contém os comandos ativos da sessão de simulação no momento em que o objeto ST_{int} foi criado. O objeto SS pode ser removido da base de dados não-refletindo nos demais relacionamentos dos objetos do ambiente. Os objetos contidos no objeto SS (RE , LOG e CD) também devem ser removidos.

Finalmente, relacionado com a agência existe o objeto BAT (Batch). Este objeto contém uma lista de comandos de simulação (apresentados no capítulo 7) que devem ser executados seqüencialmente dentro de uma sessão de simulação. Esta sessão está relacionada a um modelo de simulação que deve estar relacionado com a mesma agência que o objeto BAT está relacionado. Os objetos BAT podem ser removidos e alterados sem que haja qualquer reflexo no relacionamento dos objetos do ambiente.

3.2 Módulos que compõem o ambiente

O ambiente de simulação é composto por 5 (cinco) módulos principais. Cada módulo é responsável por determinadas funções, como será visto nos demais capítulos. A figura 3.2 apresenta os módulos principais do ambiente de simulação.

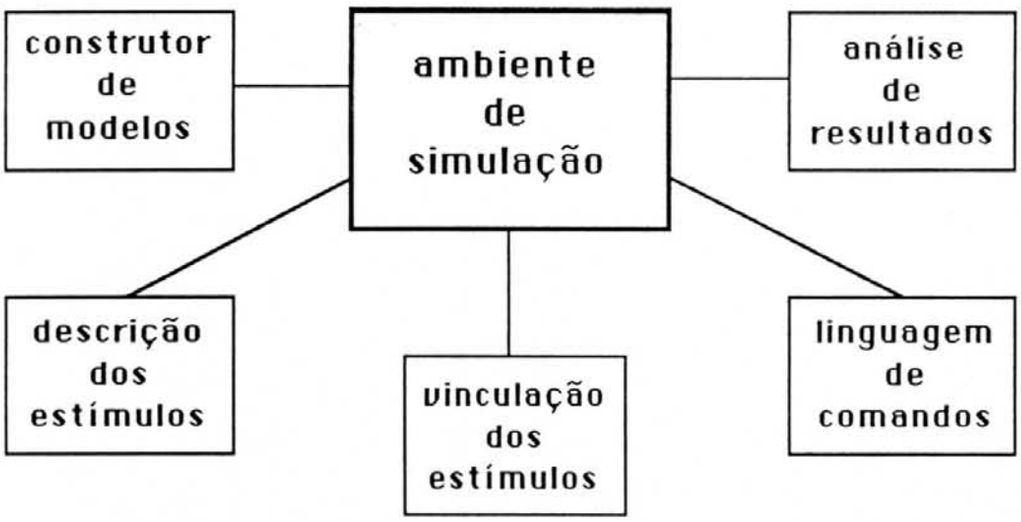

Figura 3.2: Módulos que compõem o ambiente de simulação.

O objetivo principal de cada módulo é o seguinte :

MCM- *Módulo do Construtor de Modelos* : obter a estrutura de dados necessária ao processo de simulação a partir das descrições de sistemas digitais armazenadas na base de dados;

MDE - *Módulo de Descrição de Estímulos* : gerar os estímulos que serão aplicados aos sinais dos modelos de simulação;

MVE - *Módulo de Vinculação de Estímulos* : vincular os estímulos gerados com as entradas primárias dos modelos de simulação;

MLC - *Módulo da Linguagem de Comandos* : fornecer comandos de modo a facilitar a visualização e o controle do processo de simulação;

MAR - *Módulo de Análise de Resultados* : analisar os resultados de simulações já realizadas através da visualização de sinais monitorados e do histórico de comandos.

3.3 Interface principal do ambiente

Ao ingressar no ambiente de simulação, o usuário encontra à sua disposição a interface principal. A figura 3.3 apresenta a estrutura completa da interface principal do ambiente de simulação com suas opções e seus sucessivos menus "pull-down".

A opção **MODELO** permite ao usuário construir um modelo de simulação, conforme o tipo de descrição (textual ou gráfica) e método (interativo ou automático) que ele desejar, armazenando-o na base de dados. Nesta opção também é possível inicializar um modelo de simulação (criar um estado inicial na base de dados) de acordo com uma das estratégias disponíveis e ainda, remover um modelo de simulação e um estado inicial armazenados na base de dados.

A opção **ESTÍMULO** permite que o usuário descreva textualmente ou graficamente os estímulos que serão aplicados às entradas primárias dos modelos de simulação. Para a descrição textual existe a opção de compilação, de modo que ambas as descrições (textual e gráfica) gerem a mesma estrutura de dados aos simuladores. Também pode ser realizado, nesta opção, a vinculação destes estímulos com as entradas primárias dos modelos de simulação armazenando-a na base de dados.

A opção **RESULTADOS** permite que o usuário ative o módulo de análise de resultados com a finalidade de analisar os resultados de simulações já realizadas.

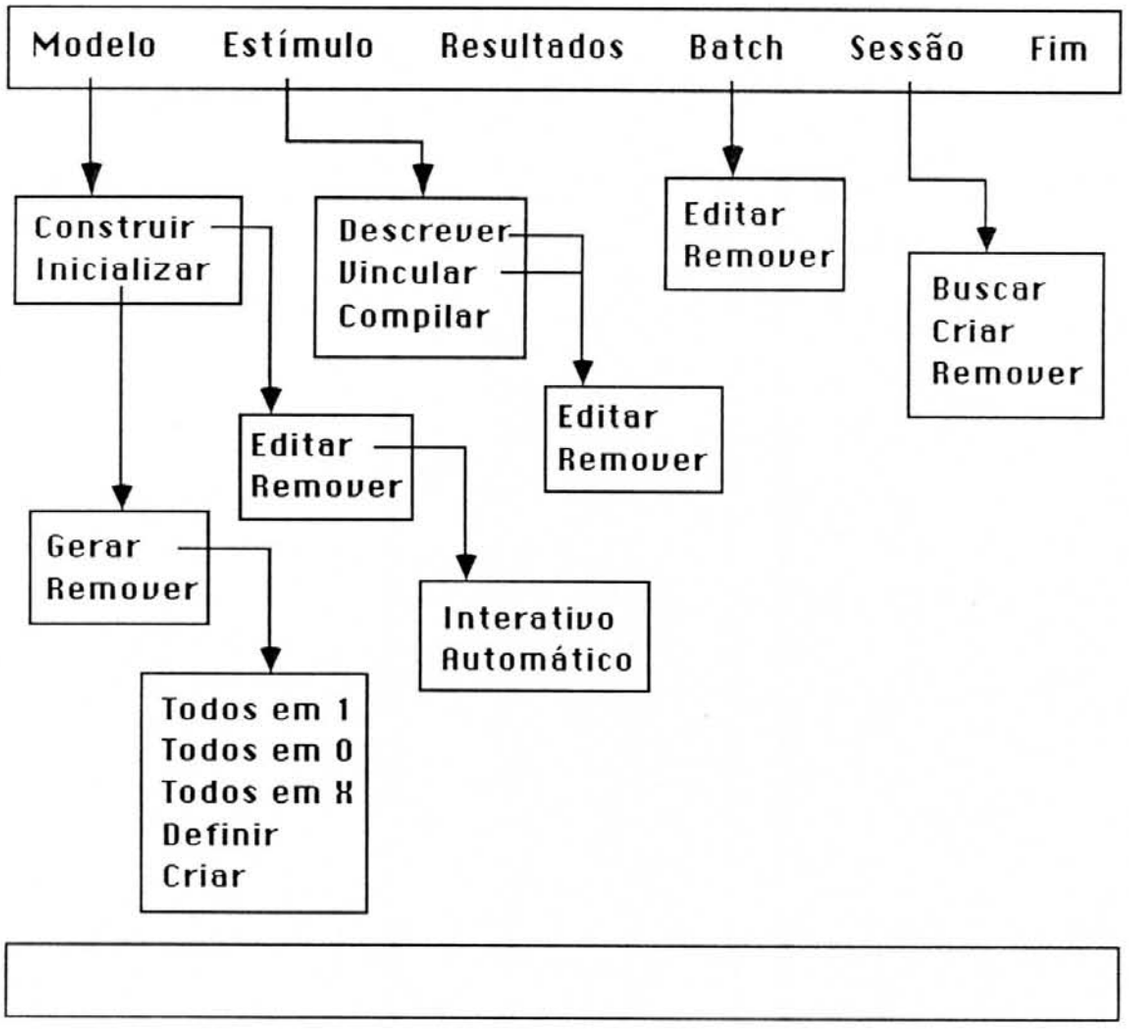

Figura 3.3: Interface principal do ambiente de simulação.

A opção **BATCH** permite ao usuário, através de um editor de textos próprio do AMPLO, armazenar na base de dados uma seqüência de comandos que devem ser executados seqüencialmente dentro de uma sessão de simulação. Os comandos fornecidos devem pertencer ao conjunto de comandos existentes para o ambiente de simulação. É possível remover da base de dados esta seqüência de comandos que foi armazenada sob a forma de um arquivo (objeto).

A opção **SESSÃO** permite ao usuário criar, buscar ou remover uma sessão de simulação armazenada na base de dados. Na sessão de simulação estão contidos os comandos utilizados para a visualização e o controle do processo de simulação. É através de uma sessão de simulação que são ativados os simuladores do AMPLO.

A opção **FIM** encerra o ambiente de simulação retornando o controle do usuário para LAGO.

4 MÓDULO DO CONSTRUTOR DE MODELOS

4.1 Introdução

O módulo do construtor de modelos (MCM) permite obter uma estrutura de dados necessária ao processo de simulação (estrutura simulável) a partir de descrições de sistemas digitais armazenadas na base de dados (geradas por compiladores e/ou editores gráficos). O construtor de modelos é o módulo que integra as demais ferramentas existentes em AMPLO como mostra a figura 4.1.

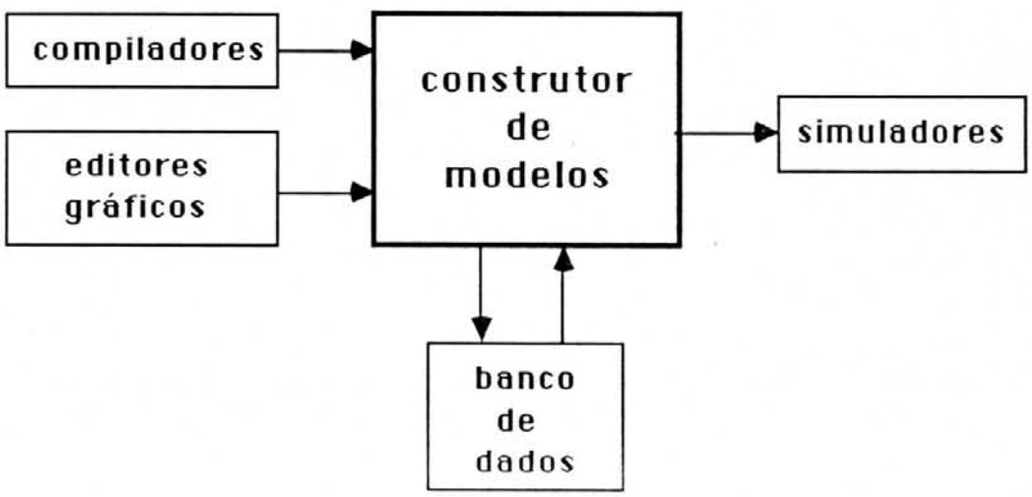

Figura 4.1: Integração do construtor de modelos.

Um modelo de simulação é uma rede de versões primitivas de agências [WAG 89]. Um modelo para uma versão composta de uma agência é construído selecionando-se versões para as ocorrências de alternativas existentes nesta versão composta, até que para todos os caminhos partindo da raiz da hierarquia tenham sido selecionadas versões primitivas. Este módulo cria o objeto modelo de si-

mulação, chamado de objeto MS, na base de dados, que está relacionado com a agência.alternativa.versão raiz que lhe deu origem.

A construção de um modelo de simulação exige a execução de dois processos :

• configurar a descrição, selecionando versões sempre que na descrição existirem ocorrências de alternativas (utilizando-se do mecanismo de navegação do "browser");

• resolver a hierarquia (expansão), obtendo uma rede final que contenha unicamente versões primitivas de agências.

A figura 4.2 ilustra a simplificação hierárquica de uma rede de versões compostas, que passa a ser constituída apenas de versões primitivas. Após a simplificação da hierarquia, as versões compostas intermediárias deixam de existir (figura 4.3).

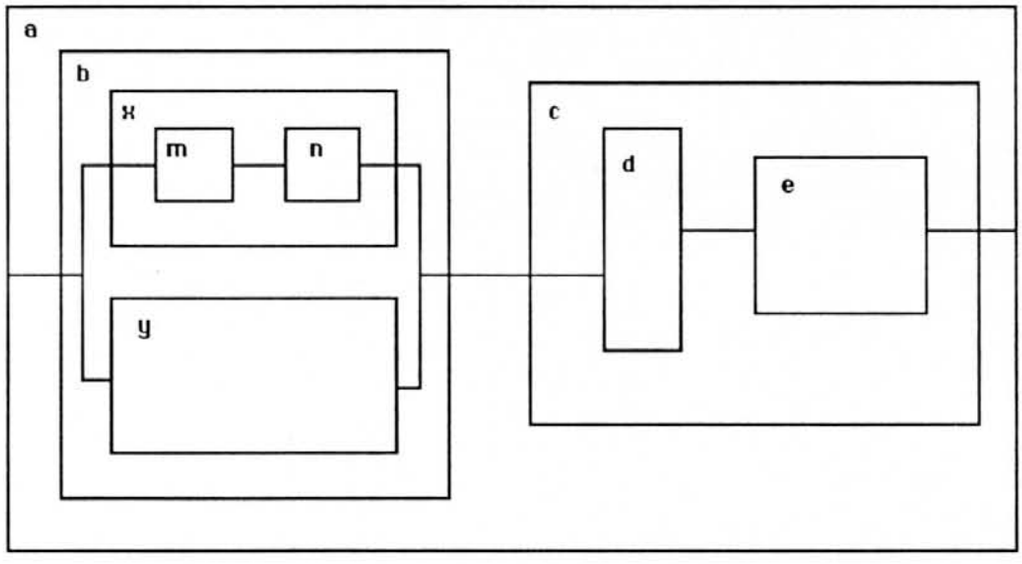

Figura 4.2: Rede de versões compostas.

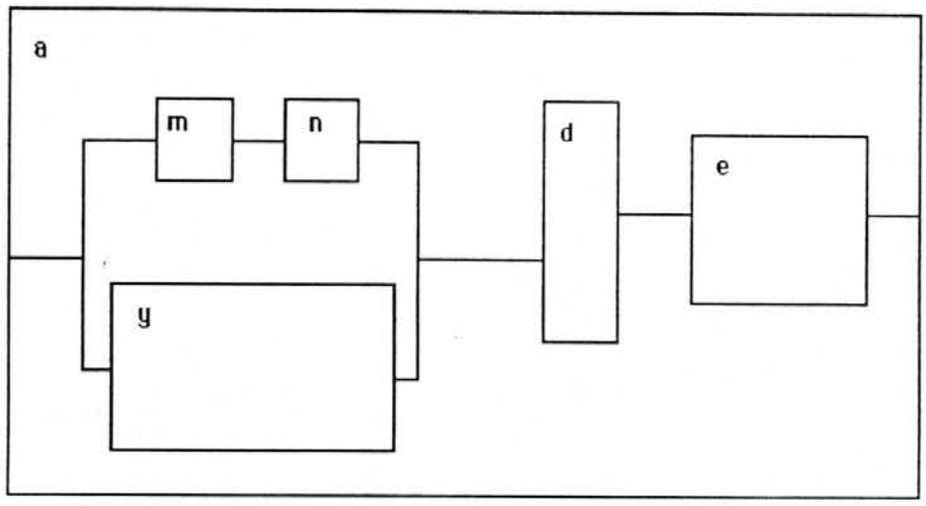

Figura 4.3: Rede de versões primitivas.

Os modelos de simulação são hierarquias de dois níveis apenas (redes de versões primitivas), ao contrário das hierarquias com qualquer profundidade, que são possíveis para as agências. Esta estrutura está relacionada com o algoritmo de simulação empregado [WAG 89], que exige modularidade na descrição, para que a simulação multi-nível seja possível, mas não permite hierarquia, para tornar a simulação mais eficiente.

Ao final da construção do modelo de simulação (MS), são criados 3 sub-objetos relacionados ao objeto MS, a saber :

• **AG** : representa a árvore geradora, que pode ser visualizada graficamente, e sobre a qual é possível realizar a navegação, inclusive com o propósito de editar o modelo de simulação. A árvore geradora contém os caminhos entre a agência a ser simulada (raiz da árvore) e as versões primitivas contidas no modelo de simulação final (folhas da árvore);

• **RA** : indica a rede de ocorrências de versões primitivas de agências representando a estrutura de dados específica para a simulação;

• **TS** : representa a tabela de símbolos hierárquica. Esta tabela realiza a conversão entre os nomes hierárquicos simulável, com o que é possível ao usuário,

durante a sessão de simulação, acessar agências e sinais de acordo com a descrição hierárquica por ele criada na base de dados.

O objeto MS deve ainda conter a lista de todos os objetos (agências, alternativas e versões) que foram utilizados na sua construção. Deste modo, podem ser realizadas consistências na base de dados no sentido de não se alterar ou remover algum destes objetos relacionados. O construtor de modelos permite que se associe ao objeto MS um cabeçalho contendo informações como o nome do projetista, data e hora da criação e outros comentários que o usuário julgar necessário.

O modelo de simulação pode ser descrito tanto textualmente como graficamente. Para ambos os tipos de descrição, a construção do modelo de simulação pode ser realizada através do método interativo ou do método automático. Estes dois métodos de construção do modelo de simulação são descritos a seguir.

4.2 Método interativo

No método interativo para a descrição gráfica, a construção do modelo de simulação é realizada com o usuário navegando através dos objetos armazenados na base de dados, utilizando-se dos mecanismos de seleção e visualização existentes no "browser". A cada nova seleção de uma versão, o sistema vai montando a árvore geradora do modelo de simulação.

Já para a descrição textual, no método interativo o usuário deve informar textualmente todos os caminhos existentes entre a agência a ser simulada (raiz da árvore) e as versões primitivas contidas no modelo de simulação final (folhas da árvore). Assim como realizado para a descrição gráfica, a árvore geradora do modelo de simulação vai sendo montada pelo sistema.

Durante a construção do modelo de simulação, o usuário pode desejar mudar o método que ele está utilizando. Deste modo, pode-se passar do método interativo para o automático através de uma tecla especial. Já o caso inverso, passagem do método automático para o interativo, só ocorre quando um critério selecionado no método automático não for satisfeito, sendo o controle transferido para o usuário.

4.3 Método automático

No método automático não existe preocupação com o tipo de descrição do modelo de simulação que está sendo construído, ou seja, não existe um procedimento específico para cada tipo de descrição. Esta característica existe porque neste método somente é informado o critério que será utilizado para a construção do modelo de simulação de modo que esta se dá sem a interferência do usuário. Existem dois critérios previamente estabelecidos para a seleção de versões em configurações dinâmicas. Os dois critérios existentes são os seguintes :

- opção pela mais recente das versões primitivas do nível escolhido;
- opção pela mais recente das versões compostas do nível escolhido.

Um exemplo é a procura de um modelo de simulação contendo unicamente versões compostas NILO (isto é, um modelo de simulação inteiramente descrito no nível de portas lógicas), e no qual sejam utilizadas as versões mais recentes (isto é, as de maior número de versões).

Antes de selecionar o critério que será utilizado, deve-se informar o nível no qual o modelo de simulação será construído. Se por algum motivo, em algum ponto da árvore geradora, o critério selecionado não puder ser cumprido, o ambiente envia uma mensagem informando que a construção do modelo de simulação para

este critério selecionado não poderá ser continuada. A partir deste momento, o usuário pode escolher entre continuar a construção utilizando-se do método interativo, continuar a construção com o método automático mas selecionando um novo critério ou ainda cancelar a construção do modelo de simulação.

Os motivos pelos quais o método automático pode ser interrompido são a não existência de uma versão primitiva (por exemplo, uma ocorrência possui 3 versões e todas elas são compostas) e a não existência de descrição para o nível indicado (por exemplo, uma ocorrência possui 5 versões, sendo 3 compostas e 2 primitivas e nenhuma delas está descrita no nível indicado). Caso o critério selecionado for o de versões compostas e não existir uma versão composta para uma determinada ocorrência, isto não causará uma interrupção no método automático uma vez que a situação indica que se chegou a uma folha da árvore geradora.

4.4 Inicialização do modelo

Para que uma sessão de simulação possa ser iniciada, deve-se selecionar os objetos que compõem o cenário de simulação para o modelo de simulação escolhido. Um destes objetos é o estado inicial do modelo de simulação. Este estado inicial, chamado de objeto ST_{inic}, é criado durante a fase de inicialização do modelo de simulação existente na interface principal do ambiente de simulação. O objeto ST_{inic} contém os valores iniciais de vários ou de todos os sinais existentes no modelo de simulação. No caso de terem sido inicializados apenas parte dos sinais, o simulador assumirá automaticamente para os demais o valor X (indeterminado). Existem várias estratégias de inicialização para a geração do objeto ST_{inic}, a saber :

1) colocar os sinais do modelo de simulação no valor 1, independente de possíveis situações de inconsistência, além de determinar onde devem ser colocadas as primeiras marcas nas agências LAÇO, se existirem;

2) colocar os sinais do modelo de simulação no valor 0, independente de possíveis situações de inconsistência, além de determinar onde devem ser colocadas as primeiras marcas nas agências LAÇO, se existirem;

3) colocar todos os sinais do modelo de simulação no valor X (indeterminado), além de determinar onde devem ser colocadas as primeiras marcas nas agências LAÇO, se existirem;

4) definir os valores dos sinais e as marcas nas agências LAÇO através de uma interação com o ambiente.

Na quarta estratégia de inicialização a interação do usuário com o ambiente é constituída pelas seguintes perguntas seqüênciais :

a) Deseja escolher um EVMS ? Permite que o usuário escolha um objeto EVMS armazenado na base de dados. Um objeto EVMS contém as ligações entre os estímulos (comportamento temporal dos sinais) e as entradas primárias do modelo de simulação. O objeto EVMS selecionado deve estar relacionado na base de dados com o modelo de simulação. O valor inicial dos sinais de entrada é obtido através do valor do estímulo no tempo 0.

b) Deseja atribuir valores manualmente ? Permite que o usuário atribua manualmente valores aos sinais de entrada do modelo de simulação. Esta pergunta é independente da resposta dada à pergunta anterior. Isto permite que mesmo escolhendo um objeto EVMS, o usuário pode alterar o valor de um sinal de entrada específico.

c) Deseja simular ? Através de um processo de simulação particular para a fase de inicialização e de acordo com o nível de simulação empregado, permite que sejam atribuídos valores aos sinais do modelo de simulação.

d) Deseja corrigir manualmente ? Independente da pergunta anterior, o usuário pode corrigir (alterar) os valores dos sinais que ele achar conveniente.

e) Deseja finalizar ? Encerra o processo de inicialização do modelo de simulação de acordo com a estratégia de definição por parte do usuário. Caso não seja desejado encerrar este processo, é realizado um retrocesso à terceira pergunta fazendo com que este processo permaneça em "loop" enquanto o usuário achar necessário. Isto é muito útil pois depois de corrigir manualmente o resultado de uma inicialização, o usuário pode repetir o processo particular de simulação, para a inicialização, agora com novos valores para os sinais.

A figura 4.4 apresenta o fluxograma do processo de inicialização de um modelo de simulação através da estratégia de definição do usuário.

Nesta última estratégia de inicialização, se o modelo de simulação for descrito graficamente, a atribuição e correção de valores de sinais é realizada através da função de apontamento. Caso o modelo de simulação esteja descrito textualmente, a atribuição dos valores segue a seguinte sintaxe :

< sinal > = < valor >

Neste caso, deve ser realizada a análise sintática sobre a expressão especificada e as seguintes consistências devem ser verificadas :

- verificar se o sinal especificado pertence ao modelo de simulação;

- verificar se o valor atribuído ao sinal está de acordo com o tipo de dado definido para este sinal. Esta última consistência também deve ser verificada se o modelo de simulação estiver descrito graficamente.

Antes de utilizar a estratégia de definição interativa, o usuário pode selecionar uma das três estratégias anteriores com a finalidade de servir como condição inicial dos valores dos sinais. Deste modo, os sinais que não forem alterados pela

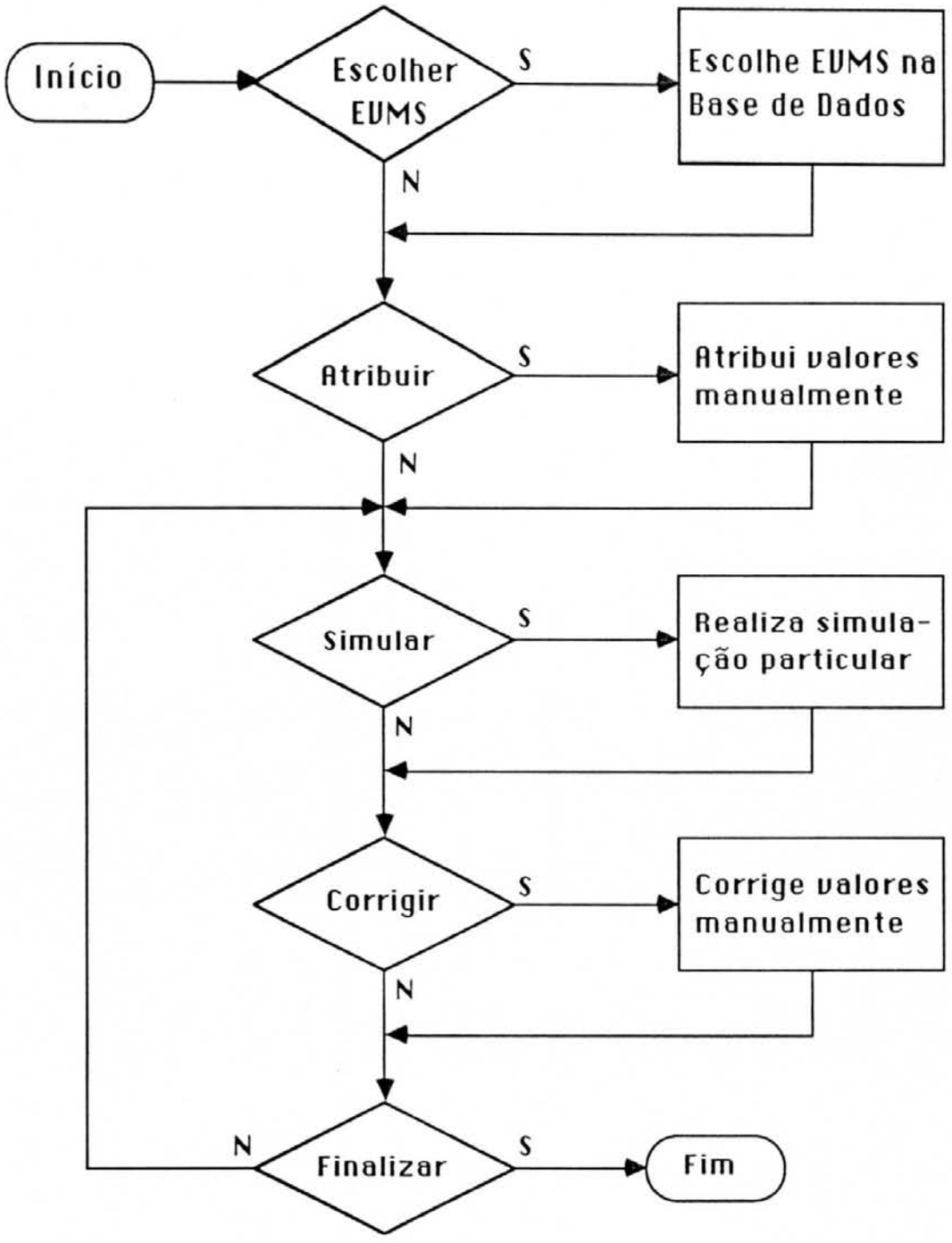

Figura 4.4: Processo de inicialização de um modelo de simulação.

definição interativa permanecerão com o valor inicial atribuído. O objeto ST_{inic} somente será armazenado na base de dados quando for selecionada a opção CRIAR no menu de estratégias de inicialização pertencente ao menu principal do ambiente de simulação.

5 MÓDULO DE DESCRIÇÃO DE ESTÍMULOS

5.1 Introdução

O módulo de descrição de estímulos (MDE) permite a geração dos estímulos a serem aplicados nas entradas primárias dos modelos de simulação e a sinais (variáveis) internos durante um processo de simulação. O arquivo de estímulos é composto por um conjunto de descrições de formas de onda de sinais. Cada descrição deste arquivo, chamada de estímulo, é constituída por uma seqüência de eventos.

Os estímulos podem ser descritos através de duas linguagens de descrição de estímulos: textualmente por intermédio da Linguagem de Descrição Temporal de Estímulos (LDTE) ou graficamente através da Linguagem Gráfica de Descrição de Estímulos (LGDE). Ambas as linguagens de descrição de estímulos criam um objeto na base de dados. Este objeto, chamado de objeto EV, indica a existência de um conjunto de estímulos não vinculado a um modelo de simulação específico. O objeto EV é armazenado na base de dados sob a forma de um arquivo.

As duas linguagens de descrição de estímulos geram objetos EV com o mesmo tipo de informação, de modo que ambas as descrições forneçam os mesmos dados aos modelos de simulação. Cada objeto EV é representado internamente por uma sequência de eventos com o seguinte formato:

ESTÍMULO TEMPO VALOR.

Este formato indica a existência de um evento no tempo TEMPO para o estímulo ESTÍMULO, ou seja, no tempo TEMPO, o estímulo ESTÍMULO recebe o valor VALOR.

As descrições LDTE (temporal) e LGDE (gráfica) são independentes dos modelos de simulação nos quais estes estímulos serão aplicados. Isto significa que, para a geração dos estímulos, não é necessária a existência de um modelo de simulação. A vinculação entre os estímulos e as entradas primárias dos modelos de simulação é realizada no módulo de vinculação de estímulos (MVE).

Como foi dito anteriormente, os estímulos são descrições de formas de onda dos sinais. Um objeto EV possui um número ilimitado de descrições de modo que ele pode conter uma ou infinitas descrições a critério do usuário. Cada estímulo descrito dentro de um objeto EV deve possuir um nome distinto, para que ele possa ser diferenciado dos demais de modo a facilitar sua seleção e referências. Também deve possuir um tipo de dado existente em uma das linguagens de descrição de hardware de AMPLO, podendo o usuário descrever em um mesmo objeto EV estímulos de diferentes tipos de dados. No entanto, todos os estímulos descritos em um objeto EV devem ter em comum o tipo da linguagem de descrição.

Para cada linguagem de descrição de estímulos (textual ou gráfica) além do objeto EV gerado na base de dados, deve ser armazenado o arquivo fonte que contém a descrição. Isto se torna necessário para futuras alterações na descrição e consultas às formas de onda dos estímulos.

5.2 Descrição Textual

A LDTE é uma linguagem textual utilizada para a descrição dos estímulos necessários para o processo da simulação. Esta descrição é armazenada na base de dados sob a forma de um arquivo textual, chamado de arquivo fonte da descrição. Como o processo de simulação precisa de informações adequadas ao seu procedimento, o arquivo fonte deve ser submetido a um processo de compilação para gerar

o arquivo utilizado pelos simuladores. Este arquivo, também armazenado na base de dados, é chamado de arquivo objeto da descrição.

O arquivo fonte é independente do arquivo objeto, ou seja, para armazenar o arquivo fonte na base de dados não é necessário passar este arquivo por um processo de compilação. Para que os arquivos objeto existam na base de dados, o seu correspondente arquivo fonte já deve existir na mesma. Na criação de um arquivo objeto na base de dados é estabelecido um relacionamento entre ele e seu correspondente arquivo fonte. Deste modo, qualquer alteração no arquivo fonte acarreta uma remoção do arquivo objeto da base de dados. Descrições textuais de estímulos somente podem ser utilizadas no ambiente de simulação se existir um arquivo objeto relacionado.

5.2.1 Construções da linguagem

A sintaxe do corpo principal da descrição textual de uma forma de onda é a seguinte :

SIGNAL < nome_do_sinal > : < tipo_do_sinal >
BEGIN
    < corpo_da_descrição >
END_SIGNAL

onde :

< nome_do_sinal > : representa o nome do estímulo cuja forma de onda será descrita;

< tipo_do_sinal > : representa o tipo de dado do sinal de acordo com os tipos existentes nas linguagens disponíveis em AMPLO;

< corpo_da_descrição > : representa a descrição da forma de onda do estímulo indicado.

As palavras SIGNAL, BEGIN e END_SIGNAL são palavras chave da sintaxe da descrição. Toda a sintaxe da linguagem é apresentada no anexo A-1. Neste ponto, serão apresentadas as principais construções da linguagem com alguns exemplos ilustrativos. O < corpo_da_descrição > pode ser visto como uma lista de ações/comandos com diferentes comportamentos, de modo a produzirem a forma de onda desejada.

As seguintes construções estão presentes na linguagem :

FOR : representa a duração em termos de número de unidades de tempo de simulação (passo adotado pelo mecanismo de avanço de tempo do simulador) na qual o estímulo possui determinado valor. Esta instrução representa um período relativo, ou seja, a partir da última unidade de tempo o estímulo passa a valer um novo valor indicado durante um determinado número de unidades de tempo especificado. Por exemplo : um determinado estímulo está descrito até a unidade de tempo 10. Se for indicado o comando FOR 15 : 1, este estímulo passa a valer 1 da unidade de tempo 10 até a unidade de tempo 25.

UP_TO : representa o instante final de unidades de tempo no qual o estímulo descrito ainda possui o mesmo valor. Esta instrução, ao contrário da anterior, indica um período absoluto, ou seja, a partir da última unidade de tempo até a unidade de tempo indicada o estímulo passa a ter o novo valor especificado. Nesta instrução deve haver uma consistência para verificar se a unidade de tempo final indicada não é inferior à última unidade de tempo em que o estímulo possui descrição. Por exemplo, para produzir a mesma forma de onda do exemplo apresentado para a instrução FOR, a sintaxe desta instrução deveria ser : UP_TO 25 : 1.

FOREVER : indica que o valor atual do estímulo se mantém inalterado até o final da simulação. Não existe um novo evento para este estímulo.

Abaixo está um exemplo de uma descrição de uma forma de onda de um estímulo que utiliza estas três instruções apresentadas.

```
SIGNAL BOOL : BOOLEANO
BEGIN
    FOR 30 : 1;
    UP_TO 55 : 0;
    FOREVER : 1;
END_SIGNAL
```

Neste exemplo, o estímulo BOOL do tipo BOOLEANO assume o valor 1 por 30 unidades de tempo (do tempo 0 até o tempo 30); então, até a unidade de tempo 55, assume o valor 0 (por 25 unidades de tempo). Após a unidade de tempo 55 é especificado que este estímulo irá assumir o valor 1 até o final da simulação. O desenho da forma de onda da descrição deste estímulo é apresentado na figura 5.1.

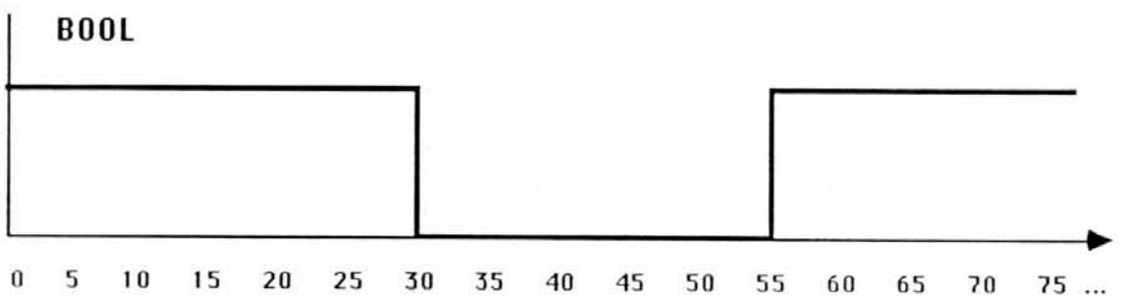

Figura 5.1: Forma de onda para a descrição do estímulo BOOL.

A gramática da linguagem permite que se realizem aninhamentos de instruções. Por exemplo :

```
SIGNAL SIN3 : VARIABLE INTEGER
BEGIN
   UP_TO 10 : 5;
   FOR 50 : BEGIN
                FOR 10 : 10;
                FOR 15 : 20;
                FOREVER : 10;
            END
   FOREVER : 5;
END_SIGNAL
```

Quando existe o aninhamento de instruções as seguintes situações podem ocorrer :

• o período externo é menor do que a soma dos períodos internos. Neste caso, quando o período interno alcança o período externo, o controle é transferido para a próxima instrução que existe fora do aninhamento e as instruções internas que excedem o período externo são desconsideradas.

• o período externo é maior do que a soma dos períodos internos. Neste caso, ocorre um erro na compilação desta descrição e o estímulo não pode ser utilizado.

• o período externo é igual à soma dos períodos internos. Isto representa uma descrição periódica correta para a forma de onda do estímulo.

Estes procedimentos de verificação são realizados durante o processo da compilação. Outra facilidade encontrada na linguagem é a descrição periódica da forma de onda. A descrição do período básico é especificado entre as palavras-chave **PERIOD** e **END_PERIOD**. Esta descrição pode ser realizada também de duas maneiras :

• através da instrução **PERIOD FOR** : a forma de onda é periódica durante um determinado número de unidades de tempo.

• através da instrução **PERIOD FOREVER** : a forma de onda do estímulo descrito contém a mesma descrição periódica ao longo do tempo.

O exemplo abaixo apresenta a descrição de uma forma de onda utilizando-se a instrução PERIOD. Esta descrição representa a forma de onda apresentada na figura 5.2.

```
SIGNAL SIG2 : TERMINAL INTEGER
BEGIN
    FOR 4 : 10;
    PERIOD FOR 8
        FOR 2 : 20;
        FOR 2 : 15;
    END_PERIOD
    UP_TO 20 : 0;
    FOREVER : 57;
END_SIGNAL
```

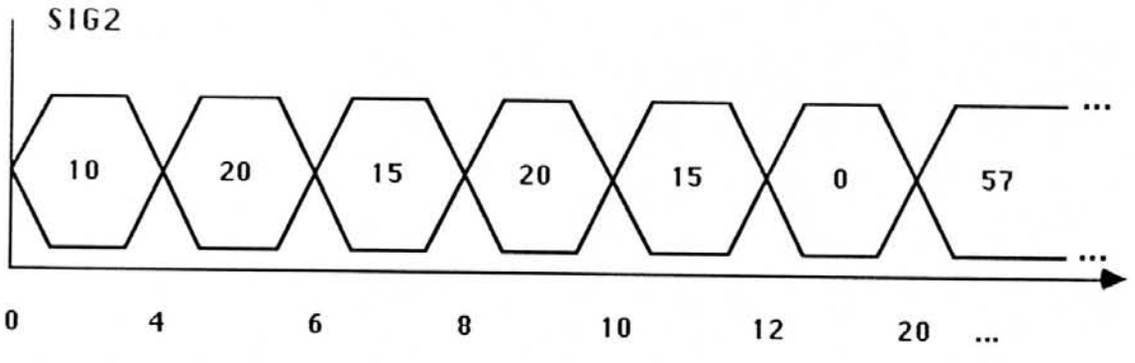

Figura 5.2: Forma de onda para a descrição do estímulo SIG2.

Esta forma de onda indica que até o tempo 4 este estímulo possui o valor 10. No tempo 4 existe um evento e o estímulo assume o valor 20 até o tempo 6 e, assim por diante. A representação gráfica desta forma de onda não tem significado quanto à amplitude apresentada (fazendo comparação com um sinal booleano), ela apenas foi especificada desta maneira.

5.3 Descrição Gráfica

A LGDE é uma linguagem utilizada para descrever graficamente as formas de onda dos estímulos. A descrição dos estímulos é armazenada na base de dados sob a forma de dois arquivos: o arquivo fonte que contém a descrição gráfica dos estímulos e o arquivo objeto que contém as informações necessárias para serem usadas em um processo de simulação. Ao contrário do que acontece na linguagem de descrição textual, os dois arquivos pertencentes à descrição gráfica permanecem sempre juntos. Isto significa que sempre que for realizada uma operação sobre uma descrição gráfica dos estímulos (por exemplo: criar, alterar, remover), esta operação será realizada sobre os dois arquivos na base de dados.

A figura 5.3 mostra a interface de interação para a descrição gráfica de estímulos. Na área de menu e orientação é informado o nome do objeto no qual o objeto EV descrito será relacionado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa). Na área de trabalho é exibida a janela de visualização utilizada para realizar a descrição dos estímulos. A escala de unidades de tempo possui um valor "default" mas pode ser modificada de acordo com as necessidades do usuário. Porém, a escala permanece constante para todas as descrições dos estímulos do objeto EV envolvido. Caso uma descrição de um estímulo seja maior do que a janela de visualização apresentada, existe a possibilidade de se realizar um "scroll" nesta janela.

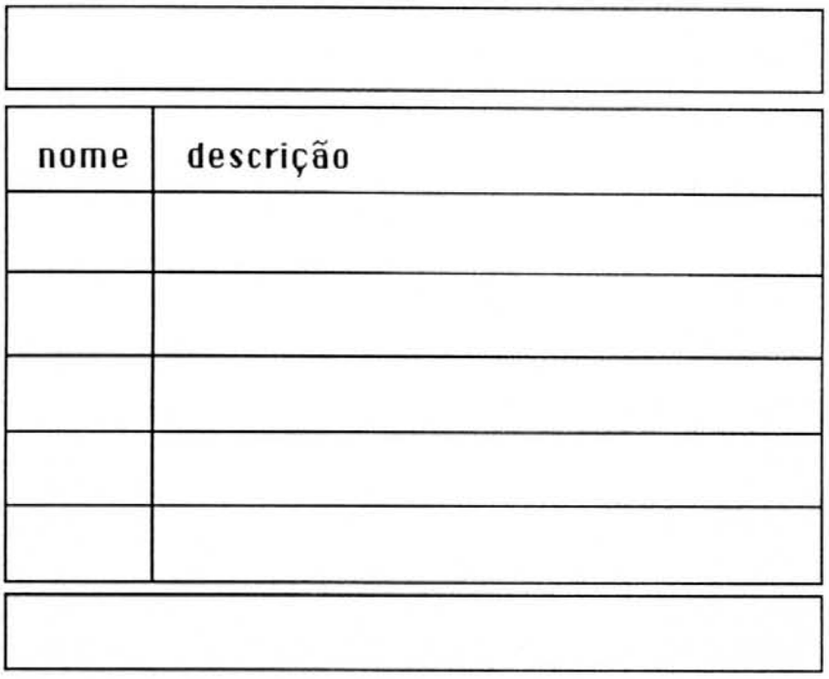

Figura 5.3: Interface da descrição gráfica de estímulos.

Para iniciar uma descrição da forma de onda de um estímulo, deve ser informado primeiramente o nome do estímulo que será descrito e o seu tipo de dado. É apresentado menu horizontal com todos os tipos de dados existentes no AMPLO para que um seja selecionado. A descrição gráfica deve apresentar todas as facilidades encontradas na descrição textual de estímulos. Deste modo, construções do tipo PERIOD, FOR, FOREVER e UP_TO devem estar presentes na descrição gráfica. As formas de onda dos estímulos são desenhadas com o auxílio do teclado, através das setas de movimentação do cursor, ou através do "mouse", se este estiver presente. De acordo com o tipo de dado do estímulo, a interação torna-se levemente diferente.

Para especificar uma forma de onda periódica, devem ser informadas as unidades de tempo que indicam o início e o final do período a ser repetido. Uma consistência é realizada para verificar se o período é válido e se existe uma descrição do estímulo dentro deste período. Após, informa-se por quantas unidades de tempo se estenderá este período. Este valor pode ser um número limitado de unidades de

tempo, correspondendo ao comando PERIOD FOR na linguagem textual, ou um número ilimitado, correspondendo ao comando PERIOD FOREVER que é indicado por uma tecla especial. Caso seja um número limite de unidades de tempo e este número não for múltiplo do tamanho do período informado, a forma de onda do estímulo se interrompe no momento do último instante de tempo informado para a repetição do período. O instante inicial de unidades de tempo em que se processa o comando PERIOD é a unidade de tempo posterior à última que contém uma informação pertencente à forma de onda do estímulo.

Outra facilidade encontrada na descrição gráfica dos estímulos é a repetição do último evento deste estímulo por um determinado número de unidades de tempo. Este procedimento corresponde aos comandos FOR, UP_TO e FOREVER na linguagem textual. Estes comandos são ativados por teclas especiais onde: para o comando FOR indica-se por quantas unidades de tempo vai se repetir este valor; para o comando UP_TO indica-se a última unidade de tempo em que este valor se repetirá; e para o comando FOREVER não é informado nada pois a forma de onda permanece inalterada até o final da simulação.

Sinais booleanos podem assumir três valores: 0, 1 ou X (que representa o estado indeterminado). Para representar os estados 0 e 1, a forma de onda é descrita como mostrado no estímulo SIN da figura 5.4. O estado 0 é indicado quando o desenho da forma de onda está em baixo e o estado 1 é indicado quando o desenho da forma de onda está em cima. Para representar um estado indeterminado, informam- se as unidades de tempo que indicam o início e o final do estado indeterminado. Após estas informações, que são dadas através de teclas especiais, o período compreendido entre o instante de tempo inicial e o final fica hachurado como é mostrado para o estímulo SN2 da figura 5.4.

No nível NILO, além dos três estados possíveis que os estímulos podem assumir (0, 1 e X), está associado a cada estado uma de quatro intensidades: E (external), D (driven), W (weak) e Z (high impedance). Para cada estado do estímulo

deve ser informada a sua intensidade, sendo que seu valor "default" é E (external). O estímulo SINILO da figura 5.4. é descrito em NILO. Os estímulos, que não são do tipo booleano, podem assumir qualquer valor. Para estes estímulos, deve-se informar o seu valor e a sua duração em unidades de tempo (comandos FOR, UP_TO e FOREVER). A forma de onda para este tipo de estímulo é mostrada pelo sinal S24 da figura 5.4.

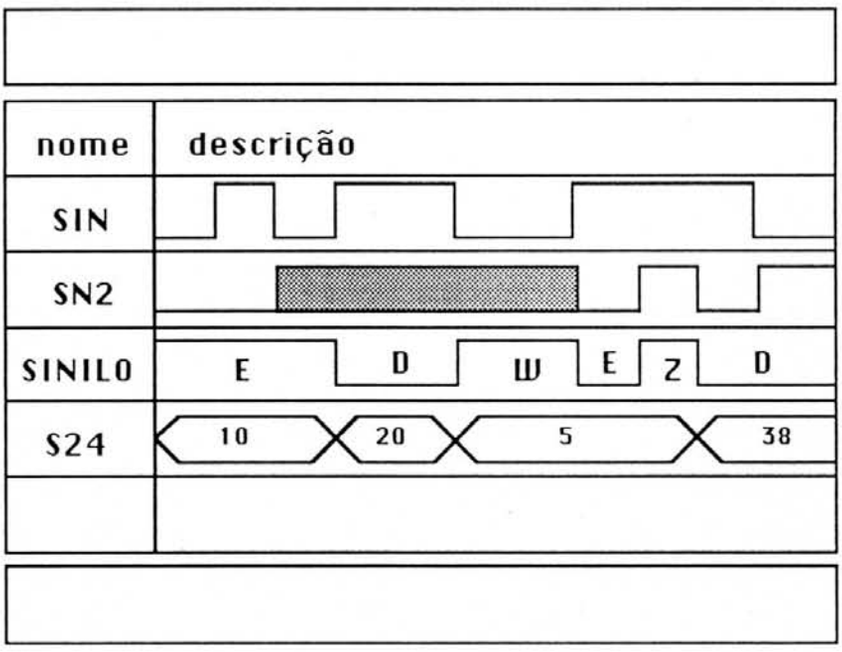

Figura 5.4: Descrição gráfica de estímulos.

Existe uma função, ativada por uma tecla especial, que permite concluir a descrição de uma forma de onda de um estímulo e iniciar outra. Se em algum ponto deste processo for constatado que um estímulo anteriormente descrito está incorreto, existe a possibilidade de selecionar este estímulo e alterar a sua descrição. Para encerrar com o processo de descrição das formas de onda dos estímulos, duas alternativas estão disponíveis para seleção (todas executadas através de teclas especiais):

- abandonar o processo, perdendo todo o trabalho executado até o momento;

• efetivar a descrição dos estímulos na base de dados. O nome do objeto EV que será criado na base de dados deve ser informado. Uma consistência é realizada para verificar se o nome fornecido ao novo objeto EV é diferente dos demais nomes dos objetos EV que estão relacionados com o mesmo objeto ao qual ele deve ser relacionado.

6 MÓDULO DE VINCULAÇÃO DE ESTÍMULOS

6.1 Introdução

O módulo de vinculação de estímulos (MVE) permite que se realize a ligação das descrições de estímulos com as entradas primárias de um modelo de simulação. A descrição do estímulo envolvido pode ser tanto textual (descrito através da linguagem LDTE) como gráfica (descrito através da linguagem LGDE).

Esta ligação cria um novo objeto na base de dados, chamado de EVMS, composto por triplas cujos elementos são o nome da entrada primária do modelo de simulação, o nome do objeto EV onde está o estímulo a ser relacionado e o nome do estímulo contido no objeto EV que será vinculado à entrada primária do modelo de simulação.

Um objeto EVMS estabelece um relacionamento entre um modelo de simulação com um ou mais objetos EV, uma vez que as entradas primárias do modelo de simulação podem ser associadas com estímulos de diferentes objetos EV. Porém, não é necessário que todos os estímulos descritos em um objeto EV envolvido na vinculação sejam associados às entradas primárias do modelo de simulação. Na realidade, o essencial é que todas as entradas primárias do modelo de simulação tenham sido associadas com algum estímulo de um objeto EV.

Existem duas formas de se realizar uma vinculação: textualmente ou graficamente. Ambas as formas de vinculação criam descrições completamente equivalentes entre si na base de dados (mesmos objetos EVMS). Isto significa que se uma determinada vinculação for realizada de uma forma, quando for necessário realizar

uma alteração na mesma, a forma de vinculação não precisa ser a mesma do que a anterior.

As duas formas de vinculação diferenciam-se entre si unicamente pela interação com o usuário. Enquanto que a vinculação textual é mais indicada para usuários com mais experiência e conhecedores dos objetos armazenados na base de dados, a vinculação gráfica é aconselhada para os usuários que ainda não possuem um conhecimento total dos objetos envolvidos na vinculação ou por aqueles que preferem os recursos gráficos pela sua clareza e consistência mais fácil. Para o apontamento dos sinais de entrada do modelo de simulação, por exemplo, na vinculação textual o usuário pode informar o nome de um sinal que não exista (neste caso o ambiente envia uma mensagem de erro), enquanto que na vinculação gráfica o usuário necessita somente apontar um dos sinais que estão exibidos na interface do modelo de simulação.

Quando da ativação da operação de vinculação, na interface principal do ambiente de simulação, deve ser informado o nome da agência, da alternativa, da versão e do modelo de simulação com quem o objeto EVMS a ser manipulado estará relacionado. Estes dados podem ser informados textualmente em ordem seqüencial ou através do "browser". Após é selecionado o tipo da vinculação a ser realizada.

Para encerrar o processo de vinculação, uma das seguintes alternativas pode ser selecionada (todas elas executadas através de teclas especiais):

- abandonar o processo de vinculação. Com isto, nada é criado na base de dados e, todo o trabalho de vinculação executado até o momento é perdido;

- salvar o estado atual do processo de vinculação em um arquivo temporário para concluir em outro momento (nada é realizado na base de dados);

- efetivar a vinculação na base de dados. O nome do objeto EVMS, que será criado na base de dados, deve ser informado. Uma consistência é realizada para

verificar se o nome fornecido é diferente dos demais nomes dos objetos EVMS relacionados com o modelo de simulação dado pela hierarquia agência.alternativa.versão.modelo.

6.2 Vinculação Textual

A vinculação textual permite que o usuário realize a ligação entre descrições de estímulos com as entradas primárias do modelo de simulação informando textualmente os dados necessários para realização desta operação.

A interface de interação mostrada na figura 6.1 é utilizada para realizar a vinculação textual. Na área de menu e orientação é informado o modelo de simulação (agência.alternativa.versão.modelo) que estará relacionado com o objeto EVMS criado. A área de trabalho é constituída por duas janelas de visualização. Na janela da tabela é apresentada a tabela de vinculações onde será realizada a vinculação dos estímulos com as entradas primárias do modelo de simulação. Na janela de consulta são apresentadas as consultas realizadas à base de dados.

Quando selecionado o modelo de simulação, o ambiente verifica quantas são as entradas primárias fazendo uma pesquisa na rede de agências do modelo de simulação. Deste modo, o número de linhas da tabela de vinculação é estabelecido pelo número de entradas primárias do modelo de simulação e a tabela de vinculações pode ser desenhada automaticamente.

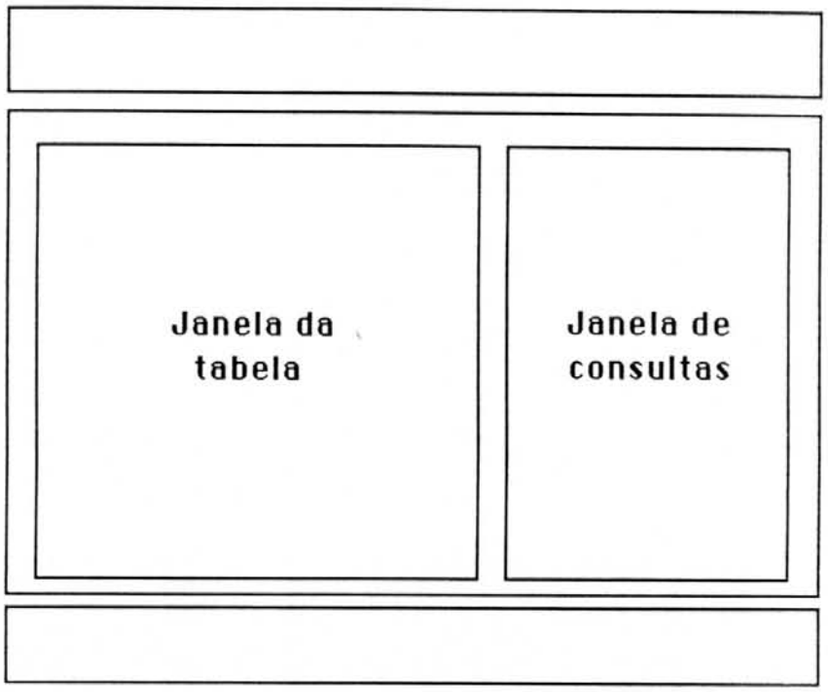

Figura 6.1: Interface para a vinculação textual.

Como as entradas primárias do modelo de simulação são conhecidas, a primeira coluna da tabela de vinculações, representada pelo nome da entrada primária do modelo de simulação, pode ser preenchida automaticamente após a construção da tabela de vinculações. Deste modo, para realizar a vinculação deve ser informado somente o nome do objeto EV e o nome do estímulo pertencente a este objeto para cada uma das entradas primárias apresentadas na tabela de vinculações cujo formato é mostrado na figura 6.2.

Para cada vinculação realizada (representa um preenchimento em uma linha da tabela de vinculações) são verificadas as seguintes consistências :

- o nome do objeto EV deve existir na base de dados e estar relacionado com a mesma agência.alternativa do modelo de simulação, relacionado com a mesma agência ou ainda sem relacionamento;

- o nome do estímulo deve pertencer ao objeto EV indicado (se este já tiver sido indicado);

• o tipo de sinal do estímulo deve ser compatível com a entrada primária do modelo de simulação, de acordo com a tabela de compatibilidades [WAP 88].

sinal	objeto EV	estímulo
nome 1		
nome 2		
nome 3		
...		
nome n-2		
nome n-1		
nome n		

Figura 6.2: Formato da tabela de vinculações.

Se alguma destas consistências não for verificada, o ambiente envia uma mensagem de erro. Todas as entradas primárias do modelo de simulação devem ser associadas a algum estímulo de um objeto EV. No momento da criação do objeto EVMS na base de dados, se esta condição não for cumprida, o ambiente envia outra mensagem de erro informando que alguma entrada primária do modelo de simulação não foi associada com algum estímulo.

Como a tabela de vinculações é construída em função do número de entradas primárias do modelo de simulação, pode ocorrer que para um determinado modelo de simulação o número de entradas primárias seja muito grande, de modo que a tabela de vinculações não possa ser totalmente exibida na área de trabalho do ambiente. Para contornar este problema, existe a possibilidade de se realizar um "scroll" para movimentar a janela de visualização.

A manipulação da tabela de vinculações é realizada através do movimento de uma barra em reverso sobre os campos a serem preenchidos. Quando esta barra em reverso estiver sobre um campo desejado, o usuário seleciona este campo e entra com o dado desejado.

Mesmo sendo a vinculação textual o processo mais indicado para usuários mais experientes, deve existir a possibilidade de se realizar consultas à base de dados sobre os objetos envolvidos em uma vinculação. Estas consultas são realizadas através de teclas especiais previamente definidas, sendo apresentadas na janela de consultas. Esta janela permite que seja realizado um "scroll" para melhor visualização das informações. As seguintes consultas neste tipo de vinculação estão disponíveis ao usuário :

- visualizar todos os objetos EV existentes na base de dados. Estes objetos EV devem estar relacionados à agência.alternativa, relacionados à agência ou sem relacionamento;

- visualizar a descrição das formas de onda dos estímulos do objeto EV indicado;

- pesquisar as informações relevantes sobre a entrada primária indicada, tais como tipo de sinal e largura em bits do sinal.

6.3 Vinculação Gráfica

A vinculação gráfica do MVE (módulo de vinculação de estímulos) permite que através das informações gráficas e interativas apresentadas sejam realizados apontamentos e seleções nos objetos adequados para que se realize a ligação entre as descrições dos estímulos com as entradas primárias do modelo de simulação.

A figura 6.3 mostra a interface de interação com o usuário para esta forma de vinculação. Esta interface é constituída de 2 áreas de orientação e de 4 janelas de visualização utilizadas para realizar a vinculação. Na área de menu e orientação é informado o nome do modelo de simulação (agência.alternativa.versão.modelo) que estará relacionado com o objeto EVMS criado. A janela 1, chamada

de janela de descrições, mostra as descrições das formas de onda dos estímulos do objeto EV ativo no momento. Esta janela de descrições pode ser tanto textual como gráfica de acordo com o tipo da descrição. A janela 2, chamada de janela de vinculação, apresenta a tabela das vinculações realizadas. A janela 3, chamada de janela da interface, exibe o desenho da interface do modelo de simulação indicado. A janela 4, chamada de janela de estímulos, mostra a lista dos objetos EV armazenados na base de dados, possíveis de serem utilizados para a vinculação de estímulos ao modelo de simulação selecionado.

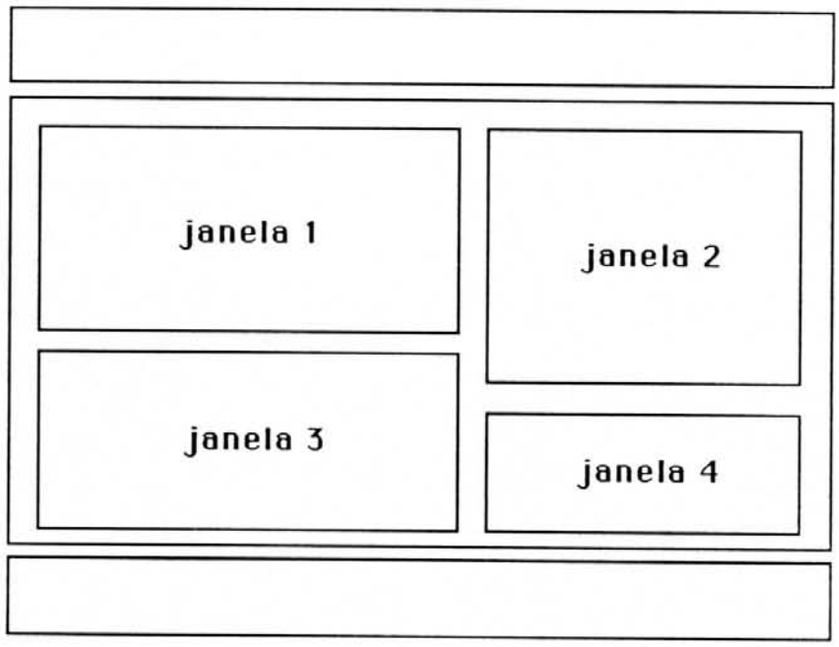

Figura 6.3: Interface para a vinculação gráfica.

Com os dados fornecidos na ativação da vinculação gráfica, as três últimas janelas de visualização podem apresentar suas informações. Deste modo, na janela de vinculação pode ser construída a tabela de vinculação da mesma maneira que na vinculação textual. Já na janela da interface, através de uma pesquisa à base de dados, obtém-se as informações gráficas da interface do modelo de simulação indicado, enquanto que para a janela de estímulos obtém-se, também da base de

dados, o nome de todos os objetos EV que estão relacionados à agência.alternativa indicada, relacionados à agência e aqueles que não tem relacionamento com ninguém.

Na vinculação gráfica existe a janela de visualização ativa que é aquela janela na qual está sendo realizado algum procedimento. Ela é representada por um traço duplo em sua borda com a cor diferente das demais. Através de uma tecla especial, pode-se mudar a janela de visualização ativa. Na janela ativa existe a possibilidade de se realizar operações de "scroll", expansão e contração dos limites da janela.

De modo diferente do que ocorre na vinculação textual, a tabela de vinculações não pode ser manipulada, ou seja, preenchida pelo usuário. A tabela de vinculações é automaticamente preenchida pelo ambiente, servindo como orientação ao usuário para saber qual a entrada primária que está sendo vinculada, quais já foram vinculadas e as que ainda faltam vincular.

Para se realizar uma vinculação, o primeiro passo a ser tomado é fazer um apontamento a um sinal de entrada do modelo de simulação apresentado na janela de interface. Após o apontamento do sinal, o ambiente automaticamente coloca em destaque este sinal na tabela de vinculações, indicando para o usuário que naquela linha da tabela está se processando uma vinculação. Apontado o sinal de entrada a ser vinculado, o usuário deve selecionar um dos nomes dos objetos EV apresentados na janela de estímulos. Quando selecionado um objeto EV, o seu nome fica em destaque na janela de estímulos e é colocado na coluna do objeto EV da linha em destaque na tabela de vinculações. Imediatamente a descrição das formas de onda dos estímulos deste objeto EV é apresentada na janela de descrições. Para completar a vinculação desta entrada primária, deve ser selecionado o nome do estímulo na janela de descrições, fazendo com que a coluna do nome do estímulo da linha em destaque na tabela de vinculações seja preenchida automaticamente pelo ambiente.

Na vinculação gráfica somente duas consistências devem ser analisadas pelo ambiente:

- verificar se o sinal apontado na janela da interface representa uma entrada primária;

- verificar se o tipo de sinal do estímulo indicado na janela de descrições é compatível com a entrada primária selecionada.

Para se realizar uma nova vinculação deve ser apontado um nome de sinal na janela da interface. Como ainda existe um objeto EV em destaque na janela de estímulos, procedente de uma seleção anterior, este objeto EV não precisa ser selecionado caso seja o mesmo a ser utilizado. Deste modo, realiza-se somente a seleção do estímulo na janela de descrições e, o ambiente preenche automaticamente as colunas do objeto EV e o nome do estímulo da linha em destaque na tabela de vinculações.

Durante o processo de vinculação gráfica, as vinculações já realizadas podem ser alteradas pelo usuário bastando para isto que seja apontada alguma entrada primária na janela da interface que já tenha sido vinculada. O novo objeto EV e o novo estímulo selecionados serão sobrepostos aos anteriores na linha da tabela de vinculações correspondente à entrada primária em destaque. As consistências continuam sendo verificadas.

7 MÓDULO DA LINGUAGEM DE COMANDOS

7.1 Introdução

O módulo da linguagem de comandos (MLC) fornece ao usuário funções para o controle da sessão de simulação. Após definido o modelo de simulação e o cenário de simulação, o usuário tem à sua disposição a linguagem de comandos. Ela foi parcialmente baseada na linguagem TPDL [PRI 89] e oferece recursos para visualizar, salvar e alterar valores ou sequências de valores de sinais quaisquer do modelo de simulação que está sendo simulado, assim como para controlar o avanço do tempo de simulação, pela definição de pontos de parada incondicionais ou condicionados à avaliação de expressões booleanas.

É possível, entre outras funções, salvar estados intermediários da simulação, que são armazenados na base de dados como objetos, criar um objeto RE (resultados), contendo todo o comportamento temporal de sinais escolhidos pelo usuário e criar objetos BAT (batch), contendo uma sequência de comandos executados.

A figura 7.1 apresenta a estrutura completa da interface da linguagem de comandos.

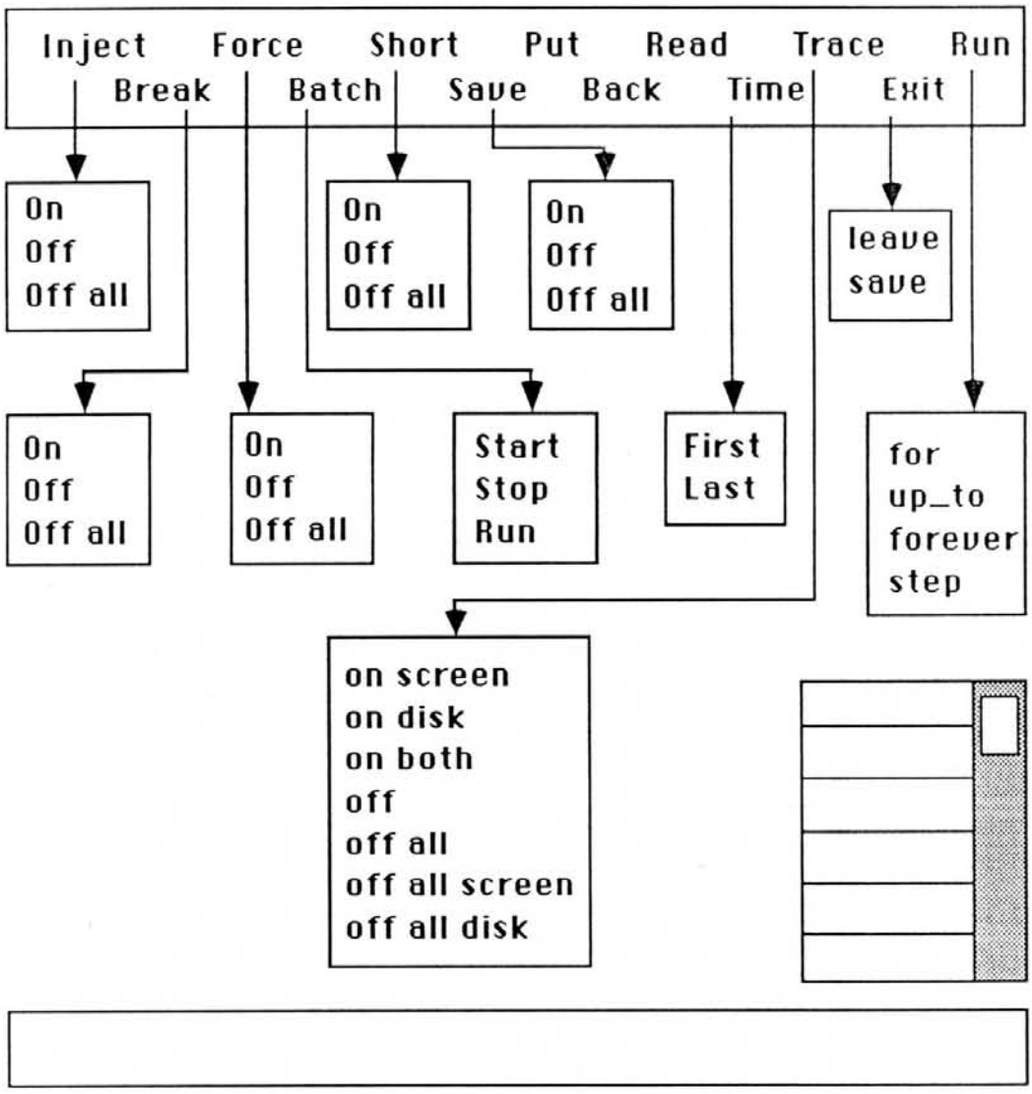

Figura 7.1: Interface da linguagem de comandos.

7.2 Recursos de visualização

No MLC (módulo da linguagem de comandos) são oferecidos recursos de visualização gráfico-interativos de modo que os comandos disponíveis se tornem mais naturais para o usuário e a sua interação com o usuário a mais amigável possível [STU 90, HAR 89].

Nos comandos nos quais existe a necessidade de indicação de um sinal ou de uma variável interna para a sua operação, existe a chamada para a função de apontamento. Nesta função é exibida na área de trabalho a representação gráfica da agência.alternativa.versão (se esta versão tiver sido descrita graficamente). Com o auxílio do cursor, que possui livre movimentação, pode-se apontar para um sinal desejado. Se esta agência.alternativa.versão não for primitiva e o sinal desejado estiver no interior de uma ocorrência aponta-se para esta e a sua representação gráfica será exibida se sobrepondo à anterior. Deste modo, pode-se apontar para o sinal desejado. Porém, se ainda esta agência não for primitiva, todo o processo pode ser repetido. Existe a possibilidade de se voltar um nível na hierarquia da descrição se a ocorrência selecionada não for a desejada. Caso uma ocorrência selecionada estiver descrita somente de forma textual uma mensagem de aviso é exibida informando que esta ocorrência não pode ser exibida graficamente. Este recurso segue os mesmos princípios que o "browser".

Outra facilidade encontrada no MLC é o cardápio SHOW exibido no canto inferior direito como mostra a figura 7.1. O cardápio SHOW é utilizado para apresentar listas de elementos relacionados ao comando ativo do momento, sendo uma área temporária que permanece ativa enquanto se está realizando uma operação que necessite de sua manipulação. O cardápio SHOW é constituído de duas partes : a área onde é apresentada a lista dos elementos, existindo a possibilidade de "scroll" horizontal e vertical e uma área que contém um indicador da posição relativa dos elementos apresentados em relação ao tamanho total da lista. À medida que se realiza um "scroll" vertical na lista dos elementos, o indicador muda de posição. O cardápio SHOW é utilizado para dois tipos de operações : consulta aos elementos que estão relacionados com o comando indicado e cancelamento de um comando já realizado. Estas operações são realizadas através da manipulação da lista dos elementos que está relacionada com o comando indicado.

Os comandos disponíveis na interface de comandos interagem com o usuário de duas formas : textual e graficamente. Quando a interface de comandos

é ativada, a interação gráfica é assumida como "default". Deste modo, todos os comandos executados na sessão de simulação serão especificados graficamente. Através de uma tecla especial, que está ativa durante a sessão de simulação, pode-se mudar a forma de interação dos comandos (de gráfica para textual e vice-versa). Em todos os comandos onde é necessária a interação textual, a entrada do texto é realizada através da área de entrada de eco. Quando for realizada uma indicação textual de um sinal do modelo de simulação que está sendo simulado, deve-se informar toda a hierarquia na qual o sinal está incluído. Isto se deve ao fato de que os sinais de diferentes agências, pertencentes ao mesmo modelo de simulação, podem possuir nomes iguais.

7.3 Comandos de simulação

A seguir são descritos os comandos fornecidos pelo ambiente de simulação na ordem em que estão dispostos na interface de comandos do módulo da linguagem de comandos (MLC). Os comandos de simulação estão disponíveis quando o controle da simulação está com o usuário no MLC. Quando o modelo de simulação está sendo simulado, o usuário não tem controle sobre o processo de simulação. O controle retorna para o usuário quando o simulador completa o processo de simulação. Neste ponto, ele pode realizar todos os procedimentos necessários para continuar a simulação.

7.3.1 Inject

O comando INJECT permite que se conecte um estímulo pertencente a um objeto EV armazenado na base de dados com um sinal interno do modelo de simulação que está sendo simulado. O estímulo conectado tem prioridade sobre as modificações realizadas no sinal interno provenientes da simulação. Deste

modo o sinal interno possui seu comportamento determinado pela forma de onda do estímulo.

Este comando pode ser especificado tanto textual como graficamente. Para ambas as especificações deve ser verificada a compatibilidade do estímulo indicado no objeto EV com o sinal interno do modelo de simulação. Para a especificação textual, a sintaxe do comando é a seguinte:

INJECT < inject_comando >

< inject_comando > ::= < nome_sinal_interno > =
< especificação_objeto_EV >

< especificação_objeto_EV > ::= < nome_objeto_EV > ,
< nome_estímulo >

Exemplo : INJECT sin = obj,estim

Na especificação gráfica, a representação gráfica da agência.alternativa.versão é exibida na área de trabalho para que um sinal interno seja apontado. Após o apontamento, todos os objetos EV armazenados na base de dados que são relacionados com a agência.alternativa, relacionados somente com a agência e os que não possuem relacionamento são apresentados para que um seja selecionado. Deste modo, todas as descrições das formas de onda dos estímulos contidas neste objeto EV selecionado são exibidas para que uma seja selecionada. Estas descrições podem ser tanto textuais como gráficas, dependendo da linguagem utilizada no módulo de descrição de estímulos (MDE).

Para o comando INJECT o significado das opções que estão disponíveis na interface de comandos é o seguinte :

ON : realiza a conexão. Neste ponto informa-se o tipo da especificação a ser utilizada (textual ou gráfica). O nome do sinal interno apontado é acrescen-

tado na lista dos sinais internos que possuem um estímulo conectado, a qual será manipulada pelo cardápio SHOW.

OFF : anula um comando INJECT através da manipulação do cardápio SHOW. O nome do sinal interno pertencente ao comando anulado é retirado da lista dos sinais que possuem um estímulo conectado.

OFF ALL : anula todos os comandos INJECT realizados. A lista dos sinais que possuem um estímulo conectado torna-se vazia.

7.3.2 Force

O comando FORCE fixa (grampeia) um valor em um sinal do modelo de simulação que está sendo simulado. Este valor possui prioridade mais alta do que qualquer valor atribuído a este sinal por um comando INJECT ou pela própria simulação. Este valor somente pode ser mudado por outro comando FORCE ou através de seu cancelamento. Uma consistência deve ser realizada para verificar se o valor informado é compatível com o tipo de dado do sinal indicado. Este comando é particularmente útil na simulação de falhas, correspondendo aos modelos de falhas "stuck-at".

Se o comando FORCE for aplicado a um sinal que esteja conectado com um estímulo por um comando INJECT, a conexão será interrompida mas a forma de onda continua avançando ao longo do tempo. Isto significa que se um comando FORCE foi executado no tempo 10, quando este comando for cancelado no tempo 25, o comando INJECT será novamente conectado com o valor da forma de onda do tempo 25 e não do tempo 10 quando este havia sido interrompido.

Este comando pode ser especificado tanto textual como graficamente. Para a especificação textual, a sintaxe do comando é a seguinte:

FORCE < force_comando >

< force_comando > ::= < nome_sinal > = < valor >

Exemplo : FORCE sn = 0

Na especificação gráfica, a representação gráfica da agência.alternativa.versão é exibida na área de trabalho para que um sinal seja apontado. Após, informa-se o valor que será atribuído ao sinal apontado. Para o comando FORCE o significado das opções que estão disponíveis na interface de comandos é o seguinte :

ON : realiza o comando. Informa-se o tipo da especificação a ser realizada (textual ou gráfica). Após executado o comando, o nome do sinal apontado é acrescentado na lista dos sinais que estão grampeados, a qual será manipulada pelo cardápio SHOW.

OFF : cancela o comando FORCE através da manipulação do cardápio SHOW. O nome do sinal pertencente ao comando cancelado é retirado da lista dos sinais que estão grampeados.

OFF ALL : cancela todos os comandos FORCE. A lista dos sinais que estão grampeados torna-se vazia.

7.3.3 Short

O comando SHORT permite que se simule uma condição de curto circuito entre dois sinais do modelo de simulação que está sendo simulado, sendo também destinado à simulação de falhas. Uma consistência é realizada para verificar se os dois sinais apontados possuem o mesmo tipo de dado e se estão no mesmo nível de hierarquia da árvore geradora do modelo de simulação. Este co-

mando pode ser especificado tanto textual como graficamente. Para a especificação textual, a sintaxe do comando é a seguinte:

SHORT < short_comando >

< short_comando > ::= < nome_sinal_1 > , < nome_sinal_2 >

Exemplo : SHORT sn1,sn2

Na especificação gráfica, a representação gráfica da agência.alternativa.versão é exibida para que os dois sinais envolvidos sejam apontados. Para o comando SHORT o significado das opções que estão disponíveis na interface de comandos é o seguinte :

ON : realiza o comando. Informa-se o tipo da especificação a ser utilizada (textual ou gráfica). Os nomes dos sinais indicados são acrescentados na lista dos sinais que estão em curto circuito. Cada elemento desta lista é composto por um par de sinais.

OFF : cancela um comando SHORT através da manipulação do cardápio SHOW. O par de sinais pertencente ao comando SHORT cancelado é retirado da lista dos sinais que estão em curto circuito.

OFF ALL : cancela todos os comandos SHORT existentes. A lista dos sinais que estão em curto circuito torna-se vazia.

7.3.4 Put

O comando PUT atribui um valor constante a um sinal do modelo de simulação que está sendo simulado. Este valor será perdido tão logo a simulação o

mude. Se para um determinado sinal já existir atribuído um valor através de um comando INJECT ou de um comando FORCE, um comando PUT neste sinal cria uma situação de erro fazendo com que uma mensagem seja enviada ao usuário. Uma consistência deve ser realizada para verificar se o valor informado é compatível com o tipo de dado do sinal indicado.

Este comando pode ser especificado tanto textual como graficamente. Para a especificação textual, a sintaxe do comando é a seguinte:

PUT < put_comando >

< put_comando > ::= < nome_sinal > = < valor >

Exemplo : PUT s1 = 13

Na especificação gráfica, a representação gráfica da agência.alternativa.versão é exibida na área de trabalho para que um sinal seja apontado. Após o apontamento, informa-se o valor que será atribuído ao sinal.

7.3.5 Read

O comando READ exibe o valor de um sinal pertencente ao modelo de simulação que está sendo simulado. Se um comando INJECT, FORCE ou PUT estiver atribuindo um valor a este sinal, esta informação também é fornecida. Este comando pode ser especificado tanto textual como graficamente. Para a especificação textual, a sintaxe do comando é a seguinte:

READ < read_comando >

< read_comando > ::= < nome_sinal >

Exemplo : READ sin_in

Na especificação gráfica, aponta-se um sinal pertencente a agência.alternativa.versão cuja representação gráfica é exibida na área de trabalho.

7.3.6 Trace

O comando TRACE permite a monitoração de sinais pertencentes ao modelo de simulação que está sendo simulado. A monitoração dos sinais pode ser realizada durante o processo de simulação ou resultar na criação de um objeto RE na base de dados para uma futura análise. Os sinais monitorados são exibidos como formas de onda [JOH 88]. A indicação do sinal a ser monitorado pode ser especificada tanto textual como graficamente. Para a especificação textual, a sintaxe é a seguinte:

TRACE < trace_comando >

< trace_comando > ::= < nome_sinal > < tipo_trace >

< tipo_trace > ::= SCREEN | DISK | BOTH

Exemplos : TRACE rst DISK
TRACE ckp SCREEN

Na especificação gráfica, aponta-se para o sinal que se quer monitorar na representação gráfica da agência.alternativa.versão exibida na área de trabalho. Para o comando TRACE o significado das opções que estão disponíveis na interface de comandos é o seguinte :

ON SCREEN : realiza a monitoração do sinal apontado no modelo de simulação somente no vídeo e durante o processo de simulação. O nome do sinal apontado é acrescentado na lista dos sinais que estão sendo monitorados juntamente com um atributo indicando que a monitoração é realizada no vídeo.

ON DISK : armazena na base de dados o resultado da simulação para o sinal apontado para uma futura análise. O nome do sinal apontado é acrescentado na lista dos sinais que estão sendo monitorados juntamente com um atributo indicando que a monitoração é armazenada na base de dados.

ON BOTH : realiza a monitoração do sinal apontado tanto no vídeo como na base de dados. O nome do sinal apontado é acrescentado na lista dos sinais que estão sendo monitorados juntamente com um atributo indicando que a monitoração é realizada tanto no vídeo como na base de dados.

OFF : cancela um comando TRACE através da manipulação do cardápio SHOW. O nome do sinal pertencente ao comando cancelado é retirado da lista dos sinais que estão sendo monitorados.

OFF ALL : cancela todos os comandos TRACE realizados. A lista dos sinais que estão sendo monitorados torna-se vazia.

OFF ALL SCREEN : cancela a monitoração de todos os sinais que estão sendo monitorados no vídeo. Os nomes dos sinais cancelados são retirados da lista dos sinais que estão sendo monitorados.

OFF ALL DISK : cancela a monitoração de todos os sinais cuja monitoração é armazenada na base de dados. Os nomes dos sinais cancelados são retirados da lista dos sinais que estão sendo monitorados.

7.3.7 Run

O comando RUN inicia um processo de simulação. O controle do ambiente é transferido para o simulador que será ativado. A simulação continua por um determinado número de unidades de tempo ou até um instante de tempo especificado ou ainda até que uma condição de parada ("break-point") seja satisfeita. A sintaxe deste comando é a seguinte:

RUN < run_comando >

< run_comando > ::= < for > | < up_to > | < forever > |
< step >

< for > ::= FOR < valor >

< up_to > ::= UP_TO < valor >

< forever > ::= FOREVER

< step > ::= STEP < valor >

Exemplos : RUN FOR 10
RUN UP_TO 150

Para este comando o significado das opções que estão disponíveis na interface de comandos é o seguinte :

FOR : realiza o processo de simulação por um determinado número de unidades de tempo.

UP_TO : realiza o processo de simulação até que a simulação atinja o instante de tempo especificado. Caso este instante de tempo seja inferior ao atual tempo de simulação, o comando ativado é ignorado.

FOREVER : realiza o processo de simulação até que uma condição de parada seja satisfeita. Um valor "default" para o número de unidades de tempo simuladas ininterruptamente é estabelecido pelo ambiente caso nenhuma condição de parada seja satisfeita.

STEP : permite que se defina um passo (duração da simulação) em unidades de tempo. Após a definição do passo, este comando fica habilitado até o momento em que uma nova opção do comando RUN seja selecionada. Existe uma tecla especial para executar o comando RUN STEP com o passo definido. Caso esta tecla seja ativada sem que o passo esteja definido nada ocorre.

7.3.8 Break

O comando BREAK permite especificar condições de parada para o processo de simulação retornando o controle do ambiente para o usuário. Estas condições de parada ("break- points") devem ser avaliadas a cada tempo da simulação. Parênteses podem ser incluídos na expressão a ser avaliada. Este comando pode ser especificado tanto textual como graficamente. Para ambas as formas de especificação deve ser realizada uma consistência para verificar a compatibilidade

do tipo do sinal apontado com o seu valor indicado. A sintaxe deste comando é a seguinte:

BREAK < expressão >

< expressão > ::=

< nome > < op_relac > < valor > < lista_expressão >

< nome > ::= < nome_sinal > |

T (variável de tempo)

< op_relac > ::= < | <= | > | >= | = | -=

< lista_expressão > ::= < op_log > < expressão > |

NADA

< op_log > ::= OR | NOR | AND | NAND

Exemplos : BREAK sin1 = 10 OR dt_in = 5

BREAK T = 10 AND (sn > 15 OR s2 = 1)

Na especificação gráfica, a representação gráfica da agência.alternativa.versão é exibida para que um sinal seja apontado. Existe uma tecla especial que indica que a expressão envolverá a variável de tempo. Após a indicação do nome (seja nome de um sinal ou da variável de tempo), menus horizontais com os operadores relacionais e os lógicos são exibidos para que um seja selecionado. Deste modo, a expressão pode ser construída interativamente.

Para o comando BREAK o significado das opções que estão disponíveis na linguagem de comandos é o seguinte :

ON : constrói uma expressão de "break-point". Informa-se o tipo da especificação a ser utilizada (textual ou gráfica). Através de uma tecla especial

indica-se se esta expressão será a edição de uma já existente ou a criação de uma nova. Na edição de uma expressão, uma é selecionada pela manipulação do cardápio SHOW. A nova expressão criada é acrescentada na lista de expressões de "break-point".

OFF : cancela uma expressão de "break-point" através da manipulação do cardápio SHOW. A expressão cancelada é retirada da lista de expressões de "break- point".

OFF ALL : cancela todas as expressões de "break- point" existentes fazendo com que a sua respectiva lista torne-se vazia.

Cada linha do cardápio SHOW representa uma expressão de "break-point". Na avaliação das expressões pelo simulador, é avaliado também o 'OU' lógico existente entre todas as expressões de "break-point".

7.3.9 Batch

O comando BATCH permite a criação de um objeto BAT na base de dados proveniente do histórico de comandos da sessão de simulação do modelo de simulação que está sendo simulado. Para este comando o significado das opções que estão disponíveis na linguagem de comandos é o seguinte:

START : indica o instante de tempo inicial a partir do qual um objeto BAT será construído. A partir deste ponto todos os comandos executados serão armazenados neste objeto BAT.

STOP : indica o instante de tempo final do objeto BAT. Informa-se o nome do objeto BAT a ser armazenado na base de dados. Uma consistência é realizada para verificar se o nome do objeto BAT criado é diferente dos demais nomes dos objetos BAT relacionados à agência.

RUN : permite executar sequencialmente os comandos contidos em um objeto BAT armazenado na base de dados a partir da unidade de tempo atual da simulação. Uma lista com todos os objetos BAT relacionados à agência é exibida para que um seja selecionado. Se por algum motivo um dos comandos deste objeto BAT selecionado não puder ser executado (for incompatível com o modelo de simulação que está sendo simulado) um aviso é enviado pelo ambiente. Neste ponto, o usuário é questionado se ele deseja continuar ou não com este objeto BAT.

Caso seja executado o comando EXIT (encerramento da sessão de simulação) sem que o objeto BAT esteja concluído, um aviso é enviado pelo ambiente para verificar se este objeto BAT será cancelado ou se será criado na base de dados com este instante final.

7.3.10 Save

O comando SAVE permite salvar o estado corrente da sessão de simulação do modelo de simulação que está sendo simulado. Cria um estado intermediário, objeto ST, na base de dados. Este objeto ST contém o valor de todos os sinais do modelo de simulação e as listas dos elementos pertencentes aos comandos que estavam ativos. Para este comando o significado das opções que estão disponíveis na interface de comandos é o seguinte:

ON : realiza o comando SAVE. Informa-se o nome do objeto ST a ser criado. Uma consistência é realizada para verificar se o nome do objeto ST criado é diferente dos demais nomes dos objetos ST relacionados ao modelo de simulação agência.alternativa.versão.modelo.

OFF : remove da base de dados um objeto ST. A lista de nomes dos objetos ST que estão relacionados ao modelo de simulação agência.alternativa.versão.modelo é exibida para que um seja selecionado.

OFF ALL : remove da base de dados todos os objetos ST que estão relacionados ao modelo de simulação agência.alternativa.versão.modelo.

7.3.11 Back

O comando BACK permite que se retorne o estado da simulação a um tempo anterior previamente salvo. É realizada uma busca de um objeto ST armazenado na base de dados. Através da manipulação do cardápio SHOW seleciona-se o objeto ST desejado. O objeto ST é representado através de seu nome junto com a unidade de tempo no qual foi criado.

Quando é realizada a volta a um tempo anterior todos os sinais do modelo de simulação recebem o valor que possuíam neste tempo, a lista dos elementos dos comandos que estavam ativos é recuperada e o histórico dos comandos posterior a este tempo é cancelado. Uma consistência é realizada para verificar se o objeto ST selecionado foi criado em um tempo anterior ao tempo atual da simulação.

7.3.12 Time

O comando TIME permite que se informe o tempo de CPU (tempo do relógio real consumido pelo processo de simulação) gasto entre dois instantes de tempo de simulação. Este comando serve para coleta de dados visando medir a eficiência dos simuladores. A sintaxe do comando TIME é a seguinte:

TIME < time_comando >

< time_comando > ::= FIRST < valor > |
LAST < valor >

Exemplo : TIME FIRST 10

Para este comando o significado das opções que estão disponíveis na interface de comandos é o seguinte :

FIRST : indica o instante de tempo de simulação inicial a partir do qual será medido o tempo de CPU. Uma consistência é realizada para verificar se este tempo informado é maior ou igual ao tempo atual da simulação.

LAST : indica o instante de tempo de simulação final até o qual será medido o tempo de CPU. Uma consistência é realizada para verificar se este tempo é maior do que o tempo indicado na opção FIRST.

7.3.13 Exit

O comando EXIT encerra a sessão de simulação retornando à interface principal do ambiente de simulação. As seguintes opções estão disponíveis para este comando:

LEAVE : abandona a sessão de simulação sem salvar o seu estado. Uma confirmação deve ser feita pelo usuário.

SAVE : salva a sessão de simulação criando o objeto SS na base de dados.

8 MÓDULO DE ANÁLISE DE RESULTADOS

8.1 Introdução

O módulo de análise de resultados (MAR) permite que sejam analisados os resultados de simulações já realizadas. Esta análise é realizada pelo usuário através de recursos de visualização oferecidos pelo MAR [MEY 90]. Para processar esta análise, o usuário deve informar o nome de uma sessão de simulação de um modelo de simulação armazenada na base de dados. Estes dados são fornecidos pelo usuário quando da ativação deste módulo pela opção RESULTADOS, na interface principal do ambiente de simulação.

Quando ativado, o MAR exibe as informações dos sinais monitorados e do histórico dos comandos executados durante a sessão de simulação. Estas informações são apresentadas em duas janelas de visualização na área de trabalho. A figura 8.1 apresenta a interface do MAR. Na área de menu e orientação é informado o nome da sessão de simulação (agência.alternativa.versão.modelo.sessão) que está sendo analisada.

Somente a janela de visualização ativa pode ser manipulada, a qual é representada por um traço duplo em sua borda e com a cor diferente das demais. Como no MAR existem duas janelas de visualização, o usuário pode realizar uma mudança de janela de visualização ativa através de uma tecla especial. Com isto, de acordo com suas necessidades, o usuário pode manipular determinada janela de visualização até encontrar as informações que lhes são interessantes. Deste modo, pode-se visualizar ao mesmo tempo informações em ambas as janelas de visualização.

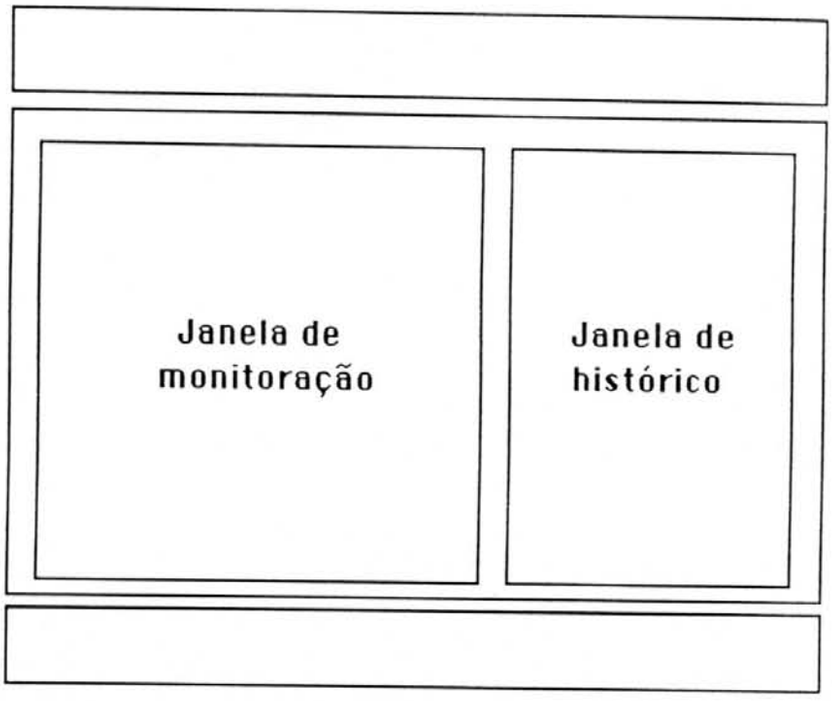

Figura 8.1: Interface da análise de resultados.

Caso a janela de visualização ativa não seja suficientemente grande para exibir as informações nela contidas, o usuário pode realizar uma operação de "scroll". Contudo, mesmo realizando esta operação, o número de informações que podem ser visualizadas ao mesmo instante está restrito ao tamanho da janela de visualização. Para solucionar esta limitação, existe uma tecla especial que permite expandir a janela de visualização ativa. Deste modo, quando a janela de visualização ativa estiver expandida, ela ocupará toda a área de trabalho, se sobrepondo à outra janela de visualização. Mesmo com a janela de visualização ativa expandida, o usuário pode realizar a operação de "scroll". Para restaurar o tamanho original da janela de visualização ativa utiliza-se a mesma tecla especial usada para realizar a expansão.

8.2 Análise da monitoração

A análise da monitoração dos sinais é realizada na janela de monitoração. Nesta janela de visualização são exibidas as formas de onda dos sinais, juntamente com seus nomes, que foram indicados para uma futura análise durante uma sessão de simulação através do comando TRACE DISK ou do comando TRACE BOTH. As formas de onda dos sinais possuem o mesmo formato do que aquelas que são geradas por uma descrição na LGDE (linguagem gráfica de descrição de estímulos).

Para uma melhor orientação ao usuário, é apresentada nesta janela de monitoração uma escala com as unidades de tempo. Esta escala tem seus limites de unidades de tempo restritos ao tamanho da janela de monitoração. Quando uma operação de "scroll" horizontal é realizada, os limites da escala de tempo são alterados, mas o número de unidades de tempo apresentadas na janela de monitoração permanece o mesmo. Isto significa que quando um "scroll" horizontal para a esquerda é realizado, os limites esquerdo e direito da escala de tempo são acrescidos de uma unidade de tempo.

O espaço físico entre os instantes de tempo da escala é fixo. Se as formas de onda dos sinais apresentadas não estão muito claras para o usuário, ele pode alterar a escala para uma melhor visualização. Esta alteração da escala é realizada através de um comando especial onde o usuário informa dois instantes de tempo, um inicial e outro final. Estes dois instantes de tempo serão os novos limites da escala em relação à janela de monitoração. Deste modo, pode-se realizar uma expansão ou uma contração na forma de onda do sinal de acordo com os limites informados. Para se expandir a forma de onda dos sinais aponta-se para dois instantes de tempo existentes na escala de unidades de tempo. Com isto, o espaço físico entre os instantes de tempo da escala será maior que o anterior pois a diferença entre os novos instantes de tempo informados é menor do que a diferença entre os instantes de tempo anteriores. Para se contrair a forma de onda dos sinais

informa-se textualmente os dois novos limites da escala de unidades de tempo de modo que a diferença entre os novos instantes de tempo informados seja maior do que a diferença entre os instantes de tempo anteriores.

A quantidade de formas de onda de sinais é limitada pela janela de monitoração. Caso o usuário deseje visualizar novas formas de onda, ele deve realizar um "scroll" vertical. Como a quantidade de sinais a serem exibidos depende do número de sinais monitorados em uma sessão de simulação, este número pode ser muito grande. Se a operação de "scroll" se tornar muito dispendiosa para o usuário por causa do número de sinais a serem exibidos, existe a possibilidade dele selecionar somente alguns para a visualização. Através de uma tecla especial é apresentada uma lista de todos os sinais que foram monitorados, podendo então o usuário selecionar nesta lista somente os sinais que ele achar mais conveniente de serem visualizados.

8.3 Análise do histórico

Nesta janela de visualização são exibidas as informações contidas no arquivo de histórico pertencente à sessão de simulação informada. Estas informações apresentadas são textuais de modo que somente podem ser realizadas operações de "scroll" sobre esta janela. Cada informação exibida na janela contém o comando que foi executado durante a sessão de simulação, juntamente com o instante de tempo no qual ele foi executado.

Este arquivo de histórico dos comandos é construído na sessão de simulação onde cada comando executado durante esta sessão de simulação é armazenado no arquivo. Este arquivo de histórico dos comandos é armazenado na base de dados juntamente com a sessão de simulação.

9 BANCO DE DADOS

9.1 Interface orientada a objetos

Todas as informações de especificação e de gerência de projeto são armazenadas em uma base de dados unificada. Esta base de dados é acessada pelas ferramentas de projeto (compiladores, editores gráficos e ambiente de simulação) e pelo usuário, através da linguagem de comandos global do sistema, onde um "browser" realiza a navegação sobre os objetos armazenados nela e permite que sejam realizadas consultas e remoções destes objetos. A figura 9.1 apresenta a maneira como as ferramentas de projeto acessam os dados armazenados na base de dados. O acesso é realizado através de uma interface orientada a objetos [BEC 88], onde os objetos são agências, alternativas, versões e objetos relacionados com o ambiente de simulação (modelos de simulação, estados, estímulos, vinculações, sessões de simulação, batchs).

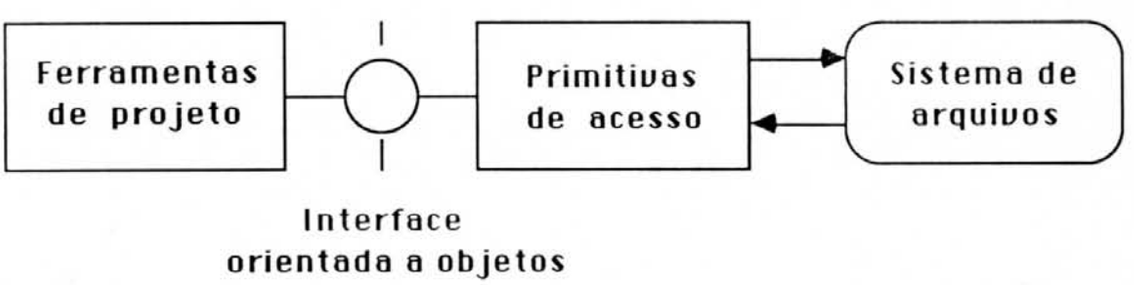

Figura 9.1: Estratégia de implementação da base de dados.

Os objetos possuem atributos como níveis de projeto e nomes. As primitivas de acesso à base de dados permitem a criação, alteração e remoção dos objetos, bem como vários tipos de consultas. Os objetos são representados internamente através de elementos de dados simples (não estruturados), mas manipulados

como objetos complexos através de operadores específicos definidos para cada tipo de objeto. As operações invocadas pelo usuário sobre os objetos são realizadas através das primitivas de acesso à interface orientada a objetos da base de dados.

Alternativas e versões são identificadas por números inteiros gerados pelo sistema no momento de sua criação pelos compiladores ou editores gráficos. Os números de alternativas e versões que foram removidos da base de dados não são reutilizados, de modo que as alternativas e versões mais recentes possuem os números maiores. Já os objetos relacionados com o ambiente de simulação são representados por nomes para a manipulação do usuário e representados internamente por números gerados pelo sistema quando de sua criação.

As consistências necessárias à manutenção das restrições de integridade referentes à composição dos objetos são efetuadas quando da realização de uma criação, alteração ou remoção de objetos na base de dados. Por exemplo, se uma alternativa de uma agência está para ser removida, e ocorrências deste tipo de agência são utilizadas em uma outra agência, a operação não é realizada. Todas as referências para esta alternativa devem ser removidas antes dela ser removida. Também não podem ocorrer alternativas sem a correspondente agência, versões sem alternativas, etc. As restrições de integridade referentes ao ambiente de simulação são discutidas em 9.3.

9.2 Primitivas do ambiente de simulação

Como já foi mencionado, é através das primitivas de manipulação que os objetos, armazenados sob a forma de tuplas simples, podem ser vistos como objetos estruturados. As primitivas disponíveis do ambiente de simulação se dividem em primitivas de recuperação e primitivas de atualização da base de dados. As primitivas de recuperação, por sua vez, se subdividem em primitivas de recuperação

de nomes e de objetos. O anexo A-2 apresenta a descrição de todas as primitivas da base de dados referentes ao ambiente de simulação. A seguir são apresentadas genericamente as funções destas primitivas, sem detalhar os parâmetros necessários para a sua utilização.

a) **Primitivas de recuperação de nomes** : são primitivas de administração de projeto, que informam ao usuário nomes de modelos de simulação, estímulos, etc.

- Informa nomes de modelos de simulação
- Informa nomes de sessões de simulação
- Informa nomes de estímulos
- Informa nomes de vinculações
- Informa estímulos utilizados em vinculações
- Informa as vinculações que utilizam o estímulo
- Informa nomes de estados
- Informa nomes de batchs

Exemplo : Através da primitiva INFORMA_MODELO a base de dados fornece o nome de todos os modelos de simulação para a agência.alternativa.versão que o usuário desejar. Caso esta hierarquia fornecida pelo usuário não estiver correta a primitiva apresentará uma mensagem de erro.

b) **Primitivas de recuperação de objetos** : são primitivas que trazem as informações a respeito dos objetos e garantem que estes objetos como um todo sejam recuperados.

- Busca modelo de simulação
- Busca sessão de simulação
- Busca estímulo
- Busca vinculação
- Busca estado

- Busca resultado
- Busca batch

Exemplo : Com a primitiva BUSCA_SESSAO, todas as informações pertencentes a sessão de simulação são buscadas da base de dados. Para isto, o usuário deve fornecer o nome da sessão de simulação juntamente com quem esta está relacionada, ou seja, a agência.alternativa.versão.modelo. São informados para o usuário a vinculação e o estado inicial utilizados na sessão de simulação, os estados intermediários criados durante a sessão de simulação, os sinais a serem monitorados e todos os comandos executados durante a sessão de simulação.

c) Primitivas de atualização da base de dados : são primitivas que permitem a criação, alteração e remoção dos objetos na base de dados. Tais primitivas são responsáveis pela manutenção das restrições de integridade que são discutidas em 9.3.

- Cria modelo de simulação
- Remove modelo de simulação
- Cria sessão de simulação
- Continua sessão de simulação
- Remove sessão de simulação
- Cria estímulo
- Altera estímulo
- Remove estímulo
- Cria vinculação
- Altera vinculação
- Remove vinculação
- Cria estado
- Remove estado
- Cria batch
- Remove batch

Exemplo : Através da primitiva CRIA_EVMS, o usuário pode armazenar na base de dados uma vinculação realizada. Para isto o usuário deve fornecer a hierarquia na qual esta vinculação estará relacionada na base de dados, ou seja, a agência.alternativa.versão.modelo. A base de dados se encarrega de verificar se a hierarquia informada está correta.

9.3 Restrições de integridade do ambiente de simulação

As restrições de integridade relativas à composição dos objetos são mantidas através das primitivas de atualização da base de dados quando da criação, alteração e remoção de objetos. As restrições de integridade para cada uma das três operações são as seguintes :

a) Criação : as operações de criação de novos objetos na base de dados devem garantir que tuplas com valores nulos ou inválidos para atributos não sejam inseridas, bem como tuplas que façam referência a objetos inexistentes. Isto representa garantir que entre os objetos do ambiente de simulação não existam :

- modelos sem a agência.alternativa.versão correspondente ou inexistente;

- vinculações, estados e sessões de simulação sem modelos de simulação correspondentes ou inexistentes;

- batch sem agência correspondente ou inexistente;

- sessão de simulação sem estado e vinculação correspondente ou inexistente;

- estímulos com agência ou agência.alternativa inexistente;

b) **Remoção** : as operações de remoção devem garantir que o objeto como um todo seja removido da base de dados. Como os objetos podem ser compostos de outros objetos, e assim por diante, deve-se garantir que a remoção será propagada por toda a hierarquia. No caso do ambiente de simulação isto significa que quando um modelo de simulação for removido, todos os estados, vinculações e sessões de simulação relacionados com ele, também devem ser removidos.

As operações de remoção de objetos da base de dados não podem ser efetivadas quando estes objetos estão relacionados com outros objetos. Deste modo, as seguintes consistências devem ser verificadas :

- estímulos que são utilizados em vinculações não podem ser removidos;

- estados e vinculações utilizados em uma sessão de simulação não podem ser removidos.

c) **Alteração** : as operações de alteração só podem ser efetuadas sobre os objetos que não estejam relacionados com outros objetos. As seguintes consistências são verificadas :

- só podem ser alterados os estímulos que não são utilizados em uma vinculação;

- só podem ser alterados os estados e vinculações que não são utilizados em uma sessão de simulação.

10 INTERAÇÃO COM OS SIMULADORES

No sistema AMPLO está disponível uma família de quatro simuladores, três simuladores específicos para cada nível de projeto suportado pelo sistema (modelos nos quais todas as versões primitivas de agências estão descritas em um mesmo nível de projeto) e um simulador multi-nível (modelos nos quais existam versões primitivas descritas em diferentes níveis de projeto). Estas ferramentas permitem a simulação de qualquer modelo gerado a partir das informações armazenadas na base de dados.

Todos os quatro simuladores são construídos segundo uma filosofia mestre-escravo [WAG 89]. O processo mestre é responsável pelo incremento do tempo de simulação, pela gerência das interações de sinais entre agências, o que resulta na chamada do processo escravo específico para o nível em que está descrita a agência, e pela interação com o ambiente. O processo escravo é um simulador orientado a eventos que avalia a função de uma agência num determinado tempo de simulação.

O interpretador da linguagem de comandos é responsável pela análise léxica e sintática dos comandos textuais fornecidos, pelas consistências determinando se um dado sinal existe ou não, etc (ver capítulo 7). Este interpretador passa alguns comandos para a execução por parte do simulador. Outros comandos são executados pelo próprio interpretador, por não implicarem em interação com o simulador. Por exemplo, realizar consultas à base de dados, executar um comando help, etc.

A comunicação do interpretador da linguagem de comandos da sessão de simulação com os simuladores, realizada via supervisor, é processada através de um sistema de mensagens. Para cada comando fornecido durante a sessão de simulação, uma mensagem é adicionada ao conjunto de mensagens que serão enviadas ao simulador. As mensagens enviadas são compostas por um código indicando o

comando executado e a informação específica para o mesmo. Já as mensagens recebidas pelo interpretador da linguagem de comandos da sessão de simulação possuem, além disto, o tempo de simulação em que foi processada esta informação. A cada vez que uma condição de parada é satisfeita, o simulador envia informações para o interpretador da linguagem de comandos da sessão de simulação. Estas serão tratadas pela sessão de simulação de acordo com o tipo do comando a que elas pertencem.

Existem dois tipos de mensagens utilizadas neste sistema de troca de mensagens entre sessão de simulação e simulador : as mensagens enviadas ao simulador que requerem uma resposta imediata e as mensagens normais que são enviadas ao simulador somente quando um comando RUN for executado (ativação do processo de simulação). As mensagens que requeram uma resposta imediata por parte do simulador são correspondentes aos comandos READ (informar o valor do sinal no tempo atual) e SAVE (armazenar o estado atual da simulação na base de dados). Todos os demais comandos da sessão de simulação produzem mensagens normais aos simuladores.

A estrutura interna dos objetos EV é interpretada em tempo real por um módulo do simulador, chamado gerador de estímulos, que se comunica com o ambiente através do sistema de mensagens. O gerador de estímulos acessa, através do objeto EVMS, diversos objetos EV simultaneamente.

As condições de parada são testadas pelo simulador a cada tempo de simulação. Caso uma das condições de parada seja atingida o controle do processo é passado para o usuário na sessão de simulação. Outro ponto importante é o momento em que é realizado o incremento do tempo de simulação. A condição de parada é testada após executadas todas as ações da agência para o tempo atual, sendo função dos novos valores dos sinais já estabelecidos, mas antes do avanço do tempo de simulação.

A simulação se dá pela seguinte sequência de ações:

• o ambiente passa o controle ao mestre enviando as mensagens provenientes dos comandos fornecidos pelo usuário e as mensagens que indicam a existência de eventos;

• o processo mestre chama o gerador de estímulos. Se há eventos para o tempo atual o gerador de estímulos envia estes valores para o mestre;

• o mestre trata os eventos (gerados internamente ou enviados pelo usuário) de acordo com seu algoritmo de simulação (o que causará interações entre agências e ativação dos processos escravos);

• o processo mestre prepara mensagens contendo os valores dos sinais que estão sendo monitorados;

• o processo mestre testa se alguma condição de parada foi atingida. Se foi, mensagens são enviadas ao ambiente, contendo o tempo atual e valores dos sinais monitorados e indicando qual condição de parada foi atingida, e o controle é passado ao usuário na sessão de simulação. Caso contrário, a simulação continua.

• o mestre incrementa o tempo de simulação.

11 CONCLUSÕES

Este trabalho apresentou as principais características do ambiente de simulação do sistema AMPLO. O ambiente de simulação foi definido de acordo com a metodologia que prevê o projeto modular e hierarquizado de sistemas digitais. A integração deste ambiente se dá : através da iteração com o usuário; a nível das estruturas de dados internas compartilhadas com o ambiente de criação e edição de agências cujos objetos gerados são manipulados no ambiente de simulação; através das consultas realizadas pelos mecanismos de navegação provenientes de LAGO; e, finalmente, através do banco de dados orientado aos objetos do ambiente.

O ambiente de simulação foi desenvolvido com a finalidade de se obter um conjunto de recursos gráfico- interativos para auxiliar o usuário no controle do processo de simulação de sistemas digitais. A principal vantagem é de se obter uma interface uniforme com todos os simuladores do sistema AMPLO.

Um aspecto importante no ambiente de simulação é a forma como são gerenciados e controlados os objetos suportados no ambiente. Estes objetos são armazenados em uma base de dados unificada de modo que o usuário tenha total controle e acesso sobre todos os objetos armazenados nela e sobre seus relacionamentos com os demais objetos. Isto permite que o usuário, através de um sistema de consultas à base de dados, possa obter todas as informações necessárias para desenvolver o seu projeto. Sem esta característica, os objetos criados ao longo de um processo de simulação, dentro do ambiente de simulação, seriam armazenados em arquivos isolados, fazendo com que o usuário realizasse ele mesmo um controle sobre os objetos e suas relações, de modo que ele não teria um sistema de consultas eficiente.

O princípio de funcionamento mestre-escravo dos simuladores representa uma vantagem considerável no ciclo de edição / simulação de agências, uma vez

que permite a independência entre a estrutura interna das ocorrências das agências e as estruturas de dados que descrevem a rede. Um modelo de simulação pode ser facilmente editado conforme a metodologia empregada no sistema AMPLO. Isto significa dizer que, uma versão primitiva de uma agência (representa uma folha na árvore geradora do modelo de simulação) pode ser substituída por outra que contenha a mesma interface (isto é, mesma alternativa de projeto) sem que seja necessário alterar qualquer das estruturas de dados presentes no modelo de simulação, necessitando somente compilar a nova versão e alterar o apontador para a versão presente no modelo de simulação. Segundo [MOT 86], no projeto de pequenos sistemas digitais, o tempo de compilação das descrições e modelos pode representar até 30modo, para as fases iniciais de projeto de pequenos sistemas, onde a descrição a ser simulada sofre numerosas alterações, o sistema AMPLO pode ser considerado como bastante adequado.

A quantidade de memória utilizada para armazenar os objetos na base de dados é um ponto muito importante a ser considerado pelo usuário. O usuário deve monitorar somente os sinais realmente necessários, uma vez que estes sinais monitorados são armazenados no objeto RE na base de dados. Mas o problema mais crítico é quanto aos objetos ST (estados iniciais e intermediários). A quantidade de memória utilizada em um estado está diretamente relacionada com o tamanho do circuito que está sendo simulado uma vez que um estado deve conter os valores de todos os sinais do circuito. O problema é que para cada modelo de simulação podem existir diversas sessões de simulação e, para cada sessão de simulação podem existir vários objetos ST, a critério do usuário. Deste modo, deve-se ter muito cuidado na criação desordenada de estados, pois estes são os objetos que consomem mais memória.

O ambiente de simulação encontra-se em fase de implementação. Uma versão inicial deverá estar disponível até o final de 1990, contendo os principais módulos do ambiente e suas características principais de modo que ele esteja operacional. Está em projeto a extensão do banco de dados para a incorporação dos

objetos e relações associados ao ambiente de simulação (as primitivas do banco de dados relativas ao ambiente de simulação já se encontram totalmente especificadas com todos os seus parâmetros e principais características). A linguagem de acesso global (LAGO) do sistema AMPLO está em implementação, assim como a interface com o ambiente de simulação. A filosofia do sistema de troca de mensagens entre o ambiente de simulação e os simuladores já está definido e, atualmente encontram-se em fase final de especificação as estruturas de dados que devem ser acessadas pelos simuladores (estes, por sua vez, encontram-se em fase final de implementação).

Já estão em andamento pesquisas sobre as futuras extensões e aperfeiçoamentos a serem realizados no ambiente de simulação. Estes têm se voltado mais para a parte de recursos de visualização para os níveis KAPA e LAÇO, como por exemplo, em LAÇO, apresentar o grafo de controle colocando, graficamente, as marcas na rede de Petri.

ANEXO A-1 SINTAXE DA LINGUAGEM DE DESCRIÇÃO TEMPORAL DE ESTÍMULOS

LDTE ::= <sinal_clock> |
<outro_sinal>

<sinal_clock> ::= **SIGNAL** <nome_sin_clock> : **CLOCK** ;
BEGIN
<descrição_clock>
END

<outro_sinal> ::= **SIGNAL** <nome_sinal> : <tipo_sinal> ;
BEGIN
<lista_ações>
END

<nome_sin_clock> ::= <ident> <fase>

<fase> ::= [<num_decimal>]

<descrição_clock> ::= <defasagem>
CLOCK_PERIOD <num_decimal>
FOR <num_decimal> : <num_binário> ;
FOR <num_decimal> : <num_binário> ;
END_CLOCK

<defasagem> ::= **PHASE_DELAY** : <num_decimal> ; |
NADA

<nome_sinal> ::= <ident> <dimensão>

<dimensão> ::= [<num_decimal> : <num_decimal>] |
NADA

<tipo_sinal> ::= **BUS** <tipo_valor> |
TERMINAL <tipo_valor> |
VARIABLE <tipo_valor> |
CONTROL

<tipo_valor> ::= (<tipo_sinal_laço>) |
NADA

<tipo_sinal_laço> ::= **INTEGER** |
BOOLEAN

<lista_ações> ::= <ação> <outra_ação>

<outra_ação> ::= <lista_ações> |
NADA

<ação> ::= <instrução_finita> |
<instrução_infinita> |
<período_finito> |
<período_infinito>

<instrução_finita> ::= <instrução> <num_decimal> : <comportamento>

<instrução> ::= **FOR** |
UP_TO

<comportamento> ::= <valor> ; |
BEGIN
<lista_ações>
END

<instrução_infinita> ::= **FOREVER** : <valor> ;

<período_finito> ::= **PERIOD** <tipo_period> <num_decimal>
<ações_tempo_relativo>
END_PERIOD

<tipo_period> ::= **FOR** |
UP_TO |
REPEAT

<período_infinito> ::= **PERIOD FOREVER**
<ações_tempo_relativo>
END_PERIOD

<ações_tempo_relativo> ::= <ação_relativa> <outra_ação_relativa>

<outra_ação_relativa> ::= <ações_tempo_relativo> |
NADA

<ação_relativa> ::= <instrução_relativa> |
<instrução_infinita> |
<período_relativo>

<instução_relativa> ::= **FOR** <num_decimal> : <comp_relativo>

<comp_relativo> ::= <valor> ; |
BEGIN
<ações_tempo_relativo>
END

<período_relativo> ::= **PERIOD** <tempo_relativo>
<ações_tempo_relativo>
END_PERIOD

<tempo_relativo> ::= **FOR** <num_decimal> |
REPEAT <num_decimal> |
FOREVER

<valor> ::= <sinal> <número> |
Z |
<ítem_array> <outro_ítem>

<ítem_array> ::= (<num_binário> , <intensidade>)

<outro_ítem> ::= <ítem_array> <outro_ítem> |
NADA

<sinal> ::= + | - |
NADA

<número> ::= <num_decimal> |
<num_octal> |
<num_hexa> |
<num_binário> <intens_opc>

<intens_opc> ::= <intensidade> |
NADA

<intensidade> ::= **E** | **D** | **W** | **Z**

<num_binário> ::= ' <dígito_binário> <lista_dig_binário>

<lista_dig_binário> ::= <dígito_binário> <lista_dig_binário> |
NADA

<num_octal> ::= " <dígito_octal> <lista_dig_octal>

<lista_dig_octal> ::= <dígito_octal> <lista_dig_octal> |
NADA

<num_decimal> ::= <dígito_decimal> <lista_dig_decimal>

<lista_dig_decimal> ::= <dígito_decimal> <lista_dig_decimal> |
NADA

<num_hexa> ::= # <dígito_hexa> <lista_dig_hexa>

<lista_dig_hexa> ::= <dígito_hexa> <lista_dig_hexa> |
NADA

<dígito_binário> ::= 0 | 1 | X

<dígito_octal> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

<dígito_decimal> ::= <dígito_octal> | 8 | 9

<dígito_hexa> ::= <dígito_decimal> |
a | b | c | d | e | f |
A | B | C | D | E | F

<ident> ::= <letra> <lista_alfanum>

<letra> ::= a..z | A..Z

<lista_alfanum> ::= <alfanum> <lista_alfanum> |
NADA

<alfanum> ::= <letra> |
<dígito_decimal> |
-

ANEXO A-2 DESCRIÇÃO DAS PRIMITIVAS DA BASE DE DADOS

1. Cria_modelo

Cria um modelo de simulação (objeto MS) relacionando-o com a agência.alternativa.versão informada.

2. Busca_modelo

Busca as informações do modelo de simulação que está relacionado com a agência.alternativa.versão informada.

3. Remove_modelo

Remove o modelo de simulação que está relacionado com a agência.alternativa.versão informada.

4. Informa_modelo

Pesquisa os nomes dos modelos de simulação existentes na base de dados que estão relacionados com a agência.alternativa.versão informada.

5. Cria_sessão

Cria uma sessão de simulação (objeto SS) relacionando-a com o modelo de simulação agência.alternativa.versão.modelo informado.

6. Busca_sessão

Busca as informações da sessão de simulação relacionada com o modelo de simulação agência.alternativa.versão.modelo informado.

7. Continua_sessão

Continua a sessão de simulação relacionada com o modelo de simulação agência.alternativa.versão.modelo informado.

8. Remove_sessão

Remove a sessão de simulação relacionada com o modelo de simulação agência.alternativa.versão.modelo informado.

9. Informa_sessão

Pesquisa na base de dados os nomes das sessões de simulação existentes que estão relacionadas com o modelo de simulação agência.alternativa.versão.modelo informado.

10. Cria_obj_estímulo_G

Cria um objeto estímulo gráfico (objeto EV) relacionando-o de acordo com o tipo de relacionamento indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

11. Cria_obj_estímulo_T

Cria um objeto estímulo textual (objeto EV) relacionando-o de acordo com o tipo de relacionamento indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

12. Vincula_obj_estímulo

Vincula o arquivo de informações passadas aos simuladores (arquivo objeto) ao estímulo textual contido no objeto EV informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

13. Busca_estímulo_G

Busca a descrição gráfica do estímulo contido no objeto EV informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

14. Busca_estímulo_T

Busca a descrição textual do estímulo contido no objeto EV informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

15. Altera_obj_estímulo_G

Altera a descrição gráfica do(s) estímulo(s) contido(s) no objeto EV informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

16. Altera_obj_estímulo_T

Altera a descrição textual do(s) estímulo(s) contido(s) no objeto EV informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

17. Remove_obj_estímulo

Remove o objeto EV (gráfico ou textual) informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

18. Informa_obj_estímulo

Pesquisa todos os objetos EV existentes na base de dados informando o seu nome, tipo (textual ou gráfico) e com quem está relacionado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

19. Informa_est_objeto

Pesquisa na base de dados os nomes dos estímulos contidos no objeto EV indicado.

20. Informa_evms_estímulo

Pesquisa na base de dados o nome das vinculações (objeto EVMS) que utilizam o objeto EV informado que está relacionado conforme o indicado (sem relacionamento, relacionado com a agência ou relacionado com a agência.alternativa).

21. Cria_evms

Cria uma vinculação (objeto EVMS) relacionada com o modelo de simulação agência.alternativa.versão.modelo informado.

22. Busca_evms

Busca as informações contidas em uma vinculação que está relacionada com o modelo de simulação agência.alternativa.versão.modelo informado.

23. Altera_evms

Altera a vinculação informada que está relacionada com o modelo de simulação agência.alternativa.versão.modelo indicado.

24. Remove_evms

Remove a vinculação informada que está relacionada com o modelo de simulação agência.alternativa.versão.modelo indicado.

25. Informa_evms

Pesquisa na base de dados os nomes das vinculações existentes que estão relacionadas com o modelo de simulação agência.alternativa.versão.modelo informado.

26. Informa_estímulos_evms

Pesquisa na base de dados os nomes dos objetos EV que estão sendo utilizados em uma vinculação que está relacionada com o modelo de simulação agência.alternativa.versão.modelo informado.

27. Cria_estado

Cria um estado inicial ou intermediário (objeto ST), conforme o tipo indicado, relacionando-o com o modelo de simulação agência.alternativa.versão.modelo informado.

28. Busca_estado

Busca o estado (inicial ou intermediário) que está relacionado com o modelo de simulação agência.alternativa.versão.modelo informado.

29. Remove_estado

Remove o estado (inicial ou intermediário) que está relacionado com o modelo de simulação agência.alternativa.versão.modelo informado.

30. Informa_estado

Pesquisa na base de dados todos os estados que estão relacionados com o modelo de simulação agência.alternativa.versão.modelo indicado, informando o seu nome e tipo (inicial ou intermediário).

31. Busca_resultados

Busca as informações necessárias na sessão de simulação (objetos RE e LOG) que está relacionada com o modelo de simulação agência.alternativa.versão.modelo indicado, para que seja realizada uma análise de resultados.

32. Cria_batch

Cria um arquivo de comandos (objeto BAT) relacionando-o com a agência informada.

33. Busca_batch

Busca o batch que está relacionado com a agência informada.

34. Remove_batch

Remove o batch que está relacionado com a agência informada.

35. Informa_batch

Pesquisa na base de dados os nomes dos batch que estão relacionados com a agência informada.

BIBLIOGRAFIA

[BEC 88] BECKER, K.; GOLENDZINER, L. G. Interface orientada a objetos para um ambiente de CAD de sistemas digitais. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 3., 24-25 mar. 1988, Recife. **Anais**... Recife: UFPe/SBC, 1988. 1v. p.213-226.

[CAR 88] CARROLL, J. M.; AARONSON, A. P. Learning by doing with simulated intelligent help. **Communications of the ACM**, New York, v.31, n.9, p.1064-1079, Sept. 1988.

[FIG 91] FIGUEIRO, J. P. **Extensão das ferramentas PIU/LINUS de especificação e controle de interfaces com o usuário.** Porto Alegre: CPGCC da UFRGS, 1991. 59p. (Relatórios de Pesquisa, 158).

[FOL 82] FOLEY, J. D.; VAN DAM, A. **Fundamentals of interactive computer graphics**. Reading: Addison-Wesley, 1982. 664p.

[FRE 88] FREITAS, C. M. D. S.; GOLENDZINER, L. G.; WAGNER, F. R. O Sistema AMPLO : Um Ambiente Integrado para Projeto de Sistemas Digitais. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 15., 17-22 jul. 1988, Rio de Janeiro. **Anais**... Rio de Janeiro: SBC, 1988. 389p. p.272-284.

[GED 88] GEDYE, D.; KATZ, R. H. Browsing the chip design database. DESIGN AUTOMATION CONFERENCE, 25., June 12-15, 1988, Anaheim. **Proceedings**... New York: ACM, 1988. 730p. p.269-274.

[GRU 89] GRUDIN, J. The case agains user interface consistency. **Communications of the ACM**, New York, v.32, n.10, p.1164-1173, Oct. 1989.

[HAR 89] HARDING, B. System simulation assures that chips play together. **Computer Design**, Tulsa, v.28, n.15, p.70-84, Aug. 1989.

[JOH 88] JOHNSON, M. E.; POORTE, J. B. A hierarchical approach to computer animation in simulation modeling. **Simulation**, San Diego, v.50, n.1, p.30-36, Jan. 1988.

[KAT 87] KATZ, R. H. et al. Design version management. **IEEE Design & Test of Computers**, Los Alamitos, v.4, n.1, p.12-22, Feb. 1987.

[LUZ 91] LUZZARDI, P. R. G. **LAGO** : Linguagem de acesso global ao sistema AMPLO. Porto Alegre: CPGCC da UFRGS, 1991. 147p.

[MEY 90] MEYER, E. VHDL design and analysis system targets concurrent engineering. **Computer Design**, Tulsa, v.29, n.11, p.110, June. 1990.

[MOT 86] MOTT , G. Advanced logic simulators deliver breaknek speed in multiuser environments. **Digital Design**, Boston, v.16, n.5, p.49-53, Apr. 1986.

[OLA 89] OLABARRIAGA, S. D.; COPSTEIN, B.; FREITAS, C. M. D. S. **Ferramentas para especificação e controle da interface com usuário**. Porto Alegre: CPGCC da UFRGS, 1989. 47p.

[PIN 88] PINHO, M. S.; COMBA, J. L. D.; OLABARRIAGA, S. D. **Pacote gráfico para editores do sistema AMPLO**. Porto Alegre: CPGCC da UFRGS, 1988. 57p. (Relatórios de Pesquisa, 102).

[PRI 89] PRINETTO, P. et al. Expressing logical and temporal conditions in simulation environments: TPDL. **Microprocessing and Microprogramming**, Amsterdam, v.26, n.4, p.241-252, Dec. 1989.

[ROS 88] ROSSON, M. B. et al. The designer as user: building requirements for design tools from design practice. **Communications of the ACM**, New York, v.31, n.11, p.1288-1298, Nov. 1988.

[SHA 75] SHANNON, R. E. **Systems simulation : the art and science.** Englewood Cliffs: Prentice-Hall, 1975. 387p.

[SIL 88] SILVA FILHO, J. F.; WAGNER F. R.; LE FAOU, C. **LAÇO** : uma linguagem para descrição de hardware no nível de sistema. Porto Alegre: CPGCC da UFRGS, 1988. 27p. (Relatórios de Pesquisa, 94).

[STU 90] STUMP, H. A designer's guide to simulation models. **Computer Design**, Tulsa, v.29, n.1, p.91-98, Jan. 1990.

[WAG 86a] WAGNER, F. R.; FREITAS, C. M. D. S.; GOLENDZINER, L. G. Equivalência de descrições textuais e gráficas de sistemas digitais num ambiente de CAD. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 13., 19-25 jul. 1986, Recife. **Anais**... Recife: UFPe, 1986. 644p. p.486-493.

[WAG 86b] WAGNER, F. R. A multi-level digital systems simulator based on nets of agencies. In: JSST CONFERENCE ON RECENT ADVANCES IN SIMULATION OF COMPLEX SYSTEMS. July 15-17, 1986, Tokyo. **Proceedings**... Tokyo: JSST, 1986. 676p. p.125-130.

[WAG 87a] WAGNER, F. R.; FREITAS, C. M. D. S.; GOLENDZINER, L. G. **Linguagens de descrição de hardware para suporte a integração do processo de projeto em AMPLO**. Porto Alegre: CPGCC da UFRGS, 1987. 18p. (Relatórios de Pesquisa, 65).

[WAG 87b] WAGNER F. R.; FREITAS, C. M. D. S. **NILO** : uma linguagem para descrição de hardware no nível de portas lógicas. Porto Alegre: CPGCC da UFRGS, 1987. 18p. (Relatórios de Pesquisa, 66).

[WAG 87c] WAGNER, F. R. **KAPA** : uma linguagem de descrição de hardware no nível de transferência entre registradores. Porto Alegre: CPGCC da UFRGS, 1987. (Relatórios de Pesquisa, 68).

[WAG 88] WAGNER, F. R.; FREITAS, C. M. D. S.; GOLENDZINER, L. G. The AMPLO system : An Integrated Environment for Digital Systems Design. In: WORKSHOP ON TOOL INTEGRATION AND DESIGN ENVIRONMENTS, Nov. 26-27, 1987, Paderborn. **Proceedings**... Amsterdam: North-Holland, 1988. p.221-232.

[WAG 89] WAGNER, F. R. Um ambiente integrado para simulação de sistemas digitais. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 4., 12-14 abr. 1989, Rio de Janeiro. **Anais**... Rio de Janeiro: SBC, 1989. 220p. p.74-82.

[WAP 88] WAGNER, P. R. **Editor gráfico REDES**. Porto Alegre: CPGCC da UFRGS, 1988. 60p.

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

Um ambiente integrado para simulação de sistemas digitais

Dissertação apresentada aos Srs.

Profa. Dra. Taisy Silva Weber

Prof. Dr. Flávio Rech Wagner

Profa Dra. Ingrid E.S. Jansch Pôrto

Visto e permitida a impressão

Porto Alegre, 26./..02./.92...

Prof. Dr. Flávio Rech Wagner
Orientador

Prof. Dr. Ricardo Augusto da L. Reis
Coordenador do Curso de Pós-Graduação
em Ciência da Computação