

CAPITULO 3

Conceitos Basicos sobre Barramentos de Campo.

Conceitos gerais, requisitos e classificação;

Barramentos de Campo à luz do Modelo OSI.

(4 Horas)

Nota: Slides em Inglês são da autoria do Professor Jean-Pierre Thomesse, Institut National Polytechnique de Lorraine, Nancy, França

Who's who

A word cloud containing various industrial communication protocols and standards. The words are arranged in a roughly circular pattern around the central title 'Who's who'. The protocols listed include: Ethernet, Hart, TTP, Unitelway, ISO 8802.5, SNMP, WorldFIP, Profibus-PA, Batibus, Sercos, BacNET, IEC 61158, P-NET, SDS, CiA, ICCP, EIBUS, CSMA-BA, EHS, CSMA-DCR, FieldBus Foundation, ControlNet, Interbus, DeviceNet, Profibus-FMS, EN 50254, M-PCCN, Profibus-DP, DWF, Modbus, TTP-A, TCP-IP, FDDI, FIPWay, TTP-C, EN 50170, TASE2, CASM, IEC, ISO 8802.4, WDPF, MMS, ISO 8802.3, Sinec, ControlFIP, PLAN, JBUS, FIPIO, LON, CSMA-CA, Seriplex, TOP, Mini-MAP, CAN, UCA, MAP, F8000, Profisafe, CSMA-CD, Bitbus, ARINC, IEC 6375, CIP, LocaFIP, UIC 556, Digital Hart, M-Bus, WITBUS, VAN, Euridis, Sycoway, GENIUS, OPTOBUS, J1850, SwiftNet.

Position of the fieldbuses

- **Complex problem**
- **Historically**
 - Connection of sensors, actuators, controllers
 - Firstly in industrial applications (short distances)
- **Development of building automation**
- **Development of cars automation**
- **Finally**
 - Real time communication systems
 - All real time applications
 - Short distances and wide area networks

Applications of the fieldbuses

- **Process control**
- **Manufacturing applications, machines, ...**
- **Building applications**
- **Embedded systems, cars, trains, ...**
- **Remote monitoring**
 - Utilities
 - Telemedicine
 - Transport

First networks-1

- **MODBUS, WDPF, SYCOWAY,...**
- **Control room networks**
- **PLC networks**
- **Technical aspects**
 - HDLC issued from telecommunication
 - Process variable image
 - Periodic traffic

Fieldbuses and other networks

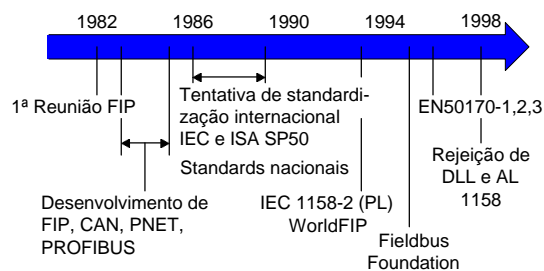
- **Use of several networks**
- **Problem of architectures**
 - Functional architecture
 - Support architecture
 - Operational architecture
- **Interoperability**
- **Fieldbuses and Internet**

Fieldbuses

- **Two main reasons for the fieldbuses origin**
 - Technological development -> connection costs decrease
 - Necessary sharing of the instrumentation
- **Two classes of first fieldbuses objectives**
 - wiring costs decreasing
 - Common and “global” status accessed by all the stations

Barramentos de Campo (*Fieldbuses*)

História da Standardização



Barramentos de Campo (*Fieldbuses*)

Requisitos Genéricos

- **Segurança e tolerância a falhas.**
- **Melhorias na manutenção.**
- **Modularidade e capacidade de evolução.**
- **Abertura, interoperabilidade.**
- **Melhor desempenho e custo inferior.**

Nota: O *fieldbus* em si não é importante.

Barramentos de Campo (*Fieldbuses*)

Necessidades e Restrições

- **De um simples meio de simplificar a cablagem até**
- **Espinha dorsal de aplicações distribuí-das tempo-real.**
- **Neste caso constituem de facto a base de um sistema operativo distribuído e tempo-real.**
- **Devem então:**
 - **Fornecer serviços para facilitar a distribuição das aplicações e o acesso aos dados;**
 - **Garantir a qualidade de serviço necessária para assegurar as restrições temporais e de segurança de funcionamento impostas pela aplicação.**
 - **Assegurar as facilidades necessárias à gestão do sistema, dos equipamentos e à respectiva manutenção.**

Barramentos de Campo (*Fieldbuses*)

Requisitos Específicos

Serviços:

- Distribuição e Configuração da aplicação.
- Acesso à informação distribuída.
- Gestão e Manutenção do sistema.

Qualidade de Serviço:

- Restrições ao nível da tolerância a falhas.
- Restrições temporais.
- Desempenho (p.ex. velocidade).

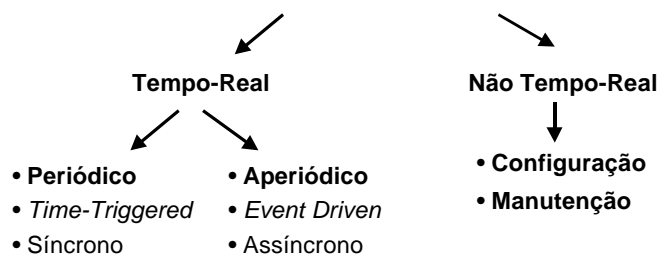
Restrições Físicas:

- Distância, potência.
- Segurança intrínseca, imunidade a ruído.

Barramentos de Campo (*Fieldbuses*)

Necessidades da Aplicação

Tráfego ao nível da aplicação

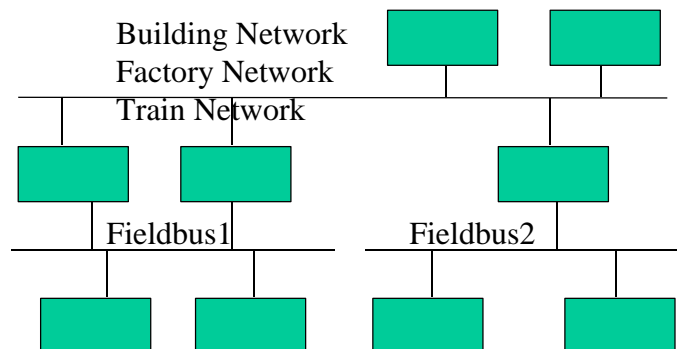


Fieldbus traffics

- **Two types of exchanges in real time applications**
 - Event communication
 - State communication
- **Different characteristics**
 - Time constraints
 - Deadline
 - Lifetime
 - Partners
 - Client - Server
 - Producer - Consumer (s)

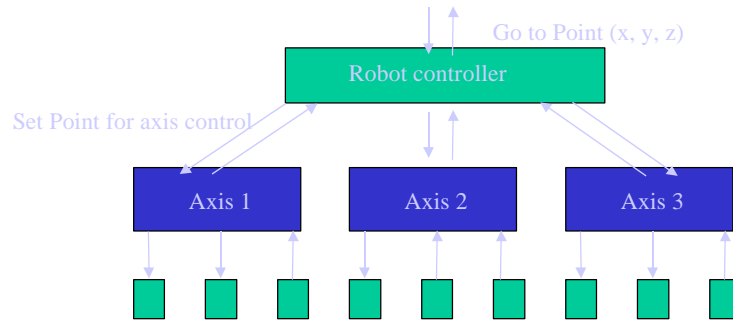
Usual operational architecture

- **Example**



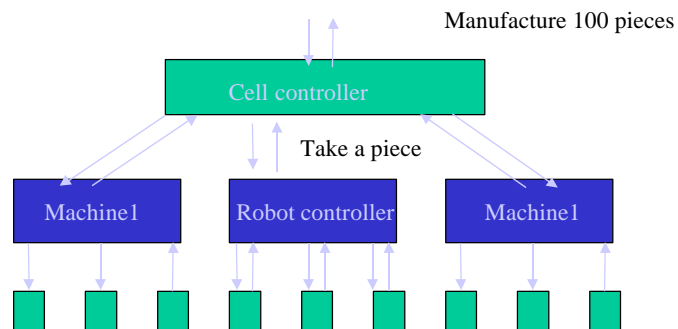
Layered architectures-4

Example 2 : Robot control



Layered architectures-5

Example 3 : Cell Control

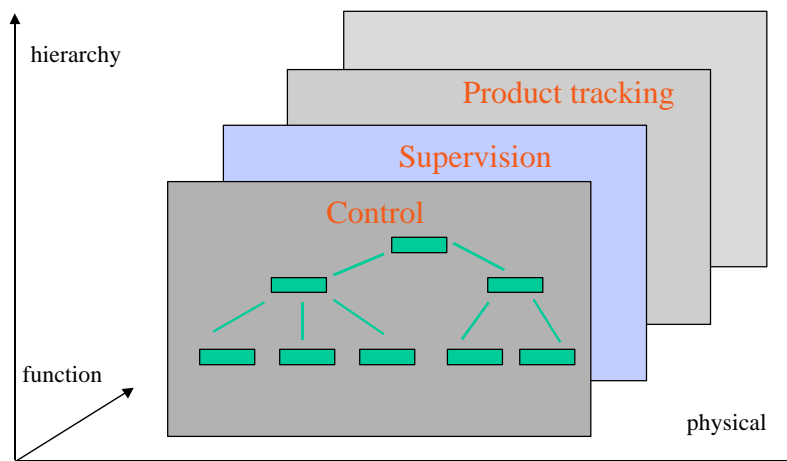


3-Axis model-1

- **3 axis**
 - Hierarchy
 - Physical process
 - Functionalities
- **Plans**
 - One plan for each function
 - (hierarchy, physical process)
- **Traffics**
 - Inside a plan
 - Between plans

3-Axis model-2

Representation

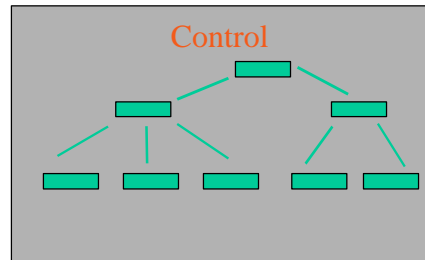


3-Axis model-3

- “Vertical” flows

- Request
- Confirmation

- Indication
- Response



3-Axis model-4

- “Vertical” flow

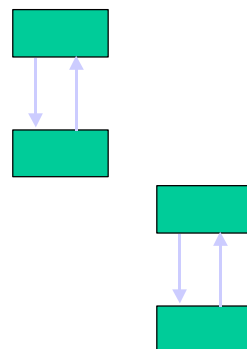
- Client-server model

- Level I+1 ----> Level I

- Request
- Confirmation

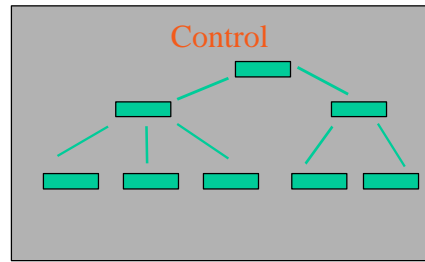
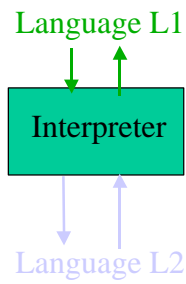
- Level I ----> Level I+1

- Indication
- Response



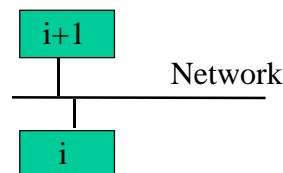
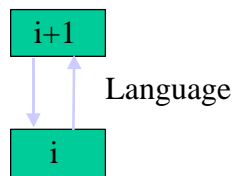
3-Axis model-5

- “Vertical” flow



3-Axis model-6

- Functional schema
- Operational schema



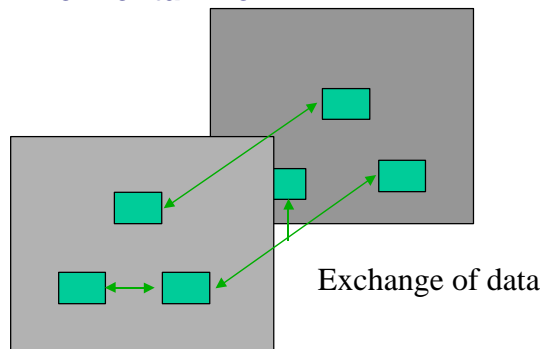
The network transmits requests, confirmations, indications, responses

3-Axis model-7

- “Vertical” flow
- Time constraints
- Response time
 - Delay between Rq and Cnf
 - Delay between Ind and Rsp

3-Axis model-8

- “Horizontal” flow

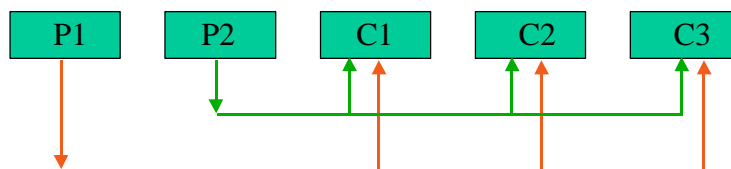


3-Axis model-9

- “Horizontal” flow
- Exchange of data
- No service required by the sender to the receiver
- Multicasting generally
- To inform concerned entities in order to provide a global coherence
- Updating of state variables in concerned entities

3-Axis model-10

- Producer (s) - Consumer (s)
- Send *With or without*
- Receive *acknowledgement*
- Message interpretation by the receivers



3-Axis model-11

- **“Horizontal” flow**
- **Time constraints**
- **No time response**
- **Related to the life time of the state variables**
- **Life time for the producer**
- **Validity times for the consumers**
- **Coherence between all the consumers**

Equipment and computing systems

- **Function**
- **Specified independently of the implementation**
- **Structure of a function decomposed into “logical” nodes**
- **Starting criterion (event, operator, call, ...)**
- **Result of the function (data, event, message, call,...)**
- **Performances and constraints**
- **Exchanges with other functions**

Equipment and computing systems

- **Tasks**
- **Data**
- **Event vs Status**
- **System management**

Synchronous systems

- **Periodic polling of the inputs**
- **Detection of the events (change of state)**
- **Computing of the outputs**
- **Applying of the outputs**
- **Generation of periodic traffic**

Asynchronous systems

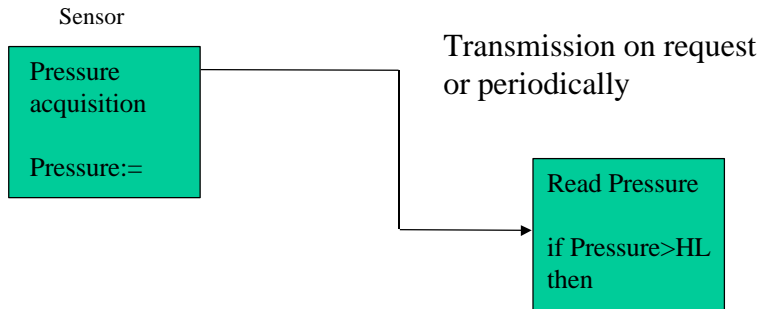
- **Multitasking**
- **Based on events treatment**
- **Tasks**
 - Periodic
 - Sporadic
- **Generation of two types of traffic**

Data

- **Types of data**
 - **Variables values (status)**
 - Physical variables
 - “Logical” variables
 - Different sizes
 - Naturally periodic or not
 - **Changes of values (events)**
 - Naturally periodic or not

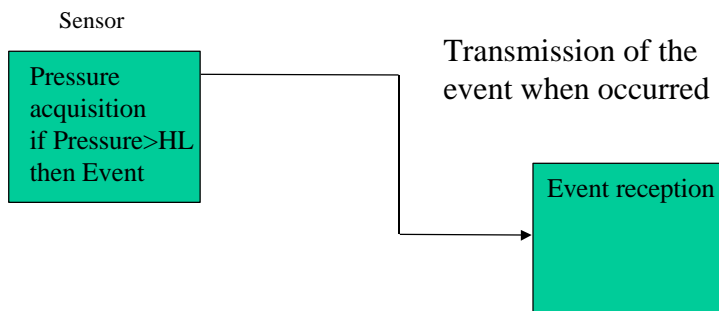
Event vs status-1

- **Status transmission**



Event vs status-2

- **Event transmission**



Event vs status-3

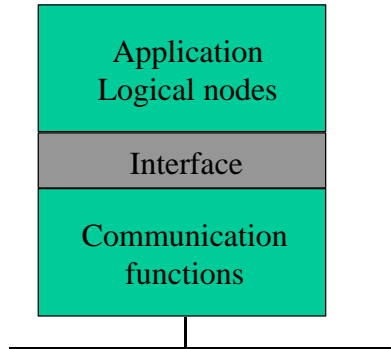
- **Event transmission**
- **Transmission sporadically**
 - Initiated by the server (sensor)
 - No control by the receiver
- **Failure detection by the sender**
 - Protocol with acknowledgement by client
- **Status transmission**
- **Transmission on request**
 - Initiated by the client
 - Waited in a delay
- **Transmission periodically**
 - Initiated by the sensor
 - Initiated by a third party
 - Waited at given moments
- **Failure detection by the receiver**

Static vs Dynamic

- **Static (defined before the run time)**
- **Dynamic (defined at the run time)**
- **This paradigm is one of the most important**
 - Behaviour of the tasks
 - Traffics
 - Scheduling
 - Tasks
 - Traffics

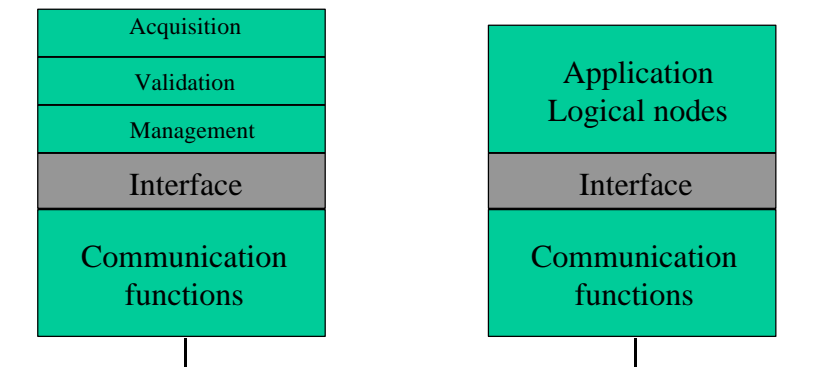
Equipment-2

- **Architecture**



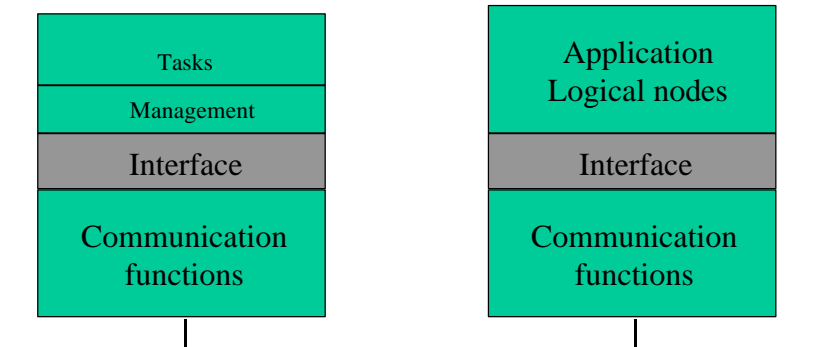
Equipment-3

- **Sensor**



Equipment-4

- PLC



Barramentos de Campo à luz do modelo OSI

- **Modelo OSI: *Open Systems Interconnection Basic Reference Model*.**
- **Projecto lançado pelo ISO (*International Organisation for Standardisation*) em 1977, concluído em 1984 [*ISO 7498*].**
- **Quadro de referência para a interconexão de sistemas abertos, em particular heterogéneos.**
- **Modelo abstracto de referência para elaborar normas de interconexão e de cooperação de sistemas distribuídos.**

Sistemas abertos e proprietários

- **Sistema Aberto:** proporciona a comunicação entre equipamentos de tipos diferentes, de construtores diferentes; têm de ser respeitadas regras de comunicação públicas, acessíveis a todos, eventualmente estruturadas segundo o modelo OSI.
- **Sistema Proprietário (ou privado):** as especificações são apenas utilizadas no sistema e não são, em geral, do domínio público.

O modelo OSI reduzido

- Modelo introduzido nos anos 80 em inúmeras redes entre as quais o projecto MAP (MAP88).
- **Misconception:** a eliminação de camadas não fundamentais é desejável para promover a satisfação de restrições tempo-real (conceito de tempo de *travessia* de camadas).
- **Camada Física:** obviamente necessária.
- **C. Ligação de Dados:** necessária para garantir a detecção de erros de transmissão e fornecer uma comunicação fiável entre duas ou mais estações.
- **Camadas Rede e Transporte :** Assumiu-se que não haveria problema (nem sequer necessidade) de controlar a passagem de tramas por nós ou estações intermédias. Como os dados nestas aplicações são curtos, não há necessidade de empacotamento. O mesmo motivo faz com que a **camada sessão** não seja necessária.

MAP88: MAP 3.0 - Implementation Release subject to errata changes - MAP/TOP Users Group, Dearborn, Julho 88.

Ligação de Sistemas e as 7 camadas do modelo OSI



- Serviços de acesso à informação e comunicação para diferentes tipos de aplicação.
- Transporte de informação, poupando aos níveis superiores os problemas relacionados com a comunicação entre dois equipamentos distantes.

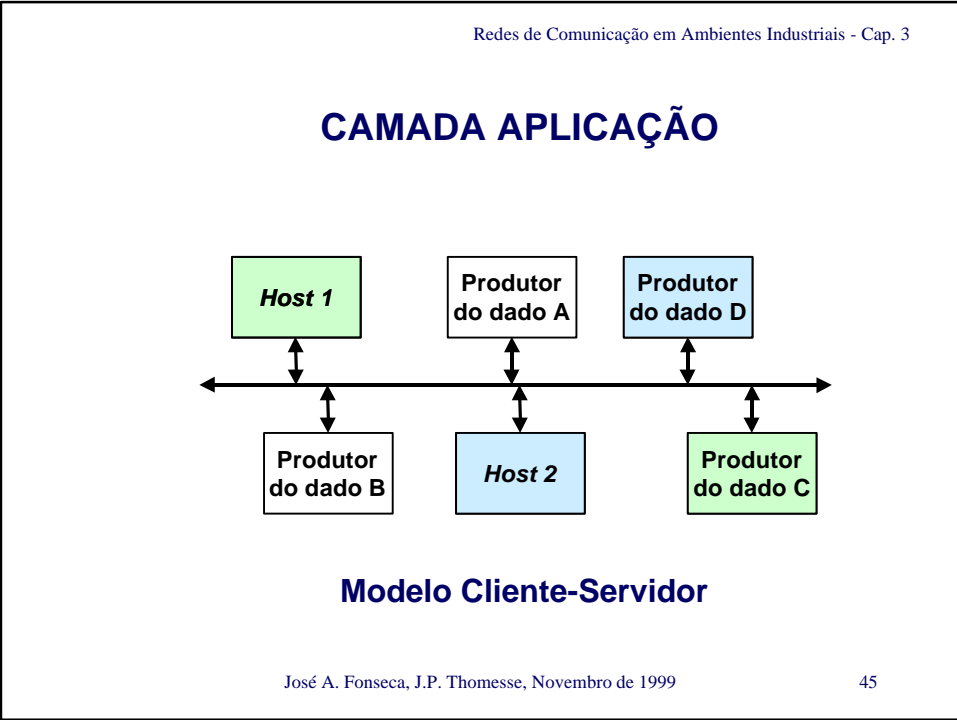
CAMADA APLICAÇÃO

Modelos:

- CS - Cliente Servidor
- PC - Produtor Consumidor
- PDC - Produtor Distribuidor Consumidor

Modelo Cliente Servidor:

- Comunicação ponto a ponto com acknowledge
- Útil para serviços ocasionais associados a transferências de dados esporádicas, configuração, manutenção:
 - Read / Write.
 - Code Loading.
 - Execution control
 - ...



Redes de Comunicação em Ambientes Industriais - Cap. 3

CAMADA APLICAÇÃO

Serviços baseados no modelo CS:

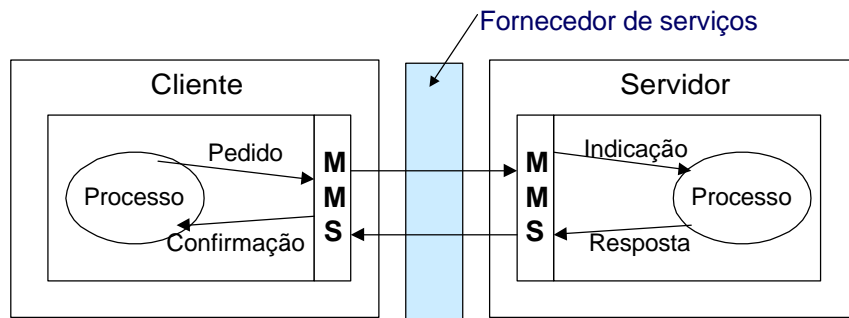
MMS - *Manufacturing Message Specification*

- MMS consiste num conjunto de serviços que permitem encaminhar as mensagens necessárias para a cooperação de tarefas em nodos diferentes e afastados.
- Especialmente concebida para a interligação de máquinas ferramentas.
- Oferece serviços e protocolos para acesso remoto a objectos, semelhantes aos utilizados por sistemas operativos, em particular tempo-real.
- Exemplo de serviços MMS:
 - Acesso a variáveis; semáforos; gestão de tarefas.

José A. Fonseca, J.P. Thomesse, Novembro de 1999 46

CAMADA APLICAÇÃO

Execução de serviços por MMS:



Classes de serviços MMS: Serviços com confirmação
Serviços sem confirmação

CAMADA APLICAÇÃO

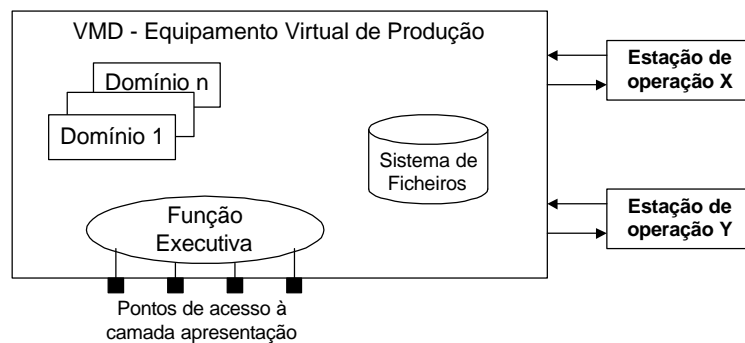
Noção de VMD (*Virtual Manufacturing Device*) associada a MMS:

- Um VMD é uma representação abstracta de um conjunto específico de recursos e funcionalidades de um dispositivo real. Inclui a correspondência da representação com os aspectos físicos e funcionais do dispositivo.
- Num VMD as funções de comando do equipamento são encapsuladas por forma a que o utilizador não necessite de saber como são feitas mas apenas como aceder a elas, comandá-las e/ou controlá-las.

CAMADA APLICAÇÃO

- **Por exemplo:**
 - Ligar um motor
 - Comandar um eixo de um robot**são funções reais que não interessam ao utilizador.**
- **O dispositivo real é representado por um conjunto de objectos visíveis e acessíveis do exterior:**
 - Velocidade de rotação de uma furadora, posição de um braço de robot, ...
- **Um VMD inclui uma colecção de recursos de diferentes tipos: variáveis, programas instanciáveis, ficheiros, ...**
- **Elementos de um VMD:**
 - Função executiva - Um ou vários domínios
 - Várias estações de operador (opcional)
 - Um sistema de ficheiros (opcional).

CAMADA APLICAÇÃO



Estrutura de um VMD em MMS

CAMADA APLICAÇÃO

- **Função Executiva:**
 - Gera todos os acessos aos recursos do VMD e executa os pedidos correspondentes a qualquer serviço MMS.
 - Deve ser programada pelo utilizador para definição do comportamento da sua aplicação servidor.
- **Domínio:**
 - Representa um subconjunto dos recursos de um VMD utilizados para um fim particular (coordenação de acções, comando de processos,...).
 - Pode conter ou não “informações”: programas ou partes de programas, dados, tabelas, ...
 - A norma MMS não indica como constituir um domínio.
 - Exemplo num Robot: - gestão do movimento;
- manutenção; - estudo de desempenho.
- ou um único domínio com todas as funções.

José A. Fonseca, J.P. Thomesse, Novembro de 1999

51

CAMADA APLICAÇÃO

- **Sistema de Ficheiros:**
 - Colecção de ficheiros de dados e programas utilizados pelos domínios e pelos serviços de gestão de programas para: operações de *upload* e *download* de programas, operações de arquivo, ...
 - Um VMD pode possuir ou não um sistema de ficheiros. Se sim, pode fazer o *upload* ou *download* de domínios utilizando as suas capacidades locais de armazenamento; Se não, tem de recorrer a outros VMDs.
 - Os serviços oferecidos devem constituir um subconjunto da norma FTAM (*File Transfer Access and Management* - ISO 8571).
- **Estação de operação:**
 - Meio que permite ao VMD efectuar entradas e/ou saídas, ou seja, interagir com o meio que o rodeia (aquisição de dados, imposição de *setpoints*, *display* de informação, ...). Pode não existir.

José A. Fonseca, J.P. Thomesse, Novembro de 1999

52

CAMADA APLICAÇÃO

- Um VMD é caracterizado por:
 - Nome do vendedor, modelo e versão do modelo.
 - Estado lógico de funcionamento com indicação das operações que pode implementar: todos os serviços, leque restrito de serviços, possibilidade ou não de alteração dos seus atributos, ...
 - Indicação de estado físico: em funcionamento (integral ou parcial), fora de serviço, aguardando manutenção, ...
 - Lista de objectos nele definidos.
 - Lista de recursos.

CAMADA APLICAÇÃO

- Objectos MMS:

Numa aplicação com base em MMS, o utilizador pode definir vários tipos (ou classes) de objectos:

 - *Unnamed variables.* - *Named variables.*
 - *Named variable lists.* - *(Scattered Variables).*
 - *Named types.* - *Semaphores.*
 - *Events (event conditions, actions and enrollments).*
 - *Journals.* - *Domains*
 - *Program invocations.* - *Operator Stations.*
 - *Files.* - *O próprio VMD.*
- Cada objecto representa uma entidade abstracta que possui atributos que podem ser consultados ou modificados por serviços MMS, podendo ser criado ou destruído, localmente ou à distância.
- Um objecto é definido pela classe, nome e atributos.

Camada Aplicação: Serviços MMS

- **Gestão de contextos / associações:**
Serviços para apoio ao estabelecimento de comunicações entre duas aplicações, implicando a alocação de recursos, estabelecimento de regras de codificação, de comprimento máximo de mensagens.
 - *Initiate* - *Conclude* - *Abort*
 - *Cancel* - *Reject*
- **Gestão de VMDs:**
 - *Status* - *UnsollicitedStatus* - *GetNameList*
 - *Identify* - *Rename* - *GetCapabilityList*
- **Gestão de Estações de Operador:**
 - *Output* - *Input*

Camada Aplicação: Serviços MMS

- **Gestão de domínios:**
Um domínio representa um subconjunto dos recursos de um VMD, permitindo a partição das funcionalidades dos equipamentos.
Os domínios podem estar cheios ou vazios e podem ser criados de forma estática ou dinâmica.
Os domínios servem de suporte a teleconfiguração e ou telecarga de estações de trabalho (PLCs, Robots, ...). Para tal esses equipamentos têm de deter as funcionalidades necessárias (interface para rede, *bootstrap/loader*, ...)
 - *InitiateDownloadSequence* - *DownloadSegment*
 - *TerminateDownloadSequence* - *UploadSegment*
 - *InitiateUploadSequence* - *LoadDomain Content*
 - *DeleteDomain* - *GetDomainAttributes*
 - ... - ...

Camada Aplicação: Serviços MMS

- **Gestão de tarefas: (*Program Invocations*)**

A expressão física de uma tarefa obtém-se pela união de um ou vários domínios onde estão guardados o código e os dados correspondentes.

A execução de tarefas num servidor pode ser controlada à distância.

Uma tarefa é caracterizada pelo nome, estado (activa, em execução, suspensa) e domínios onde reside.

Serviços:

- *CreateProgramInvocation* - *DeleteProgramInvocation*
- *Start* - *Stop* - *Resume* - *Reset*
- *Kill* - *GetProgramInvocationAttributes*
- ... - ...

Camada Aplicação: Serviços MMS

- **Gestão de variáveis:**

Unnamed: correspondem a endereços físicos nos equipamentos; são intrínsecas ao VMD não podendo ser criadas ou destruídas.

Named: Identificam variáveis que podem ser criadas ou destruídas num VMD.

List: reúne num só nome um conjunto de variáveis MMS, named ou unnamed.

Type: idêntico às linguagens de programação (Boolean, bit string, integer, unsigned, float, BCD, ...).

- *Read* - *Write* - *InformationReport*
- *DefineNamedVariable* - *DeleteVariableAccess*
- *GetVariableAccessAttributes* - *Define NamedType*
- *DefineNamedVariableList* ...

Camada Aplicação: Serviços MMS

- **Gestão de semáforos:**

Os semáforos permitem a sincronização, o controlo e a coordenação entre utilizadores MMS:

- Acesso aos objectos MMS
- Acesso a recursos físicos
- Sincronização entre aplicações.

Token: tokens idênticos, # livres / # ocupados, pode ser criado ou apagado dinamicamente.

Pool: tokens identificados, criado na configuração.

Serviços:

- *TakeControl* - *RelinquishControl* - *DefineSemaphore*
- *DeleteSemaphore* - *AttachToSemaphore*
- *ReportSemaphoreStatus* - *ReportPoolSemaphoreStatus*
- *ReportSemaphoreEntryStatus*

Camada Aplicação: Serviços MMS

- **Gestão de jornais:**

Os objectos deste tipo permitem o registo e a pesquisa de informação ordenada cronologicamente: acontecimentos, variáveis, textos (cada VMD pode ter nenhum ou vários jornais).

Conteúdo: nome do cliente fonte da informação, datação e informação.

Serviços:

- *CreateJournal* - *DeleteJournal* - *ReadJournal*
- *WriteJournal* - *InitializeJournal*
- *ReportJournalStatus*

Camada Aplicação: Serviços MMS

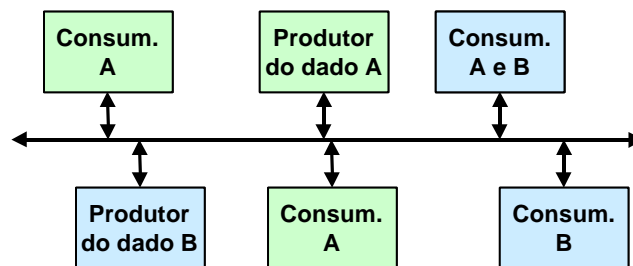
- **Gestão de ficheiros:**

Nota: não é permitida a alteração do conteúdo de um ficheiro.

Serviços:

- *FileOpen* - *FileRead* - *FileClose*
- *FileRename* - *FileDelete* - *FileDirectory*
- *ObtainFile*

CAMADA APLICAÇÃO



Modelo Produtor-Consumidor

Camada Ligação de Dados

Aspectos importantes:

- a) Como aceder ao meio de comunicação partilhado.
 - b) Identificação do ou dos nós a quem se destina a informação.
 - c) Existência de mecanismos para assegurar a correcta transferência de informação.
- **MAC - Medium Access Control**
 - **Endereçamento**
 - **LLC - Logical Link Control**

Camada Ligação de Dados

MAC

- Normalmente considerada uma subcamada do DLL.
- Permite evitar conflitos que podem vir a corromper a informação a transmitir.
- Os protocolos utilizados têm uma influência directa na capacidade de resposta temporal do fieldbus.

Camada Ligação de Dados

Endereçamento

- **Influencia a capacidade do fieldbus de suportar modelos de aplicação (cliente/servidor ou produtor/consumidor).**
- **Determina a possibilidade de multicast, broadcast.**

Camada Ligação de Dados

LLC

- **Normalmente considerada uma subcamada do DLL.**
- **Nalguns fieldbuses não está claramente separada da subcamada MAC.**
- **Serviços típicos: Empacotamento, mecanismos de detecção e correcção de erros, estabelecimento de ligações entre nós, *acknowledge*.**

Camada Ligação de Dados

Principal divisão do MAC:

- **Acesso controlado.**
 - **Controlo distribuído.**
 - **Controlo centralizado.**
 - **Acesso controlado explicitamente (p.ex. token).**
 - **Acesso controlado implicitamente (p.ex. tempo).**
- **Acesso não controlado.**

Camada Ligação de Dados

Controlo centralizado: Master-Slave

- **Um nó específico, o Master, faz acessos cíclicos (polling) aos nós secundários, os Slaves.**
- **Simple de implementar e com comportamento previsível.**
- **Resposta temporal depende essencialmente do Master desde que os Slaves respondam aos polls num intervalo de tempo de duração limitada.**
- **Bem adaptados à transmissão de tráfico periódico.**

Camada Ligação de Dados

Controlo centralizado: Master-Slave

- Normalmente adoptam o modelo C/S (por vezes com facilidade de broadcast/multicast para os Slaves).
- Bem adaptados quando a capacidade de processamento está essencialmente concentrada num nó (o Master).
- PROFIBUS, MIL-STD1553B, Bitbus

Camada Ligação de Dados

Controlo centralizado:

PDC - Produtor, Consumidor, Distribuidor

- O acesso dos Produtores ao Bus é controlado por um nó central chamado árbitro.
- O árbitro endereça uma variável (entidade lógica contendo informação) em vez de endereçar a estação que confere ao Produtor o direito de utilizar o bus.
- As variáveis/entidades têm de ter IDs únicos para evitar colisões.

Camada Ligação de Dados

Controlo centralizado:

PDC - Produtor, Consumidor, Distribuidor

- O consumo simultâneo pelos Cs permite manter consistência espacial da informação distribuída no sistema.
- Bem adaptado ao tráfico periódico.
- Mal adaptado ao tráfico aperiódico devido aos pedidos de comunicação dos nodos serem dinâmicos.

Camada Ligação de Dados

Controlo centralizado:

PDC - Produtor, Consumidor, Distribuidor

- Tabela de requisitos no árbitro.
- Escalonamento e tabela de dispatching.
- Verificação da escalonabilidade.
- Técnica de produção das variáveis periódicas.
- Pedidos e produção de aperiódicas.
- FTT - CAN: Uma versão PDC sobre CAN.

Barramentos de Campo (*Fieldbuses*)

MAC e LLC

- **MAC por endereçamento:**
 - Centralizado, estação mestre concede acesso ao barramento enviando explicitamente uma mensagem.
 - Dois tipos: endereçamento de estações (Interbus, Bitbus), identificação de objectos (WorldFIP).
- **Multiplexagem temporal (TDMA)**
 - TDMA - *Time Division Multiple Access*.
 - É atribuído periodicamente um tempo de transmissão a cada estação endereçada implicitamente. A transmissão inicia-se num instante indicado localmente pelo relógio da estação.
 - Os relógios podem ser coordenados centralizadamente (SERCOS) ou de forma distribuída (TTP - Time Triggered Protocol - Kopetz).

Barramentos de Campo (*Fieldbuses*)

MAC e LLC (cont.)

- **Passagem de Testemunho (token) :**
 - A estação que detém o token pode emitir durante um período de tempo com limite máximo, devendo então passá-lo a outra.
 - Em Fieldbuses esta técnica só é utilizada em PROFIBUS associada a endereçamento.
- **MAC tipo CSMA**
 - CAN e higher layer protocols baseados nele: DeviceNET, SDS, CANopen.
 - Outras variantes para resolução determinista de colisões (Le Lann e Rivierre, 1994).

Barramentos de Campo (*Fieldbuses*)

Alguns standards

Os standards mais divulgados:

- **PROFIBUS (origem: Alemanha)**
- **WorldFIP (FIP - *Factory Instrumentation Protocol*; origem: França)**
- **CAN (CAN - *Controller Area Network*; origem: Alemanha)**

Barramentos de Campo (*Fieldbuses*)

PROFIBUS

Características fundamentais:

- **Dois níveis funcionais:**
 - *Token Ring* lógico
 - *Master-Slave*
- **Acesso aos dispositivos de campo segundo modelo cliente-servidor.**
- **Versões bem adaptadas aos mais diversos ambientes industriais.**

Barramentos de Campo (*Fieldbuses*)

PROFIBUS

**Figuras do PROFIBUS podem ser encontradas
nos sites indicados para este standard.**

Estrutura de comunicações industriais baseada em PROFIBUS

José A. Fonseca, J.P. Thomesse, Novembro de 1999

77

Barramentos de Campo (*Fieldbuses*)

PROFIBUS

**Figuras do PROFIBUS podem ser encontradas
nos sites indicados para este standard.**

Modelos de Comunicação

José A. Fonseca, J.P. Thomesse, Novembro de 1999

78