INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

RICARDO HIDEKI KUBO

PROPOSTA DE SISTEMA DE CONTROLE PARA ALOCAÇÃO DE RECURSOS MULTIFUNCIONAIS E TRANSPORTE (VGAs) TOLERANTE A FALHA

SÃO PAULO
2017

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

DEPARTAMENTO DE MECÂNICA – CAMPUS SÃO PAULO

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

PROPOSTA DE SISTEMA DE CONTROLE PARA ALOCAÇÃO DE RECURSOS MULTIFUNCIONAIS E TRANSPORTE (VGAs) TOLERANTE A FALHA

RICARDO HIDEKI KUBO

Dissertação apresentada ao Programa de Pós-graduação *Stricto Sensu* em Mestrado Acadêmico em Engenharia Mecânica do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus São Paulo, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Mecânica, área de concentração: **Automação e Integração da Manufatura.**

Orientador: Prof. Dr. Francisco Yastami Nakamoto

Coorientador: Prof. Dr. Osvaldo Luis Asato

SÃO PAULO
2017

FICHA CATALOGRÁFICA

K95d Kubo, Ricardo Hideki

Proposta de sistema de controle para alocação de recursos multifuncionais e transporte (VGAs) tolerante a falha / Ricardo Hideki Kubo.

São Paulo: [s.n.], 2017.

113f.: il.

Orientador: Prof. Dr. Francisco Yastami Nakamoto
Coorientador: Prof. Dr. Osvaldo Luis Asato

Dissertação (Mestrado Acadêmico em Engenharia Mecânica) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, IFSP, 2017.

1. Sistemas flexíveis de manufatura 2. Regeneração
3. Máquina-ferramenta multifuncional 4. Sistema de transporte
5. Rede de Petri I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. II. Título

CDD 621

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

MESTRADO ACADÊMICO EM ENGENHARIA MECÂNICA

PROPOSTA DE SISTEMA DE CONTROLE PARA ALOCAÇÃO DE RECURSOS MULTIFUNCIONAIS E TRANSPORTE (VGAS) TOLERANTE A FALHA

RICARDO HIDEKI KUBO

ORIENTADOR: PROF. DR. FRANCISCO YASTAMI NAKAMOTO

COORIENTADOR: PROF. DR. OSVALDO LUIS ASATO

A banca examinadora composta pelos membros abaixo aprovou esta dissertação:

Prof. Dr. Osvaldo Luis Asato
(Coorientador – IFSP – SPO)

Prof. Dr. Givanildo Alves dos Santos
(Membro Interno – IFSP – SPO)

Prof. Dr. Diolino José dos Santos Filho
(Membro Externo – EPUSP)

São Paulo, 25 de agosto de 2017.

Dedico este trabalho à esposa Fabiana e a nossa Naomi;

Aos meus pais, Katsuaki e Satico; e irmãos,

Yukari e Julio.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo consentimento de mais uma oportunidade em minha vida.

Agradeço aos meus pais Katsuaki e Satico, por toda a compreensão, incentivo e apoio incondicional durante o desenvolvimento deste trabalho.

Agradeço à minha esposa Fabiana, por toda a paciência, compreensão, incentivo e apoio incondicional durante o desenvolvimento deste trabalho.

Agradeço aos meus orientadores, Prof. Dr. Francisco Yastami Nakamoto e Prof. Dr. Osvaldo Luis Asato, pela oportunidade, pelo incentivo em todos os momentos, pela constante orientação, pelo voto de confiança, por toda a paciência e pela amizade, minha eterna gratidão.

Agradeço aos colegas do Grupo de Automação e Integração da Manufatura, pela amizade, apoio e incentivo.

Agradeço a todos do Departamento de Mecânica do IFSP que contribuíram direta ou indiretamente para o desenvolvimento deste trabalho.

RESUMO

A competição entre as empresas pelo mercado consumidor global está cada vez mais seletiva devido a vários fatores como o curto ciclo de vida dos produtos, maior quantidade dos modelos do produto, menor custo de fabricação, rápida disponibilidade dos produtos no mercado e requisitos de qualidade integrados a políticas de preservação ambiental. Todos esses fatores contribuíram na evolução do sistema de controle dos Sistemas Produtivos (SP), tornando um sistema altamente flexível tanto em volume quanto na customização de produtos. Para o sistema de controle proposto neste trabalho, os dados de entrada são encaminhados pela área de planejamento de produção, que fornecem a programação (*scheduling)* a ser executada no chão de fábrica, e o problema ocorre quando uma determinada máquina-ferramenta entra no estado inoperante (falha), tornando indisponível este recurso e busca-se regenerar o sistema com o emprego das Máquinas-Ferramenta Multifuncionais (MFM), controle de vias e veículos transportadores. A proposta apresenta um sistema de controle de alocação de recursos com a capacidade de regenerar o SP assegurando a disponibilidade funcional dos recursos ao considerar o conceito de flexibilidade funcional das MFMs, associada à flexibilidade de rotas e mediante o emprego de Veiculos Guiados Automaticamente (VGAs) para a realocação de recursos em tempo real. A arquitetura de controle para a alocação de recursos é modelada utilizando Rede de Petri (RdP).

Palavras-chave: Sistemas flexíveis de manufatura, regeneração, máquina-ferramenta multifuncional, sistema de transporte, Rede de Petri.

ABSTRACT

The competition among companies across the global consumer market is increasingly selective, owing to a number of factors such as the short product lifecycle, larger quantities of product models, lower manufacturing cost, ready availability of products for the final consumer, and quality requirements integrated with environmental policies. All these factors contributed to the evolution of the Production Systems (PS) control system, making it a highly flexible system both for volume as well for product customization. For the control system proposed in this work, input data is sent by the production planning area, which provides the scheduling to be executed on the factory floor, and the problem occurs when a certain machine tool get in an inoperative state (failure), making it unavailable and it is sought to regenerate the system with the use of Multifunctional Machine Tools (MMT), track control and transporters vehicles. The proposal is to present a system for the modeling of resource allocation control systems that has the capacity to regenerate the PS, ensuring the functional availability of resources considering the functional flexibility concept of MMT associated to the flexibility of routes through the use of Automated Guided Vehicle (AGVs) for the reallocation of resources in real time. The control architecture for resource allocation is modeled using Petri Nets (PN).

Keywords: Flexible manufacturing systems, regeneration, multifunctional machine tool, transport system, Petri Nets.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representa os processos "A" e "B" em grafos MFG 17
Figura 2 – Representa a ocorrência de falha no recurso R1 18
Figura 3 – Representa os processos compartilhando o recurso MFM 19
Figura 4 – Evolução de uma váriavel fisica em relação ao tempo. 23
Figura 5 – Evolução de estados em SED 25
Figura 6 – Representação em RdP de duas atividades em sequência. 25
Figura 7 – Representação do paralelismo em RdP de duas atividades 26
Figura 8 – Representação em RdP da sincronização de duas atividades 26
Figura 9 – Representação em RdP do conflito entre duas atividades. 27
Figura 10 – Sistema de produção. 30
Figura 11 – Evolução do centro de torneamento 39
Figura 12 – Evolução das máquinas vista por meio das peças 40
Figura 13 – Modelo estrutural de um sistema de controle de SED proposto por Miyagi (1996) 42
Figura 14 – Representação do sistema de controle proposto por Santos Filho (2000) 43
Figura 15 – Representação do sistema de controle proposto por Nakamoto (2008) 44
Figura 16 – Representação do sistema de controle proposto por Asato (2015) 45
Figura 17 – Formas básicas de arquiteturas de controle de sistemas produtivos 47
Figura 18 – Diagrama de hierarquia funcional 49
Figura 19 – Proposta de arquitetura de controle para SP 53
Figura 20 – Detalhamento do Dispositivo de realização de Controle. 53
Figura 21 – Fluxo das informações entre os módulos da arquitetura proposta. 55
Figura 22 – *Layout* de exemplo de um SP. 57
Figura 23 – Procedimento para síntese dos grafos do sistema de alocação de recursos. 58
Figura 24 – Exemplo de sequência de recursos de três processos.. 59
Figura 25 – Exemplo da síntese do GSF a partir da sequência de recursos 60
Figura 26 – Grafo de controledo GCP derivado do GSF 61
Figura 27 – Refinamento sucessivo - Execução da atividade 62
Figura 28 – Refinamento das funcionalidades no módulo GSF. 63

Figura 29 – Inclusão do *box* controlador e *box* capacidade no módulo GDF 64
Figura 30 – Exemplo de fusão dos *boxes* controladores.....65
Figura 31 – Metodologia PFS/E-MFG para a síntese do CDT 66
Figura 32 – E-MFG do módulo CDT.....69
Figura 33 – PFS do módulo gerador. 70
Figura 34 – Algoritmo para geração do módulo controle.....71
Figura 35 – E-MFG do módulo gerador CUV. 72
Figura 36 – Zonas de carregamento/descarregamento e zona de parada.....74
Figura 37 – Grafo GAR das vias 75
Figura 38 – E-MFG do controle de vias.....76
Figura 39 – Ilustração da comunicação entre os módulos. 79
Figura 40 – Representação Gráfica de um exemplo de uma Rede de Petri.. 91
Figura 41 – Exemplo de RdP para verificação da alcançabilidade.....92
Figura 42 – Exemplo de RdP para verificação da limitabilidade.....93
Figura 43 – Exemplo de RdP para verificação da propriedade de vivacidade 94
Figura 44 – Elementos básicos do MFG 97
Figura 45 – Elementos básicos do E-MFG.....99
Figura 46 – Exemplo de estrutura de uma marca individual 100
Figura 47 – Os boxes funcionais básicos e um exemplo de marca individual composta.....101
Figura 48 – Box controlador alterando o estado de uma marca individual.....101
Figura 49 – Box controlador alterando o estado de uma marca individual.....102
Figura 50 – Alteração dos atributos das marcas decorrentes do disparo.....102
Figura 51 – E-MFG com as interfaces de transmissão e recepção.....103
Figura 52 – Elementos PFS. 105
Figura 53 – Pseudocódigo do algoritmo *Dijkstra* 108
Figura 54 – Exemplo de grafo para cálculo do caminho mínimo.....109
Figura 55 – Processo de cálculo do algoritmo de *Dijkstra*.....109
Figura 56 – Representação da árvore binomial com 13 nós 111
Figura 57 – Representação do nó de um *Heap* Binomial.....111
Figura 58 – Representação do nó de um *Heap* Binomial.....112

LISTA DE TABELAS

Tabela 1 – Tipos de flexibilidade em sistemas produtivos. 38
Tabela 2 – Atributos da marca no módulo GDF. 65
Tabela 3 – Atributos da marca no módulo CDT. 67
Tabela 4 – Atributos da marca no módulo CUV 70
Tabela 5 – Atributos da marca no módulo CU.....74

LISTA DE ABREVIATURAS E SIGLAS

ANSI	*American National Standards Institute*
AGV	*Automated Guided Vehicle*
CDT	Controle de Designação de Transportadores
CP	Controle de Processos
CR	Controle de Recursos
CV	Controle de Vias
CUV	Controle de Utilização de Vias
DF	Designação de Funcionalidades
DR	Designação de Recursos
DT	Designação de Transporte
E-MFG	*Enhanced-Mark Flow Graph*
GAR	Grafo de Alocação de Recursos
GCP	Grafo Controle de Processos
GDF	Grafo Designação de Funcionalidades
GSF	Grafo Sequência de Funcionalidades
ISA	*International Society of Automation*
JIT	*Just in Time*
MC	Manufatura Celular
ME	Manufatura Enxuta
MMT	*Multifunctional Machine Tools*
MFG	*Mark Flow Graph*
MFM	Máquinas-Ferramenta Multifuncional
PFS	*Production Flow Schema*
PMBOK	*Planning, Management Body of Knowledge*

PN	*Petri Nets*
PS	*Production System*
RdP	Rede de Petri
SAR	Sistema de Alocação de Recursos
SED	Sistema a Eventos Discretos
SFM	Sistema Flexível de Manufatura
SHM	Sistemas Holísticos de Manufatura
SMA	Sistema Multi-Agente
SP	Sistema Produtivo
SPF	Sistema Produtivo Flexível
SRM	Sistema Reconfigurável de Manufatura
SVC	Sistema de Variáveis Contínuas
TG	Tecnologia de Grupos
VGA	Veículo Guiado Automaticamente

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO 14

1.1 Motivação do Trabalho 16
1.2 Objetivo 19
1.2.1 Objetivo Geral 19
1.2.2 Objetivos Específicos 19
1.3 Metodologia de Pesquisa 20
1.3.1 Escopo 20
1.3.2 Especificação 21
1.3.3 Desenvolvimento 21
1.3.4 Testes de Validação 21
1.3.5 Implantação 21
1.4 Estrutura do Trabalho 22

CAPITULO 2 - SISTEMAS DINÂMICOS 23

2.1 Sistemas a Eventos Discretos (SED) 24
2.2 Características de SEDs 25
2.2.1 Sequência 25
2.2.2 Paralelismo 26
2.2.3 Sincronização 26
2.2.4 Conflito 27

CAPITULO 3 - SISTEMAS PRODUTIVOS 28

3.1 Sistemas de Apoio à Produção 29
3.2 Classificação de Sistemas Produtivos 31
3.2.1 Tipos de Operações Realizadas 31
3.2.2 Número de Estações de Trabalho 32
3.2.3 *Layout* do Sistema 32
3.2.4 Níveis de Automação e de Apoio Humano 32
3.2.5 Variedade de Peças ou Produtos 32
3.3 Sistemas de Manufatura 33
3.4 *Job Shop* 34
3.5 Complexidade em Sistema Produtivo 35

3.6 Flexibilidade do Sistema Produtivo 37
3.7 Flexibilidade da Máquina-Ferramenta Multifuncional (MFM) 38
3.8 Falha em um Sistema Produtivo 40
3.8.1 Falha no Equipamento de um Sistema Produtivo 41

CAPITULO 4 - SISTEMAS DE CONTROLE PARA SEDs 42

4.1 Modelo Conceitual do Sistema de Controle 42
4.2 Arquitetura de Controle 45
4.3 Norma ANSI/ISA-S95 47
4.4 Controle de Alocação de Recursos 50
4.4.1 Dinâmica da Alocação de Recursos 50

CAPÍTULO 5 - PROPOSTA DO TRABALHO 52

5.1 Arquitetura Proposta 52
5.2 Dinâmica de funcionamento 55
5.3 Método para Implementar os Módulos do Sistema de Alocação de Recursos 58
5.4 Síntese dos Grafos 59
5.5 Síntese do Grafo de Designação de Transporte (DT) 66
5.6 Comunicação entre os módulos 77
5.7 Execução dos Módulos 80

CAPÍTULO 6 - CONCLUSÕES 81

6.1 Contribuição deste Trabalho 81
6.2 Trabalhos Futuros 82

REFERÊNCIAS BIBLIOGRÁFICAS 83

ANEXO A – FERRAMENTAS DE MODELAGEM 90

A.1 Rede de Petri 90
A.2 Propriedades da Rede de Petri 92
A.2.1 Alcançabilidade 92
A.2.2 Limitabilidade e Segurança 93
A.2.3 Vivacidade 93

A.2.4 Reversabilidade....94
A.2.5 Cobertura....95
A.2.6 Persistência....95
A.3 Grafo de Alocação de recursos (GAR)....96
A.4 *Mark Flow Graph* (MFG)....97
A.5 *Enhanced Mark Flow Graph* (E-MFG)....99
A.5.1 Marcas Individuais....100
A.5.2 Marcas Individuais Compostas....100
A.5.3 Manipulação dos Atributos das Marcas....101
A.5.4 Dinâmica das Regras de Disparo....102
A.5.5 Elementos Comunicadores....103
A.6 *Production Flow Schema* (PFS)....105
A.7 Mapeamento GAR/MFG....106

ANEXO B – FERRAMENTA COMPUTACIONAL PARA DETERMINAR O CAMINHO MÍNIMO....108

B.1 Algoritmo de *Dijkstra* com *Heap Binomial*....108

CAPÍTULO

1 INTRODUÇÃO

As empresas estão constantemente realizando melhorias e um dos elementos essenciais para as empresas manterem a vantagem competitiva e sua atuação no mercado é a redução de custos nos processos de fabricação aliados a minimização de recursos ociosos no Sistema Produtivo (SP) (YAMAZAKI, 2016). E dentro desse contexto, o SP deve ser altamente flexível no que se refere ao volume de produção e a customização de produtos além da ampla integração entre clientes, fornecedores e empresas (SHROUF *et al.,* 2014). Com isso, a alta variedade de produtos associada a um ciclo de vida cada vez mais curto, exige um SP ágil e flexível que possa adaptar seu comportamento dinâmico para as possíveis mudanças de ordens de produção e das condições de operações por meio da reconfiguração e otimização de suas instalações e dos recursos disponíveis (WEYER *et al.,* 2015; SHROUF *et al.,* 2014).

O Sistema Flexível de Manufatura (SFM) tem a característica de produzir vários modelos diferentes na linha de produção e possui equipamentos que realizam mais de uma operação no sistema de manufatura. O equipamento deve ser dotado de dispositivos que possam alterar suas operações proporcionando a execução de processos distintos no mesmo equipamento (PEIXOTO, 2012). O SFM se distingue por considerar a diversidade dos produtos, as características adaptativas das máquinas e as propriedades de similaridade dos processos (CAVALCANTE *et al.,* 2010; PEIXOTO, 2012). As principais vantagens apresentadas em um SFM são o aumento do nível de utilização dos equipamentos, a melhora da qualidade do produto, a redução de custos, o rastreamento dos produtos ao longo da produção e menor dependência do conhecimento específico do operador (SAVORY et al., 1991; PEIXOTO, 2012). O desempenho do SFM é dependente da correta alocação dos recursos aos múltiplos processos existentes e da programação da sequência das atividades.

O Sistema Reconfigurável de Manufatura (SRM) apresenta a habilidade de ajustar a capacidade e funcionalidade da produção a mudanças imprevistas de demanda por meio da reconfiguração do sistema, ou seja, deve ser capaz de ajustar rapidamente a capacidade de produção por meio de rearranjo ou mudanças de

componentes como máquinas, esteiras, dispositivos e programas de controladores (MEHRABI *et al.,* 2000). A adaptação rápida a mudança por meio da reconfiguração deve permitir a modularidade, adaptabilidade e escalabilidade do produto de forma operacional por meio de *hardware* aberto e arquiteturas de *software* com funcionalidade *plug and play* para permitir a conexão sem que os demais equipamentos sejam desligados ou reconfigurados (MENDES *et al.,* 2008). O SRM está sendo gradualmente desenvolvido para atingir a funcionalidade do SFM, porém ainda existe uma escassez de literatura sobre o planejamento e programação do SRM em aplicações industriais. As pesquisas sobre o planejamento e programação de produção ainda se concentram principalmente no SFM (AZAB *et al.,* 2014). Nesse cenário, o planejamento da produção torna-se uma atividade cada vez mais complexa, principalmente na ocorrência de mudanças imprevistas na programação. Estes desafios não são abordados de forma eficiente quando se utiliza um SRM e, neste contexto, o planejamento da produção tem de ser ágil e confiável ao mesmo tempo (HEES *et al.,* 2014).

O Sistema Multi-Agente (SMA) é uma tecnologia de *software* capaz de modelar e implementar comportamentos individuais e sociais em sistemas distribuídos. Nesse tipo de sistema, agora chamados de agentes, são ativos e podem postar seus serviços e se submeterem a serem solicitados em suas funcionalidades, mas assumem papel proativo para iniciar a comunicação com outros agentes, propor negociação e alocar serviços (PEIXOTO, 2012). O SMA utilizando o conceito de Sistemas Holísticos de Manufatura (SHM) para a otimização de recursos no chão de fábrica e planejamento em tempo real, são geralmente encontrados em ambientes acadêmicos ou laboratoriais e têm reduzidas aplicações industriais com exceção de algumas áreas no qual o SMA é amplamente difundido como o comércio eletrônico e as cadeias de fornecimento. No entanto, são raras as aplicações no campo industrial, principalmente no campo da produção. Isto se deve, provavelmente, ao tipo de ambiente, às restrições em tempo real e à integração com sistemas físicos de hardware, o que não é o caso de outros domínios, como o comércio eletrônico e cadeias de fornecimento (LEITÃO *et al.,* 2011).

Nesse ambiente competitivo, a contínua necessidade de melhorias nos processos e equipamentos aliados ao avanço tecnológico da automação industrial, abrangendo as tecnologias de comunicações, processamento de informações e da

mecatrônica, combinaram diversos processos de usinagem como furação, torneamento, fresamento, mandrilhamento e rosqueamento em uma única máquina (MORIWAKI, 2008; EIMARAGHY *et al.*, 2012). Essas máquinas são conhecidas como Máquinas-Ferramenta Multifuncionais (MFM), com capacidade de agregar ao SP mais um nível de flexibilidade, a flexibilidade funcional. A flexibilidade se tornou um pré-requisito fundamental para as empresas manterem sua competitividade e podem ser aplicadas em diversos níveis de um SP como, por exemplo, no transporte de itens e nas estações de trabalho (NAKAMOTO, 2008).

A classe de SP abordados neste trabalho é do tipo *Job Shop*, na qual ocorre a execução simultânea de múltiplos processos com compartilhamento de um mesmo conjunto finito de recursos.

1.1 Motivação do Trabalho

Atualmente no chão de fábrica do SP existem as MFM que possibilitam executar múltiplas tarefas (como tornear, fresas, furar etc) no mesmo recurso, e por outro lado existe a programação (*scheduling*) que define um determinado recurso para executar uma determinada etapa do processo.

Diante de uma situação não prevista, como a quebra de um determinado recurso no chão de fábrica, o atual sistema de alocação de recursos não considera (“não enxerga”) as múltiplas funcionalidades contidas na MFM, deixando a MFM ociosa mesmo possuindo a funcionalidade do recurso que entrou no estado inoperante.

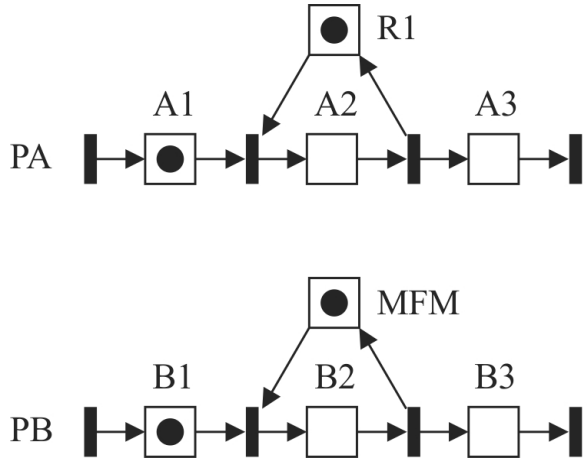
Um fato importante a ser destacado é que, atualmente, as abordagens convencionais de alocação dinâmica de recursos não preveem a utilização de máquinas multifuncionais, evidenciado a necessidade de revisão do paradigma de alocação de recursos para que o enfoque seja na alocação de funcionalidades.

A utilização de MFMs em um SP introduz uma flexibilidade funcional adicional, porém se a alocação de recursos for baseada em uma sequência pré-definida de tarefas, não será possível a exploração de toda a capacidade funcional de uma MFM, pois nesse caso, a MFM será tratada como uma máquina convencional contendo uma única funcionalidade. Esta alocação determínistica de recursos (abordagem estática) é derivada diretamente a partir do plano de produção,

porque evidentemente não considera a ocorrência de eventos anormais no SP (HE *et al*., 2008; LI *et al*., 2010).

Para melhor compreensão do cenário no qual o sistema convencional de alocação de recursos não explora as múltiplas funcionalidades contidas na MFM, a figura 1 ilustra um SP contendo dois processos denominados PA e PB que são descritos usando o MFG (Mark Flow Graph). O processo PA é composto pelas atividades A1, A2 e A3 e o processo PB é composto pelas atividades B1, B2 e B3.

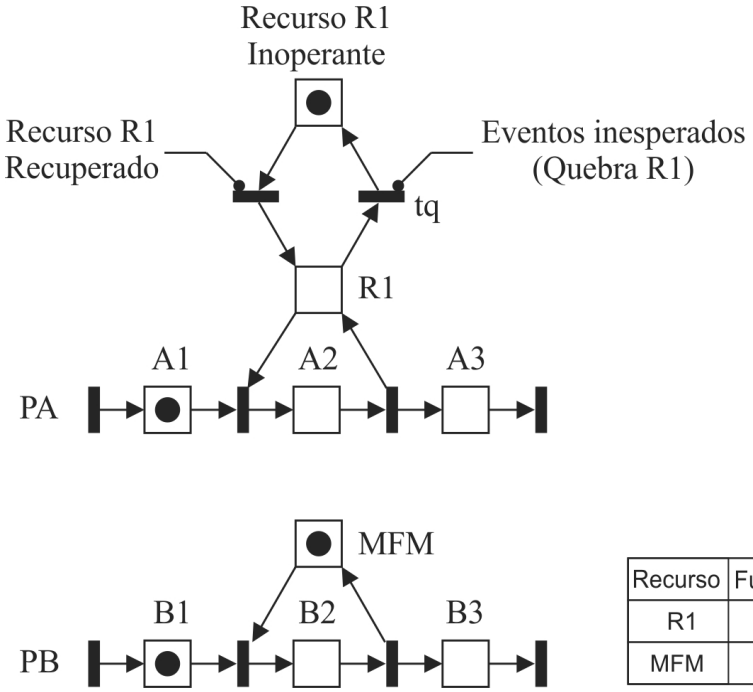
O processo PA possui o recurso R1 com funcionalidade F1 para a execução da atividade A2 e o processo PB possui o recurso MFM com as funcionalidades F1 e F2, sendo utilizada a funcionalidade F2 nesse caso.



Recurso	Funcionalidade
R1	F1
MFM	F1 e F2

Figura 1 - Representa os processos “A” e “B” em grafos MFG.
(Adaptado de ASATO, 2015)

Na ocorrência de um evento não programado como a quebra do recurso R1, a transição tq é habilitada, conforme figura 2, levando o recurso R1 ao estado de falha. Com isso, existe o risco do travamento do processo PA pela indisponibilidade do recurso R1 para a execução da atividade A2, mesmo sabendo que a funcionalidade F1 está disponível no recurso MFM para executar a atividade A2 quando não estiver executando a atividade B2.



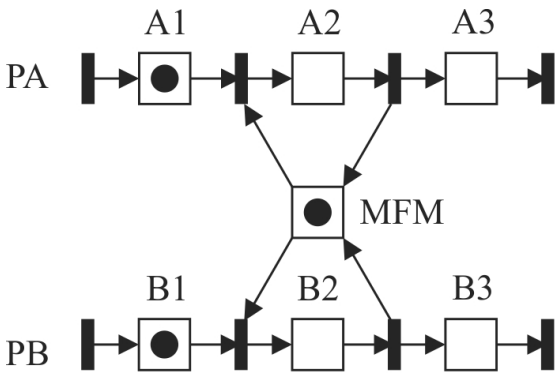
Recurso	Funcionalidade
R1	F1
MFM	F1 e F2

Figura 2 - Representa a ocorrência de falha no recurso R1.
(Adaptado de ASATO, 2015)

Este cenário ilustra a limitação da abordagem convencional, em que a alocação de recursos é baseada em uma sequência de tarefas previamente planejada (ASATO *et al*., 2011).

Devido à dificuldade de alocar os recursos que possuem múltiplas funcionalidades, em tempo real, a proposta do trabalho é desenvolver um método de modelagem do sistema de controle de alocação de recursos que considere o conceito de flexibilidade funcional para regenerar o sistema produtivo diante de um evento não esperado, como por exemplo, a quebra de máquina. Este método permite a exploração da flexibilidade funcional da MFM para a realocação de máquina e também definir o recurso de transporte com uso do VGA mais apropriado para realizar a atividade de transporte da peça que está sendo usinada a fim de regenerar o SP. Com a aplicação dessa nova abordagem no exemplo da figura 1, o sistema de controle de realocação de recursos deverá identificar que a atividade A2 pode ser executada na MFM de forma compartilhada, sem comprometer a execução do processo PA.

A figura 3 ilustra o compartilhamento do recurso MFM entre os processos PA e PB.



Recurso	Funcionalidade
R1	F1
MFM	F1 e F2

Figura 3 - Representa os processos compartilhando o recurso MFM.
(Adaptado de ASATO, 2015)

A ocorrência de eventos não programáveis além de comprometer o desempenho do processo produtivo, pode levar todo o sistema a um estado totalmente indesejável chamado de *deadlock* (travamento), caracterizado pelo impedimento do fluxo de processos devido à falta de recursos, informações ou insumos. Neste sentido, o enfoque do trabalho é apresentar sob uma abordagem dinâmica, uma solução com base na alocação de funcionalidades associada a um sistema de transporte.

1.2 Objetivos

1.2.1 Objetivo Geral

A proposta deste trabalho deve permitir a execução simultânea de múltiplos processos com compartilhamento de um mesmo conjunto finito de recursos com a ampliação e adequação da metodologia proposta em Asato (2015), que considera o conceito de flexibilidade funcional para regenerar o sistema produtivo com a proposta de Nakamoto (2008), com a incorporação de um sistema de transporte com uso de VGA e controle de vias.

1.2.2 Objetivos Específicos

1 – Propor uma arquitetura de controle de processos que tenha como enfoque a alocação de funcionalidades com uso de Máquinas-Ferramenta Multifuncionais (MFM) e um sistema de transporte por Veículo Guiado Automaticamente (VGA) com tolerância a falha nos recursos (máquinas).

2 – Propor uma evolução no modelo de Designação de Funcionalidades, no qual, seja identificado por meio dos atributos da marca, o processo que realizou a requisição de funcionalidades. O modelo também deverá realizar a ordenação das requisições de funcionalidades antes de iniciar o atendimento das requisições.

3 – Propor uma evolução no modelo CDT, na qual a lista de transportadores disponíveis esteja sempre atualizada antes de iniciar a designação de transportadores.

4 – Propor uma evolução no modelo CUV, na qual a execução do procedimento para geração do E-MFG do módulo de controle, seja executada no momento em que não existir nenhuma mensagem de transportador comunicando o término do trajeto.

1.3 Metodologia de Pesquisa

A metodologia de pesquisa aplicada nesse trabalho será baseada no PMBOK (*Planning, Management Body of Knowledge*), que subdivide o gerenciamento de integração do projeto em cinco etapas principais que são a iniciação, o planejamento, a execução, o monitoramento e controle e por fim o encerramento. Para a área de engenharia de automação industrial, pode-se dizer que os mesmos são caracterizados pelas seguintes etapas (DE MORAES *et al.,* 2007):

- Escopo;
- Especificação;
- Desenvolvimento;
- Testes de Validação;
- Implantação.

1.3.1 Escopo

Seu principal objetivo é a elaboração da proposta e da documentação básica para avaliação técnica do projeto de engenharia. Nessa etapa são identificados o objetivo geral, o objetivo específico, a motivação do trabalho e a análise preliminar dos possíveis modelos de controle.

1.3.2 Especificação

Após a análise da etapa anterior, esta etapa consiste no projeto conceitual básico que resultou na especificação funcional do sistema. Nessa etapa é realizado o levantamento bibliográfico pertinente ao trabalho, tais como, os sistemas produtivos e os sistemas de controle.

1.3.3 Desenvolvimento

Esta etapa consiste no desenvolvimento do projeto de engenharia, é desenvolvido o modelo de sistema de controle, os módulos pertencentes ao modelo do sistema de controle e por fim é apresentada a forma como os módulos se comunicam entre si.

1.3.4 Testes de Validação

Esta etapa consiste em um conjunto de simulações e verificações para análise da operacionalidade e consistência dos programas dos controladores e interfaces além de ensaios com dispositivos de sensoriamento e de automação conectados sem a presença do processo industrial. Essa etapa não fez parte desse trabalho.

1.3.5 Implantação

Esta etapa consiste nos testes após a implementação física da automação na planta, na qual deverão ser medidos e aprovados os parâmetros e funcionamento dos recursos, dos elementos de sensoriamento e as condições operacionais do software de automação. Essa etapa não fez parte desse trabalho.

1.4 Estrutura do Trabalho

O Capítulo 1 é dedicado à introdução, na qual são apresentadas as motivações do trabalho, o objetivo geral, objetivo específico e a metodologia de pesquisa aplicada no trabalho.

O Capítulo 2 é dedicado para apresentar um breve estudo sobre sistemas dinâmicos, com foco nos sistemas a eventos discretos.

O Capítulo 3 é dedicado para apresentar o conceito de sistemas produtivos, sistemas de apoio a produção, classificação de sistemas produtivos, sistemas de manufatura, *job shop*, complexidade em sistemas produtivos, flexibilidade de sistemas produtivos e falhas em sistemas produtivos.

O Capítulo 4 é dedicado para apresentar o conceito de sistema de controle, arquitetura de controle, a norma ANSI/ISA-S95, controle de alocação de recursos e a dinâmica da alocação de recursos.

O Capítulo 5 é dedicado para apresentar a proposta do trabalho.

O Capítulo 6 é dedicado para as principais conclusões e sugestões de contribuições na evolução deste trabalho. Por fim, a bibliografia e os anexos sobre as ferramentas de modelagem e a ferramenta computacional para determinar o caminho mínimo**.**

CAPÍTULO

2 SISTEMAS DINÂMICOS

Os sistemas dinâmicos podem ser analisados dentro da mecânica newtoniana clássica como ‘forças e energia produzindo um movimento’. Dessa forma, podemos aplicar o termo dinâmico a todos os fenômenos térmicos, químicos, fisiológicos, ecológicos e muitos outros fenômenos que possam ser regidos pelas equações diferenciais ou de diferenças, em que o tempo é a variável independente (ROSÁRIO, 2005).

Os sistemas podem ser classificados a partir de diversos critérios e uma possível divisão é a de Sistema de Variáveis Contínuas (SVC) e Sistema a Eventos Discretos (SED) (HO, 1987; HO, 1989).

Os sistemas dinâmicos dirigidos pelo tempo são sistemas que atuam quando trata de sistemas de controle contínuo, como por exemplo, as variáveis do nível de pressão, temperatura, vazão entre outros, já na automação, existe o predomínio dos sistemas dinâmicos dirigidos por eventos que são os sistemas acionados por eventos, como por exemplo, a linha de montagem das empresas automobilísticas (DAVID *et al.,*1994).

A figura 4 ilustra a representação da evolução de uma variável contínua em relação ao tempo.

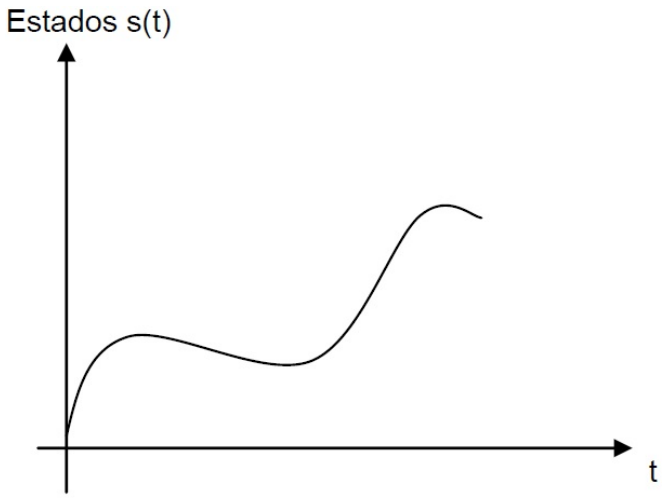


Figura 4 – Evolução de uma variável física em relação ao tempo.
(ASATO, 2015)

2.1 Sistemas a Eventos Discretos (SED)

Os SEDs estão presentes em muitas aplicações tecnológicas, porém existem dificuldades em encontrar um modelo matemático conciso que o tornem computacionalmente viável, diferente do modelo matemático de equações diferenciais que modelam os sistemas dinâmicos de variáveis contínuas (HO, 1991).

Segundo Ho (1987), SED tornou-se essencial devido a inúmeras e importantíssimas aplicações em sistemas de automação da manufatura por possuir como principal característica a modificação de seu estado representado por estados lógicos. Estes sistemas não são representados por equações diferenciais ou de diferenças e são conhecidos como sistemas dinâmicos acionados por eventos (MORAES *et al.,* 2007).

O SED é um sistema invariante no tempo, de estado discreto e não linear dirigido por eventos instantâneos que ocorrem em intervalos de tempos irregulares e desconhecidos, caracterizando no sistema um primeiro nível de indeterminismo com relação ao tempo (CASSANDRAS, 1993; RAMADGE *et al.,* 1988; SANTOS FILHO, 2000).

De acordo com Villani (2000), um evento discreto é entendido como um evento no qual sua ocorrência pode ser considerada instantânea, ou seja, de duração desprezível dentro do sistema. Dessa forma, o SED pode ser representado por uma série de estados que se mantém constante durante um período de tempo e se modificam de forma instantânea.

Os SPs abordados neste trabalho fazem parte da classe de SEDs e possuem como característica a realização de processos produtivos compostos por uma sequência de etapas, sendo cada uma dessas etapas interpretadas como uma ação ou conjunto de ações que transformam o item que está sendo processado.

A figura 5 ilustra o comportamento de um SP visto como um SED, na qual, no eixo da abscissa, estão os eventos e, no eixo da ordenada, estão os respectivos estados discretos.

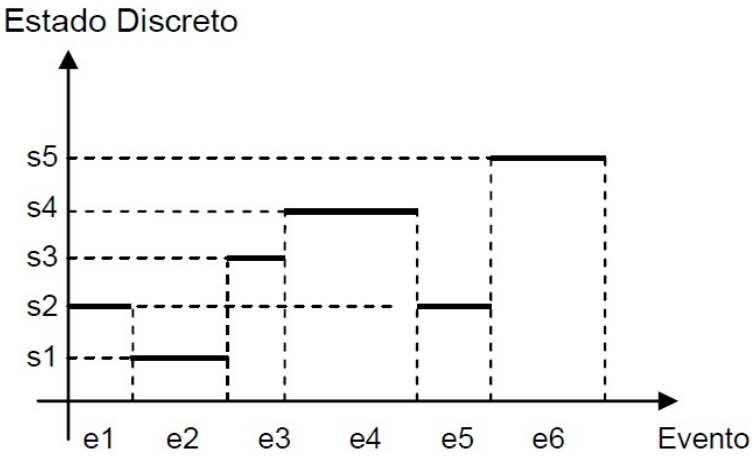


Figura 5 – Evolução de estados em SED.
(ASATO, 2015)

2.2 Características de SEDs

O comportamento dinâmico de SEDs pode ser descrito por meio da evolução de seus estados e são realizados a partir da ocorrência de eventos considerados instantâneos e algumas das características encontradas nos processos em SEDs são: sequência, paralelismo, sincronização e conflito (SOUZA, 2015).

2.2.1 Sequência

Característica que representa a execução de uma ação, desde que seja atendida uma determinada condição, ou seja, uma determinada atividade deve ser concluída para que outra atividade seja iniciada.

A figura 6 ilustra a representação em RdP de duas atividades em sequência.

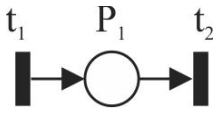


Figura 6 – Representação em RdP de duas atividades em sequência (adaptado de SOUZA, 2015).

2.2.2 Paralelismo

Característica que indica que as atividades podem ser executadas concomitantemente sem que haja interferência da execução entre si, mas ambos com a mesma origem, ou seja, uma única condição inicial.

A figura 7 ilustra a representação em RdP de duas atividades em paralelas.

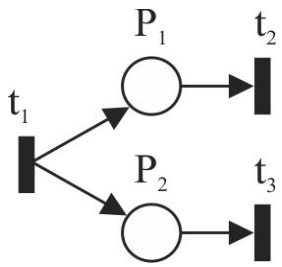


Figura 7 – Representação do paralelismo em RdP de duas atividades (adaptado de SOUZA, 2015)

2.2.3 Sincronização

Característica que indica que uma atividade depende da execução de outras atividades. A sincronização pode ser realizada entre atividades em paralelo e entre atividades com origens independentes.

A figura 8 ilustra a representação em RdP da sincronização de duas atividades.

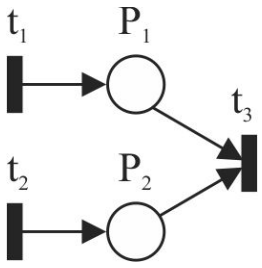


Figura 8 – Representação em RdP da sincronização de duas atividades (adaptado de SOUZA, 2015).

2.2.4 Conflito

Característica que indica que apenas uma das atividades pode ser executada e a execução de uma das atividades inibe a execução das outras atividades que estavam em conflito.

A figura 9 ilustra a representação em RdP de conflito entre duas atividades.

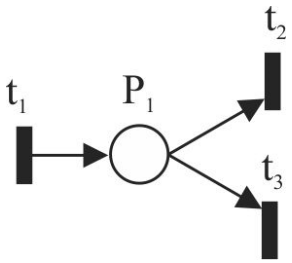


Figura 9 – Representação em RdP de conflito entre duas atividades (adaptado de SOUZA, 2015).

CAPÍTULO

3 SISTEMAS PRODUTIVOS

O SP é definido por um sistema composto de máquinas, pessoas, robôs e outros dispositivos em que as atividades dos processos são executadas por meio da transformação da matéria-prima, que ao final do processo resultará em um produto (GROOVER, 2011; SANTOS FILHO *et al*., 2011). Dessa forma, o objetivo de um SP consiste em agregar valor a um determinado produto por meio da execução de tarefas que podem ser interpretadas como transformação de estado ou prestação de serviço, entendendo-se que no caso da prestação de serviço, o item que recebe o serviço teve seu estado alterado. Com isso, a organização e posteriormente a especificação da tecnologia de automação a ser empregada no SP será em função da interpretação das tarefas (MIYAGI *et al.,* 2000; SANTOS FILHO, 2000; GAITHER *et al.,* 2001).

Matsusaki (2004), afirma que os processos produtivos de um SP são constituídos por diversas etapas, em que as etapas podem ser interpretadas como um conjunto de ações que transformam o estado do item que está sofrendo o processo, ou seja, o objetivo de um SP pode ser entendido como a execução dos processos produtivos.

A evolução dinâmica de um SP é regida pela ocorrência de eventos que geram mudanças nas condições e nos estados do sistema podendo provocar situações de conflito, concorrência e assincronismo (GROOVER, 2011; SANTOS FILHO *et al*., 2011). Considerando a evolução dos estados dentro de um SP, podemos classificá-los como um SED, que é um sistema no qual a evolução ocorre dinamicamente em função da ocorrência de eventos (HO, 1989; CASSANDRAS, 1993).

Com a execução simultânea e paralela de vários processos por meio de um conjunto finito de recursos compartilháveis no chão de fábrica, perde-se o sequenciamento das atividades do processo tornando praticamente impossível a previsão de todos os estados alcançáveis do sistema além de gerar um alto grau de dificuldade no modelamento do seu comportamento (NAKAMOTO, 2008). Os estados alcançáveis de um SP composto por diversos elementos independentes podem ser representados pela equação 1.1 (MIYAGI, 1996).

$$N_t = \prod_{i=1}^{m} N_i \qquad (1.1)$$

Em que:

N_t = Número total de estados alcançáveis;

m = Elementos do sistema;

N_i = Número de estados individuais de cada elemento.

Dentre os possíveis estados alcançáveis do sistema pode-se citar a indisponibilidade de recursos que possui capacidade de levar o sistema ao estado de travamento (*deadlock*) dos processos, e dentro desse contexto, a indisponibilidade dos recursos no SP torna-se um fator fundamental para a redução da produtividade e do aumento de custos pré-definidos (ASATO, 2015).

O SP tratado neste trabalho envolve a produção de bens materiais, itens ou produtos em que suas operações podem ser caracterizadas pela execução simultânea de múltiplos processos utilizando um conjunto finito de recursos. Dessa forma, pertencente à classe de SP de produção do tipo *Job-Shop*.

3.1 Sistemas de Apoio à Produção

Segundo Groover (2011) os sistemas de produção podem ser divididos em duas categorias ou níveis denominados de:

1. Instalações: As instalações do sistema de produção representam as fábricas, máquinas, equipamentos e os sistemas computadorizados que controlam as operações de produção. Incluem também o *layout* da fábrica referente à organização física dos equipamentos, que são organizados em grupos lógicos.

2. Sistemas de apoio à produção: É o conjunto de procedimentos utilizados pela empresa para gerenciar a produção, garantir o atendimento dos requisitos de qualidade e solucionar problemas técnicos e logísticos.

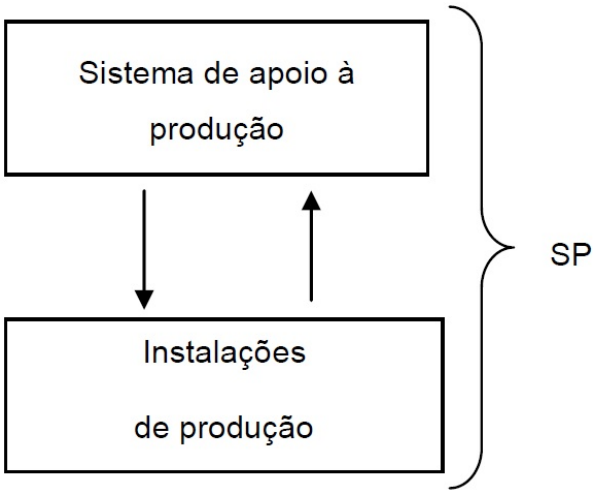


Figura 10 – Sistema de produção.
(Adaptado de GROOVER, 2011)

O apoio à produção envolve um ciclo de atividades que são descritos por quatro funções: (1) funções de negócios, (2) projeto do produto, (3) planejamento da produção e (4) controle da produção.

1. Funções de negócio: É o principal meio de comunicação com o cliente e representam o início e o fim do ciclo de processamento de informações. Nesta categoria estão vendas e marketing, previsão de vendas, registro de pedido, contabilidade de custo e cobrança ao cliente.

2. Projeto do produto: A partir do departamento de vendas e marketing se inicia o ciclo de eventos do projeto de um novo produto. Se necessário, são gerados protótipos que podem envolver funções de pesquisa e desenvolvimento.

3. Planejamento da produção: Todas as informações e documentações que compõem o projeto do produto seguem para a função do planejamento da produção. Nesta fase incluem o planejamento do processo, do plano-mestre, dos requisitos do projeto e do planejamento da capacidade.

3.1 Planejamento do processo: Consiste na determinação da sequência de atividades e operações necessárias para a produção da peça.

3.2 Plano mestre-produção: É uma lista de produtos a serem produzidos, data de entrega e quantidades.

3.3 Planejamento das necessidades de materiais: Planejamento da aquisição de matéria-prima, peças e itens para que estejam disponíveis quando necessário.

3.4 Planejamento da capacidade: Conhecer a capacidade produtiva para que não sejam listadas quantidades de produtos mais altas do que a capacidade de produção da fábrica.

4. Controle da produção: Essa fase preocupa-se com o controle e gerenciamento das operações físicas realizadas na fábrica com vista à execução do plano de produção.

3.2 Classificação de Sistemas Produtivos

Segundo Groover (2011), os SPs podem ser classificados por fatores que os definem e os distinguem, na qual:

- Tipos de Operações Realizadas;
- Números de Estações de Trabalho;
- *Layout* do Sistema;
- Nível de Automação e Apoio Humano;
- Variedade de Peças ou Produtos.

3.2.1 Tipos de Operações Realizadas

Segundo Groover (2011), nesta classificação, os SPs possuem no nível mais alto, a distinção entre as operações de processamento nos itens individuais e de montagem de combinar peças individuais em entidades montadas. Além disso existe a distinção nas tecnologias das operações individuais de processamento e de montagem. Ainda, conforme o autor, existem alguns parâmetros adicionais dos produtos que influenciam no projeto do SP que são o tipo de material processado, tamanho e peso da peça do produto, complexidade da peça ou do produto e a geometria da peça.

3.2.2 Número de Estações de Trabalho

De acordo com Groover (2011), o número de estações de trabalho é um fator vital para a classificação de SPs por exercer enorme influência sobre o desempenho do SP como capacidade de carga de trabalho, velocidade de produção e estabilidade. Conforme o número de estações cresce, a quantidade de trabalho a ser realizada no SP também cresce traduzindo em maior velocidade de produção quando comparada com a produção de uma única estação. Porém mais estações de trabalho significa que o sistema é mais complexo se tornando mais difícil de ser gerenciando e mantido.

3.2.3 *Layout* do Sistema

Relacionado com o número de estações de trabalho e a forma como as estações de trabalho são organizadas em um SP. O *layout* das estações é um importante fator para determinar o sistema de manuseio de material mais apropriado. Os *layouts* das estações de trabalho podem ser organizados para roteamento fixo, de maneira linear como uma linha de produção ou como roteamento variável podendo ter diferentes configurações possíveis (GROOVER, 2011).

3.2.4 Níveis de Automação e de Apoio Humano

O nível de automação é outro fator que caracteriza o SP, no qual as máquinas podem ser operadas manualmente, semiautomatizadas ou automatizadas. Inversamente ao nível de automação, está a proporção do tempo de dedicação de um operador em cada estação de trabalho. O nível de apoio humano é a proporção de tempo que o operador necessita estar presente na estação (GROOVER, 2011).

3.2.5 Variedade de Peça ou Produto

Segundo Groover (2011), outra classificação de um SP é a sua capacidade de lidar com variações nas peças ou nos produtos que fabrica. As variações podem ser do tipo ou cor do plástico de peças injetadas, variações nos componentes eletrônicos montados em uma placa de circuito impresso, variações no tamanho das placas de circuito impresso manipuladas por uma máquina de posicionamento de componentes, variações na geometria de peças fabricadas por maquinário e variações nas peças e opções em um produto montado.

Existem três casos de variedade de peça ou produtos que são o modelo único, modelo em lote e o modelo misto.

No modelo único, todas as peças ou produtos fabricados são idênticos não existindo variação e o equipamento associado a esse modelo é especializado e projetado para máxima eficiência.

No modelo em lote, diferentes peças ou produtos são fabricados pelo sistema, mas são produzidos em lotes pois é necessária uma troca na configuração física e/ou programação do equipamento devido as diferenças significativas no tipo da peça ou do produto.

No modelo misto, diferentes peças ou produtos podem ser fabricados no SP, porém com a limitação das diferenças serem pouco significativas (leve variação entre peças ou produtos). Dessa forma, o sistema é capaz de manipular as diferenças sem necessidade de trocas demoradas de configuração ou de programação, possibilitando a produção contínua e não em lotes de tipos diferentes de peças ou produtos.

3.3 Sistemas de Manufatura

A Manufatura Enxuta (ME), também conhecida como *Just in Time* (JIT) ou modelo Toyota nasceu na empresa Toyota. É um modelo baseado em importantes aspectos como a eliminação de qualquer fonte de desperdício, a participação dos empregados e critérios de qualidade (PIAO *et al.,* 2007; ESPEJO *et al.,* 2007). Esse modelo incorporou ferramentas e métodos nos processos produtivos que facilitaram a análise dos problemas presentes em uma linha de montagem além de solucionar os problemas presentes em um SP por meio de conceitos de melhora da qualidade (RAJADELL *et al.,* 2010).

A Tecnologia de Grupo (TG) é uma filosofia de manufatura na qual as peças semelhantes são agrupadas e identificadas para tirar vantagem de suas similaridades em projeto e produção. As peças são dispostas em famílias, na qual possuem características semelhantes de manufatura e projeto. É utilizado o termo Manufatura Celular (MC) quando existe uma organização do trabalho como o agrupamento de máquinas ou a agregação de máquinas ou processos em células dedicadas à produção de uma peça ou famílias de produtos (GROOVER, 2011).

Um Sistema Flexível de Manufatura (SFM) é uma célula de manufatura TG altamente automatizada, composta por grupos de estações de processamento interligadas por um sistema automatizado de manuseio e armazenamento de materiais controlados a partir de um sistema distribuído de computação. O motivo do SFM ser chamado de flexível é devido a capacidade de controlar um conjunto de peças simultaneamente em diversas estações de trabalho, na qual as quantidades de produção e a variedade de tipos de peças podem ser ajustadas em resposta às mudanças de demanda (GROOVER, 2011).

O Sistema de Manufatura Integrada por Computador (CIM), evoluiu como resultado da evolução dos sistemas computacionais e permitiram maior agilidade nas etapas de projeto e na fabricação de um produto (AARDAL, 1995). Esse modelo é dividido em níveis em que no primeiro nível corresponde ao chão de fábrica e no segundo nível são estabelecidos os sistemas de controle, as atividades de projeto e prototipagem do produto utilizando conceitos de Projeto Auxiliado por Computador (CAD) (KATEEL *et al.,* 1996; WANG *et al.,* 2007). Integrado aos dois primeiros níveis está a supervisão industrial por meio de sistemas SCADA (*Supervisory Control Architecture and Data Acquisition*), em que é realizado o monitoramento e configuração do SP. Um nível superior se encontra a parte gerencial do processo que permite estruturar os sistemas de planejamento de necessidades de materiais e planejamento dos recursos de manufatura (RAY, 1998; ZHAO *et al.,* 2009).

3.4 *Job Shop*

O *Job Shop* (JS) consiste no planejamento de um conjunto de tarefas que devem ser executadas em um conjunto de máquinas, na qual para cada tarefa é definida uma sequência de operações a serem processadas, cada uma em uma máquina específica, que executa uma única operação e o seu processamento não pode ser interrompido após o seu início (PREVIERO, 2016).

Existem algumas restrições aos sistemas do tipo *job shop* que são (ARAÚJO JUNIOR, 2006):

- **Precedência**: quando as operações de cada tarefa têm que ser executas na ordem prevista.
- **Recirculação**: quando é necessário processar duas ou mais operações de um mesmo *job* na mesma máquina.
- **Capacidade**: cada máquina pode processar somente uma tarefa de cada vez.
- **Preempção**: cada operação deve ser concluída sem interrupção, ou seja, não preemptivo.

Dessa forma, o JS é um ambiente formado por máquinas que devem processar um conjunto de *jobs*, em que cada *job* possui uma sequência própria de operações, cujo processamento nas respectivas máquinas demandam de intervalos fixos.

Neste trabalho a classe de Sistemas Produtivos abordados é do tipo *Job Shop*, com múltiplos processos e número finito de recursos.

3.5 Complexidade em Sistema Produtivo

A complexidade de um SP está relacionada com o intenso compartilhamento de um conjunto finito de recursos por meio da execução simultânea de múltiplos processos e pode ser dividida basicamente em dois tipos de domínios: o domínio funcional e o domínio físico (SANTOS FILHO, 2000; SUH, 2005).

No domínio físico, a complexidade é classificada em estática e dinâmica (DESHMUKH *et al.*, 1998). A complexidade estática se refere à estrutura e configuração do sistema, por exemplo, número de produtos, variedade de produtos e variedade de elementos no sistema. A complexidade dinâmica está relacionada com a incerteza do comportamento do sistema para um período de tempo específico (SUH, 2005; PAPAKOSTAS *et al.*, 2009).

Para o presente trabalho a interpretação da complexidade é dada segundo a qual o comportamento dinâmico de um SP como um todo não partilha das mesmas particularidades do comportamento dinâmico de uma das partes do SP.

Neste contexto, a complexidade de modelar o comportamento dinâmico de um SP com MFMs é derivada de três tipos de indeterminismos sendo os dois

primeiros indeterminismos descritos por Santo Filho (2000) e o terceiro e último indeterminismo descrito por Asato (2015):

- *Indeterminismo em relação ao tempo*: Impossibilidade de determinar o momento exato em que ocorrerá um evento devido a existência de paralelismos, conflitos e assincronismo dos processos.
- *Indeterminismo em relação a eventos*: Impossibilidade de determinar a sequência que ocorrerão os eventos devido a existência e execução simultânea de múltiplos processos com compartilhamento de um conjunto finito de recursos.
- *Indeterminismo em relação à sequência de uso de funcionalidades nos recursos que possui flexibilidade funcional*: Impossibilidade de determinar, no caso das MFMs, qual funcionalidade precede outra porque existe o indeterminismo da sequência de ocorrência de eventos.

Os três indeterminismos destacados acima são considerados somente na implementação de uma solução para a alocação/realocação dinâmica dos recursos de transformação inerentes ao processo (ASATO, 2015; NAKAMOTO, 2008).

A existência de atividades de transportes entre as estações de trabalho resulta em mais dois indeterminismos dependentes do sistema de transporte adotado (NAKAMOTO, 2008).

- *Indeterminismo em relação à designação do transportador*: Relacionado com a impossibilidade de determinar o transportador que será designado para atender uma solicitação de transporte, caso seja utilizado um transportador do tipo universal.
- *Indeterminismo em relação à atividade de transporte*: Relacionado com as rotas e uso das vias.

Dessa forma, a presença dos indeterminismos em um SP torna desafiador o projeto do sistema de controle por resultar em uma explosão combinatória de estados alcançáveis, sendo necessária escolher adequadamente a arquitetura de controle e aplicar técnicas que possam restringir os estados alcançáveis impedindo que o sistema evolua para um estado indesejável.

3.6 Flexibilidade do Sistema Produtivo

Para que um sistema produtivo seja flexível, são identificados três recursos: (1) capacidade de identificar e distinguir os diferentes tipos de peças ou produtos processados pelo sistema, (2) a rápida troca das instruções operacionais e (3) da configuração física (GROOVER, 2011).

Para que um SP seja considerado flexível, são necessárias algumas características como a habilidade de identificar e distinguir os vários tipos de processos executados pelo sistema, alterando as instruções de configurações físicas e as de operação de acordo com os recursos disponíveis (GROOVER, 2011).

A flexibilidade é um atributo que caracteriza o SP pela produção simultânea de variados produtos e sua capacidade em se adaptar a diferentes dinâmicas de seus processos, como por exemplo, alterações de demanda de tipos de produtos e/ou quantidade de itens de produtos (NAKAMOTO, 2008; SANTOS FILHO *et al.*, 2011; GROOVER, 2011).

A flexibilidade de um SP pode ser definida como a habilidade de mudar ou reagir com pouco esforço, custo, tempo e mantendo seu alto desempenho (MATSUSAKI *et al.*, 2006).

De acordo com Groover (2011), a flexibilidade de um SP pode ser classificada conforme apresentada na tabela 1.

Tabela 1 – Tipos de flexibilidade em Sistemas Produtivos (GROOVER, 2011).

Tipos de Flexibilidade	Definição	Fatores
Máquina	Capacidade da máquina executar uma variedade de operações.	Tempo de setup, facilidade de reprogramação, capacidade de armazenar ferramentas, habilidade e versatilidade dos operadores no sistema.
Produção	Capacidade de produção de uma variedade de processos.	Flexibilidade das máquinas
Mix	Habilidade de alterar o mix de produção enquanto mantém a mesma quantidade total de produção.	Similaridade das operações de processo e flexibilidade das máquinas.
Produto	Capacidade de produzir um novo produto.	Similaridade do novo produto para classificá-lo em uma família de produto existente e flexibilidade das máquina.
Rotas	Capacidade de produzir mediante sequências alternativas.	Similaridade de mix, similaridade de estações de trabalho, duplicação de estações de trabalho.
Volume	Habilidade de produzir de forma econômica baixo e alto volume de um mesmo produto.	Nível de trabalho manual executado na produção.
Expansão	Capacidade de expansão do sistema para aumentar a quantidade total produzida.	Aquisição de estações de trabalho, disponibilidade física.

Outro ponto importante é que a flexibilidade é influenciada por três aspectos (GROOVER, 2011; SANTOS FILHO, 2000; PELS *et al.*, 1997):

- Seleção de tecnologias;

- Custo de implementação;

- Escolha da arquitetura de controle.

3.7 Flexibilidade da Máquina-Ferramenta Multifuncional (MFM)

Para atender o competitivo mercado consumidor, a contínua necessidade de implementar melhorias nos processos produtivos, resultaram na integração de múltiplas funcionalidades de fabricação em uma única máquina como furação, torneamento, fresamento, mandrilhamento e rosqueamento. Essa evolução é motivada pelo aumento de eficiência do SP, originada pela capacidade de aplicar múltiplos processos de fabricação em uma única máquina contribuindo inclusive na redução da complexidade do SP. Atualmente essas máquinas estão sendo introduzidas nas indústrias de transformação como as indústrias automobilística e

aeroespacial (MORIWAKI, 2008; CHOI *et al.,* 2013; BRECHER *et al.*, 2016; EIMARAGHY *et al.*, 2012).

A função fundamental da máquina-ferramenta é de transformar a matéria-prima com dadas propriedades mecânicas em peças acabadas atendendo a requisitos de qualidade que envolvam dimensões, geometrias e acabamento. Para atender essa função, a MFM possibilita controlar diversos eixos lineares e circulares com o objetivo de produzir peças complexas com alta precisão e qualidade além de visar a redução do tempo global de usinagem e oferecer além da flexibilidade operacional a flexibilidade funcional em somente uma única máquina (MORIWAKI, 2008; EIMARAGHY *et al.*, 2012).

A figura 11 ilustra a evolução das máquinas-ferramenta, em específico o centro de torneamento.

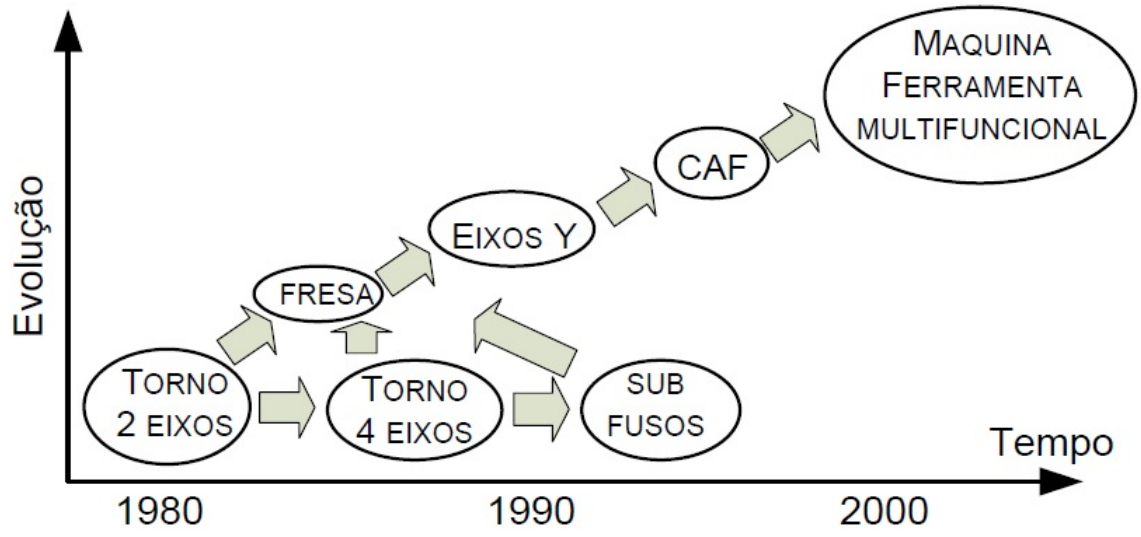


Figura 11 – Evolução do centro de torneamento.
(Adaptado de MORIWAKI, 2008)

A figura 12 ilustra a evolução das máquinas-ferramenta do ponto de vista das peças que as máquinas são capazes de usinar.

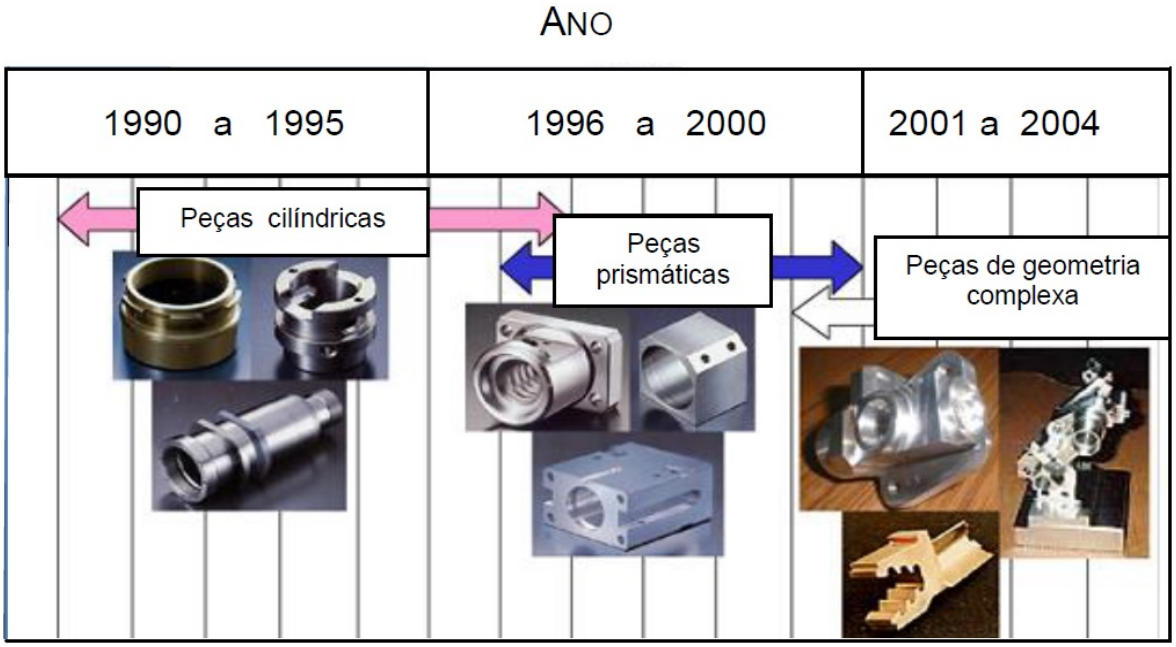


Figura 12 – Evolução das máquinas vista por meio das peças.
(Adaptado de MORIWAKI, 2008)

3.8 Falha em um Sistema Produtivo

A previsibilidade da ocorrência de uma falha no sistema produtivo com exatidão se torna praticamente impossível e seu tratamento deve ser executado assim que solicitado (RIASCOS, 2002). O tratamento de falhas em um sistema pode ser entendido como um mecanismo com 4 fases de atuação que são (DA SILVA, 2016):

1- Estimação: Fase que realiza um diagnóstico para identificar a falha.

2- Planejamento: Fase em que ocorre a tomada de decisão da ação a ser executada.

3- Execução: fase que é executada a ação.

4- Aprendizagem: Fase em que os dados relevantes são armazenados para facilitar outras atuações do tratamento de falhas.

As fases de tratamento de falhas atuantes neste trabalho são somente a fase de planejamento e a fase de execução. Neste trabalho as fases de Estimação e de Aprendizagem não são contempladas. Para a fase de Aprendizagem, é interessante a utilização de técnicas de inteligência artificial que permitam trabalhar com aprendizagem contínua no sistema.

3.8.1 Falha no Equipamento de um Sistema Produtivo

A previsibilidade de todos os estados alcançáveis de um SP se torna praticamente impossível e dentre os estados possíveis, um dos estados alcançáveis e indesejáveis de um SP é o estado de falha de equipamentos. Os equipamentos físicos não possuem risco de falha nulo e tal ocorrência pode resultar em sérios riscos à integridade física das pessoas, ao meio ambiente, perdas econômicas decorrentes de danos ao próprio equipamento e perdas econômicas de travamento (*deadlock*) de um processo produtivo (SALLAK *et al.,* 2008; SQUILANTE JR *et al.,* 2011).

A norma ABNT 5462 afirma que as falhas podem ser classificadas como:

- *Falha aleatória*: Qualquer falha que seu instante de ocorrência se torna imprevisível, a não ser no sentido probabilístico.
- *Falha por deterioração*: Falha que resulta de mecanismos de deterioração os quais determinam uma taxa de falha crescente com o tempo devido ao seu envelhecimento.
- *Falha catastrófica*: Falha repentina que não pode ser prevista e resulta na total incapacidade do item de desempenhar suas funções.

Neste trabalho, a falha de um equipamento fica caracterizada quando o mesmo não pode desempenhar suas funções (primárias ou secundárias) mesmo não estando completamente inoperante.

É importante salientar que neste trabalho as falhas no equipamento aplicáveis são somente as classificadas como aleatórias ou por deterioração.

CAPÍTULO

4 SISTEMAS DE CONTROLE PARA SEDs

O controle pode ser definido como a aplicação de ações pré-determinadas para que o objeto de controle atinja os objetivos que reflitam no comportamento dinâmico desejado ao sistema (MIYAGI, 1996; SANTOS FILHO, 2000). O comportamento dinâmico desejado no SP é o cumprimento dos planos de produção estabelecidos pelo planejamento e que podem apresentar propriedades de assincronismo por depender da satisfação das condições de evolução do sistema e de paralelismo quando o sistema de controle SED é formado por subsistemas e as evoluções dos passos ocorrem de forma paralela (independente) (MIYAGI, 1996).

Com isso, a presença de assincronismo e de paralelismo ao sistema de controle SED ocasiona uma competição na utilização dos recursos compartilhados, exigindo a aplicação de técnicas adequadas com a finalidade de solucionar problemas desta natureza (MIYAGI, 1996; SANTOS FILHO, 2000; NAKAMOTO, 2008).

4.1 Modelo Estrutural do Sistema de Controle

A figura 13 ilustra o modelo estrutural básico de um sistema de controle de SED apresentado por Miyagi (1996), na qual a atuação do sistema de controle sobre o objeto de controle é realizada por meio dos dispositivos de atuação e a realimentação é feita por meio dos dispositivos de detecção. O elemento humano atua no sistema por meio dos dispositivos de comando e monitora o sistema de controle por meio dos dispositivos de monitoração. É um exemplo de modelo estrutural no qual os elementos estão classificados e é estabelecida uma conexão lógica entre eles.

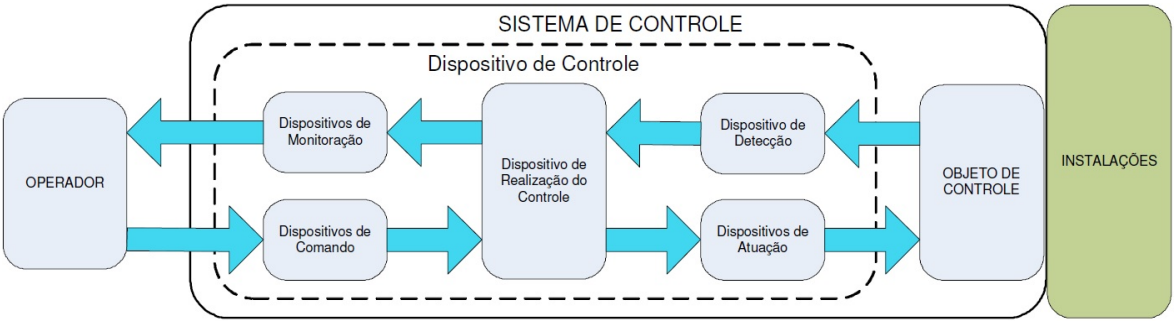


Figura 13 – Modelo estrutural de um sistema de controle de SED proposto por Miyagi (1996).

Este modelo mostra a relação entre os componentes do sistema não considerando situações no qual exista o compartilhamento de recursos entre os processos. Nessa situação existe a perda da estruturação do sistema e como consequência, a perda de controle do estado global do sistema.

Diante dessa limitação, Santos Filho (2000) propôs uma divisão do sistema de controle em dois níveis de semânticas distintas denominadas de Controle de Processos (CP) e Controle de Recursos (CR). Nesta arquitetura houve uma evolução, o bloco CP que era único, passou a ser visto como dois blocos de controle gerando o módulo CP e o módulo CR. O módulo CP é responsável pelo sequenciamento das atividades dos processos e pela requisição dos recursos e o módulo CR é responsável pela utilização dos recursos entre as atividades de cada um dos processos.

A figura 14 ilustra o sistema de controle proposto por Santos Filho (2000).

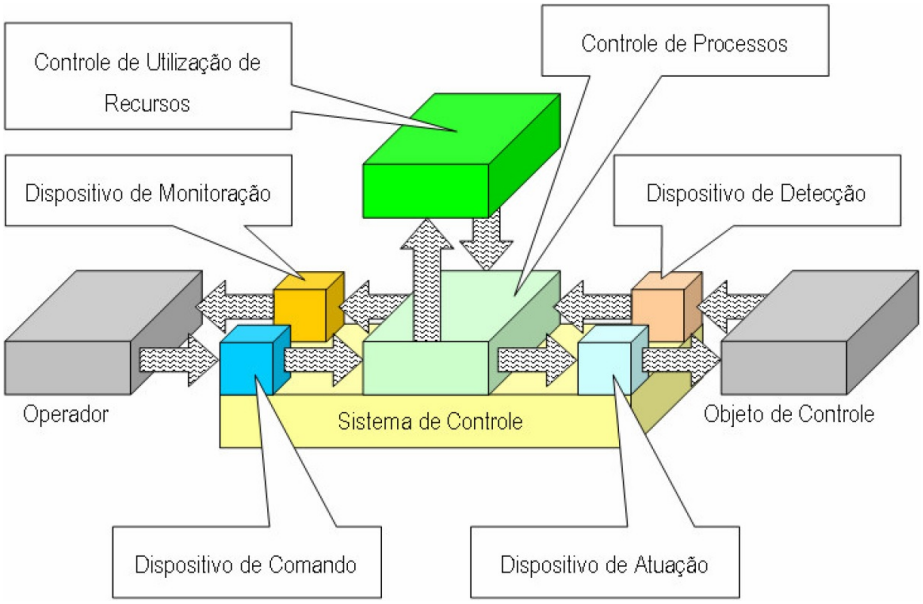


Figura 14 – Representação do sistema de controle proposto por Santos Filho (2000).

As principais vantagens apresentadas com a divisão do controle são a possibilidade de modularizar a estrutura de controle, a distinção semântica dos controles e a aplicação de restrições na abordagem de explosões de estados. Esta lógica de controle é utilizada em diferentes paradigmas como arquitetura distribuída, arquitetura hierárquica e a arquitetura heterárquica.

Nakamoto (2008) propôs uma arquitetura de controle modular que visa a realização efetiva do controle do SP mediante a modularização das funções de controle de acordo com o domínio considerado, ou seja, o controle de atividades de transformação preocupa-se somente com a produção do item (produto). Por fim, o controle de atividades de transporte fornece o serviço de transporte de itens entre as estações de trabalho para o controle de atividades de transformação. Porém, a atividade de transporte envolve o compartilhamento de recursos entre os transportadores que, no caso específico, são as vias. Desta forma, o controle de atividades de transporte também deve gerenciar a utilização destas vias pelos VGAs a fim de evitar o problema de travamento dos processos.

A figura 15 ilustra o sistema de controle proposto por Nakamoto (2008).

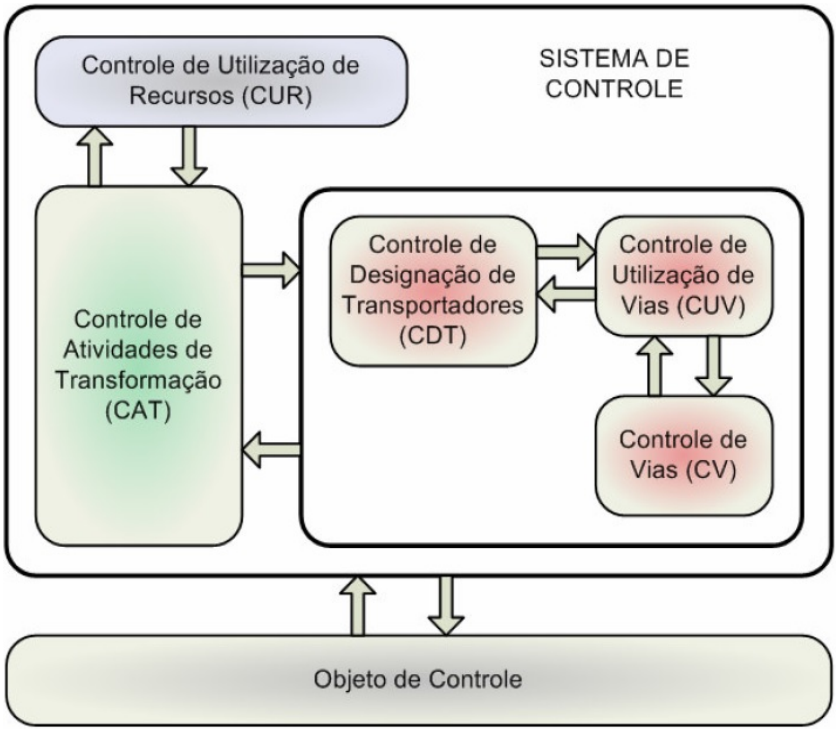


Figura 15 – Representação do sistema de controle proposto por Nakamoto (2008).

Asato (2015) propôs uma arquitetura de controle de alocação de recursos com capacidade de tratar eventos não programados, como a ocorrência de quebra ou indisponibilidade de recursos (máquinas) no SP. Como solução, explora-se a disponibilidade de recursos com flexibilidade funcional e para controlar o uso desses recursos, utiliza-se o conceito de sequências de funcionalidades. Portanto, quando ocorre uma falha no processo causado por problemas nos recursos, o sistema de

alocação de recurso utiliza o conceito de flexibilidade funcional para realocar os recursos e, consequentemente, retirar o sistema do estado de travamento.

A figura 16 ilustra o sistema de controle proposto por Asato (2015).

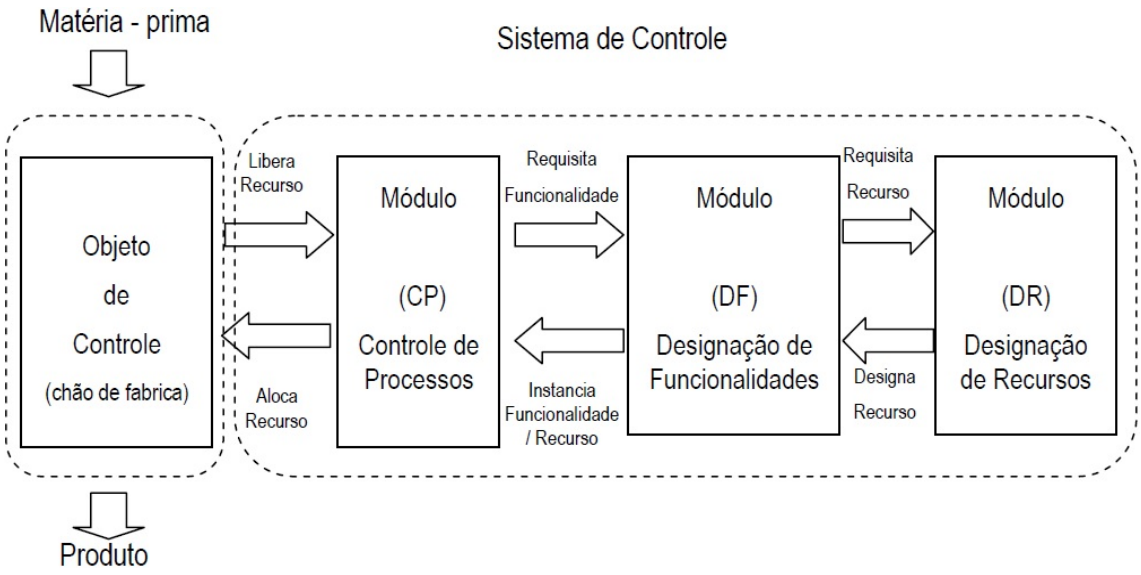


Figura 16 – Representação do sistema de controle proposto por Asato (2015).

4.2 Arquitetura de Controle

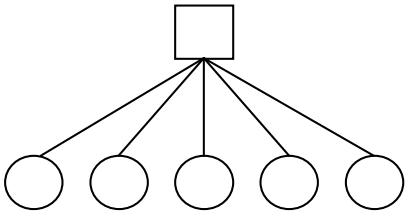
A definição de arquitetura de um sistema é dada como sendo a estrutura e organização lógica de funcionamento de um sistema, na qual estão determinadas as interfaces, inter-relações e restrições entre seus componentes (DILTS *et al.,* 1991; PELS *et al.,* 1997). Desta forma, as arquiteturas podem ser classificadas em (PELS *et al.,* 1997; MATSUSAKI, 2004; SANTOS FILHO, 1998):

- *Arquitetura centralizada*: Possui como característica o gerenciamento de todas as funções de planejamento e informações de evolução dos processos produtivos por meio de um computador ou controlador central. As principais desvantagens desse tipo de arquitetura é o crescente tempo de resposta de acordo com o porte do sistema e a limitação de realizar modificações pela capacidade de processamento do controlador central. A execução de múltiplos processos simultâneos por meio do intenso compartilhamento de recursos flexíveis, torna o processo global do sistema em um sistema complexo e dessa forma, a abordagem centralizada se torna impraticável.

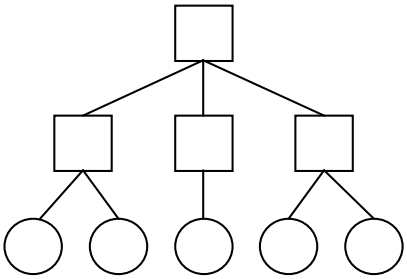
- *Arquitetura distribuída*: Possui como característica a distribuição das estratégias de controle entre diversos elementos. Com isso, o sistema alcança plena autonomia de controle local além da cooperação para a tomada de decisões globais.

As arquiteturas distribuídas são consideradas uma evolução da arquitetura centralizada principalmente quanto ao quesito flexibilidade no processo produtivo e nesse sentido, os sistemas distribuídos podem ser classificados em (PELS *et al.,* 1997; DILTS *et al.,* 1991; MATSUSAKI, 2004; BONGAERTS *et al.,* 2000):

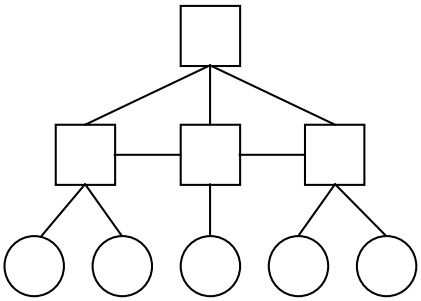
- *Hierárquico*: Apresenta níveis e módulos de controle dispostos em níveis de forma piramidal no qual as atividades possuem inter-relacionamentos verticais entre o módulo e seus subordinados (DILTS *et al.,* 1991; PELS *et al.,* 1997; BONGAERTS *et al.,* 2000). As funções de controle superiores são decompostas para os níveis inferiores conforme a abordagem de comando *top-down* e realimentação de informações *bottom-up* (BONGAERTS *et al.,* 2000).
- *Heterárquico*: Caracterizada pelo elevado grau de autonomia de seus elementos de controle para que ocorra cooperação na resolução de problemas de controle (MATSUSAKI, 2004). As estratégias de controle são realizadas por meio da negociação entre os elementos de controle (BONGAERTS *et al.,* 2000).
- *Híbridos*: Apresenta características hierárquicas e heterárquicas de controle, no qual os elementos possuem uma relação *master-slave* considerando certo grau de autonomia e cooperação entre os controladores (MARTINS, 2005).



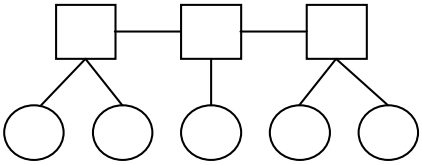
(a) Forma Centralizada



(b) Forma Hierárquica



(c) Forma Hierárquica Modificada



(d) Forma Distribuída

Obs. Os elementos □ representam dispositivos de controle.

Figura 17 – Formas básicas de arquiteturas de controle de sistemas produtivos (SANTOS FILHO, 1998).

4.3 Norma ANSI/ISA-S95

O propósito da norma ANSI/ISA-S95 (*American National Standards Institute* / *International Society of Automation*) é a criação de um padrão que defina a interface dos processos de manufatura com os processos empresariais (ANSI/ISA, 2005). O objetivo é a redução de riscos, custos e erros associados à implementação dessas interfaces e a definição da troca de informações de forma robusta, segura e econômica, sendo que o mecanismo de troca deve preservar a integridade das informações de cada sistema. Com isso, a norma trata alguns pontos importantes como a padronização das terminologias, das interfaces de troca de dados, das funções ou atividades a serem executadas por diferentes sistemas e a distribuição dessas funções em uma hierarquia (ASATO, 2015). Segundo Dobrev (2008), as padronizações tratadas pela norma ANSI/ISA-S95 servem para:

- Dar ênfase às boas práticas de integração dos sistemas de controle com os sistemas corporativos durante o ciclo de vida dos sistemas;
- Ser utilizada para melhorar a capacidade de integração existente dos sistemas de controle da manufatura com os sistemas corporativos;
- Ser aplicada independentemente do grau de automação.

A norma ANSI/ISA-S95 foi concebida para fornecer os seguintes benefícios (ASATO, 2015):

- Redução do tempo para se atingir os níveis de produção plena de novos produtos;
- Desenvolvimento de novas ferramentas para implementar a integração dos sistemas de controle com os sistemas corporativos;
- Permite que as empresas identifiquem suas necessidades de forma mais otimizada;
- Redução de custos na automatização dos processos de manufatura;
- Otimização das cadeias de fornecimento;
- Redução dos esforços do ciclo de vida de engenharia.

O foco da norma ANSI/ISA-S95 está na redução dos riscos, custos e erros operacionais com a implementação de interfaces do processo de manufatura com os processos empresariais e a definição da troca de informações entre as interfaces. Dessa forma, a norma estabelece um modelo de atividades e fluxo de dados denominando-o de diagrama de hierarquia funcional na qual são ilustrados os diferentes níveis de hierarquia do modelo funcional (NAKAMOTO, 2008; ASATO, 2015).

A figura 18 ilustra o diagrama de hierarquia funcional.

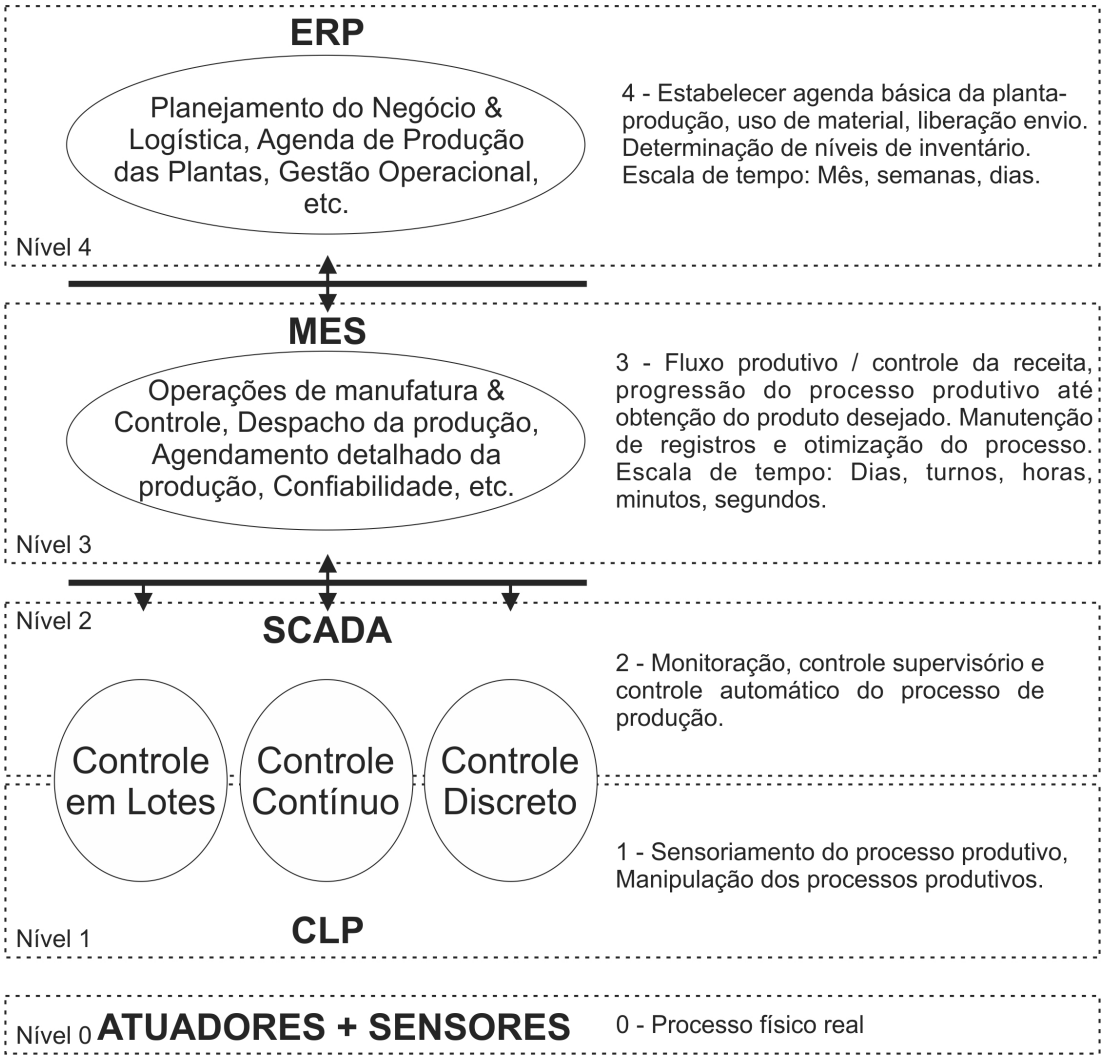


Figura 18 – Diagrama de hierarquia funcional.
(Adaptado ANSI/ISA, 2005)

Basicamente, o diagrama de hierarquia funcional identifica os seguintes níveis (NAKAMOTO, 2008):

- *Nível 0*: Relacionado ao processo físico real;
- *Nível 1*: Relacionado com as atividades envolvidas na coleta de sinais e manipulação dos processos físicos;
- *Nível 2*: Relacionado com as atividades de monitoração e controle local dos processos físicos. Neste nível o equipamento é tratado dentro de um centro de trabalho conforme a hierarquia proposta na norma ANSI/ISA-S95;

- *Nível 3*: Relacionado com as atividades que realizam a progressão dos processos para gerar os produtos finais, incluindo as atividades de manutenção de registros e coordenação dos processos. Neste nível a escala de tempo sobre as áreas de centros de trabalho são em dias, turnos, horas, minutos e segundos.
- *Nível 4*: Relacionado com as atividades do negócio que são necessárias para manter o sistema de manufatura. Entre as atividades estão o estabelecimento do planejamento de produção, determinação dos níveis de inventário e garantia de que os materiais são liberados em tempo e no lugar certo para a produção. Neste nível a escala de tempo opera em mês, semanas e dias.

4.4 Controle de Alocação de Recursos

Controlar um SP significa impor um conjunto de ações que tem como finalidade assegurar o comportamento dinâmico desejado para se alcançar um determinado objetivo (SANTOS FILHO *et al.,* 2011). Em um SP o comportamento dinâmico desejado é o cumprimento dos planos de produção com uma variedade de produtos, utilizando os recursos disponíveis com a aplicação de regras de controle para a execução das atividades dos processos (ASATO, 2015). A definição da forma de alocação e utilização dos recursos pelos processos é responsabilidade do Sistema de Alocação de Recursos (SAR).

4.4.1 Dinâmica da Alocação de Recursos

Um processo é um conjunto sequencial de atividades a serem executadas para se resultar em um produto. Com isso, a existência de vários processos simultâneos com compartilhamento de recursos resulta em uma competição entre os processos na alocação de recursos e uma das formas para tratar esse problema é o estabelecimento de regras de alocação de recursos para garantir a realização de todas as atividades (SANTOS FILHO, 2000; TRICAS *et al.,* 2014):

- *Regra 1 – Requisitar Recurso*: É a verificação da disponibilidade do recurso para a execução da atividade;
- *Regra 2 – Alocar o Recurso*: Autorizada a requisição, é a reserva do recurso pelo processo;

- *Regra 3 – Utilizar o Recurso*: Alocado o recurso, é a execução da atividade pelo processo;
- *Regra 4 – Liberar o Recurso*: Concluída a atividade, somente o processo que o alocou poderá liberar o recurso para outros processos.

Dessa forma, o sistema de alocação de recurso é o responsável pela escolha do recurso para a execução de cada atividade que compõem os diversos processos.

CAPÍTULO

5 PROPOSTA DO TRABALHO

Inicialmente, as primeiras arquiteturas dos sistemas de controle de alocação de recursos possuíam um único módulo que era responsável pelo controle dos recursos e pelo sequenciamento das atividades dos processos (MIYAGI, 1996). Dessa forma, nos SPs que possuíam o compartilhamento de recursos entre processos, ocorria à perda da estruturação do sistema e como consequência, a perda de controle do estado global do sistema. Diante dessa limitação, foi proposta a divisão do sistema de controle em dois níveis denominados de controle de processos (responsável pelo sequenciamento das atividades dos processos e pela requisição dos recursos) e de controle de recursos (responsável pela utilização dos recursos entre as atividades de cada um dos processos). Com a evolução tecnológica dos sistemas mecatrônicos, foi possível o uso cada vez maior das MFMs e diante desta nova realidade, foi necessário desenvolver uma nova arquitetura para alocar os recursos (máquina e VGA) em tempo de execução (ASATO, 2015; NAKAMOTO, 2008) e baseado nesta arquitetura desenvolve-se este trabalho.

5.1 Arquitetura Proposta

A proposta deste trabalho é desenvolver um sistema de controle para alocação de recursos multifuncionais com transporte por VGA mediante a ocorrência de falha do recurso para regenerar o SP. Diante da ocorrência de falha de uma determinada máquina, o sistema de controle irá designar o recurso e o VGA mais apropriado para regenerar o SP.

A figura 19 ilustra a arquitetura proposta no trabalho com base nos trabalhos de Asato (2015) para alocação de recurso multifuncional e no trabalho de Nakamoto (2008) para designar o VGA, realizar os cálculos das rotas e controle de vias.

A figura 20 ilustra o detalhamento do dispositivo de realização de controle.

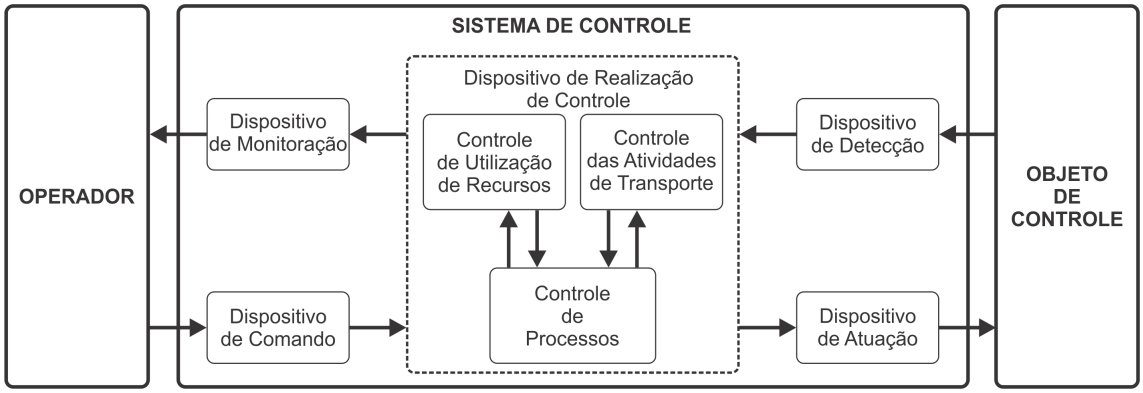


Figura 19 - Proposta de arquitetura de controle para SP.

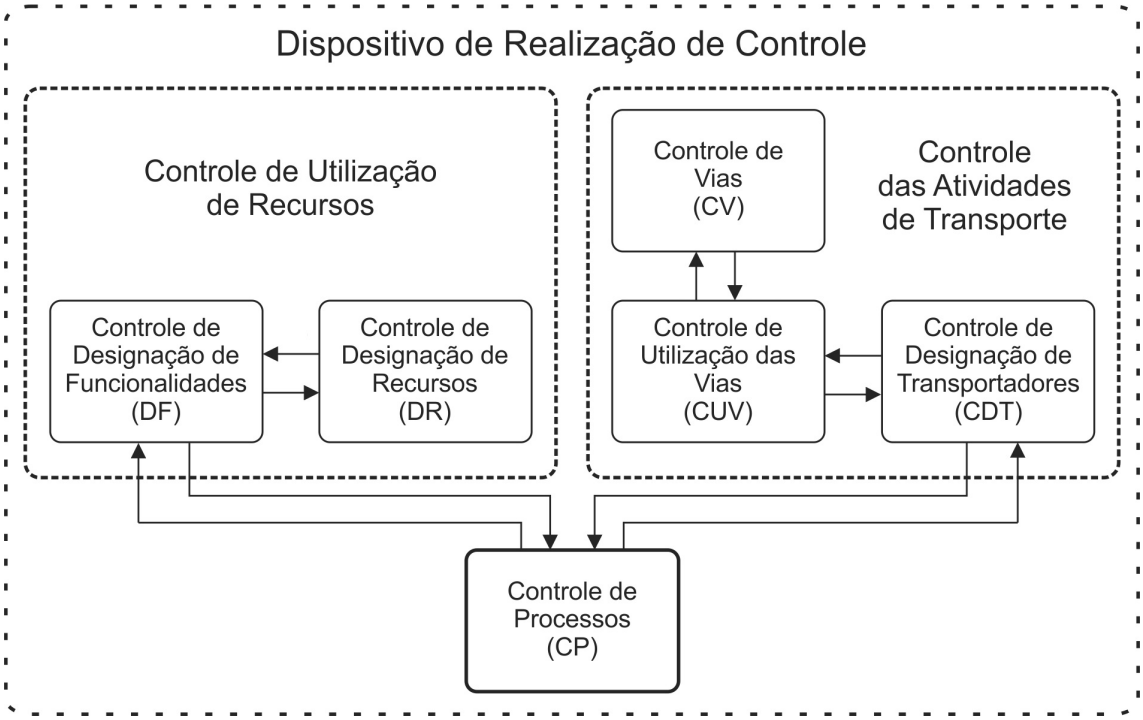


Figura 20 – Detalhamento do Dispositivo de Realização de Controle.

Controle de Processos

- *Módulo CP:* Módulo responsável por realizar uma função de controle com a sequência de funcionalidades dos processos produtivos pertencentes ao SP por meio de um grafo (derivado da Rede de Petri). Por meio do grafo é possível descrever a interface entre os recursos do chão de fábrica e os processos do SP com as funcionalidades a serem executadas. Este módulo também estabelece a sequência das atividades a serem executadas por meio da alocação e liberação de recursos.

Controle de Utilização de Recursos

- *Módulo DF*: Módulo responsável pela estruturação das funcionalidades de mesmo nome do SP. A estruturação consiste no agrupamento das funcionalidades de mesmo nome e na conversão das sequências de produção em sequências de funcionalidades. Dessa forma, a alocação de recursos passa a ter referência na funcionalidade e não mais no recurso.
- *Módulo DR:* Módulo responsável pela execução de um algoritmo heurístico que resulte na designação do recurso mais adequado para substituir um recurso inoperante do SP.

Controle das Atividades de Transporte

- *Módulo CDT:* Módulo responsável pela execução de um algoritmo heurístico que resulte na designação do transportador (VGA) para a realização da atividade de transporte entre as estações de trabalho com a definição dos trajetos de transporte.
- *Módulo CUV:* Módulo responsável pelo gerenciamento da utilização das vias pelos transportadores.
- *Módulo CV:* Módulo responsável pelo controle do tráfego de transportadores pelas vias.

O processo de transporte consiste em duas atividades, uma atividade de carregamento na origem (trajeto I) e outra atividade de descarregamento no destino (trajeto II).

A arquitetura do sistema de controle proposta neste trabalho possibilita regenerar o SP diante de um evento não programado como a falha de uma máquina, utilizando as múltiplas funcionalidades disponíveis nas MFMs e realizar a realocação de recursos mediante o emprego do sistema de VGAs com flexibilidade de rotas.

5.2 Dinâmica de funcionamento

O módulo CP realiza o controle da execução sequencial das atividades de cada processo que são executadas nas estações de trabalho do SP por meio da alocação e liberação de recursos. Entre a execução das atividades de transformação, existe a atividade de transporte que irá movimentar os itens entre as estações de trabalho.

A figura 21 ilustra o fluxo de comunicação entre os módulos.

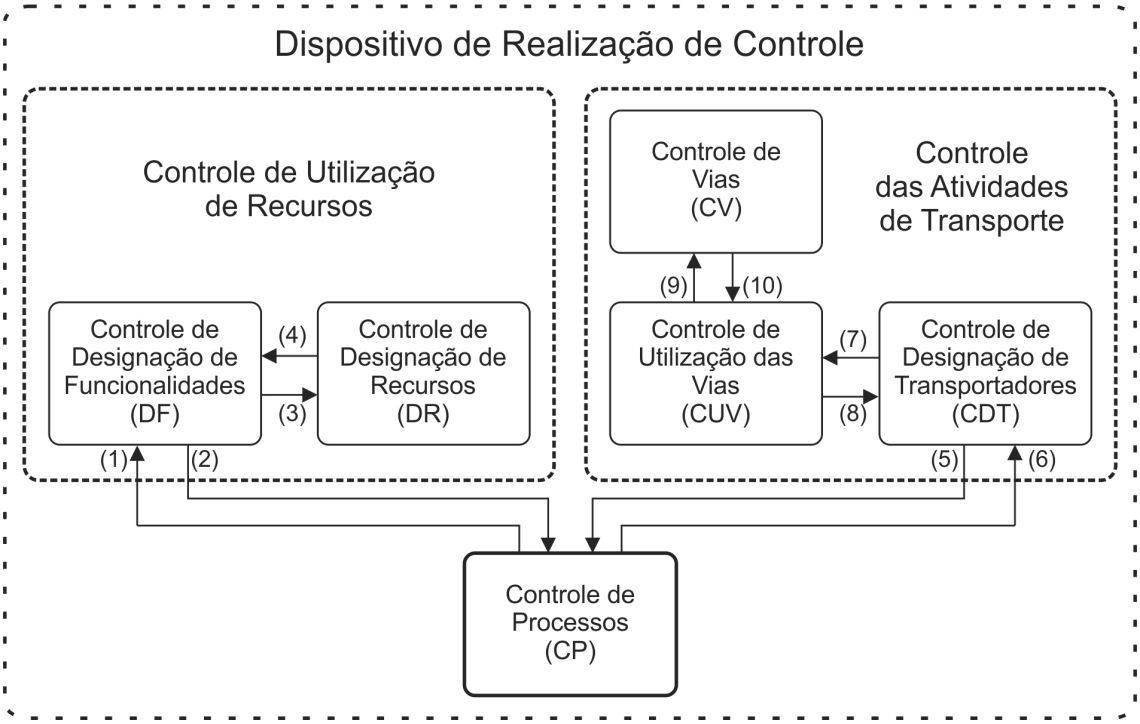


Figura 21 - Fluxo das informações entre os módulos da arquitetura proposta.

A alocação de recursos do SP é definida pela programação do planejamento de produção, porém na ocorrência de um evento inesperado como a falha ou indisponibilidade de algum recurso (máquina) no chão de fábrica, o módulo CP enviará um pedido de requisição de um novo recurso para o módulo DF, representado pelo fluxo (1). O módulo DF será responsável em identificar o recurso e a funcionalidade afetada e informar o módulo DR, representada pelo fluxo (3). De posse dessas informações e por meio da execução de algoritmos heurísticos, o módulo DR deve determinar o recurso mais apropriado para substituir o recurso inoperante e retornar essa informação ao módulo CP, representado pelo fluxo (4). O fluxo (6) representa as requisições de transporte do módulo CP ao módulo CDT.

O módulo CDT recebe do CP as requisições de transporte com as seguintes informações:

- Identificação do processo;
- Estação de trabalho origem;
- Estação de trabalho destino.

Com essas informações, o módulo CDT organiza os pedidos em uma fila de requisições de acordo com a ordem de chegada dos pedidos. Dessa forma, a atividade de designação do transportador dependerá da existência de transportadores disponíveis. De posse das informações referentes à disponibilidade dos transportadores, o módulo CDT irá determinar por meio da execução de algoritmos heurísticos o transportador responsável pela atividade de carregamento e de descarregamento de peças entre os recursos baseado no critério de menor distância percorrida entre a posição atual do transportador e do recurso solicitante da atividade de transporte. Após a designação do transportador, o módulo CDT retorna ao processo solicitante a informação de que o transporte será atendido, representado pelo fluxo (5) e informa ao módulo CUV o transportador que foi designado para a atividade de transporte, representado pelos fluxos (7) e respectiva resposta pelo fluxo (8).

O fluxo (9) indica a atualização das marcas. A comunicação do módulo de controle CUV com o CV é indicada pelos fluxos (9) e (10) que representam a requisição para trânsito em vias e as respectivas autorizações. Uma vez concluída a utilização do recurso e/ou do transportador, ele ficará disponível para a próxima alocação.

A figura 22 ilustra o *layout* de exemplo de um SP, neste exemplo o *layout* é composto de uma entrada de itens a produzir (ENTRADA), uma saída de itens produzidos (SAÍDA) e um estacionamento de transportadores. As vias desse exemplo possuem sentido único e indicadas por meio das setas. Existem vias exclusivas para as atividades de carregamento e descarregamento nas estações de trabalho com a finalidade de não obstruir as vias principais.

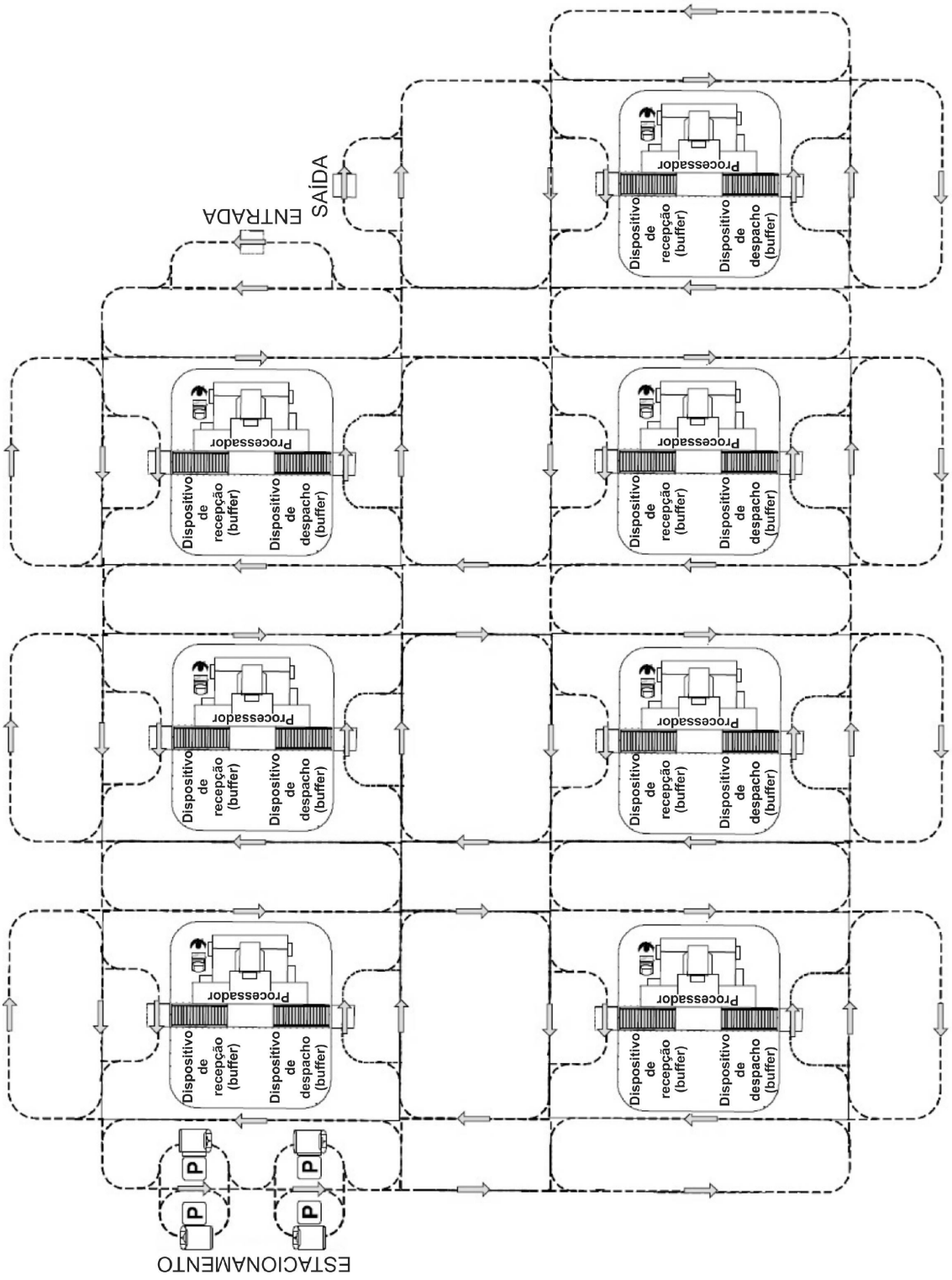


Figura 22 – *Layout* de exemplo de um SP.
(Adaptado de NAKAMOTO, 2008)

5.3 Método para Implementar os Módulos do Sistema de Alocação de Recursos

Para a implementação dos módulos do sistema da alocação de recursos, é necessário gerar 3 grafos: Grafo Sequência de Funcionalidades (GSF), Grafo para o Controle de Processos (GCP) e Grafo para a Designação de Funcionalidade (GDF).

Para iniciar a especificação das funções de controle da alocação de recursos, é necessário gerar primeiramente a sequência de recursos fornecida pelo setor de planejamento e converter os recursos em funcionalidades, essa conversão é denominada como GSF. A partir do GSF derivam os grafos GDF e GCP.

A figura 23 ilustra o procedimento para a síntese dos grafos do sistema de controle da alocação de recursos.

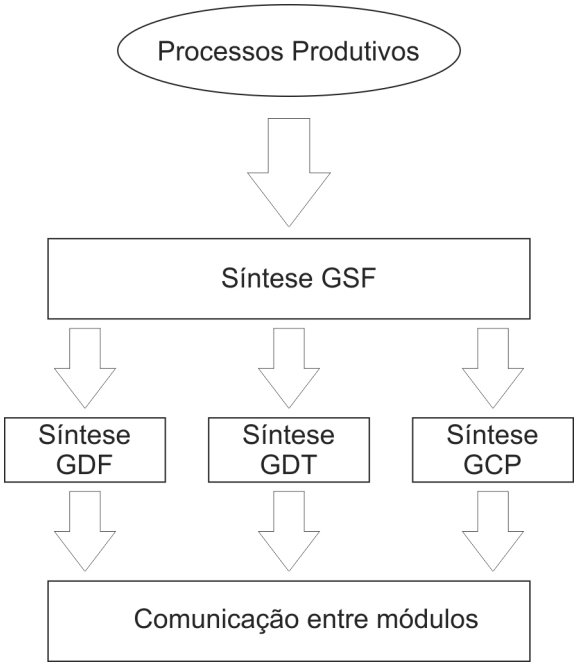


Figura 23 - Procedimento para síntese dos grafos do sistema de controle de alocação de recursos. (Adaptado de ASATO, 2015)

5.4 Síntese dos Grafos

Etapa 1: Síntese do Grafo Sequência de Funcionalidade (GSF)

Por meio da sequência de recursos (*scheduling*) estabelecida pelo planejamento da produção, é obtido o grafo de sequência de funcionalidades (GSF). Na figura 24, apresenta-se um exemplo descrito por meio da ferramenta de modelagem E-MFG (*Enhanced-Mark Flow Graph*) de três processos com a sequência de recursos utilizados para a produção de produtos.

Na qual:

i) Entrada: representa a entrada de matéria-prima;

ii) Para i≥1,...,N-1 têm-se que Ai representa a ação do evento (transição) que indica o início de utilização do recurso Ri;

iii) Para i=N têm-se que Ai representa o evento que finaliza a utilização dos recursos para realização de um determinado processo que envolve até o recurso Ri-1;

iv) Saída: representa a saída do produto.

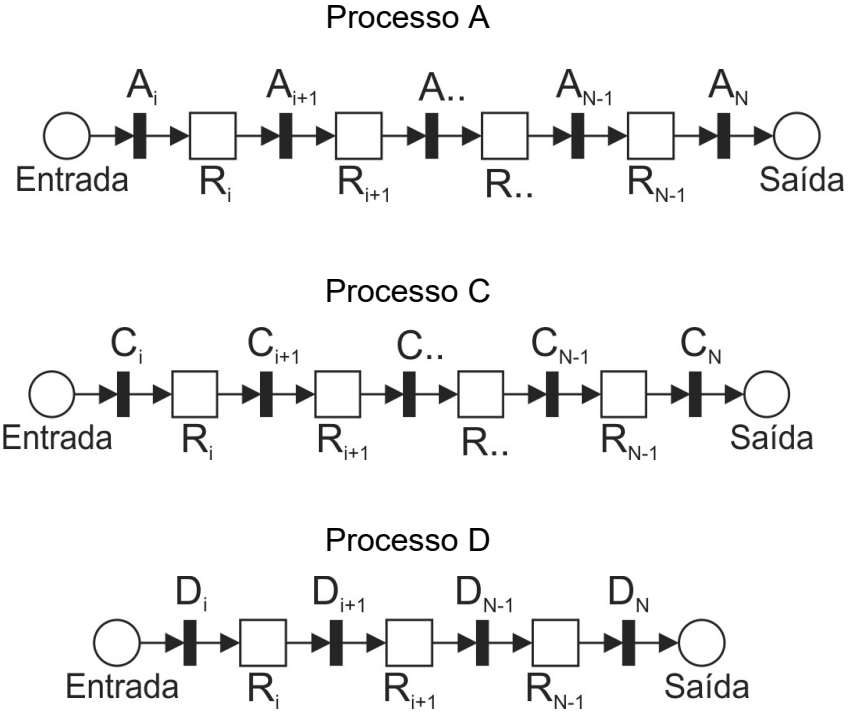


Figura 24 - Exemplo de sequência de recursos de três processos.
(Adaptado de KUBO *et al.,* 2016).

Dessa forma, com a conversão do grafo de sequência de recursos no qual cada recurso é convertido pela respectiva funcionalidade, é possível obter o GSF.

A figura 25 ilustra a conversão da sequência de recursos dos processos em uma sequência de funcionalidades.

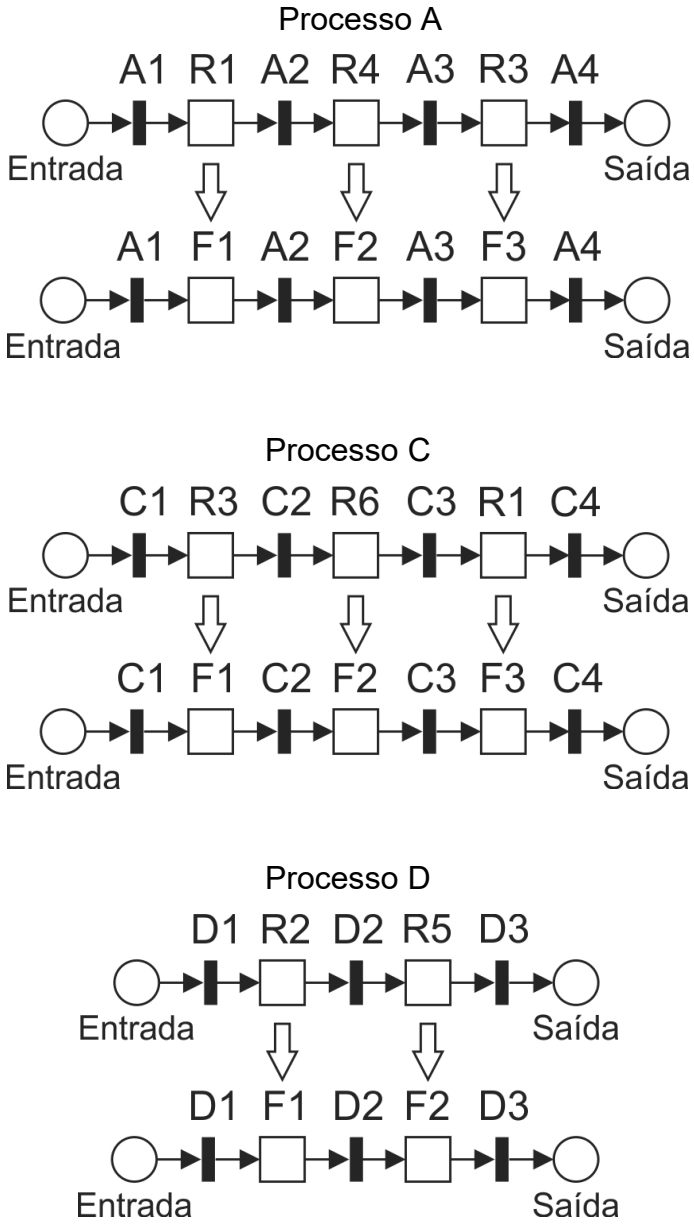


Figura 25 - Exemplo da síntese do GSF a partir da sequência de recursos.
(Adaptado de KUBO *et al.,* 2016).

Etapa 2: Síntese do Grafo para o Controle do Processo (GCP)

Para a síntese do grafo, deve-se considerar o grafo GSF e cada funcionalidade identificada entre as atividades A_i e A_{i+1}, conforme ilustrado na figura 25. Realiza-se um refinamento sucessivo do E-MFG gerado a partir do grafo GSF, substituindo o *box* “Fi” por uma sequência de *boxes*, com a seguinte semântica associada a cada um deles: requisita funcionalidade, requisita transporte e executa atividade.

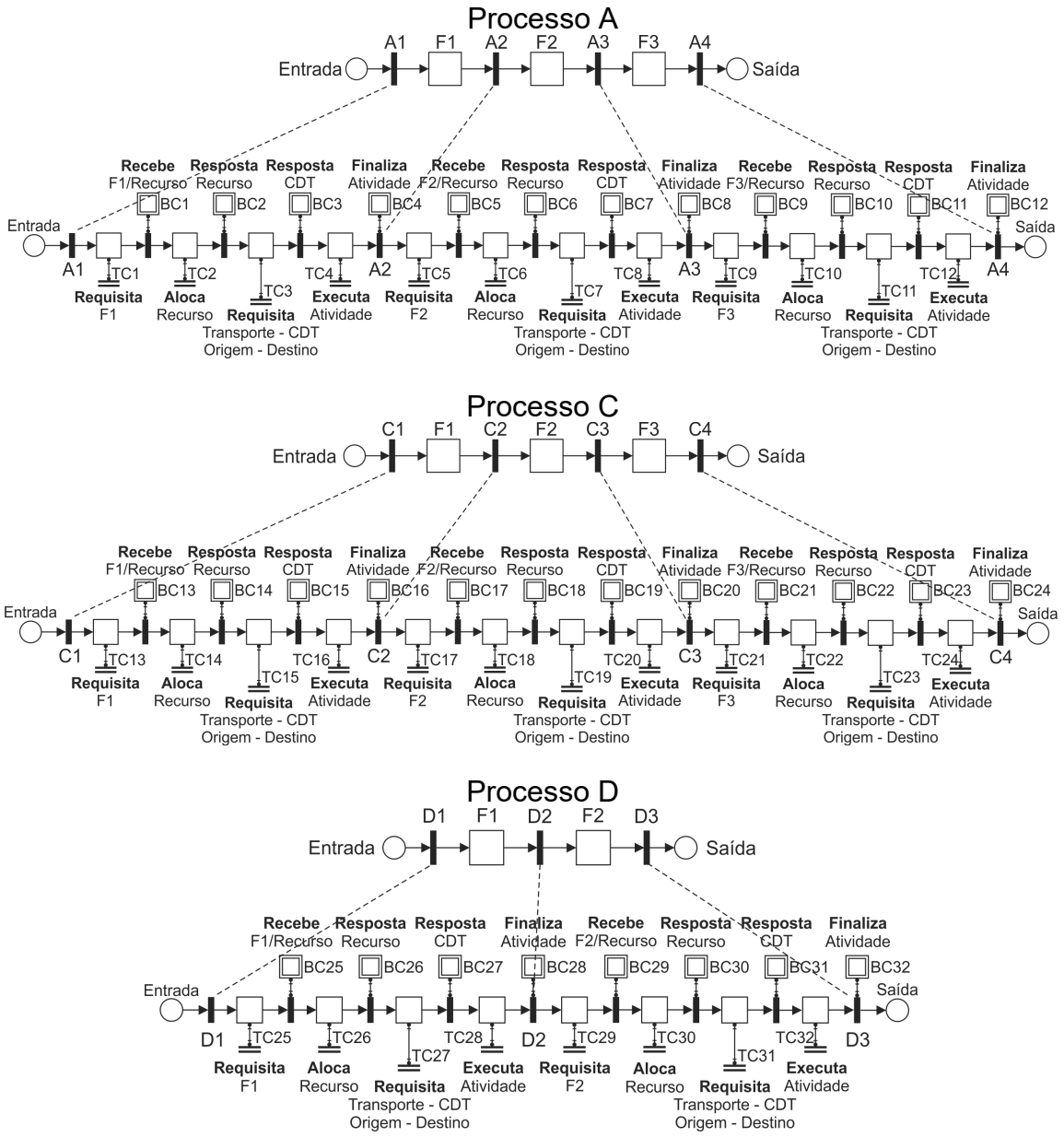


Figura 26 - Grafo de controle do GCP derivado do GSF.

A figura 27 ilustra o refinamento do elemento Executa Atividade. Essa atividade é executada pelo recurso após o transporte da peça até a estação de trabalho. O *box* B34 indica que o processo aguarda o transporte. Por meio do *box* de comunicação BC59, o recurso informa que o transportador iniciou a atividade de carregamento. A evolução da marca para o *box* B36 indica que a peça está sendo transportada para a estação de trabalho destino. A estação de trabalho destino informa por meio do *box* de comunicação BC60 o início da atividade de descarregamento da peça. Em seguida o item aguarda no dispositivo de recepção da estação de trabalho a execução da atividade de transformação, indicada pelo *box* BC61. O recurso informa o módulo CP por meio do *box* de comunicação BC62 a conclusão da atividade de transformação e o item é colocado no dispositivo de despacho da estação de trabalho. O *box* controlador BCT8 atualiza os atributos (recurso atual e próximo recurso).

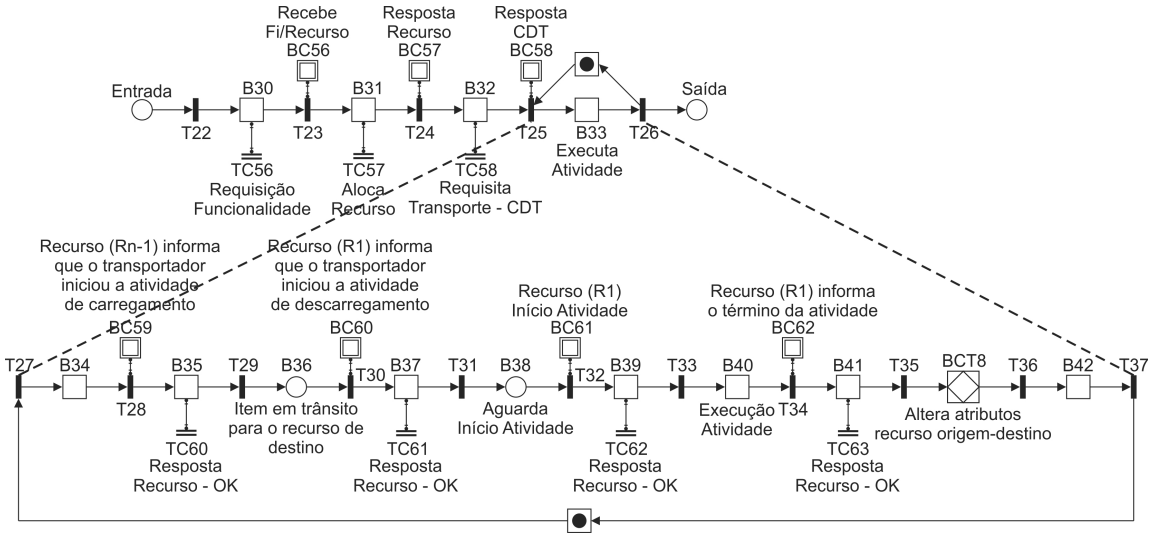


Figura 27 – Refinamento sucessivo – Execução da atividade.
(Adaptado de NAKAMOTO, 2008)

Etapa 3: Síntese do Grafo para a Designação de Funcionalidade (GDF)

Para a síntese do GDF é necessário realizar o refinamento e a fusão dos *boxes* controladores de mesmo nome. Para o refinamento do processo A, por exemplo, são consideradas as funcionalidades descritas no módulo GSF entre as transições A_i e A_{i+1}; A_{i+1} e A_{n-1}; A_{n-1} e A_n. Com isso, o refinamento é realizado por meio da inclusão de um *box* de entrada (*buffer* de entrada), um *box* controlador (representação da funcionalidade) e de um *box* de saída (*buffer* de saída).

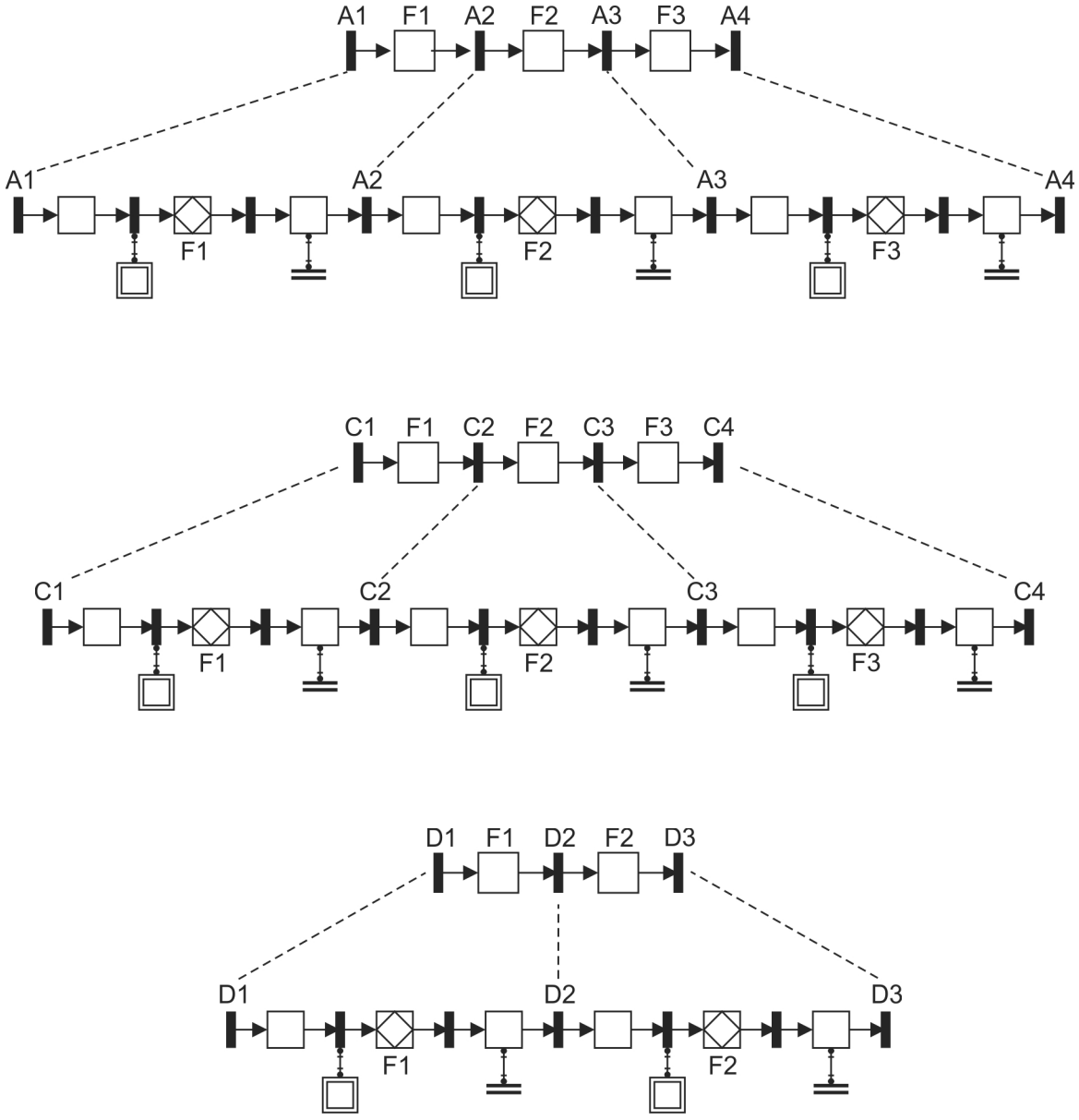


Figura 28 - Refinamento das funcionalidades no módulo GSF.
(Adaptado de KUBO *et al.,* 2016)

Concluída a etapa de refinamento das funcionalidades do módulo GSF, é necessário incluir um conjunto de *boxes* controladores e um conjunto de *boxes* capacidade. Os *boxes* controladores P1, P2 e P3 serão responsáveis pela atualização do atributo *atrib1* da marca, identificando o processo que realizou a requisição da funcionalidade ao módulo GSF, os *boxes* capacidade B1, B2 e B3 não são de capacidade unitária e serão responsáveis em organizar as requisições em uma estrutura FIFO (*first in*, *first out*) com o recebimento das requisições pelos *boxes* de recepção. O *box* capacidade B1 organiza as requisições recebidas pelos *boxes* de recepção BC33, BC36 e BC39, o *box* capacidade B2 organiza as requisições recebidas pelos *boxes* de recepção BC34, BC37 e BC40, o *box* capacidade B3 organiza as requisições recebidas pelos *boxes* de recepção BC35 e BC38.

A figura 29 ilustra o refinamento das funcionalidades no módulo GSF com a inclusão dos *boxes* controladores P1, P2 e P3 e do *box* capacidade B1.

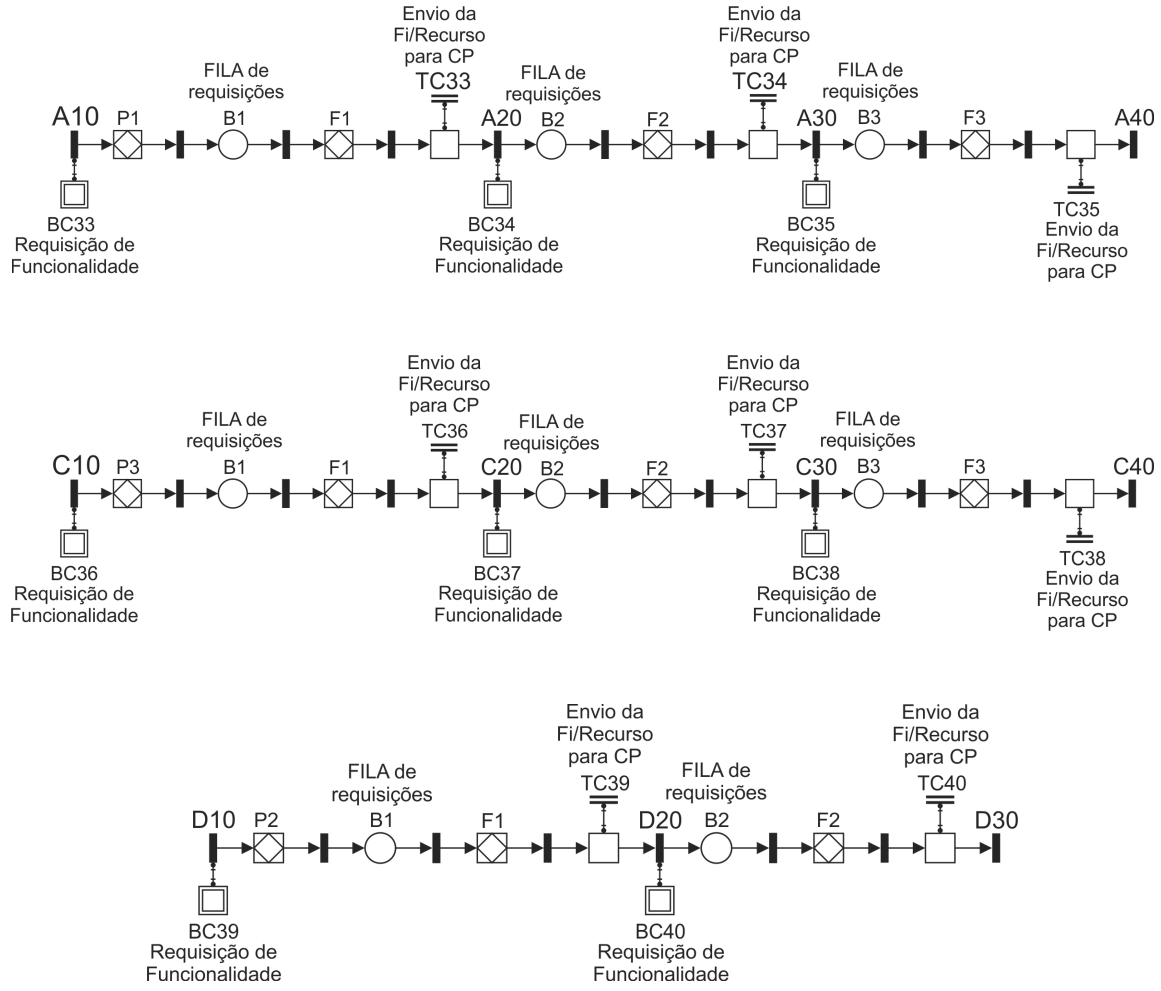


Figura 29 –Inclusão do *box* controlador e do *box* capacidade no módulo GDF.

Tabela 2 - Atributos da marca no módulo GDF.

Atributo	Descrição	Valor
atrib1	Identificador do processo	Inteiro
atrib2	Funcionalidade	Inteiro
atrib3	Recurso	Inteiro

Após a inclusão do *box* controlador P1, P2 e P3 e dos *boxes* capacidade B1, B2 e B3, integra-se os grafos GSF por meio da fusão dos *boxes* capacidade de mesmo nome e dos *boxes* controladores de mesma funcionalidade. Com isso, a designação do recurso mais adequado para substituir um recurso inoperante do SP é realizada por meio de um algoritmo heurístico implementado no módulo DR.

Dessa forma, inicialmente o módulo GSF receberá as requisições de funcionalidades por meio dos *boxes* de recepção BC33, BC36 ou BC39. Os *boxes* controladores P1, P2 e P3 irão atualizar o atributo *atrib1* da marca identificando o processo que realizou a requisição da funcionalidade. O *box* capacidade B1 irá organizar todas as requisições de funcionalidades em uma estrutura FIFO (*first in, first out*) para que o atendimento das requisições de funcionalidades seja de acordo com a ordem de recebimento por meio dos *boxes* de recepção BC33, BC36 e BC39.

A figura 30 ilustra a fusão dos *boxes* capacidade de mesmo nome e dos *boxes* controladores de mesma funcionalidade.

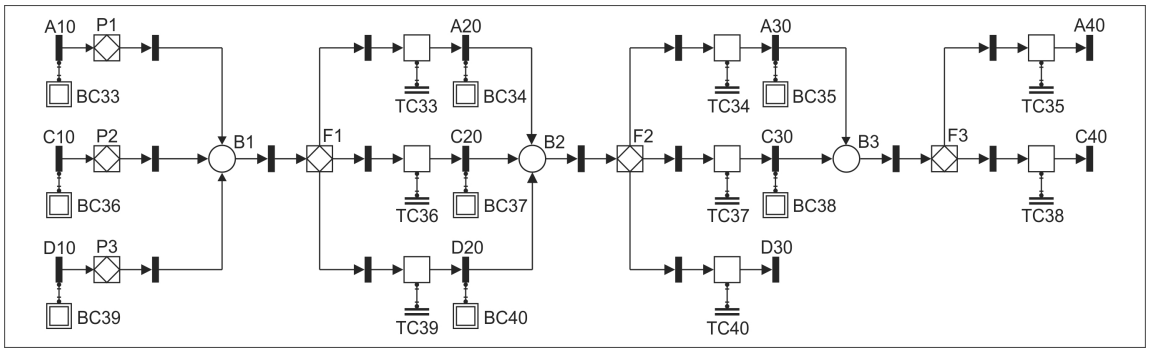


Figura 30 - Exemplo de fusão dos *boxes* controladores.

Etapa 4: Síntese do Algoritmo para Designar o Recurso (DR)

A designação do recurso pelo módulo DR é um algoritmo heurístico para tomada de decisão, ou seja, na ocorrência de um evento inesperado como falha do recurso, o algoritmo realiza a busca de outro recurso a ser alocado de acordo com a funcionalidade do recurso a ser realocado e manter o processo produtivo (ASATO, 2015).

5.5 Síntese do Grafo de Designação de Transporte (DT)

Etapa 1: Síntese do Grafo Controle de Designação de Transporte (CDT)

A síntese do módulo CDT é baseada na metodologia PFS/E-MFG e realizada em três atividades denominadas: atividade de atualização dos estados dos transportadores, atividade de organização da fila de requisições e atividade de designação do transportador.

A figura 31 ilustra a metodologia PFS/E-MFG.

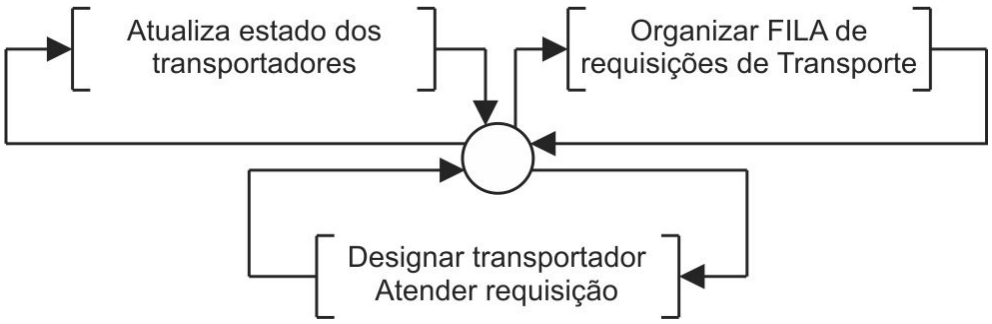


Figura 31 - Metodologia PFS/E-MFG para a síntese do CDT.
(Adaptado de NAKAMOTO, 2008)

Para a geração do E-MFG deve-se considerar os atributos da marca. No módulo CDT a marca possui oito atributos conforme descrito na tabela 3.

Tabela 3 - Atributos da marca no módulo CDT.

Atributo	Descrição	Valor
atrib1	Identificação do retorno de mensagem	Inteiro
atrib2	Identificador do processo	Inteiro
atrib3	Recurso Origem	Inteiro
atrib4	Recurso Destino	Inteiro
atrib5	Identificador do transportador VGA	Inteiro
atrib6	Fila de requisições de transporte	Ponteiro
atrib7	Lista de transportadores disponíveis	Ponteiro
atrib8	Trajeto de transporte	GAR

A figura 32 ilustra o E-MFG gerado a partir do PFS da figura 31. O *box* de recepção BC50 recebe a requisição de transporte do módulo CP e a marca evolui para o *box* capacidade B19, o *box* B19 não é de capacidade unitária. Destaca-se que as marcas nos *boxes* capacidades são armazenadas em uma estrutura FIFO (*first in, first out*).

O atendimento é iniciado se houver transportador disponível (*atrib7*), representado pelo disparo da transição T4. O *box* B21 unifica as informações de requisições e dos transportadores em uma única marca, o arco que conecta o box B20 e a transição T4 filtra os atributos *atrib7* e *atrib8*, enquanto que o arco que une B25 e T4 filtra os atributos *atrib1*, *atrib2*, *atrib3*, *atrib4*, *atrib5* e *atrib6*.

A designação do transportador e a definição do trajeto é realizada pelo *box* controlador BCT2 que adota o critério de menor caminho baseado no algoritmo de *Dijkstra* com *Heap Binomial* (ANEXO). Na designação do transportador são atualizados dois atributos *atrib5* e *atrib8*, que são respectivamente, a identificação do transportador e o trajeto de transporte. A marca no *box* B22 dispara as transições de comunicação TC49 e TC50 que enviam uma resposta ao módulo CP e a ordem de transporte ao CUV, respectivamente. Será enviada uma resposta ao módulo CP referente ao pedido de transporte recebido pelo módulo CDT, que somente após a efetiva designação, poderá informar ao módulo CP que o pedido de transporte será atendido. O módulo CUV recebe do CDT as informações referentes à designação: transportador, estação de trabalho origem, estação de trabalho destino e o trajeto (*atrib3*, *atrib4*, *atrib5* e *atrib8*).

A marca no *box* B23 informa ao transportador que foi designado pelo CDT a sua alocação e aguarda uma resposta de confirmação do transportador (*atrib7*) pelo *box* de recepção BC52, após a confirmação do transportador, o *box* controlador BCT3 executa a cópia dos dados do primeiro da fila para os atributos *atrib1*, *atrib2*, *atrib3* e *atrib4*. Esta cópia é necessária para o módulo CDT possa sempre atender o primeiro pedido da fila.

O transportador informa o módulo CDT o término da atividade de transporte que lhe foi designado por meio de mensagem enviada ao *box* BC53 para que se torne disponível para a próxima designação (*atrib7*). A atualização da disponibilidade dos transportadores é realizada pelo *box* BCT4 que atualiza o *atrib7* referente a lista de transportadores disponíveis. Este módulo é independente do SP, constituindo-se em um módulo de controle padrão para designação dos transportadores.

Uma evolução implementada nesse modelo se refere a inclusão de mais uma condição na regra de início de designação do transportador, a condição implementada é que não existir nenhuma mensagem de disponibilidade de transportador no *box* capacidade B24, garantindo dessa forma que a lista de transportadores disponíveis esteja sempre atualizada antes de iniciar a designação do transportador.

Nesse modelo as três condições para que a transição T4 seja habilitada são: não existir mais requisições de transporte no *box* capacidade B20, não existir mensagem de disponibilidade de transportadores no *box* capacidade B24 e o valor do atributo *atrib7* ser True.

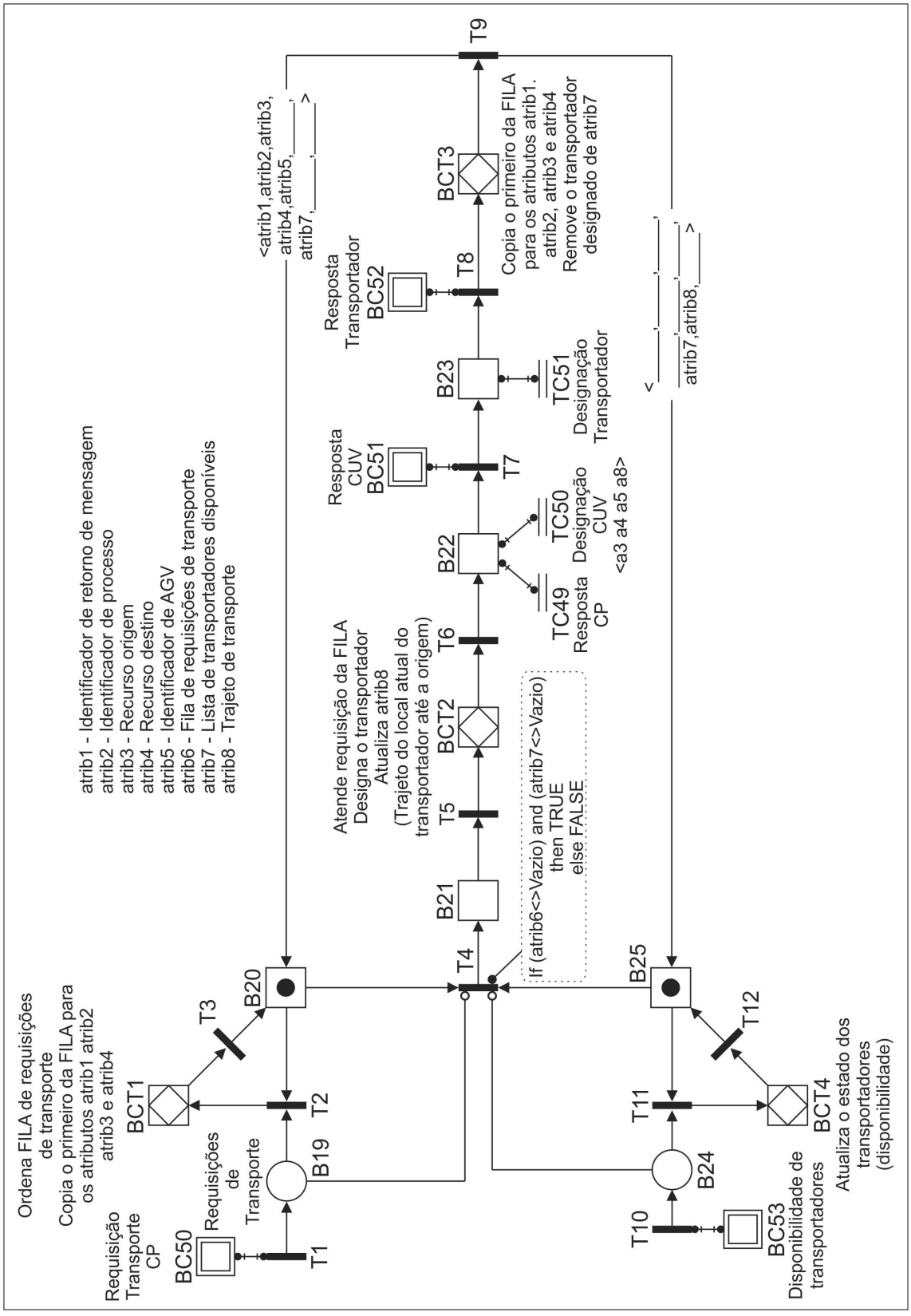


Figura 32 – E-MFG do módulo CDT.
(Adaptado de NAKAMOTO, 2008)

Etapa 2: Síntese do Grafo Controle de Utilização de Vias (CUV)

A síntese do módulo CUV é baseada na metodologia PFS/E-MFG na qual a ordem de transporte enviada pelo módulo CDT é inserida na fila de designações, que altera o atributo *atrib7* indicando a necessidade de atualizar o módulo controle CUV.

A figura 33 ilustra o PFS do módulo gerador.

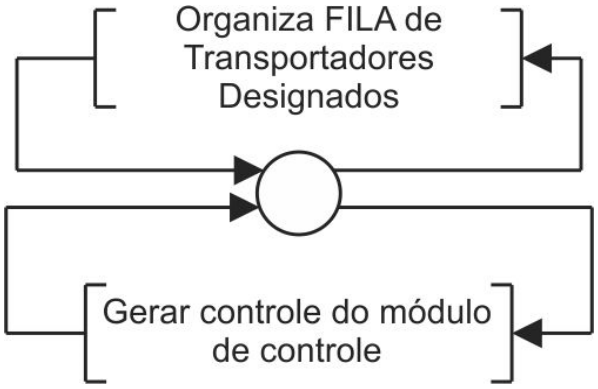


Figura 33 - PFS do módulo gerador.
(Adaptado de NAKAMOTO, 2008)

Para a geração do E-MFG deve-se considerar os atributos da marca. No módulo CUV a marca possui sete atributos conforme descrito na tabela 4.

Tabela 4 - Atributos da marca no módulo CUV.

Atributo	Descrição	Valor
atrib1	Identificador do transportador	Inteiro
atrib2	Recurso Origem	Inteiro
atrib3	Recurso Destino	Inteiro
atrib4	Trajeto de transporte	GAR
atrib5	Término do trajeto	Boleano
atrib6	Fila de transportadores	Ponteiro
atrib7	Atualiza controle	Boleano

O módulo CUV deve ser atualizado sempre que receber um novo pedido de transporte ou quando o transportador percorrer o trajeto 1. A atualização do controle CUV é realizada pela regra da transição T17 que verifica o valor do *atrib7*. O *box* controlador BCT7 executa um algoritmo que cria o E-MFG com as regras adicionais de controle para evitar o *deadlock* nas vias.

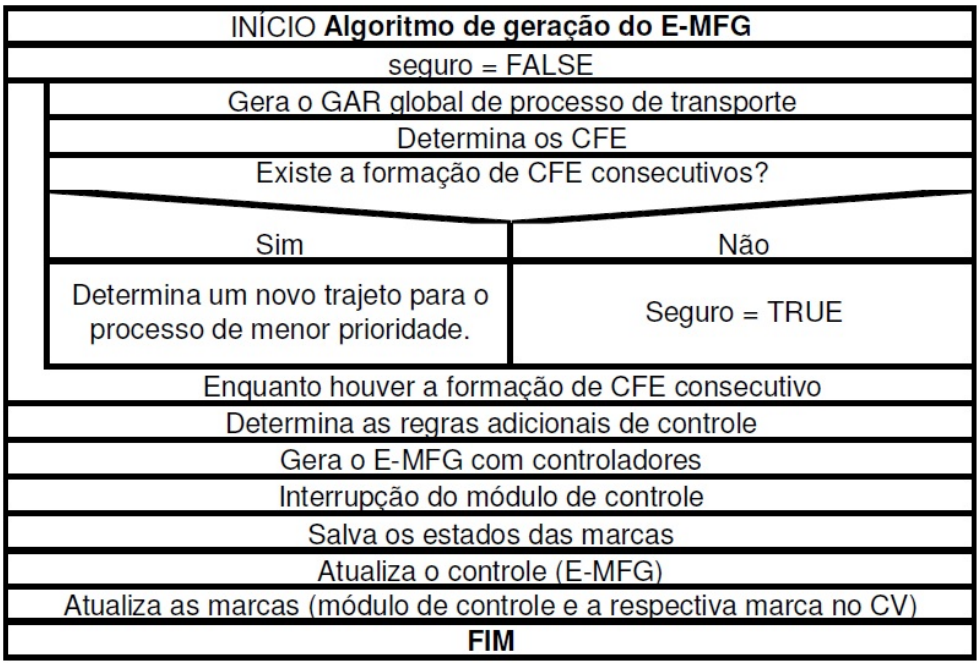


Figura 34 – Algoritmo para geração do módulo controle.
(NAKAMOTO, 2008)

Em caso de formação de CFE consecutivo, o algoritmo calcula um novo trajeto baseado no algoritmo executado pelo CDT para o transporte de menor prioridade. Nesse caso, a maior prioridade será sempre o primeiro da fila de ordenação das requisições de transporte e o de menor prioridade será sempre o último da fila de ordenação das requisições de transporte. Concluída a identificação dos CFEs consecutivos, o algoritmo determina as regras adicionais de controle para evitar o *deadlock*. As prioridades das designações devem ser contempladas nas regras adicionais de controle, desta forma, em caso de conflito de transportadores para o tráfego na via, a preferência será a designação de maior prioridade, desde que sejam respeitadas as regras para evitar o travamento.

O módulo CUV recebe a requisição do transportador para o tráfego da via (CV) e, de acordo com as regras adicionais de controle, permite ou não o tráfego. Destaca-se que o atributo da marca do CUV é composto apenas pelo identificador do transportador e as atividades de carregamento e descarregamento realizadas pelo transportador, não será de responsabilidade dos módulos CUV e CV. Essas atividades são tarefas do próprio controlador do transportador em comunicação com a estação de trabalho.

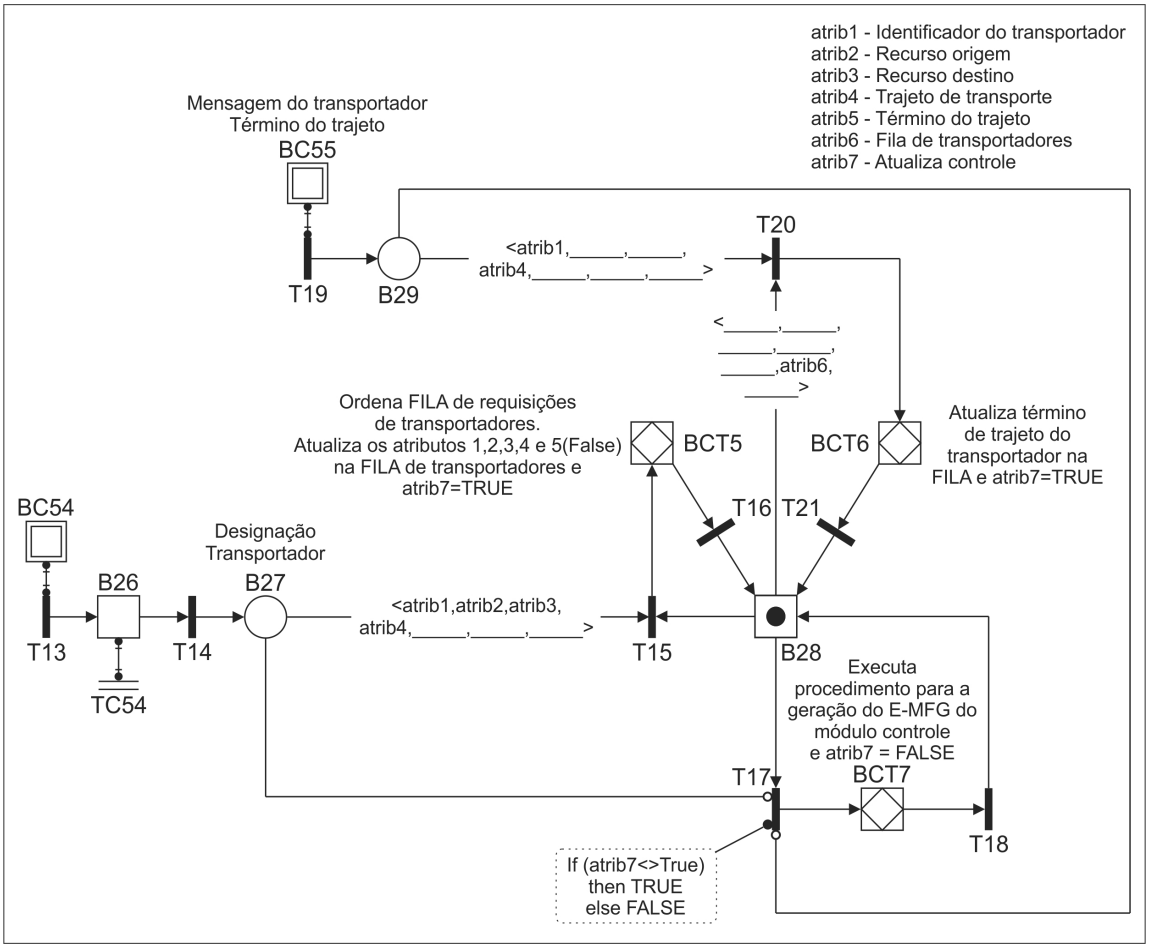


Figura 35 – E-MFG do módulo gerador CUV.
(Adaptado de NAKAMOTO, 2008)

A mensagem de designação do transportador enviada pelo CDT ao módulo CUV é recebida pelo *box* BC54 no CUV. O disparo da transição T13 no CUV gera uma marca com as informações da mensagem no *box* B26 que envia uma mensagem de retorno ao CDT pela transição de comunicação TC54. A seguir a marca evolui para o *box* capacidade B27, o *box* B27 não é de capacidade unitária.

Destaca-se que as marcas nos *boxes* capacidades são armazenadas em uma estrutura FIFO (*first in, first out*). Se a marca com a lista de designação de transportador estiver disponível no *box* B28, então é realizada a ordenação da fila de designações e a alteração do atributo *atrib7* para *True* afim de atualizar o módulo de controle CUV. O disparo da transição T17 permite a evolução da marca para o *box* controlador BCT7 que executa um algoritmo, gerando-se assim o módulo de controle do CUV. Após a execução do algoritmo, o atributo *atrib7* é alterado para *False*.

A mensagem de retorno do CUV enviada pela transição de comunicação TC54 é recebida pelo *box* BC51 do CDT, habilitando-se desta forma a transição T7. A seguir, é realizado o procedimento de designação do transportador e a resposta do mesmo, por meio de ações que são realizadas pelos *boxes* B23 e BC52.

Uma evolução implementada nesse modelo se refere a inclusão de mais uma condição na regra de início do procedimento para geração do E-MFG do módulo de controle, a condição implementada é que não existir nenhuma mensagem do transportador comunicando o término do trajeto por meio do *box* capacidade B29, garantindo dessa forma, que a fila de transportadores esteja sempre atualizada antes de iniciar o procedimento para geração do E-MFG do módulo de controle.

Etapa 3: Síntese do Grafo Controle de Vias (CV)

O controle de tráfego é originado do layout do SP baseada na metodologia GAR/E-MFG. Todas as marcas do E-MFG do controle de tráfego gerado representam os transportadores do SP e seu atributo é o trajeto a ser percorrido. No layout proposto, as zonas de carregamento e de descarregamento nas estações de trabalho são distintas, possuindo uma via exclusiva para este fim. Com isso, as vias principais do SP ficam livres para o tráfego dos transportadores.

O GAR das vias possui as seguintes características:

- Todos os pontos de carregamento, descarregamento, estacionamento, estações de entrada, estações de saída e pontos de parada são identificados como nós principais;
- Todos os nós de cruzamento de vias são identificados;
- Todos os nós são conectados com arcos direcionais que representam as vias.

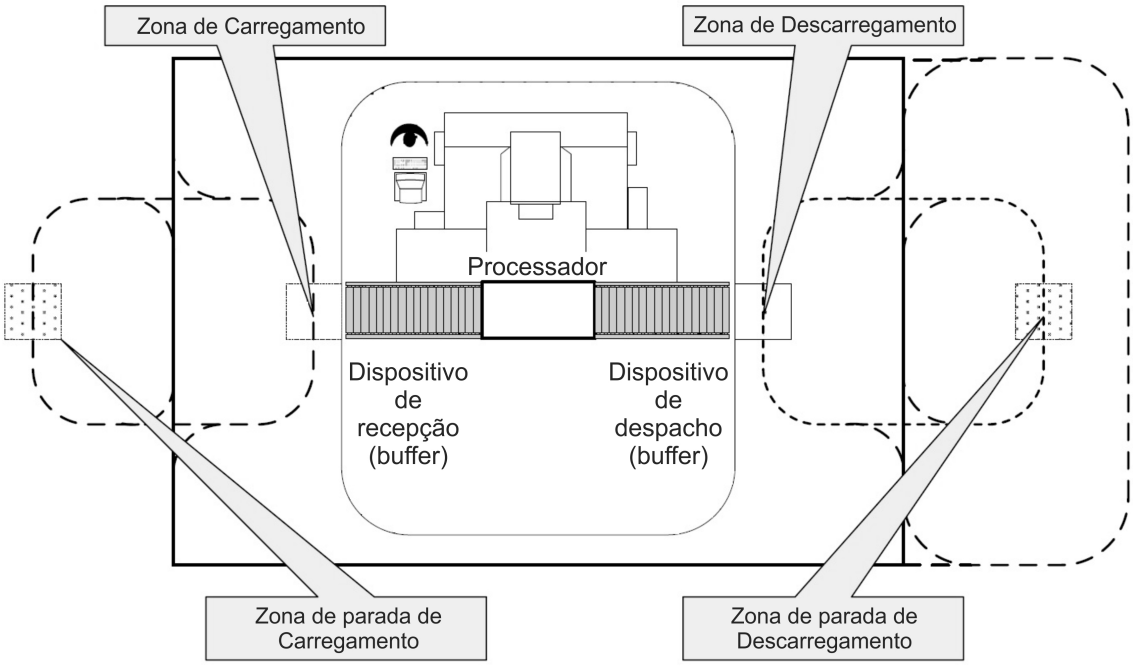


Figura 36 – Zonas de carregamento/descarregamento e zona de parada.
(Adaptado de NAKAMOTO, 2008)

Os atributos da marca do módulo CV são apresentados na tabela 5.

Tabela 5 - Atributos da marca no módulo CU.

Atributo	Descrição	Valor
atrib1	Identificador de transporte	Inteiro
atrib2	Estado	Inteiro
atrib3	Trajeto	GAR do trajeto

O mapeamento do GAR das vias é apresentado na figura 37, na figura 38 é apresentado o E-MFG gerado a partir do GAR das vias mediante a metodologia GAR/E-MFG. O E-MFG gerado possui como característica todos os *boxes* de capacidade unitária, garantindo assim que somente um transportador estará utilizando determinada zona. A zona de controle é composta por um nó e o arco orientado e seu acesso só é permitido com a autorização do módulo CUV. A marca no módulo CV representa o transportador que possui como atributo o trajeto a ser percorrido que é composto por uma sequência de zonas.

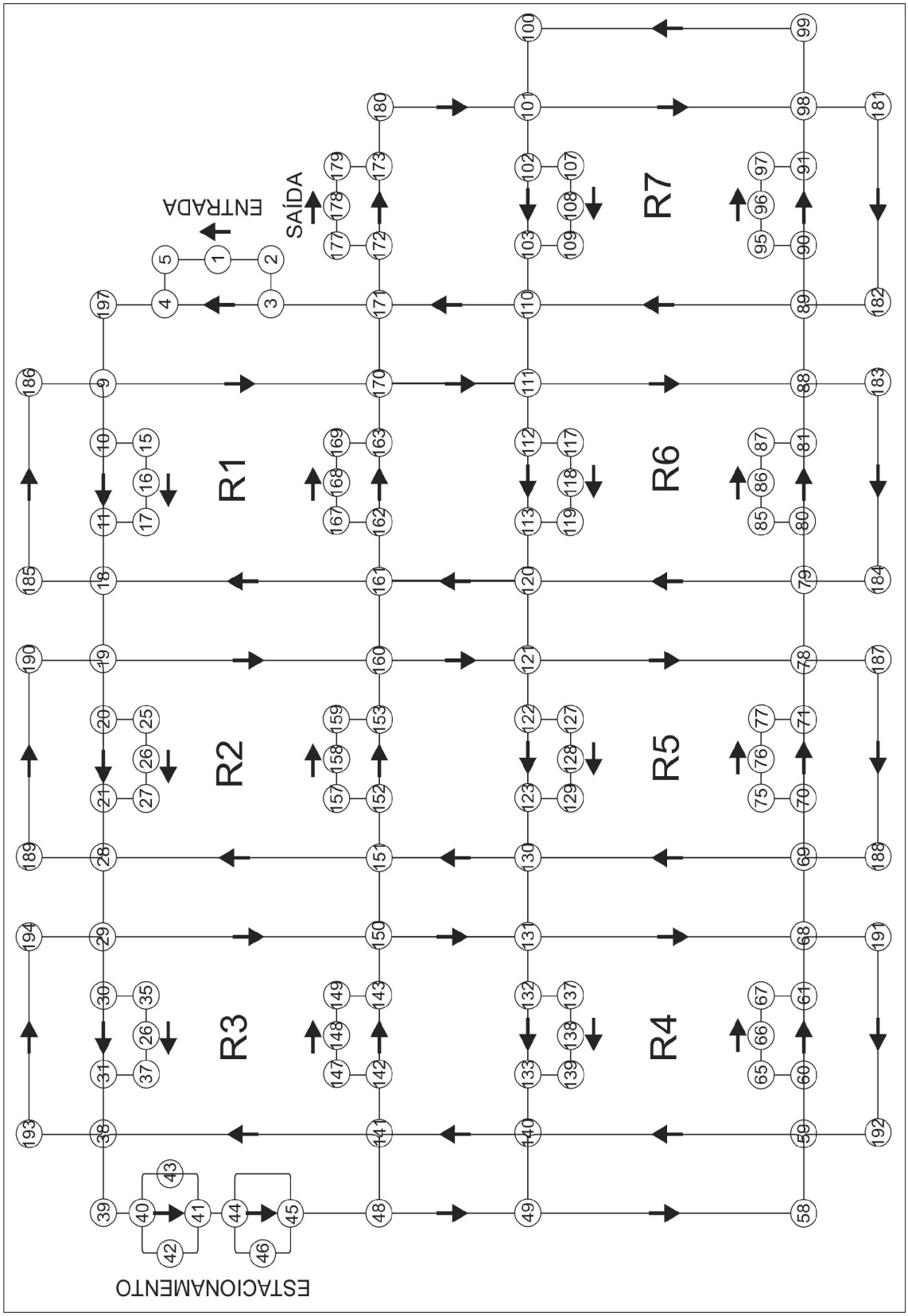


Figura 37 – Grafo GAR das vias.
(Adaptado de NAKAMOTO, 2008)

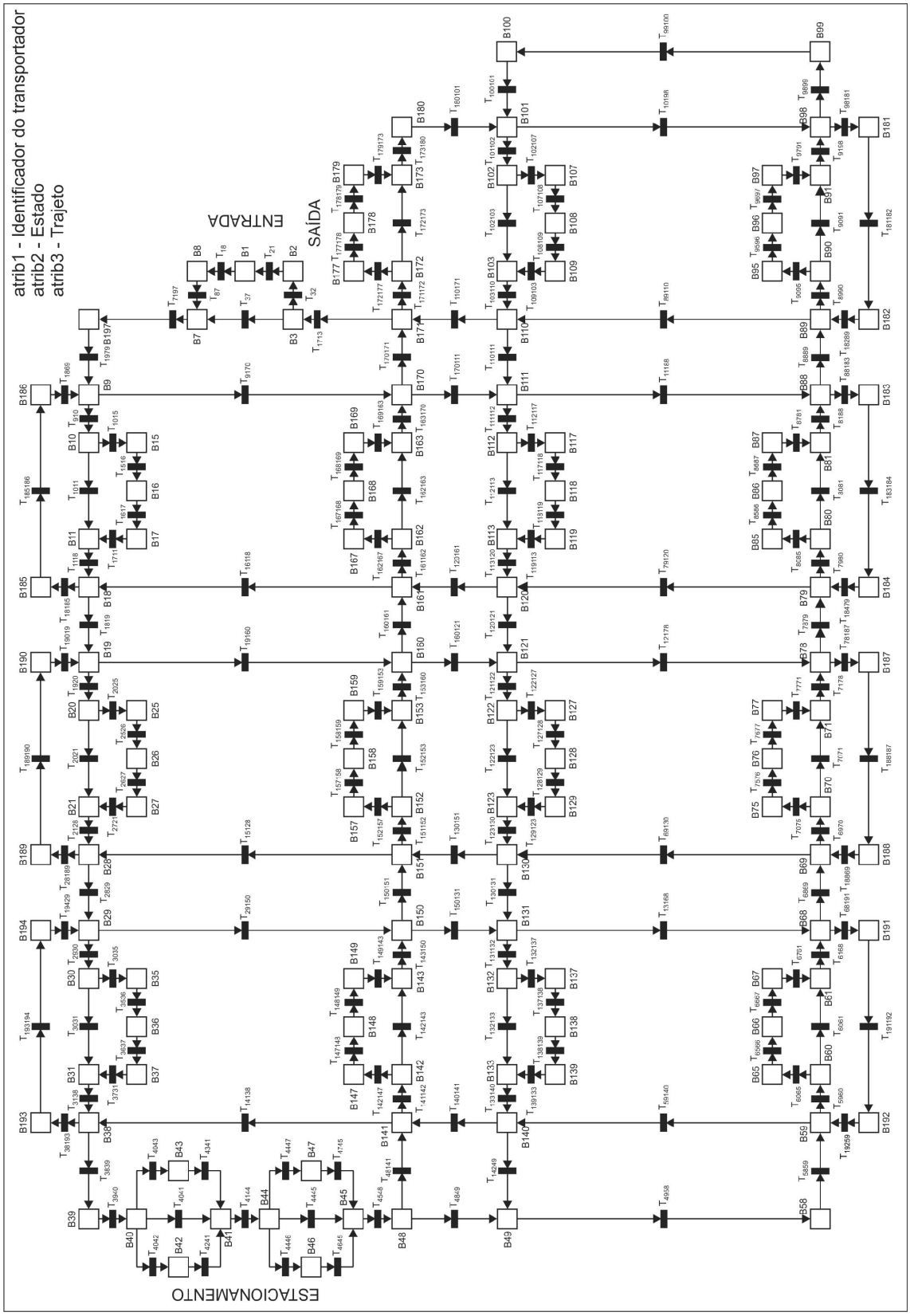


Figura 38 – E-MFG do controle de vias.
(Adaptado de NAKAMOTO, 2008)

5.6 Comunicação entre os módulos

Após a geração dos grafos: GSF, GCP e GDF para facilitar o entendimento da forma que é realizada a comunicação entre os módulos com os processos, foi gerada a ilustração da figura 39, na qual as trocas de informações entre os módulos são realizadas por meio dos arcos de envio e dos arcos de recepção dos elementos comunicadores.

O disparo na transição A1 do módulo GCP faz com que a marca evolua para o *box* iniciando a requisição da funcionalidade (envio de mensagem ao módulo GDF por meio da transição de comunicação TC1) afetada pela falha ou indisponibilidade do recurso no chão de fábrica. O módulo GDF recebe a requisição de funcionalidade por meio do *box* de comunicação BC33, que habilita a transição A10 permitindo a evolução da marca para o *box* controlador P1. O *box* controlador P1 irá atualizar o atributo *atrib1* da marca para identificar o processo que originou a requisição da funcionalidade. Após a atualização do *atrib1*, a marca evolui para o box capacidade B1 que irá organizar em FIFO *(first in, first out)* as requisições de funcionalidades. O *box* controlador F1 irá processar a marca por meio de um algoritmo heurístico que irá identificar o recurso mais apropriado para o atendimento da requisição de funcionalidade e enviará uma resposta por meio da transição de comunicação TC33 para o *box* de comunicação BC1, que habilita a transição e a marca evolui para o *box* iniciando o envio da mensagem por meio da transição de comunicação TC2 para alocar o recurso. O recurso retorna uma mensagem ao módulo GCP confirmando sua alocação e a marca evolui para o *box* que irá requisitar o transporte por meio da transição de comunicação TC3.

O módulo CDT recebe a requisição de transporte por meio do *box* de comunicação BC50, nesse momento é gerada uma marca com os atributos da mensagem no *box* capacidade B19 e esta é armazenada de forma sequencial por ordem de chegada. A transição T27 só poderá ser habilitada após o atendimento de duas condições que são, requisição da atividade de transporte e disponibilidade de transportadores.

A marca resultante no *box* B25 possui informações sobre a lista de transportadores disponíveis. Na sequência o *box* B21 unifica todas essas informações em uma única marca, a marca evolui para o *box* controlador BCT2 que

irá designar o transportador pelo critério de menor caminho. A designação do transportador atualiza os atributos *atrib5* e *atrib8,* que se referem ao identificador do transportador e o trajeto respectivamente.

A marca evolui para o *box* B22 que irá enviar uma mensagem por meio das transições de comunicação TC49 e TC50 respectivamente para os módulos GCP e CUV. A mensagem para o módulo GCP será o identificador de retorno de mensagem e para o módulo CUV será a designação do transportador com as informações de identificador de transportador, recurso de origem, recurso de destino e o trajeto.

A mensagem de designação enviado do módulo CDT para o módulo CUV é recebida pelo *box* de comunicação BC54. Com o disparo da transição T37, a marca evolui para o *box* B27 que irá enviar uma mensagem de retorno ao módulo CDT por meio da transição de comunicação TC16. Em seguida a marca evolui para o *box* B28 e aguarda a disponibilidade da marca com a lista de designação dos transportadores para iniciar a ordenação da fila de designações e atualizar o *atrib7* para *TRUE* (para atualizar o módulo de controle do CUV). O disparo da transição T17 irá evoluir a marca para o box controlador BCT7 que irá executar um algoritmo para gerar o módulo de controle CUV.

Após a execução do algoritmo, o atributo *atrib7* é alterado para *FALSE*. A mensagem de retorno do módulo CUV é recebida pelo *box* BC51 habilitando a transição T7, dando início ao procedimento de designação do transportador além da sua resposta por meio dos *boxes* B23 e BC52. O *box* controlador BCT3, remove o transportador designado do *atrib7* da marca que retorna ao *box* B20 aguardando uma nova designação de transportadores.

A mensagem de retorno enviada pela transição TC49 para o GCP é recebida pelo *box* de comunicação BC3 e a partir desse momento, o processo aguarda a realização das atividades de carregamento, do transporte e do descarregamento pelo transportador designado. O início e término da atividade no recurso são verificados pela comunicação entre o GCP e o recurso.

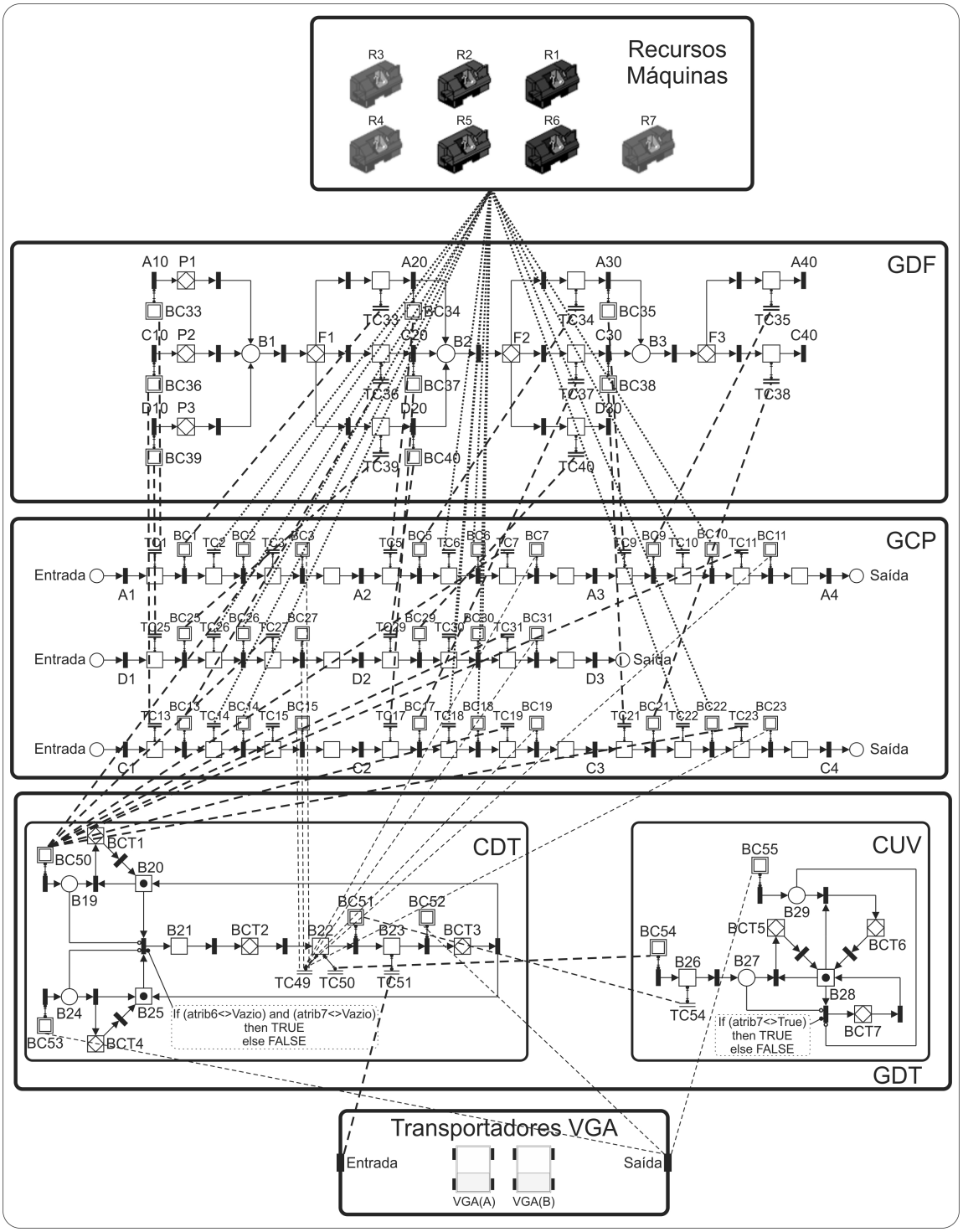


Figura 39 - Ilustração da comunicação entre os módulos.

5.7 Execução dos Módulos

Para a execução dos módulos são consideradas as seguintes condições iniciais:

- O módulo CDT está em estado de espera aguardando a entrada de requisição de transporte;
- O módulo CDT não realiza nenhuma designação de transportador enquanto houver requisição de transporte;
- A designação dos transportadores no módulo CDT será no máximo igual ao número de transportadores disponíveis no sistema.
- O módulo CUV inicia sem nenhuma designação de processo de transporte e com uma marca no *box* B2;
- Os transportadores VGA(A) e VGA(B) estão inicialmente posicionados no estacionamento, nos boxes 42 e 43 respectivamente.

CAPÍTULO

6 CONCLUSÕES

Nos sistemas convencionais de controle de alocação de recurso, existe a dificuldade em alocar as MFMs porque o sistema foi desenvolvido para alocar o recurso (máquina) e não a funcionalidade do recurso, por isso não são consideradas as múltiplas funcionalidades contidas nas MFMs. Com o intuito de regenerar o sistema produtivo assegurando a devida reação à ocorrência de falhas nos recursos (máquinas), a arquitetura proposta neste trabalho mostrou como é realizada a síntese dos módulos de controle, a comunicação do módulo de controle com outros módulos, a exploração da flexibilidade funcional que possibilita que o sistema atue no comportamento dinâmico alocando o recurso de acordo com a funcionalidade solicitada, a designação do VGA e o cálculo da rota de menor caminho.

6.1 Contribuição deste Trabalho

A contribuição fundamental deste trabalho é a apresentação da arquitetura de sistema de controle modular abrangendo os módulos responsáveis pela flexibilidade operacional (alocação de recursos com flexibilidade funcional) associada à flexibilidade de rotas para a realocação de recursos, mediante o emprego do sistema de transporte com VGAs.

Desta forma, também foram implementadas as seguintes contribuições:

- Evolução no modelo de Designação de Funcionalidades, no qual, o processo que realiza a requisição de funcionalidades, seja identificado por meio dos atributos da marca para em seguida realizar a ordenação das requisições de funcionalidades para iniciar o atendimento das requisições.
- Evolução no modelo CDT, na qual a lista de transportadores disponíveis esteja sempre atualizada antes de iniciar a designação de transportadores.

- Evolução no modelo CUV, na qual a execução do procedimento para geração do E-MFG do módulo de controle seja executada no momento em que não existir nenhuma mensagem de transportador comunicando o término do trajeto.

6.2 Trabalhos Futuros

Sugestão para trabalhos futuros:

- Propor técnicas de projeto do arranjo físico virtual que permita utilizar o conceito de MFM para o controle de ciclos fechados de espera;
- Propor um modelo de negociação utilizando agentes para otimizar a produtividade e evitar *deadlock* devido a múltiplos processos e uso de MFM para regeneração do sistema mediante falhas;
- Propor uma evolução da arquitetura proposta para a abordagem de sistemas distribuídos e colaborativos.
- Propor uma arquitetura que usa a flexibilidade funcional das MFMs, associados à flexibilidade do sistema de transporte para gerar rotas alternativas de produção quando houver quebra de máquinas.

REFERÊNCIAS BIBLIOGRÁFICAS

AARDAL, M. A Model of CIM. Advanced Semiconductor Manufacturing Conference and Workshop. ASMC Proceedings IEEE/SEMI, pp.148-158, 1995.

AMERICAN NATIONAL STANDARD. ANSI/ISA-S95.00.03.2005. Enterprise-Control System Integration Part 3: Activity Models of Manufacturing Operations Management, 2005.

ARAÚJO JUNIOR, L.O. Método de Programação de Sistemas de Manufatura do Tipo Job Shop Dinâmico não Determinístico. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2006.

ASATO, O.L. Regeneração de Sistemas Produtivos Mediante a Realocação Dinâmica de Recursos com Flexibilidade Funcional. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2015.

ASATO, O.L., PESSOA, M.A.O., JUNQUEIRA, F., SANTOS FILHO, D.J., OKAMOTO JUNIOR, J., MIYAGI, P.E. Using the Enhanced - Mark Flow Graph for Dynamic Resource Allocation in Distributed Manufacturing. International Journal of Computer Integrated Manufacturing, Vol. 29, pp. 1-14, 2015.

ASATO, O.L., JUNQUEIRA, F., SANTOS FILHO, D.J., MIYAGI, P.E., ARAUJO JUNIOR, L.O. Control of Productive Systems with Functional Flexibility Level. ETFA 16th IEEE International Conference on Emerging Technologies and Factory Automation Proceedings of ETFA, pp. 1–4, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5462: Confiabilidade e Mantenabilidade. Rio de Janeiro: ABNT, 1994.

AZAB, A., NADERI, B. Modelling the Problem of Production Scheduling for Reconfigurable Manufacturing Systems. 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Capri, pp. 76-80, 2014.

BANG-JENSEN, J., GUTIN, G. Diagraphs – Theory, Algorithms and Applications. London: Springer-Verlag, 754p, 2002

BRECHER, C., BOIS-REYMOND, F-DU., NITTINGER, J., BREITBACH, T., DO-KHAC, D., FEY, M., SCHIMIDT, S. Qualifying Multi-Technology Machine Tools for Complex Machining Processes. CIRP Journal of Manufacturing Science and Technology, Vol. 13, pp. 1–14, 2016.

BOGAERTS, L., MONOSTORI, L., McFARLANE, D., KÁDÁR, B. Hierarchy in Distributed Shop Floor Control. Computers in Industry, Vol. 43, pp. 123-137, 2000.

CARDOSO, J., VALETTE, R. Redes de Petri. Florianópolis: Editora da UFSC, 212p, 1997.

CASSANDRAS, C.G. Discrete Event Systems – Modeling and Performance Analysis. Richard D. Irwin, Inc., and Aksen Associates, Inc., 790p, 1993.

CASSANDRAS, C.G, LAFORTUNE, S. Introduction to Discrete Event Systems. Second Edition. New York: Springer Vergag, 2007.

CAVALCANTE, A., PEREIRA, C.E., BARATA, J. Component-Based Approach to the Development of Self-X Automation Systems. IFAC Workshop on Intelligent Manufacturing Systems, Lisboa, 2010.

CHOI, Y.H., KIM, S.T., SEO, T.Y., KIM, K.H. Analysis of Dynamic Cross Response Between Spindles in a Dual Spindle Type Multi-Functional Turning Machine. Journal of Power and Energy Engineering, Vol. 1, pp. 20-24, 2013.

CORMEN, T.H., LEISERSON, C.E., RIVEST, R.L., STEIN, C. Algoritmo – Teoria e Prática. Tradução da 2° edição americana. Editora Campus, 2002.

DAVID, R., ALLA, H. Petri Nets for Modeling of Dynamic Systems - A Survey. Automatica: Pergamon, Vol. 30, n.2, pp. 175-202, 1994.

DA SILVA, R.M. Controle de Sistemas Reconfiguráveis de Manufatura. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2016.

DE MORAES, C.C., CASTRUCCI, P.L. Engenharia de Automação Industrial. Rio de Janeiro, Ed. LTC, ed. 2, 2007.

DESHMUKH, A.V, TALAVAGE, J.J, BARASH, M.M. Complexity in Manufacturing Systems – Part 1: Analysis of Static Complexity. IIE Transactions 30, pp. 645, 1998.

DILTS, D.M, BOYD, N.P, WHORMS, H.H. The Evolution of Control Architecture for Automated Manufacturing Systems. Journal of Manufacturing Systems, Vol. 10, pp. 79-93, 1991.

DOBREV, M., GOCHEVA, D., BATCHKOVA, I. An Ontological Approach for Planning and Scheduling in Primary Steel Production. Intelligent Systems. IS ’08. 4th International IEEE Conference, 2008. Vol. 1, pp. 614-619, 2008.

ELMARAGHY, W., ELMARAGHY, H., TOMIYAMA, T., MONOSTORI, L. Complexity in Engineering Design and Manufacturing. CIRP Annals Manufacturing Technology, pp. 793-814, 2012.

ESPEJO, M., MOYANO, J. Lean Production: Estado Actual y Desafíos Futuros de la Investigación. Investigaciones Europeas de Dirección y Economía de la Empresa. Vol. 13, n.2, pp. 179-202, 2007.

GAITHER, N., FRAIZER, G. Administração da Produção e Operações. Ed. Thomson Pioneira, ed.8, 2001.

GROOVER, M.P. Automation, Production Systems, and Computer-Integrated Manufacturing. New Jersey: Ed. Prentice-Hall, 2011. pp. 43-60.

HASEGAWA, K., TAKAHASHI, K., MASUDA, R. OHNO, H. Proposal of Mark Flow Graph for Discrete System Control. Transactions of the Society of Instrument and Control Engineers, Vol. 20, n.2, pp.122-129, 1984.

HE, L., LIU, Y.X., XIE, H.L., ZHANG, Y. Research on Job Shop Scheduling Aiming at Instant Customerisation. IEEE International Conference on Automation and Logistics, pp. 290-294, 2008.

HEES, A., REINHART, G. Approach for Production Planning in Reconfigurable Manufacturing Systems. 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Capri, pp. 70-75, 2014.

HO, Y.C. Basic Research Manufacturing Automation and Putting the Cart Before the Horse. IEEE Transactions on Automatic Control, Vol. 32, pp.1042-1043, 1987.

HO, Y.C. Special Issue on Dynamics of Discrete Event Systems. Proc of IEEE, Vol. 77, pp. 3-6, 1989.

HO, Y., CAO, X. Perturbation Analysis of Discrete Event Dynamic Systems. Kluwer Academic Publishers, 1991.

HU, H., ZHOU, M., LI, Z. Deadlock Resolution Method for Automated Manufacturing Systems Modeled with Petri Nets. IEEE international Conference on Systems Man and Cybernetics (SMC), pp. 2551-2556, 2010.

KATEEL, G., KAMATH, M., PRATT, D. An Overview of CIM Enterprise Modeling Methodologies. WSC ’96 Proceedings of the 28th Conference on Winter Simulation, 1996.

KUBO, R.H., ASATO, O.L., SANTOS, G.A, NAKAMOTO, F.Y. Sistema de Controle de Alocação de Recursos Multifuncionais para Regenerar o Sistema Produtivo. IX Congresso Nacional de Engenharia Mecânica (CONEM), Fortaleza, 2016.

KUBO, R.H., ASATO, O.L., SANTOS, G.A, NAKAMOTO, F.Y. Modeling of Allocation Control System of Multifunctional Resources for Manufacturing Systems. 12th IEEE International Conference on Industry Applications (INDUSCON), Curitiba, 2016.

LEITÃO, P., VRBA, P. Recent Developments and Future Trends of Industrial Agents. 5th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2011, pp. 15-28, 2011.

LI, X., GAO, L., ZHANG, C., SHAO, X. A Review on Integrated Process Planning and Scheduling. Int. J. Manufacturing Research 5(2), pp. 161-180, 2010.

MARTINS, R.W.C. Uma Arquitetura Modular para Controle de FMS. Dissertação (Mestrado) – Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, São Carlos, 2005.

MATSUSAKI, C.T.M. Modelagem de Sistemas de Controle Distribuídos e Colaborativos de Sistemas Produtivos. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

MATSUSAKI, C.T.M., SANTOS FILHO, D.J. Modeling of Distributed Collaborative Control Systems of Production Systems. ABCM Symposium Series in Mechatronics, pp. 345-352, 2006.

MEHRABI, M., ULSOY, A.G, KOREN, Y. Reconfigurable Manufacturing Systems and their Enabling Technologies. International Journal of Manufacturing Technology and Management. Vol. 1, n. 1, pp. 114-131, 2000.

MENDES, J.M., LEITÃO, P., COLOMBO, A.W., RESTIVO, F. Service-Oriented Control Architecture for Reconfigurable Production Systems. 6th IEEE International Conference on Industrial Informatics (INDIN), Daejeon, 2008.

MIYAGI, P.E. Controle Programável - Fundamentos do Controle de Sistemas a Eventos Discretos. São Paulo: Ed. Edgard Bluncher Ltda, 1996.

MIYAGI, M., KISIL, M., SANTOS FILHO, D.J., MARUYAMA, N., MIYAGI, P.E. Modelagem de Serviço de Saúde Utilizando Rede de Petri. 4° SBAI – Simpósio Brasileiro de Automação Inteligente, São Paulo, 2000.

MORIWAKI, T. Multi-Functional Machine Tool. CIRP Annals Manufacturing Technology, pp. 736-749, 2008.

MURATA, T. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, pp. 541-580, 1989.

NAKAMOTO, F.Y. Projeto de Sistemas Modulares de Controle para Sistemas Produtivos. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2008.

NAKAMOTO, F.Y., SANTOS FILHO, D.J., MIYAGI, P.E. Resource Allocation Control in Flexible Manufacturing Systems Using the Deadlock Avoidance Method. ABCM Series in Mechatronics, pp. 454-460, 2008.

PAPAKOSTAS, N., EFTHYMIOU, K., MOURTZIS, D., CHRYSSOLOURIS, G. Modelling the Complexity of Manufacturing Systems Using Nonlinear Dynamics Approaches. CIRP Annals – Manufacturing Technology 58, pp. 437, 2009.

PEIXOTO, A.P. Desenvolvimento de Sistemas de Automação da Manufatura Usando Arquiteturas Orientadas a Serviços e Sistemas Multi-Agentes. Dissertação (Mestrado) - Escola de Engenharia da Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

PELS, H.J., WORTMANN, J.C., ZWEGERS, A.J.R. Flexibility in Manufacturing: An Architectural Point of View. Computers in Industry, Vol. 33, pp. 199-217, 1997.

PETERSON, J.L. Petri Net Theory and the Modeling of Systems. Englewood Cliffs, N.J., Prince-Hall, 1981.

PIAO, H., HUANG, J., WANG, C. The Process Modeling and Integration for JIT Automotive Supply Logistics Based on IDEF9000. International Conference on Automation and Logistics. IEEE, P. O. T, pp. 215-219, 2007.

PREVIERO, W.D. Estratégias de Resolução para o Problema de Job-Shop Flexível. Tese (Doutorado) – Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo, 2016.

RAJADELL, M., SÁNCHEZ, J. Lean Manufacturing: La evidencia de una necesidad. Ediciones Díaz de Santos, 2010.

RAMADGE, P.J., WONHAM, W.M. Modular Supervisory Control of Discrete-Event Systems. Mathematics of Control, Signals and Systems, n. 1, pp. 13-30, 1988.

RAY, A. Networking for Computer-Integrated Manufacturing. Network IEEE, Vol. 2, n. 3, pp. 40-47, 1988.

REISIG, W. Petri Nets: An Introduction. Springer-Verlag, Berlin Heidelberg, 1985.

RIASCOS, L.A.M. Metodologia para Detecção e Tratamentos de Falhas em Sistemas de Manufatura Através de Rede de Petri. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2002.

ROSÁRIO, J.M. Princípios de Mecatrônica. Prentice Hall, São Paulo, 2005.

SALLAK, M., SIMON, C. AUBRY, J. A Fuzzy Probabilistic Approach for Determining Safety Integrity Level. IEEE Transaction on Fuzzy Systems, Vol. 16, pp. 239-248, 2008.

SANTOS FILHO, D.J. Controle de Sistemas Antropocêntricos de Produção Baseado em Redes de Petri Interpretadas. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 1998.

SANTOS FILHO, D.J. Aspectos do Projeto de Sistemas Produtivos. Tese de Livre Docência - Escola Politécnica da Universidade de São Paulo, São Paulo, 2000.

SANTOS FILHO, D.J., NAKAMOTO, F.Y., JUNQUEIRA, F., MIYAGI, P.E. Task Control of Intelligent Transportation Vehicles in Manufacturing Systems. Mechatronics Series 1: Intelligent Transportation Vehicles, Vol. 1, Dubai: Bentham Science Publishers, pp. 146-169, 2011.

SAVORY, P. MACKULAK, G, COCHRAN, J. Material Handling in a Flexible Manufacturing System Processing Part Families. Simulation Conference, pp. 375-381, 1991

SHROUF, F., ORDIERES, J., MIRAGLIOTTA, G. Smart Factories in Industry 4.0: A Review of the Concept and of Energy Management Approached in Production Based on the Internet of Things Paradigm. 2014 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 697-701, 2014.

SIEGWART, R., NOURBAKHSH, I.R., SCARAMUZZA, D. Introduction to Autonomous Mobile Robots. MIT Press, Vol. 2, pp. 382-383, 2011.

SOUZA, J.A.L. Mitigação de Falhas Críticas em Sistemas Produtivos. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2015.

SQUILLANTE JR, R. Diagnóstico e Tratamento de Falhas Críticas em Sistemas Instrumentados de Segurança. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2011.

SUH, N.P. Complexity: Theory and Applications. Oxford University Press, 2005.

TAKAHASHI, K., HASEGAWA, K., MIYAGI, P.E. Hierarchical Programming of Sequential Control Systems by Mark Flow Graph. ETFA 7th IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 2, Barcelona, pp. 1297-1305, 1999.

TANENBAUM, A.S. Sistemas Operacionais Modernos. LTC - Livros Técnicos e Científicos Editora S.A, pp. 163-179, 493p, 1995.

TRICAS, F.; COLOM, J.M.; MERELO, J.J.; Computing Minimal Siphons in Petri Net Models of Resource Allocation Systems: An Evolutionary Approach, 308 PNSE'14 – Petri Nets and Software Engineering, 2014. p.307-322.

YAMAZAKI, T. Development of Hybrid Multi-tasking Machine Tool: Integration of Additive Manufacturing Technology with CNC Machining. 18thCIRP Conference on Electro Physical and Chemical Machining (ISEM XVIII), pp. 81-86, 2016.

WANG, S., ZHANG, J., LIU, W. A Novel Flexible System for Denture Design and Manufacture. Complex Medical Engineering. CME IEEE/ICME International Conference On, pp. 1965-1969, 2007.

WEYER, S., SCHMITT, M., OHMER, M., GORECKY, D. Towards Industry 4.0 - Standardization as the Crucial Challenge for Highly Modular, Multi-Vendor Production System. IFAC International Federation of Automatic Control, pp. 579-584, 2015.

WIENDAHL, H.P., SCHOLTISSEK, P. Management and Control of Complexity in Manufacturing. CIRP Annals – Manufacturing Technology 43, pp. 533, 1994.

ZHAO, Q., CHEN, D. The Research to Power SCADA Based on J2EE Framework. Information Engineering. ICIE'09. Wase International Conference On, pp. 435-438, 2009.

ZHOU, M., ZURAWSKI, R. Petri Nets and Industrial Applications: A Tutorial. IEEE Transactions on Industrial Electronics, Vol. 41, n. 6, pp. 567-583, 1994.

ANEXO A – FERRAMENTAS DE MODELAGEM

ANEXO A.1 Rede de Petri

A Rede de Petri (RdP) é um método de estudo de sistemas dinâmicos a eventos discretos apresentado na tese de doutorado de Carl A. Petri em 1962 para o estudo da comunicação entre autômatos (CARDOSO *et al.,* 1997).

Como ferramenta matemática, um modelo de RdP pode ser descrito por um conjunto de equações algébricas lineares que representam o comportamento do sistema e possibilitam executar uma análise formal das suas propriedades (ZHOU *et al.,* 1994). Uma RdP é um grafo orientado composto por dois tipos de nós, uma chamada de transição (*transitions*) e outra chamada de nós (*places*). Para os arcos do grafo são associados números inteiros e eles podem partir das posições para as transições ou das transições para as posições. Cada posição pode conter um número de marcas (*tokens*) que respeitando certas condições, podem se mover ao longo dos arcos, respeitando os sentidos destes (PETERSON, 1981; REISIG, 1985; MURATA, 1989; CASSANDRAS *et al.,* 2007; CARDOSO et al., 1997).

Formalmente, é adotada a seguinte definição: Uma RdP é uma quíntupla (P, T, A, W, m_0) em que (NAKAMOTO, 2008):

- P = {$þ_1$, $þ_2$, $þ_3$,...$þ_n$} é um conjunto finito de posições ou lugares
- T = {t_1, t_2, t_3,...t_m} é um conjunto finito de transições
- A é um conjunto finito de arcos pertencentes ao conjunto (P X T) U (T X P), em que (P X T) representa o conjunto dos arcos orientados de $þ_i$ para t_j, também designados por ($þ_i$, t_j), e (T X P) representa o conjunto dos arcos orientados de t_i para $þ_j$, ou (t_i, $þ_j$)
- W é a função que atribui um peso *w* (um número inteiro) a cada arco
- m_0 é um vetor cuja i-ésima coordenada define o número de marcas (*tokens*) na posição $þ_i$, no início da evolução da rede.
- Os conjuntos T e P são distintos, i.e., T ∩ P = Ø
- n = |P| é a cardinalidade do conjunto P, o número de posições da RdP.
- m = |T| é o número de transições da RdP.

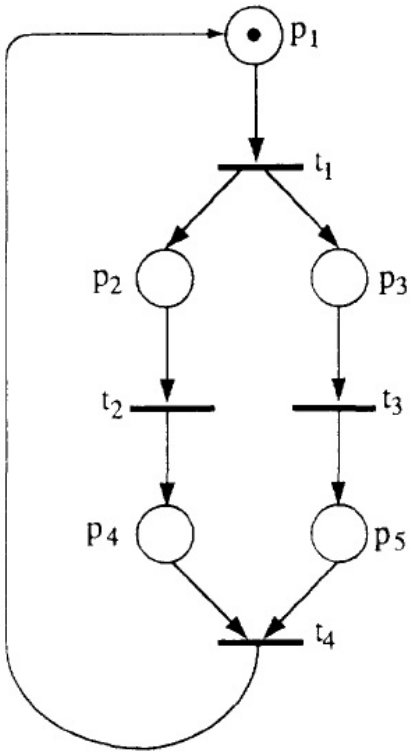


Figura 40 - Representação Gráfica de um exemplo de uma Rede de Petri (ZHOU *et al.,* 1994).

A técnica de construção de modelos por refinamento (*TOP-DOWN*), se inicia de uma visão sintética do sistema a eventos, definindo-se apenas duas possíveis condições “sistema em operação” e “sistema não em operação”, essas duas possíveis condições estão associadas com duas transições que são responsáveis pela mudança entre as condições e também aos comandos externos necessários ao sistema. Existe também a técnica de construção de modelos por agrupamento (*BOTTOM-UP*), nessa técnica de modelagem, se inicia a construção de redes para descrever os subsistemas simples e após isso, os subsistemas são agrupados de forma a se formar o modelo final. Como as RdPs possuem como característica a facilidade de interconexão, é possível formar uma única RdP composta de vários subsistemas (DE MORAES *et al.,* 2007).

ANEXO A.2 Propriedades da Rede de Petri

O comportamento dinâmico de uma RdP pode ser caracterizado pela verificação de determinadas propriedades. Abaixo são descritas as propriedades dependentes de uma marcação inicial (MURATA, 1989).

A.2.1 Alcançabilidade

A alcançabilidade é a base fundamental para o estudo das propriedades dinâmicas de qualquer sistema. O disparo de uma transição habilitada irá alterar a distribuição das marcas em uma rede assim como uma sequência de disparos resultará em uma sequência de marcações. Uma marca M_n é dita alcançável a partir de uma marca M_o se existir uma sequência de disparos que transforma a marca M_o em M_n. Uma sequência de disparos ou ocorrências são denotadas por $\sigma = M_o\ t_1\ M_1\ t_2\ M_2 ... t_n\ M_n$ ou simplesmente por $\sigma = t_1\ t_2 ... t_n$. Neste caso, M_n é alcançável a partir de M_o por σ e escrevemos $M_o\ [\alpha > M_n]$. O conjunto de todas as marcas possíveis acessíveis a partir de M_o numa rede (N,M_o) é denotado por $L(N,M_o)$ ou simplesmente $L(M_o)$.

O problema de alcançabilidade de uma RdP é saber se $M_n \in R(M_o)$ para uma determinada marcação M_n em uma rede (N,M_o). Em algumas aplicações, pode ser interessante o estudo de marcas de um subconjunto de lugares e não se preocupar com o resto dos lugares em uma rede.

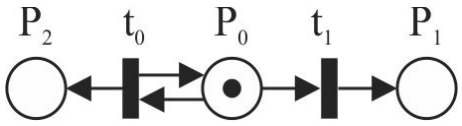


Figura 41 – Exemplo de RdP para verificação da alcançabilidade.
(Adaptado de SOUZA, 2015)

A.2.2 Limitabilidade e Segurança

Diz-se que uma RdP (N,M_o) é k-limitada ou simplesmente limitada se o número de marcas em cada lugar não exceder um número finito k para qualquer marcação acessível a partir de M_o. Uma RdP (N,M_o) é dita segura se for 1-limitada. Frequentemente os lugares de uma RdP são representados por buffers e registradores para o armazenamento de dados. Com a verificação de que a RdP é limitada e segura, fica garantido que não ocorrerá *overflow* nos *buffers* ou registradores, sem importância da sequência de eventos.

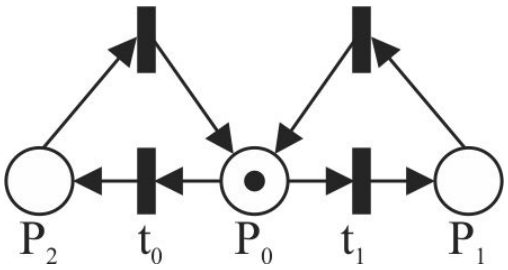


Figura 42 – Exemplo de RdP para verificação da limitabilidade.
(Adaptado de SOUZA, 2015)

A.2.3 Vivacidade

O conceito de vivacidade está relacionado com a total ausência de *deadlocks* (estado indesejável de travamento) em sistemas operacionais. Uma RdP (N,M_o) é considerada viva se, independente de qual marcação é alcançada a partir de M_o, qualquer transição da rede pode ser disparada por alguma sequência de disparo, ou seja, uma RdP viva garante a operação livre de *deadlocks*, independente da sequência de disparo escolhida. A vivacidade é uma propriedade ideal, porém praticamente impraticável de ser verificada de forma ideal. Dessa forma, é relaxada a condição de vivacidade sendo definida por diferentes níveis:

- Morta (L0-viva) se uma transição *t* não for possível de ser disparada em qualquer sequência a partir de M_o;
- L1-viva se a transição *t* pode ser disparada pelo menos uma vez em alguma sequência de disparos a partir de M_o;

- L2-viva se dado um número inteiro positivo *k*, a transição *t* pode ser disparada pelo menos *k* vezes em alguma sequência de disparos a partir de M_o;
- L3-viva se a transição *t* aparece infinitamente, em alguma sequência de disparos a partir de M_o;
- L4-viva ou viva se a transição *t* é L1-viva para qualquer marcação M e $R(M_o)$.

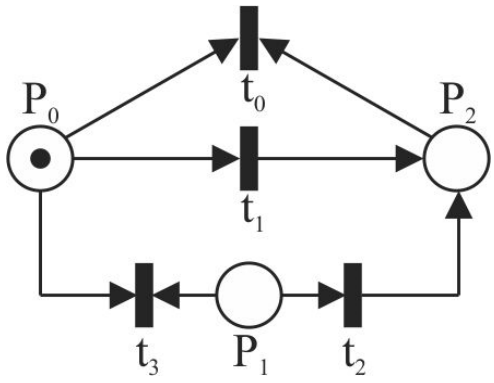


Figura 43 – Exemplo de RdP para verificação da propriedade de vivacidade.
(Adaptado de SOUZA, 2015)

A.2.4 Reversibilidade

Uma RdP é reversível se para cada marcação M em $R(M_o)$, M_o for alcançável a partir de M. Assim, em uma rede reversível pode-se sempre retornar a marca ao estado inicial. Existem muitas aplicações, na qual não é necessário retornar ao estado inicial, portanto relaxamos a condição de reversibilidade e definimos um estado de origem. A marca M' é a origem se para cada marca M em $R(M_o)$, M' é alcançável a partir de M.

A.2.5 Cobertura

Uma marcação M em uma RdP é coberta se existe uma marcação M’ em $R(M_o)$, tal que M’(p) ≥ M(p) para cada p na rede. A propriedade de cobertura é fortemente relacionada à vivacidade L1. Se M for a marca mínima para habilitar a transição *t* para o seu disparo, então a transição *t* é morta se e somente se M não estiver coberta.

A.2.6 Persistência

Uma RdP é chamada de persistente se, para quaisquer duas transições habilitadas, o disparo de uma delas não desabilita a outra. Uma transição em uma rede chamada de persistente, quando habilitada, permanece habilitada até que dispare. A persistência é relacionada a uma rede de conflitos.

ANEXO A.3 Grafo de Alocação de recursos (GAR)

O Grafo de Alocação de Recursos (GAR) (SANTOS FILHO, 2000) é um grafo bipartido e não marcado que representa a utilização de recursos pelos processos. Um GAR é constituído por nós e arcos orientados no qual um par contendo um nó e seu respectivo arco orientado de saída é denominado par de alocação e um nó e seu respectivo arco de entrada é denominado par de requisição.

No par de alocação, o nó representa um determinado recurso R_n, de acordo com o modelo E-MFG desenvolvido para controlar os processos, e o arco orientado de saída representa a etapa do processo A_n que aloca R_n para ser executada conforme o grafo E-MFG. No par de requisição, o nó representa um determinado recurso R_n e o arco orientado de entrada representa a etapa do processo A_n que requisita R_n para ser utilizado na próxima etapa do processo, de acordo com o respectivo grafo E-MFG de controle.

Um GAR pode ser representado por meio de uma Matriz de Adjacência (BANG-JENSEN *et al.,* 2006). Seja G um GAR de ordem n, ou seja, n recursos: R. A matriz de adjacência é uma matriz n x n, na qual o valor de cada elemento $e_{ij} = a_{ij} = (r_i, r_j,)$ da matriz é determinado da seguinte maneira: Se ai≠Ø e a_j≠Ø então $e_{ij} = a_{ij}$ do contrário $e_{ij} = 0$; Por exemplo, um processo q que possui a seguinte sequência de utilização de recursos: r_1, r_3 e r_2.

A representação por meio de uma Matriz de Adjacência admitindo-se uma entrada e uma saída para o processo (q_{in} e q_{out}) é:

$$
\begin{array}{c}
 \\
r_1 \\
r_2 \\
r_3 \\
q_{IN} \\
q_{OUT}
\end{array}
\begin{array}{c}
\begin{array}{ccccc} r_1 & r_2 & r_3 & q_{IN} & q_{OUT} \end{array} \\
\begin{pmatrix}
0 & 0 & q_{1,3} & 0 & 0 \\
0 & 0 & 0 & 0 & q_{2,OUT} \\
0 & q_{3,2} & 0 & 0 & 0 \\
q_{IN,1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

ANEXO A.4 Mark Flow Graph (MFG)

O MFG (HASEGAWA *et al.,* 1984; MIYAGI, 1996) é uma rede interpretada derivada de Redes de Petri (PETERSON, 1981; REISIG, 1985; MURATA, 1989) desenvolvida para a modelagem e controle de sistemas. O MFG é composto basicamente pelos seguintes elementos estruturais:

a) As transições que indicam a ocorrências de eventos;

b) Os boxes que representam as pré e pós-condições;

c) Os arcos orientados que estabelecem uma relação causal entre os eventos e as condições;

d) As portas que habilitam ou inibem a ocorrência dos eventos e;

e) As marcas que indicam a manutenção de uma condição.

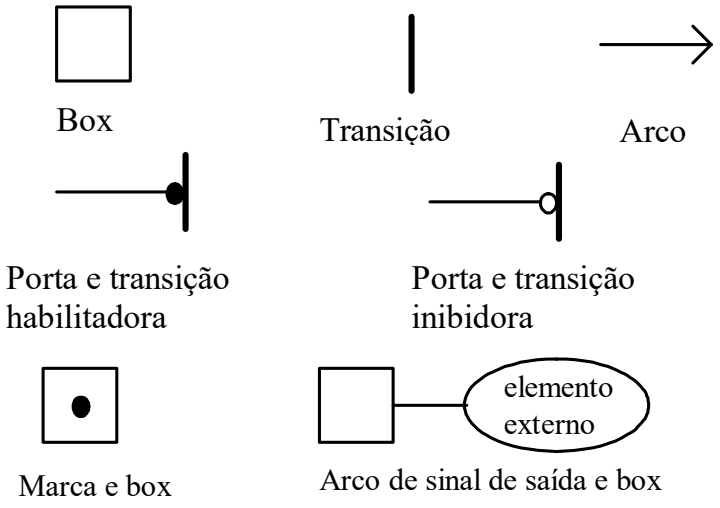


Figura 44 - Elementos básicos do MFG.
(Adaptado de HASEGAWA *et al.,* 1984)

No processo de modelagem de um SP, os boxes representam as condições, operações ou tarefas associadas aos dispositivos e as transições representam o início e término de um processo. O comportamento dinâmico do sistema é indicado pela evolução das marcas no grafo, de acordo com uma regra pré-definida de disparo das transições e que correspondem ao fluxo de itens (materiais ou informações) no sistema real.

Para representar o interfaceamento do modelo do sistema com o sistema real existem ainda dois elementos estruturais: os arcos de sinal de saída e as portas habilitadoras/inibidoras. No caso das portas, ainda há uma classificação em portas externas ou internas, dependendo da natureza do sinal de origem.

ANEXO A.5 Enhanced Mark Flow Graph (E-MFG)

A ferramenta de modelagem E-MFG inclui as marcas individuais e os elementos estruturais do MFG (*Mark Flow Graph*) (TAKAHASHI *et al.*, 1999) e permite a manipulação de marcas com atributos sem, no entanto, fugir do modelo de rede elementar convencional. Possui capacidade de modelar e controlar as alterações de informações das marcas e seleção das tarefas associadas aos *boxes* (SANTOS FILHO *et al*., 2011).

O E-MFG é composto basicamente dos elementos estruturais do MFG, conforme apresentado na figura 45:

i) Transição: representa a ocorrência dos eventos;

ii) *Box:* representa a pré e pós-condição de eventos;

iii) Marca: indica a manutenção de uma condição;

iv) *Box* Controlador: exerce a função de controlar o estado dos atributos de uma marca;

v) Arco orientado: relaciona a conexão entre o *box* e a transição, ou seja, relação entre os eventos e as condições;

vi) Porta e transição inibidora: responsável por inibir a ocorrência de eventos;

vii) Porta e transição habilitadora: responsável por habilitar a ocorrência de eventos.

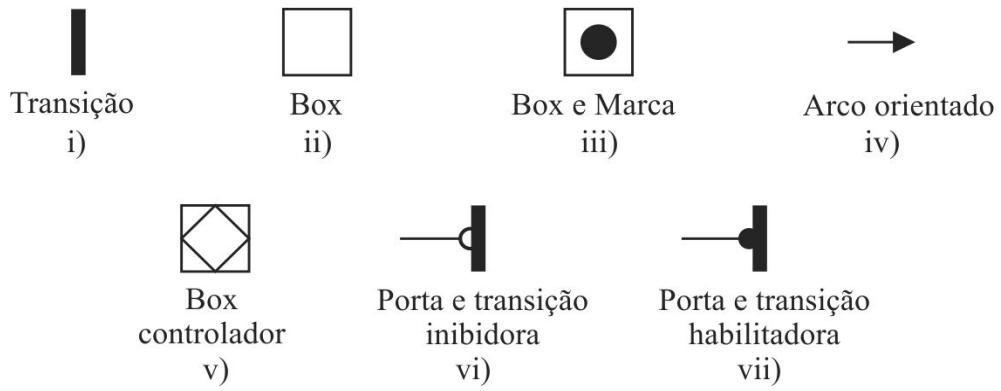


Figura 45 - Elementos básicos do E-MFG.

A.5.1 Marcas Individuais

Os elementos estruturais básicos constituem um caso particular em que não há marcas individuais e não há regras adicionais associadas às transições. No E-MFG as marcas individuais são acompanhadas de um vetor de atributos (garante a individualidade da marca) e que pode estar associada a diversas informações referentes ao produto, processo e ao controle (NAKAMOTO, 2008).

A figura 46 ilustra um exemplo de estrutura de uma marca individual.

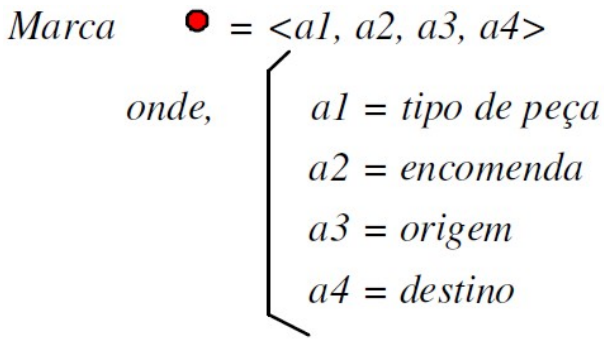


Figura 46 – Exemplo de estrutura de uma marca individual (SANTOS FILHO *et al.,* 2011).

A.5.2 Marcas Individuais Compostas

No modelamento de processos de agrupamento (montagem, carga, etc) e dispersão (desmontagem, descarga, etc) por meio de *boxes* agrupadores e dispersores em E-MFG, a marca que representa o conjunto de itens é denominada marca individual composta. Essa denominação é justificada pela sua estrutura que armazena a informação de cada item antes do agrupamento, ou a distribui para cada item após a dispersão. As informações armazenadas são representadas por um atributo adicional, por meio de um código de controle (NAKAMOTO, 2008).

A figura 47 ilustra os boxes funcionais tipo agrupador, dispersor e capacidade e um exemplo de marca individual composta.

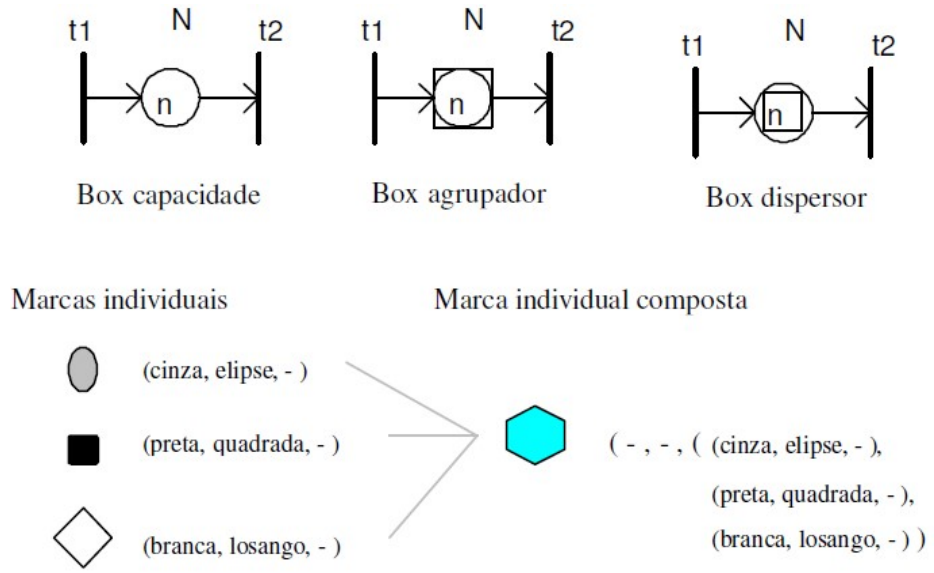


Figura 47 – Os boxes funcionais básicos e um exemplo de marca individual composta (NAKAMOTO, 2008).

A.5.3 Manipulação dos Atributos das Marcas

As manipulações dos atributos das marcas podem ser realizadas de duas formas, uma forma é utilizar o método de atualização condicional e outra forma a filtragem seletiva. O método de atualização condicional é baseado no conceito que os estados dos atributos podem ser atualizados de acordo com as funções de controle descritas por regras de produção, por outro lado, o método de filtragem seletiva é baseado na especificação dos atributos que podem ou não ser alterados na ocorrência de um evento (NAKAMOTO, 2008).

A figura 48 ilustra um exemplo de box controlador alterando os atributos de uma marca pelo método condicional.

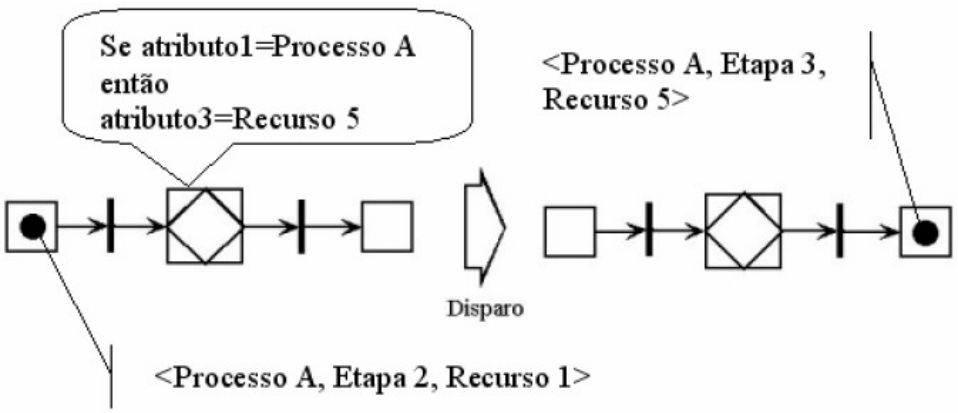


Figura 48 – Box controlador alterando o estado de uma marca individual (NAKAMOTO, 2008).

A figura 49 ilustra a atuação do procedimento filtragem seletiva.

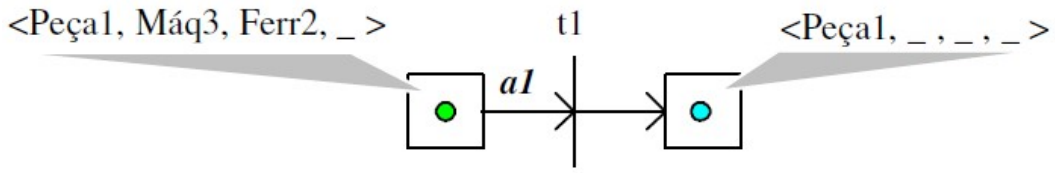


Figura 49 – Box controlador alterando o estado de uma marca individual (SANTOS FILHO, 2000).

A.5.4 Dinâmica das Regras de Disparo

A dinâmica das regras de disparo é dividida em três níveis hierárquicos estabelecidos por regras de decisão. O primeiro nível corresponde com as regras de restrições adicionais de disparo, o segundo nível corresponde com as regras de habilitação de disparo e o terceiro nível corresponde com as regras de realização de disparo que verificam as regras de arbitragem em situações que envolvam conflito e a verificação das regras de filtragem seletiva dos atributos (NAKAMOTO, 2008).

A figura 50 ilustra a alteração dos atributos das marcas decorrentes do disparo.

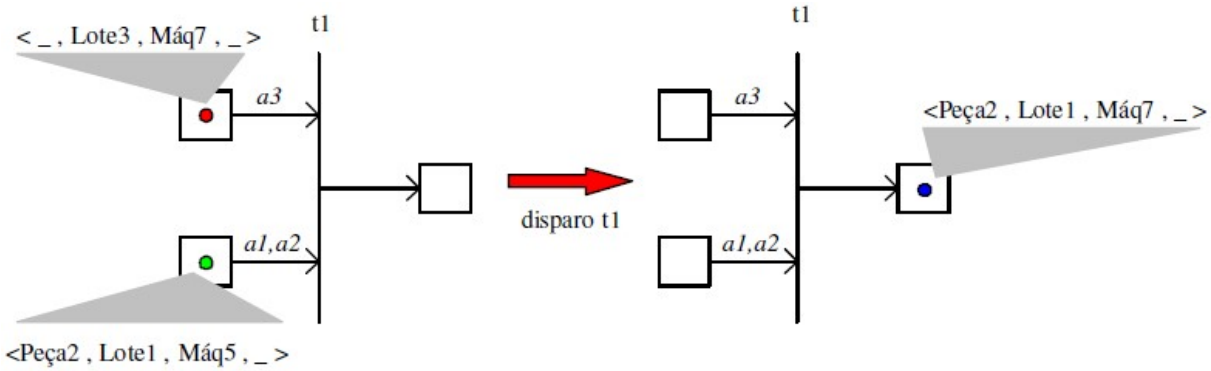


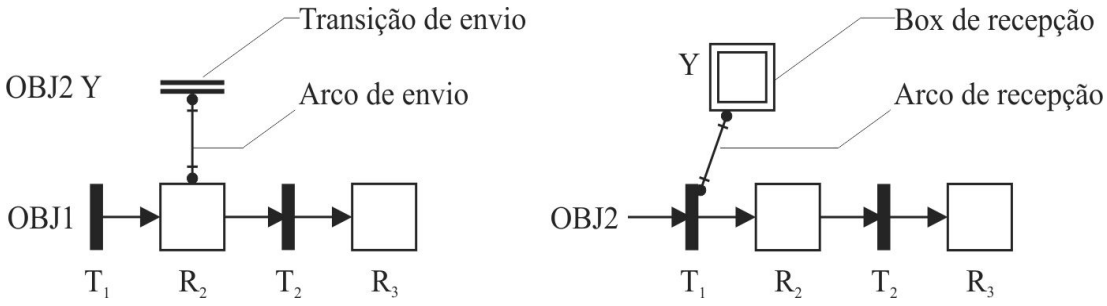
Figura 50 – Alteração dos atributos das marcas decorrentes do disparo (SANTOS FILHO, 2000).

A.5.5 Elementos Comunicadores

A incorporação de elementos comunicadores na técnica de modelagem E-MFG ocorreu devido a necessidade de uma modelagem que viabilizasse a colaboração entre os sistemas (MATSUSAKI, 2006). Com isso, a modelagem E-MFG com comunicadores, além de incluir os elementos da interface de transmissão e da interface de recepção, mantém a individualidade das marcas nos elementos estruturais do E-MFG (SANTOS FILHO *et al*., 2011).

A interface de transmissão envia eventuais mensagens assíncronas quando o *box* a ela conectada estiver marcado. A interface de recepção de mensagem recebe a mensagem e dependendo da informação recebida, realiza a ativação ou desativação da transição (MATSUSAKI, 2006).

A figura 51 ilustra as interfaces de transmissão e recepção dos elementos comunicadores.



(a) Interface de transmissão b) Interface de recepção

Figura 51 - E-MFG com as interfaces de transmissão e recepção.
(Adaptado de MATSUSAKI, 2004)

Na qual:

- Interface de envio de mensagem: substituem os arcos de sinal de saída do MFG e transmitem informações aos elementos externos, por meio da emissão de uma mensagem assíncrona eventual no díspar da transição de emissão;
    - Arco de envio (ou arco transmissor): conecta a transição de envio ao box de origem;

    - Transição de envio (ou transmissor): indica a ocorrência de envio de uma mensagem. A inscrição indica o método chamado.

- Interface de recepção de mensagem: substituem as portas habilitadoras e inibidoras na função de habilitar ou inibir a ocorrência de eventos; constituem-se de um box receptor para conter a marca referente à mensagem recebida e de arcos receptores que conectam o box receptor às transições a serem habilitadas ou inibidas.

    - Box de recepção (ou receptor): indica a recepção de uma mensagem; os atributos da marca contêm os dados recebidos;

    - Arco de recepção (ou arco receptor): conecta o box de recepção a transição que inicia o processamento desta mensagem.

Neste trabalho o comportamento dinâmico de um SP é classificado como um sistema a eventos discretos (SED), que são sistemas intensamente utilizados para descrever, analisar e controlar processos de um SP. Os processos executados no SP são constituídos por um conjunto finito de atividades sequenciais e pré-definidas que caracterizam o comportamento dinâmico que rege a evolução dos estados do sistema por meio da ocorrência de eventos (NAKAMOTO *et al*., 2008). Para a modelagem, análise e especificação do controle, será utilizada nesse trabalho a ferramenta de modelagem e controle E-MFG (*Enhanced - Mark Flow Graph*) que é uma extensão da Rede de Petri.

ANEXO A.6 Production Flow Schema (PFS)

O PFS é uma técnica desenvolvida para sistematizar e facilitar a modelagem por redes sendo utilizado para descrever, graficamente e conceitualmente, os processos relacionados com a produção de itens (peças, produtos, informações, etc) sob forma de sequências de etapas de atividades e de distribuição (MIYAGI, 1996). Essa técnica consiste de nós de elementos-atividade, nós de elementos distribuidores e arcos de fluxo. Os elementos-atividades podem ser expandidos em duas transições e um lugar (lugar-atividade) (NAKAMOTO, 2008).

A figura 52 ilustra os elementos do PFS.

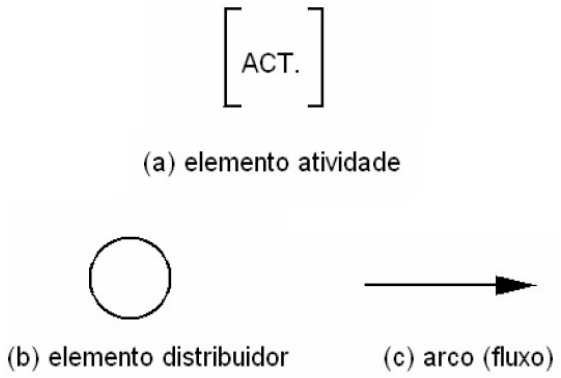


Figura 52 – Elementos PFS.
(Adaptado de MIYAGI, 1996)

O elemento-atividade pode ser expandido em duas transições e um lugar (lugar-atividade). Quando é necessária a indicação do início e da conclusão de uma atividade, é distinguido a transição de entrada (como a transição de início) da transição de saída (como transição final).

O elemento-distribuidor pode ser expandido em lugar (lugar-distribuidor) com transições a entrada e a saída (NAKAMOTO, 2008). Caso sejam expandidos dois elementos conectados um ao outro em uma rede lugar-transição, a transição de saída da primeira etapa e a entrada da etapa seguinte forma uma única transição (NAKAMOTO, 2008).

ANEXO A.7 Mapeamento GAR/MFG

Uma vez que o GAR é um grafo transição que representa um autômato finito, é possível mapeá-lo em uma rede (SANTOS FILHO, 2000; REISIG, 1985; CASSANDRAS, 1993).

Considerando um autômato de estados finitos dado pela quíntupla (**E**, **X**, **f**, $\mathbf{x_0}$, **F**) na qual (CASSANDRAS, 1993):

- **E**: É um alfabeto finito;
- **X**: É um conjunto de estado finitos;
- **F**: É uma função de transição de estados **f**: **X** x **E** -> **X**;
- $\mathbf{X_0}$: É o estado inicial tal que $\mathbf{x_0}$ Є **X**;
- **F**: É um conjunto de estados tal que **F** $\underline{\mathbf{C}}$ **E**.

Considerando o grafo MFG bipartido, representado por uma sêxtupla (**B**, **T**, **A**, $\mathbf{G_I}$, $\mathbf{G_E}$, **S**) na qual (MIYAGI, 1996):

- **B** = {B1,..., Bi} com (i≥1) é um conjunto de boxes;
- **T** = {T1,...,Tj} com (j≥1) é um conjunto de transições;
- **A** é um conjunto de arcos orientados;
- **Gi** = $\{G_I,...,G_{Im}\}$ com (m≥0) é um conjunto de portas internas;
- $\mathbf{G_E}$ = $\{G_E,...,G_{En}\}$ com (n≥0) é um conjunto de portas externas;
- **S** = $\{S_1,...,S_P\}$ com (p≥0) é um conjunto de arcos de sinais de saída.

Dado um autômato de estado finitos (CASSANDRAS, 1993) é possível construir uma rede de Petri elementar (REISIG, 1985), uma vez que se pode associar cada estado global representado pelo autômato a uma marcação na rede de Petri correspondente. Para a construção de um modelo MFG a partir de um GAR, segue-se o seguinte procedimento (CASSANDRAS, 1993):

- Cada estado em **X** é definido como um box em **B**;

- A cada par de estados (**x**, **‘x**), na qual **x’** = **f**(**x**,**e**) para **e**Є**E**, define-se uma transição em **T** tal que **T**={(**x**,**x’**):**x**Є**X**, **x’** = **f**(**x**,**e**) para qualquer **e**Є**E**};
- A seguir, define-se um arco a partir de cada par (**x**, **t**) e (**t**, **x’**) com **x**Є**X** e **x’**Є**X** e **t**Є**T** tal que **A**={(**x**, **t**): **x**Є**X, t**Є**T**}U{(**t**, **x’**):**x’**Є**X**, **t**Є**T**}.

A partir deste procedimento obtemos o modelo MFG correspondente ao GAR.

ANEXO B – FERRAMENTA COMPUTACIONAL PARA DETERMINAR O CAMINHO MÍNIMO

ANEXO B.1 Algoritmo de *Dijkstra* com *Heap* Binomial

O Algoritmo de *Dijkstra* foi desenvolvido em 1959 pelo matemático holandês Edsger Wybe *Dijkstra*, este algoritmo é indicado para a resolução de problemas de caminhos mais curtos que possuam uma única origem em um grafo orientado ponderado G=(V,E) e que não possua pesos negativos de arestas (CORMEN *et al.,* 2002).

```
Algoritmo Dijkstra
Início
        inicialize a distância para todos os nós em G = infinito
        inicialize o predecessor de todos os nós em G = vazio
        enquanto H não estiver vazio faça
                u = o nó com menor rótulo extraído de H
                para cada v adjacente a u:
                        se rótulo[v] > rótulo[u] + distância[u, v]:
                                rótulo[v] = rótulo[u] + distância[u, v];
                        predecessor[v] = u;
                atualiza a posição de v em H
                        Fimse
                Fim para
        Fim enquanto
Fim Dijkstra
```

Figura 53 – Pseudocódigo do algoritmo *Dijkstra* (CORMEN *et al.,* 2002).

De acordo com Nakamoto (2008), o algoritmo de *Dijkstra* mantém um conjunto de nós cujos pesos desde a origem já foram calculados, dessa forma, o algoritmo seleciona repetidamente o nó que possui a menor estimativa (peso) e calcula os pesos dos nós adjacentes. Com isso, o algoritmo produz resultados otimizados adicionando vértices à árvore de caminho mínimo pelo processo de relaxamento de uma aresta, ou seja, consiste em verificar se existe a possibilidade de melhorar o caminho obtido até o momento (CORMEN *et al.,* 2002; BANG-JENSEN *et al.,* 2006).

O algoritmo de *Dijkstra* pode assumir qualquer valor positivo de peso de aresta e como a busca garante a solução ótima ao encontrar todos os caminhos de

custo mínimo entre qualquer ponto de origem e de destino, essa característica introduz uma complexidade adicional no algoritmo sendo necessário utilizar o conceito de estrutura de dados baseado em pilhas (SIEGWART *et al.,* 2011).

A seguir é apresentado um exemplo de grafo para cálculo do caminho mínimo e o processo de cálculo dos pesos utilizando o algoritmo de *Dijkstra* (NAKAMOTO, 2008).

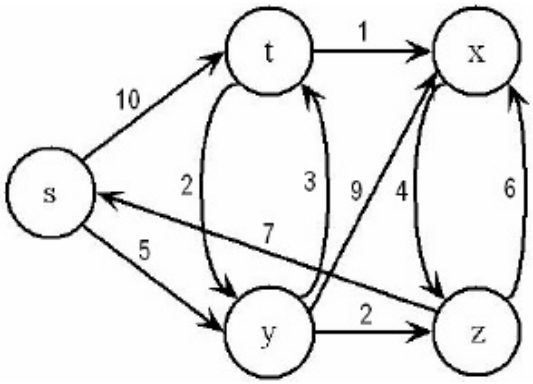


Figura 54 – Exemplo de grafo para cálculo do caminho mínimo (NAKAMOTO, 2008).

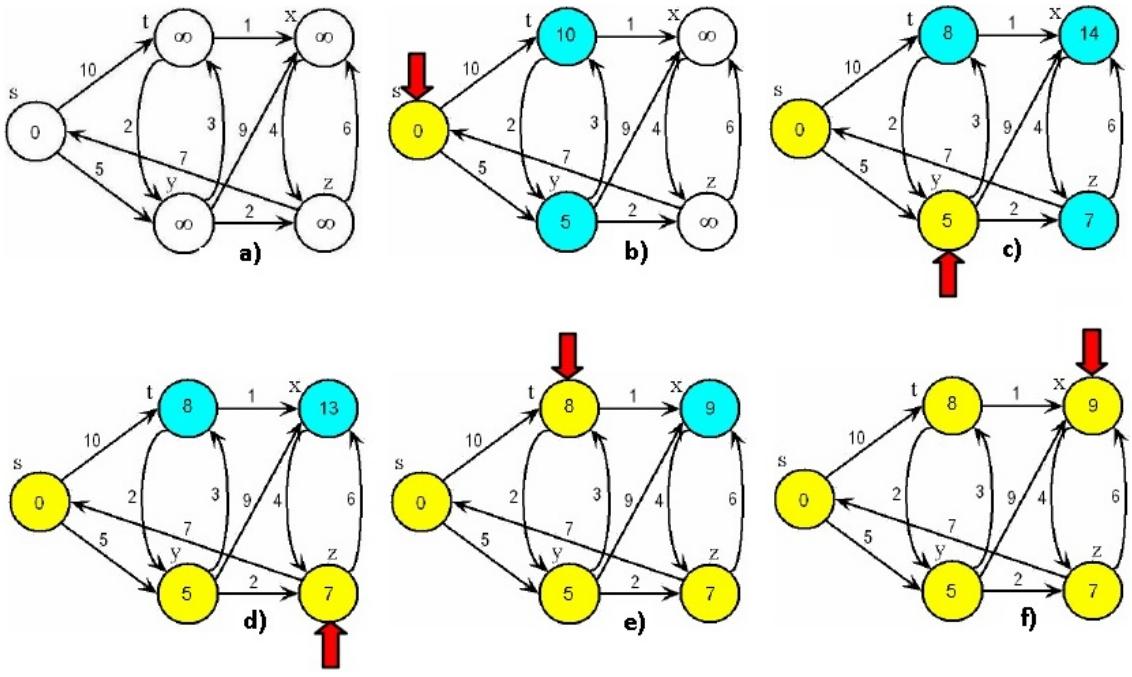


Figura 55 – Processo de cálculo do algoritmo de *Dijkstra* (NAKAMOTO, 2008).

No processo de cálculo dos pesos, o algoritmo de *Dijkstra* realiza uma estimativa inicial do menor caminho ou menor custo e atualiza o resultado da estimativa de forma sucessiva a partir do nó inicial. O valor do nó precedente é alterado e armazenado para cada nova estimativa, dessa forma, após todos os nós serem calculados, são obtidos todos os valores de menor caminho ou menor custo partindo do nó inicial do grafo até os demais nós do grafo.

A estrutura de dados *Heap Binomial*, também conhecida como fila de prioridades (TANENBAUM *et al.,* 1995), será empregado no algoritmo de *Dijkstra* para determinar o nó com o menor custo. O principal objetivo é a redução do tempo de execução do algoritmo de *Dijkstra* na determinação do nó com menor custo. Um *Heap Binomial* é uma árvore ordenada que possui as seguintes propriedades:

1. Existem 2k nós;
2. A altura da árvore é k;
3. Existem exatamente (k_i) nós na profundidade i, para i=0,1,...,k;
4. A raiz tem grau k, o qual é maior que o de qualquer outro nó;
5. Se os filhos da raiz são numerados da esquerda para a direita por k-1, k-2,...,0, o filho i é a raiz de uma subárvore Bi.

Um *Heap* Binomial H é definido como um conjunto de árvores binomiais que satisfaz as seguintes propriedades:

1. Cada árvore binomial H obedece à propriedade de *heap* mínimo, ou seja, a chave de um nó é maior ou igual à chave de seu pai;
2. Existe no máximo uma árvore binomial em H cuja raiz tem grau k, para qualquer inteiro não negativo k.

Na figura 56 é apresentado um *heap binomial* com 13 nós. O *heap* consiste nas árvores binomiais B0, B2 e B3 com 1, 4 e 8 nós respectivamente, totalizando 13 nós (n=13). Observa-se que as árvores estão ordenadas com um *heap* binomial mínimo, ou seja, a chave de qualquer nó não é menor que a chave de seu pai.

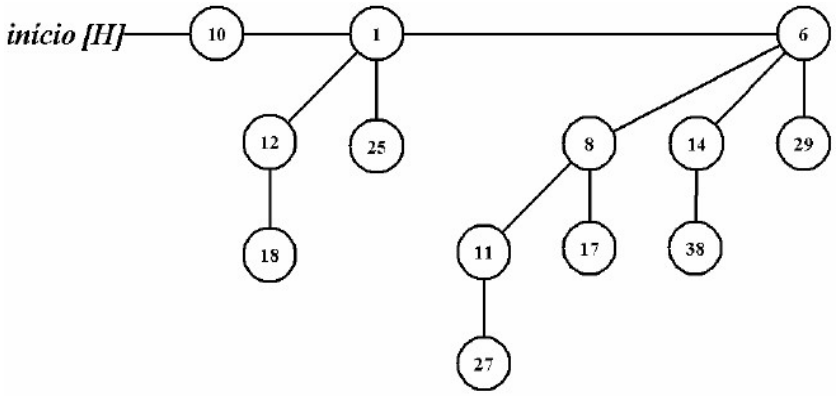


Figura 56 – Representação da árvore binomial com 13 nós (NAKAMOTO, 2008).

As raízes das árvores binomiais dentro de um *Heap* Binomial estão organizadas em uma lista de raízes (nós 10, 1 e 6) que estão em ordem decrescente de grau (quantidade de filhos).

A figura 57 ilustra a representação de um nó Heap Binomial na qual a estrutura armazena 3 ponteiros, a saber: ponteiro para o pai (ancestral), para o filho e para o irmão. A única situação em que o ponteiro para o pai é NULO será quando o nó for uma raiz.

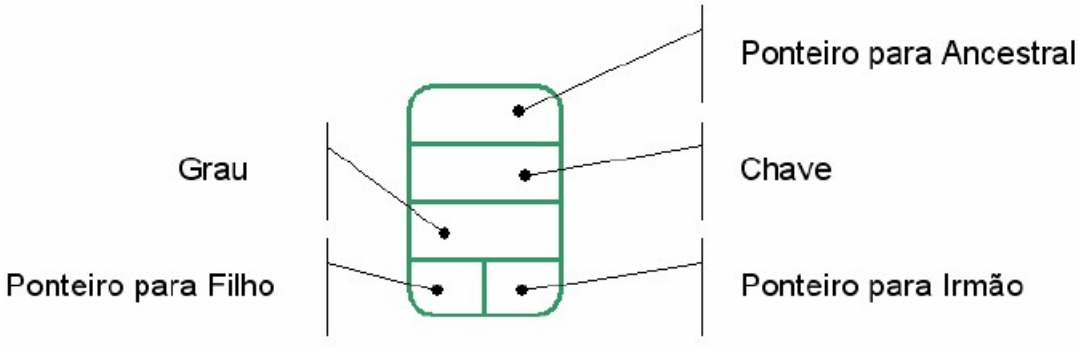


Figura 57 – Representação do nó de um *Heap* Binomial (NAKAMOTO, 2008).

A figura 58 ilustra a representação mais detalhada do *Heap* Binomial H na qual cada árvore binomial é armazenada na representação de filho da esquerda, irmão da direita e cada nó armazena o grau correspondente.

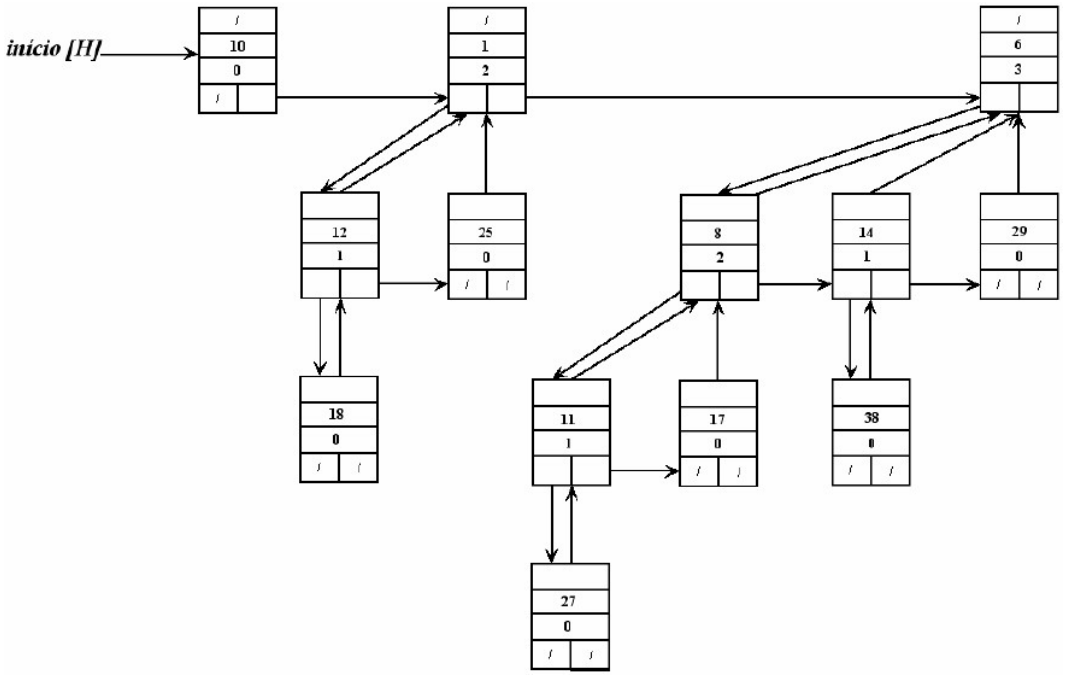


Figura 58 – Representação do nó de um *Heap* Binomial (NAKAMOTO, 2008).

As principais operações de um *Heap* Binomial são as seguintes (CORMEN *et al.,* 2002):

- BINOMIAL_HEAP_INSERT(H, x): Insere o nó x, cujo campo chave já foi preenchido anteriormente;
- BINOMIAL_HEAP_MINIMUM(H): Retorna um ponteiro para o nó do *heap* H cuja chave é mínima;
- BINOMIAL_HEAP_EXTRACT_MIN(H): Elimina o nó do *heap* H cuja chave é mínima, retornando um ponteiro para o nó;
- BINOMIAL_LINK(y, z): Torna o nó y o novo início da lista ligada de filhos do nó z;
- BINOMIAL_HEAP_UNION(H1, H2): Cria e retorna um novo *heap* que contém todos os nós dos *heaps* H1 e H2. Os *heaps* H1 e h2 são “destruídos” por esta operação;
- BINOMIAL_HEAP_DECREASE_KEY (H, x, k): Atribui ao nó x dentro do *heap* H o novo valor de chave k, que se supõe não ser maior que o seu valor de chave atual;

    - BINOMIAL_HEAP_DELETE(H, x): Elimina o nó x do *heap* H;
    - BINOMIAL_HEAP_MERGE(H1, H2): Intercala as listas de raízes de H1 e H2 em uma única lista ligada que é ordenada por grau em ordem crescente;
- MAKE_BINOMIAL_HEAP (): Cria e retorna um novo *heap* que não contém nenhum elemento.