Parte 1 - Introduzione

Indice

- Analisi, progetto, realizzazione, ciclo di vita
- Analisi strutturata
  - Strumenti dell'analisi
  - Esempi
- Progettazione strutturata

Analisi Strutturata

Si avvale di questi strumenti

- Diagramma di Flusso dei Dati (DFD, Data Flow Diagram)
- Specifiche di Processo (Pspec, Process Specifications)
- Dizionario dei Dati (DD, Data Dictionary)

- *Al termine dell'analisi viene prodotto il Documento di Specifica composto da DD, Pspecs e DD.*

- Produce una specifica semiformale, quasi eseguibile

T. De Marco "Structured Analysis and System Specification" Yourdon Press, Prentice-Hall, 1978

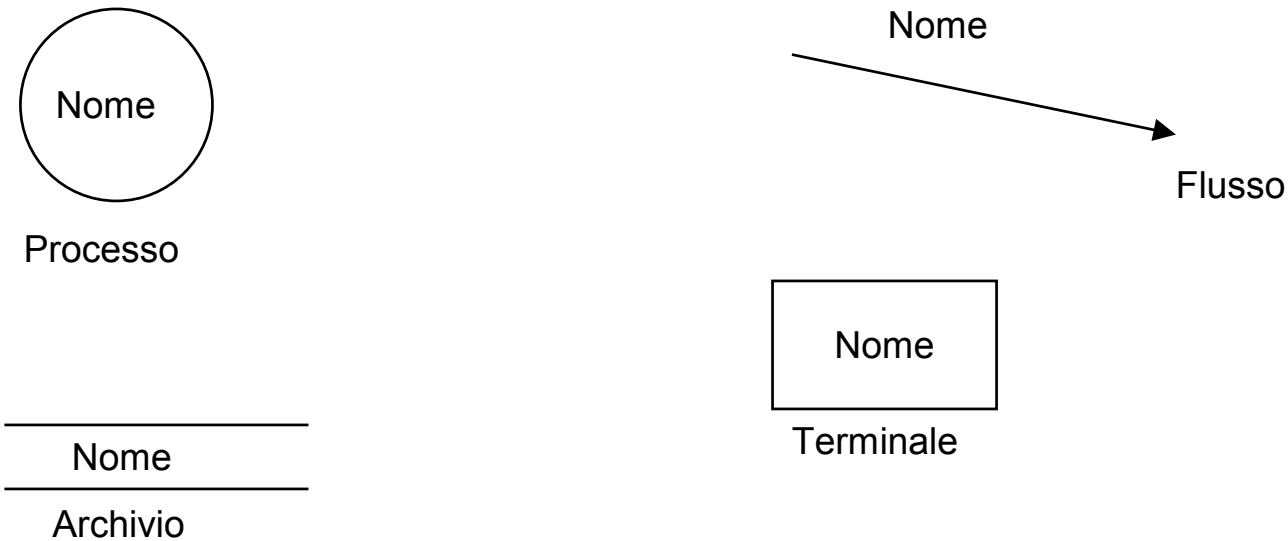
Diagrammi di Flusso dei Dati

- I DFD descrivono la scomposizione delle attività svolte dal sistema e i flussi di informazione tra le diverse attività
- Il DFD mostra il *flusso dei dati (delle informazioni) non il flusso di controllo.*
- Questa è la differenza essenziale rispetto alla *flowchart,* che invece rappresenta il flusso di controllo.
  - In un DFD, tipicamente non ci sono cicli: I dati passano attraverso centri di trasformazione
  - Cicli e decisioni hanno a che fare con il controllo.

- Un DFD è un grafo nel quale compaiono questi simboli:
  - Flussi (Data Flow)
  - Processi (Process)
  - Archivi (Data store)
  - Elementi terminali (sorgenti/destinazioni, source/sink))

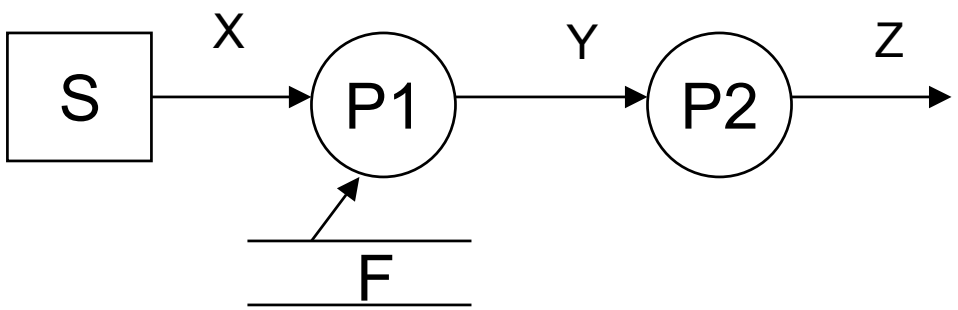
Simboli del DFD

- Flusso: arco orientato etichettato con un nome
- Processo: “palla” etichettata con un nome e un numero
- Archivio: è una coppia di barre etichettata con un nome all’interno
- Elemento terminale: rettangolo con un nome all’interno



Vengono usate svariate notazioni alternative alle precedenti

Esempio



Il flusso X arriva dalla sorgente S al processo P1 e viene trasformato nel flusso Y. P1 per trasformare X in Y richiede il file F.

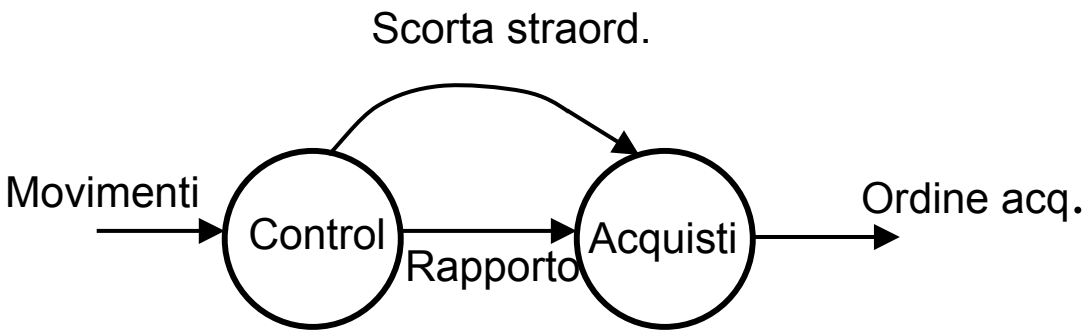
Il flusso Y viene trasformato successivamente in Z da P2.

X, Y e Z sono “pacchetti” di dati: un pacchetto X viene trasformato in un pacchetto Y.

Flusso Dati

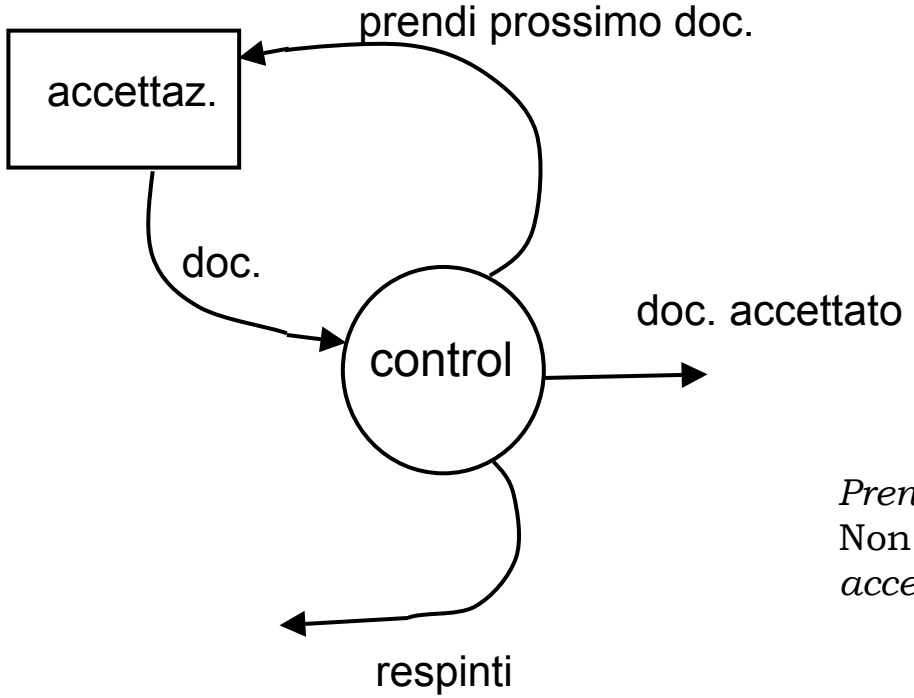
- Un flusso dati è un canale orientato attraverso il quale fluiscono pacchetti informativi di composizione nota
- Uno stesso flusso può raccogliere due unità informative diverse, ma solo se l'una è sempre accompagnata dall'altra

- Nella figura, i due flussi Scorta straord e Rapporto non sono riunibili in un solo flusso: il primo passa in condizioni straordinarie (magazzino privo di una merce), il rapporto viene prodotto ogni tanto; inoltre la composizione è diversa.



Cosa non è un flusso dati

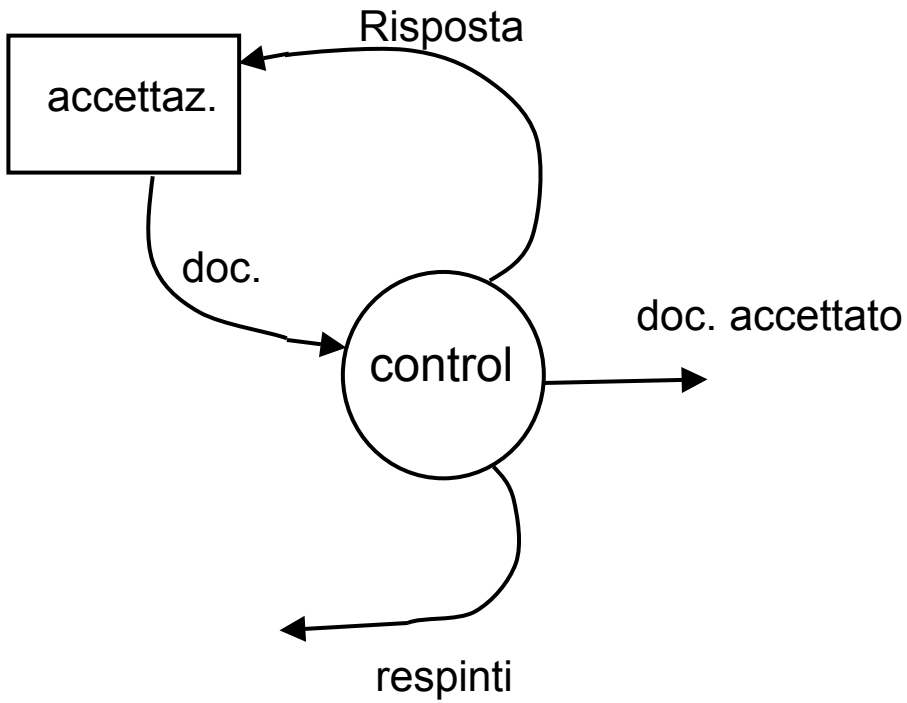
- Un flusso dati non è un flusso di controllo
- Un flusso dati non è un segnale per attivare/disattivare processi
- Un flusso dati non è un dato di controllo (switching item)



Prendi prossimo doc non è un flusso dati. Non c'è alcun dato che passa da *control* a *accettazione*.

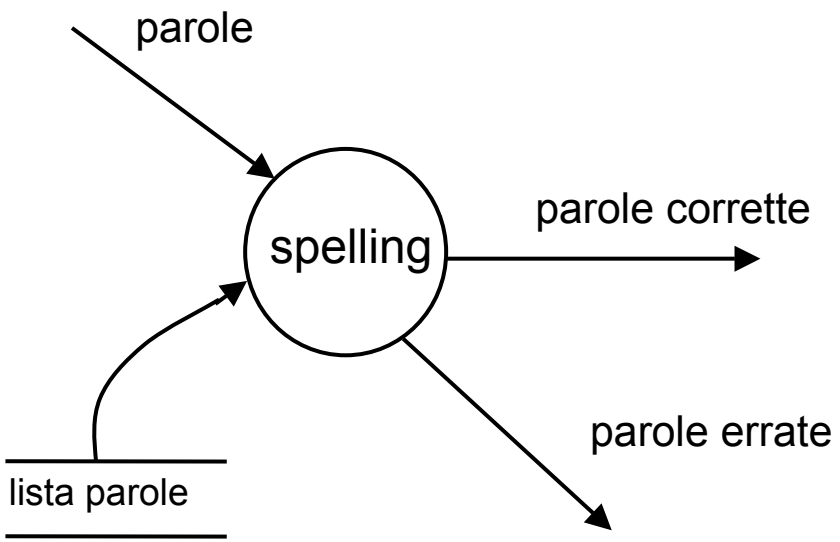
- Il controllo apparirà solo entro la specifica del processo

Cosa non è un flusso dati



Processo

- Un processo è una trasformazione di flussi di dati entranti in flussi di dati uscenti



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

Archivio, terminale

- **Archivio:** documenta una qualunque entità in grado di conservare l'informazione per un uso successivo. E' un deposito temporaneo dei dati
- non facciamo distinzioni: un archivio può essere una base di dati, un file, un insieme di record, un settore di disco, una memoria eeprom.

    - Convenzione sui diagrammi: i flussi che entrano e escono dall'archivio non vengono etichettati se portano pacchetti informativi aventi la stessa composizione dell'archivio.

- **Terminale:** rappresenta un'entità esterna al sistema che interagisce con il medesimo. Sono gli elementi da cui scaturiscono gli ingressi e a cui vengono dirette le uscite.

Criteri per l'analisi

- Identificare tutti gli ingressi e le uscite, ovvero identificare il contesto cui si applica l'analisi.
  - Definire il confine
  - Nel dubbio, allargare i confini
  - Spesso la parte automatizzata è solo una parte dell'applicazione
- Denominare in modo appropriato i flussi di interfaccia con l'esterno
- Denominare i processi in relazione ai loro input/output data flow
- Ignorare i problemi di inizializzazione e terminazione
- Omettere i dettagli banali come percorsi di errore (almeno all'inizio)
- Evitare ogni informazione di controllo
- Tenersi pronti a ricominciare tutto da capo

- L'analisi può comprendere anche parti che non saranno automatizzate

Nomi

Due componenti diversi non possono avere lo stesso nome

Nomi per flussi e archivi

- Dare un nome a tutti i flussi
  Se non si sa dare un nome è segno che il flusso non è ben identificato
- Il nome non deve implicare alcuna forma di elaborazione
  In particolare non deve contenere verbi in forma attiva
- Evitare nomi insignificanti *(dato, informazione)*
- Evitare di raggruppare sotto uno stesso flusso dei flussi che di per sé vanno trattati separatamente (anche se c'è coesione temporale)

Nomi per i processi

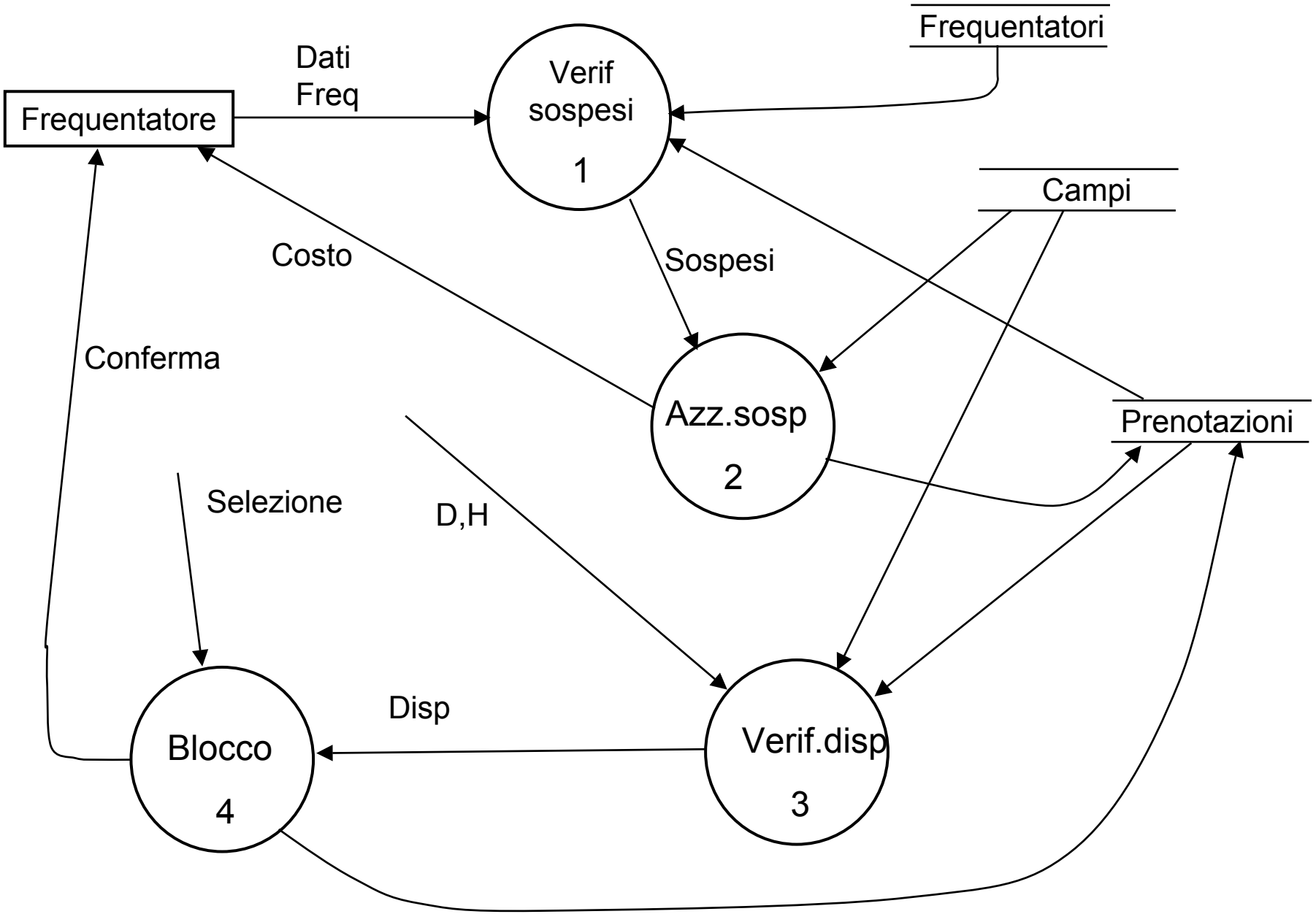
- Possibilmente il nome dovrebbe contenere un verbo in forma attiva e un oggetto ben definiti.
  Se ci sono due verbi, probabilmente, è meglio avere due processi
- Evitare nomi insignificanti *(elaborazione, trattamento)*
- Scomporre se il processo è mal descrivibile
  Non saper dare un nome è indice di cattiva scomposizione

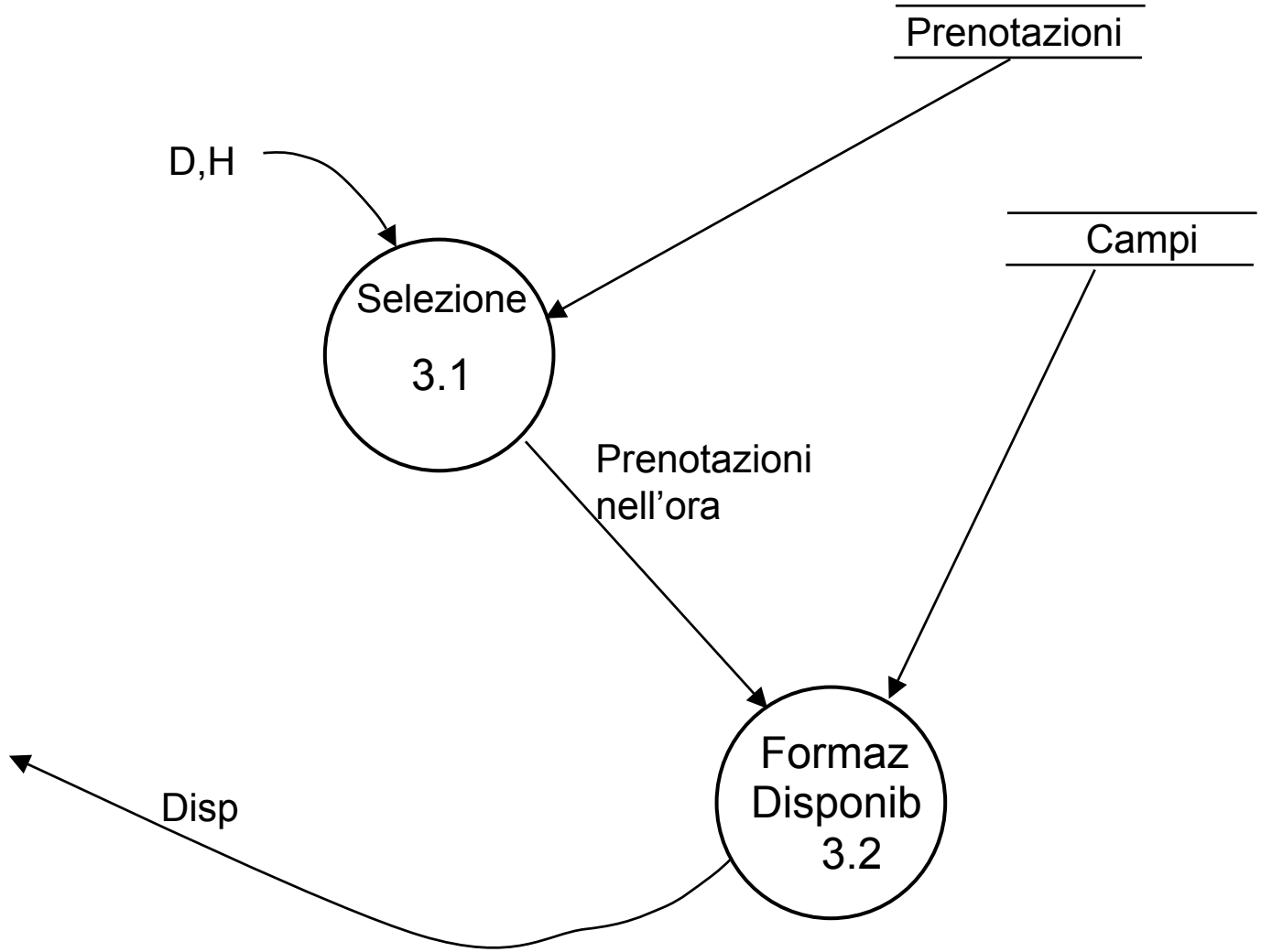
Criteri

- Considerare solo le funzioni di regime. Le inizializzazioni e le terminazioni possono essere trattate (di solito con facilità) in un secondo tempo
- Omettere il trattamento dei banali errori. La regola è mettere un eventuale flusso *respinto/scartato/non accettato* e proseguire con il percorso principale dell'analisi
- Omettere flussi di controllo. I flussi di controllo si scoprono se non si riesce a individuare il tipo di informazione che fluisce su di essi.
- Omettere le informazioni che hanno solo funzione di controllo (esempio *indicatore di inizio di elaborazione*). Il test è questo: un flusso dati che entra in un processo e muore lì è probabilmente informazione di controllo. Un flusso dati vero subisce una trasformazione e si manifesta (trasformato) come flusso dati in uscita.

Scomposizione

- A partire dal diagramma di Contesto (DC) di sviluppa un albero di DFD secondo queste convenzioni
  - Ciascun processo può essere *elementare* o *non elementare*
  - Ad un processo elementare è associata una *Specifica di processo* che ne specifica le funzionalità
  - A ciascun processo non elementare è associato un DFD *figlio* che ne documenta la funzione
- Il DFD figlio eredita il numero d'ordine del processo padre
  - I processi rappresentati sul DFD figlio sono numerati in modo da prendere il numero d'ordine del processo padre, seguito da un numero progressivo.
    - Per esempio: il DFD che esplode il processo 2 viene numerato come 2, i processi su tale DFD vengono numerati come 1.1, 1.2, 1.3,..
  - In sintesi:
    - Il processo "sistema" nel diagramma di contesto viene numerato come 0; il DFD che esplode sistema ha numero d'ordine 0 e i processi su di esso vengono numerati come 1, 2, 3, ..
    - I processi sul diagramma k sono numerati come k.j
    - se il processo k.j non è primitivo, allora esso viene esploso dal DFD k.j


Frequentatori
Dati
Freq
Frequentatore
Verif
sospesi
1
Campi
Costo
Sospesi
Conferma
Azz.sosp
2
Prenotazioni
Selezione
D,H
Disp
Blocco
4
Verif.disp
3


Prenotazioni
D,H
Campi
Selezione
3.1
Prenotazioni
nell'ora
Formaz
Disponib
3.2
Disp

Scomposizione

Diagramma 0

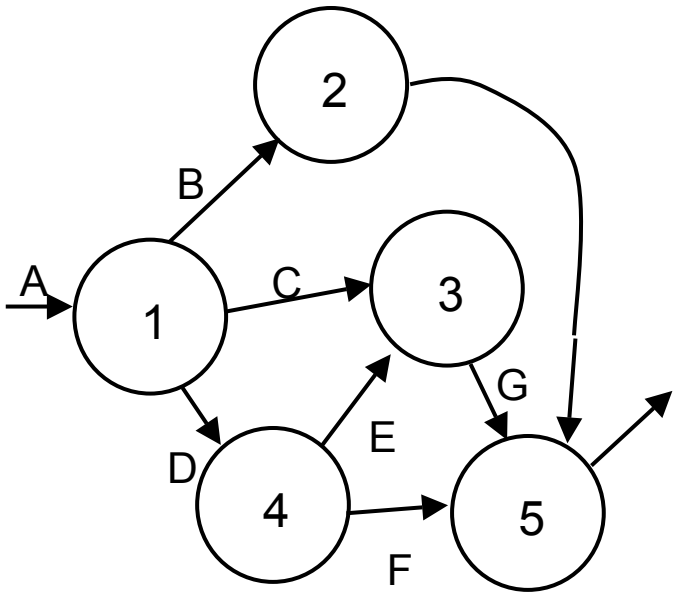
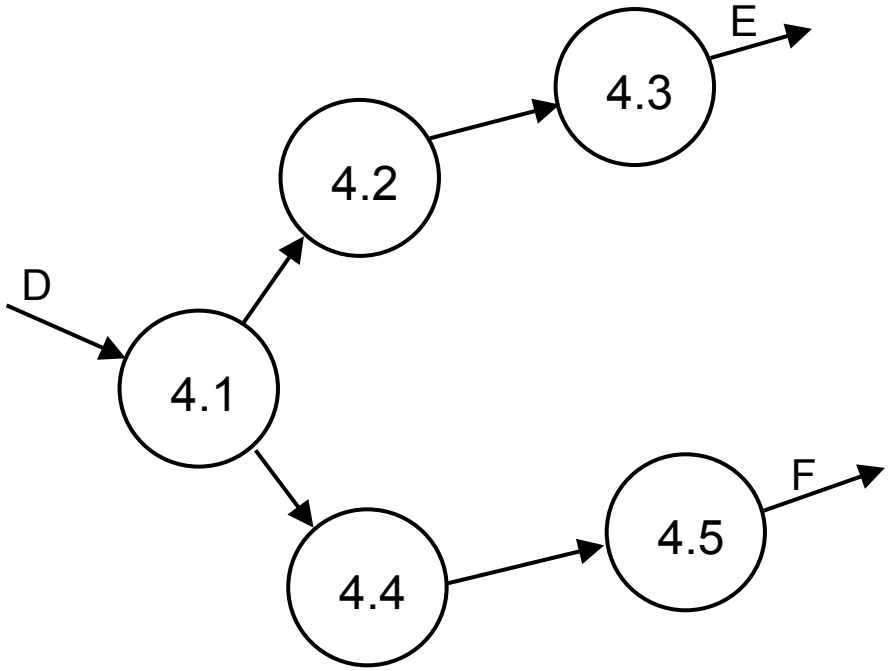


Diagramma 4



Bilanciamento

- Il bilanciamento è la condizione di consistenza sulla conservazione dei flussi nell'esplosione del processo: a ciascun flusso in ingresso/uscita dal processo padre deve corrispondere un flusso nel diagramma figlio.

- Il diagramma k.j si dice bilanciato se tutti i flussi di ingresso (di uscita) del diagramma corrispondono a flussi di ingresso (di uscita) del processo k.

- Un flusso che interessa il processo k.j sul diagramma k può essere scomposto in più flussi componenti al momento in cui viene riportato sul diagramma k.j (decomposizione parallela)

Criteri pratici

- Il numero di palle che massimizza la leggibilità è 7 (+/- 2)

- I processi che stanno in uno stesso DFD devono stare ad un livello di astrazione omogeneo. Se ciò non accade conviene effettuare un ripartizionamento topologico:
  - Esplodere i processi troppo aggregati, fino a raggiungere un livello di dettaglio omogeneo
  - Riaccorpare processi diversi, in modo da ottenere livelli omogenei

- La scomposizione di un processo può terminare:
  - se può essere specificato in una pagina di testo
  - se il processo presenta un solo flusso di ingresso e un solo flusso di uscita

Specifica dei processi

- A ciascun processo è associata una specifica (Pspec) che documenta il comportamento funzionale del processo specificando le trasformazioni che generano i flussi di uscita da quelli di ingresso

- Linguaggi tipici di specifica:
  - Linguaggio naturale
  - Linguaggio naturale strutturato (pseudo codice)
  - Diagrammi di flusso

Dizionario dei dati

- Raccoglie e descrive in modo rigoroso (?) tutte le entità presenti nei DFD

- Per ciascun flusso il dizionario riporta:
  - la struttura del pacchetto informativo
  - il significato ed eventuali altre informazioni relative a ciascun campo

- Per ciascun archivio il dizionario riporta
  - la struttura dei record che lo compongono
  - l'eventuale meccanismo di accesso
  - (come allegato) il diagramma di struttura di archivi molto complessi, come le basi di dati

- Diversamente dai DFD la struttura del dizionario dati non è ben definita. E' lasciata alle scelte dell'analista.

Case tools

- Case sta per *Computer aided software engineering:* sistema software di aiuto allo sviluppo (e alla manutenzione)
- Dovrebbero servire a fare aumentare la produttività dell'analista-progettista-programmatore
- Chi usa un case deve conoscere la metodologia
- Qualcosa di più di un "tecnigrafo": un case tool è soprattutto utile per evitare incongruenze, dimenticanze, etc.:
  - Costruzione dello scheletro delle PSpec
  - Modifica di un nome: lo strumento la riporta in tutti i luoghi in cui il nome è usato
  - Modifica della struttura di un archivio: segnalazione di eventuali incongruenze con le manipolazioni effettuate da qualche processo
  - Traccia delle modifiche
  - etc.

Case e analisi strutturata

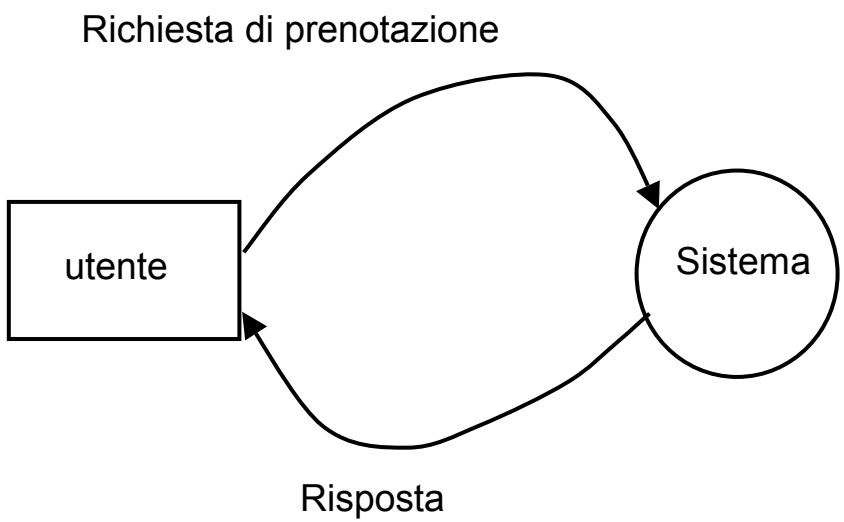
- L'analisi strutturata è stata la prima metodologia ad essere portata su case.
- Quasi tutti i case contengono l'analisi e il progetto strutturati. Spesso con simboli diversi.
- A parte la facilità di disegno (non sempre apprezzabile!), il grande vantaggio sta nella costruzione automatica del Dizionario dei dati: per ogni nome viene inserita la corrispondente voce, l'operatore è invitato a fornire le informazioni che il sistema non ottiene in modo automatico dal disegno.
- Alcuni case forniscono uno scheletro di codice C dei processi.

Esempio - Gestione prenotazioni albergo

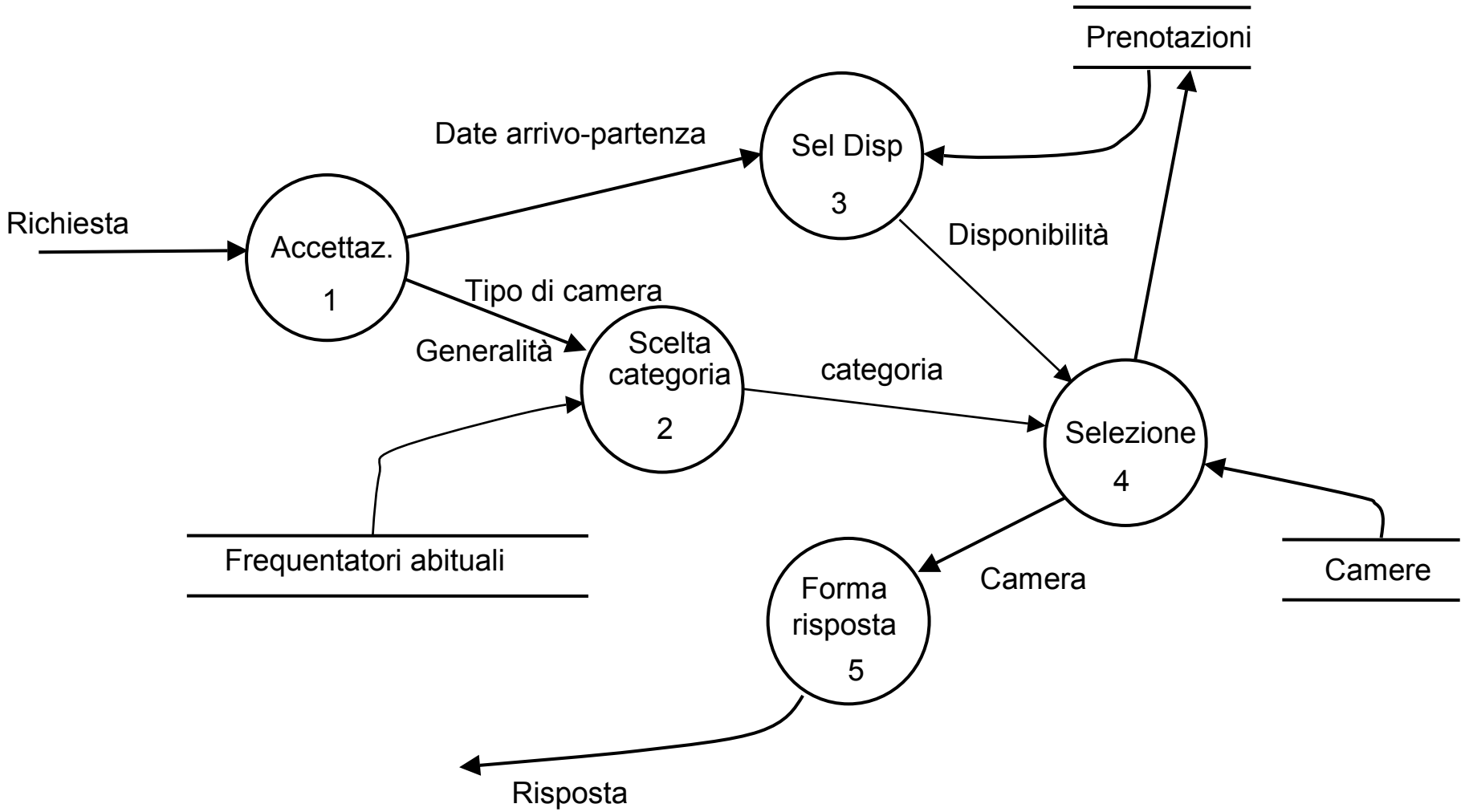
- L'albergo riceve richieste di prenotazione. Una richiesta contiene:
  - generalità del cliente
  - notti per cui si intende prenotare
- L'albergo risponde (la risposta porta al suo interno la conferma o la notifica di non avvenuta prenotazione)
- L'albergo ha camere di differente categoria. Nella richiesta viene indicato il tipo di camera che si traduce in una categoria interna
  - Viene assegnata una camera disponibile di pari livello o immediatamente inferiore.
  - Per i frequentatori abituali la categoria viene dedotta dalla base di dati (non è necessario che diano il tipo)

Esempio - Gestione prenotazioni albergo

Diagramma di contesto



Esempio - Gestione prenotazioni albergo (cont)



Progettazione strutturata

- Nel modello a cascata la progettazione segue la specifica dei requisiti, il cui prodotto assume in ingresso per produrre in uscita un documento di progetto. Quest'ultimo descrive il *partizionamento* del sistema e l'*organizzazione* delle sue parti.
- Scopo della progettazione strutturata è la determinazione dei componenti (moduli) e delle relative interfacce: "quali componenti, connessi in quale modo, risolveranno un problema ben specificato
- Ovviamente si deve mirare a ottimizzare questi aspetti:
  - efficienza
  - affidabilità
  - manutenibilità
  - flessibilità
  - Etc..

  Il contesto determina quali aspetti hanno maggior o minor rilevanza
- la regola fondamentale è "**divide et impera"**

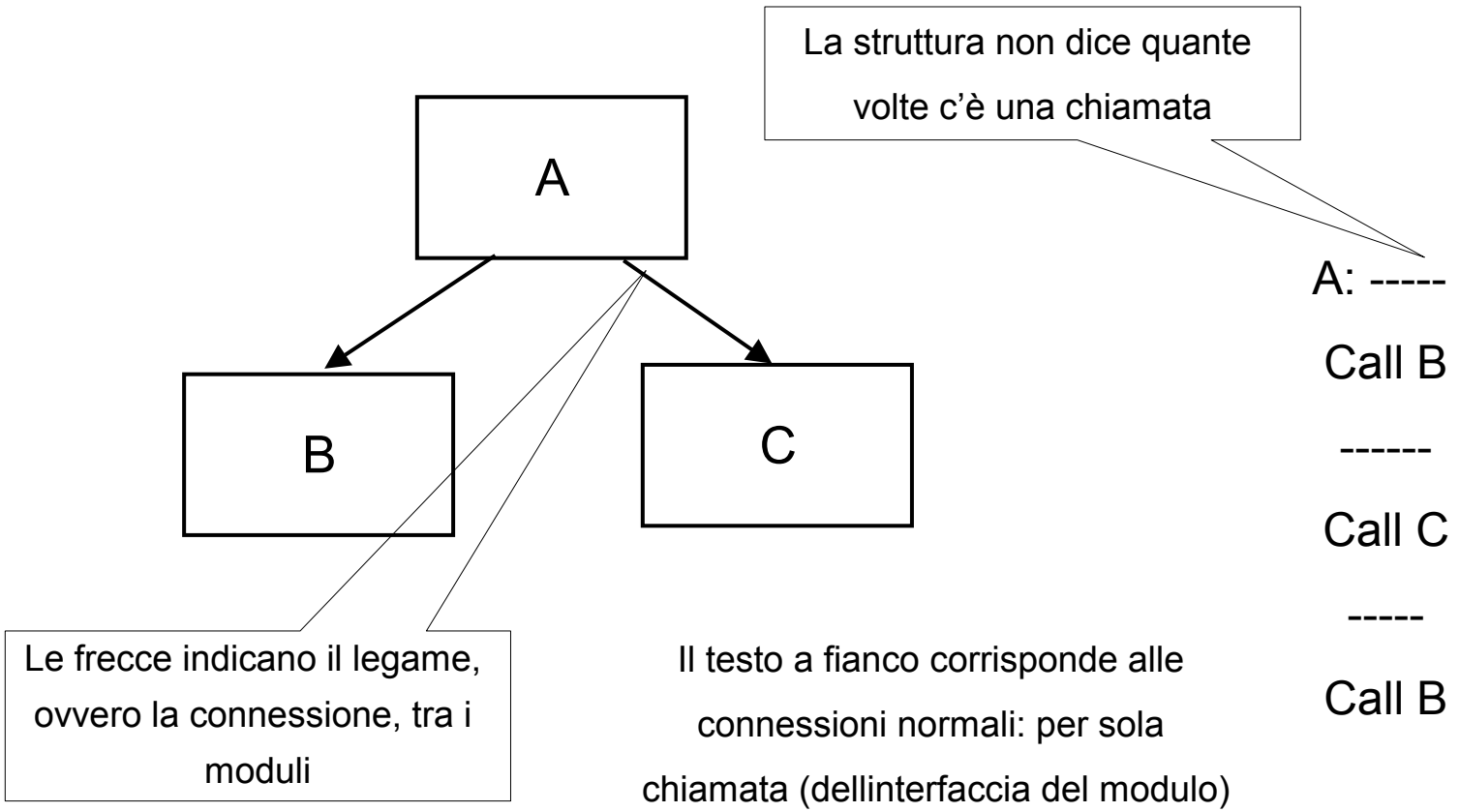
...Progettazione strutturata

Si avvale di:

- **una notazione,** *la carta strutturata*, con cui si documenta la strutturazione del sistema in moduli. E' il formalismo con cui si presenta il prodotto del progetto
- **due tecniche,** *l'analisi delle trasformazioni e l'analisi delle transazioni,* attraverso le quali si produce la carta strutturata a partire dalle specifiche strutturate
- **due criteri euristici,** *l'accoppiamento e la coesione,* per valutare la bonta del progetto ovvero per confrontare due progetti derivati dalle stesse specifiche

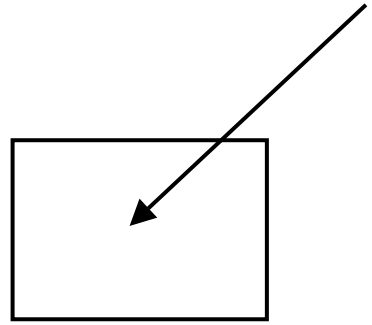
Carta strutturata

- Lo schema mostra la relazione che c'è tra il modulo A, il chiamante, e i moduli chiamati (B e C)



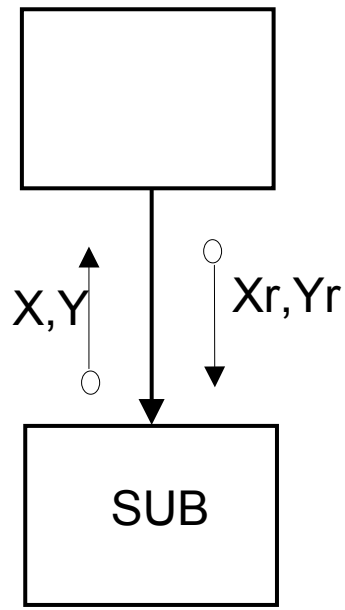
… Carta strutturata

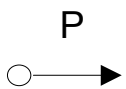
Connessione patologica: accesso a una variabile interna, salto all'interno



Passaggio dei parametri.

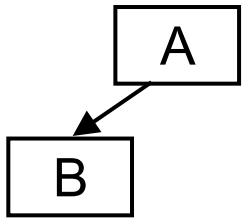
CALL SUB(X,Y,Xr, Yr)





Indica il verso in cui viene passato il parametro P

Accoppiamento



- Misura il grado di interdipendenza tra i moduli
  - Quanto B influenza A
  - Qual è la probabilità che modificando B si debba modificare A?
  - Quanto dobbiamo sapere di B per capire A?
- Non esiste una metrica oggettiva: l'accoppiamento è tanto maggiore quanto maggiore è il grado di conoscenza che si deve avere di un modulo per intervenire sull'altro.
- Elementi che influenzano l'accoppiamento:
  - Tipo di interconnessione tra i moduli (normale o patologica)
  - Complessità dell'interfaccia
    - *Coeteris paribus, un'interfaccia con due parametri si capisce meglio di una con 200*
    - *Uso coerente dei nomi (per il programma i nomi sono irrilevanti per le persone no: non ha senso che la funzione* `FIND(...)` *restituisca* `F` *se trova quello che cerca*
  - Tipo di informazione che passa attraverso l'interfaccia
    - Dati: basso accoppiamento
    - Controllo: alto accoppiamento
  - Momento dell'accoppiamento: più è differito tanto è minore
    - Due moduli accoppiati dal linker hanno minor accoppiamento di due accoppiati in compilazione

...Accoppiamento

```
for (i=0; i<80; i++)
```

```
#define  MAXLINE 80
for (i=0; i<MAXLINE; i++)
```

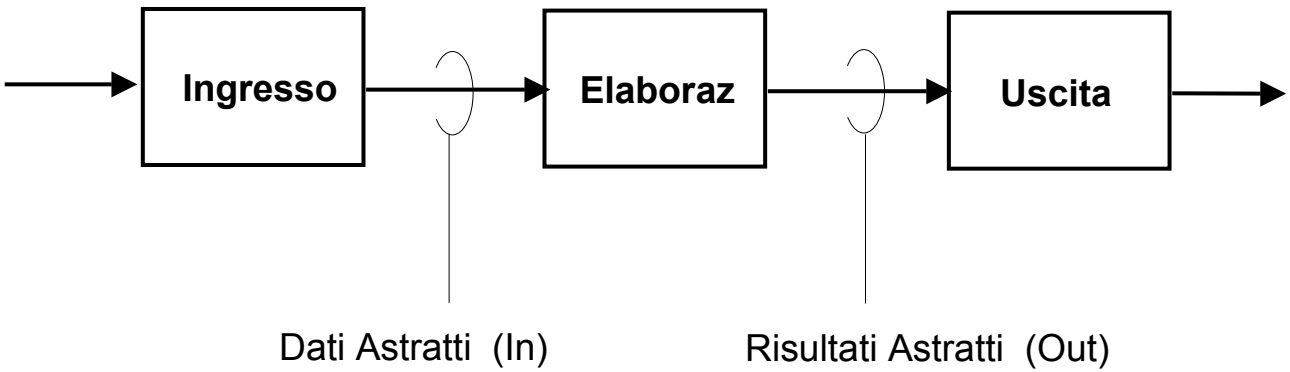
```
readline(line, Maxline)
char* c;
int Maxline;
for (i=0; i<Maxline; i++)
```

Accoppiamento decrescente

Coesione

- Dà conto di quanto forti siano le relazioni tra gli elementi che compongono un modulo
- Classificazione della coesione:
  - **incidentale:** si raccolgono in un modulo un gruppo di statement ripetuto più volte. Tipica della modularizzazione a codifica avvenuta
  - **logica:** Es: un modulo che acquisisce tutti gli input del sistema. Operazioni analoghe messe assieme anche se operano su dati assulamente non coesivi
  - **temporale:** Es: si raccoglie in un modulo tutte le operazioni di inizializzazione, anche se queste sono scorrelate
  - **procedurale:** Es: moduli costruiti a partire da flow-chart. Anche se i moduli hanno un solo ingresso e una sola uscita è probabile che trattino dati assolutamente scorrelati
  - **di comunicazione:** Es: moduli definiti a partire da un data flow con le unità raccolte che operano sugli stessi input e/o generano gli stessi output
  - **sequenziale:** Quando l'output della prima costituisce l'input della seconda
  - **funzionale:** Quando tutte le entità in modulo concorrono in maniera essenziale alla realizzazione di una medesima funzionalità
- Obiettivo della progettazione strutturata: organizzare il sistema in modo che sia minimo l'accoppiamento tra i moduli e in modo che sia massima la loro coesione.

Struttura tipica di sistema di elaborazione

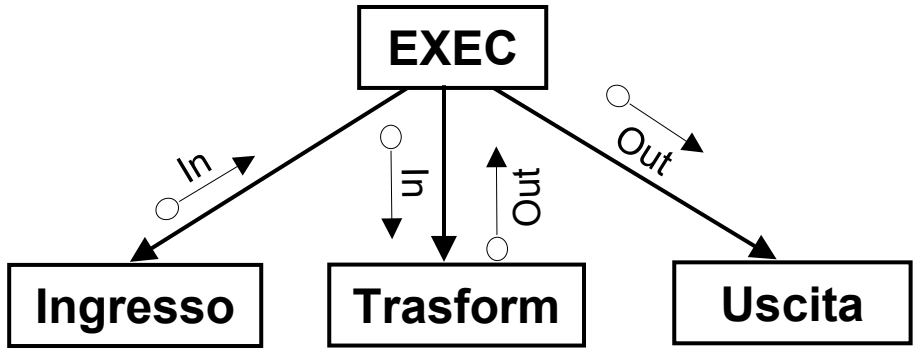


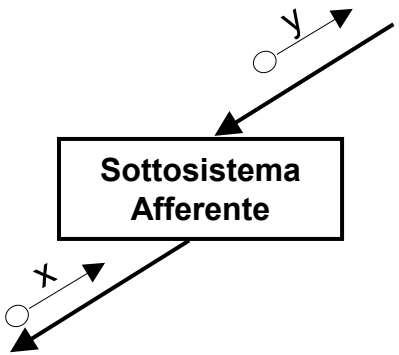
L'analisi trasformazionale riflette lo schema:

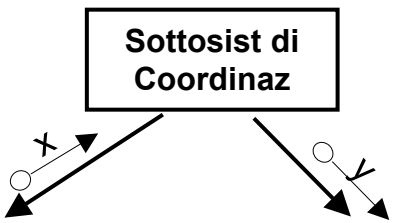
- Un modulo afferente: Ingresso
- Un modulo di trasformazione: Elaboraz.
- Un modulo efferente: Uscita
- Un modulo di coordinamento o controllo: raccoglie i dati astratti dal modulo afferente, li fa elaborare dal modulo di trasformazione, manda i risultati al modulo efferente

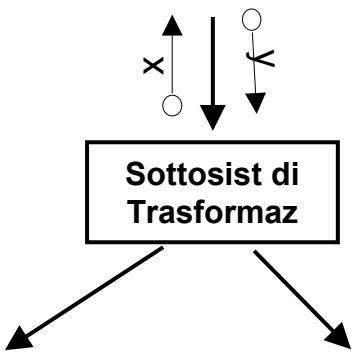
Organizzazione

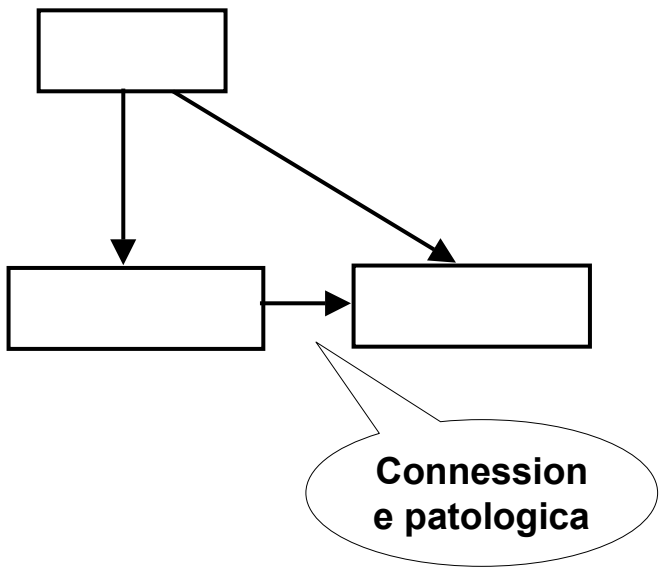
Schema corrispondente











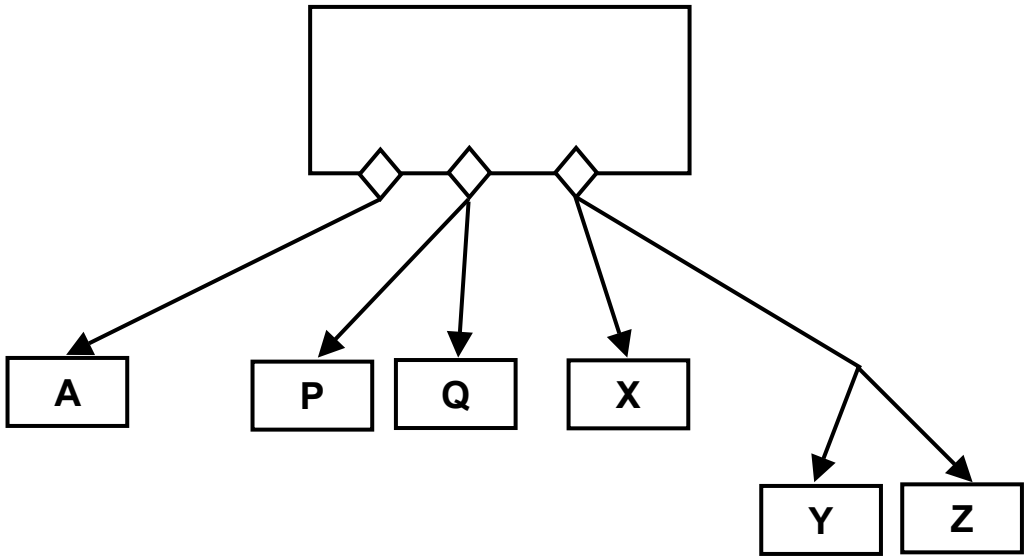
Rappresentazione delle strutture di controllo

Alternativa

If …. then A;

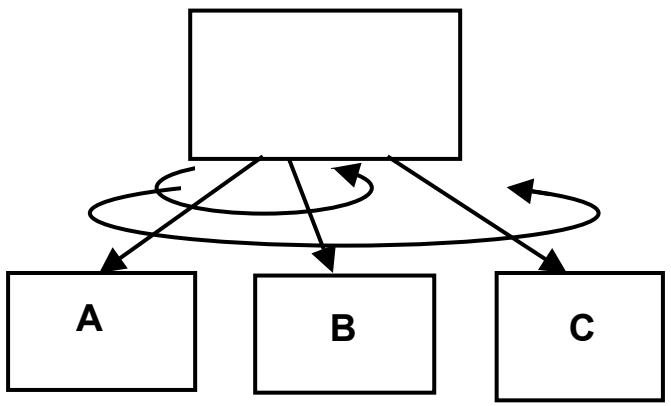
if ….. then P else Q;

if …. Then X else {Y;Z};



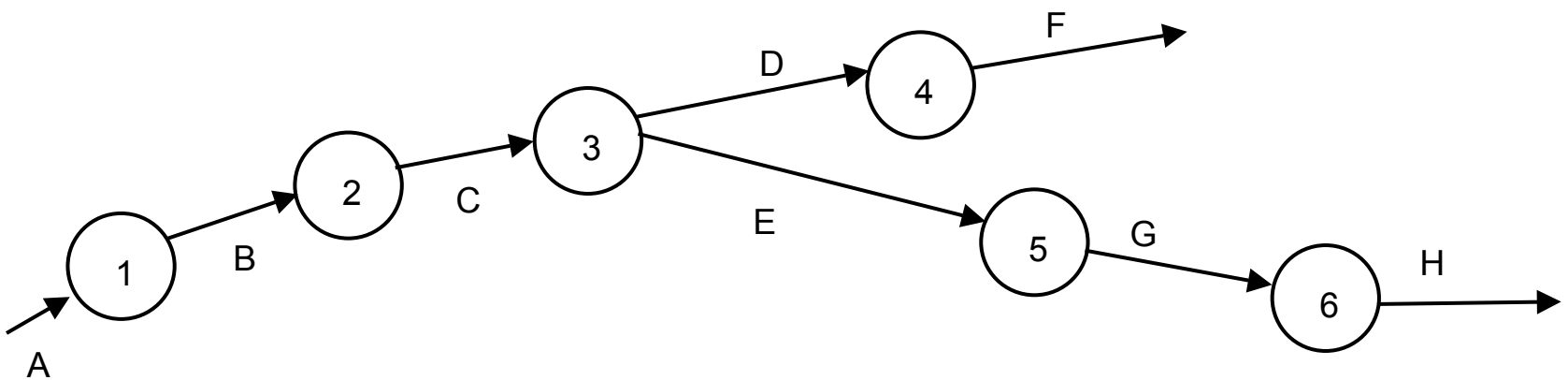
Cicli

```
while B
  {
   while B2
      {A;B};
   };
```

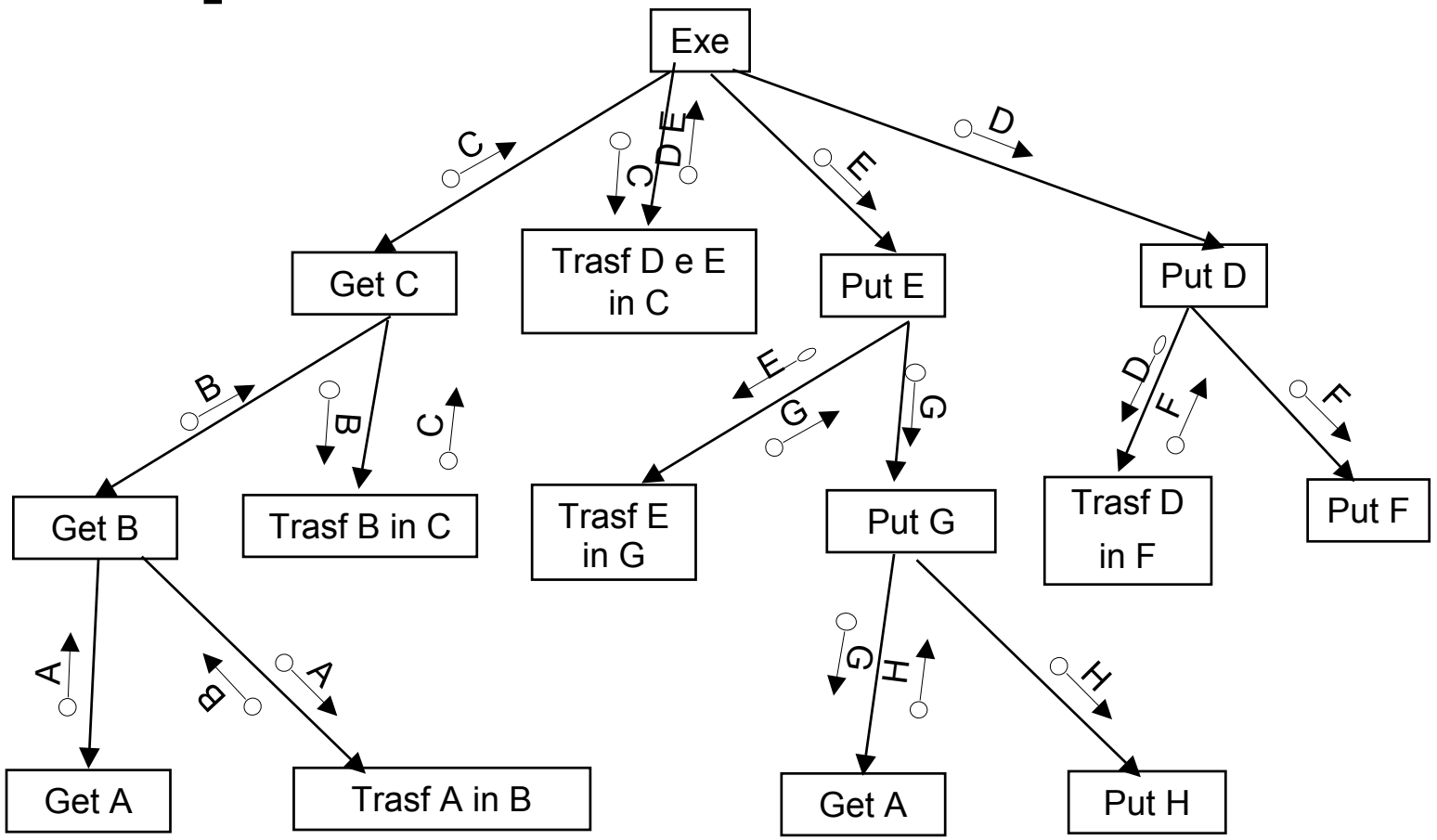


Costruzione della CS

- Se si prende un DFD si osservano processi che portano informazione in ingresso, che l'informazione diventa via più astratta passando attraverso trasformazioni fino a un nodo di trasformazione centrale, a partire dal quale l'informazione diventa sempre meno astratta
- Figuratamente si tratta di prelevare il nodo centrale e *tirare su il grafo*. Ciò porta ad avere:
  - uno o più rami afferenti
  - uno o più rami efferenti
  - la trasformazione centrale



Esempio


Exe
C
C
D
E
E
D
Get C
Trasf D e E
in C
Put E
Put D
B
B
C
F
G
G
D
F
Get B
Trasf B in C
Trasf E
in G
Put G
Trasf D
in F
Put F
A
B
A
G
H
H
Get A
Trasf A in B
Get A
Put H

Fattorizzazione di una gerarchia di DFD

Tecnica top-down a partire dal diagramma di livello 0

- Sul diagramma di livello 0 vengono individuati elementi afferenti, efferenti e le trasformazioni centrali
- Si definisce l'Executive radice (il sistema)
  - Per ogni ramo afferente viene definito un modulo subordinato afferente
  - Per le trasformazioni centrale viene individuato un modulo di trasformazione
  - Per ogni ramo efferente viene definito un modulo subordinato efferente

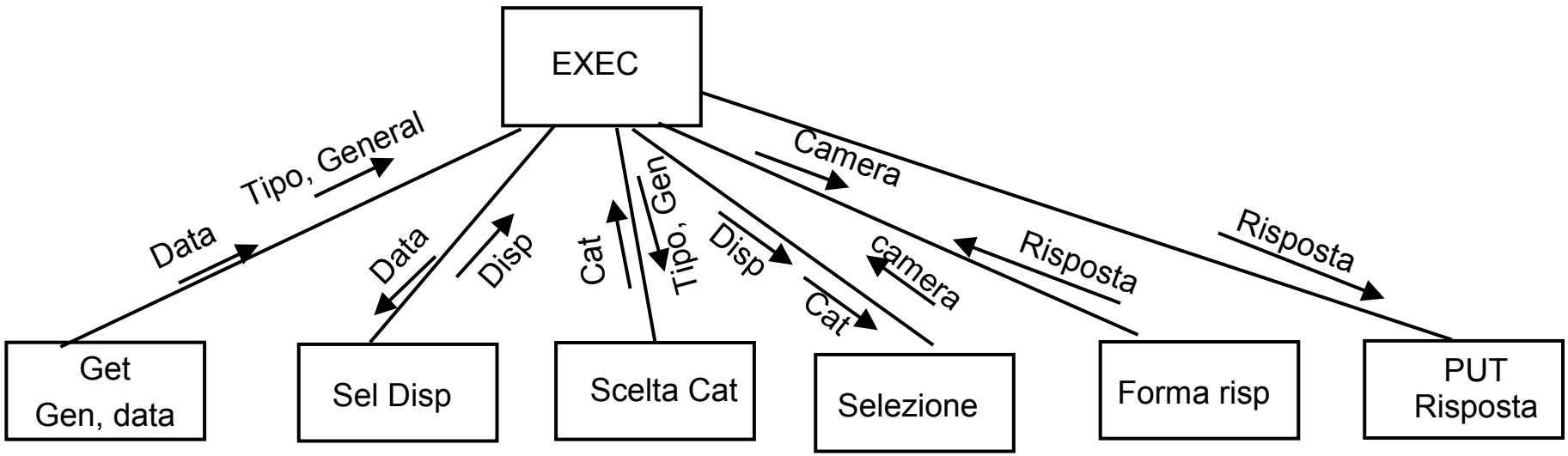
- Fattorizzazione di un modulo di trasformazione
  - Se il modulo è primitivo la fattorizzazione si arresta
  - Se il modulo non è primitivo si considera il DFD figlio. Su questo DFD si opera come sopra, salvo sostituire il modulo fittizio Executive con il modulo di trasformazione che si sta considerando

… Fattorizzazione di una gerarchia di DFD

- Fattorizzazione di un modulo afferente
  - Per costruzione un modulo afferente corrisponde a un ramo afferente sul DFD
  - Per costruzione i dati al massimo livello di astrazione sono quelli in uscita al ramo
  - Per ciascun ramo, i processi che producono in uscita uno o più dei dati in uscita dal ramo sono trasformazioni a cui viene fatto corrispondere un modulo di trasformazione
  - Si aggiungo quindi i moduli di tipo afferente che portano i dati ai rispettivi moduli di trasformazione
  - *In conclusione la strutturazione di un modulo afefrente produce moduli di trasformazione e moduli afferenti, NON moduli efferenti*
- Fattorizzazione di un modulo afferente
  - Si procede in modo duale rispetto al caso dei moduli afferenti

Esempio - Gestione prenotazioni albergo (cont)

Carta di struttura



Centri di transazione