

**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**MODELAGEM E AVALIAÇÃO DE SISTEMAS
DINÂMICOS A EVENTOS DISCRETOS
UTILIZANDO REDE DE PETRI COLORIDA**

Por

JULIO CESAR DE LIMA LOPES

Recife, fevereiro de 2014

**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**MODELAGEM E AVALIAÇÃO DE SISTEMAS DINÂMICOS
A EVENTOS DISCRETOS UTILIZANDO REDE DE PETRI
COLORIDA**

JULIO CESAR DE LIMA LOPES

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de **Mestre em Engenharia Elétrica**.

**ORIENTADORA: PROFA. FERNANDA MARIA RIBEIRO DE
ALENCAR, PH.D.**

Recife, fevereiro de 2014

©Julio Cesar Lima Lopes, 2014

Catálogo na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

L864m Lopes, Julio Cesar de Lima.
 Modelagem e avaliação de sistemas dinâmicos a eventos discretos
 utilizando rede de petri colorida / Julio Cesar de Lima Lopes. - Recife: O
 Autor, 2014.
 86 folhas, il., gráfs.

 Orientadora: Profa. Dra. Fernanda Maria Ribeiro de Alencar.
 Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG.
 Programa de Pós-Graduação em Engenharia Elétrica, 2014.
 Inclui Referências.

 1. Engenharia Elétrica. 2. Sistemas de manufatura. 3. Modelagem de
 sistemas. 4. Avaliação de Sistemas. 5. Sistemas dinâmicos a eventos
 discretos. I. Alencar, Fernanda Maria Ribeiro de. (Orientadora). II. Título.

UFPE

621.3 CDD (22. ed.)

BCTG/2014-146



Universidade Federal de Pernambuco
Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

JULIO CESAR DE LIMA LOPES

TÍTULO

***“MODELAGEM E AVALIAÇÃO DE SISTEMAS DINÂMICOS A EVENTOS
DISCRETOS UTILIZANDO REDE DE PETRI COLORIDA”***

*A comissão examinadora composta pelos professores: FERNANDA MARIA RIBEIRO DE ALENCAR, DES/UFPE, JAELSON FREIRE BRELAZ DE CASTRO, CIN/UFPE e RICARDO MASSA FERREIRA LIMA, CIN/UFPE, sob a presidência do primeiro, consideram o candidato **JULIO CESAR DE LIMA LOPES** **APROVADO.***

Recife, 26 de fevereiro de 2014.

CECÍLIO JOSÉ LINS PIMENTEL
Coordenador do PPGEE

FERNANDA MARIA RIBEIRO DE ALENCAR
Orientadora e Membro Titular Externo

RICARDO MASSA FERREIRA LIMA
Membro Titular Externo

JAELSON FREIRE BRELAZ DE CASTRO
Membro Titular Externo

Aos meus filhos, pela compreensão,
carinho, amor e paciência
durante a realização deste trabalho.

AGRADECIMENTOS

A Deus, que está sempre presente em todos os momentos de minha vida.

A minha orientadora Prof.^a Dra. Fernanda Alencar pela oportunidade, constante orientação e tudo o que aprendi com ela durante a produção deste trabalho, principalmente, pelas correções e apoio contínuo que permitiram a evolução desta pesquisa.

Aos Professores Dr. Paulo Romero Martins Maciel, Ricardo Massa e Alex Sandro Gomes pelas valiosas sugestões e contribuições no desenvolvimento deste trabalho.

A todos os funcionários do departamento, em especial a secretária do programa de pós-graduação de engenharia elétrica, Andrea Tenório Pinto, por todo apoio.

Ao amigo Josenildo Fernando pelo apoio e incentivo constante à vida acadêmica.

E em especial a Carla Abigail pela cumplicidade e apoio constante em todos os momentos.

Agradeço a todos meus amigos que torceram por mim, para que eu concluísse com sucesso esse desafio.

Um forte abraço!

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

MODELAGEM E AVALIAÇÃO DE SISTEMAS DINÂMICOS A EVENTOS DISCRETOS UTILIZANDO REDE DE PETRI COLORIDA

JULIO CESAR DE LIMA LOPES

Recife, fevereiro de 2014

Orientadora: Profa. Fernanda Maria Ribeiro de Alencar, PH.D.

Área de Concentração: Comunicações

Palavras-chave: Sistemas de Manufatura, Modelagem de Sistemas, Avaliação de Sistemas e Sistemas dinâmicos a Eventos discretos.

Números de Páginas: 86

Nas últimas décadas a modelagem tem recebido uma atenção especial tanto dos pesquisadores, como também pela indústria, que cada vez mais comprova a eficiência da técnica e sua excelente aceitação. Qualquer projeto de sistemas dinâmicos a eventos discretos, como por exemplo, os sistemas automatizados de manufatura, podem e devem ser modelados. Isso deve ocorrer para que se consiga determinar as melhores condições de operação e detecções de erros de projetos, num processo de avaliação, antes que o sistema seja implantado. Erros significam custos que podem inviabilizar todo um projeto. Somente por meio da modelagem e simulação é possível responder assertivamente, antecipadamente e quantitativamente, sobre o *desempenho* a ser obtido por um sistema, levando em consideração que a quantificação da interrupção não prevista e seus efeitos podem ser mensurados, o que possibilita a tomada de decisão e a execução de ações sobre o sistema. Neste trabalho a modelagem sistemas dinâmicos a eventos discretos é implementada através da Rede de Petri Colorida, para que seja possível realizar simulações e tomarem-se decisões em nível de especificação. Uma revisão da literatura foi realizada com vistas a identificar abordagens de modelagem e ferramentas para os sistemas dinâmicos a eventos em sistemas de produção. Apresenta-se um experimento realizado numa MPS (Sistema de Produção e Manufatura), consistindo na análise dos problemas reais enfrentados na indústria numa linha de produção, com o uso da Rede de Petri Colorida.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

MODELLING AND EVALUATION OF DYNAMIC SYSTEMS USING DISCRETE EVENTS OF COLORED PETRI NET

Julio Cesar Lima Lopes

February, 2014

Supervisor: Profa. Fernanda Maria Ribeiro de Alencar, PH.D.

Area of Concentration: Communications

Keywords: Manufacturing Systems, Systems Modeling, Evaluation of Dynamic Systems and discrete events.

Number of pages: 86

In recent decades modeling has received much attention from researchers, but also by industry, which increasingly proves the efficiency of the technique and its excellent acceptance. Any project of dynamic discrete event, such as automated manufacturing systems systems, can and should be modeled. This should occur for one to determine the best operating conditions and error detection project, an evaluation process before the system is deployed. Errors mean that costs can derail an entire project. Only through modeling and simulation can respond assertively advance and quantitative information on the performance to be obtained by a system, taking into account that quantification of unforeseen disruption and its effects can be measured, which enables the decision making and implementation of actions on the system . In this work the dynamic modeling discrete event systems is implemented by Petri Net Colored, so that you can perform simulations and make decisions-level specification. A literature review was performed in order to identify approaches and tools for modeling dynamic systems in the event of production systems. It presents an experiment conducted in MPS (Manufacturing and Production System), consisting in the analysis of the real problems facing the industry in a production line, with the use of colored Petri net

Sumário

1	INTRODUÇÃO.....	12
1.1	CONTEXTUALIZAÇÃO.....	12
1.2	MOTIVAÇÃO.....	13
1.3	PROBLEMA.....	14
1.4	OBJETIVO.....	15
1.5	RELEVÂNCIA.....	15
1.6	MÉTODO.....	16
1.7	ESTRUTURA DA DISSERTAÇÃO.....	17
2	SISTEMA A EVENTOS DISCRETOS.....	18
2.1	FUNDAMENTOS DA TEORIA DE SISTEMAS A EVENTOS DISCRETOS.....	18
2.2	EVENTO.....	19
2.3	SISTEMAS CONTROLADOS PELO TEMPO E SISTEMAS BASEADOS EM EVENTOS.....	20
2.4	PROPRIEDADES CARACTERÍSTICAS DE SISTEMAS A EVENTO DISCRETOS.....	21
2.5	EXEMPLOS DE SISTEMAS A EVENTO DISCRETOS.....	22
2.6	SISTEMAS DE MANUFATURA.....	24
2.7	SISTEMAS DE TRÁFEGO.....	26
2.8	FORMAS POSSÍVEIS PARA MODELAGEM.....	27
2.9	MODELOS EXISTENTES.....	29
2.10	CONSIDERAÇÕES FINAIS.....	29
3	CONTROLADORES LÓGICOS PROGRAMÁVEIS (CLP).....	30
3.1	OPERAÇÃO BÁSICA DO CLP.....	32
3.2	ARQUITETURA BÁSICA DO CLP.....	34
3.3	CONSIDERAÇÕES FINAIS.....	35
4	REDES DE PETRI.....	36
4.1	FORMALISMO DE MODELAGEM POR REDE DE PETRI.....	36
4.2	DEFINIÇÃO DA REDE DE PETRI.....	38
4.3	ESTRUTURA DA REDE DE PETRI.....	39
4.4	CLASSIFICAÇÃO DA REDE DE PETRI.....	44
4.5	PROPRIEDADES DA REDE DE PETRI.....	45
4.6	O MODELO MATEMÁTICO.....	47
4.7	REDE DE PETRI COLORIDA.....	50
4.8	VIABILIDADE ECONÔMICA DE MODELAGEM POR REDE DE PETRI.....	52
4.9	CONSIDERAÇÕES FINAIS.....	55
5	MODELAGEM DO SED UTILIZANDO REDE DE PETRI - MODELO PROPOSTO.....	56
5.1	APRESENTAÇÃO DO SISTEMA DE DISTRIBUIÇÃO DE PEÇAS.....	56
5.2	ETAPAS DESENVOLVIDAS NA EXECUÇÃO DA MODELAGEM.....	58
5.3	CONSIDERAÇÕES FINAIS.....	73
6	CONSIDERAÇÕES FINAIS E TRABALHO FUTUROS.....	74

6.1	TRABALHOS FUTUROS	76
7	ANEXOS.....	78
	REFERÊNCIAS	83

LISTA DE ABREVIACÕES

MPS – Sistema de Produção e Manufatura

SFMs – Sistemas Flexíveis de Manufatura

SEDs – Sistemas a Eventos Discretos

RP – Rede de Petri

PU – Processo Unificado

CLP – Controlador Lógico Programável

CPU – Unidade Central de Processamento

ABNT – Associação Brasileira de Normas Técnicas

NEMA – National Electrical Manufacturers Association

IEC – International Electrotechnical Commission

RAM – Random Access Memory

EPROM – Erasable Programmable Read Only Memory

EEPROM – Electrically Erasable Programmable Read Only Memory

NVRAM ou FLASH-EPROM – Non-Volatile Random Access Memory

AV – Ambiente Virtual

RPCs – Rede de Petri Coloridas

LISTA DE FIGURAS

Figura 1.1 - Modelagem do SED.....	17
Figura 2.1 - Sistema de fila.....	23
Figura 2.2 - Sistema Industrial de filas.....	26
Figura 2.3 - Uma simples interseção T controlada por semáforos	27
Figura 3.1 - Estrutura Básica de um CLP	31
Figura 3.2 - Lógica de contato elétrico.....	33
Figura 3.3 - Lógica por meio do CLP.....	33
Figura 3.4 - Estrutura básica do CLP	34
Figura 3.5 - Exemplos de algumas CPU's disponíveis no mercado	34
Figura. 4.1 – Componente Lugar.....	39
Figura 4.2 - a) Lugar com um token e b) com vários tokens	40
Figura 4.3 - Componente Transição	40
Figura 4.4 - Componente Arco	41
Figura 4.5 - Ligação Lugar x Transição	41
Figura 4.6 - Ligação Transição x Lugar	41
Figura 4.7 - Marca.....	42
Figura 4.8 - Notação Formal	43
Figura 4.9 - Notação Gráfica	44
Figura 4.10 - Representação através de Rede de Petri do ano letivo	48
Figura 4.11 - Expressão.....	52
Figura 4.12 - Evolução do custo de um sistema em suas diferentes fases.....	53
Figura 4.13 - Comparativo da evolução dos custos com e sem aplicação da simulação	54
Figura 4.14 - Esquema do conceito de transformação da realidade x resultado em realidade.....	55
Figura 5.1 - Estação de Distribuição de Peças.....	57
Figura 5.2 - Estação de Separação de Peças	57
Figura 5.3 - Estratégia de modelagem de SED com Rede Petri Colorida	59
Figura 5.4 - Caracterização do SED	60
Figura 5.5 - Girador do Magazine	62
Figura 5.6 - Fluxograma de funcionamento do sistema de distribuição	63
Figura 5.7 - Modelagem	64
Figura 5.8 - O Modelo.....	68
Figura 5.9 - Conversão LADDER x Rede de Petri.....	70
Figura 5.10 - Conversão do Modelo em Rede de Petri em LADDER.....	71
Figura 5.11 - Resultado do experimento	71
Figura 7.1 - Exemplo de Programa a partir do modelo de Rede de Petri	78
Figura 7.2 - Exemplo de Programa a partir do modelo de Rede de Petri	80

LISTA DE TABELAS

Tabela 5.1 - Lista de dispositivos com funções.....	61
Tabela 5.2 – Planta	65
Tabela 5.3 - Avaliação do experimento.....	69
Tabela 5.4 - Comparativo entre os Trabalhos relacionados.....	72

CAPÍTULO 1

1 INTRODUÇÃO

Este capítulo tem por finalidade discutir os aspectos de maior relevância para o tema apresentado, destacando a automação dos sistemas de manufatura e quais os principais problemas oriundos desses sistemas. São realizadas as considerações iniciais e colocadas à motivação, o problema, o objetivo do trabalho, a relevância, método e a estrutura dessa dissertação.

1.1 CONTEXTUALIZAÇÃO

A evolução natural do sistema produtivo e a globalização deram origem a um período de mercados escassos, com uma maior exigência de qualidade e maior competitividade. As empresas tiveram que se modernizar reduzindo o tempo de resposta de lançamento dos seus produtos. A diversidade, as flutuações de demanda, o curto ciclo de vida dos produtos, além do crescimento das expectativas dos clientes em termos de qualidade e tempo de entrega, constituem atualmente desafios com que as empresas devem lidar para permanecerem competitivas e sobreviverem no mercado.

Nesse contexto, agilidade e flexibilidade são conceitos reconhecidos como requisitos para um sistema de manufatura satisfazer as necessidades de um mercado global competitivo. Isso significa que o sistema deve responder rapidamente a mudanças de produção tanto no que se refere a volume como a variedade, de forma efetiva, confiável e a um baixo custo. De maneira geral, o tempo de implantação de um sistema automatizado deverá ser menor possível, uma vez que novos produtos requerem novos requisitos em relação à automação [MOORE et al., 2003].

Porém, [MORAES & CASTRUCCI, 2010] afirmam que as empresas estão ainda dependendo recursos na implantação de um sistema automatizado, pois a maior parcela do tempo utilizada na implantação é dedicada à detecção e correção de falhas e a otimizações, em uma etapa posterior à instalação do sistema. Neste contexto, propõe-se uma abordagem inovadora para lidar com o ciclo de desenvolvimento de sistemas automatizados de manufatura.

Face ao exposto, observa-se uma migração dos tradicionais sistemas de produção

baseados em produção de lote para uma arquitetura em que se tem maior flexibilidade, caracterizando os sistemas de manufatura [GROOVER, 2011], também chamados de Sistemas Flexíveis de Manufatura (SFMs). Esses sistemas produzem uma variedade de produtos modificando a sua configuração de acordo com o planejamento da produção, fazendo expandir um novo campo de conhecimento, o da teoria de Sistemas a Eventos Discretos (SED). Tais sistemas tem em comum a maneira pela qual percebem as ocorrências no ambiente à sua volta e sua importância faz com que seja altamente desejável encontrar soluções para problemas relacionados ao seu controle.

O surgimento dos SED justifica-se, entre outras coisas, em face da necessidade de um tratamento formal requerido por diversos sistemas construídos pelo homem, como redes de comunicação, sistema de tráfego automatizado, sistemas computacionais e os sistemas de manufatura, esse especificamente foco da presente pesquisa.

Torna-se então imperativo dispor de metodologias sistemáticas de projeto para atingir o desempenho desejável e evitar falhas. Fazem-se necessários métodos de modelagem que consideram as ocorrências de falhas e que possibilite resolver o problema da incerteza sobre a real capacidade de produção do sistema. Dessa forma, será possível a análise do desempenho dos SED sob diversos pontos de vista ou situações que ocorrem no dia a dia no gerenciamento dos Sistemas Automatizados de Manufatura.

1.2 MOTIVAÇÃO

Qualquer projeto de sistemas dinâmicos a eventos discretos (como os sistemas automatizados de manufatura) pode ser modelados. Isso ocorre para se determinar as melhores condições de operação e se detectar erros de projetos antes que o sistema seja implantado. Erros significam custos que podem inviabilizar todo um projeto, conforme visto anteriormente.

Estudos revelam que a diagnose de falhas em Sistemas a Eventos Discretos (SEDs) tem despertado grande interesse nos últimos anos. Dois paradigmas norteiam a diagnose de falhas em SEDs:

1. as falhas a serem diagnosticadas são eventos não observáveis, isto é, eventos cujas ocorrências não podem ser registradas por sensores;
2. a ocorrência de falhas altera o comportamento do sistema, porém não necessariamente leva o sistema a uma parada; por exemplo, em sistemas de

manufatura, a ocorrência de uma falha não diagnosticada pode levar a uma degradação dos indicadores de eficácia global dos equipamentos (disponibilidade, eficiência e qualidade).

O projeto desses sistemas requer a construção de um modelo a eventos discretos do sistema que capture tanto o comportamento normal quanto o comportamento do sistema levando-se em consideração a ocorrência da falha.

Um sistema automatizado de manufatura é constituído de vários controladores lógicos, cuja programação é realizada de forma intuitiva, sem a observância de critérios, que levam a erros de programação que só serão identificados na fase de testes realizados no sistema real (já implantado).

Assim, surge a necessidade e a inquietação de se investigar um método de modelagem e avaliação de sistemas dinâmicos a eventos discretos. Como possível solução, propõe-se modelar um sistema de manufatura de produção a partir do modelo de Rede de Petri Colorida. Com isso, espera-se analisar as ocorrências de falhas na programação dos controladores lógicos. Ao mesmo tempo em que proporciona um sistema previamente validado, livre de erros e falhas que permita uma maior segurança no desenvolvimento dos programas previstos para os controladores programáveis.

1.3 PROBLEMA

Diante da necessidade de se evitar falhas nos sistemas é importante discutir: Como prover um sistema previamente validado, livre de erros e falhas que permita uma maior segurança e qualidade no desenvolvimento dos programas dos controladores lógicos programáveis (CLP's) utilizando rede de Petri Colorida?

Assim, o problema tratado neste trabalho é a fragilidade das análises feitas para validação técnica e econômica dos projetos de sistemas de manufatura. As ferramentas de simulação existentes, como por exemplo, os controladores lógicos programáveis, auxiliam nos testes de validação, porém não conseguem exprimir o correto funcionamento dos equipamentos conforme abordagens em estudos anteriormente realizados.

1.4 OBJETIVO

O objetivo geral desta pesquisa é propor uma abordagem de avaliação em Sistemas Dinâmicos a Eventos Discretos usando Rede de Petri Colorida para a programação de CLP's, a ser implementado num modelo real.

Nesse contexto os objetivos específicos são:

1. Obter um modelo de sistema de Evento Discreto utilizando rede de Petri Colorida;
2. Simular a rede de Petri Colorida;
3. Implementar e validar a rede de Petri Colorida no sistema real.

1.5 RELEVÂNCIA

Com a modelagem do sistema de manufatura de produção a partir do modelo de Rede de Petri Colorida, serão evitadas ocorrências de falhas na programação dos controladores lógicos. O modelo de Rede Petri Colorida proporcionará um sistema previamente validado, sendo detectados os possíveis erros. Dessa forma, permite-se uma maior segurança no desenvolvimento dos programas previstos para os controladores programáveis, por isso é que será adotado esse modelo na presente pesquisa.

Fatores importantes fortalecem as vantagens da utilização da Rede de Petri (RP) representadas pelas seguintes considerações:

1. A RP tem sido usada com sucesso para modelar, controlar e analisar sistemas a eventos discretos, que são caracterizados pela concorrência dos processos, pelo paralelismo, por serem assíncronos, por possuírem *deadlocks*, conflitos e processos de eventos direcionados;
2. A RP permite a implementação de análise em tempo real;
3. A RP constitui-se como uma teoria bem fundamentada para a verificação de propriedades qualitativas.

Sob esta ótica, é relevante a modelagem e avaliação desses sistemas sendo possível responder assertivamente, antecipadamente e quantitativamente, sobre o desempenho a ser obtida pelo sistema, levando-se em consideração que a quantificação da interrupção não prevista e seus efeitos podem ser mensurados, o que possibilita a tomada de decisão e a execução de ações sobre o sistema.

1.6 MÉTODO

Para [GIL, 2009], a pesquisa tem um caráter pragmático, é um “processo formal e sistemático de desenvolvimento do método científico”. “O objetivo fundamental da pesquisa é descobrir respostas para problemas mediante o emprego de procedimentos científicos”.

Esta pesquisa investiga os resultados da aplicação da simulação e modelagem do projeto de uma nova linha de manufatura. Na pesquisa serão usadas ferramentas de modelagem e simulação, o que também a caracteriza como simulação. Sendo assim, o método de pesquisa utilizado foi à pesquisa bibliográfica, a modelagem e a simulação.

A simulação é a técnica de estudar o comportamento de reações de determinados sistemas através de modelos. A simulação de processos permite fazer uma análise de um sistema sem a necessidade de interferir no mesmo. As possíveis mudanças a serem realizadas e suas consequências irão interferir apenas no modelo computacional e não no sistema real. Na simulação, constrói-se um modelo lógico matemático que representa de forma mais precisa o sistema que esteja sendo estudado. Esse modelo incorpora valores para recursos disponíveis, distâncias, tempos.

A simulação é um processo de experimentação que utiliza um modelo suficientemente detalhado de um sistema para determinar como este responderá as mudanças em sua estrutura, ambiente ou condições de contorno [CASSANDRA & LAFORTUNE, 1999]. O termo simulação refere-se à representação de um processo dinâmico ou sistema de forma permitir sua análise, por considerar o problema atual muito grande e/ou complexo [HARREL & BATEMAN, 2002]. Em muitos casos, o propósito é analisar e entender o comportamento do sistema em função de cenários alternativos [LAW & KELTON, 2000]. Assim tem-se no fluxograma abaixo relacionadas às etapas realizadas nessa pesquisa, ou seja, inicialmente, foi realizada a revisão da literatura onde foram pesquisados artigos, periódicos e dissertações com as palavras chaves relacionados com o tema do trabalho. Em seguida foi definido um modelo, isto é, foi definido qual o modelo do SED que seria utilizado no trabalho. Uma vez definido o modelo, partiu-se para implementação do modelo e posteriormente para a sua simulação e avaliação, onde será visto com mais detalhe no capítulo dos resultados.

Figura 1.1 - Modelagem do SED

Fonte: [próprio autor]

1.7 ESTRUTURA DA DISSERTAÇÃO

Além deste capítulo 1 destinado à introdução descreve algumas considerações iniciais, contextualiza o presente estudo e apresenta a motivação, o problema, o objetivo do trabalho, a relevância, metodologia e a estrutura da dissertação.

O capítulo 2 apresenta fundamentos da teoria de sistemas a eventos discretos: conceitua Evento, demonstra Sistemas controlados pelo tempo e sistemas baseados em eventos, relatam propriedades características de sistemas a evento discretos, exemplifica sistemas a evento discretos, aborda os sistemas de manufatura, os sistemas de tráfego, formas possíveis para modelagem e os modelos existentes.

No capítulo 3 é apresentado o CLP (Controlador Lógico Programável) as operações básicas com seu uso e sua arquitetura básica.

No capítulo 4 é apresentado o formalismo de modelagem por Redes de Petri, discute-se definição, estrutura, classificação e propriedades. No final do capítulo aborda-se modelo matemática da Rede de Petri e ainda a definição e estrutura da Rede de Petri colorida e a demonstração da viabilidade econômica da Rede de Petri.

O capítulo 5 é apresentado à proposta de modelagem de Sistemas Dinâmicos a Eventos discretos utilizando Rede de Petri Colorida, e ainda, a análise dos resultados obtidos.

Finalmente, no capítulo 6 são apresentados as conclusões e os trabalhos futuros.

2 SISTEMA A EVENTOS DISCRETOS

Este capítulo tem como finalidade apresentar uma revisão bibliográfica a respeito dos fundamentos da teoria de sistemas a eventos discretos e sua relevância. São levantadas as definições de evento, sistemas controlados pelo tempo e sistemas baseados em eventos. Ainda neste capítulo é feita a apresentação das propriedades características de sistemas a evento discretos, exemplos de sistemas a evento discretos, sistemas de filas, sistemas de manufatura e sistemas de tráfego. Finalmente, apresentam-se formas possíveis para modelagem, modelos existentes e controladores lógicos programáveis (CLP).

2.1 FUNDAMENTOS DA TEORIA DE SISTEMAS A EVENTOS DISCRETOS

Nesta seção serão apresentados os principais conceitos para o estudo de Sistemas a eventos discretos.

Um sistema a eventos discretos (SED) é um sistema a estado discreto, dirigido por eventos, ou seja, sua evolução de estado depende inteiramente da ocorrência de eventos discretos, assíncronos no tempo. Neste sentido, [ATTIÉ, 1998] escreve que “quando o espaço de estados de um sistema é naturalmente descrito por um conjunto discreto, e as transições de estado são observadas somente em pontos discretos do tempo, associam-se estas transições a eventos”. O conceito de evento é um desses conceitos primitivos, cuja compreensão deve ser deixada à intuição, mais do que a uma exata definição. Não se pode, porém, deixar de enfatizar que um evento deve ser pensado como de ocorrência instantânea e como causador de uma transição no valor (discreto) do estado do sistema.

Segundo [CURY, 2001], diz-se que um sistema a eventos discretos (SED) é um sistema dinâmico que evolui de acordo com a ocorrência de eventos físicos, em intervalos de tempo em geral irregulares e desconhecidos. Para [MORAES, 2010] sistemas a eventos discretos são sistemas que respondem aos eventos discretos externos e internos com sinais também discretos, de acordo com rígidas regras de causa e efeito ou, então, com regras estatísticas. Nos SED, há que se considerarem diferentes métodos de análise conforme circunstâncias específicas. Se os eventos externos são raros e, portanto, de inexpressiva descrição estatística (por exemplo, o apertado de um botão de emergência) e se, de outro

lado, o funcionamento interno desses sistemas obedece a regras rígidas e tempos de reação fixam, o método de análise deve ser determinístico. Se os eventos externos ocorrem frequentemente aleatoriamente, a ferramenta ideal é a análise estatística. Portanto, na realidade, tanto a análise determinística quanto a estatística são desejáveis, desde que utilizadas no momento certo.

Atualmente são inúmeros os sistemas a eventos discretos, pela sua fundamental importância na ordenação da vida civilizada contemporânea; ocorrem nas indústrias, nos serviços prestados ao público, nos processos burocráticos, nos softwares de tempo real e dos bancos de dados, nas manufaturas. Em geral, em tais sistemas intervêm eventos externos importantes e não programáveis, enquanto internamente existe uma lógica rigorosa de causas e efeitos [MORAES, 2010].

2.2 EVENTO

“Evento” é um conceito primitivo e necessita de uma boa base intuitiva para que seja possível se compreender o seu significado. Deve ser enfatizado que um evento deve ser pensado como sendo alguma coisa que acontece instantaneamente e que causa transições de um valor de estado para outro. [ATTIÉ, 1998].

Um evento pode ser identificado como uma ação específica, por exemplo: alguém aperta um botão; um computador deixa de funcionar; a chave de ignição de um automóvel é ligada; etc. Por outro lado, um evento também pode ser o resultado de várias condições que acontecem de forma inesperada.

A ocorrência de um evento causa, em geral, uma mudança interna no sistema, a qual pode ou não se manifestar a um observador externo.

Além disso, uma mudança pode ser causada pela ocorrência de um evento interno ao próprio sistema, tal como o término de uma atividade ou o fim de uma temporização.

Em qualquer caso, essas mudanças se caracterizam por serem abruptas e instantâneas. Ao perceber um evento, o sistema reage imediatamente, acomodando-se em tempo nulo numa nova situação, onde permanece até que ocorra um novo evento.

Neste trabalho será usado o símbolo e para denotar um evento. Ao considerar um sistema afetado por tipos diferentes de eventos, definem-se um conjunto E cujos elementos são todos esses eventos. Claramente, E é um conjunto discreto.

São exemplos de eventos, o início e o término de uma tarefa (mas não sua execução), a chegada de cliente a uma fila ou a recepção de uma mensagem em um sistema de comunicação.

O conceito de evento pode ser mais bem entendido com a ajuda do seguinte exemplo: um sistema de armazenamento de cargas. Neste caso pode-se perfeitamente verificar que há, no mínimo, dois eventos: um evento é a “chegada de produto” e o outro é a “chegada de caminhão”. Neste caso, pode-se definir um conjunto de eventos $\mathbf{E} = \{\mathbf{P}, \mathbf{T}\}$ onde \mathbf{P} denota o evento “chegada de produto”, e \mathbf{T} denota o evento “chegada de caminhão”, que corresponde à “saída do produto”.

2.3 SISTEMAS CONTROLADOS PELO TEMPO E SISTEMAS BASEADOS EM EVENTOS

Em sistemas de estados contínuos o estado muda, geralmente, com mudanças de tempo. Isto é particularmente evidente em modelos a tempo-discreto: um sinal de “*clock*” determina a sequência de amostras a serem obtidas, pois é esperado que a cada marcação desse sinal, ocorra uma mudança no estado do sistema. Neste caso, a variável de tempo (\mathbf{t} , em tempo contínuo, ou \mathbf{k} , em tempo-discreto) é uma variável independente que aparece como sendo o argumento de toda a contribuição de entrada dos estados, e em funções de saída. Por essa razão, esses sistemas são denominados dirigidos pelo tempo.

Em sistemas de estados discretos, as mudanças de estado só ocorrem em certos pontos, por transições instantâneas e, a cada uma dessas transições, pode-se associar um evento. Suponha que exista um relógio pelo qual é tomado o tempo, e considere as duas possibilidades:

1. A toda marcação do sinal de *clock*, um evento \mathbf{e} será selecionado de um conjunto fixo \mathbf{E} . Se nenhum evento acontecer, pode-se pensar em um "evento nulo" como pertencendo a \mathbf{E} ($\mathbf{e} \in \mathbf{E}$) cuja propriedade é não causar nenhuma mudança de estado.
2. Em vários momentos de tempo (não necessariamente conhecidos com antecedência, e não coincidindo com as marcações de tempo), algum evento determinado \mathbf{e} , irá ocorrer.

Há uma diferença fundamental entre 1 e 2 acima. Em 1, as transições de estado são sincronizadas pelo relógio, isto é, a toda marcação de tempo, um evento (ou nenhum evento) é selecionado. O tempo é responsável por toda e qualquer possível transição de estado. Em 2, todo evento e , de \mathbf{E} ($e \in \mathbf{E}$), define um processo distinto pelo qual os momentos de tempo, quando e acontece, são determinados. As transições de estado são o resultado das combinações desses processos de eventos assíncronos e simultâneos.

Além disso, estes processos não precisam ser independentes um do outro. A distinção entre 1 e 2 dá origem às definições de sistemas dirigidos pelo tempo (1) e sistemas dirigidos por eventos (2). É importante ressaltar que a ideia de transições de estado baseadas em eventos corresponde a uma noção familiar que é de uma “interrupção” em sistemas de computador. Enquanto muitas das funções em um computador são sincronizadas por um relógio, e são controladas pelo tempo, outras são resultados de chamadas assíncronas que podem acontecer a qualquer hora. Exemplo disso é o pedido de um usuário externo ou uma mensagem de intervalo que pode acontecer como resultado de eventos específicos, mas completamente independentes do relógio do computador.

Em resumo, têm-se duas grandes classes de sistemas: os sistemas acionados pelo tempo “*time-driven*”; e os sistemas acionados por eventos “*event-driven*”. Os primeiros são descritos por equações diferenciais na variável tempo e são chamados de contínuos no tempo. Já os últimos são descritos por Álgebra de Boole, álgebra dióide, autômatos finitos, redes de Petri, programas computacionais e são chamados de Eventos Discretos [MORAES, 2010]. Essa última categoria é a que está sendo considerada neste trabalho.

2.4 PROPRIEDADES CARACTERÍSTICAS DE SISTEMAS A EVENTO DISCRETOS

A maioria dos sistemas de controle em engenharia é baseada em modelos de equações diferenciais ou em equações a diferenças lineares. Para usar esses modelos matemáticos, os sistemas devem ser de estado contínuo e possuírem o mecanismo de transição de estado dirigido pelo tempo. A primeira propriedade permite definir o estado por meio de variáveis contínuas que podem assumir qualquer valor real (ou complexo). Quantidades físicas comuns como posição, velocidade, aceleração, temperatura, pressão, fluxo etc., estão nesta categoria desde que se possam definir naturalmente as derivadas

para estas variáveis contínuas. A segunda propriedade vem o fato de que o estado geralmente evolui em função do tempo.

Os sistemas considerados neste trabalho são os Sistemas Dinâmicos a Eventos Discretos (SDED) ou, mais amplamente, Sistemas a Eventos Discretos (SED). As suas principais características são: o espaço de estado é um conjunto discreto; e, o mecanismo de transição de estados é baseado em eventos. Essas propriedades levam a seguinte definição de SED: um sistema de estado discreto baseado em eventos, isto é, aquele onde a evolução dos estados depende somente da ocorrência de eventos discretos assíncronos.

Muitos sistemas, particularmente tecnológicos, são na realidade sistemas de estados discretos. Até mesmo se este não for o caso, para muitas aplicações de interesse, uma visão de estado discreto de um sistema complexo pode ser necessária [VIEIRA, 2007]. Alguns exemplos simples de sistemas de estados discretos são:

1. o estado de uma máquina pode ser selecionado de um conjunto como {LIGADA, DESLIGADA} ou {OCUPADO, OCIOSO, LIVRE};
2. um computador que executa um programa pode ser visto como estando em um de três estados: {ESPERANDO POR INSTRUÇÕES, EXECUTANDO, PARADO};
3. inventário de tipo qualquer que consiste de valores discretos (por exemplo, produtos, unidades monetárias, pessoas) tem um espaço de estado natural nas grandezas não negativas $\{0,1,2,\dots\}$,
4. jogos que em sua maioria podem ser modelado como tendo um espaço de estado discreto (em xadrez, por exemplo, toda possível configuração do tabuleiro define um estado); o espaço resultante é enorme, mas é discreto.

Estes eventos são, por sua natureza, instantâneos, o que lhes confere um caráter discreto no tempo, ou seja, isso decorre do fato de que o estado só pode mudar no tempo em pontos discretos, que correspondem fisicamente a ocorrências assíncronas de eventos discretos.

2.5 EXEMPLOS DE SISTEMAS A EVENTO DISCRETOS

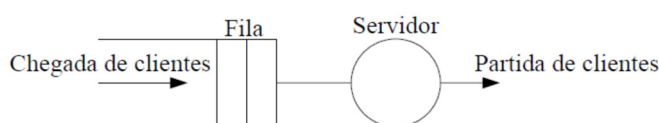
Nesta seção serão apresentados três exemplos de SED utilizados no mundo real e experiências comuns em engenharia, com ênfase ao sistema de manufatura, objeto de interesse da presente pesquisa.

Sistemas de filas

O termo fila decorre de um fato intrínseco que em muitos dos sistemas mais comuns, para se usar certos recursos, deve-se esperar. Por exemplo, para usar os recursos de um caixa de banco, as pessoas formam uma fila e esperam; para usar o recurso de um caminhão, produtos acabados esperam em um armazém. De forma similar, para se usar os recursos da CPU, várias tarefas esperam em algum lugar no computador até que seja dado acesso às mesmas por mecanismos potencialmente complexos.

Há três elementos básicos em um sistema de filas, representado na Figura 2.1 que são: as entidades que fazem a espera para utilização dos recursos, estas entidades são usualmente denominadas clientes; os recursos para os quais a espera é realizada, desde que os recursos provejam alguma forma de serviços aos clientes, são genericamente chamados de servidores; e o espaço onde a espera é realizada, a este elemento dar-se o nome fila.

Figura 2.1 - Sistema de fila



Fonte: [CURY, 2001]

A cada chegada, o cliente, ou se dirige ao SERVIDOR e é servido, ou tem que esperar primeiro na FILA até que o servidor esteja disponível. Após ser atendido, cada cliente parte. Exemplos de clientes são: pessoas (por exemplo, esperando em um banco ou em um ponto de ônibus), mensagens transmitidas de algum meio de comunicação, tarefas, trabalhos ou transações executadas em um sistema de computador, produção em um processo de fabricação e carros que usam uma Rede de estradas.

Exemplos de servidores são: pessoas (por exemplo, caixas de banco ou caixas de Saída de supermercado), canais de comunicação responsáveis pela transmissão de mensagens, processadores de computador ou dispositivos periféricos, vários tipos de máquinas usadas na fabricação e semáforos que regulam o fluxo de carros.

Exemplos de filas são encontrados em vários locais, como por exemplo, banco, pontos de ônibus ou supermercados. Porém, filas também estão presentes em Redes de

comunicação ou sistemas de computador onde também são alocadas formas menos tangíveis de clientes, como telefonemas ou tarefas a serem executadas em áreas de espera.

Graficamente, um simples sistema de fila será representado como mostrado na Figura 1. O círculo representa um servidor, e uma caixa aberta representa uma fila que precede a este servidor. Aberturas de fila indicam os clientes em modo de espera. Os clientes são vistos como chegando à fila e partindo do servidor. Supõe-se ainda que o processo de servir os clientes, normalmente leva uma quantidade estritamente positiva de tempo (caso contrário não haveria espera).

Assim, um servidor pode ser visto como um "bloco de atraso" que retém um cliente por algum tempo até a realização do serviço.

Visto como um SED, o sistema de fila da Figura 1 tem um conjunto de eventos $E = \{\mathbf{a}, \mathbf{d}\}$, onde $\{\mathbf{a}\}$ denota um evento de chegada e $\{\mathbf{d}\}$ denota um evento de saída.

Uma variável de estado é o número de clientes na fila ou o comprimento da fila. Assim, o espaço de estado é o conjunto de valores não negativos $X = \{0, 1, 2, \dots\}$.

2.6 SISTEMAS DE MANUFATURA

Segundo [GROOVER, 2011], os sistemas de manufatura são como uma coleção de equipamentos e recursos humanos integrados, cuja função é realizar uma ou mais operações de processamento e/ou montagem na matéria-prima, na peça ou em conjunto inicial de peças. Segundo [NAKASHIMA; GUPTA, 2003] os Sistemas de Manufatura são caracterizados por um baixo custo de produção na quantidade certa, com o mínimo estoque e de acordo com a demanda.

Os sistemas que satisfazem todas essas características são chamados de Sistemas Flexíveis de Manufatura (SFMs). Esses sistemas normalmente possuem várias estações automatizadas e são capazes de roteamentos variáveis entre as estações. [GROOVER, 2011].

Segundo [SHIM; SIEGEL, 1999], os SFMs são definidos como sistemas de produção controlados por computador com tecnologia suficientemente adequada para produzir uma variedade moderada de produtos dentro de um volume moderado flexível.

Os SFMs produzem uma variedade de produtos modificando as suas configurações de acordo com o planejamento da produção. Essa flexibilidade permite uma alocação

rápida dos recursos, mas incrementa a complexidade de controle do sistema [UZAM; ZHOU, 2007].

Desta forma, equipamentos autônomos, cooperando entre si através de comunicação via Redes industriais, aliados a um fluxo contínuo de diferentes produtos pelas linhas de produção, tornam as plantas de produção cada vez mais difíceis de serem projetadas, especificadas e administradas [ARAÚJO; BECKER; PEREIRA, 2001].

Os sistemas de manufatura são também sistemas muitas vezes convenientemente descritos por modelos de filas. Em geral, tem-se:

1. Clientes: peças ou itens; “*pallets*”;
2. Servidores: máquinas; dispositivos de manuseio e transporte (robôs, esteiras...);
3. Filas: armazéns; “*buffers*”.

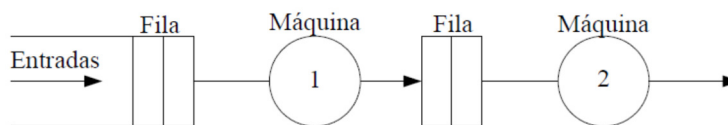
Ou seja, em um processo industrial, os clientes são as peças ou partes de peças da produção. Essas peças estão dispostas para o acesso aos vários servidores da indústria que são as máquinas que executam operações específicas e dispositivos de manipulação de material, como robôs e correias transportadoras. [WU; ZHOU; LI 2008].

Quando as peças não estão sendo trabalhadas, elas são armazenadas em uma fila até que o servidor libere o acesso para a próxima operação que está disponível. Por causa de reais limitações físicas, filas em um sistema industrial têm normalmente capacidades finitas.

Uma vez mais, modelos de filas provêm uma conveniente descrição para sistemas industriais. Um exemplo simples é mostrado na Figura 2.2, onde as peças passam por duas máquinas, sendo a capacidade da primeira fila infinita, enquanto a capacidade da fila da segunda máquina limitada a dois.

Como resultado, é possível que uma parte de serviço da máquina 1 seja completado porém a máquina 2 esteja ocupada e além disso a fila esteja completa. Neste caso, a peça tem que permanecer na máquina 1 embora não requeira mais nenhum serviço; além disso, são forçadas outras peças a esperar o acesso na máquina 1 permanecendo em fila.

O conjunto de eventos fixado para este exemplo é $\mathbf{E} = \{\mathbf{a}, \mathbf{c}_1, \mathbf{d}_2\}$, onde \mathbf{a} é uma chegada para a primeira máquina; \mathbf{c}_1 é uma conclusão de serviço da primeira máquina; e, \mathbf{d}_2 é uma partida para a fila da segunda máquina, demonstrado na Figura 2.2.

Figura 2.2 - Sistema Industrial de filas

Fonte: [CURY, 2001]

Observe que o evento não implica em movimento de uma peça da máquina 1 para a fila da máquina 2, desde que esta possibilidade esteja bloqueada. O estado do sistema pode ser definido como um vetor $\mathbf{x} = [x_1, x_2]^T$ correspondendo aos comprimentos de fila das duas máquinas. Neste caso, x_2 é restrito aos valores $\{0, 1, 2, 3\}$.

Porém, note que quando $x_2 = 3$, a máquina 1 é bloqueada, pois acabou de executar o serviço na peça e a fila da segunda máquina está completa. Para modelar o fenômeno de bloqueio necessitamos introduzir uma variável adicional B que x_2 pode gerar.

O espaço de estado se torna o conjunto discreto $\mathbf{X} = \{(x_1, x_2): x_1 \geq 0, x_2 \in \{0, 1, 2, 3, B\}\}$.

Para ilustrarmos a flexibilidade do processo modelado (dependendo do nível de detalhe que se deseja capturar) pode-se gerar um espaço de estado alternativo que pode ser: $\mathbf{X} = \{x_1, x_2\} : x_1 \in \{I, W, B\}$ e $x_2 \in \{I, W\}$ onde x_1 é o estado da primeira máquina, que pode assumir os seguintes valores: inativo (**I**), trabalhando (**W**) ou bloqueado (**B**), e x_2 é o estado da segunda máquina, que pode assumir os seguintes valores: inativo (**I**) ou trabalhando (**W**). Neste modelo, não são focalizados os comprimentos das filas, mas sim os estados lógicos de cada máquina.

2.7 SISTEMAS DE TRÁFEGO

Considere, agora, como exemplo uma simples interseção em **T** (Figura 2.3). Há quatro tipos movimentos de veículos:

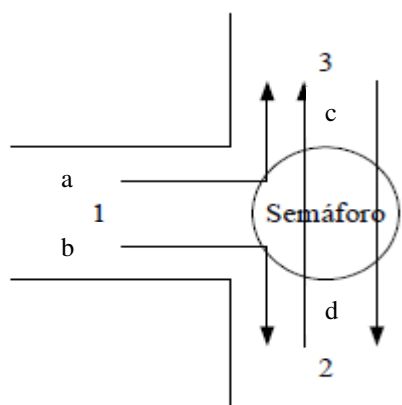
1. veículos vindo de ponto 1 e virando para o ponto 2;
2. veículos vindo de 1 e virando para o ponto 3;
3. veículos que vão diretamente do ponto 2 ao 3,
4. veículos que vão do ponto 3 ao 2. O semáforo funciona da seguinte forma: fica vermelho para os veículos vindo da posição 1 e verde para os veículos vindo das posições 2 e 3, permitindo assim os movimentos **c** e **d**, ou ao contrário,

vermelho para os veículos vindo das posições 2 e 3 e verde para os veículos vindo da posição 1, permitindo os movimentos **a** e **b**.

Neste caso, o conjunto de eventos é determinado por: $E = \{a_{12}, a_{13}, a_{23}, a_{32}, d_{12}, d_{13}, d_{23}, d_{32}, g, r\}$, onde $a_{12}, a_{13}, a_{23}, a_{32}$ são as chegadas de veículo em cada uma das quatro possibilidades e d_{12}, d_{13}, d_{23} e d_{32} são as partidas de veículo quando o semáforo permite o tráfego, g e r indicam o estado do semáforo.

Um possível espaço de estado é definido pelos comprimentos de fila formados pelos quatro tipos de veículo e o estado do próprio semáforo, isto é : $X = \{(x_{12}, x_{13}, x_{23}, x_{32}, y)\}$: $x_{12}, x_{13}, x_{23}, x_{32} \geq 0$, $y \in \{g_1, g_2, g_3, r_1, r_2, r_3\}$, onde $x_{12}, x_{13}, x_{23}, x_{32}$ são os quatro comprimentos de fila, e y é o estado da luz (g_i e r_i denotam, respectivamente, verde e vermelho para os veículos que vem dos pontos indicados).

Figura 2.3 - Uma simples interseção T controlada por semáforos



Fonte: [CURY, 2001]

2.8 FORMAS POSSÍVEIS PARA MODELAGEM

[HAMILTON et al, 2006] destacam que a modelagem de um sistema a eventos discretos pode ser feita de três formas possíveis:

1. Linguagem de programação convencional (exemplo: C, C++, Java), que provoca uma sobrecarga de informações, onde toda linha de código é um detalhe de como o sistema deve efetivamente trabalhar.

2. Linguagem informal (exemplo: linguagem natural, tal como inglês e português), que gera ambiguidade e confusões excessivas nas descrições textuais.
3. Linguagem formal (exemplo: geradas por Autômatos, geradas por Redes de Petri), que evita ambiguidade. Idealmente, uma linguagem de modelagem formal deve ter uma notação simples com significado bem definido. A notação da linguagem deve ser pequena o bastante para ser aprendida facilmente e deve ter uma definição ambígua do significado da notação.

Existem na literatura diversas abordagens que apresentam metodologias, linguagens e padrões para a modelagem de aplicações industriais. As principais abordagens utilizadas são em linguagem formal e consistem da evolução dos autômatos de estados finitos.

Por exemplo, do ponto de vista da modelagem [ZHOU et al, 1993], apresenta como alternativa a modelagem e a análise dos SFMs utilizando Rede de Petri, bem como a análise do tempo de ciclo através da conversão da Rede para um Grafo Marcado com o uso de técnicas de redução. [KUO & HUANG, 2000] utilizam a Rede de Petri Colorida para a modelagem, análise e diagnósticos de falha para os SFMs. [SAITOU et al, 2002] utilizam Rede de Petri Colorida para a modelagem dos recursos e o planejamento da produção em um sistema flexível de manufatura, onde considera o objetivo de redução de custo, através do uso de um algoritmo genético que tem como regra o menor tempo de operação.

[FANTI, 2004] propõe uma estratégia de controle para gerenciar os recursos em sistemas que permitem o compartilhamento dos mesmos com a finalidade de prevenir os *deadlocks* utilizando a modelagem por Rede de Petri Colorida. [GIUA & BASILE, 2004] mostram o uso de um observador de estado para estimar o estado da planta, isto é, as marcas de uma Rede de Petri. [LI et al, 2009] também utiliza a modelagem como estratégia para prevenção de *deadlock* em Sistemas Flexíveis de Manufatura através de Rede de Petri.

Na área de automação, [CHAMBERS et al, 2001] destacam argumentos que apontam para o uso de metodologias baseadas em máquinas de estados por ser o formalismo mais conhecido pelos engenheiros. Porém, é de conhecimento geral entre os pesquisadores da área que esses diagramas, em sua forma básica, geram o problema de crescimento exponencial de estados para a modelagem de aplicações complexas que envolvem paralelismo [WU et al., 2011].

Outro formalismo explorado para o processo de modelagem e verificação de aplicações industriais são as Redes de Petri [MURATA, 1989], [MACIEL et al., 1996].

Uma visão geral das Redes de Petri voltada para a área de automação industrial está baseada na referência [MORAES, 2010].

2.9 MODELOS EXISTENTES

Foram desenvolvidos até o momento vários modelos para SEDs sem que nenhum tivesse afirmação como universal. Esses modelos refletem diferentes tipos de SEDs em como diferentes objetivos na análise dos sistemas em estudo.

Alguns modelos mais utilizados para sistemas a eventos discretos são:

1. Redes de Petri com e sem temporização [LIMA & OLIVEIRA, 2007], [DAVID & ALLA, 2005];
2. Cadeias de Markov [BOLCH et al., 2006];
3. Teoria de Linguagens e Autômatos [KESTEN et al., 1998], [RAMADGE, 1989];
4. Rede de Petri Colorida [MENESES & BARROSO, 2011], [MENESES; et al, 2012], [JENSEN & KRISTENSEN, 2009], [CASSANDRAS & LAFORTUNE, 2008], [ROBIDOUX et al., 2010], [FANTI, 2004], [SAITOU et al., 2002], [KUO & HUANG, 2000].

2.10 CONSIDERAÇÕES FINAIS

Caracterizou-se neste capítulo, a classe denominada Sistemas a Eventos Discretos, classe esta de sistemas foco de estudo ao longo desta dissertação. Apresentou-se a fundamentação teórica, importante para a continuidade dessa pesquisa, pois seus formalismos serão empregados na representação destes sistemas. Discutiram-se ainda, formas possíveis de modelagem e também a demonstração de modelos já existentes, fatores de contribuição na execução da presente pesquisa.

3 CONTROLADORES LÓGICOS PROGRAMÁVEIS (CLP)

De acordo com a Associação Brasileira de Normas Técnicas (ABNT), o controlador lógico programável (CLP ou do inglês, *Programable Logic Computer* – PLC) é um equipamento eletrônico digital com hardware e software compatíveis com aplicações industriais.

A *National Electrical Manufacturers Association* (NEMA), de acordo com a *International Electrotechnical Commission* (IEC), segundo a norma IEC 61131-1, define CLP como sendo um aparelho eletrônico digital que utiliza uma memória programável para o armazenamento interno de instruções em implementações específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética para controlar, através de módulos de entradas e saídas, vários tipos de máquinas ou processos.

O CLP deve ser projetado para operar em ambiente industrial sujeito a condições ambientais adversas, por exemplo, ambientes com temperaturas elevadas e/ou empoeirados.

O princípio básico de funcionamento de um CLP é a execução repetitiva do de um ciclo de ações enquanto estiver em funcionamento, chamada de varredura ou *scan* do CLP. Nesse ciclo têm-se as seguintes atividades: leitura das entradas; execução do programa; e, atualização das saídas. Durante a leitura das entradas, o CLP captura todas as informações dos sensores e as armazena em uma memória denominada memória de entrada. Após o armazenamento das informações de entrada, a programação do CLP será executada. Os resultados da execução do programa serão armazenados em outra memória denominada, memória de saída. Por fim, as informações contidas na memória de saída são recuperadas e atualizam-se todas as saídas do CLP.

O tempo total para execução dessas tarefas, chamado de ciclo de varredura ou *PLCcycle scan*, depende da velocidade e características do processador utilizado, do tamanho do programa de controle, além da quantidade e tipo de pontos de entrada/saída.

Inicialmente, a forma básica de programação foi realizada através de uma linguagem oriunda dos diagramas elétricos de relés, chamada *Ladder* pois todos os fabricantes de CLPs adotaram *Ladder* como a linguagem padrão para programação de seus equipamentos [SILVEIRA & SANTOS, 1999], por esse aspecto é que será utilizado tal linguagem nesta pesquisa. A princípio, isso popularizou os CLPs nos ambientes industriais pela simples tradução dos diagramas elétricos de relés em diagramas *Ladder*, porém, em

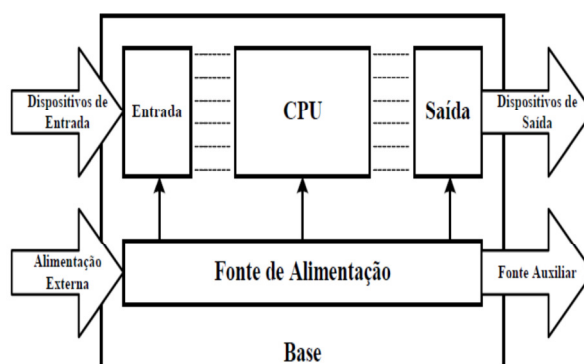
um momento posterior, provocou a dependência por técnicos com conhecimentos específicos nessa linguagem. Consequentemente, linguagens, métodos e padrões de mais alto nível têm sido propostos nos meios acadêmicos e profissionais através de organizações como a *International Electrotechnical Commission* (IEC) [HTTP://www.iec.ch/] e *PLCopen* [HTTP://plcopen.org], na tentativa de maximizar a eficiência para a construção e manutenção de sistemas de automação.

Em situações críticas onde o processo não pode esperar todo o tempo de execução do ciclo de varredura, por exemplo, quando ele deve executar uma rotina imediatamente ou quando o sinal de entrada é emitido por um tempo inferior ao tempo do ciclo, o CLP deve ter a capacidade de interromper a execução do ciclo de varredura para, prioritariamente, atender tais situações. Entretanto, apesar de não ser regra geral, a compreensão do ciclo de varredura é suficiente para se conhecer o princípio de funcionamento de um CLP.

O CLP e seus periféricos são projetados de forma a poder ser integrados dentro de um sistema de controle industrial e, finalmente, usados em todas as funções às quais são destinados.

Os principais blocos que compõem um CLP (Figura 3.1) são: unidade central de processamento (CPU); módulos de entrada e saída (do inglês, *In/Out* – *I/O*); fonte de alimentação; e, base ou *rack*.

Figura 3.1 - Estrutura Básica de um CLP



Fonte: [EBEL, 2006]

A unidade central de processamento (CPU) compreende o processador, o sistema de memória e os circuitos auxiliares de controle. O sistema de memória compreende a Memória de programa e as memórias de dados. A memória de programa armazena as instruções do software aplicativo e do usuário (programas que controlam a máquina ou a

operação do processo), que são continuamente executados pela CPU. Pode ser memória RAM, EPROM, EEPROM, NVRAM ou FLASH-EPROM. As memórias de dados armazenam temporariamente os estados de entrada/saída (I/O), marcadores ou *presets* de temporizadores, contadores e valores digitais para que a CPU possa processá-los. A cada ciclo de varredura, a memória de dados é atualizada. Geralmente é uma memória RAM, sendo também conhecida como memória de rascunho.

Os módulos de entrada/saída (I/O). Podem ser discretos com sinais digitais, contatos normalmente abertos, contatos normalmente fechados ou analógicos.

A fonte de alimentação é o dispositivo responsável pela tensão de alimentação fornecida à CPU e aos Módulos (circuitos) de I/O. Em alguns casos proporciona saída auxiliar de baixa corrente.

A base ou *rack* serve de conexão mecânica e elétrica entre a CPU, os módulos de I/O e a fonte de alimentação. Contem o barramento de comunicação entre os dispositivos, no qual os sinais de dados, endereço, controle e tensão de alimentação estão presentes.

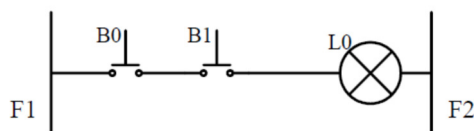
3.1 OPERAÇÃO BÁSICA DO CLP

A CPU controla todas as ações do CLP executando a leitura das condições e estados dos dispositivos por meio dos módulos de I/O. Essas condições são armazenadas em memória para serem processadas pelo programa de aplicação desenvolvido pelo usuário e armazenado na memória no CLP. O processador atualiza os status dos dispositivos de saída por meio dos módulos de I/O, realizando a lógica de controle, para garantir o ciclo de varredura.

A programação pode ser feita através de uma programação manual ou com software de programação no computador para posterior transferência. De acordo com a norma IEC 61131-3 que define as linguagens utilizadas nos CLPs, *Ladder Diagram (LD)*, *Function block diagram (FBD)*, *Structured text (ST)*, *Instruction list*, *Sequential function chart (SFC)*, tais linguagens citadas foi escolhida a *Ladder Diagram (LD)*, pois é a linguagem padrão de todos os controladores existentes. Esta linguagem é a representação lógica da sequência elétrica de operação. Na Figura 3.2, tem-se um exemplo de diagrama que mostra uma lógica de contato elétrico, onde os contatos dos botões B0 e B1 estão colocados em

série com L0, que representa uma lâmpada. Ao pressionar os botões B0 e B1, simultaneamente, a lâmpada L0 entra em funcionamento.

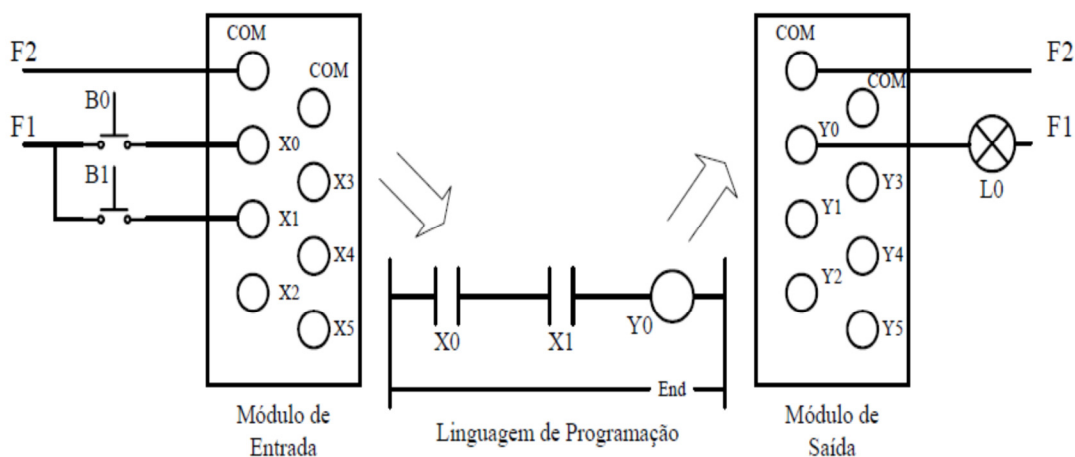
Figura 3.2 - Lógica de contato elétrico



Fonte: [EBEL, 2006]

Na Figura 3.3, tem-se outro exemplo, mais complexo, onde a mesma função realizada na lógica de contato anterior (Figura 3.2) é feita de forma diferente. Nesta, os botões B0 e B1 estão ligados no cartão de entrada do CLP e a lâmpada está ligada no cartão de saída do CLP. A lógica utilizada em contatos elétricos (Figura 3.2) é feita através de ligações elétricas. No, caso do CLP a lógica é realizada através de software de programação, o que determina diagrama diferentes a depender da linguagem utilizada. Particularmente, a linguagem utilizada na Figura 3.3 é a linguagem *Ladder*, que tem uma grande semelhança com a linguagem de contatos elétricos.

Figura 3.3 - Lógica por meio do CLP



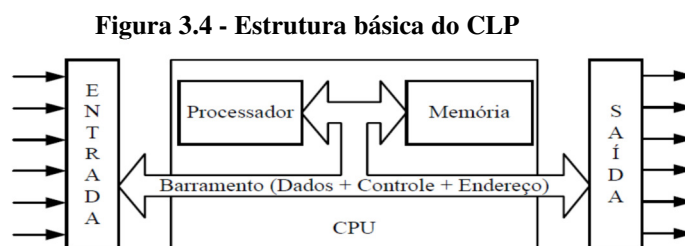
Fonte: [EBEL, 2006]

De acordo com as Figuras 3.2 e 3.3 vê-se que a lógica implementada pelo CLP é muito similar à convencional [SILVEIRA & SANTOS, 1999], sendo que os dispositivos de entrada (chaves B0 e B1) são conectados aos módulos de entrada e os dispositivos de

saída (lâmpada L0) aos módulos de saída. O programa de aplicação determina o acionamento da saída em função das entradas. Qualquer alteração desejada nesta lógica é realizada por meio de alterações no programa, permanecendo as mesmas conexões nos módulos de I/O.

3.2 ARQUITETURA BÁSICA DO CLP

A Figura 3.4 ilustra a estrutura básica de um CLP que compreende o processador, o sistema de memória, os barramentos de dados, de endereços e de controle, além dos circuitos auxiliares de controle, onde as entradas e saídas se comunicam com a CPU através de um barramento óptico.



Fonte: próprio autor

A Figura 3.5 apresenta três modelos de CPUs disponíveis em uma mesma família de CLP, relacionando algumas de suas características, tais como: número de entrada e saída, tipo modular ou compacto, protocolo de comunicação, tensão de alimentação, tipos de saídas (transistor, triac e relé) e saídas e entradas analógicas.

Figura 3.5 - Exemplos de algumas CPU's disponíveis no mercado



Fonte: [EBEL, 2006]

3.3 CONSIDERAÇÕES FINAIS

Apresentaram-se neste capítulo, as configurações básicas do Controlador Lógico Programável, sua arquitetura de *hardware*, a normatização da linguagem do *software* utilizado, como também, a descrição do princípio de operação de leitura das entradas e saídas, execução de programas e processamento de informações. Com isso tem-se a ferramenta onde será validado o modelo proposto neste estudo.

4 REDES DE PETRI

Este capítulo tem como finalidade apresentar o formalismo de modelagem por redes de Petri (RP). Apresenta-se a definição, a estrutura, classificação e as propriedades da rede de Petri. No final do capítulo aborda-se o modelo matemático da rede de Petri e ainda a definição e a estrutura da rede de Petri colorida.

4.1 FORMALISMO DE MODELAGEM POR REDE DE PETRI

Segundo [MURATA, 1989] uma rede de Petri (RP) é uma ferramenta de extrema importância, pois permite: modelar o comportamento de sistemas dinâmicos a eventos discretos; descrever as relações existentes entre condições e eventos; visualizar propriedades tais como, paralelismo, sincronização e compartilhamento de recursos.

Utilizando o conceito de rede de Petri, é possível simular o funcionamento de um sistema ou mesmo realizar análises de seu desempenho [ZHOU et al, 2010]

A rede de Petri utiliza uma ferramenta gráfica para visualização das estruturas e das relações entre os sistemas modelados, que pode ser comparada com Diagramas de Bloco, Diagramas de Sequência ou conceito de grafos.

Segundo [HEUSER, 1990] as primeiras aplicações de RP aconteceram em 1968, no projeto norte-americano *Information System Theory*, da A.D.R. (*Applied Data Research, Inc.*). Muito da teoria inicial, da notação e da representação de RP foi desenvolvido naquele projeto e foi publicado em seu relatório final, destacando-se sua aplicação na análise e na modelagem de sistemas com componentes concorrentes.

A década de setenta marcou o desenvolvimento da teoria de RP e a expansão de seu campo de aplicação. Embora ocorresse uma ampla divulgação acadêmica ao longo de vários anos, o seu potencial só foi amplamente reconhecido na metade da década de oitenta, onde essa teoria foi usada para implementações práticas nas áreas de informática e manufatura devido à disponibilidade de compartilhamento de novos recursos computacionais [VALENCIA, 2004]. Na década de oitenta surgiram às chamadas redes de Petri de alto nível, como por exemplo, as numéricas, as de predicado/transição [GENRICH & LAUTEBACH, 1978] e as coloridas [JENSEN, 1983]. Surgiram também extensões de RP estocásticas [MARSAN & CONTE, 1984], assim suas utilizações expandiram-se.

As redes de Petri começaram a ser utilizadas para representar modelos conceituais de um sistema real. Conforme [HEUSER, 1990], um modelo conceitual é um modelo de uma área de uma organização, o qual não envolve detalhes de implementação e descreve tanto as propriedades estáticas quanto as propriedades dinâmicas do sistema modelado.

Conforme [CARDOSO & VALETTE, 1997], redes de Petri também são aplicadas para avaliação de desempenho, análise e verificação formal de sistemas discretos, tais como: protocolos de comunicação, controle de oficinas de fabricação, concepção de software em tempo real e/ou distribuído, sistemas de informação (organização de empresas), sistemas de transporte, logística, gerenciamento de base de dados, interface homem-máquina e multimídia.

Como ferramentas matemáticas e gráficas, as RP servem para oferecer um ambiente uniforme para a modelagem, análise formal e simulação de sistemas e eventos discretos, permitindo uma visualização simultânea da sua estrutura e comportamento.

Modelos de redes de Petri podem ser aplicados em qualquer área ou sistema que possa ser descrito graficamente e que necessite representar atividades paralelas ou concorrentes [MURATA, 1989]. Essa flexibilidade pode ser notada na literatura, pois o conceito de rede de Petri está sendo difundido e aplicado nas mais diversas áreas como: computacional, com elaboração de modelos para análises de Base de Dados Distribuídos [VOOS, 1980] ou modelos para análises de programas paralelos ou concorrentes [MURATA, 1989]; e, na área industrial, com elaboração de modelos de sistemas de controle da indústria ou da produção. Por isso, o uso de redes de Petri para a modelagem de sistemas de manufatura tem sido muito pesquisado [PROTH et al. 1997], [DICESARE et al. 2002], [DESROCHERS & AL-JAAR, 1995], [ZHOU et al. 1992], ao formalismo de redes de Petri cabe disponibilizar um arcabouço para especificação e verificação de SFM.

Face ao exposto, nesta pesquisa utilizam-se redes de Petri, especificamente redes de Petri Coloridas Hierárquicas (*HCPN – Hierarchical Coloured Petri Nets*), [JENSEN et al. 2007], para a especificação formal de Sistemas Flexíveis de Manufatura (SFM) e para a visualização do comportamento dos modelos. Além de trabalharem com marcas diferenciáveis, tais redes apresentam tratamento de aspectos temporais e que permitem a representação de tipos de dados abstratos, diferencial que as outras extensões existentes no final da década de oitenta e início da de noventa não apresentavam.

4.2 DEFINIÇÃO DA REDE DE PETRI

O conceito de Rede de Petri (RP) foi introduzido por Carl Adam Petri (1962), em sua tese de doutorado em 1962. Trata-se de uma ferramenta de modelagem gráfica e matemática, bastante flexível e que possui grande capacidade de representação, tanto dos sistemas mais simples como de sistemas mais complexos.

As redes de Petri possuem grande poder de expressão e podem ser utilizadas para modelar sistemas paralelos, concorrentes, assíncronos e não determinísticos. Têm sido aplicados em variadas áreas, como na modelagem de sistemas de hardware/software, sistemas dinâmicos a eventos discretos, protocolos de comunicação e análise de desempenho [MACIEL et al. 1996].

Segundo [PETERSON, 1981] o estudo de sua teoria permite que um sistema modelado por uma RP possua representação matemática, e está sendo muito popular na modelagem de sistemas concorrentes. Além disso, a representação do sistema permite que o modelo seja simulado e o seu comportamento seja observado dinamicamente.

Entre as principais características dessa ferramenta estão a facilidade de interpretação gráfica, identificação de estados e ações numa maneira clara e a possibilidade de representação de sistemas dinâmicos em diferentes níveis de detalhe [MURATA, 1989], [PETERSON, 1981].

Segundo [ZHOU & ZHURAWSKI, 1994] como ferramenta matemática, um modelo em rede de Petri pode ser descrito por um sistema de equações lineares ou outros modelos matemáticos que refletem o comportamento do sistema, o qual possibilita a análise formal do mesmo. Esta característica permite realizar a verificação formal das propriedades comportamentais do sistema.

Segundo [ARATA, 2005] a representação dos sistemas físicos através da rede de Petri permite a verificação das chamadas “boas qualidades” como: limitação, vivacidade e reversibilidade, que em termos práticos correspondem respectivamente à existência de um número finito de estados válidos, a não existência de auto travamento (*deadlocks*) e a possibilidade do processo ser repetido infinitamente sem a sua degradação.

Segundo [HASEGAWA, 1996] a rede de Petri tem como características: a representação da dinâmica e a estrutura do sistema segundo o nível de detalhamento desejado; a identificação do estado e ações de modo claro e explícito, que facilita a monitoração do sistema em tempo real; a representação de forma natural das características

funcionais dos SEDs (sincronização, assincronismo, concorrência, causalidade, conflito, compartilhamento de recursos, etc.), e ainda possui uma semântica formal que permite que o mesmo modelo possa ser utilizado tanto para a análise de propriedades comportamentais (análise qualitativa e/ou quantitativa).

Além disso, sua executabilidade torna-se extremamente favorável para simulação, rápida prototipação e programação, podendo ser considerada como uma linguagem de nível similar ao *Prolog* e *Smalltalk*.

Uma Rede de Petri é descritiva como um tipo particular de grafo orientado. As Redes de Petri são como um multigrafo [PETERSON, 1981], pois permite múltiplos arcos de um nó para outro, com peso e bipartido que permite modelar as propriedades estáticas de um sistema a eventos discretos.

Utiliza-se nessa pesquisa como aporte norteador as definições de [MURATA, 1989], [MACIEL, 1996] e [ARATA, 2005].

4.3 ESTRUTURA DA REDE DE PETRI

Desde o surgimento do conceito de rede de Petri, muitos grupos de pesquisas trabalharam para desenvolver a teoria básica, no qual um modelo de rede de Petri é composto basicamente por quatro componentes: Lugares, Transições, Arcos e Marca, e chamado de Modelo Condição-Evento [PETRI, 1962]. Como a utilização das redes de Petri cresceu significativamente, novos componentes foram criados com o intuito de aumentar o poder de abstração e modelagens.

Lugar é o elemento passivo responsável por armazenar as informações correspondentes às condições que devem ser certificadas para que os eventos ocorram. Modela a condição ou estado de um agente, um componente de hardware ou software, definindo o estado do sistema. Um lugar é representado por uma Figura geométrica circular, podendo ser representado como é mostrado na Figura 4.1,

Figura. 4.1 – Componente Lugar



Lugar

Fonte: [MURATA, 1989]

Lugares podem ser marcados ou não. Um lugar marcado é representado por um ou mais pontos desenhados dentro do círculo que representa o lugar. Esses pontos são chamados de fichas (*tokens*). A quantidade de *fichas* que um lugar possui é o que define o seu estado e é chamado de marcação deste lugar. O conjunto de lugares da rede de Petri forma o conjunto das variáveis de estado que representam o sistema modelado.

Assim, as marcações de todos os lugares em um dado momento é uma característica importante da rede, representando o estado do sistema naquele momento, a Figura 4.2 mostra a marcação sendo representada por um círculo. Esse estado global, composto pelos estados de cada lugar, é chamado de marcação de rede.

Figura 4.2 - a) Lugar com um *token* e b) com vários *tokens*



Fonte: [MURATA, 1989]

A Transição é o elemento encarregado de realizar a caracterização das mudanças de estado do sistema, modelando eventos, operações e estruturas de decisão, entre outros, a Figura 4.3 mostra uma transição onde é representada uma Figura geométrica que é um retângulo.

Figura 4.3 - Componente Transição



Fonte: [MURATA, 1989]

As transições, por outro lado, são elementos ativos que representam ações que podem ocorrer e que modificam o estado do sistema (marcação da rede). Os lugares que estão ligados a uma transição, através de arcos, são o que define quando uma transição pode ocorrer e como o estado é modificado após a sua ocorrência.

O componente Arco é o elemento que interliga lugares a transições, ou vice-versa, encadeando condições e eventos, a Figura 4.4 é composta de uma linha com uma flecha na sua extremidade representando o arco.

Figura 4.4 - Componente Arco



Fonte: [MURATA, 1989]

Um arco ligando um lugar a uma transição é mostrado na Figura 4.5 que representa uma condição que deve ser verdadeira para que aquela transição ocorra. Chama-se a esse lugar de pré-condição de transição [MACIEL et al. 1996].

Figura 4.5 - Ligação Lugar x Transição



Fonte: [MURATA, 1989]

Para cada arco que liga qualquer lugar a essa transição deve existir pelo menos um *token* naquele lugar. Se esta condição for satisfeita, diz-se que a transição está habilitada para ocorrer [MURATA, 1989]. Já um arco que liga a transição a um lugar representa a condição que se torna verdadeira após a ocorrência da transição (Figura 4.6). Esse lugar é denominado de pós-condição de transição [MACIEL et al. 1996].

Figura 4.6 - Ligação Transição x Lugar



Fonte: [MURATA, 1989]

Assim, quando essa transição ocorrer, deverá ser adicionada uma ficha ao lugar que é sua pós-condição para cada arco que liga a transição a este lugar.

As fichas (*tokens*) existentes nas pré-condições da transição e que a tornaram habilitada são removidos após sua ocorrência. Diz-se que eles foram “consumidos” pela transição. Todo esse processo que acontece quando uma transição ocorre é chamado de disparo da transição.

Quando mais de um arco liga os mesmos elementos da rede, digamos n arcos, é convencionalmente representado por apenas um arco, acompanhando da notação numérica (rótulo) “ n ”, indicando que se trata de um arco de peso n , ou seja, que tem o mesmo valor de n arcos [MURATA, 1989].

A tarefa de avaliar as transições habilitadas e de proceder aos seus disparos, levando a rede a passar por uma sequência de estados ao longo do tempo é a simulação da rede, também chamada de “*token game*”.

A simulação é uma ferramenta importante para estudo do sistema, pois, através dela é possível compreender vários aspectos sobre o seu comportamento.

A Marca ou Ficha: representa um recurso disponível ou uma característica habilitada. O posicionamento dessas fichas em alguns lugares do grafo constitui a marcação. Sua evolução permite modelar o comportamento dinâmico do sistema. Uma marca (Figura 4.7) numa rede de Petri é definida como uma entidade que reside nos lugares, e representa as pré/pós-condições de disparo de uma transição que serão checadas e alteradas pelas transições, caso estas venham a disparar [PALOMINO, 1995].

Figura 4.7 - Marca



Fonte: [MURATA, 1989]

Assim, cada evento de um sistema pode ser representado por uma transição, sendo sua ocorrência representada por um disparo desta transição.

As pré e pós-condições (ou entradas e saídas do sistema) são representadas pelos lugares, de forma que os arcos direcionados realizam a interligação dos lugares que representam as pré-condições (entradas) com as transições e das transições com os lugares

de pós-condições (saídas), permitindo deste modo o fluxo das marcas, que simbolizam os dados, através do sistema [PALOMINO, 1995].

Em [PALOMINO, 1995] é definido que, caso o peso dos arcos que ligam lugares e transições seja igual a 1 (um), a Rede de Petri é dita de Rede de Petri Ordinária. Nos casos onde sejam permitidos múltiplos arcos (utilização de peso nos arcos), ele a denomina de Rede de Petri Generalizada.

Do ponto de vista formal, uma Rede de Petri é uma quintupla, $PN = (P, T, F, W, M_0)$ [MURATA, 1989]:

$P = \{p_1, p_2, \dots, p_m\}$ 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 (relação de fluxo),

$W: F \rightarrow \{1, 2, 3, \dots\}$ é a função peso,

$M_0: P \rightarrow \{1, 2, 3, \dots\}$ é a marcação inicial (quantidade de marcas em cada lugar),

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

A estrutura de uma Rede de Petri $N = (P, T, F, W)$ sem qualquer especificação de marcação inicial é denotada por N [MURATA, 1989].

Uma Rede de Petri com uma dada marcação inicial é denotada por (N, M_0) [MURATA, 1989].

As fichas residem apenas em lugares, representando a disponibilidade de recursos ou o status da operação, por exemplo, [PALOMINO, 1995].

Trabalhos mais teóricos de Redes de Petri são baseados na definição formal das estruturas descrita acima. Contudo, uma representação gráfica de sua estrutura é muito mais útil para ilustrar os conceitos da teoria de Redes de Petri [PETERSON, 1981].

As Figuras 4.8 e 4.9 ilustram as equivalências das duas representações (formal e gráfica), exemplificando a forma de tradução do modelo formal para o gráfico e vice-versa, no qual uma mesma Rede de Petri é apresentada nas duas notações:

Figura 4.8 - Notação Formal

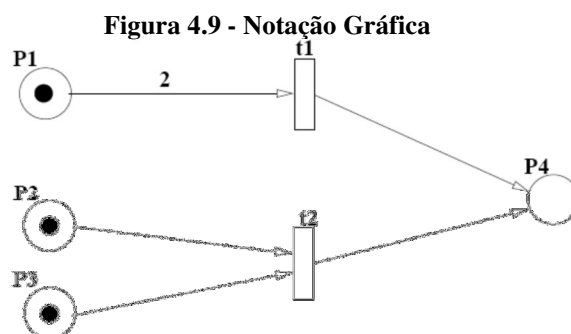
$$P = \{P_1, P_2, P_3, P_4\}$$

$$T = \{t_1, t_2\}$$

$$F = \{(P_1, t_1), (P_2, t_2), (P_3, t_2), (t_1, P_4), (t_2, P_4)\}$$

$$W(P_1, t_1) = 2, W(P_2, t_2) = W(P_3, t_2) = W(t_1, P_4) = W(t_2, P_4) = 1$$

$$M_0 = [11110]^T$$



Fonte: [MURATA, 1989]

Cada lugar de uma Rede de Petri indica um determinado estado. A marcação inicial do sistema representa o seu estado inicial e consiste no número de marcas contidas nos lugares antes de qualquer ocorrência de eventos na rede [PALMA, 2001] [MURATA, 1989].

A execução de uma Rede de Petri é controlada pelo número e distribuição de marcas na rede. Marcas residem nos lugares e controlam a execução das transições. Uma Rede de Petri é executada através do disparo de transições [PETERSON, 1981].

4.4 CLASSIFICAÇÃO DA REDE DE PETRI

As redes de Petri podem ser divididas em diversas classes, cada uma delas com seu próprio poder de modelagem e características peculiares. A classificação aqui apresentada é a utilizada por [MACIEL et al. 1996], essa é uma entre as diversas encontradas na literatura. Assim, considera-se neste trabalho que as redes podem ser ordinárias e não ordinárias ou de alto nível.

As Redes de Petri Ordinárias se caracterizam pelo tipo de suas marcas. São exemplos nessa classe as RP Binárias ou Condição-Evento, RP *Place-Transition*, RP Não-Ponderadas e RP Ponderadas. Enquanto que as Redes de Petri Não-Ordinárias ou de Alto Nível são aquelas cujas marcas incorporam alguma semântica, viabilizando sua diferenciação. Esta semântica pode ir desde a atribuição de valores ou cores às marcas, até a adoção de noções de tipos de dados abstratos, conferindo-lhes um grande poder de

expressão. São exemplos nessa classe RP Predicado-Evento, RP Hierárquicas e RP Coloridas.

Desta forma, a complexidade da RP, tanto ordinária quanto a não ordinária, encontra-se dividida: parte é representada pela própria estrutura da rede – tal como sucede nas RP ordinárias – e outra parte é representada pelas inscrições da rede.

Numa rede de alto nível em que cada lugar apenas possa conter, no máximo, uma marca pode pensar nos lugares variáveis, e nas marcas como valores de variáveis. Este tipo de redes surge como uma generalização das RP seguras ou binárias. Cada lugar já não corresponde necessariamente a uma variável *Booleana* (presença ou ausência de uma marca), mas pode corresponder a qualquer tipo de variável.

Se cada lugar pode conter mais do que uma marca (caso geral) podemos ver o lugar como uma estrutura de dados que contem um ou mais valores de um determinado tipo. Esta estrutura de dados é geralmente vista como um multiconjunto. O multiconjunto é a estrutura de dados que associada ao lugar, menos o afasta do lugar da RP ordinária.

No presente estudo será utilizada a rede de Petri de alto nível, especificamente a Rede de Petri Colorida, devido à complexidade do SED e ainda, por trata-se de um conjunto de variáveis distintas, onde há necessidade de associar essas variáveis, de forma a gerar elementos específicos com características próprias, fazendo-se necessário a utilização da Rede de Petri Colorida por a mesma utiliza-se de recursos de linguagem de programação, que facilita a associação, o que com outro tipo de rede de Petri, o nível de complexidade seria maior.

4.5 PROPRIEDADES DA REDE DE PETRI

As propriedades das RP encontram-se extensamente documentadas [MURATA, 1989], [DAVID & ALLA, 2005], [PETERSON, 1981], [ZHOU & ZHURAWSKI, 1994]. Em [MURATA, 1989] as propriedades das RP são classificadas em comportamentais ou estruturais conforme sejam ou não dependentes de marcação inicial.

As propriedades comportamentais segundo [MURATA, 1989] são: a alcançabilidade, onde um estado x , é dito alcançável a partir do estado inicial x_0 , caso exista uma sequência de transição iniciando-se a partir de x_0 , de tal forma que o estado pode se tornar x ; a limitabilidade é quando o número de fichas em cada lugar não excede o

número finito para qualquer marcação alcançável a partir da marcação inicial; a vivacidade ocorre quando independente da marcação que for alcançada a partir da marcação inicial, for possível disparar qualquer transição da rede através de alguma sequência de disparo ou simplesmente que não possua deadlocks; a reversibilidade ocorre quando for possível disparar a partir de qualquer marcação acessível uma sequência que leve a rede a uma marcação inicial; a cobertura são grafos onde cada par de vértices é ligado por um e somente um caminho simples; a persistência ocorre quando para cada duas transições habilitadas, o disparo de uma delas não desabilita a outra; à distância sincrónica, está relacionada ao grau de dependência mútua entre duas transições em uma rede de Petri; e a justiça, que serve para qualquer par de transições, onde o número de disparo de uma transição enquanto a outra não disparar, fica limitado.

Já as propriedades estruturais são: a vivacidade estrutural é quando existe uma marcação viva para N ; a controlabilidade ocorre quando qualquer marcação for atingível de qualquer marcação; a limitabilidade estrutural ocorre para qualquer marcação inicial finita; a controlabilidade ocorre quando qualquer marcação é atingível a partir de qualquer marcação; a receptibilidade, consistência, justiça-B estrutural.

A execução de uma RP é controlada pelo número e distribuição de marcas na rede. Marcas residem nos lugares e controlam a execução das transições.

Uma Rede de Petri é executada através do disparo de transições [PETERSON, 1981].

Uma transição dispara removendo marcas dos lugares de entrada e criando novas marcas que são distribuídas para seus lugares de saídas [PETERSON, 1981].

Uma transição pode disparar se estiver habilitada. Uma transição é dita habilitada se cada um de seus lugares de entrada possuir pelo menos uma quantidade de marcas maior ou igual ao peso dos arcos que conectam esses lugares a esta transição [PALMA, 2001], [PETERSON, 1981].

Existem algumas transições especiais que não possuem lugares de entrada ou de saída, e recebem denominações diferentes de acordo com os elementos que possui. Uma transição sem lugares de entrada é denominada Transição Fonte, e uma que não possua lugares de saída são denominadas Transição Sumidouro. Uma transição fonte sempre está habilitada. O disparo de uma transição sumidouro consome fichas, mas não produz nenhuma [PALMA, 2001].

A RP provê vantagens quanto a sua utilização, sendo usada com sucesso para modelar, controlar e analisar sistemas a eventos discretos, que são caracterizados pela concorrência dos processos, pelo paralelismo, por serem assíncronos, por possuírem *deadlocks*, conflitos e processos de eventos direcionados, características constantes em Ambientes Virtuais. Também tem sido estudada extensivamente nas últimas três décadas, resultando na definição de numerosas extensões e técnicas para análise, permitindo a implementação de análise em tempo real, característica fundamental para realidade virtual.

No presente estudo a RP traz como vantagem a possibilidade do controle e da análise do SED em tempo real.

4.6 O MODELO MATEMÁTICO

O modelo das redes de Petri pode ser descrito de formas variadas. Existe uma abordagem que as define em termos da teoria de *bag* (extensão de conjunto que aceita repetição de elementos), outra que as define em termos de álgebra matricial e também uma baseada no conceito de relações.

No presente estudo, será utilizada a representação matricial, que é definida como se segue:

Definição

Uma Rede de Petri é uma quintupla, $PN = (P, T, F, W, M_0)$ [MURATA, 1989], onde,

P é o conjunto finito de lugares;

T é o conjunto finito de transições;

$F: P \times T \rightarrow N$ é matriz de entrada (ou de pré-condições);

$W: P \times T \rightarrow N$ é matriz de saída (ou de pós-condições);

M_0 é a marcação inicial

Para exemplificar a definição, a Figura 4.10 apresenta a suposição de que se deseja representar em um ano letivo de uma Universidade. O ano letivo começa com o primeiro semestre letivo, seguido das primeiras férias (de julho), logo após, tem-se o segundo semestre letivo, e finalmente as férias de final de ano. Assim, o ano letivo poderia ser representado conforme a Figura. $R_{\text{Ano_Letivo}} = (P, T, I, O, K)$, onde

o conjunto de lugares P é

$P = \{1^\circ \text{ Semestre, férias do } 1^\circ \text{ Semestre, } 2^\circ \text{ Semestre, férias do } 2^\circ \text{ Semestre}\};$

o conjunto de transições T é

$T = \{\text{GozarFérias}1^\circ\text{Semestre, Iniciar } 2^\circ\text{Semestre, GozarFérias}1^\circ\text{Semestre, Iniciar } 1^\circ\text{Semestre}\}$

o conjunto de bags de entrada I é

$I = \{I(\text{GozarFérias}1^\circ\text{Semestre}) = [1^\circ\text{Semestre}], I(\text{Iniciar } 2^\circ\text{Semestre Período}) = [\text{GozarFérias}1^\circ\text{Semestre}], I(\text{GozarFérias}2^\circ\text{Semestre}) = [2^\circ\text{Semestre}], I(\text{Iniciar } 1^\circ\text{Semestre Período}) = [\text{GozarFérias}2^\circ\text{Semestre}]\};$

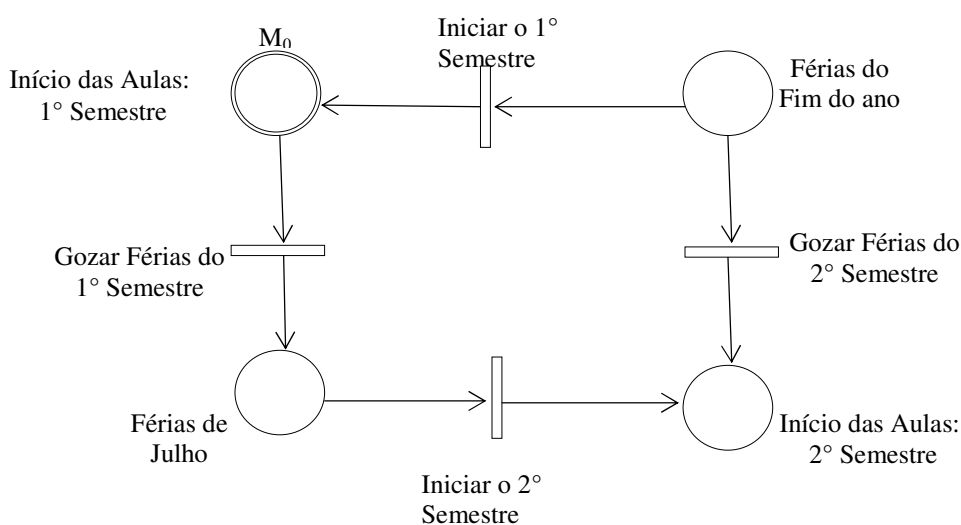
o conjunto de bags de saída O é

$O = \{O(\text{GozarFérias}1^\circ\text{Semestre}) = [\text{Férias de Julho}], O(\text{Iniciar } 2^\circ\text{Semestre Período}) = [2^\circ\text{Semestre}], O(\text{GozarFérias}2^\circ\text{Semestre}) = [\text{Férias do fim do ano}], O(\text{Iniciar } 1^\circ\text{Semestre Período}) = [1^\circ\text{Semestre}]\};$

E, o conjunto de capacidades dos lugares é

$K = \{K_{1^\circ \text{ Semestre}} = 1, K_{\text{Férias de Julho}} = 1, K_{2^\circ \text{ Semestre}} = 1, K_{\text{Férias fim de ano}} = 1\}.$

Figura 4.10 - Representação através de Rede de Petri do ano letivo



Fonte: [FRANCÊS, 2003]

Definição: a estrutura de uma rede de Petri, segundo o ponto de vista matricial, é uma quintupla $\mathbf{R} = (\mathbf{P}, \mathbf{T}, \mathbf{I}, \mathbf{O}, \mathbf{K})$, onde \mathbf{P} é um conjunto finito de lugares. \mathbf{T} é um conjunto finito de transições, $\mathbf{I}: \mathbf{P} \times \mathbf{T} \rightarrow \mathbf{N}$ é a matriz de pré-condições. $\mathbf{O}: \mathbf{P} \times \mathbf{T} \rightarrow \mathbf{N}$ é a matriz de pós-condições. \mathbf{K} é o vetor das capacidades associados aos lugares ($\mathbf{K}: \mathbf{P} \rightarrow \mathbf{N}$) [PETERSON, 1981].

Os conjuntos de lugares e transições são idênticos àqueles vistos para a definição 1.

A matriz \mathbf{I} (pré-condições) é

$$I = \begin{array}{c|cccc} & \text{Férias Julho} & \text{Iniciar 2ºsem} & \text{Ferias fim de ano} & \text{Iniciar 1ºSem} \\ \hline & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 1 \end{array}$$

A matriz \mathbf{O} (pós-condições) é:

$$O = \begin{array}{c|cccc} & \text{Férias Julho} & \text{Iniciar 2ºsem} & \text{Ferias fim de ano} & \text{Iniciar 1ºSem} \\ \hline & 0 & 0 & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \end{array}$$

É importante ressaltar que as matrizes \mathbf{I} e \mathbf{O} representam as pré e pós-condições, respectivamente, de todas as transições da rede.

Definição: a estrutura de redes de Petri, usando-se relações, é formada por uma quintupla $\mathbf{R} = (\mathbf{P}, \mathbf{T}, \mathbf{A}, \mathbf{V}, \mathbf{K})$, onde \mathbf{P} é o conjunto de lugares, \mathbf{T} o de transições, \mathbf{A} o conjunto dos arcos e \mathbf{V} corresponde ao conjunto de valorações desses arcos. Os elementos de \mathbf{A} são arcos que conectam transições a lugares ou lugares a transições ($\mathbf{A} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$). Assim, os elementos de \mathbf{A} podem ser agrupados em dois subconjuntos - o conjunto das entradas às transições e o de saída às transições, $\mathbf{I} = \{(\mathbf{p}_i, \mathbf{t}_j)\}$ e $\mathbf{O} = \{(\mathbf{t}_j, \mathbf{p}_i)\}$, respectivamente [MURATA, 1989].

Tem-se que os conjuntos de lugares (**P**), de transições (**T**) e de capacidades (**K**) permanecem inalterados. Entretanto, na notação que utiliza relações, há o surgimento de dois novos conjuntos: o conjunto de arcos (**A**) e o conjunto de valores para esses arcos (**V**).

O conjunto de arcos **A** é

$$\mathbf{A} = \{(1^\circ\text{Semestre, GozarFérias1}^\circ\text{Semestre}), (\text{GozarFérias1}^\circ\text{Semestre, Férias de Julho}), (\text{Férias de Julho, Iniciar 2}^\circ\text{Semestre}), (\text{Iniciar 2}^\circ\text{Semestre}), (2^\circ\text{Semestre o, Férias do Fim do ano}), (\text{GozarFérias2}^\circ\text{Semestre, Férias do Fim do ano}), (\text{Férias do Fim do ano, Iniciar 1}^\circ\text{Semestre}), (\text{Iniciar 1}^\circ\text{Semestre, 1}^\circ\text{Semestre})\}$$

O conjunto de valores dos arcos **V** é

$$\mathbf{V} = \{1, 1, 1, 1, 1, 1, 1, 1\}$$

O modelo apresentado possibilitou o melhor entendimento da RP e suas principais propriedades. A seguir será feita uma explanação da rede de Petri Colorida, que será utilizada para validação do modelo proposto no presente estudo.

4.7 REDE DE PETRI COLORIDA

As redes de Petri coloridas (RPCs) surgiram da necessidade em representar sistemas muito grandes e complexos, que são encontrados em aplicações industriais reais. Utilizando-se redes de Petri ordinárias, o tamanho desses sistemas se tornava um grande complicador para sua modelagem e estudo [MACIEL et al., 1996], [JENSEN,1994]. Assim, outra abordagem mais apropriada se fez necessária. A ideia por traz das redes de Petri coloridas é unir a capacidade de representar sincronização e concorrência das redes de Petri com o poder expressivo das linguagens de programação. Através dessa união, sistemas cujo estudo, anteriormente, era impraticável, tornaram-se passíveis de estudo.

Uma rede de Petri colorida é uma rede de Petri na qual as fichas (*tokens*) possuem um tipo e um valor associados, permitindo que eles possam ser diferenciados entre si. Desta forma, um lugar na rede de Petri passou a representar não apenas certa condição, mas toda uma classe de situações que podem se apresentarem de diferentes formas, de acordo com os valores das fichas presentes em sua marcação. Esses valores são então

utilizados em expressões que são calculadas durante a avaliação de transições, bem como durante o disparo destas.

No início, a diferenciação das fichas era feita por cores, daí o nome de rede de Petri colorida. Atualmente, são utilizados tipos de dados estruturados, permitindo que operações bastante complexas sejam representadas na rede [MACIEL et al., 1996]. As redes de Petri coloridas são, provavelmente, as redes de alto nível, que mais interesse tem levantado.

O salto que se deu ao passar das redes de Petri para as redes de Petri coloridas é, mais do que em qualquer outro tipo de rede de Petri de alto nível, extremamente semelhante ao que se dá ao passar da programação em linguagens *Assembly* para linguagens de alto nível como *Pascal* ou *C*.

O principal objetivo da RPCs é a redução do tamanho do modelo, permitindo que fichas individualizadas (coloridas) representem diferentes processos ou recursos numa mesma sub-rede. Elas recebem esse nome porque as fichas contêm dados capazes de distingui-las uma das outras em contraponto com as redes de Petri padrão, onde as fichas são indistinguíveis.

Em RPC, as fichas são representadas por estruturas de dados complexas. Desse modo, as fichas podem conter informações. Além disso, cada lugar armazena fichas de certo tipo definido e arcos realizam operações sobre elas. As transições determinam a dinâmica da RPC e podem apresentar “expressões de guarda”. Estas, por sua vez, indicam os tipos de fichas que possibilitam habilitar uma transição. Para que uma transição esteja habilitada em uma RPC, devem existir fichas suficientes nos lugares de entrada da transição. Além disso, essas fichas devem possuir valores que sejam correspondentes às expressões associadas aos arcos que ligam estes lugares à transição.

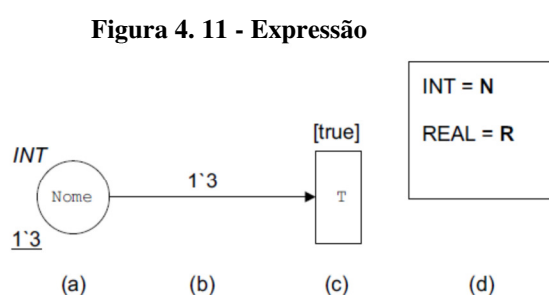
Uma RPC é composta por três partes: estrutura, inscrições e declarações [MACIEL et al., 1996].

A estrutura é um garfo direcionado, com dois tipos de nós (lugares e transições), com arcos interconectando nós de tipos diferentes. Graficamente, os lugares são representados por círculos ou elipses, e transições, simbolizadas por retângulos. Esse tipo indica o conjunto de valores que os *tokens* contidos neste lugar podem representar. O tipo de um lugar também é chamado de seu conjunto de cores (color set). As declarações envolvem tipos, funções, operações e variáveis. Quando a expressão do arco é avaliada, ela gera um multiconjunto (conjunto com vários tipos) de fichas coloridas. Por fim, as

inscrições são anotações associadas aos elementos da rede. Os tipos de inscrições são diferentes para lugares, transições e arcos.

Conforme ilustrado na Figura 4.11, as inscrições de lugares possuem três tipos de inscrições: nome, tipo e expressão de inicialização. O nome não possui valor semântico, sendo apenas um elemento que facilita a sua identificação. O tipo indica o conjunto de possíveis valores associados ao lugar. O lugar só poderá conter fichas com valores deste tipo. A expressão de inicialização, por sua vez, é uma expressão escrita na linguagem de representação adotada. Esta expressão indica a marcação inicial a ser atribuída ao lugar.

As inscrições de transição possuem dois tipos de inscrições: nome, que também não possui significado formal, e expressão. E as inscrições de arcos que possuem como inscrição apenas uma expressão. Essa expressão substitui o peso que existia nas redes anteriores. Seu papel é semelhante: indica o conjunto de fichas que devem ser retirados ou adicionados aos lugares adjacentes a esse arco no caso do disparo da transição. Como agora as fichas armazenam valores, torna-se necessário o uso de uma expressão mais complexa.



Fonte: [MURATA, 1989]

A transformação entre as RP e as RPC consiste, fundamentalmente, na substituição de conjuntos de lugares por um só lugar “colorido”, ou seja, contendo marcas coloridas. Essas marcas coloridas permitem a representação de cada um desses lugares através de valores distintos.

4.8 VIABILIDADE ECONÔMICA DE MODELAGEM POR REDE DE PETRI

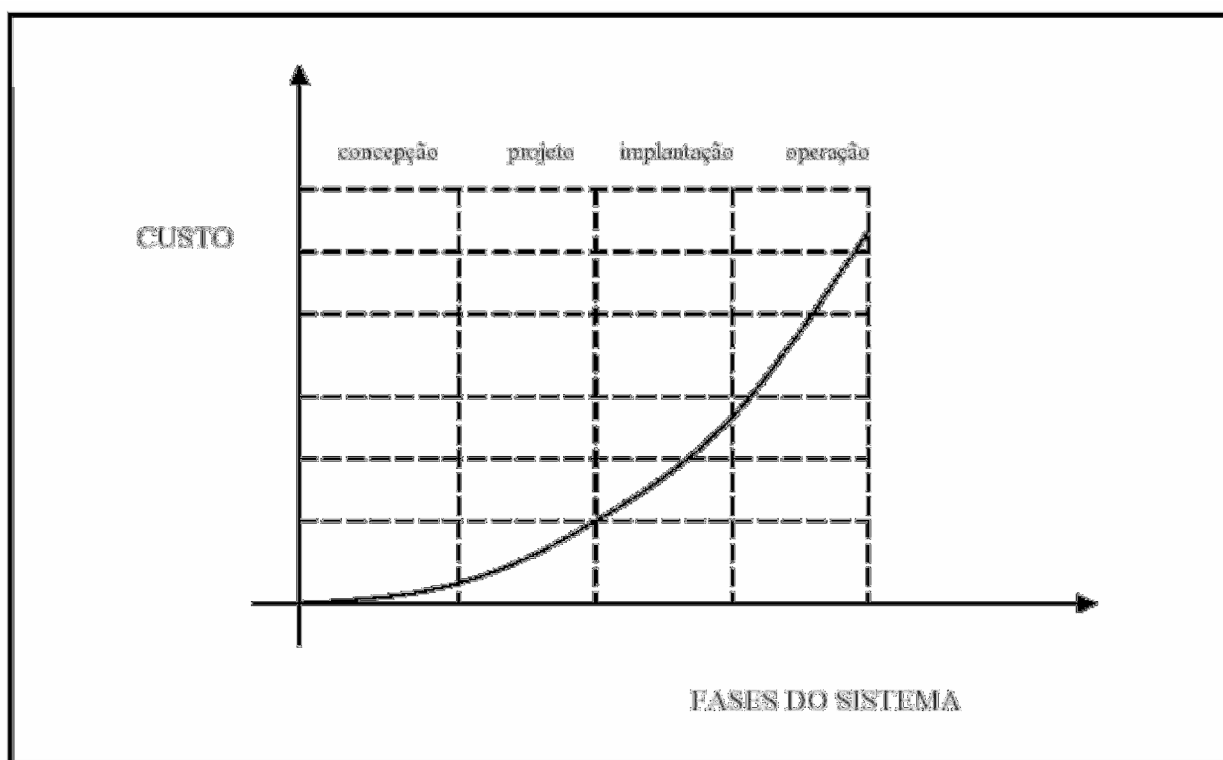
Segundo [KELLNER et al., 1998], o maior motivo para se criar um modelo de simulação ou utilizar qualquer outro método de modelagem é que esta é a forma de menor custo para se obter importantes resultados, quando os custos, os riscos ou a logística de

manipulação do sistema real é proibitiva. Os principais ganhos com a simulação são obtidos com a identificação e eliminação de problemas e ineficiências, chegando até a verificação da inviabilidade do projeto, antes da implantação.

Uma das dificuldades da justificativa econômica da simulação e modelagem é o fato de geralmente não se conseguir avaliar qual será a economia total gerada no projeto, conseguindo-se este valor somente após a sua conclusão.

Durante o seu desenvolvimento um projeto ou sistema passa por diferentes fases com características distintas entre si, ou seja, cada qual com o seu tempo e custo. Quando a simulação é aplicada no desenvolvimento do projeto ou sistema, a relação entre o custo e as fases do projeto é alterada (Figura 4.12). A fase de concepção do projeto ou sistema não sofre nenhum impacto, pois a aplicação da simulação é realizada após esta etapa.

Figura 4.12 - Evolução do custo de um sistema em suas diferentes fases

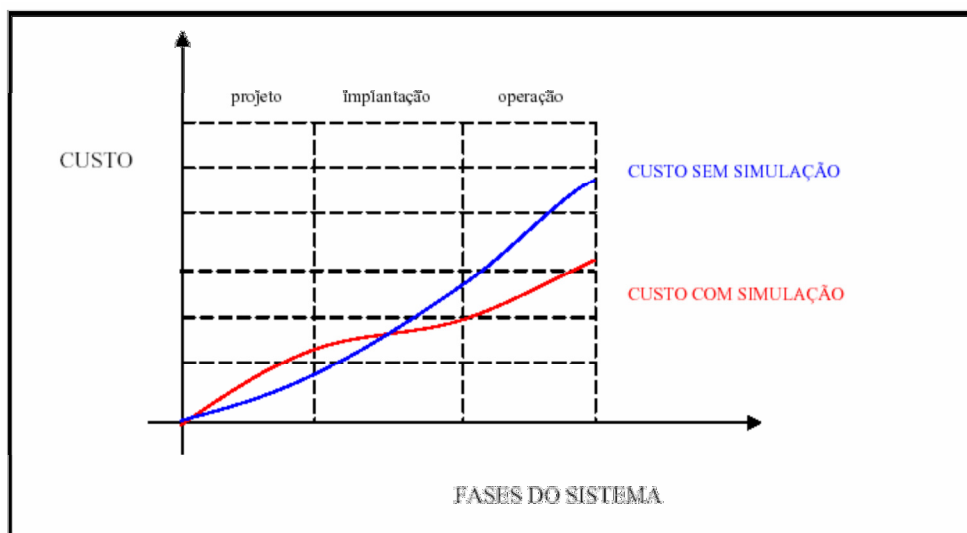


Fonte: [HARREL, 2000]

Em um sistema onde é aplicada a simulação, o custo na fase de projeto fica mais elevado, mas é justamente nesta fase, com o auxílio da simulação, onde se consegue

planejar um sistema mais otimizado, conseqüentemente nas fases de implantação e operação se obtém a esperada redução do custo [BATEMAN et al., 1997] (Figura 4.13).

Figura 4.13 - Comparativo da evolução dos custos com e sem aplicação da simulação

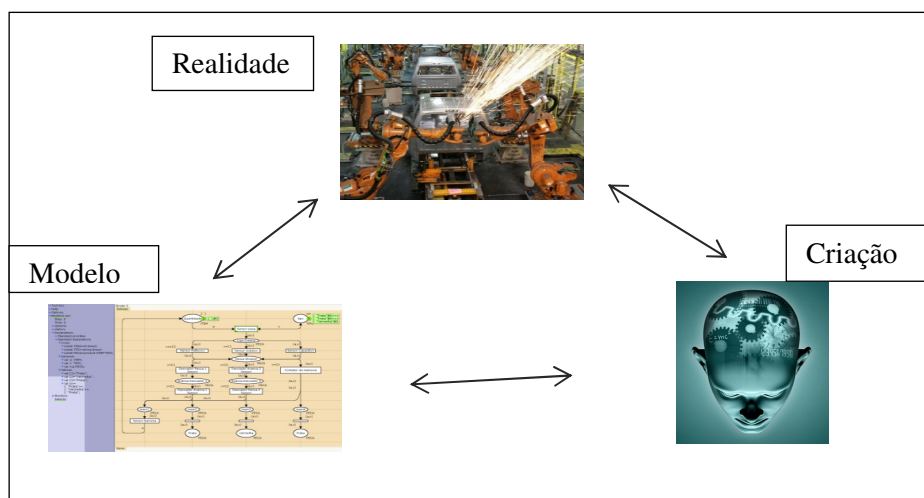


Fonte: [HARREL, 2000]

Ainda na Figura 4.13, pode-se observar que a área sobre a curva dos custos com a simulação é menor que a dos custos sem a simulação, ou seja, o custo total no final é menor quando a simulação é utilizada.

Segundo [HARREL, 2000] e [LAW & KELTON, 1982], simulação é a imitação de um sistema real modelado em computador para avaliação e melhoria do seu desempenho. Ainda, segundo (BANKS, 2000), a simulação e modelagem envolve a criação de uma história artificial da realidade e com base nela são realizadas observações e inferências nas características de operação do sistema real representado, a Figura 4.14 esquematiza este conceito da transformação da realidade em modelo e novamente dos resultados em realidade.

Figura 4.14 - Esquema do conceito de transformação da realidade x resultado em realidade



Fonte: [KUMAR; PHROMMATHED, 2005]

Segundo [KUMAR & PHROMMATHED, 2005], antes da implantação direta de qualquer mudança no sistema produtivo, o mapeamento do processo, a análise dos dados e a simulação computacional podem reduzir com sucesso o risco da ineficácia de uma operação de reprojeto no mundo real.

4.9 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os conceitos básicos para a compreensão da estrutura e funcionamento de uma rede de Petri, como classificação, propriedades, modelo matemático, uma abordagem na rede de Petri Colorida e a demonstração da viabilidade econômica da rede de Petri, objeto central da modelagem proposta nesta pesquisa.

5 MODELAGEM DO SED UTILIZANDO REDE DE PETRI - MODELO PROPOSTO

Neste capítulo serão descritas as etapas da modelagem, apresentado o Sistema de Distribuição de Peças que será modelado, a ferramenta utilizada e o desenho da modelagem propostas nesse estudo.

5.1 APRESENTAÇÃO DO SISTEMA DE DISTRIBUIÇÃO DE PEÇAS

O modelo utilizado para simulação, modelagem e validação foi um equipamento didático da empresa FESTO denominado de MPS (Sistema de Produção e Manufatura), este sistema permite o desenvolvimento de competências nas áreas de automação e integração da produção, considerando o desenvolvimento e a concepção do sistema, a fim de permitir que, embora em ambiente didático, o possa ser desenvolvido experimento o mais próximo possível da realidade dos sistemas de produção encontrados na indústria.

É importante salientar que só os sistemas que utilizam componentes industriais permitem reproduzir condições reais de funcionamento, abordando, além da funcionalidade, aspectos de precisão e repetibilidade.

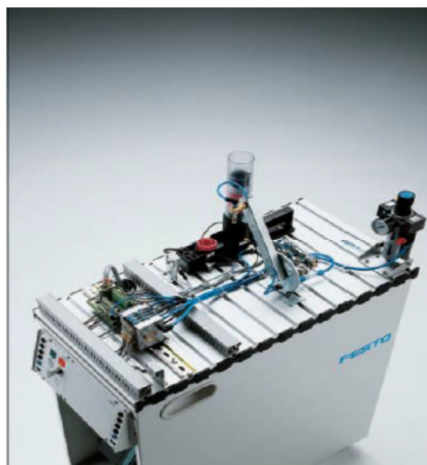
É observado na concepção um dos requisitos mais importantes deste sistema de treinamento para automação industrial está baseado, da forma mais ampla possível, na utilização real de componentes industriais, conceitos de manufatura, softwares, arquiteturas de redes de comunicação industrial e controle de sistemas, bem como integrar tecnologias de automação fundamentais, como por exemplo: Pneumática, Controladores Lógicos Programáveis, Robótica, Manipulação, Logística, Produção e Controle de Qualidade.

Assim, através da integração de diferentes sistemas de manufatura dentro de um processo de produção, permite a abordagem de variadas tecnologias de automação, de produção, de comunicação e de controle, possibilitando a cobertura de um número de habilidades técnicas usuais no ambiente industrial real, inclusive nas áreas de projeto, desenvolvimento de programas, manutenção, dentre outras.

Os padrões utilizados nos sistemas mecânicos, elétricos e de comunicação (protocolos, faixas de tensão, etc.) são os já consagrados e compatíveis com os disponíveis no mercado real industrial.

A Estação de Distribuição de Peças (Figura 5.1) pode ser utilizada como unidade “stand alone” para um treinamento completo, ou em conjunto de toda a instalação. A interligação com outras Estações/Instalações é feita através de interfaces mecânica, elétrica, sem fio (sinais ópticos) e/ou rede de comunicação em protocolo industrial.

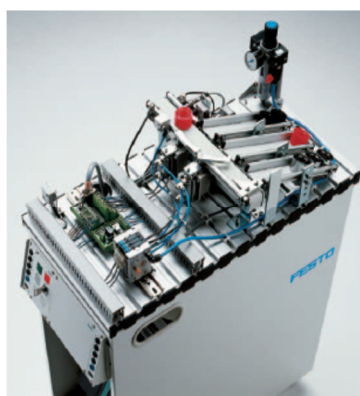
Figura 5.1 - Estação de Distribuição de Peças



Fonte: [EBEL, 2006]

A Estação de Separação de Peças (Figura 5.2) inclui: módulo de esteira; sensores ópticos; sensor indutivo; desviadores ativos; módulo de parada e liberação; rampas de armazenamento; relé; terminal de válvulas.

Figura 5.2 - Estação de Separação de Peças



Fonte: [EBEL, 2006]

Essa MPS possui uma esteira com guias laterais e roletes para deslizamento da cinta transportadora, possui ainda, um sistema de ajuste de tensão da cinta transportadora, como também, sensores ópticos sendo por reflexão difusa e por retro reflexão utilizados

para detecção de material em movimento na esteira. Todos os sensores têm com cabo de fibra óptica para condução da luz.

O sensor indutivo é utilizado para detecção de peças metálicas. Existem dois desviadores ativos que são atuados pneumaticamente com sensores magnéticos de fim de curso, para seleção do fluxo de material na estação. Tem-se ainda nessa estação um módulo de parada e liberação de peças, atuado pneumaticamente e três rampas para armazenamento de peças.

Os motores elétricos das esteiras são comandados por um relé com acionamento manual e possui ainda terminal de válvulas compacto composto de três válvulas.

5.2 ETAPAS DESENVOLVIDAS NA EXECUÇÃO DA MODELAGEM

A estratégia de desenvolvimento, modelagem e implementação da automação de áreas produtivas é interesse, pois considera desde a análise de viabilidade econômica até a especificação de hardware e maneiras de acompanhamento, a verificação e validação dos sistemas. Aqui na presente pesquisa, foi elaborada uma estratégia de modelagem de SED com rede de Petri Colorida que é composta de 04 etapas, levantamento, análise, modelagem e a implementação, descrita na Figura 5.3.

No presente estudo tem-se a modelagem das plantas, a primeira trata-se de uma planta de distribuição, é uma planta constituída de um módulo de distribuição de peças, denominado magazine de peças, onde existe um mecanismo de transporte, denominado girador, cuja função é transportar a peça do magazine para a próxima planta denominada MPS.

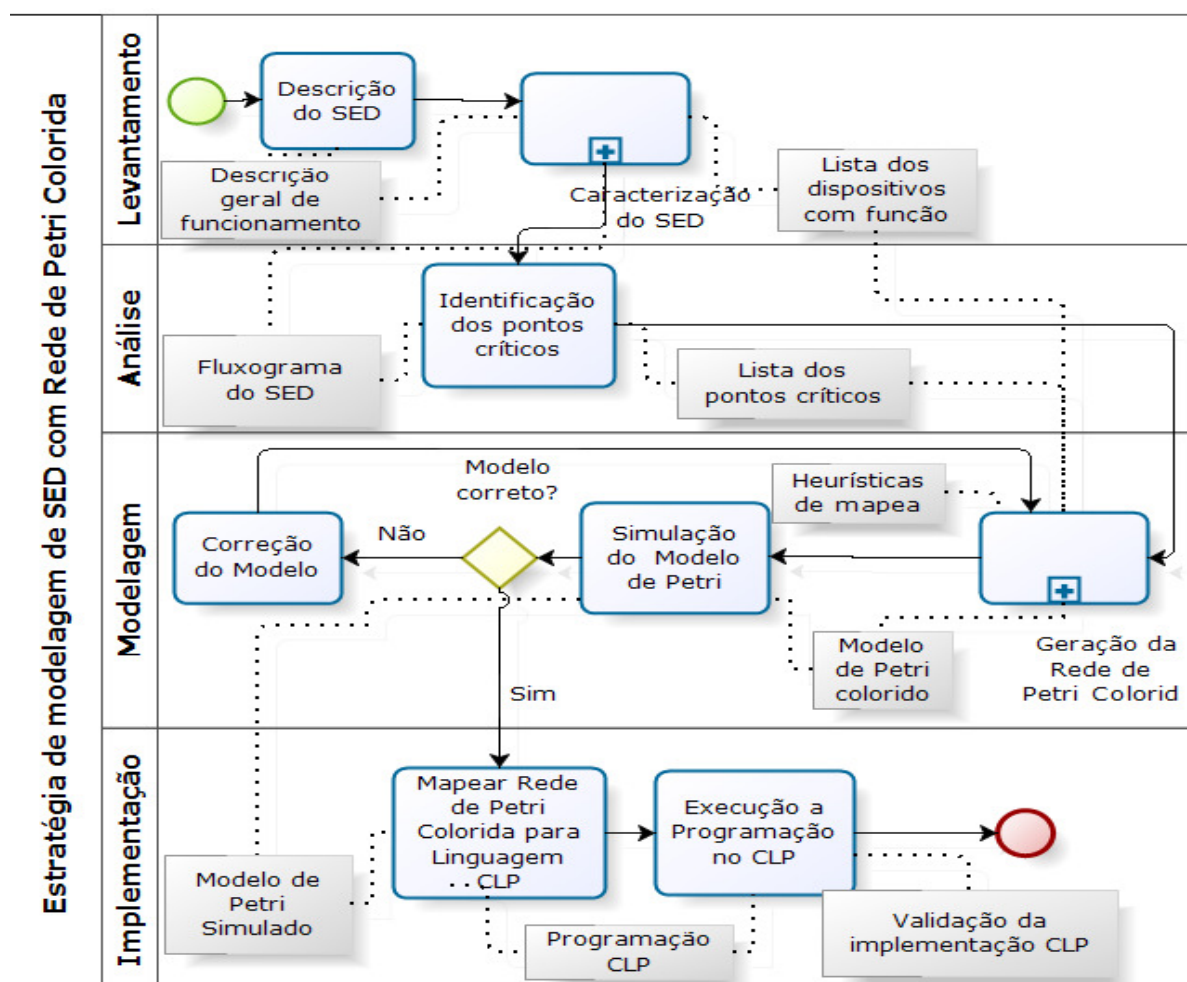
A segunda planta é a de separação, cuja função é separar as peças de acordo com cada cor, uma vez que o processo é composto de peças das cores preto, prata e vermelha. As peças eram processadas da seguinte maneira, inicialmente quando a peça era detectada pelo sensor, ligava-se uma esteira transportadora onde a mesma era verificada, qual seria a sua cor, e em seguida a peça era descartada no local determinado para sua cor específica.

Iniciando a modelagem, na primeira etapa, ocorre o levantamento, onde se tem uma atividade de descrição do SED, onde são analisadas as especificidades do sistema, suas funções, seus objetivos, o resultado apresentado é a descrição geral do funcionamento do

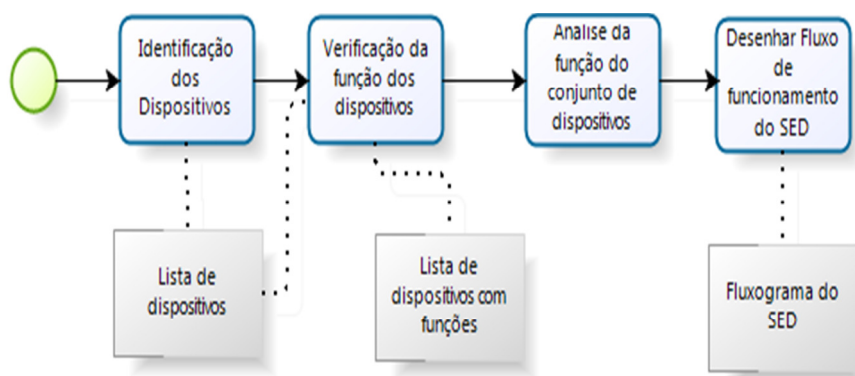
sistema que vai servir de entrada para um macro processo chamado de caracterização do SED.

Na caracterização do SED tem-se um sub-processo descrita na Figura 5.4, onde é feita a identificação dos dispositivos, ou seja, realiza-se a identificação de todos os dispositivos que compõem o sistema de manufatura, originando uma lista com os dispositivos, neste caso foram identificados os seguintes dispositivos: magazine de peças, atuador do magazine, sensor de peça, sensor do avanço do magazine, girador do magazine, sensor de giro posição MPS, sensor de recuo do magazine, sensor de peça do magazine, sensor do girador posição magazine, válvula de vácuo, sensor de vácuo e purgador.

Figura 5.3 - Estratégia de modelagem de SED com Rede Petri Colorida



Fonte: [próprio Autor]

Figura 5.4 - Caracterização do SED

Fonte: [próprio Autor]

Com a lista gerada são verificadas as funções de cada dispositivo, fazendo uma associação entre a função do dispositivo identificado no sistema, como exposto na Tabela 5.1.

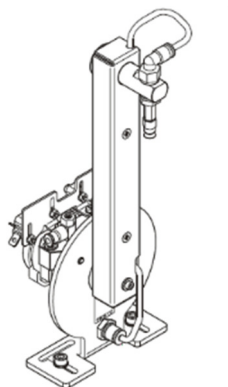
Tabela 5. 1 - Lista de dispositivos com funções

DISPOSITIVO	FUNÇÃO NO SISTEMA
Magazine de Peça	Abastecer o sistema com peças, através de um atuador.
Atuador do Magazine	Movimentar a peça no magazine
Sensor de peça	Identificar a existência de peça no magazine.
Sensor do Avanço do Magazine	Verificar se o atuador foi avançado.
Girador do Magazine	Transportar a peça do magazine para o MPS seguinte do sistema.
Sensor de Giro posição MPS	Verificar se o girador do magazine encontra-se na posição MPS.
Sensor de Recuo do Magazine	Verificar se o atuador do magazine foi recuado.
Sensor de Peça do Magazine	Verificar se a peça está na posição para ser transportada.
Sensor do Girador Posição Magazine	Verificar se o girador do magazine encontra-se na posição magazine.
Válvula de Vácuo	Fixar a peça ao girador
Sensor de Vácuo	Verificar se o se o vácuo foi acionado.
Purgador	Purgar a peça do girador para próxima MPS.

Fonte: [próprio autor]

A próxima etapa desenvolvida é a análise da função do conjunto de dispositivos. No caso da modelagem da estação distribuição de peças (Figura 5.1), o conjunto magazine é responsável por entregar a peça para ser transportada, contando com os dispositivos como sensores e atuadores, para essa ação. Tem-se ainda, o girador magazine (Figura 5.5) responsável pelo transporte de peças do magazine para a próxima estação.

Figura 5.5 - Girador do Magazine



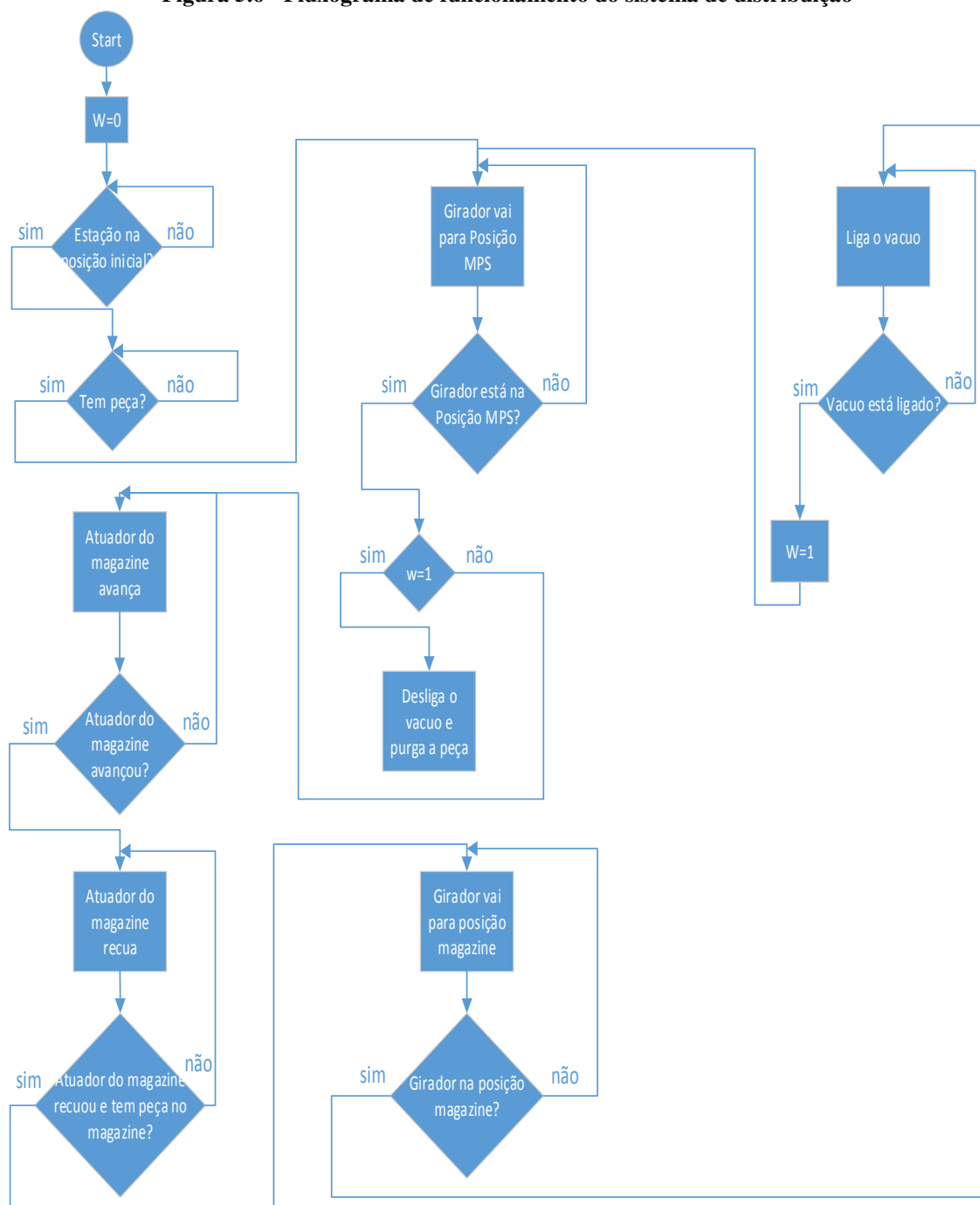
Fonte: [EBEL, 2006]

A atividade seguinte é o desenho do fluxo de funcionamento do SED onde é elaborado o fluxograma de funcionamento do sistema visando atender a funcionalidade do sistema descrito anteriormente, conforme evidenciado na Figura 5.6 que representa o fluxograma da estação de distribuição de peças.

Na segunda etapa, a partir do fluxograma gerado, realiza-se a análise, ou seja, a identificação dos pontos críticos, um estudo preliminar do problema permite a identificação de possíveis pontos críticos da produção, essa identificação pode ocorrer através de fluxogramas de funcionamento do sistema, como por exemplo, intertravamento, que consiste em um bloqueio entre dois elementos do sistema, de forma que esses elementos só entrem em funcionamento de maneira alternada, isto é, quando um elemento estiver em funcionamento o outro deve está impossibilitado de funcionar, pode-se ainda, identificar possíveis colisões entre elementos de atuação, de transporte do sistema, uma vez que foi identificada a função de cada elemento no conjunto e tem-se o fluxo de funcionamento desse sistema.

Uma vez identificados esses pontos críticos por uma análise quantitativa com alto grau de fidelidade à realidade, procede-se uma avaliação de impacto no sistema produtivo de um ponto de vista estratégico e que carrega possíveis soluções apontadas de maneira coletiva, como por exemplo, a substituição de equipamentos que permitam melhorar o desempenho global e não somente de produtividade local.

Figura 5.6 - Fluxograma de funcionamento do sistema de distribuição



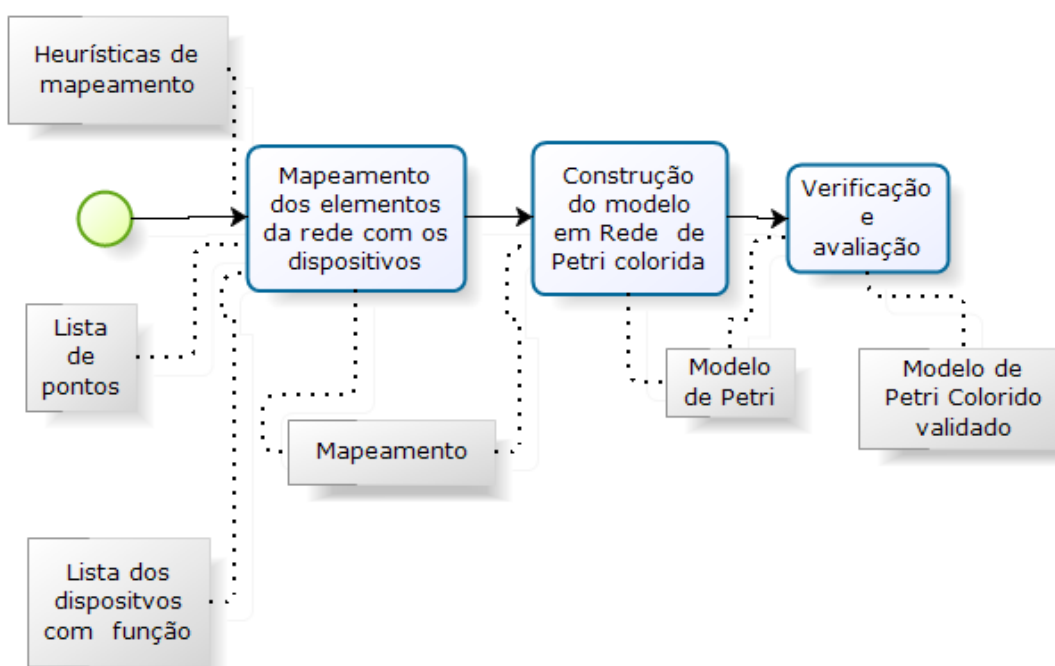
Fonte: [próprio autor]

No dispositivo da Figura 5.5 – Girador do Magazine pode-se identificar a sua aplicação e dentro do fluxograma sendo um ponto crítico, pois, o transporte do produto só depende deste dispositivo gerando uma limitação do girador tanto no sentido MPS e Magazine.

Após a identificação dos pontos críticos, vai-se para a terceira etapa, a modelagem (Figura 5.7). Nesse subprocesso, para a realização do mapeamento dos elementos da rede de Petri com os dispositivos, leva-se em consideração que é necessário identificar as seguintes heurísticas, todos os dispositivos de qualquer sistema identificado como elemento de saída (atuador, lâmpada, válvula), são representados no modelo de Rede de Petri, por um “ESTADO”. Como também, todos os dispositivos de qualquer sistema identificado como elemento de entrada (sensores e botões), são representados no modelo de Rede de Petri, por uma “TRANSIÇÃO”. Os ARCOS serão representados pela matéria-prima do processo em estudo.

Estas heurísticas foram definidas devido à dificuldade de modelar elementos de chão de fábrica, tais como, botão, atuador, sensor, assim, foi necessário estabelecer uma relação entre esses elementos e os elementos da Rede de Petri, onde na revisão da literatura realizada, não se obteve nenhuma demonstração com esses elementos, pois todos os exemplos encontrados, apenas expressava a preocupação da modelagem como um todo, e não com os elementos que compõem o processo, por isso foi necessário mapear os elementos anteriormente exposto, como os elementos de chão de fábrica, estabelecendo as heurísticas acima apresentadas.

Figura 5.7 - Modelagem



Fonte: [próprio autor]

Estabelecida essas heurísticas, verificando a lista de pontos críticos e a lista dos dispositivos com função, executa-se o mapeamento dos elementos da rede com os dispositivos, conforme descritos nas heurísticas, onde na Tabela 5.2, é mapeado os dispositivos fazendo relação com os elementos da rede de Petri. Com isso, é construído o modelo de rede de Petri Colorida e é feita a verificação e validação desse modelo.

A ferramenta de modelagem, análise, simulação de Rede de Petri utilizado neste trabalho foi à CPN Tools, que foi desenvolvida na Universidade de Aarhus, Dinamarca, podendo ser obtida livremente através do endereço <http://cpntools.org/>.

Tabela 5.2 – Planta

Descrição do Elemento do Sistema	Elementos da Rede de Petri Colorida	
	ESTADO	TRANSIÇÃO
Magazine de Peça	Quantidade	-x-
Sensor de peça	-x-	sensor pca
Atuador Magazine Avança	Atuador Mgz avc	-x-
Atuador Magazine Recua	Atuador Mgz rec	-x-
Sensor de Avanço do Magazine	-x-	Sensor_Av_magz
Girador na posição MPS	girador_pos_MPS	-x-
Girador posição Magazine	girador_pos_magz	-x-
Sensor de giro posição MPS	-x-	sensor_gira_pos_MPS
Sensor de recuo do Magazine	-x-	sensor_rec_magz
Sensor de peça do Magazine	-x-	sensor_pca_magz
Sensor girador posição Magazine	-x-	sensor_gira_pos_magz
Liga Vácuo	Liga Vácuo	-x-
Desliga Vácuo	Desliga Vácuo	-x-
Sensor do vácuo	-x-	sensor vácuo
Purgador	Purgador	-x-

Fonte: [próprio autor]

A CPN Tools foi escolhida por suportar várias descrições de modelos de classe da Rede de Petri e por ser considerado de alto nível [JENSEN & KRISTENSEN, 2009].

Ainda por suportar hierarquia e restrição de tempo, muito importante na simulação do modelo real, uma vez que seria necessário validar o modelo em sua totalidade, utilizando diferentes tipos de fichas especificadas por números naturais e representadas visualmente por cores: vermelha, azul, verde, conforme a necessidade.

A simulação do modelo de Rede de Petri foi feita de forma identificar possíveis falhas no modelo, na presente pesquisa foram objeto de estudo as seguintes falhas: alcançabilidade, deadlock e lopp.

Simula-se no intuito de verificar algumas propriedades que definem a Rede de Petri, como alcançabilidade, ou seja, se todos os grafos de marcação são acessíveis. Ainda o deadlock, onde é verificado se a rede atende a essa propriedade, pois este é o principal problema operacional em Sistemas a Eventos Discretos que é a parada total das habilitações subsequentes de um estado, implica na perda da vivacidade de uma forma particular, onde a vivacidade trata-se de uma transição viva, dado um estado inicial X_0 se somente se todas as suas transições são vivas. Para eliminar o deadlock requer usualmente modificar a Rede de Petri por meio de adequados estados e transições adicionais.

Outra verificação é quanto à existência de self-loop, para averiguar se a rede é pura, isto é, (p,t) é tal qual que o estado (p) é o lugar de entrada e um lugar de saída para a mesma transição (t) .

Nos modelos SED é comum encontra-se self-loop para habilitação de determinadas transições nos programas, assim, faz-se necessário a análise para que a Rede de Petri tenha um lopp infinito, fazendo com que o sistema apresente erro e não consiga finalizar o programa.

Realizada a simulação e verifica as propriedades, caso o modelo não esteja correto é feito o ajuste na rede de Petri e mais uma vez realiza-se uma simulação para identificação de possíveis falhas.

Na Figura 5.8 foi gerado o modelo de Rede de Petri, a partir da lista de associação apresentada na tabela 5.2, onde conforme definido estado inicial “quantidade” é o ponto de partida da rede, esse estado representa a quantidade de fichas, onde a ficha representa o produto ou material a ser processado no sistema.

Conforme fluxograma de funcionamento apresentado na Figura 5.6, o sistema inicia a partir da detecção de peças no magazine, essa detecção acontece através de um sensor

por se tratar de um elemento de entrada e conforme e definido na heurística, anteriormente mencionada, ele será representado como uma transição, chamada sensor de peça (sensor_pca).

Essa transição possibilita o disparo para dois estados conforme identificados no fluxo que representa dois dispositivos de saída, ou seja, o atuador do magazine (Atuador Mgz_avc) e o girador posição magazine (girador_pos_MPS), onde esses dispositivos de acordo com a heurística são representados por estados.

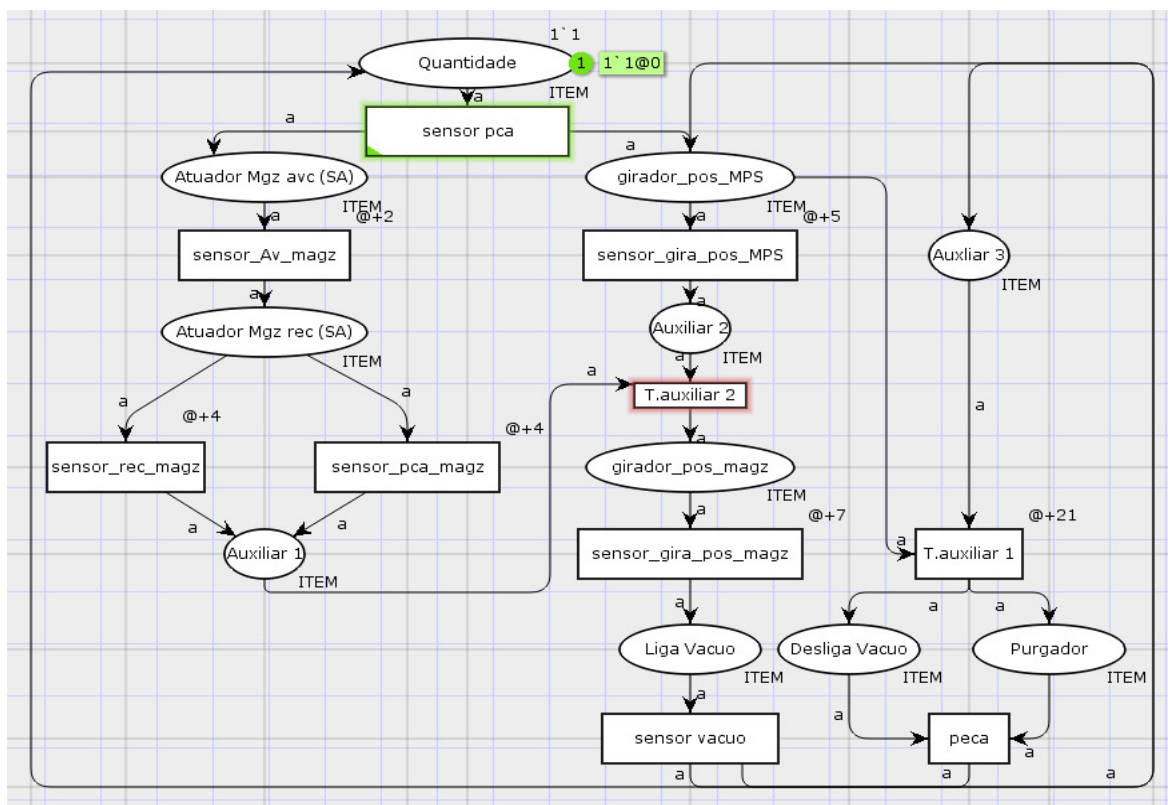
Os dois estados habilitam as transições, sensor de avanço do magazine (sensor_Av_magz) e sensor de giro MPS (sensor_gira_pos_MPS). Nestas transições foram estabelecidos base de tempo diferente, onde o girador posiçãoMPS levaria um tempo maior que o atuador do magazine, estes tempos foram estabelecidos de forma aleatória e posteriormente adequado ao modelo real.

A partir deste momento, as duas transições estão habilitadas para o disparo, como a transição de tempo menor que é o caso de sensor de avanço de magazine (sensor_Av_magz), ele fará seu disparo primeiro, indo para o estado atuador do magazine recua (Atuador Mgz rec) que habilitará duas transições, sensor recua magazine (sensor_rec_magz) e sensor peça magazine (sensor_pca_magz). As duas transições tem tempos iguais definidos, onde nesse momento, o estado girador posição MPS (girador_pos_MPS) que habilita a transição sensor de giro posição MPS (sensor_gira_pos_MPS), a transição dispara indo para o estado auxiliar 2 que aguarda o disparo das duas transições, sensor recua magazine (sensor_rec_magz) e sensor peça magazine (sensor_pca_magz) indo para o estado auxiliar 1, que juntamente com o estado auxiliar 2 habilitará a transição auxiliar 2.

A transição auxiliar 2 dispara indo para o estado girador posição magazine (girador_pos_magz), que habilita a transição sensor giro posição magazine (sensor_gira_pos_magz) essa transição dispara e vai para o estado liga vácuo, onde este estado habilita o disparo da transição sensor vácuo, que vai para os dois estados, auxiliar 3 e girador na posição MPS que habilita a transição auxiliar 1.

Esta transição ao disparar vai para os estados desliga vácuo e purgador, habilitando a transição peça que fecha o ciclo do processo do modelo descrito no fluxograma de funcionamento.

Figura 5.8 - O Modelo



Fonte: [próprio autor]

Por fim, a quarta etapa, a implementação, que tem como atividade mapear a rede de Petri Colorida para a linguagem *LADDER*, gerando programa do CLP onde será descarregado na CPU do Controlador Lógico Programável e executa o programa.

Para a validação da implementação da atividade foram realizados dois experimentos com a participação de 35 programadores de CLP de nível técnico.

Um experimento foi à programação da planta de distribuição e o segundo experimento foi a planta de separação. Distribuído da seguinte maneira: os 35 programadores foram divididos em 02 grupos, o primeiro grupo com 18 técnicos e o segundo com 17 técnicos, conforme descrito na tabela 5.3, onde o primeiro grupo programou a Planta de Distribuição a partir do fluxograma de funcionamento e o segundo grupo programou a partir da Rede de Petri, e para o segunda Planta os grupos foram invertidos onde o primeiro grupo programou desta vez utilizando a Rede de Petri e o segundo grupo utilizando apenas o fluxograma conforme mostra na tabela 5.3.

Tabela 5.3 - Avaliação do experimento

Grupos	Planta de Distribuição		Planta de Separação	
	Fluxograma	Rede de Petri	Fluxograma	Rede de Petri
Grupo 1 (18 técnicos)	Programou	Não Programou	Não Programou	Programou
Grupo 2 (17 técnicos)	Não programou	Programou	Programou	Não programou

Fonte: [próprio autor]

Foi realizado um nivelamento explicando sobre a Rede de Petri e o que representava cada elemento descrito na rede como também o fluxograma e os critérios estabelecidos para programação como a linguagem a ser utilizada que foi a linguagem *LADDER* por ser uma linguagem universal entre os controladores lógicos e que não poderia realizar nenhum teste parcial o total do programa impossibilitando a prática comumente utilizada na indústria para programação dos CLP's de forma de tentativa e erro, uma vez que o software do CLP tinha possibilidade de simular o funcionamento e o teste seria realizado direto nas plantas utilizadas no experimento. Foram estabelecida as heurísticas para conversão dos elementos descritos anteriormente como o que seria considerado entrada, saída fazendo a relação com os elementos da planta como os sensores, atuadores e válvulas, abaixo na figura 5.9 mostra a tabela de conversão que foi explicada para que os experimentos fosse realizado, onde cada arranjo descrito no modelo de Rede de Petri com a relação de como representar na linguagem *LADDER*.

Figura 5.9 - Conversão LADDER x Rede de Petri

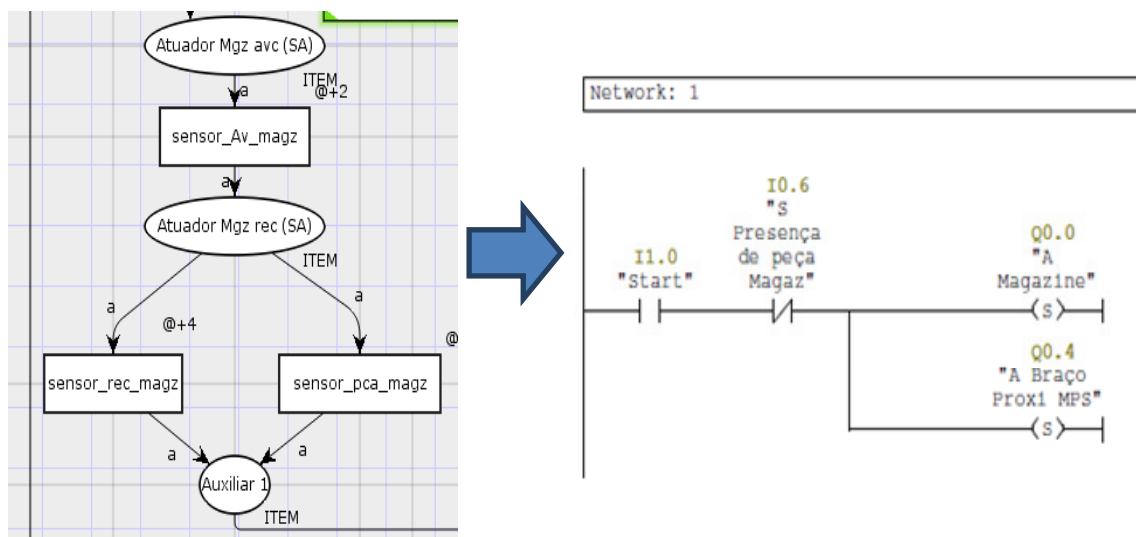
Lógica	Rede de Petri	Linguagem ladder
Lógica AND	<p>Nós = $m + n + 1$ Links = $m + n$</p>	<p>Nós = $m + n$ Links = $m + 2n$</p>
Lógica OR	<p>Nós = $2m + n$ Links = $m(1 + n)$</p>	<p>Nós = $m + n$ Links = $2(m + n)$</p>
Modelo Sequencial	<p>Nós = $2(m + n + n') - 1$ Links = $m + n + 2n'$</p>	<p>Nós = $m + n + 2n'$ Links = $3(n' + 1) + (m - 1) + 2(n - 1)$</p>
Lógica AND Temporizada	<p>Nós = $m + n + 1$ Links = $m + n$</p>	<p>Nós = $m + n + r + 1$ Links = $m + 2n + r + 2$</p>
Lógica OR Temporizada	<p>Nós = $2d + n$ Links = $d(1 + n)$</p>	<p>Nós = $3d + n$ Links = $5d + 2n$</p>
Modelo Sequencial Temporizado	<p>Nós = $2d + m + n - 1$ Links = $2d + m + n - 2$</p>	<p>Nós = $4d + m + n - 2$ Links = $6d + m + 2n - 3$</p>

Fonte: [Dias, 2005]

A partir da Rede de Petri e utilizando a heurísticas e a lógica de conversão para a linguagem *LADDER* conforme representa a figura 5.10, apresenta parte do programa onde podemos identificar alguns elementos como atuador MgZ avc sendo representado como um estado e segunda a heurística estabelecida será representado por saída que é identificada na linguagem *LADDER* como sendo o endereço Q0.0 na figura 5.10, ainda podemos identificar também o elemento sensor_Av_magz sendo representado na Rede de Petri

como uma transição e na linguagem sendo representado como um contato identificado pelo endereço IO.6 da figura.

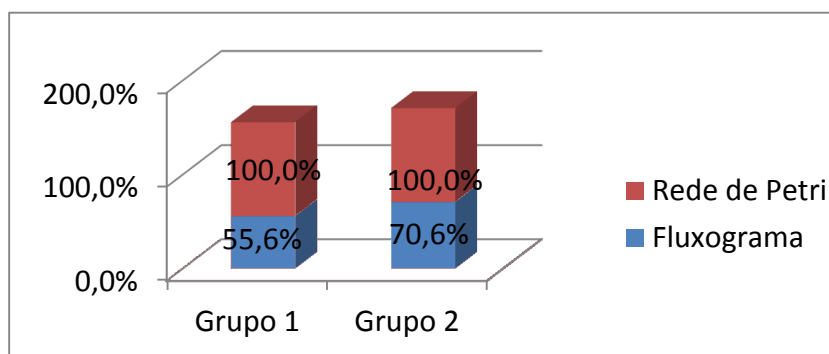
Figura 5.10 - Conversão do Modelo em Rede de Petri em LADDER



Fonte: [próprio autor]

O resultado verificado conforme observado no gráfico da figura que para os dois grupos que realizaram a programação a partir da Rede de Petri obtiveram êxito ao contrário dos que programaram com base no fluxograma de funcionamento que obtiveram resultados de 55,6% e 70,6%, para os grupos um e dois respectivamente, ilustrado na Figura 5.11, comprovando que é possível modelar sistema sem a necessidade de ter o equipamento real e que também prever erros de programas sem a necessidade de testar ou como na prática industrial que se baseia no método de tentativa e erro.

Figura 5.11 - Resultado do experimento



Fonte: [próprio ator]

Diante da revisão da literatura entre vários trabalhos analisados foi feito um comparativo entre a proposta atual deste trabalho e mais dois trabalhos que mais se aproximaram do proposto conforme apresentado na tabela 5.4, onde foram verificadas algumas características como identificação de elementos básicos da linguagem *LADDER* e Rede de Petri, onde em todos os trabalhos foram identificados os elementos descritos nos trabalhos; outra característica foi a caracterização dos dispositivos, nesse tópico os outros autores apenas tinha a preocupação de caracterizar o sistema como um todo, o que na proposta deste trabalho é utilizar a Rede de Petri para elementos de chão de fábrica e por isso da necessidade de caracterização de todos os elementos dos dispositivos; o fluxo de funcionamento, onde a necessidade de conhecer características do funcionamento a qual se deseja modelar não foi identificada, os trabalhos tratam logo de forma genérica na representação da Rede de Petri, sem a necessidade de mostrar como o sistema funciona através de um fluxo; A lógica de conversão, a identificação de arranjos em Rede de Petri sendo representado em uma lógica *LADDER*, definindo assim as principais portas lógicas conforme representado na figura 5.9; o modelo em Rede de Petri e a descrição do funcionamento foi uma característica identificada em toda a literatura consultada.

Tabela 5.4 - Comparativo entre os Trabalhos relacionados

Características	[Dias, 2005]	[Santos, 2008]	Proposta Atual
Identificação dos elementos básicos de Linguagem Ladder e Rede de Petri	X	x	x
Caracterização dos dispositivos			x
Fluxo de funcionamento			x
Lógica de conversão RP e Ladder	X		x
Modelo em Rede de Petri	X	x	x
Descrição do funcionamento	X	x	x

Fonte: [próprio autor]

5.3 CONSIDERAÇÕES FINAIS

Foram abordadas neste capítulo as etapas previstas na execução da modelagem, em seguida foi apresentado o modelo real, as ferramentas utilizadas e a estruturação da modelagem. Discutiu-se ainda, que simulação e a modelagem é uma ferramenta capaz de fornecer resultados para análises mais elaboradas a respeito da dinâmica do sistema, permitindo desta maneira uma interpretação mais profunda e abrangente do sistema estudado.

6 CONSIDERAÇÕES FINAIS E TRABALHO FUTUROS

Percebe-se que com a evolução natural do sistema produtivo e a globalização, as empresas tiveram que se modernizarem, ou seja, modernização nos processos de produção, como também na redução do intervalo de tempo entre o lançamento de novos produtos.

Nesse cenário de alta competitividade do mundo globalizado, cada vez mais são solicitados sistemas de produção de baixo custo, melhor qualidade, alta flexibilidade e confiabilidade ao longo da cadeia produtiva, de forma manter a lucratividade necessária para auto sustentação dos negócios. Assim, ocorre uma migração dos sistemas de produção, dos tradicionais baseados em produção de lote para uma arquitetura em que se tem maior flexibilidade, caracterizando os sistemas de manufatura [GROOVER, 2011], também chamados de Sistemas Flexíveis de Manufatura (SFMs), que conseguem produzir uma variedade de produtos modificando a sua configuração de acordo com o planejamento da produção, conforme foi visto ao longo desse estudo, assim faz-se expandir um novo campo de conhecimento, o da teoria de sistemas a eventos discretos (SEDs).

No SEDs o sistema dinâmico, é visto como aquele que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos de tempo em geral irregulares e desconhecidos, e são também, sistemas acionados por eventos [MORAES, 2010].

Para tal pode-se dispor de metodologias sistemáticas para atingir o desempenho desejável e evitar falhas, então a proposta deste trabalho foi justamente prover um método de modelagem, onde foram considerados as ocorrências de falhas e que possibilitou resolver o problema da incerteza sobre a real capacidade de produção do sistema, e a partir daí, foi feita a análise da desempenho dos mesmos sob diversas situações que ocorrem no dia a dia no gerenciamento dos Sistemas de Manufatura.

Evidencia-se também, com essa pesquisa que qualquer projeto de sistemas dinâmicos a eventos discretos (como os sistemas automatizados de manufatura) podem ser modelados. Isso pode ocorrer para que se consiga determinar as melhores condições de operação e detectar erros de projetos antes que o sistema seja implantado. Erros significam custos que podem inviabilizar todo um projeto, conforme visto anteriormente.

Sabe-se ainda que hoje a programação dos controladores lógicos em um sistema de manufatura é realizada de forma intuitiva, onde critérios não são observados, possibilitando a ocorrência de possíveis erros de programação que só são identificados nos testes realizados no sistema real.

Assim, retomando a problemática deste estudo quanto à fragilidade das análises realizadas para uma validação técnica e econômica do projeto de manufatura, reitera-se que a modelagem de um sistema de manufatura de produção a partir do modelo de Rede de Petri Colorida, analisa as ocorrências de falhas na programação dos controladores lógicos, deixando de ser de forma intuitiva, identificando os erros de programação previamente, acabando assim com a fragilidade anteriormente realizada, que não conseguem exprimir a total confiabilidade do funcionamento do equipamento, como também se ganha no tempo das respostas dos diferentes cenários propostos.

Esse modelo de Rede Petri Colorida proporcionou um sistema previamente validado, com os experimentos realizados nesta pesquisa, que permitiu uma maior segurança no desenvolvimento dos programas previstos para os controladores programáveis.

Com a modelagem do sistema de manufatura de produção a partir do modelo de Rede de Petri Colorida, evitou-se a ocorrências de falhas na programação dos controladores lógicos, essa verificação ocorreu através de uma simulação de um sistema real, conforme descrito anteriormente.

Fatores importantes fortalecem as vantagens da utilização da Rede de Petri representadas pelas seguintes considerações e que devem ser considerados:

- A RP tem sido usada com sucesso para modelar, controlar e analisar sistemas a eventos discretos, que são caracterizados pela concorrência dos processos, pelo paralelismo, por serem assíncronos, por possuírem *deadlocks*, conflitos e processos de eventos direcionados;
- A RP permite a implementação de análise em tempo real.
- A RP constitui-se como uma teoria bem fundamentada para a verificação de propriedades qualitativas.

Sob esta ótica, reitero a relevância da modelagem e avaliação desses sistemas sendo possível responder assertivamente, antecipadamente e quantitativamente, sobre o desempenho a ser obtida pelo sistema, levando-se em consideração que a quantificação da interrupção não prevista e seus efeitos podem ser mensurados, o que possibilita a tomada de decisão e a execução de ações sobre o sistema.

Esta pesquisa investigou os resultados da aplicação da simulação e modelagem do projeto de uma nova linha de manufatura.

Pode-se ainda ressaltar nesta pesquisa a introdução do problema da avaliação da desempenho dos sistemas modernos de manufatura, que através da Rede de Petri Colorida possibilitou-se a modelagem, a análise e diagnose de falha para os SFMs.

Face ao exposto e com a investigação realizada, o método de modelagem e avaliação de sistemas de manufatura flexíveis, é de suma importância para o aumento da produtividade essencial para o desenvolvimento industrial devido aos benefícios alcançados com esse método que permite a análise da desempenho dos sistemas sob diversos pontos de vista ou situações que ocorrem no dia a dia no gerenciamento dos Sistemas de Manufatura.

Assim, através das simulações do sistema de manufatura, foram identificados possíveis erros de projetos corrigindo-os antes do término do programa. A utilização detalhada do modelo proposto facilitou a programação do Controlador Lógico Programável, que através dos experimentos validaram o modelo proposto.

Reitero que os experimentos realizados confirmaram a utilidade e eficiência das redes de Petri Colorida na modelagem e análise dos SFMs. Confirmando-se assim, a possibilidade de se utilizar a metodologia desenvolvida nesse estudo para auxílio a projetos (modelagem, análise e especificação) em sistemas flexíveis de manufatura ao permitir a efetiva compreensão das funcionalidades do sistema e explicitando as interações entre os diferentes dispositivos envolvidos.

Evidenciou-se ainda a combinação entre modelagem e simulação apoiando a tomada de decisão, trazendo vantagens significativas na fase de planejamento do projeto e na execução de uma linha de manufatura em alternativa aos métodos usualmente empregados na indústria.

6.1 TRABALHOS FUTUROS

Diante do que foi abordado no decorrer deste trabalho que aborda a fragilidade e forma de programar dos controladores lógicos programáveis nos processos industriais, gerou-se a inquietação em relação a forma como é elaborado os programas nos CLP, assim a pesquisa teve esse propósito de solucionar os problemas relacionados, descrevendo uma metodologia através de um modelo representado através de uma ferramenta que foi

definida pela Rede de Petri Colorida, a partir da complexidade exigida do sistema analisado, onde o modelo proposto conseguiu atingir os objetivos propostos neste trabalho.

No início a pretensão era de desenvolver uma interface capaz de interpretar a Rede de Petri e descarregar o programa diretamente na memória do Controlador lógico Programável, por conta do tempo exigido e dos conhecimentos relacionados com os protocolos de comunicação e o domínio com linguagem de programação não foi possível a sua implementação, assim fica como possibilidade para trabalhos posteriores.

Como outros trabalhos futuros, também, pode-se evidenciar a continuidade deste tema, pois como foi visto a Rede de Petri é uma ferramenta poderosa podendo ser utilizada em varias áreas e processos distintos, destaca-se a modelagem de processo contínuos, utilizando Rede de Petri estocástica, a transformação de modelos a partir da Rede de Petri, até mesmo a colaboração no melhoramento da ferramenta CPNtool no processo de modelagem e transformação de modelos, como também o desenvolvimento de modelos matemáticos computacionais para caracterização de sistemas de automação utilizando redes de Petri.

7 ANEXOS

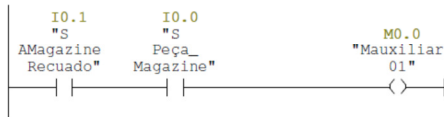
Figura 7.1 - Exemplo de Programa a partir do modelo de Rede de Petri

SIMATIC MPS DP FESTO PETRI 08/23/2013 02:00:42 PM
GAB\01Distribuing\CPU 313C-2 DP\...\OB1 - <offline>

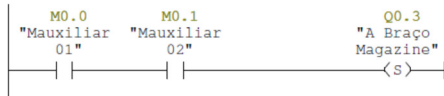
Network: 3



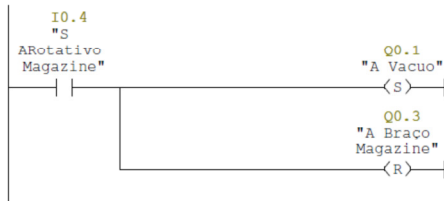
Network: 4



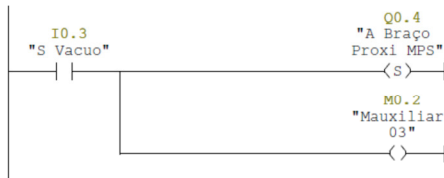
Network: 5



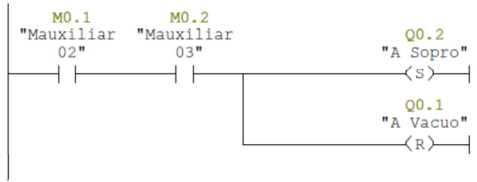
Network: 6



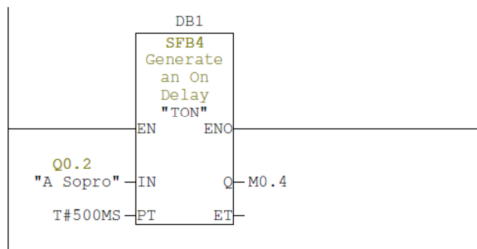
Network: 7



Network: 8



Network: 9



Network: 10



Segundo exemplo, do programa também utilizando o modelo de Rede de Petri.

Figura 7.2 - Exemplo de Programa a partir do modelo de Rede de Petri

SIMATIC MPS DP FESTO PETRI OZIEL\ 08/15/2013 04:06:46 PM
01Distribuing\CPU 313C-2 DP\...\FC1 - <offline>

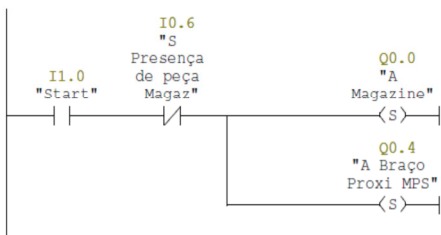
FC1 - <offline>

"Distribuing" Programa Baseado na Rede de Petri
Name: Family:
Author: Version: 0.1
 Block version: 2
Time stamp Code: 08/15/2013 02:45:57 PM
 Interface: 08/14/2013 01:08:25 PM
Lengths (block/logic/data): 00248 00138 00006

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1 MPS Distribuing Baseada na Rede de Petri Ozziel Alves

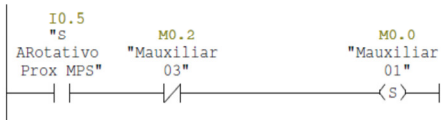
Network: 1



Network: 2



Network: 3

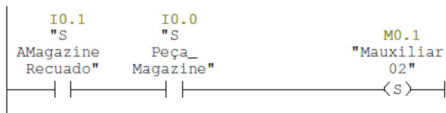


SIMATIC

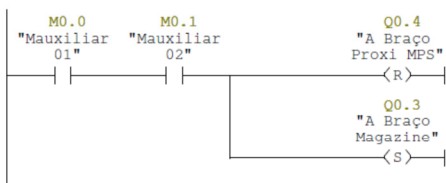
MPS DP FESTO PETRI OZIEL\
01Distribuing\CPU 313C-2 DP\...\FC1 - <offline>

08/15/2013 04:06:46 PM

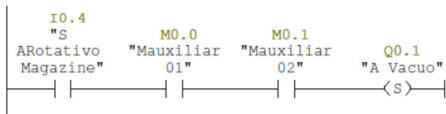
Network: 4



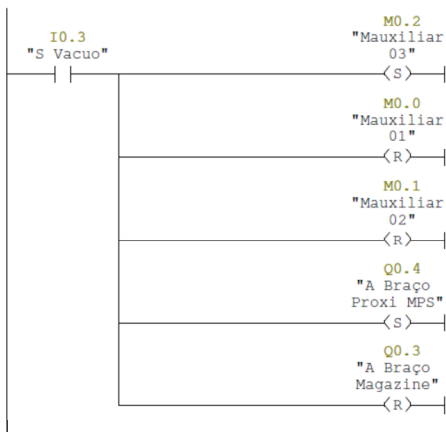
Network: 5



Network: 6



Network: 7

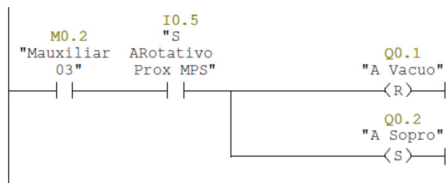


SIMATIC

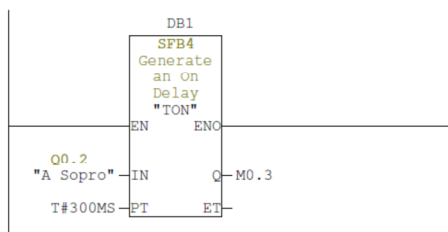
MPS DP FESTO PETRI OZIEL\
01Distribuing\CPU 313C-2 DP\...\FC1 - <offline>

08/15/2013 04:06:46 PM

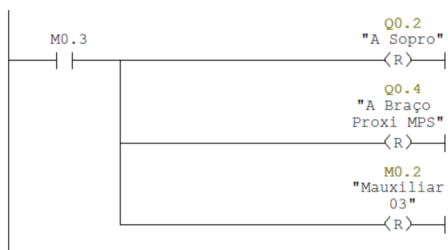
Network: 8



Network: 9



Network: 10



REFERÊNCIAS

- ARATA, W. **Representação Computacional de Sistemas a Eventos Discretos Considerando a Heterogeneidade e a Integração dos Modelos**. São Paulo, 2005. 188f. Tese de Doutorado – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.
- ARAÚJO, J.J., BECKER, L.B., PEREIRA, C.E., **Interface de Comunicação entre Ambiente de Modelagem Orientado a Objetos e Sistemas Supervisórios**, SBAI, 2001.
- ATTIÉ, S.S., **Automação Hidráulica e Pneumática Empregando a Teoria de Sistemas a Eventos Discretos**. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1998.
- BANKS, J. **Introduction to Simulation**. *Proceedings of the Winter Simulation Conference* Atlanta, 2000.
- BATEMAN, R. E.; BOWDEN, R. O. ; GOGG, T. J.; HARRELL, C. R.; MOTT, J. R. A.BOLCH, G. GREINER, S. MEER, H.; TRIVEDI, K, S. **Queueing Networks and Markov Chains. Modeling and Performace Evaluation with Computer Science Applications**. 2 ed., New Jersey; John Wiley & Sons, Inc., 2006.
- CARDOSO, J., VALETTE, R. **Redes de Petri**, Florianópolis: Editora da UFSC, 220pp., 1997.
- CASSANDRAS, C. and LAFORTUNE. **Introduction to Discrete Event Dynamic Systems**, Springer, New York, N.Y., USA, 2008.
- CHAMBERS, C., MIKE H., and JUDITH B. **Introduction X-Machine models to verify PLC Ladder Diagrams**, Computer in Industry 45 93), pp. 277-290, 2001
- CURY, J.E.R., **Teoria de Controle Supervisório de Sistemas a Eventos Discretos**. V Simpósio Brasileiro de Automação Inteligente. Canelas-RS, 2001.
- DAVID R. and ALLA, H. **Discrete, Continous, and Hybrid Petri Nets**. Springer, Berlin-Heidelberg, Germany. 2005.
- DE MORAES, CÍCERO C. and PLINIO L. CASTRUCCI, **Engenharia de Automação Industrial**, Rio de Janeiro p RJ. LTC Editora. 2001.
- DESROCHERS, A.A., AL-JAAR, R. Y., **Applications of Petri Nets in Manufacturing Systems: Modeling, Control and Performance Analysis**. IEEE Press, 1995.
- DIAS, J.R.S, **Um Laboratório para um Curso de Automação Industrial utilizando a Teoria de Sistemas a Eventos Discretos**. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, Brasil, 2005.
- DICESARE, F., HARHALAKIS, G., Proth, J.M., Silva, M.,Vernadat, F. B., **Practice in Petri Nets**, *Proceedings of Sixth International Workshop on Discrete Event Systems*, IEEE, 2002.
- DU, Y., C. JIANG e MC ZHOU, **Um modelo baseado em redes de Petri para a verificação das obrigações e responsabilidade na Cooperativa de Sistemas**, IEEE Trans. em Sistemas, cara, e Cibernética: Parte A, vol. 39, N °, 2, pp 299-308, mar. 2009.

- EBEL, FRANK., PANY, MARKUS, **Manual MPS - Festo**, Festo Didactic GmbH & Co. KG. D-73770 Denkendorf, 2006.
- FANTI, M.P. **Deadlock Resolution Strategy for Automated Manufacturing Systems Including Conjunctive Resource Service**. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, n.01, pp. 80-92. 2004.
- FRANCÊS, C.R.L., **Introdução as Redes de Petri**. Laboratório de Computação Aplicada – LACA. Universidade Federal do Pará – UFPA, 2003.
- GIL, A. C. **Métodos e Técnicas de Pesquisa Social**. 6.ed. São Paulo: Atlas, 2009.
- GIUA, A.S. e BASILE, F., **Observer-Based State-Feedback Control of Timed Petri Nets with Deadlock Recovery**. *IEEE Transactions on Automatic Control*, pp.17-29, vol. 29, ISSN 0018-9286, 2004.
- GROOVER, MIKELL. **Automação Industrial e Sistemas de Manufatura**. Tradução Jorge Ritter, Luciana do Amaral Teixeira, Marcos Vieira. Revisão técnica José Hamilton Chaves Gorgulho Júnior. 3. ed. – São Paulo: Pearson Prentice Hall, 2011.
- HAMILTON, KIM and RUSSEL MILES, **Learning, UML, 2.0**, O'Reilly. 2006.
- HARREL, CHARLES R.; GHOSH, BIMAN K.; BOWDEN, ROYCE. **Simulation Using Simulation Using ProModel®**. McGraw-Hill, 2000.
- HASEGAWA, K, **Modeling, Control and Deadlock Avoidance of Flexible Manufacturing Systems**, In: Conferências Plenárias of XI CBA. São Paulo, SBA. P37-51. 1996.
- HEUSER, C.A. **Modelagem Conceitual de Sistemas: Redes de Petri**. Material de apoio utilizado no curso de pós-graduação em Ciências da Computação da UFRGS., 1990.
- JENSEN K., and KRISTENSEN L.M., **Coloured Petri Nets. Modelling and Validation of Concurrent Systems**. 1 ed. Springer.2009.
- JENSEN K., KRISTENSEN L. M., and WELLS L. **Coloured Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems**. *International Journal on Software Tools for Technology Transfer*. Volume 9. Issue 3-4, pp. 213-254, .2007.
- KESTEN, Y., PNUELI, A. and RAVIV, L. **Algorithmic Verification of Linear Temporal Logic Specifications**. *Automata, Languages and Programming*, pp. 1-16, 1998.
- KUMAR, S. and PHROMMATHED, P. **Improving a Manufacturing Process by Mapping and Simulation of Critical Operations**, 2005.
- KUO, C. and HUANG, H., **Failure Modeling and Process Monitoring for Flexible Manufacturing Systems Using Colored Timed Petri Nets**, *IEEE Transactions on Robotics and Automation*, Vol. 16, n.3, 2000.
- LAW, A. M. and KELTON, W. D. **Simulation Modeling and Analysis**. New York: Mc Graw-Hill, 1982.
- LEE, JS, MC ZHOU e PL HSU, **Multi-paradigma de Modelagem para Sistemas Dinâmicos Híbridos Usando Petri Net Framework**, *IEEE Trans on Systems, Man, e Cibernética*, n ° 2, pp 493-498, mar. 2008.

- LI, Z., S. ZHU e MC ZHOU, **Aplicação de um Dividir e Conquistar Estratégia para Prevenção de Deadlock em Sistemas Flexíveis de Manufatura Baseado em Redes de Petri**, *IEEE Transactions on Systems, Man and Cybernetics*, Parte C, jan. 2009.
- LI, ZW, MC ZHOU, e NQ WU, **A Avaliação e Comparação da Base de Rede de Petri Políticas Prevenção de Conflito de Sistemas Flexíveis de Manufatura**, *IEEE Trans. em Sistemas, cara, e Cibernética: Parte C*, vol. 38, No. 2, pp 173-188, mar. 2008.
- LIMA, E. A. e OLIVEIRA, P. J. **Projeto de Análise e Controle de Supervisório de Sistemas a Eventos Discretos Usando Redes de Petri Temporais**. Technical report. Centro de Pesquisa e Tecnologia da Universidade do Estado da Bahia, Salvador. 2007.
- MACIEL, PAULO R.M., LINS, RAFAEL D. e CUNHA, PAULO R.F. **Introdução às Redes de Petri e Aplicações**. Campinas, Instituto de Computação da UNICAMP, 1996.
- MENESES. F. A. A., BARROSO G.C e PRATA, B. **Restrições de Controle sobre Cores Decompostas: uma proposta no controle supervisório de sistemas a eventos discretos utilizando redes de Petri coloridas**. *Revista Controle & Automação*. Vol.23 n° 3/mai. e jun.. 2012.
- MENESES. F. A. A. e BARROSO G.C. **Fusion of Supervisors in Discrete Event Systems Using Coloured Petri Nets**. *Buletinul Institutului Politehnic din Iasi. Automatic Control and Computer Science*, Tome LVIII (LXI), Fasc 4., 2011.
- MOLINARI, LEONARDO. **Inovação e Automação de Testes de Software**. 1.ed. – São Paulo. Editora Érica, 2010.
- MORAES, CÍCERO COUTO DE e CASTRUCCI, PLÍNIO DE LAURO. **Engenharia de Automação Industrial**. 2 e.d. Rio de Janeiro: LTC, 2010.
- MURATA, T., **Petri Nets: properties, analysis, and applications**, *Proc. IEEE* pp., 541-580, 1989.
- NAKASHIMA, K. and GUPTA, S.M., **Performance Evaluation of a Supplier Management System with Stochastic Variability**, *Int. J. Manufacturing Technology and Management*, Vol. 5, no. 1-2, 2003.
- PALOMINO, R., **Uma Abordagem para a Modelagem, Análise, Controle de Sistemas de Produção utilizando as Redes de Petri**. Florianópolis, 1995. Dissertação de Mestrado em Engenharia da Produção, UFSC, 1995.
- PETERSON, J.L., **Petri Net Theory and the Modeling of Systems**, *Prentice Hall*, 1981.
- PETRI, CARL A. **Communication With Automata**. Tese de Doutorado, Bonn, Universidade de Bonn, 1962.
- PROTH, JEAN-MARIE AND XIAOLA XIE. **PetriNets: A Tool for Design and Management of Manufacturing Systems**, *John Wiley& Sons. NewYork*, 1997.
- RAMADGE, P. and WONHAM W. **The Control of Discrete Event Systems**. *Proceedings of the IEEE*, 77. pp. 81-89, 1989.
- ROBIDOUX, R., H. XU, L. XING, and MC ZHOU, **Verificação Automática de Dinâmicas de Confiabilidade Diagramas de Blocos Usando Colored Petri Nets**, *IEEE Transactions on Systems, Man and Cybernetics*, Parte A, 40 (2), pp. 337-351, mar. 2010.

- SAITOU, K., MALPATHAK, S., QVAM, H., **Robust Design of Flexible Manufacturing Systems Using Colored Petri Nets and Genetic Algorithm**, *Journal of Intelligent Manufacturing*, 13, pp. 339-351, 2002.
- SANTOS, R.S.B., **Modelagem e Análise de Performance de Sistemas Flexíveis de Manufatura baseado em Redes de Petri Temporizadas: Estudo de Caso da Indústria Automobilística**. Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2008.
- SHIM, J.K. and SIEGEL J.G. *Operations Management. Conceptions & Strategies for quick mastery. Business review series. New York.* 1999.
- SILVEIRA, P.R., SANTOS, W.E., **Automação e Controle Discreto**. Editora Érica, 1ª edição, São Paulo, 1999.
- UZAM, M., ZHOU, M.C. **Uma Abordagem Interativa Síntese de Política de Prevenção Impasse Petri Net-Base para Sistemas Flexíveis de Manufatura**. *IEEE Trans. On Systems, Man and Cybernetics*, parte A, vol 37. pp. 362-371, 2007.
- VALENCIA R., **Modelagem de Sistemas C3I para Navios de Guerra através de Redes de Petri**. 2004. 185f., São Paulo, Dissertação de Mestrado – Escola Politécnica, Universidade de São Paulo, São Paulo, 2004.
- VERSTEEGH, F.; SALIDO, M.A; GIRET, A. *A Holonic Architecture for the Global Road Transportation System. Journal of Intelligent Manufacturing, Springer US*, v.21 n.1, pp.133-144, 2010.
- VIEIRA, D.V., **Método de Implementação de Controle de Sistemas a Eventos Discretos com a aplicação da Teoria de Controle Supervisório**. Tese de Doutorado. Universidade Federal de Santa Catarina, SC, 2007.
- VOOS, K. *Using Predicate/Transition-Nets to Model and Análise Distributed Data Base Systems. IEEE Trans. On Software Eng.* V.6, p-801-806, nov. 1980.
- WU, N. e MC, ZHOU, **Redes de Petri de token inteligentes para modelagem e controle de sistemas de manufatura automatizados reconfiguráveis com mudanças dinâmicas**, 2011.
- WU, N., MC ZHOU, e Z. LI, **Resource-Oriented Petri Net para Deadlock Avoidance em Sistemas Flexíveis de Montagem**, *IEEE Trans. em Sistemas, cara, e Cibernética*: pp 56-69, jan. 2008.
- ZHOU, M.C., DICCESSARES, F., RUDOLPH, D, L., **Design and Implementation of a Petri Net Based Supervisor for a Flexible Manufacturing System**, *Automatica*, Vol 28, n.6, 1199-108, 1992.
- ZHOU, M.C., McDERMOTT, K., PATEL, P., A., **Petri Nets Synthesis and Analysis of a Flexible Manufacturing System Cell**, *IEEE Transactions on Systems, Man and Cybernetics*, Vol,23, n.2, 1993.
- ZHOU, M.C., ZHURAWSKI, R., **Petri Nets and Industrial Applications: A Tutorial**, *IEEE Transactions on Industrial Eletronics*, Vol.41, n.6, 1994.
- ZHOU, MC e ZW LI, edição Especial sobre. **Redes de Petri para o Sistema de Controle e Automação**, *Asian Journal of Control* , Vo. 12, No. 3, pp 237-239, mai. 2010.