



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Curso de Graduação em Engenharia Elétrica

JONAS DANTAS DE MIRANDA NETO

**AVALIAÇÃO DE FLUXO DE TRABALHO UTILIZANDO REDES  
DE PETRI COLORIDAS: ESTUDO DE CASO DA UNIDADE DE  
NEGÓCIOS ROBÓTICA DA EMPRESA COMAU**

Campina Grande, Paraíba  
Abril de 2013

JONAS DANTAS DE MIRANDA NETO

AVALIAÇÃO DE FLUXO DE TRABALHO UTILIZANDO REDES  
DE PETRI COLORIDAS: ESTUDO DE CASO DA UNIDADE DE  
NEGÓCIOS ROBÓTICA DA EMPRESA COMAU

*Trabalho de Conclusão de Curso submetido à  
Unidade Acadêmica de Engenharia Elétrica da  
Universidade Federal de Campina Grande  
como parte dos requisitos necessários para a  
obtenção do grau de Bacharel em Ciências no  
domínio da Engenharia Elétrica.*

Área de Concentração: Controle & Automação

Orientador:

Antonio Marcus Nogueira Lima, Dr.

Campina Grande, Paraíba  
Abril de 2013

JONAS DANTAS DE MIRANDA NETO

AVALIAÇÃO DE FLUXO DE TRABALHO UTILIZANDO REDES  
DE PETRI COLORIDAS: ESTUDO DE CASO DA UNIDADE DE  
NEGÓCIOS ROBÓTICA DA EMPRESA COMAU

Trabalho de Conclusão de Curso submetido à Unidade  
Acadêmica de Engenharia Elétrica da Universidade  
Federal de Campina Grande como parte dos requisitos  
necessários para a obtenção do grau de Bacharel em  
Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Controle & Automação

Aprovado em        /        /

**Professor Avaliador**  
Universidade Federal de Campina Grande  
Avaliador

**Professor Antonio Marcus Nogueira Lima, Dr.**  
Universidade Federal de Campina Grande  
Orientador, UFCG

Dedico este trabalho aos meus pais, meu irmão, à minha família, amigos e namorada por todo apoio e carinho dado.

## AGRADECIMENTOS

À Deus, “porque Dele e por Ele, e para Ele, são todas as coisas; glória, pois, a Ele eternamente. Amém.” (Rm 11:36).

Aos meus pais, Jonas Dantas de Miranda Filho e Walquíria da Costa Miranda, por todo carinho, amor, atenção, dedicação e presença na minha vida. Agradeço pelas lições e ensinamentos, além das demonstrações de ética, caráter, responsabilidade, coragem e perseverança que estarão sempre comigo. Com certeza, se fui capaz de atingir tal feito, foi pelo apoio incondicional recebido destes dois grandes guerreiros, que, com muito orgulho, tenho o prazer de chamar de “*meu pai*” e “*minha mãe*”.

Ao meu irmão, Wilson da Costa Miranda, que sempre foi fonte de inspiração e motivação para que eu melhorasse a cada dia, agindo de maneira correta e honesta. Obrigado pelas risadas e gargalhadas, além dos momentos de companheirismo, “*biaaatch*”!

À *Minha Chica*, Juliana de Sales Silva, por ter me apoiado e dado suporte, além de aguentar minhas ausências, aperreios e brincadeiras.

À minha família, por sempre estar ao meu lado, dando muito amor, carinho, amparo e alegrias.

À família da minha namorada, por sempre me receber e tratar com respeito e atenção.

Ao meu orientador, que facilitou a realização deste trabalho, compartilhando conhecimentos e guiando para que o mesmo fosse feito da melhor maneira.

A todos os professores da Universidade Federal de Campina Grande e da *Iowa State University* que fizeram parte deste meu caminho, por terem partilhado suas ideias, conceitos, teorias e erudição.

À CAPES, pela bolsa concedida para que eu pudesse aprofundar meus estudos e conhecimentos nos Estados Unidos, realizando um intercâmbio acadêmico.

Ao Professor Dr. José Sérgio da Rocha Neto, pela oportunidade de trabalhar no LIEC e bolsa de pesquisa concedida.

À professora Dr. Maria de Fátima Queiroz Vieira e aos integrantes do Laboratório de Interface Homem-Máquina – LIHM, especialmente Raffael Carvalho e Ademar Netto.

À Adail Ferreira da Silva Paz e Tchaikovsky Oliveira, por todo suporte prestado na Coordenação do curso de Engenharia Elétrica.

A todos os amigos e grandes amigos que conquistei durante o curso: as noites em claro foram mais prazerosas e fáceis de resistir em virtude da presença de vocês. Em especial: José Maurício, Luiz Carlos, Tony Moura, Sidney Aciole, Walter Vermehren, Anderson Machado, Antonio Agripino, Felipe Nóbrega, Daphne De La Torre, Juan Cruz, Maria Leticia, Pierre Camilo, Cândido Neto, Caio Luiz, Daniel Aguiar, Ricardo Chinarro, Coriolano Sá, Leonardo Campos, Cintya Brilhante, Mayanna Bú, Arthur Pequeno, Yonatha Melo, José Júnior, Igor Torres, Camila Souto, Diego Lopes, João Cabral, Jessiedna Araújo, José do Patrocínio, Mariana Herminia, irmãos Fonseca e Fábio Brito (*in memoriam*)

Ao “rachinha da sexta”, pelas estórias, lances memoráveis e momentos de descontração.

*“No que diz respeito ao empenho, ao compromisso, ao esforço e à dedicação, não existe meio termo. Ou você faz uma coisa bem ou não faz.”*

Ayrton Senna

## RESUMO

O presente trabalho tem por objetivo avaliar o fluxo de trabalho, em um estudo de caso, utilizando redes de Petri Coloridas. Também busca levantar características do processo, bem como verificar o impacto de falhas e distribuições de recursos. O enfoque dado é no tempo dispendido no sequenciamento, tentando otimizá-lo. Para tanto, baseado nos conceitos de redes de Petri Coloridas e *workflow*, foi criado um modelo de representação na ferramenta *CPN Tools*. Diversos cenários foram simulados, variando a quantidade de robôs que deveriam ser trabalhados e a quantidade de pessoas disponíveis para realizar as tarefas. Com o estudo, chegou-se a conclusão que a forma com que os recursos são escolhidos e distribuídos afetam diretamente o desempenho do processo. Também foram identificados pontos de retenção e a melhor forma para definir a equipe de trabalho.

**Palavras-chave:** *CPN Tools*, Redes de Petri Coloridas, *Workflow*, Distribuição de Recursos.

## ABSTRACT

This study aims to evaluate the workflow in a case study using Coloured Petri nets. It also seeks to identify characteristics of the process, as well as verify how failures and a resource-constrained scenario affect the overall performance. The focus of the study is to optimize activities execution time. To do so, based on the concepts of Coloured Petri nets and workflow, a model was developed using the software CPN Tools. Several scenarios were simulated, varying the number of robots and the resources available. The study concluded that the way in which resources are chosen and distributed directly affects the performance of the process. It also pointed out how to best build a team and activities that may block the process.

**Keywords:** CPN Tools, Coloured Petri Nets, Workflow, Resource-Constrained.

## LISTA DE ILUSTRAÇÕES

FIGURA 1. RELAÇÃO ENTRE TAREFA, ITEM EM TRABALHO E EVENTO. ....	6
FIGURA 2. REDE MARCADA. ....	8
FIGURA 3. REDE APÓS DISPARO DA TRANSIÇÃO <i>TI</i> .....	9
FIGURA 4. REDE COM <i>SELF-LOOP</i> . ....	10
FIGURA 5. REDE PURA. ....	10
FIGURA 6. REPRESENTAÇÃO DO SEQUENCIAMENTO DE ATIVIDADES. ....	11
FIGURA 7. REPRESENTAÇÃO DA DISTRIBUIÇÃO DE FICHAS. ....	11
FIGURA 8. SINCRONIZAÇÃO DE ATIVIDADES.....	11
FIGURA 9. ESCOLHA NÃO DETERMINÍSTICA. ....	11
FIGURA 10. REPRESENTAÇÃO DE GRAFOS DIRECIONADOS.....	14
FIGURA 11. GERADOR DE FICHAS. ....	25
FIGURA 12. SUBPÁGINA DO GERADOR. ....	25
FIGURA 13. BLOCO DE ATIVIDADE. ....	26
FIGURA 14. SUBPÁGINA DO BLOCO DE ATIVIDADE. ....	26
FIGURA 15. BLOCO DE RECURSOS. ....	27
FIGURA 16. SUBPÁGINA DO BLOCO DE RECURSOS. ....	27
FIGURA 17. BLOCO DE FALHA. ....	28
FIGURA 18. SUBPÁGINA DO BLOCO DE FALHA. ....	29
FIGURA 19. LIGAÇÃO ENTRE BLOCO DE ATIVIDADE E BLOCO DE RECURSO. ....	30
FIGURA 20. PARTE DA REDE CRIADA PARA REPRESENTAÇÃO DO PROCESSO. ....	30
FIGURA 21. VISUALIZAÇÃO DA REDE COMPLETA SEM PARALELISMO. ....	31
FIGURA 22. REDE COM DESENVOLVIMENTO DE ATIVIDADES EM PARALELO CRIADA NO <i>CPN Tools</i> . ....	32
FIGURA 23. TELA DA FERRAMENTA <i>CPN Tools</i> . ....	32
FIGURA 24. UMA DAS TOPOLOGIAS DE REDE PENSADAS PARA SIMULAÇÃO. ....	34
FIGURA 25. CONCEITO FINAL DA TOPOLOGIA DA REDE. ....	35
FIGURA 26. GRÁFICO EM QUE PODE SER VISUALIZADA A TAXA DE UTILIZAÇÃO (%) EM FUNÇÃO DO NÚMERO DE FICHAS PARA A CONFIGURAÇÃO SETE.....	44

# LISTA DE TABELAS

TABELA 1. VANTAGENS NO USO DE WORKFLOW .....	4
TABELA 2. DADOS DAS SIMULAÇÕES COM CINCO PESSOAS E DEZ FICHAS.....	37
TABELA 3. DADOS DAS SIMULAÇÕES COM CINCO PESSOAS E VINTE FICHAS. ....	37
TABELA 4. DADOS DAS SIMULAÇÕES COM CINCO PESSOAS E TRINTA FICHAS. ....	38
TABELA 5. DADOS DAS SIMULAÇÕES COM SETE PESSOAS E DEZ FICHAS. ....	39
TABELA 6. DADOS DAS SIMULAÇÕES COM SETE PESSOAS E VINTE FICHAS.....	39
TABELA 7. DADOS DAS SIMULAÇÕES COM SETE PESSOAS E TRINTA FICHAS. ....	39
TABELA 8. TEMPO DE EXECUÇÃO TOTAL PARA AS CONFIGURAÇÕES 02 E 07.....	40
TABELA 9. RESULTADOS PARA SIMULAÇÕES COM E SEM FALHAS PARA A CONFIGURAÇÃO SETE E DEZ FICHAS.....	42
TABELA 10. TEMPO TOTAL DE EXECUÇÃO PARA A CONFIGURAÇÃO SETE E DEZ FICHAS. ....	42
TABELA 11. RESULTADOS PARA SIMULAÇÕES COM E SEM FALHAS PARA A CONFIGURAÇÃO SETE E VINTE FICHAS.....	42
TABELA 12. RESULTADOS PARA SIMULAÇÕES COM E SEM FALHAS PARA A CONFIGURAÇÃO SETE E TRINTA FICHAS.....	42
TABELA 13. TEMPO TOTAL DE EXECUÇÃO PARA A CONFIGURAÇÃO SETE E VINTE FICHAS.....	43
TABELA 14. TEMPO TOTAL DE EXECUÇÃO PARA A CONFIGURAÇÃO SETE E TRINTA FICHAS.....	43
TABELA 15. TEMPOS OBTIDOS COM A REDE EM PARALELO PARA A SÉTIMA CONFIGURAÇÃO.....	44

# SUMÁRIO

1	Introdução.....	1
1.1	Objetivo Geral.....	2
1.2	Objetivos Específicos.....	2
1.3	Estrutura da Monografia .....	2
2	Fundamentação Teórica.....	3
2.1	<i>Workflow</i> .....	3
2.2	Elementos do <i>Workflow</i> .....	5
2.2.1	Eventos .....	5
2.2.2	Tarefa.....	6
2.2.3	Processo.....	6
2.2.4	Roteamento.....	6
2.2.5	Ativação.....	7
2.3	Redes de Petri .....	7
2.3.1	Definição Formal .....	7
2.3.2	Redes Marcadas e Transições .....	8
2.3.3	Redes Elementares.....	10
2.4	Redes de Petri Coloridas Não Hierárquicas .....	12
2.4.1	Definição Formal .....	12
2.4.2	Componentes de uma Rede de Petri Colorida .....	12
2.5	Propriedades de Redes de Petri .....	14
2.5.1	Grafos Direcionais .....	14
2.5.2	Espaço de Alcançabilidade .....	15
2.5.3	Alcançabilidade .....	15
2.5.4	Reversibilidade .....	15
2.5.5	Vivacidade .....	16
2.6	Redes de Petri Coloridas Hierárquicas.....	16
2.6.1	Módulos.....	17
2.6.2	Definição Formal .....	17
2.7	Redes de Petri Temporizadas .....	18
2.8	Redes de Petri no Contexto de <i>Workflow</i> .....	19
2.9	Justificativa para Estudo do Problema .....	20
3	Metodologia.....	22
3.1	Descrição do Problema .....	22
3.2	Pressupostos.....	23
3.3	Elaboração do Modelo .....	24
3.3.1	Gerador.....	25
3.3.2	Bloco de Atividade .....	25
3.3.3	Bloco de Recurso.....	27

3.3.4	Bloco de Falha .....	28
3.3.5	Conceito Geral .....	29
3.4	<i>CPN Tools</i> .....	32
4	Resultados .....	34
4.1	Evolução do Modelo .....	34
4.2	Características para Simulação .....	35
4.3	Equipe com Cinco Pessoas .....	37
4.4	Equipe com Sete pessoas .....	38
4.5	Análise do Processo .....	41
4.6	Atraso por Falha e Taxa de Utilização .....	41
4.7	Paralelismo Intrínseco .....	44
5	Conclusão .....	46
	Bibliografia .....	48

# 1 INTRODUÇÃO

Pode-se caracterizar a economia atual pelo rápido crescimento da competição global, o que torna necessário que empresas e organizações, entre outros aspectos, revejam a maneira de realizar negócios, sua estrutura organizacional e a condução de processos internos. Neste contexto, melhorar processos do negócio tem sido, com frequência, o foco das atenções das organizações, já que as empresas buscam reduzir custos e aumentar lucros, melhorando sua eficiência.

Com o desenvolvimento tecnológico, novas formas de aumento de produtividade foram possibilitadas em diversos setores e indústrias. Na indústria automobilística, por exemplo, a introdução de robôs na linha de montagem acarretou diversos benefícios, como: maior exatidão na realização de tarefas, maior confiabilidade, aumento na quantidade de horas trabalhadas e menor tempo de execução, além de possuírem a capacidade de realizar esforços mecânicos, como transporte de cargas e peças pesadas, superiores à capacidade humana.

Longe de ser apenas um sonho capitalista, o panorama de crescimento baseado em automação industrial e de processos, além da robotização de linhas de montagem, se tornou necessário para que empresas se mantenham vivas em um mercado que, dia a dia, torna-se mais dinâmico, competitivo e globalizado. E é neste cenário de automação que entra a Comau do Brasil Indústria e Comércio Ltda., já que é uma empresa líder em soluções robotizadas.

Contudo, apesar de ser líder em soluções robotizadas e instalação de linhas de produção, as atividades realizadas nos robôs, preparando-os para instalação, são executadas manualmente, um processo realizado por diversas pessoas e que envolve diferentes atividades, o que acarreta em variação na forma de execução, podendo, dessa forma, gerar um resultado não desejado e atrasos.

Surge, portanto, a necessidade de uma análise mais profunda dos processos organizacionais realizados por esta empresa. Dessa forma, para satisfazer estas necessidades, nascem dois conceitos: a reengenharia, que consiste em uma análise dos processos organizacionais para repensá-los, melhorá-los e reimplantá-los, e *workflow*. Este está relacionado à automação dos processos e se desenvolveu e evoluiu tanto que

hoje podemos nos referir a ele como um conjunto de tecnologias e metodologias de automação de processos: a tecnologia de *workflow*.

*Workflow* é uma tecnologia que vem sendo cada vez mais adotada por empresas na coordenação de processos, principalmente em empresas que apresentam alto grau de dinamismo nos seus processos. Pode-se dizer que o objetivo principal do *workflow* é capturar a descrição formal dos processos do negócio e dar suporte a automação destes.

## 1.1 OBJETIVO GERAL

Este trabalho objetiva avaliar o fluxo de trabalho realizado na unidade de negócio Robótica da Comau do Brasil Indústria e Comércio Ltda. utilizando redes de Petri Coloridas.

## 1.2 OBJETIVOS ESPECÍFICOS

Com este trabalho pretende-se:

- ✓ criar um modelo para representação das atividades realizadas;
- ✓ avaliar o impacto de falhas;
- ✓ identificar atividades que podem causar retenções;
- ✓ verificar o impacto da escolha e distribuição de recursos.

## 1.3 ESTRUTURA DA MONOGRAFIA

Este documento está organizado da seguinte forma: no Capítulo 2 é apresentado a Fundamentação Teórica, em que são discutidos conceitos sobre *workflow*, Redes Petri e Redes Petri Coloridas, além de definições matemáticas sobre este tema. Já no Capítulo 3, tem-se a Metodologia utilizada no trabalho, a descrição do problema em questão, bem como a maneira de elaboração do modelo utilizado e hipóteses de base. Em seguida, no Capítulo 4, são apresentados os Resultados. Neste é feito a análise dos resultados obtidos. No Capítulo 5, o último deste trabalho, tem-se a Conclusão.

## 2 FUNDAMENTAÇÃO TEÓRICA

No decorrer deste capítulo serão apresentados conceitos de *workflow*, Redes Petri e Redes Petri Coloridas, bem como as respectivas definições formais.

### 2.1 WORKFLOW

*Workflow* pode ser entendido como a automação de procedimentos organizacionais, em parte ou integralmente, cujos documentos, informações ou atividades são passadas de um participante para outro de acordo com um conjunto definido de regras, para atingir ou contribuir com o alcance dos objetivos de uma organização (HOLLINGSWORTH,1995).

Um *workflow* pode ser feito, organizado e executado de forma manual, contudo, geralmente, é feito o uso de ferramentas de *software* ou computadores para sua implementação. A relação entre *workflow* e automação é feita constantemente, mas, a definição feita por SARIN e ABBOT (SARIN et. al., 1994) destaca essa diferença conceitual: “a ênfase dos sistemas de *workflow* está em usar computadores para auxiliar a execução e gerenciamento de processos do negócio, que são compostos por diversas tarefas individuais, e não no uso de computadores para automatizá-las.”.

Outras definições de *workflow* são citadas abaixo:

- ✓ segundo ELLIS (1995), os sistemas de *workflow* são pensados e projetados para auxiliar grupos de pessoas a realizar seus procedimentos no trabalho, armazenando o conhecimento de como os fluxos de trabalho devem ocorrer num caso padrão;
- ✓ um sistema de *workflow* define, cria e gerencia a execução de um fluxo de trabalho utilizando um *software*. Enfim, os sistemas de *workflow* modelam e executam os processos de negócio. (GARY et al., 1997).

Em adição as definições apresentadas, outros dois tópicos são, em geral, abordados e desejados por usuários de sistemas de *workflow*: possibilidade de distribuir

peças em grupos, respeitando regras de organização, e monitoramento de aspectos da execução do *workflow*, gerando dados para avaliação do progresso em um processo do negócio ou em parte dele.

É possível observar e notar um conceito intuitivo bastante forte do que é *workflow*. Este conceito está associado à melhoria dos processos, nos quais a tecnologia de *workflow* é utilizada para analisá-los de forma coerente, melhorá-los e controlá-los para se atingir um melhor resultado. Desse modo, soluções de *workflow* buscam a coordenação de atividades utilizando um modelo do processo comum sem, necessariamente, ocorrer a automação das atividades.

Portanto, o propósito de um sistema de *workflow* é coordenar as entidades envolvidas na execução de um processo. A coordenação pode ser entendida como o gerenciamento das dependências entre as atividades. Dois tipos de problemas de coordenação que podem ser lidados pelo gerenciamento de *workflow* são:

- ✓ dependência de dados entre atividades, em que uma atividade depende do resultado de uma ou mais atividades;
- ✓ compartilhamento de recursos disponíveis, em que, por exemplo, uma pessoa ou determinada máquina só é capaz de realizar uma atividade por vez.

O uso de sistemas de *workflow* trazem diversas vantagens às organizações. Na tabela a seguir (BECKER et al., 2002), alguns desses benefícios são apresentados:

Tabela 1. Vantagens no uso de *workflow*

<b>Metas de Eficiência</b>	<b>Descrição</b>	<b>Atividades Suportadas pelos Sistemas de Workflows</b>
Eficiência do Processo	Otimização de características do processo, como tempo de execução, número de falhas (a serem minimizados) e cumprimento de prazos (a ser maximizado).	Coordenação de atividades através do controle do fluxo, de prazos, etc.
Eficiência dos Recursos	Uso eficiente dos recursos disponíveis para execução dos processos.	Coordenação de cronogramas de execução e de escalonamento de recursos.

Eficiência de Mercado	Posicionamento da organização no relacionamento com os parceiros de mercado. Inclui: previsão confiável de prazos de entrega, comunicação transparente e eficiente com fornecedores e clientes, processos otimizados de aquisição e distribuição.	Previsão comportamental/departamental de atividades e resultados através da padronização de procedimentos.
-----------------------	---	--

As maneiras como os sistemas de *workflow* podem ser implementados são bem diversas e distintas, com o uso de diferentes tecnologias e infraestruturas. Contudo, elementos comuns estão presentes em todos os sistemas de *workflow*. No tópico a seguir estes elementos são explicados mais detalhadamente. Vale ressaltar que, em geral, o sistema de *workflow* não contém detalhes sobre o conteúdo do processo, somente atributos e condições necessárias para o desenvolvimento do processo.

## 2.2 ELEMENTOS DO *WORKFLOW*

Um sistema de *workflow* é composto por cinco entidades, as quais são explicadas e detalhadas abaixo. É importante salientar que o sucesso de um sistema de *workflow* depende da qualidade do modelo criado para representação do processo (AALST e HEE, 2002).

### 2.2.1 EVENTOS

O objetivo primário de um sistema de *workflow* é trabalhar com eventos. Exemplos de eventos são: entrada de um paciente em um hospital, realização de empréstimo em um banco, uso do seguro, entre outros. Cada evento tem uma identidade única, o que torna possível referenciar o evento em questão. Além disso, cada evento tem um tempo de vida limitado, ou seja, ele tem início e fim. Entre o início e fim do evento, este sempre apresenta um estado característico. Este estado consiste de três elementos: os atributos do evento, as condições que foram satisfeitas e o conteúdo do evento. Diversas variáveis podem ser associadas a cada evento, e os atributos deste são utilizadas para gerenciar e verificar seu acompanhamento.

### 2.2.2 TAREFA

A identificação de tarefas possibilita a estruturação de *workflows*. Conceitualmente, a tarefa é uma unidade de trabalho indivisível e, por isso, é sempre realizada completamente. Contudo, a natureza atômica da tarefa depende do contexto no qual a tarefa está inserida e como ela é definida.

Quando mencionamos o termo ‘tarefa’ não nos referimos à execução de uma tarefa em um processo. Para evitar confusão no entendimento, é comum denominar um item em execução como ‘item em trabalho’ (*work item*, em inglês). Desse modo, *work item* é a combinação entre o evento e a tarefa que está para ser iniciada.

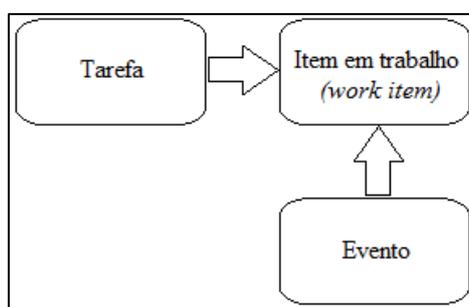


Figura 1. Relação entre tarefa, item em trabalho e evento.

### 2.2.3 PROCESSO

O processo indica quais tarefas precisam ser colocadas em execução e mostra a ordem em que estas devem ser executadas. A ordem em que as tarefas são iniciadas varia de acordo com as características de cada evento. Assim, condições são utilizadas para decidir qual ordem deverá ser seguida. Portanto, processo pode ser entendido como a combinação de tarefas e condições.

Pode-se, também, entender o termo processo como o procedimento que deverá ser adotado para um determinado tipo de evento.

### 2.2.4 ROTEAMENTO

O ciclo de vida de um evento é definido no processo. Neste sentido, isto é definido como o roteamento do evento. A interação entre diferentes partes do evento determina quais tarefas precisam ser executadas e em que ordem. Três tipos de roteamento são interessantes para o desenvolvimento deste trabalho:

- ✓ roteamento sequencial: é a forma mais simples de roteamento. As tarefas são conduzidas uma após a outra. Usualmente existe uma dependência evidente entre as tarefas ligadas sequencialmente;
- ✓ roteamento paralelo: é quando duas tarefas podem ser executadas ao mesmo tempo. Neste caso, a execução de uma tarefa não interfere na outra;
- ✓ roteamento seletivo: quando se tem uma escolha entre tarefas. Esta escolha depende de propriedades do evento em questão.

### 2.2.5 ATIVAÇÃO

Uma tarefa só pode ser executada se o estado do evento permita, ou seja, se todas as condições para que a mesma seja realizada forem satisfeitas. Assim, se a tarefa está pronta para ser executada, mas os recursos necessários não estiverem disponíveis, esta não será ativada. Isto é o que se chama de *disparo*: a tarefa se torna um item em execução quando é disparada por um recurso.

## 2.3 REDES DE PETRI

As redes de Petri são uma ferramenta gráfica e matemática que pode ser utilizada em um número diverso de aplicações em que o entendimento de eventos e evoluções simultâneas é importante. Ela pode ser utilizada para se estudar e obter informações de sistemas de processamento de informação que são caracterizados por serem concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e/ou estocásticos.

### 2.3.1 DEFINIÇÃO FORMAL

Formalmente, a estrutura de redes de Petri é formada por uma quintupla (MURATA, 1989):

$$PN = (P, T, F, W, M_0), \quad (1)$$

em que:

- ✓  $P = \{p_1, p_2, \dots, p_m\}$  é um conjunto finito de lugares;
- ✓  $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições;

- ✓  $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos;
- ✓  $W : F \rightarrow (1, 2, 3, \dots)$  é uma função de ponderação;
- ✓  $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  é a marcação inicial e  $P \cap T = \emptyset$  e  $P \cup T \neq \emptyset$ .

Se a rede de Petri não tiver nenhuma marcação inicial específica, ela é denotada por:

$$N = (P, T, F, W). \quad (2)$$

### 2.3.2 REDES MARCADAS E TRANSIÇÕES

Para representar o estado da rede em um determinado momento, são utilizadas marcas (*tokens* em inglês, *jetons* em francês), que pode ser entendido como informações atribuídas aos lugares. Pode-se definir uma rede marcada pela dupla  $(N, M_0)$ , em que  $N$  é a estrutura da rede e  $M_0$  é a marcação inicial. A marcação  $M_0$  é a distribuição das fichas nos lugares, sendo representado por um vetor coluna cuja dimensão é o número de lugares e elementos (CARDOSO; VALETTE, 1997). Na Figura 2 é possível visualizar esta condição.

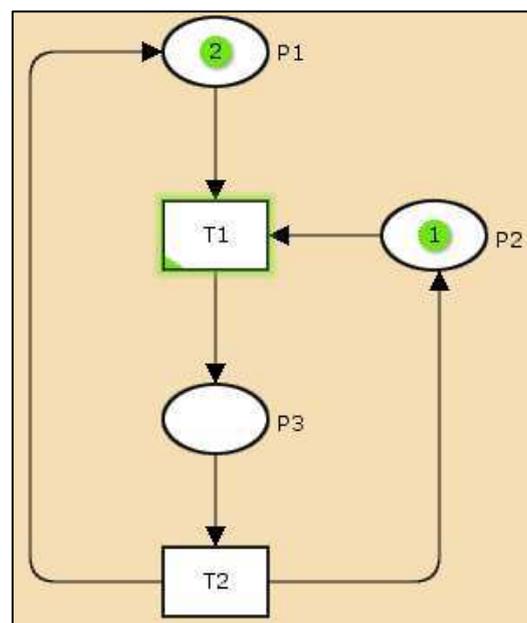


Figura 2. Rede marcada.

O comportamento de vários sistemas pode ser descritos em função de estados de sistema e suas mudanças. Para simular o comportamento dinâmico de um sistema, o

estado ou a marcação em uma rede de Petri muda de acordo com as seguintes regras de transição:

1. Uma transição  $t$  é dita habilitada se cada lugar de entrada  $p$  de  $t$  está marcada com pelo menos  $w(p,t)$  fichas, em que  $w(p,t)$  é o peso do arco de  $p$  para  $t$ .
2. Uma transição habilitada pode ou não ocorrer, dependendo se o evento ocorreu ou não.
3. A ocorrência de uma transição habilitada consome  $w(p,t)$  fichas de um lugar de entrada  $p$  de  $t$  e adiciona  $w(p,t)$  fichas para cada lugar de saída  $p$  de  $t$ , sendo que  $w(p,t)$  é o peso do arco de  $t$  para  $p$ .

Na Figura 2, é possível ver que no lugar  $P1$  possui duas fichas e o lugar  $P2$  contém uma ficha. Também é possível visualizar na Figura 2 que a transição  $T1$  está habilitada. Quando a transição  $T1$  ocorre, fichas são consumidas dos lugares  $P1$  e  $P2$  e são adicionadas ao lugar  $P3$ , como pode ser visto na Figura 3. Com a ocorrência de  $T1$ , o estado do sistema mudou e agora a transição  $T2$  está habilitada, pois as condições para que a mesma ocorra são satisfeitas.

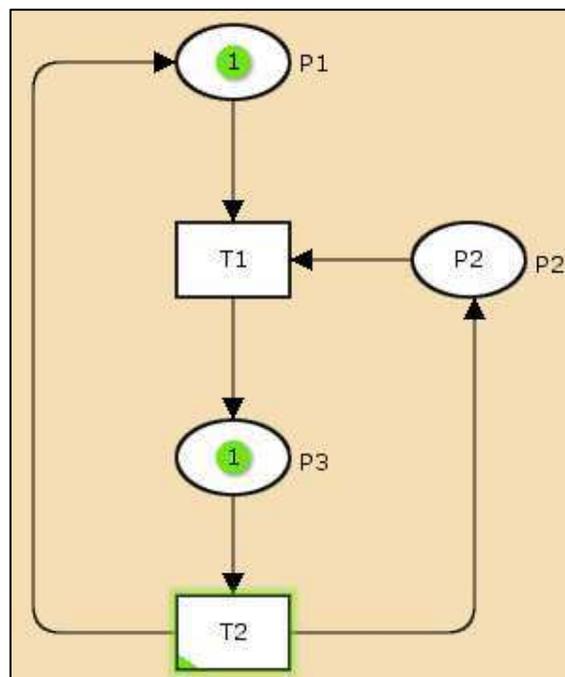


Figura 3. Rede após disparo da transição  $T1$ .

Uma transição que não possui nenhum lugar de entrada associado com ela é chamada de uma *transição fonte* e uma sem nenhum lugar de saída é chamado de uma

*transição poço*. É importante observar que uma transição fonte está sempre habilitada e que a ocorrência de uma transição poço consome fichas, mas não gera nenhuma.

Outra característica de uma rede de Petri é que se  $p$  é um lugar de entrada e um lugar de saída de uma transição  $t$  ao mesmo tempo, a rede Petri é chamada de *auto laço* (*self-loop*) (Figura 4). Uma rede de Petri é dita *pura* se não tem auto laço (Figura 5). Se uma rede de Petri tem peso unitário em todos os seus arcos, ela é classificada como *ordinária*.

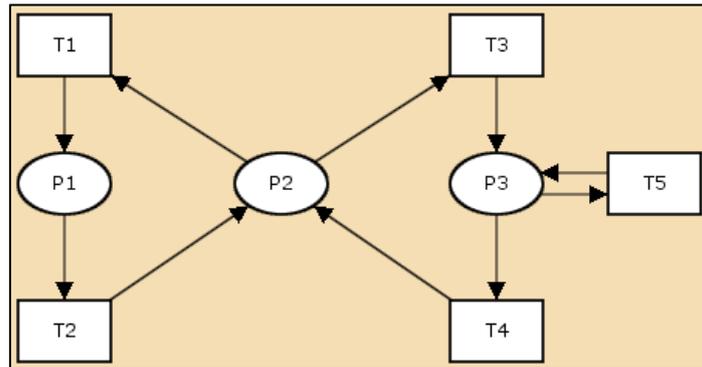


Figura 4. Rede com *self-loop*.

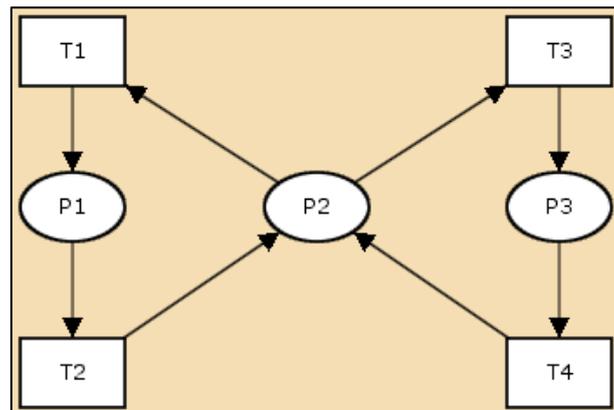


Figura 5. Rede pura.

### 2.3.3 REDES ELEMENTARES

Aqui serão apresentadas e representadas redes que possuem conceitos básicos, mas que podem ser aplicadas ou combinadas em redes, possibilitando a criação de redes bem complexas (FRANCÊS, 2003).

1. Sequenciamento: ocorre quando a realização de uma ação está condicionada a finalização de uma ação anterior.

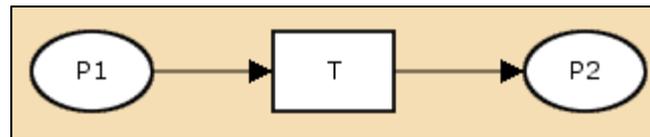


Figura 6. Representação do sequenciamento de atividades.

2. Distribuição: este tipo de construção é utilizado para criação de atividades paralelas a partir de uma atividade inicial.

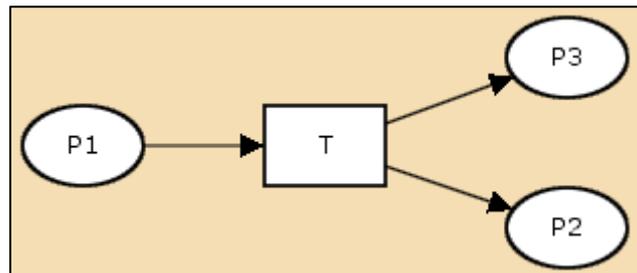


Figura 7. Representação da distribuição de fichas.

3. Junção ou sincronização: é a construção utilizada para unir duas atividades em um ponto comum.

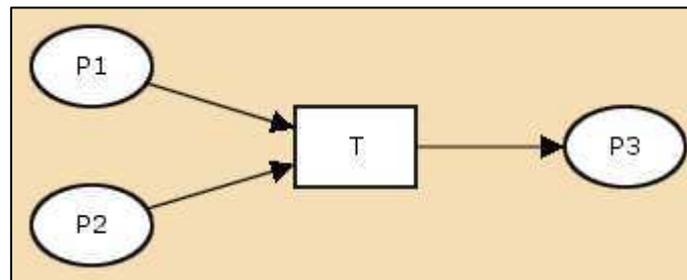


Figura 8. Sincronização de atividades.

4. Ocorrência não determinística: acontece quando a ficha é passada para um ramo da rede em virtude uma escolha não controlada.

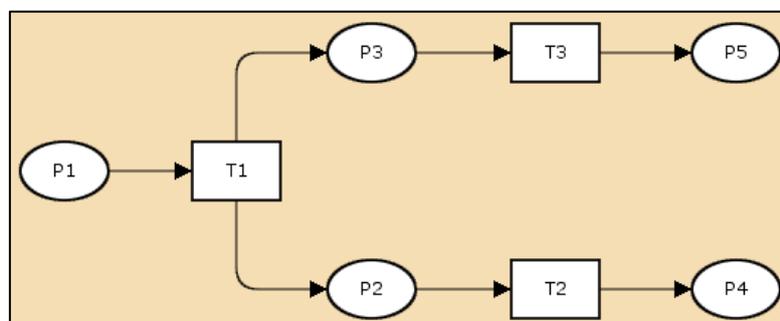


Figura 9. Escolha não determinística.

## 2.4 REDES DE PETRI COLORIDAS NÃO HIERÁRQUICAS

Segundo JENSEN (2009), redes de Petri Coloridas (em inglês, *CP-nets* ou *CPNs*) é uma linguagem gráfica para construção de modelos de sistemas concorrentes, bem como a análise de suas propriedades. Uma *CPN* é linguagem para modelagem a eventos discretos que combina as características de redes de Petri convencionais com a linguagem de programação de alto nível.

A linguagem de modelagem utilizada por redes de Petri Coloridas não são direcionadas para uma classe de sistema específica, mas tem como objetivo permitir a representação de uma grande variedade de classes de sistemas que podem ser caracterizados como concorrentes. Exemplos destes sistemas são processos de negócios, *workflow* e negócios de manufaturas.

### 2.4.1 DEFINIÇÃO FORMAL

Uma rede de Petri colorida não hierárquica pode ser definida como a seguinte tupla:

$$CPN = (P, T, A, \Sigma, N, C, G, E, I), \quad (3)$$

em que:

1.  $P$  é um conjunto finito de *lugares*;
2.  $T$  é um conjunto finito de *transições*;
3.  $A$  é um conjunto finito de *arcos* tal que  $P \cap T = P \cap A = T \cap A = \emptyset$ ;
4.  $\Sigma$  é um conjunto finito e não vazio de *cores*;
5.  $N$  é a *função* nó. Esta é definida como  $N : A \rightarrow P \times T \cup T \times P$ ;
6.  $C : P \rightarrow \Sigma$  é a *função cor*;
7.  $G : T \rightarrow \text{EXPR}_V$  é a *função de guarda*;
8.  $E : A \rightarrow \text{EXPR}_V$  é a *função de arco*;
9.  $E : A \rightarrow \text{EXPR}_\emptyset$  é a *função de inicialização*;

### 2.4.2 COMPONENTES DE UMA REDE DE PETRI COLORIDA

- ✓ O conjunto de *cores* determina as funções, operações e tipos de dados que podem ser utilizados nas inscrições de rede, como expressões de arco, guardas e expressões de inicialização;

- ✓ *Arcos, lugares e transições* são definidos por três conjuntos:  $A$ ,  $P$  e  $T$ , respectivamente. Por definição, estes conjuntos são finitos para evitar problemas, como a possibilidade de ter um número infinito de arcos entre dois nós;
- ✓ A função *nó* associa cada arco a um par, sendo que o primeiro elemento é o nó de origem e o segundo o nó de destino. Os dois nós têm de ser de tipos diferentes, isto é, um deve ser um lugar, enquanto que o outro deve ser uma transição;
- ✓ A função *cor* associa cada lugar  $p$  da rede a uma cor  $C(p)$ . Isto quer dizer que cada ficha no lugar  $p$  deve ter um tipo pertencente ao conjunto de cores;
- ✓ A função *guarda*  $G$  associa cada transição  $t$  a uma expressão booleana em que todas as variáveis possuem tipos pertencentes a  $\Sigma$ . Ao criar uma rede, é possível omitir as expressões de guarda, que sempre serão consideradas verdadeiras;
- ✓ A função *de arco*  $E$  associa cada arco  $a$  da rede a uma expressão. Isto significa que cada expressão de arco relaciona um multiconjunto a um conjunto de cores do lugar  $p$  adjacente;
- ✓ A função *de inicialização*  $I$  associa cada lugar da rede a uma expressão, em que cada expressão é um multiconjunto sobre o conjunto de cores.

Dessa forma, uma *CPN* é composta essencialmente por uma estrutura, um conjunto de inscrições e um conjunto de declarações. Como as redes de Petri, as *CPNs* são também grafos direcionados e bipartidos. Entretanto, ao invés de pesos inteiros, aos arcos são associadas inscrições que determinam dinamicamente quantas e quais fichas devem ser removidas ou adicionadas aos lugares associados, na ocorrência de uma transição. Inscrições, denominadas guardas, podem ser também associadas às transições. Guardas restringem a ocorrência de transições a determinadas condições. O estado inicial de uma *CPN* também é determinado por inscrições associados aos lugares. Cada inscrição é, em geral, uma expressão construída a partir de constantes, variáveis e operadores previamente definidos. Uma *CPN* também possui um conjunto de declarações para indicar a natureza dos elementos citados nas diversas inscrições, à semelhança de uma área de declarações de uma linguagem de programação qualquer. (LIMA e PERKUSICH, 2007, p. 328).

Vale ressaltar também que, por razões históricas, no contexto de redes de Petri Coloridas utiliza-se a expressão *conjunto de cores (colour set)* ao invés de tipos de dados e, como consequência disto, cada valor recebe a denominação de *cor (colour)*, que é entendido como um tipo de dado (inteiro, real, lista, entre outros). Assim, cada lugar contido na rede está relacionado a um conjunto de cores, que delimita o tipo de marcação que o lugar pode receber.

## 2.5 PROPRIEDADES DE REDES DE PETRI

### 2.5.1 GRAFOS DIRECIONAIS

De acordo com (JENSEN E KRISTENSEN, 2009), um *grafo direcionado* consiste de um conjunto de *nós*  $N$  e um conjunto de *arcos*  $A$  com identificações de um conjunto  $L$ . O conjunto de arcos é um subconjunto de  $N \times L \times N$  e um elemento  $(n, l, n')$  pertencente a  $A$  que representa um arco que conecta um nó  $n$  a um nó  $n'$  com uma identificação  $l$ . Isto pode ser visto na Figura 10.

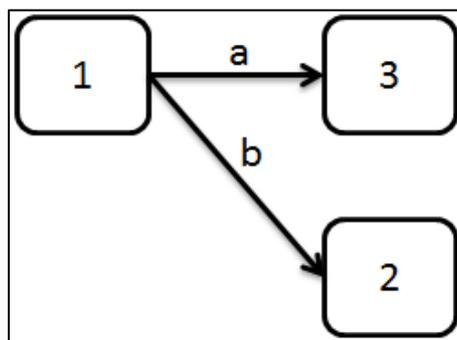


Figura 10. Representação de grafos direcionados.

Para a Figura 10, temos que  $N = \{1, 2, 3\}$ ,  $L = \{a, b\}$  e o conjunto de arcos é definido como  $A = \{(1,a,3), (1,b,2)\}$ . Um grafo direcionado é finito se o número de nós e arcos são finitos. O conjunto de nós que não possui arcos de saída são denominados *nós terminais*.

Formalmente, um grafo direcionado com identificação nos arcos do conjunto  $L$  é uma dupla

$$DG = (N,A), \quad (6)$$

em que:

1.  $N$  é um conjunto de *nós*;
2.  $A \subseteq N \times L \times N$  é um conjunto de *arcos*.

Vale ressaltar que a definição acima permite a existência de arcos paralelos, isto é,  $(n, l, n')$  e  $(n, l', n')$  que conectam o mesmo par de nós.

### 2.5.2 ESPAÇO DE ALCANÇABILIDADE

O espaço de alcançabilidade de uma rede de Petri é um grafo direcionado  $SS$  em que o conjunto de nós  $N_{SS}$  corresponde ao conjunto de marcações alcançáveis  $\mathcal{R}(M_0)$ . O conjunto de arcos  $A_{SS}$  corresponde aos elementos de ligação e existe um elemento de ligação  $(t, b)$  que conecta um nó  $M$  a um nó  $M'$  se e somente se  $(t, b)$  está habilitada no nó  $M$  e a ocorrência de  $(t, b)$  neste nó leva para o nó  $M'$ .

Formalmente, o *espaço de alcançabilidade* de uma rede de Petri colorida é um grafo direcional  $SS = (N_{SS}, A_{SS})$  com identificação nos arcos do conjunto  $BE$ , sendo que:

1.  $N_{SS} = \mathcal{R}(M_0)$  é o conjunto de nós;
2.  $A_{SS} = \{(M, (t, b), M') \in N_{SS} \times BE \times N_{SS} \mid M \xrightarrow{(t,b)} M'\}$  é o conjunto de arcos.

### 2.5.3 ALCANÇABILIDADE

Uma marcação  $M'$  é dita *alcançável* a partir da marcação  $M \in \mathcal{R}(M_0)$  se, e somente se,  $M' \in \mathcal{R}_{SS}(M)$ , isto é, se, e somente se, existe um caminho finito que leva de  $M$  até  $M'$  no estado de espaço. Se for possível encontrar  $M'$  partindo de  $M$ , então, disse que  $M'$  é alcançável a partir de  $M$ ; caso não seja, diz-se que  $M'$  não é alcançável a partir de  $M$ .

### 2.5.4 REVERSIBILIDADE

Uma marcação inicial  $M_{home}$  é chamada de *home marking* se ela pode ser alcançada de qualquer marcação alcançável, ou seja, para todo  $M \in \mathcal{R}(M_0)$  existe um  $M_{home} \in \mathcal{R}(M)$ . É importante ressaltar que para a rede apresentar esta característica, tem de ser *possível* alcançar a *home marking* de qualquer lugar da rede, mas não existe garantia que a mesma será alcançada.

Portanto, formalmente temos que  $\forall M \in \mathcal{R}(M_0) : M_{home} \in \mathcal{R}(M)$ .

### 2.5.5 VIVACIDADE

Esta propriedade está relacionada com a ocorrência de bloqueios em uma rede de Petri. Assim, um *nó morto* (*dead marking*) é um nó que não possui transições habilitadas, isto é,  $t \in T, \neg(M \xrightarrow{t})$ . Vale lembrar que a notação  $M \xrightarrow{t}$  denota que a transição  $t$  está ativa para o nó  $M$ . Caso a transição  $t$  seja uma *transição morta* na marcação inicial  $M_0$ , ela não está habilitada em nenhum nó alcançável  $M$ , ou seja,  $\neg(M \xrightarrow{t})$  para todo  $M \in \mathcal{R}(M_0)$ . Uma transição morta pode ser removida do modelo sem alterar o comportamento do mesmo.

Uma transição é *viva* na marcação inicial  $M_0$  se para todos os nós alcançáveis  $M \in \mathcal{R}(M_0)$  existe uma marcação  $M' \in \mathcal{R}(M)$  na qual  $t$  está ativa. Isto significa que  $t$  sempre poderá ser habilitada. Vale ressaltar que uma transição poder ser não-morta sem ser viva. Dizer que uma marcação é não-morta (*non-dead*) implica que a transição pode ocorrer uma vez, mas que não sempre acontecerá.

Desse modo, formalmente temos que:

1.  $M$  é um *nó morto* se e somente se  $\forall t \in T : \neg(M \xrightarrow{t})$ ;
2.  $t$  é uma transição morta se e somente se  $\forall M \in \mathcal{R}(M_0) : \neg(M \xrightarrow{t})$ ;
3.  $t$  é uma transição viva se e somente se  $\forall M \in \mathcal{R}(M_0) \exists M' \in \mathcal{R}(M) : M' \xrightarrow{t}$ ;

## 2.6 REDES DE PETRI COLORIDAS HIERÁRQUICAS

É comum observar sistemas de produção que contém subsistemas parecidos ou similares. Neste tipo de cenário, o modelo criado em rede de Petri para representação do sistema real apresentará estruturas similares, que podem ser representadas por uma única estrutura de alto nível, o que possibilita representar modelos de grandes dimensões com o uso de estrutura de alto nível menores.

### 2.6.1 MÓDULOS

Cada *módulo* de uma *CPN* hierárquica é uma *CPN* não hierárquica. Desse modo, aquela consiste de um conjunto finito de lugares  $P$ , um conjunto finito de transições  $T$ , um conjunto de arcos  $A$ , um conjunto finito não vazio de *conjunto de cores*  $\Sigma$ , uma *função de cor*  $C$ , um *guarda*  $G$ , uma *função de arco*  $E$  e uma *função de inicialização*  $I$ . Isto quer dizer que cada módulo pode ter seu *conjunto de cores* e declaração de variáveis próprias.

Cada módulo também pode conter um conjunto possivelmente vazio de *substituições de transição* (*substitution transitions*)  $T_{sub} \subseteq T$ . Os *input socket places*  $P_{sock}^{in}(t)$  de uma transição de substituição  $t$  são os conjuntos de lugares de entrada da transição e os *output socket places*  $P_{sock}^{out}(t)$  são os conjuntos de lugares de saída. Por último, os *input/output socket places*  $P_{sock}^{i/o}(t)$  são os conjuntos de lugares de entrada/saída da transição. Os *socket places*  $P_{sock}(t)$  de uma substituição de transição  $t$  é a união das entradas, saídas e entradas/saídas da transição.

Assim, como uma definição formal de módulos, tem que um *módulo* de uma rede de Petri colorida é uma tupla com quatro componentes

$$CPN_M = (CPN, T_{sub}, P_{port}, PT), \quad (4)$$

em que:

1.  $CPN = (P, T, A, \Sigma, N, C, G, E, I)$  é uma rede de Petri não hierárquica;
2.  $T_{sub} \subseteq T$  é um conjunto de transições de substituição;
3.  $P_{port} \subseteq P$  é um conjunto de *port places*;
4.  $PT : P \rightarrow \{IN, OUT, I/O\}$  é uma *função do tipo de port* que associa um tipo de *port* para cada *port place*.

### 2.6.2 DEFINIÇÃO FORMAL

A construção de redes de Petri hierárquicas é feita utilizando-se *transições de substituições* e *lugares de fusão* (*fusion places*), que, de acordo com (LIMA e PERKUSICH, 2007), são estruturas que permitem especificar um conjunto de lugares como funcionalmente um único lugar, ou seja, se uma ficha é removida ou adicionada de um dos lugares, uma ficha idêntica é adicionada ou removida de todos os outros

lugares pertencentes ao conjunto. *Fusion set*, ou *conjunto de fusão*, é a nomenclatura utilizada para denominar um conjunto de *lugares de fusão*.

Portanto, uma rede de Petri hierárquica é uma quadrupla:

$$CPN_H = (S, SM, PS, FS), \quad (5)$$

sendo que:

1.  $S$  é um conjunto finito de *módulos*;
2.  $SM : T_{sub} \rightarrow S$  é uma *função de submódulo* que associa um submódulo para cada substituição de transição;
3.  $PS$  é uma *função de relação port-socket* que realiza uma *relação port-socket*  $PS(t) \subseteq P_{sock}(t) \times P_{port}^{SM(t)}$  para cada transição de substituição  $t$ ;
4.  $FS \subseteq 2^P$  é um conjunto não vazio de conjuntos de fusão.

As propriedades discutidas na Seção 2.5 também podem ser generalizadas para redes de Petri Coloridas hierárquicas, trocando-se lugares por lugares de fusão e transições por transições de substituição.

## 2.7 REDES DE PETRI TEMPORIZADAS

Na teoria inicialmente apresentada de rede de Petri, a utilização do conceito de tempo não foi contemplada. A introdução deste conceito permite avaliar quão eficiente é um sistema e suas operações, bem como modelar e validar sistemas em tempo real, em que o funcionamento correto do sistema depende do encadeamento das atividades. Além disso, utilizar uma *CPN* temporizada possibilita avaliar e medir a performance do sistema.

Uma rede temporizada pode ser determinística ou não-determinística, a depender se o atraso é conhecido ou probabilístico. Ela também pode ser chamada de T-temporizada ou P-temporizada, isto é, se o atraso está associado a uma transição ou a um lugar.

Formalmente, é possível definir uma rede de Petri P-temporizada não determinística como uma dupla  $(PN, \delta)$  (LIMA e PERKUSICH, 2007), em que:

1.  $PN$  é uma rede de Petri marcada  $(P, T, F, W, M_0)$ ;

2.  $\delta : P \rightarrow [0, +\infty)$  é a função atraso.  $\delta(p_i) = d_i$ , sendo que  $d_i \in [0, +\infty)$  é o atraso de tempo associado ao lugar  $p_i \in P$  da rede.

Desse modo, se considerarmos uma rede de Petri com  $n$  e  $m$  lugares, podemos denominar o vetor atraso como  $V_d = (d_1, d_2, \dots, d_i, d_m)^T$ , em que cada componente  $d_i$  é igual a  $\delta(p_i)$ . Assim, podemos ter representações e modelos como: *um operador  $O_1$  realiza a atividade  $X$  por  $d_i$  unidades de tempo*. Para o funcionamento correto de uma rede de Petri temporizada, temos algumas regras:

1. Toda marcação depositada em um lugar  $p_i$  fica indisponível por um certo tempo  $\tau \geq \delta(p_i)$ ;
2. A ocorrência de atrasos em transições é considerada nula, seguindo as regras de redes de Petri não-temporizadas.

## 2.8 REDES DE PETRI NO CONTEXTO DE *WORKFLOW*

A representação de redes de Petri tem se mostrado bastante útil, já que permite uma fácil interpretação e entendimento, além de possuir uma base matemática sólida, o que gera segurança no seu uso. Segundo SALIMIFARD & WRIGHT (2001), na modelagem de workflow utilizando redes de Petri, cada tarefa é representada por uma transição correspondente. Lugares representam as pré e pós-condições ou ainda, os recursos requeridos para a execução de uma determinada tarefa. Os arcos representam as relações lógicas entre as tarefas e o próprio fluxo de trabalho.

Desde que a teoria de redes de Petri foi utilizada pela primeira vez para modelar fluxo de atividades (*workflow*) por ZISMAN (1997), vários autores publicaram trabalhos em que se buscava essa interação entre os dois assuntos. Entre eles, pode-se citar: CINDIO et al. (1988), LI et al. (1993), FERSCHA (1994), MERZ et al. (1995), SCHÖMIG & RAU (1995), ELLIS & NUTT (1996), NUTT (1996), WIKARSKI (1996), OBERWEIS et al. (1997), AALST (1998), ADAM et al. (1998), BADOUEL & OLIVER (1998).

De acordo com HOLT (2000), as redes de Petri permitem criar planos de sistemas caracterizados por um nível de concorrência. Na estrutura de *workflow*, tais planos são esboços globais ou parciais da forma específica de realizar alguns tipos de

atividades complexas e distribuídas. Desde que todo o processo possa ser considerado como um sistema que envolve operações complexas e concorrentes, o uso das redes de Petri em seu planejamento ou análise parece uma ideia natural.

AALST & HEE (2002) identificam três razões principais para aplicar redes de Petri na modelagem de *workflow*:

1. As redes de Petri possuem semântica formal além da sua natureza gráfica;
2. Elas podem modelar explicitamente os estados do sistema e;
3. Existem variedade e disponibilidade de técnicas de análise.

A principal força de redes de Petri na modelagem de *workflow* é resultante da combinação de: fundamentação matemática, representação gráfica compreensiva e, possibilidade de executar simulações e verificações (MERZ et al.,1995).

## 2.9 JUSTIFICATIVA PARA ESTUDO DO PROBLEMA

Nas últimas décadas, o estudo e utilização de técnicas de gerenciamento se tornaram cada vez mais difundido entre as organizações. De acordo com *PROJECT MANAGEMENT INSTITUTE* (2008), o aumento na aceitação do gerenciamento de projetos indica que o uso de conhecimentos, processos, habilidades, ferramentas e técnicas adequadas pode ter um impacto significativo no sucesso de um projeto.

Com o crescente aumento da complexidade das grandes empresas e suas operações, gerenciar e melhor controlar seus processos e ativos deixa de ser uma necessidade e passa a ser um requisito básico para o sucesso e bom desempenho da instituição, além de influenciar no produto e/ou serviço final que será entregue.

Segundo KERZNER (2006), são diversos os fatores que fazem com que as empresas adotem o gerenciamento de projetos em seus negócios, como:

- ✓ Competição;
- ✓ Padrões de qualidade;
- ✓ Resultados financeiros;
- ✓ Fatores tecnológicos;
- ✓ Aspectos sociais;
- ✓ Fatores políticos.

Contudo, ainda que tenha acontecido um evidente aumento, por parte das empresas, na utilização do gerenciamento de projetos, é possível observar altos índices de falhas em projetos (FORMAN *apud* RAD e RAGHAVAN, 2000).

De acordo com VARGAS (2009), grande parte dos erros deriva de falhas gerenciais que podem ser prevenidas, como:

- ✓ Metas e objetivos mal estabelecidos;
- ✓ Pouca compreensão da complexidade do projeto;
- ✓ Estimativas financeiras pobres e incompletas;
- ✓ Sistema de controle inadequado.

Diante do exposto, torna-se importante um estudo e análise que indique pontos a serem melhorados no processo de controle e de execução das atividades relacionadas à instalação de novas linhas robotizadas.

## 3 METODOLOGIA

Neste capítulo apresenta-se a descrição do problema e os conceitos envolvidos na criação do modelo, bem como a descrição de suas várias partes, funções e funcionamento.

### 3.1 DESCRIÇÃO DO PROBLEMA

A Comau do Brasil tem como carro-chefe seus robôs industriais. Dessa forma, quando se deseja realizar a instalação de soluções robotizadas, várias etapas são necessárias. Dentre estas etapas tem-se a preparação dos robôs para instalação na linha de produção. Esta preparação é realizada por uma equipe, que é composta por mecânicos, robotistas e analistas, e nada mais é do que a realização de diversas atividades para que os robôs fiquem prontos para funcionarem.

A execução destas atividades é o foco deste estudo, pois a forma como elas serão desempenhadas e relacionadas uma com as outras terá efeito direto no resultado final obtido, podendo-se ter atrasos, gargalos e até mesmo bloqueios, parando temporariamente o projeto. Assim, entender o processo como um todo, bem como suas particularidades, é essencial para um planejamento adequado, sabendo como melhor responder a variações de demanda, formação da equipe de trabalho e quais atividades merecem maior atenção.

Ao detalhar criteriosamente o que se faz necessário e o que normalmente é feito nos robôs para instalação, chega-se a cerca de cem atividades que são realizadas, dependendo do tipo de aplicação. Contudo para o estudo aqui feito, duas ações foram tomadas: primeiro, foi feita a agregação e agrupamento de atividades, facilitando a análise e realização do estudo, mas sem perder informações relevantes ao processo; segundo, focou-se nas atividades iniciais, ou seja, nas atividades executadas nos robôs até o momento em que o mesmo já pode ser transportado para o local em que ele funcionará.

As atividades que serão avaliadas estão listadas abaixo:

- ✓ movimentação e desembalagem dos robôs;

- ✓ inspeção e *check list* do estado do controlador e do braço;
- ✓ controle de correta instalação braço/controlador;
- ✓ alimentação do controlador;
- ✓ energização do robô;
- ✓ controle do sistema operacional do robô;
- ✓ controle das posições de calibração do robô;
- ✓ geração dos aplicativos a serem instalados;
- ✓ instalação dos aplicativos;
- ✓ teste de movimentação dos robôs;
- ✓ configuração dos aplicativos;
- ✓ desligamento do robô;
- ✓ reembalagem do robô;
- ✓ documentação interna para a saída do robô da sede da Comau do Brasil.

## 3.2 PRESSUPOSTOS

Para construção do modelo, foram utilizadas as seguintes considerações:

1. Não foi considerado na criação do modelo o tempo dispendido nas reuniões de definições do projeto e nem reuniões com o cliente;
2. O tempo de fabricação, tempo de envio para o Brasil e desembaraço na Receita Federal não foram considerados, já que a Comau do Brasil não tem controle sobre estas atividades;
3. O tempo de execução de cada atividade é definido pela função que a executará. Portanto, funções diferentes podem apresentar tempos de execução diferentes;
4. Dependendo da atividade a ser executada, a possível falta temporária de recursos físicos, como local para armazenamento do robô ou ferramenta para execução da tarefa, foi representada por atraso;
5. O modelo contempla as atividades realizadas no robô preparando-o para instalação no local de funcionamento, também conhecidas como atividades “*in house*”;

6. Não foi contemplada a possibilidade de parar a execução de uma atividade para alocar o recurso utilizado em outra operação, devolvendo o recurso e continuando a execução daquela posteriormente;
7. Foi considerado para criação do modelo que as atividades iniciadas sempre seriam finalizadas, podendo ou não ocorrer atraso. Dessa forma, não foi modelado a possibilidade de reiniciar a execução de uma atividade.

### 3.3 ELABORAÇÃO DO MODELO

Para a elaboração do modelo, teve-se a ideia de se utilizar a teoria de Redes Petri. Dentre as vantagens de se utilizar modelos de Redes Petri, estão (PALOMINO, 1995):

- ✓ Ferramenta de modelagem hierárquica com teoria matemática bem definida e fundamento prático;
- ✓ Conflitos podem ser modelados de maneira simples e eficiente;
- ✓ Representação de sistemas, utilizando uma abordagem *top-down*, bem como *bottom-up*, em diferentes níveis de abstração, o que facilita sua decomposição em subsistemas funcionais, com clara relação entre os subsistemas;
- ✓ Métodos de análises definidos e fortes, no que se refere à quantidade de informações fornecidas;
- ✓ Possibilidade de decomposição de sistemas, com simplicidade e fácil compreensão, além de poder representar sincronização e concorrência entre tarefas, o que implica em flexibilidade;
- ✓ Unidades de recursos físicos podem ser representadas de maneira apropriada.

Tomou-se como referência o artigo “*Analysis of Resource-Constrained Processes with Colored Petri Nets*” (NETJES et al., 2005). Contudo, comparando o modelo descrito no artigo citado e o modelo criado para este trabalho, foi utilizado um número maior de bloco de atividades, além de ter sido considerada a existência de paralelismo durante a execução do processo e ocorrência de falhas.

Nas seções seguintes serão apresentadas as partes que compõem o modelo criado.

### 3.3.1 GERADOR

Para simular a chegada de vários robôs, foi criado um gerador que produz uma quantidade de fichas variável, dependendo da análise desejada. A transição de substituição do gerador e sua subpágina são mostradas nas Figura 11 e Figura 12.

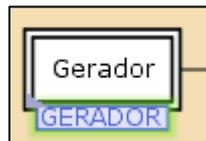


Figura 11. Gerador de fichas.

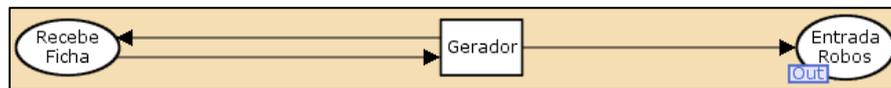


Figura 12. Subpágina do gerador.

Foi utilizada, para gerar fichas, uma distribuição exponencial, com frequência de chegada de doze robôs por hora. Para limitar a quantidade de *tokens* que são inseridos na rede, ou seja, para limitar o número de robô que serão trabalhados é utilizada uma expressão booleana na transição *Gerador*, mostrada na Figura 12. Esta expressão, também chamada de *guarda* (*guard*, em inglês), verifica se uma condição é verdadeira ou não. Caso seja verdadeira, a transição é habilitada e pode ser disparada.

### 3.3.2 BLOCO DE ATIVIDADE

Fazendo uso da hierarquia presente em Redes Petri, foi criada uma transição de substituição (*substitution transition*), que pode ser vista na Figura 13, para representação do bloco de atividade. Esta leva a uma subpágina (Figura 14), que detalha seu funcionamento, contendo os campos com as informações necessárias para diferenciação entre atividades.



Figura 13. Bloco de atividade.

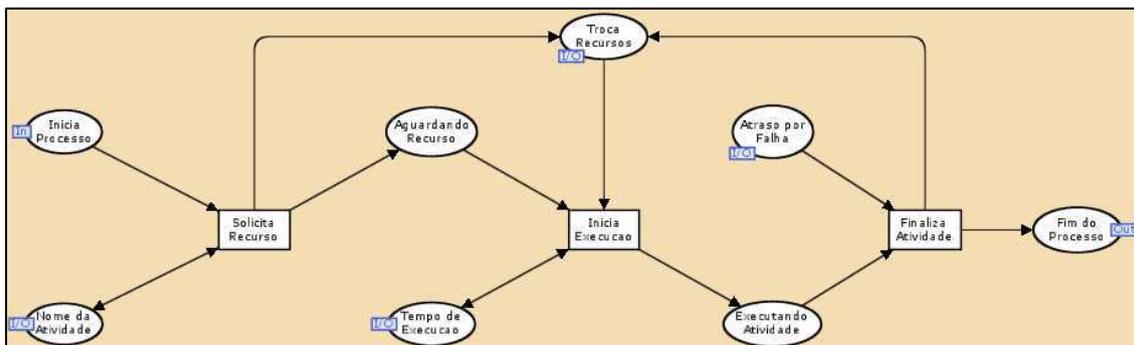


Figura 14. Subpágina do bloco de atividade.

O funcionamento do bloco de atividade ocorre da seguinte forma: no lugar *Nome da Atividade*, a marcação indica qual atividade será realizada. De forma semelhante, no lugar *Tempo de Execução* é indicado quanto tempo a realização da tarefa consumirá. O lugar *Inicia Processo* recebe a ficha para que o desenvolvimento da atividade se inicie. Quando a transição *Solicita Recurso* é habilitada, é enviado para o lugar *Troca Recursos* o pedido do recurso que executará a atividade em questão. Enquanto é feita a verificação da disponibilidade deste, a ficha vai para o lugar *Aguardando Recurso*, que, como o nome sugere, indica que está se esperando o recurso para dar continuidade. Quando o recurso é disponibilizado, a transição *Inicia Execução* é habilitada e a marcação vai para o campo *Executando Atividade*. O campo *Atraso por Falha* indica se ocorreu algum problema na execução da tarefa, o que acarretará em atraso. Ao fim da execução, a transição *Finaliza Atividade* é disparada e o recurso que estava sendo utilizado na realização da tarefa é devolvido ao *Bloco de Recursos*, indicando que ele não mais está sendo utilizado e a ficha vai para o lugar *Fim do Processo*, permitindo que a próxima atividade possa ser iniciada.

A maneira como este bloco foi construído permite sua reutilização, em função de as configurações básicas serem comuns. Assim, em virtude desta característica e de como a rede foi idealizada, tal bloco foi replicado quatorze vezes.

### 3.3.3 BLOCO DE RECURSO

De modo semelhante ao *Bloco de Atividade*, foi desenvolvida uma transição de substituição, que pode ser vista na Figura 15, para o *Bloco de Recurso*. Aquela leva a subpágina mostrada na Figura 16. O bloco de recursos não é variável de acordo com certa atividade e nele estão contidos todos os recursos disponíveis para realização das atividades. Sua função é verificar a disponibilidade de recursos, disponibilizar estes e gerenciar os pedidos.



Figura 15. Bloco de recursos.

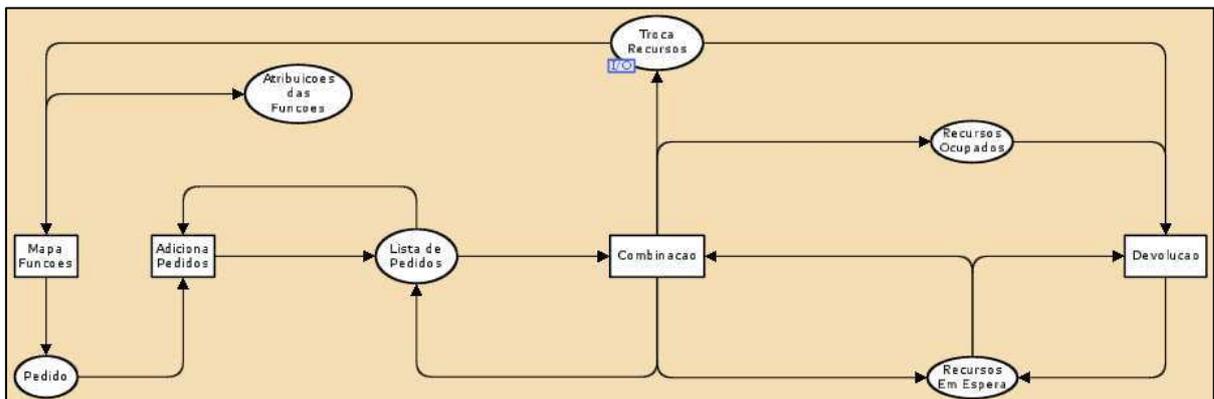


Figura 16. Subpágina do bloco de recursos.

O bloco de recurso funciona da seguinte forma: o lugar *Troca Recursos* faz a interface entre o Bloco de Recursos e os Blocos de Atividades do modelo. Dessa forma, todos os pedidos de recurso, bem como a disponibilização e devolução dos mesmos é feito por este bloco. No campo *Atribuição das Funções* está contido a associação entre as atividades e as funções que podem executá-las. Assim, quando uma ficha chega no lugar *Troca Recursos*, a transição *Mapa Funcoes* é habilitada e um *token* é colocado no lugar *Pedido* contendo as informações do atividade – *id* e *t* –, bem como as pessoas que podem executá-la. Em seguida, a transição *Adiciona Pedidos* é habilitada, colocando o pedido que chegou no final de uma lista, que fica salva no lugar *Lista de Pedidos*. Quando a transição *Combinacao* é habilitada, ela verifica se alguma função contida no lugar *Recursos Em Espera* pode executar alguma tarefa que está aguardando recurso. Quando a combinação é feita, o recurso é disponibilizado para execução, sendo retirado

do *Recursos Em Espera* e colocado em *Recursos Ocupados*. Ao finalizar a execução da tarefa, a transição *Devolucao* é habilitada, retirando o recurso do lugar *Recursos Ocupados* e colocando em *Recursos Em Espera*, o que significa dizer que a pessoa está disponível para ser utilizada para realizar outra atividade.

### 3.3.4 BLOCO DE FALHA

O Bloco de Falha indica se ocorreu falha no desenvolvimento da atividade ou não. Para modelar a ocorrência de falha foi utilizada uma distribuição de Bernoulli. Esta distribuição pode ter diferentes probabilidades de sucesso para diferentes atividades. Desse modo, a chance de ocorrer uma falha durante a execução varia e esta possibilidade é representada na função *falha0X*, em que *X* identifica a atividade para qual se está calculando a possibilidade de falha e, conseqüentemente, atraso.

O intervalo do atraso gerado também pode variar para cada atividade. Assim, o  $[atraso_{min}, atraso_{max}]$  pode ser diferente para cada atividade. Este intervalo é modelado utilizando uma distribuição uniforme discreta, ou seja, dentro do intervalo especificado, a probabilidade de ocorrência para um número é a mesma para qualquer outro número. A chance de falha bem como o tempo de atraso gerado foi baseada em dados obtidos durante a experiência de estágio.

Na Figura 17 temos a representação da transição de substituição do bloco de falha e na Figura 18, a subpágina do mesmo.



Figura 17. Bloco de falha.

O bloco de falha se interliga com o bloco de atividade pelo lugar *Atraso01*, que tem uma relação de *port/socket* com o lugar (em vermelho) de mesmo nome mostrado na Figura 20. Este se relaciona com o lugar *Atraso por Falha* da Figura 14. Já as fichas colocadas no lugar *Entrada Falha01* vem do lugar *Verificacao de Falha da Atividade01* visualizado na Figura 20, pois estes também possuem um relação de *port/socket*. Ainda vale ressaltar que o *token* é colocado no lugar *Verificacao de Falha da Atividade01* quando ocorre o disparo da transição *Kick Off*.

No caso de a possibilidade de ocorrência de falha de certa atividade ser baixa, tornando dispensável a utilização do bloco de falha para aquela atividade, o disparo da

transição *Kick Off* (Figura 20) coloca uma ficha diretamente no lugar *Atraso por Falha* (Figura 14).

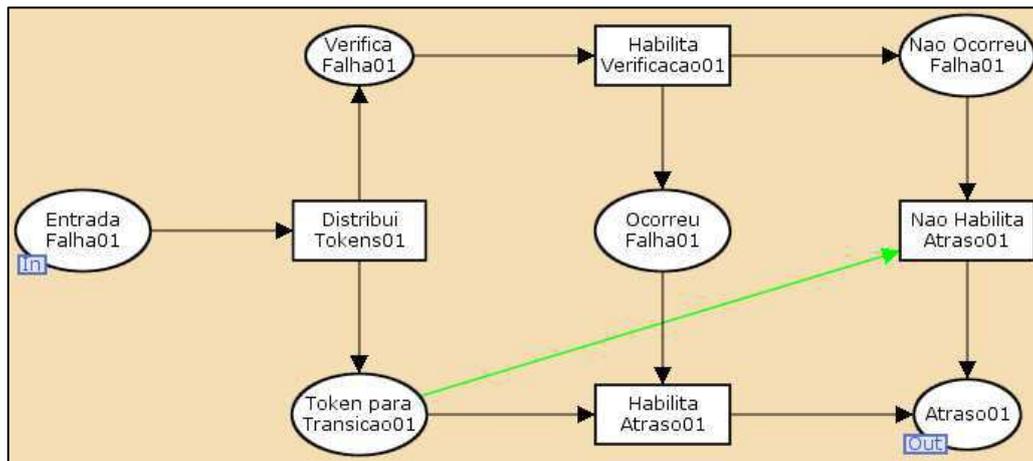


Figura 18. Subpágina do bloco de falha.

O funcionamento deste bloco ocorre da seguinte maneira: ao receber a ficha no campo *Entrada Falha01*, a transição *Distribui Tokens01* é habilitada, colocando *tokens* nos lugares *Token para Transicao01* e *Verifica Falha01*. No arco que liga a transição *Distribui Tokens01* ao lugar *Verifica Falha01*, existe uma expressão de arco (*arc expression*) que habilita a execução da função *falha01*. Esta função indica se ocorreu ou não algum problema na execução da atividade, o que gerará um atraso. Caso tenha ocorrido falha na execução, a sequência de funcionamento a partir deste ponto é a seguinte: transição *Habilita Verificacao01* habilitada, uma ficha é colocada no lugar *Ocorreu Falha01*, a transição *Habilita Atraso01* dispara e uma ficha é colocada no lugar *Atraso01*. Caso não ocorra falha na execução, a sequência de disparo é: *Habilita Verificacao01*, *Nao Ocorreu Falha01*, *Nao Habilita Atraso01* e *Atraso01*.

Assim como o bloco de atividade, a construção aqui feita também é reutilizável.

### 3.3.5 CONCEITO GERAL

Em virtude da rede ser grande, o que dificulta a visualização completa aqui, bem como os detalhes de suas ligações, serão mostradas partes de redes ou das ligações dos blocos, objetivando um melhor entendimento das ligações e funcionamento.

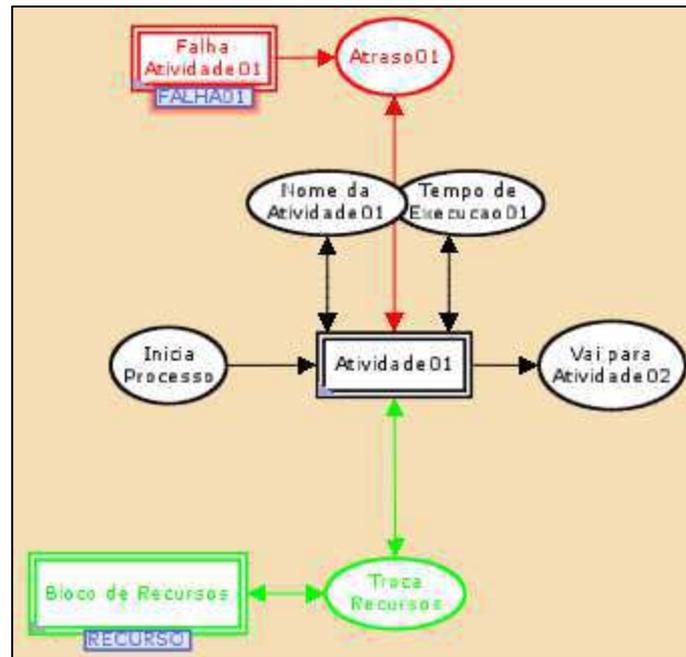


Figura 19. Ligação entre Bloco de atividade e bloco de recurso.

É possível visualizar na Figura 19 as ligações contidas entre o Bloco de Atividade e o Bloco de Recursos. Em caso do aumento de atividades que devem ser representadas, o Bloco de Recursos é replicado de acordo com a necessidade – no caso deste trabalho, será utilizado quatorze vezes tal bloco. Contudo, a ligação para o lugar *Troca Recursos* será sempre a mesma, ou seja, não é necessário replicar este lugar ou o *Bloco de Recursos*.

Na Figura 20 é possível visualizar a interligação entre alguns *Blocos de Atividade*, bem como a relação estabelecida com o *Bloco de Recursos*. Também é possível visualizar o conceito de paralelismo que pretende ser desenvolvido.

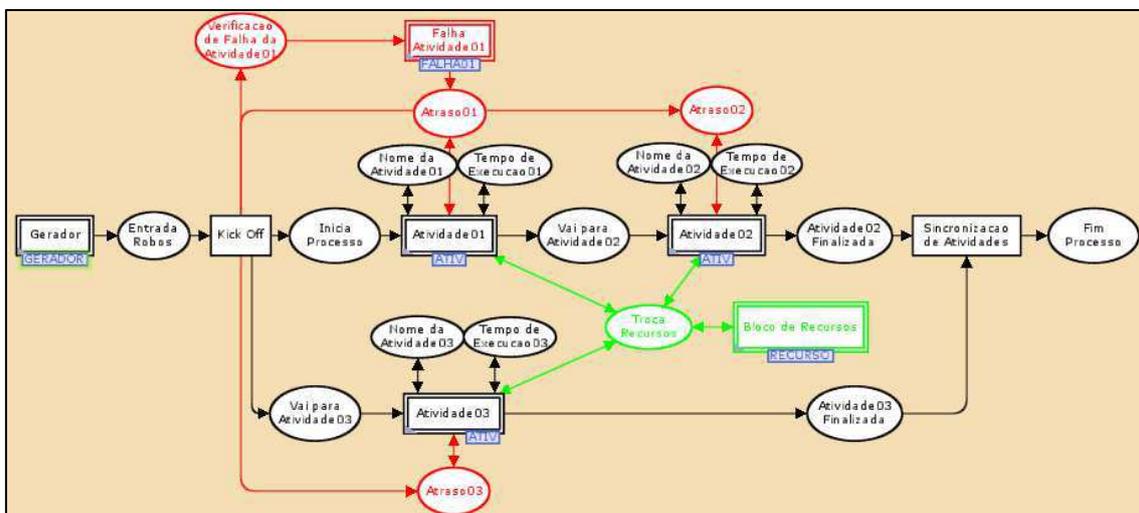


Figura 20. Parte da rede criada para representação do processo.

Veja que podemos ter dois tipos de paralelismo na rede durante a execução das atividades. O primeiro é referente a execução concorrente de diferentes atividades em robôs diversos, ou seja, enquanto em um robô R01 pode-se estar executando a segunda atividade, no robô R02 pode estar executando a primeira. Este tipo de paralelismo é possibilitado pelo compartilhamento de recursos. Já o segundo tipo de paralelismo possível é aquele intrínseco a rede, exemplificado pela *Atividade03* na Figura 20. Neste caso, o paralelismo é uma característica da rede, apesar de somente ocorrer caso o compartilhamento de recursos permita.

Portanto, neste último cenário, poderíamos ter a segunda e terceira atividades sendo executadas no robô R01 e a primeira sendo realizada no robô R02. Conceitualmente, o paralelismo intrínseco é obtido quando o disparo de uma transição coloca fichas na entrada de duas atividades, permitindo que ambas possam ser executadas. Já o fim do paralelismo ocorre com a sincronização de atividades, feito conectando os lugares de saída das atividades em questão a uma transição. Desta forma, a transição somente será habilitada quando as duas atividades forem finalizadas.

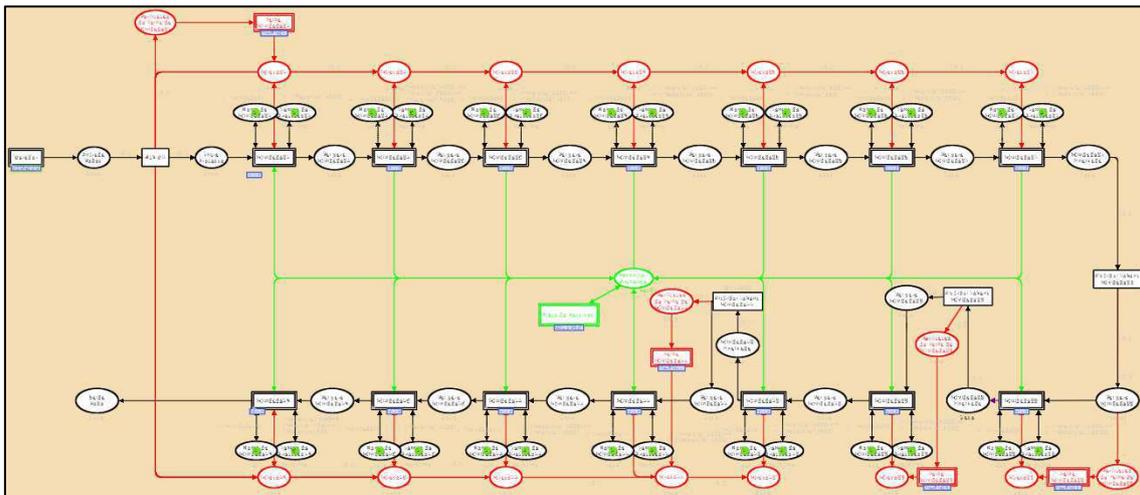


Figura 21. Visualização da rede completa sem paralelismo.

Na Figura 21 é possível ver a rede completa sem o conceito de paralelismo intrínseco, ou seja, as atividades são desenvolvidas sequencialmente para cada robô. Já na Figura 22 tem-se o conceito da rede com paralelismo intrínseco.

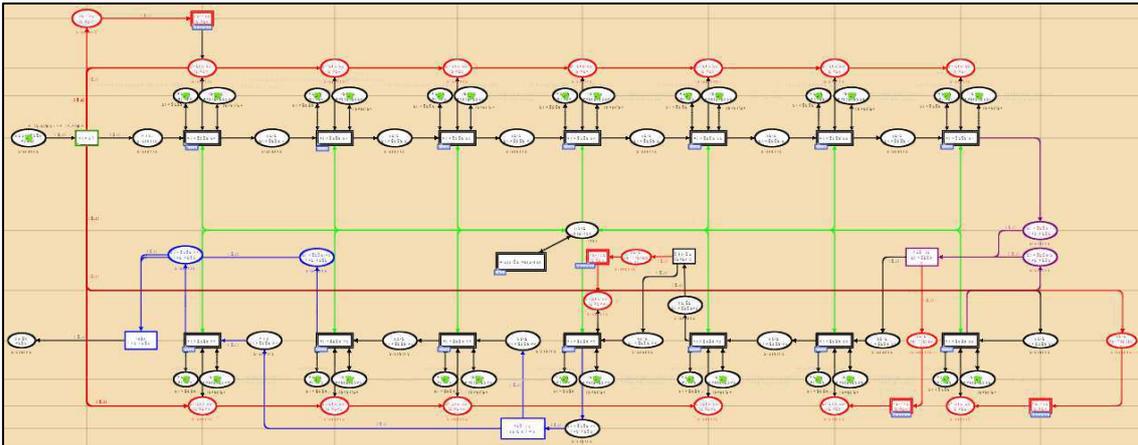


Figura 22. Rede com desenvolvimento de atividades em paralelo criada no *CPN Tools*.

Em virtude do tamanho das redes apresentadas nas duas figuras anteriores, a clareza dos detalhes e informações contidas é reduzida. Contudo, como foi feita a identificação prévia dos blocos, seu funcionamento pode ser entendido sem maiores dificuldades.

### 3.4 *CPN TOOLS*

O *software* utilizado no trabalho para o desenvolvimento do modelo e avaliação do mesmo foi o *CPN Tools*, que é uma ferramenta aberta que pode ser obtido no endereço: <http://cpntools.org/download>.

Esta ferramenta facilita a criação de redes de Petri Coloridas, bem como a verificação e simulação da rede criada. Na Figura 23 é possível visualizar a tela do *software*, bem como os *palettes* existentes.

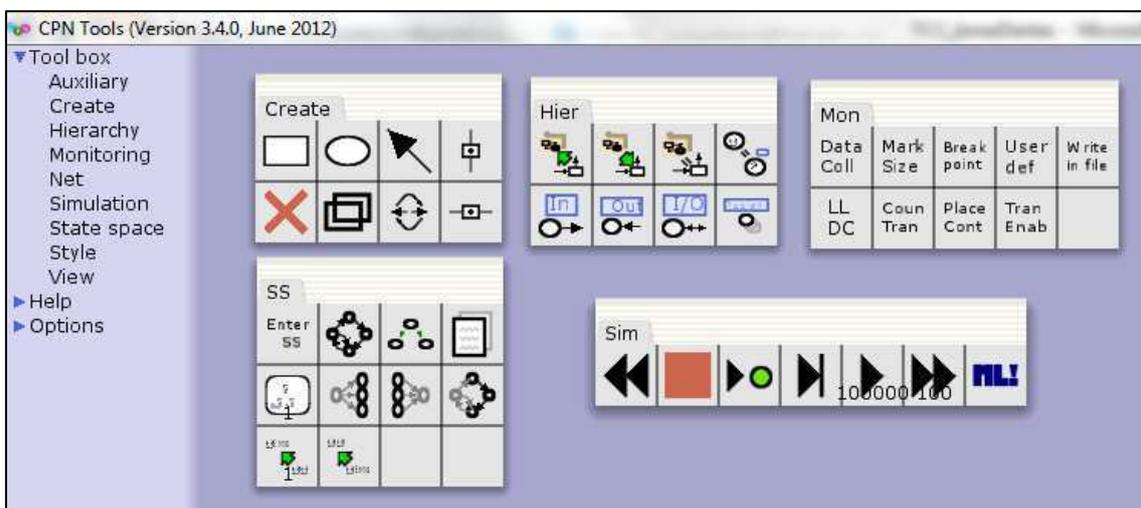


Figura 23. Tela da ferramenta *CPN Tools*.

Cada *palette* pode ser entendido como um pequeno conjunto de ferramentas contidas no programa para facilitar o uso e criação de redes, bem como o teste de funcionamento e de propriedades.

O *palette Create* contém elementos para facilitar a criação de uma rede. Assim, basta selecionar o que se deseja adicionar: uma transição – representada pelo retângulo – ou um lugar – representado pelo círculo. A seta é utilizada para criação de arcos. O *x* vermelho é utilizado quando se deseja excluir um elemento da rede. Os dois retângulos sobrepostos servem para copiar um elemento da rede e a seta dupla serve para mudar o sentido do arco.

A aba *Hierarchy* é disponibilizada para auxiliar na criação de redes de Petri hierárquicas. Já a aba *Monitoring* permite a criação de “monitores”, que servem para analisar o funcionamento da rede, obtendo dados e informações como o número de vezes que certa transição foi disparada ou quantas fichas foram depositadas em um lugar da rede. O *palette State Space* permite verificar o estado de espaços, bem como criar o grafo direcionado e as características de um nó desejado. Além disso, ele também tem a opção de gerar um relatório contendo informações sobre as propriedades da rede de Petri. Por último, a aba *Simulation* contém elementos para realizar simulações na rede criada.

Informações mais detalhadas sobre cada *palette* pode ser obtido no endereço eletrônico: <http://cpntools.org/documentation/gui/palettes/start>. Além disso, outras informações acerca da ferramenta e suas funcionalidades podem ser obtidas no site: <http://cpntools.org>.

## 4 RESULTADOS

Neste capítulo, apresentam-se os resultados obtidos e as observações realizadas. Vale ressaltar que todos os dados apresentados foram gerados pela ferramenta utilizando um intervalo de confiança de 95%.

### 4.1 EVOLUÇÃO DO MODELO

Na Figura 21 é possível visualizar o modelo da rede final utilizado nas simulações que geraram as estatísticas aqui apresentadas. Outros tipos de topologia de rede foram considerados, como apresentadas no Capítulo 3. Contudo, em virtude das limitações da ferramenta utilizada, as outras versões foram desconsideradas. Um dos principais motivos para se descartar o uso das outras topologias foi o fato de que não foi possível utilizar o *CPN Tools* para verificar as propriedades da rede construída, garantindo, portanto, que o modelo criado funcionava de acordo com o esperado e respeitava os conceitos de rede de Petri citados no Capítulo 2 deste trabalho.

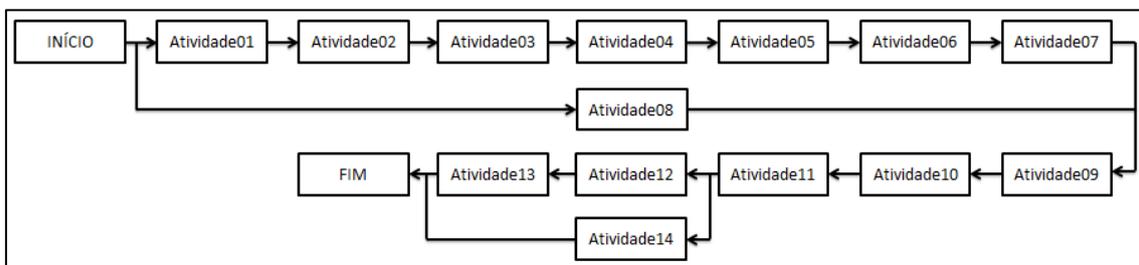


Figura 24. Uma das topologias de rede pensadas para simulação.

Pode-se ver na Figura 24 que a rede apresentava paralelismo intrínseco, com as atividades oito e quatorze. Contudo, quando a rede foi criada com esta característica, não foi possível, utilizando o *CPN Tools*, criar a árvore de alcançabilidade e verificar outras propriedades da rede.

Outra característica que foi pensada para a rede foi a inclusão da possibilidade de falha. Contudo, em virtude do demasiado esforço computacional para iniciar a ferramenta com todas as atividades apresentando possibilidade de erro na execução, o bloco de falha foi colocado somente em nas atividades consideradas mais importantes para uma representação mais próxima da realidade.

Portanto, em função das limitações do *software* utilizado, o modelo foi simplificado para melhor realização de simulações e testes. Esta simplificação, entretanto, não implicou em distorções do modelo com relação a realidade. Na Figura 22 é possível ver o modelo criado para representação do conceito representado na Figura 24, em que haveria a possibilidade de paralelismo intrínseco ao modelo.

Em razão dos motivos apresentados nesta seção, chegou-se no conceito do modelo visualizado na Figura 25, sem atividades em paralelo intrínsecas ao modelo. Esta concepção deu origem a rede mostrada na Figura 21 e, utilizando a ferramenta descrita na Seção 3.4, foi possível verificar suas propriedades e características.

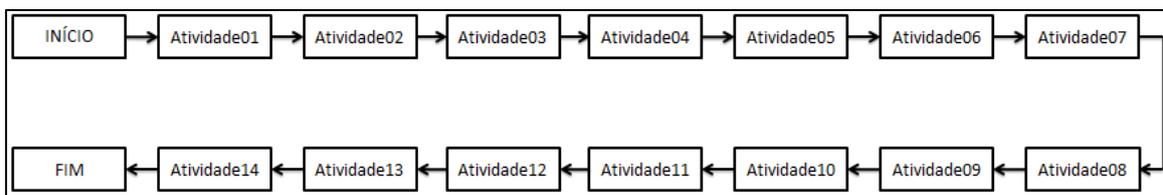


Figura 25. Conceito final da topologia da rede.

## 4.2 CARACTERÍSTICAS PARA SIMULAÇÃO

Após a realização de diversas simulações e verificação de cenários variados, percebeu-se que certas escolhas de tamanho de equipe e distribuição de recurso, ou seja, quais funções apresentariam mais ou menos componentes, resultavam em bloqueios durante a execução das atividades ou em características indesejadas, como alto tempo de execução e/ou baixa taxa de utilização dos recursos.

Por exemplo, foi verificado que a equipe não pode ser formada somente por mecânicos nem que ela não pode conter nenhum mecânico, pois ambos os cenários levará a um bloqueio. Outra observação feita foi que equipe muito pequena, com três ou quatro integrantes, apresentaram um tempo elevado para realização das atividades, indicando que ela estaria atuando sobrecarregada. Já equipe grandes, acima de oito componentes apresentam um tempo de execução satisfatório, mas uma taxa de utilização baixa – 60% ou inferior. Ainda foi observado que a utilização de mais de um mecânico somente se justifica para quantidade de robôs elevadas – 40 ou mais robôs chegando num curto intervalo de tempo –, o que não condiz com a realidade encontrada na empresa.

Desse modo, para verificar as propriedades da rede, bem como gerar as estatísticas para análise, foram utilizadas oito configurações diferentes para a distribuição de pessoas/funções. As distribuições utilizadas foram:

- ✓ Configuração 01: cinco pessoas, sendo um mecânico, dois robotistas e dois analistas;
- ✓ Configuração 02: cinco pessoas, sendo um mecânico, três robotistas e um analista;
- ✓ Configuração 03: cinco pessoas, sendo um mecânico, um robotistas e três analistas;
- ✓ Configuração 04: sete pessoas, sendo dois mecânicos, três robotistas e dois analistas;
- ✓ Configuração 05: sete pessoas, sendo um mecânico, três robotistas e três analistas;
- ✓ Configuração 06: sete pessoas, sendo dois mecânicos, quatro robotistas e um analista;
- ✓ Configuração 07: sete pessoas, sendo um mecânico, cinco robotistas e um analista;

Utilizando estas configurações, foram feitas simulações com o gerador produzindo dez fichas, vinte fichas e trinta fichas, pois são quantidades mais próximas e mais comuns de acontecerem no dia-a-dia da empresa. As estatísticas principais observadas foram o intervalo médio de execução e a taxa média de utilização dos recursos. Para cada configuração inicial, o modelo foi rodado cinquenta vezes e os resultados destas cinquenta simulações foram utilizados para se obter os dados que foram analisados.

Também vale destacar que no modelo utilizado nas simulações que geraram os dados para as análises feitas nas Seções 4.3, 4.4 e 4.5 foi utilizado o bloco de falha. Na Seção 4.6 é feita a discussão dos resultados com dados gerados sem a utilização do mesmo.

### 4.3 EQUIPE COM CINCO PESSOAS

Para as situações em que foram utilizadas dez fichas e cinco pessoas no total, mas variando a forma de distribuição em termos de função, chegou-se aos seguintes resultados:

*Tabela 2. Dados das simulações com cinco pessoas e dez fichas.*

Combinação de Recursos	Taxa de Utilização dos Recursos	Tempo de Execução	
		Mínimo	Máximo
Um mecânico, dois robotistas e dois analistas	73%	55969	65181
Um mecânico, três robotistas e um analista	73%	57129	64560
Um mecânico, um robotista e três analistas	73%	54738	62659

Observando os dados da Tabela 2, é possível visualizar que a taxa de utilização de recursos para os três casos é a mesma, contudo uma variação no tempo de execução acontece. O menor tempo mínimo médio e o menor tempo máximo médio foram obtidos com a execução do modelo com a última configuração de recursos mostrados na Tabela 2. Já o tempo médio de execução para a configuração 03 é de 58990.

É possível ver também que a configuração apresentada na segunda linha da Tabela 2 apresenta a menor variação entre seu intervalo mínimo e máximo. Já a primeira variação apresenta a maior diferença entre os intervalos mínimo e máximo. Isto é maléfico no sentido que quem fizer o cronograma de execução do projeto terá um menor grau de precisão para estimar o fim das atividades.

*Tabela 3. Dados das simulações com cinco pessoas e vinte fichas.*

Combinação de Recursos	Taxa de Utilização dos Recursos	Tempo de Execução	
		Mínimo	Máximo
Um mecânico, dois robotistas e dois analistas	82%	100312	110297
Um mecânico, três robotistas e um analista	82%	103679	113820
Um mecânico, um robotista e três analistas	81%	99115	109473

Tabela 4. Dados das simulações com cinco pessoas e trinta fichas.

Combinação de Recursos	Taxa de Utilização dos Recursos	Tempo de Execução	
		Mínimo	Máximo
Um mecânico, dois robotistas e dois analistas	83%	149622	158744
Um mecânico, três robotistas e um analista	83%	153006	161800
Um mecânico, um robotista e três analistas	83%	145612	154540

Para as Tabela 3 e Tabela 4, a análise feita para a Tabela 2 pode ser repetida. Assim, considerando somente o tempo de execução como fator de escolha, já que a taxa de utilização para as três configurações é a mesma, a melhor configuração para execução das atividades seria ter na equipe um mecânico, um robotista e três analistas.

Entretanto, se levado em conta o fator financeiro, o custo da hora de um analista é maior do que o custo da hora de um robotista. Por exemplo, levando em conta o tempo mínimo de execução da Tabela 2 e subtraindo o tempo gasto pelas atividades que somente o mecânico pode realizar, se dividirmos igualmente o tempo restante por quatro e considerarmos que o custo da hora de um analista é  $CH_A$  e de um robotista é  $CH_R$  e que  $CH_A$  é 50% maior que  $CH_R$ , o custo total para realização das atividades será 20% maior para a terceira configuração do que para a segunda. Utilizando este mesmo raciocínio, a primeira configuração proposta apresentaria um custo cerca de 10% maior que a configuração que apresenta o melhor custo, que seria a segunda.

Desse modo, economicamente, é mais vantajoso utilizar a configuração 02, que teve um tempo médio de execução de 61066, do que a configuração 03, apesar de ter um tempo de execução médio um pouco maior.

#### 4.4 EQUIPE COM SETE PESSOAS

Nas tabelas (Tabela 5, Tabela 6 e Tabela 7) a seguir são mostrados os tempos de execução e taxa de utilização para dez, vinte e trinta fichas considerando haver sete pessoas na equipe.

Tabela 5. Dados das simulações com sete pessoas e dez fichas.

Combinação de Recursos	Taxa de Utilização dos Recursos	Tempo de Execução	
		Mínimo	Máximo
Dois mecânicos, três robotistas e dois analistas	63%	46800	55139
Um mecânico, três robotistas e três analistas	66%	44286	53779
Dois mecânicos, quatro robotistas e um analista	63%	47595	54248
Um mecânico, cinco robotistas e um analista	66%	45814	56813

Tabela 6. Dados das simulações com sete pessoas e vinte fichas.

Combinação de Recursos	Taxa de Utilização dos Recursos	Tempo de Execução	
		Mínimo	Máximo
Dois mecânicos, três robotistas e dois analistas	74%	80546	87277
Um mecânico, três robotistas e três analistas	79%	74165	82838
Dois mecânicos, quatro robotistas e um analista	74%	82190	88985
Um mecânico, cinco robotistas e um analista	80%	75732	83284

Tabela 7. Dados das simulações com sete pessoas e trinta fichas.

Combinação de Recursos	Taxa de Utilização dos Recursos	Tempo de Execução	
		Mínimo	Máximo
Dois mecânicos, três robotistas e dois analistas	77%	115455	122517
Um mecânico, três robotistas e três analistas	83%	105940	114076
Dois mecânicos, quatro robotistas e um analista	77%	118209	124986
Um mecânico, cinco robotistas e um analista	83%	108547	115784

Analisando as informações contidas nas tabelas da Seção 4.2, fica evidente que para uma média de dez robôs, a taxa de utilização dos recursos é baixa – 66% no melhor

caso. Isto significa dizer que, considerando uma jornada de trabalho de oito horas e meia ou 510 minutos, a pessoa estaria efetivamente trabalhando apenas 337 minutos.

Com o aumento do número de robôs, ter uma equipe com sete pessoas se torna mais justificável e rentável, já que você utiliza as pessoas disponíveis com maior frequência, além de ter uma redução no tempo de execução das atividades. Por exemplo, ao comparar as estatísticas para vinte robôs da equipe com configurações 02 e 07, é possível ver que a taxa de utilização dos recursos são próximas – 82% e 80%, respectivamente –, mas o tempo de execução mínimo médio é cerca de 25% menor.

Ainda com as mesmas configurações, mas com o aumento de robôs, a opção com sete pessoas precisou de quase 40% menos tempo para executar as atividades e teve a mesma taxa de utilização, 83%. Na Tabela 8 é possível verificar o ganho quando se leva em conta o processo total.

*Tabela 8. Tempo de execução total para as configurações 02 e 07.*

Combinação de Recursos	Quantidade de Fichas	Tempo Total de Execução	
		Mínimo	Máximo
Um mecânico, três robotistas e um analista	20	207565	223337
Um mecânico, cinco robotistas e um analista	20	151443	162797
Um mecânico, três robotistas e um analista	30	306999	319440
Um mecânico, cinco robotistas e um analista	30	217281	227898

Levando em conta o intervalo mínimo, o ganho de tempo supera os 35% quando comparando a equipe de sete pessoas com a de cinco executando a sequência de atividades em vinte robôs. Se considerado o intervalo máximo, o ganho supera os 40%.

Analisando as informações obtidas com as simulações para equipes de cinco e sete pessoas, é possível perceber que a formação de equipe mais adequada, levando em conta tempo de execução, taxa de utilização e fatores econômicos, seria aquela que apresenta uma maioria de robotistas ao invés de analistas. Os índices obtidos também indicam que ter um mecânico na equipe é o mais interessante, já que, quando o número de mecânicos aumentou na equipe, a taxa de utilização diminuiu e o tempo de execução aumentou.

## 4.5 ANÁLISE DO PROCESSO

Avaliando outras estatísticas geradas pelo *CPN Tools*, é possível verificar que a atividade de movimentação e desembalagem do robô pode se tornar um gargalo caso o número de robôs a serem trabalhados aumentem para 45 ou 50 robôs. Isto é justificável pelo fato de que somente o mecânico pode realizar esta atividade. Assim, um aumento do número de robôs chegando para realização das atividades o sobrecarregará.

É interessante observar, contudo, que a atividade de reembalar o robô não apresenta um tendência tão grande a se tornar um gargalo como a primeira atividade, apesar de somente o mecânico executá-la. Estatisticamente, a décima terceira apresenta 30% da possibilidade apresentada pela primeira atividade. Isto pode ser explicado pelo fato de que o processo suaviza o fluxo de chegada nesta atividade, diminuindo a intensidade de chegada sem afetar o processo como um todo.

Ainda analisando os números gerados, vê-se que o aumento do número de robôs que devem ser preparados para instalação causa um aumento de pedido de recursos proporcional em todas as atividades situadas entre a segunda e décima atividades. As atividades situadas após a atividade de número dez não apresentam tal comportamento. Portanto, as atividades localizadas no meio do processo devem receber uma maior atenção caso ocorra um pico de chegada de robôs.

## 4.6 ATRASO POR FALHA E TAXA DE UTILIZAÇÃO

Ao se representar falhas, esperava-se verificar e avaliar o impacto negativo que ela teria sobre o processo. Contudo, contrariando expectativas, ao se comparar os resultados das simulações com a presença de falha e o resultado sem a presença de falha, a diferença no tempo de execução médio, bem como a taxa de utilização não difere significativamente entre as situações. Para alguns aspectos, não se tem diferença nas estatísticas geradas.

Tabela 9. Resultados para simulações com e sem falhas para a configuração sete e dez fichas.

Combinação de Recursos	Presença de Falha na Rede	Taxa de Utilização dos Recursos	Tempo de Execução	
			Mínimo	Máximo
Um mecânico, cinco robotistas e um analista	SIM	66%	45814	56813
Um mecânico, cinco robotistas e um analista	NÃO	67%	44427	47169

O tempo médio de execução, considerando falhas, é de 49560. Caso não seja considerado as falhas, tem-se um tempo de 45883, o que gera uma diferença média de 8%. Já levando o tempo total de execução, temos os valores mostrados na Tabela 10:

Tabela 10. Tempo total de execução para a configuração sete e dez fichas.

Combinação de Recursos	Presença de Falha na Rede	Tempo Total de Execução	
		Mínimo	Máximo
Um mecânico, cinco robotistas e um analista	SIM	92209	107856
Um mecânico, cinco robotistas e um analista	NÃO	88717	94068

Assim, analisando os dados da Tabela 10, percebe-se que a diferença máxima no tempo de execução é um pouco inferior a 15%.

Tabela 11. Resultados para simulações com e sem falhas para a configuração sete e vinte fichas.

Combinação de Recursos	Presença de Falha na Rede	Taxa de Utilização dos Recursos	Tempo de Execução	
			Mínimo	Máximo
Um mecânico, cinco robotistas e um analista	SIM	80%	75732	83285
Um mecânico, cinco robotistas e um analista	NÃO	80%	73984	78900

Tabela 12. Resultados para simulações com e sem falhas para a configuração sete e trinta fichas.

Combinação de Recursos	Presença de Falha na Rede	Taxa de Utilização dos Recursos	Tempo de Execução	
			Mínimo	Máximo
Um mecânico, cinco robotistas e um analista	SIM	83%	108547	115784
Um mecânico, cinco robotistas e um analista	NÃO	83%	107859	112518

Assim, percebe-se que a influência do atraso gerado por falha sobre o tempo de execução médio é menor percentualmente ao se aumentar o número de robôs trabalhados. O mesmo é observado para o tempo total de execução.

*Tabela 13. Tempo total de execução para a configuração sete e vinte fichas.*

Combinação de Recursos	Presença de Falha na Rede	Tempo Total de Execução	
		Mínimo	Máximo
Um mecânico, cinco robotistas e um analista	SIM	151443	162797
Um mecânico, cinco robotistas e um analista	NÃO	147881	157006

*Tabela 14. Tempo total de execução para a configuração sete e trinta fichas.*

Combinação de Recursos	Presença de Falha na Rede	Tempo Total de Execução	
		Mínimo	Máximo
Um mecânico, cinco robotistas e um analista	SIM	217281	227898
Um mecânico, cinco robotistas e um analista	NÃO	215495	223361

Foi escolhida a sétima configuração para a realização das simulações sem a presença de falha, pois foi a configuração que apresentou um dos melhores resultados com a equipe de sete pessoas. Também foram realizadas estas mesmas simulações para a segunda configuração, cuja equipe tem cinco pessoas. Em função de os resultados obtidos serem bastante similares percentualmente aos mostrados para o cenário com sete pessoas e receberem a mesma análise, as tabelas referentes a este caso não serão mostradas.

Vale destacar que analisar esta influência foi bastante interessante, pois mostra a robustez que o sistema apresenta contra falhas ao se compartilhar recursos, já que, caso não houvesse o compartilhamento de recursos, o atraso mínimo obtido seria um pouco inferior a 50% do tempo de execução ótimo. No pior caso, o atraso por falha poderia chegar a mais de 60% do tempo de execução caso não existissem falhas.

A mesma robustez pode ser encontrada para a taxa de utilização, pois, caso não houvesse paralelismo, os recursos ficariam, no melhor cenário, 23% do tempo parado, pois a décima atividade, que é o teste de movimentação dos robôs, impõe esta restrição de execução. Além disso, ao se realizar as simulações, percebeu-se que existe um limite

de saturação para a taxa de utilização, que gira em torno de 86%, como pode ser visto no gráfico mostrado na Figura 26.

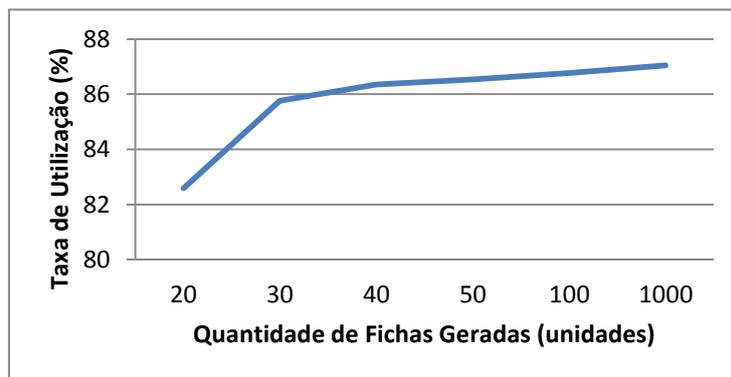


Figura 26. Gráfico em que pode ser visualizada a taxa de utilização (%) em função do número de fichas para a configuração sete.

## 4.7 PARALELISMO INTRÍNSECO

Apesar de não ter sido possível verificar as propriedades da rede quando se tinha atividades que poderiam ser realizadas em paralelo em virtude do conceito de rede, considerou-se que a análise dos dados seria válida. A concepção de rede utilizada foi a mostrada na Figura 24, em que se tem paralelismo em duas atividades. Para realização dos testes foi considerada a equipe com sete componentes, sendo um mecânico, cinco robotistas e um analista. O tempo médio obtido para execução pode ser visualizado na Tabela 15:

Tabela 15. Tempos obtidos com a rede em paralelo para a sétima configuração.

Combinação de Recursos	Quantidade de Fichas	Tempo Médio de Execução
Um mecânico, cinco robotistas e um analista	10	45712
Um mecânico, cinco robotistas e um analista	20	50920
Um mecânico, cinco robotistas e um analista	30	63609

Quando se compara os resultados obtidos com esta configuração de rede com os resultados obtidos para a mesma equipe e quantidade de fichas, mas com uma topologia de rede diferente, sem atividades em paralelo, percebe-se que o ganho que se conquista ao utilizar a topologia mostrada na Figura 24. No primeiro caso, o ganho é de cerca de 8%. Contudo, ao aumentar a quantidade de fichas geradas, o benefício se torna maior,

chegando a ter um tempo de execução médio 55% mais rápido para a segunda linha da Tabela 15.

Foi observado, entretanto, que umas das desvantagens de se utilizar uma rede com paralelismo é ter um menor grau de exatidão com relação ao tempo de execução médio mínimo e máximo. Neste ponto, a topologia de rede que somente apresenta atividades realizadas sequencialmente se mostra melhor.

## 5 CONCLUSÃO

O presente trabalho teve por objetivo avaliar o fluxo de trabalho das atividades realizadas na unidade de negócio Robótica nos robôs Comau, preparando-os para instalação no local de trabalho, utilizando os conceitos de redes de Petri Coloridas associados ao *workflow*.

A revisão bibliográfica realizada foi de fundamental importância para que o trabalho pudesse ser executado, já que foi embasado nos assuntos citados no primeiro parágrafo. Este conhecimento também facilitou o uso da ferramenta *CPN Tools*, que foi utilizada para criar o modelo de representação das atividades e extrair informações do mesmo.

Durante o desenvolvimento do trabalho e também do modelo, foram encontradas algumas limitações, como restrições do *software* utilizado para analisar propriedades e gerar algumas informações. Desse modo, foi necessário alterar algumas configurações e concepções da rede. Entretanto, estas mudanças não descaracterizam este estudo e com o modelo desenvolvido foi possível elucidar questionamentos sobre o processo.

O trabalho aqui descrito alcançou os objetivos propostos. Foram identificados pontos que podem causar retenções no desenvolvimento do processo, bem como a melhor forma de se montar uma equipe visando obter um tempo menor para execução do processo e uma maior taxa de utilização dos recursos. Assim, percebeu-se que a equipe formada por sete integrantes apresenta uma maior flexibilidade, o que é se torna mais interessante para o negócio.

O impacto de falhas também foi verificado. Chegou-se a conclusão de que o fato de se compartilhar recursos diminui a influência de atrasos no resultado final. A mesma conclusão foi feita para a taxa de utilização das pessoas da equipe, que é melhorada quando se tem o compartilhamento durante a execução simultânea do processo em diferentes robôs.

Como sugestões para trabalhos futuros, pode-se realizar a simplificação do modelo, permitindo, desta forma, realizar um maior detalhamento das atividades realizadas, bem como a inclusão de outras tarefas, tornando o modelo mais adequado para outros estudos e investigações acerca do processo. Também é possível modelar outras variáveis, como a disponibilidade de espaço e de ferramentas. Explorar a rede

com paralelismo intrínseco demonstra ser bastante interessante e, fazendo uso da ferramenta correta, os dados obtidos serão de grande valia.

## BIBLIOGRAFIA

- AALST, W.M.P.V.D.. *The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers*. p. 1-53. 1998.
- AALST, W.V.D, HEE, V. K.. *Workflow Management: Models, Methods and Systems*. MIT Press. Cambridge, Massachusetts, London, England. 2002.
- ADAM, N. R., ATLURI, V. HUANG, W.K.. *Modeling and Analysis of workflows Using Petri Nets. Journal of Intelligent Information Systems*. Ed. Rusinkiewicz, M., Abdelsalan, S.H. v. 10 n. 2. P 1-29. Special Issue on Workflow and Process Management. 1998.
- BADOUEL, E. e OLIVER, J.. *Reconfigurable Nets, a Class of High Level Petri Nets Supporting Dynamic Changes within Workflow Systems*. Publication Interne IRISA PI 1163. 1998.
- BECKER, J.; ZUR MUEHLEN, M; GILLE, M.. *Workflow Application Architectures: Classification and Characteristics of Workflow-based Information Systems*, In: FISHER, Layna. (Ed): *Workflow Handbook*, Lighthouse Point, Florida. 2002.
- CARDOSO, J. e VALETTE, R.. *Redes de Petri*. Editora da Universidade Federal de Santa Catarina. UFSC. 1997.
- CINDIO, F. et al. *CHAOS: a knowledge-based system for conversing within offices, office knowledge representation, management and utilization*. Lamersdorf, W.Elsevier Science / B.V. North-Holland.p.257-275. 1988.
- ELLIS, C. e ROZENBERG, G.. *Dynarnic Change Within Workflow Systems. Proceedings of ACM- Conference on Organizational Computing Systems*.1995.
- ELLIS, C. e NUTT, G. J.. *Workflow: the process spectrum*. In: NSF workshop on Workflow and process automation in information system: state-of-the-art and future directions., Athens, Proceedings. p.140-145. 1996.
- FERSCHA, A.. *Qualitative and quantitative analysis of business workflows using generalized stochastic petri nets*. In: WORKFLOW MANAGEMENT - challenges, paradigms and products, Linz,1994. *Proceedings*. Oldenbourg Verlag. p. 222-234. 1994.
- FRANCÊS, C. R. L.. *Introdução às redes de Petri*. Universidade Federal do Pará. UFPA. 2003.
- GARY, K.; LINDQUIST, T.; KOEOEHNEMANN, H.; SAUER, L.. *Automated Process Support for Organizational and Personal Processes. Proceedings of the*

*International ACM IGGROUP Conference on Supporting Group Work: The Integration Challenge (GROUP-97)*, 1997.

HOLLINGSWORTH, David. **Workflow Management Coalition** - The Workflow Reference Model, Document Number TC00-1003, Document Status - Draft 1.1. Disponível em: <http://www.aiai.ed.ac.uk/WfMC> am 04.07.1996. 1996. Acesso em: 21 mar. 2013.

JENSEN, K., KRISTENSEN, L. M., **Coloured Petri Nets**, DOI 10.1007/b95112, (C) Springer-Verlag Berlin Heidelberg. 2009.

KERZNER, H.. **Project Management Logic Puzzles**. John Wiley & Sons. 2006.

LI, J. et al.. **MAS: a declarative formalism for hierarchical representation of procedural knowledge**. *IEEE Transaction on Knowledge and Data Engineering*, v.6, n.4, p.639-643. 1994.

LIMA, A. M. N. e PERKUSICH, A.. **Enciclopédia de automática: controle e automação, volume I**, AGUIRRE, L. (editor); BRUCIAPAGLIA, A.; MIYAGI, P.; HIROSHI, R.; TAKAHASHI, C. (editores associados)1. Ed., São Paulo, Blucher : 313-331. 2007.

MERZ, M. et al.. **Workflow modeling and execution with coloured petri net in COSM**. In: Workshop On Applications Of Petri Net To Protocols Within The International Conference On Application And Theory Of Petri Net, 16. *Proceedings*. p.1-12. 1995.

MURATA, T.. **Petri Nets: properties, analysis and applications**., *Proceedings of the IEEE77(4)*: 541-580. 1989.

NETJES, M., AALST, W.M.P.; REIJERS, H.A.. **Analysis of Resource-Constrained Processes with Colored Petri Nets**. In K. Jensen, editor, *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005)*, volume 576 of DAIMI, pages 251-266, Aarhus, Denmark. University of Aarhus. 2005.

NUTT, G.J.. **The Evolution towards Flexible Workflow Systems**, *Distributed Systems Engineering*, vol3-4, p. 276-294. 1996.

OBERWEIS, A., SCHÄTZLE, R., STUCKY, W., WEITZ, W., ZIMMERMANN, G.. **A Petri-Net Based Approach to Workflow Management**. Ed. Krallmann, H. *Wirtschaftsinformatik'97*, Springer-Verlag. P. 557-580. 1997.

PALOMINO, R. C. **Uma abordagem para a modelagem, análise e controle de sistemas de produção utilizando redes de Petri**. Universidade Federal de Santa Catarina. UFSC. 1995.

PMI – Project Management Institute. **Um Guia do Conhecimento em Gerenciamento de Projetos** (Guia PMBOK) – Quarta Edição. PMI Publications, 2009.

RAD, P. F. e RAGHAVAN, A.. *Establishing na Organizational Project Office*. In: AACE International Transactions, 2000.

SALIMIFARD, S.; WRIGHT M. (2001). *Petri net based modelling of workflow systems: an overview*. European Journal of Operational Research, v. 134, p. 664-676. 2001.

SARIN, S. K. ; ABBOT, K. R.. *Experiences with workflow managements: Issues for the next generation*. Proceedings of Computer-Supported Cooperative Workf. Chapel Hill, USA. 1994.

SCHÖMIG, A. K., RAU, H. *A Petri Net Approach for the Performance Analysis of Business Processes*, University of Würzburg, Report n. 116 Seminar at IBFI, Schloss Dagstuhl, May 22-26. 1995.

VARGAS, R. V.. *Gerenciamento de Projetos – Estabelecendo Diferenciais Gerenciais Competitivos*, 7 ed., Brasport. 2009.

ZISMAN, M.D. *Representation, Specification and Automation of Office Procedures*, Penssylvania. Tese (Doutorado) - University of Penssylvania Wharton School of Business. 1977.

WIKARSKI, D.. *An introduction to modular process net*. Berkeley, International Computer Science Institute. /Technical Report TR-96-019/. 1996.