

# Ingegneria del Software

## 8. Diagrammi di attività

Dipartimento di Informatica  
Università di Pisa  
A.A. 2014/15

# so far...

- Modello del dominio
  - Modello statico: diagrammi delle classi
  - Modello dinamico : **diagrammi di attività (business model)**  
[diagrammi di macchina a stati]
- Requisiti
  - Modello statico: diagramma dei casi d'uso
  - Modello dinamico: narrazioni associate ai casi d'uso

# diagrammi di attività

- Modellano il flusso di lavoro (workflow, business model)
  - di una computazione (prospettiva software) o
  - di un processo/attività (business)
- Una attività descrive la coordinazione di un insieme di azioni, ed è centrata
  - su sequenzialità e concorrenza delle azioni, nonché
  - sulle condizioni che le abilitano,
  - piuttosto che sui classificatori che eseguono queste azioni
- Predecessori: flow chart e reti di Petri

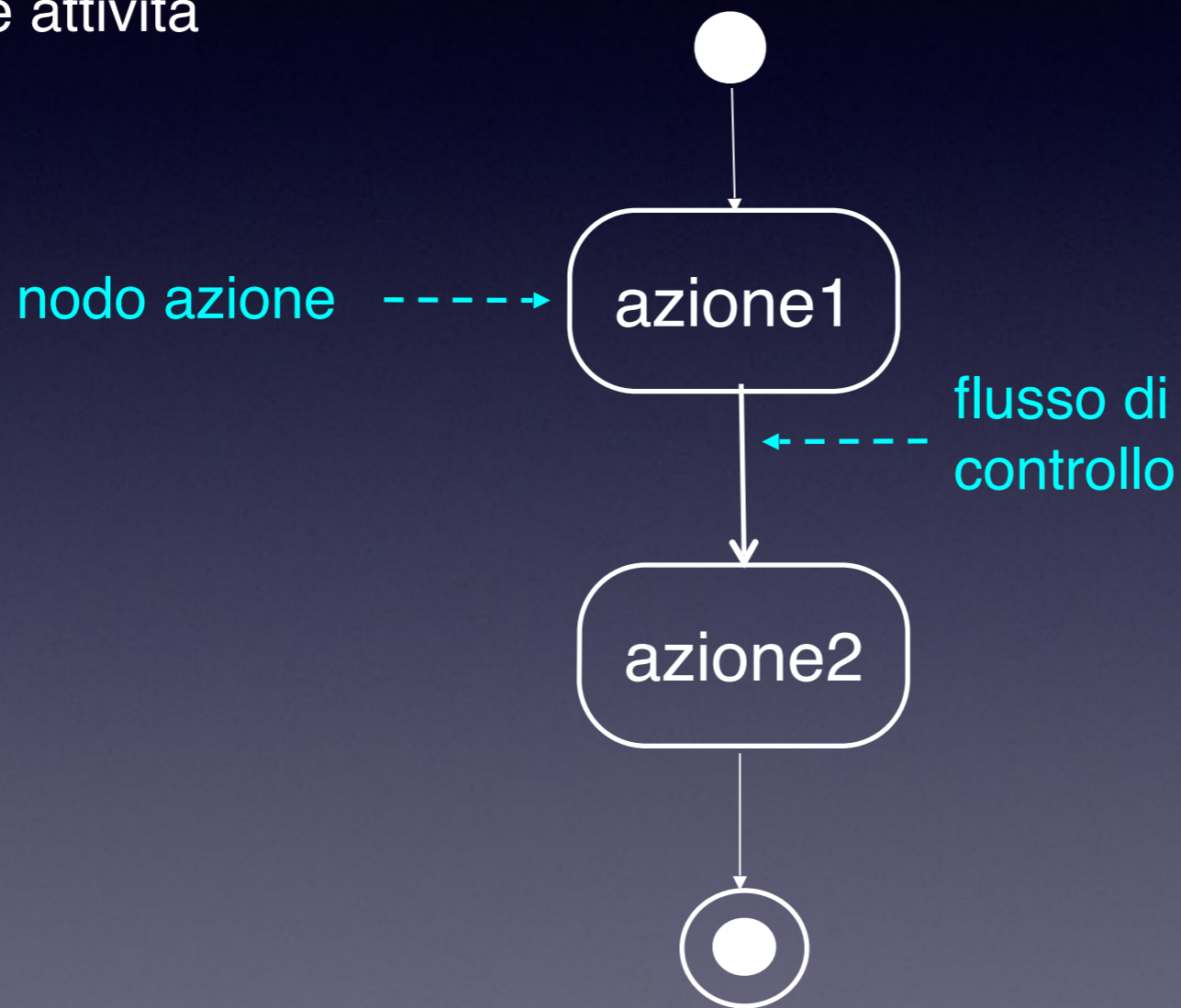
# inizio e fine

Nome attività

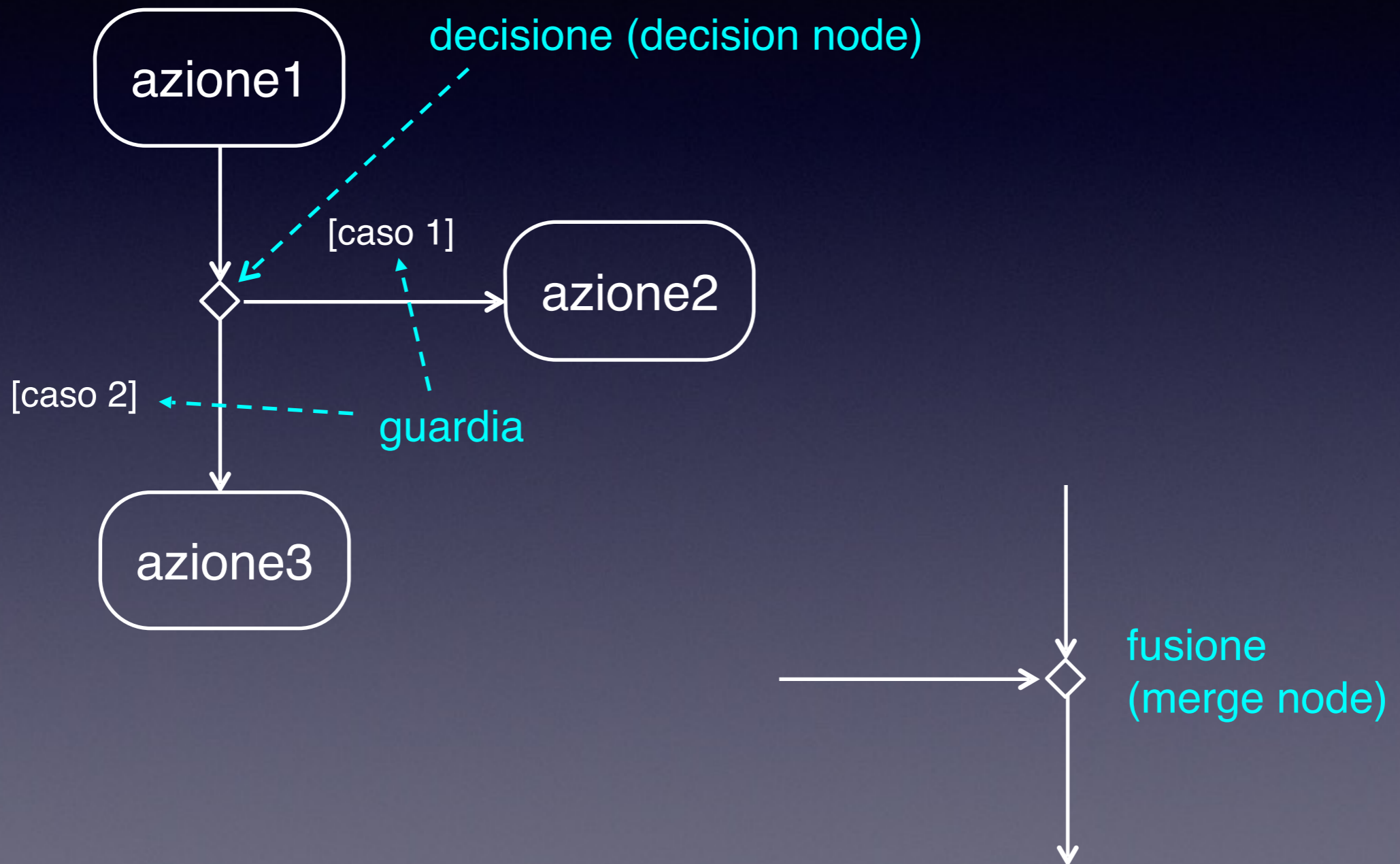


# sequenza

Nome attività

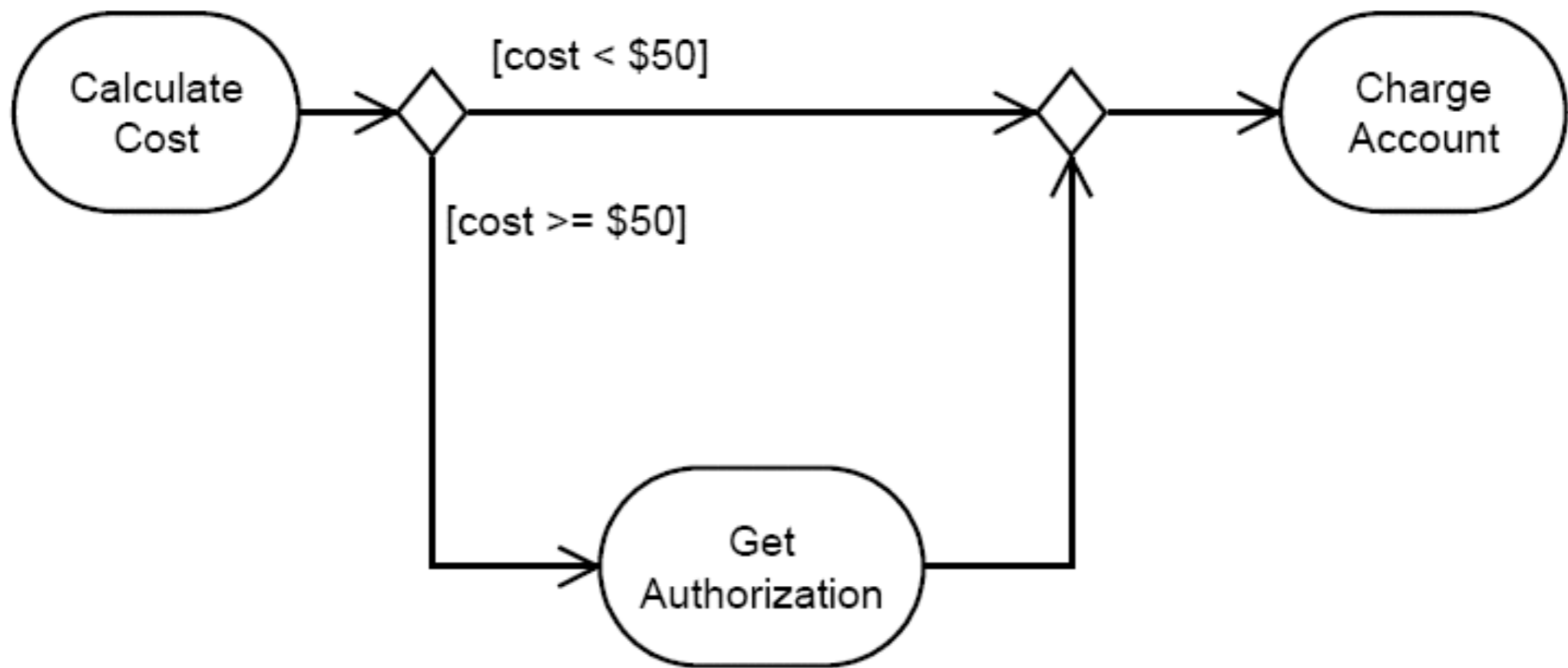


# scelta



# un esempio

*esempio “Branching” e “merging”*

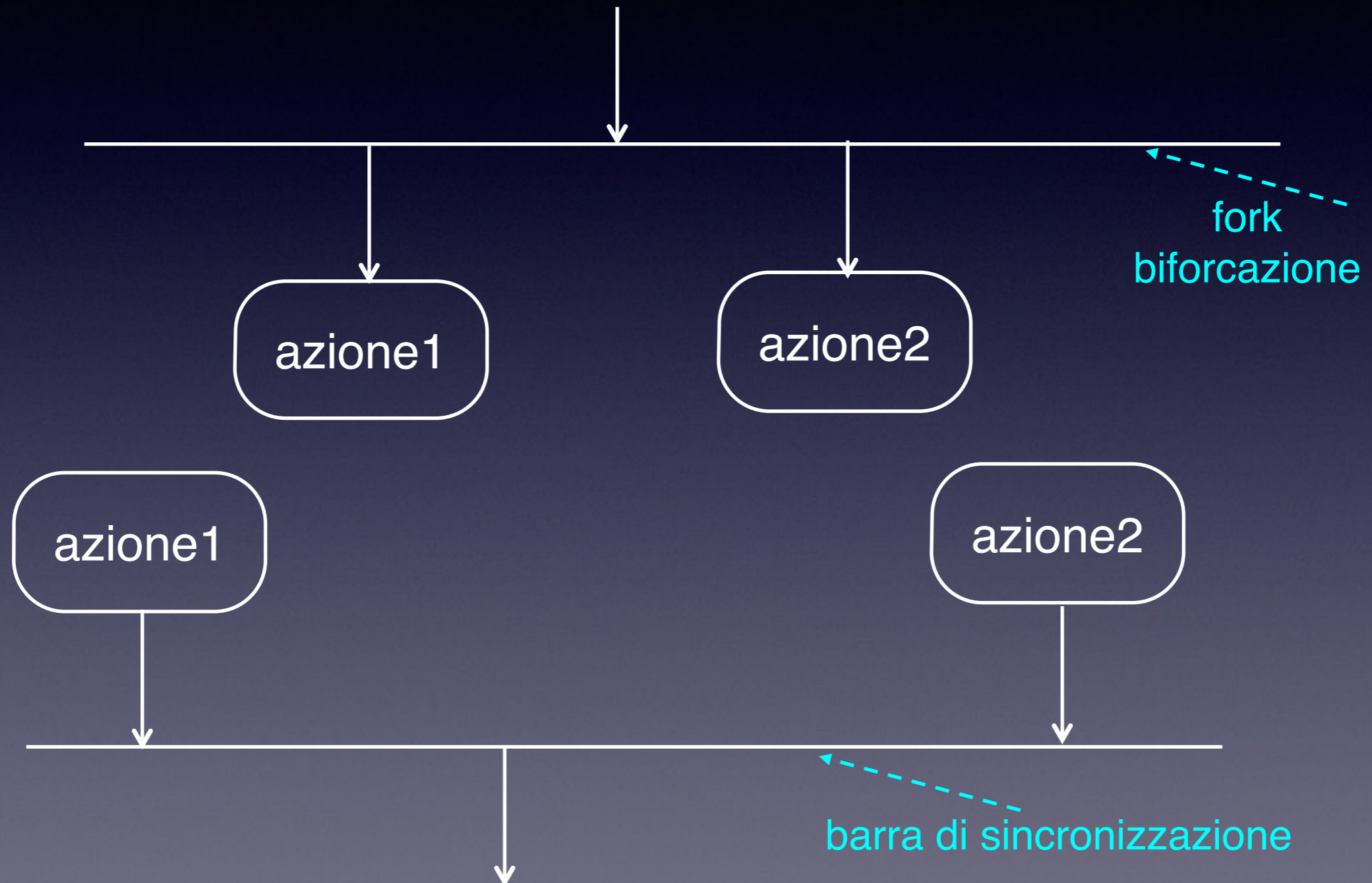


# decisione e fusione

- Semantica a “token”
- Le guardie devono coprire tutte le possibilità
  - in caso si utilizza [else]
- È bene che le guardie siano mutuamente esclusive
  - altrimenti si verifica un comportamento non definito
- Le condizioni di guardia sono sempre indicate tra [ ]
  - (come avviene in generale in UML)
- Ad un nodo decisione non corrisponde obbligatoriamente un nodo fusione
  - il ramo potrebbe essere chiuso da un nodo di fine flusso



# fork e join

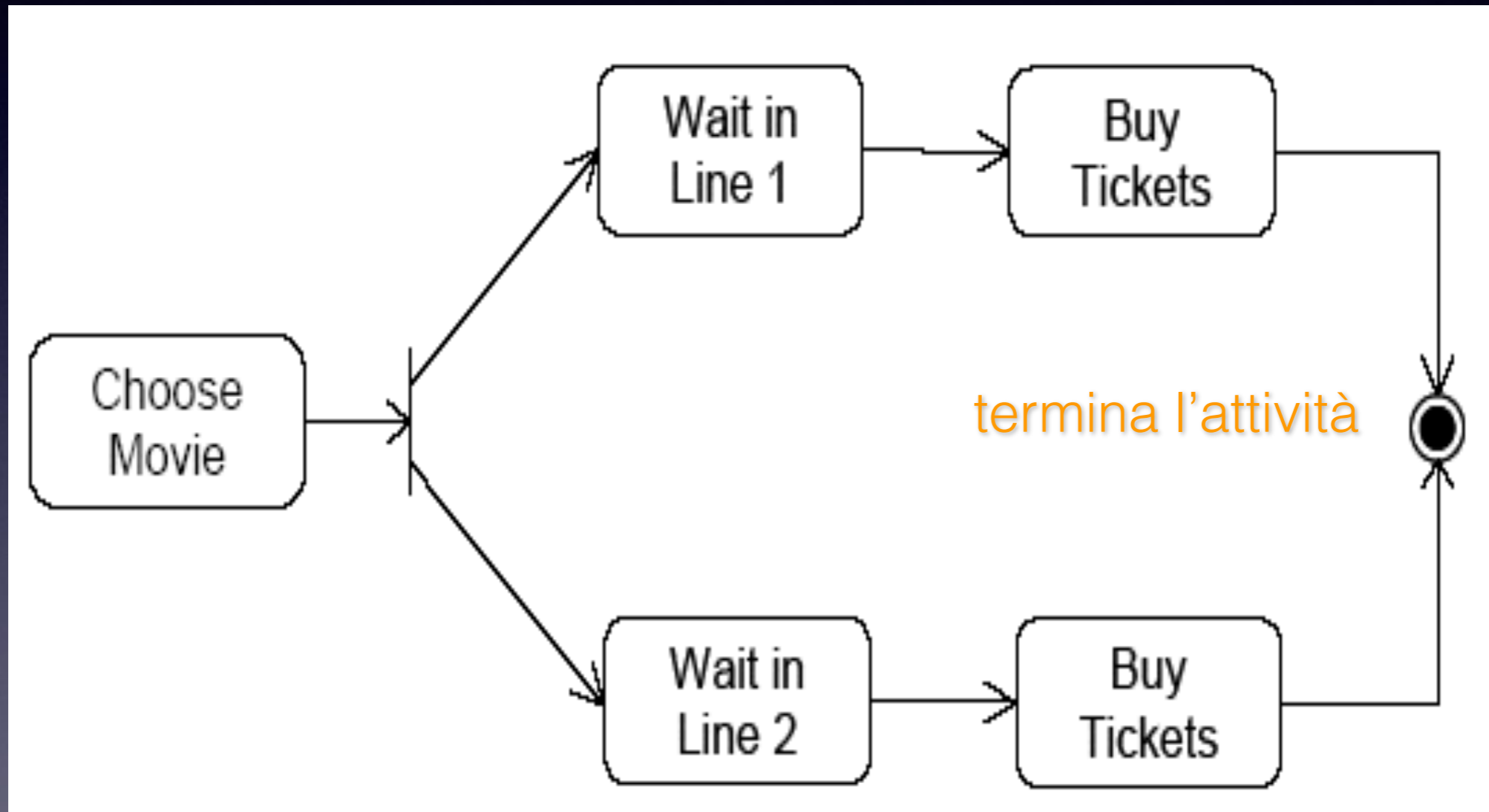


# fork e join

- Ancora semantica a token
- A un fork non corrisponde obbligatoriamente un join

- Una attività non può essere interrotta da segnali esterni (azioni atomiche)
- In UML 2 solo una freccia entrante per azione (standard precedente: una azione inizia solo quando tutti i flussi l'hanno raggiunta)

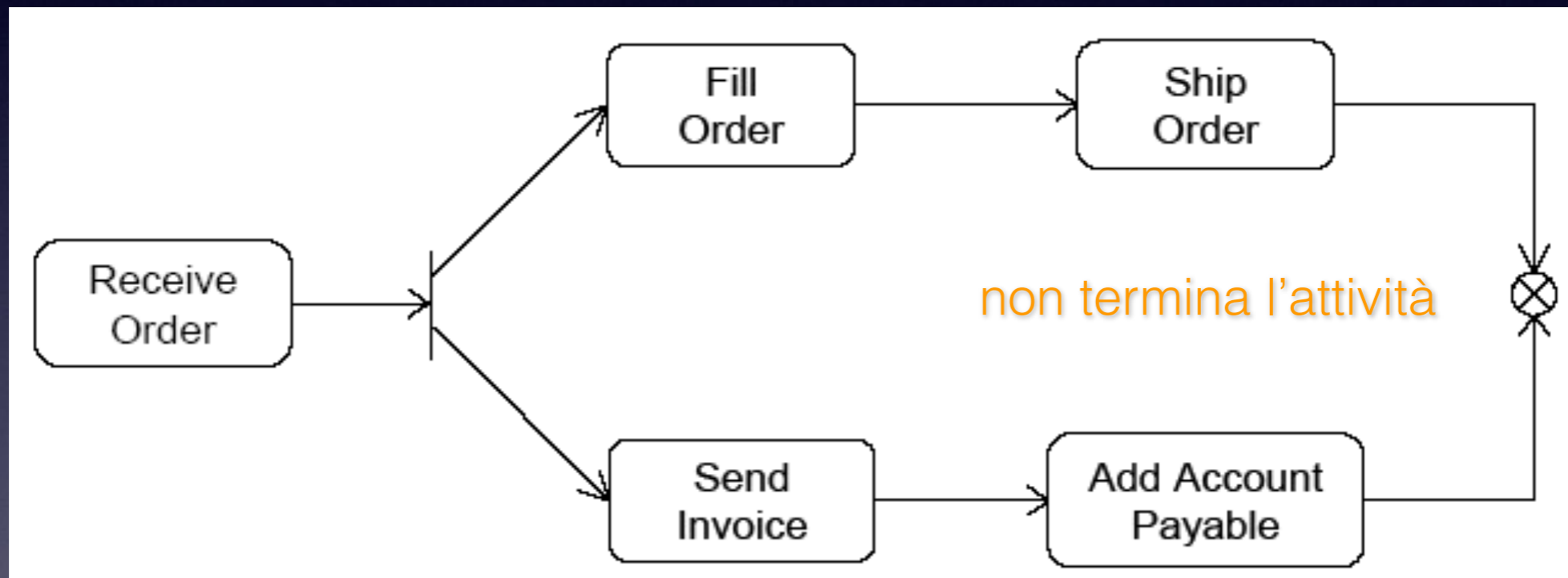
# un esempio



in due persone ad acquistare un biglietto, il primo che compra finisce tutto

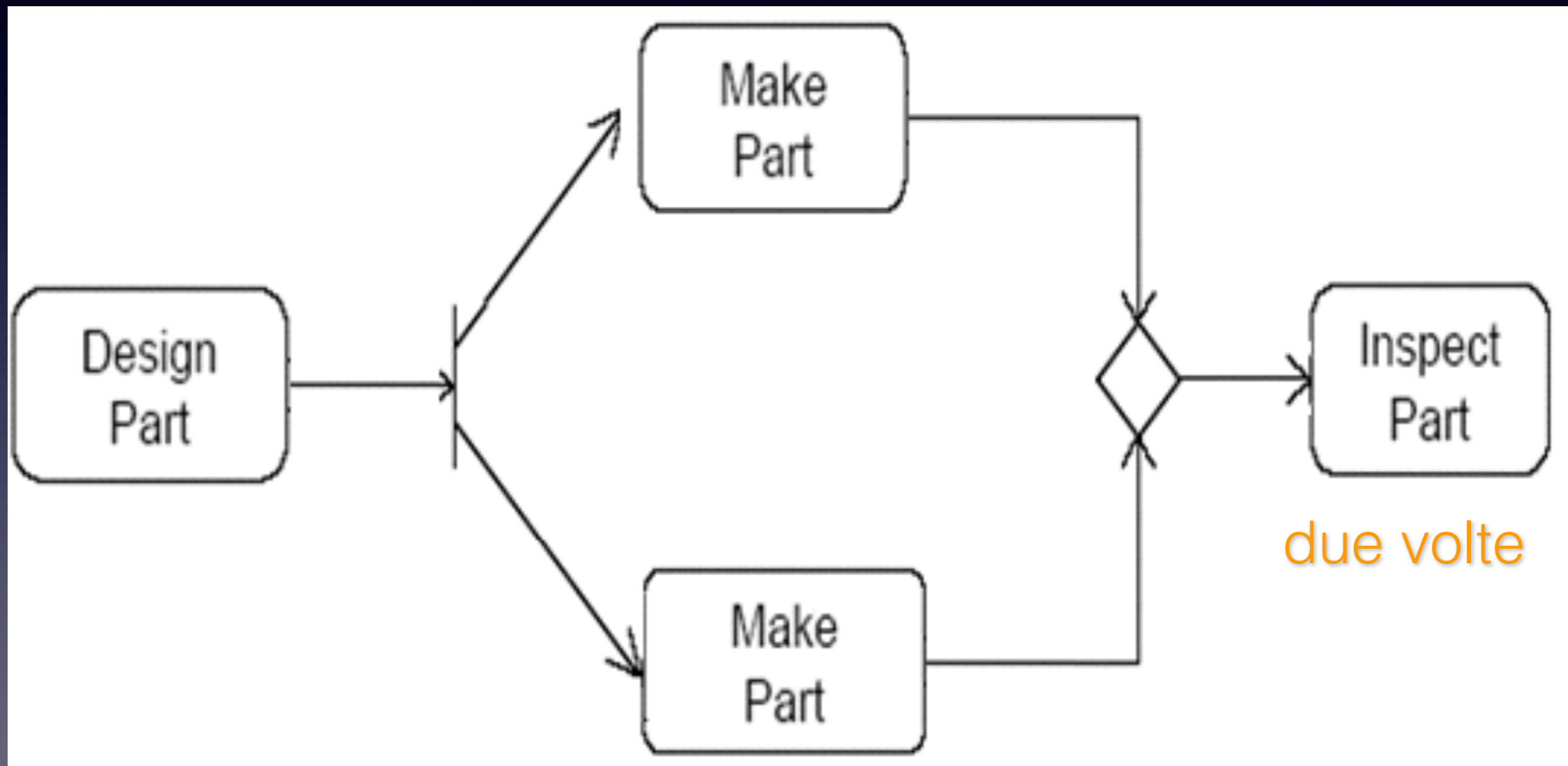
# fine di flusso

- Un nodo **finale** in una attività distrugge tutti i token che lo raggiungono. Non ha altri effetti sull'esecuzione dell'attività [def.]



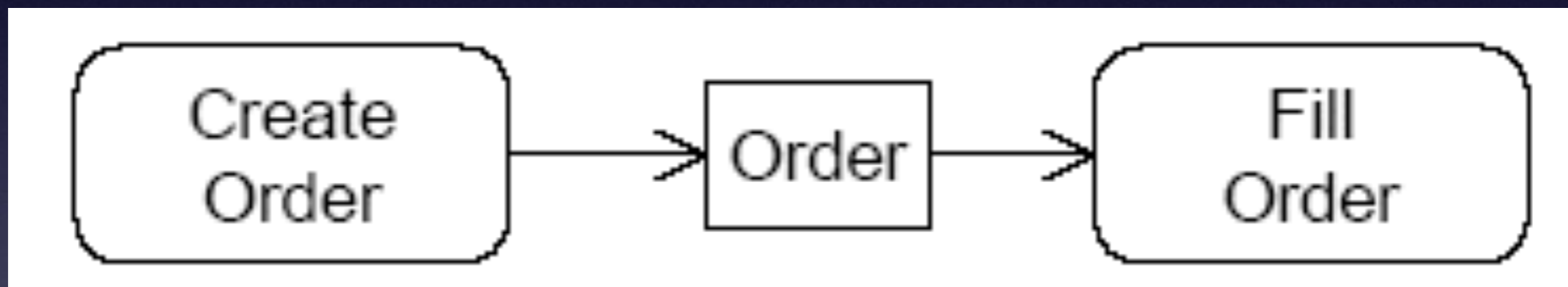
- Usare con cautela: qui meglio una join
- Dubbia la semantica: finisce l'attività quando sono finiti i token?
  - Superstructure non lo dice, ma parrebbe di no... [v. lucido con fig12.27 sstruct.]

# fork e merge



# dati

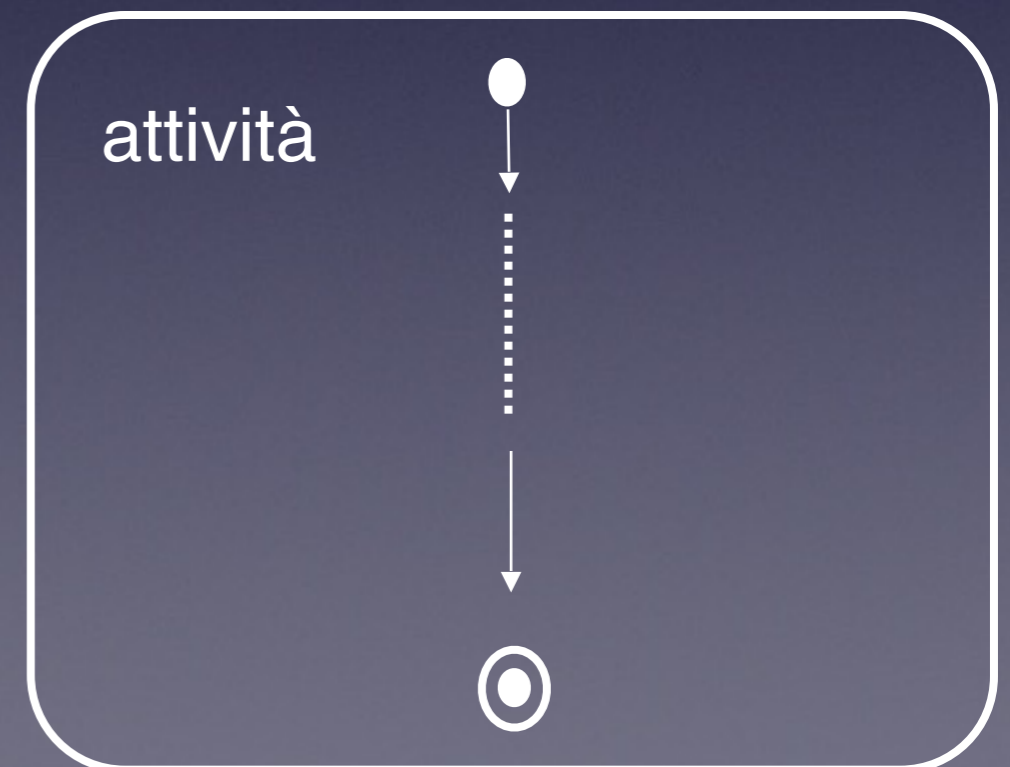
- Le azioni possono avere input e output: i dati sono scambiati tramite oggetti (nodi oggetto)



- Gli oggetti scambiati
  - possono avere uno stato (ad esempio [paid])
  - hanno il nome del classificatore (maiuscolo e non sottolineato)

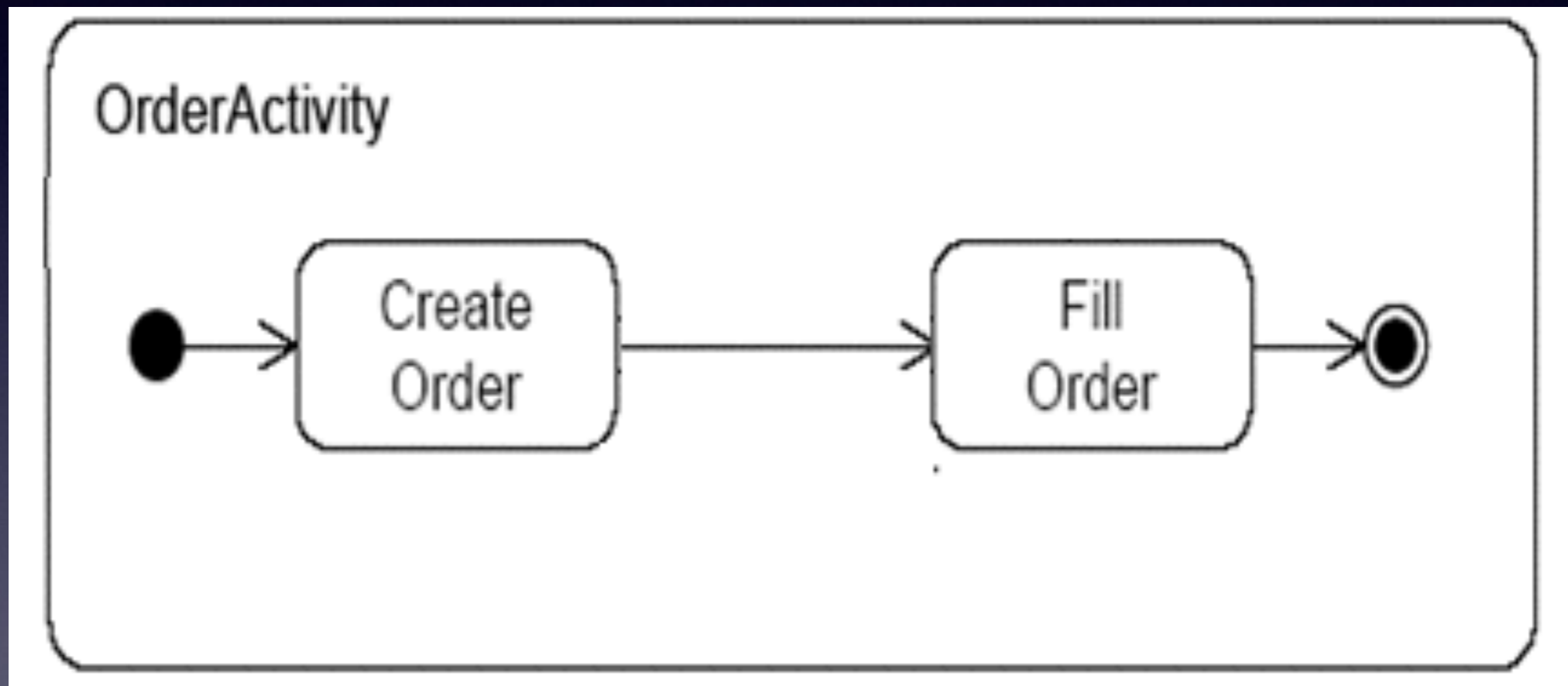
# sotto-attività

- Una azione ne può includere (chiamare) un'altra (che è secondaria)
- La sintassi grafica è il “rastrello” (rake)
  - elemento posto in basso a destra





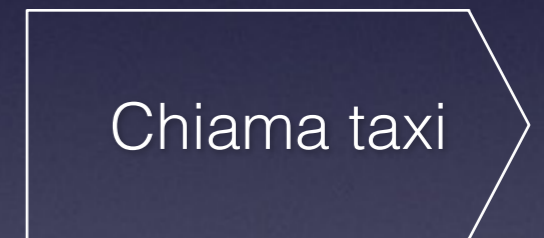
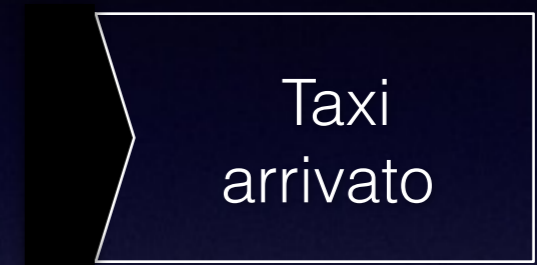
# (sotto-)attività



azione OrderActivity con rastrello in altro diagramma

# azioni legate agli eventi

- Accettazione di evento esterno
- Invio di un segnale
- Accettazione di evento temporale



# azione di accettazione evento

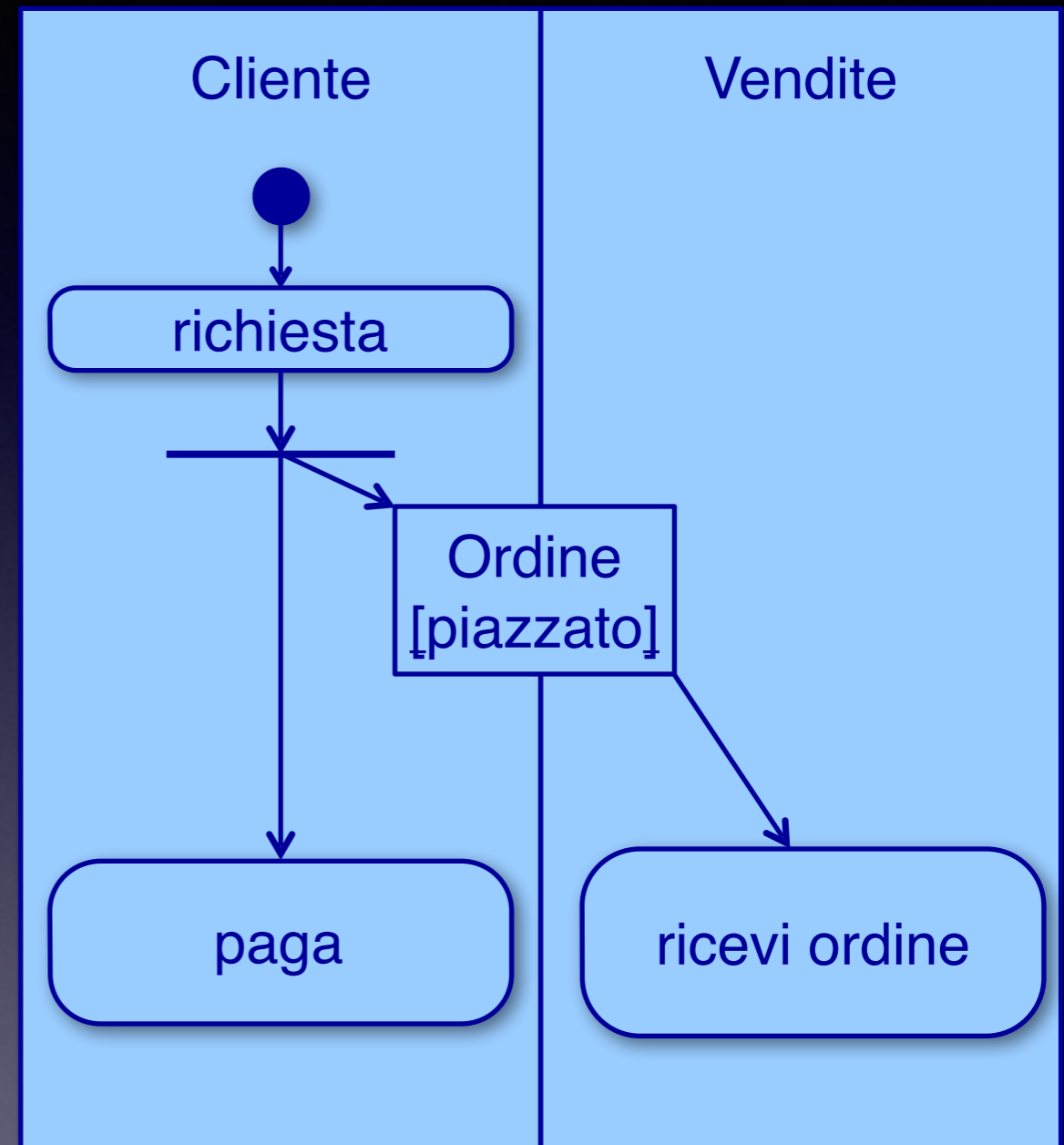
- L'arco entrante non è necessario
- In tal caso, l'azione rimane abilitata e può ricevere altri eventi
- Stessa cosa vale per la clessidra

## **Semantics**

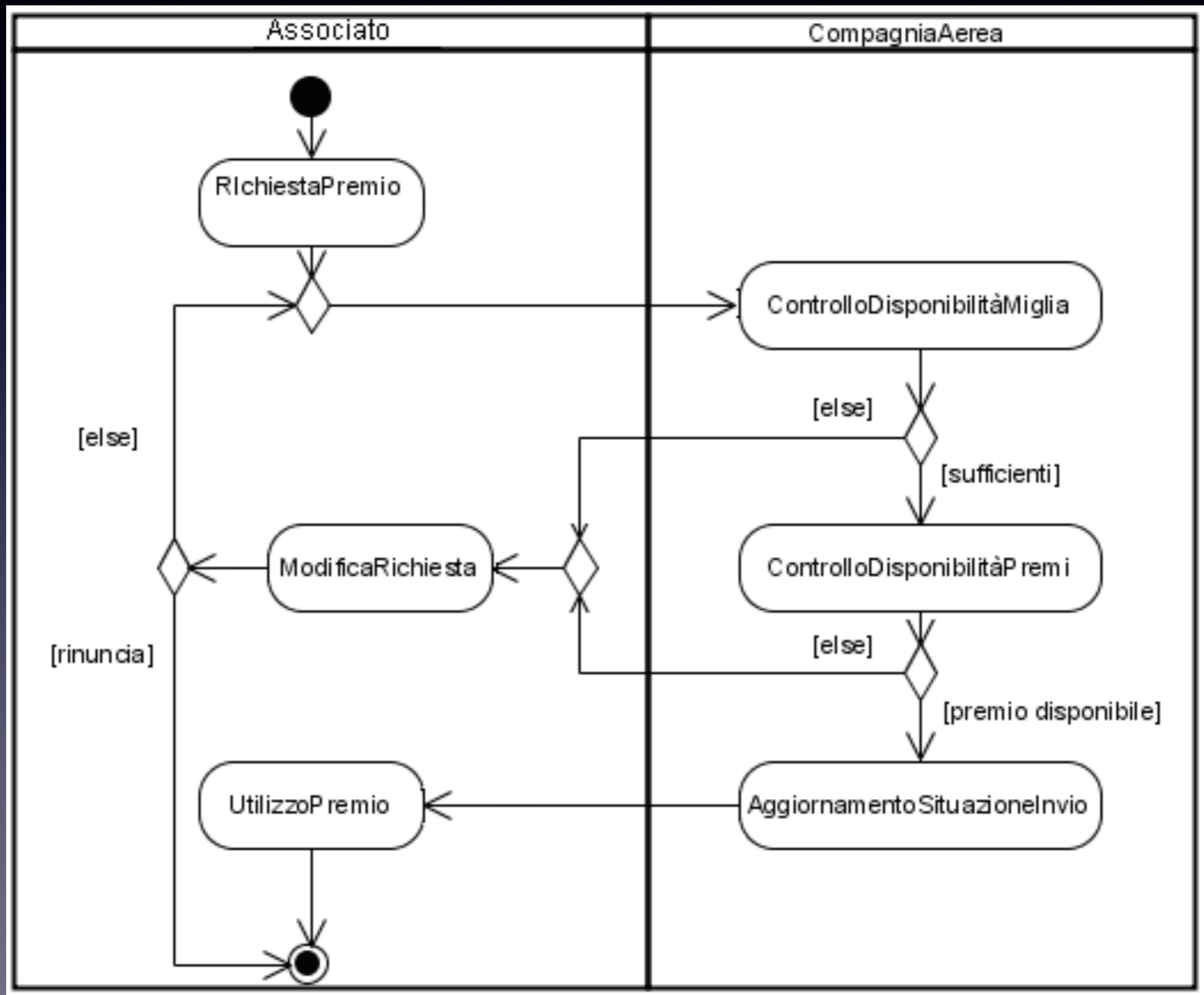
If an `AcceptEventAction` has no incoming edges, then the action starts when the containing activity or structured node does, whichever most immediately contains the action. In addition, an `AcceptEventAction` with no incoming edges remains enabled after it accepts an event. It does not terminate after accepting an event and outputting a value, but continues to wait for other events. This semantic is an exception to the normal execution rules in Activities. An `AcceptEventAction` with no incoming edges and contained by a structured node is terminated when its container is terminated.

# partizioni

- Una *partizione*
  - divide le azioni in gruppi, e
  - spesso corrisponde alla divisione in unità operative in un modello di business
- Permette di
  - descrivere le partizioni, e
  - assegnare le responsabilità

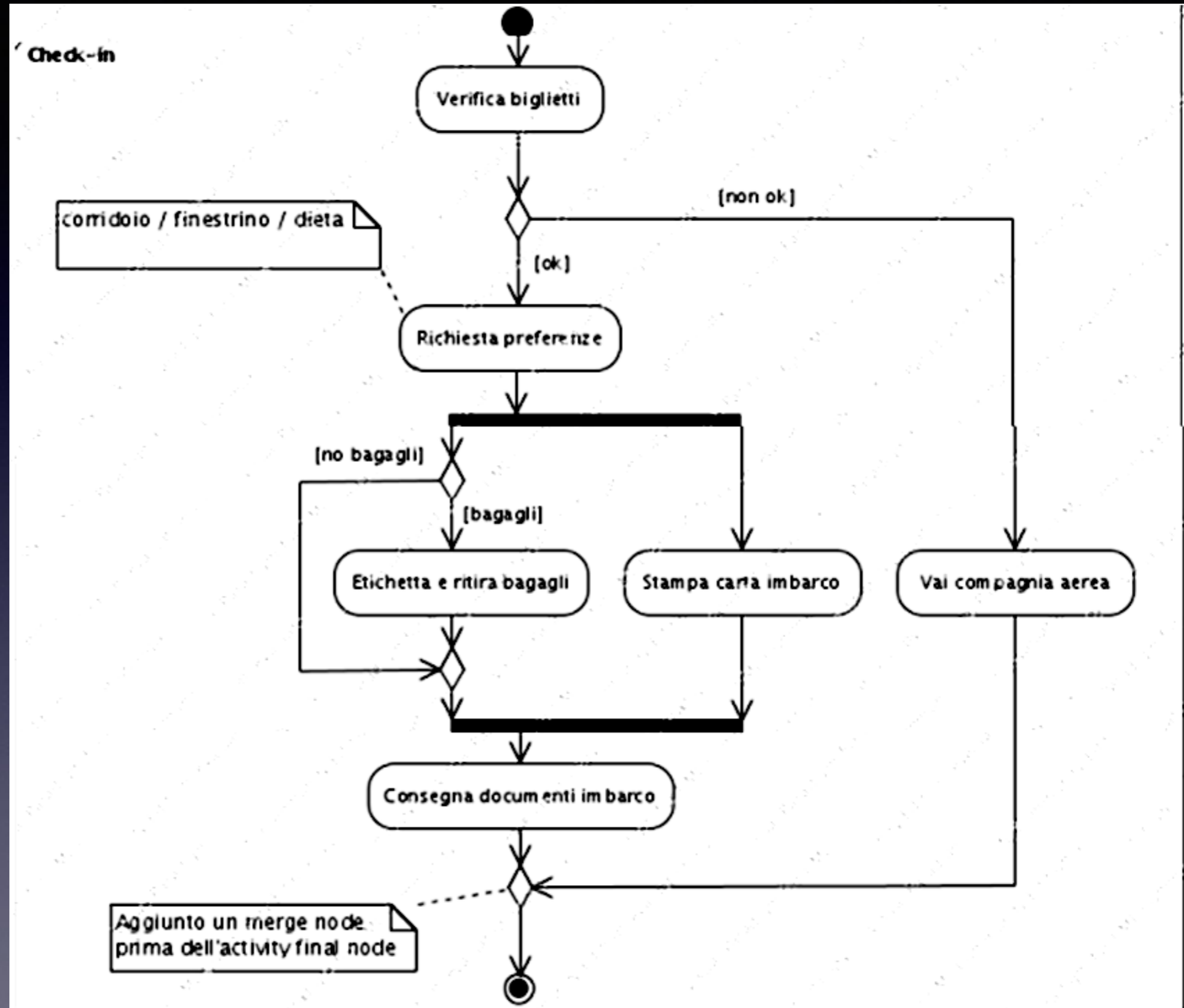


# esempio



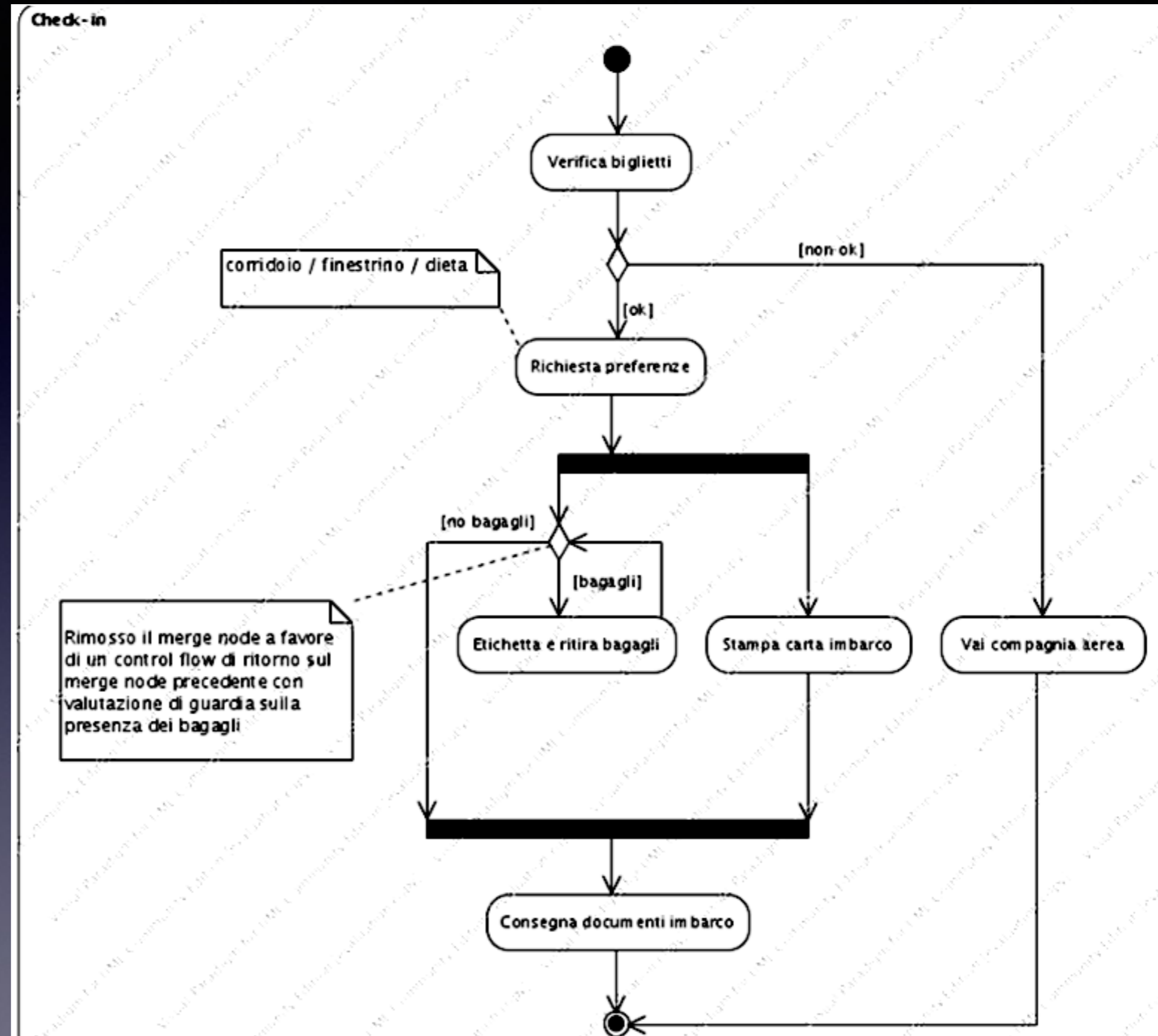
# esercizi di stile: check-in

1



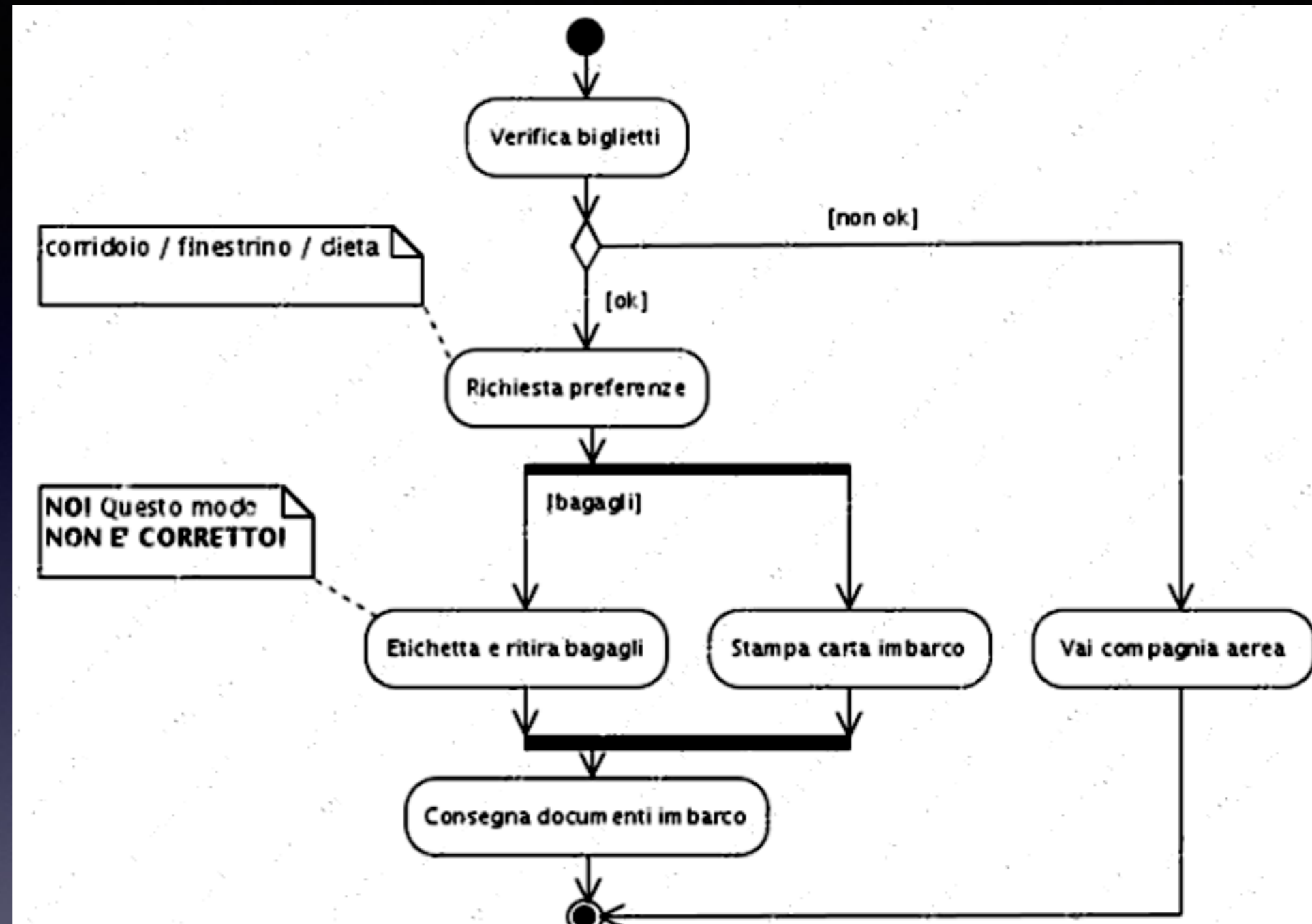
# esercizi di stile: check-in

## 2



[errato anche in Arlow]

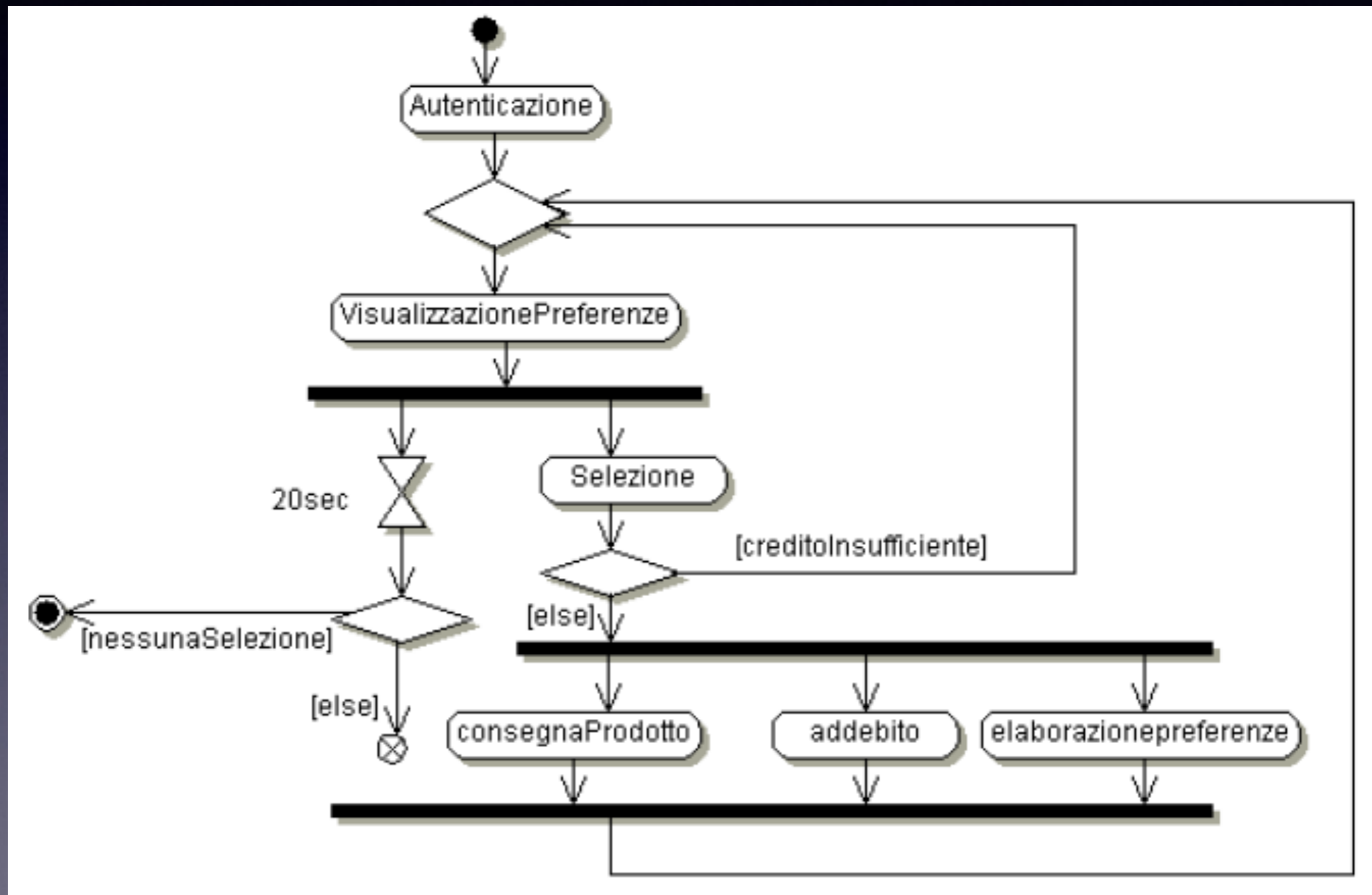
# esercizi di stile: check-in 3



- Gli output possono avere guardie, anche se aumenta il rischio di modelli non ben definiti [def.]
- Gli archi la cui guardia fallisce non ricevono il token. Si deve far sì che la join non dipenda dall'arrivo dei token provenienti da archi guardati (UML infrastructure Specification, p. 363)



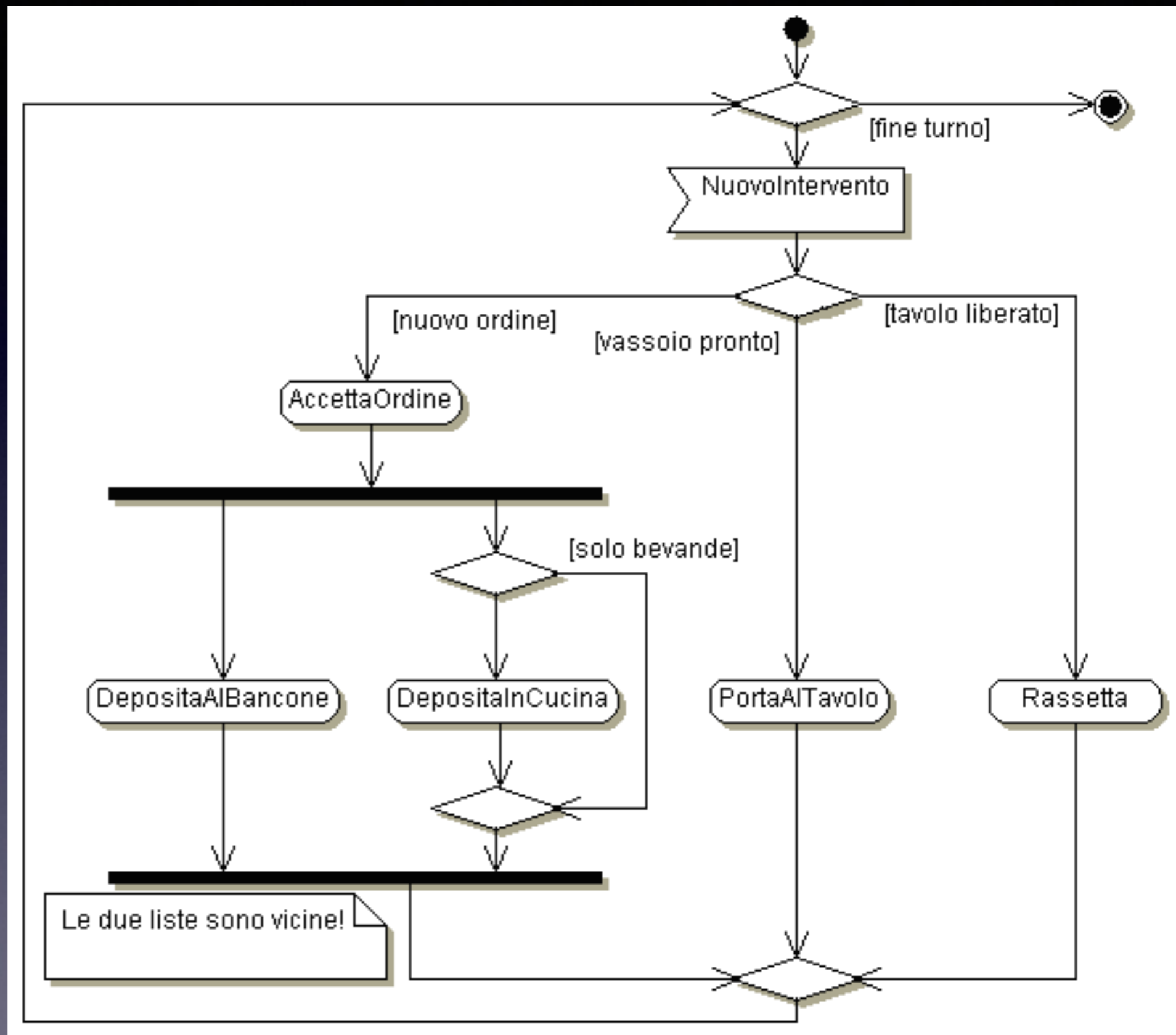
# un esempio di time-out



# esercizio: MyPub

- Il proprietario di un pub vuole introdurre un sistema per la raccolta e la gestione delle ordinazioni dei clienti, per migliorare il servizio diminuendo i tempi di attesa e semplificare le interazioni tra i dipendenti.
- Il pub è situato nella zona centrale di una città di medie dimensioni. D'inverno il pub ha 25 tavoli nei locali interni, mentre d'estate può utilizzare la piazza antistante per altri 15 tavoli. I clienti sono serviti ai tavoli o al bancone. Il servizio ai tavoli prevede la raccolta delle ordinazioni da parte dei camerieri e la successiva consegna delle bevande e degli snack ordinati. Di norma l'ordinazione di un tavolo è raccolta una volta sola anche se, sempre più spesso, i clienti richiamano il cameriere per ordinare ulteriori bevande e snack. Il cameriere accetta solo ordinazioni che fanno riferimento a bevande e snack presenti sul menu giornaliero.
- Raccolta l'ordinazione, il cameriere la consegna al bar e in cucina. Uno dei baristi prende un'ordinazione dalla lista di quelle da servire (rispettando l'ordine temporale di consegna), prepara le bevande indicate su un vassoio che appoggia sul bancone, a disposizione del cameriere. Lo stesso accade in cucina per gli snack, preparati da uno dei cuochi. Il cameriere preleva le bevande e gli snack e li consegna ai clienti.
- Prima di lasciare il pub, il cliente passa dalla cassa e comunica al cassiere il numero del suo tavolo, ottenendo il conto che può pagare sia in contanti sia mediante carta di credito o bancomat. Il conto è calcolato utilizzando i prezzi indicati nel menu. Il cassiere rilascia uno scontrino fiscale, a prova dell'avvenuto pagamento. In seguito a un'ordinanza del sindaco, nel periodo estivo non è permesso servire bevande alcoliche dopo le 22, fino alla chiusura.

# soluzioni one 1



# soluzione 2

