

UM SISTEMA DE EDIÇÃO DE DIAGRAMAS (SED)

Márcio Vinholes Ferreira

Nelson Hamasaki

Walcir Fontanini

Centro Tecnológico para Informática-CTI/IA

Caixa Postal 6162

13081 - Campinas- SP.

ABSTRACT: This paper presents the description of a general diagram editor (SED), a multipurpose tool for graphic edition of diagrams, flowcharts, etc. The proposed facility is aimed at the development of applications involving the problem representation by means of a set of graphical symbols. The SED comprises two modules, the first one creates a symbol library and the second one edits diagrams from that library. At a final stage a diagram description file can be generated to be used by an application program.

1. INTRODUÇÃO

Este artigo apresenta a descrição do SED (Sistema de Edição de Diagramas), uma ferramenta de uso geral para a edição gráfica de diagramas, sinóticos, fluxogramas, etc., que visa dar apoio ao desenvolvimento de aplicações envolvendo a representação do problema por meio de um conjunto de símbolos pictóricos[1].

O SED é parte integrante do projeto CADMAC, um ambiente integrado para otimização e simulação de sistemas discretos e contínuos. O CADMAC terá facilidades para realizar coleta e armazenamento de dados, efetuar análises estatísticas, modelar sistemas graficamente utilizando o SED, simular sistemas discretos e contínuos, apresentar resultados na forma de gráficos e histogramas e gerar relatórios para documentação, além de outras facilidades para otimização. Este projeto está atualmente em desenvolvimento no Instituto de Automação do Centro Tecnológico para Informática.

A seção 2 deste artigo descreve as características principais do sistema, a seção 3 trata da implementação no que se refere a metodologia e estruturas de dados utilizadas, a seção 4 ilustra a aplicação no projeto CADMAC, e finalmente a seção 5 apresenta as conclusões e propostas de ampliação deste trabalho.

2. DESCRIÇÃO DO SED

O SED é composto por dois módulos independentes. O primeiro é responsável pela criação de uma biblioteca de símbolos definida pelo usuário. O segundo pela edição de um diagrama utilizando a biblioteca criada anteriormente. Como resultado final da edição são geradas descrições do diagrama que incluem as características e parâmetros dos símbolos e informações sobre as ligações entre os símbolos. (Figura 2.a).

O usuário cria a biblioteca de símbolos através do Módulo "Editor de Símbolos" (EDSIMB). Para o SED um símbolo é composto por primitivas gráficas e descrição de parâmetros e conexões.

Uma vez definida a biblioteca com a simbologia utilizada, o "Editor de Diagramas" (EDDIAG) permite que sejam criados esquemas ligando esses símbolos.

O EDSIMB, através de uma interface homem-máquina baseada em teclas de funções, permite a elaboração dos símbolos utilizando diversos estilos de linha, primitivas e cores, auxiliado por um Grid de tamanho variável. Também são oferecidas as operações necessárias à manutenção das bibliotecas de símbolos.

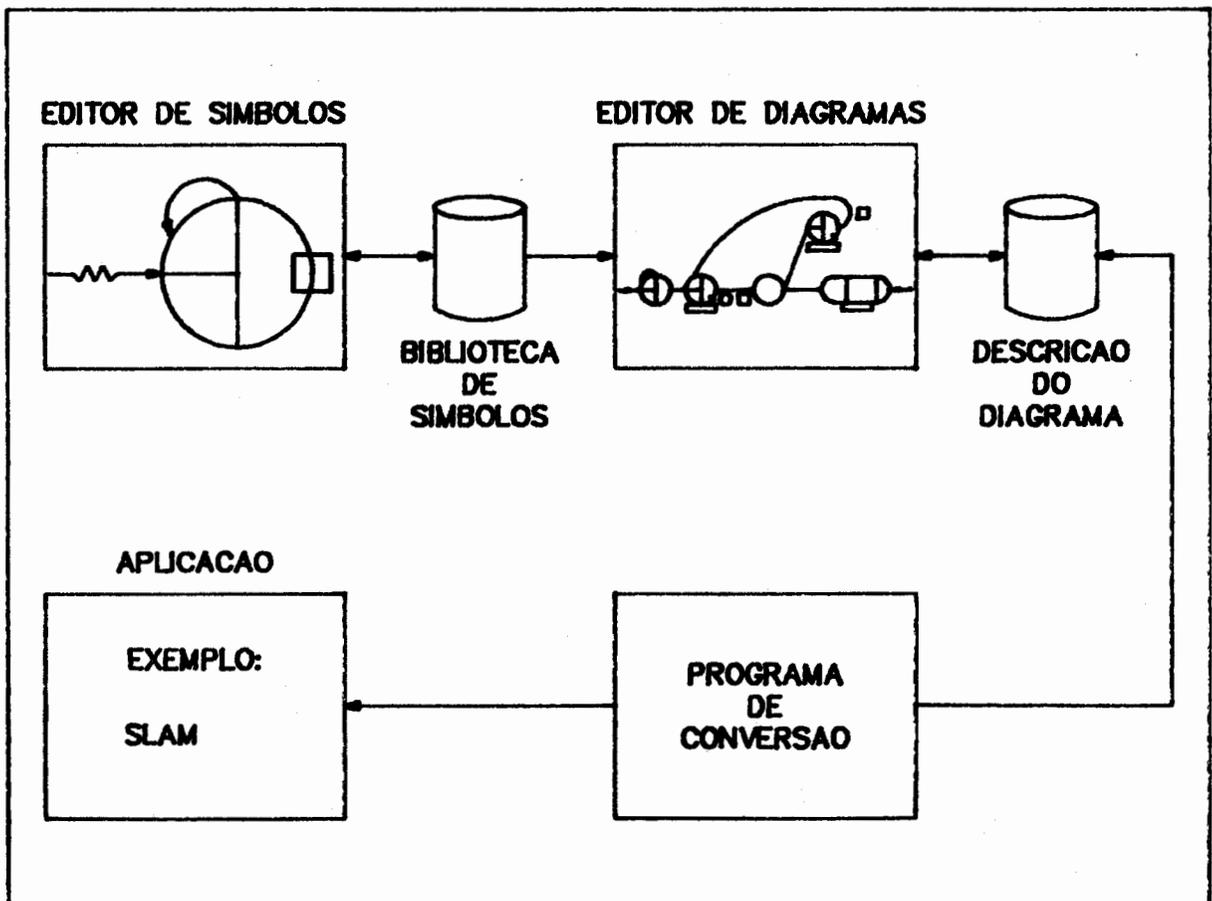


FIGURA 2.a - Estrutura funcional do SED.

O EDDIAG, operando com a mesma filosofia do EDIMB, oferece ao usuário a possibilidade de inserção e deleção de símbolos e conexões selecionados diretamente pelo seu identificador ou através de um menu de ícons. Rotinas que efetuam uma verificação da consistência do diagrama impedem a sobreposição de dois símbolos e o cruzamento de uma conexão sobre um símbolo. Os parâmetros associados aos símbolos e conexões podem ser alterados a qualquer instante durante a edição.

Com o objetivo de detectar conexões não efetuadas e parâmetros não preenchidos é possível realizar um teste de consistência do diagrama gerando um arquivo contendo a descrição destas ocorrências.

Para facilitar a visualização do diagrama, o EDDIAG possui recursos para efetuar Zoom de tamanho variável, centralizar um símbolo na tela e desenhar um Grid baseado no símbolo de maior dimensão.

Finalizada a edição de um diagrama, arquivos de descrição podem ser gerados para que programas desenvolvidos especialmente para cada aplicação possam interpretá-los e assim gerar um resultado final da aplicação.

Os possíveis arquivos gerados pelo EDDIAG são: um arquivo descritivo das instâncias de símbolos, contendo as conexões feitas e os parâmetros preenchidos em cada instância, e um arquivo das conexões realizadas identificando as instâncias de símbolos interligadas e os parâmetros das conexões.

3. IMPLEMENTAÇÃO

3.1. Metodologia de Projeto [2]

O desenvolvimento do SED foi baseado no conceito de Ciclo de Vida do Software, que compreende as seguintes etapas:

- Especificação dos Requisitos: através do planejamento do sistema, gerando como produto um documento textual;
- Especificação Funcional: usando Diagrama de Atividades do SADT (Structured Analysis and Design Technique) e Diagrama de Uso de Dados de Jackson;
- Projeto Básico do Sistema: utilizando Diagrama Hierárquico de Funções (DHF);
- Codificação em linguagem estruturada (Linguagem C);
- Testes: através de "walkthrough" e estudos de casos incluindo contra-exemplos.

Um aspecto considerado importante no projeto de SED foi a portabilidade em relação ao ambiente e a base gráfica utilizada. A possibilidade de utilização do SED em diferentes sistemas (VAX VMS, MS DOS, UNIX, etc.) determinou a escolha da linguagem C [3], que permite a criação de versões em diversos ambientes com pequeno esforço de transporte.

O projeto da estrutura de dados teve como premissa a disponibilidade de uma base gráfica que oferecesse ao sistema rotinas para o traçado de primitivas básicas (retas, círculos, arcos e texto) e para o mapeamento de coordenadas de mundo para dispositivo [1]. Tais recursos são facilmente encontrados em bases e núcleos gráficos disponíveis para a grande maioria de terminais de alta resolução.

As rotinas auxiliares para o traçado do diagrama e manutenção das janelas foram incluídas entre as rotinas gráficas dos editores, procurando, deste modo, manter uma certa independência da base em uso.

3.2. Estruturas de Dados

As informações relativas aos símbolos de uma biblioteca e as ocorrências de símbolos e conexões de um diagrama são armazenadas em estruturas dinâmicas organizadas em forma de listas [4].

A definição do símbolo compreende três partes distintas: uma Seção de Desenho que contém dados gráficos para gerar o símbolo na tela; uma Seção de Parâmetros com especificações dos valores dos parâmetros do símbolo e uma Seção de Conexões com informações para ligação com outros símbolos, conforme ilustra a figura 3.2.a.

A seção de desenho contém primitivas gráficas: segmento de reta especificado por dois pontos, segmento de arco especificado por dois pontos e centro, círculo especificado pelo centro e raio, texto e preenchimento de área.

A seção de parâmetros contém: tipo do parâmetro (inteiro, real, string ou caracter), valor default, visibilidade do identificador do parâmetro para o

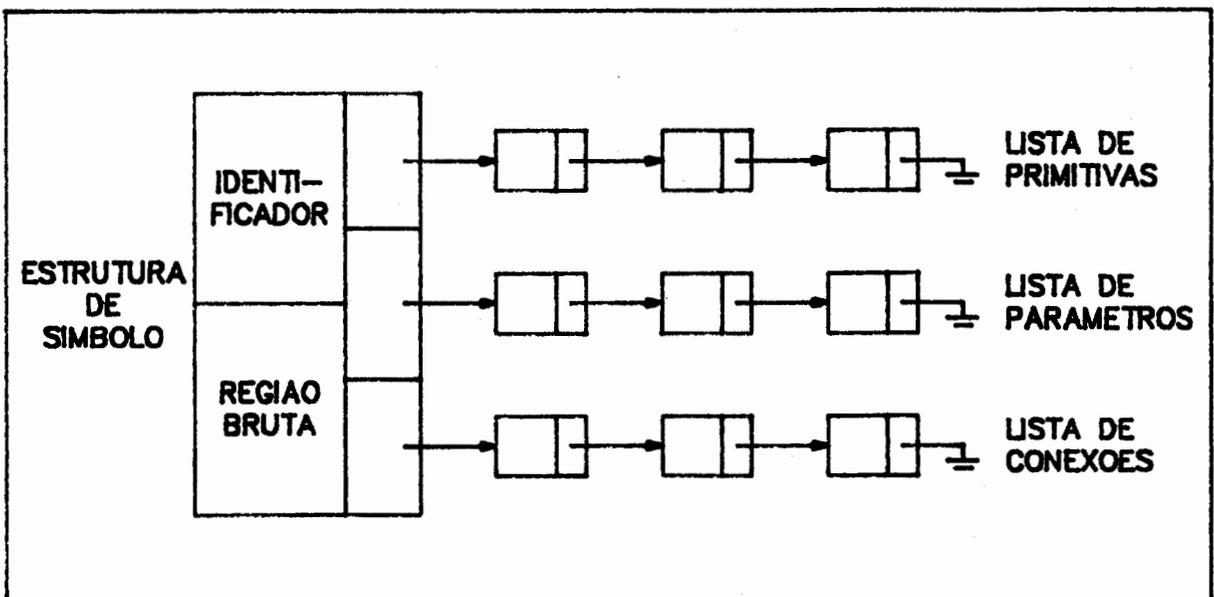


FIGURA 3.2.a - Estrutura de Dados de Símbolo

usuário, região do símbolo para ecoar valor na tela e formato ou máscara para validação do parâmetro.

A seção de conexão contém informações para a construção de regiões retangulares indicando os pontos de conexão do símbolo, seu respectivo identificador, visibilidade ao usuário e tipo de restrição de conexão (entrada, saída e entrada-saída), permitindo editar grafos dirigidos.

Através das primitivas gráficas é calculada a região bruta ocupada por um símbolo, dado utilizado pelo EDDIAG para evitar a sobreposição de dois símbolos ou um de símbolo e uma conexão.

O EDDIAG carrega a biblioteca em memória no início da edição, quando é criada a Lista de Modelos de Símbolos. Cada elemento da lista, chamado Modelo do Símbolo, é idêntico à estrutura de símbolo do EDSIMB.

Através de um arquivo configurador são feitas as descrições dos tipos de conexões que podem ser efetuadas entre dois símbolos, o estilo de linha, cor e aos parâmetros de conexões a serem preenchidos, que dão origem a Lista de Modelos de Conexões.

Durante a edição de um diagrama cada ocorrência de símbolo e conexão é armazenada na Lista de Instâncias de Símbolos e Lista de Instâncias de Conexão, respectivamente. Nestas encontram-se informações necessárias a localização dos símbolos e conexões do diagrama, os parâmetros efetivamente preenchidos, um número identificador de instância e uma referência aos modelos correspondentes. A estrutura completa de um diagrama pode ser observada na figura 3.2.b.

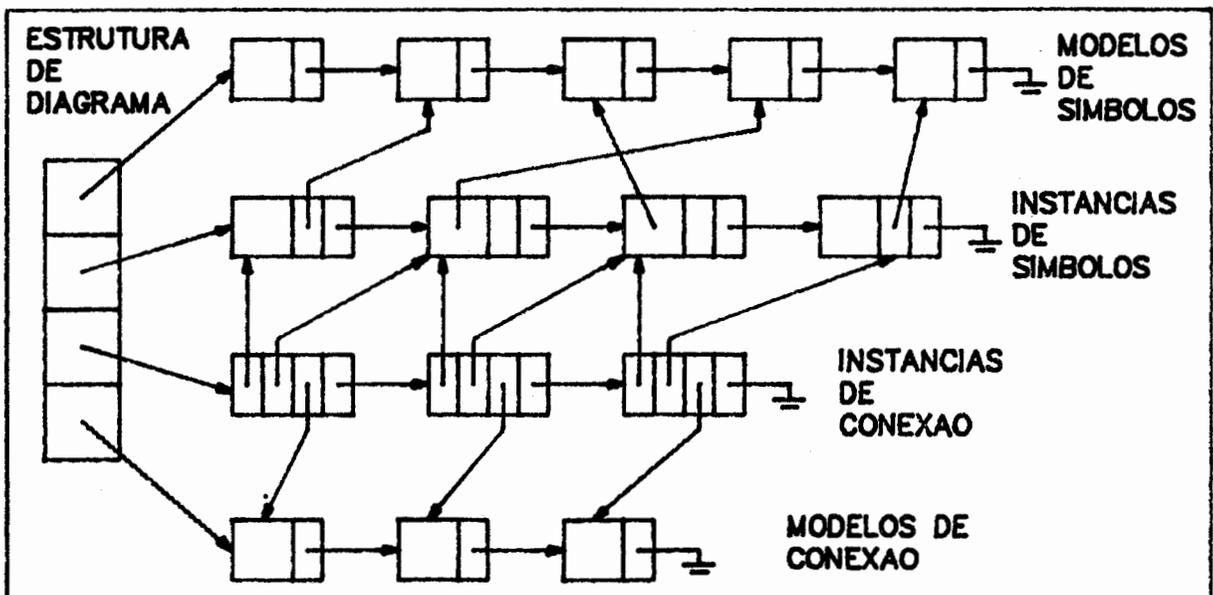


FIGURA 3.2.b - Estrutura de Dados de Diagrama

Os dados comuns a todas as instâncias (descrição gráfica, localização relativa das regiões de conexão, valores default de parâmetros) são armazenadas uma única vez na lista de modelos e compartilhadas através de uma referência à esta lista presente em cada instância.

As informações gráficas dos símbolos encontram-se em forma parametrizada em relação ao centro da região bruta. Podem ser obtidas para cada instância utilizando o seu ponto de referência.

Uma Instância de Conexão, representada por uma sequência de um ou mais segmentos de retas, possui referência aos dois símbolos que estão nas suas extremidades.

4. EXEMPLO DE APLICAÇÃO

Como exemplo de aplicação do SED discute-se a seguir a sua utilização como entrada de dados para o CADMAC.

O CADMAC contém um simulador de sistemas discretos e contínuos. Possui uma linguagem de interação com usuário tanto na forma textual como pictórica (MACSIM) [5]. Pode-se especificar um sistema a ser simulado criando-se um

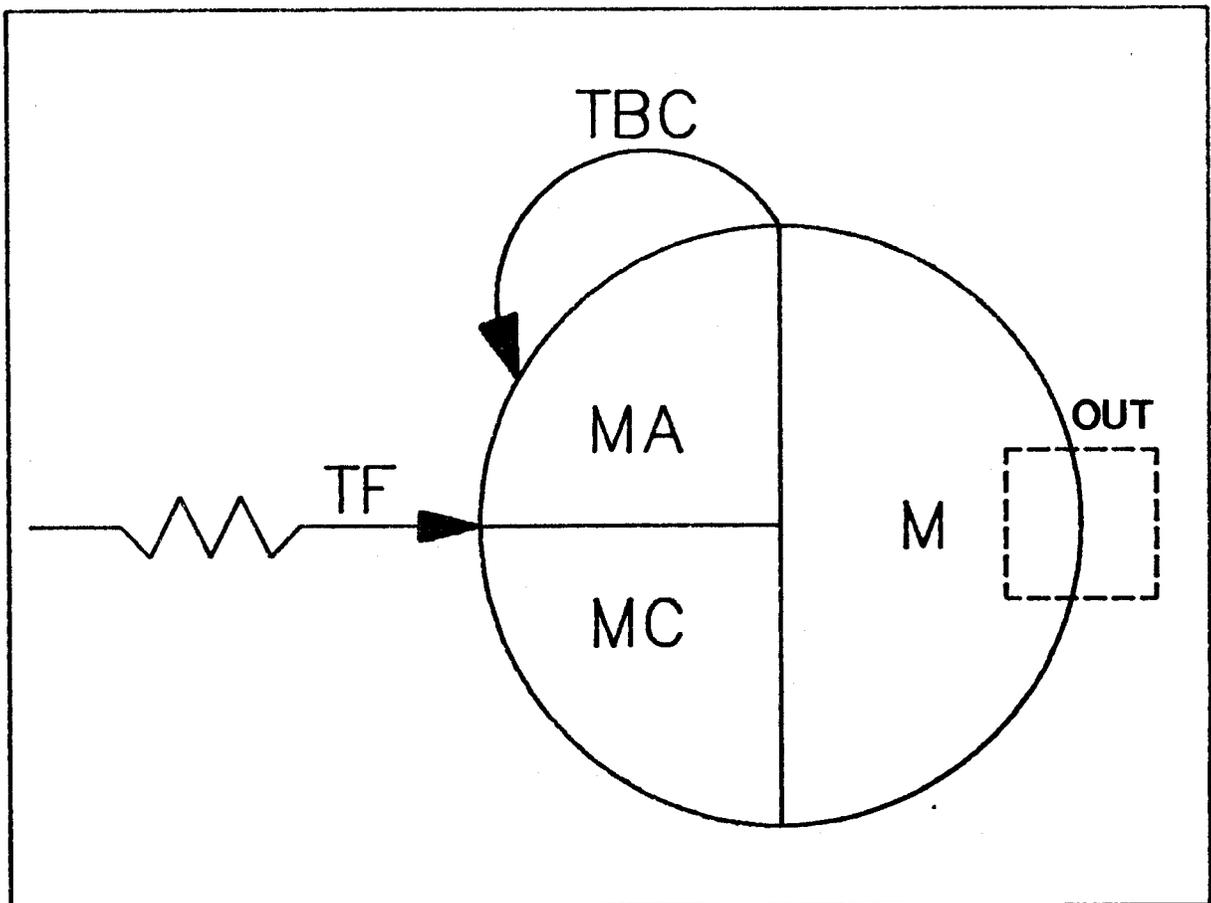


FIGURA 4.a - Símbolo CREATE

programa nesta linguagem textual, ou então elaborar um diagrama com símbolos que representam conceitos de simulação. Esta segunda abordagem é mais confortável para os processos cognitivos do ser humano, pois um desenho aparentemente contém mais informações do que um texto [6].

Alguns símbolos do MACSIM são: CREATE - cria entidades, QUEUE - fila de espera, ACTIVITY - executa uma determinada atividade, COLCT - coleta estatística, GOON - usado para modelar atividades sequenciais, ASSIGN - assinala valores a variáveis de simulação.

A figura 4.a mostra um símbolo CREATE gerado através do EDSIMB. A área retangular com linha pontilhada indica a região de conexão, visível apenas quando for efetuada alguma conexão. O título nesta área é a identificação da conexão.

A figura 4.b é um diagrama completo editado através do EDDIAG contendo vários outros símbolos do MACSIM, representando um modelo para a simulação de uma linha de montagem de televisores, conforme exemplo da página 149 da referência [5].

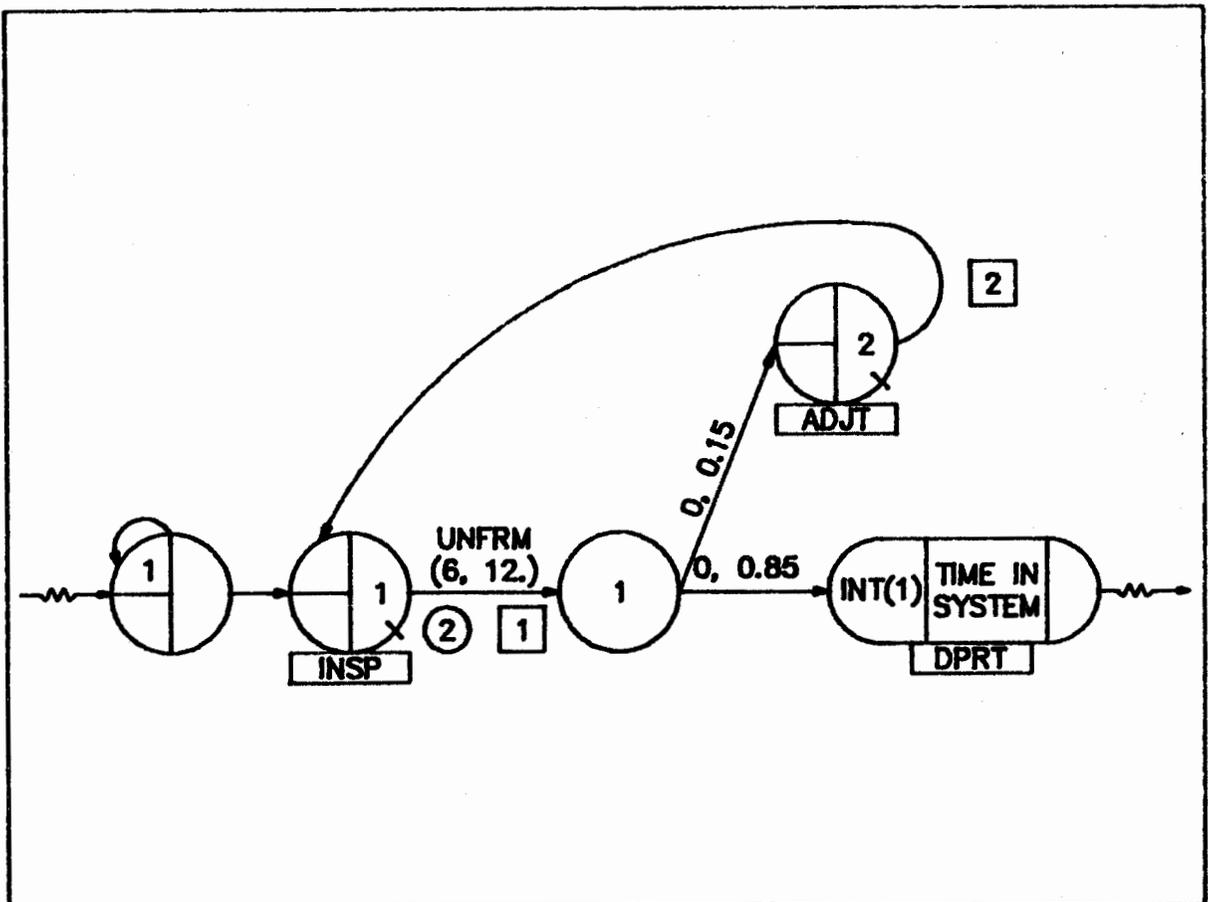


FIGURA 4.b - Linha de Montagem de Televisores

5. CONCLUSÃO

Este artigo apresentou um sistema de edição de diagramas, sinóticos, esquemas, fluxogramas, etc. flexível o bastante para que possa ser configurado para diversas aplicações e equipamentos.

Foram descritas as estruturas de dados envolvidos e sua aplicação ao MACSIM, simulador de sistemas discretos e contínuos.

Como expansão futura deste trabalho, o SED será dotado com a capacidade de manipular diagramas hierárquicos onde os símbolos são representação de outros diagramas.

6. BIBLIOGRAFIA

- 1 - Foley, J.D., van Dam A. Fundamentals of Interactive Computer Graphics
1 ed. Addison-Wesley 1982.
- 2 - Pressman, Roger S. Software Engineering: A practitioner's approach
1 ed. Singapore McGraw-Hill 1982 352p.
- 3 - Kernigham, B.W., Ritchie D.M. The C Programming Language
2 ed. Englewood-Cliffs Prentice-Hall 1978 228p.
- 4 - Wirth, Niklaus Algorithms + Data Structures = Programs
1 ed. USA Prentice-Hall 1976 350p.
- 5 - Pritsker, A. Alan B. Introduction to Simulation and SLAM II
1 ed. (USA) John Wiley & Sons 1986 840p.
- 6 - Galitz, Wilbert O. Handbook of Screen Format Design
2 ed. rev. aum. North-Holland 1985 225p.