

## Esigenze dei sistemi per controllo di processo e loro legame con il sistema di comunicazione.

In un sistema distribuito per controllo di processo, le prestazioni ottenibili sono strettamente dipendenti dalle caratteristiche del sistema di comunicazione da cui dipende lo scambio informativo fra i vari processi applicativi.

Livello più critico: livello di campo

### Principali vantaggi dell'uso del Fieldbus:

- riduzione nel numero e costo dei cablaggi,
- riduzione del costo di installazione, operazione, e manutenzione,
- migliore accuratezza nelle misure
- informazioni tempestive per la diagnostica dei dispositivi
- possibilità di condividere informazioni

**PROBLEMA: LA SERIALIZZAZIONE DEL TRAFFICO**

## Parametri fondamentali per la caratterizzazione del sistema di comunicazione

- Il **Throughput** misura la quantità di informazione che il sistema di comunicazione è in grado di trasferire:
  - influenza il numero di dispositivi che possono essere connessi
- La **Timeliness** misura la tempestività con cui l'informazione viene trasferita fra i vari processi:
  - in processi non "time critical" essa può essere espressa in termini di tempo di ritardo medio,
  - Nel caso di processi "time critical" può rappresentare l'abilità di consegnare le variabili periodiche entro opportune "deadlines",
  - oppure può rappresentare l'abilità a supportare tempestivamente lo scambio di sequenze di dati attivate da particolari eventi (es, allarmi).
- L'**Affidabilità** dipende da caratteristiche del protocollo di comunicazione, il tipo di mezzo fisico usato, le caratteristiche dell'Hardware del sistema, l'uso di ridondanze e di opportune strategie di recupero.

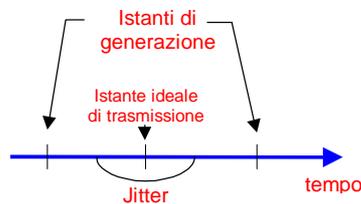
## Caratteristiche del flusso dati Periodico

E' generato da **processi periodici** che eseguono azioni ripetute ad una frequenza costante.

Ciascun dato prodotto periodicamente viene consumato da uno o piu' processi consumatori,



L'istante reale di trasmissione è sempre collocato in un intorno dell'istante ideale, noto come **jitter**.



## Caratteristiche del flusso dati Asincrono

- Generato da **processi che evolvono nel tempo, in un modo imprevedibile a priori.**
- **Indipendenza tra l'attività sul mezzo di comunicazione** (scandite da un clock di sistema) **e quelle dei processi produttori del traffico.**
- informazioni che prodotte in maniera casuale non caratterizzate da vincoli temporali particolarmente stringenti. (Es. **interrogazioni ad una banca dati o file transfer.**)
- informazioni che sono prodotte solo una volta e che sono caratterizzate da stringenti vincoli temporali (Es. **allarmi**)
- informazioni che sono prodotte periodicamente e che sono caratterizzate da stringenti vincoli temporali. La produzione inizia in istanti non prevedibili a priori, e poi si ripete nel tempo in modo periodico (Es. **attivazione di un encoder.**)

## Esempio: FieldBus per il controllo nell'industria di processo.

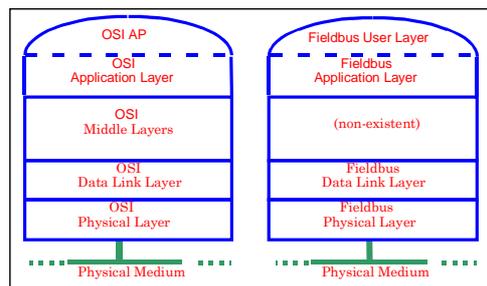
- Caso sviluppato da NAMUR principalmente per rispondere alle esigenze dell'industria chimica. Il Fieldbus viene utilizzato per interconnettere sensori/attuatori intelligenti e/o convenzionali, con i PLC.
- lunghezza del bus fino a **1000m**, con diramazioni dell'ordine di 10 -20 metri
- Possibilità di utilizzare **diverse topologie**
- Fino a **30 dispositivi** direttamente connessi ad un singolo Bus
- Un impianto completo deve contenere diversi Fieldbus (da 10 a 100)
- Possibilità di utilizzare strutture di **bus ridondanti**.
- Uso di "**Twisted pair**" con possibilità di uso opzionale di fibra ottica
- Necessità di **alimentazione attraverso il Bus**.
- Possibilità di una versione in **Sicurezza intrinseca**.
- Un dato è di norma usato da un solo dispositivo
- Tempo di ritardo tra il dispositivi sorgente e destinazione dell'ordine di **100 msec**.
- Possibilità di **aggiungere, rimuovere o cambiare** i dispositivi senza la necessità di bloccare il funzionamento di altri dispositivi. Inoltre, il guasto di un componente non deve propagarsi e compromettere il funzionamento del Fieldbus.

## Esempio: FieldBus per i control loop .

- Questo caso è stato sviluppato dall'ISA SP50 per le specifiche H1. Nel caso base considerato, il Fieldbus deve essere capace di interconnettere 5 nodi e collegare i seguenti dispositivi:
  - 1 equipment collegato al processo
  - 1 indicatore locale o portatile (opzionale)
  - da 1 a 4 equipment nella "control room" al di fuori della zona pericolosa.
- La comunicazione deve essere in grado di fornire le seguenti prestazioni:
  - Frequenza di aggiornamento (scan rate) 4 volte al secondo.
  - Dati utente di 5 bytes per ogni dispositivo connesso.
  - Necessità di spedire messaggi, per informazioni ausiliarie, nella misura del 40% rispetto ai dati utente.
- Distanza massima di 1900 metri fra i dispositivi.
- Necessità di alimentazione attraverso il Bus.
- Uso di due twisted-pair in un singolo cavo multiplo.

## Punti chiave nell'architettura dei Fieldbus

- non tutte le funzionalità supportate dal modello OSI sono necessarie;
- le funzionalità previste non sono sufficienti per una applicazione e ne vanno introdotte di nuove;
- alcune funzionalità possono essere spostate in altre parti dell'architettura;
- la complessità del modello impedisce il soddisfacimento di vincoli temporali.



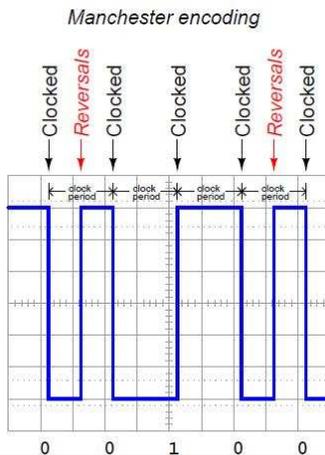
Per tutte queste ragioni, le architetture dei vari tipi di Fieldbus appaiono notevolmente diversificate. In generale la maggior parte dei Fieldbus utilizza una architettura a tre soli livelli

## Caratteristiche del Physical layer

- Il Physical layer è responsabile della **codifica dei dati** e della risoluzione dei problemi di interfacciamento al mezzo fisico.
- Da esso dipendono la **distanza** a cui è possibile trasmettere le informazioni, la **Banda** consentita, il **numero di dispositivi** che è possibile connettere al mezzo fisico.
- **Power on the Bus**
- **Intrinsic safety**

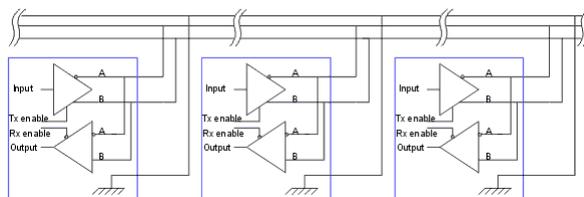
## Codifica Manchester

- Una delle codifiche più usate è la codifica Manchester che si caratterizza per il fatto di essere **trasparente** (rispetto ai dati) e **self clocking**



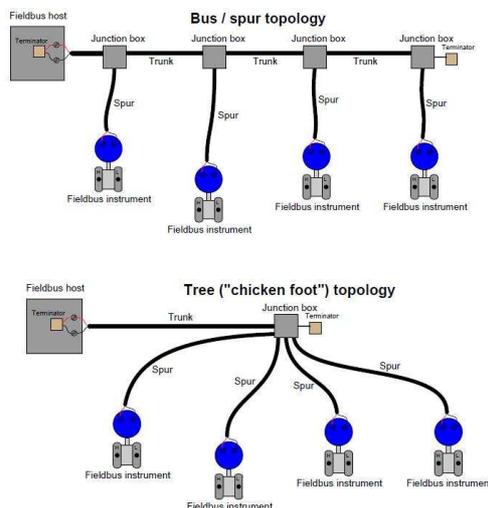
## EIA RS-485

- E' una specifica a livello fisico di una connessione seriale a due fili **half-duplex** e **multipoint**.
- Lo standard specifica un sistema di gestione del segnale in forma differenziale: una polarità indica un livello logico 1, quella inversa indica il livello logico 0.
- La differenza di potenziale deve essere di **almeno 0,2 V**, ma qualsiasi tensione compresa tra +12 V e -7 V è accettata.
- Supporta le linee multi-drop, in cui coesistono più ricevitori e trasmettitori sulla stessa coppia di fili. Al fine di evitare conflitti è ovviamente necessario che **un solo trasmettitore alla volta** sia attivo



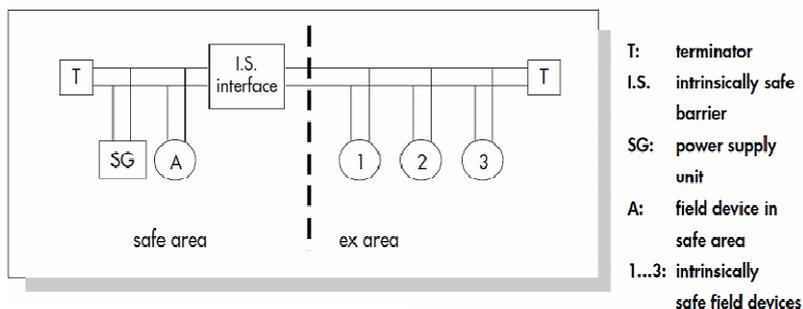
## Topologie dei BUS

- Quelle più usate sono a **Bus** o a **stella**



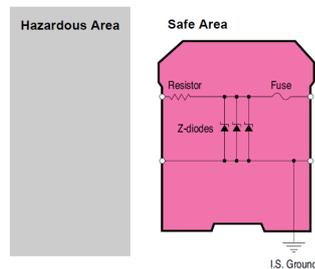
## Intrinsic Safety

- Intrinsically safe equipment è definito come “**equipment and wiring which is incapable of releasing sufficient electrical or thermal energy under normal or abnormal conditions to cause ignition of a specific hazardous atmospheric mixture in its most easily ignited concentration.**”
- Ciò viene ottenuto limitando la potenza disponibile sul mezzo, ad un livello inferiore a quello necessario per accendere il GAS.



## Barriere a Diodo Zener

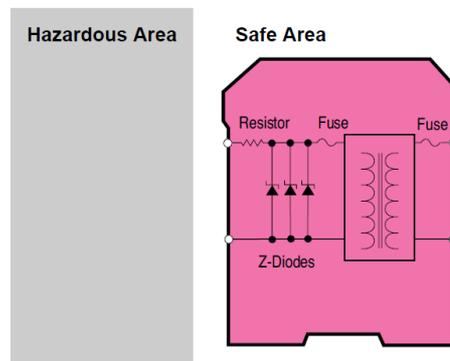
- Le barriere a diodo Zener **limitano l'energia** nell'area pericolosa ad un livello inferiore a quello necessario per accendere il miscuglio aria/gas.
- Ciò viene ottenuto mediante una protezione contro i seguenti tipi di faults:
  - **1.Cortocircuito dei cavi connessi al lato "hazardous area"** della barriera.
  - **2.Cortocircuito verso terra dei cavi connessi al lato "hazardous area"** della barriera.
  - **3.Errato collegamento o guasto dell'alimentazione (power supply)** che consente ad una tensione pericolosa di essere applicata al lato "Safe" della Barriera.



La barriera impedisce alle tensioni pericolose di trasmettersi alla "hazardous area" come conseguenza di queste condizioni.

## Barriera a Trasformatore di Isolamento

- Una **Transformer Isolation Barrier (TIB)** contiene una "zener barrier" per limitare la tensione e la corrente.
- Il trasformatore in una TIB fornisce un **elevato grado di isolamento** fra gli avvolgimenti primario e secondario.
- **Non è richiesta una connessione alla terra** come invece necessario per la barriera a diodi zener.



## Influenza del DLL sul Throughput

- Il Throughput dipende essenzialmente da **uso di elevati Bit rate** nel Physical layer ed **utilizzo efficiente della banda**
- L'uso di valori elevati di Bit rate ha un certo costo:
  - hardware più costoso.
  - uso di cavi di migliore qualità,
  - maggiore sensibilità ai disturbi
  - limite della massima lunghezza che può essere coperta dal bus.
- L'uso efficiente della banda dipende essenzialmente dal **formato delle frames** e dall'uso di **opportune strategie di accesso** ai mezzi fisici.
- Il formato delle frames del DLL, con i vari campi di supporto influenza l'efficienza trasmissiva. **Tutti i protocolli sono equivalenti** tranne **CAN** ed **Interbus-S**

## Influenza del DLL sul Throughput

- **Gestione centralizzata dell'accesso.**
  - adottata nel Fieldbus standard IEC 1158 e nel Fieldbus WorlFip.
  - non è molto efficiente poichè ogni frame utile deve essere preceduta da una **frame di servizio** (il token).
- **Gestione distribuita dell'accesso.**
  - adottata nel Profibus,
  - prevede la circolazione di un token
  - la **perdita di efficienza è limitata** poichè i token sono in numero molto minore rispetto alle frames utili
- **Gestione distribuita dell'accesso, di tipo CSMA.**
  - utilizzata dalla rete Ethernet e dal CAN
  - può essere **molto efficiente** poichè non prevede l'uso di token.
  - In Ethernet le collisioni possono degradare l'efficienza.
- **Approccio Single-frame**
  - usando una sola frame per tutte le stazioni, **limita in maniera massiccia l'overhead** costituito dalle informazioni di supporto.

## Influenza del DLL sulla Timeliness.

Il DLL, condiziona fortemente il comportamento di un Fieldbus dal punto di vista della timeliness.

**Gestione centralizzata dell'accesso:** il DLL affida ad una particolare stazione (Il **Link Active Scheduler**, nel Fieldbus standard IEC 1158 ed il **Bus Arbitrator**, nel WorldFip) la responsabilità per l'accesso al mezzo.

- nel caso di traffico periodico è necessario fare **uso di una tabella di schedulazione** (statica, dinamica)
- La generazione della tabella di schedulazione può rappresentare **un'operazione complessa** quando bisogna schedulare molte variabili con dinamica diversa
- questo approccio è **ottimo per quanto concerne la timeliness.**
- **Non è efficiente per il traffico asincrono**
- il **LAS fa uso di un token**
- il **B.A. colleziona esplicitamente le richieste** asincrone.

## Influenza del DLL sulla Timeliness.

- **Gestione distribuita dell'accesso:** l'accesso al mezzo fisico avviene mediante il possesso di un **token**.
- Per il traffico periodico, occorre definire un parametro chiamato "**Target Token Rotation Time, TTRT**" che limita il tempo max che un token può impiegare per compiere un giro. Presenta tre limitazioni:
  - ogni stazione acquisisce il token con una **periodicità costante come valor medio**.
  - quando sono presenti dati periodici con periodi differenti, il TTRT dovrà essere **fissato in relazione al dato più critico**.
  - la banda viene **distribuita in modo equo** fra le varie stazioni e ciò potrebbe costituire un problema quando una stazione produce più traffico delle altre.
- Per il traffico asincrono, l'approccio distribuito è **soddisfacente**.

## Influenza del DLL sulla Timeliness.

- **Gestione distribuita basata su protocolli di tipo CSMA:** ogni stazione tenta di accedere al mezzo fisico, non appena questo appare libero.

### Collisione

- **In Ethernet:**
  - non è possibile **distinguere** fra traffico ciclico ed asincrono.
  - il comportamento del protocollo **non è deterministico**
  - definendo opportunamente **l'ambiente operativo** (numero di stazioni, workloads, uso di Hubs e Switches, ecc) può essere adottata in diverse applicazioni
- **In CAN**
  - la collisione **non distrugge** tutti i messaggi coinvolti ma fa sopravvivere quello a più alta priorità.
  - permette di introdurre una sorta di **schedulazione automatica on-line**

## Influenza del DLL sulla Timeliness.

- **Gestione centralizzata basata su Single frame.**
  - utilizza un **Master** che spedisce una unica frame per scambiare dati con tutte le stazioni
  - Poichè lo scambio avviene sotto il controllo di un master, le cui temporizzazioni sono state fissate in fase di configurazione, la **Timeliness è pienamente rispettata per il traffico ciclico.**
  - il traffico asincrono, questo fa uso di **fette di banda** rese disponibili allo scopo
  - il contesto per il quale Interbus-S è stato progettato, è quello dei sistemi di sensori/attuatori che scambiano solo **brevi informazioni**, ed in cui **la maggior parte del traffico è ciclica.**

## Influenza del DLL sull'Affidabilità.

- **Gestione centralizzata dell'accesso:**
  - Il Master della comunicazione, costituisce un punto di vulnerabilità del sistema.
  - Un guasto nel LAS o nel BA blocca l'intero sistema;
  - Presenza di un Master di riserva
  - Tempo morto per la sostituzione del Master guasto.
- **Gestione distribuita dell'accesso (con token)**
  - non esiste un unico punto di vulnerabilità del sistema
  - Tempi morti per la riconfigurazione del ring logico
- **Gestione distribuita basata su protocolli di tipo CSMA**
  - il guasto di una stazione ha una influenza locale, e non determina malfunzionamenti sull'intero sistema
- **Gestione centralizzata basata su Single frame:**
  - il sistema è centralizzato e non prevede alcuna ridondanza;
  - un guasto della stazione Master, produce inevitabilmente l'arresto dell'intero sistema di controllo.

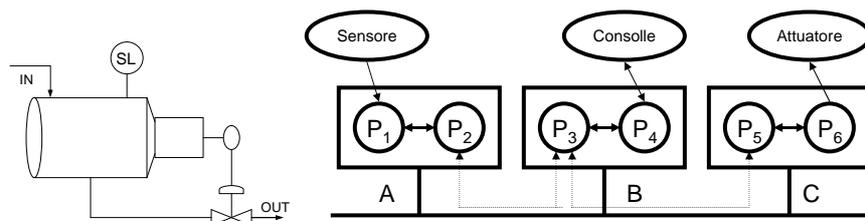
## Application Layer

A differenza degli altri livelli i suoi utenti non sono Entità ma **Processi Applicativi**.

Consente ai processi applicativi (End users) di accedere a tutte le funzioni della rete.

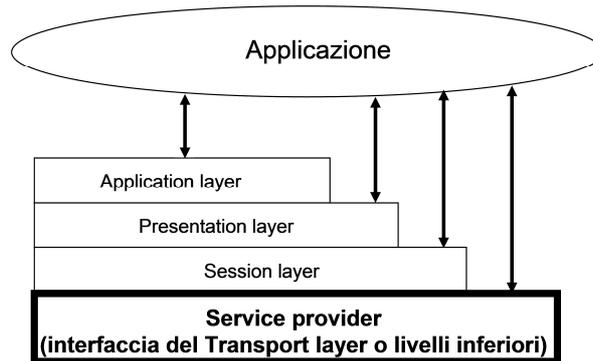
Sebbene più processi applicativi possano risiedere nella stessa stazione di una rete di calcolatori, i protocolli di comunicazione dell' Application layer si occupano solo delle interazioni fra processi applicativi che risiedono in stazioni differenti.

I processi che cooperano nell'ambito di una stessa stazione richiedono una comunicazione locale che è supportata dal sistema operativo della stazione stessa.



## Application Layer

- Apre una finestra per accedere ai Servizi dei livelli inferiore .
- Application, Presentation e Session layer possono essere visti come la parte di comunicazione di un Sistema Operativo Distribuito .
- Se il processo utente ha bisogno delle funzionalità offerte dai livelli più alti le utilizza, altrimenti può bypassare uno o più livelli, rinunciando alle loro funzionalità



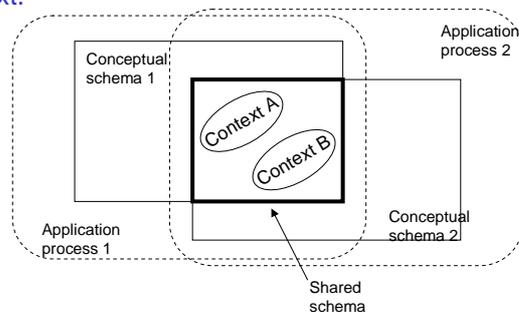
## Comunicazione fra processi

I Processi possono comunicare solo se "l'Universo del Discorso" coincide o si sovrappone.

- L'Universo del discorso (che costituisce l'insieme delle informazioni associate ad un processo) è descritto formalmente da uno **Schema Concettuale** (che definisce tipi e variabili) .

- La porzione condivisa dallo schema concettuale è chiamata "**Shared schema**".

Se uno Shared schema è troppo ampio esso viene suddiviso in più **Application Context**.

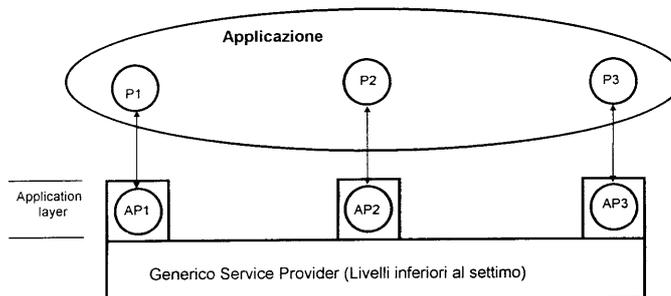


## Application Process (1)

Un Application process (AP) è la rappresentazione astratta di quegli elementi di un sistema reale che, attraverso l'elaborazione di informazioni realizza un'applicazione. Un AP è quindi il componente di un sistema distribuito che può essere univocamente identificato ed indirizzato nell'Application layer .

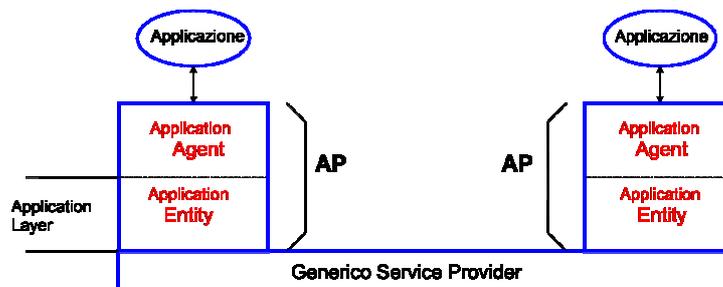
Un AP non coincide con il processo applicativo vero e proprio (che risiede ad un livello superiore all'Application layer) ma è la rappresentazione astratta di alcune delle sue caratteristiche che sono coinvolte nella comunicazione fra i processi reali.

La comunicazione fra processi reali è quindi definita in termini di comunicazione fra AP.

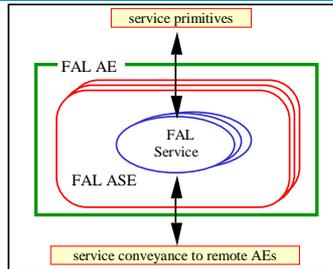


## Application Process (2)

- Un Processo Applicativo può essere suddiviso in due parti :
  - APPLICATION AGENT
  - APPLICATION ENTITY
- L'Application Agent interfaccia il particolare sistema ai servizi applicativi standard
- L'Application Entity esegue le attività di comunicazione, richieste da un Processo Applicativo, indipendenti dal particolare sistema.



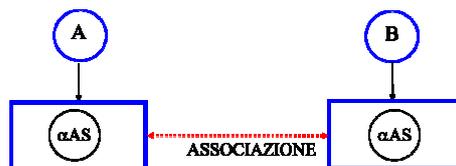
## Application Service Elements



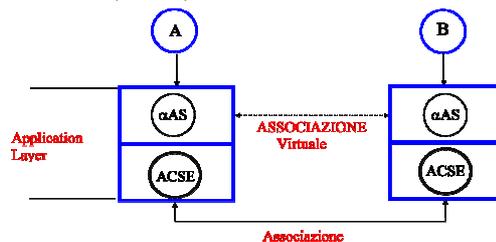
- ASE: è un modulo che realizza una parte delle funzionalità di una Application Entity
- Possono essere divise in due gruppi :
  - gruppo legato alla peculiarità dello standard considerato:
    - Job Transfer & Manipulation
    - File Transfer Access & Manipulation
    - Commitment, Concurrency & Recovery
    - Manufacturing Message Specification (MMS)
  - gruppo necessario per creare un ambiente (definito Associazione) mediante cui vengono forniti i servizi applicativi.

## Application Service Element

Association ASE: ha il compito di creare un ambiente (definito Associazione) mediante cui vengono forniti i servizi applicativi cioè supporta gli altri ASE con dei servizi di comunicazione.



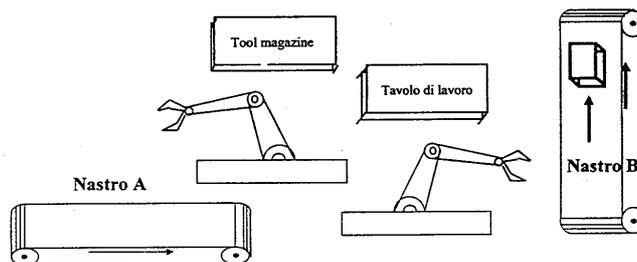
Questa associazione potrebbe anche essere risolta all' interno di ogni singola ASE, ma la soluzione ottimale è quella di avere un' ASE specifica, col compito di creare l' associazione che può essere utilizzare da tutte le altre ASE, chiamata Association Control Service Element (ACSE) .



## Manufacturing Message Specification (MMS)

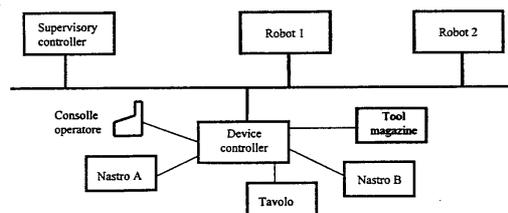
Il Manufacturing Message Specification (MMS) è un Application Service Element (ASE), che abilita la comunicazione fra i dispositivi intelligenti che si trovano nelle applicazioni per controllo di processo.

MMS è parte della specifica del MAP 3.0 ed è sotto revisione presso la Electronic Industry Association (EIA) per la pubblicazione come standard RS-511



L'MMS può essere suddiviso in due aree principali : il **Core Standard** ed il **Companion Standard**.

## Manufacturing Message Specification (MMS)



Il "**Core standard**" fornisce una modellazione estremamente generica, che non rappresenta alcuna classe particolare di dispositivi.

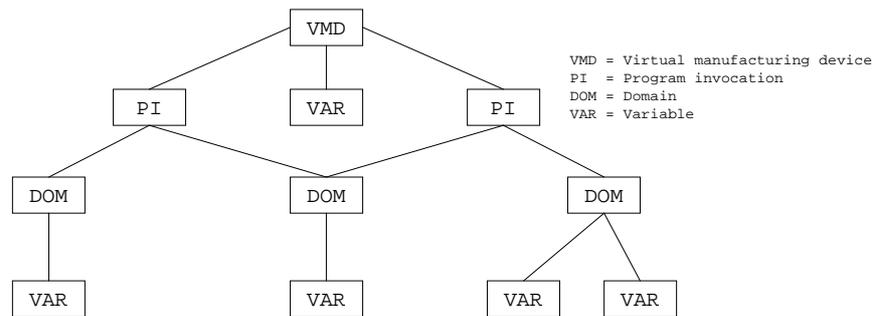
Tutti i dispositivi elettronici disponibili per l'automazione di fabbrica forniscono un certo insieme di servizi comuni.

L'obiettivo dei "**Companion standard**" è soddisfare esigenze specifiche di particolari classi di dispositivi, integrandole nel modello definito dal "core". Ognuno di essi può specificare :

- attributi aggiuntivi per ogni classe di oggetti definita da MMS.
- servizi supplementari.
- estensioni ai servizi già disponibili, specificabili mediante l'aggiunta di dati supplementari che concorrono a formare le PDU.

## Manufacturing Message Specification (MMS)

L'elemento fondamentale del "core" è il **Virtual Manufacturing Device (VMD)** che descrive gli aspetti visibili di un dispositivo. All'interno del contesto definito da VMD, le risorse disponibili sono descritte tramite oggetti quali: Domain, Program invocation, Eventi, Variabili, ecc...



Il **VMD definisce l'immagine standardizzata di un vero dispositivo**, il quale può essere visto e controllato tramite un mezzo di comunicazione. Ciò implica che il VMD contenga solo una descrizione parziale del vero dispositivo fisico sottostante, ed in particolare la parte destinata a risolvere il problema del suo controllo.

## Manufacturing Message Specification (MMS)

L'MMS utilizza un modello di tipo client - server per descrivere i servizi forniti alle applicazioni, con il client che effettua le richieste di servizi ed il server che le esegue

**Variabile access** : sono stati definiti diversi servizi. Fra questi, particolarmente importanti sono i servizi di **Read** e **Write** che permettono ad un client di leggere/scrivere il valore di una o più variabili remote su di un server ed il servizio di **Information Report** che consente ad un processo di spedire informazioni ad un processo pari, senza bisogno di essere interrogato.

Poiché i processi risiedono su stazioni differenti della rete, sono richiesti opportuni meccanismi che permettono di definire e gestire i "data types" delle variabili. A tale scopo sono presenti i servizi :

- Define named type : permette ad un client di definire i tipi di dato su un server.
- Delete named type : cancella il tipo definito precedentemente su un server.
- Get named type attributes : permette ad un client di ottenere gli attributi dei tipi associati a dei nomi simbolici in un server.

## Manufacturing Message Specification (MMS)

- **Message passing** : si usa quando occorre trasferire blocchi di informazione. A seconda della natura e della dimensione del messaggio da passare, il "message passing" fra due processi può prendere sia la forma di un File transfer che di un "Domain transfer". Il Domain è un oggetto astratto che viene impiegato per rappresentare un sottinsieme di risorse del VMD, dedicate ad uno scopo specifico.

I servizi principali sono:

- Initiate download sequence, richiesto da un processo client per chiedere ad un processo Server di creare un dominio e iniziare il suo trasferimento.
- Download segment, usato dal processo Server per richiedere ad un client di trasferire un segmento di informazione.
- Terminate Download sequence, usato da un Server per indicare ad un processo client che la sequenza di download è terminata.

## Manufacturing Message Specification (MMS)

- **Resource sharing** : quando in un sistema esistono risorse condivise, si pone il problema della loro gestione. A questo scopo nell' MMS è disponibile un opportuno set di servizi per la sincronizzazione delle attività di risorse condivise realizzata mediante Semafori che possono essere predefiniti o creati dinamicamente.

I servizi per la gestione dei semafori sono :

- Define semaphore - permette di creare e configurare un nuovo semaforo.
  - Delete semaphore - permette di cancellare un semaforo.
  - Take control - permette di richiedere l' acquisizione del controllo di un semaforo.
- 
- **Program management**: diverse applicazioni richiedono un sistema capace di gestire diversi processi in modo concorrente. I servizi di Program management dell' MMS supportano i "multitasking systems" mediante operazioni che permettono di gestire l' attivazione di programmi remoti attraverso i servizi : *create, start, stop, resume, manage programs* (i.e. processes).

## Manufacturing Message Specification (MMS)

- **Event management** : Il funzionamento corretto dei sistemi automatici richiede il monitoraggio di certe variabili che evidenziano il verificarsi di certi eventi.

Gli eventi possono rappresentare la deviazione da condizioni di sicurezza e devono essere non solo rivelati ma anche trattati in modo opportuno.

I principali servizi per la gestione degli eventi sono : *Define event condition, Delete event condition, Report event condition status, Define event action, Event notification, Acknowledge event notification.*

- **Journal** : i servizi di gestione del Journal forniscono uno strumento per la registrazione ed il recupero di informazioni, ordinate cronologicamente, che riguardano eventi, valori assunti da variabili, e stringhe di testo che possono essere relative a commenti fatti dall' operatore.

I Servizi previsti sono : *Create journal, delete journal, Initiate journal, Report Journal status, Read journal, Read journal.*