

Estruturas de Controle para Sistemas de Manufatura Computadorizados

SISTEMAS DE CONTROLE

“Para produzir com flexibilidade, não basta comprar um FMS cujo software de controle seja fechado. É necessário que este software seja aberto para atualizações pelos próprios engenheiros da empresa que adquiriu o sistema, do contrário o FMS será inflexível.”

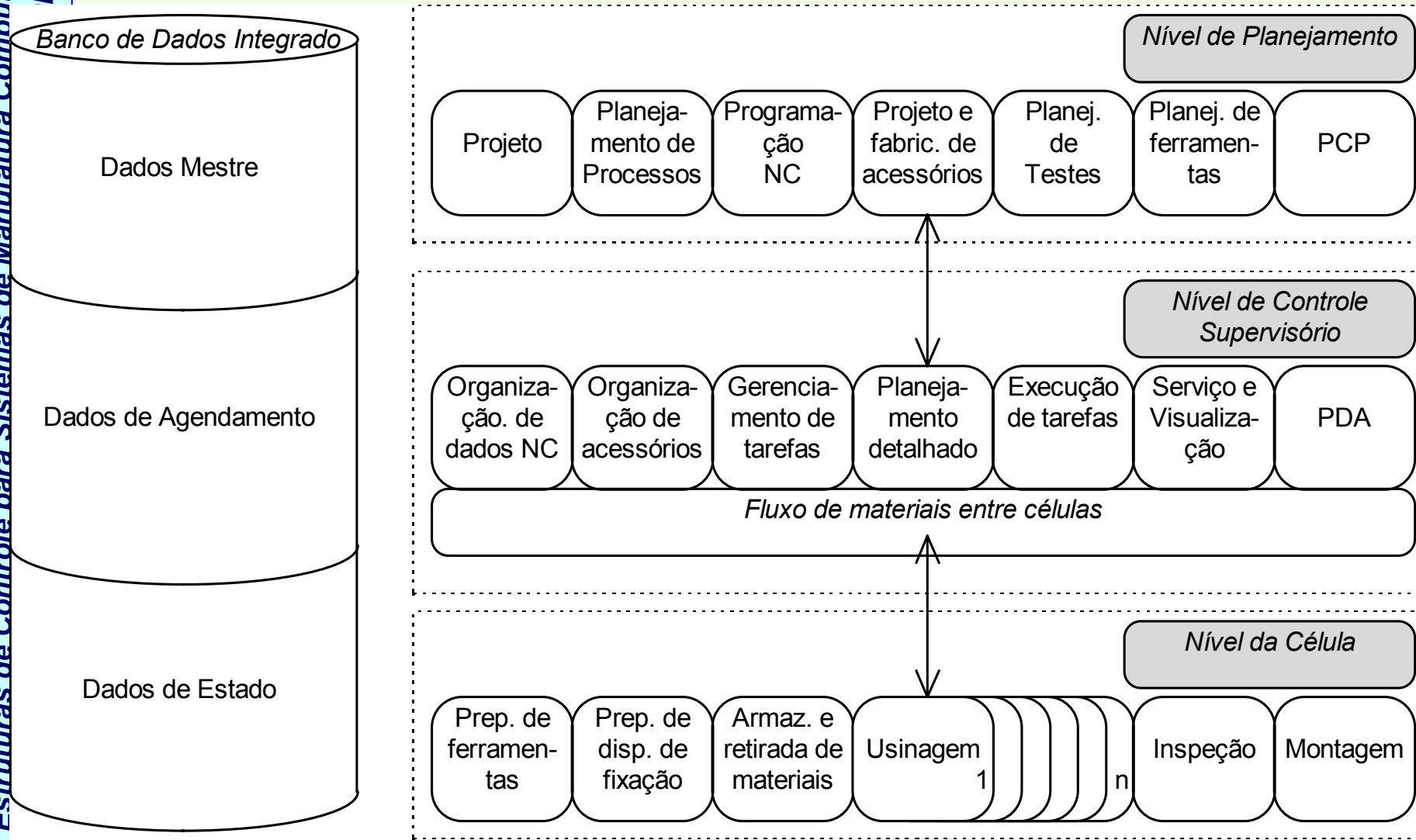
INTRODUÇÃO

- CAM → refere-se às atividades de processamento sobre as peças (p.ex. usinagem, soldagem, conformação).
- Envolve a execução de **tarefas de controle**.
- A **entrada**, o **monitoramento** e a **avaliação dos dados** em conjunto com o controle podem também ser funções dessa área.
- Sistemas de controle → podem ser estruturados segundo vários pontos-de-vista.

INTRODUÇÃO

- Distinção entre estruturas de controle *funcionais*, de *hardware* e de *software*.
- A representação do controle e de sistemas de controle em termos de **estrutura** → auxilia seu projeto e promove a sua compreensão.

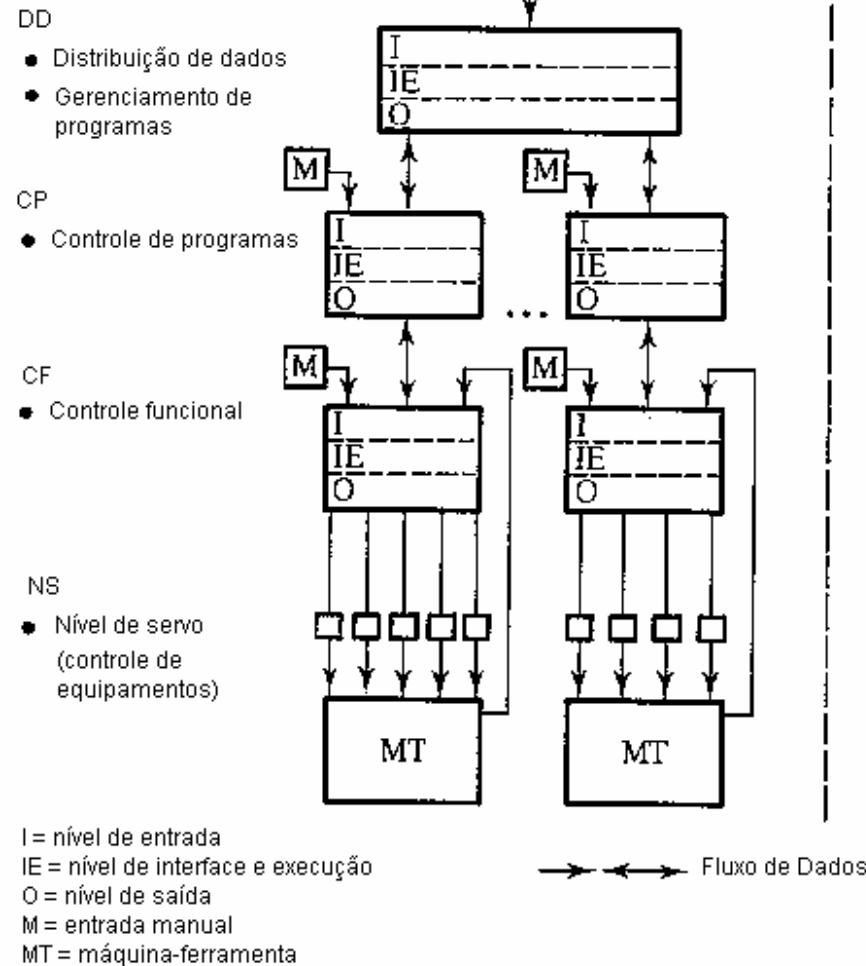
ESTRUTURA FUNCIONAL



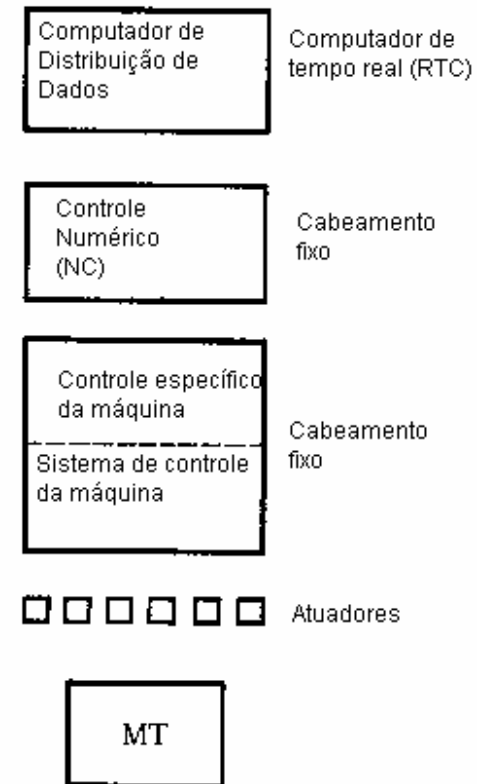
Estrutura hierárquica de um sistema de manufatura com níveis de processamento de informações

ESTRUTURAS DE HARDWARE

Estrutura funcional em níveis de controle



Tecnologia NC



Estrutura de um sistema de controle de máquina (NC) típica dos anos 70

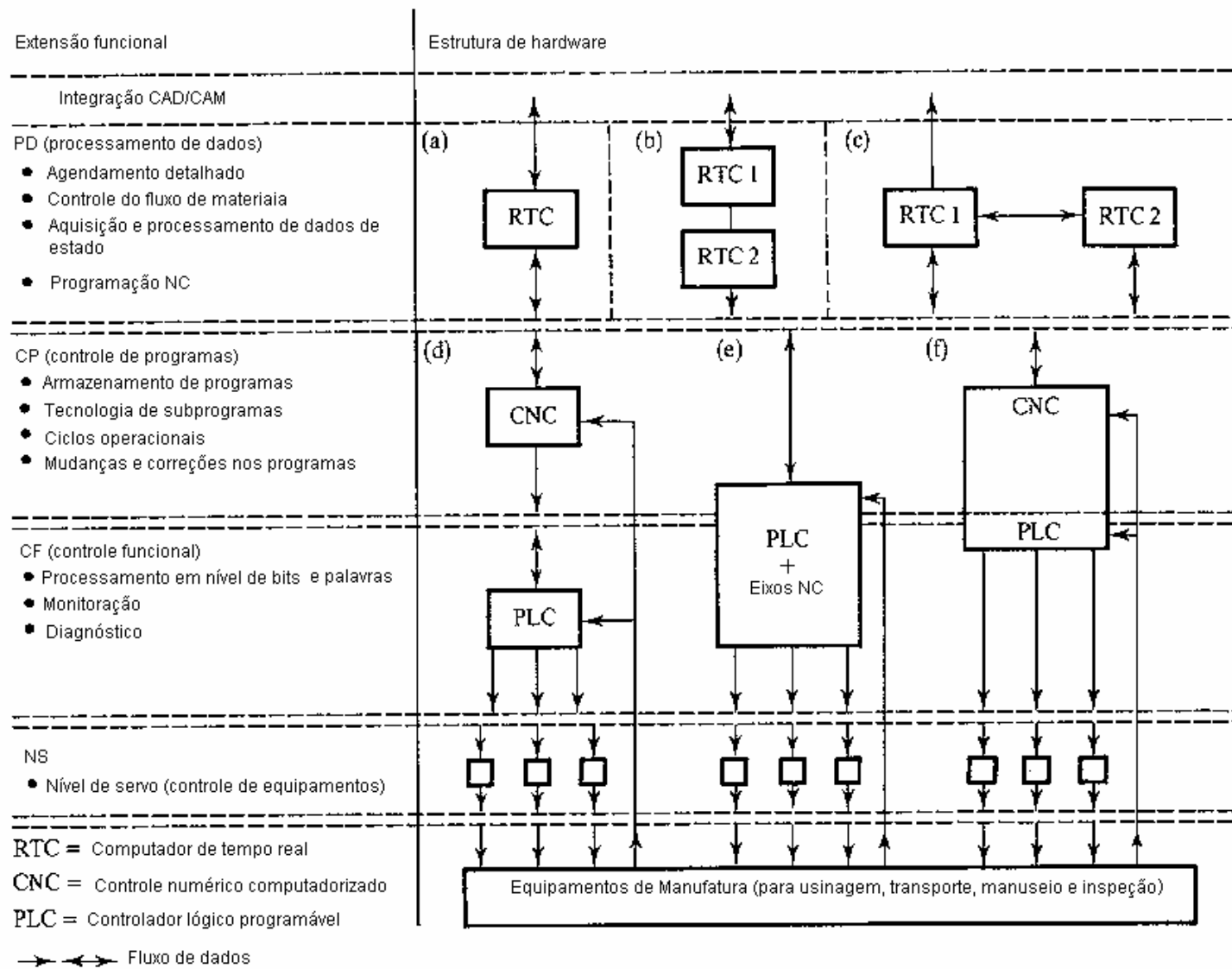
ESTRUTURAS DE HARDWARE

- Controle do programa; Controle funcional; Níveis de controle de equipamentos.
- Primeiras implementações de **sistemas CNC** continham:
 - distribuição de dados;
 - níveis de gerenciamento de programas.
- Final dos anos 1960 → primeiras **funções de controle** a serem efetuadas utilizando-se um computador em tempo real.
- Controle do programa → remoção de uma linha de programa & emissão de comandos de funções.
- Controle funcional → associa comandos de funções a comandos de saída para atuadores, levando em conta a realimentação.

ESTRUTURAS DE HARDWARE

- Deve-se perceber na figura → **entradas manuais** indicam que a interligação automática entre os níveis a princípio não é necessária.
- Características dessa estrutura incluem:
 - Correspondência funcional e de hardware dos níveis com uma implementação com **cabeamento fixo** (*hard-wired*) dos níveis de programa e controle (NC e PLC).
 - Cada nível possui subníveis de **entrada, processamento e saída**.

ESTRUTURAS DE HARDWARE

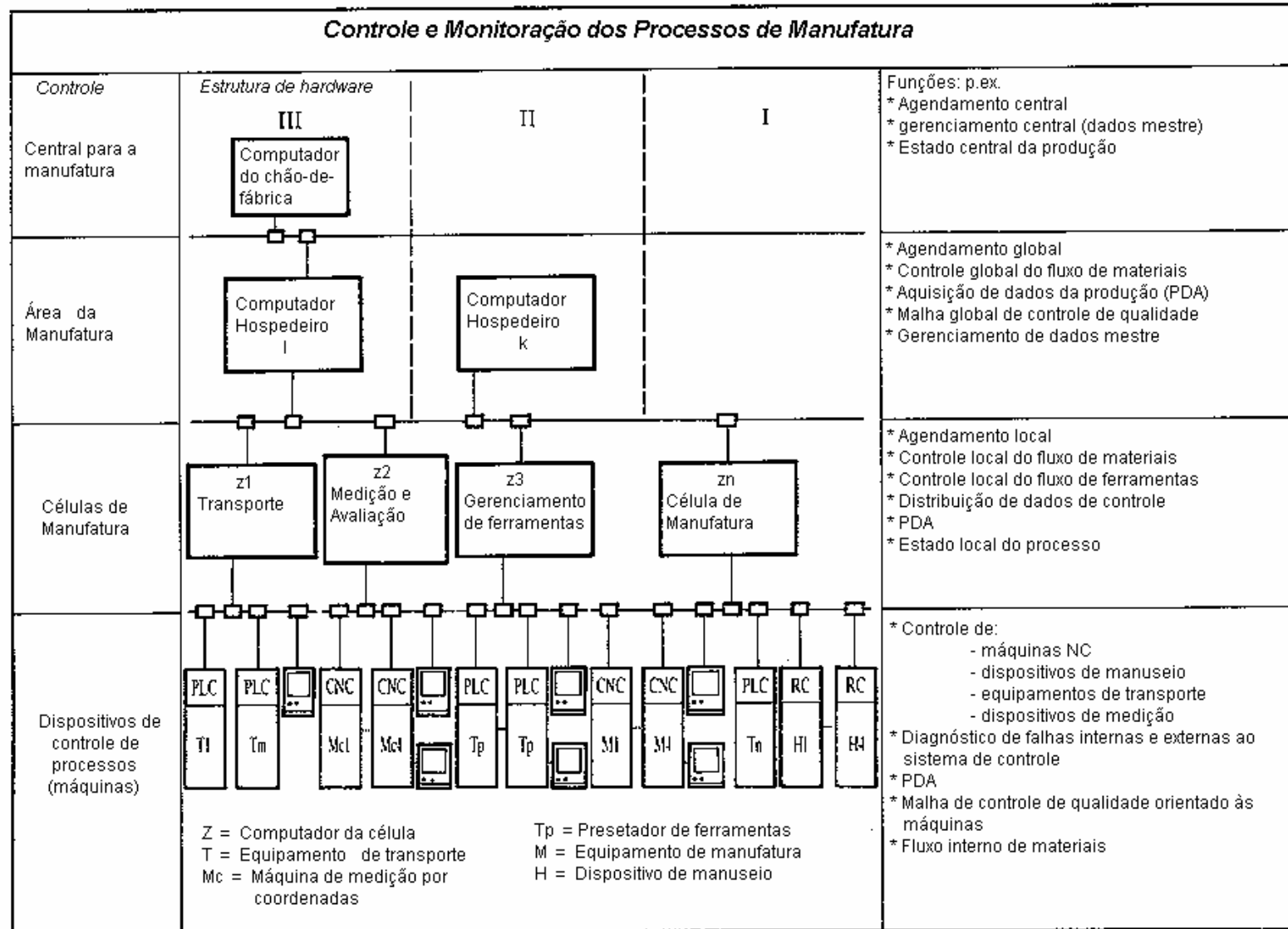


Estrutura de um sistema de controle

ESTRUTURAS DE HARDWARE

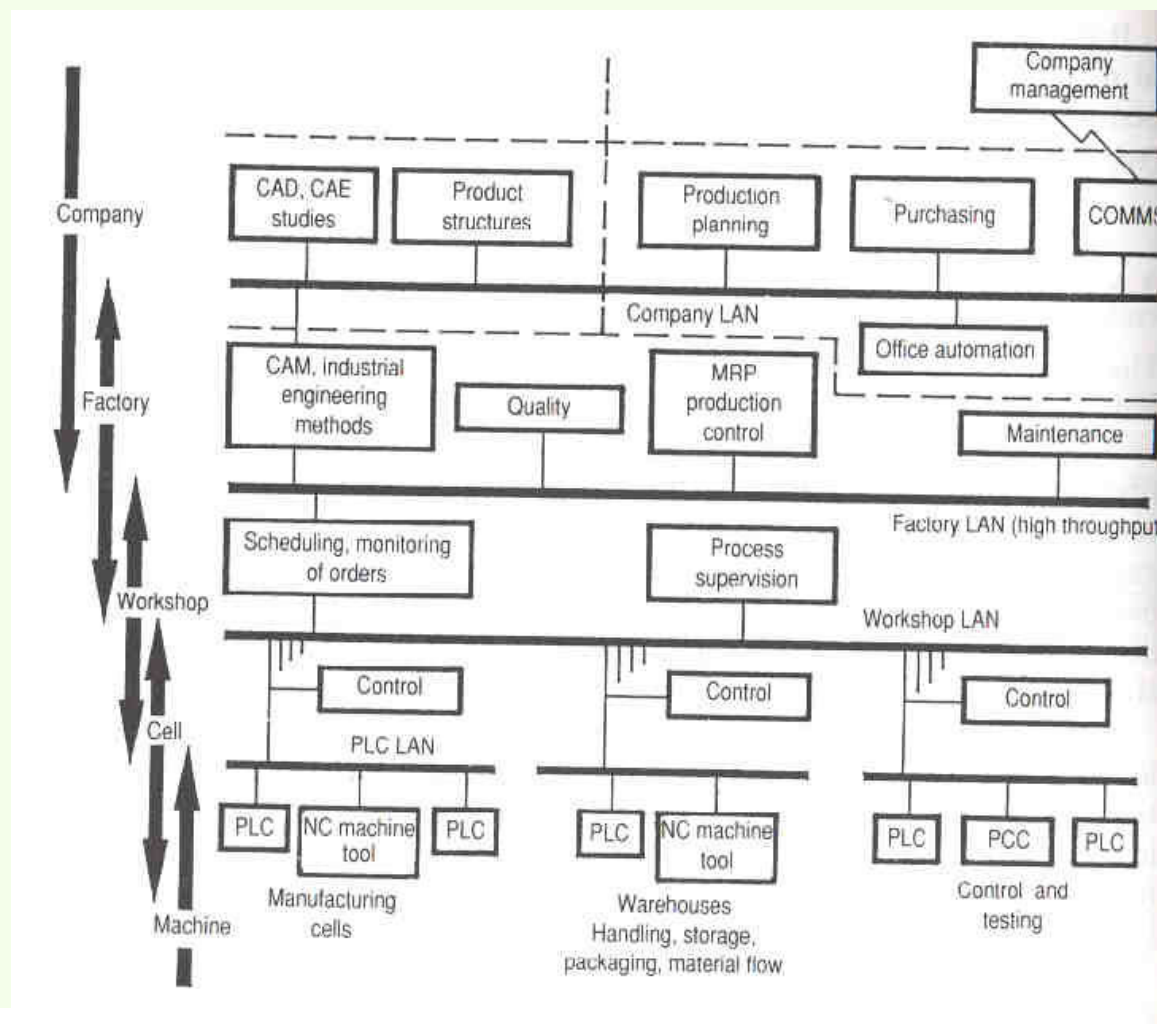
- Figura → **contorno** entre os níveis funcional e de hardware **não é mais claramente definido** → níveis de hardware cobrem **mais do que um nível funcional** → extensões funcionais são agora possíveis, e soluções *soft-wired* (CNC, PLC) são a regra.
- Componentes de **hardware** dos sistemas de controle:
 - computadores controladores,
 - dispositivos de controle do processo e de máquinas,
 - sistemas de comunicação.
- Avanços nos computadores → mudança no escopo funcional e na estrutura de sistemas de controle → p.ex. uso de **microcomputadores com características de controle em tempo real** (deve-se lembrar o aumento significativo nos **preços de software**).

ESTRUTURAS DE HARDWARE



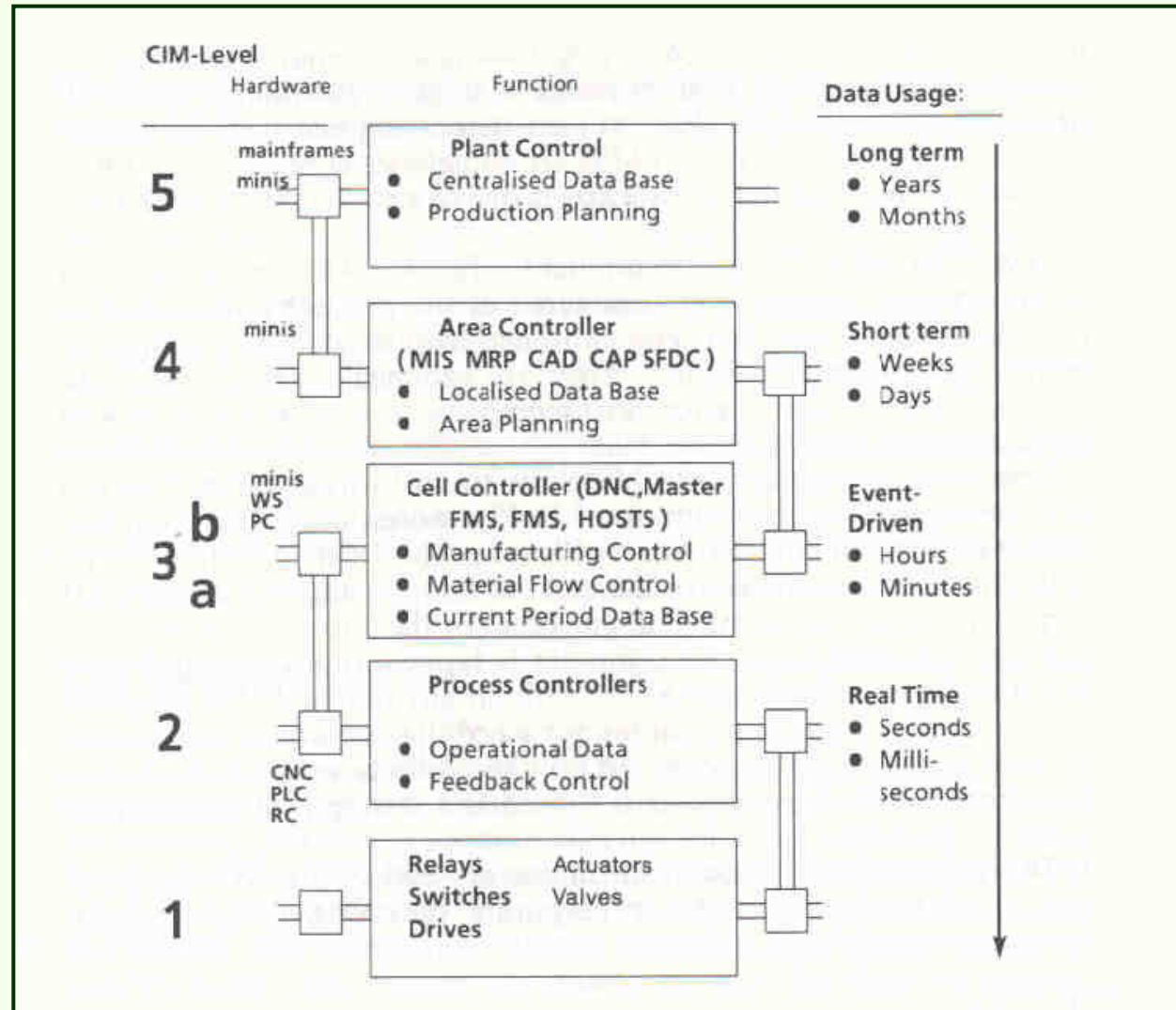
Nível hierárquico num CIM

ESTRUTURAS DE HARDWARE



Nível hierárquico num CIM

ESTRUTURAS DE HARDWARE



Nível hierárquico num CIM

ESTRUTURAS DE HARDWARE

- Próxima figura → **FMS** para a produção de **peças prismáticas usinadas com precisão**.
- Equipamentos deste sistema:
 - 4 centros de usinagem com armazenamento de ferramentas
 - 1 máquina de medição por coordenadas.
- Controle central → executado por um computador hospedeiro.
- Memória principal → 512Kbytes de RAM, e dois discos rígidos de 10Mbytes cada.
- 2 terminais são usados para servir o sistema de controle → um desses terminais localiza-se na área de fixação da peça, e uma impressora está ligada para mensagens operacionais.

ESTRUTURAS DE HARDWARE

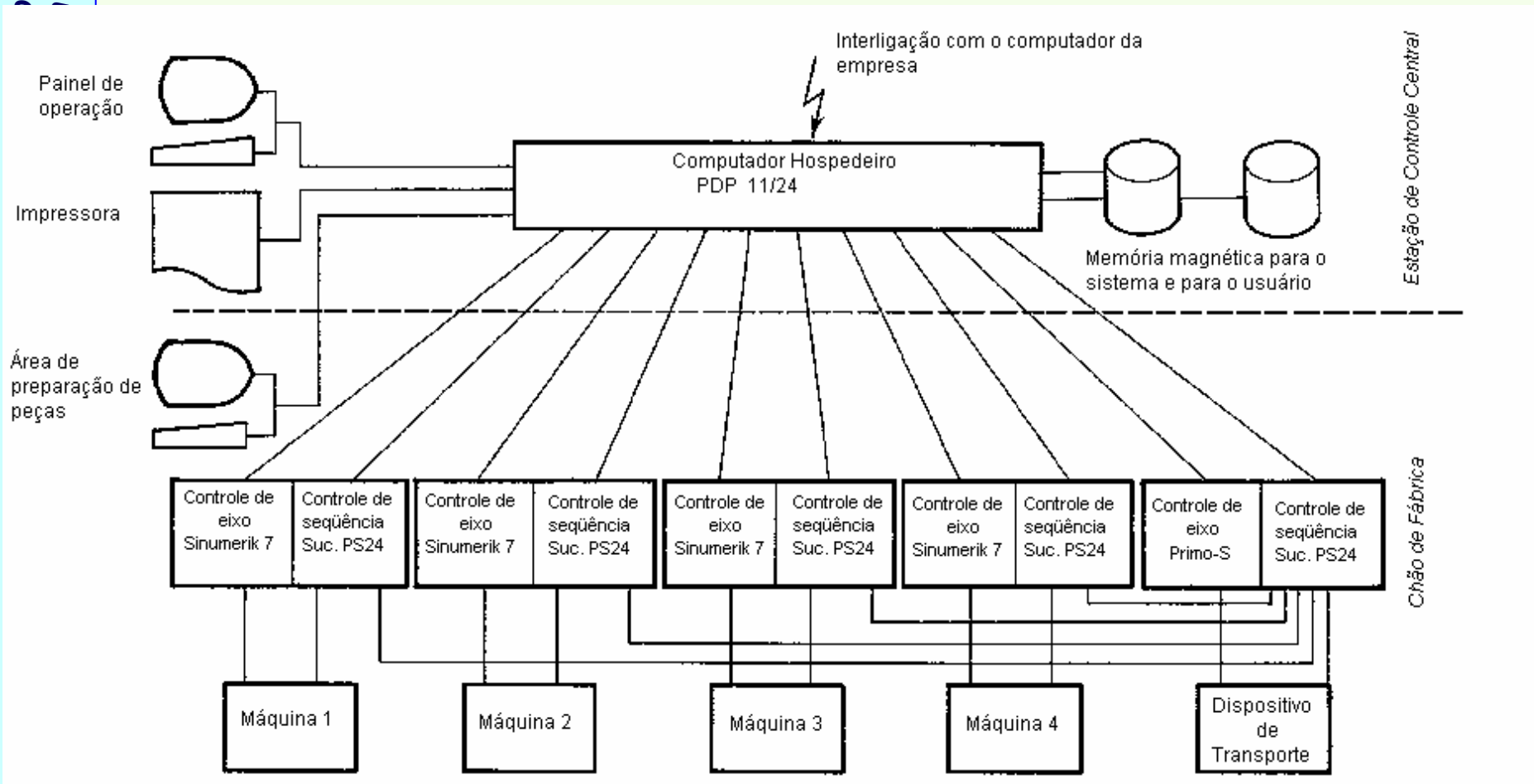


- 1 - Painel de operação
- 2 - Máquina de medição por coordenadas
- 3 - Armazenamento central de peças

- 4 - Sistema de transporte de peças
- 5 - Centros de usinagem CNC

Um FMS na empresa Zeiss (Alemanha)

ESTRUTURAS DE HARDWARE



Configuração de hardware do FMS na Zeiss

ESTRUTURAS DE HARDWARE

- Nível de controle → sistemas de controle para as estações de usinagem e do dispositivo de transporte → interligado através de **interfaces seriais** (padrão **RS232C**) com o computador hospedeiro num **arranjo tipo estrela**.
- Controles padronizados com **interfaces DNC** → usados para controlar as 4 estações de usinagem;
- **PLCs** são usados para o **controle da seqüência**.
- O **NC** e o **PLC** são interligados através de interfaces seriais com o computador hospedeiro.
- Sistema de controle para o dispositivo de transporte inclui o **controle do eixo para posicionar o veículo**, e o **controle de seqüência para o manuseio dos pallets**.

ESTRUTURAS DE HARDWARE

- Sistema de **controle do dispositivo de transporte** → responsável pelo controle:
 - da posição de fixação da peça,
 - do trocador de pallets da máquina CMM.
- Para **sincronizar** os dispositivos de transporte e os **trocadores de pallets** do centro de usinagem durante a transferência de pallets → **PLC** das máquinas & os dispositivos de transporte são também interligados.

ESTRUTURAS DE HARDWARE



Mazak FMS

ESTRUTURAS DE HARDWARE



Mazak FMS

ESTRUTURAS DE HARDWARE



Mazak FMS

ESTRUTURAS DE HARDWARE



Mazak FMS

ESTRUTURAS DE HARDWARE

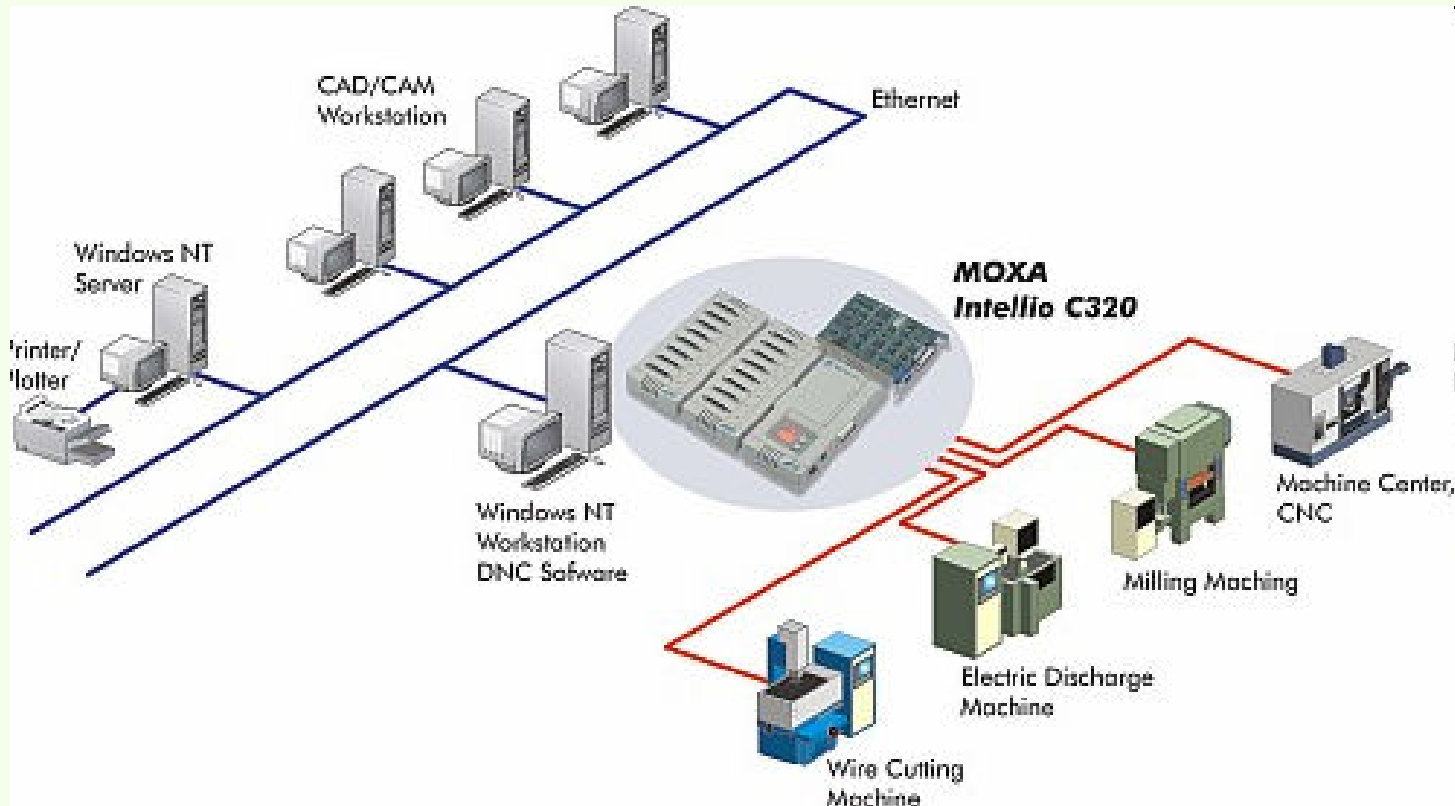


Mazak FMS

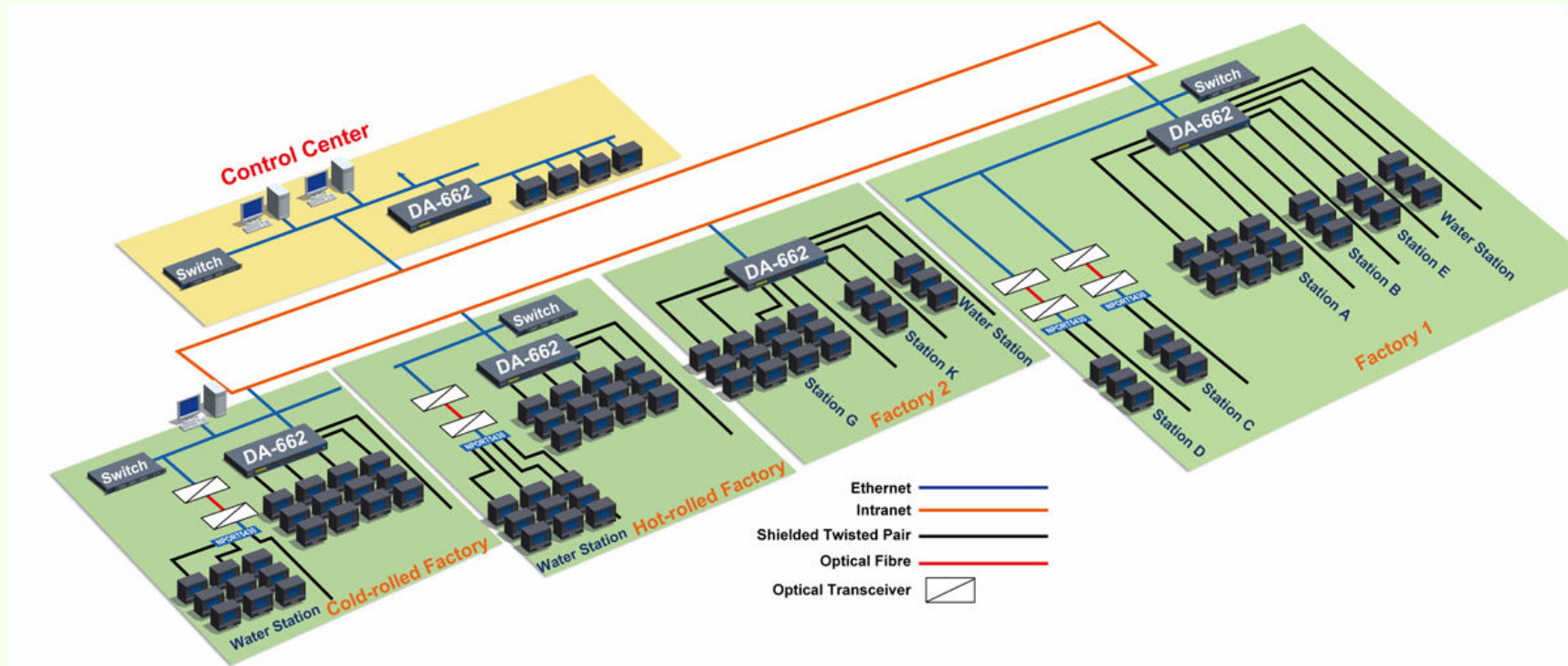
ESTRUTURAS DE HARDWARE



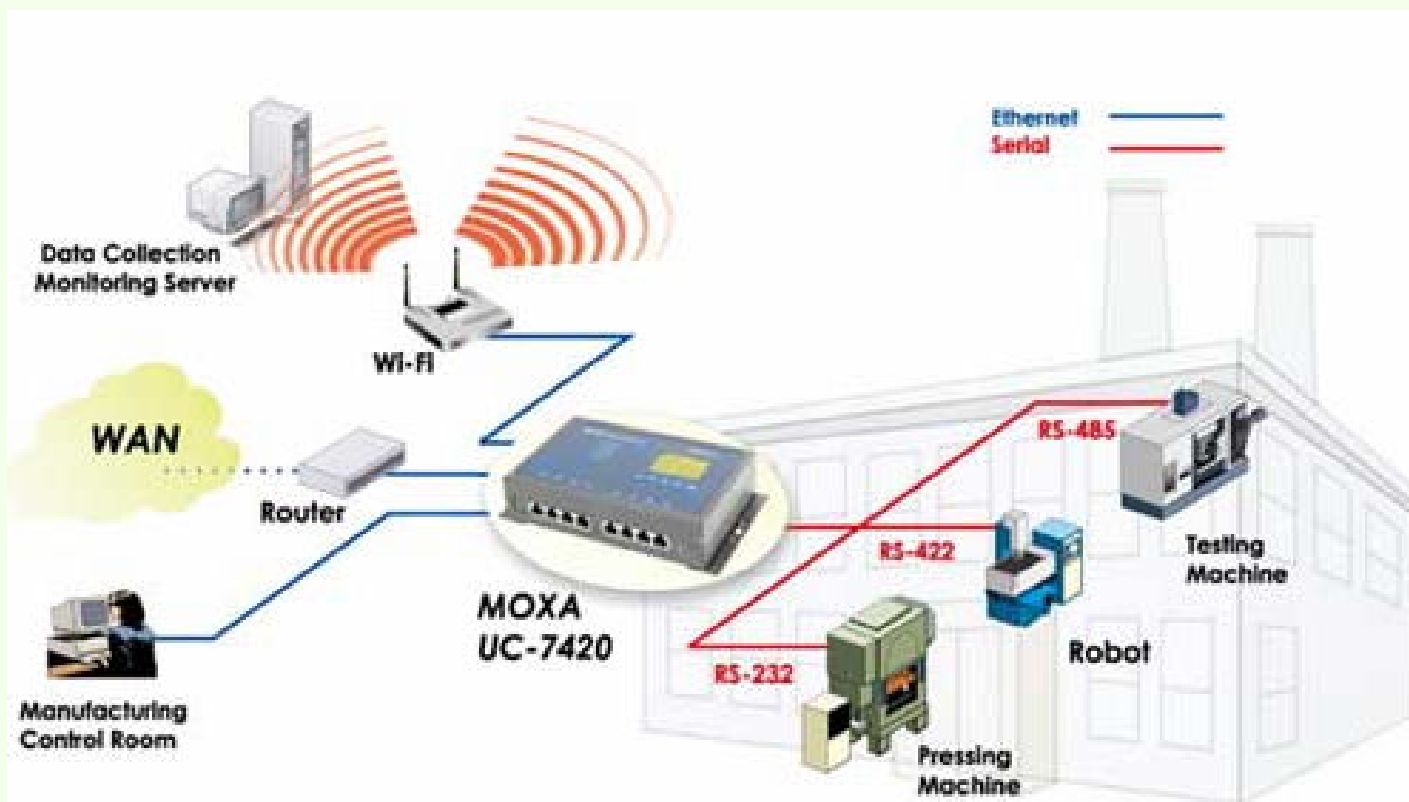
ESTRUTURAS DE HARDWARE



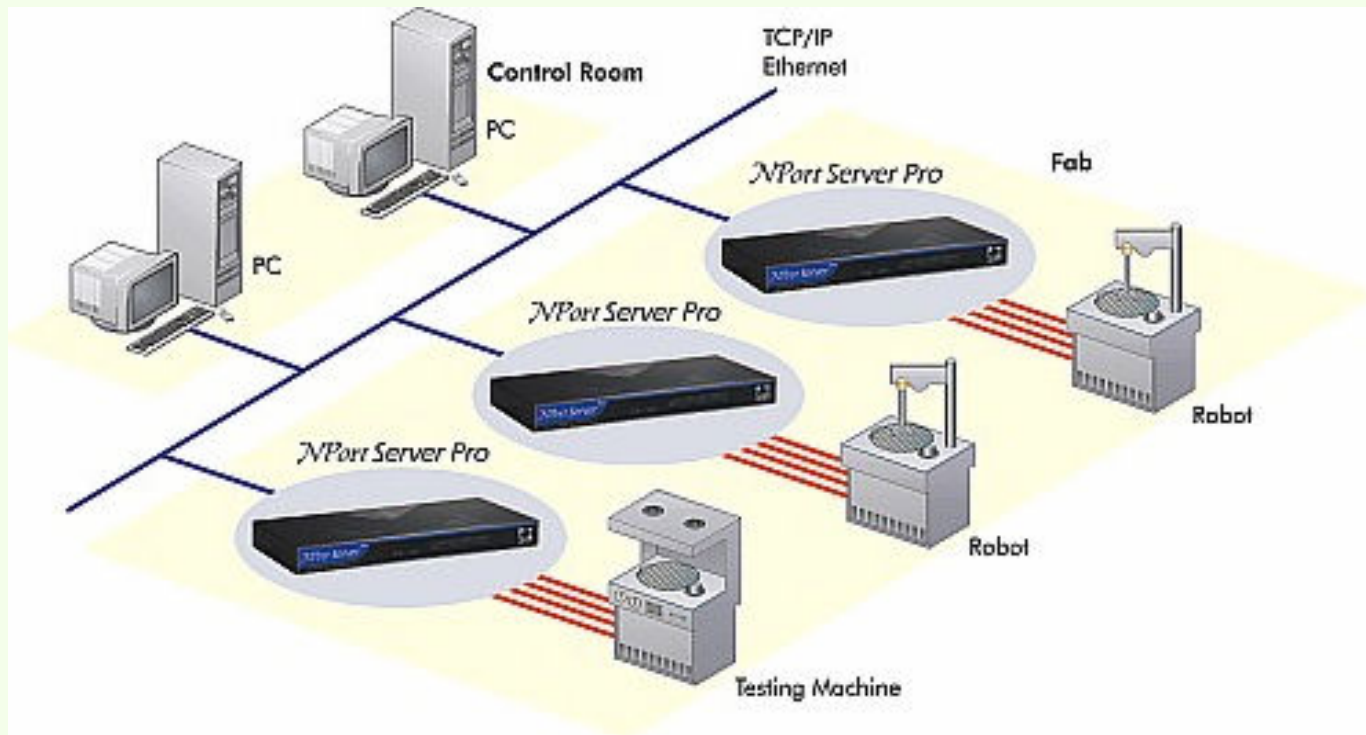
ESTRUTURAS DE HARDWARE



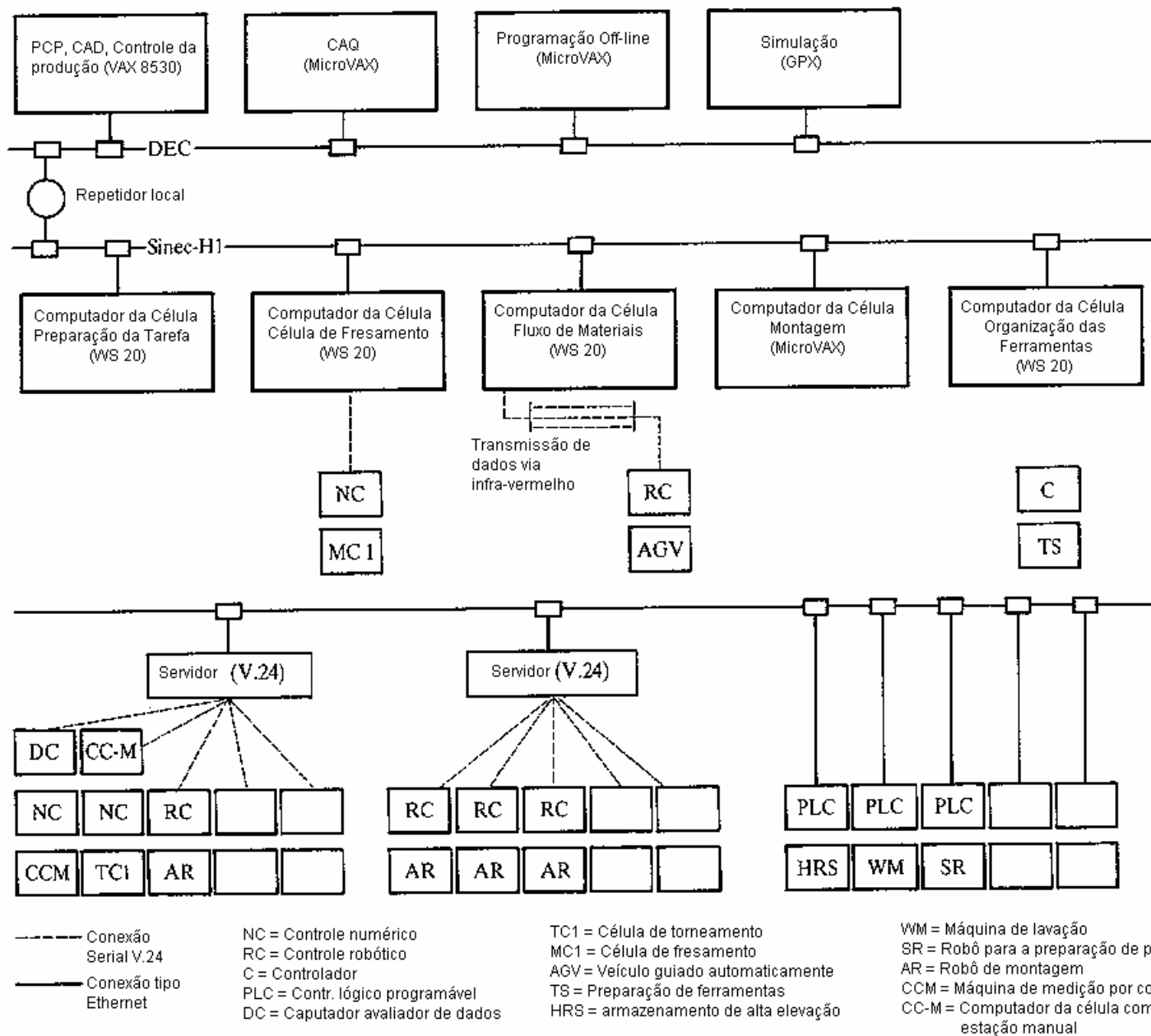
ESTRUTURAS DE HARDWARE



ESTRUTURAS DE HARDWARE

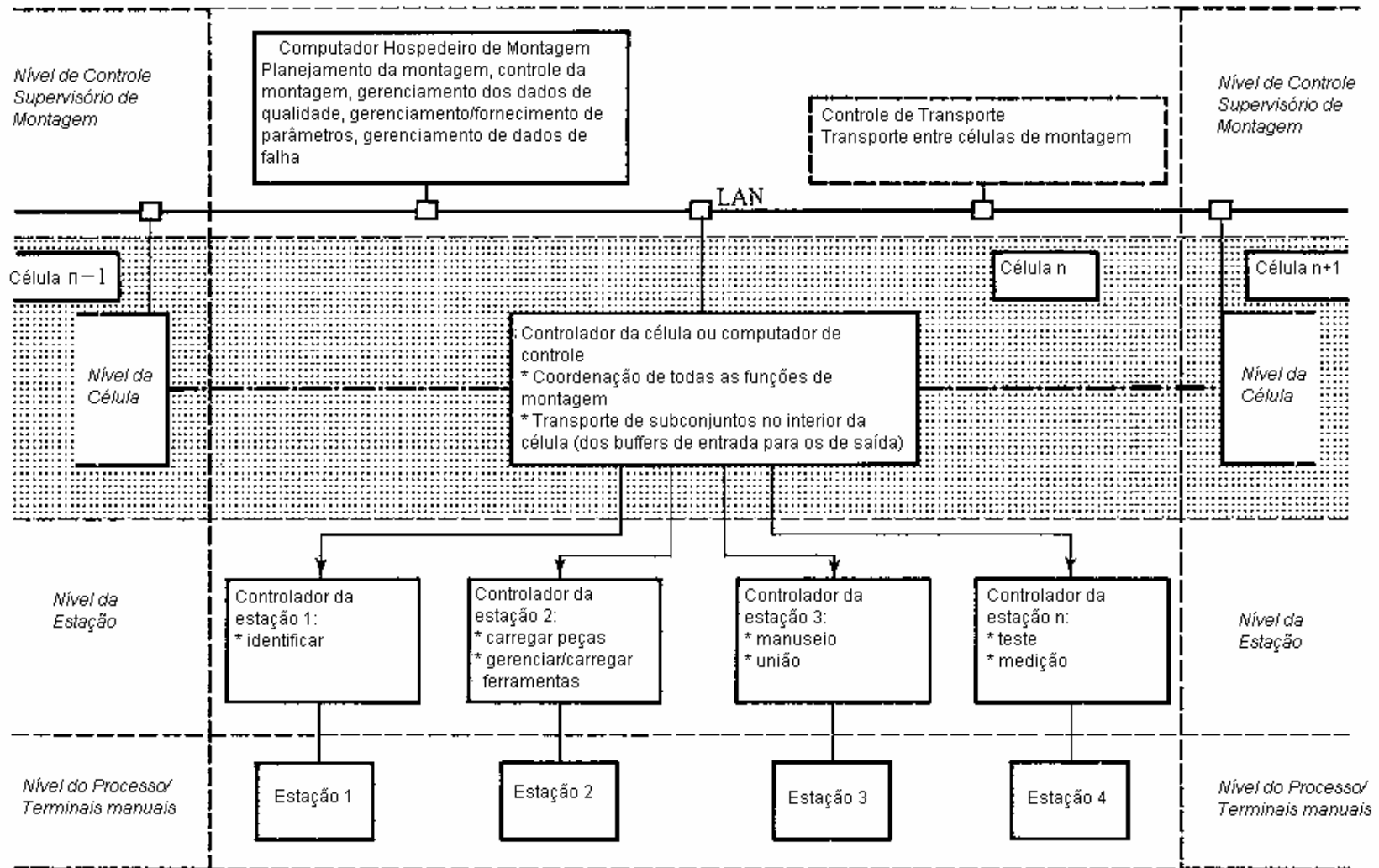


ESTRUTURAS DE HARDWARE



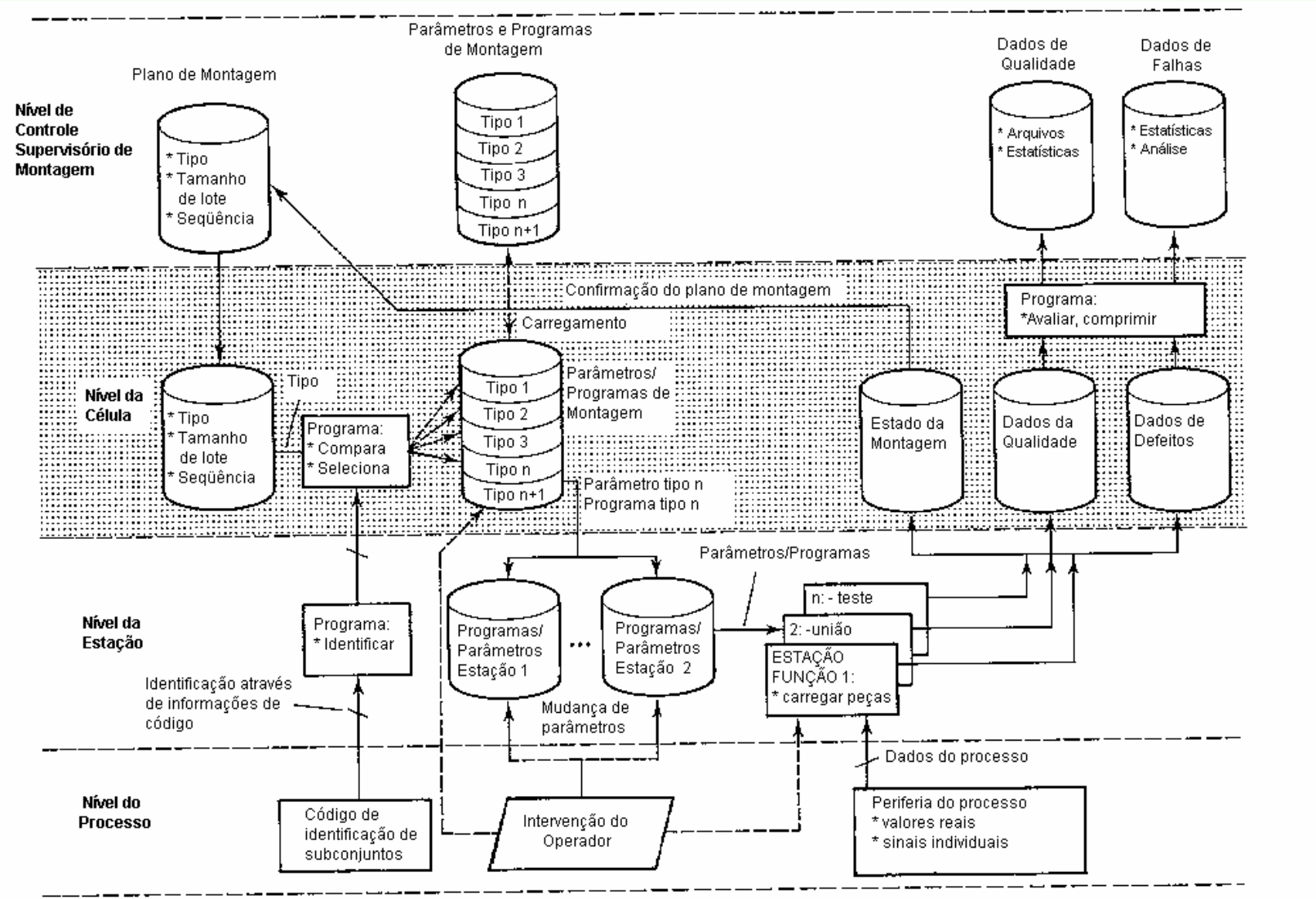
Arquitetura de computadores e comunicação de uma instalação CIM numa instituição em Stuttgart (Alemanha)

ESTRUTURAS DE HARDWARE



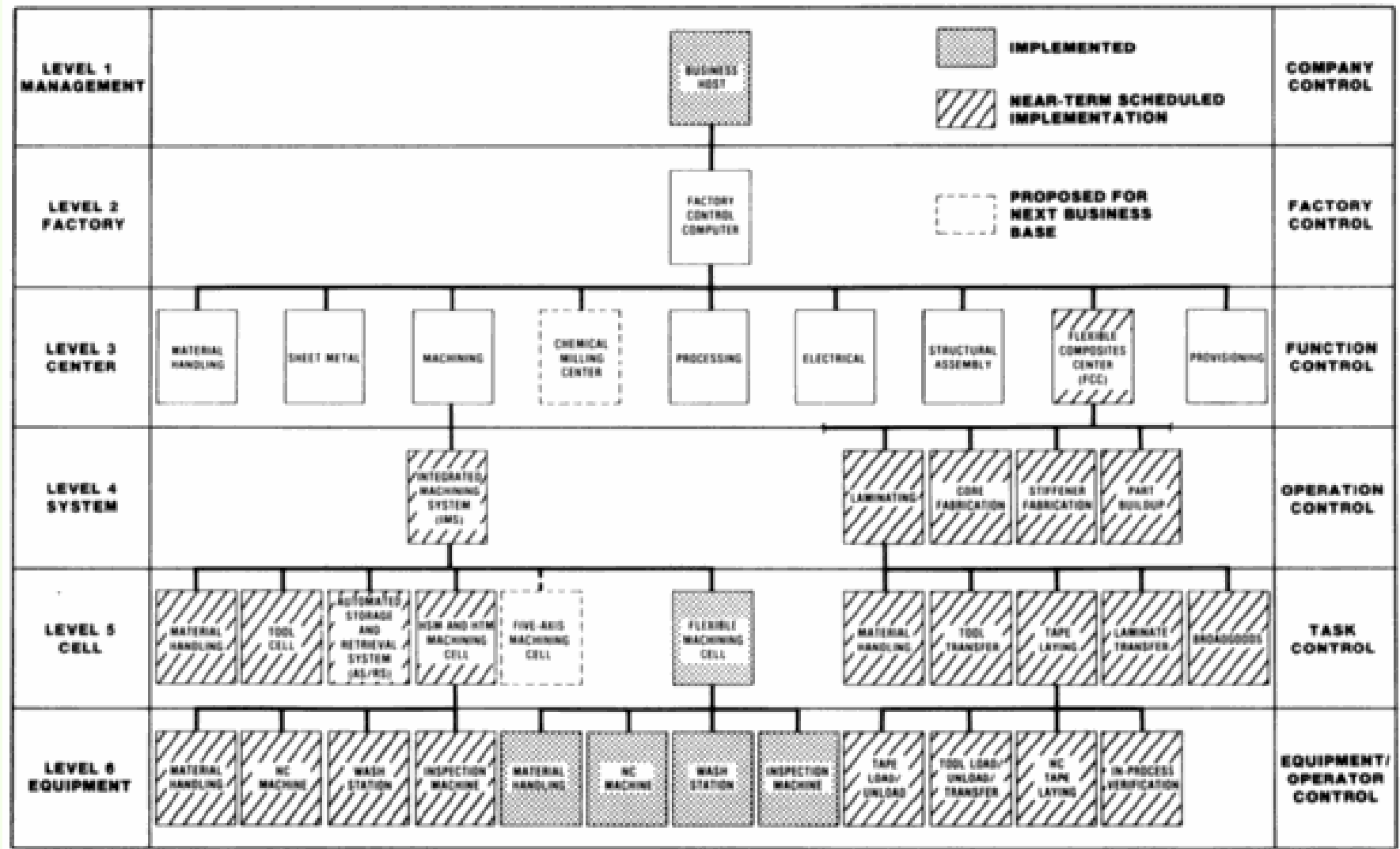
Arquitetura de controle de uma célula de montagem

ESTRUTURAS DE HARDWARE



Fluxo de informações numa célula de montagem

ESTRUTURAS DE HARDWARE



Fluxo de informações numa célula de montagem

ESTRUTURAS DE SOFTWARE

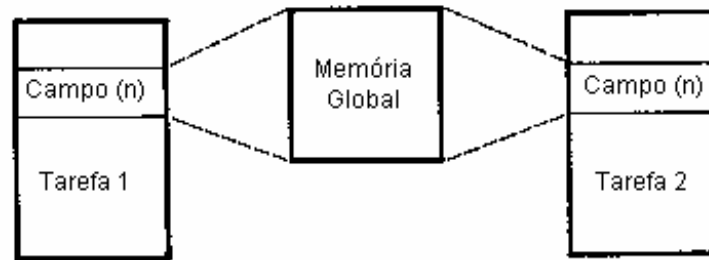
- Mesmo quando FMSs foram desenvolvidos no início → tentativas foram feitas para construir **softwares modulares** → cada módulo era definido de forma precisa, incluindo uma função interna para a sua execução.
- Módulos do usuário para **controlar e monitorar os fluxos de dados e materiais** na fábrica → obtêm suas instruções a partir de **realimentação** do processo de manufatura.
- Várias mensagens de **alarme dos equipamentos e entradas do operador** no chão de fábrica devem ser obtidas.

ESTRUTURAS DE SOFTWARE

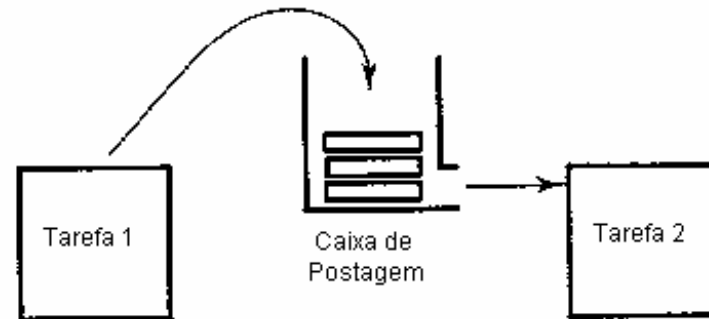
- Para dar suporte à **construção modular** → funções são implementadas em **blocos funcionais** que são executados como programas (**tarefas**) em separado (**independentes**) → exigência de computadores **multi-tarefa** para a sincronização das tarefas.
- Linguagens normalmente utilizadas em sistemas de controle são **Fortran e C** → devem incluir a execução e a troca de dados entre tarefas e com dispositivos de controle orientados ao processo (p.ex. controladores de máquinas).
- Para satisfazer as exigências → chamada a sub-rotinas do sistema operacional.

ESTRUTURAS DE SOFTWARE

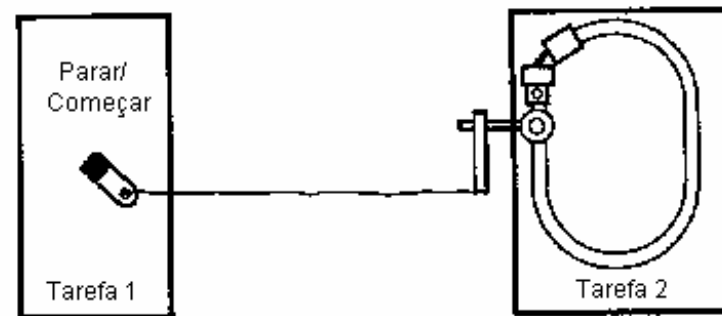
- Sistemas operacionais → fornecem características para uma troca rápida de dados entre tarefas (ver figura):
 - memória compartilhada, seções globais (áreas comuns de dados na memória de trabalho);
 - caixas de postagem (para mensagens).
- Sincronização de tarefas e os dispositivos de controle orientados ao processo pode envolver:
 - interrupção de software;
- Uso de **subrotinas do sistema operacional** → tarefas do sistema de controle tornam-se parcialmente específicas do sistema operacional, e portanto **dependentes do tipo de computador utilizado.**



Troca de dados via memória global



Troca de dados entre caixas de postagem

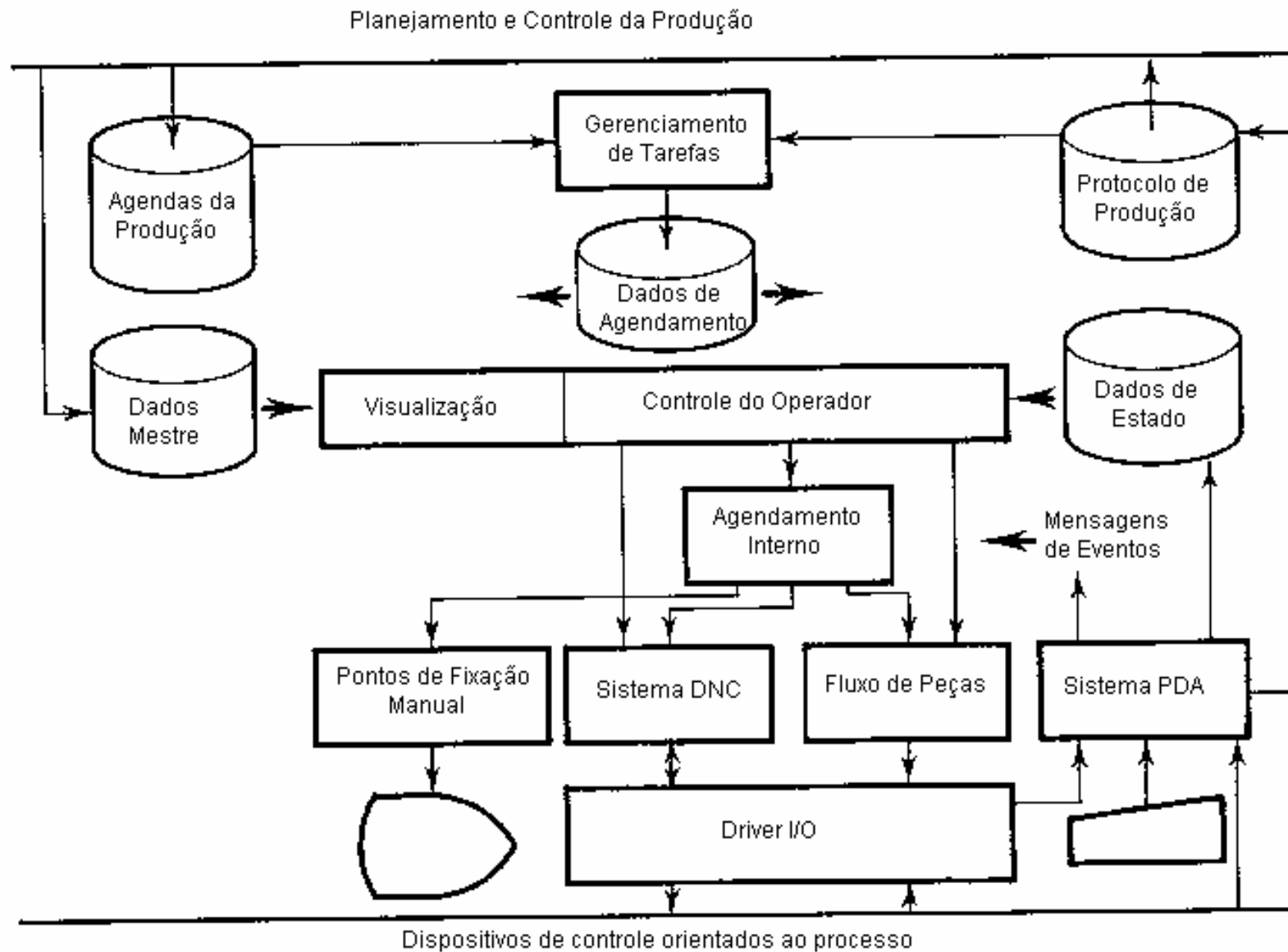


Sincronização da tarefa via interrupções

Possibilidades para a troca de dados e sincronização de tarefas

ESTRUTURAS DE SOFTWARE

- Primeiros sistemas de controle → **estrutura funcional do sistema de controle era mapeado para os programas de controle** (ver figura).
- Nessa estrutura → módulos funcionais possuem várias interfaces.
 - P.ex. **agendamento interno** baseia-se:
 - nas interfaces com os blocos referentes ao fluxo de peças,
 - no sistema DNC,
 - no sistema de aquisição de dados da produção (PDA),
 - no operador,
 - nos dados mestre,
 - nos dados temporários,
 - nos dados de estado.



Componentes de uma estrutura de controle funcional e orientada ao software

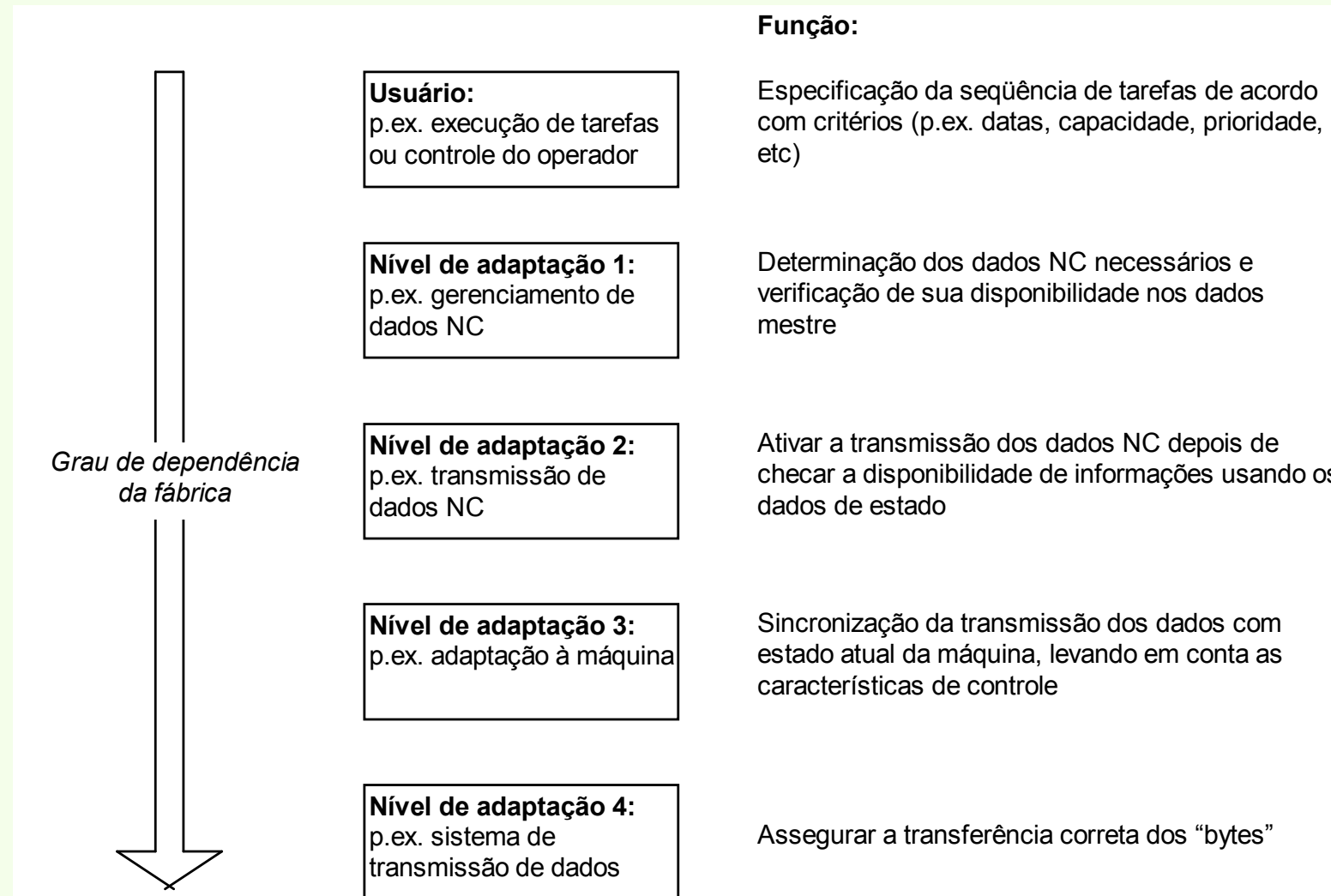
ESTRUTURAS DE SOFTWARE

- Implementação de cada uma destas interfaces individuais requer que o programa chame várias subrotinas do sistema operacional → blocos funcionais individuais podem conter mais do que **50% de comandos específicos do sistema operacional** → teste dos módulos funcionais e a colocação do sistema de controle em operação são **dificultados** por esta estrutura.
- Além disso → estrutura do programa fortemente contém ligações em rede → troca de módulos funcionais ou a **adaptação a uma outra fábrica torna-se praticamente impossível**, e extensões são difíceis.
- Entretanto → projeto de **estruturas flexíveis de produção com controle auxiliado por computador** tem tido um efeito benéfico tanto no desenvolvedor como no usuário, pelo qual mais tentativas têm sido feitas para produzir **software adaptável de controle**.

ESTRUTURAS DE SOFTWARE

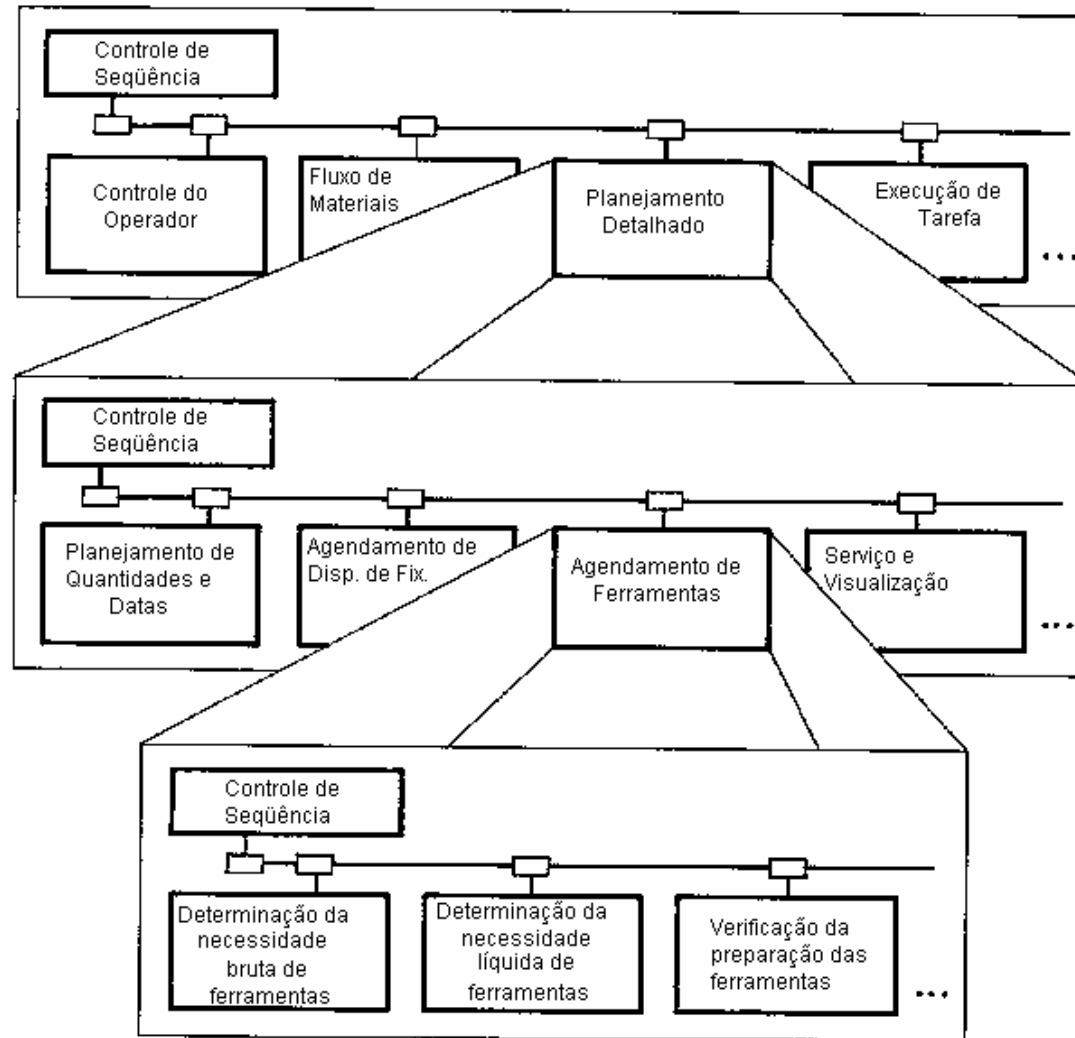
- **Sistemas abertos de controle** devem ser desenvolvidos com as seguintes características:
 - Independência da fábrica: funções de um sistema de controle devem ser especificadas; deve-se buscar o **compartilhamento** de tarefas entre as funções independentes da fábrica e aquelas que são específicas da fábrica.
 - Figura a seguir mostra tal subdivisão para o exemplo de um **DNC**, onde vários níveis de interfaces com diferentes graus de dependência da fábrica são especificados.

ESTRUTURAS DE SOFTWARE



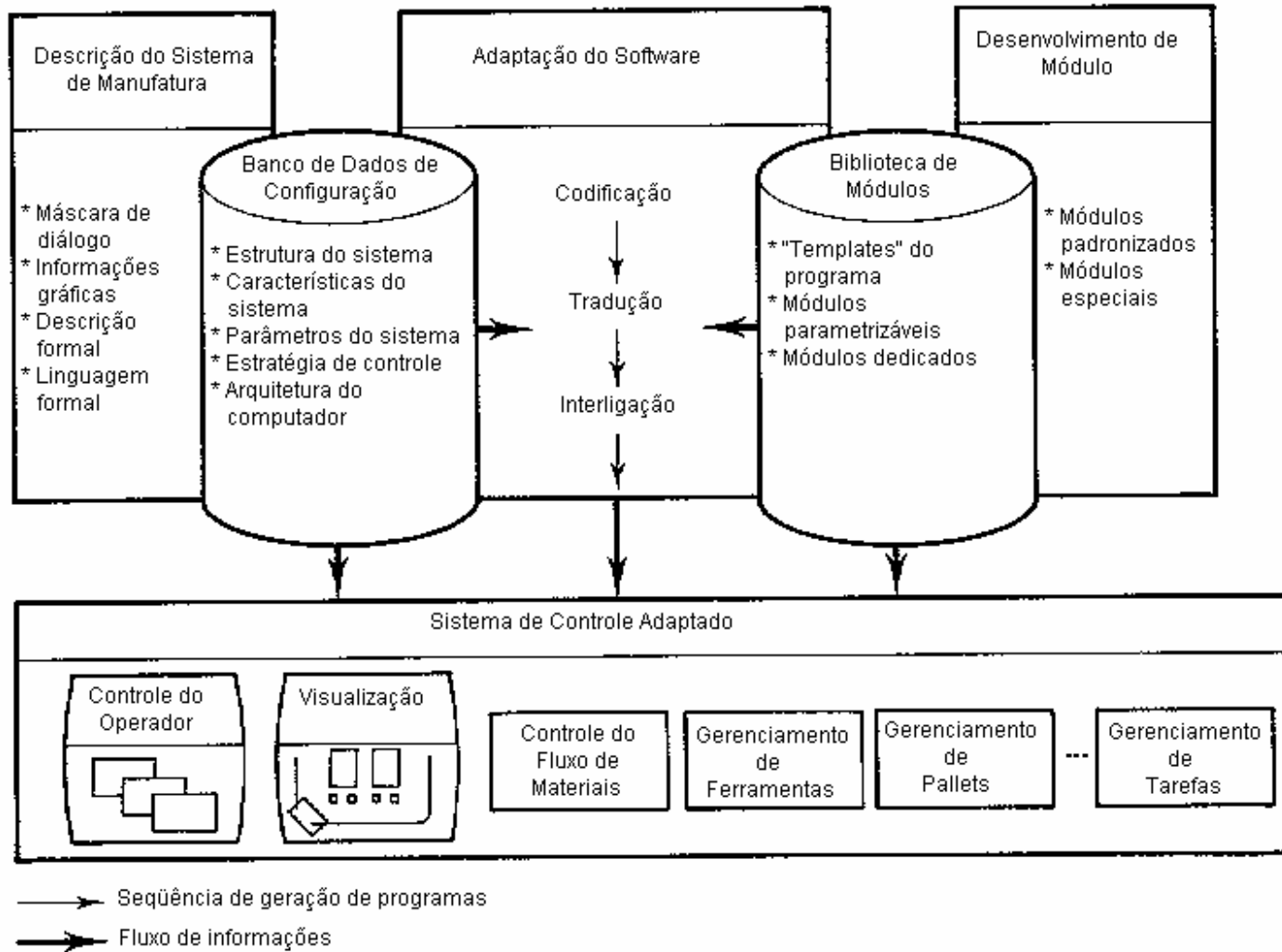
Obtendo-se independência modular através de interfaces padronizadas

ESTRUTURAS DE SOFTWARE



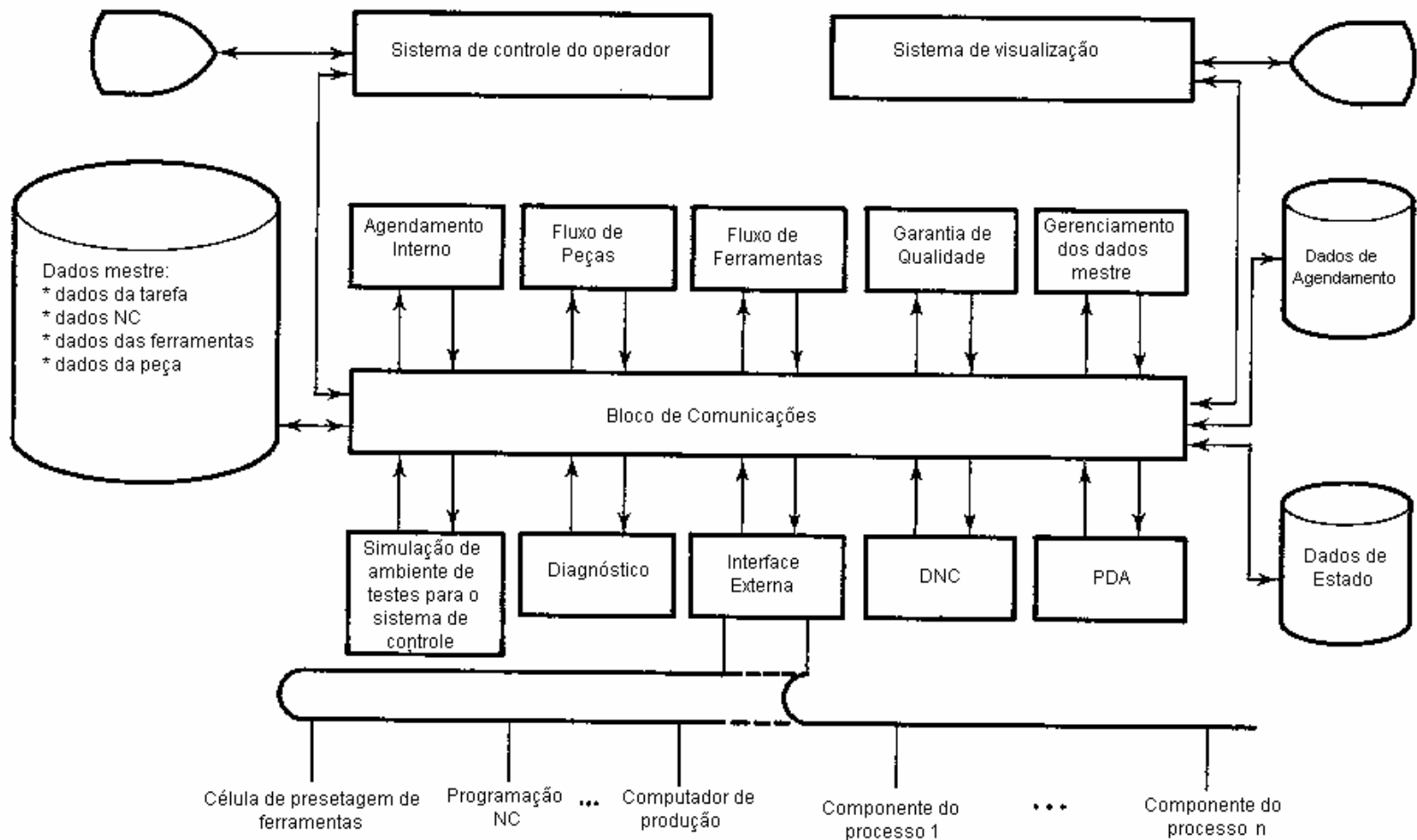
Descrição hierárquica da função de controle “planejamento detalhado”

ESTRUTURAS DE SOFTWARE



Sistema de controle adaptável para a manufatura e montagem flexível

ESTRUTURAS DE SOFTWARE



A estrutura de software do sistema de controle do instituto ISW

ESTRUTURAS DE SOFTWARE

Além da possibilidade do projeto e consolidação gradual, este procedimento também oferece as seguintes vantagens:

- Blocos funcionais do programa de controle podem ser dispostos em **vários computadores**.
- Troca mais fácil de blocos funcionais através de **mensagens padronizadas** entre blocos funcionais e componentes do processo.
- O sistema de controle é mais **modular** e a adaptação a computadores específicos é simplificada.

ESTRUTURAS DE SOFTWARE

- Dados necessários para o sistema de controle são subdivididos nos seguintes grupos: **dados mestre; dados de agendamento (temporários); dados de estado.**
 - Dados mestre são praticamente constantes ao longo do tempo.
 - Inclui (p.ex.):
 - descrição de um acessório de manufatura,
 - dados de planejamento (p.ex. planos de processo).
 - Dados mestre consistem principalmente de dados mais ou menos permanentes que são menos frequentemente acessados → é suficiente armazenar estes dados num **armazenador periférico**, como discos ou fitas.
 - **Acesso simultâneo a estes dados é disponível a um número de programas (tarefas).**

ESTRUTURAS DE SOFTWARE

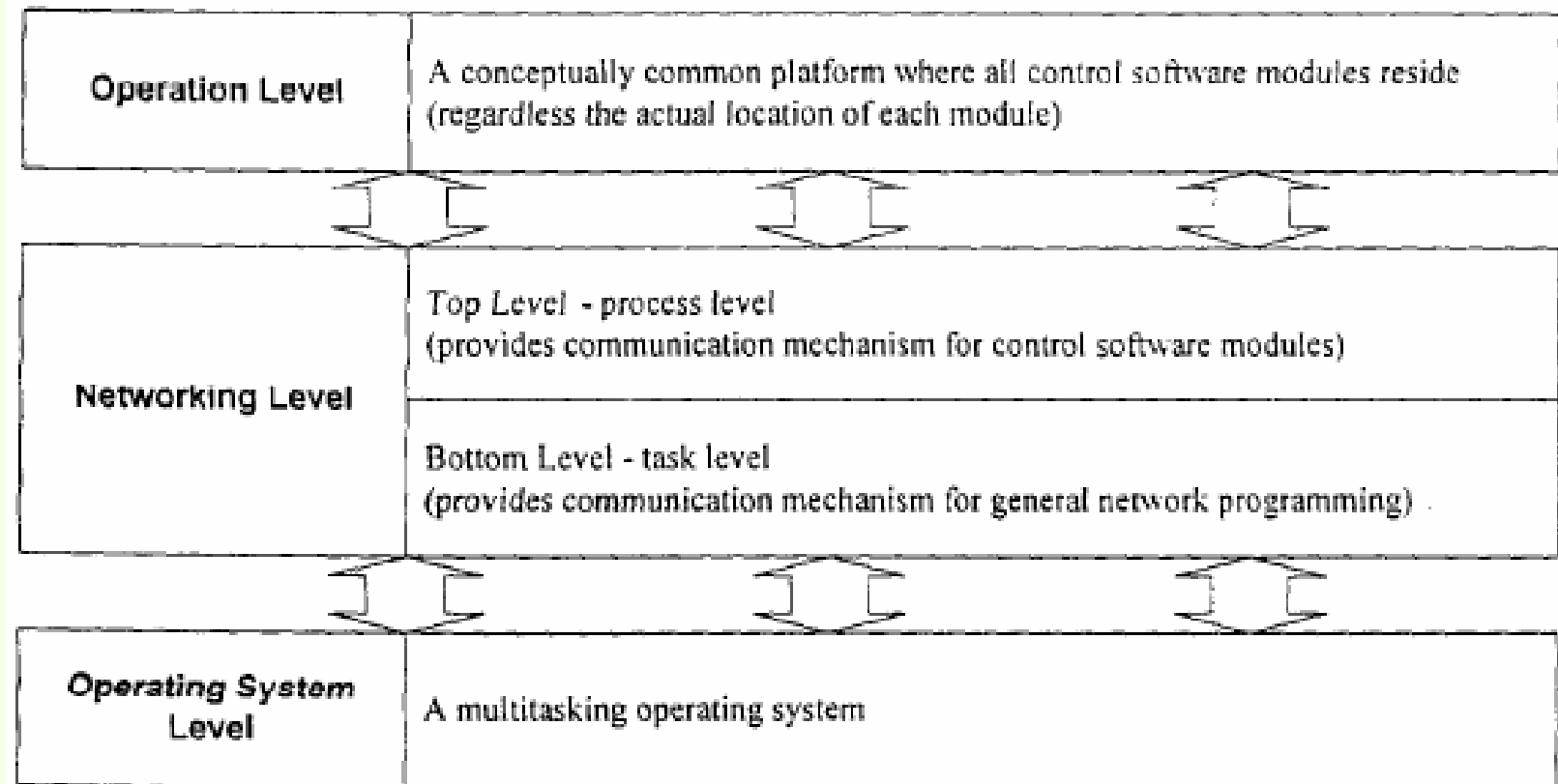
- Dados de agendamento → período de planejamento tal como um **turno, um dia ou um período de manufatura**.
- Exemplos típicos:
 - agendamento de **tarefas**;
 - lista de suprimento de **ferramentas** que declara quais ferramentas devem ser preparadas para o dado período planejado.
- Depois que todas as **avaliações estatísticas** para o dado período forem completadas → dados de agendamento são implicitamente **excluídos**.

ESTRUTURAS DE SOFTWARE

- Dados de estado → todos os dados que **variam no tempo** (p.ex. entradas descrevendo a localização atual e o estado dos acessórios no sistema).
 - Particularidade → definição de um **contorno do sistema**, pois os dados são excluídos logo que um acessório deixa o sistema.
 - Dados de estado (que são como uma “**foto**” do sistema) incluem todos os dados que mudam frequentemente e rapidamente.
 - Em particular → controle do fluxo de materiais & agendamento interno ou planejamento detalhado frequentemente **lêem o estado do sistema** enquanto o sistema de aquisição de dados da fábrica atualiza os dados → portanto, recomenda-se que estes dados não sejam armazenados num **dispositivo periférico** → o acesso à leitura seria **muito lento** e haveria um travamento constante para prever inconsistências → deve-se considerar o armazenamento do estado do sistema **diretamente na memória** de trabalho do computador → segurança: como os dados na memória são perdidos em caso de quebras, o estado do sistema completo é armazenado ciclicamente num dispositivo periférico.

DEPENDÊNCIA DO HARDWARE

ESTRUTURAS DE SOFTWARE



Decomposição hierárquica do software em 3 níveis

DEPENDÊNCIA DO HARDWARE

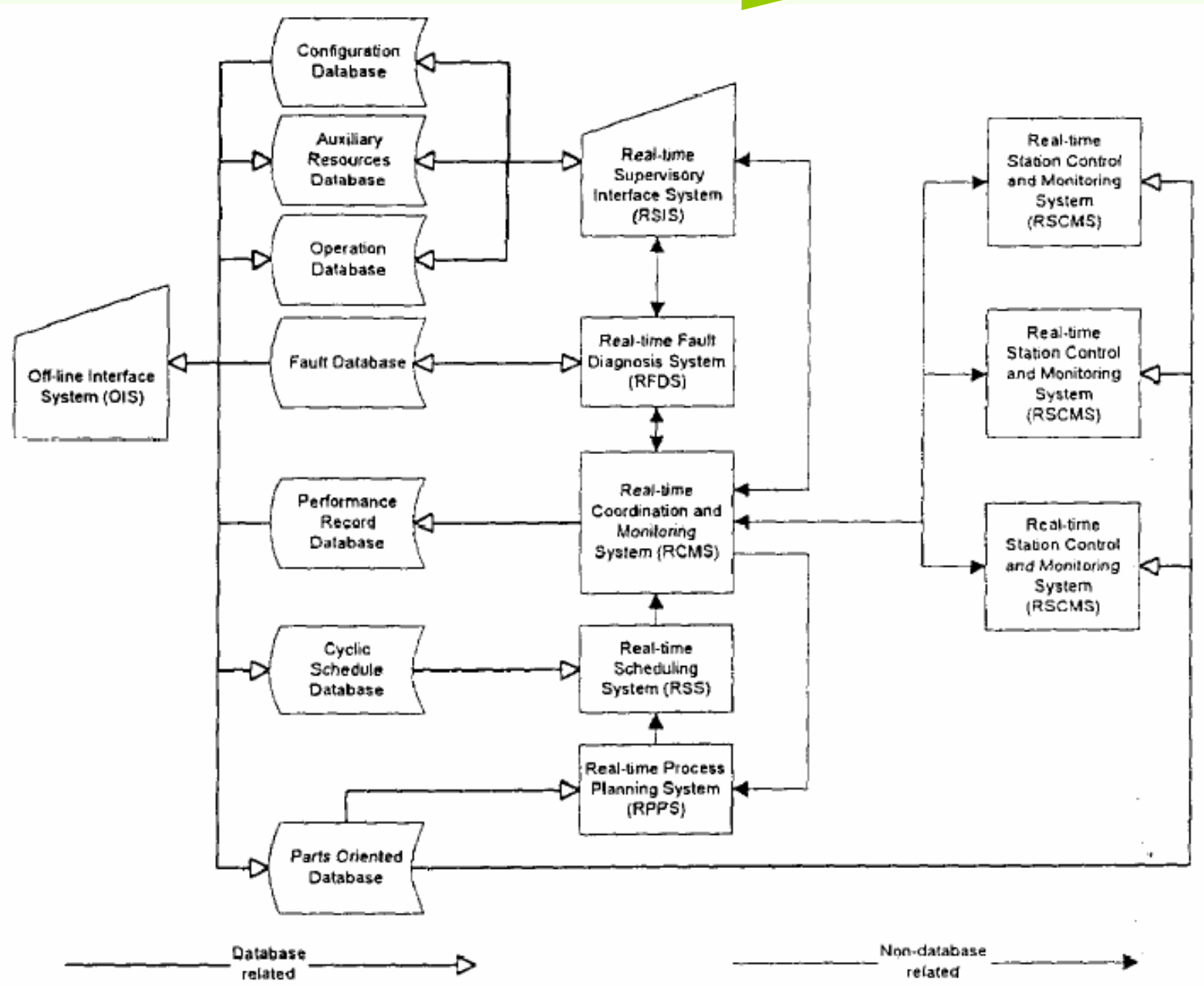
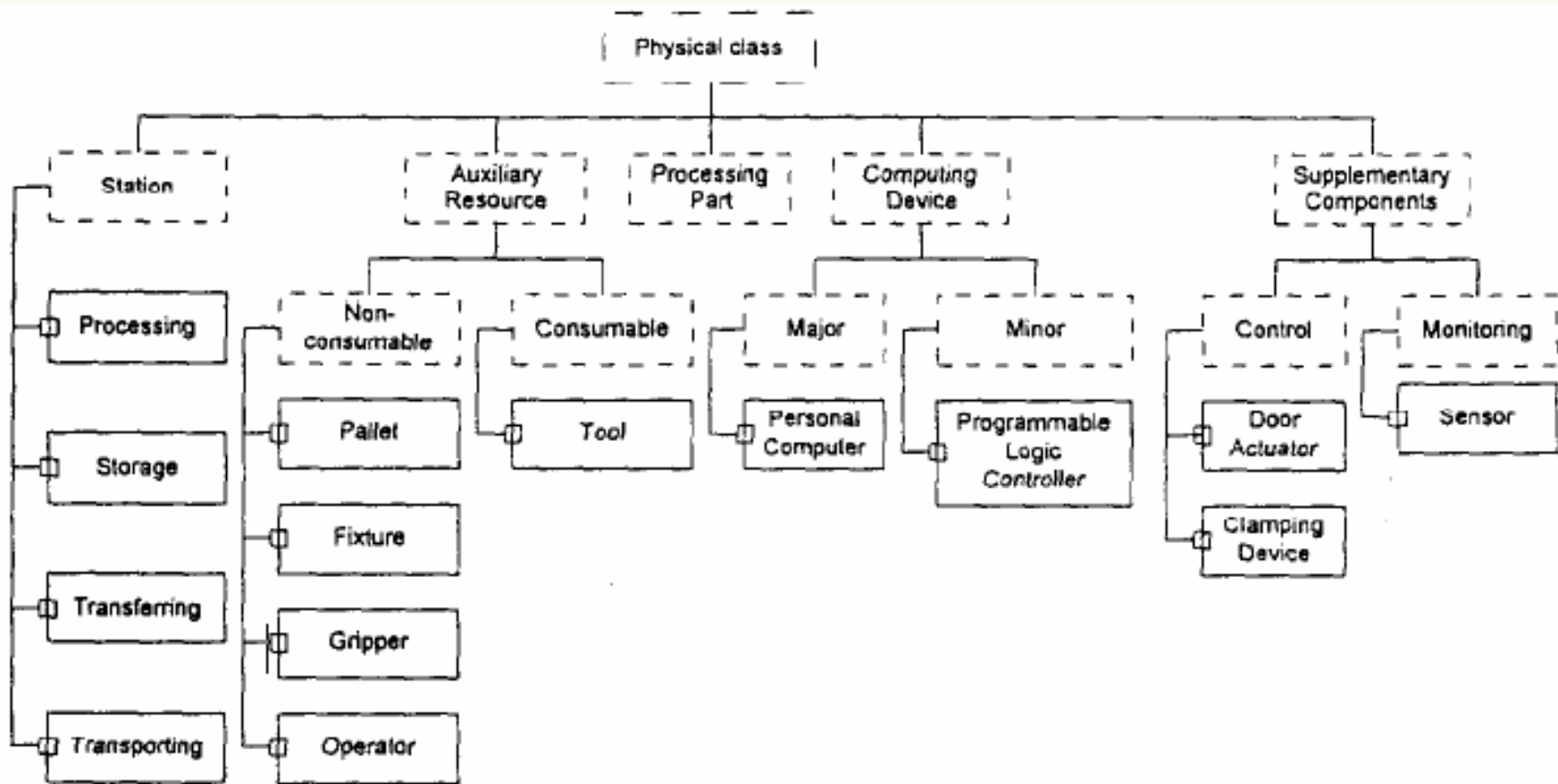


Diagrama de blocos funcionais no nível da Operação

ESTRUTURAS DE SOFTWARE



Hierarquia de classes usando-se Orientação a Objetos

Exemplo de Desenvolvimento do Software de Controle de FMS

INTRODUÇÃO

- Ele deve ser robusto, e deve conferir flexibilidade ao FMS.
- FMSs → equipamentos flexíveis (máquinas CNC, robôs, AGVs, AS/RS)
- Software de controle não considera a inclusão de novas máquinas, peças, mudança nos algoritmos de controle, etc.
- Desenvolver o software de controle para FMS não é uma tarefa fácil.
- Características típicas desse software:
 - específico;
 - custo elevado;
 - difícil de ser modificado;
 - freqüentemente a maior fonte de inflexibilidade de FMSs.

INTRODUÇÃO

- Fontes de inflexibilidade:
 - forte acoplamento das funções executadas pelo software;
 - codificação rígida da lógica de controle;
 - informações sobre o roteamento e seqüencialmente da peça embutidos no software de controle;
 - em geral → falta de uma abordagem estruturada no desenvolvimento do software de controle.
- Maioria dos FMSs são vendidos a empresas de manufatura como sistemas "prontos", adquiridos dos fornecedores → conhecimento do software não reside nas empresas usuárias, e mudanças na lógica/software podem ser feitas apenas pelo fornecedor do FMS.

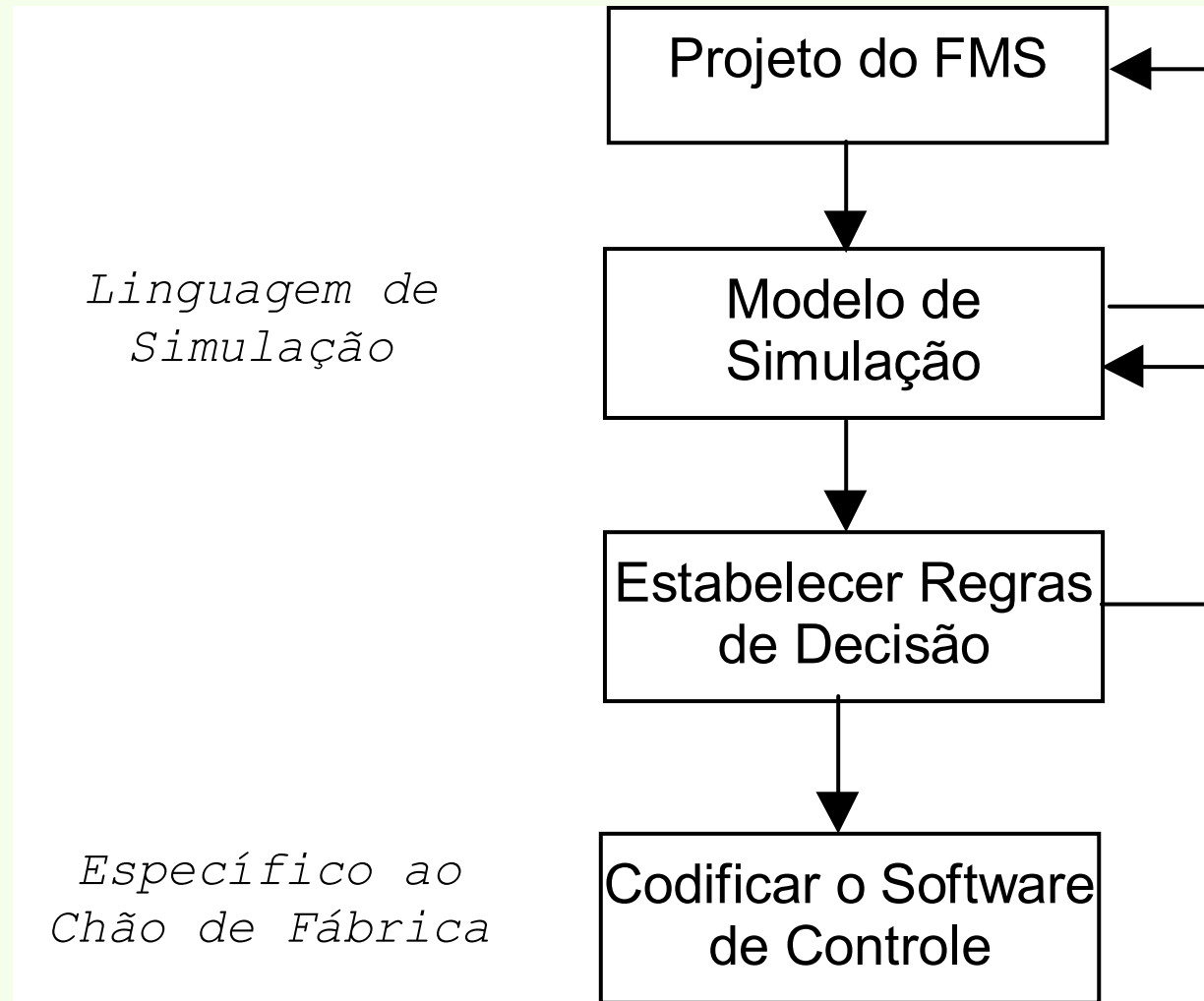
INTRODUÇÃO

- Melhor solução ? → permitir aos operadores do sistema e/ou engenheiros de manufatura modificar o software de controle, no caso de mudanças físicas no sistema e exigências da produção, usando ferramentas de desenvolvimento.
- Estas mesmas ferramentas de software podem ser usadas pelos "integradores", para reduzir o tempo, esforço e custo de desenvolvimento do software.
- Serão discutidos os conceitos e abordagem de desenvolvimento do software de controle de FMS, e também ferramentas de software para auxiliar esse desenvolvimento ⇒ RapidCIM

OBJETIVOS

- Tornar um FMS operacional (i.e. totalmente automatizado, controlado por computador) logo após a chegada e posicionamento dos equipamentos.
- Conceito que permite definir módulos genéricos de controle, com interfaces bem definidas:
 - desenvolvidos de uma maneira independente;
 - customizados com base em dados específicos da instalação;
 - baseados em modelos formais de comportamento do controlador;
 - adequado para a geração automática dos módulos do software de controle;
 - reconfiguráveis para a adaptação a mudanças no ambiente, a partir da modificação nos dados, e a conseqüente regeneração do software de controle.
- Modelo tradicional para o desenvolvimento do software de controle de FMS (ver próxima figura).

OBJETIVOS

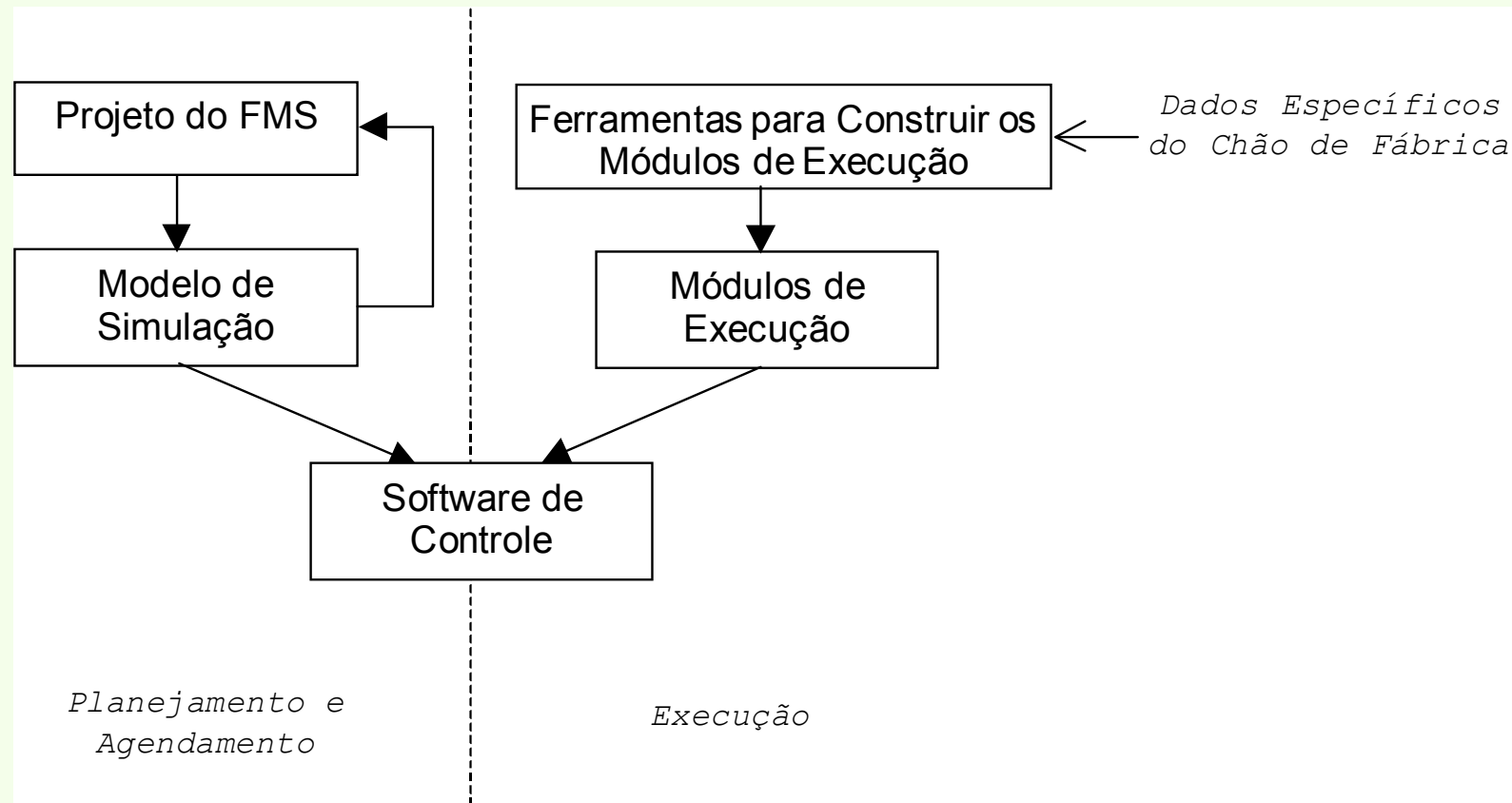


Abordagem tradicional para o desenvolvimento de software de controle de FMS

OBJETIVOS

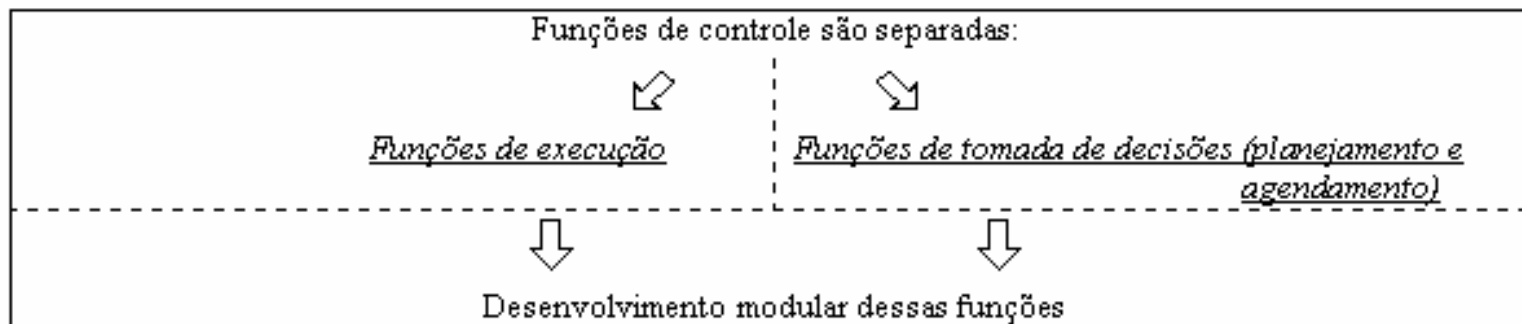
- Modelo de Simulação do FMS → avaliação da:
 - configuração de hardware (nº de máquinas, pallets, *buffers*, etc.);
 - lógica do fluxo de peças (regras de expedição, algoritmo de agendamento, etc.) segundo alguns critérios de desempenho;
- Depois de decidir-se o hardware e a lógica do fluxo de peças → inicia-se o desenvolvimento do software de controle.
 - lógica de controle é implementada numa certa linguagem (p.ex. C++), e tipicamente escrita do nada.
 - freqüentemente a lógica de controle e a lógica de execução são entrelaçadas.
 - não há um modelo padrão aceito para estruturar o software de controle → o software torna-se dependente da instalação, inflexível, não portátil.
- Abordagem proposta → ver próxima figura.

OBJETIVOS



Abordagem proposta

OBJETIVOS



Abordagem proposta

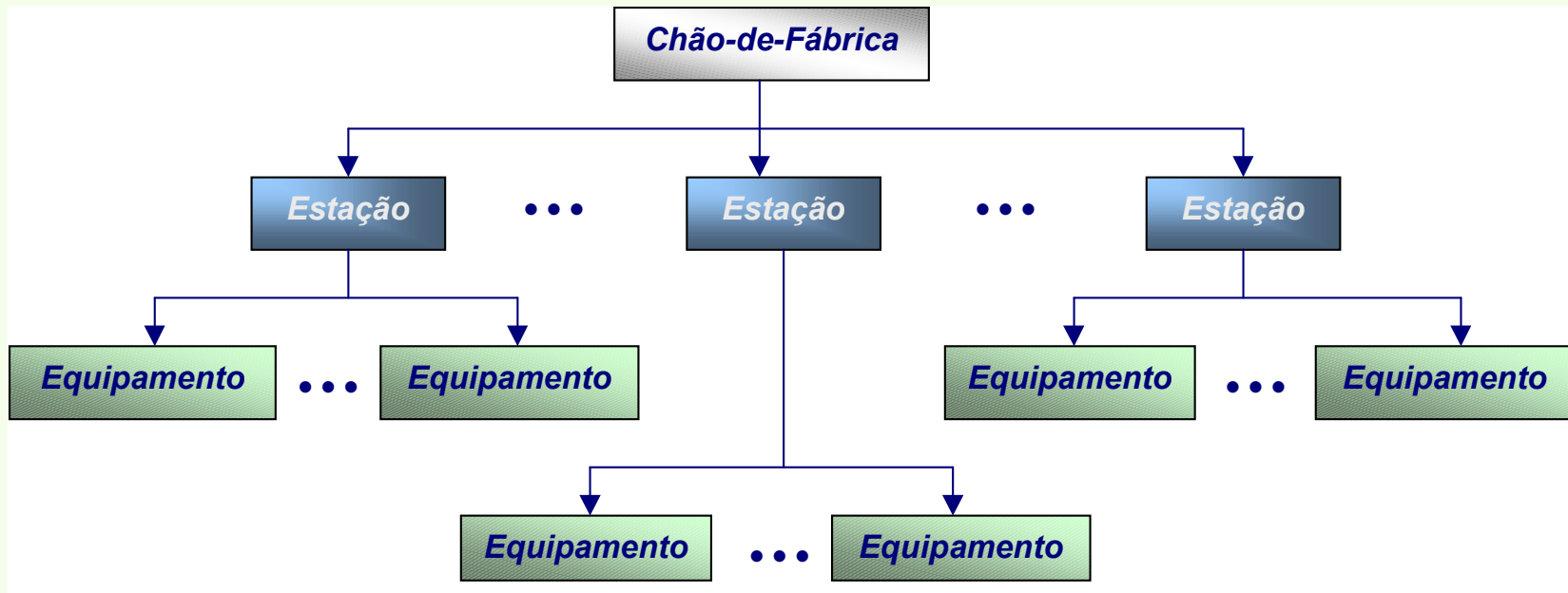
OBJETIVOS

- Porções de execução dos controladores podem ser geradas automaticamente usando as ferramentas desenvolvidas:
 - estrutura formal para os módulos de execução;
 - metodologia bem definida de desenvolvimento → reduzindo o tempo e esforço de desenvolvimento;
 - uso de um modelo de simulação para as funções de tomada de decisão → o modelo é desenvolvido somente 1 vez, e usado para:
 - simulação ("off-line");
 - controle em tempo real ("on-line") → redução ainda maior no tempo de desenvolvimento do módulo de tomada de decisões do controlador.

OBJETIVOS

- **Componentes do RapidCIM:**
 - Arquitetura de controle do chão de fábrica → deve fornecer um procedimento para o projeto e construção de um sistema de controle do chão de fábrica (hierarquia mostrada na próxima figura).

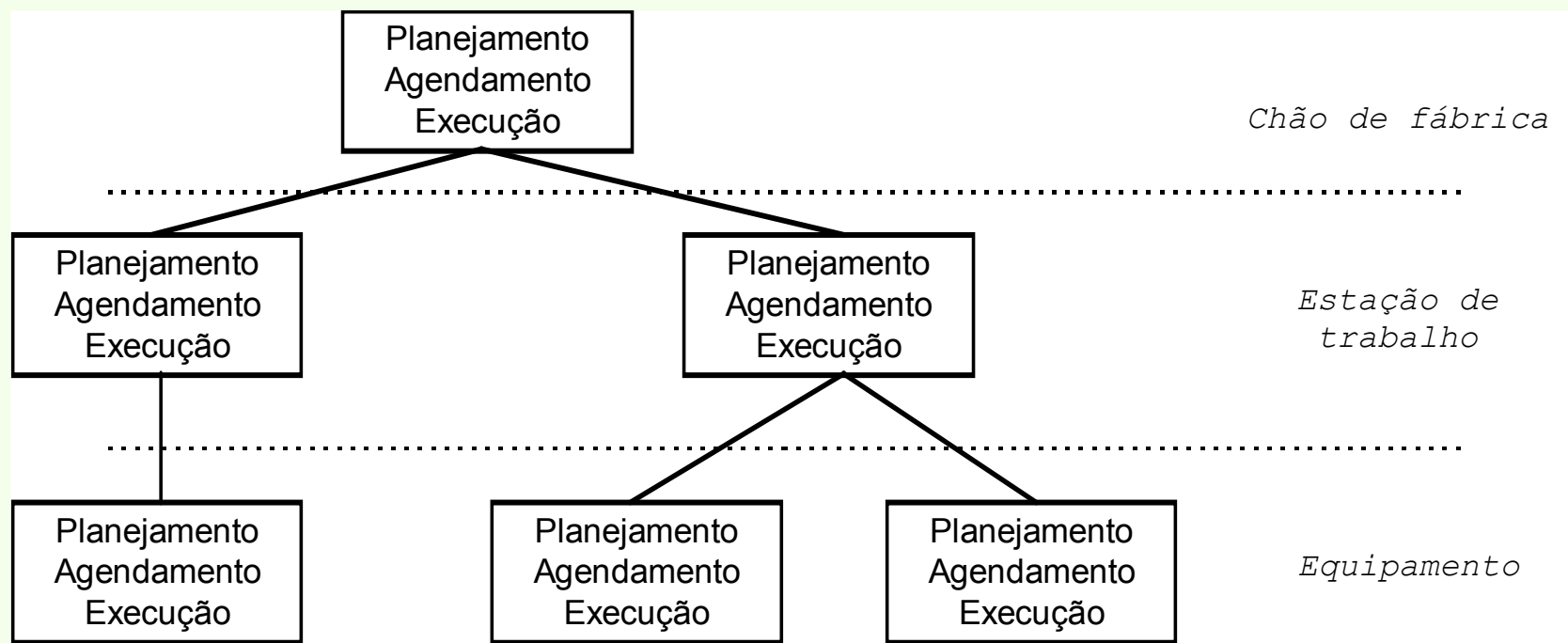
OBJETIVOS



Estrutura hierárquica do controlador do FMS

OBJETIVOS

- Funções de cada controlador → planejamento, agendamento, execução → estrutura genérica do controlador nesta arquitetura (ver figura abaixo).

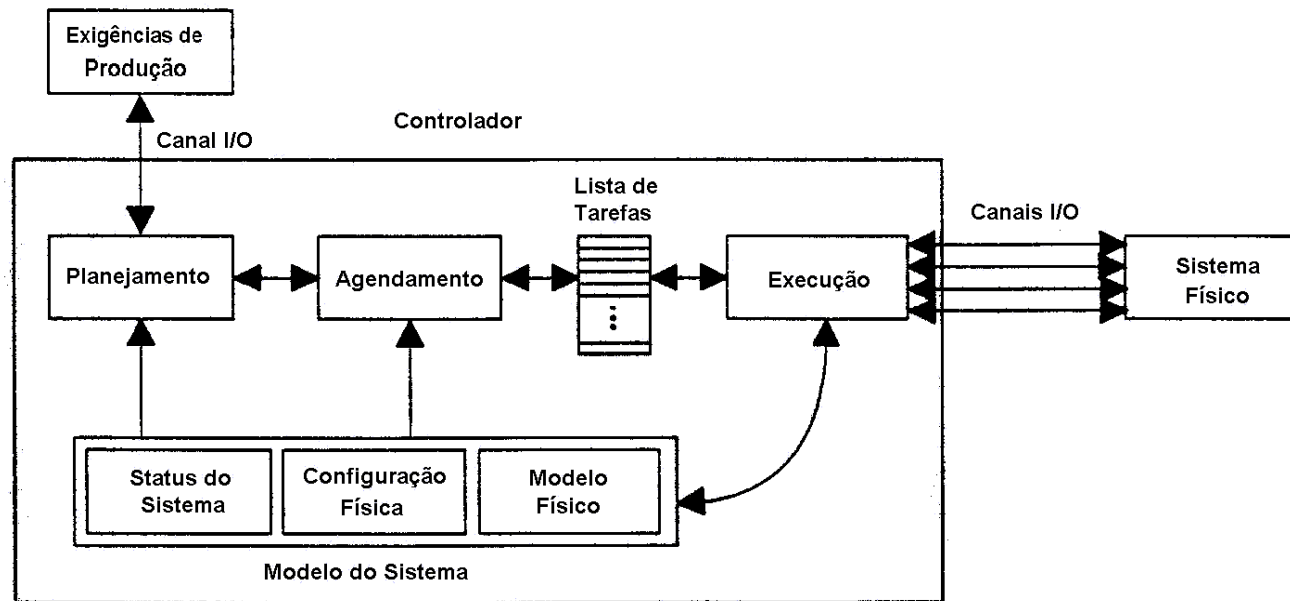


Estrutura hierárquica do controlador do FMS

OBJETIVOS

- Nível do equipamento → o mais baixo na hierarquia → existe somente um controlador no nível do equipamento para cada equipamento no sistema.
- Nível da estação → subconjunto de equipamentos que interagem diretamente entre si (p.ex. robô com máquina CNC → operações de carregar e descarregar).
- Nível do chão de fábrica → mais alto na hierarquia → responsável por coordenar as atividades nas estações; fornece também a interface do usuário com o sistema de controle.
- Função de planejamento → determina o que o controlador deve fazer para satisfazer as exigências.
- Agendamento → determina uma "boa" seqüência das tarefas planejadas
- Execução → executa as tarefas individuais.
- RapidCIM → sistema de controle de chão de fábrica "**plug and play**" que pode ser facilmente criado e reconfigurado.

OBJETIVOS



Estrutura do controlador

MODELO DOS RECURSOS DA FÁBRICA E PLANOS DE PROCESSOS

- Modelo de Recursos → descreve os equipamentos do chão de fábrica + ferramentas + dispositivos de fixação + relações entre eles.
- É uma das partes onde o usuário introduz dados específicos para desenvolver o sistema de controle.
- Classes → **MP, MH, MT, AS**
- Planejamento do Processo → outra fonte para a implementação de dados específicos (mantido separado do software de controle) → desacoplar o fluxo e o roteamento de peças ao software de controle → pode-se assim modificar o "part mix" e os dados do fluxo de peças, sem modificar o software de controle.

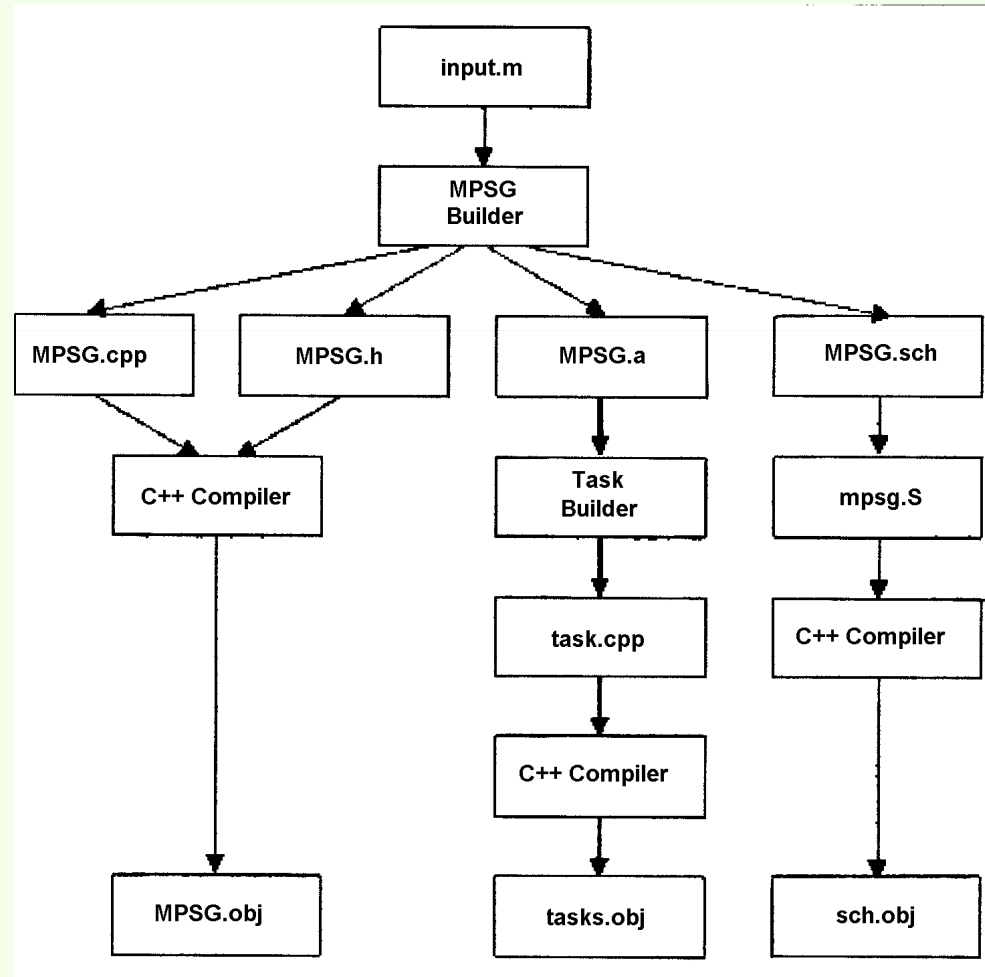
MODELO DOS RECURSOS DA FÁBRICA E PLANOS DE PROCESSOS

- Planos de processos → devem ser estruturados e representados de uma maneira tal que permitam seu uso em todos os níveis de controle, e forneçam detalhes consistentes com as funções de execução disponíveis em cada nível.
- Com alternativas!

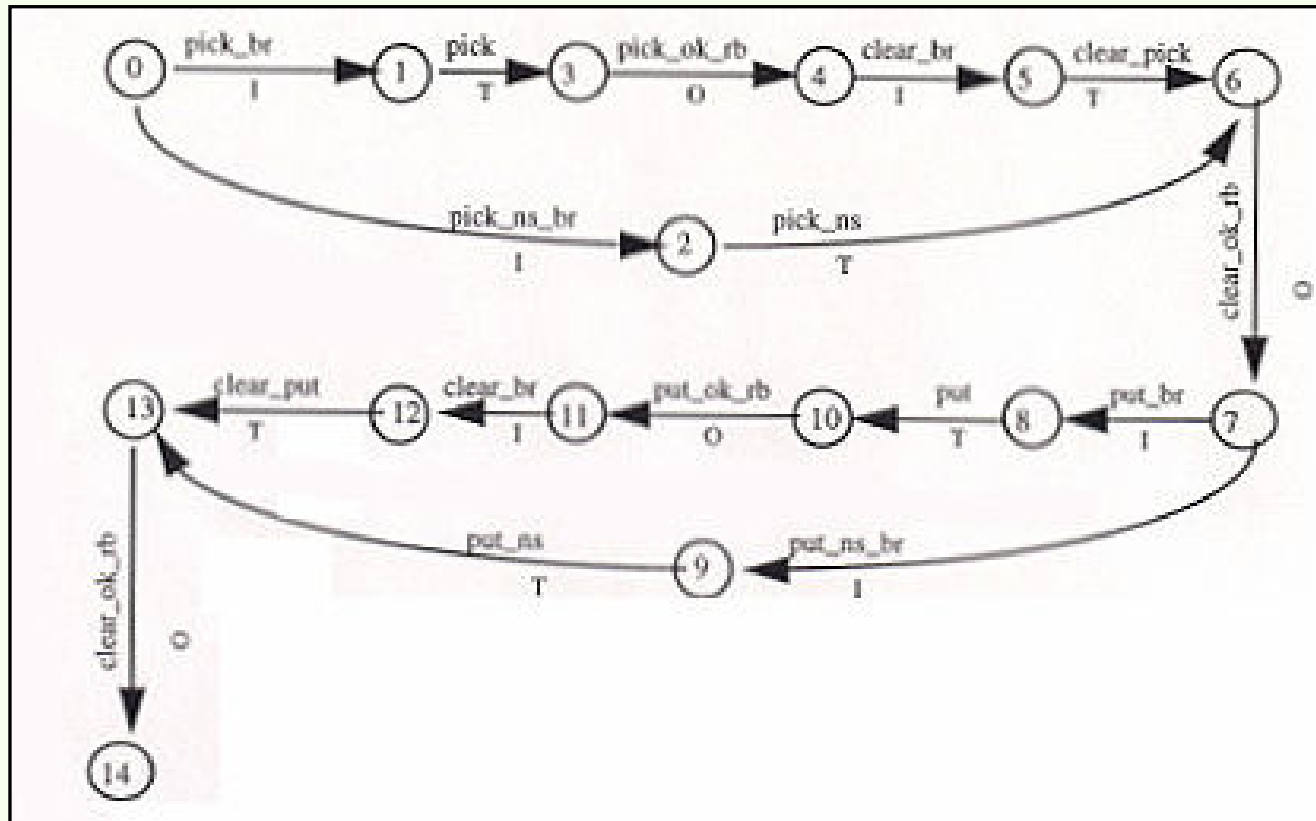
MODELOS FORMAIS DE EXECUÇÃO

- MPSG → "Message-based Part State Graph" → autômato finito \mathbb{R} usado para descrever a interação entre o módulo executor e o sistema físico.
- Controle supervisorio num ambiente de controle distribuído é executado enviando-se/recebendo-se mensagens para/de outros controladores, e efetuando ações físicas.
- Exemplo de ações físicas:
 - enviar um sinal para a máquina CNC (**MP**) para iniciar um ciclo de usinagem;
 - mover o robô (**MH**) para a máquina para descarregar a peça;
 - transportar uma peça entre 2 posições usando um AGV (**MT**).

MODELOS FORMAIS DE EXECUÇÃO



Estrutura do software de construção do Executor

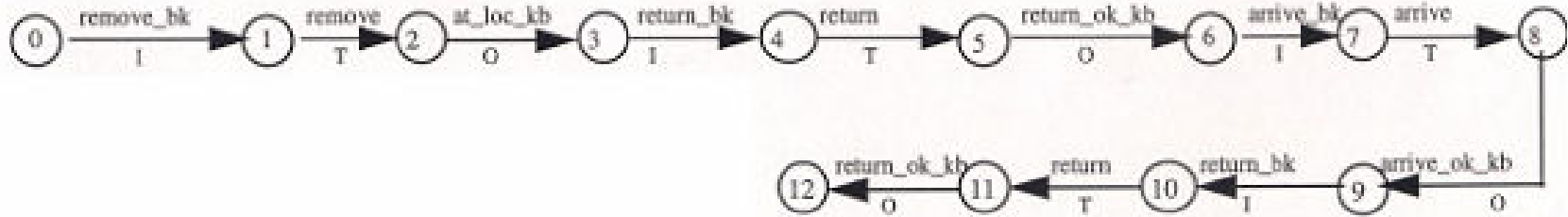


```

%% definitions
%class CONTROLLER_B
%object M1
%output M1
%states 16
%in 5
%out 3
%tasks 5
%final 14
%assign pick_br pick_ns_br
%%
%% in arc transitions
0 pick_br 1
0 pick_ns_br 2
4 clear_br 5
7 put_br 8
7 put_ns_br 9
11 clear_br 12
%%
%% out arc transitions
3 pick_ok_rb 4
6 clear_ok_rb 7
%% in arc transitions
0 pick_br 1
0 pick_ns_br 2
4 clear_br 5
7 put_br 8
7 put_ns_br 9
11 clear_br 12
%%
%% out arc transitions
3 pick_ok_rb 4
6 clear_ok_rb 7
    
```

MPSG genérico para um MH

MODELOS FORMAIS DE EXECUÇÃO



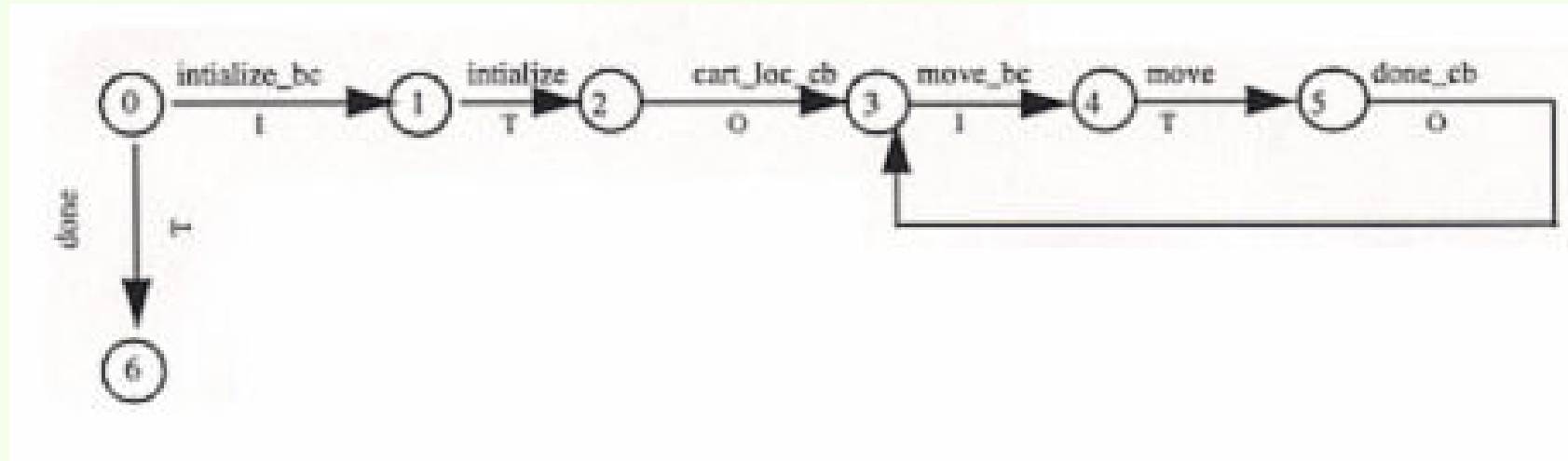
```

%% definitions
%class CONTROLLER_B
%object KARDEX
%output kardex
%states 12
%in 3
%out 3
%tasks 3
%final 12
%assign remove_bk

%%
%% in arc transitions
0 remove_bk 1
3 return_bk 4
6 arrive_bk 7
9 return_bk 10
%%
%% out arc transitions
2 at_loc_kb 3
5 return_ok_kb 6
8 arrive_ok_kb 9
11 return_ok_kb 12
%%
%% task arc transitions
1 remove 2
4 return 5
7 arrive 8
10 return 11
%%
    
```

MPSG genérico para um AS

MODELOS FORMAIS DE EXECUÇÃO



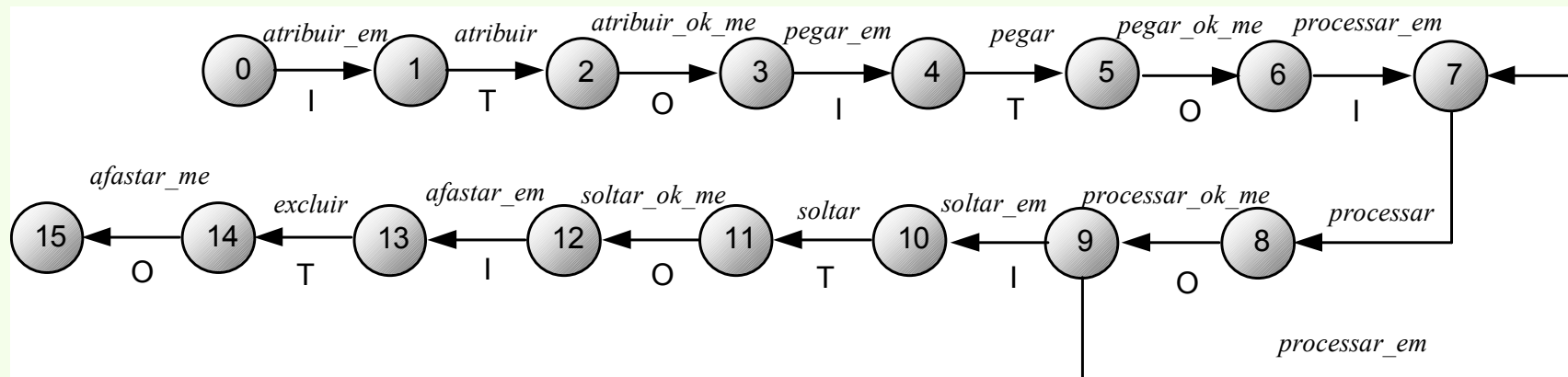
```

%% definitions
%class CONTROLLER_B
%object cartrac
%output cartrac
%states 7
%in 2
%out 2
%tasks 3

%final 6
%assign initialize_bc done
%%
%% in arc transitions
0 initialize_bc 1
3 move_bc 4
%%
%% out arc transitions
2 cart_loc_cb 3
5 done_cb 3
%%
%% task arc transitions
0 done 6
1 initialize 2
4 move 5
%%
    
```

MPSG genérico para um MT

MODELOS FORMAIS DE EXECUÇÃO

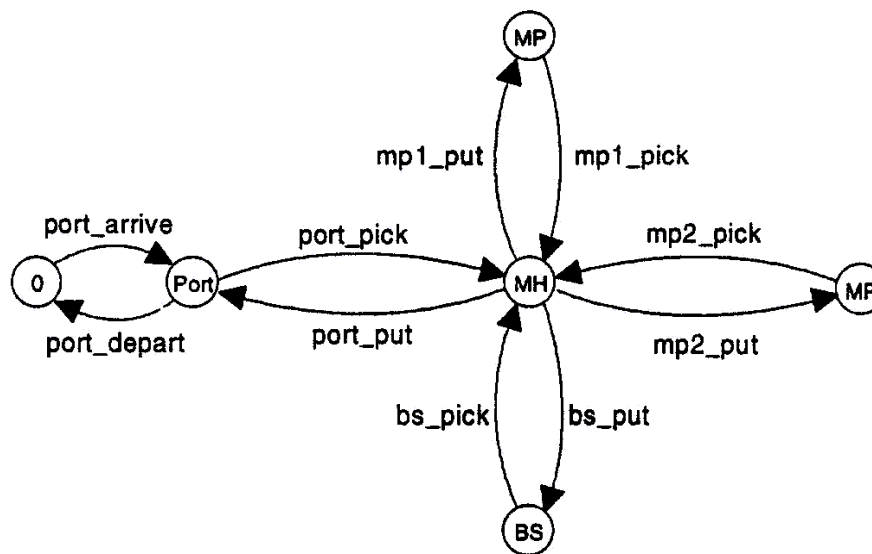
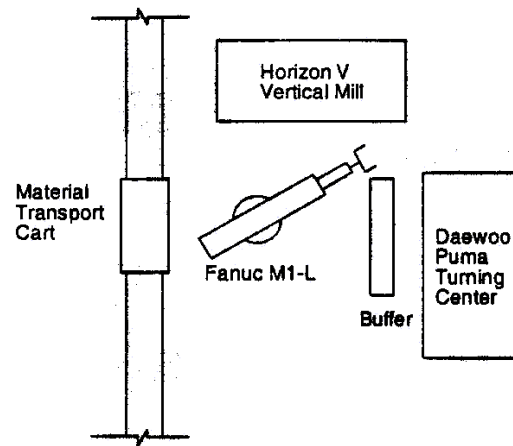


```

%% definitions
%class CONTROLLER_B
%object fadal
%output fadal
%states 16
%in 5
%out 5
%tasks 5
%final 15
%assign assign_bm
%%
%% in arc transition
0 assign_bm 1
3 grasp_bm 4
6 process_bm 7
9 process_bm 7
9 release_bm 10
12 clear_bm 13
%%
%% out arc transitions
2 assign_ok_mb 3
5 grasp_ok_mb 6
8 process_ok_mb 9
11 release_ok_mb 12
14 clear_ok_mb 15
%%
%% task arc transitions
1 assign 2
4 grasp 5
7 process 8
10 release 11
13 delete 14
%%
    
```

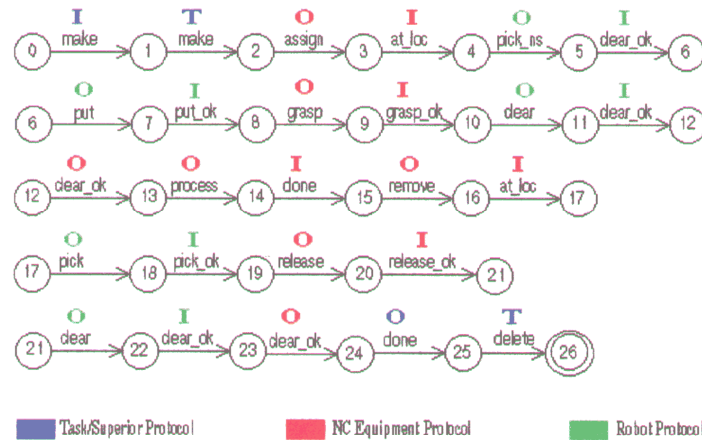
MPSG genérico para um MP

MODELOS FORMAIS DE EXECUÇÃO



*Estação
exemplo e o seu
modelo físico
correspondente*

MODELOS FORMAIS DE EXECUÇÃO



MPSG Controller Generator Input File

```

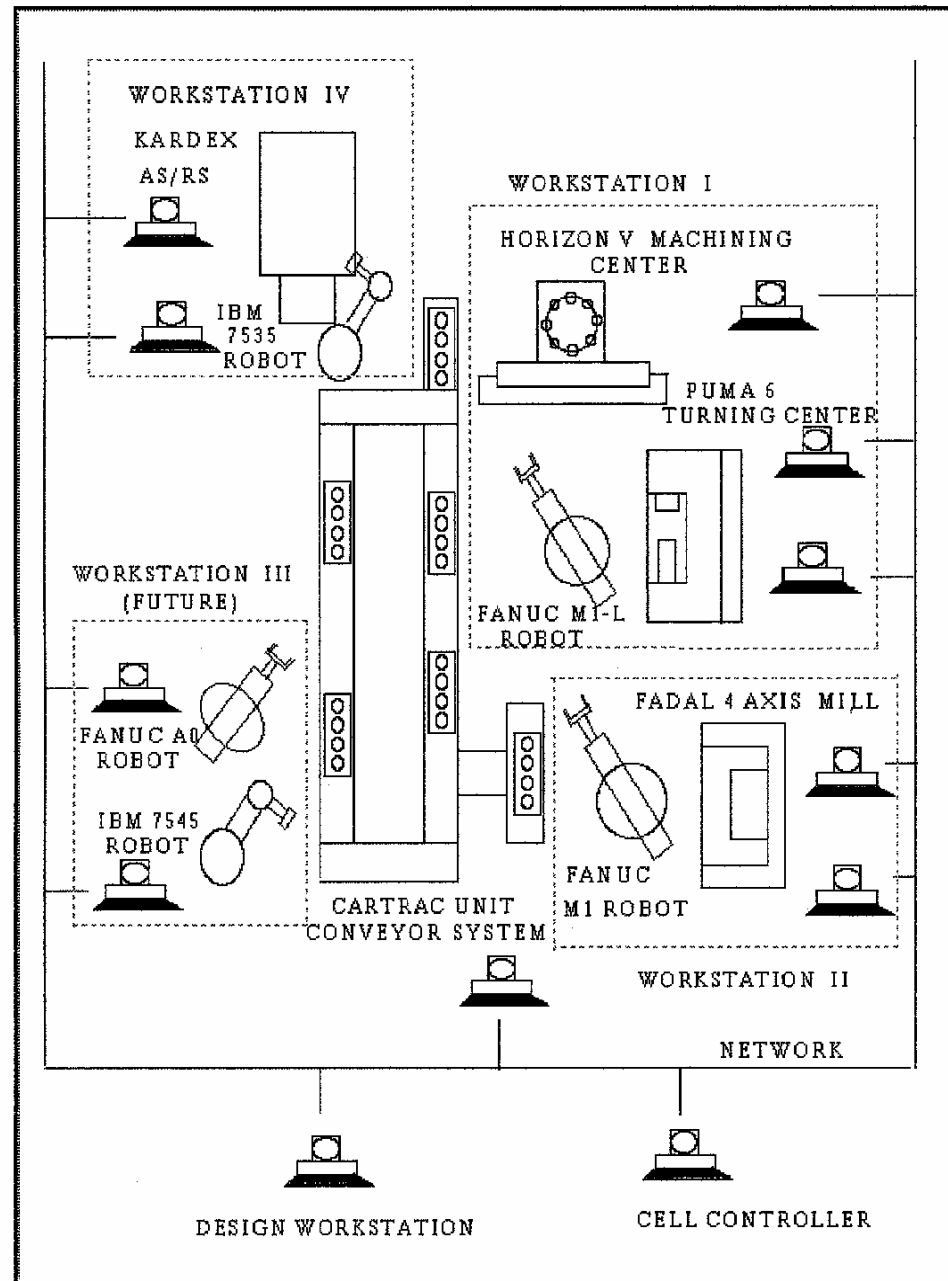
%% definitions
%class CONTROLLER_B
%object simple
%output simple
%states 27
%in 8
%out 11
%tasks 2
%final 26
%assign make
%%

%% in arc transitions
0 make 1
3 at_loc 4
5 clear_ok 6
7 put_ok 8
9 grasp_ok 10
11 clear_ok 12
14 done 15
16 at_loc 17
18 pick_ok 19
20 release_ok 21
22 clear_ok 23
%%

%% out arc transitions
2 assign 3
4 pick_ns 5
6 put 7
8 grasp 9
10 clear 11
12 clear_ok 13
13 process 14
15 remove 16
17 pick 18
    
```

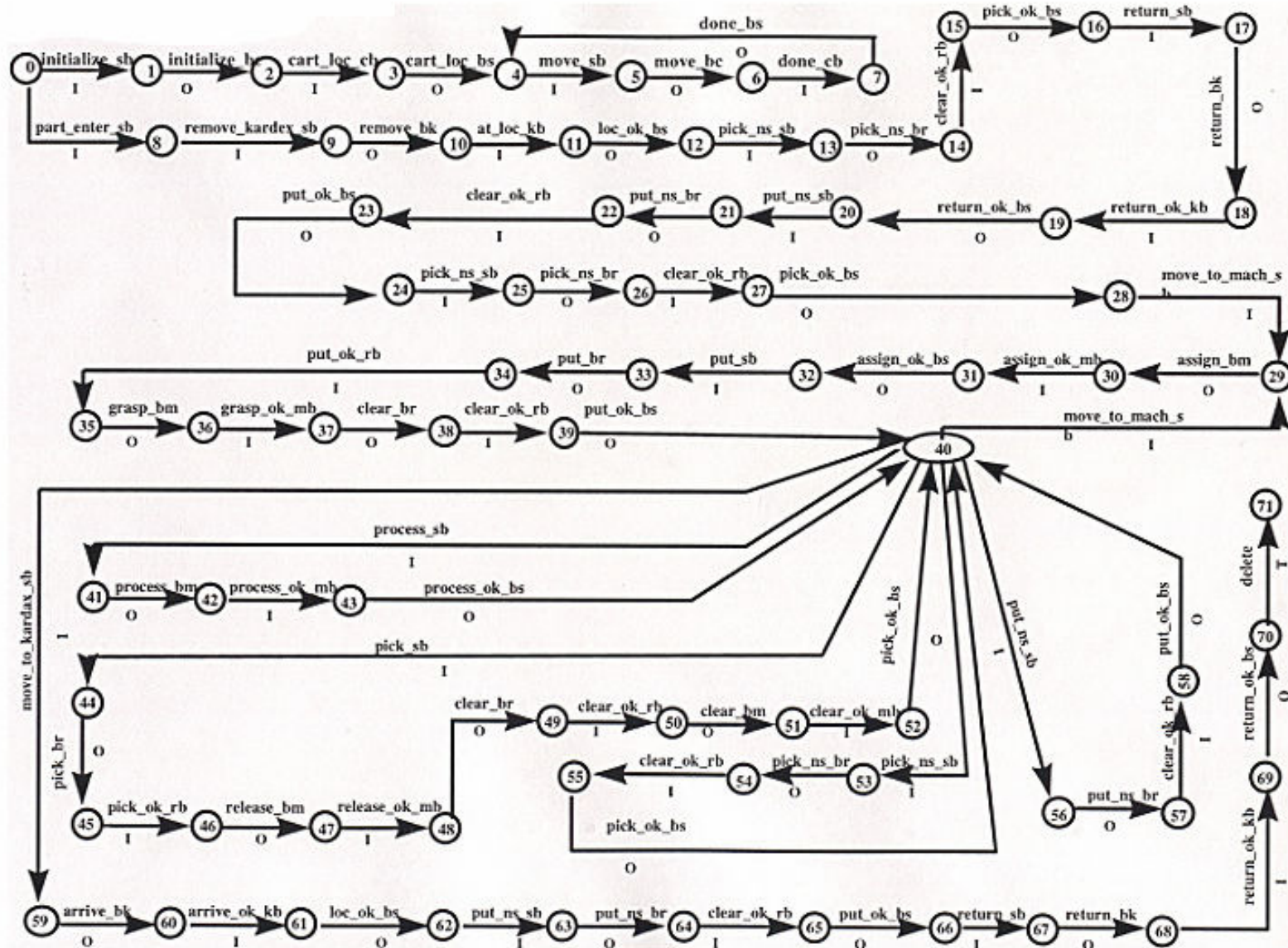
*MPSG para uma
 estação simples*

MODELOS FORMAIS DE EXECUÇÃO



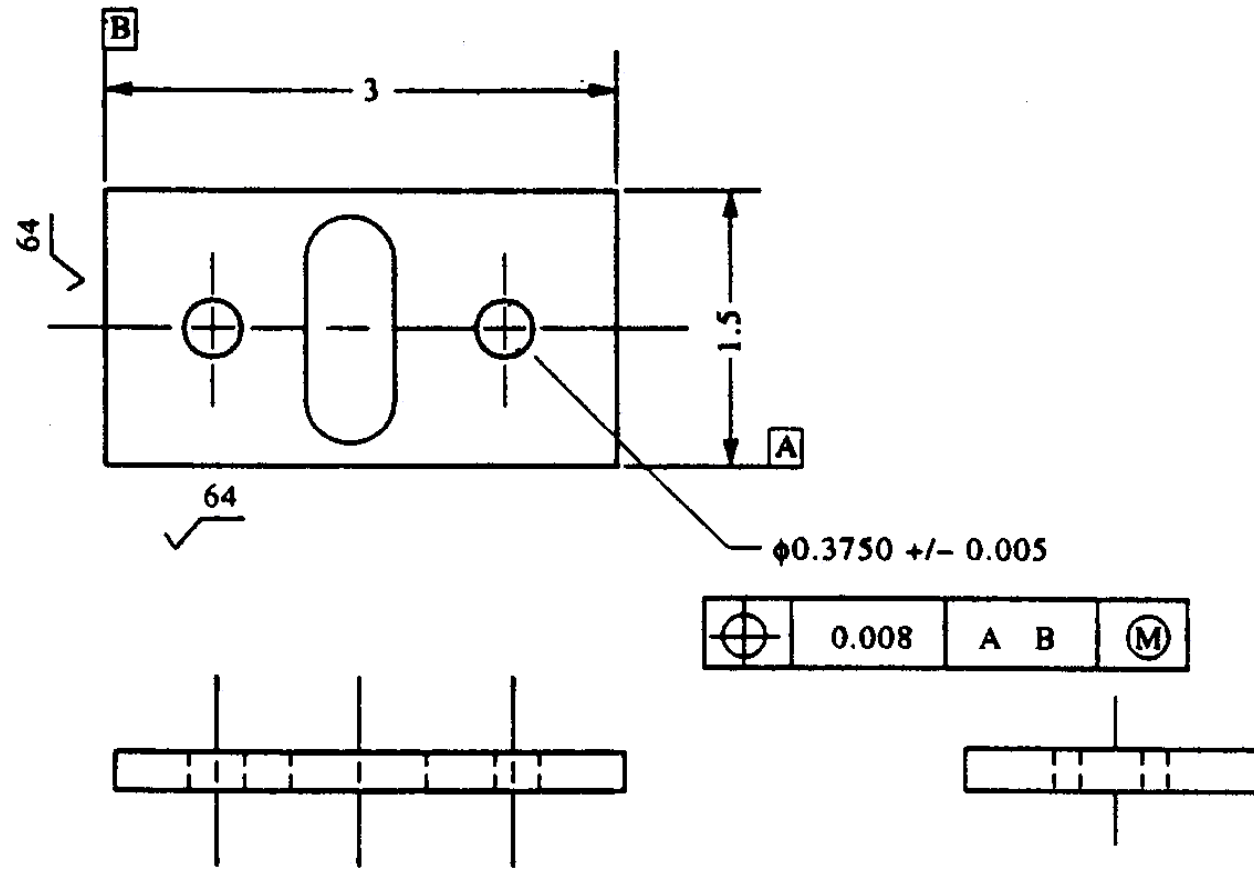
*Equipamentos no
CIMLab e o MPSG
para o Executor (1)*

MODELOS FORMAIS DE EXECUÇÃO



Equipamentos no CIMLab e o MPSG para o Executor (2)

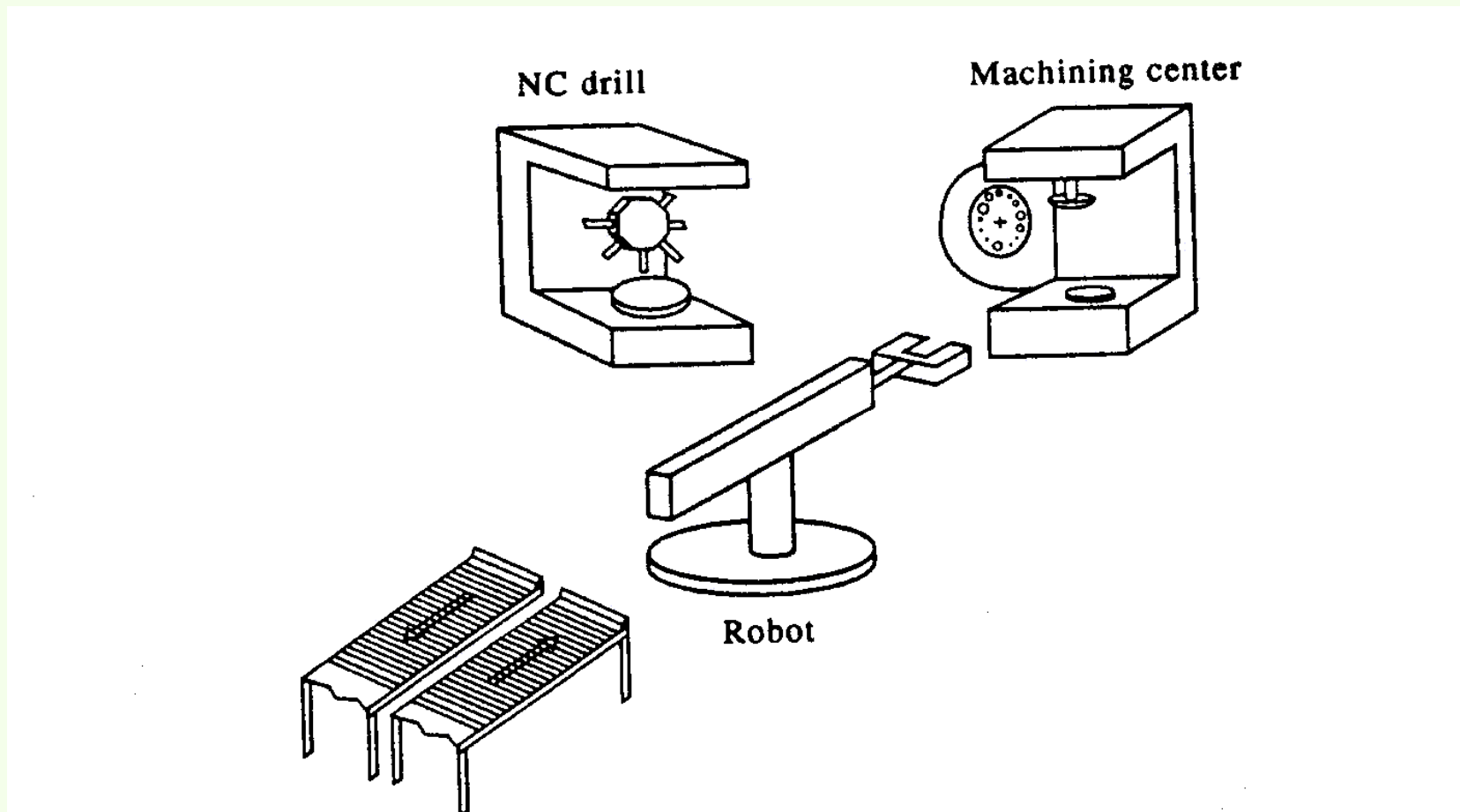
Plano de Processo Contendo Alternativas, e Seu Uso em Diferentes Níveis de Controle



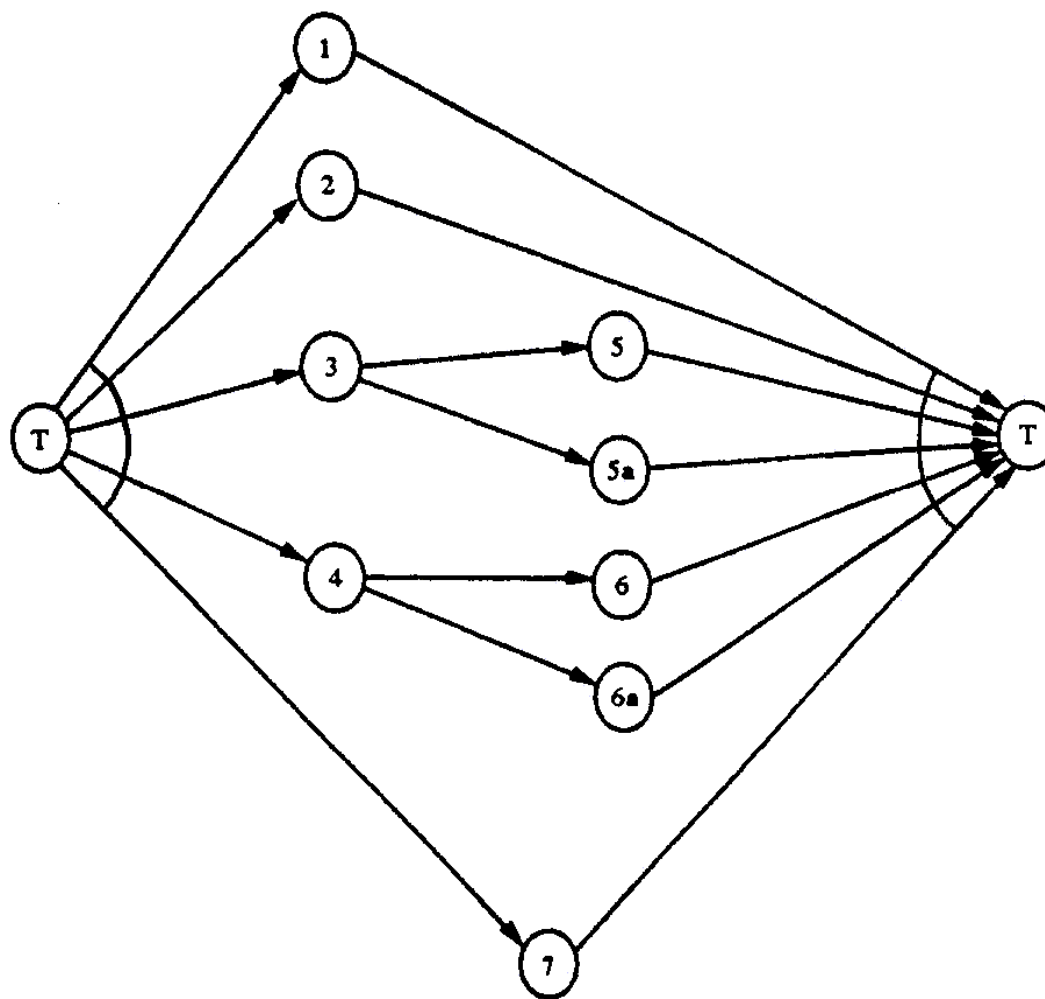
Uma peça exemplo

Oper.	Description	Feature	Feature Spec.	Time (Es.) (min)	Tooling	Machine	Fixture
1	Side Mill	Locat surf A	Mill Face A	0.6	1.0" endmill	Mach Cntr	Fxt F # 1 Fxt F # 2
2	Side Mill	Locat surf B	Mill Face B	0.35	1.0" end mill	Mach Cntr	Fxt F # 1 Fxt F # 2
3	Twist drill	Hole # 1	Rough dri H # 1	0.33	0.3595" drill	Mach Centr Drill	Fxt F # 2 Fxt F # 3
5	Ream	Hole # 1	Finish H # 1	0.25	0.375" ream	Mach Cntr Drill	Fxt F # 2 Fxt F # 3
5a	Bore	Hole # 1	Finish H # 1	0.34	0.375" bore	Mach Cntr Drill	Fxt F # 2 Fxt F # 3
4	Twist drill	Hole # 2	Rough dri H # 2	0.33	0.3595" drill	Mach Cntr Drill	Fxt F # 2 Fxt F # 3
6	Ream	Hole # 2	Finish H # 2	0.25	0.375" ream	Mach Cntr Drill	Fxt F # 2 Fxt F # 3
6a	Bore	Hole # 2	Finish H # 2	0.34	0.375" bore	Mach Cntr Drill	Fxt F # 2 Fxt F # 3
7	Slotmill	Slot # 1	Mill Slot # 1	0.75	0.5" endmill	Mach Cntr	Fxt F # 1 Fxt F # 2

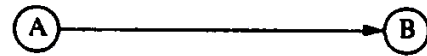
Tarefas (operações) e resumo dos recursos para a peça ilustrada na figura 1, ordenadas por feature



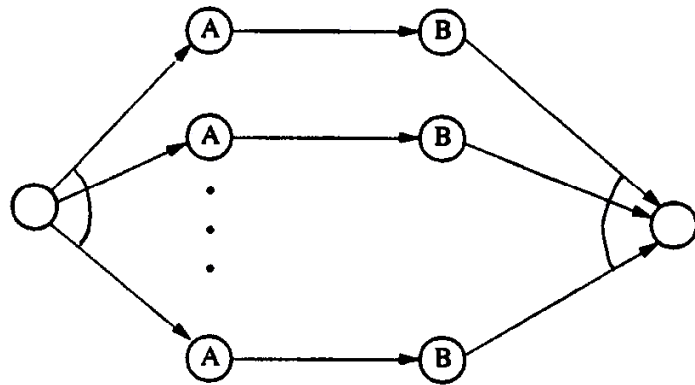
O sistema de manufatura usado para ilustrar o controle



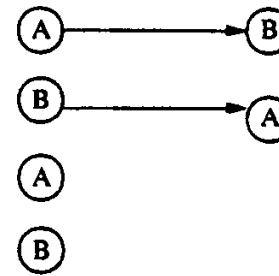
O grafo e/ou para a peça exemplo



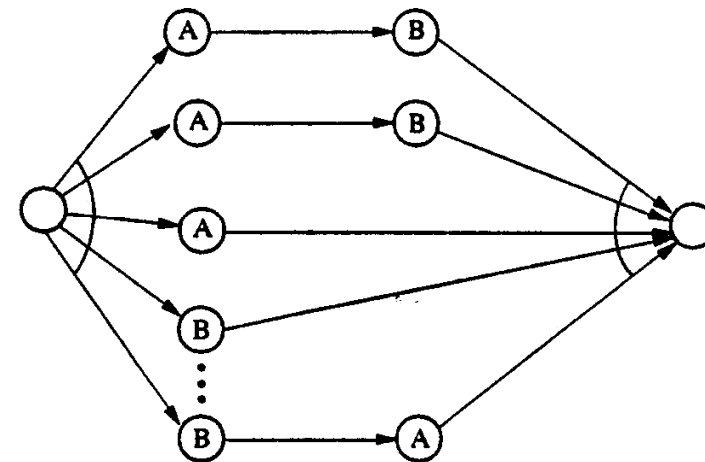
(a) Single part task graph



(b) Multiple part task graph



(a) Single part task graph

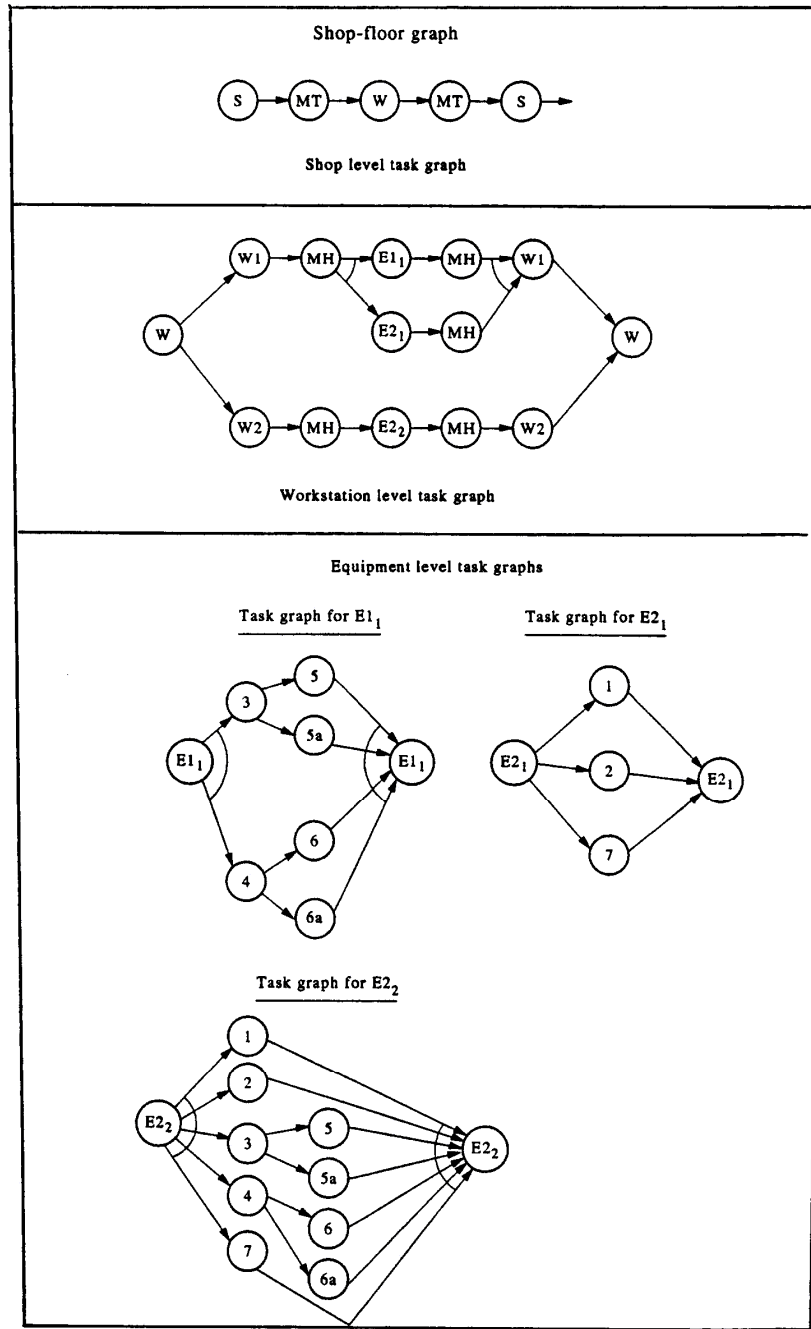


(b) Multiple part task graph

Grafo de tarefas nos modelos de Johnson e Jackson



*Grafo de tarefas no modelo de Jackson com **MH***



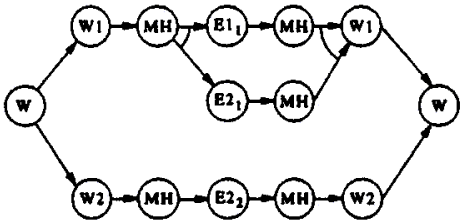
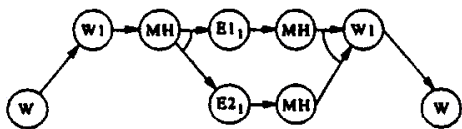
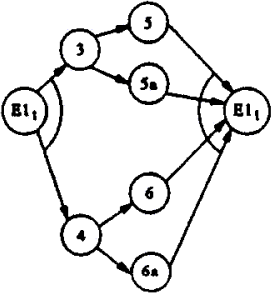
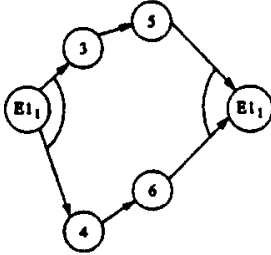
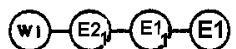



*Grafos e/ou para a
peça e para o
sistema*

Resource	Tasks/operations	Additional resources
E1 ₁ -Drill	3, 4, 5, 5a, 6, 6a	Standard fixture (F2)
E2 ₁ -MachCntr	1, 2, 7	Standard fixture (F1)
E2 ₂ -MachCntr	1, 2, 3, 4, 5, 5a, 6, 6a, 7	Magnetic fixture (F3)

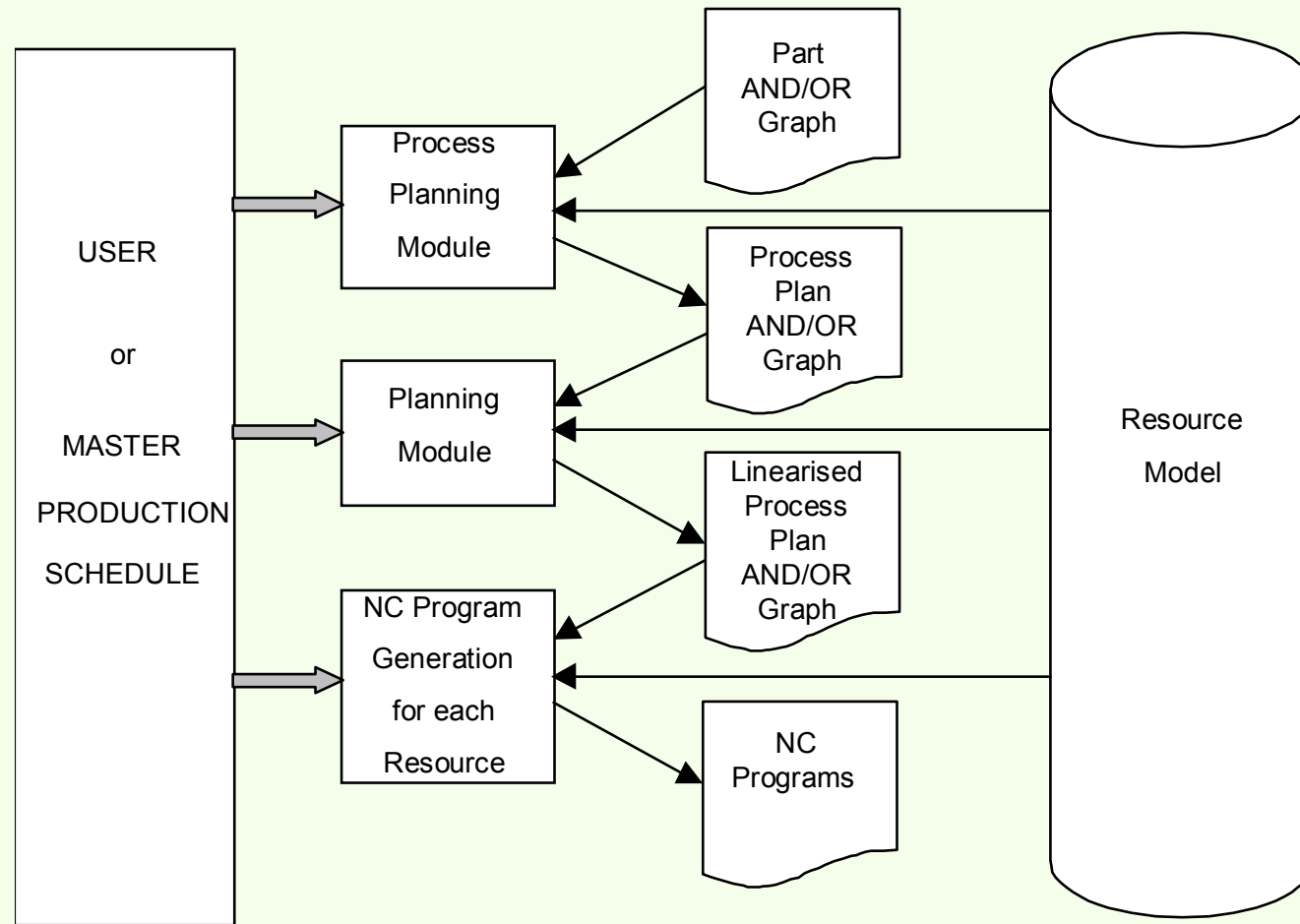
$W1 = \{E1, E2_1\}; W2 = \{E2_2\}.$

Uma lista de tarefas/operações derivadas do resumo do roteamento de operações

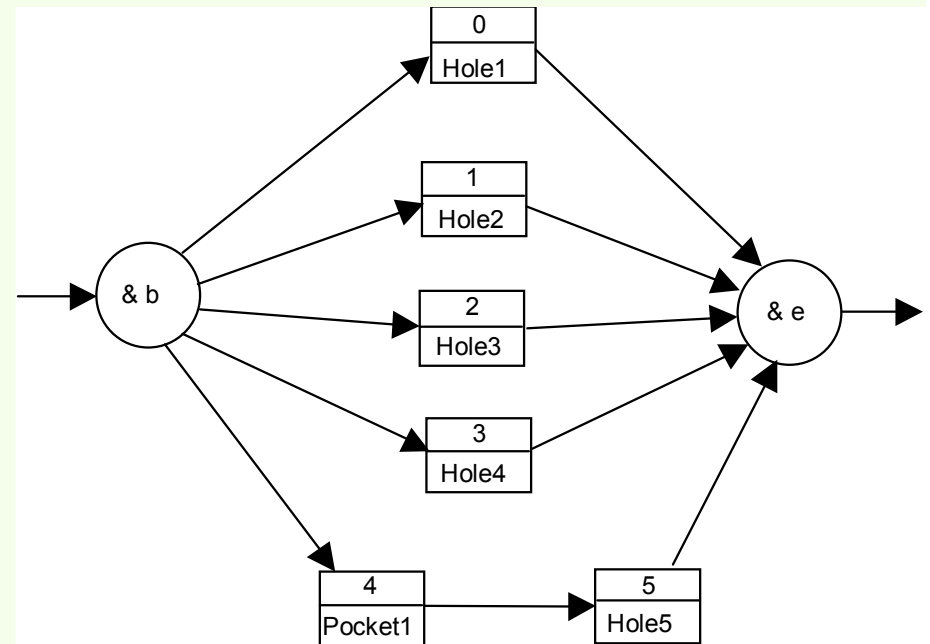
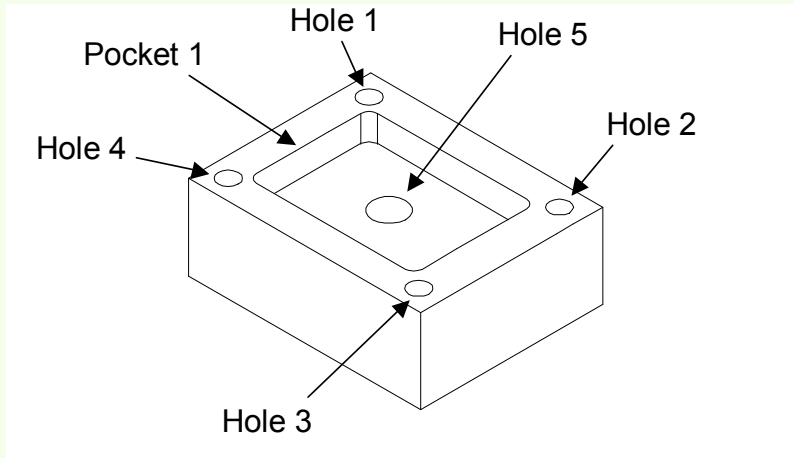
Activity	Shop	Workstation	Equipment
Planning	 <p>No alternatives appear in the graph. The graph need not be altered</p> <p>DeOR</p> 	 <p>Depending on the criteria and procedure(s), alternatives are removed from the graph.</p> 	<p>For E1</p>  <p>Depending on the criteria and procedure(s), alternatives are selected.</p> 
Scheduling	<p>No AND nodes here. Planning produced a serial graph.</p>	<p>Depending on the criteria and scheduling procedures, tasks must be sequenced (serialized).</p> 	<p>Depending on the criteria and scheduling procedures, tasks are sequenced at the machine.</p> 

Detalhes do controle para múltiplas peças no exemplo considerado

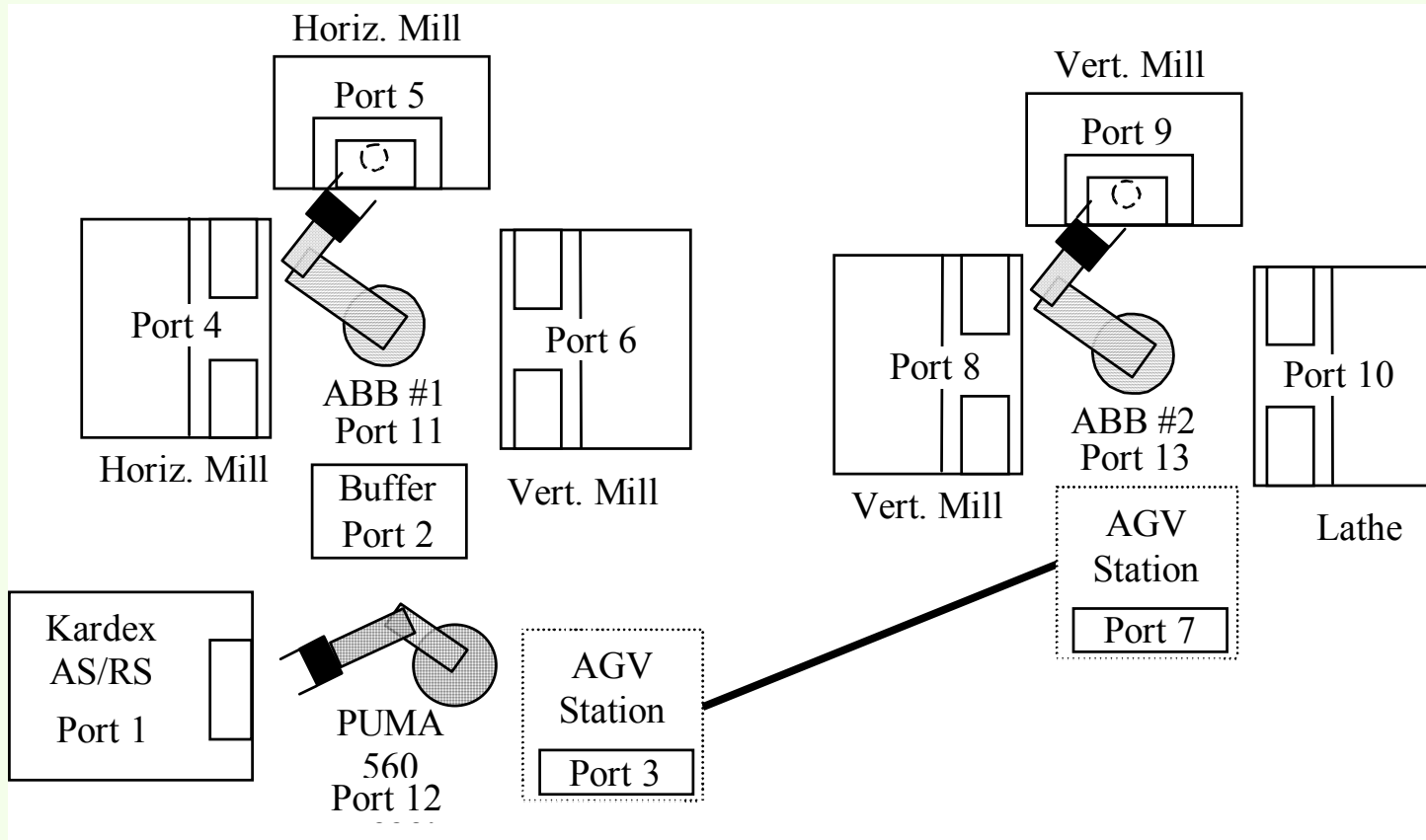
Plano de Processos com Alternativas para a Fabricação de um Lote num FMS



Grafo E/OU para uma Peça



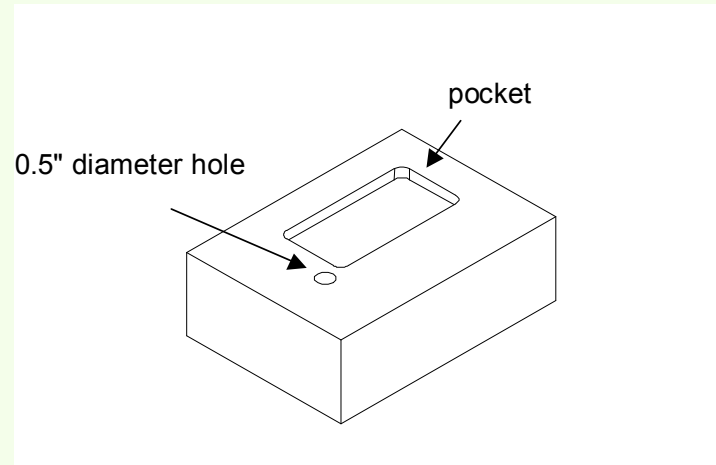
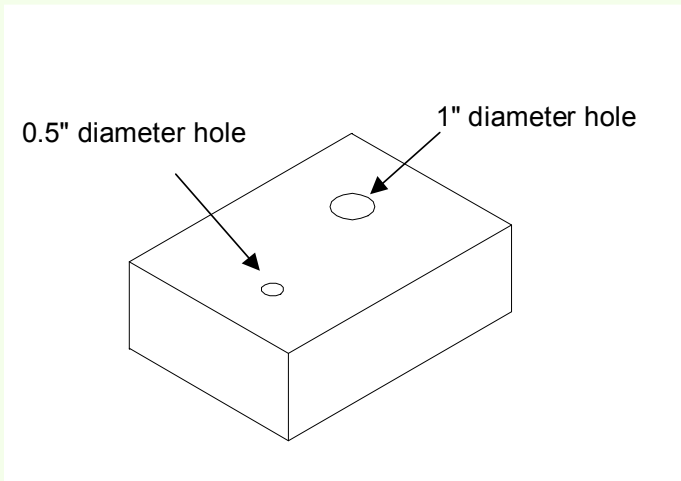
Sistema de Manufatura Considerado



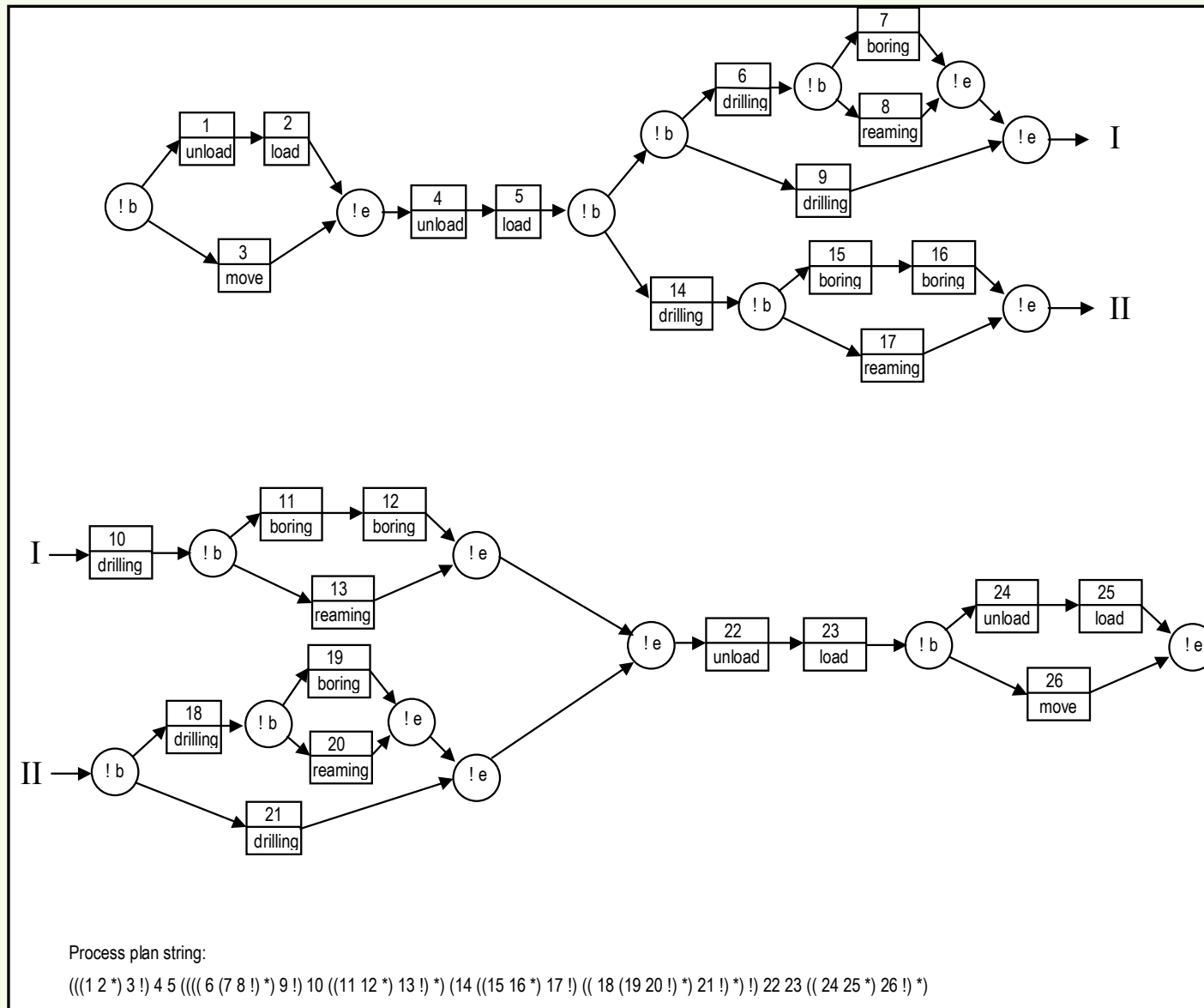
Estratégia para a Geração do Programa NC

```
G00 X <x> Y <y> H0 M6 T <tool>  
H <tool offset> Z <z + 0.10>  
Z <z + 0.10>  
G01 Z <z - depth> S <rpm> F <feed> M3 M8  
G00 Z <z + 0.10>  
Z <z + 0.10> M5
```


Peças a Serem Fabricadas



Plano de Processo Gerado



Plano de Processo Gerado

Process plan string:

```
(( (1 2 *) 3 ) 4 5 ((( (6 (7 8 ) *) 9 ) 10 (( (11 12 *) 13 ) *) (14 (( (15 16 *) 17 ) (( (18 (19 20 ) *) 21 ) *) ) 22 23 (( (24 25 *) 26 ) *)
```

Identificação dos Nós

Node ID	Resource	Operation	From Port → To Port /Coordinates (x,y) and diameter (∅)	Node ID	Resource	Operation	From Port → To Port /Coordinates (x,y) and diameter (∅)
1	PUMA robot	unload	1 → 12	14	Vertical M.C.	drilling	(1.5, 2.0); 0.4375"
2	PUMA robot	load	12 → 2	15	Vertical M.C.	boring	(1.5, 2.0); 0.48"
3	Human	move	1 → 2	16	Vertical M.C.	boring	(1.5, 2.0); 0.5"
4	ABB#1 robot	unload	2 → 11	17	Vertical M.C.	reaming	(1.5, 2.0); 0.5"
5	ABB#1 robot	load	11 → 6	18	Vertical M.C.	drilling	(5.0, 3.0); 0.9375"
6	Vertical M.C.	drilling	(5.0, 3.0); 0.9375"	19	Vertical M.C.	boring	(5.0, 3.0); 1.0"
7	Vertical M.C.	boring	(5.0, 3.0); 1.0"	20	Vertical M.C.	reaming	(5.0, 3.0); 1.0"
8	Vertical M.C.	reaming	(5.0, 3.0); 1.0"	21	Vertical M.C.	drilling	(5.0, 3.0); 1.0"
9	Vertical M.C.	drilling	(5.0, 3.0); 1.0"	22	ABB#1 robot	unload	6 → 11
10	Vertical M.C.	drilling	(1.5, 2.0); 0.4375"	23	ABB#1 robot	load	11 → 2
11	Vertical M.C.	boring	(1.5, 2.0); 0.48"	24	PUMA robot	unload	2 → 12
12	Vertical M.C.	boring	(1.5, 2.0); 0.5"	25	PUMA robot	load	12 → 1
13	Vertical M.C.	reaming	(1.5, 2.0); 0.5"	26	Human	move	2 → 1

Programas NC Gerados

Hole \varnothing 0.5"	Pocket
%	%
N001 M64	N001 M64
N002 G80 G40 G17	N002 G80 G40 G17
N003 E2 G90 X0 Y0	N003 E2 G90 X0 Y0
N004 G00 X1.5000 Y2.0000 H0 M6 T1	N004 G00 X1.5000 Y2.0000 H0 M6 T1
N005 H1 Z1.1000	N005 H1 Z2.1000
N006 Z1.1000	N006 Z2.1000
N007 G01 Z0.5000 S829 F9.9531 M3 M8	N007 G01 Z1.5000 S829 F9.9531 M3 M8
N008 G00 Z1.1000	N008 G00 Z2.1000
N009 Z1.1000 M5	N009 Z2.1000 M5
N010 G00 X1.5000 Y2.0000 H0 M6 T2	N010 G00 X1.5000 Y2.0000 H0 M6 T2
N011 H2 Z1.1000	N011 H2 Z2.1000
N012 Z1.1000	N012 Z2.1000
N013 G01 Z0.5000 S458 F6.8755 M3 M8	N013 G01 Z1.5000 S458 F6.8755 M3 M8
N014 G00 Z1.1000	N014 G00 Z2.1000
N015 Z1.1000 M5	N015 Z2.1000 M5
N016 G00 X5.0000 Y3.0000 H0 M6 T3	N016 G00 X4.0000 Y3.1000 H0 M6 T4
N017 H3 Z1.1000	N017 H4 Z2.1000
N018 Z1.1000	N018 Z2.1000
N019 G01 Z0.5000 S363 F7.9832 M3 M8	N019 G01 Z1.7500 S2723 F16.3408 M3 M8
N020 G00 Z1.1000	N020 X5.5000 Y3.7000
N021 Z1.1000 M5	N021 X2.5000 Y3.7000
N022 H0	N022 X2.5000 Y2.5000
N023 G00 X0.0000	N023 X5.5000 Y2.5000
N024 G92 X0	N024 X5.5000 Y3.7000
N025 E0 X0 Y0	N025 G00 X4.0000 Y3.1000 Z1.7500
N026 M65	N026 G01 Z1.5000 S2723 F16.3408 M3 M8
N027 M2	N027 X5.5000 Y3.7000
%	N028 X2.5000 Y3.7000
	N029 X2.5000 Y2.5000
	N030 X5.5000 Y2.5000
	N031 X5.5000 Y3.7000
	N032 G00 X4.0000 Y3.1000 Z1.5000
	N033 G00 X4.0000 Y3.1000 Z2.1000 M5
	N034 H0
	N035 G00 X0.0000
	N036 G92 X0
	N037 E0 X0 Y0
	N038 M65
	N039 M2
	%

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

- Quando deseja-se uma elevada disponibilidade organizacional de sistemas de manufatura e montagem controlados por computador, é essencial que dados e materiais (**ferramentas, dispositivos de teste, dispositivos de fixação**) sejam identificados **sem ambigüidade** (isto é, sejam únicos), e sejam preparados no tempo certo.
- Acessórios de manufatura → peças, ferramentas, dispositivos de teste, dispositivos de fixação (ver figura).

Banco de dados:

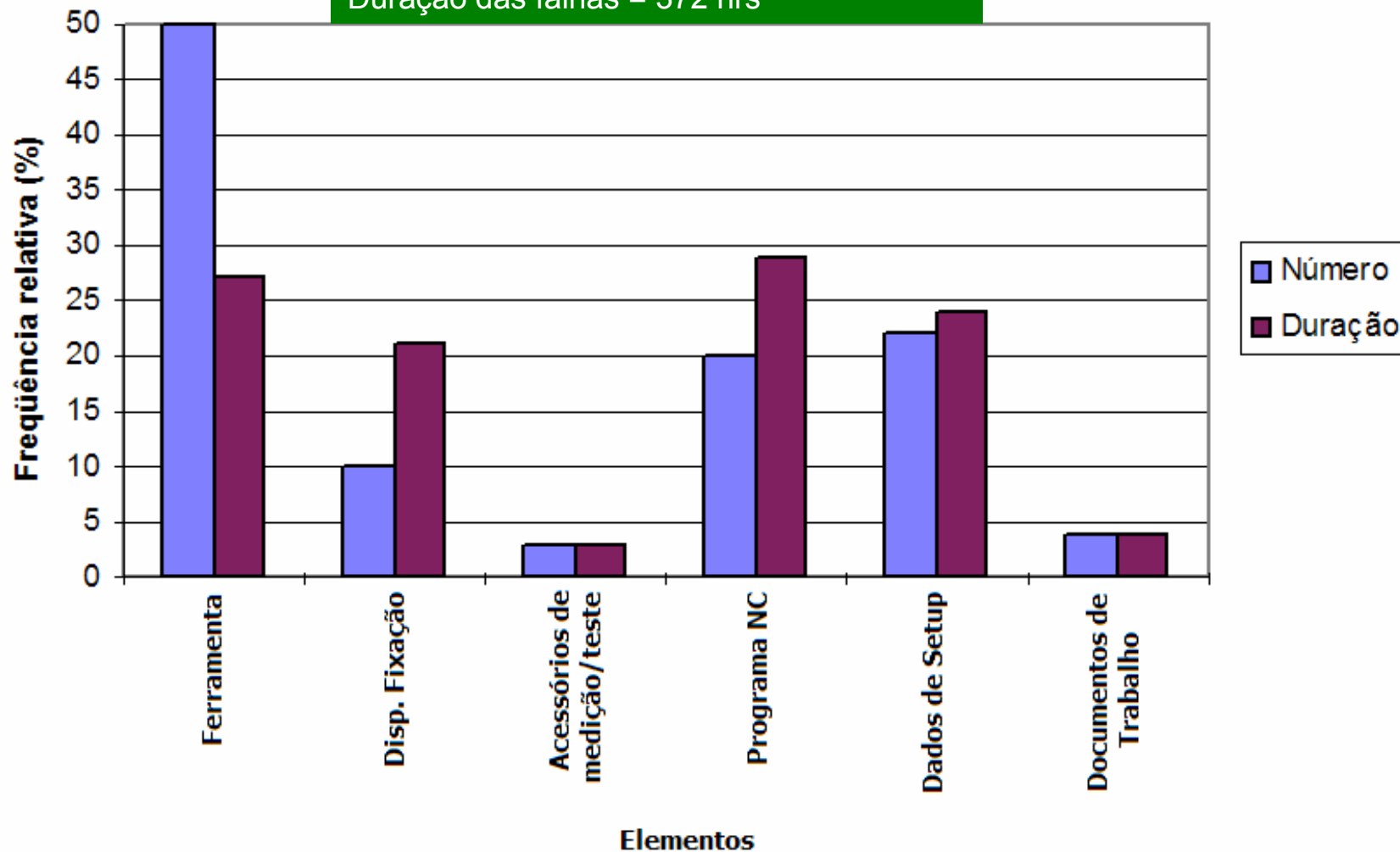
Número de registros de dados = 442

Período estudado = 24 dias (2 turnos cada)

Números de máquinas NC = 9

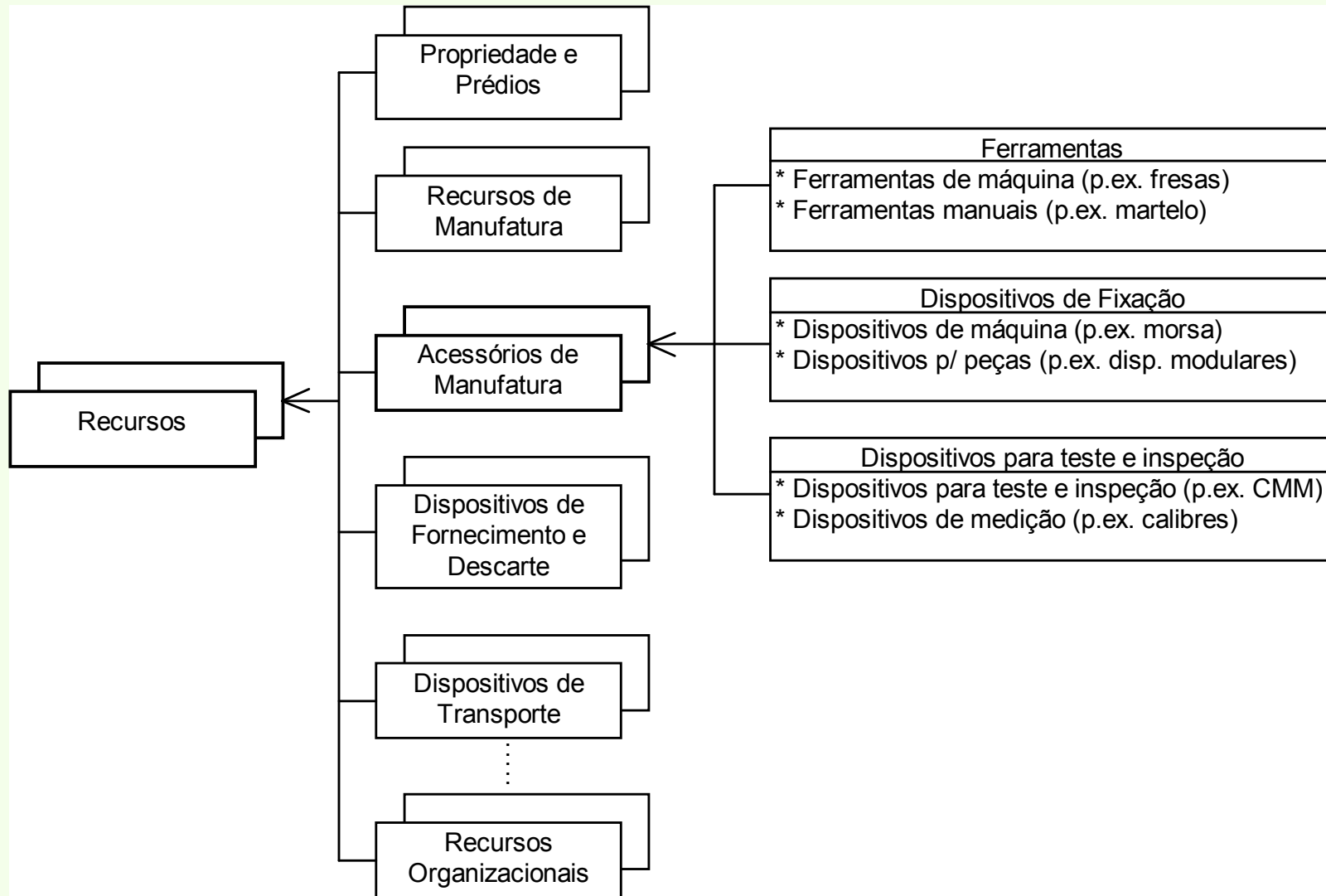
Número de falhas = 442

Duração das falhas = 372 hrs



Número e visualização de falhas no fornecimento de acessórios

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

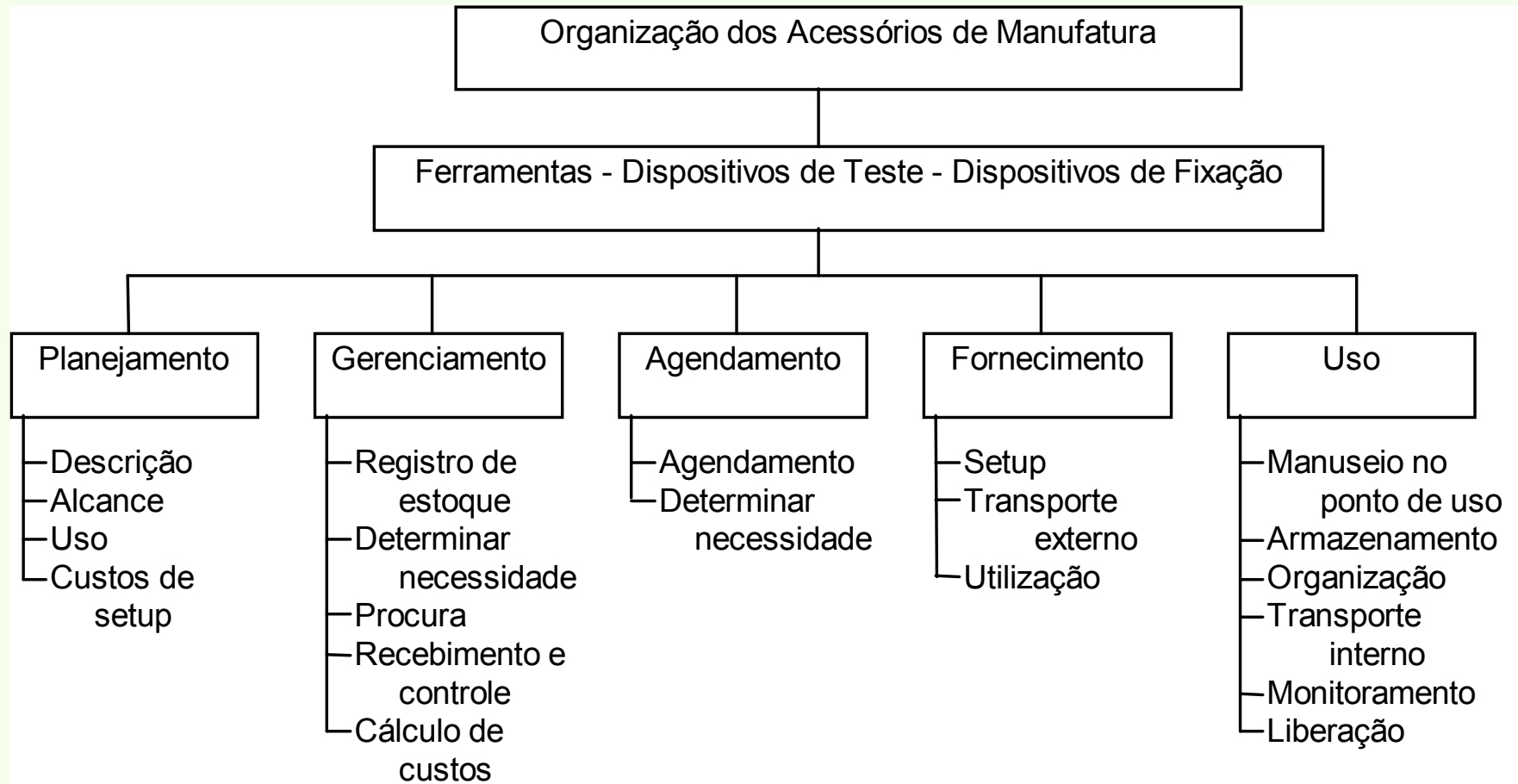


Estrutura dos acessórios de manufatura como parte dos recursos

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

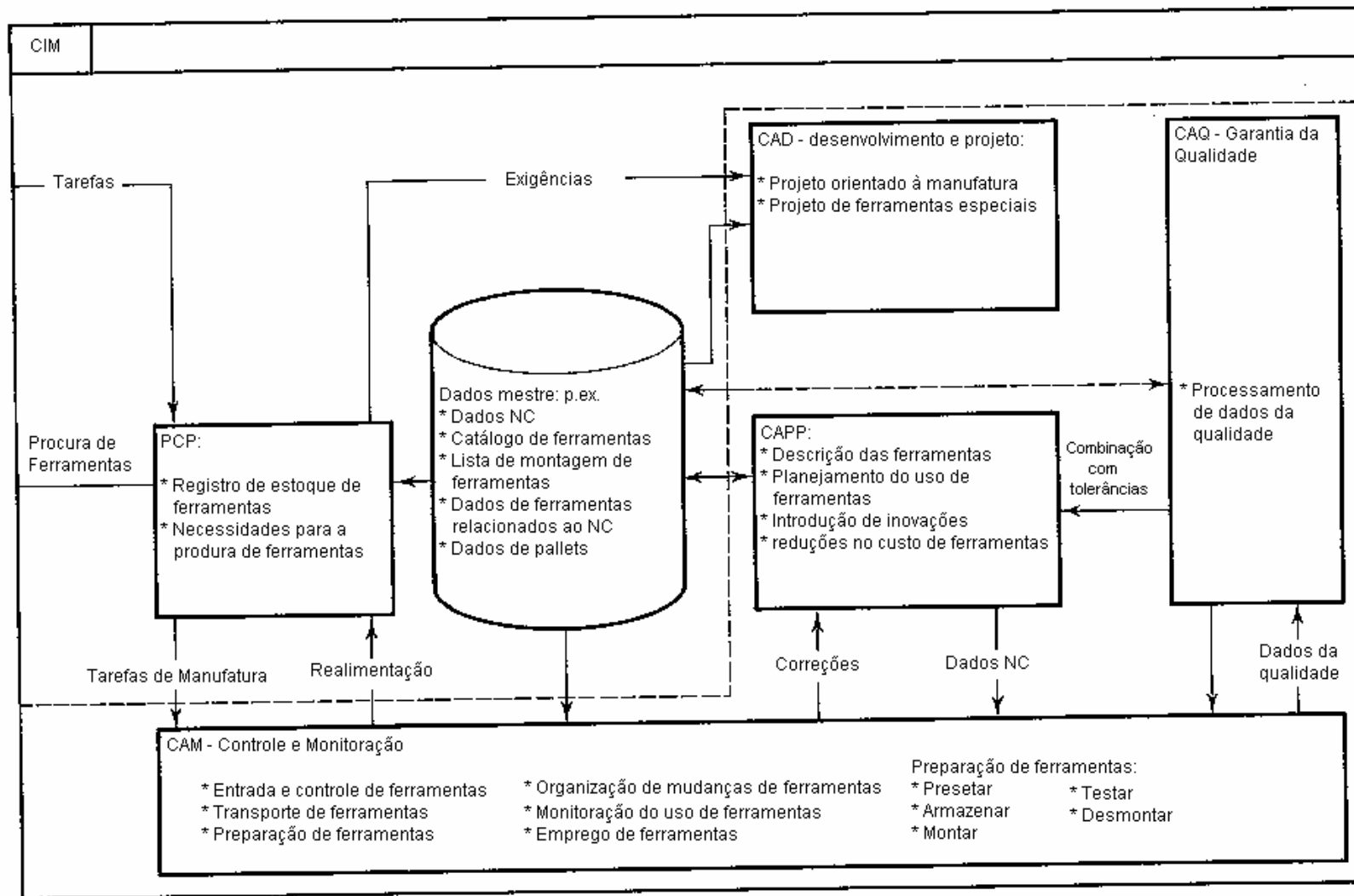
- Organização de acessórios na empresa pode ser subdividida em **5 áreas funcionais** (ver figura):
 - planejamento,
 - gerenciamento e controle,
 - agendamento,
 - fornecimento,
 - uso.
- Esta classificação refere-se também ao tempo: enquanto tarefas de planejamento são executadas durante períodos longos, o uso é ligado diretamente ao processo de manufatura.

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR



Acessórios de manufatura

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR



A tarefa do gerenciamento de ferramentas dentro do conceito de CIM

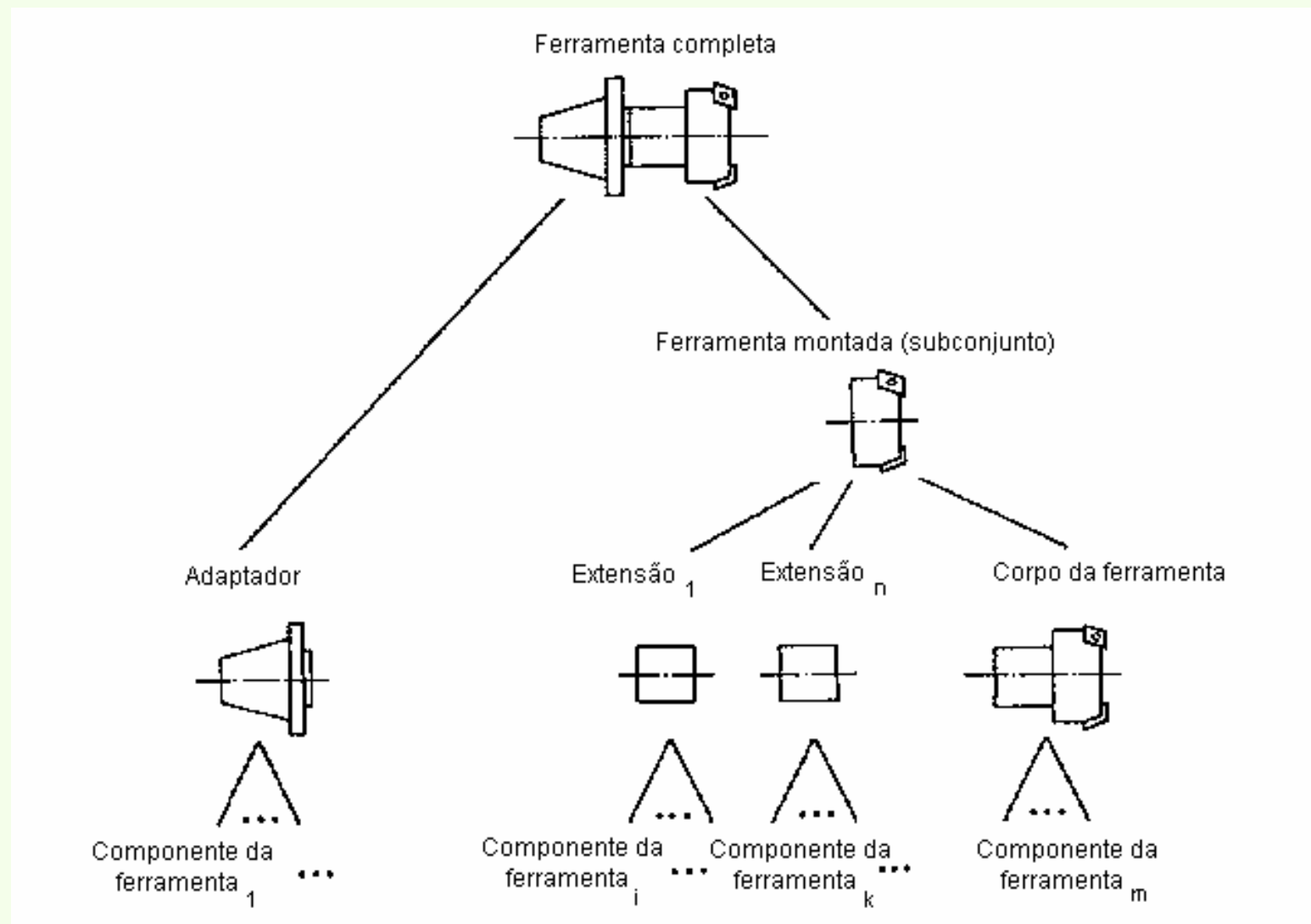
ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

- Fator importante para uma organização eficiente das ferramentas:
 - normalmente **códigos simples de identificação** não são suficientes, devido às condições que limitam a tarefa de usinagem.
 - áreas operacionais individuais tais como o planejamento de tarefas requerem **critérios de classificação ou busca** para encontrar as ferramentas.
- Dados de ferramentas em geral não são classificados de acordo com características e condições temporais tais como o **período de validade**.
- Ferramentas são distinguidas pelas seguintes características: *geométricas; tecnológicas; organizacionais; estatísticas.*

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

- Características geométricas e tecnológicas podem determinar a classificação de uma ferramenta.
- Classificação temporal corresponde às seguintes categorias: **constante** (dados mestre); **constante durante um período** (dados temporários); **variável** (dados de estado).
- Maioria dos conjuntos de dados de ferramentas consistem de dados mestre.
- Uma classificação desde **ferramentas completas** (CT) até **componentes individuais de ferramentas** (ITC) pode ser considerado vantajoso (ver figura).

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR



Componentes e subconjuntos de ferramentas

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

<i>Nome do Campo</i>	<i>Formato</i>	<i>Descrição</i>
<i>Dados de Identificação</i>		
TTN	N,8	Número do tipo de ferramenta (chave primária para os dados mestre)
TTI	N,3	Identificação do tipo de ferramenta
DPTN	A,20	Número do tipo de ferramenta
CUTEDG	A,20	Número do suporte
<i>Dados Organizacionais</i>		
TTYID	A,20	Identificação da ferramenta em formato texto
MAGTY	N,2	Identificação de diferentes tipos de magazines para ferramentas pequenas/grandes, cabeçotes de mandril, etc...
WEIGH	I	Identificação do peso para os dispositivos de manuseio (em N)
<i>Dados Tecnológicos</i>		
CMO	A,6	Código do material da ferramenta
*CHIP	R	Espessura de cavaco máxima possível
*MINC	R	Profundidade de corte máxima
*MINSPANR	R	Espessura de cavaco mínima
...		

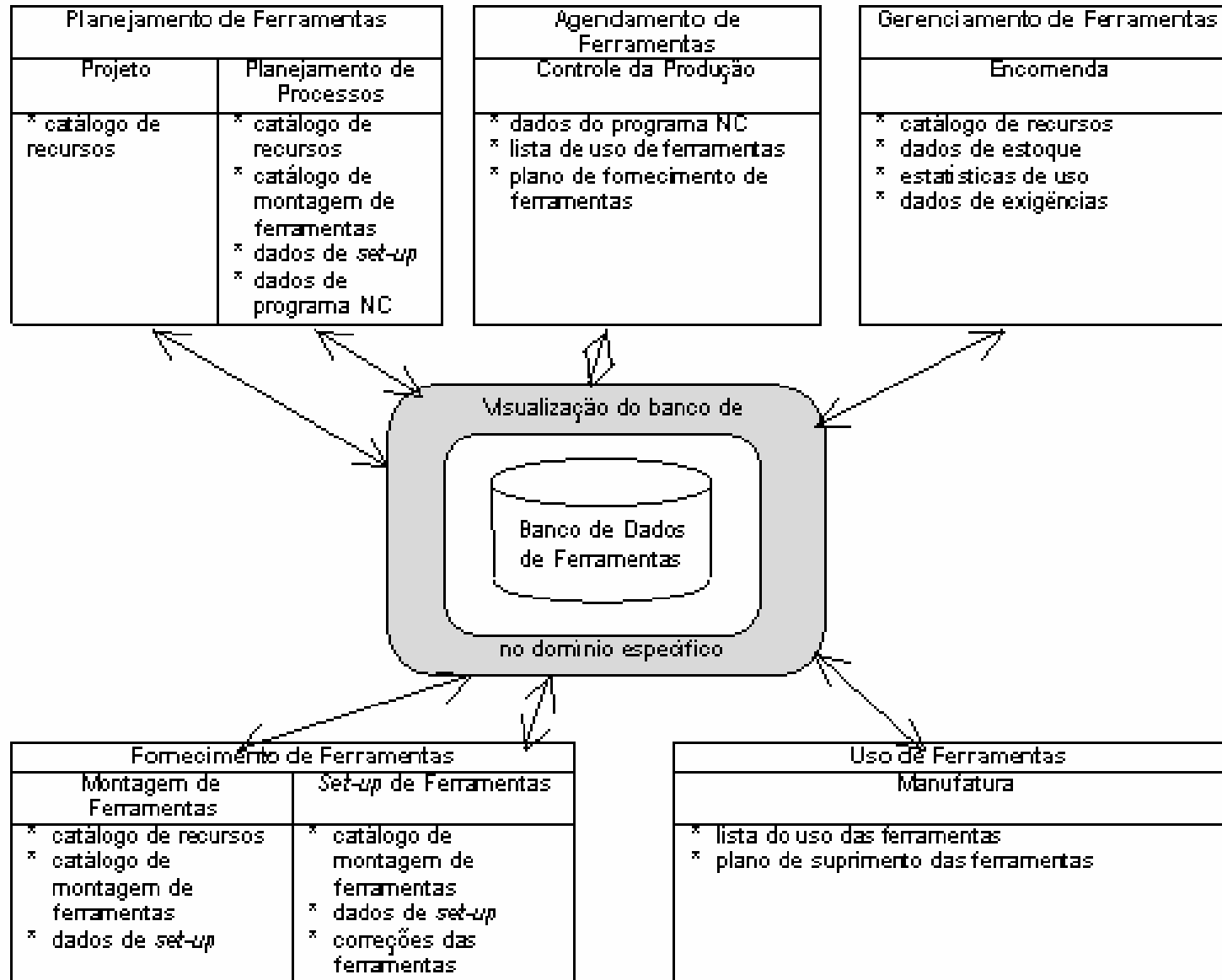
Dados mestre contendo informações sobre ferramentas - Parte 1

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

Nome do Campo	Formato	Descrição
Dados Geométricos		
TAD	R	Ajuste da dimensão L entre a quina da ferramenta e o ponto de referência de ajuste
PTAD	R	Tolerância positiva
NTAD	R	Tolerância negativa
QADJ	R	Dimensão de ajuste Q
QPTOL	R	Tolerância positiva
...		
Desses dados, são necessárias combinações referentes às tarefas. Seguem dois exemplos abaixo:		
Plano de fornecimento de ferramentas		
TTN	N,8	Número do tipo de ferramenta
JNO	A,20	Número da tarefa
MCNO	I	Número da máquina/célula
DOU	D	Data de uso
Dados específicos do programa NC		
NCPNO	A,12	Número do programa NC
TNO	N,8	Número do tipo de ferramenta
AJOB	A,32	Número da tarefa de montagem
OPLIF	I	Vida necessária de operação, padronizada em minutos/10
BRM	R	Multiplicador referente ao risco de quebra no caso de usos especiais
...		

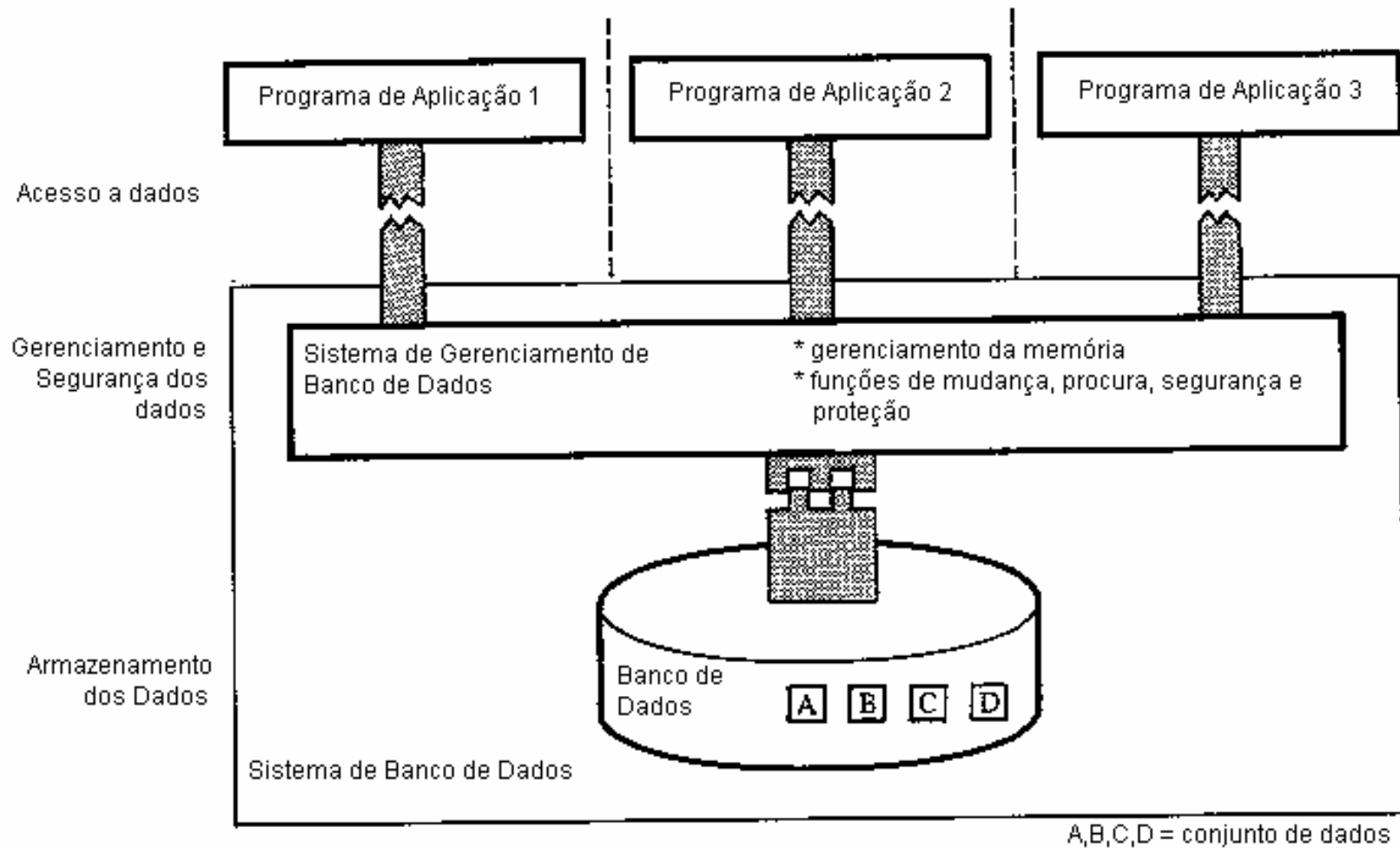
Dados mestre contendo informações sobre ferramentas - Parte 2

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR



Um banco de dados de ferramentas (mostra-se as 5 áreas funcionais descritas anteriormente)

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

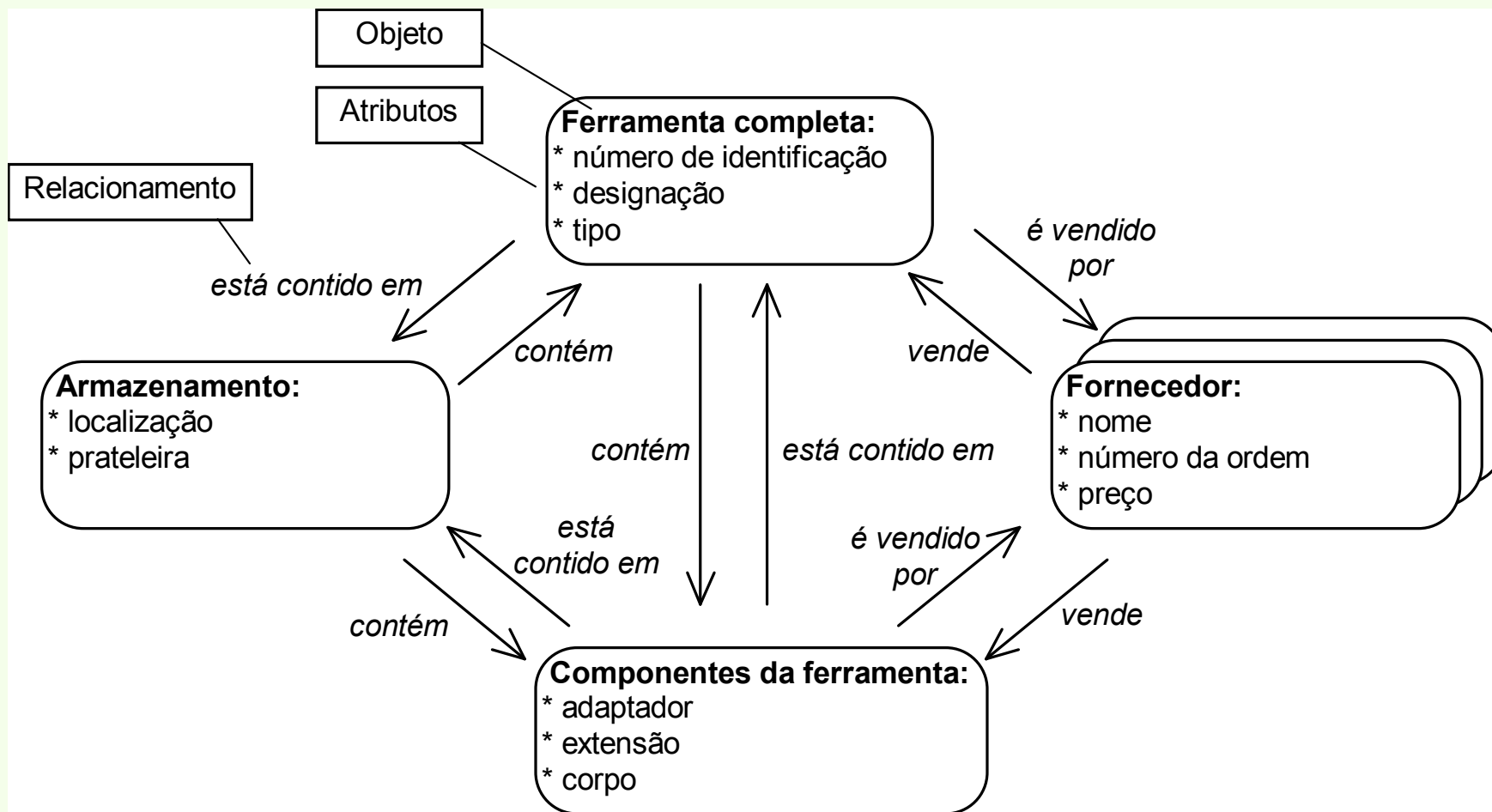


Um banco de dados integrado

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

- Manutenção de um sistema de organização de ferramentas ?
- Antes que um grande banco de dados de ferramentas for planejado e projetado, uma análise e uma estruturação pormenorizada dos dados deve ser feita.
- Passos mais importantes:
 - análise dos dados existentes sobre as ferramentas;
 - análise e descrição dos objetos relevantes e seus atributos;
 - análise e descrição das relações relevantes entre os objetos.
- Um **objeto** é uma entidade real que pode ser descrita por atributos e que possui uma significação específica (ver figura).

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR



Um modelo de dados de ferramentas

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

The diagram shows a table with three columns and four rows. The columns are labeled 'Número de Identificação', 'Designação', and 'Tipo completo de ferramenta'. The rows contain data for different tool types: 'Broca de centro', 'Broca helicoidal', 'Fresa de face', and an ellipsis. Labels with leader lines point to various parts of the table: 'Nome' points to the title 'Ferramenta Completa'; 'Tabela' points to the table structure; 'Campo' points to a cell; 'Linha (registro de dados)' points to a row; 'Atributo' points to a column; and 'Coluna' points to the table's column structure.

Ferramenta Completa		
Número de Identificação	Designação	Tipo completo de ferramenta
1100.2870	Broca de centro	BC
1100.2871	Broca helicoidal	BH
1100.2872	Fresa de face	FF
⋮	⋮	⋮

Tabelas de banco de dados para uma ferramenta completa

ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

<i>Ferramenta completa</i>		
Nº de identificação	Designação	Tipo Completo de Ferramenta
1100.2870	Broca de centro	BC
1100.2871	Broca helicoidal	BH
1100.2872	Fresa de face	FF

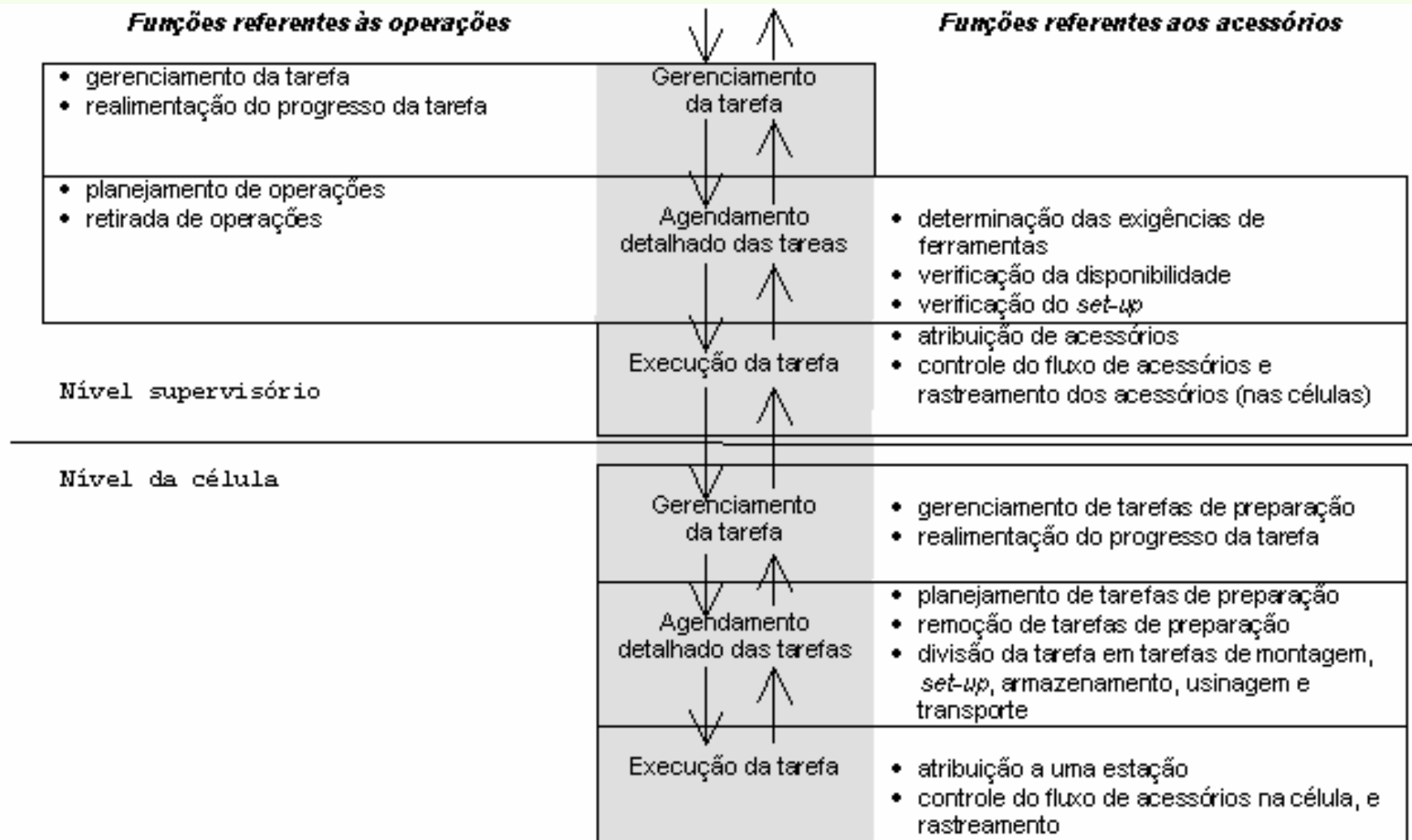
<i>Armazenamento</i>		
Nº de identificação	Localização	Nº da prateleira
1100.2870	Armazenamento A	12
1100.2871	Armazenamento B	17
1100.2872	Armazenamento C	8

<i>Fornecedor</i>			
Nº de identificação	Nome	Nº da Ordem	Preço (R\$)
1100.2870	L1	A12.736	246,36
1100.2870	L2	1348	241,36
1100.2871	L1	A13.924	299,99
1100.2872	L1	A12.878	235,35
1100.2872	L2	1435	230,35
1100.2871	L3	K132.465	220,00
1100.2870	L3	K234.465	231,81

<i>Lista de Peças</i>			
Nº de identificação	Adaptador	Extensão	Corpo
1100.2870	SK.1100	SH.1020	CD.1005
1100.2870	SK.1200	SH.2050	SD.5010
1100.2871	SK.1100	MK.1000	STC.2020

Modelo de banco de dados para o exemplo

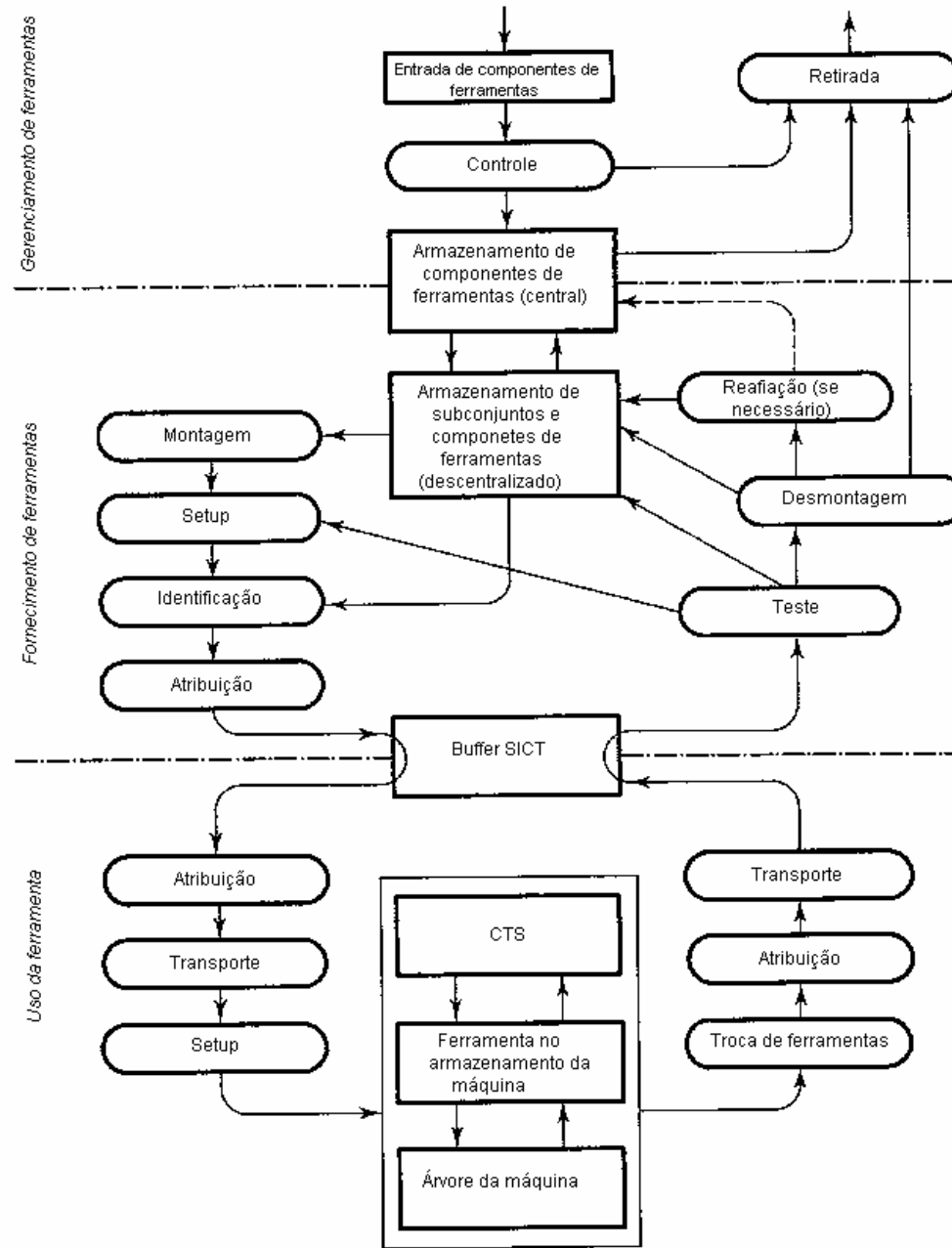
ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR



Distribuição hierárquica de tarefa para o fornecimento de acessórios

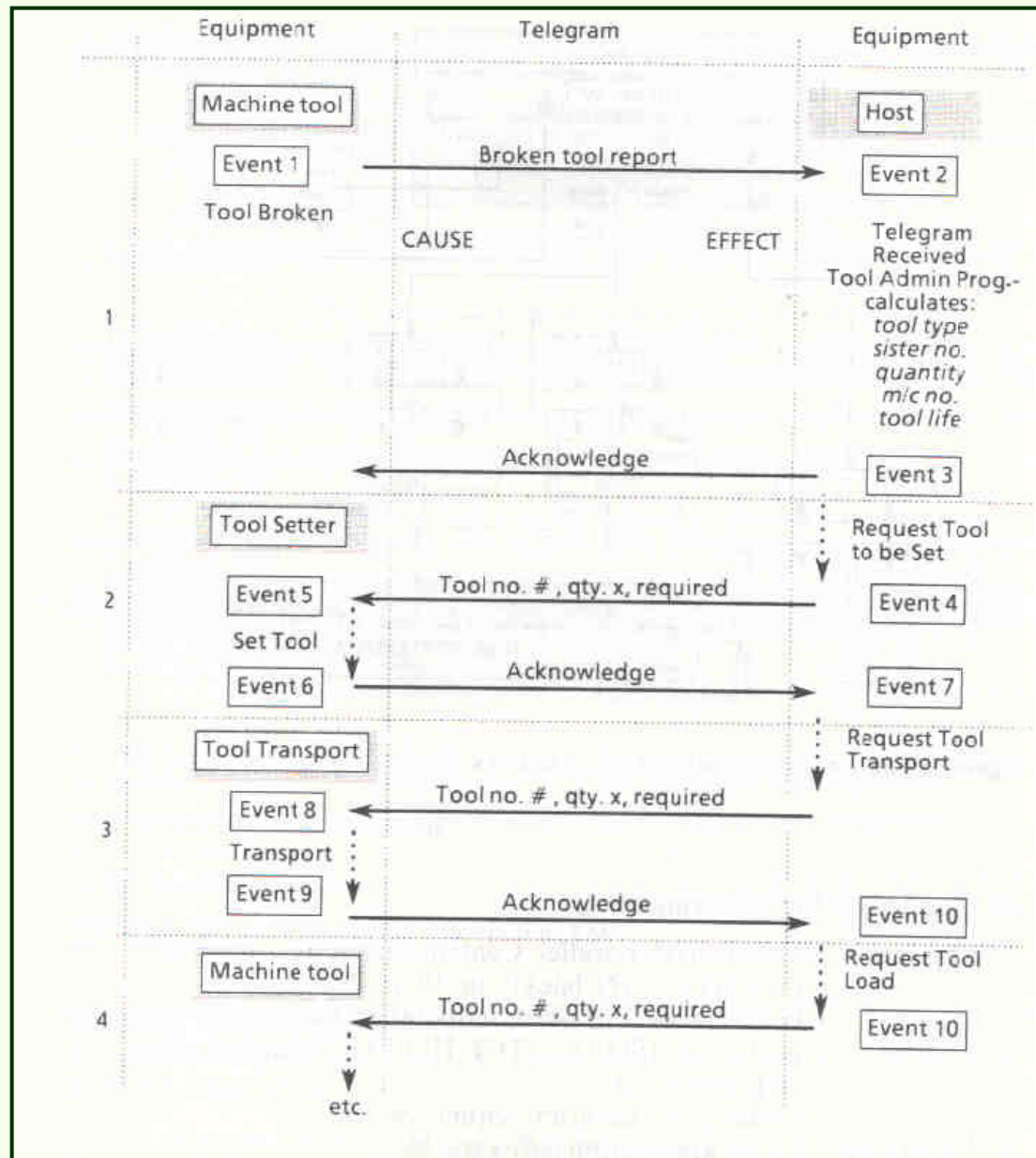
ORGANIZAÇÃO DE ACESSÓRIOS DE MANUFATURA ASSISTIDA POR COMPUTADOR

- Existe um grande potencial para redução de custos que deve ser explorado. Dentre esses incluem:
 - **Sincronização do fluxo de acessórios com o fluxo de peças em vários equipamentos**, em conjunto com a preparação precisa de dados para as áreas de planejamento e controle (ver próxima figura).
 - Assume-se que o sistema de manufatura consiste de várias máquinas e de um armazenamento central de ferramentas (CTS) e também de armazenamentos de ferramentas próximo às máquinas (MTS).



Fluxo de ferramentas

Armazenamento
 Função
 SICT = Ferramenta completa identificada unicamente
 CTS = Armazenamento central de ferramentas



Fluxo de ferramentas

DESENVOLVIMENTO DO SOFTWARE DE CONTROLE DE FMSs

- Desenvolvimento do software de controle de FMS
- Planejamento do processo contendo alternativas
- Paper 3
- Manufatura a distância
- Dissertação do Hugo Gaspar Santos
- Video FMS SOCIESC (trabalho Hugo - RobotStudio)
- Video FMS SOCIESC (trabalho Hugo)