

WANDERLEY LIU

**APLICAÇÃO DA METODOLOGIA
MFG/PFS NO DESENVOLVIMENTO DE
SISTEMAS DE INFORMAÇÕES DE
INDÚSTRIAS DE MANUFATURA**

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de Mestre em
Engenharia

Área de Concentração:
Engenharia Mecânica

Orientador:
Prof. Dr. Paulo Eigi Miyagi

São Paulo

1993

DEDALUS - Acervo - EPMN



31600009945

Aos meus pais e aos meus irmãos, agradeço todo o apoio e paciência que me acompanharam durante o trabalho.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Paulo Eigi Miyagi, pelas pacientes instruções, pelas diretrizes seguras, pela constante supervisão e, principalmente, pela oportunidade concedida para o desenvolvimento de uma pesquisa aplicada.

A Rubens Giacon e Carlos Alberto Garcia, coordenadores do projeto UNI da Mercedes-Benz do Brasil, pelo fornecimento do tema da pesquisa, pela bolsa de estudos e por todo o suporte oferecido para a viabilização do trabalho.

A Geraldo Acceturi Araújo, Gilberto Jun Kuzuhara e demais integrantes do Departamento de Processos da Mercedes-Benz do Brasil, pela oportunidade de participar do projeto do Sistema de Gerenciamento da Produção, uma das motivações principais da dissertação.

A todos os membros do grupo de pesquisa em redes de Petri da Mecatrônica, em especial ao Prof. Diolino José dos Santos Filho, cujas discussões e críticas foram fundamentais na consolidação dos conceitos do trabalho.

A Srta. Márcia Yagui, pelo cuidado e zelo nas revisões finais do texto, pelo auxílio na elaboração transparências e pelas críticas adicionais, que foram contribuição importante para o enriquecimento do trabalho.

A todos que, direta ou indiretamente, contribuíram para a realização dessa dissertação..

SUMÁRIO

Lista de figuras

Lista de abreviaturas

Resumo

"Abstract"

1 . INTRODUÇÃO	1
2 . CONCEITOS FUNDAMENTAIS	5
2.1 Sistemas de Manufatura	5
2.1.1 Definição	5
2.1.2 Sistemas Integrados de Manufatura (SIM)	6
2.1.3 Atividades de um sistema de manufatura	7
2.1.4 Condições e relações entre atividades	8
2.2 Sistemas de informações	10
2.1.2 Considerações a respeito da informação	10
2.2.2 SIs aplicados a sistemas de manufatura	12
2.2.3 Exemplos de SIs	14
2.2.4 Requisitos para o desenvolvimento de SIs	15
2.3 Sistemas de eventos discretos	16
2.3.1 Utilização de modelos formais	16
2.3.2 Sistemas de eventos discretos e modelos	17
2.4 Metodologias	18
2.4.1 Introdução	18
2.4.2 O estado da arte	20
2.4.3 O MFG e a metodologia PFS/MFG	21
2.4.4 A metodologia proposta	22
3 . CONSTRUÇÃO DO MODELO CONCEITUAL	27
3.1 A definição das necessidades	27
3.1.1 Conceito	27
3.1.2 Atividades para a definição das necessidades	28
3.1.3 Características da etapa	30
3.1.4 Levantamento das necessidades	31

3.1.5	Análise das necessidades	32
3.2	Construção de modelos conceituais	33
3.2.1	A linguagem PFS - Production Flow Schema	33
3.2.2	Representação de características de SEDs em PFS	35
3.2.3	Relações através de fluxos secundários	37
3.2.4	Refinamento das atividades	38
3.2.5	Abordagens para a construção de modelos	39
3.3	Aplicações da etapa em desenvolvimento de SIs	41
3.3.1	Aplicações de SIs em controle e supervisão	41
3.3.2	Elementos de um sistema de informações	42
3.4	Exemplos de aplicação da metodologia	43
3.4.1	Identificação dos objetivos do sistema	43
3.4.2	Levantamento das necessidades	44
3.4.3	Organização dos conhecimentos	45
3.4.4	Modelo da interface com o usuário	48
3.4.6	Modelo da interface com as máquinas	50
4	CONSTRUÇÃO DE MODELOS FUNCIONAIS	52
4.1	Introdução	52
4.1.1	As redes de Petri	52
4.1.2	O MFG - Mark Flow Graph	53
4.2	O modelo interpretado híbrido MFG/PFS	54
4.2.1	Representação das atividades e dos elementos distribuidores	54
4.2.2	Representação dos recursos	56
4.2.3	A metodologia e as propriedades do modelo	57
4.3	A especificação do controle do fluxo de dados	60
4.3.1	Deteção e controle de "dead"-lock	60
4.3.2	O problema do conflito	61
4.3.3	Outros aspectos de controle	62
4.4	Extensões do modelo MFG	64
4.4.1	Modelo estendido com marcas individualizadas	64
4.4.2	Modelo estendido para simulação	66
4.5	Simplificações do modelo	67
4.6	Exemplos	68
4.6.1	Módulo de interface com o usuário	68
4.6.2	Controle do sistema de comunicações	70

4.6.3 Controle da interface com as máquinas	72
5 . APLICAÇÕES DA METODOLOGIA EM ASPECTOS DE IMPLEMENTAÇÃO	74
5.1 A metodologia e o desenvolvimento do software aplicativo	74
5.1.1 Nível global	75
5.1.2 Nível local	76
5.2 A metodologia e o projeto de redes de comunicação	80
5.2.1 O problema das redes de comunicações	81
5.2.2 O modelo de referência OSI	82
5.2.3 A metodologia como auxílio no desenvolvimento e especificação de redes de comunicação	84
5.2.4 Modelos para a camada de enlace	87
5.3 A metodologia e o projeto de dispositivos	92
6 . TÓPICOS RELATIVOS À ANÁLISE E SIMULAÇÃO	93
6.1 Análise a nível qualitativo	93
6.1.1 Análise de "dead-locks"	94
6.1.2 Validação dos modelos	95
6.2 Análise a nível quantitativo	96
6.2.1 Aspectos temporais na simulação	97
6.2.2 Pontos de observação no grafo	97
6.3 Ferramentas para a simulação em MFG.....	98
6.3.1 Notação MFG	99
6.3.2 Algoritmo para a construção de simuladores	101
6.4 Exemplos.....	103
6.4.1 Validação do MMPM	103
6.4.2 Dimensionamento do tempo de resposta do sistema de supervisão da produção	106
7 . CONCLUSÕES	112
REFERÊNCIAS BIBLIOGRÁFICAS	115
Apêndices	

LISTA DE FIGURAS

Fig.2-1 - Factory Automation System	9
Fig.2-2 - Fluxos de informações	13
Fig.2-3 - Ciclo de Vida e Produção de Software	19
Fig.2-4 - Etapas da metodologia proposta	25
Fig.3-1 - Fluxograma de atividades para a definição das necessidades	29
Fig.3-2 - Elementos do PFS	34
Fig.3-3(a) - Representação de seqüência	35
Fig.3-3(b) - Concorrência de atividades	36
Fig.3-3(c) - Paralelismo de atividades	36
Fig.3-3(d) - Sincronização de atividades	36
Fig.3-4 - Interpretações típicas dos fluxos secundários	37
Fig.3-5 - Refinamento das atividades	38
Fig.3-6(a) - Exemplo de representação por atigramas	39
Fig.3-6(b) - Exemplo de representação por datagramas	40
Fig.3-7 - Aplicação de SI em sistemas de controle e supervisão	41
Fig.3-8 - Partes de um sistema de informações	42
Fig.3-9 - Representação de uma máquina	45
Fig.3-10 - Módulo de interface com o usuário	47
Fig.3-11 - Modelo do sistema de comunicações	49
Fig.3-12 - Modelos da UAD	50
Fig.4-1 - Elementos do MFG	53
Fig.4-2 - Representação em MFG dos elementos PFS	56
Fig.4-3 - Representação dos recursos	57
Fig.4-4 - Caminhos de atividade	58
Fig.4-5 - Estruturas proibidas	59
Fig.4-6 - Estruturas com "dead-lock"	59
Fig.4-7 - Controle par evitar "dead-lock"	61
Fig.4-8 - Representação geral da lógica de arbitração	62
Fig.4-9 - Controle de início e fim de atividade	63
Fig.4-10 - Limitação do número de marcas na atividade	63
Fig.4-11 - Exemplo de utilização do E-MFG	65
Fig.4-12 - Elementos temporizados do MFG	66
Fig.4-13 - Simplificações do modelo	67

Fig.4-14 - Aspectos de controle do fluxo de dados da atividade de interface com o usuário	69
Fig.4-15 - Controle do fluxo de dados da atividade sistema de comunicações	71
Fig.4-16 - Controle do fluxo de dados da atividade interface com as máquinas	73
Fig.5-1 - Modelo PFS de um SI nível global	76
Fig.5-2 - Modelos de um processo em MFG	77
Fig.5-3 - Exemplo de um módulo de aplicação utilizando objetos pré-definidos	79
Fig.5-4 - Modelo OSI-ISO de sete camadas	82
Fig.5-5(a) - Modelo PFS da Máquina de Protocolos do MMS	85
Fig.5-5(b) - Modelo MFG da Máquina de Protocolos do MMS	86
Fig.5-6(a) - Modelo PFS do protocolo bit-alternante	88
Fig.5-6(b) - Refinamento do PFS em MFG	89
Fig.5-6(c) - Especificação do controle do fluxo	90
Fig.5-7 - Controle de acesso ao meio	90
Fig.6-1 - Algoritmo para construção de simuladores MFG	101
Fig.6-2 - Modelo MFG para simulação do MMPM	104
Fig.6-3 - Seqüência de troca de primitivas gerada pelo MMPM	105
Fig.6-4 - Modelo para simulação do sistema de supervisão da produção	106
Fig.6-5 - Modelo para simulação de uma estação	107

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledge
CIM	Computer Integrated Manufacturing
CSP	Communication Sequential Process
DFD	Data Flow Diagram
E-MFG	Mark Flow Graph Estendido
F-MFG	Mark Flow Graph Funcional
GBD	Gerenciador de Base de Dados
GBDL	Gerenciador de Base de Dados Local
ISO	International Standards Organization
LAN	Local Area Networks
LLC	Logical Link Control
MAC	Media Acces Control
MFG	Mark Flow Graph
MMPM	Manufacturing Message Protocol Machine
MMS	Manufacturing Message Specifications
OSI	Open Systems Interconnection
PFS	Production Flow Schema
REQ	Reques
RPC	Remote Procedure Call
SED	Sistema de Eventos Discretos
SI	Sistema de Informações
TFA	Total Factory Automation
UAD	Unidade de Aquisição de Dados

RESUMO

A integração de sistema de manufatura tem sido assunto de diversas pesquisas e investigações. Uma forma de se obter tal integração consiste no desenvolvimento de um sistema de informações apropriado para gerenciar a troca de informações e dados entre subsistemas de uma fábrica. A complexidade da fábrica e de seus subsistemas sugere a utilização de uma metodologia que possa além de modelar essa troca de informações, definir a política de controle dessas trocas.

Mark Flow Graph, uma técnica derivada de redes de Petri, e Production Flow Schema tem definido uma metodologia chamada MFG/PFS, que demonstrou-se ser uma ferramenta eficiente para modelar e especificar sistemas produtivos.

A proposta desse trabalho é investigar a modelagem e especificação de sistemas de informações utilizando a metodologia MFG/PFS. Modelos adequados para sistemas de informações são construídos, baseados nas características de sistemas produtivos. A análise dos modelos também é considerada, incluindo tópicos relacionado à validação e análise quantitativa. O trabalho fornece também três exemplos de sistemas de informações aplicados em três níveis distintos de abstração da fábrica.

ABSTRACT

Integration of manufacturing systems has been a subject of several researches and investigations. One way to achieve such integration is to develop a proper information system to manage the information and data exchange between subsystems in a factory. The complexity of the factory and its subsystems suggests the employment of a methodology that can properly model this exchange of information, as well as define the policy of controlling these exchanges.

Mark Flow Graph, a technique derived from Petri nets, and Production Flow Schema have defined a methodology called MFG/PFS, which has demonstrated to be a efficient tool to model and "specify" the control of production systems.

The purpose of this research is to investigate the modelling and specification of information systems by using a methodology based in MFG/PFS. Appropriate models for information systems are constructed, based on operationat features of production systems. The analisys of the models is also considered, including topics related to validation and quantitative analisys. The research provides also three examples of information systems applied in three levels of abstraction in the factory.

CAPÍTULO 1

INTRODUÇÃO

Em uma indústria de manufatura, podem ser identificadas diversas atividades necessárias para que suas funções, enquanto sistema produtivo, possam ser desempenhadas. São atividades como concepção e projeto de produtos e peças, planejamento do processos, planejamento e escalonamento da produção, além da própria fabricação dos produtos. Em indústrias modernas, devido à evolução tecnológica, principalmente a nível de desenvolvimento da informática, tais atividades são frequentemente auxiliadas por recursos computacionais e são geralmente identificadas como sistemas "CA" (Computer Aided). Alguns exemplos desses sistemas "CA" são:

- sistema CAE - Computer Aided Engineering (concepção de novos produtos);
- sistema CAD - Computer Aided Design (projeto);
- sistema CAM - Computer Aided Manufacturing (fabricação);
- sistema CAPP - Computer Aided Process Planning (planejamento de processos);
- sistema CAP - Computer Aided Production (programação e controle da produção).

As tendências da indústria moderna indicam que há uma necessidade de se integrar tais sistemas de forma a potencializar os benefícios de cada um deles, contribuindo para melhorar o desempenho do sistema global. O aspecto de integração de tais sistemas define o estudo dos Sistemas Integrados de Manufatura, ou CIM (Computer Integrated Manufacturing), como é mais conhecido.

O estudo de integração de sistemas de manufatura pretende tornar a indústria mais competitiva, provendo-a de certas qualidades produtivas para melhor satisfazer as necessidades do mercado. Tais qualidades são, por exemplo, aumentar a qualidade dos

produtos sem perda de produtividade e capacitar o sistema de permitir modificações em sua linha de produto em curtos espaços de tempo para satisfazer demandas específicas de consumo.

Observa-se que existem várias atividades em um sistema de manufatura podem ser realizadas independentemente uma das outras. Entretanto, seu funcionamento pode ser otimizado através da coordenação e interação com outras atividades. Essas interações, na prática, podem ser vistas como troca de informações, o que define um fluxo de informações no sistema.

O fluxo de informações mais as atividades que realizam ações sobre esse fluxo definem um sistema de informações (SI), que será responsável por gerar e gerenciar as informações do sistema de manufatura. Assim, o desenvolvimento de um SI adequado deve contribuir para a obtenção de uma maior integração do sistema de manufatura.

Motivação e justificativa do trabalho

Dada a complexidade dos sistemas envolvidos numa fábrica, o desenvolvimento de um SI adequado não é uma tarefa trivial. Essa complexidade justifica a utilização de sistemáticas bem definidas para auxiliar e direcionar o desenvolvimento dos sistemas, que são as chamadas metodologias. É possível encontrar uma grande quantidade de metodologias para desenvolvimento de SIs, geralmente baseadas em análise estruturada de sistemas. Cada uma dessas metodologias possui suas vantagens, entretanto a grande maioria carece do problema de não terem sido desenvolvidas voltadas especificamente para aplicações em ambiente de manufatura, onde aspectos como paralelismo e concorrência entre processos produtivos são importantes.

As redes de Petri [Pet77-, Agw79-, Rei85-, Mur89-, Val90-] e técnicas derivadas, como é o caso do MFG (Mark Flow Graph) [Has80-, Sek88-, Myg89b, San93-], têm sido introduzidas com sucesso como ferramentas eficientes para aplicações em sistemas de manufatura. Entretanto, dependendo da complexidade dos sistemas a serem considerados,

os modelos construídos podem tornar-se muito difíceis de serem compreendidos e tratados. Para resolver esse problema, é proposta uma metodologia onde pode-se sintetizar modelos em MFG a partir de um modelo conceitual construído com base em uma linguagem denominada PFS (Production Flow Schema) [Myg88-].

Objetivos do trabalho

A metodologia baseada em MFG/PFS é utilizada principalmente para modelagem e especificação do controle de sistemas produtivos. Nesses trabalhos [Myg88-, Sek88-, San93-] o objeto de controle são os processos em chão-de-fábrica, que se caracterizam por realizar transformações no material para a obtenção de produtos acabados.

O objetivo principal do trabalho é propor a metodologia MFG/PFS para aplicações em desenvolvimento de sistemas de informações em ambiente de manufatura. Nessa aplicação, o objeto de controle passa a ser o fluxo de informações e dados que circulam dentro da fábrica. Dessa forma, aplicações de natureza distintas podem ser realizadas através de uma mesma técnica padrão.

Estudo de caso real

Além de contribuir para realizar a integração entre subsistemas através do desenvolvimento de SIs, existe uma segunda motivação para esse trabalho, que é um estudo de caso real e atual na indústria brasileira. Esta é representada por um problema em uma empresa de manufatura de grande porte [Mbb93a], consistindo no desenvolvimento de um SI para supervisionar e monitorar a produção em chão-de-fábrica.

O desenvolvimento desse sistema implica aspectos bastante heterogêneos, incluindo desde o desenvolvimento de software aplicativo para o usuário, até a especificação de uma pequena rede de comunicações e de um hardware de uma unidade de aquisição de dados. Esse sistema será freqüentemente citado como exemplo no decorrer do trabalho.

Organização do trabalho

O capítulo 2 apresenta alguns conceitos fundamentais usados no decorrer do trabalho, incluindo o conceito de sistemas de eventos discretos, no qual se incluem os sistemas considerados nesse trabalho. Nesse capítulo também é descrita a proposta da metodologia para o desenvolvimento dos sistemas de informações.

No capítulo 3, serão abordados aspectos para a definição do problema e construção de modelos conceituais em PFS. Os modelos em PFS devem representar claramente o projeto do fluxo de dados (e informações) do sistema.

O capítulo 4 discute aspectos para a obtenção de modelos funcionais em MFG (modelos que especificam o controle do fluxo de dados) a partir de modelos conceituais em PFS.

No capítulo 5, será mostrado como a metodologia pode auxiliar o desenvolvimento de sistemas de informações do ponto de vista de implementação, onde deverão ser considerados os aspectos tecnológicos para a sua realização.

O capítulo 6 discute tópicos relacionados à análise dos modelos. Nesse capítulo, alguns modelos propostos serão analisados e validados.

O capítulo 7, por fim, sintetiza o trabalho, indica as contribuições e propõe alguns trabalhos futuros.

CAPÍTULO 2

CONCEITOS FUNDAMENTAIS

Nesse capítulo serão definidos sistemas de manufatura e sistemas de informações (SI) e outros aspectos importantes para o desenvolvimento desse trabalho. Além disso, será apresentado o conceito de Sistema de Eventos Discretos (SED), cujo estudo gerou a metodologia proposta para esse trabalho.

2.1 Sistemas de manufatura

2.1.1 Definição

A palavra sistema possui várias definições, entretanto, para o objetivo do trabalho, um sistema genérico será definido, sob o ponto de vista funcional e de comportamento, como um conjunto de elementos ou subsistemas independentes que se interagem de uma forma coordenada visando realizar uma determinada função. Cada um dos subsistemas, por sua vez, pode ser visto também como um sistema independente, composto de outros subsistemas, que também através de uma ação coordenada, visam realizar uma determinada função ou atividade dentro do sistema ao qual pertencem.

Um sistema de manufatura é, portanto, um sistema que se compõe de uma série de elementos independentes que se interagem visando realizar a função da manufatura, que é, sob um ponto de vista simples, realizar a atividade de transformar material bruto em produtos acabados. Naturalmente essa atividade engloba aspectos e funções bastante distintas para poder ser bem desempenhada.

A definição apresentada mostra uma característica comum nos sistemas que é a de possibilitar um estudo em diferentes níveis de abstração, o que permite uma abordagem hierárquica em seu estudo. Essa característica, que permite tratar cada subsistema como um sistema autônomo, será bastante explorada nesse trabalho.

2.1.2 Sistemas Integrados de Manufatura (SIM)

Para se melhorar o desempenho da função de um sistema de manufatura, motivado pela necessidade de se obter certas qualidades produtivas já comentadas na introdução, existem dois passos principais a serem adotados:

- (1) melhorar o desempenho de cada um dos subsistemas para melhorar o desempenho do sistema global;
- (2) estudar as relações entre subsistemas para que cada um possa desempenhar suas funções visando um desempenho melhor do sistema global, ou seja, obter um maior grau de integração entre subsistemas.

Considerando a fábrica como um todo, o estudo de integração de subsistemas objetiva atingir os chamados Sistemas Integrados de Manufatura (SIM), também conhecido por CIM - Computer Integrated Manufacturing, devido a uma característica de informatização dos subsistemas, ou ainda TFA - Total Factory Automation, terminologia bastante usada no Japão. O assunto é tema de diversas investigações [Tei87-, Bol87-, Ito88-], entretanto, existe um consenso que a grande dificuldade reside principalmente na necessidade de se integrar uma grande quantidade de sistemas de características distintas, evitando as chamadas ilhas de automação.

Para ilustrar o problema das ilhas de automação, tem-se como exemplo uma linha de produção. Um linha de produção é um sistema de manufatura na medida em que se compõe de uma série de subsistemas independentes (máquinas) cujas atividades são integradas para produzir peças acabadas (produtos) a partir de material bruto. Nesse sistema de manufatura deseja-se sempre melhorar a produtividade da linha. Pode-se decidir, então, pela substituição de algumas máquinas convencionais por outras mais eficientes (por exemplo, máquinas CNC). Entretanto, a simples substituição das máquinas sem observar as suas relações dentro do sistema global pode não melhorar efetivamente o

desempenho da linha de produção, pois existe a possibilidade de que a implantação dessas máquinas possa fazer com que outras máquinas se tornem gargalos na mesma linha .

O exemplo acima demonstra a importância do estudo da integração de sistemas, principalmente em aplicações orientadas a sistemas de manufatura.

2.1.3 Atividades de um sistema de manufatura

Sob um ponto de vista funcional, cada subsistema dentro de um sistema de manufatura pode ser visto como uma atividade com uma capacidade funcional determinada. Além disso, devido a uma característica hierárquica do sistema comentada no item 2.1.1, cada atividade, por representar um subsistema, também pode ser decomposto em outras e assim por diante. Portanto, uma fábrica ou qualquer sistema de manufatura pode, a princípio, ser descrito por um conjunto de atividades atuando coordenadamente.

Cada sistema particular possui seu conjunto próprio de atividades e dificilmente essas são coincidentes. Considerando a fábrica como um todo, tem-se como exemplo [Gom86-, Kus88-], o seguinte conjunto de atividades necessárias na fábrica, que são baseadas no conceito de ciclo de vida do produto:

- concepção de novos produtos;
- projeto de detalhamento;
- planejamento de processos;
- marketing e planejamento global da produção;
- programação e scheduling;
- controle da produção;
- fabricação ou produção.

Observa-se que da lista acima apenas a atividade de fabricação atua diretamente em material, realizando a sua transformação. As demais atividades atuam em uma outra entidade denominada informação. Tais atividades fornecem à atividade de fabricação informações de quando, quanto, como e o que produzir. A integração dessas atividades deve garantir que as informações cheguem no momento certo, sejam precisas e relevantes para o bom desempenho da atividade de produção. Além disso, um bom desempenho dessas atividades envolve necessariamente uma eficiente troca de informações entre si. É nesse aspecto que reside a importância do desenvolvimento de um sistema de informações adequado para elevar o grau de integração de sistemas de manufatura.

2.1.4 Condições e relações entre atividades

Pode-se identificar, através das considerações acima, dois itens principais sobre as quais se realizam as atividades dentro de um sistema de manufatura, que são o material e a informação. Esses dois itens determinam na fábrica os fluxos responsáveis pela interrelação entre as atividades.

Em [Has88-], pode-se visualizar a existência desses dois fluxos através do diagrama da Fig. 2-1, que é o que o autor chama de uma estrutura típica de um sistema de automação de fábrica (Factory Automation System). Observa-se que existe na figura um fluxo de material, na horizontal, sobre ao qual são aplicadas as atividades de estocagem, processamento, montagem, teste e empacotamento. Na vertical, identifica-se um fluxo de informações, que sofre atividades como "design" de produto e do processo, planejamento e gerenciamento da produção, supervisão e controle do processo e da produção e controle de qualidade.

O estudo desses fluxos e de como são distribuídos entre atividades é importante para o estudo de integração do sistema. Para tanto o sistema deve conter também condições passivas que representam elementos distribuidores. Tais condições podem

representar, por exemplo, a situação para o início ou de término de uma atividade, direcionando os fluxos do sistema e definindo as relações entre atividades.

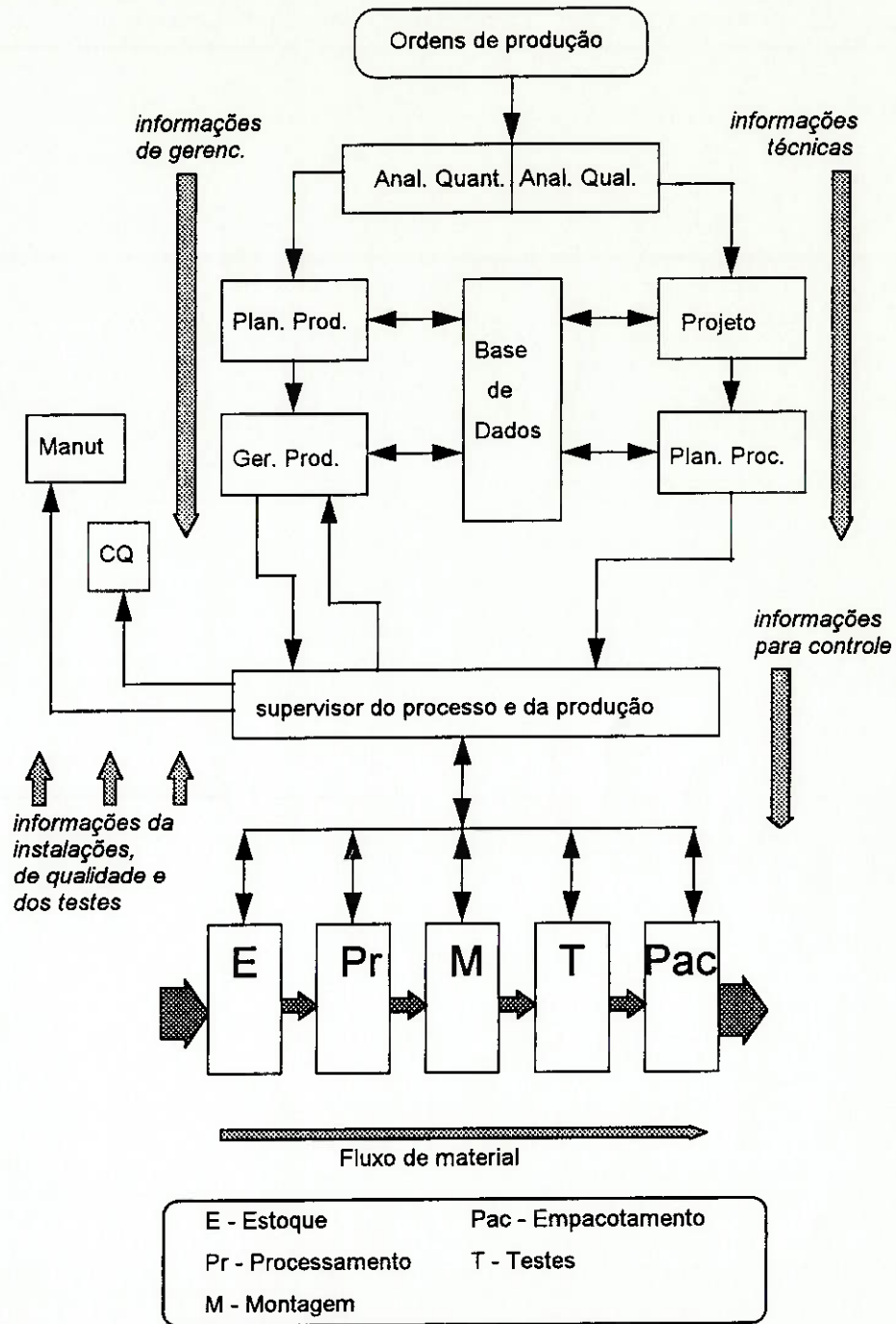


Fig. 2-1 - Factory Automation System [Has88-]

Em síntese, um sistema pode ser decomposto em dois tipos de elementos: as atividades e as condições. As atividades representam os processos e operações que constituem a funcionalidade do sistema, e as condições representam os estados internos do sistema que definem as relações entre atividades.

2.2 Sistemas de informações (SIs)

Analogamente à definição de sistemas de manufatura, um sistema de informações pode ser identificado por um conjunto de elementos, atividades e condições, que se interagem coordenadamente visando realizar as funções de fornecer, gerenciar e armazenar informações.

2.2.1 Considerações a respeito da informação

As informações desempenham uma função chave no sistema, pois estas são as matérias-primas com as quais o sistema deverá trabalhar. Os seguintes aspectos devem, então, ser considerados:

i) diferenciação entre informações e dados

Em teoria da informação, identifica-se sempre três entes, que são o transmissor, o receptor e o canal por onde flui a informação. A quantidade de informação transmitida é definida pelo grau de mudança de estado do receptor. Ou seja, quanto mais o receptor se alterar devido ao conteúdo da comunicação, maior será a quantidade de informação transmitida [Lat83-]. Dessa forma, a palavra informação só possui algum sentido quando relacionada a um receptor. "Informações" que são irrelevantes ou que não se relacionam aos seus receptores são apenas dados. Pode-se, então, formalizar uma definição de dados como um conjunto de linguagens e símbolos que se relacionam de forma a representar pessoas, objetos, eventos e conceitos, possuindo um caráter objetivo (diferente da informação, que possui caráter subjetivo). Por outro lado, informações, em SIs, podem ser

definidas como resultado de uma série de operações sobre os dados com o objetivo de modificar o receptor, aumentando a percepção e o grau de conhecimento do receptor com respeito à realidade [Bur79-].

O aspecto subjetivo da informação, em desenvolvimento de SI, torna-a importante em etapas iniciais de desenvolvimento, como a definição do problema e das necessidades do sistema. Em etapas posteriores, como na especificação, geralmente trabalha-se com dados. Essas etapas serão discorridas no decorrer do trabalho.

ii) tipos de informações

É interessante também classificar a informação em dois tipos, as informações gerenciais e as operacionais. Essa classificação será importante posteriormente para organizar a definição das necessidades do sistema. As informações gerenciais são informações que, fornecidas aos seus usuários, auxiliam atividades do tipo tomada de decisões. São informações como relatórios de produção e estado atual do processo. As informações operacionais, por outro lado, podem ser caracterizadas por ordens e requisições que irão gerar um procedimento ou uma atividade. São, por exemplo, requisição de relatórios, ordens de início de processo e alocação de recursos, entre outras.

iii) qualidades da informação

O desenvolvimento de um SI deve considerar que as informações sejam fornecidas aos usuários observando-se certos atributos desejáveis que ela deve possuir, que são, segundo [Cas88-]:

- (1) relevância, que é a característica que aproxima a informação de seu usuário. Uma das atividades principais dentro do desenvolvimento de SIs é realizar o levantamento das informações relevantes aos usuários do sistema;

- (2) precisão, que reflete o grau de correção que a informação necessita ter de acordo com o usuário. Por exemplo, a produtividade de uma máquina medida através do número de peças produzidas em um turno deve ter sua precisão de acordo com o nível da gerência que necessita dessa informação;
- (3) oportunidade, que se relaciona com o tempo em que a informação chega ao usuário. Existe um intervalo de tempo em que a informação possui alguma utilidade. Se ela for fornecida antes ou depois desse intervalo, seu valor enquanto informação pode não ser relevante.

Esses aspectos devem ser observados quando do desenvolvimento do sistema, analisados e quantificados quanto a sua importância.

Como observação final, existe ainda um aspecto prático, que é o de realizar uma avaliação de custo-benefício para obter as informações. Como todo trabalho de engenharia, às vezes a obtenção de uma informação, para um determinado caso, possui um custo muito alto tendo em vista os objetivos do sistema. Esta observação pode levar à conclusão de que obter a informação é inviável.

2.2.2 SIs aplicados em sistemas de manufatura

Considera-se inicialmente a estrutura de uma fábrica. Apesar de geralmente ser complexa, pode-se identificar, para facilidade de análise, diversos níveis de hierarquia envolvendo decisões de gerência e de controle. Seja, por exemplo, uma gerência de produção de uma determinada família de peças. Tem-se aqui uma função de controle e gerenciamento da produção a nível de chão-de-fábrica. O gerente ou qualquer outro agente responsável pela tomada de decisões tem identificado duas atividades básicas relacionadas a informações:

- (1) o gerente recebe informações de chão-de-fábrica referentes à produção, como por exemplo, tempo médio de produção por peça, quantidades produzidas e grau de utilização por máquinas, entre outras informações, sob a forma de relatórios;
- (2) o gerente também emite ordens à produção, definindo cronogramas e os processos de fabricação, baseado em requisições de níveis superiores e nas informações que recebe da produção.

No primeiro caso reconhecem-se informações do tipo gerencial, que serão utilizadas principalmente para auxiliar a atividade de tomada de decisões. Tem-se, nesse caso, um fluxo ascendente de informações, que partem do chão-de-fábrica para a gerência.

No segundo caso, observam-se informações do tipo operacional, que são utilizadas para definir atividades e operações decorrentes de ordens e requisições. Tem-se aqui, portanto, um fluxo descendente de informações, que partem da gerência em direção ao chão-de-fábrica. A Fig. 2-2 abaixo esquematiza as considerações acima:

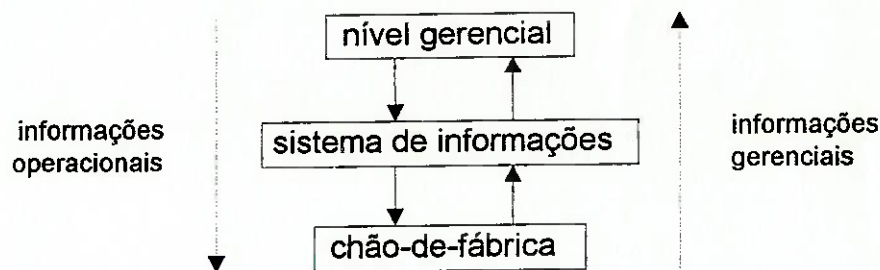


Fig. 2-2 - Fluxos de informações

Entre o nível gerencial e a produção em chão-de-fábrica existe um sistema de informações que controla os fluxos identificados na figura acima. O sistema de informações deve fornecer aos seus usuários informações com características de relevância, precisão e oportunidade, ou seja, deve-se definir corretamente quais as

informações que o usuário necessita para desempenhar suas tarefas, com a precisão necessária a nível de valores quantitativos e sem atrasos a nível de tempo.

Observa-se, no exemplo acima, que o nível gerencial pode ser representado por um dispositivo controlador. O sistema de informações, nesse caso, fornece informações para o controlador e este toma as decisões necessárias e requisita operações na produção, como faria um usuário humano. Independente da natureza de seus usuários, o SI deve fornecer informações com as qualidades de relevância, precisão e oportunidade, embora as necessidades de projeto se alterem caso a caso.

2.2.3 Exemplos de SIs

O exemplo que motiva o presente trabalho é o sistema de supervisão de produção de motores em uma indústria montadora de veículos. A gerência deseja receber informações on-line da produção sobre a produtividade e o estado de cada máquina. A gerência necessita também de relatórios estatísticos, de frequência quinzenal ou mensal, que indiquem a quantidade total de peças produzidas por máquina, seus respectivos tempos médios de processamento, além de certos índices de desempenho, como o grau de utilização de uma máquina*. Outra necessidade do sistema é a de que os dados devem ser coletados e transportados automaticamente. Dessa forma, evita-se os freqüentes atrasos gerados pelo sistema tradicional, onde os dados são coletados manualmente através de relatórios escritos e posteriormente inseridos num sistema central de processamento para a geração dos relatórios estatísticos.

Um segundo exemplo é o desenvolvimento de um controlador de célula flexível de manufatura de aplicação geral. O controlador é um dispositivo que, a partir do conhecimento do comportamento do sistema a ser controlado, tenta aproximar o mesmo de um certo comportamento desejado. Ele necessita de um SI, que, além de realizar a

* Grau de utilização é a razão entre o tempo total de processamento e o tempo total disponível para processamento da máquina.

aquisição de dados e fornecer informações ao usuário, permita emitir as ordens ao processo sob forma de sinais de controle.

O terceiro exemplo consiste no desenvolvimento de um SI para integrar os subsistemas da fábrica através do gerenciamento e controle de troca de informações entre atividades. Os exemplos acima pretendem garantir a aplicabilidade do trabalho em diversos níveis de abordagem dentro da fábrica. O primeiro exemplo será utilizado no trabalho para mostrar aspectos de modelagem utilizando a metodologia proposta. O segundo e o terceiro exemplos serão apresentados, respectivamente, nos apêndices C e D.

2.2.4 Requisitos para o desenvolvimento de SIs

O desenvolvimento de sistemas de informações freqüentemente é confundido com desenvolvimento e produção de software. No ambiente da manufatura, o desenvolvimento de SIs implica também em especificar e projetar os dispositivos e equipamentos que implementam as funções do sistema. Esses dispositivos e equipamentos, mais o software do sistema correspondem às tecnologias utilizadas para a implementação. Assim, o desenvolvimento de um SI para aplicações modernas em sistemas de manufatura deve conter o desenvolvimento dos seguintes elementos:

- (1) redes de comunicação, que correspondem ao canal por onde dados e informações irão transitar;
- (2) dispositivos e equipamentos que compõem o hardware do sistema, que podem consistir, por exemplo, de unidades de aquisição de dados industriais, controladores locais de máquinas ou ainda controladores de processos, como é o caso de controladores de células flexíveis;
- (3) software do sistema e dos dispositivos, incluindo a definição e estruturação de uma base de dados e implementação dos aplicativos do sistema.

A evolução tecnológica oferece uma série de recursos para a implementação de um SI, sendo necessário escolher a tecnologia que melhor implemente as necessidades do sistema. Por exemplo, o projeto de redes de comunicações consiste na escolha de uma arquitetura de protocolos adequada, que atenda aos requisitos de tempo, capacidade e confiabilidade do sistema para a troca de mensagens.

É ainda necessário conhecer o sistema para definir suas necessidades. Além disso, tais necessidades podem variar conforme o nível de abstração em que se está trabalhando. A estruturação da base de dados deve variar de acordo com o nível em que se trabalha. Por exemplo, o desenvolvimento de um SI no nível fábrica define características de base de dados e redes de comunicações diferentes do desenvolvimento de um SI no nível célula flexível.

2.3 Sistemas de eventos discretos (SEDs)

2.3.1 Utilização de modelos formais

A complexidade de uma fábrica e a necessidade de se analisá-la para elaborar uma especificação correta do sistema sugere a utilização de técnicas formais para a sua descrição. Através da utilização de modelos formais, consegue-se:

- (1) obter uma melhor visualização das componentes e de suas interrelações dentro do sistema. Dessa forma, facilita-se a sua compreensão e análise, tornando a sua implementação mais eficiente;
- (2) documentar o sistema mais eficientemente e padronizar a forma de comunicação entre projetistas. O uso de modelos formais evita possíveis ambigüidades encontradas em uma descrição informal, pois uma descrição informal é mais subjetiva sob o ponto de vista de interpretação se comparado às técnicas formais de descrição;

- (3) organizar os conhecimentos do projeto. A simples necessidade de confecção de modelos obriga o projetista a questionar o sistema, facilitando a organização das idéias e aumentando o seu conhecimento sobre o projeto;
- (4) realizar análises ou simulações, de forma a permitir validar o próprio modelo do ponto de vista de suas propriedades e também de permitir o levantamento de parâmetros quantitativos para o projeto do sistema.

Considerando agora sistemas de manufatura, é necessário utilizar uma técnica de modelagem que seja capaz de modelar as características encontradas nessa classe de sistemas. Além disso, deseja-se que essa técnica seja aplicável também a sistemas de comunicação e sistemas de controle para poder realizar o desenvolvimento de um SI. Neste contexto, sistemas de manufatura, sistemas de comunicação e sistemas de controle podem ser vistos como sistemas pertencentes à classe dos chamados Sistemas de Eventos Discretos (SED), que são a base para a investigação de diversos temas ligados à manufatura e automação industrial [Myg89-, Sil89-]. A sua compreensão é, então, importante para o desenvolvimento desse trabalho.

2.3.2 Sistemas de Eventos Discretos (SED) e modelos

Diferentes de sistemas a variáveis contínuas, SED é uma classe de sistemas caracterizada pela natureza dinâmica discreta em suas componentes. Em um sistema de manufatura discreta, o processo não é controlado por variáveis contínuas, como por exemplo, pressão e temperatura, como ocorre em processos em indústrias químicas, e sim pela ocorrência de eventos que determinam a evolução de estados, o que incorpora a esse tipo de sistema um caráter não-determinístico.

Observa-se que, além disso, tais sistemas possuem certas características, como paralelismo e concorrência de atividades, que inviabiliza o uso de técnicas tradicionais de

descrição, como o diagrama de blocos, PERT, gráfico de Gantt, autômatos finitos etc [Has80-].

A ferramenta de modelagem deve, então, considerar os seguintes aspectos, quando aplicada a sistemas de manufatura ou a SED em geral:

- (1) possibilitar a identificação de características de paralelismo e concorrência de eventos e atividades;
- (2) ser capaz de descrever o comportamento dinâmico do sistema através da evolução de seus estados;
- (3) representar o sistema através de uma abordagem hierárquica e de refinamentos sucessivos para que seu desenvolvimento possa ser feito em diversos níveis de abstração;
- (4) possuir ou permitir o desenvolvimento de técnicas de análise por simulação.

Existe um outro aspecto importante referente à técnica de modelagem, que é o de sua simplicidade de uso e compreensão, na medida em que qualquer linguagem de descrição exige que ela possua um alto grau de comunicabilidade entre seus usuários. Uma linguagem que possua recursos gráficos de visualização geralmente atende melhor a tais requisitos do que uma linguagem puramente descritiva.

2.4 Metodologias

2.4.1 Introdução

As metodologias fornecem o suporte necessário para o desenvolvimento de sistemas, na medida que estas organizam e estruturam as etapas de trabalho numa forma sistemática.

Para a produção de software tem-se disponível uma sistemática consolidada, chamada Ciclo de Vida e Produção de Software, no qual se baseia grande parte das

técnicas de análise estruturada de sistemas. O Ciclo de Vida e Produção de software constitui-se de uma metodologia passo a passo, conforme a Fig. 2-3 [Sek88-]:

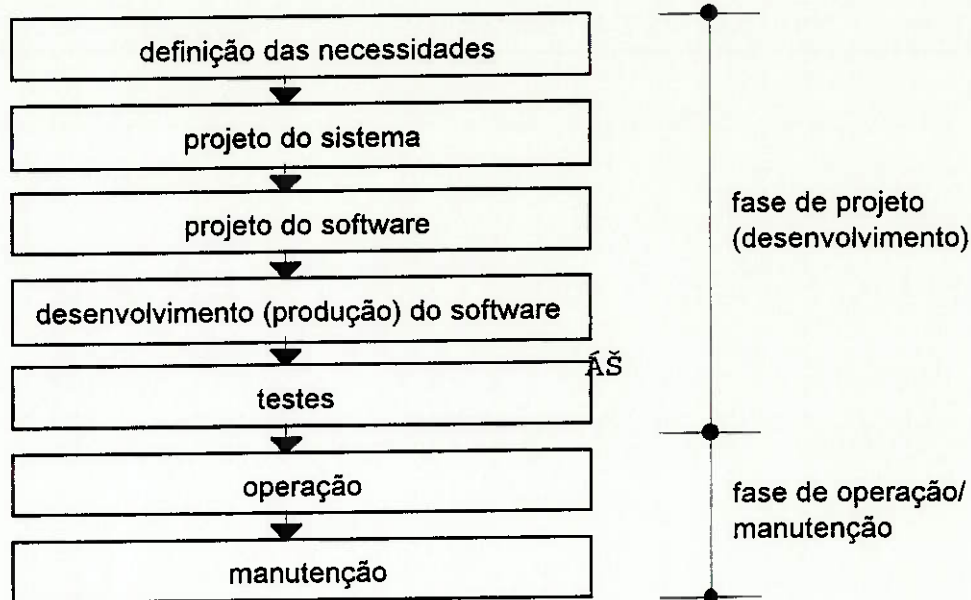


Fig. 2-3 - Ciclo de Vida e Produção de Software

Entretanto, como já comentado, os sistemas de informações em ambientes de manufatura caracterizam-se por não apenas conter elementos de software, mas também por se constituírem de outros elementos como dispositivos, equipamentos e máquinas que podem ser caracterizados como o hardware do sistema. Além disso, grande parte das metodologias baseadas no Ciclo de Vida do Software não foram desenvolvidas prevendo-se aplicações em sistemas com as características encontradas em SEDs, como é o caso dos sistemas propostos nesse trabalho.

Dessa forma motiva-se o desenvolvimento de uma nova metodologia que aborde sistemas de informações em ambiente de manufatura. Essa metodologia deverá basear-se

nas etapas do Ciclo de Vida do Software, pois os sistemas atuais de processamento digital de softwares também são uma classe de SED.

2.4.2 O estado da arte

Muitas técnicas de representação de SEDs objetivam simplesmente realizar análise a nível de avaliação de desempenho. São geralmente técnicas oriundas de pesquisa operacional, como por exemplo, técnicas baseadas em teoria de filas [Cao89-] e modelos de formulação algébrica [Coh89-]. Tais técnicas não são adequadas para realizar a especificação do sistema, pois não são orientadas para representar a ordem e seqüência de eventos e atividades durante a evolução do SED.

Existem outras técnicas que permitem representar a seqüência dos eventos de SEDs, além de possibilitarem a construção de modelos que contenham características de paralelismo e não-determinismos. São, por exemplo, linguagens do tipo CSP (Communication Sequential Process) [Sil90-, Giu91-], que é eficiente para a realização do modelo funcional, entretanto não permite derivar modelos para simulação e obtenção de parâmetros quantitativos.

As redes de comunicação também possuem uma série de linguagens para sua representação, como é o caso do LOTOS e do ESTELLE [Bru87-]. São linguagens que podem representar SEDs e também podem gerar trabalhos aplicados à manufatura. Entretanto, apesar do poder de expressão inerentes a essas linguagens, elas carecem também de ferramentas para obtenção de parâmetros quantitativos. Outro ponto negativo é que tais linguagens geralmente carecem de recursos gráficos de representação, o que as tornam pouco amigáveis.

Em 1962 foi criada a rede de Petri, que desde então têm sido objeto de inúmeras investigações até os dias de hoje. As redes de Petri consistem de uma linguagem de grafos orientados e elementos que permitem a modelagem de características de paralelismo e não determinismos. Além disso, ela permite o acompanhamento de seu comportamento

dinâmico através da evolução de seus estados internos. Tal característica torna as redes de Petri bastante adequadas para realizar controle de atividades produtivas e para implementar ferramentas analíticas, uma vez que se pode acompanhar cada um de seus estados internos e retirar informações, inclusive quantitativas, de cada um deles.

Além disso, as redes de Petri foram criadas visando o desenvolvimento de sistemas de comunicação, o que as torna uma ferramenta especialmente adequada para modelagem de redes de comunicação.

O universo de aplicabilidades das redes de Petri é vasto, incluindo aplicações em desenvolvimento de arquitetura de sistemas de hardware e de software, especificação e projeto de redes de comunicações, modelagem de sistemas produtivos e realização de controle de processos de fabricação. As redes de Petri possuem aplicação também em desenvolvimento de sistemas de processamento com arquitetura paralela, em modelos orientados a objeto e auxiliam o desenvolvimento de sistemas baseados em conhecimento.

2.4.3 O MFG e a metodologia MFG/PFS

O grande número de aplicações em redes de Petri motivou o aparecimento de uma série de extensões de acordo com as necessidades de cada tipo de aplicação. Assim, surgiram as redes temporizadas, as redes estocásticas, as redes C/E (condição/evento), as redes coloridas e as redes predicado/transição [Rei85-, Val90-, Miy90-]. Posteriormente será feita uma descrição das redes de Petri e de alguns de seus aspectos mais importantes para esse trabalho.

Derivada das redes de Petri, surgiu em 1972 o MFG - Mark Flow Graph [Has80-], voltado para aplicações em sistemas de manufatura. O MFG pode ser definido como uma rede de Petri do tipo C/E, em que se foi introduzido elementos de linguagem, além de relações de interface com o meio externo. Essas características tornam o MFG apropriado para o desenvolvimento de controladores. Entretanto, afóra a utilização desses novos

elementos, pode-se utilizar o MFG da mesma forma que as redes de Petri C/E, de modo a aproveitar todo o formalismo já consolidado em redes de Petri para o MFG.

O MFG e as redes de Petri são ferramentas eficientes para modelar o sistema a nível funcional. Além disso, pode-se derivar ferramentas para realizar análises e simulações. Entretanto para níveis mais altos de abstração é necessário recorrer a outras técnicas de descrição que facilitem a concepção do modelo do sistema e que, através de uma série de refinamentos sucessivos, possa-se atingir a descrição em MFG ou em redes de Petri. Com isso, é possível desenvolver uma metodologia que abranja todas as etapas de desenvolvimento do sistema, desde a sua concepção até a sua especificação e projeto.

Em 1988 surgiu o modelo MFG/PFS [Miy88-] em que o modelo MFG é derivado por refinamentos sucessivos a partir de um modelo PFS (Production Flow Schema). O PFS modela o sistema através de elementos ativos, que representam as atividades; de fluxos, que representam as relações entre atividades; e de elementos passivos, que modelam a distribuição dos fluxos entre atividades. O modelo MFG/PFS possui diversas aplicações e é a base para se definir a metodologia de desenvolvimento de sistemas proposta nesse trabalho.

2.4.4 A metodologia proposta

O desenvolvimento de um SI para aplicações em sistemas de manufatura deve se basear nas etapas definidas pelo Ciclo de Vida e Produção de Software e utilizar o MFG/PFS como ferramenta de modelagem. A seguir serão apresentadas as etapas do Ciclo de Vida e Produção de Software e de como estas se relacionam com a técnica de modelagem adotada. Observa-se, que, conforme já discutido anteriormente, além da especificação e da produção do software, deve-se considerar também a especificação e a implementação de dispositivos e equipamentos de que compõe o hardware do sistema.

i) definição do problema (das necessidades)

A definição do problema engloba uma série de atividades, como por exemplo, a identificação dos objetivos finais, o levantamento e a análise das necessidades de informações dos usuários e a organização das idéias através da construção de modelos. A etapa é caracterizada por abordar a modelagem do sistema em nível conceitual, ou seja, em um nível onde a representação é realizada por elementos ativos que se interrelacionam entre si através dos fluxos definidos pelo sistema. Portanto, para uma efetiva compreensão do problema e também para derivar uma especificação funcional adequada elabora-se um modelo conceitual do sistema utilizando o PFS. Outras técnicas para modelagem conceitual e definição de necessidades podem ser encontradas em [Ydv83-], embora boa parte desse trabalho seja direcionado para uma abordagem voltada a modelos de organização.

No caso de desenvolvimento de SIs, a especificação dos fluxos do sistema em PFS pode ser comparada ao projeto do fluxo de dados utilizando o Diagrama de Fluxos de Dados (DFD), encontrado nas referências [Gan79-, Dav87-]. Entretanto o DFD não permite a modelagem de estruturas típicas de SEDs, como por exemplo, paralelismo de atividades. Além disso, o DFD foi concebido para realizar projetos de sistemas de software, não sendo adequado para construir modelos a nível conceitual, onde a função básica é auxiliar a descrição e compreensão do problema.

ii) especificação funcional do sistema

A metodologia MFG/PFS define a realização do controle do fluxo após sua definição. A partir do modelo conceitual em PFS pode ser obtido um modelo MFG, onde os fluxos podem ser representados dinamicamente através de uma evolução de marcas no modelo e onde o controle dos fluxos pode ser realizado através do controle dessa evolução de marcas. Esse modelo representa a especificação funcional do sistema, de onde se pode derivar o projeto dos elementos de hardware e software.

O próprio modelo MFG/PFS propõe uma metodologia em que inicialmente são definidos os fluxos do sistema e depois é realizado o controle. Em sistemas de manufatura, a metodologia permite, através da especificação do fluxo de material, a definição das funções de controle para o projeto de CPs (Controladores Programáveis). Em desenvolvimento de SIs, o objeto de controle é o fluxo de informações ou de dados. A especificação do controle dos dados é realizado em diversos níveis e define aspectos de projeto como controle de acesso a dados e controle de transmissão e recepção de mensagens em redes de comunicação.

A especificação funcional do sistema, descrita basicamente pelos modelos funcionais, define os requisitos para o projeto do sistema, tanto a nível de software como hardware. Uma análise dos requisitos de projeto permite definir a aquisição e a alocação de equipamentos e o desenvolvimento de dispositivos e do software do sistema.

iii) projeto do hardware e software do sistema

A nível de projeto de hardware, a especificação permite definir o conjunto de dispositivos e equipamentos necessários para a implementação do SI. No caso de existir a necessidade de implementação de dispositivos, tem-se especificado não só o comportamento dinâmico desse dispositivo, mas também as funções de controle necessárias para o seu projeto.

A nível de projeto de software a especificação pode gerar, através de eventuais novos refinamentos dos modelos funcionais, algoritmos, que deverão auxiliar a geração de novos programas e servir de documentação para manutenção e testes.

Além disso, o modelo funcional pode derivar um modelo estendido para análise e obtenção de parâmetros quantitativos de projeto, tanto de software como de hardware. O modelo em MFG/PFS pode ser transcrita para uma notação própria, que é a estrutura de entrada para uma ferramenta de simulação baseada em MFG, que será utilizada nesse trabalho.

iv) produção de hardware e software do sistema

O desenvolvimento do sistema a nível de produção final depende da habilidade dos projetistas e programadores de realizarem um sistema eficiente, que seja fácil de testar, operar e manter. A metodologia proposta prevê que grande parte do trabalho seja realizado a nível conceitual e de especificação funcional, o que deve facilitar diretamente a tarefa de produção do software e hardware, além da integração de equipamentos adquiridos, alocados ou implementados para o sistema.

Em síntese, pode-se esquematizar a metodologia MFG/PFS através da Fig.2-4 abaixo:

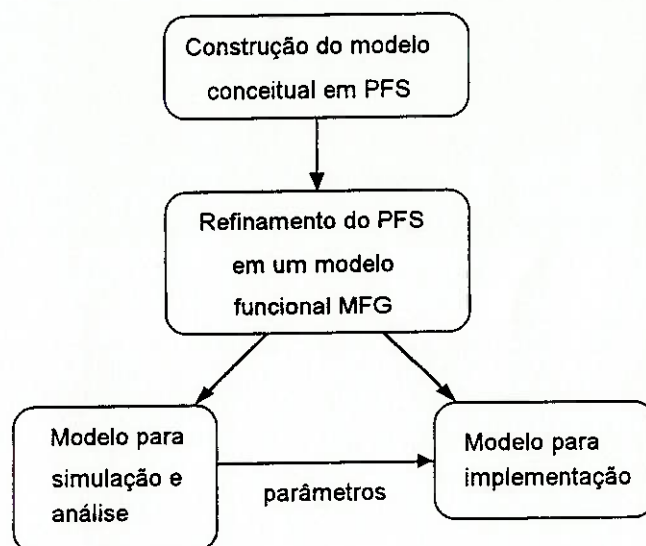


Fig. 2-4 - Etapas da metodologia proposta

A metodologia MFG/PFS tem sido utilizada eficientemente em modelagem de sistemas produtivos e especificação de dispositivos de controladores, conforme cita as referências [Myg88-, Sek88-, San93-]. O presente trabalho propõe a aplicação da mesma metodologia em desenvolvimento de SI em ambiente de manufatura, verificando se os recursos para modelagem e análise do MFG/PFS, eficientes quando o objeto de controle é

o fluxo de material, são eficientes também na especificação do controle do fluxo de informações. Dois aspectos principais motivam essa possibilidade:

- (1) existem semelhanças entre o fluxo de material e o fluxo de dados, pois ambas entidades podem ser transportadas (ex., movimentação de material por robôs e de comunicação de dados por rede), processadas e armazenadas (ex., em "palletes" no caso de material e em banco de dados no caso das informações);
- (2) a rede de Petri, cujo princípio derivou o MFG, foi inicialmente concebida para aplicações em sistemas de informações, como ferramenta para modelagem de processos concorrentes [Pet81].

Um exemplo de aplicação da metodologia em fluxo de material é apresentado no apêndice A. Nos próximos capítulos será feita uma revisão da metodologia, com aplicação e exemplos no fluxo de informações. Os exemplos de aplicação desse trabalho foram elaborados tomando como base modelos existentes de sistemas produtivos em MFG, assim como modelos existentes de SIs em redes de Petri. A maioria destes modelos foram desenvolvidos durante a elaboração desse trabalho.

CAPÍTULO 3

CONSTRUÇÃO DO MODELO CONCEITUAL

Conforme a metodologia proposta, a primeira etapa consiste na definição das necessidades e na construção dos modelos conceituais em PFS. A construção do modelo conceitual auxilia a organização das idéias e do conhecimento do sistema, permitindo a realização de análises e possibilitando a derivação imediata das especificações e dos modelos funcionais. O trabalho tem por objetivo, nessa fase, estudar a etapa de definição do problema e fornecer uma base para a elaboração de modelos em PFS.

3.1 A definição das necessidades

3.1.1 Conceito

Considera-se inicialmente os subsistemas de que se compõem um sistema genérico qualquer. Cada subsistema, inicialmente, pode ser considerado como uma "caixa preta", não se descrevendo a um primeiro nível de abstração o seu comportamento interno. Nesse nível, é importante apenas considerar as interrelações entre subsistemas, definindo as suas respectivas funções dentro do sistema global. No caso de SIs, estas interrelações podem ser identificadas como fluxos de informações na entrada e saída em cada subsistema. A fase de definição de necessidades consiste na determinação e caracterização qualitativa e quantitativa de cada uma dessas informações de entrada e saída.

Seja, por exemplo, o sistema de supervisão da produção de uma indústria montadora de veículos, que constitui um caso exemplo para esse trabalho. Considere o sistema como uma grande "caixa preta". A definição das necessidades de informações consiste em definir a relação dessa caixa com suas entidades externas e com o usuário. Um levantamento inicial pode definir, por exemplo, os seguintes tipos de informações:

- (1) as entradas do sistema podem ser definidas como informações do tipo requisições de relatórios da produção e requisição do status das máquinas sob o alcance do SI (informações operacionais);
- (2) na saída do sistema podem ser identificadas, entre outras, informações como emissão dos relatórios, exibição do status atualizado das máquinas, mensagem de recusa de serviços (informações gerenciais);
- (3) é possível identificar também uma interface com o processo, o que define um fluxo de sinais de controle, que o SI deverá processar para gerar as informações ao usuário.

Além da identificação dos tipos de informações, deve-se também caracterizar as mesmas a nível quantitativo. É importante, portanto, levantar alguns parâmetros como, por exemplo, frequência média para a emissão de cada um dos relatórios e tempos médios relativos ao processo. Observa-se que, após a definição das necessidades de informações, ou seja, após a definição das relações com entidades externas, torna-se mais fácil especificar o comportamento interno da "caixa preta" e, conseqüentemente, do sistema a ser desenvolvido.

3.1.2 Atividades para a definição das necessidades

É necessário, então, realizar um levantamento de dados para a definição das necessidades, observando-se os objetivos finais do sistema. As necessidades são levantadas e analisadas para se obter uma definição mais precisa e correta. A elaboração de um modelo adequado e correto nessa fase da metodologia é de fundamental importância, não só porque auxilia a organizar as idéias e os dados coletados no levantamento, mas também porque as outras etapas dentro da metodologia serão realizadas baseadas no refinamento dos modelos construídos. Os modelos construídos são avaliados e revisados, podendo exigir a execução de novos levantamentos ou até mesmo

uma redefinição dos objetivos finais, se, por exemplo, forem detectadas falhas estruturais no sistema que inviabilizem o desenvolvimento de um SI.

Baseado nas considerações anteriores, define-se um fluxograma (ou um ciclo) de atividades conforme ilustrado na Fig. 3-1.

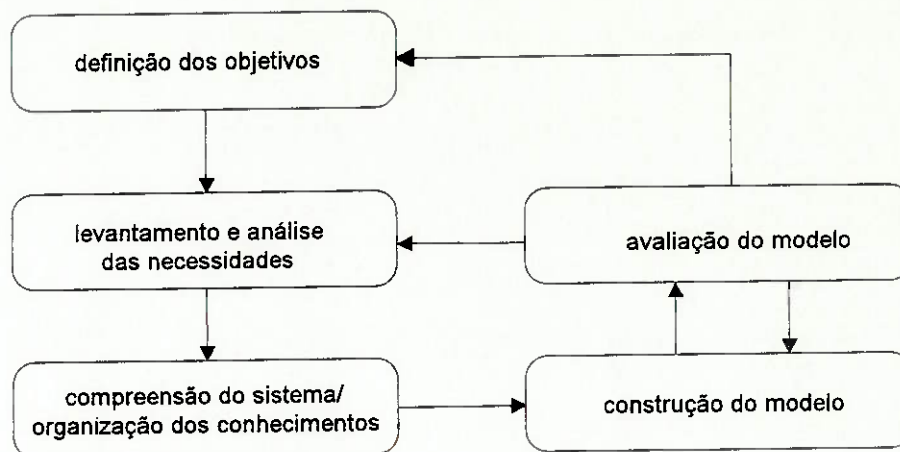


Fig. 3-1 - Fluxograma de atividades para a definição das necessidades

É importante que o objetivo final do sistema seja bem definido, pois o mesmo será utilizado como referência a uma série de decisões a serem tomadas no decorrer do desenvolvimento. Em sistemas de informações, os objetivos geralmente são de fornecer ao usuário, informações com características como relevância, correção e oportunidade. Por exemplo, no sistema de supervisão de uma montadora de veículos o objetivo final do sistema pode ser o de agilizar a tomada de decisões do usuário através do fornecimento de informações oportunas.

Uma vez identificado os objetivos, as necessidades são levantadas e analisadas e todo o conhecimento é organizado em modelos conceituais. Os modelos são avaliados, revisados e novos levantamentos são realizados até que se obtenha os modelos adequados.

As avaliações são realizadas através de uma análise dos modelos. Nessas avaliações podem ser detectadas falhas na construção de modelos, falhas no levantamento

de necessidades ou até mesmo falhas estruturais no sistema, o que pode implicar na redefinição dos objetivos. Além disso, deve ser feita uma análise de viabilidade para que o desenvolvimento do sistema possa ser validado a nível de relação custo-benefício.

3.1.3 Características da etapa

A etapa de definição de necessidades caracteriza-se por ser muito importante dentro da metodologia, pois o desempenho das etapas subsequentes depende diretamente desta. Um erro nessa fase dificilmente é detectado em fases posteriores ao desenvolvimento, sendo visível apenas na fase de operação, quando as necessidades do usuário não forem plenamente satisfeitas. O desenvolvimento do sistema partindo-se de uma redefinição das necessidades implica em custos elevados de correção, devido a um efeito acumulativo. Assim, por este efeito, um erro detectado no início do desenvolvimento em geral tem um custo de correção bem menor do que se encontrado nas fases finais, conforme as referências [Gan78-, Dav87-], que apresentam gráficos de custo de correção por fase de projeto.

Uma característica que distingue essa fase das demais é a necessidade de organizar e tratar relações humanas, pois a atividade é conduzida através de relações entre projetista, usuário, gerentes, fornecedores de equipamentos, entre outros. O ser humano tem considerações e necessidades diferentes conforme as circunstâncias e a forma de transmissão, geralmente por linguagem verbal, que pode conter inconsistências e ambigüidades. Como já visto, uma das formas de abordar o problema é através da padronização de uma técnica formal de descrição e documentação para realizar a comunicação e conciliar diferentes necessidades.

Outra característica consiste na necessidade de se conhecer bem o sistema ou processo em que se vai trabalhar. Essa característica é fundamental para a definição e análise das necessidades do sistema, e também para a concepção e especificação das

soluções. As técnicas de descrição formal permitem a organização do conhecimento através da criação de modelos.

Por fim, a definição e análise das necessidades podem implicar em críticas e reconsiderações ao sistema e aos objetivos, na medida em que permite a detecção de erros estruturais no próprio sistema e permite também determinar a viabilidade dos objetivos.

3.1.4 Levantamento das necessidades

O levantamento é caracterizado como uma coleta de informações necessárias para aumentar o conhecimento do sistema e realizar a definição das necessidades. Isso é feito, dentro da metodologia, através da construção de modelos. A realização do levantamento pode ser feita de duas formas principais, que são o levantamento por documentação ou por entrevistas.

i) levantamento por documentação

Uma forma de se realizar o levantamento por documentação é fazer um estudo dos relatórios já existentes utilizados pelo usuário. Desses relatórios pode-se retirar algumas de suas necessidades sem a realização da entrevista. A análise desses relatórios pode também fornecer indicações para outras possíveis necessidades não explicitadas pelo usuário. Obviamente, os relatórios já existentes podem não suprir plenamente as necessidades de informações do usuário e nesse caso a realização da entrevista pode ser necessária.

O levantamento por documentação pode também ser realizado através da leitura de manuais e documentação técnica. Seja o exemplo do desenvolvimento de um controlador de célula. O objetivo é desenvolver um SI que permita fornecer informações adequadas ao controlador para que esse possa realizar as ações adequadas de controle. A definição das necessidades é dificultada, pois não é possível a realização de entrevistas. A

solução, nesse caso, é obter um maior conhecimento das atividades do controlador a nível funcional para, através do conhecimento de suas funções, definir suas necessidades.

ii) levantamento por entrevistas

Nesse tipo de levantamento, a definição das necessidades torna-se mais específica (e subjetiva) de acordo com o usuário. Um dos problemas mais comuns em entrevistas é o de interpretação. O usuário nem sempre é capaz de traduzir em palavras suas necessidades e o projetista, por não estar familiarizado com o dia a dia do processo, nem sempre possui a percepção correta ou exata do que foi pedido, o que pode gerar erros de interpretação. O usuário geralmente trata suas necessidades de uma forma intuitiva, o que dificulta explicitá-las formalmente. Esse aspecto em sistemas de software motivou o aparecimento da técnica de prototipagem e a utilização de modelos formais. Nesse caso a utilização de modelos deve facilitar a descrição formal das necessidades, podendo servir de linguagem comum de comunicação entre o usuário e o projetista.

Convém ainda comentar um outro problema bastante comum com relação às entrevistas, que consiste nas eventuais resistências dos entrevistados para fornecer informações. Embora não seja objetivo desse trabalho tratar tais resistências, é muito importante considerá-las quando do desenvolvimento real dos projetos de SIs. Podem ser encontradas na referência [Bor86-] recomendações para o entrevistador minimizar tais resistências durante a entrevista.

3.1.5 Análise das necessidades

Após o levantamento das necessidades, é necessário adequá-las considerando os objetivos finais do sistema, e analisá-las para verificar a necessidade de novos levantamentos. Pode-se então, classificar as necessidades em:

- (1) necessidades não suficientemente claras;
- (2) necessidades contraditórias;
- (3) necessidades incorretas;
- (4) necessidades não realizáveis com a tecnologia disponível;
- (5) necessidades não realizáveis por razões de prazos ou custo-benefício.

A classificação acima deve facilitar a avaliação do levantamento e da definição das necessidades. Necessidades não claras, contraditórias e incorretas podem implicar na realização de novos levantamentos. As necessidades não realizáveis deverão ser discutidas e analisadas, podendo implicar em reavaliações do objetivo final do sistema.

3.2 A construção de modelos conceituais

Grande parte desse capítulo é dedicado à elaboração de modelos conceituais em PFS. A metodologia propõe a especificação dos fluxos de informações ou dados através de um refinamento passo a passo das atividades. O problema pode ser dividido em módulos e em diversos níveis de abstração, facilitando a sua definição e facilitando também a divisão do trabalho entre os projetistas. Para auxiliar a elaboração dos modelos, o trabalho apresenta algumas estruturas comuns em modelos PFS e abordagens para sua concepção. Por fim, apresenta-se um exemplo para ilustrar a execução da etapa de definição e a elaboração de modelos através da especificação dos fluxos do sistema.

3.2.1 A linguagem PFS - Production Flow Schema

O PFS - Production Flow Schema - é uma linguagem concebida para descrever modelos conceituais de sistemas produtivos a eventos discretos [Myg88-]. É uma linguagem baseada em grafos orientados definida por dois elementos básicos, que são as atividades e os elementos de distribuição. As atividades são representadas entre dois colchetes e os elementos de distribuição por círculos, conforme a Fig.3-2 (a).

Além desses elementos, existem também arcos orientados que realizam a conexão entre as atividades e os elementos de distribuição. Tais arcos representam os fluxos identificados dentro do sistema. O PFS distingue dois tipos de fluxos: fluxo principal, representado por arcos conectados aos colchetes, e fluxo secundário, representado por arcos entre os colchetes, conforme a Fig.3-2 (b).

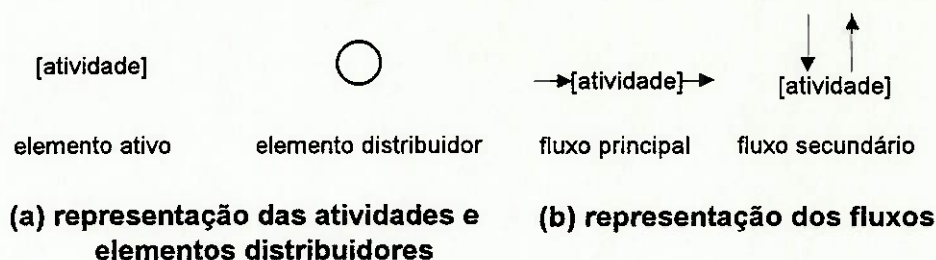


Fig.3-2 - Elementos do PFS

Cada atividade entre colchetes pode ser refinado em sub-atividades com as suas respectivas interações. O modelo permite, portanto, que se trabalhe com estruturas hierárquicas e que o desenvolvimento possa ser feito através de refinamentos sucessivos. Permite também a modularização, onde o sistema pode ser decomposto para que possa ser desenvolvido em partes e por pessoas diferentes.

O PFS é um modelo interpretado de grafos orientados, ou seja, os elementos de linguagem são associados a entidades do sistema real, permitindo a representação do sistema a nível conceitual, ou seja, o sistema é representado através da interrelação de seus elementos funcionais, definindo o que o sistema vai fazer sem necessariamente definir como será feito. Tais interrelações no PFS são representados através dos fluxos, o que permite dizer que o PFS realiza a especificação dos fluxos do sistema.

Por exemplo, em modelagem de sistemas produtivos para aplicações em sistemas de controle, o PFS realiza a especificação do fluxo de material do sistema, onde as atividades podem representar operações sobre o material, como processamento e

transporte, e os elementos passivos podem representar estruturas como "buffers" ou estoques intermediários de peças. Após a definição dos fluxos, pode-se especificar o controle do sistema e obter o modelamento a nível funcional, utilizando um MFG interpretado, que pode ser derivado diretamente do PFS.

Da mesma forma, em desenvolvimento de SIs, pretende-se utilizar a metodologia MFG/PFS, inicialmente modelando-se em PFS os fluxos de informações e dados do sistema e depois realizando o controle desses fluxos através de um modelo interpretado híbrido MFG/PFS.

3.2.2 Representação de características de SEDs em PFS

Os sistemas de manufatura e os SIs abordados nesse trabalho apresentam algumas características próprias de SEDs, que podem ser representadas em PFS. As estruturas seguintes devem facilitar a concepção de modelos em PFS:

i) noção de sequência de eventos e atividades

A estrutura da Fig.3-3(a) representa um sequência de 3 atividades, onde o início de uma só pode ocorrer ao término da anterior. Os colchetes "[" e "]" representam, respectivamente, início e fim de atividade. Os elementos distribuidores podem representar a disponibilidade de informações para início da próxima atividade.



Fig. 3-3(a) - representação de sequência

ii) concorrência de atividades

Quando existem duas ou mais atividades com fluxos derivados de ou convergentes a um único elemento distribuidor, existe uma concorrência de atividades, conforme

ilustrado na Fig.3-3(b). Essa estrutura indica que apenas uma das atividades será realizada ou finalizada na seqüência.

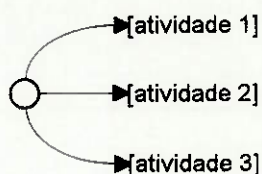


Fig. 3-3(b) - concorrência de atividades

iii) paralelismo de atividades

Quando existem dois ou mais elementos distribuidores com fluxos derivados de uma atividade, ocorre um paralelismo entre as atividades na seqüência desses elementos distribuidores. A Fig. 3-3(c) mostra que as atividades 2 e 3 iniciam independentemente ao término da atividade 1 e, portanto, podem ser realizadas paralelamente.

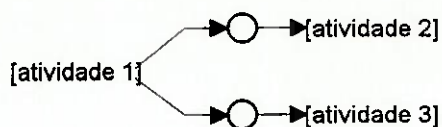


Fig. 3-3(c) - paralelismo de atividades

iv) sincronização entre atividades

Por exemplo, a atividade 3, no caso da Fig.3-3(d) só pode iniciar após o término das atividades 1 e 2.

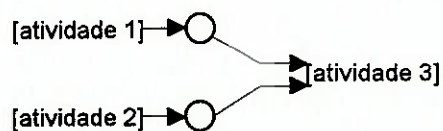


Fig. 3-3(d) - sincronização de atividades

3.2.3 Relações através de fluxos secundários

As relações entre as atividades podem ser definidas através dos fluxos do sistema. O fluxo principal define estruturas como as apresentadas no item 3.2.2 (seqüência, paralelismo e concorrência). Os fluxos secundários podem definir relações como a comunicação assíncrona, comunicação síncrona e chamadas de procedimentos externos, como ilustra a Fig. 3-4.

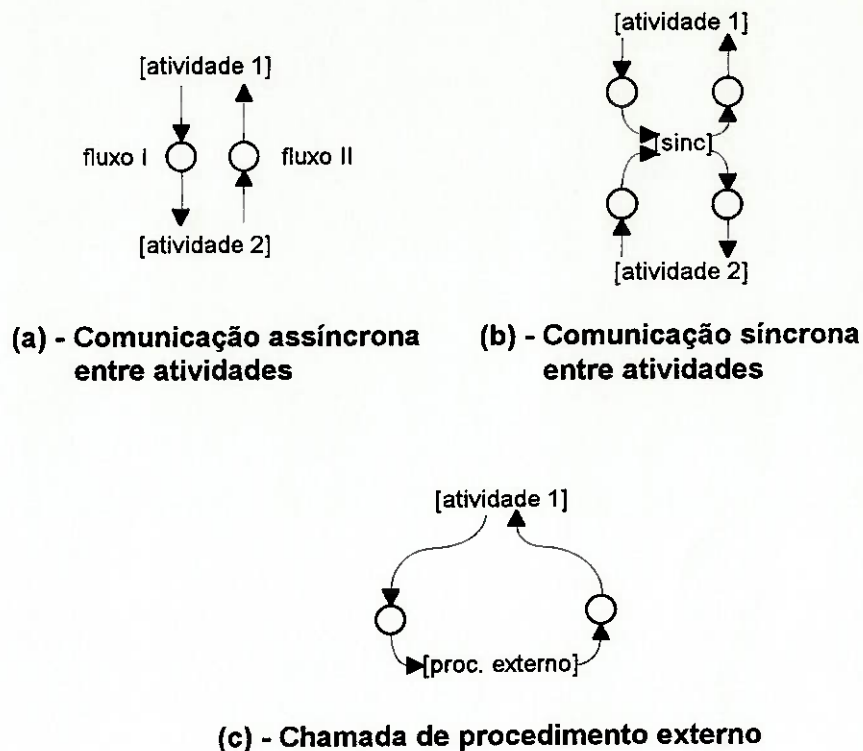


Fig. 3-4 - Interpretações típicas dos fluxos secundários

A comunicação assíncrona, ilustrada na Fig. 3-4(a), pode representar trocas de informações entre atividades. Por exemplo, a atividade 1 envia uma requisição de informações para a atividade 2, representado pelo fluxo I, e a atividade 2 envia a resposta à atividade 1 através do fluxo II. A chamada de procedimento externo é representada na Fig.3-4(c) e geralmente é utilizada para representar a utilização de um mesmo processo

por várias atividades. O procedimento externo poderia ser, por exemplo, um gerenciador de banco de dados, que geralmente é requisitado por várias atividades dentro do sistema. Por fim, a Fig. 3-4(b) representa a estrutura de uma comunicação síncrona, onde, no caso, as atividades 1 e 2 se sincronizam através da atividade "sync" para realizar a comunicação.

3.2.4 Refinamento das atividades

O PFS permite uma abordagem hierárquica, de forma que o modelo pode ser detalhado passo a passo através de refinamentos sucessivos, organizando o problema e dividindo-o em diversos níveis de abstração. As estruturas apresentadas acima podem ser utilizadas no refinamento das atividades, entretanto devem ser tomados alguns cuidados. Recomenda-se que na elaboração dos modelos, sejam utilizados blocos bem formados, como ilustrados na Fig. 3-5.

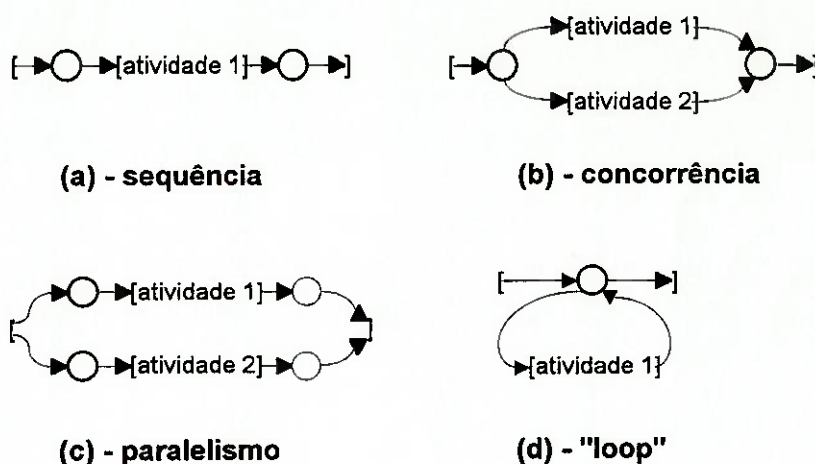


Fig. 3-5 - Refinamento das atividades

Num bloco bem formado, sempre que o fluxo é dispersado em dois ou mais, seja em situação de paralelismo ou de concorrência, ele deve convergir respectivamente, em forma de uma sincronização ou em convergência para um elemento distribuidor, conforme representado nas Fig. 3-5(b) e 3-5(c). Observe também a Fig. 3-5(d). Nesse caso o

elemento distribuidor possui duas possibilidades, que é o de realizar a atividade interna l ou de encerrar a atividade global, o que caracteriza um loop (ou um ciclo repetitivo).

Na prática o refinamento deve utilizar combinações das estruturas acima.

3.2.5 Abordagens para a construção de modelos

Existem duas abordagens para realização de modelos e usualmente diz-se que são complementares. Essas abordagens podem ser ilustradas através da representação de dois tipos de diagramas usados para a construção de modelos conceituais [Val90-, Ydv83-]:

- (1) os "atigramas";
- (2) os "datagramas".

Na construção por atigramas, os elementos principais da linguagem são as atividades que se interrelacionam através de arcos que representam os fluxos de dados ou informações. O exemplo da Fig.3-6(a) abaixo mostra uma representação típica com abordagem atigrama:

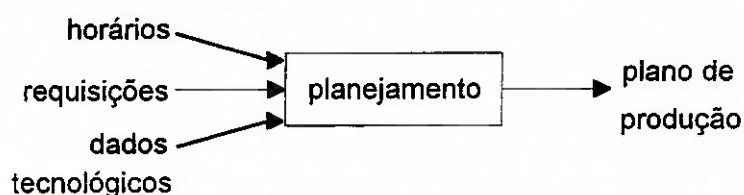


Fig. 3-6(a) - Exemplo de representação por "atigramas"

Observa-se que no exemplo da Fig.3-6(a), a atividade de planejamento necessita das informações "horários", "requisições" e "dados tecnológicos" para ser realizada e produz em seu término a informação "plano de produção". Nesse caso o elemento

principal é a atividade planejamento e a as informações correspondem a fluxos que entram e saem de cada atividade.

Na construção por datagramas, a ênfase é dada aos elementos passivos (dados e objetos) que se relacionam através de ações, eventos ou atividades (elementos ativos), conforme exemplo abaixo da Fig.3-6(b):

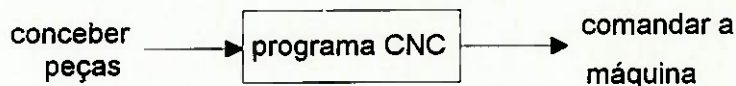


Fig. 3-6(b) - Exemplo de representação por "datagramas"

Nesse caso, o elemento principal é a informação "programa CNC", a entrada é a atividade "conceber peças" e a saída é a atividade "comandar a máquina".

O PFS possui elementos de linguagem que podem representar a atividades e dados, permitindo a construção do modelo através das duas abordagens. A princípio, a concepção de modelos em PFS segue uma abordagem de atigramas, onde inicialmente são definidas as atividades que compõe a funcionalidade do sistema. Entretanto, verifica-se que certas estruturas são melhor concebidas se for adotada uma abordagem de datagramas, pois os elementos passivos podem ser associados a objetos como banco de dados e buffers, que constituem uma referência importante na criação de modelos de sistemas de informações.

Essa característica provê o PFS de uma certa flexibilidade ao projetista, que dependendo de seu grau de conhecimento, pode dispor das duas abordagens para a construção do mesmo modelo.

3.3 Aplicações da etapa em desenvolvimento de SIs

As considerações apresentadas fornecem uma base para se construir modelos em PFS de sistemas genéricos caracterizados como SEDs. Convém, então, estudar alguns aspectos relacionados a aplicações de SIs em sistemas de manufatura.

3.3.1 Aplicações de SIs em sistemas de controle e supervisão

Seja a seguinte estrutura descrita na Fig. 3-7.

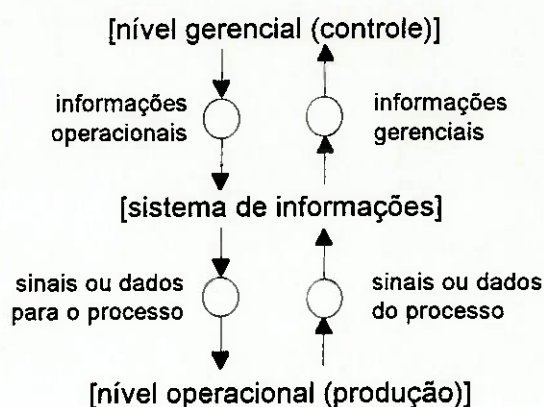


Fig. 3-7 - Aplicação de SI em sistemas de controle e supervisão

O nível gerencial representa entidades como usuários que supervisionam e controlam a produção ou então dispositivos tipo controladores de processo. As entidades a esse nível necessitam de informações gerenciais para definir suas ações. As ações são comunicadas ao processo através das informações operacionais.

O nível operacional representa as entidades que realizam as tarefas solicitadas pelo nível gerencial. Num sistema de manufatura, tais entidades estão associadas a elementos do sistema produtivo, como máquinas e dispositivos, responsáveis pelas operações sobre o fluxo de material da fábrica.

Para realizar a troca de informações entre os dois níveis, define-se um SI, responsável por fornecer as informações gerenciais ao nível gerencial (como por exemplo, relatórios de produtividade ou o status do processo) e fornecer informações operacionais ao nível operacional (como por exemplo, ordens de serviço e requisições de início de ciclo de produção ou de transporte de material). Para tanto, o SI deve gerenciar a aquisição e emissão de dados e sinais de controle do e para o processo.

A definição das necessidades é realizada, inicialmente, definindo quais informações gerenciais ou operacionais serão oferecidas ao usuário na forma de requisições ou serviços. É necessário, portanto, verificar os objetivos que motivam o desenvolvimento do SI e realizar o levantamento das necessidades. É necessário também definir os sinais que deverão realizar a interface do SI com o nível operacional, baseadas no levantamento das necessidades. A elaboração de um modelo do nível operacional, de acordo com as necessidades do projeto, pode ser bastante útil para organizar a definição dos sinais de controle. A próxima etapa é construir o modelo conceitual do SI.

3.3.2 Elementos de um sistema de informações

Genericamente o desenvolvimento de um SI pode ser decomposto em três partes distintas, conforme representado na Fig. 3-8.

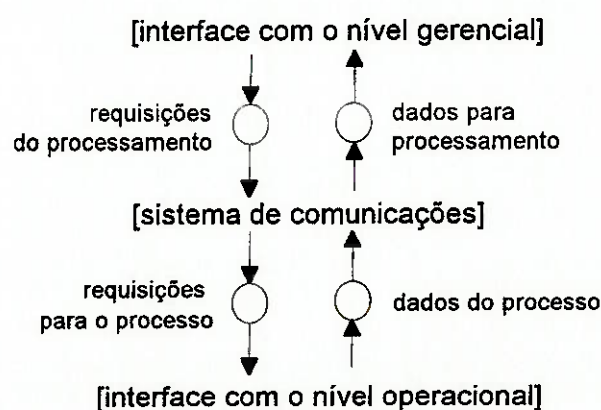


Fig. 3-8 - Partes de um sistema de informações

A interface com o nível gerencial representa geralmente o processamento dos dados provenientes do nível operacional fornecidos pelo sistema de comunicações, sendo que a especificação desse módulo deve derivar o desenvolvimento de um software. A interface com o nível operacional geralmente define uma especificação de um hardware que realiza o interfaceamento com o processo, captando os sinais necessários e eventualmente emitindo sinais para a operação das máquinas e equipamentos do processo.

O sistema de comunicações é geralmente representado por uma rede de comunicações e, como tal, sua especificação é realizada através da definição de uma arquitetura de protocolos. O conhecimento da tecnologia de redes é muito importante para o desenvolvimento de SIs, e o assunto será abordado com maiores detalhes no Cap. 5 desse trabalho. O modelo do sistema de comunicações, entretanto, pode ser refinado de forma que os requisitos do projeto de redes possam ser estruturados, permitindo uma abordagem em níveis hierárquicos, conforme recomendado pela ISO* [Gom86-, Tan89-].

3.4 Exemplo de aplicação da metodologia

Para ilustrar os conceitos abordados, a metodologia será aplicada nesse capítulo na construção de modelos conceituais em PFS do primeiro caso exemplo apresentado no item 2.2.3. O exemplo consiste no desenvolvimento de um SI que deve gerar informações para auxiliar a tarefa de supervisão da produção de motores de uma indústria montadora de veículos.

3.4.1 Identificação dos objetivos do sistema

A primeira etapa da definição do problema consiste na identificação dos objetivos finais do sistema. O objetivo principal, nesse caso, é agilizar a tomada de decisões através

* A arquitetura de redes é estabelecido pela norma ISO 7492 - Information processing systems - Open Interconnection, Basic Reference Model. Essa norma é conhecida como modelo de referência OSI (Open Systems Interconnection) e é a base para a padronização de arquiteturas de redes de comunicações.

do fornecimento de informações aos usuários, com características desejáveis como relevância, precisão e principalmente oportunidade. Por exemplo, de posse de relatórios da produtividade de cada máquina, a gerência pode detectar a existência de operações gargalos, podendo definir uma realocação de máquinas, uma redefinição do processo ou até mesmo uma aquisição de máquinas novas.

3.4.2 Levantamento das necessidades

Definidos os objetivos, devem ser realizados os levantamentos das necessidades e de dados relevantes para o desenvolvimento do sistema. As entrevistas, nesse caso, são fundamentais e definem as necessidades específicas de informações dos usuários a serem fornecidas pelo SI. Nesse exemplo, constatou-se a necessidade de dois tipos básicos de informações:

- (1) informações on-line da produção, onde a qualquer instante o usuário pode monitorar a quantidade de peças produzidas e operações realizadas em cada máquina, com os respectivos tempos médios de processamento de cada peça e ainda verificar a condição de cada máquina, se parada ou operando;
- (2) relatórios estatísticos, de frequência quinzenal ou mensal, que mostre para cada máquina a quantidade total de peças produzidas e seus respectivos tempos médios de processamento, além de certos índices de desempenho, como o grau de utilização, citado no Cap.2.

Além dessas, pode-se haver a necessidade de se gerar outras informações, como por exemplo, informações para controle de qualidade (ex., quantidade de peças refugadas) e tempos médios para a preparação de cada máquina.

3.4.3 Organização dos conhecimentos

O sistema a ser desenvolvido pode ser considerado uma aplicação típica de SI em sistemas de controle e supervisão, e pode ser modelado inicialmente conforme a estrutura da Fig. 3-7. O nível gerencial é representado por um usuário e o nível operacional pelas máquinas. Entretanto, para o caso específico desse exemplo, o modelo é modificado, pois a interface entre o sistema de informações e o nível operacional deve ser representada apenas por um fluxo ascendente partindo do nível operacional para o SI e não por dois, pois o SI apenas monitora a produção sem emitir sinais de controle. Ou seja, o sistema pode ser comparado a um controle de malha aberta, contendo apenas a aquisição de dados sem atuar no processo.

É interessante realizar a esse nível a modelagem da funcionalidade do nível operacional, pois dessa forma pode-se definir melhor os sinais de controle necessários para a especificação do sistema. Nesse exemplo, deseja-se apenas observar e registrar o status de cada máquina a cada instante, se operando ou parada, e contabilizar o tempo de processo de cada operação. Para esse objetivo, cada máquina pode ser descrito através do seguinte modelo:

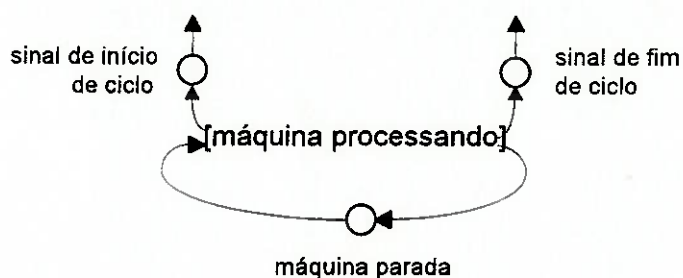


Fig. 3-9 - Representação de uma máquina

Para os objetivos desse sistema, apenas a utilização de dois tipos de sinais (sinais de início e fim de ciclo, conforme ilustra a Fig. 3-9) provenientes do processo são suficientes para caracterizar o estado da máquina e gerar as informações para o usuário.

Conforme já citado no item 3.3.2, o SI pode ser dividido em três módulos funcionais, que são os módulos de interface com os níveis gerencial (usuário) e operacional (máquinas) e um sistema de comunicações (rede). Cada um desses módulos será discutido separadamente a seguir.

3.4.4 Modelo da interface com o usuário

O modelo do módulo de interface com o usuário pode ser representado conforme a Fig. 3-10. Para facilitar a visualização do modelo, os fluxos secundários externos às atividades foram representados por linhas pontilhadas e os fluxos internos por linhas contínuas.

Pelo modelo, a interface com o usuário apresenta duas atividades principais que são executadas em paralelo, que são o processamento de informações e a interface com o sistema de comunicação. O processamento das informações representa o software de aplicação do usuário e inicia-se quando se recebe uma requisição proveniente do usuário, representado na figura por um fluxo secundário externo de entrada na atividade. Ao término dessa atividade, pode-se identificar um outro fluxo secundário, que representa as informações que o usuário requisitou. A interface com o sistema de comunicações representa o fornecimento de dados atualizados para o processamento de informações. Essa atividade envia requisições ao nível operacional através do sistema de comunicações e recebe dados atualizados do processo.

A relação entre as duas atividades é definida pela utilização compartilhada de um Gerenciador de Banco de Dados (GBD). Na figura, ambas as atividades realizam uma chamada a um mesmo procedimento externo, o GBD.

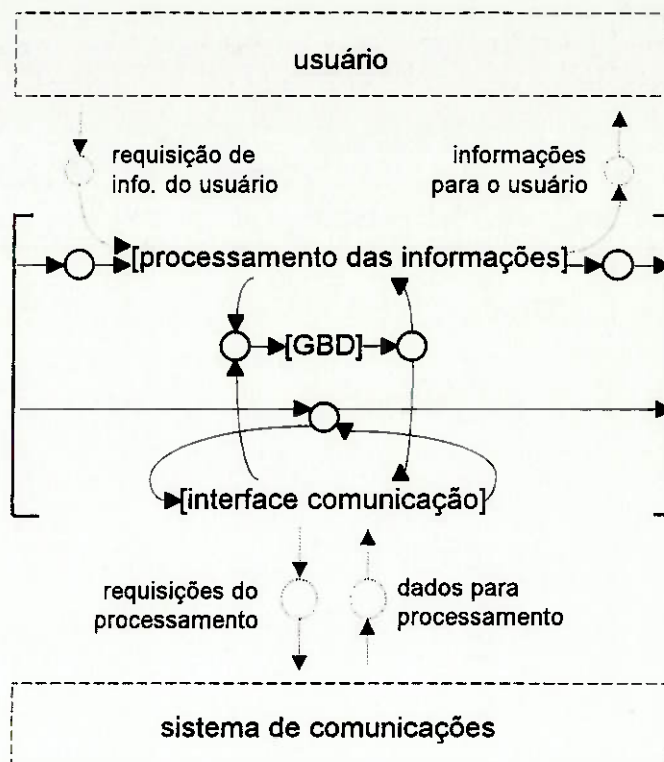


Fig. 3-10 - Módulo de interface com o usuário

Assim, o processamento de informações realiza uma chamada ao GBD toda vez que necessitar realizar uma consulta ao banco de dados e a interface de comunicação também realiza uma chamada ao GBD toda vez que possuir dados para serem atualizados.

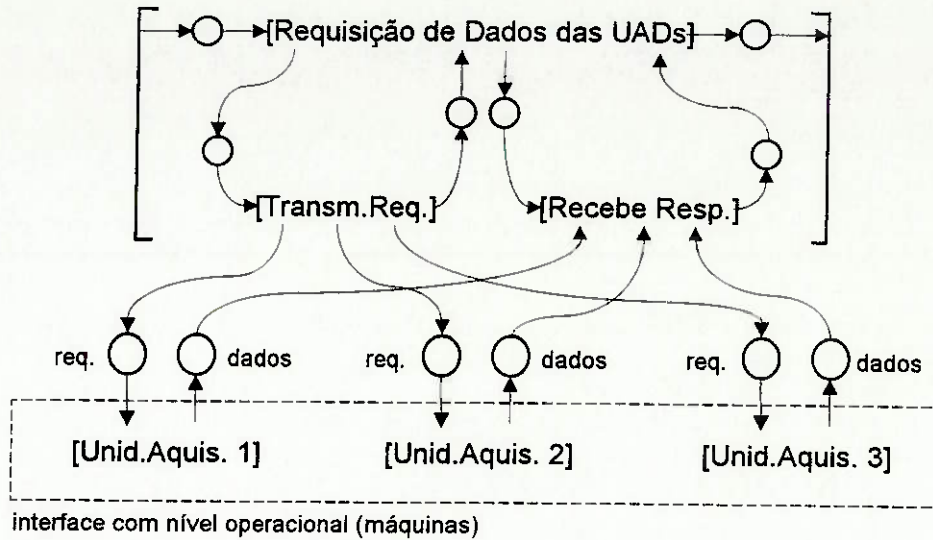
Observa-se a utilização de blocos bem formados na construção do modelo. A interface com a comunicação apresenta uma estrutura do tipo "loop", como foi ilustrado na Fig. 3-5(d), enquanto que o processamento utiliza um refinamento do tipo seqüência representado na Fig. 3-5(a). A interpretação disso é que, enquanto eventualmente processa-se informações para o usuário, o sistema está constantemente realizando a comunicação com o nível operacional e atualizando a base de dados.

3.4.5 Modelo do sistema de comunicação

O sistema de comunicações é a atividade responsável pelo gerenciamento e controle do transporte de dados entre as atividades. Existem várias formas de se realizar a comunicação de dados e nesse exemplo será adotado uma técnica bastante simples, que é a utilização de uma filosofia mestre-escravo. Nesse tipo de filosofia, um mestre envia requisições para cada uma das estações escravo e apenas a que recebeu a requisição pode responder, se possuir dados para serem transmitidos. Essa técnica, onde o escravo só responde quando é chamado, denomina-se "polling". O mestre inicia uma atividade de [Requisição de Dados da UADs]. Essa atividade realiza duas chamadas de procedimentos externos, um procedimento de transmissão de requisição e outro de recepção de resposta, conforme representado na Fig. 3-11(a). As UADs (Unidades de Aquisição de Dado) são os dispositivos responsáveis pelo interfaceamento das máquinas (nível operacional) com o sistema de comunicações e representam as estações escravo.

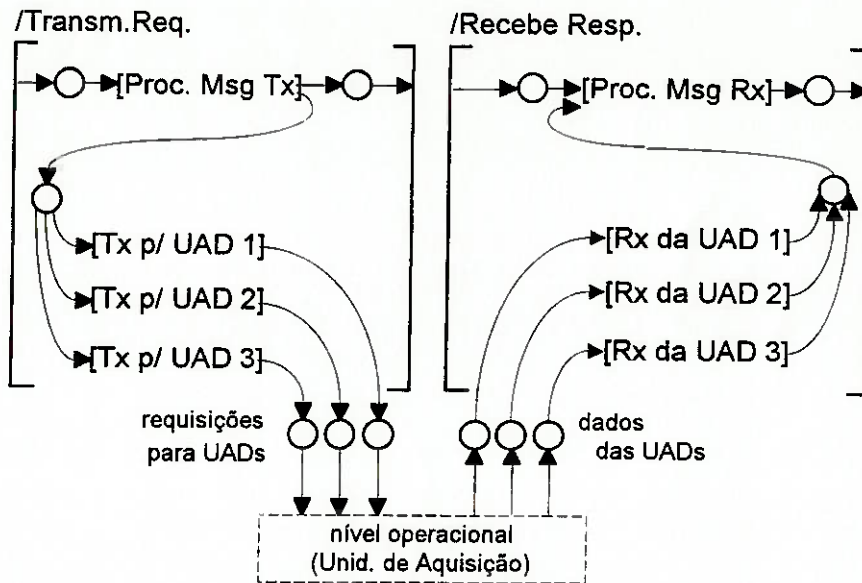
A atividade [Transm. Req.] é responsável pelo endereçamento das requisições às respectivas UADs. Pela Fig. 3-11(b), pode-se observar que essa atividade possui internamente uma estrutura de concorrência, onde apenas uma das atividades concorrentes (transmissão de dados para uma determinada UAD) é realizada na seqüência.

Além disso, na mesma figura, existem as atividades [ProcMsgTx] e [ProcMsgRx], que são responsáveis, respectivamente, pelo processamento da transmissão e recepção de mensagens. As atividades [Tx p/UAD] e [Rx da UAD] representam a transmissão e recepção de dados pelo meio físico para cada uma das estações.



UA 1 = Unidade de Aquisição de Dados da máquina 1

(a) - Modelo da atividade Sistema de Comunicações



(b) - Refinamento das atividades Transmissão de Requisição e Recepção de Resposta

Fig. 3-11 - Modelo do sistema de comunicações

3.4.6 Modelo da interface com as máquinas (Unidades de Aquisição de Dados)

As UADs representam a interface do sistema de comunicações com o nível operacional e consistem de dispositivos que realizam a aquisição de sinais das máquinas, processando-os e transformando-os em dados para serem enviados e processados, gerando as informações requisitadas pelo usuário. A Fig. 3-12 mostra sugestões de modelos para definir as UADs.

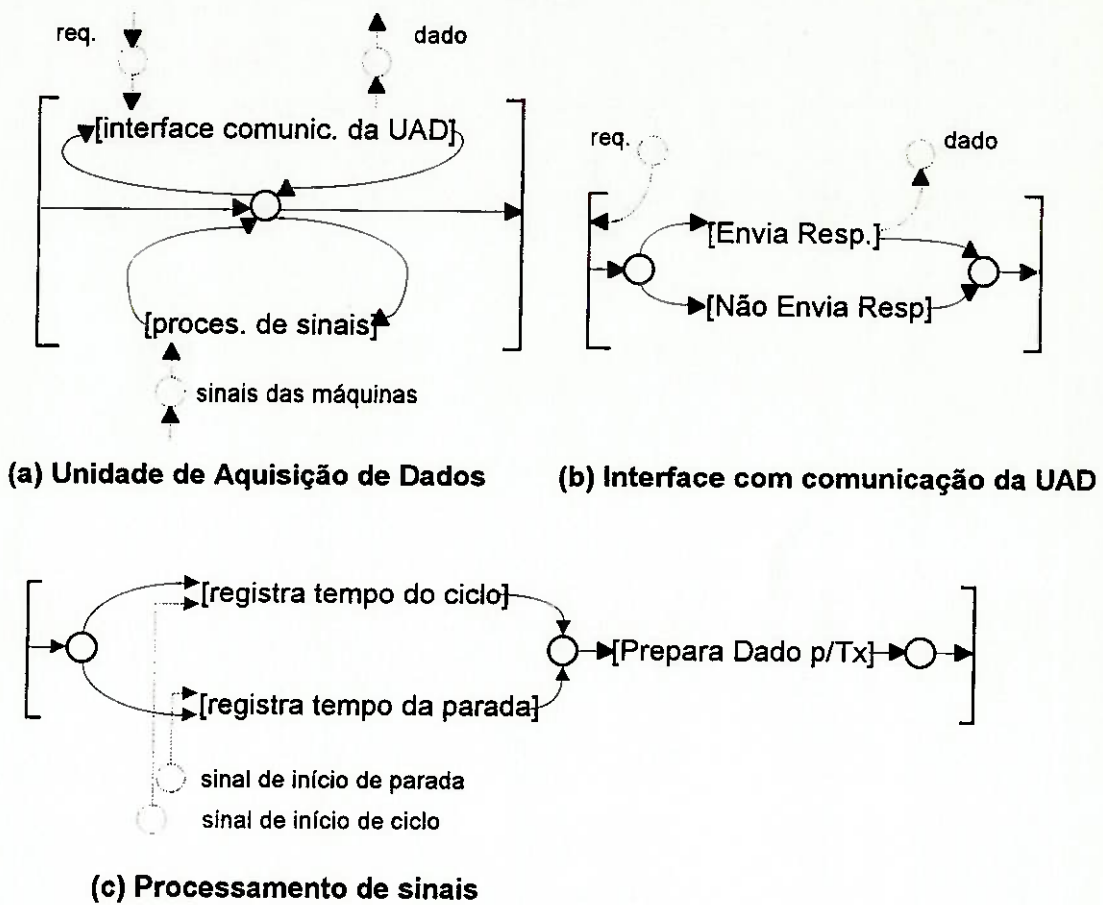


Fig. 3-12 - Modelos da UAD (Unidade de Aquisição de Dados)

O primeiro nível de detalhamento das UADs é apresentado na Fig. 3-12(a). Observa-se duas atividades com estrutura tipo "loop" compartilhando um mesmo elemento distribuidor. O elemento distribuidor define uma concorrência de atividades, em que apenas uma pode estar sendo realizada de cada vez. Apenas ao término de uma atividade é que se pode iniciar outra e, dessa forma, segundo o modelo, ou a UAD realiza o processamento dos sinais ou realiza a comunicação.

As figuras 3-12(b) e 3-12(c) detalham, respectivamente, a interface de comunicação e o processamento de sinais da UAD. A interface com a comunicação possui a opção de transmitir dados ou não, dependendo da existência de dados para serem transmitidos. O processamento de sinais recebe os sinais do processo, identifica-os e registra um tempo associado a cada um deles. Essas informações são formatadas em dados que são colocados à disposição para a transmissão.

A nível conceitual, as estruturas de concorrência representam um não-determinismo que definem um problema de conflito quando os modelos forem refinados a nível funcional. Tais conflitos deverão ser resolvidos por lógicas adicionais de controle. Esse próximo passo da metodologia será apresentado no capítulo seguinte.

CAPÍTULO 4

CONSTRUÇÃO DE MODELOS FUNCIONAIS

Após a especificação do fluxo de dados e informações e da divisão do problema em módulos e em níveis de abstração, é necessário realizar o controle desses fluxos, de forma a se conseguir gerar uma especificação do sistema para a sua implementação. Essa especificação do controle dos fluxos define o modelo funcional do sistema, a partir do qual se pode realizar a implementação através da derivação para algoritmos de controle ou para modelos de análise. A especificação do modelo funcional é definida na metodologia através da utilização do MFG, que é uma técnica derivada de redes de Petri.

4.1 Introdução

4.1.1 As redes de Petri

As redes de Petri [Pet77-, Agw79-, Rei85-, Mur89-, Val90-] consistem de um diagrama de grafos orientados, contendo lugares, transições e marcas, e permite representar o comportamento dinâmico de sistemas através de uma evolução das marcas, definidos por uma sequência de disparos de transições.

As transições só podem ser disparadas se todos os arcos de entrada estiverem conectados a lugares marcados, o que define suas pré-condições de disparo.

As características das redes de Petri tornam-a ferramenta adequada para especificar o comportamento do sistema, através da observação de seus estados intermediários. Além disso são muito comuns técnicas derivadas de redes de Petri que objetivem realizar análise de desempenho de sistemas através de simulações, utilizando para isso extensões temporizadas e estocásticas. Em especial, existem vários trabalhos de aplicações voltadas para a análise de desempenho de protocolos para redes de comunicações utilizando simuladores baseados em rede de Petri [Mar87-, Mat89-].

Devido a diversidade de aplicações das redes de Petri, surgiram diversas classes e extensões, como as redes C/E, redes coloridas e redes com predicados.

Em modelos de redes de Petri, é importante que o modelo construído apresente algumas propriedades desejáveis, que são: a limitação ("boundedness"), a vivacidade ("liveness") e a reversibilidade ("reversibility"). Para a análise dessas propriedades, existem técnicas bastante consolidadas, como por exemplo, análise por redução, construção de grafos de alcançabilidade e análise por invariantes. As propriedades e as técnicas de análise podem ser encontradas em [Pet81-, Mur89-, Val90-].

4.1.2 O MFG - Mark Flow Graph

O MFG foi desenvolvido, baseado no conceito das redes de Petri, objetivando realizar o controle de sistemas produtivos de eventos discretos [Myg89b]. Como as redes de Petri, o MFG também é um grafo bipartido, composto pelos seguintes elementos, conforme a Fig.4-1.

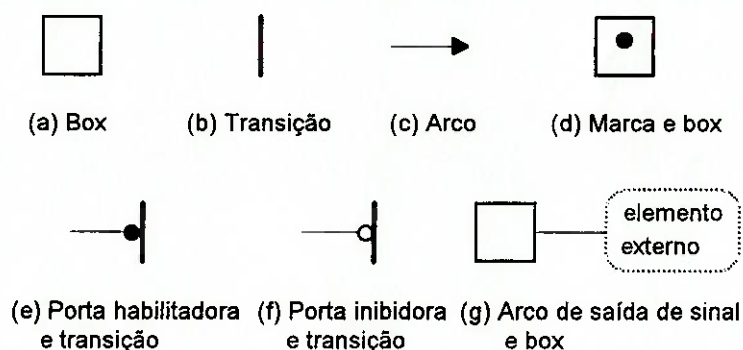


Fig. 4-1 - Elementos do MFG

O MFG pode ser considerado como pertencente a uma classe de redes de Petri do tipo condição-evento, que se caracterizam por ser um grafo seguro ("safe") [Myg89b], ou seja, permite a presença de apenas uma marca em cada box, de forma que devem ser definidas também pós-condições (não há marcas nos boxes de saída da transição) para o

disparo das transições. Além disso, no MFG, existem arcos inibidores e habilitadores, que controlam o fluxo das marcas, correspondendo a pré-condições adicionais para o disparo das transições. No apêndice B desse trabalho encontram-se exemplos que ilustram a dinâmica de disparos de transições do MFG. A dinâmica dos disparos das transições determina a evolução das marcas do grafo.

A metodologia desse trabalho é baseada na utilização de um modelo interpretado do MFG, ou seja, um modelo cujos elementos, como transições, boxes e marcas, são associadas a entidades do sistema real.

4.2 O modelo híbrido MFG/PFS interpretado

O PFS realiza o modelo do sistema em nível conceitual, especificando o fluxo de informações e dados do sistema e definindo os níveis de abstração do problema. A partir do modelo conceitual pode-se obter um modelo funcional, utilizando para isso um modelo interpretado de MFG, que na verdade é um modelo híbrido MFG/PFS [Myg88a]. O modelo funcional é responsável pela especificação do controle do fluxo de dados em cada nível de refinamento do PFS.

4.2.1 Representação das atividades e dos elementos distribuidores

No modelo interpretado do MFG, os elementos ativos são representados basicamente por duas transições e um box interno, que representa a condição de execução da atividade. Dessa forma, os colchetes dos elementos ativos do PFS, "[" e "]", são interpretados, respectivamente, como uma transição de entrada, representando o início da atividade, e uma transição de saída, representando o término da atividade.

O MFG interpretado pode representar 4 tipos básicos de atividades [Has87-, Myg88a], relacionadas a seguir e ilustradas na Fig.4-2(a).

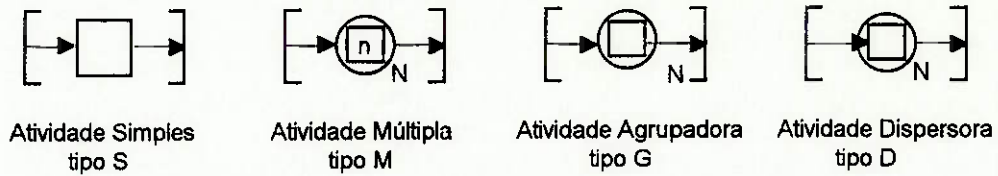
- (1) elemento ativo simples (tipo S): a atividade contém um box simples, com capacidade para no máximo uma marca;
- (2) elemento ativo múltiplo (tipo M): a atividade contém um box capacidade de valor N, que modela uma conexão em série de N boxes simples. Pode representar, por exemplo, uma fila de requisições de acesso a um banco de dados.
- (3) elemento ativo agrupador (tipo G): contém um box capacidade de valor N, que aguarda a acumulação de N marcas para enviar uma na saída. Pode representar, por exemplo, um "buffer" de comunicação que aguarda N mensagens para serem enviadas em um pacote só.
- 4) elemento ativo dispersor (tipo D): contém também um box capacidade de valor N, recebendo uma marca na entrada e enviando N marcas na saída. Esse box pode representar, por exemplo, a divisão de um arquivo grande para ser transmitido em N pacotes de comunicação.

Tais atividade correspondem a macro-elementos, que auxiliam a construção de modelos em MFG na medida em que "abreviam" estruturas que freqüentemente se repetem nos modelos. Alguns chamam o modelo que utiliza tais macro-elementos de MFG-Funcional (F-MFG) [San93-].

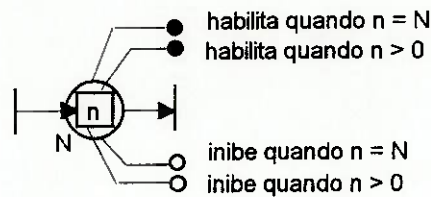
Observa-se que os boxes internos das atividades representam condições que podem ser utilizadas para derivar portas de sinais lógicos e controlar os fluxos de dados do sistema, conforme ilustrado na Fig.4-2(b). Assim, um box simples pode fornecer um sinal de controle lógico "1" ou "0", dependendo da existência ou não de marcas em seu interior, ou seja, dependendo da condição interna de execução de atividade.

O box capacidade, por sua vez, pode fornecer sinais de controle derivado do círculo, que assume valor "1" quando $n = N$, ou derivado do quadrado, que assume valor "1" quando $n > 0$ (onde $n =$ número de marcas no box).

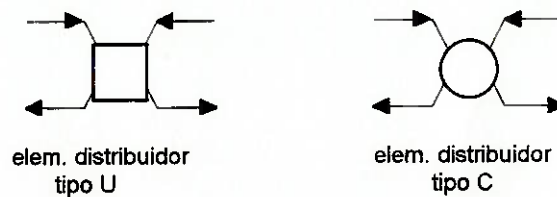
Os elementos distribuidores também podem ser representados por box simples, com capacidade 1 (tipo U), ou por box capacidade (tipo C), conforme ilustra a Fig. 4-2(c).



(a) Representação em MFG dos elementos ativos



(b) Interpretação dos sinais de controle



(c) Representação em MFG dos elementos distribuidores

Fig. 4-2 - Representação em MFG dos elementos PFS

4.2.2 A representação de recursos

O início de atividades geralmente está vinculado à disponibilidade de recursos para a sua execução. Tais recursos podem ser, por exemplo, unidades de processamento, banco de dados ou o meio físico para a transmissão de dados.

A condição de disponibilidade de recursos necessários para o início da atividade é representado por um box ligando a transição fim até a transição início da atividade. O recurso é modelado através de uma ou mais marcas nesse box, como se observa na Fig. 4-3(a).

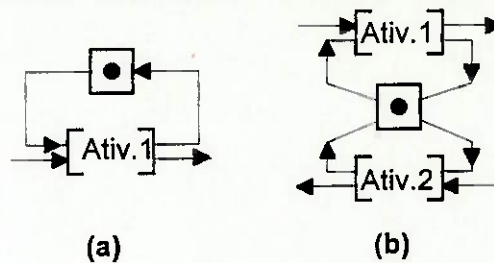


Fig.4-3 - Representação dos recursos

Observa-se que, se uma outra atividade necessitar da mesma condição para iniciar, ou seja, ela também necessita do mesmo recurso, pode-se definir uma estrutura de compartilhamento de recursos, conforme ilustra a Fig. 4-3(b). Essa estrutura pode ser utilizada para representar, por exemplo, uma comunicação do tipo "half-duplex", onde o meio físico deve ser compartilhado entre a transmissão e a recepção.

A estrutura de compartilhamento de recursos é também bastante útil para modelar mecanismos de sincronização entre processos concorrentes, também chamado de semáforos. Tais mecanismos são comuns em desenvolvimento de controle de sistemas distribuídos através de primitivas P e V operando nos semáforos [Agw79-, Pet81-].

4.2.3 A metodologia e as propriedades do modelo

A utilização da metodologia deve permitir a construção de modelos que apresentem algumas propriedades desejáveis, como a limitação, a vivacidade e a reversibilidade. A observação e análise dessas propriedades permitem validar o modelo sob o ponto de vista funcional.

i) limitação

O MFG é definido como uma rede segura, ou seja, para cada box permite-se apenas a ocupação de apenas uma marca. Na verdade, com a utilização dos macro-elementos, cada box pode conter no máximo um número finito N de elementos, o que garante a propriedade de quantidade de marcas limitadas na rede. Essa característica é importante do ponto de vista de desenvolvimento, pois em sistemas reais não existem capacidades infinitas, de forma que N pode ser utilizado como um dos parâmetros para dimensionamento do sistema.

ii) vivacidade (rede sem "dead-locks")

Na metodologia, a construção de redes "dead-free" implica inicialmente que todas as rotas que a marca percorre internamente dentro da atividade tenham sempre como partida a transição de início e terminem na transição de fim. Tais rotas são denominados caminhos de atividade (Fig.4-4).

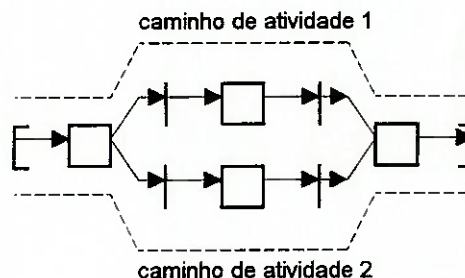


Fig.4-4 - Caminhos de atividade

Dessa forma, estruturas como a invasão e o vazamento (Fig.4-5) são proibidas. Para tanto, recomenda-se a utilização dos blocos bem formados ou combinações destes, abordados no capítulo 3. A construção de modelos utilizando blocos bem formados pode ser comparada à produção de software utilizando programação estruturada.

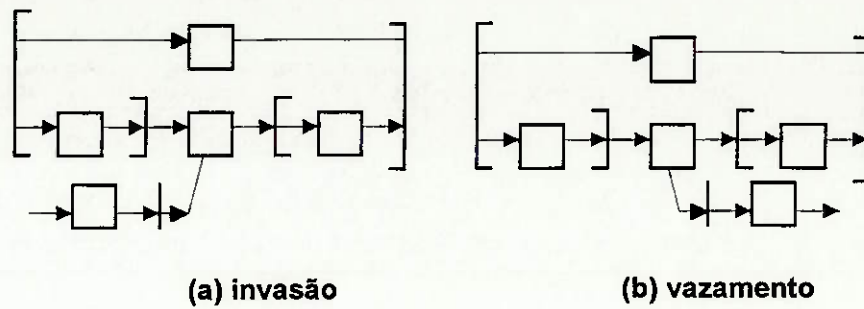


Fig. 4-5 - Estruturas proibidas

As relações entre atividades através de fluxos secundários devem ser bem observadas, para não gerar situações como a da Fig.4-6(a). Além disso, na especificação do controle através da utilização de gates habilitadores e inibidores, deve-se também evitar situações como a da Fig.4-6(b).

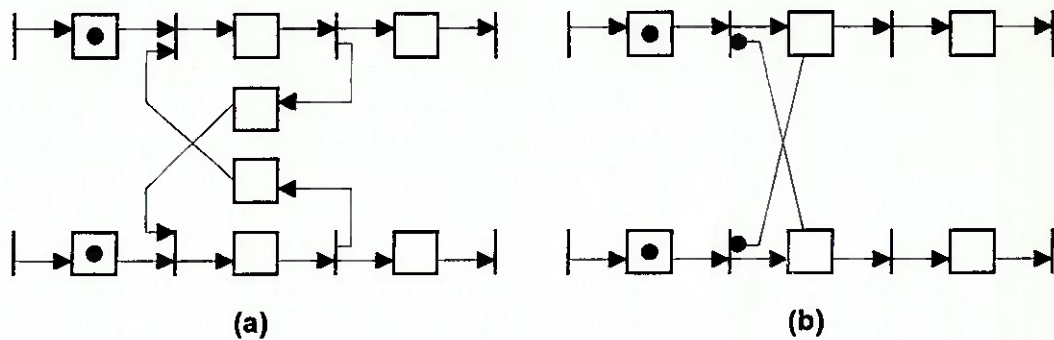


Fig. 4-6 - Estruturas com "dead-lock"

iii) reversibilidade

A reversibilidade é uma propriedade que permite o retorno ao estado inicial sempre que um processo é iniciado. Na metodologia, tal propriedade é garantida, se o sistema for "dead-free", através da condição de disponibilidade de recursos, que implicitamente representa um fluxo da marca da transição fim para a transição início.

4.3 A especificação do controle do fluxo de dados

A partir do modelo em PFS, obtém-se um modelo MFG/PFS interpretado utilizando as considerações acima, com respeito à representação dos elementos ativos, passivos e recursos. O modelo híbrido permite a monitoração do fluxo de dados através da observação da evolução das marcas.

O próximo passo da metodologia é representar as especificações do controle do fluxo de dados. A especificação do controle deve garantir, sobretudo, que o comportamento do fluxo de dados esteja de acordo com as especificações desejadas para o sistema. Além disso, pode-se garantir a vivacidade da rede através da imposição de novas condições de controle na rede para impedir situações de "dead-lock".

4.3.1 Detecção e controle de "dead-lock"

Através de análises ou simulações, pode-se detectar situações de "dead-lock"* no sistema. Nesses casos, para a obtenção de um sistema vivo, é necessário que seja colocado uma lógica adicional de controle. Seja o exemplo da Fig.4-7(a).

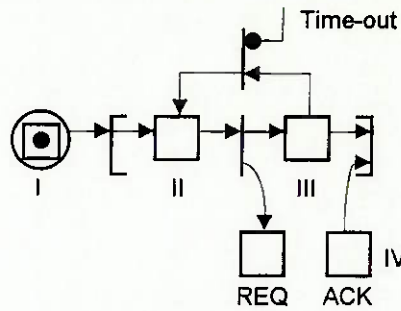
O sistema representa uma parte de um protocolo de comunicação simples. Nesse protocolo, a mensagem é transmitida e aguarda a chegada de uma resposta (reconhecimento ou ACK). Se a resposta não chegar após um certo tempo ("time-out"), ela é retransmitida até se obter a resposta.

O box I é um box capacidade e representa uma fila de espera de dados aguardando a transmissão. O box II representa uma condição de início de transmissão, o box III representa a espera da resposta e o box IV representa a chegada da resposta. Uma simulação do modelo permite que se chegue à condição da Fig.4-7(b).

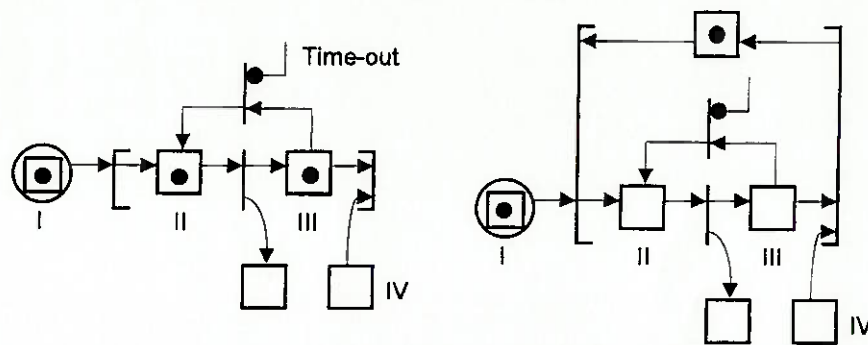
Se houver a perda de muitas mensagens para o meio físico, ou então se a estação receptora estiver quebrada, não haverá chegada de marcas no box IV. Nesse caso, o

* O "dead-lock" pode ser de dois tipos: o estrutural e o condicional, que serão definidos no Capítulo 6. Ambas as situações podem ser detectadas pela simulação.

sistema entra em estado de "dead-lock". Uma das formas de se evitar esse problema é controlar a entrada de marcas no início da atividade utilizando uma condição de "recurso" disponível. No caso, o recurso é a própria atividade de transmissão de mensagens. Assim, a atividade só permite a entrada de apenas uma marca por vez, evitando-se a situação de "dead-lock", conforme ilustra a Fig.4-7(c).



(a) modelo de transmissão de dados com retransmissão



(b) "dead-lock"

(c) solução do "dead-lock"

Fig.4-7 - Controle para evitar "dead-lock"

4.3.2 O problema do conflito

O modelo conceitual divide o problema em diversos níveis de abstração de forma a facilitar a sua análise. Em cada nível, freqüentemente a solução do problema pode ser vinculada à solução do conflito, de forma que o seu controle é um dos aspectos mais importantes a serem abordados dentro da metodologia. O conflito, no MFG, caracteriza-se quando existem dois (ou mais) arcos de saída ou dois (ou mais) arcos de entrada

ramificando ou convergindo para um box. Portanto, estruturas de concorrência, estruturas repetitivas e estruturas de compartilhamento de recursos são estruturas que apresentam conflitos.

A solução do conflito geralmente está vinculada ao controle do início ou término de atividades, utilizando, para isso, sinais de habilitação ou inibição, ou ainda arcos, que, através de uma lógica de controle, arbitram a transição que irá disparar.

A Fig.4-8 mostra a representação genérica de uma lógica de arbitração. Geralmente essa lógica é realizada por sub-grafos MFG ou por sinais provenientes de elementos externos. A definição da lógica de arbitração define a solução do conflito e também a solução do problema.

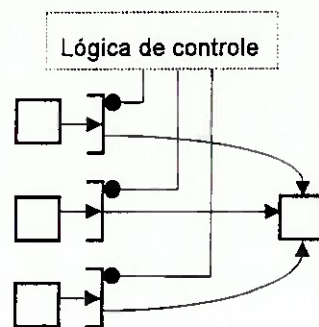


Fig. 4-8 - Representação geral da lógica de arbitração

4.3.3 Outros aspectos de controle

i) controle de início e fim de atividade

Arcos e portas de controle podem determinar o início e fim de atividades. Seja o exemplo da Fig.4-9, que representa a atividade de atualização de dados em um sistema de banco de dados distribuído. Nesse tipo de sistema, cada estação local possui uma cópia dos dados das demais estações remotas. Quando uma estação local requisita a atualização de sua base de dados, deve ser verificado inicialmente se alguma outra estação não iniciou um processo de atualização. Essa condição pode ser fornecida por um controle central e é representada pelo arco inibidor derivado do box I.

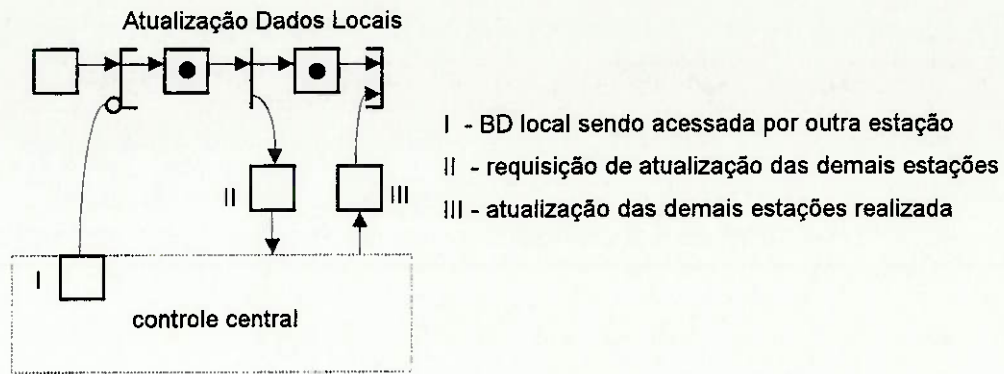


Fig.4-9 - Controle de início e fim de atividade

Uma vez iniciada a atividade, além da base local ser atualizada, o processo requisita ao controle central a atualização das demais estações (box II) e apenas quando todas tiverem sido atualizadas é que a atividade pode encerrar. Tal condição é representada pelo arco proveniente do box III.

ii) limitação do número de marcas nas atividade

Seja o caso de uma comunicação em que existe um meio físico contendo N canais multiplexados, ou seja, é possível transmitir simultaneamente N mensagens na linha. A Fig.4-10 mostra essa situação. A capacidade da linha é representada por um box capacidade de valor N. Se a linha já estiver cheia, é possível colocar uma condição para que a transição de início não seja disparada, associando a ela um sinal inibidor derivado do box capacidade, limitando em N o número de marcas na atividade.

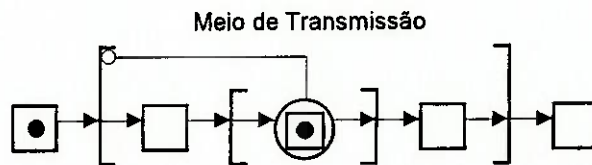


Fig.4-10 - Limitação do número de marcas na atividade

4.4 Extensões do modelo MFG

Há ocasiões em que os elementos definidos pelo MFG, além dos macro-elementos do MFG funcional, não são suficientes para representar adequadamente os modelos. Nesses casos, a metodologia permite a utilização de ferramentas estendidas derivadas do MFG para a construção dos modelos funcionais. As extensões pretendem aumentar o poder de expressão do MFG de duas formas:

- (1) aumentar a capacidade da ferramenta para a representação de dados. Nesse caso, existe uma tendência para a utilização de marcas individuais [Rei92-], ou seja, as marcas podem ser diferenciadas por atributos que as identificam uma das outras;
- (2) permitir a representação do conceito de tempo. A introdução de tempo nos modelos é bastante comum, principalmente quando há a necessidade de se avaliar o desempenho de um sistema e analisar seus parâmetros quantitativos [Val90-].

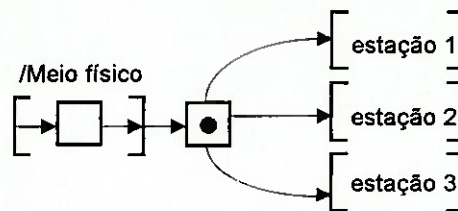
Observa-se que o PFS não impõe restrições para a utilização das extensões do MFG. Os modelos em PFS objetivam especificar fluxos e interrelações, enquanto a proposta do MFG é realizar o controle desses fluxos e representar o comportamento dinâmico do sistema. Assim, a utilização de estruturas definidas nas ferramentas estendidas tem por objetivo facilitar a modelagem do controle do sistema. Tais extensões serão abordadas a seguir.

4.4.1 Modelo estendido com marcas individuais

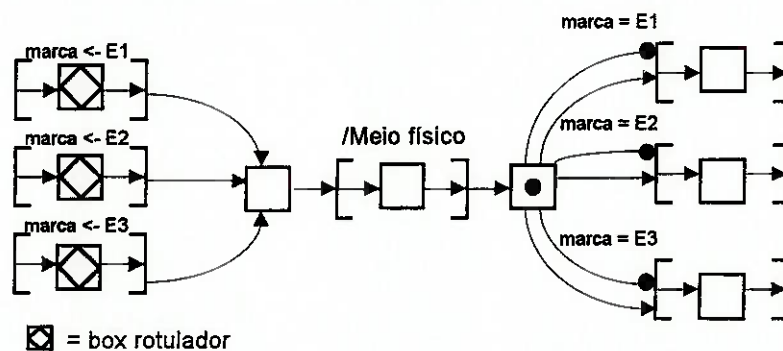
Em certos níveis de abstração, a especificação dos fluxos de dados pode aumentar demasiadamente a quantidade de boxes e transições, dificultando a compreensão e a análise do modelo. Além disso, em sistemas reais, algumas situações de conflito são resolvidas por informações embutidas nos dados, como é o caso do problema de endereçamento de uma mensagem a uma determinada estação, conforme a Fig.4-11(a).

Pela figura, a mensagem, representada pela marca, possui 3 destinos possíveis. No sistema real, o conflito é resolvido pelo endereço contido na mensagem. Portanto, para representar o modelo adequadamente, é interessante que a própria marca possua atributos que as identifiquem uma das outras, para que ela mesma possa arbitrar o conflito. Esse tipo de situação motiva a utilização de uma extensão com maior capacidade de modelagem de dados. Como exemplo de um modelo MFG com marcas individuais, será adotado nesse trabalho uma ferramenta denominada MFG estendido ou E-MFG [San93-].

A Fig.4-11(b) apresenta a solução do problema de endereçamento utilizando a representação em E-MFG. Observa-se a existência de um box rotulador, que define os atributos da marca. Assim, quando a marca entra no meio físico, ela já possui a informação da sua estação destino. A arbitração é representada pelas portas habilitadoras que possuem inscrições em seus arcos que selecionam o caminho das marcas.



(a) Problema do endereçamento



(b) arbitração pelos atributos das marcas

Fig. 4-11 - Exemplo de utilização do E-MFG

Além desses elementos, o E-MFG permite a definição de regras nas transições para realizar operações nas marcas. Define também inscrições em arcos diretos de entrada de transições para selecionar os campos dos atributos que deverão acompanhar a marca após o disparo. O E-MFG está apresentado no apêndice E.

4.4.2 Modelos estendido para simulação

Uma das vantagens da utilização de técnicas baseadas em redes de Petri ou MFG é a possibilidade de fazer análises no modelo através de uma teoria já consolidada, que inclui métodos analíticos como análise por invariantes e construção de grafos de alcançabilidade [Pet81-, Val90-]. Além disso, a capacidade de representar o comportamento dinâmico do sistema torna o MFG uma ótima ferramenta para realizar simulações [Hsg87-]. Aspectos relativos a análise serão abordados no Cap.6.

A simulação permite a obtenção de certos parâmetros quantitativos para o projeto do sistema através da observação do comportamento dinâmico do modelo. Para que se obtenham resultados quantitativos através da simulação, é importante que o modelo possa representar a noção de tempo no sistema. O tempo, no modelo do MFG para simulação, é representado basicamente por dois elementos: o box temporizado e a transição temporizada, conforme ilustrado na Fig.4-12.



Fig.4-12 - Elementos temporizados do MFG

O box temporizado representa a duração de uma certa condição. O box, nesse caso, retém a marca durante um certo intervalo de tempo especificado. Um box

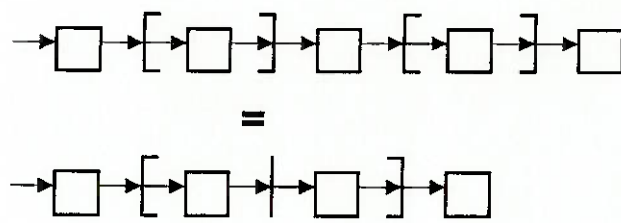
temporizado de tempo τ pode representar, por exemplo, que uma determinada máquina demora um tempo τ para realizar uma determinada operação.

A transição temporizada especifica um atraso de um tempo τ para o seu disparo no momento em que as pré e pós-condições de disparo são estabelecidas. Se durante esse intervalo de tempo τ as condições de disparo desaparecerem, a transição temporizada zera a contagem do tempo e aguarda o reestabelecimento das condições de disparo. Esse tipo de comportamento é bastante adequado para modelar processos com "time-out".

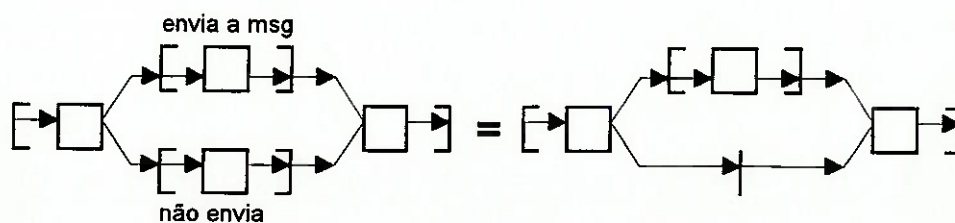
4.5 Simplificação do modelo

O modelo interpretado admite algumas simplificações para facilitar a sua compreensão e análise.

A primeira consiste no caso em que o elemento distribuidor encontra-se entre duas atividades, onde o término da anterior coincide com o início da posterior. Nesse caso, tais atividades são conectadas diretamente através de uma transição, conforme ilustrado na Fig.4-13(a).



(a)



(b)

Fig.4-13 - Simplificações do modelo

A segunda simplificação é com respeito a atividades que não influenciam a interpretação do modelo, de forma que podem ser reduzidas a uma transição. Seja o exemplo da Fig.4-13(b), onde existe uma concorrência entre a atividade [envia msg] e [não envia msg]. Na verdade, a atividade [não envia msg] representa apenas um caminho alternativo para as marcas caso as condições de disparo de [envia msg] não estejam satisfeitas. Esse tipo de construção pode ser comparada a estruturas do tipo IF-THEN sem ELSE encontradas em programação estruturada.

4.6 Exemplos

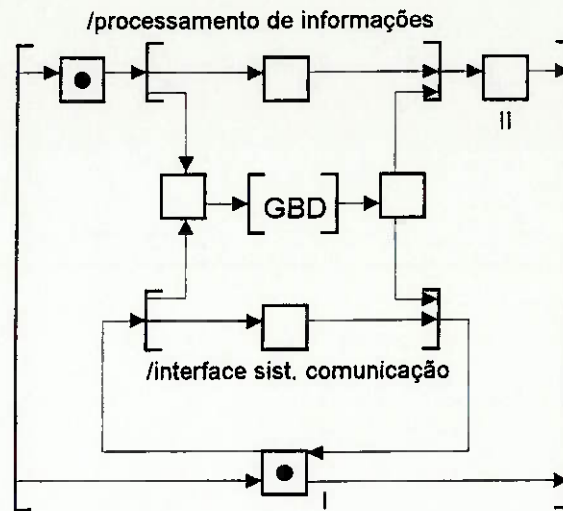
Para ilustrar os conceitos abordados nesse capítulo, serão apresentados os respectivos modelos MFG dos exemplos apresentados no final do capítulo 3. Cada um dos módulos do SI também serão abordados separadamente.

4.6.1 Módulo de interface com o usuário

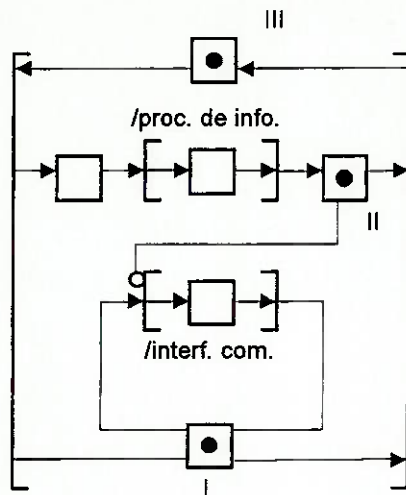
A Fig.3-10 apresenta a especificação dos fluxos em PFS. A Fig.4-14(a) apresenta uma sugestão do mesmo modelo em MFG/PFS, contendo a marcação inicial. A esse nível, observa-se alguns conflitos que definem dois problemas principais, que são o controle de término da atividade global e o controle de acesso à base de dados.

i) controle do término da atividade global

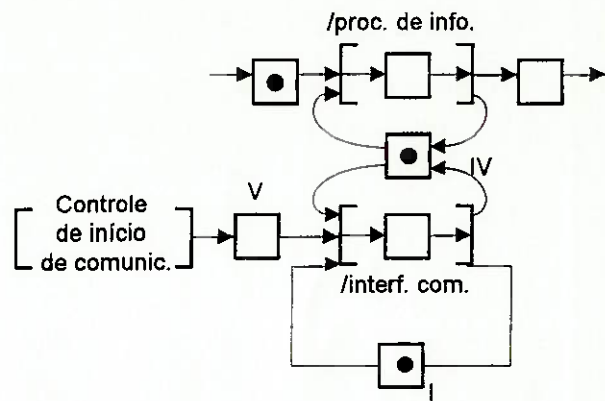
Na Fig.4-14(b), existe um conflito no box I, onde a marca pode entrar na atividade [interface comunicação] ou então finalizar atividade global (condição possível quando os boxes I e II estão marcados). Uma forma de resolver esse conflito é colocar um arco inibidor na transição de entrada da [interface comunicação] conectado ao box II. A transição de saída global é disparada e o modelo volta à marcação inicial, que é garantida pelo box III. (o GBD - Gerenciador de Base de Dados foi suprimido para facilitar a representação).



(a) Modelo MFG da interface com usuário



(b) Controle do fim da atividade



(c) Controle do acesso ao GBD

Fig.4-14 - Aspectos de controle do fluxo de dados da atividade de [interface com usuário]

ii) controle do acesso ao GBD (Gerenciador de Base de Dados)

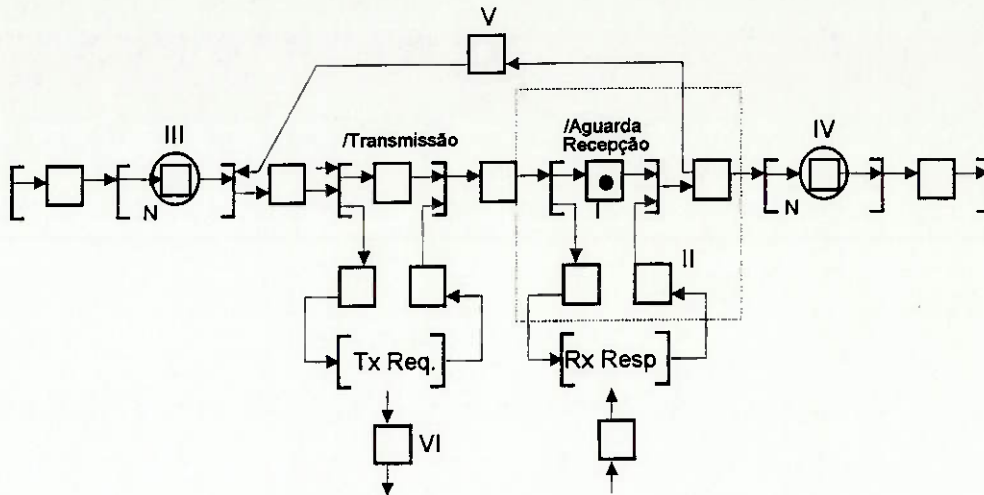
O segundo problema refere-se à decisão para o acesso da atividade GBD, que é um procedimento externo compartilhado entre dois processos. O problema do gerenciamento de acesso à GBD consiste em controlar o início e término das atividades [processamento de informações] e [interface com o sistema de comunicações].

Uma possível solução é apresentada na Fig.4-14(c), onde o box IV representa uma variável do tipo semáforo, que sincroniza os dois processos permitindo a entrada de apenas uma marca por vez em GBD. Assim, essa estrutura define o acesso à GBD por ordem de chegada de marcas. Além disso, pode-se definir uma estrutura para controlar o início da atividade de interface com o sistema de comunicações. Essa estrutura, representada na figura pela atividade [controle de início de comunicação], deve gerar uma marca no box V, indicando uma condição para que a comunicação possa ser iniciada. Se for associado um tempo a esse módulo, pode-se definir os intervalos de tempo entre aquisição dos dados das máquinas do nível operacional através do sistema de comunicações.

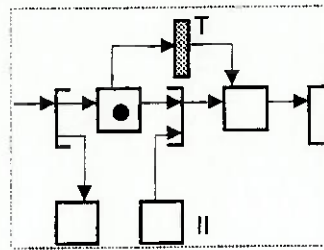
4.6.2 Controle do sistema de comunicações

A Fig.4-15(a) representa uma especificação em MFG/PFS do modelo da Fig.3-11(a). Observa-se a utilização do box agrupador (IV) e do box dispensor (III), ambos com capacidade N. Uma marca que entra em III gera N marcas, onde cada marca representa um requisição de dados para uma máquina. O box IV recolhe as N marcas e as agrupa, permitindo o retorno do sistema ao estado inicial. O box V garante que apenas uma mensagem seja enviada por vez.

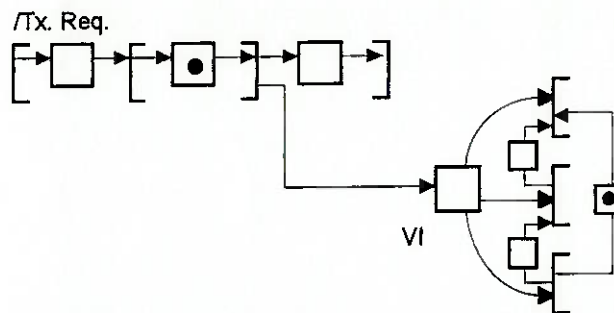
Uma simulação nesse modelo permite verificar que existe uma possibilidade de que não se chegue nenhuma marca no box II, o que pode ocasionar um "dead-lock" no sistema. Para evitar essa situação, uma lógica adicional de controle deve ser inserida no modelo, conforme ilustrado na Fig.4-15(b).



(a) Modelo MFG do sistema de comunicações



(b) Controle de chegada de resposta



(c) Controle de conflito por ordem fixada

Fig. 4-15 - Controle do fluxo de dados da atividade [Sistema de comunicações]

Essa lógica é baseada em uma transição temporizada T, que retém a marca no box I e aguarda a chegada da resposta no box II. Se após um certo intervalo de tempo a marca não chegar em II, a transição T é disparada, permitindo a continuidade da evolução da marca, eliminando o "dead-lock".

O segundo problema relaciona-se ao problema do endereçamento da UAD, representado por um conflito no box VI. Observa-se que o sistema de comunicações foi especificado sob uma configuração "mestre-escravo", onde um escravo é chamado por vez. Portanto, pode-se utilizar uma estrutura de ordem fixa para resolver o conflito, conforme ilustrado na Fig.4-15(c).

4.6.3 Controle da interface com as máquinas

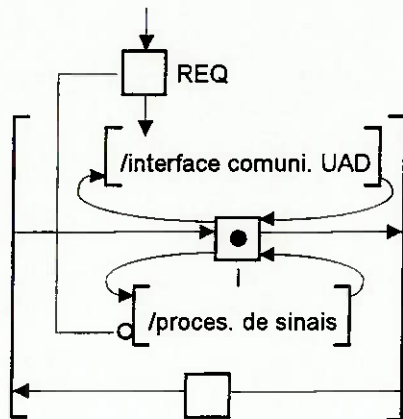
A Fig.4-16(a) mostra uma possível especificação de controle do modelo da Fig.3-12(a). O conflito no box I define uma concorrência entre as duas atividades internas e o fim da atividade global. A chegada de uma marca no box REQ impõe uma prioridade de execução para a atividade [interface de comunicação da UAD]. O controle de fim da atividade global pode ser realizado por um sinal externo (por ex., um sinal de "desliga" da UAD), que não será representado para simplificar o modelo.

A Fig.4-16(b) mostra a solução do conflito apresentado no modelo da Fig.3-12(b). O box II e III representam, respectivamente, a existência de dados para a transmissão. Tais boxes podem ser implementados como variáveis de controle, representando dois estados possíveis do sistema, arbitrando o conflito e definindo as condições para disparo de uma das transições através dos arcos habilitadores. A mudança de estado do sistema é representado pela passagem da marca de um box para outro, também definido por sinais de controle. As condições para a mudança de estado são:

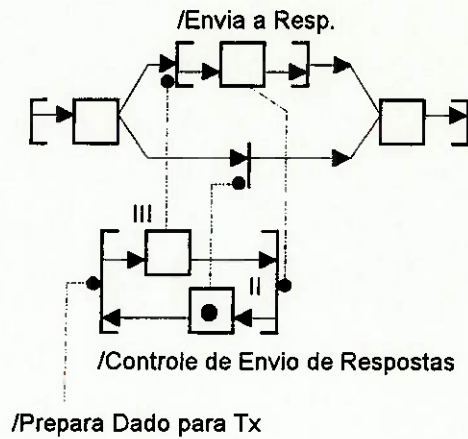
- (1) a ocorrência da atividade [Prepara Dado para Tx], mostrado na Fig.4-16(c) e, nesse caso o box II é marcado, indicando a existência de dados para serem transmitidos;

(2) a ocorrência da atividade [Envia a Resp], que retira a marca do box II e a coloca em III, indicando que, localmente, não há mais dados para serem transmitidos.

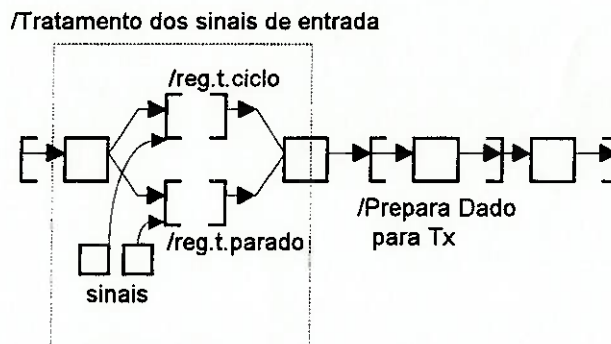
Dessa forma, pode-se especificar o comportamento dinâmico do dispositivo, facilitando a sua implementação.



(a) Unidade de aquisição



(b) interface de comunicação da UAD



(c) Processamento de sinais

Fig.4-16 - Controle do fluxo de dados da atividade de [interface com as máquinas]

CAPÍTULO 5

ASPECTOS DE IMPLEMENTAÇÃO

O desenvolvimento de SIs para ambiente industrial envolve atualmente o conhecimento de uma série de tecnologias para a sua implementação. Em princípio, a metodologia poderia ser aplicada independente da tecnologia utilizada [Ger87-], entretanto é interessante abordar como a metodologia pode auxiliar a implementação das componentes dos SIs tendo em vista os aspectos tecnológicos.

Conforme visto no capítulo 2, a implementação de SIs compõe-se de três partes principais, que são especificações e (ou) desenvolvimento de redes de comunicações, dos softwares aplicativos do sistema e dos dispositivos que compõem o hardware do sistema. Os aspectos tecnológicos mais importantes relativos a estes pontos e como a metodologia MFG/PFS pode auxiliar em seu desenvolvimento são abordados a seguir.

5.1 A metodologia e o desenvolvimento do software aplicativo

Com relação ao desenvolvimento de aplicações, pode-se identificar dois níveis do problema:

- (1) nível global, onde se define a arquitetura da base de dados em estruturas distribuídas ou centralizadas;
- (2) nível local, onde os aplicativos são especificados, gerando o fluxo de controle dos respectivos programas.

A nível global, devem ser definidas as políticas de acesso aos dados para a especificação do seu controle.

A nível local, devem ser considerados aspectos de padronização de interface com o usuário e com o sistema de comunicações. Também devem ser considerados aspectos de controle e de interação entre processos paralelos.

A metodologia fornece suporte para os aspectos acima na medida em que permite modelar as relações entre os processos e especificar as políticas de controle.

5.1.1 Nível global

O nível global considera o sistema de informações como um todo, onde pode existir uma variedade de subsistemas definidos para serem utilizados em diferentes aplicações. Tais aplicações podem estar localizadas geograficamente em locais distintos e definir a sua própria estrutura de dados. A esse nível, é importante definir uma política de organização dos dados do sistema.

Existe uma tendência atual para a utilização de uma política distribuída de processamento e alocação de dados. A utilização de sistemas distribuídos permite aspectos como compartilhamento de recursos, custos mais baixos, confiabilidade, possibilidade de crescimento gradual e utilização de equipamentos de fabricantes diferentes, o que os torna bastante atrativos, se comparados aos sistemas centralizados [Tan89-].

A Fig.5-1 mostra um modelo PFS de um sistema distribuído consistindo de dois subsistemas e um sistema de comunicações. Os subsistemas estabelecem uma relação de "cliente-servidor", onde o subsistema cliente envia requisições de dados ao servidor, através de uma série de estruturas de chamada de procedimento externo. O servidor fornece o serviço, acessando a sua base de dados através do GBDL (Gerenciador de Banco de Dados Local). A utilização do GBDL é compartilhada com os aplicativos locais, sendo necessária a definição de um controle de acesso à GBDL.

Em sistemas distribuídos, é comum que cada estação de trabalho ora seja cliente ora seja servidor. Nesse caso, também pode-se definir um controle para definir o modo que o sistema irá operar.

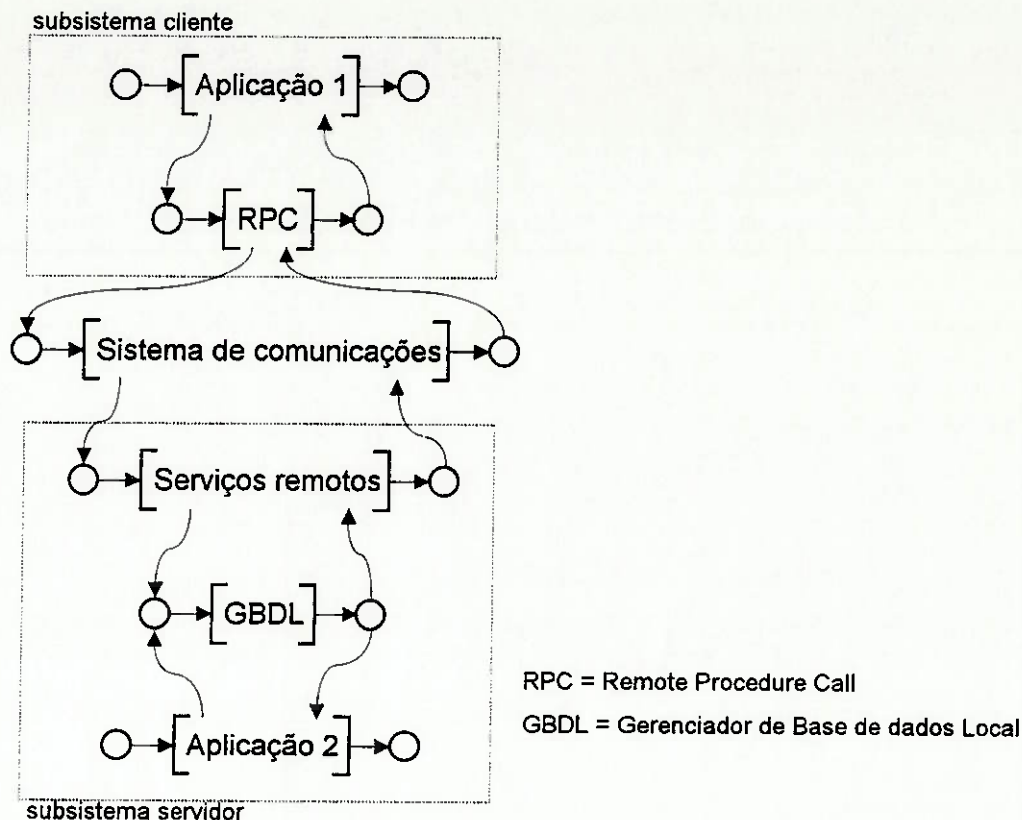


Fig.5-1 - Modelo PFS de um SI nível global

5.1.2 Nível local

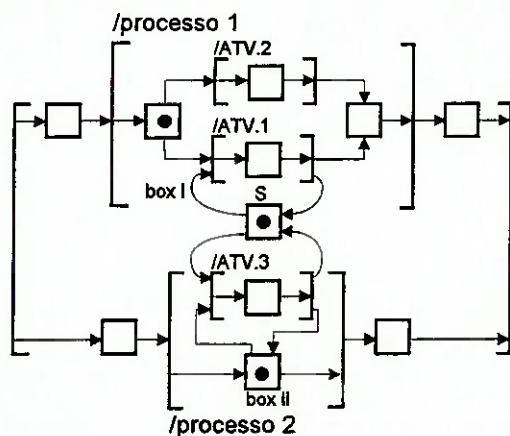
Em um sistema local, deve-se considerar a implementação de aplicativos com processos concorrentes e paralelos. Deve-se considerar também aspectos de padronização de interface homem-máquina e de interface com o sistema de comunicações. A metodologia auxilia a definição do fluxo de controle dos programas aplicativos, fornecendo uma técnica estruturada para definir seus módulos funcionais. Cada um desses aspectos será considerado separadamente.

i) definição do fluxo de controle de programas aplicativos.

A metodologia auxilia a implementação de módulos funcionais em programas aplicativos fornecendo uma ferramenta que realiza a especificação do sistema utilizando uma técnica estruturada.

Os blocos bem formados, por exemplo, podem ser comparados a estruturas comuns em programação estruturada. Uma concorrência entre duas atividades em MFG/PFS pode ser interpretado como uma estrutura do tipo IF-THEN-ELSE (no caso de mais de duas atividades, uma estrutura do tipo CASE) e a estrutura tipo "loop" pode estar representando um WHILE-DO. Além disso, a especificação de controle pode definir as variáveis lógicas para a decisão do IF-THEN-ELSE ou definir a condição de término do loop WHILE-DO.

A metodologia fornece uma boa ferramenta para especificar o controle de processos concorrentes ou paralelos. A Fig.5-2 ilustra uma situação em que dois processos paralelos são sincronizados através da definição de um semáforo S. A implementação desse modelo utiliza as primitivas P e V que operam em S e está apresentada ao lado da figura.



(a) Modelo MFG

processo 1 // processo 2

```

processo 1           processo 2
begin                begin
if <condição box I>  while <condição box II>
then begin           begin
  P(S);              P(S);
  ATV 1;             ATV 3;
  V(S);              V(S);
end                  end;
else ATV 2;

```

Obs: primitivas P e V são definidas como:
P(S): if S>0 then S:=S-1 else "aguarda"
V(S): S:=S+1

(b) Algoritmo equivalente do modelo

Fig.5-2 - Modelo de um processo em MFG

As técnicas de análise e simulação verificam se a relação entre os dois processos não provocam situações de "dead-lock".

ii) padronização de interface homem-máquina

Um aspecto importante na elaboração dos aplicativos é garantir que a forma de entrada e a apresentação de dados não varie muito de aplicação a aplicação. Os programas, dessa forma, tornam-se mais "amigáveis", tornando a sua utilização mais eficiente e amenizando eventuais resistências a sistemas informatizados dentro da fábrica.

As técnicas mais modernas de programação tendem a realizar projeto de software com reutilização [Luc89-]. Nesse tipo de abordagem, geralmente define-se uma biblioteca padrão de programas a ser utilizada no desenvolvimento das aplicações do usuário.

Atualmente tais bibliotecas são montadas através da definição de objetos que são instanciados nos programas aplicativos. Um objeto, a nível de implementação, contém atributos, representados por uma estrutura de dados próprias, e ações, que podem ser realizadas nesses atributos.

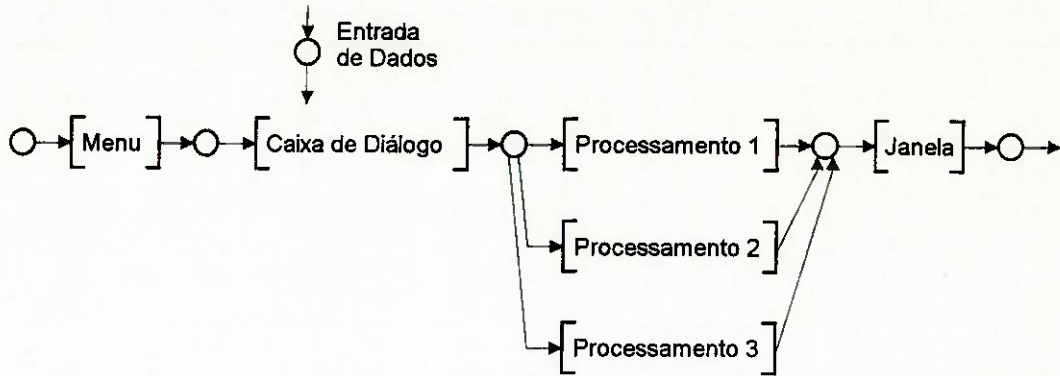
Assim, a definição de uma biblioteca contendo objetos tipo janelas e barras de menu padroniza as aplicações a nível de entrada e apresentação de dados, uniformizando a forma e os procedimentos de interação da máquina com o usuário.

O MFG/PFS permite que se possa observar as relações entre os objetos definidos na biblioteca através de fluxos. Seja o exemplo da Fig.5-3(a).

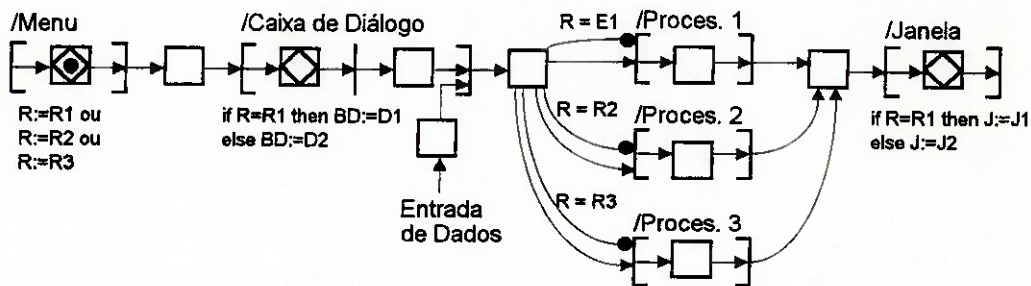
O menu, o box de diálogo e as janelas são objetos instanciados de uma biblioteca para padronizar a forma de apresentação do aplicativo para o usuário. Cada um desses objetos possui atributos e parâmetros que os definem (ex., cor, tamanho das janelas, itens do menu) e possuem procedimentos próprios (movimentação e redimensionamento de janelas, edição de dados de entrada), manipulando tais atributos através de parâmetros.

A utilização do MFG/PFS permite monitorar a execução do programa, verificando o comportamento dinâmico do aplicativo. Para isso cria-se um modelo funcional,

mostrado na Fig.5-3(b). Observa-se a utilização do E-MFG para a simplificação do modelo. Nesse modelo, a marca em um box define a instância de um objeto (ex., uma janela) e o atributo da marca identifica as características do objeto.



(a) Modelo PFS de um módulo de aplicação do usuário



marca possui atributos <R, BD, J, c>, onde:
 R corresponde o tipo de relatório desejado (R1, R2 ou R3)
 BD corresponde ao formato da caixa de diálogo (D1 ou D2)
 J corresponde ao formato da janela (J1 ou J2)
 c é um código de controle (ver anexo C)

(b) Modelo convertido em E-MFG

Fig.5.3 - Exemplo de um módulo de aplicação utilizando objetos pré-definidos

iii) padronização da interface com a rede de comunicações

Outra padronização importante é a da interface com o sistema de comunicações, que é realizada através da definição de serviços oferecidos pela rede. A padronização desses serviços permite que as aplicações possam ser desenvolvidas independentemente da implementação do sistema de comunicações.

As características próprias da máquina onde se localiza o aplicativo tornam-se transparentes à aplicação do usuário, sendo apenas consideradas durante a definição da biblioteca. Com isso, modificações na máquina implicam em apenas algumas modificações nos objetos definidos na biblioteca.

Uma forma de padronização bastante comum em sistemas de comunicações é a utilização de bibliotecas baseadas em serviços definidos pelo padrão MMS* (Manufacturing Message Specifications) [Bry88-]. O MMS define os serviços de comunicação entre dispositivos comuns em sistemas de manufatura, como robôs, máquinas CNC e controladores.

A utilização de bibliotecas definidas com base em serviços padrões de comunicação tornam o aplicativo portátil e executável independente da máquina em que será processado.

5.2 A metodologia e o projeto de redes de comunicações

A metodologia fornece um suporte para a especificação e o desenvolvimento de redes de comunicação, em especial na especificação do comportamento dinâmico de protocolos de comunicação. As redes desempenham um papel fundamental no desenvolvimento de sistemas distribuídos, sendo que muitos dos problemas de sistemas de informações a nível de implementação concentram-se na definição de uma rede adequada. Grande parte desse capítulo é, portanto, dedicado a esse assunto.

* O MMS é padronizado pela norma ISO/DIS 9505 - Manufacturing Message Specification. Part 1: Service Specification & Part 2: Protocol Specification, Draft 6, 1987.

5.2.1 O problema das redes de comunicações

Em sistemas de manufatura, o problema principal em redes de comunicações é escolher uma arquitetura de protocolos adequada para garantir a interconectibilidade entre equipamentos e estações de trabalho. Um protocolo de comunicação consiste de um conjunto de regras e convenções que devem ser estabelecidas para garantir a comunicação entre dois entes. Entretanto, dada a complexidade e a quantidade de aspectos que devem ser considerados para acordar a comunicação entre duas entidades, dividiu-se o problema em diversos níveis ou camadas, onde cada camada é definida por um ou mais protocolos. As camadas definem o que se chama uma arquitetura de protocolos.

A ISO (International Standards Organization) define uma arquitetura de sete camadas, denominado modelo de referência OSI (Open Systems Interconnection - norma ISO 7492). Embora exista uma série de sistemas de comunicação importantes que não obedecem à arquitetura de sete camadas OSI, esse modelo será usado para ilustrar o problema das redes de comunicações.

Segundo a terminologia ISO, cada camada pode definir um ou mais protocolos de comunicação que oferecem serviços às camadas superiores e utilizam serviços das camadas inferiores. As camadas se interagem entre si através de trocas de primitivas de serviços. Dessa forma, cada serviço é acessado através de um conjunto de primitivas próprias.

Observa-se, então, que tais características sugerem um modelo em que a funcionalidade e o comportamento de cada camada é representada por uma atividade, e a interrelação entre atividades é representada por fluxos secundários, que podem ser interpretados como um fluxo de primitivas entre camadas, conforme ilustrado na Fig.5-4. É importante definir quais problemas de comunicação são abordados em cada camada, para servir de referência ao projetista da rede.

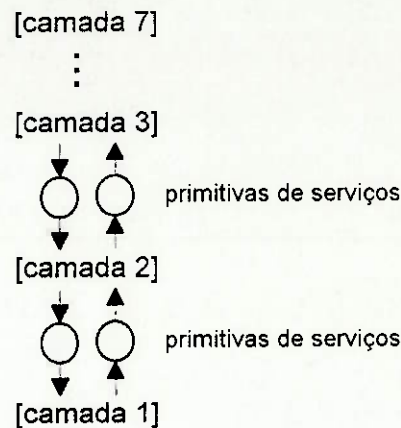


Fig. 5-4 - Modelo OSI-ISO de sete camadas

5.2.2 O modelo de referência OSI

As sete camadas especificadas no modelo OSI (norma ISO 7492) definem sete níveis do problema de comunicação. Cada nível pode ser considerado separadamente para o projeto de redes. Tem-se então os seguintes níveis:

- (1) camada física - nessa camada são tratados itens de projeto a nível físico, como a taxa de transmissão, o tipo de cabo a ser utilizado, número de pinos do conector da rede, forma de modulação do sinal etc.
- (2) nível enlace - pode ser dividido em dois níveis, o LLC (Logical Link Control) e MAC (Media Access Control)*. A sub-camada LLC gerencia a transmissão de dados entre dois nós adjacentes e define aspectos como o formato dos dados para comunicação, o gerenciamento de retransmissões, de perda ou duplicação de dados. A sub-camada MAC define a política de acesso ao meio físico, arbitrando quem terá direito de transmitir e, conseqüentemente, utilizar o meio físico.

* As sub-camadas LLC e MAC são estabelecidas pelo padrão IEEE 802 [Gom86-].

- (3) nível rede - define o roteamento dos pacotes de dados entre diversos nós até atingir o destino. Entretanto, em LANs*, pelo porte e pelo fato de todos os nós estarem conectados em um mesmo cabo, as funções da camada de rede geralmente não são necessárias.
- (4) camada transporte - a camada de transporte tem função básica de aceitar os dados do nível superior, dividi-los, enviá-los e garantir que todos as partes cheguem corretamente ao destino, ou seja, é responsável pela qualidade da transmissão. Existem algumas arquiteturas em que as funções da camada de transporte são implementadas pela camada de enlace, o que frequentemente provoca confusões quando se tenta delimitar as funções de cada camada.
- (5) camada sessão - a função mais importante da camada de sessão é o chamado gerenciamento de diálogo, ou seja, controla a transmissão e recepção de dados considerando a sua semântica, podendo estabelecer delimitadores que organizam os dados de acordo com o seu conteúdo. A camada sessão, comparativamente, é uma camada com funções reduzidas e algumas arquiteturas implementam algumas de suas funções, como a chamada de procedimento remoto (RPC - Remote Procedure Call), no nível de aplicação.
- (6) camada de apresentação - a esse nível deve-se gerenciar a representação de dados para que sistemas diferentes tenham a mesma interpretação semântica em relação à forma de representação binária dos dados. A esse nível define-se também funções como criptografia e compressão de dados.
- (7) camada de aplicação - o aspecto mais importante nesse nível é a definição dos serviços da rede a serem oferecidos às aplicações dos usuários. Um exemplo de serviços que podem ser oferecidos nessa camada a nível de manufatura é o já citado MMS.

* LAN - "Local Area Network " ou redes locais: redes localizadas num mesmo prédio, adequadas para aplicações em manufatura.

Observa-se que, em algumas aplicações, não é necessário que todas as camadas sejam implementadas. Em aplicações próximas ao chão-de-fábrica, é comum a utilização de redes que possuem apenas as camadas física, enlace e aplicação. Nesse caso, as funções das camadas rede, transporte, sessão e apresentação ou não são necessárias ou estão implementadas por funções em outras camadas.

5.2.3 A metodologia como auxílio no desenvolvimento e especificação de redes de comunicação

Geralmente os protocolos especificam a interação entre primitivas de serviços entre seus dois níveis adjacentes, oferecendo serviços à camada superior e requisitando serviços à camada inferior. Para implementar essa interação, o comportamento interno do protocolo muitas vezes é representado por uma máquina de estados estendida. Esse comportamento define o que se chama máquina de protocolos, que pode existir em cada uma das camadas do modelo OSI.

A metodologia permite a modelagem dessa máquina de protocolos passo a passo a partir de um modelo conceitual em PFS e obtendo-se um modelo funcional em MFG. A máquina de protocolos pode, então, ser validada, utilizando-se as técnicas de análise próprias do modelo ou realizando uma simulação.

Na Fig.5-5 é apresentada as etapas para a obtenção de uma máquina de protocolos que especifica o comportamento interno da camada aplicação quando do oferecimento de um serviço MMS. Essa máquina [Alv90-] é denominada MMPM (Manufacturing Message Protocol Machine) e acessa diretamente a camada de enlace. Nesse exemplo, especificou-se uma camada de enlace tipo 3, sem conexão com reconhecimento.

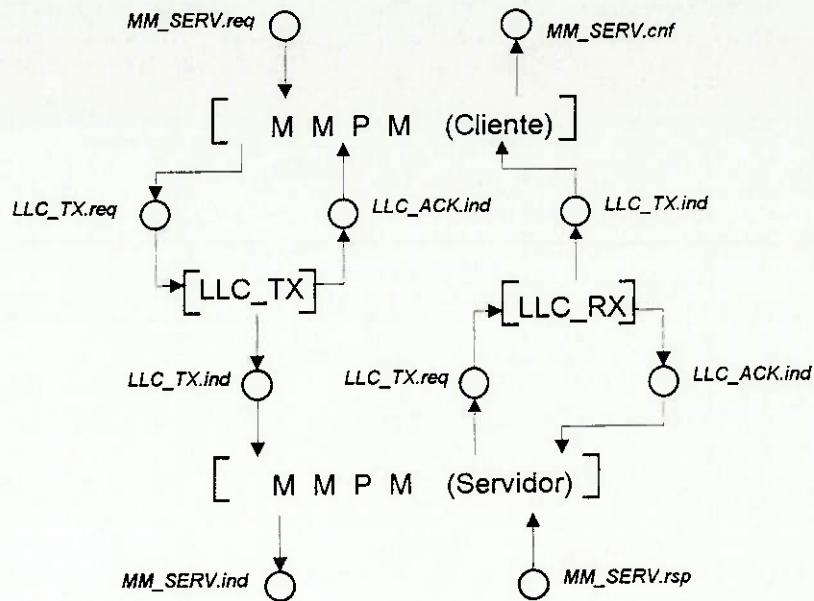


Fig.5-5(a) - Modelo PFS da Máquina de Protocolo do MMS

Os elementos distribuidores representam as primitivas de serviços utilizados no modelo. Para esse modelo, foram utilizados quatro tipos de serviços, conforme mostram as tabelas a seguir.

Serviços

MM_SERV	Serviços do MMS
LLC_TX	Transmissão do cliente no nível de enlace
LLC_RX	Transmissão do servidor no nível de enlace
LLC_ACK	Reconhecimento de mensagem transmitida

No modelo OSI, para cada serviço, podem ser definidas até quatro classes de primitivas, que são:

Classes de primitivas (XXXX é um dos serviços)

XXXX.req	Requisição de serviço
XXXX.ind	Indicação de chegada de mensagem
XXXX.rsp	Transmissão de resposta
XXXX.cnf	Confirmação de chegada de resposta

A Fig.5-5(b) mostra, então, a especificação da máquina de protocolos do MMS em MFG.

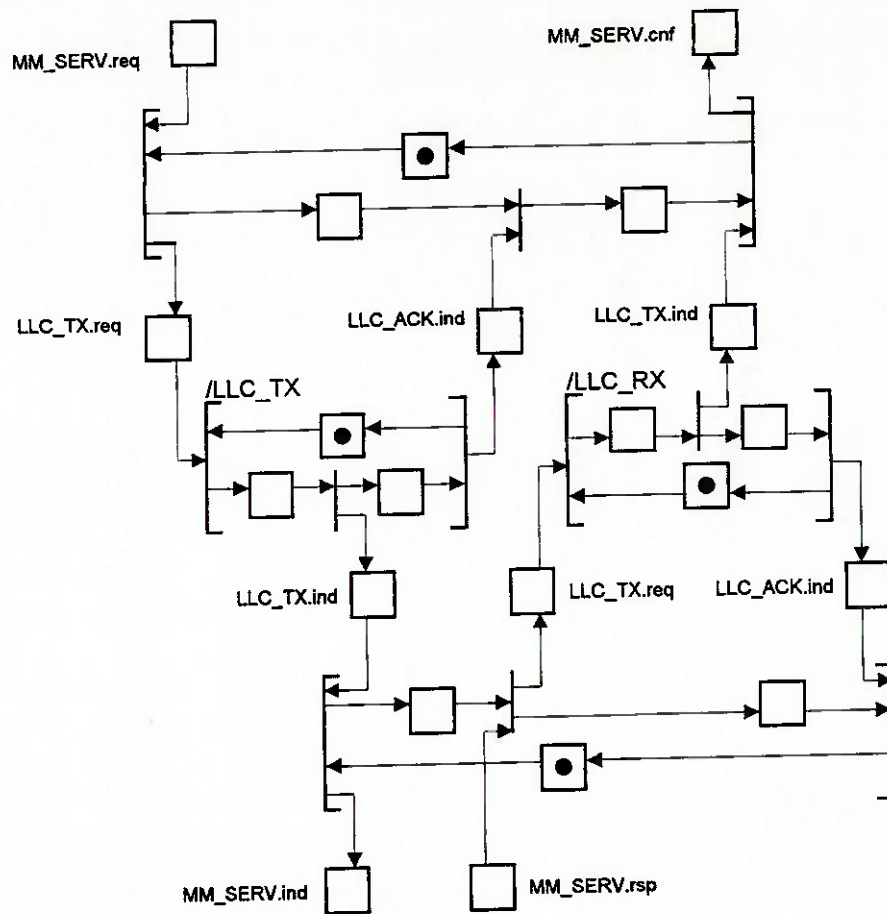


Fig.5-5(b) - Modelo MFG da Máquina de Protocolo do MMS

5.2.4 Modelos para a camada de enlace

Grande parte dos trabalhos pesquisados relacionados a protocolos de comunicação, principalmente a nível de modelagem e análise, refere-se à camada de enlace. A razão principal é a de que essa camada apresenta a maioria dos conceitos e problemas envolvidos com redes de comunicações [Nog91-], sendo que, dependendo de sua especificação, ela pode englobar serviços e funções internas realizadas tipicamente por outra camada. Além disso, para projeto de redes proprietárias, de pequeno ou médio porte, a especificação da comunicação de dados frequentemente confunde-se com a especificação da camada de enlace.

Assim, serão apresentados dois exemplos para a camada de enlace utilizando a metodologia. O primeiro exemplo é uma especificação de um protocolo da sub-camada LLC (Logical Link Control) denominado bit alternante. O segundo exemplo consiste na utilização da metodologia para especificar políticas de controle do acesso ao meio físico (sub-camada MAC).

i) protocolo bit alternante

O protocolo bit alternante é especificado da seguinte forma:

- (1) existem dois estados lógicos, E0 e E1, para o transmissor e para o receptor;
- (2) o transmissor envia uma mensagem e aguarda o reconhecimento (ACK). Se o reconhecimento chegar, ele muda de estado e habilita a transmissão de novas mensagens. Se não chegar após um certo tempo, ele retransmite a mensagem, sem mudar de estado;
- (3) o receptor aguarda a chegada da mensagem e envia um ACK. Se ele estiver no estado E0 e o dado chegar com indicação do estado E0, ele envia um ACK ao transmissor e envia a mensagem para a camada superior. Se ele estiver em um estado e o dado vier

com indicação de outro estado, isso significa que o ACK enviado foi perdido e que a mensagem recebida é uma retransmissão desnecessária. Nesse caso, o receptor envia o ACK mas não envia a mensagem para a camada superior.

A seguir serão apresentadas as etapas definidas da metodologia para a construção do modelo do protocolo bit-alternante em MFG/PFS.

ETAPA 1 - ELABORAÇÃO DO MODELO CONCEITUAL EM PFS

Pela Fig. 5-6(a), existem duas atividades independentes, a transmissão e a recepção, que se sincronizam através das mensagens REQ e ACK.

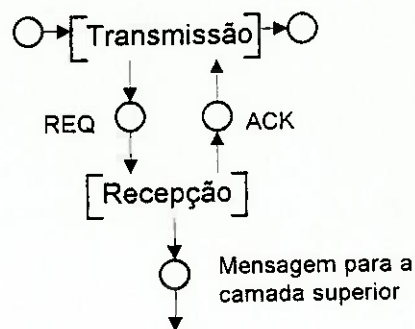


Fig. 5-6(a) - Modelo PFS do protocolo bit-alternante

ETAPA 2 - CONSTRUÇÃO DO MODELO MFG

O sistema retransmite a mensagem caso não se chegue um ACK durante um certo tempo T . Essa condição representa um "time-out" e é modelada pela transição temporizada T . Na recepção, existem duas ações possíveis quando se chega um REQ. $T1$ representa a ação quando a mensagem recebida é um REQ retransmitido (nesse caso, houve a perda de um ACK). $T2$ representa a ação caso o REQ recebido seja o aguardado. Nesse caso, a mensagem é enviada para a camada superior no receptor, conforme representado na Fig. 5-6(b).

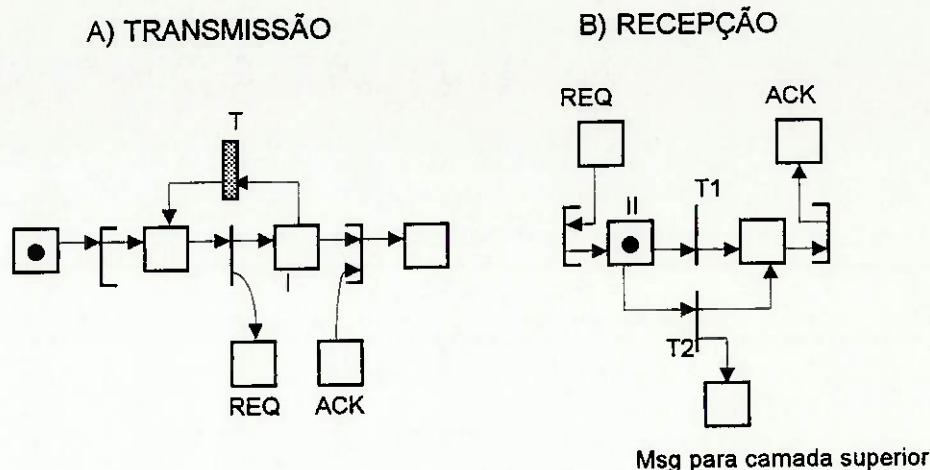


Fig.5-6(b) - Refinamento do PFS em MFG

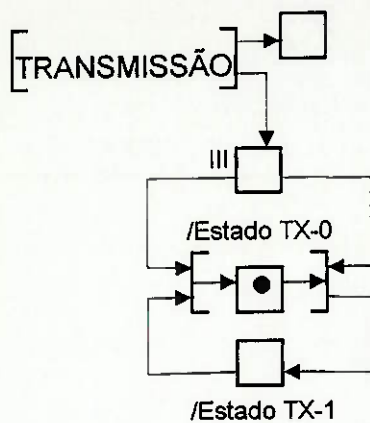
ETAPA 3 - ESPECIFICAÇÃO DA LÓGICA DE CONTROLE

Para representar o controle do protocolo bit-alternante, é necessário adicionar uma lógica adicional no modelo acima. Inicialmente, representam-se os estados 1 e 0, tanto para a transmissão quanto para a recepção. Têm-se, então, conforme a Fig.5-6(c), os estados [Estado0-TX] e [Estado1-TX] no transmissor e [Estado0-RX] e [Estado1-RX] no receptor.

Na transmissão, a mudança de estado ocorre no fim da atividade, quando o transmissor recebe um ACK. O fim da atividade, então, coloca uma marca no box III, indicando a condição para mudança de estado.

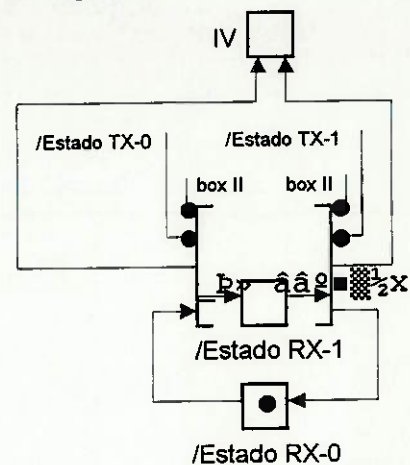
Na recepção, a mudança de estado está condicionada à existência de uma marca no box II, indicando a chegada de uma nova mensagem. Além disso, é necessário também que a mensagem recebida indique que o transmissor está no mesmo estado que o receptor. Portanto, os disparos das transições que efetuam a mudança de estado no receptor estão condicionadas a arcos habilitadores que representam as condições citadas acima. Havendo uma mudança de estado no receptor, o box IV é marcado. O box IV pode ser utilizado como condição para arbitrar o conflito estabelecido na Fig.5-6(b) entre as transições T1 e T2.

A) TRANSMISSÃO



box III - indica que o transmissor deve mudar de estado

B) RECEPÇÃO

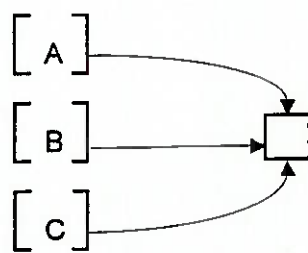


box IV - indica que o receptor mudou de estado e que recebeu a mensagem aguardada

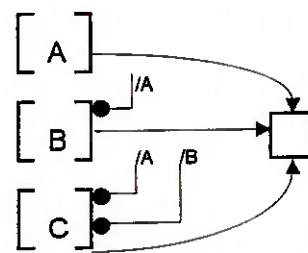
Fig.5-6(c) - Especificação do controle do fluxo

ii) controle do acesso ao meio

O problema de controle do acesso ao meio pode ser representado na metodologia como a definição da política para arbitrar um conflito [Fig.5-7(a)]. Uma solução possível, é definir prioridades para o acesso, como representado na Fig.5-7(b), onde as atividades [A] e [B], nessa ordem, possuem prioridade de transmissão em relação a [C].



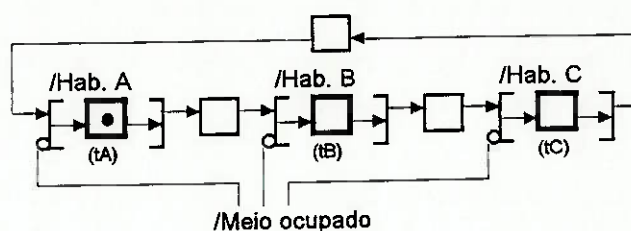
(a) Problema do acesso ao meio



(b) Arbitração por prioridades

Fig.5-7 - Controle de acesso ao meio

Outra forma de solucionar o problema é especificar um controle central que habilita ciclicamente cada uma das atividades para a transmissão. É como se cada uma das atividades recebesse uma ficha para poder transmitir. Possuindo dados para transmissão, e de posse da ficha, a atividade tem acesso ao meio físico. Caso não possua dados, a ficha é passada imediatamente para outra atividade. Essa filosofia de passagem por ficha é conhecido como protocolo "token-bus" e é um dos padrões estabelecidos para comunicação em chão-de-fábrica, conforme representado na Fig.5-7(c).



t_A, t_B, t_C = tempo para verificar se a estação deseja acessar o meio

$[Hab.A], [Hab.B], [Hab.C]$ = atividades que contêm as condições para habilitar, respectivamente, as atividades A, B e C

Fig.5-7(c) controle de acesso por "token-bus"

Pela figura, existem três atividades, $[Hab.A]$, $[Hab.B]$ e $[Hab.C]$, que correspondem às condições para habilitar a transição de fim das atividades em conflito $[A]$, $[B]$ e $[C]$. A marca circula ciclicamente pelas três atividades. Em cada atividade, existe um tempo de permanência da marca, representado por um box temporizado, que corresponde ao tempo para verificar se a atividade deseja acessar o meio. Além disso, o início de uma atividade de habilitação só pode ocorrer se o meio estiver livre, condição representada pelos arcos inibidores na figura.

5.3 A metodologia e o projeto de dispositivos

Além dos aspectos de desenvolvimento de redes de comunicação e de software aplicativo, a metodologia também auxilia a especificação dos dispositivos de interface entre o processo e o sistema de comunicações. São dispositivos como controladores locais de máquinas, unidades de aquisição de dados e controladores de células flexíveis.

Um exemplo aplicado em projeto de dispositivos é a especificação da UAD, mostrado nos capítulos 4 e 5. O modelo apresentado nas Fig.3-12 e Fig.4-16 especifica os módulos funcionais e comportamento dinâmico do dispositivo. A partir dessa especificação, o projetista pode definir os requisitos tanto para o projeto de hardware do dispositivo quanto para a sua aquisição.

Um outro exemplo é a especificação em PFS de um controlador de célula programável para controle de sistemas dirigido a eventos discretos. Essa especificação encontra-se na referência [Myg92-].

CAPÍTULO 6

TÓPICOS RELATIVOS A ANÁLISE E SIMULAÇÃO

Nos capítulos anteriores, foram apresentados as etapas para a obtenção de modelos em PFS e em MFG para a especificação de SIs. Uma vez construídos os modelos, é interessante que os mesmos possam se dispor de ferramentas para análises, antes de se iniciar a implementação. A análise de modelos MFG pode ser realizada tanto a nível quantitativo como qualitativo. Os aspectos envolvendo análise de modelos são discutidos a seguir.

6.1 Análise a nível qualitativo

A nível qualitativo, identificam-se dois aspectos principais a serem analisados no modelo:

- (1) Representação adequada das especificações do sistema real;
- (2) apresentação de algumas propriedades desejáveis, como a limitação, a vivacidade e a reversibilidade* .

No primeiro caso, é importante verificar se a seqüência de eventos definidos pelo comportamento dinâmico do modelo é condizente com o comportamento esperado. Por exemplo, no caso da especificação de um protocolo de comunicação, deve-se verificar se as primitivas são enviadas na ordem correta e se estão de acordo com padrões recomendados por normas técnicas. A análise, nesses casos, pode implicar em correções no modelo, conforme sugere o fluxograma ilustrado na Fig.3-1 do Cap.3.

* Propriedades citadas e apresentadas no Cap.4.

No segundo caso, o modelo é analisado do ponto de vista de suas propriedades. Na verdade, conforme discutido no Cap.4, grande parte desse tipo de análise é sintetizado, considerando modelos baseados em MFG, na tarefa de detecção, análise e correção de "dead-locks".

6.1.1 Análise de "dead-locks"

Existem dois tipos de "dead-locks", o "dead-lock" estrutural e o "dead-lock" condicional [Myg89b]. No "dead-lock" estrutural, o grafo MFG não é vivo, independentemente da marcação inicial. No "dead-lock" condicional, a vivacidade do grafo é condicionada à marcação inicial, ou seja, esse tipo de "dead-lock" ocorre devido a uma marcação inicial inadequada no grafo.

O "dead-lock" estrutural é geralmente relacionado com alguma falha na estrutura do grafo e sua análise independe da dinâmica de disparos do modelo. A sua correção geralmente envolve a correção da estrutura do modelo MFG. Para a sua detecção, tem-se disponível métodos analíticos e simulações.

O "dead-lock" condicional, por sua vez, depende da evolução dos estados do grafo e sua correção muitas vezes é feita através da introdução de um controle adicional, como foi feito no exemplo do Cap. 4 [Fig.4-7]. Entretanto, sua detecção não é trivial, pois carece de ferramentas analíticas eficientes, e freqüentemente só é possível através de simulações.

Em redes de Petri, tem-se disponível uma série de técnicas para efetuar a análise do modelo do ponto de vista de suas propriedades, conforme já citado. Entretanto, a aplicação de tais técnicas no MFG é prejudicada pela utilização de elementos não previstos nas técnicas de análise de redes de Petri, como é o caso dos arcos habilitadores e inibidores. Dessa forma, tais técnicas podem ser aplicadas apenas a alguns casos específicos de modelos em MFG, como é o caso do exemplo da máquina de protocolos da Fig.5-5(b), ou então redefinidas, do ponto de vista formal, para poder ser aplicada também ao MFG.

Existem ainda alguns trabalhos voltados especificamente para a detecção de "deadlocks" estruturais em modelos MFG, como é o caso da análise por balanceamento [Egu80-]. Entretanto, mesmo nesses trabalhos, não se considera nem a marcação da rede nem o comportamento dos sinais de controle.

Assim, para realizar a análise de modelos contendo todos os elementos definidos pelo MFG, incluindo os macro-elementos funcionais e os arcos de controle, é recomendável, então, que seja realizada uma simulação do modelo.

6.1.2 Validação dos modelos

Outro aspecto importante da análise qualitativa, além da verificação das propriedades, é verificar se o grafo representa adequadamente o comportamento do sistema que modela. Uma pequena simulação pode ser útil para verificar se a seqüência de eventos e condições descritas pelo seu comportamento dinâmico está de acordo com o que se deseja.

Esse aspecto é bastante útil, particularmente no que se refere à especificação de protocolos para redes de comunicações e verificar sua conformidade com padrões estabelecidos em normas técnicas.

Seja o exemplo do padrão estabelecido pelo modelo OSI da ISO. Pelo OSI, uma arquitetura de rede pode ser dividida em sete camadas e cada camada interage com a adjacente através de trocas de primitivas de serviços. A ISO estabelece quatro classe de primitivas, conforme comentado no Cap.5, que são o REQUEST (req), o INDICATION (ind), o RESPONSE (rsp) e o CONFIRMATION (cnf). As primitivas devem ser enviadas na ordem apresentada acima, de forma que a máquina de protocolos especificada deve gerar tal seqüência. Considerando a rede de comunicações como um SED, a troca de primitivas pode corresponder a eventos no sistema, e uma simulação pode verificar se a seqüência de eventos gerada corresponde à normalizada pela ISO. Será apresentado posteriormente como exemplo a validação do MMPM, apresentado no Cap.5.

6.2 Análise a nível quantitativo

Na Fig.2-4 do Cap.2, observa-se que, a partir do modelo funcional MFG, define-se um modelo para implementação e um modelo para simulação, de onde são retirados parâmetros quantitativos para a implementação. No desenvolvimento de SIs (e de outros sistemas genéricos) é interessante que alguns parâmetros sejam dimensionados antes de se iniciar a implementação. São parâmetros como tempos de resposta, capacidades de recursos, tamanhos de "buffers" e valores de "time-out", que caracterizam o sistema do ponto de vista quantitativo e, muitas vezes, definem a viabilidade de sua implementação.

Para realizar a simulação objetivando o levantamento de valores quantitativos, é importante inserir a noção de tempo no modelo, através da utilização dos boxes e transições temporizadas, recursos apresentados no Cap. 4.

Simulação de SEDs é tema de diversas investigações, conforme mostra as referências [Has87-, Leu90-]. Para os objetivos desse trabalho, é interessante mostrar que a metodologia também permite uma abordagem quantitativa a nível de análise de desempenho.

Convém observar que os modelos para simulação a nível quantitativo diferem ligeiramente do modelo funcional obtido na especificação. Como o objetivo, nesse caso, é dimensionar alguns parâmetros de projeto, é possível realizar algumas simplificações ou modificar a forma de representação de algumas estruturas, de modo a adequar o modelo para a simulação. O modelo para simulação pode, por exemplo, representar apenas parte da funcionalidade especificada do sistema ou então uma determinada condição crítica em que o sistema deve ser submetido para que seu desempenho seja avaliado.

6.2.1 Aspectos temporais na simulação

Existem vários trabalhos em redes de Petri que tratam do tema simulação a nível de análise de desempenho. Grande parte desses trabalhos são voltados para modelos de redes com parâmetros temporais estocásticos, como é o caso de GSPN e do DSPN* [Mar87-]. Em contra-partida, os simuladores baseados em MFG objetivam aplicações específicas em sistemas produtivos e sistemas de manufatura, e, nesse tipo de aplicação, a utilização de tempos determinísticos geralmente é suficiente para a obtenção de resultados satisfatórios. Assim, tempos associados a usinagem de peças, movimentação de material e preparação de máquinas são representados em simulações por valores determinísticos.

Entretanto, em modelos de avaliação de desempenho e levantamento de parâmetros quantitativos de SIs, muitas vezes pode ser necessária a utilização de parâmetros estocásticos, principalmente se for necessário descrever parâmetros como taxas de transmissão de sistemas de comunicações ou frequência de perdas de mensagens no meio. Ainda assim, muitos problemas podem ser resolvidos sem a utilização de valores estocásticos. Portanto, é importante comentar que, para os objetivos do trabalho, serão adotados exemplos onde a utilização de parâmetros determinísticos não prejudique a análise dos resultados.

6.2.2 Pontos de observação no grafo

Para uma análise eficiente dos resultados, deve-se estipular um valor razoável para o tempo de simulação. Não é possível acompanhar a cada instante a seqüência de disparos de transições no grafo. Dessa forma, é necessário que alguns pontos de observação sejam fixados e convenientemente interpretados. Os seguintes pontos de observação podem ser sugeridos [Mat89-]:

* GSPN: General Stochastic Petri Net; DSPN: Deterministic and Stochastic Petri Net

- (1) tempo de permanência (médio e máximo) de marcas em um box;
- (2) tempo de ausência (médio e máximo) de marcas em um box;
- (3) número total de disparos em uma transição;
- (4) quantidade (média e máxima) de marcas em um box (box do tipo capacidade).

Dependendo dos objetivos da simulação, algumas dessas grandezas podem ser necessárias e outras não.

6.3 Ferramentas para simulação em MFG

Para a simulação de sistemas produtivos, existem diversas ferramentas de simulação baseadas em MFG. Uma dessas ferramentas é o sistema AIPUSP [Luc89-], que consiste de um ambiente computacional para auxiliar o projeto, modelagem e análise de sistemas produtivos, baseado na linguagem MFG/PFS.

O AIPUSP auxilia a construção de modelos através de uma interface gráfica inteligente e de ferramentas específicas que permitem a reutilização de módulos do projeto. A linguagem utilizada é baseada no MFG/PFS, definido na referência [Myg88-]. Através de um tradutor, essa linguagem é convertida para a notação MFG, que é o formato de entrada de dados do analisador.

Para esse trabalho, há a necessidade de se utilizar modelos com transição temporizada, recurso inexistente na maioria das ferramentas disponíveis, incluindo o AIPUSP. Essa necessidade motivou a implementação de um analisador próprio, que incluísse o recurso da transição temporizada, além de fornecer os pontos de observação adequados para a análise de desempenho.

6.3.1 Notação MFG

A entrada de dados em simuladores baseados em MFG é feita geralmente através da notação MFG. A notação MFG compõe-se de quatro módulos:

i) descrição das transições e das conexões

Nesse módulo são descritas as transições e os respectivos boxes conectados. O identificador da transição e os boxes conectados são separados por ":". Os boxes são separados entre si por vírgulas, sendo que boxes de entrada possuem um sinal "-" precedendo-os. Se a transição for temporizada, antes da declaração das conexões insere-se uma estrutura do tipo " = TM/n", onde n é o parâmetro de tempo da transição. Por exemplo, seja a seguinte declaração:

$$T1 = TM/10 : -b2, -b4, b5, b6$$

$$T2 : -b2, b3, b4, b5$$

A transição T1 é temporizada, com parâmetro 10 unidades de tempo, e conectada aos boxes de entrada b2 e b4 e aos boxes de saída b4 e b5. A transição T2 não é temporizada, possuindo um box de entrada (b2) e três boxes de saída (b3, b4, b5).

ii) descrição dos boxes

Nesse módulo são descritos os boxes, seu tipo e respectivos parâmetros. Os boxes podem ser de cinco tipos: NL (simples), CA (capacidade), UP (agrupador), DN (dispersor) e TM (temporizado). Seja o exemplo a seguir:

$$b1 = NL$$

$$b2 = CA/5$$

$$b3 = TM/15$$

Nesse exemplo, b1 é um box unitário, b2 é um box com capacidade de até 5 marcas e b3 é um box temporizado com duração 15 unidades de tempo.

iii) descrição das portas de controle

Nesse módulo são descritos as relações estabelecidas entre boxes e transições através de arcos habilitadores e inibidores. A condição de inibição é representada por um sinal "-", a condição para habilitar ou inibir quando número de marcas (n) é igual ao parâmetro N do box é representada por um sinal "#". Seja o exemplo a seguir:

g1: -b1, #b5

g5: -#b3

g6: b8

Pelo exemplo, a transição T1 possui um sinal inibidor proveniente do box b1 e um sinal habilitador para $n=N$ proveniente de b5. A transição T5 é inibida quando $n=N$ em b3 e a transição T6 é habilitada quando b8 for marcado.

iv) marcação inicial

Por fim, deve-se indicar a marcação inicial, especificando os boxes marcados e a respectiva quantidade de marcas inicialmente em cada um deles. Tem-se então:

m1: 1

m2: 6

Nesse exemplo, o grafo possui inicialmente uma marca em b1 e seis marcas em b2.

6.3.2 Algoritmo para a construção de simuladores

A construção de simuladores baseados em MFG segue o algoritmo esquematizado abaixo na Fig.6-1.

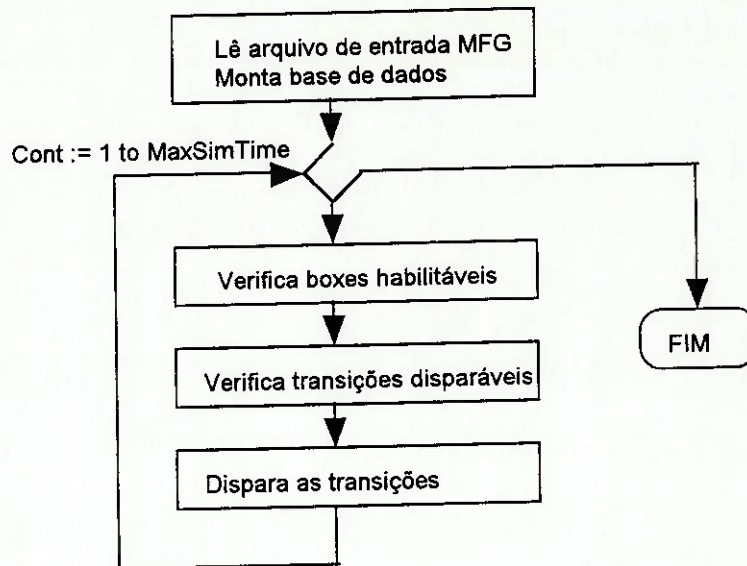


Fig.6-1 - Algoritmo para construção de simuladores MFG

i) Montagem da estrutura de dados

O arquivo de entrada em notação MFG é lido e interpretado e a estrutura de dados é montada. Sugere-se utilizar estruturas de lista ligada e ponteiros com alocação dinâmica de variáveis para a implementação. Nesse módulo, é inserida também a marcação inicial do grafo.

ii) Verificação dos boxes habilitáveis

Box habilitável é aquele que fornece uma condição de disparo "true" às transições conectadas em seus arcos de saída. Um box é habilitável quando:

- (1) for unitário e estiver marcado;
- (2) for temporizado e estiver marcado por um tempo $t = \tau$, onde τ é o parâmetro do box;
- (3) for do tipo capacidade ou dispensor e tiver ao menos uma marca;
- (4) for do tipo agrupador e tiver N marcas, onde N é o parâmetro no box.

iii) Verificação das transições disparáveis

Uma transição é disparável quando:

- (1) todas as pré-condições são estabelecidas, ou seja, quando todos os boxes de entrada na transição forem habilitáveis;
- (2) todas as pós-condições são estabelecidas, ou seja, todos os boxes de saída estiverem vazios ou com quantidade de marcas inferior ao parâmetro de sua capacidade; no caso de box dispensor, uma marca já é suficiente para impor uma pós-condição falsa na transição;
- (3) todas as condições das portas habilitadoras e inibidoras estiverem satisfeitas;
- (4) se a transição for temporizada, todas as condições acima devem ser estabelecidas por um tempo $t = \tau$, onde τ é o parâmetro da transição, para que a transição seja disparável.

iv) Disparo das transições

O disparo das transições é realizado da seguinte forma:

- (1) todas as transições disparáveis são disparadas;
- (2) os boxes unitários, capacidades e dispensadores na entrada da transição são decrementadas de uma marca;
- (3) os boxes agrupadores de entrada são decrementados de N marcas (N é o parâmetro do box);

- (4) os boxes unitários, capacidade e agrupadores de saída são adicionados de uma marca;
- (5) os boxes dispersores de saída são incrementados de N marcas (N é o parâmetro do box);
- (6) as transições temporizadas com pré e pós-condições estabelecidas, além das condições das portas satisfeitas, têm o seu respectivo tempo de atraso para o disparo decrementado de uma unidade de tempo.

Especificou-se como pontos de observação os parâmetros tempo total em que cada box ficou marcado e número total de disparos de cada transição.

6.4 Exemplos

Serão apresentados a seguir dois exemplos, um de análise qualitativa, que corresponde à validação da especificação do MMPM (Manufacturing Machine Protocol Machine), e outro de análise quantitativa, que corresponde ao dimensionamento e avaliação de tempos de resposta, taxa de amostragem e "time-out" do sistema de supervisão de produção de motores.

6.4.1 Validação do MMPM (Manufacturing Message Machine Protocol)

O primeiro exemplo consiste na validação do MMPM. O modelo será validado se a especificação da máquina de protocolos gerar a seqüência de primitivas de serviços corretas definidas pelo MMS (norma ISO/DIS 9506). Da Fig.5-5(b), pode-se derivar o modelo da Fig.6-2. A interpretação dos boxes e transições é mesma utilizada na Fig.5-5(b).

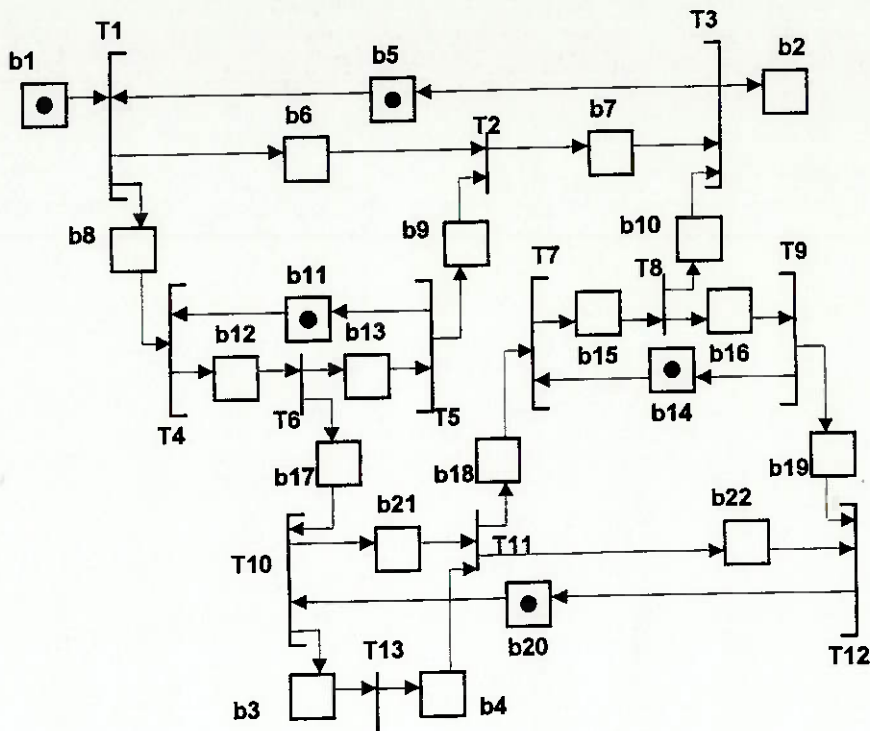


Fig.6-2 - Modelo MFG para simulação do MPPM

Em relação ao modelo para especificação, foi apenas acrescentado a transição T13, que representa a emissão da resposta a uma requisição no momento que se chega uma primitiva do tipo INDICATION no receptor, representado pelo box b3.

Após uma pequena simulação, obteve-se a seguinte tabela contendo seqüência de boxes marcados e a interpretação de cada um dos boxes:

Tempo	Boxes marcados	Interpretação do box
1	b1	MM_SERV.req
2	b8	LLC_TX.req
3	-	-
4	b17	LLC_TX.ind
5	b3, b9	LLC_ACK.ind, MM_SERV.ind

Tempo	Boxes marcados	Interpretação do box
6	b4	MM_SERV.rsp
7	b18	LLC_TX.req
8	-	-
9	b10	LLC_TX.ind
10	b2, b19	LLC_ACK.ind, MM_SERV.cnf

Pela tabela acima, considerando a interpretação dos boxes, pode-se concluir que o modelo gera a seguinte seqüência de troca de mensagens, ilustrado na Fig.6-3.

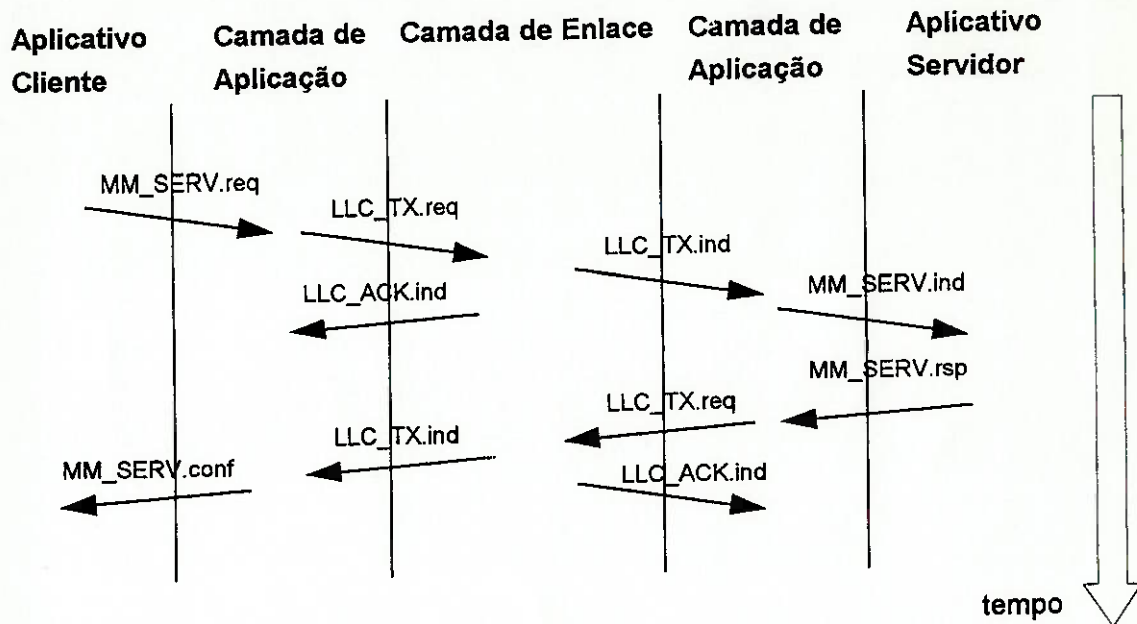


Fig.6-3 - Seqüência de troca de primitivas gerada pelo MMPM

A seqüência de troca de primitivas está de acordo com as recomendações da ISO [Tan89-], de forma que o modelo da máquina de protocolos pode ser então validado.

6.4.2 Dimensionamento de tempo de resposta do sistema de supervisão da produção

Seja o exemplo do SI para a supervisão da produção de motores de uma indústria montadora de veículos. O objetivo da simulação é levantar a relação entre valores apropriados para "time-out" e taxa de varredura da rede. Os modelos utilizados para simulação a nível de análise quantitativa foram modificados em relação aos modelos das Fig.4-14, 4-15 e 4-16. Como exemplo, será considerado um SI para a supervisão de dez máquinas, cada uma com uma unidades de aquisição de dados, formando uma estação. Os modelos e os resultados obtidos serão apresentados a seguir.

i) modelos

O modelo da Fig.6-4 é baseado no modelo da estação-mestre da Fig.4-15(a). Como o objetivo é avaliar parâmetros quantitativos para o projeto do SI, o modelo pôde ser bastante simplificado. Por exemplo, no modelo abaixo, as transições T3 e T4, além do box b3, representam as atividades de transmissão e recepção do modelo completo.

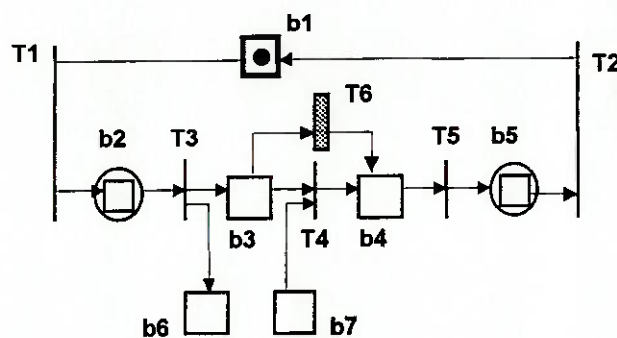


Fig.6-4 - Modelo para simulação do sistema de supervisão da produção

O box B1 e a transição T6 são temporizados. O tempo de B1 corresponde ao atraso para disparar T1 e iniciar a comunicação, ou seja, o tempo de B1 corresponde ao intervalo de tempo entre aquisições de dados do sistema. Cada vez que T1 é disparado, o

o sistema realiza uma varredura nas estações, coletando os dados disponíveis de cada uma delas. T6 corresponde ao "time-out" do sistema de comunicações, ou seja, é o tempo que o sistema aguarda por uma resposta de uma determinada estação.

Do box b6 partem arcos diretos para cada uma das estações. O conflito do box b6 será arbitrado por uma estrutura de ordem fixada. Se a estação possuir dados para resposta, uma marca será colocada no box b7. Nesse exemplo, deverão existir 10 arcos convergindo para b7, cada um proveniente de uma determinada estação.

A Fig.6-5 mostra o modelo da estação 1. Os modelos das demais estações seguem a mesma estrutura da figura, diferindo apenas na enumeração dos boxes e das transições.

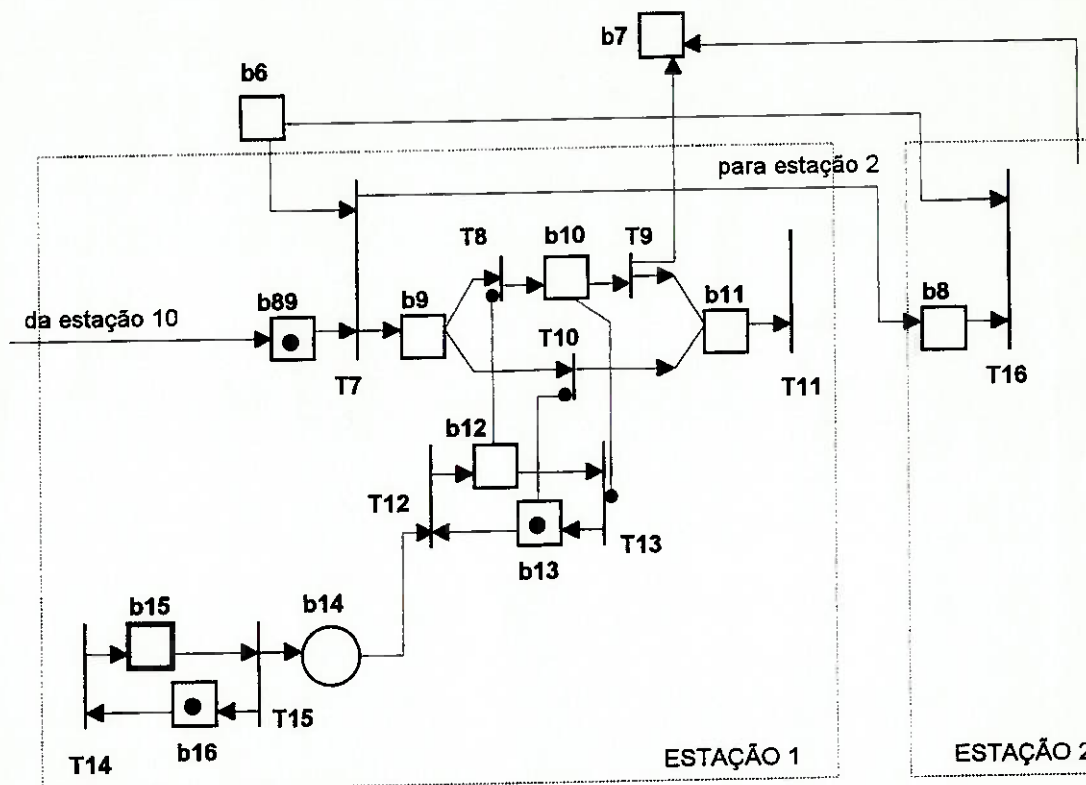


Fig.6-5 - Modelo para simulação de uma estação

Na figura, os boxes b8 e b89 correspondem a condições para habilitar uma determinada estação, ou seja, são boxes que pertencem à lógica de ordem fixada para a arbitragem do conflito. O box b89 marcado indica que, quando b6 estiver marcado, será escolhida a estação 1 e a transição T1 será disparada. O disparo T1 coloca uma marca em b8, de forma que a próxima estação a ser arbitrada será a estação 2.

A seguir, será apresentada uma descrição sucinta da interpretação de alguns elementos importantes no modelo da estação 1. As demais estações possuem boxes e transições análogos.

- (1) b15 e b16: representam a operação da máquina. B16 é um box temporizado que representa o processamento de uma determinada peça. A duração desse processamento é representado pelo parâmetro τ desse box. B15 representa a condição em que a máquina não está processando. Pode-se associar também um tempo a esse box, se o tempo de parada da máquina for relevante;
- (2) b14: o disparo de T15 significa que uma peça foi terminada e representa um evento que deve ser transformado em dado para ser enviado. B14 acumula tais dados, representando, portanto, um "buffer" para armazenamento de dados;
- (3) b12 e b13: representam as condições para arbitrar o conflito entre T8 e T10. B12 marcado significa a existência de dados para enviar e B13, a condição inversa. Havendo marcas em b14, T12 dispara e b12 fica marcado;
- (4) b10: representa a condição de envio de dados. B10 marcado habilita a transição T13, que por sua vez coloca uma marca em b13, indicando que não há dados para transmissão.

ii) parâmetros utilizados

A nível de comunicação, os seguintes parâmetros serão considerados:

- (1) mensagens de tamanho fixo de 12 bytes;
- (2) taxa de transmissão de 9.600bps;
- (3) "overhead" para o processamento de recepção de mensagens de 5ms.

A somatória dos tempos de transmissão, recepção e processamento de recepção é de 25ms. Pelo modelo, de acordo com a implementação do simulador utilizado, tal valor corresponde a 5 unidades de tempo, que é o número de transições em seqüência que uma marca percorre para chegar de b3 a b4, sem passar por T6.

No caso do tempo de processamento das máquinas, serão utilizados valores que variam de 25 segundos a 3 minutos [Mbb93b].

Para o "time-out", são utilizados valores que variam de 40 a 75ms. O tempo de varredura é da ordem de segundos.

No apêndice F encontra-se um exemplo de um dos arquivos de entrada usado no simulador em notação MFG, além dos resultados obtidos nessa simulação.

iii) resultados

Os resultados foram baseados em simulações com tempo total estipulado em 720.000 unidades de tempo, que corresponde a 1 hora de processamento no sistema real. Serão analisados parâmetros como tempo de resposta (TR) e fração do tempo (FT) total gasto com a comunicação. Também serão levantados os parâmetros TMB1, que é o tempo total em que o box b1 esteve marcado, e NDT1, que é o número total de disparos da transição T1 durante o tempo de simulação.

Para "time-out" de 10 UT (50ms) e taxa de varredura entre 500 e 2.000 UT (2,5 a 10s), obteve-se os seguintes resultados, apresentados na tabela abaixo:

	2,5 segundos	5,0 segundos	7,5 segundos	10 segundos
TMB1	591.684	651.156	673.780	685.652
NDT1	1.181	650	448	342

Seja o caso da taxa de varredura de uma aquisição de dados das estações a cada 2,5 segundos. Como o tempo total de simulação é 720.000 e o box b1 esteve marcado pelo tempo de 591.684, conclui-se que o box esteve sem marcas durante o tempo de 128.316. Esse valor corresponde ao tempo total gasto pelo sistema para realizar a comunicação. Em 720.000 unidades de tempo, isso significa que o sistema utiliza 17,8% do seu tempo de processamento para realizar a aquisição de dados, enquanto que os 82,8% restantes estão disponíveis para o processamento de informações e geração de relatórios. O número de disparos da transição T1 (NDT1) indica quantas varreduras foram feitas na rede para a aquisição de dados. Dividindo-se o tempo total para se realizar a comunicação pelo número total de varreduras, obtêm-se o tempo médio de resposta do sistema de comunicações, que é de 108 unidades de tempo, ou aproximadamente, 540ms. Assim, para esse caso, tem-se $TR = 540ms$ e $FT = 17,8\%$.

Para as demais taxas de varreduras, seguindo o roteiro de cálculos apresentado acima, tem-se:

Para taxa de 5,0s: $TR = 525ms$ e $FT = 9,5\%$

Para taxa de 7,5s: $TR = 515ms$ e $FT = 6,4\%$

Para taxa de 10,0s: $TR = 500ms$ e $FT = 4,7\%$

Observa-se que os tempos de respostas TR não variam muito e que a fração de tempo gasto com a comunicação FT diminui à medida em que o intervalo de tempo entre varreduras é aumentado. Deve-se observar também que um tempo muito alto entre

varreduras provoca atrasos significativos na atualização dos dados on-line. Considerando que o menor tempo para o processamento de uma peça é de 25s, um valor de 5,0s na taxa de varredura pode ser adequado para essas circunstâncias.

Além disso, foram realizadas algumas simulações onde variou-se o valor de "time-out" para taxas de varredura de 5,0s. Foram obtidos os seguintes resultados:

Para "time-out" de 50ms: TR = 525ms e FT = 9,5%

Para "time-out" de 40ms: TR = 440ms e FT = 8,08%

Para "time-out" de 75ms: TR = 759ms e FT = 13,0%

Observa-se que, quanto menor o valor de "time-out", melhor torna-se o desempenho do sistema a nível de tempos de resposta e tempo gasto com a comunicação. De acordo com os resultados obtidos, pode-se dimensionar o "time-out" em 40ms e a taxa de varredura em 5,0s. Nesse caso, ter-se-ia um tempo de resposta médio de 440ms e a utilização de 9,5% do tempo total disponível de processamento para a comunicação.

Outros ensaios poderiam ser realizados, como por exemplo, aumentar a quantidade de máquinas (estações) no sistema e verificar o desempenho do SI nessas condições. Para tempos de respostas melhores, poderia se optar em aumentar a taxa de transmissão ou diminuir o tamanho das mensagens. Para condições mais críticas e para dimensionamento de valores de "time-out" mais precisos e mais próximos do tempo de enlace, pode ser necessária a utilização ferramentas com tempos estocásticos, conforme já discutido. Entretanto, considerando que o objetivo desse capítulo é apenas mostrar a capacidade para a realização de análises, tanto a nível qualitativo quanto quantitativo, em modelos baseados em MFG, optou-se por um ensaio em que a utilização de tempos determinísticos não comprometesse a análise dos resultados.

CAPÍTULO 7

CONCLUSÕES FINAIS

Conclusões e contribuições do trabalho

A primeira contribuição do trabalho consiste na ampliação do universo de aplicabilidade da metodologia MFG/PFS. O MFG/PFS foi concebido visando a modelagem e a especificação do controle de sistemas produtivos. Nesse trabalho verificou-se que a metodologia também é eficiente na modelagem e especificação de SIs. Sendo a informação um dos itens principais que definem as relações entre subsistemas da fábrica, pode-se afirmar que esse trabalho também contribui no estudo de integração de sistemas de manufatura.

O MFG mostrou ser, como as redes de Petri, uma ferramenta eficiente para modelagem de SIs a nível funcional. Em relação às redes de Petri, devido à utilização dos macro-elementos funcionais e das portas de controle, existe um certo ganho no poder de expressão da linguagem, embora isso tenha acarretado uma perda no poder de análise, pois os métodos analíticos disponíveis em redes de Petri só podem ser aplicados em modelos MFG que não utilizem os elementos citados acima. De qualquer forma, a análise de modelos MFG ainda pode ser feita através de simulações.

Uma contribuição importante do trabalho foi a utilização do PFS como ferramenta para auxiliar a construção de modelos conceituais de SIs. O PFS constitui-se de uma ferramenta eficiente, pois permite auxiliar a definição do problema sem especificar a sua implementação, ou seja, o PFS auxilia a compreensão do sistema que se deseja controlar sem especificar as estratégias de controle. Em grande parte das metodologias para o desenvolvimento de SIs, a etapa de definição do problema carece de ferramentas formais de auxílio.

Por fim, pode-se sintetizar o trabalho como um proposta de metodologia para desenvolvimento de SIs em ambiente de manufatura, que apresenta as seguintes características:

- (1) Fornece uma técnica estruturada para a construção de modelos em MFG/PFS. A elaboração de modelos utilizando os blocos bem formados e observando as relações dos fluxos secundários entre atividades é semelhante a confeccionar software utilizando programação estruturada. Dessa forma, obtêm-se modelos mais claros, mais legíveis e mais fáceis de serem corrigidos e analisados.
- (2) Propõe uma forma sistematizada e organizada para a modelagem, a especificação e o controle de sistemas caracterizados como SEDs que devem ser considerados na implementação de SIs, como é o caso de sistemas distribuídos, redes de comunicações e dispositivos controladores.
- (3) Permite a representação de estruturas com paralelismo e concorrência nos modelos.
- (4) Possui ferramentas para análise e simulação, de forma que os modelos podem ser validados e verificados quanto ao seu comportamento dinâmico, tanto a nível de análise de propriedades quanto a nível de obtenção de parâmetros quantitativos de projeto.
- (5) É expandível na medida que futuras extensões do MFG, como é o caso do E-MFG utilizado nesse trabalho, poderão ser consideradas.

Sugestões para trabalhos futuros

Além dos aspectos apresentados nesse trabalho, existem outros que podem ser investigados dentro do tema MFG/PFS e modelagem de sistemas de informações. Algumas sugestões para trabalhos futuros serão apresentados a seguir:

i) desenvolvimento de uma extensão estocástica do MFG:

A utilização de parâmetros estocásticos no MFG deve possibilitar o desenvolvimento de simuladores mais eficientes do ponto de vista de análise de desempenho.

ii) desenvolvimento de ferramentas de simulação de extensões do MFG que utilizem marcas individuais:

A utilização de marcas individuais é uma tendência atual considerando o tema MFG e redes de Petri e, por ser uma teoria relativamente recente, tem-se disponível poucas ferramentas de análise e simulações.

iii) estudo da utilização do PFS como ferramenta para a construção de modelos organizacionais

Embora o PFS tenha sido concebido visando aplicação em ambiente de manufatura, sua aplicabilidade pode ser estendida para outras áreas, como por exemplo, construção de modelos conceituais de organizações, auxiliando a definição das necessidades de organizações. O PFS possui a vantagem de ser uma técnica formal e de fácil derivação para modelos funcionais em MFG.

iv) investigação de métodos analíticos para a verificação de propriedades, validação de modelos e detecção de "dead-locks"

Métodos como construção de grafos de alcançabilidade e análise de invariantes podem ser investigados quanto da possibilidade de aplicação em modelos MFG contendo portas de controle e macro-elementos funcionais.

Referências bibliográficas

- [Agw79-] AGERWALA, Tilak. Putting petri nets to work. **Computer**, vol.12, n.12, p.85-94, dezembro, 1979.
- [Alv90-] ALVES, Luiz C. A. & PEDROZA, Aloysio C. P. Metodologia para o desenvolvimento do protocolo mms para controle de robôs. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 8., Belém, SBA, 1990. **Anais**. Belém, 1990, p.463-468.
- [Bol87-] BÖLZING, D. et al. Cim-integrated engineering concepts-dreams and reality. **Industrial & Production Engineering**, vol.11, n.3, p.28-43, 1987.
- [Bor86-] BORJA, Jorge. **Ingenieria de software**. Buenos Aires, editorial Kapelusz, 1986.
- [Bru87-] BRUIJNING, J; SPECS CONSORTIUM. Evaluation and integration of specification languages. **Computer Networks and ISDN Systems**, n.13, p.75-89, 1987,
- [Bry88-] BRYANT, Sheff. Implementation of cell control using mms. In: Enterprise Networking Event '88 International, Baltimore, EUA, 1988. **Proceedings**. EUA, 1988. p.1-15.
- [Bur79-] BURCH JR., John G.; STRATER, Felix R.; GRUDNITSKI, Gary. **Information systems: theory and practice**. 2. ed. EUA, John Willey & Sons, 1979.
- [Cao89] CAO, Xi-Ren - A comparison of the dynamics of continuous and discrete event systems. **Proceedings of the IEEE**, vol.77, n.1, p7-12, jan, 1989.
- [Cas88-] CASSARO, A.C. **Sistemas de informações para tomada de decisões**. SãoPaulo, Livraria Pioneira Editora, 1988.
- [Coh89-] COHEN, Guy et al. Algebraic tools for the performance evaluation of discrete event systems. **Proceedings of the IEEE**, vol.77, n.1, p39-58, jan, 1989.
- [Dav87-] DAVIS, William S. **Análise e projeto de sistemas**. Trad. de Newton Dias de Vasconcellos. Rio de Janeiro, LTC, 1987.
- [Egu80-] EGUCHI, Fujio. **Deadlock analysis of a discrete system and its application to system design**. Tokyo, 1980. 103p. Dissertação (mestrado). Department of Control Engineering, Tokyo Institute of Technology.
- [Gan79-] GANE, Chris & SARSON, Trhish. **Structured system analysis: tools and techniques**. Prentice Hall, Inc., 1979.

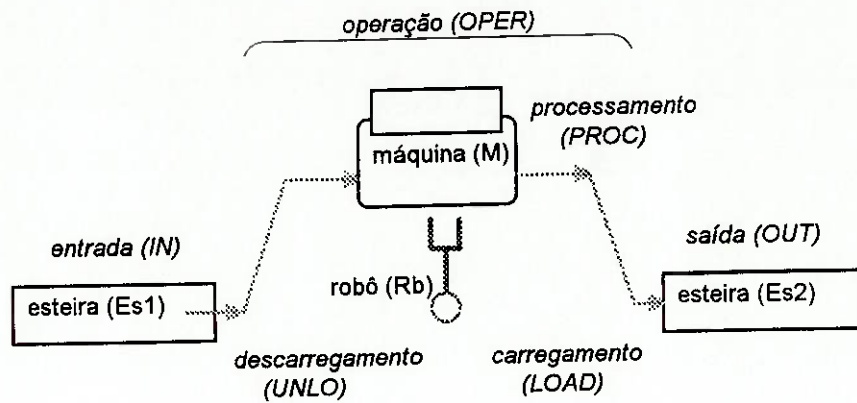
- [Ger87-] GERELLE, E. G. R.; KIRMSER, P. G. The methodology and architecture of computer integrated manufacturing. In: BERNOLD, T. **Artificial intelligence in manufacturing**. Holanda, Elsevier Science Publisher, 1987. p263-267.
- [Giu91-] GIUSTI, Armando E. **Descripción y validación de hardware**. Campinas, R.Vieira Gráfica e Editora, 1991
- [Gom86-] GOMIDE, F. A. C; NETTO, M. L. A. **Introdução a automação industrial informatizada**. Buenos Aires, editorial Kapelusz, 1986.
- [Has80-] HASEGAWA, Kensuke; MASUDA, Ryosuke. Mark flow graph and its application to complex sequential control system. In: INTERNATIONAL CONFERENCE ON SYSTEM SCIENCE, 13., Hawaii, 1980. **Proceedings. SICE**, 1980. p194-203.
- [Has87-] HASEGAWA, Kensuke et al. Simulation os discrete production systems based on Mark Flow Graph. **System Science**, vol.13, n1-2, p1-22, 1987.
- [Has88-] HASEGAWA, Kensuke; MIZUTANI, Takahashi. A survey on industrial automation and robotics. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL, 3., São Paulo, 1988. **Anais (tutoriais e surveys)**. São Paulo, ABCPAI/SUCESU, 1988. p129-147.
- [Ito88-] ITO, Yoshimi. Conceptualizing the future factory system. **Manufacturing Review**, vol.1, n.4, p252-258, dezembro, 1988.
- [Kus88-] KUSIAK, A. Computer integrated manufacturing: a structural perspective. **IEEE Network**, vol.2, n.3, p14-22, maio, 1988.
- [Lat83-] LATHI, B.P. **Modern Digital and Analog Communication Systems**. Japan, Holt-Saunders International Editions, 1983.
- [Leu90-] LEUNG, Ying-Tat; SURJ, Rajan. Performance evaluation of discrete manufacturing systems. **IEEE Control Systems Magazine**, vol.10, n.4, p77-86, junho, 1990.
- [Luc89-] LUCENA, Carlos J.P. et al. The specification of a knowledge based environment for the design of production systems. In: SYMPOSIUM ON INFORMATION CONTROL PROBLEMS IN MANUFACTURING TECHNOLOGY, 6., Madrid, 1989. **Proceedings**. Madrid, IFAC, 1989.
- [Mar87-] MARSAN, M.A. et al. An accurate performance model of csma/cd bus lan. In: ROZENBERG, G. **Lecture Notes in Computer Science**, n.266. Springer-Verlag, 1987. p146-161.

- [Mat89-] MATTIAS, Hamilton F. **Metodologia para análise de desempenho de sistemas de computação descritos através de redes de Petri temporizadas estendidas**. São Paulo, 1989. Dissertação de mestrado. Departamento de Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo.
- [Mbb93a] MERCEDES BENZ DO BRASIL. **SGP - Sistema de Gerenciamento da Produção: Documentação técnica**. São Paulo, MBB, 1993.
- [Mbb93b] MERCEDES BENZ DO BRASIL. **FPLIV: Fertigungsplan. Documentação técnica**. São Paulo, MBB, 1993.
- [Myg88a] MIYAGI, Paulo E. **Control system design, programming and implementation for discrete event production systems by using mark flow graph**. Tóquio, 1988. 125p. Doctor Thesis. Graduate School of Science and Engineering, Tokyo Institute of Technology.
- [Myg88b] MIYAGI, Paulo E.; FURUKAWA, Celso M. Mark flow graph e production flow schema: unificação e estruturação de sistemas de controle. In: CONGRESSO BRASILEIRO DE AUTOMAÇÃO INDUSTRIAL, 3., São Paulo, 1988. Anais. São Paulo, ABCPAI/SUCESU, 1988. p257-264.
- [Myg89a] MIYAGI, Paulo E. et al. Sistemas de manufatura: tendências em modelagem, análise e projeto. In: CONGRESSO BRASILEIRO DE ENGENHARIA MECÂNICA, 10., Rio de Janeiro, 1989. Anais. Rio de Janeiro, ABCM, 1989. p491-494.
- [Myg89b] MIYAGI, Paulo E.; HASEGAWA, Kensuke; TAKAHASHI, Kouji. Mark flow graph (MFG) para modelamento e controle de sistemas de eventos discretos. São Paulo, PMC, EPUSP, 1990. (Monografias em Automação e Inteligência Artificial, no. 1, vol. 1, 1989)
- [Myg90-] MIYAGI, Paulo E. **Modelagem e controle de sistemas produtivos: aplicações da teoria de redes de Petri**. São Paulo, PMC, EPUSP, 1990. (Monografias. Departamento de Engenharia Mecânica, no. 55, 1990).
- [Myg92-] MIYAGI, Paulo E. et al. Specification of task control for production systems. In: The IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems, Kobe, 1992. **Proceedings**. Japan, 1992. p.1505-1510.
- [Mur89-] MURATA, Tadao. Petri nets: properties, analysis and applications. **Proceedings of the IEEE**, vol.77, n.4, p541-580, abril, 1989.
- [Nog91-] NOGUEIRA, José M.S. **Protocolos de comunicação: conceitos, serviços, especificação e testes**. Belo Horizonte, Departamento de Ciência da Computação da Universidade Federal de Minas Gerais, 1991.

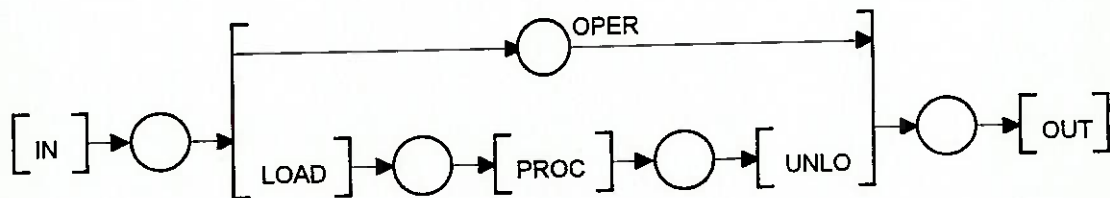
- [Pet77-] PETERSON, James L. Petri nets. **Computing Surveys**, vol.9, n.3, p223-252, setembro, 1977.
- [Pet81-] PETERSON, James L. **Petri net theory and the modeling of systems**. New Jersey, Prentice-Hall Inc., 1981.
- [Rei85-] REISIG, W. **Petri nets: an introduction**. New York, Springer-Verlag, 1985.
- [Rei92-] REISIG, W. **A primer in Petri nets design**. New York, Springer-Verlag, 1992.
- [San91-] SANTOS Fo., D. J.; MIYAGI, Paulo E. Sistemas de eventos discretos e seu controle. In: CONGRESSO BRASILEIRO DE INSTRUMENTAÇÃO E SISTEMAS DE AUTOMAÇÃO/CONGRESSO INTERNACIONAL DA ISA, 1., São Paulo, 1991. **Anais**. São Paulo, ISA, 1991. p2.1-2.12.
- [San93-] SANTOS Fo., D. J. **Proposta do MFG estendido para modelagem e controle de sistemas integrados de manufatura**. São Paulo, 1993. Dissertação (mestrado). Escola Politécnica da Universidade de São Paulo.
- [Sek88-] SEKIGUCHI, T. **Sequential Control Engineering: new theory and design technique**. Institute of Electrical Engineers, Tokyo, Japan, 1988. /em japonês/.
- [Sil89-] SILVA, Manuel; VALETTE, Robert. Petri nets and flexible manufacturing. In: ROZENBERG, G. **Lecture Notes in Computer Science**, n.424. Springer-Verlag, 1989. p374-417.
- [Tan89-] TANNENBAUM, Andrew S. **Computer Networks**. 2. ed. EUA, Prentice-Hall, Inc, 1989.
- [Tei87-] TEICHOLZ, Eric; ORR, Joel N. **CIM-Handbook**. EUA, McGraw-Hill, 1987.
- [Val86-] VALLETE, Robert. Nets in production systems. In: ROZENBERG, G. **Lecture Notes in Computer Science**, n.254, vol.II. Springer-Verlag, 1986. p191-217.
- [Val90-] VALETTE, Robert. **Les reseaux de Petri**. Toulouse, França, L.A.A.S./C.N.R.S., 1990. /apostila de curso/
- [Ydv83-] YADAV, Surya B. Determining an organization's information requirements: a state of art survey. **Data Base**, p3-20, spring, 1983.

APÊNDICE A - EXEMPLO DE APLICAÇÃO DA METODOLOGIA NO FLUXO DE MATERIAL

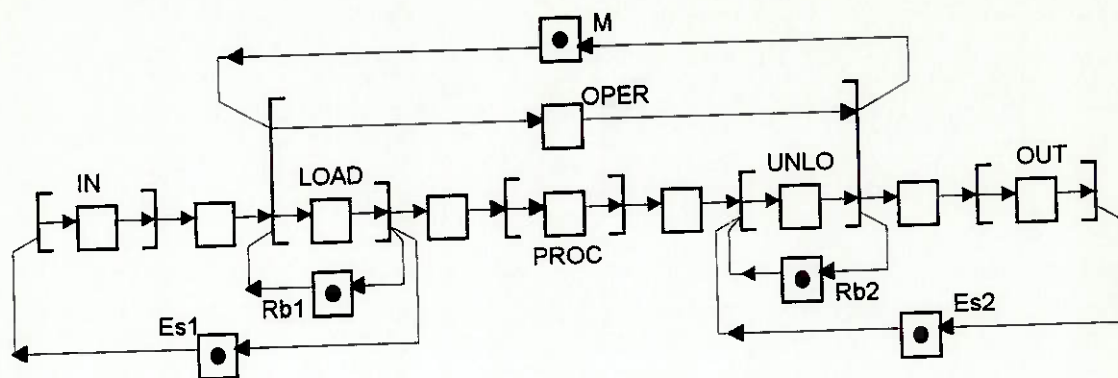
O exemplo a seguir consiste de uma estação contendo uma máquina, que realiza o processamento das peças, um robô, responsável pela carregamento e descarregamento da máquina, e duas esteiras, que representam as entradas e saídas do sistema. Essa situação é ilustrado na figura abaixo.



Observa-se que existe um fluxo de material, iniciando na esteira de entrada (IN) e terminando na esteira de saída, passando pela máquina, onde o material será carregado, processado e descarregado. Pode-se, então, obter o modelo PFS da figura abaixo.



A próxima etapa consiste transformar o modelo PFS em um modelo MFG, alocar os recursos necessários às atividades e especificar um controle adicional, se necessário. Dessa forma, obtém-se o modelo MFG da figura abaixo.



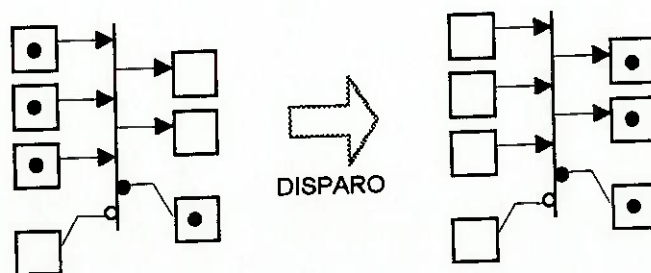
APÊNDICE B - A evolução das marcas em MFG

O estado de um sistema pode ser representado pelo arranjo das marcas no grafo MFG, definindo sua marcação. O comportamento dinâmico do sistema é representado pela alteração dos estados causada pela ocorrência de eventos. Para que isto seja representado no MFG, definem-se as regras de disparo de transições que correspondem à ocorrência de eventos.

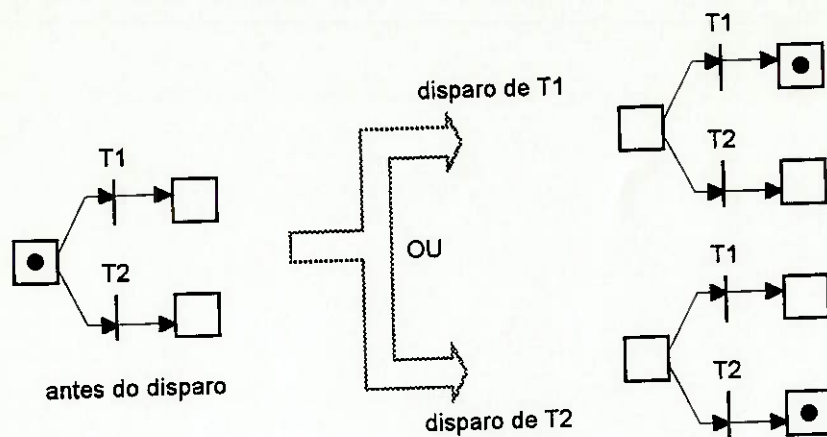
Para ocorrência dos disparos das transições, é necessário que as seguintes pré e pós-condições sejam satisfeitas:

- não existe box no lado de saídas com marcas,
- não existe box no lado de entrada sem marcas,
- não existe arco habilitador que esteja no estado de desabilitação,
- não existe arco inibidor que esteja no estado de inibição.

No disparo, as marcas no interior de todos os boxes no lado de entrada das transições disparáveis desaparecerem e imediatamente surgem marcas no interior de todos os boxes no lado de saída. Isto é, considera-se que o disparo ocorre num intervalo de tempo infinitamente pequeno. Na figura abaixo, um exemplo de disparo é ilustrado.



Existem situações em que existe um não-determinismo na definição da seqüência de disparos, o que caracteriza um conflito, conforme a figura abaixo.



Tanto as pré e pós-condições de T1 como as de T2 estão satisfeitas. Nesses casos é necessário especificar condições adicionais para garantir que a evolução das marcas possa representar eficientemente o comportamento dinâmico desejado do sistema.

APÊNDICE C - EXEMPLO DE UM SI PARA A FÁBRICA

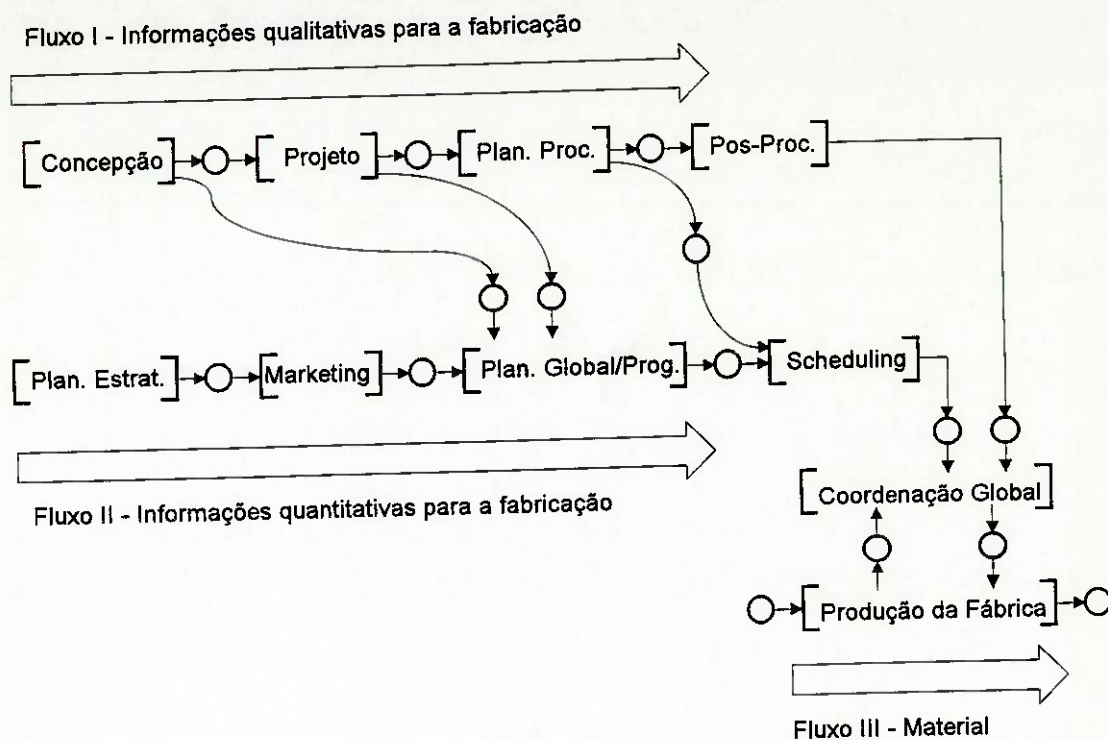
A integração de atividades em uma fábrica é tema de diversas investigações e, devido à sua complexidade, apresenta uma série de abordagens distintas. A abordagem utilizada nesse trabalho será voltada ao controle e gerenciamento dos fluxos de informações da fábrica, ou seja, à especificação de um SI adequado para auxiliar a integração de atividades e subsistemas fabris.

Modelo conceitual dos fluxos de informações em uma fábrica

Em [Has88-], tem-se esquematizado um sistema onde são definidos dois fluxos principais, o fluxo de material e o fluxo de informações, conforme foi apresentado na Fig.2-1. Nessa mesma figura, observa-se que o fluxo de informações foi dividido em dois fluxos principais:

- (1) um fluxo com informações quantitativas para a fabricação, que define aspectos como quanto e quando produzir;
- (2) um fluxo com informações tipo qualitativas para a fabricação, que define aspectos como o que e como produzir.

Em [Bol87-], tem-se esquematizado alguns blocos funcionais e suas interrelações através de trocas de informações. Em [Gom86-, Kus88-] são apresentadas as principais atividades que se relacionam com o ciclo de vida do produto. Com base nas informações retiradas das referências acima, pode-se chegar ao modelo PFS da figura abaixo.



Nesse modelo identifica-se a existência de um fluxo de material (Fluxo III) e de dois fluxos principais de informações, um contendo as informações qualitativas (Fluxo I) e outro contendo informações quantitativas (Fluxo II).

No fluxo I, pode-se identificar atividades como concepção de novos produtos, projeto de peças, planejamento dos processos de fabricação e pós-processamento, que é a atividade responsável pela geração dos programas para máquinas CNC.

No fluxo II, tem-se as atividades de planejamento estratégico e marketing, que definem as necessidades de produção considerando as demandas do mercado e a capacidade produtiva da fábrica. A partir dessas necessidades, é realizado o planejamento global, a programação e o "scheduling", onde são alocados os recursos e definidos os cronogramas de produção.

Ambos os fluxos, I e II, convergem para a atividade de coordenação global. Essa atividade é responsável pelo controle da produção a nível global e, portanto, necessita das informações geradas pelas demais atividades para efetuar adequadamente a sua tarefa.

Organização dos dados

Observa-se que cada uma das atividades gera uma série de informações. A geração dessas informações pode ser auxiliada por ferramentas computacionais. Além disso, tais informações, seguindo a utilização de uma política distribuída para organização de dados, podem ser armazenadas numa estrutura local de dados. A ferramenta computacional, nesse caso, além de auxiliar a execução das tarefas inerentes à atividade, realiza a função de um gerenciador de base de dados local.

A tabela abaixo organiza as atividades citadas no modelo, relacionando-as com o sistema computacional de auxílio e com o tipo de informações e dados gerados.

ATIVIDADE	SISTEMA	DADOS
Concepção	Sistema CAE	lista de peças e dados tecnológicos
Projeto	Sistema CAD	dados geométricos
Plan. de Processos	Sistema CAPP	folhas de processo e dados de ferramentas
Pós-Processamento	Gerenciador de programas NC	programas NC
Plan. Global/Programação	Sistema CAP/MRP II	planejamento da produção (longo prazo)

Scheduling	Sistema de auxílio ao scheduling	scheduling (planejamento a curto prazo)
Coordenação Global	sistema CAM	informações de máquinas

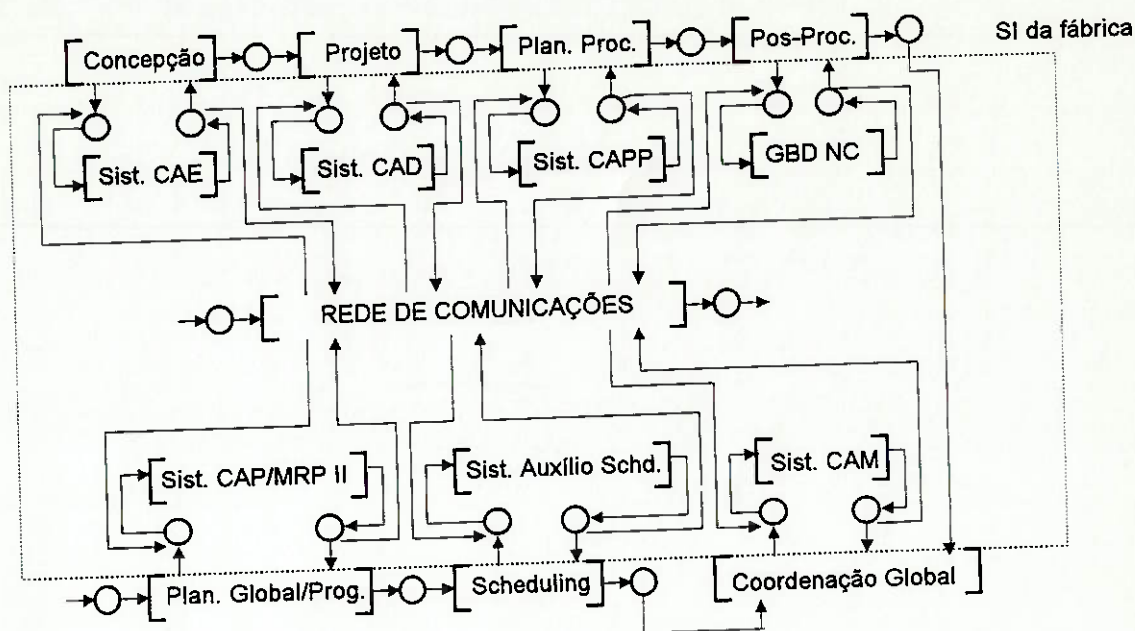
Num sistema distribuído, a base de dados local de cada atividade pode ser compartilhada com outras, permitindo, dessa forma, a troca de informações, aumentando o nível de integração do sistema. No modelo apresentado, esse aspecto é identificado por alguns fluxos secundários entre atividades distintas. Observa-se, por exemplo, que a atividade de programação da produção necessita de informações relativas ao produto, como a lista de peças e seus respectivos dados geométricos, que são fornecidos pelas atividades de concepção e projeto. Além disso, no mesmo modelo, tem-se que a atividade "scheduling" necessita também das folhas de processos geradas pela atividade de planejamento de processos.

A implementação de um sistema distribuído não é trivial e inclui aspectos bastante complexos, como a organização da estrutura de dados do sistema, o controle de consistência de dados compartilhados e a especificação de uma rede de comunicações adequada.

SI da fábrica

Na figura abaixo tem-se esquematizado o sistema de informações da fábrica, onde cada uma das atividades é auxiliada por um sistema computacional, que por sua vez são integrados entre si através da rede de comunicações. Observa-se que a esse nível de detalhamento, identificam-se imediatamente dois problemas. O primeiro consiste no compartilhamento das estruturas locais de dados, gerenciadas pelos sistemas computacionais. Deve-se criar uma estrutura que permita o acesso a estruturas remotas de

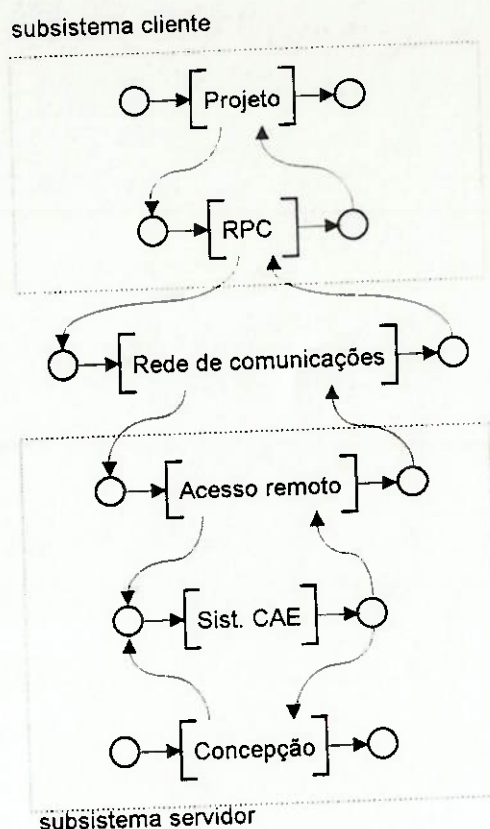
dados. O segundo problema consiste na especificação de uma rede de comunicações adequada. Ambos os aspectos serão abordados a seguir.



i) acesso a dados remotos

Para implementar o acesso remoto e o compartilhamento de dados locais, é recomendável a utilização de uma filosofia "cliente-servidor", conforme discutido no Cap.5. Nesse tipo de filosofia, a atividade que requisita um acesso a uma base de dados remota é denominada cliente e a que fornece os dados é chamada servidor. O cliente requisita um dado através de uma chamada de procedimento remoto (RPC - Remote Procedure Call) [Tan89-]. O RPC utiliza os serviços oferecidos pela rede para realizar a comunicação, tornando-os transparentes ao aplicativo local. No servidor, existe um módulo de interface com a rede que realiza o acesso à base de dados remota.

Essa estrutura está descrita na próxima figura, onde a atividade de projeto requisita uma consulta remota à base de dados do sistema CAE.



Esse tipo de estrutura pode ser aplicada às demais atividades que necessitem compartilhar os dados locais ou acessar dados remotos.

ii) rede de comunicações

Para que possa haver o compartilhamento das estruturas de dados, é necessária a definição de uma rede de comunicações. A especificação da rede consiste basicamente na especificação de uma arquitetura de protocolos adequada. A metodologia pode auxiliar a especificação dessa rede através da especificação das máquinas de protocolos nos diversos níveis de abstração em que se divide o problema das redes.

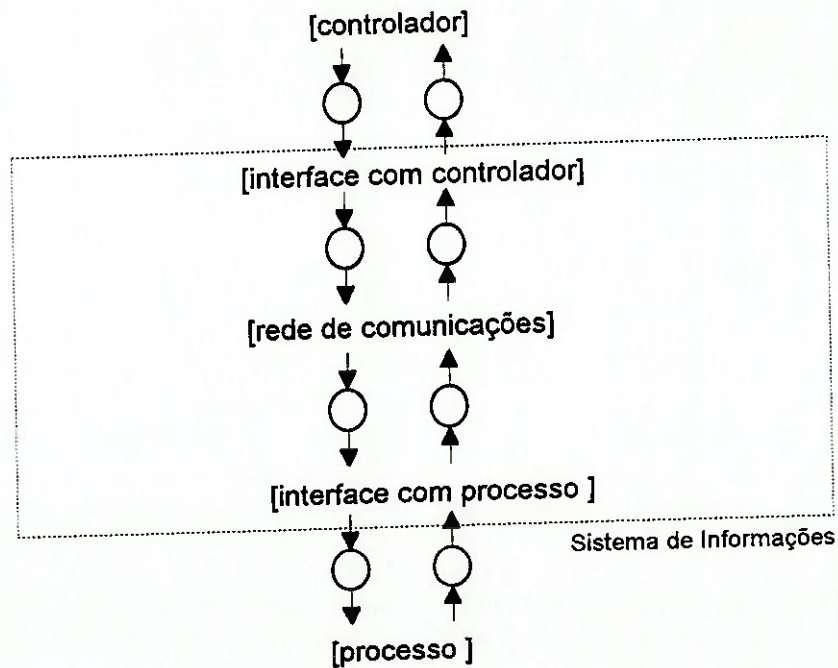
Cada uma das máquinas pode ser validada, verificando se a seqüência de primitivas gerada está de acordo com os padrões adotados. O modelos em MFG/PFS possuem

recursos que facilitam a modelagem do comportamento da máquina de protocolos e técnicas eficientes para a realização de análises.

APÊNDICE D - EXEMPLO DO DESENVOLVIMENTO DE UM SI PARA UMA CÉLULA FLEXÍVEL DE MANUFATURA

O exemplo a ser apresentado a seguir é o desenvolvimento de um SI para controlar o fluxo de informações de uma célula de manufatura. Nessa célula de manufatura, o nível gerencial é representado por um controlador programável, que realiza o controle da célula, monitorando seu "status" e tomando ações para que o comportamento dinâmico do processo aproxime do comportamento especificado. Nesse exemplo, o objetivo do SI é gerenciar o fluxo de dados do sistema, fornecendo periodicamente o "status" do processo ao controlador e enviando aos dispositivos as respectivas ações de controle.

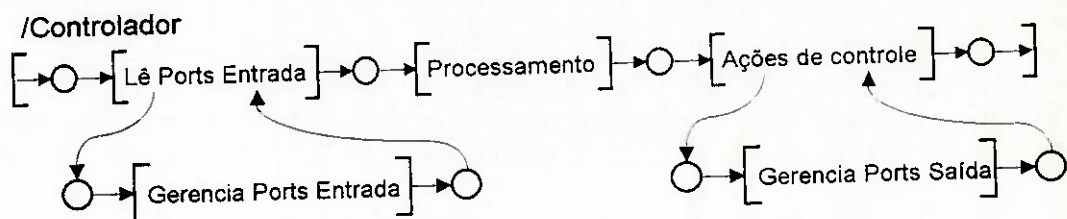
A figura abaixo corresponde ao modelo conceitual do sistema. Nesse modelo, estão representados o controlador, o processo e o SI. O SI compõe-se de três partes, uma que corresponde à interface com o controlador, outra que realiza a interface com o processo e a terceira, que é a rede de comunicações.



A seguir serão descritas as atividades apresentadas no modelo acima.

Controlador

A figura abaixo apresenta a funcionalidade do controlador, do ponto de vista de especificação do SI.



O controlador, funcionalmente, necessita verificar o estado do processo, realizar um processamento e tomar as ações de controle necessárias para garantir que o sistema apresente o comportamento desejado. O estado do processo é definido por "ports" de entrada, que podem ser implementados por variáveis booleanas ou por registradores. As ações são definidas por comandos representados por condições nos "ports" de saída. O SI, nesse exemplo, deve ser responsável por gerenciar esses "ports", atualizando periodicamente os "ports" de entrada e realizando as ações definidas pelos "ports" de saída. Cada uma das atividades do modelo será discutida a seguir.

i) processamento

O processamento utiliza uma estrutura interna de dados do controlador. Essa estrutura de dados deve representar as atividades e condições do processo a cada instante, refletindo a sua evolução de estados e seu comportamento dinâmico. Um controlador programável deve, portanto, permitir que essa estrutura de dados possa variar de acordo

com o processo que se deseja controlar. Uma forma de implementação é utilizar o próprio MFG/PFS como linguagem de programação do controlador, conforme apresentado em diversos trabalhos [Myg88a, Myg92-].

ii) leitura dos "ports" de entrada

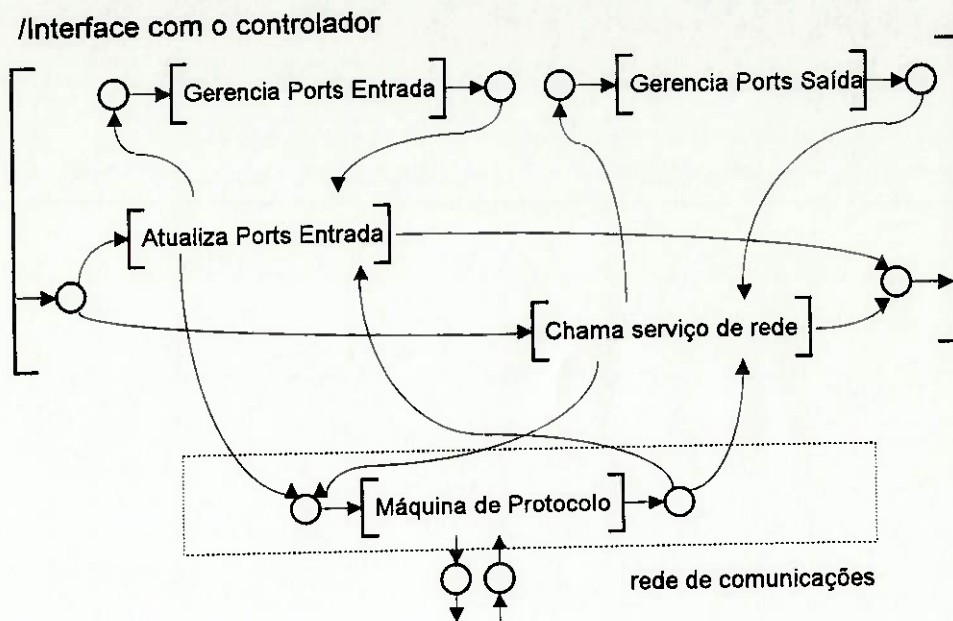
Periodicamente o SI atualiza os "ports" de entrada, colocando valores adequados que representem o estado atual do processo. O controlador, por sua vez, realiza a leitura desses "ports" para atualizar a sua estrutura interna de dados. Dessa forma, é interessante definir uma atividade de gerenciamento dos "ports" de entrada, que será compartilhada por outras duas atividades, da mesma forma que ocorre com a atividade [GBD] no exemplo da Fig.3-10.

iii) ações de controle

Da mesma forma que os "ports" de entrada, os "ports" de saída também necessitam de um gerenciador. O controlador realiza as ações de controle no processo colocando valores nesses "ports". O SI realiza um mapeamento desses valores com comandos pré-definidos. Uma vez definido, o comando é enviado ao processo através da rede de comunicações. Também nesse caso o gerenciador dos "ports" é compartilhado entre duas atividades.

SI - Módulo de interface com o controlador

A interface do SI com o controlador realiza basicamente duas atividades principais, que guardam entre si uma relação de concorrência, que são a atualização dos "ports" de entrada e a chamada de serviços da rede, conforme ilustra a figura abaixo.



A atualização dos "ports" de entrada e a chamada de serviço da rede foram comentados no item anterior. Ambas as atividades requisitam serviços da rede para desempenhar suas tarefas. A atualização dos "ports" de entrada deve ser realizada de tempos em tempos, quando é estabelecida uma comunicação com todos os dispositivos conectados no processo. A chamada de serviço identifica o respectivo serviço ou seqüência de serviços de comunicação representados pelos "ports" de saída.

SI - Módulo rede de comunicação

Cada um dos serviços oferecido pela rede deve instanciar uma máquina de protocolos. Pode-se adotar, por exemplo, o MMPM especificado na Fig.5-5, e, dessa forma, pode-se padronizar a arquitetura da rede de comunicações.

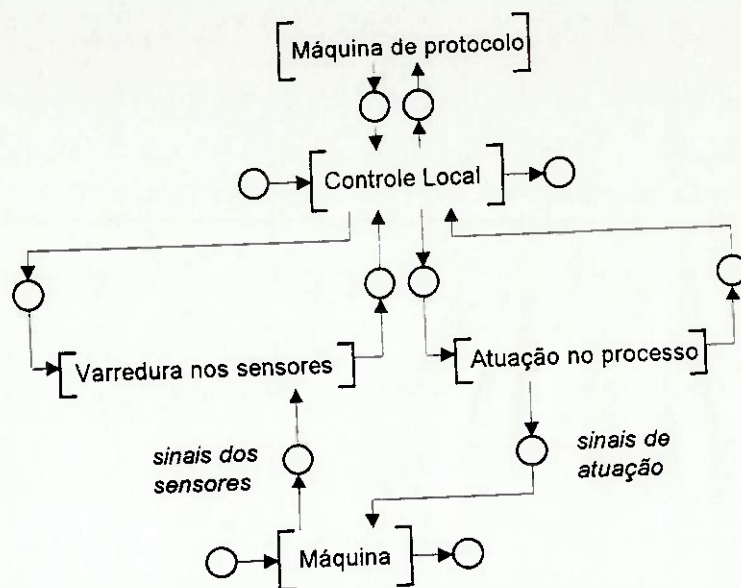
Observa-se que o modelo do MMPM apresenta também um modelo de máquina de protocolo nível enlace. O protocolo nível enlace é bastante simples, não possuindo funções de retransmissão ou identificação de pacotes duplicados. Tal protocolo deve ser suficiente na maioria dos casos de aplicação em células de manufatura. Havendo a

necessidade, entretanto, outras máquinas de protocolos podem ser utilizadas, como foi o caso do protocolo "bit-alternante", especificado no Cap.5. Em especial, tais protocolos podem ser bastante úteis quando se deseja também automatizar a função de "downloading" utilizando o controlador. Nesse caso, os dados transferidos são arquivos cujas dimensões são geralmente bem maiores que o tamanho das mensagens de comunicação consideradas até então. Em certos casos, pode-se haver a necessidade de se partir o arquivo em partes ("frames") para serem transmitidas.

Considerando o aspecto de modelagem, foi citado que cada serviço requisitado instancia uma nova máquina de protocolos. Na fase de especificação utilizando o MFG, se cada serviço corresponder a uma nova máquina de protocolos, o grafo final irá conter uma quantidade muito grande de boxes e transições. Entretanto, a estrutura do grafo é a mesma para cada um dos serviços, sendo que apenas as primitivas trocadas são diferentes. Assim, pode ser interessante que a fase de especificação em MFG sejam consideradas a utilização de marcas individuais [Rei92-, San93-], de forma que se pode utilizar um único grafo, onde as instâncias de cada uma das máquinas de protocolos é representada por uma marca diferenciada no grafo, ao invés de um outro grafo.

SI - Módulo de interface com o processo

A interface com o processo consiste basicamente de um controle local, que monitora a operação do dispositivo sob seu alcance e efetua as ordens enviadas pelo controlador relacionadas a tal dispositivo. Além disso, o controlador local deve se relacionar com a rede, garantindo a comunicabilidade com o controlador do processo. A próxima figura mostra um modelo para a interface com um dispositivo do processo (ex., uma máquina ou um robô) contendo um controlador local.



O controle local recebe uma primitiva de indicação de chegada de serviço, que pode ser uma requisição de "status" ou um comando de atuação no dispositivo. O controle responde com uma primitiva tipo resposta e envia à rede o "status" do dispositivo ou um "frame" indicando serviço realizado. Além disso, o controle local gerencia os sinais de controle dos sensores e dos atuadores do dispositivos.

Uma tendência atual para implementação do controle local consiste em definir objetos e variáveis que possam representar a funcionalidade do dispositivo, criando assim uma espécie de "dispositivo virtual". Assim, é realizado um mapeamento das funções do dispositivo real no dispositivo virtual e, através da utilização dos sensores e atuadores, ações e eventos ocorridos no dispositivo real são refletidos no dispositivo virtual.

Esse tipo de implementação garante uma maior portabilidade com relação aos dispositivos em chão-de-fábrica. Por exemplo, a troca de uma máquina por uma mais eficiente, porém de fabricante diferente, não implica em alterações na estrutura de comunicação, sendo apenas necessário uma redefinição do mapeamento entre o dispositivo "virtual" e o real, pois a funcionalidade do equipamento não foi modificada.

APÊNDICE E - Elementos do MFG estendido (E-MFG)

O E-MFG é uma ferramenta que propõe elevar o poder de expressão do MFG funcional de duas maneiras:

- inserindo atributos às marcas, individualizando-as;
- associar regras de produção às transições, derivadas de IA (Inteligência Artificial), para representar regras adicionais de controle para o disparo das transições.

O E-MFG foi escolhido para esse trabalho pois sua elaboração foi realizada visando facilitar a representação das estratégias de controle a nível de modelagem funcional, obtendo-se uma ferramenta de nível mais alto. O seu estudo pretende também validar aplicações de outras extensões com marcas individuais na metodologia. Seus aspectos principais são descritos a seguir.

i) marcas individuais

Cada marca possui um conjunto de atributos que lhes garantem a individualidade. No exemplo abaixo especifica-se uma marca, que representa uma mensagem em um sistema de comunicações, com três campos de atributos: um campo de endereço de destino, um campo de dados e um campo de "frame". O campo de "frame" é definido quando há necessidade de se dividir o dado em partes para ser transmitido e é representado por um número que identifica a ordem dos frames para recompor o dado original no receptor. Além disso, toda marca em E-MFG possui um campo *c*, que é bastante importante quando do tratamento de marcas individuais compostas.

MARCA = <endereço, dado, frame, *c*>

Assim, o segundo "frame" de uma mensagem de requisição de dados destinada à estação 3 pode ser representado da seguinte forma:

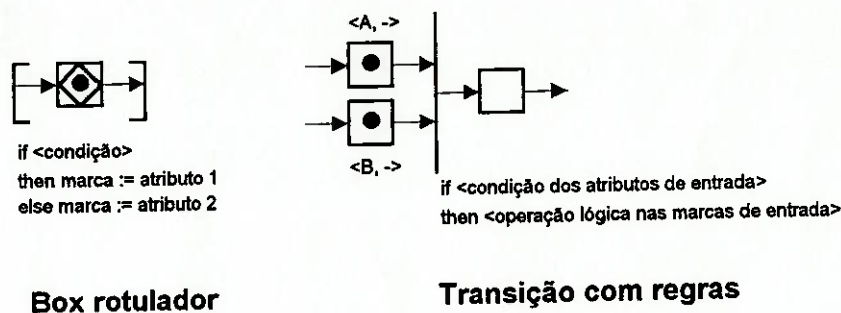
MARCA = <estação 3, REQ, frame 2, ->

ii) regras nos boxes e transições

O E-MFG define um elemento denominado box rotulador, que através de regras do tipo "se-então-senão", cria ou modifica os atributos das marcas.

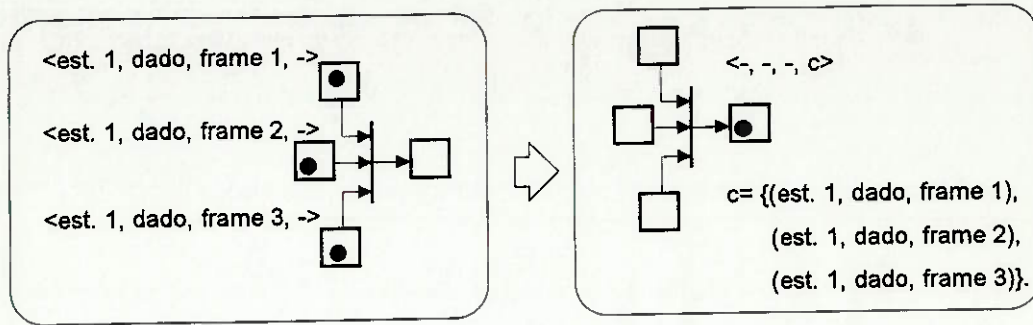
As transições também podem apresentar regras adicionais de disparo, representado também por regras do tipo "se-então-senão" inscritas. Se a transição não possuir tais inscrições, ela se comporta como uma transição comum do MFG. As regras podem representar operações a serem realizadas nos atributos das marcas de entrada para definir os atributos das marcas de saída.

O exemplo abaixo ilustra as considerações acima.



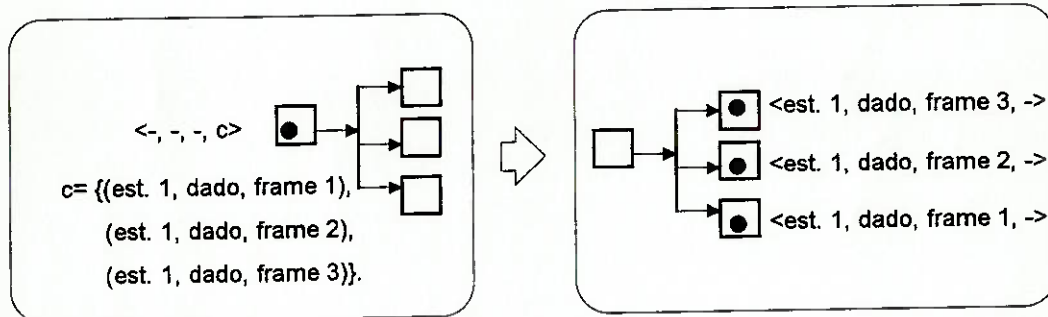
iii) marcas individuais compostas

Uma possível regra para as transições é "agrupar" os atributos de todas as marcas de entrada em uma única marca individual na saída, criando uma marca individual composta. A figura abaixo ilustra esse caso.



A marca composta possui todos os campos de atributos nulos, com exceção do campo de controle, que deve conter os atributos de todas as marcas antes do disparo da transição. Observa-se que o box agrupador do MFG Funcional, quando utilizado com marcas individuais, apresenta esse comportamento.

Da mesma forma, uma marca individual composta que entra numa transição pode ser decomposta em várias marcas individuais simples, bastando apenas associar a respectiva regra na transição, conforme ilustra a figura abaixo.

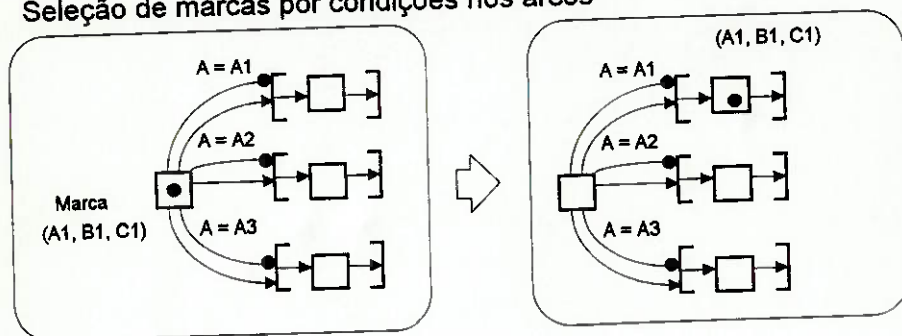


Da mesma forma pode-se associar esse comportamento ao box dispersor do MFG Funcional.

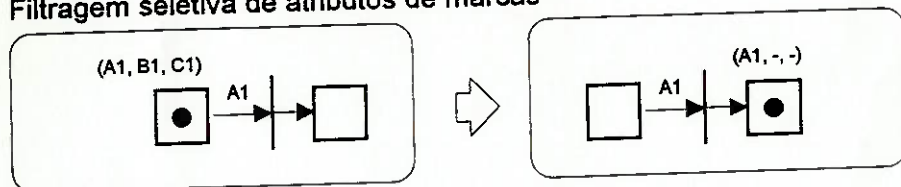
iv) inscrições nos arcos

Uma inscrição num arco habilitador ou inibidor habilita ou inibe uma transição, de acordo com a condição inscrita no arco. Tais condições não realizam modificações nos atributos das marcas. Em contra-partida, uma inscrição num arco direto impõe uma filtragem seletiva na marca. Nesse caso, as inscrições nos arcos diretos estabelecem os campos dos atributos que deverão acompanhar a marca após o disparo da transição. O exemplo abaixo mostra os aspectos mencionados acima.

Seleção de marcas por condições nos arcos



Filtragem seletiva de atributos de marcas



APÊNDICE F - EXEMPLO DE MODELO EM NOTAÇÃO MFG E RESULTADOS

O exemplo abaixo mostra o arquivo de entrada em notação MFG para uma das simulações realizadas. Nesse exemplo, utilizou-se um "time-out" de 10 unidades de tempo, definido pelo parâmetro da transição t6. O tempo entre varreduras é definido pelo box b1, e nesse caso, é de 1000 unidades de tempo. Nas demais simulações, manteve-se os mesmos dados, variando-se apenas os valores dos parâmetros de b1 e t6.

t1: -b1, b2			
t2: -b5, b1			
t3: -b2, b3, b6			
t4: -b3, -b7, b4			
t5: -b4, b5			
t6=TM/10: -b3, b4			
t7: -b6, -b89, b8, b9	t16: -b6, -b8, b17, b18	t25: -b6, -b17, b26, b27	t34: -b6, -b26, b35, b36
t8: -b9, b10	t17: -b18, b19	t26: -b27, b28	t35: -b36, b37
t9: -b10, b7, b11	t18: -b19, b7, b20	t27: -b28, b7, b29	t36: -b37, b7, b38
t10: -b9, b11	t19: -b18, b20	t28: -b27, b29	t37: -b36, b38
t11: -b11	t20: -b20	t29: -b29	t38: -b38
t12: -b13, -b14, b12	t21: -b22, -b23, b21	t30: -b31, -b32, b30	t39: -b40, -b41, b39
t13: -b12, b13	t22: -b21, b22	t31: -b30, b31	t40: -b39, b40
t14: -b16, b15	t23: -b25, b24	t32: -b34, b33	t41: -b43, b42
t15: -b15, b14, b16	t24: -b24, b23, b25	t33: -b33, b32, b34	t42: -b42, b41, b43
t43: -b6, -b35, b44, b45	t52: -b6, -b44, b53, b54	t61: -b6, -b53, b62, b63	t70: -b6, -b62, b71, b72
t44: -b45, b46	t53: -b54, b55	t62: -b63, b64	t71: -b72, b73
t45: -b46, b7, b47	t54: -b55, b7, b56	t63: -b64, b7, b65	t72: -b73, b7, b74
t46: -b45, b47	t55: -b54, b56	t64: -b63, b65	t73: -b72, b74
t47: -b47	t56: -b56	t65: -b65	t74: -b74
t48: -b49, -b50, b48	t57: -b58, -b59, b57	t66: -b67, -b68, b66	t75: -b76, -b77, b75
t49: -b48, b49	t58: -b57, b58	t67: -b66, b67	t76: -b75, b76
t50: -b52, b51	t59: -b61, b60	t68: -b70, b69	t77: -b79, b78
t51: -b51, b50, b52	t60: -b60, b59, b61	t69: -b69, b68, b70	t78: -b78, b77, b79
t79: -b6, -b71, b80, b81	t88: -b6, -b80, b89, b90		
t80: -b81, b82	t89: -b90, b91		
t81: -b82, b7, b83	t90: -b91, b7, b92		
t82: -b81, b83	t91: -b90, b92		
t83: -b83	t92: -b92		
t84: -b85, -b86, b84	t93: -b94, -b95, b93		
t85: -b84, b85	t94: -b93, b94		
t86: -b88, b87	t95: -b97, b96		
t87: -b87, b86, b88	t96: -b96, b95, b97		

Apêndices

b1 = TM/1000
 b2 = DN/10
 b3 = NL
 b4 = NL
 b5 = UP/10
 b6 = NL
 b7 = NL

b8 = NL
 b9 = NL
 b10 = NL
 b11 = NL
 b12 = NL
 b13 = NL
 b14 = CA/5
 b15 = TM/4500
 b16 = NL

b17 = NL
 b18 = NL
 b19 = NL
 b20 = NL
 b21 = NL
 b22 = NL
 b23 = CA/5
 b24 = TM/12630
 b25 = NL

b26 = NL
 b27 = NL
 b28 = NL
 b29 = NL
 b30 = NL
 b31 = NL
 b32 = CA/5
 b33 = TM/17800
 b34 = NL

b35 = NL
 b36 = NL
 b37 = NL
 b38 = NL
 b39 = NL
 b40 = NL
 b41 = CA/5
 b42 = TM/24000
 b43 = TM/6000

b44 = NL
 b45 = NL
 b46 = NL
 b47 = NL
 b48 = NL
 b49 = NL
 b50 = CA/5
 b51 = TM/14530
 b52 = NL

b53 = NL
 b54 = NL
 b55 = NL
 b56 = NL
 b57 = NL
 b58 = NL
 b59 = CA/5
 b60 = TM/9600
 b61 = TM/5000

b62 = NL
 b63 = NL
 b64 = NL
 b65 = NL
 b66 = NL
 b67 = NL
 b68 = CA/5
 b69 = TM/15510
 b70 = NL

b71 = NL
 b72 = NL
 b73 = NL
 b74 = NL
 b75 = NL
 b76 = NL
 b77 = CA/5
 b78 = TM/8400
 b79 = NL

b80 = NL
 b81 = NL
 b82 = NL
 b83 = NL
 b84 = NL
 b85 = NL
 b86 = CA/5
 b87 = TM/5600
 b88 = NL

b89 = NL
 b90 = NL
 b91 = NL
 b92 = NL
 b93 = NL
 b94 = NL
 b95 = CA/5
 b96 = TM/11000
 b97 = TM/5600

g8 : b12
 g10 : b13
 g13 : b10
 g17 : b21
 g19 : b22
 g22 : b19
 g26 : b30
 g28 : b31
 g31 : b28
 g35 : b39

g37 : b40
 g40 : b37
 g44 : b48
 g46 : b49
 g49 : b46
 g53 : b57
 g55 : b58
 g58 : b55
 g62 : b66
 g64 : b67

g67 : b64
 g71 : b75
 g73 : b76
 g76 : b73
 g80 : b84
 g82 : b85
 g85 : b82
 g89 : b93
 g91 : b94
 g94 : b91

m2 : 10
 m13 : 1
 m16 : 1
 m22 : 1
 m25 : 1
 m31 : 1

m34 : 1
 m40 : 1
 m43 : 1
 m49 : 1
 m52 : 1

m58 : 1
 m61 : 1
 m69 : 1
 m70 : 1
 m76 : 1
 m79 : 1

m85 : 1
 m88 : 1
 m89 : 1
 m94 : 1
 m97 : 1

Após a simulação, obteve-se os seguintes resultados, onde TBM_i é o tempo total que o box i esteve marcado e ND_j é o número total de disparos da transição j . O tempo total de simulação é de 720.000 unidades de tempo

TBM1=651156	TBM50=49	ND1=650	ND49=49
TBM2=61290	TBM51=719950	ND2=651	ND50=50
TBM3=61032	TBM52=50	ND3=6510	ND51=49
TBM4=6510	TBM53=6867	ND4=678	ND52=651
TBM5=61986	TBM54=651	ND5=6510	ND53=49
TBM6=6510	TBM55=49	ND6=5832	ND54=49
TBM7=678	TBM56=651	ND7=651	ND55=602
TBM8=6207	TBM57=28821	ND8=159	ND56=651
TBM9=651	TBM58=691179	ND9=159	ND57=49
TBM10=159	TBM59=49	ND10=492	ND58=49
TBM11=651	TBM60=470449	ND11=651	ND59=49
TBM12=89769	TBM61=249551	ND12=159	ND60=49
TBM13=630231	TBM62=6885	ND13=159	ND61=651
TBM14=159	TBM63=651	ND14=160	ND62=46
TBM15=719840	TBM64=46	ND15=159	ND63=46
TBM16=160	TBM65=651	ND16=651	ND64=605
TBM17=6825	TBM66=30325	ND17=56	ND65=651
TBM18=651	TBM67=689675	ND18=56	ND66=46
TBM19=56	TBM68=46	ND19=595	ND67=46
TBM20=651	TBM69=719953	ND20=651	ND68=47
TBM21=31513	TBM70=47	ND21=56	ND69=46
TBM22=688487	TBM71=6651	ND22=56	ND70=651
TBM23=56	TBM72=651	ND23=57	ND71=85
TBM24=719943	TBM73=85	ND24=56	ND72=85
TBM25=57	TBM74=651	ND25=651	ND73=566
TBM26=6921	TBM75=47655	ND26=40	ND74=651
TBM27=651	TBM76=672345	ND27=40	ND75=85
TBM28=40	TBM77=85	ND28=611	ND76=85
TBM29=651	TBM78=719914	ND29=651	ND77=86
TBM30=24202	TBM79=86	ND30=40	ND78=85
TBM31=695798	TBM80=6393	ND31=40	ND79=651
TBM32=40	TBM81=651	ND32=41	ND80=128
TBM33=719959	TBM82=128	ND33=40	ND81=128
TBM34=41	TBM83=651	ND34=651	ND82=523
TBM35=7023	TBM84=72935	ND35=23	ND83=651
TBM36=651	TBM85=647065	ND36=23	ND84=128
TBM37=23	TBM86=128	ND37=628	ND85=128
TBM38=651	TBM87=719871	ND38=651	ND86=129
TBM39=14634	TBM88=129	ND39=23	ND87=128
TBM40=705366	TBM89=659361	ND40=23	ND88=651
TBM41=23	TBM90=651	ND41=24	ND89=43
TBM42=575976	TBM91=43	ND42=23	ND90=43
TBM43=144024	TBM92=651	ND43=651	ND91=608
TBM44=6867	TBM93=4266	ND44=49	ND92=651
TBM45=651	TBM94=715734	ND45=49	ND93=43
TBM46=49	TBM95=43	ND46=602	ND94=43
TBM47=651	TBM96=473556	ND47=651	ND95=44
TBM48=27914	TBM97=246444	ND48=49	ND96=43
TBM49=692086			