



**FACULDADE DE TECNOLOGIA SENAI CIMATEC  
CURSO DE ESPECIALIZAÇÃO EM  
AUTOMAÇÃO, CONTROLE E ROBÓTICA**

**SIMULAÇÃO DA ESTAÇÃO DE  
MONTAGEM DA PLANTA CIM  
UTILIZANDO-SE DE REDE DE PETRI  
INTERPRETADA**

**ADRIELLE TAVARES COITINHO**

**2011**

**FACULDADE DE TECNOLOGIA SENAI CIMATEC  
CURSO DE ESPECIALIZAÇÃO EM  
AUTOMAÇÃO, CONTROLE E ROBÓTICA**

**ADRIELLE TAVARES COITINHO**

**SIMULAÇÃO DA ESTAÇÃO DE  
MONTAGEM DA PLANTA CIM  
UTILIZANDO-SE DE REDE DE PETRI  
INTERPRETADA**

Trabalho apresentado ao curso de Especialização em Automação, Controle e Robótica da Faculdade de Tecnologia SENAI CIMATEC.

Orientador: Prof. Eduardo José Lima II

**SALVADOR  
2011**

## **DEDICATÓRIA**

**Aos meus pais**

## **AGRADECIMENTOS**

Agradeço aos meus pais por estarem sempre ao meu lado em todos os desafios da minha vida.

Ao professor Eduardo pela orientação e apoio.

A todos que direta ou diretamente contribuíram para o desenvolvimento desse trabalho.

## RESUMO

Esse trabalho tem como objetivo criar um simulador para a estação de montagem da planta CIM do Senai-Cimatec – Bahia em sistema supervisorio (Eclipse E3) utilizando-se de redes de Petri Interpretada. O simulador permitirá a visualização da integração da estação de montagem com o sistema de controle da planta CIM sem colocar em risco o sistema real. Devido a ausência de um CLP na estação de montagem, a comunicação entre os dois sistemas é feita através de entradas e saídas digitais que podem ser facilmente representadas utilizando-se das propriedades da rede de Petri Interpretada.

**Palavras-chave:** Supervisão, Redes de Petri Interpretada e Eclipse E3.

## LISTA DE FIGURAS

Figura 1 – rede de Petri .....	7
Figura 2 - Grafo de marcações acessíveis.....	11
Figura 3 - Transição quase viva e não viva.....	13
Figura 4 – Grafo das marcações (transição quase viva).....	13
Figura 5 - Uma rede de Petri representando uma atividade paralela .....	14
Figura 6 - Representação de uma situação de conflito .....	15
Figura 7 – Interação da RdP com o ambiente externo.....	17
Figura 8 - Temporização da rede de Petri (T-Temporizada) .....	18
Figura 9 - Temporização da rede de Petri (P-Temporizada).....	19
Figura 10 – O problema dos filósofos: a) utilizando uma RdP Ordinária .....	21
Figura 11 – Área de Telas .....	26
Figura 12 – Arquitetura Planta CIM .....	28
Figura 13 – Porta Objeto .....	29
Figura 14 – Estação de Montagem.....	29
Figura 15 – Etapa de reconhecimento e recebimento das peças .....	32
Figura 16 – Etapa de encaminhamento das peças da esteira transportadora para o armazém e o processo de montagem .....	33
Figura 17 – Etapa de devolução do produto final e paletes para a planta.....	34
Figura 18 – Etapa de recebimento das peças e encaminhamento para o armazém – Elipse E3 .....	36
Figura 19 – Etapa da montagem do Porta Objetos – Elipse E3 .....	36
Figura 20 – Etapa de devolução do produto final e paletes para a planta – Elipse E3 .....	37
Figura 21 – Propriedades do CheckBox_Presença_Vagão – Elipse E3.....	37
Figura 22 – Propriedades da FICHA_INICIO – Elipse E3.....	38
Figura 23 – Propriedade da FICHA_L2 – Elipse E3 .....	38
Figura 24 – Propriedade da FICHA_L3 – Elipse E3 .....	39
Figura 25 – Propriedade da FICHA_ROBO – Elipse E3 .....	39
Figura 26 – Propriedade da FICHA_L5 – Elipse E3 .....	39
Figura 27 – Propriedade do CheckBox_Robo_Exec_PRG1 – Elipse E3.....	40
Figura 28 – Propriedade da FICHA_L6 – Elipse E3 .....	41

Figura 29 – Propriedade da FICHA_L7 – Elipse E3 .....	41
Figura 30 – Propriedade do CheckBox_Robo_Exec_PRG10 – Elipse E3 .....	42

# SUMÁRIO

<b>CAPÍTULO 1: INTRODUÇÃO .....</b>	<b>1</b>
1.1 OBJETIVO .....	5
1.2 OBJETIVOS ESPECÍFICOS.....	5
1.3 METODOLOGIA.....	5
<b>CAPÍTULO 2: REDE DE PETRI (RDP) .....</b>	<b>6</b>
2.1 PRINCIPAIS ELEMENTOS.....	6
2.2 DEFINIÇÕES: FORMAL E GRÁFICA.....	7
<b>2.2.1 Definição Formal.....</b>	<b>7</b>
<b>2.2.2 Rede Marcada .....</b>	<b>7</b>
2.3 REPRESENTAÇÃO MATRICIAL .....	8
2.4 EVOLUÇÃO DA UMA REDE.....	9
<b>2.4.1 Transição Sensibilizada.....</b>	<b>9</b>
<b>2.4.2 Disparo de uma Transição.....</b>	<b>9</b>
<b>2.4.3 Seqüência de Disparo .....</b>	<b>10</b>
<b>2.4.4 Conjunto de marcações acessíveis.....</b>	<b>11</b>
2.5 PROPRIEDADES .....	12
<b>2.5.1 Propriedades Comportamentais.....</b>	<b>12</b>
<b>2.5.2 Propriedades Estruturais.....</b>	<b>14</b>
2.6 VARIAÇÕES DA REDE DE PETRI (RdP) ORDINÁRIA .....	15
<b>2.6.1 RdP Interpretadas .....</b>	<b>16</b>
<b>2.6.2 RdP T-Temporizada.....</b>	<b>18</b>
<b>2.6.3 RdP P-Temporizada .....</b>	<b>18</b>
<b>2.6.4 RdP Estocástica .....</b>	<b>19</b>
<b>2.6.5 RdP Temporal.....</b>	<b>20</b>
<b>2.6.6 RdP Colorida.....</b>	<b>20</b>
<b>2.6.7 RdP Predicado-Transição.....</b>	<b>21</b>
<b>2.6.8 RdP a Objeto.....</b>	<b>22</b>
<b>CAPÍTULO 3: SISTEMA SCADA.....</b>	<b>23</b>
3.1 ARQUITETURA .....	24
3.2 ELIPSE E3.....	25
<b>CAPÍTULO 4: DESENVOLVIMENTO.....</b>	<b>28</b>
4.1 PLANTA CIM .....	28
4.2 ESTAÇÃO DE MONTAGEM .....	29
4.3 MODELAGEM DA ESTAÇÃO DE MONTAGEM .....	30
<b>4.3.1 Etapa de reconhecimento e recebimento das peças .....</b>	<b>32</b>



4.3.2	<i>Etapa de encaminhamento das peças da esteira transportadora para o Armazém e o processo de montagem.....</i>	<b>33</b>
4.3.3	<i>Etapa de devolução do produto final e paletes para a planta .....</i>	<b>34</b>
<b>CAPÍTULO 5: RESULTADOS.....</b>		<b>35</b>
5.1	<i>Script relacionado à etapa de recebimento das peças e encaminhamento para o armazém</i>	<b>37</b>
<b>CAPÍTULO 6: CONCLUSÕES .....</b>		<b>43</b>
<b>CAPÍTULO 7: ANEXO.....</b>		<b>44</b>
<b>CAPÍTULO 8: REFERÊNCIAS BIBLIOGRÁFICAS.....</b>		<b>52</b>

## CAPÍTULO 1: INTRODUÇÃO

Mudanças significativas têm ocorrido no projeto de sistemas de manufatura, motivadas pelo o aumento no número e variedade de produtos e pelas exigências de mais precisão e melhor qualidade. Essas tendências exigem melhorias contínuas do produto além de um sistema capaz de ser projetados para ser flexível, compreensível, confiável, capaz de produzir produtos de elevada qualidade com custos e tempo reduzidos [FERREIRA, 2010].

No contexto atual de automação são buscadas cada vez mais novas metodologias para a supervisão e controle nos mais variados tipos de ambientes, além das indústrias, para atender as exigências do mercado. Há algumas décadas o advento dos controladores lógico-programáveis (CLPs) determinou um grande passo no crescimento da automação da indústria, e eles se tornaram uma importante ferramenta na área de controle dos sistemas, estando presentes na maioria dos ambientes automatizados [COSTA, 2009]. Com o intuito de padronizar os símbolos gráficos, a seqüência dos eventos, os requisitos dos equipamentos, as etapas de projeto e as linguagens de programação dos CLPs, foi criado um comitê internacional da IEC, que teve como resultado a norma IEC 61131 que foi dividida em 8 partes. A terceira parte, conhecida como IEC 61131-3, trata-se das linguagens de programação que foram definidas em cinco tipos [ROSÁRIO, 2007]:

- IL (*Instruction list* ou lista de instruções)
- ST (*Structured text* ou texto estruturado)
- LD (*Ladder diagram* ou diagrama Ladder)
- FBD (*Function block diagram* ou diagrama de blocos de função)
- SFC (*Sequential function chart* ou diagrama funcional seqüencial), também conhecido como Grafcet.

Dentre essas cinco linguagens, a LD e o SFC são as mais utilizadas na programação de CLPs embora as mesmas possuam algumas limitações:

### Limitações do Ladder [ZHOU, 1995]:

*“...quanto maior o sistema de controle, mais difícil se torna determinar as especificações iniciais de projeto (como o sistema opera) examinando a lógica de controle. Isto torna o diagnóstico de problemas neste sistema difícil.*

*Programas Ladder curtos (menos de 100 linhas) não são difíceis de entender. Entretanto, conforme o sistema se torna mais complexo, ele pode facilmente crescer para 1000 linhas ou mais, com as seguintes conseqüências:*

- 1. A complexidade para validar e entender a lógica cresce exponencialmente (em geral).*
- 2. Torna-se mais difícil fazer modificações no programa para refletir mudanças das especificações funcionais do sistema.*
- 3. A tarefa de diagnóstico e manutenção do programa torna-se mais difícil.”*

### Limitações do Grafcet [LIMA, 2002]:

*“...todas as transições simultaneamente habilitadas são disparadas ao mesmo tempo. Na rede de Petri (que será abordada mais adiante) se várias transições estiverem em conflito e estiverem simultaneamente habilitadas, tanto pela marcação da rede como por sua interpretação, somente uma delas pode disparar. Assim o comportamento dessa rede de Petri é não determinístico. Porém, se em uma rede de Petri Interpretada sua interpretação for tal que, para qualquer marcação, todas as transições em conflito tiverem condições de disparo que nunca serão verdadeiras ao mesmo tempo, seu comportamento será determinístico...”*

Quando se trabalha com sistemas automatizados complexos, surge a necessidade de se criar uma interface de maneira a facilitar o trabalho da equipe da operação e manutenção do sistema. Nos trabalhos de implantação e testes, é muito difícil avaliar o que está acontecendo pela análise do funcionamento da planta, diretamente pelo programa do CLP [MORAES, 2007]. Assim surge a necessidade da criação de uma interface amigável.

Dentre as metodologias encontradas para supervisão e controle nos mais variados tipos de ambientes os ferramentais matemáticos e gráficos se tornaram aliados da automação, na busca de estruturas formais que garantam o funcionamento dos sistemas de uma forma completa e individual. Entre os formalismos encontrados no mundo da automação destacam-se as redes de Petri, que consistem em uma importante ferramenta matemática e gráfica que permite a modelagem de um sistema, visualização e verificação de modelo, entre outras vantagens [COSTA, 2009]. A rede de Petri foi proposta por Carl Petri, em 1962, para representar sistemas a eventos discretos [CARDOSO, 1997]. Uma rede de Petri é facilmente transformada em linguagem de CLP, seja Ladder ou Grafcet. Dentre os diversos tipos de redes de Petri têm-se as redes de Petri: Ordinárias, Interpretadas, Colorida, Predicado-Transição, a Objetos, Temporizada, Temporais e Estocásticos.

A proposta desse projeto implica em criar um simulador para supervisionar a estação de montagem da planta CIM do Senai-Cimatec Bahia utilizando redes de Petri Interpretadas e o Elipse E3, onde será possível simular a interação da estação com elementos externos, tais como o CLP, e verificar situações de colisão devido ao comportamento indevido do sistema de controle. Esse simulador é um exemplo de que é possível simular a integração de uma planta com outros sistemas sem colocar em risco o sistema real.

Conforme citado por ROSÁRIO [2007], a simulação de um sistema pode ser definida como a capacidade de projetar o modelo de um sistema real e conduzir experimentos com esse modelo a fim de compreender o comportamento do sistema e avaliar estratégias para a sua operação. A simulação de sistemas normalmente é utilizada:

- Na impossibilidade de realização experimental com o sistema real.
- Quando a experimentação utilizando o sistema real é indesejável.
- Para a compressão ou expansão da escala de tempo.
- Para a avaliação do desempenho do sistema.
- Para treinamento e instrução.

A modelagem a eventos discretos, utilizando redes de Petri Interpretadas, fornece um modelo fiel do comportamento dinâmico e das relações de dependências entre os recursos, servindo como base para o desenvolvimento dos

códigos de controle supervísório no software Elipse E3 [SANTOS, 2007]. A modelagem é uma ferramenta de comunicação e documentação, é um meio de se lidar com sistemas complexos, é uma forma que permite analisar o sistema e detectar possíveis falhas além de ajudar entender melhor o seu funcionamento. Nesse trabalho optou-se por utilizar o software Elipse E3 por ser um software com boa documentação disponível e pela experiência de uso adquirida durante o curso. A existência do CLP será representada por uma simulação no próprio software.

## **1.1 OBJETIVO**

Criar um simulador para a estação de montagem da planta CIM do Senai-Cimatec – Bahia, com o intuito de simular a interação da estação com elementos externos, tais como o CLP, e verificar situações de colisão devido ao comportamento indevido do sistema de controle.

## **1.2 OBJETIVOS ESPECÍFICOS**

- Decompor as ações executadas pela Estação de Montagem.
- Modelar a Estação estudada utilizando rede de Petri Interpretada.
- Criar módulo de gerenciamento da Estação utilizando-se do software supervisor Elipse E3.
- Simular os possíveis eventos indesejáveis ao funcionamento correto da planta.

## **1.3 METODOLOGIA**

Com a finalidade de alcançar o objetivo desse trabalho foi feito inicialmente um estudo teórico sobre as redes de Petri e o sistema SCADA.

No capítulo 2 serão apresentadas as características da rede de Petri além de suas variações, o que ajudou a definir sobre a utilização da rede de Petri Interpretada na modelagem da Estação de Montagem. No capítulo 3 serão apresentadas as características do sistema SCADA.

De posse desses conhecimentos teóricos e do entendimento do funcionamento da Estação foi possível modelá-la e criar o simulador no Elipse E3, o que será apresentado no capítulo 4. No Capítulo 5 e 6 serão apresentados os resultados e as conclusões respectivamente, enquanto que no Anexo serão apresentados as implementações feitas no Elipse E3 na construção do simulador.

## **CAPÍTULO 2: REDE DE PETRI (RdP)**

A rede de Petri é uma ferramenta gráfica e matemática que se adapta bem a um grande número de aplicações em que as noções de eventos e de evoluções simultâneas são importantes. Foi proposto por Carl Petri para modelar a comunicação entre autômatos, utilizados, na época, para representar sistemas a eventos discretos [CARDOSO, 1997]. Sistemas a eventos discretos são sistemas modelados de tal forma que as variáveis de estado variam bruscamente em instantes determinados e que os valores das variáveis nos estados seguintes podem ser calculados diretamente a partir dos valores precedentes e sem ter que considerar o tempo entre estes dois instantes.

### **2.1 PRINCIPAIS ELEMENTOS**

Os principais elementos das redes de Petri são os lugares, as transições, as marcas e os arcos orientados.

- a. Lugares (representados por um círculo): Pode ser interpretado como uma condição, um estado parcial, uma espera, um procedimento, um conjunto de recurso, um estoque, uma posição geográfica num sistema de transporte, etc. Em geral, todo lugar tem um predicado associado, por exemplo, *máquina livre*, *peça em espera* (Figura 1) [CARDOSO, 1997].
- b. Transições (representadas por uma barra ou um retângulo): é associada a um evento que ocorre no sistema, como o evento *iniciar a operação* (Figura 1) [CARDOSO, 1997].
- c. Marca ou Ficha (representado por um ponto no lugar): É um indicador significado que a condição associada ao lugar é verificada. Pode representar um objeto (recurso ou peça) numa certa posição geográfica (num determinado estado), ou ainda uma estrutura de dados que se manipula. Por exemplo, uma ficha no lugar *máquina livre* indica que a máquina está livre (predicado verdadeiro). Se não tem fichas neste lugar, o predicado é falso, por conseguinte a máquina não está livre. Se no lugar *peças em espera* houvesse três fichas, indicaria que existem três peças em espera.
- d. Arcos Orientados (representados por uma seta): Conectam lugares e transições.

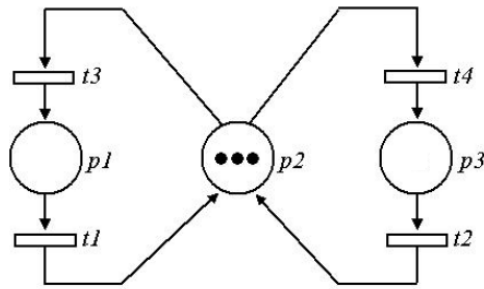


Figura 1 – rede de Petri  
(Fonte: CARDOSO, 1997)

O modelo da rede de Petri da figura 1 será utilizado como exemplo nas definições apresentadas nos próximos itens.

## 2.2 DEFINIÇÕES: FORMAL E GRÁFICA

### 2.2.1 Definição Formal

Segundo [MURATA, 1989] uma rede de Petri é uma quintupla:

- $P = \{p_1, p_2, \dots, p_m\}$  é um conjunto finito de lugares.
- $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições.
- $F$  é um conjunto de arcos.
- $W : F \rightarrow \{1, 2, 3, \dots\}$  é a função do peso.
- $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  é a marcação inicial.

### 2.2.2 Rede Marcada

Uma rede marcada  $N$  é uma dupla:

$$N = \langle R, M_0 \rangle \quad (2.1)$$

onde,

$R$  é uma rede de Petri.

$M_0$  é a marcação inicial dada pela aplicação.

$$M: P \rightarrow \mathbb{N} \quad (2.2)$$

O número de fichas existente no lugar  $p$  é representado por  $M(p)$ , sendo  $M(p) \geq 0$ . A marcação  $M$  de todos os lugares é representada por um vetor coluna de dimensão igual à quantidade de lugares. A marcação representa o esta-



do do sistema descrito pela rede de Petri num determinado instante. O disparo de transições ocasionam uma evolução da marcação, o que corresponde a evolução do estado da rede. O vetor coluna da marcação dos lugares da rede de Petri da Figura 1 é dado por:

$$M_0 = [0 \ 3 \ 0]^T \quad (2.3)$$

onde, T é a transposta do vetor.

### 2.3 REPRESENTAÇÃO MATRICIAL

Defini-se:

- $P = \{p_1, p_2, p_3, \dots\}$  o conjunto finito de lugares.
- $T = \{t_1, t_2, t_3, t_4, \dots\}$  o conjunto finito de transições.
- $\text{Pré} = (P \times T) \rightarrow \{0, 1, 2, 3, \dots\}$  a matriz de dimensão  $n \times m$  que define os arcos que conectam lugares a transições e seus respectivos pesos. Se  $\text{Pré}(P_i \times T_j) > 0$  existe pelo menos 1 arco conectando o lugar  $P_i$  à transição  $T_j$ .
- $\text{Pós} = (P \times T) \rightarrow \{0, 1, 2, 3, \dots\}$  a matriz  $n \times m$  que define os arcos que conectam transições a lugares e seus respectivos pesos. Se  $\text{Pós}(P_i \times T_j) > 0$  existe pelo menos 1 arco conectando a transição  $T_j$  ao lugar  $P_i$ .
- $n = \text{número de linhas} = \text{número de lugares}$  e  $m = \text{número de colunas} = \text{número de transições}$ .

A partir da figura 1, abaixo, tem-se:

$$\text{Pré} = \begin{array}{cccc|c} & t_1 & t_2 & t_3 & t_4 & \\ \hline & 1 & 0 & 0 & 0 & p_1 \\ & 0 & 0 & 1 & 1 & p_2 \\ & 0 & 1 & 0 & 0 & p_3 \end{array} \quad \text{Pós} = \begin{array}{cccc|c} & t_1 & t_2 & t_3 & t_4 & \\ \hline & 0 & 0 & 1 & 0 & p_1 \\ & 1 & 1 & 0 & 0 & p_2 \\ & 0 & 0 & 0 & 1 & p_3 \end{array}$$

Defini-se a matriz de incidência C a partir de Pré e Pós:

$$C = \text{Pós} - \text{Pré} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & 1 & -1 & -1 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

A matriz de incidência  $C$  corresponde a estrutura da rede de Petri, independente do estado. Cada coluna corresponde à maneira como a rede vai se comportar quando do disparo da transição associada àquela coluna [PALOMINO, 2005]. Na matriz  $C$  encontrada a partir da rede de Petri da Figura 1, a coluna referente à transição  $t_1$  demonstra que quando esta é disparada é retirada uma ficha no lugar  $p_1$ , colocada uma ficha do lugar  $p_2$  e não há alteração no lugar  $p_3$ .

## 2.4 EVOLUÇÃO DA UMA REDE

### 2.4.1 Transição Sensibilizada

Uma transição  $t$  está sensibilizada ou habilitada se e somente se:

$$\forall p \in P, M(p) \geq \text{Pre}(p, t) \quad (2.4)$$

isto é, se o número de fichas do lugar  $p$  for maior ou igual ao peso do arco que liga este lugar à transição  $t$ .

### 2.4.2 Disparo de uma Transição

Numa rede de Petri, uma das condições para uma transição disparar é que a mesma esteja sensibilizada. Disparando uma transição  $t$  sensibilizada, uma nova marcação  $M'$  é obtida, tal que:

$$M'(p) = M_0(p) - \text{Pré}(p, t) + \text{Pós}(p, t) \quad (2.5)$$

$$M'(p) = M_0(p) + C(p, t)$$

Na rede de Petri da Figura 1, após o disparo da transição  $t_3$  tem-se a partir da equação (2.5):

$$M'(p) = M_0(p) + C(p, 3)$$

$$C(p, 3) = \text{Pós}(p, 3) - \text{Pré}(p, t)$$

$$C(p, 3) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

$$M'(p) = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

### 2.4.3 Seqüência de Disparo

Na rede de Petri da Figura 1 as transições  $t_3$  e  $t_4$  estão sensibilizadas pela marcação inicial  $M_0 = [0 \ 3 \ 0]^T$  e podem disparar levando a uma marcação  $M_1$ . Supondo que a transição  $t_3$  dispare  $M_1 = [1 \ 2 \ 0]^T$ . Após seu disparo, a transição  $t_4$ , ainda sensibilizada, dispare levando a marcação  $M_2 = [1 \ 1 \ 1]^T$ . Defini-se como *seqüência de disparo* ( $s$ ) a ordem de disparo das transições que leva a uma determinada marcação. Portanto:

- Para  $M_0 \xrightarrow{t_3} M_1, s = t_3$
- Para  $M_1 \xrightarrow{t_4} M_2, s = t_4$
- Para  $M_0 \xrightarrow{t_3 t_4} M_3, s = t_3 t_4$

Diz-se que a seqüência  $s = t_3$  é disparável, a partir da marcação  $M_0$ , com a notação  $M_0 \xrightarrow{t_3} M_1$ .

Um *vetor característico* de dimensão  $n$ , onde  $n$  é igual ao número de transições existentes na rede de Petri, é associado à seqüência de disparo e cada elemento  $s(t)$  representa a quantidade de vezes que a transição  $t$  foi disparada.

No exemplo anterior, o vetor característico da seqüência  $s = t_3 t_4$  é dado por:

$$s(t) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Os valores nulos de  $s(t_1) = 0$  e  $s(t_2) = 0$  indicam que as transições  $t_1$  e  $t_2$  não dispararam nesta seqüência, enquanto que,  $s(t_3) = s(t_4) = 1$  indica que as transições  $t_3$  e  $t_4$  dispararam uma única vez nesta seqüência.

A execução de uma rede de Petri resulta em dois tipos de seqüência: uma seqüência de disparo  $s$  e outra de marcações ( $M_0, M_1, M_2, \dots$ ). A evolução de uma rede de Petri para uma seqüência  $s$  é dada, então, pela equação [PALOMINO, 2005]:

$$M' = M - \text{Pré}.S + \text{Pós}.S \quad (2.6)$$

$$M' = M + S(\text{Pós} - \text{Pré})$$

$$M' = M + C.S \quad , \quad M \geq 0, s \geq 0 \quad (2.7)$$

A equação 2.7 é chamada *equação fundamental* da rede de Petri.

A caracterização da marcação na teoria da rede de Petri por meio de um vetor  $M$  é precisa (fornece todas as informações sobre o estado). Já a representação de uma seqüência  $s$  através do vetor  $\mathbf{s}$  não considera a ordem de disparo das transições. A ordem de disparo é perdida, existindo, por conseguinte, uma perda de informação a respeito da evolução da rede [CARDOSO, 1997]. Portanto, a existência de um vetor característico  $\mathbf{s}$ , solução da equação 2.7, não garante que a seqüência  $s$  possa ser realmente disparada e que exista de fato a marcação  $M'$ . É necessário que a marcação inicial seja tal que as transições sejam de fato disparadas para cada marcação intermediária.

#### 2.4.4 Conjunto de marcações acessíveis

O conjunto de marcações acessíveis  $A(R, M)$  de uma rede de Petri marcada é o conjunto das marcações que podem ser atingidas a partir da marcação inicial, através de uma seqüência de disparo. Se este conjunto é finito pode-se representá-lo sob a forma de um grafo  $GA(R, M)$ , cujos nós são as marcações acessíveis do conjunto  $A(R, M)$ . Um arco orientado liga dois nós  $M_i$  e  $M_j$  se existe uma transição  $t$  sensibilizada que permite passar de uma marcação  $M_i$  a uma outra  $M_j$  [CARDOSO, 1997].

O grafo de marcações acessíveis da Figura 1 é representado na figura abaixo:

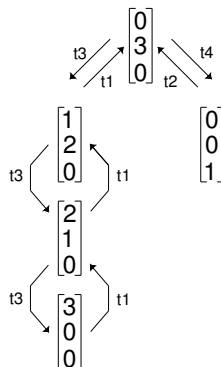


Figura 2 - Grafo de marcações acessíveis

(Fonte: CARDOSO, 1997)

## 2.5 PROPRIEDADES

As principais propriedades relativas às redes de Petri podem ser divididas em dois tipos: o primeiro se refere à dinâmica da rede, ou seja, à rede de Petri marcada (dependente de marcação inicial). O segundo é referente à parte estrutural da rede ou rede de Petri não marcada (independente de marcação inicial) [CARDOSO, 1997].

### 2.5.1 Propriedades Comportamentais

- **Alcançabilidade**

Se existir uma seqüência de disparos “ $s$ ” que leve a uma marcação  $M'$  a partir da marcação inicial  $M_0$  então esta é uma marcação alcançável. Se todas as marcações alcançáveis forem atingidas a partir da marcação inicial a rede de Petri é dita *alcançável* [CREPALDI, 2006].

- **Limitação**

Considerando um número inteiro  $k$  qualquer, uma rede de Petri é *limitada* se em cada um dos lugares da rede a quantidade de fichas não ultrapassar o valor de  $k$ . Então, a rede é dita *k-limitado* [CREPALDI, 2006].

- **Vivacidade**

**Transição quase viva:** Uma transição é quase viva se é possível sensibilizá-la por uma marcação  $M'$  a partir da marcação inicial  $M_0$ . Diz-se que  $t$  pode ser sensibilizada a partir de  $M$  através do disparo da seqüência  $s$ .

**Transição viva:** Uma transição  $t$  deve poder ser sensibilizada a partir de qualquer marcação  $M'$ , através de uma seqüência  $s$  de disparo.

Observando a Figura 3 e o seu grafo correspondente na Figura 4, percebe-se que, para a marcação inicial  $M_0 = [0 \ 0 \ 1 \ 1]^T$ , a transição  $d$  é quase viva pois só pode ser disparada uma única vez e não viva porque após este disparo a transição  $d$  é bloqueada, não sendo mais sensibilizada por nenhuma transição. Já as transições  $a$ ,  $b$ ,  $c$ ,  $f$  são vivas pois para qualquer marcação acessível a

partir de  $M_0$  é possível encontrar uma seqüência  $s$  de disparo que leva a uma marcação que as sensibiliza.

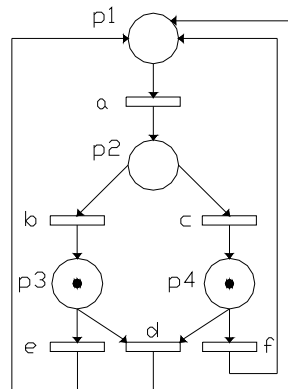


Figura 3 - Transição quase viva e não viva  
(Fonte: CARDOSO, 1997)

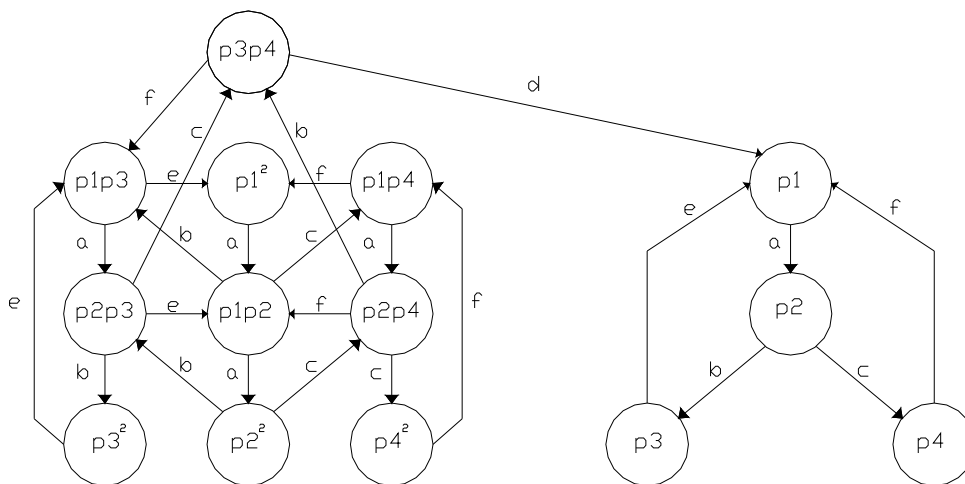


Figura 4 – Grafo das marcações (transição quase viva)  
(Fonte: CARDOSO, 1997)

**Rede marcada viva:** Uma rede de Petri marcada é viva se e somente se todas as suas transições são vivas.

É importante observar que:

- ✓ Somente é considerada a rede de Petri marcada.
- ✓ Uma rede de Petri viva garante que nenhum bloqueio pode ser provocado pela estrutura da rede (deadlocks). No entanto, ela não prova a

ausência de eventuais bloqueios provocados por uma má interação entre a rede de Petri e seu ambiente externo (a exemplo da rede de Petri interpretada).

- ✓ Uma rede de Petri viva garante também a ausência de partes mortas (nunca atingidas) desde que não haja uma má interpretação entre a rede e o seu ambiente externo.

A partir do grafo de marcações acessíveis, Figura 2, da rede de Petri da Figura 1, observa-se que suas transições são vivas e, portanto, trata-se de uma rede de Petri marcada viva para a marcação inicial  $M_0 = [0 \ 3 \ 0]^T$ .

- **Reiniciabilidade**

Uma rede de Petri é dita reiniciável se, a partir de qualquer marcação acessível  $M'$  pertencente à  $GA(R, M)$  existir uma seqüência de disparo “s” que possibilite à rede retornar à sua marcação inicial  $M_0$ .

## 2.5.2 Propriedades Estruturais

- **Paralelismo**

Atividades paralelas ou concorrentes podem ser facilmente expressadas em termos de redes de Petri. Por exemplo, na Figura 5, as atividades paralelas ou concorrentes são representadas pelas transições  $t_2$  e  $t_3$ . Em geral duas transições são ditas concorrentes se elas ocorrem independentemente, isto é, uma transição pode disparar antes ou depois ou em paralelo com a outra, como no caso de  $t_2$  e  $t_3$  [PALOMINO, 2005].

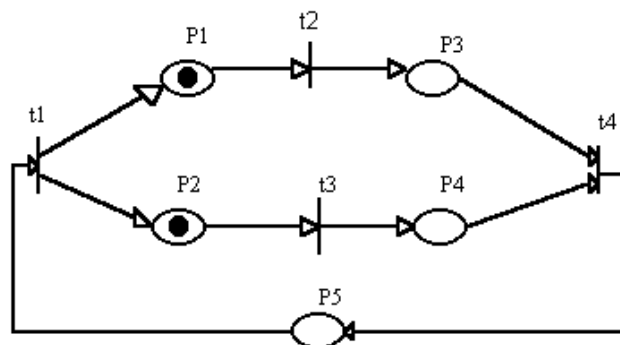


Figura 5 - Uma rede de Petri representando uma atividade paralela  
(Fonte: PALOMINO, 2005)

- **Conflito**

Um conflito (escolha ou decisão) numa rede de Petri (Figura 6 ) é uma situação que pode levar a uma disputa não determinística de recursos (fichas) [PALOMINO, 2005]. Por ocorrência do conflito entende-se a existência, num dado estado, de duas (ou mais) possibilidades, excludentes, de evolução [CARDOSO, 1997].

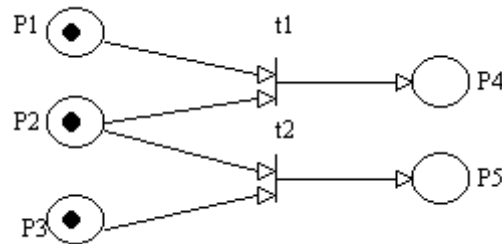


Figura 6 - Representação de uma situação de conflito  
(Fonte:PALOMINO, 2005)

## 2.6 VARIAÇÕES DA REDE DE PETRI (RdP) ORDINÁRIA

Os conceitos e propriedades das redes de Petri até então apresentadas utilizou-se do modelo original da rede conhecida como rede de Petri Ordinária. Desde a apresentação desse primeiro modelo, foram desenvolvidos vários trabalhos práticos e teóricos sobre RdP que levaram ao desenvolvimento de muitas variantes do modelo tendo em vista aplicações mais específicas [CREPALDI, 2006].

Conforme explicitado por PALOMINO [1995] as RdP Ordinárias possuem baixo poder de modelagem por representarem apenas relações de causa e efeito entre os eventos e as condições. A sua utilização é restringida, portanto, a diversos tipos de sistemas pertencentes a classe de sistemas (dinâmicos) de eventos discretos, onde sincronização externa e o tempo não intervêm. Numa *RdP Ordinária* todos os arcos têm o mesmo peso o qual é 1, existindo apenas um só tipo de ficha, a capacidade de lugares é infinita (isto é, o número de fichas não é limitado pela capacidade dos lugares), o disparo de uma transição pode acontecer se cada lugar precedente tiver no mínimo uma ficha, e nenhum tempo é envolvido.



No intuito de resolver tais limitações foram desenvolvidos, a partir do modelo original, variações das RdP Ordinárias, também chamadas de primitivas ou autônomas. Dentre as variações da RdP tem-se:

- As Interpretadas.
- As Temporizadas, Temporais e Estocástica.
- As Coloridas.
- As Predicado-Transição.
- As A Objeto.

### **2.6.1 RdP Interpretadas**

Interpretar uma rede implica antes de tudo dar sentido concreto a um modelo matemático, associando, aos lugares, transições e fichas, os elementos existentes no sistema [CARDOSO, 1997].

Na rede de Petri Ordinária, a transição é disparada desde que esteja sensibilizada. A sensibilização é condição necessária e suficiente para o disparo: se não houver conflito a transição dispara. Já na rede de Petri Interpretada, são associadas variáveis às transições da rede, representando condições e ações existentes no sistema. Tais variáveis podem indicar o estado de atuadores, sensores, etc., permitindo, assim, modelar a interação com o ambiente externo [CARDOSO, 1997]. Portanto, para uma transição disparar em uma rede de Petri Interpretada é necessário que a mesma esteja sensibilizada e que as condições a ela associada sejam verdadeiras.

Às vezes certos tratamentos não são sistematicamente executados imediatamente após o término dos tratamentos precedentes. Suas execuções podem ser submetidas a condições que tratam com certos dados (instruções condicionais do tipo "if", por exemplo). Tal fenômeno será descrito por um lugar com várias transições de saída. Do ponto de vista da teoria da RdP, isto caracteriza uma situação de conflito. O disparo de uma transição em conflito ocorre quando a sua condição suplementar de disparo, cujo valor depende dos dados, for verdadeira [CARDOSO, 1997]. A condição de conflito pode ser facilmente resolvida utilizando rede de Petri Interpretada.

Segundo CARDOSO (1997), a rede de Petri Interpretada é uma rede não autônoma que introduz a interação com o ambiente ou dados contidos na ficha.

O mesmo afirma que modelar um sistema aberto (sistema comunicando-se em permanência com o ambiente externo) utilizando a RdP consiste em estruturá-lo em duas partes:

- a) **Controle:** descreve todos os encadeamentos potenciais de eventos e de atividades. Este controle é descrito pelas redes de Petri Ordinárias.
- b) **Dados:** é a parte operativa do sistema. Descreve ao mesmo tempo as estruturas de dados internas ao sistema e os cálculos que são feitos sobre estes dados, sem especificar em quais instantes eles são realizados. Além dos dados internos, esse cálculo intervém nas informações provenientes no mundo exterior.

A interpretação da RdP especifica as ligações entre a parte controle de um lado e a parte dados e ambiente externo de outro lado. A Figura 7, abaixo, descreve essa estruturação. A interpretação exprime-se da seguinte forma:

- Associação de condições às transições.
- Associação de ações às transições.
- Associação de tratamento aos lugares.

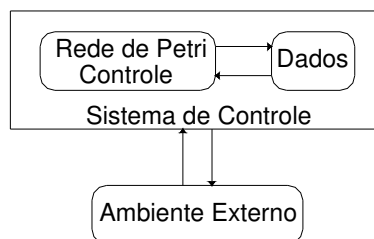


Figura 7 – Interação da RdP com o ambiente externo  
(Fonte: CARDOSO, 1997)

A RdP Interpretada oferece uma boa estruturação da parte *controle*. Entretanto, não existe um método para estruturar os *dados*.

Para obter maiores informações sobre o comportamento dinâmico da RdP Interpretada é necessário simulá-la. Embora a simulação seja um método de avaliação não exaustivo (pois é impossível prever todas as configurações possíveis das variáveis), é possível, ao menos, testar o comportamento do sistema modelado para as configurações previstas [CARDOSO E VALETTE, 1997].

### 2.6.2 RdP T-Temporizada

Foi apresentada, em 1973, por Ranchandani em sua tese de Doutorado. O princípio básico deste tipo de rede é a associação de um único parâmetro temporal a cada transição da rede de Petri Ordinária, representando assim a duração da sua sensibilização. Este parâmetro é o tempo, depois de disparada a transição, de retardo da disponibilização da ficha no lugar de saída da transição [CREPALDI, 2006].

CARDOSO E VALLENTE (1997) afirmam que o disparo não é instantâneo, mas possui uma duração. Esta associação de tempo à transição só tem sentido se a transição é interpretada como uma atividade e não como um evento instantâneo.

Na Figura 8 a seguir as duas redes são equivalentes e mostra a substituição de uma transição por uma seqüência *transição-lugar-transição*. Neste exemplo a transição  $t'_1$  corresponde ao início da atividade, o lugar  $p'$  a atividade sendo executada, com duração igual a *Tempo*, e a transição  $t''_1$  o fim da atividade.

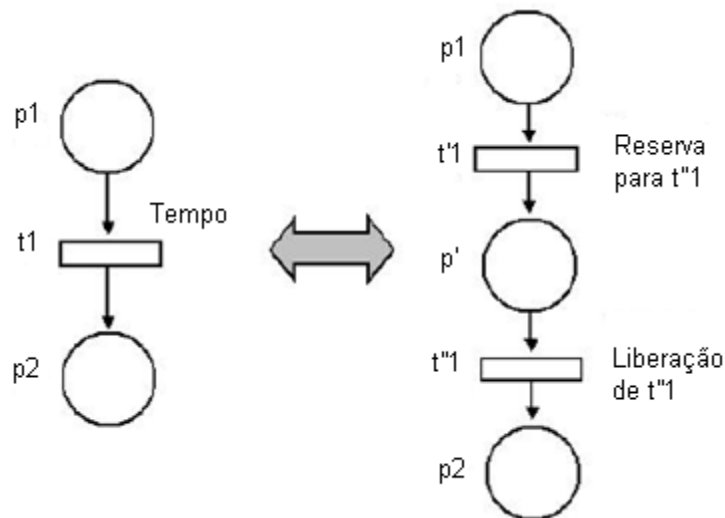


Figura 8 - Temporização da rede de Petri (T-Temporizada)  
(Fonte: CARDOSO, 1997)

### 2.6.3 RdP P-Temporizada

Difere da RdP T-Temporizada quanto a associação de um período a cada lugar da rede ao invés das transições. Este período determina o tempo gasto para se executar a atividade representada pelo lugar.

Suponha que as redes da Figura 9 representem a execução de uma atividade de A. Ambas as redes são equivalentes e mostram a substituição de um lugar por uma seqüência *lugar-transição- lugar*. Neste exemplo, sendo  $p_1$  a atividade com duração igual a Tempo, o lugar  $p'_1$  corresponde à atividade sendo executada, a transição  $t'$  corresponde ao evento *tempo transcorrido* e o lugar  $p''_1$  corresponde a uma eventual espera (sincronização com outras atividades) após o final da atividade A. Enquanto a atividade A está sendo executada, a ficha não pode ser utilizada para disparar a transição. A ficha é dita *indisponível*. Desde que a atividade estiver acabada, a ficha se torna *disponível* e  $t_2$  pode eventualmente ser disparada. Somente as fichas tidas como *disponíveis* podem sensibilizar a transição.

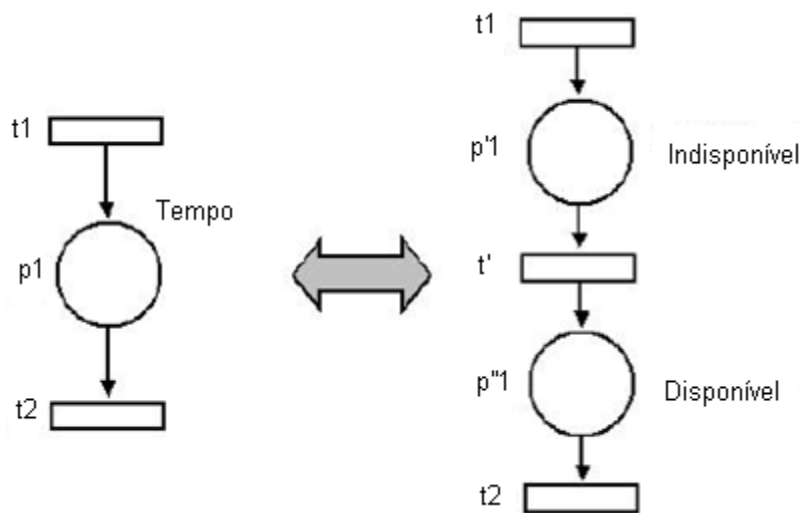


Figura 9 - Temporização da rede de Petri (P-Temporizada)  
(Fonte: CARDOSO, 1997)

#### 2.6.4 RdP Estocástica

Nas RdP T-Temporizadas e P-Temporizadas o tempo para se realizar uma determinada tarefa é constante. Porém existem sistemas em que a duração da tarefa pode variar. Nesses casos utiliza-se RdP Estocástica onde são associados tempos randômicos aos disparos de cada transição que são distribuídos seguindo uma lei exponencial [CREPALDI, 2006].

### 2.6.5 RdP Temporal

É associado a cada transição um intervalo de tempo ( $tempo_{min}$ ,  $tempo_{max}$ ). Diferente do que ocorre nas RdP T-Temporizadas, o disparo da transição é instantâneo mas, para se concretizar, a transição deve estar sensibilizada por um período de no mínimo igual a  $tempo_{min}$ , e no máximo igual a  $tempo_{max}$ .

### 2.6.6 RdP Colorida

Vários estudos foram desenvolvidos no sentido de encontrar um método para estruturar os *dados*. Os modelos encontrados ficaram conhecidos como redes de Petri de Alto Nível (RdPAN). Compõem as RdPAN as RdP Colorida, as RdP Predicado-Transição e as RdP A Objeto.

Os sistemas começaram a ficar mais complexos, e essa complexidade significa, às vezes, a composição de vários processos semelhantes. Nesse caso, quando se utiliza a RdP Ordinária é necessário modelar o comportamento geral do sistema sem precisar a identidade de cada processo, o que acarreta em falta de informação, ou então modelar, individualmente cada um dos processos, o que torna o modelo pouco prático de se trabalhar devido ao número de interações existentes e do tamanho da rede [CARDOSO, 1997].

Com as RdP Coloridas as fichas foram diferenciadas associando cores às mesmas. Como consequência, a cada lugar se associa ao conjunto de cores das fichas que podem pertencer a esse lugar. A cada transição se associa a um conjunto de cores que correspondem às diferentes maneiras de disparar uma transição. As etiquetas dos arcos não são mais números inteiros como nas RdP Ordinárias, mas funções que representam matrizes de inteiros. Para cada cor de uma transição é necessário descrever quais cores de fichas serão retiradas dos lugares de entrada (pré-condição) e quais cores de fichas serão colocadas nos lugares de saída (pós-condição) [CARDOSO, 1997].

A RdP Colorida pode ser convertida para a sua RdP Ordinária equivalente, convertendo cada lugar a um número de vagas equivalentes ao número possível de fichas coloridas do lugar. Cada transição é convertida para um número de transições correspondentes às diferentes formas de disparos. A Figura 10 mostra o equivalente da RdP Ordinária (a) para o modelo da RdP Colorida (b).

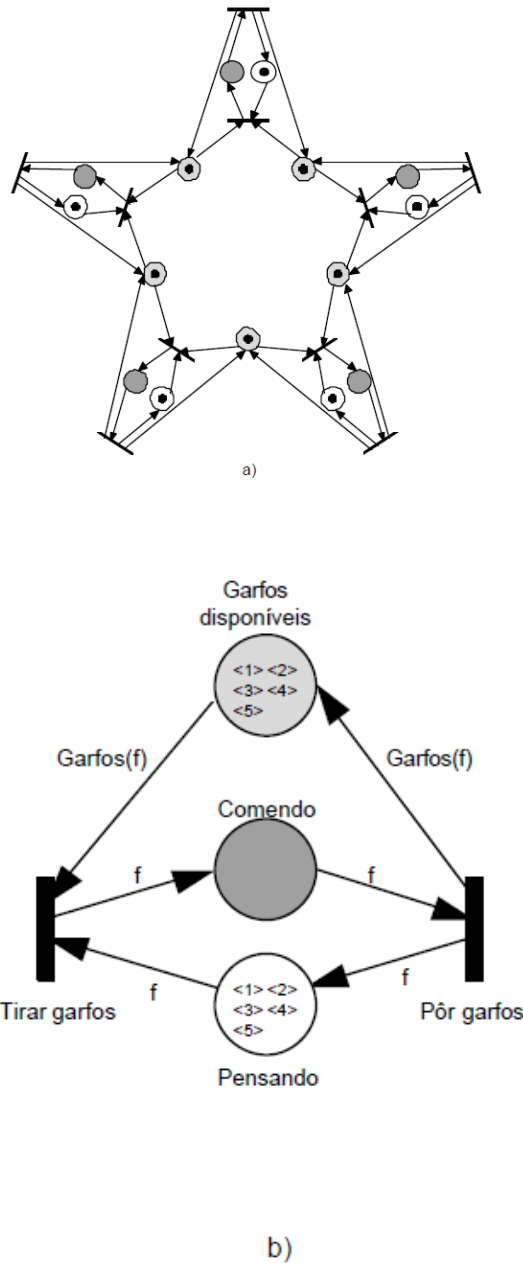


Figura 10 – O problema dos filósofos: a) utilizando uma RdP Ordinária  
 b) Utilizando RdP Colorida  
 (Fonte: CARDOSO, 1997)

### 2.6.7 RdP Predicado-Transição

Nas RdPs Predicado-Transição, as transições de uma RdPs Ordinária são consideradas como regras num sistema de lógica proposicional (sem variáveis), e o poder de descrição é aumentado substituindo-se por regras lógicas de primeira ordem (regras de variáveis). Com isto, a representação fica mais concisa. A transição passa a descrever uma família de eventos e não mais so-

mente um evento. A família é definida pelo conjunto de substituições possíveis de variáveis por valores [MELO, 2006]. Ao invés de regras do tipo:

- Se peça do tipo  $pc_1$  em espera e máquina  $M_2$  disponível, fazer usinagem  $u_{12}$ ;
- Se peça do tipo  $pc_2$  em espera e máquina  $M_2$  disponível, fazer usinagem  $u_{22}$ ;
- Se peça do tipo  $pc_i$  em espera e máquina  $M_j$  disponível, fazer usinagem  $u_{ij}$ ;

tem-se regras do tipo: Se peça do tipo  $\langle x \rangle$  em espera e máquina  $\langle y \rangle$  disponível, fazer uma usinagem  $\langle u \rangle$ .

As variáveis  $\langle x \rangle$ ,  $\langle y \rangle$  e  $\langle u \rangle$  assumirão, respectivamente, valores no conjunto de constante  $\{pc_i\}$  descrevendo as peças em espera, o conjunto  $M_j$  das máquinas livres, e o conjunto  $\{u_{ij}\}$  das opções a serem realizadas.

#### 2.6.8 RdP a Objeto

A RdPs a Objetos utiliza-se dos conceitos da RdPs Predicato-Transição, adicionado ao modelo conceitos de Classes de Objetos, atributos, métodos e herança. Conceitos abstraídos da teoria de Orientação a Objeto [CARDOSO, 1997].

Uma classe de objetos é definida por um conjunto de atributos, também chamados propriedades, e um conjunto de operações ou métodos que permitem manipular os valores dos atributos. Pode-se definir uma classe a partir de uma herança. A nova classe – chamada subclasse – herda as definições de atributos da classe anteriormente definida. Estas definições podem ser completadas por atributos e operações específicas. As classes são apenas definições. Um objeto particular é uma instância de classe de objeto. Pode-se atribuir valores aos atributos destes objetos e executar as operações [CARDOSO, 1997].

Considerando-se um sistema de manufatura, os atributos dos objetos podem ser, por exemplo, para a classe peça, o tipo de peça, material, operações já realizadas, e para a classe máquina, os tipos de operações que podem ser realizadas etc.

### **CAPÍTULO 3: SISTEMA SCADA**

Os sistemas supervisórios SCADA (Sistema de Controle Supervisório e Aquisição de dados ou *Supervisory Control and Data Acquisition*) podem ser vistos como sistemas que supervisionam ou monitoram processos executados em uma planta industrial, através da visualização de variáveis da planta que está sendo automatizada, bem como das ações tomadas pelo sistema de automação [SILVA, 2005]. São aplicados principalmente com a finalidade de tornar possível o reconhecimento de possíveis falhas em componentes da planta fabril antes que essas falhas aconteçam efetivamente, partindo-se da premissa que a eminente ocorrência de uma falha geralmente pode ser detectada através de mau funcionamento e pequenas anomalias no comportamento do sistema antes que a falha realmente aconteça, permitem ainda o acompanhamento destas falhas em forma de histórico [SILVA, 2005].

A utilização de sistemas SCADA permite uma série de vantagens se comparados com os painéis convencionais [VIANA, 2008].

- Redução de gastos com montagem de painéis de controle e projeto;
- Redução de custos da aquisição de instrumentos de painel, pois no sistema SCADA são virtuais;
- Eliminação de custos com peças de reposição, pois tratam-se de instrumentos virtuais;
- Redução de espaço necessário para a sala de controle;
- Dados disponíveis em formato eletrônico, facilitando a geração de relatórios;
- Praticidade da operação, pois os instrumentos são apresentados ao operador em um simples clique do dispositivo apontador.

Os principais sistemas SCADA de hoje oferecem três funções básicas [VIANA, 2008]:

- **Funções de supervisão:**

Inclui todas as funções de monitoramento do processo tais como: sinóticos animados, gráficos de tendência de variáveis analógicas e digitais, relatórios em vídeo e impressos, etc.



- **Funções de operação:**

Atualmente os sistemas SCADA substituíram com vantagens as funções da mesa de controle. As funções de operação incluem: ligar e desligar equipamentos e seqüência de equipamentos, operação de malhas PID, mudança de modo de operação de equipamentos, etc.

- **Funções de controle:**

Controle DDC ("Digital Direct Control")

Alguns sistemas de supervisão possuem uma linguagem que permite definir diretamente ações de controle, sem depender de um nível intermediário de controle representado por remotas inteligentes. Todas as operações de entrada e saída são executadas diretamente através de cartões de I/O ligados diretamente ao barramento do micro, ou por remotas mais simples. Os dados são amostrados, um algoritmo de controle como um controlador PID, por exemplo, é executado, e a saída é aplicada ao processo (ação direta sobre uma variável manipulada). Isto entretanto só é possível quando a velocidade do processo assim o permite. Em alguns casos requisitos de confiabilidade tornam desaconselhável este tipo de solução.

Controle Supervisório:

Nesta classe de sistemas, os algoritmos de controle são executados pela unidade terminal remota (RTU), mas os set-points para as malhas de controle são calculados dinamicamente pelo sistema de supervisão de acordo com o comportamento global do processo. Esta arquitetura possui maior confiabilidade que os sistemas DDC e traz a vantagem de atuar sobre um grande número de malhas de controle simultaneamente enquanto o operador geralmente só consegue atuar malha a malha com um sistema convencional.

### 3.1 **ARQUITETURA**

O sistema SCADA basicamente se consiste de um computador principal (servidor principal) e que também pode ser chamada de Unidade Terminal Mestre (UTM) e que se conectam a várias Unidades Terminais Remotas (UTR) ou Controladores Lógicos Programáveis (CLP).

As UTRs ou CLPs são responsáveis por adquirir os dados dos sensores em campo através de entradas analógicas e digitais e também por comandar equipamentos utilizando as saídas digitais comandando uma abertura ou fechamento de uma chave. Além disso, eles verificam os valores adquiridos em campo e envia esses dados para a UTM através de uma rede de dados, que pode ser fibra ótica, rádio ou qualquer outro meio que forneça a comunicação. A UTR também possui um papel de filtrar as medidas obtidas em campo de forma que a UTR só envia aqueles dados que tiveram alguma mudança exatamente para evitar um possível congestionamento na rede de dados. A UTM por sua vez tem a função que pegar todo o dado provindo das UTRs, analisá-los verificando valores limites e se houve algum evento expressivo como a mudança de algum estado ocasionado por algum sistema de proteção automático, e mostrar todas as informações sensíveis em forma de alarmes na tela do computador do operador de forma que ele possa tomar uma decisão caso seja necessário. A tela que o operador ver em seu micro computador é chamada de Interface Homem - Máquina (IHM) e nessa interface que ele poderá visualizar as medidas de campo, alarmes e também enviar comandos para que um dispositivo em campo seja acionado, mesmo que esse dispositivo esteja a quilômetros de distância é exatamente essa funcionalidade uma das grandes vantagens do SCADA.

Existem atualmente no mercado vários *softwares* supervisórios, dentre os quais Axeda Supervisor e Wizcon, da Axeda Systems Inc; iFIX, da Intellution; Elipse SCADA e Elipse E3, da Elipse; Indusoft, da Indusoft e InTouch, da Wonderware [SANTOS, 2007]. Nesse trabalho optou-se por utilizar o sistema Elipse E3 com a função de supervisão e a existência do CLP será representada por uma simulação no próprio software.

### **3.2 ELIPSE E3**

Elipse E3 é um software desenvolvido pela Elipse Software para desenvolvimento de sistemas de supervisão e controle de processos. Totalmente configurável pelo usuário, permite a monitoração de variáveis em tempo real, através de gráficos e objetos que estão relacionados com as variáveis físicas de campo. Também é possível fazer acionamentos e enviar ou receber informações para equipamentos de aquisição de dados.

Os sistemas construídos com o E3 geralmente partem da coleta de dados em tempo real de equipamentos de aquisição de dados ou de controle, como os CLPs (Controladores Lógicos Programáveis) e UTRs (Unidades Terminais Remotas). Estes equipamentos normalmente possuem alguma interface que permitem sua conexão ao software, como padrões seriais RS232, RS422 ou RS485; rádio ou modem (linhas privadas ou discadas), TCP/IP ou UDP/IP, placas ligadas diretamente no barramento do computador e outros. O E3 lê e escreve dados dos equipamentos através de módulos (drivers de comunicação) que implementam o protocolo (de Domínio público ou privado) disponível em cada um desses equipamentos. Esses drivers podem ainda estar em um formato próprio da Elipse Software ou no formato OPC (OLE for Process Control).

De posse dos dados, é possível criar várias maneiras para exibir, analisar, controlar, comandar, gravar ou divulgar tais informações, dentre elas:

**Telas:** permitem criar uma IHM (Interface Homem-Máquina) localmente, pela rede local ou Internet, com o intuito de exibir o status atual ou passado dos dados em várias formas, com o auxílio de um editor gráfico e objetos específicos

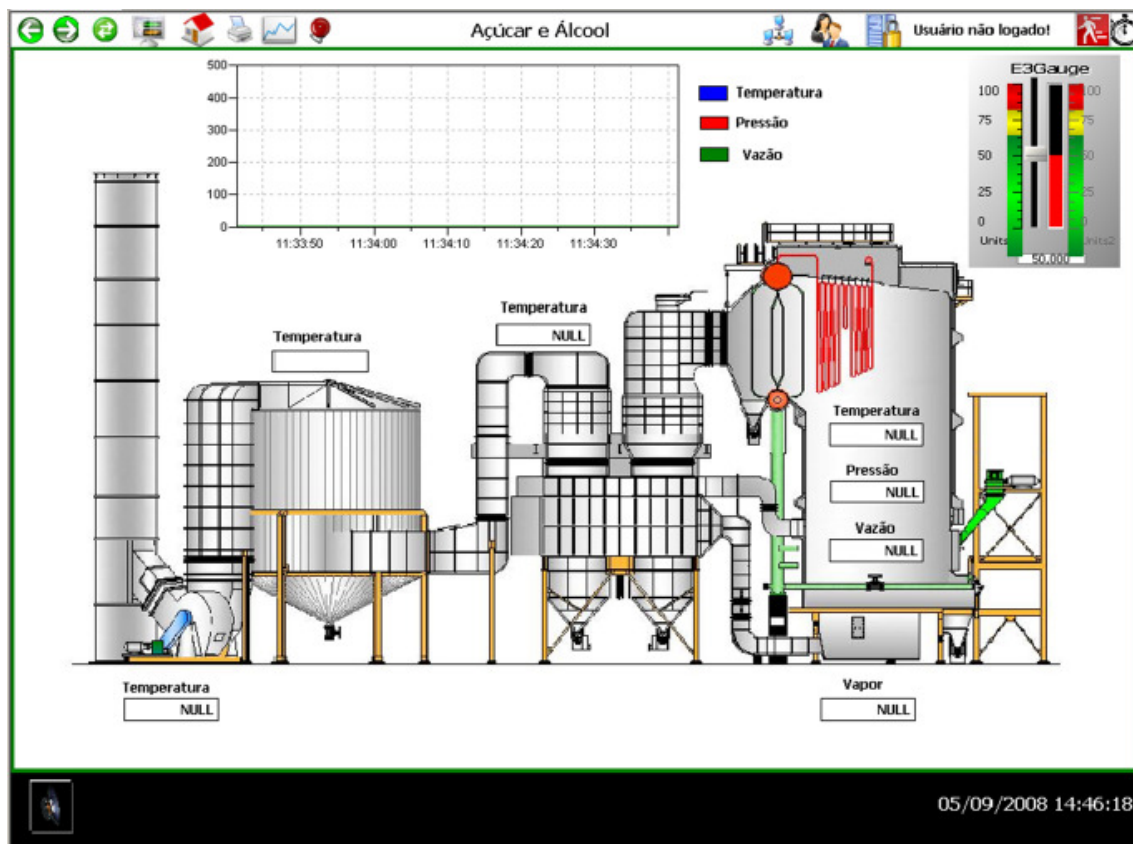


Figura 11 – Área de Telas  
(Fonte: ELIPSE SCADA, 2008)

- **Alarmes e Eventos:** monitoram o acontecimento de situações específicas.
- **Históricos:** armazenam os dados em bancos de dados relacionais.
- **Relatórios:** permitem visualizar e imprimir os dados, dentre outros módulos e possibilidades.

O Elipse E3 também oferece uma poderosa ferramenta de scripts orientados a Eventos e a Objetos, permitindo operações matemáticas, lógicas, manipulação de estruturas, além de fontes modificáveis e cores diferentes para palavras chaves.

A simulação da estação de montagem da planta CIM foi toda construída utilizando-se das ferramentas de script e de associações existentes no Elipse E3.

## CAPÍTULO 4: DESENVOLVIMENTO

### 4.1 PLANTA CIM

A planta CIM do Senai-Cimatec Bahia dispõe de uma variedade de equipamentos industriais. Entre os equipamentos e sistemas instalados, estão: robôs manipuladores industriais de sistema de transporte, sistema de inspeção por visão computadorizada e por contato, estação de tratamento superficial de peça, centros de usinagem que dispõe de torno e fresa industriais e didáticos, estação de transporte por veículo motorizado, estação de armazenamento e recuperação automática de peças (AS/RS) e estação de montagem com dispositivos de prensa. A planta tem como produto final um “Porta Objeto”, composto por quatro peças (2 cilindros e 2 chapas), sendo montado na estação de montagem, estação esta objeto de estudo para a implementação desse trabalho. A Figura 12 e a Figura 13 a seguir ilustra a planta CIM e o seu produto final, respectivamente.

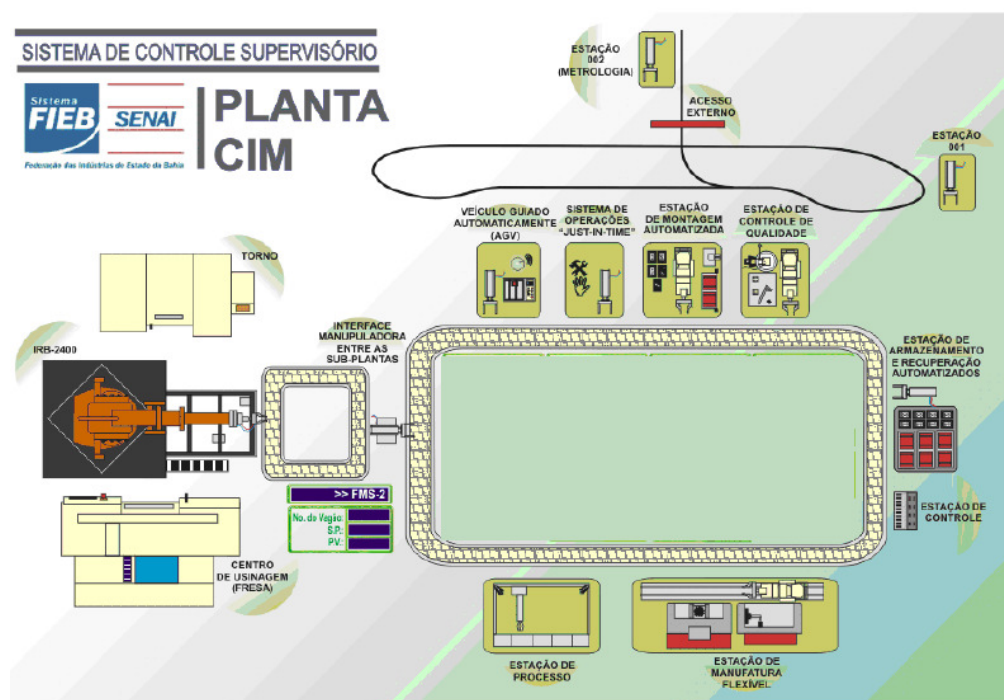


Figura 12 – Arquitetura Planta CIM

(Fonte: RIBEIRO, 2009)

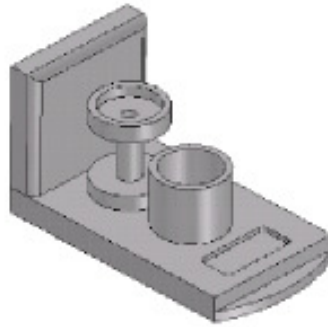


Figura 13 – Porta Objeto  
(Fonte: RIBEIRO, 2009)

## 4.2 ESTAÇÃO DE MONTAGEM

A estação de montagem (Figura 14) é composta por:

- 1 Atuador Pneumático.
- 1 Armazém com 4 Buffer.
- 1 Robô Mitsubishi RV-4A.



Figura 14 – Estação de Montagem  
(Fonte: RIBEIRO, 2009)

Após a usinagem, as 4 peças que compõem o Porta Objeto são encaminhadas em vagões, através da esteira transportadora, para a estação de Montagem. Há 32 vagões dispostos na esteira onde cada vagão contém um palete capaz de armazenar qualquer tipo de peça.

A identificação e o recebimento da peça a ser processada na estação de montagem são feitos através de 6 sensores indutivos sendo: 1 para enviar ao

CLP principal informação que há um vagão em frente à estação e 5 que identificam, através de um código binário (5 posições na parte inferior do vagão que podem ou não ter placas metálicas), qual dos vagões está na frente da estação. A estação envia essas informações para o CLP principal que estabelece comunicação com a mesma através de entradas e saídas digitais. O CLP principal é responsável pelo todo o controle da planta. É através dele que se faz o reconhecimento do vagão fazendo atuar o atuador pneumático que segura ou não o vagão na frente da estação de montagem. O robô então retira do vagão palete e peça e os colocam em um dos buffers do armazém que possui a capacidade para 4 buffers. Após o recebimento das 4 peças inicia-se o processo de montagem do produto final, utilizando a prensa e a cola como ferramentas de suporte. Após a montagem, o produto é enviado pelo robô ao armazém e, na presença de um vagão vazio na esteira transportadora, o conjunto peça e palete são enviados para o mesmo. Os demais paletes vazios presentes no armazém são devolvidos à planta da mesma maneira que o produto final e inicia-se todo o processo da estação de montagem.

O capítulo a seguir contém a descrição da modelagem da estação de montagem utilizando rede de Petri Interpretada. A prensa e a cola não serão apresentadas na modelagem por serem acionadas pelo robô e não pelo CLP.

### **4.3 MODELAGEM DA ESTAÇÃO DE MONTAGEM**

Para um melhor entendimento, a modelagem será apresentada em 3 etapas conforme descritas abaixo:

4.3.1. Etapa de reconhecimento e recebimento das peças.

4.3.2. Etapa de encaminhamento das peças da esteira transportadora para o armazém e o processo de montagem.

4.3.3. Etapa de devolução do produto final e paletes para a planta.

A Tabela 1 contém todas as entradas e saídas digitais, da rede de Petri Interpretada, utilizadas na modelagem da estação. As entradas e as saídas digitais da rede de Petri correspondem respectivamente às saídas e as entradas digitais do CLP principal.

ENTRADA DA REDE	SAÍDA DA REDE
PRG1	Robo_exec_PRG1
PRG2	Robo_exec_PRG2
PRG3	Robo_exec_PRG3
PRG4	Robo_exec_PRG4
PRG5	Robo_exec_PRG5
PRG6	Robo_exec_PRG6
PRG7	Robo_exec_PRG7
PRG8	Robo_exec_PRG8
PRG9	Robo_exec_PRG9
Ativa_Atuidor	Presença_Vagão
	Numero_Vagão
	Buffer_1_Ocupado
	Buffer_2_Ocupado
	Buffer_3_Ocupado
	Buffer_4_Ocupado

Tabela 1 – Entradas e saídas digitais da rede de Petri Interpretada

onde,

**Ativa\_Atuidor:** CLP principal ordena que a estação ative o atuador pneumático para segurar peça.

**PRG“X”:** Robô retira peça “X” e paleta do vagão presente na esteira transportadora e os colocam no buffer “X” do armazém, sendo X = 1, 2, 3 e 4.

**PRG5:** Robô executa programa de montagem.

**PRG6:** Robô devolve peça pronta e 1° paleta para vagão vazio na esteira transportadora.

**PRG7, PRG8 e PRG9:** Robô devolve o 2°, 3° e 4° paleta, respectivamente, para o vagão vazio na esteira transportadora.

**Presença\_Vagão:** Estação de montagem informa ao CLP principal que há um vagão em frente a estação.

**Numero\_Vagão:** Estação de montagem informa ao CLP principal o número do vagão que se encontra em frente a estação.

**Robo\_exec\_PRG“Y”:** CLP principal ordena à estação que o robô execute os programas PRG“Y”, sendo Y = 1, 2, 3, 4, 5, 6, 7, 8 e 9.

**Buffer\_“Z”\_Ocupado:** Estação de montagem informa ao CLP principal que o buffer “Z” do armazém encontra-se ocupado, sendo Z = 1, 2, 3 e 4.



### 4.3.1 Etapa de reconhecimento e recebimento das peças

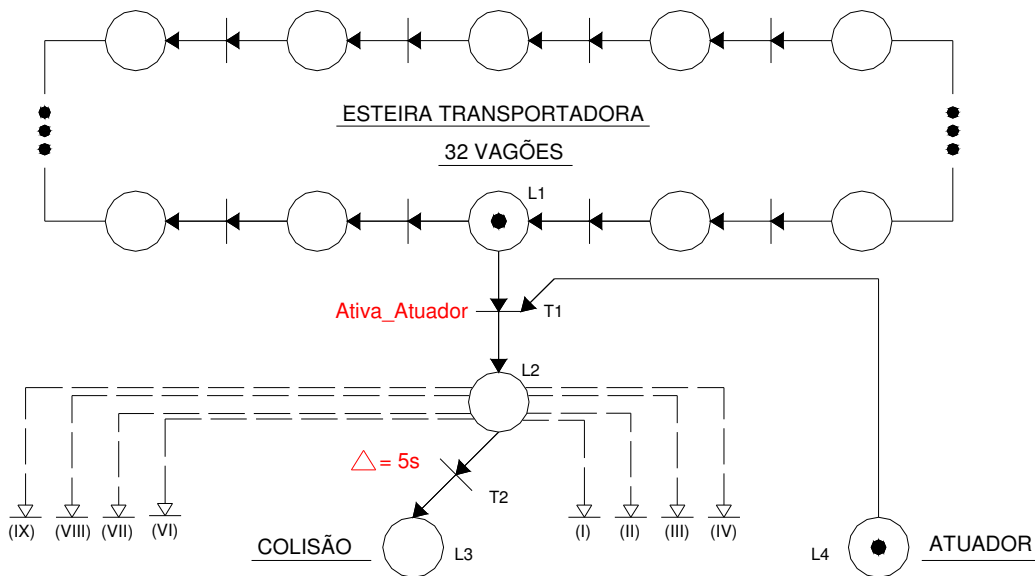


Figura 15 – Etapa de reconhecimento e recebimento das peças

#### Descrição:

Para que a transição T1 seja disparada é necessário que a mesma seja sensibilizada (lugares L1 e L4 marcados) e que o ponto digital *Ativa\_Atuidor* seja levado ao nível lógico 1 (característica de uma rede de Petri Interpretada). A mudança de nível é determinada pelo CLP principal que fará o reconhecimento da peça e informará à estação o tipo de programa a ser executado. O tipo de programa está sendo representado pelos algarismos romanos da figura acima.

Caso nenhum programa seja iniciado durante 5s ( $\Delta=5s$ ) significa que houve uma colisão. Essa colisão pode ser devido, por exemplo, a um envio de uma tarefa do CLP principal para o robô sendo que este não está disponível para executar a mesma devido a um problema mecânico.

### 4.3.2 Etapa de encaminhamento das peças da esteira transportadora para o Armazém e o processo de montagem

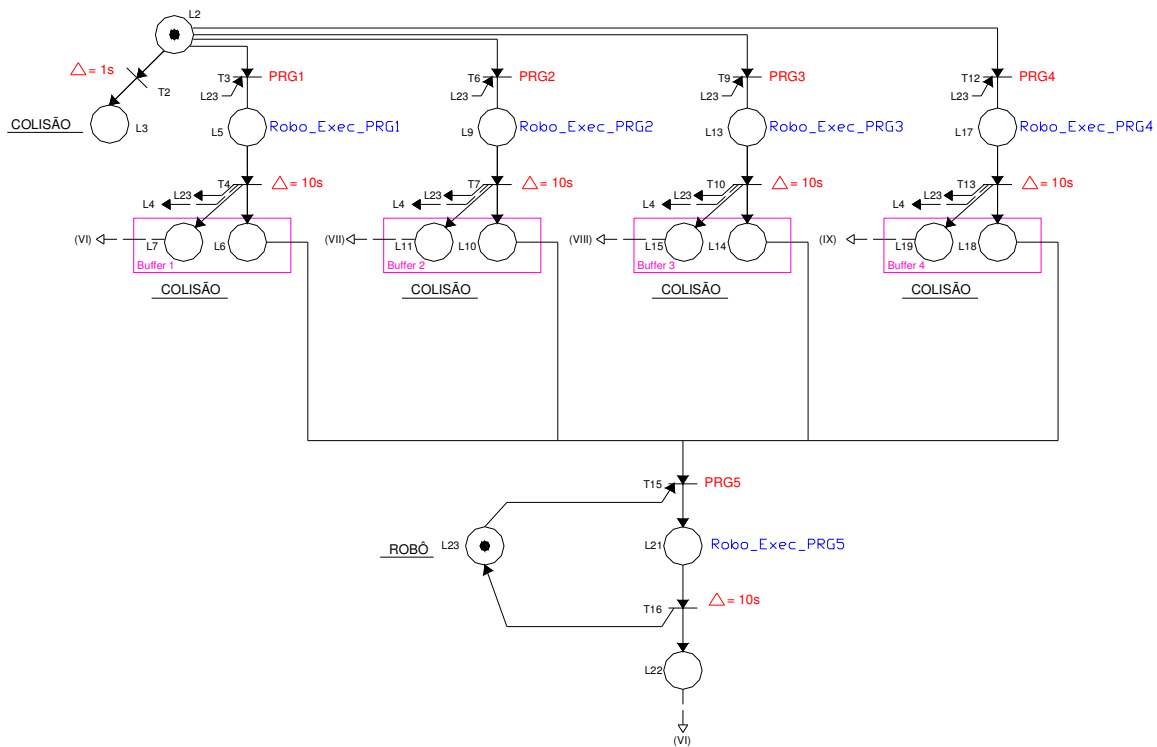


Figura 16 – Etapa de encaminhamento das peças da esteira transportadora para o armazém e o processo de montagem

#### Descrição:

Após reconhecimento e recebimento da peça o CLP informa à estação o programa que deve ser executado pela mesma. Como a modelagem é igual para os programas (PRG1, PRG2, PRG3 e PRG4) será considerado o PRG1 para explicação detalhada.

Seja a primeira peça a chegar à estação uma peça cilíndrica. A transição T3 será disparada caso esteja sensibilizada (lugares L2 e L23 marcados) e o ponto digital PRG1 seja levado ao nível lógico 1. Após disparo de T3, o robô informará ao CLP que está executando o programa (Lugar L5 marcado). Foi arbitrado um tempo de 10s para execução dos programas PRG1, PRG2, PRG3 e PRG4. A transição T4 será disparada se o lugar L5 estiver marcado e o tempo de 10s tenha sido transcrito. Os lugares L4 e L23 correspondem respectivamente ao atuador e ao robô. Ambos estarão livres para executar a próxima tarefa após disparo da transição T4. Os lugares L6 e L7 correspondem ao buffer 1 do

armazém, sendo L6 representando a peça e o L7 o palete. Cada buffer só pode receber uma peça por vez. A estação informa ao CLP principal, através da saída digital Buffer\_1\_Ocupado, que o buffer 1 encontra-se ocupado, para evitar que o CLP principal solicite a execução do mesmo tipo de programa. Caso o CLP principal solicite, mesmo assim, a execução do mesmo programa na chegada de uma nova peça à estação de montagem, o robô e o atuador serão ativados, caso estejam disponíveis, e haverá uma colisão. A transição T15 será disparada após as 4 peças chegarem ao armazém e após o ponto lógico PRG5 ser levado para 1. Arbitrou-se 10s o tempo de montagem do produto.

### 4.3.3 Etapa de devolução do produto final e paletes para a planta

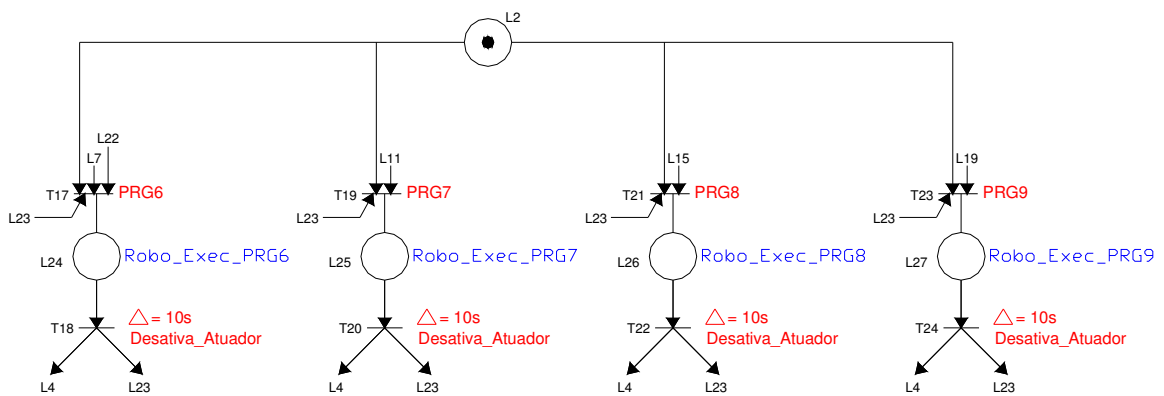


Figura 17 – Etapa de devolução do produto final e paletes para a planta

#### Descrição:

Após a montagem o robô devolve a planta os paletes vazios e o produto final. A transição T17 será disparada se o robô estiver disponível, o lugar L2 estiver marcado, representando um vagão vazio, além dos lugares 7 e 22. Arbitrou-se 10s a execução da tarefa. Concluindo a tarefa, o robô e o atuador estão livres para executar outra atividade. As demais transições da Figura 17 se comportam de maneira similar à transição T17 exceto que não precisarão ter o lugar 22 marcado para ser sensibilizadas.

## **CAPÍTULO 5: RESULTADOS**

A modelagem da estação de montagem foi toda desenvolvida no Eclipse E3 utilizando-se de uma poderosa ferramenta de scripts, orientados a eventos e a objetos, e de associações. Existem inúmeras associações possíveis que podem ser feitas no Eclipse E3. Utilizou-se apenas de três delas na modelagem da estação de montagem: *Layer*, *Visible* e *Value*.

- ***Layer***

Esta propriedade define em quais camadas o objeto deve aparecer. O valor representa uma máscara de 32 bits, um bit para cada camada. Portanto, podem ser definidas até 32 camadas individuais. Assim, objetos podem ser agrupados logicamente e mostrados ou escondidos apenas com a modificação da máscara da propriedade *Layer*.

- ***Visible***

Esta propriedade define a visibilidade do objeto. Se configurada como *True*, o objeto fica visível, desde que também sejam obedecidos os seguintes fatores: o objeto-pai deste objeto também deve estar visível, e a propriedade *Layer* deste objeto também deve estar presente na camada da Tela.

- ***Value***

Indica o valor inicial do objeto. Tem comportamento booleano; se *True*, o objeto inicia marcado, do contrário, seu estado inicial é não marcado. O valor padrão desta propriedade é *False*. É normalmente utilizado na presença de um CheckBox.

Nesse capítulo serão apresentadas as telas e a explicação de algumas associações e scripts utilizados na modelagem apresentada no capítulo 4.

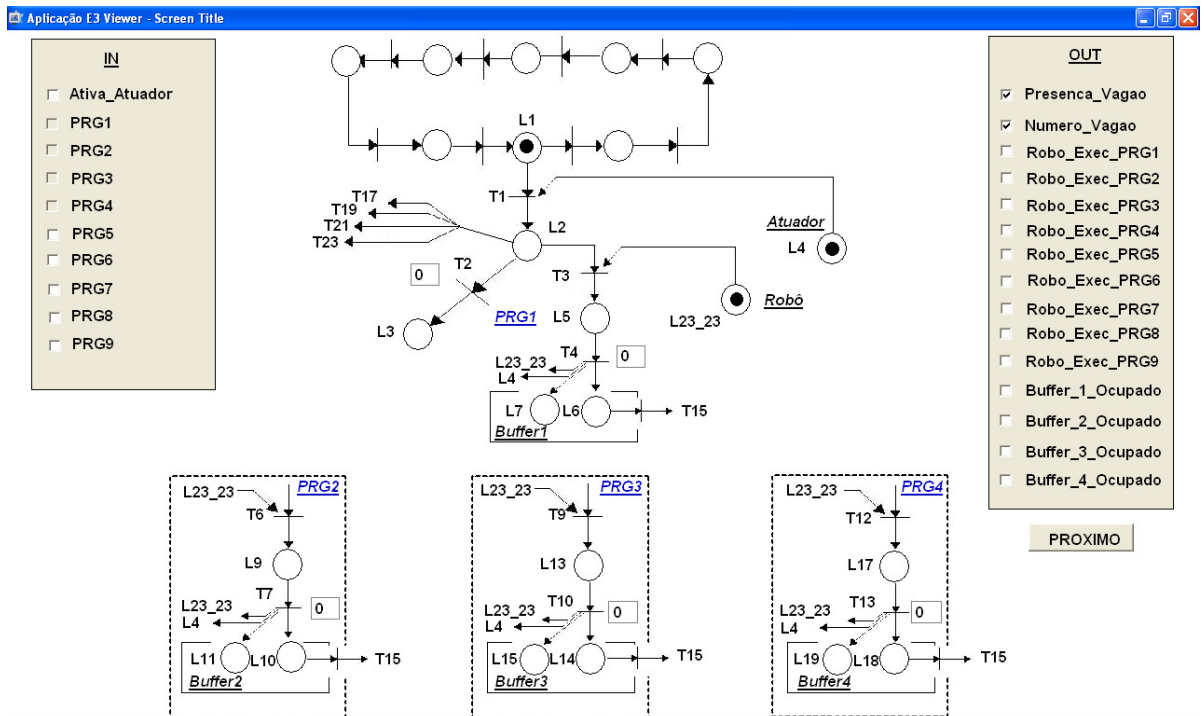


Figura 18 – Etapa de recebimento das peças e encaminhamento para o armazém – Elipse E3

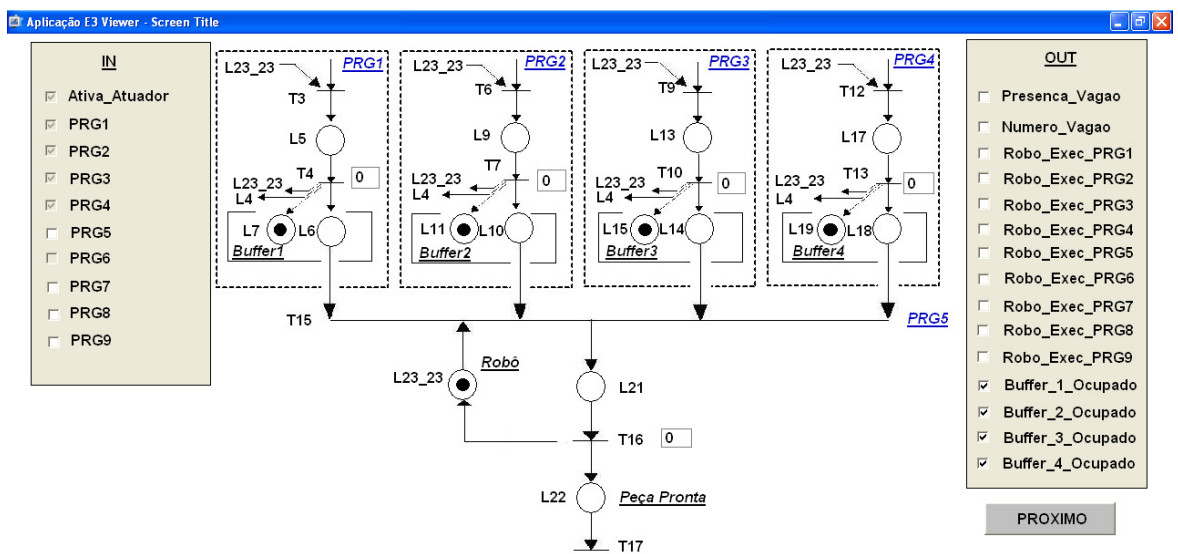


Figura 19 – Etapa da montagem do Porta Objetos – Elipse E3

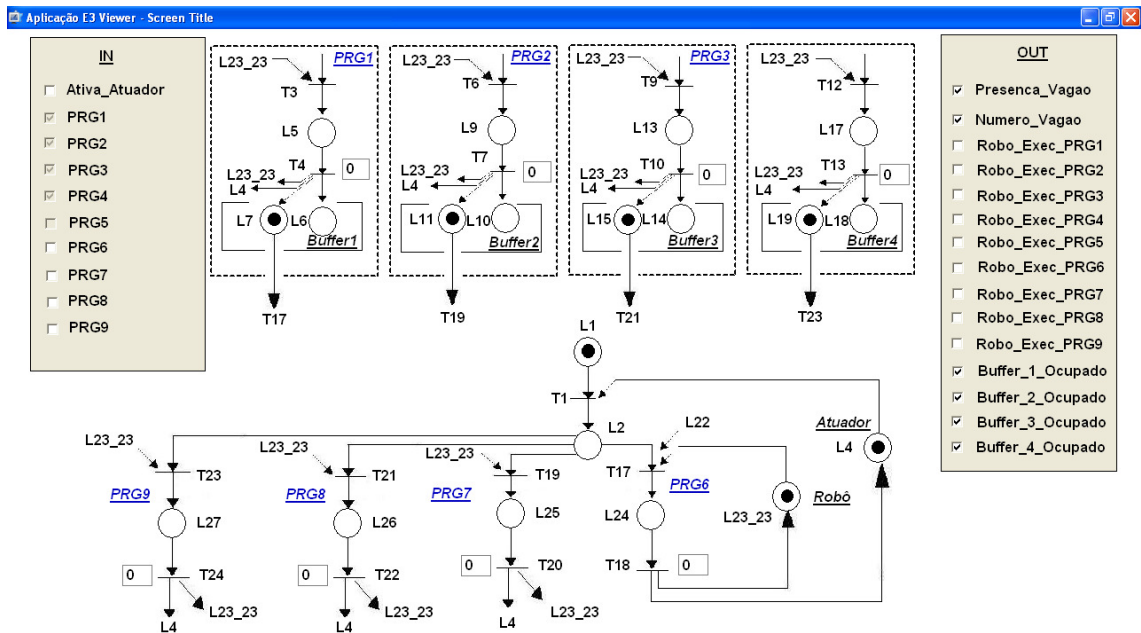


Figura 20 – Etapa de devolução do produto final e paletes para a planta – Elipse E3

## 5.1 Script relacionado à etapa de recebimento das peças e encaminhamento para o armazém

- **CheckBox\_Presenca\_Vagão e CheckBox\_Numero\_Vagão**

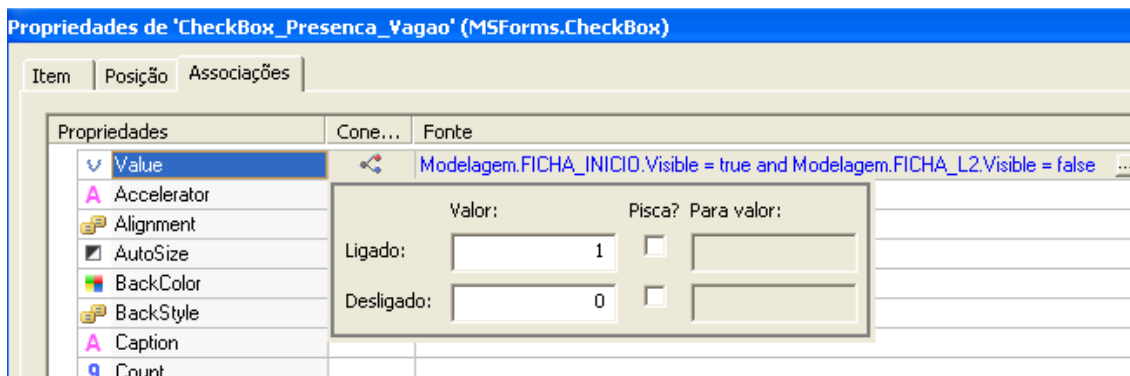


Figura 21 – Propriedades do CheckBox\_Presenca\_Vagão – Elipse E3

O CheckBox\_Presenca\_Vagão será verdadeiro se a FICHA\_INICIO (ficha que representa vagão em frente a estação) estiver visível e a FICHA\_L2 não estiver visível. O mesmo acontece com o CheckBox\_Numero\_Vagão.

- **Lugar L1 e L4**

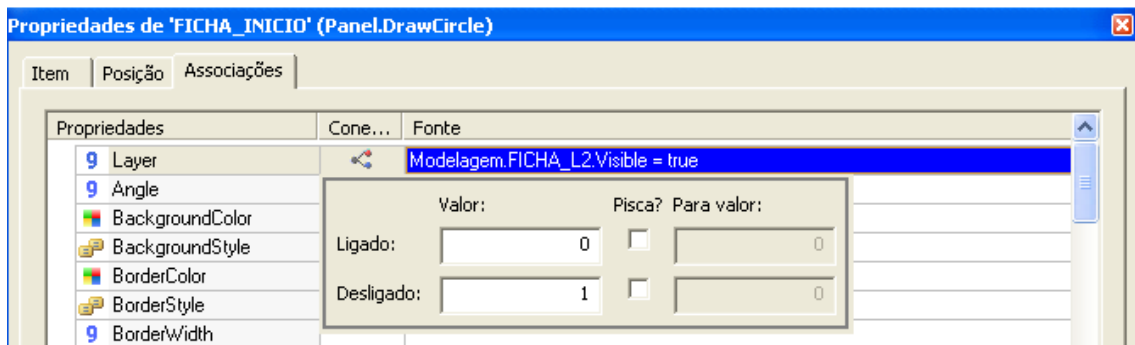


Figura 22 – Propriedades da FICHA\_INICIO – Elipse E3

Utilizou-se da propriedade *Layer* do Elipse E3. Ou seja, se a FICHA\_L2 (ficha que representa o atuador ativado) estiver visível, o layer da FICHA\_INICIO (lugar L1) será desligado fazendo a ficha deste ficar invisível. O mesmo acontece com a FICHA\_ATUADOR (lugar L4).

- **Lugar L2**

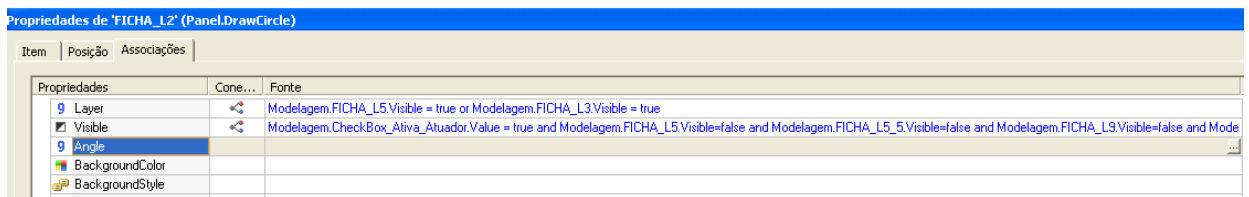


Figura 23 – Propriedade da FICHA\_L2 – Elipse E3

Utilizou-se das propriedades *Layer*, *Visible* e *Value* do Elipse E3.

O layer da FICHA\_L2 estará desligado se a ficha do lugar L5 (ficha que representa robô executando o programa PRG1) ou do lugar L3 (ficha que representa robô com problema) estiver visível. A FICHA\_L2 estará visível se o CheckBox\_Ativa\_Atuador for verdadeiro e se as fichas dos lugares L5, L9, L13 e L17 (fichas que representa o robô executando o programa PRG1, PRG2, PRG3 ou PRG4, respectivamente) não estiverem visíveis.

Toda vez que a FICHA\_L2 estiver visível o contador Dados1.Robo\_Quebrado é iniciado. Essa implementação foi feita através de um *script*.

```
Sub [FICHA_L2.Visible]()
Application.GetObject("Dados1.Robo_Quebrado").Enabled = true _
End Sub
```

- **Lugar L3**

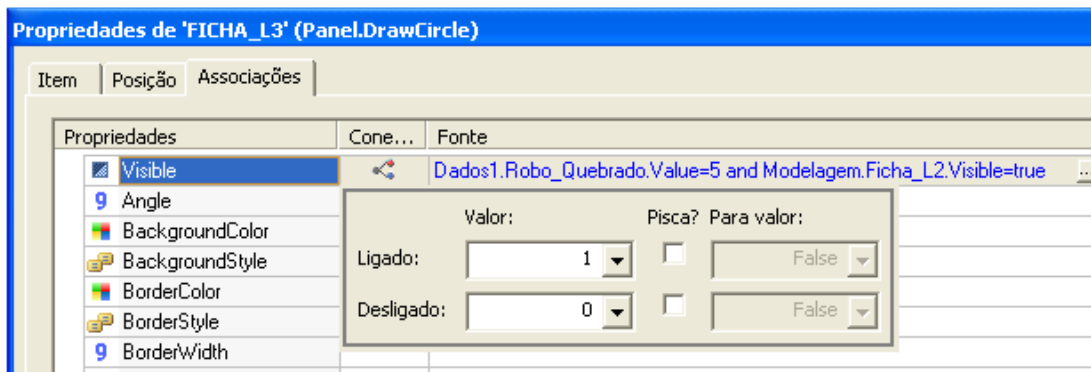


Figura 24 – Propriedade da FICHA\_L3 – Elipse E3

Utilizou-se da propriedade *Visible* do Elipse E3.

A FICHA\_L3 (ficha que representa robô com problema) estará visível se a FICHA\_L2 estiver visível e se o contador Dados1.Robo\_Quebrado for igual a 5.

- **Lugar L23\_23**

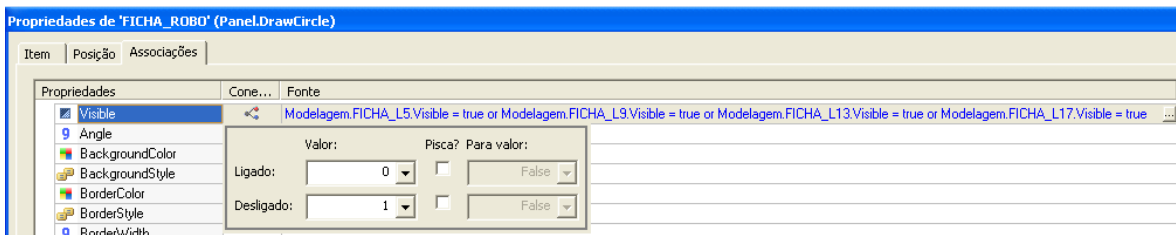


Figura 25 – Propriedade da FICHA\_ROBO – Elipse E3

A FICHA\_ROBO estará visível (indicando robô livre) desde que as fichas dos lugares L5, L9, L13 e L17 (fichas que representa o robô executando o programa PRG1, PRG2, PRG3 ou PRG4, respectivamente) não estejam visíveis.

- **Lugar L5**

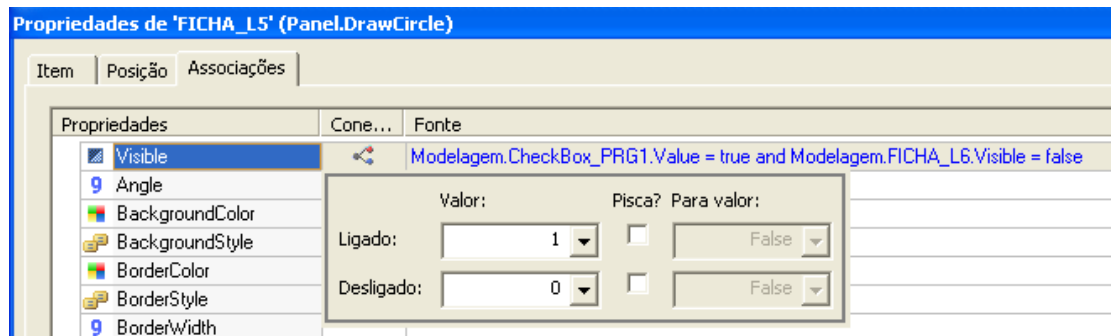


Figura 26 – Propriedade da FICHA\_L5 – Elipse E3



A FICHA\_L5 representa que o robô executando o programa 1 (PRG1). O robô só executará o PRG1 se os lugares L2 e L23\_23 (lugar que representa o robô livre ou ocupado) estiverem marcados e se o CheckBox\_PRG1 for verdadeiro.

Toda vez que a FICHA\_L5 estiver visível o contador Dados1.Exec\_PRG1 é iniciado indicando a duração da execução do programa. Esta implementação foi feita através de um *script*.

```
Sub [FICHA_L5 Visible]()
```

```
Application.GetObject("Dados1.Exec_PRG1").Enabled = true _
```

```
End Sub
```

Além disso, ao executar qualquer programa o robô informa a estação de controle tal execução.

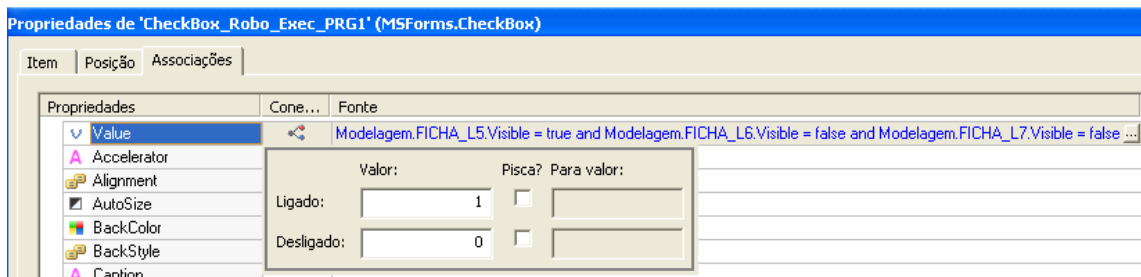


Figura 27 – Propriedade do CheckBox\_Robo\_Exec\_PRG1 – Elipse E3

O CheckBox\_Robo\_Exec\_PRG1 será verdadeiro desde que a FICHA\_L5 esteja visível e as fichas dos lugares L6 e L7 não estejam visíveis.

- **Lugares L6 e L7**

A FICHA\_L6 (ficha que representa que a peça foi colocada pelo robô no buffer 1 do armazém) estará visível quando o contador Dados1.Exec\_PRG1 for igual a 10, tempo estimado para a execução do programa PRG1.

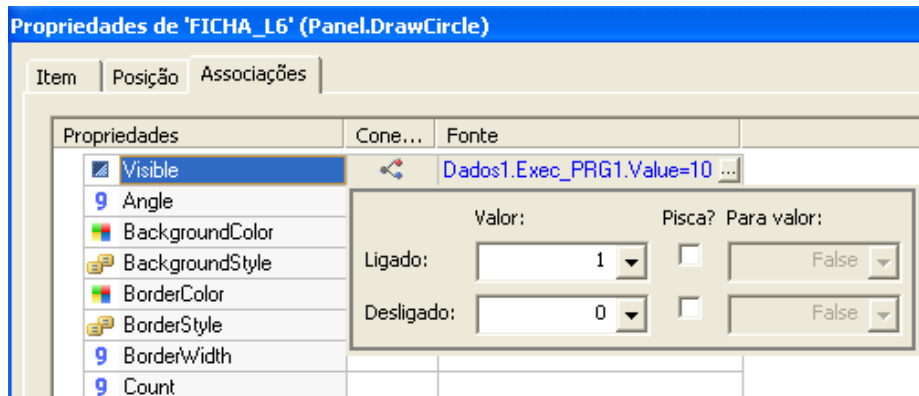


Figura 28 – Propriedade da FICHA\_L6 – Elipse E3

A FICHA\_L7 (ficha que representa o palete no buffer 1 do armazém) estará visível quando a FICHA\_L6 estiver visível.

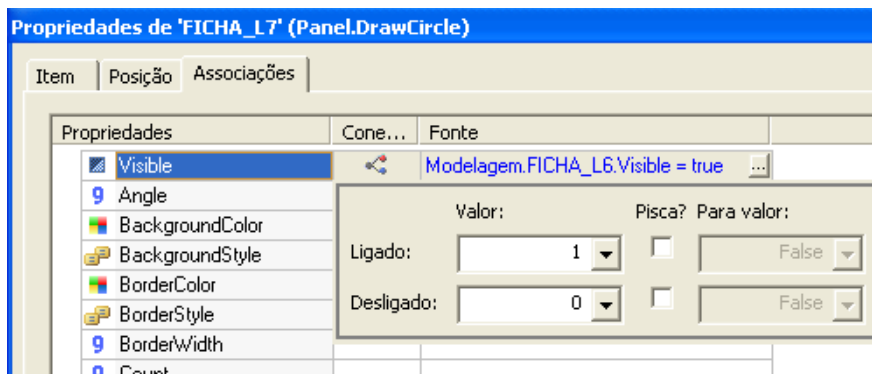


Figura 29 – Propriedade da FICHA\_L7 – Elipse E3

- **CheckBox\_Buffer\_1\_Ocupado**

Estando as fichas L6 e L7 visíveis (fichas que representam peça e palete no buffer 1 do armazém) a estação informa à estação de controle que o buffer 1 está ocupado. O CheckBox\_Buffer\_1\_Ocupado será verdadeiro quando o lugar L7 estiver marcado (FICHA\_L7 visível).

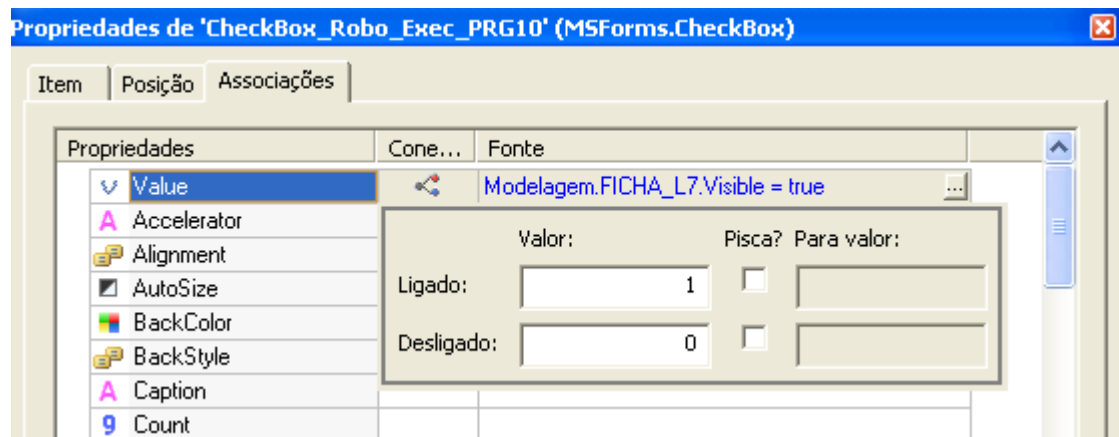


Figura 30 – Propriedade do CheckBox\_Robo\_Exec\_PRG10 – Elipse E3

Para as etapas de montagem do Porta Objeto e da devolução do mesmo e dos paletes para a planta CIM foram feitas associações e scripts equivalentes aos aqui apresentados.

## **CAPÍTULO 6: CONCLUSÕES**

Esse trabalho pôde comprovar, através de exaustivos testes, que a abordagem baseada na rede de Petri Interpretada se apresenta como uma solução que pode auxiliar o desenvolvimento do sistema de supervisão de um sistema flexível de manufatura assim como a análise da sua integração com elementos externos e com o sistema de controle.

Além disso, é possível utilizar-se de um software de supervisão, como o Eclipse E3, para simular o funcionamento de um sistema sem colocar em risco o sistema físico real. Com o Eclipse E3 foi possível representar os objetos e demonstrar, através de execução de scripts e associações, as interligações existentes entre as fichas da rede e entre estas e o ambiente externo, permitindo assim a implementação do simulador.

Enfim, o simulador criado para estação de Montagem permitiu compreender a importância de se ter uma boa integração entre as estações que compõem um sistema de manufatura além de permitir realizar testes em busca de melhorar o desempenho do sistema.

## CAPÍTULO 7: ANEXO

Nesse capítulo serão apresentadas todas as implementações feitas no Elipse E3 na construção do simulador.

- **Ativa\_Atuator**

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_ATUADOR.Visible=true or Modelagem.FICHA_ATUADOR.Layer=true
Enabled		Modelagem.FICHA_ATUADOR.Visible = true and Modelagem.FICHA_INICIO.Visible = true and Modelagem.FICHA_L2.Visible = false
Accelerator		

- **PRG1**

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L5.Visible=false
Enabled		Modelagem.FICHA_L2.Visible =true and Modelagem.FICHA_ROBO.Visible = true and Modelagem.FICHA_L5.Visible = false and Modelagem.FICHA_L3.Visible=false
Accelerator		

- **PRG2**

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L9.Visible=false
Enabled		Modelagem.FICHA_L2.Visible =true and Modelagem.FICHA_ROBO.Visible = true and Modelagem.FICHA_L9.Visible = false and Modelagem.FICHA_L3.Visible=false
Accelerator		


- **PRG3**

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L13.Visible=false
Enabled		Modelagem.FICHA_L2.Visible =true and Modelagem.FICHA_ROBO.Visible = true and Modelagem.FICHA_L13.Visible = false and Modelagem.FICHA_L3.Visible=false
Accelerator		


- **PRG4**

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L17.Visible=false
Enabled		Modelagem.FICHA_L2.Visible =true and Modelagem.FICHA_ROBO.Visible = true and Modelagem.FICHA_L17.Visible = false and Modelagem.FICHA_L3.Visible=false
Accelerator		


- **PRG5**

Propriedades de 'CheckBox_PRG5' (MSForms.CheckBox)		
Item	Posição	Associações
Propriedades		
Value		
Enabled		<a href="#">PRG_2.FICHA_L21.Visible=false and Dados1.Exec_PRG5.Value=3</a>


- **PRG6**

Propriedades de 'CheckBox_PRG6' (MSForms.CheckBox)		
Item	Posição	Associações
Propriedades		
Value		
Enabled		<a href="#">PRG_3.FICHA_L24.Visible=false and Dados1.Exec_PRG6.Value=10</a>


- **PRG7**

Propriedades de 'CheckBox_PRG7' (MSForms.CheckBox)		
Item	Posição	Associações
Propriedades		
Value		
Enabled		<a href="#">PRG_3.FICHA_L25.Visible=false and Dados1.Exec_PRG7.Value=10</a>

- **PRG8**

Propriedades de 'CheckBox_PRG8' (MSForms.CheckBox)		
Item	Posição	Associações
Propriedades		
Value		
Enabled		<a href="#">PRG_3.FICHA_L26.Visible=false and Dados1.Exec_PRG8.Value=10</a>

- **PRG9**

Propriedades de 'CheckBox_PRG9' (MSForms.CheckBox)		
Item	Posição	Associações
Propriedades		
Value		
Enabled		<a href="#">PRG_3.FICHA_L27.Visible=false and Dados1.Exec_PRG9.Value=10</a>

- **Presença\_Vagao**

Propriedades de 'CheckBox\_Presenca\_Vagao' (MSForms.CheckBox)

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_INICIO.Visible = true and Modelagem.FICHA_L2.Visible = false
Accelerator		

- **Numero\_Vagao**

Propriedades de 'CheckBox\_Numero\_Vagao' (MSForms.CheckBox)

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_INICIO.Visible = true and Modelagem.FICHA_L2.Visible = false
Accelerator		

- **Robo\_Exec\_PRG1**

Propriedades de 'CheckBox\_Robo\_Exec\_PRG1' (MSForms.CheckBox)

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L5.Visible = true and Modelagem.FICHA_L6.Visible = false and Modelagem.FICHA_L7.Visible = false
Accelerator		

- **Robo\_Exec\_PRG2**

Propriedades de 'CheckBox\_Robo\_Exec\_PRG2' (MSForms.CheckBox)

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L9.Visible = true and Modelagem.FICHA_L10.Visible = false and Modelagem.FICHA_L11.Visible = false
Accelerator		

- **Robo\_Exec\_PRG3**

Propriedades de 'CheckBox\_Robo\_Exec\_PRG3' (MSForms.CheckBox)

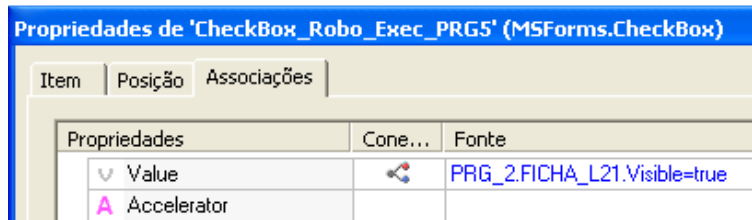
Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L13.Visible = true and Modelagem.FICHA_L14.Visible = false and Modelagem.FICHA_L15.Visible = false
Accelerator		

- **Robo\_Exec\_PRG4**

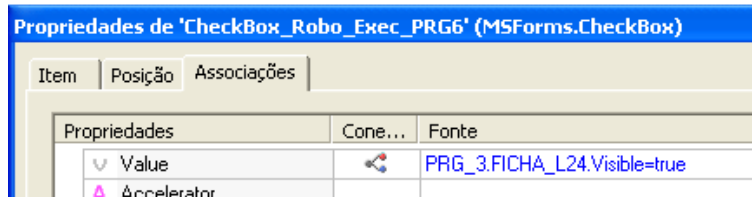
Propriedades de 'CheckBox\_Robo\_Exec\_PRG4' (MSForms.CheckBox)

Item	Posição	Associações
Propriedades		
Value		Modelagem.FICHA_L17.Visible = true and Modelagem.FICHA_L18.Visible = false and Modelagem.FICHA_L19.Visible = false
Accelerator		
Alignment		

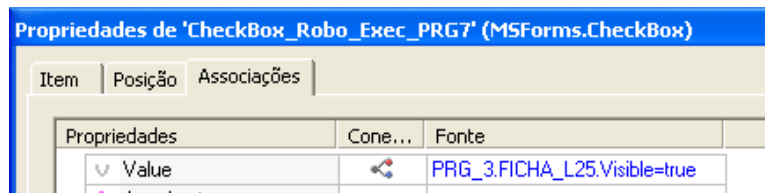
- **Robo\_Exec\_PRG5**



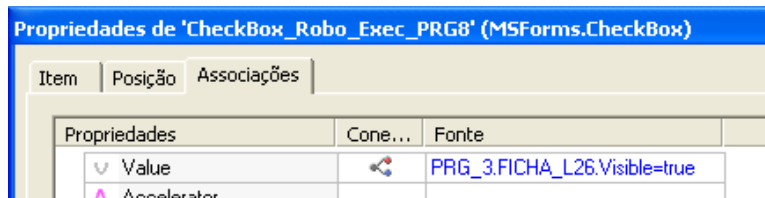
- **Robo\_Exec\_PRG6**



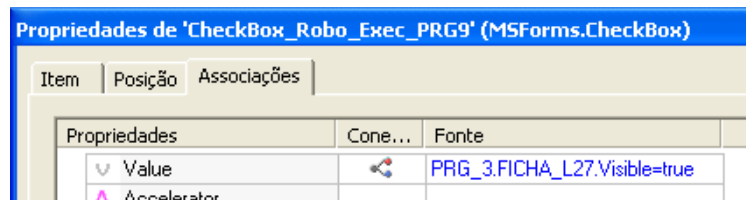
- **Robo\_Exec\_PRG7**



- **Robo\_Exec\_PRG8**

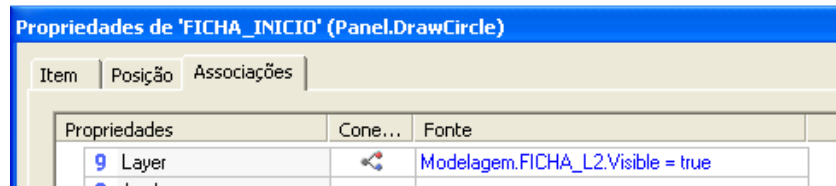


- **Robo\_Exec\_PRG9**

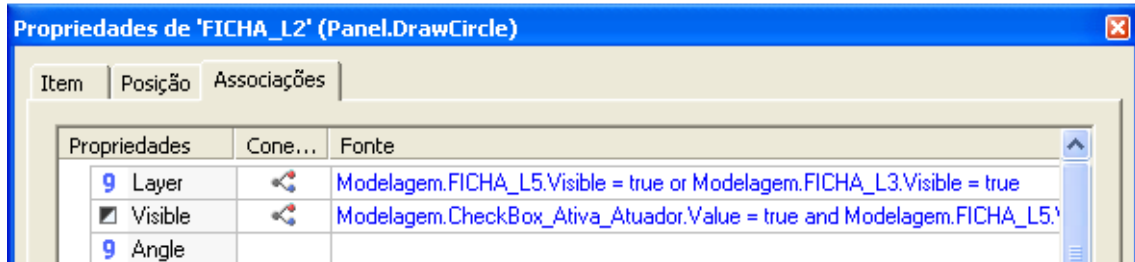




- **Ficha\_Inicio (L1)**



- **Ficha\_L2 (L2)**

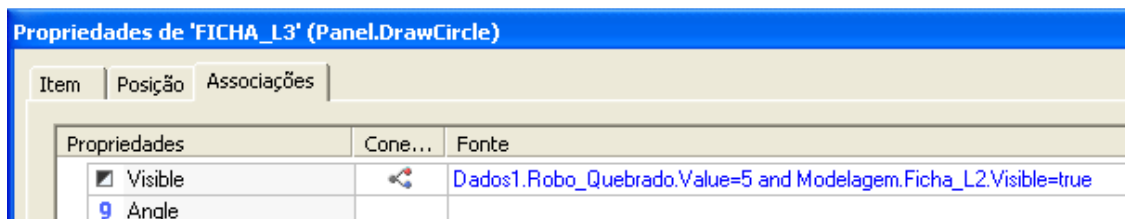


Para a propriedade Visible tem-se:

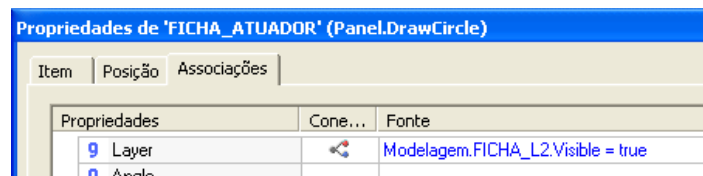
```

Modelagem.CheckBox_Ativa_Atuator.Value = true and
Modelagem.FICHA_L5.Visible=false and
Modelagem.FICHA_L5_5.Visible=false and
Modelagem.FICHA_L9.Visible=false and
Modelagem.FICHA_L9_9.Visible=false and
Modelagem.FICHA_L13.Visible=false and
Modelagem.FICHA_L13_13.Visible=false and
Modelagem.FICHA_L17.Visible=false and
Modelagem.FICHA_L17_17.Visible=false
  
```

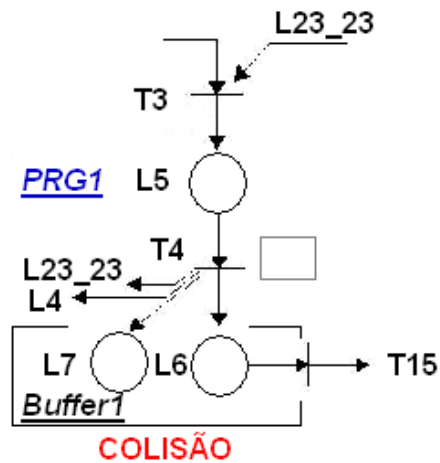
- **Ficha\_L3 (L3)**



- **Ficha\_Atuator (L4)**



- Ficha\_L5, Ficha\_L6, Ficha\_L7 e Colisão – PRG1



Propriedades de 'FICHA\_L5' (Panel.DrawCircle)

Propriedades	Cone...	Fonte
<input checked="" type="checkbox"/> Visible		Modelagem.CheckBox_PRG1.Value = true and Modelagem.FICHA_L6.Visible = false
9 Angle		

```

Modelagem PRG_2 PRG_3 Dados1
FICHA_L5 L5 Visible
L5 Visible : Chamado quando a propriedade/expressão 'Modelagem.FICHA_L5.Visible' for uma condição verdadeira.
Script <Coloque um comentário na linha 2 do script para descrevê-lo>
1 Sub [FICHA_L5_L5 Visible] ()
2 | Application.GetObject("Dados1.Exec_PRG1").Enabled = true _
3
4 End Sub

```

Propriedades de 'FICHA\_L6' (Panel.DrawCircle)

Propriedades	Cone...	Fonte
<input checked="" type="checkbox"/> Visible		Dados1.Exec_PRG1.Value=10
9 Angle		

Propriedades de 'FICHA\_L7' (Panel.DrawCircle)

Propriedades	Cone...	Fonte
<input checked="" type="checkbox"/> Visible		Modelagem.FICHA_L6.Visible = true
9 Angle		

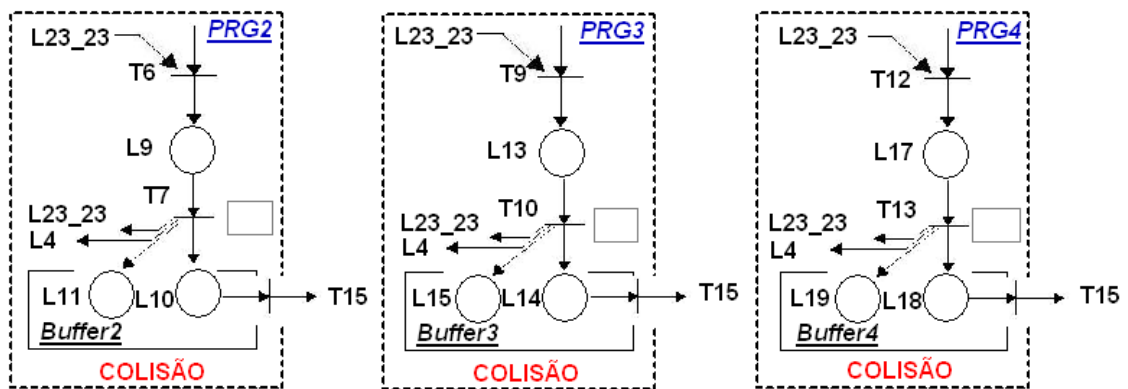
Propriedades de 'COLISÃO\_PRG1' (Panel.DrawString)

Propriedades	Cone...	Fonte
Value		
<input checked="" type="checkbox"/> Visible		Modelagem.FICHA_L5.Visible=TRUE

Propriedade	
<input checked="" type="checkbox"/> AutoRestart	<b>False</b>
CounterType	0 - Preset
<b>A</b> DocString	
<input checked="" type="checkbox"/> Enabled	<b>False</b>
<b>9</b> Increment	1
<b>A</b> Name	<b>Exec_PRG1</b>
<b>A</b> PathContainer	Dados1
<b>A</b> PathName	Dados1.Exec_PRG1
<b>A</b> PathVolume	D:\PÓS-GRADUAÇÃO\
<b>9</b> Preset	10
<input checked="" type="checkbox"/> ResetCounterWhenEnabled	<b>False</b>
<b>9</b> Value	0

Dados1.Exec\_PRG1 - Propriedades

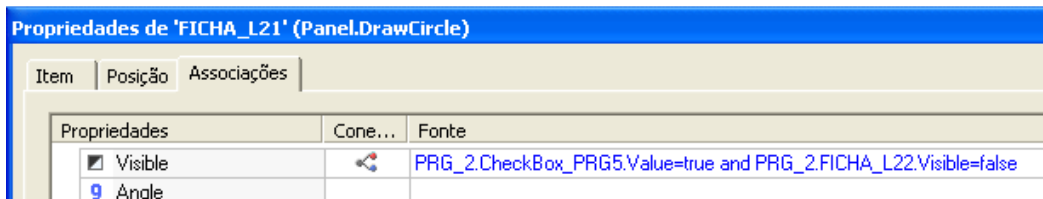
Para os programas PRG2, PRG3 e PRG4 foi desenvolvida a mesma lógica de funcionamento do programa PRG1.



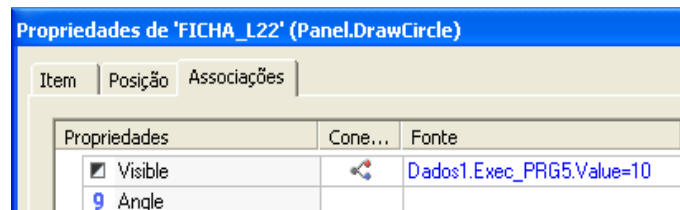
- **Ficha\_Robo (L\_23\_23)**

Propriedades de 'FICHA_ROBO' (Panel.DrawCircle)		
Item	Posição	Associações
Propriedades		
<input checked="" type="checkbox"/> Visible	↔	Modelagem.FICHA_L5.Visible = true or Modelagem.FICHA_L9.Visible = true or Modelagem.FICHA_L13.Visible = true or Modelagem.FICHA_L17.Visible = true
<b>9</b> Angle		

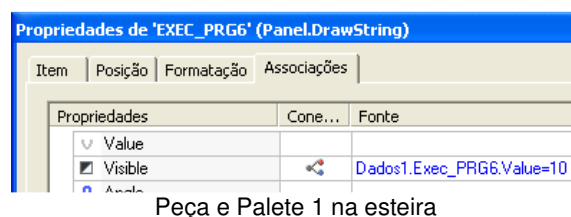
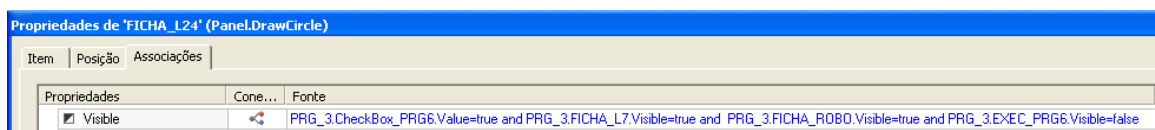
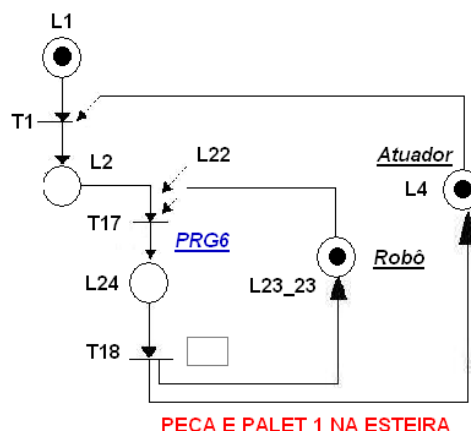
- **Ficha\_21**



- **Ficha\_22**



- **Ficha\_L24, Peça e Paleta 1 na esteira – PRG6**



Para os programas PRG7, PRG8 e PRG9 foi desenvolvida a mesma lógica de funcionamento do programa PRG6.

## **CAPÍTULO 8: REFERÊNCIAS BIBLIOGRÁFICAS**

- ALMEIDA, Francisco Rafael. **Sistema Scada e Aplicação**. Trabalho final de graduação. Universidade Federal do Ceará, 2009.
- CARDOSO, Janette; VALETTE, R. **Redes de Petri**. Editora da UFSC, Florianópolis, 1997, 212 p.
- COSTA, Eduard. **Redes de Petri: um paradigma de modelagem para a automação**. Revista Mecatrônica Atual, 2009.
- CREPALDI, Caetano P. **Redes de Petri e sua aplicabilidade na modelagem de Sistemas Computacionais**. Trabalho final de graduação. Universidade Estadual de Montes Claros, 2006.
- ELIPSE SCADA. **Manual do Usuário E3**, 2008.
- FERREIRA, J. Carlos. **Antigos e Novos Sistemas de Manufatura**. Universidade Federal de Santa Catarina, 2010.
- LIMA, Eduardo José. **Uma metodologia para a implementação através de CLPs de controle supervisorio de células de manufatura utilizando Redes de Petri**. Dissertação. Universidade Federal da Bahia, 2002.
- MELO, Marcos M. **SimRP – Simulador de Redes de Petri Flexível com geração de código VHDL**. Dissertação de Mestrado. Universidade de Brasília, 2006.
- MORAES, Cícero Couto De; CASTRUCCI, Plínio de Lauro. **Engenharia de automação industrial**. Rio de Janeiro: LTC 2007. 347 p.
- MURATA, Tadao. (1989). **Petri Nets: Properties, analysis and applications**. **Proceeding of the IEEE**, 77(4), pp. 541-580.
- PALOMINO, R. C., **Uma Abordagem para a Modelagem, Análise e Controle de Sistemas de Produção Utilizando Redes de Petri**. Dissertação (Mestrado em Engenharia da Produção). Universidade Federal de Santa Catarina, 2005.
- RIBEIRO, A. da S., **Modelagem e Especificação de controle de sistemas flexíveis de manufatura utilizando redes de Petri de Alto Nível**. Dissertação (Mestrado em Mecatrônica). Universidade Federal da Bahia , 2009.

ROSÁRIO, João Maurício. **Princípios de Mecatrônica**. São Paulo: Prentice Hall, 2007. 356 p.

SANTOS, Hugo Gaspar. **Desenvolvimento de um supervisor modular para uma célula flexível de Manufatura**. Trabalho final de graduação. Universidade Federal de Santa Catarina, 2007.

SILVA, Jucelino Alves. **Automação Industrial. Sistema SCADA**. Trabalho final de graduação. Universidade Estadual de Londrina, 2005.

VIANA, William da Silva. **Sistema Scada Supervisorio**, 2008.

ZHOU, M. C.: Twiss, E. **A comparison of relay ladder logic programming and petri nets approach for sequential industrial control systems**. In: Proceeding of the 4th IEEE Conference on Control Application, 1995, p. 748-753.