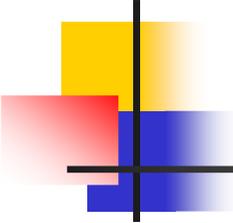


# ANALISI DEI SISTEMI AD EVENTI

---

Anno accademico 2014/15

Ludovica Adacher  
adacher@dia.uniroma3.it

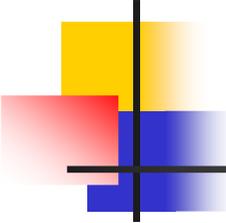


# INTRODUZIONE

---

Negli ultimi vent'anni si è evidenziata la necessità di studiare i sempre più numerosi sistemi realizzati dall'uomo, tendenzialmente molto complessi, considerati non tradizionali rispetto alla trattazione classica proprie della Teoria dei Sistemi e del Controllo.

Questi sistemi dinamici, i cui stati assumono diversi valori logici o simbolici, piuttosto che numerici, in corrispondenza dell'occorrenza di eventi, non sempre possono essere descritti in termini numerici.

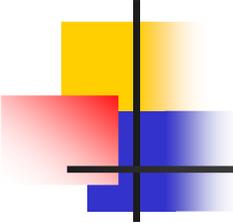


# INTRODUZIONE

---

Ne sono esempi significativi i processi produttivi, le reti di elaboratori elettronici, di trasporto, di comunicazione e sistemi formati per integrazione delle suddette tipologie di sistemi.

Esempi di eventi sono: l'arrivo di un cliente nel sistema o la sua partenza da esso, il completamento di una lavorazione o il guastarsi di una macchina in un sistema di produzione, la trasmissione/ricezione di un pacchetto di dati in una rete di telecomunicazioni, il verificarsi di un disturbo o il cambiamento del segnale di riferimento in un complesso sistema di controllo



# INTRODUZIONE

---

L'evoluzione nel tempo di un sistema con tali caratteristiche sembra essere descritta da sequenze di occorrenze di cambiamenti discreti e qualitativi del sistema, ignorando i micro cambiamenti che avvengono continuamente.

## **SISTEMA** (qualitativa)

Ente fisico che risponde alle sollecitazione esercitata da una certa azione producendo una reazione.

Per sviluppare tecniche di progetto, di controllo e/o di valutazione delle prestazioni di un sistema sulla base di specifiche predefinite è necessaria una definizione **QUANTITATIVA : MODELLO FORMALE.**

# INTRODUZIONE

Variabili che evolvono nel tempo

- CAUSE ESTERNE AL SISTEMA (INGRESSI)

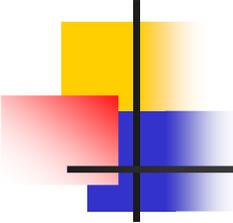
Grandezze il cui andamento nel tempo può essere indipendente dal tipo di sistema

- EFFETTI (USCITE)

Grandezze il cui andamento nel tempo dipende, almeno in parte dal tipo di sistema e dalle cause esterne



In generale, non è immediato legare in modo semplice l'uscita con l'ingresso, cioè realizzare la dipendenza ingresso/uscita.

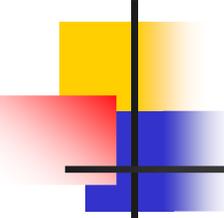


# INTRODUZIONE

---

## **STATO**

Rappresenta il comportamento del sistema ad un dato istante di tempo, concentrando in sé l'informazione sul passato e sul presente del sistema  $X(t)$  è lo stato all'istante  $t$

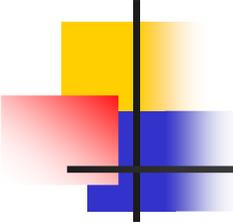


# INTRODUZIONE

---

Un sistema ad eventi discreti (SED) si può definire come un sistema dinamico il cui comportamento è caratterizzato dall'occorrenza di eventi istantanei con un cadenzamento irregolare non necessariamente noto.

Le caratteristiche indiscusse possedute da un SED sono fondamentalmente legate all'evoluzione dinamica basata sull'occorrenza **asincrona** degli eventi, anziché sull'avanzamento sincrono del tempo, e al fatto che almeno alcune delle variabili che descrivono il comportamento di un SED sono discrete.



# INTRODUZIONE

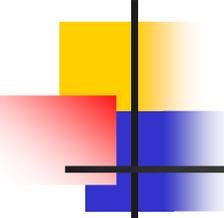
---

Dal punto di vista formale, un SED può essere considerato come un sistema dinamico, con un opportuno spazio di stato e un proprio meccanismo di transizione di stato.

Un sistema ad eventi discreti è un sistema il cui comportamento dinamico è caratterizzato dall'accadimento **asincrono** di eventi che individuano lo svolgimento di attività di durata non necessariamente nota.

Formalmente, un sistema ad eventi discreti è caratterizzato da:

- un insieme **E** degli eventi accadibili;
- spazio di stato costituito da un insieme discreto **X**
- **evoluzione** dello stato event-driven, cioè regolata dagli eventi: lo stato evolve nel tempo in dipendenza dell'accadimento di eventi asincroni, appartenenti all'insieme E

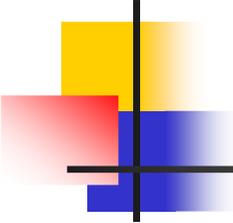


# INTRODUZIONE

---

L'equazione che descrive l'evoluzione dello stato a partire dallo stato iniziale  $x_0$  è:

- $x_{k+1} = f(x_k, e_k) \quad k \in \mathcal{N}$
- $x_{k+1}$  è lo stato del sistema dopo l'accadimento del k-esimo evento
- $e_k$  è il k-esimo evento accaduto dall'istante iniziale considerato, che fa transire lo stato da  $x_k$  a  $x_{k+1}$
- $f: X \times E \rightarrow X$  è la funzione di transizione di stato



# INTRODUZIONE

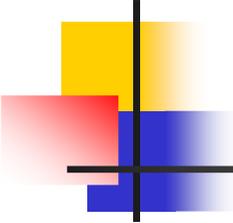
---

L'esempio più semplice per esplicitare i concetti fin qui enunciati riguardo ai SED, è senza dubbio il sistema a coda o ad accodamento. Un sistema di questo tipo può essere considerato come il blocco elementare con cui costruire le rappresentazioni di molte tipologie di SED.

Un sistema a coda si fonda su tre componenti fondamentali:

- le entità che attendono per utilizzare le risorse, dette clienti
- le risorse per cui ci si accoda, detti serventi o servitori
- lo spazio in cui si attende, che è la coda vera e propria

I clienti possono essere persone, messaggi in reti di telecomunicazioni, task in computer, semilavorati in sistemi di produzione, veicoli in reti di trasporto, ecc. Esempi di serventi corrispondenti sono invece ancora persone, canali di comunicazione, processori, macchine, semafori, ecc.



# INTRODUZIONE

---

Visto come un SED, il sistema a coda è caratterizzato dall'insieme di eventi

$E = \{a, p\}$  con

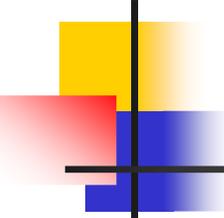
- a: evento di arrivo di un cliente;
- p: evento di partenza di un cliente.

La variabile di stato più intuitiva è il numero di clienti in coda; in questo caso si pone  $X = \{1, 2, 3, \dots\}$

Per specificare completamente le caratteristiche di un sistema a coda bisogna ancora definire:

- La capacità della coda, cioè il numero di clienti che possono accodarsi (spesso considerato illimitato);
- La disciplina di accodamento, cioè la regola con cui si sceglie il prossimo cliente da servire tra quelli in coda.

Collegando tra loro più blocchi elementari coda si costruiscono reti di code.



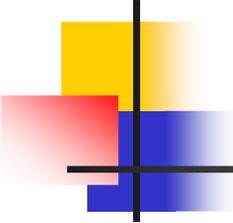
# INTRODUZIONE

---

E' attraverso i Modelli ad Eventi Discreti che si effettua un'astrazione del comportamento dei sistemi, registrando l'occorrenza di determinati eventi discreti (traccia/traiettoria degli eventi).

Un MED è un modello matematico in grado di rappresentare l'insieme delle tracce degli eventi che possono essere generate da un sistema.

Le tracce possono essere rappresentate con due diversi livelli di astrazione, sulla base dei quali si distinguono : modelli logici e modelli temporizzati.

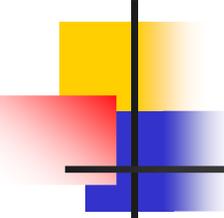


# INTRODUZIONE

---

Nei **MODELLI LOGICI** la traccia degli eventi è costituita semplicemente da una sequenza di eventi  $\{e_1, e_2, \dots\}$ , in ordine di occorrenza, senza alcuna informazione circa i tempi di occorrenza degli eventi; dato uno stato iniziale  $x_0$ , la traiettoria dello stato verrà costruita nel tempo la sequenza di stati  $\{x_0, x_1, x_2, \dots\}$ , risultanti dall'accadimento della sequenza di eventi, ma non è possibile specificare gli istanti di tempo in cui avvengono le transizioni di stato.

Nei **MODELLI TEMPORIZZATI** invece la traccia degli eventi è costituita da una sequenza di coppie  $\{e_1 t_1, e_2 t_2, e_3 t_3, \dots\}$ , dove ogni evento  $e_i$  è accoppiato al suo tempo di accadimento,  $t_i$ , eventualmente stocastico: dato uno stato iniziale  $x_0$ , la traiettoria dello stato verrà costruita nel tempo la sequenza di stati  $\{x_0, x_1, x_2, \dots\}$ , risultanti dall'accadimento della sequenza di eventi, si sa che le transizioni di stato avvengono negli istanti di occorrenza degli eventi.

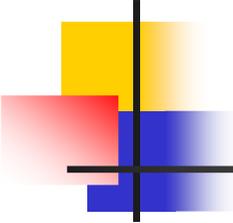


# INTRODUZIONE

---

I **modelli logici** rendono agevole lo studio delle proprietà qualitative del sistema e consentono quindi di effettuare l'analisi strutturale di un SED, mentre i **modelli temporizzati** permettono di studiare i diversi comportamenti nel tempo del sistema, pertanto sono indispensabili qualora si voglia effettuare l'analisi prestazionale di un SED.

Nella formulazione del modello logico è fondamentale specificare l'insieme delle traiettorie ammissibili, ossia le sequenze di eventi fisicamente realizzabili. A questo scopo può essere adottato uno dei noti formalismi sviluppati per rappresentare le transizioni di stato in un SED, come gli AUTOMI, o le RETI DI PETRI.

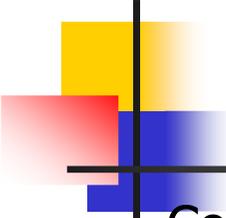


# INTRODUZIONE

---

La necessità di modelli per descrivere il funzionamento dei sistemi è una costante di tutti i problemi di ingegneria: non è possibile progettare alcunché se non si dispone di un modello adeguato.

Peraltro, il tipo di modello che serve può essere molto diverso, a seconda dell'uso che se ne deve fare. Per esempio, il modello dinamico di un sistema che si presta per il progetto di un sistema di controllo è generalmente molto più semplice di un simulatore dello stesso sistema.



# INTRODUZIONE

---

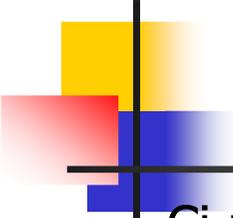
Cosa vogliamo descrivere con i modelli nel contesto del controllo logico?

Vogliamo descrivere il funzionamento di impianti molto complessi ed eterogenei:

celle robotizzate, centri di lavorazione meccanica, impianti batch/chimici, ecc.

Ciascuno di questi può essere visto come un insieme di sotto-sistemi, dispositivi, macchinari, ecc. opportunamente interconnessi, ognuno dei quali può essere modellizzato con molto dettaglio (v. modello motore, serbatoio, ecc.).

A noi interessa studiare questi processi ad un livello di astrazione più elevato, in cui si evidenzino le *sequenze di operazioni*, con i relativi problemi di sincronizzazione, parallelismo, ecc.



# INTRODUZIONE

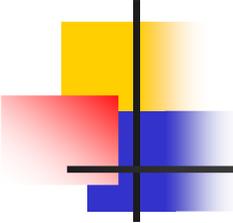
---

Ci poniamo domande come:

- Che operazione devo svolgere dopo l'operazione  $X$ ?
- Le operazioni  $X$  e  $Y$  possono essere svolte in parallelo?
- In quali condizioni non devo eseguire l'operazione  $X$ ?
- Ci sono risorse sufficienti per svolgere le operazioni che mi servono?

Questo modo di ragionare è tipico dei sistemi manifatturieri, le cui caratteristiche macroscopiche sono descrivibili con *condizioni logiche di funzionamento discrete*, senza valori numerici, come ad es. "macchina pronta per la lavorazione", "macchina in attesa", "macchina guasta".

Tali condizioni logiche cambiano in modo istantaneo da un valore all'altro, ad es. con un *comando* "accendi la macchina", oppure con un *segnale* di "fine corsa raggiunto". Normalmente, non è noto a priori né quale sia il nuovo valore, né l'istante temporale in cui avviene il cambiamento.



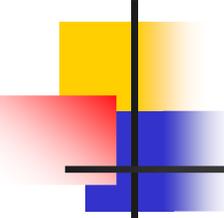
# INTRODUZIONE

---

L'evoluzione di tali sistemi può allora essere studiata in termini di cambiamento delle condizioni logiche di funzionamento discrete, per effetto di sequenze di comandi/segnali.

Si evidenziano così alcuni funzionamenti tipici, come:

- evoluzione parallela e asincrona (macchine in parallelo → vanno sincronizzate)
- presenza di scelte (bivio in una linea)
- condivisione di risorse (magazzino di utensili o pallet condiviso da più macchine, vincoli di mutua esclusione nell'allocazione delle risorse)



# INTRODUZIONE

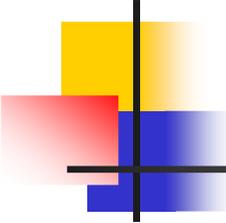
---

Che tipo di modello matematico ci serve per descrivere questi fenomeni?

Osservando il parallelismo che sussiste fra i concetti seguenti:

- condizione logica di funzionamento  $\leftrightarrow$  stato (discreto)
- sequenza comandi/segnali  $\leftrightarrow$  sequenza di ingressi

è facile capire che lo strumento che ci serve è una qualche forma di *sistema dinamico*.

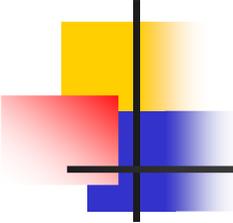


# INTRODUZIONE

---

I sistemi dinamici che conosciamo (sistemi a tempo continuo o discreto), però, hanno alcune caratteristiche che non si prestano bene alla descrizione dei fenomeni che abbiamo citato in precedenza:

- lo spazio di stato è continuo, ovvero le variabili variano in modo continuo sull'asse reale, mentre a noi interessa esprimere concetti come "serbatoio pieno" o "serbatoio vuoto" (invece di "il serbatoio contiene  $X$  litri di acqua") → **sistemi a stato discreto**
- l'evoluzione è guidata dal "tempo", ovvero lo stato può variare ad ogni istante; nel nostro caso lo stato cambia solo in certi istanti, con transizioni istantanee (da "macchina occupata" a "macchina libera") → **sistemi ad eventi**: lo stato varia quando si verifica un evento (istantaneo)

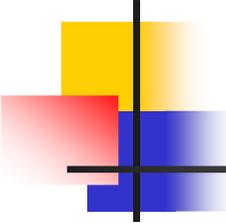


# INTRODUZIONE

---

Nei sistemi guidati dagli eventi (event-driven) la modellizzazione e l'analisi sono rese complicate dal fatto che occorre specificare i meccanismi asincroni di occorrenza degli eventi nel tempo.

Tra sistemi time-driven e event-driven c'è concettualmente la stessa differenza che sussiste in un calcolatore tra le operazioni sincronizzate dal clock e quelle gestite tramite interrupt.

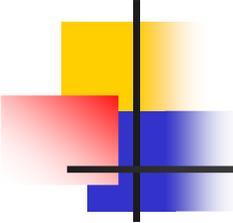


# INTRODUZIONE

---

I sistemi dinamici si dividono in:

- sistemi dinamici a tempo continuo o discreto l'evoluzione è guidata dal "tempo"
- sistemi dinamici ad eventi discreti l'evoluzione è guidata dall'accadimento degli "eventi", considerati per semplicità istantanei, che accadono ad intervalli irregolari non noti a priori
- sistemi ibridi l'evoluzione è determinata sia dal tempo sia da eventi

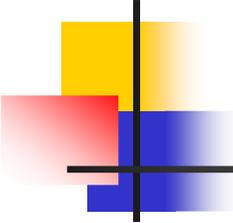


# INTRODUZIONE

---

Un sistema ad eventi discreti è caratterizzato dai seguenti elementi:

- Le variabili di stato assumono valori numerici discreti (cioè una quantità finita o numerabile di valori) o sono descrivibili in termini simbolici (parole, stringhe, ecc.);
- Gli stati cambiano in corrispondenza dell'occorrenza di eventi, i quali anch'essi possono essere descritti in termini non numerici.



# INTRODUZIONE

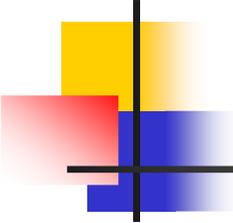
---

ESEMPIO:

Lo stato assume solo i quattro valori  $x_1, x_2, x_3$  e  $x_4$ , e cambia solo in alcuni istanti  $(t_1, t_2, t_3, t_4)$ , in corrispondenza degli eventi  $e_1, e_2, e_3, e_4$ .

Assumendo che il sistema sia deterministico (nel senso che la legge che determina lo stato successivo in corrispondenza dell'occorrenza di un evento sia unica), l'informazione completa è fornita dalla sequenza (temporizzata) di eventi:

$$(t_1, e_1) (t_2, e_2) (t_3, e_3) (t_4, e_4)$$



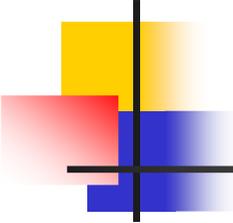
# INTRODUZIONE

---

Spesso, tuttavia, non ci interessa *quando* il sistema entra in un determinato stato o *quanto a lungo* il sistema rimane nel medesimo stato, ma piuttosto l'ordinamento degli eventi (e quindi di transizioni):

$e_1, e_2, e_3, e_4.$

Ci interessa cioè se un evento accade *prima* o *dopo* un altro. Eliminando la temporizzazione, stiamo di fatto modellizzando il *comportamento logico* del sistema.



# INTRODUZIONE

---

Un modello logico di questo tipo consente di:

- distinguere le sequenze di eventi che sono compatibili con delle specifiche di comportamento
- verificare se un determinato stato è raggiungibile, e con quale sequenza di eventi
- verificare se il sistema si blocca in uno stato