

Macchina a stati

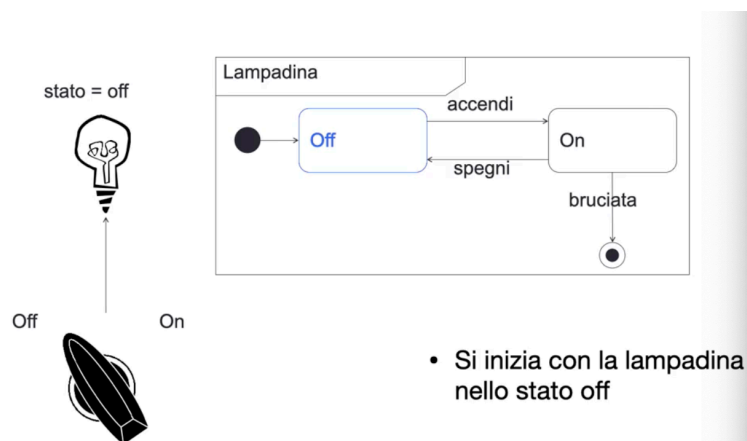
Friday, 31 March 2023

14:18

- Servono per modellare il comportamento dinamico di classificatori
 - o Classi
 - o Casi d'uso
 - o Sotto sistemi
 - o Interi sistemi
- La useremo nella progettazione per mostrare il comportamento delle classi ordine
 - o Risponed ad eventi esterni
 - o Ha un ciclo di vita definito
 - o Il suo comportamento potrebbe dipendere dal passao

Tipologie degli oggetti:

- o Indipendenti dallo stato
 - o Dipendenti dallo stato
 - Modellano comportamenti di un modello reattivo/complesso
- Cos'è?

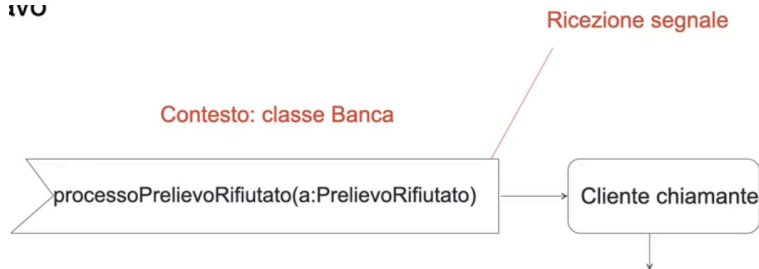


- o Cerchio pieno = Stato iniziale
 - o Accendi/Spegni = Eventi
 - Sono possibili avere delle chiamate interne
- Abbiamo anche eventi di segnale che sono pacchetti asincroni inviati tra oggetti
- Non ha operazioni siccome ha l'unico scopo di trasportare informazioni

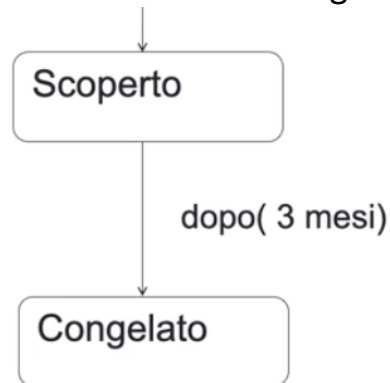




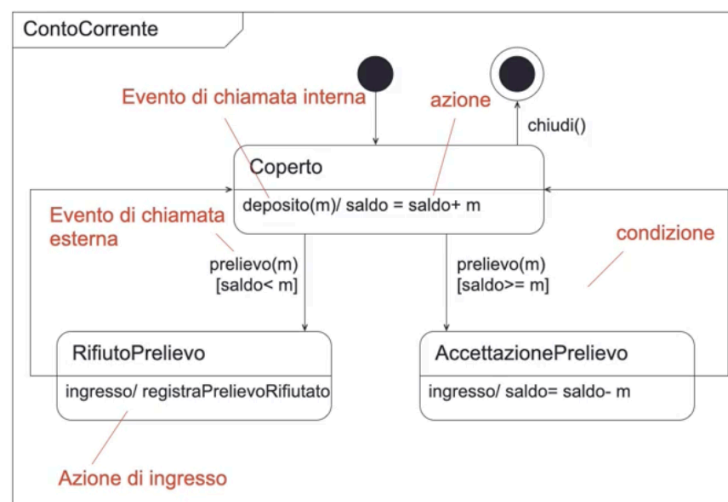
E per riceverlo



- Bruciata = Stato finale
- Frecce = Transizioni
 - E' possibile avere condizioni
 - Possiamo metterci degli eventi temporali, aka delay

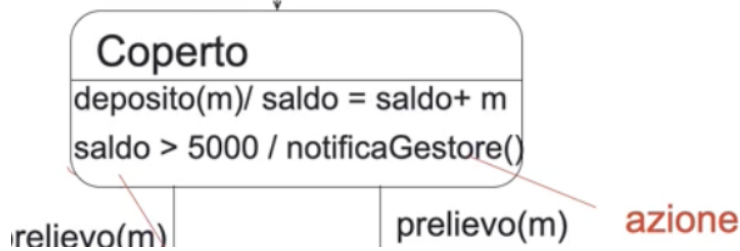


- On/off = stati
 - Condizioni di una vita di un oggetto dove soddisfa condizione, esegue attività, aspetta evento



- Azione: Sotto deposito possiamo anche metterci delle condizioni che.

Se ve re chiamano una funzione



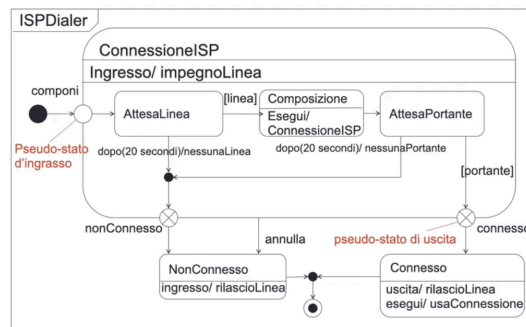
■ Stati composti

Aka uno stato che ha degli sotto stati

E possono essere:

- Semplici

Con 1 sola regione



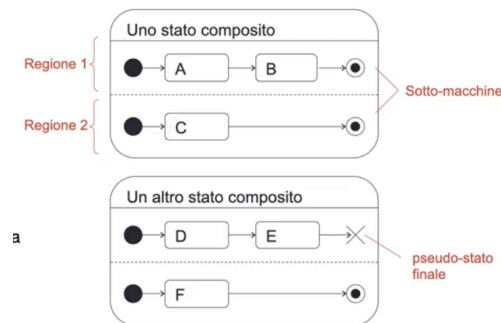
Noi qui iniziamo con ConnessioneISP

Che è il nostro pseudo stato d'ingresso

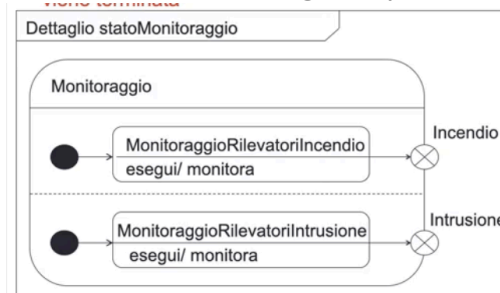
E da qui entriamo da impegno linea, dove abbiamo uno stato d'ingresso.

- Ortogonali

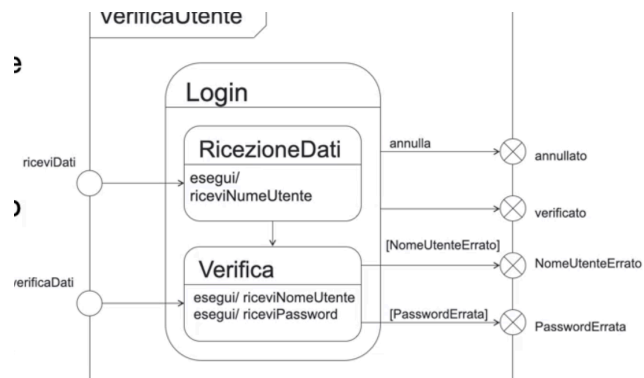
Con 2 o più regioni



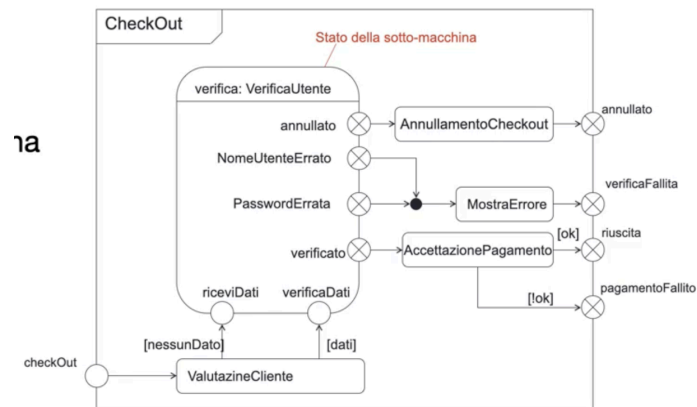
Qui tutte le due regioni partono concorrentemente



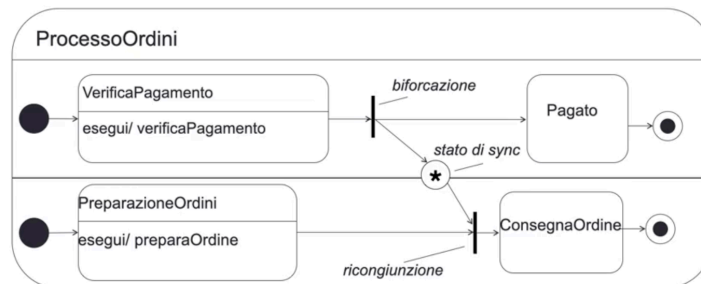
Qui abbiamo una uscita sincronizzata, ed è uno o l'altro



- Questo poi la possiamo utilizzare come "stato della sotto macchina", aka diventa una funzione dove noi diamo input, e restituisce output

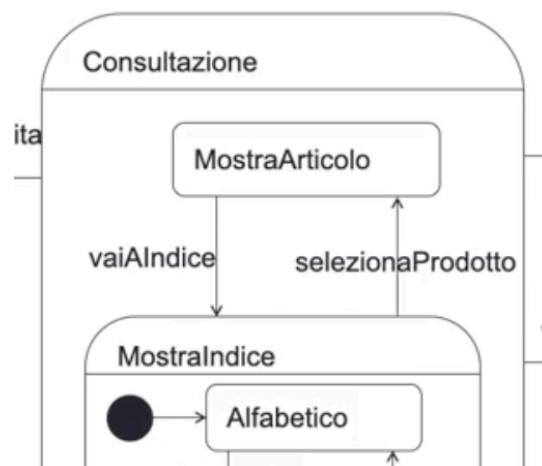


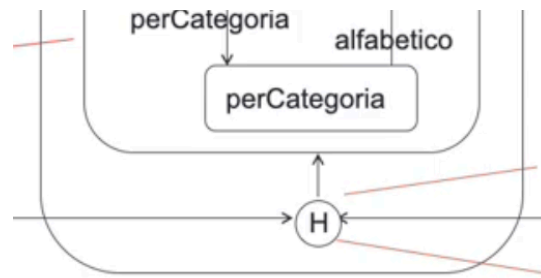
- Biforcazione e sync



Con biforcazione creiamo thread, in sync si sincronizzano N
E la biforcazione può unire più thread

- Stati con memoria semplice
Praticamente





Quando noi usciamo da questo stato, questo si ricorda in che stato eravamo, e quando ritorniamo dentro ci reindirizza dove eravamo precedentemente
Possiamo metterci * vicino ad H per poter far ricordare anche livelli più bassi, es: mostraIndice -> perCategoria