

Copyright 2005 Scott W. Ambler

UML 2.0: ACTIVITY DIAGRAM

Ingegneria del Software 2023/2024

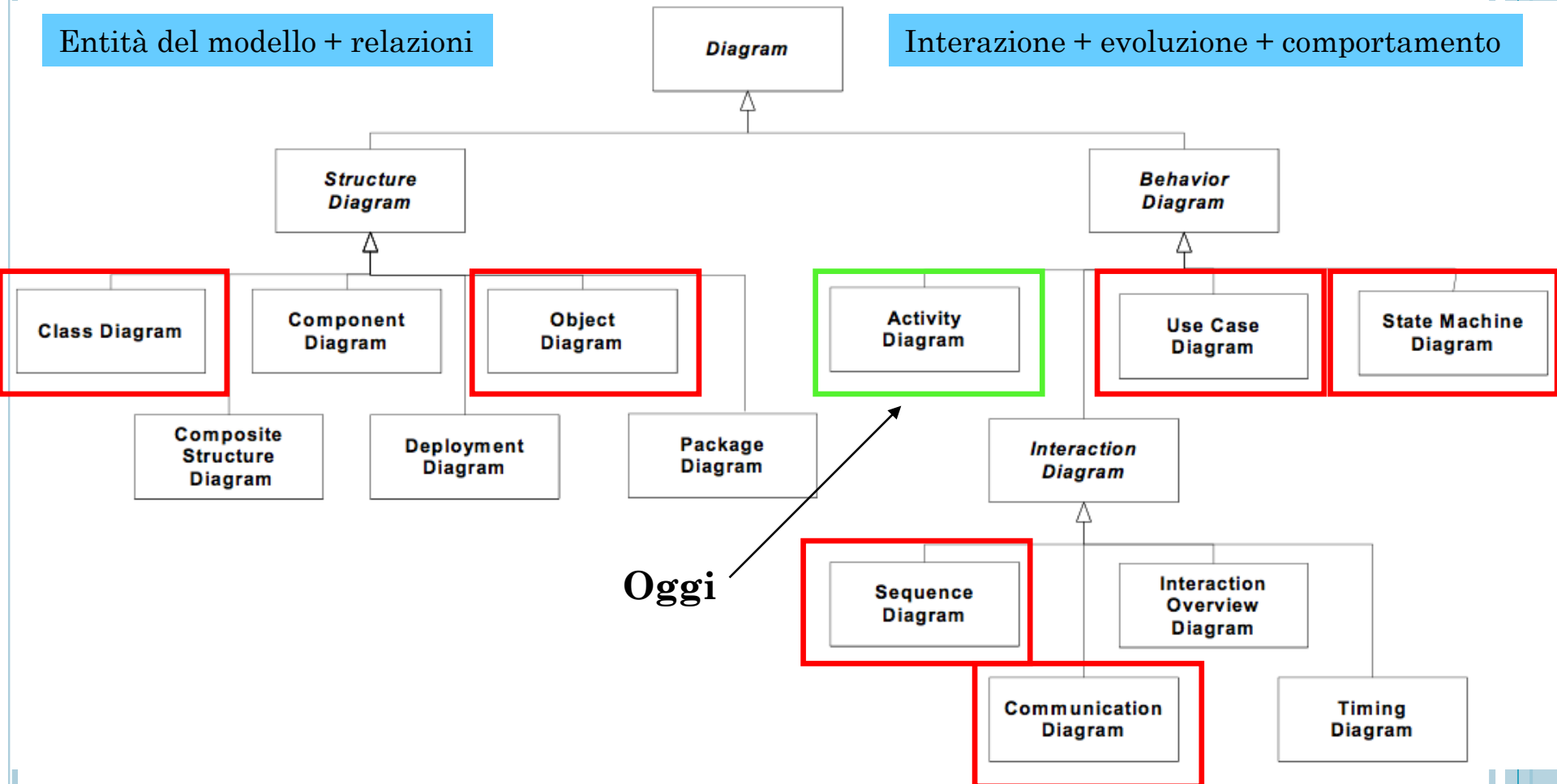
ORIENTIAMOCI ...

Modellazione statica

Entità del modello + relazioni

Modellazione dinamica

Interazione + evoluzione + comportamento

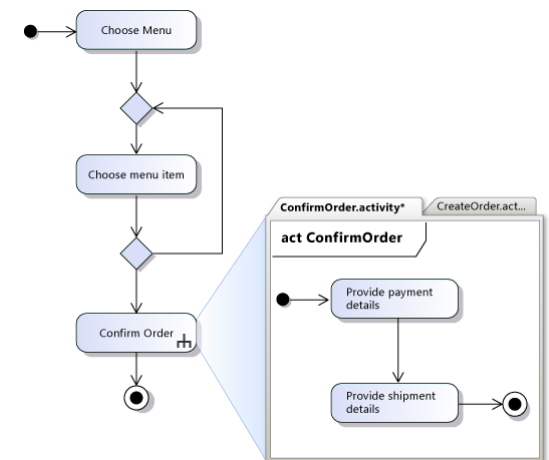


AGENDA

◉ Diagrammi delle attività

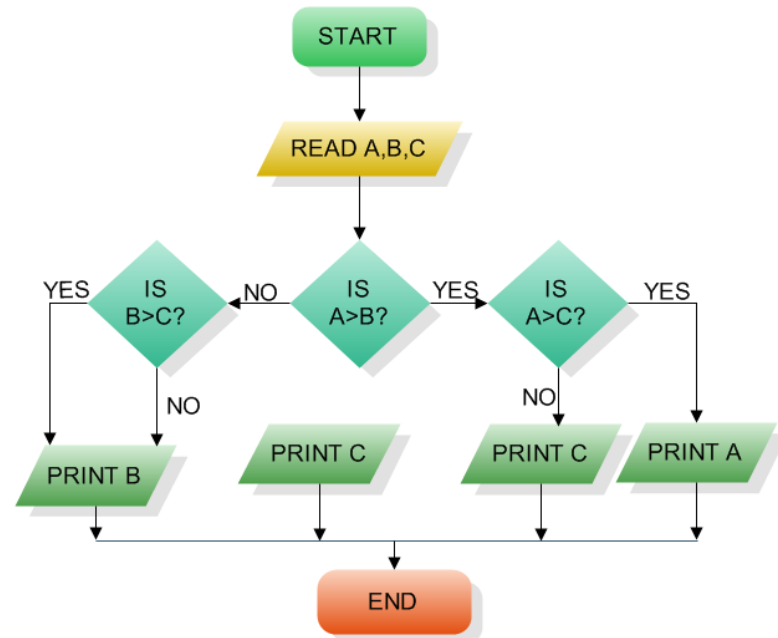
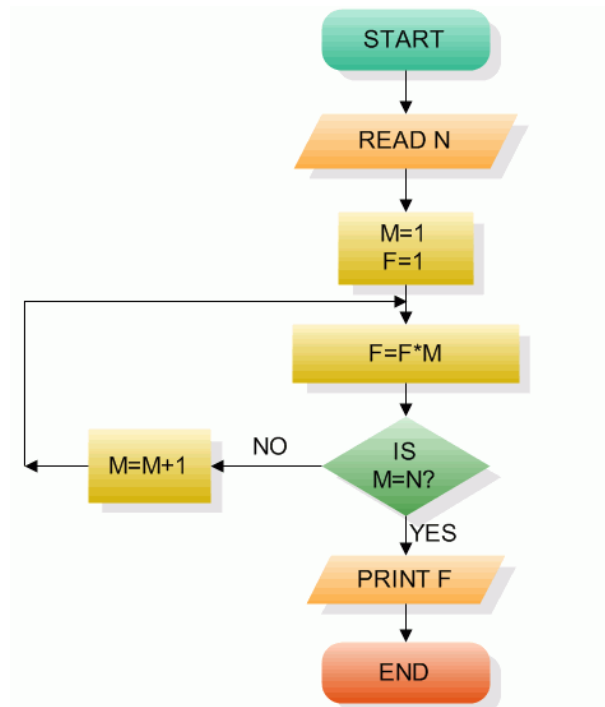
- Dove e come sono usati?
 - ◉ Diagrammi ‘poliedrici’
- Semantica basata sul concetto di token
- Notazione
 - ◉ Nodi, Flussi, Swimlanes ...
 - ◉ Diagrammi complessi
 - ◉ Vediamo solo costrutti più usati

◉ Esempi d'uso



FLOWCHART

- Li avete già visti?
- Ve li ricordate?



Esprimono algoritmi in modo grafico!

DIAGRAMMI DELLE ATTIVITÀ

- Descrivono **come viene svolta** un'attività relativa ad una qualsiasi entità, che può essere un:
 - operazione di una classe, classe, componente, caso d'uso, persona fisica, procedura, processo di business ...
- “come viene svolta” = quale è il **flusso di azioni** che devono accadere (o che accadono)
 - quali azioni (cioè la loro semantica)
 - in che ordine

DIAGRAMMI DELLE ATTIVITÀ

- Descrivono **come viene svolta** un'attività relativa **ad una qualsiasi entità**, che può essere un:
 - operazione di una classe, classe, componente, caso d'uso, persona fisica, procedura, processo di business ...
- “come viene svolta” = quale è il **flusso di azioni** che devono accadere (o che accadono)
 - quali azioni
 - in che ordine
- Anche **più di una entità** contemporaneamente:
 - Ad esempio modellare il processo di business **gestione ordine** da parte di un'azienda
 - Il **cliente** manda l'ordine
 - L'**azienda** lo riceve
 - L'**azienda** controlla se la merce è in magazzino
 - L'**azienda** prepara il pacco e la fattura

DOVE E COME SONO USATI?

- Usati in contesti diversi, fasi diverse dello sviluppo e per scopi diversi

DOVE E COME SONO USATI?

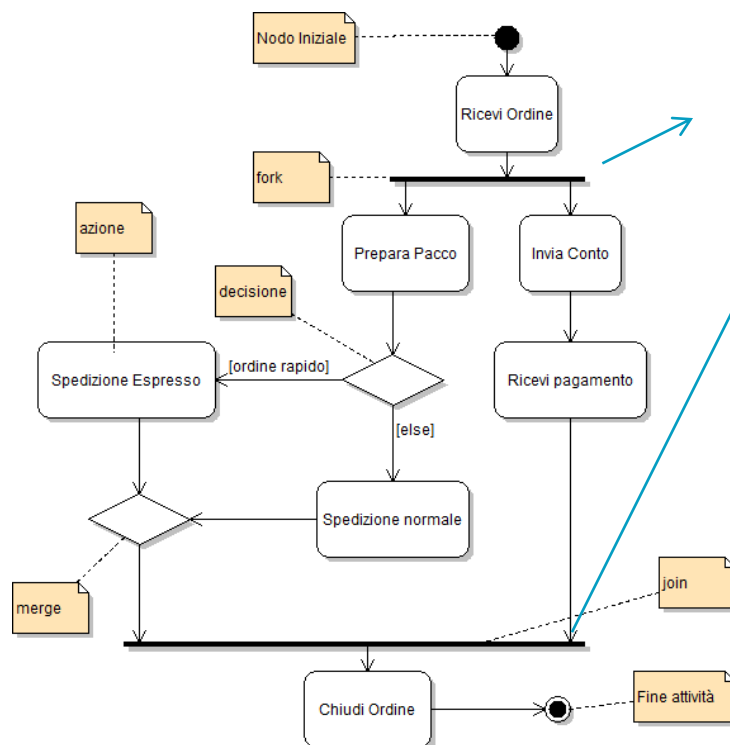
- Usati in contesti diversi, fasi diverse dello sviluppo e per scopi diversi
- Modellare **processi di business** e workflow (**analisi**)
 - Capire il business e il dominio prima di implementare il sistema
 - Es. Gestione di un ordine in un'azienda
Acquisire dottorato di ricerca in un Università
- Modellare il flusso di un caso d'uso (**analisi**)
- Modellare un'operazione di una classe (**progettazione**)
 - Alternativa allo pseudocodice
- Modellare un algoritmo (**progettazione**)
 - Simili a un flowchart (alternativa allo pseudocodice)
- Come linguaggio di programmazione (**codifica**)
 - UML eseguibile

ATTIVITÀ

- Un'attività è costituita da un **flusso di azioni** che ne sono i mattoni

Ricevi ordine

Spesso l'azione è descritta informalmente in linguaggio naturale
- Simili a flowchart, con la differenza che supportano l'elaborazione **concorrente/parallela**

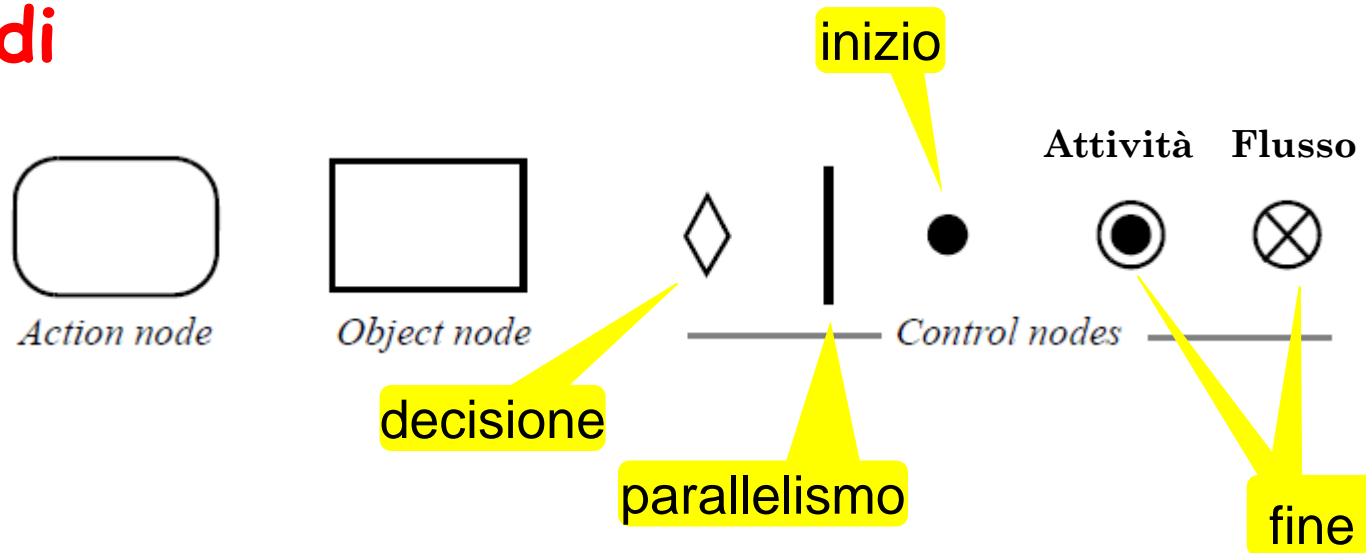


Denotano parallelismo

Gestione ordine

ATTIVITÀ: NOTAZIONE

Nodi



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

Flussi



‘frece che collegano i nodi’

NODI AZIONE

- Un azione può essere espressa in modo informale
- Un'azione può invocare un'altra **attività**
- Un azione può invocare un'operazione di una classe
 - anche 'comportamento' se siamo in prospettiva concettuale
- Volendo (ma è raro) si può anche scrivere un frammento di pseudo-codice all'interno del simbolo di azione

Manda la fattura al cliente

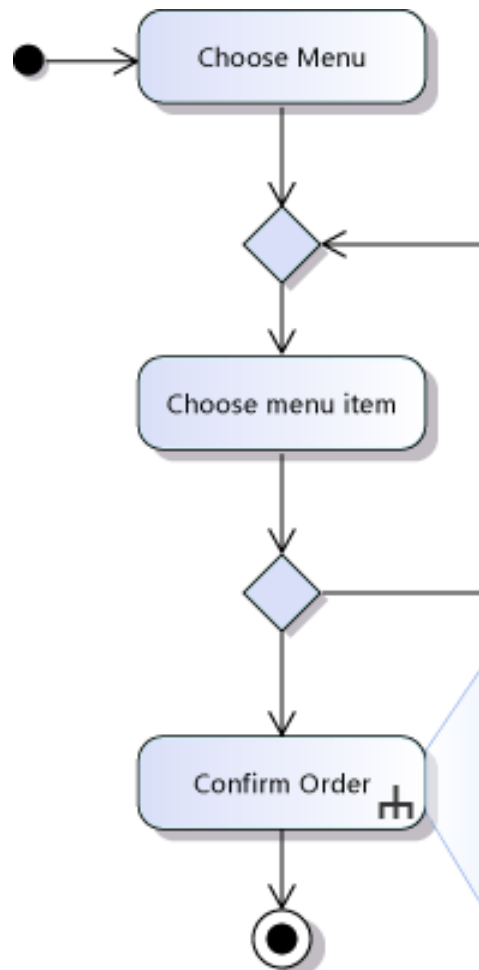
Preparare fattura cliente

rh

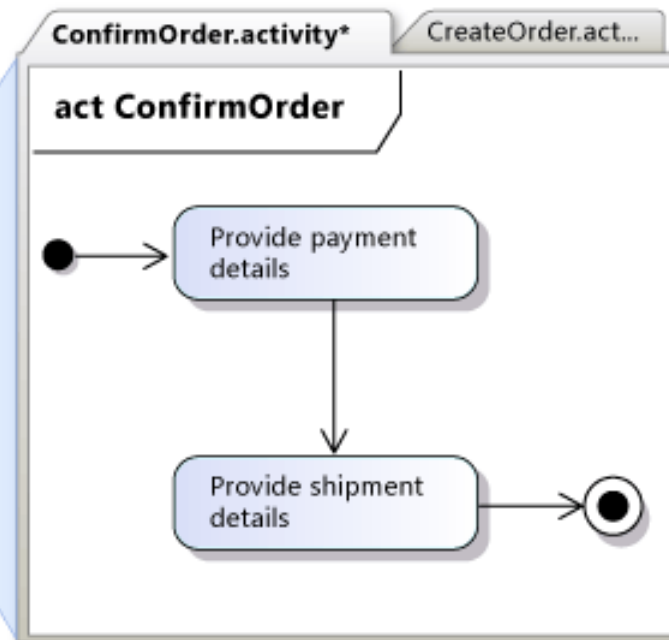
Fare Fattura
(Contabile::componiFattura)

```
FOR every Employee  
  calculate salary  
  print check  
ENDFOR
```

SIMBOLO RAKE

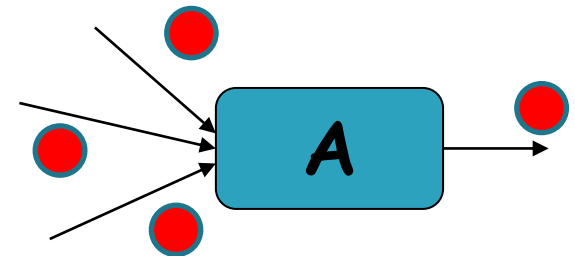


Indica che un azione è specificata a parte da un'altro diagramma di attività



TRANSIZIONI E TOKEN

- Per capire la semantica dei diagrammi di attività, bisogna capire il concetto di **token**
 - Sono come pedine che viaggiano lungo il diagramma



- Un **nodo azione** viene eseguito quando sono presenti i token su tutti gli archi in entrata
- Al termine di un'azione, sono generati i token su tutti gli archi in uscita

PROCESSO DI BUSINESS: GESTIONE ORDINE

- Supponiamo di voler descrivere un **processo di business Gestione d'ordine** da parte di un'azienda mediante un activity diagram
- Il processo può essere descritto brevemente così:
 - Quando un'azienda riceve un ordine partono in parallelo due flussi:
 - Preparazione pacco e spedizione del pacco
 - Invio conto/fattura al cliente e ricezione pagamento
 - L'ordine viene chiuso solo se entrambi i due flussi precedenti terminano

ACTIVITY DIAGRAM: ALTO LIVELLO

Molto ad alto livello (rake)
- Cerca merce in magazzino
- Prepara il pacco

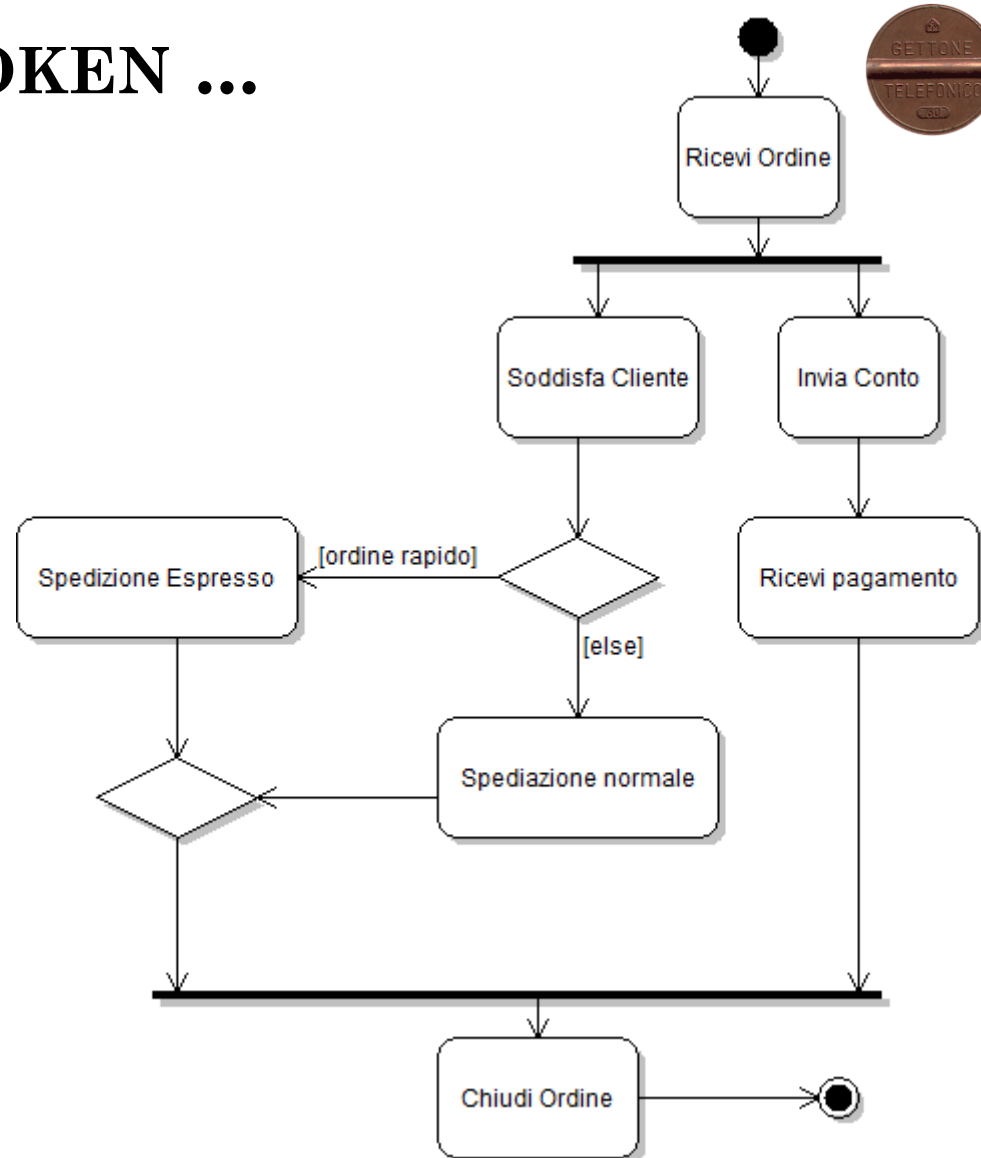
Nell'ordine sono contenute tutte le info sull'ordine e sul tipo di spedizione

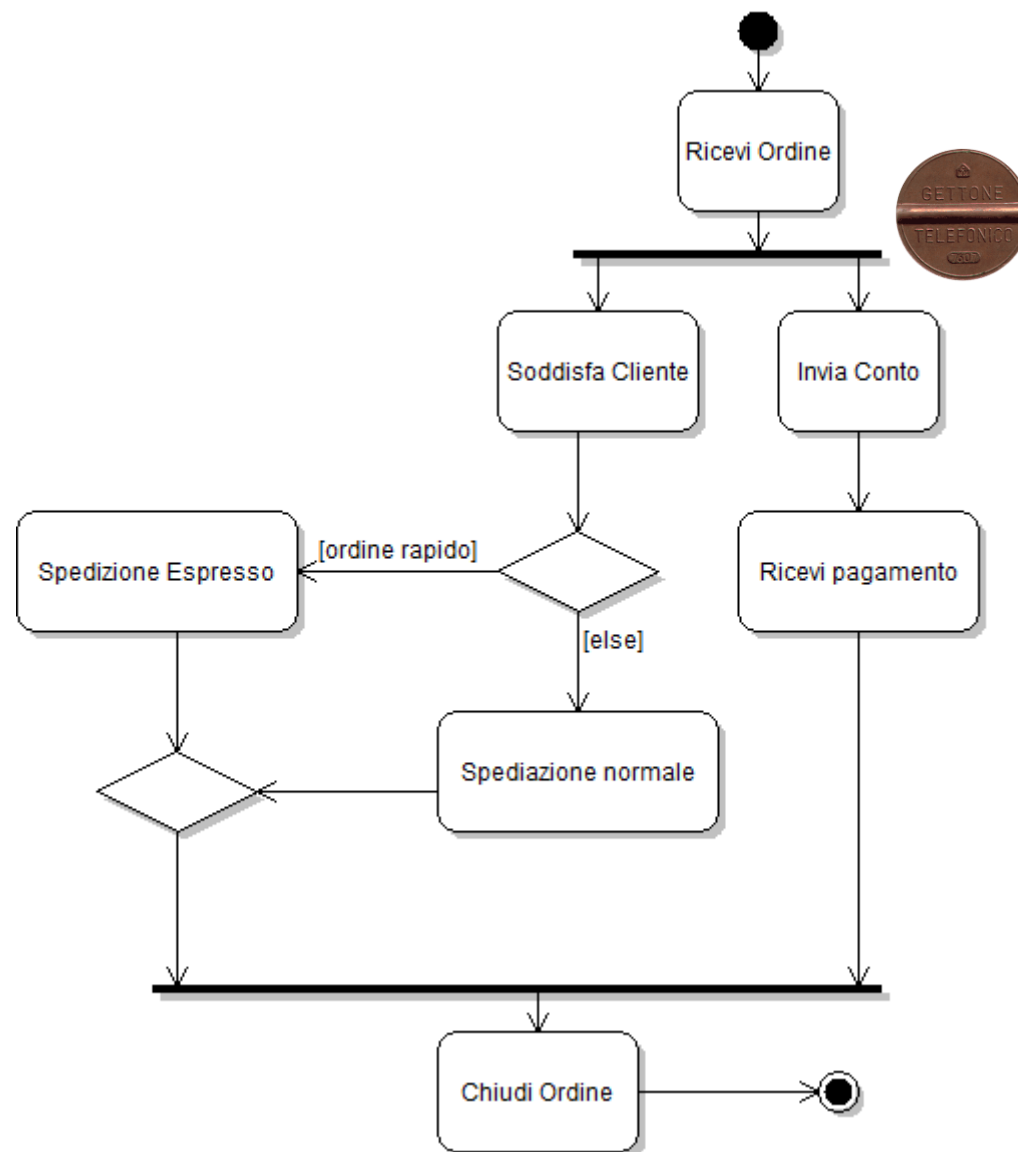


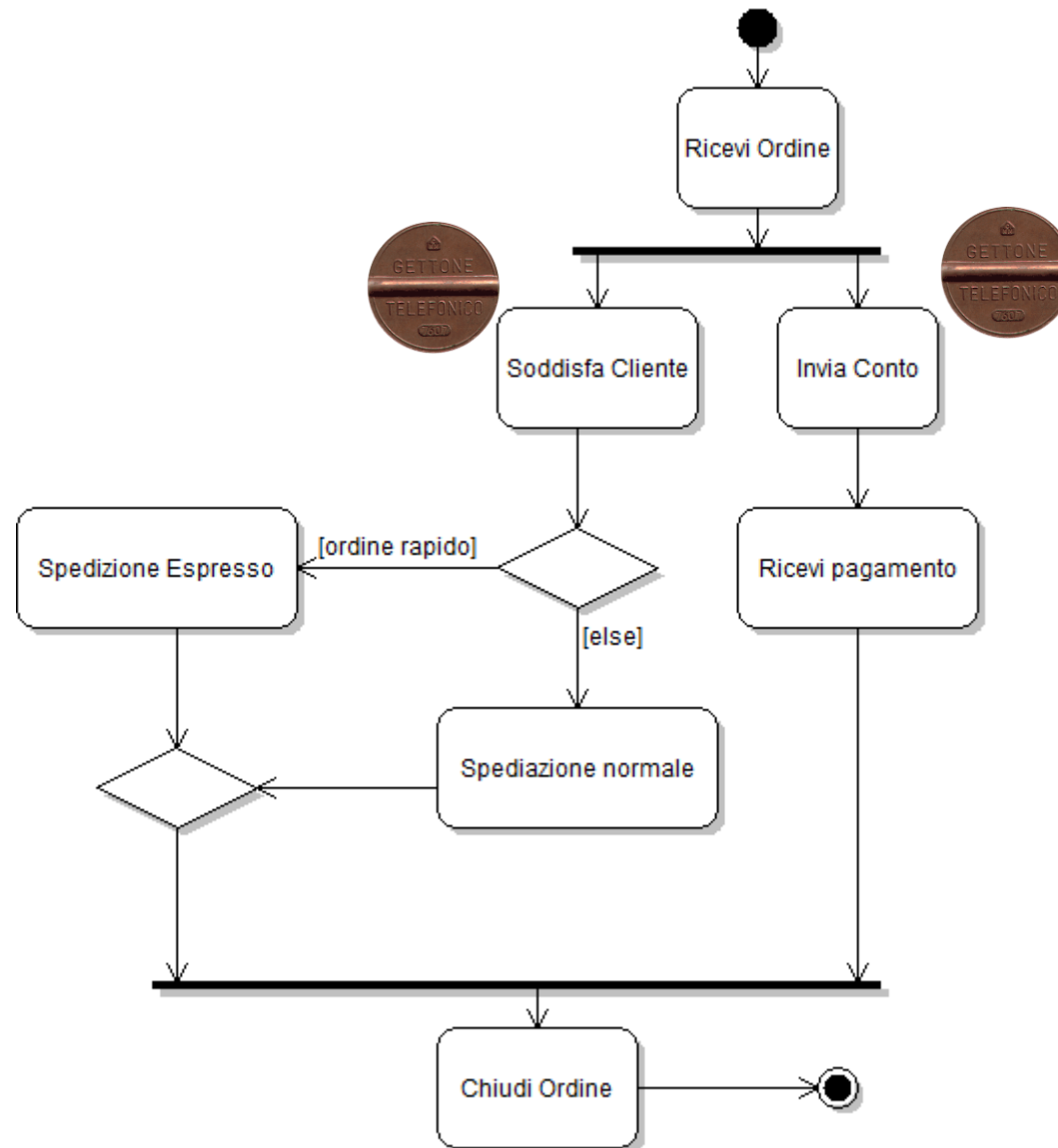
Modelliamo come avviene il processo di gestione di un ordine

Quando il processo è attivato viene generato un token

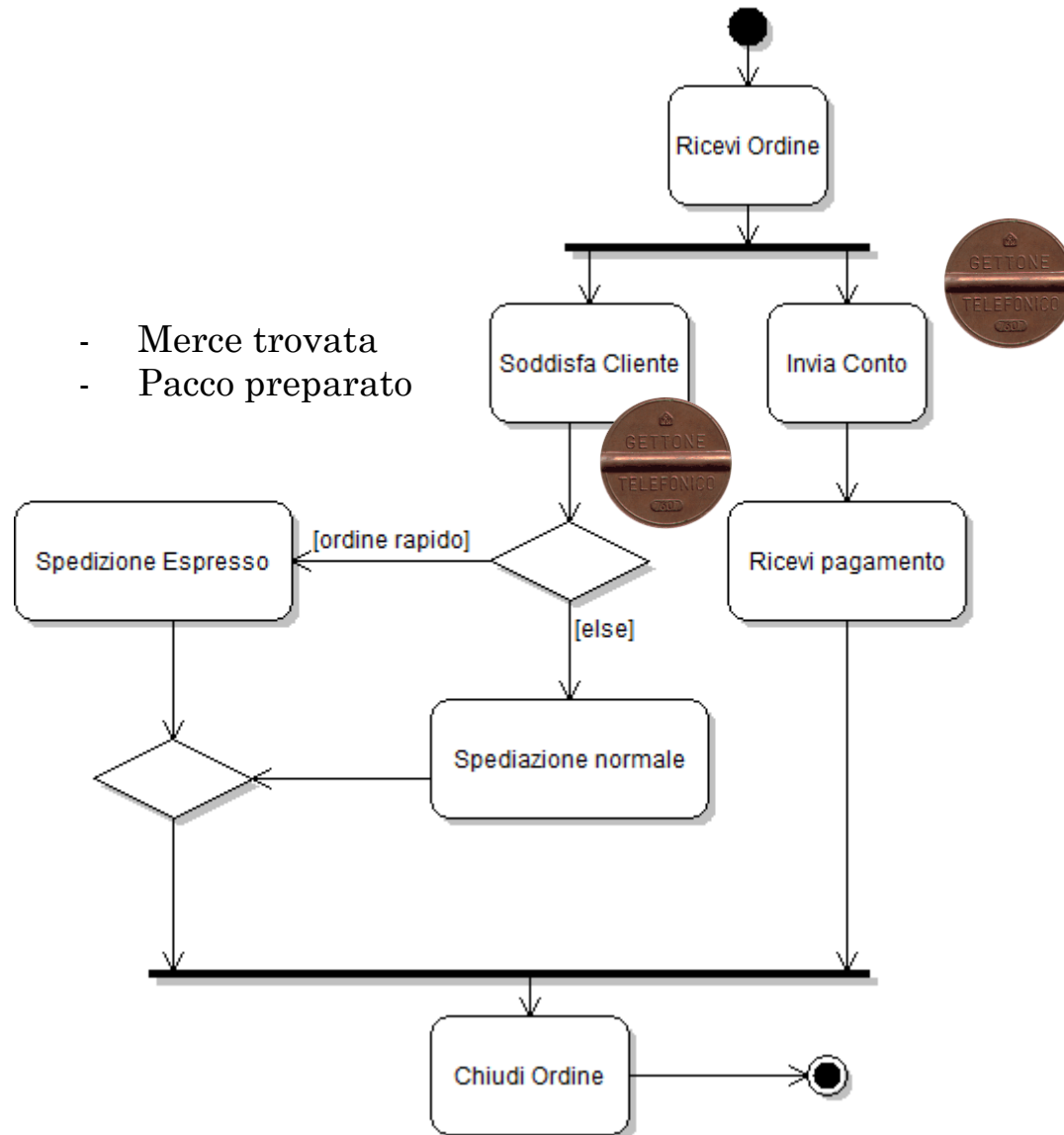
ECCO I TOKEN ...

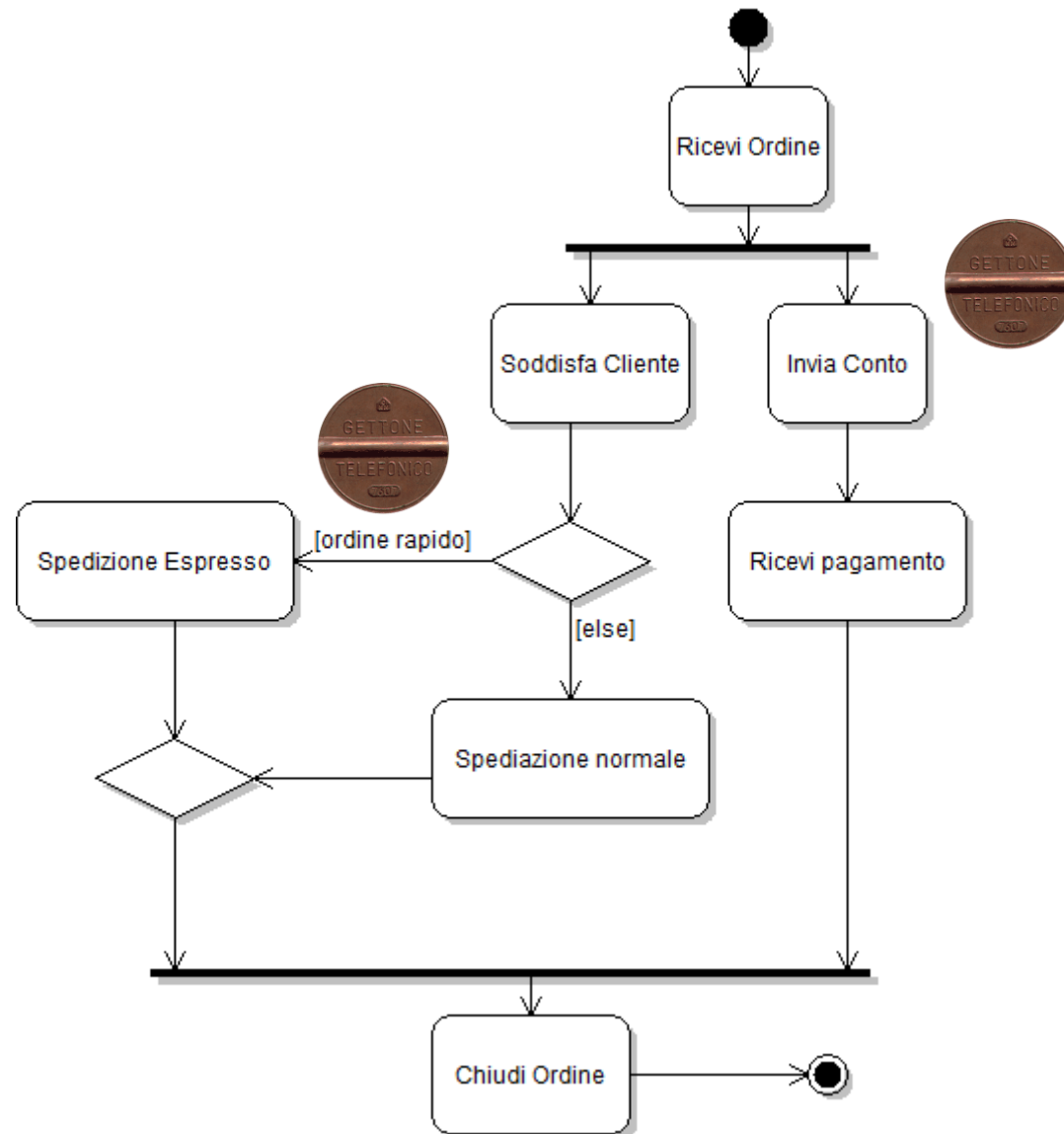


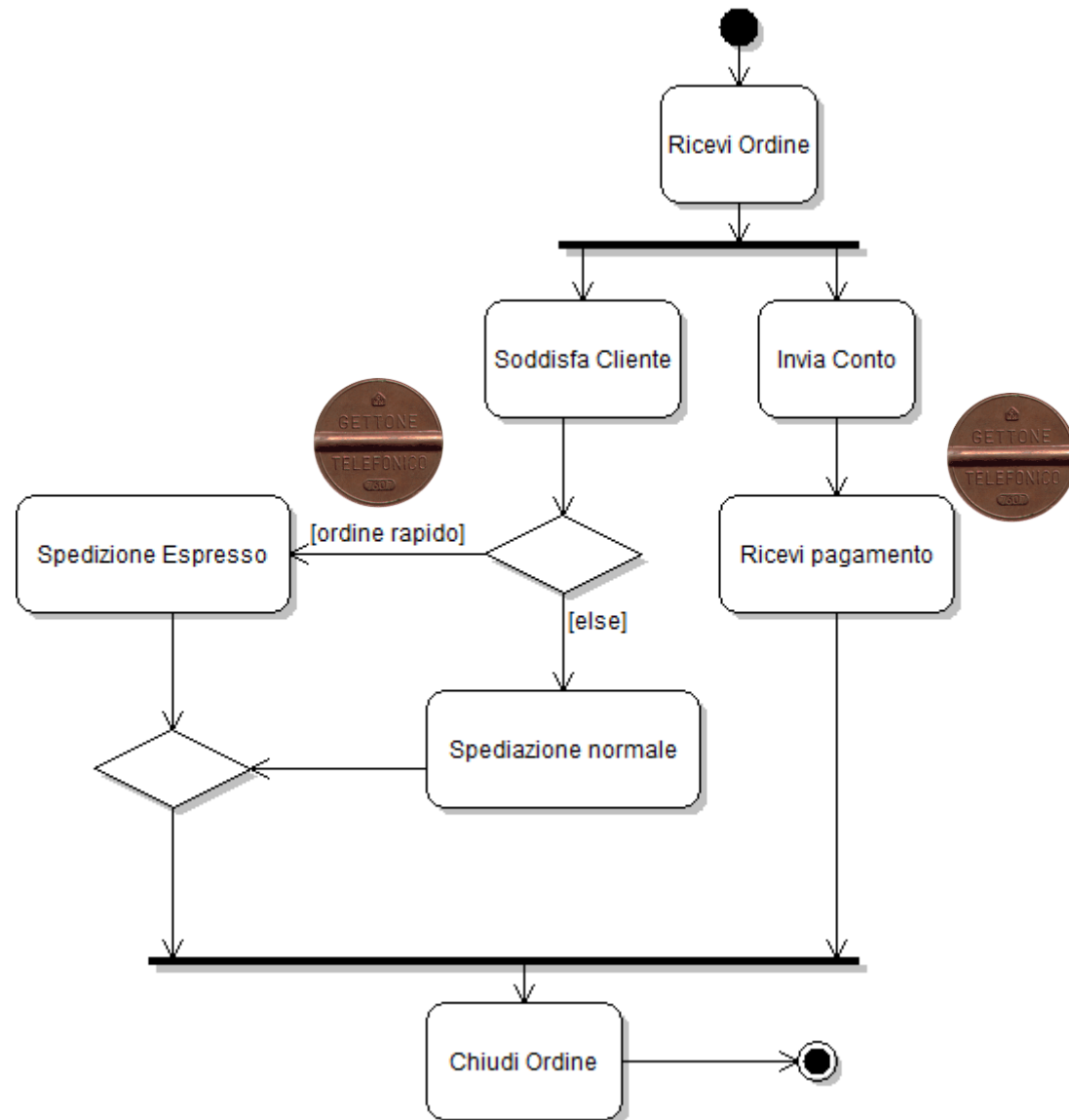


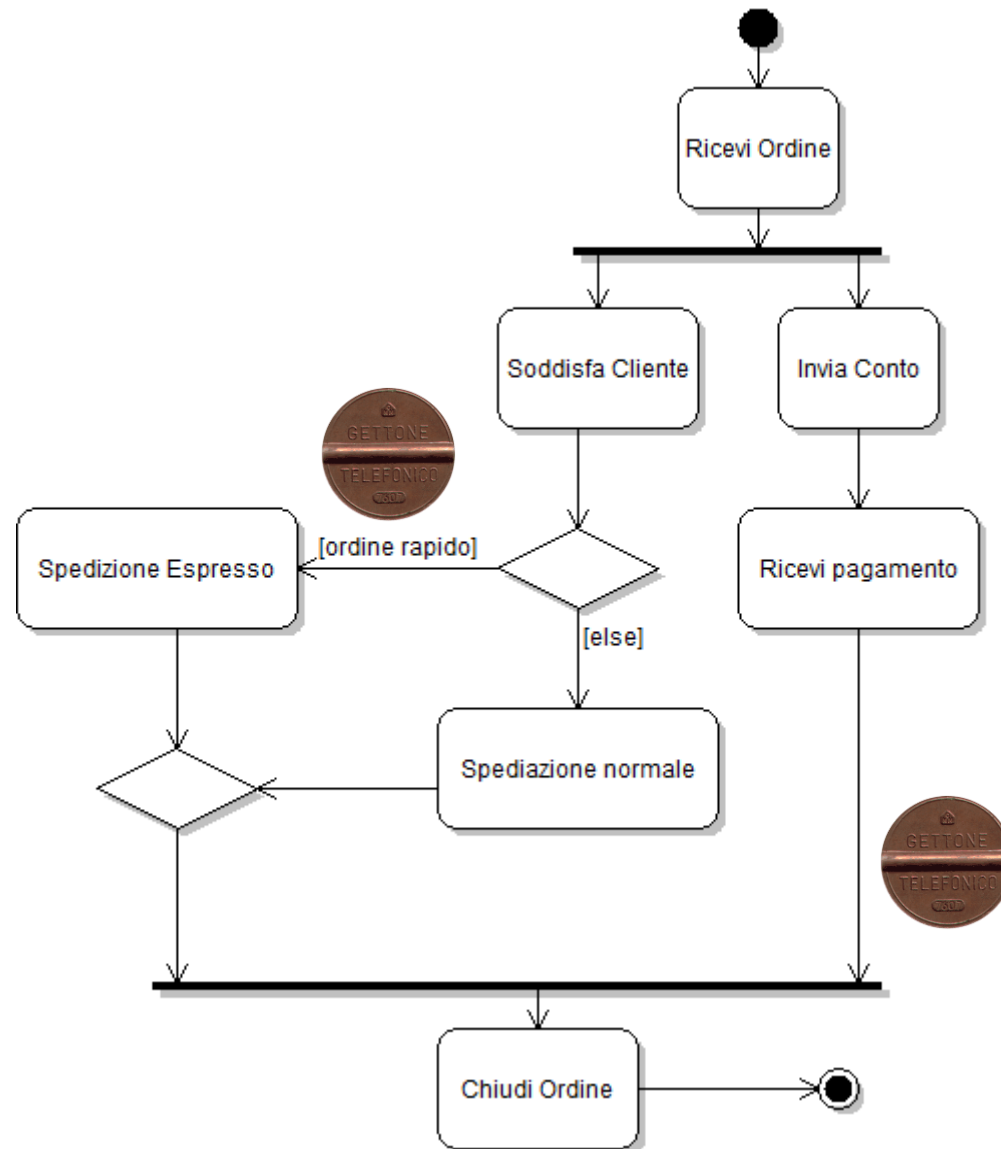


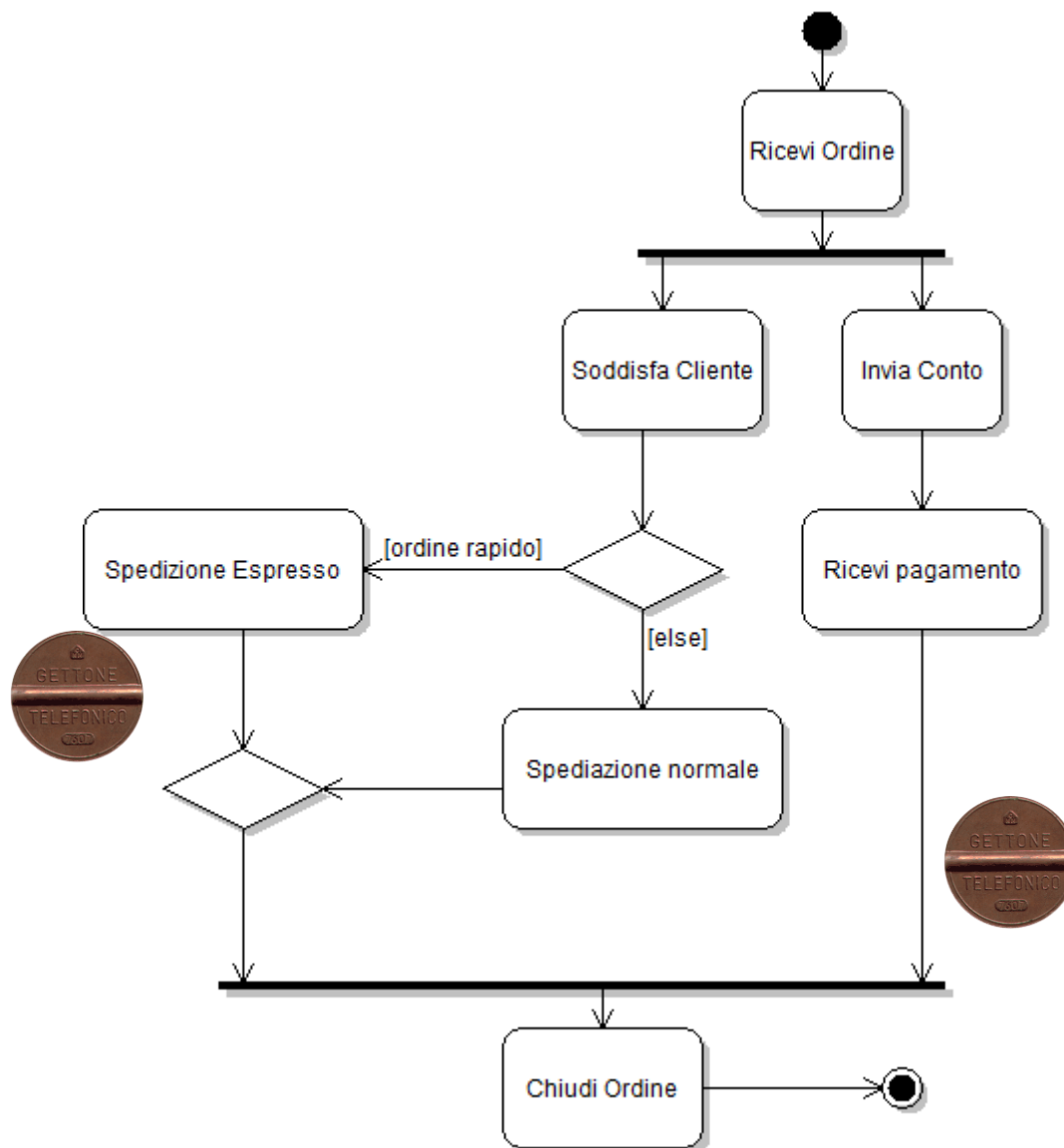
- Merce trovata
- Pacco preparato

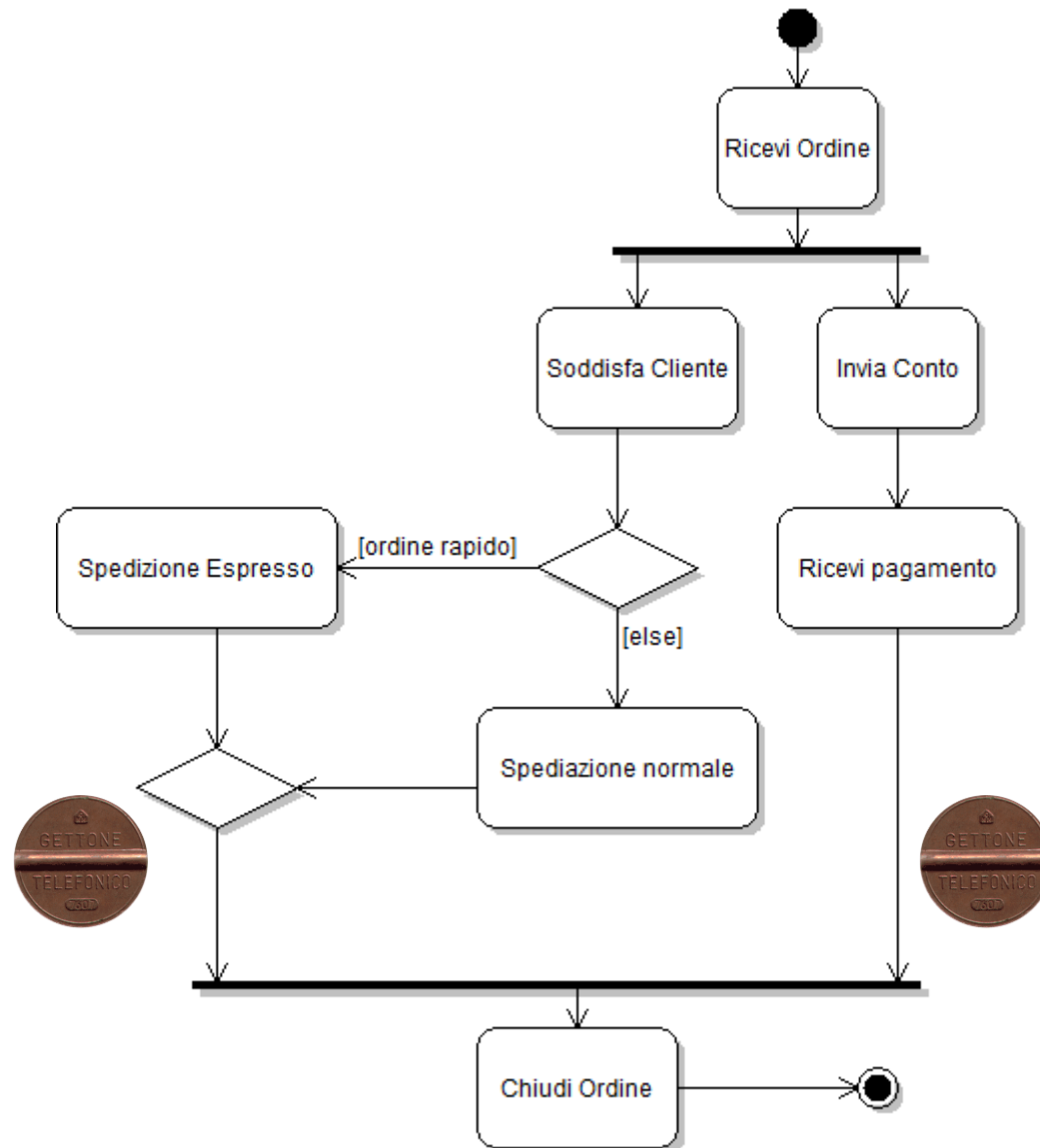


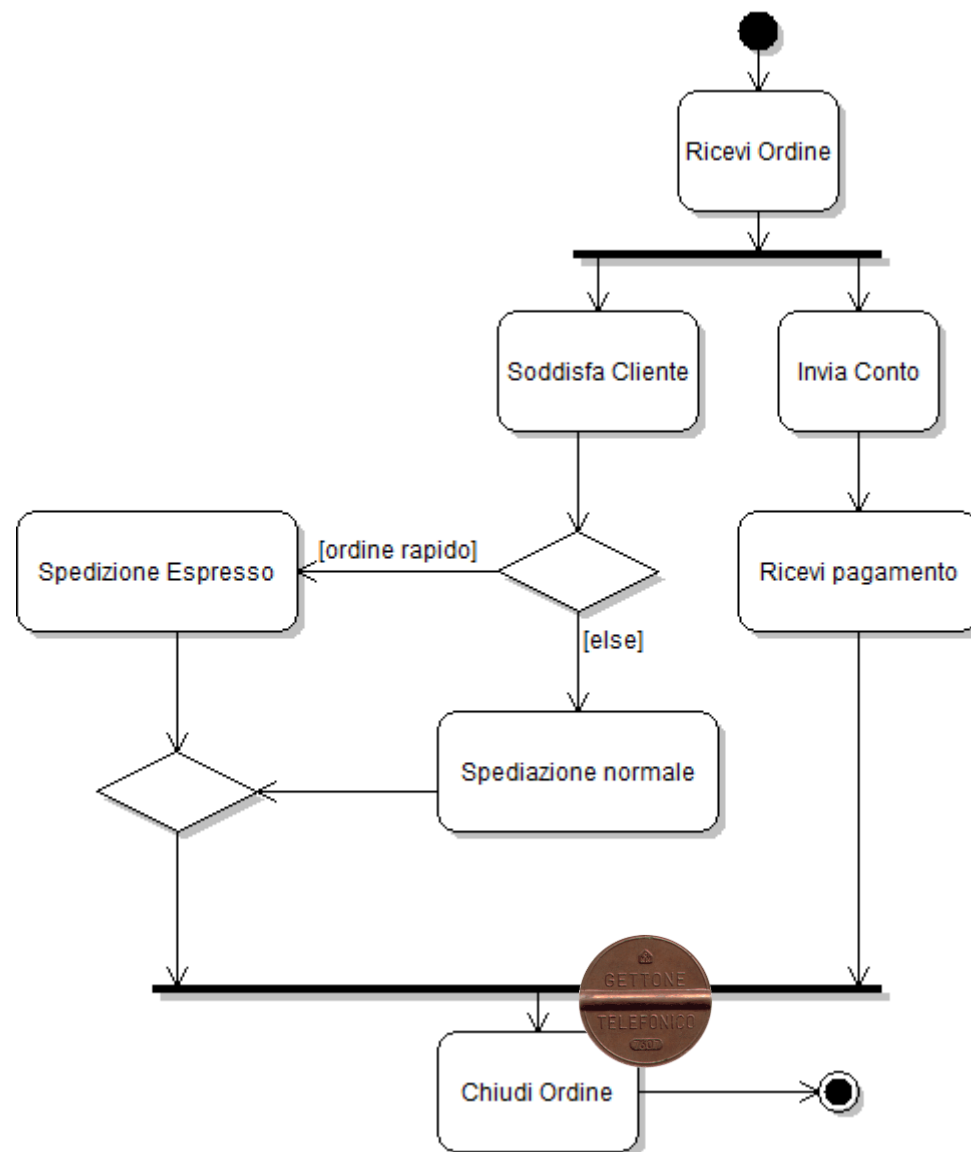


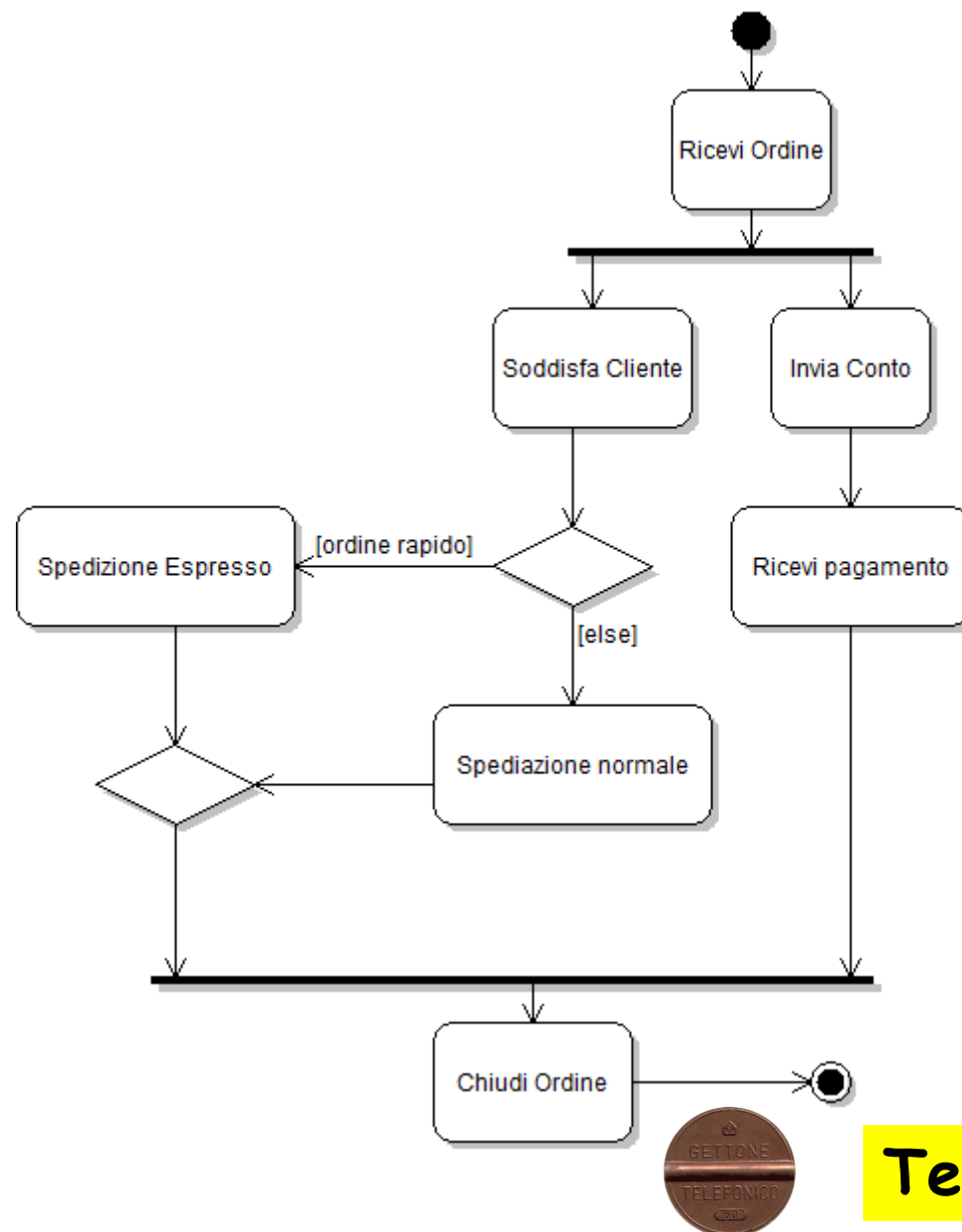








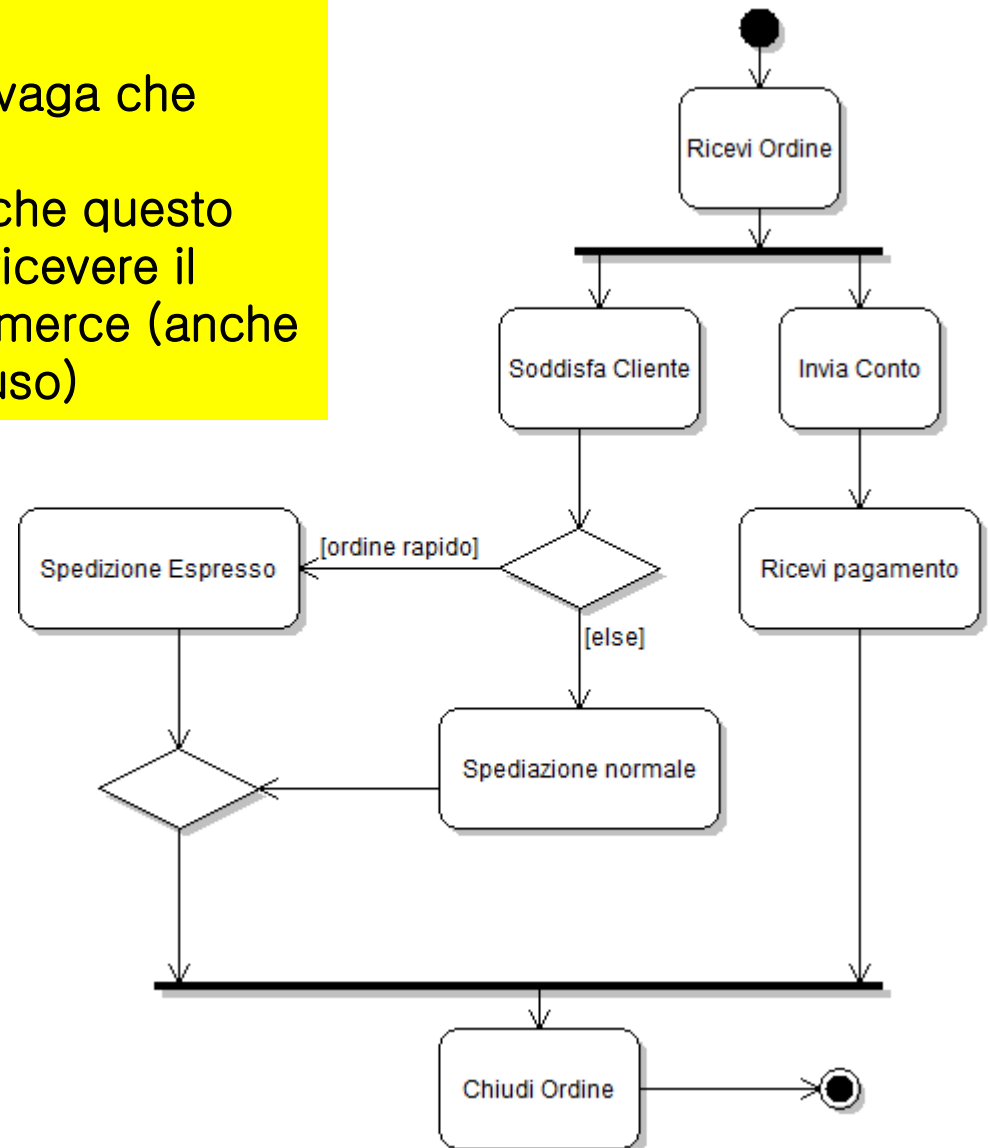




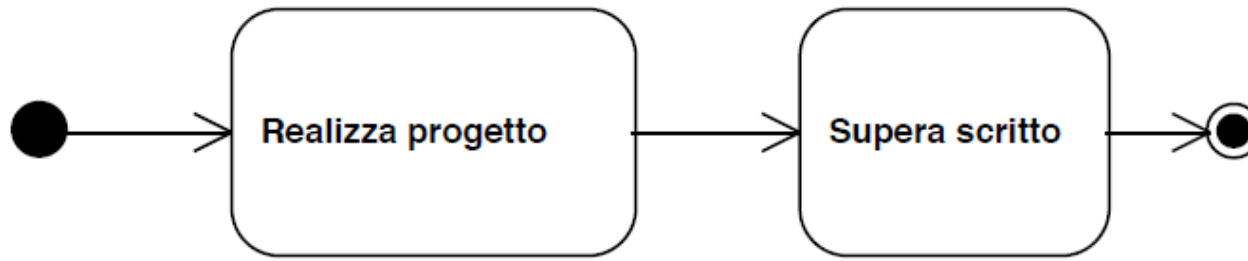
Termina!

CONSIDERAZIONI

- Manca un azione di preparazione del conto/fattura
- Soddisfa cliente è un attività vaga che andrebbe specificata meglio
- Bisogna essere consapevoli che questo Activity diagram permette di ricevere il pagamento senza spedire la merce (anche se poi l'ordine non viene chiuso)

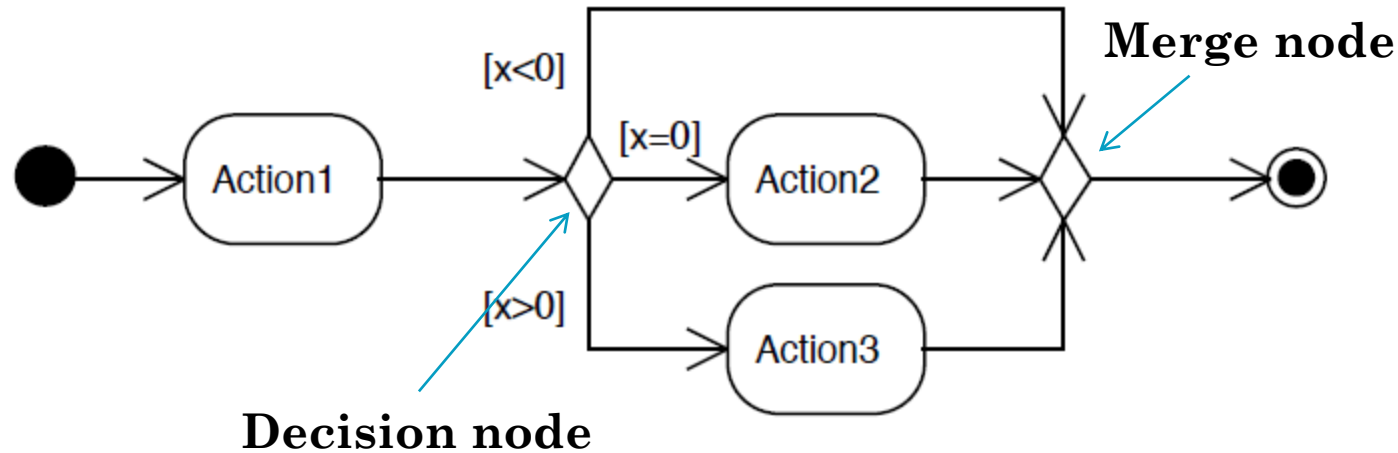


NODI INIZIALI E FINALI (DI ATTIVITÀ)



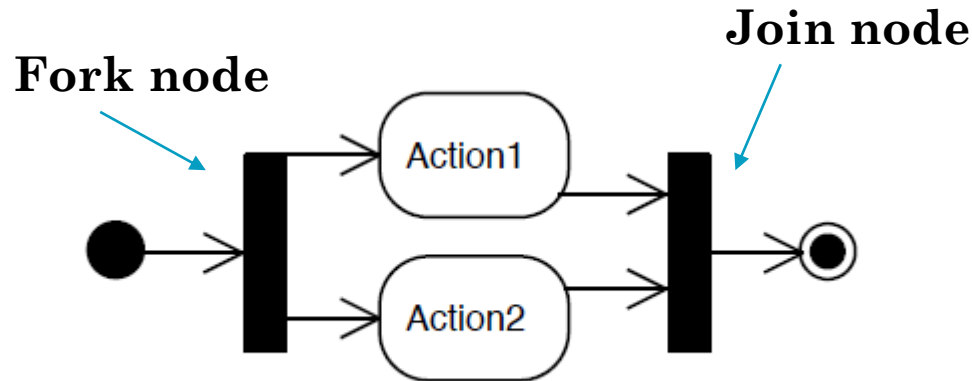
- Il disco nero marca l'inizio dell'attività
 - genera un token
- Quando un token raggiunge un disco nero bordato (**nodo finale di attività**), l'attività ha termine
 - distrugge tutti i token
- In un diagramma deve esistere **un nodo iniziale e zero o più nodi finali**
 - Il nodo iniziale fa partire un flusso di esecuzione, il primo nodo finale raggiunto ferma tutti i flussi

NODI DECISIONE E FUSIONE



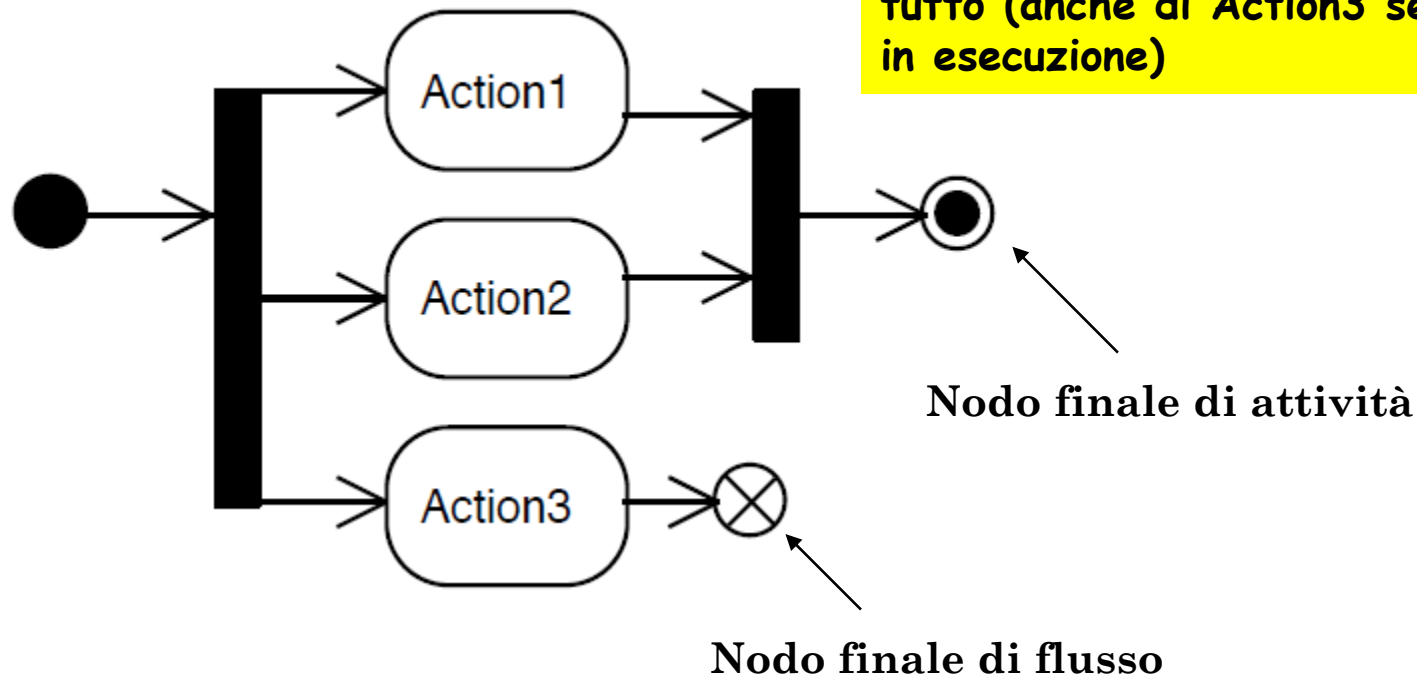
- I **nodi decisione** hanno un input e vari output
 - copiano i token in entrata su uno degli output a seconda della **guardia**
 - se più di una guardia è vera: **scelta non deterministica**
 - se nessuna guardia è vera: l'attività si blocca
- I **nodi fusione (merge)** hanno vari input e un solo output, sul quale vengono indirizzati i token in ingresso

NODI FORK/JOIN



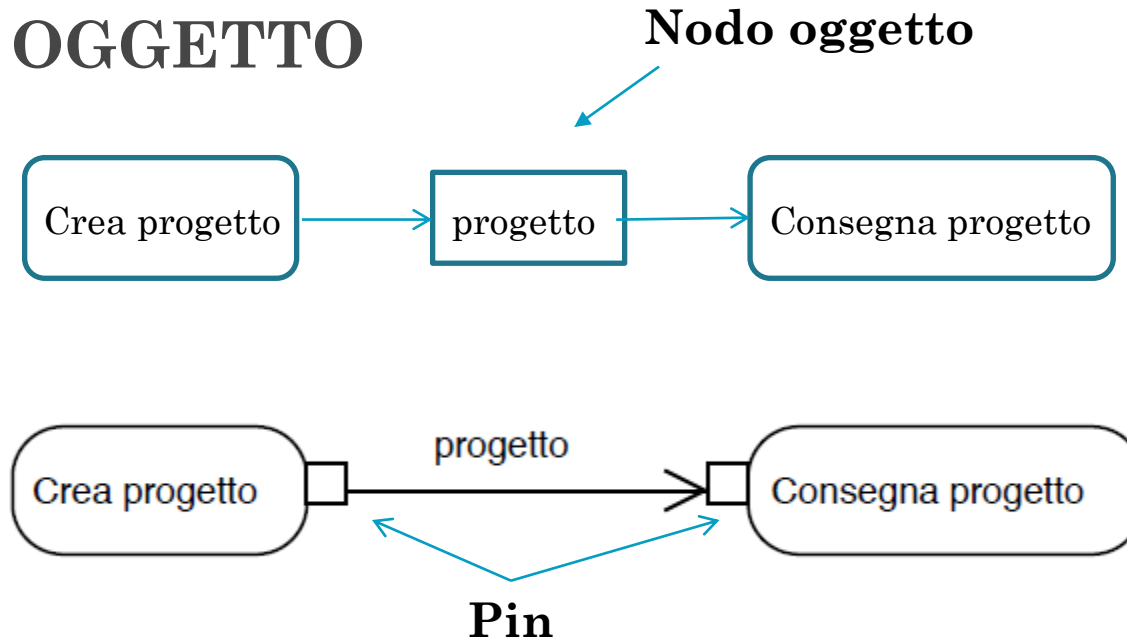
- I nodi **Fork** dividono un'esecuzione in più flussi concorrenti
- I nodi **Join** sincronizzano e riuniscono i flussi
- I **nod**i **Fork** hanno un ingresso e varie uscite
 - i token in ingresso sono duplicati su tutte le uscite
- I **nod**i **Join** hanno vari ingressi e una sola uscita
 - Quando sono presenti token su tutti gli ingressi, viene prodotto un token in uscita
 - L'attività è ferma ad attendere tutti i token

NODI FINALI DI FLUSSO



- Quando raggiunti da un token, causano la terminazione solo del flusso che li ha toccati
- Il raggiungimento di un **nodo finale di attività** causa comunque la terminazione di tutti i flussi

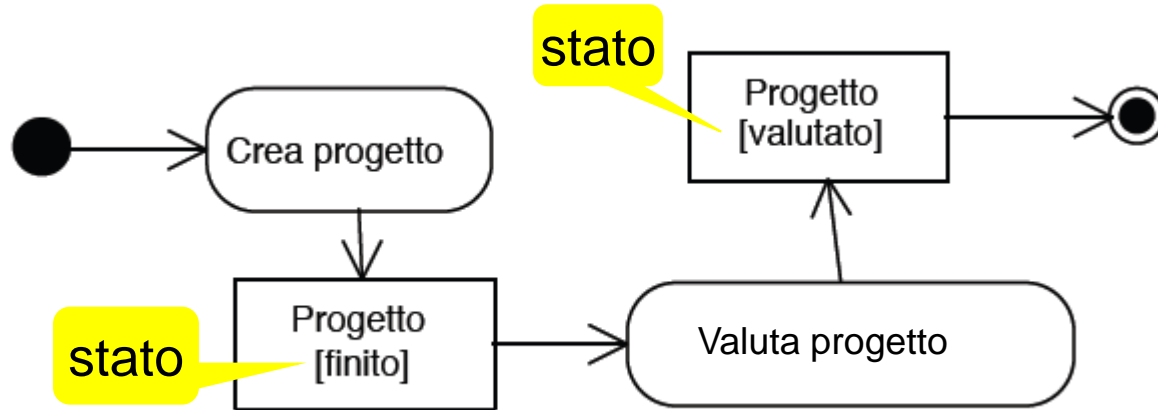
NODI OGGETTO



- Servono per **modellare gli oggetti in input e output delle azioni**
- Rappresentano veri e propri oggetti
 - **Prospettiva concettuale**: oggetti del mondo reale
 - **Prospettiva software**: oggetti OO

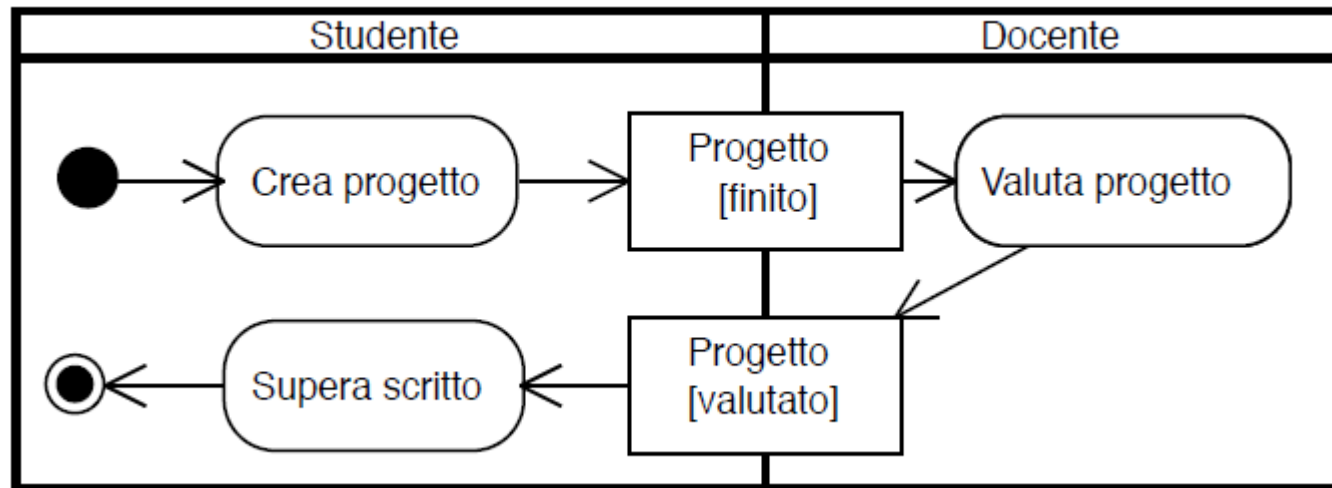
```
Progetto progetto = CreaProgetto()  
ConsegnaProgetto(progetto)
```


STATO DEGLI OGGETTI



- Spesso risulta conveniente aggiungere lo **stato di un oggetto** per mostrarne l'evoluzione durante l'attività
- Gli stati devono essere coerenti con la macchina a stati associata all'oggetto
 - Se presente nel modello ...

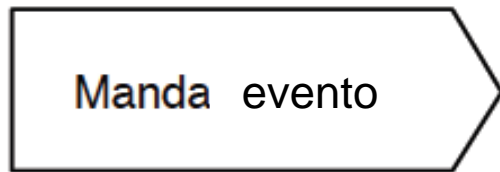
PARTIZIONI (SWIMLANES)



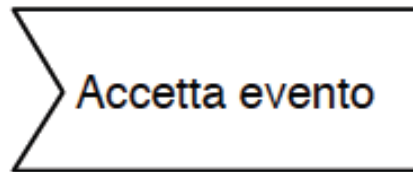
- Specificano chi esegue le azioni
 - Chi è il **partecipante**
- Suddividono il flusso dell'attività, ma non ne modificano il significato
 - **Rendono il diagramma + informativo**
- La suddivisione può essere orizzontale o verticale

EVENTI (1)

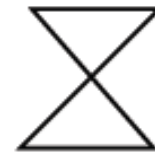
- Ci sono alcuni **nodi azione specializzati** che gestiscono **l'invio e la ricezione di eventi**
 - Anche eventi temporali
 - 25 ottobre 2019



Ad un'altra entità



Da un'altra entità

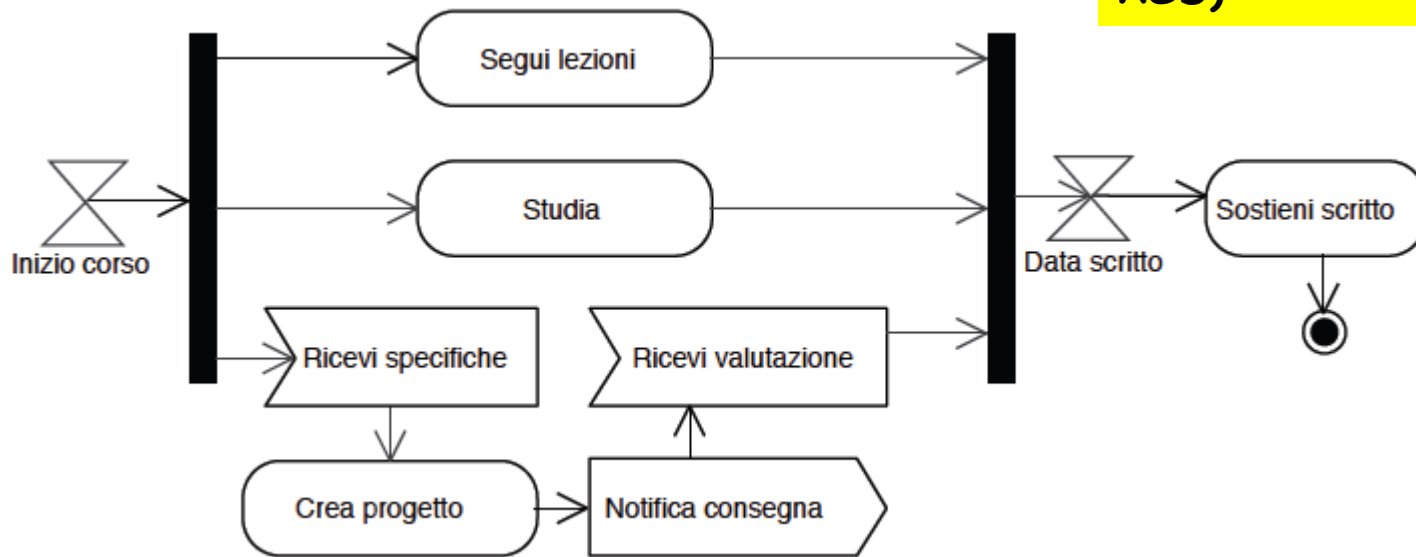


Accetta evento temporale

- L'invio di eventi è **asincrono e non blocca l'attività**. La ricezione blocca l'attività finché l'evento non si verifica

EVENTI (2)

Modella le azioni che deve compiere uno studente per sostenere un esame (es. ASD)

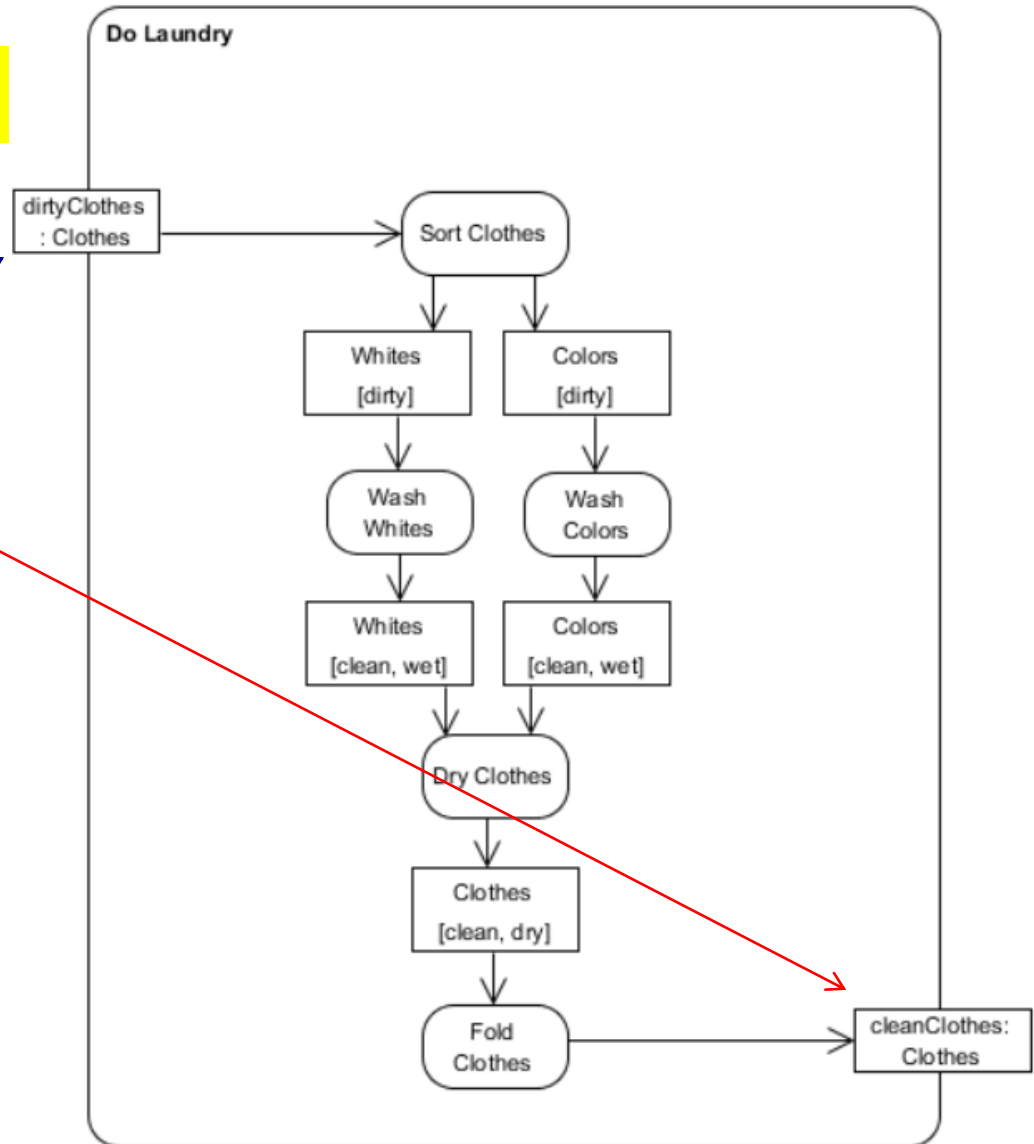


- I **nodi ricezione** sono attivi quando hanno token su tutti gli archi in entrata (se ne hanno) oppure durante l'intera vita dell'attività (se non ne hanno)
 - generano token alla ricezione
- La ricezione di eventi temporali funziona nello stesso modo, i token sono generati in base ad un'espressione temporale
 - Es. il 24 settembre 2020

PARAMETRI E VALORI DI RITORNO

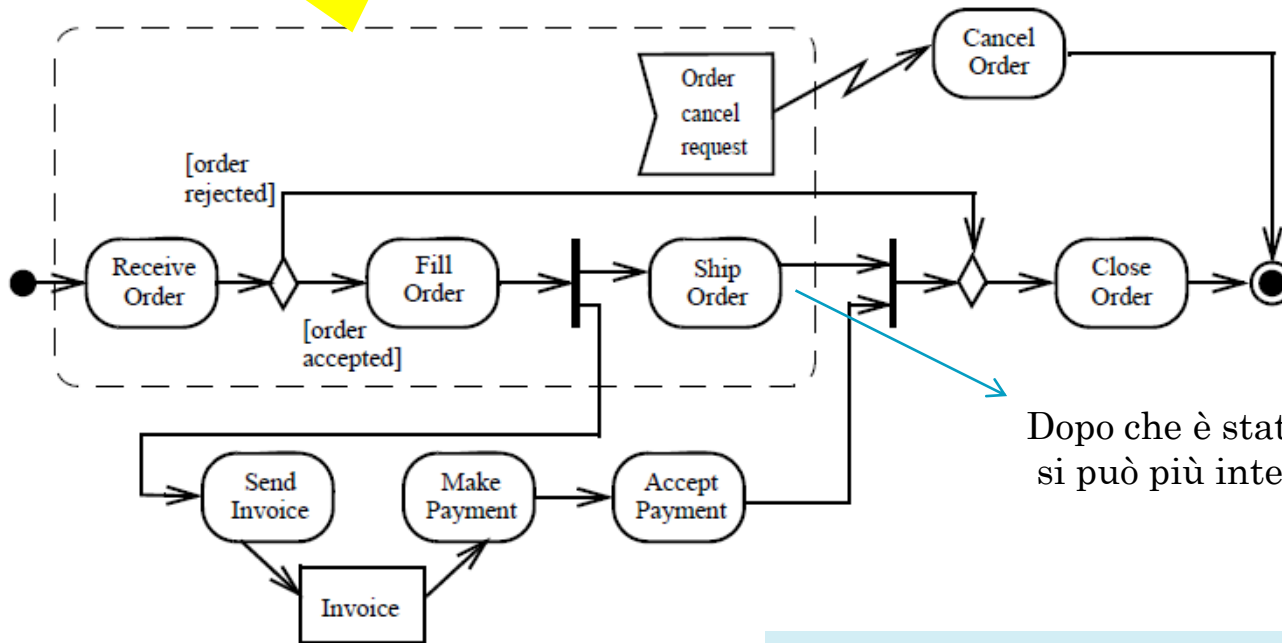
Attività: 'Fare bucato'

- Parametri di input e valori di ritorno, se esistono, si rappresentano come nodi oggetto sul bordo dell'attività



Limiti della
regione
interrompibile

REGIONI INTERRUPIBILI



Dopo che è stata fatta la consegna non si può più interrompere l'ordine

L'ordine è cancellato solo se un token si trova all'interno della regione al momento della ricezione del segnale

- Si usano per specificare l'interruzione dell'attività **forzata** in una regione
 - Esempi: **eccezioni**, **interrupt**, **situazioni di errore**

MODELLARE USE CASE (1)

ATM: Use case "prelevare contante"

Breve descrizione: Il sistema permette al cliente di **prelevare un certo ammontare** in contanti dal suo c/c

Attori primari: Cliente

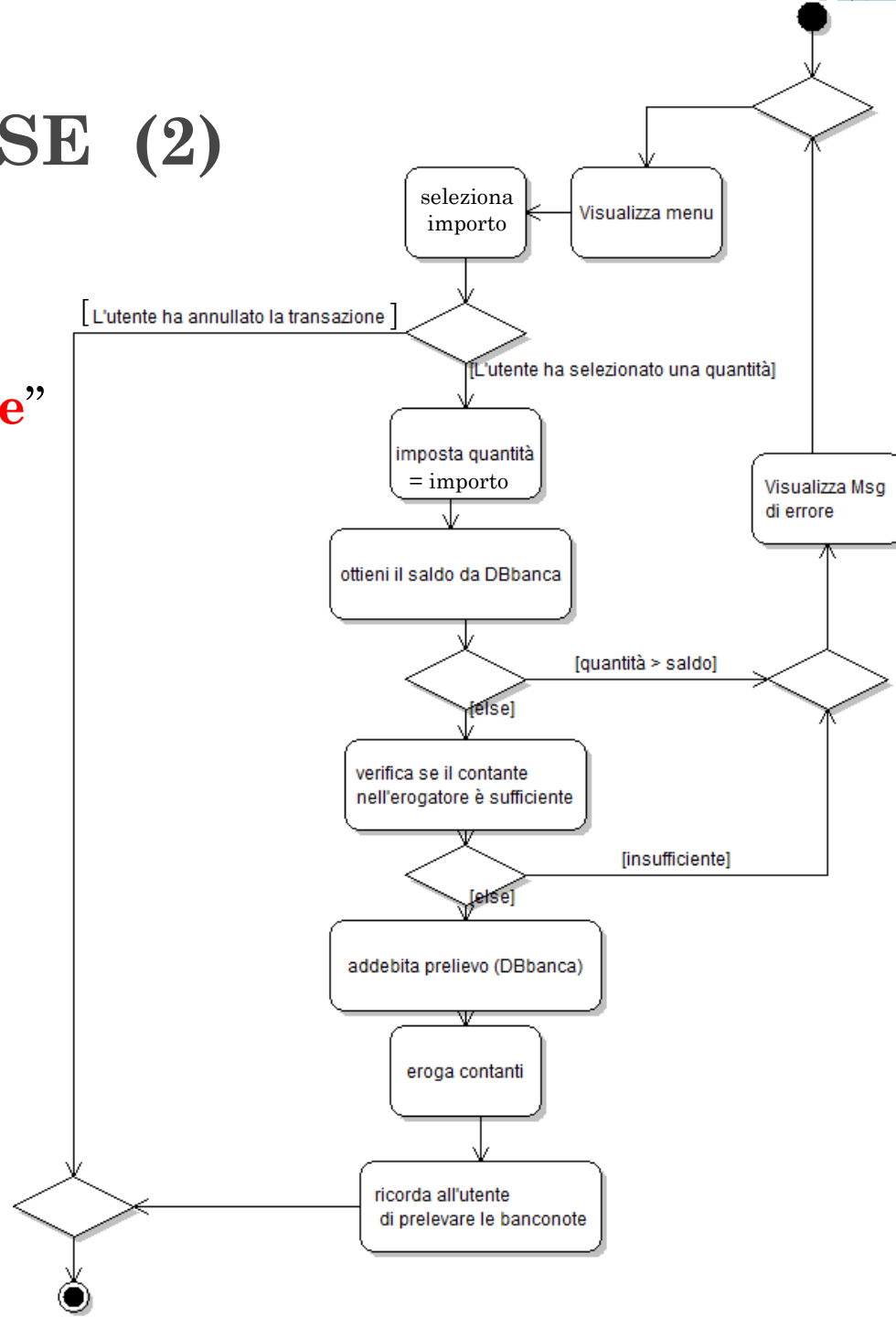
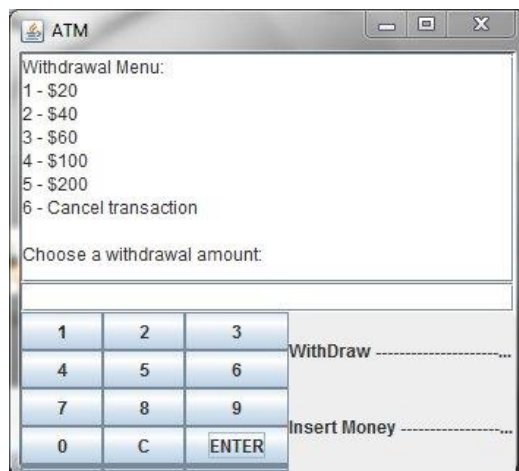
Sequenza degli eventi principale:

1. Il Sistema visualizza un menu con alcune opzioni: 20\$, 40\$, 60\$, 100\$, 200\$, cancel
2. Il Cliente seleziona un ammontare usando la tastiera
3. **Se** L'ammontare non supera il saldo attuale del c/c del cliente **e** non supera l'ammontare disponibile nello sportello
 - 3.1 Il Sistema addebita l'ammontare sul c/c (lo sottrae al saldo attuale)
 - 3.2 Il Sistema eroga le banconote
 - 3.3 Il Sistema visualizza un messaggio per ricordare al cliente di prendere le banconote
4. **Altrimenti**
 - 4.1 Il Sistema visualizza un messaggio (selezionare un ammontare inferiore o soldi non disponibili nello sportello) e torna al punto 1

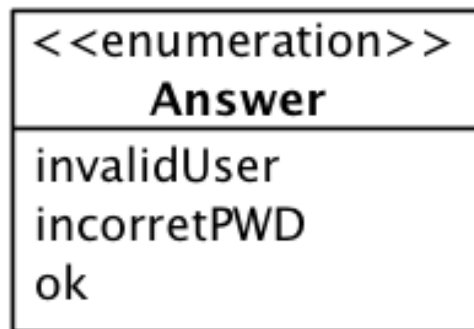
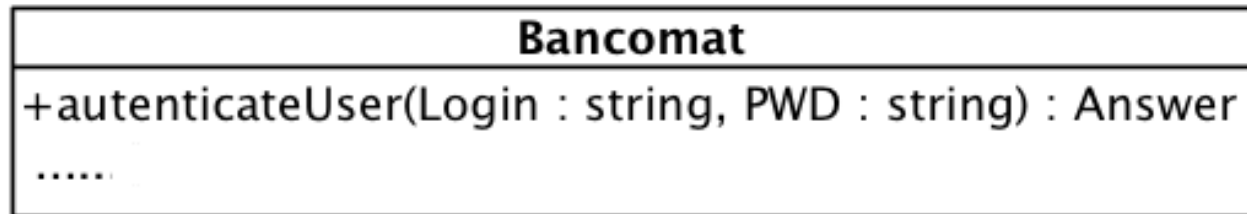
MODELLARE USE CASE (2)

ATM

- Use case “**prelevare contante**”

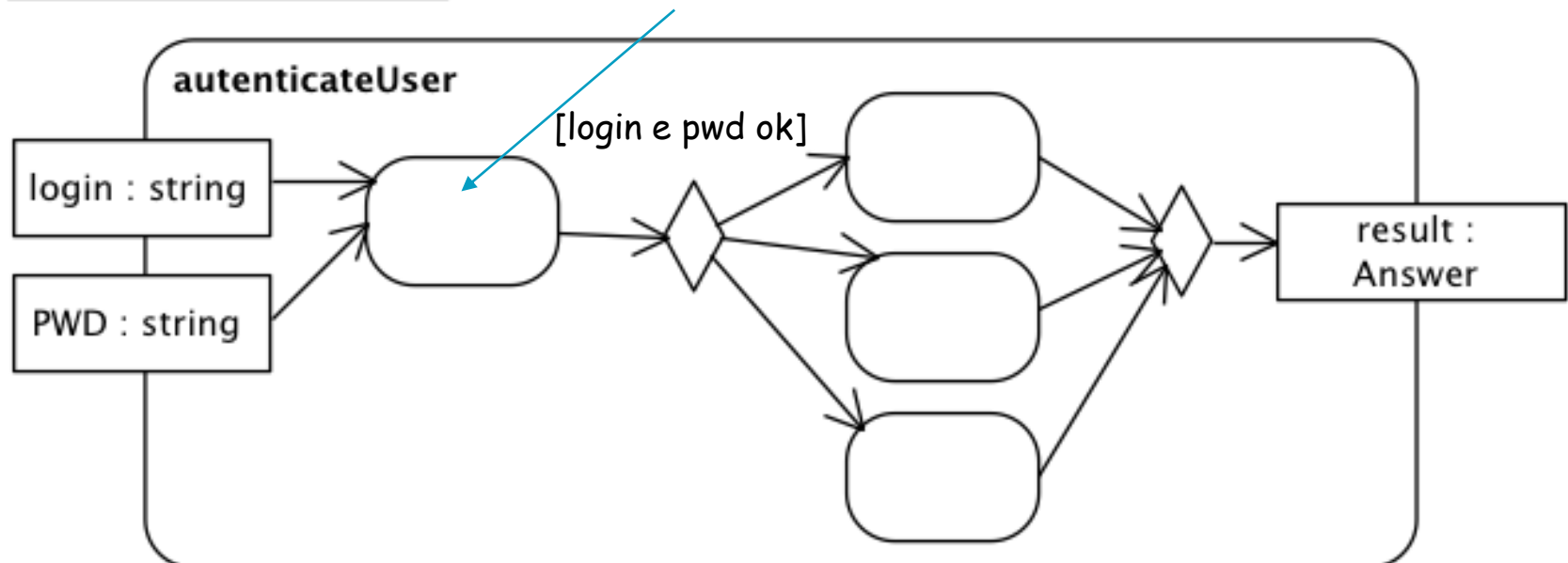


MODELLARE OPERAZIONI



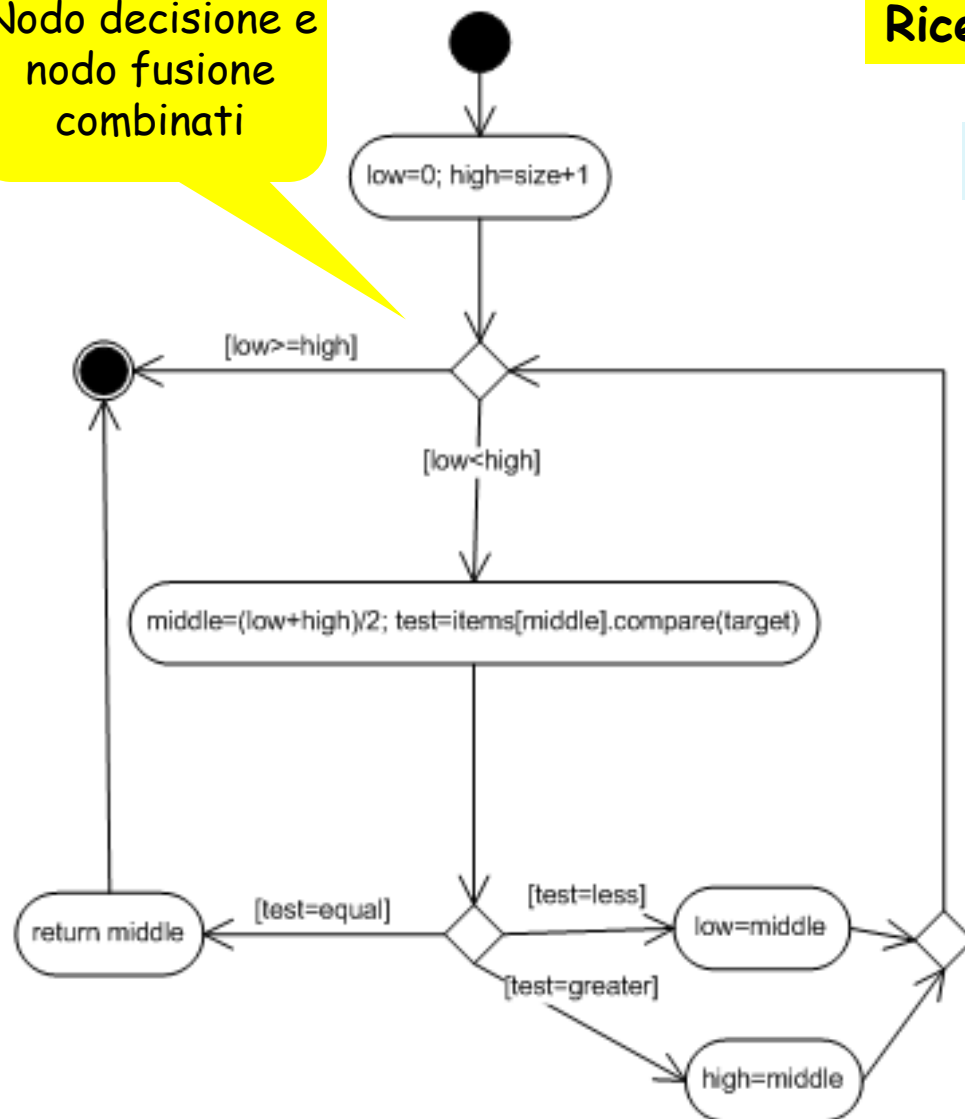
**Esercizio per casa:
completare!**

Cerca login e PWD nel DB



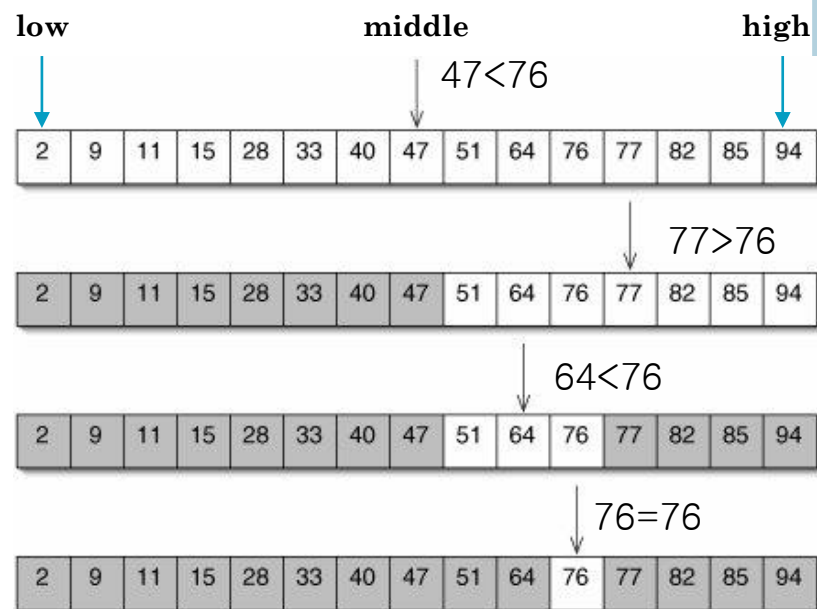
MODELLARE UN ALGORITMO

Nodo decisione e
nodo fusione
combinati



Ricerca binaria in un array ordinato

Target = 76

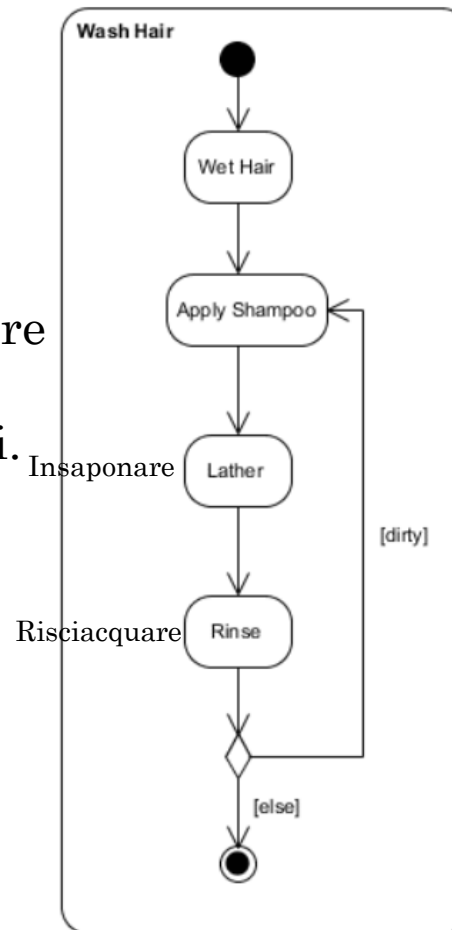


DOMANDA QUIZ SBARRAMENTO

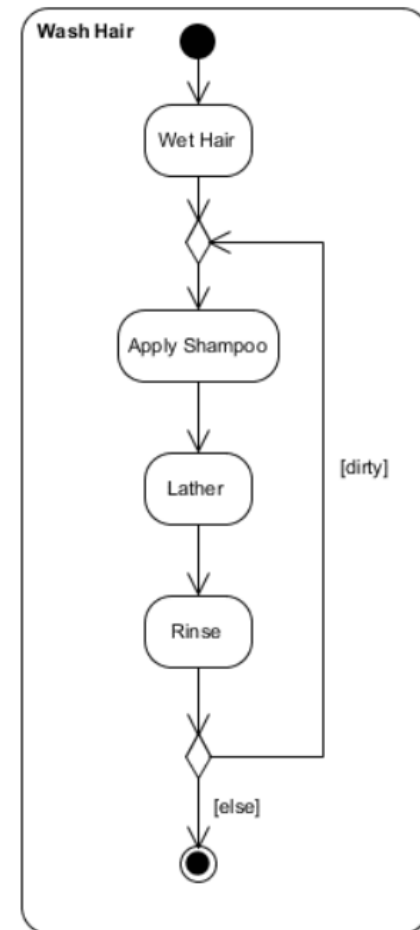
- Confrontare i seguenti activity diagram. Quale delle seguenti affermazioni è vera?

- ☐ I due diagrammi sono equivalenti, cioè esprimono la stessa attività
- ☐ Il merge node tra le azioni 'Wet Hair' e 'Apply Shampo' nel diagramma Due è superfluo. Si può eliminare e collegare il flusso a 'Apply shampo'
- ☐ I due diagrammi non sono equivalenti. L'attività Uno si blocca sull'azione 'Apply shampo'
- ☐ Entrambi i diagrammi non sono corretti logicamente perchè le azioni 'Lather' (insaponare) e 'Rinse' (risciacquare) nella realtà sono eseguite in parallelo

Uno

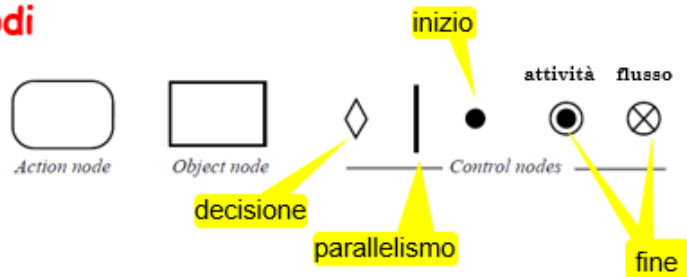


Due

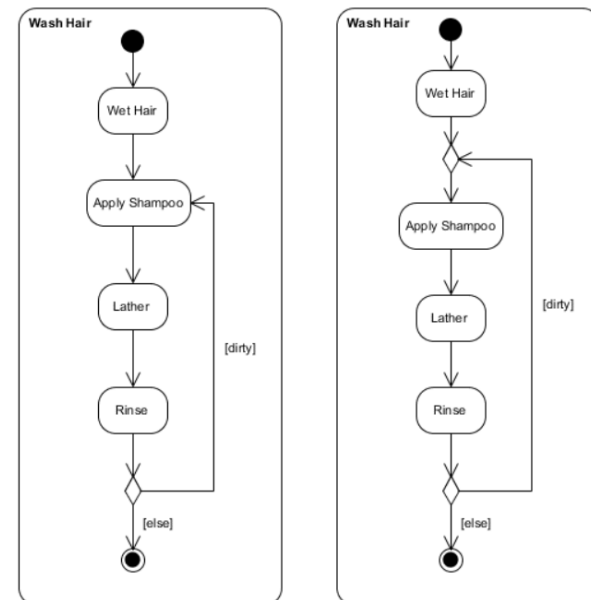


RIASSUMENDO

Nodi



Diagrammi poliedrici!

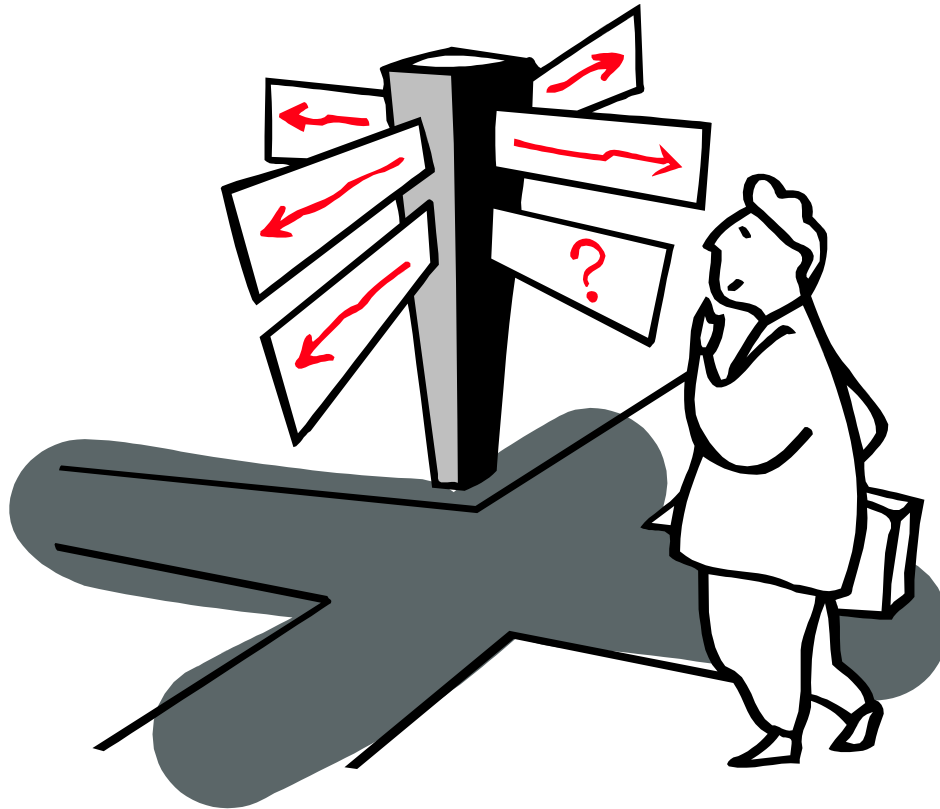


MATERIALE E RIFERIMENTI

- Sugli Activity Diagram c'è **molto molto** di più
 - Vedere:
 - UML 2 Activity and action Models, Journal of Object Technology 2003, Conrad Bock
- Per realizzare la seguente presentazione sono stati utilizzati:
 - UML distilled M. Fowler
 - Le slide di Angelo di Iorio (UniBo) AA. 2010-2011



THE END ...



Domande?