Macchine a stati

Macchina a stati

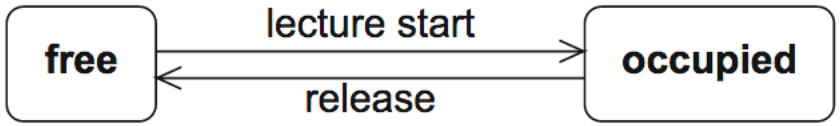
- Una macchina a stati descrive il comportamento dinamico delle istanze di un classificatore (per esempio degli oggetti istanza di una classe).
- Per costruire una macchina a stati dobbiamo individuare gli stati significativi in cui si puo' trovare un oggetto durante la sua vita.
- Inoltre dobbiamo descrivere come da ciascuno di questi stati l'oggetto può passare (transire) in un altro.
- Le transizioni avvengono in risposta al verificarsi di un evento. Gli eventi sono tipicamente;
  - messaggi inviati da altri oggetti
  - eventi generati internamente
- Una macchina a stati è rappresentata con un grafo di stati e transizioni, associata a un classificatore

Stato

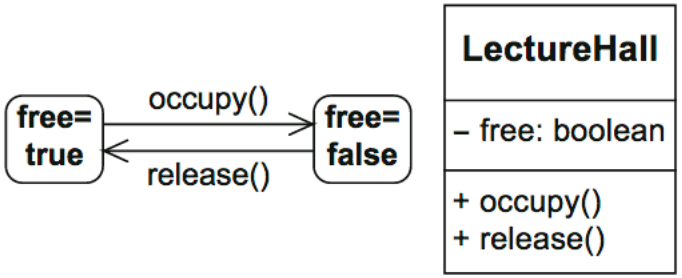
- Uno stato è un insieme di valori (di alcune variabili significative) di un oggetto:
  - È un'astrazione dello stato concreto dell'oggetto (caratterizzato dai valori di tutte le variabili)
  - Rappresenta uno stato significativo
  - È caratterizzato dal dare la stessa risposta qualitativa ad eventi che possono accadere.
- Uno stato ha un nome unico
- Uno stato può essere composito (più avanti)

Importante il livello di dettaglio

Per modellare un'aula



che verra' poi specificato a livello di implementazione




```
class LectureHall {
  private boolean free;
  public void occupy() {
    free=false;
  }
  public void release() {
    free=true;
  }
}
```

Sintassi di base

Gli stati sono rappresentati con rettangoli arrotondati

- Il disco nero marca l'inizio. Non è uno stato vero e proprio ma un marcatore che punta allo stato da cui partire.
- Il disco nero bordato (nodo finale), indica la terminazione.
- Possono comparire in qualunque numero all'interno di un diagramma

Transizione

- Una transizione collega tra loro due stati, è rappresentata con una freccia
- L'uscita da uno stato definisce la risposta dell'oggetto all'occorrenza di un evento, viene presa solo se la condizione è vera, e comporta l'esecuzione delle azioni specificate

eventi ::= evento | evento , eventi (disgiunzione)

azioni ::= azione | azione, azioni (sequenza)

Tutti opzionali anche se l'evento è bene che ci sia.

Solo nelle transizioni di completamento (più avanti) l'evento non serve.

Esempio stati di una lampadina

Descriviamo la vita di una lampadina

Lampadina
accesa : boolean = false
accendi() spegni()

OSSERVAZIONE: gli eventi (anche se non tutti) corrispondono alle operazioni della classe

Evento

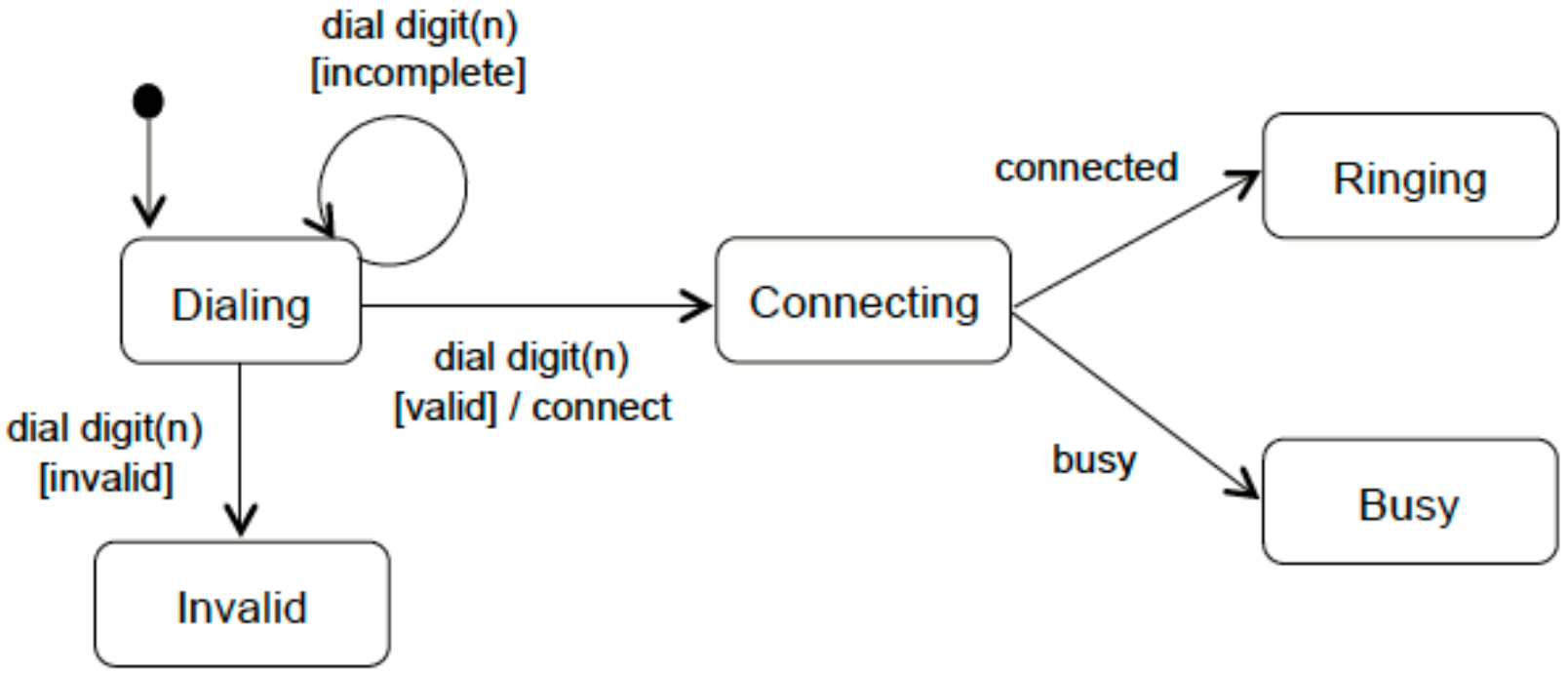
- Un evento è l'occorrenza di un fenomeno collocato nel tempo e nello spazio
- Un evento occorre istantaneamente
- Modellate qualcosa come un evento se ha delle conseguenze
- Gli eventi che arrivano in uno stato per cui non è prevista alcuna transizione vengono ignorati
- È ammesso il non–determinismo: un evento può fare da trigger a più transizioni:
  - Se le due transizioni escono dallo stesso stato, ne viene scelta una non-deterministicamente

Tipi di evento

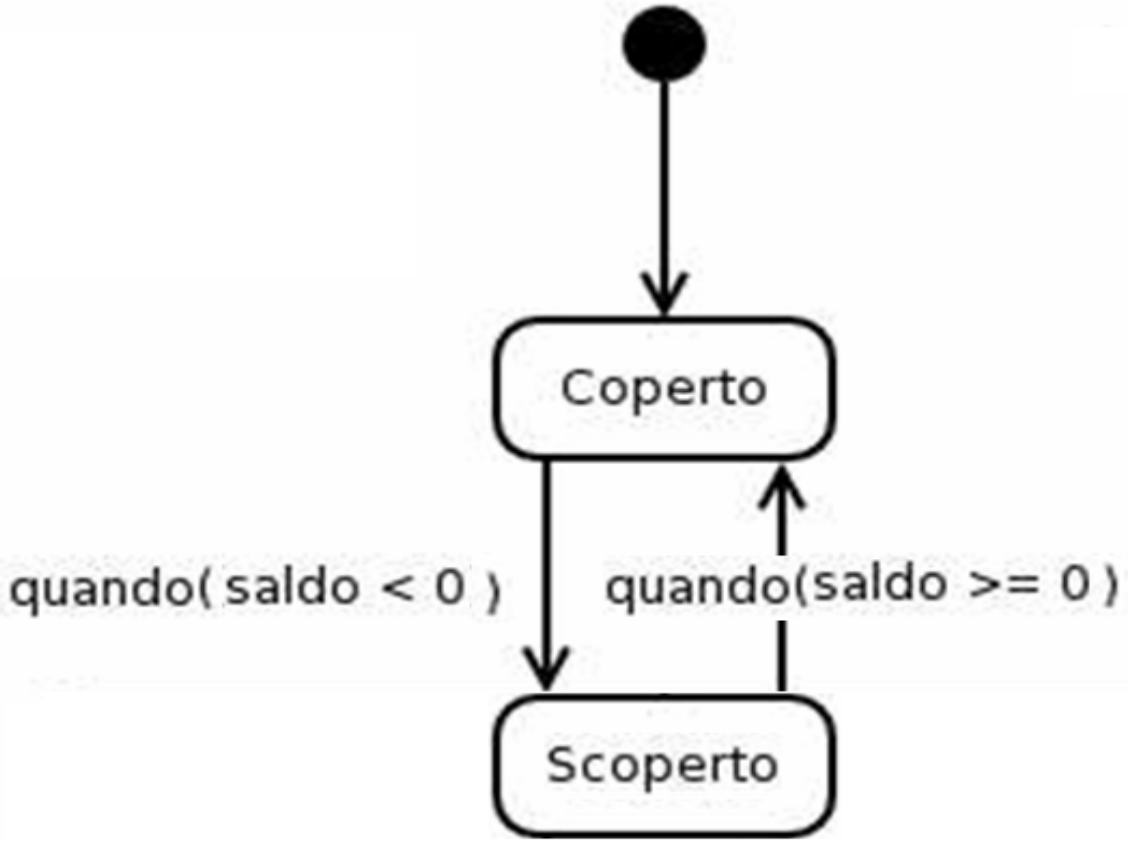
- Operazione o segnale op(a:T)
  - la transizione è abilitata quando l'oggetto (in quello stato) riceve una chiamata di metodo / un segnale con parametri (a) e tipo (T) (i parametri sono opzionali)
- Evento di variazione **quando**(exp)
  - la transizione è abilitata appena l'espressione diventa vera
  - l'espressione può indicare un tempo assoluto o una condizione su variabili
  - spesso in inglese: when(exp)
- Evento temporale **dopo**(time)
  - la transizione è abilitata dopo che l'oggetto è stato fermo "time" in quello stato
  - spesso in inglese: after(time)

Evento operazione o segnale

- Operazioni della classe telefono (dial digit(n)) o segnali (busy e connected)



Eventi di variazione (esempio)



- Un evento occorre in modo istantaneo
- una condizione non è istantanea
- è istantaneo il momento in cui diventa vera

Eventi temporali (esempio)

Dopo che l'oggetto è stato 3 mesi nello stato Scoperto, transisce nello stato Congelato

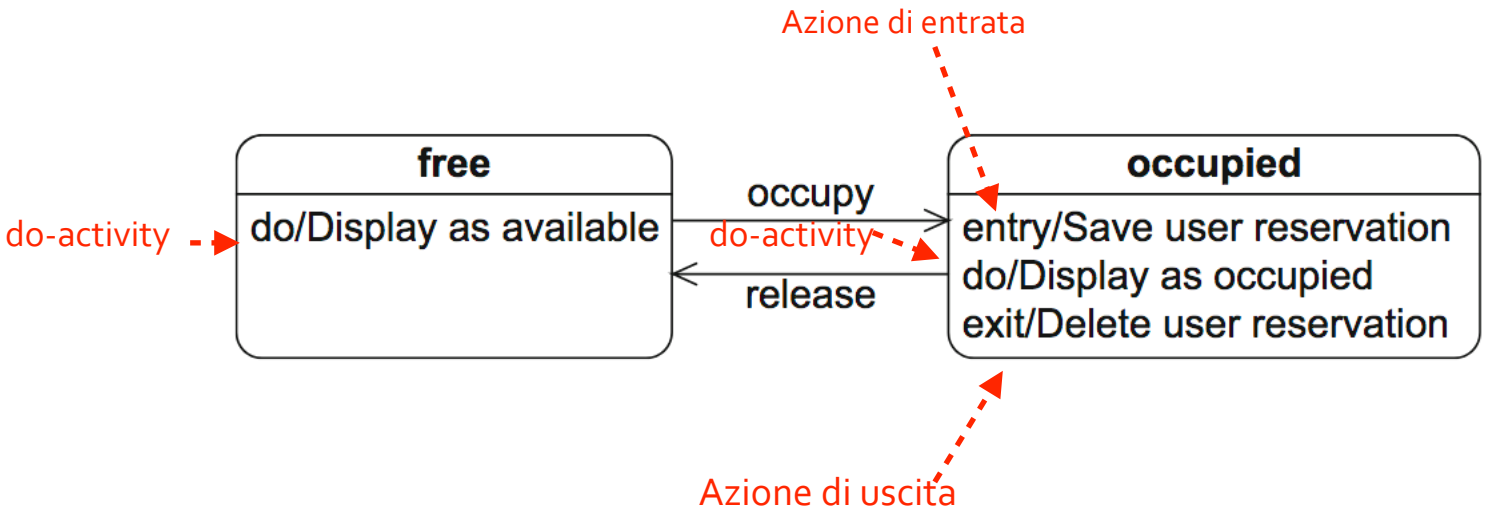
Transizioni e attività interne

- Transizione interna: risposta a un evento che causa solo l'esecuzione di azioni. Esempi:
  - Azione di entrata: eseguita all'ingresso in uno stato
  - Azione di uscita: eseguita all'uscita di uno stato
  - Transizione interna: risposta ad un evento

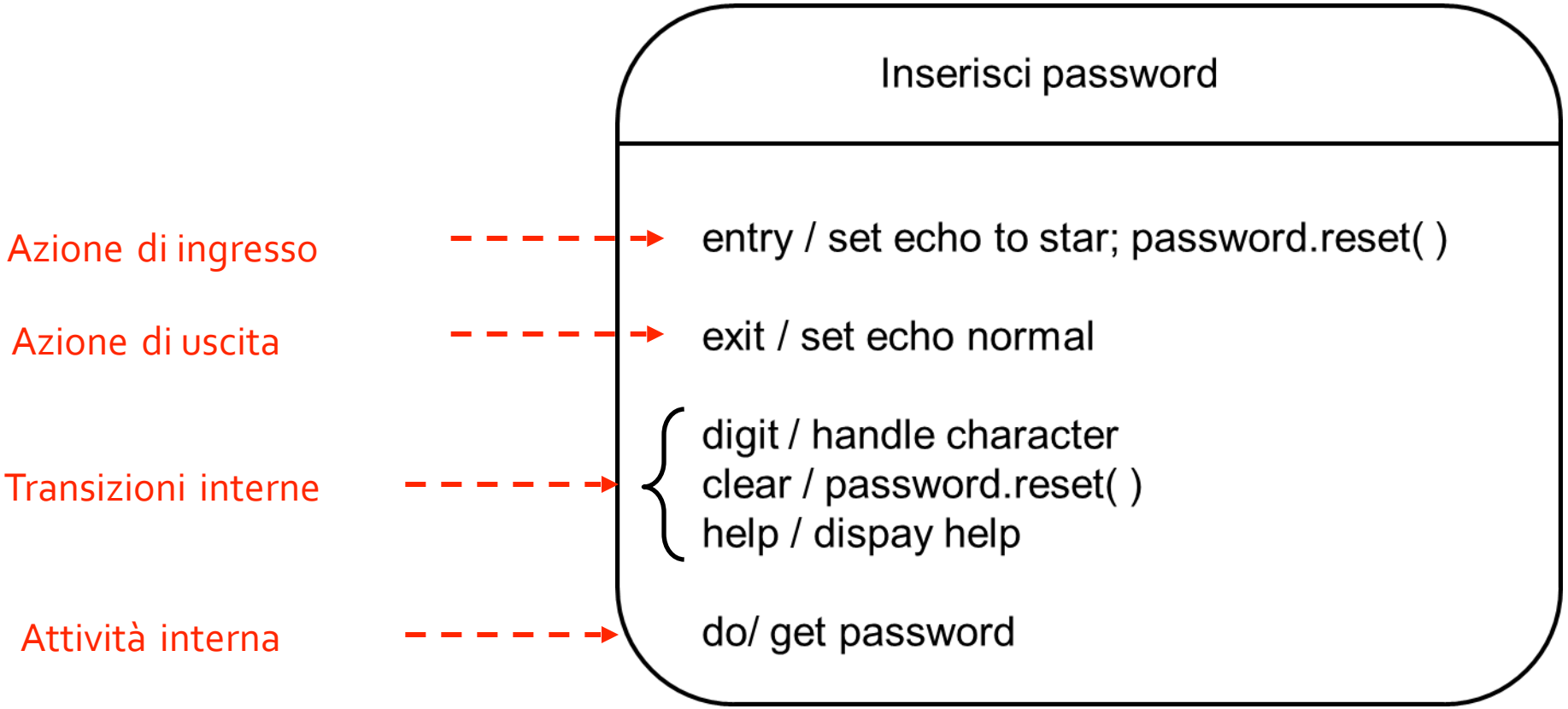
- Attività interna (Do-activity): eseguita in modo continuato mentre l'oggetto si trova in quello stato (senza necessità di un evento scatenante), al contario di tutte le altre azioni che sono atomiche:
  - consuma del tempo
  - può essere interrotta (quando un evento fa uscire dallo stato)

Esempio

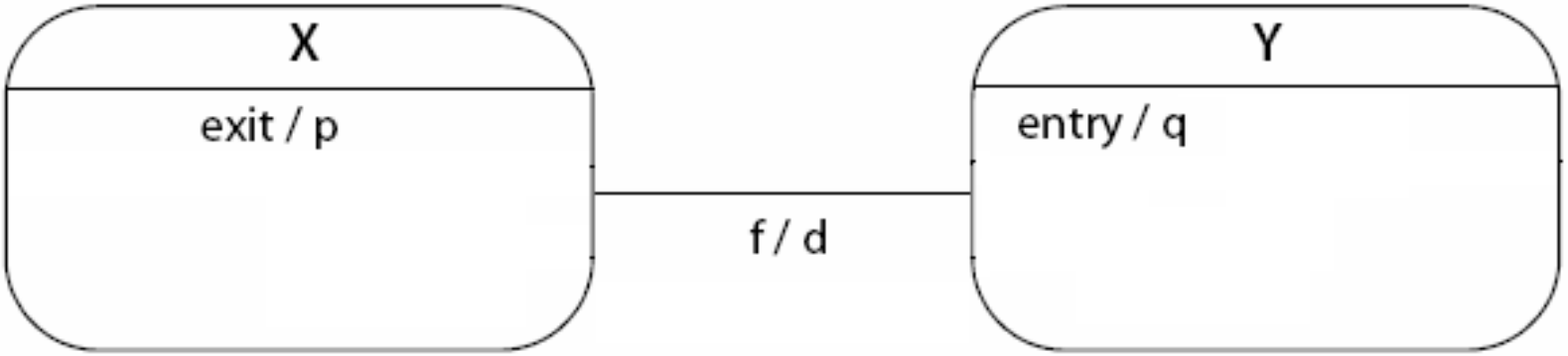
Ritorniamo sull'esempio dell'aula:



Sintassi


Inserisci password
Azione di ingresso
entry / set echo to star; password.reset( )
Azione di uscita
exit / set echo normal
Transizioni interne
digit / handle character
clear / password.reset( )
help / dispay help
Attività interna
do/ get password

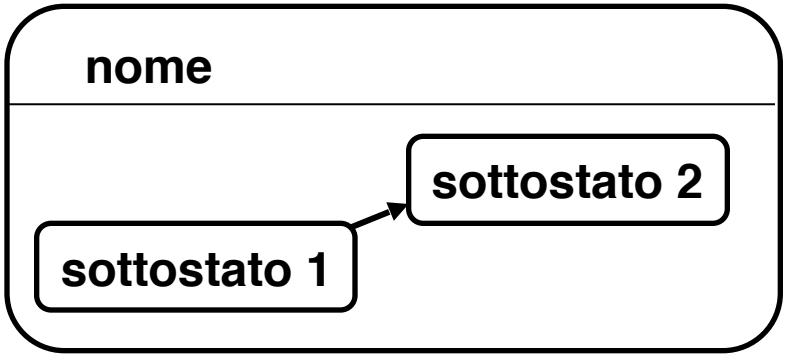
Esempi



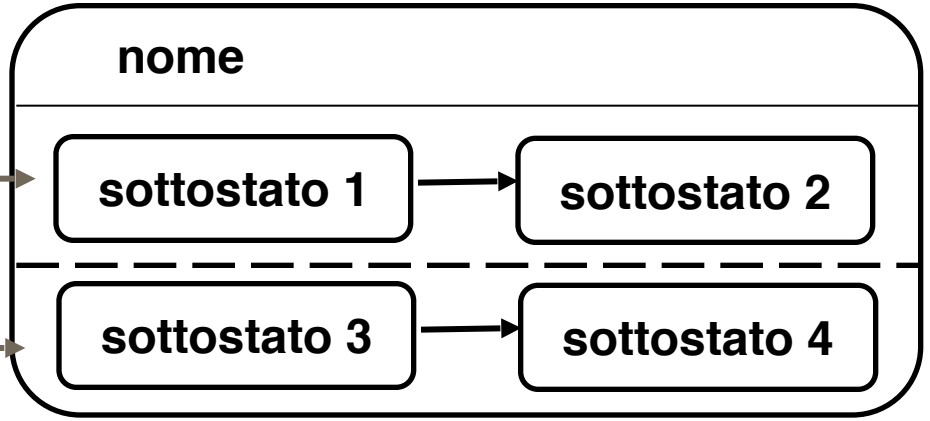
effective result: f / p; d; q

Stati compositi

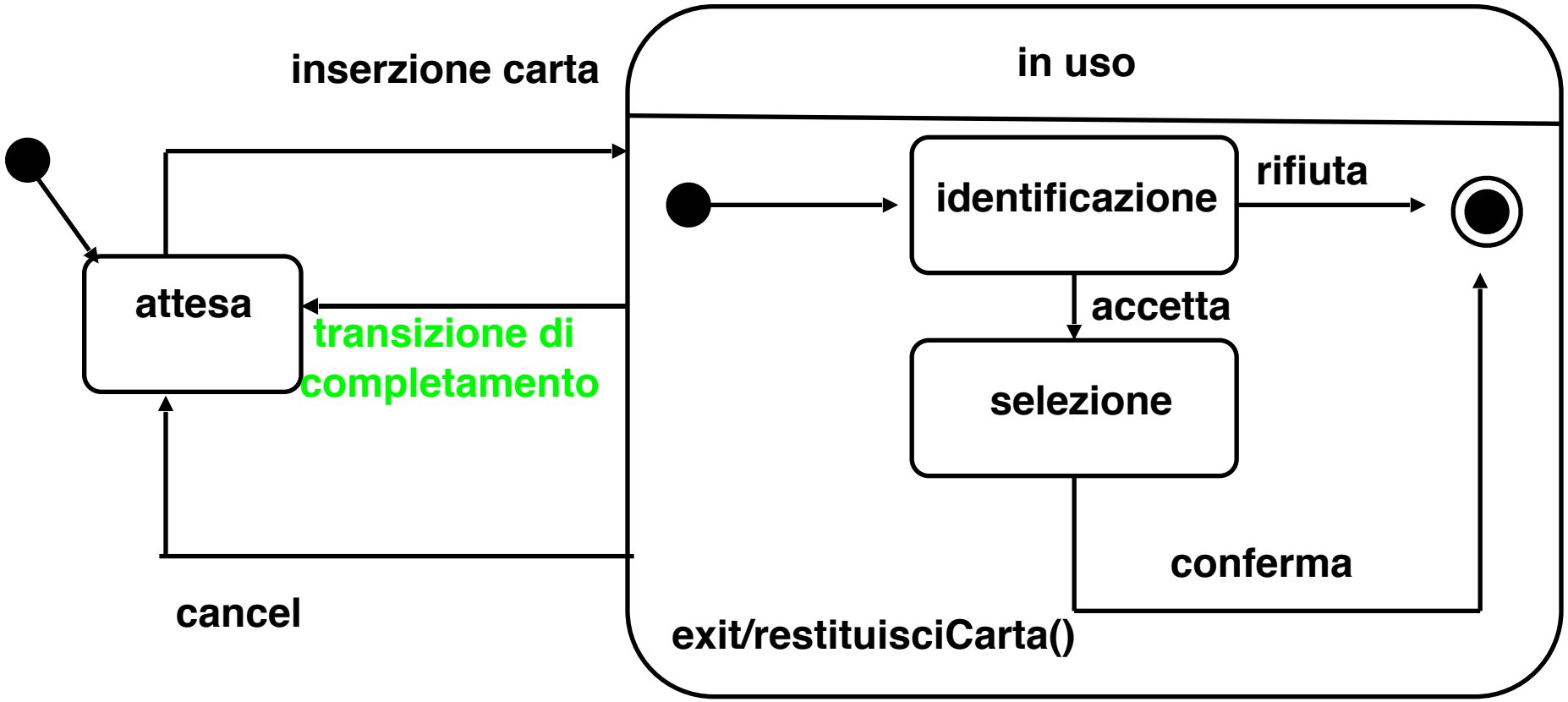
- Composito sequenziale:
- Un sottostato attivo in ogni istante



- Composito parallelo:
- sottostati attivi contemporaneamente
- uno per regione

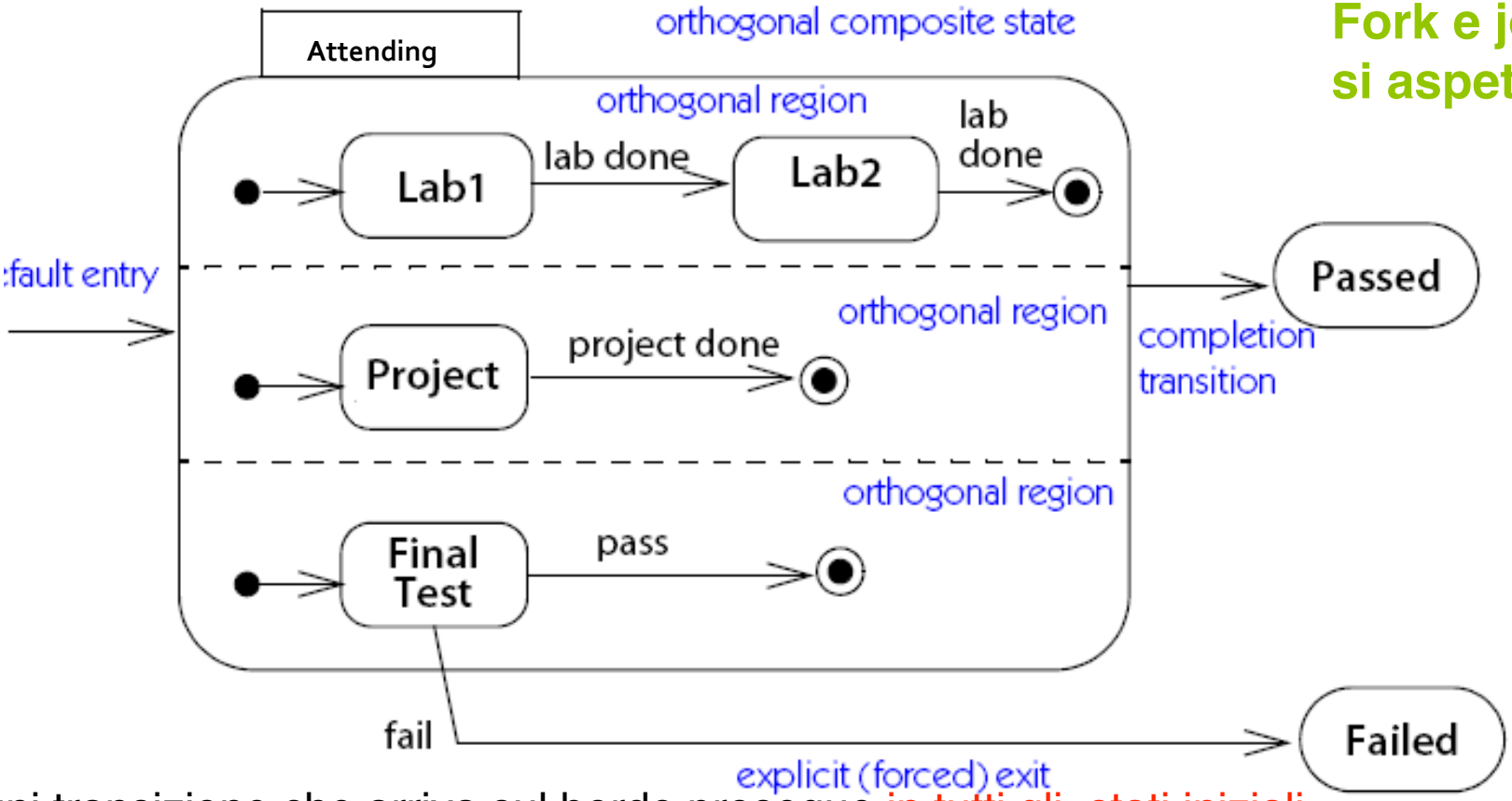


Esempio stato composito sequenziale



- Ogni transizione che arriva sul bordo prosegue nello stato iniziale
- Dallo stato finale (dopo le exit) si prosegue nella transizione di completamento
- Ogni transizione (non di completamento) che parte dal bordo si intende possibile da un qualsiasi stato interno

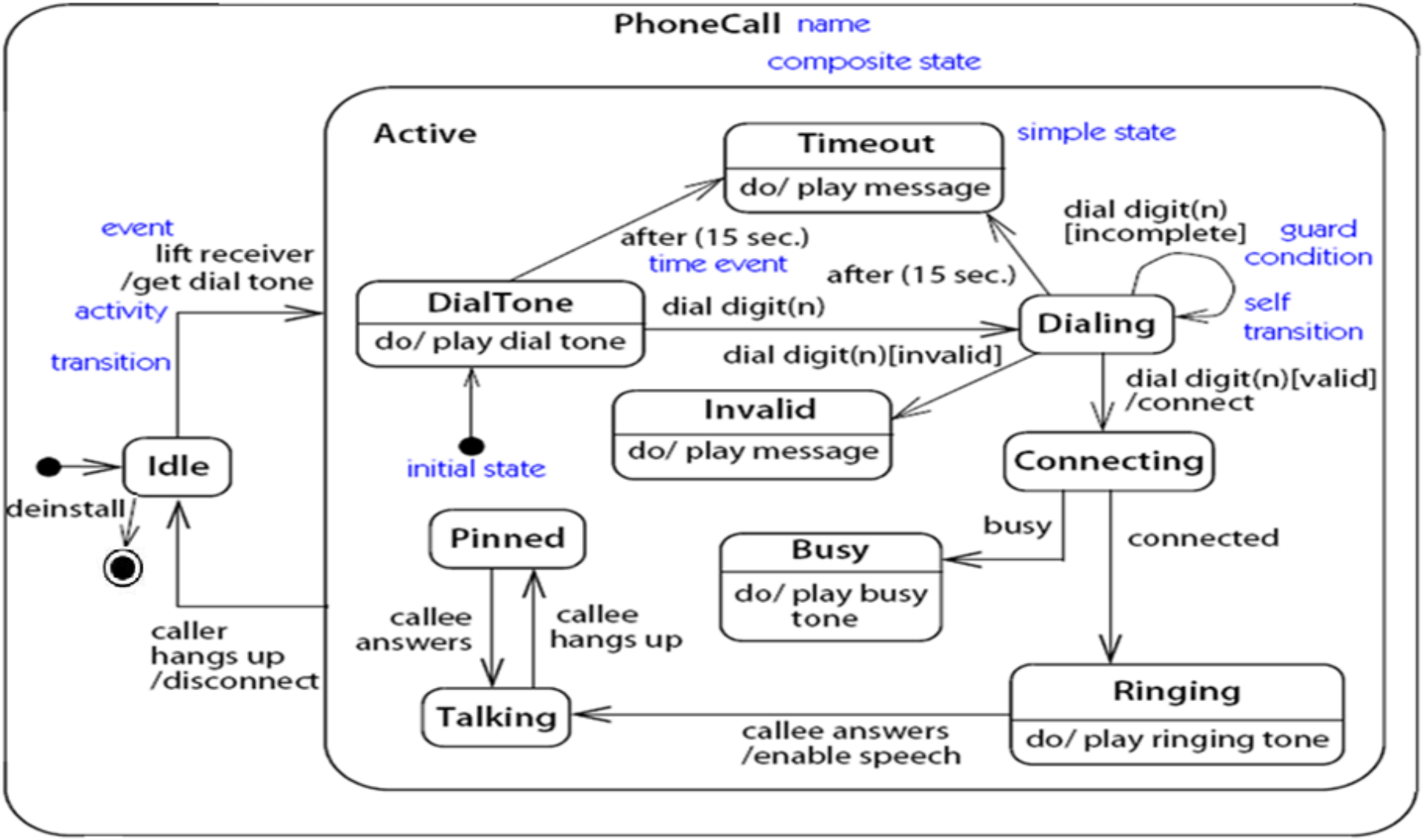
stato composito parallelo
un esempio: corso di laboratorio



Fork e join implicite: si aspetta per uscire

- Ogni transizione che arriva sul bordo prosegue in tutti gli stati iniziali
- Una volta raggiunti tutti gli stati finali si prosegue nella transizione di completamento
- Possono esserci transizioni che bucano il bordo e si intendono possibili dal solo stato interno a cui sono collegate

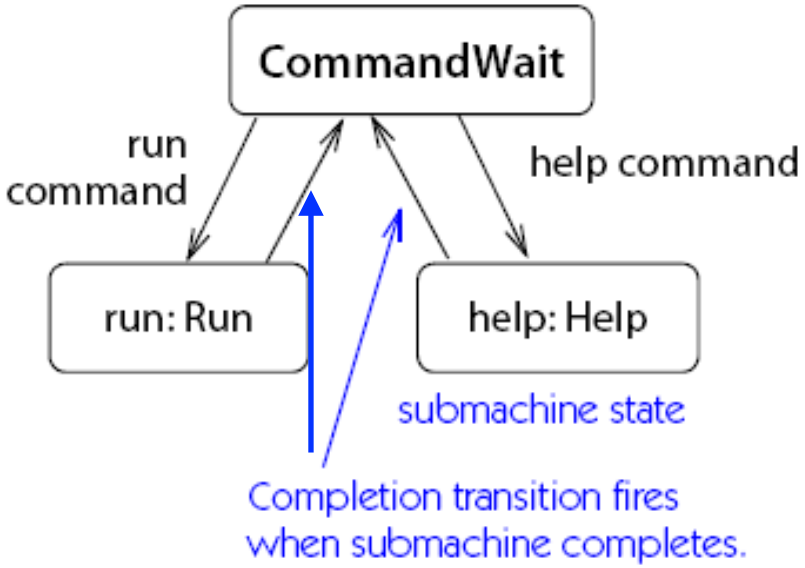
Esempio: stato composito sequenziale senza stato finale

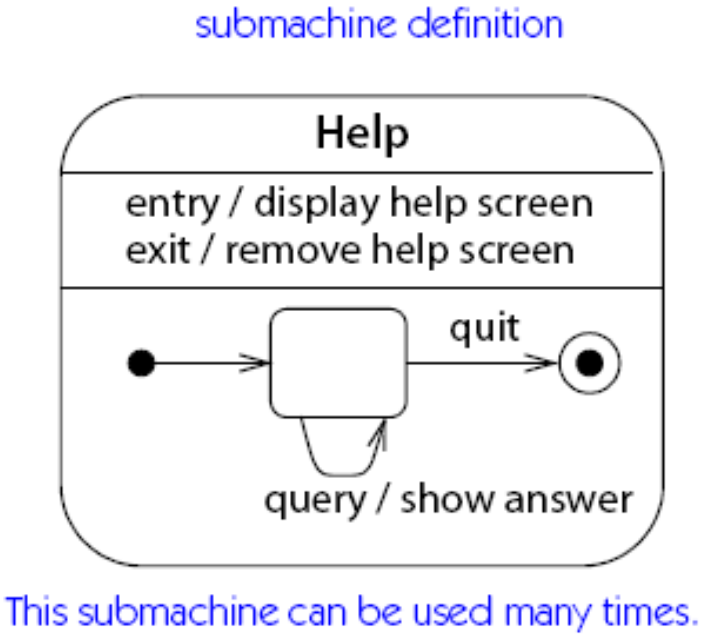


Pinned=trattenuto

Sottomacchine

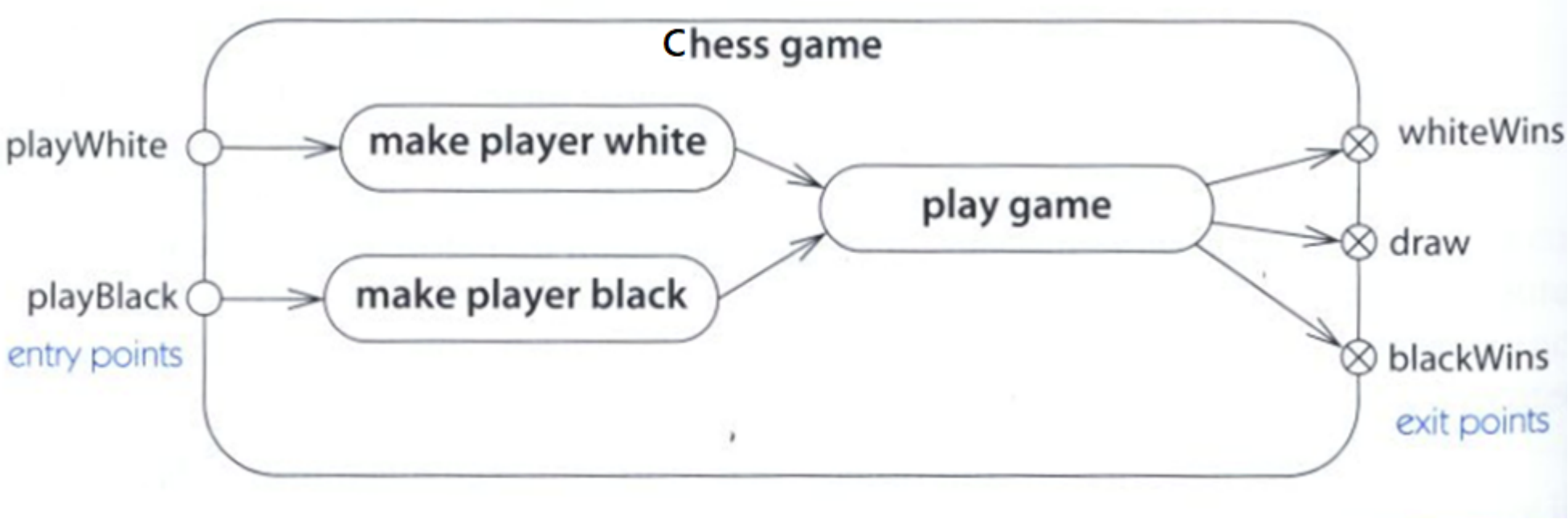
- Si usa quendo si vuole descrivere uno stato composito in un diagramma a parte, per leggibilità o per definirlo una volta per tutte e riusarlo in più contesti.
- La sottomacchina ha un nome (tipo), le istanze di uso si indicano con nomeIstanza:Tipo



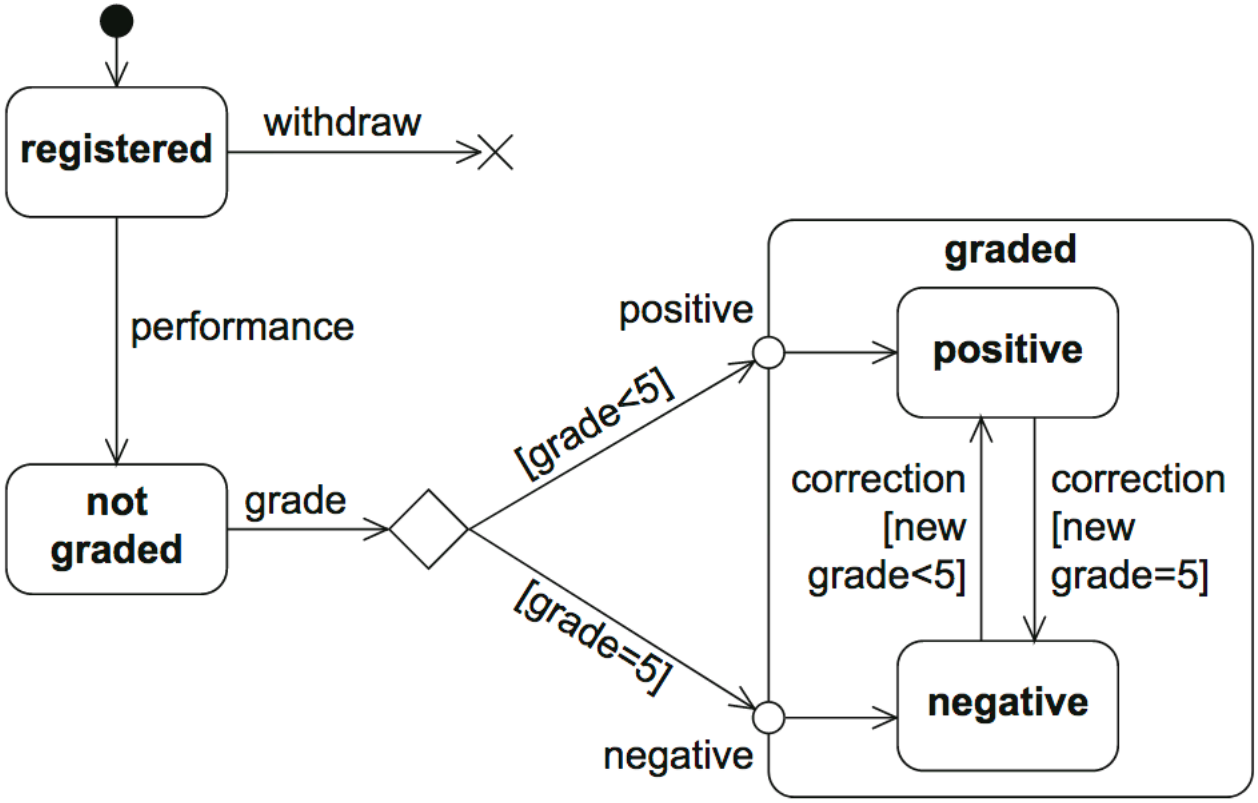


Sottomacchine: entry e exit points

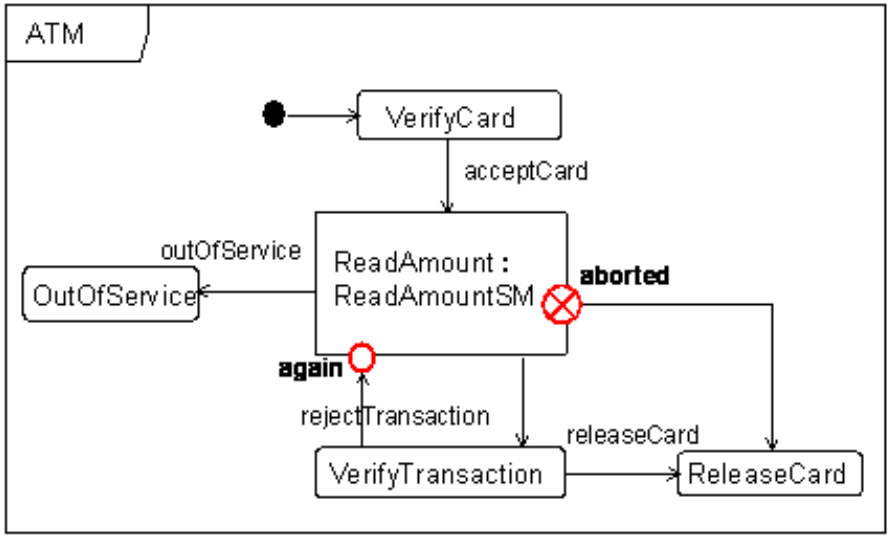
Una sottomacchina può definire entry and exit points che servono per collegare le transizioni della macchina principale

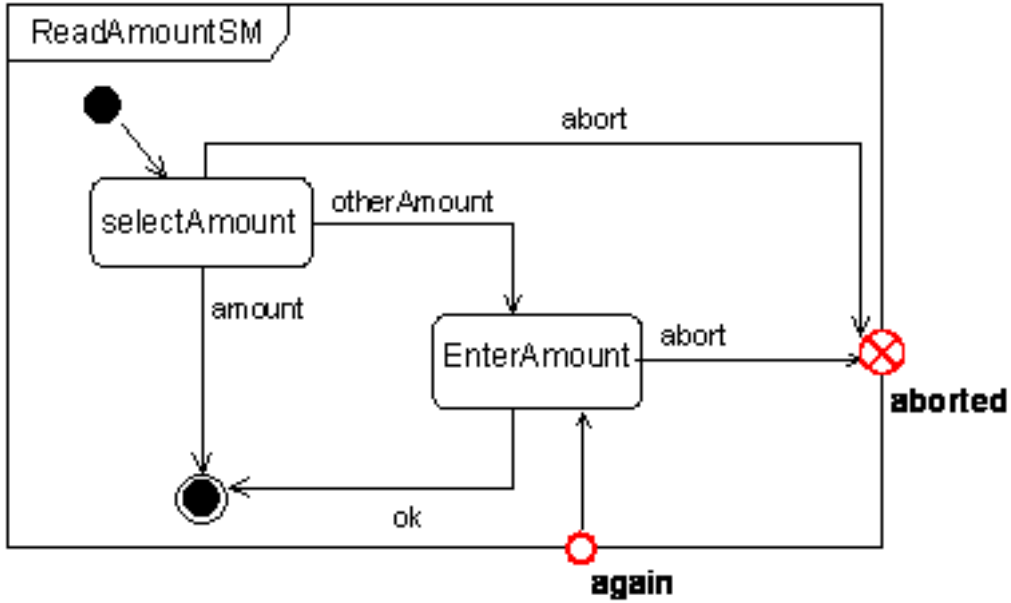


Esempio


registered
withdraw
performance
not graded
grade
[grade<5]
[grade=5]
positive
negative
graded
positive
negative
correction [new grade<5]
correction [new grade=5]

Un altro esempio



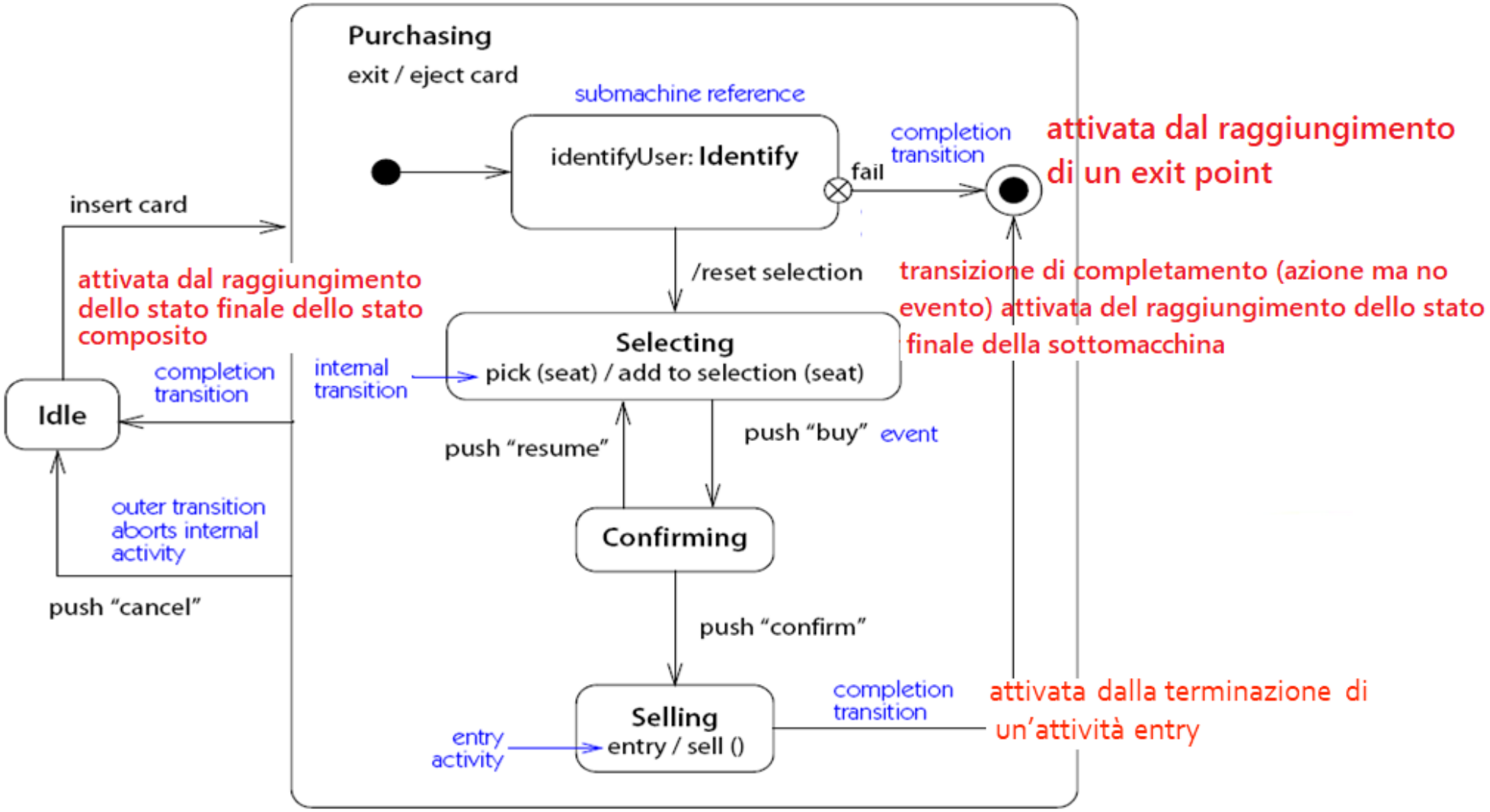


Transizioni di completamento: riassunto

- Senza evento, scattano al raggiungimento
  - Della terminazione di un'attività composita, i.e. al raggiungimento
    - Dello stato finale in un stato composito non-ortogonale
    - Degli stati finali di tutte le regioni ortogonali di un stato composito
    - Di un exit point
  - Alla terminazione di entry e/o di do activity (la exit activity viene eseguita quando scatta la transizione di completamento)
  - Di uno pseudo–stato giunzione (lo vedremo in un attimo)

- Hanno priorità sugli eventi normali

Transizioni di completamento: esempio



Altri tipi di stato (pseudostati)

Giunzione 

Decisione 

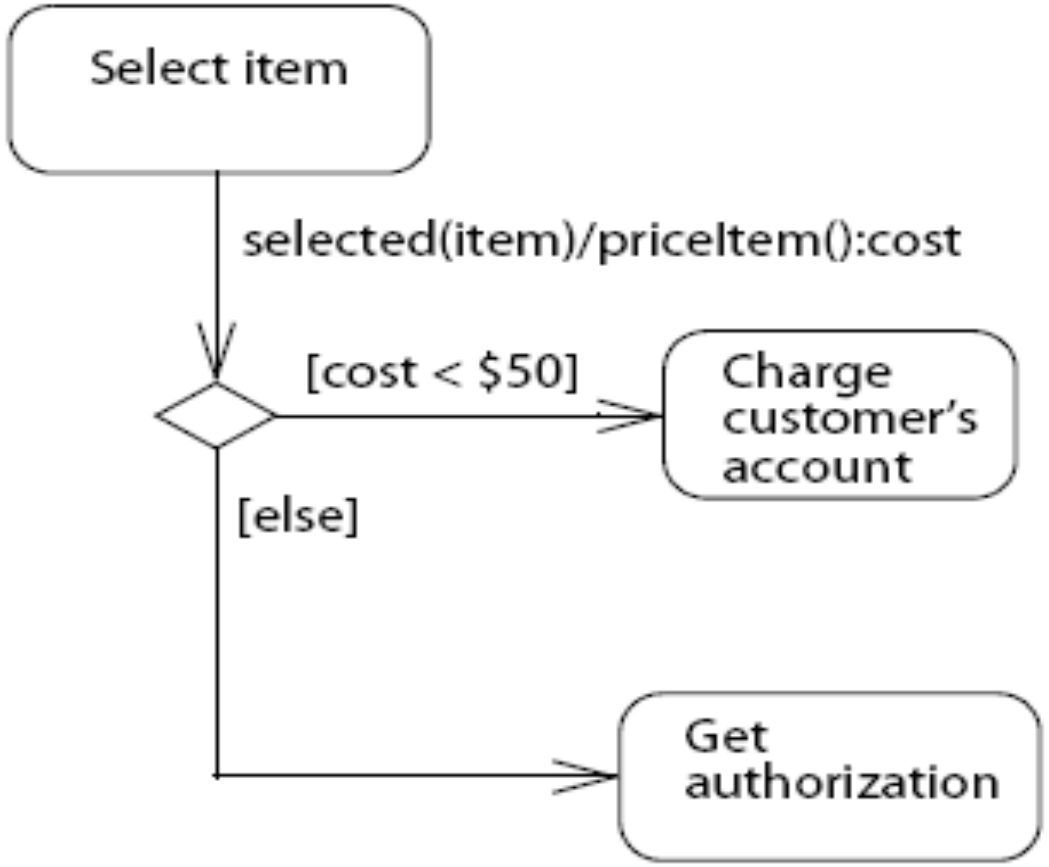
Fork, join

Storia 

Iniziale 

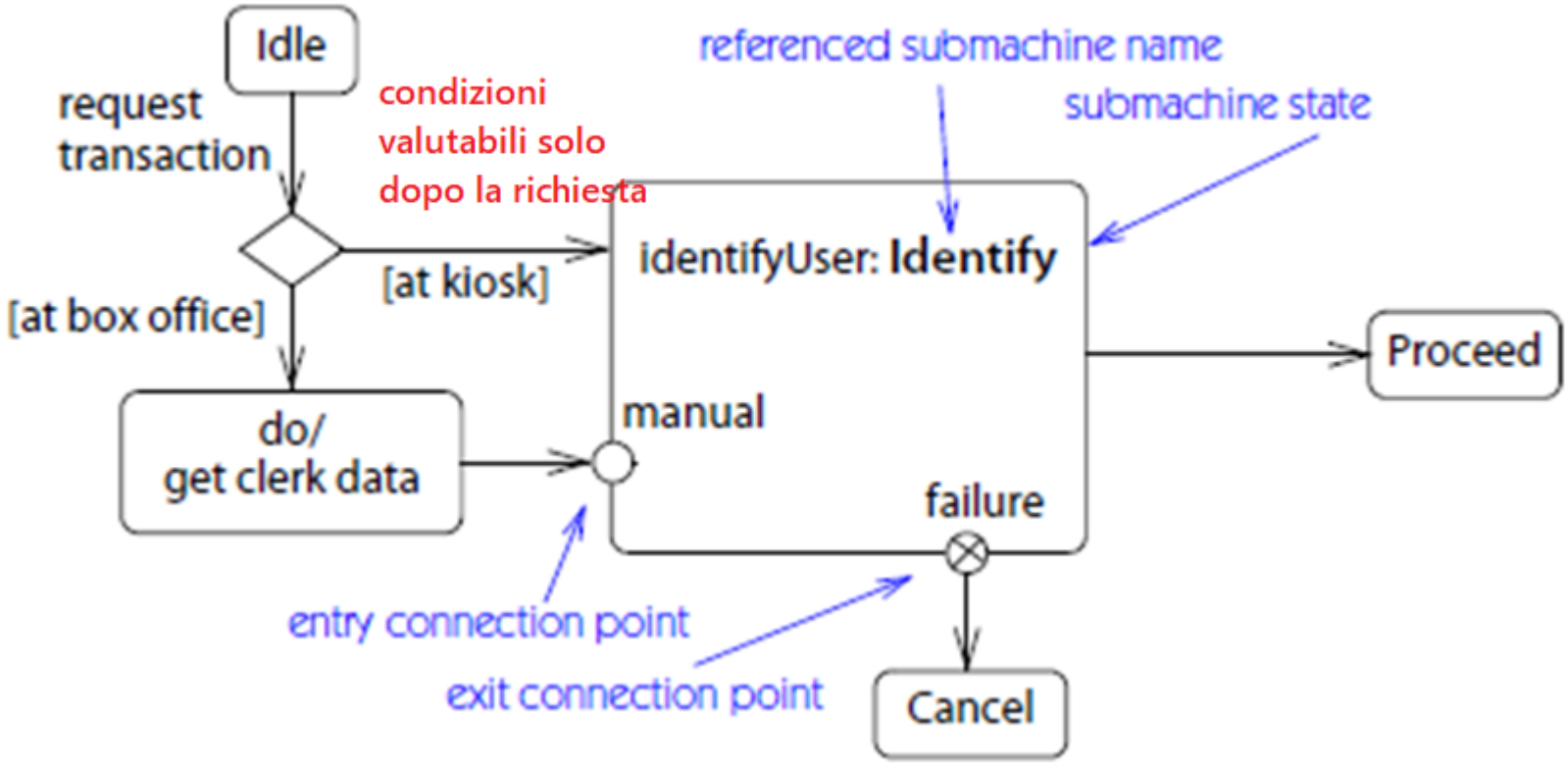
Finale 

Esempio di choice



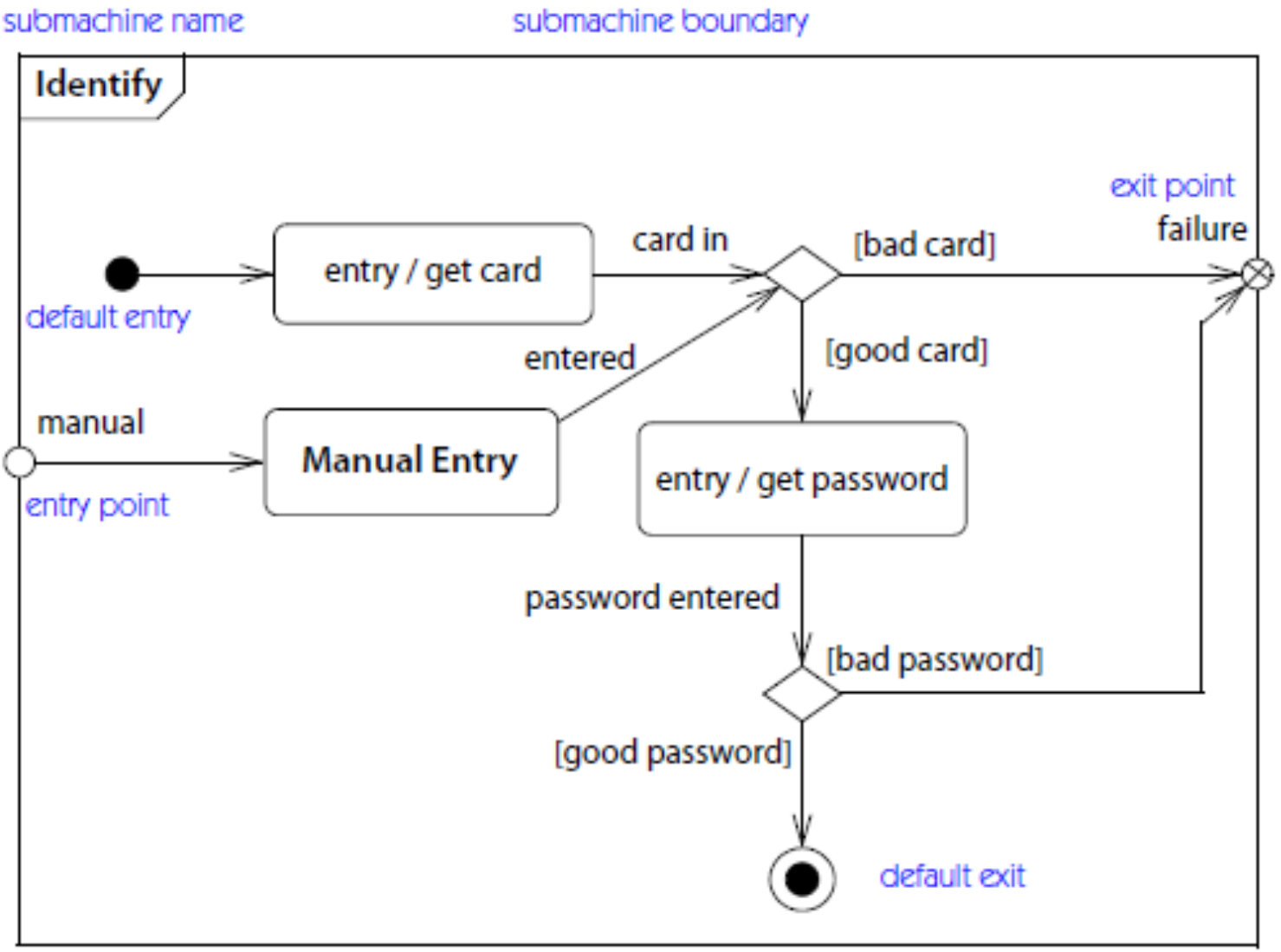
- Condizioni valutate dinamicamente
- Come per la choice dei diagrammi di attività:
  - La disgiunzione delle guardie deve valere "true"
  - È ammesso il non-determinismo

Variazione sul tema
Acquisto biglietti a clienti con account: identificazione di default leggendo una carta, manuale, con inserimento nome



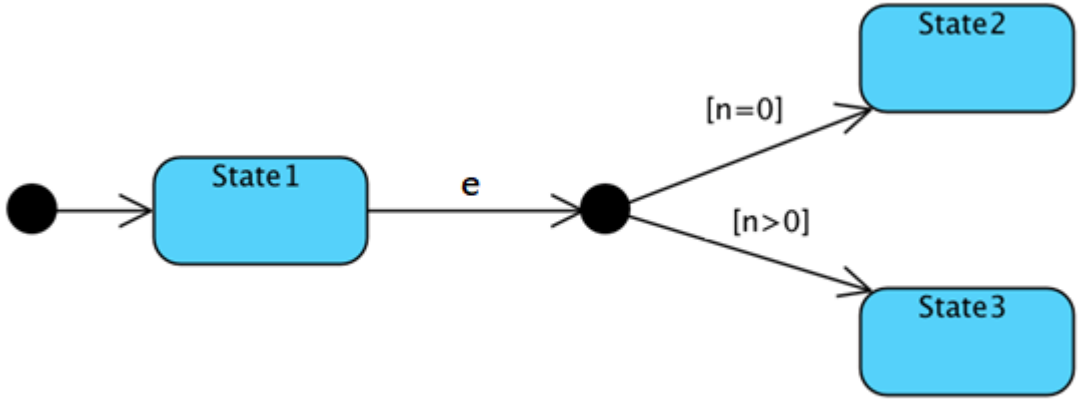
O l'identificazione avviene leggendo la carta o recandosi da un addetto il quale inserisce i dati della carta

Esempio acquisto biglietti: sottomacchina Identify



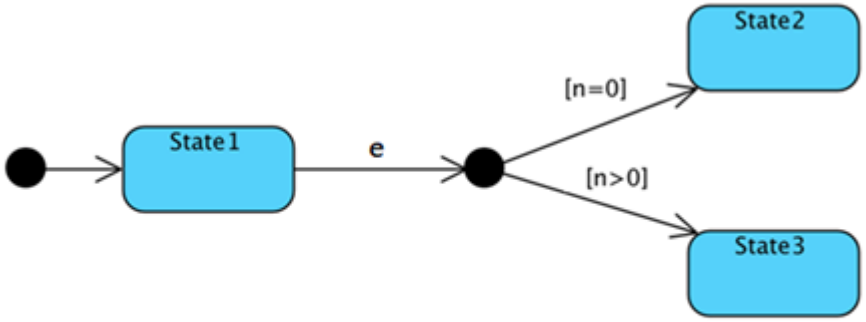
Pseudo-stato di giunzione

- Uno pseudo-stato da cui escono e/o entrano due o più transizioni. Eventuali condizioni sono valutabili in modo statico (prima dell'evento e)
- Se n<o l'evento e viene ignorato e si rimane nello stato 1



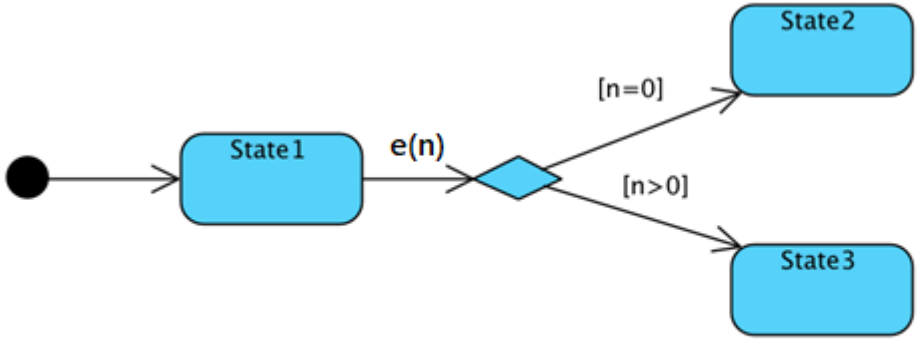
Giunzione vs choice

- Giunzione (statica)



Le guardie sono valutate prima di uscire da State1. Se n<0, l'evento e viene ignorato e nessuna transizione viene presa

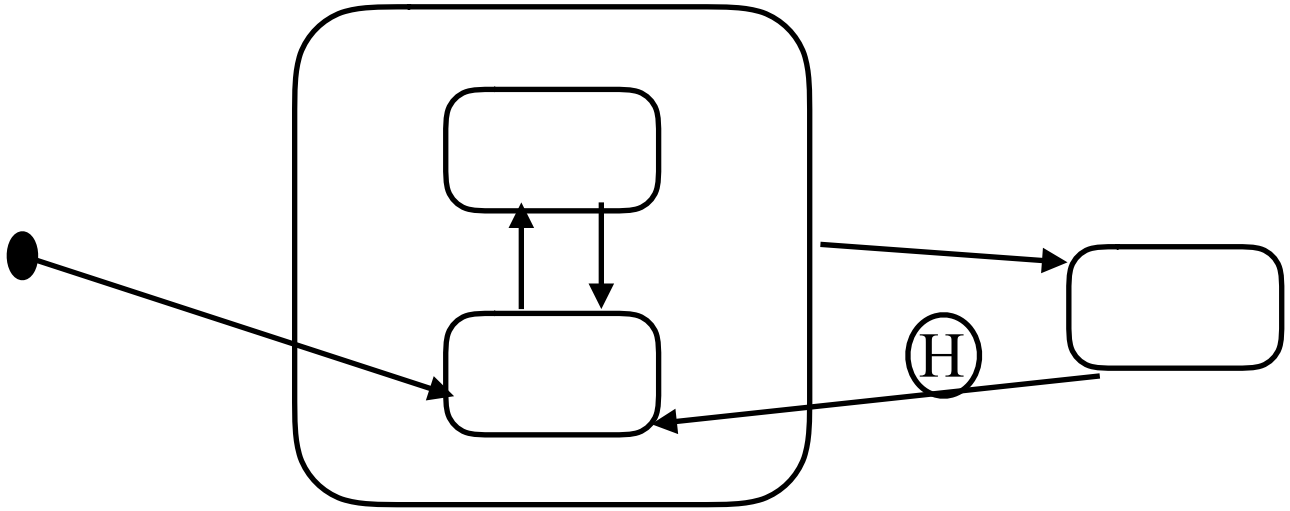
- Choice (dinamica)



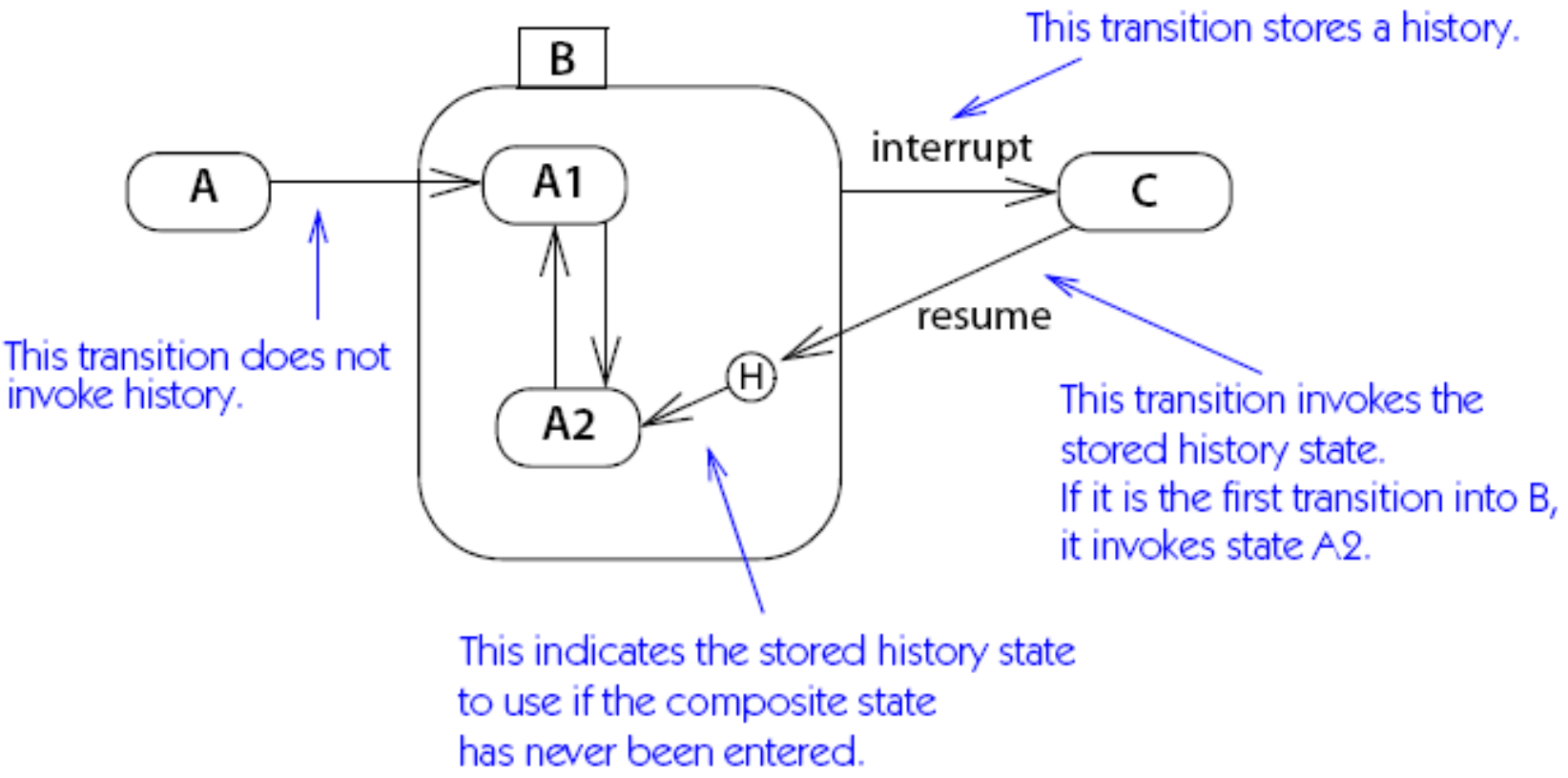
Le guardie sono valutate dopo e(n). In questo esempio occorre avere garanzia che n sia maggiore o uguale a zero

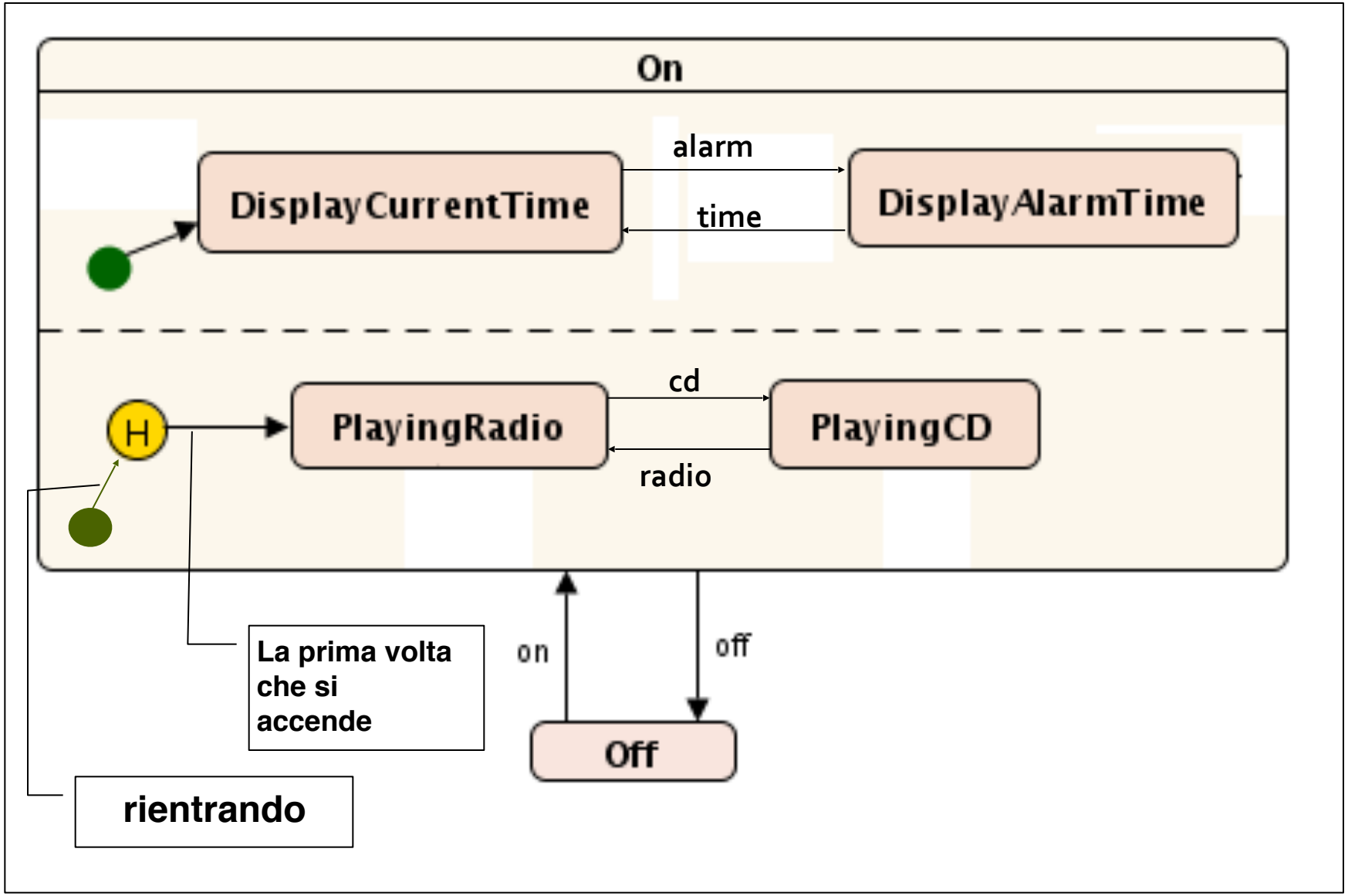
History State

Gli History state servono in presenza di una transizione esterna da uno stato composito, per ricordarsi quale era l'ultimo stato attivo, in modo da poter rientrare esattamente in quello stato

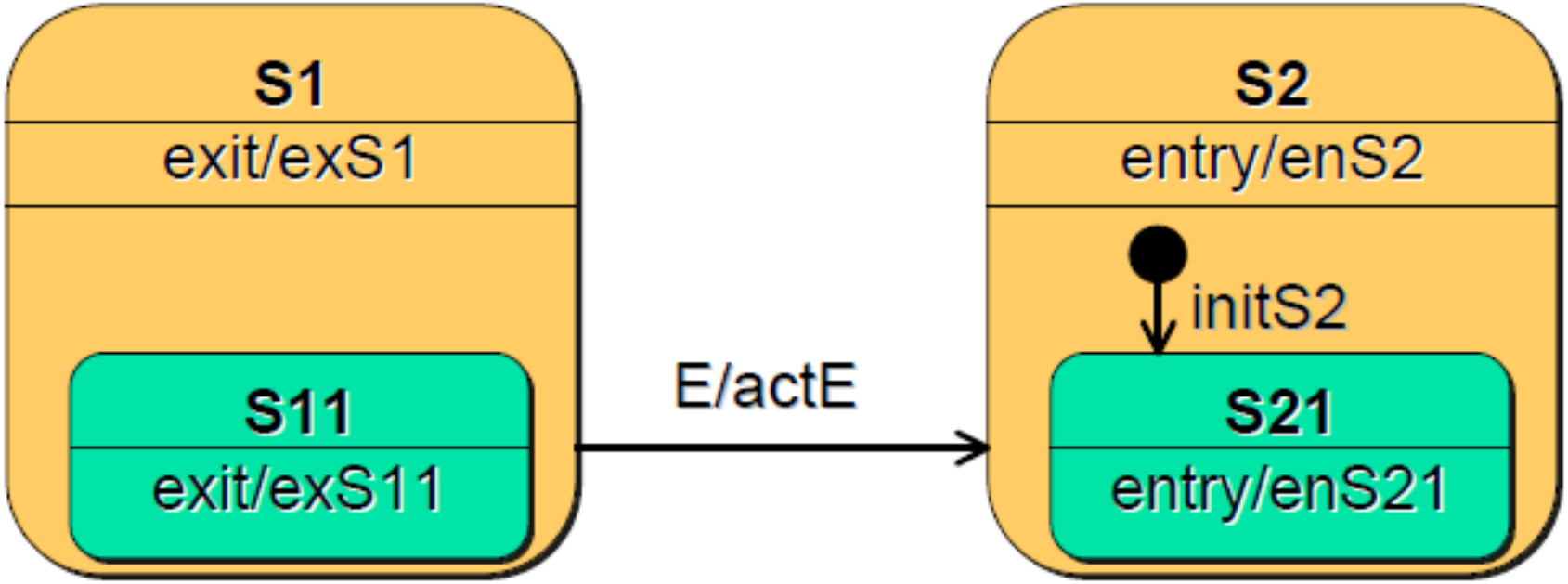


History state




On
DisplayCurrentTime
alarm
time
DisplayAlarmTime
H
PlayingRadio
cd
radio
PlayingCD
La prima volta che si accende
on
off
Off
rientrando

Esempio



Actions execution sequence:

exS11 ⇨ exS1 ⇨ actE ⇨ enS2 ⇨ initS2 ⇨ enS21

Descrivere il modello dinamico: macchina a stati o attività?

- Come scegliere il diagramma più appropriato:
  - Se il focus è
    - mettere in ordine in un insieme di azioni da fare → attività
    - mostrare l'evoluzione di un oggetto in risposta a eventi → stati
  - Il diagramma di macchina a stati parla dell'evoluzione nel tempo delle istanze di un classificatore
  - il diagramma di attività parta di un'agenda di azioni da fare.

Descrivere il modello dinamico: nomi degli stati e delle azioni

- Nomi degli stati:
  - aggettivi: attivo,
  - participi passati: accesa, spenta, pinned
  - gerundi: dialing, connecting
  - Altri: inAttesa
- Nomi delle azioni:
  - verbi all'indicativo, imperativo o infinito: crea, inviare
  - sostantivi che indicano un'azione: interrogazione DB

- Non è una regola fissa e spesso nella pratica si disattende (eccezioni anche negli esempi visti), ma seguire la regola è un buon metodo per costruire i diagrammi correttamente
  - Errore comune nei compiti confondere stati e azioni

Syllabus

- UML@Classrom: Cap 5

Appendice

Da UML Reference Manual

Tipi di eventi

Table 7-1: *Kinds of Events*

Event Type	*Description*	*Syntax*
call event	Receipt of an explicit synchronous call request by an object	op (a:T)
change event	A change in value of a Boolean expression	**when** (exp)
signal event	Receipt of an explicit, named, asynchronous communication among objects	sname (a:T)
time event	The arrival of an absolute time or the passage of a relative amount of time	**after** (time)

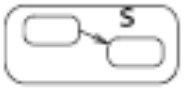
Tipi di tranzizioni

Table 7-2: *Kinds of Transitions and Implicit Effects*

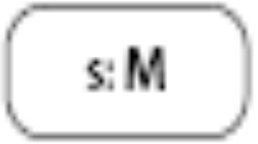
Transition Kind	*Description*	*Syntax*
entry transition	The specification of an entry activity that is executed when a state is entered	entry/ activity
exit transition	The specification of an exit activity that is executed when a state is exited	exit/ activity
external transition	A response to an event that causes a change of state or a self-transition, together with a specified effect. It may also cause the execution of exit and/ or entry activities for states that are exited or entered.	e(a:T)[guard]/activity
internal transition	A response to an event that causes the execution of an effect but does not cause a change of state or execution of exit or entry activities	e(a:T)[guard]/activity

Tipi di stato

Table 7-3: *Kinds of States*

State Kind	*Description*	*Notation*
simple state	A state with no substructure	
orthogonal state	A state that is divided into two or more regions. One direct substate from each region is concurrently active when the composite state is active.	
nonorthogonal state	A composite state that contains one or more direct substates, exactly one of which is active at one time when the composite state is active	
initial state	A pseudostate that indicates the starting state when the enclosing state is invoked	
final state	A special state whose activation indicates the enclosing state has completed activity	
terminate	A special state whose activation terminates execution of the object owning the state machine	
junction	A pseudostate that chains transition segments into a single run-to-completion transition	
choice	A pseudostate that performs a dynamic branch within a single run-to-completion transition	
history state	A pseudostate whose activation restores the previously active state within a composite state	

Sottomacchine

State Kind	*Description*	*Notation*
submachine state	A state that references a state machine definition, which conceptually replaces the submachine state	
entry point	A externally visible pseudostate within a state machine that identifies an internal state as a target	
exit point	A externally visible pseudostate within a state machine that identifies an internal state as a source