

Metodologia para Conversão de Rede de Petri Interpretada para Grafcet Visando Projetos de Automação Industrial

Hugo Vinicius Dias Silva e Marcos Banheti Rabello Vallim

Programa de Pós-Graduação em Engenharia Elétrica

UTFPR - Universidade Tecnológica Federal do Paraná. Câmpus Cornélio Procópio

Av. Alberto Carazzai, 1640 - Centro - Cornélio Procópio - Paraná - Brasil

Email: a1218735@pos.utfpr.edu.br, mvallim@utfpr.edu.br

Abstract— This article aims to contribute to the study of Discrete Event Systems (DES), presenting a methodology for control logic synthesis targeting industrial automation projects, converting a high power formalism of abstraction to programming language of Programmable Logic Controllers (PLC). The proposed methodology allows to convert a Petri Net Interpreted directly to Grafcet without compromising the readability and reliability of the generated algorithm. The methodology is validated presenting the results of the modeling and control of Sorting of Boxes process, considering performance specifications. The proposed methodology consists of seven conversion steps, performing the modeling, analysis and creation of Grafcet control, increasing the reliability of the control logic.

I. INTRODUÇÃO

Os sistemas automatizados estão presentes na maioria das indústrias. Tais sistemas possuem a característica de evoluir de acordo com a ocorrência de eventos independentemente da passagem do tempo, realizando funções que demandam alto grau de confiabilidade [1]. Por isso, faz-se necessário o desenvolvimento de ferramentas capazes de identificar estados indesejados ainda em fase de projeto.

Neste contexto a Rede de Petri (RdP) é uma das ferramentas utilizadas para modelar e analisar o comportamento dinâmico presente em Sistemas a Eventos Discretos (SEDs), proporcionando ao projetista uma visão geral do comportamento do sistema [2]. Porém, a implementação da RdP em um Controlador Lógico Programável (CLP) é uma tarefa difícil.

Algumas técnicas propõem a tradução dos elementos da rede diretamente para a linguagem de programação de CLP, o que na maioria dos casos torna o algoritmo gerado ilegível, extenso e não reutilizável, diminuindo a aplicabilidade da RdP nos projetos de automação industrial [3], [4], [5].

O objetivo deste artigo é contribuir para os estudos de SEDs, apresentando uma metodologia de conversão de RdP para uma linguagem de programação compreendida pelos CLPs, gerando algoritmos legíveis e reutilizáveis. As etapas para conversão da RdP para Grafcet são apresentadas, e para validar a proposta realiza-se a implementação do controle de um processo de Triagem de Caixas.

O artigo está estruturado da seguinte forma: Na Seção, 2 são apresentados os conceitos básicos de RdP. Na Seção 3,

são mostradas as definições da linguagem Grafcet. Na Seção 4, são apresentados os detalhes da metodologia para conversão de RdP para Grafcet. Na Seção 5, é mostrado um exemplo de aplicação da metodologia para validação da proposta. Finalizando, na Seção 6 as conclusões são apresentadas.

II. REDE DE PETRI

A RdP é um formalismo que possibilita a representação gráfica com um alto nível de abstração, apresentada por Carl Adam Petri em sua tese de doutorado em 1962, inicialmente aplicada para avaliações de desempenho e representação do fluxo de dados nos protocolos de comunicação [2]. Recentemente, este formalismo é aplicado na modelagem e análise de sistemas de automação industrial, representando de forma eficiente o comportamento assíncrono presente nos SEDs.

Uma RdP é definida matematicamente por uma quádrupla [2], mostrado na equação 1:

$$R = \langle P, T, Pre, Post \rangle \quad (1)$$

onde, P é o conjunto de lugares finitos, T é o conjunto de transições finitas, Pre é uma matriz com dimensões $P \times T$, que relaciona os lugares anteriores com as transições precedentes e $Post$ é uma matriz com dimensões $P \times T$, que relaciona os lugares posteriores com as transições anteriores. O conjunto de lugares do modelo é definido por $P = \{p_1, p_2, p_n, \dots\}$, e o conjunto de transições por $T = \{t_1, t_2, t_m, \dots\}$ e as matrizes de aplicação são $Pre = \{p_n, t_m\}$ e $Post = \{p_n, t_m\}$, onde n é a quantidade de lugares e m o número de transições existentes no modelo.

As transições t_m são representadas graficamente por uma barra horizontal, associando os lugares que antecedem a transição com os lugares posteriores, ligados através de arcos.

Os lugares p_n representam os possíveis estados que o sistema modelado pode assumir, descrito graficamente por um círculo.

Uma ficha sobre um lugar do modelo define que tal lugar está marcado, e a combinação de todas as fichas do modelo descrevem o atual estado do sistema, graficamente representada por um ponto dentro do círculo (lugar).

Os arcos conectam os lugares às transições do modelo, e cada arco possui um peso específico.

Uma RdP é considerada binária se todos os lugares contidos no modelo assumem somente a marcação 0 ou 1, considerando a marcação inicial e o acontecimento dos possíveis disparos das transições. De modo simplificado, uma ficha simboliza se a condição associada ao lugar é verdadeira ou não.

Para que uma transição seja ativada, é necessário que as condições relacionadas às transições sejam atendidas. Uma vez disparada, a transição muda as posições das fichas entre os lugares conectados pelos arcos, retirando as fichas dos lugares que antecedem e depositando nos lugares posteriores.

Para que seja possível identificar a marcação atual da rede (M'), utiliza-se a matriz de incidência, que é obtida através da subtração da matriz $Post$ pela Pre , logo, o produto da matriz de incidência (C) com a sequência de disparo das transições (s) adicionado à marcação inicial da rede (M), definem o estado atual do sistema modelado, conforme apresentado na equação 2.

$$M' = M + Cs \quad (2)$$

Uma RdP pode ser utilizada para representar diversas entidades abstratas, dependendo do nível de abstração utilizado no seu desenvolvimento. Sendo assim, um lugar poderá representar uma entidade física ou não, como uma peça a ser processada ou uma máquina fora de operação.

A. Rede de Petri Interpretada

As Redes de Petri Interpretadas (RdPI) associam as variáveis dos sistema às transições da rede, que descrevem as condições e ações existentes no processo. Estas variáveis podem indicar os estados dos atuadores ou sensores do sistema, permitindo a modelagem das interações com o ambiente externo [6].

Uma RdPI é dividida em duas partes, sendo elas: controle e dados. O controle descreve todas as evoluções possíveis do processo relacionadas aos eventos. Os dados descrevem as estruturas de dados internas ao sistema e as informações recebidas do mundo externo. Os elementos que constituem os dados de uma Rede de Petri Interpretada são as condições e as ações.

As condições C_i e as ações A_i são expressas como disjunção e conjunção *booleanas* e são associadas às respectivas transições. As condições e ações são representadas graficamente por (C_i, A_i) ao lado da transição pertencente.

Resumidamente, os passos para modelar um sistema utilizando RdPI são [6]:

- 1) Encontrar a estrutura da Rede de Petri que representa as atividades concorrentes, paralelas e sequenciais, e descrever os eventos associados ao meio externo.
- 2) Analisar a rede gerada verificando as propriedades. Caso necessário, corrigir os eventuais problemas.
- 3) Simular a Rede Interpretada (controle e dados) para extrair o comportamento do sistema.

III. GRAFCET

Originado na década de 70, com foco na França, o Grafcet surgiu para unificar os conceitos e técnicas propostas anteriormente, sendo regulamentado em 1993 através da IEC 1131-3 [7]. O nome Grafcet derivou-se de “Graph”, pois o modelo desenvolvido utiliza elementos gráficos para representação. Utilizado principalmente na soluções de problemas cuja as ações de comando são sequenciais ou de tempo dependente, problemas que a solução utilizando representação Ladder, poderia se tornar confusa e muitas vezes inviáveis [8].

Também considerado um caso particular da RdP, o Grafcet é definido com uma RdP binária, onde as matrizes Pre e $Post$ só podem assumir valores de 0 ou 1, correspondente às variáveis booleanas que simboliza a presença ou ausência em um etapa [8].

Através da organização dos elementos do Grafcet o comportamento dinâmico é formado e seus elementos são divididos da seguinte maneira [8]:

- **Etapas:** simbolizam um estado no qual o comportamento do circuito de comando não se relaciona com as entradas e saídas atuais do sistema. Por definição, uma etapa é representada graficamente por um quadrilátero, sendo identificada por números seguidos ou não por abreviações. Durante a evolução do Grafcet, uma etapa pode estar ativa ou inativa. A etapa inicial é definida por uma quadrilátero duplo e representa a etapa que será ativada logo após o início do funcionamento do sistema.
- **Transições:** representadas graficamente por um traço perpendicular aos *arcos orientados* e conectando à etapa anterior com a etapa posterior à transição.
- **Arcos Orientados:** responsáveis por indicar a sequência de evolução do Grafcet, interligando etapas e transições, e sendo interpretados da parte superior para inferior, representada por uma linha contínua.
- **Ações:** representam os efeitos que devem ser realizados quando alcançada uma determinada etapa do Grafcet. Cada ação é representada graficamente por um retângulo alocado à direita da etapa em que está associado. Alguns qualificadores podem ser associados às ações, alterando a maneira em que as ações são executadas. Como por exemplo a inserção dos qualificadores S (Set) ou R (Reset), utilizados como elementos memorizadores de ações.
- **Receptividade:** condição lógica associada à cada transição, e seus estados lógicos, habilitado ou não habilitado, ativam ou não as transições associadas.

Devido ao maior nível de representatividade, o Grafcet se popularizou no ambiente industrial, possibilitando a construção de algoritmos de controle mais legíveis.

IV. METODOLOGIA PARA CONVERSÃO DE RDPI PARA GRAFCET

Utilizando as definições de RdPI apresentadas em [9] e utilizando como base a proposta de construção do Grafcet

apresentada em [10], [8], podem ser estabelecidas as seguintes etapas de tradução:

- 1) Criar para cada elemento do sistema uma RdPI que associa às condições e ações essenciais para a evolução do processo;
- 2) Agregar todas as RdPI criadas com os elementos auxiliares (temporizadores e contadores) e organizar as condições e ações de cada transição em uma tabela de dados;
- 3) Analisar as propriedades da RdPI geral e simular a rede em conjunto com a tabela de dados, buscando comprovar a correta evolução do processo.
- 4) Criar módulos com os elementos que possuem condições e ações compartilhadas;
- 5) Converter cada módulo para Grafcet, considerando as condições como a receptividade da transição e as ações como etapas;
- 6) Marcar as etapas iniciais de cada Grafcet com o estado inicial de cada elemento;
- 7) Implementar no CLP todos os Grafkets gerados, que serão executados simultaneamente.

Na Figura 1 é apresentado um esquemático do método de conversão proposto.

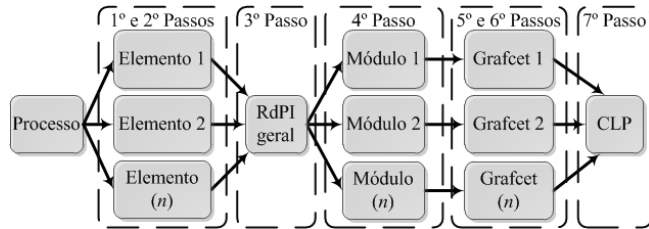


Fig. 1. Esquemático do método utilizado

V. EXEMPLO DE APLICAÇÃO

Para validar a proposta realiza-se o controle de um sistema contido no *software* ITS PLC (Interactive Training System for Programmable Logic Controller) Professional Edition que dispõe de cinco sistemas industriais virtuais, são eles: (a) Triagem de Caixas, (b) Batelada, (c) Paletizadora, (d) Pick and Place e (e) Armazém automático, conforme mostrado na Figura 2. Com o uso deste *software* é possível utilizar vários ambientes sem a necessidade da montagem física da planta industrial [11], [12].

O ITS PLC é instalado em um computador convencional, conectado ao CLP através de uma placa de condicionamento de sinal que utiliza uma placa de aquisição de dados (DAQ) e uma placa de isolamento de sinal. Logo, os sinais elétricos recebidos do sistema virtual são repassados em forma tensão para o CLP por meio da DAQ, entre 0 a 24 Vcc.

Para o controle do processo utiliza-se um CLP modelo S7 – 300 da marca Siemens, onde o algoritmo de controle é gravado e executado, este CLP está instalado em uma módulo didático que possui para cada entrada e saída do CLP, *bornes* para conexão da placa de condicionamento de sinal,

facilitando o acesso aos módulos de I/O (*Input/Output*) do CLP. O esquema de comunicação do CLP ao computador é apresentado na Figura 2.

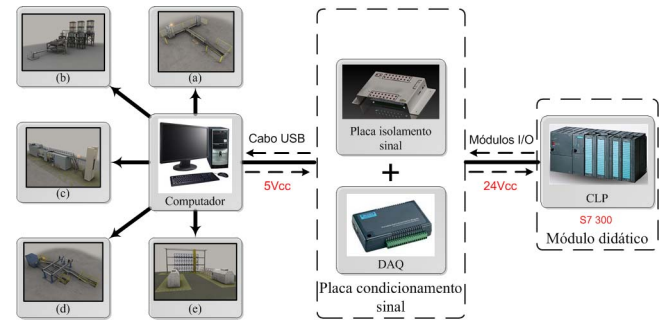


Fig. 2. Esquemático de comunicação

Neste trabalho utiliza-se o processo de Triagem de Caixas, pois este está presente em várias indústrias. Este sistema contém um conjunto de esteiras que são acionadas individualmente para que as caixas sejam despachadas corretamente. Na Figura 3 é mostrada a alocação das esteiras transportadoras, os sensores e atuadores no ambiente.

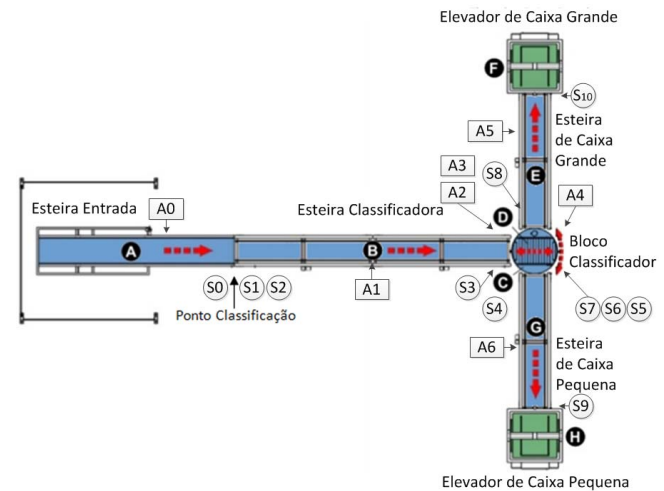


Fig. 3. Sistema de Triagem de Caixas

Conforme descrito na Figura 3, o processo de Triagem de Caixas é iniciado com o acionamento da *Esteira de Entrada* (A), que transporta as caixas até a *Esteira Classificadora* (B), onde é realizada a classificação da caixa. A *Esteira Classificadora* transporta a caixa classificada até o *Bloco Classificador* (C), que realiza um giro de 90°. Conforme o tipo de caixa classificada, os roletes (D) do *Bloco Classificador* deverão ser acionados de modo diferente: caso a caixa seja pequena, os roletes deverão levar a caixa em direção do *Elevador de Caixa Pequena* (H) através da *Esteira de Caixa Pequena* (G); caso a caixa seja classificada como grande, os roletes deverão enviá-la para o *Elevador de Caixa Grande* (F) através da *Esteira de Caixa Grande* (E).

Os sensores de presença de caixas instalados na planta virtual são responsáveis por repassar ao CLP as informações

sobre o sistema. Uma relação dos sensores utilizados é apresentada na Tabela I.

TABELA I
Relação de Sensores do Sistema

Sensores	Descrição	Endereço
S0	Saída da <i>Esteira Entrada</i>	I0.0
S1	Sensor de caixa baixa	I0.1
S2	Sensor de caixa alta	I0.2
S3	Saída da <i>Esteira Classificadora</i>	I0.3
S4	BC posição carregar	I0.4
S5	BC posição descarregar	I0.5
S6	Caixa no <i>Bloco Classificador</i>	I0.6
S7	Entrada caixa <i>Esteira Pequena</i>	I0.7
S8	Entrada caixa <i>Esteira Grande</i>	I1.0
S9	Saída caixa <i>Esteira Pequena</i>	I1.1
S10	Saída caixa <i>Esteira Grande</i>	I1.2
B1	Botão iniciar pressionado	I1.4

(BC) - Bloco Classificador

Para que seja possível a classificação das caixas, ao final da *Esteira de Entrada* há dois sensores (*S1* e *S2*), que adquirem a altura das caixas que passam da *Esteira de Entrada* à *Esteira Classificadora*.

Os atuadores realizam mudanças nos estados dos elementos que compõem o sistema, avançando esteiras ou realizando giros no *Bloco Classificador*, garantindo a interação direta do CLP com o ambiente controlado. A relação de atuadores utilizados no processo de Triagem de Caixas é apresentada na Tabela II.

TABELA II
Relação de Atuadores do Sistema

Atuador	Descrição	Endereço
A0	Avança <i>Esteira de Entrada</i>	Q0.0
A1	Avança <i>Esteira Classificadora</i>	Q0.1
A2	Avança roletes BC	Q0.2
A3	Récula roletes BC	Q0.3
A4	Gira <i>Bloco Classificador</i>	Q0.4
A5	Avança <i>Esteira Caixa Grande</i>	Q0.5
A6	Avança <i>Esteira Caixa Pequena</i>	Q0.6
L1	Acende sinal luminoso	Q1.0

(BC) - Bloco Classificador

Para exemplificar a conversão da RdPI para Grafcet, utiliza-se a representação da lógica de controle atribuída ao sistema de Triagem de Caixas, realizando as etapas:

1ª Etapa: na primeira etapa da conversão, cada elemento que constitui o sistema de Triagem de Caixas é modelado individualmente. Para a *Esteira de Entrada* são identificados dois estados: esteira aguardando acionamento, representado no modelo pelo lugar *Esteira_Ent_Parada* e o estado em que a esteira está em funcionamento, descrito no modelo pelo lugar *Esteira_Ent_Operando*. Duas transições (*t₀* e *t₁*) são utilizadas para conectar os estados descritos, logo, obtém-se a RdPI mostrada na Figura 4:

Para o desenvolvimento da RdPI da *Esteira Classificadora* a informação do tipo de caixa que está sobre a esteira é relevante, logo três estados são criados: *Esteira_Class_Transp_Cx_G* para o estado em que a *Esteira Classificadora* transporta caixas do tipo grande;

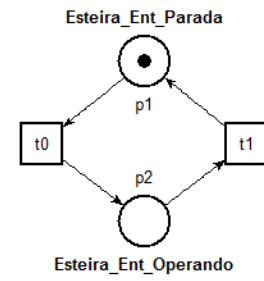


Fig. 4. RdPI da *Esteira de Entrada*

Esteira_Class_Transp_Cx_P para o transporte de caixas pequenas e *Esteira_Class_Parada* para o estado ocioso da *Esteira Classificadora*, conforme apresentado na Figura 5.

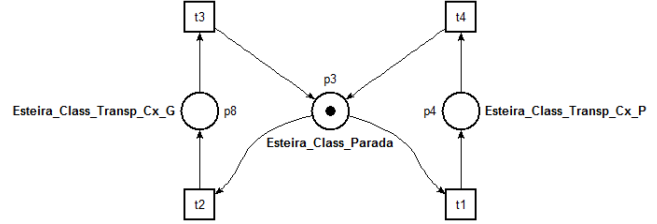


Fig. 5. RdPI da *Esteira de Classificadora*

O *Bloco Classificador* é semelhante à *Esteira Classificadora*, pois o tipo de caixa que está sobre o bloco define a direção em que os roletes serão acionados após a conclusão do giro. Assim, a RdPI possui estados que representam tal informação do processo, que são eles: *Bloco_gira_Elev_Grande* que é o estado em que o *Bloco Classificador* está posicionado para enviar a caixa para o *Elevador de Caixa Grande*, *Bloco_gira_Elev_Pequeno* quando o bloco está posicionado para enviar caixa para o *Elevador de Caixa Pequena* e o estado *Bloco_parado*, representando a ociosidade do *Bloco Classificador*. A RdPI do elemento descrito é mostrada na Figura 6.

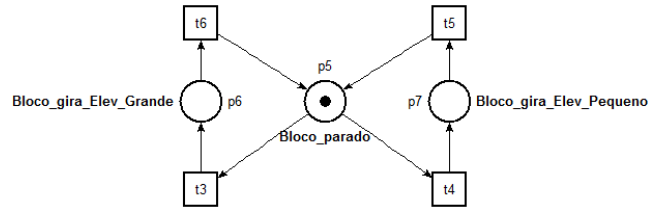


Fig. 6. RdPI do *Bloco Classificador*

Os elementos *Esteira de Caixa Grande* e *Esteira de Caixa Pequena* apresentam estados de funcionamento semelhantes, logo as RdPIs que representam a lógica de controle possuem dois lugares: *Esteira_Cx_Grande_Parada* e *Esteira_Cx_Pequena_Parada* para os estados em que as esteiras estão ociosas e *Esteira_Cx_Grande_Operando* e *Esteira_Cx_Pequena_Operando* para quando as esteiras estão avançando. As RdPIs da *Esteira de Caixa Grande* e

Esteira de Caixa Pequena são apresentadas na Figura 7(a) e 7(b), respectivamente.

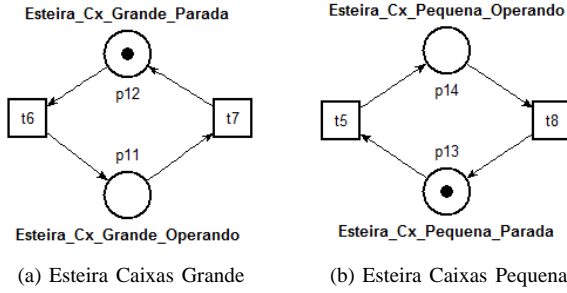


Fig. 7. RdPI de controle das *Esteiras de Caixas Grande e Pequena*

2ª Etapa: após a modelagem da lógica de controle de cada elemento que compõe o sistema, realiza-se a união das RdPI considerando as transições que são compartilhadas pelos elementos e a especificação de funcionamento do processo. A RdPI de controle do processo é apresentada na Figura 8.

Utilizando a RdPI de controle e as especificações de funcionamento é necessário analisar todas as transições contidas no modelo, buscando definir qual a condição que deverá ser satisfeita e a ação associada ao disparo da transição. Para exemplificar, utiliza-se a transição t_0 , que quando disparada representa a mudança de estado da *Esteira de Entrada*, que inicialmente está na condição parada. Sendo assim, uma condição que deve ser satisfeita para que a *Esteira de Entrada* entre em operação é o acionamento do Botão B1 ($B1 = 1$). Logo, esta condição deve ser agregada à transição t_0 . Após o acionamento do Botão B1 a *Esteira de Entrada* entrará em funcionamento, então, a ação associada ao disparo da transição t_0 será o avanço da esteira ($est_entr = 1$). Todas as transições da RdPI de controle do processo e suas condições e ações estão organizadas em uma tabela de dados, mostrada na Tabela III.

TABELA III

Tabela de Dados - Descrição das Condições e Ações		
t_n	Acontecimento	Condições ; Ações (C_i ; A_i)
t_0	Apertar_B1	($B1 = 1$; $est_entr = 1$)
t_1	Avanc_Est_Class_P	($\downarrow S0 = 1 \wedge \downarrow S2 = 0$; $est_entr = 0 \wedge est_class = 1 \wedge Cx_P = 1$)
t_2	Avanc_Est_Class_G	($\downarrow S0 = 1 \wedge \downarrow S2 = 1$; $est_entr = 0 \wedge est_class = 1 \wedge Cx_G = 1$)
t_3	Carrega_Bloco_G	($S3 = 1 \wedge Cx_G = 1$; $est_class = 0 \wedge rolo_avanc = 1$)
t_4	Carrega_Bloco_P	($S3 = 1 \wedge Cx_P = 1$; $est_class = 0 \wedge rolo_avanc = 1$)
t_5	Descarrega_Bloco	($S7 = 1$; $est_peq = 1 \wedge rolo_avanc = 0 \wedge Cx_P = 0$)
t_6	Descarrega_Bloco	($S8 = 1$; $est_grande = 1 \wedge rolo_avanc = 0 \wedge Cx_G = 0$)
t_7	Despacha_caixa	($\downarrow S10 = 1$; $est_grande = 0$)
t_8	Despacha_caixa	($\downarrow S9 = 1$; $est_peq = 0$)

Com base na RdPI de controle do processo de Triagem de Caixas, realiza-se a análise das boas propriedades da rede, considerando suas propriedades estruturais e marcações.

3ª Etapa: nesta etapa a RdPI é analisada e simulada, buscando-se identificar possíveis problemas antes da implementação. Para este procedimento utiliza-se o *software*

Tina 3.0 (Time Petri Net Analyzer) apresentado em [13]. Tal ferramenta possibilita a representação gráfica, análise e simulação da RdP, auxiliando no processo de modelagem do processo. Utilizando o recurso de simulação disponível na ferramenta, é possível disparar as transições contidas na RdP e visualizar o próximo estado alcançado, sendo útil na visualização da evolução do processo em tempo de projeto.

Analisando a RdPI gerada para o controle do processo de Triagem de Caixas, obtém-se que a rede é viva, logo, todas as transições contidas no modelo são sensibilizadas através de uma sequência de disparo conhecida. A RdPI analisada também é definida como limitada, pois o número de marcações da rede permanece constante considerando a sua marcação inicial. A rede é reiniciável, pois partindo da marcação inicial da rede existe uma sequência de disparo que retorna a sua marcação inicial.

Analisando as propriedades estruturais do modelo é possível identificar informações relacionadas à dinâmica do sistema, como a sequência de disparo das transições, para que seja realizada uma tarefa relevante. Realizando uma análise estrutural da RdPI de controle identifica-se um invariante de transição, que descreve a sequência de disparo para classificação das caixas do tipo pequena (t_0, t_1, t_4, t_5, t_8) e do tipo grande (t_0, t_2, t_3, t_6, t_7).

Através da análise estrutural também é possível identificar o invariante de lugar, que descreve os elementos conservativos presentes no sistema. A RdPI apresentou como invariante os lugares mostrados na Tabela IV.

TABELA IV
Invariantes de Lugar

Lugares	Elemento do sistema
p_1, p_2	<i>Esteira de Entrada</i>
p_3, p_4, p_8	<i>Esteira Classificadora</i>
p_5, p_6, p_7	<i>Bloco Classificador</i>
p_{13}, p_{14}	<i>Esteira Caixa Pequena</i>
p_{11}, p_{12}	<i>Esteira Caixa Grande</i>

Concluída a etapa de análise, realiza-se a simulação da RdPI com o apoio do *software* Tina. Nesta etapa, para cada disparo das transições é necessário verificar na tabela de dados quais as condições e ações envolvidas, buscando avaliar a dinâmica do processo e possíveis problemas lógicos.

4ª e 5ª Etapas: após a análise das boas propriedades do modelo, identifica-se com o uso da RdPI de controle e da tabela de dados quais as condições e ações que estão atribuídas a cada elemento que compõe o sistema. Após a identificação das condições e ações em comum, criam-se módulos de controle descritos em Grafcet. Cada módulo é responsável pelo acionamento e desacionamento de apenas um elemento do sistema.

Para que a *Esteira de Entrada* entre em funcionamento é necessário que o Botão B1 (transição t_0) seja pressionado, sendo assim, a condição lógica ($B1 = 1$) deve ser satisfeita para que a esteira inicie o avanço ($est_entr = 1$). Já para que a *Esteira de Entrada* retorne à condição parada, um sinal representando que a caixa classificada saiu da esteira é necessário, as transições t_1 e t_2 representam o acontecimento deste evento, retornando a

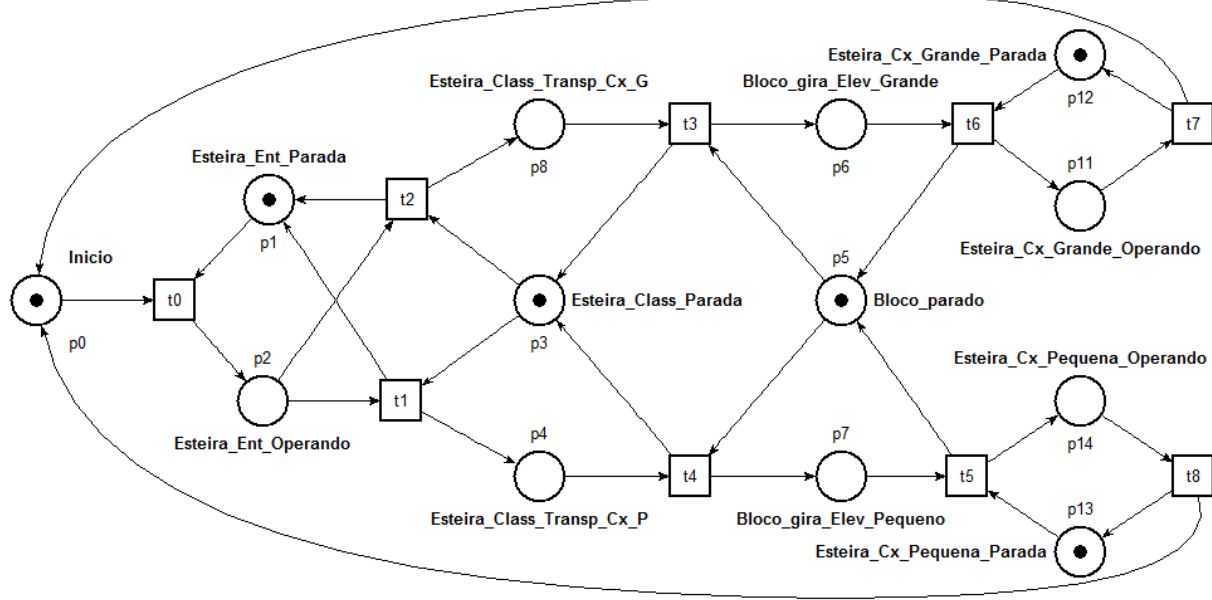


Fig. 8. RdPI de controle do processo de Triagem de Caixas

marcação da rede para o estado onde as esteiras estão ociosas, sendo necessário uma simplificação da expressão booleana $\downarrow S0 = 1 \wedge \downarrow S2 = 0$ e $\downarrow S0 = 1 \wedge \downarrow S2 = 1$, resultando na condição $\downarrow S0 = 1$.

Com base nas condições e ações associadas a *Esteira de Entrada*, cria-se o Grafet de controle, onde as condições de início e fim de operação são as receptividades das transições. O Grafet de controle da *Esteira de Entrada* é mostrado na Figura 9.

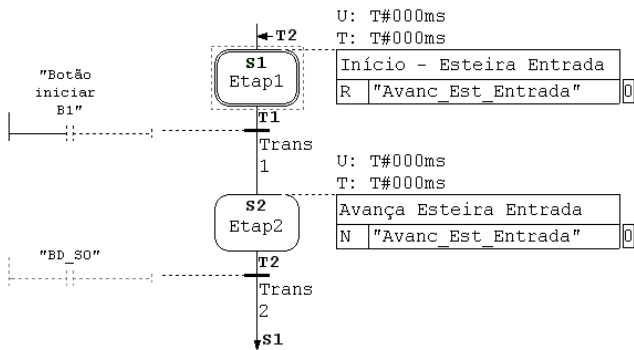


Fig. 9. Grafet de controle *Esteira de Entrada*

Na Figura 9 a condição “BD.S0” contida na transição T2 significa que o sinal S0 deve passar de 1 para 0 para que a condição seja verdadeira, isto é, uma borda de descida.

Para o acionamento da *Esteira Classificadora* estão associadas as transições t_1 e t_2 , realizando uma simplificação das expressões booleanas $\downarrow S0 = 1 \wedge \downarrow S2 = 0$ e $\downarrow S0 = 1 \wedge \downarrow S2 = 1$, obtem-se $\downarrow S0 = 1$, mas para implementação desconsidera-se o sinal de borda de descida do sensor S0. Para voltar a *Esteira Classificadora* ao estado ocioso a caixa que está sobre a esteira deve chegar ao *Bloco Classificador*, o sinal $\downarrow S3=1$

representa este evento. Desta forma, estas condições e ações serão descritas no Grafet de controle, mostrado na Figura 10.

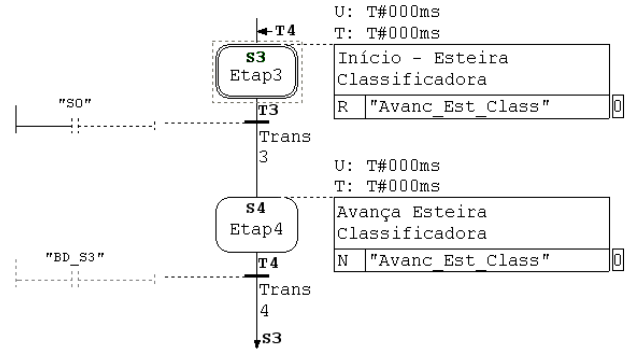


Fig. 10. Grafet de controle *Esteira Classificadora*

Para que o *Bloco Classificador* inicie o processo de carga da caixa classificada é necessário que a caixa passe pelo sensor de presença de caixas (S3). Esta condição é identificada realizando-se a simplificação da expressão booleana associada às transições t_3 e t_4 . Para que o *Bloco Classificador* retorne ao estado parado, a caixa que está sobre o bloco deverá ser repassada para a *Esteira de Caixa Grande* ou a *Esteira de Caixa Pequena*. Tais condições são apresentadas nas transições t_5 e t_7 , e realizando-se a simplificação das expressões, tem-se que $\downarrow S7 = 1 \vee \downarrow S8 = 1$. O Grafet de controle baseado nestas condições e ações é mostrado na Figura 11.

A *Esteira de Caixa Grande* iniciará seu avanço após a entrada de uma caixa sobre a esteira, esta condição está agregada à transição t_6 , que possui a condição $S8 = 1$. Para o retorno da esteira ao estado parada é necessário que a caixa saia completamente da esteira, este evento é descrito pela transição t_7 , onde a expressão que garante esta condição é

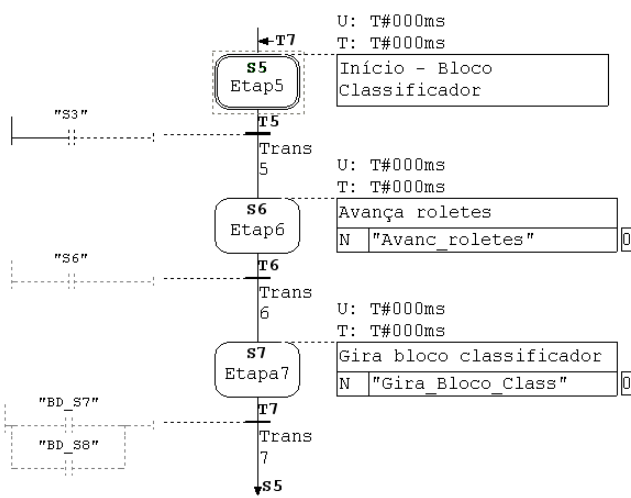


Fig. 11. Grafset de controle *Bloco Classificador*

$\downarrow S10 = 1$. Considerando as condições e ações da *Esteira de Caixa Grande* cria-se o Grafset de controle mostrado na Figura 12.

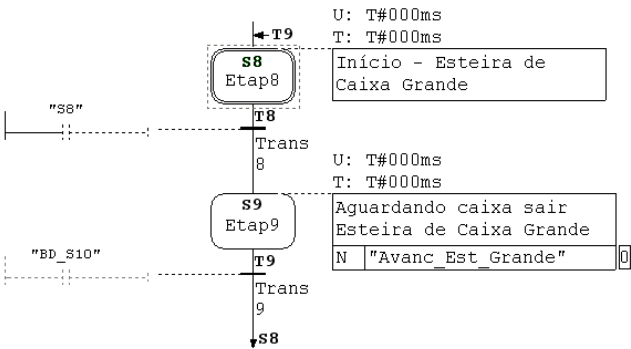


Fig. 12. Grafset de controle *Esteira de Caixa Grande*

A *Esteira de Caixa Pequena* começará a avançar somente com caixa sobre a mesma, tal condição é descrita na transição t_5 que está agregada à condição $S7 = 1$, e para o retorno ao estado ocioso, a transição t_8 , que possui a condição $\downarrow S9 = 1$, deve ser satisfeita. O Grafset que realiza o controle da *Esteira de Caixa Pequena* considerando estas condições é mostrado na Figura 13.

Observando a análise dos invariantes de transição da RdPI do processo é possível identificar dois conjuntos de transições $(t_0, t_1, t_4, t_5, t_8)$ e $(t_0, t_2, t_3, t_6, t_7)$, o primeiro conjunto de transições representa a classificação de uma caixa do tipo pequena e o segundo conjunto a triagem de uma caixa grande. Com base nestas informações, cria-se um Grafset do *Módulo de Classificação* que armazena o tipo da caixa classificada e realiza o acionamento dos roletes do *Bloco Classificador*, direcionando a caixa classificada para o elevador correto.

Este *Módulo de Classificação* é criado utilizando as transições que indicam o tamanho da caixa sobre a *Esteira de Classificação*, as transições t_1 e t_0 apresentam o contador "Cx_P" e "Cx_G", que indicam o tipo de caixa. Após

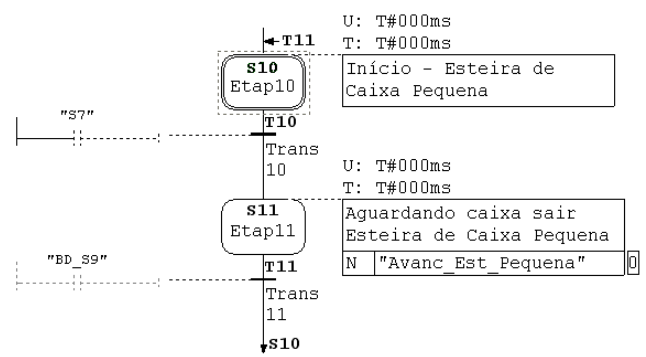


Fig. 13. Grafset de controle *Esteira de Caixa Pequena*

a classificação da caixa, os roletes do *Bloco Classificador* deverão ser acionados conforme a especificação de funcionamento. Para a caixa pequena os roletes deverão recuar até que a caixa saia ($\downarrow S7 = 1$) do *Bloco Classificador*. Para caixas grandes os roletes deverão avançar até que a caixa esteja fora do bloco ($\downarrow S8 = 1$). O Grafset para o módulo de classificação é mostrado na Figura 14.

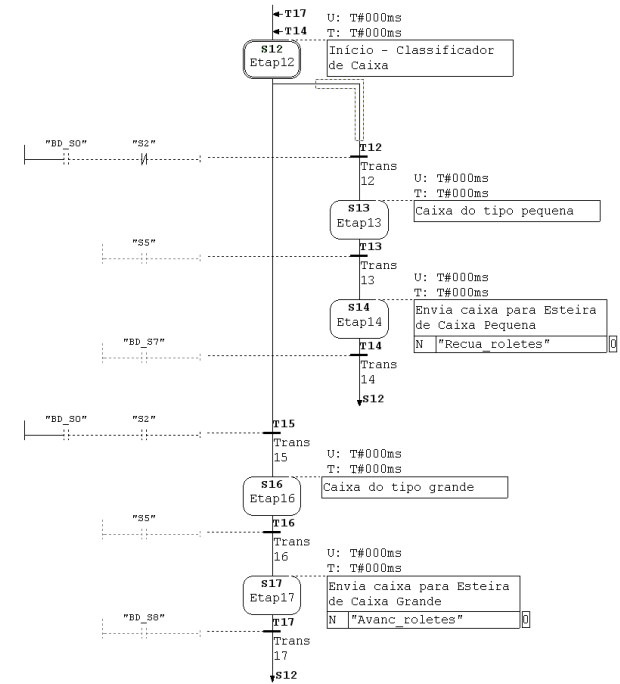


Fig. 14. Grafset controle *Módulo de Classificação*

Nesta etapa de conversão é possível inserir ou remover elementos de apoio, como os contadores e detectores de borda de subida, para garantir a perfeita implementação do controle. A abstração de algumas etapas do sistema também é possível, permitindo a criação de um modelo mais legível.

6ª Etapa: com base nos Grafsets gerados e na RdPI que representa o processo é necessário marcar as etapas iniciais dos Grafsets. Essa marcação é realizada observando-se os elementos modelados na primeira etapa da conversão, sendo necessária para que o sistema controlado inicie em estado

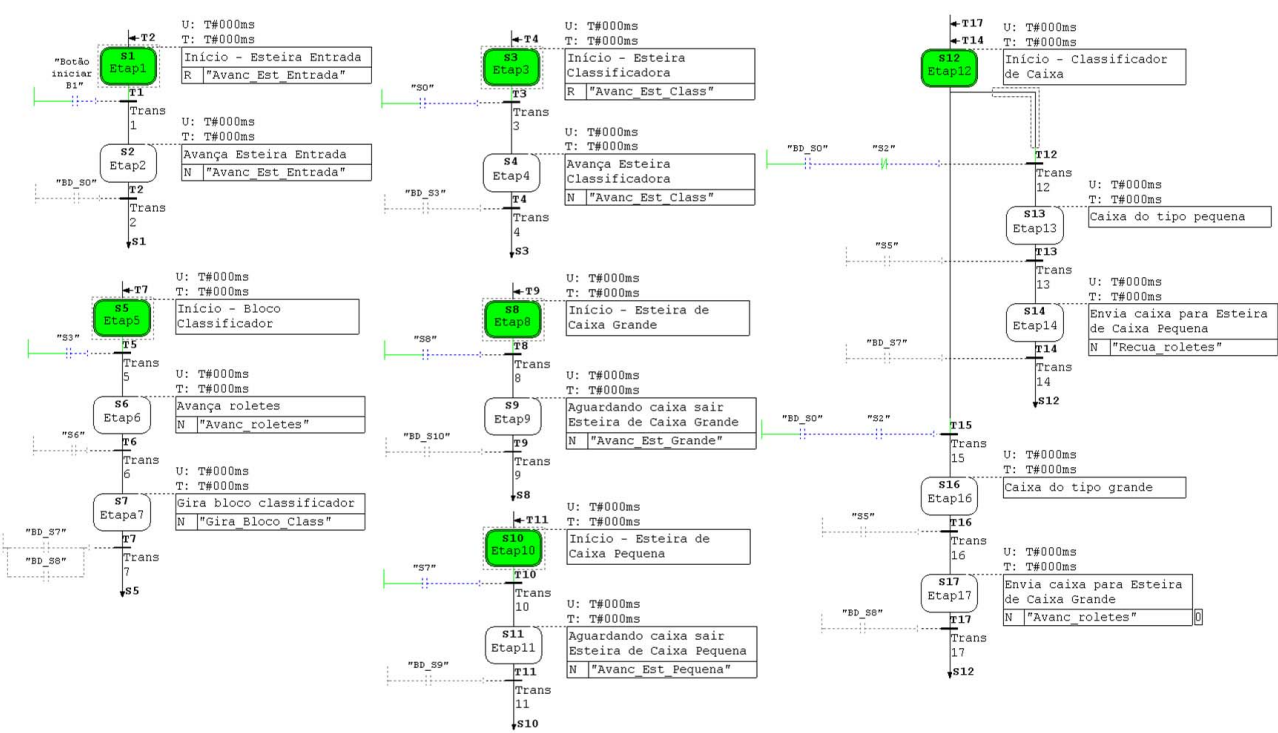


Fig. 15. Grafcet de controle do processo de Triagem de caixas

conhecido pelo projetista.

7ª Etapa: utilizando todos os Grafkets gerados nas etapas anteriores realiza-se a implantação em um CLP, que realizará o controle do processo. Desta forma, todos os Grafkets serão executados simultaneamente, proporcionando o controle individual de cada elemento do sistema, conforme mostrado na Figura 15.

VI. CONCLUSÃO

Este artigo apresentou uma metodologia para conversão de Rede de Petri Interpretada diretamente para uma linguagem de programação destinada aos Controladores Lógicos Programáveis. Tal metodologia mostrou-se útil para a síntese da lógica de controle de processos industriais, possibilitando a conversão de um formalismo com alto poder de abstração, que representa de forma eficiente o alto grau de paralelismo, para a linguagem de programação Grafcet.

Esta metodologia garante que as boas propriedades presentes no modelo do processo sejam mantidas nas etapas de criação dos Grafkets de controle, sintetizando uma lógica de controle livre de bloqueios e de fácil implementação, melhorando assim, a aplicabilidade das Redes de Petri Interpretada em projetos de automação industrial.

AGRADECIMENTOS

Agradecimento à CAPES-DS pelo auxílio financeiro.

REFERÊNCIAS

[1] C. Cassandras and S. Lafortune, *Introduction to discrete event systems*, ser. Springer Engineering. New York: Springer Science Business Media, 2008.

[2] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of The IEEE*, vol. 77, pp. 541–580, 1989.

[3] M. Uzam and A. H. Jones, "Discrete event control system design using automation petri nets and their ladder diagram implementation," *International Journal of Advanced Manufacturing Systems, special issue on Petri Nets Applications in Manufacturing Systems*, vol. 14, no. 10, pp. 716–728, oct 1998.

[4] V. Vlad, C. Ciufudean, A. Graur, and C. Filote, "An example of modeling manufacturing systems using petri nets and the iec 61499 standard," in *Proceedings of the 13th WSEAS international conference on Systems*, ser. ICS'09. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2009, pp. 357–363.

[5] M. V. Moreira, D. S. Botelho, and J. C. Basilio, "Ladder diagram implementation of control interpreted petri nets: a state equation approach," *IFAC Discrete-Event System Desing*, vol. 1, pp. 85–90, 2009.

[6] A. Ramirez-Trevino, I. Rivera-Rangel, and E. Lopez-Mellado, "Observability of discrete event systems modeled by interpreted petri nets," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 4, pp. 557–565, aug. 2003.

[7] IEC, "IEC 61131-3, 2nd Ed. Programmable Controllers - Programming Languages," *International Electrotechnical Commission*, 2003.

[8] R. David, "Grafcet: a powerful tool for specification of logic controllers," *Control Systems Technology, IEEE Transactions on*, vol. 3, no. 3, pp. 253–268, sep 1995.

[9] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri nets*, 2nd ed. Berlin: Springer, 2010.

[10] —, *Petri nets and Grafcet: tools for modelling discrete event systems*. New York: Prentice Hall, 1992.

[11] B. T. Vigário, A. P. Magalhães, and F. T. Freitas, "Modern computer games technology in systems and control education," *7th Portuguese Conference on Automatic Control*, 2006.

[12] B. Riera, P. Marange, F. Gellot, O. Nocent, A. P. Magalhães, and B. Vigário, "Complementary usage of real and virtual manufacturing systems for safe plc training," *The 8th IFAC Symposium on Advances in Control Education*, 2009.

[13] B. Berthomieu and F. Vernadat, "Time petri nets analysis with tina," in *Quantitative Evaluation of Systems, 2006. QEST 2006. Third International Conference on*, sept. 2006, pp. 123–124.