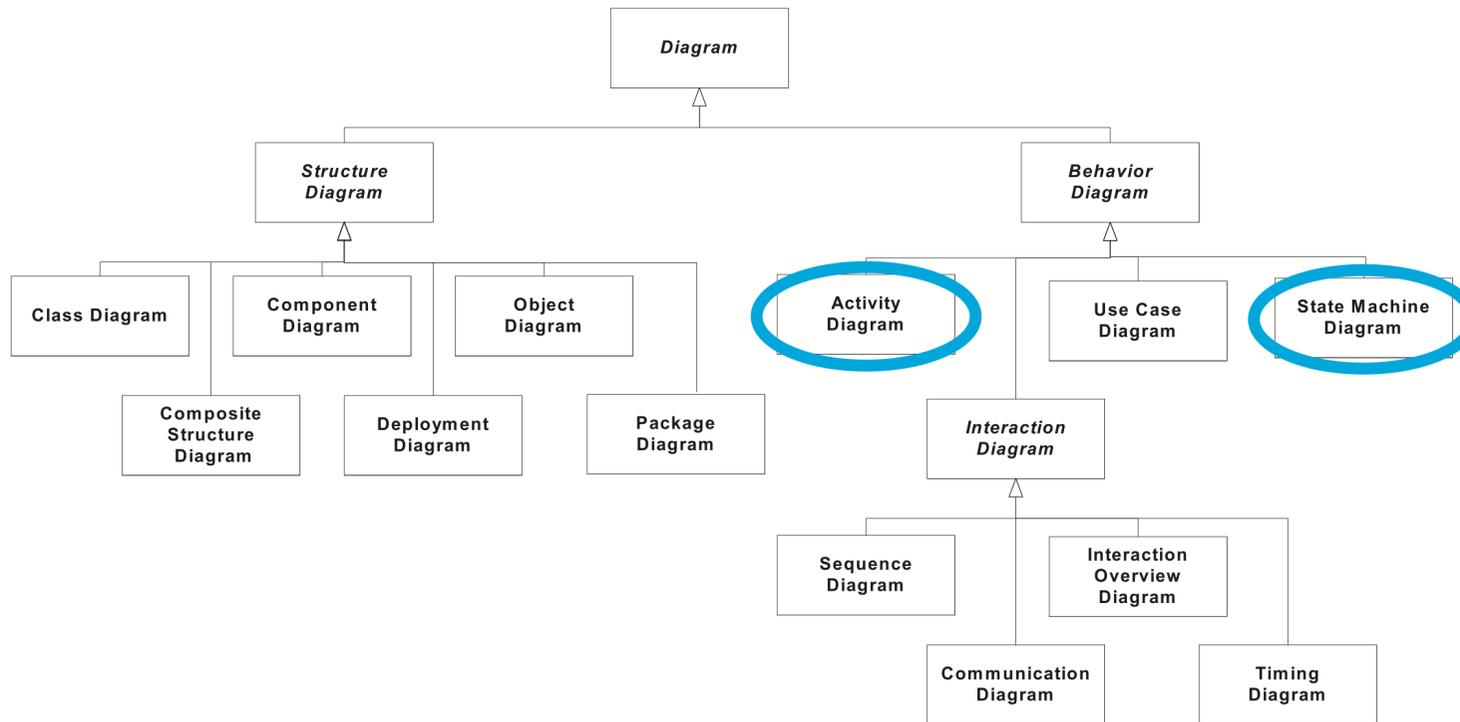


MODELLAZIONE CON UML:
DIAGRAMMA DEGLI STATI

Prof. Mariacarla Staffa

TASSONOMIA DEI DIAGRAMMI UML

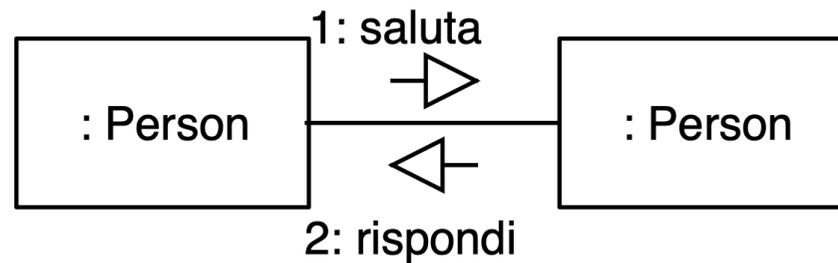


DISTINZIONE TRA DIAGRAMMA DELLE ATTIVITÀ E DIAGRAMMA DEGLI STATI

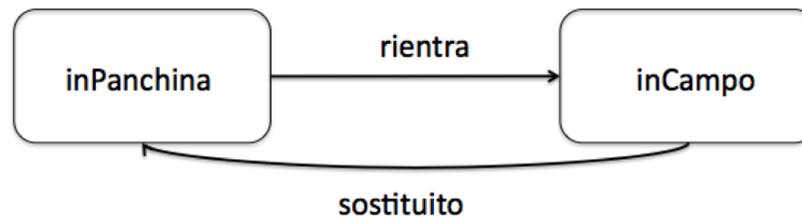
- I diagrammi di attività (activity diagram) e stato (state machine diagram) sono diagrammi che descrivono comportamento.
- Il diagramma di attività modella un comportamento (che riguarda una o più entità) come un insieme di azioni organizzate secondo un flusso.
- Il diagramma di stato modella il comportamento (generalmente di una sola entità) come variazioni del suo stato interno.

UN PASSO INDIETRO: INTERAZIONE VS. MACCHINA A STATI

- Interazione: un insieme di oggetti che si scambiano messaggi per raggiungere un dato obiettivo

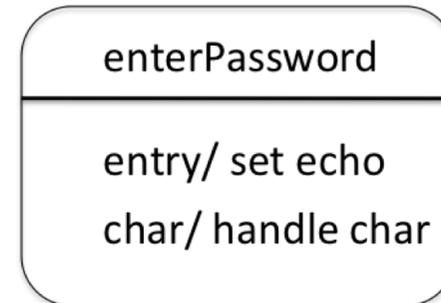


- Macchina a stati: descrive la sequenza di stati in cui si trova un oggetto durante il suo ciclo di vita e in risposta a eventi



MACCHINA A STATI

- Qualunque classificatore UML può essere associato a una macchina a stati che descrive il funzionamento delle sue istanze.
- Uno stato è una condizione o situazione nella vita di un oggetto in cui esso:
 - soddisfa una condizione,
 - esegue un'attività o
 - aspetta un evento



MACCHINA A STATI

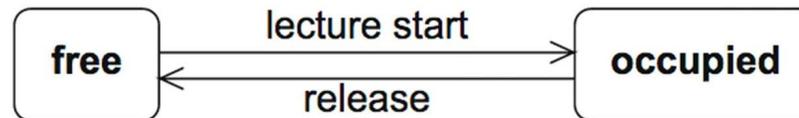
- Una macchina a stati è un grafo stati transizioni che descrive il comportamento delle istanze di una classe
 - in generale di un classificatore
- Occorre individuare gli stati significativi in cui si può trovare un oggetto durante la sua vita
- Inoltre dobbiamo descrivere come da ciascuno di questi stati l'oggetto può passare (transire) in un altro

Le transizioni avvengono in risposta al verificarsi di un evento. Esempio di eventi:

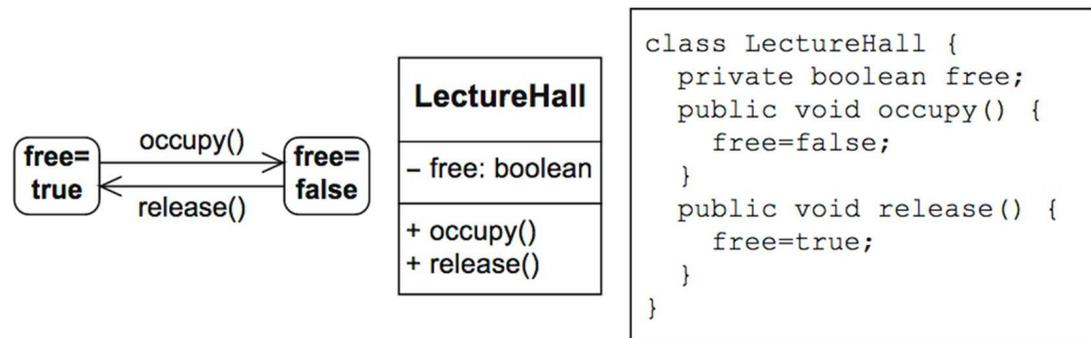
- messaggi inviati da altri oggetti
- eventi generati internamente

IMPORTANTE IL LIVELLO DI DETTAGLIO

Per modellare un'aula



verrà poi implementato come



SEMANTICA

- La macchina a stati riceve occorrenze di eventi che vengono salvati su una coda ed estratti uno alla volta.
- La semantica di questi eventi è di tipo run-to-completion: l'occorrenza di un evento viene estratta solo dopo che la macchina a stati ha finito di processare quella precedente.
- Se ci sono più transizioni eseguibili in un dato momento (es. 2 transizioni dallo stesso stato con lo stesso evento e due condizioni diverse, entrambe vere), solo una viene eseguita.

SINTASSI

Stato:

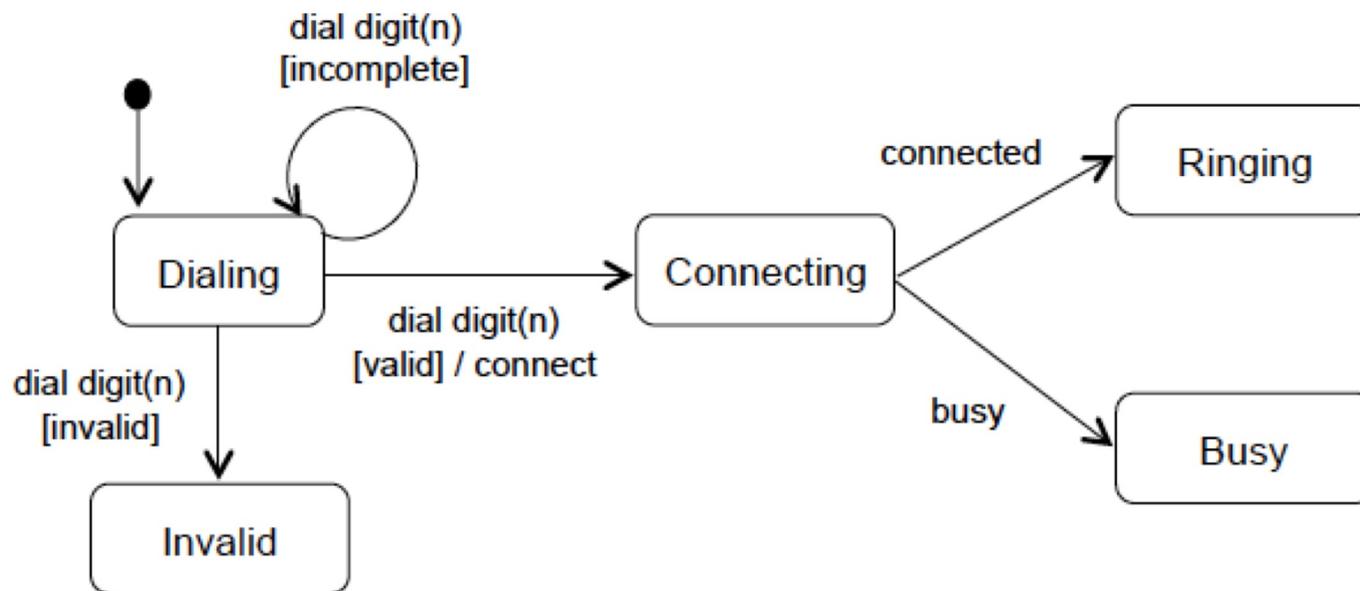
Stato iniziale: ●

Stato finale: ⊙

TRANSIZIONI

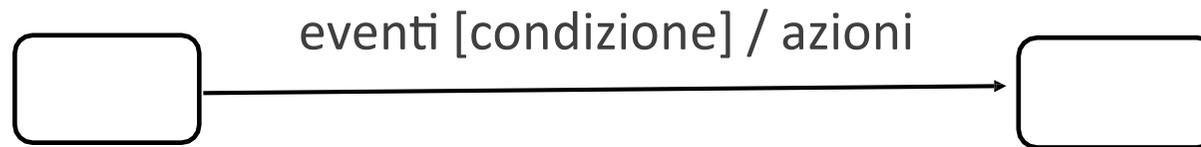
- Ogni transizione, oltre allo stato *origine* e *destinazione*, può specificare:
 - Event: un 'trigger' che attiva il passaggio di stato
 - Guard: una condizione che, se vera, permette il passaggio di stato
 - Action: un'azione che risulta dal cambio di stato
 - Sintassi: `event[guard]/action`
- La transizione avviene come risposta a uno degli eventi (quando la guardia è vera), e al momento della transizione il contesto esegue l'azione specificata
- Sono tutti opzionali

UN ESEMPIO



TRANSIZIONE

- L'uscita da uno stato
 - definisce la risposta dell'oggetto all'occorrenza di un **evento**, viene presa solo se
 - la **condizione** è vera quando occorre l'evento comporta l'esecuzione delle **azioni**
 - specificate



eventi ::= evento | evento , eventi (disgiunzione) azioni ::= azione
| azione, azioni (sequenza)

Tutti opzionali anche tranne l'evento

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

ESEMPIO: LAMPADINA

Descriviamo la vita di una lampadina

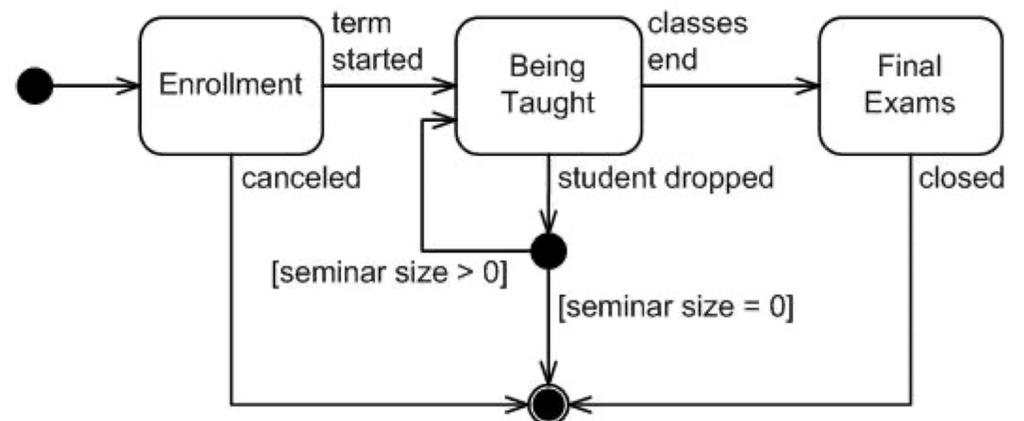
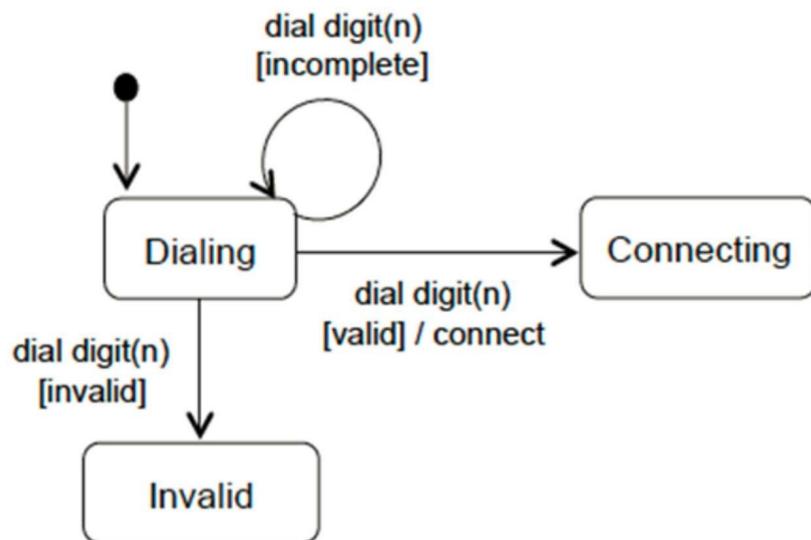


Lampadina
accesa : boolean = false
accendi() spegni()

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

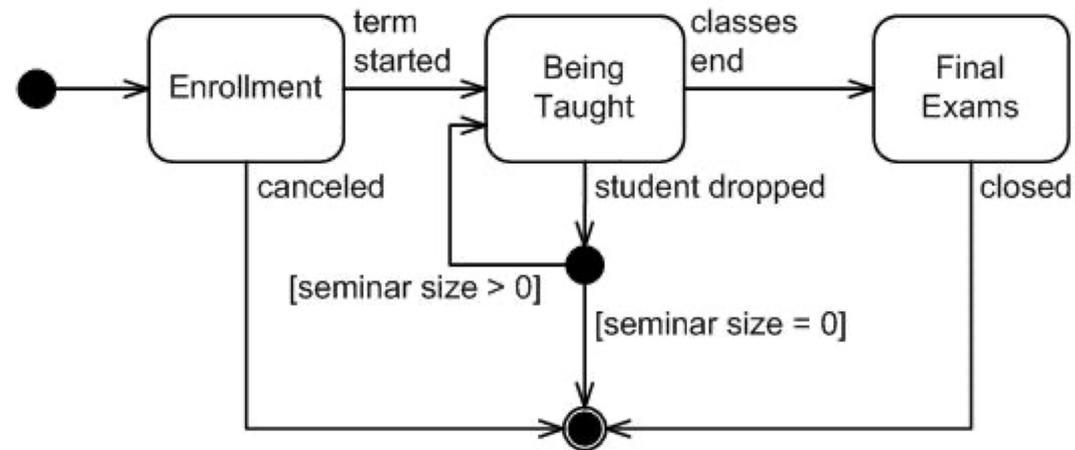
EVENTO OPERAZIONE

- Operazioni della classe telefono (dial digit(n))



ESEMPIO 2

■ Student Enrollment



(PSEUDO)STATI INIZIALI E FINALI

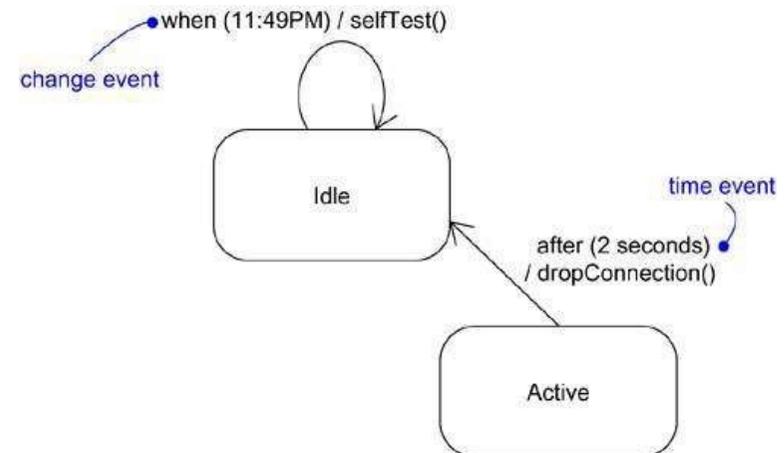
- Gli esempi precedenti mostrano due pseudo-stati, per avviare e bloccare la macchina a stati:
 - Il disco nero marca l'inizio dell'esecuzione. Non è uno stato vero e proprio ma un marcatore che punta allo stato da cui partire.
 - Il disco nero bordato (nodo finale), indica che l'esecuzione è terminata.
- Possono comparire in qualunque numero all'interno di un diagramma (o di uno stato composito, che vedremo in seguito).

EVENTO

- Un evento occorre istantaneamente
- Gli eventi che occorrono quando l'oggetto è in uno stato per cui non è prevista alcuna transizione etichettata con quell'evento vengono ignorati
- È ammesso il non-determinismo: un evento può fare da trigger a più transizioni che escono dallo stesso stato :
 - Ne viene scelta una non-deterministicamente
- Livello di astrazione/dettaglio: Introducete un evento se ha degli effetti

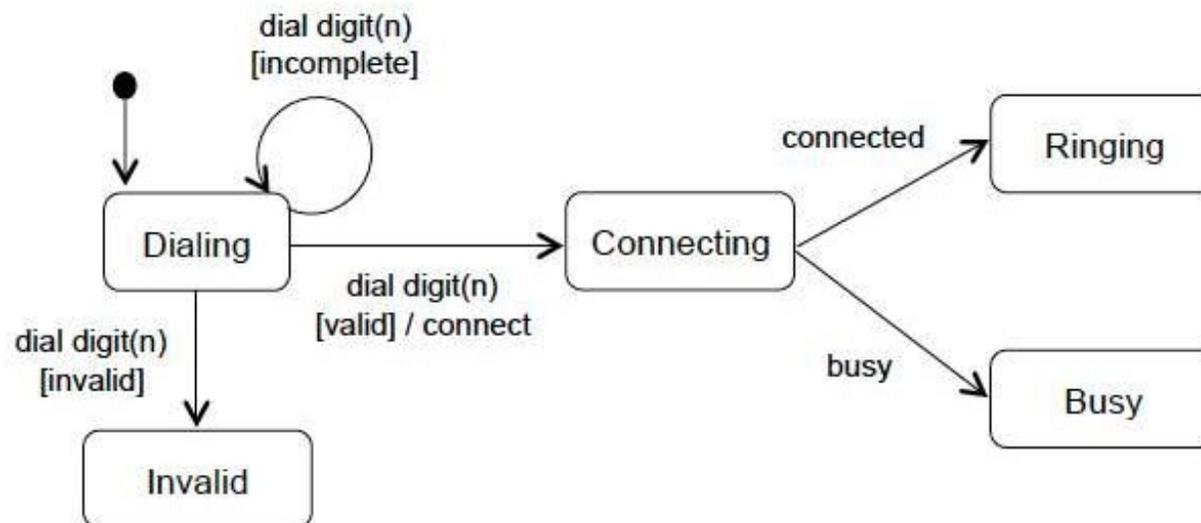
TIPI DI EVENTO

- 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 **when(exp)**
 - la transizione è abilitata appena l'espressione diventa vera
 - l'espressione può indicare un tempo assoluto o una condizione
- Evento temporale **after(time)**
 - la transizione è abilitata dopo che l'oggetto è stato fermo "time" in quello stato

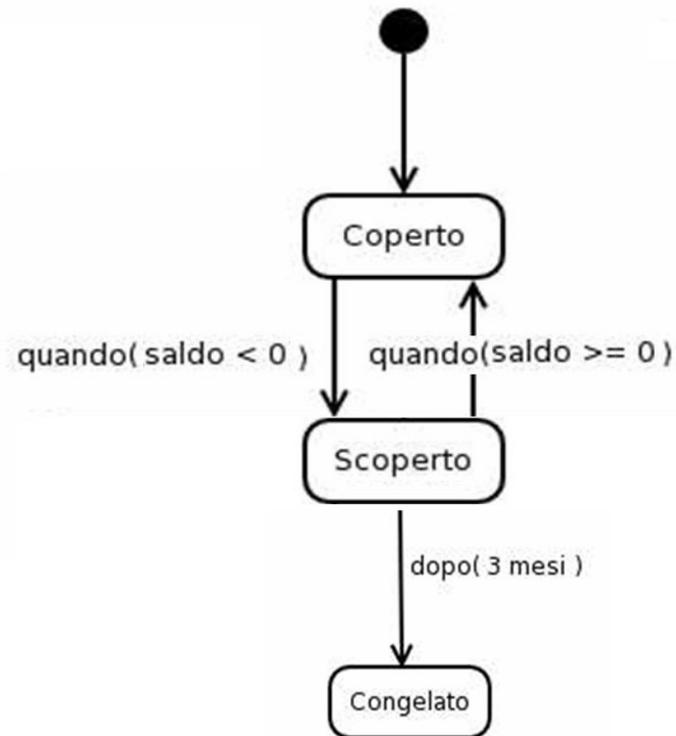


SEGNALI

segnali (busy e connected)



EVENTI DI VARIAZIONE ED EVENTI TEMPORALI



■ Eventi di variazione, a cosa servono?

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

■ Evento temporale

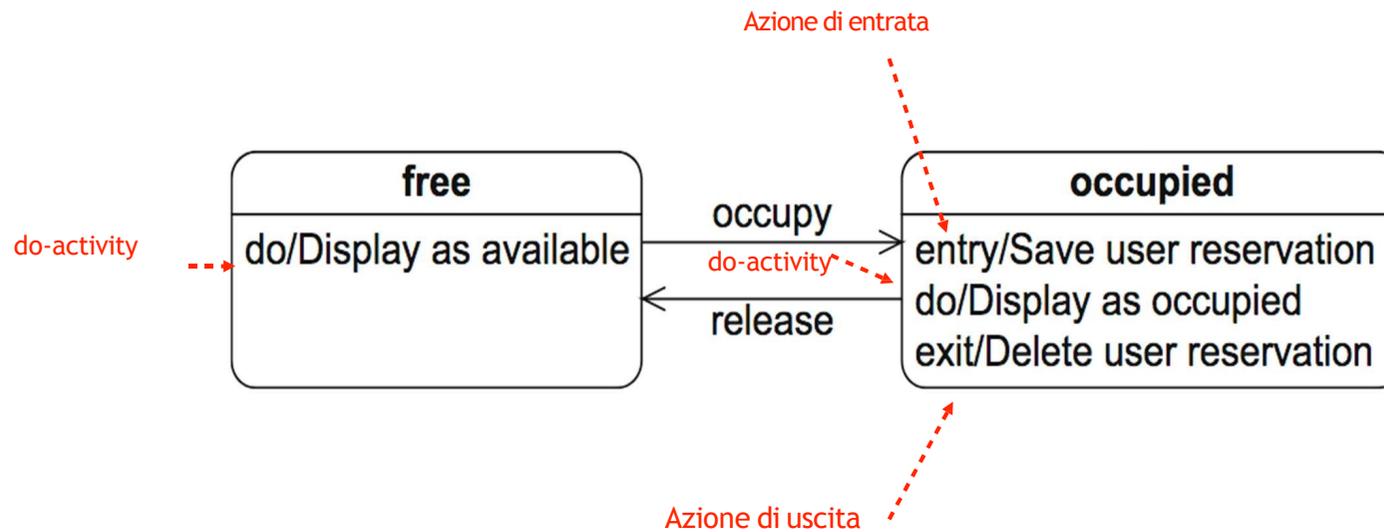
- Dopo che l'oggetto è stato 3 mesi fermo nello stato Scoperto, transisce nello stato Congelato

ENTRY/EXIT/DO

- All'interno di uno stato si possono usare alcune azioni speciali, indicate tramite keyword:
 - *entry*: eseguita quando l'oggetto entra nello stato
 - *exit*: eseguita quando l'oggetto esce dallo stato
 - *do* (*do-activity*): eseguita mentre l'oggetto è nello stato
- Una *self-transition* attiva sempre le *entry* ed *exit*, le *internal activities* invece no
- Una *do-activity* non è 'istantanea' ma può durare per un intervallo di tempo ed essere interrotta (da altri eventi)

ENTRY, EXIT, TRANSIZIONI E ATTIVITÀ INTERNE

- Ritorniamo sull'esempio dell'aula:

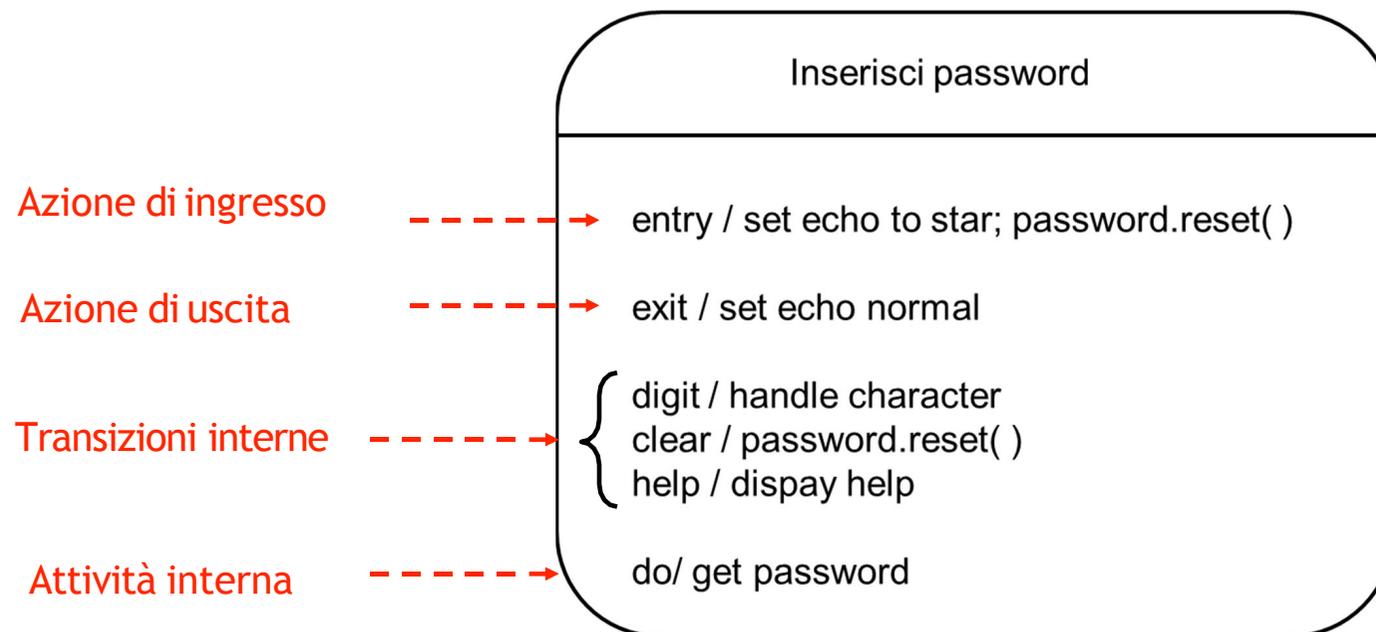


ENTRY, EXIT, TRANSIZIONI E ATTIVITÀ INTERNE

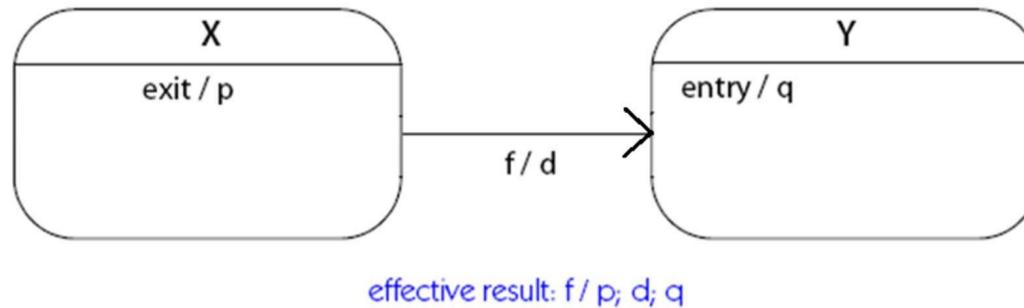
- **Azione di entrata:** eseguita all'ingresso in uno stato
- **Azione di uscita:** eseguita all'uscita di uno stato
- **Transizione interna:** risposta ad un evento

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

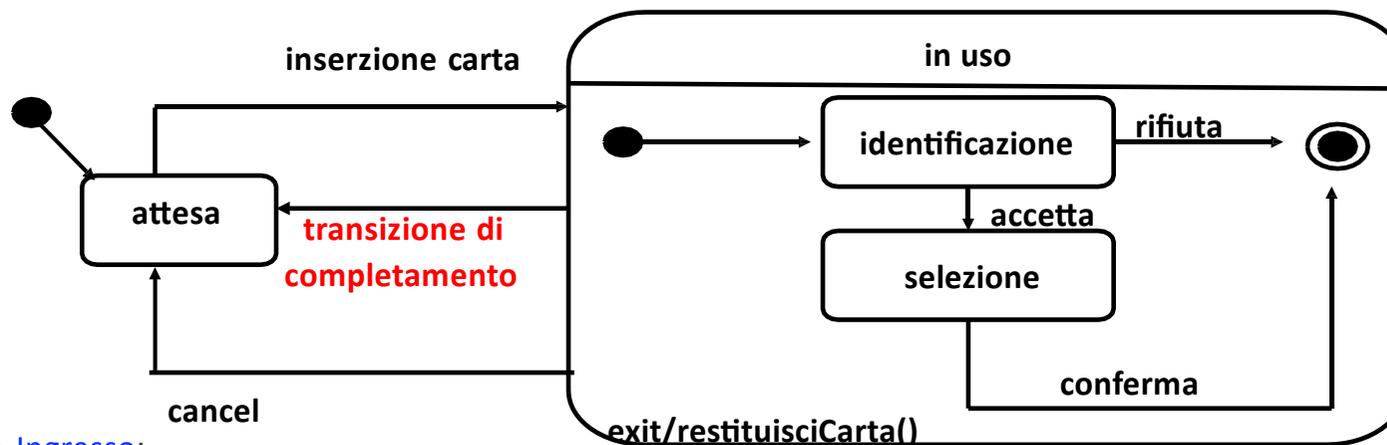
ENTRY, EXIT, TRANSIZIONI E ATTIVITÀ INTERNE



ENTRY, EXIT, TRANSIZIONI E ATTIVITÀ INTERNE: ESEMPI



(ESEMPIO DI) STATO COMPOSITO (SEQUENZIALE)



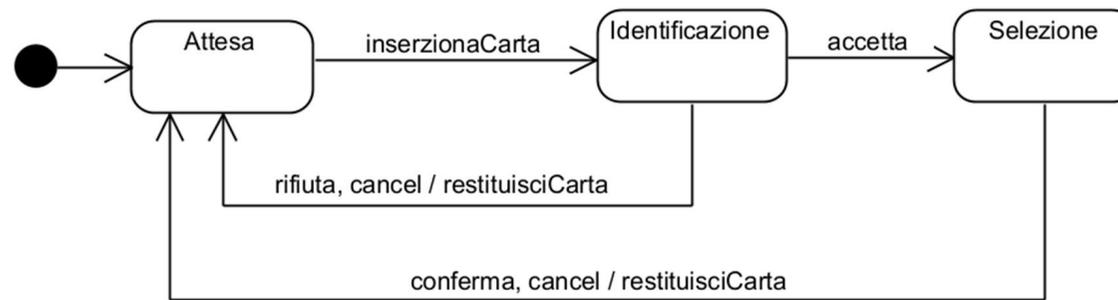
- **Ingresso:**

- Una transizione (inserzione carta) che arriva sul bordo prosegue nello stato iniziale dello stato composito
- Una transizione di ingresso può anche avere come target uno stato interno (non nell'esempio)

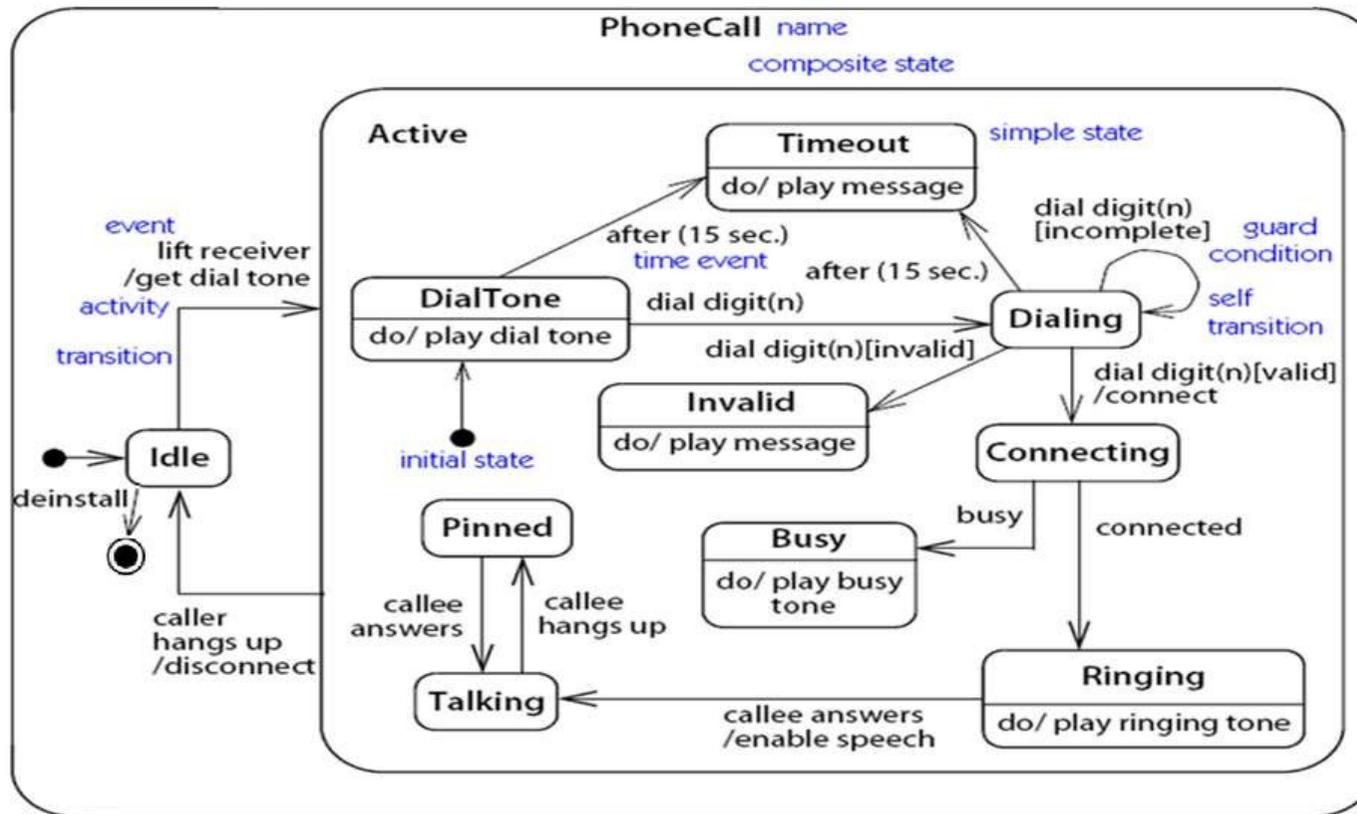
- **Uscita**

- Una transizione (cancel) che parte dal bordo si intende possibile **da un qualsiasi stato interno**
- Dallo stato finale (dopo le exit) si prosegue nella **transizione di completamento** (non ha evento)
- Possono esserci transizioni che "bucano" il bordo e si intendono possibili **dal solo stato interno** a cui sono collegate (non nell'esempio)

SENZA STATO COMPOSITO

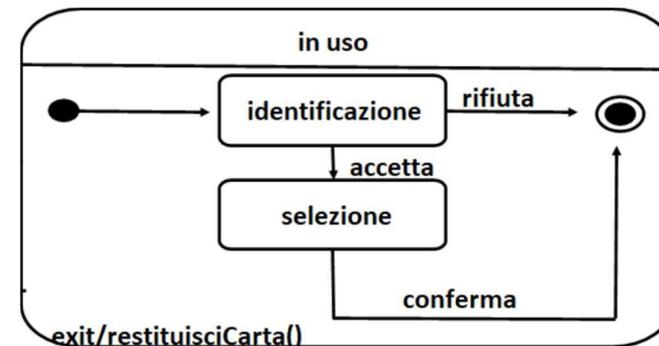


ESEMPIO: STATO COMPOSITO SEQUENZIALE (SENZA STATO FINALE)

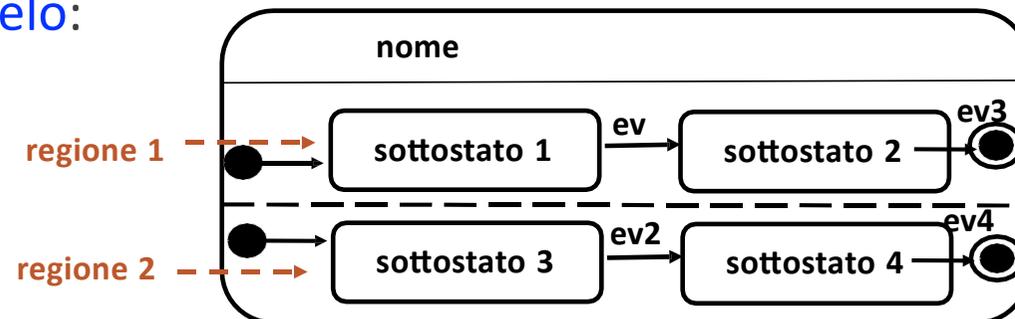


STATI COMPOSITI: SEQUENZIALI E PARALLELI

- Composito **sequenziale**:
- Un solo sottostato attivo in ogni istante

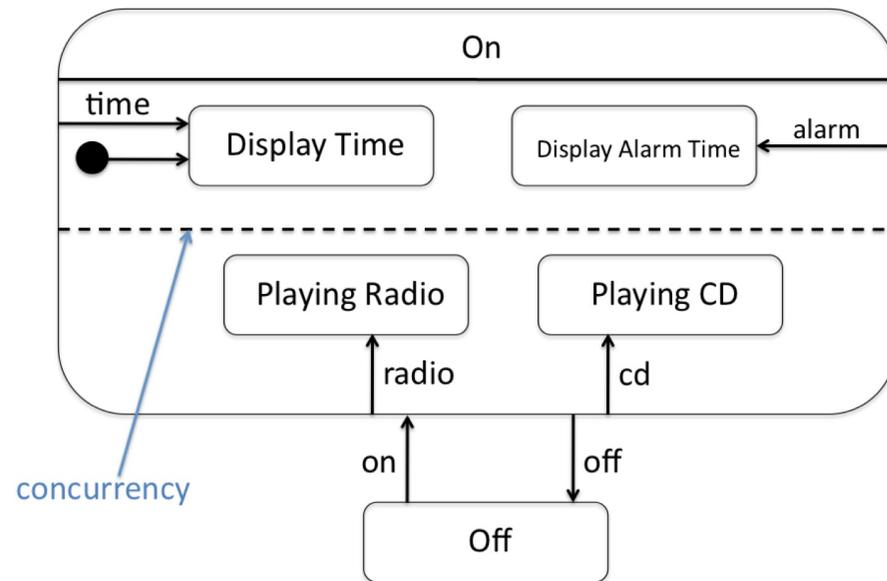


- Composito **parallelo**:
- sottostati attivi contemporaneamente
- uno per regione



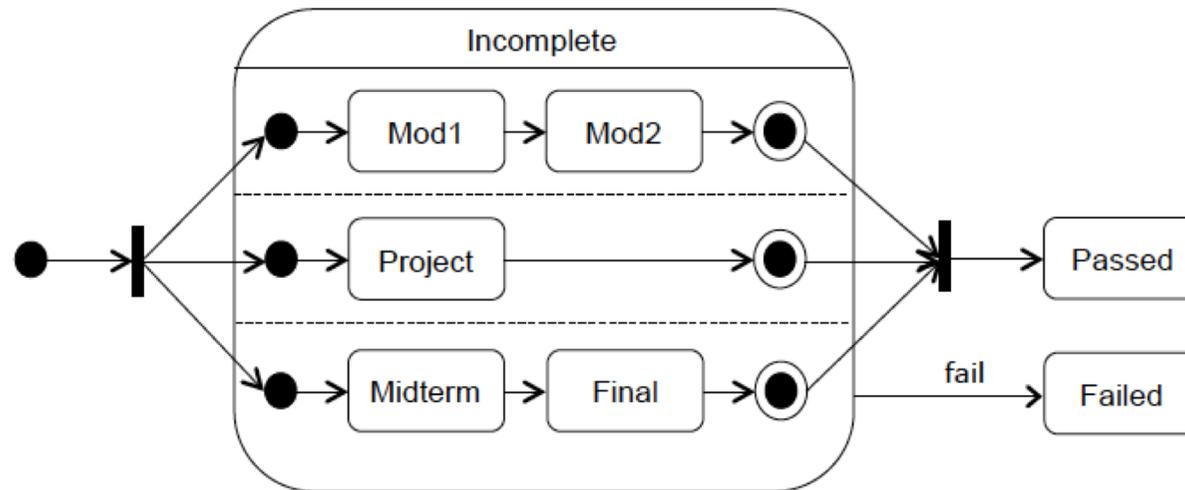
STATI COMPOSITI E CONCORRENZA

- Gli stati composti sono utili per modellare la concorrenza. Si divide lo stato composto in (sotto-) diagrammi ortogonali eseguiti in mutua esclusione
- Il diagramma sarebbe molto meno chiaro senza questo approccio (bisogna considerare tutte le possibilità)
- Esempio: una radiosveglia che o mostra l'orario o fa ascoltare musica

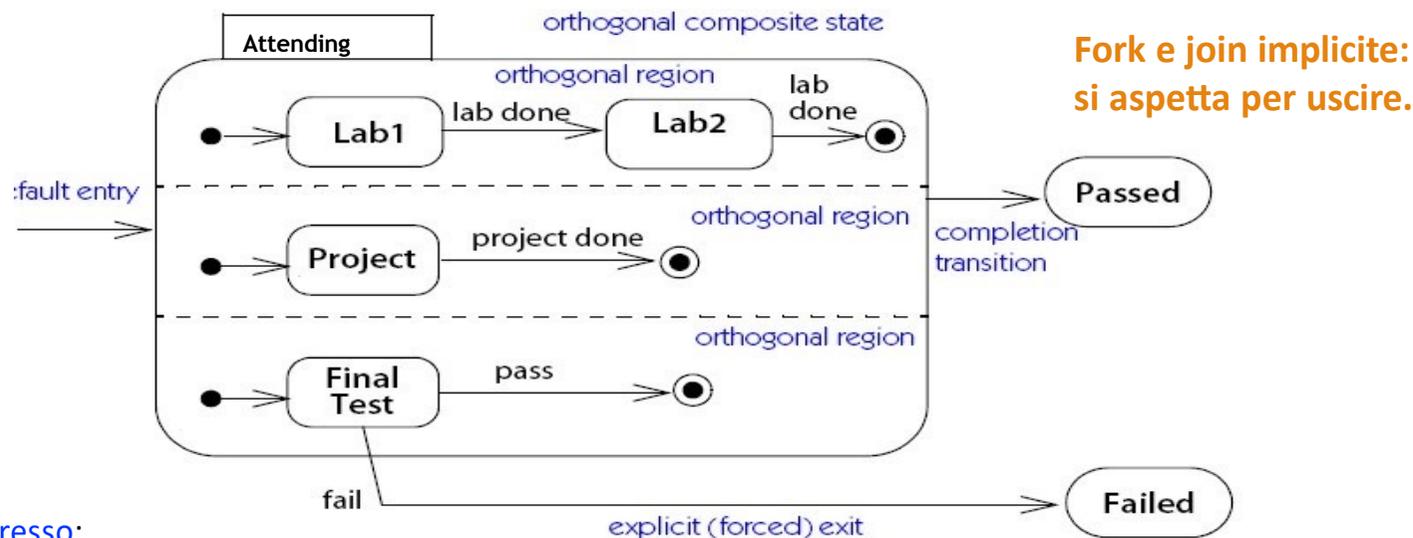


STATI COMPOSITI E SINCRONIZZAZIONE

- Gli stati composti sono inoltre utili per modellare la sincronizzazione. Si divide lo stato composto in (sotto-)diagrammi e si usano gli operatori di fork e join (che vedremo tra poco)
- Esempio: le attività e prove che uno studente deve superare per concludere un corso



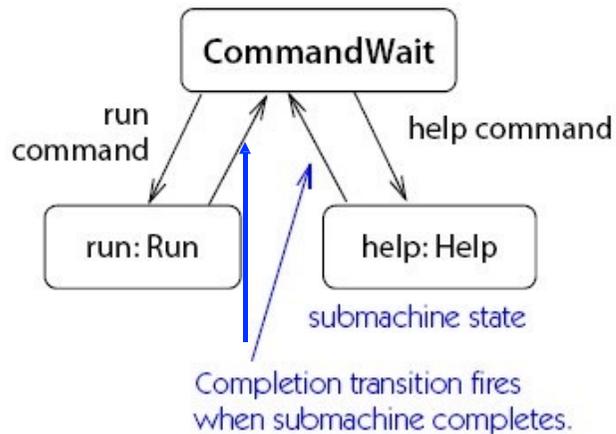
(ESEMPIO DI) STATO COMPOSITO PARALLELO



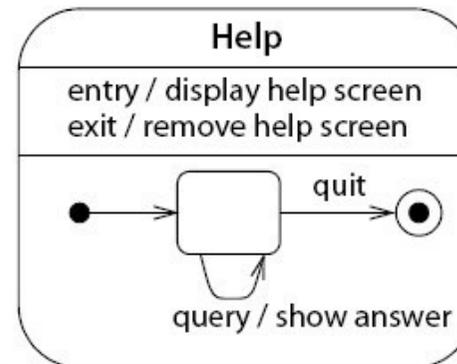
- Ingresso:
 - Una transizione (default entry) che arriva sul bordo prosegue **in tutti gli stati iniziali**
- Uscita
 - Una transizione che "buca" il bordo (fail) si intende possibile **solo** se l'evento avviene quando la macchina si trova in **nello stato interno** a cui è collegata: **fa uscire da tutti i sottostati**
 - Una volta raggiunti **tutti gli stati finali** si prosegue nella transizione di completamento
 - Una transizione che parte dal bordo si intende possibile **da un qualsiasi stato interno**, **fa uscire da tutti i sottostati** (non nell'esempio)

SOTTOMACCHINE

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



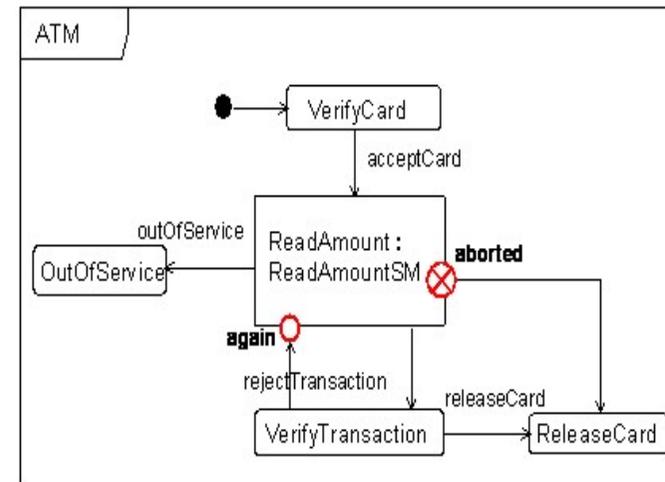
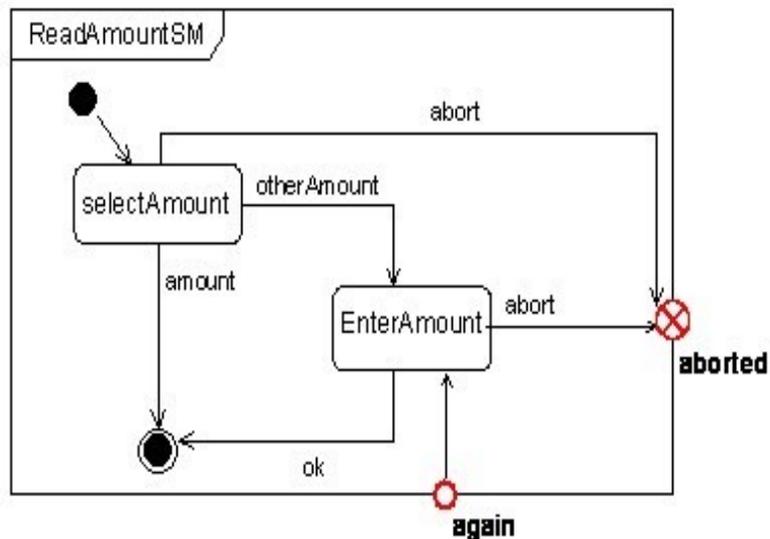
submachine definition



This submachine can be used many times.

SOTTOMACCHINE: ENTRY AND EXIT POINTS

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



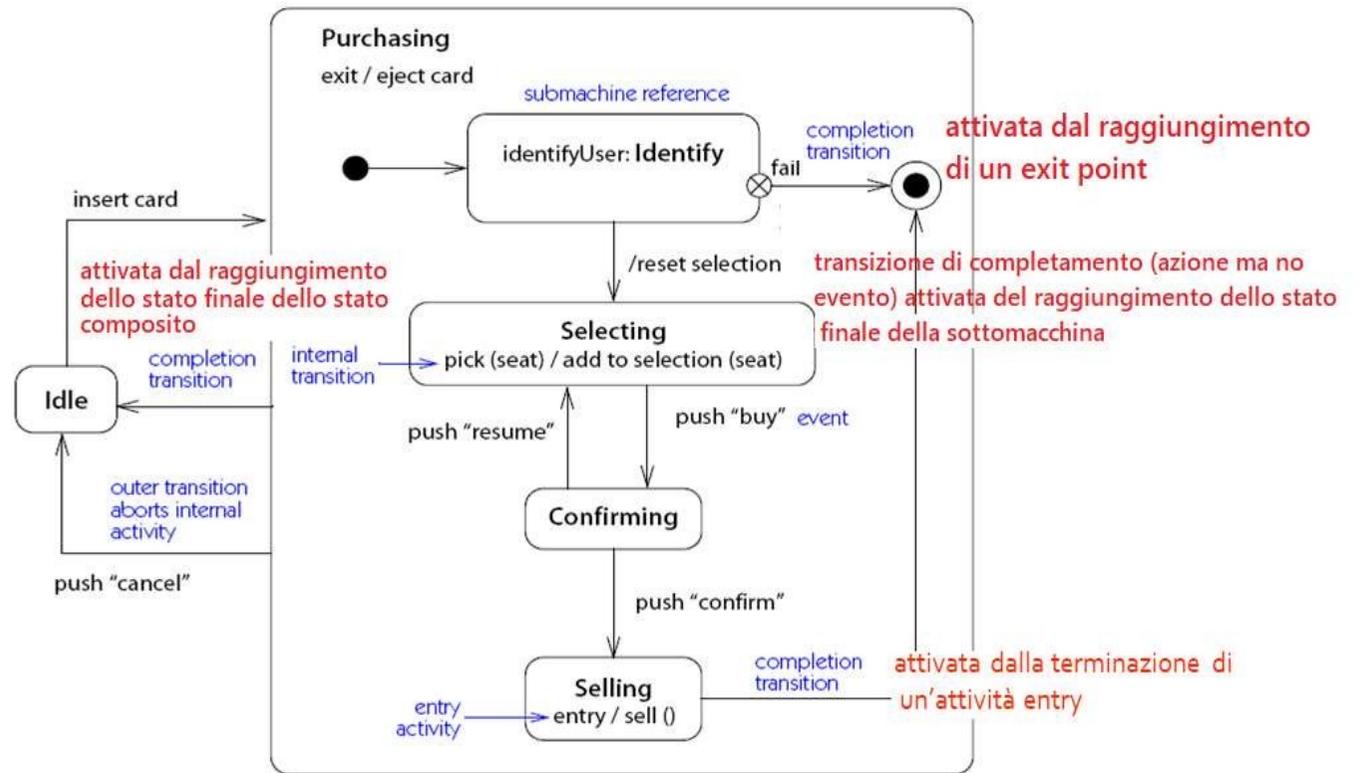
TRANSIZIONI DI COMPLETAMENTO

- Si tratta di transizioni che non hanno un evento associato (ma possono avere una guardia).
- Nel caso di uno stato semplice, una transizione di completamento è eseguita al termine dell'attività di quello stato (fine azioni *do*).
- Nel caso di uno stato composito o *submachine state* una transizione di completamento è eseguita quando si giunge in uno stato finale oppure un *exit point*.

TRANSIZIONI DI COMPLETAMENTO: TUTTI I CASI

- Senza evento, scattano al raggiungimento:
 1. Della terminazione di un'attività composta, i.e. al raggiungimento
 - Dello stato finale in un stato composto sequenziale
 - Degli stati finali di tutte le regioni ortogonali di un stato composto parallelo
 - Di un exit point
 2. Alla terminazione di entry e/o di do activity (la exit activity viene eseguita quando scatta la transizione di completamento)
 3. Di uno pseudo-stato giunzione (lo vedremo in un attimo)

TRANSIZIONI DI COMPLETAMENTO: ESEMPIO



ALTRI TIPI DI STATO (PSEUDOSTATI)

Giunzione



Storia

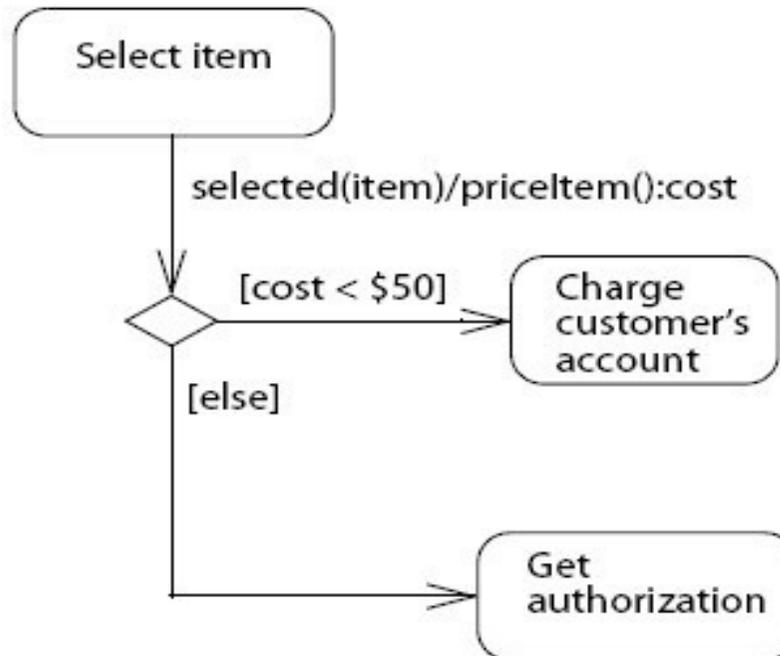


Decisione

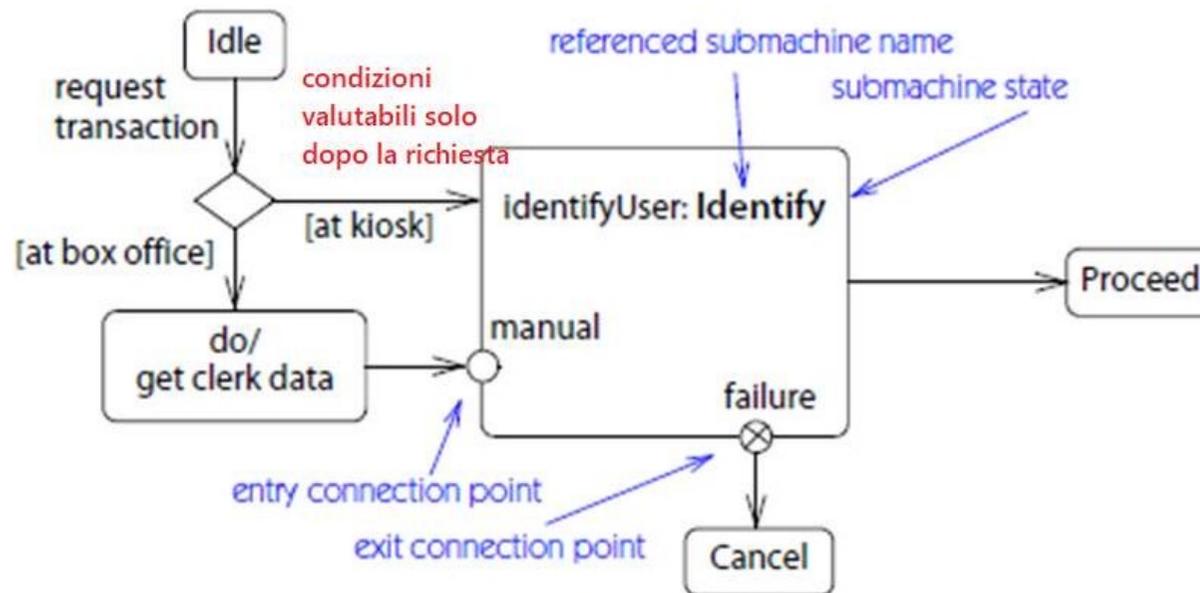


ESEMPIO DI CHOICE

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

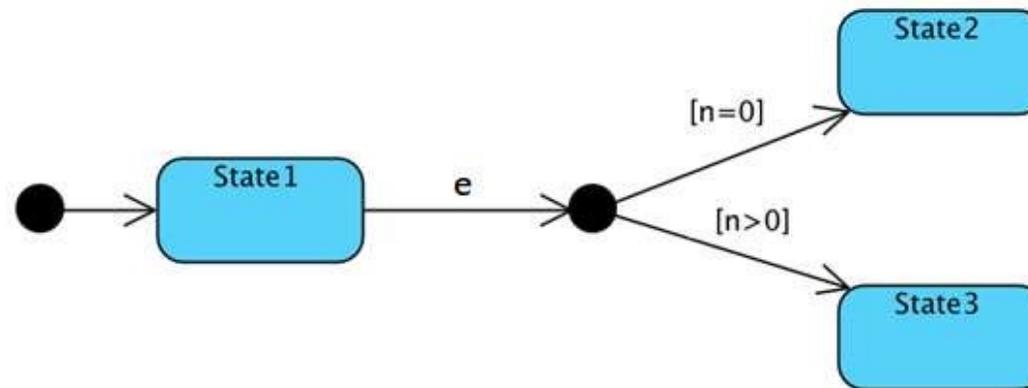


ESEMPIO DI CHOICE (CON INVOCAZIONE DI SOTTO-MACCHINA)



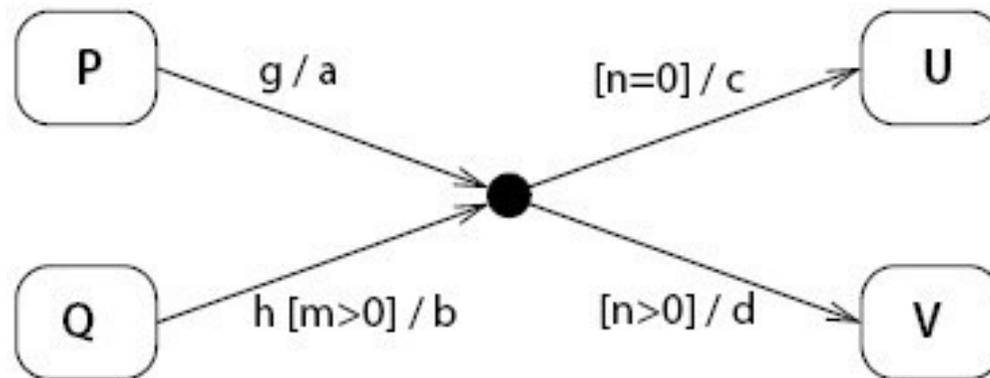
GIUNZIONE

- 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 $n < 0$ l'evento e viene ignorato e si rimane nello stato 1

GIUNZIONE, UN ALTRO ESEMPIO



Equivalent transitions:

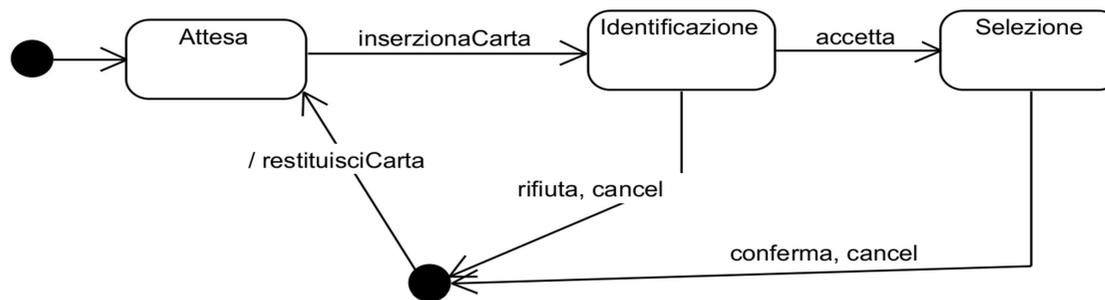
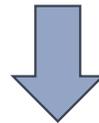
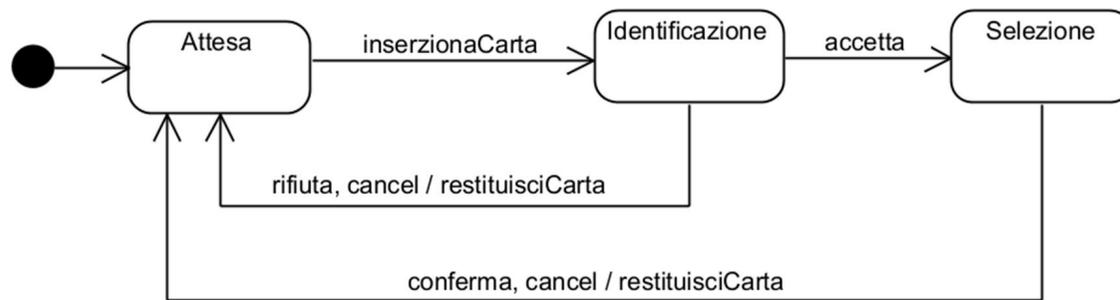
$P g [n=0] / a; c U$

$P g [n>0] / a; d V$

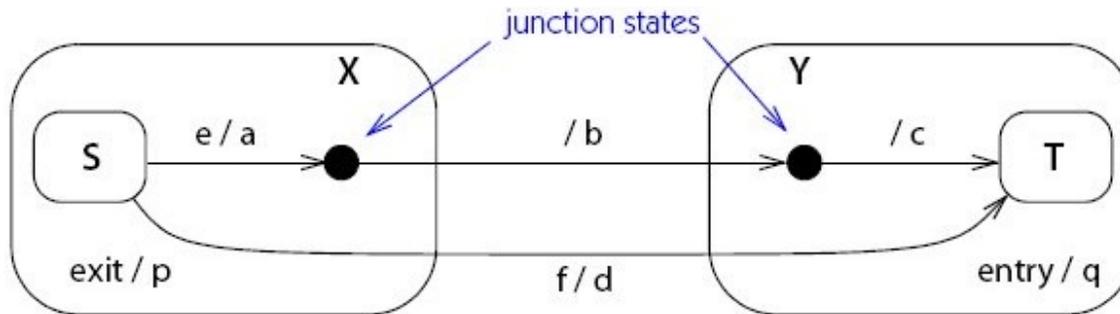
$Q h [m>0 \text{ and } n=0] / b; c U$

$Q h [m>0 \text{ and } n>0] / b; d V$

L'ESEMPIO VISTO PRIMA CON PSEUDOSTATO GIUNZIONE



GIUNZIONE, CON ENTRY E EXIT ACTIONS



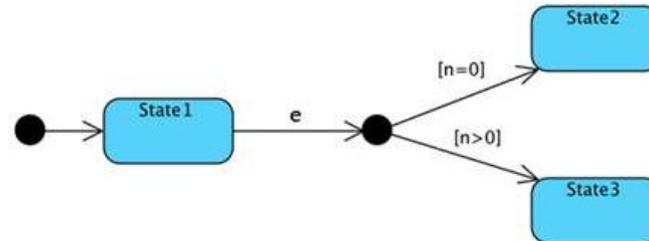
effective result

$e/a; p; b; q; c$

$f/p; d; q$

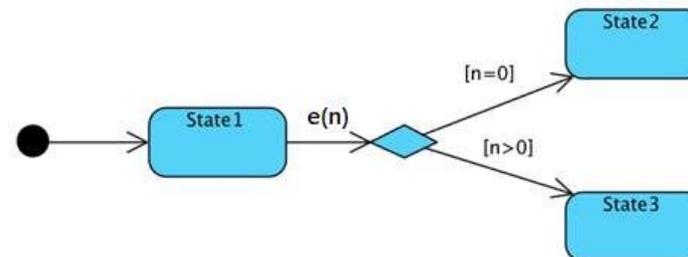
GIUNZIONE VS CHOICE

■ Giunzione (statica)



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

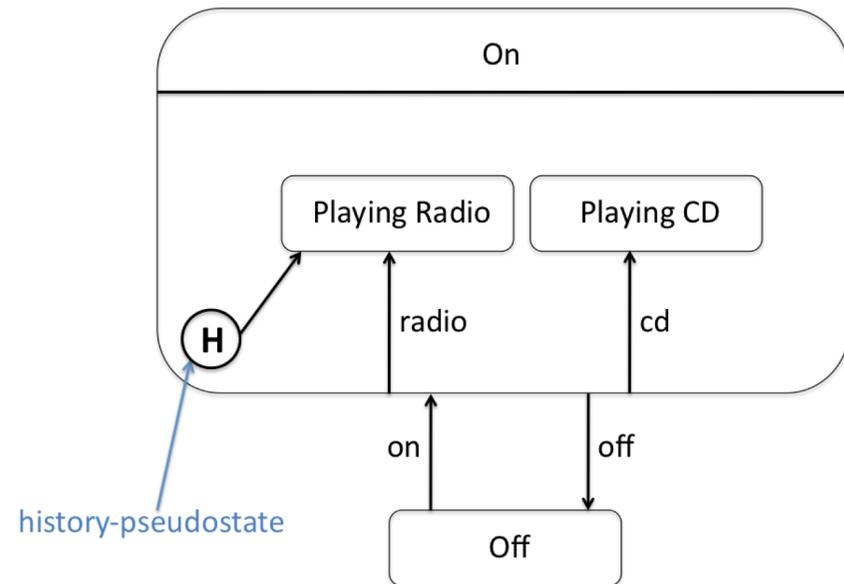
■ Choice (dinamica)

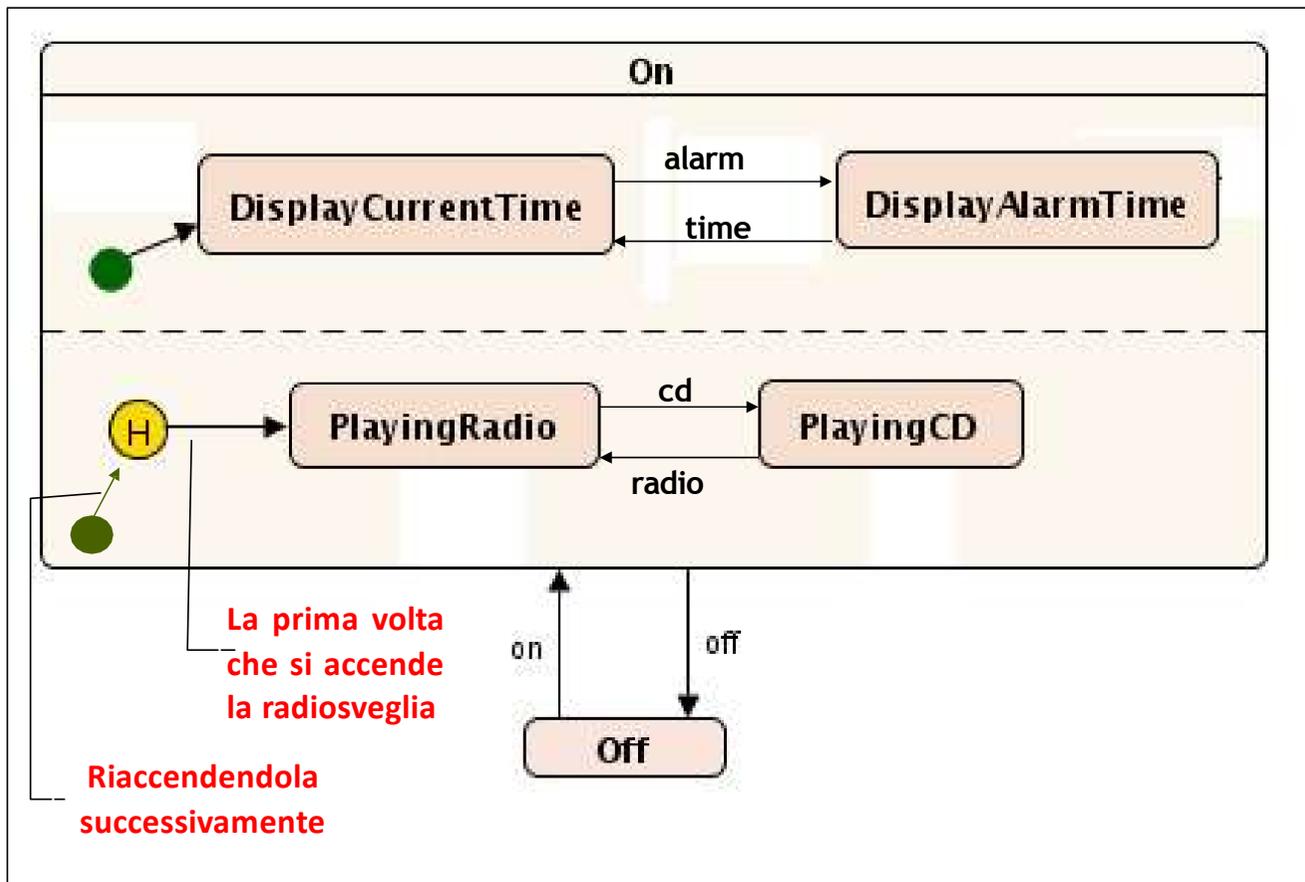


Le guardie sono valutate dopo $e(n)$. In questo esempio occorre avere garanzia che n sia maggiore o uguale a zero

HISTORY PSEUDOSTATE

- Un history pseudostate indica il più recente stato interno attivo di uno stato composito
- Utile per ripristinare stati precedenti
- Esempio: alla ri-accensione di una radio/CD si seleziona radio o CD in base alla scelta attiva allo spegnimento





TIPS AND TRICKS



MACCHINA A STATI O ATTIVITÀ?

- Come scegliere il diagramma più appropriato per descrivere il modello dinamico
- Ricordiamo:
 - Il diagramma di macchina a stati parla dell'evoluzione nel tempo delle istanze di un classificatore
 - il diagramma di attività parla di un'agenda di azioni da fare.

Se il focus è

- mettere in ordine un insieme di azioni da fare attività
- mostrare l'evoluzione di un oggetto in risposta a eventi stati

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 e spesso nella pratica si disattende (eccezioni anche negli esempi visti), ma seguire la questa prassi è aiuta a costruire i diagrammi correttamente
- Inoltre: Errore comune nei compiti confondere stati e azioni

APPENDICE

Da UML Reference Manual

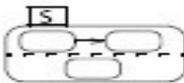
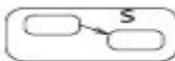
Table 7-1: *Kinds of Events*

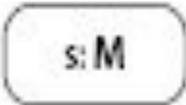
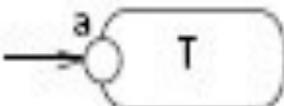
<i>Event Type</i>	<i>Description</i>	<i>Syntax</i>
call event	Receipt of an explicit synchronous call request by an object	op (a:T)
change event	A change in value of a Boolean expression	when (exp)
signal event	Receipt of an explicit, named, asynchronous communication among objects	sname (a:T)
time event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

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

<i>Transition Kind</i>	<i>Description</i>	<i>Syntax</i>
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

Table 7-3: Kinds of States

<i>State Kind</i>	<i>Description</i>	<i>Notation</i>
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	

<i>State Kind</i>	<i>Description</i>	<i>Notation</i>
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	

ESERCIZI CONSIGLIATI

- Completare Esercizio Automobile visto in aula
- Dare un diagramma di macchina a stati che modelli
 - L'evoluzione nel tempo della classe Utente (o Situazione Utente) di Myair
 - L'evoluzione nel tempo del semaforo (ex. Semafori)