

Esigenze dei sistemi di Automazione e loro legame con il sistema di comunicazione.

In un sistema distribuito di Automazione, le prestazioni ottenibili sono strettamente dipendenti dalle caratteristiche del sistema di comunicazione da cui dipende lo scambio informativo fra i vari processi applicativi. Ciò è particolarmente vero quando si scende ai livelli più bassi della gerarchia del controllo, cioè al livello di campo. A tale livello, gli scambi informativi, secondo la filosofia che va ormai affermandosi, sono gestiti dai Fieldbus, le cui caratteristiche influenzano il modo in cui le variabili di misura, controllo ed attuazione sono scambiate.

Va sottolineato come, nel seguito, faremo genericamente riferimento ai sistemi di Automazione in un senso molto lato, comprendendo con tale denominazione non solo i sistemi industriali automatizzati, (in altre parole i sistemi di manufacturing, i sistemi Batch e quelli per il controllo di processi continui) ma anche i sistemi automotive e di Home/Building automation.

Come abbiamo visto, i vantaggi dell'uso del Fieldbus in tali sistemi sono molteplici. I principali possono essere sintetizzati in: riduzione nel numero e costo dei cablaggi, riduzione del costo di installazione, operazione, e manutenzione degli impianti, migliore accuratezza nelle misure (grazie alla trasmissione digitale delle informazioni), possibilità di ottenere informazioni tempestive per la diagnostica dei dispositivi (i FieldBus sono in genere dotati di efficienti meccanismi di fault detection and recovery ed i dispositivi di campo di nuova generazione sono dotati di intelligenza in grado di facilitare la loro diagnostica) e possibilità di condividere informazioni che nei classici sistemi centralizzati sono confinate nei singoli dispositivi di controllo (quali ad esempio i PLC).

L'uso dei Fieldbus introduce la necessità di serializzare tutto il traffico prodotto dai "field devices" e dai sistemi di controllo; ciò comporta l'esigenza di adottare opportuni meccanismi di gestione della larghezza di banda disponibile per la trasmissione, tali da garantire i time-constraints (a volte particolarmente critici) di tutti i processi afferenti al sistema di controllo. Inoltre, per venire incontro alle esigenze trasmissive dei processi tempo-critici che caratterizzano le applicazioni industriali, i sistemi FieldBus prevedono opportuni modelli di comunicazione tra processi. Il termine "larghezza di banda" viene usato, in questo contesto, con il generico significato di quantità di informazione, e quindi di bit, che il sistema di comunicazione può teoricamente trasportare.

Tre parametri possono essere considerati fondamentali per la caratterizzazione del sistema di comunicazione:

- Throughput
- Timeliness
- Affidabilità.

Il **Throughput** misura la quantità di informazione che il sistema di comunicazione è in grado di trasferire. E' un parametro importante a livello di campo, poiché da esso dipende

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

il numero di dispositivi che possono essere connessi ad un singolo Fieldbus. In sistemi di piccola dimensione, con pochi dispositivi di campo da connettere, il *throughput* di un Fieldbus può non rappresentare un problema. Ma se il sistema è complesso, è caratterizzato da molti dispositivi, la possibilità di utilizzare un unico Fieldbus rappresenta un grosso vantaggio poiché introduce una notevole semplificazione nei cablaggi e facilita lo scambio informativo fra i vari processi.

La **Timeliness** misura la tempestività con cui l'informazione viene trasferita fra i vari processi produttori e consumatori. La *timeliness* può essere interpretata in diversi modi, a seconda delle caratteristiche del processo considerato. Ad esempio, in processi non "time critical" essa può essere espressa in termini di tempo di ritardo medio, intendendo che il comportamento del sistema non dipende da una singola variabile, ma dall'insieme di tutte le variabili di sistema. Nel caso di processi "time critical" la *timeliness* può essere espressa in diversi modi. Può rappresentare l'abilità del Fieldbus di consegnare le variabili periodiche di attuazione e controllo (nel caso di sistemi campionati) entro opportune "deadlines" soddisfacendo nel contempo le esigenze (in termini di tempo di ritardo medio) del traffico non periodico, oppure può rappresentare l'abilità a supportare tempestivamente lo scambio informativo di sequenze di dati attivate da particolari eventi (es, allarmi).

La *timeliness* rappresenta un parametro chiave del Fieldbus, per certi versi molto più importante del throughput. Un *throughput* insufficiente può essere recuperato attraverso l'uso contemporaneo di più Fieldbus, mentre una *timeliness* insoddisfacente rende impossibile il corretto funzionamento di un processo e non può essere recuperata.

L'**affidabilità** costituisce un altro parametro molto importante del Fieldbus che, supportando lo scambio delle variabili di misura, attuazione e controllo rappresenta un elemento chiave dell'intero sistema. Va subito detto che l'affidabilità di un sistema di comunicazione è un parametro di difficile valutazione poiché dipende da diversi fattori quali le 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. Per tale motivo, la valutazione dell'*affidabilità* viene realizzata spesso in modo qualitativo, fornendo all'utente informazioni che siano di ausilio nella sua scelta.

Nel seguito, saranno esaminate le esigenze di comunicazione dei vari tipi di traffico che possono esistere in un sistema di controllo di processo e saranno poi discussi e confrontati, alcuni dei più importanti Fieldbus attualmente in uso, con particolare attenzione a quelli che stanno emergendo sul mercato. Il confronto fra tali Fieldbus, mira ad evidenziare le aree applicative più adatte a ciascuno di essi, identificandone i limiti e a fornire utili indicazioni per la realizzazione di sistemi distribuiti per controllo di processo.

2. Requisiti Real-Time in un ambiente di Controllo di Processo distribuito.

Il flusso informativo prodotto dai processi tipici degli ambienti di process control, pur differenziandosi anche fortemente, in relazione alle caratteristiche dello specifico impianto considerato, può essere categorizzato in periodico ed asincrono.

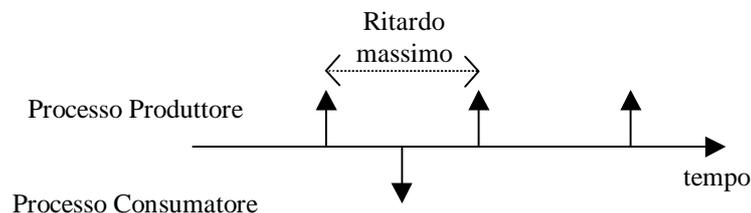
- **Flusso Periodico.** E' generato da processi periodici che eseguono azioni ripetute ad una frequenza costante. Un esempio e' costituito da un processo di campionamento che riceve da un sensore il segnale analogico e che produce un segnale digitale con una frequenza pari a quella di campionamento. Ciascun dato prodotto periodicamente viene consumato da uno o piu' processi consumatori, che sono anch'essi processi periodici. Il

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

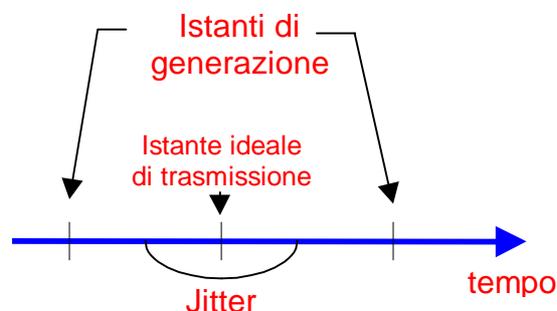
dato è caratterizzato da un suo tempo di vita che corrisponde all'intervallo fra la generazione di due campioni consecutivi, come mostrato in figura.



Un esempio di quanto detto e' rappresentato da un processo consumatore che ricostruisce un segnale digitale trasmesso, per ottenere il segnale analogico originario. Generalmente, ciascun processo consumatore di una variabile aggiornata periodicamente, deve ricevere ciascun valore entro un massimo intervallo ammissibile che è appunto il tempo di vita.



La trasmissione dei dati periodici, dovrebbe in teoria avvenire ad istanti fissi all'interno del tempo di vita; tuttavia l'istante reale di trasmissione è sempre collocato in un intorno dell'istante ideale, noto come jitter. Sempre riferendosi al processo consumatore che ricostruisce il segnale analogico originario, il jitter massimo ammissibile e' rappresentato dall'intervallo di campionamento.



Poiche' il flusso informativo periodico e' legato a processi la cui dinamica e' sempre conosciuta, il periodo di produzione, quello di consumo ed il jitter sono valori noti a priori.

- **Flusso Asincrono.** E' legato a processi che evolvono nel tempo, in un modo che non può essere previsto a priori. L'aggettivo asincrono viene utilizzato allo scopo di mettere in evidenza l'assoluta indipendenza tra l'attività sul mezzo di comunicazione (scandite da un clock di sistema) e quelle dei processi produttori di tale traffico. In un sistema di controllo di processi industriali, vi sono numerosi esempi di flusso asincrono; basti pensare agli allarmi generati da un sistema di supervisione, ad operazioni di *download* di configurazioni di *controller* intelligenti, o ad interrogazioni di banche dati centralizzate o distribuite. Nell'ambito del flusso di informazioni asincrone, e' necessario distinguere il flusso informativo in base ai vincoli temporali che lo caratterizzano :

- 1) informazioni che sono prodotte in maniera casuale e che non sono caratterizzate da vincoli temporali particolarmente stringenti. Esempi di tali informazioni sono interrogazioni ad una banca dati o *file transfer*. In tal caso non vi e' alcun vincolo temporale

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

particolarmente stringente sulla consegna di tali informazioni. In altre parole non esistono particolari esigenze sulla durata del *delivery time* inteso come l'intervallo temporale tra la produzione dell'informazione e la sua consegna al o ai processi consumatori.

2) Informazione prodotta una sola volta. L'importanza che tale informazione riveste nell'ambito dell'intero sistema di controllo, comporta la necessita' di ridurre il piu' possibile il *delivery time*: l'informazione prodotta deve essere consegnata quasi istantaneamente a partire dalla sua trasmissione. Si pensi ad esempio ad un allarme che viene generato una sola volta e che deve essere ricevuto immediatamente dal sistema di supervisione dell'impianto. La gestione di tali informazioni asincrone e' piuttosto critica quando, come ad esempio nel caso di allarmi, non e' possibile conoscere a priori l'istante di tempo in cui il dato è prodotto.

3) Informazioni la cui produzione inizia in istanti non prevedibili a priori, e poi si ripete nel tempo in modo periodico. Si pensi ad esempio ad un sistema di controllo che inizia a ricevere informazioni da un *encoder* solo quando esso arriva a fine corsa. L'istante in cui l'encoder arriva a fine corsa non e' prevedibile, ma da quel momento in poi vi sara' un flusso informativo periodico tra l'encoder ed il sistema di controllo. Anche in questo caso la criticita' dell'informazione prodotta richiede un ritardo nella trasmissione dell'informazione nullo od in ogni caso molto breve. Le uniche informazioni note sono solo il periodo ed il massimo jitter ammissibile nella generazione del flusso informativo, ma non il suo istante d'inizio. Dunque, anche in tal caso, la gestione di tale flusso informativo e' particolarmente critica a causa della difficolta' di conoscere a priori l'istante d'inizio.

Va precisato come, in alcuni casi, i vincoli temporali possono essere talmente stringenti, o i dati trasmessi essere costituiti da uno o pochi bit, da rendere impraticabile l'utilizzo dei Fieldbus. E' questo ad esempio il caso in cui un encoder, collegato ad un motore in corrente continua, genera impulsi da trasmettere al microcontrollore del motore. Tali impulsi possono essere codificati con un solo bit, e vanno trasmessi ad una frequenza così elevata (anche ad intervalli di qualche microsecondo) da rendere inapplicabile l'uso di un Fieldbus. In tali casi, la soluzione più applicabile è la connessione diretta dell'encoder al microcontrollore.

Allo scopo di sintetizzare i requisiti che un Fieldbus deve possedere per soddisfare i vincoli dei vari tipi di processo è utile riportare i dati presentati in una analisi realizzata dal CENELEC nel 1991 [ref. CLC/BT/WG 62-6 n°1]. In tale studio vengono esaminati i due principali tipi di ambienti operativi in cui è possibile utilizzare i Fieldbus, e precisamente quello relativo a "Process Control Industry" e quello relativo "Discrete Piece Manufacturing". Per ciascuno dei due ambienti vengono esaminati 3 casi di cui sono analizzate le caratteristiche e le esigenze di comunicazione.

2.1. Process Control Industry

2.1.1 FieldBus nella sezione di controllo.

Questo caso è stato sviluppato dall'UTE e analizzato dall'ISA-SP50 insieme alla "Discrete Piece Manufacturing Industry" tenendo in considerazione Fieldbus conformi alle specifiche H2 (le specifiche H1 ed H2 dei fieldbus sono state definite dall'ISA e fanno riferimento alla diversa criticità dei sistemi in termini di caratteristiche di comunicazione). Il

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

Fieldbus viene utilizzato per supportare una intera sezione del processo, che impiega almeno 200 sensori/attuatori connessi ad una singola sezione di Fieldbus.

Le ipotesi valide per la valutazione del flusso di dati sono:

- 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.
- In media, due utenti per ogni singolo dato, con una distanza di 1000 metri.
- Necessità di alimentazione attraverso il Bus.
- Possibilità di una versione in Sicurezza intrinseca.

Questo scenario può essere applicato nella strumentazione di processi standard. Nel caso di macchine rotative (quali ad esempio compressori, pompe, turbine, ecc.) vanno considerati dei requisiti addizionali. In tal caso, la frequenza di aggiornamento va modificata in :

- 20 msec per gli I/O analogici
- 10 msec per gli I/O on-off.

La quantità di strumenti che si utilizza viene ridotta ad un massimo di 40 I/O analogici e 10 dispositivi di I/O che multiplexano 160 sensori/attuatori on-off con una lunghezza ridotta di 500 metri.

2.1.2 FieldBus per il controllo nell'industria di processo.

Questo caso è stato 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. E' stato verificato, come, l'80% di tutti i dispositivi usati in una applicazione possano essere interconnessi mediante Fieldbus. Ci si aspetta che siano supportate le seguenti caratteristiche:

- 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.

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

- 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.
- Deve essere possibile implementare funzioni di servizio (quali ad esempio quelle relative alla configurazione, monitoraggio, comando, messa a punto, ecc.) attraverso tool che non siano sviluppati da uno specifico fornitore. Tali tools possono essere attaccati permanentemente (o no) al bus.
- La dimensione ed il tipo dei dati trasportati attraverso il Fieldbus deve essere definita dall'utente e non deve dipendere da uno standard. Tali caratteristiche devono essere configurate attraverso il network management, prima di entrare in modalità operative.

2.1.3 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.

2.2. Discrete Piece Manufacturing

In questo tipo di ambiente operativo, i processi da controllare sono molto più veloci rispetto al caso del controllo di processo, per cui sono richieste prestazioni più spinte dal parte del sistema di comunicazione.

2.2.1 FieldBus nella costruzione di componenti elettronici.

I seguenti requisiti sono stati sviluppati da NEMA.

- 32 dispositivi connessi al bus
- ogni messaggio dell'utente contiene da 8 a 64 bit di informazione.
- Periodo di aggiornamento dei dati (scan period) 8 ms.

2.2.2 Fieldbus per il controllo di motori elettrici in industrie pesanti.

Questo caso è stato elaborato con riferimento alle acciaierie, dove vengono utilizzati motori di grande potenza per il trasporto del materiale grezzo o semilavorato. Il Fieldbus è utilizzato per trasferire informazioni ai controllori (drives) dei motori. A causa delle differenti modalità operative presenti in uno stesso ambiente, è prevista la coesistenza di tre diversi flussi informativi su uno stesso link di comunicazione, e precisamente:

- Primo flusso di messaggi:
 - ciclo di aggiornamento dei dati, di 20 ms
 - ogni messaggio utente contiene 48 bit
 - ogni dispositivo invia/riceve 3 messaggi (uno dei quali in broadcast)
- Secondo flusso di messaggi:
 - ciclo di aggiornamento dei dati, di 200ms
 - ogni messaggio utente contiene 80 bit
 - ogni dispositivo invia/riceve 2 messaggi
- Terzo flusso di messaggi:
 - non esistono specifici vincoli temporali
 - ogni messaggio utente contiene fino a 1Kbytes
 - ogni dispositivo invia/riceve 1 messaggio.

2.2.3 Fieldbus nelle celle di lavorazione dell'industria automobilistica

In questo tipo di ambiente operano in genere diversi PLC per il controllo di utensili e dispositivi di movimentazione, con diverse tipologie di flussi di informazioni:

- Primo tipo di flusso di informazioni:
 - ciclo di aggiornamento dei dati, di 10 ms
 - ogni messaggio utente contiene 40 bit
 - sono previsti fino a 32 dispositivi
- Secondo tipo di flusso di informazioni:
 - ciclo di aggiornamento dei dati, di 100ms
 - ogni messaggio utente contiene 40bits per il multiplexing di sensori ed attuatori di tipo on/off
 - sono previsti fino a 32 dispositivi
- Terzo tipo di flusso di informazioni:
 - ciclo di aggiornamento dei dati, di 100ms
 - ogni messaggio utente contiene 160 bits per il multiplexing di sensori ed attuatori analogici.
 - sono previsti fino a 55 dispositivi.

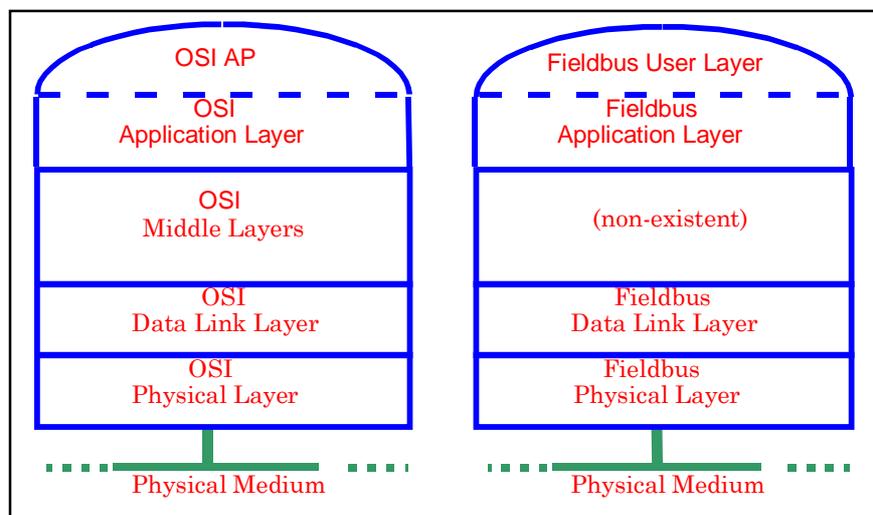
3. Alcune considerazioni sui punti chiave da considerare nell'architettura dei Fieldbus, per il loro confronto.

Ricordiamo, che il modello di riferimento ISO-OSI per i sistemi di comunicazione prevede una architettura stratificata, con 7 livelli funzionali, ciascuno dei quali è responsabile per la fornitura di specifici servizi ai livelli superiori e quindi in definitiva ai processi utente.

Ricordiamo anche che, a dispetto di quanto specificato nell'architettura standard, oggi è universalmente accettato il fatto che si deroghi dal modello standard a 7 livelli per diverse ragioni fra le quali ad esempio:

- non tutte le funzionalità supportate 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 abbastanza diversificate (così come molto diversi sono i protocolli e servizi forniti). In generale. La maggior parte dei Fieldbus utilizza una architettura a tre soli livelli



Tre livelli sono infatti il numero minimo indispensabile per la realizzazione di un sistema di comunicazione completo. In effetti nel passato sono stati presentati alcuni Fieldbus per i quali erano stati specificati e standardizzati solo i primi due livelli, ma si è trattato di soluzioni incomplete, nelle quali era poi lasciato all'utente il compito di implementare le funzioni previste nell'Application layer. E' stato questo, per esempio, il caso del CAN, anche per il quale tuttavia recentemente è stato specificato e standardizzato un opportuno Application layer.

Come noto i tre livelli presentati nella figura e cioè Physical, Data link ed Application forniscono tre differenti ed indispensabili funzionalità:

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

- 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, ecc.
- Il **Data link layer** è responsabile della gestione dell'accesso al mezzo fisico e dello scambio informativo fra i nodi interconnessi.
- L'**Application layer** deve mettere a disposizione una serie di funzionalità che consentono ai processi utente, un facile scambio informativo.

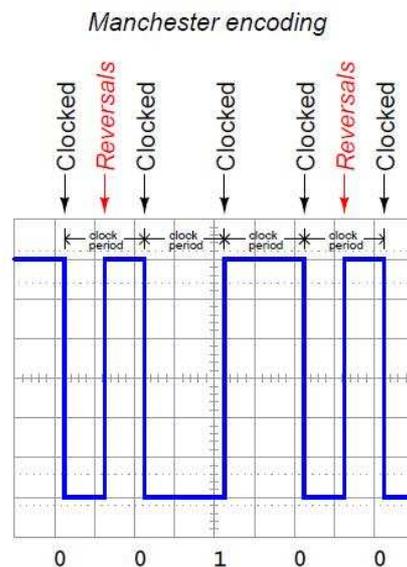
Ai livelli citati si aggiunge un livello trasversale (non rappresentato in figura) chiamato network management, con la funzione di gestire la configurazione dei nodi, il recupero degli errori, il controllo dei processi trasmissivi e l'accesso alle informazioni.

3.1 caratteristiche del Physical layer

Il physical layer rappresenta il substrato su cui si basa il funzionamento del Fieldbus. Esso deve mettere a disposizione un bit rate il più elevato possibile, usare codifiche efficaci e consentire topologie flessibili.

Soffermiamoci su due aspetti: codifica e layout della rete.

Una delle codifiche più usate è la codifica Manchester che si caratterizza per il fatto di essere trasparente (rispetto ai dati) e self clocking. Diversi tipi di fieldbus (Fieldbus



foundation, WorldFip) usano questa codifica.

EIA RS-485, equivalente allo standard Europeo CCITT V11, è 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: la differenza tra la tensione presente sui due fili costituisce il dato in transito. 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 per un'operazione valida, ma qualsiasi tensione compresa tra +12 V e -7 V permette il corretto funzionamento del ricevitore.

La EIA RS-485 specifica soltanto le caratteristiche elettriche del trasmettitore e del ricevitore. Non indica né raccomanda alcun [protocollo](#) per la trasmissione dei dati

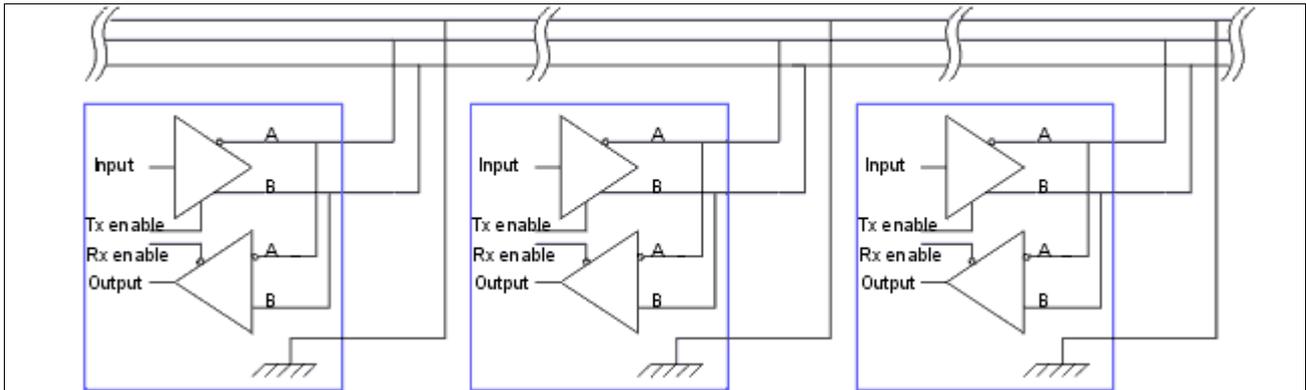
Supporta le linee multi-drop, cioè linee in cui coesistono più ricevitori e trasmettitori sulla stessa coppia di fili. Al fine di evitare conflitti è ovviamente necessario che un solo

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

trasmettitore alla volta sia attivo. Questo implica l'uso di trasmettitori che, oltre alle uscite corrispondenti allo zero e all'uno, possano gestire anche un "terzo stato" in cui l'elettronica appare come fisicamente non collegata alla linea (stato detto ad alta impedenza, three-state o Hi-Z).

I ricevitori possono invece essere tutti attivi contemporaneamente ed in genere lo sono effettivamente.

La topologia più usata con questo standard è quella a due fili (oltre alla massa) rappresentata nello schema seguente. Questa connessione permette la trasmissione bidirezionale (ma ovviamente non contemporanea) tra due o più nodi che, dal punto di vista elettrico, sono tra loro equivalenti.

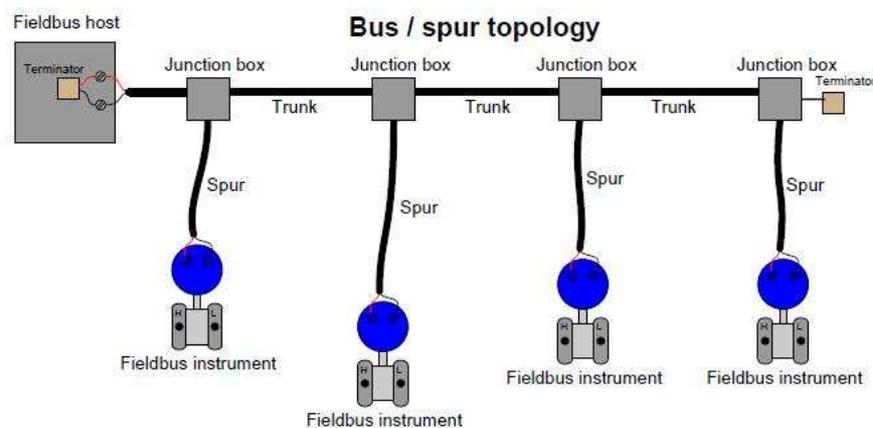


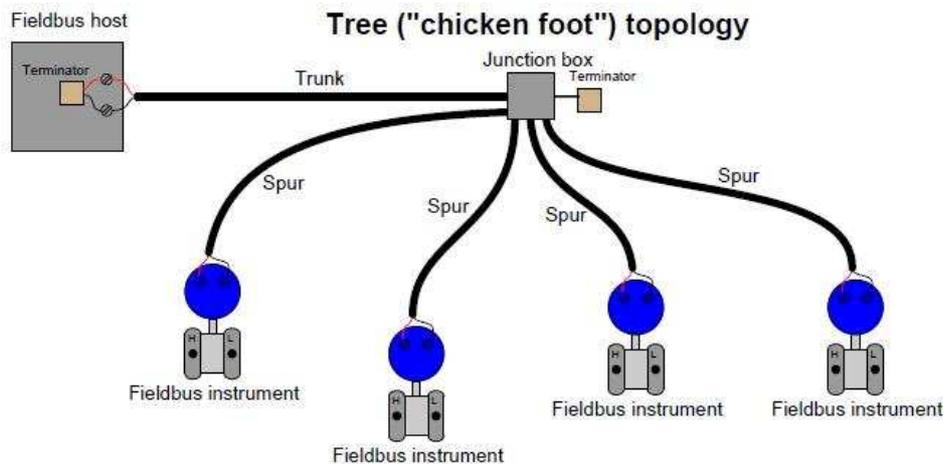
Lo standard originario permette la connessione di massimo 32 ricevitori ma utilizzando integrati a basso assorbimento tale limite può essere abbondantemente superato.

Dal punto di vista delle topologie, quelle più usate sono a Bus o a stella.

La topologia a bus prevede un certo numero di dispositivi interconnessi mediante delle Junction box.

La topologia a stella, anche definita a "zampa di gallina" prevede un certo numero di dispositivi collegati alla stessa Junction box.





3.2 Caratteristiche del Data Link layer.

Dei tre livelli citati, (ferma restando l'importanza del ruolo svolto dagli altri due), quello più caratterizzante un Fieldbus è il Data Link layer dal quale principalmente dipendono le caratteristiche del sistema in termini di Throughput, Timeliness e Reliability.

Infatti, è nel Data Link layer che vengono implementate le strategie di gestione del mezzo fisico che permettono l'allocazione di porzioni di banda ai singoli processi applicativi ed il rispetto di specifici vincoli temporali. Per questa ragione nel seguito, il confronto fra i vari tipi di Fieldbus sarà centrato sulle caratteristiche dei vari protocolli di Data Link layer che essi utilizzano.

Nell'analizzare le caratteristiche del DLL, occorre ricordare che il problema principale che esso deve risolvere, deriva dall'uso di un canale trasmissivo comune in sostituzione dei collegamenti punto - punto dei sistemi convenzionali. Occorre quindi implementare opportune strategie che permettano il contemporaneo soddisfacimento dei tre requisiti base: Throughput, Timeliness, Reliability.

3.2.1 Influenza del DLL sul Throughput

Il Throughput dipende essenzialmente da due caratteristiche: uso di elevati Bit rate nel Physical layer ed utilizzo efficiente della banda messa a disposizione dal Physical layer. Se è ovvio che valori elevati di Bit rate permettono di ottenere throughput elevati è pur vero che ciò ha un certo costo. L'aumento del bit rate richiede hardware più costoso, l'uso di cavi di migliore qualità, aumenta la sensibilità ai disturbi e spesso limita la massima lunghezza che può essere coperta dal bus. E' perciò molto importante curare anche l'altro aspetto e cioè l'uso efficiente della banda che 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 (in genere si tratta di un preambolo, start/end delimiter, campo di controllo, source/destination address, FCS) ha una notevole importanza sulla efficienza trasmissiva poichè, la trasmissione, anche di un solo bit di dati, richiede in ogni caso la presenza di una frame completa, con tutti i suoi campi di supporto. L'efficienza è tanto migliore quanto più è elevato il rapporto dati/campi di supporto. Da questo punto di vista va rimarcato come in generale tutti i protocolli di DLL dei vari Fieldbus esistenti, siano comparabili. Va segnalato il CAN, che ha una struttura di frame un pò più efficiente delle altre e soprattutto l'Interbus-S che ha invece una struttura di frame molto più efficiente rispetto a quella utilizzata negli altri Fieldbus.

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

•La strategia di accesso al mezzo fisico rappresenta un altro aspetto molto importante dal punto di vista dell'ottimizzazione dell'efficienza complessiva del sistema. Da questo punto di vista gli approcci adottati sono essenzialmente tre:

1. **Gestione centralizzata dell'accesso.** Tale strategia, adottata nel Fieldbus standard IEC 1158 e nel Fieldbus WorldFip, prevede una stazione particolare, con funzioni di controllore dell'accesso che distribuisce le autorizzazioni alle varie stazioni del bus. Tale stazione autorizza ogni singola trasmissione da parte delle altre stazioni, mediante l'invio di uno specifico messaggio (una specie di token da utilizzare per una sola trasmissione). Tale approccio non è efficiente per il throughput poichè ogni frame utile deve essere preceduta da una frame di servizio (il token).

2. **Gestione distribuita dell'accesso.** Tale strategia, adottata nel Profibus, prevede la circolazione di un token che viene periodicamente acquisito dalle varie stazioni Master, le quali possono utilizzarlo per un definito intervallo di tempo. In tal caso, la perdita di efficienza è limitata poichè i token sono in numero molto minore rispetto alle frames utili (cioè quelle che trasportano i dati).

3. **Gestione distribuita dell'accesso, basata su meccanismi di tipo Carrier Sense Multiple Access CSMA.** Tale strategia (utilizzata dalla rete Ethernet e dal CAN) può dimostrarsi più efficiente delle prime due, poichè non prevede l'uso di frame di servizio (cioè, token) e quindi permette di utilizzare tutta la larghezza di banda per la trasmissione di frames contenenti dati. E' tuttavia condizionata dal problema delle collisioni che possono introdurre delle considerevoli perdite di larghezza di banda (ciò non è vero per il CAN che grazie al particolare meccanismo di arbitraggio utilizzato non spreca banda nelle collisioni).

4. Anche se abbiamo parlato di tre approcci, per l'accesso al mezzo fisico, vale la pena di citare anche un quarto approccio, meno noto, utilizzato dall'Interbus-S. Questo approccio, chiamato **Single-frame**, permette di utilizzare una sola frame per tutte le stazioni, limitando in maniera massiccia l'overhead costituito dalle informazioni di supporto presenti nelle frames (i vari delimitatori, i campi di controllo, indirizzo, FCS, ecc.). Per tale ragione, l'efficienza trasmissiva ottenibile è notevolmente migliore di quella offerta dagli altri approcci. Unico neo, è il fatto che tale approccio funziona su un ring e non su un bus.

3.2.2 Influenza del DLL sulla Timeliness.

La Timeliness dipende da diverse caratteristiche del sistema. Innanzi tutto va ricordato, come abbiamo spiegato nella prima parte di quest'analisi, che esistono diversi tipi di timeliness, legati alle diverse esigenze dei processi. Tuttavia, possiamo genericamente valutare la timeliness, come la capacità del sistema di comunicazione, di consegnare al consumatore, un dato, col minor ritardo possibile dalla sua produzione. Sappiamo che un dato periodico, per esempio, va consegnato prima della produzione di un nuovo valore, mentre un dato non periodico va consegnato, in generale, il più presto possibile. Il DLL, condiziona fortemente il comportamento di un Fieldbus da questo punto di vista. Confrontiamo quindi, i differenti approcci utilizzati nei vari DLL.

•**Gestione centralizzata dell'accesso:** la gestione centralizzata del DLL affida ad una particolare stazione (Il Link Active Scheduler, LAS nel Fieldbus standard IEC 1158 ed il Bus Arbitrator, BA nel Fieldbus WorldFip) la responsabilità per l'accesso al mezzo fisico. Da questo punto di vista, la gestione è diversa per il traffico periodico e per quello asincrono:

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

- nel caso di traffico periodico è necessario fare uso di una tabella di schedulazione (che può essere prodotta off-line e caricata in fase di configurazione del sistema, o può essere generata on-line durante il funzionamento) per decidere quando autorizzare la trasmissione di un singolo dato, periodico. La generazione della tabella di schedulazione può rappresentare un'operazione complessa quando bisogna schedulare molte variabili con dinamica diversa, poichè occorre trovare, per ciascuna di esse, un intervallo di trasmissione compatibile con i suoi vincoli temporali. Attraverso la tabella di schedulazione, il master della trasmissione (il LAS o il BA) spedisce le autorizzazioni per trasmettere, alle varie stazioni. Se la tabella di schedulazione è costruita correttamente, questo approccio è ottimo per quanto concerne la timeliness.

- nel caso di traffico asincrono, il problema che sorge è legato al fatto che esso non può essere schedulato, per definizione. Il LAS/BA pertanto non è in grado di autorizzarlo in maniera semplice. Gli approcci usati per risolvere il problema possono essere diversi. Nel Fieldbus IEC, per esempio, il LAS fa circolare, negli intervalli di tempo liberi nella sua tabella di schedulazione, un token che può essere utilizzato dalle stazioni, per un certo intervallo di tempo e poi restituito. In tal modo, le stazioni che hanno del traffico asincrono, hanno periodicamente la possibilità di trasmettere. Nel Fieldbus WorldFIP, invece, il BA colleziona esplicitamente le richieste asincrone da parte delle varie stazioni, collegandole alle autorizzazioni periodiche che invia per il traffico ciclico. Dopo di che, quando trova nella propria tabella di schedulazione uno spazio libero, invia una esplicita autorizzazione a trasmettere.

• **Gestione distribuita dell'accesso:** l'accesso al mezzo fisico avviene mediante il possesso di un token. Ciò mentre consente alle varie stazioni di acquisire periodicamente il diritto a trasmettere e quindi garantisce una timeliness media, ha diverse implicazioni sui vari tipi di traffico.

- 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 fra le varie stazioni. Ciò garantisce che dopo un certo tempo (che al max può essere 2 volte TTRT) ogni stazione acquisirà il token e avrà diritto a trasmettere. IL TTRT stabilisce quindi la velocità a cui il token ruota fra le varie stazioni. Questo approccio possiede tre limitazioni. La prima è relativa al fatto che una stazione acquisisce il token con una periodicità costante come valor medio. Ciò significa che l'istante di trasmissione di un dato non è certo, ma è variabile nell'intorno di un valor medio. La trasmissione è quindi affetta da un jitter che può anche avere un valore considerevole. La seconda considerazione è legata al fatto che, quando sono presenti dati periodici con periodi differenti, il TTRT dovrà essere fissato in relazione al dato più critico ed il token dovrà ruotare alla velocità max fra le varie richieste dai dati. Ciò può introdurre ulteriori Jitter, se i periodi dei dati non sono multipli del periodo più breve, e causa sprechi di larghezza di banda poichè il token verrà fatto circolare a vuoto, fra le stazioni che producono traffico con il periodo più breve. La terza limitazione, infine, è legata al fatto che 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, poichè la rotazione del token garantisce ad ogni stazione che, dopo un certo tempo, acquisirà l'accesso al mezzo fisico. Quindi, l'eventuale traffico asincrono prodotto, potrà essere servito ogni volta che si acquisisce un token (dopo aver servito il traffico ciclico che di solito ha una priorità maggiore).

•**Gestione distribuita basata su protocolli di tipo CSMA:** ogni stazione tenta di accedere al mezzo fisico, non appena questo appare libero. Se due stazioni tentano contemporaneamente, si ha una collisione. La collisione rappresenta il vero (ed unico problema) di questo approccio, che per altri versi si comporta in genere molto bene. L'approccio CSMA è usato come detto in due protocolli di rete: nella Ethernet e nel CAN, con alcune differenze che assumono grande rilevanza nel contesto del controllo di processo.

Nella rete Ethernet, la collisione distrugge i messaggi coinvolti, che dovranno essere rischedulati (dalla stazione sorgente) per una nuova trasmissione dopo un ritardo calcolato in modo random. Non è possibile distinguere fra traffico periodico o asincrono (se non all'interno della stazione); entrambi verranno trattati allo stesso modo. Per tali motivi, il comportamento del protocollo non è deterministico ed i ritardi introdotti nella trasmissione delle informazioni, possono in teoria essere indefiniti. Questo comportamento, ha impedito per lungo tempo l'uso del protocollo Ethernet, come fieldbus, nell'ambiente dell'automazione industriale. Va però detto, che oggi il protocollo Ethernet è stato rivalutato (soprattutto a causa della sua enorme diffusione in tutti gli ambienti applicativi) e i suoi limiti, per quanto riguarda il ritardo introdotto, sono stati ridimensionati. Infatti, definendo opportunamente l'ambiente operativo (numero di stazioni, workloads, uso di Hubs e Switches, ecc.) esistono un gran numero di applicazioni di Automazione/controllo di processo in cui le prestazioni fornite da Ethernet sono perfettamente accettabili.

Nel CAN, invece, la collisione non distrugge tutti i messaggi coinvolti ma fa sopravvivere quello a più alta priorità. Questo comportamento, unito all'esistenza di una priorità dei messaggi, permette di introdurre una sorta di schedulazione automatica on-line, che permette di soddisfare sia le esigenze del traffico ciclico che di quello asincrono. Anzi, assegnando ad esempio a messaggi di allarme (che sono asincroni, ma critici) le più alte priorità, si permette a questi di essere trasmessi subito, scavalcando perfino il traffico ciclico. Con una buona scelta delle priorità da assegnare ai messaggi, è possibile soddisfare senza problemi, sia le esigenze del traffico ciclico, che di quello asincrono. L'unico limite del CAN (che può essere insuperabile) è la limitata lunghezza del Bus che, ad 1Mbit/sec, è al max di 40metri.

•**Gestione centralizzata basata su Single frame.** Come accennato, questo approccio è utilizzato dall'Interbus-S e 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. Per quanto riguarda il traffico asincrono, questo fa uso di fette di banda rese disponibili allo scopo, e viene servito abbastanza bene. Va detto che 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.

3.2.3 Influenza del DLL sull'affidabilità.

Anche l'affidabilità delle reti è fortemente influenzata sia dai protocolli del DLL che dalle specifiche strategie di recupero adottate dalle stazioni. In questo paragrafo, verranno discusse le implicazioni generali dei vari approcci di gestione 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; per tale ragione, è sempre presente nel sistema un altro LAS/BA in stato di Stand-by,

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

che entrerà in funzione dopo il guasto di quello attivo. La sostituzione del master comporta sia un certo periodo di inattività della rete, sia il rischio che il nuovo LAS/BA non posseda una tabella di schedulazione del traffico, aggiornata.

- **Gestione distribuita dell'accesso:** poichè non esiste un elemento con maggiori responsabilità degli altri, non esiste un unico punto di vulnerabilità del sistema. Per tale motivo, il guasto di una stazione non può di norma bloccare la rete, anche se determina un periodo di inattività legato alla necessità di riconfigurare il sistema. La riconfigurazione si rende necessaria per riorganizzare il ring virtuale, necessario per il passaggio del token fra le varie stazioni, quando una stazione si guasta.

- **Gestione distribuita basata su protocolli di tipo CSMA:** in questo caso, non solo la gestione dell'accesso al mezzo fisico è distribuita fra le varie stazioni, ma non esiste un token da passare. Per tale motivo, il guasto di una stazione ha una influenza locale, e non determina malfunzionamenti sull'intero sistema. Non esiste pertanto la necessità di riconfigurare alcunchè, ed il sistema sopravvive senza problemi.

- **Gestione centralizzata basata su Single frame:** in questo caso, il sistema è centralizzato e non prevede alcuna ridondanza (che d'altronde sarebbe difficile implementare poichè il sistema utilizza una struttura Hw a livello fisico, sotto il diretto controllo del Master). In ogni caso, in questo tipo di sistema, il master contiene anche gli elementi di controllo del processo, per cui un guasto della stazione master, produce inevitabilmente l'arresto dell'intero sistema di controllo.

3.3 Application layer

3.3.1. Generalità

L'Application layer rappresenta l'ultimo livello della pila OSI; a differenza degli altri livelli i suoi utenti non sono Entità ma Processi Applicativi. Esso è quindi particolarmente importante perché consente ai processi applicativi (End users) di accedere a tutte le funzioni della rete. Sebbene sia possibile per un processo applicativo accedere direttamente ai servizi offerti dai livelli più bassi (dal Presentation al Data link layer), le funzionalità offerte dall'Application layer semplificano notevolmente le problematiche legate alla cooperazione fra processi richiesta per realizzare un'applicazione distribuita.

Il comportamento dei processi cooperanti è dinamico in quanto essi possono iniziare o terminare le loro interazioni in qualsiasi istante. Poichè l'interazione fra processi applicativi avviene attraverso lo scambio di messaggi, tale interazione corrisponde ad una comunicazione fra processi.

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.

Si consideri ad esempio il sistema raffigurato nella pagina seguente, Fig. 3.3.1 che mostra un recipiente del quale si voglia controllare il livello di un liquido in presenza di un flusso in ingresso e di uno in uscita. Il flusso d'uscita può essere variato per mezzo di una valvola di controllo. La regolazione è basata su un controllo retroazionato.

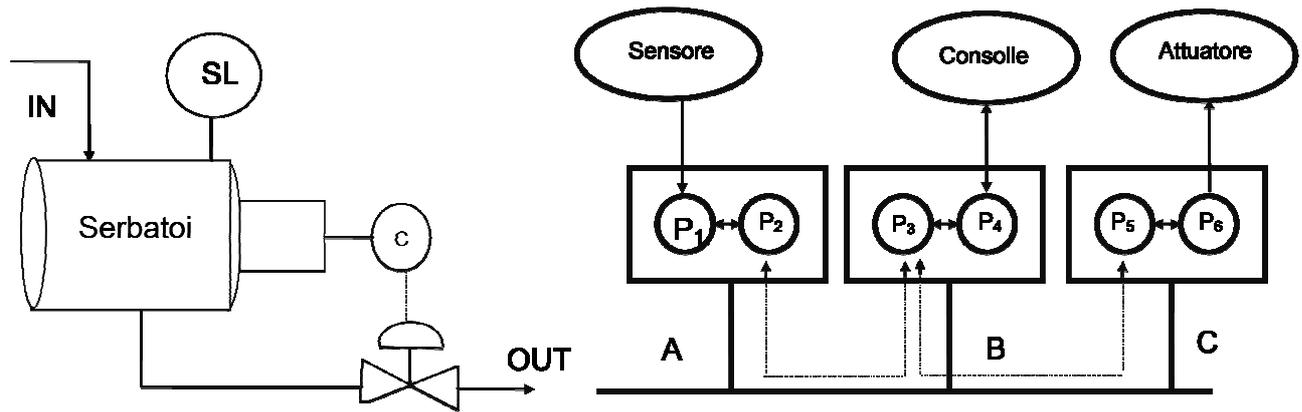


Fig.3.3.1 un serbatoio ed il suo controllo mediante un sistema distribuito

Come si vede dalla fig. 3.3.1, il sistema di controllo è realizzato mediante tre nodi:

Il nodo A contiene due processi:

- P1 che acquisisce i dati del sensore di livello SL.
- P2 che elabora i valori forniti da P1 e li trasferisce a P3

Il nodo B contiene due processi:

- P3 che realizza l'algoritmo di regolazione
- P4 che gestisce la comunicazione con la console dell'operatore (locale).

Il nodo C contiene due processi:

- P5 che riceve i comandi prodotti da P3
- P6 che trasferisce i dati alla valvola di regolazione.

Appare chiaro come, mentre i processi P1, P4, P6 esauriscono le loro attività localmente, P2, P3, P5 richiedono per la loro cooperazione il supporto del sistema di comunicazione. E' proprio la presenza dell'Application layer che consente ai processi applicativi di ragionare in termini di variabili da scambiare piuttosto che di indirizzi dei nodi.

In un sistema distribuito Application, Presentation e Session layer (questi due ultimi livelli come si è già detto sono ormai assenti nei sistemi attuali) nel loro complesso possono essere visti come quella parte di un Sistema Operativo Distribuito (SOD) che risolve le problematiche di comunicazione). Le funzionalità di comunicazione sono indispensabili in quanto un SOD deve supportare la cooperazione tra processi remoti. Possiamo pensare che il nostro processo applicativo utilizzi la struttura formata dai livelli OSI di cui il nucleo è rappresentato dall'interfaccia offerta dal Transport layer.

Tale interfaccia rappresenta il livello più basso dell'architettura, attraverso cui si accede direttamente ai servizi di trasporto dati. Il processo applicativo (l'applicazione) può in questo sistema operativo distribuito, utilizzare i vari servizi forniti dai livelli Session, Presentation e Application, o può accedere direttamente al transport layer.

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. Come si vede dalla fig.3.3.2., l'applicazione può accedere direttamente al Transport layer, può accedervi

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

transitando per il Session layer, Presentation layer ed Application layer, utilizzando le funzionalità di tutti e tre i livelli più alti dell'OSI.RM.

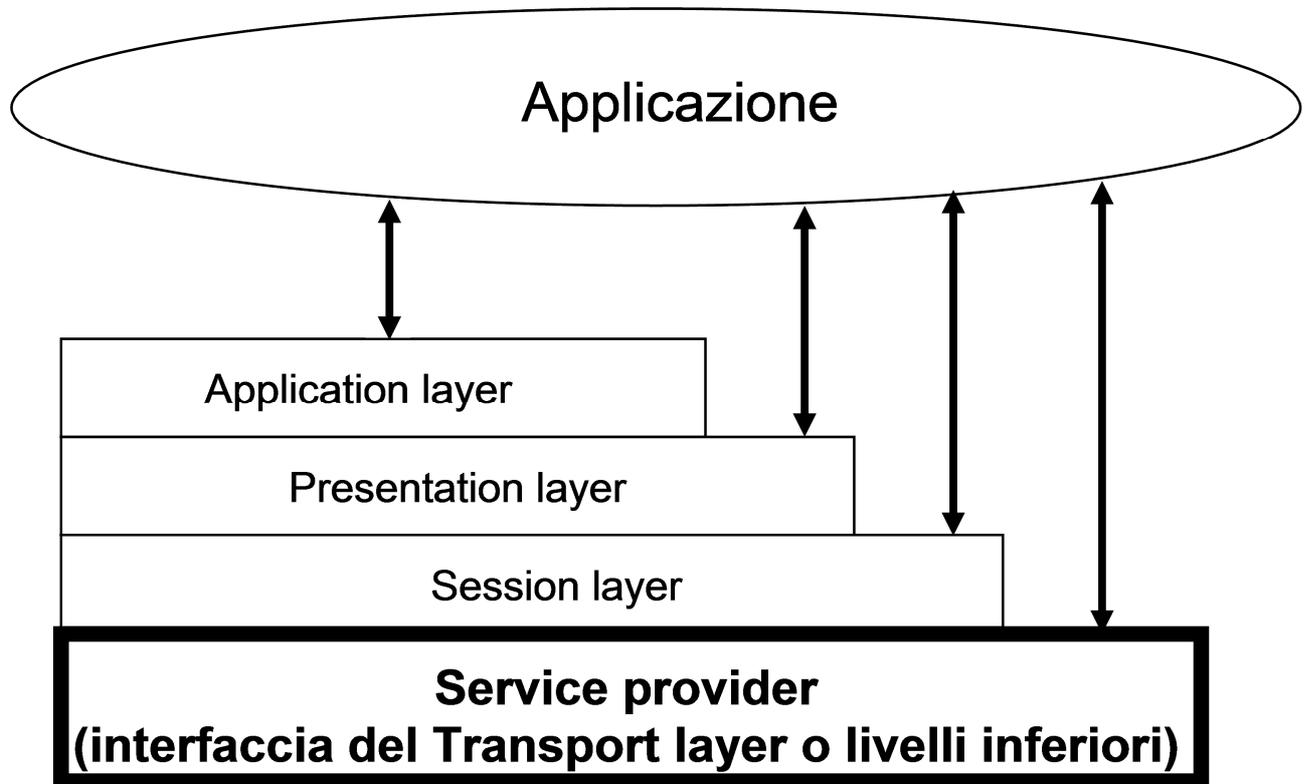


Fig.3.3.2.: Architettura a livelli del Sistema operativo distribuito. Le frecce mostrano come l'applicazione possa raggiungere il **Service provider** attraversando uno o più layers.

Va però rimarcato come nei livelli più bassi (campo) delle applicazioni di automazione, caratterizzate da vincoli di real-time, sia possibile derogare da tale prescrizione e realizzare sistemi con un numero ridotto di livelli (tipicamente Physical, Data link, Application). Tale approccio, che costituisce una palese violazione dei principi dell'OSI-RM, è ormai accettato negli ambienti di controllo di processo dove le esigenze temporali dei processi applicativi impongono l'uso di architetture veloci, con un numero ridotto di livelli.

3.3.2 Application Context

I vari processi possono comunicare solo se il loro “**Universo del Discorso**” coincide o si sovrappone. L'Universo del Discorso (che costituisce l'insieme delle informazioni associate ad un processo) viene descritto formalmente attraverso quello che si chiama “*Schema Concettuale*”.

Quest'ultimo è **una struttura dati che definisce tutte le informazioni che costituiscono l'universo del discorso**. In pratica, lo *Schema Concettuale* è quella parte di processo che definisce tipi e variabili, è cioè il modo di rappresentare l'informazione usata dai processi.

Se due processi sono espressi con lo stesso Universo del Discorso, usando una rappresentazione comune, ovvero se coincide il modo di rappresentare l'informazione sia sotto forma di bit che di variabili (coincide quindi la sintassi con cui le informazioni sono

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

rappresentate ed il tipo di informazione coinvolta nella cooperazione fra i processi applicativi), allora è possibile la comunicazione e quindi l'interoperabilità fra i processi. L'interoperabilità non richiede tuttavia che ci sia una completa coincidenza degli universi del discorso dei processi, ma basta che ci sia una adeguata **sovrapposizione fra gli schemi concettuali**.

La porzione condivisa dello *schema concettuale* è chiamata **“Shared schema”**.

Se uno Shared schema è troppo ampio esso può essere suddiviso in più “Application Context” (i contesti A e B di Fig.3.3.3).

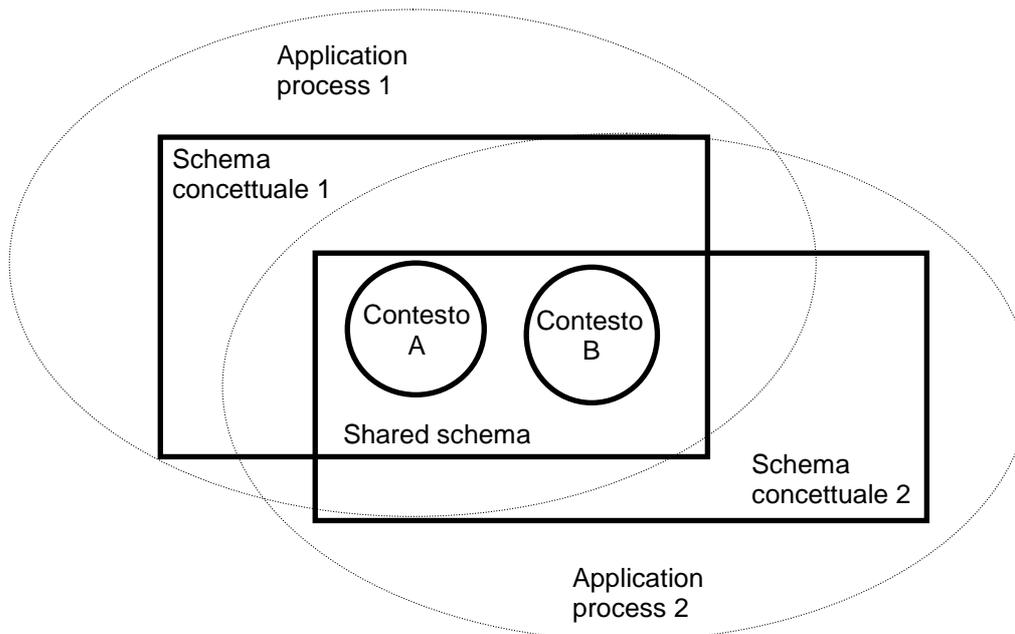


Fig.3.3.3 Schemi concettuali dei processi Applicativi

Infatti, poiché spesso gli schemi concettuali sono molto complessi e due processi possono interagire in modo diverso (nel senso che possono scambiarsi informazioni di diverso tipo a seconda dei casi) può essere utile considerare anziché l'intero Shared schema, dei suoi sottoinsiemi che rappresentano quegli aspetti che sono importanti per un certo tipo di interazione.

Se ad esempio due processi stanno cooperando scambiandosi delle informazioni relative ad alcune variabili di controllo, la cosa importante è che tali variabili siano espresse colla stessa sintassi. Non è importante che coincida tutta la rappresentazione poiché dobbiamo controllare solo un particolare contesto dei due processi, che è un sottoinsieme dell'intero “shared schema”.

Come puntualizzato nel modello di riferimento ISO_OSI, il compito principale del Presentation layer è quello di effettuare una traduzione delle informazioni in una notazione comune. Quindi, piuttosto che effettuare la traduzione di tutto l'universo del discorso basterà semplicemente realizzare la traduzione di quel particolare contesto relativo all'informazione che deve essere scambiata. Nel Protocollo MAP, a livello di Facility level e Shop level si opera con dati molto complessi, relativi alla pianificazione e gestione di tutte le attività (anche quelle relative ai vari tipi di controllo, fino al livello di campo) e la varietà dei dati rende complessa la definizione degli Shared schema e degli Application

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

Context, per cui è importante avere il supporto di un livello che si fa carico di risolvere i problemi di traduzione e compatibilità delle sintassi. Ciò spiega perché i nodi Full MAP utilizzino tutti e 7 i livelli dell'architettura, compreso il presentation layer.

Quando ci spostiamo ai livelli più bassi dell'impianto (il livello di campo e di controllo) la differenziazione fra i vari tipi di dati è più limitata e può essere gestita anche in assenza di un livello (il presentation layer) che mette a disposizione una sintassi astratta (Abstract Syntax N.1). I problemi di compatibilità fra i dati può essere risolta all'interno dell'Application layer, rendendo possibile un'architettura a tre soli livelli.

3.3.3. Application Process

Abbiamo finora parlato di processo applicativo (Application Process, AP) in termini generali, senza distinguere esplicitamente fra AP ed applicazione, che è l'insieme di funzioni realizzate mediante la cooperazione di varie AP.

Nell'OSI-RM, 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.

E' importante sottolineare che una 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.

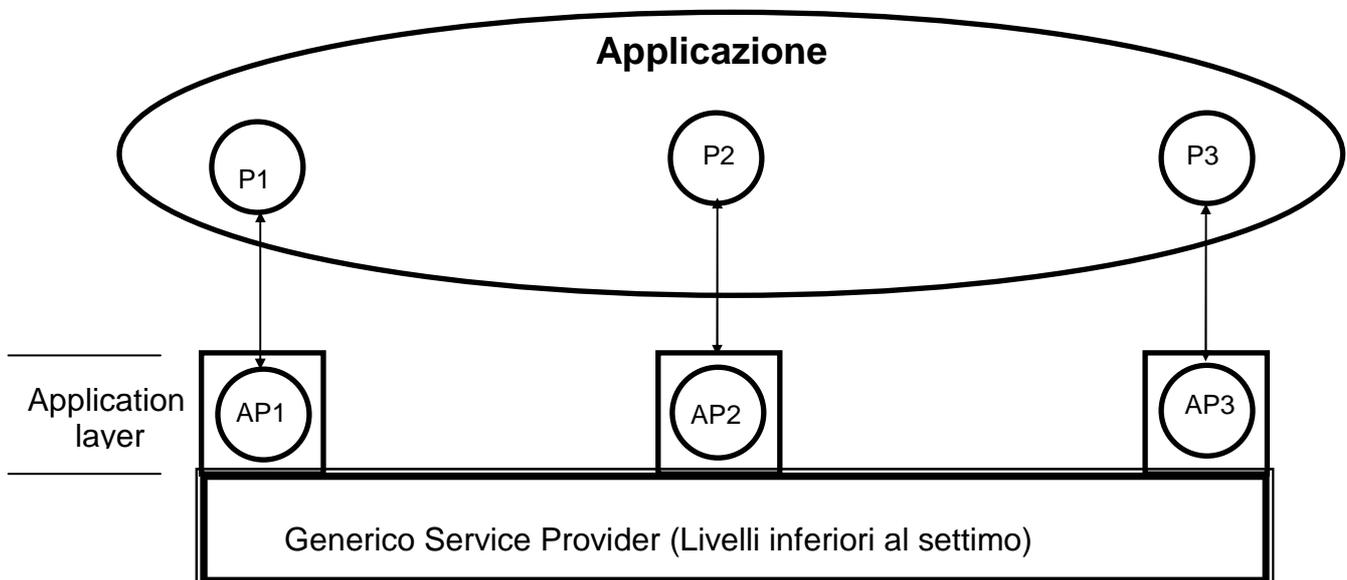


Fig.3.3.4: Relazione tra processi applicativi reali ed *Application Process*.

Questo concetto è illustrato in Figura 3.3.4 dove P1, P2, P3 sono tre processi che realizzano una particolare applicazione, mentre AP1, AP2, AP3 sono gli APs relativi a P1, P2, P3. AP1, AP2, AP3 cooperano sfruttando il sottostante Service Provider che può essere sia un sistema di comunicazione completo, con tutti i livelli dello stack OSI, che un sistema compatto (per applicazioni di tipo tempo-critico) con i soli livelli Physical e Data Link.

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

Un Application Process può essere suddiviso in due parti:

- **APPLICATION AGENT**
- **APPLICATION ENTITY**

L'Application Agent interfaccia il particolare sistema ai servizi applicativi standard, resi disponibili dall'Application layer, e contiene quegli aspetti dell'AP non direttamente coinvolti nella comunicazione. Ad esempio può contenere degli algoritmi per l'elaborazione dell'informazione o per acquisire localmente dati da un sensore o interagire con l'utente.

Tale interfacciamento diventa rilevante poiché, mentre nei livelli inferiori al settimo abbiamo ragionato in termini di entità considerando una struttura omogenea, costituita da una sola entità che comunica con modalità standard con altre entità simili, nell'Application layer consideriamo processi applicativi (cioè software) che hanno una struttura completamente diversa da quella del sistema di comunicazione della macchina e differiscono da un'applicazione all'altra. Si ha quindi il problema di creare un opportuno mapping tra il processo applicativo reale e il sistema di comunicazione. Tale mapping viene realizzato proprio attraverso quello che abbiamo definito Application Agent, cioè una funzionalità che interfaccia il processo vero e proprio (che rappresenta l'utente del sistema) col sistema stesso.

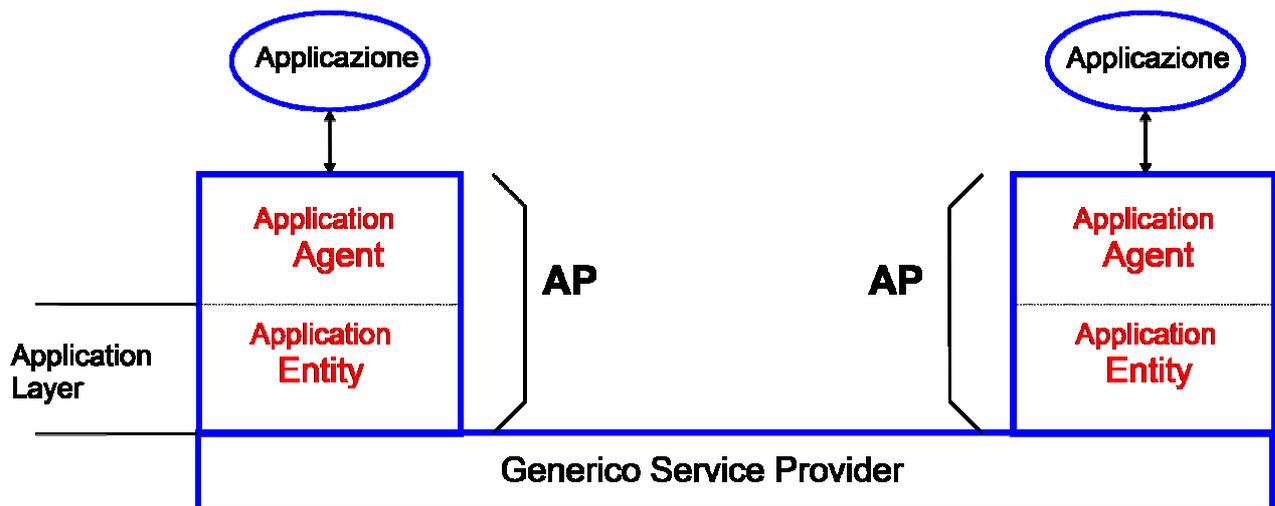


Fig.3.3.5: La suddivisione dell'Application Process in Agent ed Entity.

Per meglio chiarire il concetto può essere utile un esempio:

Si consideri un sensore che fornisce una certa variabile (segnale) per il sistema; come si può collegare la variabile al sistema ed inviare le informazioni al dispositivo di controllo? Nasce così l'esigenza di un "mapping" fra la variabile fisica ed il sistema. E' compito dell'Application Agent, creare all'interno del sistema una rappresentazione astratta della variabile. Ciò sarà realizzato da un software che leggerà il valore della grandezza fisica (temperatura, tensione) e la rappresenterà come una variabile all'interno del sistema.

La seconda parte in cui è suddiviso un Application Process è costituita dall'Application Entity, che supporta funzionalità di comunicazione. Una AE fornisce all'AP un set di servizi per supportare la comunicazione, che vengono raggruppati in Application Service

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

Elements (ASE) logicamente correlati e definiti in modo da costituire moduli funzionali. Ogni ASE fornisce dunque un insieme di servizi omogenei.

La comunicazione fra APs può avvenire solo attraverso AEs dello stesso tipo. Le AEs non hanno un'esistenza separate dagli AP che esse supportano; infatti, ciascuna AE identifica uno ed uno solo AP nel senso che, una specifica AE può appartenere solo ad uno specifico AP. Peraltro, un AP può essere supportato da AEs di tipo differente, così come AEs dello stesso tipo possono supportare differenti APs (una AE per ogni AP).

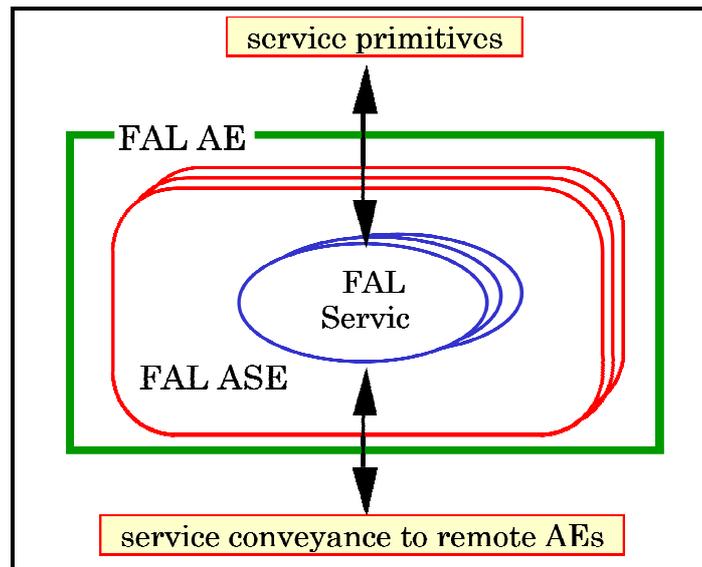


Figura 3.3.6: Struttura di una Application Entity.

Gli ASE sono dunque dei moduli specializzati per realizzare particolari funzionalità. Possono essere divisi in due gruppi:

un primo gruppo è legato alla peculiarità dello standard considerato, vale a dire un gruppo che svolge funzioni che sono correlate allo scopo del processo applicativo. In questo gruppo possiamo considerare alcuni ASE che sono stati definiti dall'ISO quando fu creato l'OSI-RM, quali per esempio:

- Job Transfer & Manipulation (JTM), che serve a trasferire lavori su macchine remote, nel caso in cui un processo applicativo abbia necessità di fare eseguire delle elaborazioni da particolari macchine;
- File Transfer Access & Management (FTAM), che si occupa dell'accesso, gestione e trasferimento dei files;
- Commitment, Concurrency & Recovery (CCR), che permette di coordinare interazioni fra più attori anche in caso di ripetuti fallimenti del sistema;
- Electronic Messaging (EM), che svolge funzioni concernenti la posta elettronica;

A questi ASE possiamo aggiungere il "Manufacturing Message Specification" (MMS) che fu definito nel protocollo MAP per abilitare la comunicazione fra dispositivi intelligenti presenti nelle applicazioni di manufacturing.

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

L'altro gruppo di ASE invece ha il compito di creare un ambiente (definito Associazione) mediante cui vengono forniti i servizi applicativi; cioè supporta gli altri ASE (appartenenti al primo gruppo) con dei servizi di comunicazione.

Ad esempio consideriamo due processi utenti A e B (vedi figura 7) i quali devono comunicare e interagiscono nell'Application layer mediante una particolare ASE (chiamata ad esempio α ASE).



Fig.3.3.7: Uso di una Associazione per la cooperazione fra due generiche ASE

Questa ASE fornisce il servizio che è richiesto attraverso una comunicazione virtuale che, nel livello applicativo, viene definita **associazione**. 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 utilizzata da tutte le altre ASE. Tale ase. Mostrata in Fig. 3.3.8, viene chiamata Association Control Service Element (ACSE).

Quindi tutte le ASE, all'interno dell'Application layer, sono supportate dal punto di vista della comunicazione da un'ASE chiamata, genericamente "**Association ASE**". Nel modello di riferimento OSI si considera solamente il tipo di associazione uno a uno, cioè un processo può comunicare solo con un altro processo.

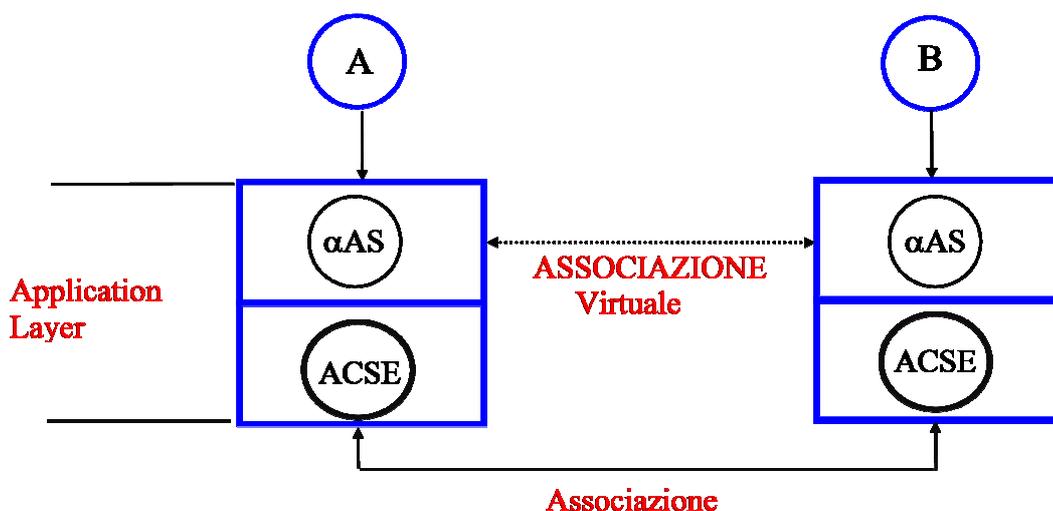


Fig.3.3.8 L'ACSE come supporto per le associazioni fra le ASE.

Nelle applicazioni di automazione è stato compiuto un passo ulteriore. In questo contesto i processi applicativi possono avere esigenze di scambi multipli cioè le interazioni a livello applicativo possono essere del tipo uno a molti. Per tali tipi di applicazioni è stato definito un modulo che supporta la comunicazione uno a molti e prende il nome di **Relationship ASE** (il relationship ASE è definito nell'Application layer del Fieldbus IEC 61158).

Si ha quindi una distribuzione di informazione, con un processo produttore ed n processi consumatori. Un tale tipo di funzionamento è reso possibile dal fatto che le reti per l'automazione non sono mai reti geografiche ma esclusivamente reti locali o più spesso Fieldbus. In questo caso è il sistema di comunicazione che fornisce questa capacità di distribuzione dell'informazione essendo in genere una rete a bus, intrinsecamente broadcast. Pertanto è già il livello di Data Link Layer che supporta il broadcast dell'informazione.

Nelle reti per applicazioni real-time o time-critical, grazie all'uso di architetture con un numero ridotto di livelli (in genere il data link layer è direttamente connesso all'Application layer) la caratteristica broadcast del data link layer può essere trasferita all'Application layer e quindi resa disponibile ai processi applicativi.

Un'altra peculiarità differenzia l'ACSE da Relationship-ASE (o simili tipi di ASE per i Fieldbus). Mentre nelle applicazioni normali quando dei processi devono comunicare viene creata on line (in tempo reale) un'associazione (come nel caso di posta elettronica), nelle applicazioni di controllo di processo le associazioni sono precostituite, cioè sono create prima e non vengono più eliminate. Ciò è legato al fatto che le applicazioni per controllo di processo di norma non cambiano nel tempo, e le associazioni tra le variabili sono note a priori (vengono definite in fase di progetto) ed implementate in fase di configurazione del sistema .

3.3.4 MMS

Tutti i tipi di ASE che abbiamo citato appartengono ormai alla storia delle reti per l'automazione, nel senso che l'evoluzione tecnologica ha portato alla definizione di nuovi tipi di ASE in linea con nuove esigenze di comunicazione e cooperazione, generate dalle moderne applicazioni di automazione.

Ciò nonostante, è importante sottolineare che i sistemi attuali utilizzano degli ASE che sono derivati da quelli che abbiamo citato, originariamente definiti dall'ISO, di cui mantengono i principi chiave.

Fra tutti, l'MMS ha avuto una particolare importanza principalmente per due ragioni:

- E' stato **definito appositamente per le applicazioni industriali** e contiene funzionalità che sono pensate per la cooperazione fra processi industriali.
- Nonostante oggi non sia più usato, i principi che sono stati definiti per MMS restano ancora validi e sono stati incorporati all'interno degli Application layer della maggior parte di Fieldbus e reti per l'automazione industriale esistenti. Pertanto, lo studio dell'MMS fornisce una chiave di lettura delle problematiche presenti nella comunicazione fra Processi Applicativi e delle strategie per risolverle.

Il Manufacturing Message Specification (MMS) è un protocollo dell'Application layer (o meglio, un Application Service Element, ASE) che abilita la comunicazione fra i dispositivi

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

intelligenti che si trovano nelle applicazioni per controllo di manufacturing. Esempi di tali dispositivi includono sistemi di visione, robots, controllori programmabili, macchine a controllo numerico, ecc. Per descrivere come le funzionalità offerte dall'MMS si adattino alle applicazioni di manufacturing, faremo riferimento ad un sistema costituito da una cella di lavorazione con 2 robot, due nastri trasportatori (conveyors), un "tool magazine" ed un tavolo di lavorazione. Il primo robot preleva i pezzi da un nastro e li deposita sul tavolo dove il secondo robot li lavora e li pone poi sull'altro nastro trasportatore. La cella di lavorazione include anche una stazione chiamata "Supervisory controller" che sebbene non direttamente coinvolta nelle attività di lavorazione funge da interfaccia fra la cella ed il resto dell'ambiente di manufacturing.

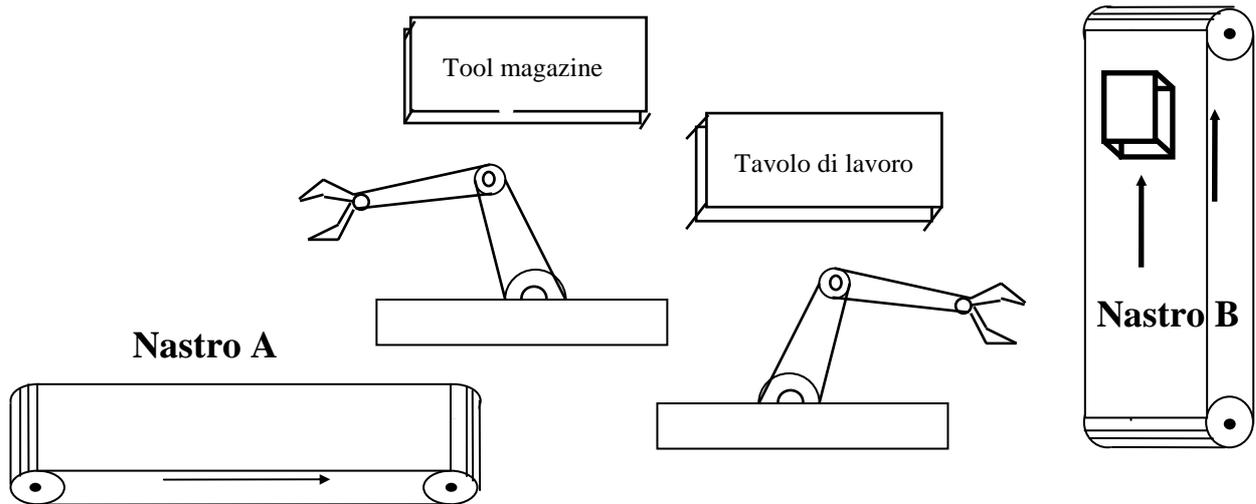


Fig.3.3.9: principali componenti di una cella di lavorazione.

Sebbene le funzioni di controllo della cella possano essere implementate in modo centralizzato, l'implementazione distribuita è quella più utilizzata. A tale scopo, la cella è configurata come una rete con quattro nodi e cioè il "device controller", il robot1, il robot 2 ed il Supervisory controller. I dispositivi che nel sistema non possono essere considerati intelligenti, non sono collegati direttamente alla rete ma vi accedono attraverso il device controller. Lo schema complessivo della rete è quello di Fig.3.3.10

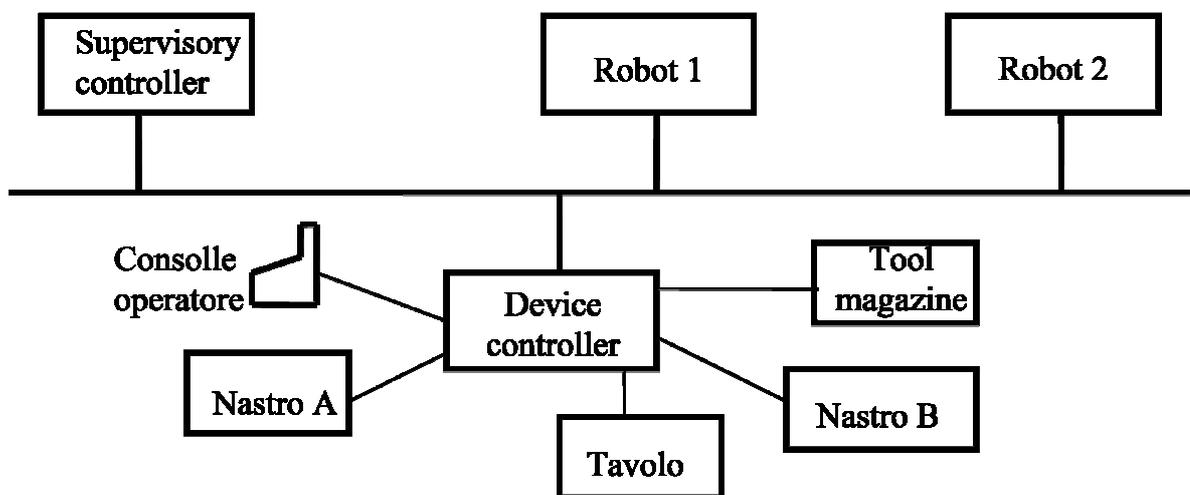


Fig.3.3.10: Interconnessione fra i componenti di una cella di lavorazione.

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

Lo standard MMS è costituito da due componenti principali: il **Core Standard** ed il **Companion Standard**.

Il “Core” fornisce un modello generico, che non rappresenta alcuna classe particolare di dispositivi. Tale modello è basato sull’osservazione che tutti i dispositivi elettronici disponibili per l’automazione di fabbrica forniscono un certo insieme di servizi comuni. Ciò rende il “core” di MMS un modello molto flessibile e facilmente applicabile in molti contesti.

D’altra parte, esigenze specifiche possono nascere nell’ambito di particolari classi di dispositivi elettronici, quali i robot, i controllori numerici e così via. L’obiettivo del “Companion standard” è proprio quello di soddisfare queste esigenze, integrandole nel modello definito dal “core”. Quindi, mentre il “Core” standard fornisce un modello applicabile ad una intera classe di dispositivi, il “Companion” standard arricchisce il modello rendendolo più specifico, cioè più aderente ad alcuni dispositivi. Come indica il nome “Companion Standard” contiene la descrizione dei dispositivi prodotti dalle varie compagnie. A tale scopo, per ogni classe di dispositivi è possibile 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.

Quindi, lo scopo del “Companion standard” è allargare l’orizzonte della standardizzazione oltre i limiti del generico dispositivo definito dal “core”, tramite l’introduzione di estensioni peculiari di una particolare sottoclasse di dispositivi.

L’elemento fondamentale del “core” è il **Virtual Manufacturing Device** (VMD) che descrive gli aspetti visibili di un dispositivo. All’interno del contesto definito dal VMD, le risorse disponibili sono descritte tramite oggetti quali: Domain, Program invocation, Eventi, Variabili, ecc. In Fig.3.3.11 sono descritte graficamente le relazioni che legano alcuni di questi oggetti fra di loro e con il VMD.

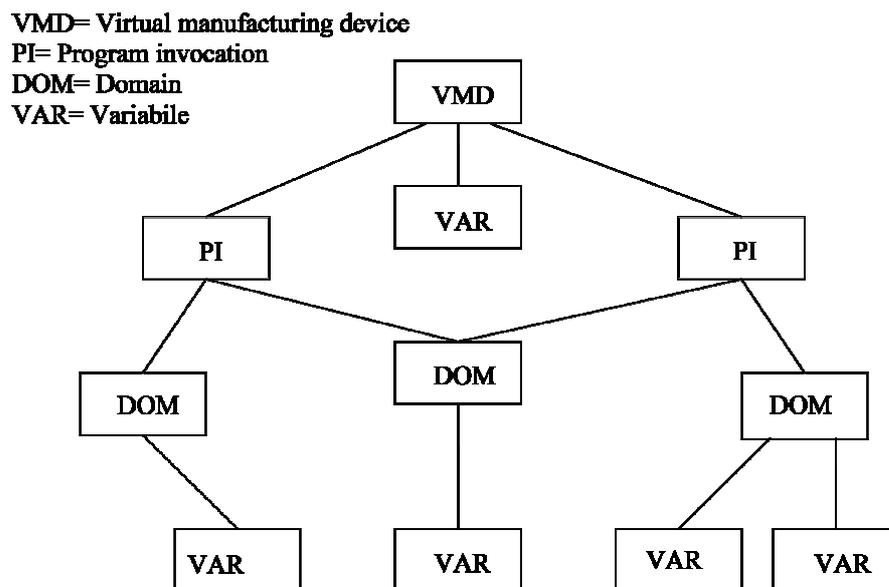


Fig.3.3.11: Relazioni fra il VMD e gli oggetti ad esso correlati

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

Il VMD è l'elemento fondamentale di tutto lo standard MMS, nel senso che tutti gli altri oggetti ed i servizi che operano su di essi hanno senso solo nel contesto definito dal VMD.

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.

Il VMD non contiene per esempio informazioni su attributi quale il volume fisico o la tensione di alimentazione del dispositivo poiché tali parametri sono irrilevanti dal punto di vista di chi deve gestire le attività che esso esegue. Al contrario, il suo stato logico e fisico sono attributi essenziali in questo contesto, perché forniscono informazioni significative sul livello corrente di operatività del dispositivo fisico.

Un VMD non può essere creato o cancellato tramite i servizi creati da MMS, poiché tali azioni sono al di fuori dello standard stesso e devono essere eseguite in base a convenzioni locali al dispositivo che sono fissate dal produttore. I servizi che operano sul VMD sono invece destinati alla sua interrogazione per conoscere i dati identificativi (*Identify*), lo stato corrente (*Status* e *Unsolicited status*), e le risorse locali ad esso (*Get name list* e *Get capability list*) o eventualmente per modificare l'identificatore di una risorsa definita localmente (*Rename*). La descrizione di un VMD contiene anche informazioni riguardanti i Companion standard che esso contempla (List of abstract syntaxes) e la lista delle macchine di Upload che sono attive presso di esso.

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. I seguenti gruppi di servizi sono integrati nell'MMS e sono quelli considerati indispensabili per soddisfare le esigenze del controllo di processo discreto (Manufacturing):

- **Variable access:** questa classe di servizi soddisfa uno dei requisiti fondamentali dell'ambiente distribuito. A questo scopo 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. I servizi di "read/write" ad esempio permettono ad uno dei due robot di comunicare all'altro il numero di pezzi presenti sul tavolo di lavoro mentre l'Information report permette di notificare periodicamente al Supervisory controller l'andamento delle operazioni. Poiché i processi risiedono su stazioni differenti della rete, sono richiesti opportuni meccanismi che permettono di definire e gestire i "data types" delle variabili

- **Message passing:** I servizi che operano su singole variabili sono, come abbiamo visto, essenziali per il controllo di processo. Quando però occorre trasferire blocchi di informazione, occorrono servizi per lo scambio di messaggi. Fondamentalmente, un messaggio può essere trasferito fra due processi o fra un processo e dispositivi specializzati quale per esempio una consolle di operatore. 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, usato 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.

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

- **Terminate Download sequence**, usato da un Server per indicare ad un processo client che la sequenza di download è terminata.

• **Resource sharing**: quando in un sistema esistono risorse condivise, si pone il problema della loro gestione. Con riferimento al sistema dei due robot, Robot1 e Robot2 questi devono ad esempio sincronizzarsi fra di loro per utilizzare il tavolo di lavoro. A questo scopo nell'MMS è disponibile un opportuno set di servizi per la sincronizzazione delle attività di risorse condivise. Questa sincronizzazione è 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).

• **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*.

4. Conclusioni.

Dall'analisi presentata è possibile trarre alcune utili conclusioni.

Innanzitutto si è visto come in un ambiente di controllo di processo siano presenti diverse tipologie di informazioni da trasmettere, con caratteristiche anche molto differenti. Ogni ambiente è caratterizzato da diverse frequenze di generazione e di consumo delle informazioni e da quantità diverse di dati da scambiare. Pertanto, per ogni ambiente va scelto il Fieldbus più opportuno.

Le differenze fra i vari Fieldbus assumono maggiore rilevanza nel DLL che ha la responsabilità di gestire l'accesso al mezzo fisico. Da questo punto di vista non si può dire che esista un Fieldbus ottimo, adatto a tutte le occasioni, poiché esso va valutato in relazione all'ambiente operativo. Dal punto di vista dell'utente è quindi importante orientare la propria scelta, non seguendo dei principi assoluti, ma verificando che le esigenze del proprio sistema siano soddisfatte.