9. Diagramma delle attività

Paola Barra
a.a. 2023/2024

Diagrammi UML

- **Diagrammi dei casi d'uso**
  - Descrivono il comportamento funzionale del sistema come sono visti dagli utenti
- **Diagrammi delle classi**
  - Descrivono la struttura statica del sistema: oggetti, attributi, associazioni
- **Diagrammi delle sequenze**
  - Descrivono il comportamento dinamico tra gli oggetti del sistema
- **Diagrammi degli stati**
  - Descrivono il comportamento dinamico di un singolo oggetto
- **Diagrammi delle attività**
  - Descrivono il comportamento dinamico di un sistema, in particolare il flusso di lavoro

Diagrammi delle attività

Diagramma delle attività

il diagramma delle attività è sostanzialmente un estensione di un diagramma di flusso che modella la transizione da un'attività all'altra e mostra come il sistema è coordinato per fornire il servizio a diversi livelli di astrazione.

Diagrammi delle attività

- Forniscono la sequenza di operazioni che definiscono un'attività più complessa
- Permettono di rappresentare processi paralleli e la loro sincronizzazione
- Possono essere considerati Diagrammi di stato particolari
  - Ogni stato contiene (è) un'azione
- Un diagramma delle attività può essere associato
  - A una classe
  - All'implementazione di un'operazione
  - Ad un caso d'uso

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)

L'attività

- Un'attività ha un nome ed è contenuta in un rettangolo con gli angoli smussati
- Il contenuto di un'attività è un grafo diretto i cui:
  - i nodi rappresentano le componenti dell'attività, come le azioni o i nodi di controllo (inizio, fine, etc)
  - gli archi rappresentano il control flow: i possibili path eseguibili per l'attivita'.

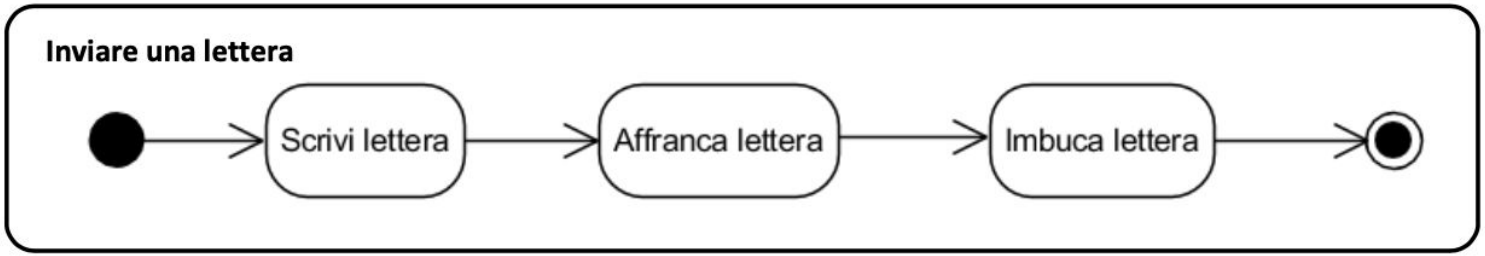
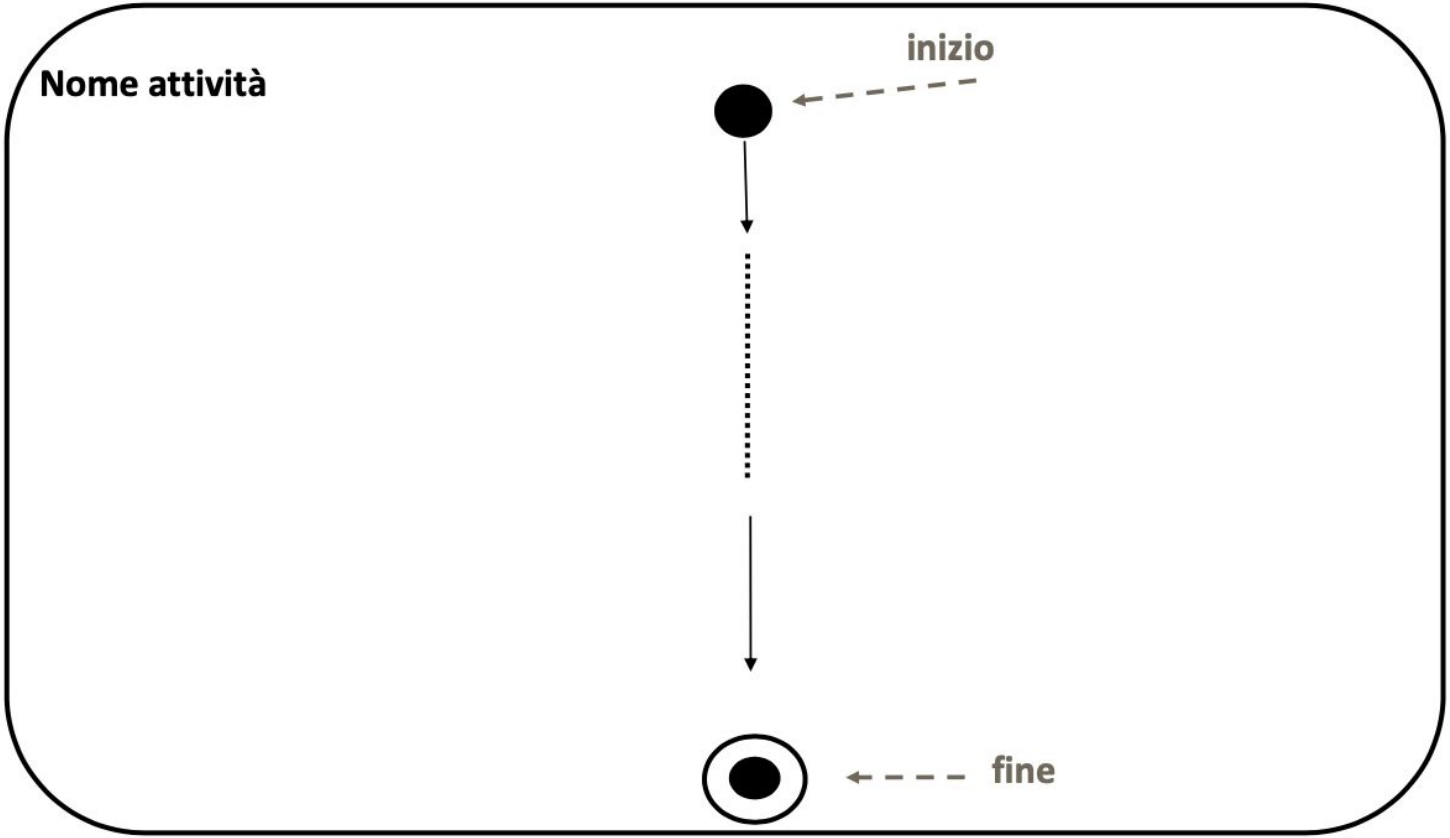


Diagramma delle attività

- Ogni diagramma delle attività ha due nodi particolari: *Inizio* e *Fine*
  - **Inizio** è il punto di partenza del diagramma, indica la prima azione da eseguire ed è rappresentato da un cerchio con solo archi in uscita
  - **Fine**: indica la conclusione dello scenario descritto e ha solo archi in entrata. L'azione finale del diagramma deve puntare sempre al nodo finale

Diagrammi di attività: inizio e fine

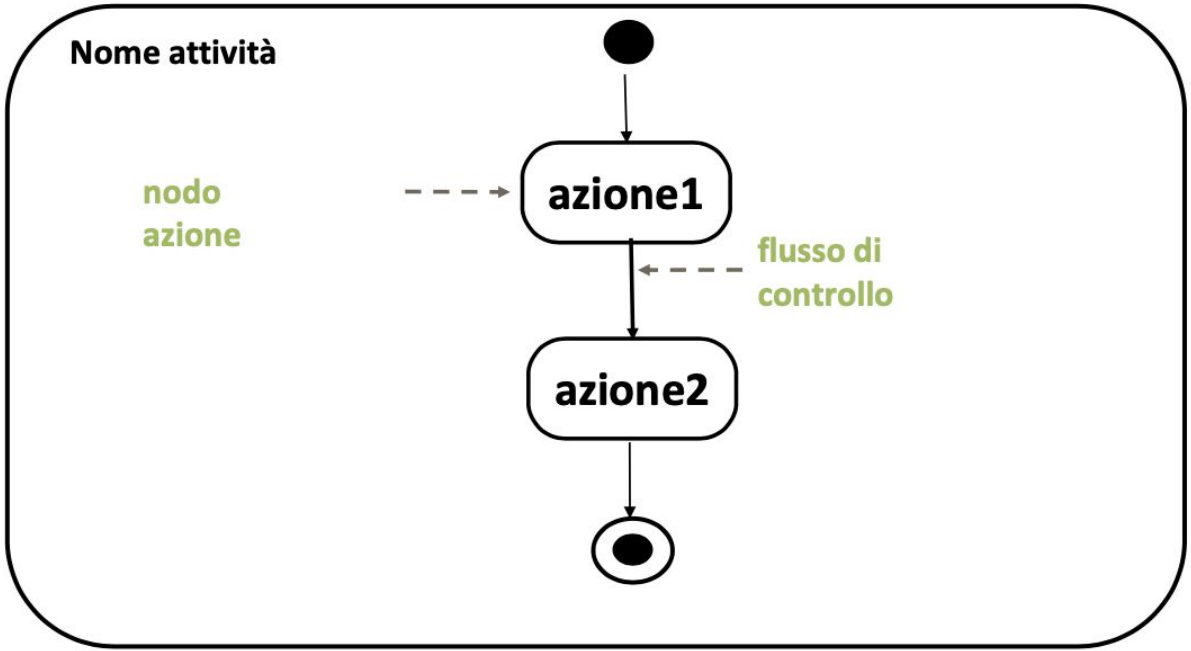


Le azioni

- Le azioni sono rappresentate anche esse da rettangoli con angoli smussati

- Possono essere specificate in linguaggio naturale
  - il nome di un'azione deve descrivere un'azione, quindi tipicamente essere un verbo
- Sono atomiche
- Oltre ai nodi azioni atomici, esistono dei nodi azione con comportamento non atomico, il cui dettaglio è specificato da diagramma di (sotto)attivita' (li vedremo tra qualche lucido)

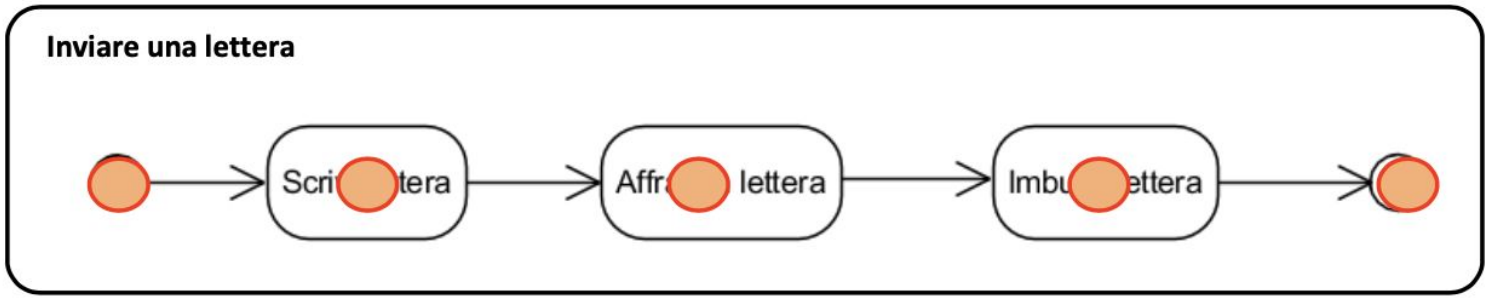
Diagrammi di attività: nodo azione



- Solo una freccia entrante e una uscente per ogni azione (vedremo perché)
- la freccia di uscita è presa appena è terminata l'azione

Transizioni

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



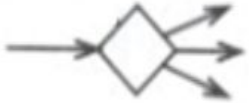
- La semantica è descritta con il token game: l'azione può essere eseguita quando riceve il token

Nodi di controllo

nodo iniziale

nodo finale

nodo di fine flusso



nodo decisione (con guardie sulle frecce uscenti)

nodo fusione

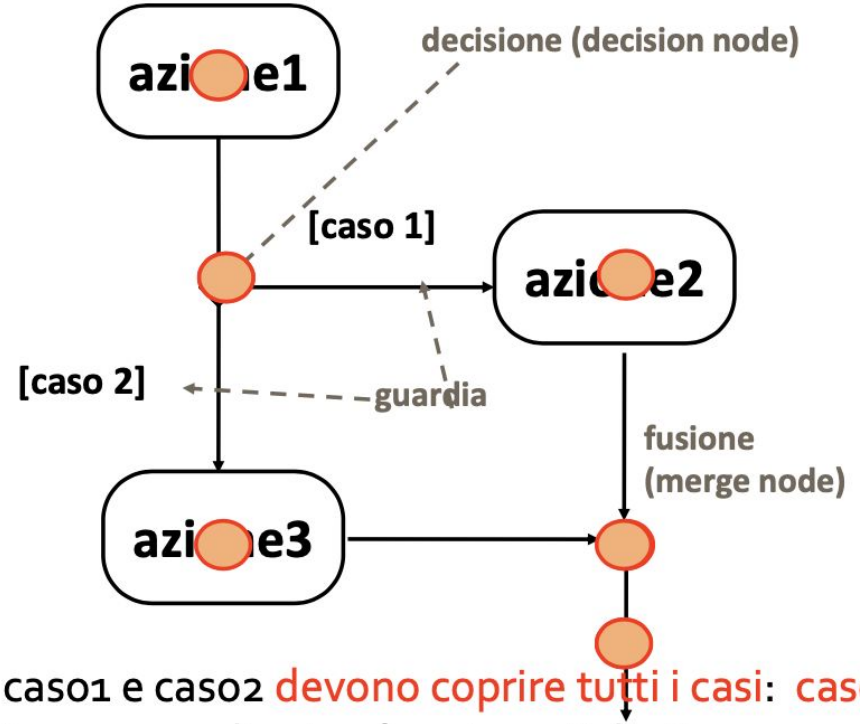
nodo di biforcazione (fork)

nodo di sincronizzazione (join)

Diagramma delle attività: scelta

- Abbiamo detto che ogni azione si attiva appena riceve un token, si esegue e poi passa il token sull'arco uscente.
- Questo meccanismo di passaggio del token viene alterato da una choice

Diagrammi di attività: scelta



- caso1 e caso2 devono coprire tutti i casi: caso 1 OR caso2=True
- In una guardia si può scrivere "else"

Esempio : classe Incident

- Un'azione è un'attività atomica: è rappresentata nel diagramma con un rettangolo arrotondato ed è identificata con una voce verbale che descrive l'azione stessa
  - *HandleIncident*: il *Dispatcher* riceve i rapporti e alloca risorse
  - *DocumentIncident*: tutti i *FieldOfficer* partecipanti e i *Dispatcher* documentano l'incidente dopo che è stato chiuso
  - *ArchiveIncident*: archiviazione delle informazioni relative all'incidente su dispositivi di memorizzazione

- Gli archi tra le attività rappresentano il flusso di controllo. Un'attività può essere eseguita solo dopo che tutte le attività precedenti sono state completate

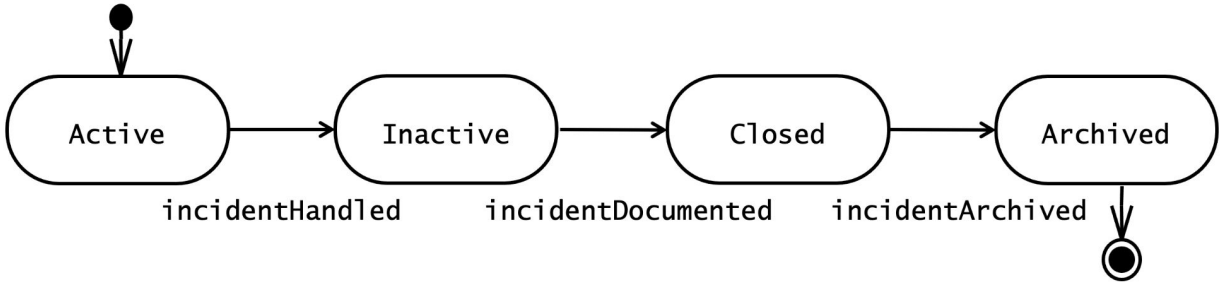


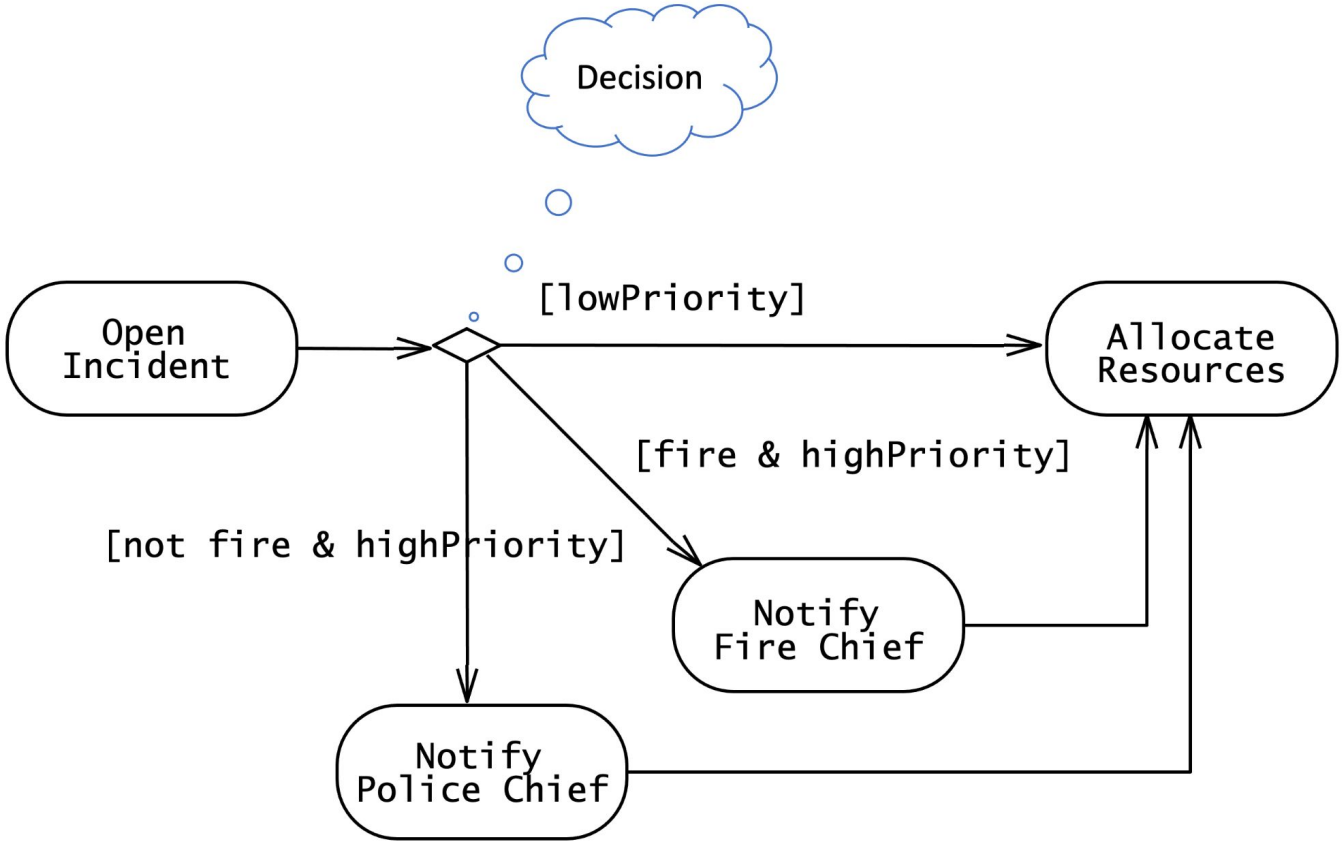
Diagramma delle attività

- Un arco, o flusso, collega tra loro i nodi
  - L'insieme degli archi del diagramma rappresenta il flusso di esecuzione complessivo
  - E' rappresentato con una freccia
- Il flusso delle azioni progredisce solo nel momento in cui l'azione considerata è completata
- Una funzionalità complessa non è costituita da una semplice successione di azioni in sequenza:
  - può presentare azioni da eseguire contemporaneamente o sotto particolari condizioni

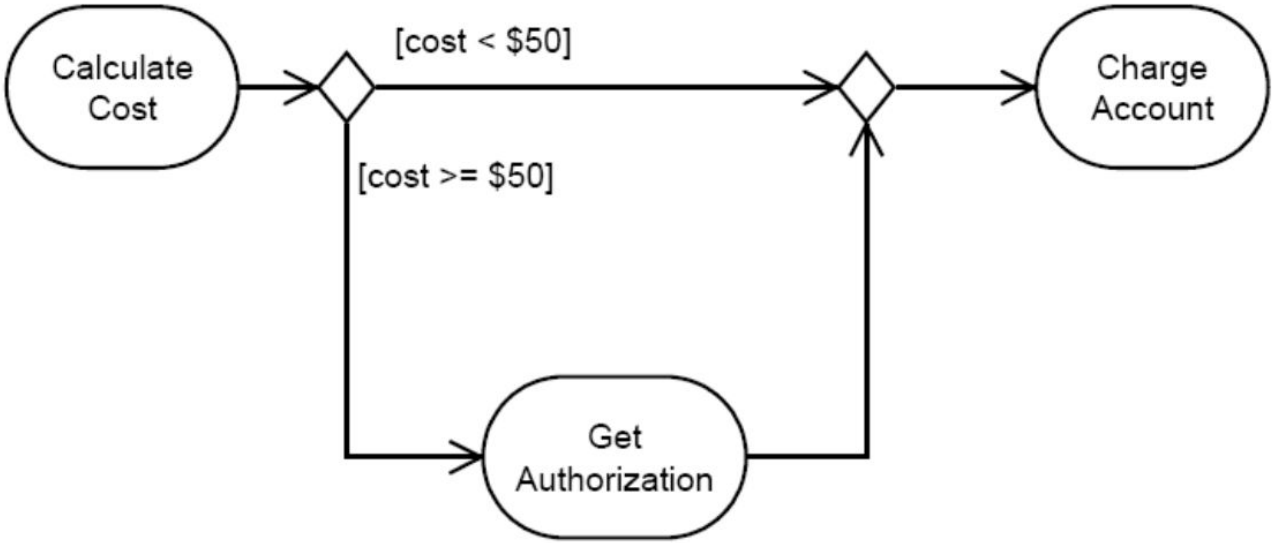
Nodi di controllo

- **Decision**
  - Indica che l'esecuzione di un'azione dipende dal verificarsi di una determinata condizione chiamata **guard**. Descrive il fatto che al termine di una particolare azione, lo scenario può proseguire in modo diverso con azioni che dipendono da specifiche situazioni che si vengono a verificare.
    - E' rappresentato con un diamante puntato da un arco e dal quale escono almeno due archi
- **Merge**
  - Duale del nodo decision e descrive un punto del processo in cui si ricongiungono due o più flussi alternativi
    - Rappresentato anch'esso da un diamante

Esempio di decisioni in OpenIncident



Decisione e fusione – decision and merge

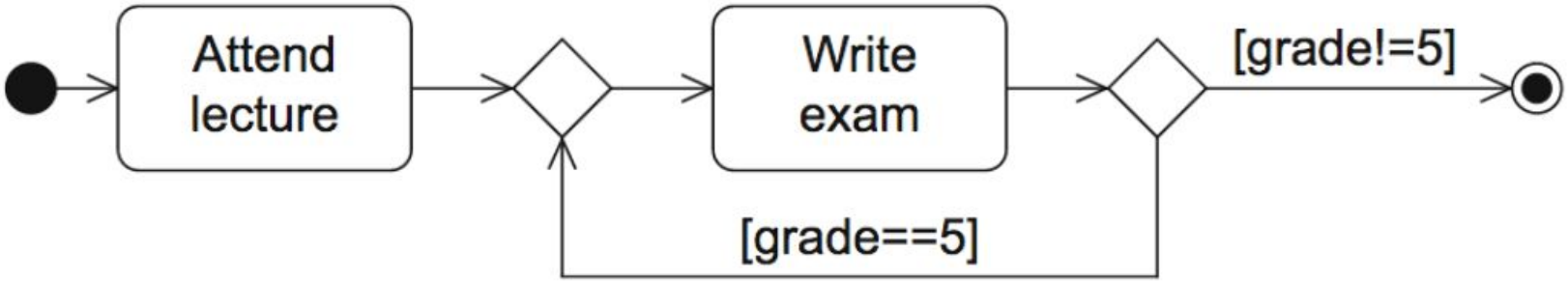


- Il token prende uno dei cammini
- Deve prendere sempre uno dei cammini

Semantica:

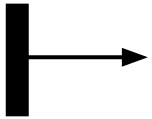
- Le guardie devono coprire tutte le possibilità
  - In caso si usa [else]
- E' bene (ma non necessario) che siano mutualmente esclusive altrimenti comportamento non definito (non deterministico).
- Le condizioni di guardia sempre tra [ ]
  - (in generale in UML)
- Dato un nodo decisione non è obbligatorio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

Loops



Nodi di controllo

- **Fork**
  - Descrive l'esecuzione in parallelo di più azioni: quelle puntate dal nodo fork sono avviate contemporaneamente ed in parallelo
    - rappresentato da una barra nera puntata da un solo arco e dalla quale partono due o più archi verso le azioni da eseguire in parallelo



- **Join**
  - E' il duale del fork. Specifica che un'azione è eseguita solo nel momento in cui le azioni precedenti hanno terminato la propria esecuzione. E' definito anche sincronizzazione perché sincronizza due rami svolti parallelamente, generando un unico flusso di esecuzione.
    - E' rappresentato da una barra nera

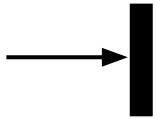


Diagramma di attività: fork e join

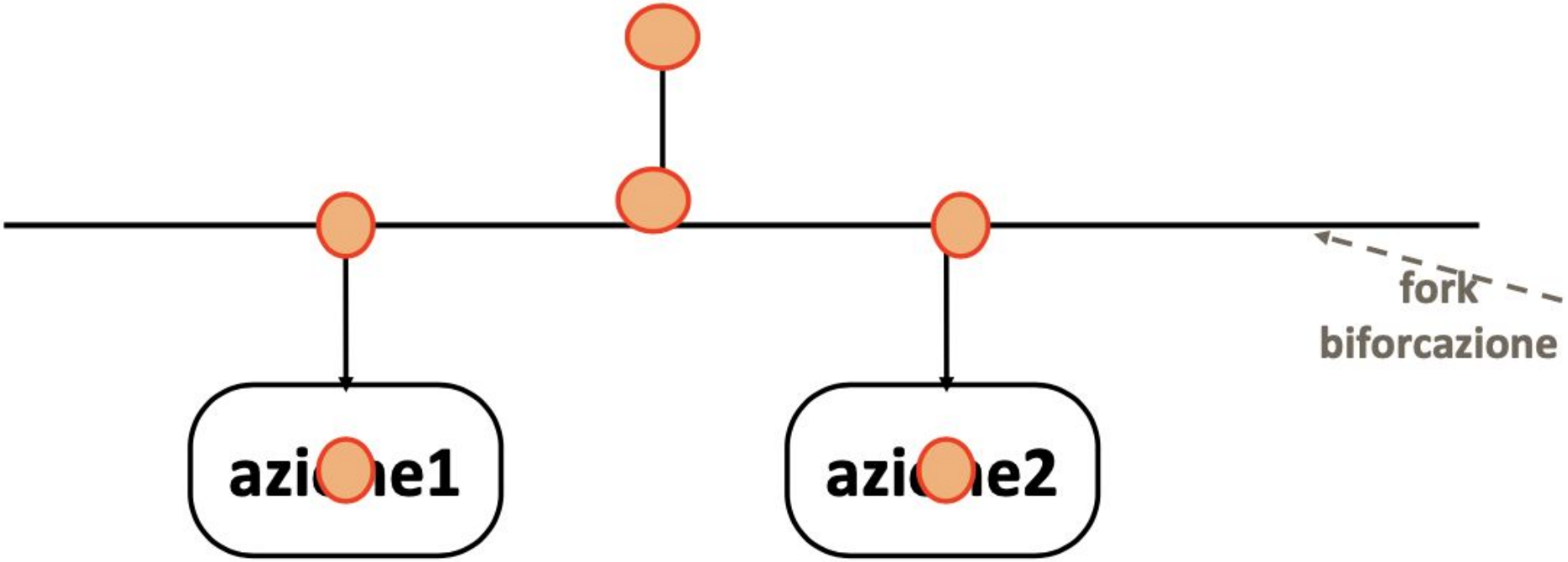
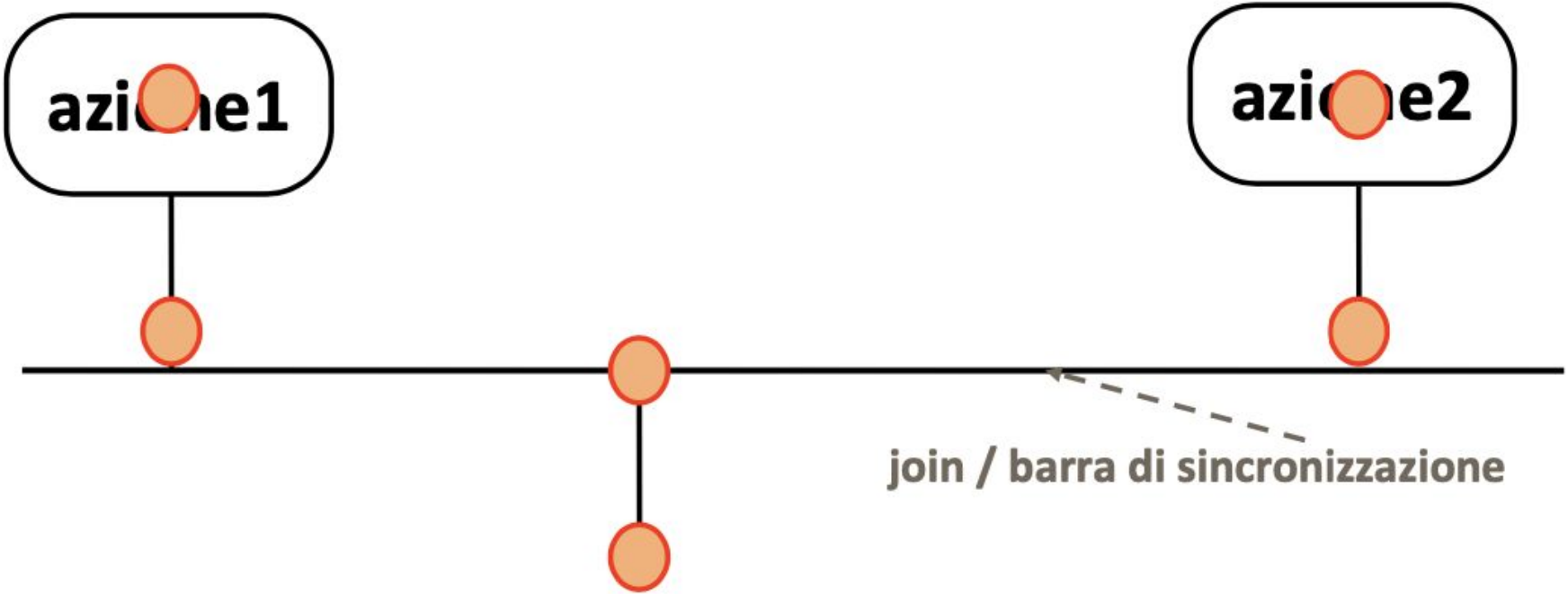


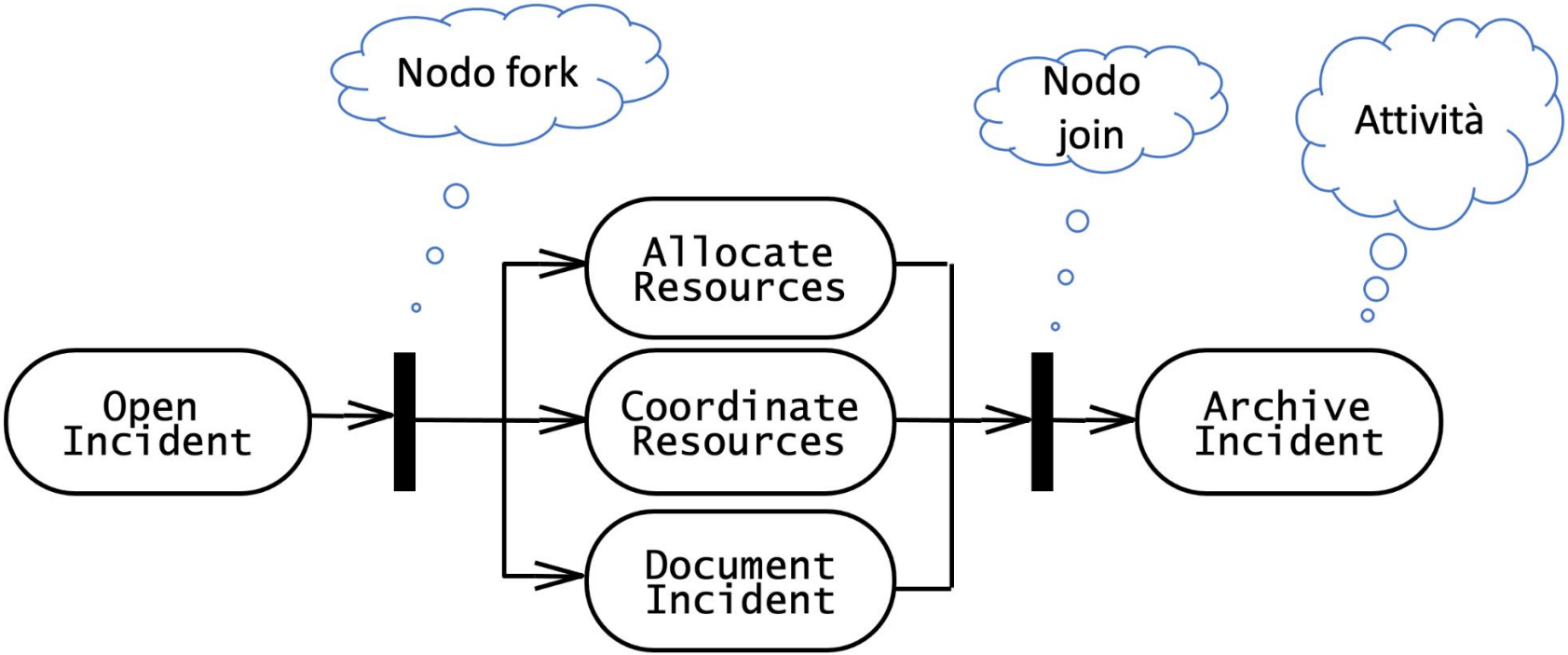
Diagramma di attività: fork e join



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

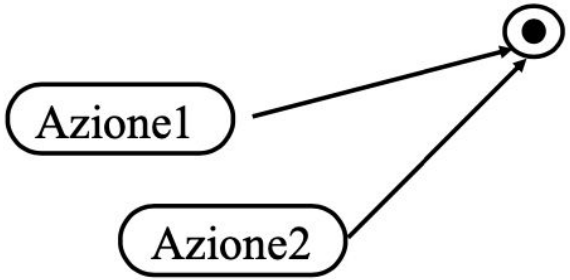
Esempio di fork e join in OpenIncident



Nodo di fine attività

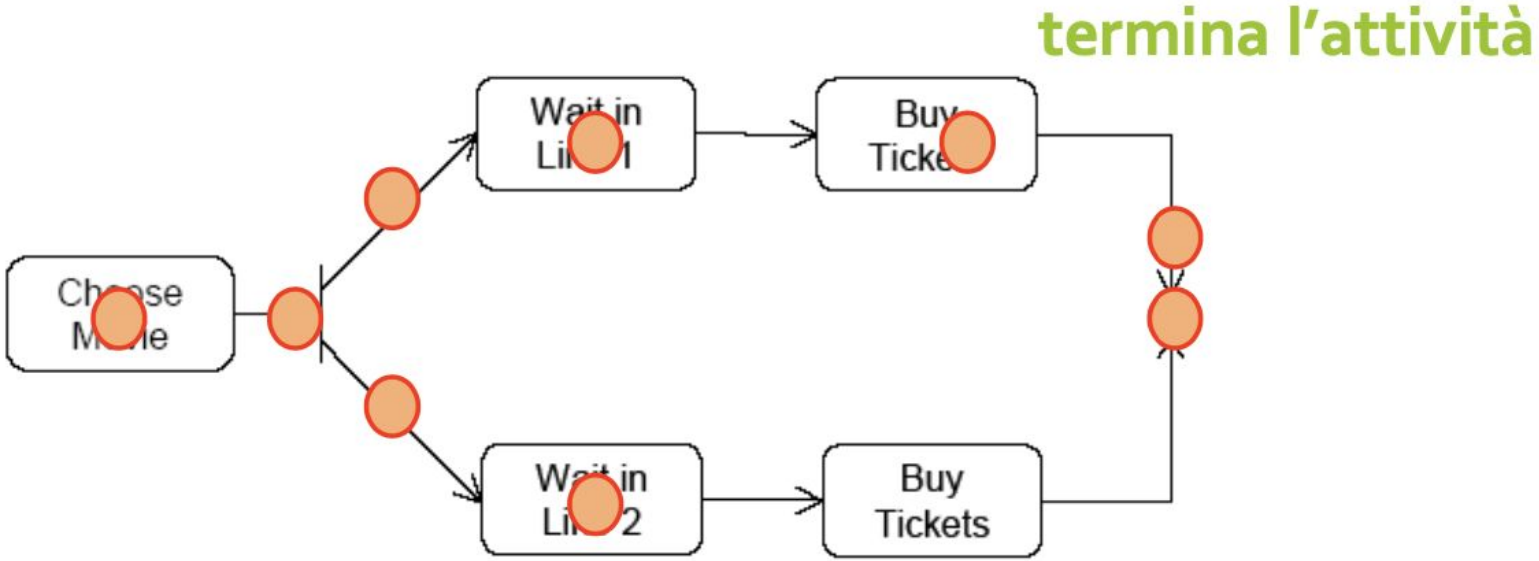
- Se un token raggiunge un nodo di fine attività , l'intera attività è terminata

- Permettiamo più archi entranti su un nodo di fine attività o di fine flusso (e solo su questi)
  - La semantica è: il primo token che arriva termina l'attività



Nodo di fine attività

- il primo che compra i biglietti termina l'attività

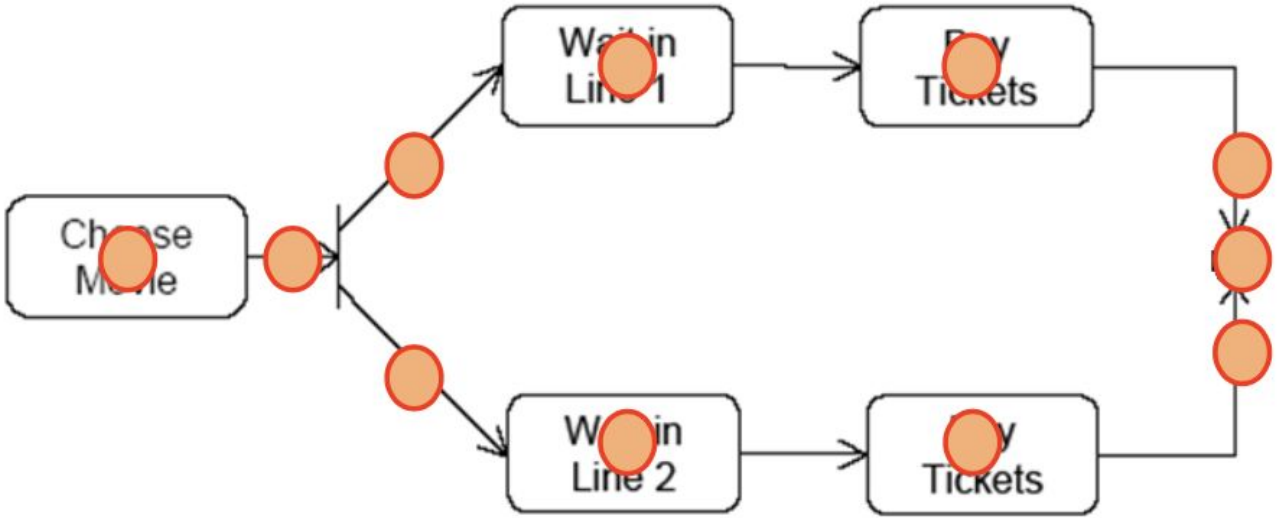


Nodo di fine flusso

- Serve per terminare un execution path non tutta l' attività .

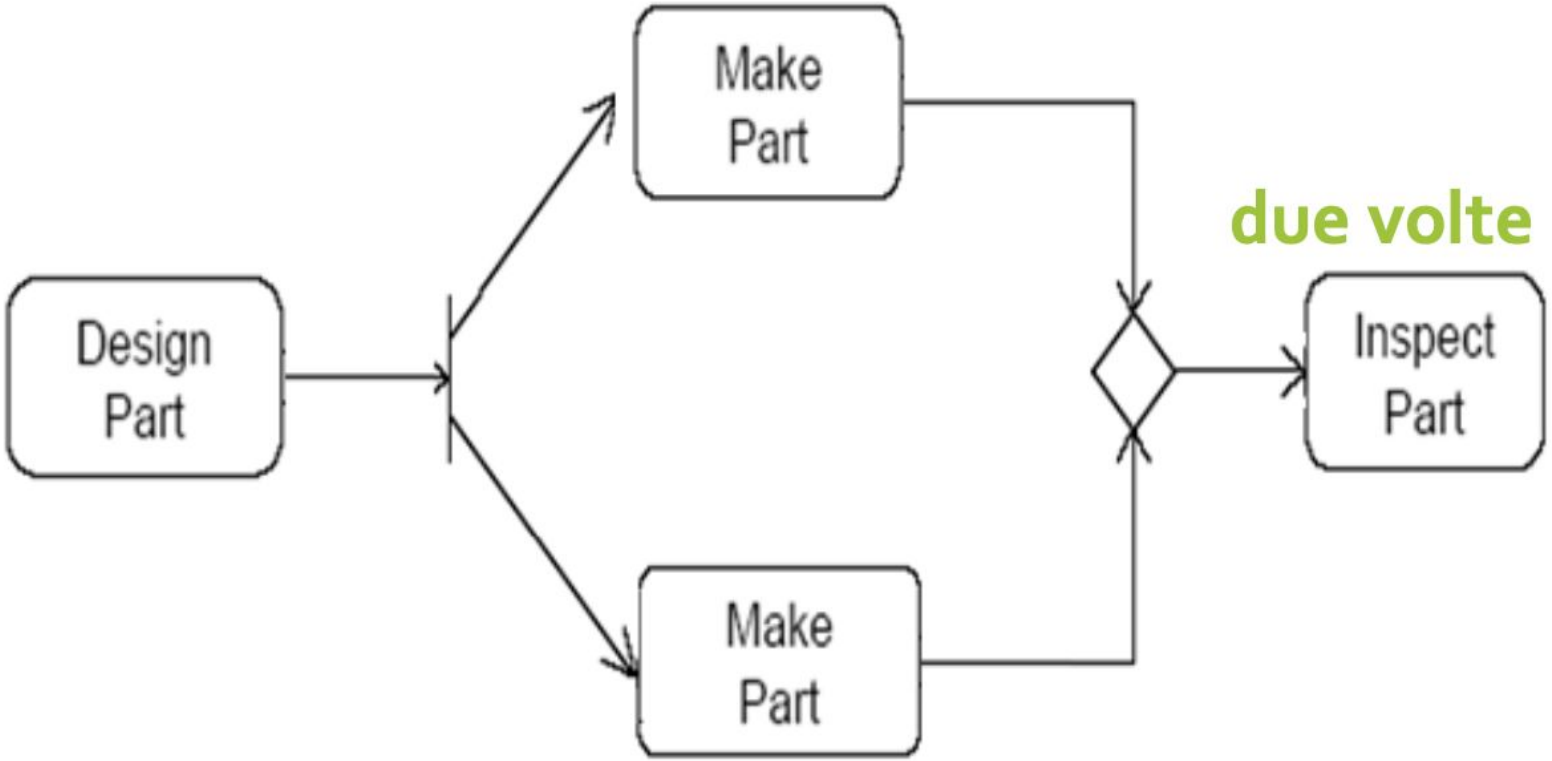
Nodo di fine flusso

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

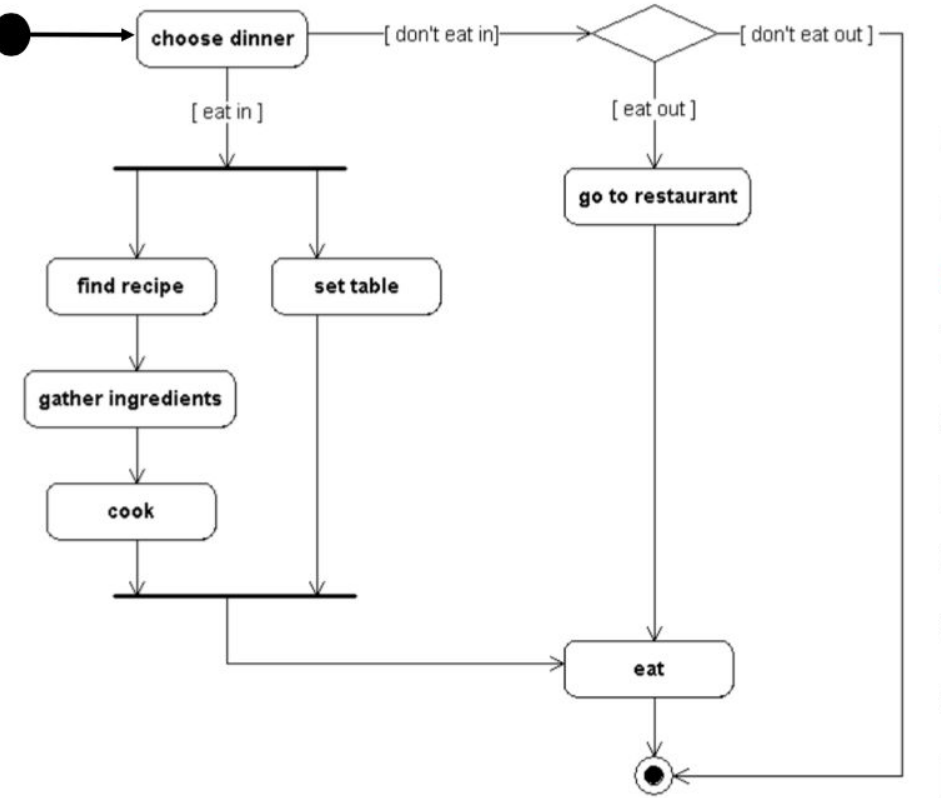


non termina l'attività, solamente si consuma il token

Fork e merge: possibile ma azioni eseguite due volte



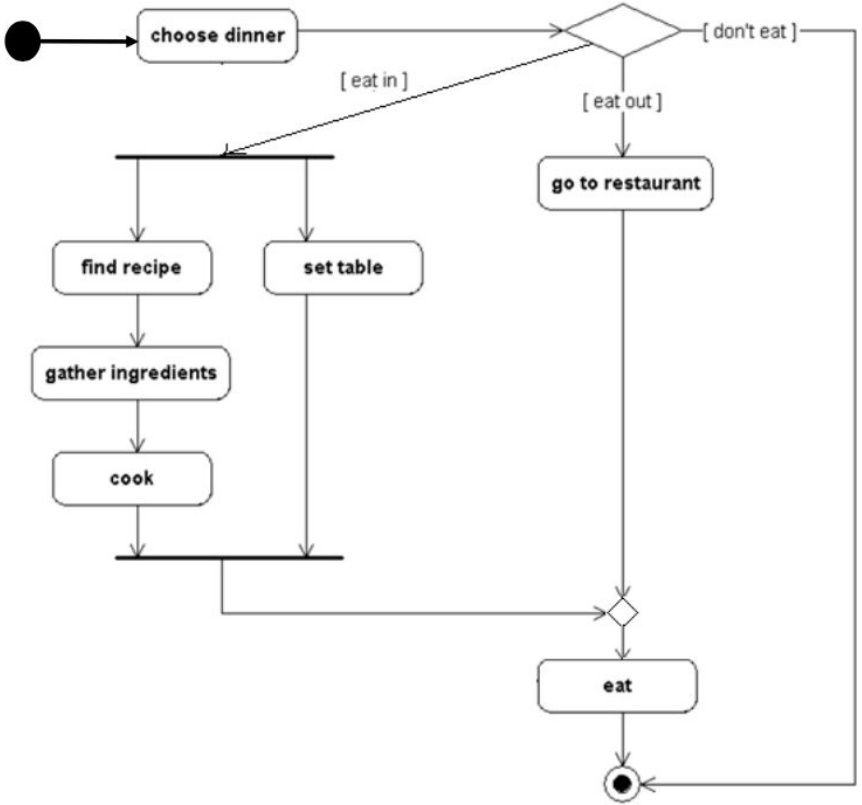
Esempio preso dal web, interessante perchè sbagliato



Anche se UML permette frecce multiple entranti/uscenti in/da un nodo, se ne sconsiglia 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 nodo fusione e dopo choose dinner serve un nodo decisione.

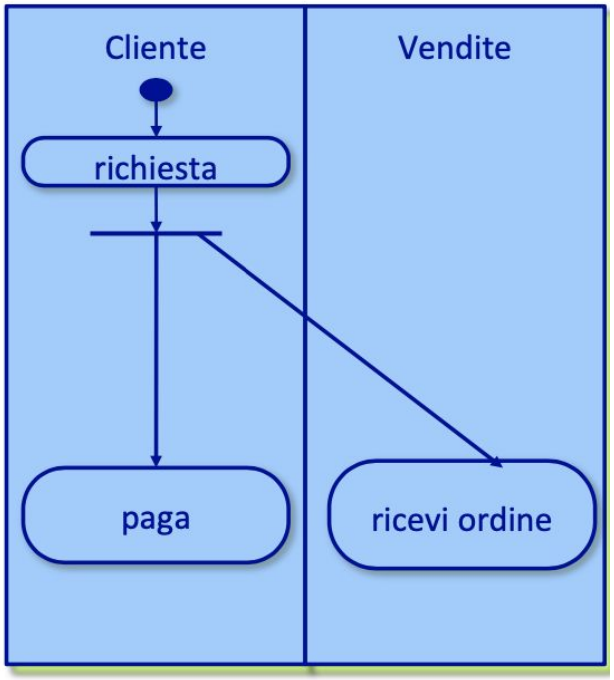
Sono tollerate due frecce entranti nello stato finale.

Swimlane o partizione

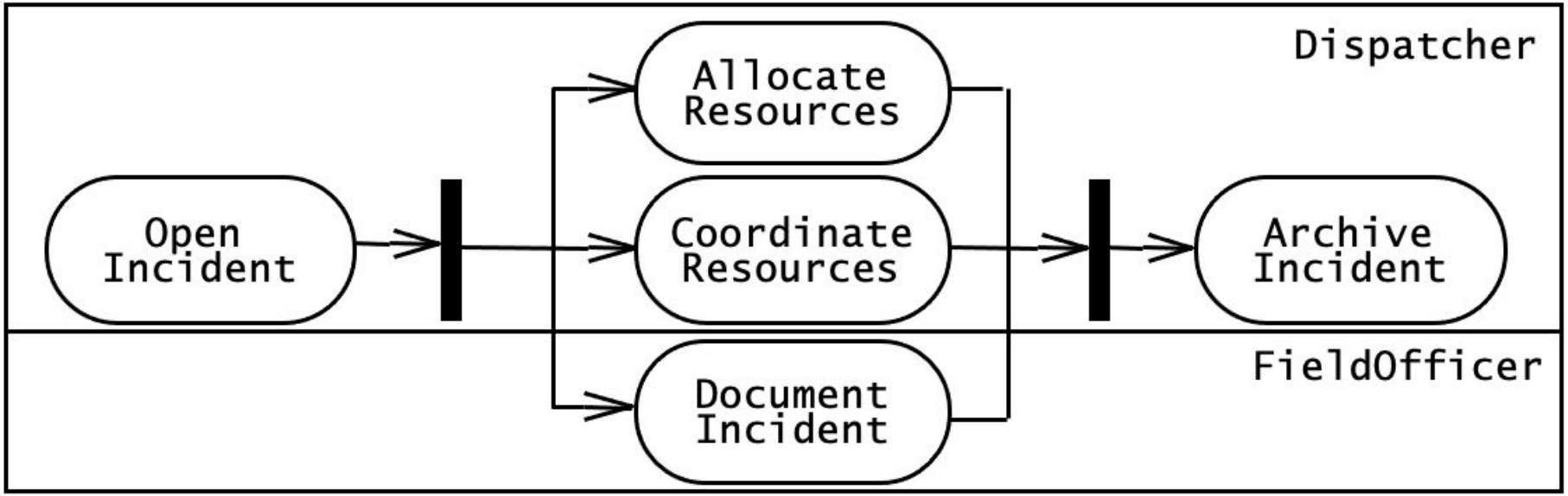
- Le attività possono essere raggruppate in ***swimlane*** per denotare l'oggetto o sottosistema che le implementa
  - Le swimlane sono rappresentate come rettangoli che racchiudono un gruppo di attività
  - Le transizioni possono attraversare le swimlane

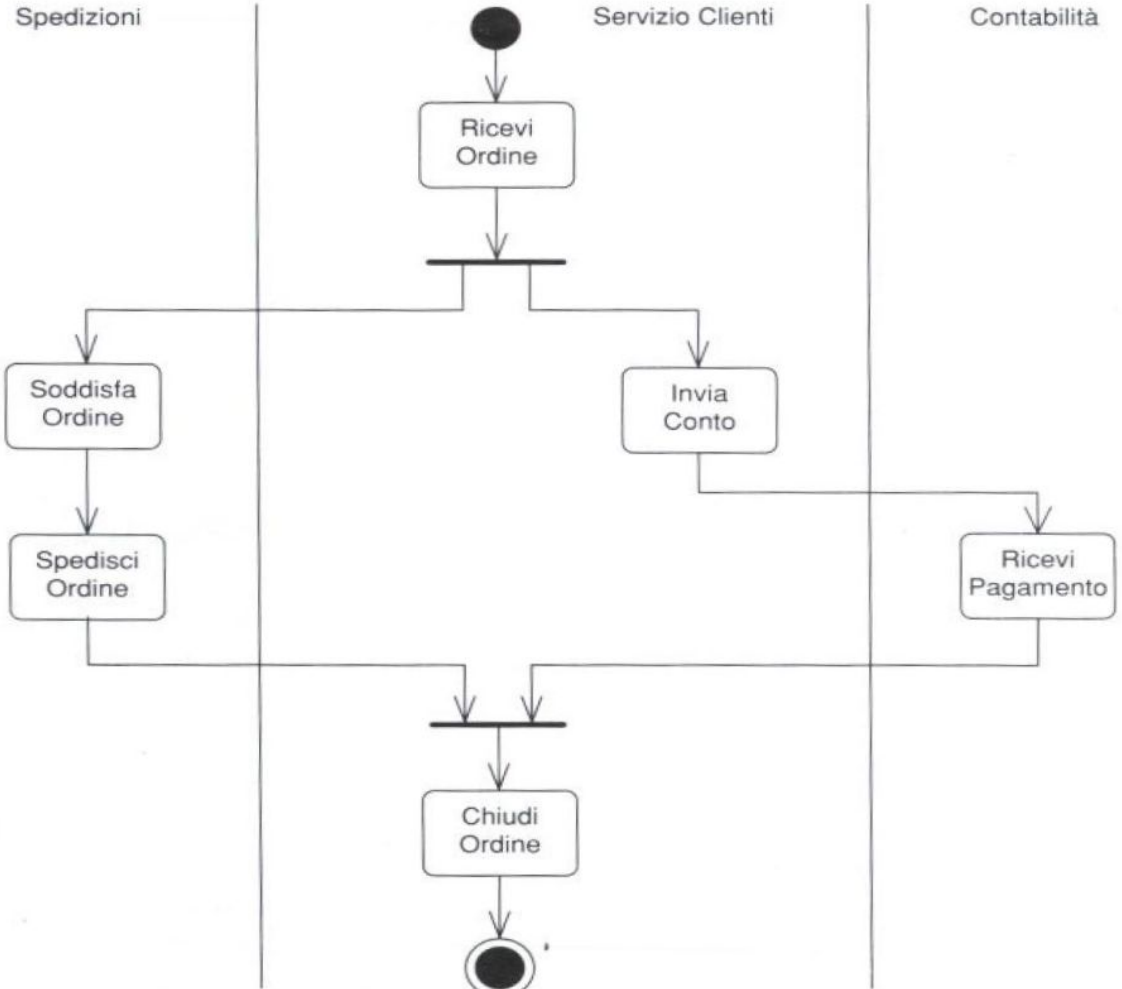
Una partizione (o swimlane) divide le azioni in gruppi.

Permette di assegnare le responsabilità delle azioni.



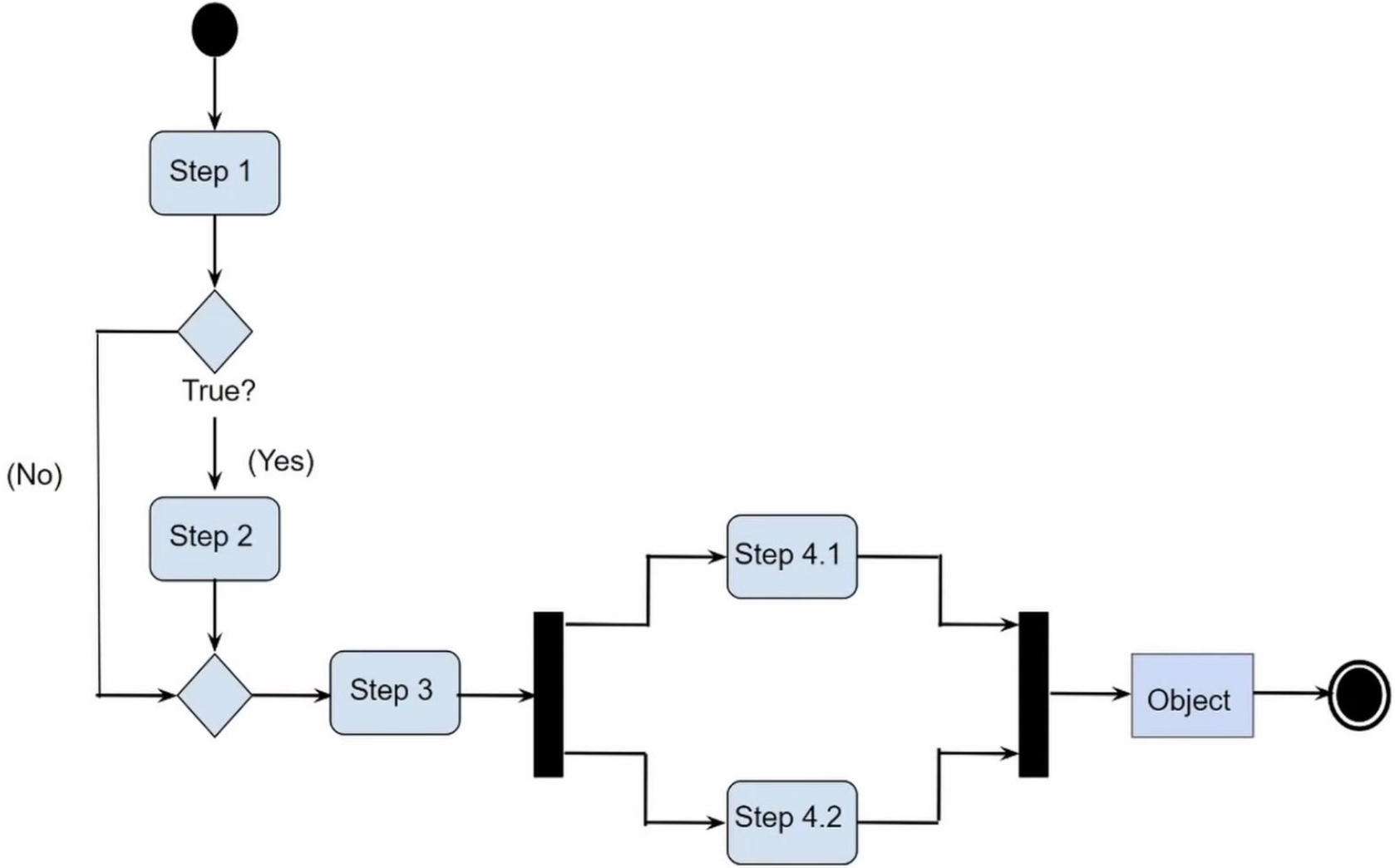
Esempio di swimlane

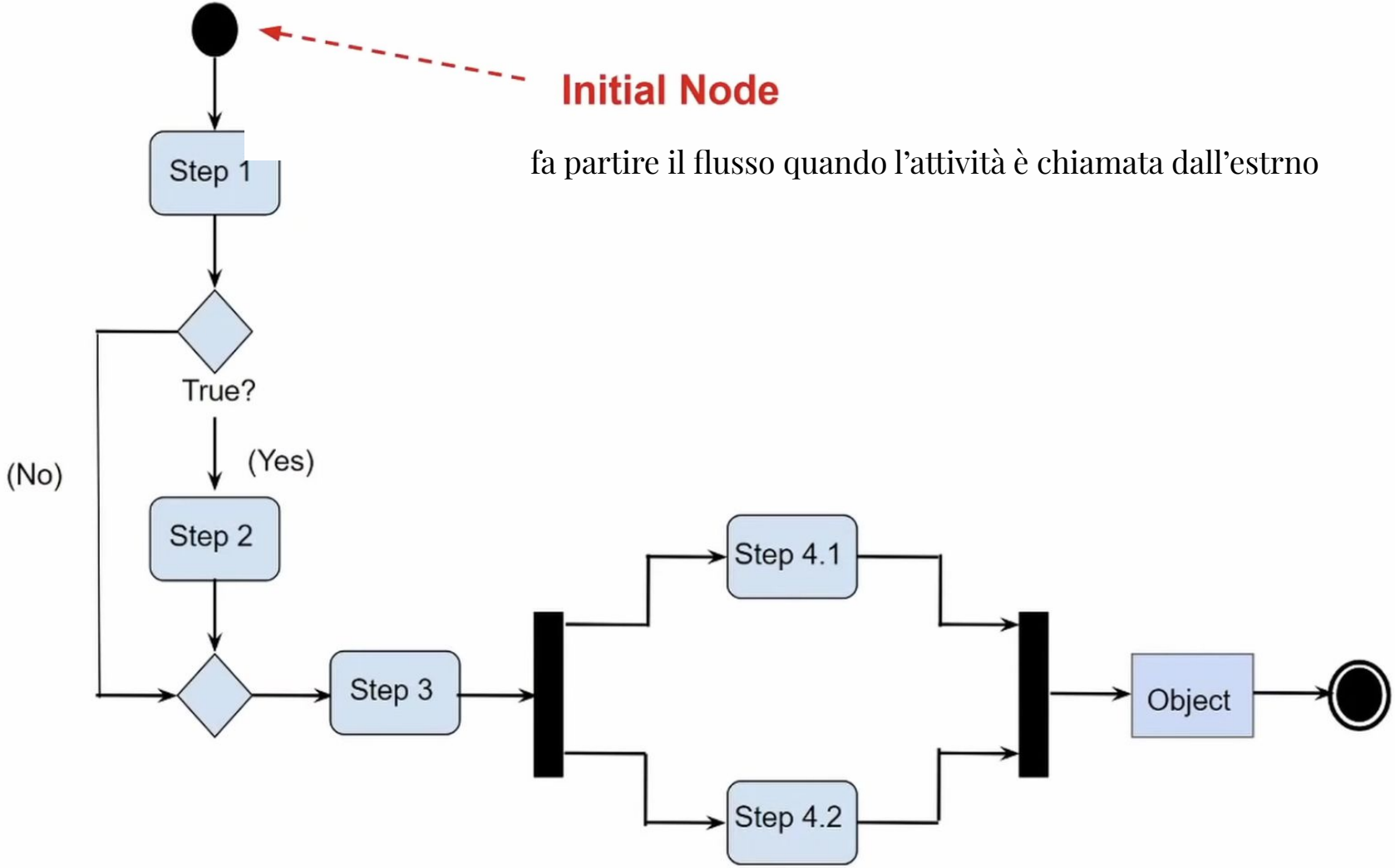



Spedizioni
Servizio Clienti
Contabilità
Ricevi Ordine
Soddisfa Ordine
Invia Conto
Spedisci Ordine
Ricevi Pagamento
Chiudi Ordine

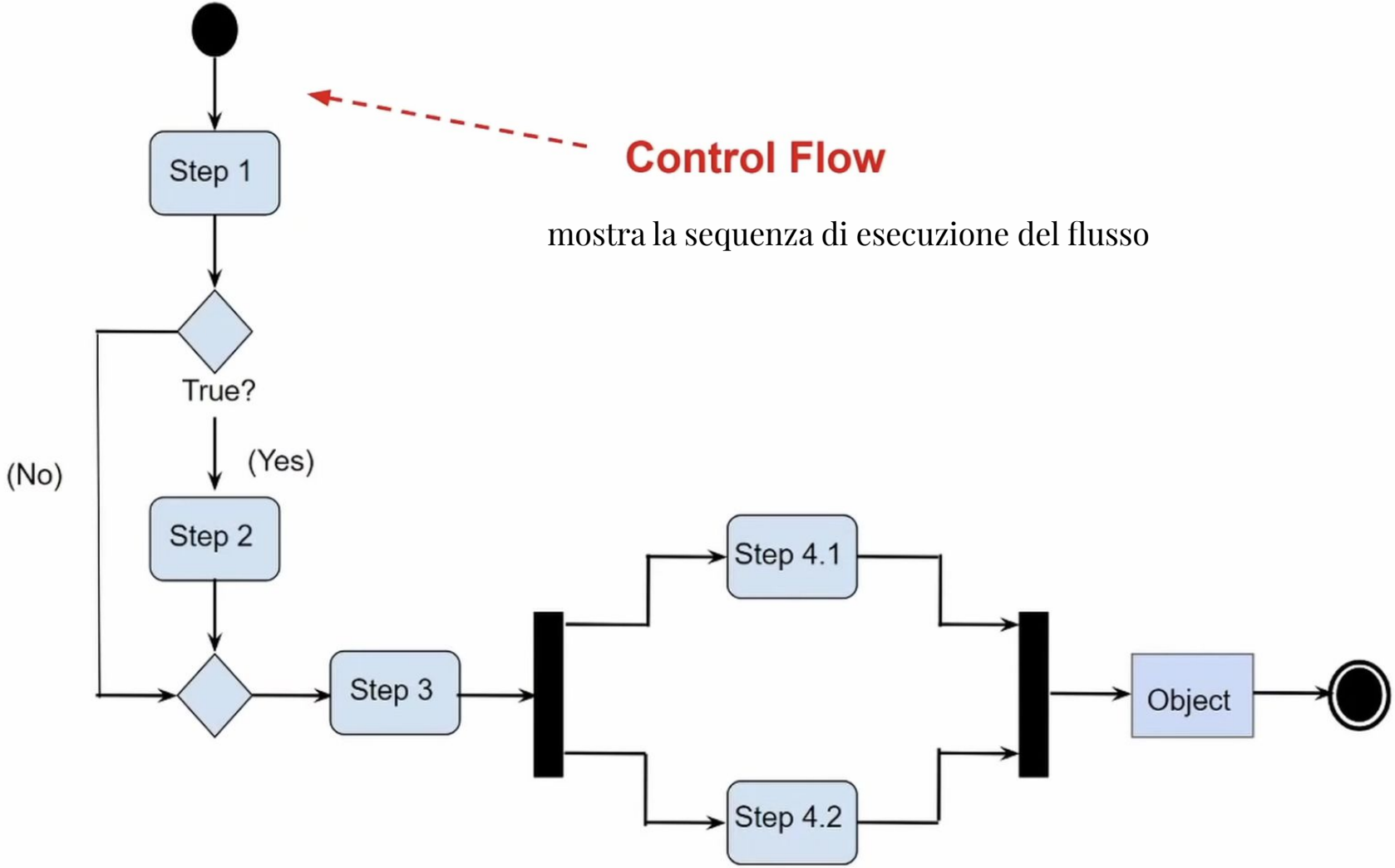
Applicazione dei diagrammi delle attività

- I diagrammi delle attività forniscono una visione task-centrica del comportamento di un insieme di oggetti
- Possono essere usati per descrivere vincoli di sequenza tra i casi d'uso, attività sequenziali tra gruppi di oggetti, o task di un progetto

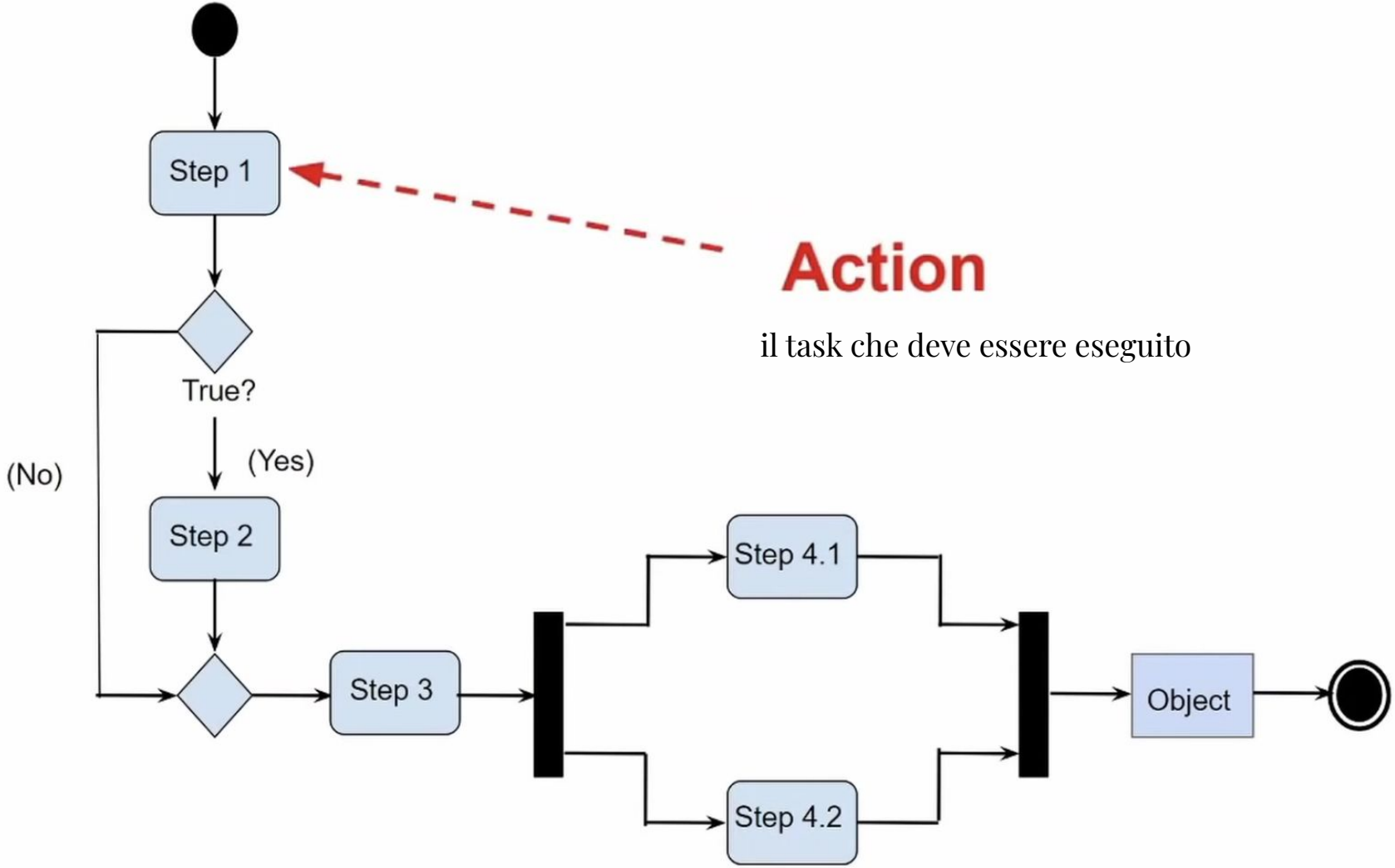

Step 1
True?
(No)
(Yes)
Step 2
Step 3
Step 4.1
Step 4.2
Object



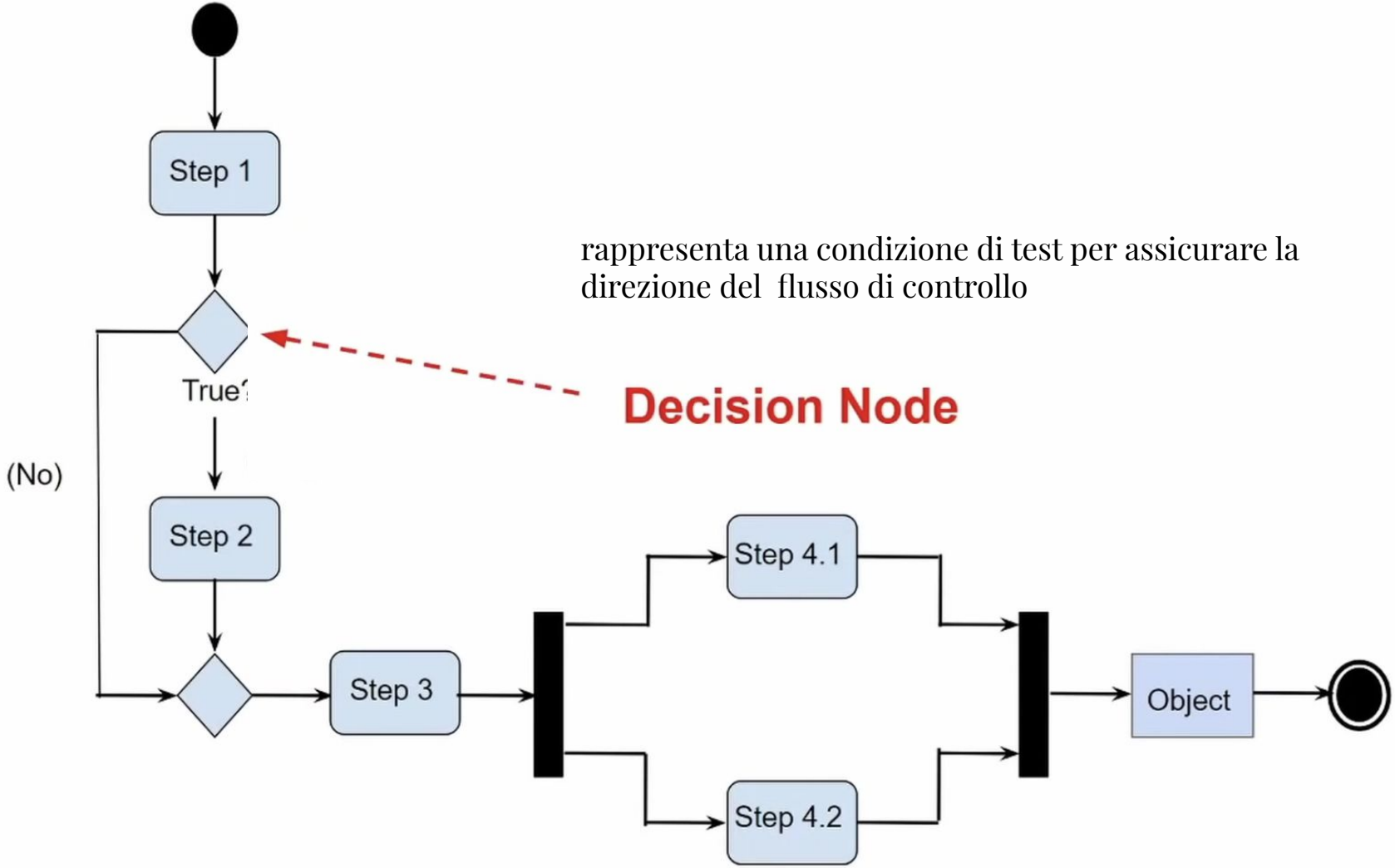
fa partire il flusso quando l'attività è chiamata dall'estrno



mostra la sequenza di esecuzione del flusso

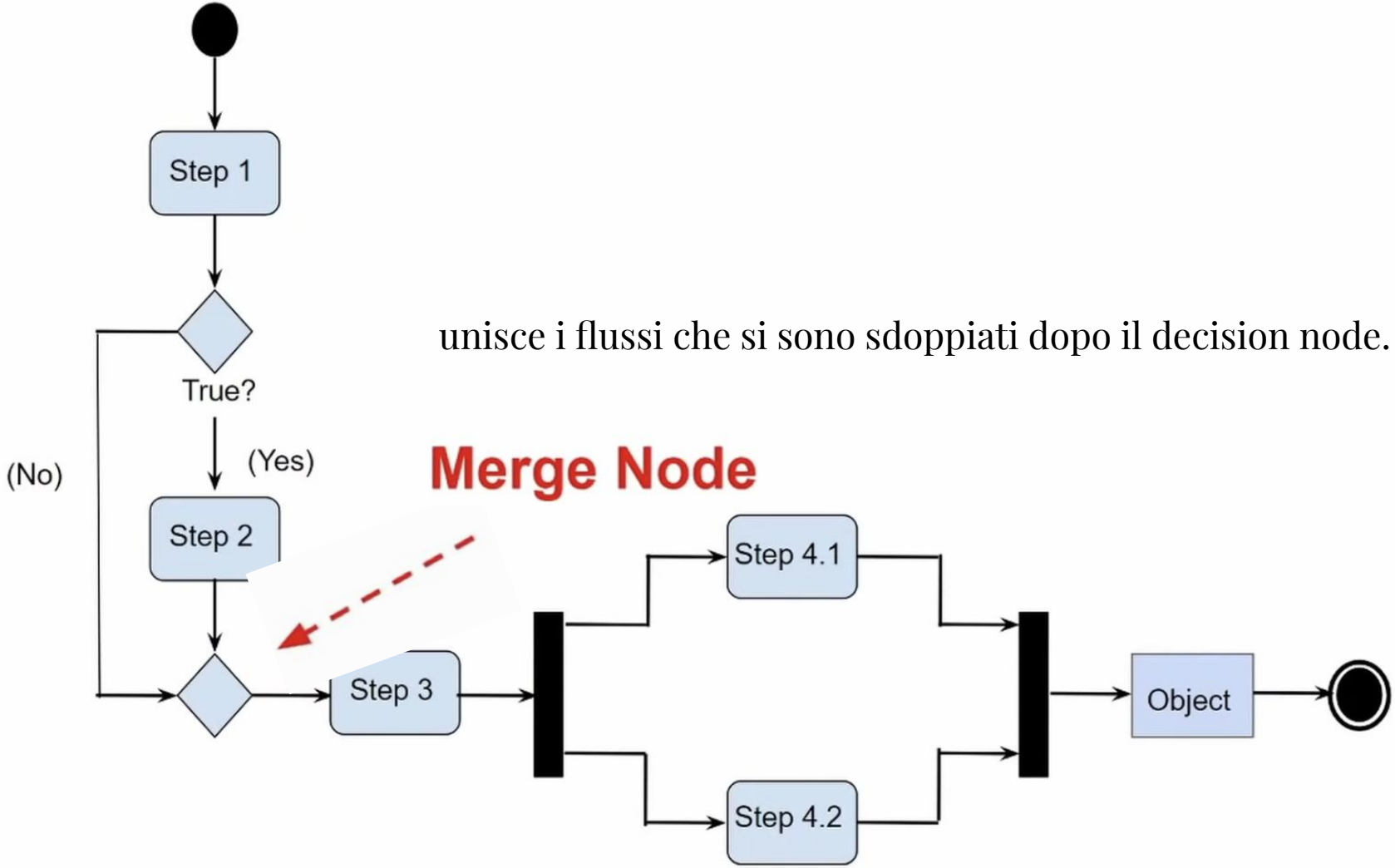


il task che deve essere eseguito

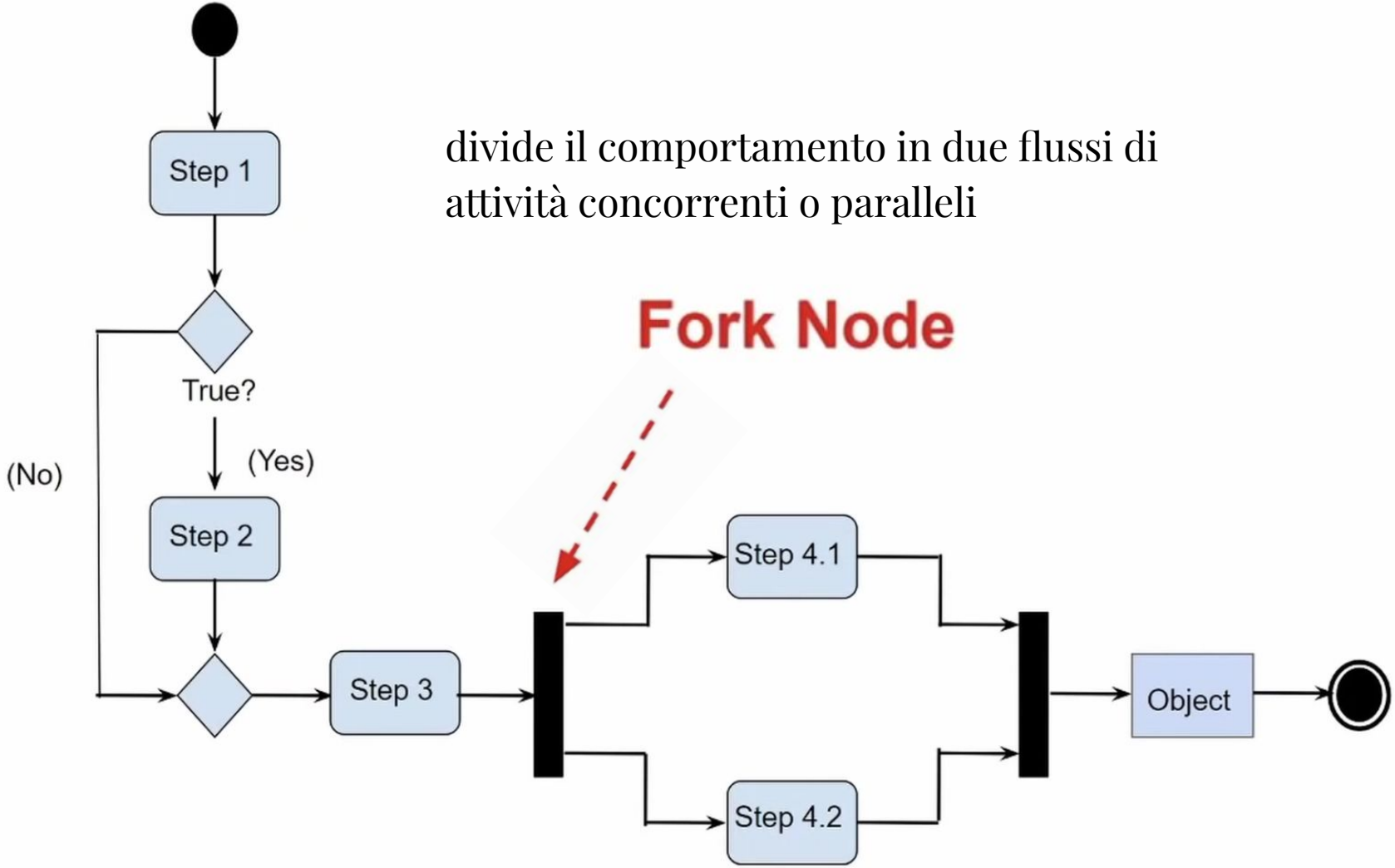


rappresenta una condizione di test per assicurare la direzione del flusso di controllo

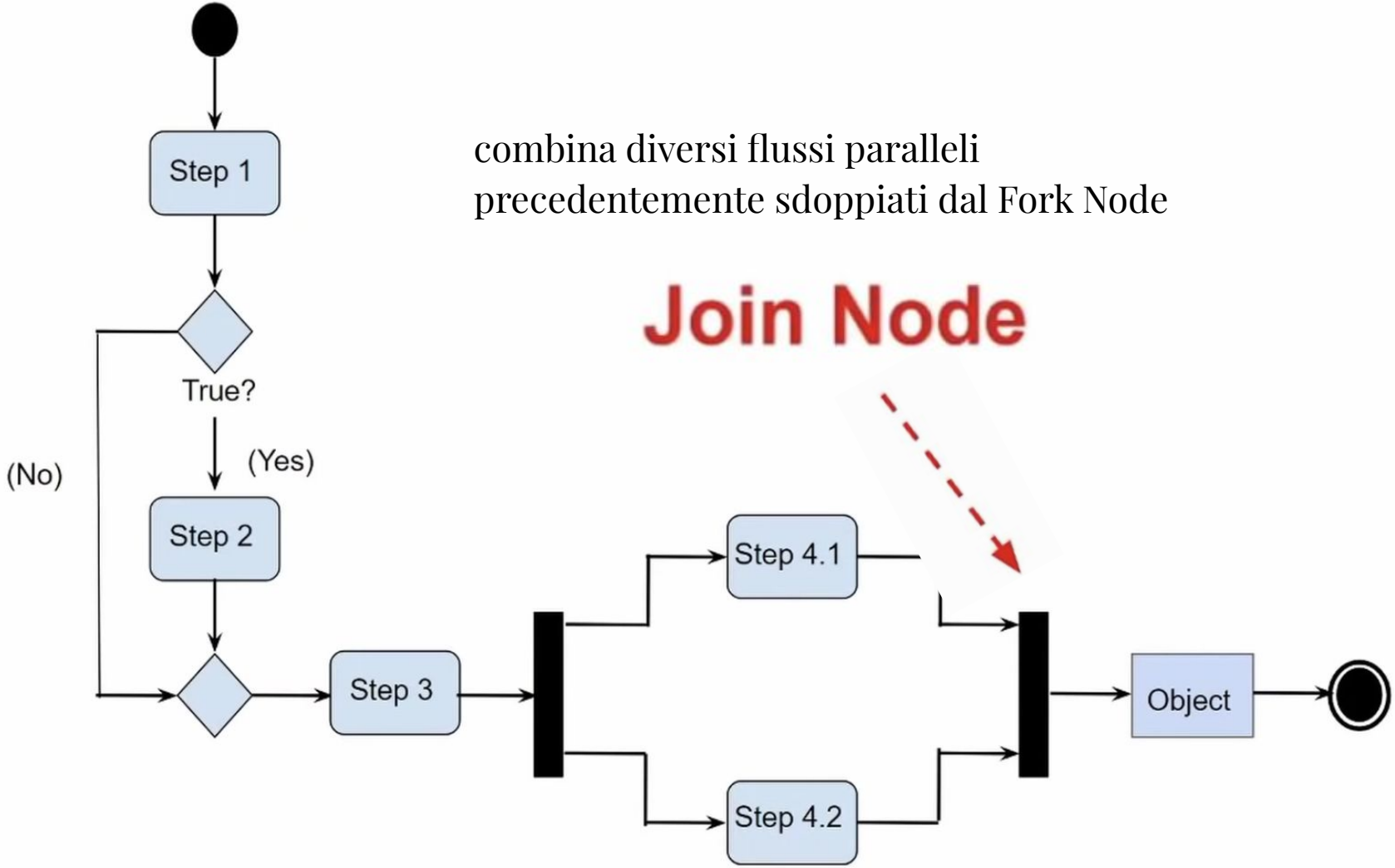
unisce i flussi che si sono sdoppiati dopo il decision node.



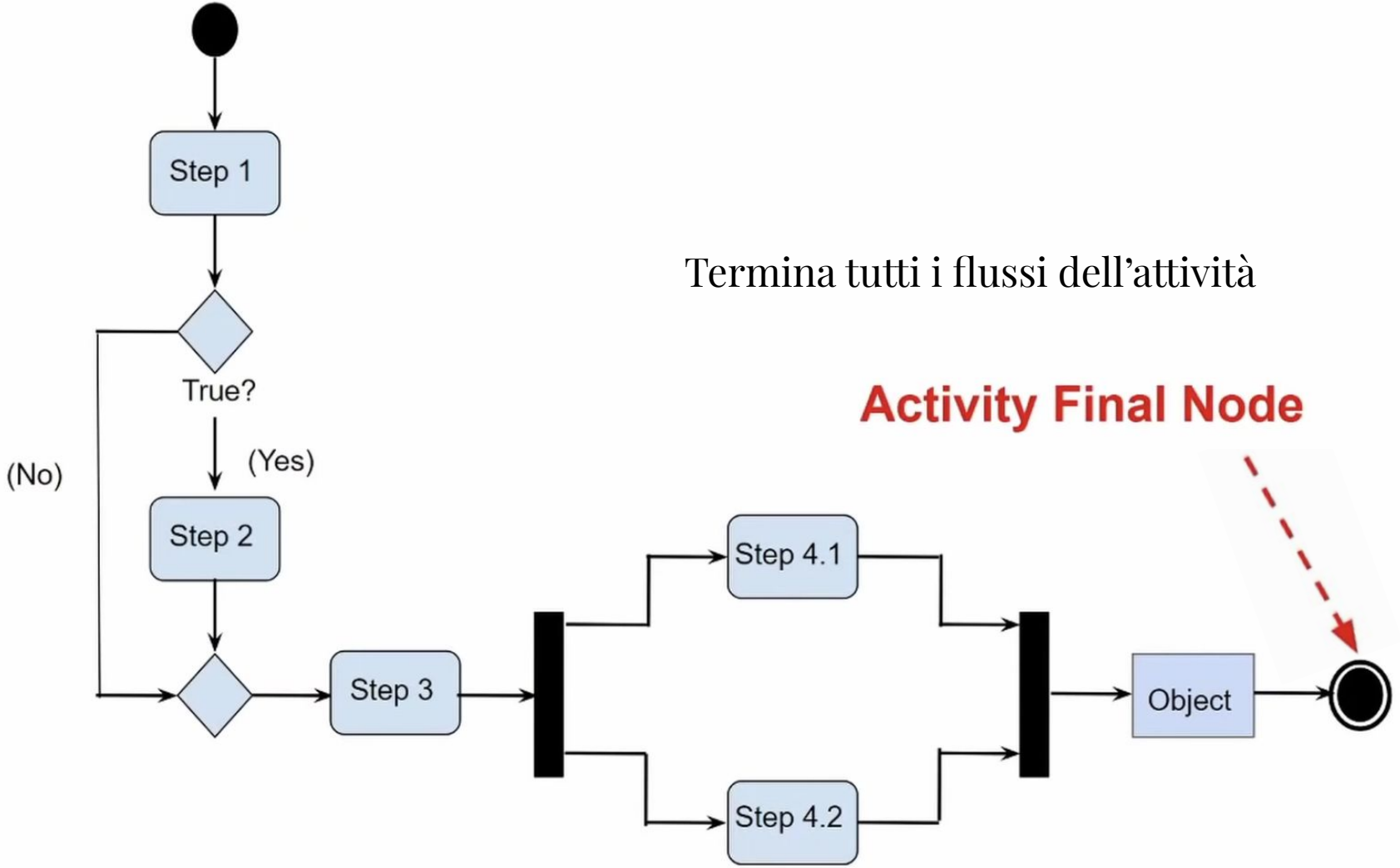
divide il comportamento in due flussi di attività concorrenti o paralleli



combina diversi flussi paralleli
precedentemente sdoppiati dal Fork Node

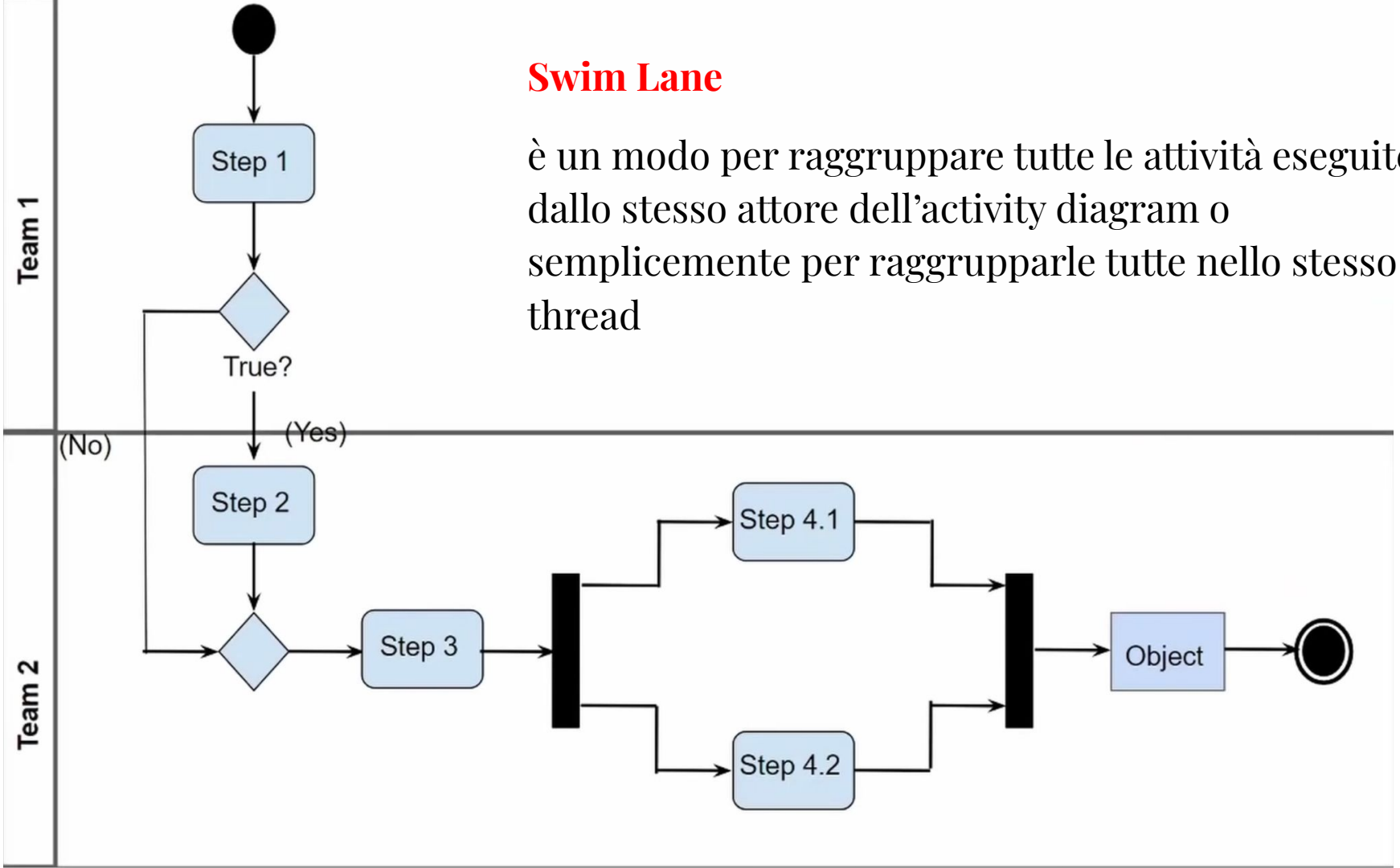


Termina tutti i flussi dell’attività

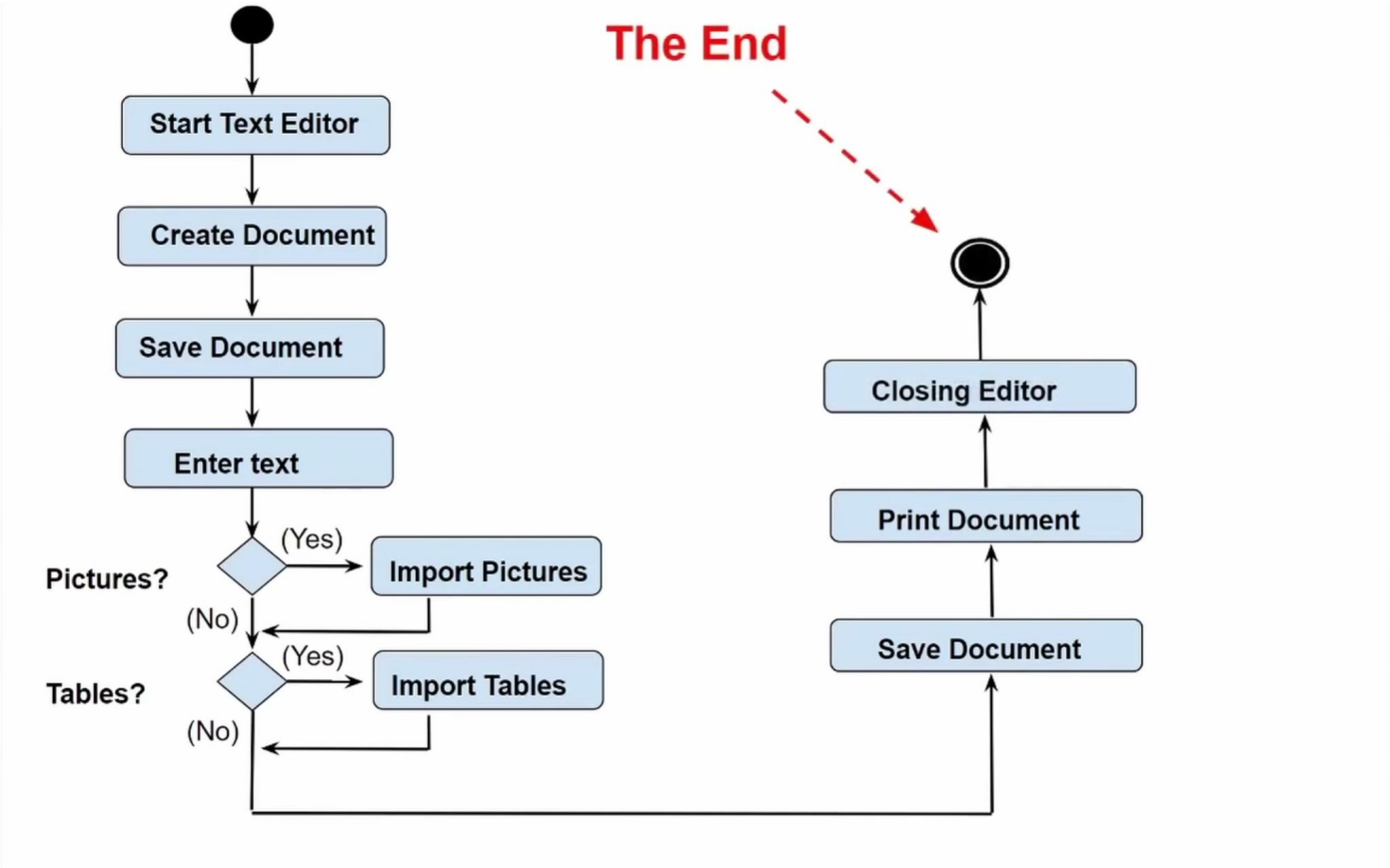


Swim Lane

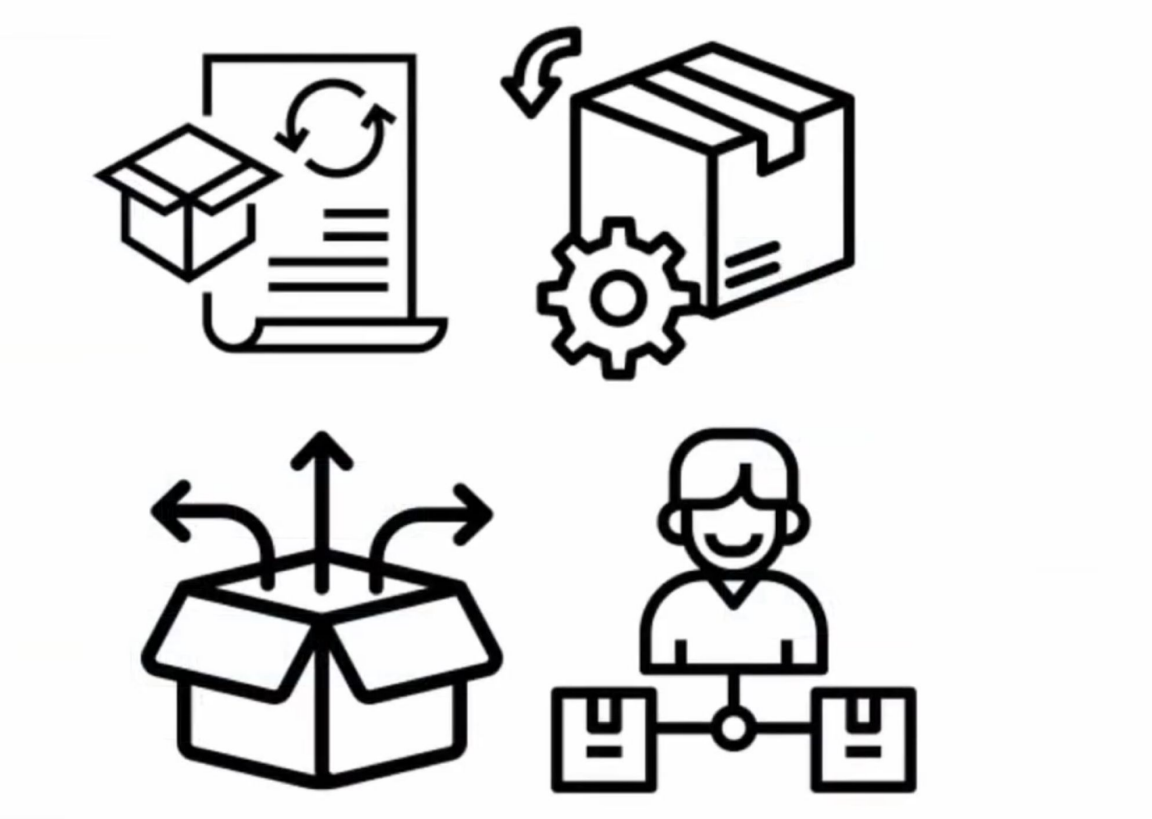
è un modo per raggruppare tutte le attività eseguite dallo stesso attore dell'activity diagram o semplicemente per raggrupparle tutte nello stesso thread

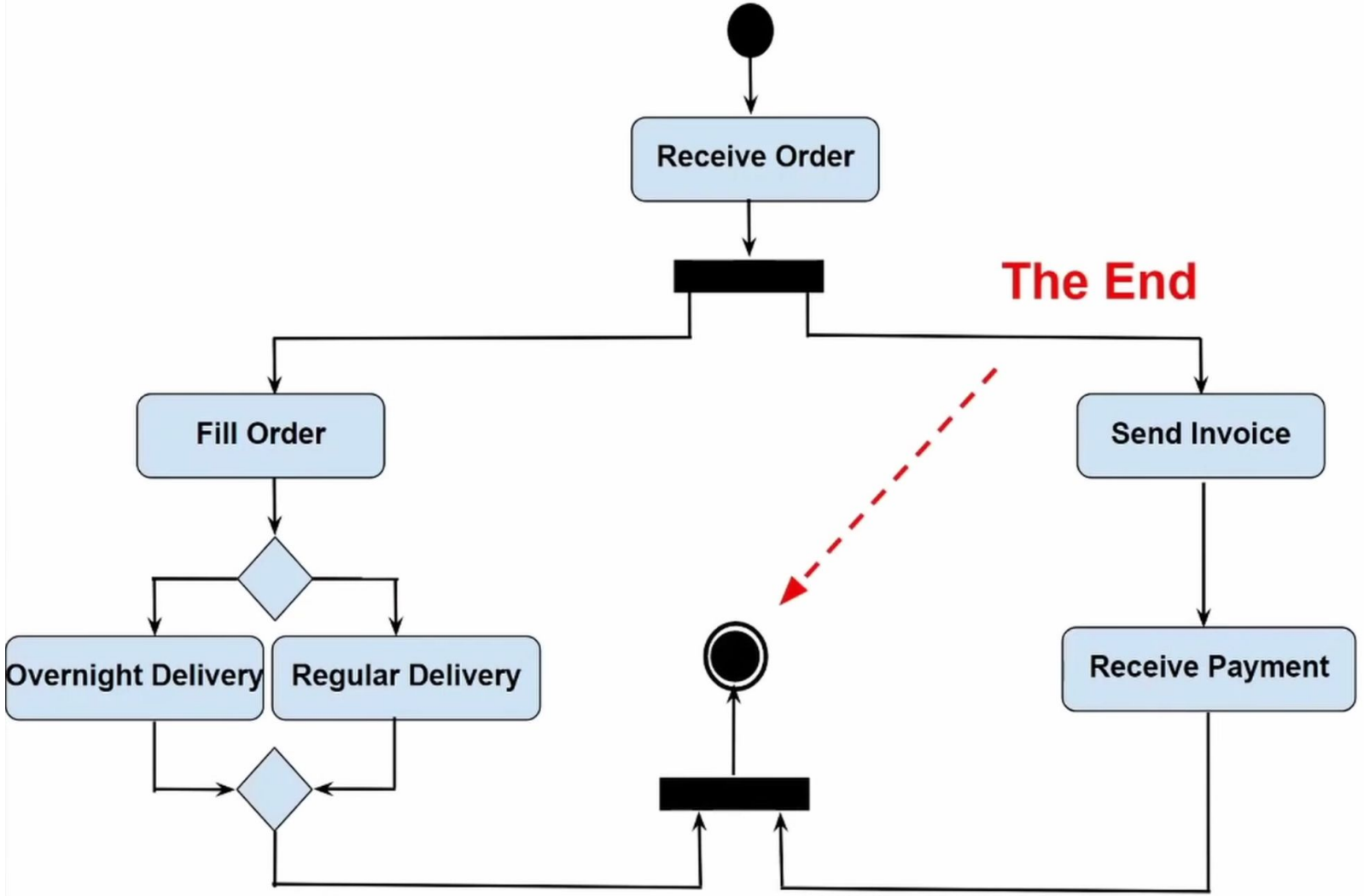


Il diagramma delle attività per la scrittura di un documento

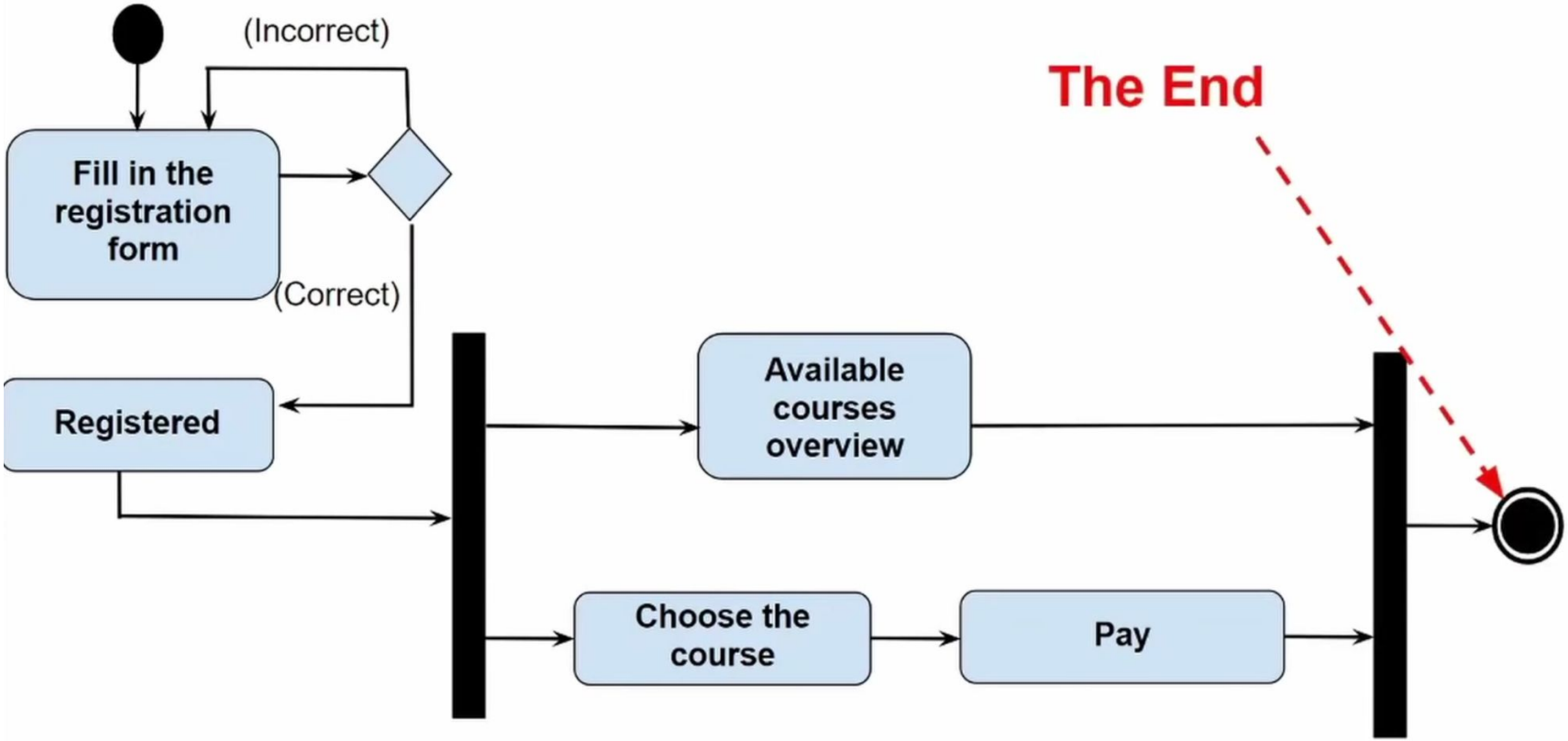


Esempio: Order Processing

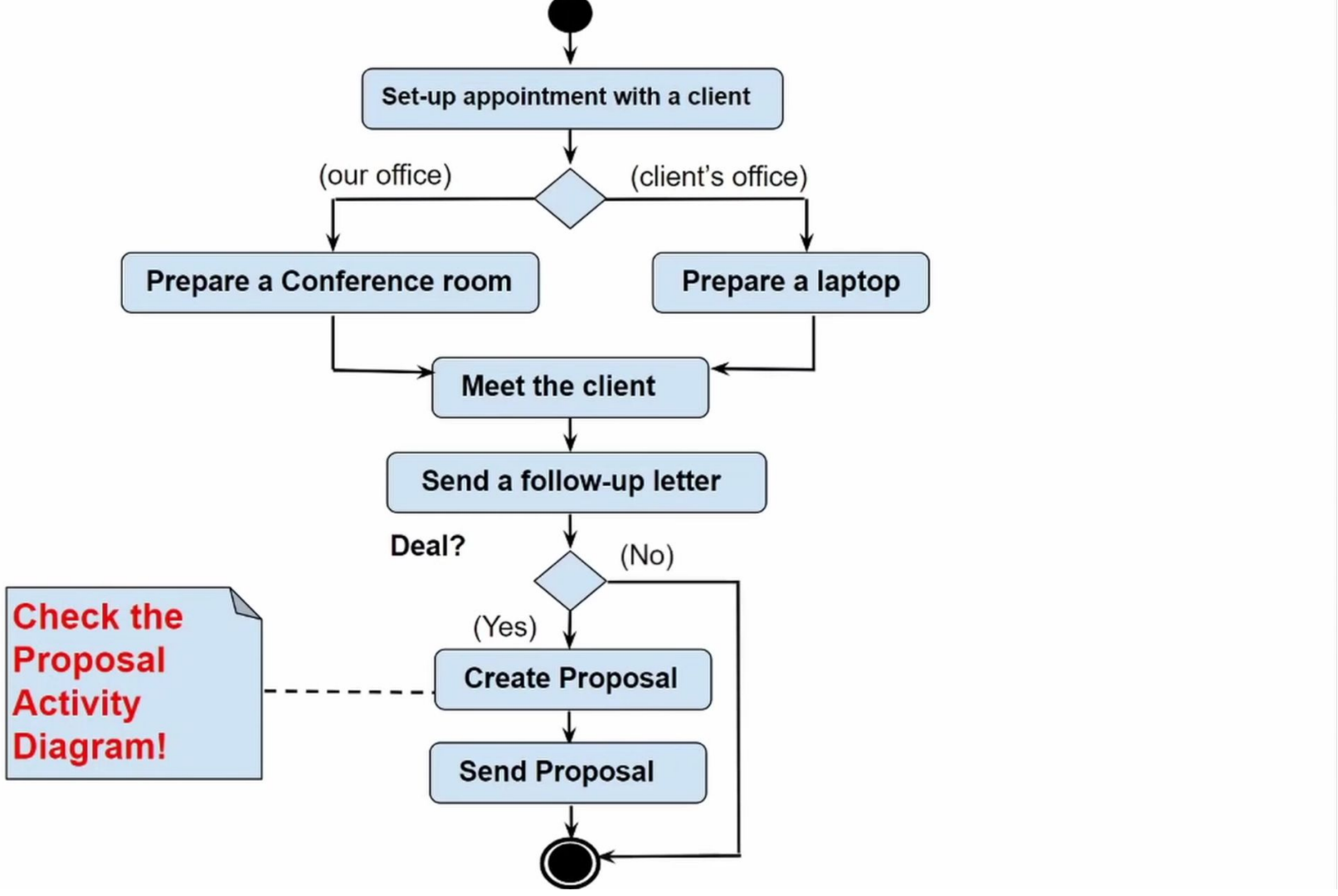


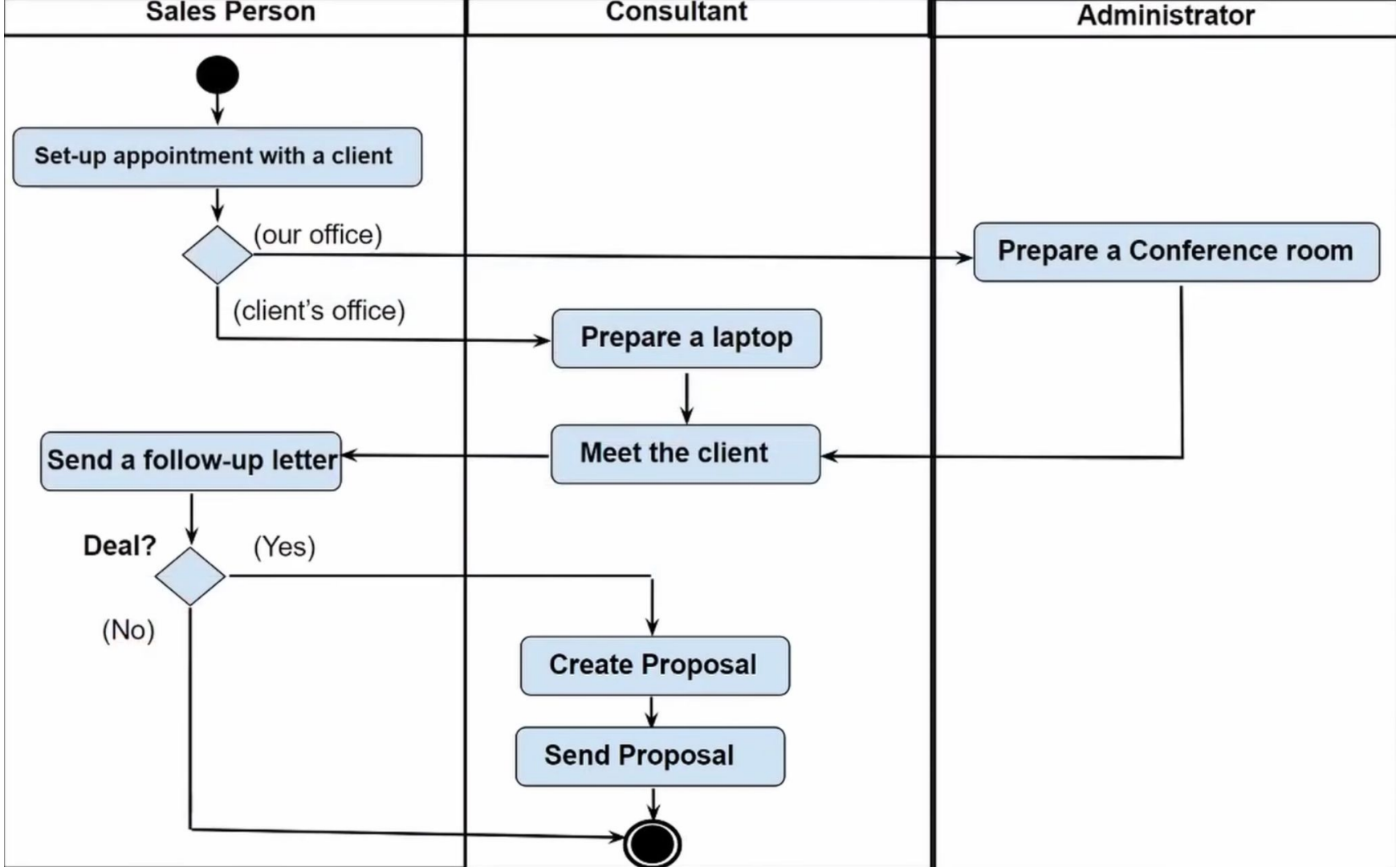

Receive Order
The End
Fill Order
Send Invoice
Overnight Delivery
Regular Delivery
Receive Payment

Esempio: corso di programmazione


(Incorrect)
Fill in the registration form
(Correct)
Registered
Available courses overview
Choose the course
Pay
The End

Esempio: onboarding new customers (con/senza swim lane)


Set-up appointment with a client
(our office)
(client's office)
Prepare a Conference room
Prepare a laptop
Meet the client
Send a follow-up letter
Deal?
(No)
(Yes)
Create Proposal
Send Proposal
Check the Proposal Activity Diagram!


Sales Person
Consultant
Administrator
Set-up appointment with a client
(our office)
Prepare a Conference room
(client's office)
Prepare a laptop
Meet the client
Send a follow-up letter
Deal?
(Yes)
(No)
Create Proposal
Send Proposal