

JULIO ARAKAKI

**ANÁLISE DE SISTEMAS DE MANUFATURA
ATRAVÉS DA METODOLOGIA MFG/PFS E
REGRAS DE PRODUÇÃO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
de título de Mestre em Engenharia.

São Paulo
1993

OK

JULIO ARAKAKI

**ANÁLISE DE SISTEMAS DE MANUFATURA
ATRAVÉS DA METODOLOGIA MFG/PFS E
REGRAS DE PRODUÇÃO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
de título de Mestre em Engenharia.

Área de Concentração:
Engenharia Mecânica

Orientador:
Prof. Dr. Paulo Eigi Miyagi

São Paulo
1993

DEDALUS - Acervo - EPMN



31600009966

*À minha família, em especial aos meus pais Kaoru e Yolanda,
pela oportunidade de estar aqui e a Lilian pela compreensão e
dedicação em todos os momentos.*

AGRADECIMENTOS

Ao Prof. Dr. Paulo Eigi Miyagi pelo apoio, orientação e confiança dedicados neste período.

Ao sempre amigo Prof. Diolino J. S. Filho pelo incentivo e ajuda nas revisões e discussões.

Aos amigos Nilson, Márcio, Arata e a todos da Mecatrônica pelo apoio.

Aos que sempre me incentivaram: Reginaldo, Harumi, Roberta, ..., enfim à todos que de uma forma ou de outra, ajudaram na realização deste trabalho.

SUMÁRIO

RESUMO.....	iv
"ABSTRACT".....	v
1 INTRODUÇÃO.....	1
1.1 AUTOMAÇÃO DA MANUFATURA.....	2
1.2 SISTEMAS AUTÔNOMOS.....	3
1.3 MOTIVAÇÃO E OBJETIVOS DO TRABALHO.....	5
1.4 ORGANIZAÇÃO DO TRABALHO.....	9
2 MODELAGEM DE SISTEMAS DE MANUFATURA.....	11
2.1 INTRODUÇÃO.....	11
2.2 SISTEMAS INTEGRADOS DE MANUFATURA (SIM).....	12
2.3 MODELAGEM UTILIZANDO REDES DE PETRI.....	14
2.3.1 Propriedades e características das Redes de Petri.....	18
2.4 DESENVOLVIMENTO HIERÁRQUICO.....	22
2.5 METODOLOGIA MFG/PFS.....	23
2.5.1 "Mark Flow Graph" (MFG).....	24
2.5.2 Regras de disparo do MFG.....	27
2.5.3 Macro elementos do MFG.....	29
2.5.4 "Production Flow Schema" (PFS).....	32
2.6 SISTEMAS BASEADO EM REGRAS.....	38
2.6.1 Aplicação de regras em Sistemas de Manufatura.....	41
3 ANÁLISE DOS SISTEMAS INTEGRADOS DE MANUFATURA.....	44
3.1 INTRODUÇÃO.....	44

3.2 O QUE É "DEADLOCK" NOS SIM ?.....	45
3.2.1 Exemplo de "deadlock"	46
3.2.2 Tipos de "deadlock" nos SIM.....	48
3.2.3 Como detectar os "deadlocks" nos SIM	48
3.3 MÉTODOS PARA ANÁLISE.....	50
3.3.1 Análise Estrutural.....	54
3.3.2 Análise Comportamental	55
3.4 LIMITAÇÃO DO MFG.....	57
3.4.1 Sistema híbrido (MFG e Regras).....	58
4 FERRAMENTA DE SIMULAÇÃO E ANÁLISE.....	61
4.1 INTRODUÇÃO.....	61
4.2 DESCRIÇÃO DA FERRAMENTA.....	63
4.3 ESTRUTURA DA FERRAMENTA	64
4.4 FUNCIONAMENTO DA FERRAMENTA	66
4.4.1 Algoritmo para solução do "deadlock" estrutural.....	68
4.4.2 Algoritmo para solução de "deadlock" condicional.....	69
4.5 DETALHES DE IMPLEMENTAÇÃO DA FERRAMENTA	70
4.5.1 Estrutura de dados.....	71
4.5.2 Jogador de marcas	75
5 EXEMPLO DE APLICAÇÃO DA FERRAMENTA.....	81
5.1 INTRODUÇÃO.....	81
5.2 DESCRIÇÃO ESTRUTURAL E FUNCIONAL DO SISTEMA EM ESTUDO ..	82
5.2.1 Estação de entrada.....	83
5.2.2 Estação de processamento 1 e estação de processamento 2	83
5.2.3 Estação de medição.....	84
5.2.4 Sistema de transporte.....	84
5.2.5 Comportamento dinâmico do sistema integrado.....	85
5.3 MODELO MFG/PFS DO SISTEMA.....	87
5.3.1 Modelo PFS do sistema.....	88
5.3.2 A atividade ENTRADA	89
5.3.3 A atividade PROCESSO_1	90
5.3.4 A atividade PROCESSO_2.....	91
5.3.5 A atividade MEDIÇÃO.....	92
5.3.6 A atividade TRANSPORTE.....	93

5.3.7 A atividade SELECIONA_PEDIDO	94
5.3.8 A atividade DESLOCA	96
5.3.9 A atividade REALIZA_TRANSPORTE.....	97
5.4 ÁRBITRO EM MFG	99
5.5 ÁRBITRO EXTERNO BASEADO EM REGRAS	102
5.5.1 Regras para o Árbitro Externo	103
6 OBSERVAÇÕES FINAIS	105
REFERÊNCIAS BIBLIOGRÁFICAS	108
LEITURAS RECOMENDADAS	113
APÊNDICE	
REDES DE PETRI.....	A1
A.1 INTRODUÇÃO.....	A1
A.2 DESCRIÇÃO FORMAL.....	A2
A.3 PROPRIEDADES DAS REDES DE PETRI.....	A4
A.4 REDES CONDIÇÃO-EVENTO (C/E)	A6
A.4.1 Regras para a rede C/E.....	A7

RESUMO

O estudo sobre o comportamento dinâmico dos sistemas de manufatura é fundamental para verificar a existência e para solucionar a ocorrência de fenômenos indesejáveis, e muito comuns, como o "deadlock" e a colisão. Propõe-se, neste sentido, a especificação e implementação de uma ferramenta protótipo para análise dos sistemas de manufatura. A ferramenta utiliza uma abordagem hierárquica e estruturada ("top-down") através da metodologia MFG/PFS ("Mark Flow Graph/Production Flow Schema"). E, para solucionar conflitos que ocorrem nos sistemas, associa-se um árbitro externo baseado em regras de Produção. Estas regras associadas são eficientes no sentido de que permite a simplificação do modelo, pois toda a lógica responsável pela representação das condições necessárias para resolver uma situação de conflito, é transferida para o árbitro externo em forma de regras.

ABSTRACT

The investigation about dynamic behavior of manufacturing systems is important to verify and to solve problems (features) like a deadlock and collision. In this context, we propose the specification and implementation of a prototype tool to analyze manufacturing systems. This tool is based on a top-down approach through of MFG/PFS methodology. To solve conflicts occurred in these systems, an external arbiter, based on Production Rules, is applied. These rules are an efficient mechanism to simplify the model, because all logical represent conditions, necessary to solve conflicts, are transferred to an external arbiter, in Production Rules.

Capítulo 1

INTRODUÇÃO

Este trabalho propõe a utilização da metodologia MFG/PFS associado a um sistema baseado em regras. O resultado deste estudo levou a implementação de uma ferramenta protótipo para auxílio na análise dos Sistemas Integrados de Manufatura (SIM). Neste sentido, apresenta-se a proposta de sua estrutura e a sua respectiva implementação. Esta ferramenta utiliza para a modelagem dos SIM a metodologia MFG/PFS [HASEGAWA (1984), MIYAGI (1988)] e a tomada de decisões na solução de conflitos, é realizada através de um sistema baseado em regras de produção ("production rules") da Inteligência Artificial [CUNHA (1987), LUCENA (1987), RICH (1985)]. Num sistema de manufatura, os conflitos refletem situações onde recursos limitados disponíveis são solicitados por várias atividades num mesmo instante. Neste caso, é adequado que se condicione a escolha através do estabelecimento de um conjunto de regras condizentes com o comportamento dinâmico que se deseja para o sistema. Neste contexto, é importante verificar que se os conflitos não forem bem resolvidos, o sistema em estudo pode alcançar uma situação, onde parte do sistema ou o sistema inteiro deixa de evoluir (estado conhecido por "deadlock").

Desta forma, observa-se que a ferramenta proposta neste trabalho objetiva avaliar um conjunto de regras incorporado a um modelo MFG/PFS, no sentido de validar o modelo híbrido obtido neste contexto, auxiliando no processo de definição de sistemas de controle de SIM.

1.1 AUTOMAÇÃO DA MANUFATURA

Determinados setores do mercado são caracterizados pela forte competitividade. Neste caso, os sistemas responsáveis pelos processos industriais envolvidos apresentam um grau de automação cada vez mais elevado, acarretando como consequência, um aumento bastante significativo do grau de eficiência desses processos.

Para que os sistemas de manufatura automatizados incorporem a característica de flexibilidade, é imprescindível a possibilidade de uma mudança rápida de produção de um determinado tipo de produto para outro, de acordo com as necessidades. Porém, observa-se que essa flexibilidade acarreta também, num aumento sensível da complexidade do controle destes sistemas. Por isso, para a especificação e implementação desses sistemas de controle deve-se utilizar ferramentas cada vez mais poderosas que possibilitem a modelagem e a análise de maneira coerente e precisa dos sistemas em estudo. Sempre que o grau de complexidade dos sistemas produtivos for elevado, haverá dificuldades em utilizar-se de métodos clássicos como uma linguagem para especificação¹ [BARATA (1992)] e a teoria de Redes de Fila [SOARES (1990)]. Esta complexidade advém do comportamento dinâmico dos SIM. Sendo os SIM do tipo Sistema a Eventos Discretos (SED) [HO (1989), SANTOS F^o (1991)], que são caracterizados principalmente, pela ocorrência de eventos aleatórios², ou seja, estes sistemas evoluem no tempo governados por uma seqüência de eventos e não há necessariamente uma pré-

¹ Exemplo: EXPRESS - linguagem para modelagem formal de informações.

² A evolução está vinculada a condições pré-estabelecidas e a ocorrência de eventos.

determinação do instante em que cada evento deverá acontecer. Neste caso, requer-se uma evolução sincronizada e um alto grau de paralelismo das atividades processadas [ALLA (1986)]. Assim sendo, utiliza-se neste trabalho uma técnica de simulação baseada em redes de Petri [REISIG (1982), REISIG (1992), PETERSON (1991)] para possibilitar a realização da análise de forma eficiente e adequada.

1.2 SISTEMAS AUTÔNOMOS

Conforme citado antes, a necessidade sempre crescente de melhorar a qualidade e aumentar a flexibilidade para a fabricação de seus produtos, está levando os setores mais competitivos da indústria a automatizar os processos. Esta automatização é realizada, a partir da utilização de subsistemas cada vez mais autônomos, integrados através de sistemas de informações eficientes. Neste contexto, o objetivo principal dos técnicos e especialistas que atuam nessa área de automação é o desenvolvimento de sistemas com características de "autonomia", propriamente dito, no sentido de otimizar o processo produtivo a partir da restrição da participação das decisões nos vários níveis do processo.

No trabalho de BOURNE e FOX (1984), definiu-se um sistema de manufatura autônomo como sendo aquele em que todas as tomadas de decisões são realizadas automaticamente. Segundo os autores, isso é realizado, principalmente, através da aplicação de tecnologias emergentes como a Robótica e as técnicas de Inteligência Artificial. Assim sendo, possibilita-se a obtenção de sistemas de manufatura apresentando um grau de autonomia bastante razoável. Entretanto, isso pode implicar num aumento considerável da complexidade do controle das operações e também do planejamento da produção. Além disso, com a diminuição dos tempos de "set-up" das máquinas, devido ao desenvolvimento tecnológico dos dispositivos eletro-mecânicos, tem-se tempos cada

vez menores para as tomadas de decisões, responsáveis pela realização do controle [SANTOS F^o (1991)].

Os sistemas baseados no conceito de automação total estão sendo alterados para baseados em antropocentrismo³, onde considera-se a intermediação entre a automação flexível e os seres humanos [ITO (1991)], pois os SIM são baseados na ocorrência de eventos, onde, em geral as condições para a seqüencialização desses eventos são definidas por um operador (ser humano). Desta forma, o objetivo principal da automação é a obtenção de sistemas automáticos e inteligentes.

Portanto, levando-se em conta esse conceito da antropocentria, uma automação eficiente de um ambiente de manufatura somente será viável do ponto de vista técnico se ocorrer as seguintes situações:

- *Padronização dos sinais (comandos) de controle.* Essa padronização é necessária devido a existência de diferentes tipos de informações que percorrem os vários dispositivos, nos diversos níveis de um SIM.
- *Padronização do planejamento da produção de pequenas unidades produtivas adequadas às necessidades do controle em tempo real ("on-line").* É necessária a utilização de alguma técnica que permita a alteração do planejamento de maneira consistente e simplificada.
- *Alto grau de expansibilidade e manutenibilidade de "hardware" e "software" para o sistema de controle.*
- *Boa tolerância a falhas.*

³ O homem é considerado como centro, ou seja, os sistemas são voltados para o homem.

- Mínimo dispêndio de homens-horas para o desenvolvimento e melhoramento do sistema.

Assim sendo, é necessário que o sistema de controle, responsável pelo gerenciamento do SIM, seja eficiente e permita, desta maneira, uma melhoria na produtividade. Isto é, uma redução no tempo de produção, juntamente com um aumento em variedade e/ou quantidade de itens a serem produzidos.

Os avanços constantes da tecnologia de fabricação dos equipamentos que podem ser utilizados em ambientes de manufatura, como por exemplo: os computadores de maior capacidade de processamento e confiabilidade ("workstations" - tecnologia RISC), os robôs e as máquinas CNCs, possibilitam o desenvolvimento de sistemas produtivos de acordo com os conceitos apresentados.

1.3 MOTIVAÇÕES E OBJETIVOS DO TRABALHO

A eficiência citada no item anterior, está diretamente relacionada com a necessidade do aproveitamento das potencialidades fornecidas pelos equipamentos que compõem os SIM, associado a uma alocação eficiente de tarefas nesses ambientes. Deve-se necessariamente, considerar um alto grau de flexibilidade que, por conseqüência, implica num sistema de controle cada vez mais complexo. Por isso é imprescindível, na implementação destes sistemas, a utilização de ferramentas poderosas que permitam uma análise detalhada do comportamento dinâmico, visando a especificação e a implementação de um conjunto de regras úteis para a integração das tarefas do sistema de controle.

Uma metodologia, conhecida como modelagem-simulação [ALLA (1986)] é bastante aplicada para o projeto de sistemas de manufatura. Esta técnica consiste na

modelagem inicial do sistema de manufatura através de algum método conhecido e, em seguida, realiza-se as alterações no modelo que forem necessárias com o objetivo de alcançar-se uma solução ótima ou próxima desta. Este método baseia-se na abordagem "top-down" em que, analisa-se os sistemas complexos estruturando-os de forma hierárquica.

A utilização desta abordagem "top-down", detalhada no capítulo 2, facilita a tarefa de implementação das regras para o controle de sistemas de manufatura complexos e de grande porte, como os SIM. Finalmente, definido o modelo contendo um conjunto de regras, é possível realizar uma simulação e posteriormente, a partir dos resultados, uma análise das propriedades qualitativas e quantitativas do sistema modelado, avaliando-se assim as possíveis soluções encontradas.

Para a implementação de um controle satisfatório e consistente necessita-se da realização de uma análise de eficiência desse controle. Essa análise deve ser realizada, normalmente, fora do ambiente produtivo ou seja, a análise deve ser "off line". A falta de recursos disponíveis e, principalmente, a impossibilidade de interromper um processo de produção em pleno funcionamento, impedem que uma análise para a validação do sistema de controle, seja realizada "on-line"⁴. A simulação e uma posterior análise dos resultados obtidos, permitem a determinação de um conjunto de regras que serão úteis para a implementação do sistema de controle.

Portanto, essa necessidade de avaliação do sistema a partir do modelo adotado, caracteriza a importância da utilização de ferramentas de simulação, que podem ser úteis no auxílio da especificação, implementação e também análise dos sistemas de controle para os SIM.

⁴ Processamento em tempo real no ambiente produtivo.

Neste trabalho, utiliza-se a metodologia MFG/PFS para a modelagem dos SIM que se baseia nas redes de Petri do tipo condição-evento [PETERSON (1991), REISIG (1992)]. Esta metodologia utiliza uma representação gráfica que permite a aplicação de uma abordagem "top-down", sistematizando desta maneira, a modularização de acordo com os processos e atividades existentes nesses ambientes de produção.

Em ambientes integrados de manufatura é comum a ocorrência de situações de conflito, que são caracterizados, principalmente, quando um mesmo recurso pode ser compartilhado por vários elementos ou atividades dentro de um processo produtivo. Um exemplo desse tipo de situação pode ser um ambiente de manufatura onde existem várias estações de trabalho (máquinas CNCs, robôs, FMS, etc.), interligadas por um sistema autônomo de transporte composto de vários veículos ("AGV - Automated Guided Vehicle") [SANTOS F^o (1992)]. Se ocorrer a situação em que várias estações solicitam transporte, a escolha do veículo que atenderá a solicitação e/ou a escolha da estação a ser atendida primeiro, depende das restrições que são inerentes ao processo de produção, como: veículo mais próximo da estação, estação com prioridades para atendimento, etc. Essas restrições evidenciam a necessidade do condicionamento dessa escolha, pois dependendo da seqüência escolhida, o sistema pode entrar em estado de "deadlock". Por essa razão é necessário a utilização de um "árbitro" que soluciona estes conflitos de acordo com as restrições inerentes ao processo produtivo. Este árbitro é responsável pela tomada de decisões em situações de conflito a partir de um conjunto de regras.

Para a análise de um sistema integrado de manufatura, semelhante ao exemplo citado, é necessário a implementação de um modelo do processo produtivo e, associado a este modelo, o modelo do "árbitro" responsável pela solução dos conflitos. Desta forma, a modelagem através do grafo MFG/PFS torna-se complexa e de difícil interpretação, pois têm-se o modelo do processo engajado ao modelo do "árbitro". Por essa razão, propõe-se neste trabalho, um "árbitro" externo, baseado em regras do tipo "IF-THEN",

normalmente utilizadas para representação de conhecimento, associadas à metodologia MFG/PFS. As regras que compõem este árbitro, representam as condições necessárias para a obtenção de soluções para os conflitos.

Utiliza-se neste trabalho o nome Árbitro Externo para este árbitro, porque é um elemento (módulo) que não faz parte do modelo em grafos MFG/PFS do sistema, ou seja ele é um elemento externo ao modelo (Figura 5.2).

Na ferramenta proposta (capítulo 4), esses conflitos são resolvidos através da aplicação do conceito de "árbitro" associado a um outro conceito chamado "jogador de marcas", responsável pela realização dos disparos das transições habilitadas. A implementação destes conceitos possibilitam que, a partir das restrições aplicadas na tomada de decisões nos conflitos, se escolha o caminho a ser seguido. Essas restrições necessárias para definir-se uma determinada tomada de decisão, são representadas através de um conjunto de regras e a avaliação (inferência) das regras é realizada através de um mecanismo de interpretação ou uma máquina de inferência, que realiza a verificação de um conjunto de regras fornecendo, em seguida, uma solução.

Nexte contexto, a grande virtude desta ferramenta é possibilitar a detecção e solução de "deadlocks". Propõe-se a aplicação de um método analítico (análise de invariantes [REISIG (1982)]) para detectar os "deadlocks" estruturais e para a solucioná-los, altera-se a estrutura do grafo a partir dos dados fornecidos pela análise dos invariantes do modelo. A simulação dos estados atingíveis pelo sistema é utilizada para a detecção de "deadlocks" condicionais e a solução é realizada através da alteração da marcação inicial e/ou do conjunto de regras que definem as condições de disparo numa situação de conflito.

1.4 ORGANIZAÇÃO DO TRABALHO

A seguir apresenta-se uma descrição resumida do conteúdo dos capítulos e apêndices que compõem o trabalho.

Capítulo 1. Introdução. Este capítulo contém uma breve explanação sobre a necessidade da automação e a utilização de sistemas autônomos aplicados a sistemas de manufatura. Apresenta-se também, as motivações e objetivos e a organização do trabalho.

Capítulo 2. Modelagem de Sistemas de Manufatura. Apresenta-se neste capítulo uma conceituação de Sistemas Integrados de Manufatura (SIM), as principais características das redes de Petri, a utilização de uma abordagem hierárquica e estruturada através da metodologia MFG/PFS e uma descrição das características básicas dos sistemas baseado em regras.

Capítulo 3. Análise dos Sistemas Integrados de Manufatura. Este capítulo identifica as necessidades para a realização da análise dos SIM. Apresenta-se alguns fenômenos indesejáveis que surgem nos ambientes integrados de manufatura e as possíveis soluções, uma descrição de algumas técnicas utilizadas para a análise e finalmente, uma limitação quando se utiliza os grafos MFG.

Capítulo 4. Ferramenta de simulação e análise. Apresenta-se neste capítulo a especificação da ferramenta proposta, detalhando-se a sua estrutura básica, a descrição dos módulos principais, a estrutura de dados, os algoritmos e alguns detalhes relevantes de implementação e o seu funcionamento.

Capítulo 5. Exemplo de aplicação da ferramenta. Neste capítulo é apresentado uma modelagem em MFG/PFS de um sistema de manufatura e a associação com o árbitro

externo baseado em regras. O exemplo corresponde a um ambiente de produção do tipo "flow shop". Além disso, como comparação, apresenta-se o modelo do árbitro através dos grafos MFG/PFS e as regras necessárias para a implementação deste árbitro externo baseado em regras.

Capítulo 6. Comentários finais. Neste capítulo apresenta-se algumas observações finais mostrando a contribuição do trabalho, suas vantagens e também as suas limitações. Neste contexto, sugere-se também algumas propostas para possíveis melhorias na ferramenta proposta.

Apêndice. Neste apêndice apresenta-se a formalização, conceitos básicos e algumas propriedades mais importantes das redes de Petri.

Capítulo 2

MODELAGEM DE SISTEMAS DE MANUFATURA

2.1 INTRODUÇÃO

Seguindo os conceitos de sistemas antropocêntricos propostos por ITO (1991), verifica-se que há uma tendência natural de utilização de técnicas gráficas para a modelagem estrutural e também do comportamento dinâmico dos Sistemas a Eventos Discretos (SED), como os Sistemas Integrados de Manufatura (SIM). Assim sendo, descreve-se algumas características das redes de Petri e a sua aplicação para modelagem de sistemas de manufatura.

Apresenta-se também a metodologia MFG/PFS, que utiliza os grafos MFG e uma descrição estruturada e hierárquica através dos grafos PFS. E, no final deste capítulo, faz-se uma descrição das características básicas dos sistemas baseados em regras que é utilizado neste trabalho, para auxiliar a solução de conflitos que ocorrem em ambientes de manufatura.

2.2 SISTEMAS INTEGRADOS DE MANUFATURA (SIM)

Os Sistemas Integrados de Manufatura são sistemas flexíveis e automatizados, compostos por sub-sistemas com diferentes graus de autonomia e controlados por sistemas computacionais complexos, que podem ser aplicados em ambientes fabris onde é considerada fundamental a integração do homem com as máquinas. Este tipo de integração é muito importante, pois está relacionado com os conceitos básicos de autonomia e de integração máquina-homem-máquina (sistemas Antropocêntricos) [ITO (1991)].

Numa configuração para os SIM a tendência é utilizar-se os sub-sistemas autônomos que possuem decisões locais (controle distribuído). Alguns dos elementos que podem compor um SIM são:

- dispositivos de controle computadorizados,
- sub-sistemas autônomos de transporte e manipulação de ferramentas,
- células de transformação baseadas em máquinas CNC e,
- sistemas para controle de fluxo de informações que permitem a comunicação entre diferentes entidades.

A Figura 2.1 mostra uma possível implementação de SIM. Nesta Figura verifica-se a grande quantidade de elementos de diferentes tipos, acarretando, desta maneira, um fluxo complexo de informações.

Considerando os SIM, sistemas do tipo condição/evento e pertencentes a uma classe de SED, existe intrinsecamente nestes sistemas as características de paralelismo, conflito e assincronismo. Neste caso, o paralelismo corresponde a ocorrência de dois eventos independentemente, o conflito surge quando uma mesma condição esta associada

a ocorrência de vários eventos simultaneamente e o assincronismo está relacionado com a não dependência do tempo, para a ocorrência de eventos, mas sim de condições estabelecidas.

Desta maneira, o comportamento dinâmico dos SIM é governado através da ocorrência de eventos, em instantes discretos e aleatórios que determinam as mudanças de estado do sistema. Ao contrário dos Sistemas à Variáveis Contínuas (SVCs) [SANTOS F^o. (1991)], que são governados por leis da física invariantes cuja descrição e análise podem ser feitas através de uma função matemática, os eventos dentro dos SIM, ocorrem em intervalos de tempo discretos, não necessariamente regulares e que não permitem a análise através de modelos matemáticos baseados apenas em equações diferenciais e integrais.

Para este tipo de sistema, está sendo estudada uma proposta para modelagem, de forma eficiente, de suas características. Além disso, é necessário uma ferramenta que permita a descrição da hierarquia e da dinâmica dos sistemas desta natureza. Neste contexto, a rede de Petri é uma ferramenta poderosa de modelagem, pois permite uma representação clara e precisa das características encontradas nos SIM.

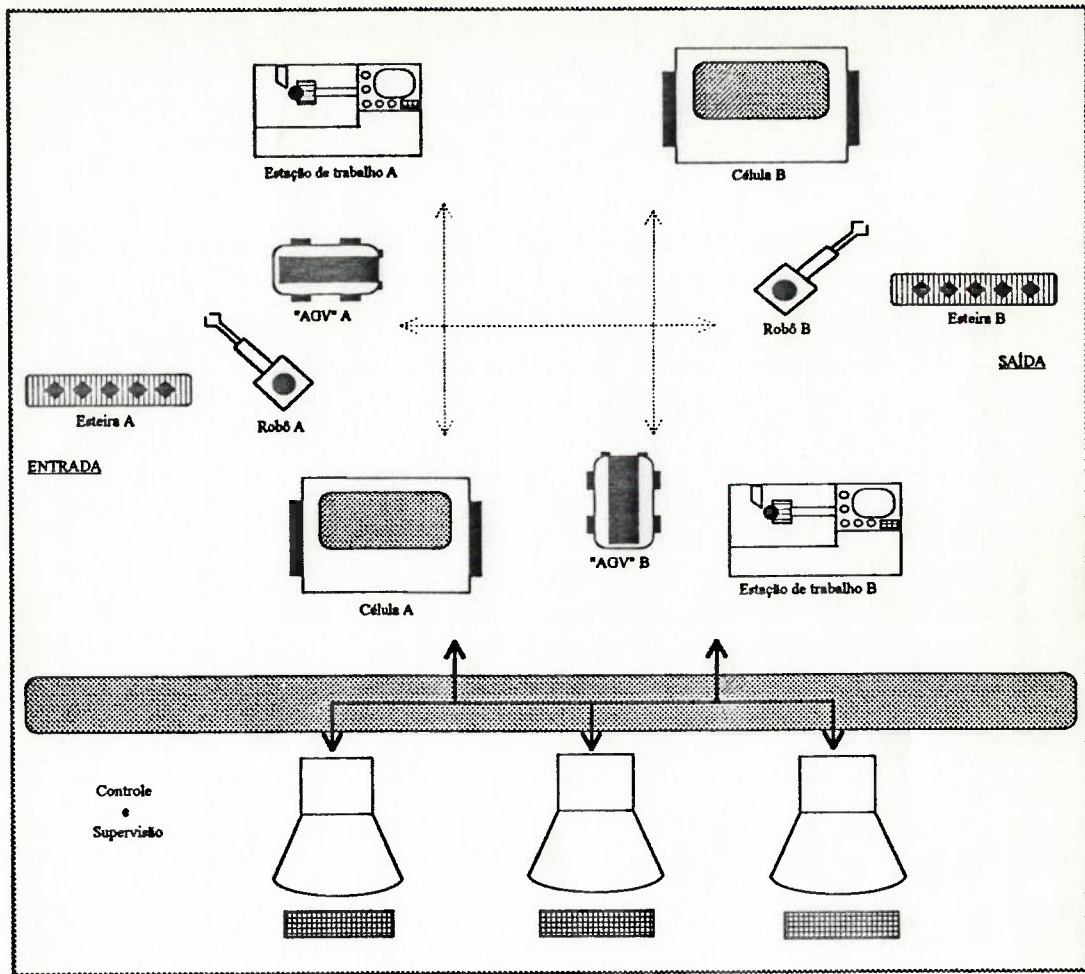


Fig. 2.1 Exemplo de um SIM.

2.3 MODELAGEM UTILIZANDO REDES DE PETRI

As redes de Petri, cujo desenvolvimento formal básico é apresentado no apêndice, são baseadas em grafos bipartidos⁵ [SZWARCFITER (1986)]. O formalismo matemático destas redes já é bastante conhecida [MURATA (1989), PETERSON (1991), REISIG (1982)]. Essas redes permitem uma representação sistemática, clara e uniforme das

⁵ Em Redes de Petri, corresponde a separação dos elementos em dois conjuntos: um de lugares e outro de transições, onde, existe ligação que une um elemento do conjunto de lugares a um elemento do conjunto transições, ou seja, não há relação entre elementos do mesmo conjunto.

especificações funcionais, inerentes ao comportamento dinâmico, dos sistemas baseados em eventos, como os SIM.

Os conceitos básicos das redes de Petri, foram criados originalmente por Carl Adam Petri na sua dissertação de doutorado submetido em 1962 à Universidade de Darmstadt, Alemanha.

Devido a sua generalidade e versatilidade, as redes de Petri podem ser aplicadas em uma variedade de áreas como: sistemas distribuídos, protocolos de comunicação, programação paralela e sistemas de manufatura. Utiliza-se estas redes, principalmente, em sistemas que podem ser representados graficamente e necessitam de representações para atividades conflitantes e/ou paralelas [MURATA (1989)].

A necessidade de obtenção de sistemas cada vez mais interativos e de fácil interpretação pelo ser humano (antropocentrismo) viabiliza a utilização da rede de Petri. Pois, para sistemas não muito complexos, a rede permite uma fácil interpretação e visualização estrutural do sistema em estudo.

A descrição dos SIM através dessas redes baseia-se na aplicação de conceitos de condição e evento, uma vez que estes sistemas são baseados nestes conceitos. Neste caso, os eventos correspondem às ações ou atividades que ocorrem no sistema e as condições ao estado que o sistema pode atingir. A Figura 2.2 apresenta um sistema de manufatura composto por uma estação de processamento e um robô que faz a carga e descarga de peças da esteira de entrada e da esteira de saída. A Figura 2.3. mostra a modelagem, em rede de Petri, da operação de carregamento da peça da esteira de entrada para a estação de processamento.

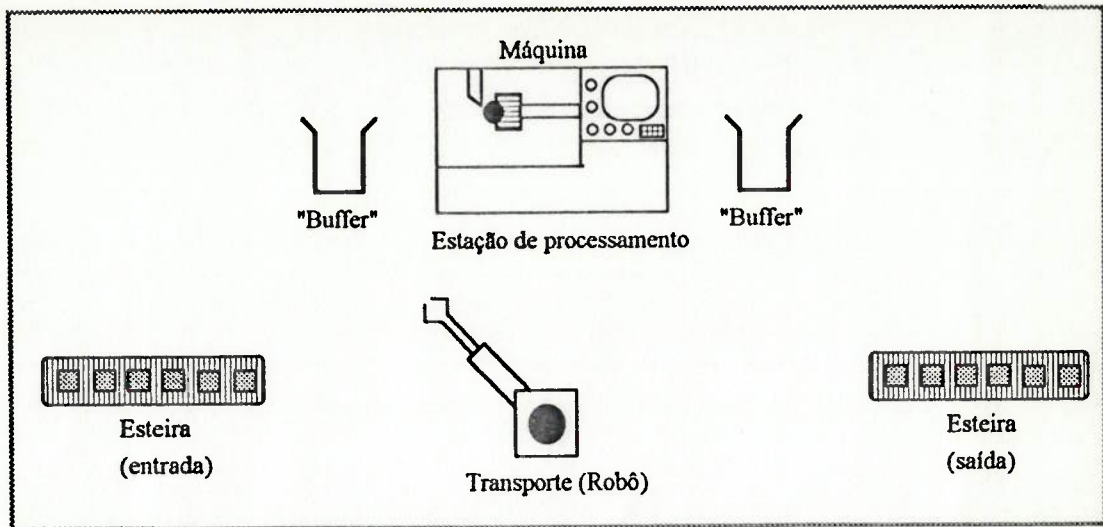


Fig. 2.2 Exemplo: carga e descarga de peças de uma estação de processamento através de um robô.

Na Figura 2.3, além da representação do modelo em rede de Petri da operação de carregamento, mostra-se também os símbolos gráficos que representam os elementos da rede de Petri. Estes elementos básicos são:

- um círculo que representa uma condição (lugar),
- uma barra representando um evento (transição),
- uma seta representando a associação entre eventos e condições e vice-versa (arco direcionado) e;
- um círculo negro que indica a manutenção da condição (marca).

A dinâmica da rede de Petri (movimentação das marcas) ocorre através do disparo das transições. E, para a viabilização da ocorrência destes disparos, necessita-se que as seguintes regras básicas sejam obedecidas:

- uma transição está habilitada para disparo se todos os lugares de entrada estão marcados,

- com o disparo da transição habilitada, retira-se as marcas dos lugares de entrada e gera-se as marcas nos lugares de saída,
- o disparo de uma transição é instantânea, ou seja, a retirada de marcas dos lugares de entrada ocorre simultaneamente com a geração de marcas nos lugares de saída.

Desta maneira, uma marca num lugar, indica que essa condição é verdadeira e, enquanto uma marca permanece neste lugar significa que a condição a ela associada está sendo mantida.

O arranjo de marcas numa rede de Petri, num determinado instante, corresponde ao estado do sistema naquele instante. No caso da Figura 2.3, as marcas indicadas nos lugares **esteira** e **b1** correspondem a marcação inicial (estado inicial do sistema). Ou seja, indicam respectivamente, uma peça na esteira pronta e o robô disponível e pronto para pegar a peça da esteira.

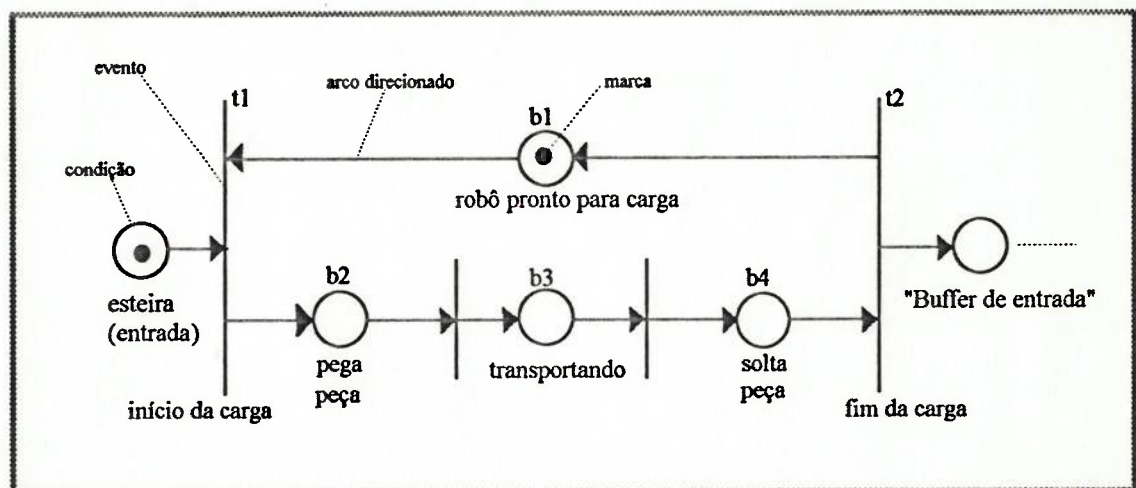


Fig. 2.3 Modelagem da operação "carregamento", através da rede de Petri.

No modelo da Figura 2.3, a transição $t1$ está habilitada. O disparo desta transição representa o início da operação de carga pelo robô. Desta maneira, as marcas em esteira e $b1$ são retiradas e no lugar $b2$ é colocada uma marca (Figura 2.4).

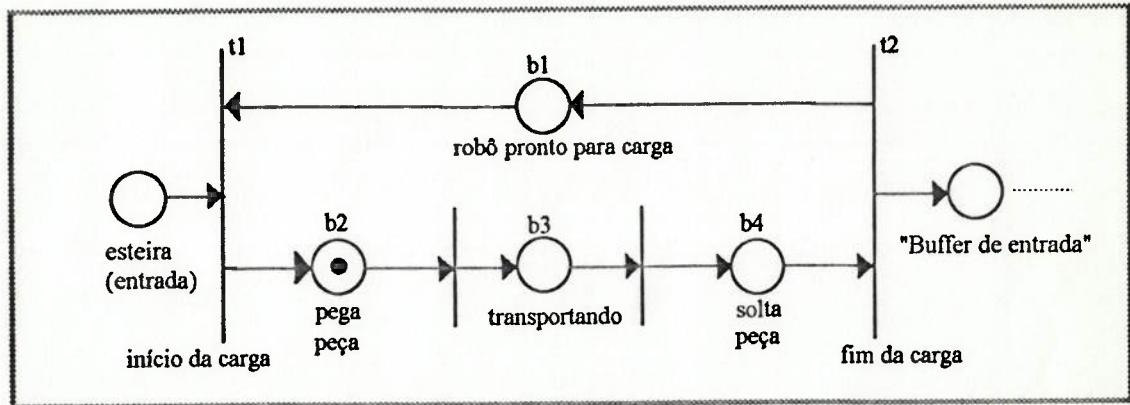


Fig. 2.4 Disparo da transição $t1$.

2.3.1 Propriedades e características das redes de Petri

Algumas propriedades são fundamentais para a descrição das características básicas das redes de Petri como [MURATA (1989)]:

- limitabilidade ("boundedness"). Relaciona-se com o número de marcas em cada lugar da rede. Neste caso, se uma rede é 1-limitado, ou seja, permite no máximo apenas uma marca nos lugares, então diz-se que esta rede é segura ("safe").
- vivacidade ("liveness"). Este conceito relaciona-se com a inexistência do estado de "deadlock", ou seja, todas as transições podem disparar.
- reversibilidade ("reversibility"). Uma rede de Petri é dita reversível, se a partir de uma marcação qualquer (estado qualquer) ela pode retornar à marcação inicial (estado inicial do sistema).

Desta forma, algumas características importantes para a representação de sistemas a Eventos Discretos (SED), através de redes de Petri são: o paralelismo, o assincronismo, o conflito e a possibilidade de utilizar o conceito de hierarquia.

O **paralelismo** ocorre quando não existe a necessidade de sincronizar ações associadas com as entidades do sistema, ou seja, se dois eventos estão habilitados para ocorrer e não há uma interação entre eles, então são eventos paralelos, podendo ocorrer independentemente [MURATA (1989)].

Em Redes de Petri é possível a representação desta característica. Desta forma, essas redes podem ser ideais para a modelagem de sistemas de controle distribuído, que contém múltiplos processos ocorrendo simultaneamente e independentemente. Na representação da Figura 2.5 em rede de Petri, verifica-se que dois eventos são paralelos quando não possuem pré-condições ou pós-condições⁶ em comum, ou seja, a ocorrência do evento **e1**, não influe nas condições para a ocorrência do evento **e2**.

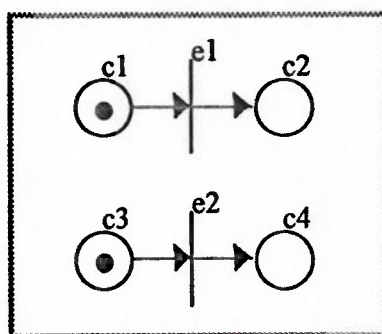


Fig.2.5 Representação de paralelismo em rede de Petri.

O **assincronismo** nas redes de Petri ocorre devido a não existência de uma medida ou de dimensão de tempo inerente a essas redes. Ou seja, os eventos (disparos de

⁶ Pré-condições e Pós-condições são lugares de entrada e lugares de saída de uma transição, respectivamente.

transições) ocorrem instantaneamente sem a utilização do conceito de "clock" para dimensionar o tempo de execução dos eventos. Assim, para a ocorrência dos eventos, basta que as condições associadas sejam satisfeitas, sem a dependência do tempo.

Segundo FREEDMAN (1991), não podemos estudar o paralelismo sem ter alguma noção da duração das operações. Entretanto, nas pesquisas sobre as redes de Petri, o conceito de tempo foi introduzido com o desenvolvimento de novas extensões sobre a rede de Petri original relacionados com a simulação de processos.

Os conflitos, que surgem quando vários eventos ficam habilitados para ocorrer devido às mesmas condições e, na ocorrência de um dos eventos, os outros são desabilitados. Na Figura 2.6 representa-se dois tipos de conflito que podem ocorrer numa rede do tipo Condição/Evento (C/E):

- a) os eventos **e1** e **e2** estão habilitados por uma mesma pré-condição e,
- b) os eventos **e1** e **e2** estão habilitados por uma mesma pós-condição. Neste caso, o lugar de saída tem a capacidade de 1 marca (rede C/E), caso contrário não existe o conflito.

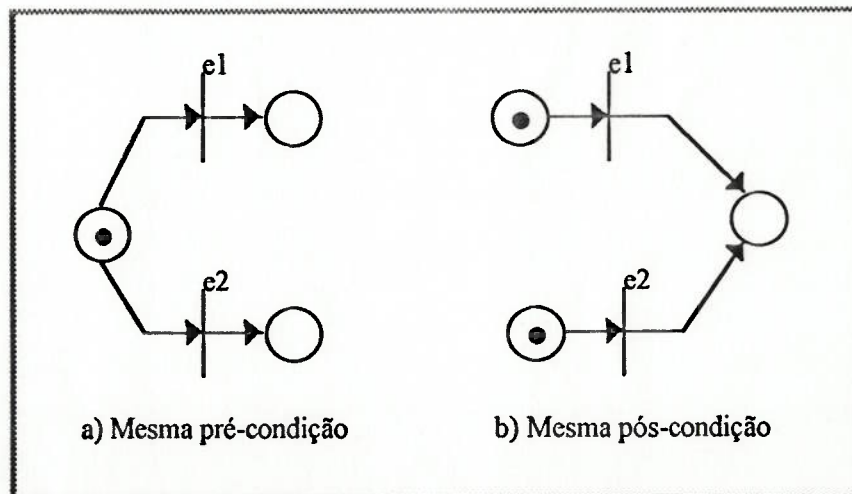


Fig. 2.6 Representação de conflito em rede de Petri.

A descrição hierárquica possibilita a representação e detalhamento dos vários níveis que podem ocorrer dentro de um sistema de manufatura. Existem algumas pesquisas sendo realizadas nesta área. Por exemplo, em SUZUKI e MURATA apud JENG (1993), foi apresentado um método de refinamento nos lugares, e provou-se que este método preserva as propriedades de limitabilidade e vivacidade das redes, depois de cada refinamento. Essa característica de hierarquia é discutida com maiores detalhes no item 2.4 deste capítulo.

Além dessas características citadas, as redes de Petri permitem: uma visualização do sistema através da sua representação gráfica, uma efetiva análise qualitativa através do modelo do sistema e uma forma consistente de associar algumas propriedades das redes (vivacidade e limitabilidade) com alguns critérios para medida de desempenho dos sistemas reais ("deadlock" e colisão).

Entretanto, apesar dessas vantagens oferecidas, existem também algumas limitações quando se utiliza determinadas classes existentes de redes de Petri, para a modelagem dos SIM.

Uma limitação importante da rede de Petri original, é dificuldade de se representar, explicitamente, o fluxo de recursos de diferentes itens (materiais) e de informações através dessas redes. Além disso, o modelo gerado pela rede de Petri original, tende a crescer muito quando o sistema em estudo é complexo e de grande porte. Reduzindo, desta maneira, a sua legibilidade e manuseabilidade.

Para suprir estas limitações, foram propostas outras redes baseadas na rede original (extensões de rede de Petri). Por exemplo, a rede de Petri colorida [ALLA (1986), JENSEN (1986)], que permite atributos nas marcas (possibilita fluxo de diferentes itens), a rede de Petri temporizada [FREEDMAN (1991)] que inclui o conceito de tempo nos seus elementos e as redes que permitem o projeto e desenvolvimento de

sistemas utilizando a abordagem hierárquica e estruturada destes sistemas (metodologia MFG/PFS) [MIYAGI (1988)].

2.4 DESENVOLVIMENTO HIERÁRQUICO

O aumento constante da complexidade dos sistemas devido, principalmente, a evolução tecnológica dos equipamentos, mostra a necessidade de simplificar ou facilitar os métodos empregados para o projeto e análise destes sistemas. A utilização de técnicas de estruturação e hierarquização permite esta simplificação desejada.

O desenvolvimento hierárquico de sistemas, também conhecido como desenvolvimento através da abordagem "top-down", é uma técnica (metodologia) muito utilizada na especificação e implementação de sistemas complexos e de grande porte. Neste caso, esses sistemas são divididos em sistemas menores, sucessivamente, até que sejam obtidos subsistemas mais simples em que a interpretação e a consequente implementação, seja facilitada. Este mesmo tipo de abordagem é muito utilizada para o desenvolvimento de sistemas de "software" de grande porte. O detalhamento é feito até chegar num nível onde a implementação desses pequenos módulos é feita utilizando alguma linguagem de programação (desenvolvimento por refinamento sucessivos).

No seu trabalho sobre simulação e análise de FMS ("Flexible Manufacturing System") baseado na abordagem hierárquica, VALAVANIS (1990) afirma que as redes de Petri possuem a capacidade de modelar sistemas de manufatura de maneira hierárquica. Isto só ocorre quando é possível a substituição de uma rede completa ou trechos dessa rede por um lugar ("place") ou transição ("transition"). Ou seja, este tipo de substituição permite o modelamento dos vários níveis funcionais existentes num sistema de manufatura. Neste caso, foi utilizado outro tipo de rede denominada *rede de Petri estendida*.

Outras pesquisas estão sendo realizadas, com o objetivo de obtenção de novos "teoremas" que estabelecem a preservação das propriedades como a limitabilidade, a segurança, a reversibilidade e a vivacidade das redes, após a aplicação da abordagem por refinamento sucessivos de transições e lugares. Esta abordagem permite tanto a obtenção de redes de Petri maiores como também a redução das redes para facilitar a tarefa de análise das suas propriedades [ZHOU (1989)].

Nos sistemas integrados de manufatura, pode-se identificar diferentes níveis de funcionalidade, sugerindo assim a utilização de níveis hierárquicos para a modelagem da sua estrutura. Por essa razão, utiliza-se neste trabalho, uma metodologia baseada na abordagem "top-down", para especificação e implementação desses sistemas. A metodologia MFG/PFS, apresentada a seguir, permite uma descrição dos diferentes níveis contidos nesses sistemas, através de elementos da sua estrutura que podem representar essa hierarquia de funcionalidade.

2.5 METODOLOGIA MFG/PFS

Verifica-se que a rede de Petri cujas características foram descritas no item 2.3, é uma ferramenta bastante eficiente para a modelagem de sistemas. Desta forma, através de sua base matemática, é possível o desenvolvimento de técnicas e ferramentas consistentes e poderosas para análise. Porém existe a dificuldade, quando se utiliza a rede de Petri para representação dos sistemas de grande porte conectados com sub-sistemas em diferentes níveis hierárquicos. Neste caso, o modelo torna-se complexo, de difícil concepção, visualização e interpretação.

A metodologia MFG ("Mark Flow Graph")/PFS ("Production Flow Schema") [MASUDA (1980), MIYAGI (1988), MIYAGI (1989), SANTOS F^o. (1992)] foi proposta com o objetivo de obter uma ferramenta adequada para a modelagem, análise e

controle de sistemas integrados de manufatura. A metodologia MFG/PFS permite uma descrição do comportamento dinâmico do sistema e possui um modelo matemático preciso devido às características e as propriedades herdadas das redes de Petri.

Esta metodologia consiste numa modelagem inicial através de um modelo conceitual (PFS) e em seguida detalha-se este modelo até a obtenção de um modelo funcional (MFG) que corresponde num SIM o nível de dispositivos e máquinas. Dependendo do sistema em estudo e do nível que se quer representar, permite-se uma representação híbrida, ou seja, com elementos MFG e PFS.

No item a seguir apresenta-se as principais características e elementos básicos utilizados na metodologia MFG/PFS.

2.5.1 "Mark Flow Graph" (MFG)

O projeto e análise dos Sistemas Integrados de Manufatura (SIM) deve ser realizada através de uma técnica de representação que descreva corretamente as características do sistema, permitindo uma descrição tanto no nível conceitual como no nível detalhado e que seja interativa para facilitar a troca de informações entre o ser humano (operadores, projetistas, etc) e o sistema. E, além disso, deve permitir a utilização de modelos matemáticos eficientes.

Como visto anteriormente, os SIM possuem características de assincronismo, conflito e paralelismo que podem provocar fenômenos do tipo:

- Colisão: fenômeno que ocorre quando, apesar de uma certa condição estar sendo mantida, o evento imediatamente anterior ocorre gerando uma informação redundante.

- "Deadlock": fenômeno que ocorre quando, eventos e condições se combinam de forma que novos eventos não possam mais ocorrer.

Portanto, o modelo que representa o sistema deve possuir as seguintes propriedades e funções:

- a) garantia de que o sistema é seguro ("safe") não permitindo, então a ocorrência de colisões;
- b) capacidade de enviar sinais para dispositivos externos;
- c) capacidade de receber sinais de dispositivos externos e a partir destes sinais controlar a ocorrência dos eventos no modelo.

Assim sendo, MFG é um grafo bipartido, baseado na rede de Petri do tipo condição-evento, de onde, herda as principais características, propriedades e a capacidade de análise e modelagem de sistemas. Além disso, é uma rede segura, pois permite apenas uma marca no elemento "box". A Figura 2.8 mostra um "box" do MFG e a respectiva equivalência em rede de Petri e a Figura 2.7 apresenta os elementos estruturais básicos do MFG e as suas inter-relações [MIYAGI, 1989].

Os elementos do MFG são:

- . **"Box"**: indica uma condição ou operação associada a um elemento do sistema a ser modelado. É representado por um retângulo.
- . **Transição**: indica um evento no sistema. O disparo de uma transição permite a mudança do estado do sistema. É representado por uma barra vertical.
- . **Arco orientado**: conecta "boxes" e transições de forma a indicar a relação entre condições e eventos. Este arco pode ser denominado arco de saída quando "sai" de um "box" ou de uma transição e arco de entrada quando "entra" no "box" ou transição. É representado por uma seta.

. **Marca:** indica a manutenção de uma condição. O arranjo de marcas no grafo corresponde a uma marcação. Este arranjo define o estado do sistema num determinado instante. É representada por um círculo de cor preta no interior do "box".

. **Porta ("Gate")⁷:** Habilita ou inibe a ocorrência dos eventos correspondentes às transições sendo portanto denominada porta habilitadora ou porta inibidora. A porta habilitadora é representada por um círculo de cor preta na extremidade do arco conectado à transição. A porta inibidora é representada por um círculo transparente (cor branca) na extremidade do arco conectado à transição. Numa porta habilitadora, se o "box" a ela associado estiver marcado, a transição no qual ela está conectada é habilitada. Em contrapartida, se o "box" associado a uma porta inibidora estiver marcada, ela inibe a transição à ela conectada. Se o sinal para a porta for proveniente de um dispositivo externo que não faz parte do modelo, a porta é denominada externa.

. **Arco de sinal de saída.** Envia uma informação do box a ele interligado para um elemento externo.

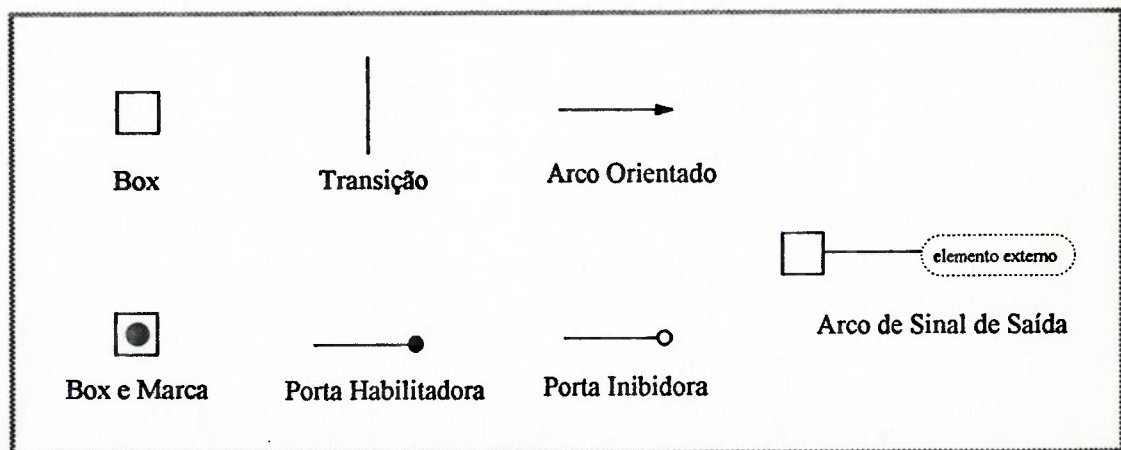


Fig. 2.7 Elementos básicos do MFG.

⁷ Arco de disparo ou gatilho.

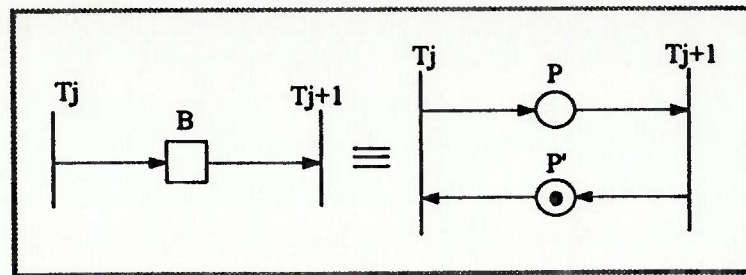


Fig. 2.8 "box" do MFG e a equivalência em rede de Petri.

2.5.2 Regras de disparo do MFG

De maneira semelhante às redes de Petri, a marcação dos "boxes", num grafo MFG, representa o estado do sistema num determinado instante. Portanto, o disparo das transições e a conseqüente mudança no arranjo das marcas permite a representação do comportamento dinâmico do sistema em estudo.

O estado inicial do sistema é representado através de uma marcação inicial dos "boxes". E, sendo uma rede que possui a propriedade de segurança ("safeness"), permite-se apenas uma marca nos "boxes".

O disparo de uma transição numa rede MFG corresponde a ocorrência de um evento, ou seja, é semelhante ao disparo de uma transição numa rede de Petri condição-evento⁸. Uma transição no MFG está habilitada para disparar se forem satisfeitas todas as seguintes regras:

1. todos os "boxes" de entrada desta transição devem ser marcados,
2. todos os "boxes" de saída desta transição não devem estar marcados,

⁸ Numa rede de Petri do tipo condição/evento, o evento só ocorrerá se todas as suas pré-condições estiverem satisfeitas e suas pós-condições não satisfeitas.

3. as portas associadas a esta transição devem estar com sinais adequados, ou seja, caso seja uma porta habilitadora, o "box" a ela conectada deve estar marcado e se for porta inibidora este "box" não deve estar marcado, e

4. se existir porta externa ela deve estar no estado habilitado se for uma porta habilitadora e desabilitado se for uma porta inibidora.

Quando uma transição está habilitada, ela dispara imediatamente, pois, o tempo de disparo desta transição é instantâneo.

O disparo da transição resulta na retirada das marcas dos "boxes" de entrada e a consequente colocação de marcas nos "boxes" de saída. A Figura 2.9 mostra como ocorre o disparo de uma transição no MFG. De acordo com essa Figura, após o disparo da transição *t*, as marcas contidas nos boxes *b1* e *b2* desaparecem e são colocadas marcas nos boxes *b4* e *b5*. O disparo da transição *t* não afeta o estado dos "boxes" *b3* e *b6*, porque estes "boxes" estão associados à transição *t* através de uma porta habilitadora. Ou seja, a marcação nos "boxes" associados a transição através de portas, não é alterada com o disparo desta transição.

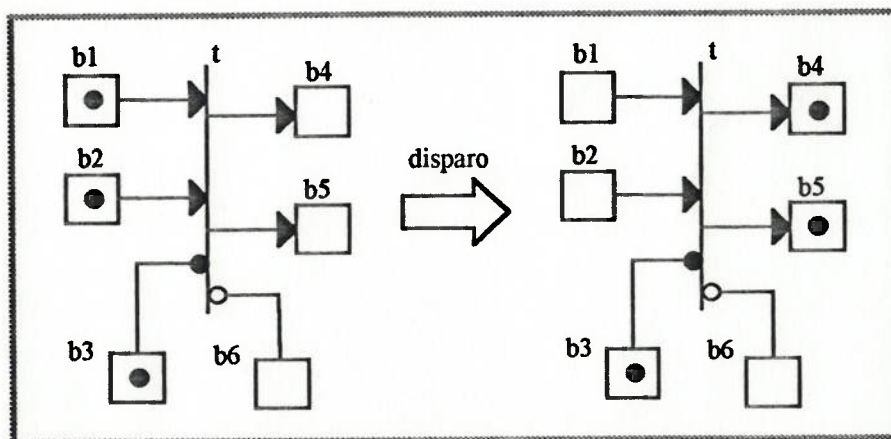


Fig. 2.9 Disparo de uma transição no MFG.

A evolução de marcas sobre o grafo MFG possibilita, desta maneira, a representação interativa do comportamento dinâmico do sistema.

2.5.3 Macro elementos do MFG

A representação de SIM pode ser feita através dos elementos básicos do MFG, entretanto, para sistemas maiores e mais complexos, a interpretação do modelo torna-se muito difícil. Desta forma, a partir da verificação da funcionalidade de elementos que compõem frequentemente os sistemas de manufatura, e com o intuito de simplificar a representação do modelo, foi proposto macro elementos a partir dos elementos estruturais básicos do MFG [HASEGAWA (1987)].

Os principais macro elementos do MFG são:

- **"box" Capacidade** que é utilizado para representar um elemento armazenador de no máximo N marcas. A Figura 2.10 mostra a sua representação e a correspondência através dos elementos básicos.

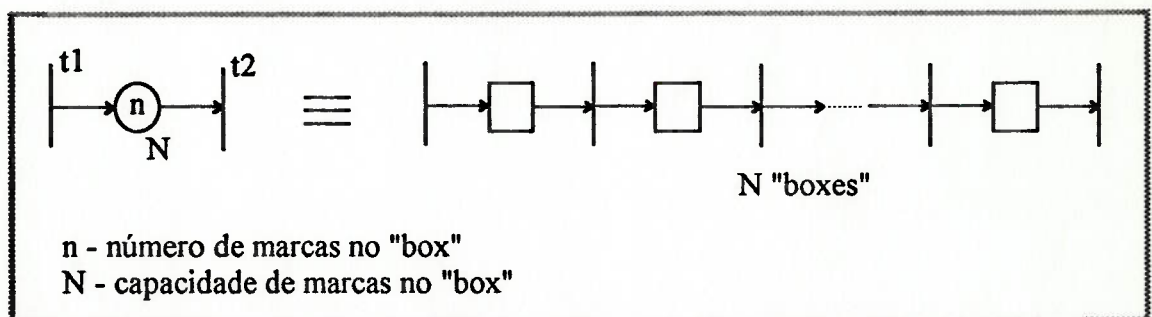


Fig. 2.10 "box" Capacidade.

Para o "box" capacidade da Figura 2.10, a condição necessária para se disparar a transição t1 é $n < N$ e para disparar t2 $n > 0$. O disparo de t1 implica na entrada de 1 marca e o disparo de t2 a retirada de 1 marca.

- "box" Agrupador que pode representar um agrupamento de peças (montagem). De acordo com a Figura 2.11, onde tem-se a representação deste "box" e a respectiva implementação através dos elementos básicos do MFG. Para este tipo de "box", entram N marcas e sai apenas 1 marca. Neste caso, a transição t1 de entrada do "box" dispara se $n < N$ e a transição t2 de saída só estará habilitado para disparar se houver N marcas no "box". E, quando t2 dispara, será gerado na saída apenas 1 marca que corresponde ao agrupamento das N marcas que entraram no "box" uma a uma.

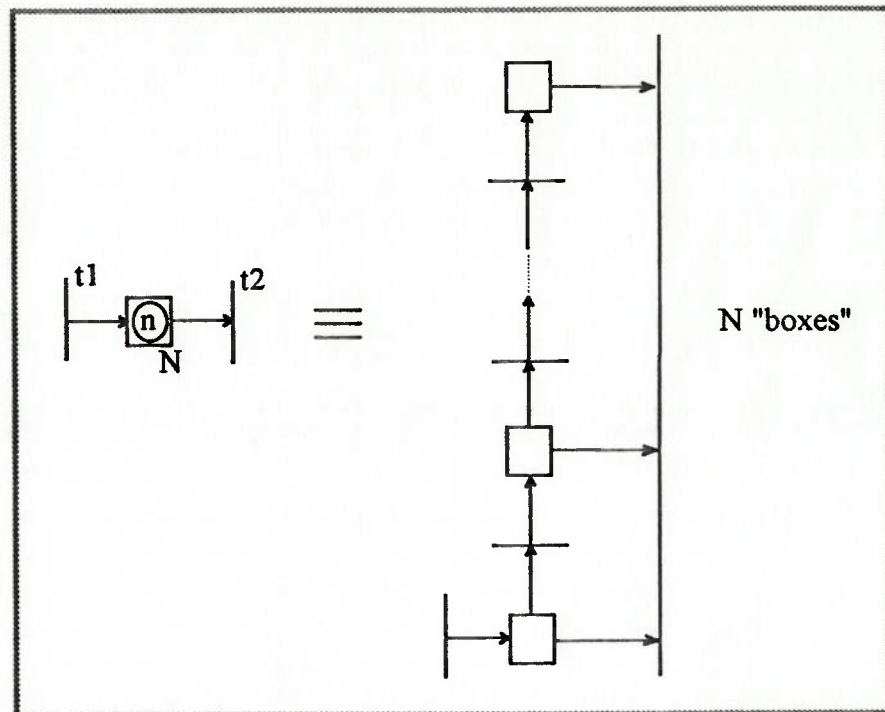


Fig. 2.11 "box" Agrupador.

- "box" Dispensor representa o processo inverso ao agrupador (desmontagem). Ou seja, a transição de entrada t1 só dispara se não existir nenhuma marca no "box" e t2 dispara se existir alguma marca no "box". Neste tipo de "box", entra

apenas 1 marca e saem N marcas. A Figura 2.12 mostra a representação deste "box" e a sua formação através dos elementos básicos do MFG.

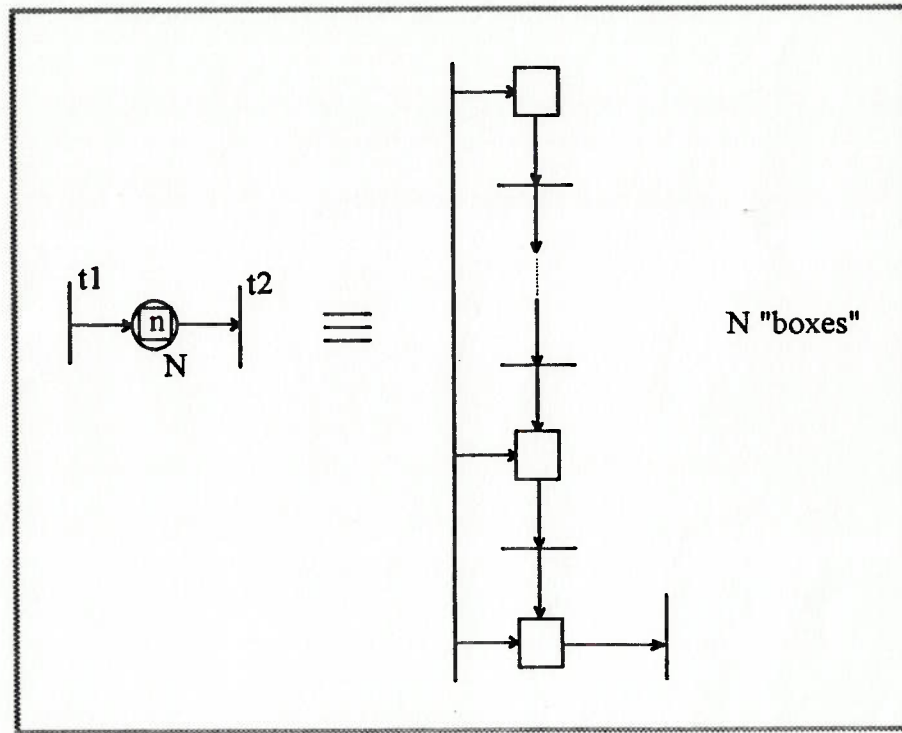


Fig. 2.12 "box" Dispensor.

Em sistemas reais, o conceito de tempo é imprescindível, por isso, foi introduzido no MFG os elementos que consideram o tempo para a representação dos conceitos de manutenção de estados. Este elementos descritos a seguir e representados na Figura 2.13 são utilizados para simulação e análise dos sistemas modelados através do MFG/PFS.

- **"box" temporizado:** a marca permanece neste "box" após decorrido o tempo (t_B) a ele associado.
- **transição temporizada:** se a transição esta habilitada, ela só dispara após decorrido o tempo (t_T) a ela associada.

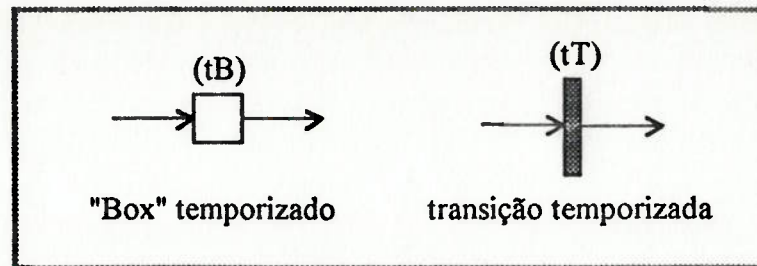


Fig. 2.13 Elementos temporizados do MFG.

2.5.4 "Production Flow Schema" (PFS)

O MFG permite uma representação num nível de detalhes próximo dos dispositivos e máquinas. Fazendo uma analogia com sistemas computacionais, modelar em MFG corresponde a programar utilizando a linguagem Assembler. Desta maneira, mesmo utilizando-se dos macro elementos do MFG, o modelo de sistemas complexos continuam de difícil compreensão. Porém, verifica-se que estes sistemas podem ser caracterizados por atividades que podem ser organizados hierarquicamente e de forma estruturada.

Assim sendo, foi proposto o "Production Flow Schema" (PFS) [MIYAGI, 1988] que fornece ao MFG uma regra para interpretação dos elementos do grafo e permite a utilização de uma abordagem "top-down" para o desenvolvimento dos modelos.

O PFS é utilizado para a representação dos níveis conceituais das atividades ou ações que ocorrem em um sistema a eventos discretos como os SIM, ou seja, num nível mais alto de abstração. Esta representação através do PFS é baseado no fluxo de itens (materiais e informações) através de seus subsistemas, onde efetivamente são executadas as operações produtivas.

Através do PFS, verifica-se que estas operações produtivas podem ser decompostas em três elementos básicos:

- **Atividade:** elemento ativo. Corresponde ao próprio processo produtivo. Neste elemento a atividade é iniciada com a chegada de itens especificados e termina com a produção de itens na saída (Figura 2.14).
- **Elemento distribuidor:** elemento passivo. Armazena os itens que saem dos elementos ativos e os distribuem para os elementos subsequentes (Figura 2.15).
- **Arcos direcionados:** representam as relações entre as atividades e os elementos distribuidores.

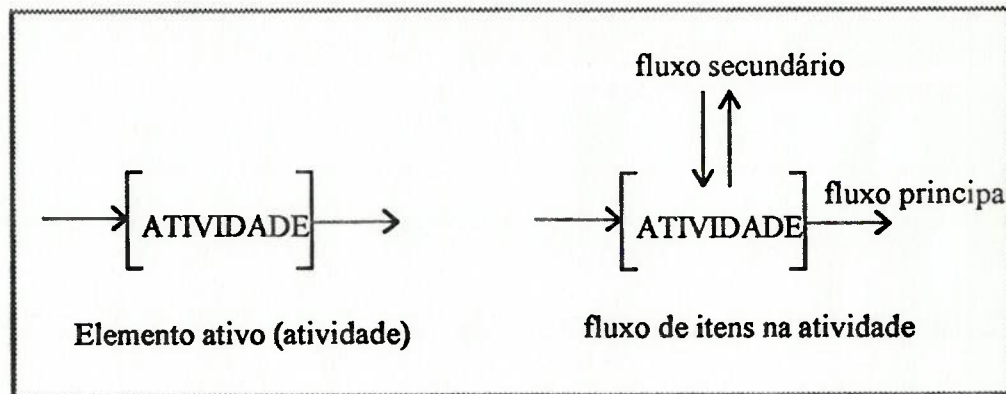


Fig. 2.14 Atividade no PFS.

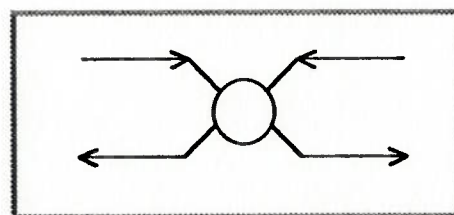


Fig. 2.15 Elemento distribuidor PFS.

Em PFS, não existe o conceito de fluxo de marcas, ou seja, não existe uma dinâmica associada ao grafo PFS.

A Figura 2.16 mostra a utilização do grafo PFS para a representação do nível conceitual do exemplo da Figura 2.2.

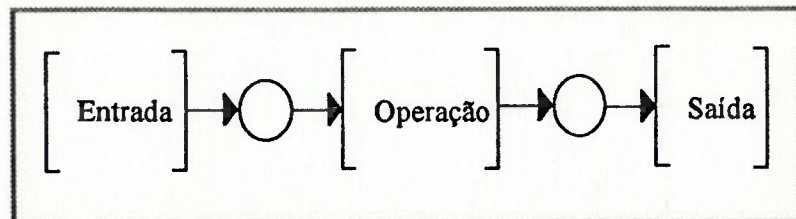


Fig. 2.16 Modelo PFS do sistema da Figura 2.2.

Uma atividade do PFS é representada em MFG por um "box" com uma transição de entrada ("[") que corresponde ao início da atividade e uma transição de saída ("]") que corresponde ao fim da atividade. Os elementos distribuidores podem ser substituídos por "boxes" de capacidade 1 ou N, dependendo da capacidade do distribuidor.

A Figura 2.17 mostra os tipos de atividades em correspondência com os "boxes" (macro-elementos) do MFG, vistos anteriormente. Essas atividades são:

- . Atividade unitária - apenas uma atividade pode estar no estado de execução.
- . Atividade de início e fim aleatórios - neste elemento, até N atividade podem coexistir simultaneamente.
- . Atividade de início simultâneo - a partir de um evento inicial, N atividades entram simultaneamente no estado de execução. Depois são finalizadas uma a uma por um evento final.

. Atividades de fim simultâneo - ocorrendo um evento inicial, N atividades entram, uma a uma, no estado de execução. No final são finalizadas simultaneamente por um mesmo evento final.

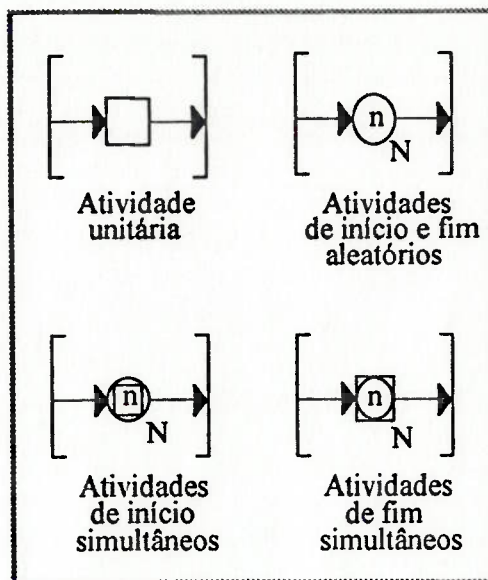


Fig. 2.17 Atividades e sua representação em MFG.

Desta maneira, detalhando-se a atividade **Operação**, obtém-se o modelo da Figura 2.18 que possui as sub-atividades **Carregamento**, **Processamento** e **Descarregamento**. A metodologia MFG/PFS permite o detalhamento para a obtenção do modelo apresentado na Figura 2.19 que representa o modelo funcional (MFG) e completo do exemplo da Figura 2.2.

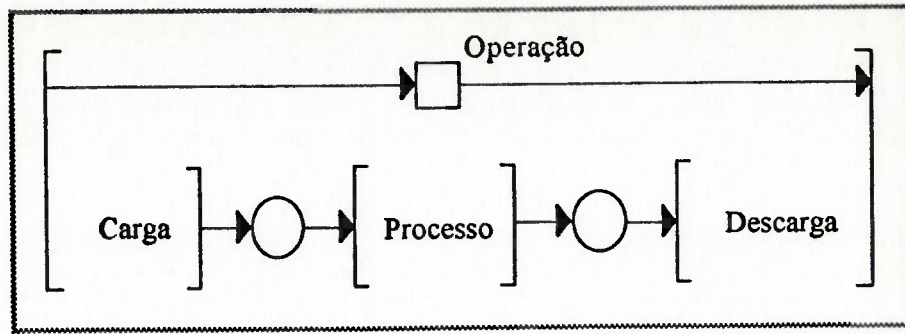


Fig. 2.18 Detalhamento da atividade Operação.

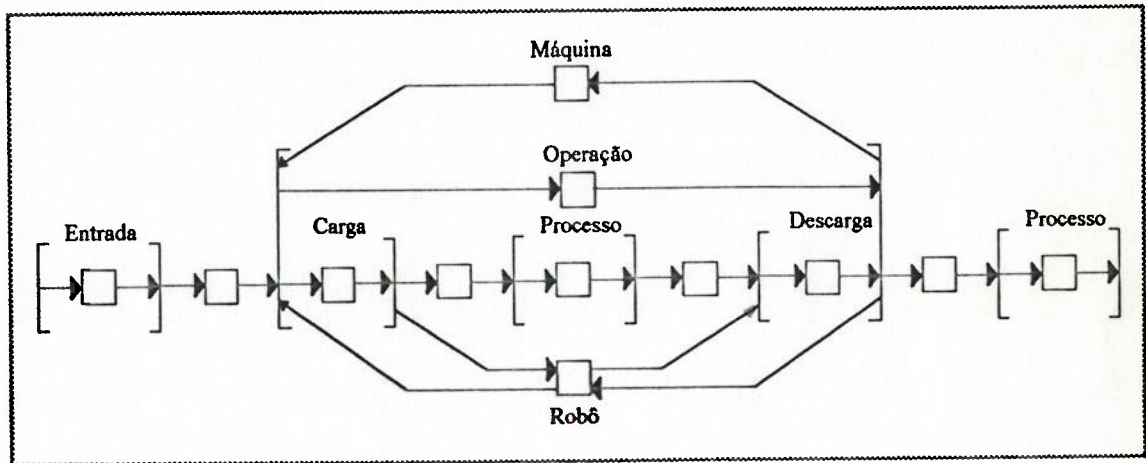


Fig. 2.19 Modelo completo do exemplo da Figura 2.3.

A utilização dos grafos MFG juntamente com os grafos PFS originou a metodologia MFG/PFS, que permite, desta maneira, a aplicação dos conceitos de hierarquia e modularização no desenvolvimento do projeto dos sistemas de manufatura. Esses conceitos e as propriedades inerentes às redes, conduzem a uma simplificação e sistematização da tarefa de especificação e modelagem de sistemas complexos e de grande porte.

Além da representação gráfica semelhante às redes de Petri, a metodologia MFG/PFS possui também uma representação textual que corresponde a linguagem MFG/PFS [MIYAGI (1988)]. Esta linguagem é muito semelhante à linguagens tradicionais como C ou Pascal pois as atividades são representadas como se fossem funções ou procedimentos das linguagens tradicionais. A representação através dessa linguagem do exemplo das Figuras 2.16, 2.18 e 2.19 pode ser:

```
/main
[
    c-box dist1(20), dist2(20);

    connect /Entrada: dist1;
    connect /Operacao: -dist1, dist2;
    connect /Saida: - dist2;
]
...
/Operacao
[
    c-box buf1, buf2;

    connect /carga: -dist1, buf1;
    connect /processo: -buf1, buf2;
    connect /descarga: -buf2, dist2;
]
...
```

2.6 SISTEMAS BASEADO EM REGRAS

Atualmente, a aplicação de técnicas de Inteligência Artificial⁹ [ARARIBÓIA (1989)][LINN (1990)][LUCENA (1987)] continua crescendo muito, principalmente em áreas como a medicina e engenharia. A utilização destas técnicas é realizada, normalmente, quando é necessário solucionar problemas caracterizados como sendo complexos, não estruturados, nebulosos, ambíguos [SPANGLER (1991)] e difícil de se obter uma formulação matemática tratável [SUBRAMANYAM (1986)].

Muitos sistemas especialistas ("expert system") existentes utilizam o enfoque de representação de conhecimentos baseados num conjunto de regras. O uso desta abordagem está relacionado com a facilidade de modelagem de determinados aspectos para tomada de decisão [BRUNO (1986)][LINN (1990)], a habilidade de assimilação de grandes quantidades de informações, a facilidade de reformulação de certas condições para o problema e a utilização de métodos eficientes para busca de soluções.

Os sistemas baseados em regras são conhecidos também como sistemas de produção ou regras de produção (vem do estudo das linguagens formais), e são, normalmente, representados através de um conjunto de regras do tipo se-então (IF-THEN).

Uma configuração básica de um sistema de produção de regras pode ser representado na Figura 2.20. Esta figura, mostra os componentes básicos [CUNHA (1987)] e as suas inter-relações.

Os componentes deste sistema de produção de regras são:

⁹ Técnicas com o objetivo de inserir inteligência nos computadores.

Conjunto de regras. É um conjunto do tipo regras de produção que possuem um formato coerente de modo a facilitar a implementação do mecanismo de inferência. Essas regras são do tipo "if-then". Elas contém as condições que representam o conhecimento do problema de uma determinada aplicação.

Base de dados (fatos). São dados que refletem um estado, um fato ou uma hipótese e, a sua organização pode ser linear ou hierárquica facilitando, dessa maneira, as alterações em seu conteúdo, através da mudança ou surgimento de novos fatos e hipóteses. Esta parte contém os fatos conhecidos ou inferidos sobre um caso específico. É a base de dados de trabalho.

Interpretador de regras (mecanismo de inferência). Contém o conjunto de funções que controlam a interação com o usuário e atualiza o estado corrente do conhecimento sobre o caso em estudo. Este mecanismo de inferência também decide qual a próxima regra a ser aplicada. O interpretador de regras ou mecanismo de inferência, deve ser muito eficiente, pois dependendo do tipo de problema que se deseja solucionar, o número de regras e a base de dados cresce bastante, tornando o sistema de regras ineficiente para determinadas aplicações.

Existem alguns métodos de inferência que objetivam melhorar a busca de soluções. Os dois métodos mais conhecidos, utilizados pelo interpretador de regras, nos sistemas baseados em regras são:

Inferência em cadeia ("forward chaining"), que parte de fatos conhecidos da base de dados para depois avaliar as regras;

Inferência dirigida por metas ("backward chaining"), que parte de uma meta a ser atingida para depois percorrer o conjunto de regras verificando quais regras têm como conclusão a meta a ser atingida.

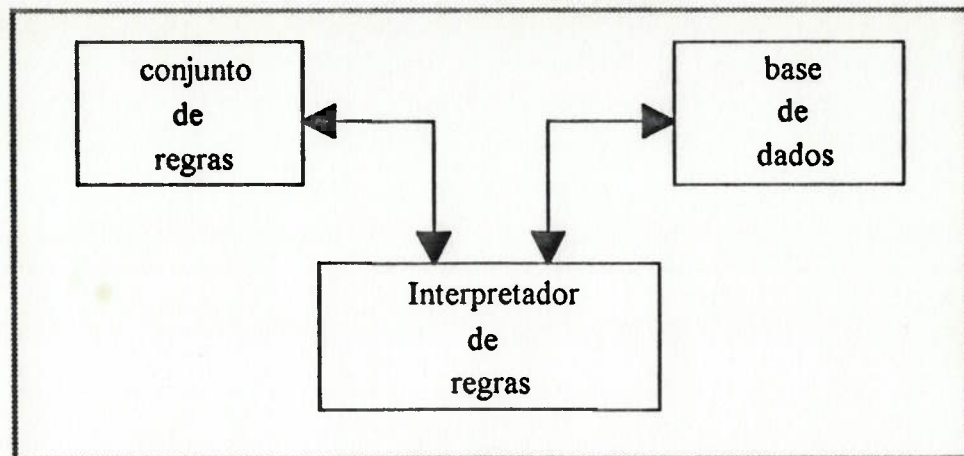


Fig. 2.20 Componentes de sistemas baseados em regras

As regras de produção do tipo se-então (if-then), são representados da seguinte forma:

Se condição_1
 condição_2
 ...
Então ação_1
 ação_2
 ...

Ou seja, se todas as condições *condição_i* forem satisfeitas, pode-se realizar as ações *ação_j*. Este exemplo demonstra a facilidade de se representar regras do tipo se-então.

Em alguns sistemas especialistas já implementados, utilizam-se outras formas de representação do conhecimento. Como as redes Semânticas [CUNHA (1987)] que é um grafo cujos nós representam os tipos e elementos nos quais são associados rótulos de linguagem natural e os arcos representam as relações entre estes elementos, porém, em

diferentes níveis de hierarquia existente entre eles e os "Frames" [CUNHA(1987), ARARIBÓIA (1989)] são compostas por estruturas padrões chamadas "slots" que permitem a representação de situações ou cenas que descrevem um determinado problema.

2.6.1 Aplicação em Sistemas de Manufatura

Nos sistemas de manufatura, verifica-se que existem situações que podem ser representados através de um sistema baseado em regras. Nos exemplos descritos a seguir apresenta-se uma representação, através de regras, de condições que são úteis e necessárias para solucionar conflitos.

O conflito surge devido a necessidade de compartilhamento de recursos como máquinas e sistema para transporte de peças. Algumas regras que representam as condições necessárias para solucionar este conflito podem ser:

Exemplo 1:

SE	uma peça chega e, existe robô livre e, existe máquina livre,
ENTÃO	pega a peça e coloque na máquina

Exemplo 2:

SE	estação 2 solicita transporte e transportador 1 está disponível
ENTÃO	atenda estação 2 solicitante

O exemplo 1 pode representar a condição necessária para que um robô possa pegar uma peça e colocar numa determinada máquina para a realização de um processo.

Já o exemplo 2, representa as condições necessárias para que um determinado transportador atenda a uma solicitação para transporte.

Verifica-se, desta maneira, que a representação de conhecimento baseado em regras deste tipo, possui algumas vantagens como [BEN (1986)]:

1. Facilidade de se acrescentar, alterar ou eliminar regras¹⁰.
2. Uniformidade na estrutura das regras. Todas são expressas na forma se-então.
3. Facilidade de interpretação em relação ao sistema físico (real).

Existem várias pesquisas sendo realizadas onde demonstra-se a utilidade e a necessidade da aplicação destas técnicas da Inteligência Artificial para a análise e simulação de SIM.

Nos trabalhos [VALETTE (1986)] e [VALETTE (1990)], Valette sugere a utilização das regras de produção, no nível de "scheduling", para a tomada de decisão para a solução de conflitos de alocação de recursos em tempo real. Neste caso, ele observa que o problema de programação ("scheduling") é complexo e o controle dos sistemas de manufatura que envolvem conhecimentos incertos e experiência humana, requerem a utilização de técnicas da Inteligência Artificial.

Em [MARTINEZ (1988)], é proposto uma aplicação de técnicas de Inteligência Artificial para controle e "scheduling" em tempo real de sistemas de manufatura. Neste trabalho é utilizado as redes de Petri coloridas juntamente com um sub-sistema baseado em conhecimentos.

Sahraoui [SAHRAOUI (1987)] propõe uma estruturação hierárquica de funções de monitoração através da integração de redes de Petri (para detecção e manipulação) e um enfoque baseado em conhecimentos (para diagnóstico e decisão).

¹⁰ Facilidade na manutenção das regras

A utilização das regras de produção para representação de conhecimentos dos sistemas de manufatura, possibilita o desenvolvimento de um sistema especialista. Um exemplo de aplicação é feito por Richard J. Linn [LINN (1990)], que utiliza um sistema especialista para controlar um sistema automático de "storage/retrieval" (sistema que manipula, armazena e recupera material). Ele utiliza o ambiente Prolog¹¹ para a implementação das regras de controle e elas são baseadas em representação do tipo se-então (IF-THEN).

¹¹ Linguagem para Inteligência Artificial.

Capítulo 3

ANÁLISE DOS SISTEMAS INTEGRADOS DE MANUFATURA

3.1 INTRODUÇÃO

Como visto anteriormente, os SIM são sistemas do tipo SED e, desta maneira, possuem as características de assincronismo, paralelismo e conflito que podem provocar, nestes sistemas, fenômenos indesejáveis como o "deadlock" e a "colisão" [MIYAGI (1988), MIYAGI (1989)].

O "deadlock", acontece nestes sistemas, quando eventos e condições se combinam de tal forma que se atinja um estado, no qual novos eventos não podem mais ocorrer. Neste caso, o sistema para de funcionar, totalmente ou parcialmente, dependendo do estado onde ocorreu este fenômeno.

A "colisão" é um fenômeno que acontece quando uma certa condição estiver sendo mantida, e o evento imediatamente anterior ocorre, gerando desta maneira, uma

informação redundante. Este tipo de fenômeno pode ser evitado garantindo-se que o sistema seja seguro ("safe").

Neste trabalho somente o fenômeno de "deadlock" será analisado, pois para a modelagem dos sistemas, utiliza-se os grafos MFG, que permite apenas uma marca no elemento "box", garantindo, desta maneira, a característica de segurança ("safeness") do modelo. Assim sendo, este fenômeno de "colisão" será evitado "por construção" do modelo MFG do sistema.

Desta maneira, apresenta-se neste capítulo uma definição para o estado de "deadlock", um exemplo de sistema de manufatura onde ocorre este estado e a descrição de algumas técnicas que auxiliam a detecção e a solução destes estados indesejáveis. Além disso, descreve-se os métodos estrutural e comportamental utilizados para análise dos SIM.

3.2 O QUE É "DEADLOCK" NOS SIM ?

Nos SIM, o "deadlock" surge como um estado ou uma situação altamente indesejável. Pois, atinge-se um estado em que, parte ou todo o sistema pára de evoluir (sistema está num estado "morto") [BANASZAK (1990), VISWANADHAM (1990)], interrompendo, nos ambientes de chão de fábrica, o processo produtivo.

Esta parada parcial ou total do processo de produção devido a este estado demonstra a necessidade da obtenção de sistemas de controle para os SIM bastante eficientes, de tal forma que o processo por ele controlado seja totalmente livre de qualquer tipo de "deadlock" (propriedade de vivacidade - "liveness" dos grafos em rede de Petri).

A Figura 3.1 apresenta um exemplo de um sistema de manufatura, que devido a uma determinada seqüência de eventos, pode entrar num estado de "deadlock" [VISWANADHAM (1990)].

3.2.1 Exemplo de "deadlock"

O exemplo corresponde a um processo de produção simples de peças, onde é utilizado uma estação de carga e descarga (C/D), um veículo autônomo de transporte (AGV - "Automated Guided Vehicle") e uma estação de transformação (máquina de comando numérico - NC).

Este sistema apresentado na Figura 3.1 funciona da seguinte forma: o AGV transporta uma peça da estação C/D para a estação de processamento NC, que realiza um processamento sobre a peça e em seguida, após a peça ser processada, o AGV transporta-a de volta para a estação C/D para ser descarregada.

Sabendo-se que o estado inicial corresponde ao AGV pronto para transportar, a estação NC está livre para receber uma peça para processamento e existe peças disponíveis na estação C/D para serem carregadas. Desta forma, observa-se a seguinte seqüência de eventos:

- 1) o AGV carrega a peça 1 e a transporta até a estação NC, que inicia o processamento desta peça;
- 2) o AGV retorna para a estação C/D e carrega a peça 2 e leva-a em direção a estação NC. Porém a estação NC está ocupada, processando a peça 1. Desta forma, o AGV espera que a estação NC termine o processamento, ficando, bloqueado à espera do fim do processamento desta peça;

3) a estação NC termina o processamento da peça 1 e espera que o AGV transporte a peça 1, já processada para a estação C/D, para ser descarregada. Desta forma, a máquina também fica bloqueada, esperando o AGV. Assim sendo, se o AGV e a máquina NC podem acomodar somente uma peça por vez e não existem "buffers" temporários, os dois recursos permanecem esperando indefinidamente, ou seja, o sistema entra num estado, onde não há possibilidade de se continuar a evolução do processo. Esta situação caracteriza o estado de "deadlock" do sistema.

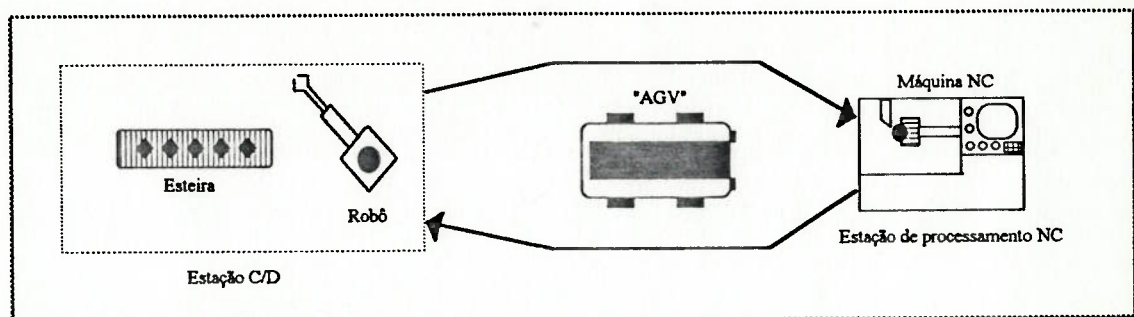


Fig. 3.1 Exemplo de um sistema de manufatura com uma estação de carga e descarga (C/D), uma estação de transformação (NC) e um sistema de transporte (AGV).

Existem duas formas que podem ajudar a evitar este "deadlock":

- a) utilização de magazines na entrada e na saída da estação. Estes magazines, para armazenamento temporário ("buffer"), liberam a estação após o término do processamento;
- b) definição de regras que permitam a escolha das operações necessárias para a continuidade da evolução do sistema.

Uma possível regra para este exemplo é: o AGV espera a estação NC terminar o processamento de uma peça, antes de pegar outra na estação C/D.

3.2.2 Tipos de "deadlock" nos SIM

No contexto dos sistemas de manufatura, verifica-se que podem existir dois tipos distintos de "deadlocks":

1) **"deadlock" condicional** - refere-se, principalmente, à alocação dinâmica dos recursos, ou seja, a medida que o processo evolui, podem ocorrer situações de conflito, como uma alocação de transportadores (robôs, AGVs) ou alocação de estações disponíveis para a realização de um determinado processo. Se este conflito não for resolvido satisfatoriamente, o sistema pode entrar num estado de "deadlock". Num sistema modelado em rede de Petri (ou grafos MFG), este fenómeno ocorre devido a uma marcação inicial inadequada, ou seja, pode-se obter um modelo livre de "deadlocks" (vivo), a partir de outra marcação inicial, adequada.

2) **"deadlock" estrutural** - ocorre devido a problemas na estrutura do grafo que representa o sistema em estudo. Desta forma, este fenómeno é consequência de uma modelagem imprecisa do sistema (alocação estática dos recursos - "layout"). Neste caso, este grafo não é vivo, independentemente da marcação inicial utilizada.

3.2.3 Como detectar os "deadlocks" nos SIM

Dependendo da complexidade do sistema, a previsão e a detecção dos fenómenos de "deadlocks" é muito difícil. Para os SIM, verifica-se que, normalmente, este estado é inerente ao processo produtivo, isto é, ele é dependente da dinâmica do sistema em estudo.

O "deadlock" estrutural pode ser detectado através de um método analítico, onde a partir da aplicação de equações matemáticas, baseado no formalismo das redes de Petri, obtém-se um resultado que possibilita a verificação da integridade desses grafos.

Um método que pode ser utilizado para a detecção de "deadlock" é a análise de invariantes [MURATA (1989), REISIG (1982)] sobre os grafos. Esta análise, permite uma verificação da estrutura do grafo através da aplicação de equações matemáticas sobre uma matriz de incidência¹² que representa o sistema em estudo. O resultado da aplicação destas equações, permite a verificação da existência de "deadlock" estrutural, causado por erros de modelagem do sistema (o modelo não representa satisfatoriamente o sistema em estudo).

Existem dois tipos de análise de invariantes: s-invariantes que consiste na verificação do número de marcas existentes ou que passam nos lugares da rede. Isto possibilita a verificação da existência ou não de um acúmulo de marcas em trechos da rede, indicando portanto, a existência de "deadlocks"; e t-invariantes que indica quantas vezes cada transição dispara, partindo-se de uma dada marcação para reproduzir esta mesma marcação. A partir deste método, pode-se verificar a existência ou não de ciclos [YOSHIDA (1988)] que podem indicar, a existência de "deadlocks", no sistema em estudo.

Por outro lado, o "deadlock" condicional também pode ser detectado através da simulação dos estados possíveis de serem atingidos pelo sistema.

A análise de invariantes e a simulação de estados, são técnicas que utilizam métodos já conhecidos e aplicados em redes de Petri. No item seguinte, descreve-se os

¹² Para uma rede de Petri com n transições e m lugares, a matriz de incidência é uma matriz $n \times m$ de inteiros que indicam as pré e pós condições das transições.

métodos aplicados nas técnicas de análise e que indiretamente foram utilizados neste trabalho.

3.3 MÉTODOS DE ANÁLISE

Segundo MURATA (1989) os métodos para análise podem ser separados por 3 grupos:

Árvore de cobertura - este método corresponde a verificação de todas as marcações alcançáveis. A partir de uma marcação inicial, é possível alcançar novas marcações de acordo com o disparo das transições. E, de cada marcação alcança-se mais marcações. Representando-se cada marcação como um nó, obtém-se uma representação em árvore das marcações. Desta forma, os nós representam a marcação gerada a partir da marcação inicial M_0 (raiz da árvore) e seus sucessores e, os arcos representam o disparo de transições que transformam uma marcação em outra. Numa rede de Petri limitada, a árvore de cobertura é chamada de árvore de alcançabilidade, pois ela contém todas as marcações possíveis, neste caso, pode-se resolver todos os problemas de análise através da árvore de alcançabilidade. A grande desvantagem é que este método é exaustivo e a sua aplicação é limitada à modelos simples, pois quanto mais complexo é o sistema, mais complexo também é o modelo, havendo, desta maneira uma explosão do espaço de estados. A Figura 3.2 mostra uma rede de Petri e a sua respectiva árvore de alcançabilidade.

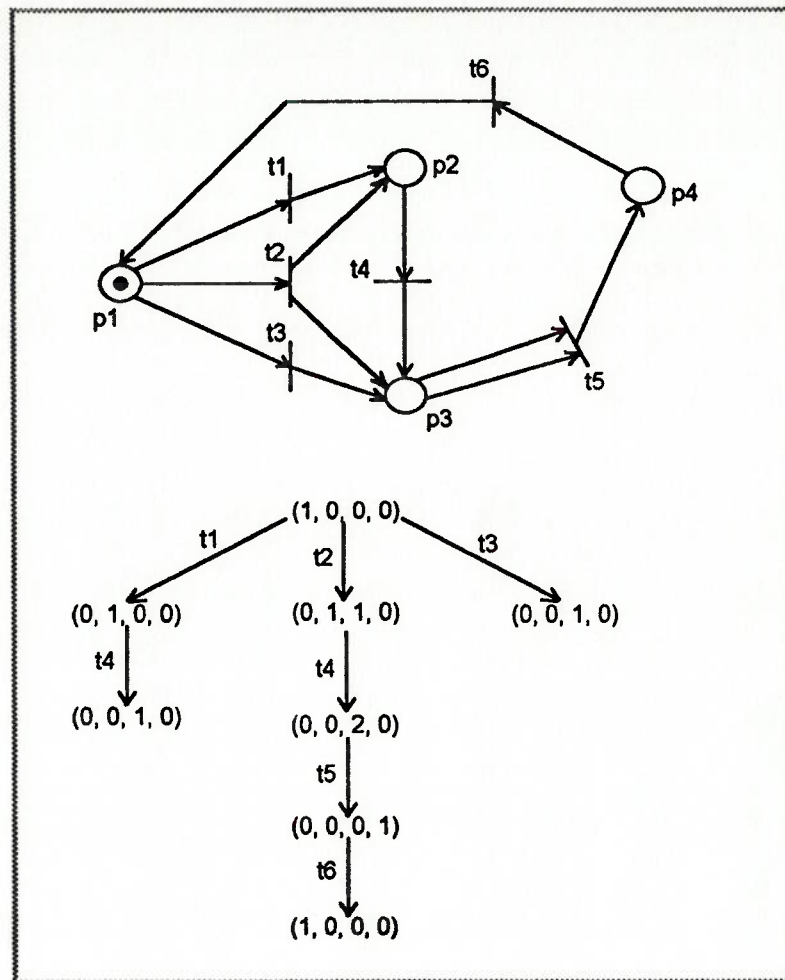


Fig. 3.2 Exemplo de rede de Petri e a respectiva árvore de alcançabilidade.

Matriz de incidência e equação de estado - para este método, o comportamento dinâmico dos sistemas em estudo, pode ser representado através de equações algébricas. Neste caso, o ideal seria a possibilidade de se descrever e analisar completamente o comportamento dinâmico das redes de Petri através destas equações. Entretanto, a solução destas equações é muitas vezes limitada por causa da natureza não-determinística inerente aos modelos de redes de Petri. A matriz de incidência de uma rede de Petri com n transições e m lugares, corresponde a uma matriz de $n \times m$ de números inteiros. Cada elemento desta

matriz, representa a forma que os lugares estão conectados com a transição, ou seja, ela representa as conexões entre lugares e transições numa rede de Petri. As colunas correspondem aos lugares e as linhas às transições, desta maneira, o número negativo indica que é um lugar de entrada, o positivo indica que é um lugar de saída e zero indica que não há relação entre o lugar e a transição correspondente.

Portanto, a matriz de incidência correspondente ao exemplo apresentado na Figura 3.2 é:

$$M_i = \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & +1 \\ +1 & +1 & 0 & -1 & 0 & 0 \\ 0 & +1 & +1 & +1 & -2 & 0 \\ 0 & 0 & 0 & 0 & +1 & -1 \end{bmatrix}$$

E a equação de estados correspondente é $p(k+1) = p(k) + M_i \cdot t(k)$, onde $p(k)$ é o estado atual, $p(k+1)$ o estado seguinte, $t(k)$ o vetor de disparos e M_i a matriz de incidência. Desta maneira, para obter-se o estado $[0 \ 1 \ 0 \ 0]^T$ à partir da marcação inicial $[1 \ 0 \ 0 \ 0]^T$ tem-se a seguinte equação:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & +1 \\ +1 & +1 & 0 & -1 & 0 & 0 \\ 0 & +1 & +1 & +1 & -2 & 0 \\ 0 & 0 & 0 & 0 & +1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

onde o 1 no vetor de disparos indica que a transição t1 foi disparada para se alcançar o estado desejado.

Regras de redução - este método consiste na redução de um modelo complexo num modelo mais simples e tratável, porém sem perder as propriedades. A técnica de transformar modelos abstratos em modelos mais refinados (refinamento sucessivos) de maneira hierárquica, pode ser utilizado para a síntese das redes. A Figura 3.3 mostra alguns exemplos de como trechos de redes podem ser simplificados.

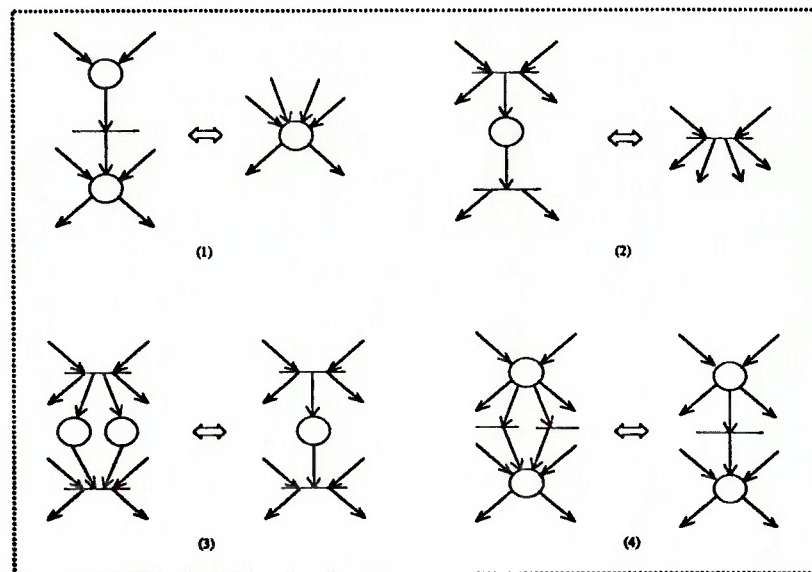


Fig. 3.3 Reduções com preservação das propriedades de segurança, vivacidade e limitabilidade

Assim sendo, as técnicas existentes de análise para os SIM têm como objetivo principal, a avaliação dos modelos, de tal forma a verificar as alternativas fornecidas e, a partir destes dados, selecionar a(s) melhor(es).

Para modelos baseados em redes, pode-se distinguir duas técnicas principais para se realizar uma análise de SIM:

- Análise estrutural que permite uma análise da estrutura da rede através da aplicação de métodos matemáticos (método da matriz de incidência e equações de estado).
- Análise comportamental, que através da verificação dos estados atingidos pelo sistema, permite a verificação do comportamento dinâmico do sistema (método da árvore de cobertura).

Verifica-se que existem influências na análise de um sistema devido a capacidade de modelagem de alguma ferramenta. Segundo Murata [MURATA (1989)], um modelo baseado numa ferramenta genérica para a representação de sistemas de manufatura (por exemplo), tende a possuir menos informações necessárias para a realização de uma análise efetiva e satisfatória.

3.3.1 Análise Estrutural

Num modelo baseado em redes, a análise estrutural depende da estrutura topológica do grafo, isto é, esta análise é independente da marcação inicial. Desta maneira, pode ser caracterizado através da utilização do método de equações de estado e matriz de incidência.

Este tipo de análise permite a obtenção de informações sobre a estrutura da rede correspondente a um sistema em estudo. Ela permite a detecção de estados de "deadlocks" estruturais no sistema.

Os métodos analíticos para a realização da análise estrutural, têm vantagens importantes quando utilizados como técnica para solução de problemas. Estas vantagens incluem:

- concisão na descrição do problema,
- solução de forma fechada,
- avaliação facilitada da influência na saída causado mudanças na entrada e,
- produção de soluções ótimas em muitos casos.

Apesar das vantagens oferecidas por este método, ele possui também algumas desvantagens importantes como:

- descrição dos sistemas pode ser não realístico devido a ocorrência de eventos não esperados (não-determinismo dos SIM).
- formulação matemática complexa, ou seja, as equações matemáticas necessárias para representar com um grau de realismo satisfatório, é praticamente intratável.

Assim sendo, o método comportamental (simulação) descrito no item seguinte, pode compensar estas desvantagens, porém sacrificando-se algumas vantagens do método analítico, principalmente a possibilidade de descrição precisa através de funções matemáticas [HOOVER (1989)].

Para as redes, pode-se realizar outro tipo de análise estrutural do modelo, o cálculo da *distância síncrona* [CHONG-YI (1985), GOLTZ (1985)]. Neste caso esta análise permite a verificação do grau de sincronismo entre os elementos da rede. Ele possibilita a verificação da quantidade de influência que um determinado elemento tem no sistema.

3.3.2 Análise Comportamental

A análise comportamental de um modelo de sistema de manufatura, também chamado de "simulação do sistema de manufatura", é necessária quando se investiga a

dinâmica do processo e as tomadas de decisões para a realização do controle destes sistemas.

A técnica de simulação é bastante aplicada em diversas áreas, pois o modelo pode representar um sistema de maneira bastante realística [HOOVER (1989)]. Comparando-se a simulação com uma técnica analítica tal como a teoria de filas [SOARES (1990)], verifica-se que o modelo utilizado na simulação não requer, suposições rigorosas para alguns sistemas em estudo.

Entretanto, quando se implementa um sistema de simulação através das redes de Petri, em um computador digital capaz de manipular redes de grande porte, com protocolos complexos e capaz de automatizar o processo de disparos, verifica-se que ocorrem muitos problemas, principalmente se o computador executa as instruções seqüencialmente. Com o avanço tecnológico para a construção dos processadores, este tipo de problema começa a ser solucionado a partir da obtenção de computadores que permitem o processamento de várias tarefas ao mesmo tempo (processamento paralelo e multi-tarefa).

Através da simulação pode-se descrever o comportamento dinâmico de sistemas complexos através da utilização de computadores de alto poder de processamento. Além disso, pode-se usar para análise de sistemas inexistentes, ou sistemas já existentes sem a necessidade de alterá-los.

Para este trabalho, a ferramenta proposta utiliza os métodos apresentados para a modelagem, simulação e análise aplicando-se a metodologia MFG/PFS. Porém, verifica-se a existência de limitações que influenciam nessa análise. No item a seguir descreve-se a limitação devido a utilização dos grafos MFG para a modelagem do comportamento dinâmico do sistema.

3.4 LIMITAÇÃO DO MFG

Conforme visto no capítulo 2, os grafos MFG possuem características que demonstram a sua utilidade para a modelagem, simulação, análise e controle dos SIM. As características principais do MFG são [MIYAGI (1993)]:

- permite a utilização do formalismo matemático (baseada nas rede de Petri),
- permite apenas uma marca nos "boxes" (é um grafo seguro),
- permite o envio de sinais de estado do modelo para dispositivos externos (através dos arcos de sinais externos),
- permite a recepção de sinais dos dispositivos externo e a partir delas controlar a ocorrência de eventos no modelo (através dos "gates" externos).

Porém, verifica-se que numa situação onde é necessária uma arbitragem para solucionar um determinado conflito, depende-se de uma implementação da lógica para o árbitro, através dos grafos MFG. Desta maneira, este árbitro torna-se complexo e de difícil interpretação. A Figura 3.5 mostra a implementação de um árbitro utilizando o grafo MFG para a solução de um determinado conflito.

A Figura 3.4 apresenta uma situação de *conflito* onde, verifica-se que as transições de entrada, da atividade A e da atividade B estão habilitadas para disparar (qualquer uma pode disparar) e o disparo de uma delas desabilita o outro. Desta forma, no exemplo da Figura 3.4 se não for pré-definida uma seqüência, pode ocorrer que somente a máquina A se utilize do recurso robô.

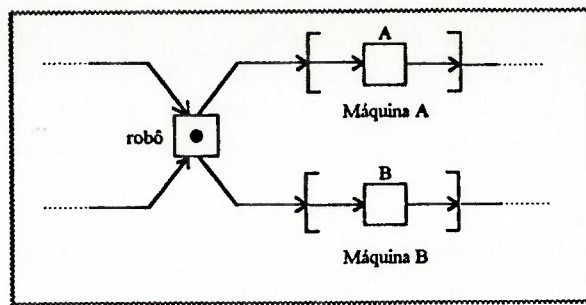


Fig. 3.4 Representação do conflito em MFG/PFS.

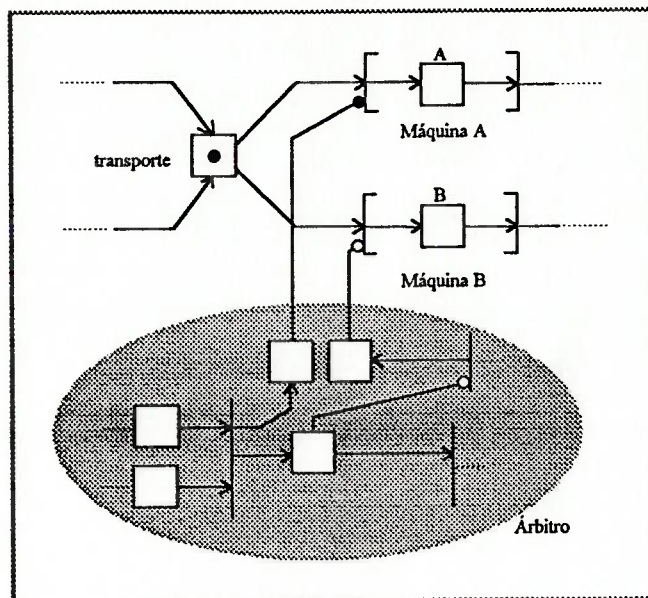


Fig. 3.5 Representação do conflito e do árbitro em MFG/PFS.

3.4.1 Sistema híbrido (MFG e REGRAS)

A denominação "sistema híbrido", corresponde a utilização do grafo MFG associado a um árbitro externo, baseado em regras. O objetivo deste árbitro é a simplificação do modelo MFG através da transferência das condições para solução dos

conflitos (lógica do árbitro) para este árbitro externo. Neste caso, o modelo MFG torna-se mais claro e, a implementação e interpretação destas condições são facilitadas pela utilização das regras. A Figura 3.6 mostra como é realizada a associação da situação de conflito representado no modelo com o árbitro externo, através da utilização dos "gates".

A troca de informações entre o modelo e o árbitro externo é realizada da seguinte forma:

- utiliza-se os "gates" para a recepção dos sinais provenientes do árbitro e,
- utiliza-se arcos de sinais de saída que enviam sinais dos elementos do modelo MFG para o árbitro externo.

Desta forma, o árbitro externo mantém a informação de estados do sistema monitorados. Assim sendo, através da marcação do grafo que corresponde aos fatos ou estado do sistema e as regras associadas, é possível solucionar as situações de conflitos.

A situação de *conflito* e a respectiva associação do modelo ao Árbitro Externo via "gates" é mostrada na Figura 3.6. Neste caso, a habilitação das transições de entrada da atividade A e da atividade B depende dos sinais dos "gates" habilitadores conectados a estas transições.

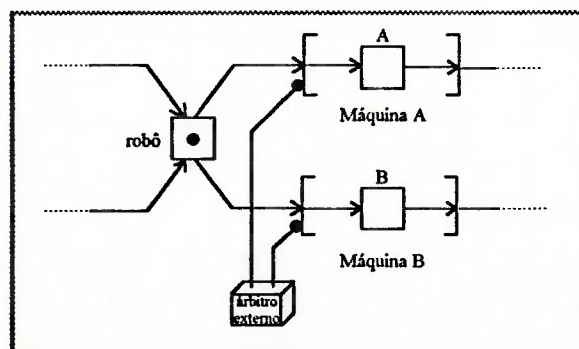


Fig. 3.6 Associação do Árbitro Externo.

A verificação deste fator limitante do MFG, pode ser visto com maiores detalhes no capítulo 5, onde, apresenta-se o modelo em MFG/PFS de um sistema de manufatura e uma comparação da implementação do árbitro em MFG e o correspondente árbitro externo, baseado em regras.

Capítulo 4

FERRAMENTA DE SIMULAÇÃO E ANÁLISE

4.1 INTRODUÇÃO

As pesquisas realizadas sobre: as técnicas para a modelagem, análise e controle de sistemas de manufatura baseadas em redes de Petri (metodologia MFG/PFS), os sistemas baseado em regras e a necessidade de se realizar análise dos SIM em "off-line" permitiram a especificação e a respectiva implementação de uma ferramenta protótipo para auxílio à projetos dos SIM.

Esta ferramenta verifica o funcionamento do sistema a partir da simulação e análise do modelo que representa o sistema em estudo. Uma aplicação importante para esta ferramenta, é a utilização para solucionar conflitos que ocorrem nos sistemas de manufatura. Neste caso, utiliza-se um "árbitro" externo baseado em regras de Produção, associado ao modelo em MFG/PFS.

A partir de uma análise topológica da estrutura da rede e da verificação dos estados atingíveis, é possível detectar "deadlocks" e a partir dos dados fornecidos, solucioná-los. Desta maneira, esta ferramenta permite a realização de análises qualitativa e quantitativa dos sistemas.

A seqüência básica e necessária para a realização da análise e simulação dos SIM é representada na Figura 4.1, onde, inicia-se a partir da elaboração de um modelo em linguagem MFG/PFS, um compilador transforma este modelo num MFG puro, de tal maneira, que possa ser realizada a simulação e análise deste sistema em estudo.

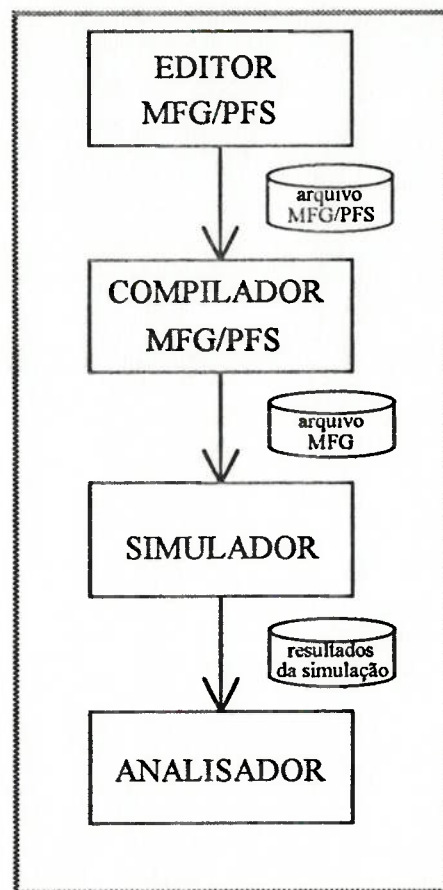


Fig. 4.1 Seqüência necessária para análise de SIM.

O bloco SIMULAÇÃO representado na Figura 4.1 corresponde à posição da ferramenta proposta, pois ela é responsável pela simulação (análise comportamental) do sistema em estudo.

4.2 DESCRIÇÃO DA FERRAMENTA

A ferramenta utiliza para a modelagem do SIM, os grafos MFG/PFS. Desta maneira, é possível a obtenção da matriz de incidência do grafo e conseqüentemente, a aplicação das equações necessárias para a realização da análise de invariantes. A tomada de decisão nos conflitos não resolvidos, são feitas a partir da avaliação das regras associadas ao modelo do sistema em estudo. As regras são baseadas em Regras de Produção. A partir da especificação do "árbitro" associado a um "jogador de marcas" (entidade que executa as regras de disparo do MFG), toma-se a decisão para solucionar conflitos. A avaliação das regras neste "árbitro" é realizada através de um interpretador de regras ou um mecanismo de inferência que escolhe uma possível solução.

O *árbitro* contém um mecanismo de inferência, responsável pela seleção das transições a serem disparadas a partir da avaliação das regras. O *jogador de marcas*, é o módulo responsável pelo disparo das transições habilitadas (inclusive aquelas que foram selecionadas pelo árbitro externo).

Os conflitos, como visto anteriormente, são decorrentes de uma necessidade de se realizar uma alocação satisfatória dos recursos disponíveis. Nesta ferramenta, estes conflitos são resolvidos através da utilização de uma técnica baseada em regras que, após a interpretação destas regras, uma solução é obtida. Isso possibilita uma alteração no comportamento dinâmico do sistema de tal forma que se evite a ocorrência de "deadlock" e que por conseqüência melhore o seu desempenho.

4.3 ESTRUTURA DA FERRAMENTA

A Figura 4.2 apresenta a estrutura básica da ferramenta. Verifica-se que o usuário interage com a ferramenta de duas formas: a primeira através da alteração da estrutura do modelo MFG/PFS e a outra forma é através da alteração do comportamento dinâmico do sistema mudando-se as regras e/ou marcações iniciais.

De acordo com a Figura 4.2, essa ferramenta de análise permite a solução dos "deadlocks" estruturais a partir da alteração da estrutura do grafo de acordo com os resultados fornecidos por uma análise de invariantes dos elementos do grafo. Os "deadlocks" condicionais que surgem a partir de uma situação de "conflito", são solucionados através da verificação dos estados atingíveis através da simulação. Neste caso, é realizada uma associação de um "árbitro", baseado em regras, com um "jogador de marcas".

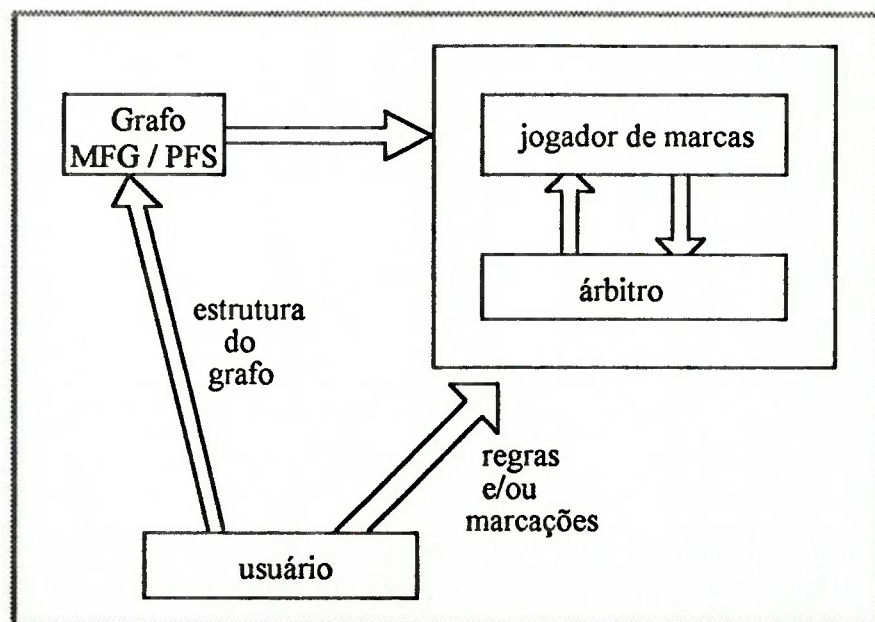


Fig. 4.2 Estrutura básica da ferramenta.

A solução de "deadlocks" existentes no sistema em estudo é feita da seguinte maneira: assim que os "deadlocks" são detectados, a ferramenta permite que o usuário corrija essa situação através da alteração do modelo MFG/PFS do processo, da marcação inicial ou do conjunto de regras associado para a solução dos conflitos. Esta interação do usuário com a ferramenta é representada na Figura 4.2.

Desta maneira, essa ferramenta possui suporte para atender às duas atividades básicas de simulação e análise que são:

(1) solucionar "deadlocks" estruturais - a partir da aplicação do método de análise de invariantes, pode-se detectar este tipo de "deadlock". Nessa fase o grafo MFG/PFS do processo produtivo é analisado, e caso o "deadlock" estrutural seja detectado, o usuário interage com a ferramenta, podendo alterar a estrutura do grafo MFG/PFS.

(2) solucionar "deadlocks" condicionais - este tipo de "deadlock" surge devido aos conflitos relacionados com o processo e a alocação dinâmica de recursos. A solução desses conflitos é realizada através de um "árbitro" que é definido como um conjunto de regras representativas das restrições necessárias para executar o controle. O sistema é simulado inicialmente a partir de um conjunto de regras pré-estabelecidas. Caso ocorra algum "deadlock" condicional, a ferramenta fornece dados que possibilita o usuário a alteração do conjunto de regras de acordo com as necessidades. Essas operações são realizadas até que um conjunto de boas regras seja obtido.

Para a verificação da existência do "deadlock" condicional é importante e necessário que o modelo em estudo, seja totalmente livre do "deadlocks" estruturais. Por isso, propõe-se a implementação do algoritmo para a detecção deste tipo de "deadlock". A Figura 4.3 apresenta uma seqüência para a realização de uma análise consistente (ou aproximadamente) do modelo.

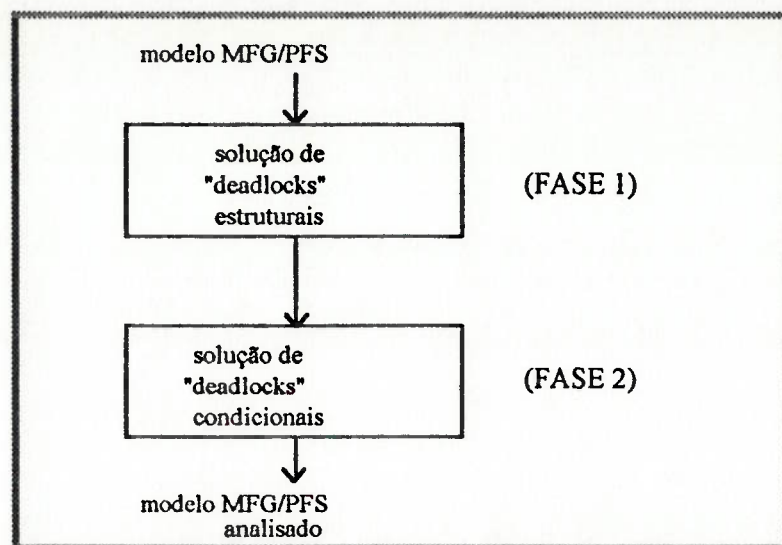


Fig. 4.3 Seqüência necessária para análise.

A escolha das opções a partir das regras e dos fatos que representam o estado atual do sistema, é feita através do interpretador de regras (máquina de inferência) que avalia esse conjunto de regras e infere um resultado, solucionando desta forma o conflito de tal forma que o processo continue a sua evolução. Os fatos (base de dados correspondente à marcação do grafo) são atualizados a medida que se obtém um novo estado do sistema.

4.4 FUNCIONAMENTO DA FERRAMENTA

Como visto anteriormente, a ferramenta verifica a existência de "deadlock" estrutural e caso exista, permite que o usuário altere o grafo. Em seguida, é feita a verificação de "deadlocks" condicionais a partir da simulação dos estados.

Deve-se especificar o conjunto inicial de regras correspondente ao sistema em estudo, de tal forma à implementar o "árbitro" externo. Neste caso, o conjunto de fatos

(base de dados) corresponde ao estado do sistema, representado pela marcação do grafo. No caso do estado inicial, os fatos são baseados na marcação inicial do grafo.

A medida que o sistema evolui, a base de dados contendo o conjunto de fatos são alterados a partir dos novos estados atingidos pelo sistema.

O sistema baseado num conjunto de regras (detalhado no capítulo 2) utilizado nesta ferramenta, tem como objetivos principais:

- simplificar a implementação do "árbitro" e
- facilitar a sua interpretação.

Assim sendo, estas regras permitem que os grafos MFG/PFS sejam mais simples de serem interpretadas e modificadas. Acarretando em m aumento na produtividade, flexibilidade e a manutenibilidade do modelo. Portanto, a estratégia de controle pode ser alterada sem muitas dificuldades, podendo ser realizada, inclusive, por uma pessoa responsável pela operação do sistema a nível de chão de fábrica, que é o especialista sobre o seu ambiente.

Quando há uma ocorrência de "deadlock", a ferramenta fornece dados que podem ser úteis e necessários para que o usuário possa solucioná-los. Ele pode a partir disso, alterar o modelo, a marcação inicial e também o conjunto de regras. Os subsídios que podem ser fornecidos pela ferramenta são: trecho do grafo onde ocorre o "deadlock", possíveis elementos responsáveis pelo "deadlock" e também as possíveis opções para a solução desses "deadlocks". Portanto, a resolução do "deadlock" condicional pode ser realizada a partir da alteração da marcação inicial ou do conjunto de regras, de acordo com os dados fornecidos pela ferramenta.

O "deadlock" estrutural, detectado por uma análise de invariantes, é resolvido a partir da alteração da estrutura do grafo, ou seja, a partir da correção do modelo do sistema em estudo.

4.4.1 Algoritmo para solução do "deadlock" estrutural

A Figura 4.4 mostra o fluxograma de um algoritmo proposto para a solução de "deadlocks" estruturais. A entrada do algoritmo é o modelo MFG/PFS do sistema. Neste modelo, aplica-se a análise da estrutura da rede (modelo MFG/PFS do sistema em estudo). Se for identificado algum "deadlock" no sistema, a ferramenta fornece dados correspondentes ao elemento, posicionamento no grafo e demais informações com objetivo de facilitar a correção do modelo.

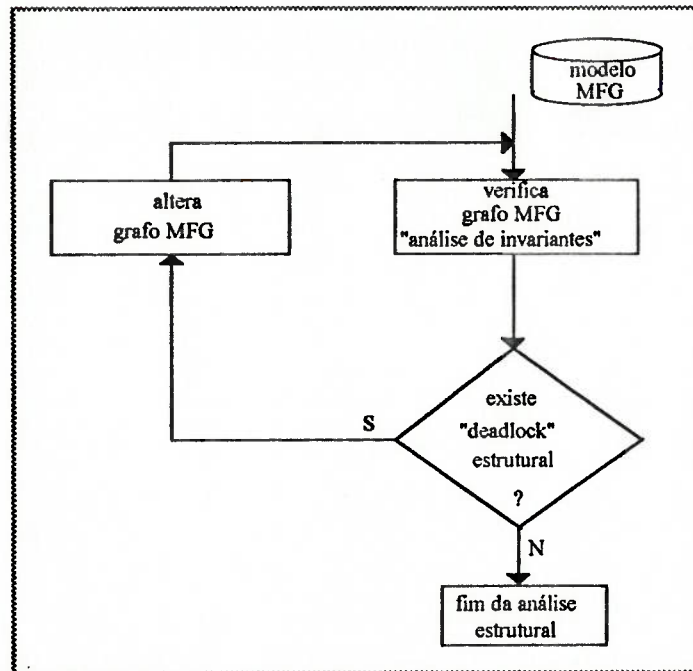


Fig. 4.4 Algoritmo para solução de "deadlock" estrutural.

4.4.2 Algoritmo para solução de "deadlocks" condicionais

A Figura 4.5 apresenta o algoritmo, utilizado pela ferramenta, para a solução de "deadlocks" condicionais. Neste caso, verifica-se os estados possíveis de serem atingidos, através do disparo das transições. No bloco *Entrada de Dados* realiza-se a especificação do modelo MFG/PFS do sistema em estudo, do conjunto de regras iniciais, da marcação inicial e dos dados que definem a maneira que será feita a simulação (tempo total, período, número de passos, etc).

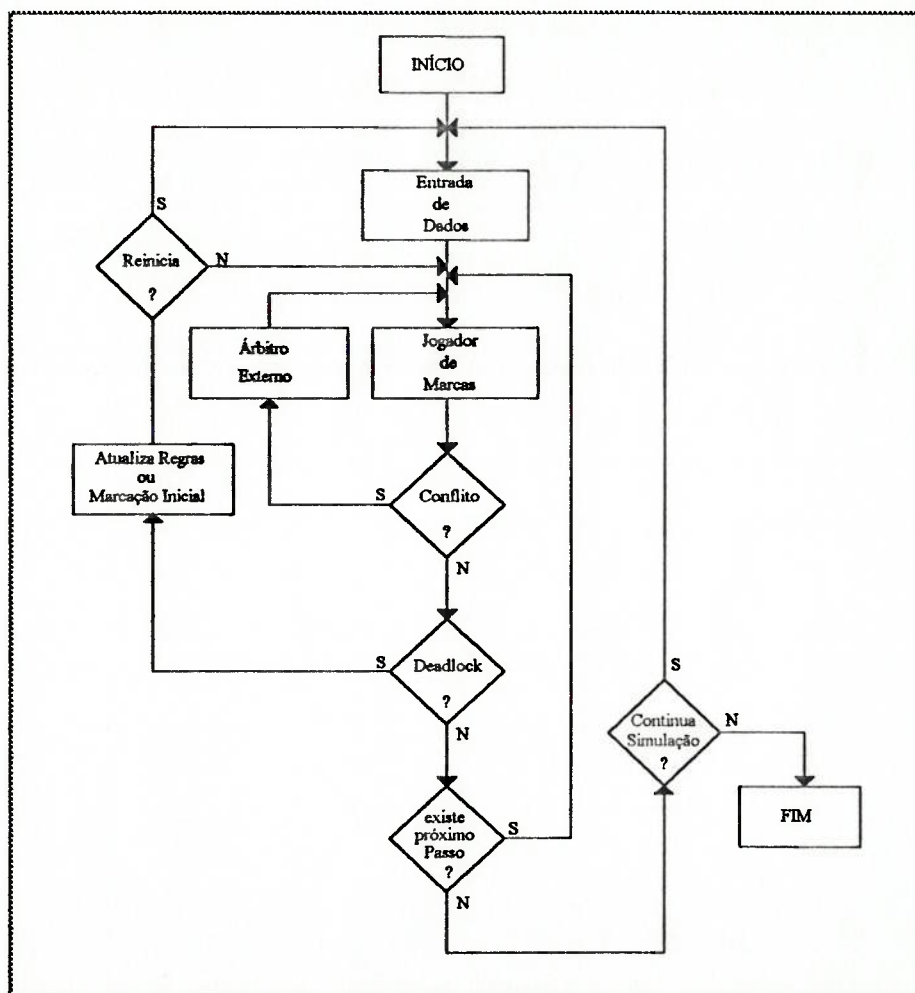


Fig. 4.5 Algoritmo para solução de "deadlock" condicional.

O *Jogador de Marcas* verifica se existem conflitos no modelo, e caso exista, o *Árbitro externo*, baseado em regras gera a solução. Se não existir o conflito, e caso exista "deadlock", pode-se alterar regras ou a marcação inicial para reiniciar ou não o processo. Se não existir "deadlock", continua no próximo passo, reinicia ou finaliza a simulação.

4.5 DETALHES DE IMPLEMENTAÇÃO DA FERRAMENTA

A implementação desta ferramenta utiliza-se a linguagem C nos módulos procedurais (simulador, editor gráfico de MFG/PFS). Para o árbitro externo, baseado em regras, propõe-se a utilização do PROLOG que permite uma implementação para as regras e para a máquina de inferência de maneira simplificada.

A listagem a seguir mostra o programa principal utilizado na ferramenta, onde verifica-se as opções para simulação e determinação da matriz de incidência.

```
#include <stdio.h>
#include "teetd.h"
#include "teglob.h"
#include "teddecl.inc"

main()
{
    int j=0, k=0, /* Variaveis auxiliares */
        escolha=0 /* Variavel para escolha de opcao */
        ;
    char NomeArq[20]; /* Vetor para nome de arquivo MFG */

    /* Imprime apresentaco inicial do programa */
    TelaApr();

    printf("Nome do arquivo MFG: ");
    gets(NomeArq);

    /* Inicia ponteiro para lista de transicoes */
    PtIn=InicTr();

    /* Carrega arquivo MFG */
    cargaMFG(NomeArq, PtIn);
```

```
while(1)
{
    FazMenu();
    escolha = getchar();
    getchar();

    switch(escolha)
    {
        case '1': /* SIMULACAO */
            /* Verifica transicoes habilitadas p/ marcacao inicial */
            PtIn=vertrans (PtIn);

            /* Ativa simulacao */
            Simula(PtIn);
            break;

        case '2': /* determinacao da MATRIZ DE INCIDENCIA */
            printf("*** Nboxes = %d\n", Nboxes);
            printf("*** Ntrans = %d\n", Ntrans);

            for (j=0; j < Nboxes; j++)
            {
                printf("B%-4d  ", (j+1));
                for(k=0; k < Ntrans; k++)
                    printf("%2d ", Minc[j][k]);
                printf("\n");
            }
            getchar();
            break;

        case '3':
            exit(0);
            break;
    }
}
}
```

4.5.1 Estrutura de dados

O elemento principal da estrutura de dados é o "record" (struct) TRANSIÇÃO, no qual existem outros elementos a ele associado como os elementos de entrada e saída e os gates. A listagem a seguir apresenta a definição da estrutura de dados utilizado nesta ferramenta.


```

/*
** -----
** Modulo.....: teetd.h
**
** Funcao.....: Contem definicoes das estruturas de dados
**
** Responsavel.: Julio Arakaki
**
** Observacoes.: Implementacao                               31/08/92
** -----
*/

/*
** Estrutura para armazenar os dados da transicao
** -----
*/
struct _transicao
{
    int          CodTr; /* Cod. da transicao */
    char         NomTr[20]; /* Nome da transicao (alto nivel) */
    int          iStatus; /* Indica se HABILITADO ou INIBIDO */
    struct _elemento *EleEnt; /* Pont. para lista de el. de entrada */
    struct _elemento *EleSai; /* Pont. para lista de el. de saida */
    struct _gate   *LisGat; /* Pont. para lista de gates */
    struct _transicao *ProxTr; /* Pont. para proximo el. da lista */
    struct _transicao *AntTr; /* Ponteiro para elem. anterior */
};
typedef struct _transicao TRANSICAO;

/*
** Estrutura para armazenar os dados dos elementos da transicao
** -----
*/
struct _elemento
{
    int          CodBox; /* Codigo do box */
    int          *AtrPas; /* Atributos a serem passados */
    struct _elemento *ProxEI; /* Ponteiro para prox. elemento */
    struct _elemento *AntEI; /* Ponteiro para el. anterior */
};
typedef struct _elemento ELEMENTO;

/*
** Estrutura para armazenar os dados dos boxes
** -----
*/
struct _box
{
    int          TipBox; /* Tipo do box */
    char         NomBox[20]; /* Nome do box (alto nivel) */
    int          Nmarca; /* Numero de marcas */
    int          iTempo; /* Para box de tempo */
    int          CapTemp; /* Capacidade ou temporizacao */
    int          VetAtr[20]; /* Vetor de atributos (maximo 20) */
};

```

```
    struct _marca    *PtMar; /* Vetor com atributos das marcas */
};
typedef struct _box BOX;

/*
** Estrutura para armazenar os dados das marcas
** -----
*/
struct _marca
{
    int CodMar;
    int VetAtr[20];
    struct _marca *ProxMar;
};
typedef struct _marca MARCA;

/*
** Estrutura para armazenar os dados dos gates
** -----
*/
struct _gate
{
    int CodBox;      /*Codigo do boxe */
    int Tipo_Gate;  /*Tipo do gate */
    struct _gate *ProxGt;
    struct _gate *AntGt;
};
typedef struct _gate GATE;
```

Neste caso, a estrutura de dados principal da ferramenta é uma lista duplamente ligada da estrutura TRANSICAO. A Figura 4.7 mostra a interligação entre dois elementos subseqüentes da lista. Cada nó (elemento) desta estrutura representa uma transição do MFG e os "boxes" e "gates" conectados nesta transição, são representado por sublistas. A Figura 4.8 mostra os elementos que compõem a estrutura TRANSICAO.

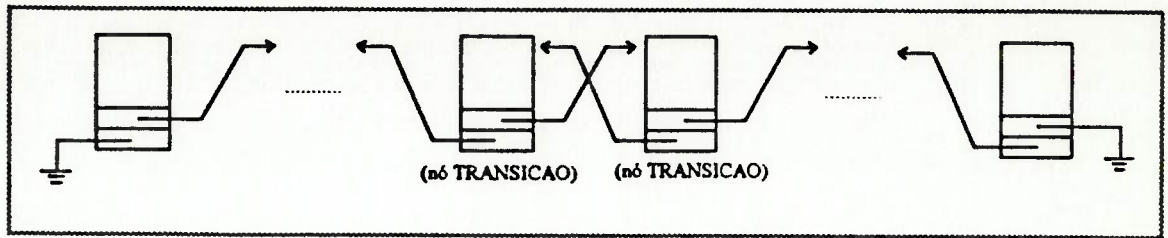


Fig. 4.7 Lista duplamente ligada de estruturas TRANSICAO.

campos	diagrama	descrição
CodTr		Código da transição
NomTr		Nome associado à transição
iStatus		Indica se a transição está HABILITADO ou INIBIDO
EleEnt	→	Ponteiro para lista de elementos de entrada da transição
EleSai	→	Ponteiro para lista de elementos de saída da transição
LisGat	→	Ponteiro para lista de gates conectadas à transição
ProxTr	→	Ponteiro para próximo elemento da lista
AntTr	→	Ponteiro para elemento anterior da lista

Fig.4.8 Descrição dos elementos da estrutura TRANSICAO.

4.5.2 Jogador de marcas

A listagem a seguir mostra um trecho da implementação do módulo "jogador de marcas" utilizado na ferramenta proposta.

```

/*
** -----
** Modulo:
**     tejogmar.c
**
** Funcoes:
**     vertrans - verifica transicoes.
**
** Observacoes:
**
** Responsavel      Data
** -----
** Julio Arakaki    25/06/92    Implementacao
**
** -----
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "teetd.h"

#define EXTERNO
#include "teglob.h"

#include "teddecl.inc"

/*
** -----
** Nome: vertrans
**
** Finalidade: Verifica transicoes, habilitando ou nao
**
** Parametros:
**     Saida:
**
**     Entrada:
**
** Retorno:
**
** Observacoes:
** -----

```

```

*/
TRANSICAO *
vertrans (PtIn)
TRANSICAO *PtIn;

{
  TRANSICAO *Ptaux=PtIn;
  ELEMENTO *PtEnt, *PtSai;
  GATE *PtGt;
  int iStatus;

  /* Percorre lista de transicoes */
  for (Ptaux=PtIn; Ptaux != NULL; Ptaux=Ptaux->ProxTr)
  {
    iStatus=TRUE;

    /* Verifica elementos de entrada */
    for(PtEnt=Ptaux->EleEnt; PtEnt != NULL; PtEnt=PtEnt->ProxEI)
    {

      /* Verifica se todos os boxes de entrada estao marcados */
      if(PtBx[PtEnt->CodBox-1].Nmarca <= 0)
      {
        iStatus = FALSE;
        break;
      }
      else
      {
        switch(PtBx[PtEnt->CodBox-1].TipBox)
        {
          case NL:
          case CA:
          case DN:
            break;
          case TM:
            if(PtBx[PtEnt->CodBox-1].iTempo !=
              PtBx[PtEnt->CodBox-1].CapTemp)
              iStatus = FALSE;
            break;
          case UP: /* n=N -> HABILITADO */
            if(PtBx[PtEnt->CodBox-1].Nmarca <
              PtBx[PtEnt->CodBox-1].CapTemp)
              iStatus=FALSE;
            break;
        }
      }
    }

    if (iStatus == FALSE)
    {
      Ptaux->iStatus = INIBIDO;
      ivetHab[Ptaux->CodTr]=-1;
    }
  }
}

```

```
        continue;
    }

    /* Verifica elementos de saída */
    for(PtSai=Ptaux->EleSai; PtSai != NULL; PtSai=PtSai->ProxEI)
    {
        /* Verifica se todos os boxes de saída estão desmarcados */
        if(PtBx[PtSai->CodBox-1].Nmarca > 0)
        {
            switch(PtBx[PtSai->CodBox-1].TipBox)
            {
                case NL:
                case TM:
                case DN:
                    iStatus = FALSE;
                    break;
                case CA:
                case UP: /* n < N -> HABILITADO */
                    if(PtBx[PtSai->CodBox-1].Nmarca >=
                       PtBx[PtSai->CodBox-1].CapTemp)
                        iStatus = FALSE;
                    break;
            }
        }
    }

    if (iStatus == FALSE)
    {
        Ptaux->iStatus = INIBIDO;
        ivetHab[Ptaux->CodTr]=-1;
        continue;
    }

    for(PtGt=Ptaux->LisGat; PtGt != NULL; PtGt=PtGt->ProxGt)
    {
        if((PtGt->Tipo_Gate == HABILITADOR) &&
           (PtBx[PtGt->CodBox-1].Nmarca == 0))
            iStatus = FALSE;

        if((PtGt->Tipo_Gate == INIBIDOR) &&
           (PtBx[PtGt->CodBox-1].Nmarca > 0))
            iStatus = FALSE;
    }

    if (iStatus == TRUE)
    {
        Ptaux->iStatus = HABILITADO;
        ivetHab[Ptaux->CodTr]=1;
    }
    else
    {
        Ptaux->iStatus = INIBIDO;
        ivetHab[Ptaux->CodTr]=-1;
    }
}
```

```

    )
    return PtIn;
}

/*
** -----
** Nome: DispHab
**
** Finalidade: Dispara as transicoes habilitadas
**
**
** Parametros:
**   Saída:
**
**   Entrada:
**
** Retorno:
**
** Observacoes:
** -----
*/
TRANSICAO *
DispHab(PtIn)
TRANSICAO *PtIn;
{
    TRANSICAO *Ptaux=PtIn;
    ELEMENTO *PtEnt, *PtSai;
    int iDisp=TRUE;

    while (Ptaux != NULL)
    {
        if (Ptaux->iStatus == HABILITADO)
        {
            /* Verifica elementos de entrada */
            for(PtEnt=Ptaux->EleEnt; PtEnt != NULL; PtEnt=PtEnt->ProxE1)
            {
                if(PtBx[PtEnt->CodBox-1].Nmarca > 0)
                {
                    switch(PtBx[PtEnt->CodBox-1].TipBox)
                    {
                        case NL:
                        case CA:
                        case DN: /* Dispensador */
                            PtBx[PtEnt->CodBox-1].Nmarca =
                                PtBx[PtEnt->CodBox-1].Nmarca - 1;
                            break;
                        case TM:
                            PtBx[PtEnt->CodBox-1].iTempo = 0;
                            PtBx[PtEnt->CodBox-1].Nmarca = 0;
                            break;
                        case UP: /* Agrupador */
                            PtBx[PtEnt->CodBox-1].Nmarca = 0;
                            break;
                    }
                }
            }
        }
    }
}

```

```
    }
    iDisp=TRUE;
  }
  else
    iDisp=FALSE;
}

/* Verifica elementos de saida */
for(PtSai=Ptaux->EleSai; PtSai != NULL; PtSai=PtSai->ProxEl)
{
  if(iDisp == TRUE)
  {
    ivetHab[Ptaux->CodTr]=1; /* Preenche vetor de habilitacao */
    switch(PtBx[PtSai->CodBox-1].TipBox)
    {
      case TM:
        PtBx[PtSai->CodBox-1].iTempo = 1;
        PtBx[PtSai->CodBox-1].Nmarca = 1;
        break;
      case NL:
      case CA:
      case UP:
        PtBx[PtSai->CodBox-1].Nmarca =
          PtBx[PtSai->CodBox-1].Nmarca+1;
        break;
      case DN:
        PtBx[PtSai->CodBox-1].Nmarca =
          PtBx[PtSai->CodBox-1].CapTemp;
        break;
    }
  }
}
Ptaux=Ptaux->ProxTr;
}
return PtIn;
}

/*
** -----
** Nome: AtuaTmp
**
** Finalidade: Atualiza boxes de tempo
**
** Parametros:
**   Saida:
**
**   Entrada:
**
** Retorno:
**
```



```
** Observacoes:
**
** -----
*/
AtuaTmp()
{
    int i=0;

    printf("\n");

    for(;;)
    {
        if(PtBx[i].TipBox >= 0 && PtBx[i].TipBox == TM && PtBx[i].Nmarca > 0)
        {
            PtBx[i].iTempo = PtBx[i].iTempo + 1;
        }
        else if (PtBx[i].TipBox < 0)
            break;
        i++;
    }
}
```

Capítulo 5

EXEMPLO DE APLICAÇÃO DA FERRAMENTA

5.1 INTRODUÇÃO

O objetivo deste capítulo é demonstrar uma aplicação da ferramenta proposta neste trabalho. Descreve-se um modelo em MFG/PFS de um sistema de manufatura do tipo "flow shop" composto por estações de processamento de peças e um robô para realizar o transporte dessas peças.

Observa-se que neste exemplo, onde apenas um transportador é utilizado, existe um conflito devido a necessidade do compartilhamento do recurso transporte. O robô deve atender às solicitações de transporte que podem ocorrer de maneira simultânea. Desta forma, é necessário determinar adequadamente qual será a estação que deve ser atendida. A ferramenta permite através do Árbitro Externo, realizar uma inferência sobre um conjunto de regras pré-estabelecidas, que representam as condições necessárias para priorizar essa escolha, utilizando como base de conhecimentos a marcação do grafo MFG

que representa o estado do sistema. Portanto, os conflitos existentes no exemplo, são resolvidos através da aplicação do conceito de árbitro externo proposto na ferramenta.

5.2 DESCRIÇÃO ESTRUTURAL E FUNCIONAL DO SISTEMA EM ESTUDO

A Figura 5.1 mostra a estrutura básica (arranjo físico dos equipamentos) de um sistema para produção de peças.

Este sistema do tipo "flow shop" é composto por:

- uma estação de entrada (EEN);
- duas estações para processamento (ET1 e ET2);
- uma estação de medição (EME) e;
- um robô transportador (RBT).

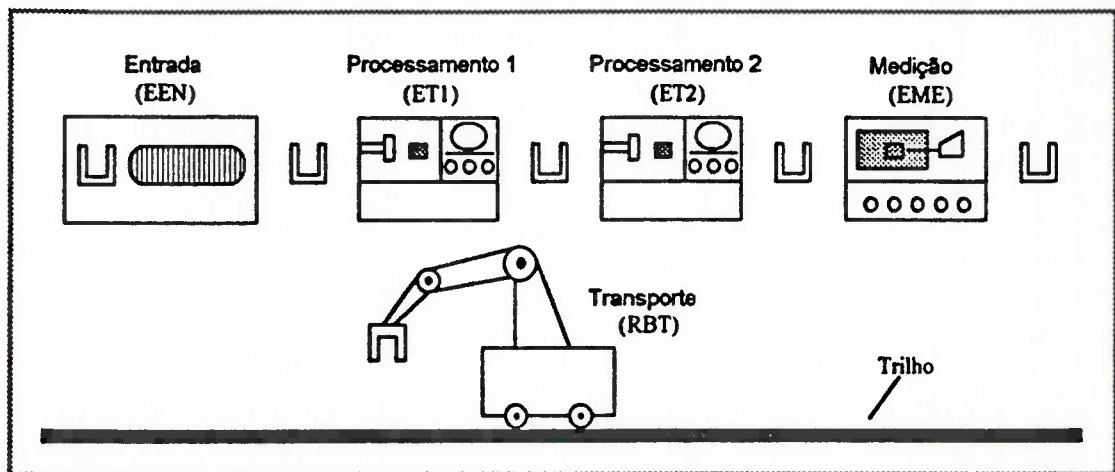


Fig. 5.1 Estrutura do sistema-exemplo

De acordo com a Figura 5.1, verifica-se que as estações ET1, ET2 e EME, possuem magazines de entrada e saída, que são utilizados como "buffers" de armazenamento temporário para integrar de forma adequada o sistema de produção.

Todas as estações deste sistema possuem a capacidade de processar apenas uma peça de cada vez. Da mesma forma, o robô pode carregar somente uma peça. Nos itens seguintes, descreve-se em maiores detalhes o funcionamento das estações e do sistema de transporte.

5.2.1 Estação de entrada

A estação de entrada EEN corresponde à estação de início de processo, composta por um magazine de entrada e uma esteira que desloca uma peça até o ponto de carga de onde o robô pega a peça e inicia o deslocamento para colocar na estação seguinte.

Esta estação é utilizada também como estacionamento do robô, ou seja, ela é utilizada como uma posição de repouso do robô quando não existir nenhuma operação de transporte para ser realizada.

5.2.2 Estações de processamento 1 e processamento 2

A estação de processamento 1 (Processo_1) é composta por um torno CNC, um magazine ("buffer") de entrada com capacidade de $N=4$ peças, utilizado para armazenamento de peças à espera do processamento e um magazine de saída com a mesma capacidade N utilizado para o armazenamento de peças já processadas e a espera do transporte. A estação recebe uma peça no magazine de entrada e, se esta estiver

pronta para realizar um processamento, então a peça é retirada do magazine e colocada no torno CNC através de um manipulador automático.

A estação de processamento 2 (Processo_2) possui uma fresa CNC e os magazines de entrada e saída, com a mesma capacidade e funcionalidade descritas na estação de processamento 1. As operações de carga e descarga são semelhantes às definidas para a estação de processamento 1 (Processo_1).

5.2.3 Estação de medição

Esta estação é composta por uma máquina de medição e também pelos magazines de entrada e saída com capacidade de $N=4$ peças. A função desta estação no processo, é a de medir as dimensões das peças que já foram processadas nas estações Processo_1 e Processo_2.

Semelhante às estações de processamento Processo_1 e Processo_2 descritas no item anterior, as operações de carga do magazine de entrada e descarga para o magazine de saída são realizadas automaticamente.

Esta estação é a responsável pela realização da última tarefa no processo. Ou seja, terminada a operação de medição, a peça já está pronta para ser colocada no magazine de saída do processo (peça acabada).

5.2.4 Sistema de Transporte

O sistema para transporte de peças é composto por um robô do tipo manipulador que se desloca sobre um trilho, garantindo, desta forma, um maior campo de ação. Este trilho permite o deslocamento do robô ao longo da linha de produção e,

conseqüentemente, o acesso à todas as estações do processo, permitindo a realização das operações de carga e descarga das peças nas estações.

Para realizar uma tarefa de transporte entre duas estações consecutivas do sistema, o robô deve sempre realizar as seguintes operações:

- 1) deslocar-se para a estação solicitante;
- 2) pegar a peça do magazine de saída da estação;
- 3) deslocar-se transportando a peça e;
- 4) colocar a peça na estação seguinte.

Portanto, é evidente que a partir do momento em que haja uma peça disponível, em algum magazine de saída de qualquer estação, esta estação torna-se solicitante.

5.2.5 Comportamento dinâmico do sistema integrado

O estado inicial é definido com o robô na estação de entrada (que corresponde ao estacionamento) e peças no magazine de entrada do sistema, prontas para serem processadas.

O processo inicia-se com o robô transportando uma peça para o magazine de entrada da estação de processamento 1. Em seguida, esta peça é transportada para a estação de processamento 2, e assim por diante. Desta forma, à medida que uma estação termina o seu processamento, a peça é colocada no magazine de saída correspondente, estando pronta para ser transportada para a estação seguinte. Neste contexto, o robô sempre transporta as peças do magazine de saída de uma estação para o magazine de entrada da estação consecutiva. Isto ocorre para todas as estações, desde a entrada da

peça no início do processo até a saída do produto acabado a ser armazenado no magazine de saída.

À medida que o processo evolui, as solicitações começam a tornar-se aleatórias em função dos tempos de transporte e de processamento. Portanto, o robô deve atender estas solicitações de acordo com uma determinada prioridade adequada para que o comportamento dinâmico do sistema seja estabelecido de forma consistente, ao longo do tempo.

As operações de carga de uma peça do magazine de entrada para a estação correspondente e a descarga de uma peça desta estação para o seu respectivo magazine de saída são realizadas automaticamente por dispositivos mecânicos (manipuladores) acoplados às estações.

Após o início do processo, se num determinado instante não ocorrer nenhum pedido de transporte por parte das estações, o robô se desloca para o estacionamento (estação de entrada), aguardando uma nova solicitação de transporte.

As máquinas associadas às estações Processo_1, Processo_2 e Medição, podem processar apenas uma peça de cada vez. Os magazines de entrada e saída destas estações são utilizadas para otimizar esse processamento. Ou seja, se uma estação estiver processando alguma peça e existir uma nova peça a espera de processamento, esta nova peça é colocada no magazine de entrada, até que a estação seja liberada. Da mesma forma, quando a estação termina o processamento sobre uma determinada peça e o robô não está disponível para a realização deste transporte, então a peça é colocada no magazine de saída da estação, liberando-a para o processamento de uma nova peça. Desta maneira, a capacidade dos magazines de $N=4$ peças, permite que estas estações processem até 8 peças.

Portanto, o problema de especificação da capacidade necessária para cada magazine de armazenamento temporário está vinculado ao problema de balanceamento da linha flexível de produção, de acordo com o produto a ser processado. A utilização de um trilho, incrementando o campo de ação do robô, objetiva otimizar o tempo de utilização desta unidade. Em contrapartida, necessita-se de magazines para armazenar as peças durante o intervalo de tempo em que o robô não está disponível. Esta solução parece mais adequada quando comparada à alternativa de alocar-se um robô para cada estação. Neste caso, além do problema de sub-utilização do recurso robô, há o problema de necessitar-se de magazines, se o tempo de processamento das estações forem diferentes. Pois caso contrário, o problema de sub-utilização propaga-se para as estações de processamento.

Neste contexto, a arquitetura proposta possibilita, inclusive, o processamento de um "mix"¹⁵ de produtos que pode otimizar o tempo de utilização de cada uma das estações.

O processamento das peças é finalizado no sistema em estudo quando: uma peça alcança o magazine de saída da estação de medição (última do sistema), o robô transporta esta peça para o magazine de saída do processo (estoque de produtos acabados). Em seguida, o robô se desloca para atender outra solicitação ou retorna para o estacionamento, caso não exista nenhuma solicitação.

5.3 MODELO MFG/PFS DO SISTEMA

Neste item, apresenta-se uma modelagem do exemplo através da metodologia MFG/PFS.

¹⁵ Variedade.

Como visto no capítulo 2, essa metodologia permite o uso dos recursos de estruturação e hierarquização. Desta maneira, o modelo do sistema é desenvolvido através da técnica "top-down", onde a partir de um modelo conceitual do sistema (PFS) obtém-se um modelo funcional (MFG) a partir de detalhamentos deste modelo, em sub-atividades.

5.3.1 Modelo PFS do sistema

A Figura 5.2 representa um modelo em PFS do exemplo dado na Figura 5.1 e a sua associação com o Árbitro Externo. Neste modelo verifica-se também, a atividade de transporte sendo compartilhada pelas estações do processo.

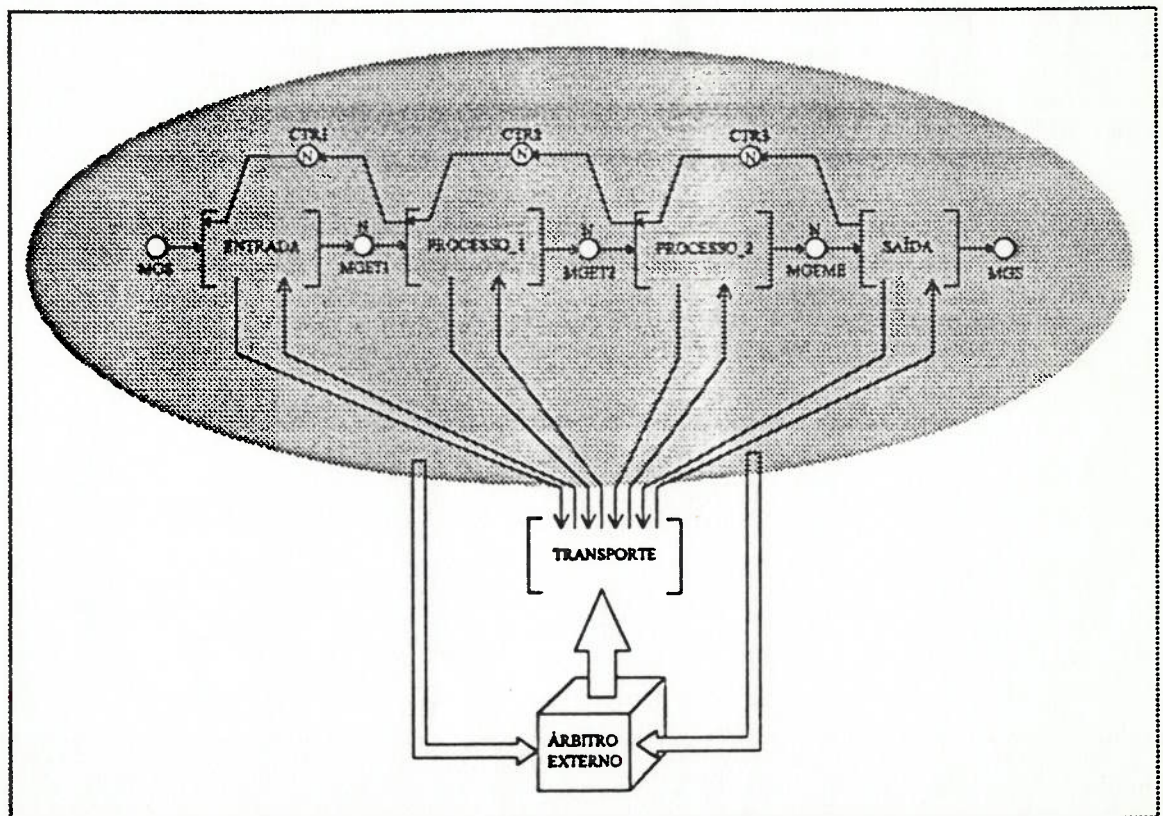


Fig. 5.2 Modelo conceitual do sistema.

As setas que entram no Árbitro Externo, indicam a capacidade de atualização do estado do sistema, através da marcação. Isto permite que este árbitro consiga armazenar os estados anteriores e atuais. A seta que sai do árbitro indica a sua atuação para a tomada de decisões (escolha de solicitações) para a solução do conflito devido ao compartilhamento do sistema de transporte.

No modelo da Figura 5.2 o "box" MGE corresponde ao magazine de entrada do sistema e MGS ao magazine de saída, os "boxes" MGET1, MGET2 e MGEME são respectivamente os magazines de entrada da estação de processamento 1 (Processo_1), processamento 2 (Process_2) e a máquina de medição (Medição). O "box" CTR1 é utilizado para evitar que a estação de entrada (Entrada) solicite transporte quando MGET1 estiver com toda a capacidade utilizada. De forma análoga, o "box" CTR2 para MGET2 e CTR3 para MGEME.

5.3.2 Atividade ENTRADA

O detalhamento da atividade ENTRADA é apresentado na Figura 5.3. Onde são utilizados os elementos:

Os "boxes" STEN para indicar a solicitação de transporte na estação de entrada, DEN para indicar o deslocamento do robô de uma posição qualquer até a entrada e TET1 para indicar a realização do transporte da estação ENTRADA para a estação PROCESSO_1.

Os arcos PEN utilizado para indicar o envio do pedido de transporte para a atividade SELECIONA_PEDIDO (detalhada na Figura 5.8), ETR para indicar o início do transporte de um peça e FTR para indicar o final de transporte. Os gates AEN que indica

o atendimento ao pedido de entrada e FDES o final do deslocamento são provenientes de boxes definidos na atividade DESLOCA (Figura 5.9).

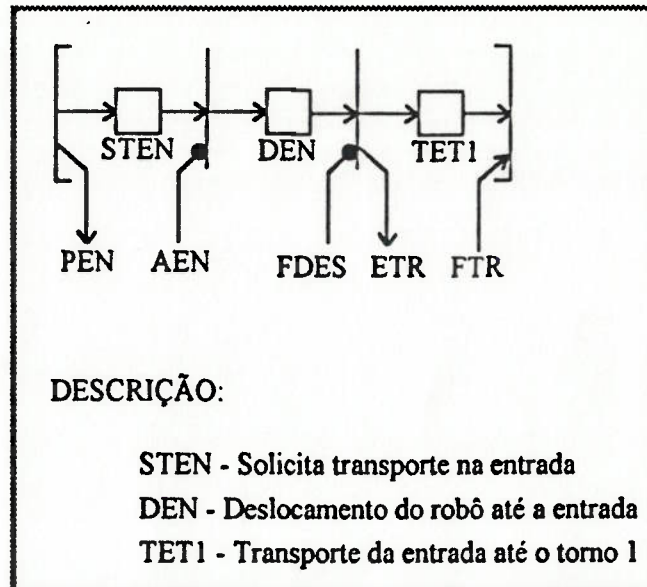


Fig. 5.3 Detalhamento da atividade ENTRADA.

5.3.3 Atividade PROCESSO_1

Detalhando-se a atividade PROCESSO_1 (torno 1) tem-se a representação dada na Figura 5.4.

O "box" temporizado PRT1 é utilizado para indicar o tempo em que a peça irá permanecer nesta estação, ou seja, o tempo para o processamento de uma peça.

MGST1 é um "box" de capacidade que representa o magazine de saída desta estação, onde são colocadas as peças já processadas para ficar a espera de transporte.

Os "boxes" STT1, DT1 e TT1T2 tem funções semelhantes aos "boxes" STEN, DEN e TET1 da atividade ENTRADA, sendo que neste caso correspondem a solicitação

de transporte no PROCESSO_1, deslocamento do robô até o PROCESSO_1 e transporte de uma peça do PROCESSO_1 até o PROCESSO_2.

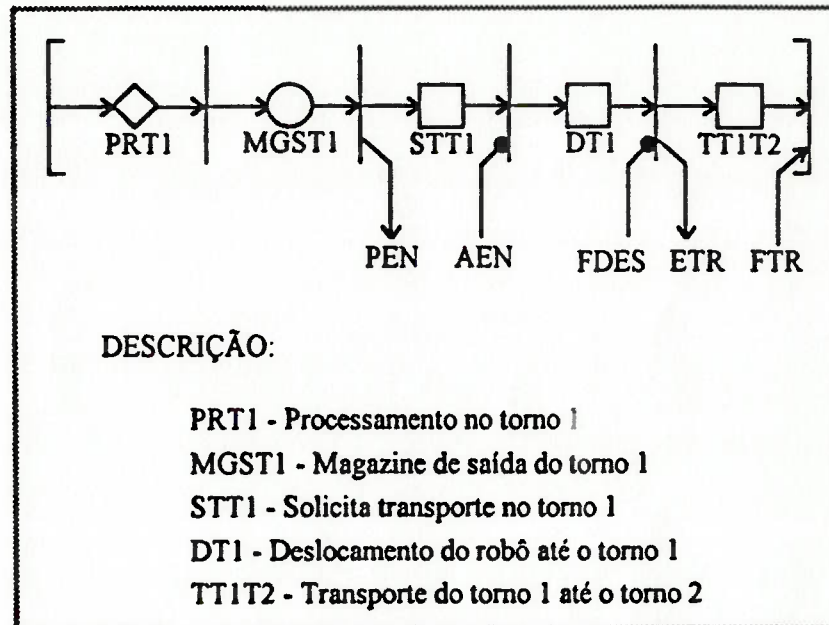


Fig. 5.4 Detalhamento da atividade PROCESSO_1.

5.3.4 Atividade PROCESSO_2

A Figura 5.5 mostra o detalhamento da atividade PROCESSO_2 cuja descrição dos elementos que compõem esta atividade é análoga a dos elementos do PROCESSO_1. Ou seja, o "box" PRT2 indica o tempo de processamento na estação PROCESSO_2, MGST2 é o magazine de saída da estação PROCESSO_2, STT2 indica a solicitação de transporte pelo PROCESSO_2, DT2 é o deslocamento do robô até o PROCESSO_2 e TT2ME é o transporte do PROCESSO_2 até a estação MEDIÇÃO.

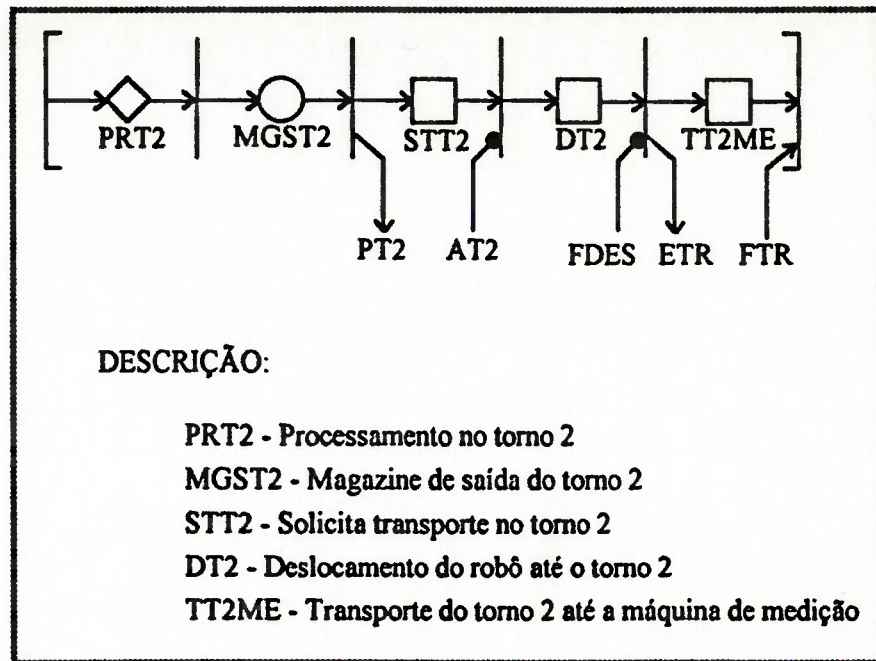


Fig.5.5 Detalhamento da atividade PROCESSO_2.

5.3.5 Atividade MEDIÇÃO

A atividade MEDIÇÃO é detalhada na Figura 5.6.

A descrição dos elementos é semelhante às atividades PROCESSO_1 e PROCESSO_2.

Esta atividade representa a última estação do processo. Neste caso, assim que a máquina termina o seu processamento, a peça é colocada pelo transportador no magazine de saída do processo MGS.

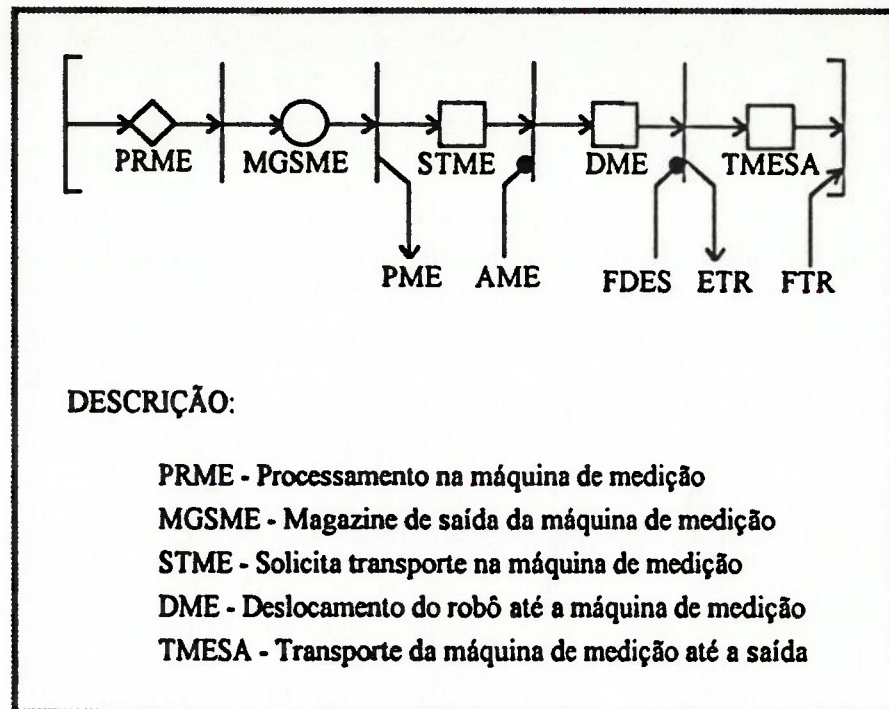


Fig.5.6 Detalhamento da atividade MEDIÇÃO.

5.3.6 Atividade TRANSPORTE

A atividade TRANSPORTE é composta pelas sub-atividades SELECIONA_PEDIDO, DESLOCA e REALIZA_TRANSPORTE. A Figura 5.7 mostra o modelo da atividade TRANSPORTE e a representação das sub-atividades componentes. O "box" RBDIS marcado, indica a disponibilidade do robô para realizar o transporte. Assim que se inicia a atividade TRANSPORTE, a marca em RBDIS é retirada e só é devolvida quando termina a sub-atividade REALIZA_TRANSPORTE.

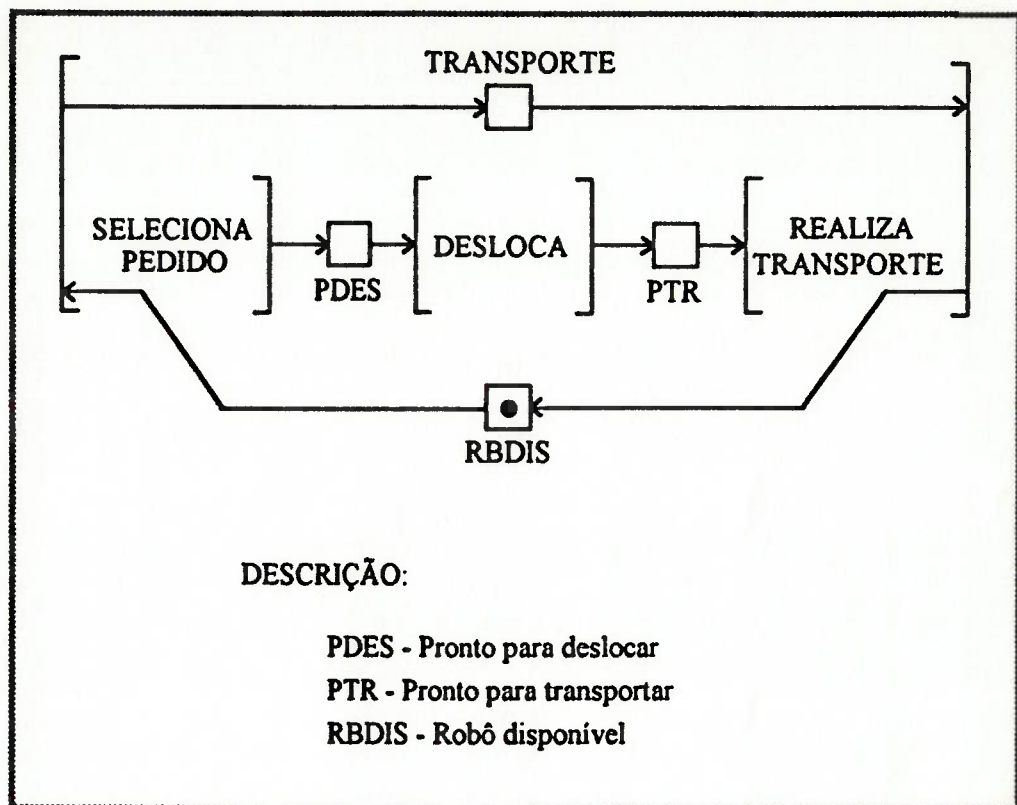


Fig.5.7 Detalhamento da atividade de TRANSPORTE.

5.3.7 Atividade SELECIONA_PEDIDO

Os "boxes" PEN, PT1, PT2 e PME, na Figura 5.8, são utilizados para a solicitação de transporte. Por exemplo, se PT1 e PME estiverem marcados, significa que existe a solicitação de transporte pelo torno 1 e também pela máquina de medição. Estas marcas são provenientes das atividades PROCESSO_1 e MEDIÇÃO, através do disparo das transições correspondentes aos eventos para solicitação de transporte.

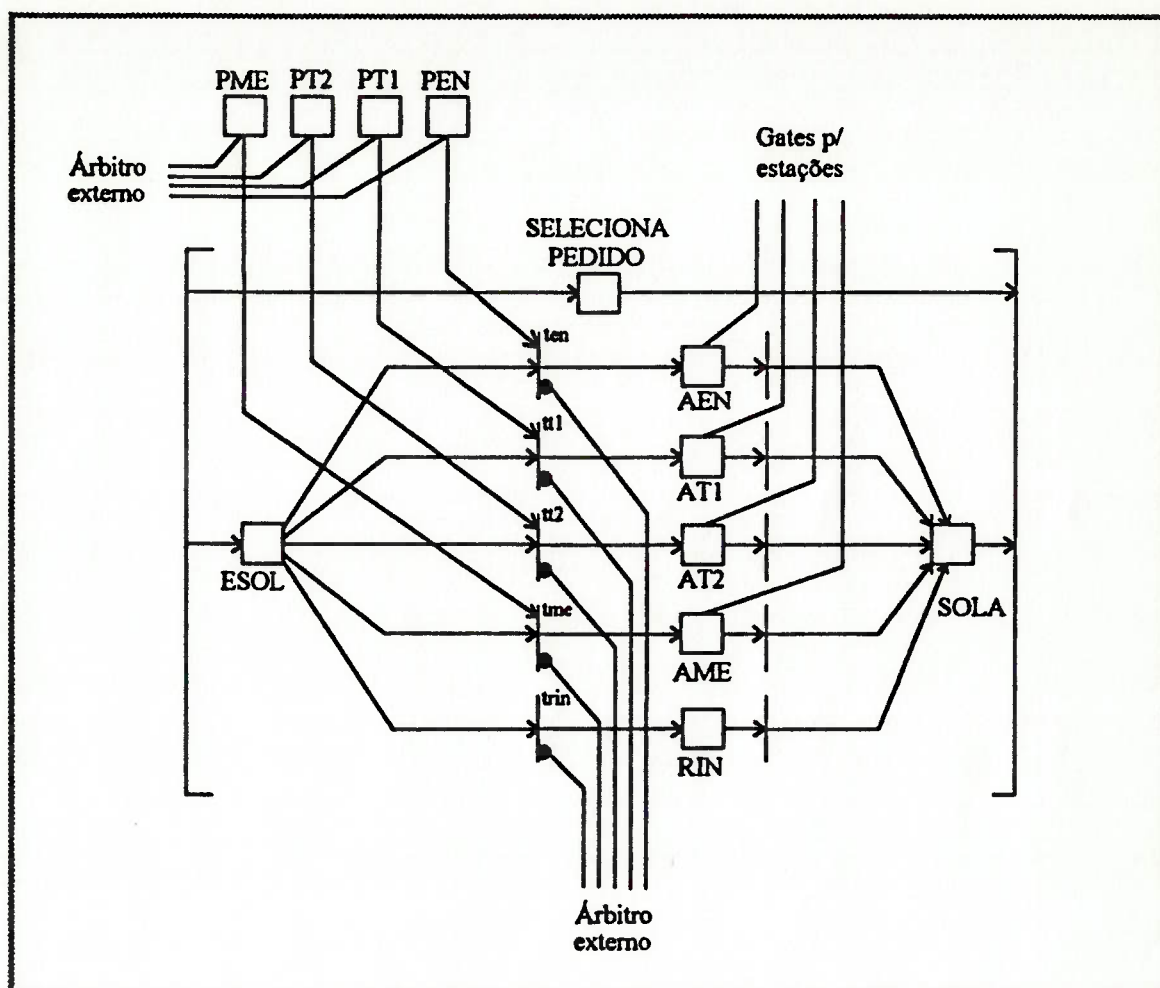


Fig. 5.8 Detalhamento da atividade SELECIONA_PEDIDO.

Os "boxes" PEN, PT1, PT2 e PME estão associados ao árbitro externo para possibilitar que este árbitro consiga armazenar a posição ou o estado anterior do transportador, através de suas marcações.

Se os "boxes" PT1, PME e também ESOL estiverem marcados (há duas solicitações: no torno 1 e na máquina de medição), as transições tt1, e tme estarão habilitados no mesmo instante. Neste caso verifica-se a existência de um conflito, que será solucionado através da escolha da transição que será disparada (escolha de qual será

a estação atendida). As portas conectadas às transições *ten*, *tt1*, *tt2* e *tme* são associadas ao Árbitro Externo que envia os sinais através destas portas para solucionar este conflito.

O "box" *ESOL* representa a espera de uma solicitação, os gates habilitadores que saem dos boxes *AEN*, *AT1*, *AT2* e *AME* para as atividades *ENTRADA*, *PROCESSO_1*, *PROCESSO_2* e *MEDIÇÃO* respectivamente, são utilizados para indicar qual a estação foi escolhida para atendimento pelo transportador. O "box" *SOLA* indica que a solicitação já foi atendida. O "gate" (porta) associado à transição *trin* corresponde ao retorno do transportador ao início do processo (estação *ENTRADA*), quando não houver nenhuma solicitação de transporte.

É importante notar que essa atividade *SELECIONA_PEDIDO* só inicia se o transportador estiver disponível. Isto é representado pelo box *RBDIS*, na Figura 5.7, conectado à transição de entrada da atividade *TRANSPORTE*.

5.3.8 Atividade DESLOCA

O detalhamento da atividade *DESLOCA* apresentada na Figura 5.9 mostra um outro conflito relacionado com o deslocamento do transportador para atender a uma solicitação de transporte. Os tempos associados aos "boxes" *T2*, *T4* e *T6* respectivamente, indicam os tempos necessários para o deslocamento entre duas estações subseqüentes, deslocamento pulando-se uma estação e deslocamento pulando-se duas estações. A marca fica retida nesses "boxes" durante o tempo a ele associado. O Árbitro Externo resolve este conflito através dos gates habilitadores conectados às transições *tt2*, *tt4* e *tt6*. A escolha de qual será o tempo de deslocamento (disparo de *tt2*, *tt4* ou *tt6*) é feita de acordo com a escolha da solicitação a ser atendida, feita na atividade *SELECIONA_PEDIDO*. O "box" *EDES* indica a espera de deslocamento, e *FDDES*,

quando marcado, representa o fim do deslocamento e envia um sinal para as estações, através dos gates a ele associado, habilitando a transição para o início de transporte.

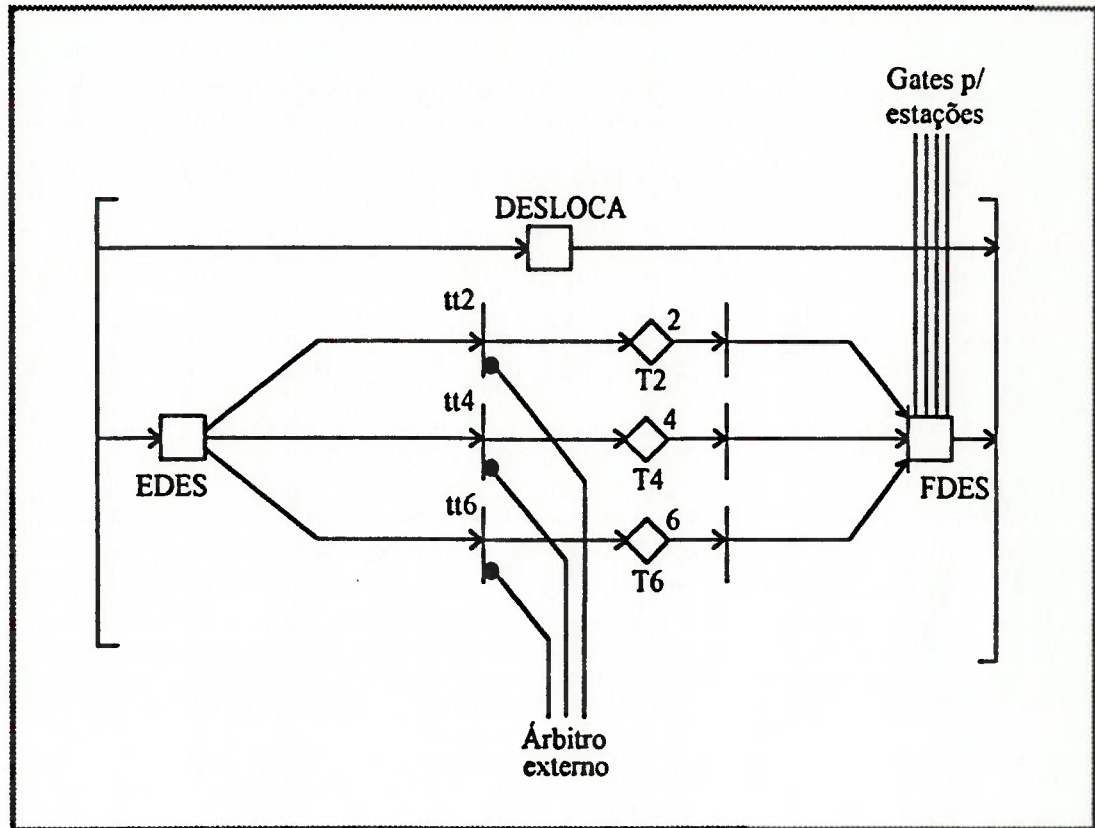


Fig. 5.9 Detalhamento da atividade DESLOCA

5.3.9 Atividade REALIZA_TRANSPORTE

Terminado o deslocamento até a estação escolhida para o atendimento, o robô está pronto para fazer o transporte da peça para a estação seguinte. O tempo de transporte é sempre 2 unidades de tempo, pois sempre a peça é levada para a estação subsequente à estação solicitante. A Figura 5.10 apresenta o detalhamento da atividade REALIZA_TRANSPORTE. Neste caso, o box de tempo com 2 unidades de tempo

associado representa a realização do transporte de uma peça. Nesta atividade existe um conflito que o árbitro externo também resolve. Ele ocorre por causa da necessidade de escolha entre realizar o transporte e retornar para o início do processo. O disparo da transição *ttr1* corresponde à realização de transporte e o disparo de *ttr2* o deslocamento do robô até o início do processo. *ETRA* indica esperando transporte e *TRAC* transporte concluída.

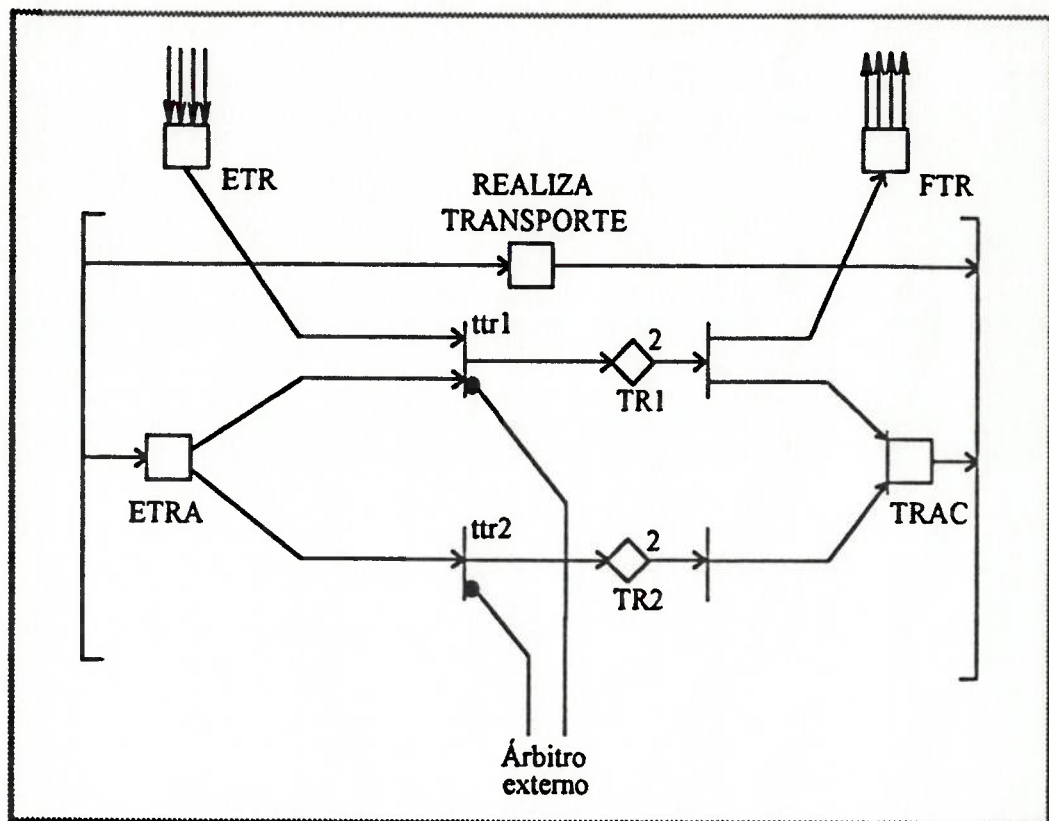


Fig. 5.10 Detalhamento da atividade *REALIZA_TRANSPORTE*.

A utilização do Árbitro Externo permite melhorar a capacidade de modelagem da metodologia MFG/PFS através da simplificação do modelo do subgrafo responsável pela solução dos conflitos. Ou seja, sem o Árbitro Externo, todas as condições (regras) necessárias para a solução dos conflitos devem ser modelados através dos grafos

MFG/PFS acarretando num aumento significativo da complexidade do modelo (vide item 5.4).

Se fosse utilizado mais de um transportador, no exemplo da Figura 5.1, o conjunto de regras que definem as condições para o compartilhamento de transporte aumenta de maneira significativa. Desta maneira, se essas condições fossem implementadas com o grafo MFG/PFS o modelo ficaria bastante complexo e de difícil interpretação.

5.4 ÁRBITRO EM MFG

Para demonstrar a complexidade do modelo contendo o árbitro responsável pela solução do conflito devido ao compartilhamento de recursos, apresenta-se na Figura 5.11 uma implementação do árbitro em grafos MFG. Este modelo, foi implementado seguindo-se apenas um tipo de prioridade: atender primeiro a estação mais próxima da saída.

Os principais componentes deste modelo são:

- SEN, ST1, ST2 e SME são responsáveis pela solicitação de transporte,
- DEN, DT1, DT2 e DME são "boxes" do tipo dispensor, a sua capacidade indicada por N no grafo indica a estação para onde o robô deverá se deslocar.
- A, B, MAB, IAB, GPAT são "boxes" o tipo capacidade onde, A indica a estação a ser atendida, B a posição atual do robô, MAB maior entre A e B, IAB a intersecção entre A e B e GPAT grava a posição atual do robô.

A marcação inicial nos "boxes" RBDIS e B indicam, respectivamente, que o robô está disponível e está na posição de estacionamento (estação EEN) devido a existência de apenas uma marca.

A prioridade neste caso, pode ser exemplificada da seguinte forma: se as estações ET1 e ET2 solicitam transporte, a prioridade de atendimento é maior para a estação ET2. Esta prioridade é definida pelos "gates" que saem dos "boxes" SEN, ST1, ST2 e SME e são conectados nas transições *ten*, *tt1*, *tt2*, *tme*.

Para explicar o funcionamento do grafo, pode-se considerar a seguinte situação: o estado inicial do sistema corresponde à marcação dada pela Figura 5.11 e existe solicitação pela estação ET1 e ET2. Neste caso, verifica-se que os "gates" inibidores que saem de ST2, desabilitam as transições *ten*, *tt1*. Desta maneira, a única transição habilitada para o disparo é *tt2*. Após o disparo desta transição, a marca de RBDIS (robô disponível) é retirada e surge uma marca em DT2, habilitando a transição de saída deste box. Esta transição dispara 3 vezes ($N=3$) colocando 3 marcas em A.

Neste instante, verifica-se que a transição *ct2* fica habilitada e neste caso só dispara uma vez colocando uma marca em MAB e IAB. Depois do disparo de *ct2*, apenas a transição *ct1* fica habilitada que dispara 2 vezes devido às 2 marcas que restaram em A. O disparo da transição *tcon*, coloca marcas em GPAT e MAB. Os "boxes" MAB, GPAT e IAB ficam, respectivamente, com 2, 2 e 1 marcas. A transição *sorvedoura*¹⁶ *epos*, fica habilitada.

Depois que a transição *epos* dispara, a atividade Deslocamento está pronta para ser realizada, pois a marcação de MAB e IAB habilitam a transição de entrada desta atividade. Verifica-se que o "gate" inibidor de MAB até a transição de saída da atividade Deslocamento, força a realização desta atividade enquanto houver marcas em MAB. Neste caso, a atividade é realizada 2 vezes, correspondendo o deslocamento do robô da posição atual (estação EEN) até a estação a ser atendida (estação ET2). Após o deslocamento, o robô realiza a atividade Transporte.

¹⁶ O disparo deste tipo de transição faz as marcas desaparecerem.

Quando a transição de saída da atividade Transporte dispara (fim do transporte) é colocado mais uma marca em GPAT e também uma marca em CATU, habilitando, desta maneira, a transição tatu que dispara o número de vezes correspondente ao número de marcas em GPAT, atualizando desta forma o "box" B que indica a posição atual do robô. No final, após a atualização da posição atual do robô, a transição tdis dispara, colocando o robô novamente a disposição para realizar uma nova tarefa.

Observa-se que neste modelo não foi implementado o retorno do robô para o estacionamento, quando não existir nenhuma solicitação de transporte.

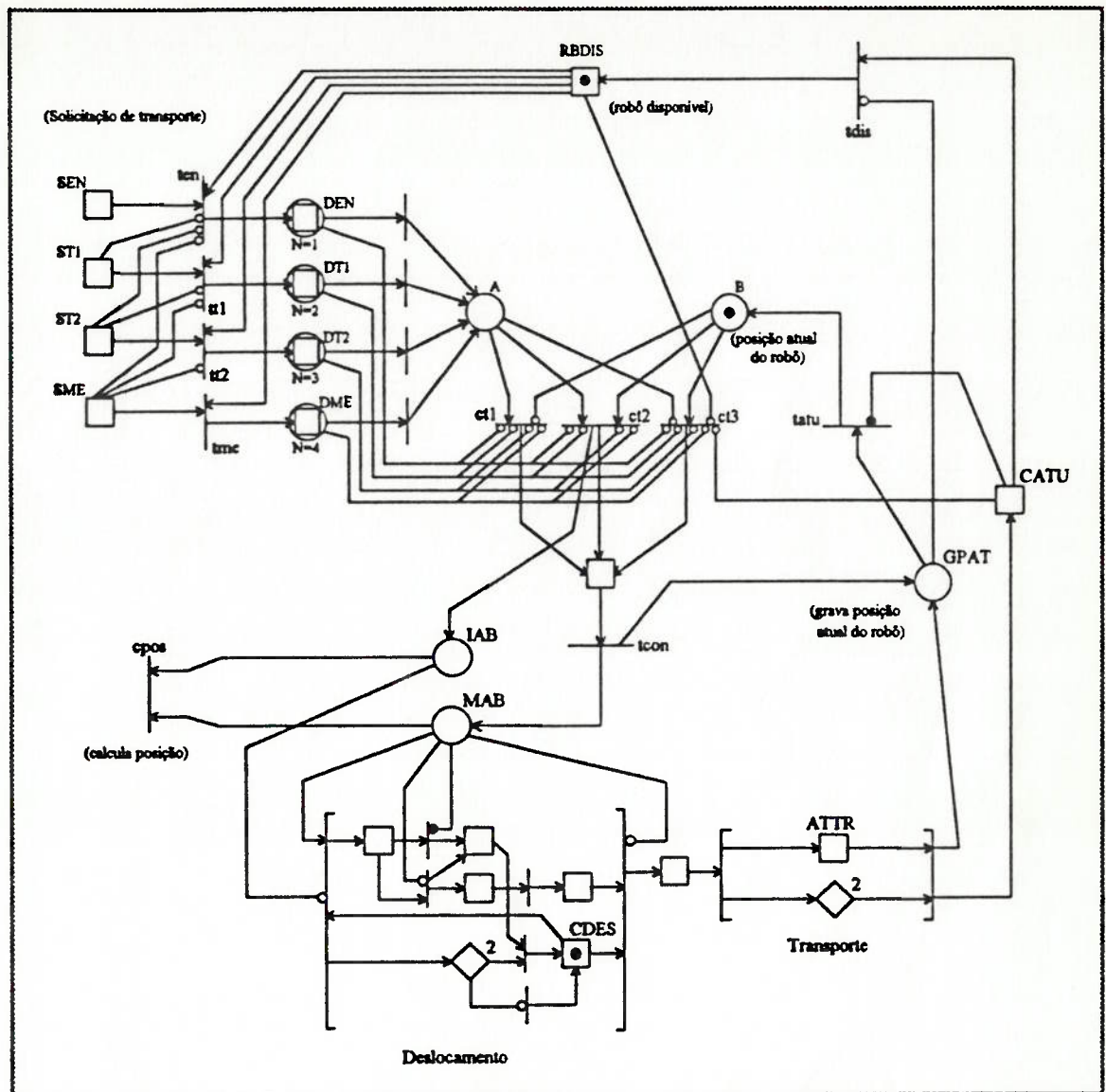


Fig. 5.11 Implementação do árbitro através do MFG.

5.5 ÁRBITRO EXTERNO BASEADO EM REGRAS

Neste exemplo, verifica-se que o sistema de transporte deve "sempre que possível" atender às solicitações de transporte das estações ENTRADA, PROCESSO_1, PROCESSO_2 e MEDIÇÃO. Porém, quando ocorre solicitações simultâneas e sendo o

transporte deste sistema composto por apenas um robô, existe a necessidade de se definir algumas prioridades para o atendimento. Desta forma, utiliza-se um árbitro baseado em um conjunto de regras que definem as condições necessárias para a solução deste conflito e conseqüentemente, definir prioridades para o atendimento. Assim sendo, este árbitro é o responsável pela escolha de qual será a estação a ser atendida primeiro num determinado instante em que ocorram solicitações simultâneas.

Neste árbitro baseado em regras, considera-se como fatos ou base de conhecimentos como sendo o estado atual do sistema (marcação dos boxes do modelo do sistema em grafos MFG). E a escolha das opções ou inferência de um resultado, é realizada a partir da aplicação dessas regras sobre a base de conhecimentos.

Desta maneira, pode-se considerar este árbitro como uma máquina de inferência que permite o fornecimento de uma solução (se ela existir) a partir de um conjunto de regras e dos fatos.

No item seguinte, descreve-se as principais regras que podem ser utilizadas para a implementação árbitro para resolver os conflitos que surgem neste exemplo, em estudo.

5.5.1 Regras para o Árbitro Externo

Para este sistema de manufatura em estudo, as regras que definem as condições necessárias para a escolha de qual será a estação solicitante que será atendida pelo robô são:

- 1) se o robô está disponível e existe apenas uma estação solicitante, então atende a estação.

- 2) **se** dentre várias solicitações, uma delas é a última do processo, **então** atende a estação do final do processo
- 3) **se** o robô está próximo da estação solicitante e se existem mais de uma solicitação, **então**, o robô atende a estação mais próxima.
- 4) **se** não há solicitações, **então** o robô retorna ao início do processo

Verifica-se, portanto que existe uma grande vantagem de se utilizar o árbitro externo baseado em regras. Pois, todas as condições e prioridades necessárias para a solução de conflitos podem ser colocadas dentro deste árbitro em forma de um conjunto de regras. Em contra partida, quando se utiliza os grafos MFG é necessário a implementação de um modelo contendo todas estas condições e prioridades para solucionar os conflitos.

Capítulo 6

OBSERVAÇÕES FINAIS

O objetivo deste trabalho foi a proposta de utilização de sistemas baseado em regras de Produção associado à metodologia MFG/PFS e também a especificação e a implementação de uma ferramenta de simulação e análise para Sistemas Integrados de Manufatura (SIM).

Para atingir estes objetivos, investigou-se as características dos sistemas de manufatura, cujo comportamento dinâmico baseia-se na ocorrência de eventos discretos (SED) e verificou-se as necessidades para o desenvolvimento de sistemas autônomos com o conceito de antropocentrismo incorporado. Desta maneira, apresentou-se as características, propriedades e elementos estruturais das redes de Petri e da metodologia MFG/PFS e a importância da utilização de uma abordagem hierárquica e estruturada ("top-down") para a modelagem e especificação do controle para os sistemas de manufatura.

Para verificar a necessidade de simulação para a análise dos SIM, investigou-se os fenômenos indesejáveis como o "deadlock" e a colisão que podem ocorrer nestes

ambientes e, algumas técnicas que possibilitam solucioná-los. Neste sentido, propôs-se uma especificação e implementação de uma ferramenta protótipo para simulação e análise dos sistemas de manufatura. Este protótipo utiliza a metodologia MFG/PFS para a modelagem e especificação do controle e para solucionar os conflitos, utiliza-se um sistema baseado em regras.

A vantagem de utilizar um sistema baseado em regras é demonstrada a partir da implementação do modelo do árbitro responsável pela solução dos conflitos (devido a alocação dos recursos de "transporte") em grafos MFG/PFS e a respectiva implementação deste árbitro externo através de regras de Produção. Neste caso, verifica-se que há uma sensível simplificação no modelo do sistema, facilitando a sua interpretação.

A modularidade e a estruturação da ferramenta proposta, permite uma flexibilidade, onde, torna-se fácil realizar alterações ou expansões. Por exemplo, pode-se substituir o módulo ("árbitro externo") por outras técnicas derivadas da Inteligência Artificial, para a representação de conhecimentos.

Portanto, as principais contribuições deste trabalho são: o aumento do poder de representação da metodologia MFG/PFS através das regras e a obtenção de uma especificação e implementação de uma ferramenta poderosa para simulação e análise de sistemas de manufatura.

Como sugestão para futuros trabalhos propõe-se:

1. Utilizar o MFG estendido [SANTOS F^o (1993)], que permite a inclusão de atributos nas marcas aumentando de maneira significativa o poder de representação do MFG. Além disso, o MFG estendido, permite a utilização da abordagem hierárquica e estruturada, inclusive para os atributos nas marcas,

através do PFS. Esta extensão do MFG pode ser aplicado, por exemplo, em sistemas com a capacidade de produção de peças diferentes e que utilizam sub-sistemas de transporte com vários AGVs e Robôs.

2. Implementar um Sistema Especialista [CUNHA (1987), LUCENA (1987)] que, a partir de conhecimentos relativos aos sistemas produtivos, selecione automaticamente as regras que definem as condições necessárias para o controle destes sistemas. Neste caso, verifica-se que através do conhecimento de um especialista (usuário) do sistema em estudo, é possível resolver os "deadlocks". Ou seja, é possível transformar estes conhecimentos do especialista numa base de conhecimento (aprendizado) para um sistema especialista. Portanto, a ferramenta poderia à partir destes conhecimentos, solucionar automaticamente os estados de "deadlocks" que surgem no sistema em estudo. Neste caso, quando ocorrer algum "deadlock" na simulação, o sistema automaticamente verificaria as causas baseando-se nos fatos e conhecimentos sobre o sistema e em seguida iria alterar, de maneira coerente, o conjunto de regras. Assim sendo, o especialista que atua sobre as regras para solucionar situações indesejáveis, poderia ser substituído, pelo menos parcialmente, por uma ferramenta inteligente capaz de solucionar essas situações de maneira semelhante.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALLA, H. ; LADET, P. Colored Petri Nets. A tool for Modeling Validation and Simulation of FMS. Flexible Manufacturing Systems: Methods and Studies A. Kusiak (Editor) Elsevier Science Publishers B. V. (North-Holland), 1986, p. 271-281.**
- ARARIBÓIA, G. Inteligência Artificial - Um curso prático. LTC - Livros Técnicos e Científicos Editora LTDA., 1989.**
- BANASZAK, Z. A.; KROGH, B. H. Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows. IEEE Transactions on Robotics and Automation, Vol. 6, Nº. 6, December 1990, p. 724-734.**
- BARATA, M. M; et al CIM System with Improved Capabilities fo Furniture Industry. UNINOVA First Demonstrator report. Universidade Nova de Lisboa, May, 1992.**
- BEN-ARIEH, D. Knowledge Based Control System for Automated Production and Assembly, Modelling and Design of Flexible Manufacturing Systems. A. Kusiak (Editor), Elsevier Science Publishers B. V. Amsterdam, 1986, p. 347-368.**
- BOURNE, D. A.; FOX, M. S. Autonomous Manufacturing: Automating the Job-Shop. IEEE Computer, September 1984, p. 76-86.**
- BRUNO, G., ELIA, A.; LAFACE, P. A Rule-Based System to Schedule Production. IEEE Computer, July 1986, p. 32-39.**

- CHONG-YI, Y. **Synchronic Distances in C/E Systems**. Lecture Notes in Computer Science "Advances in Petri Nets". Springer Verlag, 1985, p. 101-121.
- CUNHA, H.; RIBEIRO, S. **Introdução aos Sistemas Especialistas**. LTC - Livros Técnicos e Científicos Editora S.A., 1987.
- FREEDMAN, P. **Time, Petri Nets, and Robotics**. IEEE Transactions on Robotics and Automation, Vol. 7, Nº. 4, August 1991, p. 417-433.
- GOLTZ, U.; CHONG-YI, Y. **Synchronic Structure**. Lecture Notes in Computer Science "Advances in Petri Nets". Springer Verlag, 1985, p. 233-252.
- HASEGAWA, K.; TAKAHASHI, K. **Proposal of Mark Flow Graph for Discrete System Control**. Transactions of SICE, Tokyo, v. 20, no. 2, 1984, p. 122-129.
- HASEGAWA, K.; TAKAHASHI, K.; MASUDA, R.; TANGE, K.; SUZUKI, T. **Simulation of Discrete Production Systems based on Mark Flow Graph**. System Science, Vol. 13, Nº. 1-2, Poland, 1987.
- HO, Y. **Dynamics of Discrete Event Systems**. Proceedings of the IEEE, vol. 77, Nº. 1, January 1989, p. 3-6
- HOOVER, S. V.; PERRY, R. F. **SIMULATION - A Problem-Solving Approach**. Addison-Wesley, 1989.
- ITO, Y. **A Desirable Production structure looking toward the 21s century - Anthropocentric Intelligence Based Manufacturing**. XI Congresso Brasileiro de Engenharia Mecânica. Anais. São Paulo, Dezembro, 1991, p. 19-32.
- JENG, M. D.; DiCesare F. **A Review of Synthesis Techniques for Petri Nets with Applications to Automated Manufacturing Systems**. IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 1, January/February 1993, p. 301-313.
- JENSEN, K. **Colored Petri Nets**. Lecture notes in Computer Science, Advances in Petri Nets, Part I, p. 248-299.
- LINN, R. J.; WYSK, R. A. **An expert system based controller for an automated storage/retrieval system**. Int. J. Prod. Res., Vol. 28, Nº. 4, 1990, p. 735-756.

- LUCENA, J. P. C. **Inteligência Artificial e Engenharia de Software**. Publicações Acadêmico / Científicas, Puc/RJ, IBM Brasil, 1987.
- MARTINEZ, J.; MURO, P. R.; SILVA, M.; SMITH, S. F.; VILLARROEL, J. L. **Merging Artificial Intelligence Techniques and Petri Nets for Real Time Scheduling and Control os Production Systems**, 12 th IMACS world congress on scientific on computation. **Anais**. Paris-França, July 1988, p. 528-531.
- MASUDA, R.; HASEGAWA, K. **Mark Flow Graph an its application to complex sequential control system**. the 13th Hawaii International Conference on System Science. **Proceedings**. 1980, p. 194-203.
- MIYAGI, P. E. **Automação de Sistemas Seqüenciais**. Apostila do curso de simulação e modelagem de sistemas seqüenciais da Escola Politécnica da USP (mecatrônica), 1993.
- MIYAGI, P. E. **Control system design, programming and implementation for discrete event production systems by using Mark Flow Graph**. Janeiro de 1988, Tokyo, Japan. Tese (doutorado) - Tokyo Institute of Technology.
- MIYAGI, P. E.; HASEGAWA K.; TAKAHASHI, K. **Mark Flow Graph (MFG) para modelamento e controle de Sistemas de Eventos Discretos**. Monografias em Automação e Inteligência Artificial, Depto. Eng. Mecânica da Escola Politécnica da USP, Vol. 1, Nº 1, 1989.
- MURATA, T. **Petri Nets: Properties, Analysis and Applications**. **Proceedings of IEEE**, Vol. 77, Nº. 4, April, 1989, p. 541-580.
- PETERSON, J. L. **Petri Net Theory and Modeling of Systems**. Prentice Hall, 1991.
- REISIG, W. **A primer in Petri Net Design**. Spring Verlag, 1992.
- REISIG, W. **Petri Nets. An Introduction**. Spring Verlag, 1982.
- RICH, E. **Artificial Inteligence**. International Student Edition, McGraw-Hill, 3ª. edição, 1985.
- SAHRAOUI, A.; ATABAKHCHE, H; COURVOISIER, M.; VALETTE, R. **Joining Petri Nets and Knowledge based Systems for Monitoring Purposes**. **IEEE**

- International Conference on Robotics and Automation. Raleigh, North Carolina, USA, Março 1987.**
- SANTOS F^o, D. J. Proposta do Mark Flow Graph estendido para a modelagem e controle de Sistemas Integrados de Manufatura. Tese de mestrado. Escola Politécnica da USP, São Paulo, 1993.**
- SANTOS F^o, D. J.; MIYAGI, P. E. Sistemas de Eventos Discretos e seu controle. 1^o. Congresso Brasileiro de Instrumentação e Sistemas de Automação e 1^o. Congresso Internacional de ISA. Anais. 1991, p. 2.1-2.12.**
- SANTOS F^o, D. J.; ARAKAKI, J.; MIYAGI, P. E. Controle de tarefas de um sistema de transporte por veículos autônomos - Aplicação da técnica MFG/PFS em Sistemas Integrados de Manufatura. 9^o. Congresso Brasileiro de Automação, CBA. Anais. UFES - Vitória/ES - Brasil, 1992, p. 958-963.**
- SOARES, L. F. G. Modelagem e Simulação discreta de sistemas. VII Escola de Computação, IME-USP, 1990.**
- SPANGLER, W. E. The role of Artificial Intelligence in understanding the Strategic Decision-Making Process. IEEE Transactions on Knowledge and Data Engineering, vol. 3, No. 2, June 1991, p. 149-159.**
- SUBRAMANYAM, S; ASKIN, R. G. An Expert Systems approach to scheduling in Manufacturing Systems. A. Kusiak (editor), Elsevier Science Publishers B.V. Amsterdam, 1986, p. 243-257.**
- SZWARCFITER, J. L. Grafos e Algoritmos Computacionais. 2^a edição, Editora Campus Ltda., 1986.**
- VALAVANIS, K. P. On the Hierarchical Modeling Analysis and Simulation of Flexible Manufacturing Systems with Extended Petri Nets. IEEE Transactions on Systems on Systems, Man and Cybernetics, Vol. 20, N^o. 1, January/February 1990, p. 94-110.**
- VALETTE, R.; SILVA, M. A rede de Petri.: uma Ferramenta para a Automação Fabril. 4^o. Congresso Nacional de Automação Industrial (CONAI). Anais. Julho 1990.**

- VALETTE, R. Nets in Production Systems. Advanced Course on PETRI NETS, September 1986, p. 1-23.**
- VISWANADHAM, N.; NARAHARI, Y.; JOHNSON, T. L. Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing System Using Petri Net Models. IEEE Transactions on Robotics and Automation, Vol. 6, Nº. 6, December 1990, p. 713-723.**
- YOSHIDA, S. Tagged MFG based Control of Material Flow in Job-Shop Production System. Dissertação de mestrado, Tokyo Institute of Technology, Japan, 1988.**
- ZHOU, M.; DICESARE, F.; DESROCHERS, A. A. A Top-down Approach to Systematic Synthesis of Petri Net Models for Manufacturing Systems. Proceedings, IEEE International Conference on Robotics and Automation, 1989, p. 534-539.**

LEITURAS RECOMENDADAS

- AL-JAAR, R. Y.; DESROCHERS, A. A. A Survey of Petri Nets in Automated Manufacturing System. 12 th IMACS world congress on scientific on computation. Anais. Paris-França, July 1988, p. 503-510.
- ARAKAKI, J.; SANTOS F^o, D. J.; MIYAGI, P. E. Análise de Sistemas Integrados de Manufatura baseada na metodologia MFG/PFS. Robótica e Automatização. Workshop Iberoamericano de Sistemas Autônomos em Robótica e CIM, Lisboa-Portugal, 2-4 Nov, 1992, p. 19-24.
- BOUREY, J.P.; CASTELAIN, E.; GENTINA, J.C.; KAPUSTA, M. C.A.S.P.A.I.M.: A Computer Aided Design of the Control System of FMS. 12 th IMACS world congress on scientific on computation. Anais. Paris-França, July 1988, p. 517-521.
- BROWNE, J. (editor) Knowledge based production management systems. North-Holland, 1989.
- GORDON, G. System Simulation. second edition, Prentice Hall, 1978.
- GREVET, J. M.; JANDURA, L.; BRODE, J.; LEVIS, A. H. Execution Strategies for Petri Net Simulations. 12 th IMACS world congress on scientific on computation. Anais. Paris-França, July 1988, p. 511-516.
- HARTMAN, J. G. Predicate/Transition Nets. Lecture Notes in Computer Science "Petri Nets: Central Models and their Properties", Springer Verlag.

- KOMODA, N.; KERA, K.; KUBO, T. An Autonomous, Decentralized Control System for Factory Automation. *IEEE Computer*, December 1984, p. 73-83.
- MARTINEZ, J.; ALLA, H.; SILVA, M. Petri Nets for the specification of FMSs. *Modelling and design of Flexible Manufacturing Systems*, Andrew Kusiak (ed.), Elsevier Science Publishers B. V., Amsterdam, 1986, p. 389-407.
- MIYAGI, P. E.; SANTOS Fº, D. J.; ARAKAKI, J. Specification of Task Control for Production Systems. *The IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems '92 Kobe. Proceedings*, Japan, Set. 16-20, 1992, p. 1505-1510.
- 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 Transactions on Industrial Electronics*, Vol. IE-33, Nº.1, February 1986, p. 1-8.
- NORONHA, S.J.; SARMA, V.V.S. Knowledge-Based Approaches for Scheduling Problems: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 3, Nº. 2, June 1991, p. 160-171.
- SHAW, M. J. P.; WINSTON, A. B. *Applications of Artificial Intelligence to planning and scheduling in Flexible Manufacturing*. A. Kusiak (editor), Elsevier Science Publishers B.V. Amsterdam, 1986, p. 223-241.
- TENG, S. H.; BLACK, J. T. Cellular Manufacturing Systems Modeling: The Petri Net Approach. *Journal of Manufacturing Systems*, Vol. 9, Nº. 1.
- VALETTE, R.; CARDOSO, J.; ATABAKHCHE, H.; COURVOISIER, M.; LEMAIRE, T. Petri nets and Productions rules for Decision levels in FMS control. 12 th IMACS world congress on scientific on computation, July 1988. *Anais. Paris-França*, p. 522-524.
- WILLSON, R. G.; KROGH, B. H. Petri Net tools for specification and analysis os Discrete Controllers. *IEEE Transactions on Software Engineering*, vol. 16, Nº. 1, January, 1990, p. 39-50.

Apêndice

REDES DE PETRI

A.1 INTRODUÇÃO

O objetivo deste apêndice é a complementação das características e propriedades das redes de Petri apresentadas no capítulo 2.

Historicamente, o conceito de redes de Petri foi introduzido pela primeira vez, por Carl Adam Petri, em sua tese de doutorado, submetido em 1962 na faculdade de Matemática e Física da Universidade de Darmstadt, Alemanha. Nesta época, Petri utilizou estas redes para a aplicação na Teoria de Sistemas de Informação.

Após a proposta inicial apresentado por Petri, várias pesquisas foram (estão sendo) realizadas, onde propostas de extensões da rede de Petri original são apresentadas, de tal forma a melhorar o poder de modelagem e análise dos sistemas baseados em eventos discretos (SED).

A natureza gráfica e a facilidade de representação de modo eficiente e consistente das características como assincronismo, paralelismo e conflito, favorecem a aplicação destas redes em diversos sistemas, cujo comportamento dinâmico são baseados em SED.

Dentre as principais áreas em que as redes de Petri estão sendo aplicadas, pode-se destacar: protocolos de comunicação, sistema de "software" distribuído, sistema de banco de dados distribuído, modelagem e análise de sistemas integrados de manufatura, redes neuronais, sistemas de memória multiprocessada, redes de computadores, automação e controle de sistemas de produção.

Para a utilização desta ferramenta de maneira consistente, é necessário que exista uma formalização matemática que comprovem as suas propriedades. Desta maneira, descreve-se a seguir alguns conceitos básicos que foram utilizados para a formalização destas redes.

A.2 DESCRIÇÃO FORMAL

Neste item apresenta-se a descrição formal básica das redes de Petri e as regras necessária para a representação de sua dinâmica.

A rede de Petri corresponde a um grafo bipartido e direcionado que pode se representado por uma quintupla do tipo: $PN = (P, T, E, S, M)$ onde,

- . $P = \{p_1, p_2, \dots, p_m\}$ ($m \geq 1$) é um conjunto finito de lugares com cardinalidade m ,
- . $T = \{t_1, t_2, \dots, t_n\}$ ($n \geq 1$) é um conjunto finito de transições com cardinalidade n ,
- . E é a função $(m, n) \rightarrow \{0, 1\}$ que define o conjunto de lugares de entrada da transição,

. S é a função $(m,n) \rightarrow \{0,1\}$ que define o conjunto de lugares de saída da transição,

. $M > 0$ é um vetor coluna $m \times 1$ que representa a marcação, onde $M(i)$ é a marca contida em $p(i)$.

Se uma rede de Petri não possuir "self-loop", que corresponde ao caminho fechado de um lugar para uma transição e voltando para o mesmo lugar (Figura A.1), então pode-se utilizar uma matriz de incidência Mi definido por $Mi = S - E$ caracteriza completamente a relação entre as condições e eventos. Sendo os pesos nos arcos igual a 1, isto é, o disparo das transições subtrai exatamente uma marca de cada lugar de entrada e adiciona exatamente uma marca em cada lugar de saída, desta forma, todos os elementos de Mi deve sempre ser um do conjunto $\{-1,0,1\}$ ¹.

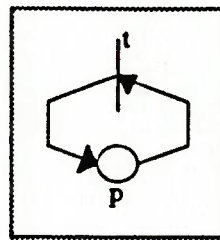


Fig. A.1 "Self-loop" em rede de Petri.

No exemplo da Figura A.2 a distribuição inicial de marcas indica a marcação inicial, que representa o estado inicial do sistema em estudo. Para esta rede de Petri, a marcação inicial $M_0 = [1 \ 1 \ 0 \ 0 \ 0]^T$ e a matriz de incidência do exemplo é:

¹ Numa rede do tipo lugar/transição, os elementos da matriz de incidência são do domínio dos inteiros, pois os arcos tem peso maior que 1.

$$M_i = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

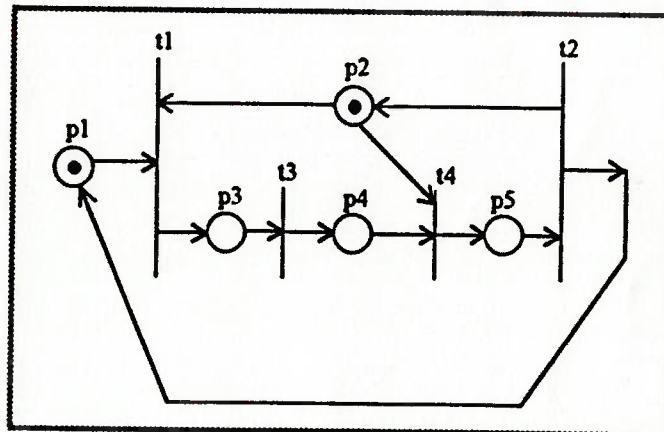


Fig. A.2 Exemplo de uma rede de Petri.

A.3 PROPRIEDADES DAS REDES DE PETRI

As redes de Petri possuem algumas propriedades que são úteis para a análise estrutural e comportamental do sistema em estudo. As principais propriedades destas redes são:

- **consistência:** existe uma marcação inicial M_0 e uma seqüência de disparos σ de tal forma que partindo-se de M_0 retorna-se novamente a M_0 onde todas as transições ocorrem pelo menos uma vez em σ [MURATA (1989)].

- **conservatividade:** para qualquer marcação M da rede, a soma das marcas é constante.
- **limitabilidade:** corresponde à verificação se uma dada rede é limitada ou não, para o número de marcas, à partir de uma dada marcação inicial. Genericamente, se o número de marcas em todos os lugares é sempre limitado por algum valor finito k , então a rede é chamada de k -limitado.
- **segurança:** esta propriedade é uma especificação da propriedade anterior. Neste caso, uma rede é dita segura quando ela é 1-limitado, isto é, o número de marcas contidas em qualquer lugar é sempre 0 ou 1. Permite a verificação da existência do fenômeno de "colisão".
- **vivacidade:** indica se uma rede é viva ou não, ou seja, se para uma dada marcação inicial, existe para cada transição, uma seqüência legal de disparos que leva a uma marcação no qual aquela transição seja habilitada. Esta propriedade é importante para a verificação da existência do estado de "deadlock" no sistema.
- **reversibilidade:** é uma propriedade onde, invertendo-se o sentido dos arcos, colocando marcas nos lugares que não os contém e retirando as marcas dos lugares que tenham, a condição de disparo não muda. A figura A.3 mostra que a condição de habilitado da transição não muda fazendo-se a reversão.

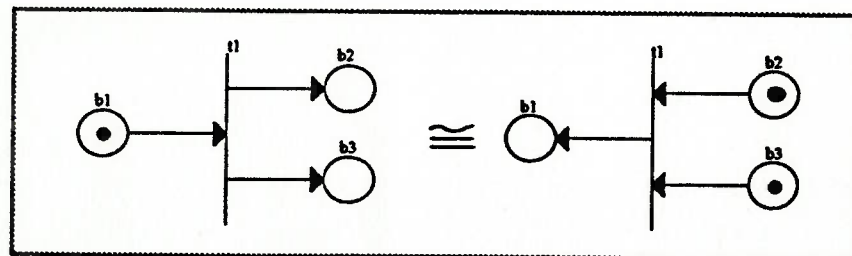


Fig. A.3 Propriedade de reversibilidade

A.4 REDES CONDIÇÃO-EVENTO (C/E)

Uma rede de Petri é do tipo Condição-Evento (C/E) quando, os eventos são associados às transições e as condições aos lugares. Os lugares e as transições são interligadas através dos arcos. A presença de uma marca num determinado lugar, indica que a condição correspondente é verdadeira. A marcação dos lugares indicam a manutenção do estado. Numa rede C/E essa marcação é dada em termos das condições Falsa ou Verdadeira. Um exemplo de um sistema produtor-consumidor [REISIG, (1992)] é apresentado pela Figura A.5 que mostra um sistema modelado através de uma rede C/E. Neste caso, verifica-se que os eventos ocorrem se as pré-condições são satisfeitas e as pós-condições não são satisfeitas. Na Figura A.5, o evento envio está habilitado se a condição produtor pronto para envio está satisfeita e as condições canal ocupado e produtor pronto para produzir não estão satisfeitas.

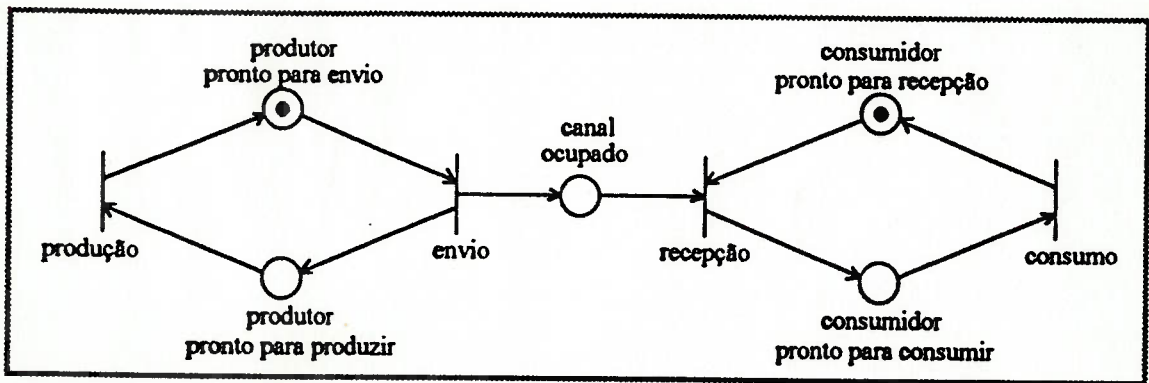


Fig. A.4 Sistema Condição/Evento.

A.4.1 Regras para a rede C/E

Para que uma transição dispare (um evento ocorra), numa rede C/E, é necessário que ela esteja habilitada e, neste caso, sejam satisfeitas as seguintes regras:

- todas as pré-condições devem estar satisfeitas,
- todas as pós-condições não devem estar satisfeitas.

Quando um evento habilitado ocorre, suas pré-condições deixam de ficar satisfeitas e as suas pós-condições ficam satisfeitas.

A distribuição das marcas nos lugares (marcação), representa o estado e a transição de estado é representado pela mudança da marcação na rede. Esta mudança na distribuição de marcas na rede, representa a dinâmica da rede. As marcas nos lugares controlam o disparo das transições e quando uma transição dispara, as marcas dos lugares de entrada são removidos e nos lugares de saída são colocadas marcas. O disparo de uma transição numa rede C/E só pode ser realizada se ela estiver habilitada.

De acordo com as regras citadas, uma transição está habilitada quando todos os lugares de entrada de uma determinada transição estão marcados e todos os lugares de saída desta transição estão sem marcas. Uma transição habilitada dispara instantaneamente. E o disparo desta transição significa a ocorrência do evento. A Figura A.5 mostra o estado do sistema após a ocorrência do evento envio que estava habilitado anteriormente.

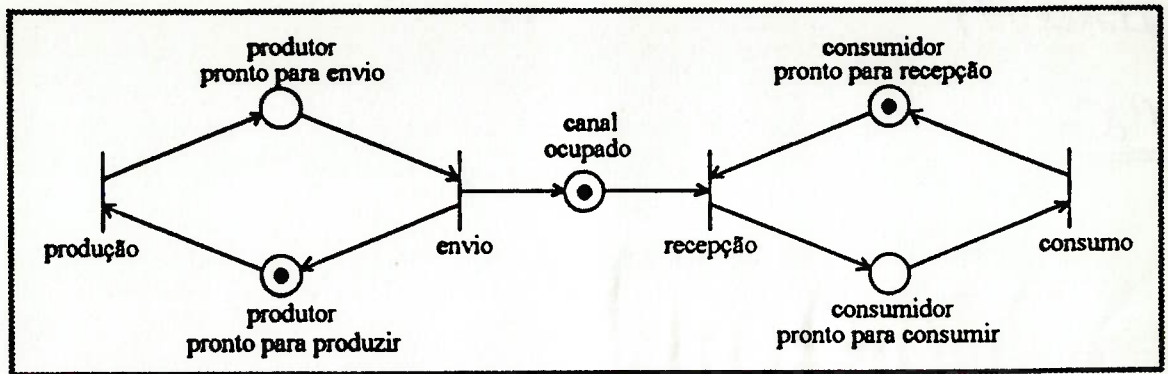


Fig. A.5 Marcação após a ocorrência do evento envio.