

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO

**“Um método para Modelagem de Controle de
AMS usando Redes de Petri Virtuais”**

ALUNO: Rodrigo Guimarães Araújo
ORIENTADOR: Prof. Dr. Orides Morandin Junior

**São Carlos
Maio/2006**

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO

**“Um método para Modelagem de Controle de
AMS usando Redes de Petri Virtuais”**

RODRIGO GUIMARÃES ARAÚJO

Dissertação de Mestrado apresentada
ao Programa de Pós Graduação em
Ciência da Computação da
Universidade Federal de São Carlos,
como parte dos requisitos para a
obtenção do título de Mestre em
Ciência da Computação.

ORIENTADOR: Prof. Dr. Orides Morandin Junior

**São Carlos
Maio/2006**

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

A663mm

Araújo, Rodrigo Guimarães.

Um método para modelagem de controle de AMS usando redes de Petri virtuais / Rodrigo Guimarães Araújo. -- São Carlos : UFSCar, 2011.

91 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2006.

1. Ciência da computação. 2. Sistemas flexíveis de manufatura. 3. Redes de Petri virtuais. 4. Arquitetura de controle. 5. Inter-travamento. I. Título.

CDD: 004 (20^a)

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

***“Um Método para Modelagem de Controle
de AMS usando Redes de Petri Virtuais”***

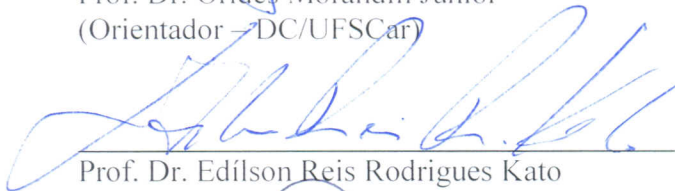
RODRIGO GUIMARÃES DE ARAÚJO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

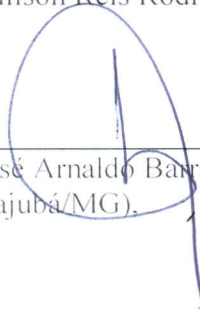
Membros da Banca:



Prof. Dr. Ordes Morandin Junior
(Orientador – DC/UFSCar)



Prof. Dr. Edilson Reis Rodrigues Kato
(UNIARA)



Prof. Dr. José Arnaldo Barra Montevechi
(UNIFEI/Itajubá/MG).

São Carlos
Maio/2006

DEDICATÓRIA

*Aos meus pais,
minha família,
minha namorada
e amigos.*

AGRADECIMENTOS

“Só existe uma coisa melhor do que fazer novos amigos: conservar os velhos.” —
Elmer G. Letterman

Aos meus pais, por sempre me darem apoio nas minhas mais diversas vontades sempre querendo o melhor para mim.

Ao Orides, que passou de orientador a amigo, por se mostrar sempre disponível a sanar minhas dúvidas e oferecer orientação de grande qualidade. Também por compreender minhas aspirações pessoais, mesmo as que não tinham relação direta com o mestrado e me dar apoio no intuito de realizá-las.

À minha namorada Letícia, que mesmo longe me deu apoio incondicional e compreendeu minhas dificuldades, além de me oferecer conforto durante todo o tempo.

Aos amigos de Campo Grande que, apesar da distância, sempre mantiveram contato e me ajudaram a lembrar dos bons e velhos tempos vividos antes da mudança de cidade.

Aos meus amigos Fernando, Néstor e Guilherme, com quem morei e pude compartilhar momentos de felicidade e de dificuldades. Com eles pude ensinar e aprender acerca de muitas coisas, porém a principal delas foi me ajudar a conhecer melhor a mim mesmo.

Ao amigo Delane, que mesmo morando em outra casa se mostrou um grande companheiro e teve grande participação em vários bons momentos vividos em São Carlos.

Ao amigo Carlos, que me acolheu em sua casa em São Paulo durante as várias vezes em que precisei ir até lá.

Ao amigo Eros, que mesmo não podendo estar muito presente, também se mostrou um grande companheiro, tanto nos momentos de estudo, quanto nos momentos de descontração.

Aos professores do Departamento de Computação da UFSCar, principalmente aos que tive oportunidade de conhecer durante o mestrado.

A todos os colegas de laboratório e de sala de aula.

Aos demais funcionários do DC.

RESUMO

ARAÚJO, RODRIGO GUIMARÃES, *Um método para Modelagem de Controle de AMS usando Redes de Petri Virtuais*. São Carlos, abril de 2006.

Um sistema automático de manufatura (AMS) é um sistema de produção onde um número discreto de matérias-primas é processado e montado por máquinas controladas e/ou robôs. Fazer o controle de um AMS é uma tarefa complexa pela existência de vários subsistemas e elementos que o compõem, e também pela necessidade de responder a questões estratégicas de produção e mercado. A complexidade de controle de AMS aumenta com as necessidades de produção e recursos tecnológicos.

Os trabalhos relativos a controle de AMS normalmente consideram características específicas do sistema. Alguns autores propõem soluções considerando a arquitetura, outros consideram abordagens por redes modulares, alguns consideram análise matemática; e assim, cada trabalho faz sua contribuição em um ponto mais específico.

Este trabalho busca considerar um conjunto mais abrangente de características inerentes ao problema de modelagem de controle de AMS. Nele são apresentadas várias abordagens inerentes às diversas características encontradas em um AMS. São mostradas diferentes opções de arquiteturas, formas diferentes de abordar hierarquicamente o problema, o uso de redes de Petri como forma de representar graficamente o problema conservando o formalismo matemático, simulação, e o uso de um *software* gráfico que contemple todas as opções anteriores.

A proposta do trabalho é indicar um método de modelagem de AMS que considere as características inerentes ao seu controle. Este método propõe oito etapas de elaboração do modelo. As etapas consideram os elementos em um nível de especificação mais genérico e que, gradualmente, vão sendo mais detalhados. Em algumas etapas são feitas simulações no intuito de validar o que já tenha sido modelado.

O trabalho chega ao fim com a aplicação do método de modelagem sobre um FMS hipotético. É considerado um planejamento da produção para cinco tipos de produtos em quantidades determinadas. A aplicação do método se mostra eficaz uma vez que, através da simulação do modelo, pode-se verificar que todos os produtos são fabricados conforme o plano de produção proposto e que eventuais *deadlocks* e conflitos podem ser evitados.

Palavras-chave: AMS, FMS, Redes de Petri Virtuais, arquiteturas de controle, intertravamento.

ABSTRACT

ARAÚJO, RODRIGO GUIMARÃES, A Modeling Method for AMS Control Using Virtual Petri Nets. São Carlos, april 2006.

An Automated Manufacturing System (AMS) is a production system where a discrete number of raws is processed and assembled by controlled machines and/or robots. Controlling AMS is a complex task, due to the several sub-systems that compose it and the need for attending strategic production issues and market. The AMS control complexity increases with the necessity of production and technologic resources.

The studies related to AMS control normally consider specific features of the system. Some authors propose solutions considering architecture, some others consider approaches by modular nets, others consider mathematic analysis; and so, each study has its contribution in a specific point.

This work considers a wider set of features related to the problem of modeling AMS control. It is presented different options of architecture, different ways to concern hierarchically about the problem, the use of Petri Nets as a way to represent graphically the problem conserving the mathematic formalisms, simulation, and the use of a graphic software that treat all the options before.

The proposal of this work is indicate a modeling method for AMS control that considers all features related to its control. This method proposes eight elaboration stages of the model. The stands consider the elements in a specification level more generic and, gradually, they are more detailed. In some stands, simulations are applied with the objective of validate what has been modeled.

The work ends with the application of the method to a hypothetic FMS. It is considered a production planning for five types of products in determined quantities. The method application is considered effective, since trough the simulation of the model, it can be verified that all the products are produced according to the production plan proposed and that eventual deadlocks and conflicts can be avoided.

Keywords: AMS, FMS, Virtual Petri Nets, control architectures, interlocking.

SUMÁRIO

<i>Introdução</i>	11
<i>Controle de Chão de Fábrica</i>	16
2.1 Problemas relacionados	16
2.2 Arquiteturas de controle	18
2.2.1 Controle centralizado	19
2.2.2 Controle hierárquico.....	20
2.2.3 Controle heterárquico	22
2.2.4 Controle híbrido	23
2.3 Paradigma holônico	25
2.4 Paradigma de Multiagentes	25
<i>Estado da Arte</i>	27
<i>Proposta</i>	37
4.1 Justificativa	37
4.1.1 Arquitetura híbrida	37
4.1.2 Simulação	38
4.1.3 Método <i>top-down</i>	39
4.1.4 Virtual Petri Nets (VPN)	40
4.1.5 CPN Tools	40
4.2 Proposta	41
<i>Verificação da Proposta</i>	51
5.1 Aplicação do método em um FMS	51
<i>Conclusão</i>	70
<i>Referências</i>	73
<i>Apêndice I – Rede de Petri Virtual</i>	81
<i>Apêndice II – Cpn Tools</i>	88

LISTA DE FIGURAS

<i>Figura 2.1 - Modelo de Controle Centralizado.....</i>	<i>19</i>
<i>Figura 2.2 - Modelo de Controle Hierárquico.....</i>	<i>20</i>
<i>Figura 2.3 - Modelo de Controle Heterárquico.....</i>	<i>22</i>
<i>Figura 2.4 - Modelo de Controle Híbrido.....</i>	<i>24</i>
<i>Figura 4.1 – Arquitetura de AMS considerado.....</i>	<i>38</i>
<i>Figura 4.2 – Exemplo de um FMS modelado após a primeira etapa.....</i>	<i>42</i>
<i>Figura 4.3 – Exemplo de um FMS modelado após a segunda etapa.....</i>	<i>45</i>
<i>Figura 4.4 – Exemplo de parte de um FMS modelado após a terceira etapa. No detalhe podem ser vistas as marcas referentes aos AGV's.....</i>	<i>46</i>
<i>Figura 4.5 – Exemplo de um módulo de FMS modelado após a quarta etapa. Este módulo refere-se à modelagem de uma máquina.....</i>	<i>47</i>
<i>Figura 4.6 – No detalhe, inscrições nos arcos a fim de definir caminhos.....</i>	<i>48</i>
<i>Figura 4.7 – Exemplo de um lugar que faz uma ligação com outro lugar, porém em módulo distinto.....</i>	<i>48</i>
<i>Figura 4.8 – Exemplo de um lugar que faz uma ligação com outro lugar, porém em módulo distinto.....</i>	<i>49</i>
<i>Figura 4.9 – Exemplo de uma transição com definição de tempo para a ação por ela representada.....</i>	<i>50</i>
<i>Figura 5.1 – Exemplo de uma transição com definição de tempo para a ação por ela representada.....</i>	<i>52</i>
<i>Figura 5.2a – Seqüência de produção do produto A.....</i>	<i>53</i>
<i>Figura 5.2b – Seqüência de produção do produto B.....</i>	<i>53</i>
<i>Figura 5.2c – Seqüência de produção do produto C.....</i>	<i>53</i>
<i>Figura 5.2d – Seqüência de produção do produto D.....</i>	<i>54</i>
<i>Figura 5.2e – Seqüência de produção do produto E.....</i>	<i>54</i>
<i>Figura 5.3 – Modelo após a primeira etapa.....</i>	<i>55</i>
<i>Figura 5.4 – Modelo após a segunda etapa. Estruturas simulando arcos inibidores foram acrescentadas.....</i>	<i>57</i>
<i>Figura 5.5 – Exemplo de parte de um FMS modelado após a terceira etapa. No detalhe podem ser vistas as marcas referentes aos AGV's.....</i>	<i>58</i>
<i>Figura 5.6 – Módulo formado pelo armazém e elementos que simulam o supervisorio.....</i>	<i>59</i>

<i>Figura 5.7 – Sub1</i>	61
<i>Figura. 5.8 – Módulo Sub2</i>	62
<i>Figura 5.9 – Módulo Sub3</i>	63
<i>Figura 5.10 – Módulo Sub4</i>	63
<i>Figura 5.11 – Módulo Sub5</i>	64
<i>Figura 5.12 – Módulo Sub6</i>	64
<i>Figura 5.13 – Módulo Sub1.1</i>	65
<i>Figura 5.14 – Módulo Sub2.1</i>	66
<i>Figura 5.15 – Módulo Sub3.1</i>	67
<i>Figura 5.16 – Módulo Sub3.1</i>	67
<i>Figura A1.1 - Representação de nós (lugares e transições) em uma PN Virtual. (Morandin, Jr., O, 1999)</i>	82
<i>Figura. A1.2 - Representação de módulos com elementos virtuais. (Morandin, Jr., O, 1999)</i>	82
<i>Figura A1.3 - Exemplos de associação entre módulos: i] um lugar virtual do módulo a, com alvo em um lugar (não virtual) do módulo b; ii] um lugar virtual e duas transições virtuais de um módulo a, com alvo em um lugar e duas transições de um módulo b; iii] um lugar virtual de um módulo a, com um alvo em um módulo b, e duas transições virtuais de um módulo b, com alvos em um módulo a. (Morandin, Jr., O, 1999)</i>	84
<i>Figura A1.4 - Exemplos de associação entre módulos: i] um lugar virtual do módulo a, com alvo em um lugar virtual do módulo b; ii] um lugar virtual e duas transições virtuais de um módulo a, com alvo em um lugar virtual e duas transições (não virtuais) de um módulo b; iii] um lugar virtual e duas transições virtuais de um módulo a, com alvo em um lugar virtual e duas transições virtuais de um módulo b; iv] duas transições virtuais e um lugar virtual de um módulo a, com alvo em duas transições virtuais e um lugar (não virtual) em um módulo b. (Morandin, Jr., O, 1999)</i>	86
<i>Figura A2.1 – Elementos básicos</i>	88
<i>Figura A2.2 – Variáveis, estruturas lógicas e tempo</i>	89
<i>Figura A2.3 – Composição de conjuntos, fusões de lugares e inscrições</i>	90

1

INTRODUÇÃO

Sistemas automáticos de manufatura (AMS – Automated Manufacturing Systems) têm sido projetados para suprir necessidades do mercado que demandam produtos que possuem baixo custo, qualidade, e pronta-entrega. A complexidade de controle de AMS aumenta com as necessidades de produção e recursos tecnológicos que não podem ser projetados usando-se seqüências de eventos discretos simples e/ou intertravamento básico. O AMS deve atender às necessidade do plano de produção, rotas de produção alternativas, operação de sistema e as disponibilidades de recursos, além do controle básico (KATO *et al*, 2001).

O problema de controle está relacionado à caracterização do sistema como sendo de evento discreto. O propósito de um controlador para estes sistemas é estabelecer e manter uma seqüência de eventos desejada. O sistema deve estar livre de *deadlocks* e o controlador não deve criar novas situações como essa. O controlador também deve satisfazer um conjunto de restrições (DESROCHERS *et al*, 1995).

O sistema de controle é responsável por manter consistentes os roteiros de produção, o compartilhamento dos recursos, tratar uma variedade de tipos de produtos e fazer com que os produtos cheguem ao final do processo sem problemas. Isso deve ser feito de maneira eficiente, de modo que o máximo de recursos possa ser utilizado ao mesmo tempo.

Redes de Petri (PN's) têm sido utilizadas como uma técnica promissora para modelagem de ambientes de manufatura automatizados. PN's é uma técnica de

modelagem matemática com representação gráfica desenvolvida por Petri em 1962 para modelar operações de sistemas de computação concorrentes. Desde então, têm sido estendidas e aplicadas a uma grande variedade de sistemas (MURATA, 1989). Hoje, existem softwares capazes de gerar modelos gráficos de PN's. Alguns oferecem a possibilidade de aplicar simulação sobre os modelos gerados e até fazer análises matemáticas.

Como uma ferramenta gráfica, PN's servem como uma técnica de modelagem visual e como auxílio à comunicação para descrição de modelos. Como uma ferramenta matemática, PN's podem ser usadas para montar equações de estado, equações algébricas, simulações e outros modelos matemáticos.

PN's são úteis para modelagem de sistemas automatizados de manufatura que podem ser caracterizados como concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e/ou estocásticos e cujas operações utilizam recursos compartilhados (DESROCHERS, 1995).

Existem algumas preocupações na modelagem de sistemas de controle de AMS. A escolha da arquitetura pode influenciar o desempenho do sistema de controle. Cada arquitetura possui vantagens e desvantagens. Cabe ao arquiteto escolher, dentre as possibilidades, aquela que melhor atende suas necessidades, tanto técnicas quanto econômicas.

Outra preocupação na hora da modelagem é a estratégia de seqüência em que se deseja fazer o modelo. São geralmente considerados dois métodos: *top-down* e *bottom-up*. O primeiro contempla a idéia de partir de um modelo macro, ou seja, mais genérico até chegar a um nível de detalhamento dos sinais digitais. Já no método *bottom-up* deve-se iniciar o trabalho de modelagem pelos elementos básicos, como os sinais digitais e depois seguir a modelagem pelos elementos que abrangem partes maiores do sistema. Esses métodos sugerem a visão do modelo através de níveis hierárquicos.

Alguns paradigmas podem ser usados no momento da implementação do sistema. Neste trabalho são apresentados dois paradigmas. O paradigma holônico apresenta uma maneira de tratar o sistema a partir de um conceito apresentado pela

primeira vez por Koestler (1969). Outro paradigma é o uso de sistemas de controle com atuação de multiagentes. Esse paradigma busca agilizar a tarefa de controle fazendo com que os elementos do sistema tenham autonomia nas tomadas de decisão.

Em geral a validação do sistema é feita de duas maneiras. Ou através de simulação, ou através de análise matemática. Existem vários *softwares* que estão habilitados para executar tal tarefa. A vantagem da simulação é sua proximidade com o mundo real, desde que o modelo de simulação seja devidamente elaborado. A análise matemática possui respaldo teórico e, por isso, é muito confiável. Contudo pode apresentar grande complexidade no momento da sua aplicação. Cabe ao arquiteto escolher qual dessas usar, ou mesmo mesclar as duas.

Os trabalhos relativos a controle de AMS normalmente consideram características específicas do sistema. Alguns autores propõem soluções considerando a arquitetura do sistema, outros consideram abordagens por redes modulares, alguns consideram análise matemática sem a preocupação com arquiteturas; e assim, cada trabalho faz sua contribuição em um ponto mais específico. Este trabalho busca considerar um conjunto mais abrangente de características inerentes ao problema de modelagem de controle de AMS.

Vários autores possuem estudos relativos aos temas apresentados anteriormente. Através de seus trabalhos, pôde-se verificar algumas vantagens e desvantagens de cada alternativa de modelagem. Dentre as várias arquiteturas existentes, a arquitetura híbrida foi considerada a que oferece mais vantagens. O método *top-down* foi escolhido por contemplar primeiramente o tratamento de elementos de nível mais alto. Como se deseja um modelo próximo do real, foi considerada a simulação quando da validação do modelo. Dentre outras vantagens, as PN's possuem grande respaldo acadêmico e facilitam a modelagem por ser uma ferramenta gráfica. Posteriormente serão apresentadas mais detalhadamente as justificativas das escolhas feitas, bem como, as descrições mais apuradas de suas características.

Neste trabalho é proposto um método que contempla todas essas alternativas de modelagem inerentes ao controle de AMS. O método é dividido em oito etapas. As etapas tem o papel de guiar o arquiteto de maneira que a modelagem seja feita de forma

metódica e o aprendizado do método possa ser didático. Após a execução das etapas espera-se obter um modelo livre de problemas e que represente o mais fielmente possível seu correspondente no chão de fábrica real.

Nos próximos capítulos serão mostradas características e particularidades dos assuntos envolvidos.

No capítulo 2 procura-se salientar as diferenças entre as arquiteturas de controle conhecidas. Cada arquitetura traz vantagens e desvantagens em relação ao problema de controle de AMS. Além de apresentar suas características, também é feito um balanço entre elas a fim de tentar descobrir qual é a mais adequada para ser usada na solução de problemas de AMS conforme suas particularidades.

Os paradigmas holônicos e de multiagentes também são descritos nesse capítulo. Serão detalhados seus conceitos e as descrições dos paradigmas.

O capítulo 3 é composto por referências a trabalhos ligados aos problemas relacionados ao controle de AMS. Alguns trabalhos tem foco na arquitetura utilizada, outros na forma de abordagem (modular, *top-down*, etc.) e outros estão mais distantes da solução proposta neste trabalho, mas norteiam o leitor para conhecer diferentes maneiras de tratar os problemas e em que ponto desse contexto se encontra este trabalho.

No capítulo 4, é apresentada detalhadamente a proposta do trabalho. Serão apresentadas as oito etapas que compõem o método proposto. As etapas podem ser fundidas ou alteradas caso o arquiteto sinta necessidade de fazê-lo. As etapas são um guia e podem ser adequadas a cada caso de modelagem.

O capítulo 5 trata da verificação da proposta. Tomado a partir de um Sistema Flexível de Manufatura (FMS) hipotético, o exemplo utilizado no capítulo foi desenvolvido tendo-se em mente as etapas do método. A descrição das características do FMS são apresentadas e então são mostradas as etapas. Cada etapa apresentada no capítulo 4 é adaptada para a modelagem do FMS. O capítulo termina com um modelo de FMS projetado para a produção de cinco tipos de produtos. O modelo final está livre

de problemas e está teoricamente pronto para ser implementado com a garantia de correta execução dos roteiros de fabricação considerados.

O capítulo 6 leva em consideração todos os capítulos anteriores a fim de analisar os vários fatores que compõem estratégias de abordagem de controle de AMS. Outro papel da conclusão é tentar medir o grau de eficiência do método proposto através de um balanço de facilidades e dificuldades encontradas durante o processo. Por fim, este capítulo tem por objetivo sugerir novos estudos relativos à proposta de maneira que esta possa ganhar mais consistência e abranger um maior número de aplicações práticas.

Este trabalho também possui dois apêndices.

A modelagem de AMS pode gerar alguns módulos separados. Diversos módulos podem fazer parte do modelo completo. Pela maneira que o método é proposto, geralmente serão produzidos vários módulos. As Redes de Petri Virtuais (VPN) são consideradas, então, como ferramenta de interligação entre os módulos gerados durante o processo de modelagem. Elas garantem o inter-travamento dos diversos módulos sem perda de características relativas ao modelo real. Suas definições estão presentes no fim do trabalho como um primeiro apêndice.

Como segundo apêndice estão algumas explicações sobre o *software* escolhido para fazer a representação gráfica dos modelos e também executar suas simulações. Como essa ferramenta possui definições próprias, decidiu-se acrescentar a descrição de seus principais elementos a fim de facilitar ao leitor o entendimento completo do trabalho, pois o entendimento de algumas particularidades do *software* serão necessárias para a assimilação das informações apresentadas nos próximos capítulos.

2

CONTROLE DE CHÃO DE FÁBRICA

Este capítulo trata de assuntos relativos ao controle de chão de fábrica. Primeiramente são apresentados alguns problemas inerentes ao controle e que devem ser levados em consideração sempre que se deseja implementar um modelo de controle em um AMS. Em seguida são apresentadas as principais arquiteturas de controle consideradas na literatura: arquiteturas centralizada, hierárquica, heterárquica e híbrida. Por fim, são descritos dois paradigmas que podem ser implantados sobre a arquitetura de controle.

O controle deve obedecer a um planejamento estratégico. Por isso, problemas de controle não são problemas localizados somente no baixo nível, mas também são problemas de planejamento que para serem resolvidos devem envolver vários níveis da fábrica. Não se deve pensar em controle sem se levar em consideração os problemas gerais de um AMS. No mais, devem ser considerados problemas de prevenção de *deadlock*, desempenho do sistema, tolerância a quebras de máquinas, supervisão, entre outros.

2.1 Problemas relacionados

A programação (*scheduling*) de FMS é um dos assuntos mais estudados relacionados a sistemas de produção modernos (O’KANE, 2000). Em sua forma mais simples a preocupação é com a alocação de trabalhos a recursos para satisfazer certas restrições tais como prazo de entrega, utilização de recursos ou custos (BAKER, 1974; GRAVES, 1981; GRUPTA, 1989 *apud* O’KANE, 2000).

Por sua complexidade intrínseca, programação é um assunto dominante que pode afetar na questão se as metas de produção serão alcançadas ou não. Em adição aos problemas relativos à programação, a condição de um sistema de produção pode ser alterada em qualquer instante. Conseqüentemente, mudanças dinâmicas na condição do sistema aumentam o já difícil problema de programação e lida com novas e inesperadas mudanças na tarefa de programação (O’KANE, 2000).

Levando-se em conta as citações anteriores é que se pode considerar a tarefa de programação reativa mais crítica e desafiadora, pois ao contrário da programação preditiva a programação reativa deve ter como principal preocupação o tempo necessário para que ela seja feita. São considerados fatores críticos toda situação em que há perda de capacidade de trabalho de algum recurso tais como quebra de máquina ou a falta de um trabalhador.

De nada adianta uma programação reativa ótima se esta consumir um tempo inaceitável para a estratégia de produção da fábrica, ou seja, a reprogramação deve valer a pena em relação ao tempo e, logo economicamente. Por isso pode-se notar a utilização de técnicas heurísticas na busca de uma solução que atenda a essa restrição de tempo. Essas técnicas podem não chegar a um resultado ótimo, mas apresentam uma boa solução em um período aceitável de tempo.

Problemas diretamente ligados ao controle também apresentam enormes desafios. O controle lida com prevenção de *deadlocks*, sincronismo de atividades paralelas, coordenação de peças a serem processadas e captação de dados para sistemas de supervisão. O principal propósito do controle é estabelecer e manter a seqüência desejada dos eventos.

O controle também deve satisfazer um conjunto de restrições. Essas restrições podem estar relacionadas à capacidade finita de manufatura, taxas de produção, falhas e taxas de reparo, disponibilidade de matéria prima, precedência de trabalhos, pedidos do cliente, bem como outros parâmetros.

Cada elemento do AMS necessita de um controle apurado para funcionar de maneira adequada, principalmente quando se leva em consideração o sistema geral que controla a interação dos diversos elementos dentro do mesmo ambiente e em coordenação de tarefas.

Os principais elementos físicos que permitem ações de controle são os diversos sensores, atuadores, manipuladores, motores e controladores. Os controladores são responsáveis por obter, armazenar e intervir no estado corrente da fábrica.

O estado corrente é adquirido através dos sinais enviados pelos sensores ao controlador, sendo que outros controladores também podem enviar sinais. Baseados nesse estado, os controladores devem disparar as ações nos atuadores, manipuladores e motores a até mesmo em outros controladores. Assim, pode-se dizer que o controlador é responsável pelo inter-travamento da fábrica conforme condições restritivas dos equipamentos e operadores.

Através de toda essa troca de dados o supervisor¹ deve ser capaz de garantir que as tarefas sejam feitas de forma correta.

2.2 Arquiteturas de controle

No controle de chão de fábrica uma arquitetura de controle deve fornecer uma boa representação do sistema descrevendo de forma completa e não-ambígua a sua estrutura e os relacionamentos existentes entre as entradas (*inputs*) e as saídas (*outputs*). Essa representação é importante para o projeto e construção de um Sistema de Controle de Chão de Fábrica. A partir da arquitetura, que descreve os componentes de controle em termos de tarefas e interações, é possível estabelecer a funcionalidade do sistema antes da sua implementação (Bie, 1989 *apud* Friedrich, L. F., 1996).

São características desejáveis em um Sistema de Controle de Chão de Fábrica (DIL, 1991 *apud* FRIEDRICH, 1996):

- Facilidade de modificação / extensibilidade;

¹ O supervisor é um elemento eletrônico responsável por gerenciar o planejamento da produção.

- Facilidade de reconfiguração / adaptabilidade;
- Confiabilidade / tolerância a falhas.

O Sistema de Controle de Chão de Fábrica pode ser projetado de acordo com algumas arquiteturas existentes. As arquiteturas mais difundidas são: a centralizada, a hierárquica, a heterárquica e a híbrida, e serão apresentadas detalhadamente a seguir.

2.2.1 Controle centralizado

Na arquitetura de controle centralizado um único controlador fica responsável por processar as informações recebidas dos controladores específicos de cada tarefa e devolver as decisões tomadas aos mesmos para que a produção continue. O controlador central, então, deve ser mais robusto e deve possuir maior poder de processamento que os demais controladores (Figura 2.1).

Razões apontadas para a utilização de controle centralizado em centros de trabalho são: a vantagem do armazenamento central, a disponibilidade de dados globais e o uso de controladores de máquina sem necessidade de capacidade de decisão (Dil, 1991 *apud* Friedrich, L. F., 1996). Entretanto, a utilização da estrutura centralizada resulta em sistemas com baixa tolerância a falhas e com falta de características como facilidade de modificação e extensão do *software* de controle.

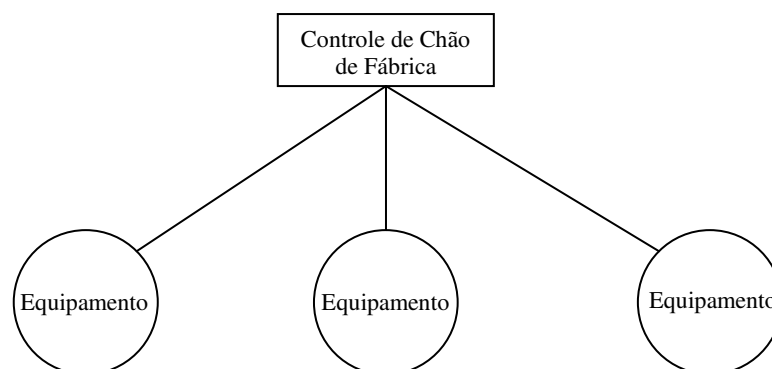


Figura 2.1 - Modelo de Controle Centralizado.

Tradicionalmente, um controle centralizado depende do desempenho do computador central (DESHMUKH, *et al.* 1995 *apud* MARTINS, 2005).

2.2.2 Controle hierárquico

A arquitetura de controle hierárquico baseia-se na idéia de níveis de controle na forma piramidal. Utiliza o conceito mestre-escravo na troca de informações entre controladores de níveis distintos. Cada nível da hierarquia tem um escopo de aplicação bem definido dentro do sistema, com uma solução de composição de elementos de arranjo físico e de controle. Neste tipo de controle é usada a filosofia de distribuição de tomada de decisão, onde cada nível toma suas decisões de acordo com comandos do nível superior e resultados do nível inferior (Figura 2.2).

A arquitetura hierárquica oferece vários benefícios em relação aos outros tipos de arquitetura. A estrutura rígida do relacionamento mestre-escravo apresentada pela arquitetura hierárquica resulta em tempos de resposta mais rápidos, tanto na célula quanto nos outros níveis (DIL, 1991 *apud* FRIEDRICH, 1996). O mapeamento da hierarquia de controle lógico para a arquitetura física é facilmente conseguido na arquitetura de controle hierárquico (RAN, 1988 *apud* FRIEDRICH, 1996).

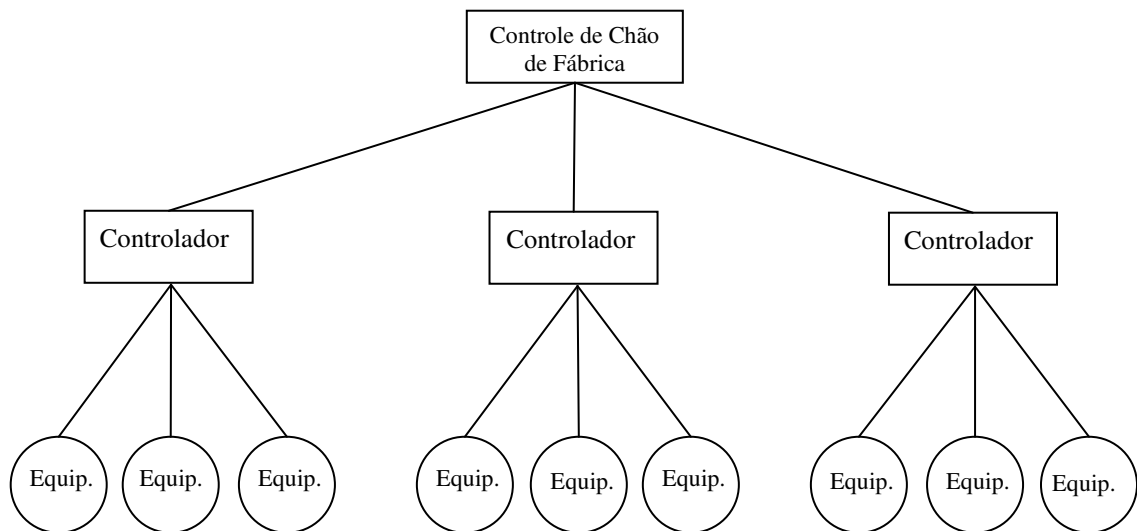


Figura 1.2 - Modelo de Controle Hierárquico.

Entretanto, algumas desvantagens também são apontadas. Uma delas se refere à rigidez que a estrutura hierárquica tende a impor desde a fase de projeto, fazendo com que qualquer mudança posterior seja de difícil implementação (DUF, 1988 *apud*

FRIEDRICH, 1996). Outra desvantagem está no fato de que a falha de um nível leva à paralisação dos controladores dos níveis inferiores.

Quando considerado todo o controle de um chão de fábrica, a estrutura de controle hierárquico tem sido vastamente implementada pelas indústrias. Um modelo bastante conhecido é o *National Institute of Standard and Technology / Advanced Manufacturing Research Facility (NIST / AMRF)* (FRIEDRICH, 1996). Ele é composto por cinco níveis: corporação, fábrica, célula, estação de trabalho e equipamento.

Cada nível possui funções específicas: planejamento do processo, gerenciamento da produção, e gerenciamento de informações estão no nível de corporação. O nível de fábrica é responsável por coordenar tarefas de produção incluindo alocação de recursos e atribuição de tarefas. O nível de célula deve controlar a seqüência de produção de peças similares em lotes de produção e apoiar tarefas tais como manipulação de materiais e transporte. É papel do nível de estação de trabalho gerar a programação (*scheduling*) da produção para as mesmas. No **AMRF**, uma estação de trabalho geralmente consiste de um robô, uma estação de processamento, um *buffer* de armazenamento e um controlador. Finalmente, o nível de equipamento é responsável por monitorar a execução das tarefas de produção.

Outro modelo hierárquico é o *Advanced Factory Management and Control Systems (AFMCS)*, criado pela *Computer Aided Manufacturing-International (CAM-I)*. Este modelo tem foco na tomada de decisões em sistemas de manufatura. Há quatro níveis de hierarquia neste modelo de controle: nível de fábrica, nível de célula, nível de estação de trabalho e nível de equipamento.

Segundo Ou-Yang (1997), alguns pesquisadores têm procurado trabalhar com o paradigma de orientação a objetos (OO) no controle de sistemas de manufatura hierárquicos. Uma solução baseada em três técnicas de modelagem (IDEF0, redes de Petri e diagramas de entidade e relacionamento) está sendo desenvolvida no controle de uma célula CNC.

O projeto **ESPRIT** (*European Strategic Program for Research and Development in Information Technology*) apresenta uma hierarquia de controle que é derivada da hierarquia **AMRF** onde, em cada nível as decisões são responsabilidade de um controlador (MEY, 1986 *apud* FRIEDRICH, 1996). Cada controlador é composto por sub-controladores ou unidades de tomada de decisão que têm a mesma estrutura interna e inclui um sistema especialista para o planejamento, interpretação e diagnóstico do nível controlado. O sistema especialista de cada nível resolve tarefas específicas de controle. Por exemplo, o controlador de centro de trabalho gera um plano diário e modifica este plano de acordo com o estado corrente da fábrica.

2.2.3 Controle heterárquico

O controle heterárquico não utiliza o conceito **mestre-escravo**. O controle heterárquico trabalha com controladores independentes, mas que cooperam entre si através da troca de informações na forma de mensagens. Na arquitetura heterárquica de controle, todos os subsistemas participantes têm: direitos iguais no acesso a recursos, acesso a todos os componentes, modo de operação independente, e funcionamento de acordo com as regras de protocolo do sistema (Figura. 2.3).

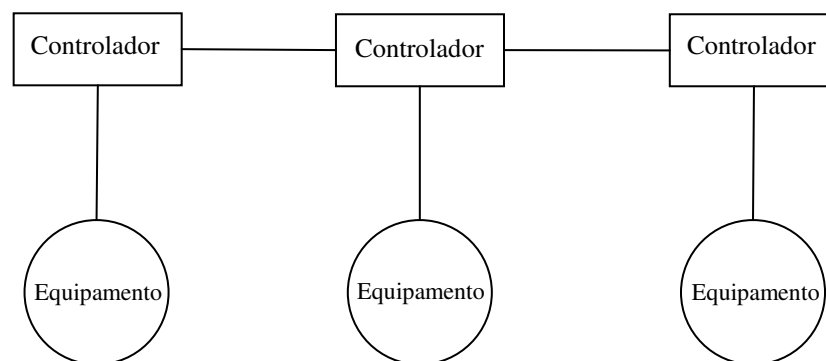


Figura 2.3 - Modelo de Controle Heterárquico.

Outra característica desta arquitetura é a eliminação ou minimização de variáveis globais com o objetivo de melhorar características, como: modularidade, facilidade de modificação e extensibilidade. Isso implica na existência de uma quantidade

significativa de informações detalhadas em cada controlador de dispositivo (equipamento).

A estrutura de controle hierárquico proporciona a possibilidade de tratamento de dados globais (ou semi-globais) e respostas mais rápidas, mas estabelece uma rigidez ao sistema que dificulta os aspectos referentes a modificação e independência dos componentes do sistema e também apresenta uma baixa tolerância a falhas.

Por outro lado, a estrutura heterárquica proporciona uma grande flexibilidade no que diz respeito à modificação do sistema e independência dos componentes, mas requer um grande volume de comunicação entre os componentes e também exige que cada componente tenha seus próprios dados localmente.

2.2.4 Controle híbrido

O projeto **ESPRIT**, citado anteriormente, possui um grupo dedicado especificamente ao estudo de sistemas de características híbridas. No **ESPRIT** (www.control.lth.se/H2C/, consultado em 15/03/2006), ainda não há uma padronização que contemple a modelagem de um sistema híbrido. Existem até algumas definições iniciais de hibrididade que fazem entender como um sistema híbrido, um sistema com eventos discretos e contínuos. Essas definições evoluem para o tratamento de arquiteturas de controle com características de hierarquia e heterárquia (Figura. 2.4).

Existem várias razões para se usar modelos híbridos a fim de representar o comportamento dinâmico de um sistema. Reduzir complexidade foi e ainda é uma razão importante para lidar com sistemas híbridos. Isso é atendido em sistemas híbridos pela incorporação de modelos de processos dinâmicos em diferentes níveis de abstração. (ANTSAKLIS; NERODE, 1998).

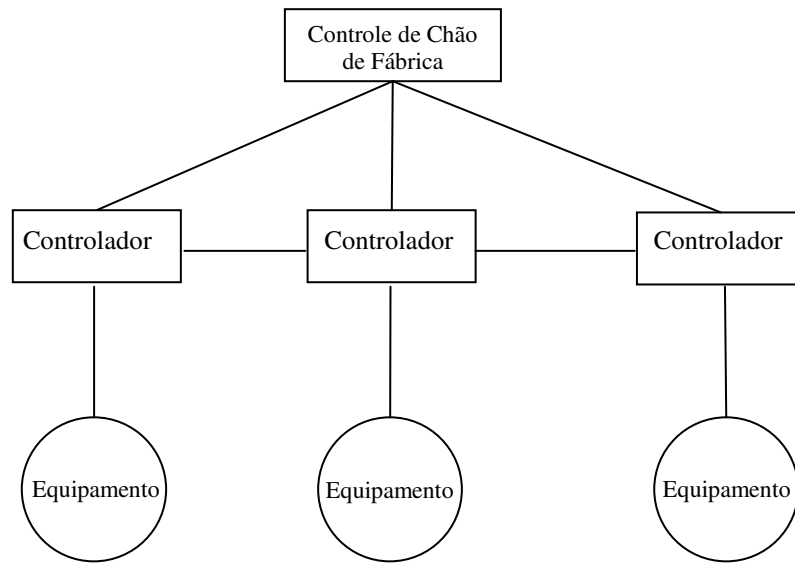


Figura 2.4 - Modelo de Controle Híbrido.

Esforços recentes na pesquisa de sistemas híbridos concentram-se tipicamente na análise de comportamentos dinâmicos e busca desenvolver controladores com garantia de estabilidade e desempenho. O advento de máquinas digitais fez os sistemas híbridos ficarem muito comuns. Uma vez que um sistema digital interage com o mundo contínuo, seu comportamento envolve o fenômeno de hibrididade, que necessita ser analisado e entendido. (ANTSAKLIS; NERODE, 1998).

Conforme MARTINS (2005), o que caracteriza uma arquitetura híbrida são: múltiplos computadores (heterogeneidade), fraco relacionamento mestre-escravo, supervisor dá início às tarefas, demais níveis terminam de executar as tarefas sem dependência direta do nível superior, tomada de decisão distribuída, banco de dados local e agregado a cada nível, combinação de reação a distúrbio com um alto e previsível nível de desempenho.

Ainda conforme MARTINS (2005), as vantagens da arquitetura híbrida são: implementação gradual, tempo de resposta rápido, comportamento adaptativo, alta flexibilidade, facilidade de reconfiguração e adaptabilidade, otimização global e resposta em tempo real.

Suas desvantagens são: desempenho computacional limitado a cada nível,

limitações técnicas dos controladores, dificuldade no tratamento de controle adaptativo, dificuldade para manter o sistema coerente.

2.3 Paradigma holônico

Uma paradigma mais recente, usado na modelagem de sistemas de controle de manufatura, é o holônico. Ele tenta mesclar as melhores características dos sistemas hierárquicos e sistemas heterárquicos, um alto e preditivo desempenho com alta robustez contra distúrbios. Com um paradigma holônico, o mapeamento de funções de controle das unidades permanece flexível, tanto que o poder de decisão pode ser distribuído entre estas unidades se necessário.

Sistemas holônicos de manufatura (HMS) vislumbram a possibilidade de se desenvolver sistemas de controle de manufatura com autonomia e capacidades inteligentes, ágeis e com rápida adaptação às mudanças do ambiente, preparados para lidar eficientemente com as ocorrências de distúrbios, e possibilitando integração fácil entre recursos de manufatura e sistemas legados (LEITÃO; RESTIVO, 2005).

HMS é um paradigma que traduz para o mundo da manufatura os conceitos desenvolvidos por Koestler (1969) para organismos vivos e organizações sociais. Manufatura holônica é caracterizada por holarquias de entidades autônomas e cooperativas chamadas holons, que representam toda a amplitude de entidades de manufatura. Um holon, como descrito por Koestler, é uma parte identificável de um sistema de manufatura que possui uma identidade única, e mais, é feita de partes subordinadas e em contra-partida é parte de um todo maior. A introdução do paradigma holônico permite uma nova abordagem para os problemas de manufatura, trazendo as vantagens de modularidade, autonomia descentralizada e escalabilidade. (LEITÃO; RESTIVO, 2006).

2.4 Paradigma de Multiagentes

Através do acervo bibliográfico encontrado hoje pode-se verificar o crescimento nos estudos de paradigmas de sistemas multiagentes (AMS's) aplicados a planejamento

e controle. “Agente” refere-se a um software associado tanto com componentes de hardware quanto funções específicas de um sistema de manufatura (MAIONE; NASO, 2003). Ele possui inteligência suficientemente capaz de ações de controle autônomo neste ambiente e manter relações de cooperação participando de acordos de associação com outras entidades a fim de alcançar seus objetivos de projeto (MATOS; SIERRA; JENNINGS, 1998).

Um agente deve ser capaz de agir sem intervenção humana direta ou de outros agentes e deve ter controle sobre suas próprias ações e estado interno. Um controle de sistema baseado em multiagentes significa um sistema em que a principal abstração usada para controlar componentes, como controlador, escalonador, é um agente. (SCHOOP; NEUBERT; COLOMBO, 2001).

Como citado anteriormente, sistemas heterárquicos de manufatura asseguram escalabilidade, reconfigurabilidade, e visa, eventualmente, expansibilidade de hardwares de manufatura, e controle reativo e inteligente de software. A idéia de aplicar multiagentes em arquiteturas heterárquicas é bastante notada entre os pesquisadores.

Um sistema multiagentes funciona como uma equipe, onde todos os elementos trabalham para um mesmo objetivo. Todos os elementos têm as mesmas informações relativas ao sistema e possuem o mesmo direito quando da tomada de alguma decisão. Porém, cada um deles deve saber qual o seu papel, de maneira que no momento em que surge um problema, o elemento mais indicado seja alocado para resolvê-lo e o sistema não se torne anárquico.

Vários fatores podem ser considerados para dizer qual o mais indicado: o elemento ocioso, o elemento com poder de processamento mais adequado ao trabalho, o elemento mais próximo, dependendo da estrutura do sistema, entre outros. Para algumas tarefas podem ser alocados mais de um elemento. No momento em que o elemento mais adequado não esteja disponível, outro elemento tem autonomia para resolver o problema, já que este elemento, como citado anteriormente, sabe da configuração total do sistema e tem conhecimento de que o elemento mais adequado está ocupado.

3

ESTADO DA ARTE

Apesar de serem apresentadas neste trabalho soluções de controle usando redes de Petri, outras abordagens também são conhecidas. Controle baseado em orientação a objetos e controle feito por diagramas de estados são algumas delas. Contudo, por razões apresentadas posteriormente, as redes de Petri foram escolhidas para serem usadas nesta dissertação. Segue a apresentação de alguns trabalhos de diferentes autores sobre o tema.

No trabalho de MORANDIN e KATO (2005) é proposto um método de modelagem e controle que possui dois passos: uma tarefa de modelagem para cada subsistema de FMS e uma tarefa de agregação.

Cada subsistema deve ser modelado usando-se redes de Petri Virtuais (VPN) de modo que um subsistema corresponda a um módulo. Tais módulos devem considerar as possíveis interações com os demais.

No trabalho proposto são considerados alguns módulos: Máquinas CNC, veículos auto guiados, caminhos e também planos alternativos de processo. A partir dos módulos são gerados modelos usando VPN's. A solução de agregação considera os nós virtuais (lugares e transições). Usando estes elementos, é necessário combinar o módulo de máquina CNC com o módulo do AGV e o resultado com o caminho do AGV. Finalmente, combinando o último módulo (plano alternativo de processos), o modelo é completado e as técnicas de análise podem ser aplicadas.

Em outro trabalho de MORANDIN e KATO (2003) é proposta uma outra abordagem de modelagem de sistemas automatizados de manufatura através de redes de Petri.

O controle de sistema automatizado de manufatura (AMS) pode ser executado de uma maneira integrada usando dois diferentes equipamentos: Sistema de Controle Supervisório (SCS) e Controlador Lógico Programável (CLP). O SCS lida com as várias seqüências de operação de controle de um sistema de produção e com a interface com o operador, enquanto o CLP lida com o controle de inter-travamento dos elementos. No entanto, os dois equipamentos não trabalham de maneira separada, pois trocam informações entre eles através de sinais digitais.

O modelo de controle do AMS envolve a elaboração de dois modelos de redes de Petri. No “passo de construção” um modelo de rede de Petri representa o programa completo do CLP. No “passo de validação” alguns elementos são adicionados ao modelo para prover um aspecto dinâmico de verificação e validação. Por fim, o modelo deve estar livre de *deadlocks* e não ter conflitos.

Resumidamente, no “passo de construção” são usadas redes de Petri para descrever a ligação entre o sistema do CLP e o controle de seqüência do SCS. Os lugares da rede representam sinais de entrada e saída de modo que um lugar pode ser entendido como uma *flag* do sinal indicando se este está ativo ou não. As transições são interpretadas como ações que devem ser tomadas pelo CLP.

No “passo de validação” o modelo de controle do sistema é simulado. Para isto, arcos, lugares e transições são incorporados ao modelo de controle da rede para simular o comportamento e a análise de desempenho. Neste passo, o principal aspecto a ser analisado é a existência de *deadlocks*.

No trabalho de YAN, WANG, CUI e ZHANG (1997), é proposta uma forma de modelar e controlar sistemas flexíveis de manufatura através da avaliação de redes de Petri de alto-nível estendida. Ele apresenta a EHLEP-N (*extended high level evaluation Petri nets*) e descreve a estrutura e as propriedades dinâmicas de FMS que podem ser usadas para desenvolver um escalonamento e controlar sistemas especialistas para FMS.

Uma EHLEP-N deve ter as seguintes características: 1) os modelos resultantes são mais simples que outros e mais fáceis de serem descritos; 2) regras (regras de escalonamento e controle, algumas que mandam ordens ao equipamento) e respostas são diretamente introduzidas no EHLEP-N para fazê-lo sincronizar com o equipamento, além de controlá-lo, assim executando a função de troca de informação com o equipamento; e (3) marcas e marcações duplas são introduzidas no EHLEP-N com uma para a observação da distribuição de marcas e avaliação de desempenho do FMS e a outra para escalonamento e controle do FMS. Respostas são indicadas por cadeias de caracteres enviadas pelo equipamento e refletindo seus estados (YAN; WANG; CUI; ZHANG, 1997).

Segundo seus autores, a EHLEP-N foi proposta a fim de estender e combinar as vantagens de redes de Petri já existentes. Porém, mantém as características vistas como vantagens e descarta as desvantagens. As marcas, marcações, marcas coloridas e marcações coloridas são definidas simultaneamente. Para aproximar a EHLEP-N com ambientes de manufatura, todas as regras (regras de escalonamento, regras de controle, etc.) e respostas externas são introduzidas diretamente no EHLEP-N, fazendo as atividades da EHLEP-N sincronizar com ambientes externos e controlar equipamentos de processamento (máquina NC, centros de processamento, etc.) e os veículos no FMS. Isso torna o significado físico das atividades da EHLEP-N mais explícita e compreensiva.

Em ZIMMERMANN, BODE e HOMMEL (1996) é primeiramente apresentado um histórico da evolução das redes de Petri no contexto de sistemas de manufatura. O trabalho passa por extensões como: PN coloridas, PN estocásticas, PN inteligentes, PN coloridas e PN modulares.

Existe a citação a respeito de um sistema gráfico para descrever sistemas de manufatura proposto por VILLAROEL, MARTÍNEZ e SILVA (1989). Este sistema chama-se GRAMAN. Seus modelos de redes de Petri descrevem hierarquicamente a execução de ordens em diferentes níveis de abstração. Esses modelos não são compilados no modelo interno, eles interagem com o modelo estrutural por um tipo de mecanismo próprio. A ordem de produção pode ser passada ao modelo de PN por um nível superior do controlador do sistema flexível de manufatura.

Em contraste a isso, a técnica de ZIMMERMANN, BODE e HOMMEL (1996) compila as especificações da rota de produção em um modelo completo, de forma que vise o desempenho e a avaliação de dependência e não o controle do sistema de manufatura.

Como o trabalho aborda o uso de uma classe restrita de PN coloridas, não é necessário esconder as PN's do arquiteto. A saber, os arcos e expressões complexos, bem como, as definições de tipos e variáveis são supérfluos. Então, é possível modelar a estrutura do sistema de manufatura e as rotas de produção com o mesmo tipo de PN dedicadas, sem a necessidade de uma linguagem de descrição adicional.

Segundo ZIMMERMANN, BODE e HOMMEL (1996), a modificação de rotas não faz necessário reprojeter completamente o modelo, ou seja, o sistema de manufatura é independente das peças sendo processadas. Um modelo completo pode ser usado para obter medidas de desempenho e dependência usando análises numéricas ou simulação.

No trabalho de EZPELETA e COLOM (1997) é apresentada uma solução baseada na idéia de que em um FMS há dois elementos principais que devem ser considerados: a arquitetura do sistema (*layout* do FMS) e o plano de trabalho correspondente aos tipos de peças que devem ser processadas. Eles propõem uma modelagem separada desses elementos. Tendo esses modelos como entradas, pode-se obter, de maneira automática, uma PN colorida que integra ambos os modelos. A arquitetura do sistema dá a estrutura da rede (o esqueleto do modelo final), enquanto os diferentes tipos de peças determinarão os domínios de cores de lugares e transições e as funções inscritas nos arcos.

A fim de lidar com *deadlock*., duas abordagens principais diferentes foram adotadas: a abordagem de prevenção de *deadlock* e a abordagem de detecção e recuperação de *deadlock*. Na primeira abordagem, a evolução do sistema é controlada a fim de assegurar que nenhum *deadlock* seja alcançado. Na segunda, quando um *deadlock* é detectado, uma estratégia de recuperação é aplicada para alterar o estado do sistema para um estado sem *deadlock*.

São desenvolvidos modelos separados da arquitetura e dos planos de processo. Essa abordagem apresenta duas vantagens principais. 1) Permite identificar os componentes do FMS e a localização das peças nele, e 2) mantém o mesmo modelo da arquitetura com objetivo de testar diferentes planos de processos. A seguir será mostrada uma síntese do modelo completo.

Aqui é descrito como integrar ambos modelos a fim de obter-se o modelo global: uma PN colorida. Cada um dos modelos contribui para o modelo final como segue: 1) A arquitetura do sistema provê a estrutura básica da CPN (o esqueleto). Os lugares representando as capacidades dos recursos são mantidos no modelo final. 2) Um plano de processo é associado a cada peça a ser manufaturada. Uma peça é modelada pelo significado da cor de uma marca. A cada estado, o lugar onde a marca representando a peça está, representa o recurso físico onde a peça pode ser encontrada.

O processamento de uma peça de acordo com seu plano será representado por duas mudanças relativas a marcas coloridas. A primeira mudança trata da localização da marca (correspondente ao movimento da peça no sistema). A segunda mudança corresponde à própria marca colorida: quando o processamento de uma peça termina, a marca colorida deve atualizar essa informação.

O formalismo das PN coloridas permite expressar essas mudanças de maneira elegante/compacta: pelos significados das funções inscritas nos arcos da rede. Tais funções asseguram que cada marca visita somente aqueles lugares correspondentes aos recursos especificados no seu plano de processamento. As funções nos arcos e os domínios de cores dos lugares e transições estabelecem as mudanças de estado que são permitidas e aquelas que não são.

No trabalho, dois problemas relacionados a ambientes de FMS são considerados. O primeiro é a modelagem de tais sistemas. O segundo é o controle desses sistemas a fim de assegurar um bom comportamento, que significa que nenhum problema de *deadlock* possa surgir. Com respeito à primeira questão, uma abordagem de modelagem é apresentada. Essa abordagem leva vantagem na separação entre a modelagem da arquitetura do sistema (considerado como composto de um conjunto de recursos com suas interconexões: possibilidade da peça fluir entre recursos) e a modelagem dos

planos de processamento correspondendo aos tipos de peças a serem processadas. A partir dessas entradas, pode-se gerar, de maneira automática, um modelo de PN colorida que integra ambos modelos. Com respeito à segunda questão, é mostrado como a política de prevenção de *deadlock* pode ser aplicada aos sistemas considerados. Do ponto de vista da modelagem, essa política pode ser também aplicada automaticamente.

No trabalho de VELASCO, GONZALEZ, MAGDALENA e IGLESIAS (1996) é apresentada uma arquitetura geral e uma plataforma desenvolvida para implementar aplicações distribuídas como um conjunto de agentes inteligentes cooperantes. Também é mostrado como essa arquitetura foi usada para implementar um sistema distribuído de controle por um processo complexo: o controle econômico de um combustível fóssil para usina de força.

Agentes nessa aplicação encapsulam diferentes entidades distribuídas de *software/hardware*: controladores *fuzzy* e neurais, um sistema de aquisição de dados, etc. Esses agentes são definidos em ADL (Agent Description Language), uma linguagem de especificação de alto nível, e intercâmbio de dados/conhecimento através de requisições de serviço usando uma linguagem comum de conhecimento-representação.

PARK, TILBURY e KHARGONEKAR (2001) apresentam uma representação formal unificada de controladores lógicos com os três modos de controle usando-se redes de Petri. Uma estrutura modular de controlador lógico é introduzida e formalizada para linhas de transferência de alto-volume. O controlador lógico modular consiste de um módulo de controle para decisão de modo e módulos de controle para controladores lógicos na estação.

Cada módulo de estação de controle é representado por módulos de conexão de operações que são projetados respeitando processos de recuperação de problemas de operações; seus algoritmos de conexão também são apresentados. Na representação formal, cada módulo de controle é representado por uma PN viva, segura e reversível, condição para que o controlador lógico modular gere um controle lógico correto é mostrado: operação de condição casual. Usando a estrutura modular de um controlador lógico, o controle lógico pode ser facilmente reconfigurado e geração automática de código é possível.

A seguir é apresentado o trabalho de LEE e HSU (2003): Para processos multi-receitas controlados remotamente, certos comandos podem levar a usos indesejáveis de recursos e estados de *deadlock*. Como um resultado do atraso de comunicação e perda de pacotes na internet, controle remoto de um grupo de certos passos são recomendados ao invés de controle de cada passo. Para esses sistemas de controle baseados na Web, o trabalho propõe uma arquitetura de três níveis com supervisão hierárquica.

No nível mais alto, o agente supervisorio garante que comandos emitidos a partir do gerente remoto, no nível médio de comando, satisfaça às necessidades de divisão de recursos e ausência de *deadlock* quando executar um número de receitas concorrentes, enquanto o controlador local, no nível mais baixo de controle, execute a operação detalhada correspondente para encontrar as tarefas requisitadas.

Nesta abordagem, PN são usadas para modelar, projetar, e verificar o agente supervisorio e controlador local, e então, a tecnologia Java e diagramas lógicos LADDER (LLD) são adotados para implementação com o propósito de alcançar supervisão online dos processos remotamente controlados. A fim de demonstrar a praticabilidade da solução desenvolvida para o supervisorio, ela é aplicada a um FMS com três receitas controlado pela Internet.

O trabalho de DESROCHERS, DEAL e FANTI (2005) apresenta uma nova extensão para PN ordinárias que usam marcas de valores complexos. Ao permitir dois tipos de marcas, “real” e “imaginária”, cada marcação no lugar contém informação de quantidade e tipo. PN de marcas de valores complexos foram projetadas para possuir similaridade com outras extensões populares de PN tais como redes temporizadas, redes estocásticas, e redes coloridas. Os autores apontam que essa aplicação simples e intuitiva de números complexos e aritmética complexa às PN’s provê uma ferramenta de modelagem única. Alguns exemplos mostram o potencial dessa classe de PN’s. O trabalho também apresenta uma nova abordagem de análise de desempenho com aplicações para gerenciamento de *supply chain*.

Em FANTI, GIUA e SEATZU (2003) afirma-se que controle de tráfego em tempo real em redes de via férrea autoriza movimentos de trens e impõe restrições de

segurança. O trabalho lida com o controle de tráfego em tempo real focado no problema de prevenção de *deadlock*. Redes de Petri coloridas são usadas para modelar os dinamos dos sistemas de rede de via férrea: lugares representam vias e estações, marcas são trens. A política de prevenção é expressa por um conjunto de restrições lineares desiguais, chamadas Restrições Gerais de Exclusão Mútua coloridas. Restrições que são reforçadas pela adição de lugares monitores apropriados. Usando dígrafos, situações de *deadlock* são caracterizadas e uma estratégia é estabelecida para definir um conjunto de Restrições Gerais de Exclusão Mútua que previnem *deadlock*.

NKETSA e VALETTE (2001) apresentam uma abordagem para simular e implementar, através de refinamentos graduais, o sistema de manufatura completo pela distribuição da simulação. Essa abordagem é baseada no uso de diferentes classes de PN para modelar diferentes níveis de um sistema de manufatura. Essas classes comparam-se a níveis de abstração de um PN de alto nível usadas para modelar o sistema de manufatura. Cada nível pode ser simulado em um processador ou em um *cluster* de processadores que podem comunicar-se entre si usando uma rede. A principal contribuição é dar a oportunidade de combinar simulação, avaliação de desempenho e emulação. A emulação significa que uma parte do sistema pode ser executado em tempo real enquanto outra parte é simulada. Baseada em níveis de abstração de PN de alto nível, subsistemas podem ser integrados passo a passo a partir do estágio de projeto até o estágio de implementação, permitindo mudanças ente componentes simulados e sistemas físicos de tempo real. Essa abordagem é conseguida pela definição de uma máquina de simulação que envolve um simulador local, um emulador e uma interface para o processo físico.

Em ZUBEREK (2004) Petri Nets temporizadas são modelos formais de sistemas concorrentes discretos. Uma vez que o tempo de duração de todas as atividades estão inclusos na descrição do modelo, muitas características de desempenho podem ser derivadas de tais modelos. No caso de ferramentas de *cluster*, modelos de rede representam o fluxo de biscoitos através de câmaras da ferramenta tão bem quanto ações consecutivas executadas pelo robô transportador. Um desenvolvimento sistemático de ferramentas de programação, baseado em uma descrição de comportamento geral da ferramenta, é proposto e usado para derivar os modelos de PN

correspondente. O desempenho das ferramentas modeladas é medido pelo uso de lugares invariáveis, sem análise exaustiva de alcançabilidade.

LI, ZHOU (2004) afirma que uma variedade de importantes métodos baseados em PN surgindo, ultimamente, em sistemas flexíveis de manufatura são para adicionar alguns lugares de controle e arcos relacionados a sifões estritos mínimos (SMS) tal que nenhum sifão pode estar vazio. Como o número de sifões mínimos cresce, em geral, exponencialmente com respeito a um tamanho de PN, suas desvantagens detém-se em que eles freqüentemente adicionam muitos lugares à rede, fazendo, assim, o modelo de rede resultante muito mais complexo do que o original. Seu estudo explora maneiras de minimizar a nova adição de lugares conseguindo o mesmo propósito de controle. Eles propõem, pela primeira vez, o conceito de sifões elementares, que são uma classe especial de sifões. O conjunto de sifões elementares em uma PN é geralmente um subconjunto próprio do conjunto de todos os SMS. Sua cardinalidade menor se torna evidente em modelos grandes de PN. Seu estudo prova que, por adicionar um lugar de controle para cada sifão elementar a fim de assegurar que ele é marcado, o *deadlock* pode ser prevenido. Comparado com métodos existentes, o novo método requer um número muito menor de lugares de controle e, ainda, é compatível com PN de larga escala. Um exemplo de FMS é usado para ilustrar as políticas e modelos propostos, e mostrar a vantagem significativa sobre métodos anteriores.

Conforme PENG e ZHOU (2004), diagramas LADDER (LD's) para controladores lógico-programáveis são um método dominante em controles de evento discreto de sistemas industriais automáticos. A funcionalidade e complexidades crescentes desses sistemas desafiaram o uso de LD's para projetar seus controladores de eventos discretos. Segundo PENG e ZHOU (2004), pesquisadores estão constantemente perseguindo ferramentas integradas que eliminam as limitações de LD's. Essas ferramentas são alvo de estudos não somente para controle, mas também para sistemas de análise, avaliação, e simulação. Eles citam que, pelas várias décadas passadas, PN's surgiram como uma importante ferramenta para prover uma solução integrada para modelagem, análise, simulação, e controle de sistemas industriais automáticos. Diferentes tipos de controladores baseados em PN são propostos com objetivos de serem aplicados na indústria. Eles escrevem que há uma necessidade por mais estudos de métodos de medida de desempenho de PN e LD com intuito de formar um

framework estruturado e integrado para o desenvolvimento de *softwares* de controladores lógicos.

4

PROPOSTA

4.1 Justificativa

Esta seção tem por objetivo justificar, entre as várias opções, as escolhas das abordagens usadas no momento da modelagem. Cada tipo de abordagem escolhido possui um conjunto de fatores considerados vantajosos e, por isso, podem contribuir de maneira eficiente na elaboração e validação do modelo.

4.1.1 Arquitetura híbrida

O contexto escolhido para a aplicação do método de modelagem possui controladores ligados a cada elemento automático do sistema. Todos os controladores estão ligados a um controlador supervisorio central. O supervisorio possui a tarefa de armazenar e indicar o plano de produção aos controladores locais. Ele também é responsável por indicar os roteiros dos produtos pelo sistema. Mesmo tendo o supervisorio como um “gerente” geral, cada controlador local também está ligado a outros controladores locais (Figura 4.1). Portanto, algumas soluções são tomadas pelos próprios controladores locais.

O próprio supervisorio também possui ligações com módulos superiores inteligentes. Tais módulos são responsáveis por elaborar um planejamento de produção inicial e então repassá-los ao supervisorio. Eles também são responsáveis por resolver possíveis problemas que possam ocorrer durante a fabricação e que necessitem de uma reformulação do plano de produção. Tais problemas geralmente necessitam de um processamento computacional mais custoso e complexo, o que não é o caso dos

controladores locais nem do sistema supervisorio. Por essa razão, tais problemas são remetidos a esses módulos inteligentes. Eles são formados por máquinas apropriadas para executar softwares mais elaborados.

A ligação dos controladores locais entre si reporta a uma arquitetura heterárquica. Porém a ligação dos controladores locais com um controlador supervisorio leva a uma arquitetura hierárquica. Essa mescla de arquiteturas faz com que a arquitetura híbrida pareça a mais adequada nesse contexto. Mais ainda, a arquitetura híbrida possui as vantagens de alta flexibilidade aliada a um rápido tempo de resposta.

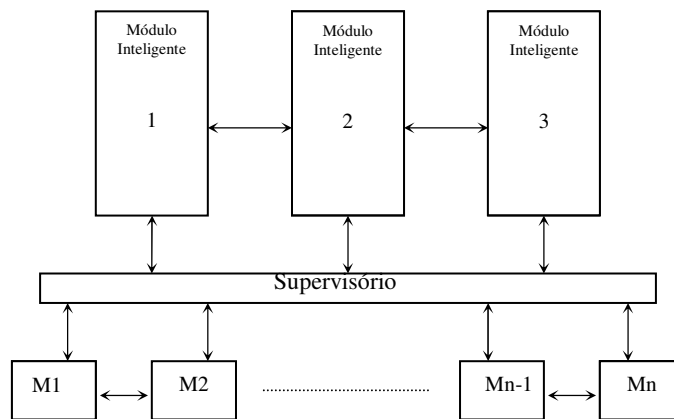


Figura 4.1 – Arquitetura de AMS considerado.

4.1.2 Simulação

Muitos autores têm dedicado seus estudos no sentido de tratar analiticamente os problemas enfrentados em AMS's. Outra frente muito estudada é a geração automática de códigos de programação (LADDER) dos elementos controladores dos hardwares. Porém, devido à complexidade inerente aos AMS's, muitos trabalhos precisam desconsiderar algumas características desses sistemas fazendo com que a sua solução teórica seja difícil de ser implementada na prática. Mesmo gerando boas soluções para problemas mais específicos, tais estudos costumam ficar um pouco aquém de soluções esperadas para problemas em sistemas reais.

Para que a simulação possa preencher adequadamente o espaço entre teoria e prática é necessário que se elabore um modelo que considere a maioria dos elementos

presentes no sistema a ser modelado. Alguns elementos são facilmente identificados para um modelo inicial, como máquinas, *buffers* e elementos de transporte. Contudo, outros elementos só serão identificados conforme as simulações vão sendo executadas, como dispositivos necessários ao inter-travamento do sistema de transporte e partes do próprio sistema supervisorio.

A simulação também permite que cenários intermediários ao início e fim da produção possam ser analisados. Dessa maneira, soluções para problemas específicos podem ser aplicadas no modelo para que a produção seja executada de maneira mais eficiente.

Outra vantagem é a possibilidade de gerar cenários problemáticos durante a simulação de um processo de produção. Quebra de máquina, quebra de elementos de transporte, alteração no plano de produção, ou seja, diferentes cenários críticos podem ser testados em momentos diferentes. Através da simulação de tais cenários pode-se estudar um mecanismo que seja capaz de tratar o fato quando ele ocorrer.

Modelos de simulação sugerem a participação humana especializada em criar, testar e interpretar os modelos, em contrapartida dos modelos automáticos. Contudo, pensa-se que o custo desse elemento a mais não é algo que possa inviabilizar o uso da estratégia em um ambiente real, dadas as vantagens nela encontradas.

4.1.3 Método *top-down*

Entende-se neste trabalho que a melhor maneira de começar a modelar um AMS específico é partindo de uma visão mais geral dos seus elementos. Logo, a observação do *layout* do chão-de-fábrica parece vir ao encontro dessa idéia, pois é ele que traz as informações de localidade dos elementos. Considerando uma arquitetura de controle que possua características de hierarquia, o resultado da primeira modelagem pode ser visto como nível 0 (zero) do modelo completo. A partir do nível 0 é possível estender o modelo em níveis inferiores conforme o nível de detalhamento dos componentes.

Através do *layout* também é possível supor, inicialmente, um possível inter-travamento entre os elementos, visto que suas localidades sugerem possíveis caminhos

de fluxo de peças. No momento dessa primeira modelagem é possível observar que o sistema de transporte traz consigo a necessidade de maior atenção, pois é ele que torna possível a interligação dos elementos localizados conforme o *layout*.

Dado à essa importância o método *top-down* de modelagem parece possuir maiores vantagens frente ao método *bottom-up*, pois confere a possibilidade de se abordar primeiramente o *layout* e então continuar a modelagem pelos elementos mais específicos.

Cada nível hierárquico do método apresenta o modelo de partes do sistema com níveis de detalhamento diferentes. Quanto mais baixo o nível, mais detalhes podem ser vistos. A partir de um certo nível é aconselhável a quebra do modelo em módulo. Isso facilita tanto a modelagem quanto a compreensão do sistema. Mesmo em módulos distintos, os elementos ficam ligados entre si. O método *bottom-up* poderia trazer problemas quando do inter-travamento entre os elementos presentes no sistema.

4.1.4 Virtual Petri Nets (VPN)

Pelo fato deste trabalho se tratar de um método de modelagem modular, existe a necessidade de se ligar os diferentes módulos presentes no modelo completo. Como o modelo é concebido através de modelagem por redes de Petri, a estratégia de ligação deve contemplar suas características e definições. Assim, uma extensão de redes de Petri parece possuir mais atributos no sentido de suprir as necessidades de ligação.

As redes de Petri Virtuais, como uma extensão de redes de Petri, conferem definições que casam com essas necessidades. Tomando nós (lugares e transições) como pontos de ligação, elas conseguem manter as formalidades referentes a redes de Petri ordinárias além de acrescentar formalidades para tratar com redes de Petri modulares.

4.1.5 CPN Tools

A ferramenta de modelagem CPN Tools possui uma grande comunidade de pesquisadores que se dedicam ao seu uso e aperfeiçoamento. Ela vislumbra todas as possibilidades de modelagem necessária para a execução do método de modelagem que

se deseja propor. Algumas particularidades de seu funcionamento estão presentes no capítulo referente aos testes. Uma explicação mais minuciosa da ferramenta pode ser encontrada no respectivo apêndice.

O recurso de simulação também é explorado na ferramenta, tornando possível a análise de passos intermediários em tempo real ou através de um arquivo `.txt` gerado durante a simulação. Entenda-se como passo o simples disparo de uma transição.

4.2 Proposta

A proposta deste trabalho é apresentar um método de modelagem para AMS's que culmine em um modelo validado para um ou mais planos de produção. Este método baseia-se no uso de diversas abordagens apresentadas e justificadas anteriormente.

O processo de modelagem é dividido de maneira que possa ser apresentado na forma de etapas. Todo o processo é feito com o uso do *software* CPN Tools. Na primeira etapa é considerado o *layout* para elaboração inicial do modelo. A segunda etapa preocupa-se com o início do inter-travamento do sistema de transporte. Com relação ao transporte, deve-se ainda, iniciar a colocação de inscrições nos arcos a fim de atribuir rotas aos produtos. Também nesta etapa deve-se considerar a definição de tipos a serem atribuídos aos lugares. Na terceira etapa são considerados alguns produtos ou veículos de transporte. Eles são necessários para que seja possível se executar a primeira simulação, que considera o fluxo desses elementos pelo sistema. Na quarta etapa há a modelagem da máquinas e respectivos *buffers*. Na quinta etapa são acrescentadas as inscrições restantes nos arcos no intuito de definir o fluxo completo das peças. Na sexta etapa é executada a tarefa de ligação entre os módulos. A sétima etapa é reservada especificamente para a simulação. Por fim, a etapa oito acrescenta tempo no modelo.

Mesmo o método sendo apresentado através de etapas, o arquiteto tem a liberdade de adaptar as etapas de forma que agilize seu trabalho. O objetivo de separar o método em etapas é explicar de maneira didática como se aplicar o método. Cada etapa será descrita detalhadamente a seguir:

- **Primeira etapa**

Primeiramente é necessário atentar-se ao *layout* da fábrica. Assim como dito anteriormente a disposição dos elementos determina a elaboração do sistema de transporte. Através dessa observação é possível identificar os principais elementos da fábrica e os possíveis rotas por onde fluirão os produtos. O nível 0 do modelo é assim formado: Lugares são adicionados para representar os pontos de carga e descarga das máquinas. Eventualmente são adicionados lugares a fim de representar os pontos de controle de transporte. Dependendo do sistema em questão, tais pontos podem existir ou não.

Os lugares podem ser ligados através dos arcos. A determinação dos arcos define o fluxo dos produtos na PN. Por definição, sempre existe uma transição na ligação de dois lugares. As transições também são usadas como forma de bifurcar um caminho oferecendo alternativas de caminhos. O exemplo a seguir considera um FMS com seis máquinas. (Figura 4.2)

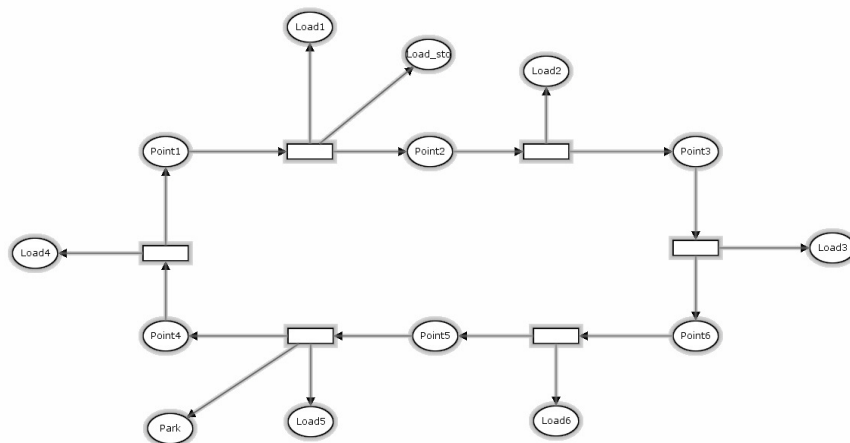


Figura 4.2 – Exemplo de um FMS modelado após a primeira etapa.

Na figura 4.2 pode ser observado os pontos de controle 1 e 2 (Point1, Point2). Do Point1 é possível que o AGV se dirija à área de carga/descarga da máquina 1 (Load1), à área de carga/descarga do armazém (Load_sto) e ao ponto de controle 2 (Point2). Isso poder ser generalizado para o resto do modelo, sendo que o estacionamento é definido como Park.

- **Segunda Etapa**

Nesta etapa deve-se verificar as restrições inerentes ao transporte. É possível que haja restrições quanto ao número de peças presentes em uma mesma posição ou a restrição de um AGV carregado com determinado produto não poder ser aceito em uma posição de carga e descarga, por exemplo.

Logo, são adicionados elementos que fazem esse papel. Estruturas específicas de PN, como arcos inibidores, possuem grande utilidade quando se deseja modelar esses tipos de restrições.

Também é interessante iniciar nesta etapa o processo de definição de tipos a serem atribuídos aos lugares. Os tipos são, na verdade, conjuntos de cores. No CPNTools existe um menu de declarações onde isso é feito.

Normalmente, define-se um mesmo conjunto para elementos com papéis similares. Por exemplo, lugares por onde passam veículos de transporte recebem o mesmo conjunto de cores. Lugares por onde fluem as peças também recebem o mesmo conjunto, desde que este seja diferente do conjunto atribuído ao grupo anterior. Um conjunto de cores para os elementos de controle também é necessário. Um elemento de controle pode ser um lugar com o papel de simular um arco inibidor por exemplo. Este é o nível 1.

Também é necessário atribuir inscrições aos arcos de maneira a indicar o caminho que uma peça ou veículo de transporte deve seguir. Juntamente com as declarações anteriores também é possível adicionar variáveis que possam auxiliar nessa tarefa (Figura 4.3).

Na figura 4.3 pode-se verificar as marcas referentes ao controle de arcos inibidores. O lugar `Inhib4` possui um marca `e` e três marcas `f`. A letra `E` presente no lado direito inferior do lugar indica o tipo do lugar, bem como o tipo de suas marcas. A lógica implícita nas inscrições dos arcos faz com que a marca `e` seja consumida assim

que um AGV ocupa o **Point4**. Assim que o AGV deixa o **Point4**, a marca **e** volta a estar presente em **Inhib4**. Também através das inscrições dos arcos pode-se verificar a utilização das marcas **f** quando da movimentação de um AGV de **Point5** até **Load5**, ou seja, quando um AGV se move do ponto de controle 5 até a área de carga/descarga da máquina 5.

- **Terceira Etapa**

Definir um lugar de entrada de peças ou matérias-primas. Neste lugar deve-se atribuir uma quantidade de peças para uma simulação prévia. Isso deve ser feito no caso de um AMS sem veículos de transporte, ou seja, com sistema de transporte fixo, como uma esteira. Já no caso de um AMS que possua veículos de transporte o mais interessante é criar as marcas que serão interpretadas como sendo os veículos. Esse processo visa verificar se o fluxo pelo sistema de transporte está funcionando de maneira correta. Eventuais ajustes devem ser feitos caso algum problema seja verificado.

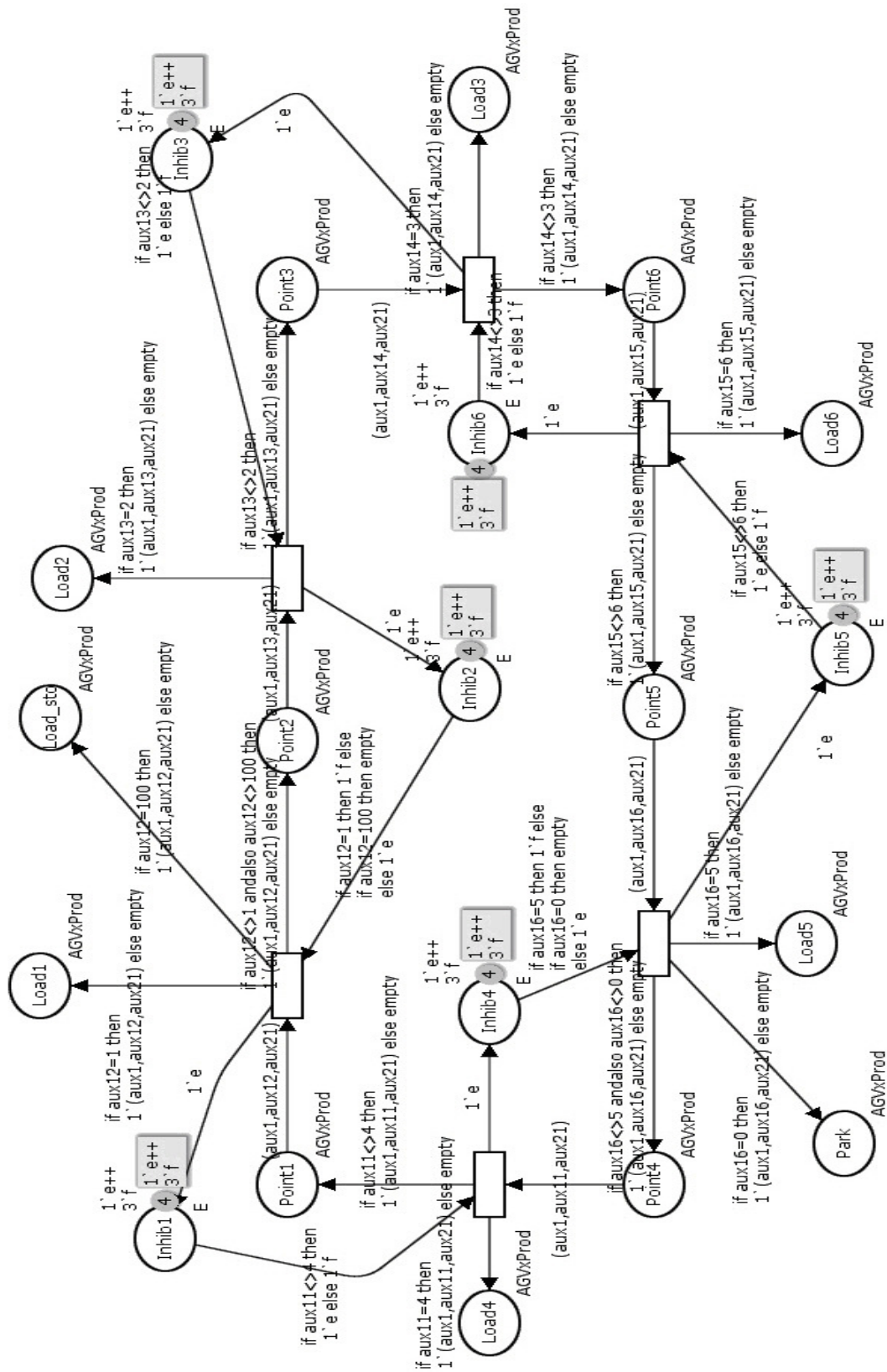


Figura 4.3 – Exemplo de um FMS modelado após a segunda etapa.

Note que ainda não são consideradas as máquinas especificamente. O resultado da simulação nesta etapa deve ser analisado de uma maneira mais genérica. Caso haja problemas os ajustes devem ser feitos (Fig 4.4).

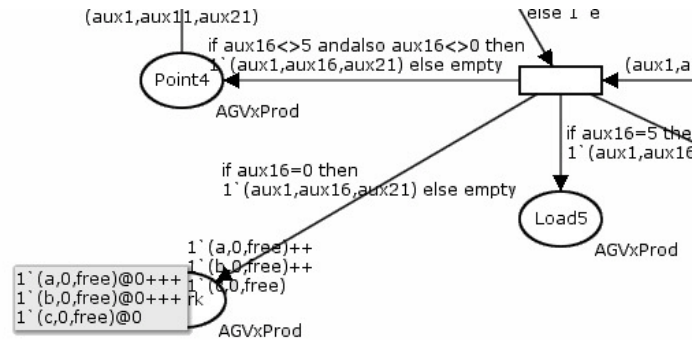


Figura 4.4 – Exemplo de parte de um FMS modelado após a terceira etapa. No detalhe podem ser vistas as marcas referentes aos AGV's.

Podem ser observados os AGV's a, b e c na figura 4.4. Os AGV's estão no estacionamento (Park) onde o tipo das marcas aceitas é AGVxProd. Apesar de não haver seta partindo de Park, os AGV's podem seguir para Point4 através de um arco presente apenas no módulo de detalhamento de Park.

- **Quarta Etapa**

Esta etapa adiciona mais um nível de detalhamento: nível 2. Aqui, as máquinas devem ser totalmente modeladas, bem como armazéns de peças e buffers. Geralmente, há a necessidade de que tais elementos sejam modelados em módulos distintos. O caminho por onde flui uma peça e seu meio de transporte deve ser considerado dentro do módulo, desde sua entrada até sua saída.

Mesmo que não haja ainda uma ligação real entre os módulos, essa tarefa cria lugares que remetem a lugares semelhantes no módulo principal. Na etapa sete haverá a interligação entre os módulos através desses lugares.

Existe a necessidade de uma definição mais apurada das peças no lugar de entrada. Deve-se considerar os tipos de peças que serão utilizadas na simulação final. Caso haja necessidade, a ordem de entrada das peças no fluxo de produção também

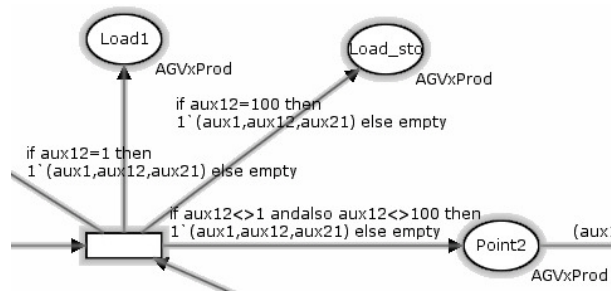


Figura 4.6 – No detalhe, inscrições nos arcos a fim de definir caminhos.

Conforme a figura 4.6, as possíveis rotas de um produto devem ser definidas através das inscrições nos arcos. Na figura, podem ser atingidos a área de carga/descarga da máquina 1 (Load1), o armazém de peças (Load_sto) e o ponto de controle 2 (Point2).

- **Sexta Etapa**

Ligar os módulos entre si. Se todas as etapas anteriores foram feitas de maneira correta deve-se identificar os lugares que representam a mesma estrutura, mas em módulos diferentes. Eles serão, então, fundidos de maneira que sejam um só, porém com representação gráfica em mais de um módulo. Caso não seja possível identificar os lugares a serem ligados, deve-se voltar ao passo anterior a fim de que a modelagem considere um lugar em comum entre os módulos (Figuras 4.7 e 4.8).

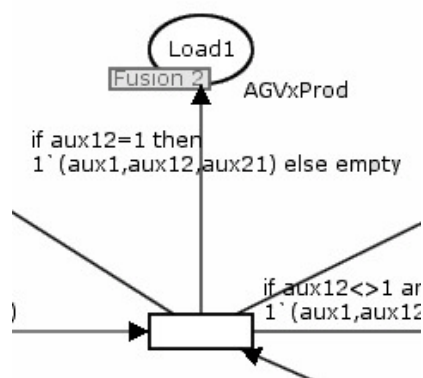


Figura 4.7 – Exemplo de um lugar que faz uma ligação com outro lugar, porém em módulo distinto.

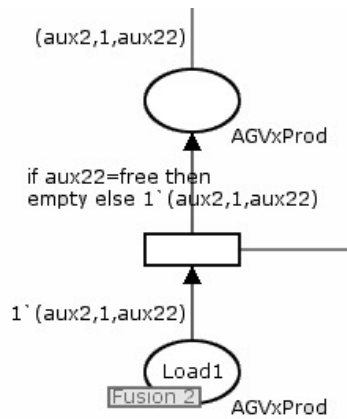


Figura 4.8 – Exemplo de um lugar que faz uma ligação com outro lugar, porém em módulo distinto.

As figuras 4.7 e 4.8 fazem parte de módulos distintos, porém possuem o lugar **Load1** em comum. A *tag Fusion 2* presente no lado inferior dos lugares indica a fusão dos lugares. Logo, eles podem ser vistos como o mesmo lugar, mesmo estando em módulos distintos.

- **Sétima Etapa**

Simular a produção das peças através do sistema. Obviamente, problemas vão aparecer. A cada problema que apareça durante a simulação um ajuste deve ser feito e a sétima etapa repetida. Os ajustes podem demandar a modelagem de novos elementos, estruturas de controle da própria rede e até a criação de novos módulos.

- **Oitava Etapa**

Adicionar o tempo necessário na execução de cada ação. Todo passo de transporte, assim como o processamento, deve receber um “tempo” representado por uma unidade de tempo hipotética. A cada problema que apareça durante a simulação um ajuste deve ser feito e a oitava etapa repetida. Os ajustes podem demandar a modelagem de novos elementos, estruturas de controle da própria rede e até a criação de novos módulos (Figura. 4.9).

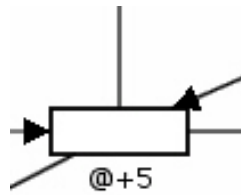


Figura 4.9 – Exemplo de uma transição com definição de tempo para a ação por ela representada.

A figura anterior apresenta a forma como o tempo é acrescentado a uma transição. O símbolo @+5 remete ao tempo corrente mais cinco unidades de tempo.

O método encerra-se assim que a simulação não possa mais prosseguir uma vez que todos os produtos já foram feitos e alocados em seus lugares devidos. É aconselhável rodar a simulação várias vezes para uma mesma configuração a fim de validá-la. A partir de então novos planos de produção podem ser testados, bem como situações críticas. Para isso, basta que estruturas da rede sejam alteradas.

5

VERIFICAÇÃO DA PROPOSTA

Este trabalho propõe um método de modelagem a fim de controlar e fazer o inter-travamento de Sistemas Automáticos de Manufatura (AMS) usando Redes de Petri. Os estudos foram baseados em um exemplos de Sistema Flexível de Manufatura (FMS) inspirados em modelos já estudados no LIAA (Laboratório de Inteligência Artificial e Automação) da UFSCar. Os sistemas modelados fizeram uso de VPN para ligar todos os seus diferentes módulos. Este método visa facilitar o entendimento de modelos gráficos sem perda de inter-travamento entre os elementos.

A escolha do sistema seguinte - FMS - deveu-se, em parte, aos estudos já realizados pelo grupo em trabalhos anteriores. Sempre foram considerados elementos de um mesmo FMS hipotético. Apesar de não haver um protótipo de FMS disponível no laboratório para testes, este tipo de sistema não deve ser desconsiderado nos estudos realizados, já que vários autores concentram neles seus estudos e trabalhos. Sua complexidade atrai vários estudos dado que sua utilidade real pode ser algo altamente eficaz.

5.1 Aplicação do método em um FMS

O *layout* do FMS considerado pode ser visto na figura a seguir na figura 5.1.



Figura 5.1 – Exemplo de uma transição com definição de tempo para a ação por ela representada.

O modelo considerado possui as seguintes características:

- Existem cinco tipos de produtos a serem produzidos, A, B, C, D e E. Cada produto possui um roteiro de fabricação a partir de uma das quatro matérias-primas existentes: x, u, v e w.
- As matérias-primas estão armazenadas inicialmente no armazém de peças (*storage*). Elas são deslocadas através do sistema pelos veículos auto guiados (AGV's).
- Cada peça é deslocada dentro do sistema seguindo um dos roteiros pré-determinados para a fabricação de determinado produto. O supervisor é responsável por indicar o roteiro a ser seguido.
- Existem três AGV's. Eles se encontram inicialmente no estacionamento (**Park**). Cada AGV recebe uma solicitação de onde deve ir. Esta solicitação pode ser recebida tanto no estacionamento quanto nas máquinas. Cada AGV pode carregar apenas um produto por vez e sua trajetória segue apenas no sentido horário através do sistema.
- Existem seis máquinas no sistema, sendo que cada uma delas possui uma área de carga e descarga capaz de acolher três AGV's. Em cada máquina também existem dois *buffers*. Um de entrada e outro de saída. Neste modelo não são consideradas as capacidades máximas dos *buffers*.
- Cada máquina possui um tempo de processamento da peça. Este tempo varia entre as máquinas, contudo é constante para uma determinada máquina, independente do produto que vai ser processado.

- Também existe cálculo de tempo na trajetória. Ou seja, é considerado o tempo levado por um AGV para se deslocar de um ponto de controle até outro.
- A trajetória é provida de pontos de controle. Não é possível que dois ou mais AGV's ocupem o mesmo ponto de controle. A partir de um ponto de controle é possível que o AGV se desloque ao ponto de carregamento de uma máquina referente ao ponto em que está. Também há a possibilidade de seguir para o ponto de carga e descarga do armazém e para o estacionamento, caso estes elementos estejam ligados ao ponto em que o AGV está.
- O roteiro de produção de cada produto é mostrado a seguir:

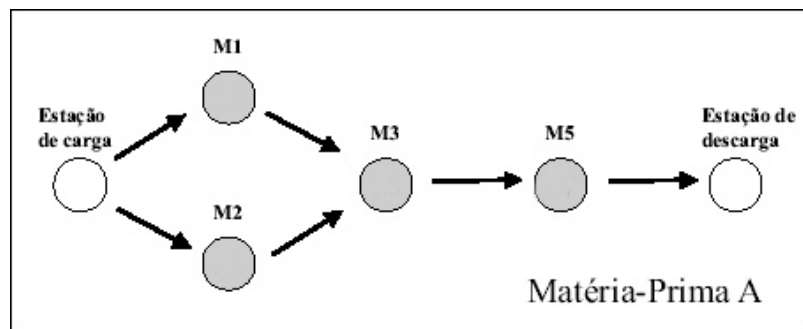


Figura 5.2a – Seqüência de produção do produto A.

Na figura 5.2a é mostrada a possibilidade de o produto A seguir por uma seqüência de máquinas, (M1, M3, M5), ou por outra (M2, M3, M5).

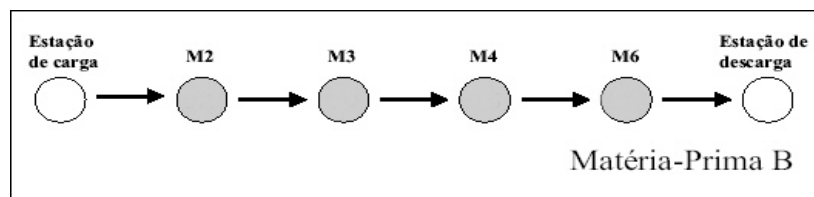


Figura 5.2b – Seqüência de produção do produto B.

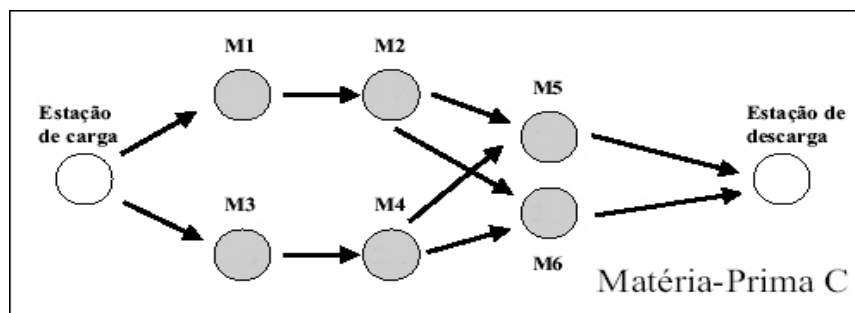


Figura 5.2c – Seqüência de produção do produto C.

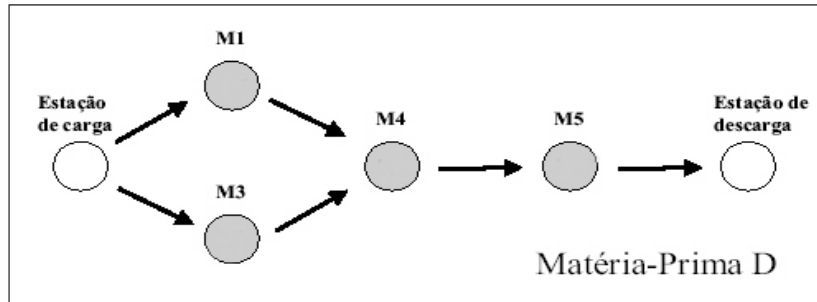


Figura 5.2d – Seqüência de produção do produto D.

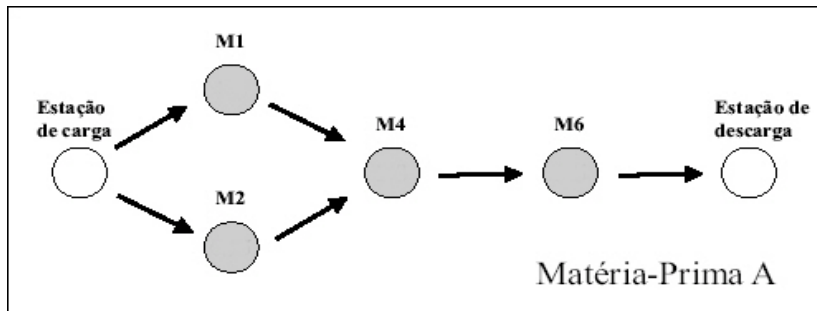


Figura 5.2e – Seqüência de produção do produto E.

A estratégia de abordagem usada aqui é a *top-down*. Outras estratégias usadas anteriormente também estão presentes neste modelo. PN modulares, temporizadas, coloridas e virtuais são particularidades em ambos modelos.

Antes de detalhar cada conjunto de elementos do modelo é preciso esclarecer que as figuras referentes a este modelo possuem elementos gráficos que representam a marcação de um lugar e também outros elementos que mostram as *tags* de fusão dos lugares. Todas as marcações presentes nas figuras representam a marcação inicial do sistema.

- **Primeira etapa**

A partir de uma observação mais geral foram criados lugares referentes a pontos de controle e ligados através de arcos de maneira que as possibilidades de caminhos já fossem estabelecidas. Para tal, também foram mapeados os lugares referentes a pontos de carga e descarga das máquinas, bem como o ponto de entrada e saída do

estacionamento e armazém de peças. Todos esses elementos juntamente com suas funções direcionam a construção correta dos arcos (Figura 5.3).

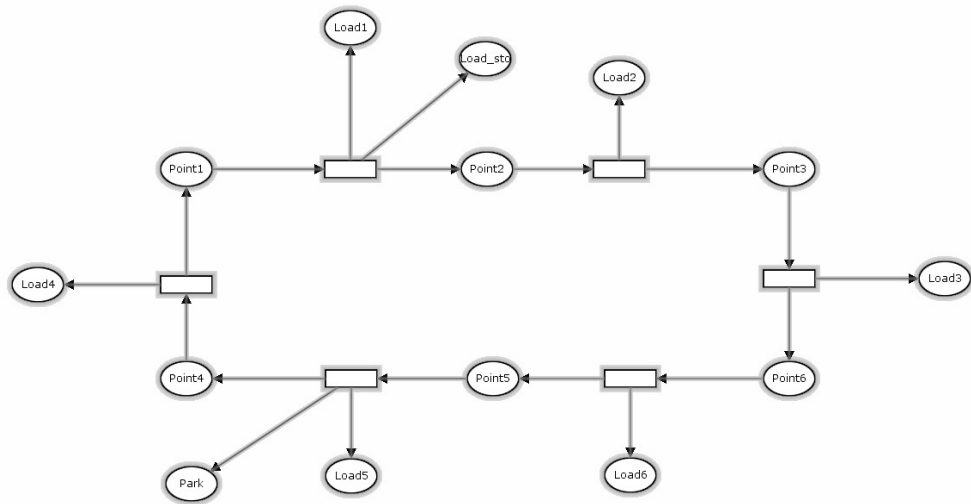


Figura 5.3 – Modelo após a primeira etapa.

Os lugares Point dizem respeito aos pontos de controle. Já os lugares Load referem-se aos pontos de carga e descarga das máquinas. Existe um lugar Load para cada uma das seis máquinas. Park representa o lugar de entrada e saída do estacionamento. Load_sto representa a entrada e saída do armazém de peças.

- **Segunda etapa**

Nesta etapa foram consideradas as restrições específicas relativas ao transporte. Cada lugar Point possui um lugar correspondente Inhib. Inhib é do tipo E. O lugar Inhib juntamente com seus arcos de entrada e saída e suas marcas servem para simular um arco inibidor. A utilização dessa estrutura garante a presença de somente um AGV em cada ponto de controle. Pode-se notar a presença de apenas uma marca de cor e na marcação inicial de cada lugar Inhib. Essa marca é consumida assim que um AGV atinge um lugar do tipo Point. Assim que a marca deixa o lugar, a marca volta ao seu lugar correspondente. Já as marcas de cor f são responsáveis pela entrada de até três AGV's no lugar que representa o ponto de carga e descarga de uma máquina situada imediatamente antes do ponto em questão. Na figura 5.4 pode-se notar que Inhib4 é

responsável pelas restrições dos lugares Point4 e Load5 através do uso das marcas e e f respectivamente.

- **Terceira etapa**

Uma das características de FMS mais difíceis de serem modeladas diz respeito ao transporte. O transporte é fator determinante no fluxo de produção. Por isso, este modelo tem foco nos AGV's. A partir de uma definição de tipos pode-se entender como sendo um AGV uma tripla formada pelos conjuntos de cores AGV, INT e Prod. O conjunto AGV possui as cores a, b e c; onde cada letra representa um dos AGV's presentes no sistema. INT é conjunto de números inteiros. Esse número indica o destino do AGV em trânsito. O conjunto Prod representa os produtos acabados ou não. Ainda representado por Prod existe a situação "free", que significa vazio, ou seja, nenhum produto está presente.

Existem três AGV's no sistema, portanto três triplas. Inicialmente os AGV's estão estacionados em Park como pode ser visto na figura 5.5. Park é do tipo AGVxProd, justamente a composição da tripla, pois $AGV \times Prod = AGV * INT * Prod$.

A partir de então pode-se fazer com que os AGV's fluam pelo sistema de maneira arbitrária. Após testes e eventuais ajustes, é possível garantir que não existem problemas com as restrições de transporte.

- **Quarta e Quinta Etapas**

Como citado na seção 4.2, estas etapas são explicadas juntamente. Durante a modelagem, elas eram uma só, porém, foram separadas para nortear mais facilmente o leitor que pretende, através deste método, fazer um modelo pela primeira vez. A explicação conjunta destas etapas também deve facilitar a compreensão do fluxo dos AGV's.

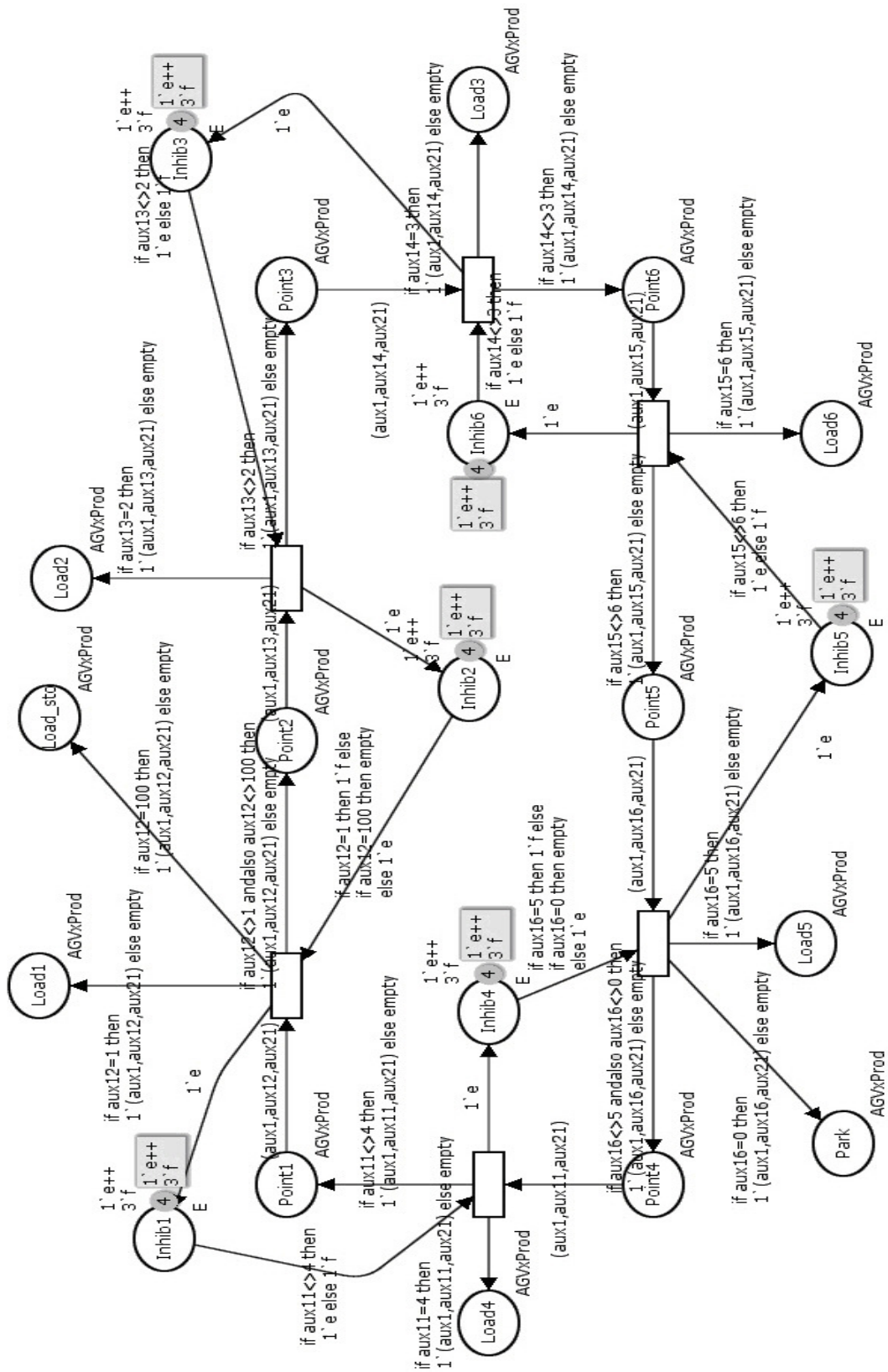


Figura 5.4 – Modelo após a segunda etapa. Estruturas simulando arcos inibidores foram acrescentadas.

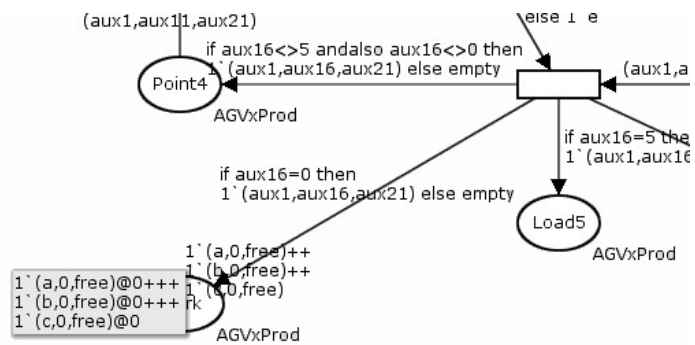


Figura 5.5 – Exemplo de parte de um FMS modelado após a terceira etapa. No detalhe podem ser vistas as marcas referentes aos AGV's.

Na prática, as primeiras solicitações vindas do supervisor são para que o AGV se dirija até o armazém, pois é lá que ele será carregado com a peça a fim de levá-la para o processamento em alguma máquina. O inteiro que caracteriza essa solicitação é 100. Assim, quando o AGV atingir Point1 e for seguir a frente, a lógica presente nos arcos fará com que a marca referente ao AGV seja colocada em Load_sto. Load_sto é do tipo AGVxProd. Esse lugar representa o lugar de carga e descarga do armazém de peças. Este lugar tem ligação com o módulo Storage, que é responsável por representar o fluxo do AGV e das peças dentro do armazém.

No Storage existem dois possíveis caminhos a seguir: pelo lugar Path_a ou Path_b. A escolha é feita pela avaliação do terceiro elemento da tripla. Se este for igual a "free", o AGV está vazio e deve seguir por Path_a para que receba uma peça, senão segue por Path_b para que descarregue um produto. Ambos são do tipo AGVxProd. Tal módulo pode ser visto na figura 5.6. O lugar Storage1 é do tipo Prods. Assim como no modelo de FMTL, Prods é um tipo usado para representar uma lista de Prod. Storage1 possui uma lista de matérias-primas. Elas estão na quantidade e ordem em que deve ser produzidas. Assim que um AGV está pronto para receber uma peça, a marca na cabeça da lista passa a compor a tripla do AGV. Essa lista faz parte do conjunto de elementos que compõem a simulação. Entenda-se, então, que sua composição é arbitrária e dependente do teste que deseja-se realizar. Ela é composta manualmente antes da simulação ser executada. Com características semelhantes, existe o lugar P1, onde há uma lista composta por marcas que representam máquinas.

até o armazém de peças. Isso faz com que o AGV não fique parado no estacionamento sendo que há peças no armazém esperando para serem transportadas. Em um FMS real, essa tarefa é exclusiva do supervisor.

Assim que sai do armazém para o **Point2**, o AGV segue pelo sistema até sua máquina destino ou até o estacionamento. Considerando que o AGV deve levar uma peça a uma máquina, ele seguirá através dos pontos de controle até que encontre sua máquina destino. A análise disso é feita a cada transição presente entre os lugares **Point** por seus arcos incidentes. Uma vez encontrada sua máquina destino o AGV segue para o lugar **Load** correspondente à máquina. Uma vez presente no lugar **Load** da máquina, deve-se analisar seu módulo correspondente.

Os módulos **Sub** representam as máquinas com seus *buffers*. Esses módulos guardam certas semelhanças com o módulo **Storage**. Pois assim como em **Storage** o AGV está passível de duas possíveis ações: descarregar uma peça ou ser carregado com uma delas. **In_Buf** recebe a marca referente à peça descarregada pelo AGV e **Out_Buf** armazena as peças processadas por **M** e prontas para serem transportadas por algum AGV. Assim como em **Storage**, um AGV, após descarregar uma peça, segue vazio para a garagem. Uma vez no lugar **M**, a marca referente à peça está pronta para ser processada. Lugares **M** são do tipo **Prod**, portanto recebem apenas marcas referentes a produtos. Assim que é processada, ação esta representada pela transição imediatamente posterior ao lugar **M**, uma solicitação é colocada em **P**. Sendo esse **P** o mesmo visto anteriormente no módulo **Storage**. Entretanto a solicitação é feita por uma marca do tipo **MACHINE** representando a máquina que processou a peça. Logo, a solicitação fica na fila até que esta esteja na cabeça da lista de solicitações e um AGV esteja no estacionamento para atendê-la. Dessa forma garante-se que a produção da máquina possa ser escoada.

Deve-se notar que a *tag* de fusão de **Inhib*a** e **Inhib*b** é a mesma. Isso significa que são o mesmo lugar, estando então separadas apenas por motivos de clareza na visualização. Deve-se lembrar também que os lugares **Inhib** presentes nos módulos **Sub** são os mesmos presentes no modelo principal, e portanto seguem a explicação de suas funções já apresentadas anteriormente.

Em alguns módulos **Sub** existem ainda dois outros lugares: um sem nome do tipo **Prod** e outro do tipo **INT** de nome **Route**. Esses lugares tem um papel importante para tratar uma característica presente no modelo proposto. Trata-se da flexibilidade das máquinas. Como uma máquina pode processar diferentes tipos de produtos, as rotas seguidas por esses a partir da máquina em questão podem ser diferentes (Figura 5.7).

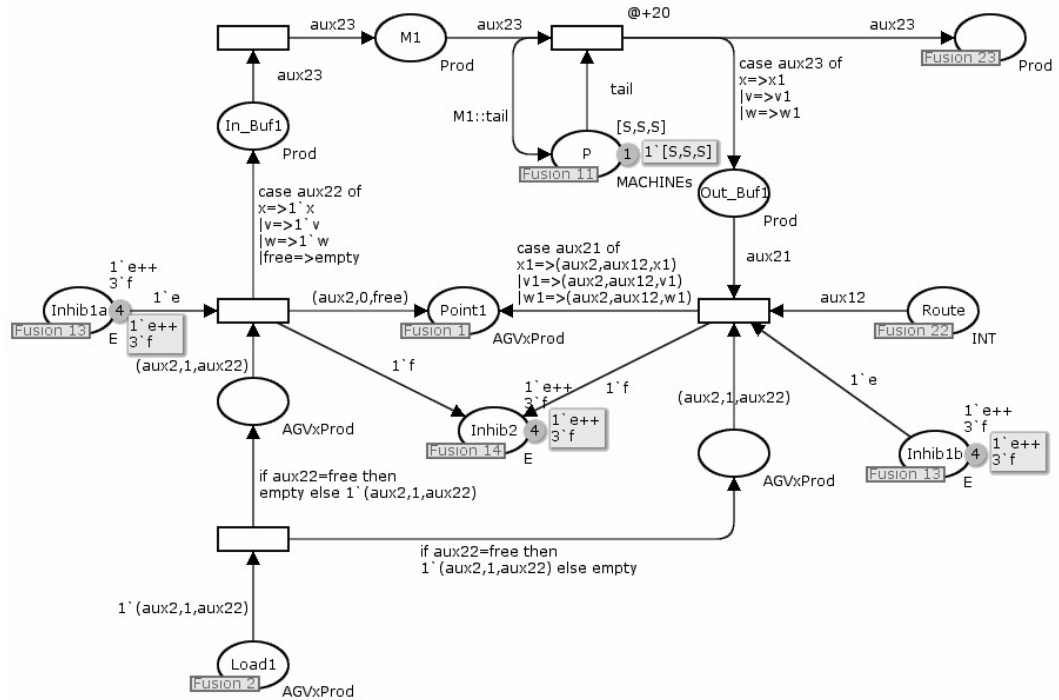


Figura 5.7 – Sub1.

Pode ser visto detalhadamente, na figura 5.7, como funciona o módulo Sub da máquina 1. O módulo **Sub1** trata de produtos do tipo x,v e w tornando-os $x1,v1$ e $w1$ respectivamente. Os produtos a serem “depositados” em **In_Buf1** podem ser vistos em seu arco incidente. Através da avaliação da variável **aux22** pode-se verificar este fato. A transição subsequente a **M1** possui a informação do tempo de processamento da peça, neste caso, 20 unidades de tempo. O arco incidente em **Out_Buf1** possui as informações referentes a peça a ser produzida de acordo com a peça de entrada. Se x , então deve-se transformá-la em $x1$; se v , então $v1$ e se w , então $w1$. Isso se deve à avaliação de **aux23**. Os arcos incidentes em **Point2** são composições diferentes de AGV.

Os lugares **Inhib1a** e **Inhib1b** são responsáveis por tratar a restrição da existência de somente um AGV em **Point1**. **Inhib1a** e **Inhib1b** são o mesmo lugar,

como pode-se notar em suas *tags* de fusão, e só estão separados por motivos visuais. Caso o AGV venha pelo lado esquerdo da figura, é consumida uma marca *e* vinda de *Inhib1a* e, portanto, de *Inhib1b* também. Caso o AGV venha pelo lado direito da figura, é consumida uma marca *e* de *Inhib1b* e, portanto, de *Inhib1a* também. Isso faz com que *Point1* seja “travado”. Já *Inhib2* recebe marcas *f* vindas de ambos os lados. Tais marcas são geradas assim que o AGV avança até *Point1*, liberando assim o ponto de carga/descarga *Load1*. Com *Inhib2* contendo marcas *f*, é viabilizada a ida de outros AGV’s até *Load1*.

É possível extrapolar a explicação anterior para o demais módulos *Sub* presentes no modelo completo. Todos eles possuem grande similaridade e por isso recebem o mesmo nome, tendo como diferenças apenas os tipos de produtos a serem produzidos e seus tempos de processamento.

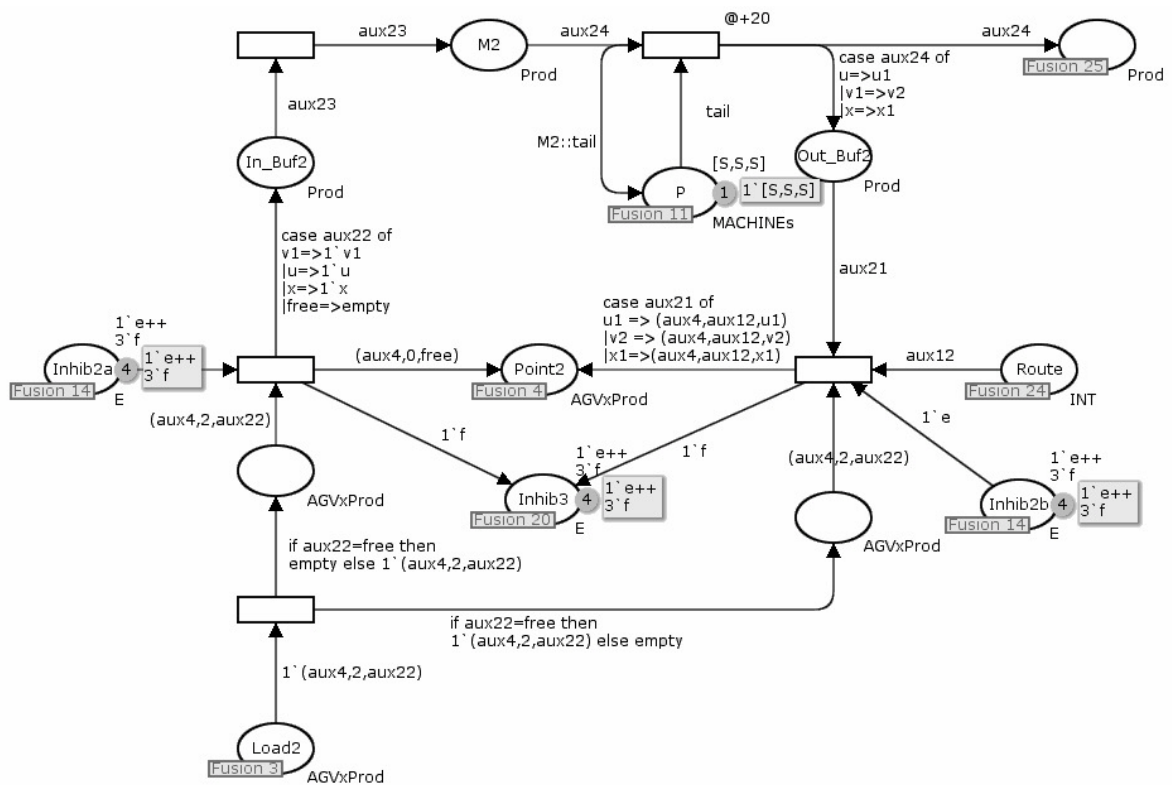


Figura. 5.8 – Módulo Sub2.

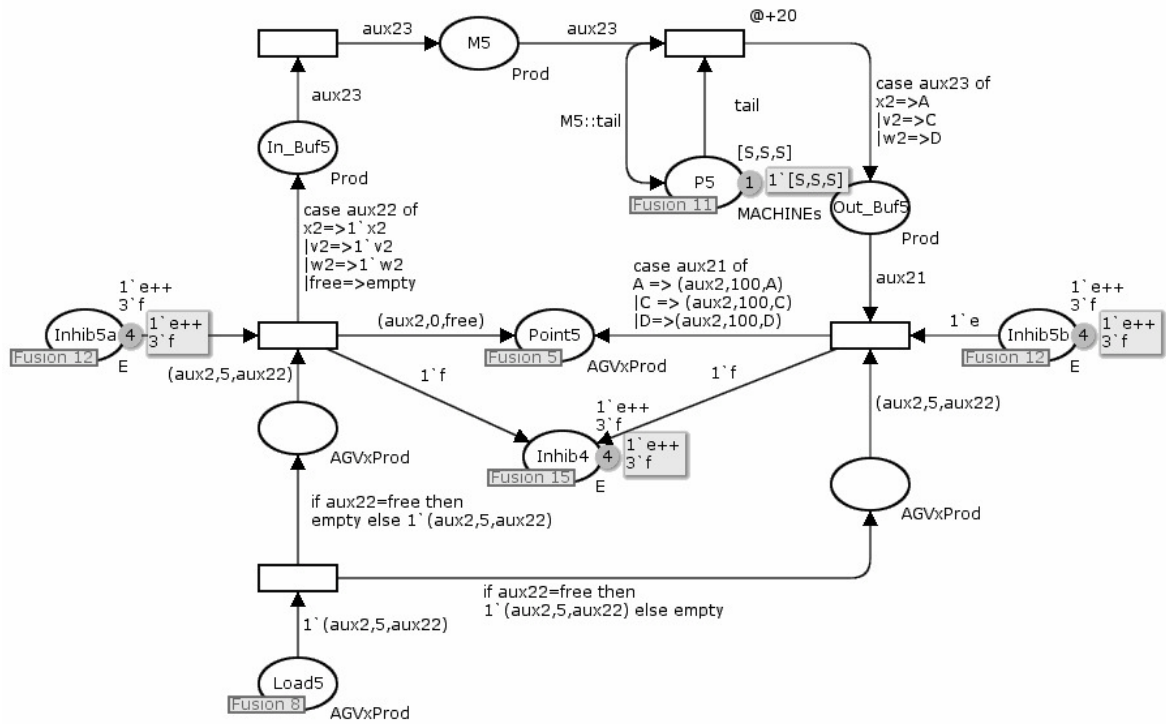


Figura 5.11 – Módulo Sub5.

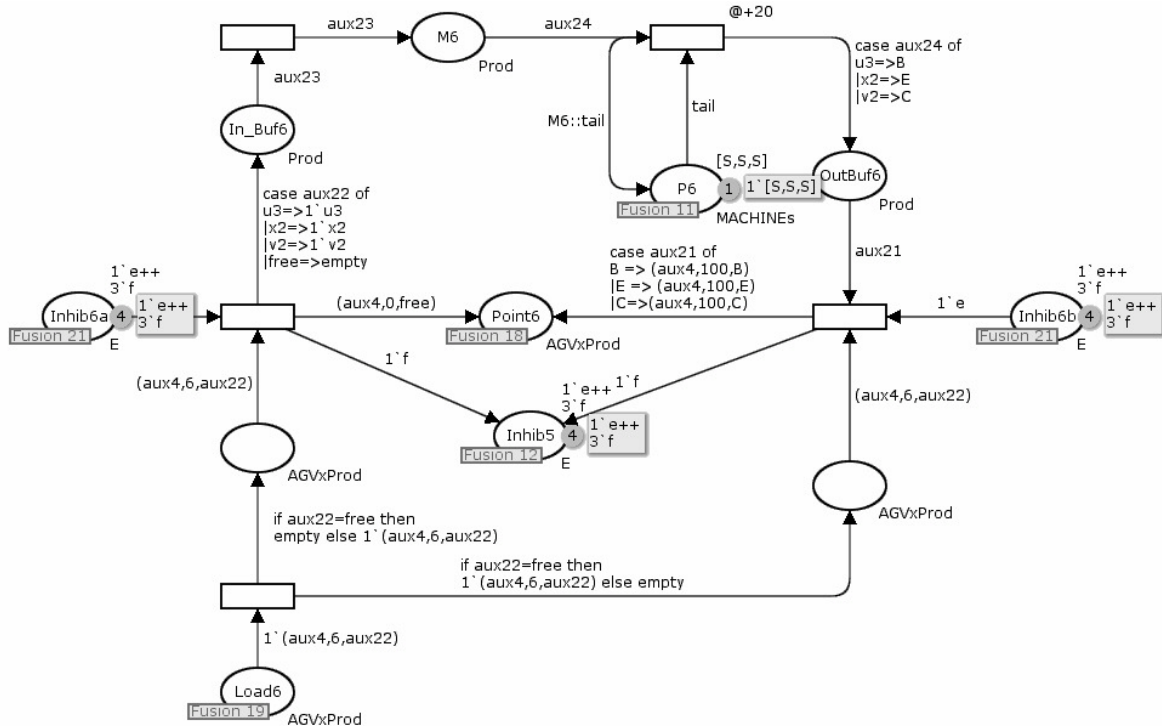


Figura 5.12 – Módulo Sub6.

Com relação aos módulos **Sub** que possuem os lugares sem nome do tipo **Prod** e **Route** do tipo **INT**, módulos referentes a eles deve estar presente no modelo. Neste modelo, módulos **Sub*.1** representam esses módulos. O lugar do tipo **Prod** presente tanto em **Sub** quanto em **Sub*.1** recebe uma marca do tipo **Prod** referente ao produto que acabou de ser processado na máquina. Conforme a marca presente nesse lugar, uma marca do tipo **E** é criada em um dos lugares sem nome de tipo **E** fazendo com que apenas uma das transições abaixo desses lugares seja habilitada. Dessa forma uma marca do tipo **INT** sairá da cabeça da lista de algum lugar de nome **Route_***. Essa lista possui marcas do tipo **INT** referentes à próxima máquina em que o produto deve ser processado. Essa lista também é criada conforme a simulação desejada e segue o plano de produção da peça. A marca gerada em **Route** é usada como referência para se compor a parte **INT** da tripla que forma o **AGV**. **AGV** este que está em **Sub** e será colocado em **Point**. Neste modelo as máquinas 1, 2, 3 e 4 possuem tal característica e portanto possuem seus módulos adicionais.

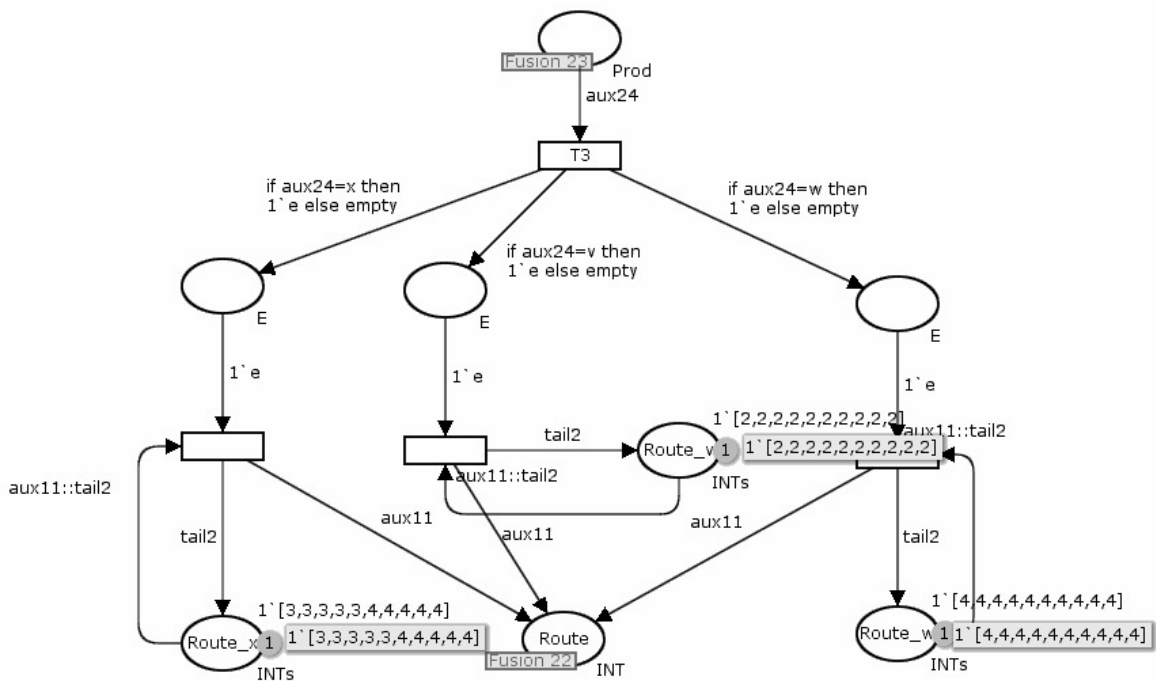


Figura 5.13 – Módulo Sub1.1.

Na figura 5.13 pode-se verificar um exemplo de módulo **Sub*.1**. A máquina referente a este modelo processa três tipos de produtos: x, v e w. Isso pode ser observado pelos arcos que saem da transição **T3**, pois cada arco possui inscrições referentes aos produtos. A informação de que produto está sendo processado na

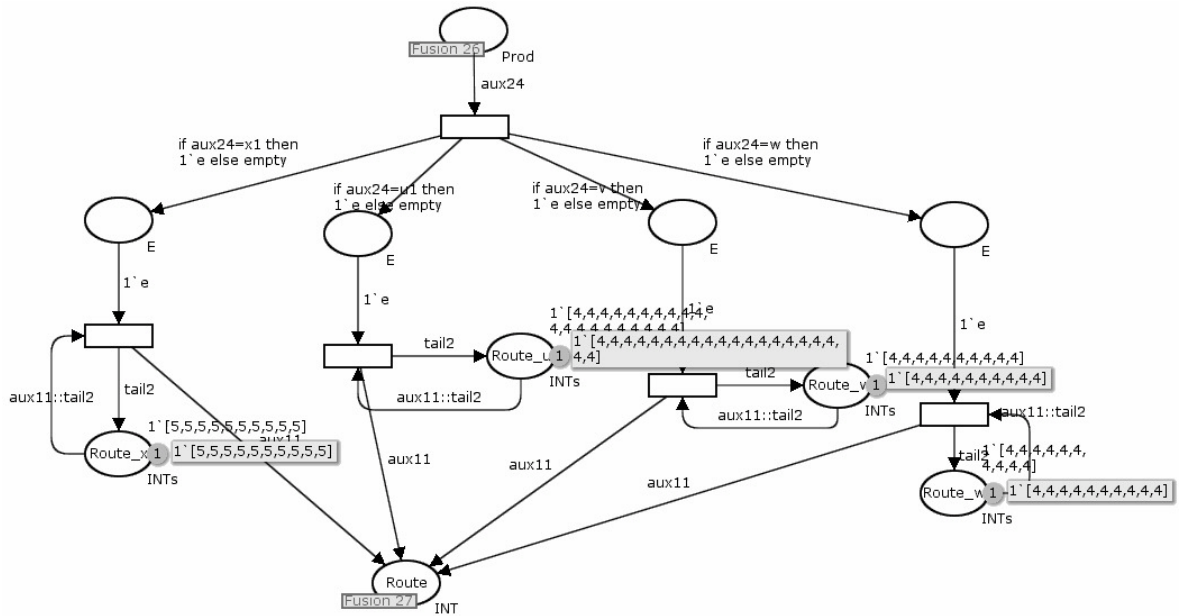


Figura 5.15 – Módulo Sub3.1.

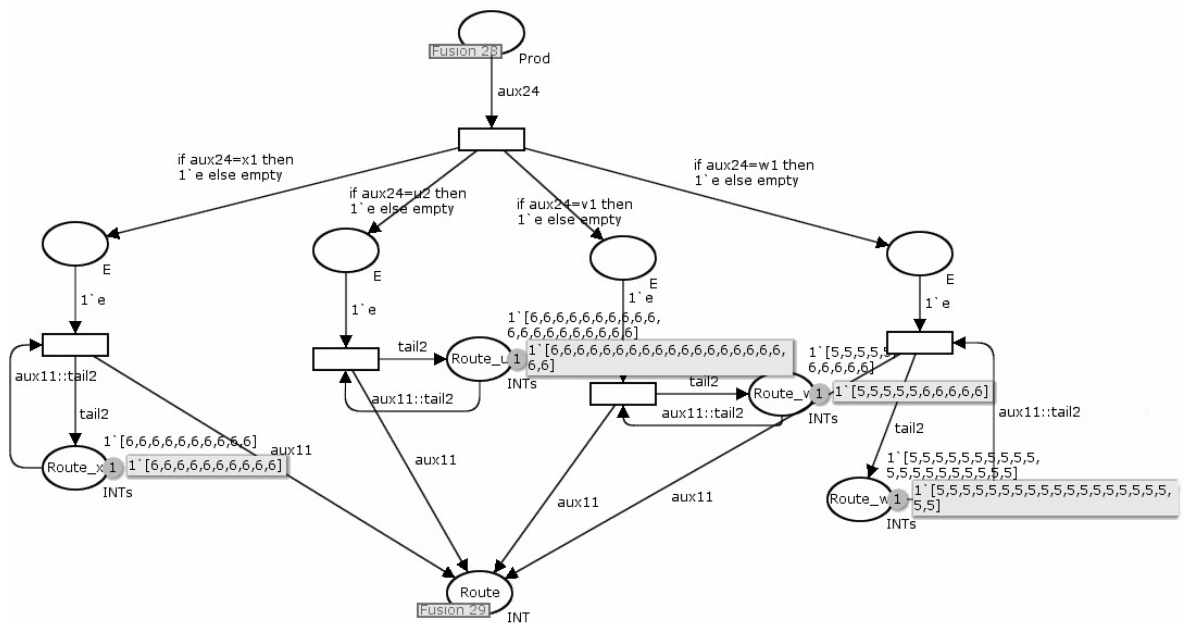


Figura 5.16 – Módulo Sub3.1.

- **Sexta Etapa**

A ligação entre módulos segue os mesmos princípios definidos para uma VPN. A ligação de um lugar virtual e um lugar alvo (não virtual) pode ser definido como:

$$(vp_{ali \rightarrow bli}) \oplus (p_{bli}) = (vp_{ali \rightarrow bli}, p_{abli})$$

Onde:

$vP_{ali \rightarrow bli}$ = lugar virtual p_i do módulo a tendo um alvo p_i em um módulo b .
 p_{bli} = lugar do módulo b .
 p_{abli} = novo lugar após a tarefa de ligação (de p_i).

O primeiro módulo é aqui chamado de módulo a . O módulo geral, que representa o FMS, é chamado b . Assim, todo lugar **Load** é um lugar virtual no módulo a que tem um lugar correspondente em b . Assim, o nó resultante na rede resultante é:

$$(vLoad_{ali \rightarrow bli}) \oplus (Load_{bli}) = (vLoad_{ali \rightarrow bli}, Load_{abli})$$

Onde:

$vLoad_{ali \rightarrow bli}$ = lugar virtual $Load_i$ do módulo a tendo um alvo $Load_i$ em um módulo b .
 i = a letra que representa o lugar de um tipo de produto (u,v,w,y ou z).
 $Load_{bli}$ = place of the module b .
 $Load_{abli}$ = novo lugar após a tarefa de ligação (de $Load_i$).

Então, lugares **Load** são virtuais no módulo a que têm um lugar correspondente em b . Os nós resultantes são como segue:

$$(vLoad_{ala \rightarrow bla}) \oplus (Load_{bla}) = (vLoad_{ala \rightarrow bla}, Load_{abla}) \text{ e } (vLoad_{alb \rightarrow bla}) \oplus (Load_{blb}) = (vLoad_{alb \rightarrow blb}, Load_{ablb})$$

Onde:

$vLoad_{ala \rightarrow bla}$ = lugar virtual $Load$ do módulo a tendo um alvo $Load$ em um módulo b .
 $Load_{bla}$ = lugar do módulo b .
 $Load_{abla}$ = novo lugar após a tarefa de ligação.
 $vLoad_{alb \rightarrow blb}$ = lugar virtual $Load$ do módulo a tendo um alvo $Load$ em um módulo b .
 $Load_{blb}$ = lugar do módulo b .
 $Load_{ablb}$ = novo lugar após a tarefa de ligação.

Após terminado o trabalho de ligação, o modelo final está completamente inter-travado e pode-se seguir para as próximas etapas.

- **Sétima Etapa**

Na simulação foi considerado um lote de vinte matérias-primas de cada tipo. Os tipos são x, u, v e w. Ao final espera-se ter dez produtos do tipo A, vinte do tipo B, vinte do tipo C, vinte do tipo D e dez do tipo E. O objetivo deste etapa é chegar ao final da simulação, ou seja, ter todos os produtos acabados e depositados no armazém. Durante a produção, um produto segue uma rota conforme seu plano de produção. Contudo, podem existir rotas alternativas para um mesmo tipo de produto. Isso acontece quando

duas ou mais máquinas podem processar um produto em uma mesma etapa de produção. Essas rotas podem ser observadas no início da seção na figura 5.2.

No caso de rotas alternativas durante a produção de uma peça, foi considerado um caminho em metade dos casos e o outro caminho na outra metade. Assim pôde-se testar todas as possibilidades. Entretanto, isso poderia ser alterado caso queira-se simular por exemplo duas máquinas com a mesma função, porém com tempos de processamento distintos. Nesse caso seria necessário desviar a maior parte das peças para a máquina mais rápida.

- **Oitava Etapa**

Nesta etapa é feita a inserção de tempos. Como modelo também contempla a existência de tempos das ações, a simulação que foi feita pela ferramenta pode ser considerada mais próxima do real do que uma modelagem que não o considera. A modelagem foi feita através de várias etapas. A cada nova simulação pôde-se notar novos problemas que foram sendo corrigidos na medida em que o modelo foi sendo concebido. Ao final pôde-se verificar que o sistema ficou livre de problemas para o sistema e plano de produção hipoteticamente considerados.

6

CONCLUSÃO

No intuito de otimizar a produção industrial, sistemas AMS são aplicados de maneira cada vez mais abrangente no chão de fábrica das indústrias. Porém, modelos de AMS não são capazes de cumprir tal propósito se modelados de maneira ruim. Até mesmo bons modelos podem ser otimizados caso sejam feitas análises e melhoramentos de alguns sistemas individuais e seus elementos. Existem vários fatores que devem ser levados em consideração quando se pretende modelar um sistema AMS.

Este trabalho apresentou esses fatores de maneira que se pudesse conhecer as vantagens e desvantagens das diversas alternativas possíveis de serem implementadas. Dentre as opções mostradas, algumas escolhas foram feitas baseadas em argumentos devidamente descritos.

Como dito no início, a proposta do trabalho era a de apresentar um método de modelagem de AMS através de uma seqüência de etapas. No capítulo 4 foram apresentadas as etapas que compõem o método. Cada etapa possui a descrição de como ela deve ser executada e o que se espera obter após sua aplicação. O propósito de dividir o método em etapas é facilitar a compreensão do leitor, fazendo com que a sua descrição seja a mais didática possível.

Considerando um modelo hipotético de FMS de seis máquinas, três AGV's, armazém de peças e *buffers* locais, o método pôde ser aplicado. Durante a modelagem,

as etapas foram surgindo de forma natural. Por fim chegou-se a um modelo que atendeu às requisições de produção de alguns produtos arbitrários.

Durante as etapas existem várias situações onde devem ser feitas simulações. Isso é possível pelo uso do *software* CPN Tools que, através de representações gráficas dos elementos, contempla essa funcionalidade. Devido ao uso do CPN Tools, o desempenho da modelagem limita-se, em alguns casos, às funcionalidades do *software*. Porém, as maiores dificuldades ficam a cargo do arquiteto que pretende elaborar seu modelo. Durante a elaboração do método foram observadas algumas delas.

Em um primeiro momento, a área de carga e descarga apresentava *deadlock*, por isso foram consideradas áreas com capacidade para mais de um veículo. Outro problema foi observado, no momento da simulação. Existia a dificuldade de colocar as peças para serem montadas em uma dada seqüência. A solução para isso foi a criação de lugares específicos para guardar seqüências. Para tentar aproximar o modelo a um sistema real os tempos de movimentação e produção tiveram que ser considerados. Essa preocupação adiciona um cuidado a mais no momento de definição dos elementos que compõem o modelo.

O modelo final gerado atendeu às requisições de um plano de produção e deixou margem para que mudanças possam ser feitas, de maneira que diferentes planos possam ser testados. Outras variáveis que apresentam bastante flexibilidade de mudança sem grandes alterações do modelo são os tempos das ações, roteiros de fabricação, rotas de deslocamento e número de produtos a serem fabricados.

O método proposto baseou-se na simulação como forma de validação do modelo, além de também utilizá-la durante a concepção do próprio método. No entanto podem ser aplicadas diversas formas analíticas de validação, já que o modelo foi desenvolvido tendo como ferramenta as redes de Petri.

Este trabalho deixa em aberto estudos relativos a paradigmas de multi-agentes e holônico. Apesar de tais paradigmas terem sido descritos no trabalho, estes não foram aplicados, pois além de ser necessário um amadurecimento maior envolta dos conceitos envolvendo os dois paradigmas, não se observou necessidade para esta primeira

abordagem do problema. Ainda assim, alguns desses paradigmas podem ser utilizados considerando apenas pequenas alterações no método proposto.

Estudos complementares possuem grande potencial para serem considerados como trabalhos futuros. Formas de melhorar o desempenho através de análises matemáticas parecem ter grande relação com o trabalho. O CPN Tools possui a opção de gerar um arquivo contendo todos os disparos de transição durante um processo de simulação apontando os passos e os tempos correspondentes. Através desse arquivo pode-se fazer uma análise de desempenho do modelo inter-travado, medindo a eficiência dos passos intermediários.

Como o modelo possui variáveis de seqüenciamento, também poderiam ser contempladas análises de outras fontes de medida de desempenho do sistema, como tempo de utilização de máquina, atraso médio de atendimento a pedidos, etc.

REFERÊNCIAS

ANTSAKLIS, P. J.; NERODE, A. **Hybrid Control Systems: An Introductory Discussion to the Special Issue.** IEEE Transactions on Automatic Control, v 43, n 4, Apr, 1998, p 457-460.

BAKER, K.R., **Introduction to Sequencing and Scheduling.** Wiley. 1974.

BASILE, FRANCESCO; CHIACCHIO, PASQUALE; VITTORINI, VALERIA; MAZZOCCA, NICOLA. **Modeling and logic controller specification of flexible manufacturing systems using behavioral traces and Petri net building blocks.** Journal of Intelligent Manufacturing, 15, 351-371, 2004.

BECK, CAROLYN L.; KROGH, BRUCE H.. **Models for Simulation and Discrete Control of Manufacturing Systems.** Proceedings - 1986 IEEE International Conference on Robotics and Automation. San Francisco, CA, EUA.

BURKHARD, H. D. **Fairness and control in multi-agent systems.** Theoretical Computer Science, v 189, n 1-2, Dec 15, 1997, p 109-127.

CHAN, FELIX T. S.; CHAN, H. K. **Analysis of dynamic control strategies of an FMS under different scenarios.** Robotics and Computer-Integrated Manufacturing, v 20, n 5, October, 2004, p 423-437.

DESROCHERS, A. A.; DEAL, T. J.; FANTI, M. P. **Complex-Valued Token Petri Nets.** IEEE Transactions on Automation Science and Engineering, v 2, n 4, October, 2005, p 309-318.

DESROCHERS, ALAN A.; AL-JAAR, ROBERT Y. **Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis.** 1995.

EZPELETA, J.; COLOM, J. M. **Automatic synthesis of colored Petri nets for the control of FMS.** IEEE Transactions on Robotics and Automation, v 13, n 3, Jun, 1997, p 327-337.

FANTI, M. P.; GIUA, A.; SEATZU, C. **A deadlock prevention method for railway networks using monitors for colored Petri nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 2, 2003, p 1866-1873.

FISCHER, K. **Agent-based design of holonic manufacturing systems.** *Robotics and Autonomous Systems*, v 27, n 1, Apr, 1999, p 3-13. Praga, República Tcheca.

FRIEDRICH, L. F. **Uma Abordagem Distribuída no Desenvolvimento e Implementação do Software de Controle de Chão-de-Fábrica em Sistemas de Manufatura Celular.** Tese (doutorado), 1996. Universidade Federal de Santa Catarina, SC, Brasil.

GI, BUM LEE; HAN, ZANDONG; JIN, S. LEE. **Automatic generation of ladder diagram with control Petri Net.** Journal of Intelligent Manufacturing, 15, 245-252, 2004.

GIGLIO, DAVIDE; PAOLUCCI, MASSIMO. **Agent-Based Petri Net models for AGV management in manufacturing systems.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 4, 2001, p 2457-2462.

GRAVES, S.C.. **A Review of Production Scheduling,** Operations Research, vol. 29, n. 4, p. 646-675, 1981.

GUPTA, M.C.; CHENG, T. C. E.. **Survey of scheduling research involving due date determination decisions.** European Journal of Operational Research, Elsevier, vol. 38(2), pages 156-166, January. 1989.

HRUZ, B. **Solution of the manufacturing transport control using Petri nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 5, 2003, p 4736-4741. Washington, DC, EUA.

HSIEH, F. S. **Collaborative timed petri net for holonic process planning.** Proceedings of the American Control Conference, v 1, 2003, p 344-349. Denver, CO, EUA.

JEHNG, WERN KUEIR; PENG, SHIH SEN; ZHOU, MENG CHU. **Petri Net Construction and Analysis of Automated Sequential Manufacturing Systems.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 4, 2001, p 2469-2474.

JENG, M. D.; DICESARE, F. **A review of synthesis techniques for Petri nets.** Proceedings of the Rensselaer's 2nd International Conference on Computer Integrated Manufacturing. Troy, NY, EUA

KAMATH, M.; VISWANADHAM, N. **Applications of Petri Net Based Models in the Modelling and Analysis of Flexible Manufacturing Systems.** Proceedings - 1986 IEEE International Conference on Robotics and Automation, 1986, p 312-317. San Francisco, CA, EUA.

KATO, E. R. R.; MORANDIN JR. **A complex automatic manufacturing systems control modeling approach using Petri Nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 4, 2001, p 2475-2480.

KATO, E. R. R.; MORANDIN JR., O.; POLITANO, P. R.; CAMARGO, H. A. **A control design approach for complex automatic manufacturing systems.** Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference. 2001.

KATO, E. R. R.; MORANDIN JR., O.; POLITANO, P. R.; CAMARGO, H. A. **A modular modeling approach for CNC machines control using Petri Nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 5, 2000, p 3147-3152.

KLEIN, S.; WENG, X.; FREY, G.; LESAGE, J. J.; LITZ, L.. **Controller Design for an FMS using Signal Interpreted Petri Nets and SFC**. Proceedings of the American Control Conference. 2002.

KLEIN, S.; FREY, G.; MINAS, M.. **PLC Programming with Signal Interpreted Petri Nets**. Proceedings of the ICATPN 2003, Eindhoven (The Netherlands), LNCS 2679, p. 440-449, Springer Verlag, 2003.

KOESTLER, A.; SMYTHIES, J.R.. **Hutchinson Beyond reductionism**. - 1969

KOTAK, D.; BARDI, S.; GRUVER, W. A.; ZOHREVAND, K.. **Comparison of Hierarchical and Holonic Shop Floor Control using a Virtual Manufacturing Environment**. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 3, 2000, p 1667-1672 IEEE. 2000.

LEE, J. S.; HSU, P. L. **A Petri-net approach to hierarchical supervision for remote-controlled processes**. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 2, 2003, p 1880-1885. Washington, DC, EUA.

LEITAO, P.; RESTIVO, F. **ADACOR: A holonic architecture for agile and adaptive manufacturing control**. Computers in Industry, v 57, n 2, February, 2006, p 121-130.

LI, Z. W.; ZHOU, M. C. **Elementary Siphons of Petri Nets and Their Application to Deadlock Prevention in Flexible Manufacturing Systems**. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans., v 34, n 1, January, 2004, Deadlock Resolution in Computer-Integrated Systems, p 38-51.

MAIONE, G.; NASO, D. **A genetic approach for adaptive multiagent control in heterarchical manufacturing systems**. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans., v 33, n 5, September, 2003, Collective Intelligence: Intelligence Through Multirobot/Multiagent Systems, p 573-588.

MARTINS, R.W.C.. **Uma abordagem de arquiteturas de controle para FMS**. Dissertação de mestrado, 2005. Universidade Federal de São Carlos, SP, Brasil.

MERTKE, T.; FREY, G.. **Formal verification of PLC-Programs generated from Signal Interpreted Petri Nets**. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 4, 2001, p 2700-2705.

MORANDIN JR., O. **Metodologia de modelagem de Sistemas Automatizados de Manufatura, utilizando Rede de Petri Virtual**. Tese (Doutorado); Escola de Engenharia de São Carlos, Universidade de São Paulo; agosto de 1999; 139 p.

MORANDIN JR., O.; KATO, E. R. R. **Virtual Petri Nets as a Modular Modeling Method for Planning and Control Tasks of FMS**. International Journal of Computer Integrated Manufacturing, v 18, n 2-3, March/May, 2005, p 100-106.

MURATA, T.. **Petri Nets: Properties, Analysis and Applications**. Proceedings of the IEEE, vol. 77, no. 4, 1989.

NARCISO, M. E.; PIERA, M. A. **Coloured petri net simulator: A generic tool for production planning**. IEEE Symposium on Emerging Technologies and Factory Automation, ETFA, v 2, 2001, p 139-147.

NKETSA, A.; VALLETE, R. **Rapid and modular prototyping-based Petri nets and distributed simulation for manufacturing systems**. Applied Mathematics and Computation (New York), v 120, n 1-3, May 10, 2001, p 265-278.

O'KANE, J. F. **A knowledge-based system for reactive schedulin decision-making in FMS**. Journal of Intelligent Manufacturing, 2000 11, 461-474.

OU-YANG, C.; LIN, J. S. **The development of a hybrid hierarchical/heterarchical shop floor control system applying bidding method in job dispatching**. Robotics and Computer-Integrated Manufacturing 14 (1998). p. 199-217.

PARK, E.; TILBURY, D. M.; KHARGONEKAR, P. P. **A modeling and analysis methodology for modular logic controllers of machining systems using Petri net formalism**. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, v 31, n 2, May, 2001, p 168-188.

PENG, S. S.; ZHOU, M. C. **Ladder diagram and Petri-net-based discrete-event control design methods.** IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, v 34, n 4, November, 2004, Learning: Advances in Multimedia Communications, Information Processing, and Education, p 523-531.

PROTH, J. M.; XIE, X.. **Petri Nets: A Tool for Design and Management of Manufacturing Systems.** 1996.

SCHOOP, R.; NEUBERT, R.; COLOMBO, A. W. **A multiagent-based distributed control platform for industrial flexible production systems.** IECON Proceedings (Industrial Electronics Conference), v 3, 2001, p 279-284. Denver, CO, EUA.

TEMPELMEIER, H.; KUHN, H.. **Flexible manufacturing systems: decision support for design and operation.** International Journal of Production Research, v 33, n 9, Sept, 1995, p 2649.

UZAM, M., JONES, A. H.; YUCEL I. **Using a Petri-Net-Based Approach for the Real-Time Supervisory Control of an Experimental Manufacturing System.** International Journal Manufacturing Technology (2000) 16:498-515.

VALCKENAERS, P.; VAN BRUSSEL, H.; WYNS, J.; BONGAERTS, L.; PEETERS, P. **Designing Holonic manufacturing systems.** Robotics and Computer-Integrated Manufacturing, v 14, n 5-6, Oct-Dec, 1998, p 455-464.

VALETTE, R.; CARDOSO, J.; DUBOIS, D. **Monitoring manufacturing systems by means of Petri nets with imprecise markings.** Proc IEEE Int Symp Intell Control 1989, 1989, p 233-238. Albany, NY, EUA.

VELASCO, J. R.; GONZALEZ, J. C.; MAGDALENA, L.; IGLESIAS, C. A. **Multiagent-based control systems: a hybrid approach to distributed process control.** *Control Engineering Practice*, v 4, n 6, Jun, 1996, p 839-845.

VILLAROEL, J. L.; MARTINEZ J.; SILVA, M. **GRAMAN: A Graphic System for Manufacturing System Design.** IMACS Symposium on System Modelling and Simulation (1989, Elsevier Science Publ.) 311-316

WANG, L. **Integrated design-to-control approach for holonic manufacturing systems.** Robotics and Computer-Integrated Manufacturing, v 17, n 1, Feb, 2001, p 159-167.

WANG, Z.; ZHANG, J.; CHAN, F. T. S. **A hybrid Petri nets model of networked manufacturing systems and its control system architecture.** Journal of Manufacturing Technology Management, v 16, n 1, Logistics and Supply Chain Management with Artificial Intelligence Techniques - Part 2, 2005, p 36-52.

WENG, X.; LITZ, L.. **Model checking of Signal Interpreted Petri Nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 4, 2001, p 2748-2752. Tucson, AZ, EUA.

WU, W.; SU, H.; CHU, J.; ZHAI, H.. **Hierarchical Control of DES Based on Colored Petri Nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 3, 2001, p 1571-1576. Tucson, AZ, EUA.

WU, W.; SU, HONGYE; CHU, JIAN; ZHAI, HAIFENG. **Hierarchical control of DES based on colored Petri nets.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 3, 2001, p 1571-1576. Tucson, AZ, EUA.

YAN, H. S.; WANG, N. S.; CUI, X. Y.; ZHANG, J. G.. **Modeling, scheduling and control of flexible manufacturing systems by extended high-level evaluation Petri nets.** IIE Transactions (Institute of Industrial Engineers),v 29, n 2, Feb, 1997, p 147-158.

ZHOU, M.C.; DICESARE, F.; DESROCHERS, A. A.. **A hybrid methodology for synthesis of Petri net models for manufacturing systems.** IEEE Transactions on Robotics and Automation, v 8, n 3, Jun, 1992, p 350-361.

ZIMMERMANN, A.; BODE, S.; HOMMEL, G. **Performance and Dependability Evaluation of Manufacturing Systems Using Petri Nets.** 1st Workshop on Manufacturing Systems and Petri Nets, 1996.

ZIMMERMANN, A.; HOMMEL, G. **Modelling and Evaluation of Manufacturing Systems Using Dedicated Petri Nets.** International Journal of Advanced Manufacturing Technology, v 15, n 2, 1999, p 132-138.

ZOUARI, B. **High-level Petri net approach for supervisory control.** Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 2, 2003, p 1161-1166. Washington, DC, EUA.

ZUBEREK, W. M. **Cluster tools with chamber revisiting - Modeling and analysis using timed Petri nets.** IEEE Transactions on Semiconductor Manufacturing, v 17, n 3, August, 2004, p 333-344.

APÊNDICE I – REDE DE PETRI VIRTUAL

Uma Rede de Petri Virtual é descrita como sendo uma quintupla, $VPN = (vP, vT, vF, vW, vM_0)$ onde:

$VP = vP \cup P$, é um conjunto finito de lugares virtual;

$VT = vT \cup T$, é um conjunto finito de transições virtual;

$VF = vF \cup F$, é um conjunto finito de arcos virtual;

$VW : VF$, é o peso da função virtual;

$VM_0 : vP \rightarrow \{0, 1, 2, 3, \dots\}$ é o estado inicial virtual;

$VP = \{vp1+n, vp2+n, \dots, vpp\}$ é o conjunto finito de lugares virtuais;

$VT = \{vt1+m, vt2+m, \dots, vtq\}$ é o conjunto finito de transições virtuais;

$VF \subseteq (VP \times VT) \cup (VT \times VP) \cup (P \times VT) \cup (VT \times P) \cup (VP \times T) \cup (T \times VP)$ é o conjunto de arcos virtuais;

$VW : VF \rightarrow \{1, 2, 3, \dots\}$ é a função peso virtual;

$VM_0 : VP \rightarrow \{0, 1, 2, 3, \dots\}$ é o estado inicial virtual;

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

$T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;

$F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos;

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

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

Além disso, considera-se:

$\bullet p_i$ = conjunto de transições de entrada do lugar p_i .

$\bullet vp_j$ = conjunto de transições de entrada do lugar virtual vp_j .

$v(\bullet vp_j)$ = conjunto de transições virtuais de entrada do lugar virtual vp_j .

$v(\bullet p_i)$ = conjunto de transições virtuais de entrada do lugar p_i .

$p_i \bullet$ = conjunto de transições de saída do lugar p_i

$vp_j \bullet$ = conjunto de transições de saída do lugar virtual vp_j

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

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

Esse conjunto de nós pode ser visto na figura A1.1.

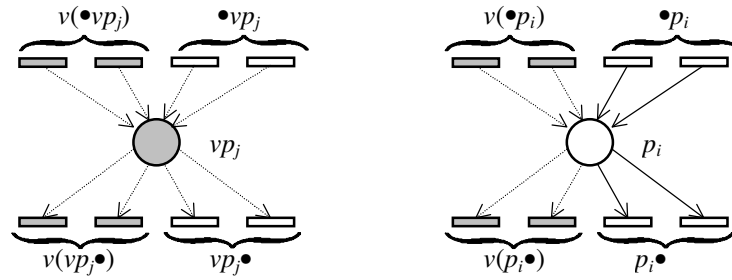


Figura A1.1 - Representação de nós (lugares e transições) em uma PN Virtual. (Morandin, Jr., O, 1999)

Além disso, considera-se:

- $\bullet t_i$ = conjunto de lugares de entrada da transição t_i .
- $\bullet vt_j$ = conjunto de lugares de entrada da transição virtual vt_j .
- $v(\bullet vt_j)$ = conjunto de lugares virtuais de entrada da transição virtual vt_j .
- $v(\bullet t_i)$ = conjunto de lugares virtuais de entrada da transição t_i .
- $t_i \bullet$ = conjunto de lugares de saída da transição t_i .
- $vt_j \bullet$ = conjunto de lugares de saída da transição virtual vt_j .
- $v(vt_j \bullet)$ = conjunto de lugares virtuais de saída da transição virtual vt_j .
- $v(t_i \bullet)$ = conjunto de lugares virtuais de saída da transição t_i .

Uma vez definida a rede de Petri virtual, essa rede é usada na construção de cada módulo. Então cada módulo é uma PN com alguns elementos virtuais (lugares, transições e arcos). Os elementos da rede, para identificarem o módulo a que pertencem, usam para um lugar i , a notação p_{ali} e, para uma transição, a notação t_{ali} . Na figura A1.2 são mostradas as formas de representação desses elementos, com uma pequena alteração.

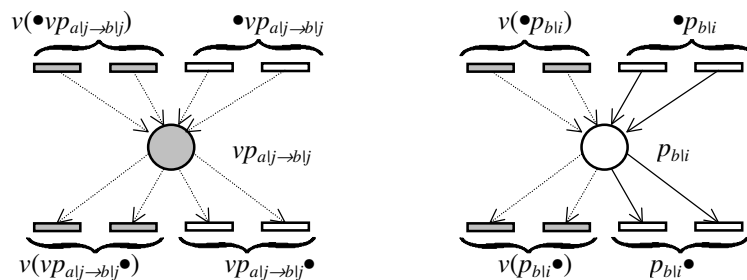


Figura. A1.2 - Representação de módulos com elementos virtuais. (Morandin, Jr., O, 1999)

A notação $vp_{alj \rightarrow bly}$ é uma forma de representação de um lugar virtual vp_i pertencente ao módulo vPN_a e esse lugar tem um alvo p_i (com a mesma identificação) pertencente ao módulo vPN_b . Assim, o nó virtual (lugar e transição) é aquele que tem um nó alvo em outro módulo, ou seja, o nó virtual é aquele que tem uma representação em um outro módulo do modelo geral, indicando representar a mesma situação (lugar ou transição). Dessa forma, o $v(\bullet vp_{alj \rightarrow bly})$ é um conjunto de transições virtuais de entrada que têm alvos mostrados a seguir:

$$v(\bullet vp_{alj \rightarrow bly})_{a \rightarrow w} = (vt_{alj \rightarrow clj}, vt_{alj \rightarrow dlj}, \dots, vt_{alk \rightarrow elk}) = v(\bullet vp_{alj \rightarrow bly})_{a \rightarrow b} \cup v(\bullet vp_{alj \rightarrow bly})_{a \rightarrow (w-b)}$$

onde:

w = conjunto de elementos alvo para montagem.

A montagem de dois módulos é realizada considerando-se os nós virtuais. Todos os nós virtuais que são alvos uns dos outros, ou que têm alvos em outros elementos serão representados por um único novo nó no modelo geral. Os demais nós permanecerão da maneira como eram representados.

De uma maneira geral devem-se associar os nós, considerando-se uma associação por vez, ou seja, associando-se dois módulos por vez até que se obtenha a associação entre todos os módulos do modelo.

O procedimento é “unir” a representação de dois nós criando-se um novo nó que indique sua origem e de tal maneira que suas entradas e saídas (lugares ou transições) sejam preservadas. As possíveis situações de associação podem ser descritas na forma de uma equação para implementação de um algoritmo de associação.

Uma primeira situação é a associação entre um lugar virtual e um lugar alvo (não virtual) que é associado da seguinte maneira:

$$(vp_{ali \rightarrow bli}) \oplus (p_{bli}) = (vp_{ali \rightarrow bli}, p_{abli})$$

onde:

$vp_{ali \rightarrow bli}$ = lugar virtual p_i do módulo a que tem um alvo p_i em um módulo b.

p_{bli} = lugar do módulo b . Esse lugar é o alvo de vp_{ali} .

p_{abli} = lugar resultante da associação de p_i .

$$\bullet p_{abli} = \bullet vp_{ali \rightarrow bli} \cup \bullet p_{bli} \cup v(\bullet p_{bli}) \cup v(\bullet vp_{ali \rightarrow bli})_{a \rightarrow (w-b)} \cup \cup (v(\bullet vp_{ali})_{a \rightarrow b} \oplus v(\bullet vp_{bli})_{b \rightarrow a})$$

$$p_{abli} \bullet = vp_{ali \rightarrow bli} \bullet \cup p_{bli} \bullet \cup v(p_{bli} \bullet) \cup v(vp_{ali \rightarrow bli} \bullet)_{a \rightarrow (w-b)} \cup \cup (v(vp_{ali} \bullet)_{a \rightarrow b} \oplus v(vp_{bli} \bullet)_{b \rightarrow a})$$

Pode-se mostrar graficamente essas associações, prevendo-se algumas possíveis situações, de acordo com o apresentado na figura A1.3.

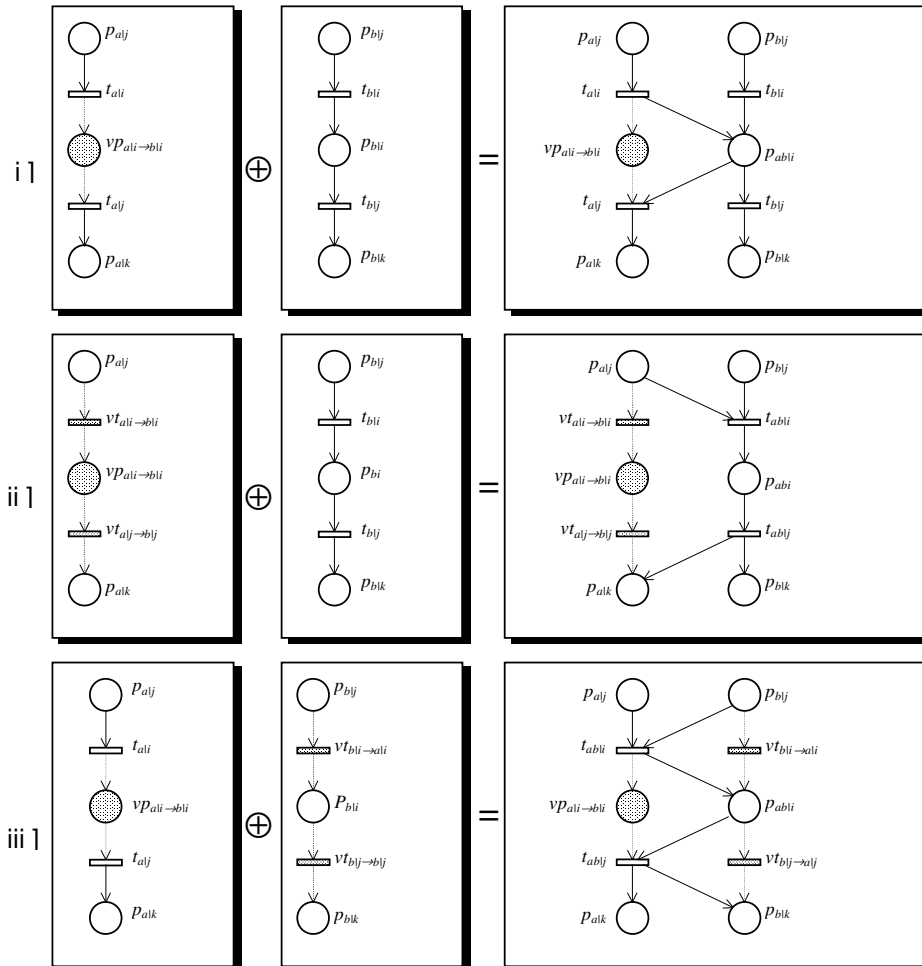


Figura A1.3 - Exemplos de associação entre módulos: i] um lugar virtual do módulo a, com alvo em um lugar (não virtual) do módulo b; ii] um lugar virtual e duas transições virtuais de um módulo a, com alvo em um lugar e duas transições de um módulo b; iii] um lugar virtual de um módulo a, com um alvo em um módulo b, e duas transições virtuais de um módulo b, com alvos em um módulo a. (Morandin, Jr., O, 1999)

Cabe destacar que, na figura representativa da associação dos módulos, embora continuem sendo apresentados os nós virtuais, eles não fazem parte do modelo associado. Eles estão colocados nas figuras, apenas com o objetivo de facilitar a interpretação da associação. Portanto, os modelos associados devem ter esses nós excluídos, assim como os arcos de entrada e saída a eles conectados.

Além dessas possibilidades, um outro tipo de associação pode ser exemplificado: a associação de dois nós virtuais. Dois nós virtuais também podem ser associados se eles são alvo de um outro. A associação de dois nós virtuais resulta em um novo nó.

O lugar virtual para um lugar virtual alvo é montado da seguinte maneira:

$$(vp_{ali \rightarrow bli}) \oplus (vp_{bli \rightarrow ali}) = (vp_{ali \rightarrow bli}, vp_{bli \rightarrow ali}, p_{abli})$$

onde:

$vp_{ali \rightarrow bli}$ = lugar virtual p_i do módulo a de tipo A que tem um alvo p_i em um módulo b de tipo B.

$vp_{bli \rightarrow ali}$ = lugar virtual p_i do módulo b de tipo B que tem um alvo p_i em um módulo a de tipo A.

p_{abli} = novo lugar resultante da associação de p_i .

$$\bullet p_{abli} = \bullet vp_{ali \rightarrow bli} \cup \bullet vp_{bli \rightarrow ali} \cup v(\bullet vp_{ali \rightarrow bli})_{a \rightarrow (w-b)} \cup v(\bullet vp_{bli \rightarrow ali})_{b \rightarrow (w-a)} \cup v(\bullet vp_{ali \rightarrow bli})_{ab \rightarrow b} \cup v(\bullet vp_{bli \rightarrow ali})_{ab \rightarrow a}$$

$$p_{abli} \bullet = vp_{ali \rightarrow bli} \bullet \cup vp_{bli \rightarrow ali} \bullet \cup v(vp_{ali \rightarrow bli} \bullet)_{a \rightarrow (w-b)} \cup v(vp_{bli \rightarrow ali} \bullet)_{b \rightarrow (w-a)} \cup v(\bullet vp_{ali \rightarrow bli})_{ab \rightarrow b} \cup v(\bullet vp_{bli \rightarrow ali})_{ab \rightarrow a}$$

Da mesma maneira como foi mostrado anteriormente, pode-se apresentar graficamente essas associações, prevendo-se algumas possíveis situações, de acordo com o apresentado na figura A1.4.

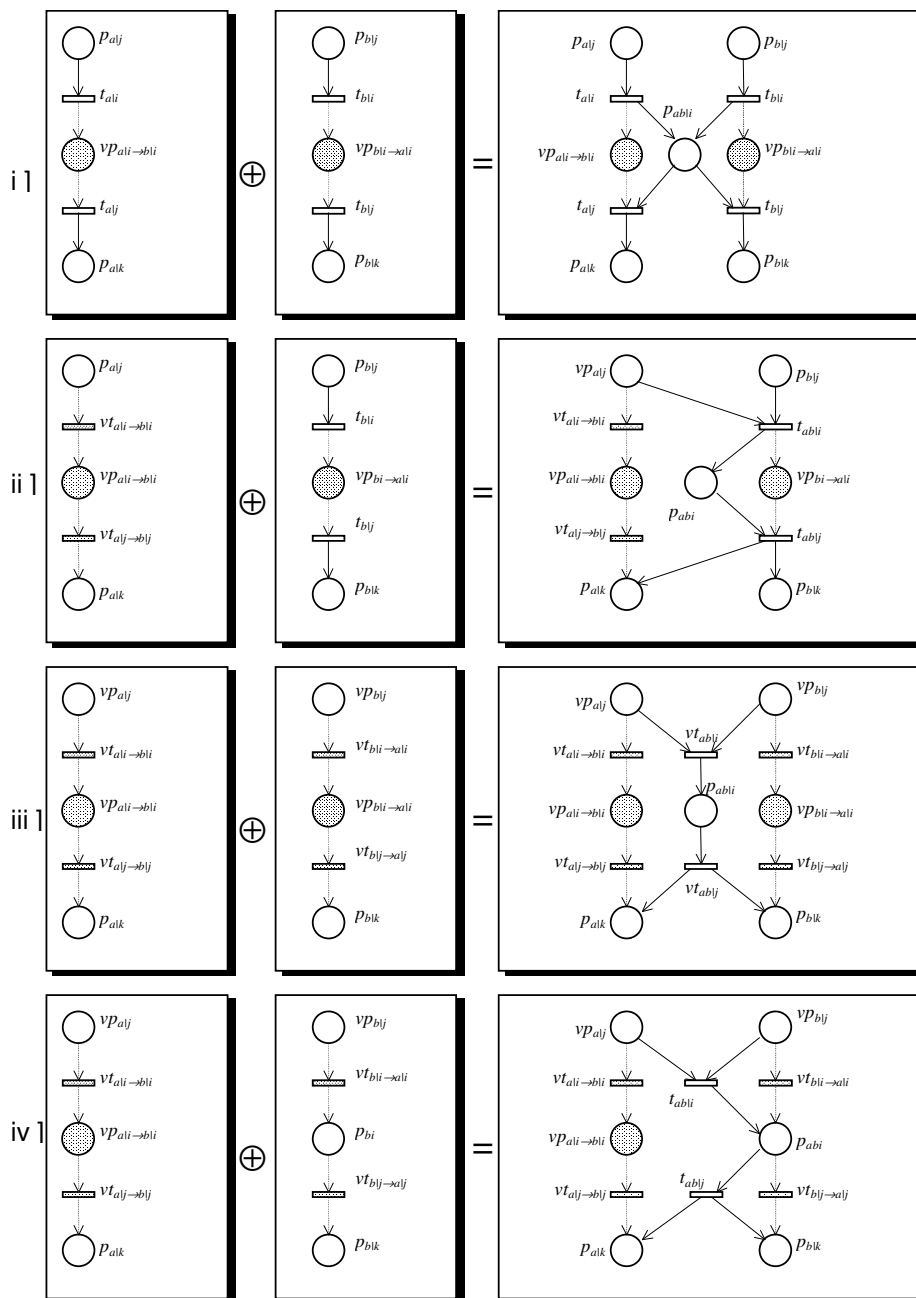


Figura A1.4 - Exemplos de associação entre módulos: i] um lugar virtual do módulo a, com alvo em um lugar virtual do módulo b; ii] um lugar virtual e duas transições virtuais de um módulo a, com alvo em um lugar virtual e duas transições (não virtuais) de um módulo b; iii] um lugar virtual e duas transições virtuais de um módulo a, com alvo em um lugar virtual e duas transições virtuais de um módulo b; iv] duas transições virtuais e um lugar virtual de um módulo a, com alvo em duas transições virtuais e um lugar (não virtual) em um módulo b. (Morandin, Jr., O, 1999)

A apresentação formal da montagem de dois módulos $VPN_a \oplus VPN_b$ é também uma quintupla $VPN_{ab} = (VP_{ab}, VT_{ab}, VF_{ab}, VW_{ab}, VM_{0ab})$ onde:

$$VP_{ab} = P_{va \rightarrow vb} \cup P_{vb \rightarrow va} \cup VP_{va \rightarrow vb} \cup VP_{vb \rightarrow va} \cup VP_a \cup VP_b$$

$$VT_{ab} = T_{va \rightarrow vb} \cup T_{vb \rightarrow va} \cup VT_{va \rightarrow vb} \cup VT_{vb \rightarrow va} \cup VT_a \cup VT_b$$

$$VF_{ab} = VF_{ab} \cup F_{ab}$$

$$VW_{ab} : VF_{ab}$$

$VM_{0/ab} : VP_{ab} \rightarrow \{0, 1, 2, 3, \dots\}$ é o estado inicial virtual;

$$VP_{ab} \cup VT_{ab} \neq \emptyset,$$

$VP = \{vp_{1+n}, vp_{2+n}, \dots, vp_p\}$ é o conjunto finito de lugares virtuais,

$VT = \{vt_{1+m}, vt_{2+m}, \dots, vt_q\}$ é o conjunto finito de transições virtuais,

$VF \subseteq (VP \times VT) \cup (VT \times VP) \cup (P \times VT) \cup (VT \times P) \cup (VP \times T) \cup (T \times VP)$ é o conjunto de arcos virtuais,

$VW : VF \rightarrow \{1, 2, 3, \dots\}$ é o peso da função,

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

$P_{va \rightarrow vb} \oplus P_{vb \rightarrow va} =$ conjunto de novos lugares.

APÊNDICE II – CPN TOOLS

Neste apêndice são considerados os elementos de redes de Petri conforme descrição do *software* CPN Tools. Restringe-se ainda ao conjunto de elementos usado no texto de dissertação.

Na figura A2.1 é exemplificada uma rede hipotética bem simples projetada usando-se o CPN Tools. Vale esclarecer que a maioria dos elementos podem sofrer alterações de tamanho em suas dimensões e as inscrições podem sofrer mudanças de posicionamento. Portanto, estão representados os elementos em uma configuração compatível com a utilizada ao longo do texto de dissertação.

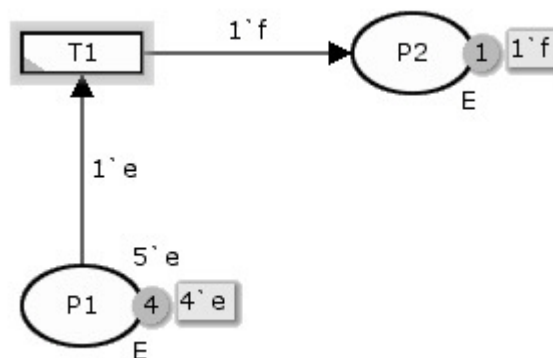


Figura A2.1 – Elementos básicos.

Lugares são representados por uma elipse. O nome do lugar pode ser colocado dentro da elipse mas também podem ser omitidos. Cada lugar admite somente um conjunto de cores ou tipo. Na figura é considerado um único conjunto chamado E. As marcas ou cores que fazem parte desse conjunto são e e f. O lugar deve estar relacionado a um conjunto. A indicação do conjunto é geralmente posicionada no canto inferior direito do lugar. Pode ser definida uma marcação inicial nos lugares. Essa marcação é posicionada no canto superior direito. Na figura, o lugar P1 possui inicialmente cinco marcas e sendo representadas por 5`e. A parte hachurada à direita de P1 mostra a quantidade de marcas presentes no lugar e suas cores em um momento

corrente. Pode-se observar que a figura não está representando a rede no momento inicial, pois o número de marcas corrente não é o mesmo que o número de marcas no momento inicial.

As transições são representadas por retângulos. Assim como nos lugares, o seu nome se encontra posicionado na parte central do elemento, caso não seja omitido. A transição T1 possui uma “aura” em sua volta. Isso acontece devido ao CPN Tools destacar as transições habilitadas para disparo.

Cada arco deve possuir uma inscrição. O arco que liga P1 a T1 possui a inscrição $1\ e$, que refere-se ao peso do arco. Neste caso uma marca e será consumida de P1 para o disparo da transição. O arco que liga T1 a P2 possui a inscrição $1\ f$, que refere-se à geração de uma marca f em P2 a cada disparo de T1. Pode-se verificar a existência de uma marca f em P2 pela informação posicionada à direita de P2.

A seguir, com base na figura A2.2, serão apresentadas características adicionais.

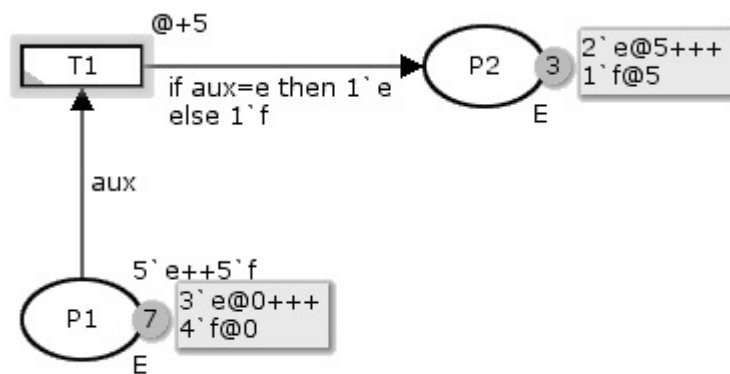


Figura A2.2 – Variáveis, estruturas lógicas e tempo.

A figura A2.2 segue a mesma idéia da figura 1, porém com conceitos mais elaborados. Como anteriormente, a posição da marcação inicial é no canto superior direito do lugar. Desta vez existem cinco marcas e e cinco marcas f inicialmente em P1. A informação à direita mostra que duas marcas e e uma marca f já foram consumidas. Ao invés de existir um peso definido no arco que liga P1 a T1, existe uma variável do tipo E chamada aux . Isso faz com que seja consumida aleatoriamente uma marca de P1. A estrutura lógica presente no arco que liga T1 a P2 faz a avaliação de aux , que pode

valer e ou f dependendo da marca consumida em $P1$. Conforme seu valor, é gerada a marca correspondente em $P2$. Se aux vale e , então é gerada uma marca e . Se aux vale f , então é gerada uma marca f . Pode-se verificar que a diferença de marcas presentes em $P1$ e sua própria marcação inicial é a mesma quantidade presente em $P2$.

Este exemplo possui uma representação de tempo. Apesar de o tempo também estar presente nas inscrições dos arcos, são considerados aqui somente a representação de tempo em transições. Essa representação está posicionada no canto superior direito de uma transição. Ela sempre é feita na forma $@+$ e o número de unidades de tempo que se deseja considerar para aquela transição. As marcas em $P1$ são mostradas com o símbolo $@$ seguido por 0 (zero). Ou seja, as marcas estão disponíveis em $P1$ do momento 0 em diante. Assim que a transição é disparada, uma marca é gerada em $P2$. Apesar de existir a informação de que a marca está presente em $P2$, elas só estarão, de fato disponíveis, no tempo 5. Por isso elas são representadas com $@5$. O CPN Tools sempre faz os disparos de transições com marcas disponíveis em um determinado tempo se todas os disparos referentes a tempos anteriores já tiverem sido feitos.

A figura A2.3 aborda características como composição de conjuntos, fusões de lugares e formas diferentes de inscrições.

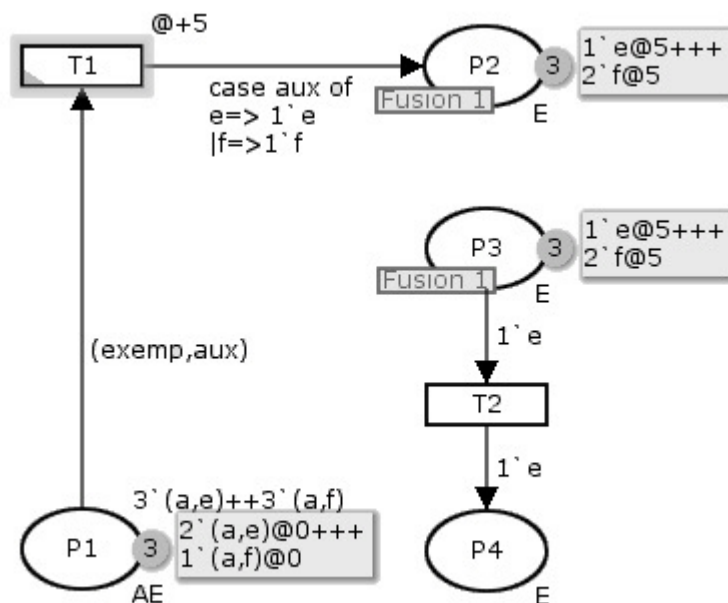


Figura A2.3 – Composição de conjuntos, fusões de lugares e inscrições.

Essa figura possui uma nova transição (T2) e dois novos lugares (P3 e P4). O primeiro ponto a ser observado é o novo conjunto (AE) atribuído a P1. AE refere-se a composição dos conjuntos A, que é formado por uma marca ou cor **a**, e o conjunto E, que é formado por **e** e **f**, da mesma forma que as figuras anteriores. Como P1 é do tipo AE, suas marcas devem ser uma composição entre marcas do tipo A e marcas do tipo E. A marcação inicial de P1 é composta por três marcas (**a,e**) e três marcas (**a,f**). Pode-se notar que três marcas já foram consumidas. A inscrição no arco que parte de P1 possui agora duas variáveis. Uma de tipo A (**exemp**) e outra de tipo E (**aux**). Cada variável “casa” com uma marca de seu tipo respectivo.

A inscrição presente no arco que parte de T1 e vai a P2 possui a mesma avaliação lógica das figuras anteriores. A avaliação baseia-se apenas na variável **aux**. Em um problema prático, dificilmente a variável **exemp** seria totalmente desprezada como aqui, contudo esse é apenas um exemplo didático. A vantagem dessa inscrição é que ela pode considerar várias alternativas sem a necessidade de se escrever muito.

Outra característica considerada é a possibilidade de fusão entre elementos. O CPN Tools contempla fusão entre lugares e transições, porém somente a fusão de lugares foi considerada neste trabalho. Os lugares que são fundidos com outros, possuem uma *tag* com o nome da fusão. Lugares com *tags* de nomes iguais são lugares fundidos. Na figura A2.3, os lugares P2 e P3 possuem a mesma *tag*, ou seja, apresentam um retângulo no canto inferior de um lugar com o mesmo nome, neste caso, **Fusion 1**. Apesar de possuírem nomes diferentes, os lugares podem ser vistos como sendo um só, já que toda marca que é gerada ou consumida em um deles é também gerada ou consumida no outro. Portanto marcas presentes em P2, também estão presentes em P3.

Apesar de P3 possuir uma marca **e** e a inscrição do arco até T2 indicar o consumo de uma marca para que o disparo possa ser realizado, T2 não dispara, já que a marca **e** presente em P3 só estará disponível no tempo 5. Quando as marcas estiverem disponíveis, a transição T2 será disparada e uma marca será gerada em P4, fazendo com que uma marca seja consumida em P3 e, conseqüentemente, em P2 também.