

UML: Diagramma delle attività'

Roberta Gori, Laura Semini
Ingegneria del Software
Dipartimento di Informatica
Università di Pisa

Riassunto lezione precedente Outline della lezione

- Lezioni precedente:
 - Descrizione del dominio: modello statico

- Questa lezione
 - Descrizione del dominio: modello dinamico
 - diagrammi di attività (business model)
 - diagrammi di macchina a stati

Diagrammi di attività

- Modellano il flusso di lavoro (workflow, business model)
 - di un compito o algoritmo o
 - di un processo/attività
- Un'attività descrive la coordinazione di un insieme di azioni. Centrata su:
 - sequenza e concorrenza delle azioni
 - e sulle condizioni che le abilitano
 - piuttosto che sui classificatori che eseguono queste azioni
- Antenati: flow charts e Reti di Petri

Diagrammi di attività

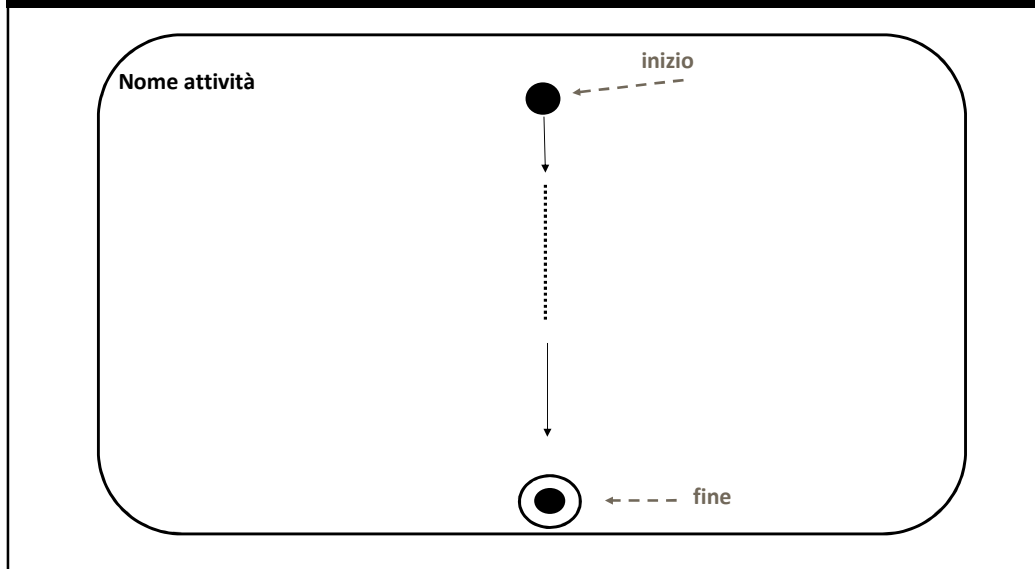
Modellano un'attività relativa a una qualsiasi entità o collezione di entità, ad esempio:

- una o più classi che collaborano in una attività comune
- uno o più attori con il sistema
- un'operazione di classe

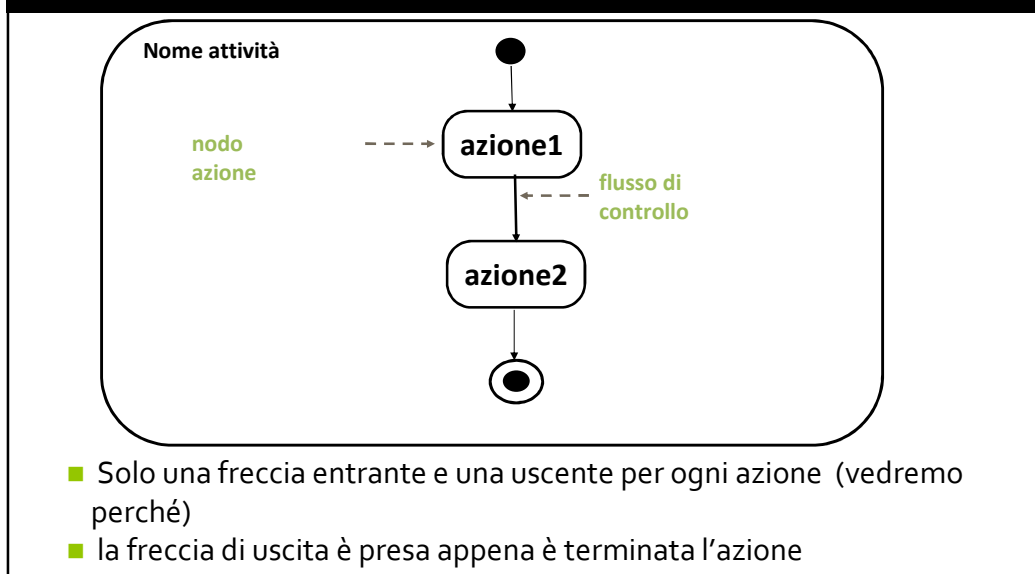
Alcuni usi dei diagrammi di attività:

- modellare un processo aziendale (analisi)
- modellare il flusso di un caso d'uso (analisi)
- modellare il funzionamento di un'operazione di classe (progettazione)
- modellare un algoritmo (progettazione o testing)

Diagrammi di attività: inizio e fine

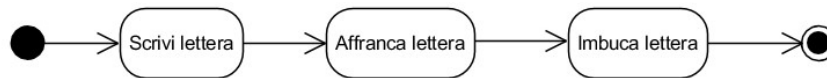


Diagrammi di attività: nodo azione



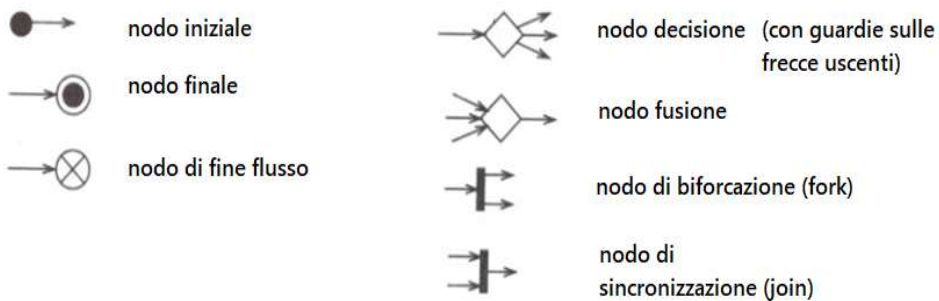
Transizioni

- Quando un'azione ha terminato il proprio lavoro scatta una **transizione automatica** in uscita dall'azione che porta all'azione successiva

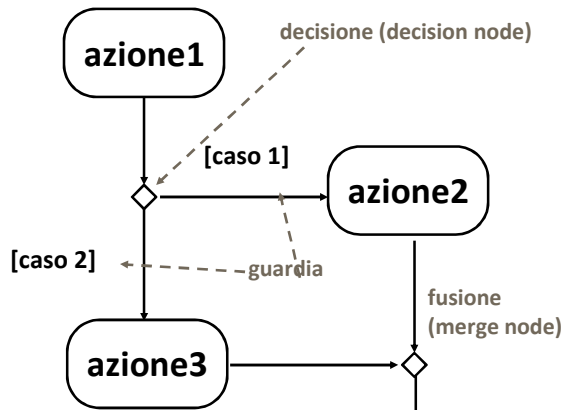


- La semantica è descritta con il token game

Elementi principali

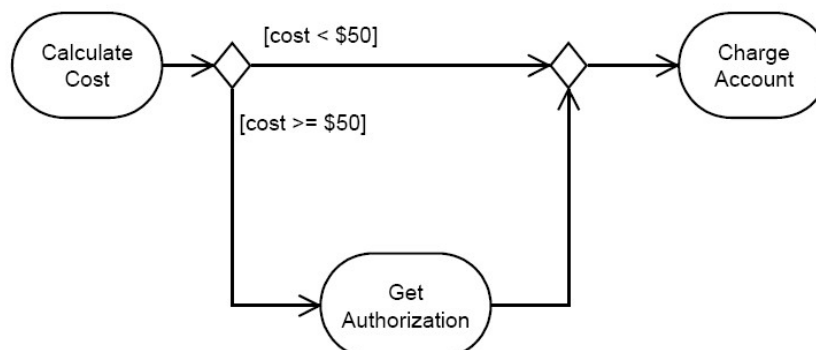


Diagrammi di attività: scelta



- caso1 e caso2 devono coprire tutti i casi: caso 1 OR caso2=True
- posso usare "altrimenti"

Esempio: Decisione e fusione / decision and merge

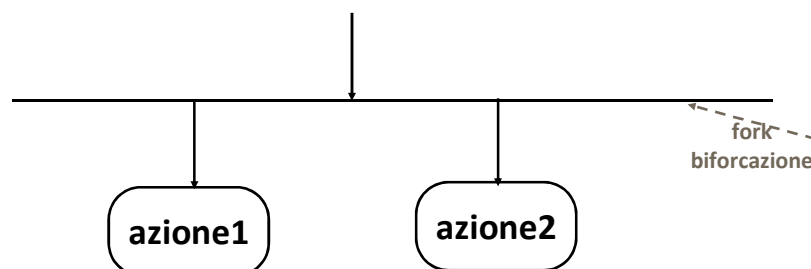


- Il token prende uno dei cammini
- Deve prendere sempre uno dei cammini

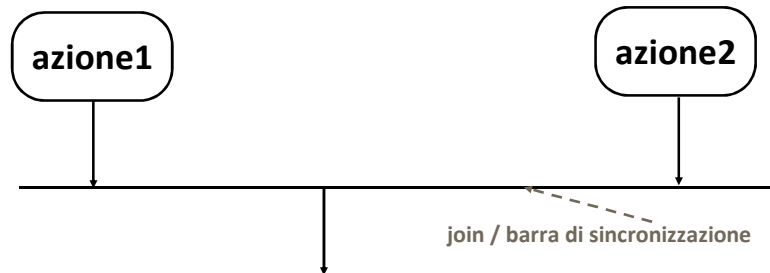
Semanica: Decisione e fusione / decision and merge

- Le guardie devono coprire tutte le possibilità
 - In caso si usa [else]
- E' bene (ma non necessario) che siano mutualmente esclusive altrimenti comportamento non definito (non deterministico).
- Le condizioni di guardia sempre tra []
 - (in generale in UML)
- Dato un nodo decisione non è obbligatorio un nodo fusione corrispondente.
 - Potrebbe per esempio esserci un nodo di fine flusso

Diagrammi di attività: fork e join



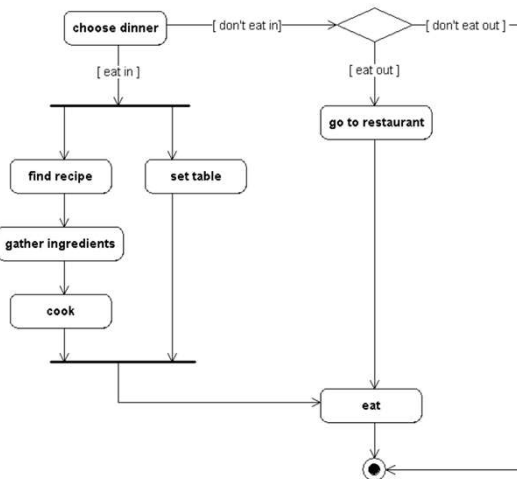
Diagrammi di attività: fork e join



Biforcazione e ricongiunzione / fork and join

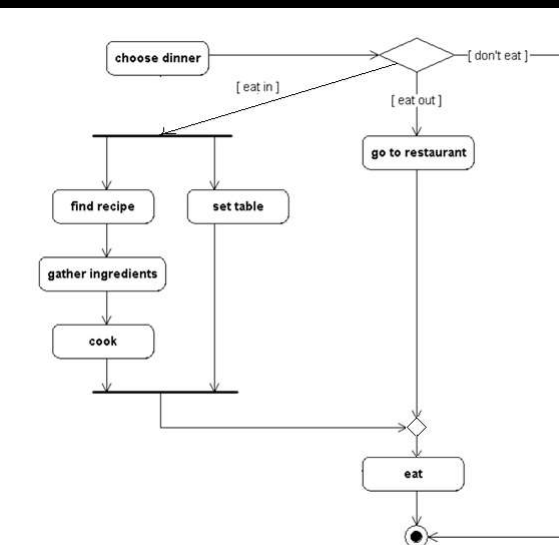
- Token game:
 - La fork moltiplica i token:
 - Dato un token in ingresso, ne "produce" uno per ogni freccia uscente
 - La join li consuma:
 - Si attende un token per ogni freccia entrante
 - Si consumano tutti e ne esce solo uno
- Non è necessaria una join per ogni fork

Esempio preso da web, interessante perché sbagliato



Anche se UML permette frecce multiple entranti/uscenti in/da un nodo, se ne sconsiglia ([vieta in questo corso](#)) assolutamente l'uso: la semantica UML in questo caso è quella della fork/join, ma poi è facile sbagliarsi e disegnare diagrammi come questo che vanno in deadlock. Infatti eat attende due token che non possono mai arrivare.

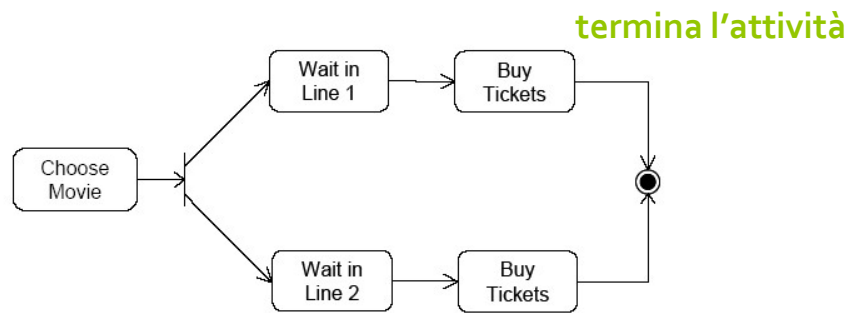
Diagramma corretto



Prima di eat serve un modo fusione e dopo choose dinner un nodo decisione. Sono tollerate due frecce entranti nello stato finale, con la semantica della fusione.

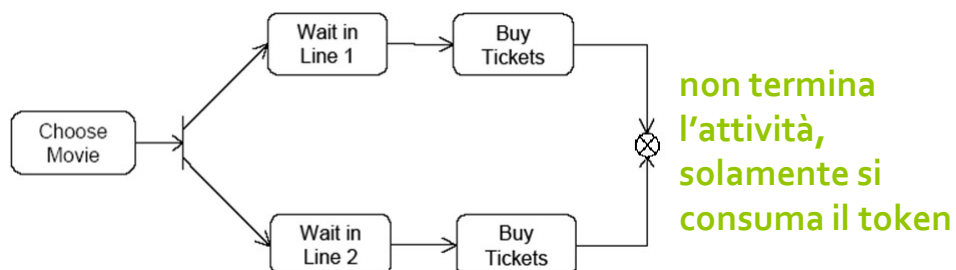
Nodo di fine attività'

- il primo che compra i biglietti termina l'attività

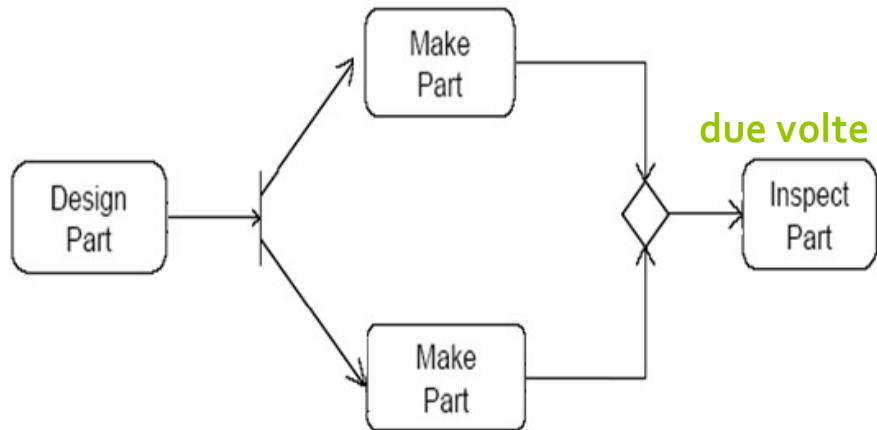


Nodo di fine flusso

- il primo che compra i biglietti **non termina l'attività**
- Vengono presi i biglietti in entrambe le code



Fork e merge: possibile ma azioni eseguite due volte



Segnali ed Eventi

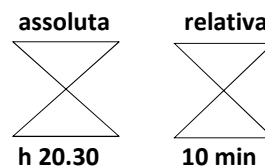
- Accettazione di evento esterno



- Invio di un segnale



- Accettazione di evento temporale

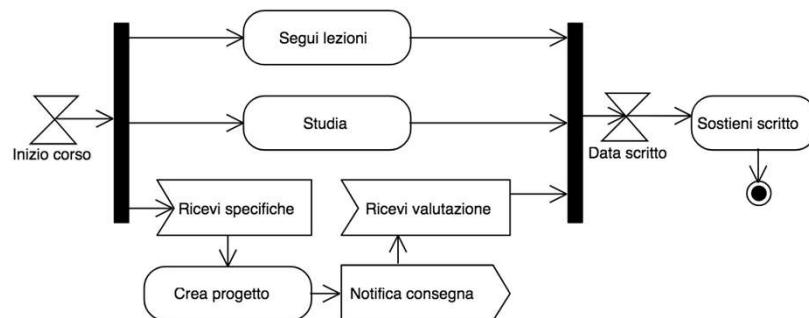


Nodi specializzati che gestiscono l'invio e la ricezione di segnali.

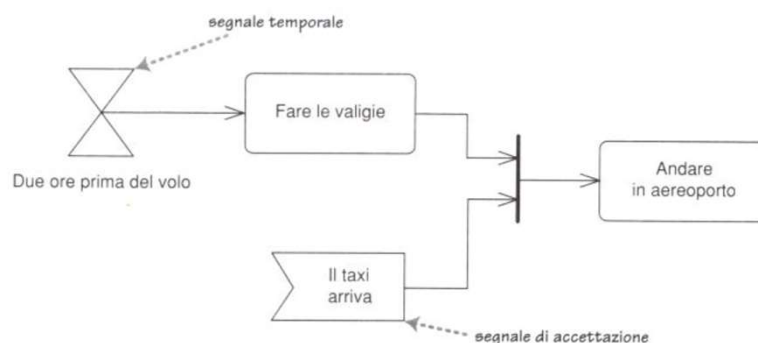
L'invio di segnali è **asincrono** e **non blocca l'attività**.

Accettazione evento esterno o accettazione di evento temporale

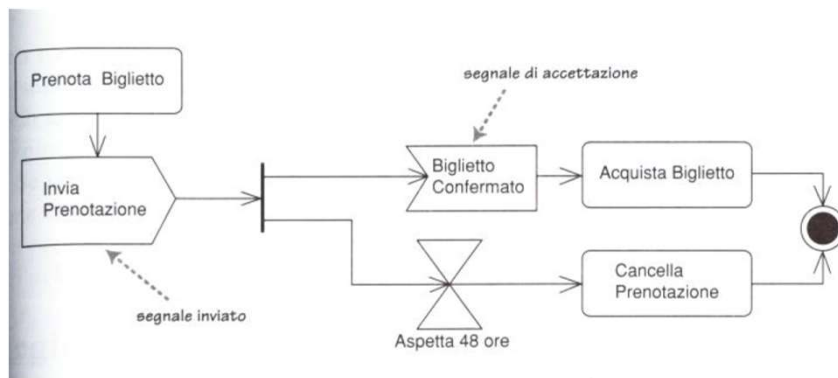
- Per accettazione evento esterno (analogo per accettazione eventi temporali) : **arco entrante non necessario**
- Se assente, quando arriva l'evento, si genera un token
- Se presente, l'azione è abilitata quando arriva il token e si attende l'evento esterno per farlo transitare



Esempio



Esempio



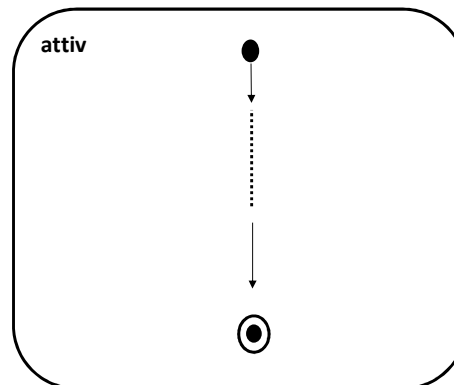
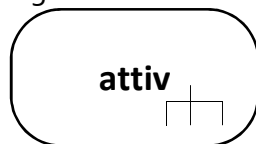
Accettazione di eventi esterni e invio segnali vs azioni

- Quando usare una azione e quando usare accettazione di eventi esterni o invio segnali:
 - Si usa una azione quando è effettuata dal classificatore/insieme di classificatori di cui si sta descrivendo il comportamento
 - I secondi si usano quando si comunica con una entità esterna

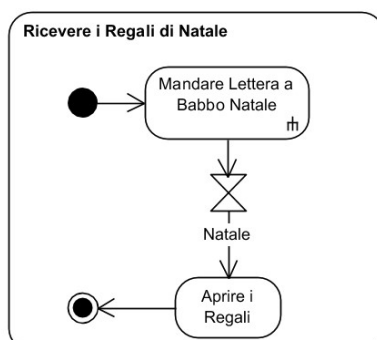
SottoAttività

- Un'azione può includere (chiamare) un'altra attività (secondaria) :

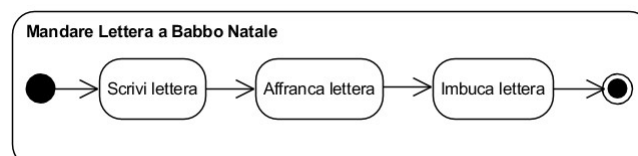
- si usa il "rastrello" (rake) per dire che l'azione include una sotto-attività
- Si descrive la sotto-attività in un diagramma a parte
- Migliora il riuso e la leggibilità



SottoAttività



- Mandare Lettera a Babbo Natale è lasciata astratta in un diagramma (Ricevere i Regali di Natale), rappresentata come un solo nodo, il rastrello dice che è descritta in un altro diagramma
- Visione bottom-up: si definisce una attività (Mandare Lettera a Babbo Natale) e poi la si riferisce in un altro diagramma (Ricevere i Regali di Natale) usando il rastrello



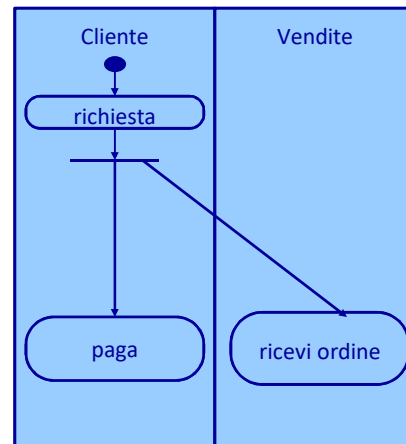
Partizioni

■ Una *partizione*

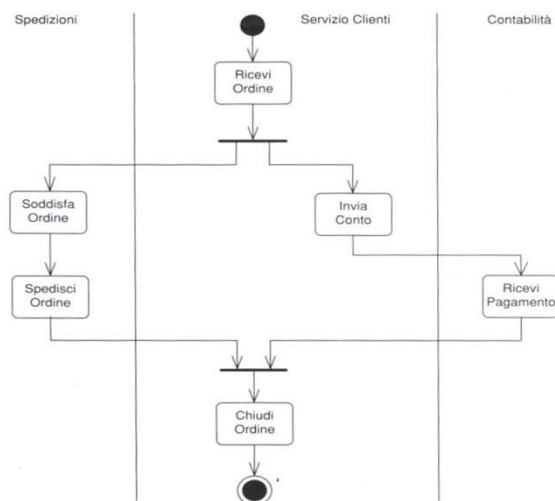
- per dividere le azioni in gruppi
- Spesso corrisponde alla divisione in unità operative in un modello di business.

■ Permettono di

- assegnare la responsabilità delle azioni



Esempio



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 può trovare un oggetto.
- Inoltre dobbiamo descrivere come da ciascuno di questi stati l'oggetto può 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)

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

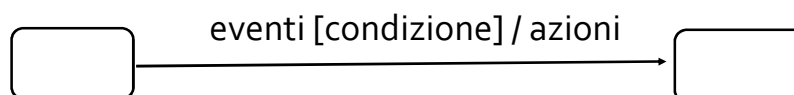
Stato: 

Stato iniziale: ●

Stato finale: ●

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



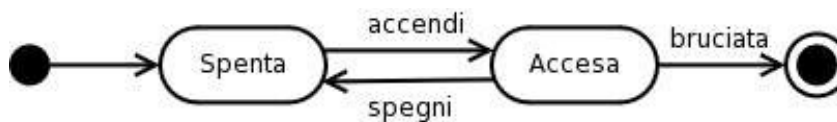
eventi ::= evento | evento , eventi (disgiunzione)
 azioni ::= azione | azione , azioni (sequenza)

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

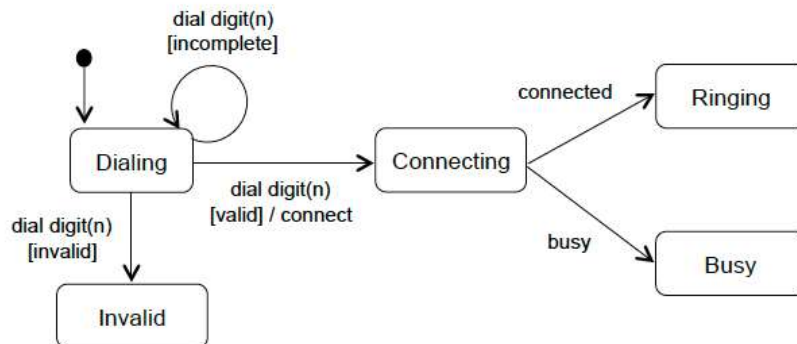
- Gli eventi che arrivano in uno stato per cui non è prevista alcuna transizione vengono ignorati
- È ammesso il non-determinismo: un evento può fare da trigger a più transizioni:
 - Se le due transizioni escono dallo stesso stato, ne viene scelta una non-deterministicamente

Tipi di evento

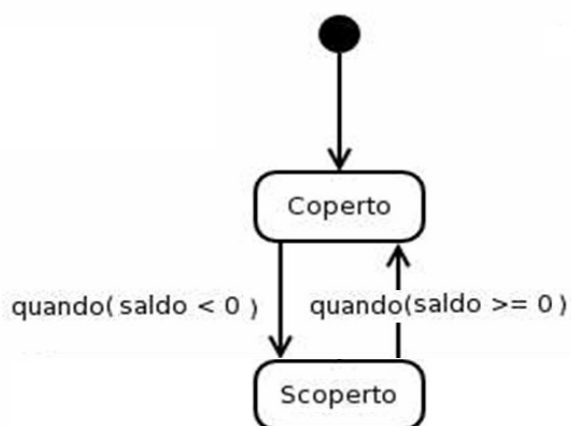
- Operazione o segnale `op(a:T)`
 - la transizione è abilitata quando l'oggetto (in quello stato) riceve una chiamata di metodo / un segnale con parametri (a) e tipo (T) (i parametri sono opzionali)
- Evento di variazione `quando(exp)`
 - la transizione è abilitata appena l'espressione diventa vera
 - 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)`

Esempio

- 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)

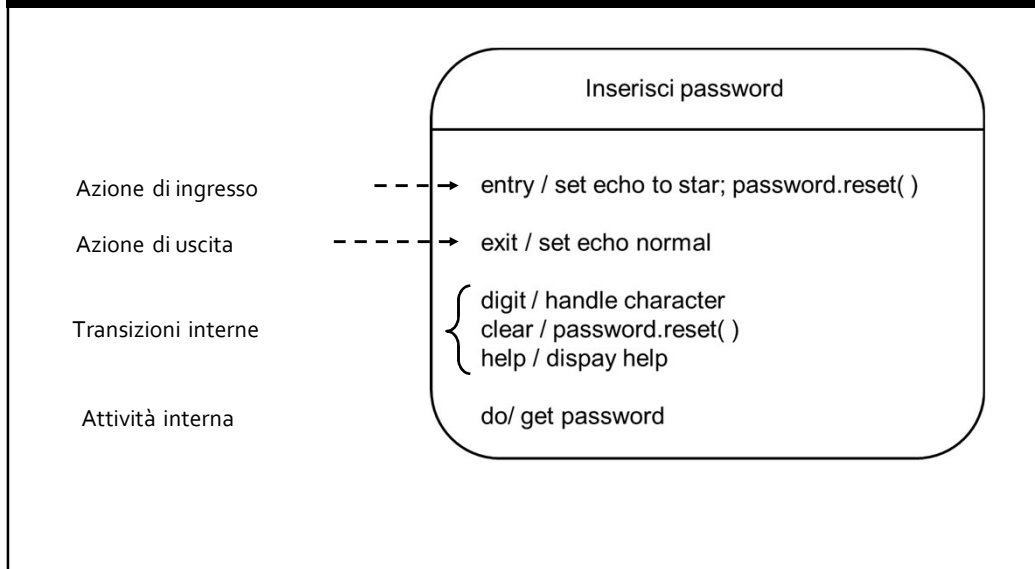


Dopo che l'oggetto è stato 3 mesi nello stato Scoperto, transisce nello stato Congelato

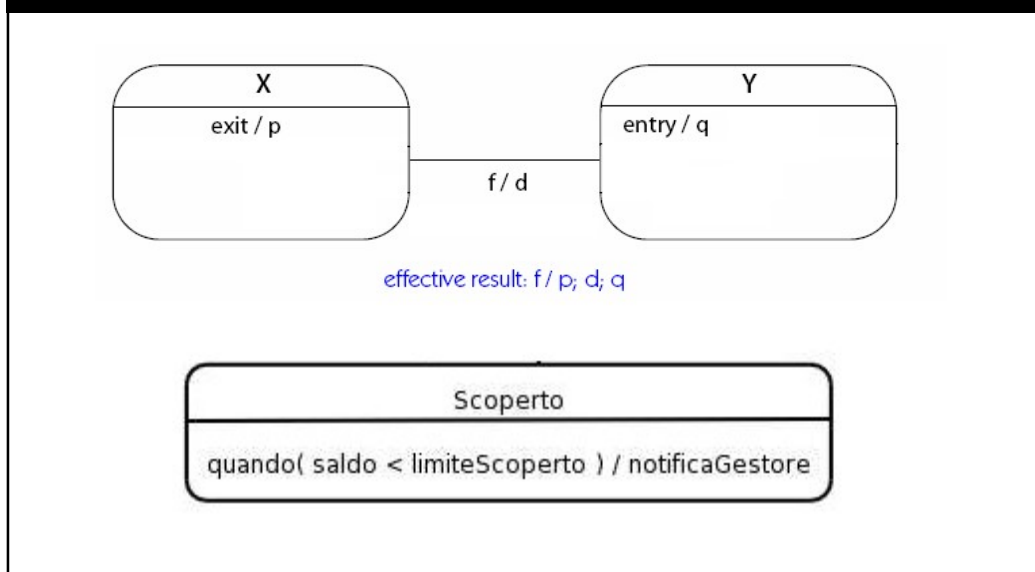
Transizioni e attività interne

- 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 contrario di tutte le altre azioni che sono atomiche:
 - consuma del tempo
 - può essere interrotta (quando un evento fa uscire dallo stato)

Sintassi

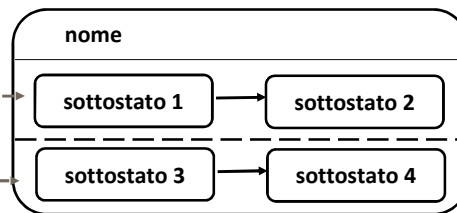
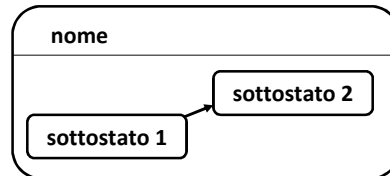


Esempi

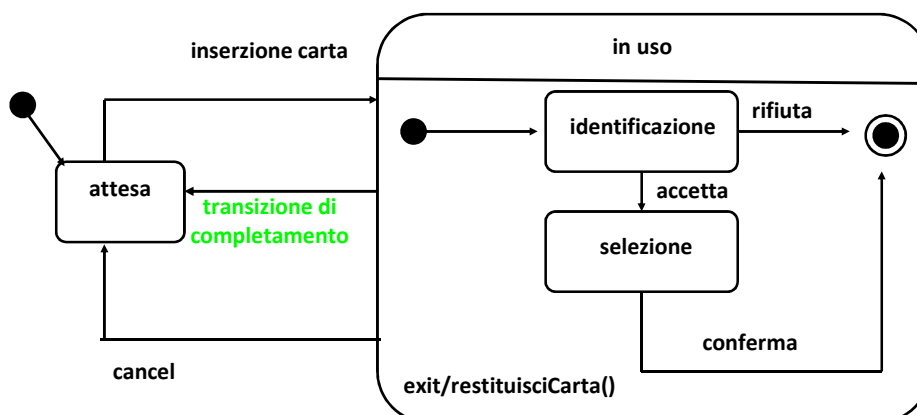


Stati composti

- Composito sequenziale:
 - Un sottostato attivo in ogni istante
- Composito parallelo:
 - sottostati attivi contemporaneamente
 - uno per regione

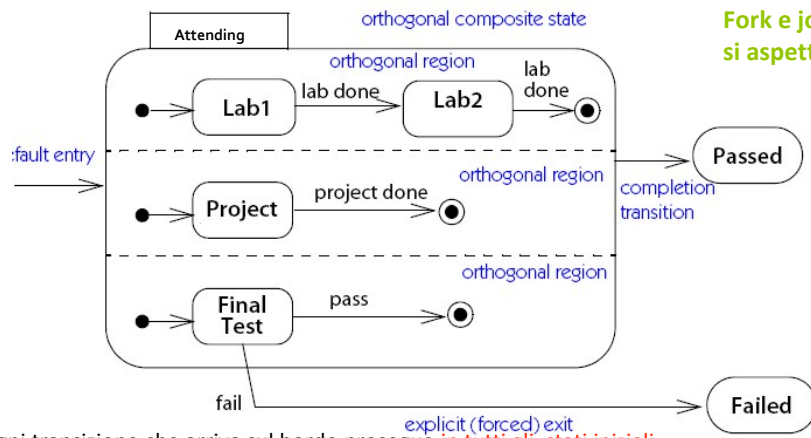


Esempio stato composito sequenziale



- 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**

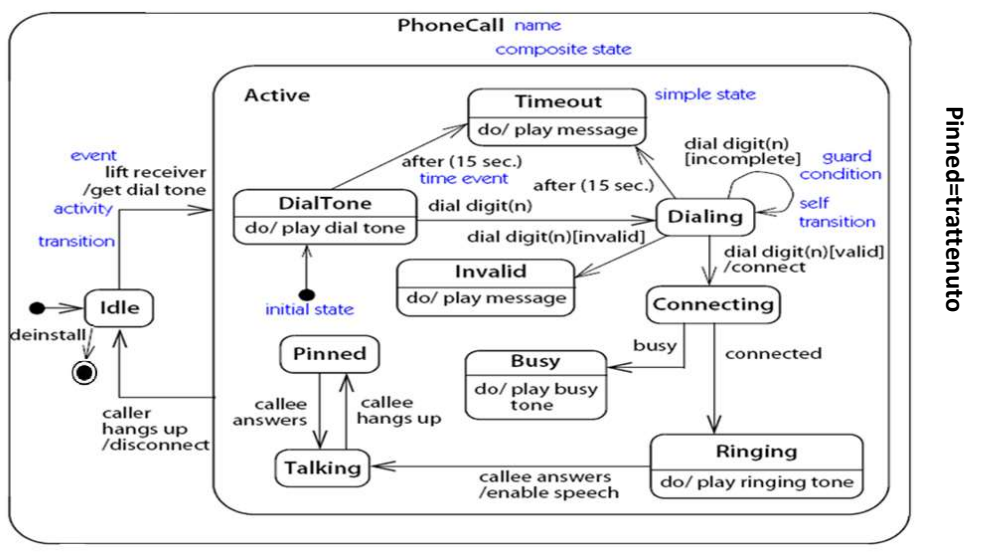
stato composito parallelo un esempio: corso di laboratorio



Fork e join implicite:
si aspetta per uscire

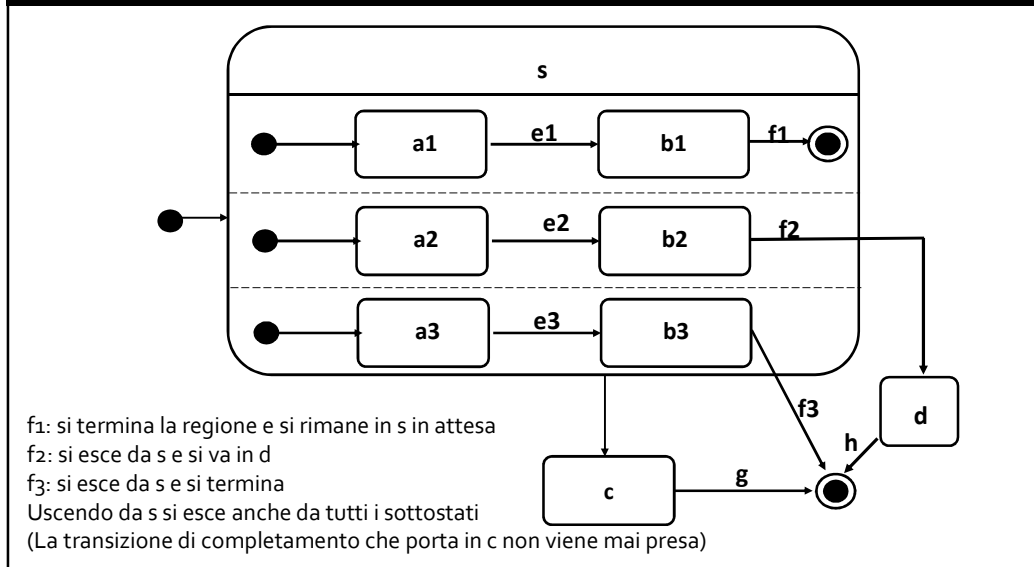
- Ogni transizione che arriva sul bordo prosegue in tutti gli stati iniziali
- Una volta raggiunti tutti gli stati finali si prosegue nella transizione di completamento
- Possono esserci transizioni che bucano il bordo e si intendono possibili dal solo stato interno a cui sono collegate

Esempio: stato composito sequenziale senza stato finale



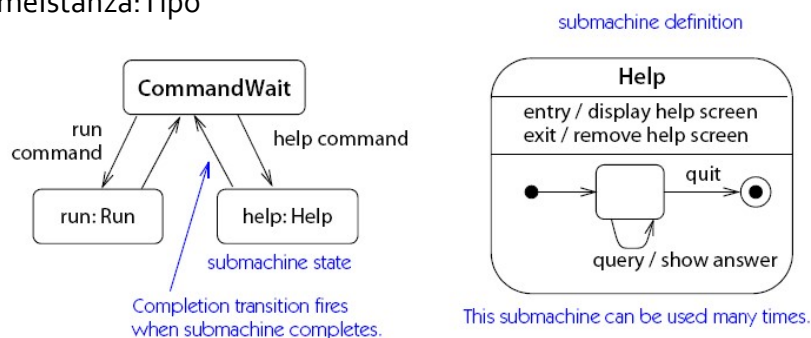
Pinned=trattenuto

Esempio di stato composito parallelo



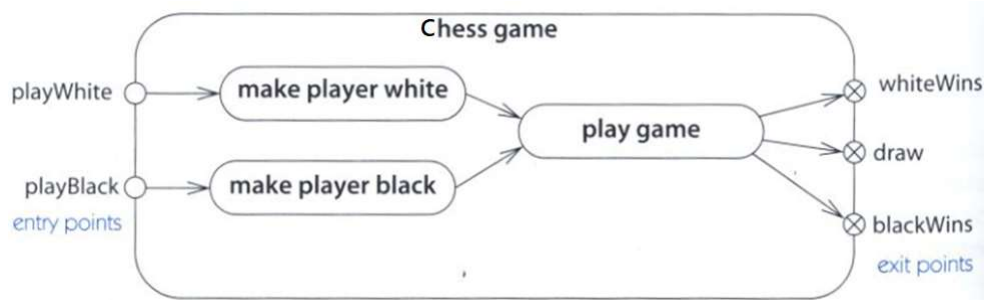
Sottomacchine

- Si usa quando 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

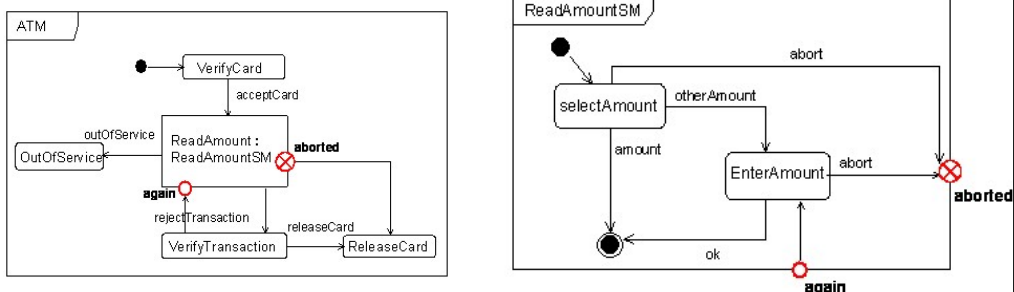


Sottomacchine: entry e exit points

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



Un altro esempio

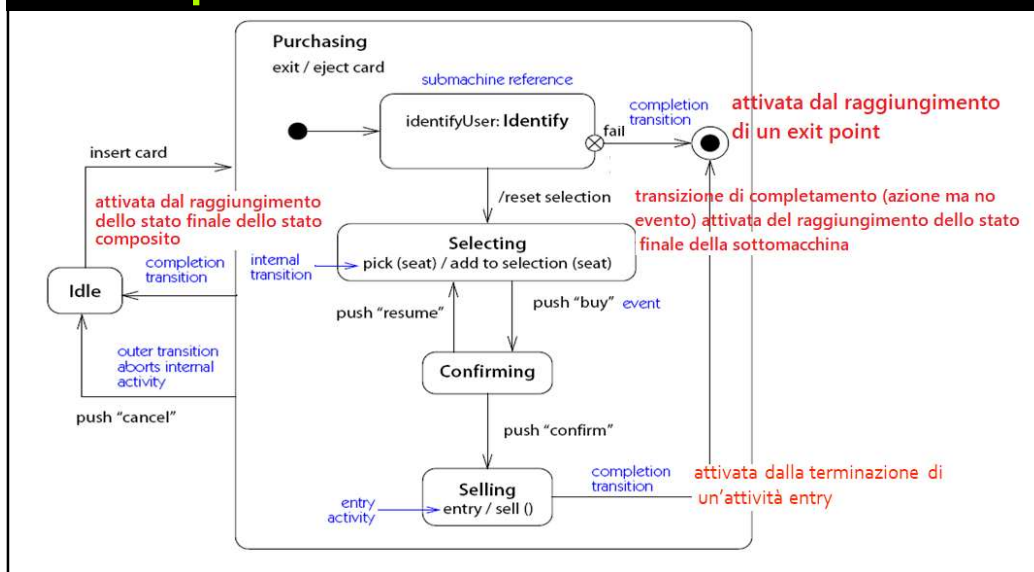


Transizioni di completamento: riassunto

- Senza evento, scattano al raggiungimento
 - Della terminazione di un'attività composta, i.e. al raggiungimento
 - Dello stato finale in un stato composto non-ortogonale
 - Degli stati finali di tutte le regioni ortogonali di un stato composto
 - 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

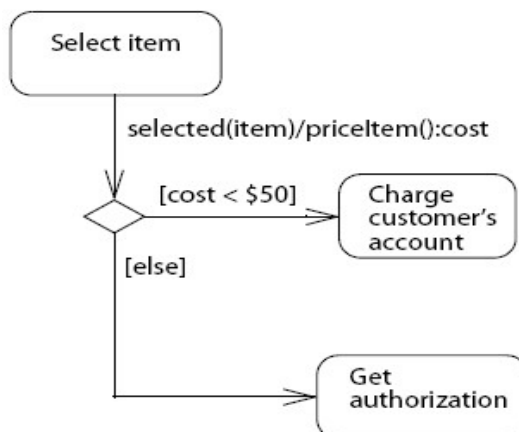
Transizioni di completamento: esempio



Altri tipi di stato (pseudostati)

Giunzione	●	Storia	⊙ (H)
Decisione	◇	Iniziale	●
Fork, join		Finale	⊙

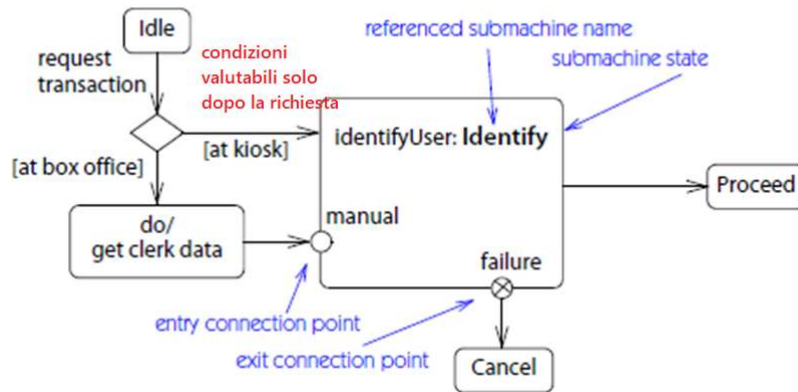
Esempio di choice



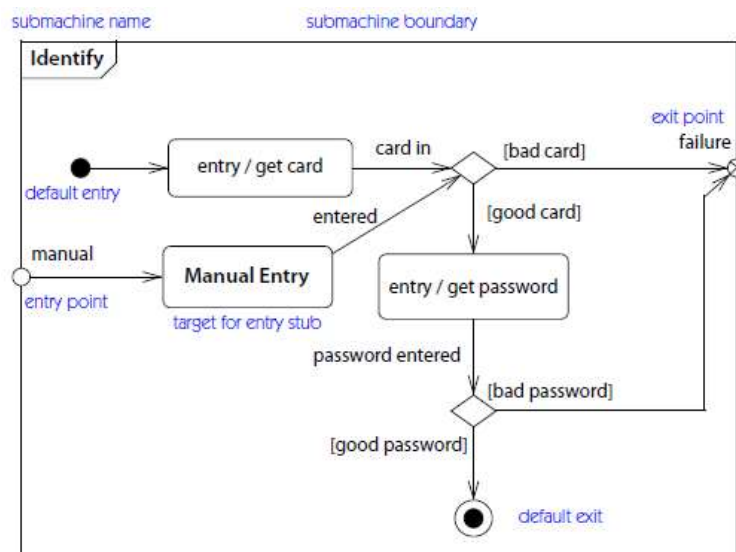
- Condizioni valutate dinamicamente
- Come per la choice dei diagrammi di attività:
 - La disgiunzione delle guardie deve valere "true"
 - È ammesso il non-determinismo

Variazione sul tema

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

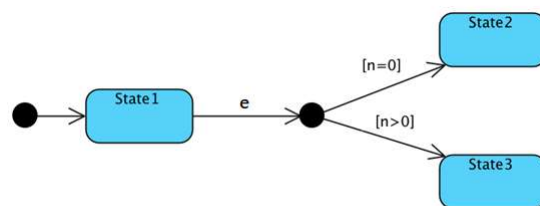


Esempio acquisto biglietti: sottomacchina Identify

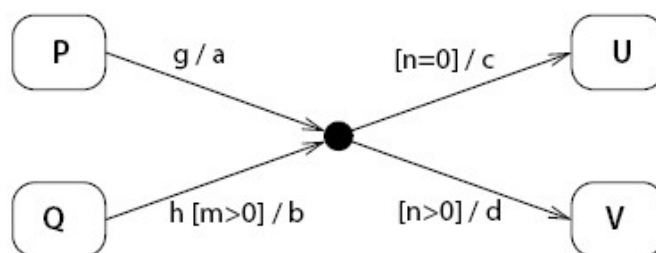


Pseudo-stato di giunzione

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



Esempio di Giunzione



Equivalent transitions:

$P \xrightarrow{g} U$ [n=0] / a, c
 $P \xrightarrow{g} V$ [n>0] / a, d
 $Q \xrightarrow{h} U$ [m>0 and n=0] / b, c
 $Q \xrightarrow{h} V$ [m>0 and n>0] / b, d

Esempio d'uso dello pseudo-stato giunzione

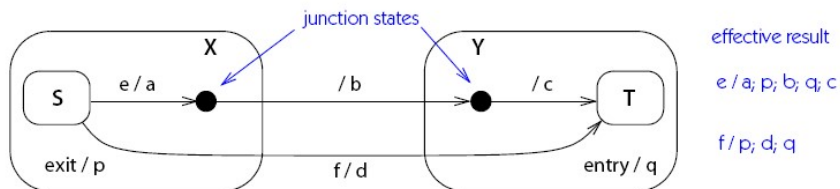
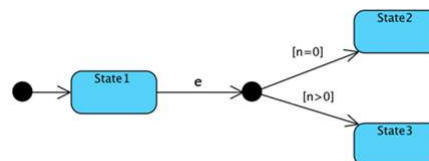


Figure 14-170. Junction pseudostates with multiple actions

59

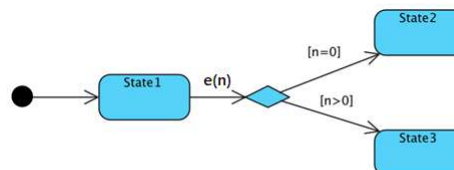
Giunzione vs choice

■ Giunzione (statica)



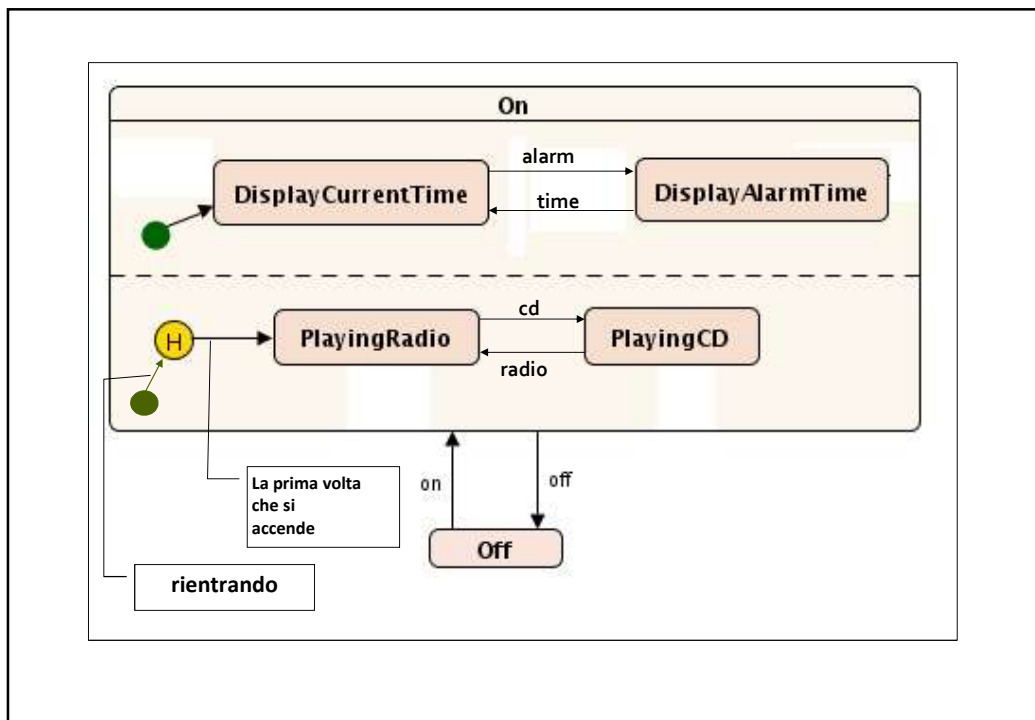
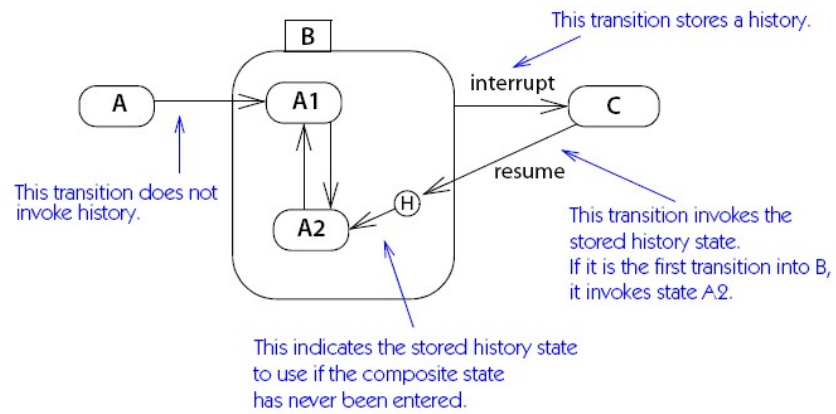
Le guardie sono valutate prima di uscire da State1. Se $n < 0$, l'evento e viene ignorato e nessuna transizione viene presa

■ Choice (dinamica)

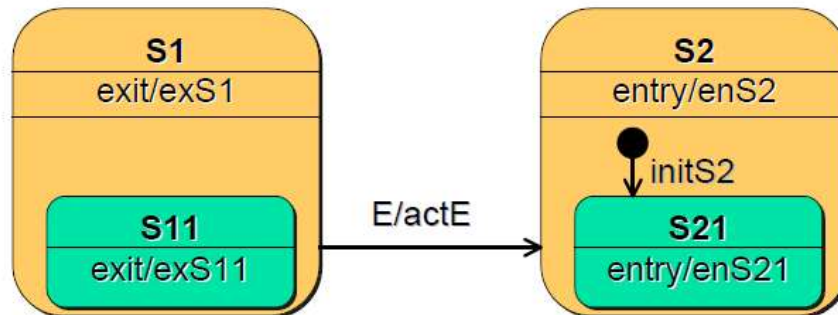


Le guardie sono valutate dopo $e(n)$. In questo esempio occorre avere garanzia che n sia maggiore o uguale a zero

History state



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 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 parla 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

Esercizi suggeriti

- Diagramma di macchina a stati
 - della classe Utente (o Situazione Utente) di Myair
 - del semaforo (ex. Semafori)
- Diagramma attività
 - EasyPark
 - Myair

Syllabus

■ Arlow:

- Paragrafi 16.3, 16.4, 16.5
- Cap 13 tranne 13.10
- Cap 19

Appendice

Da UML Reference Manual

Tipi di eventi

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)










Tipi di tranzizioni

Table 7-2: Kinds of Transitions and Implicit Effects

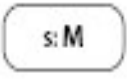

<i>Transition Kind</i>	<i>Description</i>	<i>Syntax</i>
entry transition	The specification of an entry activity that is executed when a state is entered	entry/ activity
exit transition	The specification of an exit activity that is executed when a state is exited	exit/ activity
external transition	A response 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

Tipi di stato

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	

Sottomacchine

<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	