Universidade de São Paulo
Escola de Engenharia de São Carlos

Modelagem de Sistemas de Eventos Discretos utilizando Rede de Petri Virtual

Serviço de Pós-Graduação EESC/USP
EXEMPLAR REVISADO
Data de entrada no Serviço 31/08/04
Ass.:

Patrícia Ferraz
Cientista da Computação
Universidade Federal de Uberlândia

DEDALUS - Acervo - EESC
31100047325

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Mecânica.

ORIENTADOR: Prof. Dr. Ricardo Yassushi Inamasu

EESC-USP
Biblioteca
Campus de S. Carlos

São Carlos
2004

Class. TESE EESC
Cutt. 2336
Tombo T225/04
Sysno 1395340

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

F381m

Ferraz, Patrícia
Modelagem de sistemas de eventos discretos utilizando rede de Petri virtual /Patrícia Ferraz . -- São Carlos, 2004.

Dissertação (Mestrado) -- Escola de Engenharia de São Carlos-Universidade de São Paulo, 2004.
Área: Engenharia Mecânica.
Orientador : Prof. Dr. Ricardo Yassushi Inamasu.

1. Rede de Petri. 2. Rede de Petri virtual. 3. Sistemas de eventos discretos. 4. Modelagem. I. Título.

FOLHA DE JULGAMENTO

Candidata: **PATRÍCIA FERRAZ**

Dissertação defendida e julgada em 07-04-2004 perante a Comissão Julgadora:

Aprovada

Prof. Dr. **RICARDO YASSUSHI INAMASU (Orientador)**
(Escola de Engenharia de São Carlos/USP)

APROVADA

Prof. Assoc. **ARTHUR JOSÉ VIEIRA PORTO**
(Escola de Engenharia de São Carlos/USP)

Aprovada

Prof. Dr. **FRANCISCO JOSÉ DE ALMEIDA**
(Universidade Metodista de Piracicaba/UNIMEP)

Prof. Associado **JONAS DE CARVALHO**
Coordenador do Programa de Pós-Graduação
em Engenharia Mecânica

Profa. Tit. **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação

Dedicatória

Aos meus pais, Carlos e Elza,
meus mestres desde o meu nascimento.
Ao meu irmão Júnior, por seu carinho.
Ao Giovanni, pelo amor, carinho e apoio.
Ao Padrinho Sebastião e à tia Elzinha,
que sempre apoiaram meus estudos.
À minha amada Avó Maria (in memorium).

Agradecimentos

Ao meu orientador, Professor Ricardo Yassushi Inamasu, pela oportunidade de desenvolver esse trabalho, incentivo, amizade, apoio e por sua paciência em me atender fora do seu horário de trabalho.

Aos Professores Arthur José Vieira Porto e Eduardo Vila, pelo incentivo, amizade e colaboração.

Aos colegas do Laboratório de Simulação Andrea, Sayuri, Marcelo, José Luiz, Ana Cristina, Sílvia, João, Sérgio, Rafael, Mamoru, Mariella, Simone, Elena, Jandira, Victor, Jader e Douglas pela amizade e conversas no "cafezinho".

Às secretárias Beth e Ana Paula, pelo, apoio, incentivo e constante vontade em ser útil.

Um agradecimento especial à Andrea e Beth pela amizade sincera e auxílio nas etapas finais do trabalho.

À todos os funcionários da EESC-USP, pela boa vontade no atendimento, pelos recursos disponibilizados e por sempre fazerem além do que simplesmente cumprirem suas obrigações.

À CAPES – Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior - pelo apoio financeiro concedido.

À toda minha família pelo apoio incondicional e compreensão pelos momentos de ausência.

À Deus, sem sua vontade a conclusão desse trabalho não seria possível.

Sumário

LISTA DE FIGURAS ..... **VI**

LISTA DE TABELAS ..... **VIII**

LISTA DE SIGLAS E SÍMBOLOS ..... **IX**

RESUMO ..... **XI**

ABSTRACT ..... **XII**

1. INTRODUÇÃO **1**

1.1. ESCOPO DO TRABALHO 1
1.2. MOTIVAÇÕES 4
1.3. ORGANIZAÇÃO DO TRABALHO 5

2. REVISÃO BIBLIOGRÁFICA **6**

2.1. CONSIDERAÇÕES INICIAIS 6
2.2. REDES DE PETRI 6
2.2.1. Elementos Básicos *6*
2.2.2. Marcação *7*
2.2.3. Definição Formal de Rede de Petri Lugar/Transição *7*
2.2.4. Habilitação e Disparo de Transição *8*
2.2.5. Propriedades Comportamentais *9*
2.2.5.1. Alcançabilidade (*Reachability*) 9
2.2.5.2. Limitabilidade (*Boundedness*) 11
2.2.5.3. Vivacidade (*Liveness*) 11
2.2.5.4. Reversibilidade (*Reversibility*) 12
2.2.5.5. Abrangência de Cobertura (*Coverability*) 13
2.2.5.6. Persistência (*Persistence*) 13
2.2.5.7. Distância simultânea (*Synchronic Distance*) 14
2.2.6. Propriedades Estruturais *15*
2.2.7. Métodos de Análise *16*
2.2.7.1. Árvore de Cobertura 16
2.2.7.2. Matriz de Incidência 17
2.2.7.3. Métodos de Redução 18
2.2.8. Algumas extensões *20*
2.3. SISTEMAS DE EVENTOS DISCRETOS 22
2.4. REDE DE PETRI E OS SISTEMAS DE EVENTOS DISCRETOS 24

3. REDE DE PETRI VIRTUAL **28**

3.1. CONSIDERAÇÕES INICIAIS 28
3.2. ALGORITMO PARA UTILIZAÇÃO DE PN VIRTUAL 29
3.3. ELEMENTOS ESTRUTURAIS 30
3.4. DEFINIÇÃO FORMAL 31
3.5. ASSOCIAÇÃO DOS MÓDULOS 33

3.5.1. Procedimento de junção *34*
3.5.2. Exemplos de associações *34*
3.6. EXEMPLO DE APLICAÇÃO 37
3.6.1. Modelagem utilizando Rede de Petri Virtual *38*

4. ESTUDO DE CASO 42

4.1. CONSIDERAÇÕES INICIAIS 42
4.2. DESCRIÇÃO DO SISTEMA 42
4.2.1. Processo de Fabricação *44*
4.3. EXECUÇÃO DO ALGORITMO PARA UTILIZAÇÃO DE PN VIRTUAL 44
4.3.1. Fase de Caracterização *44*
4.3.2. Fase de Modelagem *45*
4.3.2.1. Módulo das Estações 45
4.3.2.2. Módulo dos AGV's 46
4.3.2.3. Módulo de Processo de Fabricação 46
4.3.2.4. Módulo de Rotas 47
4.3.3. Fase de Associação *49*
4.3.3.1. Junção dos Módulos Estação e AGV 49
4.3.3.2. Junção dos Módulos de Rotas e Estação-AGV 51
4.3.3.3. Modelo Final 53

5. COMENTÁRIOS FINAIS E CONCLUSÕES 55

5.1. TRABALHOS FUTUROS 57

6. REFERÊNCIAS BIBLIOGRÁFICAS 58

Lista de Figuras

FIGURA 2.1. ILUSTRAÇÃO DA REGRA DE TRANSIÇÃO (MURATA, 1989) 9
FIGURA 2.2. REDE DE PETRI (CARDOSO & VALETTE, 1997) 10
FIGURA 2.3. GRAFO DE MARCAÇÕES ACESSÍVEIS (CARDOSO & VALETTE, 1997) 10
FIGURA 2.4. REDE DE PETRI NÃO LIMITADA (CARDOSO & VALETTE, 1997) 11
FIGURA 2.5. REDE DE PETRI NÃO VIVA (MURATA, 1989) 12
FIGURA 2.6. REDE DE PETRI PERSISTENTE (MURATA, 1989) 14
FIGURA 2.7. REDE DE PETRI (MURATA, 1989) 14
FIGURA 2.8. ÁRVORE DE COBERTURA 16
FIGURA 2.9. MÉTODOS DE REDUÇÃO (MURATA, 1989) 19
FIGURA 2.10. REDE DE PETRI REDUZIDA (MURATA, 1989) 20
FIGURA 2.11. MODELO DE FILAS DE UM SED 23
FIGURA 2.12. GRAFO ORIENTADO 23
FIGURA 3.1. NOTAÇÃO DOS NÓS EM UMA REDE DE PETRI VIRTUAL 31
FIGURA 3.2. PRÉ-CONJUNTOS E PÓS-CONJUNTOS DE TRANSIÇÕES VIRTUAIS 33
FIGURA 3.3. PRÉ-CONJUNTOS E PÓS-CONJUNTOS DE LUGARES VIRTUAIS 33
FIGURA 3.4. EXEMPLO 1 DE ASSOCIAÇÃO DE MÓDULOS 35
FIGURA 3.5. EXEMPLO 2 DE ASSOCIAÇÃO DE MÓDULOS 35
FIGURA 3.6. EXEMPLO 3 DE ASSOCIAÇÃO DE MÓDULOS 36
FIGURA 3.7. EXEMPLO 4 DE ASSOCIAÇÃO DE MÓDULOS 36
FIGURA 3.8. SISTEMA DE MANUFATURA FLEXÍVEL (JENG, 1995) 38
FIGURA 3.9. MÓDULO M_A 39
FIGURA 3.10. MÓDULO B 39
FIGURA 3.11. MÓDULO R 39
FIGURA 3.12. MÓDULO M_B 40
FIGURA 3.13. MÓDULO M_C 40
FIGURA 3.14. REDE DE PETRI FINAL 41
FIGURA. 4.1. ESQUEMA DO SISTEMA AUTOMATIZADO DE MANUFATURA 43
FIGURA 4.2. EXEMPLO DE MOVIMENTO DE PEÇA ENTRE ESTAÇÃO E AGV 43
FIGURA 4.3. MÓDULO E 45
FIGURA 4.4. MÓDULO A 46
FIGURA 4.5. MÓDULO P 47
FIGURA 4.6. MÓDULO R 48
FIGURA 4.7. MÓDULOS ESCOLHIDOS NO PASSO 3.1. DO ALGORITMO 49
FIGURA 4.8. RESULTADO PARCIAL DA JUNÇÃO DO MÓDULO A E MÓDULO E 50

FIGURA 4.9. MÓDULO EA 51
FIGURA 4.10. MÓDULO EAR 52
FIGURA 4.11. MODELO FINAL 54

Lista de Tabelas

TABELA 1. DEFINIÇÃO FORMAL DE REDE DE PETRI (MURATA, 1989) 7
TABELA 2. INTERPRETAÇÃO DOS LUGARES PARA M_A 39
TABELA 3. INTERPRETAÇÃO DOS LUGARES PARA B 39
TABELA 4. INTERPRETAÇÃO DOS LUGARES PARA R 39
TABELA 5. INTERPRETAÇÃO DOS LUGARES PARA M_B 40
TABELA 6. INTERPRETAÇÃO DOS LUGARES PARA M_C 40
TABELA 7. INTERPRETAÇÃO DOS ELEMENTOS DO MÓDULO E 45
TABELA 8. INTERPRETAÇÃO DOS ELEMENTOS DO MÓDULO A 46
TABELA 9. INTERPRETAÇÃO DOS ELEMENTOS DO MÓDULO P 47
TABELA 10. INTERPRETAÇÃO DOS ELEMENTOS DO MÓDULO R 48

Lista de Siglas e Símbolos

AGV	Veículo auto guiado (*Auto Guide Vehicle*)
CNC	Comando numérico computadorizado
F	Conjunto de arcos de uma Rede de Petri
$L(M_0)$	Conjunto de todas as seqüências de disparo possíveis
M	Marcação de uma Rede de Petri
M(p)	Número de marcas do lugar p de uma Rede de Petri
M_0	Marcação inicial de uma Rede de Petri
MFG	*Mark Flow Graph*
MPN	Rede de Petri Modular (*Modular Petri Net*)
OOPN	Rede de Petri Orientada a Objeto (*Object Oriented Petri Net*)
P	Conjunto de lugares de uma Rede de Petri
PN	Rede de Petri (*Petri net*)
POO	Programação Orientada a Objetos
$R(M_0)$	Conjunto de todas as possíveis marcações alcançáveis de M_0
SADT	*Strutured Analysis and Design Technique*
SAM	Sistema Automatizado de Manufatura
SED	Sistema de Evento Discreto
SFC	*Seqüencial Function Chart*
SVC	Sistema de Variável Contínua
T	Conjunto de transições de uma Rede de Petri
vF	Conjunto de arcos virtuais de uma Rede de Petri Virtual
VF	Conjunto de arcos virtuais e reais de uma Rede de Petri Virtual
VM_0	Marcação inicial de uma Rede de Petri Virtual
vP	Conjunto de lugares virtuais de uma Rede de Petri Virtual
VP	Conjunto de lugares virtuais e reais de uma Rede de Petri Virtual
VPN	Rede de Petri Virtual (*Virtual Petri Net*)
vT	Conjunto de transições virtuais de uma Rede de Petri Virtual
VT	Conjunto de transições virtuais e reais de uma Rede de Petri Virtual
VW	Função peso dos arcos virtuais e reais de uma Rede de Petri Virtual
W	Função peso dos arcos de uma Rede de Petri

σ	Seqüência de disparo.
$\bullet t_k$	Conjunto de lugares de entrada da transição t_k
$v(\bullet t_k)$	Conjunto de lugares virtuais de entrada da transição t_k
$v(t_k\bullet)$	Conjunto de lugares virtuais de saída da transição t_k
$t_k\bullet$	Conjunto de lugares de saída da transição t_k
$\bullet vt_i$	Conjunto de lugares de entrada da transição virtual vt_i
$v(\bullet vt_i)$	Conjunto de lugares virtuais de entrada da transição virtual vt_i
$v(vt_i\bullet)$	Conjunto de lugares virtuais de saída da transição virtual vt_i
$vt_i\bullet$	Conjunto de lugares de saída da transição virtual vt_i
$\bullet t_k$	Conjunto de lugares de entrada da transição t_k
$v(\bullet t_k)$	Conjunto de lugares virtuais de entrada da transição t_k
$v(t_k\bullet)$	Conjunto de lugares virtuais de saída da transição t_k
$t_k\bullet$	Conjunto de lugares de saída da transição t_k
$\bullet vt_i$	Conjunto de lugares de entrada da transição virtual vt_i
$v(\bullet vt_i)$	Conjunto de lugares virtuais de entrada da transição virtual vt_i
$v(vt_i\bullet)$	Conjunto de lugares virtuais de saída da transição virtual vt_i
$vt_i\bullet$	Conjunto de lugares de saída da transição virtual vt_i

Resumo

FERRAZ, P. (2004). *Modelagem de Sistemas de Eventos Discretos Utilizando Rede de Petri Virtual*. São Carlos, 2004. 62p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

Rede de Petri é uma poderosa ferramenta de modelagem gráfica e matemática bastante aplicada no desenvolvimento de projetos de Sistemas de Eventos Discretos. Porém a sua aplicação na análise e interpretação de tais sistemas torna-se inviável por resultar em modelos grandes, com muitos elementos gráficos. Para solucionar tal problema, vários pesquisadores têm concentrado esforços no desenvolvimento de novas extensões e métodos de síntese de Rede de Petri, para reduzir o tamanho dos modelos e assim facilitar a sua aplicação e análise de sistemas grandes e complexos. Rede de Petri Virtual é uma nova extensão de Rede de Petri que possibilita a modelagem de tais sistemas de forma modular. Cada elemento do sistema é representado por um módulo e a comunicação entre eles é feita através dos nós virtuais. Esse trabalho formaliza a definição de Rede de Petri Virtual, desenvolve algoritmo e procedimento de junção dos módulos para gerar o modelo final, uma Rede de Petri ordinária que representa o sistema completo.

Palavras-chave: Rede de Petri, Rede de Petri Virtual, Sistemas de Eventos Discretos, Modelagem.

Abstract

FERRAZ, P. (2004). *Modeling of Discrete Events Systems using Virtual Petri Net.* São Carlos, 2004. 62p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

Petri Net is a powerful graphical and mathematical modeling tool commonly used to project Discrete Events Systems. The increasing complexity of such systems does not allow the use of Petri Net Tools due to the large size of the models (many graphical elements), which is difficult to understand and analyze. Due to this fact, researchers have been made efforts to the development of new synthesis methods for Petri Nets, in order to reduce the models size and become easier its use and the analysis task of the systems properties. Virtual Petri Net is a new kind (extension) of Petri Net, combining its best in control and representation of discrete events systems to the best of modular modeling. This kind of Petri Net allows to build up models of complex systems from the modules that represent its elements, linked by the virtual nodes. The present work formalizes the definition of Virtual Petri Net. Also develops an algorithm to its use in the discrete events systems and presents a procedure to link and assemble the modules in the whole model. The modular modeling makes easier the understanding and graphical visualization of the system, keeping the final model the same features of the common Petri Nets.

Keywords: Petri Net, Virtual Petri Net, Discrete Events Systems, Modeling.

1. Introdução

1.1. Escopo do Trabalho

O mercado consumidor tem procurado cada vez mais produtos com alta qualidade e baixo custo. Esta é uma tendência mundial que tem provocado mudanças radicais nas estruturas organizacionais e produtivas das empresas. Tais mudanças têm o objetivo de capacitar as empresas a competirem em um mercado cada vez mais exigente e com crescente interesse em novos produtos.

Para que as empresas possam sobreviver nesse mercado competitivo e globalizado, tornou-se vital a introdução de novas tecnologias organizacionais e produtivas, onde a eficiência, a produtividade e a segurança são os principais aspectos que devem ser considerados ao projetar ou modificar um sistema. Por esse motivo, é necessário fazer planejamento e estudo da viabilidade antes de qualquer mudança nos processos da empresa. Ferramentas de modelagem, principalmente as gráficas e de simulação, auxiliam na tomada de decisão (ROBERTS & DESSOURKY, 1998; CUBERT & FISHWICK, 1998 E PENG & CHEN, 1996).

Considerando esses fatores nos quais o objetivo é o projeto de Sistemas de Eventos Discretos (SED), ferramentas gráficas tais como diagramas, esboços e tabelas, constituem a maneira básica de representar tais sistemas. A aplicação dessas ferramentas requer o uso de apropriadas metodologias de projeto e ferramentas de suporte para a construção desses modelos, incluindo *softwares* (SANTAREK, 1998).

No passado, SED's eram bastante simples e as soluções intuitivas ou *ad hoc* eram suficientes para solucionar a maior parte dos problemas. A crescente complexidade

dos sistemas feitos pelo homem possibilitaram, através da aplicação generalizada da informática, que tais sistemas fossem elevados a um nível de complexidade tal que métodos formais mais detalhados tornaram-se necessários para seu projeto e análise (RAMADGE & WONHAM, 1989).

Rede de Petri (PN) é uma poderosa ferramenta de modelagem gráfica e matemática. Sua modelagem gráfica tem sido amplamente utilizada em simulação, análise e controle de sistemas de eventos discretos, como ferramenta visual. A modelagem matemática pode ser usada para montar equações de estado, equações algébricas, simulações e outros modelos matemáticos.

Entre as aplicações de Rede de Petri podem ser citadas: avaliação de desempenho, análise e verificação formal em sistemas discretos, protocolos de comunicação, controle de oficinas de fabricação, concepção de *software* de tempo real e/ou distribuído, sistemas de informação (organização de empresas), sistemas de transporte, logística, gerenciamento de base de dados, interface homem-máquina e multimídia (CARDOSO & VALETTE, 1997).

Embora a representação gráfica seja uma vantagem da Rede de Petri, a característica mais importante do modelo é o fato de ser formal. Sua vantagem em relação a outros modelos, como o SADT, que também oferece uma boa ferramenta gráfica de especificação, é a obtenção de mais informações sobre o comportamento do sistema modelado, através da análise de suas propriedades.

RIGHINI (1993) e RILLO (1987) apresentam as características de Rede de Petri que a torna adequada na modelagem de sistemas complexos:

i) Permite a representação de diferentes tipos de sistemas;
ii) É capaz de representar processos paralelos e eventos sincronizados facilmente;
iii) É formal (tem consistência matemática);
iv) É de fácil aprendizado, funcionando como linguagem de comunicação entre especialistas de diversas áreas;

v) É não determinística, nesse sentido a seqüência de disparos de transições não é especificada no modelo e pode ser determinada por políticas externas;
vi) A existência de poderosas extensões de PN permite a escolha da técnica de modelagem mais adequada, de acordo com as características do sistema a ser examinado;
vii) Representa aspectos estáticos e dinâmicos;
viii) É uma ferramenta gráfica, o que facilita o entendimento;
ix) Contém o conceito de estado parcial e
x) Possui métodos de análise que podem ser usados comercialmente.

As Redes de Petri são aplicadas com sucesso para modelar sistemas tais como: sistemas de transporte por AGVs (*auto guide vehicle*), intertravamento de máquinas, identificação de conflito no fluxo de peças, sistema de gerenciamento de *buffer*, plano de processo alternativo e outros, auxiliando na busca por soluções de projeto e controle de tais subsistemas.

Sistemas reais exigem modelos grandes para representá-los, com muitos elementos gráficos, tornando sua análise e interpretação uma tarefa difícil (RIGHINI, 1993). Visando solucionar esse problema, várias extensões de PN foram propostas, tais como: PN Modular (MPN) e PN Orientada a Objetos (OOPN), entre outras.

Além de outras extensões, alguns autores como LEE & FAVREL[1] apud WANG (1996) concentram esforços nos métodos de síntese, para reduzir o tamanho de uma Rede de Petri e facilitar a análise das propriedades do sistema. Porém o esforço na construção e desenvolvimento de um *software* baseado em PN não é reduzido porque depende de propriedades que não existem, como: modularidade, reusabilidade e manutenibilidade que são comumente solicitadas em sistemas de controle modernos.

[1] LEE, K.H. & FAVREL, J. (1985). Hierarquical reduction method for analysis and decomposition of Petri Nets, *IEEE Transation Systems Man & Cybernetics*, v. 15, n. 1, p. 272-280 apud WANG, L.C. (1996). Object-oriented Petri Net for modeling and analysis of automated manufacturing systems. *Computer Integrated Manufacturing Systems*, v. 26, n. 2, p. 111-125.

Com o objetivo de auxiliar na modelagem de Sistemas, INAMASU (1995) criou uma nova extensão de PN, a Rede de Petri Virtual. Com ela é possível modelar um sistema de forma modular, trabalhando cada um dos seus elementos de forma independente, porém mantendo a comunicação entre eles através dos nós virtuais, os quais recebem essa nomenclatura porque referenciam elementos de outros módulos e não fazem parte do objeto modelado.

Através dos nós virtuais, é possível a junção dos módulos, gerando um modelo final representativo do sistema completo. O modelo final é uma Rede de Petri Lugar/Transição, que não contém elementos virtuais, tornando possível empregar as técnicas de análise de PN Lugar/Transição para avaliar conflitos, vida, limitação, persistência, entre outras propriedades e técnicas de análise.

1.2. Motivações

O crescimento contínuo dos sistemas e a aplicação cada vez mais generalizada da informática nos sistemas de eventos discretos têm resultado em modelos de Rede de Petri complexos, com muitos elementos gráficos, que tornam a sua aplicação na modelagem de projetos inviável. A solução para viabilizar o uso de uma ferramenta tão poderosa quanto a Rede de Petri é o seu uso modular. A modularização fornece flexibilidade ao projeto, podendo alocar e realocar os recursos do sistema da maneira desejada, mudando o modelo final e os resultados. Há também a possibilidade de reaproveitar módulos de um sistema em outros sistemas semelhantes, resultando em economia de tempo no projeto.

O objetivo desse trabalho é formalizar a definição e aplicação das Redes de Petri Virtuais aos Sistemas de Eventos Discretos. Para tanto, melhora a definição de VPN fornecida por INAMASU (1995) e MORANDIN JÚNIOR (1999), define um algoritmo para sua utilização e um procedimento de junção dos módulos através dos nós virtuais. Dessa forma, viabiliza seu uso de forma clara e intuitiva, permitindo a implementação desta metodologia em aplicativos de *software*.

1.3. Organização do trabalho

A seguir, é apresentado o conteúdo dos capítulos que compõem o trabalho.

Capítulo 2: Revisão Bibliográfica
Apresenta uma revisão de Rede de Petri Lugar/Transição e de Sistemas de Eventos Discretos. Define as Redes de Petri, e apresenta suas propriedades comportamentais e estruturais, métodos de análise, principais extensões e aplicação aos sistemas de eventos discretos.

Capítulo 3: Rede de Petri Virtual
Nesse capítulo é formalizada Rede de Petri Virtual, sua definição, algoritmo de utilização, procedimento de junção de nós virtuais e exemplifica a associação de módulos através de um exemplo de sistema de manufatura.

Capítulo 4: Estudo de Caso
Um exemplo de Sistema Automatizado de Manufatura é modelado utilizando a metodologia apresentada no capítulo 3.

Capítulo 5: Comentários Finais e Conclusões
Apresenta as conclusões sobre o trabalho realizado e as sugestões para continuidade sobre o assunto.

Capítulo 6: Referência Bibliográfica
Apresenta o material utilizado na pesquisa bibliográfica.

2. Revisão Bibliográfica

2.1. Considerações Iniciais

Este capítulo apresenta uma revisão bibliográfica da definição, propriedades, métodos de análise e extensões das Rede de Petri Lugar/Transição, Sistemas de Eventos Discretos, e por último, alguns trabalhos do uso de PN na modelagem, análise e simulação de SED's.

2.2. Redes de Petri

2.2.1. Elementos Básicos

Redes de Petri (PN) é um tipo particular de grafo direcionado, com um estado inicial chamado marcação inicial (M_0). Consiste de dois tipos de nós, lugares (P) e transições (T), ligados por arcos (F) que saem de lugares para transições e de transições para lugares.

Lugares são representados graficamente por círculos e indicam um estado ou condição do sistema. As transições são representadas por caixas ou barras e indicam um evento. Os arcos possuem pesos (W) que são números inteiros positivos. Pesos unitários geralmente são omitidos. Os lugares cujos arcos chegam a uma transição *t* são conhecidos como lugares de entrada de *t*. E os lugares cujos arcos saem de *t* são os lugares de saída de *t*.

2.2.2. Marcação

Uma marcação, denotada por M, é um vetor coluna cuja dimensão é dada pelo número total de lugares da rede e assinala para cada lugar um número inteiro não negativo. Se uma marcação atribui a um lugar *p* um número inteiro *k*, diz-se que *p* está marcado com *k* marcas. Marcas são representadas graficamente por pontinhos pretos que percorrem os lugares da PN. M(p) é o número de marcas do lugar *p*. Uma rede pode ou não ter uma marcação inicial (M_0), ou seja, uma distribuição inicial de marcas nos lugares.

2.2.3. Definição Formal de Rede de Petri Lugar/Transição

A definição formal (MURATA, 1989) de Rede de Petri, N, é dada por uma quíntupla: $N = (P, T, F, W, M_0)$, onde:

TABELA 1. Definição Formal de Rede de Petri (MURATA, 1989)

$P = \{p_1, p_2, ..., pn\}$	conjunto finito de lugares	○
$T = \{t_1, t_2, ..., tn\}$	conjunto finito de transições	▯
$F \subseteq (P \times T) \cup (T \times P)$	conjunto de arcos	→
$W: F \rightarrow \{1, 2, 3, ...\}$	função peso	$\xrightarrow{W}$
$M_0: P \rightarrow \{0, 1, 2, 3, ...\}$	marcação inicial	⊙

$P \cap T = \emptyset$ e $P \cup T \neq \emptyset$

A estrutura da Rede de Petri N = (P, T, F, W) sem qualquer marcação inicial específica é denotada apenas por N. Caso tenha uma marcação inicial, é denotada por (N, M_0).

A nomenclatura para definir um pré-conjunto e um pós-conjunto de transições e lugares de entrada e saída é apresentada a seguir:

$\bullet p = \{t \mid (t,p) \in F\}$ é o conjunto de transições de entrada de p,

$p\bullet = \{t \mid (p,t) \in F\}$ é o conjunto de transições de saída de p,

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

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

Onde: $\bullet p \subseteq$ T, $p\bullet \subseteq$ T, $\bullet t \subseteq$ P e $t\bullet \subseteq$ P.

2.2.4. Habilitação e Disparo de Transição

O comportamento de muitos sistemas pode ser descrito pelo seu estado e suas mudanças. Para simular o comportamento dinâmico de uma PN, um estado (marcação) é mudado de acordo com a seguinte **regra de transição** (MURATA, 1989):

a) Uma transição t é dita estar habilitada se cada lugar de entrada p de t é marcada com no mínimo w(p,t) marcas, onde w(p,t) é o peso do arco de p para t.
b) Uma transição habilitada pode ou não disparar.
c) O disparo de uma transição habilitada t remove w(p,t) marcas de cada lugar de entrada p de t e adiciona w(t,p) marcas para cada lugar de saída p de t, onde w(t,p) é o peso do arco de t para p.

A Fig. (2.1) ilustra a Regra de Transição usando a reação química da água: $2H_2 + O_2 \rightarrow 2H_2O$. As duas marcas em cada lugar de entrada da transição t da Fig. (2.1-a) mostra que duas unidades de H_2 e O_2 estão disponíveis e que a transição t está habilitada (MURATA, 1989). Depois do disparo da transição t, a marcação será mudada conforme pode ser visto na Fig. (2.1-b) e a transição t não está mais habilitada.

FIGURA 2.1. Ilustração da Regra de Transição (MURATA, 1989)

2.2.5. Propriedades Comportamentais

Algumas propriedades de PN dependem da marcação inicial, ou seja, leva em consideração o estado da rede no momento inicial. Tais propriedades são chamadas de "Propriedades Comportamentais" e levam em consideração o comportamento dinâmico da rede.

2.2.5.1. Alcançabilidade (*Reachability*)

Alcançabilidade é a base fundamental para o estudo das propriedades dinâmicas de alguns sistemas. O disparo de uma transição habilitada mudará a distribuição de marcas (marcação) na rede de acordo com a Regra de Transição (apresentada na seção 2.2.4). Uma seqüência de disparos resultará em seqüências de marcações. Uma marcação M_n é dita ser alcançável de uma marcação M_0 se existe uma seqüência de disparos que transforma M_0 para M_n. Um disparo ou seqüência de ocorrência é denotado por $\sigma = M_0\ t_1\ M_1\ t_2\ M_2 \ldots t_n\ M_n$ ou simplesmente, $\sigma = t_1\ t_2 \ldots t_n$. No caso, M_n é alcançável de M_0 através de σ e escreve-se $M_0[\sigma > M_n$. O conjunto de todas as possíveis marcações alcançáveis de M_0 através de σ em uma rede (N, M_0) é denotada por $R(N, M_0)$ ou simplesmente $R(M_0)$. O conjunto de todas as possíveis seqüências de disparo de M_0 em uma rede (N, M_0) é denotada por $L(N, M_0)$ ou simplesmente $L(M_0)$.

O problema da alcançabilidade para Redes de Petri é descobrir se $M_n \in R(M_0)$ em uma rede (N, M_0). Em algumas aplicações, pode ser interessante analisar apenas as marcações de um subconjunto de lugares. Isto conduz ao problema de submarcações

alcançáveis em que procura-se descobrir se $M'_n \in R(M_0)$, onde M'_n é qualquer marcação que pertença ao subconjunto de lugares combinados com a marcação M_n.

Segundo CARDOSO & VALETTE (1997) se o conjunto $R(M_0)$ é finito, ele pode ser representado através de um grafo, cujos nós são as marcações acessíveis. Um arco orientado liga dois nós M_i e M_j se existe uma transição *t* habilitada que permite passar da marcação M_i a M_j.

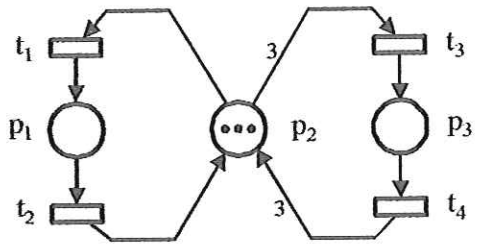

FIGURA 2.2. Rede de Petri (CARDOSO & VALETTE, 1997)

A Fig. (2.3) representa o grafo das marcações acessíveis da Rede de Petri da Fig. (2.2), cuja marcação inicial é $M_0 = [0\ 3\ 0]^t$. Esse grafo facilita a análise da Alcançabilidade.

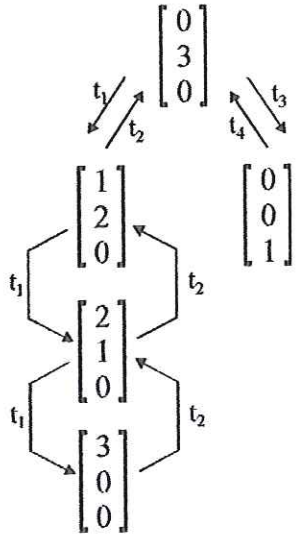

FIGURA 2.3. Grafo de marcações acessíveis (CARDOSO & VALETTE, 1997)

2.2.5.2. Limitabilidade (*Boundedness*)

Um lugar *p* de uma Rede de Petri (N, M_0) é *k*-limitado ou simplesmente limitado, se $M(p) \leq k$. Se $k = 1$, diz-se que o lugar é **binário** ou **seguro**. Uma Rede de Petri N é k-limitada se, e somente se, todos os seus lugares são k-limitados. Se todos os lugares de uma rede são 1-limitados ou binários, diz-se que a rede é **segura**.

Para a Rede de Petri da Fig. (2.2), com $M_0 = [0\ 3\ 0]^t$, tem-se que o lugar p_3 é 1-limitado e os lugares p_1 e p_2 são 3-limitados.

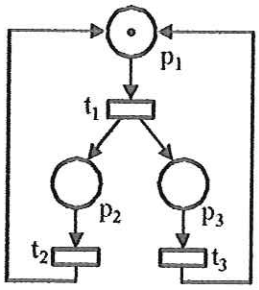

FIGURA 2.4. Rede de Petri não limitada (CARDOSO & VALETTE, 1997)

Considere a Rede de Petri da Fig. (2.4), com a marcação inicial $M_0 = [1\ 0\ 0]^t$. Cada vez que a seqüência $\sigma = t_1\ t_2$ é disparada, uma marca é introduzida no lugar p_3. Este lugar é não limitado e por sua vez a rede também não o é.

Lugares em Rede de Petri são geralmente utilizados para representarem armazenamento (*buffers*) e registros de armazenamento temporário de dados. Verificando a limitabilidade da rede, garante-se que não ocorrerá *overflows* nos *buffers* ou nos registros, não importando a seqüência de disparo.

2.2.5.3. Vivacidade (*Liveness*)

Uma transição *t* de uma Rede de Petri (N, M_0) é viva se for possível habilitá-la de qualquer marcação $M' \in R(M_0)$, através de uma seqüência σ de disparo. Uma Rede de Petri (N, M_0) é viva se, e somente se, todas as suas transições são vivas.

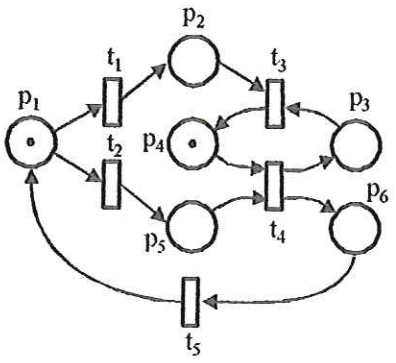

FIGURA 2.5. Rede de Petri não viva (MURATA, 1989)

A rede da Fig. (2.2) é viva para $M_0 = [0\ 3\ 0]^t$. A rede da Fig. (2.5) é um exemplo de rede não viva, pois nenhuma transição pode disparar se t_1 não disparar primeiro.

Uma transição *t* em uma Rede de Petri (N, M_0), pode ter diferentes níveis de vida:

- Morta se *t* nunca será disparada, para qualquer seqüência de disparo a partir de M_0;
- L1-viva (potencialmente disparável) se existe pelo menos uma seqüência que dispara *t*, a partir de M_0;
- L2-viva se, dado um número inteiro *k*, *t* pode ser disparada pelo menos *k* vezes em alguma seqüência de disparo a partir de M_0;
- L3-viva se *t* aparece infinitamente, ou muitas vezes em alguma seqüência de disparo a partir de M_0 e
- L4-viva ou viva se *t* é L1-viva para toda marcação M em $R(M_0)$.

Uma Rede de Petri viva garante operações livres de bloqueios (*deadlocks*), não importando qual seqüência disparável é escolhida. Garante também a ausência de partes mortas (nunca atingidas).

2.2.5.4. Reversibilidade (*Reversibility*)

Uma Rede de Petri (N, M_0) é dita reversível se, para cada marcação M em $R(M_0)$, M_0 é alcançável por M. Então, uma rede reversível pode sempre voltar para o seu estado

inicial. Em muitas aplicações, não é necessário voltar ao estado inicial, basta voltar para algum outro estado (estado interno). Uma marcação M' é dita ser um "estado interno" se, para cada marcação M em $R(M_0)$, M' é alcançável por M.

2.2.5.5. Abrangência de Cobertura (*Coverability*)

Uma marcação M em uma Rede de Petri (N, M_0) é dita ser abrangente de cobertura (*coverable*) se existir uma marcação M' em $R(M_0)$ tal qual $M'(p) \geq M(p)$, para cada lugar *p* na rede. Abrangência de cobertura está relacionada a L1-vivacidade (potencialmente disparável). Considere que M é a marcação mínima necessária para habilitar uma transição *t*. Então *t* é morta se e somente se M não é abrangente de cobertura, isto é, *t* é L1-viva se e somente se M é abrangente de cobertura.

2.2.5.6. Persistência (*Persistence*)

Uma Rede de Petri (N, M_0) é dita persistente se, para quaisquer duas transições habilitadas, o disparo de uma não desabilita a outra. Uma transição em uma rede persistente, uma vez habilitada, permanecerá habilitada até que dispare. A persistência está relacionada a sistemas livres de conflito e é muito útil no contexto de programas paralelos e em circuitos assíncronos que independem de velocidade. A rede da Fig. (2.6) é persistente, o disparo de t_1 habilita t_2 e t_3, mas o disparo de t_2 não desabilita t_3 e vice-versa. Já a rede da Fig. (2.2) é não persistente, t_1 e t_3 estão habilitadas para $M_0 = [0\ 3\ 0]^t$ o disparo de t_1 desabilita t_3.

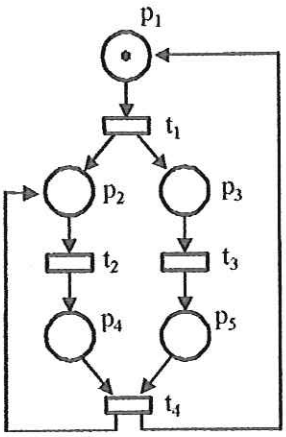

FIGURA 2.6. Rede de Petri persistente (MURATA, 1989)

2.2.5.7. Distância simultânea (*Synchronic Distance*)

A distância simultânea é um conceito fundamental introduzido por C. A. Petri. É uma métrica relacionada ao grau de dependência mútua entre dois eventos em um sistema condição/evento. Define-se a distância simultânea entre duas transições t_1 e t_2 em uma Rede de Petri (N, M_0) por:

$$d_{12} = \max\left|\overline{\sigma}(t_1) - \overline{\sigma}(t_2)\right| \qquad (1)$$

onde σ é uma seqüência de disparo iniciada por alguma marcação M em $R(M_0)$ e $\overline{\sigma}(t_i)$ é o número de vezes que a transição t_i, i = 1, 2 dispara em σ. Por exemplo, para a rede da Fig. (2.7) tem-se que $d_{12} = 1$, $d_{34} = 1$ e $d_{13} = \infty$.

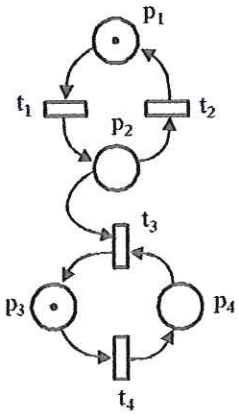

FIGURA 2.7. Rede de Petri (MURATA, 1989)

A distância simultânea dada pela Eq.(1) representa a melhor métrica definida para redes condição/evento. Contudo, há algumas dificuldades quando ela é aplicada a classes mais gerais de Rede de Petri.

2.2.6. Propriedades Estruturais

As propriedades estruturais de uma PN são independentes da marcação inicial e portanto, são determinadas pela estrutura topológica da rede e segue algumas definições:

i) Uma PN é estruturalmente limitada se é limitada para qualquer marcação inicial;

ii) Uma PN é estruturalmente viva se é viva para qualquer marcação inicial;

iii) Uma PN é conservativa se, para qualquer marcação inicial M_0 e uma marcação $M \in R(M_0)$, existe um vetor $X_{(n+1)}$ tal que:

$$X_i \neq 0, \text{ para qualquer } i = 1, ..., n \text{ e } X^T M = X^T M_0$$

iv) Uma PN é estruturalmente consistente se há uma marcação M e uma seqüência de disparo σ, chamada seqüência de disparo cíclica, e um vetor de disparo X, tal que:

- σ causa um retorno da rede para M;
- X não tem elementos nulos, isto é, σ dispara cada transição pelo menos uma vez.

v) Uma PN é parcialmente consistente se o vetor de disparo X tem alguns elementos nulos (algumas transições não vão disparar).

vi) Uma PN é completamente controlável se qualquer marcação é alcançável a partir de qualquer marcação inicial.

vii) Uma PN é repetitiva se existe uma marcação finita M_0 e uma seqüência de disparo σ, tal que os elementos do vetor de disparo associado X são infinitos. Se X contém somente algumas transições, a rede é parcialmente repetitiva.

2.2.7. Métodos de Análise

Os métodos de análise de PN podem ser classificados em três grupos (MURATA, 1989): o método da árvore de cobertura (ou de alcançabilidade), a matriz de incidência e técnicas de decomposição ou redução.

2.2.7.1. Árvore de Cobertura

Para uma Rede de Petri (N, M_0), pode-se construir sua Árvore de Cobertura. A partir de M_0 obtém-se *n* novas marcações, sendo *n* o número de transições habilitadas em M_0. De cada nova marcação, pode-se obter outras conforme o número de transições habilitadas. Este processo resulta em uma árvore, sendo a marcação inicial M_0 a raiz da árvore e cada transição habilitada por esta marcação um ramo. As marcações obtidas através do disparo das transições são calculadas e o processo recomeça para cada nova marcação obtida. A construção de um ramo é interrompida desde que seja encontrada uma marcação igual a uma outra já calculada, denominada "velha".

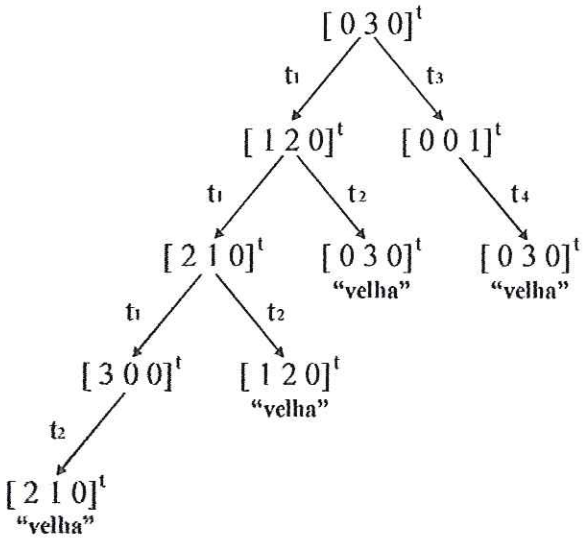

FIGURA 2.8. Árvore de Cobertura

A Fig. (2.8) apresenta a Árvore de Cobertura para a Rede de Petri da Fig. (2.2), com marcação inicial $M_0 = [0\ 3\ 0]^t$, que constitui a raiz da árvore. E como as transições t_1 e t_3 estão habilitadas a partir de M_0, elas formarão dois novos ramos da árvore cujos

nós são as marcações calculadas no disparo delas. E assim sucessivamente até que seja encontrada uma marcação "velha".

A partir da Árvore de Cobertura, algumas propriedades podem ser estudadas para uma Rede de Petri (N, M_0):

- A rede é Limitada se não acontecem "*loops*" na árvore, ou seja, $R(M_0)$ é finito.
- A rede é Salva se somente '0's e '1's aparecerem nos nós (marcações) da árvore.
- Uma transição *t* é morta se ela não aparece nos rótulos dos arcos (ramos) da árvore.
- Se M é alcançável de M_0, então existe um nó rotulado M' tal que $M \leq M'$.

Para uma Rede de Petri limitada, a árvore de cobertura recebe o nome de **Árvore de Alcançabilidade**, desde que ela contenha todas as marcações possíveis alcançáveis, ou seja, $R(M_0)$ é finito.

2.2.7.2. Matriz de Incidência

O comportamento dinâmico de muitos sistemas estudados na Engenharia pode ser descrito por equações diferenciais ou equações algébricas. A matriz de incidência analisa e descreve o comportamento dinâmico de sistemas modelados em Rede de Petri através de algumas equações. Entretanto, a resolução dessas equações é algo limitado, parte devido à natureza não-determinística inerente aos modelos de Rede de Petri e parte devido à restrição das soluções que devem ser inteiras e não-negativas.

Para uma Rede de Petri com *n* transições e *m* lugares, a matriz de incidência $A = [a_{ij}]$ é dada por uma matriz *n* x *m* de números inteiros e sua entrada típica é dada por:

$$a_{ij} = a_{ij}^{+} - a_{ij}^{-} \qquad (2)$$

onde $a_{ij}^{+} = w(i,j)$ é o peso do arco da transição i para seu lugar de saída j e $a_{ij}^{-} = w(j,i)$ é o peso do arco da transição i para seu lugar de entrada j.

$$A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & -3 & 1 \\ 0 & 3 & -1 \end{bmatrix} \tag{3}$$

A Rede de Petri da Fig. (2.2) possui a matriz de incidência, A_{4x3}, apresentada na Matriz (3).

A partir da matriz de incidência é possível obter qualquer marcação M_d alcançável de M_0 através da seqüência de disparo $\{u_1, u_2, ..., u_d\}$, onde u_k é um vetor coluna de n-1 '0's e uma entrada não zero, um '1' na posição *i* que indica que a transição *i* dispara. A Eq. (4) é conhecida como a Equação Fundamental da Rede de Petri.

$$M_d = M_0 + A^t \sum_{k=1}^{d} u_k \tag{4}$$

Ainda para a rede da Fig (2.2), tem-se que a aplicação da Eq. (4) para obter M_1:

$$M_1 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

2.2.7.3. Métodos de Redução

Para facilitar a análise de um sistema grande, modelado em Rede de Petri, pode-se reduzir o modelo para outro mais simples, desde que o último preserve as propriedades a serem analisadas do primeiro. As técnicas de redução apresentadas nessa seção são transformações simples que podem ser usadas para analisar vivacidade, segurança e limitabilidade.

Dada uma Rede de Petri (N, M_0), tem-se que (N', M_0') é a nova Rede de Petri obtida após a aplicação de algum desses métodos de Redução. Então, (N', M_0') é viva, segura ou limitada se, e somente se, (N, M_0) é viva, segura ou limitada.

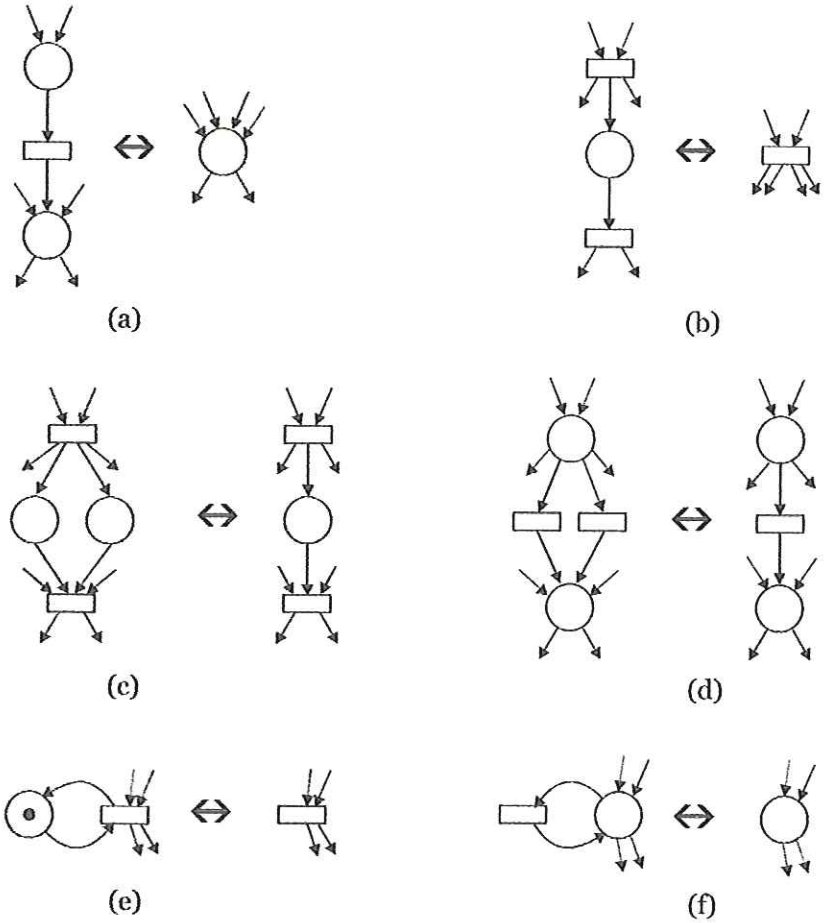

FIGURA 2.9. Métodos de Redução (MURATA, 1989)

A Figura (2.9) apresenta seis operações que preservam as propriedades de vivacidade, segurança e limitabilidade:

- fusão de lugares em série (Figura 2.9-a);
- fusão de transições em série (Figura 2.9-b);
- fusão de lugares em paralelo (Figura 2.9-c);
- fusão de transições em paralelo (Figura 2.9-d);
- eliminação de *loops* nos lugares (Figura 2.9-e) e
- eliminação de *loops* nas transições (Figura 2.9-f).

Por exemplo, a rede mostrada em Fig. (2.7) pode ser reduzida para a rede mostrada na Fig. (2.10) depois do disparo de t_2 para remover a marca de p_1, faz-se a fusão de t_1

e t_2 em t_{12}, e a fusão de t_3 e t_4 em t_{34}. As redes mostradas nas Fig. (2.9) e Fig. (2.10) são limitadas e não-vivas.

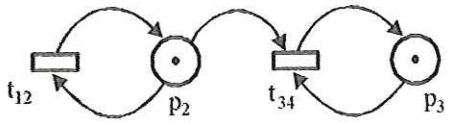

FIGURA 2.10. Rede de Petri reduzida (MURATA, 1989)

2.2.8. Algumas extensões

Redes de Petri têm sua origem na dissertação de Carl Adam Petri, submetida em 1962 a Faculdade de Matemática e Física da Universidade Técnica de Darmstadt, Alemanha Ocidental.

Com o intuito de aumentar sua aplicabilidade a sistemas variados e aumentar o poder de PN na representação dos mesmos, muitas extensões de PN têm sido propostas desde a sua criação. As extensões mais importantes são:

- Rede de Petri Condição/Evento
- Rede de Petri Lugar/Transição
- Rede de Petri Predicado/Transição
- Rede de Petri Colorida
- Rede de Petri Orientada a Objetos
- Rede de Petri Modular
- MFG (*Mark Flow Graph)*
- Grafcet
- Rede de Petri Temporizada
- Rede de Petri Estocástica
- Rede de Petri Hierárquica
- Rede de Petri com Lógica Nebulosa
- SFC (*Sequencial Function Chart*)

Segundo MOORE & GUPTA (1996), pode-se dividir Redes de Petri e suas extensões em dois grupos. O primeiro introduz novos atributos que aumentam o poder de abstração, tais como PN Estocástica, PN Orientada a Objeto e PN com Lógica Nebulosa. O segundo grupo aumenta a facilidade no projeto de sistemas, tais como: MFG, SFC, PN Colorida, PN Hierárquica, PN Modular e outras.

O Grafcet e o MFG são duas das técnicas mais representativas para a aplicação industrial e que essencialmente são equivalentes. Eles representam a base teórica do SFC (MIYAGI, 1996).

RIGHINI (1993) e JENG & DICESARE (1992) apresentam a extensão modular de Rede de Petri, no qual um modelo grande é subdividido em submodelos, denominados módulos. Marcas fluem de um módulo para outro, através dos elementos globais (lugares e transições), responsáveis pela conexão do modelo. Essa extensão permite o estudo de uma determinada sub-região sem necessidade do desenvolvimento do modelo completo.

Uma Rede de Petri orientada a objetos (OOPN) é modular. Baseia-se no conceito de objetos e passagem de mensagens do paradigma de programação orientada a objetos (POO). A passagem de mensagens promove a intercomunicação entre os objetos através de marcas coloridas. A construção de um modelo em OOPN consiste em dois passos:

a) Construção do modelo básico. Considera somente o comportamento dinâmico de cada objeto físico e a passagem de mensagem.
b) Construção do modelo completo. Envolve a política de controle/decisão na resolução de conflitos. (MEDEIROS, 1998 e WANG, 1996).

2.3. Sistemas de Eventos Discretos

Sistemas de Eventos Discretos (SED) são sistemas concebidos pelo homem, definidos pela ocorrência dinâmica de eventos físicos discretizados em intervalos irregulares e desconhecidos (RAMADGE & WONHAM, 1989). São exemplos de SED's os sistemas de manufatura, de robótica, de comunicação, gerenciamento de tráfego e redes de computadores, entre outras.

Nos SED's, a ocorrência de um evento, por exemplo, a chegada de uma peça numa máquina CNC, causa a alteração na configuração interna ou nos estados do sistema, a qual será mantida até a ocorrência de um novo evento. Esses eventos são por natureza instantâneos (tempo de duração desprezível em relação à duração dos estados) o que lhe confere caráter discreto. Em contraposição, existem os chamados Sistemas de Variáveis Contínuas (SVC) dirigidos pelo tempo, e cuja dinâmica é descrita por variações contínuas (GUSTIN, 1999).

Para auxiliar o projeto de sistemas complexos, pesquisadores têm desenvolvido e aprimorado as técnicas existentes, para produzir especificações mais inteligentes, facilitando a transição entre o projeto e a implementação (AZZOPARDI & HOLDING, 1997). As principais características dos SED's que devem ser identificadas, tratadas e analisadas na modelagem de projetos são: sincronismo, concorrência, conflitos, compartilhamento de recursos, ligações preferenciais, seqüência de eventos, não determinismo e gargalhos (*deadlocks*).

Os primeiros pesquisadores da teoria dos SED's basearam a representação de tais sistemas nos modelos de linguagem formal e autômatos. Porém, os autômatos apresentavam grande complexidade computacional na representação de sistemas grandes, constituídos por inúmeras interações de seus subsistemas. Devido à vantagem na representação gráfica e distribuída dos estados do sistema e a eficiência computacional, PN tem sido largamente usada nos controle de SED's (WU, SU & CHU, 2002).

MIYAGI et al. (1989) aborda as principais metodologias aplicadas na modelagem de SED's considerando que as principais dificuldades estão relacionadas com a inadequação dos modelos para tratar sistemas concorrentes e assíncronos com alto grau de complexidade. As metodologias abordadas nesse trabalho são:

a) Teoria das Filas: Nessa técnica, o modelo é concebido com base no fluxo de materiais onde técnicas de análise de fila são aplicadas para investigar o comportamento do sistema. A técnica permite a modelagem de paralelismo e concorrência. Porém, o parâmetro de análise desejado é expresso em termos da função de propagação do ganho (diferença entre os instantes de ocorrência dos eventos numa seqüência). Sua vantagem é a baixa carga computacional.

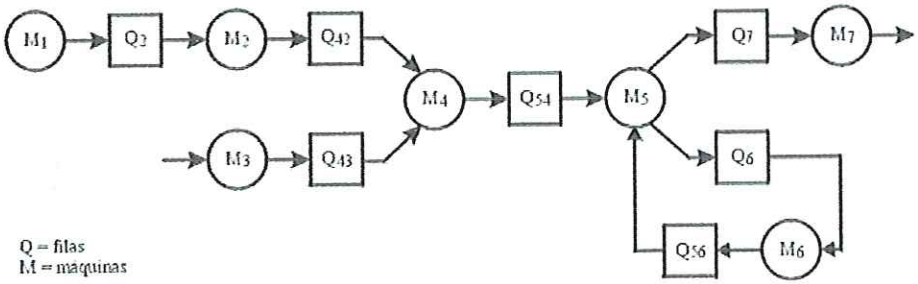

FIGURA 2.11. Modelo de filas de um SED

b) Formulação Algébrica: essa metodologia é aplicada basicamente para a modelagem de processos repetitivos. Seu ponto forte é que as informações sobre o desempenho do sistema podem ser obtidas diretamente das equações algébricas, ao contrário das representações gráficas que necessitam ser simuladas, o que às vezes consome muito tempo. Uma limitação é que a seqüência de eventos e de recursos do sistema é fixa.

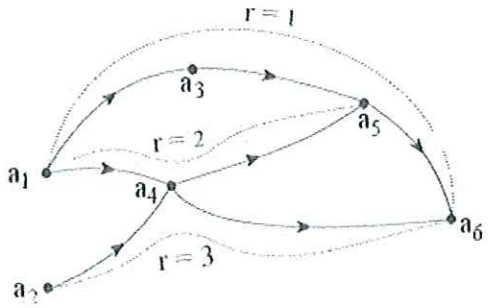

FIGURA 2.12. Grafo orientado

c) Linguagens Formais: Essa metodologia modela SED's através de autômatos finitos segundo a teoria de linguagens formais. Os autômatos finitos são determinísticos a menos que seja considerado um número infinito de estado e não são capazes de modelar concorrência e paralelismo de forma satisfatória.

d) Redes de Petri: Dentre os modelos baseados na teoria de redes, as Redes de Petri se destacam pela precisão, simplicidade e elegância com que conseguem modelar as principais características de diversos tipos de sistemas. A abordagem dessa metodologia utiliza a *Mark Flow Graph* (MFG) que é um grafo derivado da Rede de Petri, onde a modelagem respeita a estrutura hierárquica do sistema, onde cada um dos níveis é representado de forma consistente, facilitando assim tarefas de depuração, análise e implementação. A sua principal vantagem é a facilidade com que se pode implementar programas computacionais que os simulam.

2.4. Rede de Petri e os Sistemas de Eventos Discretos

Ferramentas de modelagem gráfica, como diagramas, esboços, tabelas, etc., são importantes nos projetos de SED's por constituírem uma maneira básica e visual de representá-los.

As ferramentas gráficas mais utilizadas são:

i) Diagrama de atividades e de dados: representam as atividades e o fluxo de dados, respectivamente, de um sistema. São usadas para modelagem de processos conectados como fluxo de material e informação.

ii) GRAI: auxilia nos processos de tomada de decisão baseado na especificação do modelo de um sistema. Consiste de um conjunto de três subsistemas: modelo físico (pessoas, materiais, etc) que transformam os componentes em produtos finais; subsistema de decisão, que controla o subsistema físico, e o subsistema de informação, que une os subsistemas físico e de decisão (ARMC, 2003).

iii)SADT (*Strutured Analysis and Design Technique*): utiliza diagramas de atividades, que são decompostas gradualmente, formando um conjunto hierárquico, aumentando os detalhes das atividades do sistema.

iv)IDEF: é uma variante do método SADT, usado na modelagem funcional; consiste de 13 diagramas, um índice e dicionário de dados mostrando a complexidade de um sistema (KBSI, 2003).

v) Linguagens formais e autômatos: linguagem formal é definida por um alfabeto e uma regra de formação; o alfabeto é um conjunto de símbolos que constróem a linguagem e a regra de formação especifica como as palavras de símbolos serão construídas (SAKHAROV, 2003). Linguagem formal auxilia na modelagem funcional de SED's através de autômatos (máquinas de estados) bastante usados na computação.

vi)Redes de Petri: são ferramentas eficientes para modelagem e análise de sistemas que se sobrepõem às ferramentas citadas devido à sua formalização matemática e ao seu caráter dinâmico. Sua modelagem gráfica é amplamente utilizada na análise e controle de sistemas de manufatura, como ferramenta visual. A modelagem matemática pode ser usada para montar equações de estado, equações algébricas, simulações e outros modelos matemáticos. Assim, além da representação visual de sistemas, é possível obter mais informações sobre o comportamento do sistema modelado, através da análise de suas propriedades e de sua capacidade de simulação.

Através das propriedades de Redes de Petri pode-se extrair informações do sistema como:

i) Lugares em Rede de Petri são freqüentemente usados na representação de *buffers* e registros de dados armazenados imediatamente. Verificar se uma rede é **limitada** ou **segura**, garante que não haverá *overflows* nos *buffers* ou registros, independente da seqüência disparada no momento;

ii) As **marcações** da PN auxiliam a identificar quais áreas do sistema estarão ocupadas ou livres em um dado momento (CECIL et al., 1992);

iii) A propriedade de **alcançabilidade** auxilia a identificar se haverá alguma área do sistema que nunca será alcançada (áreas mortas);

iv) Se uma rede é **viva** e o modelo está correto, isso indica a ausência de *deadlocks* nas operações do sistema (CECIL et al., 1992) e

v) A **persistência** está relacionada a redes livres de conflitos.

FELDMANN, SCHNUR & COLOMBO (1996) apresenta a abordagem da utilização de Redes de Petri para modelar o controle de uma célula de produção flexível (sistema de manufatura), que conduz para o projeto de sistema de eventos discretos distribuído. Nesse trabalho, a ocorrência de conflitos são identificados, classificados e tratados e a complexidade da modelagem é tratada de forma modular.

PROTH & SAVI (1992) são os precursores da modelagem por módulos para sistemas de manufatura. Com o aumento da complexidade dos sistemas de eventos de discretos em geral, a modelagem utilizando Redes de Petri tornou-se muito onerosa devido à quantidade elevada de elementos na rede. A proposta é a modelagem de cada elemento do sistema modularmente e agrupá-los para obter o modelo de todo o sistema.

MORANDIN JÚNIOR et al. (2000) também aborda a modelagem por módulos de sistemas de manufatura automatizados utilizando Redes de Petri, considerando os recursos compartilhados do sistema. A proposta é a modelagem de cada elemento do sistema e a junção dos modelos obtendo um modelo final representativo de todo o sistema.

AUSFELDER et al. (1994) propõe a modelagem por módulos e hierárquica de sistemas de manufatura flexível orientada a seus elementos físicos, cuja ferramenta de modelagem escolhida é a Rede de Petri Orientada a Objetos (OOPN). Na OOPN, as marcas são chamadas objetos e seus valores podem ser testados e alterados por métodos definidos pelas transições. O modelo final é obtido através da junção de transições.

WU, SU & CHU (2002) apresenta o método de modelagem de controle de SED's utilizando PN com arcos habilitados (recente extensão de PN). Os arcos habilitados têm a vantagem, sob os arcos ordinários, por não conduzir uma rede ao conflito.

MORAES & CASTRUCCI (2002) apresentam uma metodologia para projeto e documentação de SED's baseada nas Redes de Petri, aplicada no ensino de alunos de graduação de Automação Industrial. Segundo os autores, as vantagens do uso das Redes de Petri aplicadas a SED são:

i) Capturar com facilidade as relações de causalidade e os vínculos estruturais dos sistemas de eventos discretos reais;
ii) Conexão intuitiva com os fenômenos reais dos SED's;
iii) Modelar conflitos, deadlocks e filas e
iv) Ter fundamento matemático, embora fortemente linear.

3. Rede de Petri Virtual

3.1. Considerações Iniciais

Rede de Petri é uma ferramenta gráfica eficiente para modelagem de sistemas discretos. Ela se sobrepõe às demais ferramentas gráficas devido a sua formalidade que, além da modelagem, permite a análise do sistema através de suas propriedades. Porém, sistemas reais geralmente produzem modelos grandes e complexos que torna sua análise impraticável.

Para a solução desse problema, técnicas de síntese e de modelagem por módulos têm sido amplamente utilizadas. JENG (1995) cita as técnicas de síntese ***top-down*** e ***bottom-up***. O método *top-down* inicia com o modelo agregado e refina-o gradativamente, até obter o modelo final. Tem a vantagem da visualização do sistema como um todo e a desvantagem da dificuldade em modelar sistemas com recursos compartilhados. O método *bottom-up* primeiro modela os subsistemas (em módulos), ignorando quaisquer interações entre eles, somente as considerando na junção dos subsistemas para obter o modelo final. Este método tem a vantagem dos subsistemas serem facilmente identificados porque eles usualmente representam recursos físicos e a principal desvantagem é que a integração dos subsistemas pode ser "forçada" para garantir propriedades ou a rede final pode não possuir propriedades desejáveis.

Dentro desse enfoque, este capítulo apresenta a Rede de Petri Virtual (VPN), criada por INAMASU (1995), que é uma nova extensão de Rede de Petri Lugar/Transição, que une o poder das Redes de Petri na representação e controle de sistemas à facilidade da modelagem por módulos, próxima ao método *bottom-up*.

3.2. Algoritmo para utilização de PN Virtual

O algoritmo que permite modelar sistemas utilizando Rede de Petri Virtual segue os passos seguintes:

1. **Fase de Caracterização**: Nesta fase é feito um estudo dos recursos do sistema a ser modelado, dividindo-o em subsistemas que instanciarão os módulos. Geralmente os módulos representam os recursos físicos do sistema, como: robôs, dispositivos de transporte, máquinas, etc.

2. **Fase de Modelagem**: Cada subsistema encontrado no passo 1 é modelado utilizando Rede de Petri Virtual. A definição dos nós virtuais (apresentados na seção 3.3) deve ser cuidadosa, pois eles representam as interações entre os subsistemas. Nesta fase, obtém-se um número de módulos de PN Virtual conforme o número de subsistemas encontrados no passo 1.

3. **Fase de Associação**: Nesta fase é realizada a junção dos módulos através dos nós virtuais. A união dos módulos é feita dois a dois.

   3.1. Escolhe-se o primeiro módulo e o segundo é obtido baseado na referência do nó virtual do primeiro.

   3.2. Se o nó virtual do primeiro referenciar um nó real, gera-se um novo nó real, que receberá os elementos de entrada e saída dos dois.

   3.3. Se o nó virtual do primeiro referenciar outro nó virtual, gera-se um novo nó virtual que receberá os elementos de entrada e saída dos dois e a referência do segundo nó virtual (vide seção 3.5.1).

   3.4. Repite-se os passos de 3.1 a 3.3 até que não haja elementos virtuais na rede final.

   Nessa fase deve-se tomar cuidado para não gerar conflitos e paralelismos indesejáveis. A rede obtida no passo 3.4, sem elementos virtuais, representará o sistema completo, essa rede é uma PN Lugar/Transição.

3.3. Elementos Estruturais

Rede de Petri Virtual herda os elementos da Rede de Petri Lugar/Transição, que desse momento em diante serão chamados de "elementos reais", e possui como elementos principais e adicionais, os **nós virtuais** (lugares e transições virtuais) e os **arcos virtuais**.

Um lugar (transição) virtual de um módulo faz referência a um outro lugar (transição) real ou virtual de outro módulo. Todos os arcos que chegam e saem de um nó virtual, é um arco virtual.

O padrão adotado neste trabalho para distinguir nós é: cor cinza para os nós reais e cor branca para os virtuais. Os arcos reais são representados por setas com linha cheia e os virtuais por linhas tracejadas. Os conceitos de Classes e Objetos da Programação Orientada a Objetos são aplicados neste trabalho. Uma classe pode ser vista como uma fôrma para gerar objetos idênticos. Os objetos são instâncias das classes. Os módulos são classes que instanciam objetos no modelo final. Caso uma classe (módulo) possua apenas um único objeto, a identificação do objeto pode ser ocultada.

As nomenclaturas adotadas para identificar os nós reais e virtuais são apresentadas nas Notações (1) e (2), respectivamente:

[tipo de nó] [identificador do módulo] [identificador do objeto] | [índice] (1)

Conforme a Notação (1) tem-se para um lugar ***i*** do módulo **A,** objeto **1**, a notação $\mathbf{p_{A1|i}}$ e, para uma transição ***i*** módulo **B,** objeto **1**, a notação $\mathbf{t_{B1|i}}$.

[tipo de nó virtual][identif. módulo][identif. objeto]|[índice] → [nó alvo]|[índice] (2)

Pela Notação (2) tem-se para um lugar virtual **i** do módulo **A,** objeto **1,** que referencia o lugar **j** do módulo **B,** objeto **2**, $\mathbf{vp_{A1|i \rightarrow B2|i}}$.

EESC - USP Biblioteca SÃO CARLOS

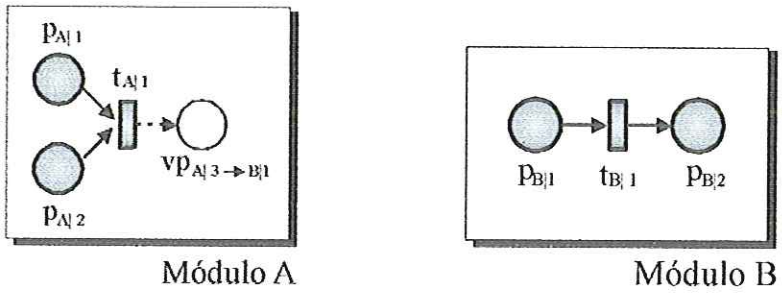

FIGURA 3.1. Notação dos nós em uma Rede de Petri Virtual

A Fig. (3.1) exemplifica as Notações (1) e (2) e o padrão de cores adotadas. O módulo A possui um nó virtual $vp_{A|3 \to B|1}$ que referencia o lugar $p_{B|1}$ do módulo B.

3.4. Definição Formal

MORANDIN JÚNIOR (1999) definiu uma Rede de Petri Virtual através da quíntupla (VP, VT, VF, VW, VM_0) apresentada na definição 1.

Definição 1: Uma Rede de Petri Virtual, VN, é definida pela quíntupla: VN = (VP, VT, VF, VW, VM_0), onde:

$VP = vP \cup P$	conjunto finito de lugares objetos
$VT = vT \cup T$	conjunto finito de transições objetos
$VF = vF \cup F$	conjunto finito de arcos objetos
$VW : VF \to \{1, 2, 3, ...\}$	função objeto de pesos
$VM_0 : VP \to \{0, 1, 2, ...\}$	estado inicial da rede

$VP \cup VT \neq \emptyset$; $VP \cup T \neq \emptyset$; $P \cup VT \neq \emptyset$ e $P \cup T \neq \emptyset$ e

$VP \cap VT = P \cap VT = VP \cap T = P \cap T = \emptyset$.

$vP = \{vp_{1+n}, vp_{2+n}, ..., vp_p\}$	conjunto finito de lugares virtuais
$vT = \{vt_{1+n}, vt_{2+n}, ..., vt_q\}$	conjunto finito de transições virtuais

$vF \subseteq (vP \times vT) \cup (vT \times vP) \cup (P \times vT) \cup (vT \times P) \cup (vP \times T) \cup (T \times vP)$ conjunto de arcos virtuais

A estrutura da Rede de Petri Virtual VN = (VP, VT, VF, VW) sem qualquer marcação inicial específica é denotada apenas por VN. Caso tenha uma marcação inicial, é denotada por (VN, VM_0).

Este trabalho complementa a definição de MORANDIN JÚNIOR (1999) com a definição 2 a seguir:

Definição 2: Uma Rede de Petri final, G, é denotada por um conjunto C = {VN's | VN's = (VP's, VT's, VF's, VW's, VM_0's), s = 1, 2, ..., *n*}, com *n* subredes (módulos) de Petri virtuais que compartilham nós virtuais. A rede final é uma rede lugar/transição G = (P', T', F', W', M'_0) de VN's tal que:

$$P' = \bigcup_{i=1}^{n} VP_i$$

$$T' = \bigcup_{i=1}^{n} VT_i$$

$$F' = \bigcup_{i=1}^{n} VF_i$$

$$W' : F \rightarrow \{1, 2, 3, ...\}$$

$$M'_0 : P \rightarrow \{1, 2, 3, ...\}$$

A nomenclatura para definir um pré-conjunto e um pós-conjunto de transições e lugares virtuais de entrada e saída é apresentada a seguir.

Pré e pós-conjuntos de transições virtuais (Fig. 3.2):

$v(\bullet p_k)$ conjunto de transições virtuais de entrada do lugar p_k

$\bullet p_k$ conjunto de transições de entrada do lugar p_k

$v(p_k\bullet)$ conjunto de transições virtuais de saída do lugar p_k

$p_k\bullet$ conjunto de transições de saída do lugar p_k

$v(\bullet vp_i)$ conjunto de transições virtuais de entrada do lugar virtual vp_i

$\bullet vp_i$ conjunto de transições de entrada do lugar virtual vp_i

$v(vp_i\bullet)$ conjunto de transições virtuais de saída do lugar virtual vp_i

$vp_i\bullet$ conjunto de transições de saída do lugar virtual vp_i

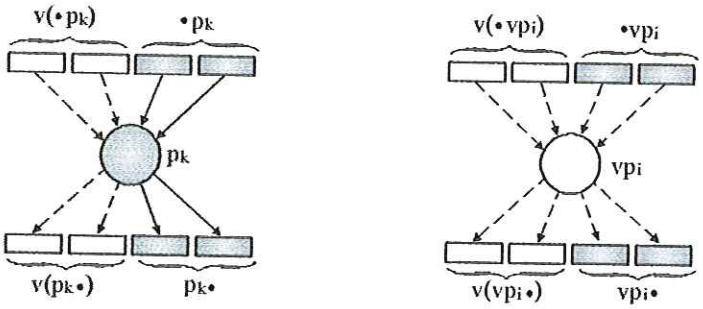

FIGURA 3.2. Pré-conjuntos e pós-conjuntos de transições virtuais

Pré e pós-conjuntos de lugares virtuais (Fig. 3.3):

$\bullet t_k$ conjunto de lugares de entrada da transição t_k

$v(\bullet t_k)$ conjunto de lugares virtuais de entrada da transição t_k

$v(t_k\bullet)$ conjunto de lugares virtuais de saída da transição t_k

$t_k\bullet$ conjunto de lugares de saída da transição t_k

$\bullet vt_i$ conjunto de lugares de entrada da transição virtual vt_i

$v(\bullet vt_i)$ conjunto de lugares virtuais de entrada da transição virtual vt_i

$v(vt_i\bullet)$ conjunto de lugares virtuais de saída da transição virtual vt_i

$vt_i\bullet$ conjunto de lugares de saída da transição virtual vt_i

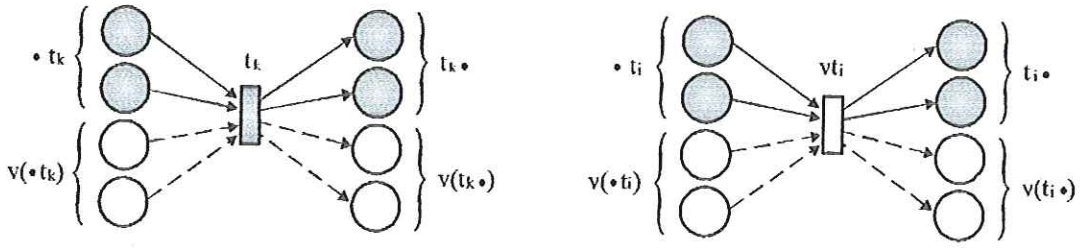

FIGURA 3.3. Pré-conjuntos e pós-conjuntos de lugares virtuais

3.5. Associação dos Módulos

Cada módulo do sistema é um subsistema modelado utilizando Rede de Petri Virtual e a junção deles é realizada considerando-se os nós virtuais. Um nó virtual de um módulo pode referenciar um nó real ou virtual de outro módulo.

3.5.1. Procedimento de junção

A associação de dois módulos segue a condição a seguir:

Se (nó virtual) referencia (nó virtual)

Então

Gera (novo nó virtual)

• (novo nó virtual) = • nó virtual ∪ • nó virtual referenciado

(novo nó virtual) • = nó virtual • ∪ nó virtual referenciado •

(novo nó virtual) ← referência do segundo virtual

Fim

Senão Se (nó virtual) referencia (nó real)

Gera (novo nó real)

• (novo nó real) = • nó virtual ∪ • nó real referenciado

(novo nó virtual) • = nó virtual • ∪ nó real referenciado •

Fim

Fim.

Ao associar dois nós, um virtual que referencia outro virtual, gera-se um novo nó virtual que terá como pré-conjunto a união dos pré-conjuntos e como pós-conjuntos a união dos pós-conjuntos dos dois. Esse novo nó virtual referenciará o nó virtual ou real do segundo virtual. Repite-se esse procedimento até que não reste nenhum nó virtual.

3.5.2. Exemplos de associações

Nesta seção, são apresentados alguns casos de associação entre dois módulos A e B, o passo intermediário da junção e o resultado final.

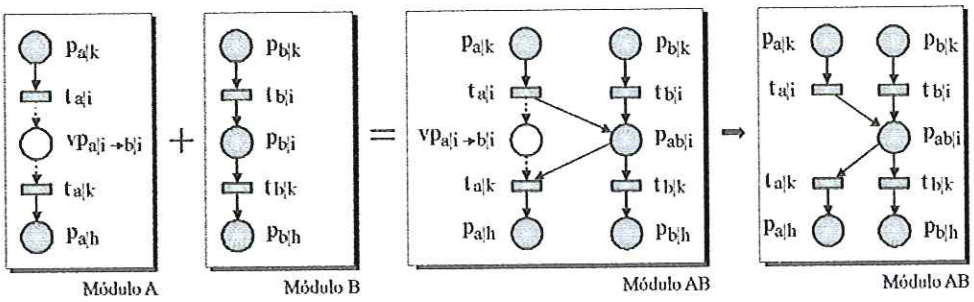

FIGURA 3.4. Exemplo 1 de associação de módulos

No primeiro exemplo de associação, apresentada na Fig. (3.4), o módulo A possui um lugar virtual que referencia um real do módulo B. Aplicando o Procedimento (apresentado na seção 3.3.1), obtém-se o módulo AB:

$$\bullet\, p_{ab|i} = \bullet\, vp_{a|i \to b|i} \cup \bullet\, p_{b|i}$$

$$p_{ab|i}\, \bullet = vp_{a|i \to b|i}\, \bullet \cup p_{b|i}\, \bullet$$

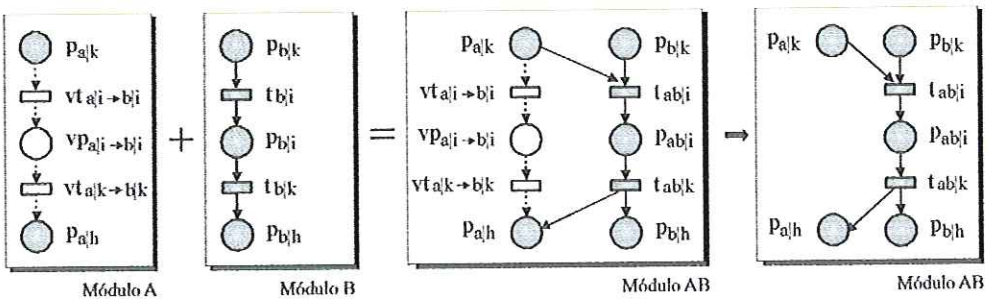

FIGURA 3.5. Exemplo 2 de associação de módulos

No segundo exemplo, Fig. (3.5), o módulo A possui três nós virtuais que referenciam três outros nós reais do módulo B. Aplicando o Procedimento (apresentado na seção 3.3.1), obtém-se o módulo AB:

$$\bullet\, t_{ab|i} = \bullet\, vt_{a|i \to b|i} \cup \bullet\, t_{b|i}$$

$$t_{ab|i}\, \bullet = vt_{a|i \to b|i}\, \bullet \cup t_{b|i}\, \bullet$$

$$\bullet\, p_{ab|i} = \bullet\, vp_{a|i \to b|i} \cup \bullet\, p_{b|i}$$

$$p_{ab|i}\, \bullet = vp_{a|i \to b|i}\, \bullet \cup p_{b|i}\, \bullet$$

$$\bullet\, t_{ab|k} = \bullet\, vt_{a|k \to b|k} \cup \bullet\, t_{b|k}$$

$$t_{ab|k}\, \bullet = vt_{a|k \to b|k}\, \bullet \cup t_{b|k}\, \bullet$$

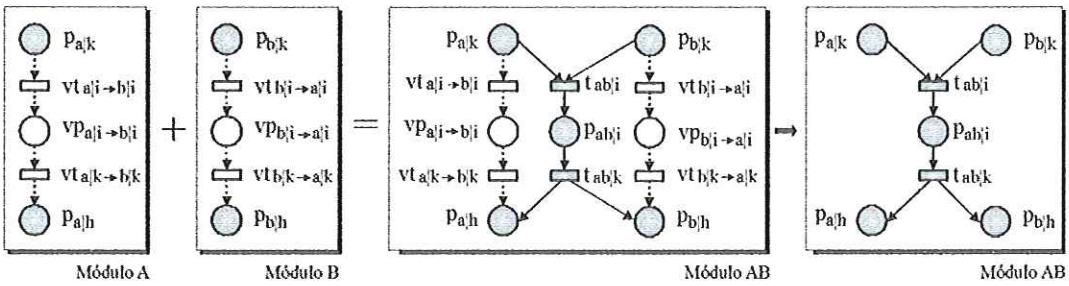

FIGURA 3.6. Exemplo 3 de associação de módulos

Neste exemplo, Fig. (3.6), o módulo A possui três nós virtuais que referenciam três nós virtuais do módulo B, e os nós virtuais de B referenciam os nós virtuais de A, ocorre uma referência cíclica, nesse caso, a junção cada um desses nós gerará três novos nós reais, onde:

$$\bullet\, t_{ab|i} = \bullet\, vt_{a|i \to b|i} \cup \bullet\, vt_{b|i \to a|i}$$

$$t_{ab|i}\, \bullet = vt_{a|i \to b|i}\, \bullet \cup vt_{b|i \to a|i}\, \bullet$$

$$\bullet\, p_{ab|i} = \bullet\, vp_{a|i \to b|i} \cup \bullet\, vp_{b|i \to a|i}$$

$$p_{ab|i}\, \bullet = vp_{a|i \to b|i}\, \bullet \cup vp_{b|i \to a|i}\, \bullet$$

$$\bullet\, t_{ab|k} = \bullet\, vt_{a|k \to b|k} \cup \bullet\, vt_{b|k \to a|k}$$

$$t_{ab|k}\, \bullet = vt_{a|k \to b|k}\, \bullet \cup vt_{b|k \to a|k}\, \bullet$$

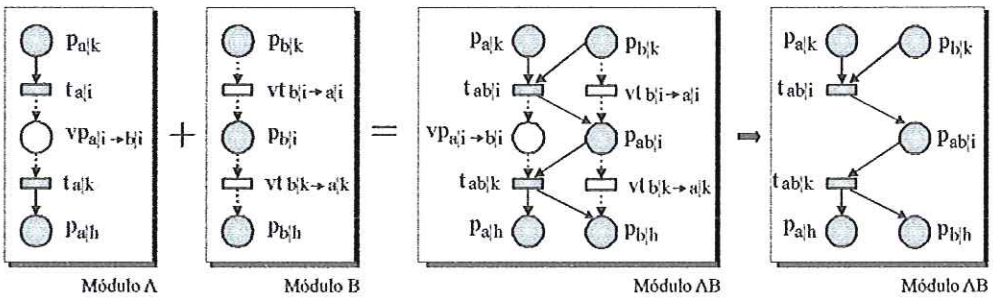

FIGURA 3.7. Exemplo 4 de associação de módulos

No Exemplo 4 (Fig. 3.7), o módulo A possui um lugar virtual que referencia um lugar em B e B possui duas transições que referenciam duas transições em A.

$$\bullet\, t_{ab|i} = \bullet\, t_{a|i} \cup \bullet\, vt_{b|i \to a|i}$$

$$t_{ab|i}\, \bullet = t_{a|i}\, \bullet \cup vt_{b|i \to a|i}\, \bullet$$

$$\bullet\, p_{ab|i} = \bullet\, vp_{a|i \to b|i} \cup \bullet\, p_{b|i}$$

$$p_{ab|i}\, \bullet = vp_{a|i \to b|i}\, \bullet \cup p_{b|i}\, \bullet$$

$$\bullet\, t_{ab|k} = \bullet\, t_{a|k} \cup \bullet\, vt_{b|k \to a|k}$$

$$t_{ab|k}\, \bullet = t_{a|k}\, \bullet \cup vt_{b|k \to a|k}\, \bullet$$

3.6. Exemplo de Aplicação

Este exemplo, extraído de JENG (1995), é uma estação de um sistema de manufatura flexível que consiste de três máquinas M_A, M_B e M_C, um robô R, e um *buffer* B, capaz de armazenar *k* peças, como mostrado na Fig. (3.8). O sistema produz um tipo de peça a partir da matéria-prima com duas operações de máquina: a primeira operação em M_A e a segunda em M_B ou M_C. A máquina M_A pode carregar matéria-prima e descarregar peça processada automaticamente e M_B e M_C podem descarregar peça pronta automaticamente.

As operações desse sistema são: M_A é carregada com matéria-prima, realiza a primeira operação e descarrega peça processada movendo-a para o B. O robô R pega a peça processada em B e carrega M_B ou M_C, a que estiver disponível para a segunda operação. Após a operação de M_B ou M_C, a peça pronta é descarregada para fora da estação.

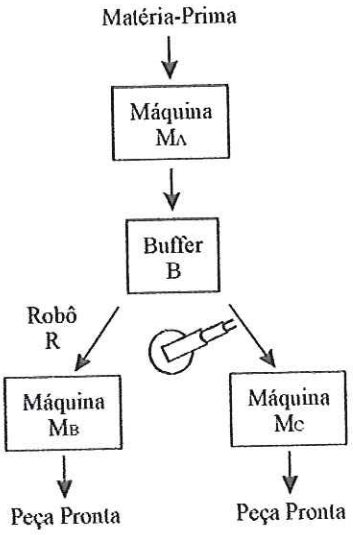

FIGURA 3.8. Sistema de Manufatura Flexível (JENG, 1995)

3.6.1. Modelagem utilizando Rede de Petri Virtual

Os passos a seguir são obtidos através do algoritmo apresentado na seção 3.2:

Passo 1: Primeiro identificam-se os subsistemas desse sistema, neste caso, composto por seus recursos físicos: máquinas M_A, M_B e M_C, o *buffer* B e o robô R.

Passo 2: Modela-se cada um dos sub-sistemas identificados no passo 1 utilizando Rede de Petri Virtual:

i) A máquina M_A é representada pela Fig. (3.9): o lugar virtual $vp_{MA|3}$ referencia o lugar $p_{B|1}$ do módulo B. É um ponto de ligação dos dois módulos e a transição virtual $vt_{MA|4}$ referencia a transição $t_{B|1}$ do módulo B.

ii) O Buffer B é representado pela Fig. (3.10): esse módulo não possui nós virtuais.

iii) O Robô R é representado pela Fig. (3.11): a transição virtual $t_{R|1}$ referencia a transição $t_{B|2}$ do módulo B, é a atividade de retirar peça processada de B.

iv) A máquina M_B é representada pela Fig. (3.12): possui uma transição virtual $vt_{MB|1}$ que referencia a transição do $t_{R|2}$ do módulo R, é a atividade da máquina M_B ser carregada pelo robô R com peça processada armazenada em B.

v) A máquina M_C é representada pela Fig. (3.13): possui uma transição virtual $vt_{MC|1}$ que referencia a transição do $t_{R|2}$ do módulo R, é a atividade da máquina M_C ser carregada pelo robô R com peça processada armazenada em B.

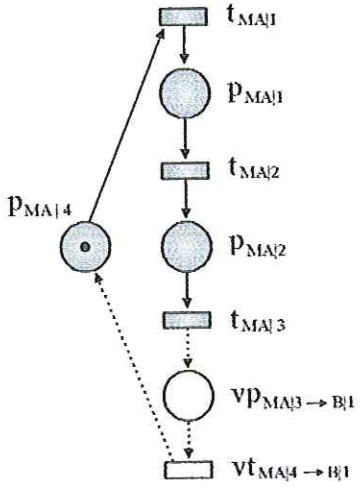

TABELA 2. Interpretação dos lugares para M_A.

	Descrição
$p_{MA|1}$	Matéria-prima carregada em M_A.
$p_{MA|2}$	Peça processada com a operação de M_A.
$p_{MA|3}$	Peça processada removida para B.
$p_{MA|4}$	M_A disponível.

FIGURA 3.9. Módulo M_A

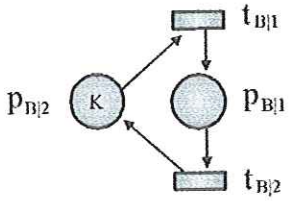

TABELA 3. Interpretação dos lugares para B.

	Descrição
$P_{B|1}$	Peça processada armazenada.
$P_{B|2}$	*Buffer* com *k* espaços disponíveis.

FIGURA 3.10. Módulo B

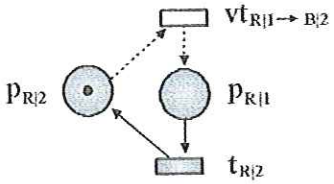

TABELA 4. Interpretação dos lugares para R.

	Descrição
$P_{R|1}$	Peça processada no manipulador do robô.
$P_{R|2}$	Robô disponível.

FIGURA 3.11. Módulo R

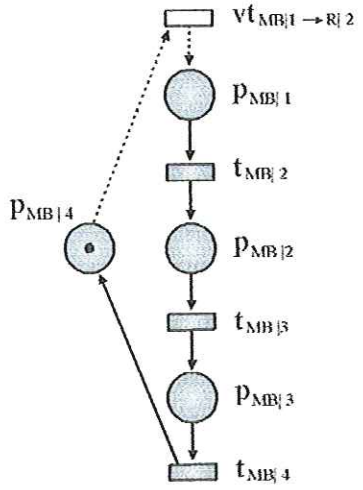

TABELA 5. Interpretação dos lugares para M_B

	Descrição
$p_{MB|1}$	Peça processada carregada em M_B.
$p_{MB|2}$	Peça pronta com a operação de M_B.
$p_{MB|3}$	Peça pronta removida para fora da estação.
$p_{MB|4}$	M_B disponível.

FIGURA 3.12. Módulo M_B

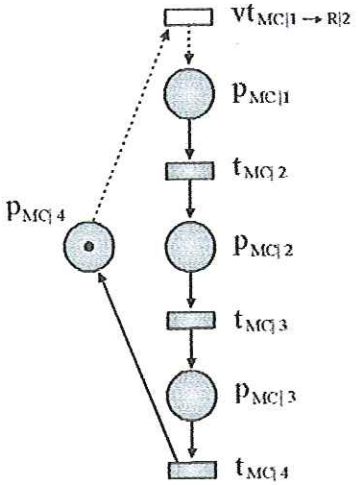

TABELA 6. Interpretação dos lugares para M_C

	Descrição
$p_{MC|1}$	Peça processada carregada em M_C.
$p_{MC|2}$	Peça pronta com a operação de M_C.
$p_{MC|3}$	Peça pronta removida para fora da estação.
$p_{MC|4}$	M_C disponível.

FIGURA 3.13. Módulo M_C

Passo 3: Consiste em fazer a junção dos módulos obtidos no passo 2 através de seus nós virtuais. Como a primeira operação do sistema começa com a máquina M_A, deve-se associar esse módulo (Fig. 3.9) ao que o seu nó virtual faz referência, no caso, ao módulo B (Fig. 3.10). Nesta primeira junção, não sobrou nenhum nó virtual, então repete-se a associação pelo módulo que referencia B, no caso o módulo R (Fig. 3.11), que possui uma transição virtual que referencia uma transição de B. Por fim, faz-se a junção dos módulos M_B e M_C (Fig. 3.12 e Fig. 3.13, respectivamente), que possuem transições virtuais que referenciam o módulo R. O resultado é mostrado na Fig. (3.14).

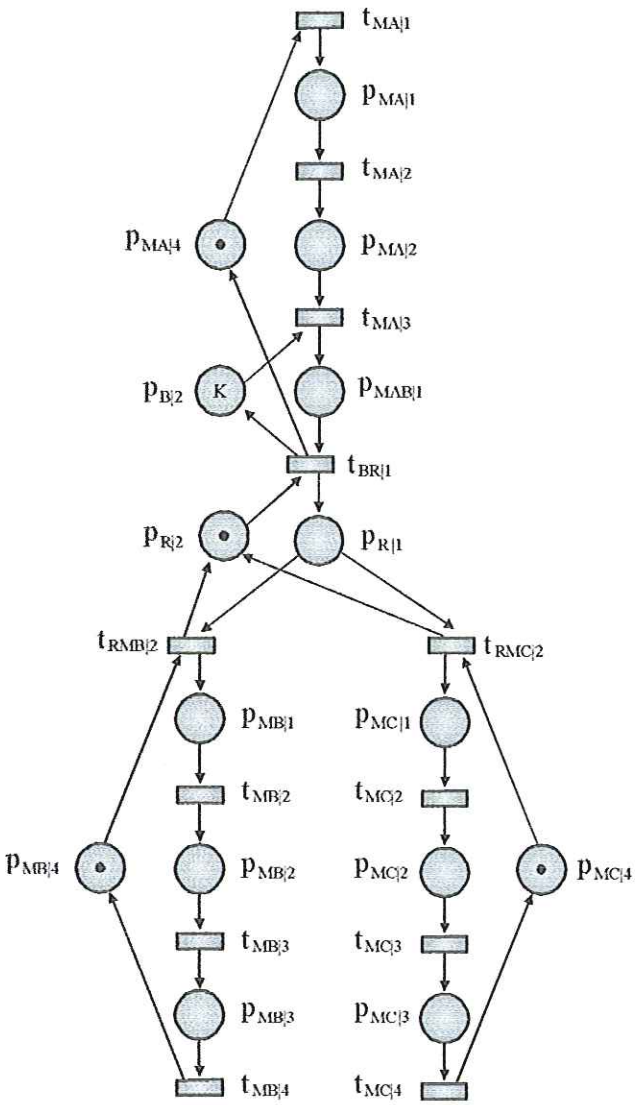

FIGURA 3.14. Rede de Petri final

4. Estudo de Caso

4.1. Considerações Iniciais

No capítulo anterior foi apresentado Rede de Petri Virtual, uma nova metodologia de projeto, modelagem, simulação e análise de Sistemas de Eventos Discretos. Esse capítulo apresenta passo a passo a modelagem por módulos de um Sistema Automatizado de Manufatura (SAM) utilizando Rede de Petri Virtual.

Uma vez que esse trabalho complementa o trabalho de MORANDIN JÚNIOR (1999), esse capítulo re-modela o estudo de caso desenvolvido por ele mostrando a diferenciação visual dos elementos reais e virtuais e a aplicação do algoritmo de junção desenvolvidos nesse trabalho.

4.2. Descrição do Sistema

O sistema utilizado nesse trabalho para exemplo de aplicação de Redes de Petri Virtuais foi apresentado em Inamasu (1995) e é um Sistema Flexível de Manufatura composto por nove estações, sendo que seis são estações de trabalho, uma estação de carga e descarga, uma estação de manutenção de AGV's e um sistema de transporte baseado em dois AGV's. O esquema desse sistema é apresentado na Figura 4.1.

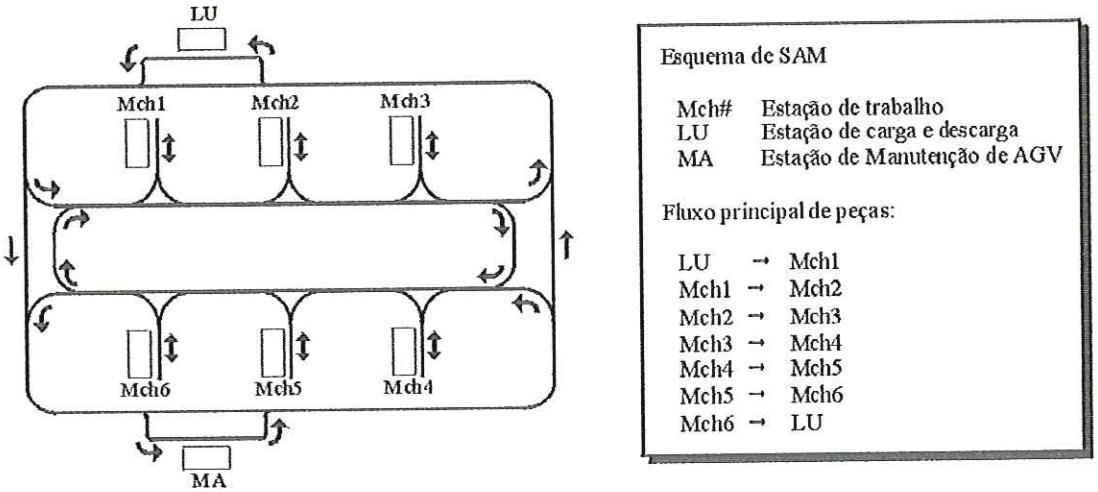

FIGURA. 4.1. Esquema do Sistema Automatizado de Manufatura (INAMASU, 1995)

As estações Mch's (Fig. 4.2) são compostas por máquinas CNC onde as peças serão efetivamente trabalhadas, armazenamento local de peça (buffer) e meio de carregamento e descarregamento de peça (manipulador) feita por AGV's. Nesse sistema deve-se trabalhar numa peça bruta, até a obtenção de um produto ou peça acabada, obedecendo-se a um plano de processos específico. A LU é o local de carga de peça bruta e descarga de peça acabada do sistema. E a MA é o local de manutenção dos AGV's.

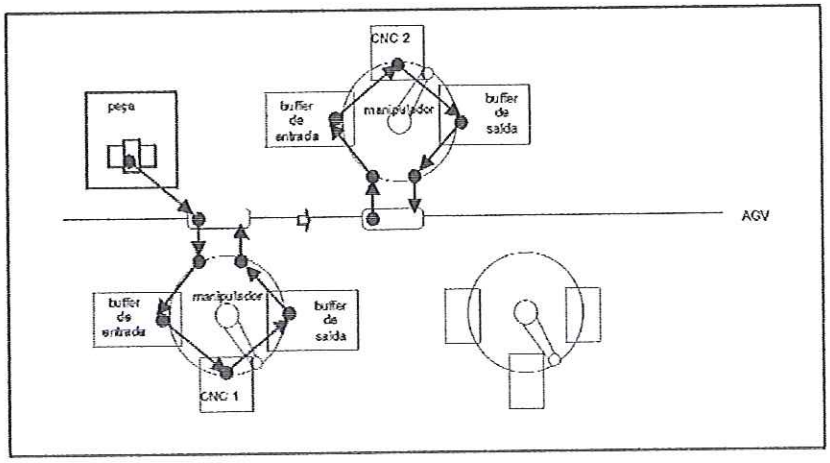

FIGURA 4.2. Exemplo de movimento de peça entre estação e AGV.

4.2.1. Processo de Fabricação

O plano de processo exemplo define uma seqüência de fabricação, para obtenção da peça acabada, que passa por três estações, partindo da estação de carga e finalizando na estação de descarga.

Portanto, para execução das tarefas, deve-se iniciar por uma estação A, depois por uma estação B e finalmente por uma C.

Uma restrição do exemplo aqui colocada é que as duas primeiras operações devem ser executadas nas estações Mch1 e Mch2 (Fig. 4.1) nessa seqüência, ou então nas estações Mch3 e Mch4, também na seqüência, e a terceira operação pode ser executada em qualquer uma das estações Mch5 ou Mch6.

Isso significa que, caso a primeira operação seja executada em Mch1, a segunda será necessariamente em Mch2. Caso a primeira operação seja executada em Mch3, a segunda será necessariamente em Mch4. A última operação é de livre opção entre Mch5 e Mch6.

4.3. Execução do Algoritmo para Utilização de PN Virtual

O algoritmo que permite modelar sistemas utilizando Rede de Petri Virtual foi explicado na seção 3.2.

4.3.1. Fase de Caracterização

Essa é a primeira fase para a modelagem de um sistema, que requer um estudo dos recursos do sistema, dividindo-o em subsistemas que formarão os módulos. Para o sistema em questão, identifica-se quatro módulos a seguir:

i) Módulo das Estações, denominado "Módulo E";

ii) Módulo dos AGV's, denominado "Módulo A";

iii) Módulo de Processo de Fabricação, denominado "Módulo P" e

iv) Módulo de Rotas, denominado "Módulo R".

4.3.2. Fase de Modelagem

Nessa fase modela-se cada um dos módulos identificados no passo 4.3.1 utilizando PN Virtual. Então, obtém-se quatro modelos conforme o número de módulos identificados no passo anterior.

4.3.2.1. Módulo das Estações

O modelo da Estação (Figura 4.3) possui dois lugares virtuais ($vp_{e|1\rightarrow a|9}$ e $vp_{e|8\rightarrow a|9}$) que representam o ponto de conexão entre a Estação e o sistema de AGV.

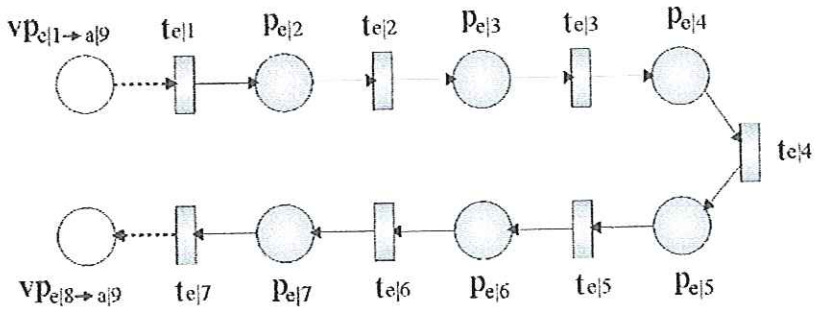

FIGURA 4.3. Módulo E

TABELA 7. Interpretação dos elementos do Módulo E

$t_{e\|1}$	Descarregar peça do AGV
$t_{e\|2}$ e $t_{e\|5}$	Carregar peça no buffer da estação
$t_{e\|3}$	Colocar peça no interior da estação
$t_{e\|4}$	Operação da estação
$t_{e\|6}$	Descarregar peça do buffer da estação
$t_{e\|7}$	Carregar peça no AGV
$P_{e\|3}$ e $p_{e\|6}$	Peça sobre o buffer da estação
$P_{e\|2}$ e $p_{e\|7}$	Peça na garra do manipulador
$P_{e\|4}$	Peça pronta para operação
$P_{e\|5}$	Peça pronta
$vp_{e\|1\rightarrow a\|9}$ e $vp_{e\|8\rightarrow a\|9}$	Peça sobre o AGV

4.3.2.2. Módulo dos AGV's

O sistema de AGV possui vários elementos virtuais que representam a operação de carga e descarga realizada pelo manipulador das estações, ponto de conexão entre os dois sistemas.

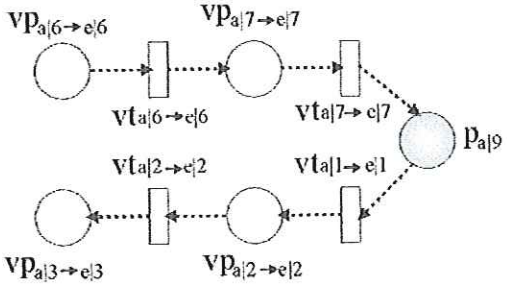

FIGURA 4.4. Módulo A

TABELA 8. Interpretação dos elementos do Módulo A

$p_{a\|9}$	Peça sobre o AGV
$vt_{a\|6\to e\|6}$	Descarregar peça do buffer da estação
$vt_{a\|7\to e\|7}$	Carregar peça no AGV
$vt_{a\|1\to e\|1}$	Descarregar peça do AGV
$vt_{a\|2\to e\|2}$	Carregar pela no buffer da estação
$Vp_{a\|2\to e2}$ e $vp_{a\|7\to e\|7}$	Peça transferida por manipulador
$Vp_{a\|3\to e3}$ e $vp_{a\|6\to e\|6}$	Estado do buffer da estação

4.3.2.3. Módulo de Processo de Fabricação

O módulo do plano de processos considera as possíveis alternativas de obtenção da seqüência de operações. Para isso, deve-se elaborar uma rede com lugares representando as estações Mch#, e conectá-las conforme a seqüência e a restrição comentada na seção 4.2.1.

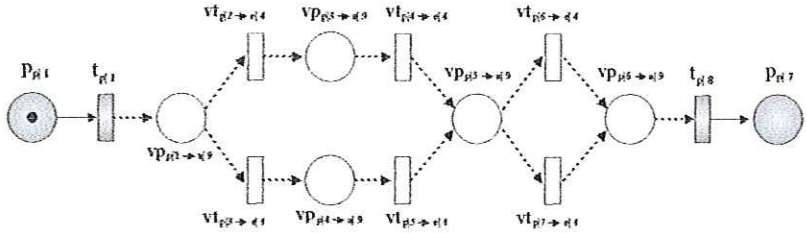

FIGURA 4.5. Módulo P

TABELA 9. Interpretação dos elementos do Módulo P

$t_{p'1}$	Carregar peça bruta no AGV
$t_{p'8}$	Descarregar peça acabada do AGV
$p_{p'1}$	Peça bruta em LU
$p_{p'7}$	Peça acabada em LU
$vt_{p'2 \to e'4}$	Operação A em Mch1
$vt_{p'3 \to e'4}$	Operação A em Mch3
$vt_{p'4 \to e'4}$	Operação B em Mch2
$vt_{p'5 \to e'4}$	Operação B em Mch4
$vt_{p'6 \to e'4}$	Operação C em Mch5
$vt_{p'7 \to e'4}$	Operação C em Mch6
$vp_{p'2 \to a9}$	Peça bruta sobre o AGV
$vp_{p'3 \to a'9}$	Peça sobre o AGV, após operação da Mch1
$vp_{p'4 \to a'9}$	Peça sobre o AGV, após operação da Mch3
$Vp_{p'5 \to a'9}$	Peça sobre o AGV, após operação B
$Vp_{p'6 \to a'9}$	Peça sobre o AGV, após operação C

4.3.2.4. Módulo de Rotas

O Módulo de Rotas segue o esquema da Fig. (4.1), cada lugar representa um trecho da rota e, uma marca num lugar indica a presença de um veículo no trecho. Para que um ou mais veículos não ocupem um único trecho num mesmo momento, os lugares são limitados a uma marca.

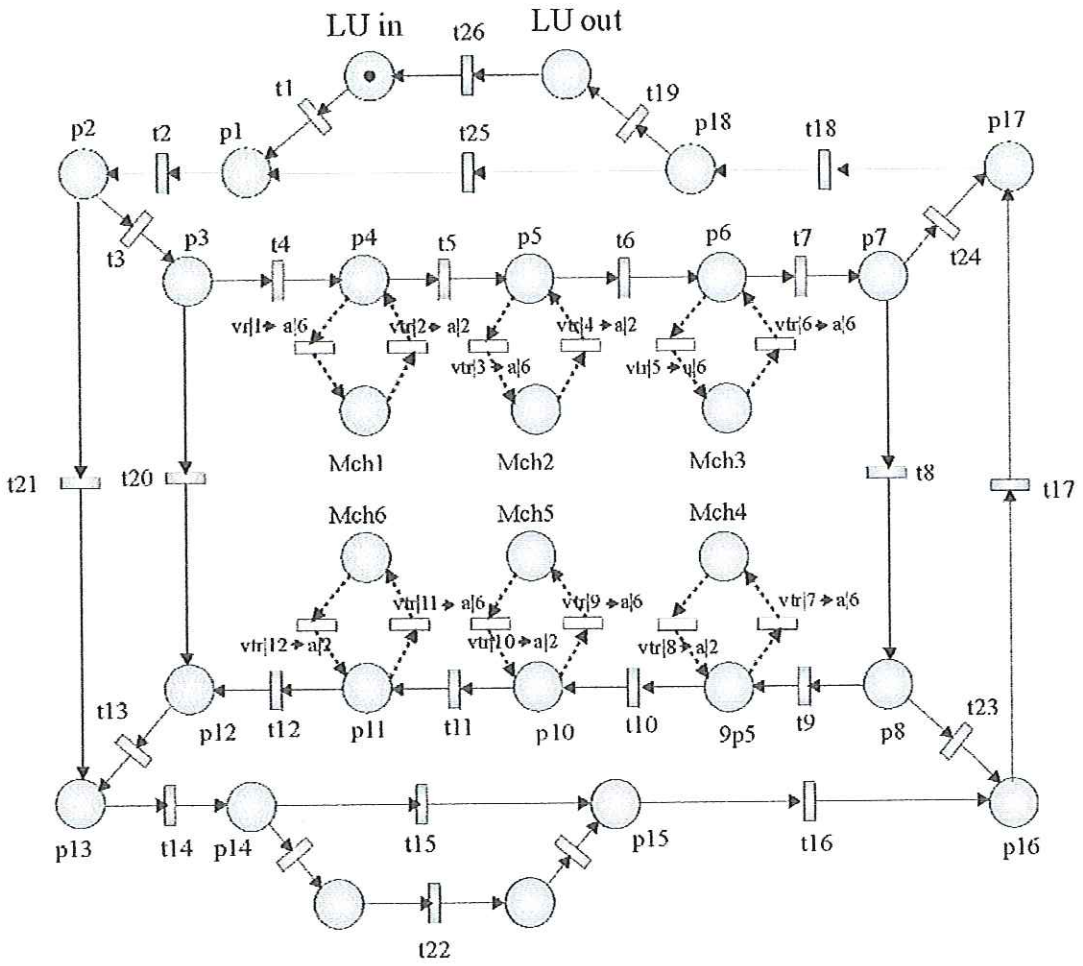

FIGURA 4.6. Módulo R

TABELA 10. Interpretação dos elementos do Módulo R

LUin	Local de carga da estação
LUout	Local de descarga da estação
Mch#	Local de carga e descarga da estação
p#	Trecho de trajetória
t#	Descolocamento do trecho
tc#0	Deslocamento para carga da estação
tc#1	Deslocamento para descarga da estação
$vt_{t\#\rightarrow a6}$	Carregamento de peça na estação
$vt_{t\#\rightarrow a2}$	Descarregamento de peça da estação

4.3.3. Fase de Associação

Nessa fase obtém-se o modelo final, que representará o sistema completo, aplicando o procedimento de junção (vide seção 3.5.1) sucessivas vezes até que não reste nenhum elemento virtual na rede. É importante lembrar do conceito de Classes e Objetos (seção 3.3) que, aplicado na obtenção do modelo final, identifica cada um dos recursos idênticos do sistema. Exemplo, o sistema desse estudo possui seis estações iguais, cada uma identificada com um número de 1 a 6. Módulos que possuem um único objeto, como é o caso de módulo de Rotas, pode ter a identificação do objeto ocultada.

4.3.3.1. Junção dos Módulos Estação e AGV

Escolhe-se o primeiro módulo e o segundo é obtido através da referencia dos nós virtuais do primeiro. O primeiro módulo é o "E" das Estações (Fig. 4.3) e o segundo, obtido através das referências é o módulo "A" dos AGV's (Fig. 4.4).

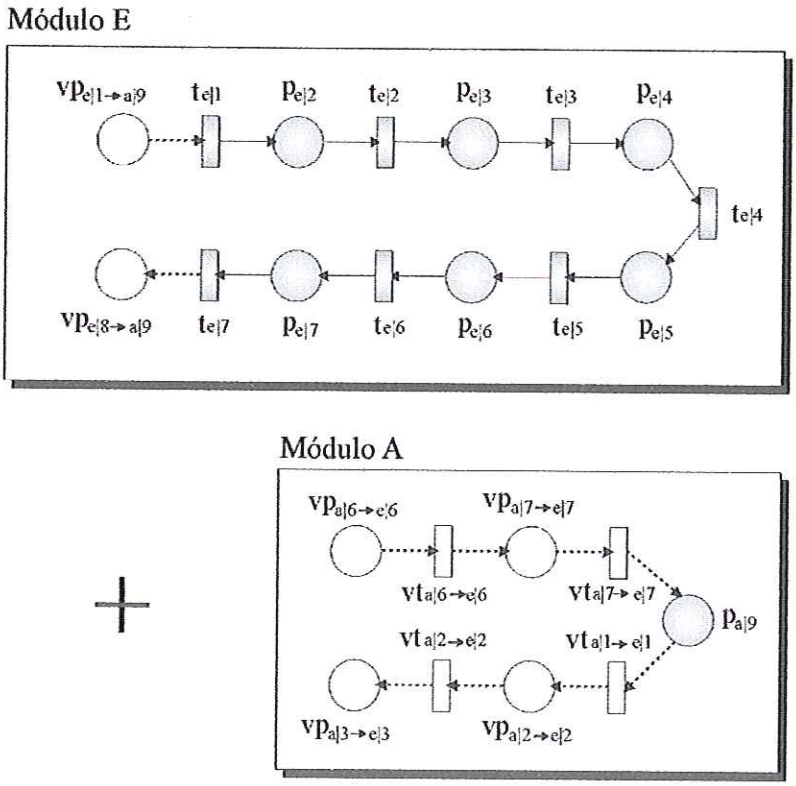

FIGURA 4.7. Módulos escolhidos no passo 3.1. do Algoritmo

O nó virtual $vp_{e|1 \rightarrow a|9}$ do Módulo E referencia um nó real do Módulo A $p_{a|9}$, nesse caso teremos um novo nó real $p_{ea|9}$ que receberá os conjuntos de entrada e de saída:

$$^{\bullet}p_{ea|9} = {}^{\bullet}vp_{e|1 \rightarrow a|9} \cup {}^{\bullet}p_{a|9}$$

$$p_{ea|9}{}^{\bullet} = vp_{e|1 \rightarrow a|9}{}^{\bullet} \cup p_{a|9}{}^{\bullet}$$

Módulo EA

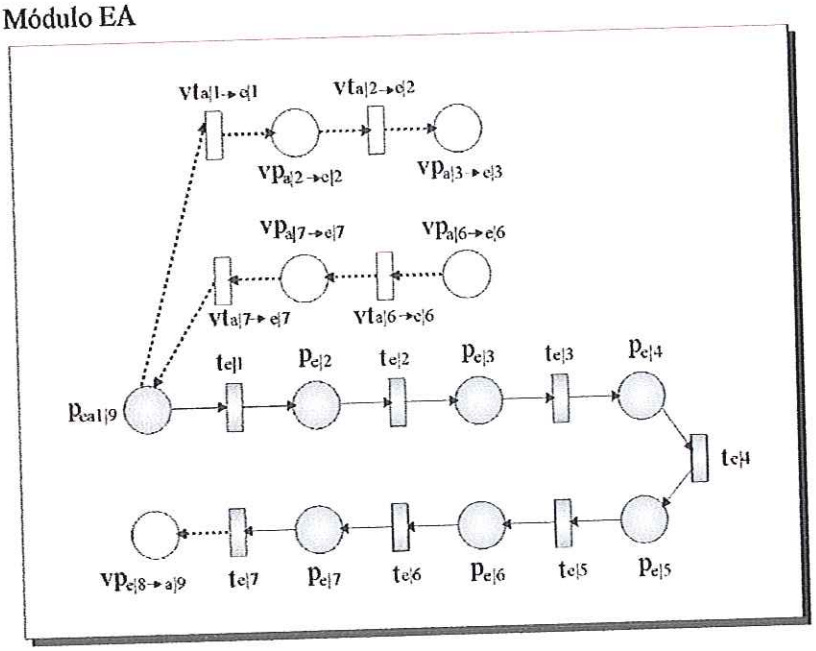

FIGURA 4.8. Resultado parcial da junção do Módulo A e Módulo E

Após a primeira junção dos modelos da Figura (4.7) ainda restaram elementos virtuais (Fig. 4.8). Nesse caso, deve-se repetir o procedimento de junção até que modelo seja constituído apenas por elementos reais. O resultados dessas repetições (*loops*) pode ser visto na Figura (4.9) para a junção dos Módulos de Estação e AGV, resultando no módulo EA. Note que os nós $p_{ea1|9}$ e $p_{ea2|9}$ foram instanciados.

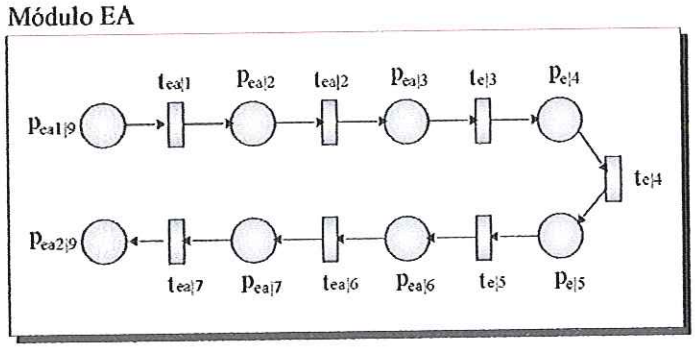

FIGURA 4.9. Módulo EA

4.3.3.2. Junção dos Módulos de Rotas e Estação-AGV

Restaram no sistema os módulos: Processo de Fabricação, Rotas e Estação – AGV. Na Figura (4.6) os nós $vt_{r\#\rightarrow a|2}$ e $vt_{r\#\rightarrow a|6}$ representam o carregamento e descarregamento da estação, respectivamente e Mch# a máquina CNC propriamente dita. Esses elementos são também identificados no módulo Estação-AGV pelos elementos tea|6 (descarregamento da estação), tea1 (carregamento da estação) e te|4 (máquina CNC). Realizando a junção desses módulos, obtém-se o resultado mostrado na Figura (4.10). Note que cada estação foi instanciada.

Módulo EAR

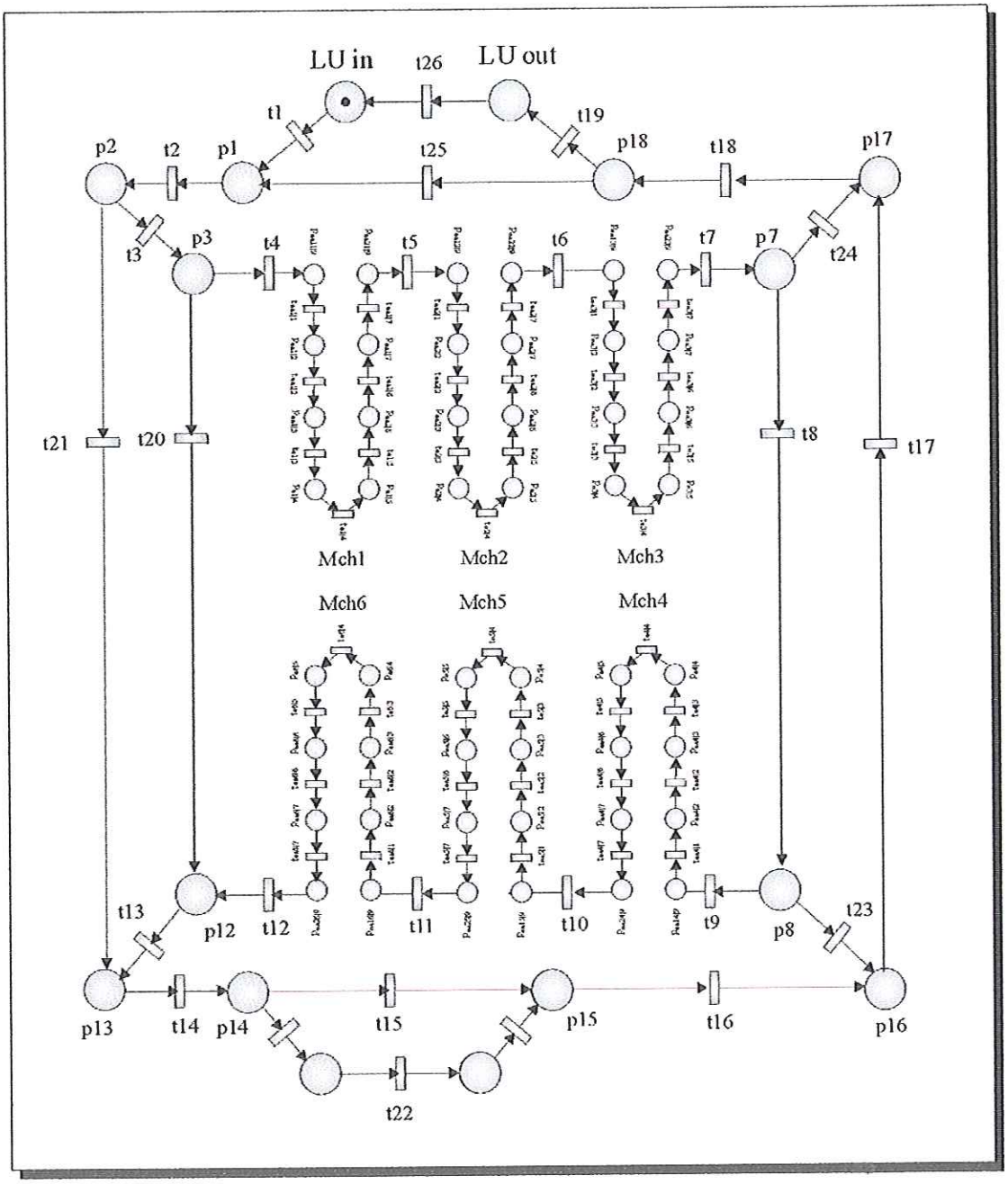

FIGURA 4.10. Módulo EAR

4.3.3.3. Modelo Final

Por fim, executa-se a junção dos módulos de Estação-AGV-Rotas e de Processos de Fabricação para obter o modelo final do sistema (Fig. 4.11). Nessa junção, são extraídas as rotas desconsideradas no processo de fabricação (seção 4.2.1) como, por exemplo, a trajetória dos AGV's até o estacionamento de manutenção MA. Conforme o modelo da Figura (4.5), ocorrem dois conflitos, o primeiro em $vp_{e|2\rightarrow a|9}$ e o segundo em $vp_{e|5\rightarrow a|9}$ e dois paralelismos em $vp_{e|3\rightarrow e|4}$ e $vp_{e|4\rightarrow e|4}$. Ambos os casos são modelados no modelo final.

Modelo Final

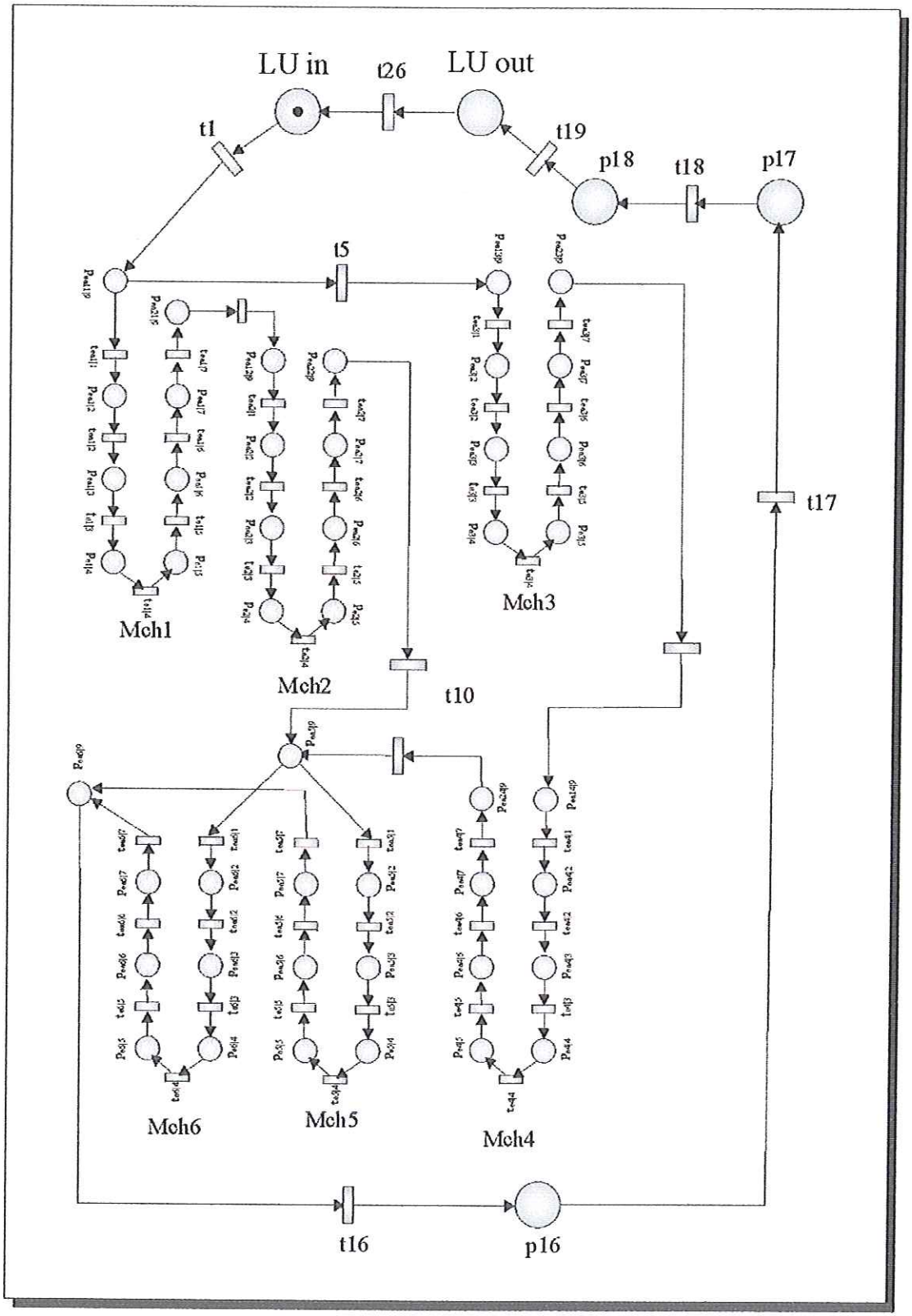

FIGURA 4.11. Modelo Final

5. Comentários Finais e Conclusões

Com o aumento da complexidade dos sistemas feitos pelos homens, os projetos de sistemas de eventos discretos necessitaram do apoio de ferramentas de modelagem, principalmente as gráficas e as de simulação. Tais ferramentas tem sido consideradas fundamentais, pois auxiliam na tomada de decisão. As ferramentas mais utilizadas são: diagrama de atividades, GRAI, SADT, IDEF, linguagens formais e autômatos e Rede de Petri.

Dentre essas ferramentas, a que fornece melhores resultados é Rede de Petri devido a sua formalidade matemática, sua representação visual do sistema e sua capacidade de simulação.

Entretanto, apesar de ser uma ferramenta poderosa no desenvolvimento de projetos de SED's, a modelagem utilizando Rede de Petri produz modelos grandes, complexos, com muito elementos gráficos, que torna a análise do sistema uma tarefa difícil.

Rede de Petri Virtual é uma nova extensão de PN que tem como objetivo minimizar os problemas citados na utilização de PN na modelagem de projetos de SED's. Baseia-se na modelagem por módulos, método *bottom-up*.

As principais contribuições desse trabalho são a diferenciação visual dos elementos reais e virtuais (seção 3.3), complementação na definição de VPN com o conceito de rede final (definição dois da seção 3.4), a especificação de um algoritmo de base para a modelagem (seção 3.2) e um procedimento de junção dos módulos através dos nós virtuais (seção 3.5.1).

É importante lembrar, que a rede final obtida através da junção dos módulos, sem elementos virtuais, é uma Rede de Petri Lugar/Transição. Isso torna possível empregar as técnicas de análise de PN para avaliar conflitos, vida, limitação, persistência, entre outras propriedades e técnicas de análise.

O procedimento de análise do modelo através das propriedades de PN (vide seções 2.2.5 e 2.2.6) não é citado, para não se perder o foco do trabalho e por ser amplamente discutido na literatura.

O primeiro passo é identificar os módulos (ou subsistemas) do sistema que geralmente são constituídos por recursos físicos como: sistema de transporte, máquinas, dispositivos de carga e descarga, entre outros.

Identificados os módulos, o passo seguinte é modelar cada um dos n módulos identificados no passo anterior, utilizando Rede de Petri Virtual. Os pontos de conexão entre os módulos são os nós virtuais e os demais elementos os nós reais.

O terceiro e último passo une os modelos em VPN obtidos no passo anterior para obter o modelo final, representativo do sistema completo. Essa união é feita baseando-se no procedimento de junção através dos nós virtuais.

O importante na modelagem por módulos é a flexibilidade que ela fornece. Possibilita alocar e desalocar recursos do sistema, modelar e simular antes que alterações bruscas sejam feitas no sistema real. Os módulos podem ser reaproveitados em sistemas semelhantes, diminuindo o tempo e custo no desenvolvimento de um novo projeto.

Analisando-se os resultados obtidos, pode-se dizer que essa abordagem permite uma construção do modelo de uma forma mais próxima do sistema real em relação a outras ferramentas de modelagem e extensões de PN. Não só pela forma de obtenção dos módulos, a partir dos recursos físicos do sistema, mas também pela possibilidade de re-uso desses módulos.

5.1. Trabalhos Futuros

As sugestões para continuidade do tema são:

i) Desenvolver template (ferramenta computacional) para VPN voltado à ferramenta computacional PNTools desenvolvida por SOARES (2001) – Laboratório de Simulação e Controle de Sistemas de Eventos Discretos, Engenharia Mecânica, EESC/USP.

ii) Desenvolver um nova extensão de PN, VPN Colorida, aplicada à Sistemas Automatizados de Manufatura. Assim, além da modelagem do sistema, seria possível representar o tráfego de informações através das cores.

iii) Estender o conceito de VPN para VPN Estocástica aplicada a análise da estimação de custos em Sistemas de Manufatura. Junção deste trabalho ao trabalho desenvolvido por SILVA (2002).

6. Referências Bibliográficas

ARMC. (2003). *Manufacturing Management* – GRAI. Advanced Manufacturing Research Centre (ARMC). http://www.amrc.co.uk/grai.htm (02 Out. 2003).

AUSFELDER C.; CASTELAIN E.; GENTINA J.C. (1994). A Method for Hierarchical Modeling of the Command of Flexible Manufacturing Systems. *IEEE Transactions on Systems, Man and Cybernetics*, v. 24, n. 4, p. 564-573, April.

AZZOPARDI D.; HOLDING D.J. (1997). *Petri Nets and OMT for Modeling and Analysis of DEDS*. Control Eng. Practice, v. 5, n. 10, p. 1407-1415.

CARDOSO, J.; VALETTE R. (1997). *Redes de Petri*. Florianópolis, UFSC.

CECIL, J.A; SRIHARI, K.; EMERSON, C.R. (1992). A Review of Petri-Net Applications in Manufacturing. *The International Journal of Advanced Manufacturing Technology*, v. 7, p. 168-177.

CUBERT, R.M.; FISHWICK, P.A. (1998). OOPM: An Object-Oriented Multimodeling and Simulation application framework. *Simulation*, v. 70, n. 6, p. 379-395, June.

FELDMANN, K.; SCHNUR, C.; COLOMBO. W. (1996). *Modularised, Distributed Real-Time Control of Flexible Production Cells, Using Petri Nets*. Control Eng. Practice, v. 4, n. 8, p. 1067-1078.

FERRAZ, P.; INAMASU, R.Y.; PORTO, A.J.V. Modelo de Implementação de um Template para Rede de Petri Virtual Aplicada a Sistemas de Manufatura Automatizados. (CD ROM). *XVI Congresso Nacional de Engenharia Mecânica*, v. 14, p. 567-574, Nov. 2001.

GEROGIANNIS V.C.; KAMEAS A.D.; PINTELAS P.E. (1998). *Comparative Study and Categorizations of High-Level Petri Nets.* The Journal of Systems and Software, n. 43, p. 133-160.

GUERRERO D.D.S.; PERKUSICH A.; FIGUEIREDO J.C.A. (1997). Modeling a Cooperative Environment Based on an Object-Based Modular Petri Net. *In Proceedings of the 9th International Conference on Software Engineering and Knowledge Engineering,* p. 240-247, Madrid, June.

GUSTIN, G.D.B (1999). *Aplicação de Redes de Petri Interpretadas na Modelagem de Sistemas de Elevadores em Edifícios Inteligentes.* São Paulo. 139p. Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo.

INAMASU, R.Y. (1995). *Modelo de FMS: Uma Plataforma Para Simulação e Controle.* São Carlos. 230p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

JENG M.D. (1995). *Modular Synthesis of Petri Nets for Modeling Flexible Manufacturing Systems.* The International Journal of Flexible Manufacturing Systems, n. 7, p. 287-310.

JENG M.D.; DICESARE F. (1992). *A modular Petri Net Synthesis Approach for Automated Manufacturing Systems.* Symposium on Flexible Automation, v.2, p.1163-1169.

KBSI. (2003). *A Structured Approach to Enterprise Modeling and Analysis.* Knowledge Based Systems, Inc. (KBSI). http://www.idef.com (02 Out. 2003).

MEDEIROS, A.D.A. et al. (1998). An Object-Oriented Petri Net modeling Tool and Abstraction Mechanisms for Cooperative Systems. *In Proc. of IEEE Int. Conf. on Systems Man and Cybernetics*, p. 172–177, San Diego, October.

MIYAGI, P.E. (1996). *Controle Programável: Fundamentos do Controle de Sistemas a Eventos Discretos*. São Paulo, Edgard Blücher.

MIYAGI, P.E.; FURUKAWA, C.M.; MARUYAMA, N.; TSUZUKI, F.S.G. Sistemas de Manufatura: Tendências em modelagem, análise e projeto. *X Congresso Brasileiro de Engenharia Mecânica*, p. 491-494, Rio de Janeiro, Dez. 1989.

MOORE, K.E.; GUPTA, S.M. (1999). *Petri Net Models of Flexible and Automatec Manufacturing Systems: A Survey*. International Journal of Production Research, v. 34, p. 3001-3035.

MORAES C.C.; CASTRUCCI P.L. Um Programa Didático em Automação Industrial. *XIV Congresso Brasileiro de Automática*, p. 1397-1402, Set. 2002.

MORANDIN JÚNIOR, O. (1999). *Metodologia de Modelagem de Sistemas Automatizados de Manufatura, Utilizando Rede de Petri Virtual*. São Carlos. 127p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

MORANDIN JÚNIOR, O.; KATO, E.R.R.; POLITANO, P.R.; CAMARGO, H.A.; PORTO A.J.V.; INAMASU. R.Y. (2000). A Modular Modeling Approach for Automated Manufacturing Manufacturing Systems Based on Shared Resources and Process Planning Using Petri Nets. *IEEE International Conference on Systems, Man and Cybernetics*, p. 3057-3062, October.

MURATA, T. (1989). Petri Net: Properties, Analysis and Application. *Proceedings of the IEEE*, v.77, n.4, p. 541-579, April.

PENG C.; CHEN F.F. (1996). Parallel Discrete Event Simulation of Manufacturing Systems: A Technology Survey. *Computers Ind. Eng*, v. 31, n. 1/2, p. 327-330.

PETERSON J.L. (1977). Petri Nets. *Computing Surveys*, v. 9, n. 3, p. 223-252, September.

PROTH, J.M.; SAVI, V.M. (1992). *A Modular Approach for Modeling Complex Manufacturing Systems.* International Conference on CAD/CAM, Robotics and Factories of the Future-Cars and FOF'92, v. 2, p. 1783-1791, Metz.1992.

RAMADGE. P.J.G.; WONHAM W.M. (1989). The Control of Discrete Event Systems. *Proceedings of the IEEE*, v. 77, n. 1, p. 81-98, January.

RIGHINI, G. (1993). Modular Petri Nets for Simulation of Flexible Production System. *International Journal of Production Research*, v. 31, n. 10, p. 2463-2477.

RILLO, M. (1987). Aplicações de Redes de Petri em Sistemas de Manufatura. *XX Congresso Nacional de Informática*, São Paulo, p. 503-510.

ROBERTS C.A.; DESSOUKY Y.M. (1998). An Overview of Object-Oriented Simulation. *Simulation*, v. 70, n. 6, p. 259-368, June.

SAKHAROV, A. (2003). *Formal Language in MathWorld.Wolfran.com.* http://mathworld.wolfram.com/FormalLanguage.html (02 Out. 2003).

SANTAREK, K. (1998). Organisational Problems and Issues of CIM Systems Design. *Journal of Materials Processing Technology*, v. 76, p. 219-226.

SEREY D.D.G. et al. (1997). An Object Based Petri Net Model: Application to Manufacturing Systems. *In Proc. of IEEE*, p. 2735-2740.

SILVA, A. R. Y. (2002). *Modelagem de Custos em Sistemas de Manufatura Utilizando Redes de Petri*. São Carlos. 95p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

SOARES, J. B; (2001). *Editor de Modelos de Sistemas de Eventos Discretos, baseado em Redes de Petri Interpretadas*. São Carlos. 56p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

SOARES, J.B.; INAMASU, R.Y.; SILVA A.R.Y.; FERRAZ P.; PORTO A.J.V. Petri Net Tools: A Graphical Open System Editor and Simulator. (CD ROM). *XVI Congresso Nacional de Engenharia Mecânica,* v. 15, p. 426-434, Nov. 2001.

WANG, L.C. (1996). Object-oriented Petri Net for Modeling and Analysis of Automated Manufacturing Systems. *Computer Integrated Manufacturing Systems*, v. 26, n. 2, p. 111-125.

WU W.; SU H.; CHU J. (2002) Supervisory Control of Discrete Event Systems Using Enabling Arc Petri Nets. *In Proc. of IEEE. Int. Conf. on Robotics and Automation*, p. 1913-1918, Washington, May.