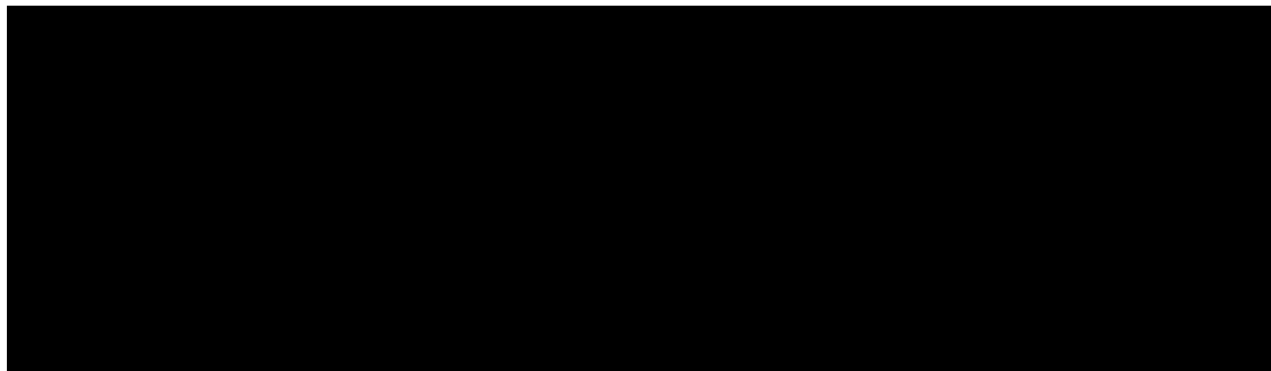


A EVOLUÇÃO DAS REDES DE PETRI INTERPRETADAS: UMA VISÃO EM TRÊS DEFINIÇÕES



Resumo: Este trabalho busca retratar a evolução da ferramenta criada por Carl Adam Petri em sua tese de doutorado em 1962. Primeiramente, surgiu apenas a base matemática voltada para a ciência da computação e, alguns anos mais tarde, o próprio Petri criou uma notação gráfica. Ao longo do tempo, e já se foram mais de cinquenta anos, pesquisadores se debruçaram sobre a tese e, assim, surgiram vários tipos de redes de Petri com suas especificidades para poder representar alguns problemas mais detalhadamente, como a descrição de sistemas a eventos discretos (SED), cuja evolução depende apenas do acontecimento de eventos (e não do tempo). Um desses tipos é a rede de Petri interpretada (RdPi), que será apresentada em três definições de diferentes autores. Posteriormente, será feita uma comparação das suas notações e a análise de suas características.

Palavras-chave: Carl Adam Petri. Rede de Petri. Rede de Petri interpretada.

1 INTRODUÇÃO

Na década de 1960, mais precisamente em 1962, o cientista alemão Carl Adam Petri defendeu a sua tese de doutorado, nomeada “Kommunikation mit Automaten” (PETRI, 1962). Nela constava uma parte matemática de base axiomática, que definiu formalmente alguns conceitos como: concorrência, sincronização, cooperação e competição. Petri substituiu o conceito de ordem temporal para ordem causal. Com isso, redes de Petri podem refletir o sistema a ser modelado (SILVA, 2012).

Silva (2012) publicou um artigo sobre os cinquenta anos da defesa da tese de Petri descrevendo a trajetória dessa importante ferramenta, que inicialmente era voltada para a ciência da computação. A ideia da tese foi empregada mais tarde em automação e em sistemas a eventos discretos (SED). Muitos autores citaram a tese de Petri apenas por usar sua ferramenta. Entretanto, a formulação de parte gráfica das redes de Petri (RdP) surgiu apenas três anos depois em uma palestra de Petri chamada “Fundamentals on the description of discrete processes”, ocorrida no 3º Colóquio sobre Teoria dos Autômatos em Hanover (BRAUER E REISIG, 2009; PETRI, 1967), e não neste trabalho, como muitos erroneamente acreditam (SILVA, 2012).

Ao longo de todos esses mais de cinquenta anos desde que a tese de Petri foi publicada, pesquisadores se debruçaram sobre ela e, assim, surgiram vários tipos de redes de Petri com suas especificidades para poder representar alguns problemas mais detalhadamente, como a descrição de sistemas a eventos discretos (SED), cuja evolução depende apenas do acontecimento de eventos (e não do tempo). Um desses tipos é a rede de Petri interpretada (RdPi). Com tantas definições diferentes de RdPi, esse artigo mostra uma discussão sobre as diferenças entre elas para embasar o professor a fim de que ele possa escolher qual a melhor forma de nortear seus alunos no estudo de engenharia a respeito de suas aplicações, e definir a melhor representação para cada tipo de problema.

2 REDES DE PETRI INTERPRETADAS

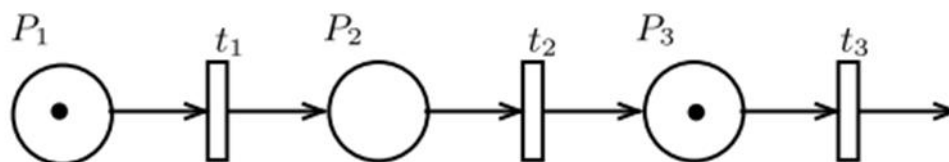
Nas subseções deste tópico serão definidos alguns tipos de RdPi tendo como referência a definição de rede de Petri dada por Murata (1989). Serão analisadas as definições de SFC (1993), Frey (2000) e Ramirez-Treviño et al (2003), respectivamente.

A rede de Petri (RdP) ordinária foi definida por Murata (1989) como um vetor 5-uplo $RdP=(P, T, F, W, M_0)$, em que:

- $P=\{p_1, p_2, \dots, p_n\}$ 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 $P \cup T \neq \emptyset$.

Os elementos gráficos da rede de Petri ordinária serão definidos com base na Figura 1. Segundo Petri (1980), o retângulo (transição) representa atividades de produção como transporte, acoplamento e desacoplamento de peças; já o círculo representa lugares nos quais recursos podem ser alocados temporariamente. Os arcos representam uma relação direcionada de acessibilidade imediata entre lugares e transições. Entretanto, arcos não são canais pelos quais os recursos podem fluir. Todos os arcos possuem um número (inteiro positivo) associado, quando este número é unitário sua escrita pode ser omitida (MURATA, 1989). Este número pode indicar a quantidade de recursos necessários (Figura 2), ou mesmo, alguma condição real (Figura 3) para disparar uma transição. As fichas são representadas por pontos pretos dentro dos lugares. Para representar a distribuição de fichas na rede (marcação da rede), é definido um vetor M de m posições, em que m é o número de lugares da RdP. $M(p)$ é o número de fichas no p -ésimo lugar.

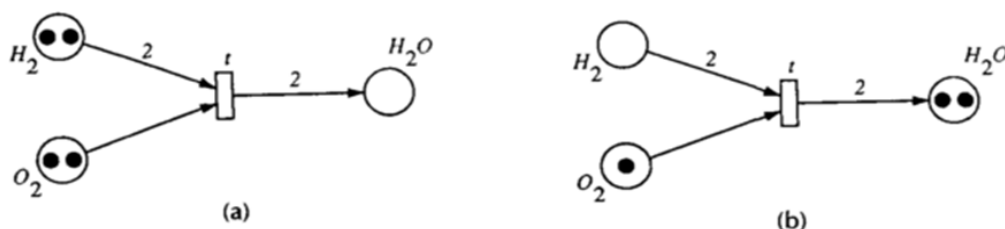
Figura 1 - Elementos de uma rede de Petri



Fonte: Adaptado de Cardoso e Valette (1997)

A Figura 2 apresenta uma rede de Petri de uma equação química. Neste caso, precisa-se de duas moléculas de H_2 e uma de O_2 para a formação de duas moléculas de H_2O .

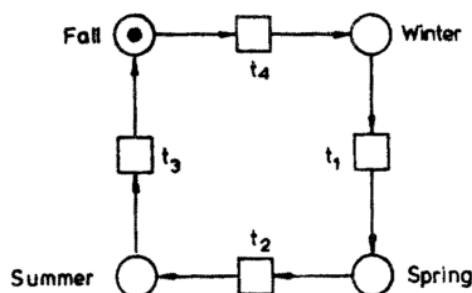
Figura 2 - Transição, lugares, arcos, peso e marcação da rede: (a) antes e (b) depois da transição



Fonte: Adaptado de Murata (1989)

Já a Figura 3 representa uma condição da passagem das estações ao longo dos anos. Tal rede poderia ser escrita de forma aberta, em linha. Entretanto, por ser uma condição cíclica, pode ser descrita de uma forma mais fácil de entender e visualizar (PETRI, 1980).

Figura 3 - Rede de Petri das estações do ano



Fonte: Petri (1980)

Segundo Cardoso e Valette (1997) e Silva (2013), uma rede de Petri é dividida em duas partes: controle e dados. O controle descreve as potenciais evoluções do processo, sendo bem descrita pela rede de Petri não interpretada. Já os dados, descrevem a estrutura interna do sistema e os cálculos feitos, incluindo as informações recebidas do meio externo, como: sensores e atuadores, objetos físicos, recursos e outras informações. Interpretar uma rede significa justamente receber sinais de entrada (associados às transições), realizar cálculos e apresentar sinais de saída (associados aos eventos), dando sentido (interpretação) aos elementos da rede (lugares e transições).

2.1 Definição de Sequential Function Chart (SFC)

A norma IEC 848 de 1988 definiu a linguagem Sequential Function Chart (SFC), uma linguagem gráfica para programação em um Controlador Lógico Programável (CLP) baseada na linguagem Grafset, da década anterior (DAVID, 1995; SILVA, 2012). Em 1993, a norma IEC 61131-3 definiu cinco linguagens de programação para controladores, dentre elas a linguagem SFC (DE MELLO et al., 2012).

Os elementos da linguagem SFC proporcionam um meio de particionar e organizar um programa de um controlador programável em um conjunto de passos e transições conectados por linhas de conexão diretas. Um conjunto de ações é associado a cada passo e um conjunto de condições é associado a cada transição (IEC, 2003), conforme definições a seguir:

Passo: Um passo representa uma situação na qual o comportamento de uma unidade de organização do programa com relação a suas entradas e saídas segue um conjunto de regras definidas pelas ações associadas do passo. Um passo é ativo ou inativo. Em qualquer momento, o estado da unidade organizacional do programa é definido pelo conjunto de etapas ativas e os valores de suas variáveis internas e de saída; (IEC, 2003)

Transição: Uma transição representa a condição em que o controle passa de um ou mais passos que precedem a transição para um ou mais passos sucessores ao longo do link direcionado correspondente. A transição deve ser representada por uma linha horizontal através do link direcionado vertical. A direção da evolução seguindo os links direcionados deve ser da parte inferior dos passos predecessores até o topo dos passos sucessores. Cada transição deve ter uma condição de transição associada, que é o resultado da avaliação de uma única expressão booleana. Uma condição de transição que é sempre verdadeira deve ser representada pelo símbolo 1 ou pela palavra TRUE.; (IEC, 2003)

Ação: Zero ou mais ações devem ser associadas a cada passo. Um passo que tenha zero ação associada deve ser considerado como tendo uma função WAIT, ou seja, esperando que uma condição de transição sucessora se torne verdadeira. Uma ação pode ser uma variável booleana, uma coleção de instruções na linguagem IL (...), uma coleção de instruções na linguagem ST (...), uma coleção de degraus na linguagem LD (...), uma coleção de redes na linguagem FBD (...), ou um gráfico de função sequencial (SFC). (IEC, 2003)

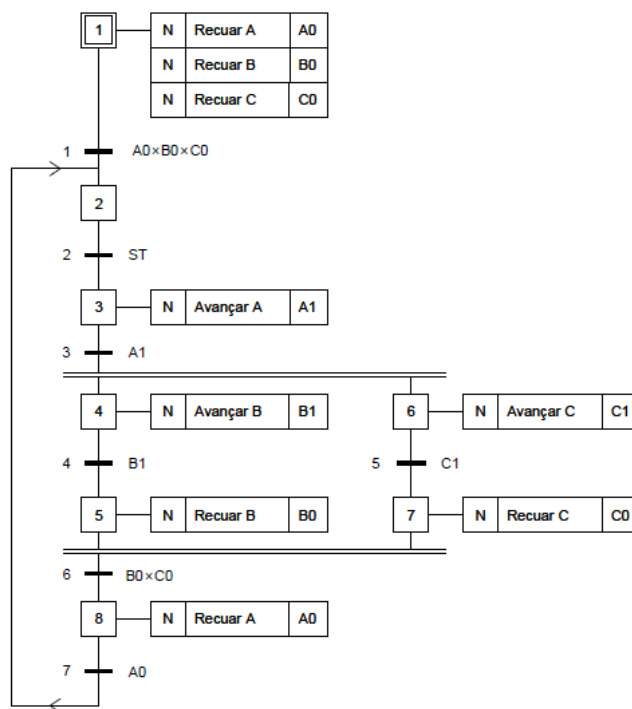
As regras de evolução de uma estrutura em SFC também foram definidas pela norma IEC 61131-3 (IEC, 2003) da seguinte forma:

A situação inicial de uma rede SFC é caracterizada pelo passo inicial que está no estado ativo na inicialização do programa ou bloco de funções que contém a rede. As evoluções dos estados ativos dos passos devem ocorrer ao longo das conexões direcionadas quando causados pela disparo de uma ou mais transições. Uma transição é ativada quando todos os passos anteriores, conectados ao símbolo de transição correspondente por conexões direcionadas, estão ativos. O disparo de uma transição ocorre quando a transição é ativada e quando a condição de transição associada é verdadeira. O disparo de uma transição faz com que a desativação (ou "reinicialização") de todos os passos imediatamente anteriores seja conectada ao símbolo de transição correspondente por conexões direcionadas, seguida pela ativação de todas as etapas imediatamente seguintes. A alternância passo/transição e transição/passo devem ser sempre mantidas nas conexões do elemento SFC, ou seja: dois passos nunca devem estar diretamente ligados; eles sempre serão separados por uma transição; duas transições nunca serão diretamente ligadas; elas sempre serão separadas por um passo. (IEC, 2003)

A Figura 4 ilustra os elementos da linguagem SFC. Os passos são representados graficamente por um retângulo, sendo que o passo inicial são dois retângulos. As ações estão associadas diretamente aos lugares, sendo que os lugares que não tiverem nenhuma ação ficam em modo de espera e necessitam de algum sinal para que a transição seguinte seja ativada (no exemplo abaixo, seria pressionar o botão ST). As transições são traços horizontais

que estão entre dois passos, que ao serem acionadas mudam o estado do sistema do passo anterior para o posterior.

Figura 4 – Exemplo da linguagem SFC



Fonte: Lima II (2007)

A Tabela 1, mostra a relação entre os elementos da RdPi e da linguagem SFC.

Tabela 1 - Relação entre elementos da RdP e SFC

Rede de Petri	SFC
Lugar	Passo
Transição	Transição
Arco	Conexão
Variáveis de entrada	Receptividade de cada transição
Variáveis de saída	Ação de cada passo

Fonte: Adaptada de de Mello et al (2012)

2.2 Definição de Frey (2000)

A RdPi de Frey (2000) pode ser definida por um vetor de nove termos representado por $RdPi = (P, T, F, M_0, I, O, \phi, \omega, \Omega)$. Em sua notação, os primeiros quatro termos representam a rede de Petri ordinária, enquanto os últimos cinco termos definem a interpretação da rede. Tais elementos são definidos como:

- I é um conjunto de sinais lógicos de entrada com $|I| > 0$;
- O é um conjunto de sinais lógicos de saída com $I \cap O = \emptyset$, $|O| > 0$;
- ϕ é um mapa que associa cada transição $t_i \in T$ com uma condição de disparo, tal que $\phi(t_i)$ = função booleana em I ;

- ω é um mapa que associa cada lugar $p_i \in P$ com uma saída $\omega(p_i) \in (0, 1, -)$, em que $(-)$ significa 'não importa';
- Ω é uma função de saída que combina as saídas ω de todos os lugares marcados, tal que $\Omega: m \rightarrow (-, 1, 0, c, r_0, r_1, c_0, c_1, c_{01})$. 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}) .

Um conceito importante no artigo de Frey (2000) é o de rede segura. Uma rede é dita segura quando os seus lugares podem apresentar, no máximo, uma ficha por vez. Essa consideração traz duas consequências imediatas: não é preciso definir o peso dos arcos, pois serão todos unitários; a transição é habilitada apenas se não houver fichas nos pós-lugares. Com isso, simplifica-se o código a ser escrito reduzindo o seu tamanho.

Quando uma transição é ativada, retira-se uma ficha de todos os lugares anteriores à transição (pré-lugares desmarcados) e coloca-se uma ficha em todos os lugares posteriores à transição (pós-lugares marcados), desde que os lugares estejam conectados a ela. Para que uma transição seja disparada, foi definido um conjunto de quatro regras a serem satisfeitas:

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 a função Ω à marcação. (FREY, 2000)

2.3 Definição de Ramírez-Treviño et al (2003)

A definição de RdPi de Ramírez-Treviño et al (2003), parte de uma definição de RdP diferente da apresentada por Murata (1989). Os elementos da rede são formados por uma quadrupla $G=(P, T, I, O)$, em que:

- $P=\{p_1, p_2, \dots, p_n\}$ é um conjunto finito de lugares;
- $T=\{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $I:P \times T \rightarrow \mathbb{Z}^+$ é a função que representa o peso dos arcos dos lugares para as transições;
- $O:P \times T \rightarrow \mathbb{Z}^+$ é a função que representa o peso dos arcos das transições para os lugares.

A rede é definida por uma dupla $N=(G, M_0)$, em que M_0 é a marcação inicial.

A interpretação da rede é dada por um vetor de seis termos representado por $Q=(N, \Sigma, \Phi, \lambda, D, \varphi)$, em que:

- $N=(G, M_0)$ é uma RdP, como definido anteriormente;
- $\Sigma=\{\sigma_1, \sigma_2, \dots, \sigma_r\}$ é o alfabeto das entradas da rede, em que σ_i é um símbolo de entrada;
- $\Phi=\{\phi_1, \phi_2, \dots, \phi_s\}$ é o alfabeto das saídas da rede, em que ϕ_i é um símbolo de saída;
- $\lambda:T \rightarrow \Sigma \cup \{\varepsilon'\}$ é a função que rotula as transições e dadas as seguintes restrições: $\forall t_j, t_k \in T, j \neq k$ se $I(p_i, t_j) = I(p_i, t_k) \neq 0$ e $\lambda(t_j), \lambda(t_k) \neq \varepsilon'$, então $\lambda(t_j) \neq \lambda(t_k)$. Neste caso, ε' representa um evento interno do sistema;

- $D:T \rightarrow T$ é a função de avanço da rede, em que $T = \tau \cup \{\varepsilon\}$ e $\tau \subset T$. Quando uma transição t_j é disparada, o nome de t_j ou o símbolo ε são associados a uma saída da rede. Neste caso, ε representa a ausência do sinal de saída;
- $\varphi:R(G,M_0) \rightarrow \{\Phi\}^q$ é uma função de saída, em que $R(G,M_0)$ é o conjunto de alcançabilidade definido para uma RdP e q é o número de saídas válidas associadas ao lugar.

3 DISCUSSÃO DAS DEFINIÇÕES

Nesta seção serão feitas algumas observações sobre as definições feitas na seção anterior e sobre as diferentes formas de se definir uma RdPi.

O SFC foi concebido para ser diretamente implementado em linguagem de programação. Além disso, pode ser convertido em linguagem Ladder para implementação em CLPs que não possuem suporte a linguagem SFC, como mostrado no COBENGE em Borges e Lima II (2017).

A linguagem SFC, padronizada a partir do Grafcet, guarda semelhanças com as redes de Petri interpretadas. Entretanto, há duas diferenças destacadas por David (1995): (i) a marca de um passo em Grafcet é booleana, enquanto a marca de um lugar em RdPi é numérica; (ii) em Grafcet, todas as transições habilitadas simultaneamente são disparadas ao mesmo tempo, enquanto em RdPi, se houver transições habilitadas e em conflito, apenas uma delas será disparada

Analisando essas duas diferenças, pode-se concluir que: (a) em RdPi, os lugares podem apresentar um número inteiro positivo de fichas (recursos): na RdPi do Frey (2000), por ser uma rede segura, o valor máximo que a função pode assumir é o unitário. No caso da RdPi de Ramírez-Treviño et al (2003), não há essa limitação. Já no SFC, os lugares são variáveis lógicas, ou seja, a cada lugar atribui-se o valor 0 (falso) ou 1 (verdadeiro); (b) em RdPi, quando há transições em conflito seu comportamento é não-determinístico, a menos que haja uma condição que selecione a transição a ser disparada; (c) se uma RdPi é segura (suas marcas se comportam como variáveis lógicas), e determinística (se não houver transições em conflito que apresentem condições de disparo verdadeiras simultaneamente), a RdPi pode ser representada em SFC.

Frey (2000) e Ramírez-Treviño et al (2003) diferem na construção da RdP ordinária. Enquanto Frey (2000) apresenta uma função para todos os arcos e não define peso para eles (já que trata a RdP como sendo segura), Ramírez-Treviño et al (2003) definem uma função peso para os arcos lugar-transição e outra para os arcos transição-lugar. Sendo assim, é possível representar recursos como o número de máquinas disponíveis ou o número de peças em um buffer, diferentemente da linguagem SFC e da RdPi segura.

Além disso, ambos os autores também diferem na interpretação da rede. Em Frey (2000), é definido um mapa que associa cada transição à sua condição de disparo, tal que $\varphi(t_i)$ é uma função booleana em I ; e um mapa que associa cada lugar com uma saída $\omega(p_i) \in (0, 1, -)$, em que $(-)$ significa ‘não importa’. Já em Ramírez-Treviño et al (2003), a função λ rotula as transições e a função D representa o avanço da rede. Ou seja, quando uma transição t_j é disparada, o nome de t_j ou o símbolo ε são associados a uma saída da rede. Neste caso, ε representa a ausência do sinal de saída.

4 CONCLUSÕES

Após esse levantamento dos diferentes tipos de RdPi, que surgiram inspiradas pelo trabalho de Petri, é possível perceber a importância que a pesquisa e o ensino têm na evolução da ciência e do conhecimento científico em geral. Uma teoria que nasceu para a ciência da computação tornou-se muito maior pelas diferentes aplicações que foram encontradas em automação e outros campos do conhecimento. Portanto, a rede de Petri é uma ferramenta fundamental para o ensino de engenharia e também para a organização industrial. Sendo assim, o professor deve explicar didaticamente aos seus alunos quais são os conceitos e as aplicações desta teoria.

REFERÊNCIAS

BORGES, Matheus U; LIMA II, Eduardo J. Conversão de Rede de Petri Interpretada em Linguagem Ladder: Metodologia e Exemplo. In: XLV Congresso Brasileiro de Educação em Engenharia (COBENGE), 2017, Joinville. **Anais**. Joinville, 2017

BRAUER, Wilfried; REISIG, Wolfgang. Carl Adam Petri and “Petri nets”. In: **Fundamental Concepts in Computer Science**. 2009. p. 129-139.

CARDOSO, Janette; VALETTE, Robert. **Redes de petri**. Editora da UFSC, 1997.

DAVID, René. Grafcet: A powerful tool for specification of logic controllers. **IEEE Transactions on control systems technology**, v. 3, n. 3, p. 253-268, 1995.

DE MELLO, André Torres Ferraz 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**. 2012.

FREY, Georg. Automatic implementation of Petri net based control algorithms on PLC. In: **American Control Conference, 2000. Proceedings of the 2000**. IEEE, 2000. p. 2819-2823.

IEC, I. E. C. 61131-3: Programmable controllers—part 3: Programming languages. **International Standard, Second Edition, International Electrotechnical Commission, Geneva**, v. 1, p. 2003, 2003.

LIMA II, E. J: **Projeto de Sistemas Sequenciais de Automação Utilizando CLPs**. Apostila DEMEC/UFMG, Belo Horizonte, 2007

MURATA, Tadao. Petri Nets: Properties, Analysis and Applications. **Proceedings of the IEEE**, v. 77, n. 4, p. 541-580, 1989.

PETRI, Carl Adam. **Kommunikation mit Automaten**. 1962.128 f. Tese (Doutorado). Fakultät Für Mathematik Und Physik, Technical University of Darmstadt, Darmstadt, 1962.

PETRI, Carl Adam. Grundsätzliches zur Beschreibung diskreter Prozesse. In: **3. Colloquium über Automatentheorie**. Birkhäuser, Basel, 1967. p. 121-140.

PETRI, Carl Adam. Introduction to general net theory. In: **Net theory and applications**. Springer, Berlin, Heidelberg, 1980. p. 1-19.

RAMÍREZ-TREVIÑO, Antonio; RIVERA-RANGEL, Israel; LÓPEZ-MELLADO, Ernesto. Observability of discrete event systems modeled by interpreted Petri nets. **IEEE Transactions on Robotics and Automation**, v. 19, n. 4, p. 557-565, 2003.

SILVA, Hugo Vinícius Dias. **Metodologia de projeto de automação industrial visando a conversão automática de redes de petri interpretadas em códigos implementáveis**. 2013. 71 f. Dissertação (Mestrado). Faculdade de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Cornélio Procopio, 2013

SILVA, Manuel. 50 years after the PhD thesis of Carl Adam Petri: A perspective. **IFAC Proceedings Volumes**, v. 45, n. 29, p. 13-20, 2012.

SIGNAL INTERPRETED PETRI NETS EVOLUTION: A VIEW IN THREE DEFINITIONS

Abstract: *This paper aims to portrait Carl Adam Petri's tool evolution created on his Ph.D. thesis, in 1962. At first, only the mathematical basis based on computer sciences appeared and, some years later, Petri himself created a graphical notation. Over a period of more than fifty years, researchers have scrutinized the thesis and, thus, several types of Petri nets have emerged with their specificities to be able to represent some problems in more detail, such as the description of discrete events systems (DES), whose evolution depends only on the event (and not on the time). One of these types is the Signal Interpreted Petri Net (SIPN), which will be presented in three definitions of different authors. Subsequently, a comparison of notations and the analysis of their characteristics will be made.*

Key-words: *Carl Adam Petri. Petri Nets. Signal Interpreted Petri Nets.*