

M. ROISENBERG*

L. A. CARVALHO**

SUMARIO

O objetivo desse trabalho é implementar a linguagem GRAFCET como gerenciadora, em um nível mais alto, de um programa tradicionalmente escrito em linguagem de diagrama de relês. Desta forma é descrita a estrutura de dados capaz de representar a linguagem e o processamento efetuado, considerando fatores como velocidade de execução e código gerado, atributos importantes na área de controle de processos em tempo real.

ABSTRACT

The aim of this paper is to describe the implementation of the GRAFCET language as a high level manager of control program traditionally write' with relay ladder diagram language. In this way, we describe the data structure able to represent the language and how the proceedings are doing, considering factors as execution speed and code generation, important qualities in real time control process area.

* Tecnólogo em processamento de dados (UFRGS, 1984) ; mestrando em ciência da computação (PGCC, UFRGS) ; P&D Altus Sistemas de Informática Ltda ; controle de processos, sistemas operacionais.

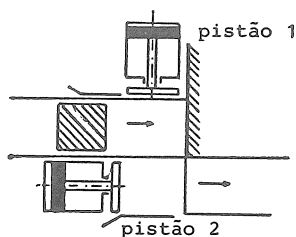
**Engenheiro eletrônico (UFRGS, 1984) ; mestrando em ciência da computação (PGCC, UFRGS) ; P&D Altus Sistemas de Informática Ltda ; controle de processos, microeletrônica.

O GRAFCET (Grafos de Comando Etapa-Transição) é uma ferramenta baseada na teoria de Redes de Petri utilizada para descrição de sistemas lógicos em geral e particularmente de processos de automação industrial. Ela se compõe de três elementos básicos: etapas; transições e arcs. As etapas ou lugares são elementos aos quais estão associadas as ações do processo. Transições são elementos aos quais estão associados eventos. Os Arcos são elementos que ligam lugares a transições e essas a novos lugares. Num grafo, um lugar pode estar ativado ou não em um determinado instante, sendo a ativação representada por uma marca.

Como o GRAFCET em certas estruturas necessárias em controle industrial se mostra pouco eficiente, optou-se por torná-lo o gerenciador das funções de controle, essas, implementadas em linguagem de relê.

Com o intuito de tornar executável a descrição de um processo em GRAFCET procurou-se uma estrutura de dados capaz de representar genericamente e de forma unívoca a linguagem.

O exemplo da figura 1 mostra um processo simples de transferência de peças e sua descrição em GRAFCET.



LUGAR

- 1: espere
- 2: avance pistão 1
- 3: avance pistão 2
- 4: recue pistão 2
- 5: pare pistão 2
- 6: recue pistão 1
- 7: pare pistão 1

TRANSIÇÃO

- 1: peça na frente do pistão 1
- 2: peça na frente do pistão 2
- 3: peça saindo
- 4: pistão 2 recolhido
- 5: pistão 1 recolhido
- 6: "1"

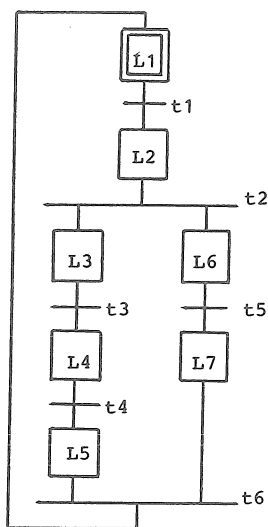


Figura 1. Descrição de um processo em GRAFCET

ARQUITETURA DO SISTEMA

O sistema é composto de um controlador programável AL-1000 fabricado pela Altus Sistemas de Informática, um terminal de programação AL 1800 do mesmo fabricante e um programador GRAFCET implementado num MAXXI da Polimax.

O controlador é o elemento responsável pela execução da lógica de controle, sendo constituído por uma UCP e interfaces de E/S. Esta UCP possui um microprocessador 8031 e 8k de memórias para armazenamento das tabelas do GRAFCET mais 16K para armazenamento das lógicas de relê. No TP AL1800 é feita a entrada edição e montagem da parte referente a lógicas de relê. Essas mesmas tarefas no que se refere ao GRAFCET são efetuadas no microcomputador MAXXI.

Os programadores, MAXXI e AL1800, comunicam-se com o Controlador através de um canal serial a 9600 baud.

ESTRUTURA DE DADOS

Tendo em vista o espaço de memória disponível limitou-se a 100 o número de lugares, sendo permitido o mesmo número de transições.

A estrutura é organizada em 3 tabelas como segue:

- Tabela de lugares (TL) com no máximo 1,5K bytes.
- Tabela de Transições (TT) com no máximo 2,5K bytes.
- Tabela de Lugares Ativos (TLA) com no máximo 15 lugares ativos em determinado instante.

A TL possui três campos de informação: as transições que necessitam deste lugar ativo para ocorrerem, o número de marcas neste lugar e o endereço inicial das lógicas em diagrama de relê que devem ser executadas em virtude do lugar estar ativo. A TT também possui três campos: os lugares que devem possuir marcação para que a transição ocorra, os lugares para onde é transferida a marcação quando a transição ocorrer e os eventos necessários para sua ocorrência. A TLA contém os apontadores que assinalam, dentro da TL, os lugares com marcação e portanto sendo executados. As tabelas que representam o processo visto na fig. 1 estão apresentadas na figura 2.

Além disso convém ressaltar que a parte referente a ação dos lugares, ou seja, aquela implementada em lógicas de relê, é executada diretamente pelo controlador.

ALGORITIMO DE EXECUÇÃO

A execução do algoritmo de controle se constitui em um laço núcleo onde além da atualização dos pontos de entrada e saída do processo controlado é executado o programa de aplicação através da manipulação da estrutura de dados e da ação referente aos lugares. Consultando a TLA, o interpretador do GRAFCET sabe quais são os lugares ativos, buscando em TL o endereço inicial das lógicas que devem ser executadas para cada lugar ativo. O retorno ao interpretador é efetuado ao final da última lógica executada, por uma instrução Retorno. Após isso, são analisadas as transições dependentes daquele lugar ativo. Esta análise consiste na verificação se os lugares de origem daquela transição estão ativos e se os eventos necessários ocorreram. Um evento é o valor lógico que deve ter um relê do sistema para que a transição ocorra. Caso o evento tenha ocorrido, os lugares destino da transição são colocados na TLA e recebem marcas, e o lugar origem, se tiver seu número de marcas igual a zero é retirado da TLA. Caso a transição não tenha ocorrido, então o próximo lugar é executado.

CONCLUSÃO

A linguagem GRAFCET possibilita a programação de sistemas de controle em um nível mais elevado do que os até hoje existentes em controladores programáveis nacionais, pois possibilita a visualização do processo como uma máquina de estados sendo possível a descrição de ações paralelas e sequenciais. Além disso representa uma nova ferramenta, favorecendo a implementação do programa descritor do processo de forma "top-down", e o uso de algoritmos de análise de Redes de Petri, hoje já conhecidos, para detecção de "dead lock", estados atingíveis, etc.

