

Diagrammi di stato e di attività: esercizi

Angelo Di Iorio

(in parte di: Gianpiero Favini)

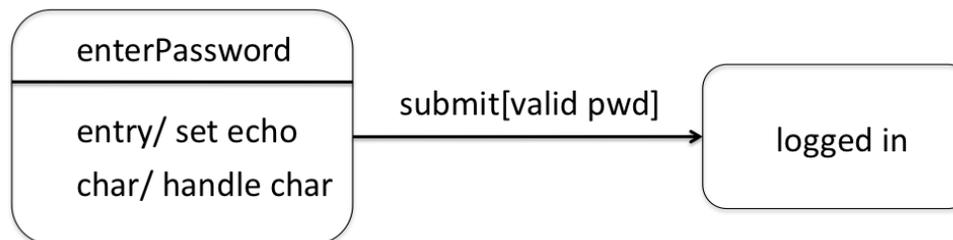
A.A. 2012-2013

Cosa sono e a cosa servono

- 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**.

Stati, eventi e transizioni

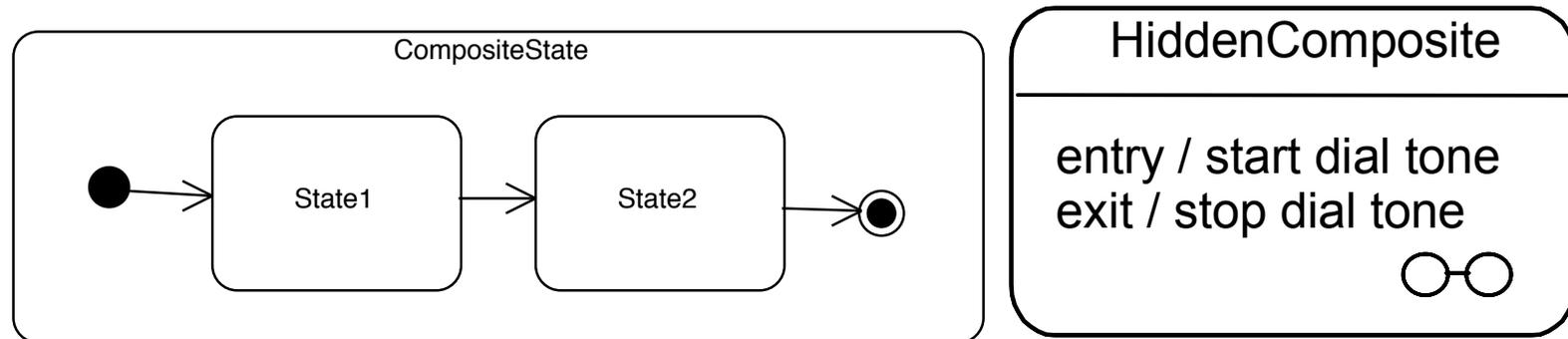
- Una **macchina a stati** descrive la sequenza di stati in cui si trova un oggetto durante il suo ciclo di vita e in risposta a eventi
- Uno **stato** è una condizione o situazione nella vita di un oggetto in cui esso: soddisfa una condizione, esegue un'attività o aspetta un evento
- Un **evento** è 'la specifica di un'occorrenza che ha una collocazione nel tempo e nello spazio'.
- Una **transizione** è 'il passaggio da uno stato a un altro in risposta ad un evento'.



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
- Uno stato può reagire ad eventi anche senza una transizione ad uno stato diverso (*internal activities*)

Stati composti

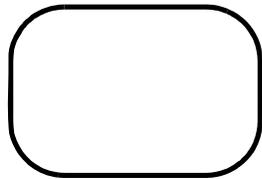


- Permettono di suddividere la complessità del modello: dall'esterno si vede un macro-stato, al cui interno vi sono altri stati.
- Si può anche creare uno stato che fa riferimento ad un'altro diagramma di macchina a stati (*submachine state*).
- Si può usare un'icona per rappresentare uno stato composto il cui comportamento interno non è mostrato.

Concorrenza e sincronizzazione

- Gli stati compositi sono utili per modellare la concorrenza. Si divide lo stato composito in (sotto-)diagrammi **ortogonali** eseguiti in **mutua esclusione**
- Gli stati compositi sono inoltre utili per modellare la sincronizzazione. Si divide lo stato composito in (sotto-)diagrammi e si usano gli operatori di *fork* e *join*

Diagramma di attività: ingredienti



Action node



Object node



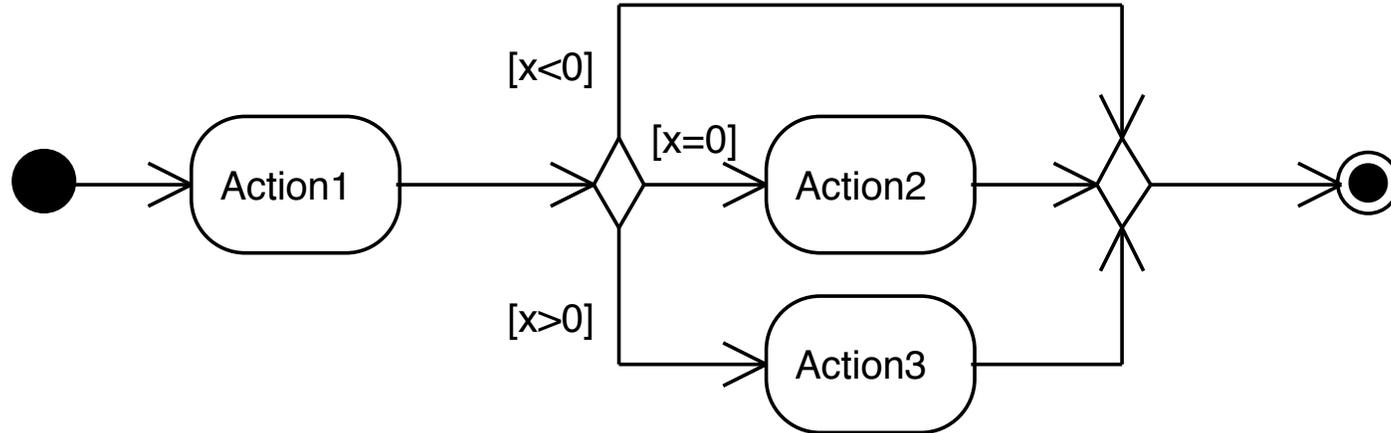
Control nodes

- Nodi **azione**: specificano unità di comportamento.
- Nodi **oggetto**: specificano oggetti usati come input e output di azioni.
- Nodi **controllo**: specificano il flusso dell'attività.

Diagramma di attività: semantica

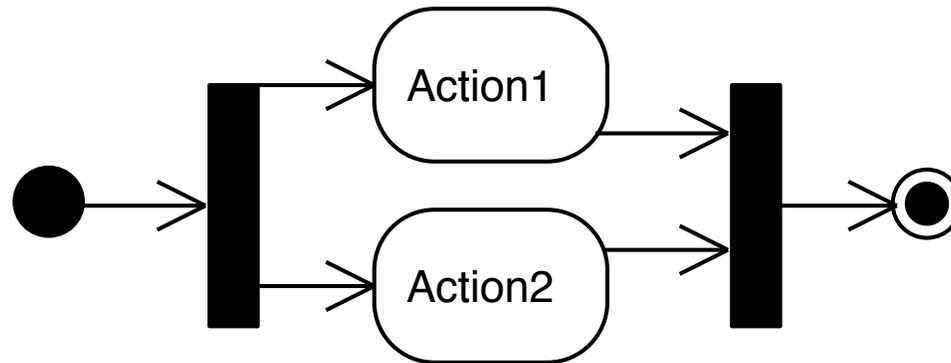
- Il flusso dell'attività è definito da entità, dette **token**, che viaggiano lungo il diagramma.
- I token possono rimanere fermi in un nodo azione/oggetto *in attesa* che si avveri una condizione su una freccia, oppure una preconditione o postcondizione su un nodo.
- Un nodo azione viene eseguito quando sono presenti token su tutti gli archi in entrata, e tutte le preconditioni sono soddisfatte.
- Al termine di un'azione, sono generati *control token* su tutti gli archi in uscita.

Nodi iniziali, finali, decisione e fusione



- Il disco nero marca l'inizio dell'attività (nodo iniziale, genera token). Quando un token raggiunge un disco nero bordato (nodo finale), l'attività ha termine.
- I nodi **decisione** hanno un input e vari output **mutuamente esclusivi**: copiano i token in entrata su uno degli output.
- I nodi **fusione** hanno vari input e un solo output, sul quale vengono indirizzati tutti i token in ingresso.

Nodi fork/join



- I nodi **fork** hanno un ingresso e varie uscite: i token in ingresso sono duplicati su tutte le uscite.
- I nodi **join** hanno vari ingressi e una sola uscita: quando sono presenti token su tutti gli ingressi, viene prodotto almeno un token in uscita.
- I nodi fork dividono un'esecuzione in più flussi concorrenti, i nodi join sincronizzano e riuniscono i flussi.

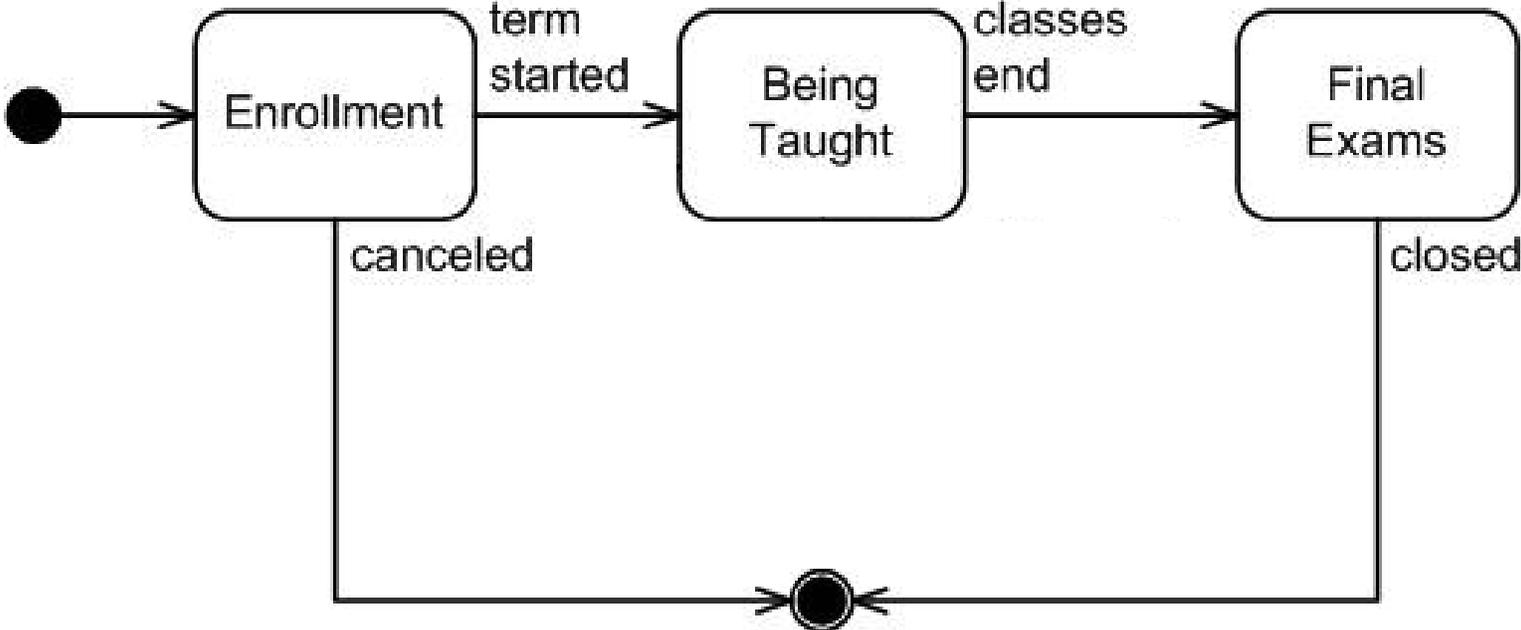
Esercizi

Esercizio seminario

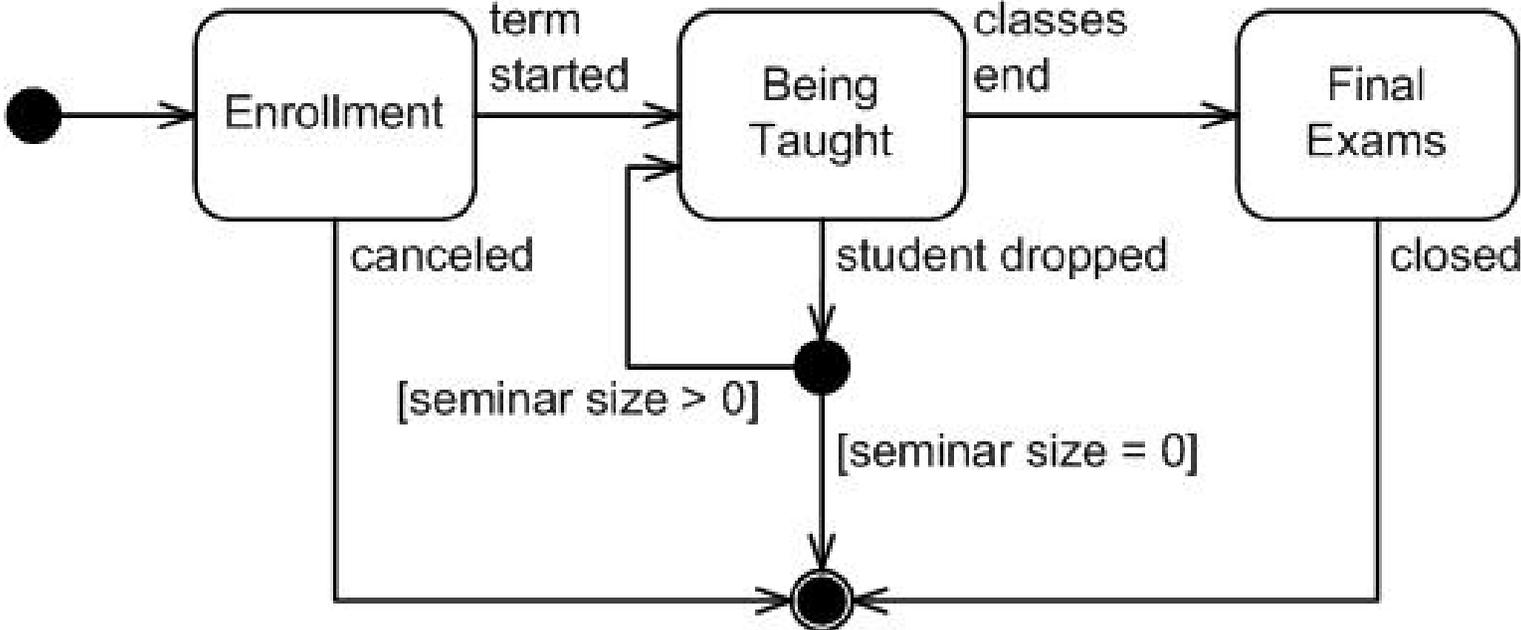
Disegnare un diagramma di stato (relativo a studente) che modella il seguente dominio:

- Uno studente può iscriversi ad un seminario fino all'inizio del semestre. Può anche cancellare l'iscrizione prima dell'inizio. Lo studente è tenuto a dare un esame e può abbandonare il seminario in qualunque momento.
- Il seminario inoltre è interrotto se tutti gli studenti abbandonano (il numero di partecipanti è 0)

Esercizio seminario



Esercizio seminario

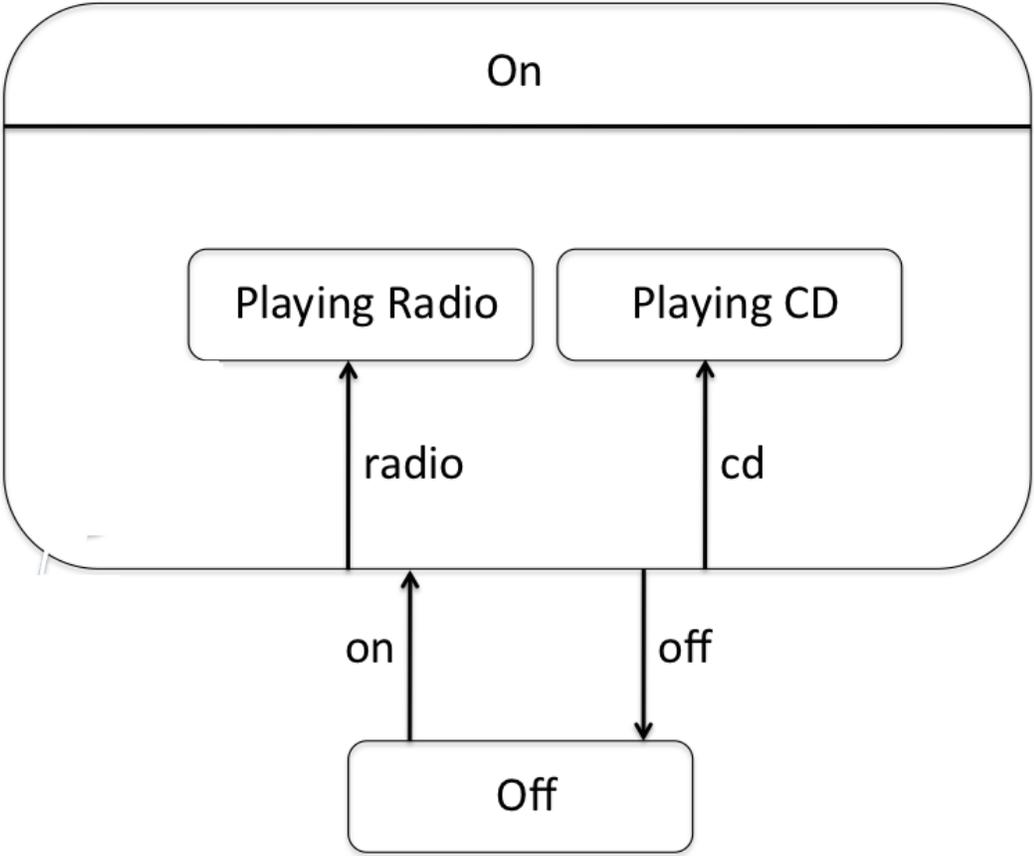


Esercizio radiosveglia

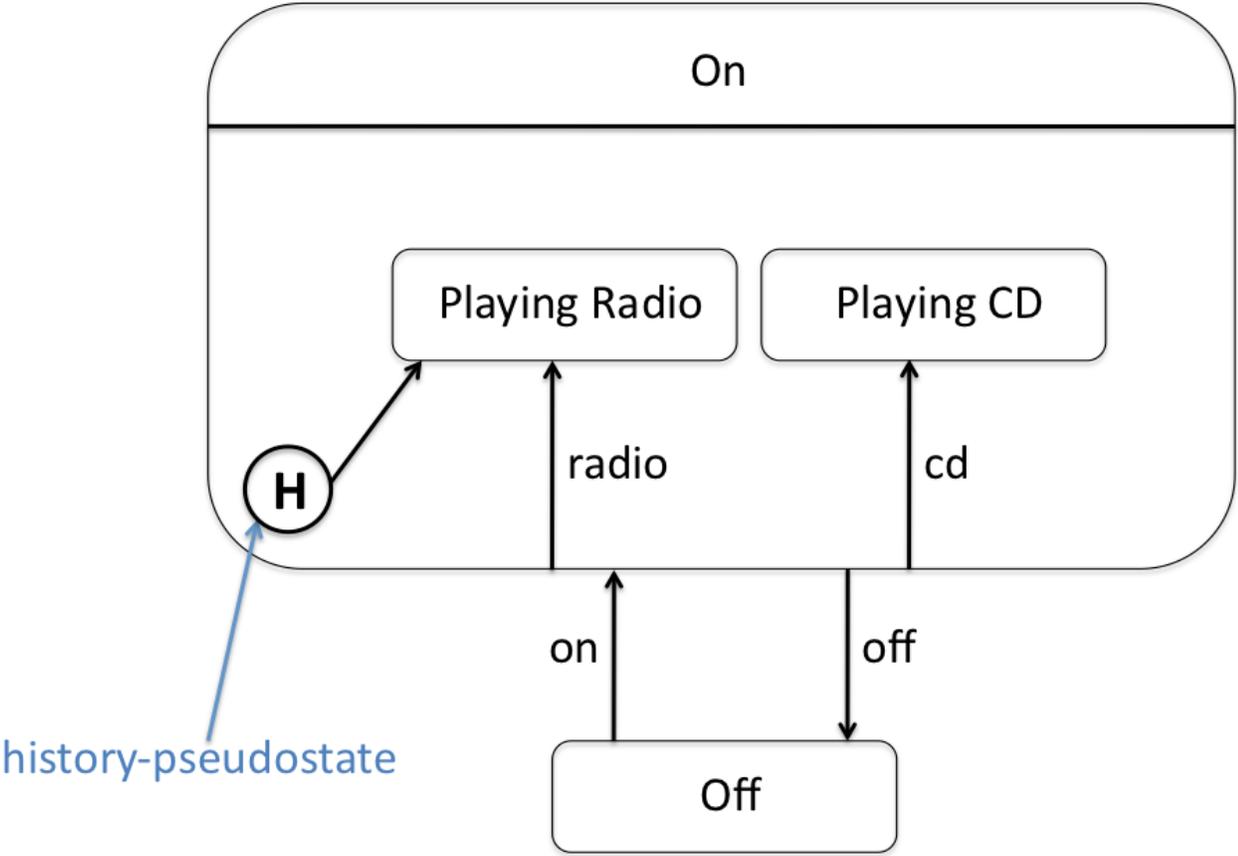
Disegnare un diagramma di stato relativo ad una radiosveglia:

- La radiosveglia può essere spenta o accesa e, se accesa, far ascoltare la radio o un CD.
- Alla riaccensione la sveglia riparte con la radio o il CD in base alla selezione attiva in fase di spegnimento
- La radiosveglia permette inoltre di ascoltare musica o visualizzare l'orario
- E' possibile visualizzare sia l'ora corrente che l'ora della sveglia

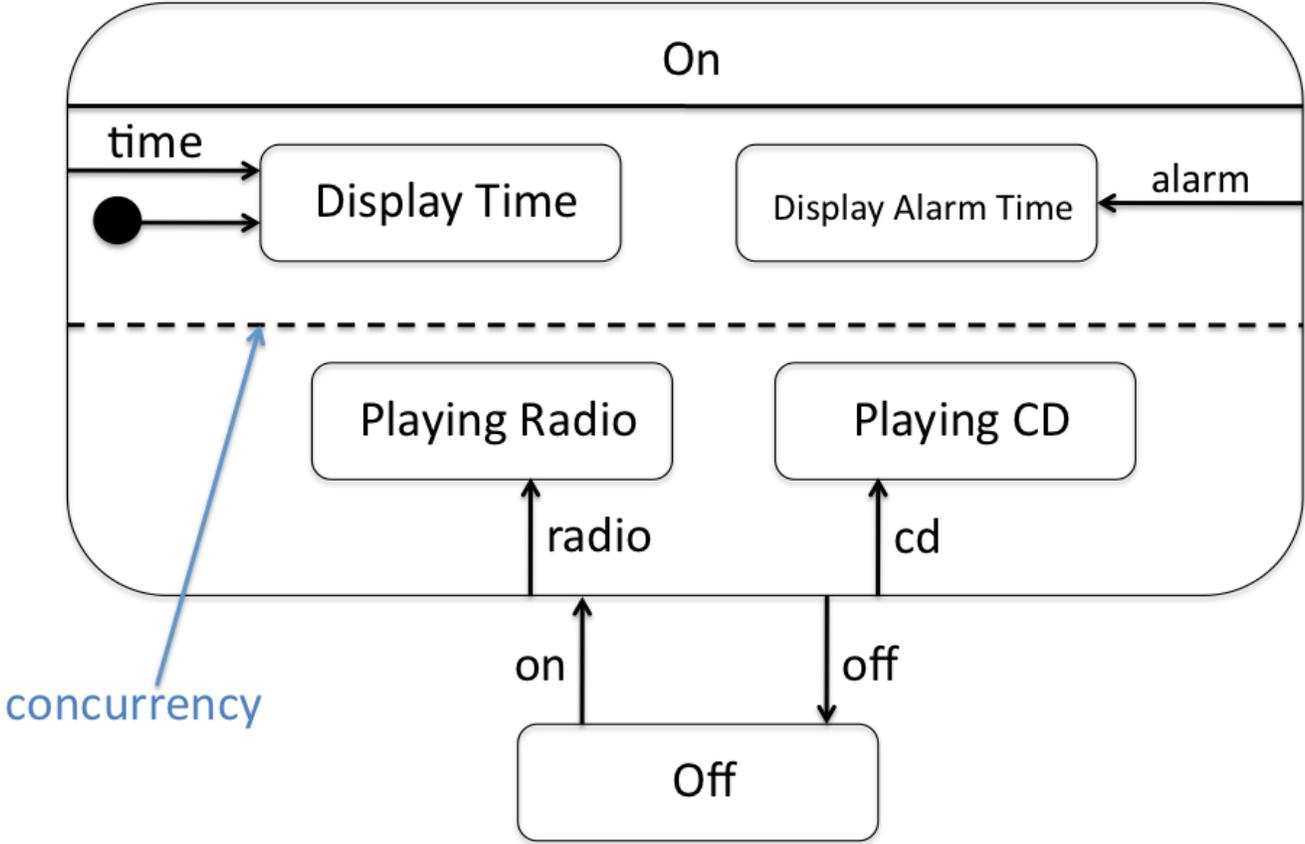
Esercizio radiosveglia



Esercizio radiosveglia



Esercizio radiosveglia

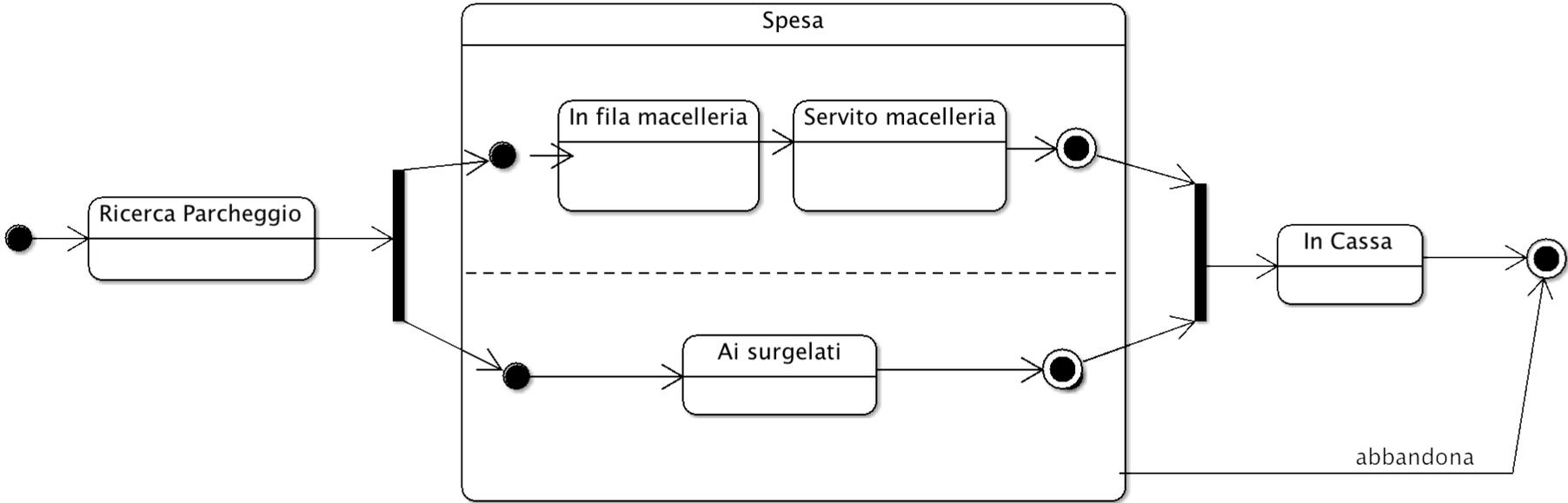


Esercizio spesa

Disegnare un diagramma di stato (relativo ad un cliente) che descrive una spesa in un supermercato:

- Il cliente cerca parcheggio prima di iniziare a fare la spesa
- Deve acquistare surgelati e carne e fa la fila in macelleria prima di essere servito.
- Finita la spesa paga alla cassa ma può decidere di andare via in qualunque momento.

Esercizio spesa



Esercizio cena

Disegnare un diagramma di attività relativo all'organizzazione di una cena:

- Si può cenare a casa, a ristorante o non cenare del tutto
- La cena a casa richiede di apparecchiare la tavola e cucinare, dopo aver trovato la ricetta e gli ingredienti necessari.

Esercizio cena

