SENZA TITOLO

Macchine a stati in UML

- Sono una versione estesa (derivante dalle StateChart [Harel]) delle normali macchine a stati (*finite state machines/finite automata*)
- Modellano il comportamento dinamico degli oggetti di una data classe (tutte le possibili storie)
  - Sono una delle possibili *viste dinamiche*
  - Si rappresentano attraverso i *diagrammi di stato* (stati e transizioni)
- Una macchina a stati descrive un oggetto come un'entità isolata che comunica con il resto del mondo attraverso il riconoscimento del verificarsi di eventi, rispondendo ad essi

Macchine a stati

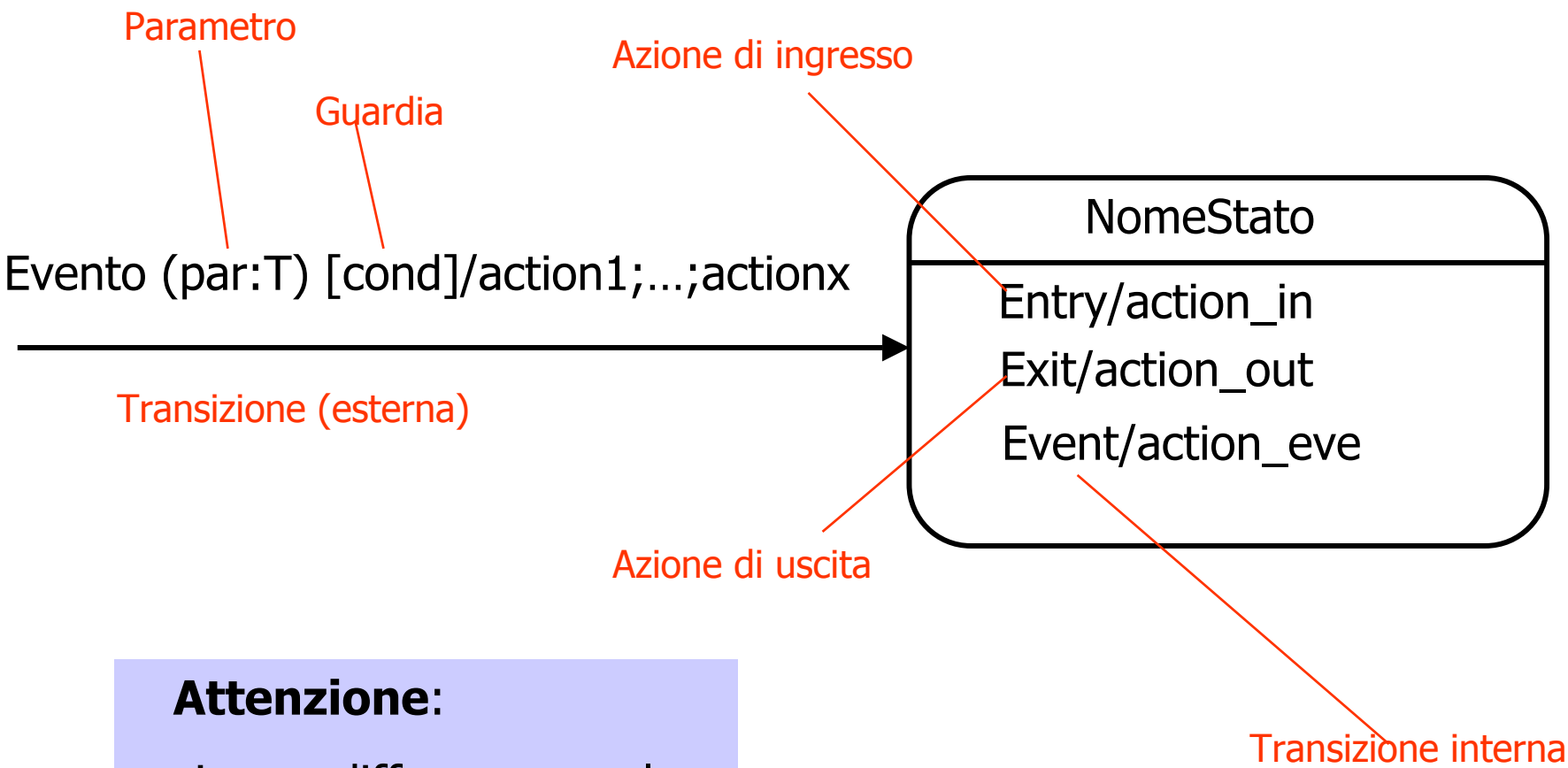
- Non mostrano le interazioni con gli altri oggetti (per queste si deve ricorrere ad altre viste dinamiche: diagrammi di sequenza e di collaborazione)
- La risposta agli eventi è funzione dello stato in cui si trova la macchina: tutti gli oggetti della classe modellata, che si trovino in un dato stato, effettuano la stessa azione al riconoscimento di un dato ingresso

- Indicate per modellare oggetti in cui sia prevalente la funzione di controllo

Estensioni

- Le macchine a stati convenzionali hanno molte limitazioni. In particolare
  - Cambiamenti di stato solo in base agli ingressi
  - Una macchina può influenzare un'altra solo attraverso le sue uscite che la seconda vede come ingressi
- Macchine a stati in UML
  - Cambiamenti di stato anche in base a vari tipi di *eventi*. Questi possono rappresentare il verificarsi di generiche condizioni (espresse sia sugli stati delle macchine che sui valori delle variabili)
    - l'interazione tra le macchine può avvenire attraverso lo scambio di informazioni generiche
  - Composizione degli stati (superstati)

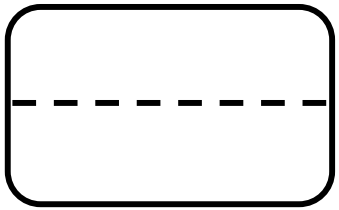
Notazione



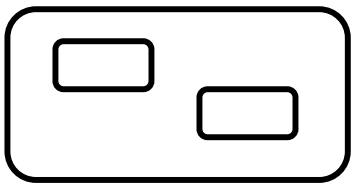
Attenzione:

ci sono differenze con le macchine a stati convenzionali

Stati composti



Stato composto concorrente

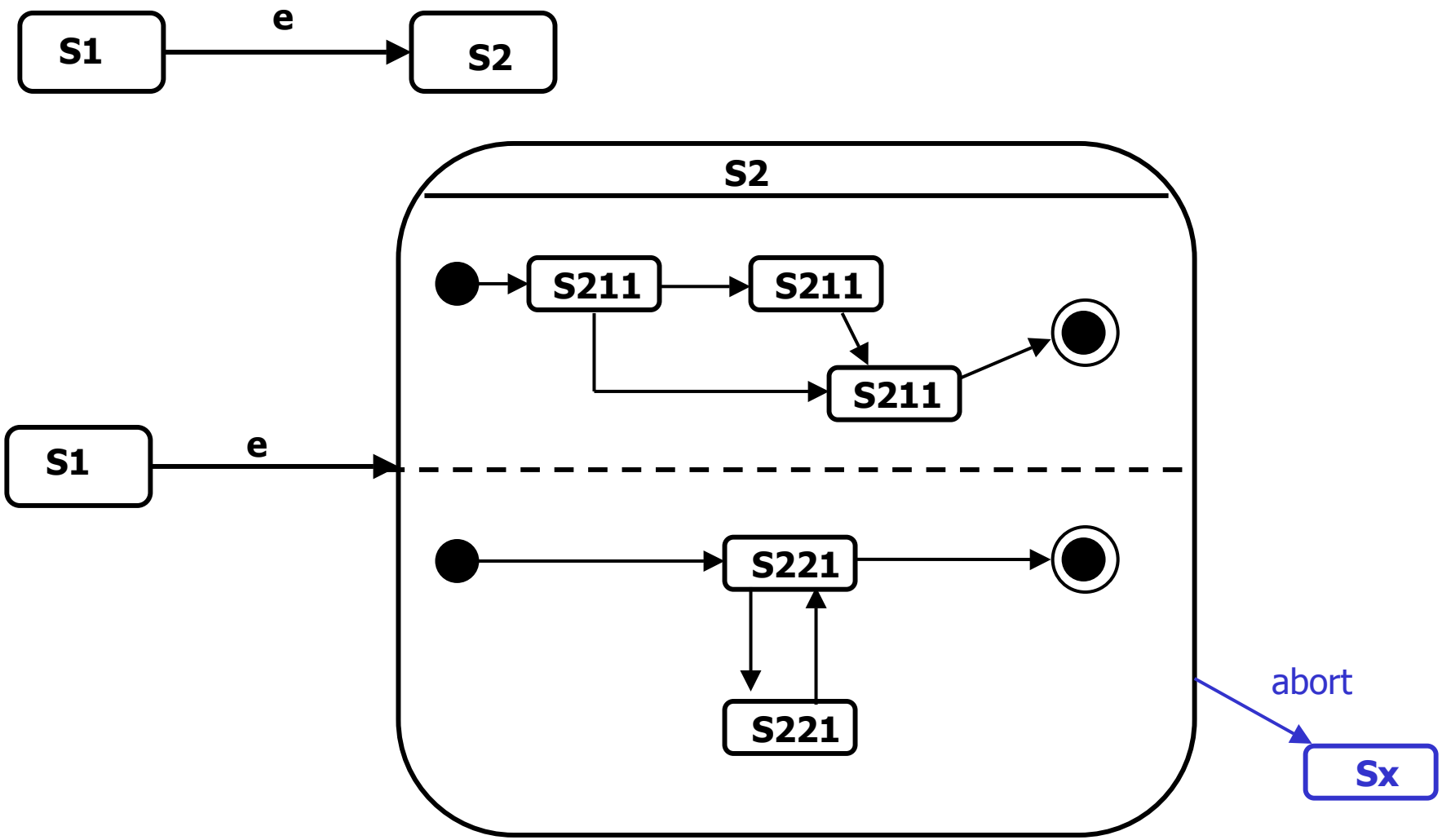


Stato composto sequenziale

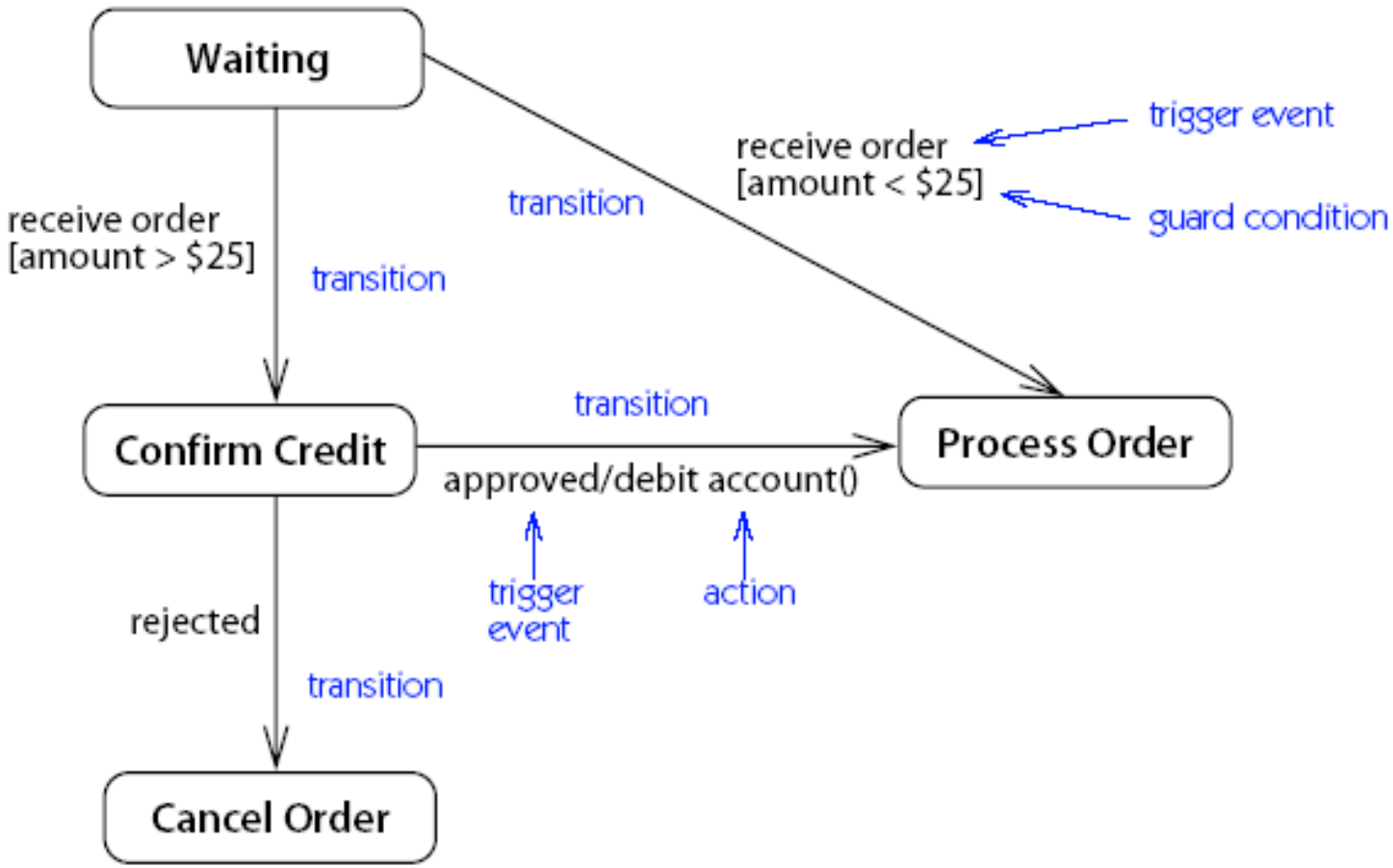
Stato iniziale (pseudo stato): indica il punto di partenza

Stato finale: indica il completamento dell'attività (dello stato che lo include)

Stati composti



Diagrammi


Waiting
receive order
[amount > $25]
transition
transition
receive order
[amount < $25]
trigger event
guard condition
Confirm Credit
transition
Process Order
approved/debit account()
trigger
event
action
rejected
transition
Cancel Order

Eventi

- Un evento è qualcosa che accade ad un certo istante di tempo
- Un evento non ha durata
- UML prevede quattro tipi di eventi
  - call
  - change
  - signal
  - time

La loro effettiva implementazione dipende dal linguaggio di programmazione/ambiente

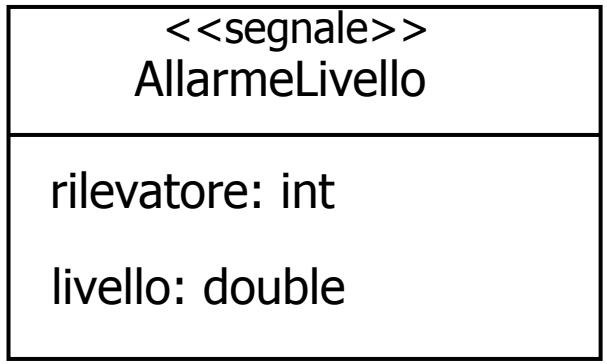
(UML è un linguaggio di modellazione!!!)

(eventi) Segnali

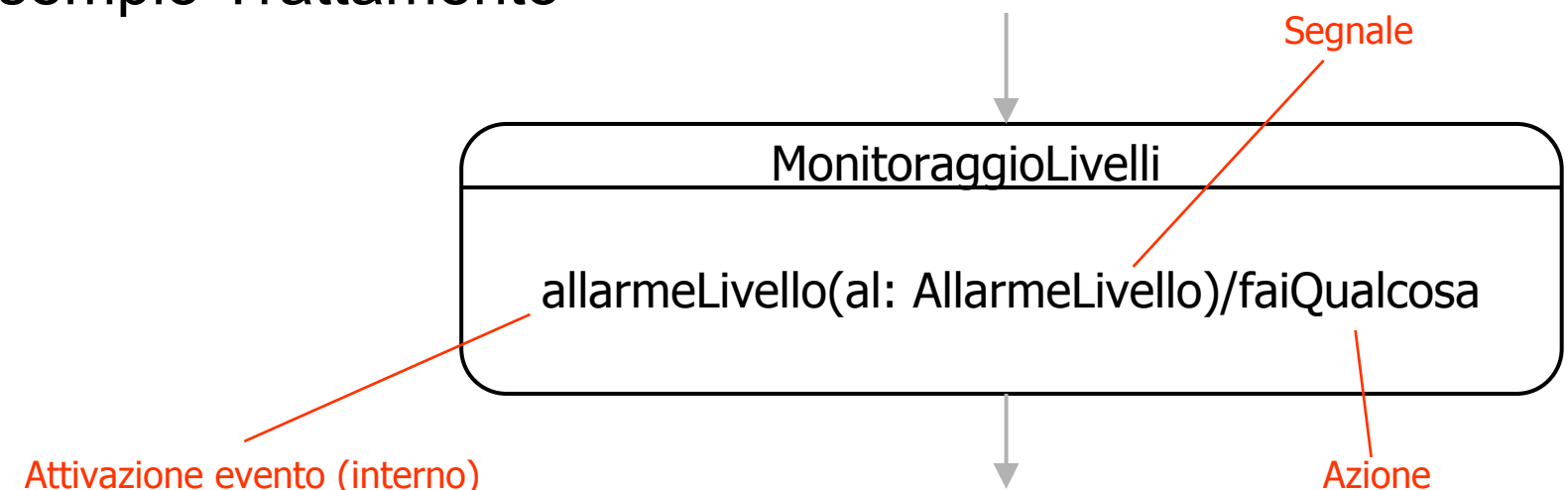
- Un segnale è un pacchetto di informazioni inviato in modo *asincrono* da un oggetto a un altro
- La ricezione del segnale è l'evento per l'oggetto ricevente
- Comunicazione (asincrona) *unidirezionale* (il trasmittente non attende la risposta)
- Il modo in cui si implementa dipende dal linguaggio.
  - Di norma il paradigma è event/listner (publish/subscribe)
- Esempi:
  - Evento pressione di un mouse
  - Evento notificato a un sottoscrittore

(eventi) Segnali

- Modellazione



- Esempio Trattamento



(eventi) Cambiamenti

- Eventi che corrispondono al soddisfacimento di una condizione booleana
- Attivati quando la condizione diventa vera
  - Formalmente si esprimono così: **`when(B)/action`**
- Presuppongono il test continuo della condizione
- Diversi dalle guardie
  - Queste vengono valutate dal ricevente solo quando accade l'evento: la transizione ha luogo solo se la guardia è vera; la guardia non viene più rivalutata
  - I cambiamenti vengono valutati con continuità (si possono fare trucchi per evitare questo spreco)
- Usarli sono se non c'è alternativa

(eventi) Call

- E' il metodo usuale con comunicano gli oggetti.
  - Per il chiamante non c'è differenza rispetto alla usuale chiamata di metodi
  - Il ricevente invece implementa il metodo come una transizione di stato:
    - tratta il metodo come un evento e decide sulla transizione
    - restituisce il controllo al chiamante
    - diversamente dalle normali chiamate può continuare la sua esecuzione in parallelo con il chiamante

(eventi) Call

- Nell'implementazione consentono di modellare il campionamento di ingressi e condizioni

```
public class UnaClasse {
  private Stato S;
      // … altre componenti
  public Execute(Parametro input){
   switch(c) {
      case s1: valutazione trigger/azione; break;
      case s2:
      ...
      case sn:
   }
}
```

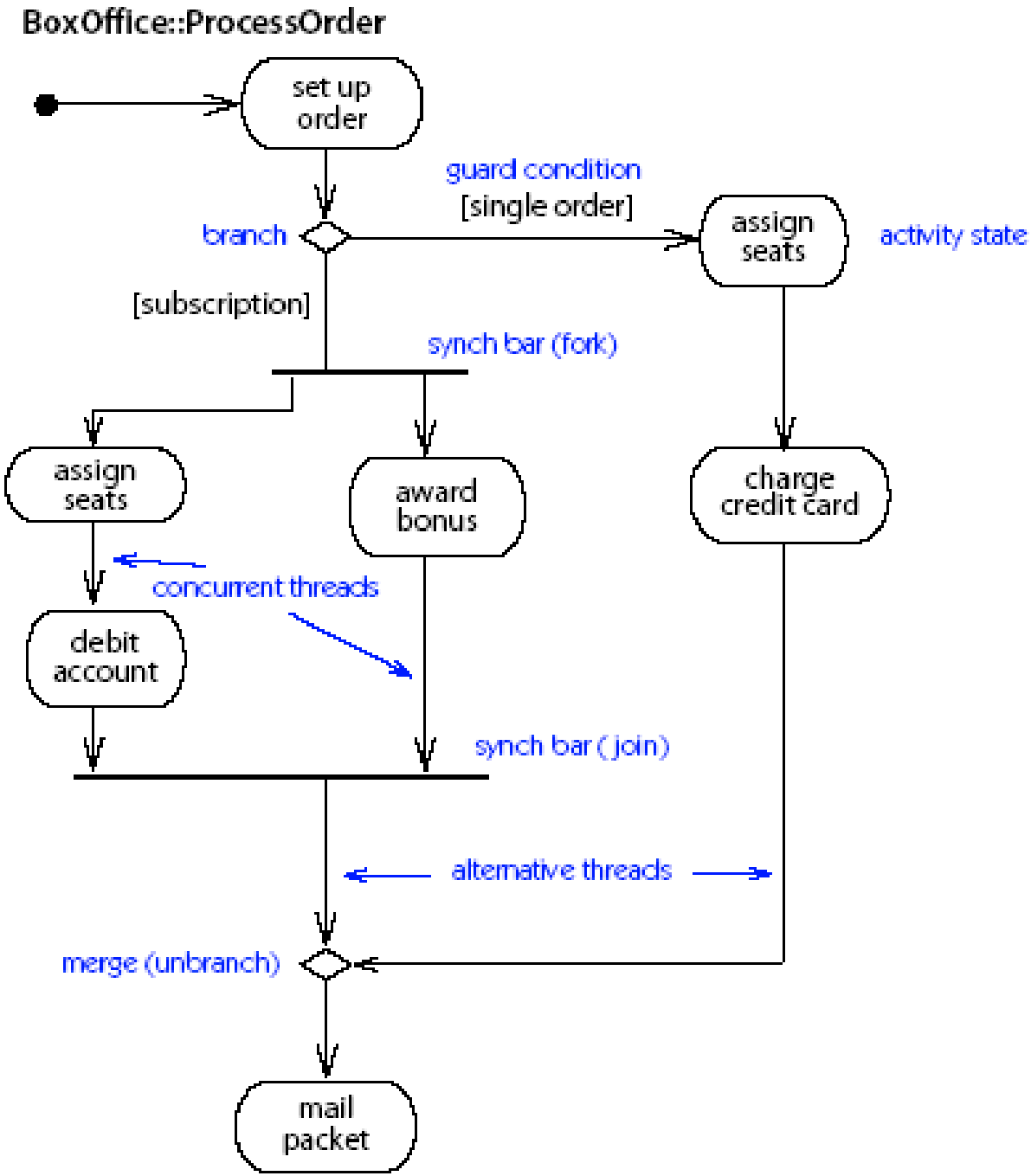
(eventi) Time

- Può essere specificato:
  - in modo assoluto (ora del giorno)
  - in modo relativo (delta trascorso da un dato evento)

- Nel modello concettuale gli eventi temporali possono essere considerati come eventi provenienti dall'universo
- Nel modello implementativo essi sono segnali generati da un qualche oggetto applicativo o dal sistema operativo

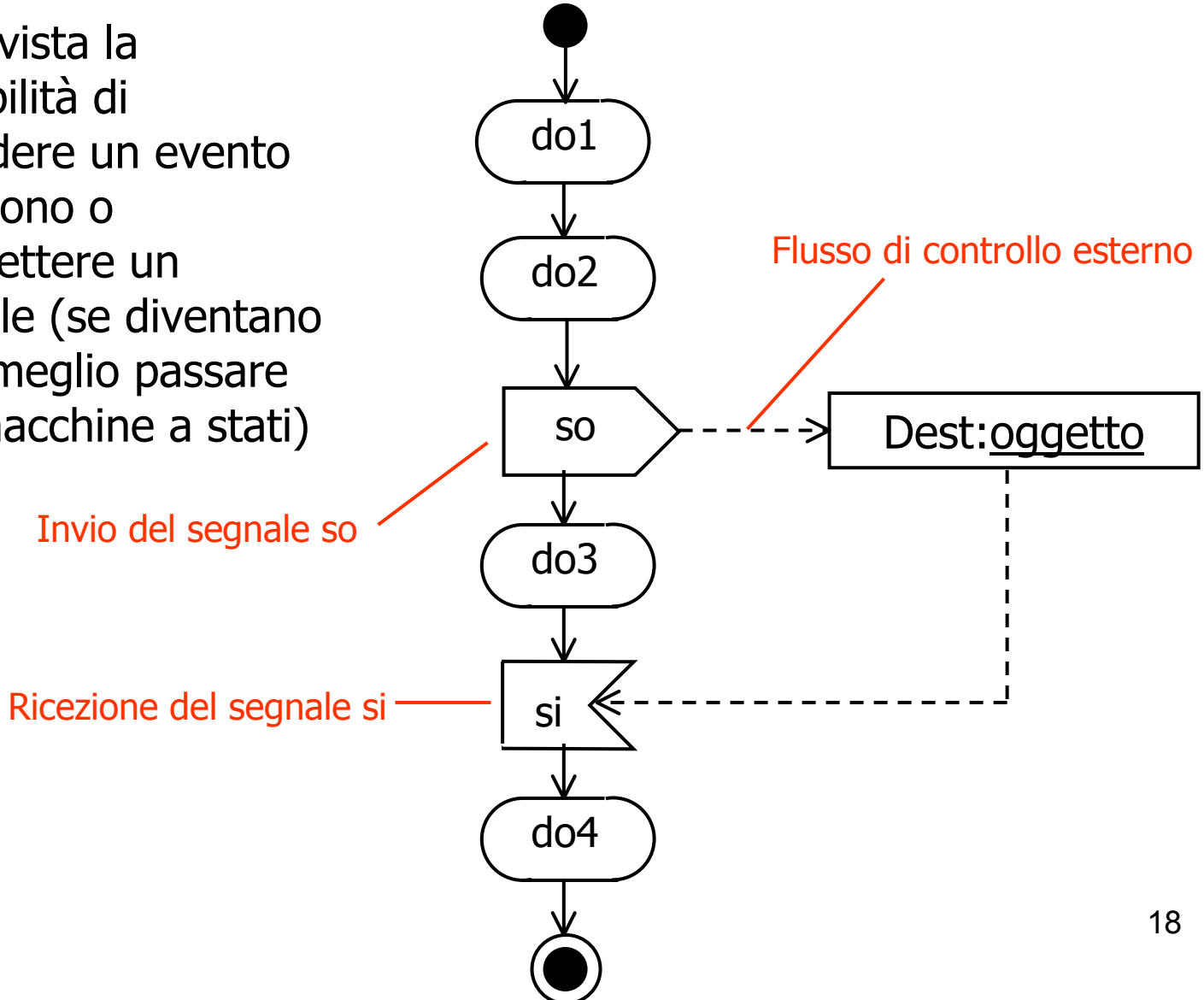
Diagrammi di attività

- Sono i *diagrammi di flusso OO*
  - usati per modellare processi (di elaborazione) come un insieme di attività e di transizioni tra queste attività
- Sono essi stessi delle macchine a stati:
  - gli stati rappresentano esecuzione di elaborazioni (non stati di un oggetto); sono detti *stati di attività*
  - uno stato di attività rappresenta l'esecuzione di uno statement o una procedura
  - si assume che l'elaborazione *proceda senza essere sollecitata da eventi*: l'uscita da uno stato di attività corrisponde al completamento dello statement/procedura, non a un evento (si parla di transione di completamento)


BoxOffice::ProcessOrder
set up order
guard condition
[single order]
branch
assign seats
activity state
[subscription]
synch bar (fork)
assign seats
award bonus
charge credit card
concurrent threads
debit account
synch bar (join)
alternative threads
merge (unbranch)
mail packet

Segnali/eventi asincroni

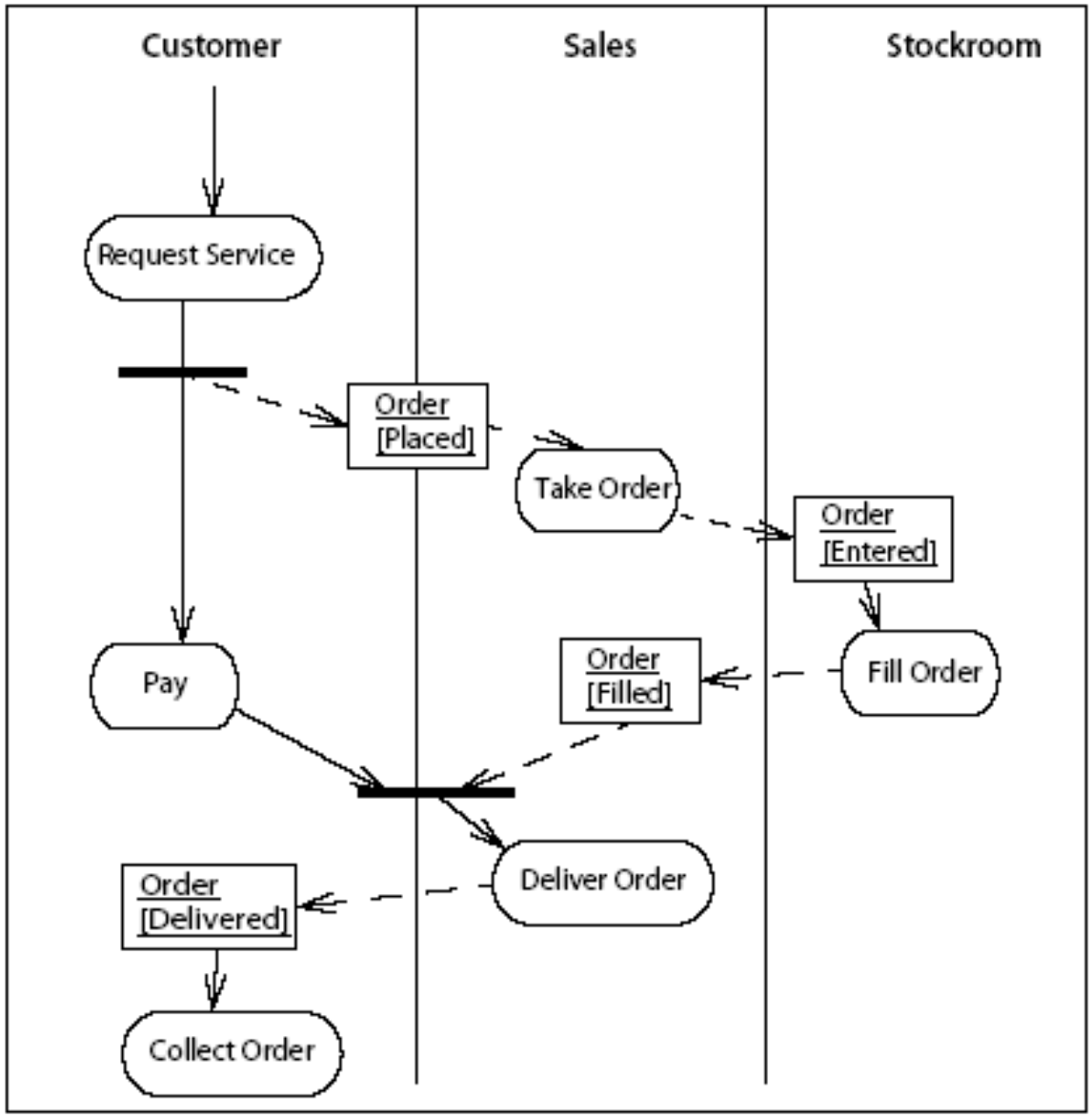
- E' prevista la possibilità di attendere un evento asincrono o trasmettere un segnale (se diventano tanti meglio passare alle macchine a stati)



Corsie e flusso di oggetti

- Le corsie vengono normalmente utilizzate per evidenziare i ruoli/le responsabilità/le unità organizzative
  - ad esempio raggruppando le attività di una componente organizzativa del sistema modellato
- Oltra al flusso del controllo il diagramma può mostrare il flusso di oggetti: ovvero gli oggetti in ingresso o in uscita alle varie attività

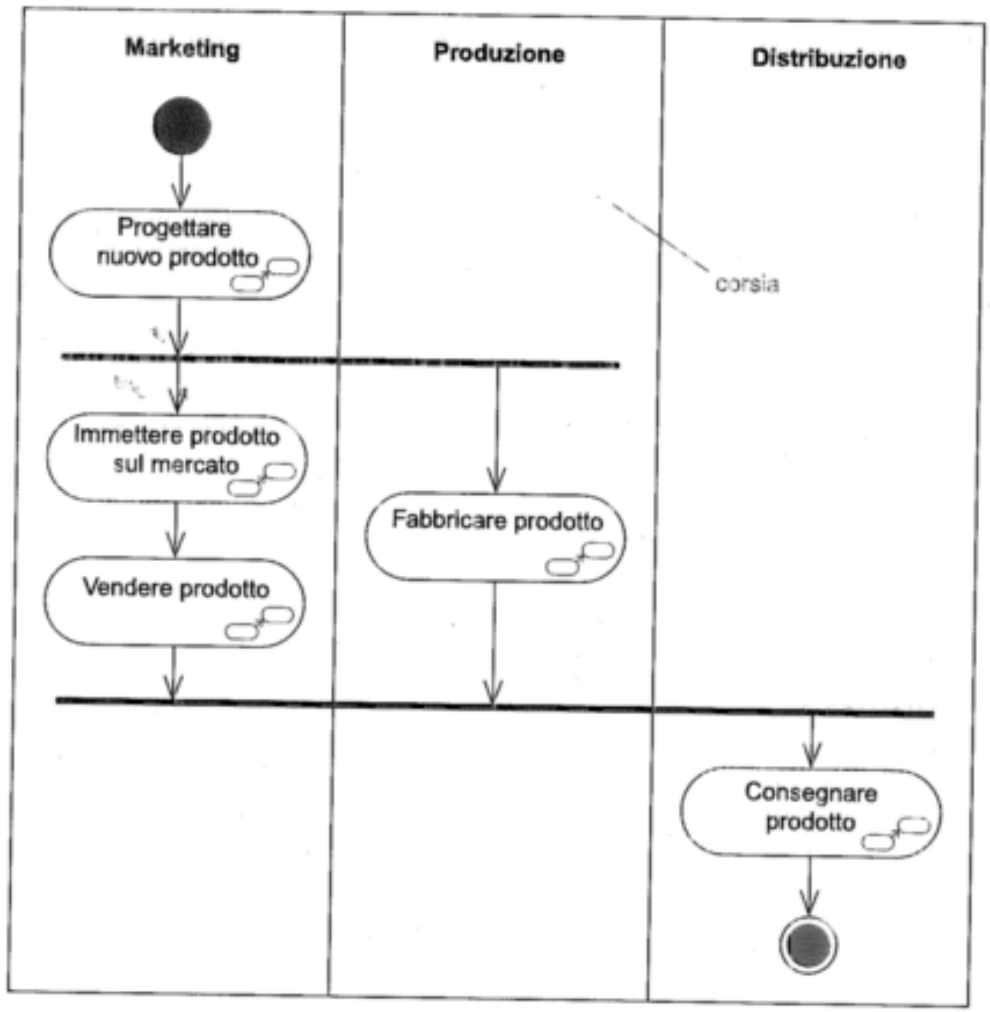
Corsie e flussi di oggetti

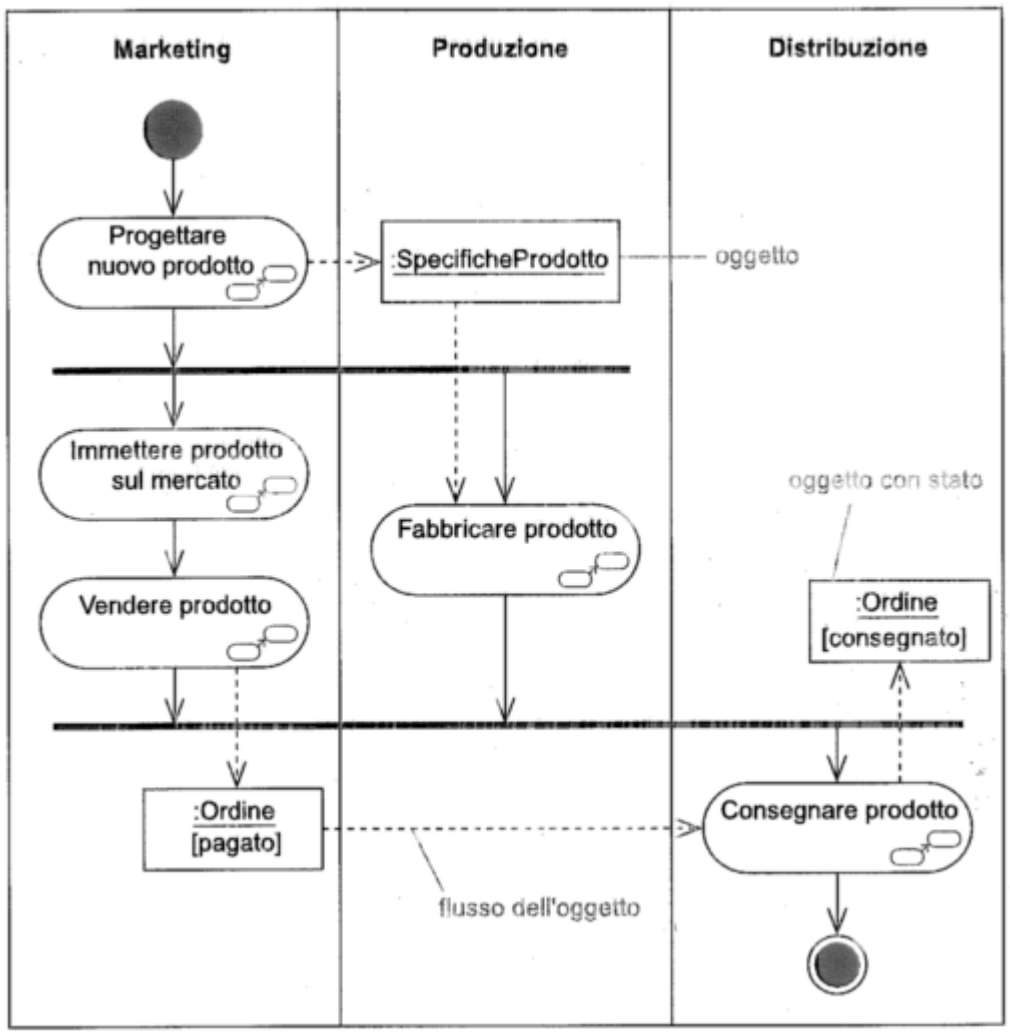


Diagrammi di attività

- Mostrano le attività, ma non gli oggetti che le svolgono; non mostrano i dettagli dell'elaborazione
- Sono il punto di partenza dell'analisi/progetto

- Ogni attività deve essere scomposta in una o più operazioni, assegnata a specifiche classi
  - identificazione delle classi
  - collaborazione tra oggetti
  - interazioni tra oggetti


Marketing
Produzione
Distribuzione
Progettare nuovo prodotto
corsia
Immettere prodotto sul mercato
Fabbricare prodotto
Vendere prodotto
Consegnare prodotto


Marketing
Produzione
Distribuzione
Progettare nuovo prodotto
:SpecificheProdotto
oggetto
Immettere prodotto sul mercato
Fabbricare prodotto
oggetto con stato
Vendere prodotto
:Ordine [consegnato]
:Ordine [pagato]
Consegnare prodotto
flusso dell'oggetto