



UNIVERSIDADE FEDERAL DE GOIÁS - REGIONAL CATALÃO
UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E OTIMIZAÇÃO



Thiago de Almeida Bastos

**VALIDAÇÃO FORMAL DE MODELOS DE MANUFATURA FLEXÍVEL
COM LÓGICA DINÂMICA: O USO DE PETRI-PDL**

DISSERTAÇÃO DE MESTRADO

CATALÃO – GO, 2018

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR
VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES
NA BIBLIOTECA DIGITAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: ☒ **Dissertação** ☐ **Tese**

2. Identificação da Tese ou Dissertação:

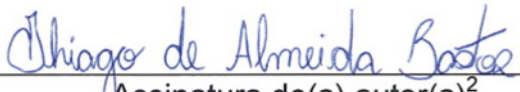
Nome completo do autor: Thiago de Almeida Bastos

Título do trabalho: Validação Formal de Modelos de Manufatura Flexível com Lógica Dinâmica: o uso de Petri-PDL

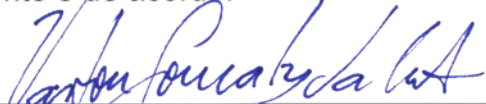
3. Informações de acesso ao documento:

Concorda com a liberação total do documento ☒ **SIM** ☐ **NÃO**¹

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.


Assinatura do(a) autor(a)²

Ciente e de acordo:


Assinatura do(a) orientador(a)²

Data: 07 / 02 / 2018

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

² A assinatura deve ser escaneada.

THIAGO DE ALMEIDA BASTOS

VALIDAÇÃO FORMAL DE MODELOS DE MANUFATURA FLEXÍVEL
COM LÓGICA DINÂMICA: O USO DE PETRI-PDL

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Modelagem e Otimização pela Universidade Federal de Goiás - Regional Catalão.

Orientador:
Vaston Gonçalves da Costa

CATALÃO – GO

2018

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

de Almeida Bastos, Thiago

Validação Formal de Modelos de Manufatura Flexível com Lógica
Dinâmica: o uso de Petri-PDL [manuscrito] / Thiago de Almeida
Bastos. - 2018.

lxxviii, 68 f.

Orientador: Prof. Dr. Vaston Gonçalves da Costa.

Dissertação (Mestrado) - Universidade Federal de Goiás, Unidade
Acadêmica Especial de Matemática e Tecnologia, Catalão,
Programa de Pós-Graduação em Modelagem e Otimização, Catalão, 2018.
Bibliografia.

Inclui siglas, abreviaturas, lista de figuras, lista de tabelas.

1. Lógica Dinâmica. 2. Sistemas Flexíveis de Manufatura. 3. Petri
PDL. 4. Métodos Formais. I. Gonçalves da Costa, Vaston, orient. II.
Título.

CDU 004.423.4



Defesa Nº 29

Ata de Defesa Pública – Dissertação de Mestrado

Aos 30 dias do mês de janeiro do ano de 2018, às 14 h: 00 min, reuniram-se os componentes da banca examinadora, professores Dr. VASTON GONÇALVES DA COSTA (presidente e orientador), Dr. MARCELO HENRIQUE STOPPA e Dr. BRUNO LOPES VIEIRA para, em sessão pública realizada na Sala de Reuniões (Centro Integrado de Pesquisa), da Regional Catalão (RC), da Universidade Federal de Goiás (UFG), procederem com a avaliação do trabalho intitulado: "Validação Formal de Modelos de Manufatura Flexível com Lógica Dinâmica: o uso de Petri-PDL", em nível de Mestrado, área de concentração *Modelagem e Otimização*, de autoria de THIAGO DE ALMEIDA BASTOS, discente do Programa de Pós-Graduação em Modelagem e Otimização (PPGMO) da UFG/RC. A sessão foi aberta pelo presidente da banca, que fez a apresentação formal dos membros da banca. A seguir, a palavra foi concedida ao discente que, dentro do tempo regulamentar, procedeu a apresentação de seu trabalho. Terminada a apresentação, cada membro da banca arguiu o candidato, tendo-se adotado o sistema de diálogo sequencial. Terminada a fase de arguição, procedeu-se a avaliação do trabalho. Os membros da banca consideraram o trabalho final: (☒) **Aprovado** (unanimidade) ou (☐) **Reprovado** (por maioria simples). Cumpridas as formalidades de pauta, às 15 h: 10 min a presidência da mesa encerrou a sessão e para constar, eu VASTON GONÇALVES DA COSTA, lavrei a presente Ata que, depois de lida e aprovada, segue assinada pelos membros da banca examinadora e pelo discente e, posteriormente, será homologada pelo Colegiado do PPGMO.

Catalão-GO, 30 de janeiro de 2018.


Prof.(a) Dr. (a): Vaston Gonçalves da Costa
Programa de Pós-Graduação em Modelagem e
Otimização, UFG/RC.
(Presidente da Banca)


Prof.(a) Dr. (a): Marcelo Henrique Stoppa
Programa de Pós-Graduação em Modelagem e
Otimização, UFG/RC.


Prof.(a) Dr. (a): Bruno Lopes Vieira
Instituto de Computação
Universidade Federal Fluminense


Discente: Thiago de Almeida Bastos
Programa de Pós-Graduação em Modelagem e
Otimização, UFG/RC.

Agradecimentos

A Universidade Federal de Goiás, ao programa de Modelagem e Otimização pela oportunidade e credibilidade.

Ao orientador e amigo Vaston Gonçalves da Costa pelos ensinamentos transmitidos ao longo desta pesquisa, de fundamental importância para a conclusão deste trabalho.

Aos colegas de trabalho do Instituto Federal Goiano - Campus Urutaí que de alguma forma facilitaram a minha dedicação aos estudos. E também às iniciativas institucionais de apoio a capacitação do servidor, das quais recebi suporte financeiro por meio da bolsa PIQ.

A minha família, que sempre me apoia e segue firme comigo nesta caminhada. E por fim, agradecer a Deus que é Pai e fonte de inspiração, sempre nos abençoando com o dom da vida e da sabedoria. A todos, meu muito obrigado!

RESUMO

BASTOS, T. DE A. *Validação Formal de Modelos de Manufatura Flexível com Lógica Dinâmica: o uso de Petri-PDL*. 2018. 68 f. Dissertação (Mestrado em Modelagem e Otimização) – Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás - Regional Catalão, Catalão – GO.

Este trabalho busca contribuir com a automação de linhas de produção e propõe uma metodologia para a verificação formal de modelos de manufatura flexível a partir da ferramenta Petri-PDL. O conceito Petri-PDL baseia-se em uma lógica multimodal associada ao esquema definido para o problema com as redes de Petri para especificar e modelar problemas sequenciais demonstrando em provas lógicas a corretude de propriedades inferidas pelo modelo. Este tratamento formal é adaptado para o tratamento de processos sequenciais flexíveis, uma vez que estes modelos são usados em muitas outras aplicações com redes de Petri. Serão considerados modelos de sistema de produção flexível encontrados na revisão sistemática para avaliar a eficiência de seu modelo e sua adaptação a este refinamento formal.

Palavras-chaves: Lógica Dinâmica, Sistemas Flexíveis de Manufatura, Petri-PDL, Métodos Formais.

ABSTRACT

BASTOS, T. DE A. *Validação Formal de Modelos de Manufatura Flexível com Lógica Dinâmica: o uso de Petri-PDL*. 2018. 68 f. Master Thesis in Modelling and Optimization – Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás - Regional Catalão, Catalão – GO.

This master's thesis seeks to contribute to the automation of production lines, and proposes a methodology for the formal verification of flexible manufacturing models by the Petri-PDL tool. The Petri-PDL framework is based on a multimodal logic associated with a scheme defined for the problem with the Petri nets to specify and model sequential problems demonstrating in logical proofs the correctness of properties inferred by the model. This formal treatment is adapted for the treatment of flexible sequential processes, since these models are used in many other applications with Petri nets. They will be considered models of flexible production system found in the systematic review to evaluate the efficiency of its model and its adaptation to this formal refinement.

Keywords: Dynamic Logic, Flexible Manufacturing Systems, Petri-PDL, Formal Methods.

LISTA DE FIGURAS

Figura 2.1 – Representação de elementos básicos de um FMS.	25
Figura 2.2 – Manufatura Integrada por Computador.	27
Figura 2.3 – Subdivisões de Controle FMS.	28
Figura 2.4 – Exemplo de uma rede de Petri com três marcações	33
Figura 2.5 – Redes de Petri Básicas	34
Figura 2.6 – Exemplo de composição utilizando os três tipos básicos	35
Figura 2.7 – Exemplos de operações de sequência e junção	35
Figura 2.8 – Exemplos de três casos de repetições.	36
Figura 2.9 – Exemplo de operações de conflito.	36
Figura 2.10 – Rede de Petri para um sistema de abertura de porta com senha.	37
Figura 3.1 – Principais vertentes encontradas para o tópico (a).	42
Figura 3.2 – Desenvolvimento da pesquisa com redes de Petri para modelagem FMS.	43
Figura 4.1 – Uma rede de Petri com duas marcações.	51
Figura 5.1 – Modelo de processo de usinagem.	55
Figura 5.2 – Modelo de processo de usinagem controlado.	56
Figura 5.3 – Modelo de processo com recursos compartilhados.	58
Figura 5.4 – Sistema de produção flexível com recursos compartilhados e duas sequên- cias de tarefas.	59
Figura 5.5 – Rede de Petri para um modelo de produção com duas tarefas.	61

LISTA DE TABELAS

Tabela 3.1 – Dados iniciais da pesquisa sistemática.	40
Tabela 3.2 – Principais autores com publicações no tópico (b).	42

LISTA DE ABREVIATURAS E SIGLAS

CAD — *Computer-Aided Design*, do português Desenho Assitado por Computador

CAM — *Computer-Aided Manufacturing*, do português Manufatura Assitada por Computador

CAPES — Portal de Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CIM — *Computer Integrated Manufacturing*, do português Manufatura Integrada por Computador

FMC — *Flexible Manufacturing Cells* do português Células de Manufatura Flexível

FMS — *Flexible Manufacturing System* do português Sistema de Produção Flexível

Máquina CNC — Máquinas de comando numérico computadorizado

SAT — Problema de Satisfatibilidade Booleana

SUMÁRIO

1	INTRODUÇÃO	19
2	CONCEITOS BÁSICOS	23
2.1	Sistemas Flexíveis de Produção	23
2.1.1	Histórico	23
2.1.2	Arquitetura FMS	24
2.1.3	Implementação de Modelos FMS	27
2.2	Definições de Lógica	29
2.2.1	Lógica Proposicional	29
2.2.2	Lógica Modal Básica	30
2.2.3	Lógica Dinâmica Proposicional	31
2.3	Redes de Petri	33
2.3.1	Redes Marcadas	33
3	REVISÃO SISTÊMICA	39
3.1	Metodologia da Pesquisa Sistemática	39
3.2	Discussões e Resultados da Pesquisa Sistemática	41
3.2.1	Métodos Formais na Modelagem com redes de Petri	43
4	LÓGICA DINÂMICA PROPOSICIONAL PARA REDES DE PETRI: PETRI-PDL	47
4.1	Linguagem e Semântica	47
4.2	Sistema Axiomático	50
4.3	Algumas Aplicações	50
5	VERIFICAÇÃO E VALIDAÇÃO NA MODELAGEM FMS	53
5.1	Avaliação de Modelos	54
5.2	Análise Formal com as Petri-PDL	57
6	CONCLUSÕES E TRABALHOS FUTUROS	63
	REFERÊNCIAS	65

Capítulo 1

INTRODUÇÃO

A comercialização de produtos especializados tem se modernizado ao longo dos tempos devido à automação industrial que, por sua vez, aumenta a capacidade de alta produtividade. Neste sentido, sistemas que permitam uma produção flexível tem despertado grande atenção das indústrias, por permitirem adequações e ajustes que demandam pouco tempo e baixo custo.

Desenvolvido na década de 60 do século XX por David Williamson, o sistema de produção flexível (FMS, do inglês *Flexible Manufacturing System*), proposto inicialmente para atender demandas da indústria siderúrgica (CHU, 1993), vem sendo adaptado para uma variedade de aplicações industriais com êxito (CAGLIANO; SPINA, 2000; NAGALINGAM; LIN, 1999). A sistemática de um FMS se adapta à rápida divisão de produtos, ao seccionamento da produção e apresenta baixo custo de produção por se ajustar bem ao ambiente de mercado imprevisível dos últimos tempos. (GROOVER, 2010). Desde sua concepção até os dias atuais os FMS passaram por grande evolução, principalmente por conta da grande ascensão dos sistemas computacionais, considerando que estes são de suma importância para se lidar com ocorrência de sequenciamento com precedência, controle de acesso, dentre outras características, em tempo de execução (VISWANADHAM; NARAHARI; JOHNSON, 1992; GROOVER, 2010).

Por conta da flexibilidade inerente dos FMS, há relatos de falhas quando ocorre aumento de produtividade e ajustes na planta de produção (CAPKOVIC, 2015; BASAK; ALBAYRAK, 2015; BARUWA; PIERA, 2015). Uma vez que problemas em planta de produção podem gerar transtornos à indústria, notadamente os de ordem financeira, antes de sua implementação as FMS devem ser projetadas, modeladas e testadas/validadas. Dentre as técnicas empregadas para se modelar as FMS destaca-se o uso de redes de Petri (BHUIYAN; GOUW; YAZDI, 2011; BOURDEAUD'HUY; TOGUYENI, 2006; CHAN; WANG, 1993; COSSINS; FERREIRA, 1992; YAN *et al.*, 1998).

Redes de Petri, postuladas na década de 60 do século XX ¹ (PETRI, 1962) para modelar comunicação com autômatos, têm se mostrado uma ferramenta gráfica e matemática para se modelar diversos tipos de sistemas (SILVA, 2012). Dentre suas principais aplicações estão a possibilidade de especificar, analisar comportamento lógico e avaliar o desempenho de sistemas.

Ademais, ainda se pode acoplar às redes de Petri mecanismos de análise poderosos, os quais permitem a verificação de propriedades e da corretude de sistemas (MURATA, 1989). Destaca-se ainda a grande quantidade de paradigmas lógicos que podem ser associados ao modelo/especificação em redes de Petri. O que, por sua vez, permite inferir propriedades de forma decidível, correta e completa, dependendo da lógica empregada.

A partir de uma pesquisa sistemática na literatura por trabalhos que empregam rede de Petri na modelagem de FMS, foi verificado, a ausência de estudos que utilizam verificação e validação de modelos através da formalização por via lógica. Por outro lado, também foram relacionadas, algumas tecnologias que, apesar de não validarem os dados processados, possuem resultados satisfatórios para variantes como: redução de erros de disputa de recursos, organização de fatores de risco e agendamento da produção. A validade destas propostas e a correspondência aos resultados apresentados serão colocadas em prova a partir da avaliação formal apresentada neste trabalho.

Das várias lógicas existentes - sejam, proposicionais, de predicados, modal, linear ou dinâmica - suas associações com redes de Petri para verificação formal acontecem de acordo com a aplicação. Para jogos, um exemplo encontrado é o trabalho de Oliveira, Julia e Passos (2011), que utiliza uma associação entre redes de Petri e lógica linear denominada *Workflow nets*.

Dentre os modelos lógicos que podem ser associados à redes de Petri para modelar e verificar propriedades de forma correta e completa, destaca-se o que emprega lógica proposicional dinâmica, Lopes, Benevides e Haeusler (2014a) propõem o que denominaram Petri-PDL como uma nova ferramenta de modelagem e verificação de sistemas.

Neste trabalho é apresentado a aplicação de redes de Petri e lógica dinâmica em sistema de produção flexível. Esta abordagem visa ampliar a utilização da ferramenta Petri-PDL para um novo ambiente de aplicação, demonstrando sua adaptação e a validade da proposta. Além de apresentar o arcabouço teórico necessário para compreender os conceitos abordados no estudo, também é apresentada uma aplicação de Petri-PDL na modelagem e verificação de um sistema de produção flexível selecionado na revisão sistemática.

O presente texto, que serve ao propósito de atender as exigências do programa de pós-graduação no que concerne ao exame de defesa da dissertação, será subdividido em capítulos da seguinte forma: no capítulo 2 serão apresentados os conceitos básicos neces-

¹ mesmo período de concepção das FMS.

sários para compreensão do trabalho; no capítulo 3 será apresentada a revisão sistemática da literatura que atesta a originalidade da abordagem da pesquisa realizada; no capítulo 4 se apresenta o conceito e exemplos de Petri-PDL; no capítulo 5 é apresentada uma aplicação de Petri-PDL em um sistema de produção flexível e por fim as conclusões e alinhamentos para trabalhos futuros.

Capítulo 2

Conceitos Básicos

A aplicação de métodos formais na avaliação da modelagem de ambientes de produção flexível se dá a partir da necessidade do entendimento e da correção dos dados avaliados neste processo de modelagem que, conseqüentemente, são aproveitados em outras etapas do controle e gerenciamento da automação industrial por computadores.

Neste capítulo serão abordados os conceitos básicos dos sistemas flexíveis de produção, as ferramentas para modelagem destes ambientes, e também, a linguagem lógica associada como método formal de verificação/validação dos dados. O entendimento destes conceitos será primordial para avaliar a relevância dos dados resultantes da aplicação de métodos formais neste processo.

2.1 Sistemas Flexíveis de Produção

Um sistema de produção flexível (FMS) é uma célula ou um conjunto delas altamente automatizadas, composta por um grupo de estações de processamento, interligadas por um sistema automatizado de manuseio e armazenamento de materiais e controladas por um computador integrado ao sistema. Suas características representam a necessidade de conciliar a eficiência da linha de produção com a flexibilidade do chão de fábrica para satisfazer uma demanda versátil a um custo razoável em ambientes com demanda pequena ou média, e com exigências de mercado muito altas (CHU, 1993).

2.1.1 Histórico

Originalmente o sistema de produção flexível foi idealizado para a indústria siderúrgica e o processo de usinagem, exigindo o desenvolvimento prévio das máquinas de controle numérico (NC) daquele tempo, para máquinas controladas por computador. O conceito FMS é creditado a David Williamson, engenheiro britânico empregado pela Molins durante meados da década de 1960. Molins solicitou, naquela época, uma patente para a invenção,

que foi concedida em 1965. O conceito foi chamado inicialmente de *System 24*, pois se acredita que o grupo de máquinas/ferramentas que compunham o sistema poderia funcionar 24 horas por dia, das quais 16 horas seriam sem assistência de trabalhadores humanos, somente controladas por computadores (GROOVER, 2010, p. 935).

A era da produção flexível, que atualmente possui variadas composições, tem início no período do aparecimento na Grã-Bretanha com a Molins, justamente após a Segunda Guerra Mundial, quando o Japão, diferente dos EUA, desenvolveu uma abordagem alternativa ao sistema de produção em massa. Tal abordagem consistia em utilizar equipes de trabalhadores com multi-habilidades, e equipadas com ferramentas automatizadas e flexíveis, para produzir uma variedade de produtos em pequenos volumes. Uma leitura complementar pode ser feita em (STRONG, 2008).

Foram introduzidas inúmeras técnicas e filosofias de melhorias na linha de produção, visando a redução de custos e a alta qualidade dos produtos. Todos estes conceitos deram origem aos conceitos que conhecemos hoje de produção adaptada ao ritmo e à necessidade. Naquela época, empresas foram obrigadas a abandonar o sistema de produção em massa e investir recursos financeiros de alta monta em tecnologias, tornando o sistema produtivo mais flexível do que antes.

2.1.2 Arquitetura FMS

Uma das principais características de cenários FMS é a capacidade de uma mesma planta de produção industrial poder ser adaptada em diferentes sequenciamentos sendo possível apresentar diferentes alocações e, ao mesmo tempo, manter suas características básicas. Consequentemente, um sistema de produção flexível é projetado para produzir peças (ou produtos) dentro de uma variedade de estilos, tamanhos e processos.

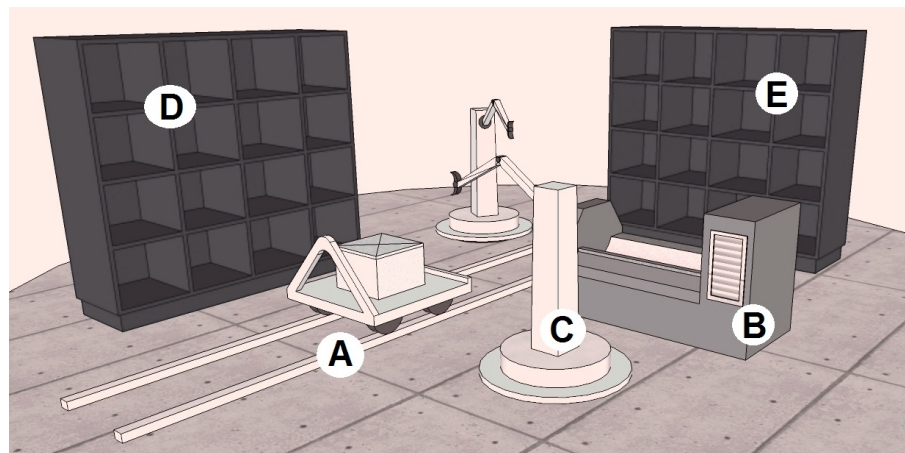
Projetos FMS passam por uma evolução rápida e constante de desenvolvimento, sendo instalados numa variedade de ambientes onde cada planta de produção costuma assumir suas próprias configurações. Contudo, alguns autores dissertam sobre suas principais características, conforme descreve a Figura 2.1. Huang e Chang (1992) sugerem que um FMS pode ser decomposto na combinação de diversas Células de Manufaturas Flexíveis (FMC do inglês *Flexible Manufacturing Cells*), outros autores como Cossins e Ferreira (1992), por exemplo, vão um pouco além e dizem que FMS é um grupo de FMC interconectadas por um sistema de transporte de peças e controlados por um sistema computacional comum.

A capacidade de realizar várias operações por uma máquina flexível é composta tanto pelo sistema automático de armazenamento e recuperação das ferramentas de controle quanto pela possibilidade de, a qualquer momento da produção, realizar a inclusão de novos programas de usinagem.

O modelo apresentado na Figura 2.1 representa o chão de fábrica em uma indústria

sob a filosofia FMS. Nele estão presentes as células de manufatura flexível entre outras partes características de sistemas deste tipo. Basicamente modelos FMS são compostos por: A - Equipamentos de Transporte, que podem ser os veículos guiados automaticamente, como está na figura, ou mesmo as esteiras mecanizadas que interconectam as células de produção; B - Máquinas de Torno/Perfuração/Frisagem, todas máquinas de comando numérico computadorizado (CNC) que, como o próprio nome diz, refere-se ao controle de máquinas programáveis por computador, executando os processos propriamente ditos para a fabricação dos produtos industrializados; C - Robôs de controle e movimentação capazes de movimentar a peça manufaturada além de executar montagens; D - Pilha de sucata para a dispensa de restos de materiais; E - Pontos de aquisição de peças para produção.

Figura 2.1 – Representação de elementos básicos de um FMS.



Fonte: Adaptado de Bourdeaud'huy e Toguyeni (2006)

O processo de produção passa por algumas etapas desde a concepção das peças até a entrega do produto finalizado. Considerando o modelo da Figura 2.1 como exemplo, pode-se perceber que a produção tem início no ponto E com a aquisição das peças. Os equipamentos em A, C, B são de uso compartilhado e sequenciado pois fazem a movimentação das peças e operações sobre elas. Dependendo da finalidade da produção as operações serão sequenciadas de formas diferentes. Por fim a peça finalizada é transportada em A até a saída do ambiente de produção

Os sistemas flexíveis normalmente se dividem em duas categorias principais: as "*machine flexibility*" que significa Flexibilidade da Máquina, se refere à habilidade do sistema de mudar para produzir novos produtos, e à habilidade de mudar a ordem de operações executadas. A segunda categoria é chamada "*routing flexibility*" ou flexibilidade de roteamento, que consiste na habilidade de usar múltiplas máquinas para fazer a mesma operação, também é a habilidade do sistema de absorver grandes mudanças, seja no volume, capacidade ou capacidade (BARUWA; PIERA, 2015).

Sistemas de fabricação são definidos como um conjunto de elementos de hardware

e software em interação dinâmica. Quando sistemas necessitam se adaptar ao ambiente de produção em tempo de execução, são geralmente caracterizados pela capacidade de produzir, ao mesmo tempo, diferentes tipos de produtos e a possibilidade de usar diferentes recursos para executar uma mesma operação.

A composição de uma planta de produção flexível pode assumir várias configurações dependendo do número de produtos a produzir e sua especificidade, pelo fato destas configurações assumir uma gama variada de produtos de porte pequeno mas em quantidades relativamente altas (GROOVER, 2010). A arquitetura FMS descrita na Figura 2.2 além de, sistemas de controle, informação e comunicação, é composta por:

Gerente FMS Responsável por integrar e gerenciar o ambiente de produção flexível e toda sua automação. É indispensável que sua especificação possua questões muito bem definidas como em um sistema operacional com: linguagem de programação, rede e protocolo de comunicação, grau de confiabilidade e segurança das informações (ALETAN, 1991). Além disso, ser capaz de:

- promover a comunicação em nível hierárquico;
- realizar o planejamento da produção;
- gerenciar o fluxo de materiais.

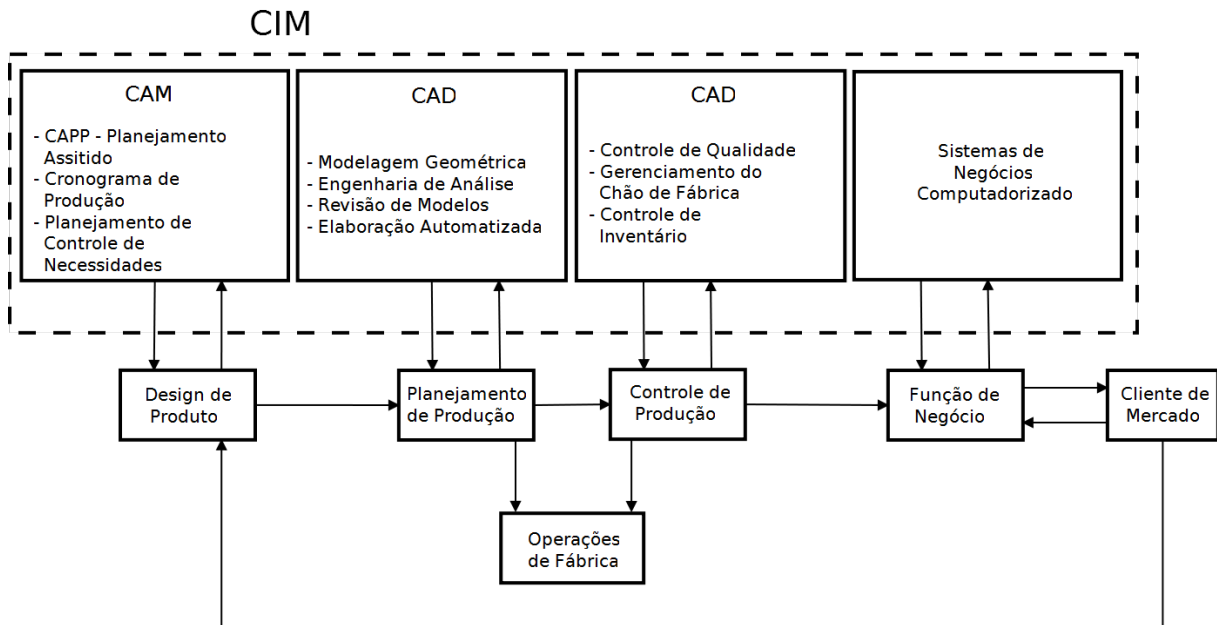
CIM (Manufatura Integrada por Computador) estão presentes nos sistemas de engenharia, junto a outros tipos de modelos controlados por computador como de desenho (CAD) e de integração da manufatura (CAM). Um sistema CIM (BHUIYAN; GOUW; YAZDI, 2011), pode ser definido como um sistema que mantém o computador no ponto central de controle da fábrica e abrange a informatização dos processos de fabricação e montagem, bem como o fluxo de informações para controle de produção, qualidade, manutenção, manuseio de materiais, e controle de estoque em um sistema completamente integrado. Dentre suas principais características, destacam-se:

- Integração;
- Escalabilidade;
- Segurança.

Estas propriedades garantem que os programas executados sejam completamente integrados sem requerer redundância dos dados evitando, assim, desperdício de investimentos (CAPUTO *et al.*, 1998).

FMC (Células de Manufatura Flexível) são literalmente estações de trabalho. Normalmente compostas pelo módulo FMS onde há efetivamente manufatura de produtos e representa a unidade básica de produção. FMC é o elemento chave na implementação de métodos da produção flexível (HUANG; CHANG, 1992).

Figura 2.2 – Manufatura Integrada por Computador.



Fonte: Adaptada de [Groover \(2010\)](#)

Um sistema FMS precisa ser realmente flexível e, conseqüentemente, capaz de responder rápido e com eficiência às mudanças de ambiente, às variações de demanda, às mudanças nas especificações de produto, bem como às situações dinâmicas dentro do próprio sistema. Por isso são utilizados sistemas extremamente complexos exigindo um controle rápido e preciso o suficiente para corrigir problemas, evitando ineficiências indesejadas no processo de produção ([CAGLIANO; SPINA, 2000](#); [NAGALINGAM; LIN, 1999](#)).

Além disso, o sistema deve estar preparado para a integração de novos módulos de acordo com a necessidade da produção ou a demanda de produtos. Estas precauções na montagem do ambiente de produção automatizado dão aporte ao bom funcionamento do sistema em uma situação de produção com escopo que fugiu dos limites pré-definidos, o que ocasionalmente pode ocorrer.

2.1.3 Implementação de Modelos FMS

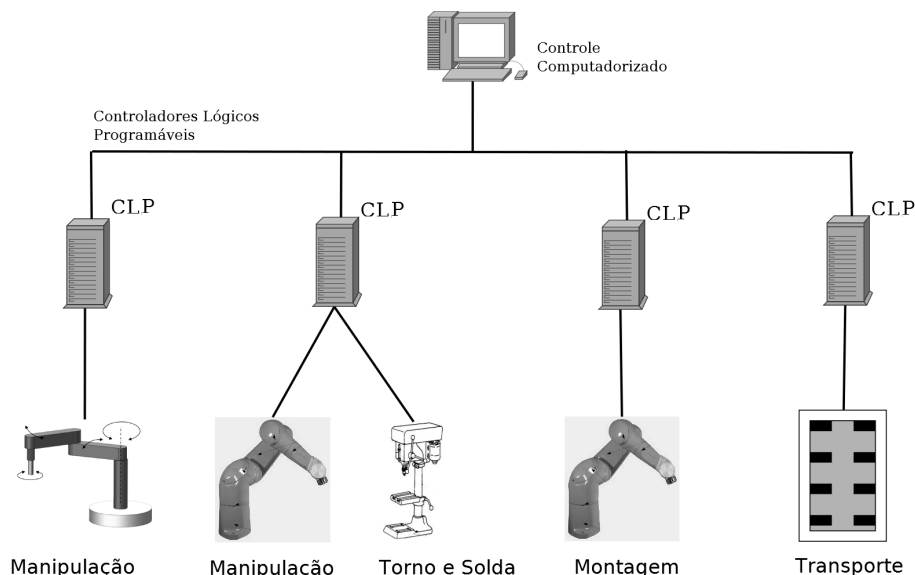
Existem várias técnicas para a modelagem do design FMS e, conforme a aplicação, podem ser usadas tanto a inteligência artificial e as redes neurais como também as redes de Petri dos mais variados tipos. Neste trabalho serão utilizadas em uma abordagem formal as redes de Petri. As redes de Petri são amplamente utilizadas nesses sistemas, pois o controle de sistemas FMS é similar ao controle de sistemas dinâmicos a eventos discretos, os quais podem ser perfeitamente modelados por rede de Petri. A seguir são listadas algumas referências que utilizam as redes de Petri:

1. avaliação de desempenho de sistemas FMS: (CHAN; WANG, 1993);
2. uso de modelos abstratos para análise de FMS: (KOCHIKAR; NARENDRAN, 1994);
3. uma estrutura para medir a flexibilidade de FMS: (KOCHIKAR; NARENDRAN, 1992).

O desempenho do agendamento de tarefas programado pelo controlador do FMS é geralmente avaliado com base em determinados critérios em relação a encomenda e a produtividade. A principal diferença entre modelos de produção usuais com as manufaturas flexíveis é a relação entre a capacidade de produzir uma gama de produtos diferentes enquanto as linhas de montagem tradicionais normalmente produzem um volume elevado de um número limitado de produtos

Cada vez mais a fabricação flexível vem sendo utilizada por suas vantagens em relação a qualidade, redução de mão-de-obra e custo de estocagem de matérias, apresentando melhoras contínuas e inovações constantes. A Figura 2.3 mostra o *layout* da comunicação entre o sistema de controle central e as ferramentas do sistema através dos controladores lógicos programáveis (CLPs). Este processo controla, de forma separada, uma série de máquinas conectadas por um sistema de manuseio, montagem e transporte de materiais computacionalmente. Neste escopo está representada a maioria das subdivisões que podem assumir um sistema de produção deste tipo.

Figura 2.3 – Subdivisões de Controle FMS.



Fonte: Adaptado de Ribeiro, II e Lepikson (2009)

Apesar das inúmeras vantagens e adaptações que os modelos de produção receberam ao longo dos anos, o processo de produção ainda não está totalmente a salvo de ocorrências. Como todo processo que é adaptado ao cotidiano e sua necessidade, constantemente a literatura é atualizada compondo novos recursos e aprendizados.

A fácil adaptação encontrada para a modelagem de sistemas de produção flexível com redes de Petri pode ainda ser melhorada através dos métodos formais, como pretende apontar esta pesquisa, garantindo que os sistemas comportem de acordo com a forma em que suas propriedades são modeladas. Isso porque, em ambientes de produção principalmente, a execução de tarefas pode representar a transição de eventos discretos, isto é, com condições a satisfazer e resultados esperados.

A fim de trazer mais segurança aos softwares que implementam a integração software/ferramenta com a diminuição de cenários imprevistos, será avaliado o uso de uma metodologia formal com a Lógica Dinâmica Proposicional associada às redes de Petri. Com foco em sistemas concorrentes, este conjunto de técnicas irá fundamentar e certificar sistemas. No Capítulo 5 será avaliado a ferramenta Petri-PDL[4] enquanto seus resultados são comparados aos modelos já encontrados pela revisão sistêmica [3] em razão da análise e prevenção de erros quanto ao cumprimento de sua especificação.

2.2 Definições de Lógica

O raciocínio lógico e as provas formais são instrumentos para demonstrar a verdade de um enunciado que pode ser expresso na forma de uma oração, onde, a partir de inferências e consequências lógicas se deduz uma premissa universal. Assim, um argumento é uma prova que visa obter afirmações conclusivas, a partir de premissas, pela aplicação de regras de inferência (GIACOMO; MASSACCI, 2000). Nesta seção serão abordados os conceitos básicos da lógica dinâmica, extensão da lógica modal, que deu origem à linguagem dinâmica proposicional utilizada neste trabalho.

Os fundamentos teóricos e a terminologia utilizada neste trabalho seguem a apresentação do trabalho de Nalon *et al.* (2015) e os trabalhos relacionados a convenção de Petri-PDL de Lopes, Benevides e Haeusler (2014a). Estes conhecimentos serão fundamentais para a composição desta pesquisa evidenciando a importância de se aplicar técnicas formais à modelagem da automação industrial, principalmente para verificação de propriedades e comportamento de programas e sistemas sequenciais, tais como correção, imparcialidade e equivalência.

2.2.1 Lógica Proposicional

Quando pretende-se lidar apenas com proposições, como o próprio nome já diz, utiliza-se a Lógica Proposicional. Para definir proposições devemos levar em conta apenas as orações, ou seja, frases ou declarações do tipo verdadeiro ou falso. Para sua definição são necessários um alfabeto, conectivos e símbolos proposicionais bem definidos.

Para apresentar as definições usaremos a notação na forma de Backus-Naur (BNF) que define sintaxe de linguagens. Uma declaração BNF é sempre do tipo "< variável> ::=

expressão" o que significa que uma "declaração" é um elemento do alfabeto e o símbolo "::<=" quer dizer que pode produzir ou pode alcançar uma "expressão" composta de vários elementos a partir do alfabeto.

Definição 1 (Linguagem Lógica Proposicional). A linguagem da lógica proposicional contém um conjunto Φ de fórmulas atômicas definidos por:

$$\langle \text{fórmula} \rangle ::= a \mid \perp \mid \neg \langle \text{fórmula} \rangle \mid \langle \text{fórmula} \rangle \wedge \langle \text{fórmula} \rangle$$

onde $a \in \Phi$.

Denota-se por Φ o conjunto de fórmulas, ou seja, *strings* geradas a partir da gramática da linguagem.

2.2.2 Lógica Modal Básica

A sintaxe da linguagem PDL definida a seguir na Seção 2.2.3, parte das lógicas modais. As lógicas modais permitem exprimir de modo explícito as noções de necessidade e possibilidade. De forma geral, a principal diferença desta lógica para a lógica proposicional consiste na presença de dois operadores unários que relativizam o conceito de verdade de uma fórmula Φ . Estes operadores denotados pelos símbolos $\Box$ e $\Diamond$ são conhecidos como modalidades denotando o sentido de necessidade e possibilidade respectivamente.

Definição 2 (Linguagem da Lógica Modal). Constituída por um conjunto enumerável Φ de símbolos proposicionais, os operadores booleanos $\neg$, $\wedge$, $\vee$ e $\rightarrow$, as constantes $\top$ e $\perp$ e dois operadores modais (ou de modalidade): $\Diamond$ e $\Box$. Suas fórmulas são definidas da seguinte forma:

$$\varphi ::= p \mid \top \mid \perp \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \Diamond \varphi \mid \Box \varphi,$$

onde $p \in \Phi$. $\Box \varphi$ pode ser lido como "é necessário que φ seja verdade".

As fórmulas da Lógica Modal possuem uma semântica de mundos possíveis, como é normalmente formada, baseando-se em estruturas matemáticas conhecidas como estruturas do tipo *Frame* e Modelo de Kripke.

Definição 3. Um *Frame* é formado por uma tupla $F = (W, R)$, onde W é um conjunto não vazio de mundos (estados ou vértices) e R é uma relação binária sobre W , isto é, $R \subseteq W \times W$.

Definição 4. Um Modelo da Lógica Modal é uma tupla $M = (F, V)$, onde F é uma estrutura e V é uma função de valoração que mapeia símbolos proposicionais em subconjuntos de W , isto é, $V : \Phi \rightarrow P(W)$.

Definição 5. Seja $\mathcal{M} = (F, V)$ um modelo. A noção de satisfatibilidade de uma fórmula φ em um modelo $\mathcal{M}$ em um mundo v , denotada por $\mathcal{M}, v \models \varphi$, pode ser definida indutivamente da seguinte forma:

1. $\mathcal{M}, v \Vdash p$ sse $v \in V(p)$;
2. $\mathcal{M}, v \Vdash \top$ sempre;
3. $\mathcal{M}, v \Vdash \perp$ nunca;
4. $\mathcal{M}, v \Vdash \neg\varphi$ sse $\mathcal{M}, v \not\Vdash \varphi$;
5. $\mathcal{M}, v \Vdash \varphi_1 \wedge \varphi_2$ sse $\mathcal{M}, v \Vdash \varphi_1$ e $\mathcal{M}, v \Vdash \varphi_2$;
6. $\mathcal{M}, v \Vdash \varphi_1 \vee \varphi_2$ sse $\mathcal{M}, v \Vdash \varphi_1$ ou $\mathcal{M}, v \Vdash \varphi_2$;
7. $\mathcal{M}, v \Vdash \varphi_1 \rightarrow \varphi_2$ sse $\mathcal{M}, v \Vdash \varphi_1$ então $\mathcal{M}, v \Vdash \varphi_2$;
8. $\mathcal{M}, v \Vdash \Diamond\varphi$ sse há $w \in W$ tal que vRw e $\mathcal{M}, w \Vdash \varphi$;
9. $\mathcal{M}, v \Vdash \Box\varphi$ sse para todo $w \in W$ tal que vRw , $\mathcal{M}, w \Vdash \varphi$.

A partir destes conceitos a noção semântica de satisfatibilidade de uma fórmula lógica pode ser entendida a partir de intuições, como a seguir. A fórmula $\Diamond\varphi$ é satisfeita em um mundo v se existe um mundo w acessível a partir de v através da relação R que satisfaz a fórmula φ . A fórmula $\Box\varphi$ é satisfeita em um mundo v se a fórmula φ é satisfeita em todo mundo w que é acessível a partir de v através da relação R .

2.2.3 Lógica Dinâmica Proposicional

A Lógica Dinâmica Proposicional é uma lógica multimodal que consiste em considerar uma generalização de estruturas *Frames* com mais de uma relação binária e uma linguagem com mais de um par de modalidades. Nesta seção, será apresentado a lógica dinâmica proposicional (PDL) projetada para a descrição e verificação de propriedades de programas e sistemas sequenciais.

Além de *fórmulas*, as PDL, possuem a noção de *programas*. A linguagem possui um conjunto finito Π de *programas básicos* e programas mais complexos que podem ser obtidos a partir dos operadores da linguagem.

Definição 6 (Linguagem PDL). consiste de um conjunto enumerável Φ de símbolos proposicionais, um conjunto finito Π de programas básicos, os conectivos booleanos $\neg$ e $\wedge$, construtores de programas; (composição sequencial), $\cup$ (escolha não-determinística) e $*$ (iteração) e um operador modal (de modalidade) $\langle\pi\rangle$, para cada programa π . As fórmulas são definidas da seguinte forma:

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\pi\rangle\varphi, \text{ com } \pi ::= a \mid \pi_1 ; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*,$$

onde $p \in \Phi$ e $a \in \Pi$.

As fórmulas desta lógica são avaliadas em um Modelo da forma de estruturas de Kripke. São válidos o seguinte padrão de abreviaturas: $\perp \equiv \neg \top$, $\varphi \vee \phi \equiv \neg(\neg\varphi \wedge \neg\phi)$, $\varphi \rightarrow \phi \equiv \neg(\varphi \wedge \neg\phi)$ e $[\pi] \varphi \equiv \neg \langle \pi \rangle \neg\varphi$.

Cada programa π corresponde a uma modalidade $\langle \pi \rangle$, onde a fórmula $\langle \pi \rangle \varphi$ significa que após a execução de π , φ eventualmente será verdade, considerando que π também é verdade. Existe também a possibilidade de utilizar $[\pi]\varphi$ (como uma abreviação para $\neg\langle \pi \rangle \neg\varphi$) indicando que a propriedade denotada por φ é mantida após qualquer execução possível de π .

A semântica PDL é formada normalmente, utilizando um diagrama de transição, que consiste em um conjunto de estados e relações binárias (um para cada programa) indicando a possível execução de cada programa em cada estado. Na literatura PDL, um diagrama de transição é chamado de *Frame*.

Definição 7 (*Frame PDL*). é uma tupla $\mathcal{F} = \langle W, R_\pi \rangle$ onde

- W é um conjunto não vazio de estados;
- R_π é uma relação binária em W , para cada programa básico $a \in \Pi$;
- Uma relação binária R_π , para cada programa π , é definida por indução a seguir
 - $R_{\pi_1;\pi_2} = R_{\pi_1} \circ R_{\pi_2}$;
 - $R_{\pi_1 \cup \pi_2} = R_{\pi_1} \cup R_{\pi_2}$;
 - $R_{\pi^*} = R_\pi^*$ onde R_π^* denota o fecho transitivo-reflexivo de R_π .

Definição 8 (*Modelo PDL*). é um par $\mathcal{M} = \langle \mathcal{F}, V \rangle$, onde $\mathcal{F}$ é um *frame* PDL, e V é uma função de avaliação $V : \Phi \rightarrow 2^W$.

A noção semântica de satisfatibilidade para PDL é definida a seguir:

Definição 9. Seja $\mathcal{M} = (\mathcal{F}, V)$ um modelo. A noção de satisfatibilidade de uma fórmula φ em um modelo $\mathcal{M}$ em um estado w , notação $\mathcal{M}, w \models \varphi$, pode ser definida indutivamente da seguinte forma:

1. $\mathcal{M}, w \models p$ sse $w \in V(p)$;
2. $\mathcal{M}, w \models \top$ sempre;
3. $\mathcal{M}, w \models \neg\varphi$ sse $\mathcal{M}, w \not\models \varphi$;
4. $\mathcal{M}, w \models \varphi_1 \wedge \varphi_2$ sse $\mathcal{M}, w \models \varphi_1$ e $\mathcal{M}, w \models \varphi_2$;
5. $\mathcal{M}, w \models \langle \pi \rangle \varphi$ sse existe $w' \in W$ tal que $w R_\pi w'$ e $\mathcal{M}, w' \models \varphi$

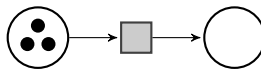
Observação 1. PDL possui um sistema dedutivo baseado no *Tableaux* (GIACOMO; MASSACCI, 2000) *web* implementado por Schmidt (2003). Cabe ressaltar que PDL de primeira ordem é indecidível conforme Harel, Tiuryn e Kozen (2000) e que o problema SAT (Problema de Satisfatibilidade Booleana) é, para a versão proposicional, EXPTIME-Completo, ou seja possuem solução computacional em tempo polinomial não conhecida, (provado por uma redução do problema *Two-Person Corridor Tiling* (BLACKBURN; RIJKE; VENEMA, 2001).

2.3 Redes de Petri

Inicialmente, Carl Adam Petri (PETRI, 1962) postulou as redes de Petri para a comunicação com autômatos. Formalmente esta foi a primeira publicação em sua tese de doutorado. De fato Petri criou a sua rede em 1939, aos 13 anos, para representar processos químicos. A partir deste primeiro trabalho foram desenvolvidas as teorias que constituem formalmente, uma ferramenta gráfica e matemática para a modelagem de diversos tipos de sistemas voltados para análise e controle de sistemas e eventos discretos, suportando atividades paralelas, concorrentes e assíncronas (SILVA, 2012). Especificar, analisar o comportamento lógico e avaliar o desempenho de sistemas são as principais motivações para o uso das redes de Petri, neste trabalho.

Definição 10 (Redes de Petri). Uma Rede de Petri é uma tripla $\mathcal{P} = \langle P, T, L \rangle$ em que P (do inglês *Places*) é um conjunto finito de lugares, T (do inglês *Transitions*) é um conjunto finito de transições com $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$ e L é uma função que define arestas direcionadas entre lugares e transições atribuindo um valor $w \in \mathbb{N}$ que representa um peso multiplicativo para a transição, dada por $L : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$.

Figura 2.4 – Exemplo de uma rede de Petri com três marcações



Fonte: O próprio autor.

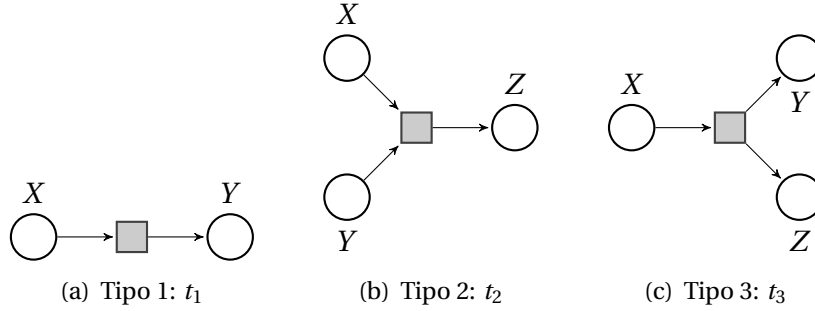
2.3.1 Redes Marcadas

Definição 11 (Redes de Petri Marcadas). As redes de Petri Marcadas são uma quadrupla $P = \langle P, T, L, M_0 \rangle$ como na Definição 10, exceto por acrescentar M_0 como uma marcação inicial (a quantidade de fichas, marcas, distribuída em cada local). Um exemplo de uma rede de Petri válida está na Figura 2.4.

O fluxo de uma rede de Petri é definido por uma relação $F = \{(x, y) \mid L(x, y) > 0\}$. Seja $s \in P$ e $t \in T$. O *preset* de t , denominado *t , é definido como ${}^*t = \{s \in P : (s, t) \in F\}$. O *postset* de t , denominado t^* , é definido como $t^* = \{s \in P : (t, s) \in F\}$. O *preset* de s , denominado

$\bullet s$, é definido com $\bullet s = \{t \in T : (t, s) \in F\}$; o *postset* de s , denominado $s^\bullet$, é definido com $s^\bullet = \{t \in T : (s, t) \in F\}$.

Figura 2.5 – Redes de Petri Básicas



Fonte: O próprio autor.

Definição 12 (Transição Habilitada). Dada uma marcação M de uma rede de Petri, dizemos que uma transição t está habilitada em M se e somente se $\forall x \in \bullet t, M(x) \geq 1$. Uma nova marcação gerada por uma transição habilitada é definida como:

$$M_{i+1}(x) = \begin{cases} M_i(x) - 1, & \forall x \in \bullet t \setminus t^\bullet \\ M_i(x) + 1, & \forall x \in t^\bullet \setminus \bullet t \\ M_i(x), & \forall x \notin \{\bullet t \setminus t^\bullet\} \cup \{t^\bullet \setminus \bullet t\} \end{cases}$$

Conforme a Definição 12, uma transição t está habilitada se cada um dos lugares de entrada contém, pelo menos o número de fichas igual ao valor de peso de t . Desta forma, para redes de Petri, t está habilitada se cada um dos lugares de entrada possui ao menos uma marca.

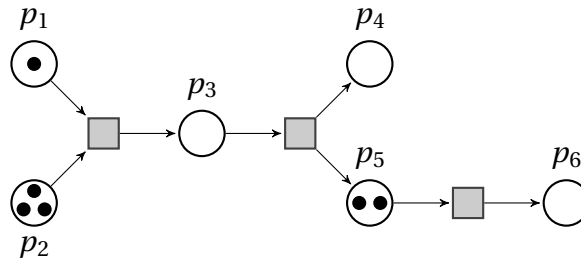
As redes de Petri possuem normalmente várias fichas no caso geral, não havendo nenhuma razão para se ter apenas uma transição viável por vez. Várias transições podem ser ativadas simultaneamente, dependendo da distribuição das fichas. Quando várias transições disparam ao mesmo tempo (disparo em paralelo) denominamos como "semântica de paralelismo".

O comportamento para um programa é descrito pelo conjunto $M = \{M_0, \dots, M_n\}$ em uma rede de Petri marcada. Para representar graficamente uma rede de Petri, usamos um círculo para representar cada $s \in P$, um retângulo para representar cada $t \in T$ e suas relações definidas por L como arestas entre lugares e transições. A quantidade de marcas M é representada como círculos cheios nos lugares correspondentes.

Este trabalho utiliza o formalismo definido para redes de Petri no trabalho de Almeida e Haeusler (1999). Conforme a noção de equivalência apresentada por eles, é possível demonstrar a formulação de redes de Petri válidas, através da composição de três tipos denominados básicos, o que facilita sua associação com a Lógica Dinâmica.

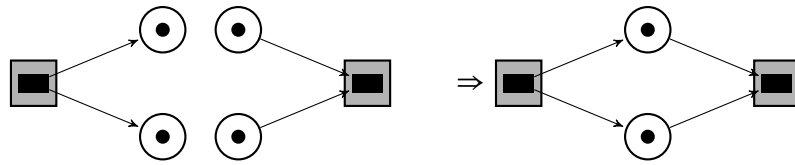
As composições mais complexas são feitas com estes tipos básicos a partir de um processo de colagem que se resume a adaptar a rede de Petri a partir dos modelos de composições descritos a seguir (ALMEIDA; HAEUSLER, 1999). Para exemplificar, a partir dos exemplos mostrados na Figura 2.5, é possível montar a composição da Figura 2.6.

Figura 2.6 – Exemplo de composição utilizando os três tipos básicos

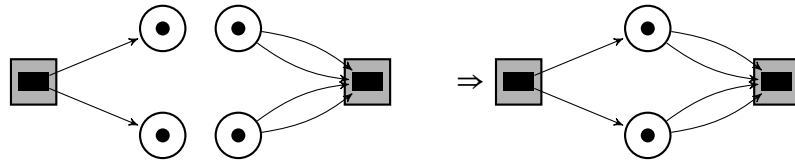


Fonte: O próprio autor.

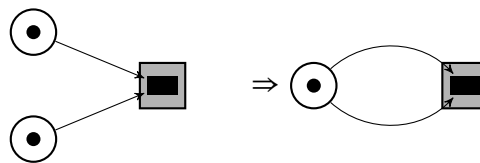
Figura 2.7 – Exemplos de operações de sequência e junção



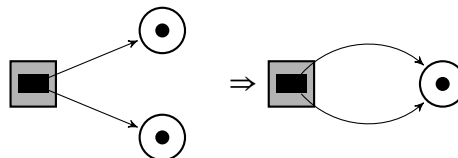
(a) Sequência: Caso 1



(b) Sequência: Caso 2



(c) Joint-L



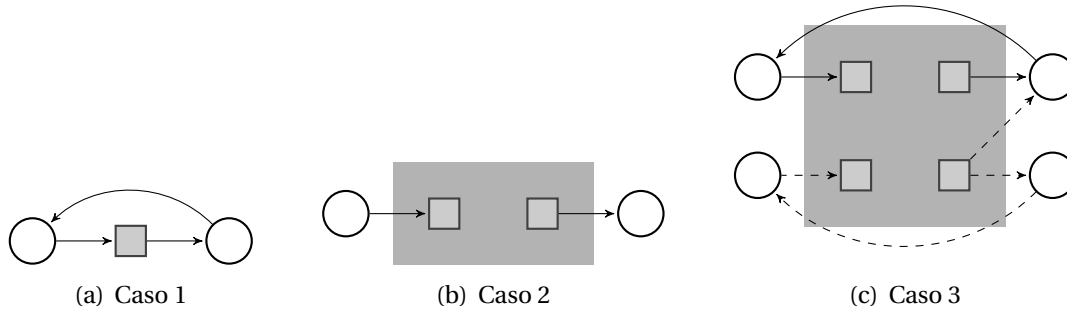
(d) Joint-R

Fonte: O próprio autor.

As operações descritas são, respectivamente: Conflito à Esquerda (*Conlict-L*) Figura 2.9(a); Conflito à Direita (*Conlict-R*) Figura 2.9(b); Casos de Sequência Figura 2.7 (a, b); Jun-

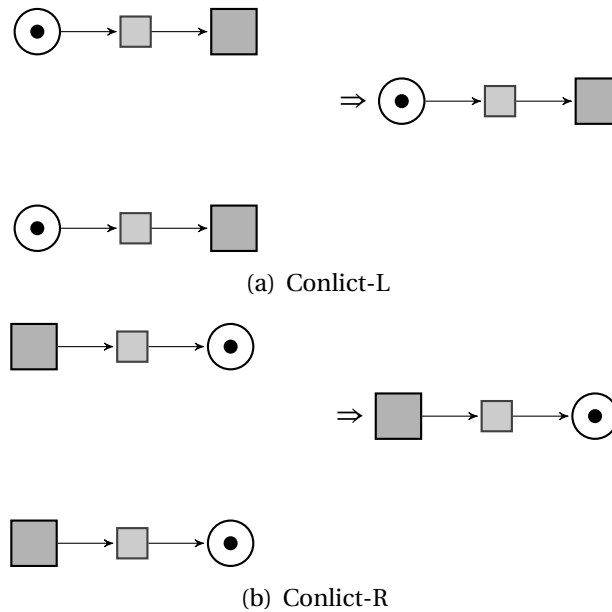
ção a Esquerda (*Joint-L*) Figura 2.7 (c); Junção a Direita (*Joint-R*) Figura 2.7 (d); e os casos de Repetição Figura 2.8 (a, b, c).

Figura 2.8 – Exemplos de três casos de repetições.



Fonte: O próprio autor.

Figura 2.9 – Exemplo de operações de conflito.



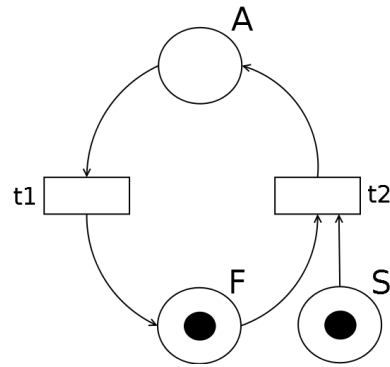
Fonte: O próprio autor.

Nota: Conforme definição, as caixas pretas sempre representam qualquer sub-rede válida de uma rede de Petri.

Um exemplo prático da utilização das redes de Petri para modelar ambientes, pode ser visto na Figura 2.10. O modelo representa um sistema para abertura da fechadura eletrônica de uma porta com senha. Uma ficha em A representa que a porta está aberta e uma ficha em F representa que a porta está fechada. Para o processo de abertura da porta é necessário que a mesma esteja fechada (ficha no lugar F) e que seja utilizada uma senha para abertura (ficha no lugar S). Quando t_2 dispara a porta é aberta e uma ficha é colocada em A , indicando que a porta foi aberta. O contrário acontece quando uma ficha está no lugar A e

a porta pode ser fechada. Ao disparar T_1 , uma ficha é colocada em F indicando que a porta está fechada.

Figura 2.10 – Rede de Petri para um sistema de abertura de porta com senha.



Fonte: O próprio autor.

Capítulo 3

Revisão Sistemática

A maneira como os dados são analisados e modelados são imprescindíveis no processo de planejamento e controle da produção em manufaturas flexíveis. Existem hoje várias abordagens para modelar FMS, tais como: redes de filas (UZAM; GELEN; SALEH, 2016), análise de perturbações (BHUIYAN; GOUW; YAZDI, 2011), modelos de otimização (HUANG; ZHOU; ZHANG, 2015), heurísticas (LEE; LEE, 2010).

O método formal como as redes de Petri modelam plantas de produção FMS são as mais adequadas a serem associadas ao processo de gerenciamento e controle de informações (BHUIYAN; GOUW; YAZDI, 2011). Este capítulo apresenta uma revisão sistemática de estudos que modelam e formalizam ambientes de produção flexível, com foco no uso de redes de Petri como ferramenta para modelagem destes ambientes. O uso da Lógica Dinâmica associada ao mecanismo de controle da modelagem será avaliado como possível solução na verificação destes dados que por muitas vezes são aproveitados em várias etapas do processo.

3.1 Metodologia da Pesquisa Sistemática

Para verificar as principais abordagens existentes hoje na academia em relação a modelagem de ambientes, validação de informações, tipos de resolução de problemas e as variadas formas que se projeta as linhas de produção flexível, foi feito uma revisão do tema deste trabalho nas bases indexadas na forma de um mapeamento sistêmico (do inglês, *Mapping Study*).

Com a aplicação do Mapeamento Sistemático na elaboração da revisão bibliográfica é possível a identificação das principais lacunas para o desenvolvimento de novas pesquisas. De posse das informações encontradas pelos autores da área em suas principais publicações ligadas ao estudo, são traçadas as linhas de raciocínio para este trabalho.

Este tipo de revisão compõe a bibliografia da pesquisa a partir de base de dados online

montando o estado da arte do tema pesquisado. As principais bases buscadas nesta revisão foram: CAPES (2016); *Scopus* (2016) e *Science Direct* (2016), além de outras bases com resultados muito similares que não precisaram ser descritos para relacionar a abrangência aos termos pesquisados.

A metodologia desta pesquisa baseia-se no trabalho de [Petersen et al. \(2008\)](#) buscando os assuntos relacionados à pesquisa com expressões-chaves e por meio de operadores lógicos, a fim de se encontrar a seleção dos artigos que montem o viés da pesquisa iniciada. Para se definir as palavras-chave da pesquisa divide-se a busca em duas vertentes. Em primeiro lugar, encontrar quais tipos de modelagens envolvendo as redes de Petri já são utilizadas para projetar a implementação de ambientes flexíveis de produção industrial. A expressão adotada foi “Petri net” and “FMS”, de forma que ao associar os dois conceitos é possível avaliar nos dados retornados pela base de dados, qual a relação entre os conceitos e como sua relação se comportou ao longo dos anos em que ocorreram estudos com pesquisa voltada à modelagem utilizando as redes de Petri.

Em um segundo momento o foco da pesquisa busca encontrar os trabalhos relacionados com a ferramenta que se deseja associar a este trabalho, como forma de validar as informações presentes na modelagem de ambientes de produção dinâmicos/flexíveis. Usamos a palavra chave “Petri-PDL”, para identificar os trabalhos que já foram submetidos a esta ferramenta e verificar a possibilidade de se adequar ao cenário de produção flexível. Como as pesquisas envolvendo a ferramenta Petri-PDL são relativamente recentes, dada sua concepção, não foi necessário adicionar outra palavra à expressão-chave sendo esta bastante restritiva.

Tabela 3.1 – Dados iniciais da pesquisa sistemática.

Resultados da pesquisa realizada em periódicos de acordo com o tema				
Base de Dados	Expressões-chave			
	"Petri nets" and "FMS"		"Petri-PDL"	
	Artigos			
	inicial	com filtro ¹	inicial	com filtro ¹
Periódicos Capes	215	49	62	21
Scopus	668	37	15	7
Science Direct	36	31	442	48
Total	919	117	519	76
Número total de artigos sem repetições entre as bases de dados	64		42	
Número de artigos selecionados após leitura do título e resumo	27		8	

Fonte: Produzido pelo autor.

Nota: com filtro¹ - filtros aplicados na pesquisa: idioma (inglês); sem restrição do tipo de publicação (artigo de periódico revisado por pares); cronologia das publicações (de 1980 a 2016). Foram descartados periódicos de áreas biomédicas.

É importante ressaltar que foram testadas outras expressões-chaves nesta pesquisa, e que os resultados apresentados foram variados tanto com a sigla apresentada quanto com a escrita sem abreviação. Contudo, foi considerado apenas estes dois segmentos que se mostraram minimamente correspondentes ao ambiente de pesquisa desejado para este trabalho, como será visto nos resultados desta revisão.

Após a pesquisa com as palavras-chaves os resultados obtidos foram exportados e organizados em uma planilha a fim de excluir as repetições e selecionar a aderência dos artigos selecionados a partir do título e resumo inicialmente. Após esta etapa com os artigos selecionados é possível interpretar os dados colhidos, proporcionar e discutir os resultados.

3.2 Discussões e Resultados da Pesquisa Sistemática

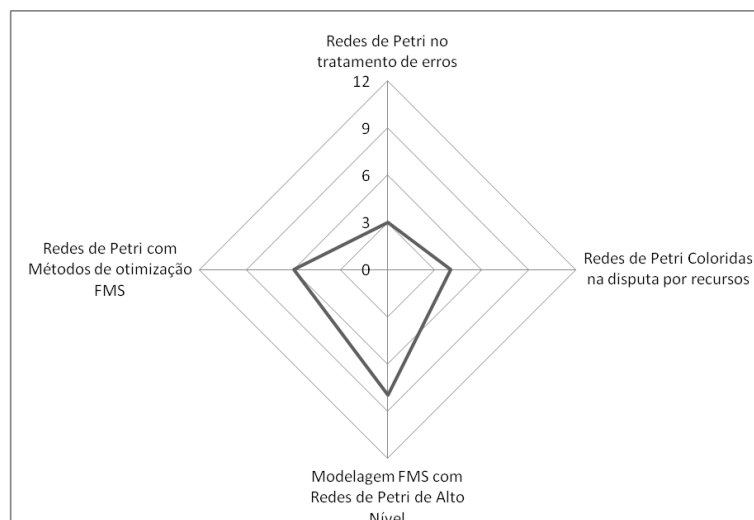
Os resultados encontrados, avaliados para esta pesquisa, estão descritos na Tabela 3.1. Como se pode observar, grande parte das bases de pesquisas de periódicos online são indexadas, inclusive as utilizadas nesta pesquisa. Logo muitas repetições precisaram ser descartadas ao fim da pesquisa resultando 35 artigos para a leitura com resultado correspondente.

Como eixos de pesquisa mais relevantes encontrados após a leitura dos artigos selecionados, cabe dividir a maioria dos trabalhos em dois tópicos principais. Entre eles se destacam: (a) “Modelagem FMS com redes de Petri” e (b) “Petri-PDL em cenários de concorrência”, definindo critérios para o desenvolvimento desta pesquisa.

Em relação ao tópico *Modelagem FMS com redes de Petri*, vários estudos foram encontrados e estão localizados principalmente no hemisfério oriental com a China, Taiwan e Coreia do Sul, e países como Turquia e Brasil. Apesar de serem encontrados poucos trabalhos na América Latina este é um estudo que tem ganhado força principalmente no Brasil com uma produção crescente a partir do ano de 2013, mesmo que não seja encontrado nenhum polo de pesquisa ou algum dos principais pesquisadores da área a nível nacional.

Em contrapartida no tópico *Petri-PDL em cenários de concorrência*, o Brasil se destaca como o maior pesquisador desta ferramenta, de acordo com esta pesquisa, sendo considerada uma tecnologia nova que tem sido apresentada em várias pesquisas da área. Vale ressaltar que um grupo de pesquisa da UFF (Universidade Federal Fluminense) possui a maioria dos trabalhos publicados na área, como pode ser visto na Tabela 3.2, os principais autores e o ano de suas publicações. Os jornais com artigos desta área que mais possuem publicações são o *Journal of Logic and Algebraic Programming* e o *Electronic Notes in Theoretical Computer Science*. A concorrência, ou disputa por recursos, ocorre quando há um paralelismo na busca por materiais e ferramentas em um ambiente, o que pode gerar erros de disputa com impasses conhecidos como *Deadlocks*. Normalmente pode ser causada por modelagens errada ou má elaborada.

Figura 3.1 – Principais vertentes encontradas para o tópico (a).



Fonte: O autor.

Tabela 3.2 – Principais autores com publicações no tópico (b).

Autores e o ano de suas publicações		
Bruno Lopes Vieira	6	2013, 2014(3) e 2015(2)
Mário Benevides	3	2013, 2014(2)
Edward Hermann Haeusler	4	2013(2), 2015(2)

Fonte: Produzido pelo autor.

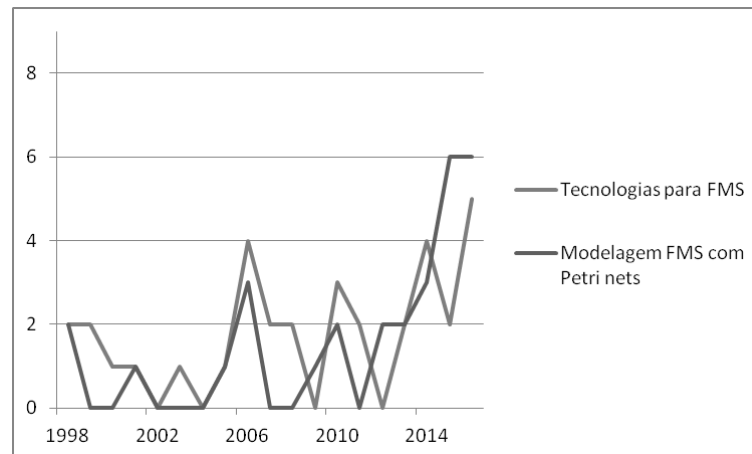
O sistema de produção flexível (FMS) originado na Grã-Bretanha oportunamente foi elaborado na mesma década em que surgiram as redes de Petri por volta dos anos sessenta (GROOVER, 2010, p.935). Logo, as abordagens mais conhecidas para estes modelos são normalmente relacionadas com o uso das redes de Petri para definir e especificar seus padrões.

Na Figura 3.2 o desenvolvimento da pesquisa pelos últimos anos e a relação entre o emprego de técnicas para formalização de moldes de produção está demonstrado no gráfico de linhas. O uso de técnicas de redes de Petri na modelagem e seu aproveitamento na elaboração de técnicas para implementar o seu controlador de acordo com a necessidade está relacionado em “Tecnologias para FMS”. Percebe-se que as duas linhas de pesquisa destacadas na figura tem desenvolvimento sincronizado e dependem entre si, pois a modelagem com redes de Petri nas abordagens vistas para FMS são também utilizadas em outras aplicações que sugerem saídas para o processamento destes dados.

Os trabalhos na modelagem de ambientes de produção alavancaram proporcionalmente à capacidade de conseguir controlar e especificar melhor o escopo de produção, melhorando o seu gerenciamento. A modelagem e o tratamento de dados em uma linha de produção flexível pode apresentar algumas variações de autor para autor nas pesquisas encontradas, porém, algo interessante é que em todas elas a forma como o dado é tratado é

sempre na utilização de alguma abordagem que represente o sequenciamento da produção de forma prática. Isso porque ao programar ambientes assim, é necessário que o entendimento de todos os passos da produção sejam muito bem claros.

Figura 3.2 – Desenvolvimento da pesquisa com redes de Petri para modelagem FMS.



Fonte: O autor.

3.2.1 Métodos Formais na Modelagem com redes de Petri

Inicialmente, Carl Adam Petri postulou as redes de Petri para a comunicação com autômatos. A partir deste primeiro trabalho foram desenvolvidas as teorias que constituem formalmente, uma ferramenta gráfica e matemática para a modelagem de diversos tipos de sistemas voltados para análise e controle de sistemas e eventos discretos, suportando atividades paralelas, concorrentes e assíncronas (PETRI, 1962).

Desde que as redes de Petri foram postuladas, pesquisadores em diferentes campos tem estendido sua aplicação, inclusive para a especificação e análise comportamental lógica, e avaliação do desempenho de sistemas automatizados, aliás, estas são as principais motivações para o uso das redes de Petri (SILVA, 2012).

Conforme Huang e Chang (1992) é importante estabelecer critérios específicos na abordagem FMS. Ademais, é possível ainda, garantir que os dados deste tipo de ambiente são melhor formalizados com as redes de Petri, não importando a variação da aplicação devido sua abordagem gráfica de fácil entendimento. Trabalhos como o de Baruwu e Pira (2015) utilizam as redes de Petri Coloridas como base para a solução de algoritmos de otimização de busca eficiente em ambientes de *scheduling* onde se deseja programar o sequenciamento de tarefas. Esta abordagem se mostrou satisfatória pela abordagem gráfica segmentada por cores, capaz de definir diferentes tratamentos a diferentes tipos de serviços e ferramentas com características próprias.

Yan *et al.* (1998) propõe, o uso de redes de Petri de Alto Nível, um formalismo que estende características relacionadas as marcas e transições, adicionando recursos como tempo

ou cores por exemplo, neste exemplo, as redes são utilizadas no melhoramento da gestão de informação por apresentar formalismo de projeto. Este tipo de modelagem mantém o poder gráfico e analítico das redes de Petri, somado à possibilidade de uma definição detalhada dos tipos de processos e características do produto manufaturado, contudo não há a aproximação destas técnicas com alguma linguagem da lógica, o que faria total diferença se fosse aliado no processo de controle do gerenciador da planta.

Portanto, hoje em dia, o que mais se usa para modelagem destes ambientes são as redes de Petri, por possuir técnicas de especialização de sistemas que possibilitam uma representação matemática e mecanismos de análise poderosa, os quais permitem a verificação de propriedades dos sistemas (MURATA, 1989).

Apesar das qualidades encontradas nas abordagens, o que se percebe é que o tratamento formal avaliado em outros trabalhos não possui mecanismos para validação e correção dos dados, objetivando apenas a classificação e gerenciamento dos dados sem validação. Para exemplificar, autores como Capkovic (2015) e Basak e Albayrak (2015) descrevem em seus trabalhos falhas que são ocasionadas quando há um aumento repentino na planta de produção, ou então quando há a necessidade de se alterar o esquema, adicionando novos subprodutos no controle de produção, o que evidencia o que foi dito, não há um tratamento para a validação dos dados posterior a sua modelagem gráfica.

É preciso ponderar que as implicações geradas no processo de modelagem são diretamente proporcionais à dimensão do espaço de trabalho, ou seja, quanto maior a planta de produção, mais difícil será o seu controle. Essa questão abre margem para que se possa aprofundar os estudos de viabilidade do emprego de novas técnicas de modelagem para este ambiente. Isso porque, a evolução de sistemas flexíveis, assim como os sistemas computacionais que os implementam, são conduzidos pela ocorrência de sequenciamentos com precedência, controle de acesso, dentre outras características, com duração instantânea no tempo.

Além do que, diversos paradigmas lógicos podem ser associados ao modelo/especificação em redes de Petri enriquecendo este processo. Dentre as lógicas que podem ser associadas às redes de Petri a maioria das apresentações se mostram incompletas para cenários flexíveis de produção. Petri-PDL (LOPES; BENEVIDES; HAEUSLER, 2014a; LOPES; BENEVIDES; HAEUSLER, 2014b) por sua vez, lidam com sistemas concorrentes beneficiando-se da representação gráfica intuitiva das redes de Petri. A aplicação deste modelo em sistemas de produção flexível pode ser adaptada por apresentar características similares empregadas em outros sistemas, o que permite supor que esta associação pode suprir as demandas que foram apontadas como lacunas para o desenvolvimento de novos projetos nesta área, a saber, verificação formal de sistemas flexíveis modelados em redes de Petri.

Além de definir uma linguagem e semântica para o modelo, um sistema axiomático de verificação de suas propriedades faz parte desta ferramenta, que pode, além de modelar

propriedades das manufaturas flexíveis, avaliar e ajudar no gerenciamento e controle das tarefas.

Ocasionalmente, é perceptível que, na maioria dos estudos, nada ou muito pouco é feito hoje em dia na verificação e validação de modelos da forma que será proposta. A formalização deste processo com as Petri-PDL pretende avaliar e corrigir os dados resultantes do processo de modelagem com redes de Petri utilizando uma linguagem lógica associada. O fato é que, como não existem hoje abordagens do tipo, e várias pesquisas relataram erros no gerenciamento e controle da produção, este se torna o marco inicial para novas pesquisas desta área.

Capítulo 4

Lógica Dinâmica Proposicional para Redes de Petri: Petri-PDL

Neste capítulo é demonstrado associação entre a Lógica Dinâmica Proposicional e a definição de programa para as redes de Petri (Petri-PDL) (LOPES; BENEVIDES; HAEUSLER, 2014a; LOPES; BENEVIDES; HAEUSLER, 2014b; NALON *et al.*, 2015). Além de definir a linguagem e semântica para o modelo, também será apresentado um sistema axiomático.

As Petri-PDL são uma extensão de lógica dinâmica proposicional que, dentre as várias linguagens lógicas, lidam com sistemas concorrentes beneficiando-se da representação gráfica intuitiva das redes de Petri além de ser um sistema formal decidível, completo e correto (do inglês *soundness*, uma terminologia utilizada em lógica).

Buscando restringir um subconjunto para redes de Petri onde é possível obter decidibilidade e completude, será considerado a especificação para redes de Petri da Seção 2.3 para compor qualquer especificação de rede.

4.1 Linguagem e Semântica

Segundo Lopes, Benevides e Haeusler (2014a), Petri-PDL é uma lógica multi-modal em que cada modalidade é um programa π com uma marcação s , isto é, $\langle s, \pi \rangle$. A linguagem Petri-PDL consiste em um conjunto enumerável Φ de símbolos proposicionais e:

Nomes de lugares: $a, b, c, d, \dots$

Nomes de transições: $t_1, t_2, t_3, \dots$, cada transição com um tipo único;

Tipos de Transição: $T_1 : xt_1y$, $T_2 : xy t_2z$ e $T_3 : xt_3yz$

Símbolo de composição de rede de Petri: $\odot$

Sequência de nomes: $S = \{\epsilon, s_1, s_2, \dots\}$ onde ϵ é uma sequência vazia. Utiliza-se a notação $a \in s$ para denotar que o nome a ocorre em s . Considere $\#(s, a)$ como o número de ocorrências do nome a em s . Uma sequência r é dita uma sub-sequência de s , $r \leq s$, quando para qualquer nome a , $a \in r$ implica $a \in s$ e $\#(r, a) \leq \#(s, a)$. Utiliza-se a notação $s < s'$ para denotar que todos os nomes ocorrendo em s também ocorrem em s' .

Definição 13. Programas são definidos como segue:

Programas Básicos: $\pi_b ::= at_1b \mid at_2bc \mid abt_3c$ onde t_i é do tipo $T_i, i = 1, 2, 3$

Programas de rede de Petri: $\pi_{PN} ::= \pi_b \mid \pi_{PN} \odot \pi_{PN}$, denotado por $\eta_1, \eta_2, \dots$

Programas de rede de Petri marcado: $\pi_{MPN} ::= s, \pi_{PN}$

Definição 14. Uma fórmula é definida como:

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle \pi_{MPN} \rangle \varphi.$$

onde $p \in \Phi$

Observação 2. Padrão de abreviações:

$$\perp \equiv \neg\top, \varphi \vee \phi \equiv \neg(\neg\varphi \wedge \neg\phi), \varphi \rightarrow \phi \equiv \neg(\varphi \wedge \neg\phi) \text{ e } [s, \pi]\varphi \equiv \neg\langle s, \pi \rangle \neg\varphi.$$

Para formar a semântica Petri-PDL utilizamos o conceito de estrutura, como de costume na lógica modal, começando com a definição da função de disparo. Abaixo é demonstrado como a marcação de um programa de rede de Petri se comporta após o disparo de uma transição.

Definição 15. O disparo de uma transição é definido pela função $f : S \times \pi_b \rightarrow S$ como segue:

- $f(s, at_1b) = \begin{cases} s_1bs_2, & \text{se } s = s_1as_2 \\ \epsilon & \text{se } a \notin s \end{cases}$
- $f(s, abt_2c) = \begin{cases} s_1cs_2s_3, & \text{se } s = s_1as_2bs_3 \\ \epsilon & \text{se } a \notin s \text{ ou } b \notin s \end{cases}$
- $f(s, at_3bc) = \begin{cases} s_1s_2bc, & \text{se } s = s_1as_2 \\ \epsilon & \text{se } a \notin s \end{cases}$
- $f(\epsilon, \eta) = \epsilon$, para todo programa de rede de Petri η .

A função de disparo só é definida para os programas básicos da rede de Petri. O comportamento de toda a rede de Petri pode ser determinado pela intercalação da aplicação desta função aos seus componentes conforme veremos na Seção 4.2 de Axiomas. As definições a seguir para *frame*, modelo e satisfatibilidade são semelhantes às apresentadas na Seção 2.2.3 para PDL, agora, com uma adaptação para os fundamentos de redes de Petri.

Definição 16. Um *frame* para *Petri-PDL* é uma 3-tupla $\langle W, R_\pi, M \rangle$, onde:

- W é um conjunto não vazio de estados;
- $M : W \rightarrow S$;
- R_{π_b} é uma relação binária sobre W , para cada programa básico π_b , satisfazendo as condições apresentadas. Assume-se $s = M(w)$

- se $f(s, \pi_b) \neq \epsilon$, $w R_{\pi_b} v$ sse $f(s, \pi_b) \in M(v)$
- se $f(s, \pi_b) = \epsilon$, $w R_{\pi_b} v$ sse $w = v$

- Uma relação binária R_η é definida por indução para cada programa de rede de Petri $\eta = \eta_1 \odot \eta_2 \odot \dots \odot \eta_n$ como segue:

- $R_\eta = \{(w, v) \mid \text{para algum } \eta_i, \exists u \text{ de modo que } s_i \in M(u) \text{ e } w R_{\eta_i} u \text{ e } u R_\eta v\}$

Onde $s_i = f(s, \eta_i)$, para todo $1 \leq i \leq n$.

Definição 17. Um modelo para Petri-PDL é um par $\mathcal{M} = \langle \mathcal{F}, V \rangle$, onde $\mathcal{F}$ é um *frame* Petri-PDL e V uma função de avaliação $V : \Phi \rightarrow 2^W$.

A noção semântica de satisfatibilidade para Petri-PDL é definida da seguinte forma:

Definição 18 (Satisfatibilidade). Seja $\mathcal{M} = (\mathcal{F}, V)$ um modelo. A noção de satisfatibilidade de uma fórmula φ em um modelo $\mathcal{M}$ para um estado w , notação $\mathcal{M}, w \Vdash \varphi$, é definida por indução com segue:

- $\mathcal{M}, w \Vdash p$ sse $w \in V(p)$;
- $\mathcal{M}, w \Vdash \top$ sempre;
- $\mathcal{M}, w \Vdash \neg \varphi$ sse $\mathcal{M}, w \not\Vdash \varphi$;
- $\mathcal{M}, w \Vdash \varphi_1 \wedge \varphi_2$ sse $\mathcal{M}, w \Vdash \varphi_1$ e $\mathcal{M}, w \Vdash \varphi_2$;
- $\mathcal{M}, w \Vdash \langle s, \eta \rangle \varphi$ se existe $v \in W$, $w R_\eta v$, $s \in M(w)$ e $\mathcal{M}, v \Vdash \varphi$.

Se $\mathcal{M}, v \Vdash A$ para todo estado v , então A é considerado válido para o modelo $\mathcal{M}$, notação $\mathcal{M} \Vdash A$. E se A é válido em todo $\mathcal{M}$, então A é considerado válido, notação $\Vdash A$.

4.2 Sistema Axiomático

O sistema de validação das proposições formadas em Petri-PDL, são formados por um conjunto de axiomas onde: p e q são símbolos proposicionais como definidos na linguagem; φ e ψ são fórmulas; $\eta = \eta_1 \odot \eta_2 \odot \dots \odot \eta_n$ são definições de programas de redes de Petri; e por fim, π é um programa rede de Petri marcado.

(PL) Tautologias da Lógica Proposicional suficientes

(K) $[s, \eta](p \rightarrow q) \rightarrow ([s, \eta]p \rightarrow [s, \eta]q)$

(Du) $[s, \eta]p \leftrightarrow \neg \langle s, \eta \rangle \neg p$

(PC) $\langle s, \eta \rangle \varphi \leftrightarrow \langle s, \eta_1 \rangle \langle s_1, \eta \rangle \varphi \vee \langle s, \eta_2 \rangle \langle s_2, \eta \rangle \varphi \vee \dots \vee \langle s, \eta_n \rangle \langle s_n, \eta \rangle \varphi,$

onde $s_i = f(s, \eta_i)$, para todo $1 \leq i \leq n$

(R_ε) $\langle s, \eta \rangle \varphi \leftrightarrow \varphi$, se $f(s, \eta) = \epsilon$

(Sub) Se $\Vdash \varphi$, então $\Vdash \varphi^\sigma$, onde σ uniformemente substitui símbolos proposicionais por fórmulas arbitrárias.

(MP) Se $\Vdash \varphi$ e $\Vdash \varphi \rightarrow \psi$, então $\Vdash \psi$.

(Gen) Se $\Vdash \varphi$, então $\Vdash [\eta]\varphi$.

Os axiomas (PL), (K) e (Du), juntamente com as regras de inferência (Sub), (MP) e (GEN), correspondem à axiomatização usual para as lógicas modais normais. O axioma (PC) expressa como ocorre a intercalação de disparos em um programa de rede de Petri e (R_ε) expressa o comportamento quando nenhuma transição é habilitada em uma rede de Petri.

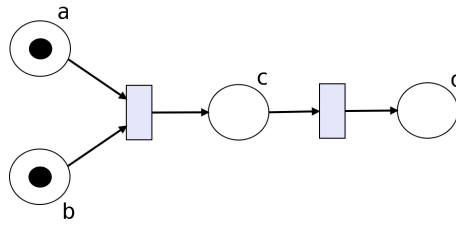
Petri-PDL provou ser um sistema correto, do inglês *sound*, completo e decidível em relação a sua semântica (LOPES; BENEVIDES; HAEUSLER, 2014a; LOPES; BENEVIDES; HAEUSLER, 2014b).

4.3 Algumas Aplicações

A partir de um exemplo básico, será explorada nesta seção a abordagem lógica da ferramenta Petri-PDL.

Exemplo 1. É possível demonstrar que a partir da configuração mostrada na Figura 4.1, após o disparo de transições o estado d é alcançado. Esta análise é expressa pela fórmula $\langle (a, b), (abt_2c \odot ct_1d)^* \rangle \top \rightarrow \langle (d), (abt_2c \odot ct_1d)^* \rangle \top$.

Figura 4.1 – Uma rede de Petri com duas marcações.



Fonte: O autor.

Assumindo a rede da figura como um processo industrial que tem a entrada de duas peças, alocadas nos estados "*a*" e "*b*". O processo industrial é composto de duas fases, uma solda e um processo de finalização da peça. De maneira intuitiva observamos que a falta de uma peça em "*a*" ou "*b*" impossibilita o disparo de transições, tornando impossível atingir o estado final "*d*". Modelando esta situação em Petri-PDL temos, $\pi = abt_2c \odot ct_1d$. Considerando $p \in \phi$ um símbolo proposicional em que p representa que a peça foi finalizada no processo de fabricação (ou seja, o estado *d* foi atingido). A verificação de que o processo atinge seu objetivo e a peça é finalizada é equivalente a verificar a fórmula $\langle (a, b), \pi \rangle p$. Para provar esta propriedade, basta utilizar o sistema de verificação Petri-PDL.

Capítulo 5

Verificação e Validação na Modelagem FMS

O modelo de fabricação competitivo dos sistemas flexíveis (FMS) são, hoje em dia, uma possibilidade para as indústrias de processamento e fabricação de alto valor agregado. A forma prática e eficaz de produzir uma variedade maior de produtos diferentes em pequenos volumes, tornou-se essencial para a concorrência do mercado cada vez mais personalizado com a globalização da economia e o rápido desenvolvimento do comércio internacional.

Para aumentar a flexibilidade de operações pré-estabelecidas, diferentes processos em um FMS podem competir pelo número limitado de recursos compartilhados. Esta competição pode resultar em espera circular de uso de recursos, além da possibilidade de entres no chão de fábrica. Toda e qualquer tempestividade ocorrida no desenvolver da fabricação deve ser prevista e pode ser evitada se o processo for formalizado de forma correta, isto é, modelado, verificado e validado.

A modelagem e verificação de sistemas concorrentes com semântica formal por sua vez, demonstra como é possível afirmar que modelos cumprem com sua especificação. Este processo fornece garantias para o aproveitamento dos dados modelados nas fases seguintes do processo de implementação de uma planta de produção. A configuração de controladores computadorizados, algoritmos de análise de concorrência e prevenção de *deadlocks*, dentre outros processos dependem dos dados modelados.

As redes de Petri, são utilizadas como forma de descrever graficamente os processos da planta industrial, como encontrado na revisão sistêmica, Capítulo 3. Esta aplicação é utilizada em diversos cenários onde a formalização dos dados em forma de rede com estados e transições compõe os estudos sobre a programação de tarefas na indústria. Entre os exemplos encontrados estão modelagens que utilizam, dentre outros tipos, as rede de Petri

coloridas, redes de Petri temporizadas ou de alto nível ¹, etc.

Ademais, a base teórica oferecida pelas redes de Petri contém uma gama extensa de ferramentas para uma interpretação gráfica e intuitiva e sua combinação com a Lógica Dinâmica esclarece como uma abordagem formal traria benefícios às indústrias. Vale ressaltar que as Petri-PDL da forma apresentada por [Lopes, Benevides e Haeusler \(2014a\)](#) é um sistema lógico completo e decidível que atua com as redes de Petri. Além disso, em outras abordagens baseadas em lógica dinâmica, às vezes é necessário traduzir a rede para a sintaxe formal, geralmente sem uma relação direta entre as redes de Petri e a linguagem de destino. Petri-PDL representa redes de Petri de forma natural, sem exigir tal tradução.

5.1 Avaliação de Modelos

A modelagem de linhas de produção tem o objetivo de descrever graficamente os processos aos quais são enumerados no chão de fábrica, descrevendo a sequência, a ordem, as possíveis precedências, dentre outras características necessárias. Quando este ambiente é totalmente automatizado e todas as suas operações são controladas por um computador todos os ambientes de controle devem ser previstos e descritos.

As rede de Petri modelam sequenciamentos através de transições que são habilitadas ao longo do tempo de acordo com o disparo de suas fichas. Ao modelar os sistemas de fabricação, as transições estão relacionadas a eventos como, por exemplo, o início da usinagem, o final da usinagem, o início de um movimento de uma parte da produção, a extremidade do movimento, etc. Os locais são relacionados a estados parciais do sistema como partes prontas para serem processadas, parte processada e parte pronta para ser transferida/finalizada.

Finalizado o processo de modelagem, os dados avaliados são traduzidos na linguagem de máquina mais adequada para os operadores de produção. Nesta fase os dados são remanejados de acordo com a modelagem pré-configurada e assumem variações distintas de acordo com a necessidade.

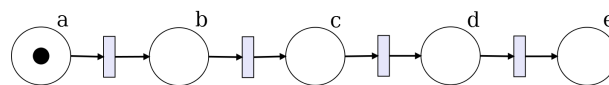
Como foram exaustivamente referenciados, sistemas concorrentes, com disputa por recursos e compartilhamento de dispositivo trazem, corriqueiramente, problemas que precisam ser previstos e evitados. Além da necessidade de modelar o ambiente FMS para implementar as ações do controle de produção, algumas necessidades acessórias também precisam de certo cuidado e planejamento.

De acordo com a literatura, vários problemas restritos a este tipo de produção foram encontrados na revisão sistêmica, inclusive com estudos que indicam soluções através dos dados modelados sobre a rede de Petri que formulou o ambiente. Alguns dos trabalhos listados no Capítulo 3, descrevem estudos com propostas e soluções relacionadas a: simu-

¹ A verificação formal destas redes dependem da sua adaptação à sintaxe formal da lógica a ser associada.

lação e modelagem do escalonamento de processos com redes de Petri de alto nível (YAN *et al.*, 1998); técnicas de otimização para reconfiguração de modelos FMS através das redes de Petri (BOURDEAUD'HUY; TOGUYENI, 2006); modelagem de um sistema de decisão para escalonamento em tempo real baseado em redes de Petri (BASAK; ALBAYRAK, 2015); identificação de padrões de repetição em modelos FMS para um algoritmo de escalonamento de processo baseado em busca eficiente (BARUWA; PIERA, 2015); recuperação de *deadlocks* para sistemas de produção flexível modelados com redes de Petri (CHEN *et al.*, 2017); eliminação de restrições redundantes para a classificação do número mínimo de lugares de controle para a rede de Petri modelada (HUANG; ZHOU; ZHANG, 2015)

Figura 5.1 – Modelo de processo de usinagem.



Fonte: Adaptado de Ribeiro, II e Lepikson (2009).

Nesta seção serão avaliados os impactos que uma abordagem formal por via lógica traria ao modelo, visto que os métodos que foram listados não o fazem.

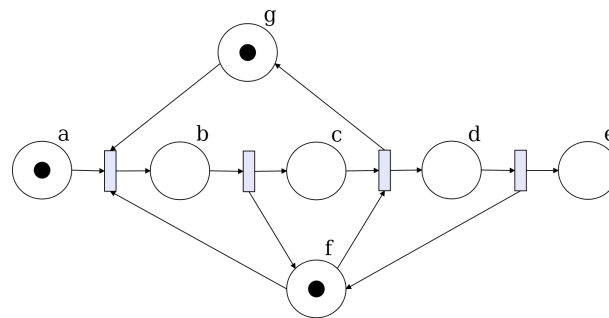
Exemplo 2 (Caso de uso: Célula de Usinagem). O modelo da Figura 5.1 representa uma rede de Petri que modela um processo de usinagem onde a célula de produção recebe um palete contendo a matéria-prima a ser processada. De acordo com o tipo de material, o robô de transporte leva o palete e coloca no *buffer* (área de armazenamento temporário) correspondente. Após liberar o palete, o robô leva a peça e coloca na máquina correspondente. Por fim, ao concluir a usinagem a parte processada é liberada conduzindo o palete para fora da célula. Uma ficha no lugar "a" representa a parte que espera ser processada; uma ficha no lugar "b" representa uma parte transportada para a máquina; uma ficha no lugar "c" representa a parte processada; uma ficha no local "d" representa uma parte removida da máquina e uma ficha no local "e" representa a finalização da peça processada.

Embora o sistema esteja sempre pronto a iniciar a usinagem de uma peça, as transições da rede só disparam quando uma peça chega na célula de manufatura flexível. Considerando que o lugar "a" seja parte de um subsistema que compõe algo maior, a entrada do sistema certamente poderá se originar em outras células ou partes deste sistema. Quando o sistema está restrito a uma lógica de controle de supervisão como o que trata o trabalho de Huang, Zhou e Zhang (2015), o modelo pode ser submetido a restrições para limitar o número de peças processadas simultaneamente (apenas uma, no caso mais simples de uma única máquina de usinagem).

Sendo assim, Ribeiro, II e Lepikson (2009), propôs uma modificação no modelo da Figura 5.1 com um bloqueio de prevenção para o caso de uma peça ser admitida na célula

sem que se termine um outro processo que já tenha sido iniciado. Neste caso, não se pode admitir uma nova parte a ser processada pela máquina, pois a célula já está com outra parte em processo. O modelo controlado da Figura 5.2 tem dois lugares adicionais ("f" e "g"), que representam, respectivamente, quando os recursos do robô e do *buffer* de operações está ocioso.

Figura 5.2 – Modelo de processo de usinagem controlado.



Fonte: Adaptado de Ribeiro, II e Lepikson (2009).

O modelo proposto neste caso de uso apresentou duas abordagens utilizando as redes de Petri: um modelo com as redes de Petri coloridas, que associa a cada transição uma interface correspondente que mascara os dados representando-os com tipos individuais de acordo com a necessidade, podendo ou não se diferenciar pela cor; e um modelo com as redes de Petri orientadas a objetos onde cada transição é definida como uma classe e os disparos das redes são como objetos instanciados. Esta análise diminui o número necessário de informações na rede para que não seja preciso tantas transições com ações visualmente irrelevantes como o começo e o fim de um processo.

Visto que o exemplo adicionou melhorias ao modelo através de um sistema de controle de supervisão e as redes de Petri de alto nível, pode-se analisar seus resultados visto que tais alterações não foram verificadas de forma lógica. As mudanças feitas em qualquer modelo visando otimizar processos devem ser validadas para que se comprove sua eficiência. Tanto os exemplos listados anteriormente quanto neste exemplo que acaba de ser descrito é possível realizar a formalização dos dados por via lógica através das Petri-PDL, como propõe este trabalho. A forma como o modelo é adaptado diz muito sobre o seu comportamento no momento em que a célula flexível entra em produção. Garantir que os dados avaliados vão corresponder a expectativa, e que a produção não irá perecer diante dos obstáculos impostos é uma questão recorrente.

O processo de verificação formal tem início com a associação dos dados modelados graficamente pela rede de Petri, de modo que as transições da rede e suas marcações são escritas na sintaxe da lógica. A ferramenta Petri-PDL apresentada no Capítulo 4 formaliza o processo de modelagem da rede através da Lógica Dinâmica para fundamentar e certificar sistemas.

Logo, um dos principais objetivos deste trabalho se resume a avaliar, por exemplo, se a forma como o Exemplo 2 adaptou o controle de concorrência com a inserção de lugares de controle no modelo da Figura 5.1 é confiável, correto, ou se por ventura, não é possível garantir que os dados modelados correspondem a expectativa do mundo real. Isto pode ser facilmente compreendido com a utilização da Petri-PDL como veremos nos exemplos da Seção 5.2.

Sabe-se que o modelo tratado é relativamente simples e que sua análise poderia ocorrer através da simulação de disparo de transições com algumas hipóteses, por exemplo. Porém o tamanho da rede pode variar de tamanho e esta técnica pode se tornar inviável. Por isso é tão interessante entender o quanto uma abordagem por via lógica, como as Petri-PDL é válida. O fato de Petri-PDL formar um sistema correto e completo garante que, uma vez que alguma sentença lógica é provada, o sistema de axiomas permite inferir que aquela tautologia é livre de erros e que não existem variações que possam levar a afirmações que derrubem a verdade.

5.2 Análise Formal com as Petri-PDL

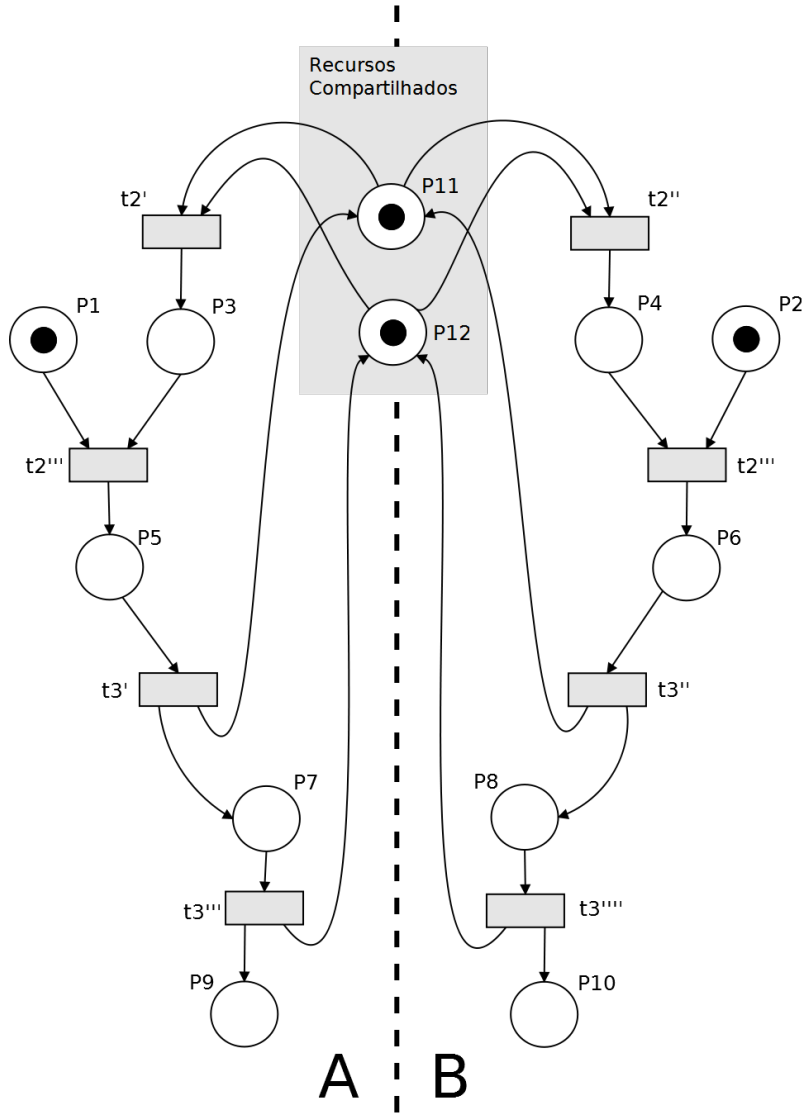
Nesta seção serão apresentados alguns exemplos da aplicação da lógica Petri-PDL em modelagens para os sistemas de produções flexíveis, encontrados na revisão sistêmica deste trabalho.

Exemplo 3. Através do modelo da Figura 4.1 é possível demonstrar que mesmo que haja uma ficha em a , enquanto faltar uma ficha na localização b não é exequível, após o disparo da rede, obter uma configuração em que c possua uma ficha. Isso pode ser expresso pela fórmula $\langle (a), abt_2c \odot ct_1d \rangle \top \rightarrow \neg \langle (c), abt_2c \odot ct_1d \rangle \top$. Esta propriedade pode ser provada através do sistema de provas para Petri-PDL (BRAGA; LOPES, 2016).

A propriedade testada no Exemplo 3 é uma das formas de verificar, com as Petri-PDL, que o modelo para o cenário de produção FMS atende as exigências do ambiente real. Como neste caso em que o modelo tinha a necessidade de garantir recursos presentes em determinados lugares da rede para certificar que os disparos pudessem acontecer.

Outra propriedade possível de ser testada é a maneira que o sistema pode lidar com a falta de resposta. Em ambientes de produção, quando um recurso é compartilhado por fases diferentes do processo de fabricação (robô ou *buffer* da célula de produção compartilhado), são caracterizados cenários com concorrência. Em casos como este, um recurso é compartilhado em *hardware* e *software* formando processos independentes. Esta configuração pode representar fases do processo produtivo onde as partes devem ser processadas separadamente, mesmo que possam futuramente compor um produto único quando o processo for finalizado.

Figura 5.3 – Modelo de processo com recursos compartilhados.



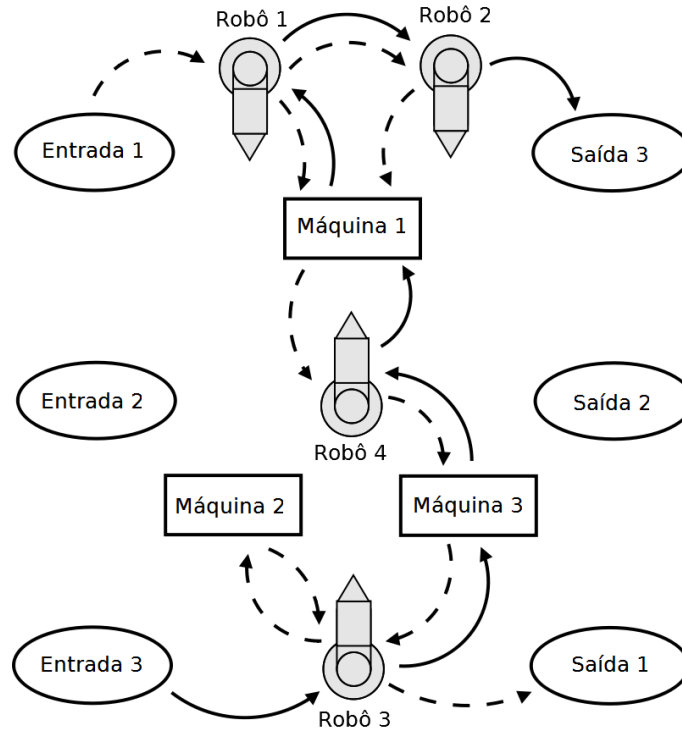
Fonte: Adaptado de Brussel, Peng e Valckenaers (1993).

Exemplo 4. No modelo da Figura 5.3 pode-se modelar Petri-PDL para comprovar a propriedade de lidar com a concorrência. Em um cenário que uma célula de manufatura flexível pode tanto realizar a usinagem de uma peça em "A", quanto realizar a movimentação/montagem de uma peça em "B", os lugares P_{11} e P_{12} descrevem o acesso aos recursos compartilhados do sistema acessados por partes diferentes do processo flexível. Desta forma se t_2'' dispara, então t_2''' será incapaz de disparar. É possível modelar Petri-PDL de forma que, quando o lado "B" da rede é ativado torna-se impossível ativar "A" até que os recursos compartilhados sejam disponibilizados. Desta forma, o modelo para o cenário é descrito, como:

$$\langle (P_1, P_2, P_{11}, P_{12}), P_{11}P_{12}t_2'P_3 \odot P_{11}P_{12}t_2''P_4 \odot P_1P_3t_2'''P_5 \odot P_4P_2t_2''''P_6 \odot P_5t_3'P_{11}P_7 \odot P_6t_3''P_{11}P_8 \odot P_7t_3'''P_{12}P_9 \odot P_8t_3''''P_{12}P_{10} \rangle \top \rightarrow \neg \langle (P_1, P_2, P_{11}, P_{12}), P_1P_3t_2'''P_5 \rangle \langle \epsilon, P_{11}P_{12}t_2'P_3 \odot P_{11}P_{12}t_2''P_4 \odot P_1P_3t_2'''P_5 \odot P_4P_2t_2''''P_6 \odot P_5t_3'P_{11}P_7 \odot P_6t_3''P_{11}P_8 \odot P_7t_3'''P_{12}P_9 \odot P_8t_3''''P_{12}P_{10} \rangle \top$$

Como t_2' é capaz de disparar, é certo que o lado esquerdo da implicação é verdadeiro pela noção semântica de satisfatibilidade [18] de Petri-PDL. Porém, para o lado direito da implicação, t_2''' não é capaz de disparar, portanto, não pode ser verdade, daí a negação é verdadeira e a fórmula é válida. Na segunda modalidade do lado direito da implicação, o símbolo ϵ corresponde ao resultado do disparo da função $f(P_1P_2P_{11}P_{12}, P_1P_3t_2'''P_5)$.

Figura 5.4 – Sistema de produção flexível com recursos compartilhados e duas seqüências de tarefas.



Fonte: Adaptado de Uzam, Gelen e Saleh (2016).

A Figura 5.4 reproduz um ambiente de produção flexível com robôs de movimentação/montagem e máquinas compartilhadas. Nesta representação pode-se observar duas rotas, ilustradas através das setas, que simbolizam o passo a passo da manufatura de dois produtos para a mesma configuração de máquinas. As atividades com início na *Entrada 1*, com rotas detalhadas pelas setas de linha pontilhada, mostram a produção de uma primeira manufatura enquanto as atividades com início na *Entrada 3*, com rotas detalhadas pelas setas de linha contínua, mostram o encadeamento de processos para uma segunda manufatura, sendo executada em paralelo nesta configuração. Percebe-se que a célula flexível ainda poderia admitir mais uma manufatura entre os pontos *Entrada 2* e *Saída 2*.

Como foi visto na Seção 2.1, células de manufatura flexível podem se adaptar, conforme a necessidade, para conseguir produzir diferentes produtos ao mesmo tempo. Logo, em uma mesma configuração, produtos relativamente diferentes podem ser processados em paralelo, desde que sejam modelados respeitando a disponibilidade de recursos e a ordem de produção de cada um deles. No entanto, a adaptação flexível dos modelos se torna cor-

riqueiramente um problema, quando a produção é prejudicada por indisponibilidade e/ou concorrência de máquinas ou recursos. Por isso, a análise do ambiente de produção da Figura 5.4 ajuda a compreender a complexidade do modelo da Figura 5.5 que será avaliado no Exemplo 5.

Várias abordagens são encontradas na literatura para a otimização de modelos FMS que apresentem inconsistências como falta ou indisponibilidade de recursos, como foi visto na revisão sistêmica [3]. Uma delas utilizadas no trabalho de Uzam, Gelen e Saleh (2016) é a análise do espaço de estados da rede. Esta abordagem, torna possível analisar a complexidade da rede e tornar visível o conjunto de todos os casos de uso. Este espaço é utilizado na análise das propriedades da rede de Petri. Para análise, monta-se um grafo onde seus vértices representam os conjuntos de estados (casos) válidos em cada instante da execução dos disparos da rede e as arestas do grafo representam as ações (ou disparos) que devem ser executados para que o sistema passe de um caso para outro.

É importante destacar esta técnica porque os autores utilizam um modelo muito conhecido da literatura como caso de uso, logo, avaliá-lo com a ferramenta Petri-PDL pode servir como base para dimensionar seu resultado e eficiência em relação à outras abordagens. Autores como Huang, Zhou e Zhang (2015), Uzam *et al.* (2016), Uzam, Gelen e Saleh (2016), Chen *et al.* (2017), utilizam técnicas a partir de heurísticas e métodos de otimização para a resolução de *deadlocks* com o mesmo modelo de rede de Petri não controlada. Este modelo está representado na Figura 5.5 e será avaliado pela ferramenta Petri-PDL no Exemplo 5.

Exemplo 5. Para mostrar a aplicabilidade da ferramenta Petri-PDL nesta abordagem será considerado o cenário descrito pelo modelo da Figura 5.5. A figura mostra uma rede de Petri com onze locais de atividades $A_1 = \{p_2, p_{12}, p_{13}, p_3, p_{14}, p_4\}$ e $A_2 = \{p_6, p_{15}, p_7, p_{16}, p_8\}$, três locais de recursos compartilhados $R = \{p_9, p_{10}, p_{11}\}$ e dois coletores/fonte de peças $P = \{p_1, p_5\}$. As atividades em A_1 e A_2 pertencem a processos de manufaturas diferentes executadas em paralelo na célula flexível. Este é um modelo propenso a *deadlocks*, pois em sua modelagem não foi considerado o impacto da execução de duas manufaturas para esta quantidade de recursos compartilhados, tampouco foram adicionados ao modelo locais de controle do uso de recursos como máquina e robô (UZAM; GELEN; SALEH, 2016). Para a verificação e validação destas propriedades o modelo será formalizado.

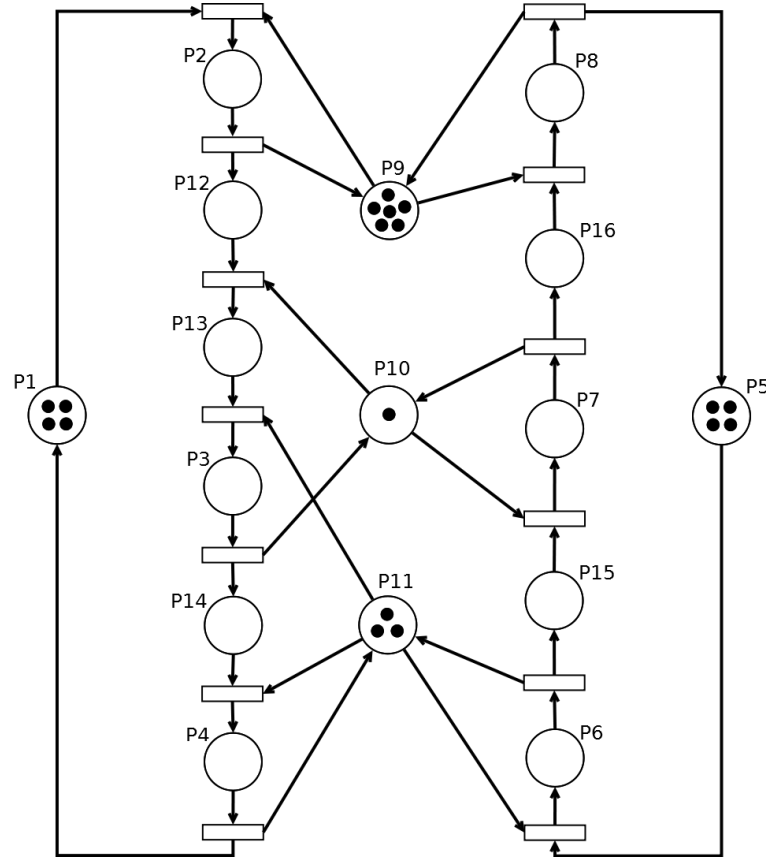
Inicialmente, a rede de Petri pode ser escrita em Petri-PDL como: $\langle (p_1 p_1 p_1 p_1, p_5 p_5 p_5 p_5, p_9 p_9 p_9 p_9 p_9 p_9, p_{10}, p_{11} p_{11} p_{11}), p_1 p_9 t_2 p_2 \odot p_2 t_3 p_{12} p_9 \odot p_{12} p_{10} t_2 p_{13} \odot p_{11} p_{13} t_2 p_3 \odot p_3 t_3 p_{10} p_{14} \odot p_{11} p_{14} t_2 p_4 \odot p_4 t_2 p_1 p_{11} \odot p_5 p_{11} t_2 p_6 \odot p_6 t_3 p_{11} p_{15} \odot p_{15} p_{10} t_2 p_7 \odot p_7 t_3 p_{10} p_{16} \odot p_{16} p_9 t_2 p_8 \odot p_8 t_3 p_9 p_5 \rangle \varphi$

É possível afirmar que após os disparos todas as peças representadas pelas fichas em p_1 e p_5 serão processadas com esta disponibilidade de recursos compartilhados?

Assumindo que fichas no lugar p_1 e p_5 indicam peças a serem processadas. A partir da fórmula inicial, quando sua execução parar φ será verdadeiro em um estado futuro. Supondo que a fórmula inicial seja verdadeira, a transição $p_1 p_9 t_2 p_2$ e $p_5 p_{11} t_2 p_6$ são ativadas paralelamente, logo $\langle (p_1 p_1 p_1 p_1, p_5 p_5 p_5 p_5, p_9 p_9 p_9 p_9 p_9, p_{10}, p_{11} p_{11} p_{11}), p_1 p_9 t_2 p_2 \odot p_5 p_{11} t_2 p_6 \rangle \langle (p_1 p_1 p_1, p_2, p_5 p_5 p_5, p_6, p_9 p_9 p_9 p_9 p_9, p_{10}, p_{11} p_{11}), p_1 p_9 t_2 p_2 \odot \psi \rangle \varphi$ é verdadeiro; então $p_1 p_9 t_2 p_2$ e $p_5 p_{11} t_2 p_6$ disparam e $\langle (p_1 p_1 p_1, p_2, p_5 p_5 p_5, p_6, p_9 p_9 p_9 p_9 p_9, p_{10}, p_{11} p_{11}), p_1 p_9 t_2 p_2 \odot \psi \rangle \varphi$ é verdadeiro.

Nota: ψ é uma abreviação para a fórmula, representando todas as demais transições da rede, conforme a fórmula inicial.

Figura 5.5 – Rede de Petri para um modelo de produção com duas tarefas.



Fonte: Adaptado de Uzam, Gelen e Saleh (2016).

Após alguns disparos é possível avaliar que a presença de apenas uma ficha no lugar p_{10} pode indicar uma indisponibilidade e acarretar disputa por recursos, levando o modelo a erros do tipo *deadlocks*. Desta forma quando a rede assume fichas em p_{15} e p_{12} duas transições podem ser habilitadas, porém existe um único recurso em p_{10} . Desta forma se $p_{12} p_{10} t_2 p_{13}$ dispara, então $p_{15} p_{10} t_2 p_7$ será incapaz de disparar até que p_{10} volte a obter uma ficha. Uma das forma de responder ao questionamento do exemplo é modelando esta situação em Petri-PDL.

Consequentemente, este cenário é descrito pela fórmula $\langle (p_1 p_1, p_2, p_5 p_5, p_6, p_9 p_9 p_9 p_9 p_9, p_{13}, p_{15}, p_{11} p_{11}), p_1 p_9 t_2 p_2 \odot p_2 t_3 p_{12} p_9 \odot p_{12} p_{10} t_2 p_{13} \odot p_{11} p_{13} t_2 p_3 \odot p_3 t_3 p_{10} p_{14} \odot p_{11} p_{14} t_2 p_4 \odot p_4 t_2 p_1 p_{11} \odot p_5 p_{11} t_2 p_6 \odot p_6 t_3 p_{11} p_{15} \odot p_{15} p_{10} t_2 p_7 \odot p_7 t_3 p_{10} p_{16} \odot p_{16} p_9 t_2 p_8 \odot p_8 t_3 p_9 p_5 \rangle \top \rightarrow \neg \langle (p_1 p_1, p_2, p_5 p_5, p_6, p_9 p_9 p_9 p_9 p_9, p_{13}, p_{15}, p_{11} p_{11}), p_{15} p_{10} t_2 p_7 \rangle \langle \epsilon, p_1 p_9 t_2 p_2 \odot p_2 t_3 p_{12} p_9 \odot p_{12} p_{10} t_2 p_{13} \odot p_{11} p_{13} t_2 p_3 \odot p_3 t_3 p_{10} p_{14} \odot p_{11} p_{14} t_2 p_4 \odot p_4 t_2 p_1 p_{11} \odot p_5 p_{11} t_2 p_6 \odot p_6 t_3 p_{11} p_{15} \odot p_{15} p_{10} t_2 p_7 \odot p_7 t_3 p_{10} p_{16} \odot p_{16} p_9 t_2 p_8 \odot p_8 t_3 p_9 p_5 \rangle \top$.

Como o disparo de $p_{10} p_{12} t_2 p_{13}$ era possível, o lado esquerdo da implicação é verdadeiro, como pode ser verificado através das regras de disparo da noção semântica de Petri-PDL [18]. Com isso, após o disparo, a falta de fichas em p_{10} inviabiliza o disparo de $p_{15} p_{10} t_2 p_7$. Portanto, o lado direito da implicação não pode ser verdade, daí a negação é verdadeira e a fórmula é válida.

Uma forma de acelerar a verificação de modelos a partir da validação com Petri-PDL, seria a utilização de verificadores automatizados de prova. Braga e Lopes (2016) propôs um modelo de verificador para redes de Petri com base na Lógica Dinâmica Proposicional, formalmente projetado pela aplicação de um mapeamento geral de estruturas Kripke para reescrever teorias para Petri-PDL. Esta automatização iria acelerar o processo de verificação de fórmulas, se tornando mais rápido e prático a validação de propriedades de um modelo a ser verificado, além da possibilidade de retornar um número maior de informações acerca do modelo. A avaliação desta abordagem pode compor, neste sentido, um dos trabalhos futuros apontados por esta pesquisa.

Depois de utilizar Petri-PDL para validar os modelos submetidos a ferramenta, é perceptível que sua abordagem pode ocorrer de diversas formas, isso porque, a forma em que o sistema avalia o que foi modelado se dá a partir das indagações submetidas à ferramenta sobre os dados modelados. A formalização de questões acerca do que foi modelado pela ferramenta Petri-PDL, pode fundamentar assertivas para o comportamento da rede, como: o espaço de estados da rede; a semântica de mundos possíveis para a rede e as possíveis inconsistências, de acordo com o ambiente testado.

Desta forma, através de questionamentos como: a possibilidade de alcançar determinado lugar; a necessidade de atingir uma transição durante os disparos; a quantidade de fichas necessárias para determinado recurso compartilhado; dentre diversas outras características, podem garantir se o modelo é correto, se ele corresponde com a expectativa e principalmente se ele pode ou não ser prejudicado pela disponibilidade ou disputa por recursos em algum momento da execução dos disparos resultando em *deadlocks*. Com isso, fica claro à adaptação do formalismo proposto para a validação de modelos de sistemas de produção flexível.

Capítulo 6

CONCLUSÕES E TRABALHOS FUTUROS

O desenvolvimento desta pesquisa iniciou-se com a necessidade natural de validar que as propostas encontradas na modelagem FMS, uma vez que modeladas com as redes de Petri, representavam a verdade. Logo, as Petri-PDL, proposta por [Lopes, Benevides e Haeusler \(2014a\)](#) para verificação formal de softwares se mostrou uma proposta capaz de verificar propriedades e comportamento de programas e sistemas sequenciais, tais como: correção, imparcialidade e equivalência.

Este trabalho descreveu uma metodologia com a ferramenta Petri-PDL para modelar formalmente e verificar as propriedades do sistema de produção com análise e correção da disputa por recursos, dentre outras características, em plantas flexíveis.

Para os testes foram utilizados modelos encontrados na revisão sistêmica para o tema. Através da pesquisa de artigos revisados nas bases indexadas com acesso na regional Catalão da UFG, foram classificados trabalhos para a formalização de modelos e validação do ambiente representado com as redes de Petri.

Dentre os principais resultados estão: a oportunidade de avaliar e validar os dados modelados para cenários FMS, com um formalismo capaz de dar garantias acerca do que foi modelado; a possibilidade de avaliar ambientes de sistemas multiagente com concorrência de recursos; e por último, a avaliação da rede, quanto a presença de fichas para designar a ordem, a precedência e a necessidade durante a execução da rede de Petri.

A validação do ambiente de produção flexível possibilitou não só a verificação de informações do modelo, como também, permitir que sejam agregados valores aos resultados de outros trabalhos que apresentavam propostas de otimização em ambientes de produção flexível sem uma validação para seus resultados com alguma linguagem lógica.

Como Petri-PDL é um formalismo decidível, correto e completo que herda as vantagens de redes de Petri, seu uso permitiu o uso de uma modelagem gráfica intuitiva onde os

recursos agregados a cada estado foram avaliados conforme sua variação durante a execução do programa, o que se mostrou adequado para modelagem industrial.

O sistema axiomático apresentado neste trabalho pertence ao trabalho de [Lopes, Benevides e Haeusler \(2014a\)](#) e representa um conjunto de regras que define teoria para mostrar verdades a partir da concatenação das regras definidas, demonstrando novos resultados dessa teoria. Todo o sistema é provado e considerado correto e completo para resolução de cenários que se adéquem à sintaxe proposta.

Enfim, a principal contribuição deste trabalho foi comprovar a eficiência da ferramenta em um ambiente novo, como as linhas de produção flexível, representando um importante nicho da indústria. Esta percepção pode servir de base para a aplicação desta ferramenta em cada vez mais escopos, não só da produção industrial, como também de outros modelos que necessitem de verificação formal.

Após verificar formalmente ambientes flexíveis e avaliar o impacto destes resultados com relação aos modelos utilizados como caso de uso, seguem algumas propostas de continuidade da pesquisa:

- Ampliar a aplicação das técnicas a outros ambientes de produção industrial;
- Avaliar o impacto da aplicação destas técnicas também em um ambiente de produção industrial real;
- Propor a utilização destas técnicas com automatizadores de prova para este cenário, de forma que, assim, sua implementação se daria de forma mais rápida. Como referência a esta proposta seria avaliado a eficiência do modelo já proposto por [Braga e Lopes \(2016\)](#)

REFERÊNCIAS

ALETAN, S. The components of a successful cim implementation. *Industrial Engineering, INST INDUSTRIAL ENGINEERS 25 TECHNOLOGY PARK/ATLANTA, NORCROSS, GA 30092*, v. 23, n. 11, p. 20–22, 1991. Citado na página 26.

ALMEIDA, E. de; HAEUSLER, E. Proving properties in ordinary petri nets using lores logical language. *Petri Net Newsletter*, v. 57, p. 23–36, 1999. Cited By 8. Citado 2 vezes nas páginas 34 e 35.

BARUWA, O.; PIERA, M. Identifying fms repetitive patterns for efficient search-based scheduling algorithm: A colored petri net approach. *Journal of Manufacturing Systems*, v. 35, p. 120–135, 2015. Cited By 9. Citado 4 vezes nas páginas 19, 25, 43 e 55.

BASAK, Ö.; ALBAYRAK, Y. Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems. *Computers and Industrial Engineering*, v. 86, p. 116–126, 2015. Cited By 4. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84946084797&doi=10.10162fj.cie.2014.09.024&partnerID=40&md5=049767c2e600d1a377c248ce119ac7d0>>. Citado 3 vezes nas páginas 19, 44 e 55.

BHUIYAN, N.; GOUW, G.; YAZDI, D. Scheduling of a computer integrated manufacturing system: A simulation study. *Journal of Industrial Engineering and Management*, v. 4, n. 4, p. 577–609, 2011. ISSN 2013-0953. Disponível em: <<http://www.jiem.org/index.php/jiem/article/view/331>>. Citado 3 vezes nas páginas 19, 26 e 39.

BLACKBURN, P.; RIJKE de M.; VENEMA, Y. *Modal Logic*. New York, NY, USA: Cambridge University Press, 2001. ISBN 0-521-80200-8. Citado na página 33.

BOURDEAUD'HUY, T.; TOGUYENI, A. A petri-net based approach for the reconfiguration of flexible manufacturing systems using optimization techniques. *IFAC Proceedings Volumes*, v. 39, n. 3, p. 367 – 372, 2006. ISSN 1474-6670. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1474667015359176>>. Citado 3 vezes nas páginas 19, 25 e 55.

BRAGA, C.; LOPES, B. Towards reasoning in dynamic logics with rewriting logic: The petri-pdl case. In: _____. *Formal Methods: Foundations and Applications: 18th Brazilian Symposium, SBMF 2015, Belo Horizonte, Brazil, September 21-22, 2015, Proceedings*. Cham: Springer International Publishing, 2016. p. 74–89. ISBN 978-3-319-29473-5. Disponível em: <https://doi.org/10.1007/978-3-319-29473-5_5>. Citado 3 vezes nas páginas 57, 62 e 64.

BRUSSEL, H. V.; PENG, Y.; VALCKENAERS, P. Modelling flexible manufacturing systems based on petri nets. *{CIRP} Annals - Manufacturing Technology*, v. 42, n. 1, p. 479 – 484, 1993. ISSN 0007-8506. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0007850607624901>>. Citado na página 58.

CAGLIANO, R.; SPINA, G. Advanced manufacturing technologies and strategically flexible production. *Journal of Operations Management*, v. 18, n. 2, p. 169 – 190, 2000. ISSN 0272-6963. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0272696399000224>>. Citado 2 vezes nas páginas 19 e 27.

CAPKOVIC, F. Timed and hybrid petri nets at solving problems of computational intelligence. *Computing and Informatics*, v. 34, n. 4, p. 746–778, 2015. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84953217406&partnerID=40&md5=2a2c2c784f279a807afb8e87d6c85695>>. Citado 2 vezes nas páginas 19 e 44.

CAPUTO, A. *et al.* Computer integrated manufacturing in small companies: a case study. *Industrial Management & Data Systems*, v. 98, n. 3, p. 138–144, 1998. Disponível em: <<http://dx.doi.org/10.1108/02635579810213143>>. Citado na página 26.

CHAN, C.-C.; WANG, H.-P. Design and development of a stochastic high-level petri net system for fms performance evaluation. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, Taylor & Francis, v. 31, n. 10, p. 2415–2439, 1993. Citado 2 vezes nas páginas 19 e 28.

CHEN, Y. *et al.* Deadlock recovery for flexible manufacturing systems modeled with petri nets. *Information Sciences*, v. 381, n. Supplement C, p. 290 – 303, 2017. ISSN 0020-0255. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025516317972>>. Citado 2 vezes nas páginas 55 e 60.

CHU, C.-H. Manufacturing cell formation by competitive learning. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, Taylor & Francis, v. 31, n. 4, p. 829–843, 1993. Citado 2 vezes nas páginas 19 e 23.

COSSINS, R.; FERREIRA, P. Celeritas: A coloured petri net approach to simulation and control of flexible manufacturing systems. *The International Journal Of Production Research*, Taylor & Francis, v. 30, n. 8, p. 1925–1956, 1992. Citado 2 vezes nas páginas 19 e 24.

GIACOMO, G. D.; MASSACCI, F. Combining deduction and model checking into tableaux and algorithms for converse-pdl. *Information and Computation*, v. 162, n. 1, p. 117 – 137, 2000. ISSN 0890-5401. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0890540199928523>>. Citado 2 vezes nas páginas 29 e 33.

GROOVER, M. *Fundamentals of modern manufacturing: materials, processes and systems*. 4th. ed. [S.l.]: JOHN WILEY & SONS, INC., 2010. v. 1. Citado 5 vezes nas páginas 19, 24, 26, 27 e 42.

HAREL, D.; TIURYN, J.; KOZEN, D. *Dynamic Logic*. Cambridge, MA, USA: MIT Press, 2000. ISBN 0262082896. Citado na página 33.

HUANG, B.; ZHOU, M.; ZHANG, G. Synthesis of petri net supervisors for fms via redundant constraint elimination. *Automatica*, v. 61, p. 156–163, 2015. Cited By 2. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84943383104&doi=10.1016/jfj>>.

[automatica.2015.08.011&partnerID=40&md5=5953ffc8dec676185c891b535bb48e2e>](#). Citado 3 vezes nas páginas 39, 55 e 60.

HUANG, H.-P.; CHANG, P.-C. Specification, modelling and control of a flexible manufacturing cell. *The International Journal Of Production Research*, Taylor & Francis, v. 30, n. 11, p. 2515–2543, 1992. Citado 3 vezes nas páginas 24, 26 e 43.

KOCHIKAR, V.; NARENDHAN, T. A framework for assessing the flexibility of manufacturing systems. *The International Journal Of Production Research*, Taylor & Francis, v. 30, n. 12, p. 2873–2895, 1992. Citado na página 28.

_____. On using abstract models for analysis of flexible manufacturing systems. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, Taylor & Francis, v. 32, n. 10, p. 2303–2322, 1994. Citado na página 28.

LEE, J.; LEE, J. S. Heuristic search for scheduling flexible manufacturing systems using lower bound reachability matrix. *Computers & Industrial Engineering*, v. 59, n. 4, p. 799 – 806, 2010. ISSN 0360-8352. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0360835210002305>. Citado na página 39.

LOPES, B.; BENEVIDES, M.; HAEUSLER, E. H. Extending propositional dynamic logic for petri nets. *Electronic Notes in Theoretical Computer Science*, v. 305, p. 67 – 83, 2014. ISSN 1571-0661. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1571066114000528>. Citado 8 vezes nas páginas 20, 29, 44, 47, 50, 54, 63 e 64.

LOPES, B.; BENEVIDES, M. R. F.; HAEUSLER, E. H. Propositional dynamic logic for petri nets. *Logic Journal of the IGPL*, v. 22, n. 5, p. 721–736, 2014. Disponível em: <http://dx.doi.org/10.1093/jigpal/jzu010>. Citado 3 vezes nas páginas 44, 47 e 50.

MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, v. 77, n. 4, p. 541–580, Apr 1989. ISSN 0018-9219. Citado 2 vezes nas páginas 20 e 44.

NAGALINGAM, S. V.; LIN, G. C. Latest developments in {CIM}. *Robotics and Computer-Integrated Manufacturing*, v. 15, n. 6, p. 423 – 430, 1999. ISSN 0736-5845. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0736584599000368>. Citado 2 vezes nas páginas 19 e 27.

NALON, C. *et al.* A calculus for automatic verification of petri nets based on resolution and dynamic logics. *Electronic Notes in Theoretical Computer Science*, v. 312, p. 125 – 141, 2015. ISSN 1571-0661. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1571066115000122>. Citado 2 vezes nas páginas 29 e 47.

OLIVEIRA, G. de; JULIA, S.; PASSOS, L. S. Game modeling using workflow nets. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Anchorage, AK: IEEE, 2011. p. 838–843. ISSN 1062-922X. Citado na página 20.

PETERSEN, K. *et al.* Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swinton, UK, UK: British Computer Society, 2008. (EASE'08), p. 68–77. Disponível em: <http://dl.acm.org/citation.cfm?id=2227115.2227123>. Citado na página 40.

PETRI, C. A. Fundamentals of a theory of asynchronous information flow. 1962. Citado 3 vezes nas páginas 20, 33 e 43.

RIBEIRO, A. da S.; II, E. J. L.; LEPIKSON, H. Implementation of a {SCADA} system for flexible manufacturing systems using object-oriented petri nets model. *{IFAC} Proceedings Volumes*, v. 42, n. 4, p. 1673 – 1678, 2009. ISSN 1474-6670. 13th {IFAC} Symposium on Information Control Problems in Manufacturing. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1474667016340381>>. Citado 3 vezes nas páginas 28, 55 e 56.

SCHMIDT, R. A. *PDL-TABLEAU*. 2003. Disponível em: <<http://www.cs.man.ac.uk/~schmidt/pdl-tableau/>>. Acessado em: novembro de 2016. Citado na página 33.

SILVA, M. 50 years after the phd thesis of carl adam petri: A perspective. *{IFAC} Proceedings Volumes*, v. 45, n. 29, p. 13 – 20, 2012. ISSN 1474-6670. 11th {IFAC} Workshop on Discrete Event Systems. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1474667015401430>>. Citado 3 vezes nas páginas 20, 33 e 43.

STRONG, A. B. *Fundamentals of composites manufacturing: materials, methods and applications*. [S.l.]: Society of Manufacturing Engineers, 2008. Citado na página 24.

UZAM, M.; GELEN, G.; SALEH, T. Think-globally-act-locally approach with weighted arcs to the synthesis of a liveness-enforcing supervisor for generalized petri nets modeling fmss. *Information Sciences*, v. 363, p. 235–260, 2016. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84947307696&doi=10.1016%2Fj.ins.2015.09.010&partnerID=40&md5=edd993f3a6641f91b253450459537ad8>>. Citado 4 vezes nas páginas 39, 59, 60 e 61.

UZAM, M. *et al.* A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems. *Journal of Intelligent Manufacturing*, v. 27, n. 5, p. 1111–1129, 2016. ISSN 1572-8145. Disponível em: <<http://dx.doi.org/10.1007/s10845-014-0938-z>>. Citado na página 60.

VISWANADHAM, N.; NARAHARI, Y.; JOHNSON, T. Stochastic modelling of flexible manufacturing systems. *Mathematical and Computer Modelling*, v. 16, n. 3, p. 15 – 34, 1992. ISSN 0895-7177. Disponível em: <<http://www.sciencedirect.com/science/article/pii/089571779290043K>>. Citado na página 19.

YAN, H.-S. *et al.* Modelling, scheduling and simulation of flexible manufacturing systems using extended stochastic high-level evaluation petri nets. *Robotics and Computer-Integrated Manufacturing*, v. 14, n. 2, p. 121 – 140, 1998. ISSN 0736-5845. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0736584597000197>>. Citado 3 vezes nas páginas 19, 43 e 55.