

Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia Mecânica

Class.
Aut.
Forma

AMBIENTE PARA O PROJETO DE INTERTRAVAMENTOS UTILIZANDO REDES DE PETRI

EDILSON REIS RODRIGUES KATO

Engenheiro Eletricista



Tese apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do Título de Doutor em Engenharia Mecânica.

Orientador: Prof. Dr. Arthur José Vieira Porto

São Carlos
1999

Class.	TESE-EEYC
Cutt.	2882
Tombo	015100

311 00008571

S/S 1072074

AMBIENTE PARA O PROLETO DE
INTERTRAVAMENTOS UTILIZANDO
REDES DE PETRI

EDILSON REIS RODRIGUES KATO

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC/USP

K190 Kato, Edilson Reis Rodrigues
Ambiente para o projeto de intertravamentos
utilizando redes de Petri / Edilson Reis Rodrigues
Kato. -- São Carlos, 1999.

Tese (Doutorado) -- Escola de Engenharia de São
Carlos - Universidade de São Paulo, 1999.

Área: Engenharia Mecânica.

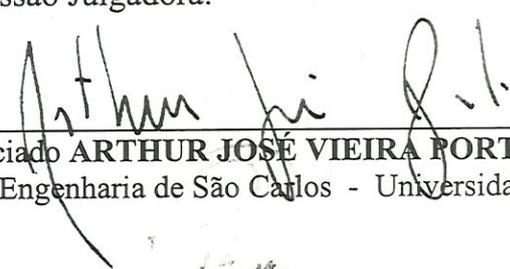
Orientador: Prof. Dr. Arthur José Vieira Porto.

1. Controladores lógico programáveis.
2. rede de Petri. 3. Comando Numérico
computadorizado. 4. CLP. 5. CNC. I. Título.

FOLHA DE APROVAÇÃO

Candidato: Engenheiro **EDILSON REIS RODRIGUES KATO**

Tese defendida e aprovada em 29.11.1999
pela Comissão Julgadora:



Prof. Associado **ARTHUR JOSÉ VIEIRA PORTO (Orientador)**
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Prof. Doutor **JOSÉ GUILHERME SABE**
(Escola de Engenharia de São Carlos - Universidade de São Paulo)



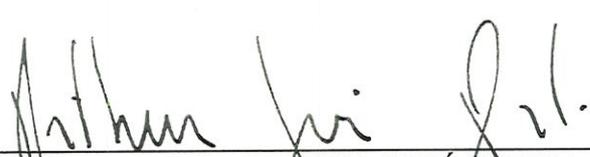
Doutor **RICARDO YASSUSHI INAMASU**
(EMBRAPA)



Prof. Doutor **ANTONIO BATOCCHIO**
(UNICAMP - Universidade Estadual de Campinas)



Prof. Doutor **ALEXANDRE QUEIROZ BRACARENSE**
(UFMG - Universidade Federal de Minas Gerais)



Prof. Associado **ARTHUR JOSÉ VIEIRA PORTO**
Coordenador da Área de Engenharia Mecânica



JOSÉ CARLOS A. CINTRA
Presidente da Comissão de Pós-Graduação

Dedicatória

Aos meus pais,
minha esposa,
meus filhos Felipe
e as recém chegadas
Júlia e Ana Beatriz

Agradecimentos

A Deus pela oportunidade e estímulo nos momentos mais difíceis.

Ao meu orientador, Prof. Dr. Arthur José Vieira Porto, pelo incentivo, amizade e apoio.

Aos amigos do grupo de Simulação, Ana Cristina, Ana Paula, Jandira, Sayuri, José Luiz, Lobão, Mamoru, Marcelo, Mário, Osvaldo Oshiro, Osvaldo Asato, Politano, Renato e Ricardo, pela amizade e valiosa ajuda no desenvolvimento do trabalho.

A todos os professores e funcionários do departamento que contribuíram com o trabalho, com ajuda e amizade, em especial as amigas Ana Paula e Bete da secretaria de pós-graduação.

Aos amigos Roberto Hideaki Tsunaki e Orides Morandin Jr., pelo apoio e incentivos constantes.

A FAPESP, processo número 96/05214-1, pela bolsa concedida.

RESUMO

KATO, Edilson Reis. R. , *Ambiente para o projeto de Intertravamentos utilizando redes de Petri*, Tese (Doutorado) - Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos - SP, Brasil, novembro 1999.

O projeto de intertravamento visa seqüenciar as atividades e eliminar problemas na execução do controle de sistemas de manufatura. Existe grande demanda por técnicas que minimizem ou solucionem as deficiências para o projeto e manutenção de intertravamentos, buscando a padronização dos equipamentos, da linguagem de programação e do processo de implementação. Esse trabalho propõe a criação de um ambiente para o projeto de intertravamentos de máquinas que compõe os sistemas de manufatura. Este ambiente permitirá ao projetista analisar e definir as necessidades de projeto, modelar e editar os intertravamentos e realizar simulações. Para isso, utilizou-se a modelagem em rede de Petri, a qual permite a integração de sistemas de diferentes áreas, tais como sistemas elétricos, pneumáticos e hidráulicos. A simulação utilizando o modelo em rede de Petri do projeto de intertravamento, mostrou-se uma ferramenta efetiva para a análise das seqüências de ciclo do sistema e das alterações que pudessem ocorrer. A análise de desempenho do sistema também mostrou-se útil, para o seqüenciamento correto das atividades e a detecção de assincronismos de funções e seqüências de funcionamento.

Palavras chave: Intertravamento; CLP; Controlador Lógico Programável; Sistemas de Manufatura; redes de Petri.

ABSTRACT

KATO, Edilson Reis R., *Interlocking Design Shell Using Petri Nets*. Tese - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos - SP, Brazil, November 1999.

The aim of interlocking design is to sequence activities and to eliminate problems with automatic system control. There is a great interest in techniques capable of attenuating or solving the deficiencies of the design and maintenance of interlocking system aiming at the standardization of equipment, programming languages and implementation process. This work proposes the creation of a shell for the design machines interlocks of automatic systems (DAS). This shell will allow the analysis and definition of design necessities, interlocking modeling and editing steps, and to run simulations. Petri Net (PN) tools were chosen for the automatic system modeling because they permit the integration of systems of different nature such as electrical, pneumatic and hydraulic. The simulation using Petri Net model of the automatic system showed to be an effective tool to analyze system cycle sequencing as well as possible disturbing alterations. The system performance analysis showed to be useful also for the correct sequencing of activities and the assyncronisms detection of function and functioning sequences.

Keywords: Interlock; PLC; Programmable Logic Controller; Manufacturing Systems; Petri Nets.

ÍNDICE

1 - INTRODUÇÃO	1
1.1 - ORGANIZAÇÃO DO TEXTO	4
2 - REVISÃO BIBLIOGRÁFICA	6
2.1 - COMPONENTES DE SISTEMAS DE MANUFATURA DISCRETA	7
2.1.1 - SISTEMAS DE MANUFATURA.....	8
2.1.2 - SISTEMAS DE MANUSEIO DE MATERIAIS	8
2.1.3 - SISTEMAS DE CONTROLE	9
2.2 - PROJETO DE INTERTRAVAMENTO DE SISTEMAS DE MANUFATURA DISCRETA	11
2.2.1 - EQUIPAMENTO.....	14
2.2.1.1 - CLP (CONTROLADOR LÓGICO PROGRAMÁVEL)	15
2.2.1.2 - CNC (COMANDO NUMÉRICO COMPUTADORIZADO)	21
2.2.2 - TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL	23
2.2.2.1 - SISTEMA ESPECIALISTAS.....	25
2.2.2.2 - APLICAÇÕES DE SISTEMAS ESPECIALISTAS EM SISTEMAS DE MANUFATURA	27
2.2.3 - FERRAMENTAS DE MODELAGEM	30
2.2.3.1 - REDES DE PETRI.....	31
2.2.4 - APLICAÇÃO DE REDES DE PETRI EM SISTEMAS AUTOMÁTICOS	36
2.2.4.1 - PROPOSTAS DE PROGRAMAÇÃO DE SISTEMAS DE MANUFATURA AUTOMÁTICO UTILIZANDO REDES DE PETRI PARA A PROGRAMAÇÃO DE CLPs.....	37
2.2.4.2 - FERRAMENTAS DE SOFTWARE BASEADAS EM REDES DE PETRI	40
2.2.4.3 - EXTENSÕES DE REDES DE PETRI PARA MODELAGEM DE SISTEMAS DE EVENTOS DISCRETOS E APLICAÇÕES EM SISTEMAS DE MANUFATURA	47
2.2.4.4 - ANÁLISE DAS REDES DE PETRI.....	50
3 - PROPOSTA DO AMBIENTE PARA O PROJETO DE INTERTRAVAMENTOS UTILIZANDO A MODELAGEM EM REDES DE PETRI	55
3.1 - ESPECIFICAÇÃO FUNCIONAL.....	55
3.2 - PROPOSTA DE IMPLANTAÇÃO DA SISTEMÁTICA	59

4 - MODELAGEM EM REDES DE PETRI DE SISTEMAS DE INTERTRAVAMENTO	64
4.1 - A ASSOCIAÇÃO DOS MODELOS DE REDES DE PETRI COM A LINGUAGEM DE PROGRAMAÇÃO CLP	65
4.1.1 - LÓGICA E	68
4.1.2 - LÓGICA OU	69
4.1.3 - COMPARADOR	70
4.1.4 - CONTADOR	75
4.1.5 - TEMPORIZADOR	79
4.2 - MODELAGEM EM REDES DE PETRI DE SISTEMAS DE MANUFATURA UTILIZANDO OS ELEMENTOS DA LINGUAGEM CLP	85
4.2.1 - PASSO DE CONTROLE	86
4.2.2 - ELABORAÇÃO DO PASSO DE CONTROLE	87
4.2.3 - PASSO DE FUNCIONAMENTO	88
4.2.4 - ELABORAÇÃO DO PASSO DE FUNCIONAMENTO	89
4.3 - A CONVERSÃO DO MODELO PARA A LINGUAGEM CLP	90
4.4 - EXEMPLO DE MODELAGEM DOS PASSOS DE CONTROLE E DE FUNCIONAMENTO	91
5 - MODELAGEM DE SISTEMAS DE INTERTRAVAMENTO PARA MÁQUINAS CNC	95
5.1 - MODELAGEM DE SISTEMAS DE INTERTRAVAMENTO PARA MÁQUINAS CNC	95
5.1.1 - ESPECIFICAÇÃO DOS ELEMENTOS NECESSÁRIOS PARA O PROJETO DO INTERTRAVAMENTO	98
5.2 - A IMPLEMENTAÇÃO DO "SOFTWARE" DO AMBIENTE PROPOSTO PARA O PROJETO DO INTERTRAVAMENTO DE MÁQUINAS FERRAMENTA CNC	101
5.2.1 - ANÁLISE E DEFINIÇÃO DAS NECESSIDADES	105
5.2.2 - EDIÇÃO DE INTERTRAVAMENTOS UTILIZANDO AS REDES DE PETRI	115
5.2.3 - SIMULAÇÃO DOS INTERTRAVAMENTOS UTILIZANDO AS REDES DE PETRI	117
5.2.4 - ANÁLISE DE DESEMPENHO	122
5.2.5 - IMPLEMENTAÇÃO DA PROGRAMAÇÃO CLP	125
5.2.6 - DOCUMENTAÇÃO DO SISTEMA	128
5.2.7 - IMPLANTAÇÃO E TESTES	130
5.2.8 - MANUTENÇÃO DO SOFTWARE DE INTERTRAVAMENTO	131

6 – VALIDAÇÃO DO AMBIENTE PROPOSTO NA IMPLEMENTAÇÃO DO PROJETO	
DE INTERTRAVAMENTO DE UM TORNO CNC	136
6.1- DESCRIÇÃO DO SISTEMA	136
6.1.1– INTERTRAVAMENTO DE ORGANIZAÇÃO	138
6.1.2– INTERTRAVAMENTO DE INICIALIZAÇÃO.....	141
6.1.3– INTERTRAVAMENTO DE SELEÇÃO DE MODO DE OPERAÇÃO	144
6.1.4– INTERTRAVAMENTO DE EMERGÊNCIA	146
6.1.5– INTERTRAVAMENTO DE COMUNICAÇÃO	147
6.1.6– INTERTRAVAMENTO DE REFERÊNCIAMENTO	148
6.1.7– INTERTRAVAMENTO DE ACIONAMENTO EM GERAL	150
6.1.8– TABELAS DE IDENTIFICAÇÃO DOS ELEMENTOS	152
6.2 - SIMULAÇÃO DO MODELO DO TORNO CNC	156
6.3 – RESULTADOS DA IMPLANTAÇÃO E TESTES DO PROGRAMA CLP GERADO.....	157
7 - CONCLUSÕES.....	158
7.1 - SUGESTÕES PARA TRABALHOS FUTUROS	159
8 - BIBLIOGRAFIA	161
ANEXO A - RELAÇÃO DOS MÓDULOS DO ADIAC	173
ANEXO B – PROGRAMAÇÃO CLP PARA UM TORNO CNC	176

Lista de Figuras:

FIGURA 2.1 - Sistemas funcionais que compõe um Sistema de Manufatura	8
FIGURA 2.2 - Decomposição Hierárquica de um Sistema de Manufatura	11
FIGURA 2.3 - Estrutura do CLP IEC	16
FIGURA 2.4 - Representação em diagrama de contatos	17
FIGURA 2.5 - Representação em DIL (diagrama de blocos lógicos)	17
FIGURA 2.6 - Comparação entre diagrama de blocos e diagrama de contatos	18
FIGURA 2.7 - A representação do GRAFCET	19
FIGURA 2.8 - Quando o desenvolvimento é possível de um S.E.	26
FIGURA 2.9 - Estrutura hierárquica da base de conhecimentos para o sistema proposto	27
FIGURA 2.10 - Exemplo de uma rede de Petri	33
FIGURA 3.1 – Diagrama de blocos do ciclo de vida do projeto de intertravamentos	56
FIGURA 3.2 – ADIAC – Ambiente de desenvolvimento de intertravamento assistido por computador	61
FIGURA 4.1 – Representação da Lógica E (AND)	68
FIGURA 4.2 - Representação da Lógica OU (OR)	69
FIGURA 4.3 – Diagramas de contatos de um comparador (DIN 19 239)	71
FIGURA 4.4 – Comparador de Igualdade em redes de Petri	71
FIGURA 4.5 - Comparador de “maior que” em redes de Petri	73
FIGURA 4.6 – Comparador de “menor que” em redes de Petri	74
FIGURA 4.7 - Comparador de “diferente de” em redes de Petri	74
FIGURA 4.8 – Diagrama de contatos de um contador	76
FIGURA 4.9 - A representação de um contador em rede de Petri	78
FIGURA 4.10 – Um tipo de representação gráfica de um temporizador	79
FIGURA 4.11 – Diagrama de tempo de um temporizador de pulso	80
FIGURA 4.12 - Representação gráfica de um temporizador em rede de Petri de tempo definido	81
FIGURA 4.13 – Repres. gráfica de um temporizador em rede de Petri limitado no tempo	81
FIGURA 4.14 – Repres. gráfica de um temporizador em rede de Petri após desligamento	82
FIGURA 4.15 – Diagramas de tempo de um temporizador de retardo	83
FIGURA 4.16 - Representação gráfica de um temporizador em rede de Petri com retardo na ligação	83
FIGURA 4.17 - Representação gráfica de um temporizador em rede de Petri com retardo no desligamento	84
FIGURA 4.18 – Modelo de controle de uma porta automática	87
FIGURA 4.19 – Programa de controle de uma porta automática em linguagem Ladder	88
FIGURA 4.20 – Modelo de funcionamento de uma porta automática	87
FIGURA 4.21 – Conversão de uma transição para linguagem CLP	91
FIGURA 4.22 – Modelo de uma punctionadeira em rede de Petri e linguagem Ladder	94

FIGURA 5.1 – Modelo adotado para estudo da estrutura de controle do sistema	97
FIGURA 5.2 – Janela de abertura do ambiente de software	102
FIGURA 5.3 – O menu File e a janela da base de dados de informações do projeto	103
FIGURA 5.4 – A janela File Locations	103
FIGURA 5.5 – O menu Support	104
FIGURA 5.6 – Descrição das características do controlador	107
FIGURA 5.7 – Descrição do modo de operação	108
FIGURA 5.8 – Descrição do tipo de comunicação	108
FIGURA 5.9 – Representação da carga e descarga de programas NC	109
FIGURA 5.10 – Representação da ocorrência de falha	109
FIGURA 5.11 – Representação da função de partida quanto a posição dos eixos	110
FIGURA 5.12 – Representação da função de partida quanto a lubrificação temporizada	110
FIGURA 5.13 – Representação da função de partida quanto ao sistema pneumático	111
FIGURA 5.14 – Representação da função de partida quanto ao sistema hidráulico	111
FIGURA 5.15 – Representação do desligamento físico da máquina	112
FIGURA 5.16 – Representação da função do dispositivo de carga e descarga	113
FIGURA 5.17 – Representação da função de remoção de cavacos	113
FIGURA 5.18 – Janela de abertura do editor “Petri Net Tools”	116
FIGURA 5.19 – Janela de atributos do “place”	117
FIGURA 5.20 – Janela de atributos de “transition”	117
FIGURA 5.21 – O menu Analysis do editor “Petri Net Tools”	119
FIGURA 5.22 – Diálogo que fornece o resultado da pesquisa das propriedades de Safeness, Boundedness e Reachability	119
FIGURA 5.23 – Diálogo de entrada para o identificador da transição a ser pesquisada	120
FIGURA 5.24 – Diálogo que descreve o resultado da pesquisa do item “dead” para a transição	120
FIGURA 5.25 – Exemplo de arquivo ASCII da árvore de alcançabilidade	121
FIGURA 5.26 – Exemplo de arquivo ASCII da seqüência de disparo das transições	122
FIGURA 5.27 – Diálogo resultante da análise por matriz de incidência: marca final fornecida é alcançável	122
FIGURA 5.28 – Análise de desempenho de uma rede de Petri	125
FIGURA 5.29 – Tela do programa “Petri Net Tools” com o menu “CLP” acionado	127
FIGURA 5.30 – Árvore de decisão principal de manutenção do software de intertravamento	132
FIGURA 5.31 – Árvore de decisão da falha na inicialização da máquina	133
FIGURA 5.32 – Árvore de decisão da falha da seqüência	133
FIGURA 5.33 – Árvore de decisão da falha elétrica	133
FIGURA 5.34 – Árvore de decisão da falha mecânica	134
FIGURA 6.1 – Visão frontal do torno e da interface homem máquina	137
FIGURA 6.2 – A representação do intertravamento do bloco de organização em redes de Petri	139

FIGURA 6.3 – A representação do intertravamento de partida dos acionamentos e do hidráulico	142
FIGURA 6.4 – Ciclo de lubrificação centralizada	143
FIGURA 6.5 – Representação do intertravamento de seleção do modo de operação	145
FIGURA 6.6 – Rotina de emergência geral em redes de Petri	147
FIGURA 6.7 – Rotina de transmissão/ recepção de status em redes de Petri	148
FIGURA 6.8 – Sinais para o referenciamento	149
FIGURA 6.9 – Bloco de referenciamento de um eixo entre redes de Petri	149
FIGURA 6.9 – Modelagem do acionamento da refrigeração	149
FIGURA A.1 - Relação dos módulos implementados	173
FIGURA B.1 - Bancada de emulação do Torno CNC e teste de programas CLP	183
FIGURA B.2 - Detalhe da placa de Entradas/Saídas (I/O) do Torno CNC.....	183

Lista de Tabelas:

TABELA 2.1 - Condições estabelecidas pela GM para controladores	15
TABELA 2.2 - Lista dos comandos STL utilizados	20
TABELA 4.1 – Descrição dos <i>lugares</i> e das <i>transições</i>	67
TABELA 4.2 – Operações de Comparação	70
TABELA 4.3 – Base de tempo	84
TABELA 4.4 – Ações e estados do sistema	92
TABELA 5.1 – Funções M	99
TABELA 5.2 – Funções H	100
TABELA 5.3 – Alarmes	100
TABELA 5.4 – Mensagens ao operador	101
TABELA 5.5 – Características de controle da máquina CNC	106
TABELA 5.6 – Condições das entradas	130
TABELA 5.7 – Condições dos flags	130
TABELA 5.8 – Características de manutenção de uma máquina CNC	131
TABELA 6.1 – Sinais internos	152
TABELA 6.2 – Lista de transições	153
TABELA 6.3 – Lista de lugares	154
TABELA 6.4 – Sinais externos de entrada	155
TABELA 6.5 – Sinais externos de saída	156
TABELA 6.6 – Condições das entradas	130
TABELA 6.7 – Condições dos flags	130
TABELA 6.8 – Características de manutenção de uma máquina CNC	131

1. INTRODUÇÃO

As recentes alterações no comportamento das empresas, caracterizado pela alta competitividade e busca de novos mercados, vêm impondo-lhes mudanças de uma forma geral.

No caso das indústrias de manufatura, a alta competitividade tem exigido uma melhor qualidade dos produtos, com redução dos custos. Além disso, observa-se uma redução no ciclo de vida de produtos. Em resposta a estas condições, as empresas têm buscado uma maior flexibilidade e integração entre os componentes de seus sistemas de manufatura.

Dentro das técnicas necessárias para se atingir uma maior flexibilidade e integração, está o emprego de tecnologias de automação (MIYAGI, 1996).

Com o aumento da complexidade dos sistemas, devido as novas exigências do mercado, a forma atual de implantação da automação do sistema, composto por relés eletromagnéticos, controladores dedicados, linguagens de programação de baixo nível e pouca integração entre os elementos, passou a ser ineficiente face aos novos ambientes que estão surgindo (por ex.: máquinas de comando numérico computadorizado (CNC) com magazine automático e vários eixos, Sistemas Flexíveis de Manufatura (FMS), etc.) (VENKATESH, ZHOU & CAUDILL, 1994).

Essa complexidade é caracterizada por uma série de fatores, (VENKATESH, ZHOU & CAUDILL, 1994), (VENKATESH, ZHOU & CAUDILL, 1995), (DAVID, 1995), (TAHOLAKIAN & HALES, 1997) :

- grande variação de arquiteturas de sistemas de produção (linhas de transferência, células de manufatura, FMS, etc.);
- tratamento de dispositivos de diferentes naturezas (pneumáticos, hidráulicos, térmicos, etc.);

- grande aumento do número de dispositivos;
- inclusão de “interfaces” e novas interações com o usuário;
- necessidade de variações de algumas características estruturais, dependentes do produto (inclusão ou exclusão de elementos, desde sensores até máquinas);
- necessidade de uma grande quantidade de profissionais envolvidos;
- alto nível de reprogramação do sistema.

Normalmente a implantação da automação do sistema integram vários profissionais de diferentes áreas (elétrico, mecânico, produção, administrativo, etc.). Esses profissionais tratam das informações do projeto utilizando ferramentas diferentes, tais como processadores de textos e editores de projeto específicos (elétrico, pneumático, mecânico, planilhas de custo, planos de processo, etc.) utilizados para realizarem uma especificação funcional de cada etapa de projeto a que eles se referem. O resultado destes estilos diferentes de se executar uma especificação funcional é a necessidade de se ajustar os diferentes documentos (TROY & McQUEEN, 1996).

Além disso, o processador de texto e muitos desses editores de projeto específicos, geralmente não satisfazem todas as necessidades do projeto e na análise da descrição funcional, pois eles não podem auxiliar automaticamente na criação do documento oferecendo verificação de consistência, informações de inspeção, ou simulação da especificação.

A correlação espacial e temporal dos sensores e atuadores, chamada de intertravamento, objetiva garantir a segurança aos operadores, evitar perdas e danos ao sistema e controlar as operações necessárias para que este sistema produza adequadamente (MIYAGI, 1996).

O projeto do intertravamento, isto é, o correto seqüenciamento e a execução de um ou mais conjuntos de ações considerando condições restritivas, visa a correta execução da lógica de controle de um sistema.

Nos controladores digitais, chamados de Controladores Programáveis, a lógica das funções de intertravamentos passou a ser programada e armazenada na sua memória. O controle de equipamentos passa a ser expresso como um programa que processa a lógica dos intertravamentos, o qual trata da partida, do funcionamento, da troca de dados, e de outras funções do sistema de manufatura a ser controlado.

No entanto, esse programa, de uma maneira geral, não utiliza linguagens de programação estruturadas dificultando a análise e validação dos sistemas modelados (VENKATESH, ZHOU & CAUDILL, 1994).

Assim os programas de controle, na maioria desenvolvidos em Diagramas de Lógica de Contatos (“*Ladder*”) ou linguagens a nível de máquina (como “*assembly*”) não possuem flexibilidade suficiente e nem condições favoráveis de manutenção (devido a sua representação ser semelhante ao projeto elétrico (MURATA et al. 1986). Estes diagramas tornam-se tão complexos que localizar a causa de um problema específico se torna uma tarefa difícil.

Assim, existem dificuldades para o projeto de intertravamento de sistemas complexos, dentre as quais pode-se citar:

- falta de ferramentas que possibilitam a obtenção de um único modelo completo de elementos de natureza distintas (operadores, sistemas pneumáticos, sistemas hidráulicos, etc.);
- falta de especialistas para integrar esses elementos com o objetivo de especificar uma descrição funcional para o projeto do intertravamento de uma forma adequada (TROY & McQUEEN, 1996);
- dificuldade de retratar hierarquia e modularidade no modelo (ZAYTOON, 1996);
- dificuldade de manutenção e de reaproveitamento (SILVA & VELILLA, 1982);
- dificuldade na análise, validação e integração entre os elementos do modelo;
- falta de uma sistemática eficiente de projeto.

Portanto existe uma grande demanda de técnicas que minimizem ou solucionem as deficiências descritas acima, buscando uma padronização descritiva dos equipamentos, das linguagens de programação e do processo de implantação do projeto dos intertravamentos (SILVA & VELILLA, 1982), (COURVOUSIER, et al., 1983), (D’SOUZA, 1994), (LEMMER, OBER & SCHENIEDER, 1995), (BAKER, GRANT & SONG, 1996), (TAHOLAKIAN & HALES, 1997).

Nesse contexto, este trabalho propõe “uma sistematização para o projeto do intertravamento de uma máquina que compõe um sistema de manufatura”. Esta sistematização terá por objetivo especificar como o projeto de intertravamento deve ser implementado, fornecendo condições para que os profissionais da área de automação

possam desenvolver, implementar e manter o controle do intertravamento de máquinas que utilizam controladores lógico programáveis (CLP)”.

Desta forma uma sistemática utilizando um ambiente de projeto de intertravamentos, tendo as redes de Petri como ferramenta de modelagem e simulação do controle é proposta para que os problemas citados anteriormente possam ser minimizados ou eliminados para o projeto de intertravamento de máquinas de um sistema de manufatura.

Esse ambiente de desenvolvimento também deverá realizar a integração entre as partes envolvidas (operadores, sistemas pneumáticos, sistemas hidráulicos, etc.) no projeto do intertravamento, e proporcionar uma infra-estrutura que leva a uma maior padronização de métodos, procedimentos e ferramentas.

A integração será alcançada através de uma abordagem modular na implementação do ambiente, utilizando módulos de definição das necessidades, de simulação, de edição do modelo, de obtenção do código CLP, de documentação, etc. ..., responsáveis pela sistematização da atividade de desenvolvimento do projeto dos intertravamentos.

O resultado final será o de possibilitar a criação de programas de intertravamento de forma sistemática, consistente, padronizada e de fácil manutenção, que possibilitam o desenvolvimento do projeto de intertravamentos minimizando ou eliminando as dificuldades anteriormente citadas.

1.1 – Organização do texto

O texto desse trabalho foi organizado da seguinte maneira:

O capítulo 2 contém uma pesquisa bibliográfica a respeito de Sistemas de Manufatura Discreta, ressaltando os níveis hierárquicos do controle automático e identificando os tipos de intertravamentos nestes níveis. São identificadas as principais ferramentas para a realização dos intertravamentos, tais como o CLP, o Comando Numérico Computadorizado (CNC) e suas ferramentas de modelagem. Uma revisão do estado da arte da utilização das redes de Petri também é realizada, contemplando sua utilização na modelagem e análise de desempenho de Sistemas de Eventos Discretos.

No capítulo 3 é apresentada a sistematização proposta para o projeto dos intertravamentos, que utiliza na sua fase de projeto, a análise e definições das

necessidades, projeto do sistema de controle e projeto e desenvolvimento do “software” de controle. Também é tratada a implementação e sua manutenção.

O capítulo 4 descreve a modelagem dos sistemas de intertravamentos utilizando as redes de Petri. São modelados os elementos básicos de um CLP para a programação dos intertravamentos, tais como contatos, bobinas, contadores, temporizadores, etc...

Neste capítulo também é abordada uma sistemática de como o projetista pode modelar a máquina, visando a compatibilidade, em termos de abstração dos elementos de intertravamentos do CLP modelados em redes de Petri.

O capítulo 5 trata da implementação do ambiente de projeto de intertravamentos proposto no capítulo 3, voltado para a utilização do projeto de máquinas, a fim de se validar a sistematização proposta.

A “shell” projetada é descrita e todos os seus elementos detalhados. Ela é composta por um sistema especialista utilizado para a análise e definição das necessidades, um editor de redes de Petri, um gerador de códigos CLP, um simulador e um sistema de implantação e testes. Além de tratar da análise dos sistemas projetados utilizando as propriedades comportamentais e de análise de desempenho das redes de Petri.

Um exemplo de implementação da metodologia é ilustrada no capítulo 6 e no anexo B para a implementação do projeto de intertravamento em um torno CNC utilizando o ambiente de projeto descrito para o intertravamento de máquinas. No primeiro, são detalhados os blocos e as funções implementadas, assim como os resultados. E no segundo, é registrada a programação CLP gerada.

E por fim, no capítulo 7, são apresentadas as conclusões, a contribuição alcançada e as propostas para trabalhos futuros.

A bibliografia está relacionada no capítulo 8.

O anexo A descreve as relações dos módulos da “shell” do sistema proposto, utilizados no desenvolvimento do ambiente de projeto implementado.

E o anexo B lista a programação CLP para o torno CNC descrito no capítulo 6 em linguagem de programação de lista de instruções e ilustra a bancada montado no Laboratório de Simulação de Sistemas Discretos da Escola de Engenharia de São Carlos da Universidade de São Paulo EESC-USP.

CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA

Os sistemas automáticos nas indústrias manufatureiras tiveram um grande desenvolvimento a partir das novas exigências da produção e da evolução tecnológica dos dispositivos semicondutores (microprocessadores, memórias, etc.).

Estas novas características da produção foram alavancadas pelas exigências dos consumidores, que passaram a buscar uma maior variedade e maior qualidade nos produtos, tornando o sistema de produção voltado para lotes de tamanhos médio e pequeno.

Deste modo foi necessária a evolução do maquinário destinado a este tipo de produção, através dos dispositivos automáticos de alimentação e remoção de peças, além do controle preciso dos eixos da máquina.

A integração das máquinas através de sua troca de dados passou a ser importante para se atingir a alta produtividade, da mesma forma que o rápido preparo ("setup") mecânico da máquina e dos programas NC para a produção.

O Comando Numérico Computadorizado (CNC) passou a fazer parte indispensável destas máquinas e a evolução e o barateamento dos microprocessadores, tornaram-no acessível para serem maciçamente implantados nas máquinas, contribuindo para que as metas da produção fossem atingidas.

Na programação de dispositivos, máquinas e sistemas automatizados, o Controlador Lógico Programável (CLP) tem se mostrado em função de suas características como robustez, confiabilidade e flexibilidade, o dispositivo mais empregado para realizar tais tarefas (TAHOLAKIAN & HALES, 1997).

No CLP são programados os intertravamentos que eram anteriormente executados na forma de ligações físicas (fiação) entre relés para controlar o funcionamento da máquina ou de um sistema de manufatura.

Existem várias definições de intertravamento e são apresentadas as que tratam dos sistemas de manufatura, tais como:

“Intertravamentos nos sistemas de manufatura são sistemas que preservam o bom funcionamento dos equipamentos, a integridade do meio ambiente, e a saúde e integridade física dos operadores” (GRHUN, 1991). Em termos de equipamento, o controle interno da máquina e as condições de seu bom funcionamento é chamado de intertravamento.

“Os Intertravamentos são disposições programadas no sistema, que detectam condições fora da área preestabelecida de trabalho ou seqüências impróprias e que podem abortar a ação em andamento ou então iniciar as ações corretivas necessárias” (DOMBROWSKI et. al. 1990).

“Intertravamentos são condições restritivas, habilitadoras ou inibidoras que determinam a operação ou funcionamento de um sistema ou máquina” (MIYAGI, 1996).

Nos próximos tópicos apresenta-se uma visão geral dos Sistemas de Intertravamento nas indústrias de manufatura discreta, principalmente no uso de equipamentos como controladores programáveis (CLPs). Localizando a atuação dos intertravamentos em cada nível de controle de um sistema de manufatura. Serão tratados o projeto e os elementos responsáveis pela sua execução (ferramentas de modelagem, equipamentos, etc.) e verificados os esforços realizados até o momento para o seu desenvolvimento.

Também apresenta-se o estado da arte do uso da modelagem de sistemas de controle utilizando as redes de Petri.

2.1 - COMPONENTES DE SISTEMAS DE MANUFATURA DISCRETA

Os Sistemas de Manufatura Discretas são os responsáveis pela produção de um determinado produto em uma empresa, incorporando tecnologias de gerenciamento e controle de todos os níveis produtivos, cuja integração pode estar ou não sob controle de uma rede hierárquica de computadores (SHARIT & SALVENDY, 1987).

Três sistemas funcionais descrevem um sistema de manufatura, sendo que a união destes pode gerar um sistema com características de atender necessidades de flexibilidade e automatização e são: o sistema de transformação, o sistema de manuseio

de materiais e o sistema de controle que podem ser vistas na Figura 2.1 (BEL et al. 1988, KUSIAK 1985 e POLITANO 1993).



Figura 2.1 – Sistemas funcionais que compõe um Sistema de Manufatura.

2.1.1 - Sistemas de transformação.

Os Sistemas de transformação são aqueles compostos por equipamentos que realizarão e/ou analisarão as transformações físicas na matéria prima para se atingir o produto final desejado.

Fazem parte destes sistemas:

- Máquinas ferramentas e Robôs industriais;
- Estações de montagem;
- Equipamentos de inspeção e controle de qualidade.

Estes equipamentos devem possuir robustez suficiente para suportar condições de trabalho mais duras, com um grau de utilização bastante elevado. Também podem possuir alto grau de automação e flexibilidade, permitindo a realização de tarefas variadas e longo tempo de funcionamento autônomo (KÜNZLE, 1990).

2.1.2 - Sistemas de manuseio de materiais.

Os sistemas de manuseio de materiais, realizam o transporte e o armazenamento de materiais (peças, "pallets", ferramentas e fixadores), possuindo a função de interligar estações de carga/descarga, manufatura, inspeção e os armazéns.

Os equipamentos que fazem parte deste sistema são:

- a) transporte:
 - i) esteiras;

- ii) empilhadeiras e transportadores manuais;
- iii) AGVs (Veículos Auto Guiado);
- iv) robôs.

b) armazenamento:

- i) armazéns convencionais diversos;
- ii) sistemas de armazenamento e recuperação automático;
(AS/RS - "Automatic Storage/Retrieval Systems");
- iii) "buffers" – pulmões intermediários.

Estes equipamentos fornecem ao sistema uma seletividade de carga, acesso aleatório às posições de armazenamento e flexibilidade de rota.

Um importante integrante deste sistema é o AGV, que se coloca na posição de fornecer ao sistema de manufatura, grande flexibilidade de programação (MORANDIN, 1993), no entanto, a complexidade de seu projeto e a determinação de seu funcionamento no sistema devem ser cuidadosamente levadas em conta, no sentido de se aproveitar tal flexibilidade (MAXWELL & MUCKSTADT, 1982).

2.1.3 - Sistemas de controle.

O sistema de controle realiza o controle de fluxo de cada equipamento, coordena os elementos de transporte e movimentação de materiais, administrando o escalonamento das peças, operações e processos, e armazenando dados internos/externos que auxiliem a tomada de decisões (POLITANO, 1996).

O sistema de controle é o responsável em integrar o Sistema de Manufatura, com o Sistema de Manuseio de Materiais e o meio externo (utilizando tecnologias como por exemplo CIM "Computer Integrated Manufacturing"), além de controlar todos os eventos, relativos às células.

Várias estruturas de controle podem ser encontradas. DESCHANEL (1989) mostra que o tratamento do sistema de planejamento e controle pode ser efetuado de três formas distintas:

- 1 - estrutura centralizada;
- 2 - estrutura distribuída;
- 3 - estrutura hierarquizada.

Das três estruturas, a que melhor dilui a complexidade do sistema é a estrutura hierarquizada, simplificando o processo de decisão, pela redução do número de transmissões de informações de um ponto interno do sistema a outro (KÜNZLE, 1990).

A literatura apresenta várias decomposições hierárquicas, das quais pode-se destacar:

"THE CHARLES STARK DRAPER LAB." (1984), que decompõe o controle de um sistema de manufatura em quatro níveis hierárquicos. O nível planta, que é o maior nível, realiza a integração com o ambiente de organização, com funções de MRP, geração de planos de processos (CAPP) e gerenciamento da base de dados e do sistema de informações.

McLean e Brown (McLEAN & BROWN, 1987), apresentam a estrutura adotada na instalação de pesquisa de manufatura automatizada (*ARMF - "Automated Manufacturing Research Facility"*) da Agência Nacional de Padrões (*NBS - "National Bureau of Standards"*) dos Estados Unidos. A estrutura de gerenciamento é baseada em cinco níveis hierárquicos: instalação ou fábrica, planta, célula, estação de trabalho e equipamento.

Biemans e Vissers (BIEMANS & VISSERS, 1989), definem um modelo de referência para um sistema de planejamento e controle da manufatura (*MPCS - "Manufacturing Planning and Control System"*). A decomposição consiste em nove níveis hierárquicos, onde cada controlador ressalta o trabalho agregado de todos os seus controladores subordinados.

Künzle (KÜNZLE, 1990), descreve uma estrutura hierárquica, com quatro níveis: equipamento, estação, célula e planta, para a especificação do controle de um FMS.

Decomposição hierárquica

Uma decomposição hierárquica funcional pode ser utilizada para melhor identificar o escopo de cada elemento dentro do sistema de manufatura. Um modelo de decomposição proposto (MENENDEZ & KÜNZLE, 1989 e McLEAN & BROWN, 1987) compreende quatro níveis hierárquicos conforme a figura 2.2: Planta, Célula, Estação de trabalho e Equipamento .

Nível Planta: É o nível mais alto que tem por função a coordenação da produção, distribuição das tarefas às células, gerenciamento de atividades associadas

ao projeto de novos produtos, alteração de produtos e a incorporação de alterações tecnológicas à estrutura de produção.

Nível Célula: Onde se localizam as Estações de Trabalho, de carga e descarga e o sistema de manuseio de material.

Nível Estação de Trabalho: é constituída por equipamentos de manufatura (máquina CNC, robô, etc.), equipamentos e dispositivos de transporte, transferência, carga/descarga e armazenamento dedicados exclusivamente à estação.

Nível Equipamento: é constituído pelos equipamentos e dispositivos que executam diretamente uma ou mais operações associadas ao processo produtivo (manufatura, transporte ou armazenamento).

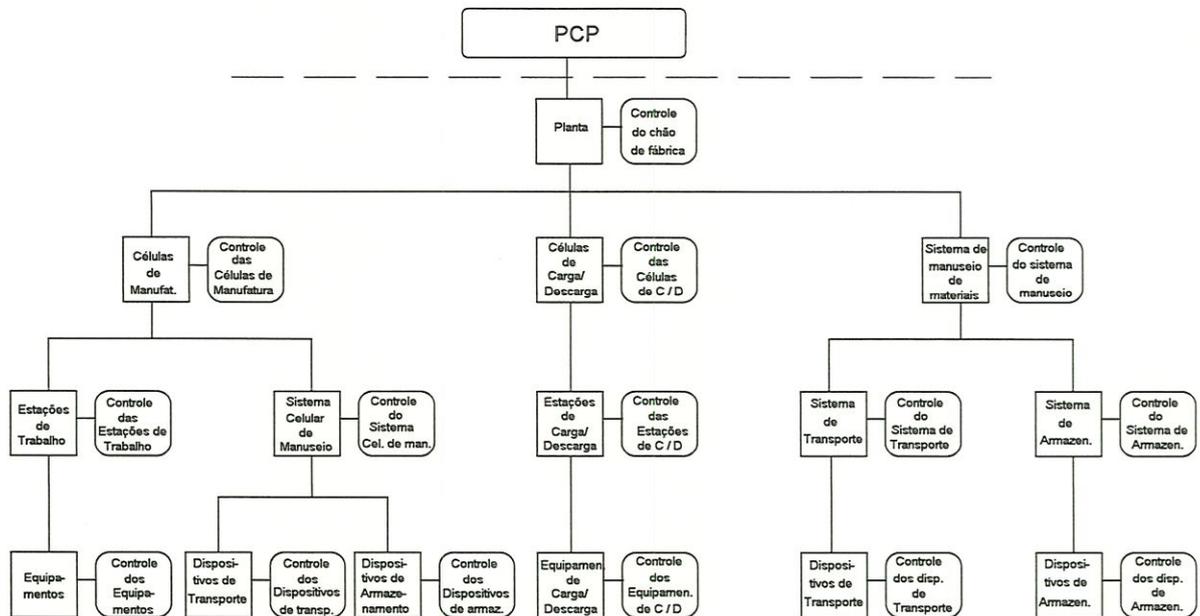


Figura 2.2 - Decomposição Hierárquica de um Sistema de Manufatura(adaptado de KÜNZLE, 1990).

O projeto do controle de intertravamentos de máquinas proposto neste trabalho está situado no nível equipamento, atuando no controle de equipamentos, como mostra a Figura 2.2.

Os intertravamentos nesse nível, irão utilizar o CLP como ferramenta para a sua execução, abordando especificamente sua relação com o CNC e a integração com os outros níveis de controle.

2.2 - PROJETO DO CONTROLE DO INTERTRAVAMENTO DE SISTEMAS DE MANUFATURA DISCRETA

Os intertravamentos são implementados considerando-se as diversas variáveis que possam influenciar o bom funcionamento das instalações, evitando-se condições que possam prejudicar ou interromper o funcionamento do sistema.

Normalmente a complexidade dos sistemas de intertravamento aumenta nos níveis mais altos da decomposição hierárquica descrita anteriormente, implicando em grandes despesas na implantação de um projeto. Desta forma, é importante que se coloque no início do desenvolvimento a questão da real necessidade de implantação, e qual o exato nível de profundidade requerido pelo sistema. É necessário avaliar se a ausência de sistemas de monitoramento e proteção automáticos podem trazer grandes prejuízos a uma empresa referente a indenizações, manutenção, reparos e reposições (LOBÃO, 1995).

Um sistema de intertravamento muito complexo apresenta demanda maior de recursos e aumenta a complexidade de “hardware” e “software”, o que induz a custos mais altos dos testes de validação. Conseqüentemente, ele pode na realidade reduzir a confiabilidade do sistema, desde que algumas falhas apareçam no seu teste e validação (VERMA & CHANDRA, 1991).

Miyagi (MIYAGI, 1996) identifica que os objetivos principais do intertravamento são garantir a segurança, evitar danos aos equipamentos e evitar o encadeamento de acidentes, e divide os intertravamentos em seis grupos:

1) Intertravamento de Partida: São condições que devem ser satisfeitas no instante de partida (inicialização) e que não são consideradas durante o estado de funcionamento normal do sistema;

2) Intertravamento de Funcionamento: São condições que devem ser observadas não somente na inicialização, mas também durante o funcionamento do sistema;

3) Intertravamentos Temporizados: Determina-se um intervalo de tempo entre o funcionamento de cada equipamento;

4) Intertravamento de não Simultaneidade: Evita que certos estados ocorram simultaneamente em diversos equipamentos;

5) Intertravamento de Seqüência: Estabelece as condições de habilitação ou inibição de equipamentos, interligados em série;

6) Intertravamento do processo: Permite a ocorrência de uma ação se todas as ações e condições das etapas anteriores forem completamente executadas e todos os preparativos para a próxima etapa forem satisfeitos.

No que diz respeito ao projeto desses sistemas, encontra-se com mais facilidade roteiros e métodos para sistemas de intertravamento em indústrias de processos contínuo (LEONARDO, 1981 e CRANMER 1991).

Técnicas de análise de risco e operabilidade (HAZOP - do inglês "Hazard e Operability Analysis") (KLETZ, 1985), são abordadas sistematicamente e com prioridade em plantas industriais. Roteiros para os estudos de problemas operacionais que podem ocorrer durante a operação e funcionamento da planta são apresentados por FREEMAN (1989) e KLETZ (1985).

O intertravamento nas indústrias manufatureiras acompanha os níveis de controle, ou seja, está localizado nas estruturas de controle dos Sistemas de Manufatura, integrando os níveis de controle e atuando individualmente em cada componente.

O uso dos intertravamentos aumentou com o advento de novas tecnologias de manufatura. As tecnologias como CIM proporcionaram uma introdução, a nível de fábrica, do termo intertravamento no meio de processos discretos, além dos utilizados em máquinas e estações de trabalho (SZELKE & MESZAROS, 1990). Nessas indústrias, os trabalhos relacionados atingem com maior intensidade esta integração dos níveis de manufatura. Poucos trabalhos detalham a parte de intertravamento dos equipamentos.

Komoda (KOMODA et al. 1984), apresenta um sistema de controle integrado com "workstations" para um FMS. A estrutura de controle é hierarquizada, no entanto possui subsistemas de controle descentralizados. A estrutura de controle é dividida em vários estágios, reagindo às condições locais segundo sua programação (intertravamentos), correspondente àquela função do sistema. Os computadores e as FMCs são conectadas via rede local.

Tashiro (TASHIRO et al. 1985), apresenta um sistema de controle, onde o *software* de controle é baseado em regras (intertravamentos) para a determinação de suas ações, usando uma lógica "IF THEN" quando os eventos ocorrem. As ações de controle selecionadas são então levadas para o sistema controlador.

Campbell (CAMPBELL, 1987), relaciona uma estrutura de controle hierarquizada com sete níveis: corporação, fábrica, chão de fábrica, área de produção, célula, estação

de trabalho e equipamento. Onde a arquitetura proposta é composta de três controladores lógicos: controlador de equipamento, controlador de estações e controlador de célula. Devido à complexidade de serem controlados, as soluções estão no nível de processamento de dados.

Jones e Rabelo (JONES & RABELO 1992), fazem uma abordagem para tomada de decisão em tempo real usando redes neurais, simulação e algoritmos genéricos. Onde a simulação, a nível de planta, prediz o seu impacto na evolução futura do sistema; os algoritmos genéricos são usados para combinar alternativas atrativas em uma única e melhor decisão; e os controladores inteligentes baseados em redes neurais, são responsáveis por intertravar as seqüências e opções de produção.

2.2.1 - Equipamento

Para o desenvolvimento do intertravamento em máquinas é necessário a completa conceituação do ambiente da máquina e dos recursos necessários para atender as especificações de integração (comunicação).

O intertravamento das máquinas, localizado no nível de hierarquia de controle como "equipamento", é responsável pelo bom funcionamento tanto da parte eletrônica e de potência, como dos fatores ligados à produtividade (avanços, velocidades, etc.).

No que diz respeito à máquinas CNC, os sistemas de intertravamento tornaram-se mais sofisticados, devido ao uso de computadores e dispositivos de estado sólido.

Pode-se então separar a atuação da máquina ferramenta de duas formas distintas, como atuando com dois processadores distintos, onde o processador referente ao CNC é o responsável por controlar o posicionamento dos eixos e a interface homem máquina (IHM), e o processador referente ao CLP tem a função de supervisionar o ambiente externo à máquina (como bombas, válvulas, sensores, contatores, etc.).

O intertravamento das máquinas, portanto, está localizado no CLP, e a abrangência de sua atuação é discutida de acordo com o sistema em questão. Isto é, o relacionamento do CLP e do CNC é tão próximo que as atuações são confundidas de acordo com a complexidade das funções programadas e oferecidas ao usuário.

Dividindo-se portanto a máquina em duas partes, a referente ao CNC e a referente ao CLP, pode-se então determinar a atuação de cada uma das partes, entendendo o funcionamento da máquina e realizando a preparação adequada para que ela possa fazer parte integrante de uma célula, pertencente a um Sistema de Manufatura.

2.2.1.1 - CLP (Controlador Lógico Programável)

O CLP (Controlador Lógico Programável) é o mais difundido equipamento eletrônico utilizado para controle de sistemas, seja contínuo ou discreto (TAHOLAKIAN & HALES, 1997). Ele é um equipamento eletrônico que opera digitalmente, utilizando uma memória de programa para armazenamento de instruções do tipo lógicas, aritméticas, de seqüenciamento e de temporização para controle de vários tipos de máquinas ou processos, através de módulos de saída e entrada, digitais e/ou analógicos (GROOVER & ZIMMERS, 1984) além de meio de comunicação com outros equipamentos ou controladores externos.

O CLP pode ser definido como um microcomputador dedicado, que examina periodicamente uma série de sinais provenientes dos sensores de uma instalação, seja ela de campo (indústrias de processo contínuo) ou de uma máquina. Tais sinais são processados e de acordo com o programa das condições de intertravamento, serão ativados sinais a atuadores (relés, lâmpadas, válvulas, etc.).

No final da década de 60 surgiram os primeiros CLPs, baseados em uma especificação técnica de 10 itens divulgada pela General Motors (Estados Unidos), descrita na tabela 2.1.

Tabela 2.1 - Condições estabelecidas pela GM para controladores (MIYAGI, 1996).

Item	descrição
1	Deve ser fácil de ser programado, isto é, as operações seqüenciais devem ser facilmente alteráveis, mesmo na própria planta.
2	Deve ser de fácil manutenção; se possível deveria ser baseado totalmente num conceito "plug in".
3	Deve possuir características operacionais de alta confiabilidade (bem maior que os dispositivos à relés), considerando-se o ambiente industrial.
4	Deve possuir dimensões menores que os painéis à relés para redução do custo do espaço físico.
5	Deve ter capacidade para enviar dados para um sistema central.
6	Deve ter preço competitivo em relação aos atuais dispositivos a relés e/ou eletrônicos.
7	Deve ter capacidade de receber sinais de entrada de 115V CA.
8	Deve ter capacidade de enviar sinais de 115V CA (2A no mínimo, para o acionamento direto de válvulas solenóides, pequenos motores, etc.
9	Deve possibilitar expansões na forma de módulos para atender sistemas de maior porte.
10	Cada unidade deve possibilitar a expansão de, no mínimo, 4000 palavras na memória do programa.

O diagrama da Figura 2.3 mostra a estrutura do CLP IEC (International Electrotechnical Commission) (IEC, 1989) e os tipos de subsistemas que podem ser suportados. Dependendo do fabricante ou da aplicação, pode-se ter múltiplos subsistemas do mesmo tipo em um CLP: vários módulos de memória, vários módulos de E/S, várias unidades de processamento, e assim por diante. Os subsistemas do mesmo tipo são ditos similares. Os subsistemas similares são tratados de forma separada, sendo possível atribuir-lhes nomes para a diferenciação de cada um (AGUIAR, 1992).

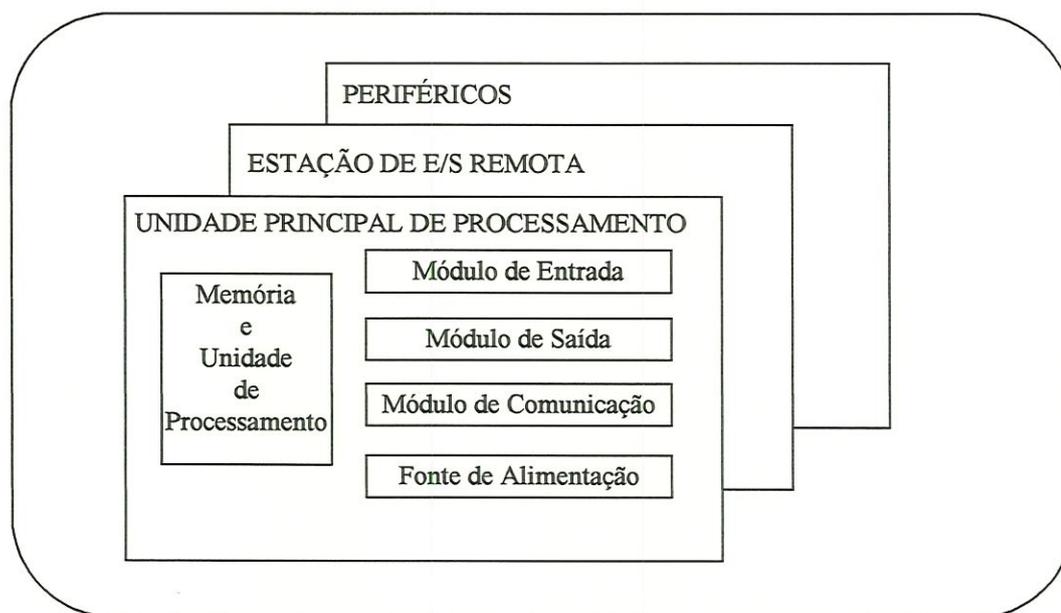


Figura 2.3 - Estrutura do CLP IEC (AGUIAR, 1992).

Vários trabalhos tratam o CLP como um componente crítico de um grande número de processos industriais, tanto na manufatura discreta como nas plantas industriais (BONFATTI et. al., 1997) (TAHOLAKIAN & HALES, 1997) (HENNING, 1999). Sendo evidenciada a preocupação de se tratar adequadamente ou fornecer alternativas mais de alto nível para a sua programação (DURAN & BATOCHIO, 1999) (BONFATTI et. al., 1997), sua especificação em tempo real (HENNING, 1999) e sua conectividade com outros equipamentos (ROHN & RENEHAN, 1998) (MITCHEL, 1998). Além de algoritmos e metodologias para melhorar o processo de desenvolvimento de sua programação e manutenção (BONFATTI et. al., 1997) (MIKULCZYNSKI et. al., 1998).

Estes trabalhos tratam também da aplicação do CLP em sistemas automáticos de manufatura de maior complexidade, como os FMS, onde são criadas metodologias e plataformas para sua implementação, baseadas no controlador CLP (McDERMOTT & YAO, 1997) (McDERMOTT & YAO, 1996).

Programação CLP

Dentre as várias linguagens de programação de alto nível, pode-se destacar abaixo, algumas linguagens utilizadas nos equipamentos de programação de CLPs (HOLLO, 1985):

- i) LADDER ou DIC (Diagrama de Contatos)
- ii) DIL (Diagramas Lógicos)
- iii) GRAFCET (GRAfo Funcional de Comando em Etapa e Transição)
- iv) Statement List (STL) ou LIS (Lista de Instruções)

- LADDER ou diagrama de contatos (MAXITEC, 1989-a).

Neste modo de programação, a representação da programação, utiliza uma simbologia gráfica similar à utilizada nos circuitos elétricos, apresentando os contatos normalmente aberto ou fechado, etc.

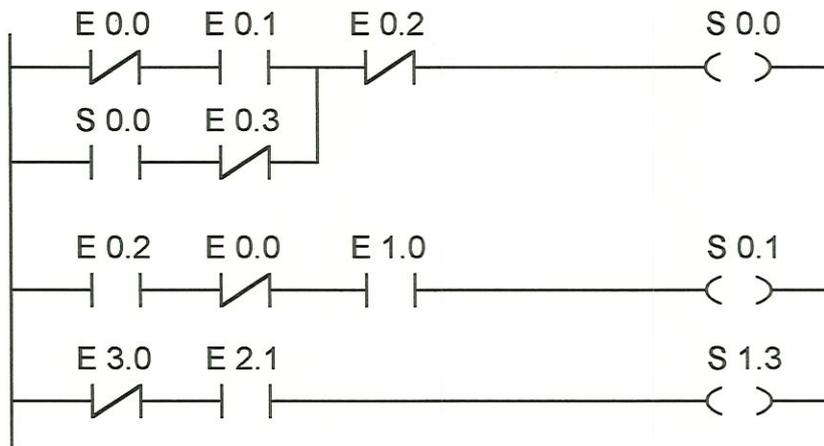


Figura 2.4 - Representação em diagrama de contatos.

A Figura 2.4 ilustra a representação em diagrama de contatos onde E0.0, E0.1, E0.2, E0.3, E3.0 e E2.1 correspondem aos sinais de entrada e S0.0, S0.1 e S1.3 correspondem aos sinais de saída do sistema a ser controlado.

- DIL (diagramas lógicos).

A programação é implementada através de blocos lógicos, da mesma forma que utilizado em eletrônica digital, como mostrado na Figura 2.5.

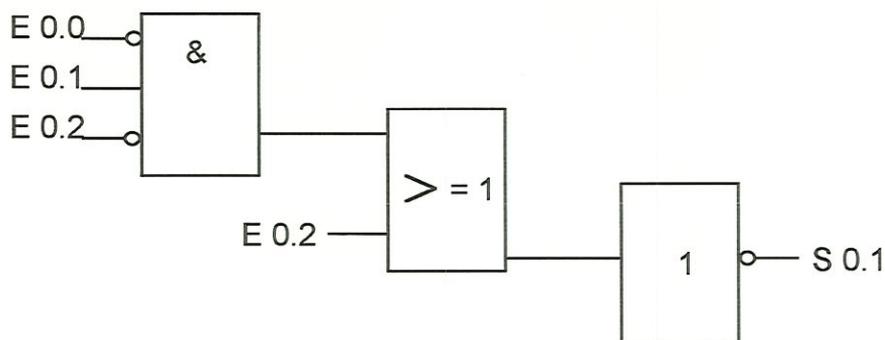


Figura 2.5 - Representação em DIL (diagrama de blocos lógicos) (SIEMENS, 1989).

Na Figura 2.6, ilustra-se a equivalência entre o DIL e a representação LADDER.

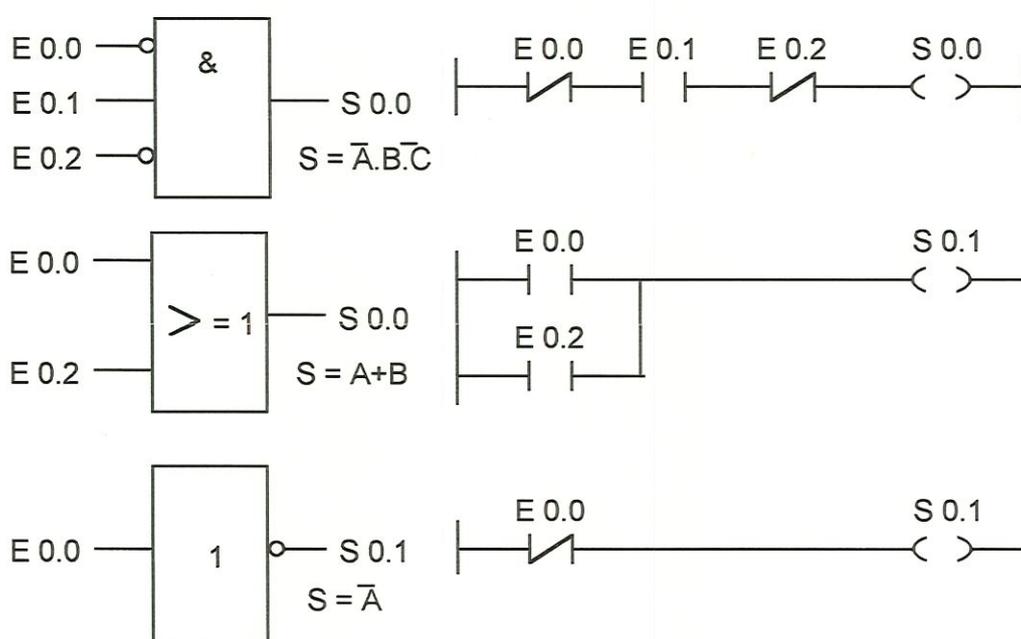


Figura 2.6 - Comparação entre diagrama de blocos e diagrama de contatos.

- O GRAFCET

O GRAFCET é um diagrama funcional, cujo objetivo é de descrever graficamente os diversos comportamentos de um automatismo seqüencial (SLAETS, 1994).

Um sistema a ser automatizado pode ser representado como uma seqüência de passos que se desenvolve sempre na mesma ordem sendo descritos por uma seqüência única que pode possuir várias seqüências alternativas ou simultâneas (paralelas), podendo ser representados como mostrado na Figura 2.7 (TELEMECANIQUE, 1989).

O GRAFCET se compõe de:

- a) Etapa: ao qual é associada uma ação;
- b) Etapa inicial, a qual representa o ponto de partida do ciclo;

- b) Transições as quais são associadas recepções;
- c) Ligações orientadas.

Através da sucessão alternada de etapas e transições, podemos definir uma etapa como a situação de ciclo de funcionamento durante o qual o comportamento do automatismo de comando permanece constante. Qualquer mudança de comportamento leva a outro passo.

REPRESENTAÇÃO DE GRAFCET

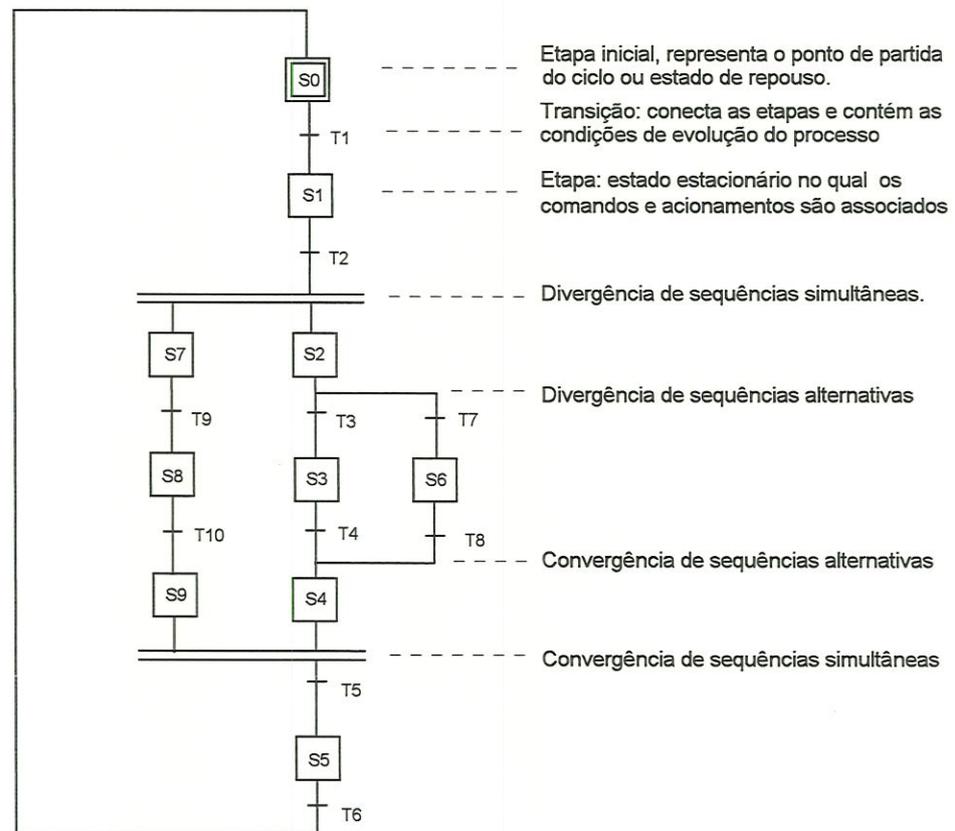


Figura 2.7 - A representação do GRAFCET (TELEMECANIQUE, 1989).

- STL ou lista de instruções (SIEMENS, 1989)

Neste tipo de representação, a programação é realizada através de "mnemônicos" das expressões "booleanas". Para a representação da equação

$S = A.\overline{B}.C + B + \overline{C}$ temos a seguinte listagem:

0000	:A	I	0.0
0001	:AN	I	0.1
0002	:A	I	0.2
0003	:O	I	0.1
0004	:ON	I	0.2

0005 := Q 0.0
 0006 :***

Onde "I" é a representação de uma entrada; "Q" é a representação de uma saída; "F" representa um flag; "T" um Temporizador; e "C" um contador. A listagem dos comandos utilizados e seus parâmetros são relacionados na tabela 2.2:

Tabela 2.2 - Lista dos comandos STL utilizados.

STL	I	Q	F	T	C	Descrição
A	X	X	X	X	X	associação lógica (E)
AN	X	X	X	X	X	associação lógica (E) negada
O	X	X	X	X	X	associação lógica (O)
ON	X	X	X	X	X	associação lógica (OU) negada
A(associação lógica (E) de expressões
O(associação lógica (OU) de expressões
)						fechar parênteses
S	X	X	X		X	acionar com retenção (nível 1)
R	X	X	X	X	X	acionar com retenção (nível 0)
=		X	X			acionar a saída, ou nível lógico
SP				X		inicia temporização com acionamento de saída durante o tempo programado
SE				X		inicia temporização com acionamento de saída durante o tempo programado e a entrada acionada
SR				X		Inicia temporização com acionamento de saída após o tempo programado
SS				X		Inicia temporização com acionamento após o tempo programado e a entrada acionada
SF				X		inicia temporização com desligamento de saída após o tempo programado
CU					X	incrementa contador de 1 unidade
CD					X	decrementa contador de 1 unidade
JU						salto incondicional
BE						fim de bloco
I						incrementa o conteúdo de ... unidades
D						decrementa o conteúdo de ... unidades
L	X	X	X	X	X	carrega o valor de um registrador
T	X	X	X	X	X	transfere o valor de um registrador

Desta forma o CLP será adotado como a ferramenta de execução do controle dos intertravamentos de uma máquina, que junto com o CNC que trata do controle de posição e velocidade dos eixos. E a linguagem utilizada para a metodologia proposta será a Ladder, pois seu uso é unânime entre os fabricantes.

2.2.1.2 - CNC (Comando Numérico Computadorizado)

No final da década de 40 e início da de 50 um consórcio formado por pesquisadores do Laboratório de Servo-mecanismos do Instituto de Tecnologia de Massachussets (M.I.T.) e da Parsons Corporation, com apoio da Força Aérea dos Estados Unidos empreendeu um projeto de adaptação de um comando numérico em uma máquina ferramenta (MACHADO, 1985).

O CNC (Comando Numérico Computadorizado) é um equipamento destinado à automação de máquinas ferramentas e máquinas em geral. Possui um microprocessador responsável pelo posicionamento e velocidade de eixos. Além da função de controle dos eixos, esse microprocessador realiza a interação entre o operador e a máquina (IHM - Interface Homem Máquina).

A IHM consiste de uma apresentação no vídeo de várias telas, selecionáveis de acordo com as necessidades do operador, apresentadas como (MAXITEC, 1989-b):

- modo de edição;
- modo de entrada de dados;
- modo de saída de dados;
- modo de trabalhos manuais (JOG);
- modo de trabalhos em automático;
- modo de referenciamento;
- modo de ensinamento (TEACH-IN).

e muitos outros, dependendo do fabricante, do usuário e do grau de sofisticação da máquina, dispostos normalmente na forma de menus e submenus.

O CNC possui grande versatilidade na alteração de dados e trocas de programas, além de agilizar a preparação rápida da máquina.

Ele possui dispositivos de integração com computadores e controladores, como por exemplo interfaces seriais, equipamentos DNC (Controle Numérico Direto), tornando viável sua integração em ambientes automatizados, como células flexíveis e auxiliando no planejamento e controle da produção (MARCOS & ORIVE, 1996).

O CNC realiza, em seu ambiente, o processamento do posicionamento de eixos, sejam estes independentes ou dependentes, realizando assim interpolação entre tais eixos. Estas interpolações podem ser lineares, circulares, helicoidais, etc., dependendo das necessidades do processo de produção.

Porto (PORTO, 1985) ressalta algumas vantagens da utilização de comandos numéricos e seus principais recursos:

- maior versatilidade devido a recursos como trocadores de ferramentas, interpolação de eixos e controle da velocidade de usinagem;
- redução do tempo de "set-up" da máquina;
- repetibilidade (responsável pela qualidade das peças);
- possibilidade de mudanças rápidas no programa de usinagem;
- controle de desgaste de ferramenta;
- simulação do programa da peça;
- integração com outros setores do processo produtivo.

Geralmente o CNC, em uma máquina ferramenta, trabalha em conjunto com o CLP, cuja função é a de gerenciamento do ambiente da máquina, possuindo para isso entradas e saídas, as quais podem ser digitais ou analógicas. Com o avanço tecnológico dos microprocessadores, o microprocessador do CLP passa a não existir em alguns fabricantes de equipamentos CNC, sendo as funções do CLP integradas no microprocessador do CNC. Tal configuração agiliza mais ainda a troca de dados entre o CLP e o CNC.

Alguns trabalhos mais recentes a respeito de CNCs, tratam dele de uma forma mais aberta, tanto de hardware como de sua programação. Permitindo a sua integração com equipamentos de vários fabricantes de forma mais simples e natural, aumentando a integração e a confiabilidade de todos os componentes do chão de fábrica (OWEN, 1996) (PROCTOR & ALBUS, 1997).

Programação CNC

Um programa CNC é uma seqüência lógica de retirada de sobre-material de uma peça. Esta seqüência deve ser escrita em códigos apropriados, de modo que o CN os interprete e emita os sinais apropriados à máquina para que esta execute a operação programada (PORTO, 1985).

Descreve-se a programação CNC de uma forma geral, pois como já dito anteriormente, o intertravamento não é realizado no CNC, no entanto para que ele seja efetuado pelo CLP, é necessário que o CNC ative funções ligadas ao CLP (liga/desliga

lubrificação, liga/desliga transportador de cavacos, etc.) durante a usinagem de uma peça.

A linguagem de programação utilizada é a chamada linguagem G (devido ao fato de que as instruções dadas à máquina virem sempre precedidas pela letra G) normalizada pela norma DIN 66025 (TRAUB, 1993).

Uma linha de comando é exemplificada abaixo, onde tem-se a numeração da linha N (chamado de bloco), a instrução de comando G, a ferramenta a ser utilizada T, funções ativadas em conjunto com o CLP M (Miscelânea), os eixos comandados X, Y ou Z, a velocidade de avanço dos eixos F e a velocidade de avanço do eixo árvore S (Spindle).

N10 G04 T10 M05

N20 G01 X50 -Z30 F100 S50

Os programas podem ser transmitidos à máquina por um equipamento leitor de fita perfurada, ou através de sistemas de comunicação serial (RS 232, RS422, etc.).

2.2.2 - Técnicas de Inteligência Artificial

A complexidade associada aos Sistemas de Manufatura, tanto no aspecto estrutural, quanto no aspecto comportamental, e a necessidade de soluções não analíticas, principalmente nos níveis mais altos da hierarquia de controle e funções de monitorização e recuperação dos eventos inesperados, tornou natural o uso de técnicas de inteligência artificial (KÜNZLE, 1990).

O problema de programação em tempo real em um sistema de manufatura é o de realizar as metas da produção na ocorrência de mudanças, devido aos recursos limitados, falhas aleatórias nas máquinas e introdução de novos pedidos. Pelo fato destes sistemas geralmente serem automatizados, os computadores podem realizar muitas das atividades de controle e roteamento. Em uma programação em tempo real de um chão de fábrica uma reprogramação automática é desejada. Tais sistemas podem utilizar uma combinação de técnicas baseadas na programação matemática e na inteligência artificial para atingir estas metas.

As abordagens matemáticas para a programação de um sistema de manufatura são muito limitadas devido à complexidade do sistema e a dificuldade de descrever sua conduta completa através de modelos. Os modelos então são aplicados a uma pequena série de questões, limitando o número de variáveis do sistemas. Nos níveis

mais altos da hierarquia de um FMS, por exemplo, tais abordagens se tornam muito complexas devido ao grande número de variáveis do sistema.

Em resposta a estas limitações dos modelos matemáticos, e as questões não numéricas, restritivas e metas qualitativas de um sistema de manufatura, o uso de inteligência artificial (AI) tem sido proposto para programar e controlar as funções do sistema de manufatura. Regras heurísticas são propostas para substituir o controle de sistemas baseados em modelos analíticos.

Sistemas especialistas, redes neurais, lógica nebulosa (fuzzy logic), etc. tem sido bastante utilizadas para o controle dos sistemas de manufatura, auxiliando nas diversas fases de tomada de decisão do sistema.

A implementação do controle combinatório de restrições na coordenação de células flexíveis de manufatura através de um sistema baseado em regras é descrita por Komoda (KOMODA et al.1984), enquanto Mainon (MAIMON, 1987) apresenta esta solução para o nível de seqüenciamento de processos de controle operacional em tempo real de um Sistema Flexível de Manufatura.

O sistema de controle pode ser baseado em regras que determinam as ações de acordo com a situação do sistema, usando regras tipo "IF THEN", quando os eventos ocorrem ou utilizando redes neurais para o seqüenciamento da produção.

Uma ferramenta de modelagem baseada em lógica nebulosa, redes de Petri e orientação a objeto, chamada de Object Network é proposta para a modelagem, análise e projeto de sistemas de manufatura inteligentes (GUDWIN & GOMIDE, 1998).

A lógica nebulosa também pode ser apresentada como uma ferramenta para expressar controle reativo em tempo real (POLITANO, 1996).

A combinação de redes de Petri com lógica nebulosa e redes neurais é descrita por Hanna (HANNA, 1996), para a modelagem de um centro de usinagem CNC. Esta técnica é baseada em 9 regras fuzzy, utilizadas para estabelecer a qualidade da rugosidade da superfície das peças acabadas.

Uma revisão sobre as técnicas de IA no aumento do desempenho e automação dos CNCs foi realizada por Rowe (ROWE, et. al., 1996), a qual cobre as pesquisas na área de controle inteligente e otimização das técnicas de usinagem, dentre as quais algumas serão apresentadas mais adiante.

Dentre ferramentas de IA, os sistemas especialistas tem sido usado com sucesso nas fases de um projeto de controle (intertravamento) de um sistema de manufatura. E

será abordado em maior profundidade devido ao seu emprego direto no ambiente de intertravamento proposto.

2.2.2.1 - Sistemas Especialistas

Sistemas Especialistas tem sido usados como um termo para identificar sistemas baseados em conhecimento. Um Sistema Especialista pode realizar tarefas de diversos campos de aplicação, resolvendo de forma satisfatória muitos problemas reais de relativa complexidade.

Um Sistema Especialista é ambiente computacional que tenta imitar a habilidade humana, aplicando as técnicas de inferência lógica a uma base de conhecimentos. Uma base de conhecimentos é diferente de uma base de dados ou arquivo de computador porque ela armazena informações sobre como realizar as tarefas, tais como regras que um especialista aprende pela experiência (BROOKS, 1987).

O Sistema Especialista tenta funcionar como a mente humana, onde a adição de uma nova regra não interfere no processo de inferência, ficando esta a disposição quando necessária.

Estabelecer quando um sistema é adequado para se aplicar a abordagem de Sistemas Especialistas tem sido uma preocupação constante na comunidade de I.A. (BELHOT, 1993).

A abordagem de aquisição de conhecimento de um Sistema Especialista, necessita em uma análise do estabelecimento de seu domínio. Duas escolhas estão envolvidas na escolha de um domínio. Uma é que ele seja um assunto de seu interesse e a outra é que o banco de conhecimentos seja pequeno o suficiente para que possa juntar uma quantidade de informações que possa ser administrada (LEVINE et al., 1988).

Neste sentido os esforços dos engenheiros de conhecimento tem revelado três critérios de decisão principais do uso de especialistas para avaliar o domínio pretendido (PEDERSEN, 1989):

- 1 - Retorno econômico;
- 2 - Necessidade do especialista;
- 3 - Conveniência da implantação.

Retorno econômico se refere aos benefícios econômicos que o sistema pode trazer. Necessidade do especialista analisa a necessidade e a escassez do especialista,

e a conveniência de implantação julga as chances do projeto proposto ser implantado com sucesso.

Outros arranjos para estes critérios neste mesmo nível podem ser sintetizados nas seguintes perguntas (WATERMAN, 1986):

- O desenvolvimento é possível ?
- O desenvolvimento é justificável ?
- O desenvolvimento é apropriado ?

Para se concluir se um desenvolvimento é possível, dois parâmetros devem ser analisados: a tarefa e o especialista humano. A figura 2.8 nos fornece esta visão.

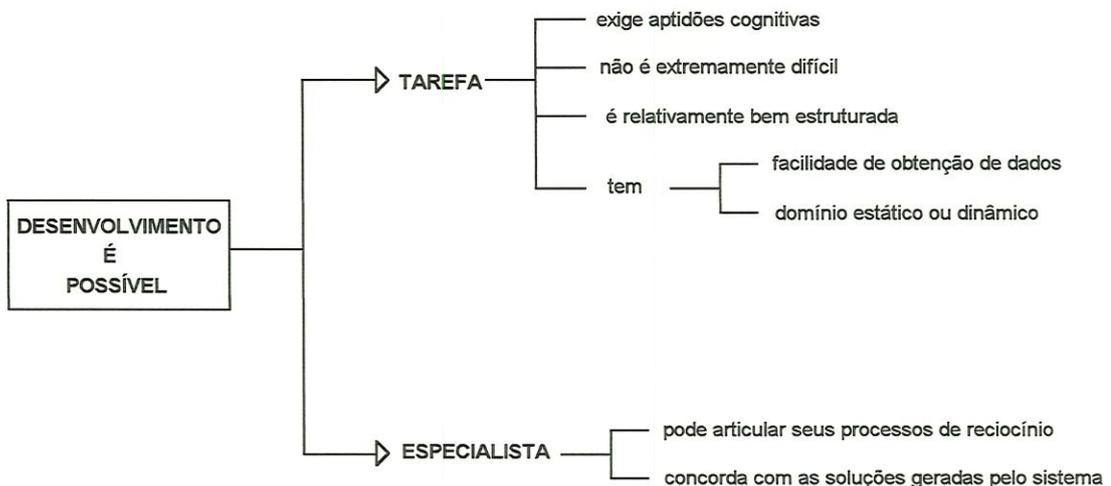


Figura 2.8 - Quando o desenvolvimento é possível de um S.E. (WATERMAN, 1986).

Um sistema especialista é justificável quando não houver a disponibilidade de especialista; quando um especialista é necessário em vários locais; quando o especialista é necessário em um ambiente hostil; ou quando a solução da tarefa tem um retorno alto.

O desenvolvimento de um sistema especialista é apropriado, quando possuir certas características inerentes, tais como permitir sua solução através de manipulação de símbolos, e de estruturas simbólicas, deve possuir uma característica essencialmente heurística, além de ser suficientemente limitado para ser gerenciado e suficientemente amplo para assegurar que haja algum interesse prático.

Utilizando os critérios descritos acima, podemos então montar uma estrutura hierárquica de base de conhecimentos como mostrado na figura 2.9.

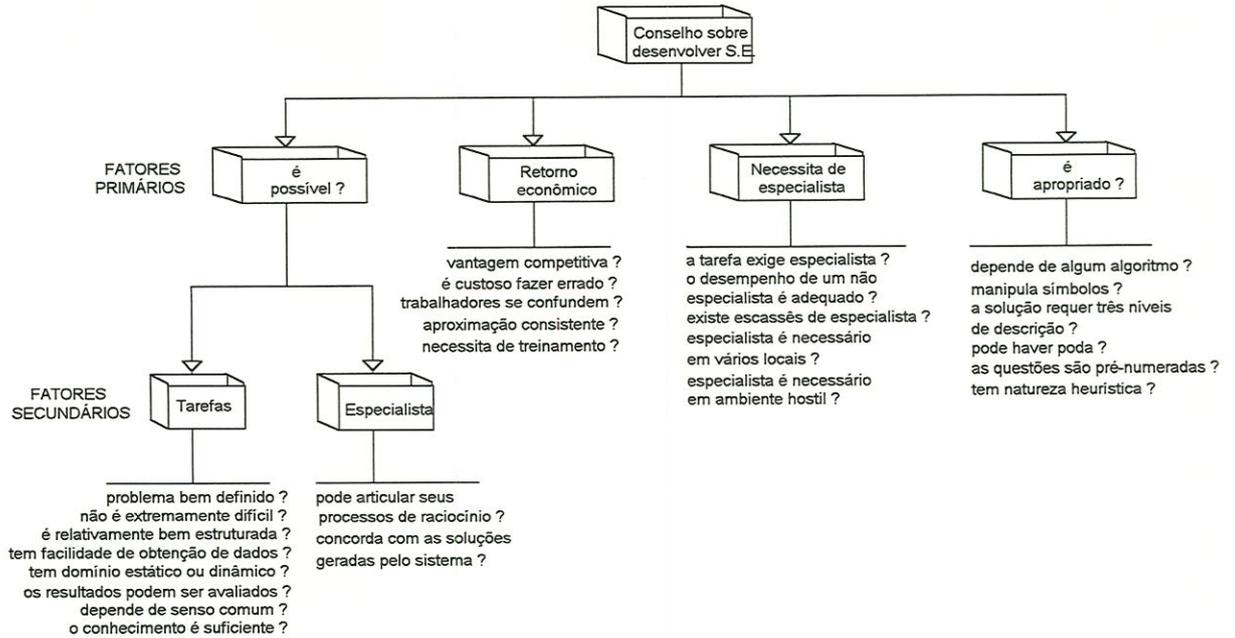


Figura 2.9 - Estrutura hierárquica da base de conhecimentos para o sistema proposto (adaptado de PEDERSEN, 1989).

2.2.2.2 - Aplicações de Sistemas Especialistas em Sistemas de Manufatura

O auxílio à tomada de decisão em um Sistema de Manufatura, se localiza tanto no planejamento da produção (PCP) como em todos os seus níveis hierárquicos descritos anteriormente. O planejamento da produção é ditado pelas metas organizacionais, fornecendo ao sistema o plano de produção, de acordo com as “due dates” ditadas pelo mercado. Enquanto que a estrutura funcional se encarrega de cumprir as tarefas frente as oscilações de mercado, mudanças de prioridades, alterações no sistema produtivo, etc.

De acordo com o Planejamento da Produção e a decomposição hierárquica do sistema de manufatura, pode-se então localizar em cada um destes níveis, os sistemas de apoio de decisão baseados em Inteligência Artificial, focalizando suas necessidades e apresentando alguns exemplos pesquisados.

Em termos de auxílio à decisão no PCP, Schoroer e Tseng (SCHOROER & TSENG, 1989) discutem o acoplamento de um sistema especialista com ferramentas de simulação de sistemas de manufatura, que usa uma interface inteligente para definir o modelo de simulação e um programa de geração automática que fornece os códigos de uma linguagem de simulação.

A junção de inteligência artificial e a simulação tradicional, se concentra em duas áreas distintas. A primeira área é o uso da AI para automatizar o problema de especificação do processo. E a segunda área, e mais difícil, é o uso de AI para automatizar a geração de códigos executáveis na linguagem de simulação desejada.

No auxílio à decisão no nível planta (definido na seção 2.1.3), Szelke e Meszaros (SZELKE & MESZAROS, 1990), apresentam os principais métodos comumente usados para resolver problemas de programação baseados em diferentes paradigmas, tais como:

- 1- Otimização combinacional para problemas clássicos job-shops;
- 2- Otimização estocástica;
- 3- Simulação de evento discreto com prioridade para regras de despacho;
- 4- Abordagens de A.I. usando sistemas especialistas e sistemas baseado em conhecimento.
- 5- Abordagens de controle teórico através da minimização ou maximização de uma das medidas de desempenho.

Da avaliação das abordagens acima, SZELKE concluiu que uma síntese de paradigmas tais como AI, controle teórico e simulação de evento discreto podem ajudar a modelar e resolver o problema de tomada de decisão em um modelo de eventos discretos, no sentido de otimizar a programação da operação de um sistema de manufatura.

Para o auxílio à decisão no nível célula, Ben-Aried (BEN-ARIED et al.1989), apresenta um método de controle do nível Célula baseado em conhecimento, enfatizando a habilidade de controle no tratamento de situações imprevistas em um mecanismo de programação ("*scheduling*") que determina tarefas às máquinas, e prepara considerações de processo sobre fatores globais, como metas de produção e distribuição de recursos. Este método é chamado pelos autores de "Generativo".

Este modelo de programação ("*scheduling*") baseado em conhecimento implica na solução para um problema de "*scheduling*" tantas vezes quanto necessário (reprogramação). Este sistema é composto de uma base de conhecimento, que é composta por quatro partes: A primeira é a base de dados estática que descreve as instalações e o produto independente do tempo de montagem. A segunda parte é a base de dados dinâmica, que armazena a situação atual do sistema. A terceira parte é o conhecimento comportamental que descreve as regras de operação, a coleta de dados de especialistas através de "corridas de simulação". A última parte é o

conhecimento procedural (algorítmico) composto de uma heurística que quantifica as várias alternativas e reduz o espaço de busca. O algoritmo heurístico é baseado no cálculo da expectativa dos tempos de espera dos componentes nas estações de trabalho. O algoritmo tenta reduzir a soma dos estados de espera.

No auxílio à decisão no nível de estação de trabalho e equipamento, pesquisas estão sendo realizadas no sentido da eliminação do operador da máquina e de uma correção rápida de falhas, para que o processo produtivo seja interrompido o menor tempo possível (correção automática).

Duggan (DUGGAN & BROWNE, 1988) consegue usando redes de Petri Temporizadas e uma lógica do sistema para disparos das transições com uma linguagem baseada em regras do tipo IF-THEN, uma ferramenta de pré-simulação que através de rápida prototipagem modela as principais características e funções de um sistema de manufatura.

Nagurka e Englert (NAGURKA & ENGLERT, 1989), descrevem os esforços da Universidade de Carnegie Mellon para criar uma máquina ferramenta autônoma para a fábrica do futuro. A máquina ferramenta deverá operar sob o controle do computador para manuseio (manipulação), fixação de peças, e corte em metal. Abordando modelos que utilizam sistemas especialistas para a substituição dos operadores humanos.

DiCesare (DICESARE et al. 1993), descreve um sistema baseado em regras para controle inteligente de uma estação de trabalho de montagem que é capaz de recuperar e corrigir automaticamente os erros que podem ocorrer. A implementação expande os conceitos de sistemas de regras de produção adicionando dados da descrição dos objetos ao modelo do sistema e apresentando o plano de montagem como uma lista de tarefas e relações de precedência, apropriadas para a sua montagem.

Qiu (QIU et. al., 1996) tratou da diagnose de falhas de um FMS utilizando sistema especialista, redes de Petri e lógica nebulosa. O sistema adquire dados em tempo real de um CLP (on-line) e os analisa de acordo com a sua lógica booleana para os sinais do CLP, realizando assim a detecção e diagnose das falhas.

Yan (YAN et. al., 1998) apresentou um sistema de modelagem da programação ("*scheduling*") e simulação de sistemas de manufatura baseado em redes de Petri estocásticas onde, com a ajuda de um sistema especialista, são colocadas algumas novas regras dinâmicas de reprogramação de acordo com o estado do sistema.

Comparações realizadas pelos autores com sistemas reais e o simulado com o sistema especialista mostram que ele é efetivo.

Neste trabalho os Sistemas Especialistas serão utilizados na fase de projeto, como uma ferramenta de modelagem da máquina e seus dispositivos. Atuando de forma que se possa realizar a análise das necessidades do projeto dos intertravamentos a nível de equipamento (máquina CNC).

2.2.3 - Ferramentas de modelagem.

Muitas metodologias de engenharia de software utilizadas na indústria para a modelagem de sistemas de manufatura não possuem uma base formal sólida, isto é, aplicam linguagens de modelagem unificadas (BOOCH, 1994), orientação a objeto (JACOBSON et. al., 1992) e outras abordagens não formais. Trabalhos recentes, mostram que uma busca por esta base formal vem se tornando cada vez mais evidente, e as redes de Petri cada vez mais vem se firmando como tal base (STÖRRLE, 1998).

Para que isso aconteça, uma linguagem de modelagem deve esclarecer uma série de conflitos (STÖRRLE, 1998):

- 1 – deve capturar as características estáticas e dinâmicas do sistema;
- 2 – deve acomodar o estilo de modelagem dos estados e de atividades de uma forma natural;
- 3 – deve permitir a sua integração em qualquer outra linguagem de modelagem;
- 4 – deve ser capaz de representar sistemas sobre múltiplos níveis de abstração.

Para a verificação da consistência, rastreamento (“debug”) e validação do modelo, é necessário os seguintes requisitos quanto ao seu formalismo:

- a – deve ser capaz de permitir simulação e animação;
- b – deve disponibilizar técnicas de análise eficientes para verificar as propriedades do modelo;
- c – deve possuir ferramentas de análise de fácil acesso.

Várias ferramentas que podem ser utilizadas para modelagem se encaixam na maioria destes critérios, tais como redes de Petri e suas extensões, Mark Flow Graph (HASEGAWA et al., 1988) (MYAGI et al., 1988) e Statecharts (BATES et. al., 1998) (RAUSH & KROGH, 1997).

O uso das redes de Petri visa fornecer esta base formal para a engenharia de software, sem retirar o conhecimento inerente das linguagens e modelagens realizadas pelos usuários, utilizadas em cada aplicação.

A rede de Petri é uma ferramenta que preenche a maioria destes requisitos, possuindo potencial para preencher os que faltam, como a integração com outras ferramentas.

Desta forma, as redes de Petri serão adotadas como ferramenta de modelagem para este trabalho devido ao seu formalismo matemático, por ser uma ferramenta gráfica bastante conhecida e por possuir características de análise e validação. Destaca-se também a sua habilidade em tratar sistemas em tempo real, os quais são caracterizados por ações assíncronas, paralelas e seqüenciais.

2.2.3.1 - Redes de Petri

Historicamente, o conceito de redes de Petri foi originado na dissertação de Carl Adam Petri, submetida em 1962 à Faculdade de Matemática e Física da Universidade Técnica de Darmstadt, Alemanha Ocidental. A dissertação foi preparada quando C. A. Petri trabalhava como cientista na Universidade de Bonn. O trabalho de Petri chamou a atenção de A. W. Holt, que o levou ao "Information System Theory Project of Applied Data Research, Inc.", nos EUA (MURATA, 1989).

Redes de Petri é uma ferramenta de modelamento gráfico e matemático aplicada a muitos sistemas, sendo uma ferramenta promissora para descrever e estudar informações processadas nos sistemas caracterizados como concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e/ou estocásticos (MURATA, 1989).

Como ferramenta gráfica, as redes de Petri, podem ser usadas como ajuda à comunicação visual, simulação de gráficos de fluxo, diagramas de blocos e redes. Marcas são usadas nessas redes para simular as atividades dinâmicas e concorrentes do sistema.

As redes de Petri têm sido propostas para uma larga variedade de aplicações, tais como o modelamento e análise de sistemas de software distribuídos, sistemas de base de dados distribuídos, programas concorrentes e paralelos, sistemas industriais, sistemas de eventos discretos, sistemas de memória de multiprocessadores, sistema de fluxo de dados computacional, sistema tolerante a falhas, circuitos assíncronos e estruturas, sistemas de compilação e operação, sistemas de informação em escritórios,

linguagem formal, e programas lógicos. Outras aplicações interessantes consideradas na literatura são as redes locais (LANs), sistemas legais, fatores humanos, redes neurais, filtros digitais e modelos de decisão (MURATA, 1989).

Nos sistemas de manufatura, redes de Petri têm sido amplamente aplicadas em modelagem, análise e controle. A capacidade das redes de Petri para modelar interações concorrentes e assíncronas dentro de um sistema de manufatura tem contribuído para seu desenvolvimento como uma poderosa ferramenta de modelagem, descrevendo o sistema graficamente e contribuindo para o entendimento das complexas interações internas dos sistema (CECIL et al., 1992).

As principais características das redes de Petri são (RILLO, 1987):

- a) é um método abstrato, o que permite a representação de diferentes tipos de sistemas;
- b) é um modelo rigorosamente formal (consistência matemática);
- c) é de fácil aprendizado, funcionando como linguagem de comunicação entre especialistas de diversas áreas;
- d) representa paralelismo e sincronização;
- e) representa aspectos estáticos e dinâmicos;
- f) é um método gráfico, que facilita o entendimento;
- g) possui métodos de análise, inclusive comerciais.

Uma rede de Petri é um gráfico direcionado consistindo de dois tipos de nós, chamado lugares e transições, onde os arcos são as ligações de um lugar para uma transição ou de uma transição para um lugar.

Na representação gráfica os lugares são desenhados como círculo, e transições como uma barra ou quadrado. Arcos são rotulados com seus pesos (positivos e inteiros), onde um arco com peso "k" pode ser interpretado como o conjunto de "k" arcos paralelos (quando rotulados com uma unidade são geralmente omitidos). Uma marcação (estado) atribui para cada lugar um inteiro não negativo. Se uma marcação atribui para o lugar "p" um inteiro não negativo "k", tem-se que "p" é marcado com "k" marcas. Para ilustrar isto, coloca-se "k" marcas no lugar "p".

Uma marcação é denotada por "M", um m-vetor, onde "m" é o número total de lugares. O p-ésimo componente de "M", denotado por $M(p)$, é o número de marcas no lugar "p".

Uma rede de Petri é uma quintupla. $RP = \{P, T, F, W, M_0\}$ onde:

$P = \{p_1, p_2, \dots, p_n\}$ é um conjunto finito de lugares,

$T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições,
 $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos (relação de fluxo)
 $W: F \rightarrow \{1, 2, 3, \dots\}$ é uma função peso
 $M_0: P \rightarrow \{0, 1, 2, 3, \dots\}$ é a marcação inicial,
 $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$.

Uma estrutura de rede de Petri $N = (P, T, F, W)$ sem nenhuma marcação inicial específica é denotada por N

Uma rede de Petri com uma dada marcação inicial é denotada por (N, M_0) .

$\bullet t_i = \{p / (p, t) \in F\}$ = é o conjunto de lugares de entrada de t

$t_i \bullet = \{p / (t, p) \in F\}$ = é o conjunto de lugares de saída de t

$\bullet p_i = \{t / (t, p) \in F\}$ = é o conjunto de transição de entrada de p

$p_i \bullet = \{t / (p, t) \in F\}$ = é o conjunto de transição de saída de p

Portanto uma transição " t_i " é habilitada para o disparo, se todos os elementos de " $\bullet t_i$ " possuir um número de marcas maior ou igual ao peso dos arcos relacionados. O disparo de " t_i " é efetuado e retira-se então de " $\bullet t_i$ " marcas equivalentes ao $W(\bullet t_i)$ e introduzindo-as ao " $t_i \bullet$ " criando um novo estado na rede.

A Figura 2.10 ilustra a forma gráfica das redes de Petri, onde pode ser verificada as notações para seus elementos (lugares, transições, arcos e marcas).

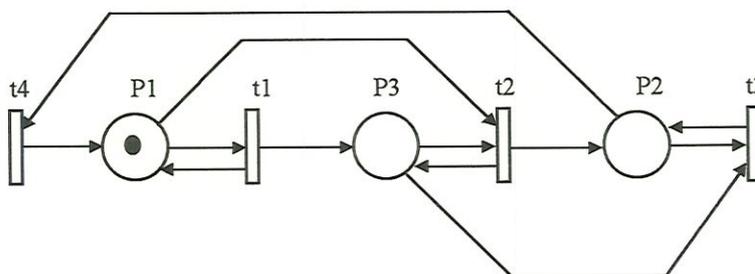


Figura 2.10 - Exemplo de uma rede de Petri

Uma das principais vantagens do uso das redes de Petri é o fato delas suportarem a análise de muitas propriedades comportamentais e a possibilidade de se realizar uma análise de desempenho do sistema com a adição do fator tempo às transições ou lugares. No seu artigo de revisão, Murata (MURATA, 1989) faz um levantamento abrangente apresentando as extensões de PN formalizadas, suas propriedades e métodos de análise, cobrindo com detalhes técnicas como árvore de alcançabilidade, matriz incidente e equações de estado, além de técnicas de redução ou decomposição.

Várias extensões das redes de Petri foram propostas para melhor se adaptar a cada tipo de sistema, visando a inserção da variável tempo (redes de Petri Temporizadas) ou a individualização das marcas (Redes Predicado-Transição (GENRICH, 1986), e redes de Petri Coloridas (CONSSINS e FERREIRA, 1992) (JENSEN, 1995), além de sua adaptação a processos estocásticos (rede de Petri Estocástica) (VISWANADHAM & NARAHARI, 1988 e BRUNO & BIGLIA, 1985). Extensões de redes de Petri individualizadas e priorizadas (CHOI et. al., 1994) que definem que atividades prioritárias devem ser disparadas antes, também tornam-se útil na modelagem de sistemas produtivos de manufatura.

Em redes de Petri Temporizadas (HOLLIDAY & VERNON, 1987), define-se adicionalmente o tempo de duração, ou atraso, " τ " associado a cada transição. Portanto o disparo de uma transição " t_i " envolve o início e o final do disparo de uma duração de " τ_i ". Durante o progresso de um disparo de " t_i ", os lugares de entrada " $\bullet t_i$ " já não possuem marcas, entretanto nos lugares de saída " $t_i \bullet$ " as marcas ainda não estão disponíveis. Portanto surge um novo estado do sistema que poderá informar além do estado de marcas, quando uma transição pode disparar definindo os tempos de disparo (tempo inicial e final) (INAMASU, 1995).

Pode-se destacar alguns trabalhos que utilizam redes de Petri relacionados com sistemas de manufatura discreta:

Em termos de modelagem dos níveis mais altos da hierarquia de um sistema de manufatura, podem ser encontrados trabalhos como o de Sternemann e Zelm (STERNEMANN & ZELM, 1998) que tratam do modelo operacional de suporte a tomada de decisão de uma empresa, utilizando redes de Petri com um base de dados SQL para a modelagem e simulação temporal do modelo da empresa.

Luo e Gu (LUO & GU, 1998) apresentam um método de modelagem de informações para um sistema de manufatura integrada por computador (CIM – Computer Integrated Manufacturing). Uma estrutura CIM é projetada e analisada através de uma conjunto de modelos em redes de Petri predicação transição estendida.

A programação de células e sistemas de manufatura automático (*scheduling*), principalmente de FMS, utilizando redes de Petri tanto para o controle como para a sua avaliação é tratada por vários autores (JENG & LIN, 1997) (LIN & LEE, 1997) (MALO et. al., 1998) (PENG & CHEN, 1998).

Inamassu (INAMASSU, 1995) apresentou em sua tese uma plataforma para simulação e planejamento de FMS (Sistemas Flexíveis de Manufatura), baseado em modelos dos componentes do sistema em redes de Petri.

Valvanis (VALVANIS, 1990) propõe um modelo para FMS onde utiliza-se uma modelagem hierárquica do sistema, assegurando as propriedades de vivacidade, limitabilidade e consistência do modelo, através de múltiplos tipos de lugares, múltiplas classes de fichas e múltiplos arcos, representando a presença e o fluxo de peças, recursos e informações de controle.

Murgatroy e Masory (MURGATROY & MASORY, 1996) descrevem uma estrutura de simulação e controle de sistemas de manufatura que permite a criação de uma simulação para o controle de um sistema de manufatura automático utilizando redes de Petri com arcos inibidores.

Ezpeleta e Colom (EZPELETA & COLOM, 1997) pesquisam a modelagem de FMS em dois componentes: layout e planos de processo. Estes modelos são construídos utilizando-se redes de Petri coloridas e depois integrados automaticamente em um único modelo para que possam ser realizadas políticas de prevenção de deadlocks.

No controle e simulação de sistemas de manuseio e transporte, vários trabalhos podem ser encontrados:

Hollier (HOLLIER, 1987) destaca a rede de Petri como ferramenta para simulação de rotas de um AGV em um sistema de manufatura flexível.

Raju e Chetty (RAJU & CHETTY, 1993) utilizam redes de Petri Temporizadas para avaliar um sistema de transporte baseado em AGVs, utilizando parâmetros como número de AGVs, quantidade de buffers e entroncamentos.

Hsieh (HESIEH & LIN, 1994) utiliza as redes de Petri para elaborar estratégias de controle de rotas e colisões em sistemas de AGVs, projetando possíveis rotas de colisões através das técnicas de análise das redes de Petri.

Chincholkar e Krishnainh (CHINCHOLKAR & KRISHNAINH, 1996) propõe a integração de um sistema de manuseio de materiais (AS/RS) utilizando redes de Petri coloridas estocásticas (SCPN) para a modelagem, simulação e análise. Na análise, a simulação das SCPNs são auxiliadas pelo método Taguchi de projeto de experimentos.

Hsieh (HSIEH, 1998) apresenta um método de modelagem usando uma classe de redes de Petri coloridas chamada de Command-place Coloured-timed Petri Nets (CP-

CTPN), a qual é capaz de modelar várias características dos sistemas de AGVs tais como as funções das zonas de controle e bloqueio de veículos, que não são geralmente avaliados em programas comerciais de propósitos gerais.

Em termos de controle de equipamentos Künzle (KÜNZLE, 1990) em sua dissertação de mestrado, utilizou redes de Petri para especificar o controle à nível de equipamento e estação de trabalho em um ambiente de manufatura flexível, caracterizando os níveis hierárquicos de controle de um sistema e as ferramentas adequadas para projeto em cada nível.

Lobão (LOBÃO, 1995) apresentou na sua dissertação de mestrado uma modelagem de um torno CNC, utilizando redes de Petri a partir de diagramas de blocos lógicos.

Huang (HUANG & YANG, 1992) utiliza a rede de Petri colorida temporizada para modelar e validar o controle de uma célula de manufatura, onde todos os elementos de controle, incluindo PLC e CNC de robôs são especificados baseado no modelo validado.

Hanna (HANNA, 1996) realiza a combinação de redes de Petri com lógica nebulosa e redes neurais para a modelagem de um centro de usinagem CNC. Esta técnica é utilizada para avaliar a qualidade da rugosidade da superfície das peças acabadas. As redes de Petri utilizadas são as Fuzzy Petri Nets que definem as variáveis de entrada, marcação e disparo de seqüências em função dos elementos de lógica nebulosa.

2.2.4 –Aplicação de redes de Petri em sistemas automáticos

As redes de Petri vem sendo aplicadas com bastante intensidade na modelagem de controle automático, como mostrado anteriormente. Nesse item serão relatados os esforços da utilização das redes de Petri no que diz respeito a sua utilização no controle de sistemas de manufatura, aos ambientes de software para a ajuda ao projetista de sistemas automáticos e nas várias extensões criadas para se adaptar a cada tipo de sistema.

Os assuntos cobertos podem ser divididos da seguinte maneira:

Propostas de programação de sistemas de manufatura automático utilizando redes de Petri para programação de CLPs; apresentado em 2.2.4.1, idealiza várias propostas para a utilização das redes de Petri como uma ferramenta de estruturação da

programação CLP, tanto na forma de metodologias como de ambiente de softwares e variações baseadas em redes de Petri.

Ambiente de Software e Softwares de redes de Petri; apresentado em 2.2.4.2, fornece os recentes desenvolvimentos de softwares e ambientes de desenvolvimento de projeto, fornecendo através de redes de Petri uma forma gráfica de se representar os elementos básicos, com suas devidas consistências matemáticas, para a modelagem de sistemas de forma geral.

Extensões de redes de Petri para modelagem de Sistemas de Eventos Discretos e aplicações em sistemas de manufatura; apresentado em 2.2.4.3, aponta algumas abordagens de projeto com redes de Petri para a redução da complexidade da representação gráfica de determinados sistemas, onde abordagens hierárquicas, modulares e orientada a objetos são utilizadas.

Análise de desempenho dos modelos em redes de Petri; , apresentado em 2.2.4.4, são focalizadas as análises sobre as propriedades comportamentais das redes de Petri para verificar o desempenho dos sistemas modelados.

2.2.4.1 – Propostas de programação de sistemas de manufatura automático utilizando redes de Petri para a programação de CLPs

Em sistemas de manufatura automatizados, o controle é realizado pelo computador que processa o sincronismo de um grupo de máquinas para a produção de uma variedade de peças. Este sistema pode ser considerado um sistema rígido devido a execução de tarefas repetitivas. A sua automação consiste na verificação de uma série de sensores e/ou transdutores, na atuação sobre válvulas e relés, além da interação de máquinas CNC e CLPs (D'SOUZA & KHATOR, 1994).

Muitas contribuições podem ser encontradas no uso de redes de Petri para a modelagem do controle em termos de planejamento de processo ou controle de supervisão automático (WILLSON & KROGH, 1990) (GIUA & DICESARE, 1994) (STREGE, TOLLE & LOYDL, 1995) (MASSBERG et al., 1990) (WONGTALADKOWN & CHEN, 1990) (CHEN & WONGTALADKOWN, 1991) (MORI, TSUKIYAMA & FUKUDA, 1990) (TANG & ZHOU, 1992) (NAGAO, et al. 1996) (JENG & SHEIH, 1993) (GRACANIN & SRINIVASAN, 1994) (BAROZZI, et al., 1994).

São destacadas as contribuições que utilizam redes de Petri na aplicação de controle de equipamentos, em termos de controladores lógico programáveis (CLPs), onde são analisadas as entradas, o processamento destas entradas segundo uma lógica de controle e são atuadas as saídas deste sistema. Estes sistemas podem ser máquinas isoladas, estações de trabalho, células de manufatura e até Sistemas Flexíveis de Manufatura (FMS) desde que o controle seja realizado por meio de controladores lógicos programáveis. Algumas pesquisas são descritas a seguir:

Chang et. al., (CHANG, et al., 1993) sugerem uma implementação de redes de Petri para aplicações em controladores programáveis, nesta implementação, um modelo de uma subclasse de redes de Petri é sugerido. O modelo sugerido, B-Petri net, é um gráfico de marcas onde os números de lugares de entrada e de saída de uma transição são limitados a dois. O B-Petri net é analisado, e uma arquitetura de implementação de hardware de um controlador programável baseada nela também é sugerida. As tabelas de memória possuem tamanho executáveis, e podem ser bem reduzidas no caso de alguma aplicação real como um controlador programável.

As Interações de processos em sistemas de manufatura concorrentes podem resultar em “deadlocks” os quais são difíceis de serem detectados durante o estágio de projeto do programa de controle. Desta forma, D’Souza (D’SOUZA, 1994) relata a aplicação de um “Modelo de Controle” para detecção de “deadlocks” em uma célula de usinagem automatizada composta por um torno CNC e um robô. O modelo de controle é gerado de uma listagem de eventos de um CLP usando a teoria das redes de Petri. O modelo é testado para “deadlocks” que poderiam possibilitar a ocorrência de falhas devido a natureza concorrente das atividades da célula. Os resultados mostram uma perspectiva promissora da aplicação do modelo de controle na depuração de programas CLP.

Para estabelecer as redes de Petri como uma alternativa para a Linguagem Ladder existe a necessidade de se justificar os métodos das redes de Petri. Isto pode ser coberto parcialmente pela comparação da complexidade das redes de Petri e dos projetos em lógica Ladder (LLD – Logic Ladder Diagram). Venkatesh, Zhou e Caudill (VENKATESH, ZHOU & CAUDILL, 1994) apresentam uma metodologia para avaliar a complexidade das redes de Petri e LLDs para o projeto de controladores sequenciais. Ela é caracterizada pelo número de elementos básicos usados para modelar uma dada lógica de controle. Mais especificamente, apresentam fórmulas analíticas para estimar o número de elementos básicos para modelar certos blocos de lógica quando redes de Petri e LLD são usados.

Uma aplicação industrial destas duas metodologias (redes de Petri e LLD) de projeto são comparadas usando um exemplo de projeto industrial (ZHOU & TWISS, 1995). As comparações são feitas sobre a habilidade de avaliar a lógica programada, a flexibilidade da lógica, a habilidade de solucionar problemas e de depurar o programa do sistema. Todavia, para aplicações em larga escala de redes de Petri na indústria, existe a necessidade de um procedimento de projeto sistemático para o desenvolvimento dos modelos em redes de Petri. Este procedimento é proposto a partir de um simples processo de conversão para formular modelos de uma dada lógica de especificação de controle (VENKATESH, ZHOU & CAUDILL, 1995).

Strege, Tolle & Loydl, (STREGE, TOLLE & LOYDL, 1995) propõe superar o problema da falta de adaptação dos operadores quando necessitam realizar algumas modificações no controle de uma FMC, através do uso de comandos específicos e informações sobre os sistemas de manufatura, que podem ser especificados através de comandos orientado a objeto, seguindo conceitos morfológicos usados por operadores. Estes conceitos são então automaticamente transformados em redes de Petri que é a ferramenta utilizada para modelagem de FMCs devido sua técnica de modelamento gráfico, facilitando sua análise matemática e sua transformação para programas CLP.

Na engenharia de controle, os modelos de sistemas controlados são a base para a síntese de controle tanto como para a análise ou simulação da sua conduta. Modelos em redes de Petri de sistemas controlados permitem a execução automática de um algoritmo de controle, sendo especificada pelos estados permitidos ou proibidos ou estados seqüenciais. O algoritmo de controle que é igualmente representado pelas redes de Petri é então automaticamente traduzido para os códigos de CLPs. O modelo do sistema controlador e a síntese do algoritmo de controle são usadas para gerar automaticamente dados para o diagnóstico do modelo baseado no sistema de diagnose do CLP proposto por Lemmer, Ober e Schenieder (LEMMER, OBER & SCHENIEDER, 1995).

Zaytoon (ZAYTOON, 1996) utiliza uma metodologia de decomposição hierárquica das redes de Petri para se obter a implementação de uma lógica de controle de um sistema de manufatura automático na forma da linguagem de programação GRAFCET. Esta metodologia permite aumentar a comunicação entre o projetista do sistema de manufatura e o programador do controle, além de gerar a lógica de controle a partir de uma especificação de alto nível e fornecer uma base para a validação do sistema de controle.

Um método de projeto e implementação de sistemas de controle de eventos discretos baseado em CLPs é apresentado por Uzam e Jones (UZAM & JONES, 1998). Eles propõem uma nova extensão das rede de Petri ordinária, para trabalhar com sensores e atuadores, chamada APN (Automation Petri Net). Nesta extensão são utilizados arcos habilitadores e inibidores, os quais podem habilitar ou desabilitar transições, e um conversor das APN para linguagem Ladder segundo o padrão IEC1131-3.

2.2.4.2 – Ferramentas de Software baseadas em redes de Petri

A criação de ferramentas de software para a edição e simulação de redes de Petri teve a partir da década de 90 um grande impulso devido às evoluções tecnológicas na informática e ao fácil acesso a essas tecnologias pelos pesquisadores e usuários.

Três grandes bancos de dados utilizados para a pesquisa destas ferramentas estão disponíveis na internet: www.daimi.aau.dk/PetriNets (CHRISTENSEN & MORTENSEN, 1997); www.informatik.uni-dortmund.de/lubeck/pf279/modprak/petri-tools.html (LÜBECK, 1995); e www.dsi.unimi.it/Users/Tesi/trompede/petri/alfa.html (TROMPEDELLER, 1993).

Uma revisão abrangente e atual é realizada por Störrle (STÖRRLE, 1998), sobre as ferramentas de softwares existentes, tanto comerciais como acadêmicas. Esta revisão utiliza como base pesquisas anteriores (FELDBRUGGE & JENSEN, 1987), (FELDBRUGGE, 1990), (FELDBRUGGE, 1993) e complementa o trabalho realizado por Wikarski (WIKARSKI, 1997) que classifica as ferramentas através de critérios baseados na engenharia de software.

A classificação, o “ranking” e a descrição da maioria destas ferramentas podem ser encontrado nesta revisão e nos bancos de dados destas ferramentas, que estão disponíveis nos endereços da internet descritos anteriormente.

Algumas ferramentas, tais como AlphaSim, Design/CPN e Netmate, que fazem parte desta revisão, serão descritas junto com outros trabalhos que tratam da automação dos sistemas de manufatura.

Também serão descritas ferramentas estruturais como Petri Net Kernel, que possibilita a criação de uma ferramenta específica e INA e PROD, que são utilizadas como ferramentas de análise, que podem ser integradas com outras ferramentas.

O ALPHA/Sim é uma ferramenta de simulação de eventos discretos de propósitos gerais (MOORE & BRENNAN, 1995). O ALPHA/Sim permite ao usuário construir graficamente o modelo de simulação, entrar com dados através de vias integradas, executar o modelo de simulação, e ver os resultados da simulação, dentro de um ambiente gráfico. As classes de rede de Petri modeladas são as estocásticas, temporais e as coloridas.

Design/CPN utiliza as redes de Petri coloridas (CPN) para a modelagem de sistemas, e a ferramenta consiste de três partes integradas: o editor de redes de Petri coloridas, que permite a construção, modificação e verificação da sintaxe de modelos em CPNs; um simulador que pode atuar de forma interativa ou automática para a simulação de CPNs; e uma ferramenta de construção e análise de árvore de alcançabilidade para modelos de CPNs (JENSEN, 1995).

Netmate é uma ferramenta de projeto, simulação, verificação e implementação de controladores seqüenciais industriais. Ela pode gerar automaticamente o programa CLP para controladores Siemens da família PLC S5. Utiliza uma extensão das redes de Petri chamada de IPN (Interpreted Petri Nets), a qual possui características hierárquicas, arcos adaptativos e temporais e pode abordar a modelagem de sistemas híbridos (FREY, 1998). Ela possui também uma verificação analítica baseada em métodos próprios e pode integrar ferramentas de análise, como INA.

INA (Integrated Net Analyzer) é uma ferramenta que suporta a análise de redes de Petri lugar – transição e redes de Petri coloridas. Possui um editor de redes de Petri, mas no entanto o usuário pode utilizar qualquer outro editor, desde que a rede seja convertida para o formato do arquivo INA. Sua análise permite tratar de informações estruturais, análise invariante, tanto de lugar como transição e árvore de alcançabilidade, verificando as propriedades de limitabilidade, vivacidade, reversibilidade, etc. (CHRISTENSEN & MORTENSEN, 1997).

PROD é uma ferramenta de análise de alcançabilidade das redes de Petri lugar transição a qual implementa vários métodos para este fim. É considerada uma ferramenta que pode interagir com outras de um ambiente Unix (CHRISTENSEN & MORTENSEN, 1997).

PEP (Programming Environment based on Petri nets) é um conjunto de componentes de modelagem, simulação, compilação e verificação. É utilizada para a modelagem de sistemas paralelos por programas paralelos, interagindo autômatos finitos, álgebra ou redes de Petri de alto ou baixo nível. Seu simulador permite

simulação automática ou dirigida pelo usuário e seus componentes de verificação possuem algoritmos de verificação das propriedades de alcançabilidade e deadlocks. PEP possui também interações com outros pacotes, como INA e PROD (STÖRRLE, 1998).

Petri Net Kernel (PNK) é uma infra - estrutura para desenvolver ferramentas baseadas em redes de Petri. O programador pode implementar funções padrões tais como carregar e salvar redes, acessar e modificar uma estrutura de rede e construir interfaces gráficas. Diferentes funções podem ser combinadas e integradas dentro de uma simples ferramenta. O PNK não é restrito a um tipo particular de rede de Petri, pois o tipo de rede de Petri pode ser especificado pelo programador (KINDLER, WEBER apud STÖRRLE, 1998).

Massberg et al. (MASSBERG et al., 1990) apresentam um ambiente de automação que pode ser usado permanentemente para suportar o ciclo de vida de um complexo sistema de produção desde as fases de planejamento, projeto até a execução da planta. A estrutura uniforme e consistente dos dados de redes de Petri, permite sua aplicação para a modelagem, simulação, controle e diagnóstico. Uma condição importante é a combinação dos componentes de transição com o tempo.

Oswald, Esser e Mattmann (OSWALD, ESSER & MATTMANN, 1990) discutem um ambiente para a construção e execução da especificação das necessidades de um sistema de propósitos gerais. "*High-order Petri nets*" são usadas para a especificação deste sistema. O ambiente consiste de um "net browser", um editor e um simulador animado da rede. Uma especificação estruturada hierarquicamente pode ser construída usando um editor de redes gráfico. O simulador animado ajuda a detectar erros e inconsistências. O "Petri net browser" permite gerenciar várias hierarquias e reutilizar redes. A máquina de simulação permite que as especificações sejam executadas em tempo real no ambiente.

Um simulador de redes de Petri Tempo Real (RTPNS), incluindo especificações de software, algoritmo interativo, regras de disparo e implementação em tempo real é apresentado (WONGTALADKOWN & CHEN, 1990) e (CHEN & WONGTALADKOWN, 1991). O software fornece uma ferramenta de projeto intuitiva com orientação gráfica e capacidade de depuração para o modelo do controle de sistemas de manufatura automatizados. O desempenho total do sistema é avaliado pelo software desenvolvido através de simulação computacional. O tempo de disparo, uma chave para a taxa de produção, pode ser otimamente selecionada pelos resultados das simulações. Uma

simulação do controle de um protótipo de sistema flexível de manufatura usando RTPNS é descrito.

Tang e Zhou (TANG & ZHOU, 1992) da NJIT (New Jersey Inst. of Tech.) apresentam uma ferramenta baseada em redes de Petri que funciona sobre uma plataforma PC-DOS. A ferramenta de software tem sido utilizada para a simulação de sistemas como o AT&T Flexible Workstation e o sistema flexível de manufatura da NJIT. Sendo que as pesquisas futuras são direcionadas para que o simulador baseado em PC possa ser utilizado para aplicações industriais.

Fleischhack e Litchblau (FLEISCHHACK & LITCHBLAU, 1993) introduzem MOBY, uma ferramenta de modelagem e análise de sistemas utilizando redes de Petri temporizada hierárquica. A ferramenta consiste de um editor para a especificação da rede e um simulador para a validação dos modelos construídos. O simulador possui um método de resolução de conflitos, o qual permite reduzir a complexidade combinatorial de procedimentos de execução do modelo, pelo uso do conhecimento específico sobre o sistema. A ferramenta pode ser usada para modelagem de processos de escritórios (serviços) tanto quanto para processos técnicos (produção).

Um sistema de suporte básico para desenvolvimento de programas de controle sequenciais chamado K-NET é apresentado (NAGAO, et al., 1993). A descrição do modelo em K-NET é baseada em redes de Petri coloridas e em redes de Petri temporizadas. A K-NET expressa concisamente a seqüência do fluxo de controle incluindo sincronização, intertravamento e concorrência, e fornece um processamento de dados para ser combinado com uma linguagem procedural convencional. A K-NET possui um editor, um simulador, gerador de relatórios e suporte para monitorar o procedimento do desenvolvimento de programas de controle para projetos básicos e detalhados com programação e teste. As funções de edição do modelo e as funções de especificação são explicadas e então um sistema de planejamento é descrito para um sistema AS/RS (NAGAO, et al. 1996) que utiliza a função de simulação K-NET a qual gera automaticamente um modelo de simulação, animações da conduta do sistema e avalia o desempenho do sistema.

Uma ferramenta de software chamada Petri Net Syntesis Tool (PNST) tem sido desenvolvida (JENG & SHEIH, 1993) para uso na modelagem por redes de Petri de sistemas de manufatura automáticos. A PNST é projetada para ser geral o suficiente para ser aplicada em diferentes tipos de redes de Petri.

Um ambiente de software para simulação e coordenação em tempo real de sistemas de manufatura é descrito (GRACANIN & SRINIVASAN, 1994). Ele é composto por um Sistema de Alocação Dinâmica de Recursos (DRAS), um sistema de software integrado para simulação, programação, gerenciamento de informação e monitoração do desempenho do sistema. Os três principais componentes do DRAS, a interface do sistema, o analisador de fluxo de trabalho e o sistema de conhecimento são descritos em maiores detalhes, tanto quanto as aplicações das redes de Petri.

O trabalho de Barozzi et al. (BAROZZI, et al., 1994) enfoca a simulação em tempo real de plantas industriais baseada em redes de Petri. O XpetreX é um sistema para modelagem, simulação em tempo real e teste de sistemas baseado em uma linguagem de redes de Petri de alto nível. Uma ferramenta derivada do XpetreX é apresentada por Camurri e Megagnini (CAMURRI & MEGAGNINI, 1998) ela é basicamente dividida em dois sistemas, o de desenvolvimento e o de execução (runtime). A sua abordagem orientada ao objeto fornece ao usuário uma facilidade de projeto e de reutilização dos módulos permitindo assim um rápido desenvolvimento das aplicações por técnicos em plantas industriais.

Um método sistemático para construir um modelo em redes de Petri lugar transição temporizado para modelar um FMS é proposto (SUN & FU, 1994). Um gerador automático de redes de Petri é proposto. O gerador automático possui uma Interface gráfica de usuário, na qual o usuário pode entrar com as informação de um FMS prático, tais como o número de AGVs, o número de máquinas, as relações geométricas entre eles e o fluxo do processo das peças a serem processadas. Então, a partir daí é gerado o modelo correspondente em redes de Petri.

Lee, Lu e Lin (LEE, LU & LIN, 1994) apresentam uma representação chamada PM-Net para a modelagem e monitoração do processo de desenvolvimento de projetos. Esta representação fornece informações para o gerenciamento do projeto em diferentes níveis de detalhes. A representação trata os dados de uma forma "bottom-up" as informações de uma forma "top-down". O diagrama de fluxo de dados (DFD) e a técnica de "Work Breakdown Structure" (WBS) são usados para a construção de uma estrutura hierárquica de desenvolvimento do processo de software.

Uma interface gráfica (GUI) para uma ferramenta Ambiente de Software de Auxílio a Modelagem (MASE) baseado em redes de Petri é proposta (TANG & HURA, 1994). A ferramenta MASE é um pacote de redes de Petri hierárquicas orientadas ao objeto. O pacote proposto permite criar, movimentar, apagar, alinhar objetos, criar sub e super redes, etc. interativamente através de ícones. Deste modo, simulações

automáticas e manuais também podem ser realizadas. A MASE suporta redes de Petri temporizadas, redes de Petri de alto nível e é proposto incluir outras extensões de rede para torná-la uma plataforma mais genérica do projeto de sistemas. As interpretações das redes de Petri podem ser adicionadas pelo uso da MASE. A rede básica é a rede de Petri orientada a objeto hierárquica (MASEHOOPN) a qual é definida no pacote. A ferramenta também suporta várias propriedades de análise tais como "safety", "boundedness", "liveness", etc.

Uma ferramenta de geração de códigos para simulação e controle baseadas em redes de Petri coloridas é introduzida por Dicesare, Gile e Kulp (DiCESARE, GILE & KULP, 1995). Esta ferramenta é usada em conjunto com um editor de redes de Petri chamado NETMAN para criar uma série de ferramentas integradas para projeto, simulação e controle. Os detalhes da geração dos códigos são discutidos, junto com algumas possibilidades de uso dos códigos gerados para ambos simulação e controle. Um exemplo uma rede de Petri estocástica é apresentado, e os resultados da simulação são comparados com a solução numérica.

O objetivo básico de TOMSPIN (Tool for Modelling with Stochastic Petri Nets) (THURNER, 1995) é fornecer aos usuários de indústria uma ferramenta que possa analisar sistemas grandes utilizando modelos GSPN (redes de Petri estocásticas generalizadas) com algumas pequenas extensões. TOMPSIN consiste de uma linguagem de descrição em rede hierarquicamente estruturada, com um editor gráfico. Depois de transformar a rede em uma estrutura de Markov, ela pode ser avaliada por análises transientes ou em estado estacionário.

Uma ferramenta gráfica para modelagem em redes de Petri temporizadas é apresentada por Liu e Robbi (LIU & ROBBI, 1995), onde o objetivo foi o de desenvolver uma ferramenta computacional para desenho, edição e simulação de redes de Petri temporizada usando programação orientada a objeto (OOP). O desenvolvimento de uma ferramenta de simulação de redes de Petri temporizada baseado em C++, chamada TiPNet, simula eventos discretos com transições estocásticas e imediatas. A depuração e o desempenho dos resultados são analisados. O Kit de ferramentas Xview é usado para construir um gráfico interativo com a interface do usuário das TiPNets junto com o OPENLOOK padrão da AT&T em uma estação de trabalho SUN rodando SunOS 4.1.X.

O TimeNET (GERMAN, et al., 1995) é um pacote de software para modelagem e avaliação de redes de Petri estocásticas (SPNs) na qual o disparo de tempo das transições pode ser exponencialmente distribuído, determinístico ou mais geralmente

distribuído. Os modelos são especificados através de uma interface gráfica. Uma classe de distribuição de disparo no tempo pode ser usada, chamada de distribuição polinomial a qual é parte da especificação de polinômios exponenciais. Outros componentes especializados são fornecidos para a avaliação do modelo: análise estacionária, análise transiente, aproximação e simulação de componentes.

Bobeanu e Filip (BOBEANU & FILIP, 1995) introduzem o ModCPN que é uma ferramenta integrada para modelagem e simulação a qual possui uma interface de gerenciamento em Windows, fornecendo suporte na modelagem, análise de propriedades estáticas e dinâmicas do modelo, além da simulação de sistemas de produção discreto. Menus "pull down" determinam os parâmetros globais do modelo, a edição do modelo, as estratégias de controle e as condições sobre as quais as ferramentas serão implementadas. A base matemática do sistema é uma descrição do modelo em termos de redes de Petri lugar/transição, colorida e temporizada.

Chao e Wang (CHAO & WANG, 1995) apresentam a XPN-FMS que é uma ferramenta de CAD (Computer Aided Design) baseada na modelagem em redes de Petri. Esta ferramenta integra as funções de desenho, análise, redução, síntese, propriedade de filas e animação de sistemas flexíveis de manufatura (FMS) em um único pacote. Usando XPN-FMS com várias entradas e comparando os resultados de saída, o usuário pode determinar como aumentar a eficiência, reduzir os custos, e localizar gargalos. Para os modelos em redes de Petri do FMS que são livre de decisões, a teoria e o algoritmo são baseados no método de uma única matriz. Este método é estendido para buscar por "loops" sub-criticos e suporte da programação da produção (scheduling), tratando com períodos de transições. XPN-FMS implementa este método para a seqüência de entradas de controle.

Um simulador gráfico para controladores lógico programável em redes de Petri é introduzido por Baker, Grant e Song (BAKER, GRANT & SONG, 1996) onde uma rede de Petri estendida, a Rede CLP (PLCnet) é definida. Um simulador baseado na PLCNet foi projetado baseado na semântica da PLCNet, com facilidades fornecidas para "setup" e controle de simulação. Ambos, os passos e respostas no tempo são fornecidas para a apresentação dos resultados da simulação. Um ambiente gráfico é fornecido para a construção da PLCNet, simulação e a apresentação dos resultados da simulação.

Uma linguagem de modelagem orientada a objeto baseada em redes de Petri é implementada por Esser (ESSER, 1997) para projetar sistemas automáticos. Esta linguagem constitui a parte principal da metodologia de projeto CodeSign utilizada no projeto de sistemas com a ferramenta Petri Net Browser.

Ceska (CESKA et. al., 1998) apresenta uma ferramenta orientada a objeto baseada em redes de Petri chamada de "PN talk ", a qual combina as vantagens das redes de Petri e a orientação a objeto. A extensão OOPN é descrita e os problemas que acompanham a análise formal das OOPNs são discutidos.

2.2.4.3 – Extensões de redes de Petri para modelagem de sistemas de eventos discretos e aplicações em sistemas de manufatura

Várias extensões são apresentadas para a modelagens de sistemas de eventos discretos, principalmente quando aplicados a sistemas de manufatura, algumas destas extensões são descritas a seguir:

Krogh e Holloway (KROGH & HOLLOWAY, 1991) tratam a supervisão da coordenação e controle de atividades concorrentes em instalações de manufatura automatizadas. A lógica de transição de estado para os equipamentos de manufatura é representada por uma extensão das redes de Petri chamada redes de Petri de Controle (CtlPN) com entradas e saídas externas a serem determinadas pelo algoritmo de controle do sistema. Formula-se os estados proibidos do problema de controle do sistema no contexto CtlPN e utilizando um algoritmo é verificado se o sistema irá evitar os estados proibidos. A formulação do problema e a síntese do algoritmo de controle é ilustrada com um exemplo de coordenação de AGV.

Uma nova classe de redes de Petri chamada de rede de Petri Parametrizada (PPNs) é proposta (GRACANIN, SRINIVASAN E VALVANIS, 1993) e aplicada para o planejamento e coordenação de sistemas inteligentes. A abordagem PPN usa uma organização hierárquica para tratar com a característica da complexidade da representação das redes, através da parametrização das transições e o uso de lugares generalizados ou parametrizados. O tempo é introduzido na PPNs em termo de um parâmetro. Um controlador de recuperação de erros é então descrito baseado nas redes de Petri parametrizadas temporizadas. Uma característica única de utilização das PPNs é que elas podem ser usadas para gerar uma planta abstrata de alto nível, com a ausência de informações completas, a qual pode ser mapeada dentro de uma planta especificada em um nível mais baixo da hierarquia (GRACANIN, SRINIVASAN E VALVANIS, 1994) . As propriedades estruturais do modelo do sistema PPN são verificadas baseadas em seu modelo de redes de Petri correspondente.

Chamas, Anneberg e Yaprak (CHAMAS, ANNEBERG & YAPRAK, 1993) propõe o conceito de redes de Petri temporizadas neurais, a qual é isomorfa com a estrutura de

redes neurais. O conceito de rede de Petri temporizada neural combina a nova orientação da rede neural com a capacidade de modelagem das redes de Petri. Uma grande vantagem na utilização da rede de Petri para modelagem (especialmente no caso neural) é a representação apropriada de eventos concorrentes e paralelos. As técnicas de redes neurais são aplicadas a redes de Petri, e o conceito de redes de Petri neurais vai além da definição básica das redes de Petri.

Uma ferramenta de modelagem baseada em redes de Petri com velocidade controlada continuamente (CSCP_N's) é introduzida (GHABRI & LADET, 1994). Podem ser usadas para modelar não somente sistemas de produção, mas também as interações existentes entre o nível de controle e a planta. A extensão das redes de Petri temporizadas com velocidade controlada continuamente é definida como o procedimento de se estabelecer o tempo necessário para o processo e o transporte na planta. Os autores discutem também as possibilidades para o uso desta ferramenta no controle de Sistemas de Eventos Discretos.

Xie (XIE, 1995) discute a questão da integração na modelagem de sistemas em larga escala usando uma abordagem modular de modelagem baseada em redes de Petri. É introduzida uma classe de redes de Petri chamada de redes Estendidas com Saída Controlada ou ECO nets. Estas redes são consistentes, livres e reversíveis. É mostrado que a integração de tais redes de Petri preserva suas propriedades.

Um modelo de rede de Petri em tempo real (RTPN) é proposto (HWANG & HO, 1995) para modelar a especificação das tarefas em tempo real no sistema em termos de eventos, ações, estados e restrições temporais. A operação básica de uma RTPN inclui identificação da transição executável, atualização das marcas, checagem das restrições temporais e cálculo das entradas/saídas compartilhadas. Um modelo de uma execução implementada em hardware é proposta para fornecer seu desempenho. O projeto detalhado do hardware dos eventos e dos módulos de estado no modelo de execução é descrito.

Ammar e Yu (AMMAR & YU, 1995) introduzem os conceitos e definições de redes de Petri Fuzzy Marking. Uma abordagem prévia sobre redes de Petri fuzzy foi desenvolvida para a especificação onde foram impostas restrições para as redes "safes", restringindo deste modo as habilidades de análise usando os métodos e ferramentas já conhecidos. redes de Petri Fuzzy Marking são apresentadas aqui como uma extensão das redes de Petri estocásticas na qual as marcas são números imprecisos representados por funções fuzzy. Um algoritmo para marcação fuzzy é apresentado. Aplicações na manufatura e controle inteligentes também são discutidos.

Yim, Lefort e Hebrard (YIM, LEFORT & HEBRARD, 1995) propõem uma nova ferramenta de modelamento gráfico chamada HyperNets, inteiramente relacionada com redes de Petri de alto nível. A semântica da HyperNets é baseada em funções e restrições teóricas. É associado um algoritmo não determinístico de semi decisão para analisar as propriedades de "correctness" e "completeness". Esse algoritmo também fornece as seqüências de disparo.

As redes de Petri coloridas são utilizadas por Arjona e Lopez (ARJONA & LOPES, 1996) como base de um método de projeto de tarefas completas em sistemas flexíveis de manufatura (FMS), que permite obter sistematicamente modelos de descrição das tarefas em redes de Petri coloridas em um nível alto de abstração. O método preserva o formalismo das CPNs sem impor restrições sobre o sistema modelado, e assim permite a modelagem de recursos compartilhados, ordens de serviço complexas e critérios de seleção das ordens.

Wieting (WIETING, 1996) apresenta uma metodologia de modelagem e simulação de sistemas híbridos chamada Hybrid High Level Nets (HyNets), a qual apresenta uma estrutura básica discreta e equações diferenciais algébricas, para representar a conduta de sistemas contínuos e o conceito de orientação ao objeto.

Flaus (FLAUS, 1996) também apresenta uma rede de fluxo híbrido, utilizada para modelar os aspectos discretos e contínuos de um sistema. Estas redes são baseadas em redes de Petri e em redes de fluxo contínuo, podendo ser usadas como uma extensão contínua e não linear das redes de Petri.

Na mesma linha de sistemas híbridos, Caradec e Prunet (CARADEC & PRUNET, 1997) propõe uma ferramenta de modelagem de sistemas híbridos baseada em redes de Petri coloridas chamada Colored Batches Petri Net (CBPN) para a simulação destes modelos.

Mohammad (MOHAMMAD et al., 1997-a) apresenta uma metodologia de modelagem e simulação de sistemas híbridos de manufatura utilizando uma extensão de redes de Petri chamada de rede de Petri Global (GPN – Global Petri Net). Utilizando as GPN os sistemas de manufatura discretos são modelados em dois níveis de abstração. Um nível de mais alta abstração é modelado na forma discreta por redes de Petri ordinárias e outro mais detalhado do seu comportamento, utilizando as GPNs.

Mohammad (MOHAMMAD et al., 1997-b) também apresenta uma ferramenta baseada nas GPNs, chamada de GPNSAT (Global Petri Net Simulation and Analysis Tool), a qual simula redes de Petri simples, temporizadas e globais e possui a

capacidade de analisar várias características de sistemas híbridos, tais como “boundedness”, estabilidade, controlabilidade e “liveness”.

2.2.4.4 - Análise das redes de Petri

Os trabalhos relativos a análise das redes de Petri para sistemas de manufatura podem ser divididos entre aqueles que se concentram em tratar suas características para solucionarem questões de controle e aqueles que se preocupam em realizar uma análise de desempenho dos sistemas de manufatura, onde o tempo é incluído na definição das redes de Petri (MOORE & BRENNAN, 1995).

Análise das características das redes de Petri

Na análise das características das redes de Petri, dois métodos têm destaque, são os métodos gráficos e construídos sobre os conceitos de árvore de alcançabilidade (isto significa que o método é dependente da marcação inicial e assim utilizado para se determinar *propriedades comportamentais*) e os métodos em matrizes que descrevem como os nós em um a rede de Petri estão interconectados (esta interconecção não é influenciada pela marcação inicial, tratando esta abordagem das *propriedades estruturais* de uma rede de Petri) (DESROCHERS & AL-JAAR, 1995), (MURATA, 1989).

A análise de várias propriedades qualitativas, incluindo *alcançabilidade*, “liveness”, “boundedness” e *reversibilidade* podem levar ao controle de um sistema de manufatura. Alcançabilidade se refere a um provável estado do sistema, “liveness” indica que o sistema é livre de “deadlock”, “boundedness” relata estouro de “buffers” e reversibilidade indica a habilidade do sistema para reiniciar a si mesmo (MOORE & BRENNAN, 1995).

O problema de “deadlock” foi bastante estudado em computação no contexto de sistemas em geral (PETERSON, 1981) (REISIG, 1985), e sistemas operacionais, (PETERSON & SILBERSCHATZ, 1985) , destes trabalhos, foram adaptadas (VISWANADHAM, et. al., 1990) quatro condições identificadas como necessárias para a ocorrência de *deadlock* nos sistemas de manufatura:

1. Exclusão mútua: quando um recurso está sendo usado por dois ou mais processos simultaneamente.
2. “No preemption”: um recurso em uso permanece em uso até que o processo que o está usando termine de executá-lo.

3. *“Hold and Wait”*: existe um processo que está prendendo pelo menos um recurso e está esperando para capturar outros recursos que estão atualmente sendo presos por outros processos.
4. *“Circular Wait”*: existe um conjunto $\{p_1, p_2, \dots, p_n\}$ de processos de espera tal que p_1 está esperando por um recurso que está preso por p_2 , e p_2 está esperando que está preso por p_3, \dots , e p_n está esperando por um recurso que está preso por p_1 .

Viswanadham et al. (VISWANADHAM, et. al., 1990) introduz o conceito de *“Deadlock Prevention”* e *“Deadlock avoidance”*, onde:

“Deadlock Prevention”: consiste em falsear uma ou mais das condições apresentadas acima, necessárias para a existência de *“deadlock”*. Isto é feito pelo uso de políticas de alocação dinâmica de recursos para eliminar completamente os *“deadlocks”*. O gráfico de alcançabilidade é a principal ferramenta de análise da prevenção dos *“deadlocks”*.

“Deadlock Avoidance”: consiste em falsear uma ou mais das condições necessárias para *“deadlock”* rastreando o estado atual e as condições futuras possíveis. Esta é uma abordagem dinâmica que utiliza uma política de se prever se no próximo passo há a possibilidade de *“deadlocks”*.

Cho et. al (CHO et. al, 1995) aborda o *“deadlock”*, considerando que o sistema já está travado. A partir da situação de *“deadlock”*, um conjunto de atitudes devem ser tomadas no sentido de se encontrar onde e o que causou este estado e como resolver o problema.

Jang et. al (JANG et. al, 1997) descreve o projeto de uma célula de manufatura, onde é realizado o projeto da arquitetura de controle, das seqüências síncronas e assíncronas da célula, a análise de prevenção de *“deadlocks”* para as redes de Petri e um estudo da recuperação de erros do sistema.

Jeng (JENG, 1997) utiliza a propriedade de reversibilidade para analisar a recuperação de erros em sistemas de manufatura, esta teoria tem sido proposta para sistematizar uma classe de redes de Petri que possui verificação de propriedades estruturais, tais como *“structural boundedness”* e *“structural liveness”*. Um algoritmo de checagem de *“liveness”* é utilizado para se checar a reversibilidade, estabelecendo assim a recuperação de erros do sistema.

Ezpeleta (EZPELETA, 1995) realiza um estudo das propriedades estruturais de uma classe especial de rede derivada das redes de Petri e propõe um controlador em

tempo real para se evitar os “deadlocks”, onde a estratégia é se acrescentar novos elementos na rede para restringir a ocorrência dos “deadlocks”. Utilizando esta política, Ezpeleta & Colom (EZPELETA & COLOM, 1997) estudaram uma metodologia de modelagem, aplicada a sistemas flexíveis de manufatura (FMS) onde o modelo é representado em redes de Petri coloridas (CPN).

Santos (SANTOS, 1998) adota uma solução para evitar os “deadlocks” no sentido de agregar características de robustez e eficiência ao sistema. Para isto é proposto um procedimento que estabelece um conjunto mínimo de compensadores para serem adicionados no controle do sistema, assegurando assim a não ocorrência de “deadlock”.

Outras abordagens, no sentido de obter-se uma solução para evitar-se os “deadlocks”, adotando estratégias em tempo real podem ser verificadas em Banazak & Krogh (BANAZAK & KROGH, 1990), Xing et. al (XING et. al, 1996) e Hasegawa (HASEGAWA, 1996).

Análise de desempenho das redes de Petri

No que diz respeito a análise de desempenho, a adição do tempo nas redes de Petri permite uma análise temporal dos sistemas de manufatura. A temporização foi introduzida em meados dos anos 70 e início dos anos 80 por Ramchandani (RAMCHANDANI, 1973), Merlin (MERLIN, 1976), Florin e Natkin (FLORIN & NATKIN, 1982), Molloy (MOLLOY, 1982), além de outros. Existem duas classes principais de redes de Petri temporais: TPNs – timed Petri nets; e SPNs – stochastics Petri nets. TPNs são redes de Petri de transições com tempo determinísticas e SPNs utilizam transições com tempos aleatórios. Outra classe, as redes de Petri estocásticas generalizadas (GSPN) incorporam tanto as transições SPN, como as transições imediatas.

A seguir são mostrados alguns trabalhos que tratam da análise de desempenho de sistemas com o uso das redes de Petri:

Cohen et. al (COHEN et. al, 1985) utilizaram as redes de Petri temporizadas para modelar e analisar sistemas de produção e algumas regras de controle, tais como *FCFS* (*First Come First Serve*). Também demonstraram o uso das TPNs para se determinar os ciclos de processos, a utilização das máquinas e os recursos críticos de um sistema de manufatura.

Abraham e Sciomachen (ABRAHAM & SCIOMACHEN, 1987) usaram as SPNs para modelar um AGV em um sistema flexível de manufatura, mostrando um bom exemplo do uso de SPNs como uma ferramenta de planejamento e projeto de hardware de um sistema.

Archetti et. al (ARCHETTI et. al, 1987) usaram SPNs para modelar e analisar uma linha de transferência onde as máquinas estão sujeitas a falhas. É também avaliado o desempenho de várias políticas de produção quando a falha ocorre.

Viswanadham e Narahari (VISWANADHAM & NARAHARI, 1988) mostraram que as GSPNs fornecem uma boa estrutura de análise de desempenho para sistemas de manufatura automatizados, utilizando para isto duas estruturas: uma célula com vários robôs de manuseio de material e um sistema de manufatura flexível simples com três máquinas e dois tipos de peças.

Hillion e Proth (HILLION & PROTH, 1989) utilizam uma classe especial de redes de Petri, chamada *Timed Event-Graphs*, para modelar e analisar sistemas *job-shop*. O modelo permite a avaliação do desempenho do sistema no estado estacionário e de processos de produção cíclicos e determinísticos, isto é, com tempos de processamentos fixos.

Zhou e Dicesare (ZHOU & DICESARI, 1990) desenvolveram a análise de desempenho de um sistema de manufatura que utiliza recursos em comum utilizando redes de Petri estocásticas (SPN), tratando do problema de "deadlock" de acordo com o número de peças presente na célula durante um certo período de tempo.

Wongtaladkown e Chen (WONGTALADKOWN & CHEN, 1990) utilizam o "Real Time Petri Net Simulator", que é um software de edição e simulação de redes de Petri, para analisar o desempenho de um FMS. O simulador possibilita a modelagem do sistema em vários níveis de abstração, dependendo da identificação dos objetivos a serem atingidos.

Watson III e Desrochers (WATSON III & DESROCHERS, 1991) utilizaram as GSPNs para a modelagem de sistemas de manufatura, utilizando funções temporais não exponenciais. A verificação do modelo foi realizada utilizando uma comparação com o desempenho de sistemas flexíveis de manufatura (FMS).

Kelling et. al (KELLING et. al, 1996) descrevem um modelo que foi estabelecido para projetar sistemas de controladores distribuídos (DCS). Vários modelos em redes de Petri estocásticas (SPN) para esquemas diferentes de comunicação foram desenvolvidos para se encontrar o desempenho dos sistemas distribuídos. O modelo foi

implementado utilizando a ferramenta de software *TimeNet*, a qual permite a modelagem de tais sistemas combinando uma interface gráfica com métodos de simulação.

Cavaliere et. al (CAVALIERI et. al, 1997) propõem uma metodologia para modelar e analisar o desempenho de FMS baseado no uso de modelos de redes de Petri estocásticas (SPN). É destacado o uso de transições probabilísticas para modelar o comportamento de cada atividade desenvolvida no sistema de manufatura.

Katsumoto et. al (KATSUMOTO et. al, 1997) utilizam um método de estimativa da qualidade, custo e data de entrega para sistemas de manufatura automatizados. O método consiste de um modelo de projeto e um modelo de processo, onde o modelo de projeto trata das etapas de execução do projeto e o modelo de processo inclui uma série de atividades dentro das etapas de projeto, cada uma com atividades específicas de desenho, codificação, revisão, teste e *debug*, utilizando GSPNs.

Wang e Wu (WANG & WU, 1998) utilizam as redes de Petri coloridas temporais orientadas ao objeto - CTOPN (Colored Timed Object Petri Net), para a modelagem de sistemas de manufatura automatizados. Um procedimento de modelagem com as CTOPN foi desenvolvido, assim como um algoritmo de prevenção de “*deadlocks*” e abordagens de sua aplicação na programação (“*scheduling*”) e controle de um sistema de manufatura automático.

Kulp e Dicesare (KULP & DICESARE, 1998) propõem uma metodologia para avaliar o desempenho de vários modelos de redes de Petri estocásticas, incluindo redes de Petri coloridas estocásticas (SCPN). A formulação utiliza a simulação para gerar uma árvore de alcançabilidade de uma rede de Petri e realizar a análise de desempenho.

Deste modo, a pesquisa realizada mostra que as redes de Petri estão crescentemente sendo usadas para a modelagem de sistemas de manufatura automáticos. Para análise e avaliação de desempenho as extensões baseadas em tempo (Timed PNs, Stochastics PNs, e Generalized Stochastics PNs) têm sido bastante aplicadas.

3 - PROPOSTA DE UM AMBIENTE DE PROJETO DE INTERTRAVAMENTOS UTILIZANDO REDES DE PETRI

A proposta deste trabalho trata de uma sistemática, para que o projetista possa desenvolver o projeto do intertravamento de máquinas que compõe um sistema de manufatura. A implementação dessa sistemática será realizada com a criação de um ambiente para o projeto do intertravamento onde são descritas as técnicas e ferramentas utilizadas, tais como sistemas especialistas, redes de Petri, linguagem de programação CLP, entre outros.

3.1 – Especificação funcional

O desenvolvimento do projeto de intertravamento de um sistema de manufatura pode ser dividido nas etapas de projeto e de implementação.

A etapa de projeto envolve a análise e definição das necessidades, o projeto do sistema de controle e o projeto e desenvolvimento do programa. A etapa de implementação pode também unir a operação e a manutenção do sistema de intertravamentos (MIYAGI, 1996).

Uma sistemática utilizando um ambiente de projeto de intertravamentos é proposta para que os problemas citados no capítulo 1 possam ser sanados para o projeto de intertravamento de um sistema de manufatura tanto na fase de projeto como de implementação, além de contemplar a integração entre os elementos das diversas áreas contidas no projeto (elétrica, mecânica, produção, programação, administração).

A sistemática deve promover uma melhoria no ciclo de vida do projeto e do desenvolvimento do programa de intertravamento de um sistema de manufatura. Em

termos da produção do programa, este ambiente pode apoiar a representação do projeto, a análise do projeto e a sua transformação em código de linguagem CLP.

A Figura 3.1 descreve a sistemática proposta para o projeto de intertravamentos em sistemas de manufatura identificando as ações a serem executadas nas fases de projeto e implementação.

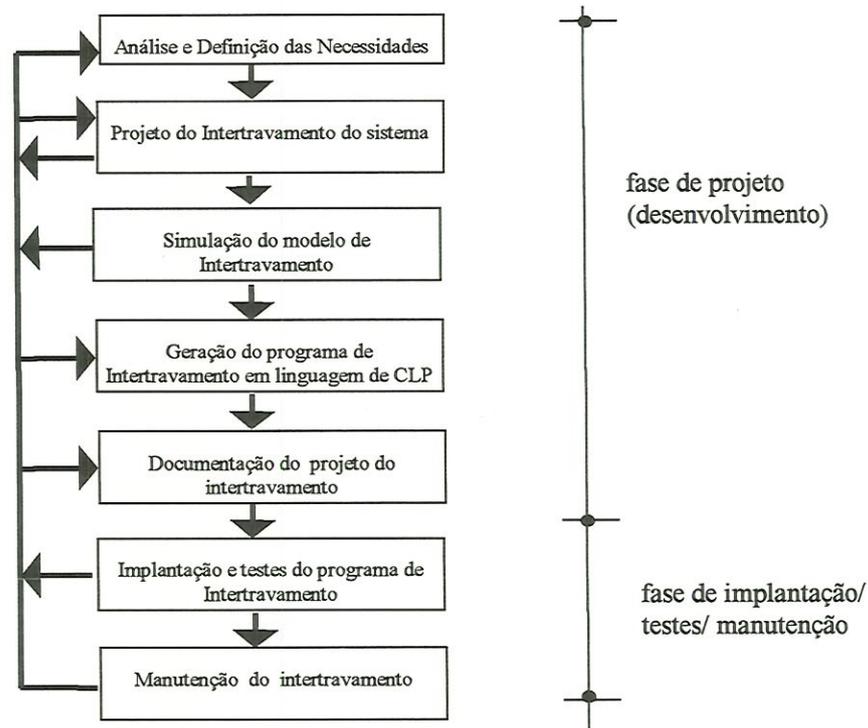


Figura 3.1 - Diagrama de blocos do ciclo de vida do projeto de Intertravamentos.

Assim, os vários passos compreendidos nestas etapas podem ser descritos e relacionados.

Passo 1: Análise e definição das necessidades.

Nesta fase todas as informações necessárias para o projeto tais como: tipo de máquina; tipo de peça a ser fabricada; esquema elétrico; projeto mecânico; esquema pneumático; tipo de comunicação existente no sistema; normas de segurança a serem adotadas; e viabilidade financeira, devem ser reunidas. As várias partes envolvidas no projeto devem tratar o sistema sobre a mesma ótica, isto é, em um nível de abstração adequado, integrar todos os elementos do sistema.

A partir destas informações, o projetista pode sintetizar as necessidades do projeto. Assim, estabelecer uma descrição preliminar do sistema a ser projetado em uma representação comum a todas as partes envolvidas.

Passo 2: Projeto do Intertravamento.

Neste passo, o projetista, a partir da modelagem preliminar do sistema, pode incorporar os intertravamentos necessários, utilizando os seus elementos básicos, tais como entradas, saídas, contadores, temporizadores, etc.... Com a elaboração dos intertravamentos utilizando estes elementos básicos, a inclusão de novos elementos de controle, para a execução das necessidades desejadas pode levar o processo novamente ao passo 1, por exemplo devido a viabilidade financeira.

Nesta fase os modelos básicos de intertravamentos e até blocos de funções dedicadas, como partida de motores, emergência, etc..., podem estar disponíveis na forma de arquivos, podendo ser usados de uma forma recuperativa, para facilitar o projeto.

Passo 3: Simulação do modelo do Intertravamento.

Após a montagem do modelo, o projeto de intertravamento deverá ser simulado. Nesta ocasião, o modelo contempla todos os elementos de natureza diferentes envolvidos no projeto do sistema, integrada com os blocos dos elementos de intertravamento. Assim esse modelo pode fornecer os estados possíveis que o sistema projetado pode assumir. Além disso, é possível de se analisar o desempenho do sistema verificando o seu comportamento no tempo.

A simulação tanto da estrutura como de desempenho do modelo pode levar o projetista a retornar à fase de projeto do intertravamento e até a de análise e definição das necessidades. Assim como assegurar que o projeto do intertravamento possa minimizar os erros na fase de implementação.

Passo 4: Geração do programa de intertravamento em linguagem CLP.

Nesta fase o modelo editado e simulado é convertido para uma linguagem CLP. Os elementos básicos e blocos de funções de intertravamentos, especificados

anteriormente são identificados no modelo do sistema e convertidos. Os outros elemento ou funções de natureza diferente, representados no modelo em níveis de abstração adequados e que foram de fundamental auxílio na visualização e simulação, estão ligados ao programa CLP na forma de sinais elétricos de entrada e saída.

Dessa forma, a geração do programa CLP pode diminuir os custos e os tempos envolvidos na implementação do programa de intertravamento no sistema desejado.

Passo 5: Documentação do projeto do intertravamento.

Durante os passos descritos anteriormente na fase de projeto, foram geradas:

- a descrição funcional do modelo, entre todas as partes envolvidas no projeto;
- o modelo completo de funcionamento do sistema desejado;
- os relatórios de simulação contemplando todas ou as principais situações (estados) possíveis que o sistema pode assumir;
- o programa CLP dos intertravamentos e sua lista de pontos de entradas e saídas a serem implementados.

Assim, a documentação preliminar do projeto de intertravamento do sistema será reunida com a documentação da fase de implementação (histórico da implantação, descrições de manutenção, etc...), podendo a primeira sofrer alterações de acordo com a implementação e a manutenção do sistema.

Passo 6: Implantação e testes do programa de intertravamentos.

Na etapa de implantação e testes o programa de intertravamentos em linguagem CLP é transferido para a memória do equipamento de controle e as condições iniciais de seu funcionamento são obtidas no modelo de intertravamento projetado, facilitando assim a partida (“*start-up*”) do sistema.

As alterações nesta fase podem ser simples, em termos de ajustes ou a inclusão de novos dispositivos. Os ajustes realizados nesta fase deverão ser incorporados no modelo, pelo editor do modelo (passo 2). A análise da simulação pode ser realizada ou não dependendo do ajuste. O programa do CLP é atualizado e o ciclo de projeto é retomado.

Uma mudança mais significativa leva o projetista novamente a fase de análise das necessidades. Retomando desta forma o ciclo de projeto por completo.

Passo 7: Manutenção do intertravamento

A manutenção do intertravamento envolve os ajustes no programa CLP na fase de implementação e posteriormente ao seu funcionamento. É executada devido as adaptações de campo dos equipamentos (por quebra ou substituição) podendo haver até a inclusão ou retirada de dispositivos e equipamentos do programa de controle do sistema dependendo do tipo de produto.

Uma alteração na etapa de manutenção implica na atualização do modelo completo do sistema, sua simulação para a validação da mudança no sistema e nos intertravamentos e a nova geração do programa CLP do sistema. Desta forma a documentação se mantém fiel ao sistema atual implementado.

A utilização desta sistemática tem por objetivo fornecer as condições necessárias para o projeto, implantação e manutenção de sistemas de intertravamento. A utilização de uma ferramenta de modelagem matematicamente consistente, que contemple a integração entre os diversos elementos de natureza diferente, junto com a simulação dos seus possíveis estados e a geração do programa CLP, poderá minimizar os problemas obtidos pela deficiência de formalização do projeto, fornecendo soluções rápidas e consistentes para o projeto e também para a manutenção de tais sistemas.

3.2. – Proposta de implantação da sistemática

A implantação dessa sistemática de projeto do intertravamento será tratada na forma da criação de um ambiente de projeto de intertravamento que possa contemplar todos os passos descritos no item anterior.

Assim, a implementação da sistemática proposta utiliza um ambiente auxiliado por computador que irá fornecer ao usuário a capacidade de sistematizar e integrar as atividades de análise e definição das necessidades do projeto, modelagem dos intertravamentos, simulação do modelo, geração do programa de intertravamento, implantação, testes, documentação e manutenção do sistema de manufatura.

Para atingir este objetivo o ambiente será baseado no uso de uma biblioteca de modelos e utiliza a rede de Petri como ferramenta de modelagem do sistema e do intertravamento.

A rede de Petri é uma das ferramentas que se encaixa no perfil da ferramenta descrita na sistemática, por ser simples (poucos elementos), facilitar o entendimento do sistema pelos profissionais envolvidos, possuir a capacidade de abstração dos vários elementos do sistema e permitir através de suas características matemáticas, a análise e simulação de consistência estrutural (alcançabilidade, vivacidade, reversibilidade, etc.) e de desempenho, fornecendo condições ao projetista para a sua validação e posterior aplicação.

A elaboração de um ambiente de projeto, utilizando o ambiente "Windows", e as redes de Petri, para a implantação da sistemática de projeto de intertravamento descrita, irá ordenar e padronizar a programação fornecendo ao projetista um programa mais eficiente e consistente traduzido para a linguagem de CLP.

Na fase de implantação, testes e manutenção, o ambiente agiliza o processo de documentação das alterações que possam vir a surgir, facilitando a atualização e mantendo a documentação do sistema fiel a sua implementação no chão de fábrica.

Neste contexto, a criação de um ambiente de projeto de intertravamento auxiliado por computador, com uma interface amigável, uma ferramenta de modelagem simples e matematicamente consistente, além do uso de simulação, poderá minimizar os problemas obtidos pela deficiência de formalização fornecendo soluções rápidas e consistentes para o projeto e também para a manutenção de tais sistemas.

Descrição do ambiente de projeto de intertravamentos

Utilizando a sistemática de projeto do intertravamento, o projetista pode desenvolver e gerar o intertravamento a partir da especificação do sistema de uma forma recuperativa e sistemática. Para isso é utilizado um ambiente ("*shell*") de integração de vários módulos como descrito na Figura 3.2.

O ambiente de projeto, chamado Ambiente de Desenvolvimento de Intertravamento Assistido por Computador (ADIAC) ilustrado na Figura 3.2 é composto por vários módulos, que descrevem os passos da sistemática proposta anteriormente.

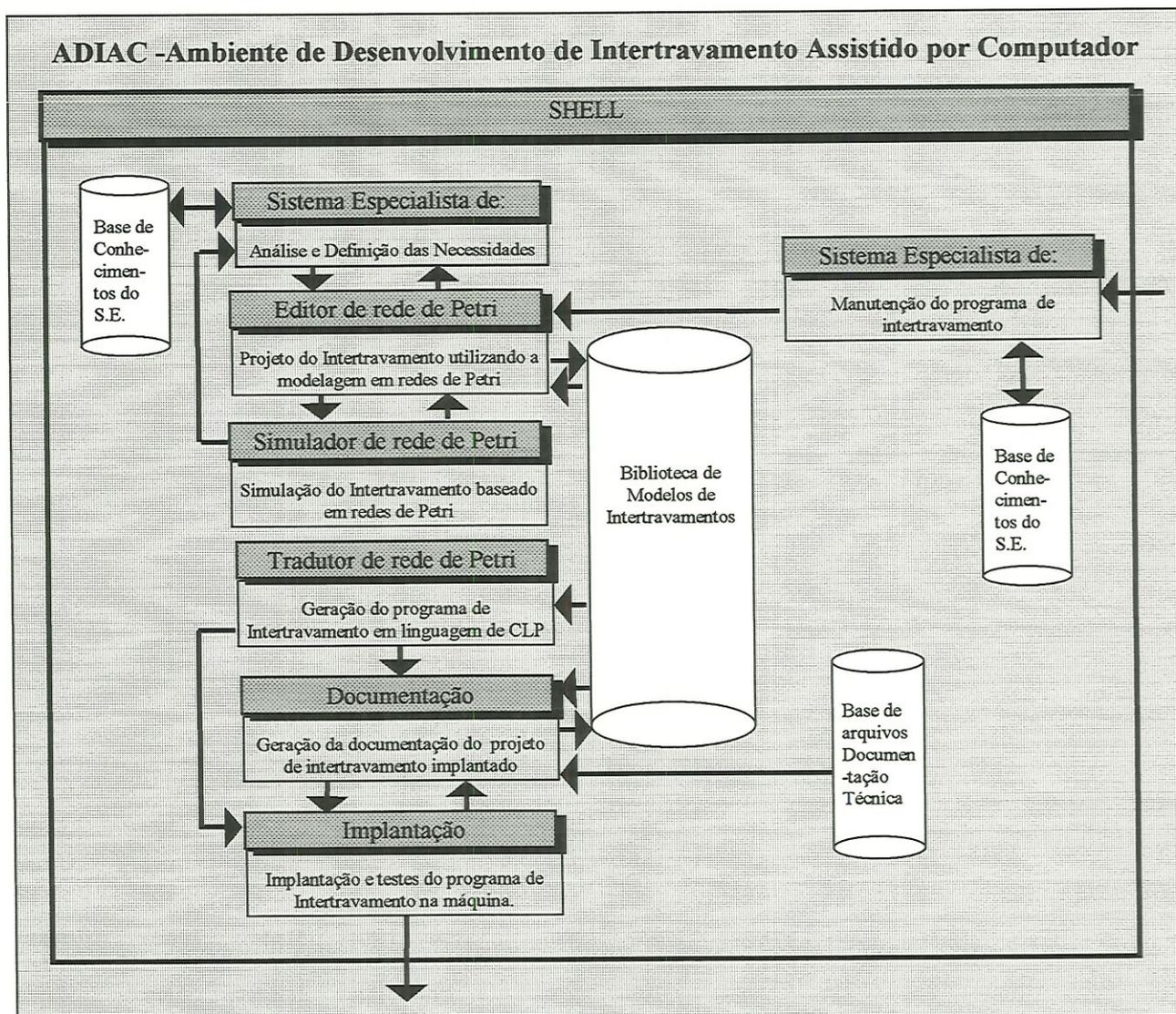


Figura 3.2 – ADIAC – Ambiente de desenvolvimento de intertravamento assistido por computador.

Para a análise e definição das necessidades é utilizado um sistema especialista o qual irá inferir uma descrição inicial do sistema a ser projetado utilizando uma base de conhecimentos própria. Esse sistema especialista fornece uma descrição básica do sistema a ser projetado, evitando o esquecimento de pontos relevantes para o projeto, além de indicar os blocos básicos, em redes de Petri, a serem utilizados para a modelagem do sistema de intertravamento.

Um editor e um simulador de rede de Petri podem modelar o sistema de intertravamento a partir dos elementos básicos e de blocos de funções descritos na forma de redes de Petri. Esses modelos de blocos de intertravamentos estarão disponíveis na forma de arquivos em uma biblioteca, que inclui inclusive o modelo do projeto que está sendo criado, a qual pode ser consultada pelo editor.

Esses elementos e blocos são selecionados pelo sistema especialista na fase de definição e análise das necessidades do sistema, permitindo a montagem de um projeto modular para o sistema desejado.

Assim, as atividades de análise e definição das necessidades do projeto serão baseadas em uma base de conhecimentos, inferida por um sistema especialista, para a especificação dos elementos da máquina e da quantidade de pontos (entradas/saídas) a serem controlados. Também um sistema especialista é utilizado para a análise inicial resolução de problemas e de manutenção dos intertravamentos.

O simulador de rede de Petri deve permitir que o modelo editado seja simulado utilizando as análises das características matemáticas das redes de Petri, tais como alcançabilidade e técnicas como matriz de incidência.

Além disso, é possível de se analisar o desempenho do sistema verificando o seu comportamento no tempo. Para isto é utilizada uma estimativa de tempo determinístico (redes de Petri temporizadas) para cada ação, sobre as seqüências alternativas do sistema, isto é, pode-se analisar o tempo de ciclo e a recuperação de uma falha no sistema através da análise do fluxo de marcas.

O tradutor de rede de Petri é o responsável pela geração do programa de intertravamento em linguagem CLP. Nesta fase é realizada uma conversão do modelo projetado em redes de Petri para uma linguagem CLP. Esta conversão pode ser realizada automaticamente através de um algoritmo de conversão, "softwares" dedicados, etc., ou por inspeção do modelo de controle da máquina.

A documentação do "software" de intertravamento reúne padrões de um banco de documentação técnica e utiliza o projeto armazenado na biblioteca de intertravamentos, fornecendo a descrição das características do projeto e das atualizações realizadas; a descrição do funcionamento do sistema; a lista de pontos de entrada, saídas e "flags"; a síntese da análise das necessidades; o modelo do sistema em redes de Petri; os relatórios da análise do desempenho do modelo; o programa CLP gerado; o histórico da sua implantação; e os registros de descrições de manutenção.

O processo de implantação e testes irá fornecer uma maneira sistemática da partida ("*start-up*") do sistema elaborando a partir da marcação inicial do modelo em redes de Petri. Os ajustes realizados nesta fase deverão ser incorporados no modelo pelo editor de redes de Petri.

Do mesmo modo que a análise e definição das necessidades, um sistema especialista irá inferir um diagnóstico inicial do sistema utilizando uma base de

conhecimentos própria dedicada a falhas e recuperação do sistema. Esse sistema especialista irá auxiliar na recuperação rápida do sistema a falhas que poderão acontecer.

Também serão realizadas a simulação de falhas aleatórias no hardware individual das máquinas e a simulação de falhas aleatórias no sistema como um todo.

Assim a elaboração de um ambiente de projeto irá ordenar e padronizar a programação do intertravamento, fornecendo ao projetista um programa mais eficiente e consistente, e com a documentação necessária.

O capítulo 5 trata da implementação deste ambiente voltado para a utilização do projeto de máquinas, a fim de se validar a sistematização do projeto do intertravamento proposta. A “shell” projetada é descrita e todos os seus elementos detalhados.

A implementação do projeto de intertravamento de um Torno CNC é descrito no capítulo 6 onde são detalhados os blocos e as funções implementadas, assim como os seus resultados, e no anexo B é registrada a programação CLP gerada pelo ambiente de projeto.

4 - MODELAGEM EM REDES DE PETRI DE SISTEMAS DE INTERTRAVAMENTO

Vários métodos de representação gráfica como GRAFCET (DAVID, 1995) e o SFC (IEC, 1992), ambos normalizados como padrões internacionais, surgiram para melhorar a legibilidade dos programas para CLP. Outras representações como o Mark Flow Graph (MFG) (MIYAGI, 1996), C-nets (MURATA et. al., 1986) e K-nets (NAGAO et al., 1993) podem ser executadas diretamente de um computador dedicado.

No entanto a maioria se limita a apenas um modelo para especificação de controle seqüencial de um sistema, não se preocupando com o formalismo matemático necessário para a análise e simulação do controle dos intertravamentos.

O uso das redes de Petri na taxonomia de controladores programáveis será adotado com o objetivo de se obter um modelo da lógica de controle do intertravamento dos sistemas automáticos de manufatura, compostos por CLPs. Sendo este modelo baseado em uma ferramenta com capacidade de análise matemática, este pode fornecer capacidade de análise tanto comportamental como temporal.

Neste capítulo serão apresentados os elementos do CLP para a modelagem dos intertravamentos em redes de Petri.

A análise matemática pode fornecer de forma consistente um programa de controle para o equipamento CLP, eliminando tempo e custo da implantação do sistema.

As características analisadas como limitabilidade, reversibilidade e *deadlocks* do sistema, são de grande importância para a avaliação do funcionamento do modelo (ZHOU & DICESARE, 1993).

Os *deadlocks* acontecem quando processos simultâneos encontram-se travados de tal modo que eles não possam ser completados (D'SOUZA & KHATOR, 1994).

A análise de *deadlocks* nos modelos de redes de Petri será realizada através do tratamento *off-line* do programa CLP para que este possa ser implementado minimizando o número de erros possíveis (VISWANADHAM, NARAHARI & JOHNSON, 1990).

As medidas de desempenho de sistemas automáticos de manufatura, baseados em redes de Petri, podem ser avaliadas por simulação utilizando redes de Petri temporais como Timed PNs (ZUBERECK, 1980) (RAVICHANDRAN & CHAKRAVARTY, 1986) , Stochastics PNs (VISWANADHAM & NARAHARI, 1988) (HATONO, YAMAGATA &

TAMURA, 1991), e Generalized Stochastics PNs (WANG & HAFEEZ, 1994) (AL-JAAR & DESROCHERS, 1990).

As redes de Petri a serem utilizadas nesta proposta são as redes de Petri *Lugar-Transição*, devido ao seu poder de representação de todos os elementos básicos de um CLP (operações booleanas, contadores, comparadores e temporizadores) sem exceder a característica ordinária de representação destas redes. Pois as extensões geradas a partir de redes de Petri são em sua maioria baseadas em redes de Petri condição-evento (GRAF CET, SFC e MFG) não possuindo facilidade para a representação gráfica de contadores e comparadores sem perder tais características.

A sistemática a ser desenvolvida utiliza também uma ferramenta de simulação que incorpora atributos de redes de Petri temporizadas, para realizar as análises de desempenho do modelo.

Nas próximas seções são apresentados os modelos dos elementos de programação CLP em redes de Petri e a forma de conversão deste modelo para a linguagem CLP.

4.1 - A associação dos modelos de redes de Petri com a linguagem de programação CLP

Recentemente o CLP tornou-se parte integrante de sistemas de controle industrial devido a sua estrutura rígida e alta capacidade de processamento (TAHOLAKIAN & HALES, 1997).

Os CLPs possuem diversas características que justificam seu uso difundido no controle industrial (TROY & McQUEEN, 1996):

- Alto Nível de confiabilidade: tipicamente 24 horas de operação por dia;.
- Alta velocidade de varredura para controle de processo em tempo-real;
- Grande número de pontos de entrada/saída para os estados de controle;.
- Manutenção de software a nível de chão de fábrica;
- Flexibilidade para modificar o software durante a operação de fábrica, por exemplo, substituir itens da instalação ou fazer consertos de emergência.

O diagrama "*Ladder*", desenvolvido em 1969, é a linguagem predominante para se programar CLPs (GAYMAN, 1988) (OTTER, 1988). A linguagem "*Ladder*" é expressa

tipicamente na forma gráfica e assemelha-se a diagramas de lógica elétrica, utilizados por eletricitistas para circuitos de relés eletro-mecânicos. Os componentes primários de um programa de lógica “*Ladder*” são os contatos, que representam os sinais de entrada no CLP, as bobinas, que representam sinais de controle de saída do CLP.

Outras linguagens de CLP (IEC 1131-3) também foram padronizadas pela Comissão de Internacional de Eletrotécnica (IEC 1992) entre elas podemos citar: lista de instrução (IL), texto estruturado (ST), diagrama de bloco de função (FBD), Funções de Gráficos Seqüenciais (SFC) e finalmente diagramas “*Ladder*” (LD). Porém nenhuma das cinco linguagens sugeridas descreve uma sistemática de projeto, e também não são aceitas em unanimidade, exceção feita aos diagramas “*Ladder*” (TAHOLAKIAN & HALES, 1997).

A associação dos modelos de redes de Petri com a linguagem de programação CLP visa a modelagem de sistemas automáticos baseados em CLPs, isto é, a representação de um sistema a ser automatizado descrito na forma gráfica, em uma representação de fácil entendimento entre as várias partes envolvidas no projeto, facilitando sua especificação e alteração.

O projetista passa a modelar o sistema de forma que a abstração das redes de Petri não fiquem restritas aos elementos de um CLP, necessários para a execução de um automatismo. São eles: entradas, saídas, sinalizadores (“*flags*”), comparadores, temporizadores e contadores (IEC-1131-3). Estes elementos são determinados utilizando os *lugares* e as *transições* individuais ou através de agrupamentos.

O CLP funciona de forma cíclica, onde há a leitura das entradas, o processamento dos comandos e a atuação nas saídas. Assim, a modelagem do automatismo, através da rede de Petri, pode ser realizado de modo que todas as equações lógicas representadas pelas *transições* sejam analisadas dentro de um ciclo do CLP.

A utilização da rede de Petri lugar - *transição*, descrita no capítulo 2, permite um detalhamento maior em relação as redes condição evento, bastante usadas para a representação dos automatismos (COURVOISIER, 1983) fornecendo ao projetista uma maior facilidade na utilização de um número maior de recursos das linguagens de CLP (IEC 1131-3), tais como temporizadores, contadores e comparadores, que não eram explicitamente representados anteriormente.

Utilizando as características comportamentais das redes de Petri, tem-se que o modelo construído para o sistema deve ser livre de “deadlocks” (“live”), os conflitos devem ser dirigidos pelo projetista, se o sistema for composto por apenas lógicas E e OU

(sem os elementos como o contador, comparador e temporizador) é limitado e podem ser reversíveis (ZHOU & DICESARE, 1993).

Nas associações das redes de Petri com os componentes da linguagem CLP têm-se os *lugares* representando um estado do sistema e pode corresponder, em termos de controle, a sinais de *entradas externas* (característico de entradas externas de CLP) ou sinalizadores internos (“flags”). Pode ainda integrar elementos específicos da lógica de controle como os contadores e pode fazer parte da lógica do modelo sem representar nenhum elemento. A capacidade de cada um destes *lugares* é especificada de acordo com a sua função na representação do elementos do CLP, sendo que quando não indicada a capacidade é infinita. Os sinais de entrada e os sinalizadores podem ser ou não negados, quando negado a marca é colocada indicando que o sinal está desativado. Sua identificação é dada através do símbolo “~”. Estes atributos podem ser descritos utilizando o formato de identificação dos *lugares* descrito na tabela 4.1.

Tabela 4.1 – descrição dos *lugares* e das *transições*.

$p_i(x;y)$	
X	I x.x → entrada externa; F x.x → sinalização (“flag”) interno; IBx → “byte” de entrada externa IWx → “word” de entrada externa FBx → “byte” de “flags” FWx → “word” de “flags”
Y	Número atual de marcas do lugar
$t_i(x;y)$	
x	= Q x.x → aciona a saída externa; e/ou S Q x.x → ativa (“set”) a saída externa; e/ou R Q x.x → desativa (“reset”) a saída externa; C x → contador; CMPx → comparador; Tx → temporizador;
y	Tempo utilizado na análise de desempenho

As *transições* representam as ações. Elas são constituídas de uma lógica funcional a partir da álgebra Booleana, que analisam os *lugares* de entrada e realizam atuações no sistema da mesma forma que os comandos de saídas de um CLP. As *transições* podem também fazer parte de elementos como contadores ou fazem parte da lógica do modelo, sem tratar de nenhum elemento.

Os *arcos* ou *arcos orientados* são os elementos de ligação entre os *lugares* e as *transições*. Os arcos podem ter pesos, permitindo que uma *transição* seja ativada

somente se o número de marcas no *lugar* que está ligada a ela for igual ou maior do que o peso do arco, arcos de peso unitário não precisam ser indicado. Após o seu disparo se o arco de ligação da *transição* com o *lugar* possui peso w o *lugar* receberá w marcas. Os *arcos* são os responsáveis pelo fornecimento da direção do fluxo que as marcas (“*tokens*”) terão através da rede.

A partir destes elementos, do funcionamento do CLP e da definição das redes de Petri, pode-se realizar algumas associações da representação de redes de Petri com linguagens de programação CLP, que podem ser utilizadas em uma sistemática de modelagem visando a programação de automatismos, que é baseada nos programas padronizados por normas internacionais para CLPs. Esta representação de modelagem será descrita a seguir.

4.1.1 - Lógica E

A lógica E é uma das principais lógicas utilizadas para intertravamentos, nela uma saída pode ou não ser acionada, dependendo da verificação das condições das entradas a ela ligadas. Estas entradas estão ligadas, como descrito na Figura 4.1.

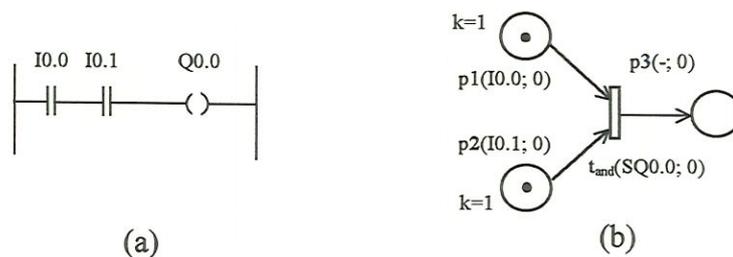


Figura 4.1 – Representação da Lógica E (AND).

A Figura 4.1 (a) mostra a representação no programa de controle de um CLP em linguagem “*Ladder*” da lógica E, para o intertravamento das redes de Petri nos sistemas de manufatura, onde a bobina Q0.0 não será energizada se as entradas I0.0 e I0.1 não estiverem ativadas ou o lugar p3 tiver uma marca. O modelo em rede de Petri desta lógica é mostrado na Figura 4.1(b) onde a *transição* pode disparar.

A análise da representação em redes de Petri (Figura 4.1 (b)), é realizada sobre a *transição*, isso é, a lógica E é caracterizada através dos *lugares* de entrada da *transição* t_{and} e do comando de saída descrito nesta *transição*. Estes *lugares* podem ser entradas externas e/ou sinais de Flags. Os *lugares* que são entradas externas são caracterizados pelo arco de peso unitário e a alteração da marcação será executada de acordo com uma mudança de estado externa do sistema. *Lugares* de saída jamais poderão ser entradas externas.

Nas *transições*, são realizadas as ações do sistema, isto é, as saídas são atuadas. Os comandos são determinados na *transição*, correspondendo a comandos de “set” (ativa), “reset” (desativa), aciona (=), os quais irão atualizar diretamente uma tabela de saída do CLP. Assim a *transição* para a representação da lógica E só pode receber *arcos* e *lugares* de capacidade e peso unitário, os quais correspondem a entradas, “flags” e *lugares* de seqüência.

Utilizando os símbolos abaixo para os pré-conjuntos e pós conjuntos, onde F é o conjunto de todos os arcos definidos anteriormente, temos (MURATA, 1989):

- t = {p | (p,t) ∈ F} = conjunto de *lugares* de entrada de t;
- t• = {p | (t,p) ∈ F} = conjunto de *lugares* de saída de t;
- p = {t | (t,p) ∈ F} = conjunto de *transições* de entrada de p;
- p• = {t | (p,t) ∈ F} = conjunto de *transições* de saída de p;

Pode-se assim, restringir os *lugares* e as *transições* para a representação da lógica E, da seguinte forma:

- ti → I x.x; F x.x; - ; k = 1
- ti• → F x.x, - ; k = 1
- pi• = •pi → = Q x.x; S Q x.x; R Q x.x;

Onde x.x significa valores de 0 a 7 para x.

4.1.2 - Lógica OU

Na lógica OU, uma saída é acionada por sinais de entrada combinados em paralelo. A Figura 4.2 (a) ilustra um exemplo onde a bobina Q0.0 é acionada se qualquer um dos sinais I0.0, I0.1 ou F0.2 estão ativados.

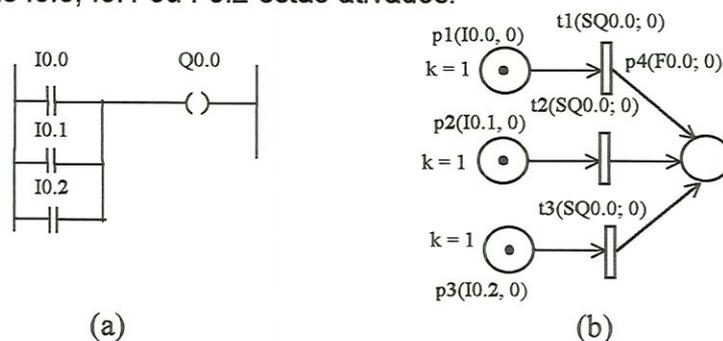


Figura 4.2 – Representação da Lógica OU (OR).

A Figura 4.2 (a) mostra o intertravamento na linguagem de CLP (“Ladder”) enquanto que a Figura 4.2 (b) mostra a sua correspondência utilizando a modelagem em redes de Petri.

Da mesma forma que a lógica E, a lógica OU utiliza os *lugares* de entrada externo e/ou “flags”, além dos sinais de seqüência, sendo que nas *transições* são realizadas também as ativações das saídas especificadas.

Como a análise é realizada sobre a *transição*, a figura 4.2 (b) pode ser traduzida em três lógicas E, onde qualquer uma das três entradas ativaria a saída Q0.0, caracterizando assim a ativação da saída como um OU. O disparo da *transição* está condicionado as entradas, de forma que na lógica OU, tantas ações quanto forem necessárias poderão ser repetidas em várias *transições*.

4.1.3 - Comparador

Outro elemento do controle programável é o comparador, o qual compara dois registros. As operações de comparação em controladores lógicos programáveis podem ser descritas na Tabela 4.2.

Tabela 4.2 - Operações de comparação

Operação	Função
! =	comparação de “igual a”
> <	comparação de “diferente”
>	comparação de “maior”
> =	comparação de “maior ou igual”
<	comparação de “menor”
< =	comparação de “menor ou igual”

A Figura 4.3 descreve as operações de “igualdade”, “maior ou igual” e “menor ou igual” no diagrama *Ladder*, sendo que para as outras operações só se altera o operador.

Para o comparador da Figura 4.3 tem-se que CMP é um atributo que identifica o comparador, V1 e V2 são as variáveis a serem comparadas (“byte” de entrada, palavra de “flags”, etc.) e Q é a saída do comparador que é acionada quando a condição de comparação é verdadeira, fazendo com que o “flag” seja ativado.

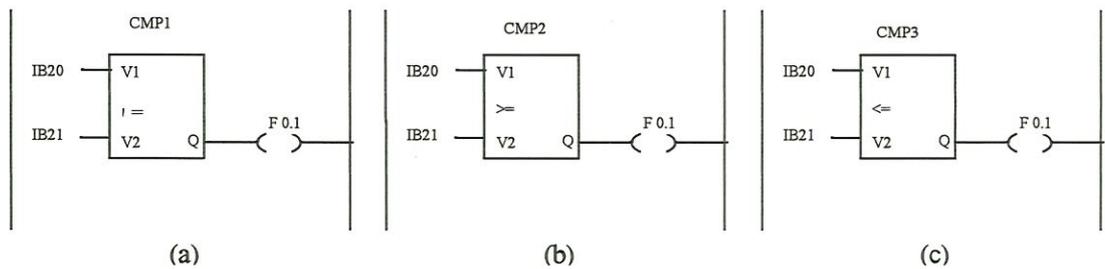


Figura 4.3 - Diagramas de contatos de um comparador (DIN 19 239).

Os CLPs possuem dois acumuladores para a comparação. As operações de carga e transferência são incondicionais e não dependem do resultado da operação lógica de nenhuma operação.

Para a condição de igualdade a Figura 4.3 (a) mostra uma comparação entre dois registros, isto é, se IB20 é “igual a” IB21, o “flag” F0.1 é ativado. O “flag” F0.1 é ativado quando IB20 for “maior ou igual a” IB21 para a Figura 4.3 (b) e para a Figura 4.3 (c) o “flag” F0.1 será ativado se o resultado da comparação for IB20 “menor ou igual a” IB21.

A Figura 4.4 mostra um comparador em redes de Petri de igualdade entre dois registros (IB20 e IB21). Onde os lugares e fluxos em vermelho representam as conexões externas.

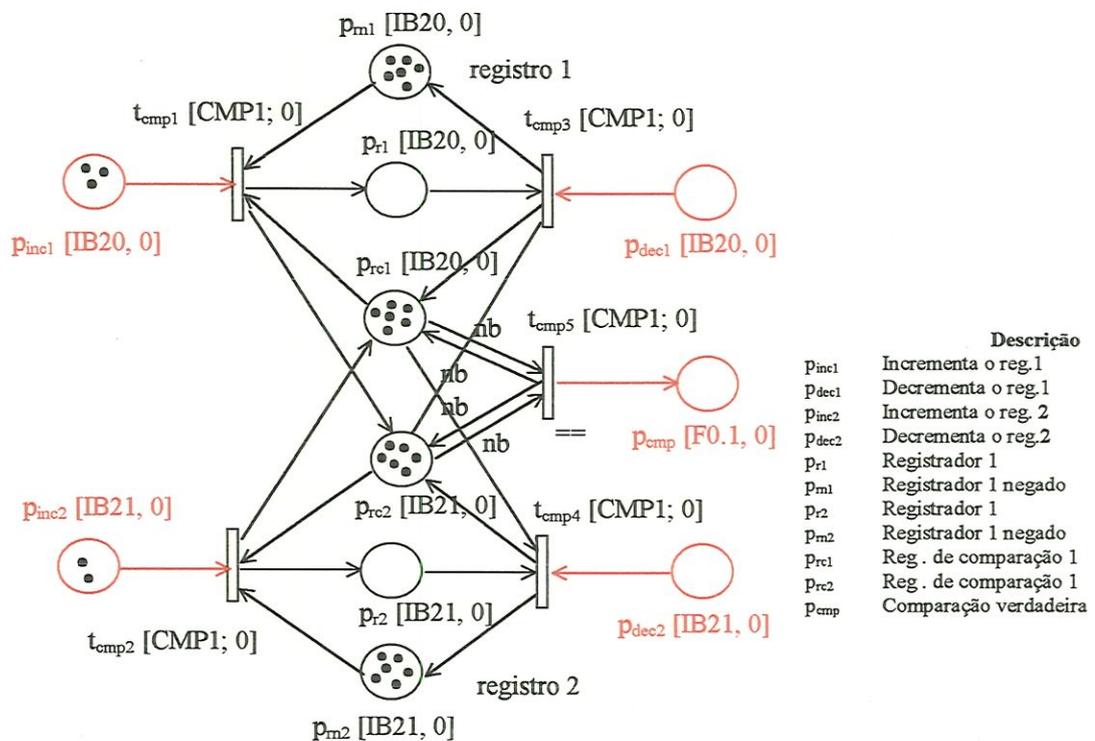


Figura 4.4 – Comparador de igualdade em redes de Petri.

Inicialmente o contador espera pela entrada dos valores dos registros, *lugares* p_{inc1} e p_{inc2} , através das *transições* t_{cmp1} e t_{cmp2} . O disparo consecutivo de cada uma destas *transições* irá “desbalancear” a quantidade de marcas dos *lugares* de comparação p_{rc1} e p_{rc2} , que estarão novamente em condição de igualdade, e conseqüentemente disparo da *transição* t_{cmp5} , quando os *lugares* p_{r1} e p_{r2} tiverem o mesmo número de marcas, configurando a igualdade e ativando o “flag” F0.1.

Na marcação inicial utilizada para exemplificar o funcionamento do comparador, tem-se que o valor máximo de comparação é “0”, no caso prático vai depender do sistema adotado (geralmente FFFFh).

Desta forma o comparador modelado em redes de Petri pode funcionar da mesma maneira que o comparador descrito anteriormente na lógica de contatos da Figura 4.3 (a).

As *transições* que compõe o comparador são descritas abaixo:

- t_{cmp1} - *transição* de carga do primeiro registrador para a comparação;
- t_{cmp2} - *transição* de carga do segundo registrador para a comparação;
- t_{cmp3} - *transição* de descarga do primeiro registrador para a comparação;
- t_{cmp4} - *transição* de descarga do primeiro registrador para a comparação;
- t_{cmp5} - *transição* de comparação dos registradores auxiliares;

Em termos de modelagem em redes de Petri pode-se então definir um comparador da seguinte maneira:

Pr = conjunto de registradores a serem comparados, onde $Pr = \{p_{r1}, p_{r2}\}$;

p_{r1}, p_{r2} são limitados a nb (n -bounded) onde $M(p_{r1}) + M(p_{m1}) = nb$ e $M(p_{r2}) + M(p_{m2}) = nb$;

ou seja: p_{m1} é o lugar complementar de p_{r1} e p_{m2} é o lugar complementar de p_{r2} ;

$$\bullet p_{r1} = p_{m1} \bullet = t_{cmp1}$$

$$p_{r1} \bullet = \bullet p_{m1} = t_{cmp3}$$

$$\bullet p_{r2} = p_{m2} \bullet = t_{cmp2}$$

$$p_{r2} \bullet = \bullet p_{m2} = t_{cmp4}$$

p_{rc1} e p_{rc2} são o par comparador de p_{r1} e p_{r2} respectivamente onde:

$$\bullet p_{rc1} = p_{r1} \bullet \cup \bullet p_{r2} + t_{cmp5}$$

$$p_{rc1} \bullet = \bullet p_{r1} \cup p_{r2} \bullet + t_{cmp5}$$

$$\bullet p_{rc2} = p_{r2} \bullet \cup \bullet p_{r1} + t_{cmp5}$$

$$p_{rc2} \bullet = \bullet p_{r2} \cup p_{r1} \bullet + t_{cmp5};$$

a sua marcação é definida como:

$$M(p_{rc1}) = M(p_{rn1}) + M(p_{r2}) \text{ e}$$

$$M(p_{rc2}) = M(p_{rn2}) + M(p_{r1});$$

t_{cmp5} é a *transição* de comparação de igualdade onde:

$$t_{cmp5} : \bullet t_{cmp5} = p_{rc1}, p_{rc2}; t_{cmp5} \bullet = p_{rc1}, p_{rc2}, p_{cmp} \text{ (externo)};$$

o peso dos fluxos dos arcos do comparador são unitários exceto:

$$w(F(p_{rc1} \times t_{cmp5})) = w(F(p_{rc2} \times t_{cmp5})) = w(F(t_{cmp5} \times p_{rc1})) = w(F(t_{cmp5} \times p_{rc2})) = nb;$$

na comparação de igualdade de dois registros:

$$M(p_{rc1}) = M(p_{rc2}) = w(F(p_{rc1} \times t_{cmp5})) = w(F(p_{rc2} \times t_{cmp5})) = \text{disparo de } t_{cmp5};$$

Os outros tipos de comparação ($>$, $<$, \geq, \dots) utilizam a mesma estrutura de comparação em redes de Petri, no entanto os elementos do par comparador são analisados individualmente como mostrado na Figura 4.5 para a comparação de “maior que” e na figura 4.6 para a comparação de “menor que”. Devido a essa estrutura semelhante e a marcação depender do sistema considerado (máximo valor de comparação) esses comparadores serão apresentados na forma de blocos, sem a representação da marcação inicial, podendo caso necessário ser utilizada a marcação da Figura 4.4.

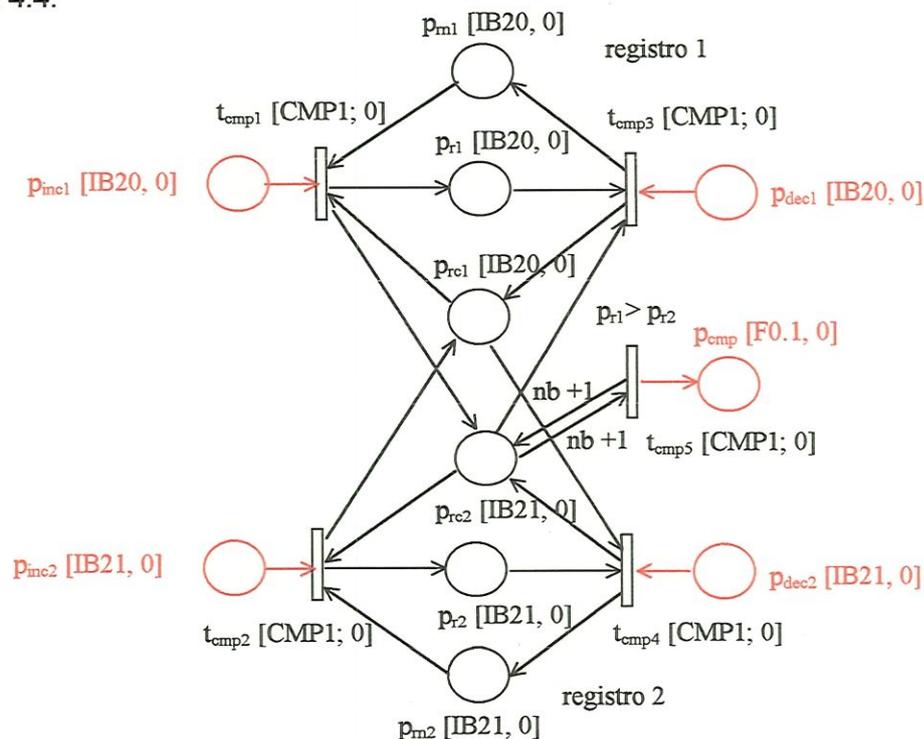


Figura 4.5 – Comparador de “maior que” em redes de Petri.

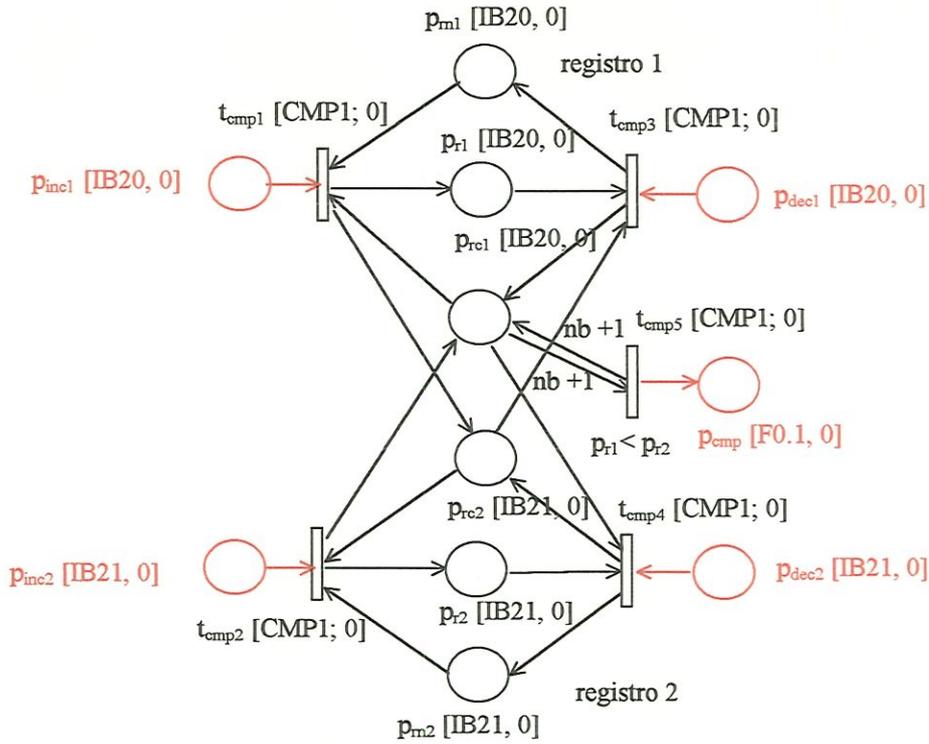


Figura 4.6 – Comparador “menor que” em redes de Petri.

E analogamente para a comparação de “menor que”, tem-se t_{cmp5} como:

$$t_{cmp5}: t_{cmp5} \bullet = p_{rc1}, p_{cmp}; \bullet t_{cmp5} = p_{rc1};$$

o peso dos fluxos dos arcos do comparador “maior que” são unitários exceto:

$$w(F(p_{rc1} \times t_{cmp5})) = w(F(t_{cmp5} \times p_{rc2})) = nb + 1; \quad \text{assim:}$$

$$M(p_{rc1}) > w(F(p_{rc1} \times t_{cmp5})) = nb + 1 = \text{disparo de } t_{cmp5} \text{ portanto } M(p_{r1}) < M(p_{r2})$$

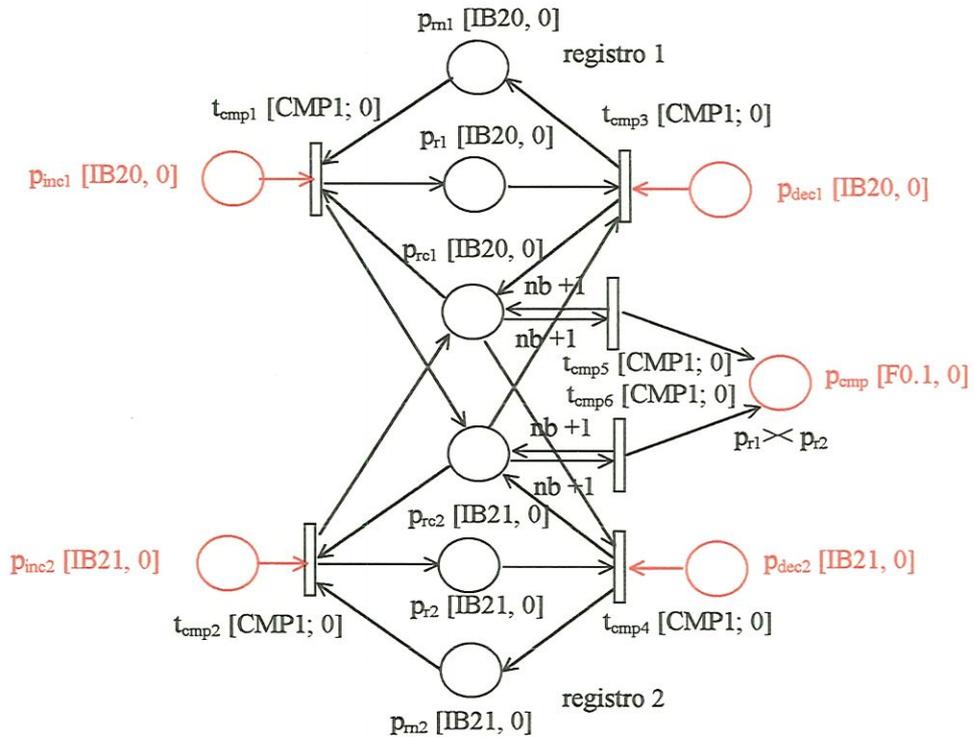


Figura 4.7 – Comparador de “diferente de” em redes de Petri

Para a comparação de “maior ou igual a” e “menor ou igual a”, substitui-se na estrutura dos comparadores os pesos $nb+1$ por nb , resultando nos disparos de t_{cmp5} para estas comparações.

O comparador de “diferente de” pode ser projetado como a união dos comparadores “maior que” e “menor que”, como mostrado na figura 4.7, ou utilizando-se o lugar complementar (“flag”) do comparador de igualdade, da mesma forma que as comparações de “maior ou igual a” e “menor ou igual a” são os *lugares* complementares de “menor que” e “maior que”, respectivamente.

Para a comparação de “diferente de”, tem-se t_{cmp5} e t_{cmp6} como:

$$\bullet t_{cmp5} = p_{rc1}; t_{cmp5}^{\bullet} = p_{cmp}; e$$

$$\bullet t_{cmp6} = p_{rc2}; t_{cmp6}^{\bullet} = p_{cmp};$$

o peso dos fluxos dos arcos do comparador de diferenças são:

$$w(F(p_{rc1} \times t_{cmp5})) = w(F(t_{cmp5} \times p_{rc1})) = nb + 1;$$

$$w(F(p_{rc2} \times t_{cmp6})) = w(F(t_{cmp6} \times p_{rc2})) = nb + 1;$$

assim:

$$M(p_{rc1}) > w(F(p_{rc1} \times t_{cmp5})) = nb + 1 = \text{disparo de } t_{cmp5} \text{ ou}$$

$$M(p_{rc2}) > w(F(p_{rc2} \times t_{cmp6})) = nb + 1 = \text{disparo de } t_{cmp6}$$

portanto $M(p_{r1}) \gg M(p_{r2})$

4.1.4 - Contador

Os contadores são dispositivos bastante usados para a contagem de pulsos ou eventos externos. Podem ser colocados em cascata para produzirem repetidamente formas de ondas necessárias para a sincronização do processo sobre controle (DITCH, 1995).

O contador realiza tanto contagem crescente como decrescente. Desta forma pode ser usado com duas entradas diferentes, por meio de dois pulsos distintos (NATALE, 1995). Para um contador decrescente de 16 bits o primeiro valor seria FFFFh, decrescendo até atingir o valor de contagem “CV”. Para a contagem crescente o primeiro valor contado seria 0001 crescendo até atingir o valor “CV”.

O valor de contagem atual é disponibilizado para comparações na forma de um registrador de contagem.

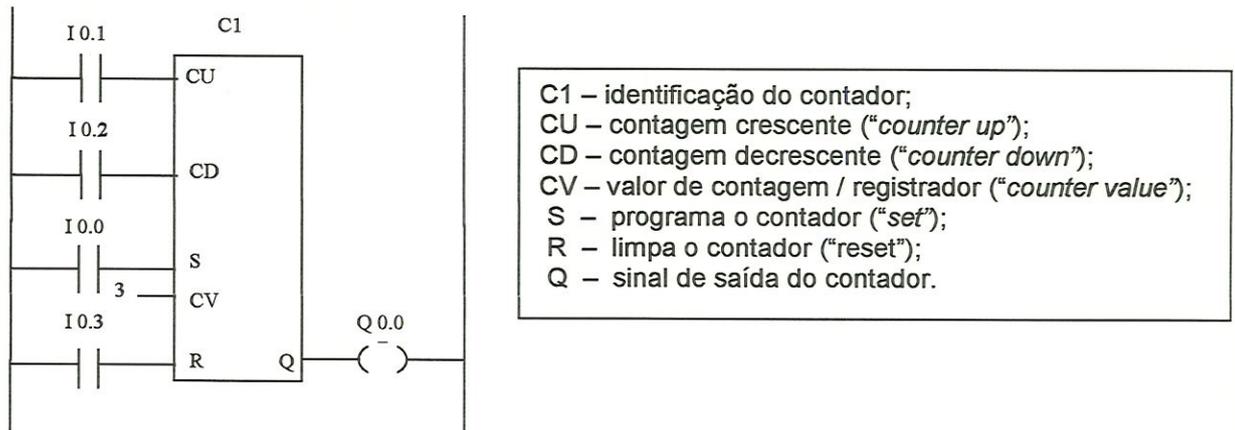


Figura 4.8 - Diagrama de contatos de um contador

O registrador de valor de contagem “CV” compara se a contagem atingiu o valor programado no caso de contagem crescente ou se o valor programado foi decrescido até atingir este valor.

A Figura 4.8 descreve um contador crescente na forma de diagrama *Ladder* (DIN 19 239) e a Figura 4.9 a sua representação em redes de Petri. Da mesma forma que anteriormente, a marcação inicial utilizada para exemplificar o funcionamento do contador tem o valor máximo de contagem de “6”, e no caso prático, o valor também depende do sistema adotado.

Na Figura 4.8 para se incrementar um contador (“CU”) é utilizada a entrada I0.1, de modo que se a contagem atingir o valor programado em “CV”, determinado pelo usuário, a saída Q0.0 será atuada. O valor 3 é programado quando I0.0 é ativado na entrada “S” do contador, sendo este armazenado no registrador. Para a contagem decrescente o valor do registrador é decrescido e se atingir o valor determinado em “CV” a saída Q0.0 sinalizaria com valor 1.

A Figura 4.9 representa um contador em rede de Petri, o qual possui um *lugar* de entrada p_s para determinar o valor de contagem “CV”, o *lugar* p_{cu} para o incremento da contagem, o *lugar* p_{cd} para o decremento da contagem, o *lugar* de reset p_r e um *lugar* de estado da contagem atingida p_a , o qual irá habilitar a *transição* imediata t , e atuar na saída Q0.0.

A estrutura do contador é muito parecida com a do comparador de igualdade de dois registros, além de se acrescentar um bloco de carga do valor de comparação do contador (“set”) e um bloco de limpeza do contador (“reset”).

A representação em redes de Petri do contador utilizado no CLP, utiliza um grupo de *transições* as quais possuem funções determinadas e são utilizadas em conjunto.

Estas *transições* são identificadas pelo número do contador. Elas irão disparar em condições especiais descritas abaixo:

- t_s : set – é disparada na rotina inicial para se colocar o valor de contagem “CV” que será comparado com o valor do registrador de contagem;
- t_{rs} : reset – pode ser disparada em qualquer contagem para reiniciar o contador dentro do programa, desta forma qualquer que seja o número de marcas no registrador, elas serão retiradas, sem alterar no entanto o valor de CV, colocando o contador na condição de reinício de contagem;
- t_{cu}/t_{cd} : up/down – realiza o incremento/decremento do valor de contagem armazenado no registrador.
- t_Q : fim – indica que o valor do registrador do contador é igual ao valor programado em “CV” ativando a saída do contador.
- t_{cmp1} - *transição* de carga do primeiro registrador para a comparação;
- t_{cmp2} - *transição* de descarga (zeramento) do primeiro registrador para a comparação;
- t_z - *transição* de zeramento do valor “CV” anterior;
- t_r - *transição* de reinício do valor de contagem acumulado;
- t_{reh} - *transição* de reabilitação do contador;
- t_i - *transição* de carga do valor do registrador a ser comparado com a contagem;
- t_{icv} - *transição* de carga do valor “CV” a ser comparado com a contagem;
- t_{hs} - *transição* de habilitação do set do contador.

Desta forma, um contador pode ser definido por três blocos diferentes, bloco de comparação, bloco de “set” e bloco de “reset”, representados na Figura 4.9 pelas cores preta, azul e verde, respectivamente. Os *lugares* e fluxos em vermelho representam as conexões externas, como já mencionado.

O bloco de comparação segue as mesmas diretrizes definidas no comparador, com o acréscimo de uma *transição* de reinício. Onde:

$$p_{r1}, p_{r2} \text{ são limitados a } nb \text{ (} n\text{-bounded) e } M(p_r) + M(p_m) = nb;$$

Da mesma forma que anteriormente a sua marcação é definida como:

$$M(p_{rc1}) = M(p_{m1}) + M(p_{r2}) \text{ e}$$

$$M(p_{rc2}) = M(p_{m2}) + M(p_{r1});$$

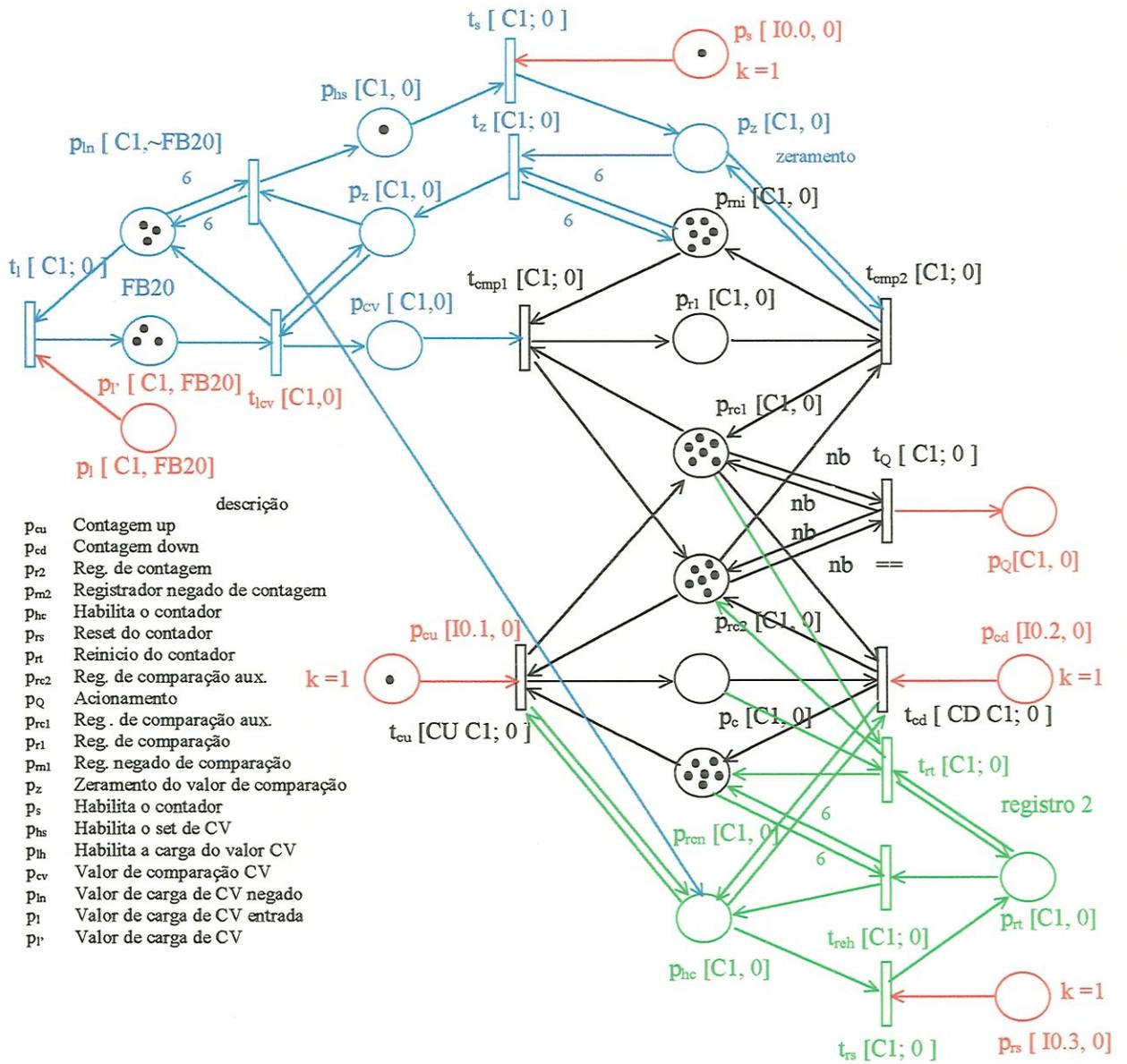


Figura 4.9 – A representação de um contador em redes de Petri.

Os lugares p_{m1} , p_{m2} e p_{ln} são os lugares complementares de p_{rc1} , p_{rc2} e p_l , respectivamente e são utilizados para o controle da carga dos registradores.

t_Q é a transição de comparação de valor atingido onde:

$$t_Q = p_{rc1}, p_{rc2}; t_Q^e = p_{rc1}, p_{rc2}, p_Q \text{ (externo);}$$

o peso dos fluxos dos arcos do comparador são unitários exceto:

$$w(F(p_{rc1} \times t_Q)) = w(F(p_{rc2} \times t_Q)) = w(F(t_Q \times p_{rc1})) = w(F(t_Q \times p_{rc2})) = nb; \text{ e}$$

os pesos dos arcos de verificação $w(F(p_{1n} \times t_{hs})) = w(F(t_{hs} \times p_{1n})) = w(F(p_{m1} \times t_z)) = w(F(t_z \times p_{m1})) = w(F(p_{m2} \times t_{reh})) = w(F(t_{reh} \times p_{m2})) = FFFFh$ (por exemplo)

O bloco de “set” executa o zeramento do conteúdo de p_{r1} , verificado através de p_{m1} permitindo assim que o valor atual do registrador programado seja carregado, utilizando as *transições* de carga t_i e t_{cv} . Após o término da carga, verificada pela *transição* t_{hs} , uma marca é colocada no lugar p_{hc} de habilitação do contador, permitindo que sejam realizadas as contagens tanto crescente como decrescente.

O bloco de “reset” permite que a qualquer momento o contador seja reiniciado, sem a necessidade de se programar novamente o valor de CV. Este bloco desabilita o contador através da *transição* t_{rs} e é o responsável em descarregar as marcas de contagem, utilizando a *transição* t_{rt} , sem interferir no “balanço” de comparador. A verificação e habilitação novamente do contador é realizada por t_{reh} , a qual reabilita o contador a partir de seu zeramento.

Os *lugares* p_s , p_{cu}/p_{cd} , p_{rs} , p_Q e p_i podem ser de vários tipos, de acordo com o modelo onde se deseja introduzir o contador. Podem ser de entrada “Ix.x”, com capacidade $k = 1$, de sinais internos “Fx.x”, com capacidade $k = 1$ ou de seqüência “-“, com $k = \infty$.

4.1.5 - Temporizador

O temporizador é o elemento do sistema de intertravamento responsável pela geração de atrasos (*delay*) nos sinais. Este elemento sincroniza a lógica de controle com os sistemas reais (acionamento de motores, visualização de displays, etc.).

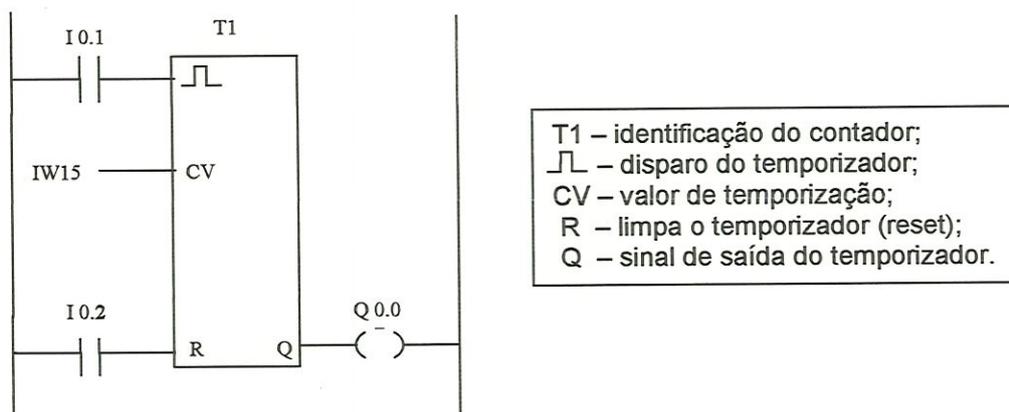


Figura 4.10 - Um tipo de representação gráfica de um temporizador

Nos circuitos com relés eletromagnéticos os temporizadores são baseados em circuitos de resistores com capacitores, circuitos pneumáticos ou motorizados. Já nos CLPs estes circuitos são substituídos por uma lógica digital, implementado através do

relógio do sistema (clock), possibilitando que diversas temporizações possam ser implementadas no sistema por software (MIYAGI, 1996).

De modo geral, o temporizador pode ser classificado baseado na relação entre o sinal de entrada e o de saída. A Figura 4.10 descreve um temporizador na forma de diagrama *Ladder* (DIN 19 239) onde:

Existem basicamente dois tipos de temporizadores (NATALE, 1995): os de pulso e os de retardo. Os temporizadores de pulso são aqueles que geram um pulso na saída ao mesmo tempo que habilitados, por um tempo limitado ou definido. Os temporizadores de retardo, são aqueles que ao serem habilitados só acionam a saída após transcorrido o tempo determinado. A figura 4.11 ilustra as formas de onda para as respostas dos temporizadores de pulso mais comuns e as figuras 4.12, 4.13 e 4.14 mostram as representações gráficas destes temporizadores em redes de Petri.

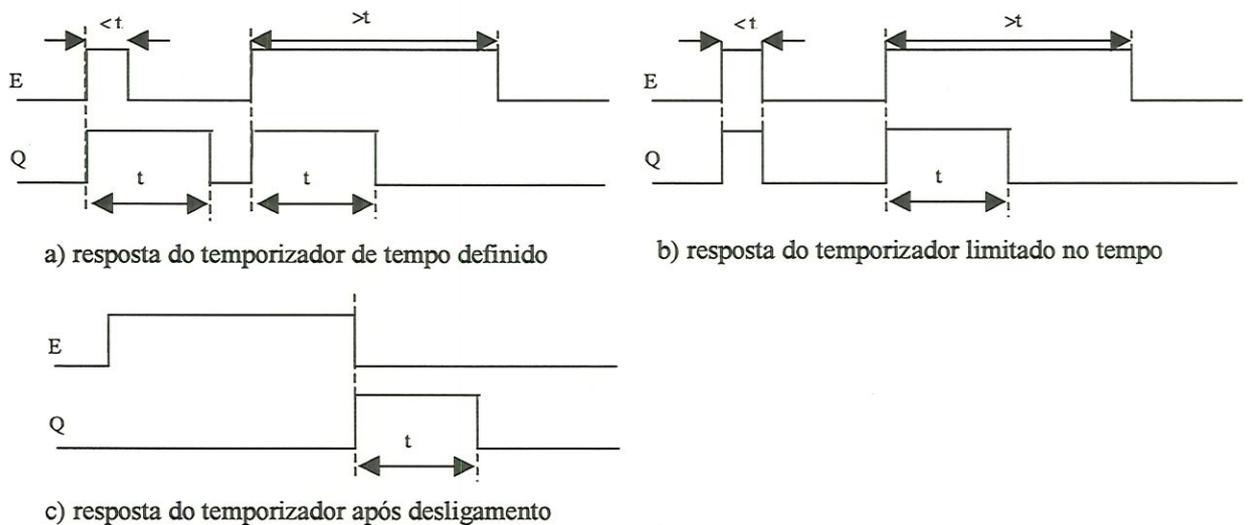


Figura 4.11 – diagrama de tempo de um temporizador de pulso

Os temporizadores de retardo são caracterizados pelas formas de onda da figura 4.15 e suas representações em redes de Petri são mostradas nas figuras 4.16 e 4.17.

A estrutura em redes de Petri do temporizador é parecida com a do contador, onde tem-se o bloco de *set* e o de *reset*, no entanto agora o acionamento da saída é ditado pelo tipo de temporizador escolhido.

Basicamente o temporizador funciona com a habilitação do lugar p_{en} , o temporizador da Figura 4.12 é de tempo definido, não havendo a necessidade de se preocupar com a condição de entrada, isto é, o tempo do acionamento sempre será executado. Novamente, na marcação inicial utilizada para exemplificar o funcionamento do modelo, tem-se que o valor máximo de temporização é “6”, no caso prático vai depender do sistema adotado.

Na representação gráfica do temporizador limitado no tempo, Figura 4.13, e nas demais representação dos vários tipos de temporizador, devido a estrutura semelhante do temporizador da Figura 4.12, serão apresentadas na forma de blocos, sem a representação da marcação inicial, podendo caso necessário ser utilizada a marcação da Figura 4.12, com exceção do limite FFFFh utilizado no lugar do limite "6".

Para o temporizador da figura 4.13 a ausência da marca do lugar p_{en} irá desabilitar o temporizador, que deve desligar a saída. Isto é, o acionamento da saída esta condicionado a condição de habilitação.

O temporizador de pulso de acionamento após o desligamento da habilitação pode ser visualizado na figura 4.14 onde a sua estrutura é alterada para que o acionamento de Q0.0 seja executado por um tempo determinado após o desligamento da habilitação do sistema (p_{en}).

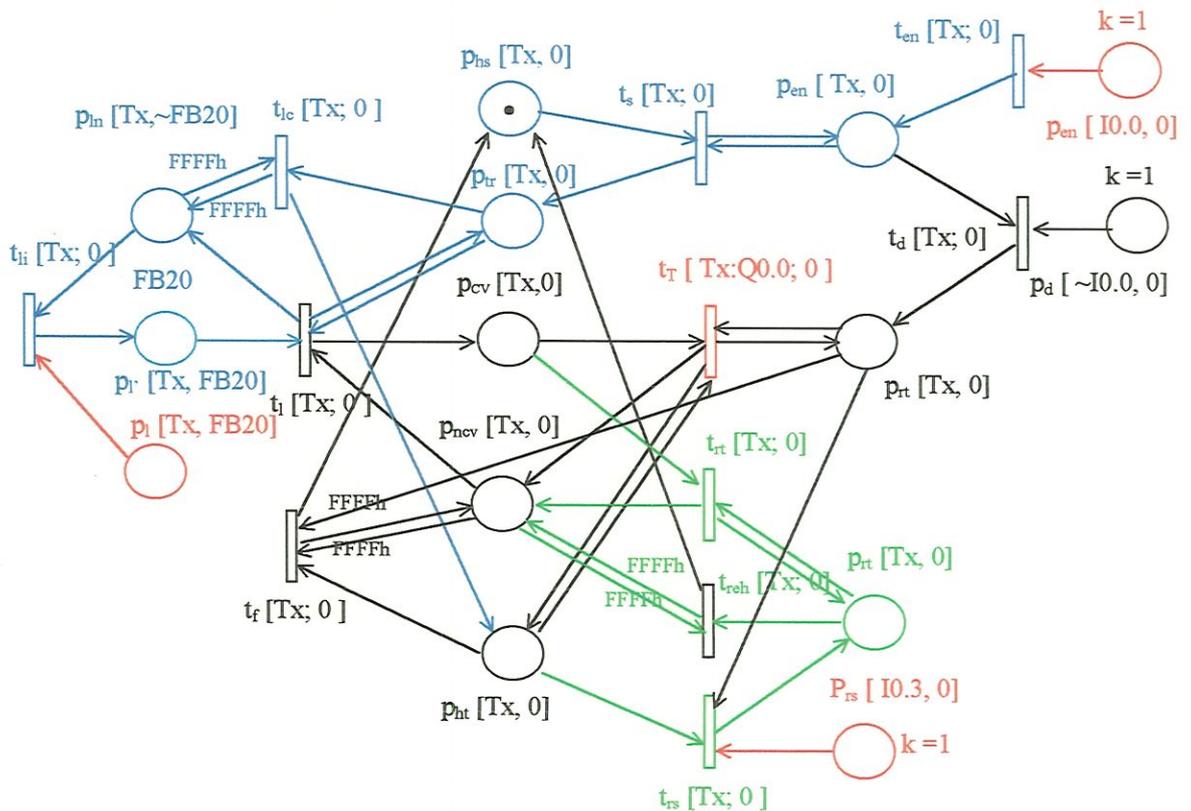


Figura 4.14 - Representação gráfica de um temporizador em redes de Petri após desligamento.

Em termos de modelagem em redes de Petri pode-se então definir um comparador da seguinte maneira:

$$M(p_{cv}) + M(p_{cvn}) = \text{tempo adimensional};$$

t_T é a *transição* de acionamento, onde a saída especificada estará atuando enquanto estiver disparando esta *transição*. Quando cessar os disparos a saída é desligada.

Os blocos de set e reset se comportam de maneira semelhante aos do contador.

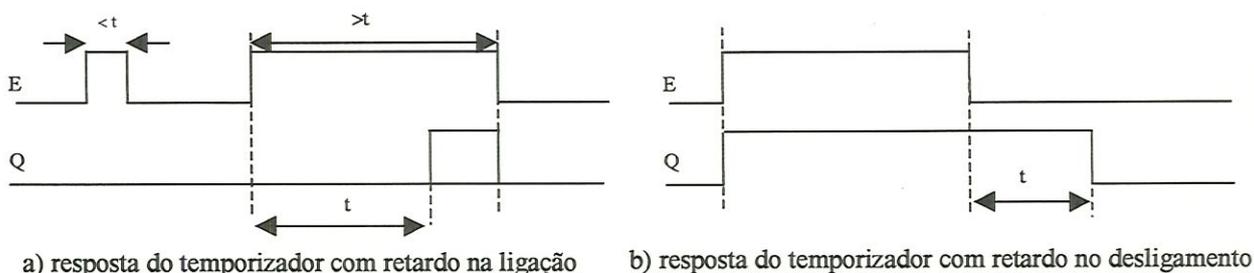


Figura 4.15 – diagramas de tempo de um temporizador de retardo

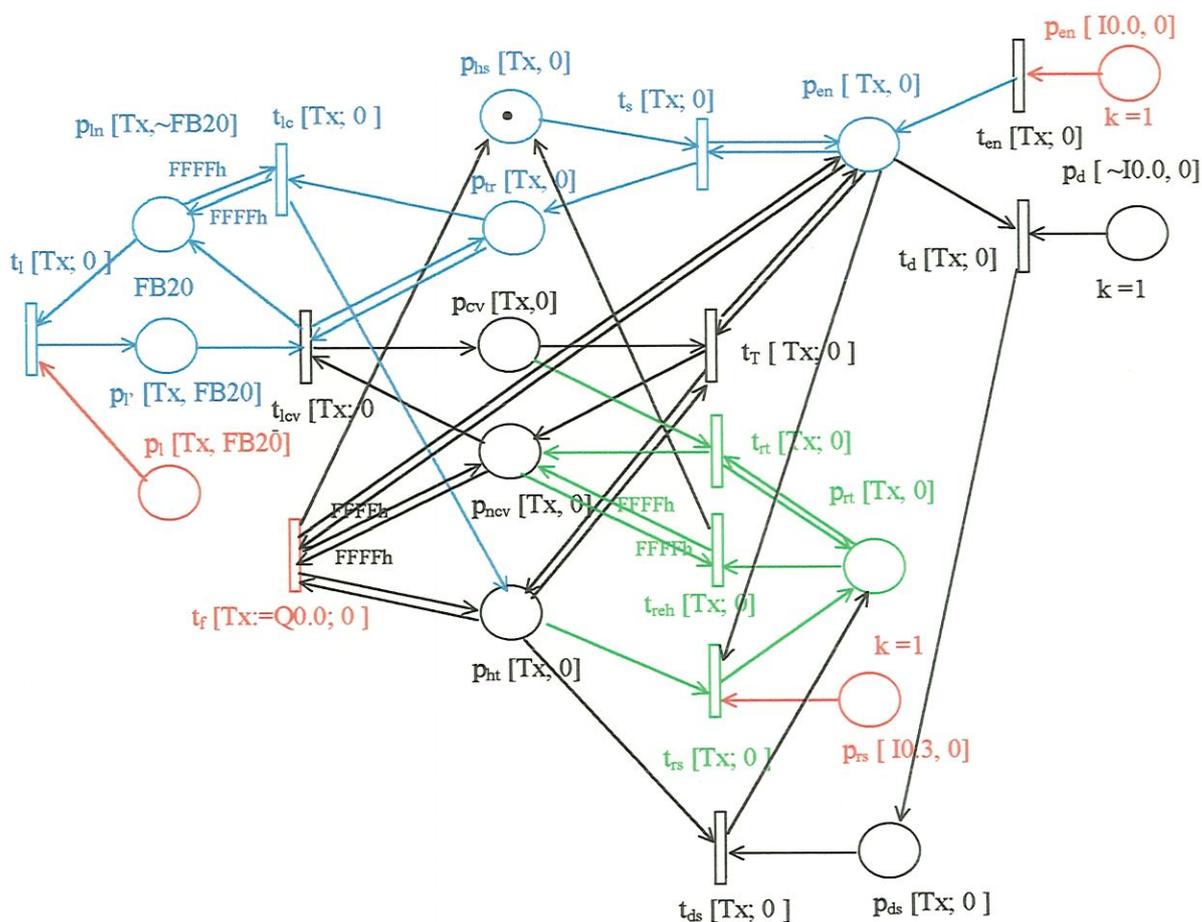


Figura 4.16 - Representação gráfica de um temporizador em redes de Petri com retardo na ligação.

A representação em redes de Petri do temporizador, assim como o contador, utiliza um conjunto de *transições* as quais possuem funções determinadas e são utilizadas em conjunto. Estas *transições* fazem a identificação do temporizador, fornecem

sua base de tempo e após transcorrido o tempo, ativam as saídas programadas. Elas disparam nas condições especiais descritas abaixo:

- t_{en} - *enable* - é disparada toda vez que se desejar realizar a contagem de um tempo;
- t_d - *disable* - é disparada toda vez que se desejar retirar a habilitação do temporizador;
- t_s - *set* - é responsável pela iniciação do temporizador com a carga do valor de contagem;
- t_{rs} - *reset* - pode ser disparada em qualquer tempo para desabilitar a contagem de tempo,
- t_T - *base de tempo* - indica a base de tempo a ser utilizada pelo temporizador e aciona a saída determinada, de acordo com a seguinte tabela 4.3.

Tabela 4.3 – base de tempo

Base	tempo (s)
0	0,01 seg.
1	0,1 seg.
2	1 seg.
3	10 seg.

- t_f - *fim* - indica que transcorreu o tempo determinado, disparando uma determinada atividade e habilitando o reinício do temporizador.

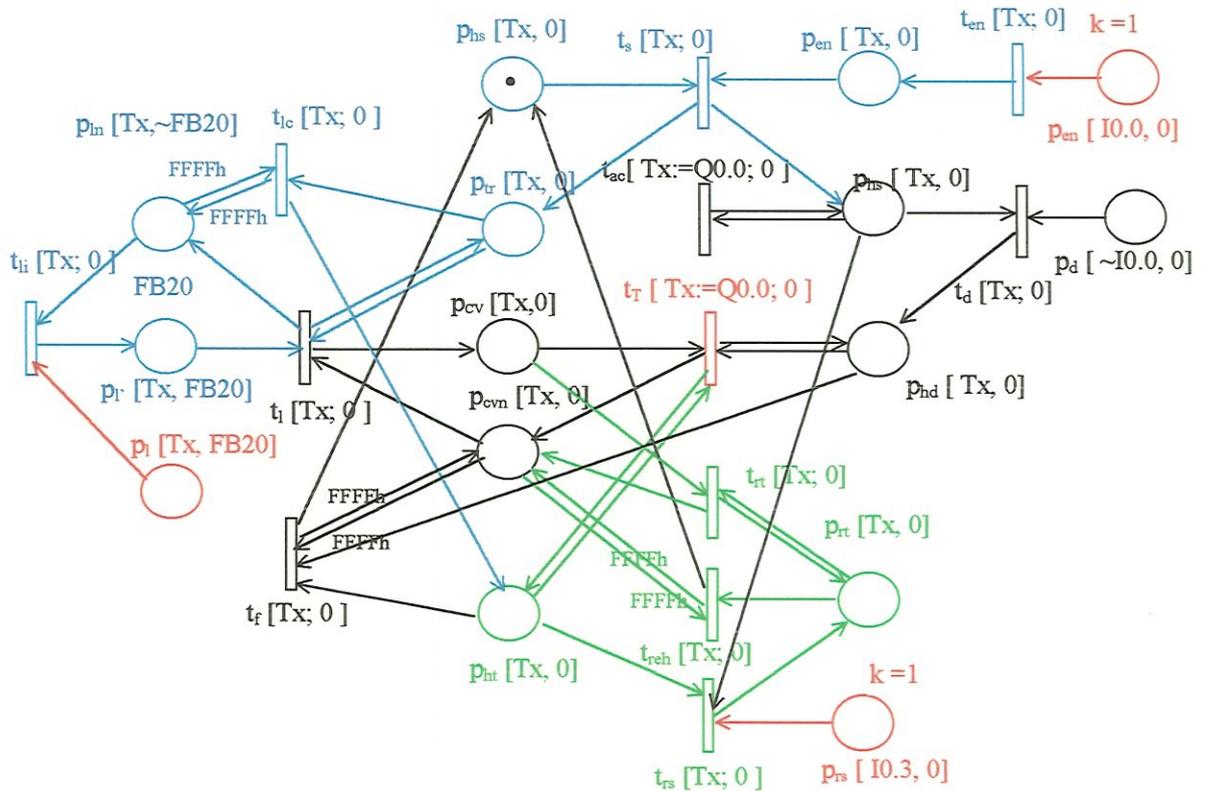


Figura 4.17 - Representação gráfica de um temporizador em redes de Petri com retardo no desligamento.

- t_{rt} - *transição* de reinício do valor de temporização;
- t_{reh} - *transição* de reabilitação do temporizador;
- t_i - *transição* de carga do valor do registrador a ser temporizado;
- t_{icv} - *transição* de carga do valor CV a ser temporizado na estrutura do temporizador;
- t_{ht} - *transição* de habilitação do temporizador.
- t_{dt} - *transição* de desabilitação do temporizador;
- t_{ac} - *transição* de acionamento inicial da saída do temporizador;

Tem-se que p_{cv} e p_{ncv} são pares complementares, assim como p_l e p_{ln} , os quais controlam o final da temporização e da carga do valor de temporização respectivamente.

t_f é a *transição* de comparação de valor de tempo atingido, que aciona um determinado procedimento na seqüência do programa.

O peso dos fluxos dos arcos do temporizador são unitários exceto os pesos dos arcos de verificação:

$$w(F(p_{cvn} \times t_f)) = w(F(t_f \times p_{cvn})) = w(F(t_{ic} \times p_{ln})) = w(F(p_{ln} \times t_{ic})) = FFFFh \text{ (por exemplo)}$$

O bloco de “set” executa o zeramento do conteúdo de p_{cv} , verificado através de p_{cvn} permitindo assim que o valor atual do tempo programado seja carregado e processado.

O bloco de “reset” permite que a qualquer momento o temporizador seja reiniciado. Este bloco desabilita o contador através da *transição* t_s e é o responsável em descarregar as marcas de contagem, utilizando a *transição* t_{rt} . A verificação e habilitação novamente do contador é realizada por t_{reh} , a qual reabilita o temporizador após o seu tempo atingir o valor zero.

4.2 – Modelagem em redes de Petri de sistemas de manufatura utilizando os elementos da linguagem CLP

A modelagem de sistemas automáticos baseados em CLPs objetiva a representação de um sistema na forma gráfica, um fácil entendimento entre as várias partes envolvidas no projeto e uma facilidade na sua especificação e alteração.

O projetista passa a modelar o sistema de forma que a abstração das redes de Petri fiquem restritas aos elementos de um CLP, necessários para a execução de um automatismo.

Desta forma as redes de Petri são identificadas de acordo com os elementos de um CLP, isto é, são utilizados conjuntos de *lugares* e *transições* para representarem não só as lógicas booleanas básicas como E ou OU, mas também elementos como temporizadores, contadores e comparadores. Estes elementos são determinados utilizando os *lugares* e as *transições* individuais ou através de agrupamentos, como descrito nas seções anteriores.

Estes elementos do CLP são introduzidos sem a descaracterização da definição das redes de Petri. As características como temporização, sensores de entrada e atuadores de saída deverão ser integrados dentro do modelo na forma de atributos. Tais atributos serão reconhecidos na sistemática de transformação para a linguagem de programação CLP e para a simulação dos modelos.

Para que o modelo do sistema seja adequado para a sua conversão em uma linguagem de programação CLP e as análises de características comportamentais sejam utilizadas para a análise destes modelos, a modelagem de sistemas automáticos envolve a elaboração deste modelo em dois passos principais: o passo de modelagem onde são utilizados os elementos de controle de um CLP e o passo onde a seqüência de funcionamento do sistema é detalhada para que sejam procedidas as análises das propriedades comportamentais das redes de Petri e as medidas de desempenho do sistema, chamado de passo de funcionamento.

4.2.1 - Passo de Controle

No passo de controle as redes de Petri são utilizadas na representação dos elementos que irão compor os intertravamentos do sistema, que são identificados através de atributos específicos para um programa de conversão adequado.

Este modelo será o utilizado para a conversão do modelo de controle para uma linguagem de programação de intertravamentos CLP. Os *lugares* representados pelos sensores, chaves fim de curso, etc., atuam diretamente nas *transições* para realizar uma ação, sem se preocupar com os *deadlocks* que poderão acontecer na fase de funcionamento do sistema.

Os estados encontrados no modelo serão associados a *flags* internos, os quais poderão ser ou não utilizados para compor outros intertravamentos.

Uma saída do CLP não pode ser utilizada como um lugar de entrada de uma *transição* e um lugar de entrada não pode ser utilizado como uma saída de uma *transição*, isto é, o subconjunto de *lugares* de entrada de uma *transição* não podem

conter *lugares* que representem uma saída externa e o subconjunto de *lugares* de saída de uma *transição* não pode conter *lugares* que representem entradas externas.

Na associações das redes de Petri com os componentes da linguagem CLP temos os *lugares* representando um estado do sistema que pode corresponder, em termos de controle, a um sinal de *entrada externo* (característico de entradas externas de CLP) ou *flags* de sinalização interna.

A capacidade de cada um destes *lugares* é especificada de acordo com a sua função na representação do elementos do CLP.

A partir da caracterização destes elementos do funcionamento do CLP e da definição das redes de Petri, pode-se realizar algumas associações da representação de redes de Petri com linguagens de programação CLP. A definição destes elementos foram descritos na seção anterior.

4.2.2 - A elaboração do passo de controle

A partir dos elementos básicos de modelagem apresentados anteriormente, o usuário poderá modelar o intertravamento de máquinas e sistemas de manufatura utilizando redes de Petri. Esta modelagem contará com atributos que particularizarão o modelo visando sua conversão em termos de intertravamentos para uma linguagem CLP.

Na modelagem de um sistema automático, o primeiro passo é a verificação das entradas e saídas necessárias, além da enumeração dos eventos discretos a serem executados pelo sistema.

Posteriormente, elabora-se através do uso das redes de Petri o modelo de controle, onde as entradas, os *flags* e as *transições* descrevem cada intertravamento individualmente (consideremos um intertravamento um bloco do sistema que possui uma lógica Booleana a qual permite o acionamento de uma saída ou *flag* do sistema).

A Figura 4.18 ilustra o modelo de controle de um sistema de abrir e fechar de uma porta, onde são utilizados dois sensores e dois atuadores. Os sensores A e B são os responsáveis por identificar o estado em que a porta se encontra, se aberta ou fechada, respectivamente. Os atuadores A e B são os responsáveis em fechar ou abrir a porta, respectivamente.

O programa CLP em linguagem Ladder correspondente ao modelo da figura 4.18 pode ser descrito na Figura 4.19.

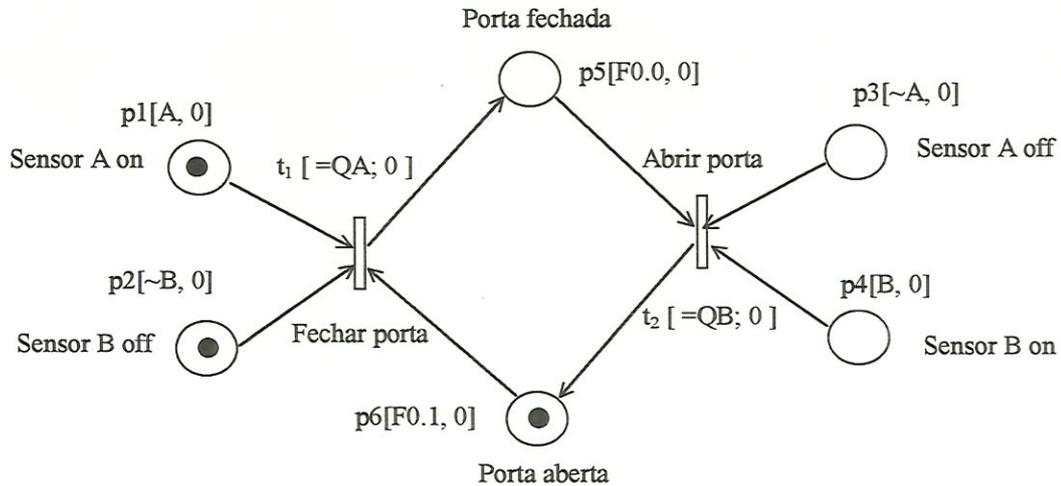


Figura 4.18 – Modelo de controle de uma porta automática.

Do modo como foi especificada a representação dos elementos em redes de Petri, a conversão do modelo passa a ser automática para uma linguagem CLP padrão. Na conversão todas as *transições* são analisadas dentro de um ciclo do CLP, de forma que as condições consideradas na marcação inicial em cada ciclo para o disparo das *transições* serão analisadas. Isto é, não há alterações das condições externas durante o processamento daquele ciclo do modelo.

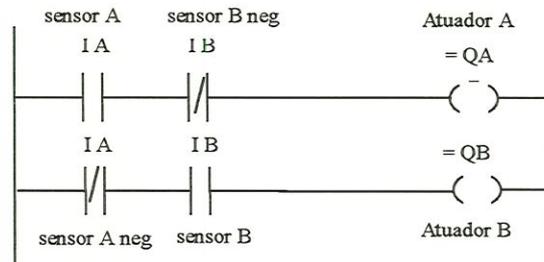


Figura 4.19 – Programa de controle de uma porta automática em linguagem Ladder.

A modelagem dos intertravamentos na forma gráfica permite além de uma facilidade visual do sistema, a não repetição de *lugares*, evitando a redundância (repetição) na programação dos blocos e as interferências de sinais nos intertravamentos as quais resultam em funcionamento falho e até a parada do sistema.

4.2.3 - Passo de funcionamento

No *passo de funcionamento* é simulada a seqüência de funcionamento da máquina ou sistema, onde os *lugares* de entrada poderão receber marcas, isto é, estes *lugares* compõe o subconjunto de *lugares* de saída de uma *transição*. Fazendo com que o sistema possa dar prosseguimento a sua seqüência de funcionamento.

Neste modelo, arcos, *lugares* e *transições* são incorporados no modelo de controle para que o sistema possa funcionar de acordo com suas necessidades. Desta

forma utilizando as características comportamentais das redes de Petri poderão ser realizadas as análises das seqüências de funcionamento do sistema.

Das características comportamentais das redes de Petri, tem-se que o modelo construído para o sistema deve ser livre de deadlocks (live), não há conflitos no sistema e sim alternativas que de acordo com a marcação podem ser dirigidas pelo projetista, e se o sistema for composto por apenas lógicas E e OU (sem os elementos como o contador, comparador e temporizador) é limitado a 1 (uma) marca para os *lugares* ($k=1$) (safe).

Para a análise de desempenho do sistema, o componente temporal é inserido nas *transições* necessárias, na forma de um atributo, para que o sistema funcione de forma sincronizada em tempo real.

4.2.4 - A elaboração do modelo no passo de funcionamento

No passo de funcionamento serão analisadas as seqüências de funcionamento do modelo, de modo que deverão ser elaboradas seqüências de funcionamento tanto correta como adversas para que se possa ser analisado os intertravamentos a serem implementados.

Estas seqüências podem ser geradas automaticamente através de um software de análise da árvore de alcançabilidade de um modelo editado em redes de Petri ou por inspeção do funcionamento e algumas situações críticas desejáveis para serem verificadas.

A Figura 4.20 ilustra o modelo de funcionamento de uma porta automática, que incorpora o modelo de controle ilustrado na Figura 4.18.

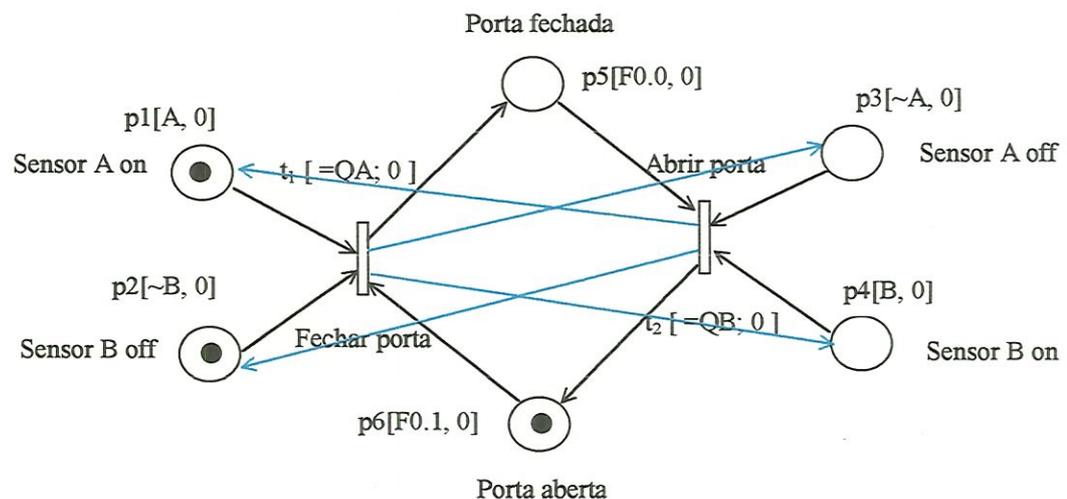


Figura 4.20 – Modelo de funcionamento de uma porta automática.

No modelo de funcionamento da Figura 4.20 são incorporados mais quatro arcos, os quais irão retornar as marcas nos sensores indicando seu próximo estado, desta forma a porta poderá abrir e fechar como projetado. A ausência de um destes fluxos levaria o sistema a uma condição de *deadlock*, mostrando que há uma falha no sistema.

A marcação inicial $M_0 = (1,1,0,0,0,1)$ indica as pré - condições de fechamento da porta, enquanto que a $M_1 = (0,0,1,1,1,0)$ indica as pré - condições para a abertura da porta.

4.3 – A conversão do modelo para a linguagem CLP

Na conversão das redes de Petri para a linguagem CLP, utiliza-se as *transições* e os *lugares* como base de conversão, estes são verificados se fazem parte do modelo de controle, através dos seus atributos de identificação. Desta forma temos que a conversão pode ser realizada do seguinte modo:

- A conversão é baseada nas *transições*. Todas as *transições* são analisadas isoladamente e são identificadas as *transições* que compõe a lógica de controle do sistema.
- Uma *transição* de controle pode ser analisada como uma lógica booleana simples, caracterizada por fluxos de entradas e de saída unitários ou integrante de outros elementos de controle.
- O elemento “comparador” pode ser caracterizado testes de registradores, onde a seção 4.1.3 descreve o seu funcionamento. O conjunto de *transições* que determina o comparador formam um grupo especificado por CMP_x , onde x é o número do comparador utilizado.
- Um “contador” pode ser caracterizado por um conjunto de *transições* e *lugares* que formam um grupo especificado por C_x , onde x é o número do contador utilizado. Estas *transições* podem exercer funções de preparação do contador, zeramento do contador, saída do contador e contagem (decrecente ou crescente). A seção 4.1.4 descreve o funcionamento do contador.
- O temporizador é caracterizado por um conjunto de *transições* e *lugares* identificados por T_x , onde x é o número do contador utilizado, e possui elementos de carga do valor de temporização e de zeramento, semelhantes

aos do contador. A seção 4.1.5 descreve os vários tipos de temporizador utilizados em um automatismo e sua representação em redes de Petri.

A Figura 4.21 mostra uma conversão para a lógica booleana simples onde os fluxos de entrada na *transição* descrevem a operação booleana AND e a *transição* irá acionar as respectivas saídas e acionar o flag F3.0.

Na modelagem, os elementos descritos podem ser utilizados como blocos que podem até serem alocados em outro arquivo diferente daquele que descreve a seqüência de controle principal do automatismo.

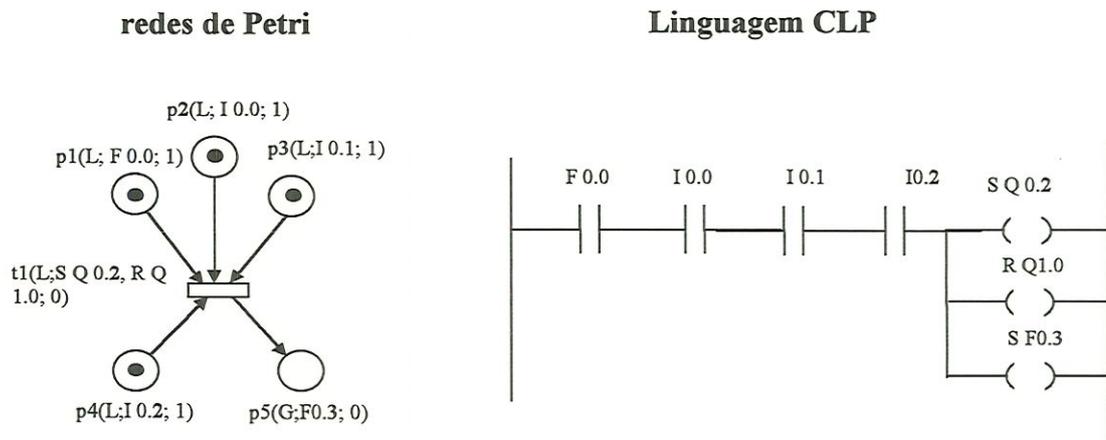


Figura 4.21 - Conversão de uma *transição* para linguagem CLP.

Para isso, na implementação da conversão de uma rede de Petri para a linguagem CLP, é necessário a utilização de um software que permita a modularização gráfica dos elementos de uma rede de Petri, sem que seja perdida as características ordinárias das redes de Petri *lugar – transição*.

4.4 – Exemplo de modelagem dos passos de controle e de funcionamento

Será utilizado um exemplo para ilustrar as funções de intertravamento usadas e o seu processo de conversão para linguagem CLP.

Considere uma punçoneira semi-automática, que se compõe de uma mesa fixa para receber a peça colocada pelo operador e um punção em posição de repouso alto. Quando o operador liga, o punção desce, permanece na posição baixa durante 5 segundos e depois sobe, voltando ao início. Esta operação se repete por 10 vezes. O

motor deve ser acionado para baixo e depois revertido para cima, existe ainda um botão de desliga (emergência) que pode interromper o processo em qualquer situação.

Deve-se determinar o que muda os comportamentos, isto é, as condições lógicas que determinam a passagem de um comportamento para o outro.

A tabela 4.4 descreve as ações e estados do sistema e os sinais necessários para esta simples automação são listados abaixo, juntamente com o relacionamento com as redes de Petri, para a descrição do intertravamento.

Tabela 4.4 - Ações e estados do sistema.

AÇÕES	ESTADOS de Sequência
Acionar para cima	Motor ligado para cima
Acionar para baixo	Motor ligado para baixo
Desligar	Motor desligado
Ligar	
Esperar 5 Seg	
Contar 10 vezes	

Entradas	I0.0	Botão de liga
	I0.1	Botão de desliga
	I0.2	Sensor baixo
	I0.3	Sensor alto
Saídas	Q0.0	Contator do motor baixo
	Q0.1	Contator do motor cima
Flags:	F0.0	Sistema desligado
	F0.1	Sistema ligado
	F0.2	Início de ciclo
	F0.3	Motor ligado para baixo
	F0.4	Motor ligado para cima
	F0.5	Reset do temporizador T1
	F0.6	Tempo atingido
	F1.0	Contagem com o contador C1
	F1.1	Set do contador C1
	F1.2	Reset do contador C1
	F1.3	Contagem atingida em C1
Temporizadores:	T1	Espera por 5 segundos
Contadores:	C1	Contagem de 10 vezes

Deste modo pode-se representar o sistema pelo uso de redes de Petri através dos estados e ações.

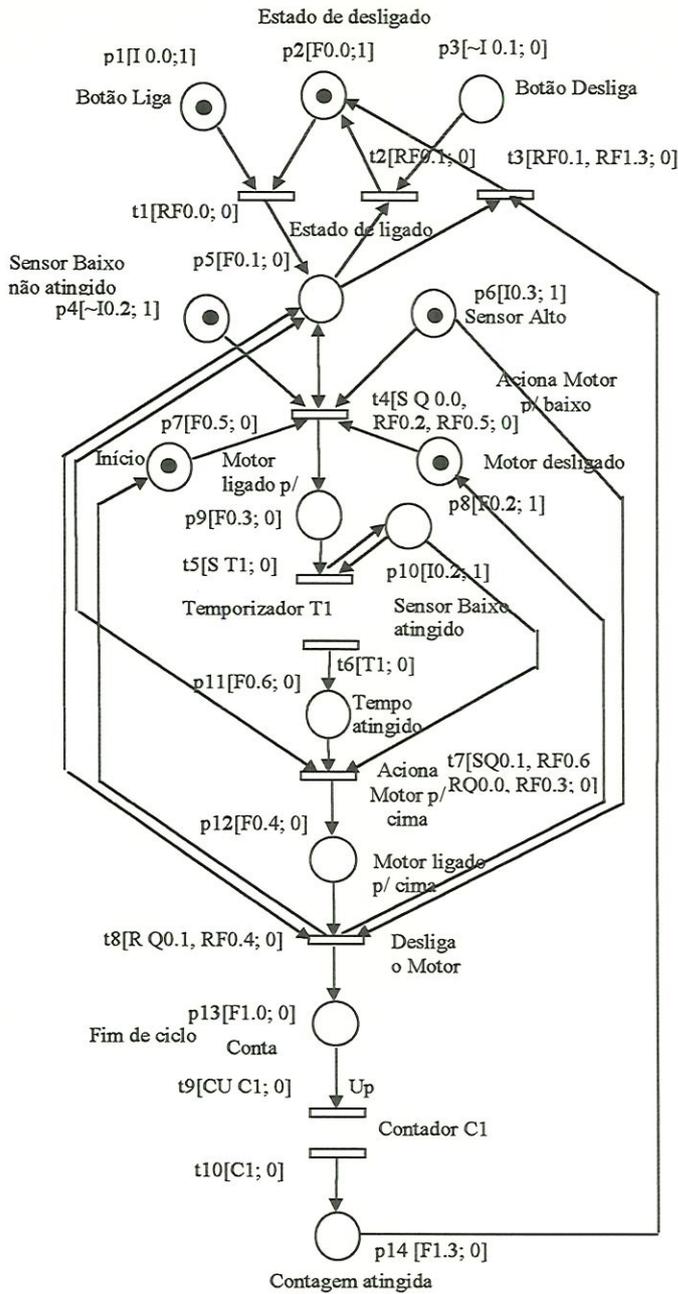
Estados: Sinais externos (botões e sensores)
 Estados de seqüência (internos)

Estados de construção do contador

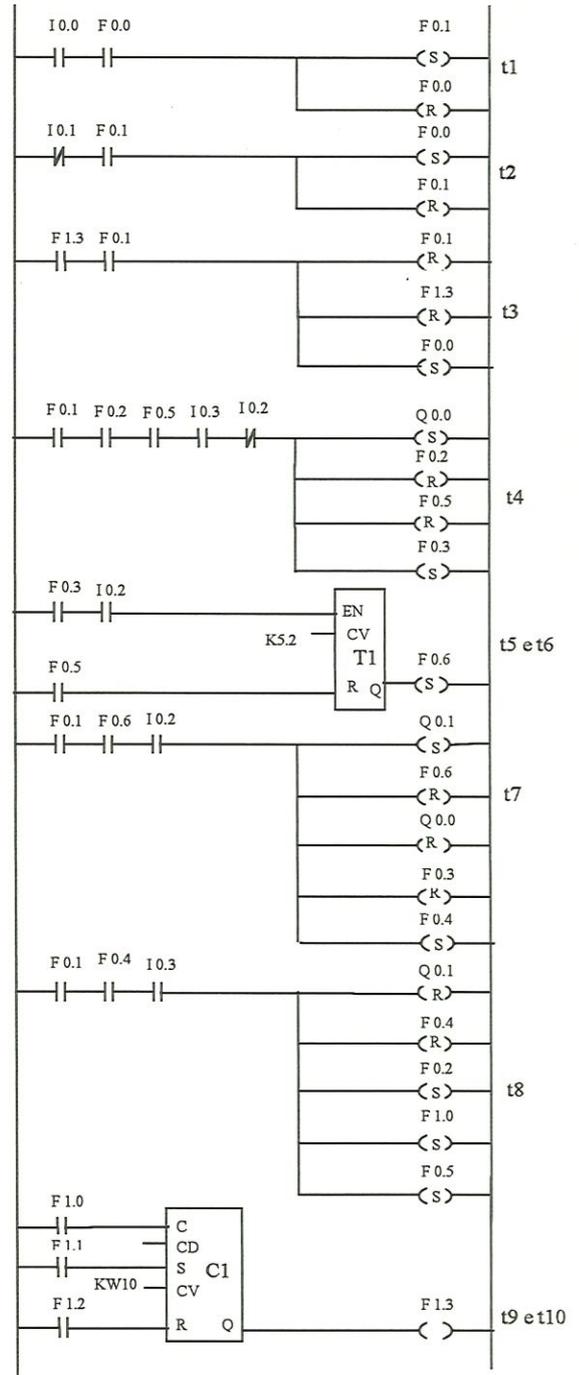
 Estado de construção do temporizador
 Ações: processo
 contagem
 temporização

Pode-se notar na Figura 4.22 que um loop é determinado no sistema para designar o ciclo da máquina. As *transições* t_4 , t_9 e t_{10} , representam a seqüência principal do programa, onde estão alocados os estados (*lugares*) e as ações (*transições*) da Tabela 4.6. Alguns estados representam as entradas do sistema (botões e sensores), e nas ações, estão localizadas as saídas do sistema (acionamento do motor).

Na Figura 4.22 as *transições* t_5 e t_6 são modulares e fazem parte do bloco do temporizador T1, descrito no item 4.1.5. As *transições* t_9 e t_{10} também são modulares e fazem parte do contador C1, descrito no item 4.1.4. devido a característica modular destes elementos, o modelo da punctionadeira em redes de Petri contempla principalmente a seqüência principal do automatismo, facilitando assim a sua visualização.



(a)



(b)

Figura 4.22 - Modelo de uma punctionadeira em rede de Petri (a) e linguagem Ladder (b).

5 - AMBIENTE DO PROJETO DE INTERTRAVAMENTOS PARA MÁQUINAS CNC

Neste capítulo, a sistemática proposta para o projeto dos intertravamentos de sistemas de manufatura, é implementada na forma de um ambiente de projeto para máquinas ferramenta CNC.

As características de funcionamento dessas máquinas serão abordadas, sendo que elas irão direcionar a criação dos módulos que compõe o ambiente, por exemplo, o sistema especialista irá tratar de questões relacionadas as necessidades de uma máquina, tanto como para a sua manutenção, o editor de redes de Petri deve permitir a inclusão de algumas funções de intercâmbio e troca de parâmetros entre o CLP e o CNC da máquina (funções M, H, parâmetros R, etc.), além de outras.

Assim, será abordada a criação do ambiente de projeto de intertravamento para máquinas CNC, e utilizando esse ambiente, é realizada a sua validação com a implementação de um torno CNC.

5.1 – Modelagem de sistemas de intertravamento para máquinas CNC

Na modelagem de redes de Petri dos intertravamentos para máquinas CNC ou da sua seqüência de funcionamento, o equipamento de controle da máquina pode ser dividido em dois, (MOREAUX & DECOTIGNIE, 1991) (MORNINGSTAR, 1993): o Comando Numérico Computadorizado (CNC) e o Controlador Lógico Programável (CLP).

O CNC trata diretamente do posicionamento e velocidade do eixos controlados, enquanto que o CLP trata do ambiente de acionamentos, sinalizações da máquina e sua interação com o meio externo.

A configuração de uma máquina CNC genérica é utilizada como base para a modelagem da máquina desejada e pode ser dividida em quatro componentes:

- 1- Estrutura mecânica;

- 2 - painel de comando (Interface Homem Máquina - IHM);
- 3 - Comando Numérico Computadorizado (CNC);
- 4 - Controlador Lógico Programável (CLP);

Nesta análise será acrescentado um quinto componente que é o sistema gerenciador da célula para que a máquina possa trabalhar em sincronismo com outros elementos do sistema de manufatura.

A Figura 5.1 ilustra estes elementos, fornecendo o modelo para o desenvolvimento do intertravamento de máquinas CNC integradas a um sistema de manufatura.

O CNC é o equipamento que possui um microprocessador responsável pelo posicionamento e velocidade dos eixos. Além da função de controle dos eixos, ele disponibiliza uma interface com o usuário, a Interface Homem Máquina (IHM).

A IHM consiste de uma apresentação em um monitor de vídeo de várias telas, selecionáveis de acordo com as necessidades do operador. Estes modos de operação podem ser apresentadas como (MAXITEC, 1989-b):

- modo de edição;
- modo de entrada de dados;
- modo de saída de dados;
- modo de trabalhos manuais (JOG);
- modo de trabalhos em automático;
- modo de referenciamento;

Além de outros, os quais dependem do fabricante, do usuário e do grau de sofisticação da máquina. Estes modos são normalmente dispostos na forma de menus e submenus e são acessados por botões e teclas.

O CNC possui grande versatilidade na alteração de dados e trocas de programas, possibilitando a preparação rápida da máquina (diminuição do setup da máquina).

Possui dispositivos de integração com computadores gerenciadores, como por exemplo interfaces seriais e equipamentos DNC (Controle Numérico Direto), tomando viável sua integração em ambientes automatizados, auxiliando no planejamento e controle da produção.

O CNC realiza o processamento do posicionamento de eixos, sejam estes independentes ou dependentes um dos outros, interpolando entre dois ou mais eixos. Estas interpolações podem ser lineares, circulares, helicoidais, etc., dependendo do nível tecnológico do equipamento necessário ao processo de usinagem.

Geralmente o CNC, em uma máquina ferramenta, trabalha em conjunto com outro equipamento microprocessado, o CLP, cuja função é a de gerenciamento do ambiente externo da máquina, isto é, todos os dispositivos que não possuem ligação direta com os eixos. Possui para isso entradas e saídas, as quais podem ser digitais ou analógicas. Com o avanço tecnológico dos microprocessadores, o microprocessador do CLP passa a não existir em alguns fabricantes de equipamentos CNC, sendo as funções do CLP e do CNC integradas em um único microprocessador. Tal configuração agiliza ainda mais a troca de dados entre o CLP e o CNC.

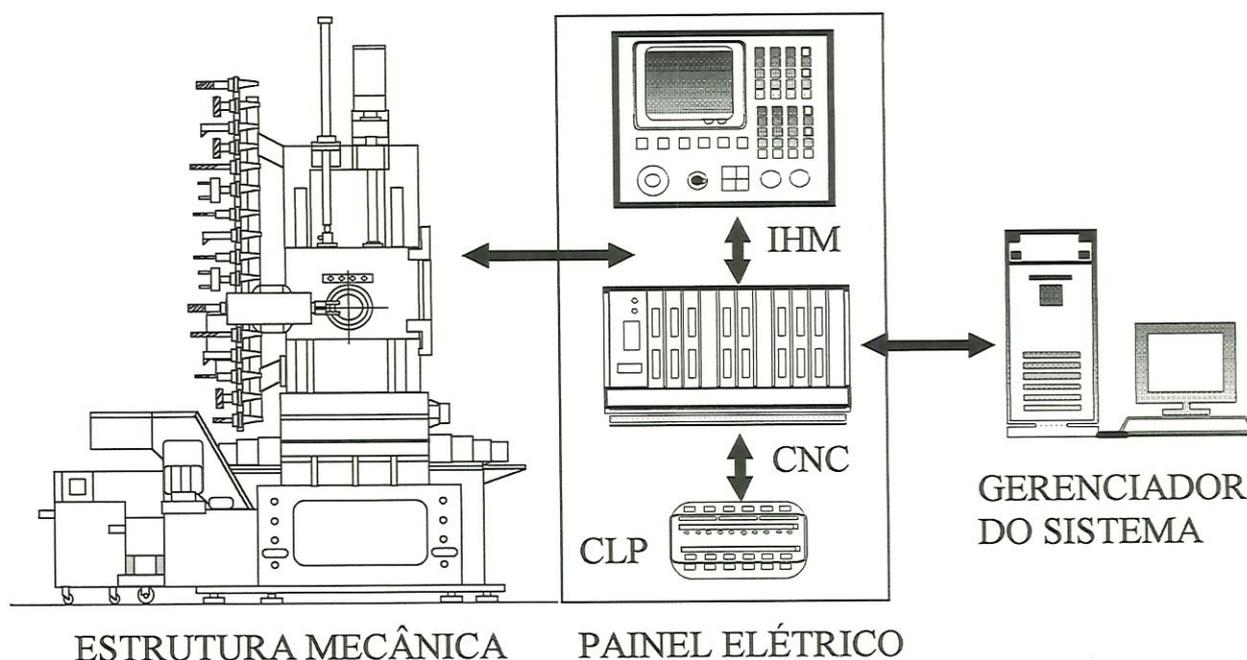


Figura 5.1 - Modelo adotado para estudo da estrutura de controle do sistema.

A estrutura mecânica possui basicamente vários elementos tais como sistemas de lubrificação de guias, refrigeração da peça, alimentação de peças, fixação para usinagem, descarga de peças, e muitos outros, dependendo da necessidade do processo.

Para configurações complexas, a composição da máquina poderá contar com um magazine para troca automática de ferramentas, uma unidade de alimentação hidráulica para os pistões acionadores do sistema de troca de ferramentas e um sistema de lubrificação centralizada automático. Além de uma unidade de extração de cavacos que pode atuar automaticamente na remoção do excesso de cavacos acumulado pelas operações de usinagem.

A integração da máquina CNC a um sistema de manufatura se realiza através da troca de sinais com o controlador da célula. O centro de usinagem envia ao controlador

seus sinais de "status" e recebe dele sinais de comando, responsáveis pela atuação na máquina.

Dos sinais de "status" enviados pelo controlador da célula, podemos relacionar aqueles referentes a informações das ações da máquina, e aqueles referentes aos alarmes ocorridos durante sua operação. Ambos os sinais são enviados ao controlador, o qual responderá ao sistema de acordo com o sincronismo com os outros elementos da célula. O controlador, então reage e comanda seqüências de ações as quais irão permitir a continuidade da execução automática da célula com o menor tempo de parada ("downtime") possível.

5.1.1 - Especificação dos elementos necessários para o projeto do intertravamento

O intertravamento localizado hierarquicamente no controle de sistemas de eventos discretos no nível "equipamento", é responsável pelo bom funcionamento eletrônico, de potência e dos fatores ligados à produtividade (avanços, velocidades, etc.) da máquina.

A máquina pode ser operada automaticamente pelo controlador da célula, que tem a função de enviar ordens de atuação, sincronizar os eventos com os outros componentes da célula e gerenciar a carga e descarga de programas CN.

O controlador da célula também deve ser responsável pelo início dos ciclos de usinagem e gerenciamento dos eventos que podem alterar os índices de desempenho das máquinas dentro da célula.

Abaixo estão listadas as comunicações necessárias entre máquina e controlador da célula:

- lista de alarmes;
- "status" da posição atual da usinagem;
- recuperação de falhas (paradas);
- acionamentos da máquina;
- preparação (*setup*) da máquina para atuação em automático.

O programa CLP é o responsável pelo gerenciamento dos componentes da máquina. Para isso, este programa torna-se o elo de ligação do CNC com o sistema de manufatura. A intensidade de troca de informações entre CLP e computador gerenciador da célula aumenta quanto mais flexível for o sistema.

A troca de dados, entre os vários elementos de uma máquina ferramenta CNC, é efetuada principalmente entre três dos componentes: o CLP, o CNC e a IHM.

Uma máquina ferramenta CNC permite o acesso às funções CNC, através do seu painel de controle (IHM), onde funções padronizadas programáveis (SIEMENS, GE FANUC) pelo usuário são implementadas através do CLP.

A troca de dados entre CLP e CNC é realizada através da área de memória (RAM) comum. Os dados são trocados na forma binária, podendo ser endereçados bit a bit, na forma de byte (8 bits) ou na forma de uma word (16 bits). Os sinais trocados entre CLP e CNC, durante a execução de um programa de usinagem são chamados de funções Miscelânea (funções M), funções de identificação de ferramentas (funções T), funções de acesso da manutenção (funções H) e as variáveis do CNC (parâmetros R) (MAXITEC, 1989-b).

As funções M enviam, do CNC ao CLP, informações dependentes da seqüência de execução do programa CN para sincronizar ações externas tais como, ligar e desligar a refrigeração da peça no momento correto da seqüência do programa.

Não há uma normalização de tais funções que seja seguida completamente em conjunto pelos fabricantes. Algumas funções M são alocadas de acordo com o tipo de máquina e de acordo com o programador. No entanto algumas funções mais comuns utilizam sempre os mesmos números, tais como mostrado na Tabela 5.1.

Tabela 5.1 - funções M

M00	Parada programada
M01	Parada condicional
M02	Fim de programa
M03	Liga eixo árvore no sentido horário
M04	Liga eixo árvore no sentido anti-horário
M05	Desliga eixo árvore
M30	Fim de programa sem retrocesso de fita

As funções H, sofrem da falta de padronização. O seu uso é facultativo por alguns fabricantes, pois pode-se agrupar todas as suas funções através das funções M, desde que o Comando Numérico em questão consiga acessar tais funções fora do modo automático. Algumas das funções H estão listadas na Tabela 5.2.

As funções T são responsáveis pela nomeação e seleção das ferramentas a serem usadas no processo de usinagem, sendo que seu acesso é feito diretamente pela linha do programa CN ou através da IHM no modo manual.

Tabela 5.2 - Funções H

H1	Desligamento do transportador de cavacos
H2	Religamento do transportador de cavacos
H3	Frenagem do eixo "Y"
H4	Desliga frenagem do eixo "Y"
H5	Liga lubrificação
H6	Desliga lubrificação
H7	Desliga bomba hidráulica
H8	Liga bomba hidráulica
H9	Trava porta
H10	Destrava porta

Os parâmetros R (MAXITEC, 1989-b) são os responsáveis pelas operações aritméticas realizadas dentro do programa CN. A comunicação CLP - CNC na forma de *byte* e *words* também é fornecida pelos parâmetros R, por isso é possível transmitir ao programa, dados de posições atuais de ferramentas, dispositivos, além da contagem de peças, em sincronismo com a seqüência do programa CN (tempo real).

O CLP e o CNC podem ser supervisionados através da IHM. Isto possibilita que o operador tenha total conhecimento do "status" do sistema. Através da IHM podem ser visualizadas as entradas e saídas do sistema, a área de memória do CLP na forma de *bit*, *byte* e *word*, a área de dados, os parâmetros de temporização e contagem e os "status" das funções M e H. Nesta interface são determinadas quais condições de intertravamento podem ser liberadas manualmente, para que a máquina volte a operar normalmente em qualquer posição de parada (travamento) ou simplesmente para teste de sensores e atuadores na sua manutenção.

As Tabelas 5.3 e 5.4 exemplificam alguns sinais trocados entre CLP e IHM na categoria de alarmes e de mensagens ao operador.

Tabela 5.3 - Alarmes.

F1	Fim de curso X + atingido
F2	Fim de curso X - atingido
F3	Fim de curso Y + atingido
F4	Fim de curso Y - atingido
F5	Sobrecarga no motor do eixo X
F6	Sobrecarga no motor do eixo Y
F7	Sobrecarga no ventilador do motor principal
F8	Sobrecarga no motor do hidráulico
F9	Falta pressão de ar
F10	Falha de lubrificação

Assim como as mensagens de falha, as mensagens de "status" são importantes e devem ser trocadas na comunicação entre a máquina CNC e o controlador da célula, além disso podem ser apresentadas no painel de comando, para o acompanhamento da operação da máquina.

Tabela 5.4 - Mensagens ao operador.

B1	Pedido de descarga do fuso
B2	Porta de alimentação aberta/destravada
B3	Porta de alimentação fechada/travada
B4	Garras recuadas
B5	Garras avançadas
B6	Fase I da usinagem terminada
B7	Início da Fase II de usinagem
B8	Trocador avançado
B9	Trocador recuado
B10	Lubrificação ligada

Através destas mensagens o controlador pode sincronizar as ações entre a máquina e os outros integrantes da célula de manufatura, e portanto, avaliar o andamento de sua produção.

5.2 – A implementação do software do ambiente para o projeto do intertravamento de máquinas ferramenta CNC.

A partir das considerações realizadas sobre as características de uma máquina ferramenta, será abordada a implementação do "software do ambiente de projeto do intertravamento" denominado ADIAC - Ambiente de Desenvolvimento de Intertravamento Assistido por Computador dedicado à máquinas ferramenta CNC.

O ambiente selecionado para o desenvolvimento do projeto foi o *Microsoft Windows* e a ferramenta de desenvolvimento do software *Borland Delphi 2.0*.

Na definição das plataformas de *hardware* e *software* adotadas, procurou-se escolher plataformas difundidas nos meios de projeto, pois praticamente todos os fabricantes de CNC e CLP possuem como alternativa as plataformas baseadas em IBM-PC e sistema operacional *Windows*. Além disso, há uma tendência destes equipamentos se tornarem de arquitetura aberta (PROCTOR & ALBUS, 1997) e utilizam estas mesmas plataformas.

A linguagem de programação foi o Pascal orientado a objeto e o projeto do “software” consistem na criação de formulários “forms” dentro do ambiente do *software Borland Delphi 2.0*. Um formulário “form” é composto por duas partes: uma interface de usuário, representando uma janela de sistema geral do Windows, e um bloco de códigos para manipular eventos e ações.

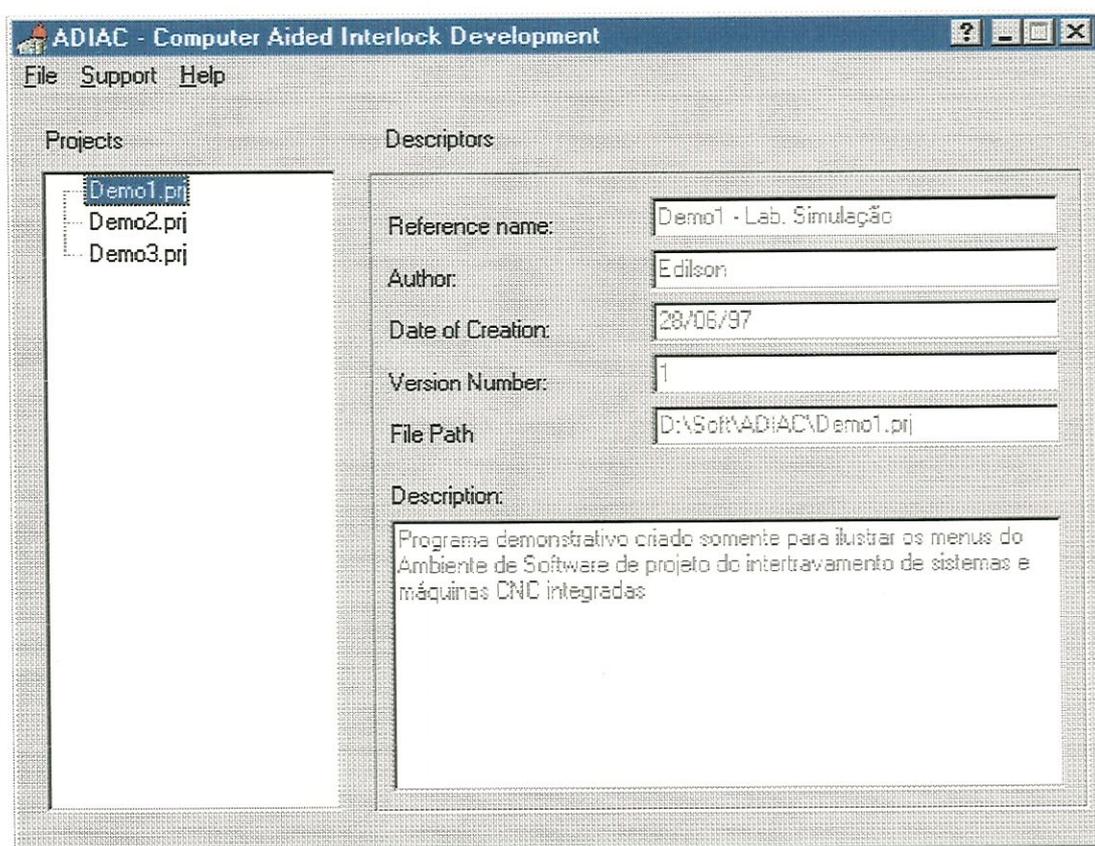


Figura 5.2 - Janela de abertura do Ambiente de *Software*.

A janela principal do sistema que é interface do usuário é composta por uma lista de descritores do projeto que contem informações gerais do sistema (possuem um nome de referência, um autor, uma data de criação, uma versão e uma descrição), esta lista de descritores compõe a base de dados das informações dos projetos. A Figura 5.2 mostra a janela implementada.

Nesta mesma janela a lista de *projects* é descrita a esquerda onde os projetos e seus componentes são localizados na forma de árvore com uma estrutura semelhante ao explorer do *Windows*. Além do arquivo da descrição do projeto, outros arquivos poderão ser visualizados e ativados, tais como os arquivos do editor de redes de Petri, da conversão CLP, de documentação, etc.

As opções *Edit* e *Create* no menu *File* abre uma outra janela que permite editar cada informação do projeto, esta janela é chamada de *Edit Project*. Os comandos

Rename, *Copy* e *Delete* do menu *File* desempenham operações sobre os arquivos de projeto. A figura 5.3 ilustra a janela *Edit Project* e o menu suspenso *File*.

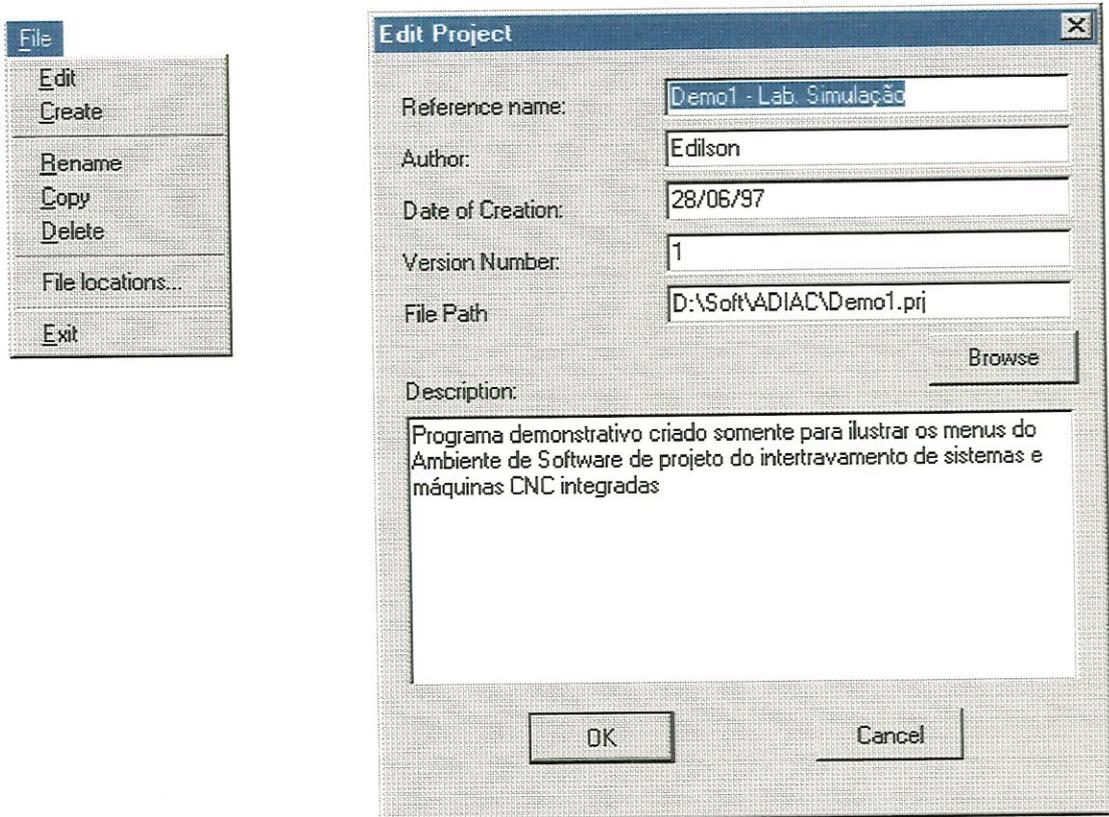


Figura 5.3 - O menu *File* e a janela da base de dados de informações do projeto.

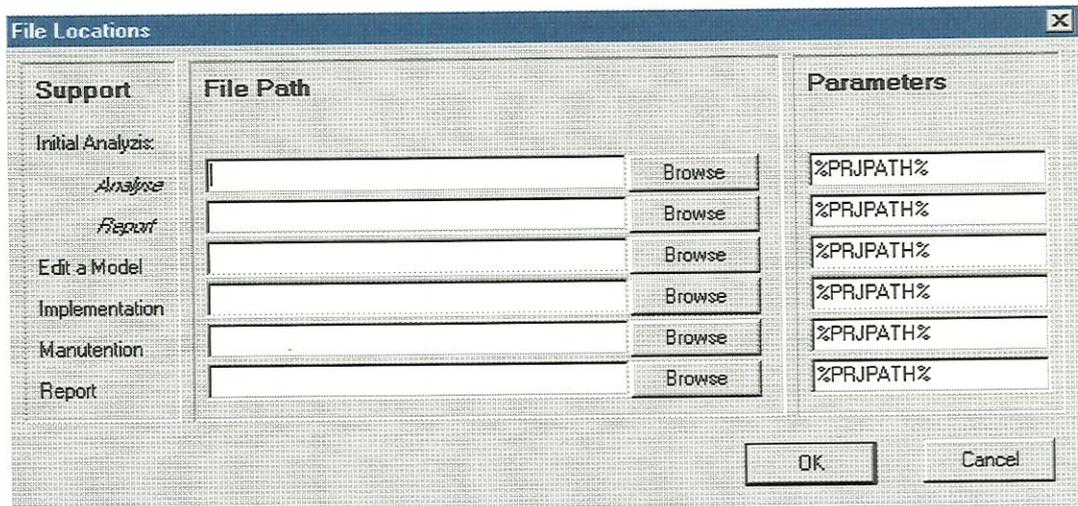


Figura 5.4 - A janela *File locations*.

A opção *File Locations* chama outra janela de diálogo, onde o usuário pode entrar com o caminho (*path*) do editor de redes de Petri, dos S.E. de análise das necessidades e de manutenção, de uma seqüência de implementação e dos locais de documentação. A Figura 5.4 mostra esta janela.

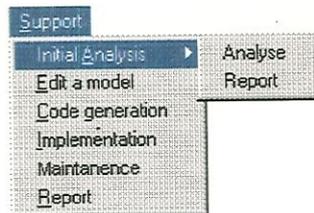


Figura 5.5 - O menu Support

O menu *Support*, descrito na Figura 5.5 consiste na *shell* por si só. O código de cada opção será implementada no decorrer do trabalho na medida que os módulos forem criados. Este menu chama uma função externa de uma biblioteca de ligação dinâmica (dynamic link library - DLL) ou executa um programa externo, tal como o editor de redes de Petri e os sistemas especialistas da análise das necessidades e da manutenção. Portanto no menu *Support* temos:

Initial Analysis - Análise inicial das necessidades;

Edit a model - Edita o modelo em redes de Petri;

Code generation - Geração de código para CLP;

Implementation - Auxílio a implementação;

Maintenance - Auxílio a manutenção.

Report - Documentação do projeto.

Initial Analysis - Executa um programa baseado em um sistema especialista para a “Análise e definição das necessidades” de projeto. É implementado pela *shell* de sistema especialista EXSYS para *Windows* e possui as funções que exhibe quadros de diálogos com as perguntas necessárias para a definição das necessidades de projeto. Possui acesso à base de conhecimentos do S.E. inferindo como resposta uma estrutura básica da máquina a ser projetada.

Edit a model – Executa um programa de edição e simulação de redes de Petri, chamado Petri Net Tools, desenvolvido pelo Laboratório de Simulação de Sistemas Discretos do Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos EESC da Universidade de São Paulo - USP, onde o editor e o simulador desenvolvidos possuem características de modularização de edição e análise temporal, além de fornecer compatibilidade com o programa de conversão de redes de Petri para uma linguagem CLP padronizada (descrito a seguir).

Code generation - Trabalha como um compilador para converter a modelagem realizada em redes de Petri no editor e testada no simulador para uma linguagem de programação CLP padrão. No caso específico do uso do Petri Net Tools, o programa de geração de códigos CLP encontra-se incorporado neste, enquanto que para outros

editores e simuladores podem ser chamados pelo caminho especificado no *File Locations*.

Implementation – Auxilia a implementação do software criado, através do uso das marcações do modelo em redes de Petri, visando estabelecer condições necessárias para o ligamento, “*start-up*”, das máquinas da mesma forma que o seu desligamento em condição segura, “*shutdown*”.

Maintenance - Também executa um programa baseado no EXSYS de um sistema especialista para auxílio a manutenção da máquina, na fase de manutenção, dentro do ciclo de vida do projeto. Este modulo é chamado a partir da necessidade de se corrigir ou alterar o sistema e possui as funções que exibir quadros de diálogos com as perguntas necessárias para o auxílio ao diagnóstico de falhas no sistema. Pode auxiliar na busca de erros ou correções a serem realizadas.

Report – É o responsável pelo endereçamento, exibindo e organizando a documentação. Acessa os documentos gerados durante o processo de projeto dos intertravamentos, tratando principalmente de suas atualizações.

Desta forma o ADIAC é a *interface* programada na Shell Delphi que possui endereçamento de vários módulos executáveis responsáveis pela ajuda ao usuário no desenvolvimento de sistemas de intertravamentos para máquinas CNC. Acessa uma base de dados de projetos, organizando a criação dos projetos quanto a informações, modelos, programação CLP, implementação e documentação. Que permite a criação de um modelo em redes de Petri do intertravamento do sistema de uma forma sistemática e eficiente.

A relação entre os módulos programados no programa de implantação do ambiente é detalhada no anexo A.

5.2.1 - Análise e Definição das Necessidades

A análise e definição das necessidades dentro da *shell* será tratada através de um banco de conhecimentos, utilizando um sistema especialista (S.E.). O S.E. será o responsável em realizar os encaminhamentos na implantação dos intertravamentos do projeto. Estes encaminhamentos são importantes para a definição da parte eletro-eletrônica e em alguns casos da estrutura mecânica da máquina.

Os resultados desta fase de análise e definição de necessidades são uma descrição das funções da máquinas e a indicação dos blocos dos modelos em redes de Petri a serem incorporados para que seja realizada a sua modelagem.

A tabela 5.5 mostra as características de controle de uma máquina. A partir destas características é realizada a estruturação do conhecimento no S.E. para que as respostas sejam inferidas.

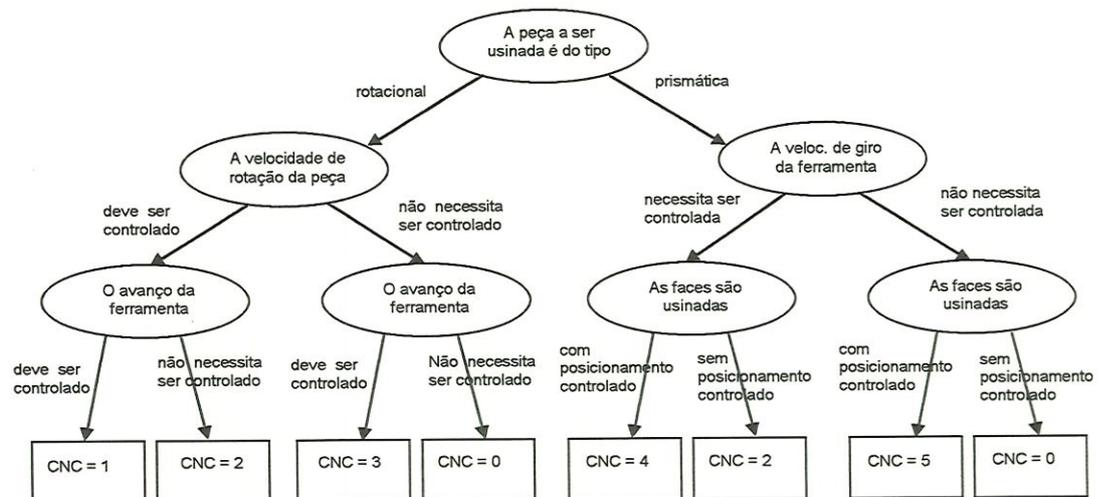
Tabela 5.5 - Características de controle da máquina CNC

Tipo de controle	CNC + CLP CLP
CNC eixos	X e Z lin. + rotacional X, Y, Z lin. + rotacional
Comunicação externa	Serial RS 232C Rede ethernet Map Top Tabela de memória (pontos E/S)
Funções de intertravamentos	Lubrificação Hidráulico Acionamento de motor principal Refrigeração Dispositivos especiais ...
Pontos de E/S	E/S digitais E/S digitais e analógicas
Inicialização da máquina	Rotinas de inicialização Rotinas normais
Eixo árvore	Com controle de velocidade Sem controle de velocidade
Retorno de falha ou falta de energia	Rotina de emergência Reinicialização do sistema Continuar a usinagem
Seqüenciamento	Com mensagem ao operador Sem mensagem ao operador
Manutenção	Com funções para liberação do intertravamento Sem funções para liberação do intertravamento
Lubrificação periódica	Manual Automática
Alarmes	Sinalização no vídeo Sonoro Indicação luminosa local Indicação luminosa na sala de controle
Modos de operação	Manual Manual e Automático Automático c/ operador Automático s/ operador
Segurança (botão de emergência)	Desligamento total Desligamento parcial

A estratégia de estruturação do conhecimento para se especificar quais as características do sistema de intertravamento pode ser representada pelas árvores de decisão descritas a seguir:

Quanto ao sistema de eixos e tipo de controlador:

A estruturação do conhecimento da Figura 5.6 leva o sistema a selecionar as opções descritas para o CNC a ser utilizado. Desta forma sugere a utilização dos blocos de referenciamento e de eixo árvore, de acordo com os eixos necessários.



CNC = 0 - sem utilizar CNC

CNC = 1 - mínimo de dois eixos lineares, com eixo rotacional controlado

CNC = 2 - mínimo de um eixo rotacional controlado

CNC = 3 - mínimo de dois eixos lineares, sem eixo rotacional controlado

CNC = 4 - mínimo de três eixos lineares, com eixo rotacional controlado

CNC = 5 - mínimo de três eixos lineares, sem eixo rotacional controlado

Figura 5.6 - Descrição das características do controlador

Quanto ao modo de operação:

De acordo com a Figura 5.7 o sistema informará os principais modos de operação a serem utilizados na máquina, e como será sua integração no sistema de manufatura.

No caso de um modo de operação sem operador integrado, existe a necessidade de se utilizar o modo de referenciamento automático. Este referenciamento é uma variação do bloco de referenciamento, onde a seqüência é disparada pelo controlador.

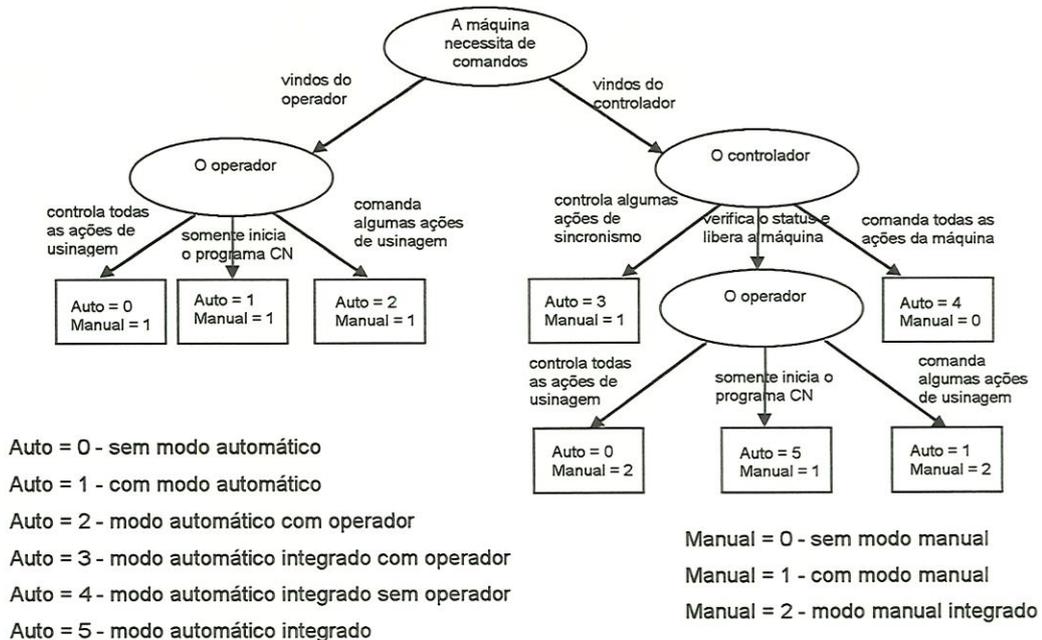
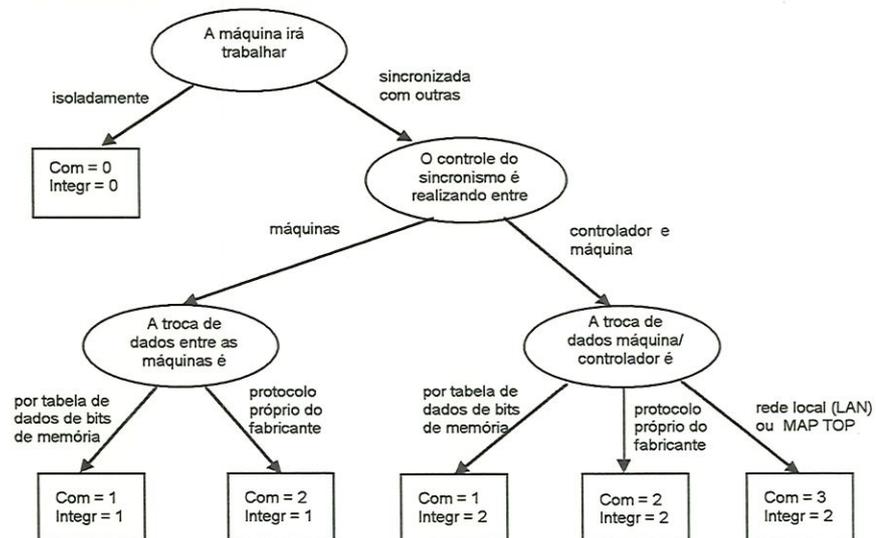


Figura 5.7 - Descrição do modo de operação.

Quanto as características de integração:



Com = 0 - sem comunicação
 Com = 1 - comunicação restrita a tabelas de bits
 Com = 2 - comunicação restrita ao *driver* do fabricante
 Com = 3 - comunicação em rede
 Integr = 0 - sem integração com outras máquinas
 Integr = 1 - com integração simples entre máquinas
 Integr = 2 - com integração a nível de célula

Figura 5.8 - Descrição do tipo de comunicação.

O bloco sugerido pela estrutura de conhecimento da Figura 5.8 é o **bloco de transmissão/ recepção**, que tem como opções a troca de dados por tabela de bits de memória, o *driver* de comunicação do fabricante ou a comunicação via rede local.

O tipo de comunicação vai depender da integração da máquina no sistema de manufatura em que ela faz parte. Quanto mais alto o nível de integração maior o fluxo de troca de dados.

A opção apresentada na Figura 5.9, de carga e descarga de programas CN é gerenciada pelo processador do CNC. A integração desta função com os intertravamentos da máquina vai depender da integração existente no sistema, isto é, da complexidade do sistema de controle adotado no sistema de manufatura.

Quanto a carga e descarga de programas CN:

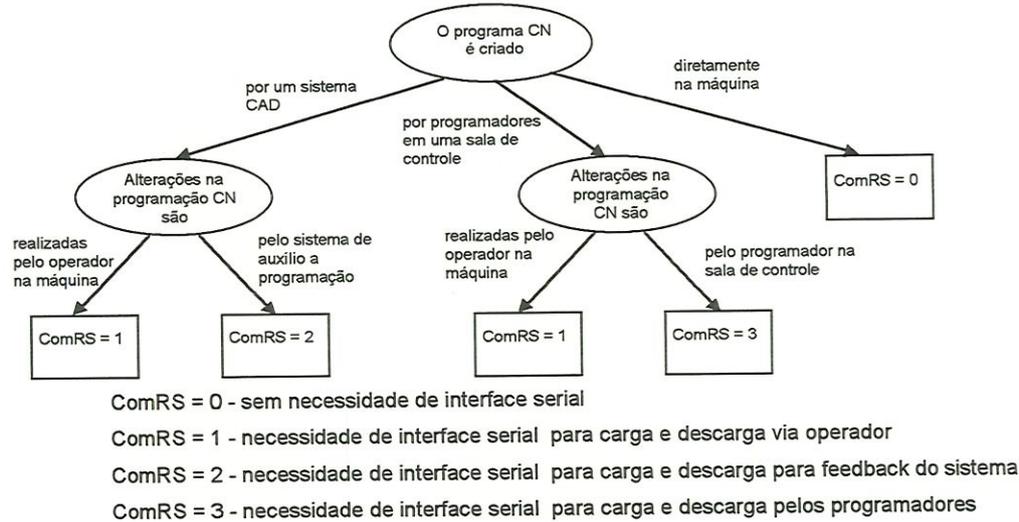


Figura 5.9 - Representação da carga e descarga de programas NC.

A Figura 5.10 descreve o comportamento da máquina na ocorrência de uma falha. O bloco de emergência pode ser utilizado na situação em que se deseja preservar algum sistema ativo na máquina, por exemplo o sistema de alimentação hidráulico, ou quando se deseja colocar a máquina de uma maneira segura, evitando-se tanto danos materiais como humanos.

Quanto a falhas em geral:



Falha = 0 - não necessita de rotina de emergência;
 Falha = 1 - necessita de uma rotina de emergência;
 Falha = 2 - somente o operador pode liberar a máquina.

Figura 5.10 - Representação de ocorrência de falha.

Quanto a partida do sistema:

Na partida algumas condições devem ser satisfeitas tais como o posicionamento inicial para o funcionamento, liberação de algumas funções da máquina somente após o referenciamento dos eixos, condições de lubrificação do sistema (níveis de óleo, pressão de óleo, etc.). As Figuras 5.11, 5.12, 5.13 e 5.14 ilustram a estruturação do conhecimento para a partida.

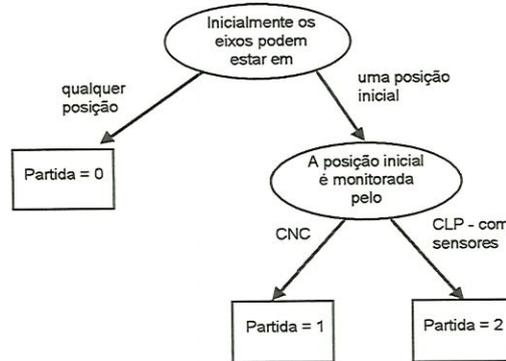


Figura 5.11 - Representação da função de partida quanto a posição dos eixos.

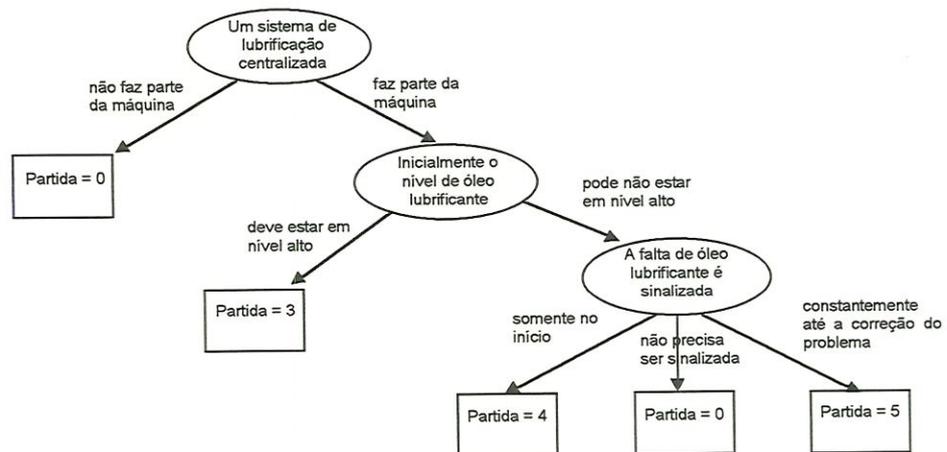


Figura 5.12 - Representação da função de partida quanto a lubrificação temporizada.

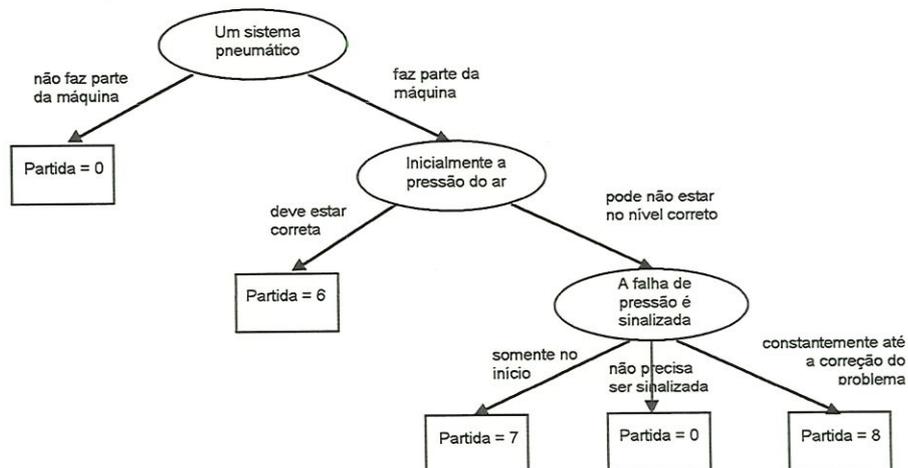


Figura 5.13 - Representação da função de partida quanto ao sistema pneumático.

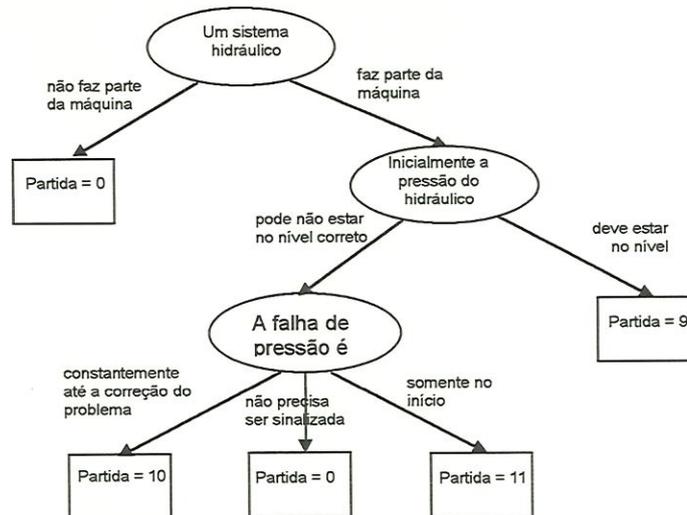


Figura 5.14 - Representação da função de partida quanto ao sistema hidráulico.

As condições para partida são:

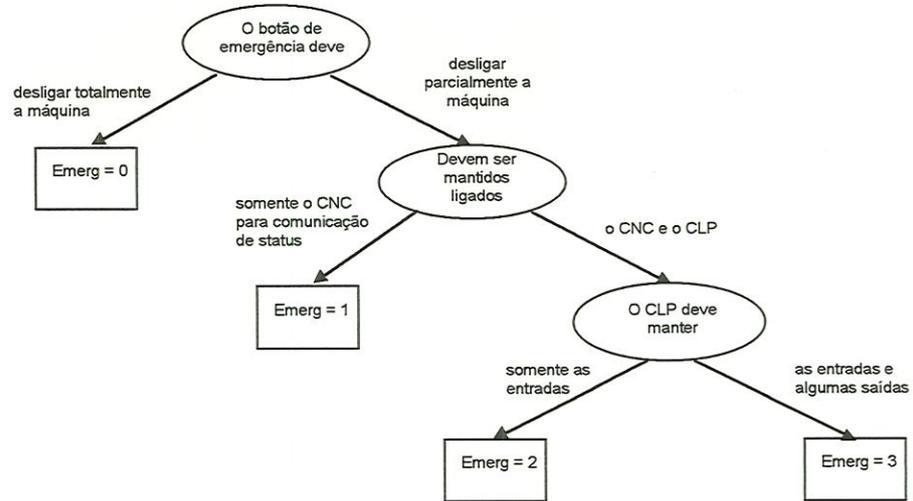
- Partida = 0 - Sem a necessidade de intertravamento de partida
- Partida = 1 - Intertravamento de partida com execução de programa NC
- Partida = 2 - Intertravamento de partida com sensores de posicionamento
- Partida = 3 - Intertravamento de partida com nível alto de óleo lubrificante
- Partida = 4 - Intertravamento de partida com alarme inicial de nível baixo de óleo lubrificante
- Partida = 5 - Intertravamento de partida com alarme de nível baixo de óleo lubrificante
- Partida = 6 - Intertravamento de partida com pressão de ar baixa
- Partida = 7 - Intertravamento de partida com alarme inicial de pressão de ar baixa
- Partida = 8 - Intertravamento de partida com alarme de pressão de ar baixa
- Partida = 9 - Intertravamento de partida com pressão do hidráulico baixa
- Partida = 10 - Intertravamento de partida com alarme inicial do hidráulico baixa
- Partida = 11 - Intertravamento de partida com alarme do hidráulico baixa

Os blocos sugeridos são:

- lubrificação;
- hidráulico;
- acionamento dos eixos;
- emergência

Quanto ao desligamento da máquina:

A Figura 5.15 ilustra a estrutura de conhecimento para o desligamento físico da máquina, de modo que é determinado o tipo de sistema que permanecerá ativo, ou não, dependendo da máquina e das suas condições de segurança.



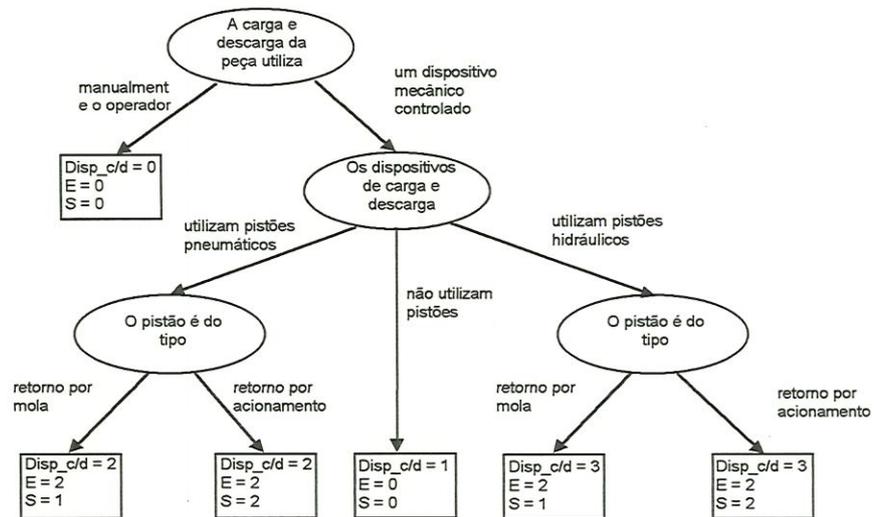
Emerg = 0 - Desligamento total do sistema;
 Emerg = 1 - Desligamento do sistema com somente o CNC ativo para comunicação;
 Emerg = 2 - Desligamento do sistema com o CNC ativo e o CLP com somente as entradas;
 Emerg = 3 - Desligamento do sistema com o CNC ativo e o CLP com entradas e saídas ativas.

Figura 5.15 - Representação do desligamento físico da máquina

Quanto as funções da máquina:

Neste item serão exemplificados as funções de carga e descarga da máquina e de remoção de cavacos, ilustradas pelas Figuras 5.16 e 5.17 respectivamente. Várias outras funções podem ser incorporadas, sendo que podem seguir o mesmo tipo de estrutura de conhecimento.

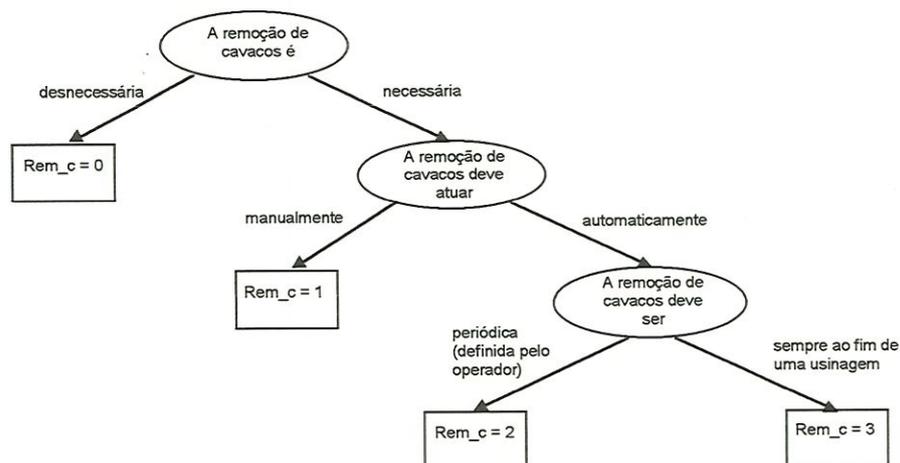
Função: Dispositivo de carga e descarga



Disp_c/d = 0 - Não utiliza dispositivo de carga e descarga
 Disp_c/d = 1 - Utiliza dispositivo de carga e descarga com acionamento manual
 Disp_c/d = 2 - Utiliza dispositivo de carga e descarga com pistão pneumático
 Disp_c/d = 3 - Utiliza dispositivo de carga e descarga com pistão hidráulico
 E = 0, 1, 2... Quantidade de pontos de entrada
 S = 0, 1, 2... Quantidade de pontos de saída

Figura 5.16 - Representação da função da máquina do dispositivo de carga e descarga.

Função: remoção de cavaco



- Rem_c = 0 - Não é necessária a remoção de cavacos na usinagem;
 Rem_c = 1 - Remoção de cavacos comandada manualmente;
 Rem_c = 2 - Remoção de cavacos periódica com tempo fixo;
 Rem_c = 3 - Remoção de cavacos com sinal de fim de programa NC.

Figura 5.17 - Representação da função de remoção de cavacos.

Analisando todas as respostas do projetista através de regras da base de conhecimento do S.E., o sistema fornece ao usuário uma descrição básica das características da máquina. A partir desta descrição básica o usuário pode realizar “ajustes” de acordo com o seu sistema atual e definir o projeto do intertravamento por completo.

Deste modo o S.E. irá auxiliar na tomada de decisão dos seguintes itens:

- 1 - O tipo de sistema a ser implantado;
- 2 - O tipo de CNC a ser adotado (comunicação, numero de eixos, etc.);
- 3 - A quantidade mínima de entradas e saídas digitais/analógicas a serem utilizadas.
- 4 - Que tipo de comunicação deve ser estabelecida;
- 5 - Como vai ser realizada a inicialização do sistema;
- 6 - Tipo de corte de energia em caso de emergência;
- 7 - Quais os modos de operação necessários;
- 8 - Quais os blocos de intertravamento em redes de Petri são necessários.

A forma de resposta do sistema é exemplificada abaixo:

Características gerais fornecidas pelo S.E.:

O sistema de controle adotado é do tipo CNC + CLP

A máquina possui modos de operação: - manual; - automático.

Os eixos a serem utilizados são X, Y e Z lineares

O eixo rotacional é desnecessário

A transferência de programas CN é realizada pela interface serial via operador

A máquina está integrada somente entre máquinas no sistema pela tabela de bits

A partida necessita de intertravamentos

O intertravamento de partida possui:

- execução de programa CN;

- lubrificação inicial pode não estar em nível alto;

- sistema pneumático pode não estar em nível alto e deve ser sinalizado;

- sistema hidráulico deve estar com a pressão alta;

A rotina de emergência é necessária.

Utilizando placas de E/S de no mínimo:

16 pontos de entrada digital

8 pontos de saída digital

4 pontos de entrada analógico

1 pontos de saída analógico

Blocos necessários:

- rotina de referenciamento automático;

- rotina de lubrificação centralizada periódica;

- rotina de refrigeração da peça;

Assim o usuário e o operador devem validar o sistema e o comando numérico adotado, identificando todas as especificações para o projeto do sistema de intertravamento e seu “software”.

São identificadas as funções de intertravamentos que devem ser geradas, através de uma busca na biblioteca de funções de intertravamento, permitindo ao usuário que modele rapidamente a máquina a ser projetada.

5.2.2 – Edição de Intertravamentos utilizando as redes de Petri

Para a edição dos intertravamentos em redes de Petri, foi utilizado o Petri Net Tools, software desenvolvido pelo Laboratório de Simulação de Sistemas Discretos da EESC - USP.

Este software tem como característica o da utilização dos elementos ordinários das redes de Petri, lugares, transição, arcos, pesos e limites, que permitem as análises básicas sobre as características comportamentais e temporais das redes de Petri.

A tela de abertura é ilustrada na Figura 5.18. A qual possui vários menus de edição e gerenciamento de projeto, além dos de conversão e análise. A janela possui vários submenus suspensos descritos abaixo:

File – de gerenciamento de projetos;

Edit – de ferramentas básicas de edição;

View – de visualização das barras de status e zoom do modelo;

Project – de adição e remoção de um arquivo .pnf ou adição de arquivo texto;

Draw – de padrão dos elementos de desenho;

Object – de rebatimento e rotação dos elementos de desenho;

CLP – de conversão do modelo para uma linguagem CLP;

Analysis – de análise das características comportamentais e desempenho;

Windows – de gerenciamento de janelas abertas;

Help – de ajuda.

Os quais possuem algumas teclas de atalho representadas na barra de ícones

loco abaixo:

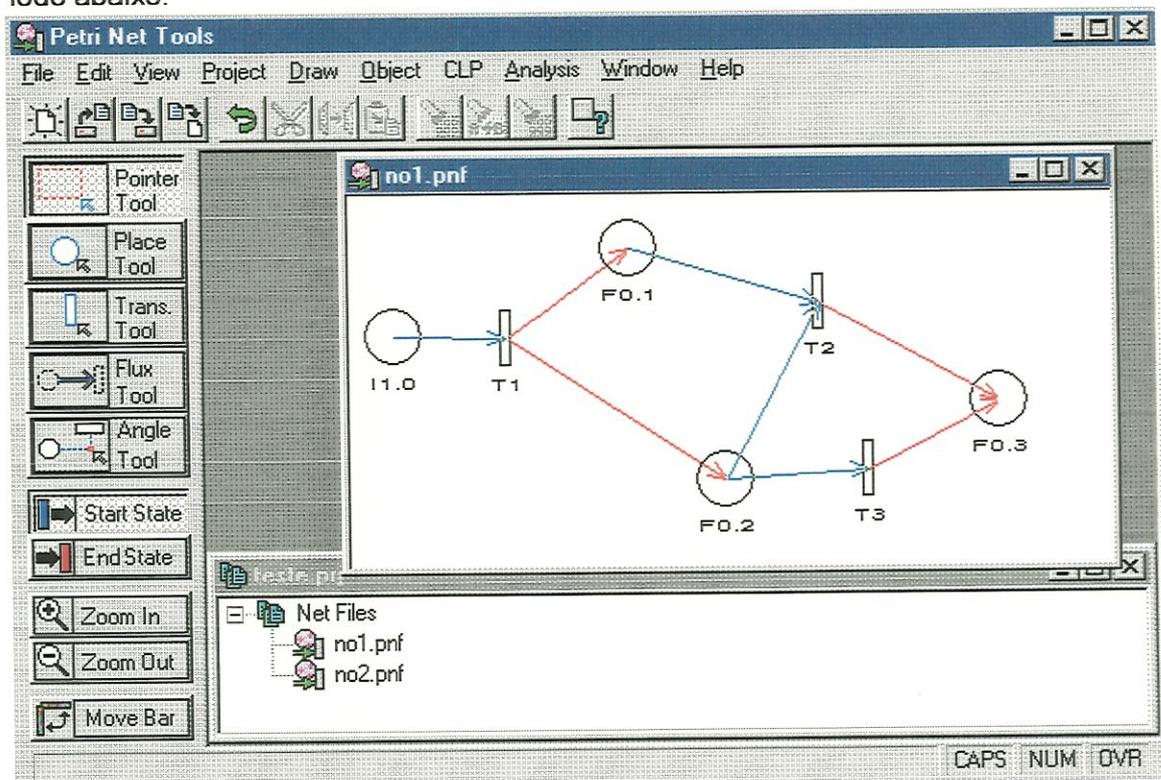


Figura 5.18 - Janela de abertura do Editor “Petri Net Tools”.

A barra de desenho da rede de Petri pode ser movida para as laterais do aplicativo e permite a utilização de ferramentas de desenho básico, tais como:

Pointer tool – seleciona os elementos de desenho do modelo;

Place tool – insere um lugar no modelo;

Trans. tool – insere uma transição no modelo;

Flux tool – insere um arco (fluxo) no modelo;

Angle tool – insere pontos de quebra nos arcos desenhados;

Start State – mostra a marcação inicial do modelo;

End State – mostra a marcação final do modelo;

Zoom in – diminui a visualização do modelo;

Zoom out – aumenta a visualização do modelo

O editor possui uma janela de gerenciamento de projetos, onde um projeto pode conter vários arquivos de modelos editados. Interligando os arquivos, de forma modular estão tanto as transições como os lugares.

Tanto a transição como os lugares possuem uma caixa de diálogo para o carregamento de seus atributos.

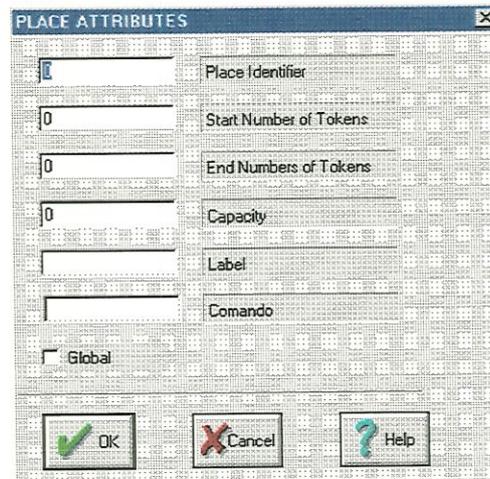


Figura 5.19 - Janela de Atributos de "Place".

Na caixa de diálogo de atributos do "Place", a identificação do lugar é colocado no primeiro campo, e no campo label são introduzidos comentários sobre o lugar, sendo que o número de marcas no lugar é dado no campo "Start number of tokens". O campo "End number of tokens" é utilizado para a marcação a ser atingida na análise das características realizada no menu simulação, como descrito no item anterior. O campo "Comando" é o responsável em determinar as entradas e flags do modelo. O botão de seleção "Global" indica se o lugar existe em outro arquivo, caso contrário é "Local". Desta forma a notação da Figura 5.19 é coberta para cada campo.

Na caixa de diálogo dos atributos da transição "Transition" a sua identificação é colocada no primeiro campo. O campo Time define o tempo de ativação dos lugares de saída para a análise temporal e no campo label comentários sobre esta transição. O campo "Comando" descreve as ações que serão realizadas pelo controlador. E o botão de seleção "Global" indica se o lugar existe em outro arquivo, caso contrário é "Local". Desta forma a notação da Figura 5.20 é coberta para cada campo.

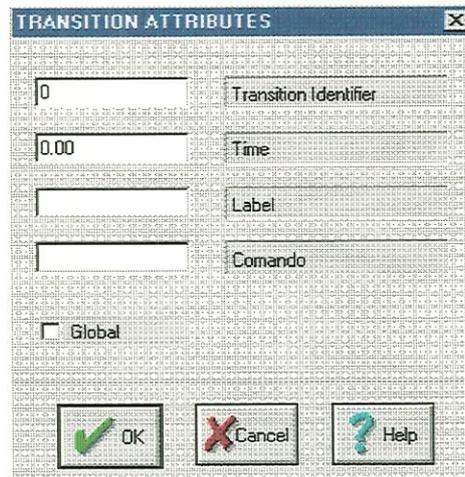


Figura 5.20 - Janela de Atributos de "Transition".

5.2.3 – Simulação dos intertravamentos utilizando as redes de Petri

O projeto do intertravamento de uma máquina ou sistema de manufatura pode ser visto como uma seqüência de eventos discretos. Estes eventos discretos possuem as características de concorrência e paralelismo; operações assíncronas; e de dirigir eventos. Como resultado destas características dinâmicas, podem ocorrer situações de travamento do sistema (*deadlock*) e conflito. A análise qualitativa destas situações serão realizadas através do uso da rede de Petri.

A análise realizada em um modelo de rede de Petri para o projeto de intertravamento pode ser dividida quanto:

- 1 – ao funcionamento correto do projeto a ser implantado, utilizando-se das análises dos "deadlocks", da limitabilidade e de sua reversibilidade;
- 2 – ao desempenho da máquina ou sistema de manufatura, utilizando-se medidas baseadas no tempo;
- 3 – a relação causa/efeito da máquina.

Para estas abordagens a apresentação das propriedades das redes de Petri e de como se deve proceder na sua análise são bastante discutidas por vários autores (ZHOU & DICESARE, 1993; DICESARE et al., 1994, ; VISWANADHAM, et. al, 1990; MURATA, 1989, DESROCHERS & AL JAAR, 1995; JENSEN, 1995; além de outros).

Desta forma torna-se desnecessária a apresentação detalhada destas propriedades e métodos de análise já consagrados e utilizados por vários softwares disponíveis na rede mundial. Assim será descrita a interface utilizada do Petri Net Tools, utilizada nesta sistemática, da mesma forma como foi descrita para o editor de redes de Petri.

A tela principal do editor e simulador de redes de Petri adotado, o Petri Net Tools, é ilustrada na Figura 5.21. O menu de análise e uma descrição de sua atuação é realizada em seguida.

No item **Analysis** encontram-se as seguintes opções no menu:

Automatic – na análise em automático, as marcas irão fluir a partir da marcação inicial M_0 até a marcação final M_d . A marcação inicial pode ser visualizada pelo botão *Start State* e a marcação final pelo botão *End State*, localizados na barra de ferramentas de construção.

Step by step – permite que cada marcação da rede seja visualizada individualmente a partir da interação com o analista, o qual tem o controle de disparo das transições para o surgimento de cada nova marcação.

Time – análise temporal através do uso do gráfico de Gant para o tempo gasto a partir de uma marcação inicial M_0 até se atingir a marcação final M_d .

Bounded / Safe / Reach / All – Todas estas opções são cobertas pela análise da árvore de alcançabilidade e analisam as propriedades de *Boundedness*, *Safeness* e *Reachability* individualmente ou de forma geral como mostrado na Figura 5.22 para o item *All*. Estas propriedades são verificadas através da rede toda a partir da marcação inicial.

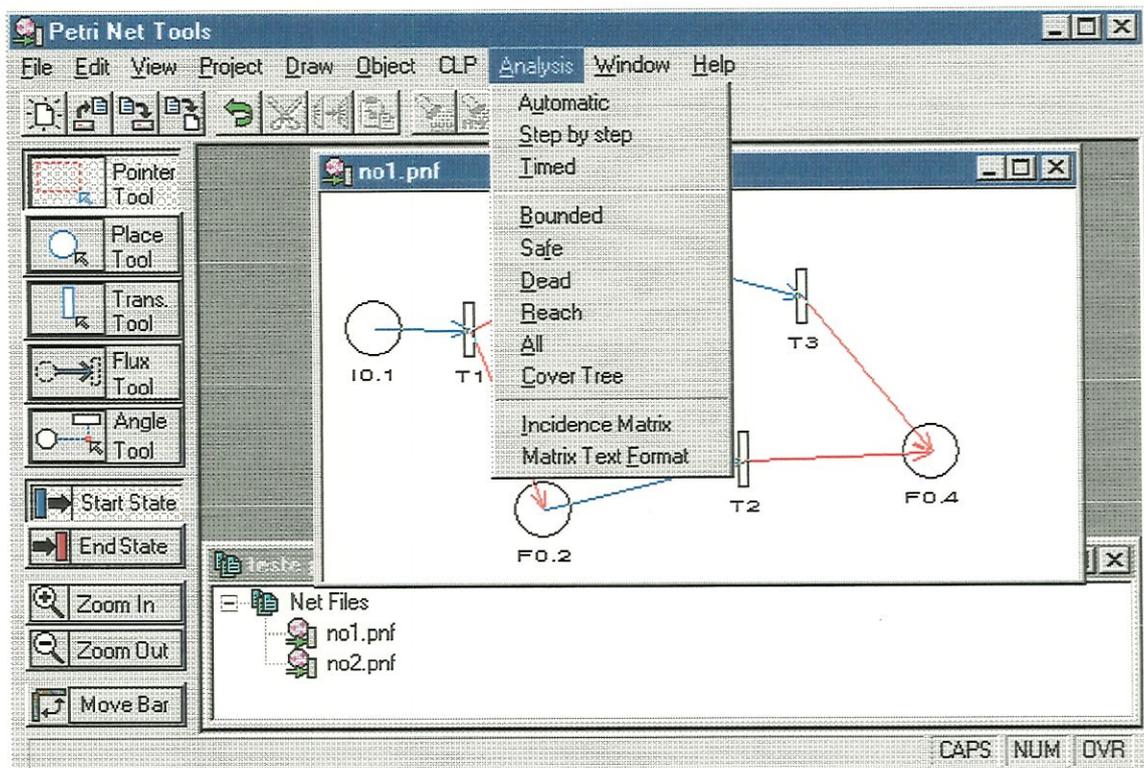


Figura 5.21 – O menu *Analysis* do editor *Petri Net Tools*.

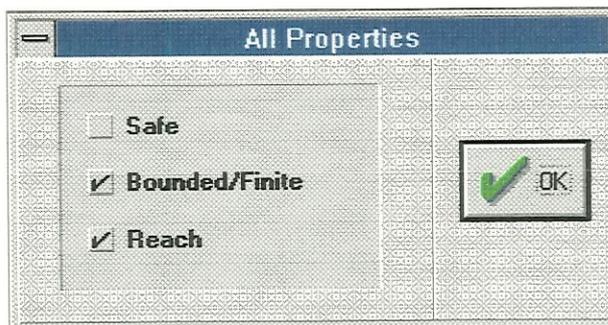


Figura 5.22 - Diálogo que fornece o resultado da pesquisa das propriedades de *Safeness*, *Boundedness* e *Reachability*.

Dead – a propriedade de Liveness é analisada individualmente sobre cada transição e é possível descobrir se uma transição pode ou não ser disparada através da rede de Petri e sua marcação inicial. O acionamento da opção **Dead** ocasiona o acionamento da caixa de diálogo ilustrada na Figura 5.23, que trata-se de uma caixa de diálogo que permite a entrada do identificador da transição a ser pesquisada. Para essa análise foi utilizada a árvore de alcançabilidade.

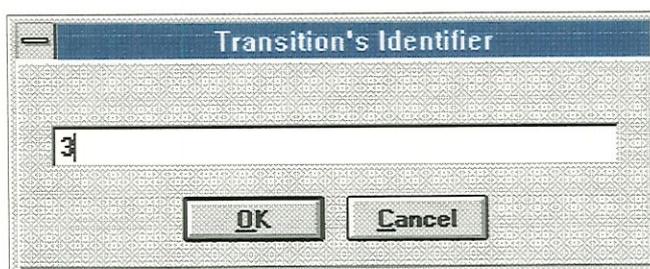


Figura 5.23 - Diálogo de entrada para o identificador da transição a ser pesquisada.

A seguir na Figura 5.24 encontra-se a caixa de diálogo de resposta que fornece o resultado da pesquisa descrita anteriormente, quanto a condição de “dead” ou não de uma transição.

Cover Tree – mostra o arquivo ASCII da árvore de alcançabilidade que é gerado automaticamente, o qual é utilizado para a análise das propriedades baseadas na árvore de alcançabilidade. Estes arquivos ASCII são criados com o mesmo caminho de diretório e nome do arquivo de representação gráfica da rede de Petri, apenas as extensões são trocadas para sua diferenciação. Estes arquivos são gerados juntamente com a simulação gráfica e descreve a árvore de alcançabilidade gerada para a rede de Petri do projeto de intertravamentos.

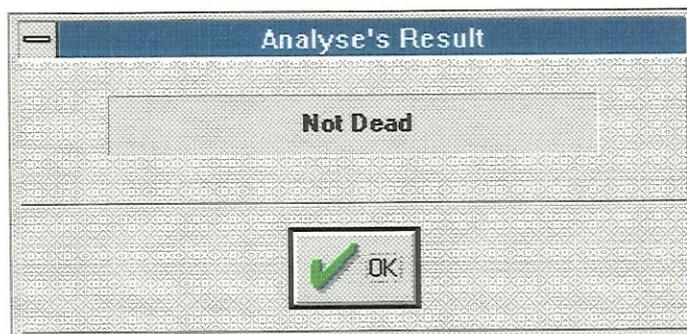


Figura 5.24 - Diálogo que descreve o resultado da pesquisa do item “dead” para a transição.

No item identificador dos lugares encontra-se descrita a numeração dos lugares do modelo, a partir desta ordem é possível a verificação de quantas marcas existem em cada lugar.

A Figura 5.25 ilustra este arquivo para uma rede fictícia e cada linha do arquivo representa um nó da árvore que foi gerado, Desta forma temos as seguintes informações:

- Rótulo do nó de árvore;
- Identificador no nó;
- Marcação da rede de Petri;
- Transição acionadora do nó;
- Marcação imediatamente anterior ao nó (o raiz não possui esse item, nem o anterior, pois não possui nó anterior).

```
// ÁRVORE DE ALCANÇABILIDADE
// Identificadores dos Places:
// M { 1, 2, 3, 4 }
Root M0(2,0,1,0)
New M1(0,1,2,0) { t(1), M0(2,0,1,0) }
New M2(1,1,1,2) { t(3), M1(0,1,2,0) }
New M3(2,1,0,4) { t(3), M2(1,1,1,2) }
New M4(0,2,1,4) { t(1), M3(2,1,0,4) }
New M5(1,2,0,6) { t(3), M4(0,2,1,4) }
Old M6(2,1,0,4) { t(2), M5(1,2,0,6) }
Old M7(1,1,1,2) { t(2), M4(0,2,1,4) }
New M8(3,0,0,2) { t(2), M3(2,1,0,4) }
Old M9(1,1,1,2) { t(1), M8(3,0,0,2) }
Old M10(2,0,1,0) { t(2), M2(1,1,1,2) }
Old M11(3,0,0,2) { t(3), M0(2,0,1,0) }
```

Figura 5.25 – Exemplo de arquivo ASCII da árvore de alcançabilidade

Através desse arquivo torna-se possível ao analista a verificação da árvore de alcançabilidade da rede e de como se procede o disparo das suas transições.

Um outro arquivo ASCII é gerado na mesma situação do anterior, no entanto as informações nele contidas dizem respeito apenas às rotas encontradas para as marcações iniciais e finais definidas pelo usuário.

Este arquivo será útil para a análise de “deadlock” e conflitos do projeto de intertravamentos, pois através dele pode-se identificar as seqüências de funcionamento da máquina em ciclo e as possíveis variações que possam ocorrer. Neste arquivo encontram-se representadas as seqüências de transições a serem disparadas para se alcançar a marca final dada.

No exemplo mostrado na Figura 5.26 as seqüências de disparo das transições se encontram entre parênteses. As rotas (paths) encontrados são delimitados por chaves, e contém o conjunto de marcas, e a transição acionadora desse conjunto de marcas.

Incidence Matrix - Através desse item, aciona-se a verificação se a marcação final M_d fornecida é alcançada a partir da resolução da equação estado que contém a matriz de incidência e o estado inicial M_0 . A caixa de diálogo ilustrada na Figura 5.27 ilustra o resultado dessa pesquisa dizendo se o conjunto de marcas finais pode ser alcançado.

```
Sequence Transition
```

```
(0,3)
```

```
(0,1,3,3,2)
```

```
Init of Path {
```

```
  M(2,0,1,0)
```

```
  t(3) M(3,0,0,2)
```

```
} End of Path
```

```
Init of Path {
```

```
  M(2,0,1,0)
```

```
  t(1) M(0,1,2,0)
```

```
  t(3) M(1,1,1,2)
```

```
  t(3) M(2,1,0,4)
```

```
  t(2) M(3,0,0,2)
```

```
} End of Path
```

Figura 5.26 – Exemplo de arquivo ASCII da seqüência de disparo das transições.

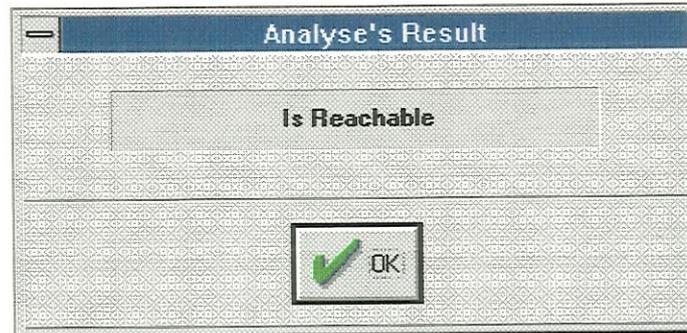


Figura 5.27 - Diálogo resultante da análise por matriz de incidência: marca final fornecida é alcançável.

Matrix Text Format - mostra o arquivo ASCII da equação de estado no formato texto contendo a matriz de incidência, a marcação inicial e final e o vetor u_k , o qual representa o número mínimo de vezes que cada transição precisa ser disparada para se alcançar o estado final, isso para cada transição de 1 a k .

5.2.4– Análise de desempenho

As redes de Petri foram concebidas para representarem o ponto de vista lógico do sistema e das suas dependências causais, sem levar em conta as relações temporais e suas restrições. Desta forma, tornou-se necessário se desenvolver um modo de modelar a quantidade de tempo que um evento gasta para ocorrer, com o objetivo de se realizar análises de desempenho baseadas no tempo (RAMCHANDANI, 1973).

As redes de Petri são uma ferramenta poderosa de análise de desempenho quando o tempo é adicionado ao disparo das transições. Tomando-se possível calcular medidas de produção tais como saída de produto acabado, taxas de produção, e outras medidas temporais que analisam o mérito de um determinado sistema de produção (DESROCHERS & AL-JAAR, 1995).

Nesta proposta estas medidas serão calculadas a partir de modelos de redes de Petri e são direcionadas para os intertravamentos de uma máquina CNC. Isto permite ao projetista (ou analista) tratar diferentes questões. Por exemplo: o que acontece ao tempo de ciclo de uma máquina se um dispositivo é adicionado ao sistema? Qual o tempo de recuperação após uma falha? A substituição de um dispositivo mais lento por um mais rápido pode diminuir/aumentar o tempo de ciclo da máquina?

Usando as redes de Petri temporizadas pode-se avaliar o desempenho do estado estacionário de um sistema de manufatura sobre um processo de produção cíclico e

determinístico. Deste forma será adotado neste trabalho o uso das redes de Petri temporizadas (TPN).

A análise da avaliação do desempenho de uma máquina ou sistema tem por objetivos o de determinar os tempos dos ciclos de execução de uma determinada seqüência de tarefas e de como pode ser diminuído para que a taxa de produção deste sistema possa se elevar. O que pode ajudar a determinar se um investimento em equipamentos automatizados podem ser justificados.

Para isto será implantado um relógio global no sistema e um algoritmo que colocará através do item *time* do menu *analysis* da interface de programação de redes de Petri que irá processar na forma de gráfico de Gant todas as transições disparadas de uma determinada seqüência em relação ao tempo.

A maioria das extensões de tempo das redes de Petri utilizam um relógio global o qual pode ser contínuo (valores reais) ou discreto (valores inteiros). Algumas extensões de tempo enfocam um atraso entre a habilitação e a ocorrência de uma transição, outras tem um atraso entre a remoção da marca de entrada e a criação das marcas de saída, ou o atraso entre a criação de uma marca e o tempo no qual a marca pode ser usada (JENSEN, 1995).

Neste trabalho a extensão de tempo deverá ser capaz de fornecer a noção de processamento de uma determinada tarefa. O tempo de processamento é alocado entre a habilitação e o disparo de uma transição, isto é, depois de habilitada, a transição somente dispara depois do tempo de processamento determinado.

A implantação requer o desenvolvimento de um algoritmo capaz de determinar o instante do disparo das transições de uma determinada seqüência em uma base de tempo e sua apresentação na forma de gráfico de Gant para que fossem analisados os disparos das transições em relação ao tempo.

O simulador do editor de redes de Petri (Petri Net Tools) tem como saída todas as seqüências possíveis para se chegar a um determinado estado final. Esses caminhos são compostos por elementos que contém o número identificador de uma transição e o resultado do disparo desta. Esse resultado é expresso em um vetor montado com o número de marcas em cada um dos lugares.

Foi necessária a criação de uma lista que contém todas as transições a serem disparadas nesta seqüência, uma lista das transições que podem ser disparadas ao mesmo tempo e um vetor que guardasse o tempo de cada uma em relação a um relógio global, sendo possível desta maneira desenvolver o gráfico de Gant para que fosse utilizado pelo projetista para as análises necessárias.

Portanto para a análise do desempenho dos intertravamentos deve-se:

- a) determinar através das análises das características das redes de Petri através dos itens: **reach**, **dead**, **bounded**, **safe** e **all**, se a seqüência da máquina ou sistema de manufatura discreta está correta;
- b) resolver os deadlocks para esta seqüência, a partir de uma marcação inicial M_0 que assegura o ciclo completo da máquina ou sistema, isto é, todas as condições de entrada e de flags devem ser satisfeitas para que da marcação inicial se atinja a marcação final que é o ciclo completo da máquina;
- c) incorporar o tempo de disparo das transições, simulando o tempo de processamento das ações por ela determinadas (são diferentes dos temporizadores utilizados para a programação CLP);
- d) ativar a opção time do menu analysis e automaticamente é gerado, baseado no relógio global, um gráfico de Gant do disparo de todas as transições pertencentes a esta seqüência.

Devem ser analisadas todas as seqüências listadas no arquivo ASCII gerado a partir da marcação inicial para determinar todos os tempos de ciclos da máquina.

Desta forma pode-se analisar quais ações dentro da seqüência podem ser realizadas em paralelo ou como otimizá-las, para que o tempo de ciclo da máquina ou sistema possa ser minimizado, da mesma forma que pode ser avaliado a adição ou retirada de acionamentos ou dispositivos novos.

A Figura 5.28 exemplifica uma análise de desempenho de uma rede simples. Se na figura a transição t_4 tivesse seu tempo de processamento reduzido em 5 segundos, o tempo total também reduziria em 5 segundos, já uma redução no tempo de processamento de t_6 não iria influenciar o tempo total.

Assim, através das redes de Petri temporizada, pode-se chegar a ciclos de máquinas mais otimizados, resultando diretamente no aumento da taxa de peças produzidas, aumentando a eficiência da máquina ou sistema de manufatura analisado.

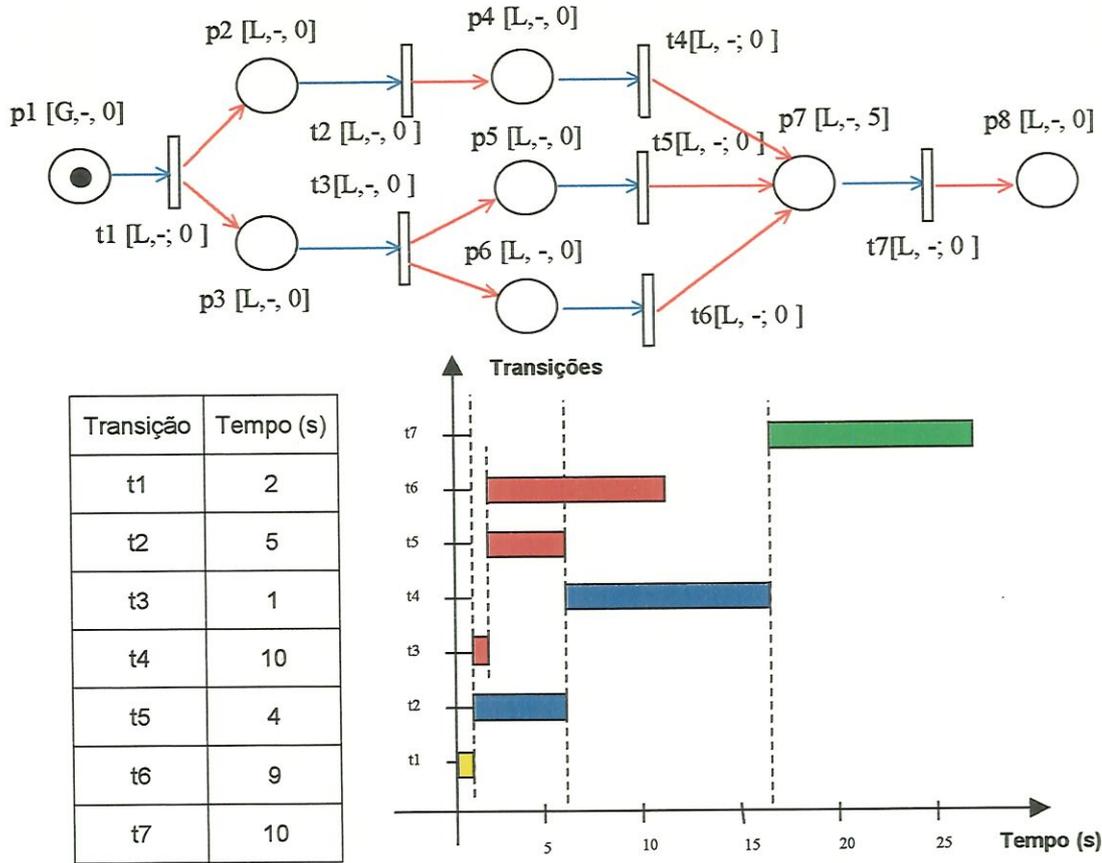


Figura 5.28- Análise de desempenho de uma rede de Petri

5.2.5 – Implementação da programação CLP

O processo de conversão do modelo simulado e validado em redes de Petri para uma linguagem de programação CLP padrão (IEC, 1992), pode ser realizado de duas maneiras: por inspeção e programação manual ou pela geração automática.

A partir da especificação dos componentes de controle de um CLP, tais como contadores, temporizadores, comparadores, etc., pode-se especificar um conversor automático do modelo em redes de Petri para uma linguagem CLP.

E embora a linguagem de programação de diagramas de contatos (Ladder) seja a linguagem de programação mais difundida, será utilizada neste texto a linguagem de programação de lista de instruções (STL). Pois enquanto os diagramas de contatos possuem uma complexidade maior de implantação devido a sua natureza gráfica, a linguagem de lista de instruções não, por ter formato texto.

Para a representação da Lista de Instruções (Statement List – STL), utilizaremos os comandos e operações básicas de acordo com a norma DIN 19 239 (minuta), utilizada em CLPs Siemens (MAXITEC, 1989-a).

Na conversão das redes de Petri para a linguagem CLP tanto na forma manual como automática, utiliza-se as transições como base de conversão, onde esta relacionada a operação booleana AND.

Pois, devido a natureza cíclica da programação CLP, dois conjuntos de instruções podem funcionar como uma operação OR, isto é, o acionamento de uma saída ou flag, pode ser realizado de várias formas no mesmo programa. Estes conjuntos de instruções de acionamento podem, posteriormente, serem simplificados diminuindo assim o tamanho do código gerado.

A forma de conversão do modelo para a linguagem de programação CLP foi descrita no item 4.3 do capítulo 4.

Para a implementação da conversão de uma rede de Petri para a linguagem CLP, no software desenvolvido no LSSD, o “Petri Net Tools”, foram encontrados alguns métodos e classes úteis já existentes no programa fonte do editor de redes de Petri. Algumas alterações e inclusões foram necessárias também no código do programa do ambiente e do editor.

No programa do editor, foi criado um novo menu (“CLP”) para execução da conversão do modelo do intertravamento em redes de Petri. O qual só pode ser visto quando algum arquivo de projeto está aberto. Esse menu possui apenas um item chamado “Convert” responsável pela conversão descrita acima.

Quando o item “Convert” é acionado, é criado um arquivo .TXT com o mesmo nome e no mesmo caminho do arquivo .PNF visualizado na janela ativa. Esse arquivo .TXT contém a lista de comandos em STL proveniente da interpretação da rede contida no arquivo .PNF.

A edição do projeto de intertravamento em redes de Petri, utiliza os modelos dos elementos básicos, descritos no capítulo 4, e os blocos de funções armazenados no banco de arquivos.

Desta forma, é realizada uma diferenciação dos comandos nas transições através de caracteres. Esses comandos representam as ações do CLP.

Na Figura 5.29 abaixo é ilustrado o menu “CLP” com o item “Convert” acionado.

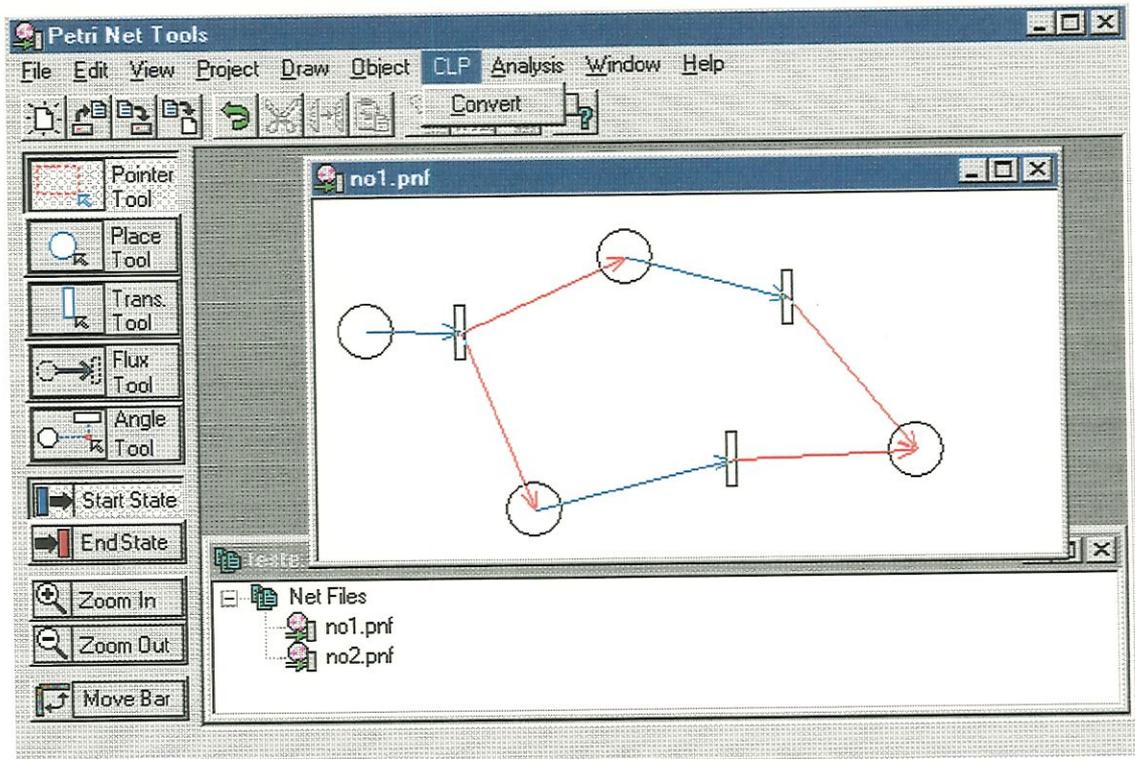


Figura 5.29 - Tela do programa Petri Net Tools com o menu “CLP” acionado.

A notação descrita na modelagem tanto para os *lugares* como para as *transições* é incorporada nas janelas de atributos dos lugares (“Places”) e das transições (“Transitions”), conforme descrito no item 5.2.2.

O campo “Comando” é então utilizado para a conversão como identificador das entradas, flags (para os lugares) e dos contadores, comparadores e temporizadores (para as transições).

Conforme é realizada a interpretação dos dados do campo “Comando”, os resultados já são enviados para o arquivo criado.

Por ser um arquivo texto, com extensão .TXT, ele pode também ser editado com qualquer editor de textos, como por exemplo o *Bloco de Notas do Windows*, caso seja necessário alguma alteração na conversão ou para o acréscimo de comentários.

Após a criação do arquivo para receber a conversão, é necessário carregar a rede de Petri para então obter os seus elementos e assim convertê-los. Para isso, carrega-se apenas o necessário para a conversão: uma lista com os nós (“Places” e “Transitions”) e uma lista com os fluxos.

Desta forma, automaticamente pode-se converter o modelo de redes de Petri em uma linguagem de programação CLP, a qual pode ser incorporada a CLPs que utilizam linguagens de programação padrão (IEC, 1992).

5.2.6 – Documentação do sistema

A documentação é a reunião de todos os documentos que foram sendo criados durante o processo de projeto. Esses documentos podem ser editados, impressos, ou armazenados em disco. É utilizado o bloco de notas do Windows como editor, permitindo o acesso direto aos documentos.

O item documentação facilita a padronização dos projetos de intertravamento, agindo como um elemento concentrador dos documentos que foram gerados no ambiente durante o desenvolvimento do projeto de intertravamentos, evitando-se documentos redundantes e agilizando o processo de implantação e de manutenção.

Os documentos criados durante as fases de projeto e de implementação, como descrito anteriormente são:

- Descrição do projeto;
- Descrição das características da máquina;
- Modelo gráfico em redes de Petri;
- Programação CLP gerada;
- Documentação da simulação;
- Roteiros de partida e de re-ligamento da máquina;
- Descritivo da manutenção.

Descrição do projeto

A descrição do projeto envolve a edição de uma ficha de um banco de dados de projetos onde o projetista pode reunir informações a respeito do nome do projeto, autor, data de criação, versão, caminho dos arquivos do projeto e uma descrição rápida a qual facilita a sua localização.

Descrição das características da máquina

A descrição das características da máquina é gerada a partir do sistema especialista de “definição das necessidades” em função da inferência realizada pelas respostas do usuário as perguntas realizadas pelo sistema especialista.

Modelo gráfico em redes de Petri

O modelo gráfico é representado na forma de vários arquivos, vários módulos, os quais podem ser analisados separadamente, facilitando assim o entendimento do programa de intertravamento da máquina ou sistema de manufatura. De acordo com a implementação da lógica de intertravamentos, são geradas as listas de transições, de lugares, de entradas, de saídas e de flags. São descritos também os sinais de alarme, as

mensagens ao operador do sistema, as funções CNC (códigos de troca de ferramenta -T, códigos de manutenção – H e os sinais miscelânea - M).

Programação CLP

A programação CLP utiliza as linguagens padronizadas como Ladder, lista de instruções (STL), etc., podendo ser aplicada diretamente a maioria dos CLPs comerciais. A documentação extra, como listas de referência cruzada, programação CLP comentada, fluxogramas de funções, podem ser geradas nos equipamentos dos fabricantes e ser incorporadas a documentação do projeto.

Documentação da simulação

A simulação dos intertravamentos em redes de Petri é baseada nas técnicas de árvore de alcançabilidade e matriz de incidência, onde características como liveness, limitabilidade e reversibilidade são tratadas para análise do intertravamento de máquinas. O gráfico de Gant gerado na análise de desempenho também pode ser acessado para a verificação da estimativa de todos os tempos durante o ciclo do sistema.

Roteiros de partida e de religamento da máquina

Na fase de implantação, a partida da máquina, isto é, o seu estado inicial é registrado através da marcação inicial, fornecendo um histórico de como a máquina ou sistema iniciou o seu funcionamento. Isto é importante para a melhor compreensão da manutenção das pré-condições utilizadas para a máquina entrar em ciclo.

Descritivo da manutenção

Um descritivo de como realizar a manutenção da máquina é gerado a partir do sistema especialista de *manutenção*, fornecendo condições de se realizar a manutenção da máquina ou sistema o mais rápido possível.

Desta forma, estes documentos podem ser manipulados e impressos individualmente e são gerados conforme o projeto é desenvolvido.

5.2.7 – Implantação e testes

A fase de implantação e testes é auxiliada com a utilização das simulações baseadas nas marcações iniciais e nas marcações finais testadas para o modelo em redes de Petri.

A partir da análise das marcações da árvore de alcançabilidade, pode-se determinar a preparação inicial para a partida “*start-up*” da máquina da mesma forma que para o seu desligamento em condição segura.

A tabela com a marcação inicial do sistema e a do sistema em ciclo pode auxiliar este consideravelmente o início de funcionamento da máquina. As Tabelas 5.6 e 5.7 exemplificam o relacionamento da marcação inicial com as tabelas de entradas, saídas e flags.

Tabela 5.6 - condições das entradas

Entradas	Lugar	estado	Descrição
I0.0	p1	0	Botão de liga
~I0.1	p2	1	Botão de desliga
I0.2	p6	0	Sensor baixo
~I0.2	p4	1	Sensor baixo negado
I0.3	p7	0	Sensor alto

Um aplicativo pode relacionar as marcações do arquivo gerado na simulação com uma a tabela de pontos de entrada, saída e flags (E/S/F) do sistema, gerando estados que podem ser testados para cada situação diferente na máquina.

Tabela 5.7 - condições dos flags

Flags	lugar	estado	Descrição
F0.0	p2	1	Sistema desligado
F0.1	p5	0	Sistema ligado
F0.2	p8	1	Início de ciclo
F0.3	p9	0	Motor ligado para baixo
F0.4	p10	0	Motor ligado para cima
F0.5	p11	0	Posições diferentes de 5
F0.6	p12	0	Posição igual a 5
F0.7	p16	0	Comparação igual a 5

O projetista tem em mãos assim, tabelas de disposições de pontos de E/S/F que permitirão testar o funcionamento da máquina conforme simulado, caso necessário.

5.2.8 - Manutenção do *Software* de Intertravamento

Na fase de manutenção do *software* de intertravamento, será utilizado um sistema especialista para direcionar o usuário a localizar e corrigir as falhas no menor intervalo de tempo possível.

Tabela 5.8 - Características manutenção de uma máquina CNC

Tipo de parada	Falha Manutenção programada Alteração da programação
Tipo de alteração da programação	Inserção de equipamento novo Alteração da seqüência
Tipo de falha	De inicialização da máquina De intertravamento de seqüência Elétrica Mecânica
Falhas de Inicialização da máquina	Lubrificação Hidráulico Acionamentos Comunicação
Falhas de intertravamento de seqüência	Falta de referenciamento de eixos Seqüência incorreta em manual
Falha Elétrica	Falta de energia Supervisão do acionamentos dos Eixos Sobrecarga nos motores
Falha Mecânica	Fins de curso atingidos Pressão do ar insuficiente Líquido refrigerante abaixo do nível Falta de lubrificante

Desta forma a base de conhecimentos deverá inferir alguns “procedimentos de emergência” baseados no estado de parada da máquina descrito a partir das respostas do usuário as perguntas do S.E..

A Tabela 5.8 sintetiza as principais características de manutenção de um sistema.

A estratégia de resolução do problema de se especificar quais as características do sistema de intertravamento pode ser representada por árvores de decisão que são descritas a seguir.

A árvore de decisão da Figura 5.30 descreve a seqüência das questões e atitudes a serem tomadas pelo usuário para vários tipos de manutenções a serem realizadas. A atuação do usuário nas falhas é de grande importância, para que o sistema retorne com o menor “downtime” possível.

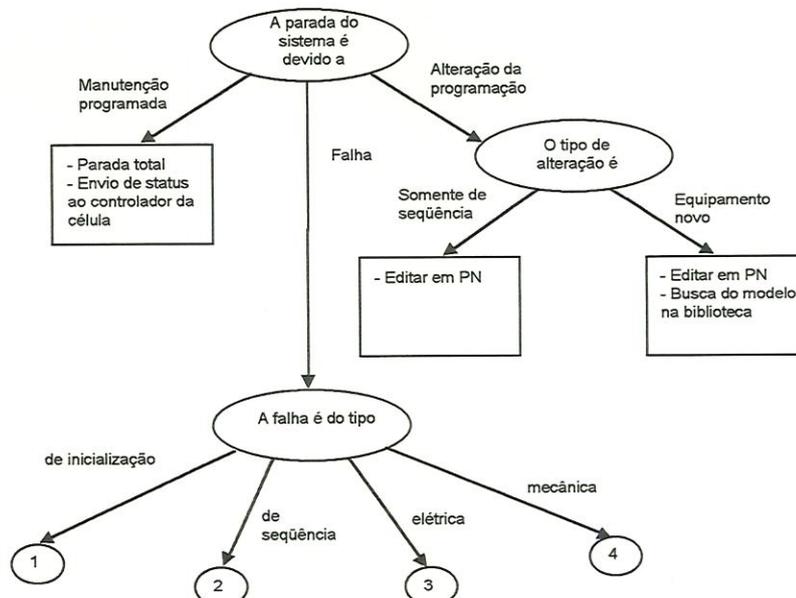


Figura 5.30 - Árvore de decisão principal de manutenção do software de intertravamento

Na manutenção programada, a parada do sistema segue um cronograma previsto. A equipe de manutenção conhece a data de parada e de retorno do funcionamento da máquina. Os tipos de alteração da programação são:

- 1) somente alteração da seqüência;
- 2) inclusão de equipamento novo no sistema.

Na alteração da seqüência o programa é editado novamente, enquanto que na inclusão de um novo equipamento, novos blocos de intertravamento em redes de Petri são procurados na biblioteca.

Quanto a falha de inicialização

A Figura 5.31 mostra as ações que devem ser tomadas quando o sistema pára em função de falha na inicialização. Desta forma o usuário pode reagir a falha encontrada na inicialização.

Na inicialização a falha pode ocorrer na lubrificação, no hidráulico, na comunicação e nos acionamentos dos eixos.

As Figuras 5.32, 5.33 e 5.34 descrevem os tipos de falha de seqüência, elétrica e mecânica, descrevendo as ações a serem executadas.

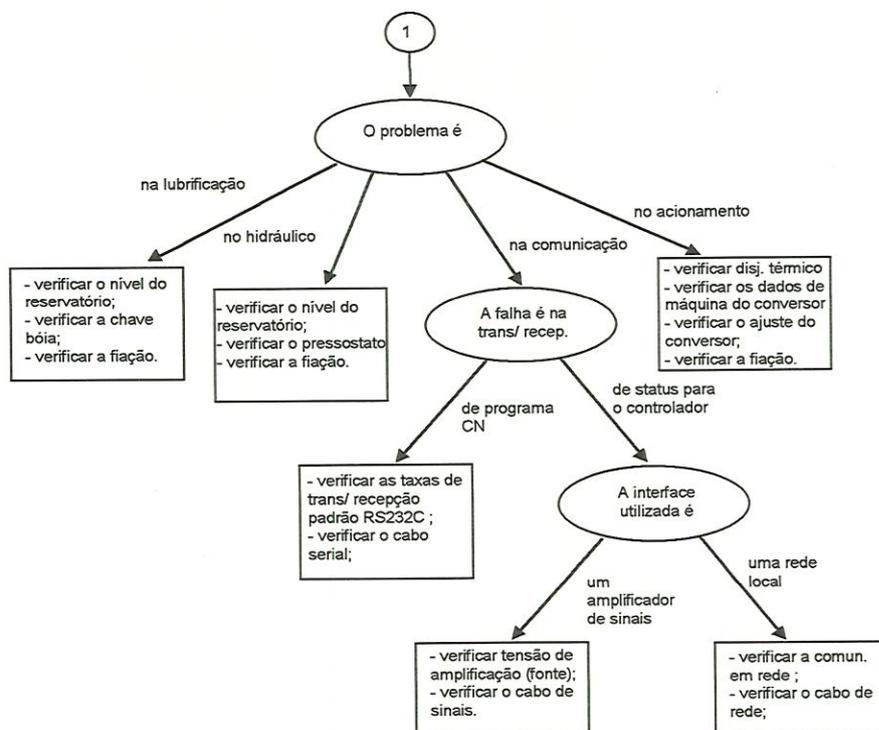


Figura 5.31 - Árvore de decisão da falha na inicialização da máquina.

Quanto a falha de seqüência

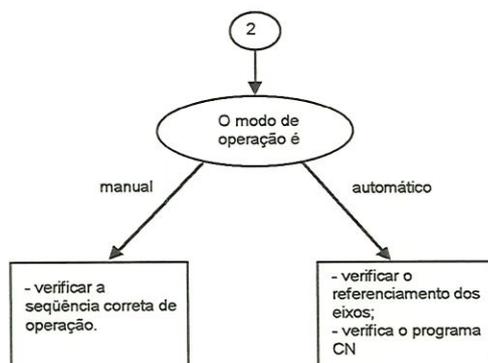


Figura 5.32 - Árvore de decisão da falha de seqüência

Quanto a falha elétrica

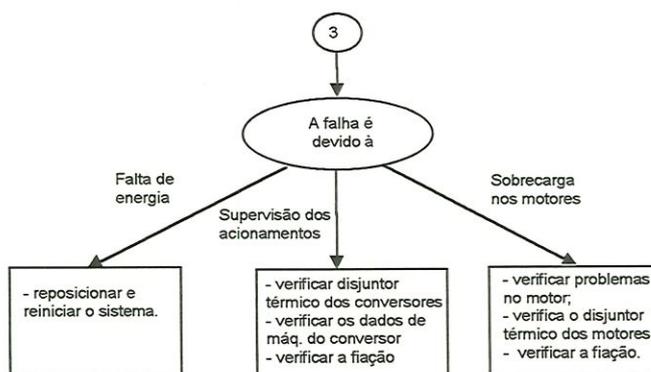


Figura 5.33 - Árvore de decisão da falha elétrica

Quanto a falha mecânica

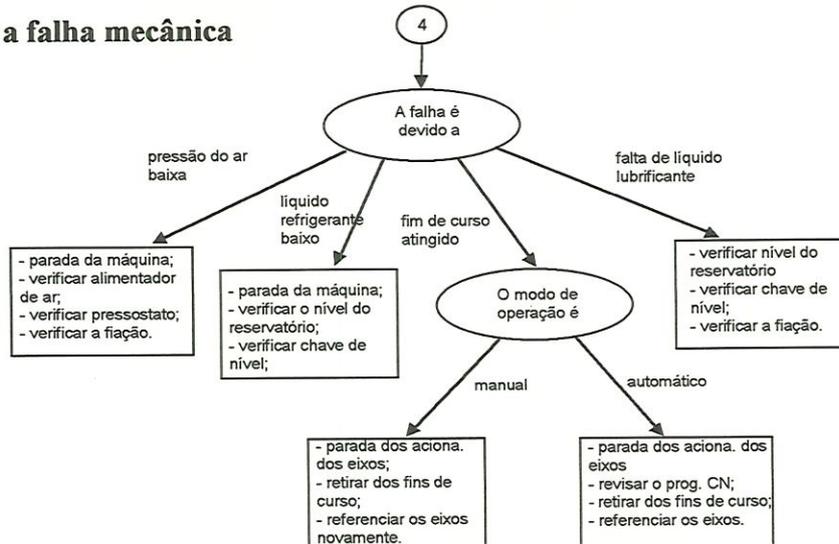


Figura 5.34 - Árvore de decisão de falha mecânica.

A resposta do sistema é dada pelo tipo de falha que ocorre. E pode ser descrita da seguinte maneira para um exemplo de uma máquina integrada a uma célula de manufatura com controlador.

Fins de curso atingidos:

Pára o programa CN imediatamente;
Enviar mensagem de alarme ao controlador;
Verificar programa CN;
Retirar os eixos do fim de curso;
Referenciar;
Corrigir programa e transmitir a máquina;
Retirar a peça e colocá-la em um "buffer" de refugos;
sincronizar o sistema;

Falha na energia elétrica:

Iniciar o sistema;
Referenciar;
Retirar a peça e colocá-la em um "buffer" de refugos;
sincronizar o sistema;
Iniciar o programa CN em modo automático.

Falha de sobrecarga nos acionamentos:

Parar os movimentos da máquina;
Comunicar a falha ao controlador;
Verificar o disjuntor térmico;
Verificar o ajuste do conversor;
Verificar a fiação.
Referenciar

Falha na supervisão dos acionamentos dos eixos:

*Parar o programa CN;
Comunicar a falha ao controlador;
Verificar o disjuntor térmico;
Verificar os dados de máquina do conversor;
Verificar o ajuste do conversor;
Verificar a fiação.
Referenciar*

Falha na pressão do ar:

*Parar o programa CN;
Parar o eixo árvore;
Recolher os eixos sem falha para posição segura;
Verificar fornecedor de ar;
Verifica o pressostato;
Verificar a fiação;
Retirar a peça e colocá-la em um "buffer" de refugos;
sincronizar o sistema;*

Falha na refrigeração da peça:

*Parar o programa CN;
Parar o eixo árvore;
Recolher os eixos sem falha para posição segura;
Verificar o nível do reservatório;
Verificar chave de nível;
Verificar a fiação;
Retirar a peça e colocá-la em um "buffer" de refugos;
sincronizar o sistema;*

Falha na lubrificação:

*Comunicar falha ao controlador;
Verificar nível do reservatório
Verificar chave de nível;
Verificar a fiação.*

Utilizando o diagnóstico do S.E., o operador terá uma ferramenta de auxílio para reagir a uma falha existente no sistema. Evitando que a máquina fique parada por mais tempo que o necessário.

6 – VALIDAÇÃO DO AMBIENTE PROPOSTO NA IMPLEMENTAÇÃO DO PROJETO DE INTERTRAVAMENTO DE UM TORNO CNC.

Como exemplo é realizada a modelagem de um torno CNC. Neste exemplo serão observadas as etapas de modelagem e simulação dos intertravamentos propostos na sistemática de projeto.

Assume-se que durante a produção, problemas como falta de energia, falha na alimentação de ar, intervenção errada do operador, etc. podem ocorrer. A simulação destas falhas irão validar as condições de intertravamento projetadas para a máquina ferramenta.

6.1 – Descrição do sistema

O sistema é um torno CNC simples, composto de:

- 3 (três) eixos X, Z + rotacional (eixo árvore);
- Placa pneumática para prender a peça;
- Bomba de refrigeração da peça;
- Sistema hidráulico;
- 1 (uma) ferramenta;
- Lubrificação das guias;

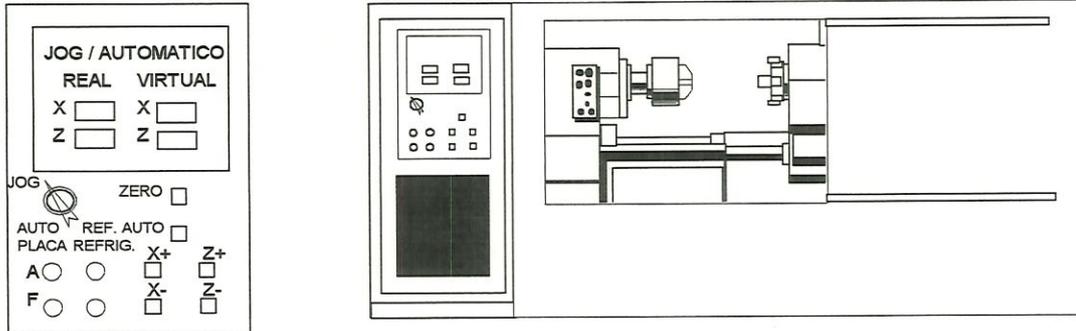


Figura 6.1 - Visão frontal do torno e da interface homem máquina

Utilizando a análise das necessidades, o sistema especialista analisa todas as respostas do projetista através de regras da base de conhecimento. O sistema fornece ao usuário uma descrição base das características da máquina.

Características gerais fornecidas pelo S.E.:

O sistema de controle adotado é do tipo CNC + CLP

A máquina possui modos de operação:

- manual;
- automático.

Os eixos a serem utilizados são X e Z lineares

O eixo rotacional é necessário

A transferência de programas NC é realizada pela interface serial via operador

A máquina está integrada somente entre máquinas no sistema pela tabela de bits

A partida necessita de intertravamentos

O intertravamento de partida possui:

- lubrificação inicial pode não estar em nível alto;
- sistema pneumático pode não estar em nível alto e deve ser sinalizado;
- sistema hidráulico deve estar com a pressão alta;

A rotina de emergência é necessária.

Utilizando placas de E/S de no mínimo:

- 24 pontos de entrada digital
- 8 pontos de saída digital
- 0 pontos de entrada analógico
- 0 pontos de saída analógico

Blocos necessários:

- rotina de inicialização
- rotina de referenciamento automático;
- rotina de acionamento do hidráulico
- rotina de acionamento da refrigeração da peça
- rotina de lubrificação centralizada periódica;

A partir da validação da especificação básica do sistema e o comando numérico adotado, são identificadas as funções de intertravamentos que devem ser geradas, através de uma busca na biblioteca de funções de intertravamento, permitindo ao usuário que modele rapidamente a máquina a ser projetada.

Na modelagem utilizando as redes de Petri, os intertravamentos podem ser descritos segundo sua funcionalidade. Deste modo são definidos para a modelagem da máquina CNC os seguintes intertravamentos:

- intertravamento de organização;
- intertravamento de inicialização;
- intertravamento da seleção de modo de operação;
- intertravamento de emergência;
- intertravamento de comunicação;
- intertravamento de referenciamento; e
- intertravamento dos acionamentos em geral.

Estes intertravamentos representam a estrutura básica de controle da máquina CNC adotada, descrita no capítulo 5, para este ambiente de desenvolvimento de projeto de intertravamento de máquinas. Esta estrutura permite que o projetista desenvolva outros componentes particulares à máquina, os quais são agregados a estes intertravamentos.

Sua modelagem segue o formato descrito no capítulo 4 (tabela 4.1), com a adição de um parâmetro gráfico (L: local ou G: global) que representa se a transição ou lugar pertence somente àquele gráfico (arquivo) ou aparece também em outro. Do ponto de vista das análises das redes de Petri não há alteração nenhuma.

São apresentados os modelos de controle do CNC aplicado ao torno CNC, os quais podem ser utilizados em qualquer máquina ferramenta, que possua este tipo de característica, sendo que os componentes de controle definidos no capítulo 4 (comparadores, temporizadores, contadores, etc.) irão compor os intertravamentos da máquina. Assim os intertravamentos são apresentado de uma forma modular (várias funções em arquivos diferentes). Estas funções são descritas a seguir.

6.1.1 - Intertravamento de organização

A Figura 6.2 descreve o modelo básico em redes de Petri do intertravamento de organização. Neste modelo, a máquina executa rotinas de inicialização e entra em regime de funcionamento podendo estar em qualquer um dos modos de operação. Os modos de

operação citados no modelo básico são: modo de referenciamento, modo de comunicação, modo manual e modo automático.

As ações comandadas pelo sistema de manufatura são realizadas em modo automático. Neste modo é possível a execução de programas em linguagem "G" padrão ISO, que são transmitidas pelo controlador e iniciadas por ele.

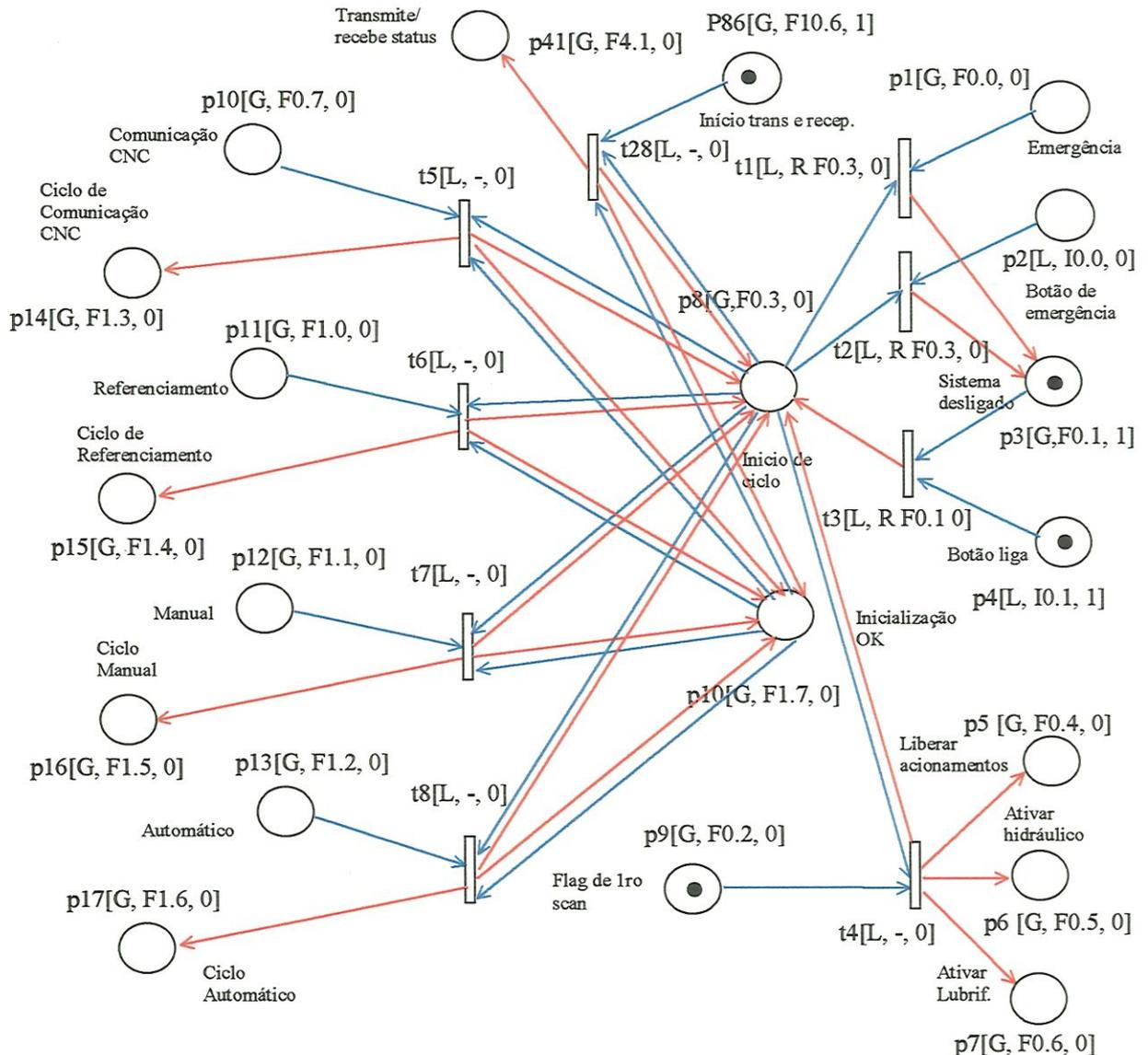


Figura 6.2 - A representação do intertravamento do bloco de organização em redes de Petri.

Durante a execução de operações de usinagem, o controlador pode comandar as ações da máquina necessárias ao seu funcionamento automático em um ambiente de manufatura. O controlador recebe o "status" de cada passo do programa CLP e caso ocorram falhas, gera as condições necessárias de sinalização a outros níveis da hierarquia de controle, por exemplo para a reprogramação rápida de uma atividade em

outra célula, gerenciando e controlando as necessidades atuais e futuras para o bom funcionamento do sistema. Através dos sinais atuais da usinagem é possível gerenciar outros dispositivos e componentes da célula, o que permite trabalhar de forma síncrona, ágil e com o menor "downtime" possível da célula.

O responsável por esse sincronismo e pelas atuações adequadas ao sistema (atuação) é o CLP, que tem a função de coordenar o meio externo e todos os elementos necessários para a execução da usinagem.

A partir da execução do referenciamento, o CLP envia o "status" ao controlador e ao CNC e as operações no modo automático são habilitadas.

O modo manual mostrado na Figura 6.2 pelo lugar P16 é o responsável pela preparação da máquina para que o sistema possa operar em automático. Neste modo de operação da máquina ferramenta, são permitidas as seguintes ações:

- a) movimentar o magazine (funções T);
- b) realizar qualquer acionamento individualmente (funções H);
- c) indicar o ponto zero para as peças a serem usinadas;
- d) introduzir ferramentas;
- e) realizar qualquer movimento nos eixos;
- f) acionar o eixo árvore (códigos "M");
- g) abrir e fechar a porta de alimentação;
- h) ligar e desligar a refrigeração da peça;
- i) acionar a lubrificação das guias;
- j) ligar e desligar o transportador de cavacos;

Estas funções são acionadas pelo preparador de máquina, através da interface homem máquina (IHM) do CNC. Sendo os sinais repassados ao CLP, que atua nas saídas necessárias.

A maioria dos acionamentos, no modo manual, são liberados sem intertravamentos. Isto facilita o trabalho da manutenção, que pode atuar na máquina sem restrições, agilizando sua preparação e manutenção. No entanto, há funções que exigem intertravamento, devido ao risco de danos ao equipamento. Ações como girar trocadores de ferramentas, devem ser realizadas sempre em posições seguras para o operador e para a máquina.

Os lugares p1, p2, p3, p4 e p8, Figura 6.2, determinam quando o sistema está ligado ou desligado, de acordo com os estados dos flags F0.0, F0.1 e F0.3 e as entradas I0.0 e I0.1. No modelo, com o sistema ligado, o flag F0.2 é ativado somente na execução do primeiro loop do CLP (primeiro scan) fazendo com que as rotinas de inicialização

como liberação de eixos, acionamento do hidráulico e da lubrificação, lugares p5, p6 e p7 respectivamente, sejam executadas em paralelo. O fim da execução de todas as rotinas de inicialização é sinalizada pelo flag F1.7, permitindo que a máquina prossiga executando outras rotinas.

Após a inicialização são ativados vários ciclos de controle da máquina, representados pelos lugares globais p14, p15, p16 e p17, selecionados a partir de entradas mutuamente exclusivas ativadas como descrito na Figura 6.3 que resultam nos flags F0.7, F1.0, F1.1 e F1.2 representados pelos lugares p10, p11, p12 e p13 que por sua vez são ativados pelas entradas I0.6, I0.7, I0.5 e I0.4 respectivamente.

A rotina de transmissão e recepção de status ao controlador da célula, é ativada pelo flag F4.1 e F10.6, representado pelo lugar global p41 e p86 respectivamente. Esta rotina é ativada após o sistema ter inicializado e é executada em paralelo com todos os outros intertravamentos da máquina. A figura 6.7 apresenta em detalhes a rotina básica de transmissão e recepção de status.

6.1.2 - Intertravamento de inicialização

A preparação da máquina tem a função de colocar o sistema em estado seguro de partida, preestabelecido pelas exigências globais do sistema. Esta rotina é efetuada apenas uma vez quando se liga a máquina, e dependendo do tipo de máquina CNC, o CLP pode realizar varias tarefas. Abaixo estão descritas as rotinas mais comuns que podem ser representadas através do uso de redes de Petri.

- liberar os acionamentos;
- ligar o hidráulico;
- acionar a lubrificação das guias;
- Posicionar o trocador de ferramentas;
- Abrir/fechar porta de alimentação;
- Inicializar sistema de fixação da peça.

Na Figura 6.2 tem-se, os intertravamentos de inicialização são ativados a partir dos *lugares globais* P5, P6 e P7, através dos flags F0.4, F0.5 e F0.6, representando a liberação de acionamentos, a ativação do hidráulico e a ativação da lubrificação, respectivamente. Outras tarefas podem ser incluídas ou não dependendo do projeto da máquina.

A Figura 6.3 descreve o intertravamento de inicialização de liberação dos servoacionamentos dos eixos, da ativação da lubrificação e da ativação do hidráulico,

necessários para o funcionamento correto da máquina em ciclo. A liberação dos eixos é executada na transição t9 e a confirmação é verificada na entrada I0.2 representada pelos lugares p18 (negado) e p20. Caso não seja confirmada a liberação, p18 com marca, há uma falha nos acionamentos, representado pelo lugar p24, flag F2.2 e uma rotina de emergência é ativada. Analogamente à liberação dos acionamentos, funciona a ativação do hidráulico representada pela ação da transição t12 e pelo sinal de entrada I0.3 nos lugares p23 (negado) e p21, em conjunto com o flag de falha no hidráulico F2.6 (lugar p28).

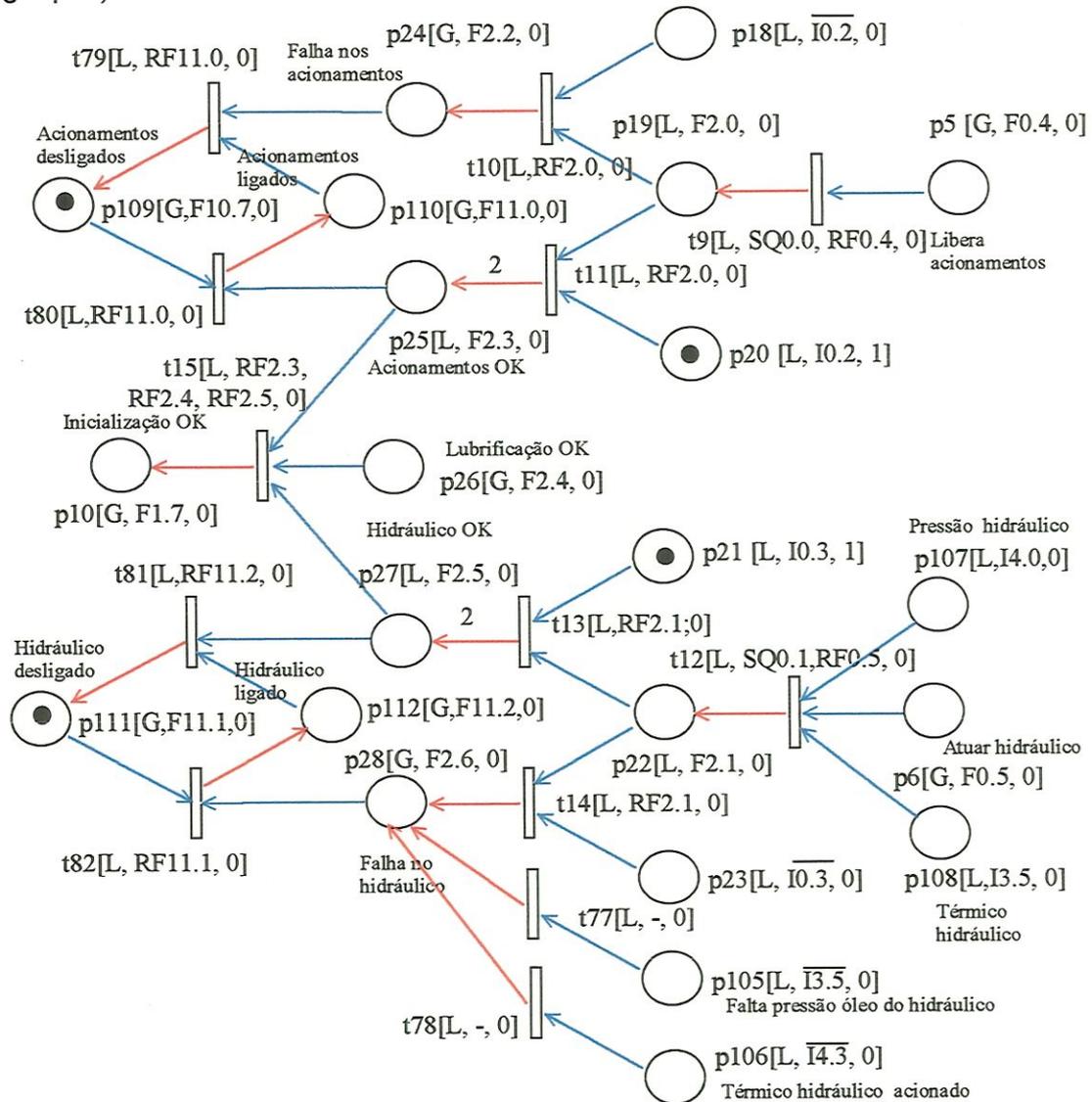


Figura 6.3 - Representação do intertravamento de partida dos acionamentos e do hidráulico

Os sinais dos flags F2.3, F2.4 e F2.5 representados pelos lugares p25, p26 e p27 respectivamente, indicam as condições normais dos acionamentos, da lubrificação e do hidráulico, ativando deste modo o lugar global p10, flag F1.7 permitindo que a máquina prossiga sua operação normal.

temporizador T2 e sendo ligado novamente em seguida. Este ciclo é repetido por três vezes. Após terminar os ciclos de lubrificação, o intertravamento é habilitado para os modos manual, lugar global p40, e automático, lugar global p13, em conjunto com um tempo de espera de 2 horas (T3). Ao mesmo tempo que é ativado o flag F2.4 do lugar p26 de lubrificação OK.

6.1.3 - Intertravamento de seleção do modo de operação

O modo de operação será detectado através de uma chave seletora de posições mutuamente exclusivas. O números de posições e modos de operação podem variar de acordo com o fabricante do CNC. A Figura 6.5 representa o intertravamento de uma chave seletora básica de quatro posições, geralmente encontradas em CNCs fabricados pela SIEMENS, GE FANUC e ALLEN BRADLEY. Outros modos de operação podem ser incluídos se necessário.

Pode-se intertravar os modos de operação utilizando-se *flags* de acordo com a necessidade do sistema. Os *flags* são utilizados pelo CNC e programados no CLP, para evitar que mudanças acidentais do modo de operação, possam comprometer o prosseguimento normal de execução do programa CNC no modo automático, evitando o que a peça seja danificada. O lugar p50 da Figura 6.5 indica a não execução do programa NC através do flag F6.2 negado (este flag é encontrado disponível em uma tabela de memória comum entre o CNC e o CLP e denominado “programa nc ativo” (MAXITEC, 1989-b)).

Os lugares p42, p46, p51 e p55 da figura 6.5 representam o estado dos sinais de entrada I0.4, I0.5, I0.6 e I0.7 que irão ativar os modos de operação automático, manual, de comunicação e de referenciamento, respectivamente, sendo cada modo de operação diferente e mutuamente exclusivo em relação ao outro. Para isso são usados dois flags, um de ativação dos lugares globais p13, p12, p10 e p11 (F4.2, F4.4, F4.6 e F5.0 respectivamente) de cada modo de operação, e outro de desabilitação dos outros modos de operação, F4.3, F4.5, F4.7 e F5.1 respectivamente.

Os flags F7.1, F7.2, F7.3 e F7.4 fornecem a informação do modo desabilitado e são representados por lugares globais p45, p49, p54 e p58, podendo serem utilizados para compor lógicas em outros intertravamentos.

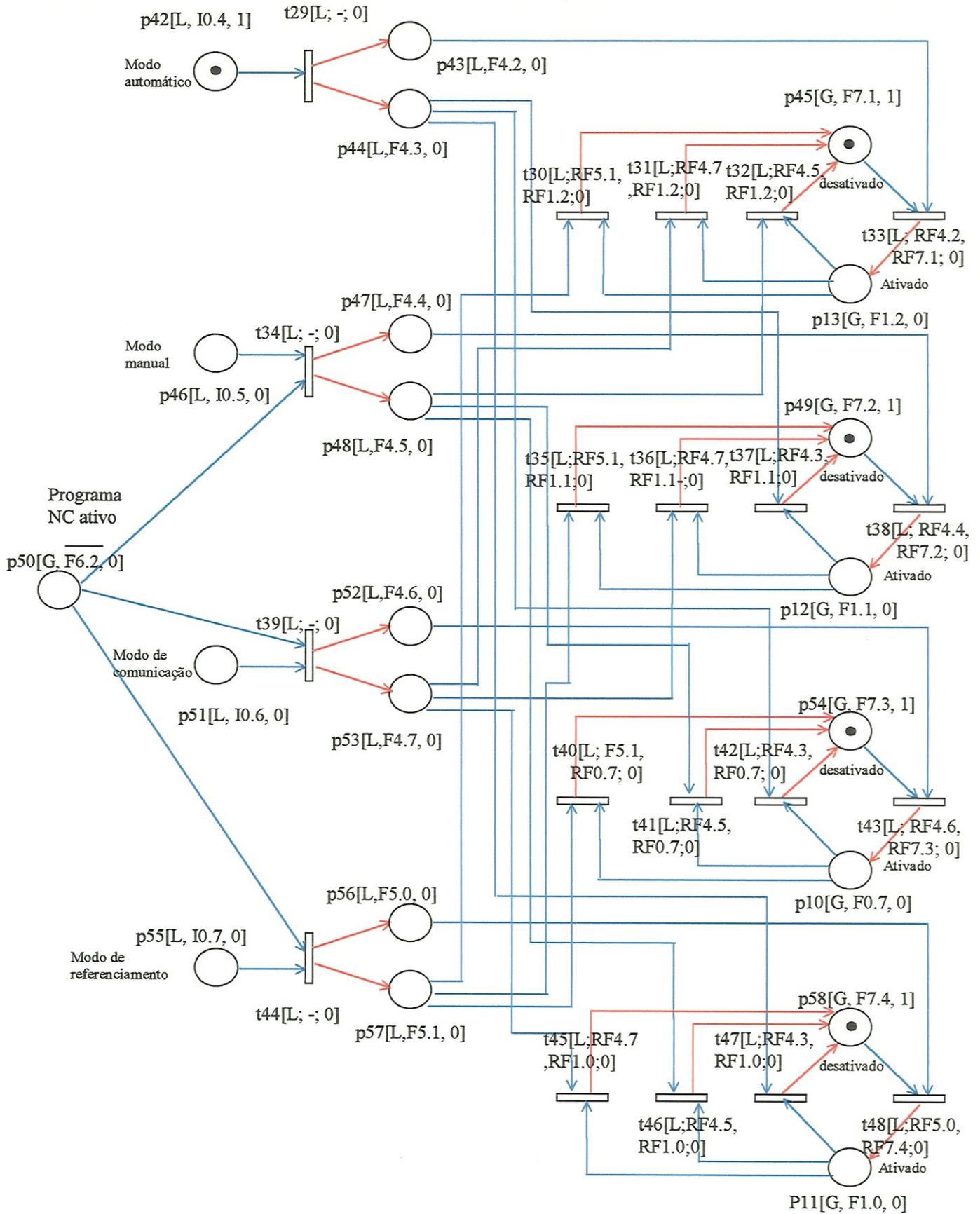


Figura 6.5 - Representação do intertravamento da seleção de modo de operação.

6.1.4 - Intertravamento de emergência

A rotina de emergência é aquela que irá realizar o monitoramento de alguns sinais que são vitais para o funcionamento do sistema. Dependendo da natureza da atividade que entra em alarme, a conduta pode ser diferente, sendo que para alguns problemas, não é necessário interromper um processo imediatamente. Por exemplo, uma falha na lubrificação das guias pode esperar pelo término da usinagem de uma peça. Em alguns casos torna-se necessário a parada imediata do sistema sob o risco de acidentes com a máquina e/ou operador. As providências a serem tomadas, nestes casos, podem ser:

- desligar o eixo árvore;
- desligar o transportador de cavacos;
- abrir/fechar a porta;
- desativar os eixos de usinagem ou colocá-los em uma posição segura;
- abortar o programa de usinagem;
- desligar/reiniciar o sistema

Na Figura 6.6 é descrita em redes de Petri a rotina de emergência para uma falha na unidade hidráulica indicada pelo lugar p28 com o flag F2.6 . No ato da falha, são ativados os flags F5.2, F5.3, F5.4 e F10.0, indicando nos lugares p88, p90, p92 e p96 as atividades de se abortar o programa NC, posicionar os eixos de forma segura e desligar o mancal, respectivamente. Também são ativados os códigos M (miscelânea) de desligamento do eixo árvore (M05) e de desligamento da refrigeração da peça (M09).

Um sinal de falha específico do hidráulico é ativado pelo flag F10.0 fazendo com que o CNC mostre uma mensagem de falha no vídeo ao mesmo tempo que é enviada ao controlador da célula.

As atividades de se desligar o eixo árvore, posicionar os eixos de forma segura e desligar a refrigeração possuem suas confirmações nos lugares p93, p94, p84 e p95, através de seus respectivos flags. O desligamento do mancal, ativado pelo lugar p92, flag F5.4 possui uma inércia de 15 segundos, no caso deste modelo, assim a lubrificação do mancal ainda permanece ligada, evitando-se danos no mancal.

Daí, a partir das condições indicadas nos lugares p93, p94, p95, p96, p84 e p88, a emergência é ativada e o sistema é desligado.

Este tipo de bloco pode ser aplicado a maioria das falhas existentes no sistema, porém algumas pequenas variações podem ocorrer. Além disso, dependendo da falha, a necessidade de executar uma seqüência de atividades pode ser facilmente incorporada neste modelo pelo uso do editor.

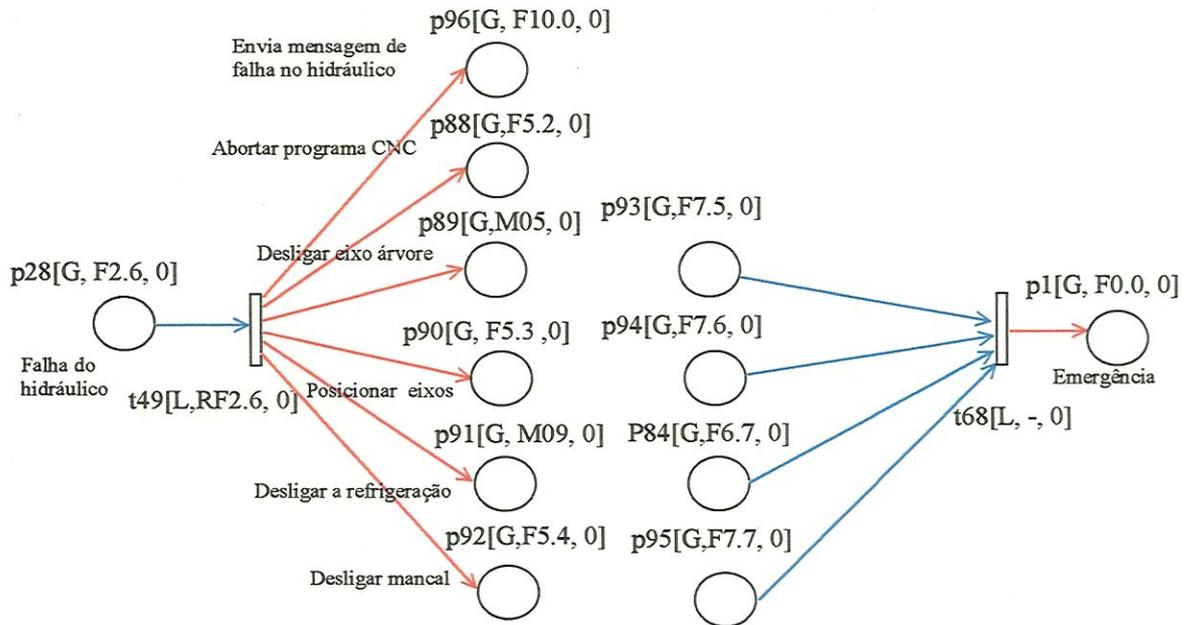


Figura 6.6 - Rotina de emergência geral em redes de Petri.

6.1.5 - Intertravamento de comunicação

A comunicação entre a máquina e o controlador da célula é mantida pelo CLP. O qual envia e recebe sinais de *status* do controlador. A máquina neste caso pode estar em qualquer modo de operação, até mesmo no modo automático usinando uma peça.

A transmissão e recepção de *status* é responsável pelo sincronismo entre máquina CNC e o controlador da célula. E é realizada na área de memória, da onde se pode ter desde sinais básicos de entrada e saída como sinais de confirmação de tarefas.

A Figura 6.7 mostra que a partir de sua ativação pelo flag F4.1, representado pelo lugar p41, a transmissão e a recepção estão habilitadas, através dos lugares p59 e p64 respectivamente, de acordo com as tarefas a serem recebidas e o status a ser enviado para o controlador da célula. A entrada I1.7 habilita a transmissão e I2.0 representado pelos lugares p60 (negado) e p63 confirmam ou não o recebimento da transmissão.

Em caso de falha o flag F5.6 do lugar p62 é ativado, executando uma rotina de emergência, a qual não necessita o desligamento imediato do sistema pois a máquina pode funcionar de maneira normal sem comunicação externa.

Na recepção a máquina solicita informações ao controlador da célula através do flag F5.7 (lugar p64) e após recebê-las, é analisada sua confirmação utilizando o flag F6.1, representado pelos lugares p66(negado) e p68. No caso de uma falha na recepção, o flag F6.2 (local p67) é ativado e o comportamento é análogo ao da transmissão.

As condições de funcionamento normal, tanto na transmissão como na recepção, ativam novamente o lugar p41, correspondente ao flag F4.1, reiniciando-se o ciclo de transmissão/recepção de status.

O modo de comunicação descrito na Figura 6.2 pelo lugar p16 tem a função de realizar a carga e descarga de programas CN. A comunicação estabelecida obedece padrões internacionais (EIA e IEC) e não é função disponível para a programação do usuário.

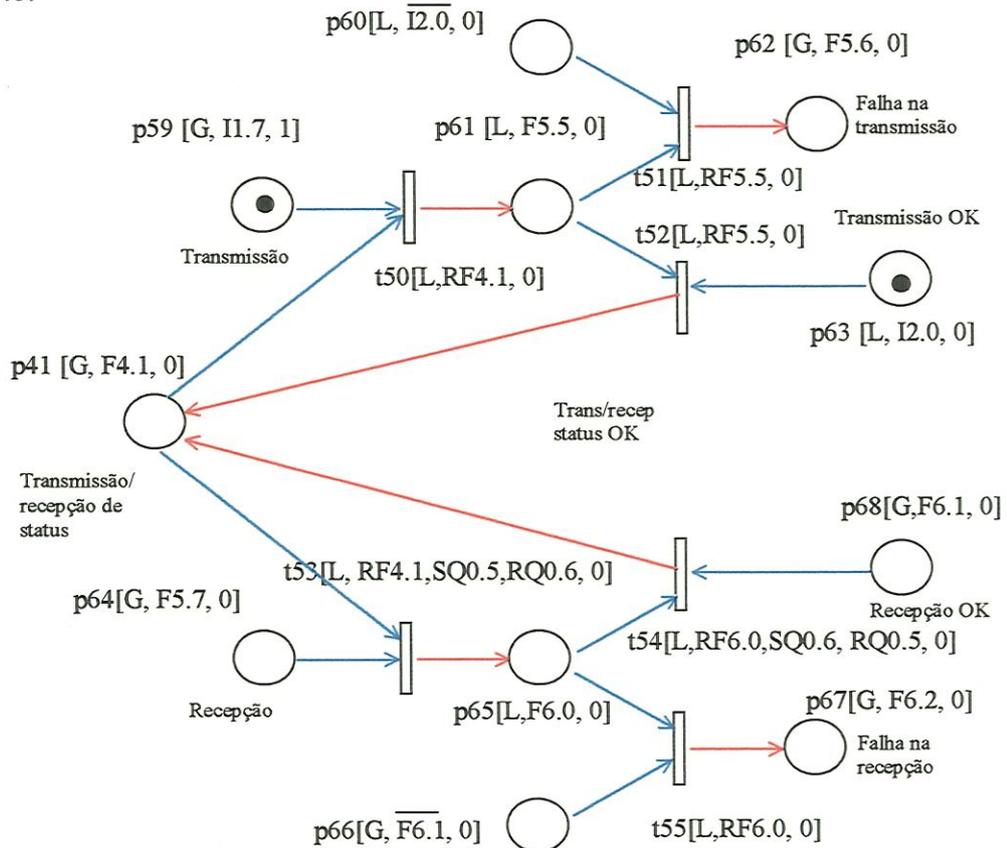


Figura 6.7 - Rotina de Transmissão/recepção de status em redes de Petri.

6.1.6 - Intertravamento de referenciamento

Para que a máquina possa estar disponível para trabalhar em modo automático, é necessário que o sistema de medição seja referenciado. O referenciamento permitirá a execução de um programa CN que está armazenado na memória do CNC, pois os ajustes de posição não serão alterados em relação a uma peça a ser usinada. Assim o referenciamento da máquina fornecerá ao sistema de medição do CNC, a posição inicial necessária à repetibilidade do sistema de coordenadas (MACHADO, 1989).

O sistema de referenciamento de eixos utiliza três sensores para cada eixo, um fim de curso positivo "+", um fim de curso negativo "-" e um fim de curso de referência

“ref”. Os sensores estão colocados ao longo do eixo de movimentação da máquina de acordo com a Figura 6.8. O comportamento dos eixos pode ser visto na forma gráfica da rede de Petri da Figura 6.9.

O bloco de referenciamento para o eixo X de uma máquina CNC, ilustrado na Figura 6.9, onde os lugares p69, p70 e p74 ilustram as entradas dos sensores de fim de curso positivo, negativo e de referência de entradas I1.2, I1.4 e I1.3, respectivamente.

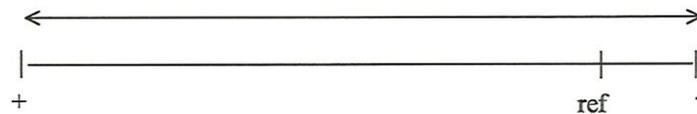


Figura 6.8 - Sinais para o referenciamento de um eixo.

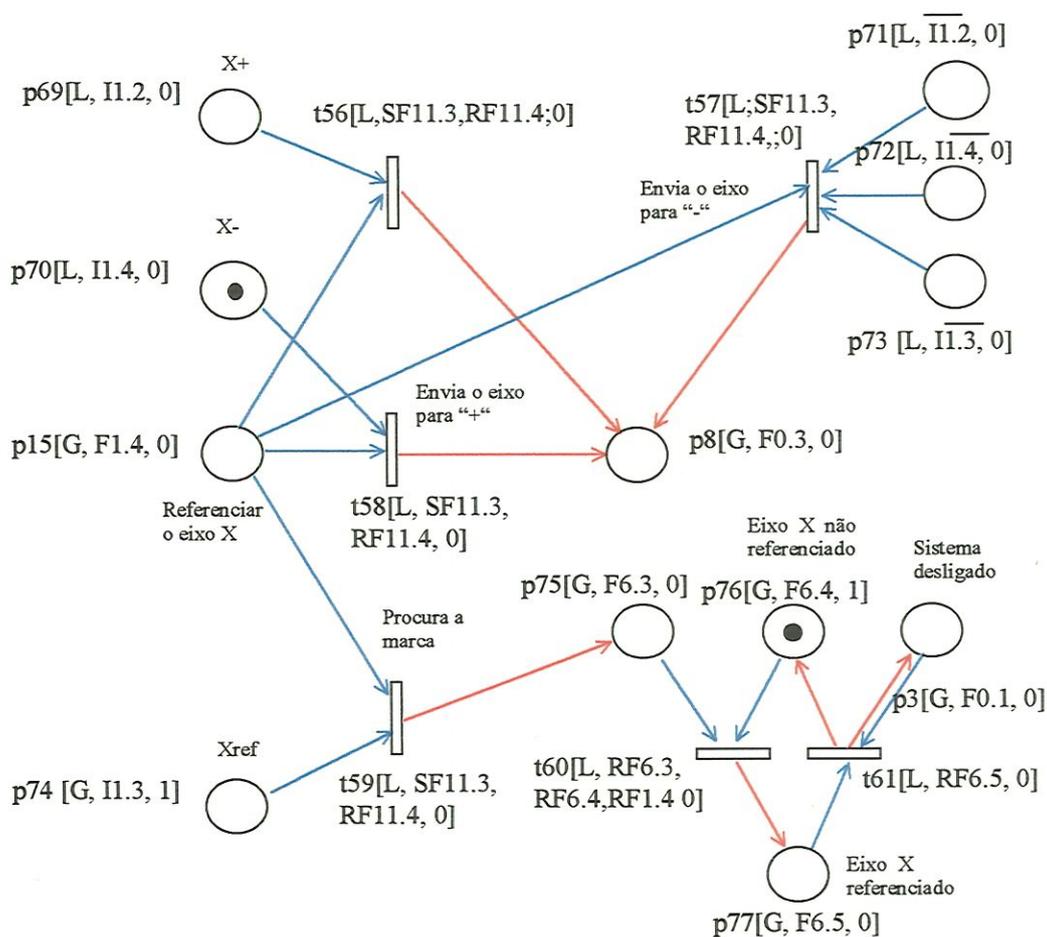


Figura 6.9 - Bloco de referenciamento de um eixo em redes de Petri.

O referenciamento é ativado pelo lugar global p15, correspondente ao flag F1.4. O eixo pode estar em qualquer posição, ao longo do indicado na Figura 6.9. Caso o eixo não estiver sobre nenhum dos sensores de fim de curso, representados pelos lugares p71, p72 e p73, que são os sinais negados de I1.2, I1.4 e I1.3, o eixo é enviado no sentido do sensor do eixo negativo “-” pela transição t57, em busca do sensor de

referência. Caso encontre o eixo negativo “-“, representado pelo lugar p70 é automaticamente enviado para o sentido positivo “+“, t58, e ao atingir o sensor de referência, representado pelo lugar p74, é ativada uma rotina, através do flag F6.3 (lugar p75), de busca da marca do encoder do sistema de medição para o posicionamento do zero absoluto da máquina. Fazendo com que o eixo saia da condição de não referenciado (p76) para referenciado (p77).

Caso o eixo esteja sobre o sensor do fim de curso positivo “+“, ele é enviado para o sentido negativo, através de t56, se estiver sobre o eixo negativo “-“, é enviado para o sentido positivo, através de p70. Ao encontrar o sensor do fim de curso de referencia representado pelo lugar p74, realiza o procedimento de busca da marca como descrito anteriormente.

A partir deste bloco, podem ser modelados quantos eixos forem necessários para a máquina CNC.

6.1.7 - Intertravamentos de acionamentos em geral

A lógica do intertravamento de acionamento/desacionamento possui uma estrutura básica onde toda ação a ser executada pelo CLP deve ser confirmada, isto é, um contato auxiliar do relê de saída a ser atuado é utilizado como uma entrada de confirmação.

A Figura 6.10 ilustra o modelo básico do intertravamento de acionamento/desacionamento utilizando-se do sistema de refrigeração como exemplo, o qual pode ser comandado tanto no modo manual como no modo automático. Outras ações podem ser comandadas por modelos semelhantes. De acordo com a característica de cada uma, as ações poderão ter intertravamentos diferentes de acionamento e até mesmo de desligamento.

Na Figura 6.10 os lugares p78, p79 e p80 determinam o acionamento da refrigeração, utilizando as entradas I1.5, I2.1 e a função M08. A entrada de verificação I2.2 é utilizada para verificar se o relê da saída atuou, sinalizando o flag F10.1, representado pelo lugar global p100, se houve falha no acionamento da refrigeração ou o flag F10.2, representado pelo lugar global p101, indicando que o comando foi executado com sucesso.

Os lugares p82, p83 e p3, realizam o desligamento da refrigeração através do botão de desliga a refrigeração, de função miscelânea de desligamento da refrigeração e do desligamento do sistema, utilizando a entrada I1.6, a função M09 e o flag F0.1 respectivamente.

No desligamento da refrigeração, a entrada de verificação é utilizada de forma complementar em relação ao ligamento do sistema, sinalizando o flag F10.3, representado pelo lugar p97. Para a condição de falha do desligamento tem-se o Flag 10.5 que aciona a rotina de emergência devido a essa falha e o flag F10.4 para o desligamento com sucesso, permitindo a atuação do Flag F6.7 de refrigeração desligada.

Deste modo os lugares p84 e p85, que representam os flags F7.0 e F6.7, determinam se o estado da refrigeração é ligado ou desligado, respectivamente.

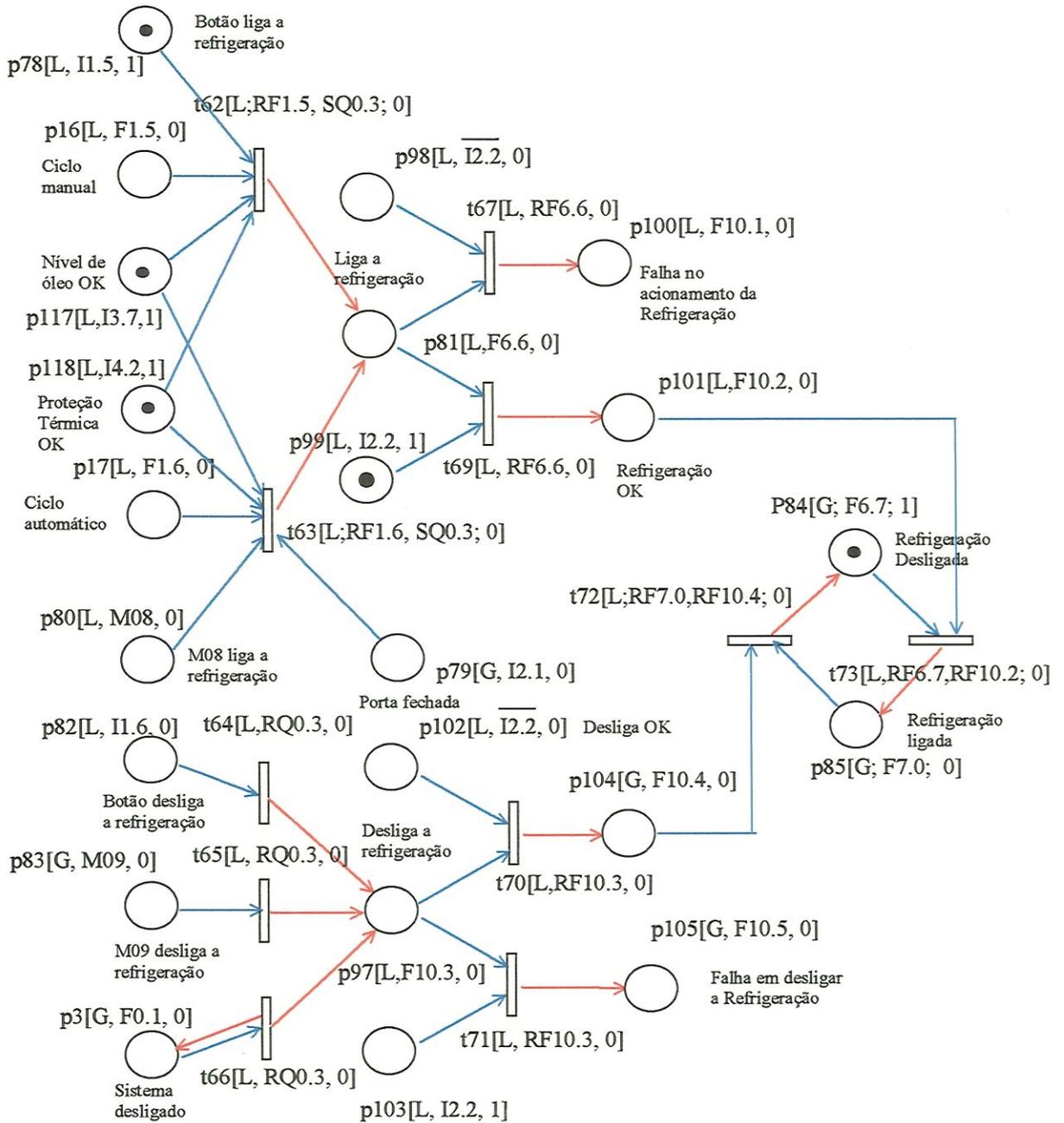


Figura 6.10 - Modelagem do acionamento da refrigeração.

6.1.8 – Tabelas de identificação dos elementos

Tabela 6.1 - Sinais internos

Flags internos		Flags internos	
F0.0	Emergência	F5.0	Ativa referenciamento
F0.1	Sistema desligado	F5.1	Desativa outros modos
F0.2	Flag de 1ro scan	F5.2	Aborta programa NC
F0.3	Sistema ligado	F5.3	Rotina de posicionamento de eixos
F0.4	Liberar acionamentos	F5.4	Desliga mancal
F0.5	Ativar hidráulico	F5.5	Transmitindo
F0.6	Ativar lubrificação	F5.6	Falha na transmissão
F0.7	Posição comunicação NC	F5.7	Inicia recepção
F1.0	Posição referenciamento	F6.0	Recebendo
F1.1	Posição manual	F6.1	Recepção confirmada
F1.2	Posição automático	F6.2	Falha na recepção
F1.3	Ciclo de comunicação NC	F6.3	Procurando a marca
F1.4	Ciclo de referenciamento	F6.4	Eixo não referenciado
F1.5	Ciclo manual	F6.5	Eixo referenciado
F1.6	Ciclo automático	F6.6	Refrigeração ativada
F1.7	Inicialização OK	F6.7	Refrigeração desligada
F2.0	Acionamentos ativados	F7.0	Refrigeração ligada
F2.1	Hidráulico ativado	F7.1	Desativa automático
F2.2	Falha nos acionamentos	F7.2	Desativa manual
F2.3	Acionamentos OK	F7.3	Desativa comunicação
F2.4	Lubrificação OK	F7.4	Desativa referenciamento
F2.5	Hidráulico OK	F7.5	Eixo árvore desligado
F2.6	Falha no hidráulico	F7.6	Eixos posicionados
F2.7	Lubrificação ativada	F7.7	Desliga lubrificação do mancal
F3.0	Lubrificação (motor ligado)	F10.0	Mensagem de falha no hidráulico
F3.1	Ciclo de lubrificação completo	F10.1	Falha acionamento da refrigeração
F3.2	Fim de um ciclo	F10.2	Refrigeração ligada com sucesso
F3.3	Flag de set do contador	F10.3	Desliga a refrigeração
F3.4	Flag de reset do contador	F10.4	Refrigeração desligada com sucesso
F3.5	Flag de contagem atingida	F10.5	Falha em desligar a refrigeração
F3.6	Falha na lubrificação	F10.6	Inicializa trans. /recepção dados
F3.7	Habilita lubrificação manual	F10.7	Acionamentos desligados
F4.0	Habilita lubrificação automática	F11.0	Acionamentos ligados
F4.1	Transmite/recebe status	F11.1	Hidráulico desligado
F4.2	Ativa automático	F11.2	Hidráulico ligado
F4.3	Desativa outros modos	F11.3	Envia eixo X para p sentido “-“
F4.4	Ativa manual	F11.4	Envia eixo X para o sentido “+”
F4.5	Desativa outros modos	F11.5	
F4.6	Ativa comunicação	F11.6	
F4.7	Desativa outros modos	F11.7	

Tabela 6.2 – Lista de *transições*

<i>transição</i>		<i>transição</i>	
t1	Desliga sistema	t43	Modo de comunicação ativado
t2	Desliga sistema	t44	Ativa modo de referenciamento
t3	Liga sistema	t45	Desativa modo de referenciamento
t4	Ativa rotinas de inicialização	t46	Desativa modo de referenciamento
t5	Ativa ciclo de comunicação CN	t47	Desativa modo de referenciamento
t6	Ativa ciclo de referenciamento	t48	Modo de referenciamento ativado
t7	Ativa ciclo manual	t49	Falha no hidráulico
t8	Ativa ciclo automático	t50	Transmissão
t9	Libera acionamentos	t51	Falha na transmissão
t10	Falha nos acionamentos	t52	Transmissão OK
t11	Acionamentos OK	t53	Recepção
t12	Liga hidráulico	t54	Recepção OK
t13	Hidráulico OK	t55	Falha na recepção
t14	Falha no hidráulico	t56	Envia o eixo para “-“
t15	Inicialização OK	t57	Envia o eixo para “-“
t16	Aciona motor da lubrificação	t58	Envia o eixo para “+“
t17	Falha na lubrificação	t59	Procura a marca
t18	Lubrificação ativada	t60	Referenciamento OK
t19	Desliga lubrificação	t61	Pedido de referenciamento
t20	Reset do contador C1	t62	Ativa refrigeração em manual
t21	Set do contador C1	t63	Ativa refrigeração em automático
t22	Contagem atingida em C1	t64	Desliga refrigeração em manual
t23	Contagem crescente em C1	t65	Desliga refrigeração em automático
t24	Restart do ciclo	t66	Sistema desligado
t25	Ativa ciclo em manual	t67	Falha acionamento da refrigeração
t26	Termina ciclo de lubrificação	t68	Ativa emergência
t27	Ativa ciclo em automático	t69	Refrigeração ligada com sucesso
t28	Ativa comunicação status	t70	Refrigeração desligada com sucesso
t29	Ativa modo automático	t71	Falha desligamento da refrigeração
t30	Desativa modo automático	t72	Desliga refrigeração
t31	Desativa modo automático	t73	Liga refrigeração
t32	Desativa modo automático	t74	Tempo T1 atingido
t33	Modo automático ativado	t75	Tempo T2 atingido
t34	Ativa modo manual	t76	Tempo T3 atingido
t35	Desativa modo manual	t77	Falha na pressão do hidráulico
t36	Desativa modo manual	t78	Disj. Térmico do hidráulico
t37	Desativa modo manual	t79	Acionamentos ligados
t38	Modo manual ativado	t80	Acionamentos desligados
t39	Ativa modo de comunicação	t81	Hidráulico desligado
t41	Desativa modo de comunicação	t82	Hidráulico ligado
t42	Desativa modo de comunicação	t83	Falha falta óleo lubrificação
t40	Desativa modo comunicação	t84	Disj. Térmico da lubrificação

Tabela 6.3 – Lista de *lugares*

<i>Lugar</i>		<i>Lugar</i>	
p1	Emergência	p61	Transmitindo
p2	Botão de emergência	p62	Falha na transmissão
p3	Sistema desligado	p63	Transmissão OK
p4	Botão liga	p64	Pedido de recepção
p5	Ligar acionamentos	p65	Recebendo
p6	Ativar hidráulico	p66	Sem confir. de recepção
p7	Ativar lubrificação das guias	p67	Falha na recepção
p8	Sistema ligado	p68	Recepção OK
p9	Flag de 1ro scan	p69	Entrada de sinal X+
p10	Posição de comunicação NC	p70	Entrada de sinal X-
p11	Posição de referenciamento	p71	Entrada de sinal negado X+
p12	Posição manual	p72	Entrada de sinal negado X-
p13	Posição automático	p73	Entrada de sinal negado Xref
p14	Ciclo de comunicação NC	p74	Entrada de sinal Xref
p15	Ciclo de referenciamento	p75	Procurando marca
p16	Ciclo manual	p76	Eixo não referenciado
p17	Ciclo automático	p77	Eixo referenciado
p18	Sem confir. dos acionamentos	p78	Botão liga refrigeração
p19	Acionamentos ativados	p79	Sensor porta fechada
p20	Confirmação acionamentos OK	p80	Sinal M08 CNC
p21	Confirmação hidráulico OK	p81	Refrigeração ativada
p22	Hidráulico ativado	p82	Botão desliga refrigeração
p23	Sem confirmação do hidráulico	p83	Sinal M09 CNC
p24	Falha nos acionamentos	p84	Refrigeração desligada
p25	Acionamentos OK	p85	Refrigeração ligada
p26	Lubrificação OK	p86	Inicializa trans./recep. dados
p27	Hidráulico OK	p87	Falha desligamento da refrigeração
p28	Falha no hidráulico	p88	Aborta programa NC
p29	Sem confirmação da lubrificação	p89	Desliga eixo árvore
p30	Lubrificação ativada	p90	Posiciona eixos
p31	Confirmação lubrificação OK	p91	Desliga a refrigeração
p32	Falha na lubrificação	p92	Desliga o mancal
p33	Lubrificação acionada	p93	Eixo árvore desligado
p34	Contagem crescente de C1	p94	Eixos posicionados
p35	Set do contador C1	p95	Lubrificação do mancal desligada
p36	Reset do contador C1	p96	Envia mensagem de falha
p37	Contagem atingida em C1	p97	Desliga a refrigeração
p38	Contador C1 negado	p98	Falha na refrigeração
p39	Habilita lubrificação manual	p99	Refrigeração ligada com sucesso
p40	Habilita lubrificação automática	p100	Falha acionamento da refrigeração
p41	Reestart do ciclo de lubrificação	p101	Refrigeração deslig. com sucesso

p42	Botão na posição automática	p102	Desligar refrigeração com sucesso
p43	Ativa modo automático	p103	Sem conf. desligamento refrig.
p44	Desativa outros modos	p104	Refrigeração desligada
p45	Automático desativado	p105	Falha na pressão do hidráulico
p46	Botão posição manual	p106	Proteção térmica do hidráulico acionada
p47	Ativa manual	p107	Pressão do hidráulico OK
p48	Desativa outros modos	p108	Disj. Térmico do hidráulico OK
p49	Manual desativado	p109	Acionamentos desligados
p50	Programa CN ativo	p110	Acionamentos ligados
p51	Botão posição comunicação	p111	Hidráulico desligado
p52	Ativa comunicação NC	p112	Hidráulico ligado
p53	Desativa outros modos	p113	Falta óleo para a lubrificação
p54	Comunicação desativada	p114	Proteção Térmica da lubrificação acionado
p55	Botão pos. referenciamento	p115	Nível de óleo da lubrificação OK
p56	Ativa referenciamento	p116	Disj. Térmico da lubrificação OK
p57	Desativa outros modos	p117	Nível de óleo da refrigeração OK
p58	Referenciamento desativado	p118	Disj. Térmico da refrigeração OK
p59	Entrada de transmissão	p119	
p60	Sem confir. de transmissão	p120	

Tabela 6.4 - Sinais externos de entrada

Sinais de entrada		Sinais de entrada	
I 0.0	Botão de emergência	I 2.4	Fim de curso Zref
I 0.1	Botão liga	I 2.5	Fim de curso Z-
I 0.2	Acionamentos ligados	I 2.6	Botão liga transportador de cavacos
I 0.3	Hidráulico ligado	I 2.7	Botão desliga transportador de cavacos
I 0.4	Modo automático	I 3.0	Confirmação transportador cavacos ligado
I 0.5	Modo manual	I 3.1	Botão prende peça
I 0.6	Modo de comunicação	I 3.2	Botão solta peça
I 0.7	Modo de referenciamento	I 3.3	Confirmação peça presa
I 1.0	Lubrificação acionada	I 3.4	Pressostato de alimentação ar comprimido
I 1.1	Botão aciona lubrificação	I 3.5	Pressostato do hidráulico
I 1.2	Fim de curso X+	I 3.6	Nível do reservatório da lubrificação
I 1.3	Fim de curso Xref	I 3.7	Nível do reservatório da refrigeração
I 1.4	Fim de curso X-	I 4.0	Disj. Térmico do transportador de cavacos
I 1.5	Botão liga a refrigeração	I 4.1	Disj. Térmico da lubrificação
I 1.6	Botão desliga a refrigeração	I 4.2	Disj. Térmico da refrigeração
I 1.7	Pedido de transmissão	I 4.3	Disj. Térmico do hidráulico
I 2.0	Transmissão ACK	I 4.4	
I 2.1	Porta fechada	I 4.5	
I 2.2	Confirmação refrigeração ligada	I 4.6	
I 2.3	Fim de curso Z+	I 4.7	

Tabela 6.5 - Sinais externos de saída

Sinais de saída	
Q 0.0	Acionamentos
Q 0.1	Hidráulico M5
Q 0.2	Lubrificação M8
Q 0.3	Refrigeração M6
Q 0.4	Mancal
Q 0.5	Pedido de recepção
Q 0.6	Recepção ACK
Q 0.7	Transportador de cavacos M9

As Tabelas 6.1, 6.2, 6.3, 6.4 e 6.5 identificam os sinais internos (flags), a lista de *transições*, a lista de *lugares*, os sinais externos de entrada e os sinais externos de saída, respectivamente, que foram utilizados para a modelagem dos blocos de funções do Torno CNC.

6.2 – Simulação do modelo do Torno CNC

A simulação consiste em verificar, as condições de travamento (“*deadlock*”), alcançabilidade, limitabilidade e reversibilidade do modelo durante o seu ciclo de funcionamento. Essas verificações são realizadas pelo projetista de acordo com necessidades do sistema proposto.

O projetista determina como a seqüência deve prosseguir, de acordo com o funcionamento da máquina ou sistema, quando é encontrada, por exemplo, uma transição “*dead*” que dependa de entradas externas e pode determinar se os conflitos são condições geradas pelo seqüenciamento das atividades ou por falha de modelagem.

Assim no projeto de intertravamentos de uma máquina, no caso o Torno CNC, descreve um caminho cíclico a partir de um ponto determinado, isto é, após o intertravamento de partida do sistema.

São determinados todos os ciclos possíveis de serem realizados na máquina ou sistema, utilizando as características funcionais do editor de rede de Petri, e alguns ciclos principais de funcionamento da máquina se destacam, tais como o ciclo: de funcionamento em automático, de funcionamento em manual, de referenciamento, etc., os quais são simulados. Também são analisados os ciclos de emergência e de recuperação de falhas.

As marcações iniciais e finais a serem utilizadas para as análises das características das redes de Petri são determinadas pelo analista. A marcação final é a marcação relativa a um ciclo completo do sistema. A marcação inicial corresponde as condições iniciais da máquina ou sistema.

Esta análise dos intertravamentos utilizando as características das redes de Petri, mostra quais os flags devem ser ativados ou desativados, se a seqüência de operação da máquina ou sistema está correta ou não e se existem possíveis ambigüidades no projeto dos intertravamentos, isto é, se há repetição desnecessária de algum comando.

Na simulação do projeto de intertravamento do Torno CNC, foram contempladas as simulação de sua seqüência de funcionamento principal e suas variantes em termos de falha de energia e falta de sinal na seqüência, e o comportamento do modelo de intertravamento foi satisfatório. Permitindo assim o passo seguinte de geração do programa de intertravamento em uma linguagem CLP.

6.3 – Resultados da Implantação e testes do programa CLP gerado

A tradução da programação para o programa CLP foi executada e implementada em um Torno CNC marca GE FANUC, montado no Laboratório de Simulação de Sistemas Discretos da Escola de Engenharia de São Carlos (EESC) da Universidade de São Paulo (USP) (descrita no Anexo B).

O seu funcionamento mostrou-se eficiente e as condições de funcionamento e falhas simuladas anteriormente foram verificadas na montagem real; outras situações também foram analisadas e o intertravamento reagiu adequadamente.

Algumas mudanças foram registradas e estas alteradas no editor e novamente o programa gerado mostrou-se satisfatório para a aplicação prática.

Assim, o projeto gerado para o Torno CNC sofreu alterações e o ambiente de intertravamento foi utilizado para a atualização da documentação do modelo e do programa CLP, além de se verificar através da simulação as alterações realizadas. Desta forma o ambiente mostrou-se uma ferramenta efetiva para a manutenção do sistema, quanto as alterações no projeto que podem vir a serem realizadas.

7 – CONCLUSÕES

O projeto de intertravamentos de sistemas de manufatura tem mostrado um aumento de sua complexidade devido as novas exigências do mercado consumidor e do desenvolvimento tecnológico crescente.

Assim, há uma forte demanda em se buscar formas de se facilitar, sistematizar e até mesmo tornar automático o projeto de tais sistemas, como foi constatada na revisão bibliográfica do capítulo 2.

Esse trabalho buscou reunir tecnologias da área de inteligência artificial, modelagem gráfica com consistência matemática, simulação e análise de desempenho, para a geração de um programa de intertravamentos, de um sistema de manufatura, em uma linguagem de controlador programável.

Para isso, foi proposta a especificação de uma sistemática para o projeto de intertravamentos e criado um ambiente para executar esta sistematização, que mostrou-se eficaz nas etapas de projeto e implementação dos sistemas de manufatura discreta, como mostrado no capítulo 3, e na validação da implementação desse ambiente para o projeto de intertravamento de máquinas no capítulo 6.

Especificamente pode-se fazer a seguinte análise a respeito do ambiente de sistematização do projeto de intertravamento executado:

1. A utilização do sistema especialista para a análise e definição das necessidades, que envolve as viabilidades, decisões do projeto e a elaboração de uma descrição funcional que integra as áreas e elementos envolvidos, facilitou a construção do modelo a ser projetado, evitando que comportamentos diferentes do previsto fossem “esquecidos” ou inconscientemente descartados e permitiu uma estruturação sistemática do modelo, favorecendo a sua reutilização e manutenção.
2. O uso da rede de Petri como ferramenta de modelagem foi adequada principalmente pela capacidade de integrar em seus níveis de abstração todas as áreas e elementos envolvidos no projeto. Isto permitiu o entendimento do controle (intertravamento) com a parte funcional da máquina (pneumática, hidráulica, etc.).
3. Em um nível de abstração adequado a rede de Petri, representou os elementos de programação CLP, tais como contadores, temporizadores, etc., em uma biblioteca de modelos que permitiram a padronização de funções e vieram a facilitar a tradução do modelo para a linguagem CLP dos sistemas de intertravamento.

4. A partir da biblioteca de intertravamentos, o ambiente pode disponibilizar os blocos de funções e facilitar a reutilização de modelos.
5. A simulação utilizando o modelo em rede de Petri do projeto de intertravamento, mostrou-se eficaz para a análise das seqüências do sistema e das alterações que pudessem ocorrer.
6. A análise de desempenho do sistema também mostrou-se útil, para o seqüenciamento correto das atividades e a detecção de assincronismos de funções e seqüências de funcionamento.
7. O programa CLP gerado foi implementado com sucesso e se mostrou eficiente para a implementação nos vários tipos de sistemas (foram testados em um CLP da Klockner Müller, e um CLP GE FANUC), as alterações da implementação se mostraram poucas e foram introduzidas no ambiente através do editor do modelo o que fez com que a documentação da programação CLP fosse atualizada automaticamente.
8. O sistema especialista de manutenção foi, para determinadas falhas (interrupção da seqüência e falha elétrica), eficiente no seu diagnóstico, podendo este, do mesmo modo que o de análise e definição das necessidades ser refinado com o aumento da sua base de conhecimento.

Pôde-se mostrar que as dificuldades que originaram este trabalho foram solucionadas pelo ambiente de projeto de intertravamento, onde:

A dificuldade de tratar uma grande variação de arquiteturas de sistemas de produção (linhas de transferência, células de manufatura, etc) e a falta de especialistas para integrar as várias áreas envolvidas, com o objetivo de especificar uma descrição funcional para o projeto do intertravamento de uma forma adequada, foram solucionadas com o auxílio de um **Sistema especialista de análise e definição das necessidades**.

A falta de ferramentas que possibilitam a obtenção de um único modelo completo; as dificuldade de retratar hierarquia e modularidade no modelo; e a dificuldade de reaproveitamento de módulos, foram tratadas utilizando um **Editor de redes de Petri**.

As dificuldade na análise e validação do modelo foram contornadas com um **Simulador de redes de Petri**.

Um **Tradutor de redes de Petri** do modelo para a linguagem CLP padrão diminuiu a dificuldade de reprogramações rápidas nos projetos de intertravamento.

E a dificuldade de diagnóstico de falhas teve por proposta ser melhorado com a utilização de um **Sistema especialista de manutenção**.

Outras dificuldades, tais como a de padronização da documentação; de geração automática de documentação; de manutenção e alterações; de análise das condições de

partida, foram minimizadas com o ambiente, com a utilização das fases de **documentação e de implementação**.

Finalmente, pode-se afirmar que a sistematização do projeto de intertravamentos para sistemas de manufatura pode ser realizada com o uso desse ambiente proposto, fornecendo condições para se desenvolver, implementar e manter estes sistemas, de uma forma padronizada e consistente, minimizando ou eliminando as dificuldades conforme descrito.

7.1 – Sugestões para trabalhos futuros

Algumas sugestões para trabalhos futuros podem contemplar:

A utilização do ambiente para a facilitar e padronização do controle do projeto e manutenção incorporando tecnologias com as de integração do chão de fábrica (CIM), fábricas virtuais e planejamento fino da produção.

A adaptação da sistemática aos novos conceitos de CNC e CLP de arquitetura aberta, os quais vem se tornando uma tendência. Através de estudos a respeito de padronização e definições referentes a estes sistemas.

A utilização do ambiente a nível de controle supervisão, com sua integração com equipamentos do tipo SDCD (Sistema Digital de Controle Distribuído), utilizado em sistemas de controle híbridos e contínuos.

A utilização de outras classes ou extensões de redes de Petri, tais como redes de Petri Coloridas, redes de Petri Híbridas, etc., para a modelagem e simulação. Ou o uso de técnicas diferentes de redes de Petri (Statecharts, Mark Flow Graph, etc.) para a modelagem e a simulação.

A introdução e adaptação de outros módulos, tais como o de técnicas de prevenção de acidentes, metodologias de implantação, etc., dependendo da necessidade do sistema.

CAPÍTULO 8 - BIBLIOGRAFIA

- ABRAHAM, C. T.; SCIOMACHEN, A.; *Planning for Automatic Guided Vehicle Systems by Petri nets* Tech. Rep. RC12288, IBM, Yorktown Heights, NY, 1986.
- AGUIAR, M. V. C. *Implementação de um servidor MMS para Controladores Lógicos Programáveis*. Paraíba, Junho de 1992, Dissertação de mestrado, Universidade Federal da Paraíba.
- AMMAR, H. H.; YU, L. *Fuzzy marking petri nets: concepts and definition* IEEE International Symposium on Intelligent Control - Proceedings 1995. p 291-297
- ARCHETTI, F.; FAGIUOLI, E.; SCIOMACHEN, A.; *Computation of the makespan in a transfer line with station breakdowns using stochastic Petri nets* Computers and Operations Research, v 14, pp 273-278, 1987.
- ARJONA, S. E.; LOPEZ, M. E.; *Synthesis of coloured Petri nets for FMS task specification*, Inter. Journal of Robotics and Automation, v11 n3 1996, p111-117.
- BAKER, H.A.; GRANT, P.W.; SONG, J., *Graphical simulator for programmable logic controllers based on Petri nets* Microprocessing and Microprogramming v 41 n 10 Jun 1996. p 737-756.
- BANAZAK Z. A.; KROGH, B. H.; *Deadlock avoidance in Flexible Manufacturing Systems with concurrently competing process flow*. IEEE Transactions on Robotics and Automation, v 6, n6, p 724-734, 1990.
- BAROZZI, S.; FERRARA, A. B.; CAMURRI, A.; FRANCHI, P.; MASTROIANNI, C.; MORELLI, C.; SAGOLEO, R.; SALOMONE, V., *Petri net-based real time simulation of industrial plants*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. IEEE, Piscataway, NJ, USA, v 2 1994, p 1983-1989.
- BATES, I. D.; CHESTER, E. G.; KINNIMENT, D. J.; *Case study in the automatic programming of a PLC based control system using statemate statecharts*. IEE Conference Publication n455-1, 1998, p832-837.
- BEL, G.; CAVAILLE, J. B. DUBOIS D.; *Flexible manufacturing systems: modelling and Simulation*. Systems and Control: Encycolpedia, Theory, Technology, Application, p - 1642-1649, 1988.
- BELHOT, R. V.; *Conceito e Desenvolvimento de Sistemas Especialistas*, Publicação interna 063/93 - Escola de Engenharia de São Carlos - USP.
- BEN-ARIED, D.; MOODIE, C. L. e CHU, C. C.; *Knowledge-based scheduling under unpredicted condicions: two approaches*, International Journal of Production Research, vol 27, n 5, pp 869-882, 1989.
- BIEMANS F. P. ; VISSERS, C. A.; *Reference model for manufacturing planning and control systems*. Journal of Manufacturing Systems, v 8 n 1 p 35-46, January 1989.
- BOBEANU, C.; FILIP, F., *ModCPN - an integrated set of tools for Petri nets-based modelling of manufacturing systems*, IEEE Symposium on Emerging Technologies & Factory Automation v 1 1995. IEEE, Piscataway, NJ, USA. p 95-103.
- BONFATTI, F.; GADDA, G.; MONARI, P. D.; *Improved process for the development of PLC software*. Proceedings of International Conference on Software Engineering, IEEE - Los Alamitos, 1997 p 400-410.
- BOOCH, G.; *Object oriented analysis and desig*. Benjamin/Cummings, 1994, sec. edition
- BROOKS, H. M.; *Expert systems and intelligent information retrieval*. Information & Processing Management, v23, n4, 1987, pp 367-382

- BRUNO, G.; BIGLIA, P.;** *Performace evaluation and validation of tool handling in flexible manufacturing systems using Petri nets.* publ. by IEEE, p 64-71, 1985.
- CAMPBELL R. L.;** *An architecture for factory control automation.* AT&T Technical Journal, v 66, n 5, p 77-85, Sept./Oct. 1987.
- CAMURRI, A., MEGAGNINI, M.;** *Petri net-object oriented architecture for plant simulation.* IEEE Inter. Conference on Systems, Man and Cybernetics, v2, 1998, p 1061-1065.
- CARADEC, M.; PRUNET, F.;** *New modelling tool for hybrid flexible systems.* Proc of IEEE International Conf. on Systems, Man and Cybernetics, v3, 1997, p2246-2251.
- CAVALIERI, S.; MIRABELLA, O.; ZINGARINO, G.;** *Petri-net based approach for FMS performance evaluation.* IECON – Proc. Industrial Eletronics conference, v3, 1997, p1204-1209.
- CECIL, J. A.; SRIHARI, K.; EMERSON ,C. R. ;** *A review of Petri-net applications in manufacturing.* Advanced Manufacturing Technology. v 7, p 168-177, 1992
- CESKA, M.; JANOUSEK, V.; VOJNAR, T;** *Object-oriented Petri nets, their simulation, and analysis.* IEEE International Conf. on Systems, Man and Cybernetics, v1, 1998, p 262-272.
- CHAMAS, N.; ANNEBERG, L.; YAPRAK, E.,** *Timed neural Petri nets,* Midwest Symposium on Circuits and Systems v 2 1993. Publ by IEEE Service Center, Piscataway, NJ, USA. p 926-929.
- CHANG, N.; PARK, J.; KOO, K.; KWON, W. H.,** *Memory-based implementation of a Petri net and its application to a programmable controller,* Plenary Session, Emerging Technologies, and Factory Automation IECON Proceedings (Industrial Electronics Conference) v 1 1993. Publ by IEEE, Computer Society Press, Los Alamitos, CA, USA, p 613-618.
- CHAO, D. Y.; WANG, D.T.,** *XPN-FMS: a CAD tool for FMS modeling, analysis, animation, and simulation using Petri nets and X window,* International Journal of Flexible Manufacturing Systems v 7 n 4 Oct 1995. p 339-360.
- CHEN, Y.; WONGLADKOWN, C.,** *A real-time control simulator design for automated manufacturing systems using Petri nets.* Proceedings - IEEE International Conference on Robotics and Automation v 3. p 2542-2547.
- CHINCHOLKAR, A. K.; KRISHNAIAH, C. O. V.;** *Simulataneous optimisation of control factors in automated storage and retrieval systems and FMS using stochastic coloured Petri Nets and the Taguchi method.* Intern. Journal of advanced Manufac. technology, v2, n2, 1996, p137-144.
- CHRISTENSEN, S.; MORTENSEN, K. H.;** *Petri net tools database, 1997,* www.daimi.aau.dk/PetriNets.
- CHO, H.;** *Graph-Thoretic Deadlock Detection and Resolution for Flexible Manufacturing Systems -* IEEE Transactions on Robotics and Automation, v11, n3, p413-421, 1995.
- CHOI, B. W.; KUO, W.; JACKMAN, J. K. ;** *Petri Net extensions for modelling and validating manufacturing systems.* Inter. Journal of Production Research, v 32, n 8, 1994,p 1819-1835.
- COAD, P.; YOURDON, E.** *Análise Baseada em Objetos.* Editora Campus, 2.ed. Rio de Janeiro, pp 225, 1992.
- COHEN, G.; DUBOIS, D; QUADRAT, J; VIOT, M.;** *A linear-system-theoretic view of discrete event processes and its use for performance evaluation in manufacturing,* IEEE Transactions on Automatic Control v AC-30, pp 210-220, 1985.

- COSSINS, R. ; FERREIRA, P. : *Celeritas : a coloured Petri net approach to simulation and control of flexible manufacturing systems*. International Journal of Production Research v 30, n 8, p 1925-1956, Aug 1992.
- COURVOISIER, M.; VALETE, R.; BIGOU, J. M.; ESTEBAN, P.; *A programmable logic controller based on a high level specification tools*. Proc. International Conf..Industrial Eletronics and Instrumentation (IECON), 1983, p174-179.
- CRANMER, D. : *Implementing a PLC system* Pace - Process and Control Engineering, v 44 n 10 . oct. 1991 pp. 68-69,.
- DAVID, R.; *GRAF CET: A Powerful tool for Specification of Logic Controllers*, IEEE Transactions on Control Systems Technology, v3, n3, sept. 1995, pp 253-268.
- DESROCHERS, A. A.; AL-JAAR, R, Y.; *Applicaitions of Petri Nets in Manufacturing Systems- Modeling, Control, and Performance Analysis*. IEEE – Computer Society Press New York, 1995, pp 346.
- DESCHANEL, F.; *Pilotage D'Une Cellule Flexible D'Usinage*, France, 1989, Tese de doutorado - Universite de Franche- Comte.
- DICESARE, F.; GOLDBOGEN, G.; FEICHT, D.; LEE, D. Y.; *Extending Error Recovery Capability in Manufacturing by Machine Reasoning*, Transactions on Systems, Man and Cybernetics, v23, n1, jan/feb 1993, pp 221-228.
- DICESARE, F.; GILE, M. R.; KULP, P, T.; *Code generation tool for simulation and control, based on coloured Petri nets*. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. v4, 1994, Piscataway, NJ, USA, pp. 3063-3068.
- DITCH, W.; *PLC Emulation using PIC Microcontrollers*, Special Autums supplement published by Elektor Eletronics, pp 31, England, 1995.
- DOMBROWSKI, L. R. ; JHONSON, W. H. ; MCCALL, R. : *Separation of interlocking and regulatory control in programmable eletronic systems* ISA Trasactions USA. v 30, n 4, p 19-25, 1990.
- D'SOUZA, K. A.; *Control model for detecting deadlocks in an Automated Maching Cell*, Computers & Industrial Engineering v 26 n 1 Jan 1994. p 133-139.
- D'SOUZA, K. A.; KATHOR, S. K.; *A survey of Petri net applications in modeling controls for Automated Manufacturing systems*, Computers in Industry v 24 1994. p 6-16.
- DUGGAN, J.; BROWNE, J.: *ESPNET - Expert-System-based simulator of Petri Nets*; IEE proceedings Control theory and Applications, v135, n4, 1988, p239-247.
- DURAN, O.; BATOCHIO A.: *Automatic PLC software generator with a natural interface*; International Journal of Production Research, v37, n4, mar 10, 1999, p805-819.
- ESSER, R.; *Object oriented Petri Net language for embedded system design*, IEEE – Proc. of the International Workshop on Software Technology and Engineering Practice, 1997, p216-223.
- EZPELETA, J., *A Petri Net based deadlock prevention policy for Flexible Manufacturing Systems*, IEEE Transactions on Robotics and Automation, v11, n2, p173-184, 1995.
- EZPELETA, J.; COLOM J. M.; *Automatic Synthesis of Colored Petri Net for the control of FMS*, IEEE Transactions on Robotics and Automation, v13, n3, p327-337, 1997.
- FELDBRUGGE, F.: *Petri net tools overview, 1989*, Proceedings Annual European Workshop: Applications and Theory of Petri Nets'88, n424, in LNCS. Springer, 1990
- FELDBRUGGE, F.: *Petri net tools overview, 1992*, Proceedings Annual European Workshop: Applications and Theory of Petri Nets'91, n674, in LNCS. Springer, 1993

- FELDBRUGGE, F.; JENSEN, K.: *Petri net tools overview, 1986*, In W. Brauer, W. Reisig and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, n255 in LNCS. Springer, 1987
- FLAUS, J. M.: *Hybrid flow nets for hybrid process modelling and control*, IEE Colloquium n256 dec 6 1996, p6/1-6/31
- FLEISCHHACK, H.; LICHTBLAU, U., *MOBY - a tool for high level petri nets with objects*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics v4 1993. p 644-649
- FLORIN, G; NATKIN, S., *Evaluation based upon stochastic Petri nets of the maximum throughput of a full duplex protocol.*, C. Girault and W. Reisig ed. *Applications and Theory of Petri nets*. New York, Springer- Verlag, pp 280-288, 1982.
- FREEMAN, R. A.: *What should you worry about when doing a risk assesment ?* Chemical Engineering Progress p 29-34, november 1989.
- FREY, G.: *Simulation of hybrid systems based on interpreted Petri nets* IEE Conference Publication n457 1998, p168-175.
- GAYMAN, D. J., *An old favorite gets new standards*, Manufacturing Engineering, january1988. pp 55-58.
- GENRICH, H. J.: *Predicate/Transition Nets. Petri Nets: Central Models and Their Properties*, Advances in Petri Nets 1986. Lect. Notes in Comp. Sci. v254, 1986, pp 207-247.
- GERMAN, R.; KELLING, C.; ZIMMERMANN, A.; HOMMEL, G.; *TimeNET - a toolkit for evaluating non-Markovian stochastic petri nets*, International Workshop on Petri Nets and Performance Models 1995.IEEE, Los Alamitos, CA, USA. p 210-211.
- GHABRI, M. K.; LADET, P., *Controlled continuous Petri nets*, Proceedings - IEEE International Conference on Robotics and Automation pt 1 1994. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. p 788-793.
- GIUA, A.; DICESARE, F., *Blocking and controllability of Petri nets in supervisory control*, IEEE Transactions on Automatic Control v 39 n 4 Apr 1994. p 818-823.
- GRACANIN, D.; SRINIVASAN, P.; *Software environment for simulation and real-time coordination of manufacturing systems*, Midwest Symp. on Circuits and Systems v 2 1994. p 1544-1547.
- GRACANIN, D.; SRINIVASAN, P.; VALAVANIS, K., *Time and error recovery with parametrized Petri nets*, Proc 1993 IEEE Int Symp. Intell. Control 1993. p 291-297.
- GRACANIN, D.; SRINIVASAN, P.; VALAVANIS, K.; *Parameterized Petri nets and their application to planning and coordination in intelligent systems*, IEEE Transactions on Systems, Man and Cybernetics v 24 n 10 Oct 1994. p 1483-1497.
- GROOVER, M. P., ZIMMERS Jr, E. W.; *CAD/CAM: Computer Aided Design and Manufacturing*; Prentice- Hall, Inc., USA, 1984.
- GRUHN, PAUL : *The pros and cons of qualitative and quantitative analysis of safety systems*. ISA Transactions ,USA, v 30, n 4, p 79-85, 1991.
- GUDWIN, R.; GOMIDE, F.: *Object Networks – a modelling tool*. IEEE world congress on computacional intelligence Conference on Fuzzy Systems, v1, 1998, p77-82.
- HANNA, M.; *Modelling productis quality from a CNC machining centre using fuzzy Petri nets with neural networks*. IEEE Symposium on Emerging Technologies and Factory Automation, v2, 1996 p 440-445.

- HATONO, I; YAMAGATA, K.; TAMURA, H.; *Modeling and on-line scheduling of Flexible Manufacturing Systems Using Stochastics Petri nets*, IEEE Transactions on Software Engineering, v 17, n2, Febr. 1991. pp 126-132.
- HASEGAWA, K.; *Modelling, Control and Deadlock avoidande of flexible manufacturing systems*, 11° CBA – Congresso Brasileiro de Automática, Anais SBA, São Paulo, SP, v1, p 37-51, 1996.
- HASEGAWA, K.; TAKAHASHI, K.; MIYAGI, P. E.; *Aplication of the Mark Flow Graph to Represent Discrete Event Production Systems and System Control*, Trans. of the Society of instrument and Control Engineers, v 24, n1, jan 1988 pp 69-75.
- HAYSLEY, D. L.; *Important factors involved in the selection of a safety interlock system*. Advances in Instrumentation, Proceedings v 47 , pt 1, p 51-59, October 1992.
- HESIEH, S; LIN, K. H. M.; *AGVS mechanism*. International Journal of Advanced manufaturing technology, n9, 1994, pp 71-78
- HENNING, D; *Synthesizing controllers from real-time specifications*. IEEE transactions on Computer Aided Design of Integrated Circuits and Systems, v18, n1, 1999, pp 33-43
- HERMANN G.; *The evolution of numerical control units in the light of integration*. Computers in Industry 17, p 341-347, 1991.
- HILLION, H. P.; PROTH, J-M.; *Performance evaluation of job shop systems using timed event-graphs* IEEE Transactions on Automatic Control, v 34, n1, Jan. pp 3-9, 1989.
- HOLLIDAY, M. A.; VERNON, M. K.; *A generalized timed Petri Net model for performance analysis*; IEEE transactions on Software Engineering, v13, n12, 1987, pp 1297-1310.
- HOLLIER, R. H.; *Automated guided vehicle systems international trends in manufacturing technology*, IFS (Publications) Ltd, 1987, pp361.
- HOLLO, F. R.; *Proposed Language Standards for Programmable Controllers*; Control Engineering, p. 260-264, September, 1985.
- HSIEH, S.; *Synthesis of AGVS by coloured – time Petri nets* International Journal of Computer Integreted Manuacturing v11, n4, jul-aug 1998, p 334-346.
- HUANG, J. T.; YANG, D. C. H.; *Precision command generation for computer controlled machines*. ASME - Production Engineering Division PED, v 58, pp 89-104, 1992.
- HWANG, C. P.; HO, C. S., *Hardware design of a real-time Petri net model for real-time tasks*, Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A/Chung-kuo Kung Ch'eng Hsuch K'an v 18 n 4 Jul 1995. p 481-492.
- ISAGRAPH; *Computer Aided Software Engineering Workbench for Open PLCs and Industrial Computers - User's guide*, CJ International, pp 428, France, 1993.
- IEC International Electrotechnic Commission, *Programmable Controllers, Part 3: Programmable Languages* , Publication 1131-1, 1992.
- INAMASU, R. Y.. *Modelo de FMS: Uma plataforma para Simulação e Planejamento*, Tese de Doutorado, Escola de Engenharia de São Carlos - Universidade de São Paulo, pp 134, São Carlos - SP , 1995.
- JACOBSON, I.; CHRISTERSON, M. JONNISON UND ÖVERGAARD, G; *Object-Oriented Software Engineering - A Use Case Driven Approach*. ACM Press, Addison Wesley, MA, 1992.
- JANG, J.; KOO, P-H; NOF-SHIMON, Y.; *Application of Design and Control Tools in a Multirobot Cell*, Computers and Industrial Engineering, v 32, n1, pp 89-100, 1997.
- JENG, M. D.; *Petri Nets for Modelling Automated Manufacturing Systems with Error Recovery*, IEEE Transactions on Robotics and Automation v 13 n5 pp 752-760 1997.

- JENG, M. D.; LIN, C. S.; *Schedulling flexible manufacturing systems containing assembly operations based on Petri net structures and dynamics*, Proceedings of the IEEE Inter. Conf. on Systems, Man and Cybernetics v 5 1997. P4430-4435.
- JENG, D. M.; SHIEH, C. L.; *Software tool for synthesizing petri nets in automated manufacturing systems*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics v 4 1993. p 302-306.
- JENSEN, K.; *Coloured Petri Nets, Basic Concepts, Analysis Methods and Pratical Use*, vol. 2 Spring-Verlag, 1995. p 174.
- JOHANSSON, K. : *Performance of logic controllers in industrial enviromment* IEEE Industry Application Society Annual Meeting p 1699-1702, 1991.
- JONES, A. T.; RABELO, L. C.; *Realtime decision making using neural nets, simulation, and generic algorithms*. Proceedings of the 2nd International FAIM Conference, Falls Church, Virginia, USA p 746-757, July 1992.
- KATSUMOTO, S.; MIZUNO, O.; KIKUNO, T.; HIRAYAMA, Y.; TAKAGI, Y.; SAKAMOTO, K., *New software project simulator based on generalized stochastic Petri net.*, Proceedings of IEEE International Conference on Software Engineering, Los Alamitos, CA, pp. 293-302, 1997.
- KELLING, C; HENZ, J.; HOMMEL, G.; *Modelling and evaluation of a distributed controller architecture by stochastic Petri nets*, International Journal of Mini and Microcomputers, v18, n3, pp. 157-163, 1996.
- KHOSHAFIAN, A. R.; *Object Orientation: Concepts, Languages, Databases, User Interfaces* John Wiley & Sons, Inc. New York, 431p, 1990.
- KLETZ, T. A.: *Eliminating potencial process hazards* Chemical Eng. p 48-68, April 1985.
- KOMODA, N.; KERA, K.; KUBO, T.; *An autonomous, decentralized control system for factory automation*. Computer, p 73-83, December 1984.
- KROGH, B. H.; HOLLOWAY, L. E.; *Synthesis of feedback control logic for discrete manufacturing systems*. Automatica v 27 n 4 Jul 1991 p 641-651.
- KULP, P.; DICESARE, F.; *Performance evaluation of colored Petri net models using simulation and moment generating functions*. Proc. of IEEE International Conference on Systems, Man, and Cybernetics, v1, 1998, p 154-159.
- KÜNZLE, L. A. *Controle de Sistemas Flexíveis de Manufatura - Especificação dos Níveis de Equipamento e Estação de Trabalho*. Curitiba, Agosto de 1990, Dissertação de mestrado, Centro Federal de Educação Tecnológica do Paraná, Universidade Federal do Paraná.
- KUSIAK, A. ; *Flexible manufacturing system: A structural approach*. International Journal of Production Research, v 23 n 6 p 1057-1073, 1985.
- LEE, K. C.; LU, I. Y.; LIN, H. H.; *PM-Net: a software project management representation model*, Information and Software Technology v 36 n 5 May 1994. p 295-308.
- LEMMER, K.; OBER, B.; SCHNIEDER, E., *Model-based programming and diagnosis for programmable logical controllers*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics v 5 1995. p 4474-4479
- LEONARDO, M.; *Técnica moderna para desenvolvimento de projeto de sistemas de comando em processos industriais*. Seminário de Instrumentação IPB, 4, Rio de Janeiro p 01-24, setembro 1981. (a)
- LEONARDO, M.; *Crítérios para dimensionamento e configuração de controladores programáveis*. Seminário de Instrumentação IPB, 4, Rio de Janeiro, p 603-618, setembro 1981. (b)

- LEVINE, R. I.; DRANG, D. E.; EDELSON, B.; *Inteligência Artificial e Sistemas Especialistas*. Trad Ratto, M. C. S. R., São Paulo, McGraw-Hill, 1988, 264p.
- LIN, J.; LEE, C. C.; *Petri net based integrated control and scheduling scheme for flexible manufacturing cells*. Computer Integrated Manufacturing Systems, v10, n2 1997, p109-122.
- LIU, B.; ROBBI, A.; *TiPNet: a graphical tool for timed petri nets*, Inter. Workshop on Petri Nets and Performance Models 1995. IEEE, Los Alamitos, CA, USA. p 212-213.
- LOBÃO, E. C. *Intertravamento de Máquinas-Ferramenta: Proposta de Roteiro para projeto e implementação, e estudo de caso em ambiente de manufatura flexível*, Dissertação de mestrado, Escola de Engenharia de São Carlos - USP, 1995.
- LUBECK, R.; *Tools description list, 1995*. www.informatik.uni-dortmund.de/luebeck/pg279/modprak/petri-tools.html.
- LUO, J.; GU, G.; *Information System design of manufacturing environments*. Journal of Computer Science and Technology, v13, dec 1998, p131-137.
- MACHADO, A.; *Comando numérico aplicado as máquinas ferramentas*. Editora Ícone, São Paulo, 1985.
- MACDERMOT, K. J.; YAO W. A.; *Developing a hibrid programmable logic controller plataform for a manufacturing system*. International Journal of Flexible Manufacturing Systems, v9 n4, Oct 1997, p367-374.
- MACDERMOT, K. J.; YAO W. A.; *PC – based programmable controller plataform fora flexible manufacturing system*. Proceedings of the Japan USA Symposium on Flexible Automation, v1, 1996, p525-528.
- MACLEAN C. R.; BROWN P. F.; *The architecture of the NBS factory automation research testbed*. IFAC. 10th Triennial World Congress. Munich, p141-146, 1987.
- MAIMON, O, Z.; *Real- time operational control of flexible manufacturing systems*. Journal of Manufacturing Systems. v 6, n 2, p 125-136, 1987.
- MALO, T. A. J.; GAVINO, C. D.; RAMIREZ, T. A.; *Petri net based control for the dynamic scheduling of a flexible manufacturing cell*. IEEE International conference on Systems Man and Cybernetics, v1, 1998, p553-557.
- MARCOS, M.; ORIVE, D.; *New solution for integrating control devices in computer integrated*. IEE conference Publication n 427/1 1996, p485-490.
- MASSBERG, W.; JANZEN, F.; MOHRLE, M.; SEIFERT, H. -J., *A Petri-net based automation environent*. Proc Rensselaer 2 Int Conf Comput Integr Manuf. Publ by IEEE, Computer Society, Los Alamitos, CA, USA. 1990. p 356-363.
- MAXITEC; *Controlador Lógico Programável MXT 130 - Instruções de Programação*, São Paulo, 1989 (a).
- MAXITEC; *CNC Sinumerik 3 - Operating Instrunctions*, edition 3.84, São Paulo, 1989. (b)
- MAXWELL, W. L.; MUCKSTADT, J. A.; *Design of automatic guided vehicle systems* . IIE transactions, v 14, n 2, p 114-124, June 1982.
- MENENDEZ, C.; KÜNZLE L. A.; *Controle de sistemas flexíveis de manufatura: uma abordagem estruturada*. UFES/SBA/SBI. 1ro Encontro Regional de Automação e Instrumentação - Vitória -ES , p 19-29, agosto de 1989.
- MERLIN, P.; *A methodology for the design and implementation of communication protocols*. IEEE Transactions on Communications v 24, n6, pp 614-621, 1976.
- MIKULCZYNSKI, T.; SAMSONOWICZ, Z.; WIECLAWEK, R.; *Using the net transformation method to program programmable logic controllers*, Control Engineering Practice, v 6, n8, Aug 1998 pp 989-996.

- MITCHEL, G. A.; *PLC I/O devices: More options for better machine control*, Control Engineering, v45, n3, Mar 1998 pp 127-134.
- MIYAGI, P. E.; HASEGAWA, K.; TAKAHASHI, K.; *A programming Language for Discrete Event Production Systems Based on Production Flow Schema and Mark Flow Graph*, Trans. of the Soc. of instrument and Control Engineers, v 24, n2, feb 1988 pp 77-84.
- MIYAGI, P. E.; *Controle Programável - Fundamentos de Controle de Sistemas de Eventos Discretos*. Editora Edgard Blücher Ltda, 1996.
- MOHAMMAD, R.; ITO, M. R.; LAWREMCE. P. D.; *GPNSAT: a new tool for simualtion and analysis of hybrid systems*. Proc of IEEE International Conf. on Systems, Man and Cybernetics, v3, 1997, p2240-2245 (a).
- MOHAMMAD, R.; LAWREMCE. P. D.; ITO, M. R.; *Hybrid modelling and simulation of manufacturing systems*. IEEE Symposium on Emerging Technologies and Factory Automation, ETFA, 1997, p555-560 (b).
- MOLOY, M. K., *Performance Analysis Using Stochastic Petri Nets* IEEE Transactions on Computers, v C31 n9, pp 913- 917, 1982.
- MOORE, K. E.; BRENNAN, J. E., *ALPHA/Sim simulation software tutorial*. Winter Simulation Conference Proceedings 1995. p 387-394.
- MOREAUX, M.; DECOTIGNIE, J. D. :*CNC and PLC: are they one and some for machine-tools?* IECON - 1991 p 924-927.
- MORNINGSTAR, D.; *Anatomy of a Machine tool*, Tooling & Production, june 1993, pp 68-74.
- MORANDIN, O. jr.; *Componentes de um protótipo de um AGV para um ambiente de manufatura integrada*. I Simpósio Brasileiro de Automação Inteligente, Rio Claro - SP, anais, p 455-462, setembro de 1993.
- MURATA, T.; *Petri Net: Properties, analysis and applications*; Proceedings of the IEEE, vol 77, n.4, p. 541-579, April, 1989.
- MURATA, T.; KOMODA, N.; MATSUMOTO, K.; HARUNA, K.; *A Petri Net-based controller for flexible and Maintainable sequence control and its applications in factory automation*, IEEE Trans. on Industrial Electronics, v IE-33, n.1, febr. 1986. pp. 1-8.
- MURGATROY, R. J.; MASORY, O.; *Petri net framework for simulation and control of manufacturing systems*, Proceedings of ASME Dynamic System and Control Division American Society of Mechanical Engineers, v58, 1996, p265-273.
- NAGAO, Y.; URABE, H.; NAKANO, S.; KUMAGAI, S., *Application of Petri nets to sequence control*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences v E76-A n 10 Oct 1993. p 1598-1606.
- NAGAO, Y.; NAKANO, S.; HIGUMA, K.; KUMAGAI, S.; *Development of a Petri net-based simulator for factory automation systems*, Electronics & Communications in Japan, Part III: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi) v 79 n 3 Mar 1996. p 64-79.
- NAGURKA, M. L.; ENGLERT P. J.; *Toward an Intelligent Machine Tool for Flexible Manufacturing*, Robotics & Comp. Integrated manufacturing, v6, n3, 1989, p229-236.
- NATALE, F.; *Automação Industrial, Ed. Érica, São Paulo, 1995, p312.*
- OSHIRO, O. T. :*Desenvolvimento de um CNC baseado no microprocessador de 16/32 bits (Motorola 68000) de baixo custo e alta flexibilidade*. São Carlos, 1993 ,Dissertação de mestrado. Escola de Engenharia de São Carlos, Universidade de São Paulo.

- OSWALD, H.; ESSER, R.; MATTMANN, R., *Environment for specifying and executing hierarchical Petri nets*. Proceedings - International Conference on Software Engineering n 12. p 164-172, 1990.
- OTTER, J. D.; *Programmable Logic Controllers: Operation, Interfacing and Programming*, New York, Prentice-Hall, 1988.
- OWEN, J. V.; *Open Up de Plant Floor*, *Manufactory Engineering*, v117, n5 nov, 1996.
- PEDERSEN, K.; *Well Structured Knowledge Bases, AI Expert*, nov 1989, pp 36-40
- PENG, C.; CHEN, F. F.; *Simulation based optimization approach to real time control and scheduling of flexible manufacturing systems* Technical Paper Society of Manufacturing Engineers, MR, 1998, p1-6.
- PEREIRA, C. E.; *Métodos de Análise de Sistemas Tempo-Real usando Técnicas de Orientação a Objetos*, Tutorial - X Simpósio Brasileiro de Engenharia de Software, UFSCar São Carlos, SP. Outubro 1996.
- PETERSON, J. L.; *Petri nets theory and the modelling of systems* Englewood Cliffs, NJ: Prentice Hall, 1981.
- PETERSON, J. L.; SILBERSCHOTZ, A.; *Operating Systems Concepts*, Reading, MA: Addison- Wesley, 2nd ed., 1985.
- POLITANO, P. R.. *Especificação e implementação de uma unidade de controle para célula flexível de manufatura*. São Carlos, Agosto de 1993, Dissertação de mestrado, Escola de Engenharia de São Carlos - Universidade de São Paulo.
- POLITANO, P. R.. *Procedimento de Programação Reativa On Line de FMS*, Tese de Doutorado, Escola de Engenharia de São Carlos - Universidade de São Paulo, pp 139, São Carlos - SP , 1996.
- PORTO, A. J. V.; *Introdução ao controle numérico*; Pub. interna, EESC - USP, 1985.
- PROCTOR, F. M.; ALBUS, J. S.; *Open-architecture controllers*, *IEEE Spectrum* june 1997. p 60-64.
- QIU, J.; WE, X.; PENG, Z.; XU, Y.; *Reserch on fault diagnostic expert system based on fuzzy petri nets for FMS machining cell*, *IEEE proceedings of International Conference on Industrial technology*, 1996, p122-125.
- RAJU, K. R.; CHETTY, O. V. K.; *Design and evaluation of automated guided vehicle systems for flexible manufacturing systems: an extended timed Petri Net-based approach*, *International Journal of Production Research*, v31, n5, maio de 1993, pp 1069-1096.
- RAUSH, M.; KROUG, B.; *Transformation between different model forms in discrete event systems*, *IEEE proceedings of International Conference on System Man and Cybernetics*, v3, 1997, p2841-2846.
- REISIG, W.; *Petri Nets: An Introduction*, Springer-Verland, 1985, 161 pg.
- RAMAMOORTHY, C. V; HO, G. S.; *Performance Analysis of Asynchronous Concurrent Systems using Petri Nets* *IEEE Transactions on Software Engineering* v SE-6, n5, pp 440-449, 1980.
- RAMCHANDANI, C.; *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets* Ph. D. Thesis, Massachusstes Institute of Technology, Cambridge, MA, 1973.
- RAVICHANDRAN R.; CHAKRAVARTY, A. K.; *Decision suport in flexible manufacturing systems using timed Petri nets*, *Journal of Manufacturing Sitems*, v4 , n2, 1986, pp 89-101.

- RILLO, M.; *Controle de Sistema de Manufatura por Redes de Petri e Regras de Produção*. XXII Congresso Nacional de Informática - SUCESSU - anais- São Paulo, p 677-681. Setembro de 1989.
- ROHN, D.; RENEHAN, J.; *Open, Flexible, and reliable?* Industrial Computing, v17, n12, dec 1998, p 20-23.
- ROWE, W. B.; LI, Y; MILLS, B.; ALLANSON, D. R.; *Aplication of intelligent CNC in grinding Computer in Industry*, v31, n1, oct 1996, p45-60.
- SANTOS, D. J. F.; *Controle de Sistemas Antropocêntricos de Produção Baseado em Redes de Petri Interpretadas*. Tese de doutorado EPUSP – São Paulo, p287, 1998.
- SCHOROER, B. J.; TSENG F. T.; *An intelligent assistant for manufacturing system simulation*, International Journal of Production Research, v27, n10, 1989, pp 1665-1683.
- SHARIT, J.; SALVENDY, G.; *A real time interactive computer model of a flexible manufacturing system*. IIE Transaction, v 19 n2 p 167-177, June 1987.
- STÖRRLE, H.; *An evaluation of High-End Tools for Petri Nets*, Ludwig Maximilians Universität München – Institut für Informatik, Bericht 9802, june 1998, p40.
- STREGE, B.; TOLLE, H.; LOYDL, H., *Automatic transformation of simple user commands to hierarchical petri nets*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics v 2 1995. IEEE, Piscataway, NJ, USA, p 1861-1866.
- SIEMENS; *Operating Instrunctions - PU 675 - Programming Unit*, Federal Republic of Germany, 1989.
- SILVA, M.; VELILLA, S.; *Programmable Logic Controllers and Petri Nets: A Comparative Study*, IFAC Software for Computer Control, Madrid Spain, 1982, p 83 – 88.
- SLAETS, A. F.; *Automação de sistemas digitais*, Notas de Aula, Dpto Eng. Elétrica, EESC - USP, 1994.
- STERNEMAN, K. H.; ZELM, M.; *Enterprise modelling for operational decision support* IEEE International Conference on Systems, Man and Cybernetics. v1, 1998, Piscataway, NJ, USA, pp.301-306.
- SUN, T. H.; FU, L. C., *Automatic Petri net generator for modeling a flexible manufacturing system*. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. v1, 1994, Piscataway, NJ, USA, pp. 670-675.
- SZELKE, E.; MESZAROS, I.; *Knowledge-based Adaptative Control of FMS Contributing to CIM*, Computer in Industry, 14, 1990, pp 89-98.
- TAHOLAKIAN, A.; HALES, W. M. M.; *PN → PLC: A methodology for designing, simulating and coding PLC based control systems using Petri nets*, International Journal of Production Research v 35 n 6 1997. pp 1743-1762.
- TANG, G. Y.; HURA, G. S., *Petri-net-based modeling assisted software environment (MASE) tool*, Proceedings - IEEE Computer Society's International Computer Software & Applications Conference 1994. IEEE, Los Alamitos, CA, USA, p 439-444.
- TANG, T.; ZHOU, M., *Development and applications of a Petri net simulation tool for manufacturing systems*. Flexible Automation 1992 Proc 92 Jpn USA Symp Flexible Autom. Publ by ASME, New York, NY, USA. p 1199-1205.
- TASHIRO, T.; KOMODA, N.; TSUSHIMA, I.; MATSUMOTO, K.; *Advanced software for combinatorial control of discrete event system-rule-based control software for factory automation*. International Conference on Computer Aided Technologies, Montreal 1985. Proceedings ACM/IEEE, p 132-137. 1985.

- TELEMECANIQUE; PL7-3 GRAFCET Language - Manual de progr.**, São Paulo, 1989.
- THE CHARLES STARK DRAPER LABORATORY;** *Automation and management systems division. Flexible manufacturing systems handbook.* New Jersey, Moyes Publications, p 392, 1984.
- THURNER, E. M.;** *TOMSPIN - a tool for modelling with stochastic Petri Nets*, International Workshop on Petri Nets and Performance Models 1995. IEEE, Los Alamitos, CA, USA. p 218-219.
- TROY, D.; McQUEEN, M.,** *A development environment for batch process control.* Computers In Industry v31 1996. p 61-84.
- TRAUB,** *TRAUBOMATIC - Manual de programação e operação de tornos CNC*, São Paulo, 1993.
- TROMPEDELLER, M.;** *Petri net tools*, 1993. Last update, 1995, www.dsi.unimii.it/Users/Tesi/Trompede/Petri/Alfa.html.
- UZAM, M.; JONES, A. H.;** *Discrete event control system design using automation Petri nets and their ladder diagram implementation.* International Journal of Advanced Manufacturing Technology, v 14, n10, 1998, p 716-728.
- VALVANIS, K. P.;** *On the hierarchical modeling analysis and simulation of flexible manufacturing systems with extended Petri Nets.* IEEE Transactions on Systems, Man and Cybernetics, v 20, n 1, p 94-110, Jan/Feb. 1990.
- VENKATESH, K.; ZHOU, M.; CAUDILL, R.,** *Comparing Ladder Logic Diagrams and Petri Nets for Sequence Controller Design through a Discrete Manufacturing System*, IEEE Transactions on Industrial Electronics v41, n6 Dec. 1994, pp. 611-619.
- VENKATESH, K.; ZHOU, M.; CAUDILL, R.,** *Design of sequence controllers using Petri net models*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics v 4 1995. IEEE, Piscataway, NJ, USA, p 3469-3474.
- VERMA, M. R.; CHANDRA, V.;** *The design and development of a fail-safe interlocking system using microprocessor for Indian Railways* IEEE Region, 4 International Conference, 10 Tencon'89 .p 511-514, november 1989.
- VISWANADHAM, N; NARAHARI, Y. :** *Stochastic Petri net models for performace evaluation of automated manufacturing systems.* Publ. by Elsevier Science Publishers B. V. North Holland, p 125-142, 1988.
- VISWANADHAM, N; NARAHARI, Y.; JOHNSON, T. L. :** *Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models* IEEE Transactions on Robotics and Automation, vol 6 no 6 Dec. 1990 p. 713 - 723.
- XIE, X.;** *Class of petri nets for modular modeling*, Proceedings - IEEE International Conference on Robotics and Automation v 2 1995. IEEE, Piscataway, NJ, USA. p 2066-2071.
- XING, K.;** *Deadlock avoidance policity for Petri Net modelling of Flexible Manufacturing Systems with shared resources.* IEEE Transactions on Robotics and Automation, v 41, n 2 pp 289-295, 1996.
- YAN, H. S.; WANG, N. S.; ZHANG, J. G.; CUI, X. Y.;** *Modelling, schedulling and simulation of flexible manufactory systems using extended stochastic hig-level evaluation Petri Nets*, Robotics and Computer Integrated Manufacturing, v14 n2 apr 1998. P121-140.
- YIM, P.; LEFORT, A.; HEBRARD, A.,** *System modelling with hypernets*, IEEE Symposium on Emerging Technologies & Factory Automation v 1 1995. IEEE, Piscataway, NJ, USA. p 37-47.

- WANG, H. P. B.; HAFEEZ, S. A.; *Performance evaluation of tandem and conventional AGV systems using generalized stochastic Petri nets*, International Journal of Production Research v 32 n 4 1994. p 917-932.
- WANG, L. C.; WU, S. Y.; *Modelling with colored timed object-oriented Petri nets for automated manufacturing systems*, Computers in Industrial Engineering, v34 n2 apr. 1998, p 463-480.
- WATERMAN, D. A.; *A Guide to Expert Systems*, London, Ed. Addison-Wesley, 1986, 419p.
- WATSON III, J. F; DESROCHERS A. A.; *Applying Generalized Stochastic Petri Nets to Manufacturing Systems Containing Nonexponential Transitions Functions*, IEEE Transactions on Systems, Man, and Cybernetics, v 21 n5, pp. 1008-1017, sept/oct. 1991.
- WIETING, R.; *Hybrid high level nets*. IEEE - Winter Simulation Conference Proc. 1996, p848-855.
- WIKARSKI, D.; *Petri Net tools – a comparative study*. Technical report, TU Berlin, Fachbereich Informatik, 1997, p 97-94.
- WILLSON, R. G.; KROGH, B. H.; *Petri Net for the Specification and Analysis of Discrete Controllers*. IEEE Transactions on Software Engineering, v16, n1, jan. 1990, p 39-50.
- WONGTALADKOWN, C.; CHEN, Y., *A real-time Petri nets simulator for automated manufacturing systems*. Proc 5 IEEE Int Symp Intell Control 90. p 999-1004, 1990.
- ZAYTOON J.; *Specification and design of logic controllers for automated manufacturing systems*, Robotics and computer Integrated Manufacturing, v 12, n4, Dec 1996, p 353-366.
- ZHOU, M; DiCESARE, F.; *Adaptative Design of Petri Net Controllers for Error recovery in Automated Manufacturing Systems*, IEEE Transactions on Systems, Man and Cybernetics, v 19, n5, pp. 963-973, sept. 1989.
- ZHOU, M; DiCESARE, F.; *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers, 1993, p223.
- ZHOU, M; TWISS, E.; *Comparison of relay ladder logic programming and Petri Net approach for sequential industrial control systems*, IEEE Conference on Control Applications - Proceedings 1995, Piscataway, NJ, USA. Pp 748-753.
- ZUBERECK, W. M. *Timed Petri Nets and preliminary performance evaluation*. Proc. 7th Annual Symp. On Computer Architecture, La Baule, France, May 1980, pp 88-96.

ANEXO A – RELAÇÃO DOS MÓDULOS DO ADIAC

A relação entre os módulos programados no Delphi 2.0 para o ADIAC podem ser visualizada na Figura A.1 abaixo:

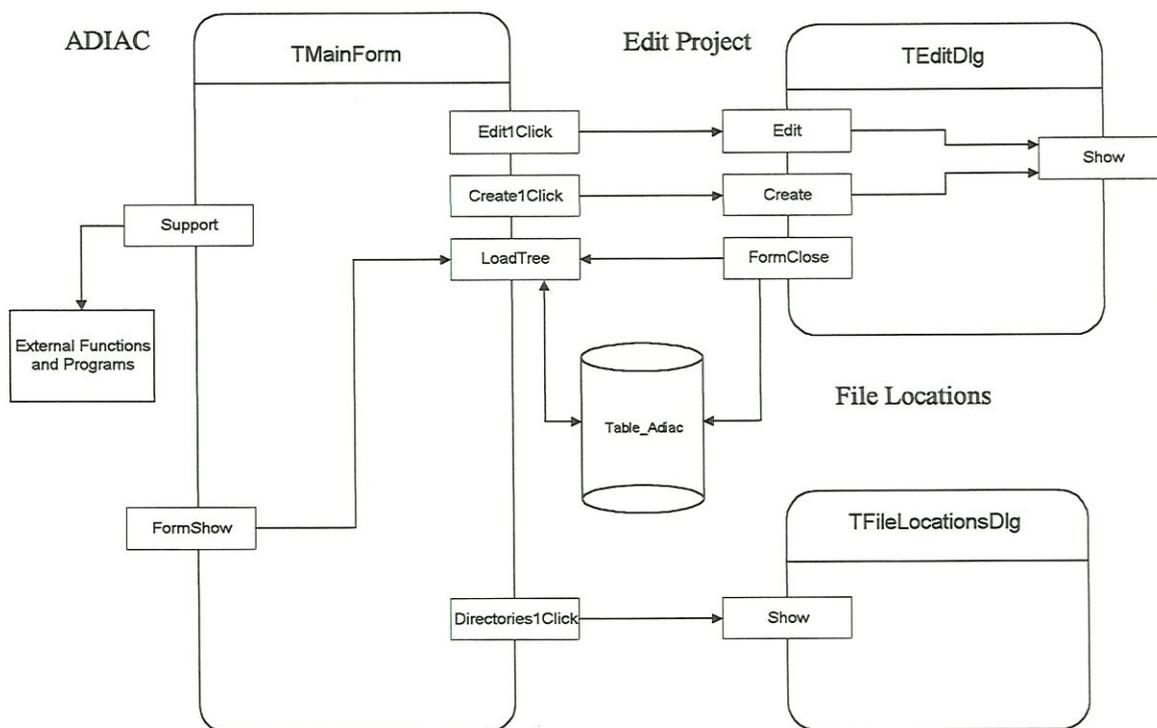


Figura A.1 - Relação dos módulos implementados

A Figura A.1 mostra que a partir do formulário principal (*TmainForm*), com um clique sobre o botão editor (*Edit1Click*) é possível abrir um registro no bando de dados da tela de edição (*TeditDlg*, que corresponde a *Edit Project*) e se editar a descrição do protótipo (*show*). De forma semelhante o botão create (*Create1Click*) cria um registro em branco para ser editado e mostrado (*show*) e permite que o registro seja gravado, adicionado no banco de dados *Table_Adiac* utilizando o path desejado (*LoadTree*). A busca no diretório é acionada pelo item *FormShow* que coloca a árvore de projetos na lista *projects*.

A função *support* liga os blocos em desenvolvimento de programas e funções externas. E a função *FileLocation*, acionada por um clique (*Directories1Click*) busca o path do editor de redes de Petri (função provisória para teste de funções implementadas até o momento).

A partir da Figura A.1 podemos descrever as definições dos módulos implementados no Delphi 2.0:

Definições dos Módulos

O projeto possui um módulo para cada janela da interface. Elas são *MainForm*, *EditDlg* e a *FileLocationsDlg*.

MainForm

Public Interface

procedure LoadTree

Carrega a lista de projetos. Deve ser chamado toda vez que uma mudança é feita na lista de projetos.

Private Information

index: Longint

Controla o projeto atual selecionado na lista de projetos. Usado para atualizar a informação do descritor.

Events Procedures

procedure FormShow(Sender: TObject)

Quando o formulário é mostrado pela primeira vez, ele chama o procedimento LoadTree para montar a janela da árvore de projeto (project list).

procedure TreeView_FilesChange(Sender: TObject; Node: TTreeNode)

Atualiza o índice da instância cada vez que o usuário seleciona um projeto de uma lista de projetos.

procedure Edit1Click(Sender: TObject)

Responde a opção Edit do menu File. Chama a caixa de diálogo EditDlg para editar uma informação do projeto.

procedure Create1Click(Sender: TObject)

Responde a opção Create no menu File. Chama a caixa de diálogo CreateDlg para criar um novo projeto.

procedure Rename1Click(Sender: TObject)

Responde a opção Rename no menu File menu. Renomeia um projeto.

procedure Copy1Click(Sender: TObject)

Responde a opção Copy no menu File. Copia um projeto.

procedure Delete1Click(Sender: TObject)

Responde a opção Delete no menu file. Apaga o projeto selecionado da lista de projetos.

procedure Directories1Click(Sender: TObject)

Responde a opção File Locations do menu File. Chama a janela FileLocationsDlg.

procedure Exit1Click(Sender: TObject)

Responde a opção Exit do menu File. Fecha o formulário.

procedure Editamodel1Click(Sender: TObject)

Responde a opção Edit a model no menu Support. Chama o programa editor de redes de Petri Net.

EditDlg

Public Interface

procedure Create

Mostra a caixa de diálogo no modo de criação.

procedure Edit

Mostra a caixa de diálogo no modo de edição.

Private Information

Cancel: Boolean

Flag que indica quando o botão Cancel foi apertado

Events Procedures

procedure FormShow(Sender: TObject)

Inicializa o formulário.

procedure FormClose(Sender: TObject; var Action: TCloseAction)

Atualiza a base de dados e a tela do MainForm.

procedure Button_OKClick(Sender: TObject)

Responde ao clique do botão OK. Fecha o formulário.

procedure Button_CancelClick(Sender: TObject)

Responde ao clique do botão Cancel. Fecha o formulário.

procedure Button_BrowseClick(Sender: TObject)

Responde ao clique do botão Browse. Chama uma caixa de diálogo aberto para localizar um arquivo de projeto.

FileLocationsDlg

Events Procedures

procedure Button_BrowseClick(Sender: TObject)

Responde ao clique do botão Browse. Chama todas as caixas de diálogo aberto para localizar o arquivo de programa de ferramentas de edição de redes de Petri.

procedure FormClose(Sender: TObject; var Action: TCloseAction)

Atualiza a base de dados.

procedure FormShow(Sender: TObject)

Inicializa o formulário.

procedure Button_OKClick(Sender: TObject)

Fecha o formulário correspondente ao clique do botão OK.

ANEXO B – PROGRAMAÇÃO CLP PARA UM TORNO CNC

Programação CLP

Bloco de organização:

A	F0.0	Emergência	
A	F0.3	Início de ciclo	t1
R	F0.3	Desliga ciclo	
S	F0.1	Sistema desligado	
A	I0.0	Botão de Emergência	
A	F0.3	Início de ciclo	t2
R	F0.3	Desliga ciclo	
S	F0.1	Sistema desligado	
A	F0.1	Sistema ligado	
A	I0.1	Botão liga sistema	t3
R	F0.1	Liga o sistema	
S	F0.3	Início de ciclo	
A	F0.3	Flag de início de ciclo	
A	F0.2	Flag de 1ro scan	
S	F0.4	Liberar acionamentos	t4
S	F0.5	Ativar hidráulico	
S	F0.6	Ativar lubrificação	
R	F0.2	Reset do flag de 1ro scan	
A	F0.3	Flag de início de ciclo	
A	F1.7	Flag de inicialização OK	t5
A	F0.7	Comunicação CNC	
S	F1.3	Ciclo de comunicação CNC	
A	F0.3	Flag de início de ciclo	
A	F1.7	Flag de inicialização OK	t6
A	F1.0	Referenciamento	
S	F1.4	Ciclo de referenciamento	
A	F0.3	Flag de início de ciclo	
A	F1.7	Flag de inicialização OK	t7
A	F1.1	Manual	
S	F1.5	Ciclo manual	
A	F0.3	Flag de início de ciclo	
A	F1.7	Flag de inicialização OK	t8
A	F1.2	Automático	
S	F1.6	Ciclo automático	
A	F0.3	Flag de início de ciclo	
A	F1.7	Flag de inicialização OK	t28
A	F10.6	Início trans. recepção de status	
S	F4.1	Ciclo de trans. recepção de status	

Bloco de partida dos acionamentos e hidráulico:

A	F0.4	Libera acionamentos	
S	Q0.0	Liga os acionamentos	t9
R	F0.4	Flag de acionamentos	
S	F2.0	Teste dos acionamentos	
AN	I0.2	Não há confirmação dos acionamentos	
A	F2.0	Teste dos acionamentos	t10
R	F2.0	Reset do teste dos acionamentos	
S	F2.2	Falha nos acionamentos	
A	I0.2	Há confirmação dos acionamentos	
A	F2.0	Teste dos acionamentos	t11
R	F2.0	Reset do teste dos acionamentos	
S	F2.3	Acionamentos OK	
A	I4.0	Pressostato do hidráulico	
A	F0.5	Auxiliar Liga hidráulico	
A	F3.5	Disj. Térmico do hidráulico OK	t12

S	Q0.1	Liga hidráulico	
R	F0.5	Auxiliar liga hidráulico	
S	F2.1	Teste do hidráulico	
A	I0.3	Há confirmação do hidráulico	t13
A	F2.1	Teste do hidráulico	
R	F2.1	Reset do teste do hidráulico	
S	F2.5	Hidráulico OK	
AN	I0.3	Não há confirmação do hidráulico	t14
A	F2.0	Teste do hidráulico	
R	F2.1	Reset do teste do hidráulico	
S	F2.6	Falha no hidráulico	
AN	I3.5	Falha na pressão do óleo no hidráulico	t77
S	F2.6	Falha no hidráulico	
AN	I4.3	Falha na pressão do óleo no hidráulico	t78
S	F2.6	Falha no hidráulico	
A	F2.2	Falha nos acionamentos	t79
A	F11.0	Acionamentos ligados	
R	F11.0	Reset de acionamentos ligados	
S	F10.7	Desliga acionamentos	
A	F2.3	Acionamentos OK	t80
A	F10.7	Acionamentos desligados	
R	F10.7	Reset de acionamentos desligados	
S	F11.0	Liga acionamentos	
A	F2.5	Falha no hidráulico	t81
A	F11.2	Hidráulico ligado	
R	F11.2	Reset do hidráulico ligado	
S	F11.1	Desliga hidráulico	
A	F2.6	Hidráulico OK	t82
A	F11.1	Hidráulico desligado	
R	F11.1	Reset do hidráulico desligado	
S	F11.2	Liga hidráulico	
A	F2.3	Acionamentos OK	t15
A	F2.4	Lubrificação OK	
A	F2.5	Hidráulico OK	
R	F2.3	Reset dos acionamentos	
R	F2.4	Reset da lubrificação	
R	F2.5	Reset do hidráulico	
S	F1.7	Inicialização OK	

Bloco de partida da lubrificação:

A	I3.6	Nível de óleo OK	
A	I4.1	Proteção térmica OK	
A	F0.6	Ciclo de lubrificação	t16
S	Q0.2	Liga o motor da lubrificação	
R	F0.6	Reset do flag de início de ciclo	
S	T1	Aciona temporizador T1	
A	T1	Tempo atingido	t74
S	F2.7	Auxiliar aciona motor da lubrificação	
A	I1.0	Há confirmação da lubrificação	t18
A	F2.7	Teste da lubrificação	
R	F2.7	Reset do teste da lubrificação	
S	F3.0	Lubrificação ligada	
AN	I1.0	Não há confirmação da lubrificação	t17
A	F2.7	Teste da lubrificação	
R	F2.7	Reset do teste da lubrificação	
S	F3.6	Falha na lubrificação	
AN	I3.6	Falha no nível de óleo da lubrificação	t83
S	F3.6	Falha da lubrificação	
AN	I4.1	Falha na proteção térmica da lubrificação	t84
S	F3.6	Falha da lubrificação	

A	F3.0	Lubrificação ligada	
R	Q0.2	Desliga motor da lubrificação	t19
S	T2	Liga temporizador T2	
R	F3.0	Flag de lubrificação ligada	
A	T2	Tempo T2 atingido	t75
S	F3.1	Conta ciclo de lubrificação	
S	F3.2	Ciclo completo	
A	F3.3	Condição de set do contador C1	
L	3	Valor de contagem	t21
R	F3.3	Reset da condição de set do contador C1	
S	C1	Set do contador C1	
A	F3.1	Conta ciclo de lubrificação	
R	F3.1	Reset da contagem do ciclo de lubrificação	t23
CU	C1	Contagem	
A	F3.4	Condição de reset do contador C1	
R	F3.4	Reset da condição de reset do contador C1	t20
R	C1	Reset do contador C1	
A	C1	Contagem atingida	t22
S	F3.5	Flag de contagem atingida	
A	F3.5	Flag de contagem atingida	
A	F3.2	Ciclo completo	
R	F3.2	Reset da contagem de ciclo	
R	F3.5	Reset do flag de contagem atingida	
S	F3.4	Flag de condição de reset do contador C1	t26
S	F2.4	Lubrificação OK	
S	F3.7	Flag de acionamento manual da lubrificação	
S	F4.0	Flag de acionamento automático da lubrificação	
A	F3.2	Flag de ciclo da lubrificação	
AN	F3.5	Flag de contagem atingida de ciclos	t24
R	F3.2	Reset do flag de ciclo de lubrificação	
S	F0.6	Flag de início de ciclo da lubrificação	
A	F3.7	Flag de acionamento manual da lubrificação	
A	I1.1	Flag de acionamento manual da lubrificação	t25
R	F3.7	Reset do flag de acionamento manual da lubrificação	
S	F0.6	Flag de início de ciclo da lubrificação	
A	I4.0	Flag de habilitação do modo automático da lubrif.	
A	F1.2	Habilita o modo automático	t27
R	F4.0	Reset do flag de habilitação do automático	
S	T3	Aciona temporizador T3	
A	T3	Tempo T3 atingido	t75
S	F0.6	Flag de início de ciclo da lubrificação	

Bloco de seleção do modo de operação:

A	I0.4	Modo automático	
S	F4.2	Ativa o modo automático	t29
S	F4.3	Desativa os outros modos	
A	F4.2	Flag de ativa o modo automático	
A	F7.1	Flag de modo automático desativado	t33
R	F7.1	Reset do flag de modo automático desativado	
R	F4.2	Reset do flag de ativa o modo automático	
S	F1.2	Flag de ativa modo automático	
A	F4.5	Flag de ativa o modo manual	
A	F1.2	Flag de ativa modo automático	t32
R	F4.5	Reset do flag de ativa modo manual	
R	F1.2	Reset do flag de ativa o modo automático	
S	F7.1	Flag de desativa modo automático	
A	F4.7	Flag de ativa o modo de comunicação	
A	F1.2	Flag de ativa modo automático	t31
R	F4.7	Reset do flag de ativa modo de comunicação	
R	F1.2	Reset do flag de ativa o modo automático	
S	F7.1	Flag de desativa modo automático	

A	F5.1	Flag de ativa o modo de referenciamento	
A	F1.2	Flag de ativa modo automático	130
R	F5.1	Reset do flag de ativa modo de referenciamento	
R	F1.2	Reset do flag de ativa o modo automático	
S	F7.1	Flag de desativa modo automático	
AN	F6.2	Programa CN ativo	
A	I0.5	Modo manual	
S	F4.4	Ativa o modo manual	134
S	F4.5	Desativa os outros modos	
A	F4.4	Flag de ativa o modo manual	
A	F7.2	Flag de modo manual desativado	138
R	F7.2	Reset do flag de modo manual desativado	
R	F4.4	Reset do flag de ativa o modo manual	
S	F1.1	Flag de ativa modo manual	
A	F4.3	Flag de ativa o modo automático	
A	F1.1	Flag de ativa modo manual	137
R	F4.3	Reset do flag de ativa modo automático	
R	F1.1	Reset do flag de ativa o modo manual	
S	F7.2	Flag de desativa modo manual	
A	F4.7	Flag de ativa o modo de comunicação	
A	F1.1	Flag de ativa modo manual	136
R	F4.7	Reset do flag de ativa modo de comunicação	
R	F1.1	Reset do flag de ativa o modo manual	
S	F7.2	Flag de desativa modo manual	
A	F5.1	Flag de ativa o modo de referenciamento	
A	F1.1	Flag de ativa modo manual	135
R	F5.1	Reset do flag de ativa modo de referenciamento	
R	F1.1	Reset do flag de ativa o modo manual	
S	F7.2	Flag de desativa modo manual	
AN	F6.2	Programa CN ativo	
A	I0.6	Modo de comunicação	
S	F4.6	Ativa o modo de comunicação	139
S	F4.7	Desativa os outros modos	
A	F4.6	Flag de ativa o modo de comunicação	
A	F7.3	Flag de modo de comunicação desativado	143
R	F7.3	Reset do flag de modo de comunicação desativado	
R	F4.6	Reset do flag de ativa o modo de comunicação	
S	F0.7	Flag de ativa modo de comunicação	
A	F4.3	Flag de ativa o modo automático	
A	F0.7	Flag de ativa modo de comunicação	142
R	F4.3	Reset do flag de ativa modo automático	
R	F0.7	Reset do flag de ativa o modo de comunicação	
S	F7.3	Flag de desativa modo de comunicação	
A	F4.5	Flag de ativa o modo manual	
A	F0.7	Flag de ativa modo de comunicação	141
R	F4.5	Reset do flag de ativa modo manual	
R	F0.7	Reset do flag de ativa o modo de comunicação	
S	F7.3	Flag de desativa modo de comunicação	
A	F5.1	Flag de ativa o modo de referenciamento	
A	F0.7	Flag de ativa modo de comunicação	140
R	F5.1	Reset do flag de ativa modo de referenciamento	
R	F0.7	Reset do flag de ativa o modo de comunicação	
S	F7.3	Flag de desativa modo de comunicação	
AN	F6.2	Programa CN ativo	
A	I0.7	Modo de referenciamento	
S	F5.0	Ativa o modo de referenciamento	144
S	F5.1	Desativa os outros modos	
A	F5.0	Flag de ativa o modo de referenciamento	
A	F7.4	Flag de modo de referenciamento desativado	148
R	F7.4	Reset do flag modo de referenciamento desativado	
R	F5.0	Reset do flag de ativa o modo de referenciamento	
S	F1.0	Flag de ativa modo de referenciamento	
A	F4.3	Flag de ativa o modo automático	

A	F1.0	Flag de ativa modo de referenciamento	t47
R	F4.3	Reset do flag de ativa modo automático	
R	F1.0	Reset do flag de ativa o modo de referenciamento	
S	F7.4	Flag de desativa modo de referenciamento	
A	F4.5	Flag de ativa o modo manual	
A	F1.0	Flag de ativa modo de referenciamento	t46
R	F4.5	Reset do flag de ativa modo manual	
R	F1.0	Reset do flag de ativa o modo de referenciamento	
S	F7.4	Flag de desativa modo de referenciamento	
A	F4.7	Flag de ativa o modo de comunicação	
A	F1.0	Flag de ativa modo de referenciamento	t45
R	F4.7	Reset do flag de ativa modo de comunicação	
R	F1.0	Reset do flag de ativa o modo de referenciamento	
S	F7.4	Flag de desativa modo de referenciamento	

Bloco de falha do hidráulico:

A	F2.6	Falha no hidráulico	
R	F2.6	Reset da falha no hidráulico	
S	F10.0	Envia mensagem de falha no hidráulico	
S	F5.2	Aborta programa do CNC	t49
S	M05	Desliga eixo árvore	
S	F5.3	Posiciona eixos em lugar seguro	
S	M09	Desliga a refrigeração	
S	F5.4	Desliga mancal	
A	F7.5	Eixo árvore desligado	
A	F7.6	Eixos posicionados em lugar seguro	
A	F6.7	Refrigeração desligada	T68
A	F7.7	Mancal desligado	
S	F0.0	Emergência	

Bloco de transmissão e recepção de status:

A	F4.1	Transmissão e recepção de status	
A	I1.7	Pedido de transmissão	t50
R	F4.1	Reset do flag de transmissão e recepção de status	
S	F5.5	Flag auxiliar de transmissão	
A	F5.5	Flag auxiliar de transmissão	
AN	I2.0	Não há confirmação de transmissão	t51
R	F5.5	Reset do flag auxiliar de transmissão	
S	F5.6	Flag de falha na transmissão	
A	F5.5	Flag auxiliar de transmissão	
A	I2.0	Há confirmação de transmissão	t52
R	F5.5	Reset do flag auxiliar de transmissão	
S	F4.1	Transmissão e recepção de status	
A	F4.1	Transmissão e recepção de status	
A	I5.7	Pedido de recepção	
S	Q0.5	Ativa pedido de recepção	t53
R	Q0.6	Reset da recepção ACK	
R	F4.1	Reset do flag de transmissão e recepção de status	
S	F6.0	Flag auxiliar de recepção	
A	F6.0	Flag auxiliar de recepção	
A	F6.1	Há confirmação de transmissão	t54
R	F6.0	Reset do flag auxiliar de transmissão	
S	F4.1	Transmissão e recepção de status	
A	F6.0	Flag auxiliar de recepção	
AN	F6.1	Não há confirmação de recepção	t55
R	F6.0	Reset do flag auxiliar de recepção	
S	F6.2	Flag de falha na recepção	

Bloco de referenciamento do eixo x:

A	I1.2	Sensor eixo X +	
A	F1.4	Flag de referenciamento do eixo X	
S	F11.3	Set do flag de envio do eixo X para o sentido “-“	t56
R	F11.4	Reset do flag de envio do eixo X para o sentido “+“	
S	F0.3	Set do flag de início	

A	I1.4	Sensor eixo X -	
A	F1.4	Flag de referenciamento do eixo X	
R	F11.3	Reset do flag de envio do eixo X para o sentido “-“	t58
S	F11.4	Set do flag de envio do eixo X para o sentido “+“	
S	F0.3	Set do flag de início	
A	I1.3	Sensor eixo Xref	
A	F1.4	Flag de referenciamento do eixo X	
R	F11.3	Reset do flag de envio do eixo X para o sentido “-“	t59
S	F11.4	Set do flag de envio do eixo X para o sentido “+“	
S	F6.3	Set do flag marca encontrada	
AN	I1.2	Sem atingir o sensor eixo X +	
AN	I1.4	Sem atingir o sensor eixo X -	
AN	I1.3	Sem atingir o sensor eixo Xref	t57
S	F11.3	Flag de envio do eixo X para o sentido “-“	
R	F11.4	Flag de envio do eixo X para o sentido “+“	
S	F0.3	Set do flag de início	
A	F6.3	Flag marca encontrada	
A	F6.4	Flag de eixo não referenciado	
R	F6.4	Reset do flag de eixo não referenciado	
R	F6.3	Reset do flag de marca encontrada	t60
R	F1.4	Reset do flag de referenciamento do eixo X	
S	F6.5	Flag de eixo X referenciado	
A	F6.5	Flag de eixo X referenciado	
S	F0.1	Flag de sistema ligado	t61
R	F6.5	Reset do flag de eixo X referenciado	
S	F6.4	Flag de eixo X não referenciado	

Bloco de acionamento da refrigeração:

A	I1.5	Botão liga a refrigeração	
A	F1.5	Flag de ciclo manual	
A	I3.7	Nível de óleo da refrigeração OK	
A	I4.2	Proteção térmica da refrigeração OK	t62
R	F1.5	Reset do flag de ciclo manual	
S	Q0.3	Liga a refrigeração	
S	F6.6	Flag de refrigeração ativada	
A	F1.6	Flag de ciclo automático	
A	I3.7	Nível de óleo da refrigeração OK	
A	I4.2	Proteção térmica da refrigeração OK	
A	M08	Liga a refrigeração pelo programa CN	
A	I2.1	Porta fechada	t63
R	F1.6	Reset do flag de ciclo automático	
S	Q0.3	Liga a refrigeração	
S	F6.6	Flag de refrigeração ativada	
A	F6.6	Flag de refrigeração ativada	
AN	I2.2	Não há a confirmação da refrigeração ligada	
R	F6.6	Reset do flag de refrigeração ativada	
S	F10.1	Flag de falha no acionamento da refrigeração	
A	F6.6	Flag de refrigeração ativada	
A	I2.2	Há confirmação da refrigeração ligada	
R	F6.6	Reset do flag de refrigeração ativada	
S	F10.2	Flag de acionamento da refrigeração OK	
A	F10.2	Flag de acionamento da refrigeração OK	
A	F6.7	Flag de refrigeração desligada	
R	F10.2	Reset do flag de acionamento da refrigeração OK	
R	F6.7	Reset do flag de refrigeração desligada	
S	F7.0	Flag de refrigeração ligada	
A	I1.6	Botão desliga a refrigeração	
R	Q0.3	Desliga a refrigeração	
S	F10.3	Flag de refrigeração desligada	
A	M09	Desliga a refrigeração pelo programa CN	
R	Q0.3	Desliga a refrigeração	
S	F10.3	Flag de refrigeração desligada	
A	F0.1	Sistema desligado	

R	Q0.3	Desliga a refrigeração
S	F10.3	Flag de refrigeração desligada
A	F10.3	Flag de desliga a refrigeração ativado
A	I2.2	Não há a confirmação da refrigeração desligada
R	F10.3	Reset do flag de refrigeração ativada
S	F10.5	Flag de falha no desligamento da refrigeração
A	F10.3	Flag de refrigeração desligada
AN	I2.2	Há confirmação da refrigeração desligada
R	F10.3	Reset do flag de refrigeração desligada
S	F10.4	Flag de desligamento da refrigeração OK
A	F10.4	Flag de desligamento da refrigeração OK
A	F7.0	Flag de refrigeração ligada
R	F10.4	Reset do flag de desligamento da refrigeração OK
R	F7.0	Reset do flag de refrigeração ligada
S	F6.7	Flag de refrigeração desligada

O modelo foi implementado e simulado no editor Petri Net Tools, sendo em seguida realizada a sua conversão para a linguagem Ladder e implementada em um emulador de um Torno CNC utilizando um equipamento CNC da GE FANUC.

Abaixo é ilustrado o sistema montado, onde as fotos da Figura B.1 ilustra a bancada completa montada com o CLP da Klockner Müller em destaque a esquerda e a placa de entradas e saídas do CLP da GE FANUC.

O CLP da Klockner Müller foi utilizado para o desenvolvimento dos testes realizados nos relatórios anteriores do intertravamento convertido do modelo em rede de Petri para uma linguagem de CLP, os modelos foram convertidos de forma manual e foram validados a modelagem dos elementos básicos de controle do CLP. A programação deste CLP é realizada no microcomputador indicado na figura.

A Figura B.2 ilustra em detalhe a placa de entradas e saídas do CLP coligado com o CNC GE Fanuc. O CNC utilizado foi o NC board, que é uma placa compatível com o barramento do Micro PC, estando esta conectada no micro em destaque da Figura B.1.

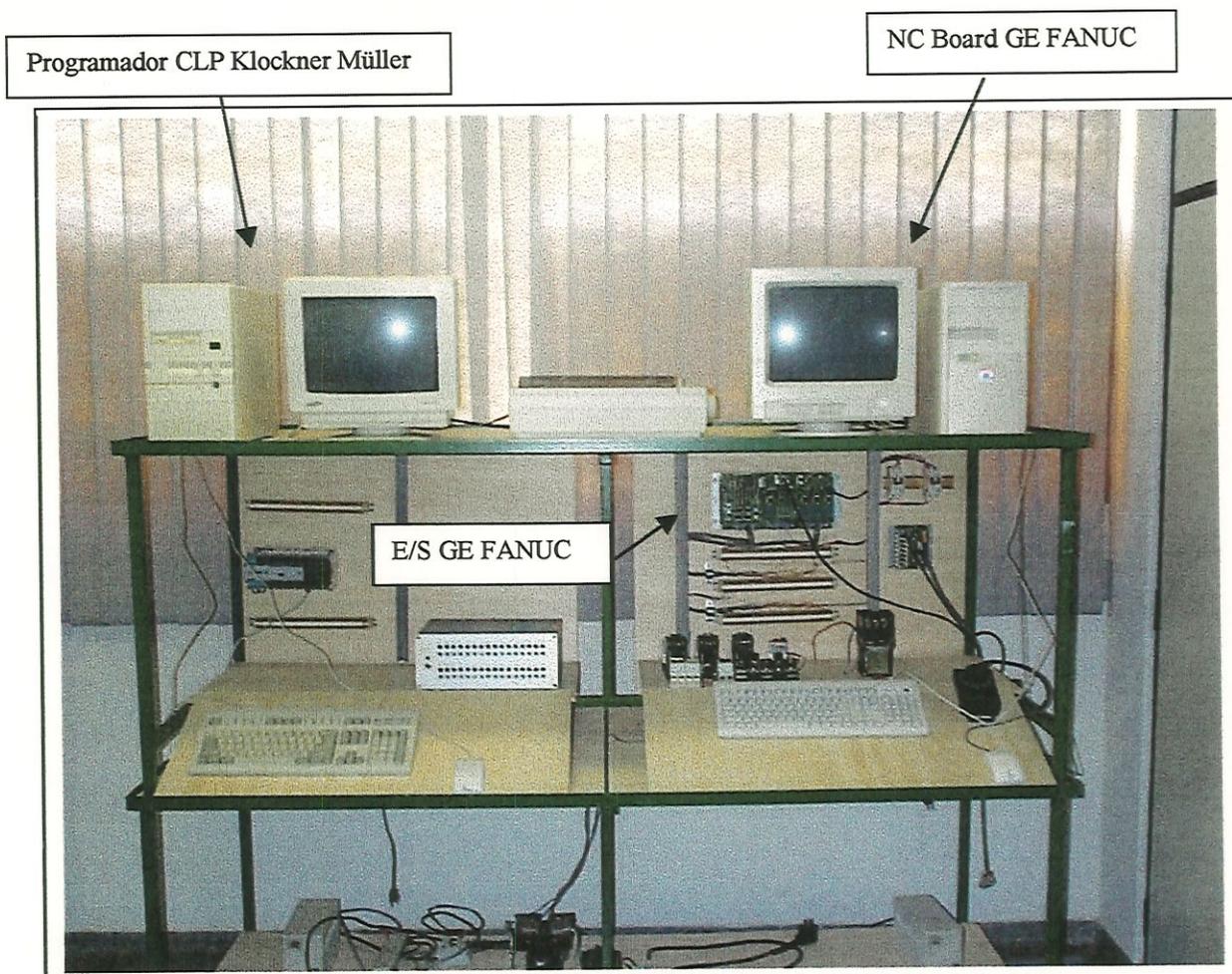


Figura B.1 – Bancada de emulação do Torno CNC e teste de programas CLP

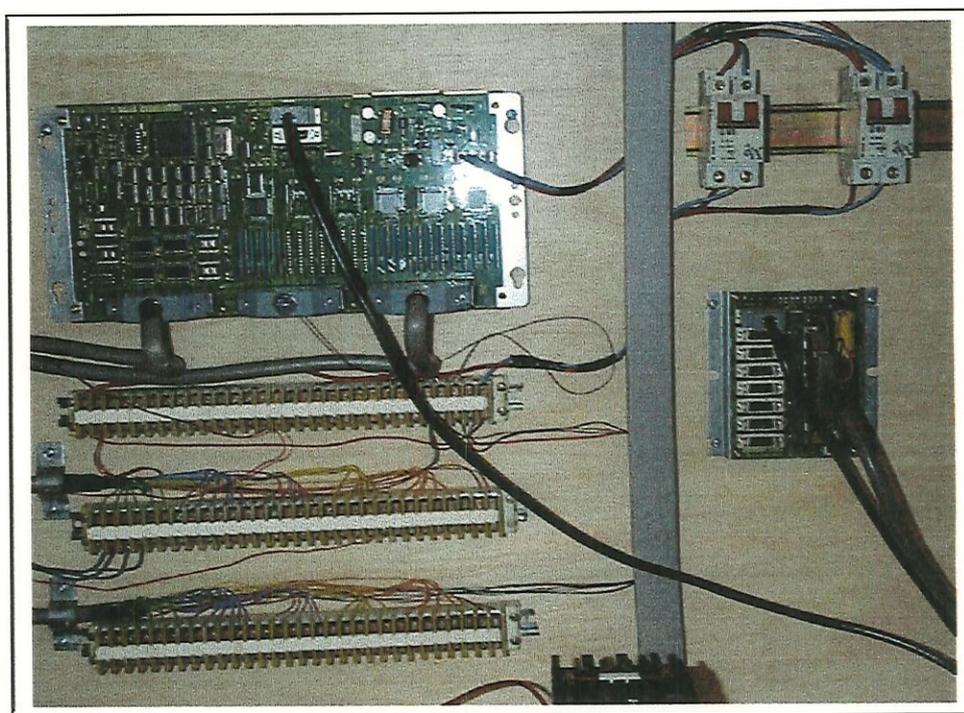


Figura B.2 – Detalhe da placa de Entradas/Saídas (I/O) do Torno CNC.