

# Introduzione ai Data Flow Diagram

versione 16 marzo 2009

©Adriano Comai

<http://www.analisi-disegno.com>

# Obiettivo

- Fornire elementi di base sui DFD (Data Flow Diagram)
- Il tema è trattato in modo più approfondito nel modulo formativo online su <http://www.adcorsi.com>

- E nel corso in aula "Analisi delle funzioni con i Data Flow Diagram":

[http://www.analisi-disegno.com/a\\_comai/corsi/sk\\_dfd.htm](http://www.analisi-disegno.com/a_comai/corsi/sk_dfd.htm)

# L'approccio strutturato

- 1966                      **programmazione** strutturata
- inizio anni '70        **disegno** strutturato
- metà anni '70         **analisi** strutturata

⇒ "primo" tentativo di fornire linee guida per le attività di sviluppo software

⇒ utilizzo di modelli formali e diagrammatici

# L'analisi strutturata

anni 70 (metà)	Analisi strutturata	Gane - Sarson, De Marco
1984	Analisi strutturata per eventi	McMenamin - Palmer
1985	Analisi strutturata per sistemi in tempo reale	Ward - Mellor

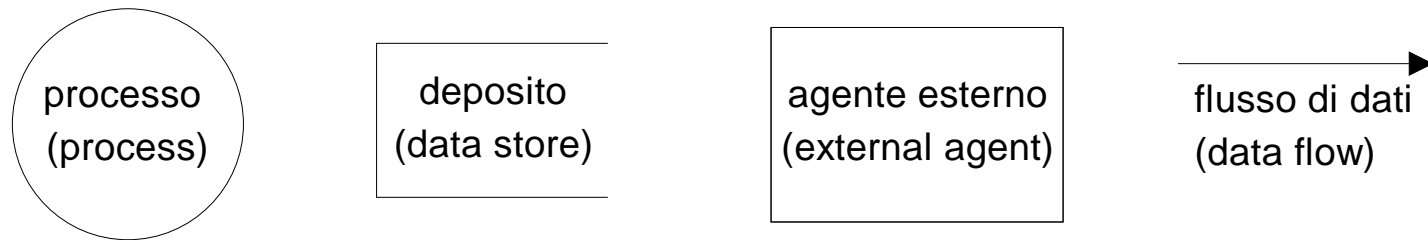
## Caratteristiche:

- definizione delle interazioni tra sistema e mondo esterno
- partizionamento gerarchico del sistema in processi, collegati tramite flussi di dati
- utilizzo del diagramma di flusso di dati - Data Flow Diagram

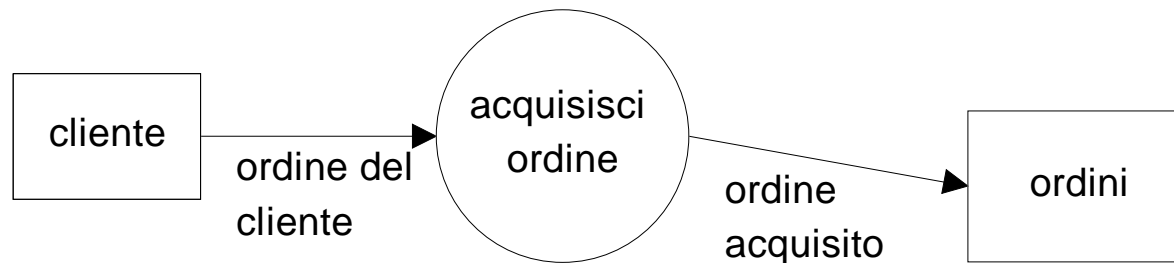
# I DFD oggi

- Oggi i Data Flow Diagram sono usati prevalentemente per attività di analisi a livello di business
- Il loro uso in ambito informatico è in declino: l'approccio Object Oriented ha progressivamente soppiantato l'approccio strutturato per l'analisi e la progettazione del software

# Elementi di base dei DFD



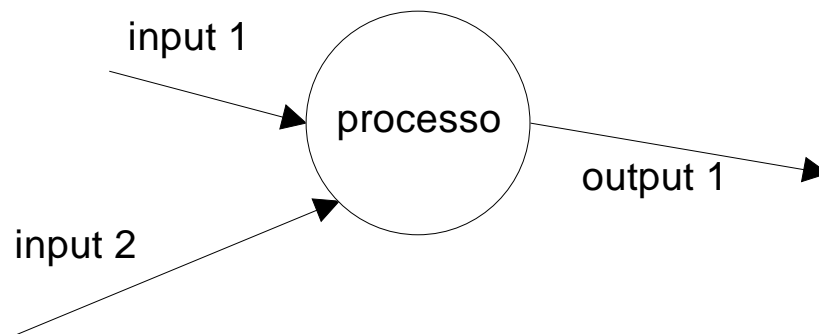
esempio:



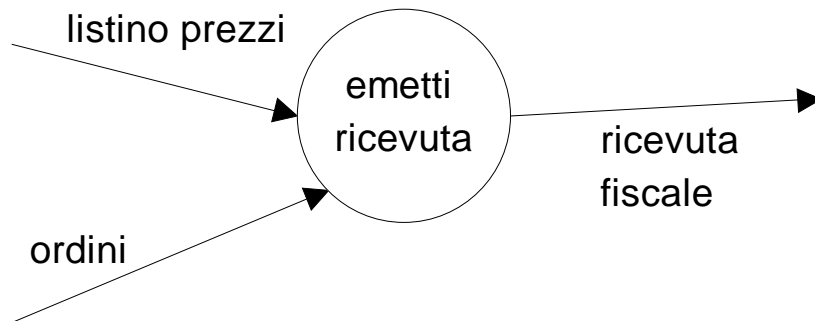
# Processo

È un'attività di trasformazione, che acquisisce dati in input e li trasforma in dati di output. Pertanto ogni processo:

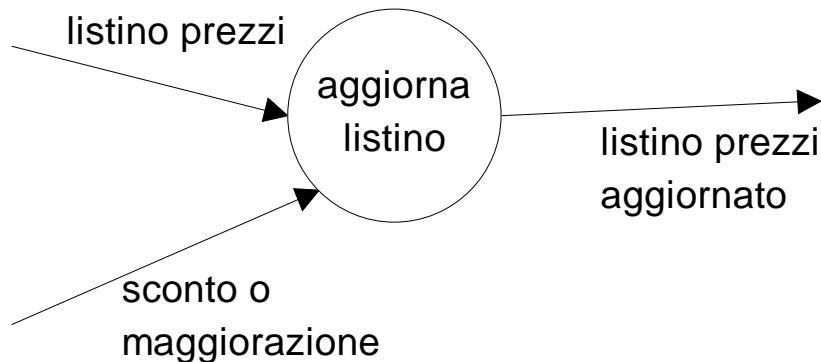
- deve essere collegato ad almeno un flusso di dati in input e ad almeno uno in output
- i flussi in output devono essere diversi rispetto ai flussi di input (in quanto oggetto di trasformazione)



# Processo: tipologie di trasformazione



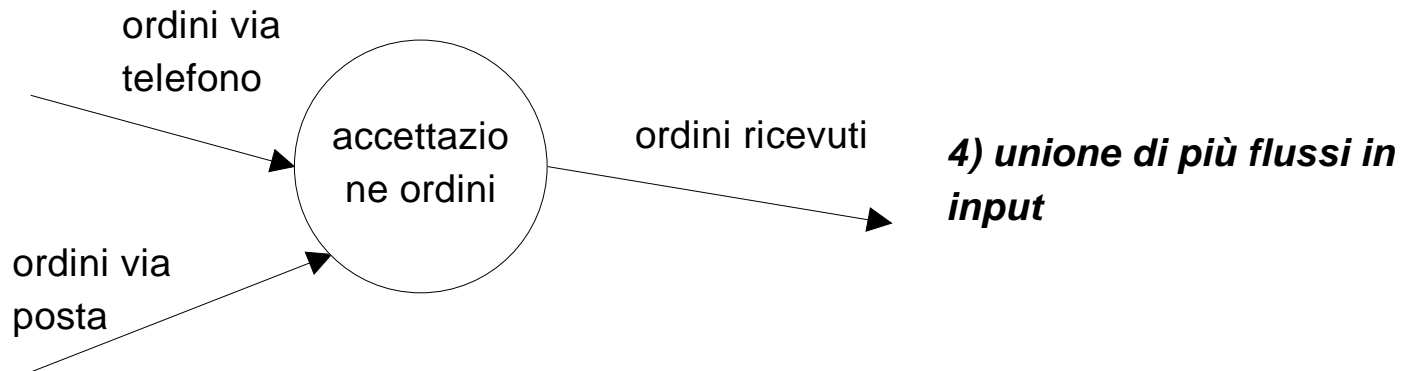
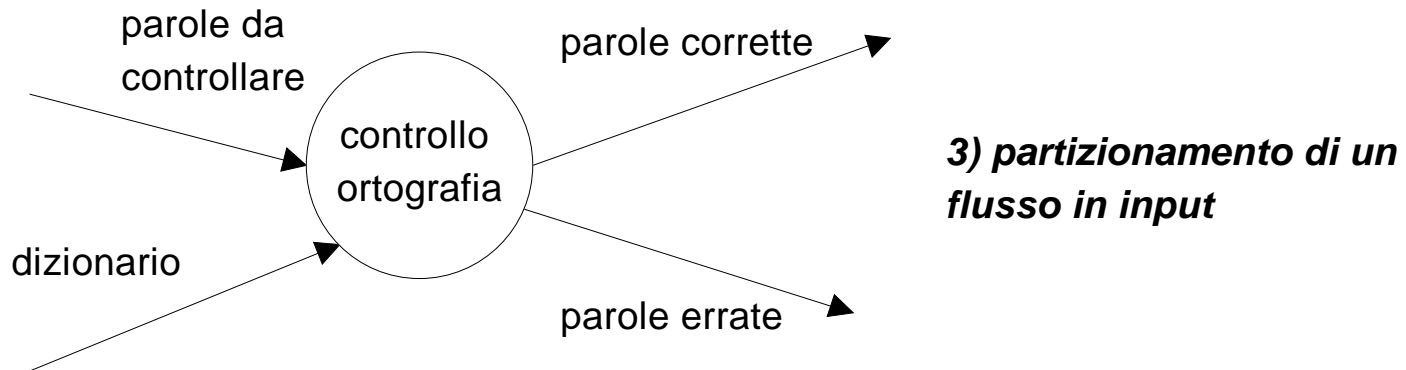
**1) produzione in output di dati diversi rispetto a quelli di input**



**2) produzione in output di dati della medesima tipologia di quelli di input, ma con valori diversi**



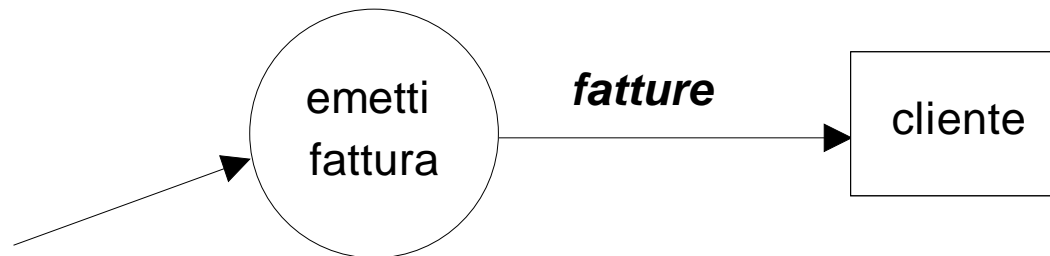
# Processo: tipologie di trasformazione (...)



# Flusso di dati

È un canale che trasporta materiale (es. dati) omogeneo:

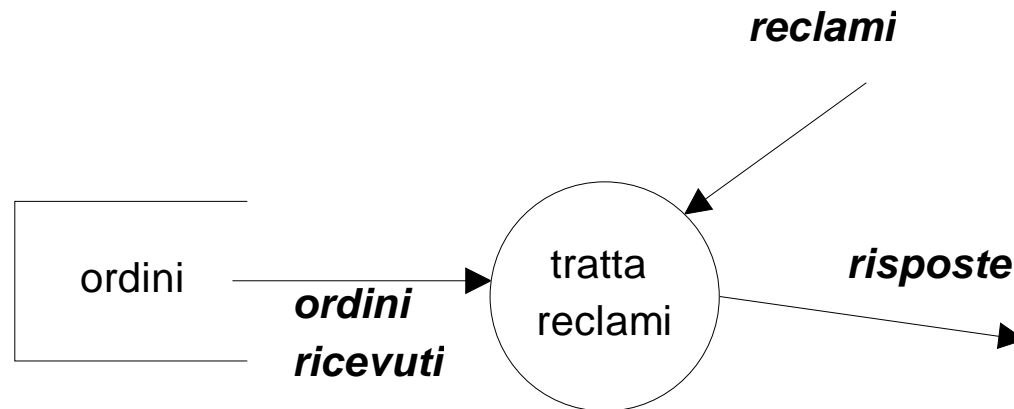
- ha una direzione
- connette due punti del sistema
- a uno dei due capi vi è necessariamente un processo (o un agente esterno) che produce il flusso o lo acquisisce



# Contenuto del flusso di dati

Il flusso può trasportare:

- dati organizzati in strutture (es. ordini memorizzati in un archivio) o non strutturati (es. reclami, risposte)
- zero, una o più occorrenze (es. dal deposito ordini possono essere letti da zero a molti ordini)



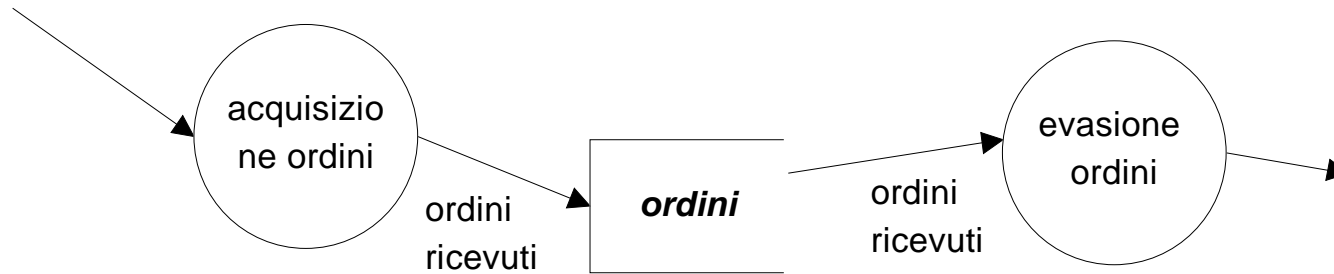
# Deposito (data store)

- è un “archivio” a cui accedono i processi del sistema
- è, per definizione, statico: mentre il flusso trasporta i dati, che sono quindi "in movimento", nel deposito i dati sono messi "a riposo", disponibili per essere trattati dai processi
- i dati contenuti nel deposito corrispondono alle entità (una o più) del sistema (rappresentabili con un modello Entity-Relationship)



# Ruolo del deposito

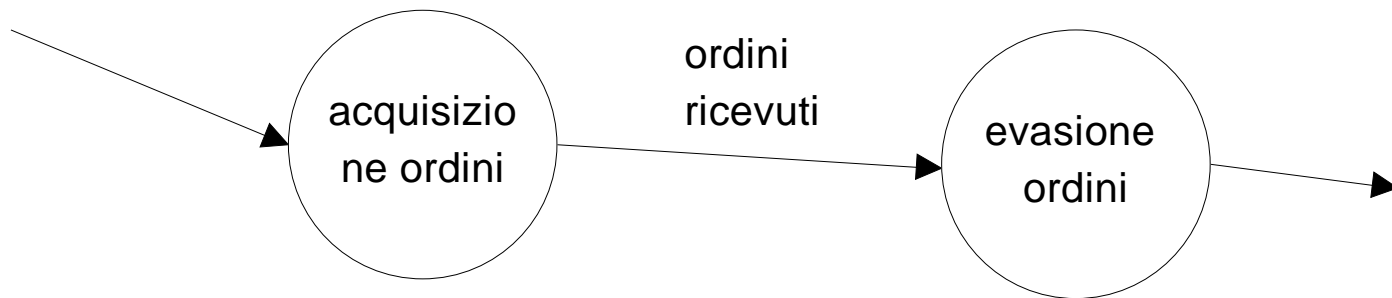
- quando la connessione tra due processi avviene tramite deposito, i processi sono "asincroni":



il processo che accede ai dati contenuti nel deposito (es. evasione ordini) può iniziare la propria attività in un momento successivo al termine dell'attività del processo che li memorizza (es. acquisizione ordini)

# Ruolo del deposito (...)

- quando la connessione avviene senza il tramite di un deposito, i processi sono "sincroni":

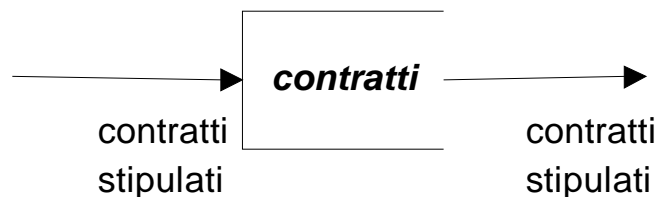
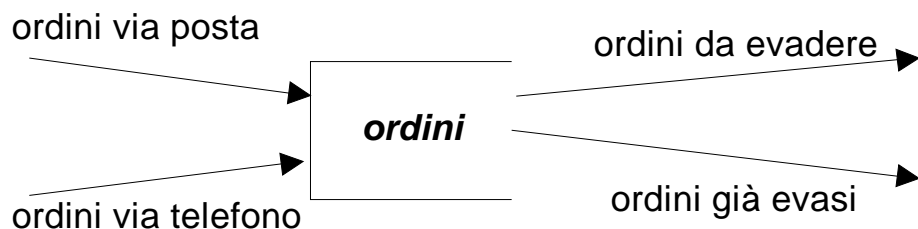


il secondo processo inizia la propria attività immediatamente al termine dell'attività del primo

# Deposito e flussi di dati

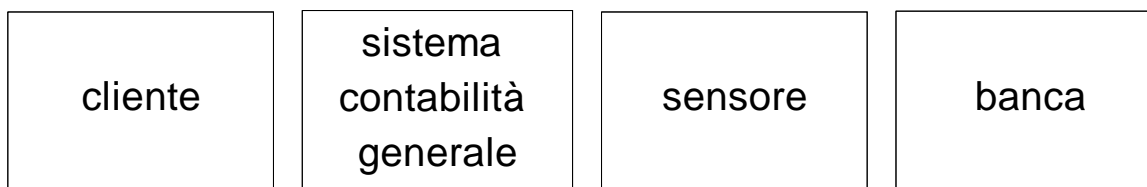
- i flussi di dati che entrano in un deposito lo aggiornano, mentre quelli che ne escono lo leggono
- i flussi in input al deposito, e in output dal deposito, aggiornano o leggono un sottoinsieme dei dati contenuti nel deposito, non necessariamente l'intero deposito:

- flussi diversi possono aggiornare e/o leggere il medesimo deposito: ciascuno di essi corrisponde ad un determinato sottoinsieme del deposito
- è anche lecito che il medesimo flusso aggiorni e legga il deposito



# Agente esterno (external agent)

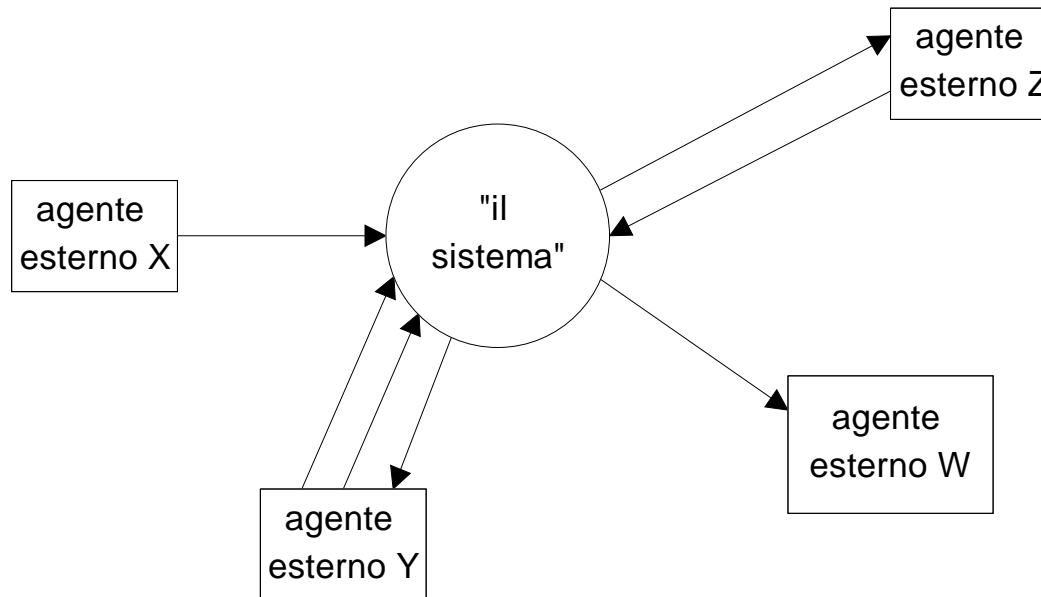
- è un sistema esterno, con il quale il sistema che analizziamo scambia informazioni in input e/o in output
- può essere una persona, un'organizzazione, un sistema hardware / software, un oggetto qualsiasi
- come ogni sistema, potrebbe essere analizzato, ma:
  - l'agente esterno è da considerarsi come una "scatola nera", della quale non ci interessano le caratteristiche interne
  - ci interessano solo gli scambi di dati (flussi) tra l'agente esterno ed il sistema che stiamo analizzando





# Diagramma di contesto

- ogni sistema è in relazione con il "mondo esterno", dal quale riceve input e verso il quale produce output



# Diagramma di contesto (...)

Il diagramma di contesto è la rappresentazione delle interazioni tra il sistema e il "mondo esterno", e contiene:

- un solo processo, che rappresenta il sistema analizzato nella sua globalità
- tutti gli agenti esterni
- i flussi che agenti esterni e sistema si scambiano
- eventuali depositi

# Ruolo degli agenti esterni

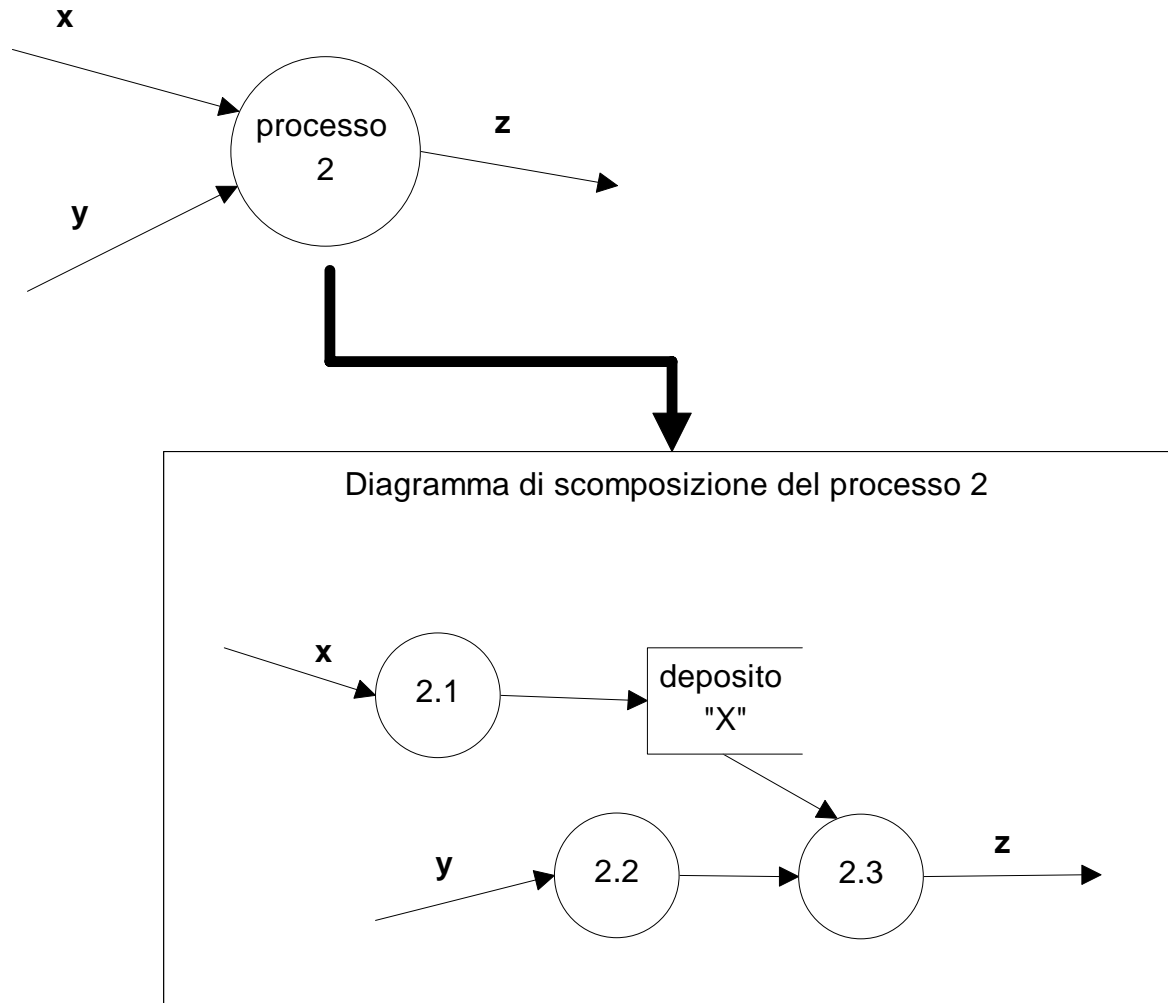
- l'individuazione degli agenti esterni è la base per la definizione del contesto del sistema analizzato
- gli agenti esterni corrispondono alle particolari entità del "mondo esterno" con cui il sistema è in relazione
- definire gli agenti esterni, e i flussi di dati che scambiano con il nostro sistema, permette di precisare i "confini" del sistema che stiamo analizzando:
  - le attività che producono i flussi indirizzati verso gli agenti esterni sono interne al sistema
  - le attività che producono i flussi che arrivano dagli agenti esterni sono al di fuori del sistema

# Scomposizione dei processi

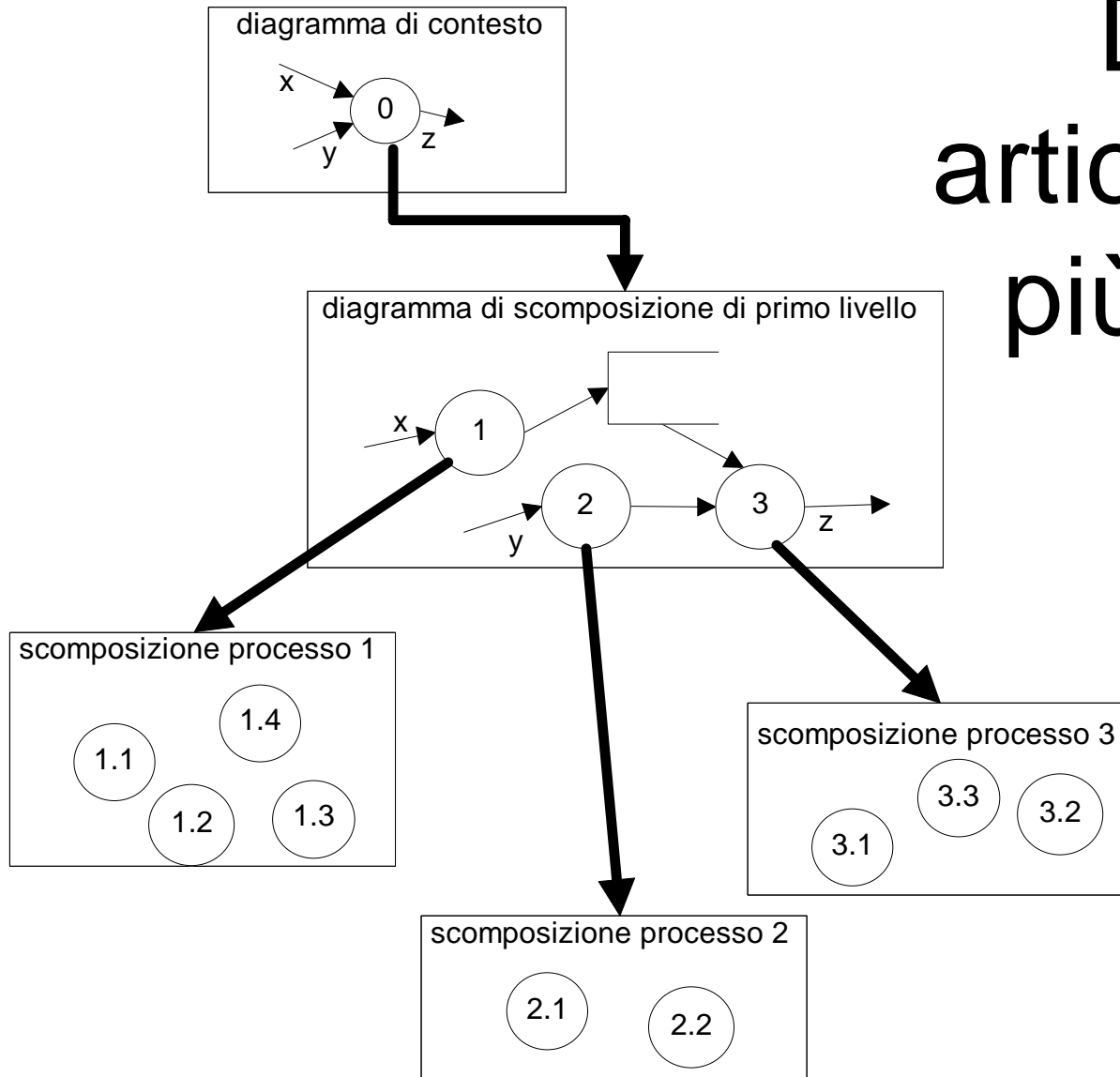
Ogni processo può essere scomposto in sottoprocessi:

- la scomposizione origina un nuovo diagramma
- regola di scomposizione: i flussi di input e di output collegati al processo "padre" devono essere collegati anche ai processi "figli" (padri e figli devono avere i medesimi input ed output "netti")
- la scomposizione è reversibile: è cioè possibile aggregare più processi in un macro-processo

# Scomposizione dei processi



# DFD articolati su più livelli



# DFD articolati su più livelli

Il meccanismo di scomposizione dei processi permette di rappresentare il sistema a diversi livelli di dettaglio:

- dal diagramma più sintetico, con un unico processo (contesto)
- attraverso una serie di diagrammi intermedi
- fino ai diagrammi di dettaglio, che evidenziano i processi elementari (non ulteriormente scomposti)

# I problemi della scomposizione

1. In che modo (secondo quali criteri) è opportuno partizionare un processo?

- Sono state proposte diverse tecniche per aiutare l'analista nella scomposizione.
- La più efficace, particolarmente nella scomposizione del diagramma di contesto, si basa sulla individuazione degli "eventi" a cui il processo deve rispondere, e nella definizione di un sottoprocesso per ciascun evento, che tratti l'evento in modo completo producendo tutte le "risposte" necessarie per soddisfarlo.



# I problemi della scomposizione

2. In quanti sottoprocessi bisogna partizionare ciascun processo?

- Non esiste una regola vera e propria. Il numero dei sottoprocessi dipende dal tipo di processo e dai criteri (dalla tecnica) utilizzata per il partizionamento.
- Comunque, poiché ogni scomposizione genera un nuovo diagramma, è importante che il diagramma risultante risulti comprensibile, e che pertanto il numero di (sotto) processi contenuti non sia troppo elevato.
- L'applicazione ai DFD di studi di psicologia sperimentale ha portato Tom De Marco a proporre un numero indicativo di 7 (+ o - 2) sottoprocessi per ogni processo, ma sono numeri da prendere con buon senso, non da applicare in modo meccanico.

# I problemi della scomposizione

## 3. Fino a che livello di dettaglio spingersi nella scomposizione?

- Ogni processo può essere più o meno complesso, e generare quindi un numero di sottoprocessi elementari molto diverso da quelli originati da un altro processo.
- Le tecniche utilizzate per il partizionamento influenzano anche il numero di diagrammi prodotti nella scomposizione, ed il livello di dettaglio necessario.
- Il livello analitico da raggiungere è sostanzialmente determinato dagli obiettivi dell'analisi

# Punti di forza dei DFD

- focus sull'interazione tra il sistema e il mondo esterno (approccio "sistemico"), e definizione chiara del contesto
- capacità di rappresentare qualunque tipo di sistema, a diversi livelli
- intuitività, immediatezza come strumento di comunicazione
- costituiscono una linea guida per gli analisti, in quanto costringono a porsi le domande a cui l'analisi deve dare risposta

# Limiti dei DFD

- rischi di orientare la scomposizione dei processi alle soluzioni tecniche (al "come bisogna implementare" anziché al "cosa va fatto"), particolarmente nei livelli più dettagliati
- le "regole sintattiche" sono limitate: la qualità dei modelli prodotti dipende fortemente dall'esperienza di chi li utilizza

Per approfondimenti e altri materiali:

<http://www.analisi-disegno.com>