

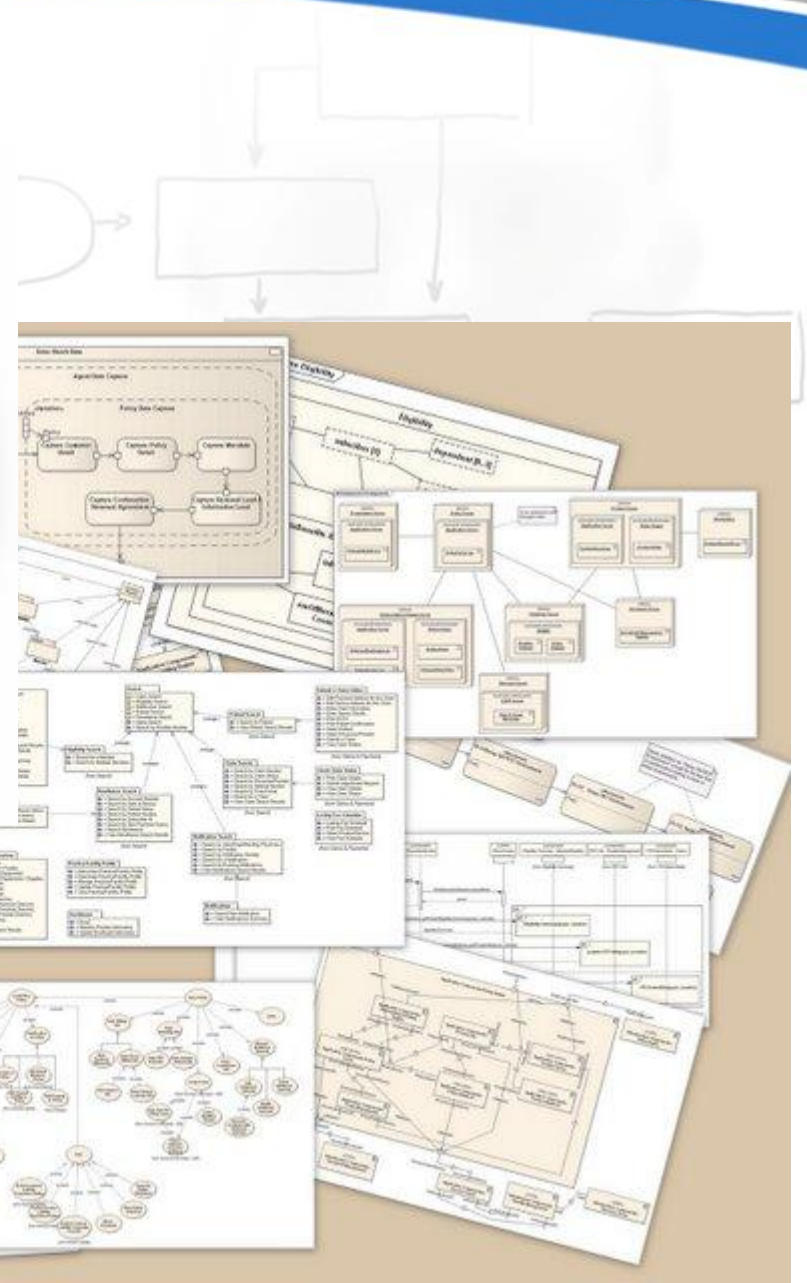
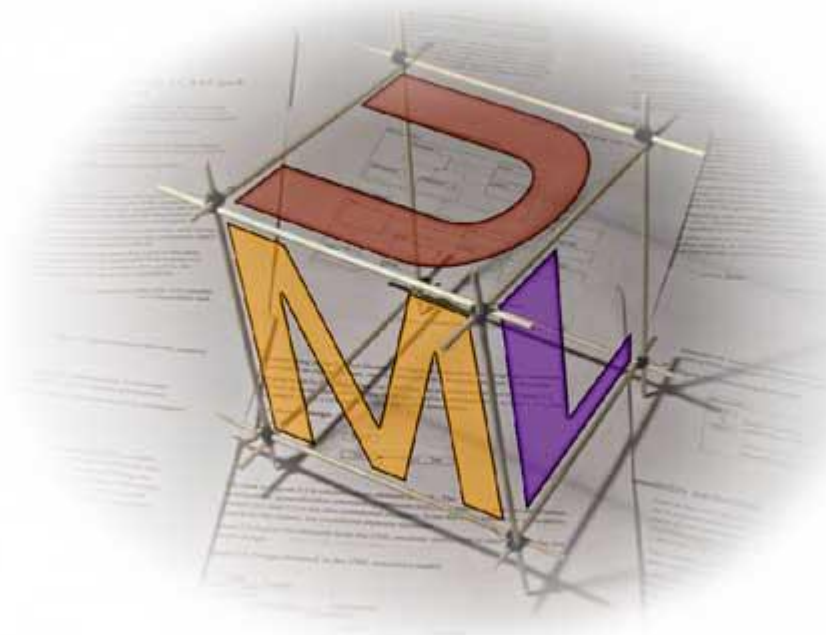


# UML e i diagrammi di stato

*Sintassi e Linee Guida*

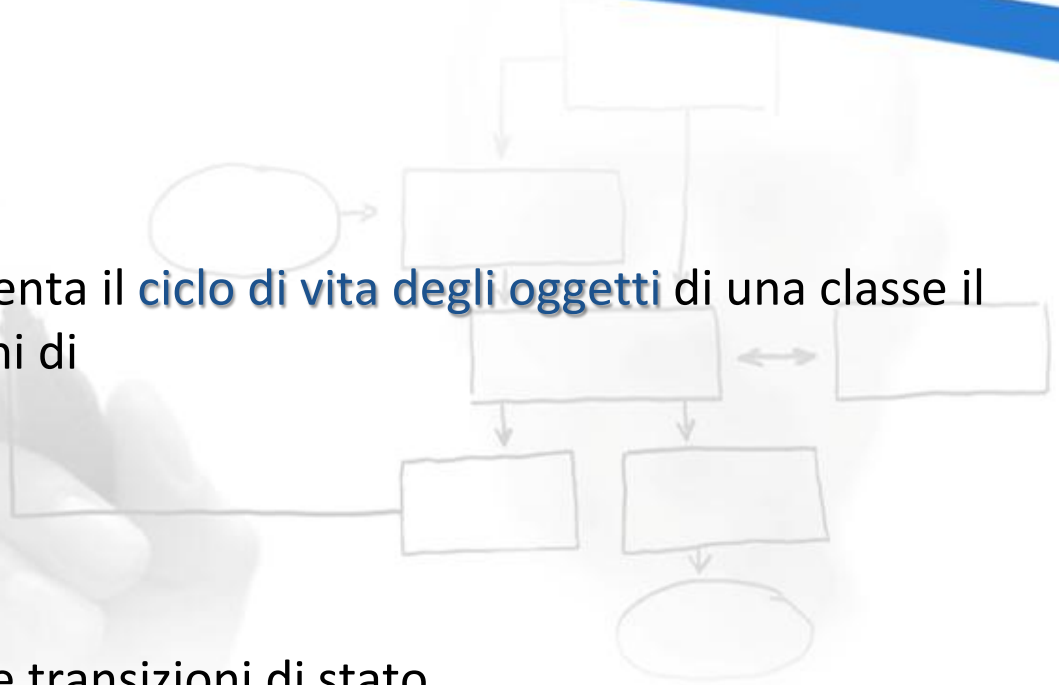
Dr. Andrea Baruzzo

[andrea.baruzzo@dimi.uniud.it](mailto:andrea.baruzzo@dimi.uniud.it)



## Definizioni

- Un diagramma di stato rappresenta il **ciclo di vita degli oggetti** di una classe il ciclo di vita è descritto in termini di
  - **Eventi**
  - **Stati**
  - **Transizioni di stato**
- Gli eventi possono attivare delle transizioni di stato
- Un evento in uno statechart corrisponde ad un messaggio in un sequence diagram
- Uno stato è costituito da un insieme di “valori significativi” assunti dagli attributi dell’oggetto che ne influenzano il comportamento

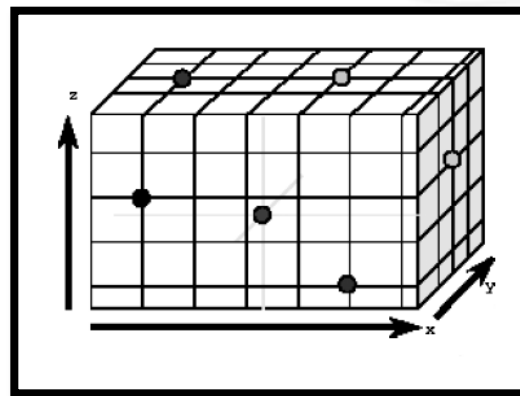


# Spazio degli stati

## ■ Spazio degli stati di una classe C:

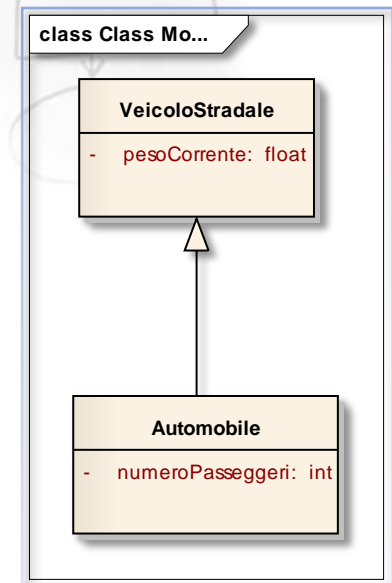
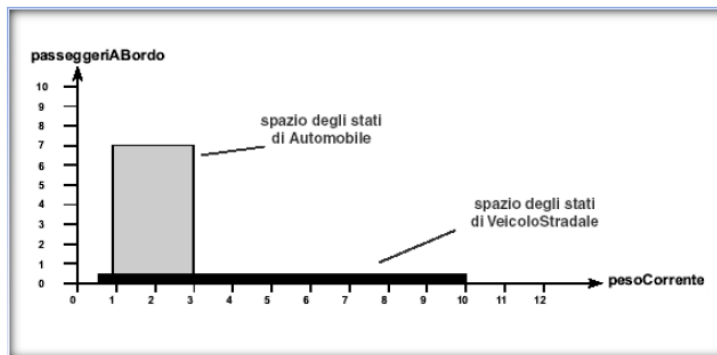
- Insieme degli stati in cui un oggetto di C può trovarsi
- È una definizione puntuale: spesso è conveniente raggruppare un insieme di valori caratterizzati da una stessa proprietà in uno stato distinto
  - È una questione pragmatica, altrimenti dovremmo fare uno stato per ogni possibile combinazione di valori dei vari attributi

```
class Punto3D {  
    private double x;  
    private double y;  
    private double z;  
    ...  
}
```



## Esempio di stati e spazio degli stati di una classe

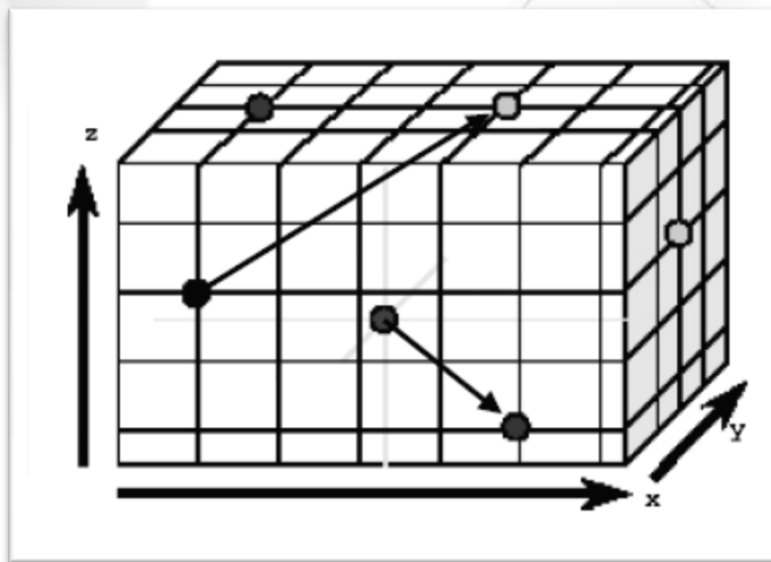
- Se potessi classificare un sottoinsieme di valori assunti da numeroPasseggeri mediante una proprietà interessante, potrei raggruppare tale sottoinsieme in (ad es.) tre stati significativi distinti, anziché 7
  - “Automobile vuota” → numeroPasseggeri = 0
  - “Passeggeri a bordo” →  $0 < \text{numeroPasseggeri} < 7$
  - “Automobile carica” → numeroPasseggeri = 7



## Transizioni di stato di una classe

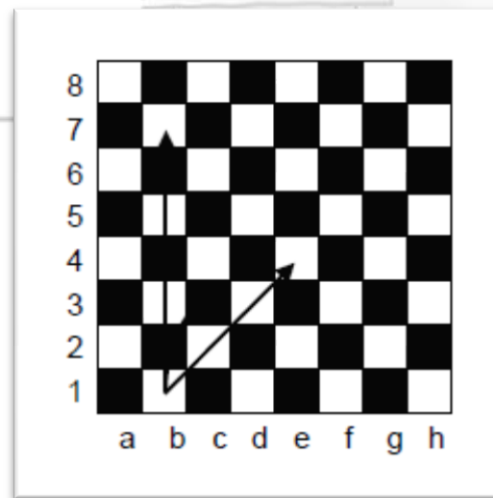
- **Transizione:** Passaggio da un punto nello Spazio degli Stati della classe ad un altro, contenuto sempre medesimo Spazio degli Stati
- Le transizioni sono provocate dall'esecuzione di particolari azioni od eventi

```
class Punto3D {  
    private double x;  
    private double y;  
    private double z;  
    ...  
    void sposta(...) {...}  
}
```



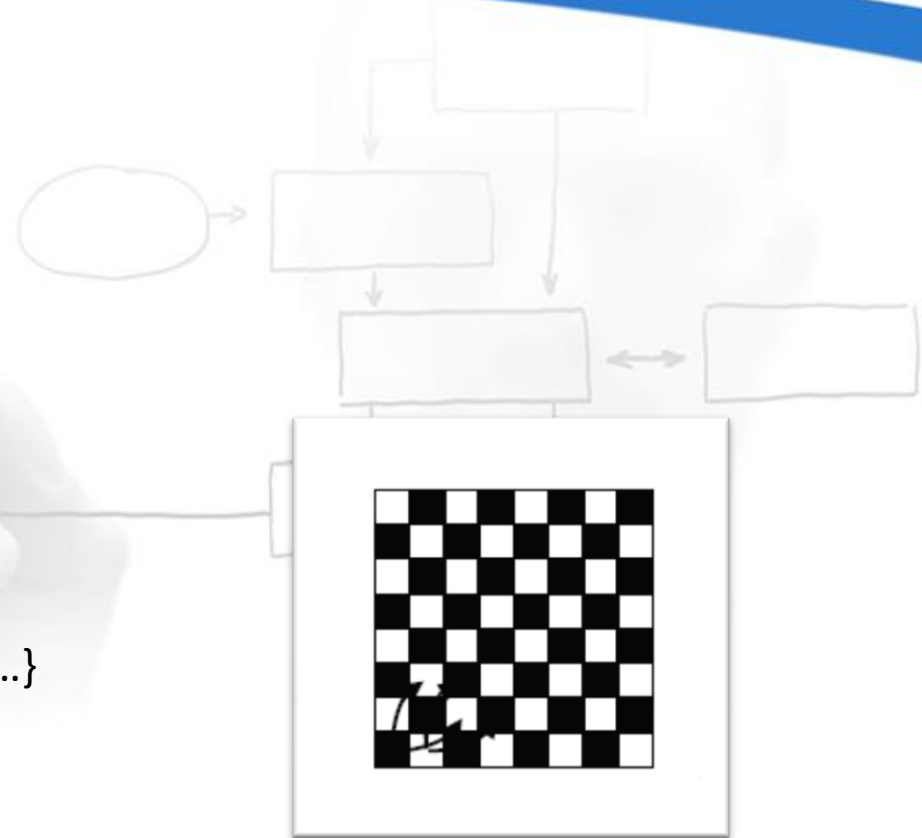
## Esempi di transizioni

- Transizione per cavallo:  $b1 \rightarrow c3$
- Transizione per regina:  $b1 \rightarrow b7$
- Transizione per alfiere:  $b1 \rightarrow e4$



# Comportamento

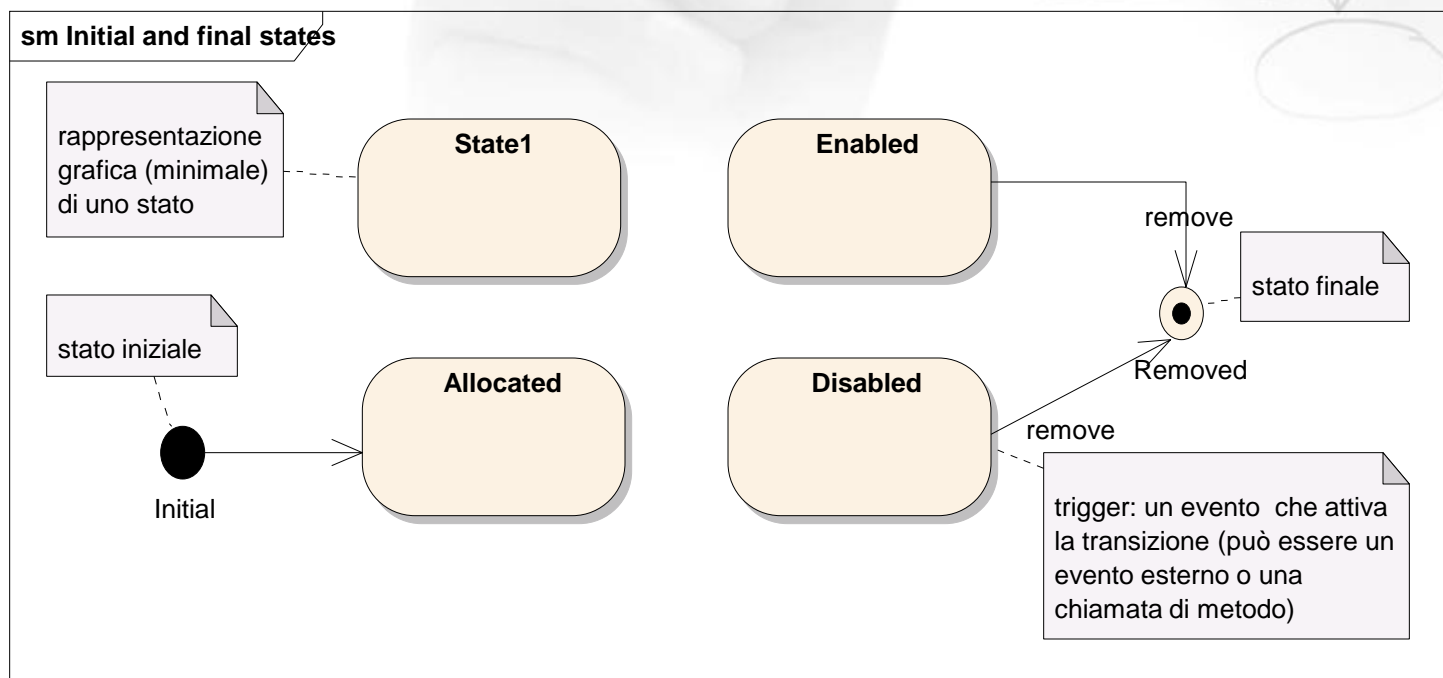
- **Comportamento** di una classe:
  - Insieme delle *transizioni di stato lecite*
- Comportamento del pezzo Cavallo:
  - {a1 → b3, a1 → c2, b1 → a3, b1 → c3, ...}





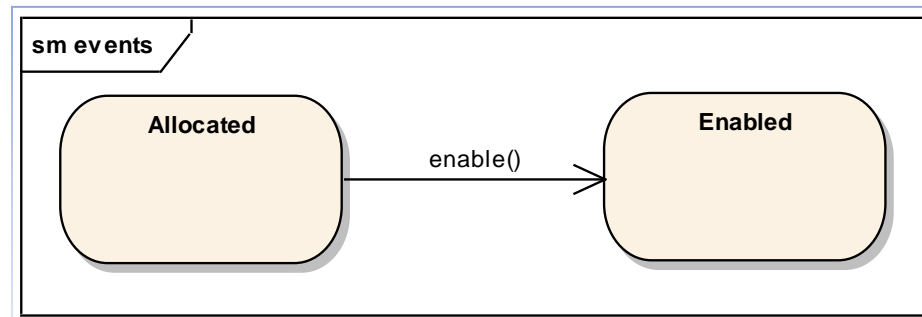
## Stati iniziale e finale

- Esistono due stati “speciali”, detti pseudostati:
  - Lo stato iniziale
  - Lo stato finale
- Un oggetto può non avere uno stato finale! (non viene mai distrutto)



# Modellare gli eventi

- Un evento può essere:
  - L'*invocazione sincrona di un metodo* (una “call”)
  - La ricezione di una *chiamata asincrona* (“signal”) – ad esempio la notifica di un’eccezione lanciata
  - Una *condizione predefinita* che diventa vera (si parla in questo caso di “change event”)
  - La fine di un “*periodo di tempo*” come quello impostato da un timer (“elapsed-time event”)
- Un evento si può rappresentare graficamente con una freccia (transizione) etichettata con il nome del metodo o della condizione associata all’evento stesso
  - Qualora l’evento corrisponda ad una chiamata a metodo, la sintassi per la chiamata è simile a quella dei diagrammi di sequenza (freccia); c’è coerenza ...

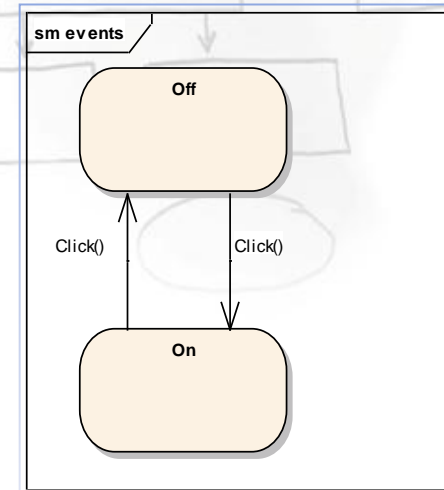
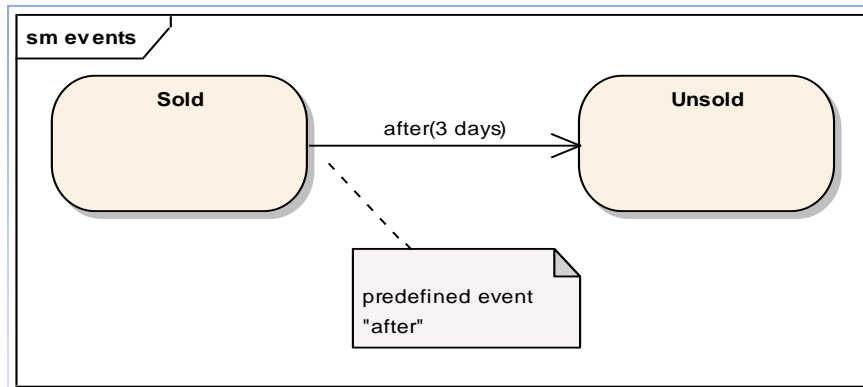


## Modellare gli eventi (cont.)

- Un evento può essere rappresentato anche mediante un'espressione testuale avente la seguente sintassi:
  - event-name '(' [comma-separated-parameter-list] ')' [['guard-condition']] / [action-expression]
- dove:
  - **event-name** identifica l'evento
  - **parameter-list** definisce i valori dei dati che possono essere passati come parametro con l'evento
  - **guard-condition** determina se l'oggetto che riceve l'evento deve rispondere ad esso (ossia eseguire il metodo associato)
  - **action-expression** definisce come l'oggetto ricevente deve rispondere all'evento

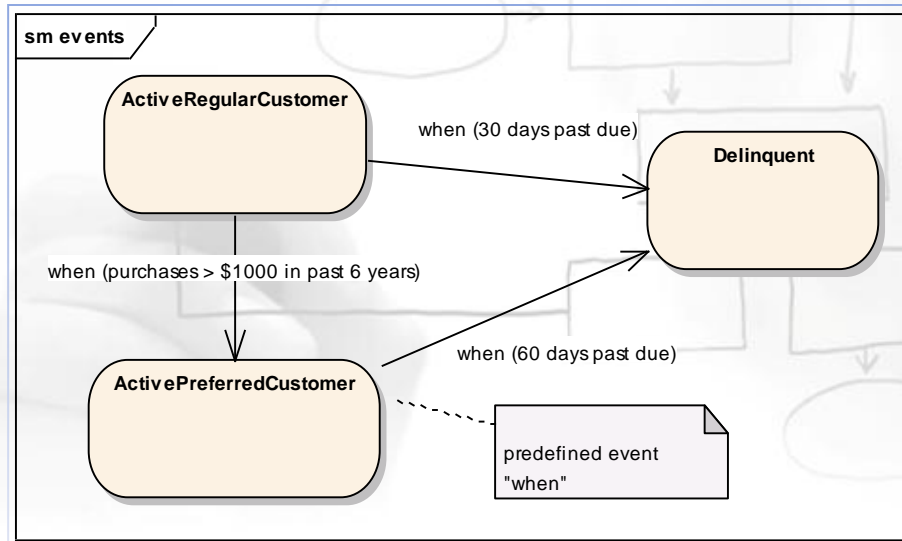
## Esempi di diagrammi di stato

- “Event + state = response”
- Lo stesso evento causa diversi comportamenti in base allo stato in cui l’oggetto che riceve l’evento si trova
- Elapsed-time Events

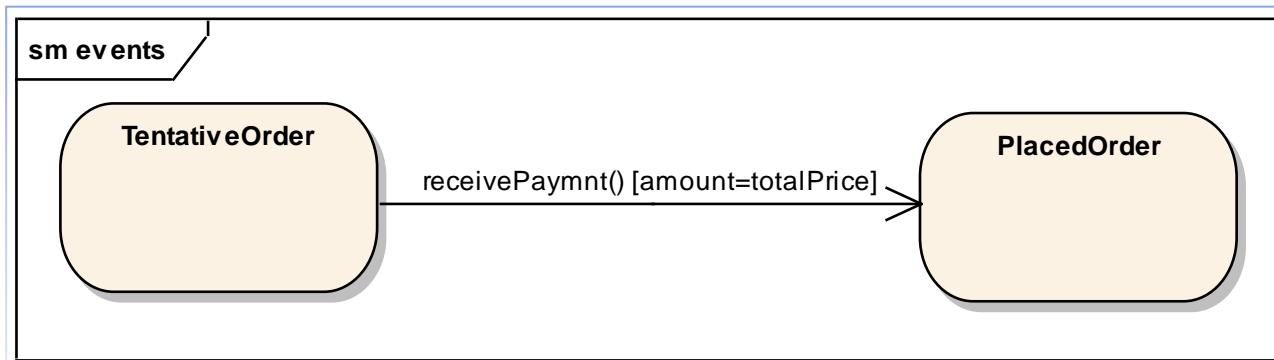


## Esempi di diagrammi di stato (cont.)

### Change event

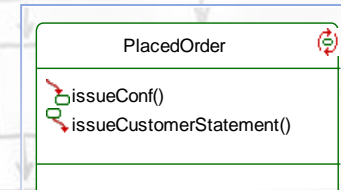


### Guarded event



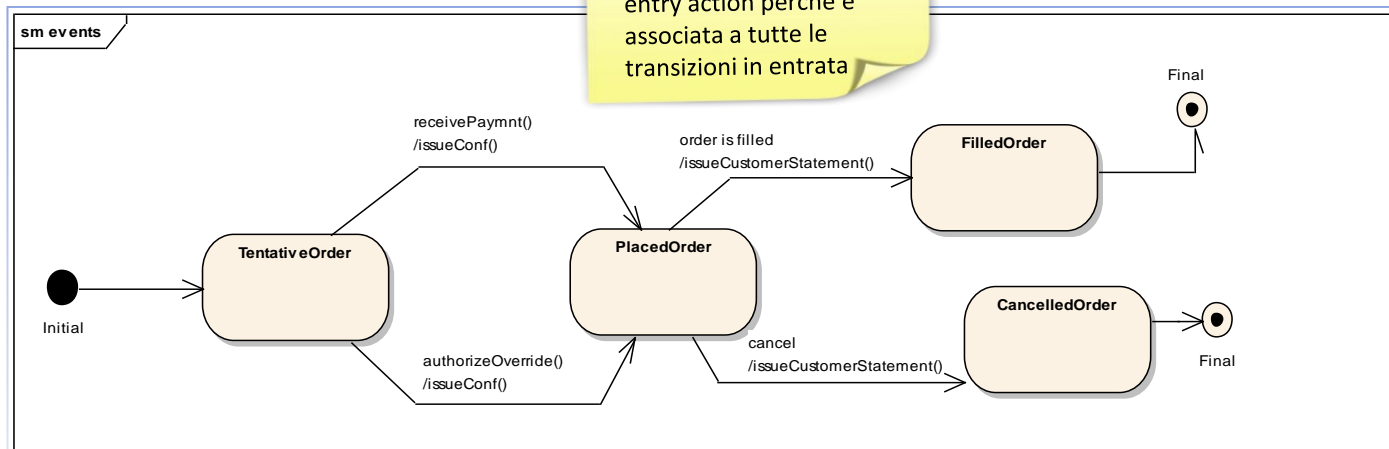
# Entry/Exit Actions

- **Entry/exit action:** azione che viene eseguita per ogni evento che causa una transizione entrante/uscente nello/dallo stato
- **entry action:** azione che viene eseguita in una transizione entrante nello stato
- **exit action:** azione che viene eseguita in una transizione uscente dallo stato



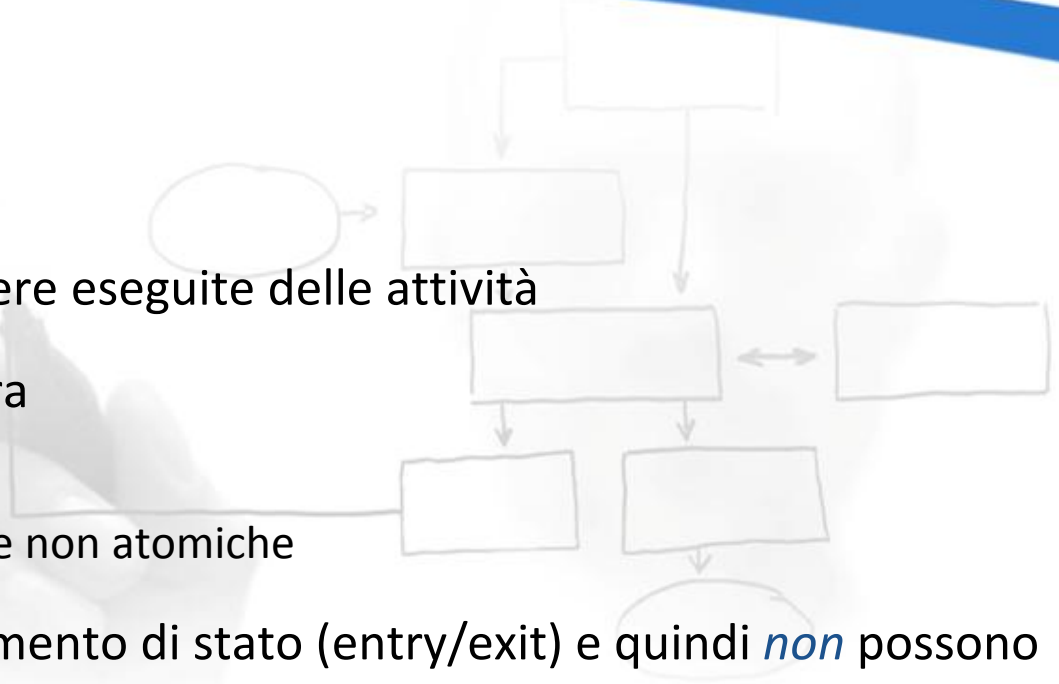
Rappresentazione grafica di entry/exit actions (tool dependent!)

`issueConf()` è una entry action perchè è associata a tutte le transizioni in entrata



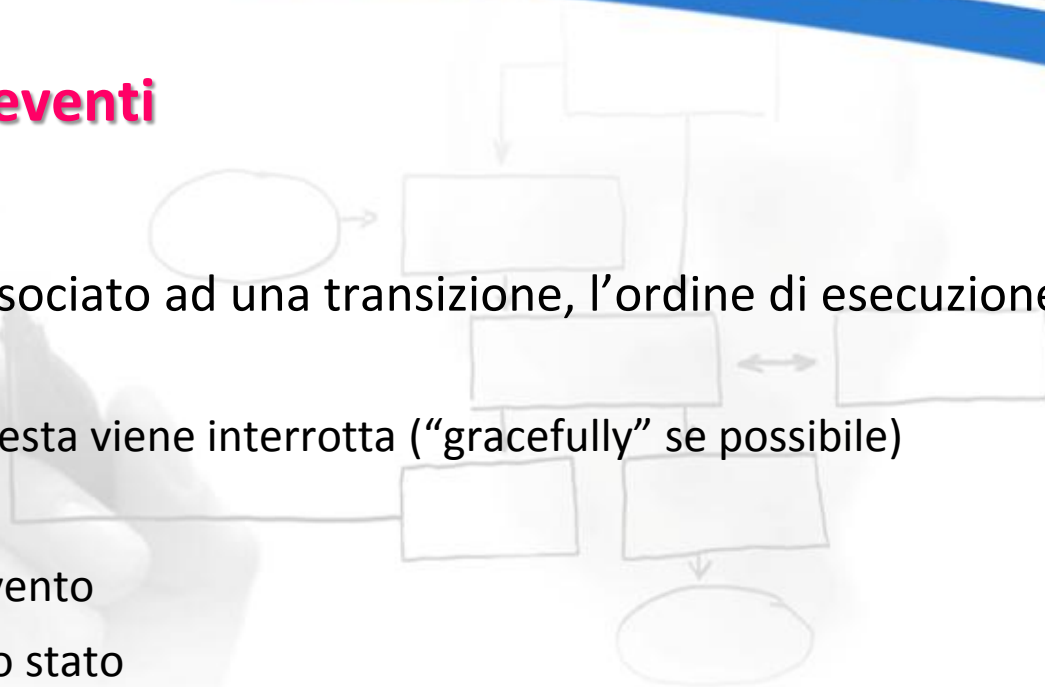
## Modellare le attività

- All'interno degli stati posso essere eseguite delle attività
- Negli statechart distinguiamo tra
  - **Azioni**: operazioni atomiche
  - **Attività**: operazioni generalmente non atomiche
- Le azioni provocano un cambiamento di stato (entry/exit) e quindi *non* possono essere interrotte
- Le attività non alterano lo stato dell'oggetto



## Ordine di esecuzione degli eventi

- Quando si verifica un evento associato ad una transizione, l'ordine di esecuzione è il seguente:
  - Se è in esecuzione un'attività, questa viene interrotta ("gracefully" se possibile)
  - Si esegue l'exit action
  - Si esegue l'azione associata all'evento
  - Si esegue l'entry action del nuovo stato
  - Si inizia l'esecuzione delle eventuali attività del nuovo stato



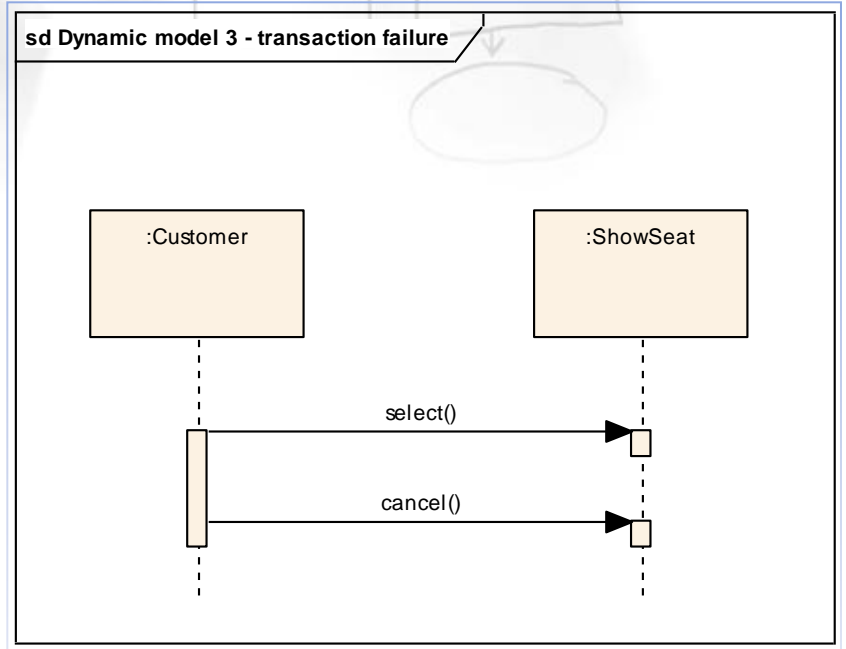
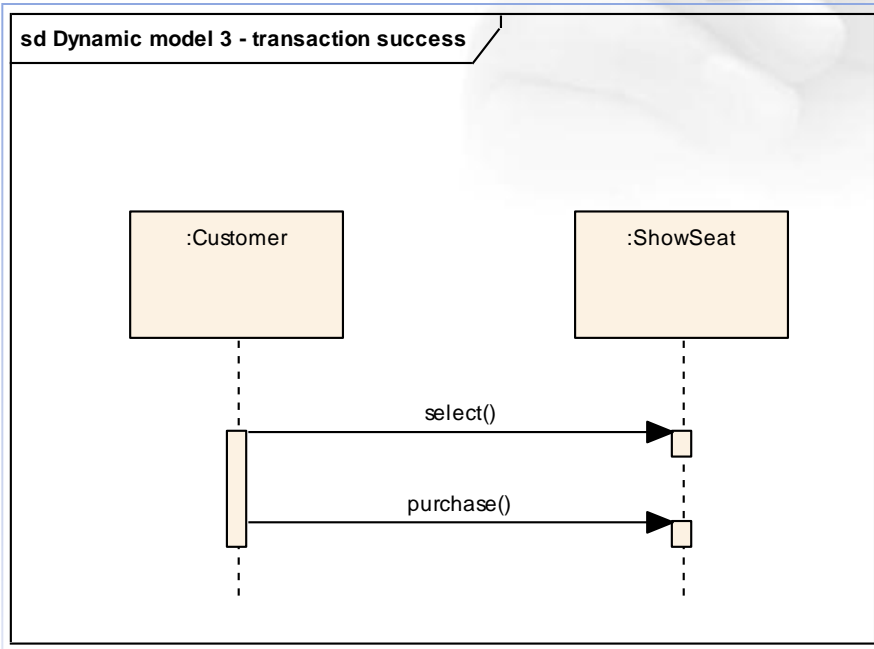


# Relazione tra diagrammi di stato e diagrammi di sequenza

- Due scenari (sequence diagram): successo e fallimento di una transazione

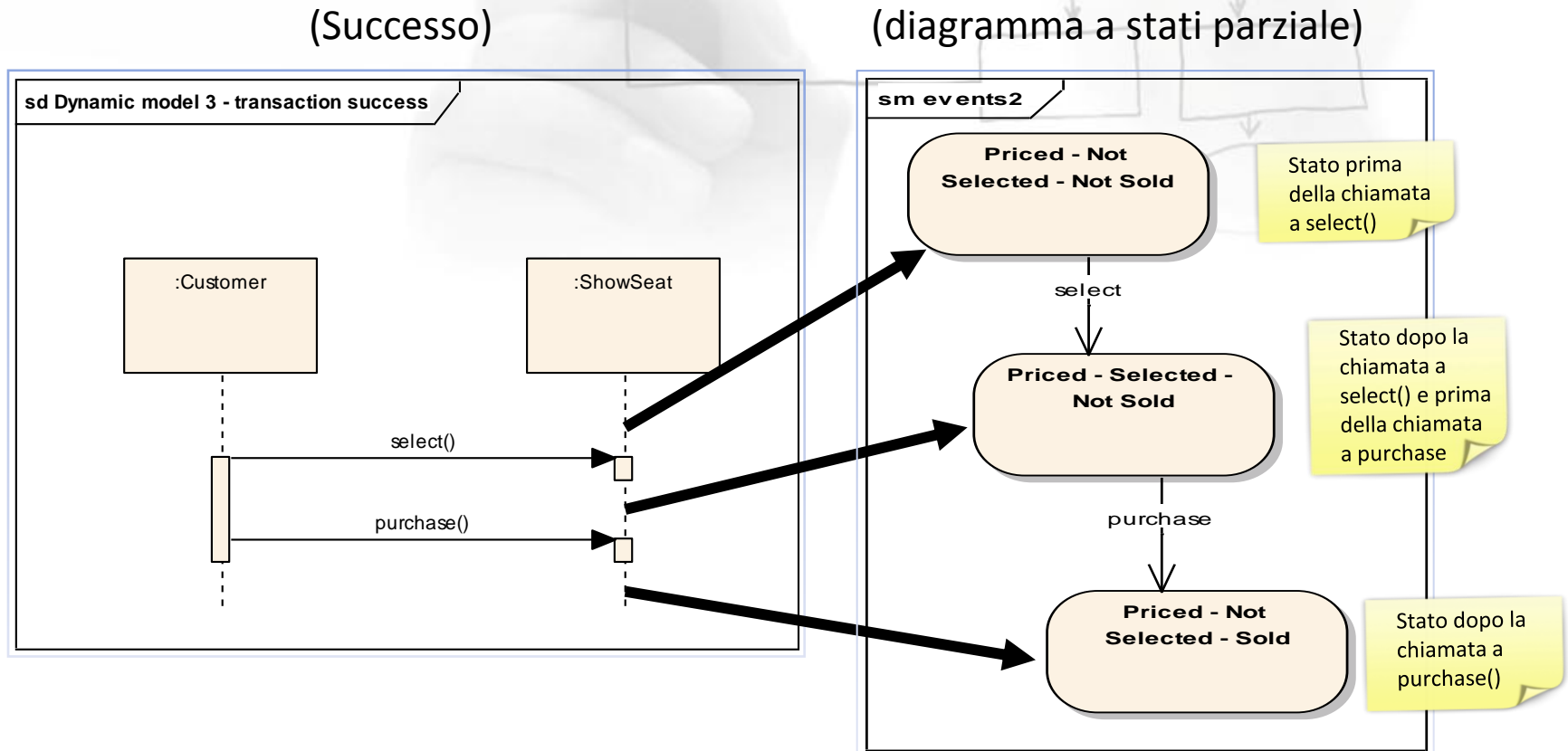
(Successo)

(Fallimento)



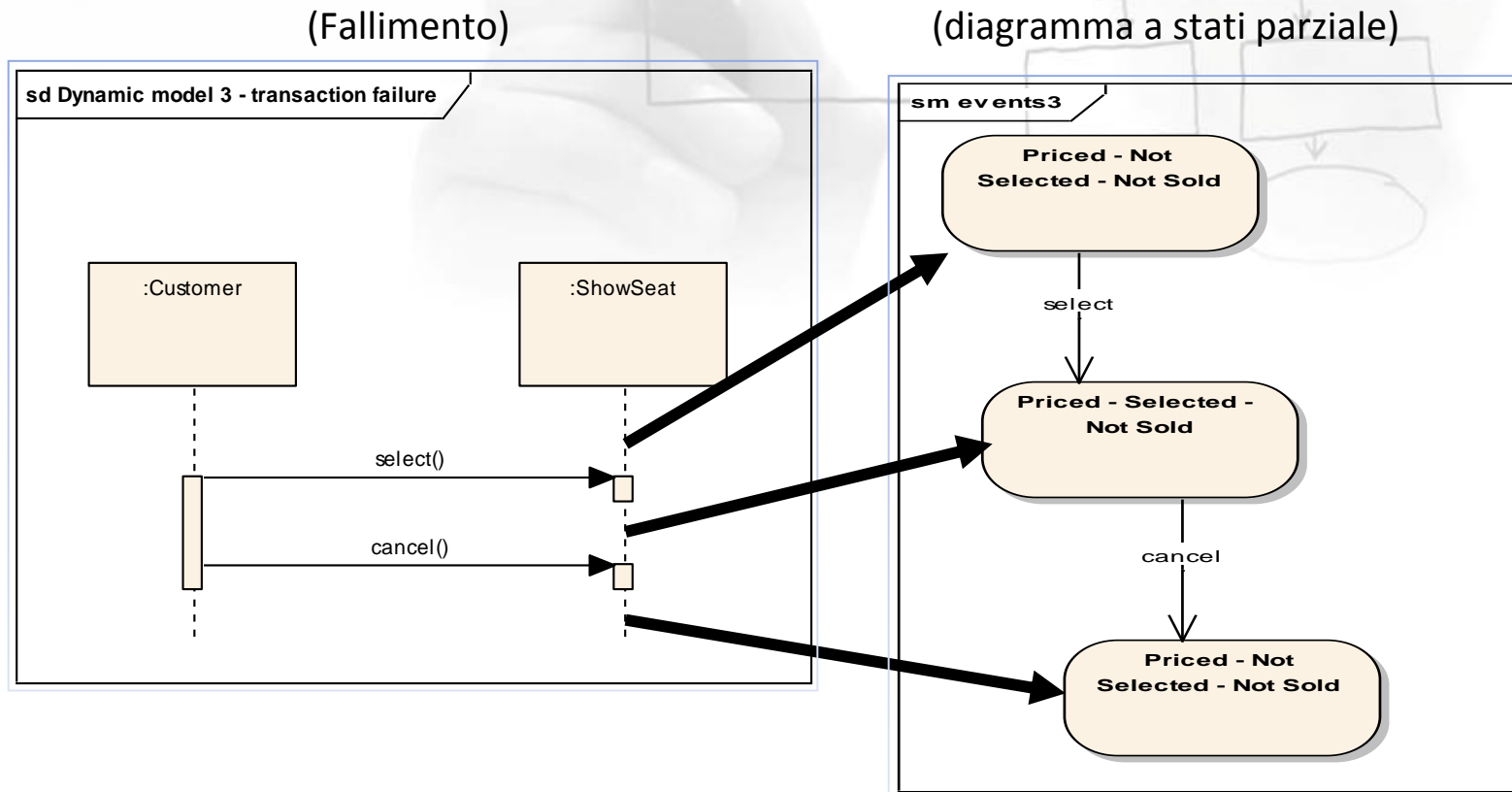
# Relazione tra diagrammi di stato e diagrammi di sequenza (cont.)

- Scenario di successo e relativo (parziale) diagramma a stati



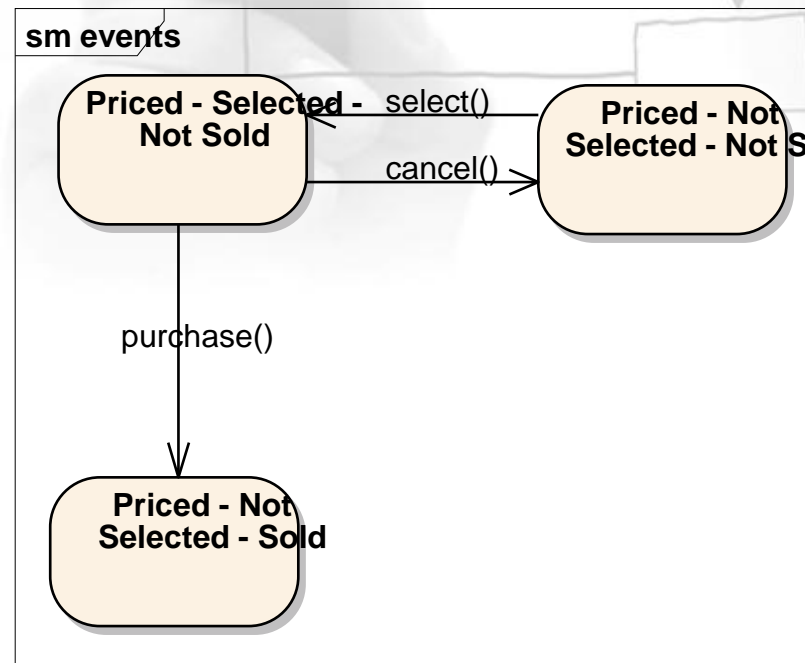
# Relazione tra diagrammi di stato e diagrammi di sequenza (cont.)

- Scenario di fallimento e relativo (parziale) diagramma a stati



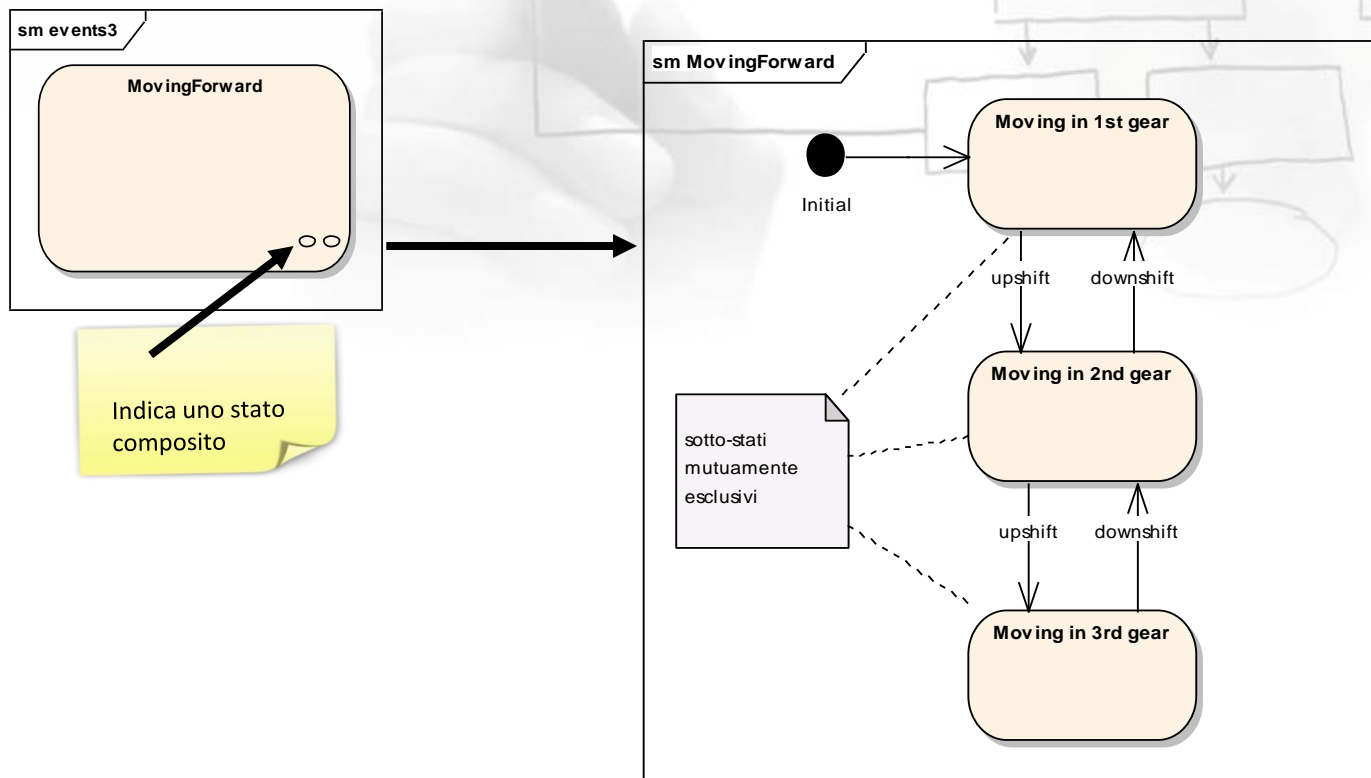
## Relazione tra diagrammi di stato e diagrammi di sequenza (cont.)

- Il diagramma a stati completo (relativo ai due scenari discussi)



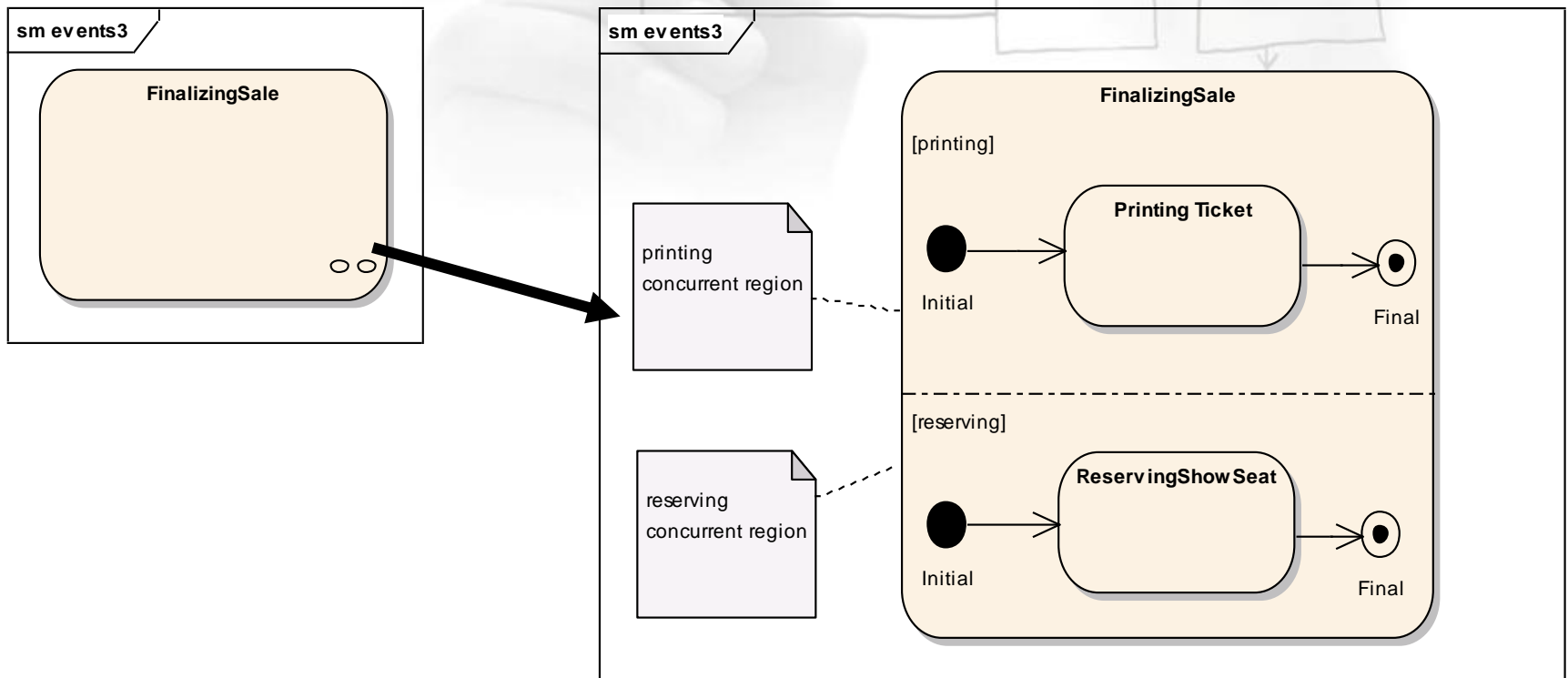
# Diagrammi di stato compositi

- Uno stato può contenere al suo interno più **sottostati mutuamente esclusivi**



## Diagrammi di stato compositi (cont.)

- Uno stato può contenere al suo interno **sottostati concorrenti**
  - Entro nello stato quando almeno una regione viene attivata
  - Esco dallo stato quando in tutte le regioni ho raggiunto lo stato finale



## Bibliografia

- [Booch et al., 2005] Grady Booch et al. *“The Unified Modeling Language User Guide 2/E”*, Addison-Wesley, 2005
- [Rumbaugh et al., 2004] J. Rumbaugh et al. *“The Unified Modeling Language Reference Manual 2/E”*, Addison-Wesley, 2004
- [Fowler, 2003] Martin Fowler. *“UML Distilled 3/E”*, Addison-Wesley, 2003
- [Larman, 2004] C. Larman. *“Applying UML and Patterns”*, Addison-Wesley, 2004
- [Pender, 2003] Tom Pender. *“UML Bible”*, Wiley&Sons, 2003



**Domande?**  
**Commenti?**  
**Dubbi?**