

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia Mecânica

Matheus Ungaretti Borges

**DESENVOLVIMENTO DE UMA  
METODOLOGIA EM REDES DE PETRI A  
OBJETOS INTERPRETADA**

Belo Horizonte, MG, Brasil

2020

Matheus Ungaretti Borges

**DESENVOLVIMENTO DE UMA METODOLOGIA EM REDES  
DE PETRI A OBJETOS INTERPRETADA**

Dissertação apresentada ao Programa de  
Pós-Graduação em Engenharia Mecânica  
da Universidade Federal de Minas Gerais,  
como requisito parcial para obtenção do  
título de Mestre em Engenharia Mecânica

Área de concentração: Projeto e Sistemas

Orientador: Prof. Dr. Eduardo José Lima II

Belo Horizonte, MG, Brasil

2020

B732d	<p>Borges, Matheus Ungaretti.          Desenvolvimento de uma metodologia em redes de Petri a objetos interpretada [recurso eletrônico] / Matheus Ungaretti Borges. - 2020.          1 recurso online (54 f. : il., color.) : pdf.</p> <p>Orientador: Eduardo José Lima II.</p> <p>Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.</p> <p>Bibliografia: f. 51-54.</p> <p>Exigências do sistema: Adobe Acrobat Reader.</p> <p>1. Engenharia mecânica - Teses. 2. Redes de Petri - Teses. I, Lima II, Eduardo José. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.</p>
	CDU: 621 (043)



**UNIVERSIDADE FEDERAL DE MINAS GERAIS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM**  
**ENGENHARIA MECÂNICA**

Av. Antônio Carlos, 6627 - Campus Universitário  
31270-901 - Belo Horizonte - MG  
Tel.: +55 31 3409.5145  
E-mail: [cpgmec@demec.ufmg.br](mailto:cpgmec@demec.ufmg.br)

**"DESENVOLVIMENTO DE UMA METODOLOGIA EM REDES DE  
PETRI A OBJETOS INTERPRETADA"**

**MATHEUS UNGARETTI BORGES**

Dissertação submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do título de "**Mestre em Engenharia Mecânica**", na área de concentração de "**Projeto e Sistemas**".

Dissertação aprovada no dia 11 de fevereiro de 2020.

Por:

---

**Prof. Eduardo Jose Lima II**

Orientador - Departamento de Engenharia Mecânica/ UFMG

---

**Prof. Carlos Andrey Maja**

Departamento de Engenharia Elétrica/ UFMG

---

**Prof. Pedro Henrique Ferreira Machado**

Instituto Federal de Minas Gerais/ IFMG

*Dedico este trabalho aos meus pais, João e Ruth,  
irmãos, Luciana e Daniel, e sobrinha-afilhada, Letícia.*

## AGRADECIMENTOS

A realização deste trabalho de mestrado que culmina nesta dissertação é fruto de muita dedicação e estudo. Além do meu próprio esforço, eu, felizmente, pude contar com o apoio de muitas pessoas queridas. Do fundo do coração, quero expressar a minha mais sincera gratidão a cada um que me ajudou de alguma forma a trilhar esse caminho.

Agradeço à minha família, meus pais e irmãos, pois eles são fonte de carinho, compreensão, inspiração e apoio. Os seus exemplos ao longo da vida foram fundamentais para a construção de quem sou hoje e também para a realização deste sonho. Que possamos celebrar juntos esta conquista e também as próximas, que certamente virão.

Ao professor Dr. Eduardo José Lima II, meu orientador. Sua disponibilidade, dedicação e cordialidade definitivamente criaram um ambiente muito agradável e propício para trabalhar. A sua confiança na minha pesquisa e as oportunidades fornecidas só enriqueceram ainda mais a minha experiência acadêmica. Felizmente, posso dizer que, além de um orientador, ganhei um amigo. Muito obrigado por tudo!

Aos meus colegas de pós-graduação e amigos, Bryan Castro e Augusto Moura, com quem convivo desde o começo da minha graduação e compartilho muitas histórias. Ao também colega de pós-graduação e amigo, Bernardo Murta, com quem dividi laboratório e conversei sobre temas além do trabalho, como música e outras coisas da vida. Ao técnico e amigo, Edson, e ao amigo Fernando Augusto pelo suporte e grande incentivo durante este trabalho. Aos demais colegas da pós, que compartilharam esses anos de mestrado comigo. A companhia e as conversas tomando um cafezinho no intervalo foram fundamentais, pois o descanso também é importante para a realização de um trabalho.

Aos amigos da vida, que nem sempre podem estar presentes fisicamente, mas que sempre estarão presentes no meu coração. E que cada encontro que a vida nos proporcionar seja uma alegria diferente para compartilhar e gravar na memória.

Aos meus professores, da infância até a pós-graduação, por todo o conhecimento e experiência transmitidos. Alguns momentos definitivamente marcaram a minha vida. Todos foram importantes para a construção de quem sou hoje. Agradeço de coração por dedicarem a vida a esta bonita profissão de ensinar as pessoas!

Agradeço também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro fornecido.

*"So carry on, there's a meaning to life  
Which someday we may find."*

— ANDRÉ MATOS (1971 - 2019)

*"Tenho o caminho do que sempre quis  
E um saveiro pronto pra partir*

*Invento o cais*

*E sei a vez de me lançar"*

— MILTON NASCIMENTO (1942 - )

*"Remember to look up at the stars and not down at your feet. Try to make sense of  
what you see and wonder about what makes the universe exist. Be curious. And  
however difficult life may seem, there is always something you can do and succeed at.*

*It matters that you don't just give up."*

— STEPHEN HAWKING (1942 - 2018)

# RESUMO

Dissertação de Mestrado  
Programa de Pós-Graduação em Engenharia Mecânica  
Universidade Federal de Minas Gerais

## DESENVOLVIMENTO DE UMA METODOLOGIA EM REDES DE PETRI A OBJETOS INTERPRETADA

AUTOR: MATHEUS UNGARETTI BORGES

ORIENTADOR: EDUARDO JOSÉ LIMA II

Local da Defesa e Data: Belo Horizonte, 11 de Fevereiro de 2020.

Na década de 1960, o matemático e cientista da computação C. A. Petri desenvolveu uma importante ferramenta que permite que eventos antes descritos apenas por relações temporais passassem a ser descritos por relações causais. O formalismo da rede de Petri (RdP) permite a representação de sistemas a eventos discretos (SED), que são descritos pela sequência de eventos que podem ocorrer e pela sequência de estados que podem ser visitados. Durante mais de 50 anos, inspirados por Petri, cientistas criaram outras definições de redes capazes de agregar mais informações. Tais redes conseguem representar sistemas mais complexos, que antes não podiam ser descritos por causa da explosão de estados dos autômatos de estados finitos. Duas delas são particularmente importantes: (i) redes de Petri interpretadas (RdPi), capazes de receber sinais de entrada, processá-los e depois fornecer sinais de saída para um Controlador Lógico Programável (CLP) executar uma função; (ii) redes de Petri a objetos (RdPo), capazes de tratar as fichas como objetos individuais portadores de informação, com diferentes atributos e métodos. Essas definições de rede nem sempre conseguem descrever um problema proposto quando usadas separadamente. Este trabalho fundamenta-se nos conceitos de lugares, transições, arcos, função peso e marcação de uma RdP (incluindo a notação gráfica), de orientação a objetos de alto nível e de interpretação de sinais de baixo nível. O objetivo deste trabalho é formalizar uma nova definição de rede de Petri que incorpore os conceitos acima: a rede de Petri a objetos interpretada (RdPoi). A notação desenvolvida é aplicada em um estudo de caso para polimento e separação de materiais recicláveis (metal, madeira e plástico preto), visto que reciclagem é um tema de destaque atualmente e que pode ter uma aplicação industrial prática. O estudo de caso inicia-se com todas as peças em seu estado bruto em um *buffer* inicial. A separação é feita pela leitura de uma combinação lógica da leitura de sinais de baixo nível de sensores indutivo, óptico e capacitivo. Uma vez que o material é identificado, tal informação é incorporada ao objeto e as devidas ações são executadas quando e se necessário. Ao final do processo, todos os materiais estarão polidos e separados em seus respectivos *buffers*. A implementação da RdPoi mostrada neste estudo de caso serve como exemplo para que novas implementações sejam realizadas em trabalhos futuros, respeitando-se as especificidades de cada sistema.

**Palavras-chave:** Rede de Petri. Rede de Petri interpretada. Rede de Petri a objetos. Orientação a objeto.

# ABSTRACT

Master's Dissertation  
Post-Graduate Program in Mechanical Engineering  
Federal University of Minas Gerais

## DEVELOPMENT OF A SIGNAL INTERPRETED OBJECT PETRI NETS METHODOLOGY

AUTHOR: MATHEUS UNGARETTI BORGES

ADVISOR: EDUARDO JOSÉ LIMA II

Defense Place and Date: Belo Horizonte, February 11<sup>th</sup>, 2020.

In the 1960s, the mathematician and computer scientist C. A. Petri developed an important tool that allows events previously described only by temporal relations to be described by causal relations. Petri Nets (PN) formalism allows the representation of discrete event systems (DES), which are described by the sequence of events that can occur and the sequence of states that can be visited. For more than 50 years, inspired by Petri, scientists have created other net definitions that can aggregate more information and thus represent more complex systems. Such nets are able to represent more complex systems, which previously could not be described because of the explosion of states of finite state automata. Two of these are particularly important: (i) Signal Interpreted Petri Net (SIPN), capable of receiving input signals, processing them, and then providing output signals for a Programmable Logic Controller (PLC) to execute a method; (ii) Object Petri Net (OPN), capable of treating the tokens as individual objects that carry information, with different attributes and methods. These net definitions are not always able to describe a proposed problem when used separately. This work is based on the concepts of places, transitions, arcs, weight and marking function of a PN (including graphic notation), object orientation (high level) and signal interpretation (low level). The objective of this work is to formalize a new definition of Petri net that incorporates the above concepts: the Signal Interpreted Object Petri Net (SIOPN). The developed notation is applied in a case study for polishing and separating recyclable materials (metal, wood and black plastic), since recycling is a prominent topic today and may have a practical industrial application. The case study begins with all blocks in their raw state in an initial buffer. The separation is done by reading a logical combination of low level signals from inductive, optical and capacitive sensors. Once the material is identified, such information is incorporated into the object and appropriate actions are taken when and if necessary. At the end of the process, all materials will be polished and separated into their respective buffers. The implementation of SIOPN shown in this case study serves as an example for new implementations to be carried out in future works, respecting the particularities of each system.

**Keywords:** Petri Nets. Signal Interpreted Petri Nets. Object Petri Nets. Object Orientation.

## LISTA DE FIGURAS

Figura 2.1 – Elementos básicos da rede de Petri .....	19
Figura 2.2 – Transição, lugares, arcos, peso e marcação da rede: (a) antes e (b) depois do disparo da transição.....	21
Figura 2.3 – Rede de Petri das estações do ano: (a) cíclica e (b) linear. ....	22
Figura 2.4 – Máquina de estados finitos representada por uma Rede de Petri. ....	22
Figura 2.5 – Fluxograma do funcionamento da RdPi. ....	24
Figura 2.6 – Dispositivos da bancada: (a) esquema e (b) montagem.....	25
Figura 2.7 – Exemplo em RdP com interpretação manual da rede .....	26
Figura 2.8 – Fluxograma do funcionamento do SFC (ciclo de varredura). ....	31
Figura 2.9 – Exemplo na linguagem SFC.....	32
Figura 3.1 – Planta simplificada do estudo de caso .....	36
Figura 3.2 – Planta detalhada do estudo de caso. ....	37
Figura 3.3 – Atuador movimentando a peça do (a) <i>buffer</i> inicial para a (b) esteira E1. ....	37
Figura 3.4 – Vista superior da estação de triagem e dos <i>buffers</i> finais. ....	38
Figura 4.1 – Transição da rede de Petri a objetos interpretada: (a) antes e (b) depois do disparo da transição.....	41
Figura 4.2 – Dinâmica da Rede de Petri a objetos interpretada. ....	42
Figura 5.1 – Rede de Petri a objetos interpretada do estudo de caso. ....	43
Figura 5.2 – Rede de Petri com as condições de disparo das transições para o estudo de caso. ....	48

## LISTA DE TABELAS

Tabela 2.1 – Função das condições de disparo para cada transição .....	27
Tabela 2.2 – Estado de saída para cada lugar da RdPi .....	28
Tabela 2.3 – Endereços de entrada, saída e memórias do CLP.....	28
Tabela 2.4 – Relação entre RdPi e SFC.....	31
Tabela 3.1 – Lógica dos sensores para identificação dos materiais. ....	37
Tabela 5.1 – Significado dos lugares no estudo de caso.....	44
Tabela 5.2 – Lógica de interpretação das condições de disparo das transições no estudo de caso.....	45
Tabela 5.3 – Lógica dos sinais de saída dos atuadores pneumáticos no estudo de caso $\omega(p_i) = [AADV, ARET, XADV, XRET, YADV, YRET, ZADV, ZRET]$ .	46
Tabela 5.4 – Lógica de orientação a objetos das condições de disparo das transições no estudo de caso.....	47
Tabela 5.5 – Funções das lógicas de interpretação e de orientação a objetos das condições de disparo das transições no estudo de caso. ....	48

## LISTA DE ABREVIATURAS E SIGLAS

CLP	Controlador Lógico Programável
RdP	Rede de Petri
RdPi	Rede de Petri interpretada
RdPo	Rede de Petri a objetos
RdPoi	Rede de Petri a objetos interpretada
SED	Sistema a Eventos Discretos

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	14
<b>2 REVISÃO BIBLIOGRÁFICA</b> .....	19
<b>2.1 Notação gráfica das Redes de Petri</b> .....	19
<b>2.2 Definições de redes de Petri</b> .....	20
2.2.1 Rede de Petri .....	20
2.2.2 Rede de Petri interpretada .....	23
2.2.2.1 Definição de FREY (2000) .....	23
2.2.2.2 Sequential Function Chart IEC (2003) como rede de Petri interpretada .....	29
2.2.3 Rede de Petri a objetos .....	32
2.2.3.1 Definição de CARDOSO; VALETTE (1997) .....	33
<b>2.3 Conceitos de orientação a objetos</b> .....	34
<b>3 METODOLOGIA</b> .....	36
<b>3.1 Formalização do estudo de caso</b> .....	36
<b>4 DESENVOLVIMENTO</b> .....	39
<b>4.1 Definição formal da Rede de Petri a Objetos Interpretada</b> .....	39
4.1.1 Dinâmica da rede .....	41
<b>5 RESULTADOS E DISCUSSÕES</b> .....	43
<b>5.1 Aplicação da notação no estudo de caso proposto</b> .....	43
<b>6 CONCLUSÃO</b> .....	49
<b>REFERÊNCIAS</b> .....	51

# 1 INTRODUÇÃO

Rede de Petri (RdP) é um conceito que surgiu na ciência da computação na década de 1960, mais precisamente em 1962, quando o cientista C. A. Petri defendeu seu doutorado com a tese sobre comunicação de autômatos, intitulada "*Kommunikation mit Automaten*" (PETRI, 1962). O formalismo matemático de PETRI permitiu substituir a relação temporal pela relação causal, além de definir conceitos como: concorrência, paralelismo, cooperação, competição e alocação de recursos, inerentes ao comportamento de sistemas a eventos discretos (SED) (RAMÍREZ-TREVIÑO; RIVERA-RANGEL; LÓPEZ-MELLADO, 2003; SILVA, 2012). Diferentemente do que é frequentemente citado, a notação gráfica desenvolvida por Petri não apareceu em sua tese de doutorado, mas apenas alguns anos depois (SILVA, 2012) em uma palestra sobre a teoria dos autômatos chamada "*Fundamentals on the description of discrete processes*", ocorrida no 3º colóquio sobre Teoria dos Autômatos, em Hanover (PETRI, 1967; BRAUER; REISIG, 2009). A representação da rede de Petri é feita em forma de grafo, a partir de elementos básicos de lugar, transição, arcos e fichas.

Sistemas a eventos discretos são modelos que descrevem sistemas que a cada instante ocupam um determinado estado discreto, que pode ser alterado dada a ocorrência de eventos. O comportamento de um SED é descrito pela sequência de eventos que podem ocorrer e pela sequência de estados que podem ser visitados (SKÖLDSTAM; ÅKESSON; FABIAN, 2007). Uma forma de representação é através do autômato de estados finitos, como exemplificado pelo próprio SKÖLDSTAM; ÅKESSON; FABIAN (2007), e que pode ser inviável para representar casos mais complexos por causa da explosão de estados. Para lidar com este problema, pesquisadores passaram a modelar SEDs usando o formalismo da RdP para representar, por exemplo: sistemas de manufatura, redes de comunicação, sistemas embarcados e sistemas de controle supervisorio.

Em homenagem aos cinquenta anos da importante criação de Petri, SILVA (2012) escreveu o artigo "*50 years after the PhD thesis of Carl Adam Petri: A perspective*". Este tema impactou fortemente a área da ciência da computação e do tratamento de dados, além de contribuir para o desenvolvimento de diversas áreas, como já previsto pelo próprio Petri em "*Introduction to General Net Theory*" (PETRI, 1980). Durante estes mais de cinquenta anos, pesquisadores se debruçaram e continuam se debruçando sobre a tese para desenvolver novas definições, criando redes capazes de agregar ainda mais informações permitindo que diversos sistemas sejam

descritos da forma mais realista possível, tornando-se assim uma ferramenta bastante versátil.

MURATA (1989) escreveu um artigo chamado "*Petri Nets: Properties, Analysis and Applications*", no qual explica os elementos básicos de uma rede de Petri, incluindo o peso e a marcação inicial. Além disso, apresenta diversos exemplos de aplicação.

No início dos anos 2000, algumas definições de rede de Petri interpretadas (RdPi) foram publicadas (FREY, 2000; RAMÍREZ-TREVIÑO; RIVERA-RANGEL; LÓPEZ-MELLADO, 2003). As RdPi são capazes de receber sinais de entrada (externos ao modelo), processá-los e gerar um sinal de saída para comandar uma ação em Controladores Lógico Programáveis (CLPs) (FREY, 2000).

CLPs são uma ferramenta poderosa amplamente empregada em operações lógicas e sequenciamento de atividades dentro da automação industrial, realizando cálculos de forma rápida e robusta. Os comandos são armazenados na memória do CLP em forma de lista e processados posteriormente (MELLO et al., 2012). A norma IEC 61131-3 padroniza cinco linguagens de programação de CLPs para aplicações industriais (IEC, 2003; MELLO et al., 2012): *Sequential Function Chart* (SFC); *Ladder Diagram* (LD); *Function Block Diagram* (FBD); *Instruction List* (IL) e *Structured Text* (ST). Segundo SILVA (2012), a linguagem SFC é derivada do Grafcet. Há uma relação entre essas linguagens de programação e redes de Petri interpretadas, como para o SFC (MELLO et al., 2012) e para o LD (BORGES; LIMA II, 2018).

A construção lógica dada pela interpretação da rede pode reduzir erros de programação em microcontroladores (BORGES; LIMA II, 2017). Além disso, é possível observar a evolução dinâmica da rede e, simultaneamente, a geração de sinais de saída a partir da ocorrência de sinais de entrada (KLEIN; FREY; MINAS, 2003), caso as regras de disparo de transição, que serão debatidas posteriormente, sejam respeitadas.

Nos anos 1960, quando a rede de Petri foi criada, ainda não existia programação orientada a objetos (LETIA; AL-JANABI, 2018). Sendo assim, os conceitos de orientação a objetos como herança, polimorfismo, encapsulamento e transmissão de mensagem tiveram que ser adaptados para serem inseridos na notação da rede de Petri. Com estas propriedades, aumenta-se a modularização e a flexibilidade de modelos simulados (ZHAO et al., 2019). SALDHANA; SHATZ (2000) e GUERRERO (2002) dissertam justamente sobre a importância de uma notação semântica para compatibilizar a orientação ao objeto com rede de Petri a objetos. Por ser uma adaptação, percebe-se que não há uma padronização nas notações (e nos nomes) de rede de Petri a objetos, sendo que várias foram apresentadas a partir do início dos anos 1980 (ZHAO

et al., 2019). Ao incorporar os conceitos de orientação a objetos, as redes de Petri a objeto (RdPo) adquiriram a capacidade de tratar as fichas como objetos individuais com diferentes atributos, passando a ser um elemento de informação (CARDOSO; VALETTE, 1997).

Portanto, é fundamental discutir a importância do uso de redes de Petri como ferramenta de representação de sistemas a eventos discretos considerando todos os avanços desta teoria desde a sua criação. As redes interpretadas permitem a interpretação de sinais de entrada e saída, enquanto as redes orientadas a objetos permitem que objetos sejam entidades dotadas de informação. Entretanto, essas redes sozinhas nem sempre são capazes de solucionar o problema apresentado em alguns casos. Sendo assim, o objetivo desta dissertação é desenvolver uma nova construção formal com notação própria capaz de aliar estas características.

## **Justificativa**

A teoria de redes é voltada principalmente para computadores e tratamento de informações, mas inspirou muitas ideias sobre estrutura e organização em outras áreas do conhecimento, principalmente as que utilizam computador. Por exemplo: engenharia, física, química, matemática, administração, entre outras (PETRI, 1980).

MURATA (1989) destaca em trecho introdutório do seu artigo:

"As redes de Petri podem ser usadas por pesquisadores experimentais (práticos) e teóricos. Assim, há um poderoso meio de comunicação entre eles: os experimentais podem aprender com os teóricos como tornar seus modelos mais metódicos, e os teóricos podem aprender com os experimentais como tornar seus modelos mais realistas."

MURATA (1989)

Este trabalho pode ser justificado como uma nova construção formal usando a teoria de redes de Petri para a descrição de sistemas a eventos discretos. Baseando-se em uma nova notação que unifica a rede interpretada e a rede a objetos para resolver problemas de controle, diagnóstico e identificação de sistemas, é possível descrever sistemas com mais informações e, portanto, mais complexos. Com esta notação, aplicações em engenharia mecânica e outras disciplinas podem ser estudadas.

## **Objetivo Geral**

Este trabalho tem como principais objetivos definir uma notação formal própria para uma rede de Petri a objetos interpretada (RdPoi) juntando informações dos dois tipos de rede e desenvolver uma metodologia a partir da construção desta notação. Esta notação será aplicada

em um estudo de caso sobre separação de materiais e poderá ser aplicada por outros pesquisadores em outros casos.

### **Objetivos Específicos**

- Estudar o formalismo das redes de Petri interpretadas;
- Estudar o formalismo das redes de Petri a objetos;
- Desenvolver uma notação para representação da interpretação da rede utilizando classes formais da rede de Petri a objetos;
- Aplicação da notação em um estudo de caso.

### **Estrutura da dissertação**

Esta dissertação é dividida em seis capítulos, além de todas as referências mencionadas ao longo do texto apresentadas ao final.

O Capítulo 1 (Introdução) é uma contextualização histórica da ferramenta desenvolvida por Petri em seu doutorado, estudada ao longo dos mais de cinquenta anos após sua publicação como, por exemplo, em aplicações em sistemas a eventos discretos e em controle supervisorio de CLPs. Também são apresentados a justificativa e os objetivos gerais e específicos do trabalho, além da descrição do que será abordado em cada capítulo.

O Capítulo 2 (Revisão bibliográfica) apresenta a notação gráfica desenvolvida por Petri, a definição formal de rede de Petri apresentada por MURATA (1989), a definição de redes de Petri interpretadas apresentada por FREY (2000) e a definição de redes de Petri orientadas a objeto apresentada por CARDOSO; VALETTE (1997), justificando essas escolhas durante o decorrer do capítulo. Por último, são explicados os conceitos de orientação a objeto, provenientes da ciência da computação.

O Capítulo 3 (Metodologia) formaliza o problema que será utilizado como estudo de caso presente nesta dissertação, cujo contexto é a separação de materiais para reciclagem, tema de destaque atualmente com possíveis aplicações industriais. As principais características do sistema são descritas a partir dos conceitos apresentados no capítulo de revisão.

O Capítulo 4 (Desenvolvimento) apresenta a construção da notação da rede de Petri a objetos interpretada (RdPoi). Nele, cada termo da notação é definido e explicado. Além disso,

também são descritas todas as condições necessárias para que as transições sejam disparadas (regras de disparo). O comportamento dinâmico da rede acontece com o disparo de transições, que promovem a mudança dos estados da rede.

O Capítulo 5 (Resultados e discussões) mostra os resultados e a discussão da aplicação da notação da rede de Petri a objetos interpretada em um estudo de caso. É definida a estrutura da rede que modela o sistema, apresentando o significado de cada lugar e de cada transição da rede e analisando suas peculiaridades para o funcionamento do sistema.

O Capítulo 6 (Conclusão) retoma a importância do trabalho de Petri em seu contexto histórico e levanta as conclusões obtidas durante o desenvolvimento do trabalho, principalmente as relativas à junção da interpretação dos sinais de baixo nível com os conceitos da orientação a objetos de alto nível. Além disso, são apresentadas sugestões de tópicos de pesquisa para trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

Dado o contexto histórico presente na introdução, o capítulo de revisão apresentará os principais conceitos e definições das ferramentas que servirão de base para o desenvolvimento proposto nesta dissertação.

### 2.1 Notação gráfica das Redes de Petri

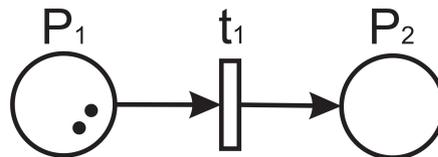
A notação gráfica criada por Petri, juntamente com o formalismo matemático já mencionado, consiste em usar elementos simples - círculos (lugares), retângulos (transições) e setas (arcos), apresentados na rede de Petri da Figura 2.1 - na construção de uma rede capaz de representar diversas situações.

Segundo a definição do próprio Petri:

"Os retângulos são transições que representam atividades de produção como transporte, montagem ou desmontagem; os círculos são lugares onde recursos podem ser armazenados temporariamente; as setas são os arcos da rede que representam uma relação direta de acesso imediato. É importante ressaltar que os arcos não são canais por onde as fichas fluem, já que não possuem nenhuma evidência de existência física."

PETRI (1980).

Figura 2.1 – Elementos básicos da rede de Petri



Fonte: Autor

Graficamente, as fichas são pontos pretos contidos nos lugares para cada estado de um sistema a eventos discretos (SED) e representam exatamente esses recursos. Por exemplo, podem ser uma peça a ser trabalhada ou mesmo uma máquina para fazer alguma atividade, como separação de peças de materiais diferentes. Além disso, segundo PETRI (1980) os recursos podem ser tratados como informação, o que é interessante justamente por ser mais complexo e mais geral. Os recursos, geralmente escassos, são modelados para limitar ou impedir a ocorrência de algum evento que ocasione problemas operacionais do sistema em questão.

## 2.2 Definições de redes de Petri

### 2.2.1 Rede de Petri

MURATA (1989) define uma rede de Petri como  $RdP = (P, T, F, W, M_0)$ , um vetor de 5 elementos em que:

- $P = \{p_1, p_2, \dots, p_m\}$  é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos;
- $W: F \rightarrow \{1, 2, 3, \dots\}$  é a função peso;
- $M_0: P \rightarrow \{0, 1, 2, 3, \dots\}$  é a marcação inicial;
- $P \cap T = \emptyset$  e  $P \cup T \neq \emptyset$ .

A função peso  $W$  é uma função nos inteiros positivos que determina quantas fichas são necessárias para que uma transição seja ativada (arcos lugar-transição) ou quantas fichas serão adicionadas a um lugar após o disparo de uma transição (arcos transição-lugar). O peso pode ser interpretado como uma série de  $k$  arcos paralelos e é graficamente rotulado por um número  $k$  escrito próximo ao arco. Quando unitário, o rótulo pode ser omitido (MURATA, 1989).

A marcação da rede é uma função que determina o estado representado pela rede a partir do número de fichas presentes em cada lugar. É representada por um vetor  $M$  de  $m$  elementos, em que  $m$  é o total de lugares da rede. Sendo assim, a marcação inicial  $M_0$  é justamente o estado inicial da rede (MURATA, 1989).

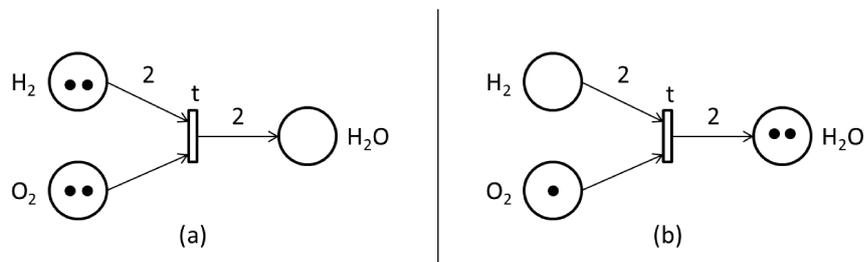
Como apresentado na introdução, redes de Petri podem ser usadas para modelar sistemas a eventos discretos. Para isso, os conceitos de condições e eventos são associados aos lugares e às transições, respectivamente. Uma das possíveis interpretações para lugares e transições apresentadas por MURATA (1989) trata justamente sobre sinais de entrada, processamento de sinais e sinais de saída, o que pode ser o início da ideia de redes de Petri interpretadas (RdPi) e sua aplicação em CLPs, como sugerido por diversos pesquisadores, como o FREY (2000), por exemplo.

A dinâmica da rede é dada pelos estados que podem ser alcançados, levando a rede de um estado para outro pelo disparo de transições. No caso da RdP, basta que as transições estejam habilitadas, ou seja, possuam fichas suficientes nos lugares que antecedem a transição a

ser disparada. No caso da RdPi, além de estar habilitada, a transição precisa ter satisfeitas suas condições de disparo (quando aplicável). Essa questão da dinâmica em RdPi será abordada novamente na Subsubseção 2.2.2.1.

A Figura 2.2 é um exemplo de Rdp que representa uma reação química. Em (a) temos a marcação da rede (por exemplo, a marcação inicial  $M_0$ ) com dois mols de hidrogênio e dois mols de oxigênio. Como o número de fichas satisfaz as condições da função peso dos respectivos arcos que conectam os lugares de hidrogênio e oxigênio, a transição  $t$  está habilitada e pode ser disparada, levando a rede para um novo estado representado por (b), em que a transição  $t$  está desabilitada. Este novo estado representa os dois mols de água que foram produzidos. Neste caso, o hidrogênio é completamente consumido e o oxigênio não, pois está em excesso.

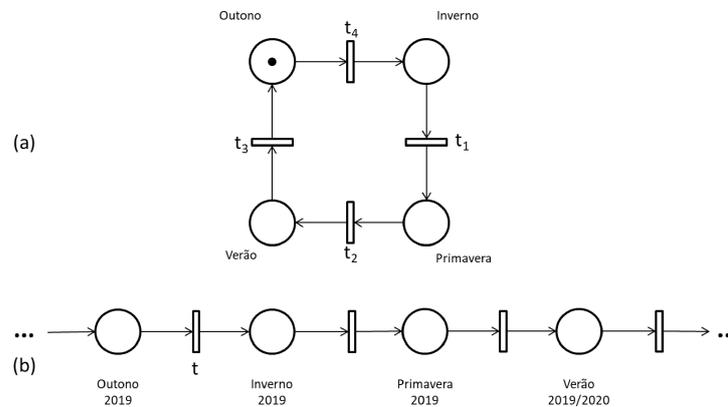
Figura 2.2 – Transição, lugares, arcos, peso e marcação da rede: (a) antes e (b) depois do disparo da transição.



Fonte: Adaptado de MURATA (1989)

A rede de Petri é uma ferramenta versátil e pode modelar desde equações químicas, como apresentado acima, até fenômenos cíclicos, como as estações do ano. A Figura 2.3(a) mostra uma RdP com as quatro estações em ciclo, sendo que o disparo das transições representam literalmente a transição de uma estação para a outra. Já a Figura 2.3(b) é uma representação de uma rede linear "infinita" das estações que leva em consideração também o ano em que aconteceu cada estação no hemisfério sul: outono 2019, inverno 2019, primavera 2019, verão 2019/2020, por exemplo. O ano de ocorrência diferencia os eventos pois primavera é sempre primavera, mas primavera 2019 é diferente de primavera 2020.

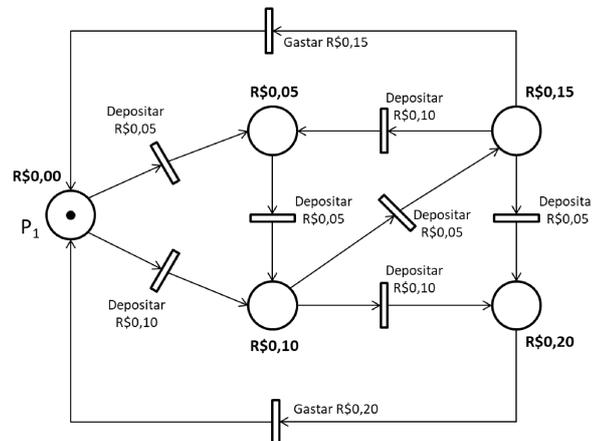
Figura 2.3 – Rede de Petri das estações do ano: (a) cíclica e (b) linear.



Fonte: Adaptado de PETRI (1980)

MURATA (1989) apresenta um exemplo de uma RdP que se comporta como uma máquina de estados finitos, ou seja, possui um número finito de estados e apenas um deles está ativo por vez. Portanto, há apenas uma ficha na marcação da rede. A Figura 2.4 apresenta uma representação esquemática em RdP do funcionamento de uma máquina de venda de doces. Há cinco lugares nesta rede, representando todos os estados possíveis para este modelo, onde cada estado representa quanto dinheiro o usuário possui na máquina. Já as transições estão relacionadas às ações que ocorrem para que a rede mude de estado. As ações possíveis são: depositar moeda de R\$0,05; depositar moeda de R\$0,10; comprar doce de R\$0,15 e comprar doce de R\$0,20. De acordo com as ações realizadas, as transições habilitadas são disparadas e permitem a mudança de estado.

Figura 2.4 – Máquina de estados finitos representada por uma Rede de Petri.



Fonte: Adaptado de MURATA (1989)

## 2.2.2 Rede de Petri interpretada

A rede de Petri interpretada foi desenvolvida para construir algoritmos lógicos de controle para serem implementados em CLPs. A interpretação da rede consiste em fornecer informações através de sinais de entrada a um controlador que irá processá-las para comandar o disparo das transições e gerar sinais de saída para comandar a execução das ações (MELLO et al., 2012).

Como apresentado na introdução, SFC é uma das linguagens de programação de CLPs para aplicações industriais padronizada pela norma IEC 61131-3 (IEC, 2003). Dadas algumas condições especiais, esta linguagem pode se comportar como uma RdPi. A seção 2.2.2.2 desta dissertação será dedicada a explicar este assunto. A definição de FREY (2000) foi escolhida para ser trabalhada justamente por possuir aplicações industriais e por ser possível representá-la em linguagens de programação de CLPs, como Ladder e SFC, por exemplo.

### 2.2.2.1 Definição de FREY (2000)

O formalismo apresentado por FREY tem os mesmos itens da definição de MURATA, exceto pela grandeza que representa o peso dos arcos. A rede de FREY é considerada segura, ou seja, só pode existir, no máximo, uma ficha em cada lugar. O avanço dele foi justamente interpretar a RdP e, para isso, foram necessárias algumas definições extras. Sendo assim, sua definição de RdPi é um vetor de 9 elementos  $RdPi = (P, T, F, m_0, I, O, \varphi, \omega, \Omega)$ , em que:

- $I$  é um conjunto de sinais lógicos de entrada com  $|I| > 0$ ;
- $O$  é um conjunto de sinais lógicos de saída com  $I \cap O = \emptyset, |O| > 0$ ;
- $\varphi$  é um mapa que associa cada transição  $t_i \in T$  com uma condição de disparo, tal que  $\varphi(t_i) =$  função booleana em  $I$ ;
- $\omega$  é um mapa que associa cada lugar  $p_i \in P$  com uma saída  $\omega(p_i) \in \{0, 1, -\}^{|O|}$ , em que  $(-)$  significa ‘não importa’;
- $\Omega$  é uma função de saída que combina as saídas  $\omega$  de todos os lugares marcados, tal que  $\Omega: m \rightarrow \{-, 1, 0, c, r_0, r_1, c_0, c_1, c_{01}\}^{|O|}$ . A combinação da saída pode ser indefinida  $(-)$ , um  $(1)$ , zero  $(0)$ , contraditória  $(c)$ , redundante em zero ou em um  $(r_0, r_1)$ , ou uma combinação de contradição e redundância  $(c_0, c_1, c_{01})$ .

O disparo de uma transição habilitada significa a remoção de uma ficha de cada pré-lugar e adição de uma ficha em cada pós-lugar conectados a ela. Entretanto, algumas regras foram definidas por FREY (2000) para que uma transição possa ser disparada.

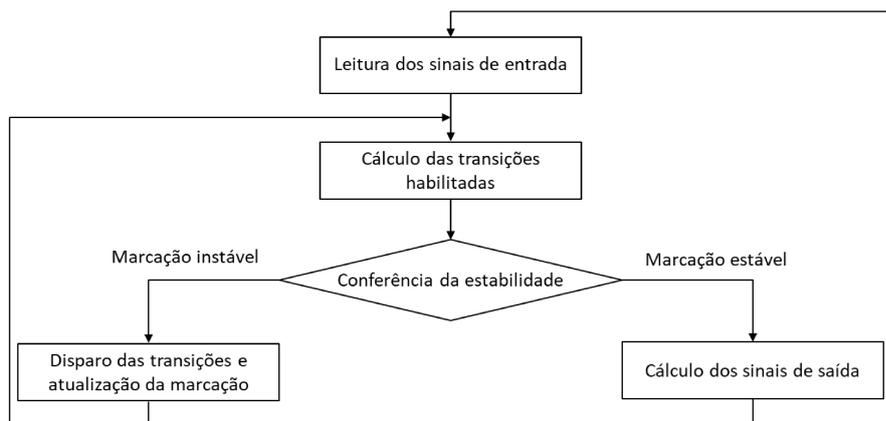
1. Uma transição é permitida, se todos os pré-lugares estiverem marcados e todos os pós-lugares estiverem desmarcados;
2. Uma transição dispara imediatamente, se estiver permitida e suas condições de disparo forem satisfeitas;
3. Todas as transições que puderem ser disparadas e que não estejam em conflito com outras transições disparam simultaneamente;
4. O processo de disparo é iterado até atingir uma marcação estável (i.e., até nenhuma transição poder disparar mais). Disparos iterados são interpretados como simultâneos. Isso também significa que uma mudança nos valores do sinal de entrada não pode ocorrer durante o processo de disparo.

Depois de atingir uma nova marcação estável, os sinais de saída são recalculados aplicando  $\Omega$  à marcação.

(FREY, 2000)

O fluxograma da Figura 2.5 (FREY, 2002) serve para explicar o funcionamento da dinâmica da RdPi, que começa com a leitura dos sinais de entrada e com o cálculo de quais transições estão habilitadas para serem disparadas. Neste instante, há a conferência de estabilidade da rede. Há duas situações possíveis para a marcação da rede: instável ou estável. Quando a marcação é instável, ou seja, quando ainda há transições habilitadas, estas são disparadas e a marcação da rede é atualizada. Quando a marcação é estável, os sinais de saída são calculados e os sinais de entrada são lidos novamente.

Figura 2.5 – Fluxograma do funcionamento da RdPi.

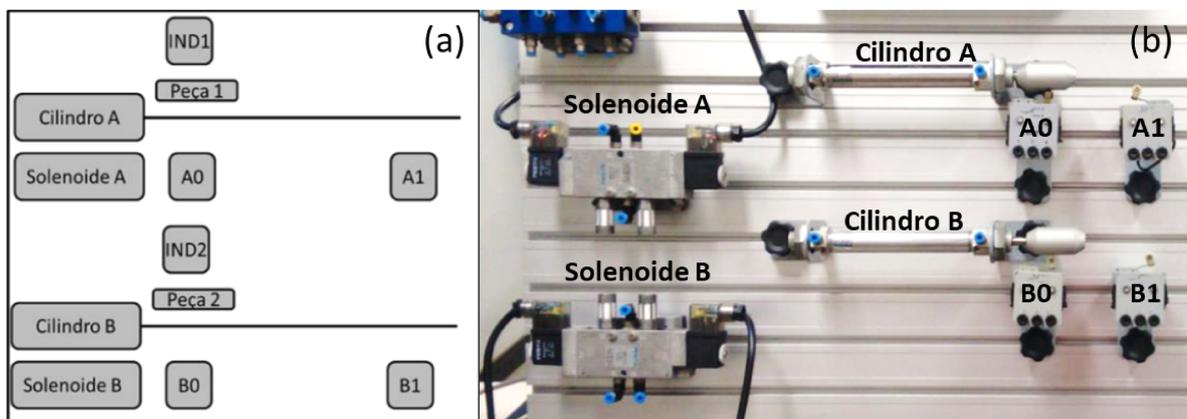


Fonte: Adaptado de FREY (2002)

### Exemplo de aplicação da notação

BORGES; LIMA II (2018) usam a notação de FREY (2000) em um exemplo de aplicação industrial para separar blocos metálicos de outros materiais. A Figura 2.6 mostra (a) o esquema e (b) a montagem de uma bancada com dois cilindros pneumáticos, quatro sensores de fim de curso, dois sensores indutivos (um para cada cilindro) e um botão de início *ST*. Os sensores indutivos e o botão *ST* não estão representados na Figura 2.6 (b).

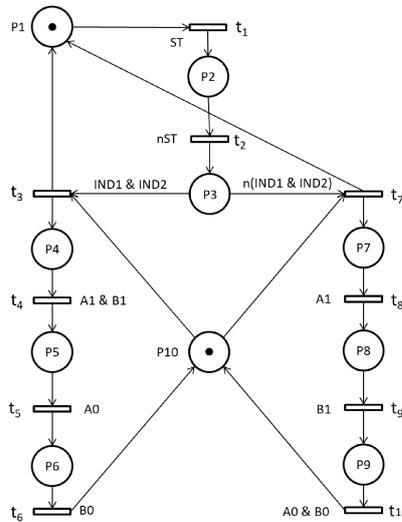
Figura 2.6 – Dispositivos da bancada: (a) esquema e (b) montagem.



Fonte: BORGES; LIMA II (2017)

Ao colocar um bloco na frente de cada cilindro, seus respectivos sensores indutivos responderão com um sinal booleano (*Verdadeiro* ou *Falso*) para informar se os blocos são metálicos ou não. O sistema executará ações diferentes dependendo da resposta obtida. Caso ambos os blocos sejam metálicos, os cilindros avançarão simultaneamente para recolher os blocos em um mesmo recipiente e depois recuarão separadamente. Caso apenas um bloco seja metálico ou caso nenhum bloco seja metálico, os cilindros avançarão separadamente para recolher o bloco metálico (se houver) e depois retornarão simultaneamente para finalizar o processo. A Figura 2.7 modela em RdPi a descrição representada com interpretação manual, em que  $\&$  significa a expressão lógica E e  $n$  significa *Falso*.

Figura 2.7 – Exemplo em RdP com interpretação manual da rede



Fonte: Adaptado de BORGES; LIMA II (2018) baseado em software de LIMA II (2002)

Aplicando as definições de RdPi no exemplo:

- $P = \{p_1, p_2, \dots, p_{10}\}$  (a rede tem 10 lugares);
- $T = \{t_1, t_2, \dots, t_{10}\}$  (a rede tem 10 transições);
- $F \subseteq (P \times T) \cup (T \times P)$  (os arcos estão representados na Figura 2.7);
- $W: F \rightarrow 1$  (neste caso, todos os arcos têm peso unitário);
- $M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$  (a marcação inicial pode ser escrita como um vetor com o número de fichas sendo um inteiro não negativo);
- $I = \{A0, A1, B0, B1, ST, IND1, IND2\}$  (os sinais de entrada são dados pelos sensores: fim de curso dos cilindros A (A0 e A1) e B (B0 e B1), botão de início (ST) e indutivos (IND1 e IND2));
- $O = \{AADV, ARET, BADV, BRET\}$  (sinais de saída são comandos a serem executados: avanço de A, retorno de A, avanço de B e retorno de B, respectivamente);
- $\varphi(t_i)$  são funções booleanas em  $I$  (Tabela 2.1) e verificam se as condições de disparo das transições são verdadeiras, em que  $\&$  significa a expressão lógica *E* e  $n$  significa *Falso*;
- $\omega$  é um mapa que associa cada lugar  $p_i \in P$  com uma saída  $\omega(p_i) \in \{0, 1, -\}^{|O|}$ , em que  $(-)$  significa ‘não importa’. Ele define os comandos a serem executados para cada marcação da rede, sendo representado na forma de vetor  $\omega(p_i) = [AADV, ARET, BADV, BRET]$  (Tabela 2.2).

Neste modelo, a árvore de alcançabilidade não apresenta nenhuma marcação em que a função  $\Omega$  é contraditória, portanto a função  $\omega$  é suficiente para definir os valores de saída.

A dinâmica da rede inicia-se com a leitura dos sinais de entrada e com o cálculo das transições habilitadas, lembrando que para o disparo de transições nas redes interpretadas não basta apenas a marcação da RdP, também é necessário que as condições apresentadas na Tabela 2.1 sejam verdadeiras. Quando do disparo das transições, a marcação é atualizada. Quando não houver mais transições habilitadas, os sinais de saída são calculados e, se necessário, serão executados os comandos apresentados na Tabela 2.2.

O disparo da transição  $t_7$  será usado como exemplo. Suponha que o lugar  $P3$  contém uma ficha, representando que ambos os cilindros possuem um bloco a sua frente. A leitura dos sensores de entrada verificará a condição dos blocos. Supondo que nenhum deles seja metálico, tanto a marcação da rede quanto a condição de disparo da transição  $t_7$  são verdadeiras, tornando-a habilitada para o seu disparo. A marcação da rede é atualizada, retirando uma ficha de  $P3$  e colocando-a em  $P7$ . Apesar de o lugar  $P7$  agora possuir uma ficha, a condição de interpretação da transição  $t_8$  ainda não é verdadeira. Com isso, não é possível dispará-la. Então, os sinais de saída do lugar  $P7$  são calculados e executados avançando o cilindro A com o comando AADV e impedindo o seu retorno com o comando ARET. Depois da execução dos sinais de saída os sinais de entrada são lidos novamente para a continuação da execução da dinâmica da rede.

Tabela 2.1 – Função das condições de disparo para cada transição

$\varphi(t_1)$	ST
$\varphi(t_2)$	nST
$\varphi(t_3)$	IND1 & IND2
$\varphi(t_4)$	A1 & B1
$\varphi(t_5)$	A0
$\varphi(t_6)$	B0
$\varphi(t_7)$	n(IND1 & IND2)
$\varphi(t_8)$	A1
$\varphi(t_9)$	B1
$\varphi(t_{10})$	A0 & B0

Fonte: Adaptado de BORGES; LIMA II (2018)

Tabela 2.2 – Estado de saída para cada lugar da RdPi

$\omega(P_1)$	[ -, -, -, -]
$\omega(P_2)$	[ -, -, -, -]
$\omega(P_3)$	[ -, -, -, -]
$\omega(P_4)$	[ 1, 0, 1, 0]
$\omega(P_5)$	[ 0, 1, -, -]
$\omega(P_6)$	[ -, -, 0, 1]
$\omega(P_7)$	[ 1, 0, -, -]
$\omega(P_8)$	[ -, -, 1, 0]
$\omega(P_9)$	[ 0, 1, 0, 1]
$\omega(P_{10})$	[ -, -, -, -]

Fonte: BORGES; LIMA II (2018)

Para que o CLP consiga executar os comandos é necessário associar os sinais de entrada, os sinais de saída e os lugares da RdPi aos endereços de entradas, saídas e memórias no CLP, como apresentado na Tabela 2.3.

Tabela 2.3 – Endereços de entrada, saída e memórias do CLP.

<b>Tipo</b>	<b>Nome</b>	<b>Endereço</b>	<b>Tipo</b>	<b>Nome</b>	<b>Endereço</b>
<b>Entradas</b>	A0	I0.0	<b>Lugares</b>	P1	M0.1
	A1	I0.1		P2	M0.2
	B0	I0.2		P3	M0.3
	B1	I0.3		P4	M0.4
	ST	I0.4		P5	M0.5
	IND1	I0.5		P6	M0.6
	IND2	I0.6		P7	M0.7
	<b>Saídas</b>	AADV		Q0.0	P8
ARET		Q0.1		P9	M1.1
BADV		Q0.2		P10	M1.2
BRET		Q0.3	-	-	

Fonte: BORGES; LIMA II (2018)

### 2.2.2.2 Sequential Function Chart IEC (2003) como rede de Petri interpretada

A norma IEC 848 de 1988 define Sequential Function Chart (SFC) como uma linguagem gráfica de programação em CLPs baseada em Grafcet, uma linguagem da década anterior (DAVID, 1995; SILVA, 2012). Em 1993, a norma IEC 61131-3 definiu cinco linguagens de programação para controladores, entre elas o SFC (MELLO et al., 2012), que será explicado nesta seção.

Os elementos do SFC proporcionam um meio de particionar e organizar programas de controladores com um conjunto de passos e transições conectados por linhas de conexão direcionadas. Um conjunto de ações é associado a cada lugar e um conjunto de condições é associado a cada transição (IEC, 2003).

Características do passo (norma IEC 61131-3 (IEC, 2003)):

- O formato do passo deve ser um retângulo ou quadrado;
- O passo inicial pode ser desenhado graficamente com linhas duplas nas bordas;
- Um passo é ativo ou inativo;
- Zero ou mais ações devem ser associadas a cada passo;
- Um passo de zero ações deve ser considerado com uma função ESPERAR (esperando que a condição da transição que o sucede se torne verdadeira);
- Saídas e entradas da unidade de organização do programa segue uma série de regras;
- O estado da unidade de organização do programa é definido por um conjunto de passos ativos e os valores das suas variáveis internas e de saída.

Características da transição (norma IEC 61131-3 (IEC, 2003)):

- Uma transição é representada por uma linha horizontal através de uma linha vertical de conexão direcionada;
- Cada transição deve ser associada a uma condição que é resultante da avaliação de uma única expressão booleana;
- A direção de evolução seguindo a linha vertical de conexão direcionada deve ser da parte inferior do(s) passo(s) predecessor(es) até o topo do(s) passo(s) sucessor(es);
- A condição de uma transição que é sempre verdadeira deve ser representada pelo símbolo 1 ou pela palavra *Verdadeiro*.

De acordo com a norma IEC 61131-3 (IEC, 2003), uma ação pode ser representada de diversas maneiras: como uma variável booleana, como uma coleção de instruções na linguagem IL, como uma coleção de afirmações na linguagem ST, como uma coleção de linhas na linguagem LD, como uma coleção de linhas/blocos na linguagem FBD, ou mesmo como um diagrama de função sequencial na linguagem SFC.

Regras de evolução da estrutura na linguagem SFC (norma IEC 61131-3 (IEC, 2003)):

- A situação inicial de uma rede SFC é caracterizada pelo passo inicial que está no estado ativo após a inicialização do programa ou bloco funcional que contém a rede;
- A evolução dos estados ativos dos passos deve ocorrer ao longo das linhas de conexão direcionadas quando causada pelo disparo de uma ou mais transições;
- Uma transição é habilitada quando todas os passos anteriores, conectados ao símbolo de transição correspondente por linhas de conexão direcionadas, estão ativos. O disparo de uma transição ocorre quando a transição está habilitada e quando a condição de transição associada é verdadeira;
- O disparo de uma transição causa a desativação de todos os passos imediatamente anteriores conectados ao símbolo de transição correspondente por linhas de conexão direcionadas, seguido pela ativação de todos os passos imediatamente seguintes;
- A alternância passo/transição e transição/passo devem sempre ser mantidas nas conexões dos elementos SFC, ou seja:
  - Dois passos nunca devem estar diretamente ligados; eles devem sempre ser separados por uma transição;
  - Duas transições nunca devem estar diretamente ligadas; elas devem sempre ser separadas por um passo.

De acordo com MELLO et al. (2012), há uma relação intrínseca entre os elementos da RdPi e da linguagem SFC, como mostrado na Tabela 2.4.

A linguagem SFC apresenta similaridades com a rede de Petri interpretada. Entretanto, DAVID (1995) destaca duas diferenças: (i) a marcação de um passo em Grafset é booleano, ou seja, a cada passo é associado o valor 0 (*Falso*) ou 1 (*Verdadeiro*), enquanto a marcação de um lugar em RdPi é numérico e pode ser representado por um inteiro não negativo de fichas (recursos). Na definição de FREY, que é uma rede segura, o valor máximo que a função peso pode

Tabela 2.4 – Relação entre RdPi e SFC.

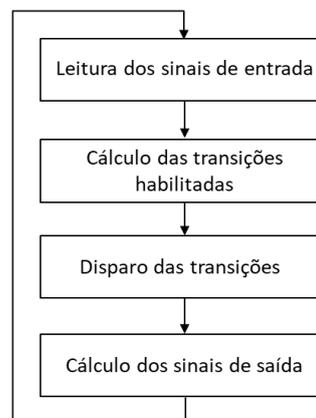
<b>Rede de Petri interpretada</b>	<b>SFC</b>
Lugar	Passo
Transição	Transição
Arco	Conector
Variáveis de entrada	Receptividade de cada transição
Variáveis de saída	Ação de cada passo

Fonte: Adaptado de MELLO et al. (2012)

assumir é unitário; (ii) em Grafset, todas as transições habilitadas simultaneamente disparam simultaneamente, enquanto em RdPi, se houver transições habilitadas em conflito apenas uma delas será disparada e seu comportamento será não-determinístico a menos que exista uma condição que selecione a transição a ser disparada. Apesar destas diferenças, se uma RdPi é segura (suas marcas comportam-se como variáveis lógicas) e determinística (se não houver transições em conflito que apresentem condições de disparo verdadeiras simultaneamente), ela pode ser representada na linguagem SFC.

Além dessas diferenças mencionadas acima, FREY (2000) pontua uma diferença no comportamento dinâmico dos estados. O fluxograma da Figura 2.5 mostra que os estados de uma RdPi podem ser instáveis (transitórios), já o fluxograma da Figura 2.8 mostra que na linguagem SFC esses estados são mantidos ativos por pelo menos um ciclo do CLP, também conhecido como ciclo de varredura. Entretanto, dada a curta duração do ciclo do CLP, esses estados podem ser vistos como quase transitórios.

Figura 2.8 – Fluxograma do funcionamento do SFC (ciclo de varredura).

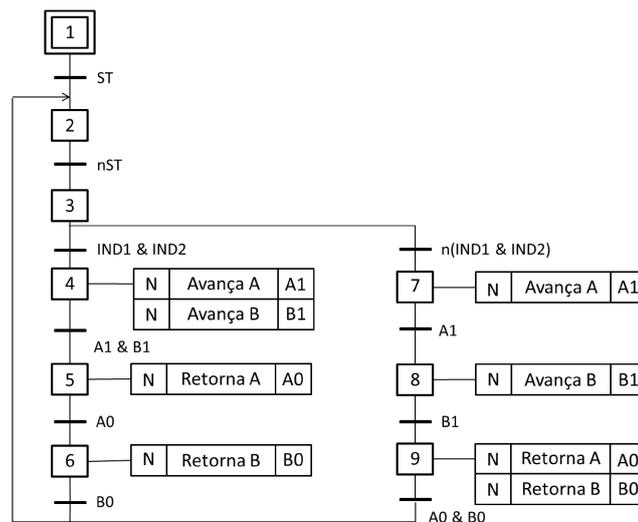


Fonte: Autor

### Exemplo de aplicação da notação

O mesmo exemplo apresentado na Seção 2.2.2.1 será agora apresentado na Figura 2.9 com a notação em linguagem SFC. Nesta figura, & significa a expressão lógica E, n significa *Falso*. Neste exemplo, a dinâmica é dada pelo ciclo de varredura. Para ocorrer uma transição, o passo anterior a ela deve estar marcado e as suas condições devem ser satisfeitas. Por exemplo, o passo inicial não está diretamente associado a nenhuma ação, portanto necessita de um sinal para ativar a transição seguinte (neste caso, o botão ST deve ser pressionado). Sendo assim, a transição será disparada e um ciclo do CLP acontecerá. Agora, será analisada a transição que conecta o passo 3 ao passo 7. Quando o passo 3 estiver marcado e os sensores indicarem que as peças não são metálicas, as condições estarão satisfeitas e a transição será executada ativando o passo 7. Este passo está associado a uma ação para avançar o cilindro A.

Figura 2.9 – Exemplo na linguagem SFC.



Fonte: Autor

### 2.2.3 Rede de Petri a objetos

Durante a pesquisa para este mestrado foram encontradas diversas notações (e nomenclaturas) de rede de Petri a objetos. Para citar algumas:

- LOOPN++ (LAKOS; KEEN, 1994);
- OPN (LAKOS, 1995a; YU; CAI, 2006; WU et al., 2016);
- Rede de Petri a objetos (CARDOSO; VALETTE, 1997);

- HOONet (HONG; BAE, 2001);
- OOPN-IDE (NIU; ZOU; REN, 2003);
- COOPN (AL-SHABIBI et al., 2003);
- OOPNM (MOTAMENI et al., 2008);
- OOUPN (SILVA et al., 2009);
- OOPN (MASRI; BOURDEAUD'HUY; TOGUYENI, 2009);
- HLOOPN (BREZOVAN et al., 2010);
- EHLPO (BREZOVAN et al., 2010);
- OOPN-SDD (KOCI; JANOUSSEK, 2013);
- OOCPPN (RAJABI; LEE, 2014);
- OOPNCM (KOCI; JANOUSSEK, 2015);
- OETPN (LETIA; AL-JANABI, 2018);
- EOOPN (ZHAO et al., 2019);
- HLPN (IEC, 2019).

Varias destas definições são muito complexas e são mais voltadas para a ciência da computação. Entretanto, a definição de CARDOSO; VALETTE (1997) foi escolhida para ser trabalhada justamente por possuir aplicações industriais em sistemas de manufatura e por conseguir representar as informações das fichas (material da peça, estado atual da peça, etc), a dinâmica da rede (sequência de operações) e os recursos necessários (máquinas, sistemas de transporte) para o funcionamento do sistema. Essas características e o formalismo da RdPo definida pelos autores CARDOSO; VALETTE permitem descrever a nova notação em RdPo, que será definida no Capítulo 4, e aplicá-la no estudo de caso que, nessa dissertação, será para realizar a separação de materiais recicláveis.

#### 2.2.3.1 Definição de CARDOSO; VALETTE (1997)

CARDOSO; VALETTE (1997) definiram redes de Petri orientadas a objeto baseando-se na definição de redes de Petri com estrutura de dados por SIBERTIN-BLANC (1985). De acordo com CARDOSO; VALETTE (1997), uma rede de Petri orientada a objeto (RdPo) é

definida por um vetor de 9 elementos,  $RdPo = (P, T, C_{lass}, V, Pre, Post, A_{tc}, A_{ta}, M_0)$ , em que:

- $C_{lass}$  é um conjunto finito de classes de objetos, eventualmente organizado em uma hierarquia e definindo para cada classe um conjunto de atributos;
- $P$  é um conjunto finito de lugares cujos tipos são dados por  $C_{lass}$ ;
- $T$  é um conjunto finito de transições;
- $V$  é um conjunto de variáveis cujos tipos são dados por  $C_{lass}$ ;
- $Pre$  é a função lugar precedente que a cada arco de entrada de uma transição faz corresponder uma soma formal de n-uplas de elementos de  $V$ ;
- $Post$  é a função lugar seguinte que a cada arco de saída de uma transição faz corresponder uma soma formal de n-uplas de elementos de  $V$ ;
- $A_{tc}$  é uma aplicação que a cada transição associa uma condição fazendo intervir as variáveis formais associadas aos arcos de entrada e aos atributos das classes correspondentes;
- $A_{ta}$  é uma aplicação que a cada transição associa uma ação fazendo intervir as variáveis formais associadas aos arcos de entrada e aos atributos das classes correspondentes;
- $M_0$  é a marcação inicial que associa a cada lugar uma soma formal de n-uplas de instâncias de objetos.

### 2.3 Conceitos de orientação a objetos

Os conceitos de orientação a objetos são fundamentais para entendermos a fundamentação das diversas redes de Petri a objetos. Segundo (LETIA; AL-JANABI, 2018), tais conceitos surgiram após as redes de Petri, portanto, precisam ser adaptados para serem usados no contexto das redes de Petri. Há duas tendências para fazer as adaptações: (i) focar na análise matemática, deixando o mais simples possível para facilitar a demonstração das propriedades; (ii) enriquecer a rede de Petri a objetos com o máximo de propriedades e noções de orientação a objetos, o que dificulta a demonstração das diferentes propriedades, mas aumenta o poder de descrição do modelo. Nessa sessão, serão apresentados os principais conceitos de orientação a objetos: classe, objeto, atributo, método, herança, polimorfismo, encapsulamento, abstração, modularização, mensagem.

Uma classe é definida por um conjunto de atributos (propriedades) e métodos, que podem ser vistos como funções com parâmetros e variáveis locais capazes de acessar variáveis implicitamente e modificar os valores dos atributos (CARDOSO; VALETTE, 1997; WERNER, 2009). Conceitualmente, as classes são apenas descrições de objetos, enquanto os objetos são instâncias da classe, ou seja, um objeto é uma entidade pertencente à classe com as propriedades que são definidas na e pela própria classe (GUERRERO, 2002).

Herança é definida como o processo de criação de uma nova classe com os atributos e métodos de uma classe já existente adicionando características únicas a esta nova classe. Essa propriedade é uma forte ferramenta para organização e estruturação do código. As classes herdeiras são conhecidas como subclasses e as classes herdadas são conhecidas como superclasses (MOTAMENI et al., 2008). Polimorfismo é quando diferentes classes possuem um método com o mesmo nome, mas este método gera ações/comportamentos diferentes no sistema (LAKOS, 1995b; ZHAO et al., 2019).

O encapsulamento é o processo de empacotar dados e métodos de um objeto, escondendo alguns detalhes de implementação para outros objetos protegendo-o de acessos externos. A proteção das informações internas de um objeto, conhecida como abstração, permite que as características de um objeto sejam modificadas sem que outras partes do programa sejam afetadas. Um objeto é composto por uma interface pública e por uma parte privada, onde ficam os dados internos e os métodos (MOTAMENI et al., 2008). Esta importante característica da orientação a objetos, conhecida como modularização, facilita a reusabilidade e um melhor entendimento do código (SIMON; RÖSCH, 2015). A estrutura na programação estruturada é contínua, ou seja, todas as funcionalidades estão contidas em um corpo de texto, e a atribuição das entradas determinam os acontecimentos. Enquanto isso, a programação orientada a objetos permite a separação do código para diferentes tarefas, os próprios nomes dos métodos já deixam mais claro o que está acontecendo no código (WERNER, 2009).

Um objeto é responsável por enviar e receber mensagens, que passam de um objeto a outro quando um método de um objeto chama o outro (ZHAO et al., 2019). Tais mensagens são enviadas pela interface do objeto emissor até encontrarem a interface do objeto de destino, onde será processada e executará algum método, caso necessário (XINHUA; YA-NI, 2016).

### 3 METODOLOGIA

Este capítulo é dedicado à formalização do problema utilizado como estudo de caso que servirá de motivação para a construção de uma definição formal de rede de Petri a objetos interpretada a partir dos conceitos e definições apresentados no Capítulo 2.

#### 3.1 Formalização do estudo de caso

Um esquema simplificado do estudo de caso é apresentado na Figura 3.1, onde é modelada uma planta industrial para polir (Estação de polimento) e separar (Estação de triagem) peças de diferentes materiais recicláveis (metal, madeira e plástico preto), que serão destinadas ao seu respectivo *buffer* final. *Buffer* é o termo técnico usado para definir um compartimento para armazenamento (de peças, no caso).

Figura 3.1 – Planta simplificada do estudo de caso



Fonte: Autor

Para separar os materiais em questão será necessária a combinação de sensores capacitivo (CAP), óptico (OPT) e indutivo (IND). O sensor CAP detecta a presença de peças no sistema, o sensor OPT é capaz de detectar peças que interrompem um fluxo de luz e o sensor IND é capaz de detectar objetos metálicos. Então, a lógica para a separação dos materiais é dada por:

- IND para o metal, pois ele é o único que responde *Verdadeiro* para o sensor indutivo;
- $OPT \ \& \ \overline{IND}$  para a madeira, pois ela responde *Verdadeiro* para o sensor óptico (interrompe o fluxo de luz) e responde *Falso* para o sensor indutivo (não é metálico);
- $CAP \ \& \ \overline{OPT} \ \& \ \overline{IND}$  para o plástico preto, pois apesar de todos responderem *Verdadeiro* para o sensor capacitivo (presença de peças), o plástico preto é o único dentre os materiais que só responde *Verdadeiro* para o sensor capacitivo.

A Tabela 3.1 sintetiza as combinações lógicas dos sensores desejadas para realizar a detecção de cada tipo de material.

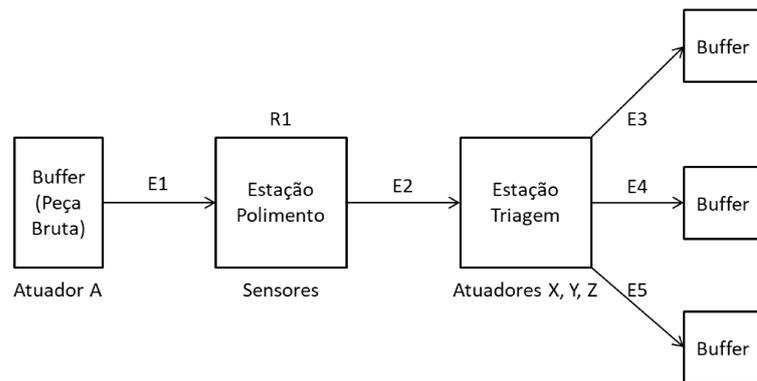
Tabela 3.1 – Lógica dos sensores para identificação dos materiais.

	Indutivo	Óptico	Capacitivo	Lógica de sensores
<b>Metal</b>	X	X	X	IND
<b>Madeira</b>		X	X	OPT & $\overline{\text{IND}}$
<b>Plástico preto</b>			X	CAP & $\overline{\text{OPT}}$ & $\overline{\text{IND}}$

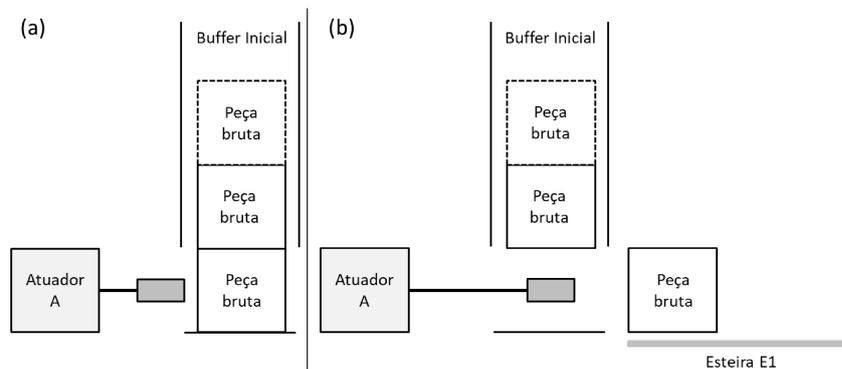
Fonte: Autor

Um esquema mais detalhado da planta é apresentado na Figura 3.2 para explicar o seu funcionamento. A Figura 3.3(a) mostra que todas as peças estão contidas em um *buffer* inicial em seu estado bruto (sem polimento). A peça localizada na parte superior do *buffer* está pontilhada indicando que novas peças podem chegar por cima. A peça localizada na parte inferior é empurrada para a esteira E1 por um atuador pneumático A, que está alinhado com a parte inferior do *buffer*.

Figura 3.2 – Planta detalhada do estudo de caso.



Fonte: Autor

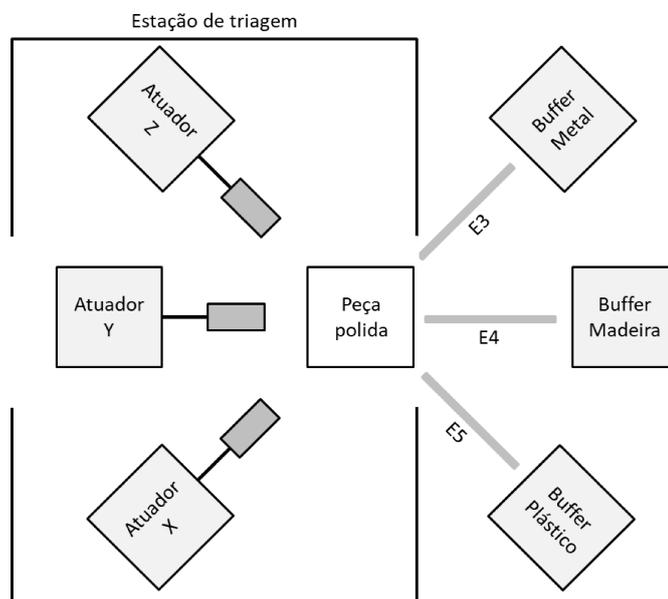
Figura 3.3 – Atuador movimentando a peça do (a) *buffer* inicial para a (b) esteira E1.

Fonte: Autor

Ao chegar na esteira E1, a peça percorre-a até a estação de polimento, onde o material da peça será conferido pela lógica dos sensores. Após a identificação da peça, o robô R1 executará o método *ligar/desligar* e conferirá qual ferramenta está acoplada em sua extremidade de trabalho e executará o método *trocar ferramenta* para adequar ao material a ser polido (se necessário). Caso já esteja com a ferramenta correta, o método *polir peça* será executado. Ao final deste método, o próprio robô R1 executará o método *empurrar peça* para a esteira E2, que leva à estação de triagem, e depois executará o método *ligar/desligar*.

Ao chegar na estação de triagem, a peça será distribuída para o seu respectivo *buffer* final de acordo com o material já identificado anteriormente. Essa ação será feita por um dos três atuadores pneumáticos X, Y, Z posicionados em direção às esteiras E3 (45°), E4 (0°) e E5 (−45°). A Figura 3.4 apresenta um esquema da vista superior com o posicionamento dos atuadores, das esteiras e dos *buffers* finais em relação à peça e às esteiras. Todos os atuadores pneumáticos possuem retorno por mola.

Figura 3.4 – Vista superior da estação de triagem e dos *buffers* finais.



Fonte: Autor

## 4 DESENVOLVIMENTO

Este capítulo é dedicado à definição formal e notação da rede de Petri a objetos interpretada (RdPoi). Como citado no Capítulo 1, essa definição se faz necessária porque em algumas situações uma RdP não é suficiente para representar um sistema. A rede de Petri interpretada (apresentada na Subseção 2.2.2), a rede de Petri a objetos (apresentada na Subseção 2.2.3) e os conceitos de orientação a objetos (apresentados na Seção 2.3) servirão de base para construir essa formalização.

### 4.1 Definição formal da Rede de Petri a Objetos Interpretada

A definição da RdPoi é um vetor de 16 elementos, tal que:

$$RdPoi = (P, T, F, W, I, O, \varphi, \omega, \Omega, C_{lass}, V, Pre, Post, A_{tc}, A_{ta}, M_0).$$

- $P = \{p_1, p_2, \dots, p_m\}$  é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos;
- $W: F \rightarrow \{1, 2, 3, \dots\}$  é a função peso dos arcos;
- $I$  é um conjunto de sinais lógicos de entrada com  $|I| > 0$ ;
- $O$  é um conjunto de sinais lógicos de saída com  $I \cap O = \emptyset, |O| > 0$ ;
- $\varphi$  é um mapa que associa cada transição  $t_i \in T$  com uma condição de disparo, tal que  $\varphi(t_i) =$  função Booleana em  $I$ ;
- $\omega$  é um mapa que associa cada lugar  $p_i \in P$  com uma saída  $\omega(p_i) \in \{0, 1, -\}^{|O|}$ , em que  $(-)$  significa ‘não importa’;
- $\Omega$  é uma função de saída que combina as saídas  $\omega$  de todos os lugares marcados, tal que  $\Omega: m \rightarrow \{-, 1, 0, c, r_0, r_1, c_0, c_1, c_{01}\}^{|O|}$ . A combinação da saída pode ser indefinida  $(-)$ , um  $(1)$ , zero  $(0)$ , contraditória  $(c)$ , redundante em zero ou em um  $(r_0, r_1)$ , ou uma combinação de contradição e redundância  $(c_0, c_1, c_{01})$ ;
- $C_{lass}$  é um conjunto finito de classes de objetos, eventualmente organizado em uma hierarquia e definindo para cada classe um conjunto de atributos;
- $V$  é um conjunto de variáveis cujos tipos são dados por  $C_{lass}$ ;

- $Pre$  é a função lugar precedente que a cada arco de entrada de uma transição faz corresponder uma soma formal de n-uplas de elementos de  $V$ ;
- $Post$  é a função lugar seguinte que a cada arco de saída de uma transição faz corresponder uma soma formal de n-uplas de elementos de  $V$ ;
- $A_{tc}$  é uma aplicação que a cada transição associa uma condição fazendo intervir as variáveis formais associadas aos arcos de entrada e aos atributos das classes correspondentes;
- $A_{ta}$  é uma aplicação que a cada transição associa uma ação fazendo intervir as variáveis formais associadas aos arcos de entrada e aos atributos das classes correspondentes;
- $M_0$  é a marcação inicial, que associa a cada lugar uma soma formal de n-uplas de instâncias de objetos.

Este vetor pode ser dividido em quatro partes: os elementos  $P$ ,  $T$ ,  $F$ ,  $W$  formam a estrutura e a parte gráfica da rede sem a marcação inicial; os elementos  $I$ ,  $O$ ,  $\varphi$ ,  $\omega$ ,  $\Omega$  dizem respeito à interpretação da rede; os elementos  $Class$ ,  $V$ ,  $Pre$ ,  $Post$ ,  $A_{tc}$ ,  $A_{ta}$  são relativos à orientação a objetos da rede; e o elemento  $M_0$  é a marcação inicial da rede.

A parte gráfica da rede evidencia a relação entre lugares  $P$  e transições  $T$  através de arcos direcionados  $F$  com a função peso  $W$  aplicada.

A interpretação da rede é dada pelos sinais lógicos de baixo nível de entrada  $I$  e de saída  $O$ . A função  $\varphi$  relaciona cada transição a uma condição de disparo, sendo que esta condição pode ou não ser satisfeita. A função  $\omega$  relaciona cada lugar com uma saída, que define os comandos a serem executados para cada marcação da rede. Já a função  $\Omega$  mostra a combinação das saídas  $\omega$  de todos os lugares marcados.

A orientação da rede é dada em alto nível pelos conceitos de orientação a objetos aplicados à notação de rede de Petri.  $Class$  define um conjunto de classes de objetos, sendo que para cada classe é definido um conjunto de atributos.  $V$  é um conjunto de variáveis (objetos instanciados) cujos tipos são dados por  $Class$ . As funções  $Pre$  (lugar precedente) e  $Post$  (lugar seguinte) correspondem a uma soma formal de n-uplas de elementos de  $V$  para os arcos de entrada e de saída de uma transição, respectivamente. Cada transição possui a função  $A_{tc}$  para avaliar as condições de disparo e a função  $A_{ta}$  para intervir nas variáveis formais dos atributos das classes correspondentes.

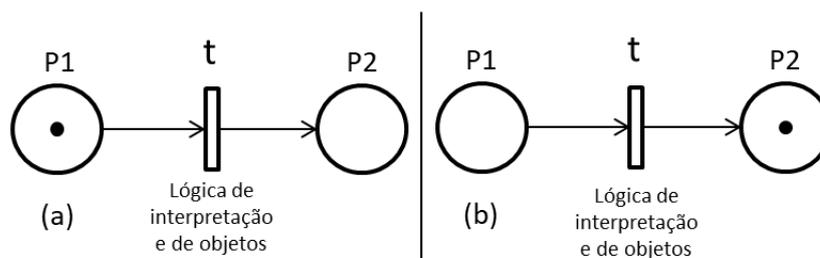
A marcação inicial  $M_0$  representa as fichas disponíveis em cada lugar no estado inicial da rede.

#### 4.1.1 Dinâmica da rede

O comportamento dinâmico da rede é dado por todos os elementos já definidos e pelas regras de disparo das transições. Para serem disparadas, as transições precisam estar habilitadas, ou seja, os lugares que precedem as transições deve estar ocupados. Além disso, também precisam que as condições da interpretação de sinais, se houver, sejam satisfeitas. Por último, em relação à orientação em objetos, a função  $A_{tc}$  verifica se as condições de atributos dos objetos são satisfeitas. Depois de tudo isso, a transição pode, enfim, ser disparada. O resultado do disparo de uma transição é a chamada da função  $A_{ta}$ , que envia mensagem para executar métodos capazes de mudar os atributos de um objeto em questão. Visualmente, o disparo da transição é a retirada das fichas dos pré-lugares e a colocação delas nos pós-lugares. Após uma transição ser disparada, o estado do sistema é atualizado e inicia-se uma nova verificação de quais transições estão habilitadas e passíveis de disparo.

A Figura 4.1 apresenta um desenho esquemático da transição da rede de Petri a objetos interpretada antes e depois do disparo da transição. Na Figura 4.1(a), é verificada a marcação das fichas e se as condições dos sinais de interpretação  $\varphi(t_i)$  e dos objetos  $A_{tc}(t_i)$  são satisfeitas. Vale ressaltar que as condições lógicas de interpretação dos sinais de baixo nível e as condições lógicas dos objetos aparecem em cada transição, se houver. Entretanto, as mensagens para execução de métodos disparados pelas transições são enviadas de forma que a alteração de atributos seja tratada internamente por cada objeto para impedir o acesso a detalhes da implementação por outros objetos. Com o disparo da transição, a ficha é transferida para o lugar  $P2$  com a mudança de atributos do objeto em questão, quando necessário. Um novo estado da rede é observado na Figura 4.1(b).

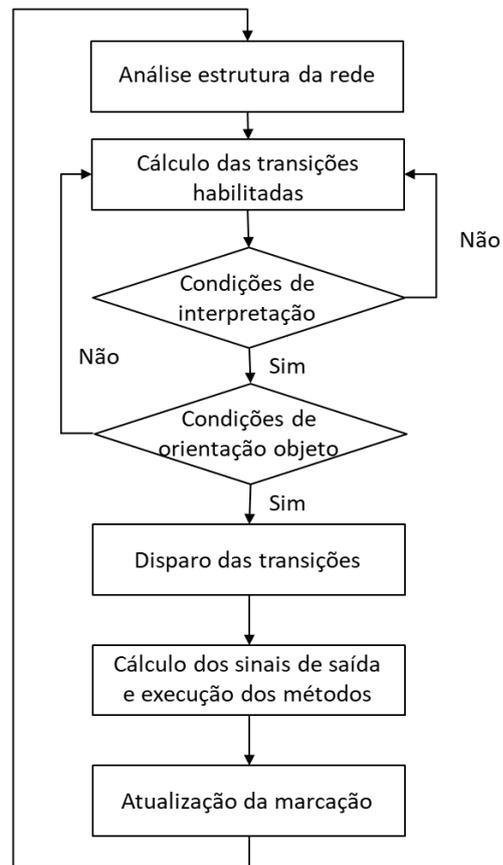
Figura 4.1 – Transição da rede de Petri a objetos interpretada: (a) antes e (b) depois do disparo da transição.



Fonte: Autor

Com o intuito de facilitar o entendimento do leitor, o fluxograma da Figura 4.2 sintetiza a evolução dinâmica da marcação da rede durante a evolução do processo.

Figura 4.2 – Dinâmica da Rede de Petri a objetos interpretada.



Fonte: Autor

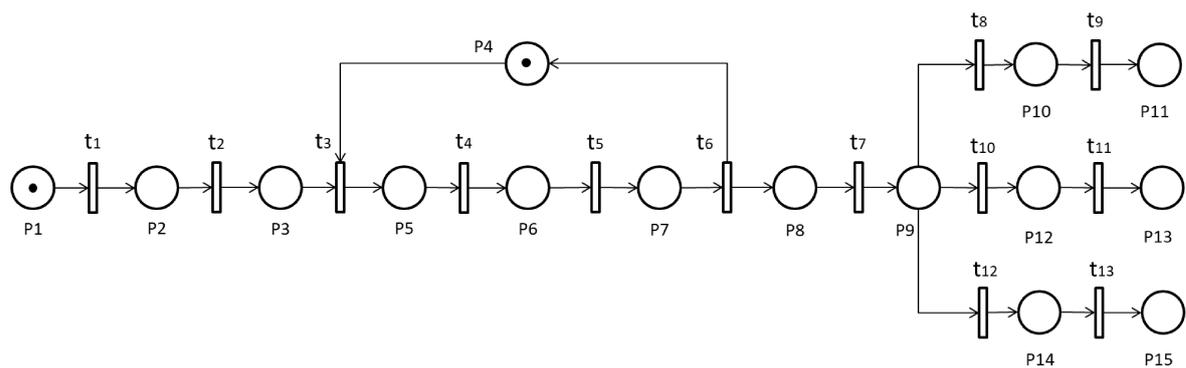
## 5 RESULTADOS E DISCUSSÕES

Neste capítulo, a notação formalizada no Capítulo 4 é aplicada no estudo de caso apresentado no Capítulo 3.

### 5.1 Aplicação da notação no estudo de caso proposto

A Figura 5.1 mostra a estrutura da rede de Petri a objetos interpretada para o estudo de caso, em que todos os arcos possuem peso unitário. Além disso, esta rede é  $k$ -limitada, em que  $k$  é o número de peças em estado bruto no lugar P1, que serão trabalhadas até chegarem uma a uma ao seu destino, representado pelos lugares P11, P13 e P15.

Figura 5.1 – Rede de Petri a objetos interpretada do estudo de caso.



Fonte: Autor

A Tabela 5.1 exibe o significado dos lugares na RdPoi. O lugar P1 representa o *buffer* inicial, onde estarão as peças sem polimento. O lugar P2 representa a esteira E1. O lugar P3 representa a estação de polimento com a peça em espera para a identificação do material pelos sensores. O lugar P4 é um recurso do sistema e representa o robô R1 dentro da estação de polimento, que executará os métodos necessários para alterar os atributos das peças. O lugar P5 representa a estação de polimento com o material da peça já identificado em espera para conferir se será necessário que o robô execute uma eventual troca de ferramenta. O lugar P6 representa a estação de polimento com a peça em espera para ser polida. O lugar P7 representa a estação de polimento com a peça em espera para ser retirada pelo robô em direção à esteira E2. O lugar P8 representa a esteira E2. O lugar P9 representa a estação de triagem, onde a peça será direcionada para o *buffer* final de acordo com o material da peça. Os lugares P10, P12 e

P14 representam as esteiras E3, E4 e E5 respectivamente, e direcionam cada peça ao seu *buffer* final. Os lugares P11, P13 e P15 representam os *buffers* finais para o armazenamento separado do metal, da madeira e do plástico preto indicando o fim do processo para cada peça e, além disso, servindo como um contador de quantas peças de cada material já foram finalizadas.

Tabela 5.1 – Significado dos lugares no estudo de caso.

P1	<i>Buffer</i> inicial
P2	Esteira E1
P3	Estação de polimento (identificação peça)
P4	Robô R1
P5	Estação de polimento (troca ferramenta)
P6	Estação de polimento (polir peça)
P7	Estação de polimento (empurrar peça)
P8	Esteira E2
P9	Estação de triagem
P10	Esteira E3
P11	<i>Buffer</i> final para o metal
P12	Esteira E4
P13	<i>Buffer</i> final para a madeira
P14	Esteira E5
P15	<i>Buffer</i> final para o plástico preto

Fonte: Autor

Considerando apenas a estrutura da rede, para que a dinâmica da rede funcione basta que os pré-lugares estejam marcados, o que pode ser verificado na Figura 5.1. É importante ressaltar que o lugar P4 está inicialmente marcado e é seguro e, portanto, possui apenas dois estados possíveis: livre, quando está marcado; e ocupado, quando está desmarcado. Entretanto, apenas a estrutura da rede não é capaz de modelar o sistema apresentado. Portanto, os parágrafos seguintes serão destinados a aplicar a notação de interpretação e de a orientação a objetos para este estudo de caso.

Na interpretação, os sinais de entrada são dados pelos três sensores para separação dos materiais e pelos sensores de fim de curso dos quatro atuadores pneumáticos com retorno por mola, representados por:  $I = [\text{IND}, \text{OPT}, \text{CAP}, \text{AO}, \text{A1}, \text{X0}, \text{X1}, \text{Y0}, \text{Y1}, \text{Z0}, \text{Z1}]$ . Os sinais de saída são comandos de avanço e retorno dos atuadores pneumáticos a serem executados:  $O = [\text{AADV}, \text{ARET}, \text{XADV}, \text{XRET}, \text{YADV}, \text{YRET}, \text{ZADV}, \text{ZRET}]$ .

A Tabela 5.2 exhibe a função  $\varphi(t_i)$  com a lógica de interpretação das condições de disparo das transições no estudo de caso. Para que a condição da transição  $t_1$  seja verdadeira é necessário que o atuador pneumático A esteja recuado, ou seja, que o sensor A0 esteja ativo.

A transição  $t_3$  verifica a lógica da combinação de sensores para a identificação do material da peça e só há três condições possíveis: IND (metal), OPT &  $\overline{\text{IND}}$  (madeira) CAP &  $\overline{\text{OPT}}$  &  $\overline{\text{IND}}$  (plástico preto). Para que a condição da transição  $t_8$  seja verdadeira é necessário que o atuador pneumático X esteja recuado, ou seja, que o sensor X0 esteja ativo. Analogamente, as transições  $t_{10}$  e  $t_{12}$  estão relacionadas com os atuadores pneumáticos Y e Z, respectivamente. As demais transições não precisam ler nenhum sinal de entrada, ou seja, não possuem condições de interpretação a serem verificadas.

Tabela 5.2 – Lógica de interpretação das condições de disparo das transições no estudo de caso.

$\varphi(t_1)$	A0
$\varphi(t_2)$	-
$\varphi(t_3)$	IND OU OPT & $\overline{\text{IND}}$ OU CAP & $\overline{\text{OPT}}$ & $\overline{\text{IND}}$
$\varphi(t_4)$	-
$\varphi(t_5)$	-
$\varphi(t_6)$	-
$\varphi(t_7)$	-
$\varphi(t_8)$	X0
$\varphi(t_9)$	-
$\varphi(t_{10})$	Y0
$\varphi(t_{11})$	-
$\varphi(t_{12})$	Z0
$\varphi(t_{13})$	-

Fonte: Autor

A Tabela 5.3 exhibe a função  $\omega(P_i)$  com a lógica das saídas dos atuadores pneumáticos no estudo de caso. Quando a transição  $t_1$  é disparada, a ação vinculada ao lugar P2 é executada avançando o atuador A e impedindo-o de recuar. O mesmo acontece com os atuadores X, Y e Z quando as transições  $t_8$ ,  $t_{10}$  e  $t_{12}$  são disparadas. Os outros lugares da rede não estão relacionados a nenhuma ação proveniente da interpretação de sinais.

Neste modelo, a árvore de alcançabilidade não apresenta nenhuma marcação em que a função  $\Omega$  é contraditória, portanto a função  $\omega$  é suficiente para definir os valores de saída.

Tabela 5.3 – Lógica dos sinais de saída dos atuadores pneumáticos no estudo de caso  $\omega(p_i) = [AADV, ARET, XADV, XRET, YADV, YRET, ZADV, ZRET]$ .

$\omega(P_1)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_2)$	[ 1, 0, -, -, -, -, -, - ]
$\omega(P_3)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_4)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_5)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_6)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_7)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_8)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_9)$	[ -, -, -, -, -, -, -, - ]
$\omega(P_{10})$	[ -, -, 1, 0, -, -, -, - ]
$\omega(P_{11})$	[ -, -, -, -, -, -, -, - ]
$\omega(P_{12})$	[ -, -, -, -, 1, 0, -, - ]
$\omega(P_{13})$	[ -, -, -, -, -, -, -, - ]
$\omega(P_{14})$	[ -, -, -, -, -, -, 1, 0 ]
$\omega(P_{15})$	[ -, -, -, -, -, -, -, - ]

Fonte: Autor

As classes *peça* e *robô* estão presentes em  $C_{class}$  e os objetos instanciados estão em  $V$ . Os atributos da classe *peça* são: (i) o material em que a peça é feito (metal, madeira, plástico preto); e (ii) o estado em que ela se encontra (bruto/polido). Já a classe *robô* possui os atributos: (i) o material da ferramenta para cortar a peça (metal, madeira, plástico preto); e (ii) o estado de funcionamento do robô (ligado/desligado). Além disso, a classe *robô* possui quatro métodos:

- *ligar/desligar* - para mudar o estado do robô;
- *trocar ferramenta* - para adequar o robô para executar o polimento da peça;
- *polir peça* - para mudar o estado da peça;
- *empurrar peça* - para levar a peça para a esteira E2 em direção à estação de triagem.

As funções *Pre* e *Post* relacionam uma soma formal de n-uplas de elementos de  $V$  com os arcos de entrada e saída das transições, respectivamente.

A Tabela 5.4 exibe a função  $A_{tc}$  com a lógica de orientação das condições de disparo das transições no estudo de caso. A transição  $t_3$  verifica se a condição do estado de funcionamento do robô é ligado. A transição  $t_4$  verifica se o material da ferramenta do robô condiz com o material da peça. A transição  $t_5$  verifica se a condição do estado da peça é bruto. A transição  $t_6$  verifica se a condição do estado da peça é polido e se a condição do estado de funcionamento

do robô é desligado. As demais transições não precisam verificar o status de atributos nem das peças e nem do robô, ou seja, não possuem condições de orientação a serem verificadas.

Tabela 5.4 – Lógica de orientação a objetos das condições de disparo das transições no estudo de caso.

$A_{tc}(t_1)$	-
$A_{tc}(t_2)$	-
$A_{tc}(t_3)$	Verificar se estado do robô é ligado
$A_{tc}(t_4)$	Verificar se ferramenta condiz com material da peça
$A_{tc}(t_5)$	Verificar se estado da peça é bruto
$A_{tc}(t_6)$	Verificar se estado da peça é polido <b>E</b> Verificar se estado do robô é desligado
$A_{tc}(t_7)$	-
$A_{tc}(t_8)$	-
$A_{tc}(t_9)$	-
$A_{tc}(t_{10})$	-
$A_{tc}(t_{11})$	-
$A_{tc}(t_{12})$	-
$A_{tc}(t_{13})$	-

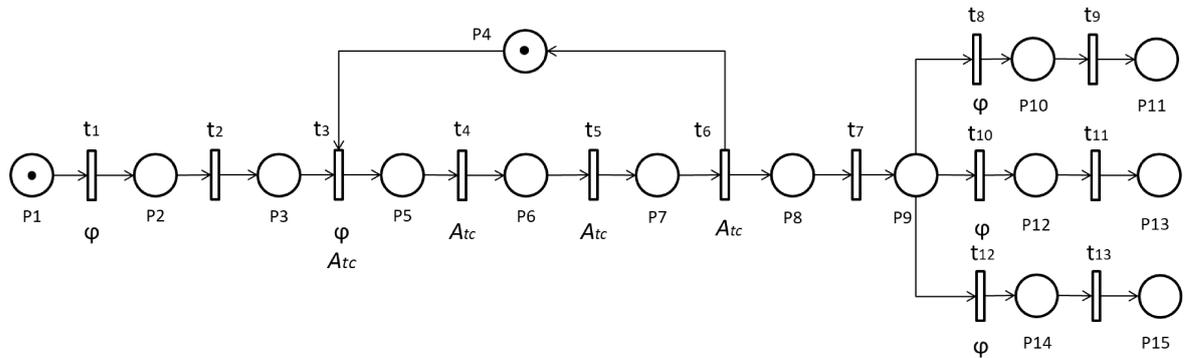
Fonte: Autor

Dadas todas as condições de interpretação e de orientação a objetos, as transições são responsáveis por aplicar a função  $A_{ta}$  para mandar mensagens para execução dos métodos *ligar/desligar*, *trocar ferramenta*, *polir peça* e *empurrar peça* para alterar os atributos dos objetos em questão sempre que necessário.

A marcação inicial da rede,  $M_0$ , é dada por um vetor em que cada elemento é uma soma formal dos objetos instanciados presentes em cada lugar no instante inicial.

A Figura 5.2 mostra a estrutura da rede de Petri a objetos interpretada do estudo de caso, agora com as funções de condição de disparo para cada transição, representadas pelas funções de interpretação  $\varphi(t_i)$  e de orientação a objetos  $A_{tc}(t_i)$ . Como todas as condições precisam ser satisfeitas para que as transições possam ser disparadas, a Tabela 5.5 exhibe tudo em um único lugar para facilitar a visualização.

Figura 5.2 – Rede de Petri com as condições de disparo das transições para o estudo de caso.



Fonte: Autor

Tabela 5.5 – Funções das lógicas de interpretação e de orientação a objetos das condições de disparo das transições no estudo de caso.

Transição	interpretação, $\varphi(t_i)$	orientação a objetos, $A_{tc}(t_i)$
$t_1$	A0	-
$t_2$	-	-
$t_3$	IND OU OPT & $\overline{\text{IND}}$ OU CAP & $\overline{\text{OPT}}$ & $\overline{\text{IND}}$	Verificar se estado do robô é ligado
$t_4$	-	Verificar se ferramenta condiz com material da peça
$t_5$	-	Verificar se estado da peça é bruto
$t_6$	-	Verificar se estado da peça é polido <b>E</b> Verificar se estado do robô é desligado
$t_7$	-	-
$t_8$	X0	-
$t_9$	-	-
$t_{10}$	Y0	-
$t_{11}$	-	-
$t_{12}$	Z0	-
$t_{13}$	-	-

Fonte: Autor

## 6 CONCLUSÃO

Este capítulo serve para destacar a importância da rede de Petri, que ainda é estudada em diversas áreas do conhecimento como os sistemas a eventos discretos (SED), por exemplo. Esta ferramenta permitiu a representação de sistemas mais complexos que antes não podiam ser descritos por causa da explosão de estados dos autômatos de estados finitos. Com isso, passou a ser empregada em aplicações industriais e sistemas de manufaturas, como o estudo de caso apresentado nesta dissertação.

Novos tipos de rede surgiram para agregar mais informações à rede, como a rede de Petri interpretada e a rede de Petri a objetos. A interpretação permite a leitura de sinais de entrada e saída em baixo nível, que são processados em CLPs, usados na indústria para executar diversas ações. A orientação a objetos teve seus conceitos posteriormente incluídos na teoria de redes, possibilitando que as fichas sejam tratadas como objetos individuais com seus atributos e métodos, comportando-se como em uma linguagem de programação de alto nível.

Neste trabalho, foi desenvolvida uma nova rede de Petri: a rede de Petri a objetos interpretada (RdPoi). Com a notação da RdPoi foi possível usar a interpretação de sinais de sensores em fichas que são entidades portadoras de informação usando os conceitos de orientação a objetos para executar métodos e modificar atributos dos objetos. A ferramenta desenvolvida neste trabalho foi fundamental para conseguir descrever de forma adequada o sistema apresentado no estudo de caso.

## Trabalhos futuros

Apesar de a definição formal de rede de Petri a objetos interpretadas (RdPoi) ter sido satisfatória para descrever o estudo de caso presente nesta dissertação, há trabalhos futuros que devem ser realizados para aprofundar os estudos e, futuramente, consolidá-la. Alguns possíveis trabalhos são listados abaixo:

- Implementação do estudo de caso em linguagens de alto nível como Arduino, Java, C++;
- Escrever este estudo de caso na linguagem SFC para implementação em um CLP;
- Testar a notação da RdPoi em outros estudos de caso para provar sua versatilidade;
- Tratamento matemático mais formal e adaptações, quando for necessário para descrever alguma situação não prevista, como a temporização e o comportamento estocástico da rede.

## REFERÊNCIAS

- AL-SHABIBI, A. et al. Prototyping object oriented specifications. In: INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF PETRI NETS. **Anais...** [S.l.: s.n.], 2003. p.473–482.
- BORGES, M. U.; LIMA II, E. J. Conversão de Rede de Petri Interpretada em Linguagem Ladder - Metodologia e Exemplo. In: XLV CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA (COBENGE), Joinville-SC. **Anais...** [S.l.: s.n.], 2017.
- BORGES, M. U.; LIMA II, E. J. Conversion Methodologies from Signal Interpreted Petri Nets to Ladder Diagram and C language in Arduino. **International Journal of Mechanical Engineering Education**, [S.l.], 2018.
- BRAUER, W.; REISIG, W. Carl adam Petri and “Petri nets”. In: **Fundamental Concepts in Computer Science**. [S.l.]: World Scientific, 2009. p.129–139.
- BREZOVAN, M. et al. High-Level Petri Nets with Object-Orientation—A High-Level Petri Net Model Allowing Multiple Inheritance and Multiple Polymorphism. In: **Intelligent Distributed Computing IV**. [S.l.]: Springer, 2010. p.219–228.
- CARDOSO, J.; VALETTE, R. Redes de petri. **Book**, [S.l.], 1997.
- DAVID, R. Grafcet: a powerful tool for specification of logic controllers. **IEEE Transactions on control systems technology**, [S.l.], v.3, n.3, p.253–268, 1995.
- FREY, G. Automatic implementation of Petri net based control algorithms on PLC. **Proceedings of the IEEE American Control Conference (ACC)**, [S.l.], v.4, n.June, p.2819–2823, 2000.
- FREY, G. **Design and formal Analysis of Petri Net based Logic Controllers**. 2002. Dissertação (Mestrado em Ciência da Computação) — Dissertation, Shaker Verlag, Aachen, Germany.
- GUERRERO, D. D. S. **Redes de Petri orientadas a objetos**. 2002. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Campina Grande.

HONG, J.-E.; BAE, D.-H. Incremental scenario modeling using hierarchical object-oriented petri net. **International journal of software engineering and knowledge engineering**, [S.l.], v.11, n.03, p.357–386, 2001.

IEC. **61131-3**: programmable controllers-part 3: programming languages. Geneva, CH: International Organization for Standardization and International Electrotechnical Commission, 2003. Standard.

IEC. **Systems and Software Engineering - High-Level Petri Nets - Part 1**: concepts, definitions and graphical notation. Geneva, CH: International Organization for Standardization and International Electrotechnical Commission, 2019. Standard.

KLEIN, S.; FREY, G.; MINAS, M. PLC programming with signal interpreted Petri nets. In: INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF PETRI NETS. **Anais...** [S.l.: s.n.], 2003. p.440–449.

KOCI, R.; JANOUŠEK, V. **Object oriented petri nets in software development and deployment**. [S.l.]: ICSEA, 2013.

KOCI, R.; JANOUŠEK, V. The Object Oriented Petri Net Component Model. In: THE TENTH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING ADVANCES. XPERT PUBLISHING SERVICES. **Anais...** [S.l.: s.n.], 2015. p.309–315.

LAKOS, C. A. **The object orientation of object Petri nets**. [S.l.]: Department of Computer Science, University of Tasmania, 1995.

LAKOS, C. A.; KEEN, C. D. **LOOPN++**: a new language for object-oriented petri nets. [S.l.]: Department of Computer Science, University of Tasmania, 1994.

LAKOS, C. From coloured Petri nets to object Petri nets. In: INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF PETRI NETS. **Anais...** [S.l.: s.n.], 1995. p.278–297.

LETIA, T. S.; AL-JANABI, D. Object enhanced time Petri net models. In: IEEE INTERNATIONAL CONFERENCE ON AUTOMATION, QUALITY AND TESTING, ROBOTICS (AQTR), 2018. **Anais...** [S.l.: s.n.], 2018. p.1–6.

LIMA II, E. J. **Uma Metodologia Para a Implementação Através de CLPs de Controle Supervisório de Células de Manufatura Utilizando Redes de Petri**. 2002. Dissertação de mestrado — Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal da Bahia (UFBA).

MASRI, A.; BOURDEAUD'HUY, T.; TOGUYENI, A. A component modular modeling approach based on object oriented Petri nets for the performance analysis of distributed discrete event systems. In: FIFTH INTERNATIONAL CONFERENCE ON NETWORKING AND SERVICES, 2009. **Anais...** [S.l.: s.n.], 2009. p.222–227.

MELLO, A. T. F. de et al. A transcription tool from Petri net to CLP programming languages. In: ABCM SYMPOSIUM SERIES IN MECHATRONICS—VOL. 5, SECTION IV—INDUSTRIAL INFORMATICS, DISCRETE AND HYBRID SYSTEMS. **Anais...** [S.l.: s.n.], 2012.

MOTAMENI, H. et al. Analysis software with an object-oriented petri net model. **World Applied Sciences Journal**, [S.l.], v.3, n.4, p.565–576, 2008.

MURATA, T. Petri Nets : properties , analysis and applications. ..., [S.l.], v.77, n.4, p.541–580, 1989.

NIU, J.; ZOU, J.; REN, A. OOPN: object-oriented petri nets and its integrated development environment. In: SOFTWARE ENGINEERING AND APPLICATIONS, SEA. **Proceedings...** [S.l.: s.n.], 2003.

PETRI, C. A. **Kommunikation mit Automaten**. 1962. 128p. v.Doktor.

PETRI, C. A. Grundsätzliches zur Beschreibung diskreter Prozesse. In: COLLOQUIUM ÜBER AUTOMATENTHEORIE, 3. **Anais...** [S.l.: s.n.], 1967. p.121–140.

PETRI, C. A. Introduction to General Net Theory. ..., [S.l.], p.1–39, 1980.

RAJABI, B.; LEE, S. P. Consistent Integration between Object Oriented and Coloured Petri Nets Models. **The International Arab Journal of Information Technology**, [S.l.], v.11, n.4, p.406–415, 2014.

RAMÍREZ-TREVIÑO, A.; RIVERA-RANGEL, I.; LÓPEZ-MELLADO, E. Observability of discrete event systems modeled by interpreted Petri nets. **IEEE Transactions on Robotics and Automation**, [S.l.], v.19, n.4, p.557–565, 2003.

SALDHANA, J.; SHATZ, S. M. Uml diagrams to object petri net models: an approach for modeling and analysis. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING. **Anais...** [S.l.: s.n.], 2000. p.103–110.

SIBERTIN-BLANC, C. High level Petri nets with data structure. **6th European Workshop on Application and Theory of Petri Nets**, [S.l.], 1985.

SILVA, J. R. et al. Introducing object-orientation in unified Petri Net approach. In: INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, GRAMADO, BRAZIL, 20. **Proceedings...** [S.l.: s.n.], 2009. v.4, p.451–459.

SILVA, M. 50 years after the PhD thesis of Carl Adam Petri: a perspective. **IFAC Proceedings Volumes (IFAC-PapersOnline)**, [S.l.], n.1976, p.13–20, 2012.

SIMON, T.; RÖSCH, S. Comparing the object-oriented extension with the classical IEC 61131-3 regarding reusability and understandability - A case study. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION, ETFA. **Anais...** [S.l.: s.n.], 2015. v.2015-October.

SKÖLDSTAM, M.; ÅKESSON, K.; FABIAN, M. Modeling of discrete event systems using finite automata with variables. In: IEEE CONFERENCE ON DECISION AND CONTROL. **Proceedings...** [S.l.: s.n.], 2007. p.3387–3392.

WERNER, B. Object-oriented extensions for IEC 61131-3. **IEEE Industrial Electronics Magazine**, [S.l.], v.3, n.4, p.36–39, 2009.

WU, K. et al. A Method for Describing Industrial Control Network Attack Using Object Petri Net. **IEEJ Transactions on electrical and electronic engineering**, [S.l.], p.216–227, 2016.

XINHUA, C.; YA-NI, S. Computer network simulation modeling based on an object oriented Petri net. **International Journal of Online Engineering**, [S.l.], v.12, n.2, p.25–28, 2016.

YU, Z.; CAI, Y. Object-oriented Petri nets based architecture description language for multi-agent systems. **IJCSNS**, [S.l.], v.6, n.1, p.123–131, 2006.

ZHAO, N. et al. An extended object-oriented petri net supporting the description and evolution of components: eoopn. **Cluster Computing**, [S.l.], v.22, p.2701–2708, mar 2019.