BC

BC

FD-1497

JORGE LUIS RISCO BECERRA

ESPECIFICAÇÃO , PROJETO E IMPLEMENTAÇÃO DE UM CONTROLADOR DE CÉLULA FLEXÍVEL APLICADO NA MANUFATURA

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

São Paulo

1993

JORGE LUIS RISCO BECERRA

Engenheiro Electrônico, Universidad Nacional Mayor de San Marcos, Lima - Perú, 1984

Nº USP 1628545

ESPECIFICAÇÃO, PROJETO E IMPLEMENTAÇÃO DE UM CONTROLADOR DE CÉLULA FLEXÍVEL APLICADO NA MANUFATURA

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

Área de Concentração:

Engenharia de Computação e Sistemas Digitais

Orientador:

Professor Dr. Moacyr Martucci Jr.

Nº Func. 512869

São Paulo

1993

Dedico meu trabalho a minha fonte eterna de fé, vontade e força: DEUS.

AGRADECIMENTOS

Ao professor Moacyr Martucci Jr., meu muito obrigado por tudo o que me ensinou nestes últimos anos, pelo apoio e incentivo, quero que saiba que me sinto muito orgulhoso de ser seu discípulo.

Aos professores Sidney e Selma pelo apoio e suas grandes amizades.

A Cezar Jacoby pelo estímulo e sua amizade.

A Graciela e Francisca pela comprensão e carinho.

A Luz Amelia pelo apoio econômico no inicio do curso.

A Mario e Jorge pelo apoio na edição desta dissertação.

Minha gratidão a todos Vocês que me apoiaram e incentivaram, cada um de seu jeito, durante todo este tempo para alcançar este objetivo de minha vida , Muito Obrigado.

SUMÁRIO

Lista de figuras e tabelas

Resumo

"Abstract"

1 INTRODUÇÃO

1.1 Motivação 4

1.2 Objetivos 6

1.3 Abrangência 7

1.4 Metodologia 8

1.5 Estrutura do trabalho 10

2 A CÉLULA FLEXÍVEL E A INTEGRAÇÃO TOTAL DA MANUFATURA

2.1 Conceitos básicos 12

2.1.1 Sistemas de manufatura tradicionais 13

2.1.2 Sistema de manufatura celular 17

2.1.2.1 Célula flexível de manufatura 18

2.1.2.2 Implementação da célula flexível 22

2.2 Integração total da manufatura (CIM) 25

2.2.1 Caracterização do CIM 26

2.2.2 A implementação do CIM, a célula e o controlador 30

2.3 O controlador e os equipamentos da célula 31

3 ESPECIFICAÇÃO DO CONTROLADOR

3.1 Análise do sistema 39

3.1.1 Elementos do sistema 39

3.1.2 Interconexão dos equipamentos 41
3.1.3 Características do sistema 42
3.2 Especificação funcional 44
3.2.1 Perfil de aplicação 45
3.2.2 Unidades funcionais 46
3.2.2.1 Unidade funcional de Aquisição e Atuação 47
3.2.2.2 Unidade funcional de Análise de dados 49
3.2.2.3 Unidade funcional de Gerenciamento de banco de dados 52
3.2.2.4 Unidade funcional de Interfaces com o usuário 53
3.2.2.5 Unidade funcional de Comunicação externa 54
3.3 Características gerais do controlador 55
3.3.1 Arquitetura aberta 55
3.3.2 Sistema modular 57
3.3.3 Disponibilidade e confiabilidade 58
3.3.4 Manutenção 59
3.4 Especificação das interfaces homem-máquina 59
3.5 Descrição operacional 62

4 DEFINIÇÃO DA ARQUITETURA DO CONTROLADOR

4.1 Metodologia de definição do controlador 66
4.1.1 Classificação das metodologias 66
4.1.2 Metodologia do projeto preliminar 69
4.1.2.1 Características gerais 70
4.1.2.2 Análise estruturado 71
4.1.2.3 Análise da concorrência 72
4.1.2.4 Definição das interfaces 73
4.1.2.5 Projeto estruturado 74
4.2 Projeto preliminar do controlador 76

4.2.1 Análise estruturado do controlador 76

4.2.2 Concorrência no controlador 82

4.2.3 Comunicação entre os módulos tarefas 85

4.3 Arquitetura do controlador de célula flexível 87

4.3.1 Arquitetura geral 87

4.3.1.1 Estrutura do nó 89

4.3.1.2 Comunicação externa 90

4.3.2 Hardware básico do controlador 91

4.3.3 Software do controlador 93

4.3.3.1 Sistema operacional 93

4.3.3.2 Software aplicativo 94

5 IMPLEMENTAÇÃO DO CONTROLADOR

5.1 Implementação do hardware 104

5.1.1 Considerações na seleção da via de comunicação 105

5.1.2 Caracterização da via selecionada 109

5.1.2.1 Módulos funcionais 109

5.1.2.2 Sub-vias 112

5.1.2.3 Considerações elétricas e mecânicas 113

5.1.2.3 Sub-sistemas do controlador de célula 114

5.1.3.1 Processador central 115

5.1.3.2 Processador de entrada e saída 117

5.1.3.3 Processador de comunicação exterior 119

5.1.3.4 Integração e teste do hardware 123

5.2 Implementação do Software 124

5.2.1 Padronização do POSIX 125

5.2.2 Sistema operacional 126

5.2.3 Software aplicativo 129

5.2.4 Integração e teste do software 132

5.3 Validação do controlador 133

5.3.1 Simulador dos elementos da célula flexível 134

5.3.1.1 Projeto do simulador 135

5.3.1.2 Plataforma de teste 137

5.3.2 Simulação da arquitetura do controlador 137

5.3.2.1 Estrategia de simulação 139

5.3.2.2 Implementação e resultados 141

6 CONSIDERAÇÕES FINAIS

6.1 Conclusões 148

6.2 Comentários 150

6.3 Continuidade do trabalho 151

6.4 Contribuição acadêmica 153

REFERÊNCIAS BIBLIOGRÁFICAS

LISTA DE FIGURAS E TABELAS

Figura 1.1 Fases da metodologia de desenvolvimento do projeto 9

Figura 2.1 Relação entre os elementos do sistema de produção e o sistema de manufatura 15

Figura 2.2 Sistemas de manufatura 16

Figura 2.3 Célula de manufatura manual 20

Figura 2.4 Célula de manufatura automática 20

Figura 2.5 Exemplo do uso da tecnologia de grupo 24

Figura 2.6 Modelo hierárquico da manufatura 28

Figura 3.1 Sistema aberto de automação aplicado na manufatura 40

Figura 3.2 Níveis hierárquicos das funções do controlador 46

Figura 3.3 Diagramas de estado mostrando os modos de operação 65

Figura 4.1 Elementos do diagrama de fluxo de dados 71

Figura 4.2 Tipos de interface de comunicação 75

Figura 4.3 Diagrama de fluxo de dados do controlador 77

Figura 4.4 Controlador no nível de tarefas 83

Figura 4.5 Interfaces de comunicação entre tarefas 86

Figura 4.6 Arquitetura do controlador 88

Figura 4.7 Hardware básico do controlador 92

Figura 4.8 Projeto estruturado no aplicação-central 98

Figura 4.9 Projeto estruturado da supervisão 99

Figura 4.10 Projeto estruturado do controle entrada-saída 100

Figura 4.11 Projeto estruturado da comunicação superior 101

Figura 4.12 Projeto estruturado da coordena-equipamentos 102

Figura 5.1 Importância das considerações de seleção das vias de comunicação 108

Figura 5.2 Módulos funcionais e sub-vias do padrão IEEE P1014 110

Figura 5.3 Sub-sistema processador central 116

Figura 5.4 Sub-sistema processador entrada e saída 118

Figura 5.5 Sub-sistema de comunicação exterior 120

Figura 5.6 Modos de interconexão do controlador
num sistema aberto de automação 122

Figura 5.7 Distribuição dos sistemas operacionais
no hardware do controlador 127

Figura 5.8 Estrutura do software aplicativo 130

Figura 5.9 Diagrama de fluxo de dados do simulador 138

Figura 5.10 Modelo da arquitetura do controlador 140

Figura 5.11 Diagrama de fluxo do algoritmo de experimentação 142

Tabela 3.1 Requisitos de tensão e corrente 48

Tabela 3.2 Tempos de resposta típicos 48

Tabela 4.1 Tipo de dados do DFD 81

Tabela 5.1 Freqüência de variação no experimento 144

Tabela 5.2 Resultados num experimento 146

Tabela 5.3 Tempos de resposta típicos no controlador 147

RESUMO

Apresenta-se a especificação, projeto e implementação de um controlador de célula flexível aplicado na manufatura, visando ser elemento de um sistema aberto de automação.

Este trabalho apresenta uma metodologia de desenvolvimento, cujas fases possuem ferramentas de projeto como a especificação padrão, a análise estruturada e a simulação, que serviram para definir a arquitetura do controlador e a validação do mesmo.

A implementação do controlador de célula foi baseada numa arquitetura aberta, e foram utilizados recursos computacionais cujas especificações e interfaces são definidas por padrões reconhecidos internacionalmente.

Finalmente, o controlador de célula atingirá um alto grau de flexibilidade permitindo a adaptação do controlador a qualquer processo de manufatura de forma simples, e também poderá ser utilizado como equipamento para o ensino e pesquisa das técnicas modernas aplicadas à automação.

ABSTRACT

It presents the especification, design, and implementation of the flexible cell controller applied to manufacture, which will be element of the automation open systems.

This work presents a suitable development methodology, whose steps has design tools as standard especification, structure analysis, and the simulation, which was utilized to the architecture definition and validation of the cell controller.

The implementation was based on the open architecture and were used computer resources whose especifications and interfaces are defined by international standars.

Finally, the cell controller will achieve a great flexibility which permits the simple adaptation to every manufacture process, and will be an equipment for the education and research of the modern techniques applied to the automation.

1 INTRODUÇÃO

O trabalho, tem como objetivo principal realizar o projeto e implementação de um controlador de célula flexível aplicado na manufatura; o projeto do controlador estará dirigido para que ele possa ser elemento de um sistema aberto de automação, conceito que será definido posteriormente, e principalmente para possuir uma arquitetura aberta, esta característica permitirá a adaptação do controlador a vários processos de manufatura, de forma simples.

A seguir apresenta-se os problemas que as empresas possuem para poder participar no mercado atual, e que originaram a aplicação de automação, e a evolução da aplicação da automação na manufatura, começando com a inclusão dos equipamentos de automação e terminando com os sistemas abertos; o objetivo final é de ressaltar a atualidade e a colaboração do trabalho de dissertação.

A produtividade e a qualidade são as principais preocupações das empresas para continuar atuando dentro do mercado atual, especialmente quando o mercado, possui características tão complexas (1), que exigem a aplicação de novas teorias, metodologias e tecnologias no sistema produtivo, com o objetivo de manter um alto grau de competitividade.

Além das características do mercado, a internacionalização das economias, aliada à formação de mercados regionais na Europa (Comunidade Econômica Européia), América do Norte (Estados Unidos, Canadá e México) e na Asia (Japão, Austrália, Koréia, Singapura e Malásia), aumentam a concorrência entre as empresas (2), particularmente, quando os países membros dos diferentes mercados, investem fortemente na aplicação da automação industrial.

Dentro desse contexto, no âmbito nacional as preocupações são as mesmas, ou seja, aumento da produtividade e da qualidade são as metas, que são atingidas através da aplicação de graus de automação, considerados como fator de competitividade, principalmente, quando se deve enfrentar as empresas internacionais, cujas estruturas produtivas utilizam os grandes avanços tecnológicos.

A manufatura, com a introdução de equipamentos de automação, como por exemplo, controladores numéricos computarizados (CNC), robôs, controladores programáveis e veículos autônomos, tem desenvolvido grandemente, obtendo melhoras na produtividade, homogeneizando a qualidade, incrementando sua capacidade automática, tornando mais flexível o controle (capacidade de se adaptar a vários processos) e estendendo fortemente sua relação com outros setores do sistema produtivo das empresas, que antes foram considerados independentes da manufatura (3).

Esta forte relação entre os setores da empresa, criou a necessidade de integração de todo o sistema produtivo, com a finalidade de compartilhar a informação dos processos e dos produtos, o que deverá permitir resolver os problemas da falta de flexibilidade da produção, dos curtos ciclos de vida dos produtos, da complexidade da produção de certos produtos e obter um maior controle e planejamento da manufatura, e como conseqüência destes benefícios, aumentar a capacidade de competência da empresa.

Visando a premissa anterior, é que aparece uma nova concepção, que impulsiona a integração coorporativa através dos recursos computacionais, permitindo o fluxo contínuo da informação em todos os departamentos da fábrica.

O CIM (Computer Integrated Manufacturing) ou Integração Total da Manufatura, nesse sentido representa o maior conceito aplicado ao sistema de

manufatura moderno, e vem sendo aplicado com certas dificuldades, pela sua complexidade e necessidade de adaptação da cultura existente atualmente (3), este último ponto, inspira a comunidade científica para propor novas alternativas nas diferentes áreas que envolve o CIM.

Um dos principais problemas encontrados na Integração Total é a interconexão dos equipamentos de automação, atualmente uma das principais fontes de preocupações de usuários e fornecedores.

Este problema, que já era encontrado em outras áreas, tinha inspirado a ISO (Internacional Standardization Organization), que criou o comitê técnico para interconexão de sistemas abertos, OSI (Open System Interconnection); deste comitê nasceu o modelo de referência para a interconexão de sistemas abertos que se tornou norma internacional ISO/TC97/SC16 7498/1 (4).

No ambiente industrial, o interesse pela integração de uma vasta gama de equipamentos totalmente heterogêneos, em todos os níveis da fábrica, fez que a General Motors junto a vários fornecedores como a Digital Equipment Corp., Hewlett-Packard e IBM embarcassem no estudo da proposta OSI, procurando solução aos problemas de conectividade aplicado aos equipamentos de automação.

Deste trabalho surge o projeto MAP (Manufacturing Automation Protocol), que envolve conjuntos de protocolos de comunicação aplicados na automação da manufatura; e paralelamente surge o grupo de trabalho NBS (National Bureau of Standards)-Boeing CO., abraçando a mesma idéia, mas voltada para as comunicações de escritório (fase de projeto), definindo o TOP (Technical Office Protocol). Estes protocolos procuram padronizar a interconexão dos diferentes equipamentos (5).

Desde então, estes esforços vêm sendo multiplicados, tomando força no ambiente industrial, pois estes protocolos padrões permitirão impulsionar a Integração Total da Manufatura (CIM), e logo permitirão perceber a necessidade de equipamentos de automação para a aplicação de sistemas abertos.

Os Sistemas Abertos de Automação são caracterizados pelo suporte de um conjunto consistente de padrões referentes à tecnologia de informação e perfis funcionais, que são utilizados para especificar interfaces, serviços e formatos, de forma a garantir a interconectividade, interoperabilidade e portabilidade de equipamentos, aplicações, dados e pessoal (6).

Como conclusão desta introdução e dentro desta última fase de evolução da automação, correspondente ao conceito de Integração Total e aos Sistemas Abertos de Automação aplicados na manufatura, é que aparece o controlador de célula flexível como equipamento de automação, cuja função é controlar os processos de manufatura representados pelas células flexíveis, que são conjuntos de equipamentos de automação, arranjados metodologicamente e dedicados à produção de uma família de partes (1), (7), (8).

Considerando a mesma filosofia das estruturas abertas, é que se propõe o projeto e implementação do controlador baseado numa arquitetura aberta, de forma que possa controlar diferentes formas de células flexíveis, isto é, facilitar a utilização do controlador em vários processos de manufatura, esta última característica, ausente nas atuais implementações de controladores.

1.1 Motivação

Três são os assuntos que serviram de motivação para o desenvolvimento do

presente trabalho, a integração do chão de fábrica, as implementações atuais dos controladores e a proposta dos sistemas abertos de automação, estes assuntos são explicados neste item.

Primeiro, dentro do contexto da implementação da Integração Total da Manufatura; um problema bastante importante é a integração do chão de fábrica com o sistema de informação gerencial, que significa, unir a informação obtida no processo, com a informação dos departamentos superiores como, projeto, vendas, gerência; isto é importante porque se trata de dois mundos com características diferentes, e cuja junção representa a chave da integração total.

O controlador de célula aqui proposto possui estas funções, permite a integração de todos os elementos do chão de fábrica, elementos de natureza heterogênea, e comunica este nível da fábrica com o sistema de informação gerencial.

Segundo, as implementações existentes do controlador possuem diversidade de formas e utilizam especialmente equipamentos que foram especificados para outras aplicações, como microcomputadores e controladores programáveis; conseguindo no final do projeto, controladores rígidos, de vida limitada e dirigidos a células específicas.

Esta falta de flexibilidade nestes controladores, motivaram para propor uma arquitetura, cujas características envolvem a flexibilidade, expansibilidade, fácilidade de atualização e adaptação a várias formas de células flexíveis.

Finalmente, a concepção dos sistemas abertos de automação, que permitirão obter um sistema de manufatura flexível, confiável e adaptável a qualquer processo, e cuja estrutura baseia-se na utilização de protocolos de comunicação e recursos de informação padrões, influíram também na determinação de colocar o controlador como

elemento de um sistema aberto de automação aplicado na manufatura.

1.2 Objetivos

A dissertação, aqui apresentada, consiste da especificação, projeto e implementação de um controlador de célula flexível aplicado na manufatura.

O primeiro passo do trabalho é apresentar a definição do comportamento funcional do controlador, através da definição de unidades funcionais, que consistirão no agrupamento das diferentes funções do controlador, com a intenção de definir adequadamente o escopo do controlador dentro do sistema de automação; como metodologia desta parte do trabalho será utilizado um padrão de especificação informal aplicado para software, mas visando o projeto do controlador como um todo.

Dando continuidade determinar-se-á a arquitetura do controlador, que deverá ser atingida como uma conclusão lógica da especificação funcional e da utilização de outra metodologia de projeto (Design Approach for Real Time) (9), que neste caso será usada na etapa denominada de projeto preliminar do controlador. Esta arquitetura deverá ser funcional, flexível e modular.

É objetivo do trabalho colocar nos atributos do controlador uma das características mais importantes, de ser um elemento de um sistema aberto de automação, e ainda mais, exige-se que a arquitetura do controlador possua uma estrutura aberta, nesse sentido deverá possuir os mecanismos e utilizar os padrões necessários que o tornará um equipamento aberto.

A implementação do controlador deverá ter como objetivo a utilização dos recursos disponíveis no mercado, especialmente aqueles de características abertas,

aproveitando a tendência dos fornecedores de desenvolver recursos totalmente abertos (10), (11), (12).

Então, dever-se-á selecionar os recursos que serão utilizados na implementação como, módulos processadores, via de comunicação padrão, sistema operacional, aplicações e protocolos de comunicação padrões.

O controlador, como equipamento, deverá representar uma solução ao problema da integração dos equipamentos heterogêneos do chão de fábrica e implicará na implementação do nível de célula no CIM. Através deste trabalho, também procura-se enfatizar a importância do controlador para o CIM.

Finalmente, a implementação do controlador visa ser um elemento do Laboratório de automação proposto em (13), e que esta sendo desenvolvido no Laboratório de Sistemas Digitais do Departamento de Engenharia de Computação e Sistema Digitais da Escola Politécnica da Universidade de São Paulo; cujo objetivo é de obter um laboratório de automação para estudos de soluções de engenharia aplicada aos problemas de automação, para a verificação de novas metodologias e para o ensino das técnicas e das aplicações dos sistemas digitais em todas as áreas da automação.

1.3 Abrangência

A preocupação principal da dissertação é o desenvolvimento do equipamento de automação, neste caso o controlador de célula flexível.

No que diz respeito ao modelo do sistema aberto de automação, é utilizado um modelo referenciado em (13), indicando que existem outros modelos, mas com a mesma filosofia de integração total (14).

Na especificação funcional utiliza-se uma metodologia informal deixando em aberto a possibilidade de utilizar uma especificação formal como foi encontrado em (15).

No que se refere à implementação e à utilização dos recursos do mercado, deixa-se claro que será explorado ao máximo estes recursos, e não está dentro do escopo do trabalho, a implementação de um sistema operacional específico, nenhuma parte do hardware, nem os aplicativos próprios do controlador, mas se enfatiza que os ditos recursos serão selecionados com as conclusões obtidas, na pesquisa realizada nas etapas anteriores do projeto.

Outra coisa importante que é bom também deixar claro, é a utilização de protocolos padrões de comunicação, pois não faz sentido a implementação de um protocolo específico aplicado ao controlador de célula flexível, uma vez que é premissa que ele seja aberto.

1.4 Metodologia

A metodologia utilizada terá as mesmas etapas do ciclo de vida das metodologias para o projeto de software de sistemas de tempo real, que será tratada como uma extensão aplicada ao projeto do controlador.

Trabalhando em cima dessa estrutura metodológica, espera-se obter um projeto estruturado, totalmente modular e de fácil manutenção, operação e atualização.

As etapas do ciclo de desenvolvimento do projeto do controlador de célula flexível serão definidas a seguir, e são representados na figura 1.1:

a) Especificação funcional: constituída pela análise de sistema e pela especificação propriamente dita, onde serão definidas as funções, os atributos, as interfaces externas, os modos de operação e as restrições do sistema.

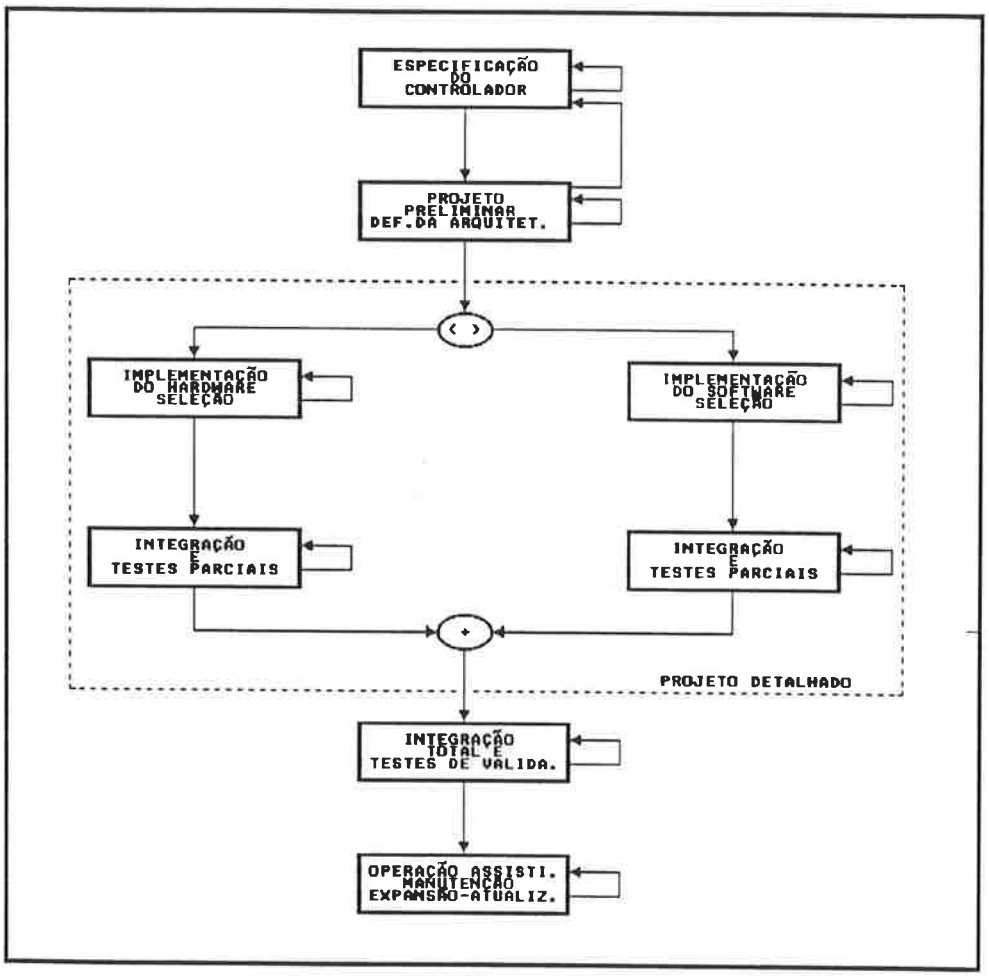

Figura 1.1 Fases da metodologia de desenvolvimento do projeto

b) Projeto preliminar: representa a consolidação da arquitetura do sistema, analisando-se a concorrência das funções e as interfaces de comunicação, além de apresentar o projeto estruturado do software do controlador.

c) Projeto detalhado: nesta fase, o projeto será dividido em dois sub-projetos, correspondente ao hardware e ao software respetivamente, é onde será realizada a

seleção dos recursos que serão utilizados.

d) Integração e testes de parciais: uma etapa que será feita paralelamente nos sub-projetos, os módulos são gradualmente integrados e testados, até constituir o sistema respectivo, também faz parte do projeto detalhado.

e) Integração total e testes de validação: a integração representa a união dos sub-projetos, e o teste final representa a validação da especificação funcional, aqui serão simulados os elementos da célula num microcomputador e para depois ser implantado no controlador para os teste finais.

Cabe ressaltar que durante todas as fases são aplicadas diversas técnicas e ferramentas de projeto, para ajudar a atingir os objetivos propostos.

1.5 Estrutura do Trabalho

A dissertação consiste de sete capítulos, basicamente seguindo a estrutura da metodologia explicada no item anterior e com a adição dos capítulos de orientação teórica e de considerações finais.

O primeiro capítulo como esta sendo apresentado, tem como objetivo mostrar porque o interesse no desenvolvimento deste projeto e os objetivos que serão atingido ao longo deste trabalho.

O segundo capítulo mostra o fundamento teórico necessário, com o objetivo de centralizar o trabalho dentro do escopo da área de atuação.

No terceiro capítulo, assim como no quarto são abordadas as fases da

especificação funcional e o projeto preliminar, respetivamente, da metodologia, ambos são complementares e como conclusão será definida a arquitetura do controlador.

O quinto capítulo explicará a seleção dos recursos que permitirão implementar o controlador de célula flexível, e a implementação propriamente dita, corresponde à fase do projeto detalhado e da integração. Aqui também será mostrada a metodologia de validação do controlador utilizando como ferramenta de teste, a simulação.

O último capítulo mostrará as conclusões, alguns aspectos críticos e as contribuições acadêmicas que serão originadas deste trabalho .

2 A CÉLULA FLEXÍVEL E A INTEGRAÇÃO TOTAL DA MANUFATURA

Neste capítulo apresenta-se o suporte teórico em torno do conceito da célula flexível e da Integração Total da Manufatura ou CIM (Computer Integrated Manufacturing), conceitos necessários para poder localizar alguns dos objetivos deste projeto.

São objetivos deste capítulo, primeiramente, mostrar a importância do sistema de manufatura celular no sistema de produção, apresentar o conceito de célula e as considerações que existem na implementação da mesma, considerando que a célula flexível é o processo sob o qual o controlador atua.

Em seguida, mostrar a relação existente entre Integração Total (CIM) e a célula, e especialmente com o controlador de célula flexível. A idéia deste último ponto é de poder mostrar, posteriormente, porque é necessário que o controlador deve ser projetado como elemento de um sistema aberto de automação da manufatura.

Finalmente apresentar as características dos elementos que compõem a célula flexível, para demostrar a falta de interoperabilidade existente nestes equipamentos.

2.1 Conceitos Básicos

O sistema de manufatura corresponde às atividades do processo de manufatura realizadas no chão de fábrica, este sistema possui os mecanismos, procedimentos e equipamentos para transformar os materiais em produtos acabados.

Para complementar as funções do sistema de manufatura, este mantém uma forte

relação com outros elementos do sistema de produção, relação baseada na troca de informação e fluxo de materiais, como pode ser visto na figura 2.1 (16), onde é mostrado o sistema de manufatura, o controle de qualidade e os outros elementos do sistema de produção.

Cabe ressaltar que qualquer mudança que acontece no sistema de manufatura reflete diretamente nos outros elementos do sistema de produção. A existência desta relação, converte ao sistema de manufatura no elemento em torno do qual o sistema produtivo é projetado (16).

A seguir, serão apresentados as características dos sistemas de manufatura tradicionais, como referência inicial, para depois definir o sistema de manufatura celular, de considerável utilização na atualidade (7), (17), (18).

2.1.1 Sistemas de Manufatura Tradicionais

Os sistemas de manufatura que tem sido utilizados tradicionalmente, são os seguintes:

a) Departamentos de produção ou "Job Shop": é o mais antigo dos sistemas de manufatura, as operações são organizadas pelas funções das máquinas ou departamentos de máquinas, encontrando-se furadeiras em um local, enquanto que os tornos ou as fresas estão em outro, segundo este arranjo fisico (lay-out), os processos de diferentes pedidos possuem diferentes caminhos ou seqüências através das máquinas, como pode ser visto na parte a da figura 2.2.

As características mais importantes dos departamentos de produção (Job-Shop) são: flexibilidade e variedade nos processos, tempo de configuração ou "setup time"

longo e variável (o setup é o tempo de configuração das máquinas para processar um novo pedido), utilização de máquinas de uso geral, os lotes de fabricação vão de pequeno para médio, existe uma forte manipulação do material e encontra-se um trabalhador por máquina.

As desvantagens deste sistema são derivadas da sua grande flexibilidade, obtendo-se tempos longos de manufatura, uma qualidade não uniforme e consideráveis estoques intermediários no processo. São os sistemas com maiores dificuldades para colocar graus de automação, mas são necessários pois muitos deles, são a fonte primária de partes e estruturas para sistemas maiores (19).

b) Linha de produção: é o processo de transformação no qual as partes elaboradas seguem uma mesma seqüência de operação ou linha de produção, como é exemplificado na figura 2.2 parte b. As linhas de produção são formadas por máquinas específicas para cada tipo de processo, portanto, existem diferentes linhas para diferentes partes fornecidas.

São características deste sistema: não flexibilidade, longo tempo de configuração (setup), fornece lotes grandes, utiliza máquinas ferramentas específicas, grandes estoques intermediários no processos.

A automação neste tipo de fábrica tem sido bem sucedida, pois as linhas de produção permanecem com a mesma configuração por um longo período de tempo e as operações de produção são altamente repetitivas (19).

c) Tipo projeto: é dirigido à produção de produtos ou serviços, os quais são de grande dimensão ou de uma classe com um conjunto de tarefas bem definidas, como por exemplo construção de navios, casas, aviões. Neste sistema os trabalhadores,

materiais e equipamentos vão ao lugar onde vai ser elaborado o projeto, as partes utilizadas são oferecidas pelos outros sistemas de manufatura.

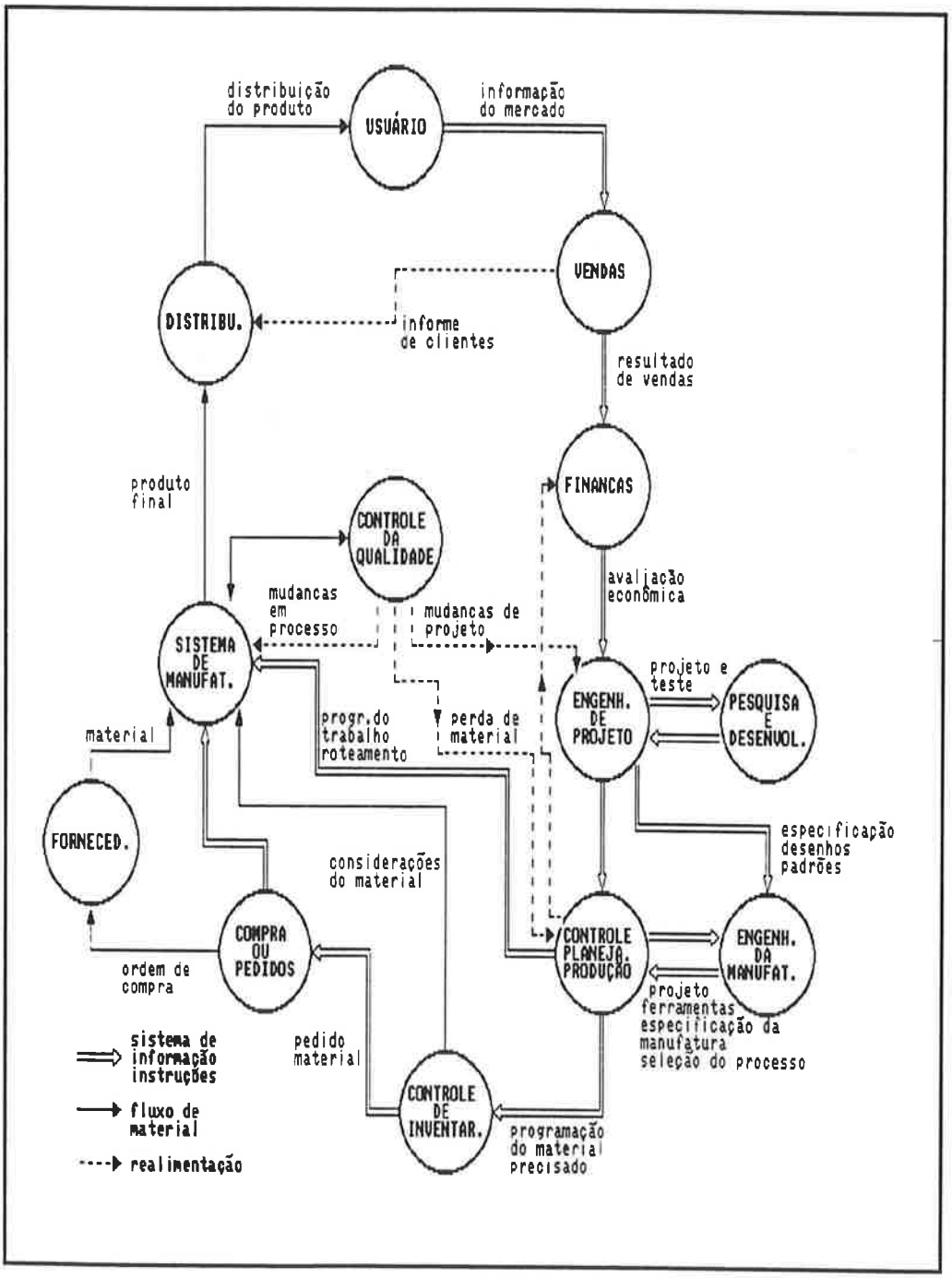

Figura 2.1 Relação entre os elementos do sistema de produção e o sistema de manufatura

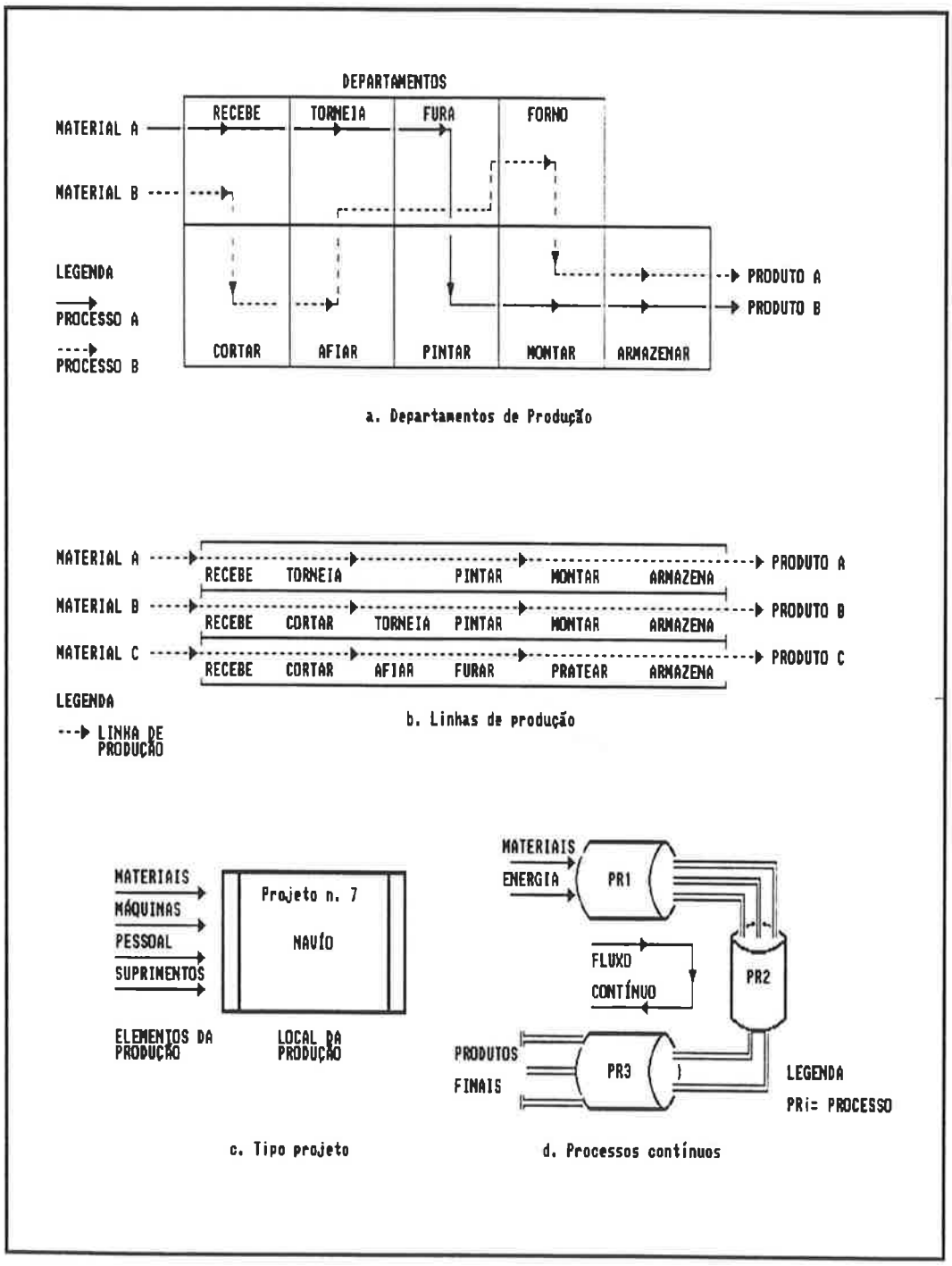

Figura 2.2 Sistemas de manufatura

Algumas de suas características são: utiliza equipamento de uso geral, tipo móvel, os tempo de configuração (setup) são variáveis e lotes de produção pequenos. Este

sistema é ilustrado na parte c da figura 2.2.

d) Sistemas contínuos: aqui os produtos fluem através de uma série de etapas em forma contínua, onde acontece a transformação. Este sistema, não tem relação com os processos discretos aqui abordados, mas representa o objetivo ideal ou a visão dos sistemas de manufatura, onde partes possam fluir como os líquidos nos sistemas contínuos.

A chave para atingir este objetivo é projetando um sistema de manufatura, de tal forma que, os lotes fornecidos sejam pequenos, porque lotes pequenos podem tornar suave o fluxo das partes (16). A parte d da figura 2.2 ilustra o sistema contínuo.

Até aqui, apresentou-se as características gerais dos sistemas de manufatura tradicionais, ressaltando os problemas que eles possuem. A seguir será apresentado o sistema de manufatura celular, como uma alternativa de solução para aumentar a produtividade e atingir a integração da manufatura.

2.1.2 Sistema de Manufatura Celular

O desenvolvimento do CIM e a utilização da técnica de produção "Just-In-Time", ou produção de uma quantidade de partes na hora certa, tem impulsionado o interesse na implementação de sistemas de manufatura celular na indústria (20). O sistema de manufatura celular implica no arranjo de máquinas e equipamentos, denominado posteriormente de célula flexível de manufatura, dedicado à produção de uma família de partes.

A idéia é agrupar as máquinas de tal forma que a distância entre elas seja mínima,

e assim permitir a fácil transferência do material dentro da célula, imitando os processos contínuos; também é bom ressaltar, que a produção é orientada a lotes pequenos, concordando com o objetivo primário do "Just-In-Time".

Os sistemas de manufatura celular são estruturas que suportam com facilidade os critérios aplicados no chão de fábrica, derivados do "Just-In-Time" e do Controle Total da Qualidade e possuem uma característica importante: são a semente para a Integração Total (16).

No item seguinte é apresentado um estudo da célula flexível, que é o elemento básico do sistema de manufatura celular.

2.1.2.1 Célula Flexível de Manufatura

O elemento essencial do sistema de manufatura celular é a célula flexível. Em torno deste conceito existem muitas definições, inclusive algumas que saem fora do escopo dos objetivos da manufatura celular, uma das definições mais adequadas é a seguinte (7), (10), (16), (17):

"A célula flexível de manufatura é a reunião de pessoas, equipamentos e processos, arranjados metodologicamente, visando a produção de uma família de partes".

A célula é constituída de equipamentos, que podem ser manuais, semi automáticos ou totalmente automáticos, sendo que os dois primeiros englobam pessoas dedicadas na operação das máquinas. A família de partes é constituída por um conjunto de partes, onde existem semelhanças entre os processos de manufatura e projeto das mesmas.

Cabe indicar que a célula é projetada de tal forma que permite a expansão da mesma, incluindo novos elementos, ou permitindo a configuração rápida para a produção de um novo membro da família, são estas características que tornam a célula flexível.

Também, que à medida em que as fábricas tendem a crescer, as células devem ter capacidade de integração e modularização, isto é, devem ser capazes de se desdobrar em formas mais gerais de automação do tipo "Sistema Flexível de Manufatura" (1), (21), (22).

O interessante da célula é que pode ser simples ou complexa, isto é, pode ser implementada manualmente, ou de maneira automática, ou em forma híbrida, isto permite, inclusive arranjar os equipamentos antigos de uma fábrica, sem necessidade de equipamentos novos (7). Segundo os equipamentos, tem-se os tipos de célula seguintes:

a) Célula manual: contém máquinas ferramentas convencionais, automáticas, como CNCs (controladores numéricos computorizados) e pessoal da produção, que previamente foram treinados para operar mais de uma máquina dentro da célula.
A figura 2.3, representa uma típica célula manual, com um arranjo fisico (lay-out) em "U", que proporciona maior flexibilidade; na célula manual o objetivo principal é melhorar a utilização dos trabalhadores, tornando-os multifuncionais, capazes de operar todas as máquinas da célula, sendo que, o número de pessoas depende do requerimento de produção da célula.

b) Célula automática: contém máquinas ferramentas programáveis e qualquer outro equipamento de automação utilizado no processo de manufatura, como controladores lógicos programáveis, robôs, veículos autônomos, um exemplo da configuração é mostrado na figura 2.4.

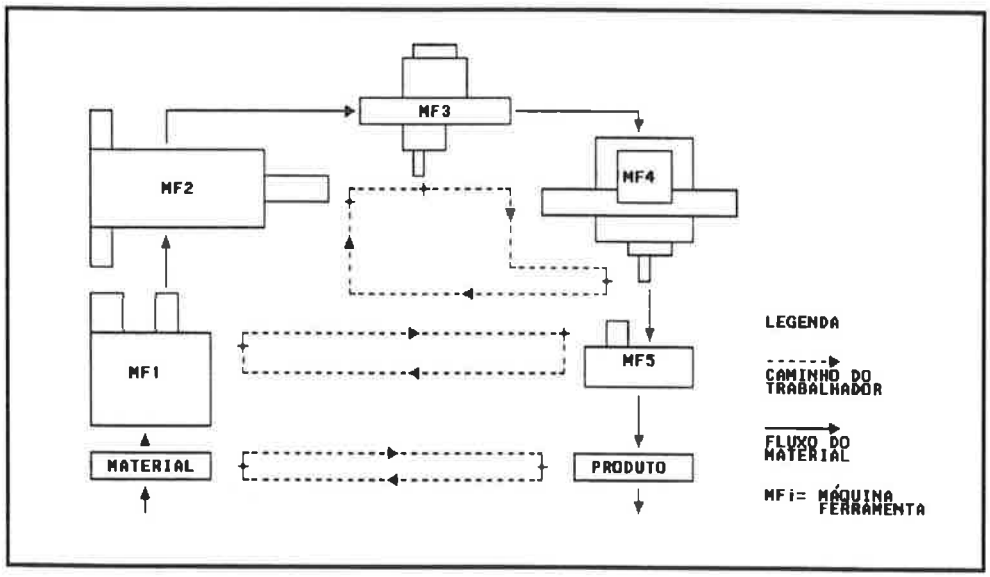

Figura 2.3 Célula de manufatura manual

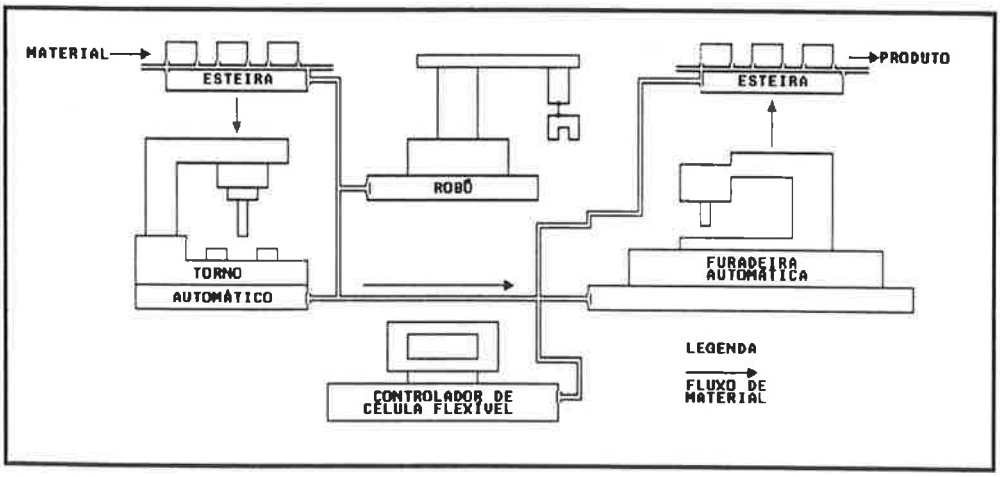

Figura 2.4 Célula de manufatura automática

Este tipo de célula vem sendo comandada pelo controlador de célula flexível, que carrega automaticamente os programas de controle para cada equipamento da célula flexível e supervisiona o fluxo automático das partes na célula flexível.

Os benefícios obtidos, após a implementação da célula flexível são os seguintes (21), (22):

- Bom início para a integração total (CIM): a implementação do sistema celular torna mais fácil a aplicação de altos graus de automação no chão de fábrica, sendo um bom começo para a implementação da integração total, inclusive as mudanças feitas no sistema de produção, na hora de implementar a célula, são aproveitadas nesse sentido;

- Ganhos na flexibilidade: as células flexíveis permitem a produção de diferentes produtos, de acordo com a demanda do mercado, pois são inerentemente flexíveis, adaptam-se a qualquer mudança, expansão ou novo pedido, com um tempo pequeno de configuração (setup);

- Redução dos tempos de configuração (setup time): a eliminação dos tempos de configuração permite a produção econômica de pequenos lotes, pois com um tempo de configuração igual a zero, o custo por parte fica independente da quantidade produzida. Os sistemas celulares são orientados na obtenção de um "setup" mínimo, e é conseguido com a determinação de uma ótima programação de manufatura (Scheduling) da família de partes, os quais possuem requisitos de produção semelhantes, o que também implica na redução do tempo total da produção;

- Homogeneidade da qualidade: a utilização de equipamentos automáticos e o controle computarizados, permitirão obter produtos de qualidade altamente homogênea;

- Permite a aplicação do "Just-In-Time": a célula flexível é um dos melhores sistemas para a aplicação das técnicas "Just-In-Time" e o controle total da qualidade, técnicas

aplicadas com bastante sucesso, pois a célula permite o fluxo suave das partes e orienta-se a produção de pequenos lotes, e fornece dados sobre o processo de produção;

- Motivação do trabalhador: a qualidade e desempenho do trabalhador cresce, pois ele está envolvido com todo o processo de manufatura, não está isolado só na operação de uma máquina;

- Redução da manipulação do material: a transferência das partes entre os elementos da célula é melhorado pela proximidade dos equipamentos, e possibilita a utilização de robôs e equipamentos automáticos de transporte com eficiência;

- Redução dos estoques intermediários no processo: a disponibilidade das partes nos elementos da célula é quase imediata, nesta característica são vários fatores que contribuem, como a redução do "setup", o espaço reduzido da célula, uma boa determinação da famílias, a aplicação do "Just-In-Time" e um bom sistema automático de transporte.

Para obter os benefícios supracitados é necessário uma boa implementação da célula, tarefa crítica, cuja realização envolve toda a estrutura da fábrica. No item seguinte são apresentados as etapas da implementação de células flexíveis e alguns comentários importantes.

2.1.2.2 Implementação da Célula Flexível

A evolução ou a implementação de um sistema de manufatura celular implica necessariamente em sérias mudanças no sistema de produção, dada a relação entre o sistema de manufatura e todos os outros elementos que compõem a produção, pelo que

se exige um planejamento sério e metodológico, inclusive com o compromisso e a dedicação de todo o pessoal da empresa (23), (24).

Basicamente a implementação de um sistema celular implica na implementação da célula flexível. As variáveis chaves que devem ser consideradas no projeto da célula são (20): número de células, dimensão da célula, número de equipamentos, composição da célula, características das partes, rota das partes, número de operações por parte, número de tipos diferentes de partes. Considerando estas variáveis, divide-se o projeto nas etapas seguintes (16), (24):

a) Formação da família: consiste na seleção da população de partes e o agrupamento das partes em famílias. Uma excelente alternativa para a determinação das famílias de partes, é a utilização da tecnologia de grupo (GT), que permite o agrupamento de partes, considerando as semelhanças que possuem na fase de projeto e no processo de manufatura, como é apresentado na figura 2.5.

   Existem diferentes métodos na tecnologia de grupo, como, o método visual, o método da codificação e outras soluções bem sucedidas, que são explicadas detalhadamente em (25).

b) Formação da célula: consiste na seleção da população de máquinas e processos e o conseqüente agrupamento destes numa área específica e com um arranjo físico (layout) adequado, dentro de uma célula, este trabalho e baseado na etapa anterior. Nesta etapa é bom ressaltar as seguintes considerações :

   - Deve existir suficiente volume de trabalho para justificar o estabelecimento da célula;

- Os processos devem ser tecnologicamente compatíveis;

- Predição da demanda por parte, a determinação de máquinas, acessórios (paletos, ferramentas) e a seleção do equipamento transportador de material (robô, e veículos autônomos) necessários por célula. Estes são necessários na equalização do trabalho dentro da célula;

- Seleção adequada do arranjo físico (Lay-out) da célula; é preferível um arranjo físico na forma de "U", porque minimiza os raios de ação (8) (24).

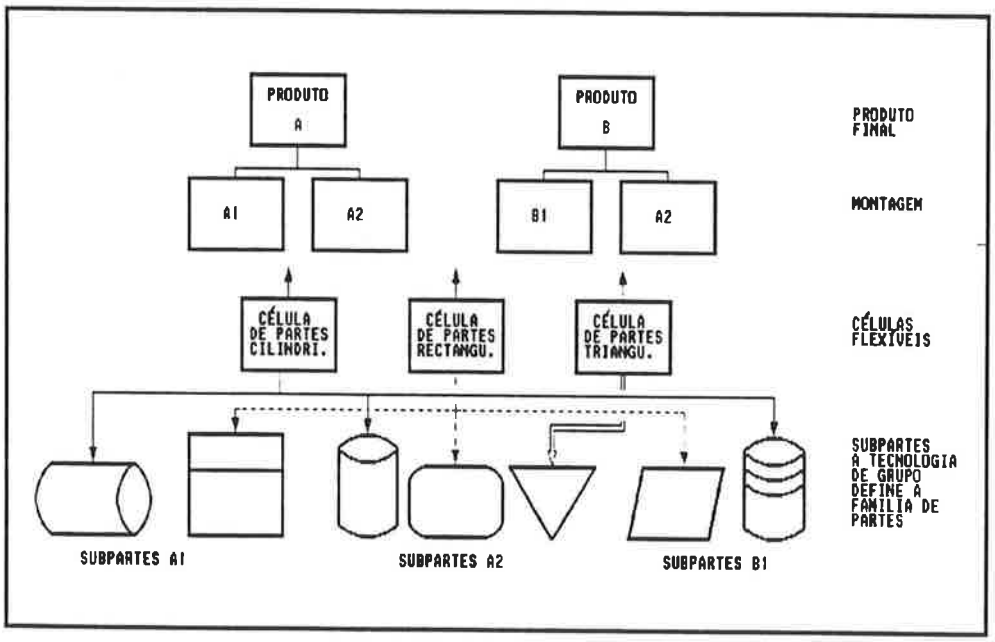

Figura 2.5 Exemplo do uso da tecnologia de grupo

Nesta etapa, também aproveitam-se as vantagens da tecnologia de grupo, em especial na formação da célula, e geralmente os diferentes métodos propostos são utilizados nestas duas etapas. Uma informação mais específica pode ser encontrada em (20), (25), (26), (27).

Em conclusão, os sistemas de manufatura celular comparados com os sistemas tradicionais, proporcionam uma série de benefícios importantes e aceitam mais rapidamente a aplicação de altos graus de automação, pelo que são considerados como uma boa alternativa para aumentar a capacidade de competência das empresas.

Outra conclusão importante está na implementação do sistema de manufatura celular, que representa uma tarefa bastante complexa, mas cujos resultados serão aproveitados no futuro, quando se decida implementar a Integração Total (CIM). A seguir, serão apresentados as principais razões para que se implante o CIM, que representa o embasamento da fábrica moderna.

2.2 Integração Total da Manufatura (CIM)

Enfrentando um mercado cada vez mais complexo, muitas fábricas estão reconhecendo que a forma de manter-se competitivas vem sendo expressado em termos de custos, qualidade e resposta rápida dentro do mercado.

Num intento de melhorar a produtividade e, portanto, incrementar a capacidade competitiva, muitas fábricas vêm adotando gradualmente o CIM (Computer Integrated Manufacturing) ou Integração Total da Manufatura, conceito que se torna como o maior aporte na consagração do sistema de manufatura moderno (3).

Deve-se enfatizar que o CIM ou Integração Total da Manufatura é um conceito, uma filosofia, cuja implementação implica na aplicação de métodos e padrões e de um acertado planejamento, tendo como objetivo final, o fluxo contínuo da informação entre todos os elementos da organização (28), (29).

Atualmente, existem problemas na implementação, especialmente nas áreas

tecnológicas, gerenciais e educacionais (3), (30), mas seu crescimento está sendo assegurado com a evolução de novas metodologias de implementação, com o surgimento de modelos e arquiteturas, com a aparição de métodos gerenciais adequados e, especialmente, pela tendência de padronização em muitos aspectos tecnológicos relacionados com a integração total.

2.2.1 Caracterização do CIM

Para entender os elementos que compõem o CIM, será analisado o CIM por dois pontos de vista, o funcional e o organizacional (31).

Do ponto de vista funcional, as funções são agrupadas em módulos, cada um destes módulos possuem aspectos computacionais, que posteriormente são aproveitados na integração total. Considerando estes aspectos computacionais, as funções representativas são as seguintes (31), (32):

a) Funções referentes ao projeto: sendo o CAD (Computer Aided Design) ou projeto assistido por computador o elemento principal.

   Este termo consiste de auxílios computacionais em todos os aspectos, hardware (como por exemplo: terminais gráficos e estações de trabalho) e software (como por exemplo: manipuladores de imagens, simuladores) e envolvem todas as funções referentes ao projeto de partes, componentes, ferramentas e acessórios.

b) Funções referentes ao planejamento do processo: as formas de roteamento das partes na célula, tipos de operação e lista de equipamentos, são exemplos típicos deste módulo, sendo que os aspectos computacionais agrupam-se em torno do CAPP (Computer Aided Process Planning).

c) Funções referentes à manufatura: englobam funções como, programação dos equipamentos e configuração do controlador de célula (32). Nesta função, o apoio computacional é o CAM (Computer Aided Manufacturing) ou a manufatura assistida por computador.

d) Funções referentes à qualidade: basicamente correspondem às funções tradicionais de inspeção das partes, com o objetivo de manter a qualidade necessária. Todos os aspectos computacionais relacionados, estão envolvidos com o termo, CAQ (Computer Assurance Quality) ou a garantia da qualidade assistida por computador.

e) Funções referentes ao armazenamento e disponibilidade dos materiais: estas funções são exercidas utilizando os equipamentos automáticos adequados, onde os materiais são armazenados ou colocados nas máquinas, eficiente e rapidamente.

Do ponto de vista organizacional, existe a necessidade de uma estrutura que suporte todas as funções anteriores, na atualidade existem muitas propostas (14), sendo o modelo da NBS (National Bureu of Standard) uma das estruturas mais aceitas, pois representa a fábrica como um sistema hierárquico modular, possuindo cada nível um conjunto de funções (19).

A figura 2.6 representa o modelo aplicado na manufatura, e que é a proposta apresentada em (13). A seguir é mostrado a definição dos níveis:

a) Coordenação: representa o nível superior na estrutura hierárquica e é denominado de sistema de informação gerencial, este nível possui três áreas funcionais: o gerenciamento da informação, o gerenciamento da produção e a engenharia de manufatura.

O primeiro é composto pelas funções de gerenciamento comercial e administrativo, o segundo, corresponde às tarefas referentes ao planejamento e coordenação da produção e o terceiro agrupa as funções do CAD e do planejamento do processo.

b) Controle: este nível é responsável pela configuração dos controladores de célula flexível e pela supervisão e coordenação do funcionamento dos equipamentos correspondentes à célula flexível.

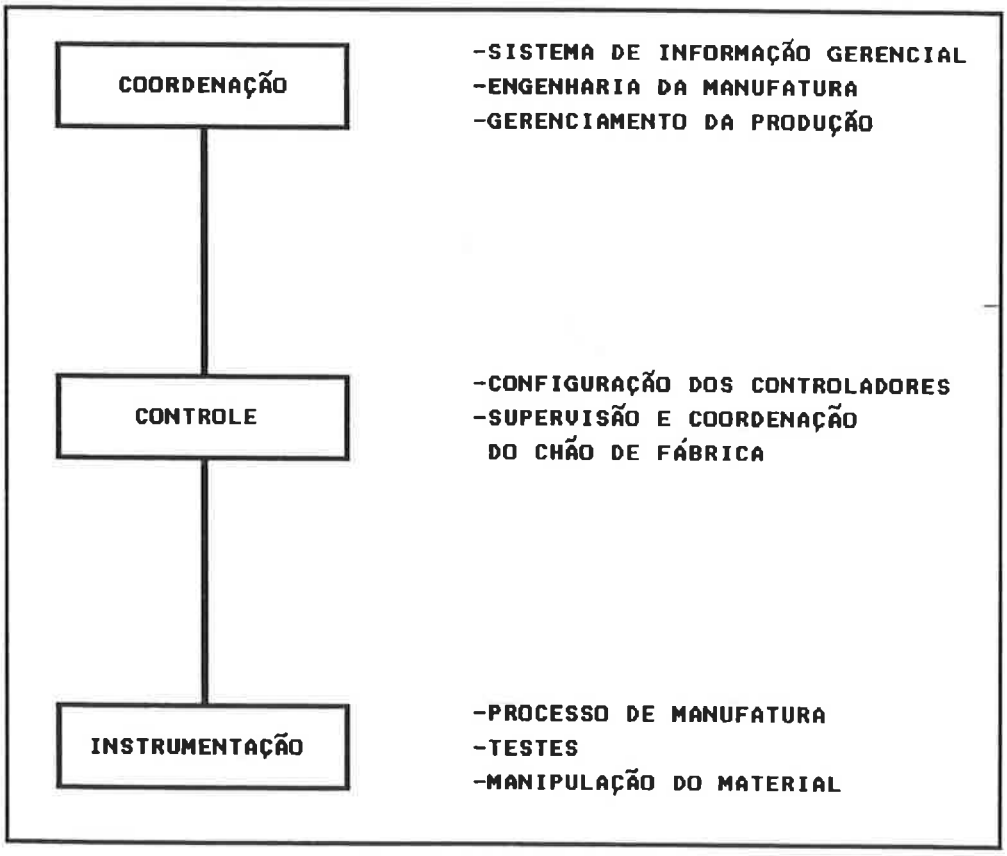

Figura 2.6 Modelo hierárquico da manufatura

c) Instrumentação: nível correspondente às funções referentes ao processo de manufatura (chão de fábrica) e está constituído pelos equipamentos de automação como robôs, máquinas ferramentas, veículos transportadores ou qualquer

equipamento inteligente utilizado na produção das partes, também estão incluídos os equipamentos de teste.

Cabe ressaltar que cada nível do modelo anterior possui equipamentos de diferentes naturezas, como por exemplo no nível superior estão os computadores de uso geral, no nível do controle o controlador de célula flexível e no nível inferior os equipamentos programáveis do chão de fábrica.

Como se pode observar são equipamentos totalmente heterogêneos, que tornam a conectividade entre níveis bastante complicada, inclusive dentro de cada nível, como no nível de instrumentação que possui equipamentos de automação totalmente heterogêneos, onde conseguir um alto grau de conectividade é quase impossível.

Os problemas anteriores evitam que o fluxo da informação através de todo o modelo hierárquico seja rápido e possa atender todos os requisitos de acesso nos diferentes níveis, portanto é necessária uma estrutura de interconexão de comunicação, cujo objetivo final será conseguir a conectividade de todos os elementos da fábrica. Esta estrutura de comunicação deve ser hierárquica e deve ser composta de sub-redes de comunicação de naturezas diferentes, pois atendem diferentes requisitos de especificação em cada nível.

Finalmente, fechando a caracterização da Integração Total (CIM), uma implementação adequada do CIM traz os benefícios seguintes (20):

- Redução dos custos de produção: considerando os custos laborais, o incremento da utilização das máquinas, a redução do material residual e a redução dos estoques;

- Incremento da competitividade: que corresponde às características de rápido

desenvolvimento do produto, de adaptação às mudanças e de flexibilidade no volume de produção;

- Melhora da qualidade: pela utilização de processos consistentes, rápido diagnóstico, redução dos erros humanos e utilização de métodos de inspeção automática.

Foram apresentadas os principais pontos para a implantação da Integração Total (CIM), os benefícios que podem obter-se com a implementação e foi ressaltado o problema de conectividade dos elementos que existem na sua estrutura. A seguir, mostra-se qual é a relação entre este conceito, a célula flexível e o controlador de célula flexível, tema da dissertação.

2.2.2 A Implementação do CIM, a Célula e o Controlador

A adoção da Integração Total (CIM) exige os seguintes requisitos: que a fase de projeto seja efetuada utilizando uma metodologia superior-inferior, isto é, começando o projeto no nível de coordenação, do modelo descrito no item anterior, e terminando no nível de instrumentação; e que a fase de implantação utilize uma metodologia inferior-superior, esta última, enxerga a integração como uma extensão da aplicação da automação, realizada em cima do chão de fábrica, visando posteriormente os níveis superiores (33).

Na implementação do CIM, o nível do chão de fábrica é considerado peça chave, e é um dos níveis com maiores problemas de integração (34). Existem dois assuntos a respeito do chão de fábrica que estão relacionados fortemente com a implementação da Integração Total e que a seguir são comentados.

Um primeiro assunto, que facilitará a implementação do CIM, é a implementação

da célula flexível (35). Uma evolução acertada na implementação do CIM, pode ser da forma seguinte, primeiro formar as células, aplicar automação e obter células automáticas, em seguida interligá-las através de comunicação de dados e criar um Sistema Flexível de Manufatura e evoluir até atingir os outros níveis superiores de automação (36).

O outro assunto importante é a integração dos equipamentos de automação heterogêneos do chão de fábrica e a integração deste nível com o nível de coordenação ou sistema de informação gerencial, ambientes totalmente diferentes da fábrica. O controlador de célula possui estas duas funções, permite a intercomunicação dos elementos da célula, entrega informação do processo em andamento e recebe programas para ser carregado nos equipamentos desde os níveis superiores da estrutura hierárquica.

Estes assuntos comentados nos parágrafos anteriores, ressaltam a forte relação que existe entre o CIM e a célula flexível e, especialmente, com o controlador de célula flexível, mostrando a importância que este equipamento possui na automação da fábrica.

No item seguinte são descritos os elementos que são governados pelo controlador de célula flexível.

2.3 O Controlador e os Equipamentos da Célula

O controlador de célula, como já foi apresentado, é o equipamento representativo do nível de controle dentro da integração da manufatura, sendo definido da maneira seguinte (23), (37):

"O controlador é o equipamento encarregado de coordenar, controlar e supervisionar as atividades dos elementos da célula flexível utilizados na produção de

uma família de partes. São objetivos importantes do controlador a integração dos elementos do chão de fábrica e a interconexão deste nível com o sistema de informação gerencial".

O principal e mais crítico problema encontrado no chão da fábrica é que os equipamentos de automação possuem seu próprio protocolo de comunicação, sistema operacional, linguagem de programação, controlador e, além disso, são de diferentes fornecedores, muitas vezes dedicados ao fornecimento só desse equipamento, e sem visar que podem ser integrados em outras células flexíveis ou sistemas (34).

Esta falta de interoperabilidade originou, também, problemas na definição da arquitetura do controlador, que seguiu um padrão, isto é, de projetar controladores particulares para cada célula flexível. Mas com utilização das arquiteturas abertas, como é a proposta desta dissertação, o controlador está tendendo a adotar uma arquitetura flexível, e deixando as propostas anteriores, como parte da evolução do controlador (37).

Os equipamentos de automação pertencentes à célula flexível que são coordenados pelo controlador de célula, são: controladores numéricos computarizados, robôs, controladores lógicos programáveis, veículos autônomos, sistema de visão e outros equipamentos de automação, indicando que a complexidade do processo determina a complexidade da célula.

A seguir, serão apresentadas as características dos elementos típicos da célula, com o objetivo de ressaltar a natureza heterogênea dos equipamentos de automação.

a) O Controlador Numérico Computarizado (CNC).- o propósito do CNC é de controlar, coordenar e monitorar as operações das máquinas ferramentas. O CNC

pode ser caracterizado pelos três elementos seguintes:

- Hardware: composto por um processador central, encarregado do controle das funções do sistema; as interfaces de entrada e saída, para manipular sensores e atuadores; e o módulo de comunicação, para receber e enviar informação ao controlador de célula;

- Software: constituído pelo software de controle e pelos aplicativos. O software de controle contém as funções do sistema operacional e os módulos de controle e de comunicação, enquanto que os aplicativos contém o interpretador, encarregado de converter os códigos de programação de CNC, como o código "G", em códigos objetos, o pós-processador encarregado de trasladar os arquivos produzidos no CAD em códigos de CNC, e que foram escritos numa linguagem APT (Automatically Programmed Tool), e os módulos de gráficos e algumas funções especiais (38).

- Mecânica: correspondente à estrutura física do CNC, como os eixos, conjunto de ferramentas e todas as características mecânicas que possuem as máquinas ferramentas tradicionais (39).

A informação da produção que possui este equipamento é muito importante, por isso que é necessário incluir nas novas gerações de CNC, módulos de comunicação que manipulem protocolos padrões, e nas antigas gerações implementar essa função pela adição de módulos ou pela substituição do controlador, por outro mais flexível (40).

b) Robô: uma definição adequada de robô é apresentada em (38), "O robô é um manipulador multifuncional e reprogramável projetado para movimentar materiais,

partes, ferramentas ou dispositivos especializados, através da programação de movimentos variáveis para o desempenho de uma variedade de tarefas".

O robô pela complexidade de sua constituição é estudado por uma área multi-disciplinar como é a Robótica. Para caracterizar o robô considera-se os seguintes aspectos (42), (43):

- Controlador: constituído pelo coordenador de braço, sistema de visão, sistema de sensores, processador coordenador, interface com o operador, comunicação exterior e monitoração da operação; estas funções, determinam a complexidade da arquitetura do controlador (44).

- Programação: geralmente os robôs são programados de duas formas: programação "em linha" onde o programador utiliza um dispositivo especial (teclado específico com um pequeno display), que permite a criação de programas e a movimentação manual do braço do robô; neste caso as trajetórias do braço são armazenadas e executadas posteriormente.

  Na programação "fora de linha" o programador cria programas num ambiente que possui uma linguagem específica de robô, um simulador e um processador gráfico apropriado, e realiza esta tarefa sem interromper as atividades normais do robô na célula, este método é o mais adequado na atualidade (45), (46).

- Mecânica: o primeiro elemento é o manipulador, que consiste de uma base e o braço do robô, incluindo a fonte de movimento, que pode ser elétrica, hidráulica, ou pneumática; a função do manipulador é proporcionar uma série de movimentos considerando certos graus de liberdade. Outro elemento é a "mão" (end-effectors), com diferentes níveis de manipulação para diferentes tipos de

trabalho, possui também sensores que proporcionam informação adicional ao controlador;

c) Controlador lógico programável (CLP): é um equipamento computadorizado muito utilizado na automação de processos, especialmente porque executa as funções de intertravamento e seqüêncialização através dos módulos análogo-digital de entrada e saída. Para a caracterização do CLP serão apresentados os seguintes aspectos:

- Hardware: composto por um processador central, que executa as funções lógicas e aritméticas que implementam os intertravamentos e a seqüêncialização; a memória que armazena os programas; os módulos de entrada e saída que são as interfaces com as variáveis dos processo externos; e os dispositivos de programação, alguns CLP possuem uma tela para programação, outros são programados através de um teclado especializado, que possui teclas com funções específicas.

- Software: a programação dos CLP é realizada utilizando linguagens de programação específicas para CLP, como por exemplo, linguagens de contatos (relés), linguagens baseadas na lógica Booleana, ou através de uma linguagem de máquina.

d) Veículo autônomo: é um tipo de veículo automático que é utilizado para transportar material ou partes na fábrica e pode ser programado para percorrer um caminho determinado. Principalmente é aplicado em sistemas flexíveis de manufatura, fábricas de carros e em manipulação de material radioativo. As características destes veículos são apresentadas a seguir (42):

- Classes de veículos: existem dois tipos de veículos autônomos: os guiados por

caminhos externos e os guiados por caminhos programados. O veículo guiado por caminhos externos segue um caminho que é implementado por um arranjo físico de cabos com emissores de sinal, ou um caminho que é pintado no chão com tintas fosforescentes.

O sinal elétrico, no caso dos cabos, é detectado pelo veículo através de sensores de indução magnética e no caso dos caminhos pintados existem fotosensores que detectam o caminho.

No veículo de caminho programado o percurso é armazenado na memória do controlador e utilizado como padrão de comparação com o caminho que esta sendo seguido, utilizando qualquer das técnicas explicadas em (42).

- Hardware: composto do módulo de comunicação, que permite a comunicação com o computador supervisor de veículos autônomos, utilizando um sinal de FM, possui a memória onde é armazenado o caminho padrão; do módulo de controle, encarregado de receber o sinal dos sensores instalados nas rodas do veículo, e do sinal do equipamento de visão, executa o algoritmo de controle para guiar o veículo.

Os três aspectos analisados dos diferentes equipamentos de automação, hardware, software e mecânicas foram utilizados para ressaltar a heterogeneidade do chão de fábrica.

Por exemplo, no hardware pode ver-se que não existe um módulo de comunicação externa dedicado, no máximo uma porta serial, e no software existem linguagens diferentes e específicos a cada equipamento sem um formato padrão, e tudo isto acontece porque estes equipamentos foram projetados sem visar a participação num

sistema flexível e integrado.

Na atualidade a função de intercomunicação esta sendo considerada fortemente nos projetos desses equipamentos, utilizando padrões, como o MAP, que ajudarão na integração do chão de fábrica (11).

Neste capítulo foi apresentado o embasamento teórico relacionado com o controlador de célula flexível, no seguinte capítulo será apresentada a primeira fase do projeto, a especificação funcional do controlador de célula flexível objeto deste trabalho.

3 ESPECIFICAÇÃO DO CONTROLADOR

Os projetos de sistemas de tempo real são tarefas complexas, devido à existência de características diferentes dos sistemas sem restrições no tempo (47).

Para o desenvolvimento de projetos desta natureza são necessárias metodologias sistemáticas da forma "superior-inferior" (Top-Down), que permitam aos projetistas dividir a complexidade do problema em partes ou módulos mais simples, facilitando o desenvolvimento do projeto.

Das diferentes metodologias de projetos propostas, todas elas concordam que a fase mais crítica é a especificação funcional ou concepção do sistema (48). Nesta fase, determina-se o comportamento funcional do sistema.

A falta de uma boa definição, nesta fase, originará muitos problemas nas fases seguintes do projeto, conseguindo-se muitas vezes um bom produto, mas com uma série de restrições na aplicação final (49).

Atualmente, uma coisa parecida está acontecendo com a implementação dos controladores de célula flexível, pois estão sendo implementados utilizando equipamentos de prateleira, como microcomputadores e controladores lógicos programáveis (CLP).

O problema destas propostas é que as arquiteturas não foram definidas tomando como base a especificação funcional do controlador, como acontece em todo projeto, mas, foram os equipamentos que definiram as arquiteturas (50).

O resultado desta forma de definição do controlador foi obter um controlador

inflexível, pois não se adapta a uma nova célula ou aceita o crescimento da célula.

Para evitar os problemas explicados nos parágrafos anteriores, este capítulo tem como objetivo principal a determinação clara e precisa do comportamento funcional do controlador de célula, visando ser a base da definição da arquitetura.

A estrutura desta especificação funcional, segue a adaptação do padrão de especificação para projetos de software descrito em (51) e estará constituída pelas etapas seguintes: análise de sistema, especificação funcional, características gerais, interfaces homem-máquina e descrição operacional.

3.1 Análise do Sistema

A finalidade desta etapa é de reconhecer os fatores gerais que impõe o sistema de automação da manufatura e que influem no comportamento funcional do controlador.

Para efeitos desta análise, considera-se o sistema aberto de automação da manufatura, baseado no modelo hierárquico definido no item 2.2.1 do capítulo anterior.

Deste modelo, são tomados em conta os seguintes aspectos do sistema: os elementos do sistema, a interconexão dos equipamentos e as características gerais do sistema.

3.1.1 Elementos do Sistema

Na figura 3.1 é mostrado um sistema aberto de automação aplicado na manufatura, com os três níveis, coordenação, controle e instrumentação. A seguir, são apresentados os equipamentos que podem ser utilizados para implementar os diferentes

níveis do sistema:

a) Nível de coordenação: Geralmente, são utilizados os computadores de propósito geral, e os equipamentos que implementam os sistemas de auxilio como o CAD, CAM, CAPP e CAQC.

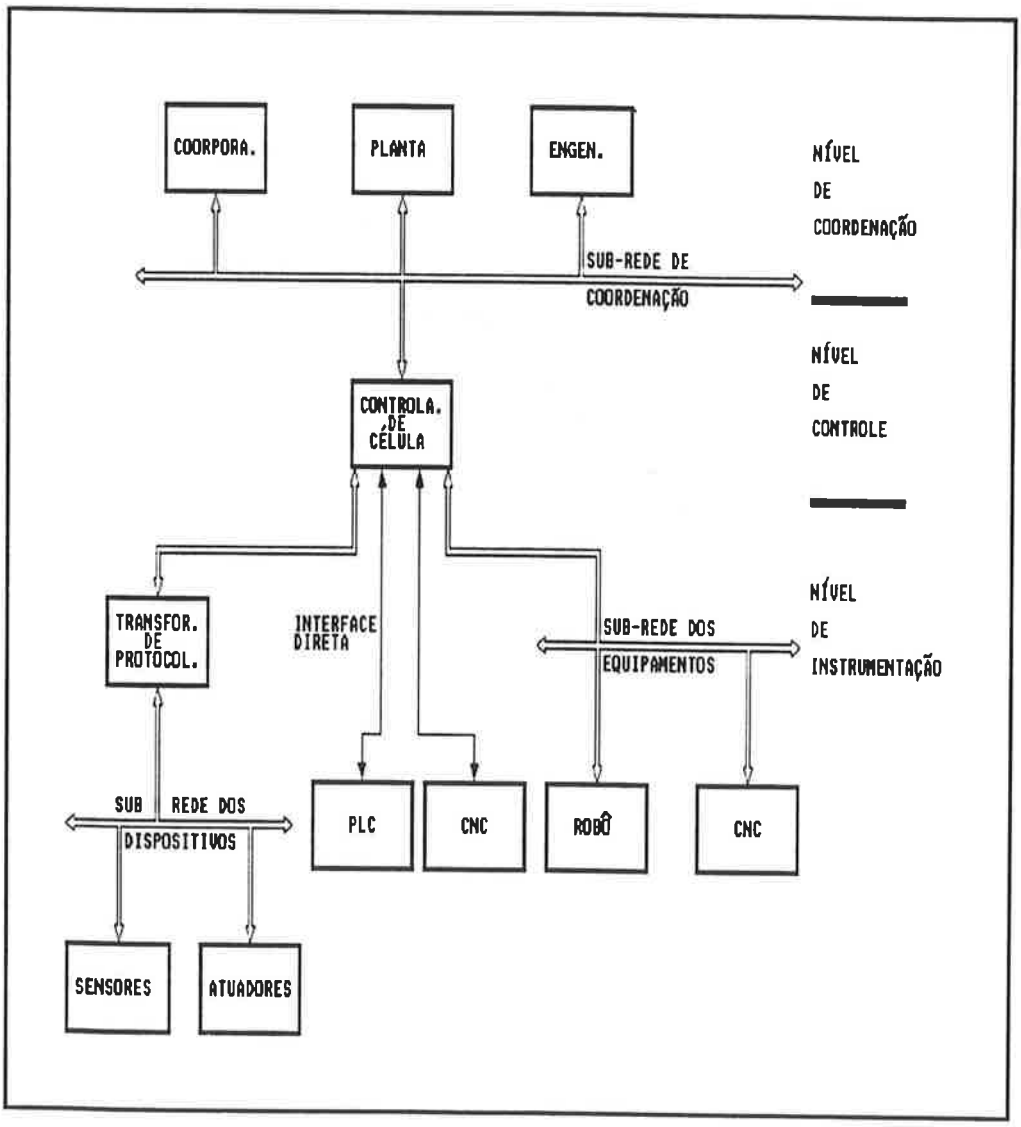

Figura 3.1 Sistema aberto de automação aplicado na manufatura

b) Nível de controle: o equipamento que implementa as funções deste nível é o controlador de célula flexível.

c) Nível de instrumentação: os elementos que implementam as funções deste nível são: os dispositivos de entrada e saída, como os sensores e atuadores; e os equipamentos de automação, como os controladores lógicos programáveis, as máquinas de controle numérico programável, os robôs, os veículos autônomos, os aparelhos de medição, ou algum outro controlador específico.

3.1.2 Interconexão dos Equipamentos

Os níveis do sistema de automação da manufatura possuem diferentes funções e equipamentos, o que dificulta o fluxo de informação na fábrica como um todo. Para conseguir a interconexão dos equipamentos heterogêneos, em cada nível utiliza-se sub-redes de comunicação, com protocolos e interfaces apropriadas.

Assim, no nível de coordenação os equipamentos comunicam-se através de uma sub-rede principal, denominada de sub-rede de coordenação.

No nível de controle, a comunicação entre os controladores de célula flexível é realizada através da sub-rede de coordenação ou da sub-rede de instrumentação.

Os equipamentos de automação do nível de instrumentação estão ligados à sub-rede de instrumentação, que pode ser de duas classes, a sub-rede dos dispositivos e a sub-rede dos equipamentos.

A sub-rede de instrumentação dos dispositivos, denominada também de Via de Campo, interliga os sensores e atuadores com capacidade de processamento e os

equipamentos de aquisição, que coletam dados de dispositivos sem capacidade de processamento utilizando interfaces físicas apropriadas.

A sub-rede de instrumentação dos equipamentos interliga os equipamentos inteligentes de automação do chão de fábrica, que são aqueles que possuem capacidade de processamento e controle próprios.

Para a interconexão entre sub-redes de diferentes níveis utilizam-se, quando é necessário, equipamentos específicos cujas funções são as de adaptar os protocolos de comunicação (52).

3.1.3 Características do Sistema

As características que envolvem os sistemas de automação da manufatura são refletidas diretamente no funcionamento do controlador. As principais são as seguintes (53):

a) Natureza distribuída: a execução das funções é geograficamente distribuída pelos diferentes setores da fábrica (54).

b) Estrutura de controle hierarquizada: a estrutura hierárquica tem sido firmemente estabelecida como um bom método de controle para a automação da manufatura, porque proporciona grandes vantagens em três aspectos importantes (14): flexibilidade, expansibilidade e confiabilidade.

c) Conectividade: são consideradas diferentes sub-redes para cada nível para manipular uma ampla faixa de requisitos de comunicação nos diferentes níveis do sistema de automação.

d) Eventos ordenados: os processos, num sistema de manufatura são em geral constituídos pela execução em seqüências lógicas e ordenadas de um conjunto de eventos.

e) Consistência dos bancos de dados: os níveis hierárquicos podem possuir bancos de dados distintos cuja informação deve ser consistente.

f) Ambiente de operação desfavorável: as operações num sistema de manufatura são fortemente influenciados pelas condições externas, como por exemplo, interferências magnéticas ou outros agentes externos de natureza aleatórios e transitórios.

Os aspectos apresentados neste item aliadas com as apresentadas em 3.1.1 e 3.1.2 permitem obter algumas conclusões, com respeito à participação do controlador no sistema de automação.

Um primeiro ponto que pode-se concluir, é que o controlador tem por objetivo implementar todas as funções do nível de controle no modelo hierárquico, isto é, deve supervisar, coordenar, controlar as células flexíveis.

Outro ponto importante é que o controlador deve ter como objetivo permitir obter dados do nível inferior e enviar-los para o nível superior e vice-versa, isto significa que deve ligar este dois níveis e ajudar no fluxo contínuo da informação em toda a fábrica.

Esta análise permite ver que o controlador possui funções correspondentes ao processamento de tempo real, com restrições no tempo e funções do processamento

transacional, onde são requeridos maior processamento computacional.

Terceiro, dada a existência de sub-redes nos diferentes níveis do sistema de automação, significa que o controlador deve possuir mecanismos adequados para poder participar como elemento deste conjunto de sub-redes.

Finalmente, as características do sistema de automação impõem considerações na funcionalidade do controlador de célula, como por exemplo, a distribuição geográfica, que obriga ao controlador a ter capacidade de comunicação com outros controladores, ou a consistência de dados, que coloca uma característica importante na gerência do banco de dados do controlador, assim também como, o ambiente desfavorável que estabelece que o controlador deve possuir mecanismos para evitar a interferência externa.

As conclusões anteriores serão considerações nas seguintes etapas da concepção funcional do controlador de célula. A seguir será apresentado a etapa da especificação funcional onde serão definidas as funções do controlador de célula.

3.2 Especificação Funcional

A especificação funcional tem por objetivo apresentar a definição das funções do controlador de célula flexível, de acordo a uma estrutura baseada em unidades funcionais, onde cada unidade funcional agrupa um conjunto de funções do controlador de célula.

Nos itens seguintes, será definido o termo perfil de aplicação do controlador, com a idéia de definir primeiro as funções do controlador de célula mínimo, que será base de um controlador mais complexo, para depois apresentar as funções do

controlador de célula.

3.2.1 Perfil de Aplicação

O perfil de aplicação do controlador de célula flexível corresponde à configuração das funções de acordo a complexidade da célula flexível (37). Existe porém um perfil de aplicação mínimo que servirá de base para a definição das funções do controlador de célula.

Com respeito ao perfil mínimo, a National Electrical Manufactures Association (NEMA) tem publicado um documento onde são definidas as funções mínimas que deve possuir todo controlador e que pode ser chamado de controlador básico ou perfil mínimo. Esta proposta define as seguintes funções mínimas (55):

a) Possuir comunicação com um operador local ou com os equipamentos do nível de coordenação, ou ambos, para poder apresentar os resultados do processo da manufatura.

b) Ter capacidade de coordenação, controle e supervisão, sob no mínimo um equipamento independente do chão de fábrica.

c) Possuir uma unidade de armazenamento local e mecanismos de acesso no banco de dados local.

Os conceitos de perfil de aplicação e unidade funcionais, apresentado a seguir, têm por objetivo, centralizar a definição do controlador e ajudar nos projetos e implementações de controladores.

3.2.2 Unidades Funcionais

As unidades funcionais são estruturas que agrupam um conjunto de funções do controlador de célula flexível, estas unidades são criadas de acordo com a relação das funções dos níveis do modelo hierárquico, portanto, tem-se unidades referentes ao nível de coordenação, ao nível de controle e ao nível de instrumentação (56), (28). A figura 3.2 apresenta as principais unidades funcionais relacionadas com o controlador, no contexto hierárquico do sistema de automação.

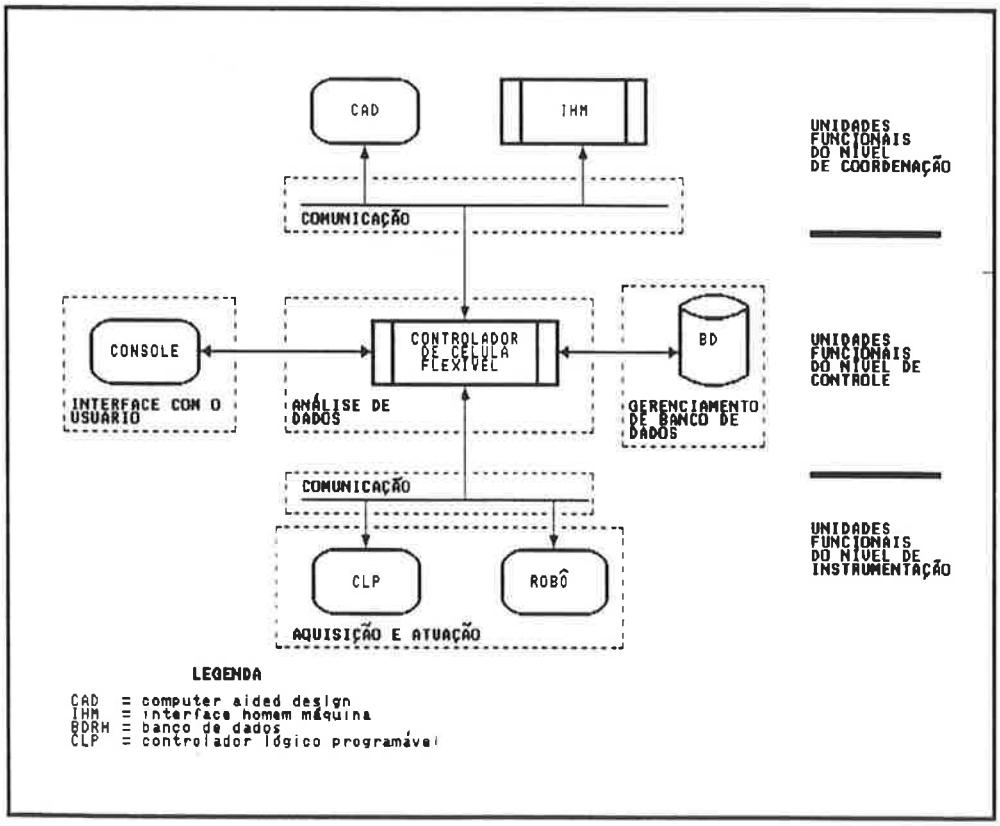

Figura 3.2 Níveis hierárquicos das funções do controlador

Dessa forma, as unidades funcionais de comunicação envolvem as funções de comunicação com as sub-redes do nível de coordenação e do nível de instrumentação.

As unidades funcionais de análise de dados, gerenciamento de banco de dados e interface com usuário estão relacionadas com as funções do nível de controle; sendo que a unidade de análise é dedicada ao processamento de dados e à configuração do controlador; a unidade de gerenciamento realiza a administração do banco de dados; e a unidade de interface é responsável pelo gerenciamento da interface homem-máquina.

Finalmente, a unidade de aquisição e atuação, referente ao nível de instrumentação, envolve as funções de entrada e saída com os equipamentos deste nível.

3.2.2.1 Unidade Funcional de Aquisição e Atuação

É a unidade funcional encarregada da coleta e envio de dados digitais e analógicos e de mensagens, correspondentes às variáveis e aos estados do processo. Os elementos que fornecem e recebem dados são os dispositivos de entrada e saída, como sensores e atuadores e os equipamentos de automação do chão de fábrica.

a) Requisitos necessários: os módulos de aquisição e atuação requerem interfaces adequadas, que devem manter os requisitos de tensão e corrente mostrados na tabela 3.1 (57).

   Um requisito importante, que também deve ser considerado, é o tempo de resposta dos dispositivos e dos equipamentos. A tabela 3.2 apresenta um resumo dos tempos de típicos (58) (59).

b) Operação: o controlador deve conter dois modos para a aquisição e atuação, o modo varredura e o modo interrupção por evento.

   No modo varredura, os pontos de amostragem são coletados numa velocidade de

varredura determinada pelo Intervalo de Varredura Básica (IVB). Este parâmetro deve ser configurado pelo operador.

	INTERFACE DE ENTRADA	INTERFACE DE SAÍDA
ANALÓGICO	4-20 ma 0 a + 1v DC 0 a + 5v DC 0 a + 10v DC 1 a + 5v DC + 5v DC +- 10v DC	4-20 ma 10-50 ma 0 a + 5v DC 0 a + 10v DC +- 2.5v DC + 5v DC +- 10v DC
DIGITAL	24v AC-DC 48v AC-DC 120v AC-DC 230v AC-DC Nível TTL 0 v Entrada isolada	12-48v AC 120v AC 230v AC 12-48v DC 120v DC 230v DC Nível TTL

Tabela 3.1 Requisitos de tensão e corrente

TIPOS	No de BITS	TEMPO DE RESPOSTA
Valores Analógicos de variação rápida	12	aprox. 1 ms
Valores Analógicos de variação lenta	12	aprox. 1 s
Entradas digital tipo evento	1	aprox. 1 ms
Entrada digital de estados	1	20-100 ms
Atuadores ligado-desligado	1	20 msg - 1 s
Contadores	16	aprox. 1 ms
Mensagens	Pacotes de bits-bytes	tipico 10 ms

Tabela 3.2 Tempos de resposta típicos

No modo interrupção-evento, por exemplo a aquisição é feita quando um certo evento ao acontecer avisa o controlador que deve iniciar uma leitura de dados. O sinal de interrupção pode ser gerado por software, por algum dispositivo ou manualmente pelo operador.

Todos os dados coletados são enviados para o banco de dados do controlador para serem utilizados na execução das outras funções próprias do controlador de célula.

3.2.2.2 Unidade Funcional de Análise de Dados

A unidade funcional de análise de dados é que determina a capacidade de controle, supervisão e coordenação do controlador. As funções, aqui especificadas, utilizam os dados que foram coletados pelo nível de instrumentação e são apresentadas a seguir:

a) Função de controle: nesta função são considerados as funções de intertravamento e seqüencialização. As operações desta natureza exigem uma ordem de execução bem determinada, pois são utilizadas na proteção dos equipamentos do chão de fábrica ou em outras atividades que exigem este tipo de operação.

O usuário utiliza as telas de configuração para ingressar os dados seguintes: pontos de controle (set-points), tempos de varredura ou intervalos de varredura básica (IVB).

Com estes dados os algoritmos de intertravamento e seqüencialização são executados e geram como resultados alarmes, relatórios de operação, disparos de dispositivos, travamentos de equipamentos, ou algum outro específico que venha a ser necessário.

b) Função de supervisão: Existem duas funções básicas de supervisão que se devem considerar: a análise estatística de processos e a análise de seguimento.

Na análise estatística de processos ou controle estatístico de processos (CEP) são executados operações estatísticas como a média, o desvio padrão, a correlação, ou outra operação necessária para o controle da qualidade. Aqui são utilizados os dados do banco de dados histórico, que mantém os valores das variáveis de todo o processo.

A análise de seguimento no tempo consiste na apresentação das variáveis do processo em função do tempo, proporcionando um indicador do desempenho de uma determinada variável, operação ou elemento da manufatura.

Dependendo do banco de dados utilizado, serão considerados duas classes de seguimento: seguimento em tempo real e o seguimento histórico, considerando que o banco de dados de tempo real é aquele que está sendo atualizado no momento do processo, e o histórico é aquele que armazena os valores já utilizados no processo.

Na análise estatística e no seguimento os dados de entrada são previamente selecionados pelo usuário, que depois do processamento são mostrados na tela através das funções de interface homem máquina por gráficos apropriados.

c) Função de coordenação: esta função é encarregada de coordenar harmoniosamente as tarefas relacionadas com o funcionamento dos equipamentos da célula flexível, através da execução do programa de coordenação. O programa de coordenação é uma seqüência de comandos que permite manter o sincronismo no funcionamento dos equipamentos do chão de fábrica para a produção de uma determinada parte.

A função de coordenação interpreta cada comando do programa de manufatura, converte no formato adequado e envia o comando para sua execução no equipamento respectivo, supervisiona a execução do comando, após a execução do mesmo analisa o comando seguinte, repetindo o ciclo. O processo de manufatura resultante pode ser mostrado através da unidade funcional de interface com o usuário por gráficos em tempo real.

d) Funções de sistema: são as funções relativas ao processo de configuração e manutenção do controlador, que são as funções de iniciação, de configuração, de testes e de manipulação de falhas.

A função de iniciação encarrega-se da ativação do sistema operacional e da execução das rotinas básicas de testes nos módulos do controlador.

Após iniciado o controlador, a função de configuração executa o programa de configuração a partir de dados fornecidos pelo usuário, ou por uma unidade de armazenamento externo, ou enviados do nível de coordenação, preparando o controlador de acordo com o processo de manufatura atual.

A função de teste é ativada quando o controlador está em funcionamento normal e executa os programas de teste utilizados para diagnóstico de falhas. São considerados dois tipos de teste: testes em linha (on-line), que são executados em operação normal, e os teste fora de linha (off-line), que são executados com o controlador isolado do processo de manufatura.

A função de manipulação de alarme também é ativada em operação normal do controlador, e responde ao pedido de outras funções do controlador em situações específicas como por exemplo, interrupções geradas por algum dispositivo em falha

(função de aquisição ou coordenação), falhas de sincronismo (função de coordenação), variável fora dos limites (função de supervisão), falhas de comunicação (função comunicação), ou falhas nos módulos de hardware (função teste).

Os resultados da função de alarme são apresentados na forma de mensagens, travamento de equipamentos, relatórios, ou execução de programas de emergência.

3.2.2.3 Unidade Funcional de Gerenciamento do Banco de Dados

O controlador de célula executa funções que implicam na utilização dos dados previamente coletados, exigindo a execução de certas tarefas em cima destes dados.

O gerenciador de banco de dados é o encarregado da manipulação dos dados e, principalmente, da manutenção da integridade da informação no banco de dados. Para conseguir executar estas funções, o gerenciador deve possuir as seguintes características (32):

- Possuir mecanismos que permitam a manipulação dos dados, como: adicionar, apagar, copiar, procurar e carregar;

- Mecanismos de acesso para o usuário local e para os usuários do nível de coordenação.

- Mecanismos de segurança, limitando o acesso aos usuários autorizados;

No controlador, o banco de dados é composto por duas partes: o banco de dados de tempo real e o banco de dados históricos, o gerenciador deve ter a capacidade de

interoperar com os dois tipos de banco de dados (60).

No banco de dados de tempo real o controlador armazena os dados coletados em tempo real nos equipamentos do chão de fábrica e os dados de configuração do controlador. Os dados armazenados são utilizados pelas funções de controle, de supervisão, de coordenação, e de sistema.

O usuário, através do gerenciador de banco de dados, determina seletivamente que dados devem ser levados para o banco de dados históricos, indicando freqüência e a condição de seleção. O banco de dados históricos armazena parte dos dados do banco de tempo real, que depois serão utilizados pela função de supervisão do controlador e pelos níveis hierárquicos superiores do sistema de automação.

Finalmente, cabe lembrar que para a implementação dos banco de dados devem ser previstas a utilização de dispositivos de armazenamento externos, especialmente para o armazenamento do banco de dados históricos.

3.2.2.4 Unidade Funcional de Interface com o Usuário

É uma unidade funcional complementar às outras unidades especificadas anteriormente e de grande importância, pois, permite a interação do usuário com o controlador de célula flexível.

As interfaces com o usuário estão compostas pelo conjunto de aplicações referentes à apresentação de telas e relatórios. A finalidade desta unidade é de apresentar a informação coletada na forma mas adequada, para que o operador da célula possa ter um melhor conhecimento das operações da produção.

As características das interfaces necessárias serão apresentadas completamente no item 3.4, intitulado interface homem-máquina.

3.2.2.5 Unidade Funcional de Comunicação

Como já foi explicado, o controlador é um equipamento de automação do nível hierárquico de controle que permite a integração da informação do nível de coordenação, e a informação referente ao processo de manufatura.

Para conseguir realizar esta função é necessário implementar no controlador mecanismos de comunicação, que permitam a comunicação do controlador com o nível de coordenação, com outros controladores de célula e com os equipamentos do chão de fábrica.

Estes mecanismos devem implementar os protocolos de comunicação referentes à sub-rede de coordenação e de instrumentação.

Para que o controlador possa enviar informação do nível de instrumentação para o nível de coordenação e vice-versa, é necessário que a função de comunicação execute a transformação dos protocolos e formatos de dados, de modo a obter um alto grau de interoperabilidade entre os elementos das duas sub-rede envolvidas.

As funções definidas nas unidades funcionais apresentadas, permitiram mostrar como deve ser o comportamento funcional do controlador de célula; para completar a concepção do controlador, são necessárias ainda outras considerações importantes, como aquelas que colocam requisitos especiais no funcionamento do controlador de célula.

No item seguinte são apresentados estes requisitos, que são chamados de características gerais do controlador de célula flexível.

3.3 Características Gerais do Controlador

Os benefícios conseguidos com a automação no nível de controle são muitos e são conseqüência das funções que o controlador possui.

Para complementar a concepção do controlador de célula flexível, são necessários atributos que colocam requisitos especiais na arquitetura do controlador.

De acordo, com os objetivos do projeto, o controlador deve ter as características seguintes: possuir uma arquitetura aberta, ser modular, possuir índices elevados de confiabilidade e disponibilidade e possuir manutenção simples e barata. A seguir são apresentados estes requisitos.

3.3.1 Arquitetura Aberta

Este requisito imposto no projeto do controlador vem como conseqüência de duas considerações importantes, que o controlador de célula deve ser um elemento de um sistema aberto de automação, ressaltando que um sistema aberto nem sempre está constituído por equipamentos abertos; e que o controlador deve adaptar-se a qualquer tipo de célula flexível com qualquer grau de complexidade.

Para entender as características e benefícios de uma arquitetura aberta apresenta-se a caracterização dos sistemas abertos de automação e baseado nesta teoria poder-se-á definir o objetivo deste requisito.

Os sistemas abertos de automação são caracterizados por suportarem alto grau de conectividade, interoperabilidade e portabilidade, entre todos os elementos que o compõem, como equipamentos, aplicações, dados e pessoas (6), (13).

Um sistema aberto de automação exige a utilização de padrões internacionais no que se refere aos recursos computacionais e aos perfis funcionais, para elaborar a especificação das interfaces, serviços e formatos de dados, e assim poder garantir as características supracitadas.

Os sistemas abertos possuem os seguintes benefícios (6):

- Flexibilidade: permite a adaptação do sistema de automação a qualquer aplicação da manufatura com qualquer grau de complexidade;

- Expansibilidade: permite a expansão harmoniosa do sistema, com a inclusão rápida de novos elementos, e a inclusão de novas tecnologias evitando a obsolescência do sistema;

- Portabilidade: permite a adaptação das aplicações, dados e pessoas a qualquer plataforma de hardware;

- Interoperabilidade: permite a utilização de protocolos e interfaces padrões, permitirão atingir um ótimo nível de interoperabilidade;

- Integração: permite a integração fácil de todos os elementos do sistema aberto de automação, constituindo um ambiente coerente e, portanto, altamente produtivo;

- Independência: evita a utilização de um ambiente específico de hardware ou

software e o compromisso tecnológico e comercial com um fornecedor exclusivo;

- Investimento: preserva os investimentos que foram realizados no desenvolvimento do software, aquisição de hardware e no treinamento de pessoal;

- Seleção de recursos: facilita, na medida que na atualidade a tendência dos fornecedores é de oferecer cada vez mais produtos com características abertas, o que ajudará ao usuário na seleção dos recursos sem visar um fornecedor específico, pois, os problemas originados pela falta de interoperabilidade serão cada vez menos freqüentes (10).

Todos estes benefícios podem ser traduzidos em critérios, necessários para atingir o principal objetivo do projeto, ou seja conseguir um controlador que possa se adaptar a qualquer tipo de configuração de célula flexível, portanto determina-se que a arquitetura do controlador deverá ser aberta.

Para atingir tal objetivo, como em qualquer sistema aberto, são utilizadas as especificações de padrões internacionais, aplicadas aos diversos recursos utilizados na implementação do controlador de célula flexível, tais como: via de comunicação, arquiteturas de protocolos, aplicações, sistema operacional e linguagens.

Em seguida é apresentado outro requisito que é conseqüência da adoção da arquitetura aberta.

3.3.2 Sistema Modular

A arquitetura do controlador de célula deve ser intrinsecamente modular, sendo que no mínimo uma função deve constituir um módulo. Por exemplo, um módulo pode

implementar a função de comunicação e outro módulo a função de aquisição.

A modularidade permite uma expansão harmoniosa do controlador, incrementando suas funções organizadamente com a inclusão de novos módulos, permitindo a evolução a um perfil de aplicação mais complexo.

A modularidade também envolve flexibilidade, isto é, a adaptação do controlador aos mais diversos processos de manufatura. O usuário pode configurar adequadamente os módulos do controlador para adaptar as funções ao processo selecionado.

Além, da expansão e da flexibilidade de configuração, uma arquitetura modular permite garantir o tempo de vida do controlador e facilita as tarefas de manutenção do controlador.

3.3.3 Disponibilidade e Confiabilidade

A função do controlador de célula no sistema de manufatura exige índices de confiabilidade e disponibilidade elevados, porque, qualquer erro no funcionamento ou interrupção das atividades do controlador provocaria prejuízos financeiros e materiais.

A confiabilidade do controlador é um atributo importante no projeto do controlador. Um alto índice de confiabilidade será atingido com a utilização das técnicas de tolerância a falha, o controlador deve ter capacidade de suportar e manipular diversos tipos de falhas, permitindo no máximo, a degradação do desempenho do sistema total.

Nos módulos que necessitam de um alto índice de disponibilidade, devem ser considerados, os seguintes requisitos (61):

- Falha rápida: cada módulo deve funcionar corretamente ou parar imediatamente;

- Independência de falha: a falha de um módulo não deve afetar o funcionamento dos outros módulos;

- Redundância: os módulos redundantes devem ser instalados e configurados de tal forma que se acontece uma falha o módulo será substituído instantaneamente.

3.3.4 Manutenção

O controlador deve possuir rotinas de testes e diagnóstico específicas para facilitar a manutenção corretiva. Com a execução das rotinas de testes e diagnóstico, determina-se os módulos que estão apresentando falha e substituindo-os rapidamente, sem comprometer a disponibilidade do controlador.

3.4 Especificação da Interface Homem-Máquina

O propósito da interface homem-máquina é de facilitar a comunicação entre o usuário, o controlador de célula flexível e o processo.

Através de uma boa interface homem-máquina, o usuário "dialoga" facilmente com o processo, ignorando completamente a complexidade técnica que envolve o controlador e o processo de manufatura.

As interfaces homem-máquina devem possuir as seguintes características:

- Linguagem de interação: através da utilização de modelos adequados tornar eficiente a interface de ação, correspondente ao conjunto de comandos com que o usuário

instrui a máquina, e a interface de percepção que corresponde à comunicação do computador com o usuário;

- Visar tempos pequenos de aprendizagem e reaprendizagem: são tempos referente ao tempo que usuário toma para conseguir uma eficiente interação com a interface e ao tempo que o usuário toma para reaprender a usar um software já conhecido;

- Apresentação adequada da interface: utilizando técnicas apropriadas que tornem a tarefa de interação mais "amigável", como por exemplo permitir uma fácil movimentação na tela da interface gráfica;

- Baixa taxa de erros: utilização de mecanismos que evitem os erros de interação, especialmente em sistemas de controle como o controlador de célula flexível.

Na atualidade, os sistema de controle da manufatura vem utilizando com bastante sucesso a interface homem-máquina com recursos gráficos, cujas características vem sendo melhoradas pela surgimento de computadores mais poderosos, de novos dispositivos de entrada e saída de desenvolvimento de novos e sofisticados softwares. Por exemplo, os manipuladores de janelas que permitem que vários processos em tempo real convivam numa mesma tela de terminal de vídeo (63).

A seguir, são apresentadas características básicas das interface gráficas que o controlador de célula deve possuir, e em seguida os equipamentos mínimos que deve suportar para facilitar a comunicação usuário-controlador.

a) Interfaces gráficas: de acordo com as funções que o controlador implementa, existem basicamente os seguintes tipos de interfaces gráficas :

- Telas de processos: utilizadas para o acompanhamento do processo de manufatura, compostas de desenhos atualizados em tempo real do processo de manufatura, mostrando os equipamentos do chão de fábricas, as variáveis de controle e as variáveis de estados associadas ao processo;

- Telas de configuração e manipulação de dados: utilizadas para a entrada dos parâmetros de configuração do controlador e dos equipamentos da célula, como por exemplo dados para o gerenciamento do banco de dados, número de elementos da célula, pontos de alarmes, definição de portas, pontos de aquisição, endereços, pontos de operação e descrição dos equipamentos;

- Tabelas e relatórios: utilizadas para complementar o acompanhamento e gerar relatórios sobre o andamento do processo, consistindo de informação resumida dos dados armazenados referentes ao processo de manufatura e apresentados em forma de tabelas apropriadas através do vídeo ou da impressora;

- Gráficos estatísticos: são utilizados como suporte para a análise estatística de processo e para a função de seguimento de variáveis;

b) Equipamentos de interface homem máquina: os equipamento necessários que o controlador de célula deve possuir para implementar as interface homem máquina, são os seguintes:

- Terminal de vídeo: será a console do controlador, através dele o usuário deve executar as diversas funções definidas anteriormente, utilizando as interfaces gráficas de apoio para estas funções;

O terminal deve permitir a ligação de um teclado de engenharia utilizado para a

configuração do controlador e de um de teclado de operação de características robustas para ser utilizado nas operações normais do controlador;

- Sinalizadores de estados: para a indicação do estado de funcionamento do controlador de célula, ou sinalização de alarmes e falhas;

- Botões e chaves: para implementação de funções como, o carregamento do sistema, reiniciação, troca de modo de operação e execução dos procedimentos de testes;

A definição das funções permitiu conhecer as grandes funções do controlador, as características gerais, os atributos especiais que o controlador deve possuir e a interface homem máquina que estabelecem as premissas necessárias para facilitar a comunicação controlador-usuário; todo este conjunto de informações apresentado permite estabelecer o comportamento funcional do controlador de célula flexível.

Para complementar os aspectos anteriores apresenta-se a descrição operacional do controlador de célula flexível, que define os estados que o controlador possui durante seu funcionamento normal.

3.5 Descrição Operacional

Para coordenar a execução das funções do controlador são necessários procedimentos ordenados de operação denominados de modos de operação. No controlador de célula flexível aqui apresentado, são considerados dois modos de operação: o modo centralizado e o modo integrado.

No modo centralizado, correspondente ao estado inicial do controlador de célula,

este assume o controle total do processo de manufatura, isolando-se do nível de coordenação.

O modo de operação centralizado é utilizado especialmente na fase de instalação do controlador, nos testes que envolvem a adaptação de uma nova célula flexível ao processo de manufatura, ou numa operação temporal, onde é necessário carregar o programa de configuração residente nos elementos de armazenamento magnéticos externos.

O outro modo de operação é o modo integrado, que é o correspondente ao estado resultante após a configuração da comunicação com o nível de coordenação da fábrica, isto é, existe um caminho para transportar informação desde o processo de manufatura até o nível de coordenação e vice-versa.

No modo integrado, o operador local do controlador trabalha em forma paralela com os operadores do nível de coordenação, pelo que se considera os comandos locais prioritários sob os comandos provenientes do nível superior.

A passagem do modo integrado para o modo centralizado é realizado de duas formas: de forma automática quando existe uma falha de comunicação do controlador com o nível de coordenação; e de forma manual, quando o operador ativa os botões do painel frontal ou executa um comando por software.

A passagem do modo centralizado para o modo integrado é realizada só de forma manual, isto é, utilizando os botões do painel frontal ou executando comandos do software do controlador.

A figura 3.3 apresenta um diagrama de estado resumindo os modos de operação

e as formas da passagem. Nesta figura, ao ligar o controlador entra no estado iniciação, que corresponde à execução da função de configuração do controlador de célula; após a transição INIT_COMP, que significa que o ciclo de iniciação foi terminado com sucesso, ele passa para o estado centralizado.

Para passar do estado centralizado ao estado integrado são necessários as transações COMAND_INTE_ATIVO e BOTÃO_INTEG, que representam o comando de software e a ação do botão no painel frontal, corresponde à passagem manual.

Para passar do estado integrado para o estado centralizado utiliza-se as transições COMAND_CENT_ATIVO e BOTÃO_CENT, que representa a passagem manual. A transição COMUNI_SUP_NÃO representa a passagem automática, que só acontece no caso de queda na comunicação com o nível superior.

Em cada estado existem transações que saem e entram no mesmo estado, eles representam as transações normais que acontecem em cada estado e que não provocam alterações de estado.

Neste capítulo apresentou-se a definição funcional do controlador de célula flexível, que é o embasamento para a definição da arquitetura do controlador de célula flexível apresentada no capítulo seguinte.

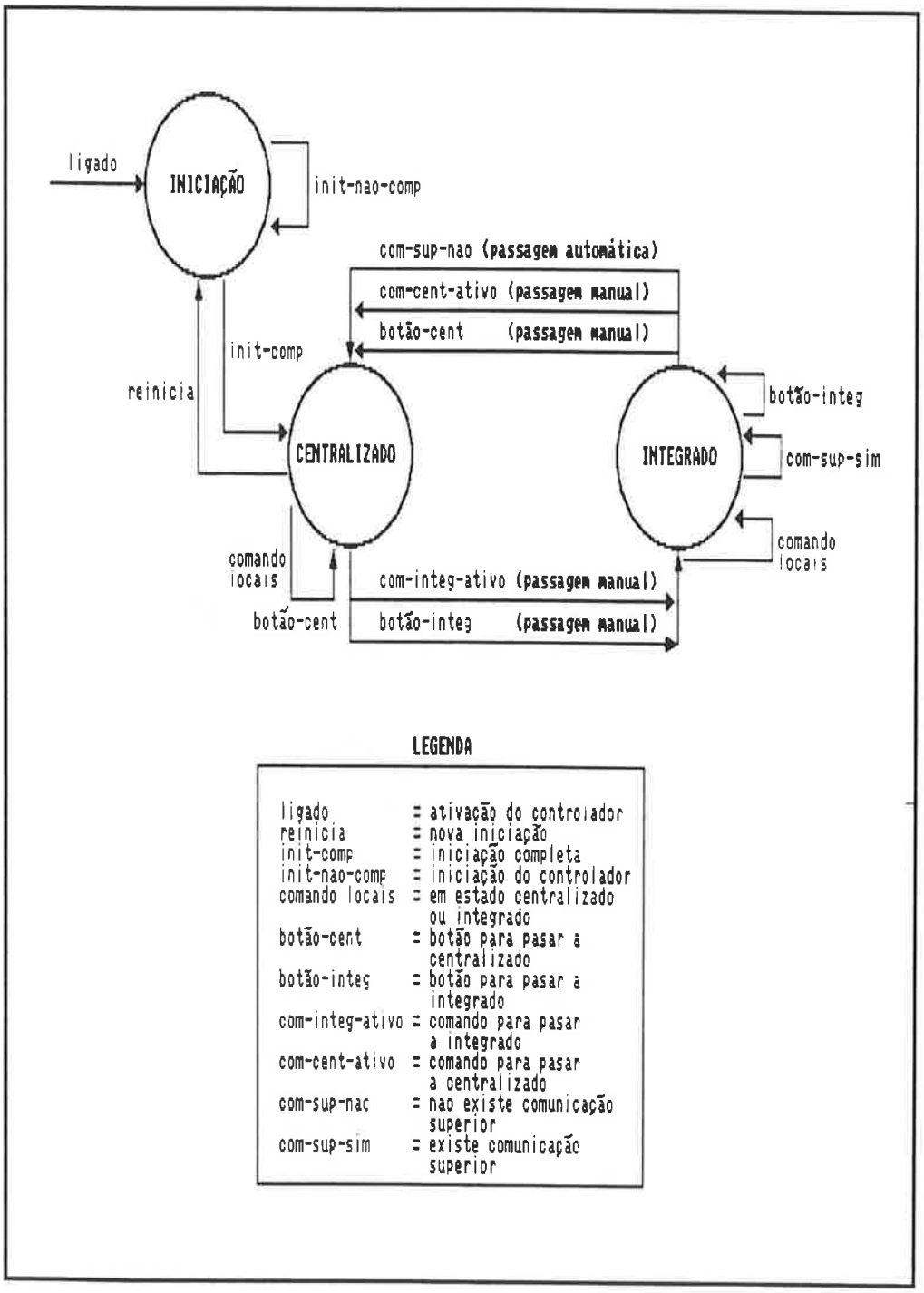

Figura 3.3 diagrama de estado mostrando os modos de operação

4 DEFINIÇÃO DA ARQUITETURA DO CONTROLADOR

A definição da arquitetura ou projeto preliminar é a fase de desenvolvimento de projeto que tem como objetivo principal a determinação da arquitetura que implementará todas as funções do controlador.

A arquitetura do controlador de célula flexível deve ser uma conseqüência lógica da especificação funcional, apresentada no capítulo anterior, e da aplicação de uma metodologia de projeto, explicada neste capítulo.

Como resultado da premissa anterior deve ser obtida uma estrutura modular e flexível, onde cada módulo representará uma ou mais funções do controlador, este resultado visa facilitar a tarefa de implementação do controlador.

Este capítulo será dividido em três partes, na primeira parte será definida a metodologia utilizada neste capítulo; na segunda parte a aplicação da metodologia de projeto no controlador; e, finalmente, apresenta-se a proposta de arquitetura do controlador de célula flexível.

4.1 Metodologia de Definição do Controlador

A seguir, será apresentada a classificação das principais metodologias de projeto, com o objetivo de determinar quais são os requisitos e as características que deverá possuir a metodologia de projeto a ser utilizada nesta fase.

4.1.1 Classificação das Metodologias

As metodologias de projeto utilizadas nesta fase são as mesmas metodologias

utilizadas para projetos de software, só que para o caso do controlador de célula, serão analisadas aquelas que são aplicadas para sistemas de tempo real.

As metodologias de projeto são caracterizadas pela forma como representam os diversos aspectos do sistema, tais como o fluxo dos dados, as funções, ou a intercomunicação entre funções; e também pelos diferentes métodos que utilizam para chegar a um sistema modular.

Existem três grupos representativos das metodologias de projeto, as orientadas ao fluxo de dados, as orientadas as comunicações, e as orientadas ao objetos, que a seguir serão apresentadas (64).

a) Orientada ao fluxo de dados: aqui o sistema é dividido em módulos de acordo ao seu comportamento funcional, cada módulo representará uma ou mais funções do sistema. Para conseguir esta divisão é utilizada a análise estruturada e o projeto estruturado descrito em (65).

   Estas metodologias são muito apropriadas para projetos onde cada função do sistema, envolve fortes operações computacionais, enquanto que a comunicação entre os processos não é muito importante.

   A principal desvantagem destes tipos de metodologias é que são inadequadas para aplicações onde existe um certo grau de paralelismo entre as funções do sistema, pois não possuem mecanismos de intercomunicação e de sincronismo que permitam definir este comportamento. Exemplos deste grupo são referenciados em (66), (67), (68).

b) Orientada às comunicações: a comunicação entre as funções é o principal aspecto

considerado para obter um sistema modular. Cada módulo é definido em termos de comunicação e sincronismo que existe entre as funções do sistema.

Estas metodologias de projeto são recomendáveis para aplicações nas quais os aspectos de operações computacionais e de estruturas dos dados são muitos simples, mas onde os aspectos de comunicação e sincronismo são os mais ressaltantes.

As desvantagens encontradas nesta classe de metodologias são: não existem mecanismos adequados para representar os dados e operações dentro de um módulo; e a representação do sistema não é de fácil entendimento, pois são baseados em aspectos matemáticos.

Neste grupo, estão principalmente todos os métodos baseados em redes de Petri, como o que esta descrito em (69).

c) Orientada a objetos: o sistema é dividido num conjunto de objetos cooperativos, onde cada objeto representa uma entidade que pode ser um dado, uma ação, ou um dispositivo, aqui o principal objetivo é definir as estruturas dos objetos e identificar o relacionamento entre os mesmos .

Estas metodologias estão se tornando bastante aceitas pelas seguintes vantagens: são ideais em sistemas onde existe paralelismo entre as funções e o projeto final possui um alto grau de modularidade. Entretanto, as metodologias de projeto orientados a objetos ainda estão em evolução (70) ainda não é recomendável sua utilização, pois ainda estão em fase de consolidação.

Conhecendo os diversos tipos de metodologia pode-se colocar certos requisitos

mínimos que a metodologia para este projeto deverá possuir, estes requisitos são os seguintes:

- Deve ser de fácil entendimento e de fácil aplicação, deve suportar ferramentas de software que auxiliem na utilização;

- Deve proporcionar mecanismos para o análise das funções paralelas;

- Deve proporcionar mecanismos para analisar a intercomunicação e o sincronismo entre tarefas;

- Deve apresentar no final a definição do projeto com um grau elevado de modularidade.

A seguir é apresentada a metodologia de projeto desta fase de projeto.

4.1.2 Metodologia do Projeto Preliminar

A metodologia selecionada para a fase de definição da arquitetura é o chamado DARTS (Design Approach for Real Time Systems) ou projeto aplicado para sistemas de tempo real apresentado em (9) e (71).

O DARTS é uma metodologia de projeto que possui características do grupo orientados ao fluxo de dados e do grupo orientado às comunicações, sendo sua utilização, prática e de rápido entendimento.

As etapas que constituem esta metodologia são: análise estruturada, análise da concorrência, definição das interfaces e projeto estruturado. Cabe ressaltar, que serão

feitas adaptações na metodologia para que seja totalmente aplicável ao projeto do controlador de célula.

4.1.2.1 Características Gerais

O DARTS considera os requisitos mínimos exigidos no item anterior para o projeto do controlador e possui as seguintes características gerais:

- Orientado ao fluxo de dados: um sistema que deverá ser facilmente adaptável e expansível, deverá possuir uma boa análise de fluxo de dados (72). A primeira etapa desta metodologia é a análise estruturada;

- Orientado as comunicações: define mecanismos de sincronização e de intercomunicação entre funções, estes mecanismos exploram as primitivas do sistema operacional;

- Possui "abstração de dados": critério utilizado para isolar as características de cada módulo independente dos outros, evitando o acoplamento entre módulos. A vantagem desta característica é que o sistema será flexível e expansível;

- Dependência de estados: define mecanismos para a manipulação de transações, cujos resultados dependem dos dados de entrada e do estado atual do sistema.

Estes foram as principais razões para a seleção da metodologia de projeto, no item seguinte será apresentada cada etapa que constitui o DARTS.

4.1.2.2 Análise Estruturada

Nesta etapa são utilizados os diagramas de fluxo de dados (DFD) como ferramentas de análise. Baseados na especificação funcional, é analisado o fluxo de dados através do sistema e determinadas as principais funções do sistema.

No DFD as funções são representadas por círculos, o fluxo de dados entre as funções por flechas, os armazenadores de dados por duas linhas paralelas e as entidades externas representadas por retângulos, como se pode observar no exemplo apresentado na figura 4.1.

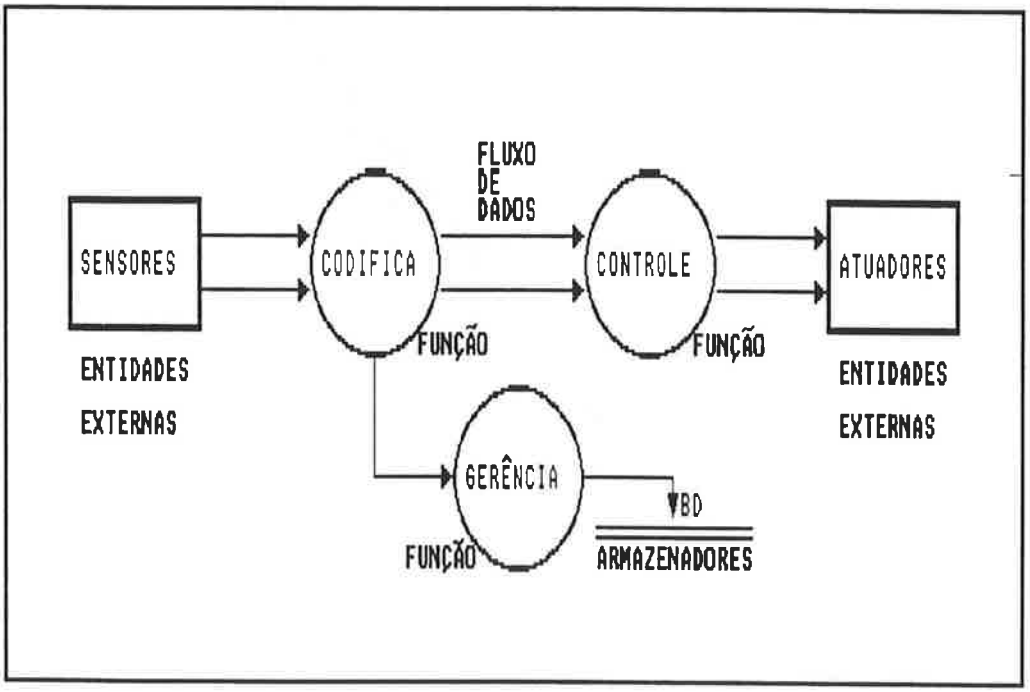

Figura 4.1 Elementos do diagrama de fluxo de dados

Os diagramas de fluxo de dados do sistema devem ser elaborados num nível suficiente de abstração para se poder determinar as principais funções do sistema, em (65) encontra-se uma explicação mais detalhada desta etapa.

Baseados neste diagrama de fluxo, na etapa seguinte serão agrupadas as funções para formar as tarefas do sistema, como será descrito a seguir.

4.1.2.3 Análise da Concorrência

É a etapa mais importante da metodologia de projeto, porque permite determinar o grau de paralelismo na execução das funções do controlador, e como resultado desta análise o sistema é dividido em tarefas, onde cada tarefa pode representar uma função ou um conjunto de funções.

A análise é realizada sob o diagrama de fluxo fornecido na etapa de análise estruturada e para determinar os módulos de tarefas concorrentes devem ser estabelecidos os seguintes critérios:

- Dependência de entrada e saída: quando uma função interatua com dispositivos de entrada e saída, a execução da função é dependente da velocidade do dispositivo, neste caso esta função representa uma única tarefa;

- Funções de tempo real: são funções que precisam ser executadas com a mais alta prioridade, estas funções representam uma única tarefa;

- Forte processamento: são as funções que consomem tempos de processamento elevados, e são consideradas tarefas de baixa prioridade, estas devem representar uma única tarefa;

- Coesão funcional: funções que possuem um intenso tráfego entre elas devem ser agrupadas numa única tarefa. Estas funções dentro da tarefa devem representar módulos separados, assim será assegurada a coesão funcional nos módulos do

projeto (48);

- Coesão temporal: funções que são executadas no mesmo tempo devem ser consideradas dentro de uma mesma tarefa, pois são executadas cada vez que a tarefa recebe um estímulo externo. Dentro da tarefa estas funções devem ser módulos independentes, pela mesma razão anterior;

- Execução periódica: as funções que precisam ser executadas periodicamente podem ser consideradas como uma tarefa separada que é ativada em intervalos de tempo pré -estabelecidos.

Esta etapa da metodologia também é chamada de projeto no nível de tarefas. No seguinte item serão apresentados os mecanismos de intercomunicação entre as tarefas.

4.1.2.4 Definição das Interfaces

Determinadas as tarefas do sistema, o passo seguinte é determinar os mecanismos de comunicação entre as tarefas. No diagrama de fluxos de dados as interfaces de comunicação não aparecem, pois estão na forma de fluxos de dados ou embutidos nos armazenadores de dados.

Nesta etapa, serão definidos dois tipos de módulos de comunicação, o módulo de intercomunicação e o módulo de sincronização.

a) Módulo de intercomunicação: encarregado de manipular as comunicações entre as tarefas, define os procedimentos de acesso e contém a estrutura dos dados transmitidos ou enviados. Este módulo utiliza as primitivas proporcionadas pelo sistema operacional.

São definidos para esta metodologia (9) dois tipos de módulos de intercomunicação:

- Módulo de intercomunicação tipo mensagem: que opera com uma fila de mensagens, pode suportar duas classes de comunicação: a fortemente acoplada e a fracamente acoplada.

  A comunicação fracamente acoplada corresponde a uma fila onde se pode armazenar várias mensagens de uma tarefa para serem utilizadas assincronamente por outra tarefa.

  Na comunicação fortemente acoplada uma tarefa envia uma mensagem e espera a resposta da outra tarefa para enviar uma nova mensagem. As figuras 4.2.a e 4.2.b, representam estes dois tipos de módulos;

- Módulo de intercomunicação tipo dados comuns: que opera com uma memória de dados comum as tarefas, que define os mecanismos de comunicação entre as tarefas, a figura 4.2.c mostra a representação deste módulo.

b) Módulo de sincronização: é o módulo principal ou supervisor dentro de cada tarefa, manipula um conjunto de sinais eventos, que são os encarregados de ativar uma tarefa. Utilizam as primitivas do sistema operacional para esperar e enviar eventos.

4.1.2.5 Projeto Estruturado

A etapa final da metodologia DARTS é denominada de nível de módulo e consiste no projeto estruturado de cada tarefa. O diagrama de fluxo de dados de cada

tarefa é desenvolvido até um bom nível de detalhe, de forma que mostre os diferentes módulos que constituem a tarefa, para depois ser aplicado o método denominado de projeto estruturado definido em (65) e em (48).

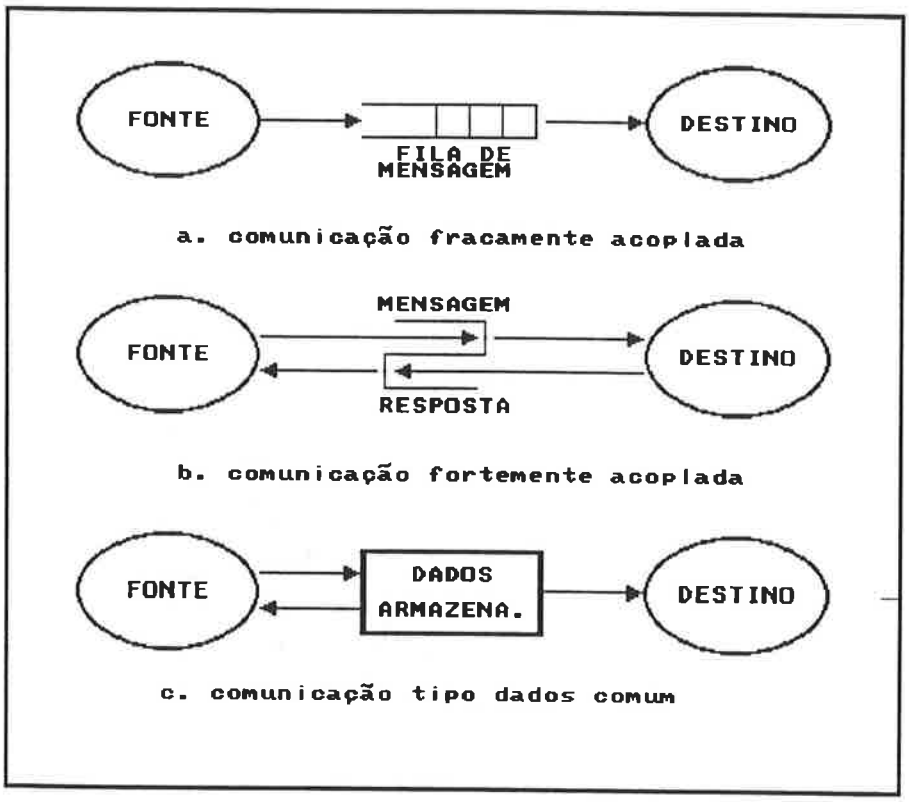

Figura 4.2 Tipos de interfaces de comunicação

O DARTS, nesta etapa do projeto estruturado, considera um módulo gerenciador de transição de estados, cuja função é manter o estado atual do sistema e uma tabela de transição de estados, que define os estados válidos e proibitivos do sistema.

Este módulo gerenciador é utilizado pelos módulos cujas respostas não dependem só dos dados de entrada mas também do estado atual do sistema.

Esta última fase da metodologia, o projeto estruturado, será utilizada no projeto do controlador, para definir o software aplicativo, sendo as duas fases anteriores

utilizadas para definir a arquitetura do controlador de célula.

Este item teve como objetivo a caracterização da metodologia de projeto, denominada DARTS, e que será utilizada na definição da arquitetura do controlador. A seguir será apresentada a aplicação do DARTS no controlador de célula flexível.

4.2 Projeto Preliminar do Controlador

As etapas da metodologia de projeto, DARTS, serão aplicadas ao projeto do controlador de célula flexível, sendo portanto o objetivo principal desta fase do projeto a obtenção de um sistema composto por módulos, de forma que cada módulo possua uma ou mais funções do controlador definidas no capítulo anterior.

Para conseguir este objetivo, primeiro é realizado uma análise de fluxo de dados entre as grandes funções do controlador de célula, depois é feita uma análise com o objetivo de agrupar as funções em módulos e conjuntamente são definidas as interfaces entre os módulos, e após estas etapas é definida a arquitetura do controlador.

4.2.1 Análise Estruturada do Controlador

O comportamento funcional do controlador será analisado utilizando os diagramas de fluxo de dados, com o objetivo de compreender melhor a interrelação e a definição das funções.

A figura 4.3 apresenta o DFD (diagrama de fluxo de dados) do controlador, determinado segundo o procedimento proposto em (67) e utilizando a ferramenta computacional PC-CASE-DFD (73).

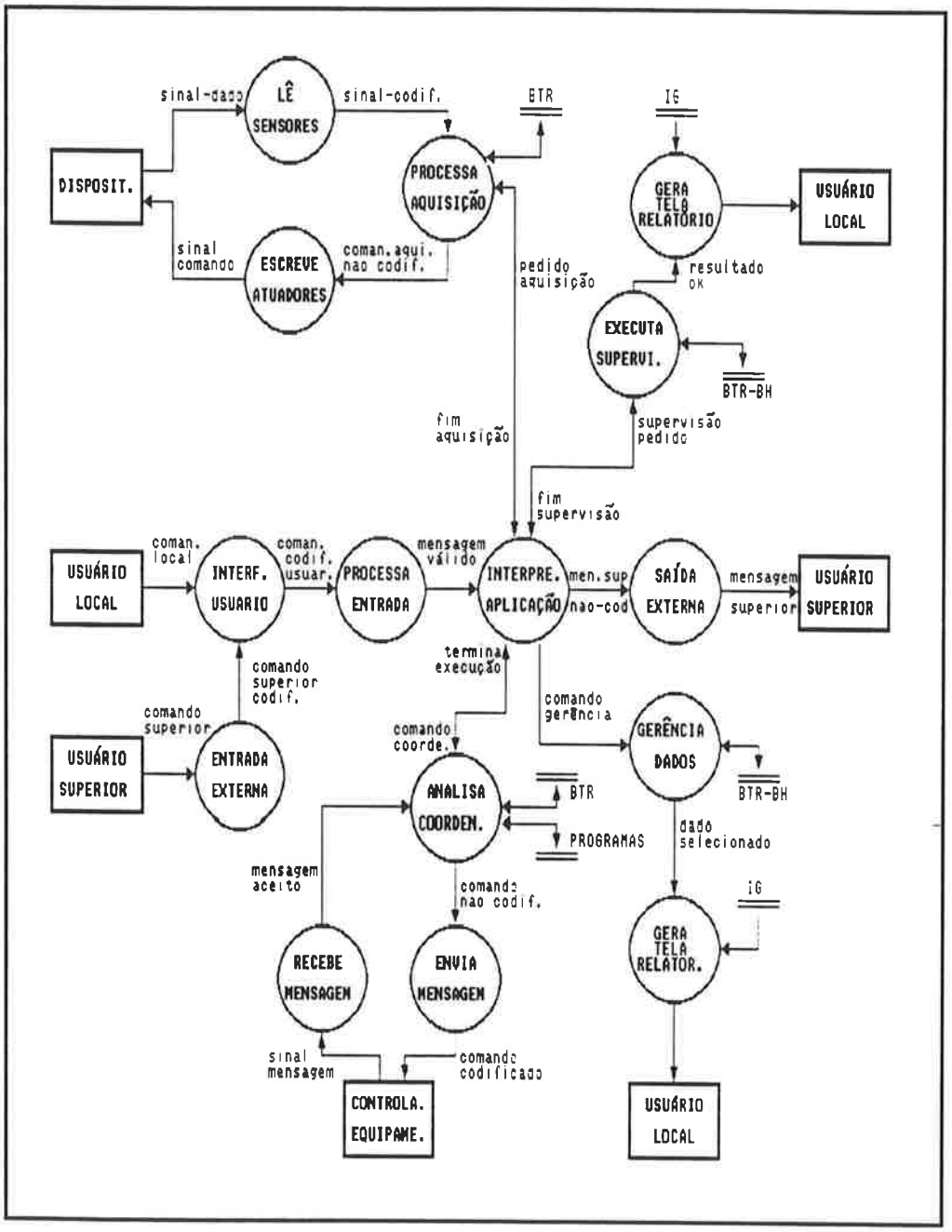

Figura 4.3 Diagrama de fluxo de dados do controlador

Para um melhor entendimento do DFD são definidos a seguir os elementos do diagrama, que constituem o seguinte: entidades externas, transformações ou funções, fluxo de dados e armazenadores.

a) Entidades externas: representados na figura como retângulos, são os elementos que forneceram dados e obtem informações do controlador, tem-se os seguintes:

- DISPOSITIVOS: representa os diferentes dispositivos de entrada e saída, sensores e atuadores do chão de fábrica;

- CONTROLADORES-EQUIPAMENTOS: representa aos equipamentos de automação do chão de fábrica;

- USUÁRIO LOCAL: encarregado de interatuar com o controlador através da console local;

- USUÁRIO SUPERIOR: representa os usuários do nível de coordenação e de outros controladores.

b) Funções: representadas por circulos, correspondem basicamente às funções do controlador, as principais funções da figura 4.3 são:

- INTERFACE-USUÁRIO: encarregada de transformar os comandos enviados pelos usuários em formatos inteligíveis pelo controlador;

- PROCESSA-ENTRADA: valida os comandos de entrada, segundo a tabela de estado do controlador (modos de operação), e envia sinais-eventos de parada e reinicio para outras funções;

- INTERPRETA-APLICAÇÃO: interpreta os comandos de entrada e os encaminha para serem executados nas respectivas funções, coordena a execução harmoniosa das funções do controlador;

- EXECUTA-SUPERVISÃO: encarregada de executar a função de seguimento;

- PROCESSA-AQUISIÇÃO: executa as funções de aquisição, supervisão e intertravamento;

- LÊ-SENSORES: transformação que converte os sinais dos sensores em formatos adequados;

- ESCREVE-ATUADORES: transmite a informação aos dispositivos em formatos adequados;

- ANALISA-COORDENAÇÃO: é encarregada da execução do programa de coordenação, descarrega os programas de manufatura nos respectivos equipamentos;

- RECEBE-MENSAGEM: coleta e interpreta as mensagens dos os equipamentos, transformado-a no formato adequado;

- ENVIA-MENSAGEM: codifica as mensagens do controlador e as transmite de forma que possa ser entendida pelos equipamentos;

- GERENCIA-DADOS: encarregada da manipulação dos bancos de dados;

- GERA-TELA-RELATÓRIO: monta e mostra as respectivas telas definidas na interface homem-máquina aos usuários;

- ENTRADA-EXTERNA: interpreta as mensagens externas, ou seja, recebe as

mensagens do nível de coordenação e de outros controladores e as interpreta;

- SAÍDA-EXTERNA: envia mensagens nos formatos exigidos pelos outros equipamentos do nível de controle e de coordenação.

c) Fluxo de dados: representam os dados que são utilizados pelas funções. A tabela 4.1 mostra os principais tipos de dados da figura 4.3.

d) Armazenadores: representados por duas linhas paralelas, são encarregados de armazenar os resultados, as configurações, os programas e os dados em geral. A figura 4.3 apresenta os seguintes armazenadores:

- Banco de dados de tempo real (BTR): são os dados do processo referentes às variáveis armazenados em tempo real,que representa o estado atual do processo;

- Banco histórico (BH): dados do banco de dados de tempo real selecionados pelo usuário para futura utilização, ou dados de configuração;

- Banco de programas (PROGRAMAS): armazena os comandos e programas destinados aos equipamentos; e

- Interfaces gráficas (IG): são os aplicativos gráficos utilizados para mostrar resultados nas tela do console.

e) Interpretação do DFD: as atividades dentro do controlador podem ser ativadas através de comandos enviados pelos diversos usuários, estes comandos são detectados pela função INTERFACE-USUÁRIO, e imediatamente validados pela

função PROCESSA-ENTRADA, que mantém uma tabela com a informação dos modos de operação do controlador.

FLUXO DE DADOS	DESCRIÇÃO
sinal-dado	sinais que representam mensagens ou dados dos estados das variáveis
sinal-comando	representa os comandos de atuação ou os dados de configuração
sinal-codifica	sinal em formato inteligível pelo controlador
comando-aquis-não -codifica	sinal que representa os comandos nao empacotados
pedido-aquisição	comando de pedido de aquisição
fim-aquisição	fim da execução da aquisição
comando-local	comandos do usuário local
coman-codif-usua	comando do usuário em formato adequado
mensagem-válido	comando válido para execução de uma função
termina-execução	sinal de fim de execução da coordenação
comando-coordena	inicio de execução da coordenação
mensagem-aceito	sinal do equipamento no formato adequado
coman-nao-codif	representa comandos de equipamentos
comand-codificado	comandos no formato de transmissão
sinal-mensagem	sinal dos equipamento nos protocolos externos
comando-gerência	inicio de execução da gerência
dado-selecionado	dados selecionados do banco de dados
mensa-sup-nao-cod	mensagens em formato do controlador enviados para o nível superior
mensagen-superior	mensagens em formato de protocolos
comando-superior	pedido do usuário superior
coman-sup-codifi	pedido superior em formato controlador
supervisão-pedido	comando pedido de execução de supervisão
fim-supervisão	sinal de fim de execução
resultados ok	sao os resultados da função de supervisão

Tabela 4.1 Tipo de dados do DFD

As mensagens válidas são tomadas pela função INTEPRETA-APLICAÇÃO, que

interpreta a mensagem, executa a operação possível e a encaminha para a função pedida.

Qualquer função que seja pedida, tais como, aquisição, supervisão, coordenação, gerência de bancos de dados, ou comunicação externa, assim termine sua execução, envia um sinal-evento, indicando a finalização, para a função INTERPRETA-APLICAÇÃO, encarregada da coordenação destas funções.

A função GERA-TELA-RELATÓRIO é uma função complementar a todas as outras funções, pois todas elas a utilizam de acordo a seu pedido.

Da análise estruturada pode-se concluir uma visualização da posição de cada função do controlador no sistema como um todo, esta posição mostra o grau de independência existente entre as funções.

As funções do controlador são definidas com maior precisão, visando ser elementos de um sistema, completando a definição da especificação funcional.

As transformações dos dados e o fluxo de dados permitem dar uma idéia do grau de complexidade de cada função, dos elementos adicionais para sua execução (armazenadores) e das interfaces entre as funções, isto é, dos argumentos que serão utilizados na definição da arquitetura.

4.2.2 Concorrência no Controlador

O objetivo da análise da concorrência é determinar o grau de paralelismo que existe na execução das funções do controlador. No final desta análise deve-se mostrar um sistema composto por módulos, denominados tarefas, onde cada um representará

uma função ou um agrupamento de funções do controlador de célula.

Da análise estruturada encontra-se a relação e a cooperação entre as funções do controlador, esta conclusão permite determinar o grau de independência de execução entre as funções, esta conclusão e as considerações do item 4.1.2.3 servem para determinar os módulos tarefas, como é mostrado na figura 4.4.

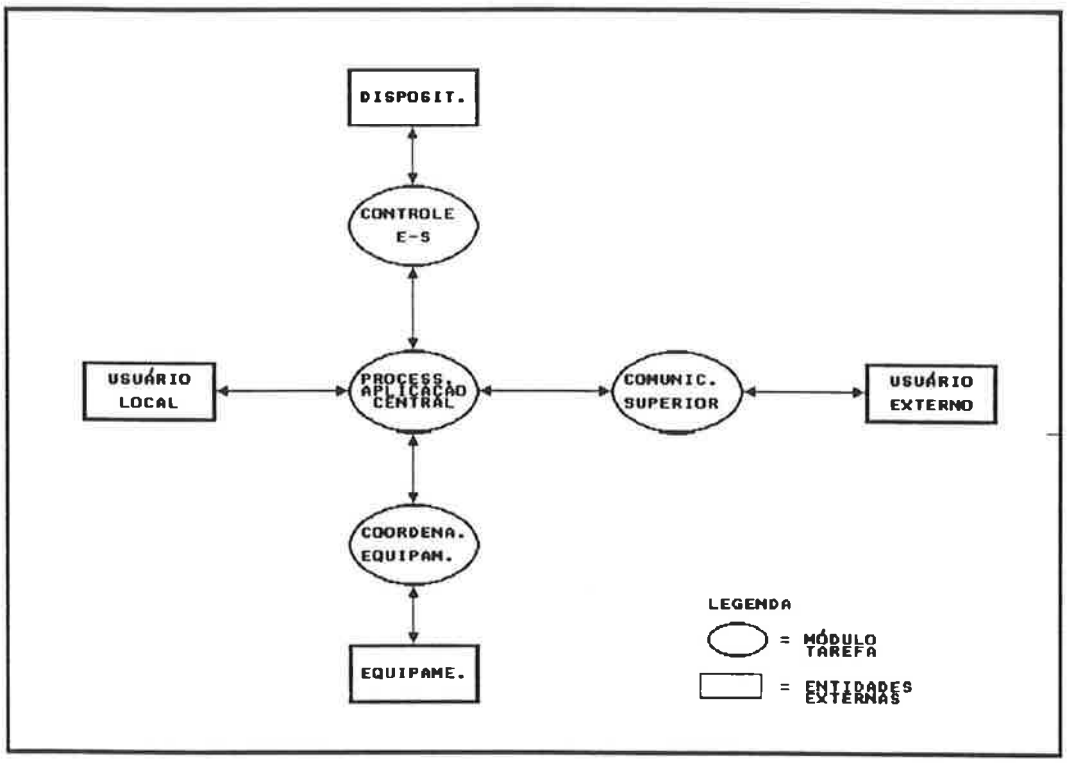

Figura 4.4 Controlador no nível de tarefas

As funções LÊ-SENSORES, ESCREVE-ATUADORES, e PROCESSA-AQUISIÇÃO do DFD da figura 4.3 formam um módulo tarefa, denominado de CONTROLE-ENTRADA-SAÍDA, e foram agrupadas pela utilização das seguintes considerações: dependência de entrada e saída, a coesão funcional e por envolver funções de tempo real.

As funções ANALISA-COORDENAÇÃO, ENVIA-MENSAGEM e RECEBE-

MENSAGEM constituem o módulo COORDENA-EQUIPAMENTOS. Aqui, as funções ENVIA-MENSAGEM e RECEBE-MENSAGEM são exemplos do critério chamado de coesão temporal, que são agrupadas com a função ANALISA-COORDENAÇÃO pela coesão funcional. Os módulos tarefas CONTROLE-ENTRADA-SAÍDA e COORDENA EQUIPAMENTO são chamados de módulos de tempo real.

O módulo tarefa COMUNICAÇÃO-SUPERIOR é constituído por duas funções: a ENTRADA-EXTERNA e a SAÍDA-EXTERNA, que são agrupadas pela utilização do critério da coesão funcional.

Todas as outras funções correspondem às funções que possuem forte processamento computacional, ou são de prioridade baixa, e são casos de coesão funcional, estas funções são agrupadas como o módulo tarefa PROCESSADOR-APLICAÇÃO-CENTRAL.

Finalmente, o sistema termina sendo constituído por quatro módulos tarefas, como são: CONTROLE-ENTRADA-SAÍDA, COORDENA-EQUIPAMENTOS, COMUNICAÇÃO-SUPERIOR, PROCESSADOR-APLICAÇÃO-CENTRAL.

O que se ressalta nesta análise é que se consegue determinar que o funcionamento interno de um módulo tarefa é realizado paralelamente aos outros módulos, e só existe uma troca de informações mínima, correspondente aos dados e mensagens utilizados na execução das funções ou os sinais de término ou começo de execução.

Para completar a análise, a seguir são mostradas as interfaces de comunicação que permitem a troca de informações entre os módulos tarefas.

4.2.3 Comunicação entre Módulos Tarefas

Determinados os módulo tarefas paralelos, o passo seguinte é determinar as interfaces entre estes módulos. Para determinar as interfaces serão utilizados os conceitos explicados em 4.1.2.4.

A figura 4.5 apresenta o controlador com seus módulos tarefas e mostrando as interfaces de comunicação. Cabe indicar que nesta fase alguns fluxos de dados foram transformados em sinais-eventos representados por flechas tracejadas.

O módulo tarefa PROCESSADOR-APLICAÇÃO-CENTRAL comunica-se com o COMUNICAÇÃO-SUPERIOR, CONTROLE-ENTRADA-SAÍDA e COORDENA-EQUIPAMENTOS, através de filas de comunicação, pois a informação entre elas corresponde a conjuntos de comandos e mensagens, estas tarefas devem possuir módulos de comunicação tipo mensagem fracamente acoplados.

Quando duas tarefas manipulam um mesmo banco de dados, como por exemplo o banco de dados de tempo real, é necessário a implementação de módulos de manipulação do tipo dados comum, representados por BTR1 e BTR2 na figura 4.5.

No PROCESSADOR-APLICAÇÃO-CENTRAL deve-se definir um módulo gerenciador de transações de estado para validar as mensagens de entrada, um módulo de sincronização para manipular sinais de parada e execução e para a coordenação da execução de todas as outras tarefas.

Os módulos tarefa CONTROLE-ENTRADA-SAÍDA, COMUNICAÇÃO-SUPERIOR, e COORDENA-EQUIPAMENTO possuem um módulo de sincronização para o tratamento dos eventos de parada, e a tarefa COORDENA-EQUIPAMENTOS

possui um módulo de manipulação de comunicação fortemente acoplada para a comunicação com os equipamento de automação.

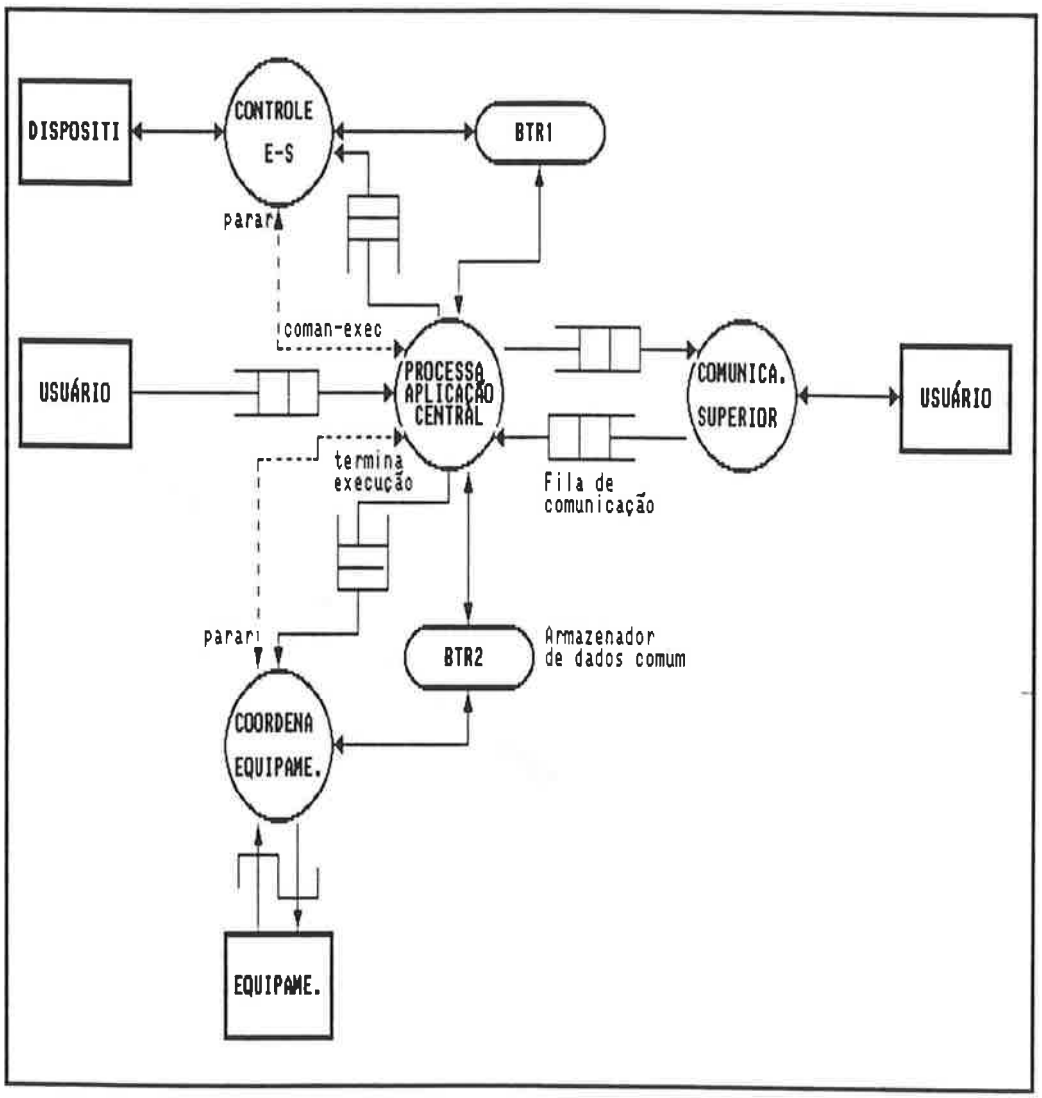

Figura 4.5 Interfaces de comunicação entre tarefas

A metodologia de projeto DARTS permitiu definir claramente as funções do controlador, detectar o paralelismo entre as funções através da definição dos módulos tarefas e se conseguiu mostrar como é a comunicação entre estes módulos. Estas conclusões são necessárias na definição da arquitetura do controlador de célula flexível apresentada a seguir.

4.3 Arquitetura do Controlador de Célula Flexível

A arquitetura do controlador deve ser uma conclusão das duas fases de projeto até agora apresentadas, a especificação funcional e o projeto preliminar.

Enquanto, a especificação funcional determina a complexidade das funções e os atributos que a arquitetura deve cumprir, a segunda fase determina, especificamente, o grau de paralelismo existente na execução das funções, a divisão das funções em módulos tarefas e a cooperação entre eles.

A característica da arquitetura que cumpre com as considerações anteriores e que está sendo muito utilizada em diversas aplicações de tempo real são as denominadas arquiteturas distribuídas, que são constituídas por múltiplos processadores, interligados através de algum meio de comunicação (74), (75).

A proposta apresentada a seguir considera as conclusões das referências (76), (77), (78), (79), (80), (81), (82), (83), que são trabalhos ao redor das arquiteturas distribuídas.

4.3.1 Arquitetura Geral

A arquitetura geral do controlador é mostrada na figura 4.6, esta arquitetura esta composta de um grupo de nós, onde cada nó é composto por um conjunto de processadores representados por P1,P2, até Pn, estes nós estão interligados através de um sistema de via de comunicação principal, e os processadores, dentro de cada nó, estão interligados através de outro sistema de via de comunicação local. O sistema se comunica com os elementos do sistema aberto de automação através de interfaces de protocolos apropriados.

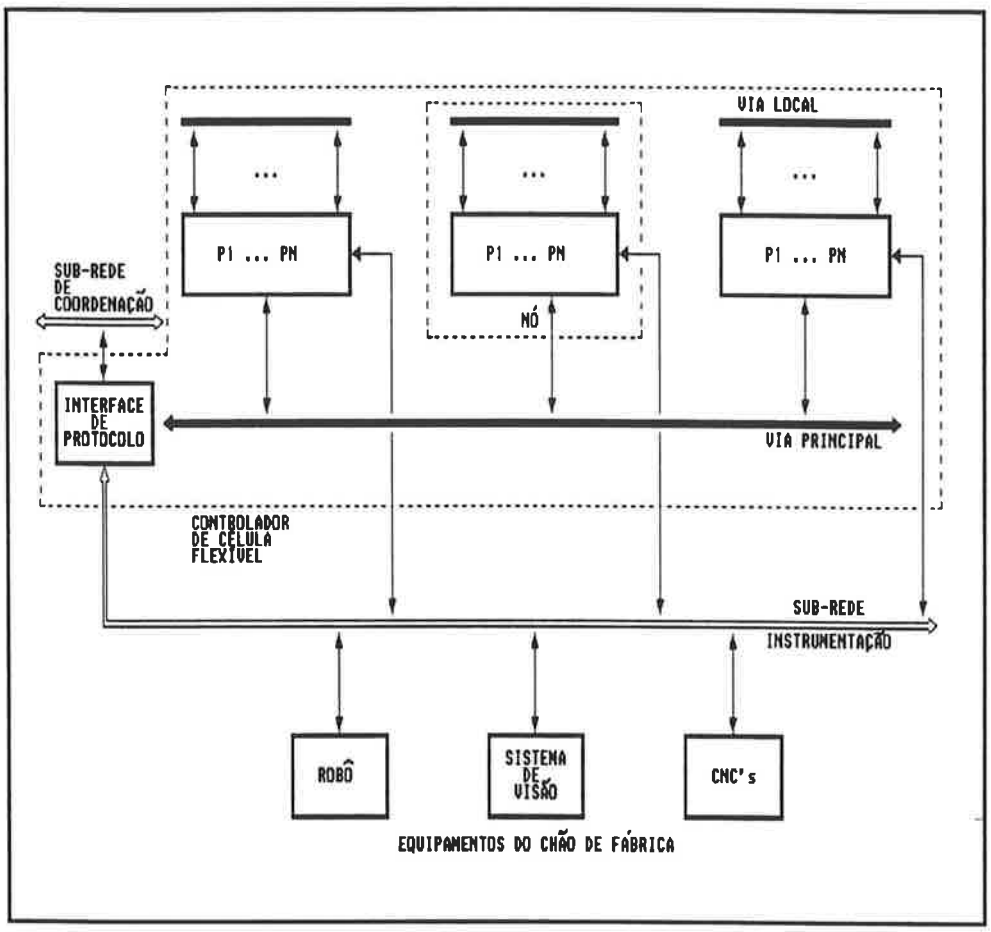

Figura 4.6 Arquitetura do controlador

Este tipo de arquitetura distribuída, possui as seguintes vantagens:

- Flexibilidade: são funcionalmente flexíveis, a flexibilidade é o requisito indispensável para conseguir a extensibilidade, permitindo o desenvolvimento de sistemas genéricos para uma ampla faixa de aplicações (84);

- Modularidade: a utilização de módulos independentes no projeto permite melhorar as características de expansão, confiabilidade e manutenção do sistema;

- Alto grau de disponibilidade e confiabilidade: são bastante apropriados para aplicar

diversas técnicas de tolerância a falha e possuem uma inerente disponibilidade, pois sua estrutura permite a fácil implementação de redundâncias;

- Prováveis ganhos no desempenho: pela distribuição da tarefas nos múltiplos processadores e pela utilização de um eficiente mecanismo de intercomunicação;

- Boa utilização dos recursos compartilhados: é possível compartilhar os recursos do sistema numa forma efetiva e econômica.

Cabe ressaltar que os benefícios conseguidos pelas arquiteturas distribuídas foram colocados como atributos na especificação funcional, tais como a flexibilidade, a modularidade e a disponibilidade, alias a própria estrutura desta arquitetura permite utilizar os módulos tarefas definidos na análise estruturada, como se fossem processadores ou nós independentes.

Nos itens seguintes são mostrados as características de cada elemento da arquitetura da figura 4.6.

4.3.1.1 Estrutura do Nó

Como já foi explicado, cada nó agrupa um conjunto de processadores, não necessariamente simétricos, formando um sub-sistema local, onde cada processador pode representar um processador simples, um sistema de memória, ou um processador dedicado.

Este sub-sistema local ou nó tem a característica de poder representar uma ou várias funções do controlador, indicando que a complexidade do nó depende das funções que implementa, do desempenho necessário e da redundância que se necessite.

Por exemplo, na configuração do controlador de célula, um sub-sistema ou nó pode representar a tarefa de coordenação, o qual está formado por um processador, que se encarrega da parte computacional, e outro processador de comunicação para a comunicação com os equipamentos e a transformação dos protocolos.

A interligação dos processadores no mesmo nó, é realizada através de um sistema de via local, que possui mecanismos apropriados de intercomunicação (85), e permite que os processadores compartilhem os recursos de hardware e software locais.

A principal vantagem desta via local e de diminuir a sobrecarga da via principal e obter um maior desempenho na execução das tarefas do controlador.

Todos os nós ou sub-sistemas são interligados através de uma via principal, que proporciona uma conexão global, permitindo a troca de informações, o aproveitamento dos recursos globais adequadamente, a implementação de redundâncias, e, principalmente, permite a operação dos sub-sistemas cooperativamente e concorrentemente.

4.3.1.2 Comunicação Externa

Um dos atributos mais importante considerado na especificação funcional é que o controlador será um elemento de um sistema aberto de automação, portanto deve possuir mecanismos que proporcionem um alto grau de conectividade.

A arquitetura proposta possui uma interface de protocolos, que representa a capacidade da arquitetura do controlador de poder se comunicar com os elementos do sistema aberto de automação.

Esta interface de protocolos representará um processador de comunicações, que será o encarregado de manipular os protocolos implementados nas duas sub-redes do sistema aberto de automação apresentadas no item 3.1.2.

Cabe ressaltar que o processador de comunicações pode ser representado por um sub-sistema ou nó completo, ou pode ser parte de um nó específico, como por exemplo, pode estar formando parte do processador de coordenação, permitindo a comunicação com os equipamentos do chão de fábrica.

A seguir será mostrada o hardware básico do controlador baseado nas conclusões da análise estruturada e na arquitetura aqui mostrada.

4.3.2 Hardware Básico do Controlador

O hardware básico do controlador no nível de sub-sistemas ou nós é mostrado na figura 4.7 e possui os seguintes sub-sistemas:

a) Sub-sistema de PROCESSADOR-APLICAÇÃO-CENTRAL: é o sub-sistema encarregado de realizar o gerenciamento do controlador em geral, a configuração do controlador e o interfaceamento com o usuário; executa ainda a função de supervisão, o gerenciamento do banco de dados e o processamentos em geral.

b) Sub-sistema de CONTROLE-ENTRADA-SAÍDA: encarregado do processo de aquisição e intertravamento. Para comunicar-se com os dispositivos, sensores e atuadores, ou algum outro equipamento do chão de fábrica, este sub-módulo implementa as interfaces físicas apropriadas e manipula os protocolos adequados da sub-rede de instrumentação.

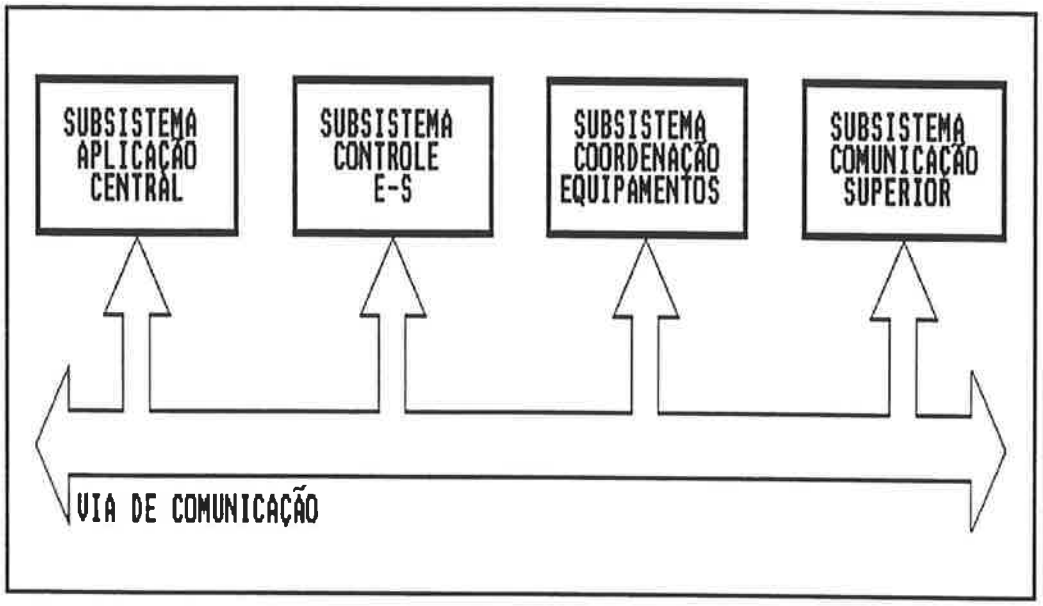

Figura 4.7 Hardware básico do controlador

c) Sub-sistema de COORDENAÇÃO-EQUIPAMENTOS: cumpre a função de coordenação entre os equipamentos de automação que formam a célula flexível, isto é, executa o programa de coordenação para a produção de uma família de partes.

Também, possui capacidade para manipular interfaces e protocolos de comunicação apropriados da sub-rede de instrumentação.

d) Sub-sistema de COMUNICAÇÃO-SUPERIOR: este sub-sistema é o encarregado de fornecer os mecanismos para que o controlador possa comunicar-se com o nível de coordenação e com outros controladores.

A via de comunicação suporta os mecanismos necessários para a comunicação adequada entre os subsistemas localizados dentro do controlador.

Finalmente, os elementos desta arquitetura básica, terão que ser implementados

visando os padrões internacionais, pois esta estrutura será uma arquitetura aberta como foi definido na especificação funcional.

4.3.3 Software do Controlador

O controlador, como foi proposto, deve ser um equipamento flexível e aberto, adaptável a qualquer processo de manufatura.

O software do controlador também deve visar estas características, de tal forma que deve possuir uma estrutura básica e modular que permita posteriormente a adaptação do software ao processo específico (72), (86).

Considerando esta premissa, a estrutura do software em cada sub-sistema deve estar constituída por dois elementos: o sistema operacional e o software aplicativo que a seguir serão apresentados.

4.3.3.1 Sistema Operacional

A arquitetura do controlador de célula flexível exige um sistema operacional de tempo real, pois existem funções do controlador que possuem esta característica. As características que o sistema operacional deve possuir são as seguintes (87):

- Deve possuir mecanismos de intercomunicação entre os sub-sistemas processadores, o que permitirá um alto grau de cooperação entre os processadores (88);

- Deve possuir mecanismos de suportem a característica multitarefa, permitindo a execução de varias funções num único processador;

- Deve possuir mecanismos que permitam ao sistema ser multi-usuário, possibilitando a operação concorrente de vários usuários;

- Em geral deve possuir características de um sistema operacional centralizado, como por exemplo rápido chaveamento entre tarefas, ocupação de pouco espaço na memória manipulação rápida de interrupções, gerenciamento de memória e outros apresentados em (85) e (81).

Estas características proporcionaram um bom conjunto de primitivas, comandos próprios do sistema operacional, que permitiram cumprir com as exigências apresentadas na definição das interfaces de comunicação entre tarefas.

4.3.3.2 Software aplicativo

O outro componente do software é o software aplicativo, que é composto pelo software de controle de cada sub-sistema, cuja constituição depende da função ou as funções que o sub-sistema configure. A seguir, será apresentado o projeto estruturado das funções implementadas em cada sub-sistema.

Para cada aplicativo associado ao sub-sistema o diagrama de fluxo de dados é redefinido num nível maior de detalhe, em seguida é aplicado o método do projeto estruturado explicado em (65), obtendo-se uma estrutura modular. Os aplicativos principais são: PROCESSADOR-APLICAÇÃO-CENTRAL, SUPERVISÃO, CONTROLE-ENTRADA-SAÍDA, COMUNICAÇÃO-SUPERIOR, COORDENA-EQUIPAMENTOS.

Na figura 4.8.a e 4.8.b apresenta-se o DFD e a estrutura modular da aplicação do sub-sistema PROCESSADOR-APLICAÇÃO-CENTRAL, são definidos o módulo

GERA-EVENTOS, encarregado da geração de sinais de parada; o módulo RECEBE-COMANDOS, encarregado de gerenciar as mensagens recebidas de outras tarefas; o módulo PEDIDO-DE-FUNÇÃO, encarregado de ativar outra função; o CODIFICA-ENTRADA, transforma os comandos de entrada; PROCESSA-ENTRADA, valida os comandos; e INTERPRETA-APLICAÇÃO que analisa o tipo de comando .

A figura 4.8.b representa a estrutura modular hierarquizada, aqui o módulo principal APLICAÇÃO-CENTRAL, chama o módulo de LEITURA (lê), que lê através do CODIFICA-MENSAGEM e envia as mensagens para o módulo VALIDA, validada a mensagem este chama o módulo PROCESSA-TRANSAÇÃO, que executa o pedido da função como gerar um evento ou uma tela e utilizando o INTERPRETA-APLICAÇÃO pode executar a supervisão, a aquisição, a coordenação ou a comunicação.

Dentro do sub-sistema PROCESSADOR-APLICAÇÃO-CENTRAL está o aplicativo de SUPERVISÃO, cujo DFD e estrutura modular hierarquizada são mostrados nas figuras 4.9.a e 4.9.b, respectivamente.

Na figura 4.9.a, o pedido de supervisão originado no processo anterior é tomado pelo módulo SELECIONA-SUPERVISÃO, que determina o tipo de supervisão, imediatamente é executado o processo pelo módulo PROCESSO-SUPERVISÃO, cujos resultados são enviados para o módulo GERA-TELA. O módulo de GERENCIA, permite ao módulo PROCESSA-SUPERVISÃO interagir com os bancos de dados e o módulo GERA-EVENTO indica o fim da tarefa.

A figura 4.9.b mostra o projeto estruturado da SUPERVISÃO, o modulo principal PROCESSA-SUPERVISÃO chama o módulo SELEÇÃO que envia a mensagem de execução para o módulo PROCESSA e os resultados são manipulados pelo módulo de SAÍDA.

Nas figuras 4.10.a e 4.10.b são apresentados o DFD e a estrutura modular hierarquizada do aplicativo do sub-sistema CONTROLE-ENTRADA-SAÍDA. Neste aplicativo quando um comando de pedido de aquisição-atuação é detectado este é validado e interpretado pelo módulo VÁLIDA-ANALISA, determinando o programa que o módulo PROCESSA-AQUISIÇÃO deve executar, imediatamente após a sua execução, as mensagens são transformadas e enviadas aos atuadores pelos módulos FORMATA e TRANSMISSÃO, respectivamente.

A leitura de estados dos sensores é realizada pelo módulo RECEBE, que os envia para o INTERPRETADOR DE PROTOCOLOS, que se encarrega de transforma-los em formatos adequados, depois disso os dados são validados e interpretados, sendo processados ou armazenados. A interpretação da estrutura hierarquizada segue a seqüência através dos módulos seguintes: INGRESSO que permite a coleta de dados, VALIDA e SELECCIONA TRANSAÇÃO que executa o processo e a atuação.

O DFD e a estrutura hierarquizada do aplicativo pertencente ao sub-sistema COMUNICAÇÃO-SUPERIOR são representados nas figuras 4.11.a e 4.11.b, respectivamente. Neste aplicativo as mensagens enviadas aos usuários superiores são validadas no módulo COORDENA-COMUNICAÇÃO, que as envia ao módulo de TRANSFORMAÇÃO, que se encarrega de codificar os dados nos protocolos respectivos, e depois são transmitidos pelo módulo ENVIA-SINAL. O mesmo procedimento é realizado na leitura dos comandos dos usuários superiores, porém no sentido inverso. A análise da estrutura hierarquizada é semelhante com a análise dos aplicativos anteriores.

O software aplicativo do sub-sistema COORDENA-EQUIPAMENTOS é representado nas figuras 4.12.a e 4.12.b, na forma de DFD e como uma estrutura

modular, respectivamente. Neste caso o módulo VÁLIDA-INTERPRETA encarrega-se de interpretar os comandos de pedido do PROCESSADOR CENTRAL e do módulo INTERPRETA e ativa o módulo EXECUTA-COORDENAÇÃO. O módulo EXECUTA-COORDENAÇÃO executa o programa de coordenação e envia os comandos para serem transformados pelos módulos TRANSFORMA e TRANSMITE em formatos adequados.

As respostas dos equipamentos são interpretadas pelo módulo DECODIFICA e INTERPRETA e enviadas ao módulo de VALIDAÇÃO, que os encaminha para serem processados ou armazenados. A análise da estrutura hierarquizada é similar ao aplicativo CONTROLE-ENTRADA-SAÍDA.

O objetivo de este capítulo foi a definição da arquitetura do controlador de célula flexível, considerando as conclusões das duas fases anteriores do projeto; no capítulo seguinte será apresentada a implementação do controlador baseado na arquitetura aqui proposta.

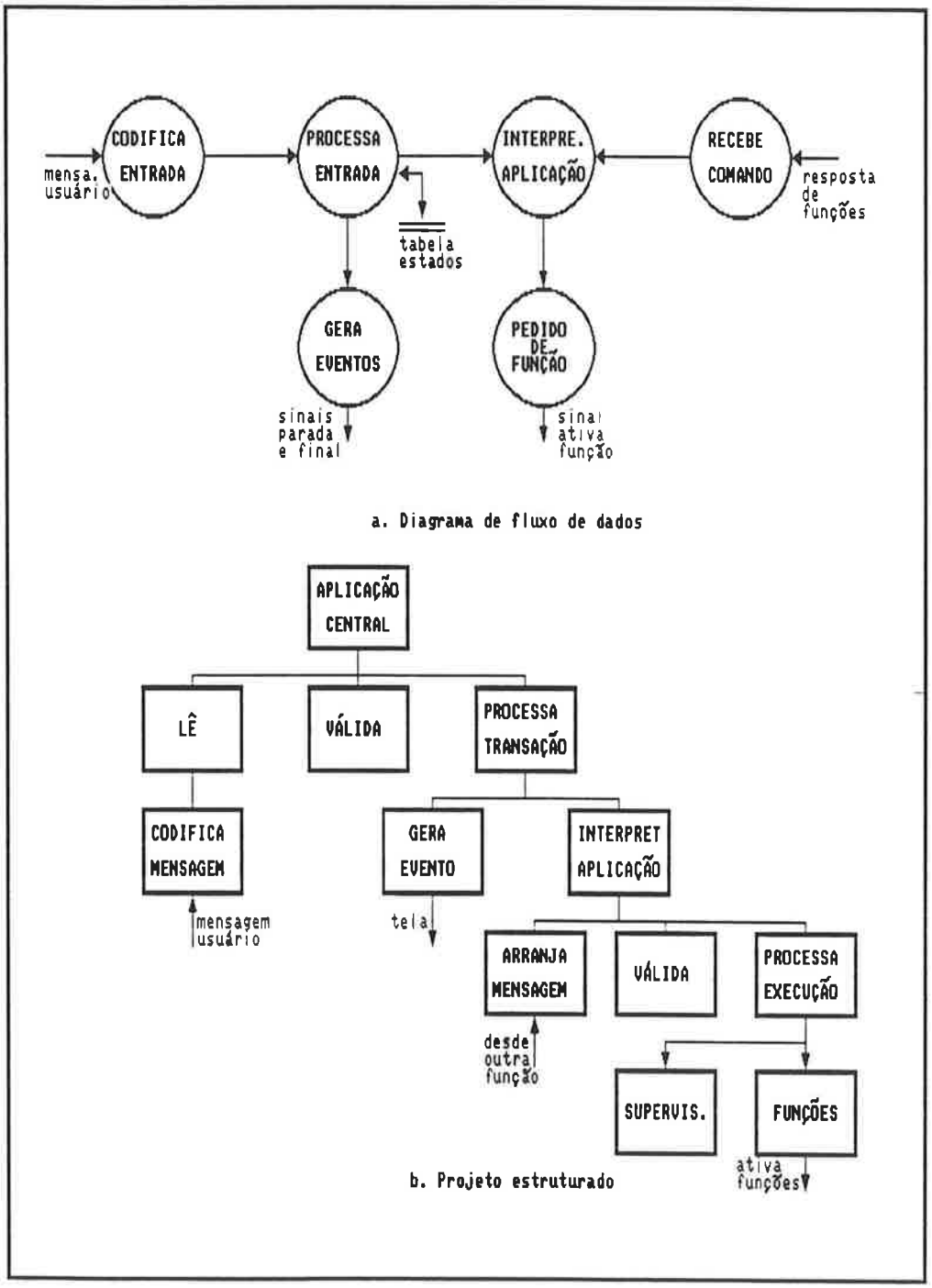

Figura 4.8 Projeto estruturado na APLICAÇÃO-CENTRAL

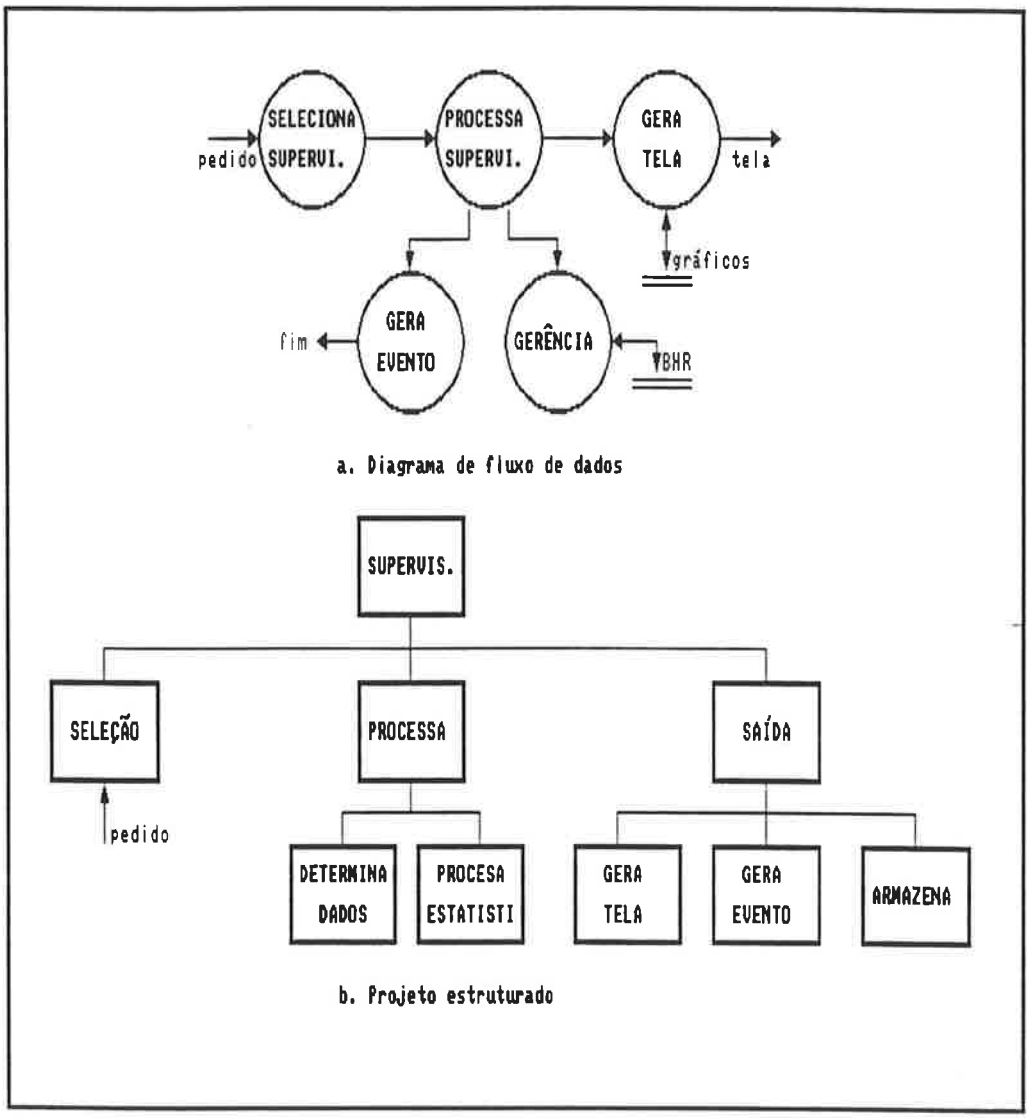

Figura 4.9 Projeto estruturado da SUPERVISÃO

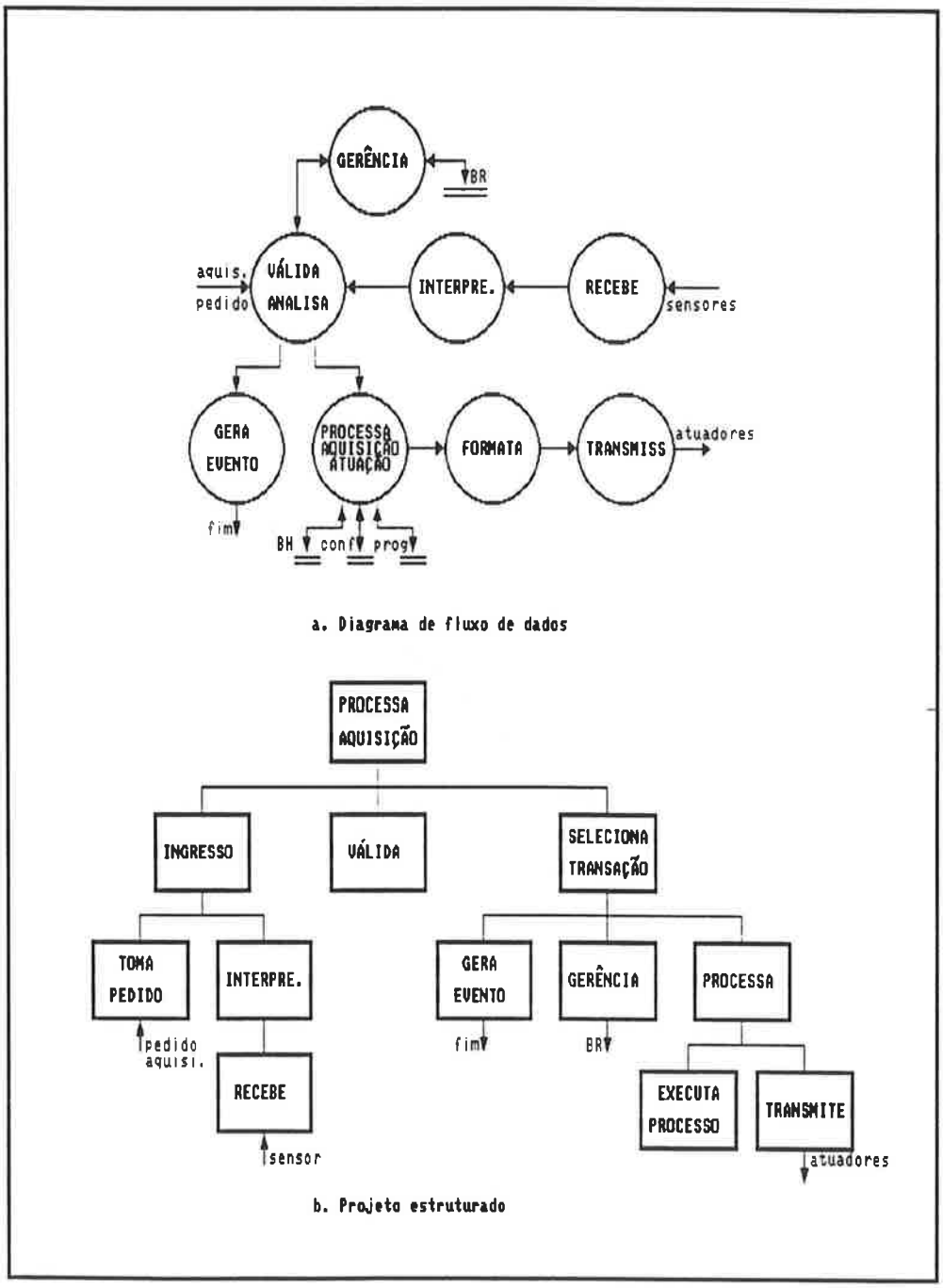

Figura 4.10 Projeto estruturado no CONTROLE-ENTRADA-SAÍDA

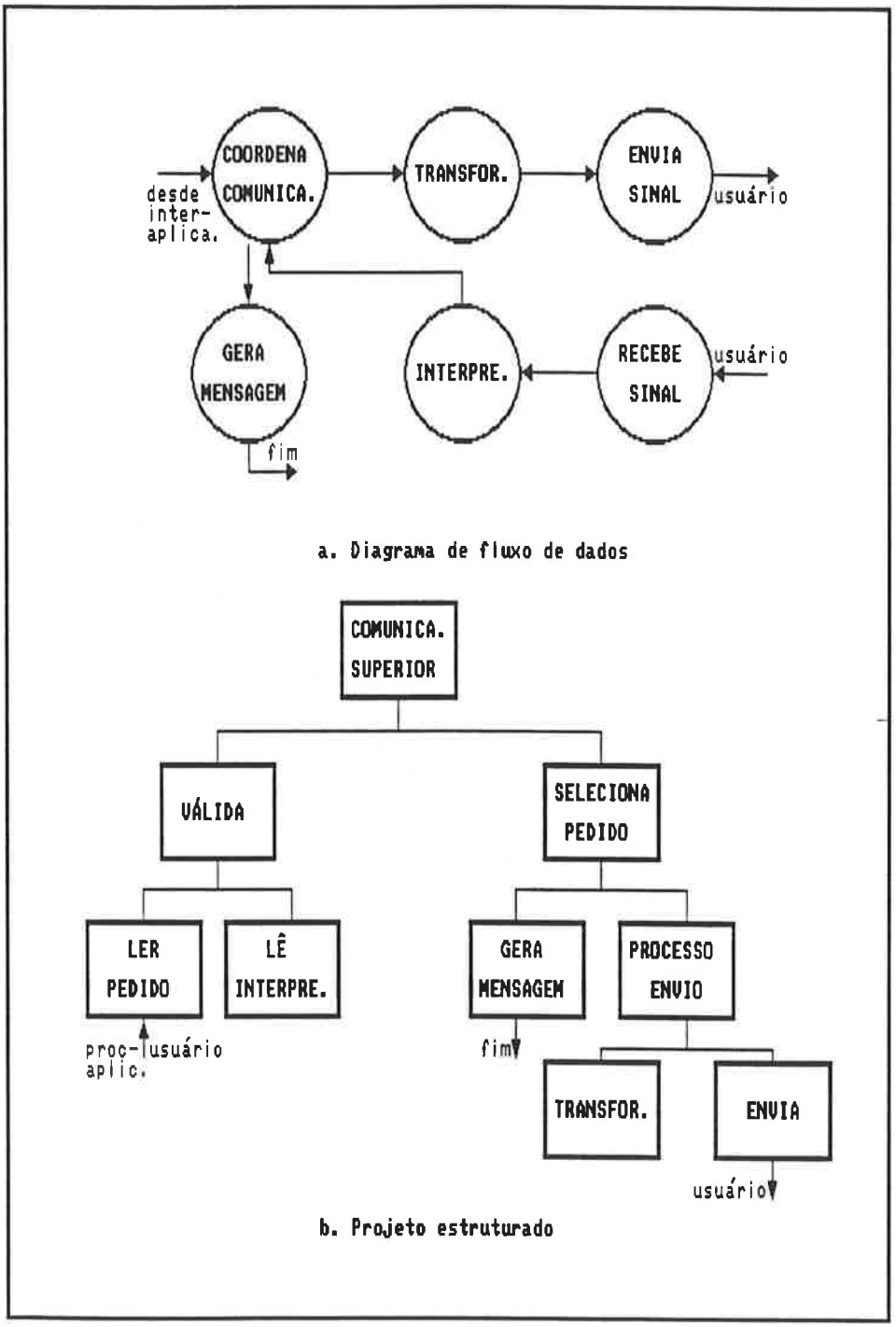

Figura 4.11 Projeto estruturado da COMUNICAÇÃO-SUPERIOR

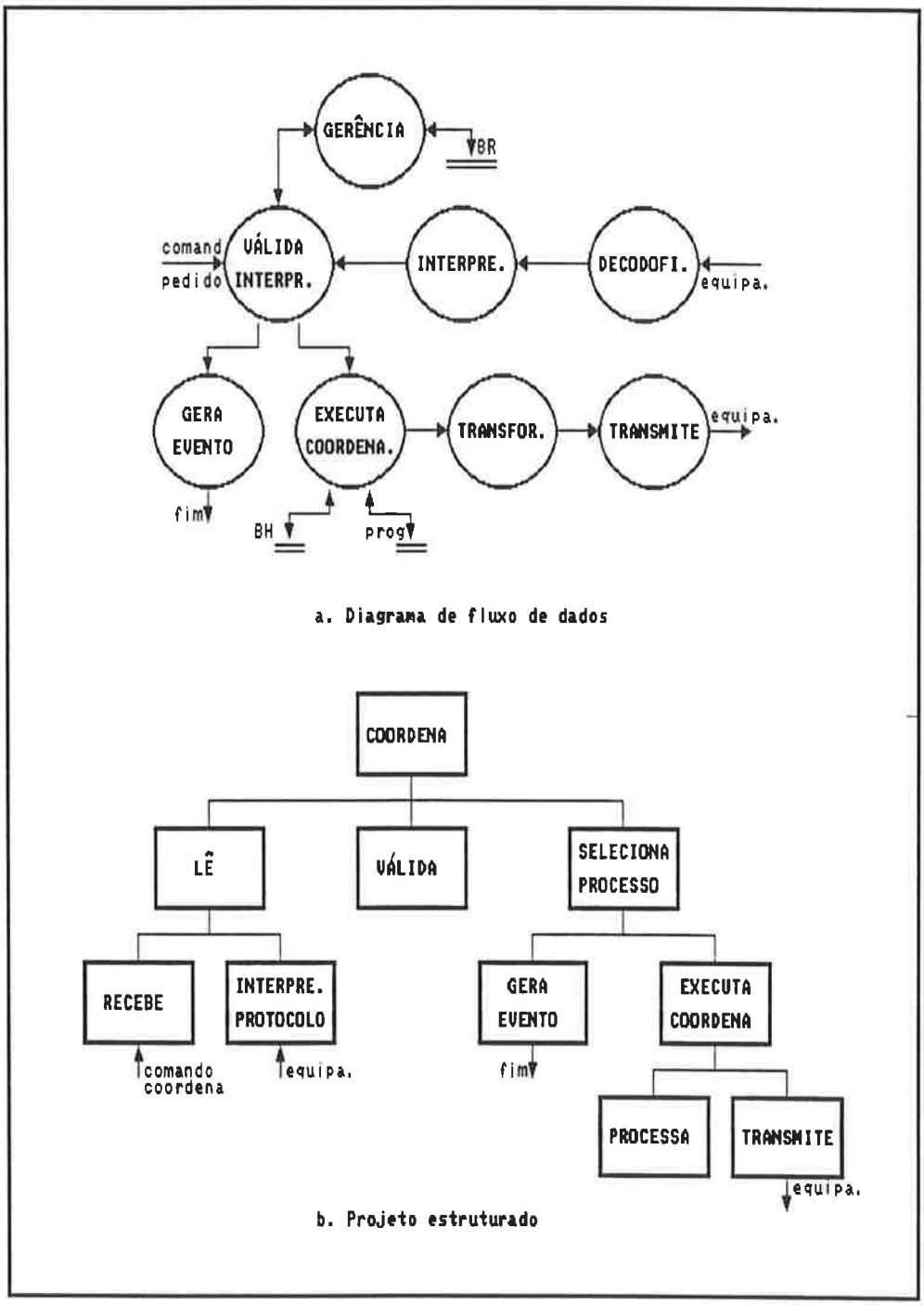

Figura 4.12 Projeto estruturado do COORDENA-EQUIPAMENTOS

5 IMPLEMENTAÇÃO DO CONTROLADOR

A implementação do controlador de célula flexível e os testes de validação correspondem às fases finais do projeto, antes de apresentar o desenvolvimento destas fases, existem duas considerações que merecem a atenção, e que serão expostas nesta introdução.

A primeira consideração importante desta fase de implementação e que ao longo da dissertação foi ressaltada, é a implementação do controlador de célula visando uma arquitetura aberta.

Esta consideração é um atributo da especificação funcional, que exige a utilização de especificações padrões e internacionais na seleção dos elementos que compõem o controlador de célula flexível.

A segunda consideração que merece destaque, nesta fase da implementação, é o chamado nível de integração do sistema, que se refere à integração e testes dos sub-sistemas selecionados e que compõem a arquitetura do controlador.

O nível de integração é determinado considerando os seguintes aspectos: a especificação funcional do controlador, a complexidade da arquitetura, o tempo de desenvolvimento e o custo total do projeto.

O nível de integração utilizado na implementação do controlador de célula é o denominado nível de módulos (Board Level), estes módulos (hardware pronto) são fabricados sem enxergar uma aplicação específica, sendo que a integração de vários destes módulos comporão o controlador de célula flexível.

Este nível de integração traz grandes vantagens ao projeto como: alto grau de flexibilidade na implementação das funções do sistema (um módulo pode implementar várias funções), grande desempenho, custo moderado, alto nível de integração, alto grau de conectividade e uma grande capacidade de seleção de recursos orientados a diversas aplicações (89), (90), (91).

Estas duas considerações, a utilização da arquitetura aberta e o nível de integração baseado em módulos, permitem ao projetista obter um alto grau de flexibilidade e liberdade na hora de selecionar os recursos, pelo que se exige adequados critérios de seleção, requisito indispensável que evita uma implementação fora do escopo da especificação, com prejuízos no custo e no interesse do usuário.

Este capítulo consistirá das seguintes etapas: na primeira apresenta-se a seleção, integração e teste parcial do hardware; na segunda serão apresentados os módulos do software, a integração e os testes parciais; e finalmente a etapa de integração total, onde apresenta-se um plano de teste de validação do controlador.

5.1 Implementação do Hardware

O hardware do controlador estará formado por um conjunto de módulos ou sub-sistemas (Boards), cujas estruturas implementarão as funções do controlador.

Para interligar estes sub-sistemas é utilizado uma via de comunicação padrão, cuja seleção é o primeiro passo da implementação.

Esta seleção complica-se quando no mercado pode-se encontrar inúmeras vias de comunicação padrões, por tais motivos exige-se uma série de considerações, que serão apresentados no sub-item seguinte.

5.1.1 Considerações na Seleção da Via de Comunicação

Devido a existência de uma série de alternativas tecnológicas, no que diz respeito às vias de comunicação, é necessário realizar uma análise baseada nos seguintes fatores: a arquitetura do controlador de célula flexível, os fatores tecnológicos e os fatores complementares.

Estas considerações foram obtidas como resultado da pesquisa realizada nas referências (89), (92), (93), (94), (95), (96), (97), (98).

Primeiro, a arquitetura do controlador é um fator que determina, a complexidade dos módulos, os mecanismos de intercomunicação, o requerimento de velocidade, e a expansão futura. Esta consideração é o principal fator limitante na seleção da via de comunicação (99).

Os fatores tecnológicos que influenciam na análise possuem os seguintes aspectos:

- Natureza da via: corresponde ao número de linhas da via e às características de multiplexação entre as vias de dados e de endereçamento;

- Acesso ao meio: define o gerenciamento do acesso à via de comunicação;

- Largura de banda: determina a velocidade máxima de transferência de dados;

- Protocolo de transmissão: avalia a técnica utilizada, podendo ser síncrona ou assíncrona;

- Interrupções: determina a estratégia e os modos de interrupção;

- Controle das mensagens: utilização de técnicas como por exemplo: envio das mensagens (Message Passing), "caixas postais", "exclusão mutua" e semáforos, definidas em (49), (85);

- Capacidade de multiprocessamento: característica de suportar vários processadores independentes;

- Suporte de software: refere-se a capacidade dos sub-sistemas padrões de suportar elementos do software, como: sistemas operacionais, ambientes de desenvolvimento e aplicações específicas;

- Vias complementares: são vias locais independentes da via padrão principal que servem para a comunicação com dispositivos de entrada e saída ou outras aplicações específicas (100);

- Dimensão do módulo: denominado também de fator de forma, especifica a dimensão do módulo padrão;

- Tipo de conector: define o conector utilizado pelo módulo padrão para conectar-se ao bastidor (Backplane).

Os fatores complementares que influenciam na análise possuem os seguintes aspectos:

- Especificação padrão: selecionar uma via cuja especificação seja totalmente aberta

aos projetistas;

- Suporte internacional: que a especificação seja respaldada por uma instituição internacional. Este e o aspecto anterior asseguram a compatibilidade de recursos dos diferentes fornecedores;

- Custo: o custo envolve hardware (módulos, acessórios) e o software utilizado para a implementação do sistema;

- Número de fornecedores: inclui os fornecedores principais e os secundários. Uma boa via deve ter um conjunto forte de fornecedores que suportem serviços e fornecem respaldo técnico, além disso, asseguram a implementação, expansão e atualização do sistema;

- Independência do processador: uma via projetada sem visar um processador específico assegura a adaptação de novas tecnologias na arquitetura do sistema;

- Utilização de circuitos integrados de alta escala de integração baseados na tecnologias VLSI (Very Large Scale Intergrated) ou ASIC (Applications Specific Integrated Circuit);

- Evolução: é um fator que permite assegurar a vida do projeto, por isso deve-se saber os planos evolutivos dos fornecedores para com a via;

- Prestigio industrial: permite conhecer as experiências de outros projetos utilizando esta via padrão;

- Simplicidade com que foram implementadas as características técnicas da via: é um

fator que permite antever os indices de confiabilidade e as dificuldades de manutenção e integração.

Para certas aplicações podem existir duas ou mais vias que sejam soluções para o problema, pelo que muitas vezes, os fatores complementares são os de maior peso na decisão, este detalhe é mostrado na figura 5.1 (95).

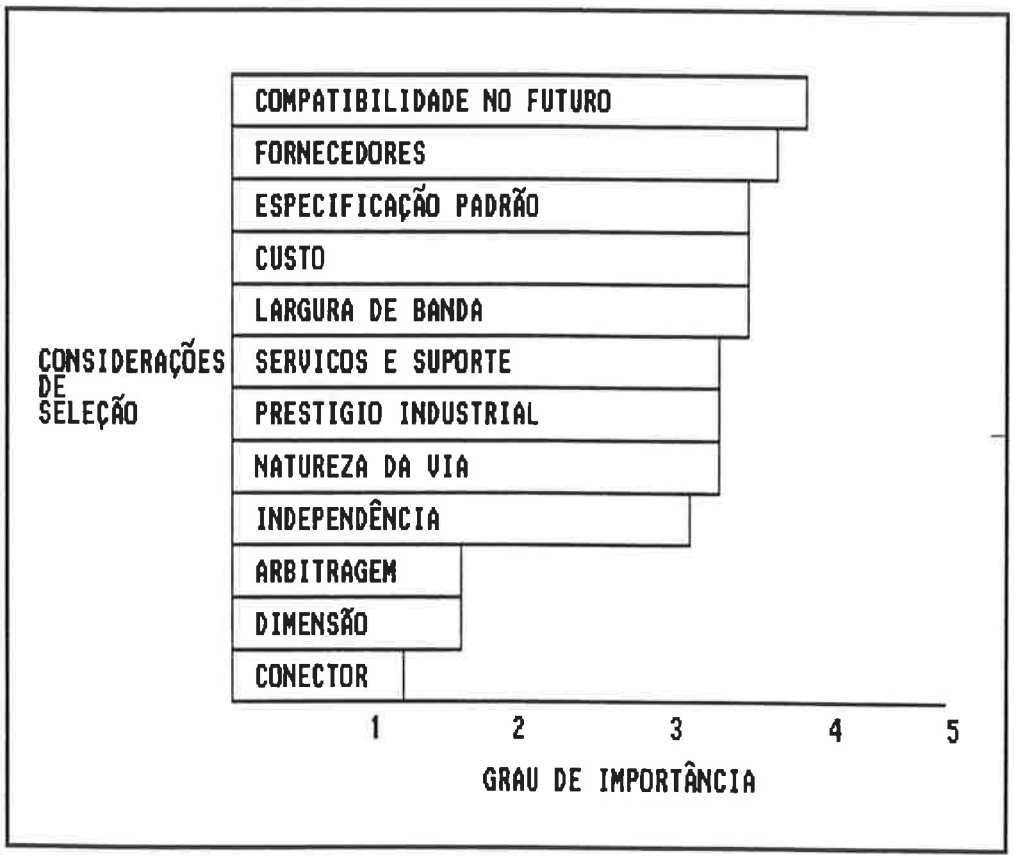

Figura 5.1 Importância das considerações de seleção das vias de comunicação

Considerando todos estes fatores é que foi selecionada a via padrão IEEE P1014 como a mais adequada para a implementação do controlador, pois possui as melhores condições nos fatores supracitados, e que a seguir será apresentada.

5.1.2 Caracterização da Via Selecionada

Para o projeto do controlador foi seleccionada a via padrão IEEE P1014, por ser uma via padrão que possui as características anteriores bem definidas dentro do escopo da especificação do controlador, e pelas diversas aplicações analisadas desta via, tanto no nível acadêmico como no industrial, o que dá um forte respaldo à decisão.

As principais características desta via estão compreendidas nos seguintes itens: os módulos funcionais, as sub-vias, e as considerações elétricas e mecânicas (101), (102), (103), que serão apresentadas a seguir.

5.1.2.1 Módulos funcionais

A via padrão P1014 pode ser dividida funcionalmente em módulos funcionais e sub-vias, como é mostrado na figura 5.2. Nesta figura os módulos fisicos denominados de sub-sistemas implementam os módulos funcionais da via e os módulos funcionais específicos do sub-sistema, estes sub-sistemas são conectados ao bastidor que representa as linhas das sub-vias da via padrão.

A seguir são apresentados as características dos módulos funcionais da via P1014:

- Relógio: proporciona um sinal de 16 Mhz para todos os sub-sistemas que o desejem utilizar;

- Monitor de potência: este módulo entrega o sinal de reinício de operação e o sinal de falha de alimentação para todo os sub-sistemas conectados ao bastidor;

- Mestre da sub-via de transferência de dados: encarregado de iniciar e controlar a transferência de dados entre ele e os módulos escravos de outros sub-sistemas. Para realizar uma transferência de dados, o mestre deve ganhar o controle da via fazendo antes o pedido ao módulo de pedido da via no mesmo sub-sistema;

- Escravo da sub-via de transferência de dados: responde ao pedido de transferência de dados, para o qual possui a capacidade de decodificar as linhas de endereçamento da via e responder aos sinais enviado pelo mestre;

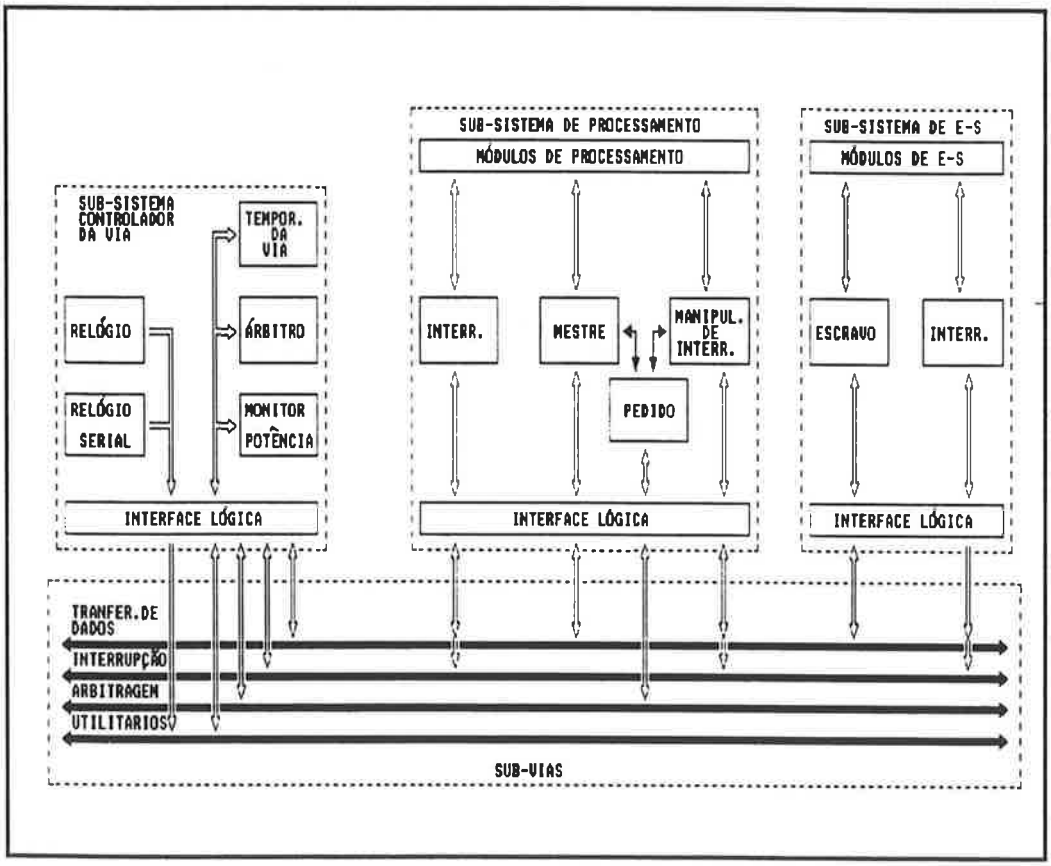

Figura 5.2 Módulos funcionais e sub-vias do padrão IEEE P1014

- Módulo de pedido da via: encarregado de pedir o controle da via para a transferência de dados. Depois que o sub-sistema ganhou o controle da via, este módulo de pedido de via avisa aos módulos mestre e manipulador de interrupções

que podem realizar a transferência de dados;

- Módulo arbitro: arbitra os pedidos de utilização da via e decide que sub-sistema ganhou o controle da via para realizar a transferência de dados;

- Módulo de interrupção: encarrega-se de gerar um sinal de interrupção quando o sub-sistema requer interromper o funcionamento normal de outro sub-sistema, e se encarrega de gerar um vetor de estado durante o ciclo de interrupção;

- Manipulador de interrupções: é o encarregado de manipular os pedidos de interrupção aplicando mecanismos apropriados de prioridade; pedir o controle da via para a transferência de dados ao módulo de pedido de via e iniciar a seqüência de serviço de interrupção baseado no vetor de estado gerado pelo módulo anterior;

- Módulo temporizador da via: encarregado de medir o tempo em cada transferência de dados e finaliza o ciclo de transferência de dados quando o tempo consumido está fora do valor configurado. Quando um mestre deseja transferir dados a um escravo cuja localização não existe, pode ficar esperando indefinidamente, este módulo evita que o sistema fique travado interrompendo a transferência;

- Relógio Serial: implementa um sinal de relógio para sincronizar a transferência de dados de forma serial, esta via serial faz parte da sub-via de utilitários que será explicada posteriormente.

Cabe ressaltar que os módulos funcionais da via, tais como, o módulo árbitro, o monitor de potência, o relógio, o módulo temporizador da via e o relógio serial, constituem o sub-sistema controlador da via, indicando que um sub-sistema mais complexo pode implementar estes módulos funcionais e outros módulos específicos do

sub-sistema.

5.1.2.2 Sub- vias

A estrutura da via IEEE P1014, apresentada na figura 5.2, mostra as quatro sub-vias que a seguir são mostradas:

- Sub-via de transferência de dados (VTD): é uma via dedicada à transferência de dados entre os sub-sistemas (102);

  Esta via permite larguras de dados, configuradas dinamicamente de 8, 16, 32 bits, o que permite a coexistência de microprocessadores de 8, 16, 32 bits. As transferências de dados podem ser realizadas assincronamente e por blocos de largura definida;

- Sub-via de arbitragem: dedicada ao controle de acesso à via, é utilizada pelo módulo pedido da via e pelo módulo árbitro.

  A arbitragem acontece quando vários pedidos de via são entregues ao módulo árbitro, este decide que sub-sistema ganha o controle da via utilizando três algoritmos que são configuráveis pelo usuário (102): o arbitragem tipo prioridade, baseado num esquema fixo de prioridades; o arbitragem "Round Robin" ou tipo rotação, onde existe uma rotação de prioridades; e a arbitragem nível simples, onde os pedidos são aceitos através uma única linha de mais alta prioridade.

- Sub-via de interrupções: é a via através da qual são pedidas as interrupções e é utilizada pelo módulo de interrupção e pelo módulo de manipulação de interrupções.

A operação de interrupção possui três fases: o pedido de interrupção, o reconhecimento de interrupção e o serviço de interrupções, e existem sete níveis de interrupções configuráveis pelo usuário (102).

- Sub-via de utilitários: são linhas de controle adicional que permitem a implementação de outras funções da via padrão, tais como, a comunicação serial, a iniciação do sistema, o diagnóstico da via, também possui as linhas de energia do sistema.

  Esta via é utilizada pelos seguinte modulos funcionais: o relógio, o módulo de potência e o módulo do relógio serial.

5.1.2.3 Considerações Elétricas e Mecânicas

As considerações elétricas permitirão garantir uma temporização apropriada dos sinais de controle, proteção contra ruídos electromagnéticos e redução do problema de acoplamento (Crosstalk) nas linhas que percorrem o bastidor.

As considerações mecânicas proporcionam informações referentes às dimensões dos módulos, bastidores, sub-bastidores, painel frontal e outros acessórios mecânicos envolvidos.

A referência (102), apresenta a especificação completa da via padrão IEEE P1014.

A caracterização da via teve como objetivo apresentar a via, o que complementa a especificação funcional do controlador e permitirá uma seleção mais exata dos sub-sistema que conformarão o controlador de célula.

5.1.3 Sub-sistemas do Controlador de Célula

Neste item são apresentados os sub-sistemas selecionados que implementarão as funções do controlador de acordo com a arquitetura proposta em 4.3.1.

Para a seleção dos sub-sistemas do controlador foram analisadas aqueles baseados na via de comunicação IEEE P1014 e aqueles que cumprem com a especificação funcional do controlador.

O objetivo inicial desta implementação é suportar a configuração mínima do controlador explicada em 3.2.1, deixando uma estrutura básica pronta para crescer harmoniosamente.

Dentro da seleção dos sub-sistemas é importante ressaltar dois detalhes, primeiro a tendência de parte dos fornecedores em obter um alto grau de integração nos sub-sistemas, por exemplo implementar um computador num módulo só (Single Board Computer), isto permitiu determinar as funções básicas do controlador neste tipo de módulo.

Segundo, um problema adicional encontrado nos sub-sistemas que utilizam o padrão IEEE P1014 é o grau de incompatibilidade entre certos fornecedores, problema que foi superado na atualidade com a utilização de circuitos integrados tipo ASIC que implementam o controlador da via do sistema, os sub-sistemas selecionados para a implementação do controlador utilizam o circuito integrado de maior tendência padrão como é visto em (99).

A seguir, são mostrados todos sub-sistema que implementam o controlador de

célula flexível.

5.1.3.1 Processador Central

A figura 5.3 apresenta o sub-sistema principal do controlador nomeado de processador central, que implementa as funções do sub-sistema Processador-Aplicação-Central mostrado no sub-item 4.3.2 do capítulo anterior, os módulos que constituem este sistema são:

- Unidade central de processamento: é o módulo encarregado da gerência e a execução das funções no sub-sistema.

  O elemento representativo deste módulo é o microprocessador 68040 de 32 bits, reconhecido como um microprocessador de alta velocidade, cujas característica principais estão na referência (104);

- Memória: para o armazenamento do sistema operacional e o software aplicativo, composto por um banco de memória principal, um de memória programável e um de memórias rápidas.

  O banco principal é implementado com 4 Mbytes de memórias dinâmicas (DRAMs), a arquitetura do sub-módulo de memória é baseada na organização entrelaçado (interleaved), que permite o acesso aos vários bancos ao mesmo tempo, otimizando o acesso seqüencial (105). A memória principal também é multiacesso, isto é, permite o acesso simultâneo de vários módulos e outros sub-sistemas.

  O banco de memórias programáveis (PROMS) suporta até 2 Mbytes e conterá o núcleo do sistema operacional, e finalmente o banco de 2 Kbytes de memórias

estáticas (SRAM), para uso geral, como por exemplo, configuração de portas seriais e armazenamento de dados referentes à configuração do controlador.

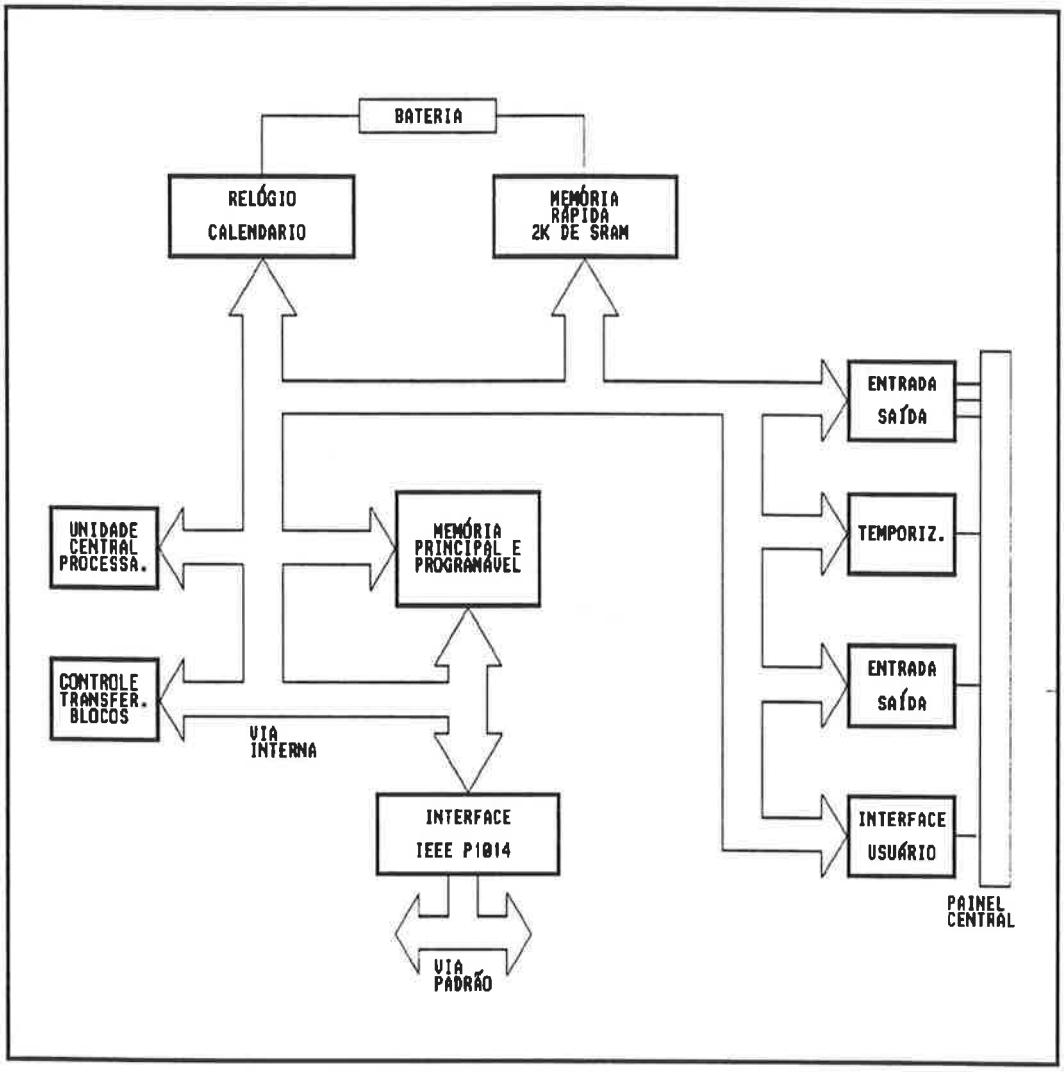

Figura 5.3 Sub-sistema PROCESSADOR CENTRAL

Cabe ressaltar que uma zona da memória principal é utilizada como caixa postal (mail box) para a transmissão de mensagens entre processadores (85).

- Controle de transferência de blocos: encarregado da transferência de pacotes de dados desde 8 até 1024 bytes numa operação simples, segundo o padrão definido na revisão D.1 do IEEE P1014 (97);

- Interface IEEE P1014: implementa o controlador da via do sistema definido no sub-item 5.1.2.1, este sub-sistema é do tipo mestre-escravo, pois permite trazer dados e ser chamado por outros sub-sistemas; ainda pode funcionar isoladamente quando os outros módulos do sub-sistema estão em falha;

- Entrada-saída: possui quatro portas seriais tipo assíncrona RS-232/432 ou RS-422, configuráveis para comunicação da console e da impressora que gera relatórios;

- Interface usuário: o painel frontal é implementado por diodos de estados que representam a falha do sistema, parada e funcionamento normal; e chaves de aborto, de reinício e de configuração;

- Temporizadores: constituído por um relógio-calendário, dois temporizadores programáveis e três contadores para uso geral.

5.1.3.2 Processador de Entrada e Saída

A figura 5.4 representa o processador de entrada e saída, implementando as funções do sub-sistema de Controle-Entrada-Saída e o sub-sistema de Coordenação da arquitetura do controlador apresentada em 4.3.2. A complexidade deste sub-sistema depende da complexidade da célula flexível e possui os seguinte módulos:

- Unidade central de processamento: é o coordenador geral e é implementado pelo microprocessador de propósito geral 68030, cujas caraterísticas principais estão em (106);

- Sistema de memória: este módulo representa o banco de dados real e suporta o

software deste sub-sistema, é implementado por 1 Mbytes de memórias dinâmicas DRAMS, por 1 Mbytes de memórias PROM e por 128 Kbytes de memórias estáticas;

- Interface IEEE P1014: possui os mesmos módulos do sub-sistema anterior, e também é do tipo mestre-escravo;

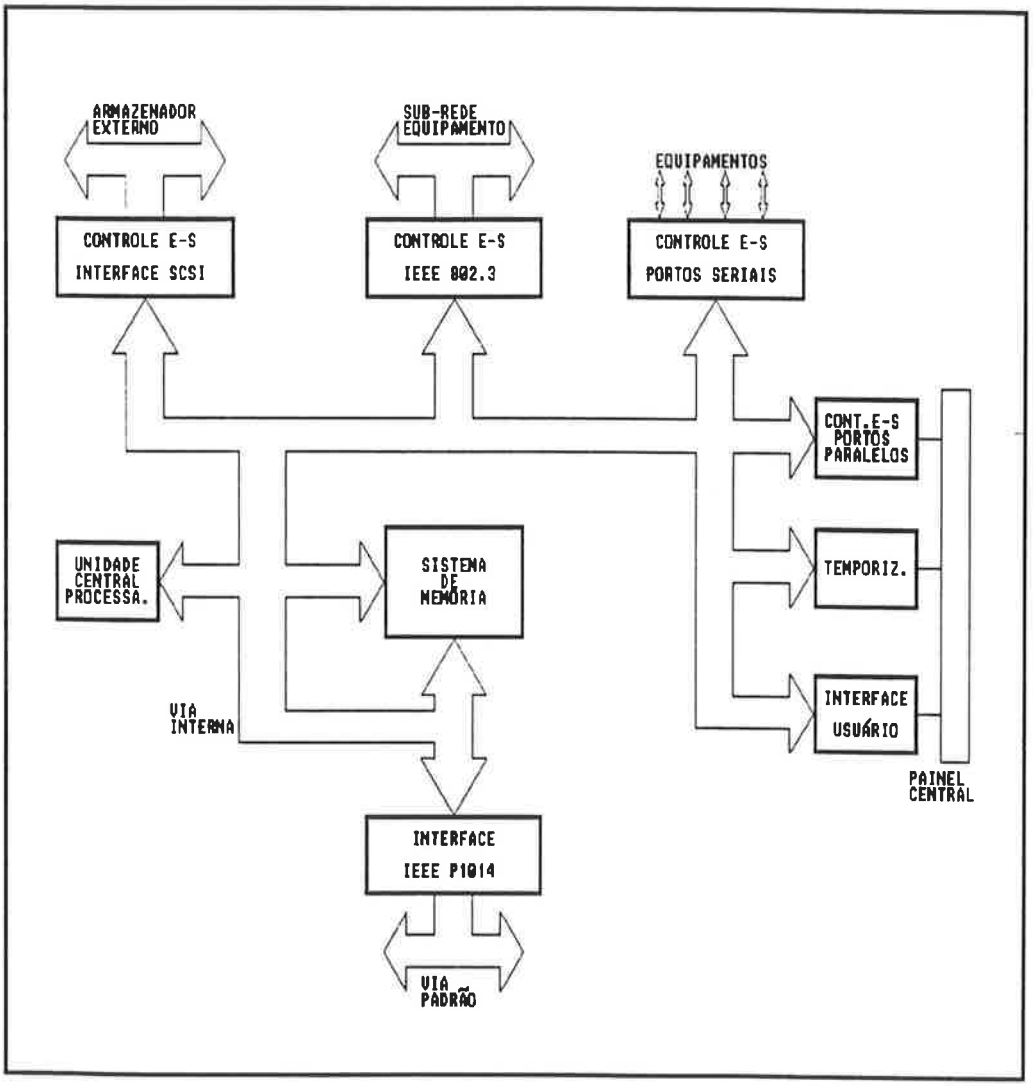

Figura 5.4 Sub-sistema PROCESSADOR ENTRADA E SAÍDA

- Controle de E-S: implementa uma interface SCSI (Small Computer System

Interface"), que é uma interface de comunicações para conectar dispositivos periféricos, como armazenadores externos e impressoras. Possui uma memória de 1 Kbyte como área de "reserva" de dados.

Implementa a interface IEEE 802.3 (5) com velocidade de transferência de 10 Mbytes/segundo, esta interface utiliza uma memória dinâmica tipo "reserva" de 64 kbytes para ser utilizada no tratamento do protocolo.

Possui quatro 4 portas seriais configuráveis RS 232/422 e 2 portas paralelas;

- Temporizadores e interface usuário: composto de temporizadores programáveis para uso geral e um relógio tipo calendário, e a interface usuário é implementada por indicadores de estado (LEDs) e chaves de "reinício".

Este sub-sistema pode ser configurado para suportar os principais protocolos da Via de Campo (Field Bus). O "Field Bus" é uma proposta em vias de padronização e que é implementada na sub-rede de instrumentação, para este caso são utilizadas as especificações padrões internacionais das vias propostas como a SP-50, o Profibus e o FIP (58), (107), (108).

5.1.3.3 Processador de Comunicação Exterior

A função principal deste sub-sistema é permitir a comunicação do controlador com os elementos das diferentes sub-redes do sistema aberto de automação apresentados no segundo capítulo. A figura 5.5 apresenta este sub-sistema de comunicação.

Neste sub-sistema são implementados os protocolos padrões MAP (Manufacturing Automation Protocol) e suas formas simplificadas como o Mini-MAP e

o EPA-MA (Enhance Performance MAP), todos estes são conjuntos de protocolos especificados para aplicações industriais baseados na proposta dos protocolos abertos da OSI (Open System Interconecction) (109), (110), (111), (112).

Aqui é considerado o documento da especificação do MAP versão 3.0 apresentada pela General Motors e referenciada em (113).

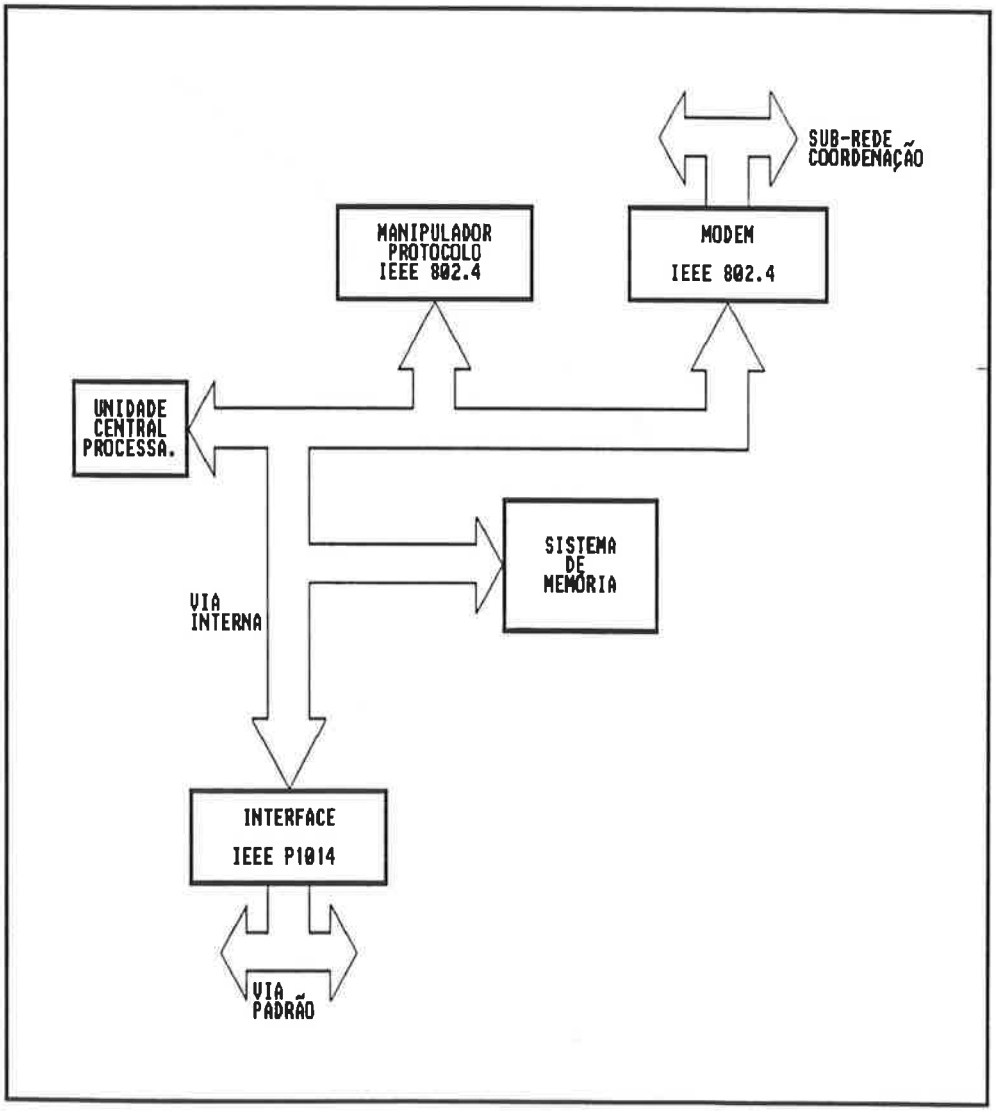

Figura 5.5 Sub-sistema de COMUNICAÇÃO EXTERIOR

O sub-sistema de comunicação é mostrado na figura 5.5 e os módulos que o compõem são os seguintes:

- Unidade central de processamento: utiliza o microprocessador 68020, com as características mostradas em (114);

- Sistema de memória: possui 1 Mbyte de memória dinâmica e 1 Mbyte de memória PROM, cujas funções são de manter a estrutura dos protocolos MAP e o núcleo do sistema operacional, respectivamente;

- Interface IEEE P1014: possui os elemento básicos, necessários para comunicar-se com toda a plataforma IEEE P1014 explicados nos sub-sistema anteriores;

- Interface IEEE 802.4 ("Token Bus" ou transmissão por ficha): composto por dois blocos, um manipulador de protocolo, encarregado de enviar os dados de acordo com a especificação do padrão IEEE 802.4; e um MODEM que se encarrega de implementar a interface fisica com a sub-rede exterior;

A figura 5.6 mostra os diferentes modos de conexão do controlador, na parte a da figura o controlador implementa os protocolos da arquitetura EPA-MAP para integrar as sub-redes de coordenação definidas com protocolos MAP e os protocolos da sub-rede de instrumentação definidas com protocolos MINI-MAP.

Na parte b da figura 5.6, o controlador é um elemento da sub-rede de instrumentação e para comunicar-se com a sub-rede de coordenação é necessário implementar equipamentos ou interfaces "ponte" que implementam o protoclo EPA-MAP.

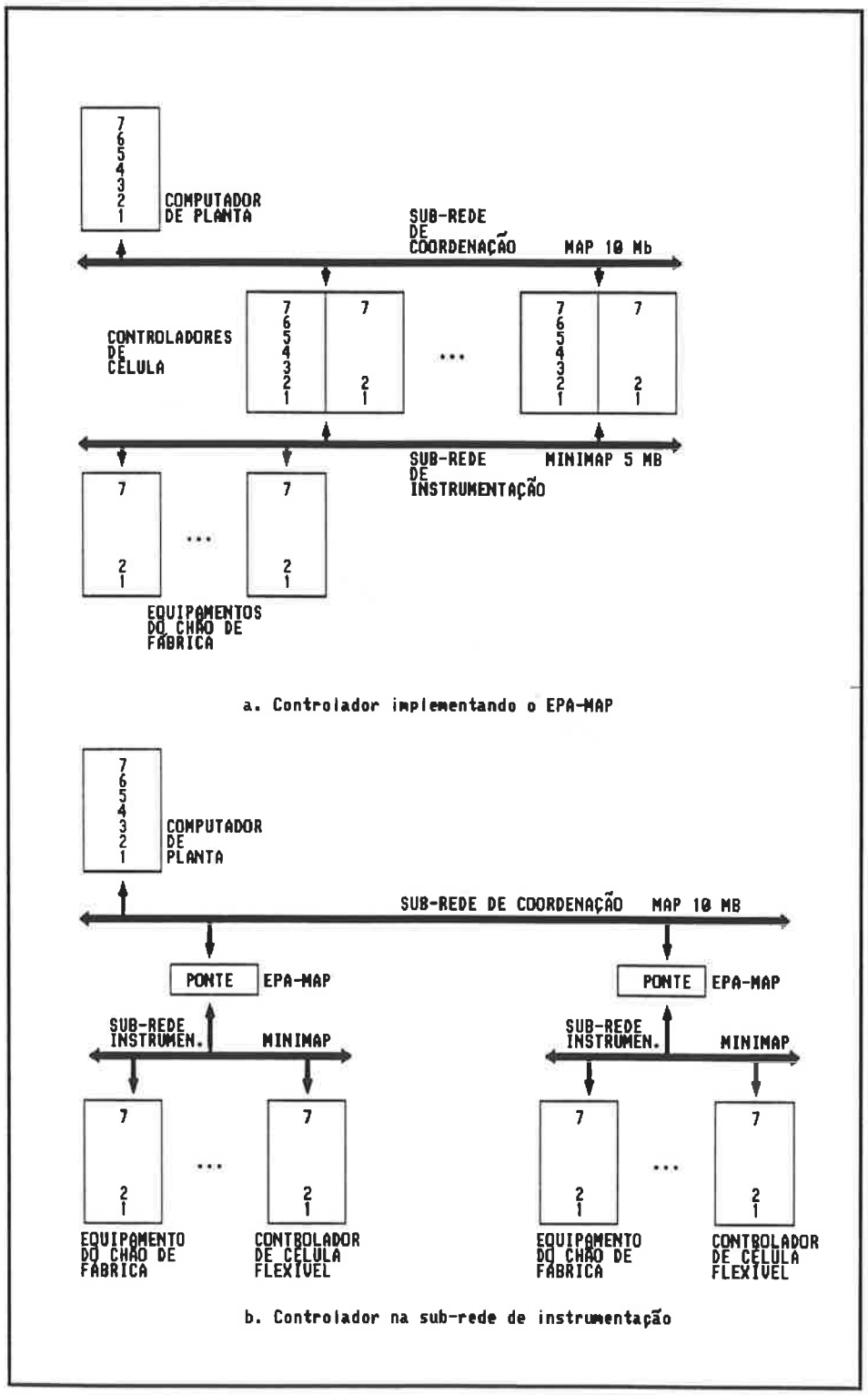

Figura 5.6 Modos de interconexão do controlador num sistema aberto de automação

5.1.3.4 Integração e Teste do Hardware

A integração do hardware do controlador consiste na interconexão dos sub-sistemas e os testes parciais das interfaces do sistema.

A primeira etapa desta fase é chamada de configuração dos módulos e está constituída pelas seguintes tarefas de configuração:

- Configuração da memória: consiste na estruturação das memórias de acordo a aplicação do sub-sistema;

- Configuração das portas de comunicação, como velocidade, paridade, largura de dados, bit de parada e modo de comunicação;

- Programação dos temporizadores, contadores e registros de identificação de mestre ou escravo;

- Configuração da interface com a via IEEE P1014, relativa à programação dos parâmetros e modos referentes ao arbitragem, interrupção e transferência de dados.

No controlador estas características foram configuradas dependendo do sub-sistema, mas em geral foram colocadas todos os mecanismos da transferência de dados, os sete níveis de interrupção e a arbitragem tipo prioridade apresentada no sub-item 5.1.2.2.

Todas as tarefas de configuração foram realizadas através de um ambiente de software, especialmente dedicado para a configuração dos sub-sistemas.

A segunda fase ou os testes são realizados no ambiente de software utilizado para a configuração dos sub-sistemas, são executados pequenos programas de testes, orientados aos testes das interfaces dos sub-sistemas, do controlador da via de comunicação, das portas e das interfaces de comunicação, estes programas basicamente possuem uma função simples, como por exemplo transferência de dados entre os sub-sistemas do controlador e entre o controlador e os equipamentos simulados fisicamente.

No item seguinte é apresentada a implementação do software do controlador de célula flexível, aspecto complementar do hardware do controlador.

5.2 Implementação do Software

A implementação do software do controlador possui os seguintes objetivos: a seleção e implementação do sistema operacional e a seleção dos aplicativos que representam as funções do controlador, como supervisão, controle, coordenação e interface homem-máquina.

O outro objetivo interessante da implementação do software é a formação de um ambiente de desenvolvimento, cujo objetivo principal é de permitir ao usuário realizar as diversas tarefas referentes ao software.

Estas tarefas abrangem a compilação e edição de programas utilizando linguagens padrões, a configuração do controlador e equipamentos, o gerenciamento de banco de dados, os programas de controle, os algoritmos de intertravamentos e diagnóstico, a interface homem máquina e todas as tarefas que possam facilitar o controle do chão de fábrica.

Estes módulos do software serão implementados visando as propostas padrões, e

em especial a proposta padrão IEEE 1003, nomeada de POSIX (Portable Operating System Interface), que a seguir é explicada.

5.2.1 Padronização POSIX

O POSIX é provavelmente o maior esforço de padronização no nível de software e impulsionado pela IEEE Technical Commitee on Operating Systems (TCOS) (115).

O objetivo principal do POSIX é compor um ambiente de desenvolvimento para um sistema aberto (Open System Environment) por isso que envolve dentro de sua estrutura, a definição de um conjunto de interfaces para a implementação deste tipo de ambiente.

Existem dezoito grupos de trabalho, identificados com a sigla IEEE 1003.X, cada grupo de trabalho é encarregado da especificação de um grupo de serviços padrões.

Estes grupos abrangem três áreas principais: a definição dos serviços e as características necessárias para implementar o ambiente aberto e conseguir a portabilidade das aplicações; a relação dos serviços com algumas linguagens específicas; e, finalmente, alguns grupos trabalham com tópicos mais abrangentes, como a segurança, ou em aplicações específicas, como a supercomputação.

Exemplos destes grupos de trabalho são apresentados a seguir, e em especial os dois principais utilizados nesta etapa de seleção:

- Grupo POSIX 1003.1: define as interfaces entre os programas de aplicação e o sistema operacional, toma como referência o sistema operacional UNIX e a linguagem de programação C;

- Grupo POSIX 1003.4: define os serviços com as características que exigem os sistemas de tempo real, que são apresentados em (47) e (116).

Este conjunto de padrões internacionais (POSIX), serão a base para a seleção dos módulos de software que são mostrados nos seguintes itens.

5.2.2 Sistema Operacional

O sistema operacional do controlador selecionado visa a definição apresentada no item 4.3.3 e os objetivos da implementação definidos.

O problema que se originou após a definição da arquitetura, no que se refere ao software, é a falta de um sistema operacional que cumpra com todas as características apresentadas em 4.3.3, pois os sistemas operacionais distribuídos e de tempo real ainda estão em etapa de desenvolvimento (64).

A solução para a implementação do sistema operacional consistiu na utilização de dois sistemas operacionais, um deles correspondente ao ambiente de desenvolvimento, e estará instalado no processador central do hardware do controlador; e o outro, correspondente ao sistema operacional de tempo real ou ambiente dedicado, que suporta as tarefas que estão mais perto do processo, ou seja, a supervisão, aquisição de dados, coordenação e que será instalado nos outros sub-sistemas.

A figura 5.7 apresenta a distribuição dos sistemas operacionais no hardware do controlador. O sistema operacional selecionado como ambiente de desenvolvimento, pelas características padrões que ele possui e por estar dentro do contexto POSIX, é o sistema operacional UNIX.

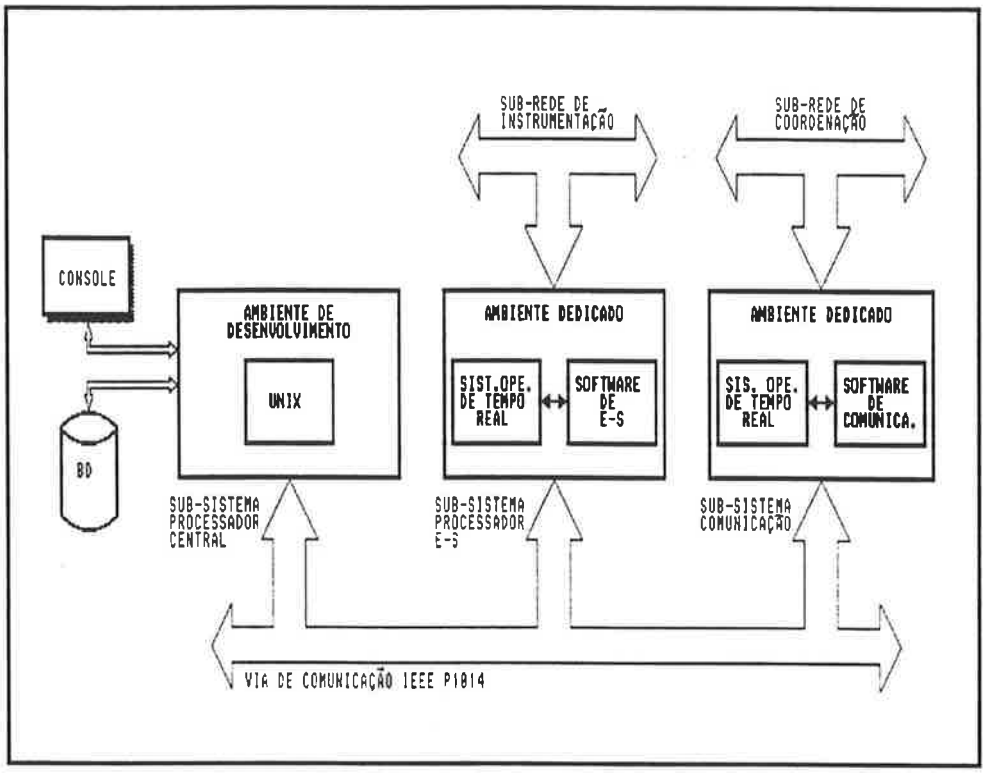

Figura 5.7 Distribuição dos sistemas operacionais no hardware do controlador

O UNIX é um sistema operacional multitarefa, multiusuário e suporta inúmeras aplicações, incluindo o controle de processos (117), (118).

O sistema operacional utilizado como ambiente dedicado e que se implementará nos sub-sistemas de controle de E/S, e de comunicação, é o sistema operacional de tempo real denominado LynxOS que é baseado na proposta POSIX/UNIX, e cujos módulos são explicadas a seguir (49),(87), (117):

- Módulo central ou "Núcleo" do sistema: constituído pelos sub-módulos, controle de tarefas, sincronismo, intercomunicação, controle de interrupções, gerenciador de memória e relógio de tempo real.

O sub-módulo de controle de tarefas se encarrega de implementar as estratégias de controle para a execução das diversas tarefas, como por exemplo a "programação de tarefas preemptivas baseado em prioridades", e mecanismos de suporte como o "controle de blocos" para o envio de dados (49).

O sub-módulo de sincronismo de tarefas é encarregado de implementar os mecanismos de sincronização entre tarefas concorrentes, como os semáforos, que são variáveis que armazenam o estado de execução das tarefas.

O sub-módulo de intercomunicação entre tarefas que permite a troca de informações entre tarefas, implementando mecanismos apropriados como, caixas postais e área de reserva de memórias (119).

O sub-módulo de controle de interrupção implementa os mecanismos e o gerenciamento de interrupções. O sub-módulo gerenciador de memória, implementa os mecanismos de alocação de memória para uso específico; e finalmente o sub-módulo de relógio de tempo real é utilizado especialmente pelo sub-módulo de controle de tarefas na temporização das tarefas;

- Módulo de serviços: é o módulo que executa as funções do sistema operacional, através de primitivas, que são comandos próprios do sistema, facilitando a interação do usuário com os elementos do hardware.

Os principais serviços são: o manipulador de arquivos, que executa operações próprias dos arquivos (chamar, apagar, armazenar, carregar); a interface de dispositivos, que manipula os dispositivos de entrada e saída; o carregador, que permite a transferência do sistema operacional e das aplicações, de dispositivos externos para a plataforma de hardware;

- Módulo interface aplicação: implementa a interface entre o usuário e os outros módulos do sistema operacional, é composto pelo interpretador de comandos, pelo compilador de linguagens, e por outros utilitários como o ambiente de teste, e o editor.

Finalmente, a comunicação entre os sistemas operacionais, o ambiente de desenvolvimento e o ambiente dedicado é realizada através de interfaces de software de comunicação apropriadas, utilizando a via de comunicação padrão do hardware do sistema.

5.2.3 Software Aplicativo

O software aplicativo do controlador implementa as funções do controlador como, a aquisição de dados, supervisão, coordenação e a apresentação de interfaces gráficas e relatórios.

A execução destas funções está fortemente relacionada com o gerenciador de banco de dados, pois este módulo será a interface entre os processos e os aplicativos (119).

A figura 5.8 apresenta a estrutura geral do software aplicativo do controlador, onde os módulos que representam uma função do controlador são executados concorrentemente, sendo o sistema operacional a "via de comunicação" entre os módulos, e o "protocolo de comunicação" estará implementado com as primitivas do sistema operacional utilizando a técnica de envio de mensagens (message passing) (119).

A seguir são explicados os módulos do software aplicativo:

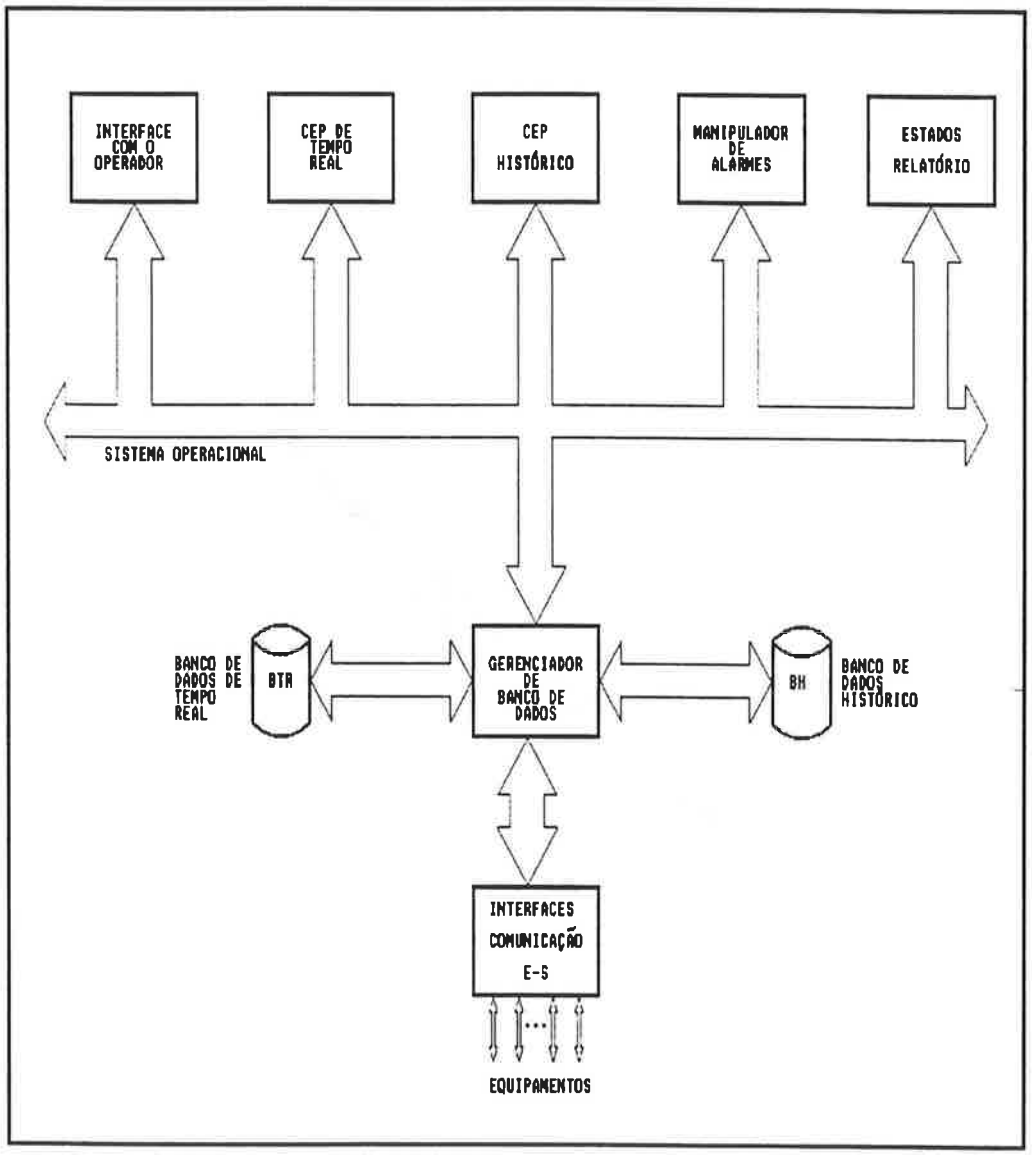

Figura 5.8 Estrutura do software aplicativo

- Gerenciador do banco de dados: é o módulo principal do software aplicativo, lê os dados coletados nas tabelas de comunicação implementadas nos sub-sistemas de aquisição; realiza as conversões de unidades e as armazena nos bancos de dados respectivos; envia os dados aos outros módulos do software; e escreve dados nos respectivos dispositivos;

- Banco tempo real: implementado pela memória dos processadores de entrada e saída e a memória do processador central; armazena os dados rescentemente coletados;

- Banco histórico: implementado pelos dispositivos de armazenamento externo, onde são armazenados o histórico de todos os valores coletados;

- Controle estatístico de processos (CEP): representado na figura 5.8 pelos módulos, CEP histórico e o CEP de tempo real, são os encarregados de implementar os algoritmos referentes ao controle estatístico de processos, estes módulos utilizam três elementos para poder executar essa função: os dados históricos, os dados de seguimento real e as interfaces gráficas;

- Interfaces com o operador: são todas as interfaces gráficas relacionadas com os outros módulos, como por exemplo: os gráficos de processo, os gráficos-tabelas e histogramas do CEP, as tabelas de configuração e os avisos de alarmes.

Neste módulo cabe ressaltar a utilização do recurso gráfico padrão "de facto" "X Window", que é um sistema de janelas baseado numa rede "transparente", que é um software que permite ao usuário acessar as aplicações que estão localizadas em outros processadores, e executá-las numa "janela" só, como se elas fossem independentes (6).

Os esforços de padronização, no que diz respeito às interfaces gráficas, vem sendo apresentadas através das propostas OSF/MOTIF (Open Software Foundation) e da Open Look, sendo as duas baseadas na proposta "X Window" (6).

- Manipulador de alarmes: recebe um sinal de alarme do gerenciador de banco de dados, analisa e reporta ao operador através das interfaces apropriadas;

- Estados-relatório: encarregado de apresentar os relatórios com dados referentes à produção, utiliza os dados dos bancos reais e histórico.

- Interface de comunicação E-S: implementa os programas de coleta e coordenação do controlador; os dados são coletados em forma de tabelas cada vez que o gerenciador de banco de dados gera uma interrupção-pedido a esta interface de E-S (119).

Finalmente para complementar as operações do ambiente de desenvolvimento, foi seleccionado o "C" como a linguagem para as atividades de programação, assim como, foi determinada a utilização do Grafcet como linguagem de controle dos equipamentos e processos (120).

5.2.4 Integração e Teste do Software

Esta fase abrange a integração e teste dos módulos do software do controlador. A estratégia compreende três fases que são explicadas a seguir:

- Implantação do sistema operacional nos módulos de hardware e teste das principais funções do sistema operacional utilizando programas de teste, como por exemplo, programas que enviam mensagens, blocos de dados, ou executem testes de

sincronismo, como são explicados em (87);

- Implementação do gerenciador de banco de dados e armazenamento de dados de testes no banco histórico, que são dados experimentais devidamente selecionados para esta tarefa;

- Implementação de cada aplicativo nos subsistemas do controlador e teste dos mesmos utilizando os dados de "testes" anteriores, conjuntamente são testadas as funções do gerenciador de banco de dados;

- Teste das funções em tempo real do gerenciador: nesta tarefa, os pontos de coleta e as portas do processador E/S, são conectados aos sensores que estarão coletando informações de um processo simulado, os dados serão armazenados no banco de tempo real, e em seguida utilizados pelos respectivos aplicativos.

Para esta fase são implementadas interfaces gráficas adequadas para a apresentação dos dados na console, utilizando as ferramentas computacionais, como o Windows, que o ambiente de desenvolvimento possui.

No item seguinte é apresentada a fase final do projeto do controlador de célula flexível denominada de validação.

5.3 Validação do Controlador

Nesta última fase do projeto, serão apresentadas as duas estrategias de validação do controlador, cujo objetivo principal é demostrar a operação do controlador e a conformidade dos requisitos funcionais e do desempenho de acordo com a especificação funcional.

A primeira estratégia consiste do plano de teste utilizando um simulador dos elementos da célula flexível, o qual será implementado num computador, nesta parte será apresentado o projeto do simulador.

A segunda estratégia corresponde à simulação da arquitetura multiprocessador do controlador utilizando um ambiente apropriado de simulação, de forma a obter resultados do funcionamento, que possam respaldar as decisões de implementação apresentadas nos item anteriores.

5.3.1 Simulador dos Elementos da Célula Flexível

A operação e funcionalidade do controlador poderá ser comprovada se este fosse conectado aos elementos da célula flexível através dos módulos de E/S.

Montar uma célula flexível real apenas para validar o controlador implementado é uma decisão não viável, pois a aquisição dos elementos da célula tornaria o custo do projeto astronômico.

Ademais, considerando que o controlador será utilizado como elemento de um laboratório, e que seria impossível equipar cada controlador com sua respectiva célula, é necessário conseguir uma solução prática, que possibilite a validação do controlador e a substituição dos elementos da célula no laboratório.

O desenvolvimento de um software de simulação dos elementos da célula é a solução apropriada. A seguir, é apresentado um resumo do plano de projeto do software de simulação dos elementos da célula flexível e em seguida, os elementos da plataforma de teste que suportam esta forma de validação.

5.3.1.1 Projeto do simulador

O simulador é uma continuação dos trabalhos realizados nesta área, e que estão referenciados na bibliografia como (121), (122), (123), (124), (125), e possuem as seguintes funções:

- Definição dos modelos gráficos: permite que o usuário configure as características próprias do modelo gráfico utilizado para cada equipamento, por exemplo, para a modelagem de um robô são armazenados os parâmetros da fórmula da cinemática, e que posteriormente serão utilizados na matriz de transformação de coordenadas, para a definição do movimento na tela (42);

- Apresentação do modelo na tela: o objeto é apresentado utilizando a técnica da projeção, o qual apresenta as características físicas do objeto em um espaço de três eixos, como se este fosse visto de um ponto (x,y,z), denominado ponto de visão (42);

- Simulação do movimento: utilizando os modelos gráficos, e as matrizes de transformação, são executados algoritmos de transformação, cujos resultados atualizam a posição do objeto na tela;

- Elaboração e execução de programas: deve possuir um ambiente que permita a criação de programas utilizando comandos simples, como por exemplo, Mover a (X,Y,Z).

- Interface com o controlador: esta função permite a troca das mensagens entre o controlador e o hardware hospedeiro do simulador, respondendo como se este fosse

o controlador dos equipamentos;

- Restrições: em princípio será projetado para simular dois equipamentos, um robô e um CNC, e será projetado para ser utilizado num microcomputador, com sistema operacional DOS.

Outra restrição importante é com referência ao dicionário de comandos, possuirá um conjunto de comandos básicos, envolvendo a realização de movimentos e processos simples, como translação de um bloco retangular, furar num ponto, trajetórias simples de movimento de robô e as referentes às ferramentas do CNC.

Para entender melhor o funcionamento do software simulador, é apresentado na figura 5.9 o diagrama de fluxo de dados básico.

Os comandos de controle enviados pelo usuário são validados no módulo VÁLIDA-PEDIDO e interpretados no PROCESSA-PEDIDO que os encaminha para as tarefas seguintes a executar.

O CONFIGURA-OBJETO encarrega-se de armazenar todos os dados referentes ao objeto (dado-objeto), como por exemplo os dados da fórmula cinemática, dimensões e, também, forma os vetores de transformação (vetor-transforma), utilizados na simulação do movimento.

O ESTRUTURA-PROGRAMA permite a criação de programas e encarrega a interpretação do programa ao INTERPRETA-COMANDO, que decodifica o comando e envia a ordem de execução ao EXECUTA-TRANSFORMAÇÃO, que em conjunto com o TRANSFORMA-GRÁFICO, executam os algoritmos computacionais para atualizar a posição do objeto e converter esta posição em vetores gráficos, respectivamente,

estrutura que é apresentada na tela do microcomputador.

Finalmente os módulos RECEBE-COMANDO, ENVIA-COMANDO E MANIPULADOR DE COMUNICAÇÃO encarregam-se de ligar o simulador com o controlador de célula flexível.

No item seguinte será apresentada a plataforma física ou de teste, que completa a metodologia de teste do controlador.

5.3.1.2 Plataforma de Teste

A plataforma de teste é composta por um microcomputador PC-AT, 386-SX, 40 Mhz, com coprocessador matemático e recursos gráficos SVGA, este equipamento é o encarregado de suportar o software simulador dos elementos da célula.

A conexão do controlador com o microcomputador "célula" é feita utilizando a interface serial do microcomputador e uma porta, também serial, do módulo processador inteligente E/S do controlador, o controlador executará o programa de coordenação de teste, que deverá incluir o programa de intercomunicação com o microcomputador.

5.3.2 Simulação da Arquitetura do Controlador

O objetivo da simulação é de antecipar como a proposta de arquitetura do controlador comportar-se-á antes de ser implementada fisicamente, estas primeiras avaliações de desempenho permitirão complementar e respaldar as decisões das etapas anteriores à fase de implementação.

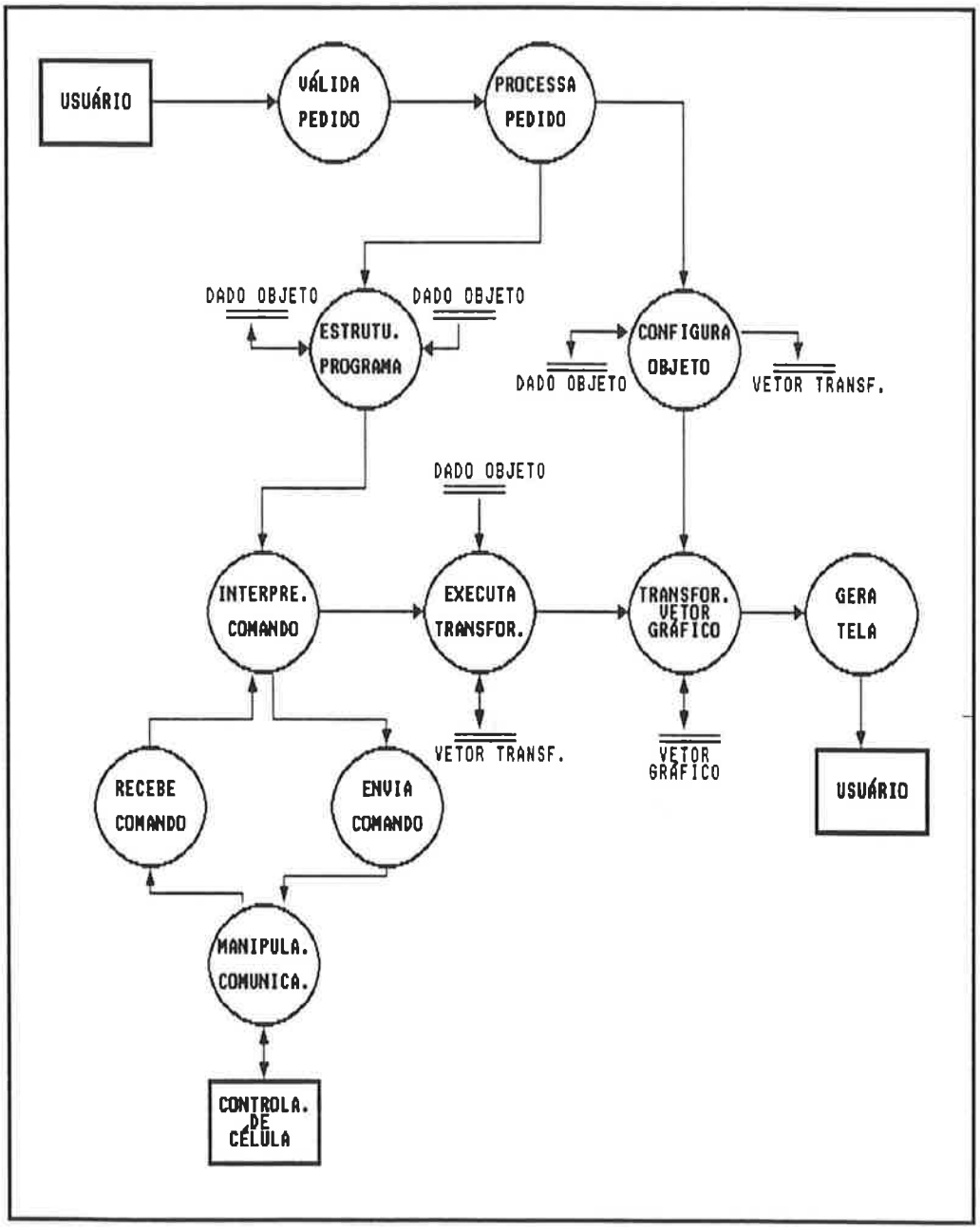

Figura 5.9 Diagrama de fluxo de dados do simulador

5.3.2.1 Estratégia de Simulação

A primeira parte da simulação consiste na definição de um modelo que represente o comportamento da arquitetura do controlador.

O modelo que a seguir será apresentado é baseado na teoria de filas, sendo este tipo de modelo o mais aplicado na simulação de arquiteturas de computadores, como pode ser visto em (127).

O modelo da arquitetura do controlador é mostrado na figura 5.10, e segue a notação gráfica da referência (128), assim, as filas e os circulos representam os algoritmos de utilização do recurso e os processos referentes aos recursos do sistema, respectivamente, estes dois desenhos são denominados de centros de serviços; e as flechas representam o fluxo das transações ou mensagens dentro do sistema.

Os três centros de serviço P1, P2, P3 representam os processadores do controlador de célula flexível; o quarto centro de serviço chamado de via, representa a via de comunicação IEEE P1014 e o algoritmo para o controle de acesso à via; os triângulos indicam que a via pode ser utilizada exclusivamente por um processador.

A seguir são apresentados as principais características do modelo de fila do controlador:

- Cada processador é considerado como uma "caixa preta", onde se executa uma tarefa independentemente dos outros processadores;

- Os processadores representados neste modelo são: o processador central, o processador de aquisição e o processador de comunicação superior da arquitetura do

controlador;

- A troca de informações entre processadores é realizada através de uma via cujo controle de acesso é gerenciado por um centro de serviço;

- Só o processador que foi determinado pelo algoritmo de acesso pode enviar e receber dados pela via;

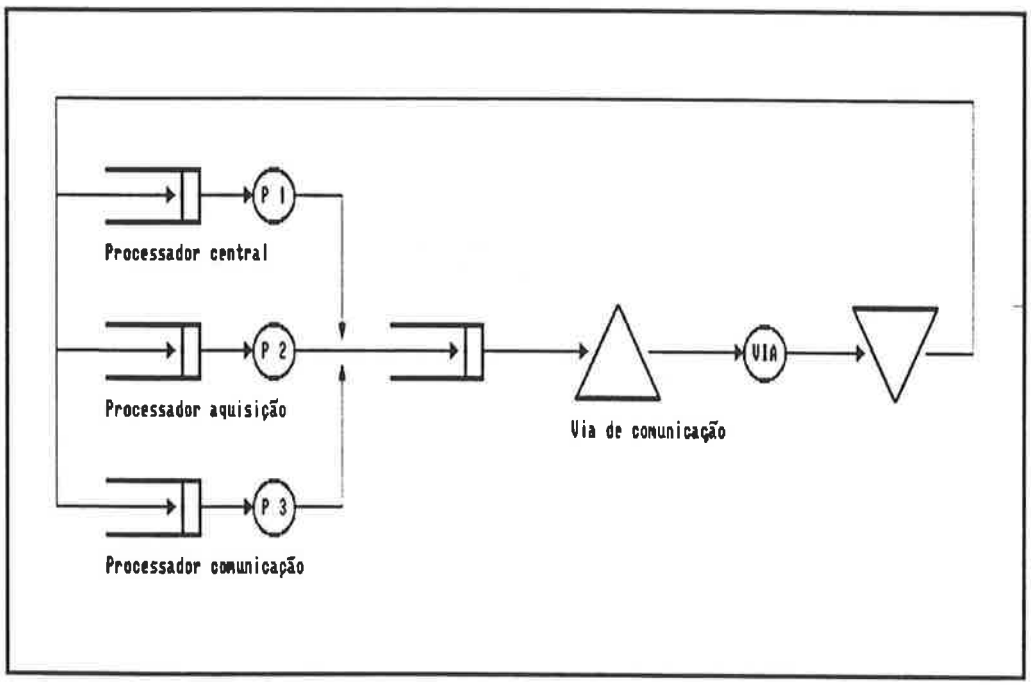

Figura 5.10 Modelo da arquitetura do controlador

- Existe um algoritmo de prioridades para o acesso da via, sendo as transações do processador de aquisição as de maior prioridade, seguida das transações do processador central e, finalmente, das transações do processador de comunicação; e

- Os tempos de chegada das transações para o pedido de utilização da via, são considerados como distribuições aleatórias.

Após definido o modelo, é mostrado o algoritmo de experimentação, cujo objetivo principal é exercitar o modelo, simulando o comportamento da arquitetura do controlador.

Neste algoritmo são consideradas como variáveis de entrada: as mensagens geradas aleatoriamente pelos processadores e a velocidade de transmissão da via de comunicação, sendo as saídas representadas pelas variáveis estatísticas, referentes ao tempo de espera na fila da via, ao tempo de transmissão e à taxa de utilização da via.

A figura 5.11 mostra o diagrama de fluxo do algoritmo de experimentação do modelo do controlador. No primeiro bloco define-se os geradores aleatórios independentes de mensagens dos processadores; o segundo bloco executa o algoritmo de acesso à via, ressaltando que é considerado o mecanismo de interrupções como prioritário; e finalmente, no terceiro bloco é simulado a transmissão das mensagens através da via.

O tempo de amostragem do experimento deve ser variável, e como resultados finais espera-se obter um relatório com as principais variáveis de desempenho do controlador, que servirão para um análise posterior.

No item seguinte será apresentada a implementação do modelo do controlador e do algoritmo de experimentação utilizando uma linguagem apropriada de simulação, assim como a análise dos resultados.

5.3.2.1 Implementação e Resultados

A implementação do modelo da arquitetura do controlador e do algoritmo de

experimentação foram realizadas utilizando um ambiente de simulação denominada GPSS (General Purpose Simulation System) (129).

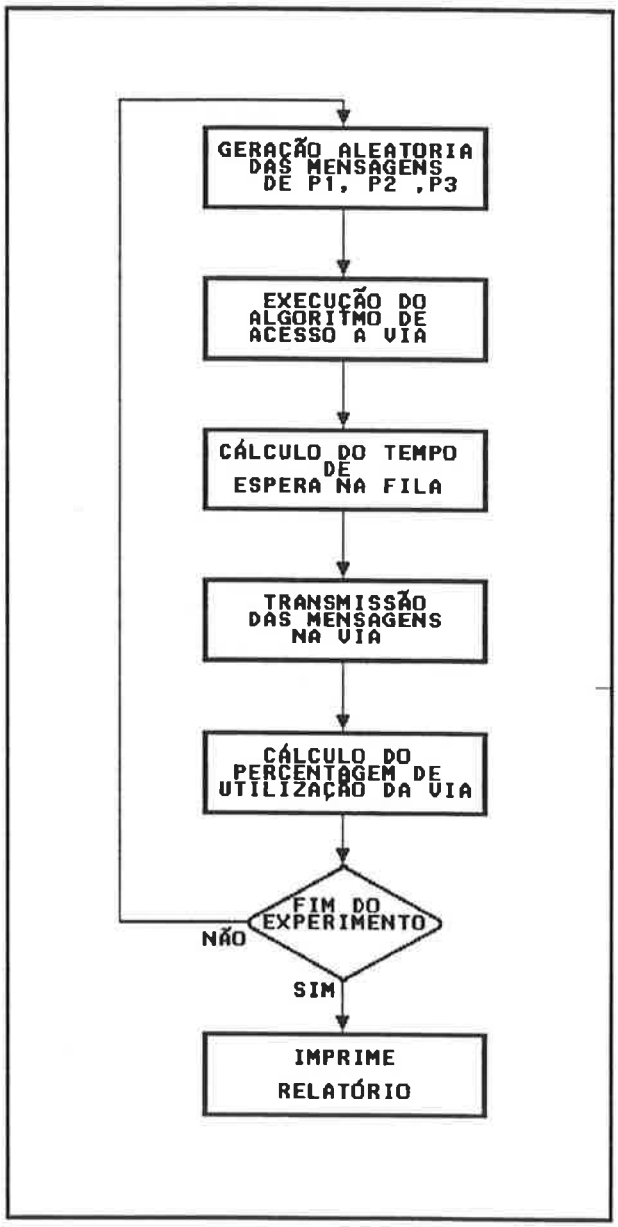

Figura 5.11 Diagrama de fluxo do algoritmo de experimentação

O GPSS é um ambiente que contém uma linguagem de simulação, onde os modelos são implementados pela combinação de um conjunto de blocos padrão, os quais definem a estrutura lógica do sistema.

Cada bloco definido em GPSS representa uma função determinada do sistema, o GPSS possui quase 50 destes blocos padrão e cerca de 100 comandos que possibilitam a programação nesse ambiente.

Uma característica importante desta ferramenta é que realiza a análise estatística das variáveis que estão envolvidas em cada bloco padrão, e apresenta uma série de relatórios, tabelas e gráficos que são de grande utilidade na análise de resultados.

A referência (129) e (130) apresenta uma descrição mais completa desta ferramenta, assim como, exemplos de programas realizados em GPSS.

Utilizando este ambiente GPSS, é elaborado o programa que implementa o modelo e o algoritmo de experimentação descrito no sub-item anterior que simula o controlador, este programa possui as seguintes características:

- Cada processador gera mensagens aleatoriamente de acordo a uma distribuição exponencial, esta função foi implementada pelo bloco GENERATE do GPSS;

- O controle de acesso à via foi implementado utilizando os blocos QUEUE e DEPART, que implementam a fila de espera, e os blocos SEIZE e RELEASE que implementam a exclusividade de utilização da via;

- O algoritmo de prioridade foi implementado utilizando o bloco PREEMPT, que permite colocar graus de prioridade em cada mensagens gerada pelos processadores;

- A transmissão das mensagens foi simulada com a utilização do bloco ADVANCE, que permite colocar a quantidade de mensagens gerada;

- Os resultados são armazenados em cada variável envolvida com cada bloco, e a apresentação dos relatórios está baseada nestas variáveis.

A execução do programa representa a experimentação do modelo proposto, sendo que a estratégia do experimento segue as seguintes considerações:

- O objetivo principal do experimento é de gerar diferentes situações de tráfego na via de comunicação, isto pode ser feito com a variação do tempo de geração e a largura das mensagens geradas pelos processadores;

- As variáveis de entrada, correspondentes ao tempo de geração das mensagens e a largura das mensagens, foram alteradas em cada experimento através dos parâmetros de entrada dos blocos GENERATE e ADVANCE;

- A freqüência de variação das variáveis de entrada estão mostradas na tabela 5.1, por exemplo para um experimento determinado foram utilizados a geração das mensagens do tipo 1 e a largura de dados da classe 3;

	GERAÇÃO DAS MENSAGENS			LARGURA DAS MENSAGENS (bytes)		
	TIPO 1	TIPO 2	TIPO 3	CLASSE 1	CLASSE 2	CLASSE 3
PROCESSADOR CENTRAL	20 ms	14 ms	20ms	1k-10k	>10k	>10k
PROCESSADOR AQUISIÇÃO	10 ms	16 ms	1 ms	<1k	=10k	<30k
PROCESSADOR COMUNICAÇÃO	50 ms	1 ms	50 ms	>10k	>10k	>10k

Tabela 5.1 Freqüência de variação no experimento

- A velocidade de transmissão da via foi considerada como constante e foi utilizado o valor característico de 10 Mbytes/s.

Finalmente, os resultados, após executado o programa de simulação, foram coletados nas variáveis de saída: tempo de espera na fila para cada processador, e a porcentagem de utilização da via.

A tabela 5.2 apresenta um conjunto de resultados para um experimento com geração de mensagens do tipo 1 e com largura das mensagens da classe 3, segundo a tabela 5.1. Nesta tabela pode-se observar os seguintes resultados: o tempo médio de espera (Tme) de cada mensagem enviada para cada processador e a porcentagem de utilização da via para cada tabela.

Pode-se ver também que a diminuição da largura das mensagens para cada tipo de processador permitiu a melhora dos tempo de espera e a diminuição do percentagem de utilização da via, este mesmo fenômeno aconteceu também nos outros experimentos.

A tabela 5.3 apresenta os tempos de resposta para um caso típico de funcionamento do controlador, neste experimento os processadores de aquisição e o central manipulam mensagens curtas (<1 kbits) com freqüência de geração bastante próxima, e com o processador de comunicação transmitindo mensagens menores que 5 kbits e numa freqüência de geração dez vezes maior que os processadores anteriores.

As conclusões obtidas dos experimentos são as seguintes:

- O controlador responderá na manipulação de transações de tempo real, pois os resultados dos experimento para este caso estiveram dentro da faixa típica de valores, mesmo nos experimentos críticos, quando a via ficava com maior sobrecarga;

- Os resultados na operação que se simula o funcionamento típico do controlador e onde existe a mistura de transações de tempo real e transações com os níveis superiores, são utilizados para concluir que o controlador proposto atendem aos requisitos das aplicações;

	UTILIZAÇÃO DA VIA = 79.8 %	
	MENSAGENS	TEMPO DE ESPERA
AQUISIÇÃO	50 kbytes	8.55 ms
CENTRAL	30 kbytes	2.09 ms
COMUNICA.	90 kbytes	26.00 ms

	UTILIZAÇÃO DA VIA = 54.9 %	
	MENSAGENS	TEMPO DE ESPERA
AQUISIÇÃO	50 kbytes	1.94 ms
CENTRAL	10 kbytes	0.45 ms
COMUNICA.	90 kbytes	10.95 ms

	UTILIZAÇÃO DA VIA = 34.3 %	
	MENSAGENS	TEMPO DE ESPERA
AQUISIÇÃO	30 kbytes	0.58 ms
CENTRAL	30 kbytes	0.10 ms
COMUNICA.	30 kbytes	0.90 ms

	UTILIZAÇÃO DA VIA = 26.6 %	
	MENSAGENS	TEMPO DE ESPERA
AQUISIÇÃO	10 kbytes	0.15 ms
CENTRAL	10 kbytes	0.12 ms
COMUNICA.	10 kbytes	1.00 ms

	UTILIZAÇÃO DA VIA = 22.9 %	
	MENSAGENS	TEMPO DE ESPERA
AQUISIÇÃO	20 kbytes	0.20 ms
CENTRAL	10 kbytes	0.19 ms
COMUNICA.	20 kbytes	0.43 ms

Tabela 5.2 Resultados num experimento

- Todos os resultados dos experimentos foram obtidos considerando a velocidade de transmissão da via de 10 Mbytes/s., porém atualmente na atualidade o padrão IEEE

P1014 alcança velocidades de 30 Mbytes/s., assim a implementação do controlador com sub-sistemas que atingem esta velocidade, permitirão obter melhores tempos de resposta e permitirá a expansão do controlador para perfis funcionais mais complexos.

	64 bits	1 kbits	10 kbits
AQUISIÇÃO	0.86 us	13.62 us	-
CENTRAL	0.9 us	15.3 us	1.4 ms
COMUNICAÇÃO	1.3 us	23.5 us	2.5 ms

Tabela 5.3 Tempos de resposta típicos no controlador

A simulação permitiu conhecer o comportamento da arquitetura do controlador antes da implementação física, e os resultados obtidos permitem a validação da proposta do controlador de célula flexível.

Neste capítulo foram apresentadas as fases de implementação e validação do projeto do controlador de célula flexível, completando o desenvolvimento da dissertação, já no capítulo seguinte serão apresentadas as considerações finais desta dissertação.

6 CONSIDERAÇÕES FINAIS

Neste último capítulo são apresentadas as considerações finais do trabalho que abrangem quatro itens, no primeiro apresenta-se as conclusões gerais da dissertação, para a seguir apresentar alguns comentários gerais respeito do projeto, para em seguida mostrar algumas propostas de temas de pesquisa, que se originaram ao longo da elaboração da dissertação e que darão continuidade ao trabalho desenvolvido, e, finalmente, comenta-se a contribuição acadêmica do trabalho.

6.1 Conclusões

As conclusões referem-se a quatro pontos importantes, a utilização dos sistemas abertos de automação, a utilização de arquiteturas abertas no projeto de equipamentos de automação, a utilização de metodologias de desenvolvimento de projeto adequadas e a integração do chão de fábrica.

Definitivamente, a concepção dos sistemas abertos deve ser utilizada nos projetos de sistemas de automação, pois as vantagens que eles oferecem assegurarão o sucesso do projeto e conseguirão a satisfação do usuário final.

O crescimento harmonioso, a adaptação a diversas formas de aplicações de modo simples, a aceitação de novas tecnologias, que assegurem a atualização do sistema, e a independência do usuário com respeito aos fornecedores de recursos de automação, são benefícios que tornam os sistemas abertos de automação como uma das estruturas mais importantes dentro dos sistemas modernos de automação.

Uma das decisões de projeto mais importantes colocadas nesta dissertação foi a definição da arquitetura do controlador como uma arquitetura aberta.

A arquitetura aberta permitirá que o controlador fique com um alto grau de flexibilidade, o que permitirá atingir uns dos objetivos colocados ao início da dissertação: a adaptação do controlador a diversas estruturas de células flexíveis de forma simples.

A implementação do controlador com arquitetura aberta é a forma mais adequada de implementação dos controladores de célula e que está dentro da concepção moderna dos projetos de equipamentos de automação, isto é, obter grande conectividade, interoperabilidade e flexibilidade, deixando como forma inadequada de implementação, aquelas de orientação específicas e rígidas.

É importante ressaltar que o controlador implementado com arquitetura aberta permitirá sua inclusão rápida e a baixo custo como elemento de um sistema aberto de automação, que foi uns dos objetivos desta dissertação.

Outro ponto importante nestas conclusões é a utilização de uma metodologia de desenvolvimento, ficou demostrado que a utilização deste tipo de metodologia permite atingir facilmente os objetivos de cada fase, e permite a passagem de uma fase a outra como se fosse um fluxo contínuo, podendo atingir um tempo de desenvolvimento curto e uma redução e detecção antecipada dos erros de projeto, de forma a obter uma boa qualidade do produto.

A aplicação da análise estruturada converteu-se numa ferramenta importante dentro do projeto e em conjunto com a especificação funcional informal, mas padrão, permitiu obter de forma clara e precisa a definição da arquitetura do controlador, e posteriormente conseguir a implementação como uma conseqüência lógica de todas as fases da metodologia de projeto.

Finalmente, a implementação do controlador como um equipamento de automação de estrutura aberta foi a solução de integração do chão de fábrica com todos os níveis do sistema de automação da manufatura, visando a constituição de um sistema de informação, estrutura que permitirá o fluxo contínuo da informação em toda a fábrica e conseqüentemente a introdução do CIM.

6.2 Comentários

Um comentário importante no projeto do controlador que tem haver com a metodologia de desenvolvimento, foi a necessidade de utilização de ferramentas computacionais tipo CASE (Computer Aided Software Engineering) nas diferentes fases do projeto, o que permitiu atingir uma maior produtividade no desenvolvimento do projeto. Nesta dissertação foram utilizadas duas ferramentas, como é mostrado no capítulo quatro e no capítulo cinco.

Um comentário ressaltante na implementação do controlador está ao redor dos recursos de hardware e software de tendência padrão, que consolidaram a decisão de adotar uma arquitetura aberta.

Foram várias as alternativas encontradas nos diversos elementos constitutivos da arquitetura do controlador, pelo que foi necessário sistematizar os critérios de seleção, tanto técnicos como os comerciais, mas sempre atendendo os padrões internacionais.

No hardware a chave foi a seleção da via de comunicação padrão como plataforma para os demais elementos, foram analisados vários critérios chegando à adoção do padrão IEEE P1014.

No software a decisão foi feita considerando a proposta de padronização

internacional chamada POSIX e na utilização de um sistema operacional padrão "de facto" como o UNIX, permitindo a seleção do software aplicativo de forma eficaz.

Não se pode deixar de comentar os protocolos padrões internacionais considerados, como o MAP que passa por uma fase de "congelamento" na sua especificação, despertando um maior interesse na utilização dentro dos sistemas de automação, e o "Field Bus" aplicado no nível de instrumentação, mas cuja padronização internacional ainda está em andamento.

Outro comentário importante é no que se refere à adquisição de recursos no mercado nacional, em todos os casos foi necessário a importação de recursos para implementação do controlador, e foi encontrado dentro da indústria local uma falta de alternativas para o hardware e para o software, ressaltando que os diferentes fornecedores estão preocupados com soluções de automação em torno do microcomputador PC-AT, sem enxergar as outras alternativas como as que foram utilizadas nesta dissertação.

6.3 Continuidade do trabalho

Ao longo da dissertação foram encontrados diversos aspectos importantes das diferentes áreas de atuação atingidas e que no futuro podem converter-se em trabalhos acadêmicos de diferentes níveis de pesquisa.

Seria muito interessante a elaboração de um trabalho acadêmico utilizando a concepção de orientação a objetos, propondo a estruturação de uma metodologia padrão, sendo aplicada ao desenvolvimento de projetos de automação ou de sistemas de tempo real, esta proposta teria que ir acompanhada de uma aplicação real, como por exemplo a implementação do nível de coordenação do sistema aberto de automação,

para validar a proposta.

A utilização da especificação formal como ferramenta de projeto pode originar um trabalho acadêmico, que envolveria o desenvolvimento ou utilização adequada deste tipo de ferramenta, mas visando as aplicações como a apresentada nesta dissertação.

A estrutura do controlador permitirá o desenvolvimento de trabalhos na área de redes, como a implementação de protocolos MAP e de protocolos tipo "Field Bus", o que permitiria um estudo prático do comportamento em ambientes reais incluindo a análise desempenho dos mesmos.

O desenvolvimento de um simulador educativo dos elementos da célula flexível com características como o que foi apresentado, permitindo a implementação de qualquer tipo de célula flexível na tela, explorando as técnicas da computação gráfica e das interfaces homem-máquina, originariam um excelente trabalho acadêmico.

Outro trabalho seria a análise de desempenho deste tipo de estruturas abertas, multiprocessadores, baseada em vias padrões, aplicadas a diferentes processos da manufatura, de forma a obter resultados práticos, que seriam utilizados como referências em outras aplicações de automação, ou como justificativa para a utilização de outras arquiteturas mais potentes como as baseadas em processadores RISC (Reduced Instruction Single Computer) ou Transputers.

A implementação de gerenciadores de banco de dados, onde exista uma mistura de bancos de dados, como bancos reais, históricos e temporais, exigem novas propostas que otimizem a manipulação dos dados, visando a proposta padrão SQL (Structured Query Language) (6). É uma linha que continua em desenvolvimento em especial a manipulação de banco de dados reais e temporais.

A proposta de um sistema aberto de automação estabelece a estrutura para atingir um fluxo contínuo de informação, de modo que, qualquer ponto do sistema possa obter facilmente informação desde qualquer outro ponto.

Todo o anterior, implica melhorar a comunicação dentro de um projeto, que é uns dos objetivos dos sistemas de informação, incluir novas técnicas de apoio aos projeto como por exemplo a multimidia, os banco de conhecimento ou alguns modelos baseados em redes neurais, e estruturas como o FDDI (Fiber Distributed Data Interface) rede de alta velocidade, dentro destes sistemas, implica propostas acadêmicas de metodologias e modelos para a implementação destes sistemas de informação, de forma a atingir um ótimo fluxo da informação dentro do projeto.

Finalmente, é bom ressaltar que o controlador de célula flexível com as características apresentadas converte-se numa "cama de testes" para o estudo de diferentes linhas de pesquisa, como por exemplo para o estudo de comportamento de transações, e implementações de sistemas operacionais aplicados a sistemas de tempo real, ou para aspectos relacionados com os sistemas distribuídos, como sistemas operacionais distribuídos, sem esquecer os estudos na área de confiabilidade.

6.4 Contribuição Acadêmica

Finalmente, neste item apresenta-se comentários dos principais aspectos que envolvem a contribuição acadêmica.

Um aspecto importante é a metodologia utilizada para este tipo de projeto, foi objetivo apresentar a metodologia de desenvolvimento como uma estrutura bem sistemática, com o respaldo acadêmico necessário, de forma a obter um esquema padrão,

e induzir a utilização desta metodologia em outros projetos da mesma natureza, e incentivando à utilização de ferramentas mais poderosas de apoio ao projeto.

A decisão de implementar o controlador com uma arquitetura aberta demonstrou a nova concepção de implementação de equipamentos de automação, e ajudou no fortalecimento da proposta dos sistemas abertos de automação, mas também criou a necessidade de transmitir estes conceitos aos usuários e à industria nacional, em especial no setor encarregado de fornecer recursos computacionais, que permitam a implementação deste tipo de sistemas, hoje sem muitas alternativas.

A maior contribuição acadêmica, que foi o principal objetivo desta dissertação, é que o controlador de célula flexível tornar-se-á um equipamento do laboratório de automação do Departamento de Engenharia da Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, cujo objetivo é o ensino e pesquisa das novas técnicas modernas aplicadas à automação, para ser utilizado pelos alunos de graduação e pós-graduação desta escola.

REFERÊNCIAS BILIOGRÁFICAS

(1) MANDELLI, G. Uma ferramenta ideal para as pequenas e medias empresas. **Máquinas e Metais**, n. 392, ano xvii, p.12-4, Jan. 1992.

(2) MALUF, L.F. Contribuição da automaço industrial e das novas técnicas de manufatura na competitividade da industria brasileira a nivel internacional. um analise da reserva de mercado. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL, 4:, São Paulo, **Anais**. São Paulo, 1990. p.133-40.

(3) CHANG, CH.H. A new perspective on realization of computer integrated manufacturing. **Manufacturing Review**, v.2, n.2, p.82-90, June 1989.

(4) ISO 7998/1, **Information processing open system interconnection basic reference model**. 1984.

(5) KAMINSKY JR., M. A. Protocols for communicating in the factory. **IEEE Spectrum**, v.23, n.4, p.56-62, Apr. 1986.

(6) DIGITAL EQUIPMENT CORPORATION, **Open systems handbook: a guide to building open systems**. Digital Equipment Corporation, 1991.

(7) MARTIN, J.M. Cell drive manufacturing strategy. **Manufacturing Engineering**, v.102, n.1, p.49-54. Jan. 89.

(8) SHONBERGER, R.J. Plant layout becomes produt-oriented with cellular, Jus-in-Time production concepts. **Industrial Engineering**, v.15, n.11, p.66-83, Nov. 1983.

(9) GOMMA, H. A. Software design method for real time system. **Communication Of The ACM**, v.27, n.9, p.938-49, Sept. 1984.

(10) LADUZINSKY, A. J. Integration focus: cell control. **Control Enginnering**, v.37, n.1, p.65-6, Jan. 1990.

(11) WILSON, B.D. Efforts towards the next generation worstation and machine controller architecture. **Manufacturing Review**, v.3, n.3, p.142-6, Sept.

1990.

(12) LADUZINSKY, A.J. Standards and interface move real time closer to control users. **Control Engineering**, v.39, n.4, p.53-4, Mar. 1992.

(13) MARTUCCI JR., M. **Estudos de estruturas de sistemas de automação**. São Paulo, 1992. Tese (Livre Docência) - Escola Politécnica, Universidade De São Paulo.

(14) DILTS, D. M., BOYD, N.P., WHORMS, H.H. The evolution of control for automated manufacturing systems. **Journal of Manufacturing Systems**, v.10, n.1, p.79-93, 1991.

(15) ZAVE, P. An insider's evaluation of PAISLEY. **IEEE Transaction on Software Engineering**, v.17, n.3, p.212-25, Mar. 1991.

(16) BLACK, J.T. Cellular manufacturing systems reduce setup time, make small lot production economical. **Industrial Engineering**, v.15, n.11, p.36-48, Nov. 1983.

(17) MASON, F. As células de manufatura podem ser a solução para as empresas. **Máquinas e Metais**, n.392, ano xvii, p.16-20, Jan. 1992.

(18) CULBRETH, C.T., POLLPETER, D.L. A flexible manufacturing cell for furniture part production. **Industrial Enginnering**, v.20, n.11, p.28-34, Nov. 1988.

(19) MCLEAN C.R., BROWN, P.F. The architecture of the NBS factory automation research tesbed. In: IFAC, TRIENNIAL WORLD CONGRESS, 10: Munich, 1987. **Automatic control**, Oxford, Pergamon, 1988. v.4, p. 141-46.

(20) SUNDARAM, R.M., LIAN, W. S. An approach for designing cellular manufacturing systems. **Manufacturing Review**, v.3, n.2, p.91-7, June 1990.

(21) ORTIZ, L.E., LIMA, A.M., MUZZI JR. J.J. Alternativas brasileiras para viabilização de sistemas De manufatura flexíveis: tendências e propostas. In:

SEMINÁRIO DE COMANDO NUMÉRICO NO BRASIL, **10:, Anais**, São Paulo, 1990, p.1-21.

(22) RHODES, D. The critical imperative in flexible manufacturing is information. **Industrial Engineering**, v.20, n.10, p.44-6, Oct. 1990.

(23) KOELSH, J.R. Flexible cells take control. **Manufacturing Engineering**, v.105, n.3, p.75-7, Sept. 1990.

(24) VONDEREMBSE, M.A., WOBSEN, G. S. Step for implementing a flexible manufacturing system. **Industrial Engineering**, v.19, n.4, p.38-48, Apr. 1987.

(25) KUSIAK, A., CHOW, W.S. Decomposition of manufacturing systems. **IEEE Journal of Robotic and Automation**, v.4, n.5, p.457-71, Oct. 1988.

(26) WEI, J.C., GAITHER, N. A. Capacity constrained multiobjetive cell formation method. **Journal of Manufacturing Systems**, v.9, n.3, p.222-32, 1990.

(27) DUMOLIEN, W., SANTEN, W.P. Cellular manufacturing becomes philosophy of management at components facility. **Industrial Enginnering**, v.15, n.11, p.72-6, Nov. 1983.

(28) DAIGLE, J.N., SEIDMANN, A., PIMENTEL, J. R. Communications for manufacturing: an overview. **IEEE Network**, v.2, n.3, p.6-13, May 1988.

(29) ANG, C.L. Planning and implementing CIM. **Computer-Aided Engineering Journal**, v.6, n.5, p.167-76, Oct. 1989.

(30) KOLHER, C., SCHMIERL, S. Computer Integrated Manufacturing(CIM) technological and organizational change in the West German capital goods industry. **Journal of Manufacturing Systems**, v.10, n.1, p.21-31, 1991.

(31) MCGUFFIN, L.J., REID, L.O., SPARKS, S.R. MAP/TOP in CIM distributed computing. **IEEE Network**, v.2, n.3, p.23-31, May 1988.

(32) KUSIAK, A., HERAGU, S. S. Computer integrated manufacturing: a structural perspective. **IEEE Network**, v.2, n.3, p.14-22, May 1988.

(33) BIERBAUM, K. Implementing CIM. The process begins on the factory floor.

In: MORRIS, K., Advanced control conference, 13:, West Lafayette, 1987. **Advanced control in computer integrated manufacturing**, West Lafayette, Purdue research foundation, 1987, p.97-103.

(34) LARIN, D.J. Cell control: what we have, what we will need. **Manufacturing Engineering**, v.102, n.1, p.41-8, Jan. 1989.

(35) WAITE, R. Building CIM- The role of the cell. In: MORRIS, K., Advanced control conference, 13:, West Lafayette, 1987. **Advanced control in computer integrated manufacturing**, West Lafayette, Purdue research foundation, 1987, p.35-40.

(36) BADIRU, A.B. Analysis of data requirements for FMS implementation is crucial to sucess. **Industrial Engineering**, v.22, n.10, p.29-32, Oct. 1990.

(37) LADUZINSKY, A.J. The origin of the cell control species. **Control Engineering**, v.38, n.1, p.65-7, Jan. 1991.

(38) TEICHOLZ, E. **CAD CAM Handbook**. New yorK, McGraw-Hill, 1985.

(39) TSENG, A.A., KOLLURI, S.P., RADHAKRISHNAN, P. A CNC machining system for education. **Journal of Manufacturing Systems**, v.8, n.3, p.207-14, 1989.

(40) MORRIS, H.M. Should you connect CNC to CIM without converting to DNC. **Control Engineering**, v.33, n.5, p.56-7, May 1986.

(41) MORRIS, H.M. How do robots fit in CIMã. **Control Engineering**, v.34, n.5, p.60-1, May 1987.

(42) BESANT, C.B., LUI, C.W.K. **Computer aided design and manufacturing**. 3 Ed., New York, Ellis Horwood, 1986.

(43) OFFODILE, O.F., JOHNSON, S.L. Taxonomic system for robot selection in flexible manufacturing cells. **Journal of Manufacturing Systems**, v.9, n.1, p.77-80, 1990.

(44) GRAHAM, J.H. Special computer architectures for robotics: tutorial and survey. **IEEE Transactions on Robotics and Automation**, v.5, n.5, p.543-

54, Oct. 1989.

(45) CHAN, S.F., WESTON. R.H., CASE, K. Robot simulation and off-Line programming. **Computer-Aided Engineering Journal**, v.5, n.4, p.157-62, Aug. 1988.

(46) RED, W.E., DIXON, M.G. Robots - programmed for Ssccess. **Manufacturing Engineering**, v.104, n.5, p.43-7, May 1990.

(47) LANN G.L. Designing real time dependable distributed systems. **Computer Communication**, v.15, n.4, p.225-34, May 1992.

(48) PRESSMAM, R. **Software engineering:** a practioner's approach. 2 ed, New York, McGraw-Hill, 1987.

(49) LAWRENCE P.D, MAUCH. K. **Real time micromputer system design:** an introduction. New York, McGraw-Hill, 1987.

(50) QUATSE, J.T., An architecture for real time controller. **Control Enginnering**, v.34, n.5, p.56-9, May. 1987.

(51) ANSI/IEEE Std 830-1984. **An american standard IEEE guide to software requirements specifications.**

(52) BALL, E., LINGE, N., KUMMER, P., TASKER, R. Local area networks bridges. **Computer Communication**, v.11, n.3, p.115-17, June 1988.

(53) CHANTAMANENI, P.R, JALOTE, P., SHIELA, Y., TRIPATHI, S.K. On fault tolerance in manufacturing systems. **IEEE Network**, v.2, n.3, p.32-9, May 1988.

(54) STANKOVIC, J.A. A perspective on distributed computer systems. **IEEE Transaction on Computer**, v.33, n.12, p.1102-15, Dec. 1984.

(55) FISHER, G.T. Cell controllers, MAP communications, and sequencial function chart programming- is the ideal batch control system. **ISA Transaction**, v.28 , n.3, p.57-73, 1989.

(56) BLANCHAR, D., ISRANI, S., KNISKEN, J. Cell controller-an emerging technology. **Control Enginnering**, v.33, n.11, p.170-4, Nov. 1986.

(57) JONES, C.T., BRYAN, L.A. **Programmer controller concepts and applications**. Atlanta, IPC/ASTEC Publication, 1986.

(58) PLEINEVAUX P., DECOTIGNIE, J.D. Time critical communication network: field buses. **IEEE Network**, v.2, n.3, p.55-63, May 1988.

(59) ROESLER, H., PIERSON, L. Open Communication Based On International Standards. **Control Engineering**, v.39, n.4, p.20-2, Mar. 1992.

(60) SINGHAL, M. Issues and appoach to design of real time database systems. **Sigmod Record**, v.17, n.1, p.19-33, Mar. 1988.

(61) GRAY, J., SIEWIOREK, D.P. High availability computer systems. **Computer**, v.24, n.9, p.39-48, Sept. 1991.

(62) MARCUS A., VAN DAMME, A. User-interfaces developments for the nineties. **Computer**, v.24, n.9, p.49-57, Sept. 1991.

(63) GLEM, L., EARLES, T. X windows comes for real time operating systems. **Control Engineering**, v.38, n.3, p.67-8, Mar. 1991.

(64) YAU, S.S., JIA, X., BAE, D. H. Software design methods for distributed computing systems. **Computer Communication**, v.15, n.4, p.213-23, May 1992.

(65) YOURDON, E. **Modern structured analysis**. Yourdon Press. 1989.

(66) ROSS, D.T. Applications and extensions of SADT. **Computer**, v.18, n.4, p.25-34. Apr. 1985.

(67) WARD, P.T., MELLOR, S.J. **Structured development for real time systems**. Yourdon Press, 1985, 3v.

(68) HATLEY, D.J., PIRBHAI, I.A. **Estratégias para especificação de sistemas de tempo real**. São Paulo, MacGraw-Hill. 1991.

(69) ALFORD, M. SREM at the age of eight; the distributed computing design system. **Computer**, v.18, n.4, p.36-46, Apr. 1985.

(70) FICHMAN, R.G., KEMERER, C.F, Object oriented and conventional analysis and design methodologies. **Computer**, v.25, n.10, p.22-39, Oct. 1992.

(71) GOMMA, H. Software development of real time systems. **Communication Of the ACM**, v.29, n.7, p.657-68, July 1986.

(72) NEMES, L. Data flow and its control in flexible manufacturing. In: IFAC, TRIENNIAL WORLD CONGRESS, 10: Munich, 1987. **Automatic control**, Oxford, Pergamon, 1988. v.4, p.153-57.

(73) INSTITUTO BRASILEIRO DE PESQUISA. **PC-CASE: DFD**. Rio de Janeiro, Base Tecnologica, 1987.

(74) SHIN, K.G. HARTS: a distributed real time architecture computer. **Computer**, v.24, n.5, p.25-35, May 1991.

(75) O'BRIEN, S., WIDDEN, M.B. The application of transputer control to a robotically-serviced manufacturing cell. **SERC/DTI Transputer Initiative**, **Mailshot**, p.23-30, Feb. 1992.

(76) NARISIMHAN S., SIEGEL, D.M., HOLLERBACH, J.M. CONDOR: a revised architecture for controlling the Utah-Mit hand. In: IEEE INTERNACIONAL CONFERENCE ON ROBOTIC AND AUTOMATION, Philadelphia, 1988. **Proceedings**, Washington, IEEE Computer Society, 1988, v.1 , p.446-9.

(77) HOPKINS, A.L., SMITH, T.B., LALA, J.H. FTMP- a highly reliable fault - tolerance multiprocessor for aircraft. **Proceedings Of The IEEE**, v.66, n.10, p.1221-39, Oct. 1978.

(78) KIECKHAFER, R.M., WALTER, C.J., FINN, A.M, THAMBIDURRAI, P.M. The MAFT achitecture for distributed fault tolerance. **IEEE Transaction On Computers**, v.37, n.4, p.398-405, Apr. 1988.

(79) LEE, I., KING, R.B., PAUL, R.P. A predictable real time kernel for distributed multisensor systems. **Computer**, v.38, n.6, p.78-83, Jun. 1989.

(80) SCHWAN, K., GOPINATH, P., BO, W. CHAOS- kernel support for objects in the real time domain. **IEEE Transaction On Computer**, v.36, n.8, p.904-16, Aug. 1987.

(81) VAN DIJK, G.J.W. EMPS: the design of an architecture for distributed homogeneous multiprocessor system. **Microprocessor And Microsystems**, v.15, n.4, p.187-94, May 1991.

(82) KOMODA, N., KERA, K., KUBO, T. An autonomous, decentralized control system for factory automation. **Computer**, v.17, n.12, p.73-82, Dec. 1984.

(83) HARIRI, S., CHOUDHARY, A., SARIKAYA, B. Architectural support for designing fault tolerant open distributed sytems. **Computer**, v.25, n.6, p.50-62, June 1992.

(84) KOPETZ, H., MERKER, W. The architecture of MARS. In: THE FIFTEENTH ANNUAL INTERNACIONAL SYMPOSIUM ON FAULT TOLERANT COMPUTING, 5:, Michigan, 1985, **Digest of Papers**, Silver Spring, IEEE Computer Society Press, 1985, p.274-9.

(85) SAVITZKY, S.R. **Real time microprocessor systems**. New York, Van Nostrand Reinhold. 1985.

(86) CSURGAI, G., KOVACS, V., LAUFER, J. A generalized model for control and supervision of unmanned manufacturing cell. In: **Software for Discrete Manufacturing**. North Holland, Elsevier Science Publishers B.V., 1986.

(87) WELLS, G.C. An aproach to the evalution of microcomputer operating systems with respect to real time issues. **Microprocessors and Microprogramming**, v.33, n.1, p.45-55, Sept. 1991.

(88) GAUTHIER, D., FREEDMAN, P., CRAYANNIS, G. MLOWANY, A.S. Interprocess communication for distributed robotics. **IEEE Journal of Robotics and Automation**, v.3, n.6, p.493-504, Dec. 1987.

(89) GEROLD, J., S. Standard architecture put industrial computer in control. **Control Engineering**, v.39, n.8, p.73-6, Jun. 1992.

(90) CHILD, J. Divide and conquer strategy for new york's traffic control system. **Computer Design**, v.31, n.3, p.109-11, Mar. 1992.

(91) HIRSH, A. VME Vs. PLC. **Computer Design**, v.31, n.3, p.29-31, Mar. 1992.

(92) ANDREWS, W. Multiprocessing to bring the next jump in perfomance. **Computer Design**, v.31, n.1, p.78-98, Jan. 1992.

(93) ANDREWS, W. 32-Bit buses contend for designers attention. **Computer Design**, v.28, n.21, p.78-96, Nov. 1989.

(94) BEASTON, J. Chossing a bus for control. Instruments & Control Systems, p.145-149, Mar. 1983, In: INTEL. **OEM systems Handbook**. Santa Clara, Intel, 1985.

(95) TITUS, J. Industrial buses. **EDN**, v.32, n.3, p.114-26, Fev. 1987.

(96) MCEACHEN, J. The bus evolution past, present and future. In: ADVANCED CONTROL CONFERENCE, INDUSTRIAL COMPUTING CONTROL AFTER 25 YEARS: MICROS TO HIERARQUIES, 10., West Lafayette, 1984. **Proceeding**. Barrington, Technical Publishing Company, 1984, p.243-259.

(97) ANDREWS, W. Enhancing the perfomance of standard buses. **Computer Design**, v.30, n.12, p.90-108, Sept. 1991.

(98) BYBEE, R. Customized buses build on industry standards to enhance perfomance. **Computer Design**, v.28, n.3, p.109-12, Feb. 1989.

(99) ANDREWS, W. Application Determines Best Approach To Multiprocessing. **Computer Design**, v.29, n.23, p.72-86, Dez. 1990.

(100) WARREN, A. Mezzanine buses struggle with standars. **Computers Design**, v.30, n.10, p.81-8, Jul. 1991.

(101) BALPH, T., ARTUSI, D. VME, a system architecture for industrial control. In: ADVANCED CONTROL CONFERENCE, INDUSTRIAL COMPUTING CONTROL AFTER 25 YEARS: MICROS TO HIERARQUIES, 10., West Lafayette, 1984. **Proceeding**. Barrington, Technical Publishing Company, 1984, p.261-67.

(102) ANSI/IEEE Std 1014-1987. **An american national standard IEEE standard for a versatile backplane bus:** VMEbus.

(103) FISHER, W. IEEE P1014-A standard for high-performance VME bus. **IEEE Micro**, v.5, n.1, p.31-41, Feb. 1985.

(104) MOTOROLA, **MC68040 Microprocessor user's manual**. Motorola Inc. 1992.

(105) HENNESY, J. L, PATTERSON, D.A. **Computer architecture a quantitive approach**. San Mateo, California, Morgan Kauffmann Publishers Inc, 1990.

(106) MOTOROLA, **MC68030 enhanced 32 bits microprocessor user's manual,** Motorola Inc. 1987.

(107) INSTRUMENT SOCIETY OF AMERICA, **SP 50. field instrument bus standard: functional guidelines**.

(108) WOOD, G.G. Current fieldbus activities. **Computer Communication**, v.11, n.3, p.118-23, June 1988.

(109) FISHER, G.T. An introduction to MAP. **ISA Transaction**, Vol. 28, n.4, p.1-11, 1989.

(110) SVOBODOVA, L. Implementing OSI systems. **Journal on Selected Areas in Communications**, v.7, n.7, p.1115-30, Sept. 1989.

(111) DAY, J.D., ZIMMERMANN, H. The OSI reference model. **Proceedings of The IEEE**, v.71, n.12, p.1334-40, Dec. 1983.

(112) WESTON, R.H., GASCOIGNE, J.D, SUMPTER, C.M. Industrial computer networks and the role of MAP, Part 2. **Microprocessor and Microsystems**, v.11, n.1, p.21-34, Feb. 1987.

(113) GENERAL MOTORS. **Manufacturing automation protocol**, Versão 3.0. Jul 1987.

(114) MOTOROLA, **MC68020 microprocessor user's manual**. 2 Ed. Prentice Hall, N.J. 1985.

(115) KUHN, R.D. IEEE'S POSIX: making progress. **IEEE Spectrum**, v.28, n.12, p.36-40, Dec. 1991.

(116) STANKOVIC, J.A. Misconception about real time computing a serious problem for next generation systems. **Computer**, v.21, n.10, p.10-9, Oct.

1988.

(117) LADUZINSKY, A. UNIX expands on factory-floor adds functionality to control. **Control Engineering**, v.38, n.12, p.81-4, Dec. 1991.

(118) LEBLOND, G., BLUST, S., MODES, W. **UNIX 5**, São Paulo, Mckron Boooks do Brasil. 1992.

(119) LEE, H., CHO, J., KIM, J. Development of a real time database managment system for production process-control applications. **Microprocessing and Microprogramming**. v.35, n.1-5, p.445-52, Sep. 1992.

(120) PERRON, S. The new generation language for controls design. **Control Engineering**, v.39, n.2, p.27-8, Fev. 1992.

(121) EYDGAHI, A.M., SHEEHAN, J. A computer animation of robotic manipulators and workcells. **IEEE Control Systems**. v.11, n.4, p.56-9, Jun 1991.

(122) WLOKA, D.W. Simulation of robot factories using robsim. In: IFAC, TRIENNIAL WORLD CONGRESS, 10: Munich, 1987. **Automatic control**, Oxford, Pergamon, 1988. v. , p.313-9.

(123) ROY, G., NORRIE, D., FAUVEL, R. 3-D simulation of CNC machine tool operation on a super minicomputer and on a PC. **Journal of Manufacturing System**, v.7, n.3, p.267-71, 1988.

(124) RAZ, T. Graphics robot simulator for teaching introductory robotics. **IEEE Transactions On Education**, v.32, n.2, p.153-9, May. 1989.

(125) WHITE, R.B., READ, R.K., KOCH, M.W., SHILLING, R.J. A graphics simulator for a robotic arm. **IEEE Transactions On Education**, v.32, n.4, p.417-29, Nov. 1989.

(126) RINE, C.D., BHARGAVA, B. Object oriented computing. **Computer**, v.25, n.10, p.6-10, Oct. 1992.

(127) WITTEN, M., BODNAR, B.L., LIU, A.C. Simulation and modeling of a single bus tightly coupled multiprocessor system. **Mathematics and Computer in**

Simulation, v. , n , p.19-31, 1987.

(128) SOARES, L.FG. **Modelagem e simulação discreta de sistemas**. São Paulo, IME-USP, 1990.

(129) MINUTEMAN SOFTWARE. **GPSS/PC general purpose simulation, reference manual**. Massachusetts, Minuteman Software, 1986.

(130) MINUTEMAN SOFTWARE. **GPSS/PC general purpose simulation, Tutorials.** Massachusetts, Minuteman Software, 1986.