

Sys 1996463

9.1 (more & um)



DANIEL MENDES LOPES

YOSHIKI DATE

PROJETO DE SISTEMA DE CONTROLE UTILIZANDO A METODOLOGIA PFS/MFG

Trabalho de Formatura
apresentado à Escola
Politécnica da Universidade de
São Paulo para a obtenção de
Graduação em Engenharia

Orient: Diolino José dos Santos

São Paulo
2000

DANIEL MENDES LOPES

YOSHIAKI DATE

PROJETO DE SISTEMA DE CONTROLE
UTILIZANDO A
METODOLOGIA PFS/MFG

São Paulo
2000

SUMÁRIO

1.	Objetivos	6
2.	Introdução Teórica	7
2.1.	Sistemas de Eventos Discretos - SED	7
2.2.	Redes de Petri	9
2.3.	Production Flow Schema - PFS	12
2.4.	Mark Flow Graph - MFG	13
2.5.	Metodologia PFS/MFG	15
3.	Metodologia de Projeto de Sistemas de Controle	17
3.1.	Análise de Necessidades	18
3.1.1.	Identificação do objetivo final do sistema	19
3.1.2.	Estudo do objetivo de controle, equipamento e instalações	20
3.1.3.	Organização dos conhecimentos sobre dispositivos e instalação	20
3.1.4.	Levantamento e análise das funções de controle	21
3.2.	Definição das necessidades	21
3.2.1.	Definição das funções de controle	22
3.2.2.	Definição do fluxo das funções de controle	23
3.3.	Projeto do sistema de controle	24

3.4.	Projeto do software de controle	24
3.5.	Desenvolvimento do software de controle e testes	25
4.	Controle de Recursos	26
4.1.	Hierarquia de Controle	26
4.2.	Modelagem da Utilização de Recursos	28
4.3.	Controle da Utilização dos Recursos	35
4.3.1.	Modelo Complementar	35
4.3.2.	Definição das Regras de Controle	37
4.4.	Observações Complementares	44
5.	Sistema Mixer	46
5.1.	Introdução	46
5.2.	Overview do sistema Mixer	46
5.3.	Descrição de cada um dos processos	49
5.3.1.	Processo de inicialização	49
5.3.2.	Processo de setup da máquina	49
5.3.3.	Processo de inserção dos componentes que formam a borracha	50
5.3.4.	Processo de mistura.	51
5.3.5.	Processo de descarregamento da borracha	52
6.	Aplicação da Metodologia no Sistema Mixer	53

6.1.	Análise das necessidades	53
6.1.1.	Identificação do objetivo final do sistema	53
6.1.2.	Estudo do objeto de controle, equipamentos e instalação	54
6.1.3.	Levantamento e análise das funções de controle.	58
7.	Diagramas PFS/MFG do sistema Mixer	59
7.1.	Diagramas traduzidos	59
7.2.	Diagramas aperfeiçoados	65
8.	Conclusões	71
	Bibliografia	72

LISTA DE ABREVIATURAS

CLP - Controlador Lógico Programável

IEC - International Eletrotechnical Committee

MFG - Mark Flow Graph

PFS - Production Flow Schema

SED - Sistema a Eventos Discretos

SFC - Sequential Flow Chart

GAR – Grafos de Alocação de Recursos

R E S U M O

Este trabalho tem como objetivo principal mostrar a aplicação da metodologia PFS/MFG na elaboração do sistema de controle de um processo industrial.

O processo estudado é o da confecção de borracha a partir de borracha bruta e aditivos. O sistema deve ser capaz de armazenar diversas receitas diferentes de borracha em um banco de dados. Estas serão processadas automaticamente, com os materiais e proporções pré-definidas.

O estudo partiu de um Diagrama de Transição de Estados confeccionado segundo a metodologia tradicional. Aplicou-se a técnica de conversão de autômatos finitos para infinitos para a determinação do grafo correspondente. Uma análise criteriosa foi realizada para sua validação, onde foram encontradas inconsistências. Foram propostas alterações para corrigi-las e aperfeiçoar o sistema.

O trabalho ressalta as vantagens em se trabalhar com a Metodologia PFS/MFG, com a utilização de grafos com autômatos infinitos, no lugar da Metodologia Tradicional, que utiliza Diagramas de Transição de Estados.

1. Objetivos

Este trabalho tem como objetivo principal mostrar a aplicação da metodologia PFS/MFG na elaboração do sistema de controle de um processo industrial. Será realizada a comparação do Diagrama MFG com o Diagrama de Transição de Estados que é normalmente utilizado como metodologia.

2.Introdução Teórica

Será apresentada, a seguir, uma introdução sobre a Metodologia PFS/MFG e outros tópicos de teoria que foram utilizados neste trabalho.

2.1. Sistemas de Eventos Discretos - SED

Os Sistemas de Eventos Discretos caracterizam-se por apresentar transições entre estados definidos, tanto de forma paralela quanto simultânea.

Segundo [Miyagi, 1996], na tecnologia do controle de SED tradicional, o Controle Seqüencial, os procedimentos específicos de cada planta eram implementados através de um circuito elétrico de relés. Porém, a verificação da evolução do procedimento de controle e a identificação de falhas do sistema é difícil de ser realizada além de seu projeto e construção serem específicos a cada instalação. Outra desvantagem na sua utilização é que a dinâmica de controle fica camuflada no diagrama, não sendo possível uma interpretação e dificultando sua manutenção. Também não é possível a estruturação do processo de controle, não permitindo um desenvolvimento top-down do projeto.

No controle de SED baseado em computadores, o Controle Programável, a estratégia é dada pela execução de um programa que define a evolução dos processos. Deste modo, aumentamos sua maleabilidade pois qualquer alteração dos procedimentos pode ser facilmente realizada via

software. Enquanto isso, o hardware torna-se menos específico podendo ser produzido em escala reduzindo-se, assim, seu custo.

As linguagens utilizadas na programação de controladores, definidas pelo IEC (International Electrotechnical Committee) são divididas em:

Tabela 2.1.1. Linguagens de programação




Textuais	Álgebra de Boole Lista de instrução Texto estruturado
Gráficas	Diagrama de relés Diagrama de bloco de função Fluxograma Elementos SFC
Tabulares	Tabela de decisão

2.2. Redes de Petri

As Redes de Petri são consideradas linguagens de programação gráficas. A do tipo Condição/Evento são utilizadas para modelar Sistemas de Eventos Discretos.

Elas são a base para as linguagens interpretadas Mark Flow Graph (MFG) e Sequential Flow Chart (SFC).

Tabela 2.2.1. Elementos formadores de Redes de Petri

Condições: elemento passivo, estados	
Eventos: elemento ativo, ações	
Arcos orientados: fluxo do sistema	

A Condição, quando satisfeita, receberá uma marca como pode ser visto na Figura 2.2.1.

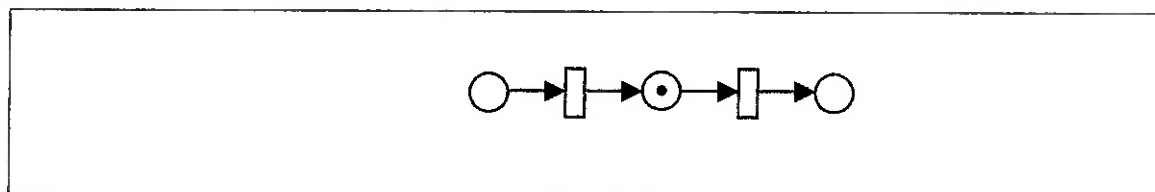


Figura 2.2.1. Condição satisfeita

Um Evento ocorre se todas as pré-condições estão satisfeitas e todas as pós-condições não estão. Quando este evento ocorrer, todas as pré-condições

deixam de ser satisfeitas e todas as pós-condições tornam-se satisfeitas, conforme a Figura 2.2.2.

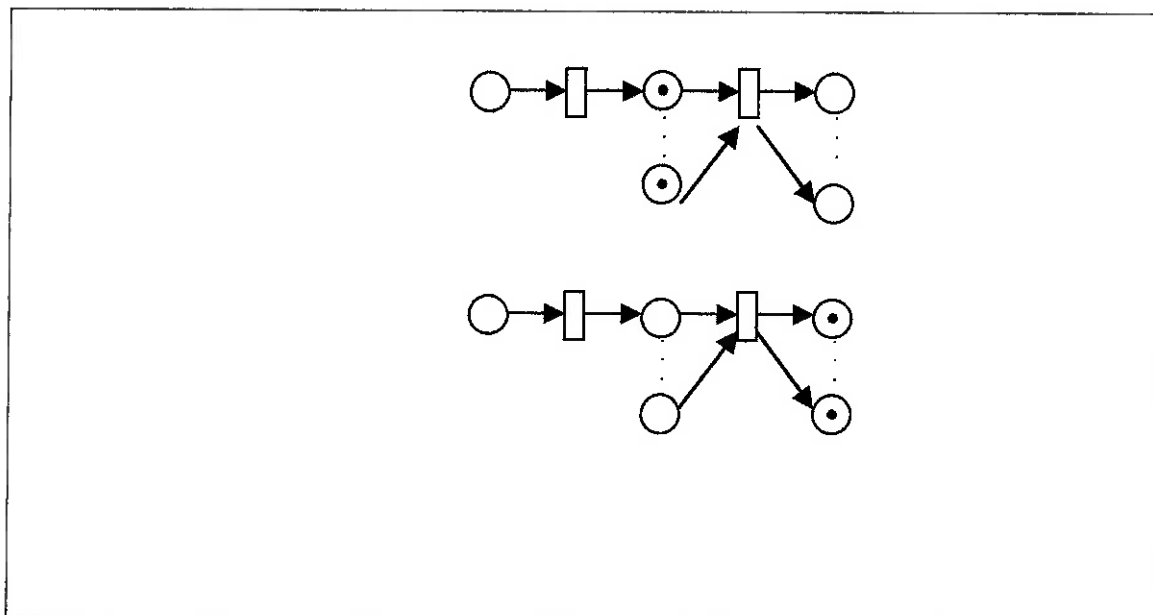


Figura 2.2.2. Ocorrência de um evento

Quando um Evento possui todas as pré-condições satisfeitas mas é impedido de ocorrer devido a um pós-condição satisfeita tem-se um Contato, como pode ser visto na Figura 2.2.3. É desejável que a rede seja livre de Contato, isto é, não apresente a possibilidade de uma pós-condição impedir um Evento, pois, dessa forma, apenas as pré-condições controlam os Eventos. Isso é desejável para facilitar o controle da planta. Para livrar uma rede de Contatos utilizam-se Complementos, como pode ser visto na Figura 2.2.4

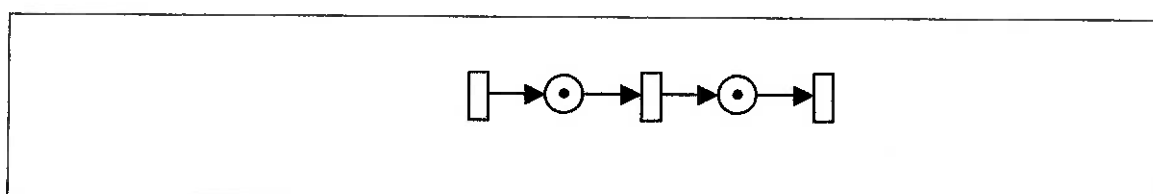


Figura 2.2.3. Contato em uma rede

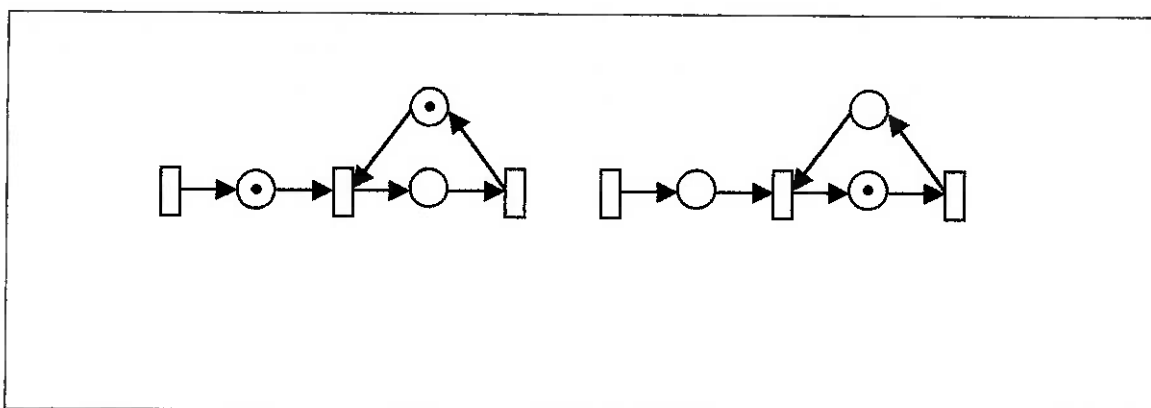


Figura 2.2.4. Complemento para livrar rede de Contato

Outro fato que deve ser evitado nestas redes é a presença de Conflitos. Dois eventos estão em Conflito entre si se ambos estão ativados e a ocorrência de um resulta na desativação de outro, como pode ser visto na Figura 2.2.5. Neste caso deve ser feita uma escolha, mas este tipo de rede não apresenta um comportamento determinístico frente a tal situação.

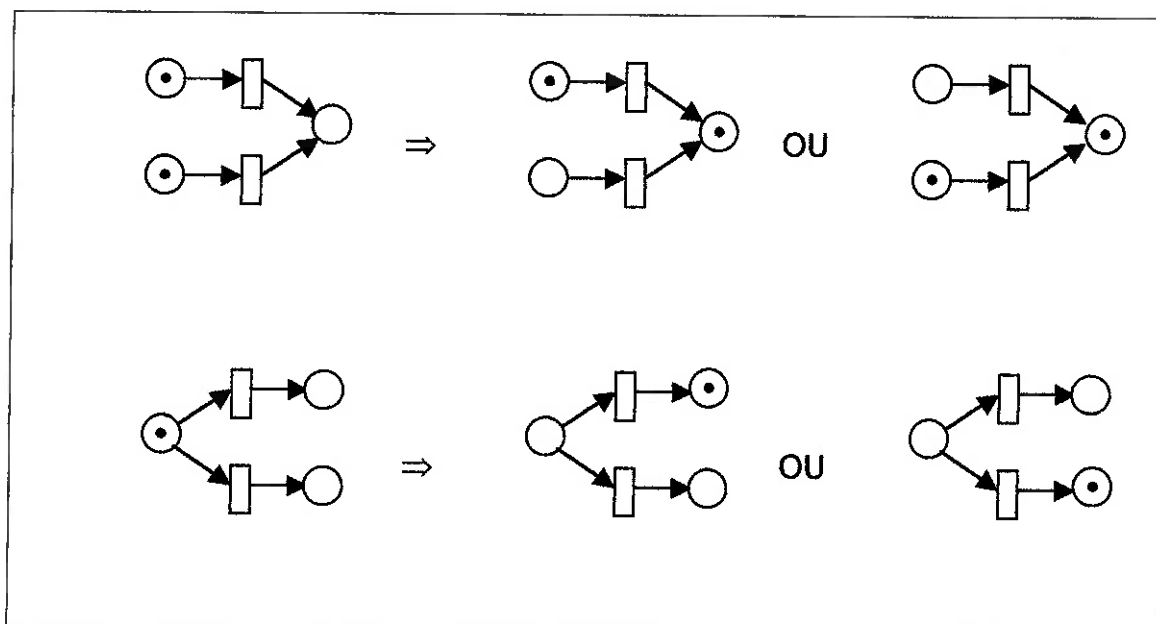


Figura 2.2.5. Tipos de Conflito

2.3. Production Flow Schema - PFS

Em SED, um evento pode incluir vários outros eventos e estados em um nível hierárquico inferior. Portanto, pode-se construir um modelo genérico, alto nível, e detalhá-lo desenvolvendo-se os eventos. Esta abordagem top-down é muito eficiente no desenvolvimento das estratégias de controle de SED.

A técnica PFS possibilita a modelagem por redes de eventos e suas relações em um nível mais alto, que pode ser detalhada em outras redes PFS, até alcançarmos um nível de comandos e transformá-la em uma rede MFG. O refinamento de uma rede PFS pode ser visto na Figura 2.3.1.

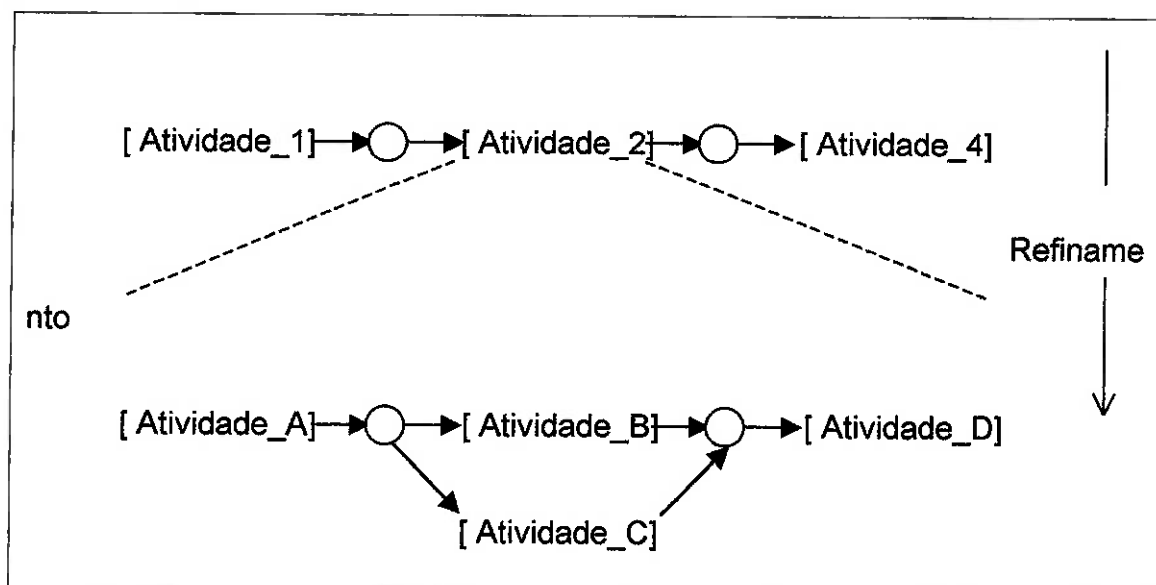




Figura 2.3.1. Exemplo de rede PFS e seu Refinamento

O PFS é formado por elementos ativos e passivos e a relação entre estes componentes. Porém, não possui marcas, isto é, não apresenta dinâmica. Os elementos PFS podem ser vistos na Tabela 2.3.1.







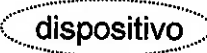


Tabela 2.3.1. Elementos formadores do PFS

Distribuidor: elemento passivo	
Atividade: elemento ativo	[Atividade]
Arcos orientados: relações do sistema	

2.4. Mark Flow Graph - MFG

A linguagem MFG é derivada de Redes de Petri com a inclusão de interpretação, isto é, pode enviar sinais de estado do modelo para dispositivos externos, e receber sinais gerados por dispositivos externos para controlar a ocorrência de eventos do modelo.

Tabela 2.4.1. Elementos formadores de MFG

Box: elemento passivo, condição	
Transição: elemento ativo, evento	
Arcos orientados: relações do sistema	
Porta inibidora: interna ou externa	
Porta habilitadora: interna ou externa	
Arco de sinal de saída	
Dispositivos externos	
Box temporizado	
Transição temporizada	

O Box Temporizado torna a transição disparável após o intervalo de tempo τ , enquanto que a Transição Temporizada dispara após ter ocorrido o intervalo de tempo τ quando todas as condições estiverem satisfeitas.

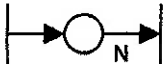


Da mesma forma que nas Redes de Petri, o disparo de uma transição ocorre se todas as pré-condições estão satisfeitas, todas as pós-condições não estão, todas as portas habilitadoras estão satisfeitas e todas as portas inibidoras não estão satisfeitas. Quando este evento ocorrer, todas as pré-

condições deixam de ser satisfeitas e todas as pós-condições tornam-se satisfeitas e as condições referentes às portas não são alteradas.

Os conflitos, que eram resolvidos de maneira não determinística nas Redes de Petri, no MFG são controlados pelas portas habilitadoras e inibidoras.

O MFG possibilita a criação de blocos macros que representam uma seqüência de atividades. Dentre alguns blocos macros, os mais utilizados estão na Tabela 2.4.2.

Tabela 2.4.2. Blocos Macro de MFG

Box capacidade:	
Box agrupador:	
Box dispensor:	

2.5. Metodologia PFS/MFG

Esta metodologia é baseada num refinamento gradativo do PFS, modelo conceitual, substituindo os elementos atividade ou distribuidor por uma rede PFS ou MFG até que se tenha uma rede MFG, modelo detalhado.

Uma atenção especial deve ser dada às condições de contorno, que devem ser respeitadas para a compatibilidade do modelo.

A Metodologia PFS/MFG dá-se pelos seguintes passos:

- 1- Identificação dos principais fluxos de atividades;
- 2- Detalhamento dos fluxos;
- 3- Detalhamento das atividades;
- 4- Introdução dos elementos de controle de recursos; e
- 5- Indicação dos sinais de controle com as plantas.

A Metodologia PFS/MFG prevê a conversão automática e simples do MFG para a descrição em SFC, linguagem padronizada pelo IEC, dos procedimentos de controle no nível dos programas de Controladores Programáveis.

3. Metodologia de Projeto de Sistemas de Controle

Em MIYAGI [1996] e SANTOS FILHO [1998] descreve-se uma metodologia de projeto de sistemas de controle baseada em procedimentos clássicos de desenvolvimento de software. Neste, considera-se o conceito de ciclo de vida do sistema de controle, que pode ser visto na Figura 3.0.1.

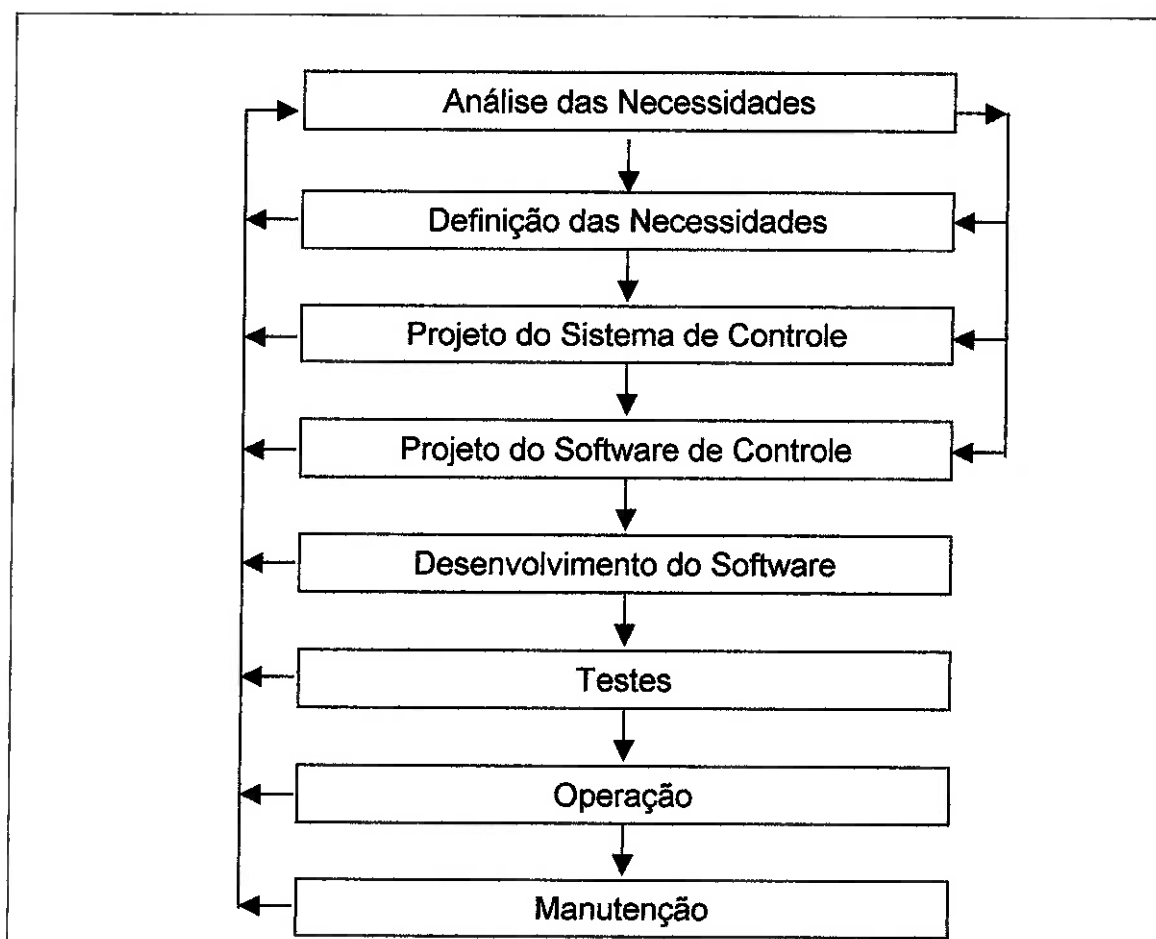


Figura 3.0.1 Ciclo de vida do sistema de controle

Em cada uma das etapas de desenvolvimento, realiza-se uma série de procedimentos que são repetidos ciclicamente a fim de se obter um refinamento gradativo. Este é o conceito de prototipagem, ilustrado na Figura 3.0.2.

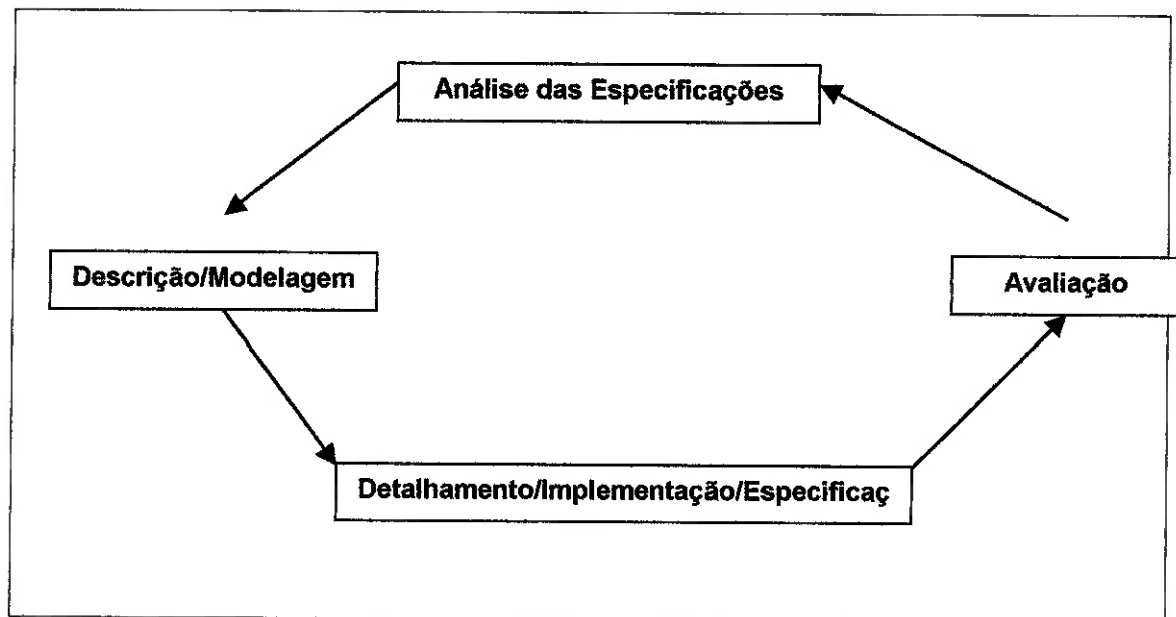


Figura 3.0.2 Ciclo de prototipagem

3.1. Análise de Necessidades

A análise das necessidades, como diz o próprio nome, consiste em definir as necessidades principais do projeto, que muitas vezes possui exigências ambíguas. Ela possui basicamente as seguintes características.

- Necessidade de organizar as relações humanas.
- Necessidade de amplo conhecimento e know-how.

- Reconsideração dos conceitos.
- Aplicação contínua do conceito de prototipagem.
- Final indeterminado.

Como pode se notar, a análise de necessidades exige grande conhecimento, sendo o primeiro passo na definição do ciclo de vida do sistema de controle e devendo, portanto ser realizada cuidadosamente. Porém, não deve-se exagerar no detalhamento, o que será realizado nas etapas posteriores.

3.1.1. Identificação do objetivo final do sistema

A identificação do objetivo final do sistema pode se dar através de diferentes abordagens:

- Especificação do sistema.
- Recursos humanos.
- Determinado domínio (alcance, range) da produção.
- Determinado orçamento de desenvolvimento.
- Determinado cronograma de desenvolvimento e implantação.

3.1.2. Estudo do objetivo de controle, equipamento e instalações

Para compreender o objetivo de controle, é necessário estudar as funções e as características de cada elemento, assim como identificar claramente as inter-relações entre estes elementos. Para esta atividade, os seguintes documentos devem ser elaborados :

- Diagrama estrutural (esquemático) do objeto de controle.
- Lista preliminar dos atuadores.
- Lista preliminar dos detectores.
- Lista preliminar de intertravamentos.
- Diagrama da infra-estrutura necessária.
- Modelagem funcional de cada dispositivo utilizando grafo PFS/MFG.

3.1.3. Organização dos conhecimentos sobre dispositivos e instalação

Nesta etapa levanta se as informações que o dispositivo de realização do controle necessita considerar em relação aos dispositivos de acionamento de motores, dispositivos de comando, dispositivos para monitoração e emissão de relatórios para efetiva realização do controle, além de dados sobre a técnica

de programação, capacidade da memória de dados, características funcionais de entradas e saída, critérios para classificação e instalação da fonte de alimentação dos acionamentos e do controle, etc.

3.1.4. Levantamento e análise das funções de controle

Nesta atividade identifica-se o que o usuário, operador ou cliente deseja concretamente executar para atingir os objetivos finais, assim como as interfaces e intervenções necessárias do operador.

Para isso utilizaremos diagramas que sistematizam o inter-relacionamento entre as funções através de uma estrutura hierárquica. Neste diagrama, no Nível 0 temos os objetivos desejados e, nos níveis posteriores detalhamentos progressivos das funções para alcançá-los.

3.2. Definição das necessidades

A tarefa de definição das necessidades consiste em analisar a especificação dos requisitos e gerar como resultado uma definição das necessidades. Para alcançar o objetivo final do sistema, uma análise em três partes : análise do objeto de controle, análise do dispositivo de controle e análise das funções.

Os documentos gerados nesta etapa são as especificações do sistema de controle que são utilizados para os seguintes fins:

- Obter a confirmação e aprovação do usuário, operador e cliente em relação às funções do sistema.
- Identificar as especificações para o projeto do sistema de controle e seu software.
- Gerar uma referência para a avaliação da qualidade na etapa de testes.
- Gerar uma ordem para o fornecimento do hardware.
- Gerar uma parte do manual de operações e de manutenção.

3.2.1. Definição das funções de controle

Consiste basicamente em definir as seguintes funções inerentes ao dispositivo de controle :

- Funções de inicialização da operação
- Função de seleção do modo de operação
- Função de seleção do local de operação
- Funções de sinalização/indicação

- Funções de comando de operação
- Funções de medição
- Função de sinalização de falha ou alarme

Na etapa são gerados os seguintes documentos na forma de especificação definitiva:

- Diagrama das funções de controle
- Lista dos dispositivos de atuação
- Lista dos dispositivos de detecção
- Lista dos dispositivos de comando e monitoração
- Lista dos intertravamento: intertravamento entre funções ou entre elementos do objeto de controle, condição de partida ,condição de operação, etc.
- Diagrama do sistema de alimentação de energia

3.2.2. Definição do fluxo das funções de controle

Para a realização das operações especificadas, deve se definir os procedimentos que ativam as várias funções de controle anteriormente definidas, isto é, deve se definir o fluxo das funções de controle.

A metodologia utilizada para se efetuar essa tarefa foi a implementação de PFS e MFG.

Nesta etapa são gerados os seguintes documentos:

- Fluxograma das funções de controle
- Fluxograma de chaveamento para as operações manuais e situações anormais
- Proposta de tratamento de exceções

3.3. Projeto do sistema de controle

As etapas referentes ao projeto do sistema de controle são:

- Definição das interfaces e alocações das funções
- Definição e alocação dos sinais de entrada e saída
- Definição da estrutura de programa

3.4. Projeto do software de controle

O projeto do software de controle consiste em implementar no dispositivo de realização do controle a lógica das funções de controle descritas nos documentos gerados nas etapas de definição de necessidades e de projeto do sistema de controle

3.5. Desenvolvimento do software de controle e testes

O software que será carregado no CLP, o programa objeto, pode ser obtido através do programa-fonte (que pode ser diagrama de relés, PFS/MFG) automaticamente ou codificando-o.

Os testes são realizados executando-se o programa de controle no dispositivo de controle e verificando-se os resultados. Esta etapa é fundamental para detectar-se falhas que podem ser corrigidas sem maiores transtornos. Atualmente, a verificação automática de erros tem aumentado a qualidade dos softwares também otimizando este processo de testes.

4. Controle de Recursos

4.1. Hierarquia de Controle

Antes de iniciar-se a discussão a respeito da modelagem do controle de recursos em si, é importante apresentar a arquitetura de controle global, envolvendo os níveis de controle a serem tratados para facilitar desde início a compreensão do contexto de cada nível de controle. A Figura 4.1 descreve a estrutura básica desse sistema.

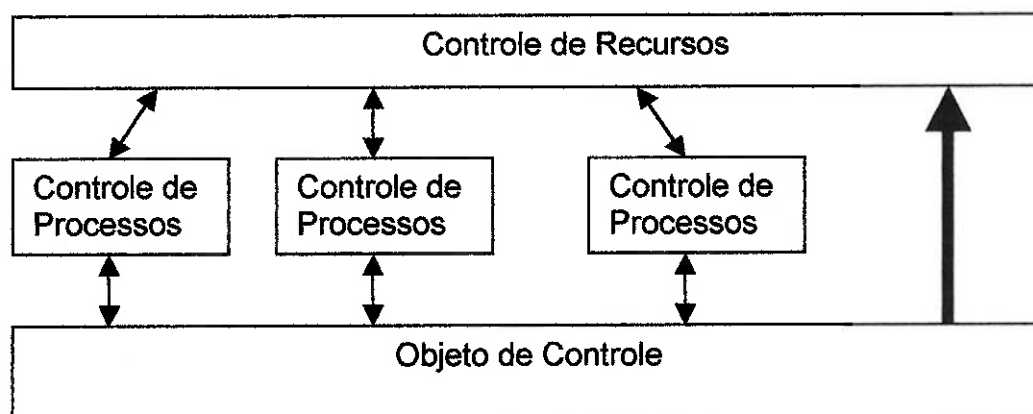


Figura 4.0.1 Arquitetura do sistema de controle de SPs

O paradigma adotado nesta arquitetura baseia-se no fato da modelagem do nível de controle de recursos utilizar como referencial os próprios recursos para iniciar o processo de modelagem em questão, diferenciando-se do contexto de modelagem orientada a processos utilizada no nível inferior da arquitetura ilustrada na Figura 4.0.1.

O objetivo passa a ser modelar como os recursos devem se alocados durante a evolução dinâmica de cada processo e agregar estratégias de controle que sejam necessárias para garantir o comportamento desejado para o sistema global.

Portanto, dependendo da lógica necessária para controlar a utilização destes recursos é possível que o processo de controle dos mesmos seja complexo.

Há ainda duas características importantes no modelo da Figura 4.0.1 que devem ser destacadas:

- A possibilidade do operador interagir com os níveis de controle
- A possibilidade do controle de recursos monitorar diretamente informações do objeto de controle

A primeira refere-se à possibilidade do operador interagir com o processo de controle fazendo parte do sistema de controle em si, de acordo com os conceitos de Sistemas Antropocêntricos de Produção.

A segunda observação refere-se à possibilidade do sistema de controle de recursos ter a capacidade de receber informações diretamente do objeto de controle, independentemente dos controladores locais, para que possa identificar o estado do objeto de controle em eventuais situações em que este procedimento seja necessário.

Nos itens seguintes é descrito o procedimento de desenvolvimento do modelo global de uso dos recursos e então, o procedimento de obtenção do modelo de controle baseado nas regras adicionais de controle que detectam o estado anterior ao 'deadlock'.

4.2. Modelagem da Utilização de Recursos

Dado que a referência para se executar a modelagem do sistema de controle de recursos são os próprios recursos, a proposta é utilizar-se como ponto de partida a modelagem a partir de Grafos de Alocação de Recursos – GAR.

Um grafo do tipo GAR é constituído por nós e arcos orientados. Um par contendo um nó e o respectivo arco orientado de saída é denominado **par de alocação** enquanto um par contendo um nó e o respectivo arco de entrada é denominado **par de aquisição**.

Em um **par de alocação** o nó representa um determinado recurso r_q e o arco orientado de saída possui uma inscrição fixa que representa a etapa do processo $s_q(j)$ que aloca r_q para ser executada. Por sua vez, em um **par de requisição** o nó também representa um determinado recurso $r_{q'}$ e o arco orientado de entrada possui uma inscrição fixa que representa a etapa do processo $s_q(i)$ que requisita $r_{q'}$ para ser utilizado na próxima etapa do processo.

A figura 5.2. ilustra graficamente a representação destes pares para o caso de um mesmo produto em uma determinada etapa de sua seqüência de produção.



Figura 4.2.1 Elementos fundamentais de um GAR

Em princípio é possível considerar que os processos são modelados a partir de uma seqüência de recursos utilizados para realizar a seqüência de operações que compõem o processo em questão. Seja Q o conjunto de n elementos diferentes a serem processados em um sistema produtivo, de acordo com a expressão 5.1.

$$Q = \{q \in N \mid q \leq n\} \quad (4.1)$$

Onde:

Q identifica o elemento

N é o conjunto dos números naturais

Formalmente, existe uma seqüência de operações associadas a cada elemento q a qual, por sua vez, agrega uma seqüência de recursos necessários. Neste contexto, é possível dividir a seqüência de operações em etapas, de maneira que a cada uma destas aloca-se um recurso específico.

Como resultado, obtém-se uma seqüência de produção p_q do elemento q , de acordo com a expressão 4.2.

$$p_q = \{s_q(0), s_q(1), \dots, s_q(L_q), s_q(L_q + 1)\} \quad (4.2)$$

Onde:

L_q representa o comprimento da seqüência de produção do elemento q .

$s_q(i)$ representa a i -ésima etapa da seqüência de produção do elemento q .

$s_q(0)$ representa a etapa inicial da seqüência de produção do elemento q .

$s_q(L_q + 1)$ representa a etapa final correspondente ao estado em que a seqüência de produção do elemento q está concluída.

A partir destas informações, define-se a seqüência de recursos r_q^* correspondente à seqüência de produção p_q do elemento q , de acordo com a expressão.

$$r_q^* = \{r_q(1), r_q(2), \dots, r_q(L_q)\} \quad (4.3)$$

Onde:

r_q^* representa o recurso utilizado na etapa $s_q(i)$ da seqüência de produção do elemento q .

A partir das informações destas três expressões, sistematiza-se a geração GAR correspondente da seguinte maneira:

- a) Através da expressão (4.1) seleciona-se um elemento.
- b) Relaciona-se o conjunto de recursos associados a este elemento através da expressão (4.3). Estes recursos são modelados através de nós no GAR.
- c) Modelam-se as etapas de processo do elemento selecionado em (4.1) de acordo com as informações obtidas pela expressão (4.2).

Uma vez que o GAR é um grafo transição que representa um autômato finito, é possível mapeá-lo em uma rede de Petri.

Definição 4.1. Considere um autômato de estados finitos dados pela quintupla (E, X, f, x_0, F) onde:

E é um alfabeto finito;

X é um conjunto de estados finito;

F é uma função de transição de estados $f: X \times E \rightarrow X$

x_0 é o estado inicial tal que $x_0 \in X$;

F é um conjunto de estados tal que $F \subseteq E$.

Definição 4.2. Considere o grafo Mark Flow Graph bipartido, representado por uma sêxtupla (B, T, A, G_I, G_E, S) onde:

$B = \{B_1, \dots, B_i\}$ com $(i \geq 1)$ é um conjunto de boxes;

$T = \{T_1, \dots, T_j\}$ com $(j \geq 1)$ é um conjunto de transições;

A é o conjunto de arcos orientado;

$G_I = \{G_{I1}, \dots, G_{Im}\}$ com $(m \geq 1)$ é um conjunto de portas internas;

$G_E = \{G_{E1}, \dots, G_{En}\}$ com $(n \geq 1)$ é um conjunto de portas externas;

$S = \{S_1, \dots, S_p\}$ com $(p \geq 1)$ é um conjunto de arcos de sinais de saída;

De acordo com CASSANDRAS[1993] dado um autômato de estados finitos (E, X, f, x_0, F) , é possível construir-se uma rede de Petri elementar (REISIG[1985]), uma vez que se pode associar a cada estado global representado pelo autômato uma determinada marcação na rede de Petri correspondente. Considerando que o MFG baseia-se em redes de Petri com a ressalva de ser um grafo interpretado que possui um mapeamento dos sinais externos através de arcos de sinal de saída e portas externas, baseado em CASSANDRAS [1993] é possível inferir que a construção de um modelo MFG a partir de um GAR implica no seguinte procedimento:

- Cada estado em X é definido como um Box em B .
- A cada par de estados (x, x') , onde $x' = f(x, e)$ para $e \in E$, define-se uma transição em T tal que $T = \{(x, x') : x \in X, x' = f(x, e) \text{ para qq. } e \in E\}$
- A seguir, define-se um arco a partir de cada par (x, t) e (t, x') com $x \in X$ e $x' \in X$ e $t \in T$ tal que $A = \{(x, t) : x \in X, t \in T\} \cup \{(t, x') : x' \in X, t \in T\}$

A partir deste procedimento obtemos o modelo MFG correspondente ao GAR.

Uma extensão importante corresponde à generalização da representação de cada recurso por um Box simples em E-MFG para a situação em que existem múltiplos recursos disponíveis, sendo adequado, portanto, utilizar-se o Box capacidade para essa finalidade.

Considerando que em um sistema produtivo pode haver um conjunto de processos ocorrendo simultaneamente, gera-se o GAR correspondente a cada elemento que será processado e, posteriormente, gera-se os grafos E-MFG de utilização dos recursos para cada processo.

Uma vez que pode existir compartilhamento de recursos entre processos, os boxes equivalentes que representam estes recursos em E-MFG estarão representados em diferentes grafos. Para obter-se um modelo global de alocação de recursos considerando os vários processos, é necessário aplicar a técnica de fusão de lugares (BARROS; PERKUSICH [1998]) nos vários grafos E-MFG correspondentes aos modelos obtidos para os vários elementos com seus processos associados.

A seguir, sintetiza-se os procedimentos que devem ser realizados para obter o modelo E-MFG global de utilização de recursos de um SP:

- a) Gerar o GAR associado a cada elemento a ser processado.
- b) Construir o grafo marcado de utilização dos recursos E-MFG equivalente a cada GAR.

- c) Aplicar a técnica de fusão de lugares para obter o grafo E-MFG de utilização dos recursos no nível global do sistema.

Há algumas observações que devem ser realizadas antes de prosseguir-se para a etapa seguinte de análise e controle do modelo de utilização de recursos.

- a) A utilização de cada recurso $r_q(j)$ é modelado por um Box capacidade $Cr_q(j)$ para permitir a representação do número de unidades deste recurso disponível no sistema.
- b) A marcação inicial é tal que não considera nenhuma etapa de processo em andamento no estado inicial, correspondendo a uma situação em que o sistema produtivo encontra-se completamente ocioso.

A figura 4.2.1 ilustra um exemplo de modelo E-MFG genérico correspondente à seqüência de produção de um elemento q e, portanto, a correspondente seqüência de recursos obtidos a partir do procedimento descrito anteriormente. Neste modelo, os elementos $C_q(0)$ e $C_q(L_q+1)$ são representados através de boxes do tipo capacidade para representar efetivamente a demanda por este processo, possibilitando o armazenamento ou acomodação temporária de elementos aguardando o processamento e de elementos prontos, respectivamente.

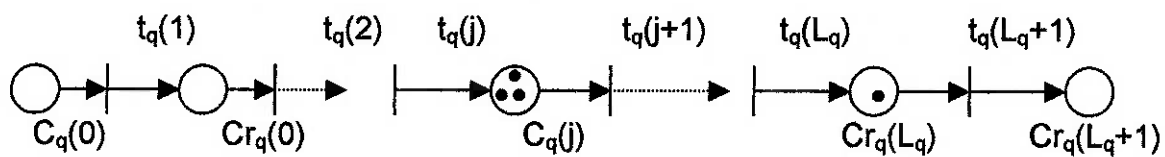


Figura 4.2.1 Modelo de uma sequência de recursos utilizados para processar q

4.3. Controle da Utilização dos Recursos

4.3.1. Modelo Complementar

O grafo E-MFG construído na seção anterior está desempenhando a função de apenas representar a utilização de recursos em cada processo e, caso haja fusão de lugares, nestes pontos há, portanto, um compartilhamento para a utilização dos correspondentes recursos.

Entretanto, um modelo que representa o compartilhamento de vários recursos e estabelece como estratégia de controle deste compartilhamento apenas a fusão de lugares, determinando um possível conflito entre processos diferentes que necessita de arbitração e qual elemento tem prioridade para ser processado, representa uma estrutura de controle inadequada para o nível de complexidade do problema.

Este problema de integrar uma série de processos diferentes para serem executado envolvendo recursos em comum determina dois graus de indeterminismo (SANTOS FILHO [1998]):

- Indeterminação em relação ao tempo, condizente com a natureza dirigida por eventos do sistema.
- Indeterminismo em relação à seqüência de ocorrência de eventos. Este seqüenciamento só está pré-definido localmente para cada processo, mas não globalmente.

Estes dois aspectos caracterizam o processo global do sistema como sendo constituído por processos locais independentes e assíncronos, de maneira que o conhecimento detalhado de cada um destes processos não é suficiente para pré-definir o conhecimento do sistema global, caracterizando, portanto, um sistema complexo no nível de controle global.

Portanto, o desafio está em como proceder para controlar efetivamente o comportamento deste sistema global. Em CONANT; ASHBY [1970] é questionado o problema de que para obter um bom controlador de um sistema é necessário que este controlador reflita o modelo do sistema em si. Em ZAMES [1981] discute-se os efeitos das incertezas de planta sobre a realimentação e propõe-se métodos de redução da sensibilidade através de aproximação inversas.

No caso dos SPs, realiza-se o Controle Qualitativo, envolvendo lógica de primeira ordem.

Em síntese, o modelo tradicional de alocação de recursos baseia-se no conceito de representar os recursos disponíveis e o controle decide sobre a possibilidade de efetuar cada alocação para realização de cada etapa de

processo em SP. Por sua vez, o modelo complementar denominado modelo de utilização de recursos baseia-se no conceito de decidir o que não pode ser alocado para a realização de determinada(s) etapa(s) de determinado(s) processo(s) em determinado(s) estados do SP.

Por exemplo, enquanto no estado inicial em que todos os recursos estão ociosos tem-se no modelo tradicional de alocação de recursos uma marcação contendo o número total de recursos disponíveis, no caso do modelo de utilização de recursos, a marcação inicial é zero ou nula.

Verificando o conceito de complementação e observado que esta é uma característica dos modelos de utilização de recursos, o próximo passo é definir as regras adicionais de controle que devem ser agregadas ao modelo em questão.

4.3.2. Definição das Regras de Controle

Já foi mencionado anteriormente que, de uma forma genérica, controlar um sistema pode ser interpretado como impor um determinado comportamento desejado para o mesmo. No caso dos sistemas complexos em que o controle que atua no mesmo é do tipo Controle Qualitativo, correspondendo portanto a um SP complexo, torna-se difícil impor um comportamento dinâmico desejado uma vez que não existe este modelo de referência pré-definido. Portanto, na maioria das situações, o que se faz é controlar para assegurar a seqüência dos processos e evitar “deadlock” do sistema.

Uma das estratégias clássicas aplicadas à validação do controle de SPs é a técnica de análise do modelo por simulação. A vantagem em utilizar-se esta técnica é que evita uma análise estrutural que pode se deparar com o problema de explosão combinatória de estados, comprometendo a obtenção de resultados positivos. A desvantagem é que a validação é sempre parcial, principalmente quando o modelo é complexo.

A) Segmentos e Ciclos Fechados em um GAR

Considerando-se um conjunto de recursos r_q^* utilizados para executar a seqüência de produção p_q do produto q , se não houver compartilhamento de recursos, o modelo GAR correspondente não possuirá ciclos fechados, uma vez que um mesmo recurso não é reutilizado durante o processamento do mesmo item.

A seqüência de conexões dos pares de alocação de uma determinada rota obedece o sequenciamento do processo correspondente à rota em questão. Este procedimento de conectar-se os pares de alocação é definido como conexão contínua (HASEGAWA [1996]).

Num modelo do tipo GAR, correspondente a um SP com um ou mais processos, onde há compartilhamento de recursos durante a realização dos processos, tem-se um ciclo fechado quando:

- A rota de um determinado processo cruza-se sobre si mesma antes de atingir a etapa final de seqüência de produção. Correspondente à situação em que há compartilhamento de recursos dentro de um

mesmo processo, ou seja, os recursos são utilizados repetidas vezes em diferentes etapas que constituem o processo.

- As rotas de processo diferentes cruzam-se antes de atingirem as etapas finais de cada sequência de produção, Corresponde à situação em que os recursos compartilhados são utilizados de forma concorrente para a execução de uma variedade de processos diferentes,

Desta forma existem dois tipos básicos de ciclos fechados: um formado pela intersecção de uma única rota sobre si mesma e o outro formado pela intersecção de várias rotas diferentes. Estes ciclos fechados em um GAR são denominados **CFE – ciclos fechados de espera**. O nó correspondente à intersecção é denominado *nó de intersecção*.

Considerando o procedimento de conexão contínua em um GAR em que há ciclos fechados, identifica-se o **PAI** (par de alocação de intersecção) como sendo o par de alocação que contém um determinado nó de intersecção. Por sua vez, o par de alocação que contém o arco que aponta para o nó de intersecção é denominado **PAT** (par de alocação terminal). Neste contexto, define-se ainda **segmento** como sendo a conexão contínua a partir de um determinado PAI até o próximo PAI.

Em síntese, um ciclo fechado é constituído por vários segmentos. A conexão de segmentos em um ciclo fechado é classificada como descontínua em virtude de não existência de uma rota única, a não ser o caso em que o

ciclo fechado possui um único segmento porque o CFE origina-se da intersecção de uma única rota sobre si mesma. A figura 4.3.1 mostra os principais elementos de um GAR citados até o momento.

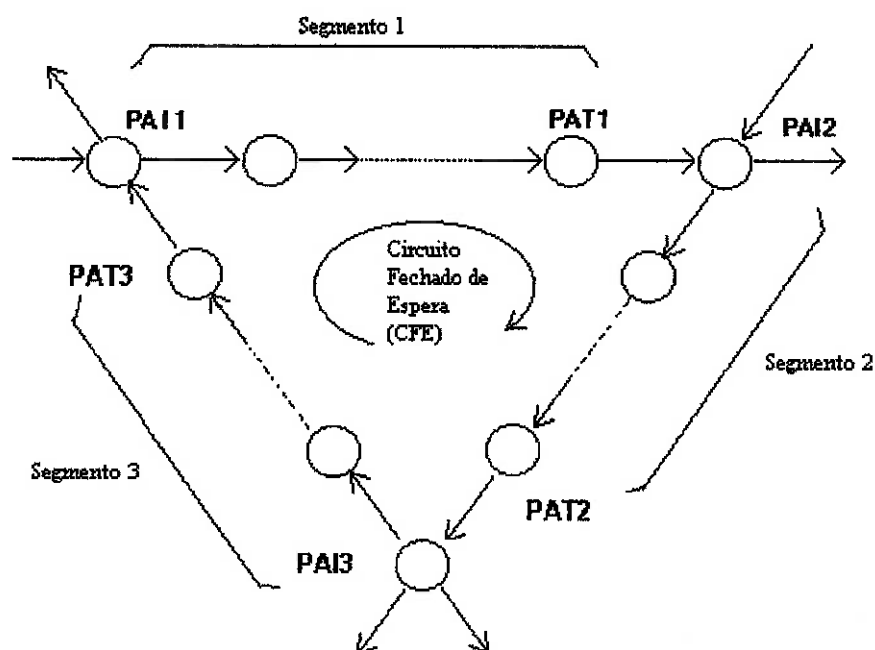


Figura 4.3.1 Modelo GAR contendo CFE, PAI e PAT (HASEGAWA et al [1996c])

B) Detecção de 'Deadlock'

Um CFE pode atingir o estado em que todas as etapas do processo ao longo do ciclo fechado estão na condição de retenção do respectivo recurso enquanto aguarda a liberação do próximo (*retenção enquanto espera*), correspondendo, portanto, ao estado em que **todos os recursos** encontram-se alocados para executarem a correspondente etapa de sequência de produção contidas no ciclo em questão. Este estado é denominado **estado de retenção**, em analogia ao 'circular wait state' definido em HASEGAWA [1996]. Portanto. É

fundamental afirmar-se que qualquer CFE pode potencialmente atingir o estado de retenção.

Considere a situação de um par de alocação em que todas as unidades de recurso associadas a ele estão ocupadas executando o número máximo de etapas de processo possível. Se houver uma requisição no sentido de avançar o processo para a próxima etapa e esta transição de estado não puder ser habilitada então o estado corrente é de bloqueio.

Baseando-se no fato de que um CFE está no estado de retenção se e somente se todos os recursos do CFE estão ocupados, então, nesta situação, o estado de retenção corresponde ao estado de 'deadlock' do fluxo de materiais no sistema o que implica em afirmar-se que todos os segmentos deste circuito estão no estado de bloqueio, aguardando a liberação. Neste contexto, tem-se o seguinte teorema (HASEGAWA [1996]).

Teorema 4.1 Supondo que um sistema possui apenas um CFE, o sistema está no estado de retenção se e somente se todos os pares de alocação do CFE estiverem bloqueados.

Portanto, o problema de 'deadlock' será abordado considerando-se a possível ocorrência de bloqueio dos pares de alocação.

C) Controle para evitar o 'Deadlock'

Em SANTOS FILHO [1998] aplicou-se uma técnica de projeto de compensadores para evita-se o 'deadlock' baseado na estratégia do FIS – Flow-in Suppresion (BANASZAK; KROG [1990], HASEGAWA et al. [1996^a],

HASEGAWA et al. [1996b]). Entretanto, independente de qual elemento um determinado recurso estiver processando, o fluxo já está limitado sem que se verifique de fato se este elemento é um potencial de risco para travar o sistema, ou seja, se corresponde efetivamente em um estado anterior ao 'deadlock' ou se, simplesmente, corresponderia a uma taxa de utilização máxima daquele recurso. Portanto, os compensadores, ainda que resolvam o problema de travamento, estão distantes de caracterizarem uma solução ótima para o problema.

Uma vez que o modelo complementar de utilização dos recursos é um tipo de representação isomórfica do GAR em termos de descrição de estados alcançáveis por este, torna-se bastante simples mapear um estado do GAR no grafo E-MFG que representa a utilização destes recursos. Portanto, se for detectado o estado anterior ao 'deadlock' basta restringir-se o disparo de determinadas transições que poderiam provocar a evolução do comportamento dinâmico do sistema para a condição de 'deadlock'.

Em YOSHIDA [1998] e SANTOS FILHO [1993] define-se estado anterior ao 'deadlock' a situação em que em um CFE possui apenas um recurso disponível em todos os seus segmentos estando os demais recursos ocupados realizando algum tipo de processamento.

Portanto, identificar os ciclos fechados de um GAR e obter os elementos que estariam sendo processados em cada um destes CFEs, corresponde, portanto, a uma determinada marcação do garfo E-MFG.

Para determinar os possíveis estados anteriores ao 'deadlock' em um dado CFE considera-se um recurso vago em cada um dos pontos de intersecção dos segmentos que compõem o CEF, uma vez que estes são também os pontos que permitem a entrada de novos elementos no ciclo.

Neste contexto, deve –se primeiramente selecionar as transições de entrada de cada Box que representa cada ponto de intersecção do CEF no GAR. A cada uma destas transições deve-se adicionar regras de controle que identificam a marcação correspondente ao estado anterior ao 'deadlock' – por hipótese assume que neste ponto de intersecção é que existe o recurso disponível – e, identificando este estado, esta transição é habilitada até que o sistema mude de estado.

Este procedimento deve ser aplicado a todos os pontos de intersecção de todos os CFEs existentes no GAR. Concluída esta etapa de especificação e inserção das regras no grafo E-MFG, este passa a ser denominado modelo do controle de utilização de recursos de um SP.

Em síntese, o procedimento para definir-se o conjunto de regras de controle que deve ser associado a determinadas transições do grafo E-MFG de controle de utilização dos recursos de um SP é o seguinte:

- a) Identificar os ciclos fechados de espera (CFEs) do GAR associados ao SP
- b) A cada CFE associar o conceito de marcação no E-MFG equivalente ao estado de 'deadlock'

- c) Em cada CFE, identificar os pontos de intersecção e:
- i) Associar para cada ponto um estado anterior ao 'deadlock'.
 - ii) Aplicar um conjunto de regras para identificar este estado e associar estas regras adicionais de controle desabilitadoras de transição de entrada do Box do grafo E-MFG correspondente ao ponto de intersecção do GAR em questão.

4.4. Observações Complementares

Neste Capítulo apresentou-se uma arquitetura de controle hierárquica para a solução do problema de modelagem do controle de SPs, principalmente no caso em que o sistema envolve processos complexos.

Especificamente, tratou-se do problema de construção do modelo de utilização de recursos a partir de um grafo não marcado (GAR). Em seguida, apresentou-se uma sistematização para obter-se o modelo E-MFG de controle a partir deste GAR, aplicando-se inclusive o método de fusão de lugares para mapear o compartilhamento de recursos entre os diversos processos em um modelo global.

Utilizando ainda as propriedades do GAR, apresentou-se um conjunto de procedimentos para serem aplicados no sentido de gerar regras adicionais de

controle às transições do grafo E-MFG para detectar os estado anterior ao 'deadlock' em si.

Desta forma, representa-se um E-MFG um modelo do sistema de controle de utilização de recursos que baseiam-se no conceito de modelo complementar em que na verdade são especificadas as regras pertinentes ao comportamento não desejado para o sistema, uma vez que o comportamento desejado é complexo para ser representado como referencial para um sistema de controle.

5. Sistema Mixer

5.1. Introdução

O sistema Mixer consiste de sub-sistemas distintos que serão descritos separadamente. Estes sub-sistemas são:

- . Mixers auxiliares
- . Seqüenciador de Mixer
- . Putup
- . Black
- . Oil

5.2. Overview do sistema Mixer

Este sistema é razoavelmente complicado e possui uma grande gama de informação. Cada subsistema tem um ou mais processos, e cada processo possui uma grande quantidade de estados e condições. Apesar da maioria dos estados e condições serem normais e esperados, alguns não são sendo necessária uma análise e modelagem correta dos processos.

O processo ocorrido no Mixer ocorre através de 5 “macro” processos que podem ser citados como:

- . Processo de inicialização
- . Processo de setup da máquina
- . Processo de Inserção dos componentes que formam a borracha
- . Processo de mistura dos componentes
- . Processo de descarregamento da borracha

Todos esses processos podem ser visualizados no diagrama de estados na figura 5.2.1.

5.3. Descrição de cada um dos processos

Os estados e transições de cada processo foram nomeados conforme o diagrama de estados.

5.3.1. Processo de inicialização

S1 - Idle (Ocioso) – Estado onde nenhum processo está sendo efetivado pelo Mixer.

E1 - Start Batch – Dispositivo de comando e atuação para o início do processo no Mixer

S2 - Process Recipe Data – Carregamento da memória do CLP da “receita” pertinente à composição da borracha, isto é, a quantidade e os componentes que formam a borracha

5.3.2. Processo de setup da máquina

S3 - Set equipment - Dispositivo de atuação na qual a máquina é preparada automaticamente para a inserção dos componentes

S4 - Wait for system ready – Estado onde a máquina está aguardando as condições necessárias para a inserção da mistura.

E3 - System ready – Dispositivos de detecção indicam que o sistema está pronto para receber a mistura

E4 - Manual mode request – Dispositivo de comando onde o processo será feito manualmente.

E2 - Auto mode request – Dispositivo de comando onde o processo será feito automaticamente.

S7 – Set equipment manual – Estado onde a máquina será preparada manualmente pelo operador

5.3.3. Processo de inserção dos componentes que formam a borracha

S5 – Run load sequence – Dispositivos de atuações onde todos os componentes serão inseridos ao Mixer conforme à “receita” feita pelo processo de inicialização.

E5 – Load Cycle Warning – Dispositivo de detecção verificou alguma condição em estado de alerta.

E6 – Charging Complete – Dispositivo de detecção verificou carregamento completado.

E10 – Load Phase Complete – Dispositivo de detecção verificou fase de carregamento completada.

S6 - Set equipment - Dispositivos de atuação preparam, automaticamente, os equipamentos para o processo de mistura.

E11 - Auto request in mix cycle - Dispositivo de comando onde o processo de mistura será realizado pelo modo automático.

E12 - Auto request in Load Cycle - Dispositivo de comando onde o processo de inserção será realizado pelo modo automático.

S9 - Manual mode processing - Estado onde o processo de mistura será realizado manualmente.

S13 - Load cycle warning - Estado onde ocorreu alguma situação de alerta no sistema.

E14, E15, E16, E17, E18 - Dispositivos de detecção, (temperatura, tempo, etc) que indicam se o processo está funcionando corretamente.

5.3.4. Processo de mistura.

S10 - Run sequence - Dispositivo de atuação aonde a mistura é feita até atingir-se as condições pré-programadas.

5.3.5. Processo de descarregamento da borracha

S11 – Drop Batch – Dispositivo de atuação onde a borracha é descarregada.

E22 – Batch Dropped – Dispositivo de detecção que indica se a borracha foi descarregada.

S12 – Reset equipment – Estado onde o Mixer está sendo reiniciado.

E19 – Mixer Discharge Cycle Complete – Dispositivo de detecção que indica se a fase de descarragamento foi realizada completamente.

S8 – Dropping Batch Manual – Estado onde o processo de descarregamento será feito manualmente.

S14 – Reporting – Dispositivo de monitoração onde todo o processo no Mixer é analisado.

S15 – Hold – Estado indicando os problemas ocorridos no processo do Mixer e esperando uma eventual linha de comando para reinicializar o processo.

E25 – Reset – Dispositivo de comando para reinicializar o processo.

6.Aplicação da Metodologia no Sistema Mixer

6.1. Análise das necessidades

6.1.1. Identificação do objetivo final do sistema

O objetivo final do sistema consiste em automatizar o processo de mistura da borracha que consiste em misturar diferentes componentes, como a borracha, pigmentação, black e óleo sob temperaturas e tempo determinados.

6.1.2. Estudo do objeto de controle, equipamentos e instalação

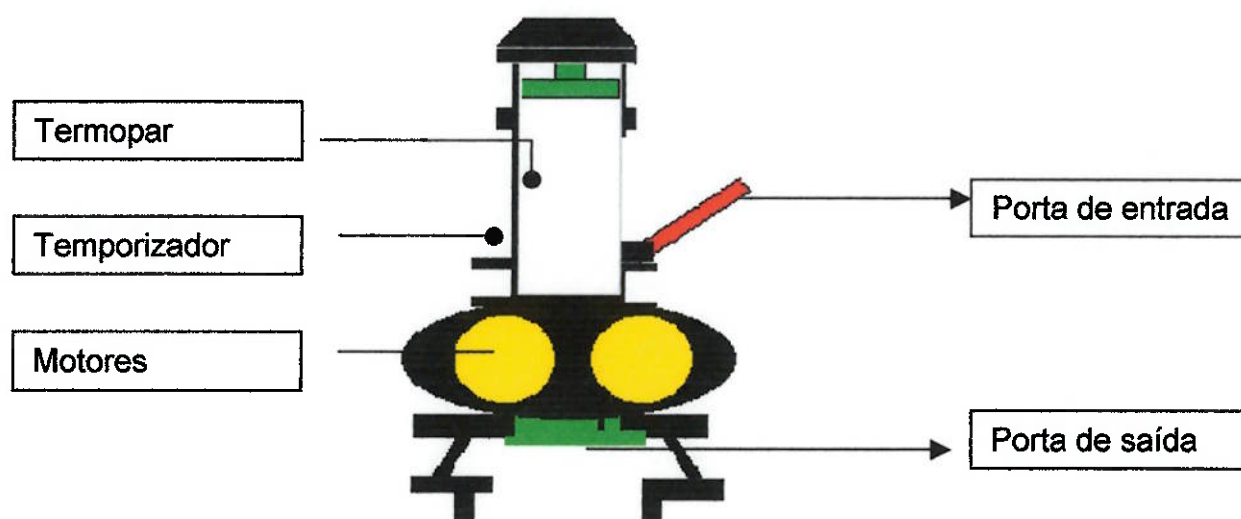


Fig. 6.1.2.1. Diagrama Estrutural do Objeto de Controle.

O carregamento da matéria-prima dar-se-á através do sistema Putup que será implementado nos próximos capítulos. Basicamente ele é constituído por um processo de pesagem de borracha e pigmentos, característicos de cada tipo de borracha, cuja quantidade é definida pelo químico responsável. Como o mixer é responsável pela mistura de diferentes tipos de borracha, o carregamento dar-se-á manualmente por um operador, auxiliado por equipamentos, onde a matéria prima é cortada e pesada automaticamente, com margens de erro pré estabelecidas. O Putup também é responsável pelo transporte da matéria prima ao Mixer através de esteiras e elevadores.

No Mixer são introduzidos, se necessário, o black e óleo, componentes comuns na borracha e que são pesados e introduzidos automaticamente, segundo pré-definição do químico.

O processamento da borracha está relacionado às variáveis: temperatura da mistura e tempo de processamento. Elas devem ser devidamente controladas para que o produto final atenda as expectativas. Essas variáveis de controle são definidas pelo químico responsável, variando para cada tipo de borracha desejada, sendo que o operador não tem acesso à mudança desses dados.

Todos os dados referentes a cada tipo de borracha processada são introduzidos ao sistema de controle através de um banco de dados.

Após a mistura ser efetuada ela é descarregada sobre um laminador onde é prensada e resfriada.

Nas tabelas a seguir, estão relacionados os equipamentos e dispositivos que formam o sistema.

Tabela 6.1.2.1. Equipamentos do Objeto de Controle.

Código	Descrição
Eq01	Corpo do Mixer
Eq02	Porta de Entrada
Eq03	Porta de Saída

Eq04	Motores
Eq05	Laminador
Eq06	Esteira de resfriagem

Tabela 6.1.2.2. Dispositivo de Comando.

Código	Descrição (Evento relacionado)
Co01	Botão Ligar ou Desligar o Mixer (E1)
Co02	Botão Seletor Modo Automático Manual (E2, E4)
Co03	Botão Modo Automático de Mistura (E11)
Co04	Botão Modo Automático de Carregamento (E12)
Co05	Botão de Reinicialização Após Falha (E25)

Tabela 6.1.2.3. Dispositivos de detecção

Código	Descrição (Evento relacionado)
De01	Sistema Pronto para Carregamento (E3)
De02	Sistema Completamente Descarragado (E19)
De03	Sensores para carregamento (E5, E6, E9, E10)
De04	Temporizador (E13, E15, E23, E26, E27)

De05	Termopar (E13, E14, E16, E21, E26, E27, E28)
De06	Potenciômetro (E13, E17)
De07	Detector de Abertura da Porta (E18)
De08	Sensor de Injeção de Aditivos (E20)
De09	Sensor de Descarregamento (E22)

6.1.3. Levantamento e análise das funções de controle.

No diagrama a seguir, temos relacionadas as funções de controle.

Tabela 6.1.3.1. Diagrama das Funções de Controle

Nível 1	Nível 2	Nível 3
Processamento Automático	Inicialização	Energização e inicialização dos equipamentos
	Recebimento	Comunicação com Banco de Dados
	Receita	Recebimento de variáveis da Receita
	Preparação Equipamentos	Envio de parâmetros aos equipamentos
	Carregamento/ Processamento	Seqüência de Carregamento
		Preparação dos equipamentos
		Ativação do sistema de Mistura
	Descarregamento	Desativação do sistema de Mistura
		Preparar equipamentos para descarga
		Descarregamento
		Preparar equipamentos
	Checagem Falhas	Checagem de sensores para detectar ocorrência de anormalidades

7. Diagramas PFS/MFG do sistema

Mixer

7.1. Diagramas traduzidos

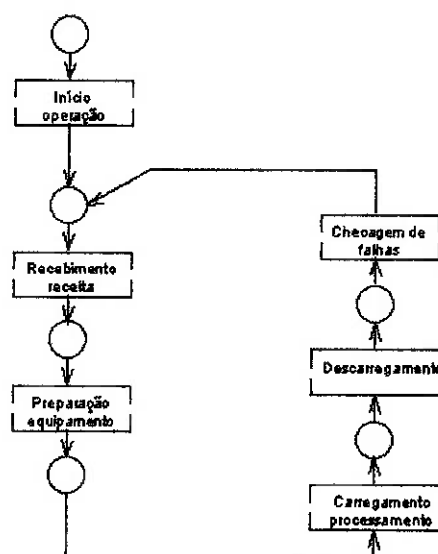


Fig. 1. Modelo PFS do Mixer

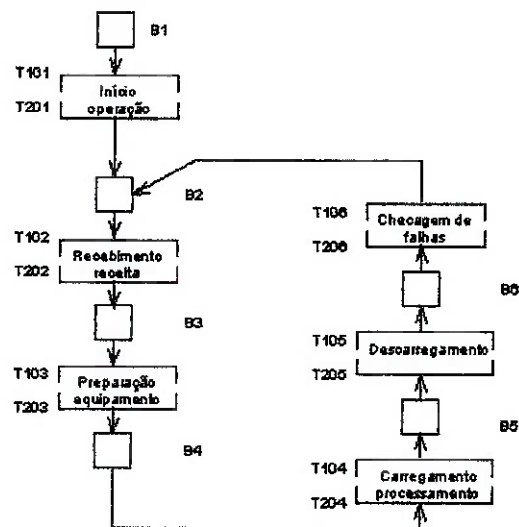


Fig. 2. Modelo MFG/PFS do Mixer

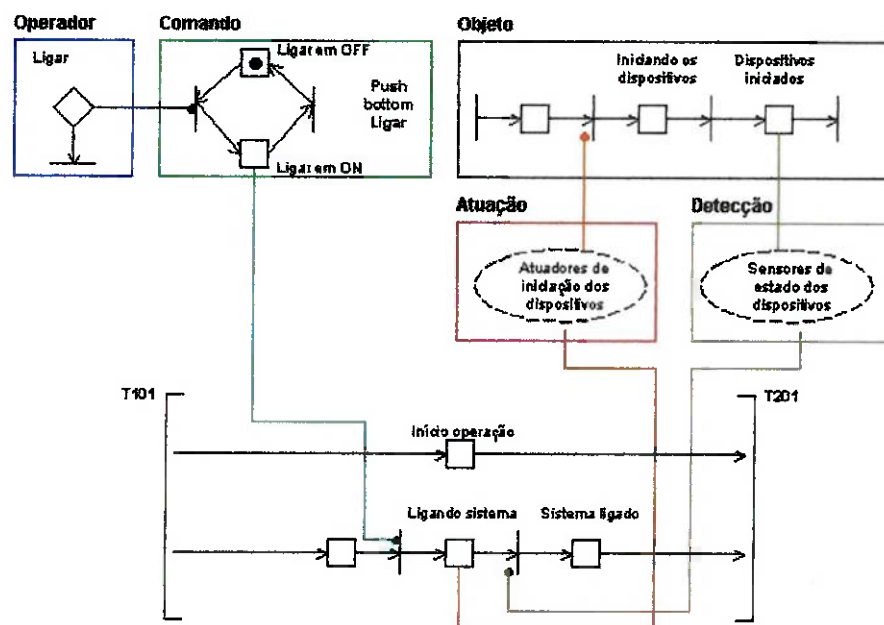


Fig. 3. Modelo MFG/PFS da atividade /Início operação e conexão com dispositivos

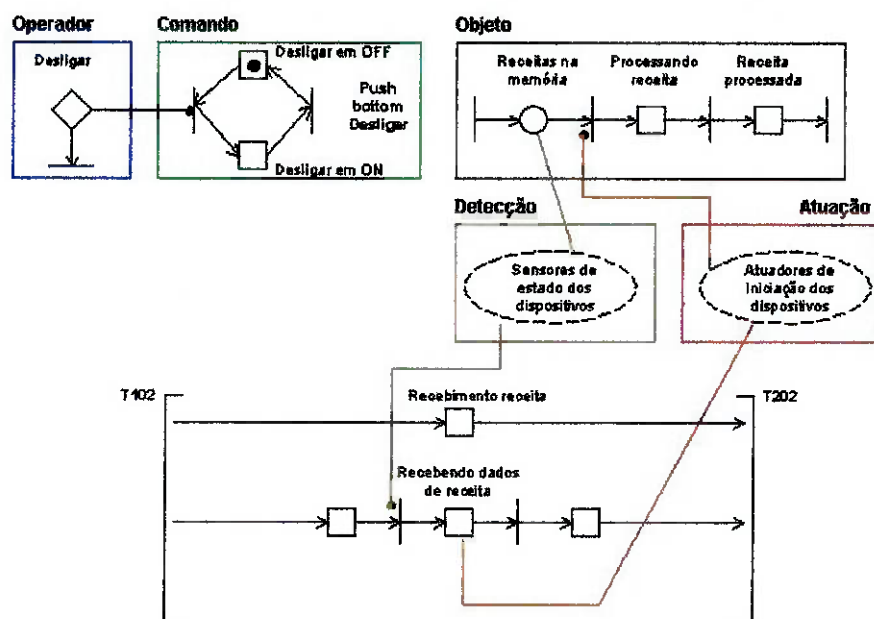


Fig. 4. Modelo MFO/PFS da atividade /Recebimento receita e conexão com dispositivos

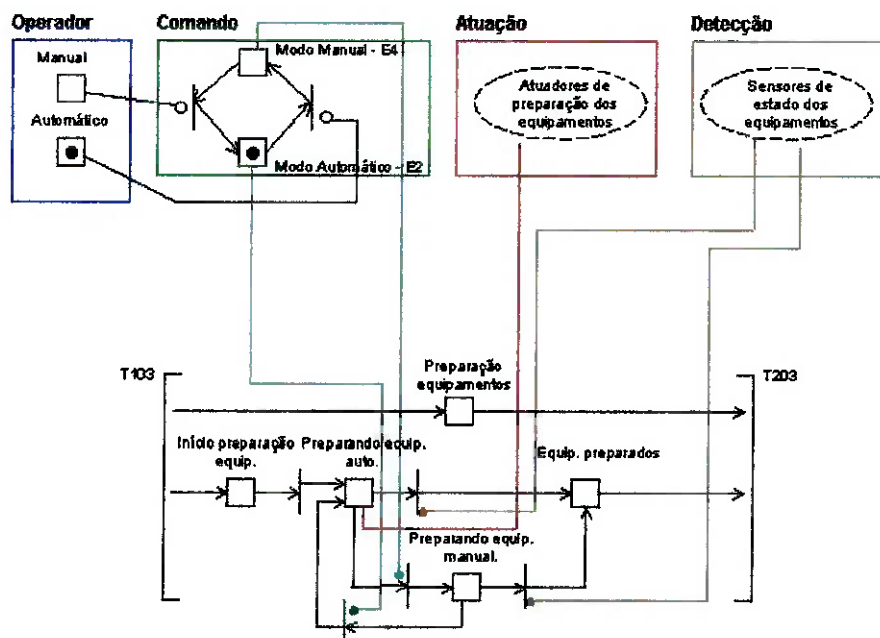


Fig. 5. Modelo MFO/PFS da atividade /Preparação equipamentos e conexão com dispositivos

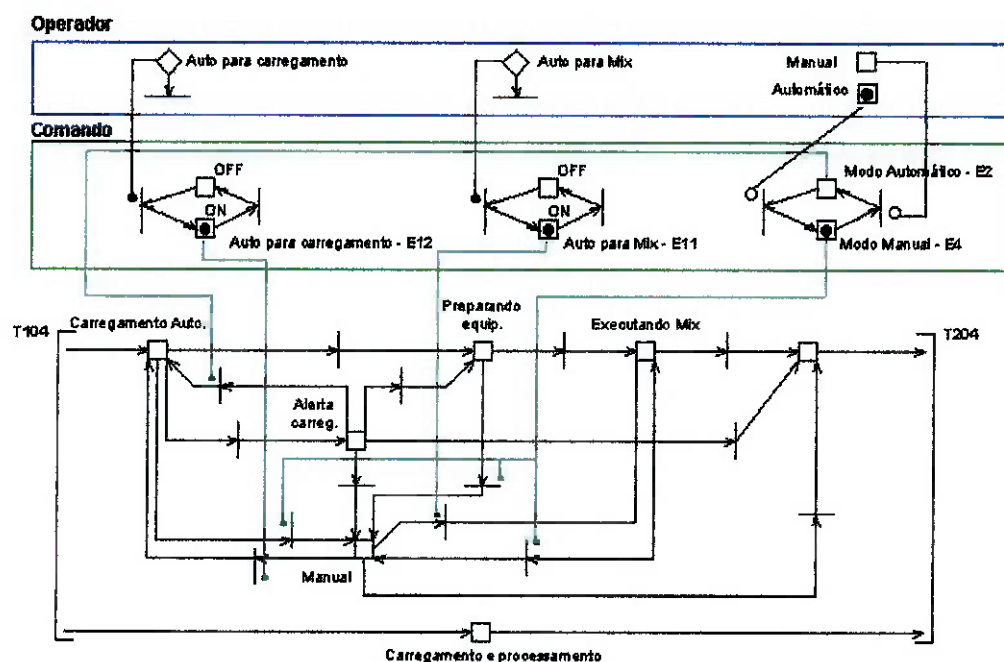


Fig. 6. Modelo MFG/PFS da atividade /Carregamento e processamento e conexão com dispositivos de comando

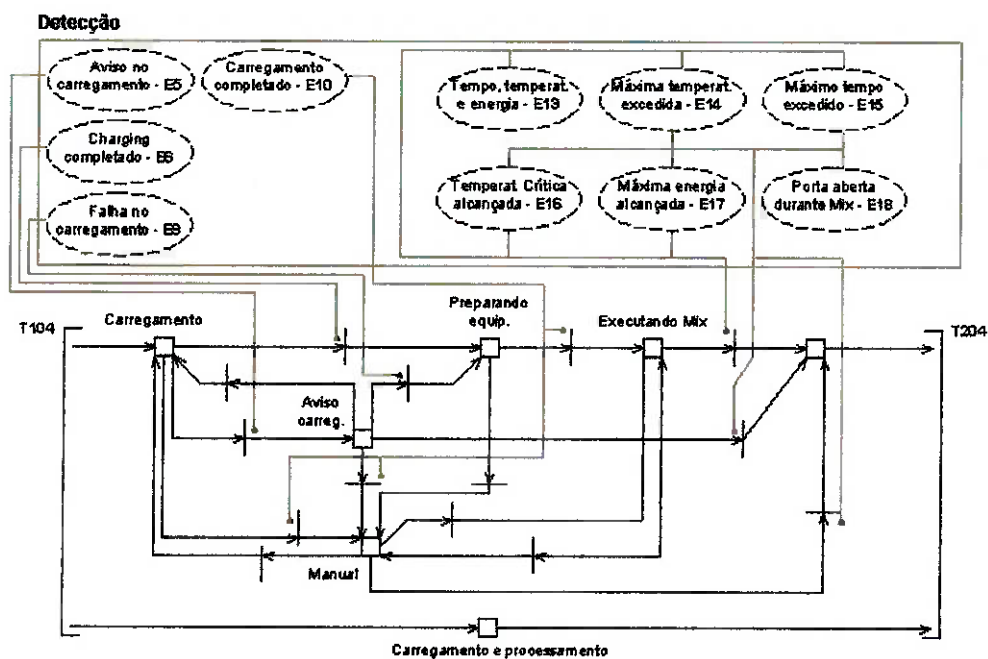


Fig. 7. Modelo MFG/PFS da atividade /Carregamento e processamento e conexão com dispositivos de detecção

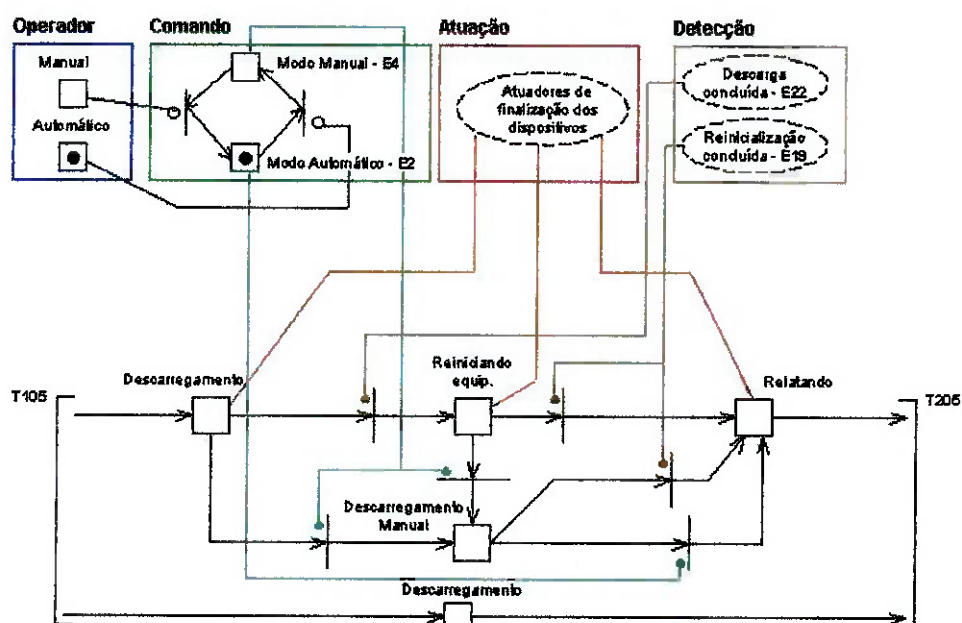


Fig. 8. Modelo MFG/PFS da atividade
Descarregamento e conexão com
dispositivos

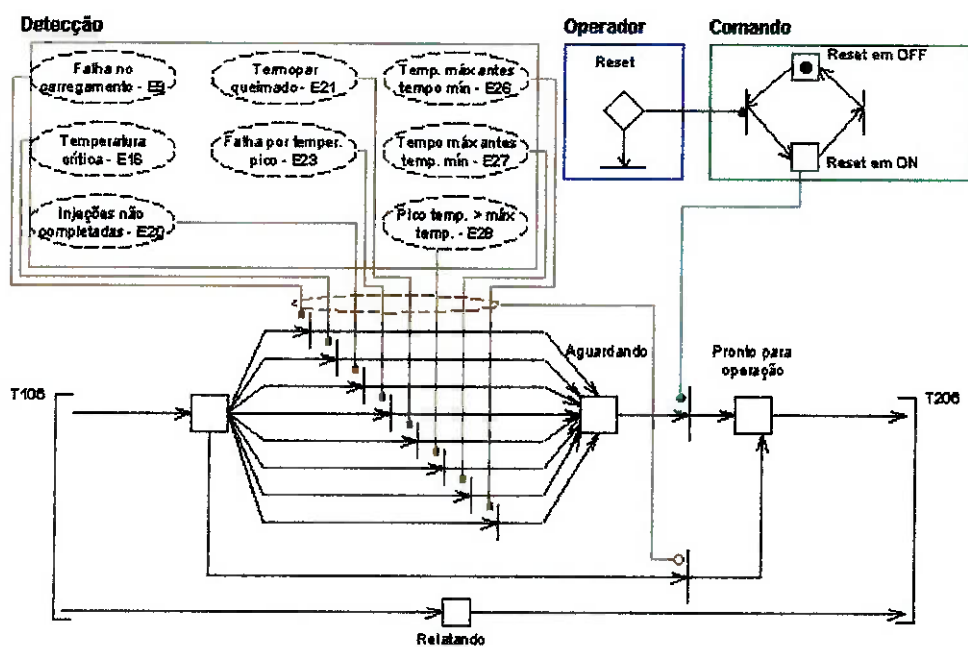


Fig. 9. Modelo MFG/PFS da atividade
Tratamento de falhas e conexão com
dispositivos

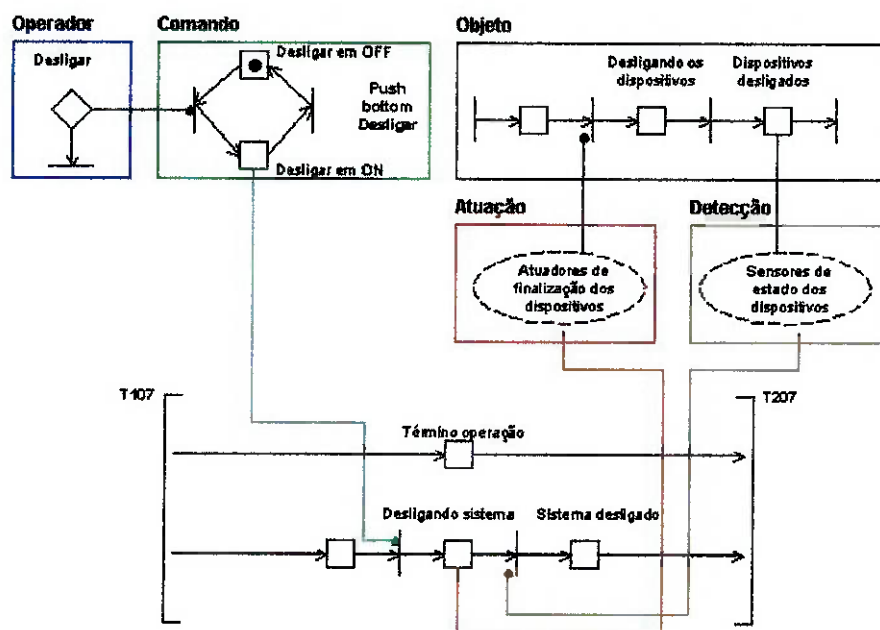


Fig. 10. Modelo MFG/PFS do da atividade
Término operação e conexão com
dispositivos

7.2. Diagramas aperfeiçoados

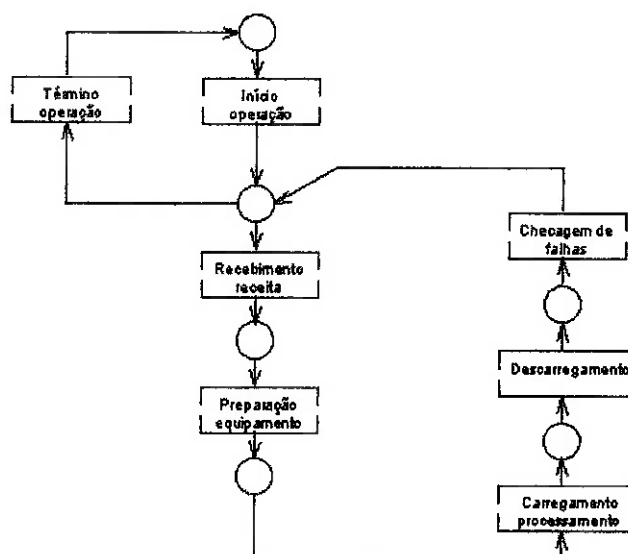


Fig. 1. Modelo PFS do Mixer

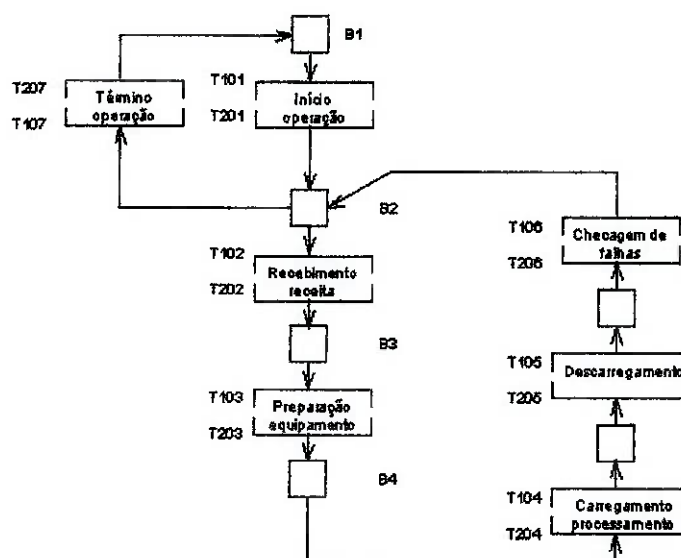


Fig. 2. Modelo MFO/PFS do Mixer

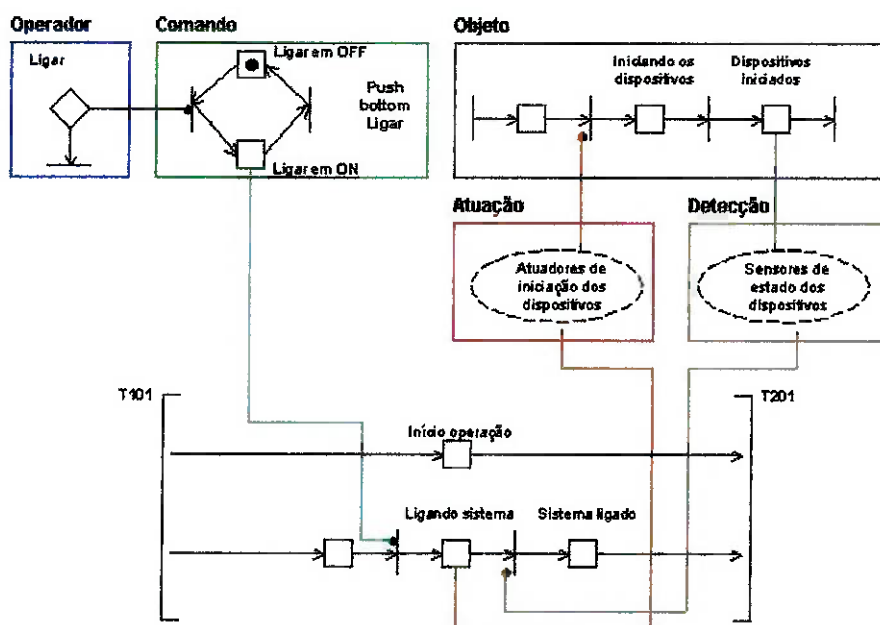


Fig. 3. Modelo MFG/PFS da atividade Início operação e conexão com dispositivos

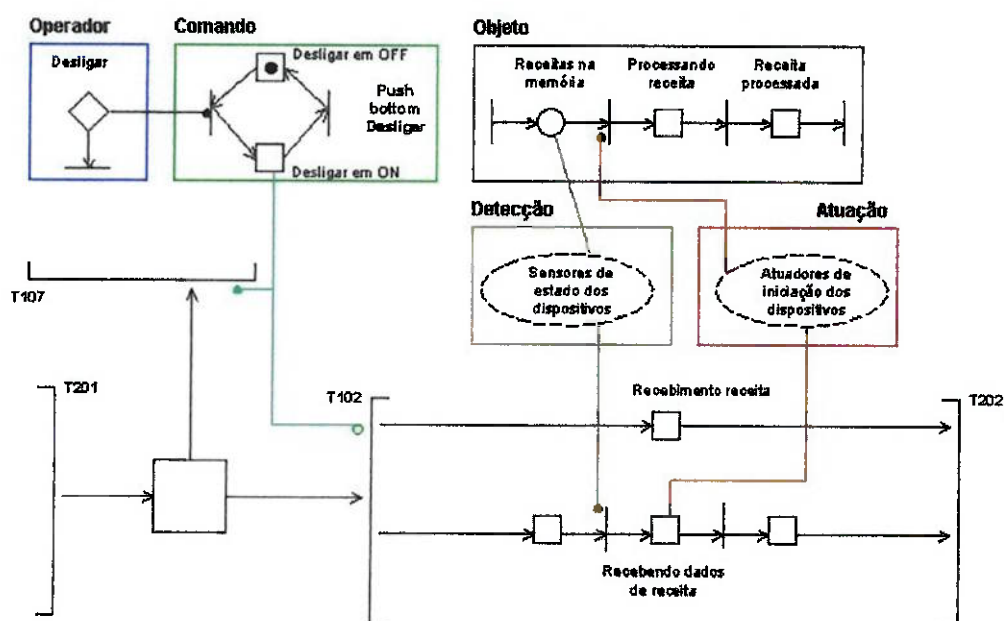


Fig. 4. Modelo MFG/PFS da atividade /Recebimento receita e conexão com dispositivos

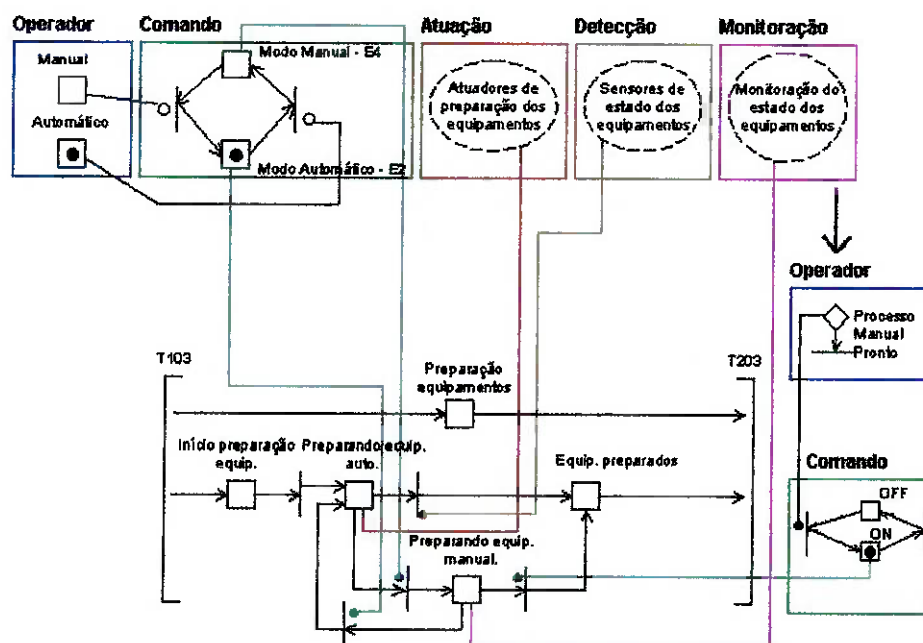


Fig. 5. Modelo MFG/PFS da atividade /Preparação equipamentos e conexão com dispositivos

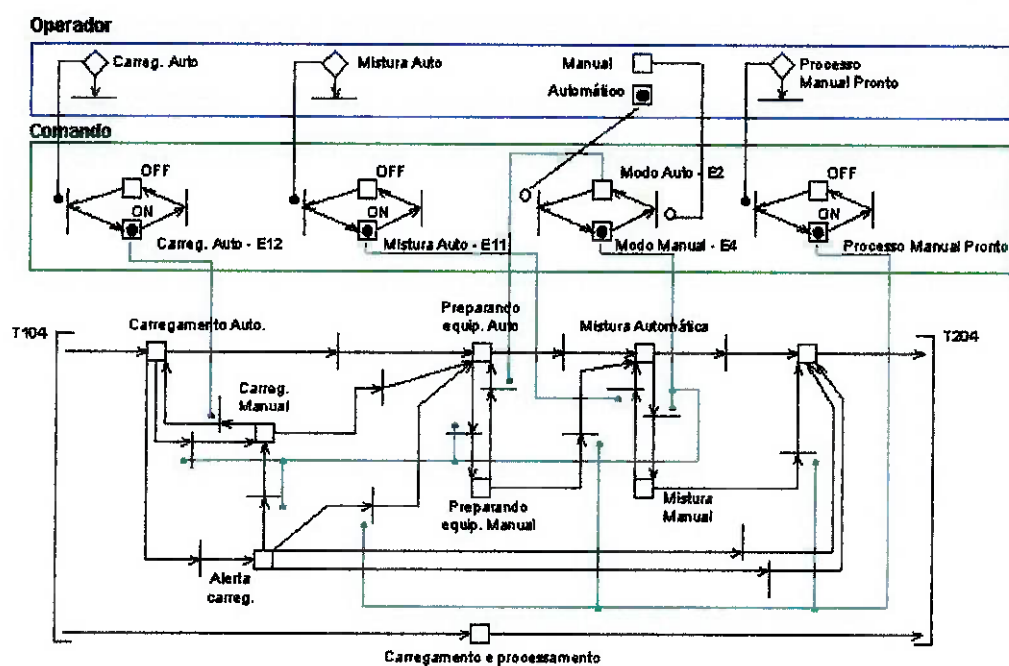


Fig. 6. Modelo MFG/PFS do da atividade /Carregamento e processamento e conexão com dispositivos de comando

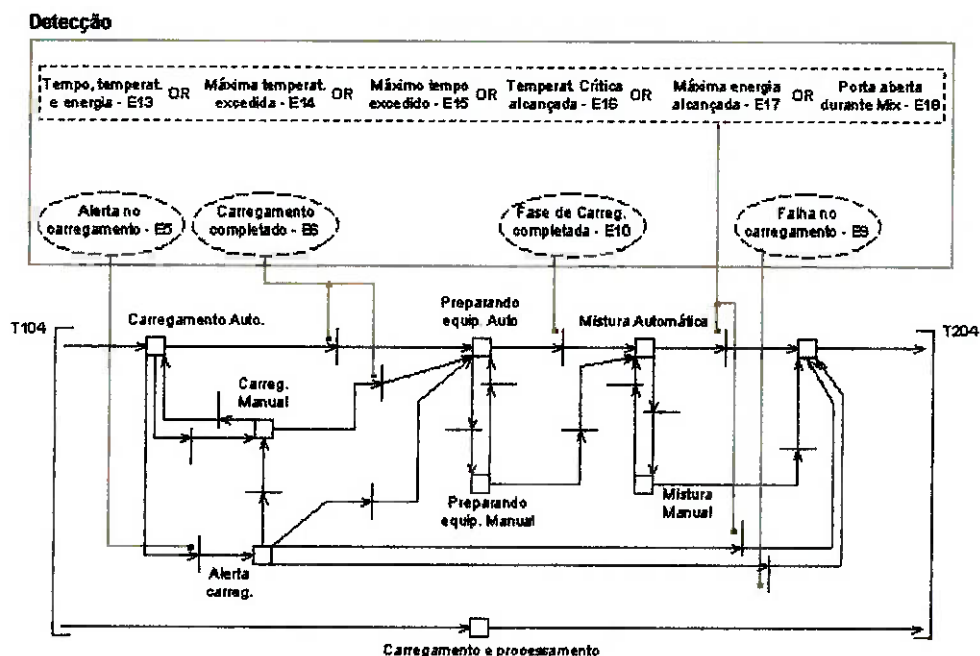


Fig. 7. Modelo MFG/PFS do da atividade /Carregamento e processamento e conexão com dispositivos de detecção

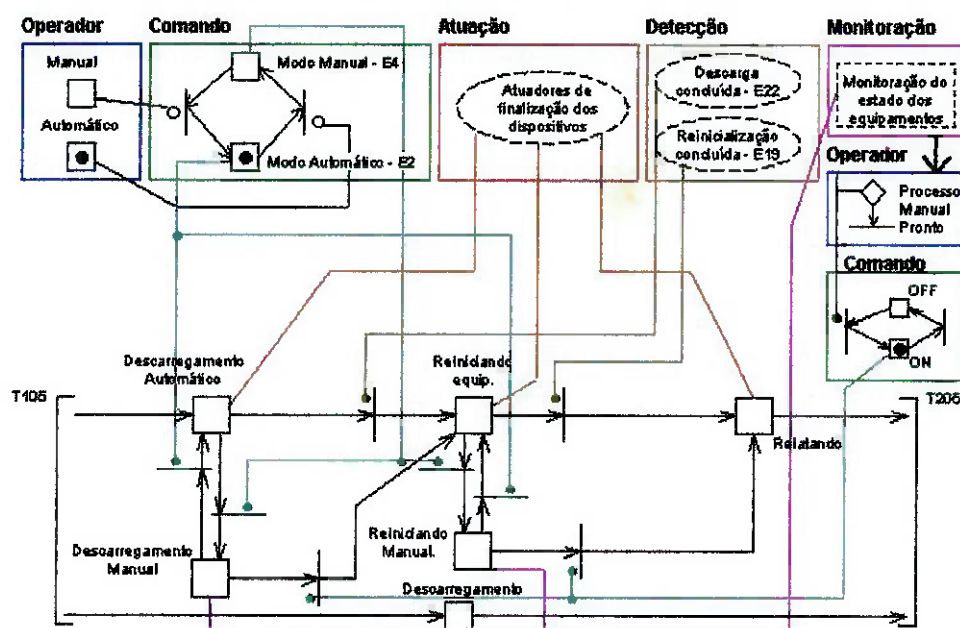


Fig. 8. Modelo MF G/PFS da atividade /Descarregamento e conexão com dispositivos

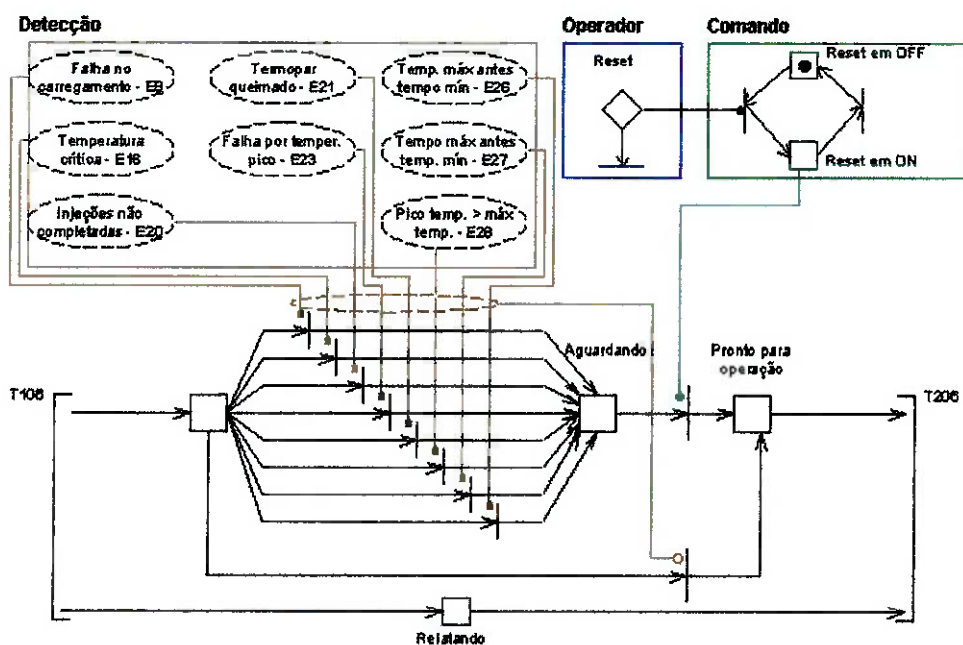
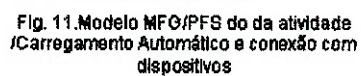
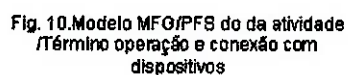


Fig. 9. Modelo MF G/PFS da atividade /Tratamento de falhas e conexão com dispositivos



8. Conclusões

Primeiramente foi possível identificar a inconsistência do Diagrama de Estados (DTE) (Autômatos Finitos), quanto a sua lógica e implementação de controladores lógicos.

Este problema foi possível identificar devido a grande discrepância existente entre o modelo real e o Diagrama de Estados (DTE), além do fato da simulação do Diagrama de Estado ser difícil de implementar devido a dificuldade em se observar os estados globais, característica própria do mesmo.

Uma das grandes vantagens em se implementar o sistema de controle através de diagramas PFS/MFG, além da sua consistência, é a sua facilidade na implementação de controladores lógicos, uma vez que a sua linguagem é bastante próxima à linguagem utilizada pelos mesmos, o que por sua vez, não ocorre com o Diagrama de Estados.

Bibliografia

- [1] BANASZAK, Z.^a; KROG, B.H. Deadlock avoidance in Flexible Manufacturing Systems with concurrently competing process flows. **IEEE Transaction on Robotics and Automation**, v.6, n.6, p.724-34, 1990.
- [2] CASSANDRAS, C.G. **Discrete Event Systems**, IRWIN, 1993.
- [3] HASEGAWA, K. et al. Graphical Analysis and Syntethesis of in Flexible Manufacturing Systems. In 17th Internacional Conference on Aplication and Theory of Petri Net, **Proceedings** Osaka, Japan, p. 161-176, 06/1996^a.
- [4] HASEGAWA, K. et al. Transform of discrete production net request/allocation graph. In: '96 Joint Symposium on System and Information, **Proceedings**. SICE, Matsuyama, Japan, p.213-18, 10/1996.
- [5] HASEGAWA, K. et al. Deadlock avoidance policy based on request/allocation graph. In: '96 Joint Symposium on System and Information, **Proceedings**. SICE, Matsuyama, Japana, p.219-24,10/1996 (in Japanese).
- [6] MIYAGI, P.E. **Control System Design, Programming and Implementation for Discrete Event Production Systems by using**

Mark Flow Graph, Japan, 1988. Tese de Doutorado – Tokyo Institute of Technology.

[7] MIYAGI, P.E. **Controle Programável – Fundamentos do controle de sistemas a eventos discretos**. S. Paulo, Ed. Edgard Blücher Ltda., 1996.

[8] SANTOS FILHO, D. J.; MIYAGI, P. E., **Sistemas de Eventos Discretos e seu Controle**. In: 1º COBISA/CINISA, Anais ISA International, São Paulo, SP, p.2.1-12, 05/1991.

[9] SANTOS FILHO, D.J.; **Proposta do Mark Flow Graph Estendido para a Modelagem e Controle de Sistemas Integrados de Manufatura**. S. Paulo, 1993. Dissertação (Mestrado) – Escola Politécnica, Universidade de S. Paulo.

[10] SANTOS FILHO, D.J.;MIYAGI, P.E. **Realização de Controle de Tarefas de Veículos Autônomos de Transporte Através do MFG Estendido com Marcas Individuais**. In: 6º CONAI –CONGRESSO NACIONAL de AUTOMAÇÃO INDUSTRIAL, Anais. SUCESU – SP/Automática, São Paulo, SP, 1994.

[11] SANTOS FILHO, D.J.;MIYAGI, P.E. **Enhanced Mark Flow Graph to Control Flexible Manufacturing Systems**. **Jornal of The Brazilian Society of Mechanical Sciences**, ABCM, Rio de Janeiro , RJ, v.17, n2, p.232-248, 1995^a.

- [12] SANTOS FILHO, D.J.;MIYAGI, P.E. Enhanced Mark Flow Graph to Control Autonomous Guided Vehicle. In: CAPE'95 COMPUTER APPLICATIONS in PRODUCTION AND ENGINEERING. Proceedings. IFIP, Beijing, China, p.856-865, 1995b.
- [13] SANTOS FILHO et al. Proposta de um Ferramenta Computacional para Análise de Grafos E-MFG por Simulação. In: 11º CBA – CONGRESSO BRASILEIRO de AUTOMÁTICA, *Anais*. SBA, São Paulo, SP, v.II, p.689-694, 09/1996.
- [14] SANTOS FILHO, D.J.;MIYAGI, P.E. Proposta de uma Ferramenta Automática de Programação de CPs a partir de Modelos MFG. In XIV COBEM – CONGRESSO BRASILEIRO de ENGENHARIA MECÂNICA, *Anais*, CD-ROM. ABCM, Bauru, SP, 12/1997.
- [15] SANTOS FILHO, D.J. **Controle de Sistemas Antropocêntricos de Produção Baseado em Redes de Petri Interpretadas**. S. Paulo, 1998. Tese (Doutorado) – Escola Politécnica, Universidade de S. Paulo.
- [16] SANTOS FILHO, D.J.;MIYAGI, P.E.; MARUYAMA, N. FRAZÃO, C. Production Systems Control Based on Interpreted Petri Nets. In: ECEC'99 6th EUROPEAN CONCURRENT ENGINEERING CONFERENCE 1999. Proceedings, Friedrich Alexander University, Erlangen-Nuremberg, Germany, 04/1999.
- [17] SANTOS FILHO, D.J. et al., Automatic Generation of Metamodels for Design of Production Systems. In: Mechatronics 2000 – 7th

MECHATRONICCS FÓRUM INTERNATIONAL CONFERENCE.

Proceedings, Atlanta, Geórgia, USA, 09/2000.

[18] YOSHIDA, S. Tagged MFG Based Control Of Materials Flow in Job-Shop Production Sustems, Dissertação de Mestrado, Tokyo Institute of Technology, Japan, 1998.