



Note sugli Statechart Diagrams

Giacomo Gabrielli

Sorgente: [Bolognesi05]



Diagrammi di Stato

- I *diagrammi di stato* (*statechart diagram*) permettono di descrivere il comportamento dinamico di un oggetto o di un sistema
- Descrivono tutti gli stati raggiungibili e come cambia lo stato dell'oggetto in relazione all'accadere degli eventi
- Un diagramma di stato è quindi composto da **stati** e da **transizioni** fra stati



Stati

- Uno **stato** è una situazione nella vita dell'oggetto in questione in cui esso soddisfa una qualche condizione, esegue una qualche attività o è in attesa di un qualche evento
- Ogni stato è etichettato con un **nome** e possibilmente con: *do / attività*



Transizioni

- Una **transizione** è una relazione che lega uno stato di partenza ed uno stato di arrivo (non necessariamente distinti)
- Ogni transizione è etichettata con tre elementi tutti opzionali:
Evento [Condizione] / Azione
- Le transizioni che riportano nello stesso stato sono dette self-transition



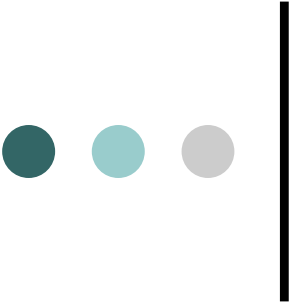
Eventi e Condizioni

- **Evento (trigger):** è l'occorrenza di uno stimolo che può innescare una transizione fra stati
- **Condizione (guard):** espressione logica (booleana) da valutare. Se da uno stato escono più transizioni, le loro condizioni devono essere mutuamente esclusive



Azioni e Attività

- **Azione:** è associata alla transizione, è considerata un processo rapido, non interrompibile da un evento (atomica)
- **Attività:** è associata allo stato, può prevedere un lasso di tempo considerevole, e può essere interrotta da un evento (non-atomica)



A Cosa Servono gli Statechart Diagram?

- Ogni oggetto in un sistema ha un certo *ciclo di vita*. Tra la sua creazione e la sua distruzione esso interagisce con altri oggetti. L'interazione può avvenire attraverso semplici chiamate sincrone di operazioni
 - Tipicamente l'invocazione di metodi di una classe
- Il comportamento dell'oggetto non necessariamente dipende dal suo passato e dunque, in quel caso, non c'è bisogno di una macchina a stati per specificarlo



A Cosa Servono gli Statechart Diagram? (ii)

- In altri casi tuttavia l'oggetto dovrà rispondere a stimoli asincroni che gli arrivano dal suo ambiente...
 - Per esempio un cellulare deve essere sempre pronto a rispondere a chiamate in entrata
- ... oppure esistono oggetti il cui comportamento dipende da certe *condizioni di stato*
 - Per esempio, sempre in un cellulare, all'arrivo di una chiamata l'attivarsi o meno della suoneria dipende dalla modalità (normale o "silenziosa") in cui si trova l'apparecchio



A Cosa Servono gli Statechart Diagram? (iii)

- Il comportamento di questi tipi di oggetti (che spesso sono istanze di classi, ma possono essere anche interi sottosistemi) viene modellato attraverso l'uso di una macchina a stati (statechart)
- Se un oggetto sprovvisto di una macchina a stati riceve un segnale, esso è ignorato



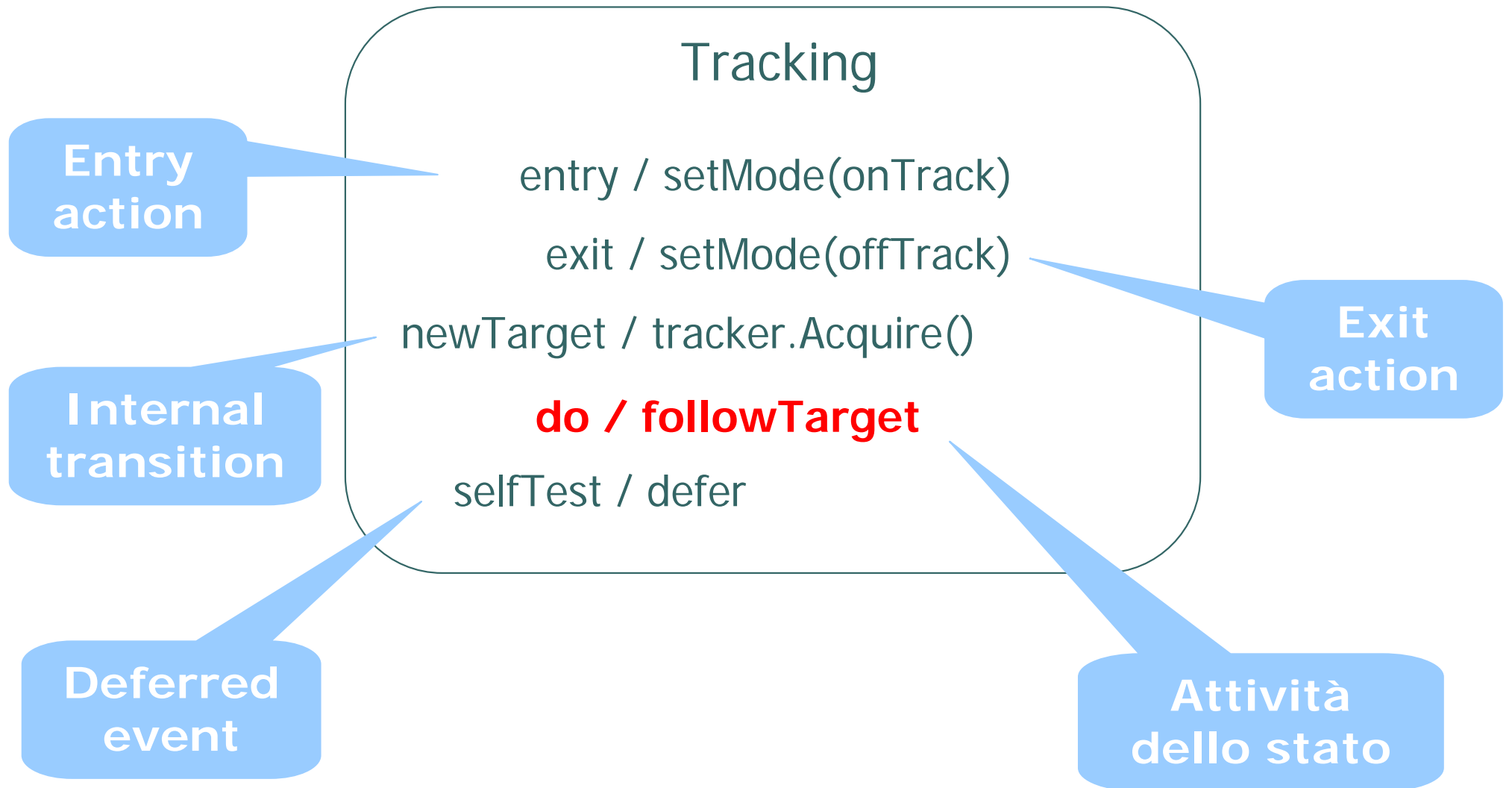
Stato: Notazione

Oltre all'**attività**, uno stato può avere:

- **Entry action:** azione eseguita ogni volta che si entra nello stato (cioè in risposta all'evento *entry*)
- **Exit action:** azione eseguita ogni volta che si lascia lo stato (cioè in risposta all'evento *exit*)
- **Internal transition:** transizione che gestisce un evento senza cambiare stato (sono diverse dalle self-transition in cui si esce dallo stato rientrando nello stesso in quanto nelle self-transition si eseguono le eventuali exit e entry action)
- **Deferred event:** sono eventi che non sono gestiti all'interno dello stato, ma che non si vuole vadano persi. Perciò la risposta all'evento è posposta a quando il sistema entrerà in uno stato in cui tale evento è gestito



Esempio di Stato

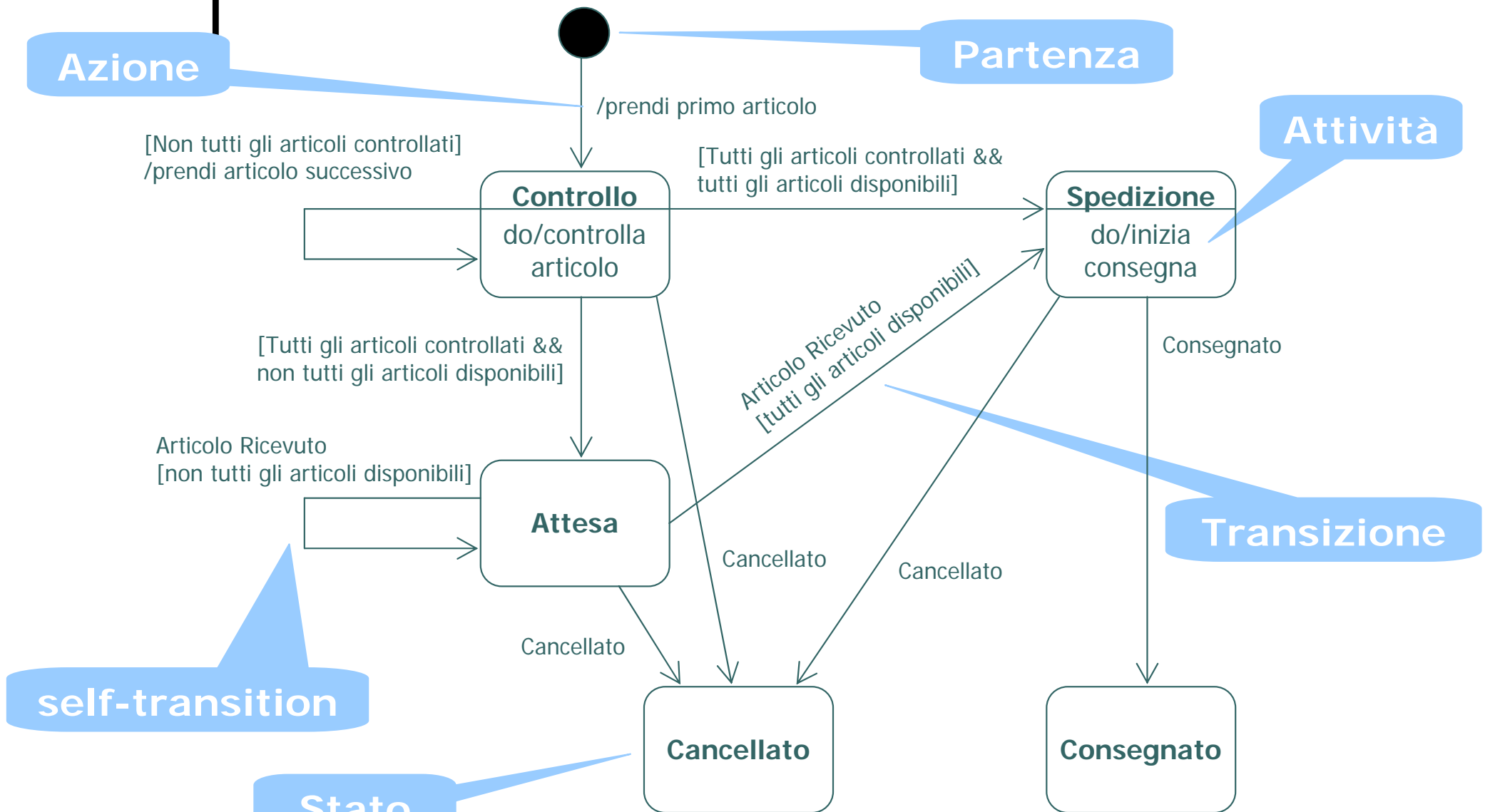




Stato Iniziale e Stato Finale

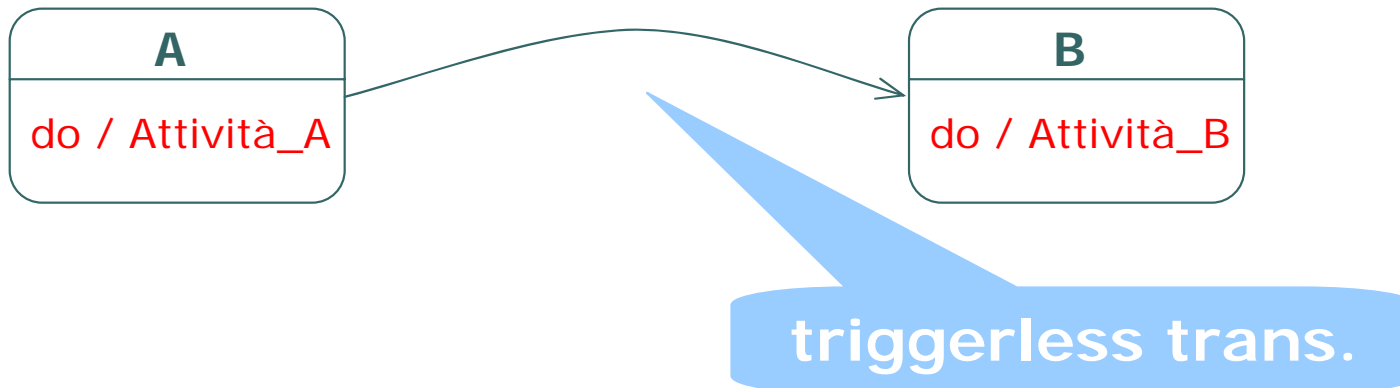
- Sono due stati particolari: il primo è il punto di partenza del diagramma ed il secondo è il punto di arrivo
- Sono indicati rispettivamente con ● e ⊙
- Negli statechart diagram che modellano il comportamento di sistemi destinati ad evolversi continuamente (per esempio sistemi embedded) lo stato finale può non essere presente

Esempio



Transizioni *Triggerless*

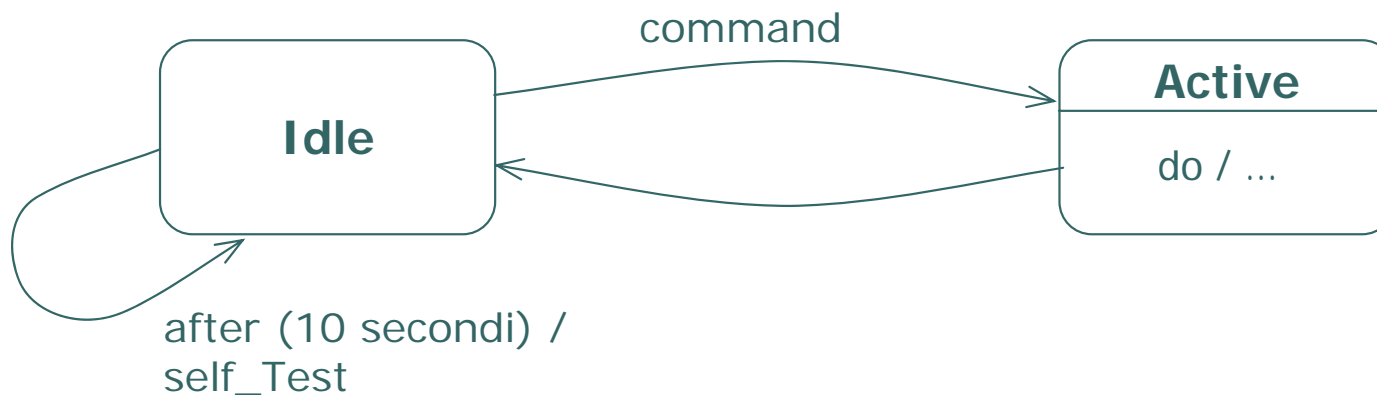
- Una transizione che non abbia nessun evento ad innescarla è detta *triggerless*



- Al termine dell'*Attività_A* si passa automaticamente nello stato B, si può dire cioè che è il termine dell'attività ad innescare la transizione fra stati

Eventi Temporali

- Sono eventi legati esplicitamente allo scorrere del tempo
 - *after* + <intervallo temporale>
 - *when* + <condizione>

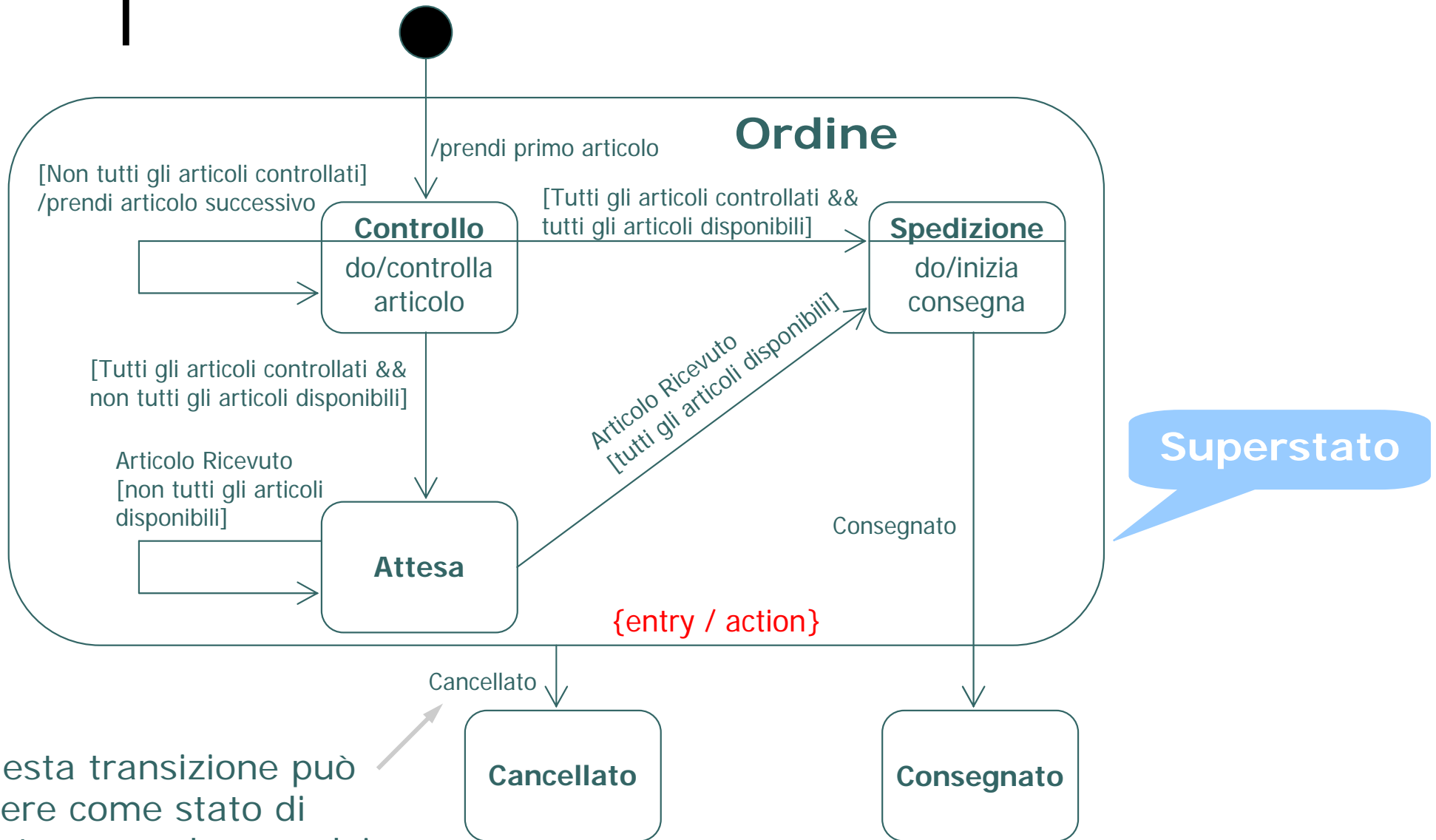




Sottostati

- Il comportamento di uno stato può essere modellato anch'esso con una macchina a stati e questo a qualsiasi livello di profondità
 - Troppi livelli penalizzano la leggibilità del diagramma
- Uno *stato composito* (cioè che contiene sottostati) può possedere entry action, exit action e deferred event con lo stesso significato che essi hanno negli stati semplici
- Una transizione che parte da un superstato rappresenta una transizione da **ogni** sottostato

Esempio



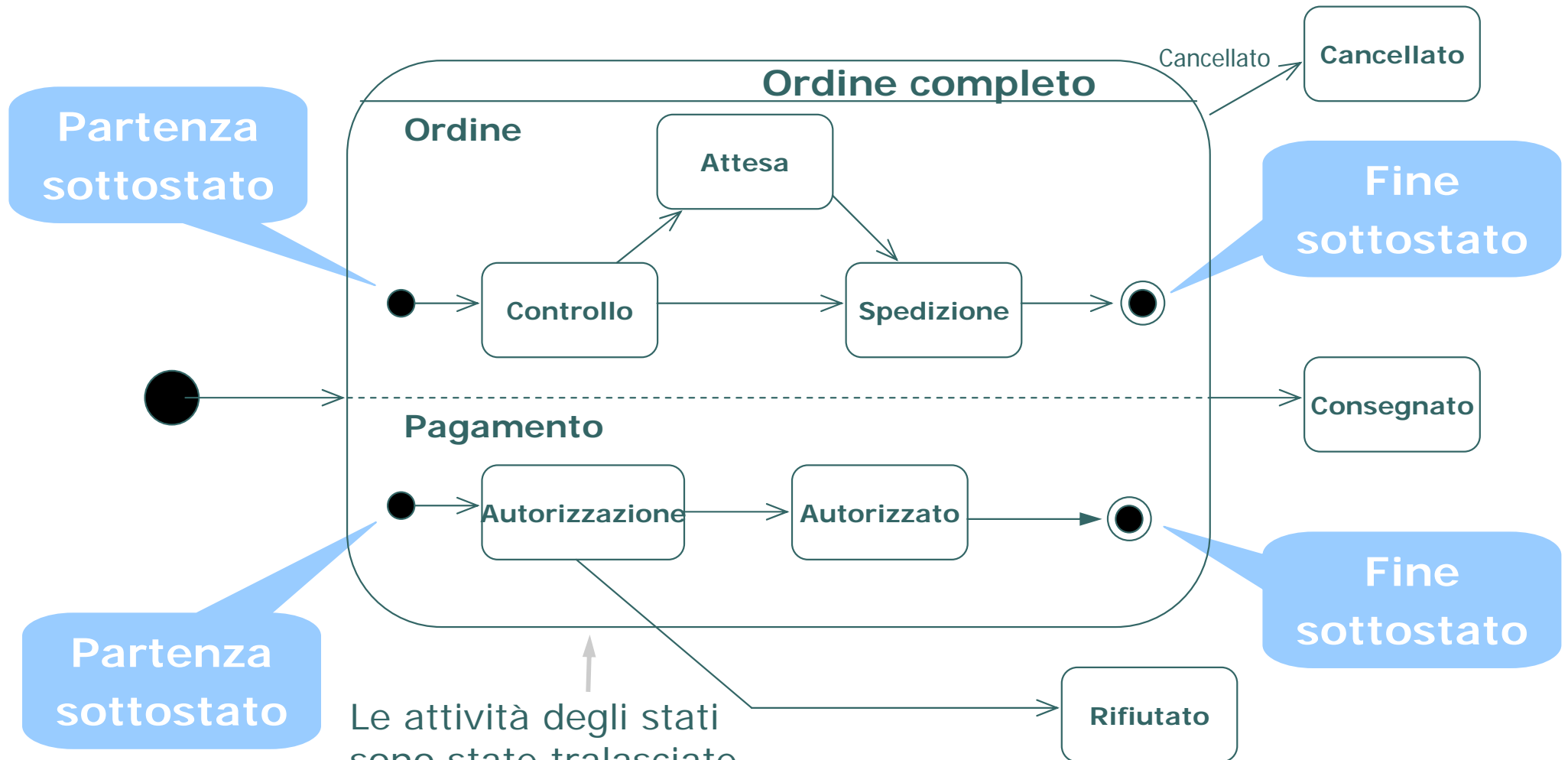
questa transizione può avere come stato di partenza qualunque dei sottostati



Stati Concorrenti

- I diagrammi di stato sono ottimali per descrivere comportamenti *concorrenti*
- Es.: controllo e spedizione di articoli in un ordine ed il suo effettivo pagamento da parte dell'acquirente
- ...

Stati Concorrenti: Esempio



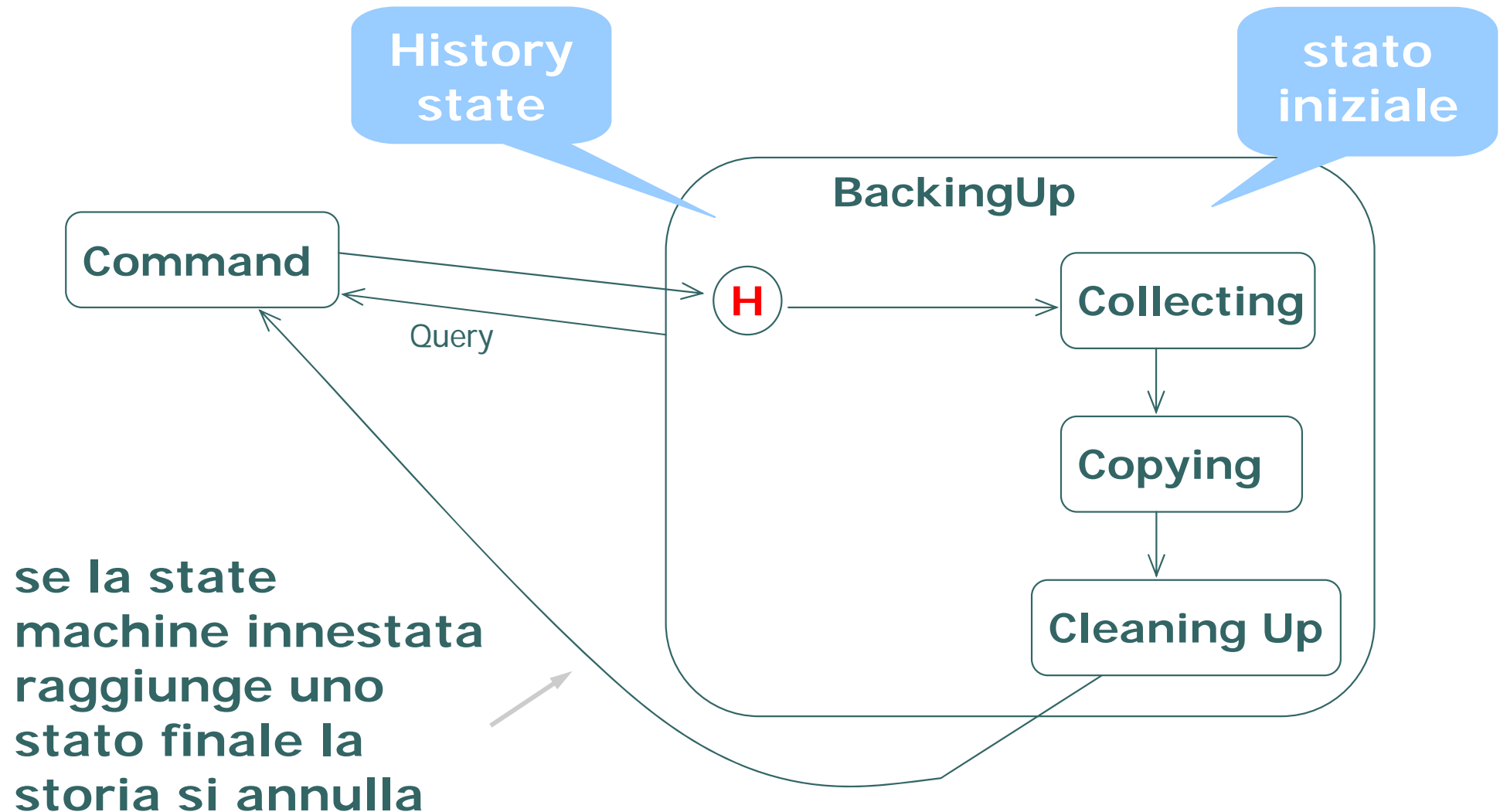
Le attività degli stati sono state tralasciate per semplicità



History State

- In generale quando, per effetto di una transizione, si entra in uno stato composito, la state machine innestata comincia dal suo stato iniziale (a meno che la transizione non specifichi come target uno specifico sottostato)
- Esistono però situazioni in cui vorremmo poter memorizzare quale sottostato era attivo nel momento in cui si è abbandonato lo stato composito
- Lo scopo è quello di riprendere da quel punto le attività una volta che si è rientrati nello stato composito
- Lo stato **H** è detto *history state*

History State: Esempio





Riferimenti

[Bolognesi05] T. Bolognesi, Slides per il corso di “Ingegneria del Software”, A.A. 2005/2006, Università di Siena