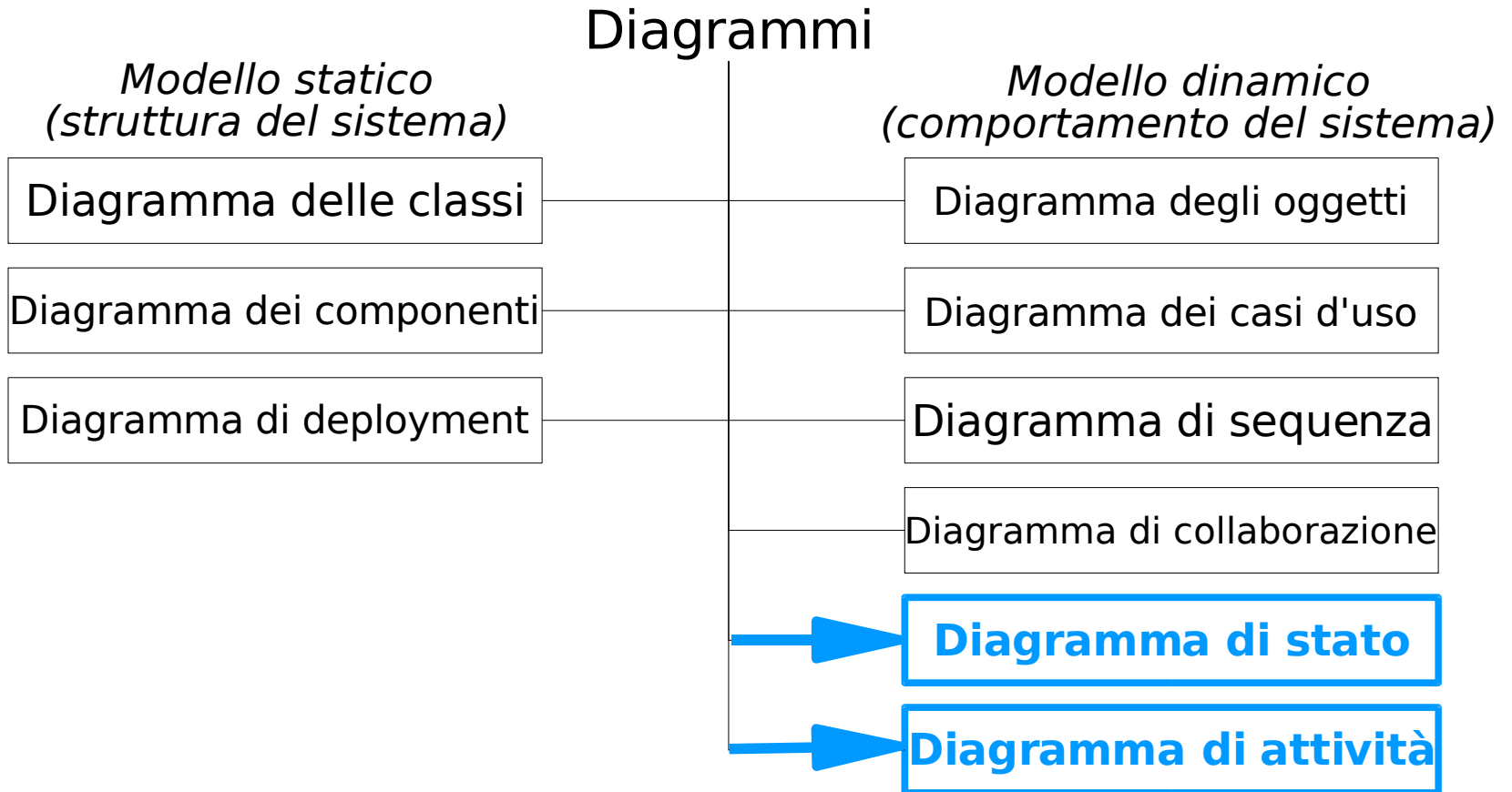


Elementi di UML (6): Diagrammi dinamici di flusso

Università degli Studi di Bologna
Facoltà di Scienze MM. FF. NN.
Corso di Laurea in Scienze di Internet
Anno Accademico 2004-2005

Laboratorio di Sistemi e Processi Organizzativi

Diagrammi di flusso dinamico



Diagrammi di flusso

- I **diagrammi di stato** e di **attività** sono diagrammi di flusso OO
- Consentono di modellare un processo o un'entità tramite
 - Stati
 - Transizioni tra stati



Diagrammi di attività

- I diagrammi di attività
 - Sono un tipo speciale di *diagrammi di stato* in cui
 - gli *stati* sono stati d'azione
 - le *transizioni* sono attivate automaticamente al completamento delle azioni e delle attività di uno stato
 - Utilizzano un piccolo sottoinsieme della sintassi UML per i diagrammi di stato

Entità reattiva

Una entità reattiva

- È un oggetto (in senso lato) che fornisce il contesto ad un *diagramma di stato*
- Risponde a eventi esterni
- Ha un ciclo di vita definito, che può essere modellato come successioni di stati e transizioni
- Ha un comportamento corrente che dipende dai comportamenti precedenti

Diagrammi di attività vs stato

- I **diagrammi di attività** vengono usati per modellare processi a cui partecipano diversi oggetti
- I **diagrammi di stato** vengono usati per modellare il ciclo di vita di una entità reattiva
 - Descrive la macchina a stati di una singola entità
 - Modella il comportamento dinamico di:
 - Classi
 - Casi d'uso
 - Sottosistemi
 - Sistemi interi


Diagrammi di stato

Macchine a stati e classi

- Per ogni classe può esistere una macchina a stati che modella tutte le transizioni di stato di tutti gli oggetti di quella classe, in risposta a diversi tipi di evento
- Gli eventi sono tipicamente dei messaggi
 - inviati da altri oggetti
 - o generati internamente
- La macchina a stati di una classe modella il comportamento degli oggetti della classe in modo trasversale per tutti i casi d'uso interessati

Sintassi di base

- Gli stati sono rappresentati con rettangoli arrotondati

Stato: 

Stato iniziale: ●

Stato finale: 

- Le transizioni indicano un possibile percorso tra due stati e sono modellate con frecce
- Gli eventi
 - sono istantanei
 - sono scritti sopra la transizione che attivano

Stati

- The UML Reference Manual definisce uno stato come:

“una condizione o una situazione della vita di un oggetto durante la quale tale oggetto soddisfa una condizione, esegue un'attività o aspetta un qualche evento”

- Affinché sia rilevante modellare degli stati di una entità deve esistere tra tali stati una differenza semantica

Stati (sintassi)

nome dello stato

Immissione Password

azioni di ingresso e
di uscita

Entry/ visualizza password

Exit/ valida password

transizioni interne

tastoAlfabetico/ echo "*"

help/ visualizza help

attività interna

Do/ get

Sintassi dell'azione: evento/ azione

Sintassi dell'attività: Do/ attività

Stati: azioni

- Come visto nell'esempio precedente, uno stato può attivare delle azioni:
 - **Entry**: le azioni si attivano non appena si entra nello stato
 - **Do**: le azioni si svolgono nel ciclo di vita dello stato
 - **Evento**: le azioni avvengono in risposta ad un evento
 - **Exit**: le azioni si attivano appena prima di uscire dallo stato
 - **Include**: chiama una *submachine*, rappresentata da un altro diagramma degli stati

Transizioni



Quando avviene *unEvento*, se la *condizioneGuardia* risulta vera, allora esegue *unaAzione*, quindi entra immediatamente nello stato B

Eventi

- Un **evento** è la specifica di un'occorrenza di interesse che ha una collocazione nel tempo e nello spazio
- Quattro tipi:
 - **Evento di chiamata**: equivale a una richiesta di esecuzione di un insieme di azioni
 - **Evento di segnale**: ricezione di un segnale
 - **Evento di variazione**: l'azione associata viene eseguita quando l'espressione *Booleana* seguente alla parola chiave **quando** risulta vera
 - **Evento del tempo**: evento che viene attivato in determinati momenti del tempo, indicati con le parole chiave **quando** e **dopo**.

Eventi di segnale (1/3)

- Un **segnale** è un pacchetto di informazioni inviato in modo **asincorno** da un oggetto a un altro
- Il **segnale** viene modellato come una classe con stereotipo, la quale ha attributi corrispondenti a tutte le informazioni comunicate.
- Ad esempio:



Eventi di segnale (2/3)

- Poiché un **segnale** permette solo il passaggio di informazioni tra oggetti diversi, non può avere alcuna operazione
- Unica *eccezione* è fatta per l'operazione implicita

Invia (elencoDestinatari)

che consente l'invio del **segnale** ad uno o più oggetti destinatari

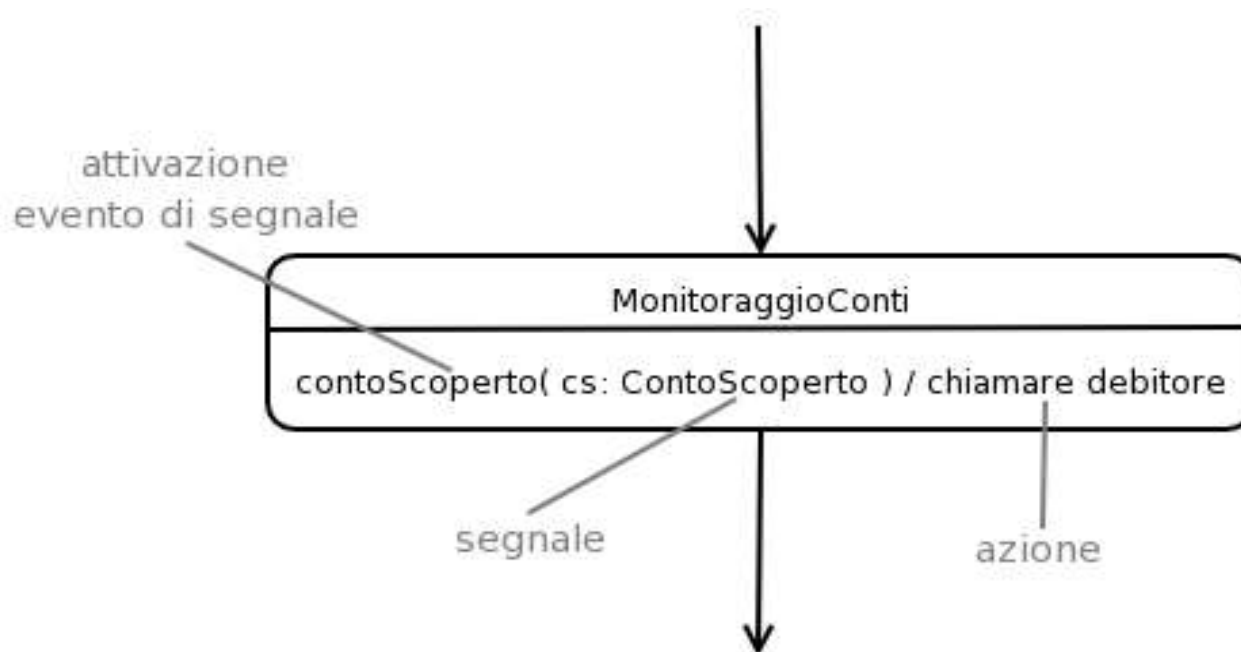
- La ricezione di un segnale da parte di un'entità reattiva può essere modellata come un **evento di segnale**

Eventi di segnale (3/3)

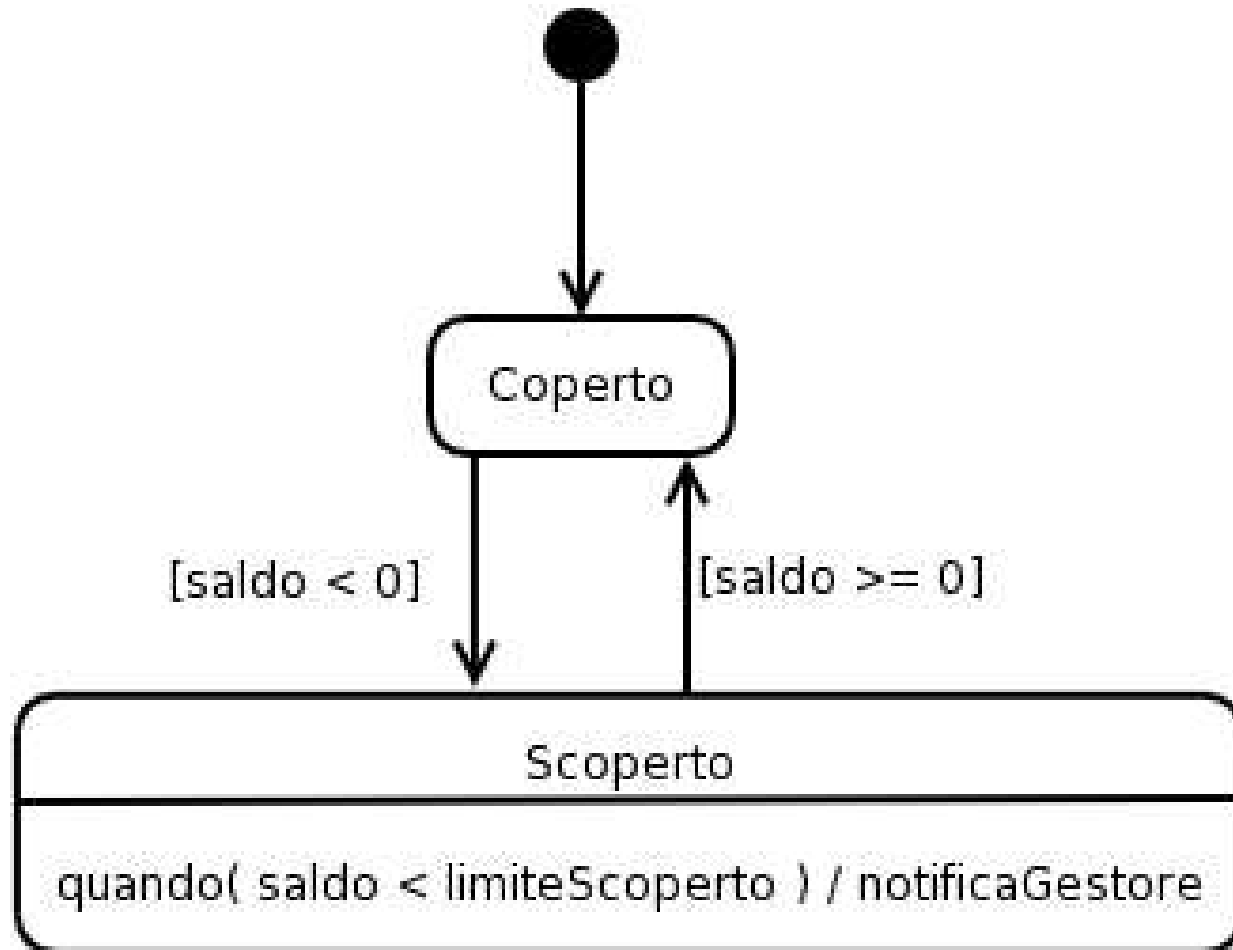
- L'attivazione di un **evento di segnale** è un *metodo* che prende come *parametro* quel tipo di **segnale**



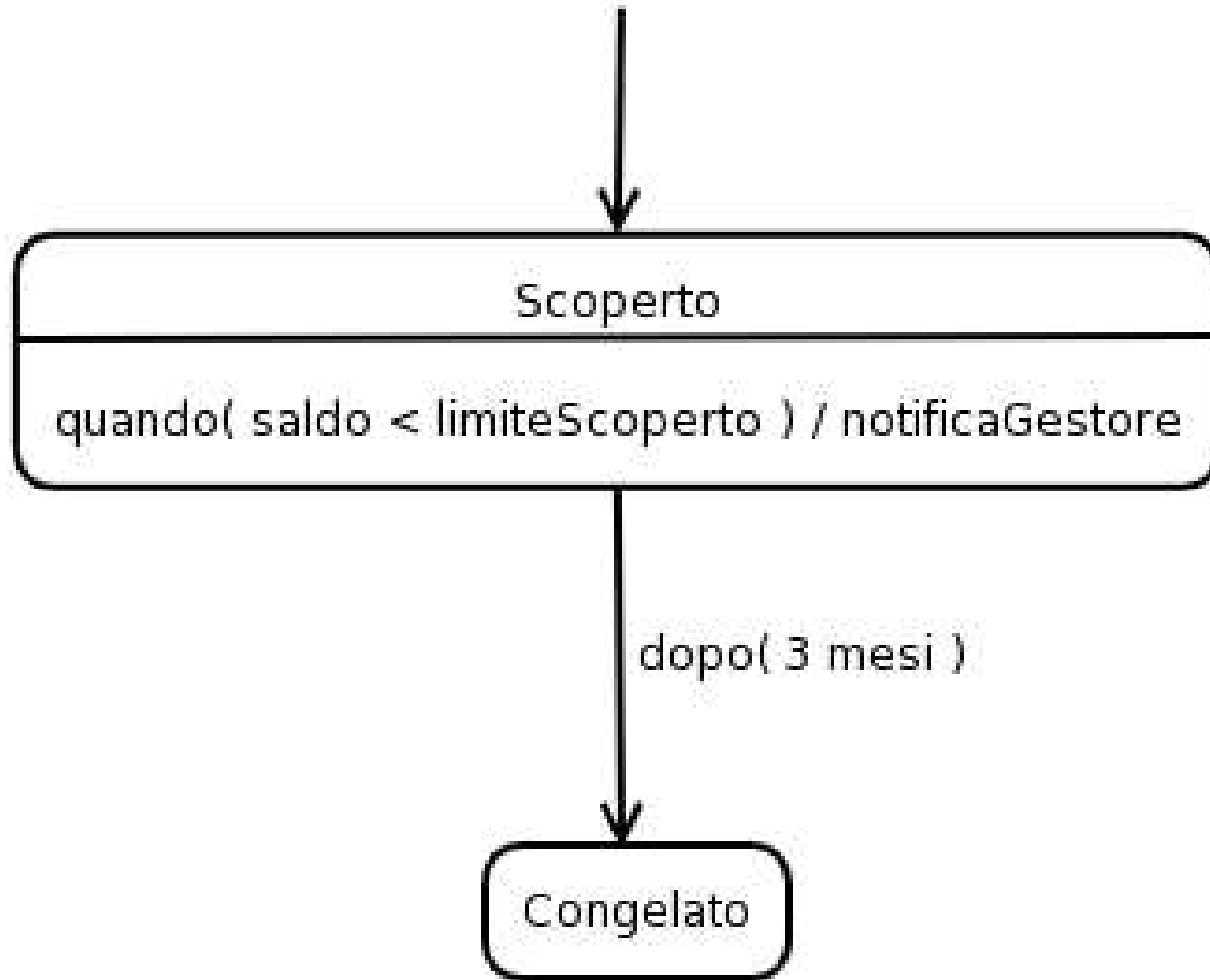
- Ad esempio dato



Eventi di variazione (esempio)

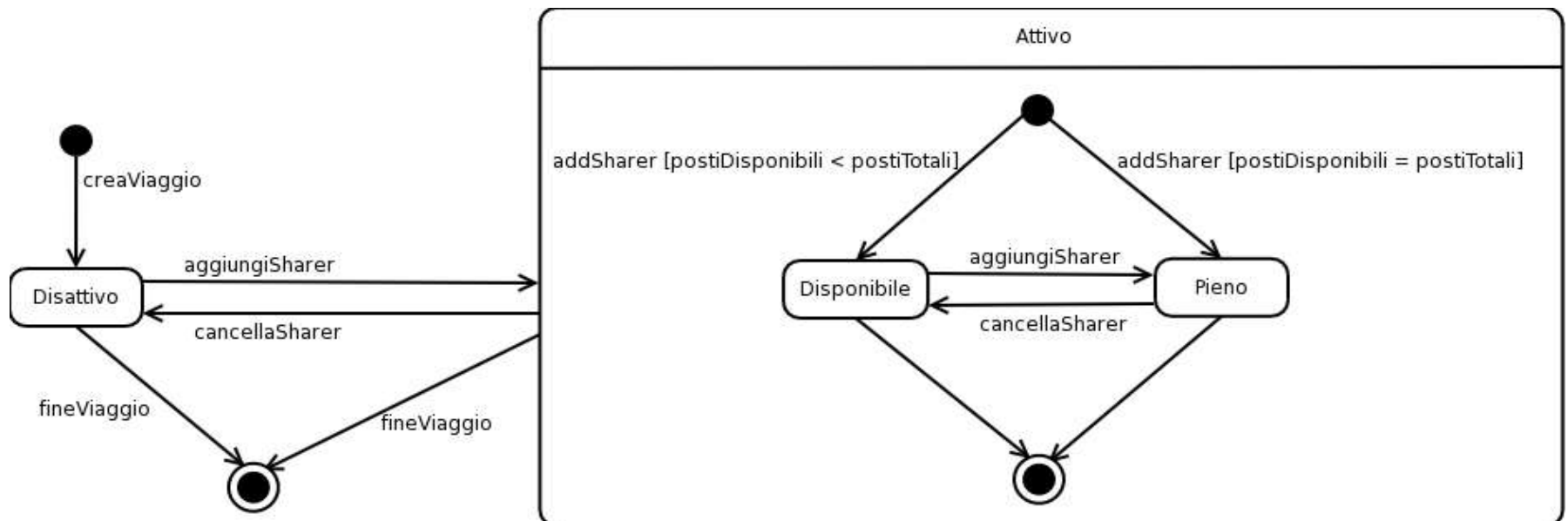


Eventi del tempo (esempio)

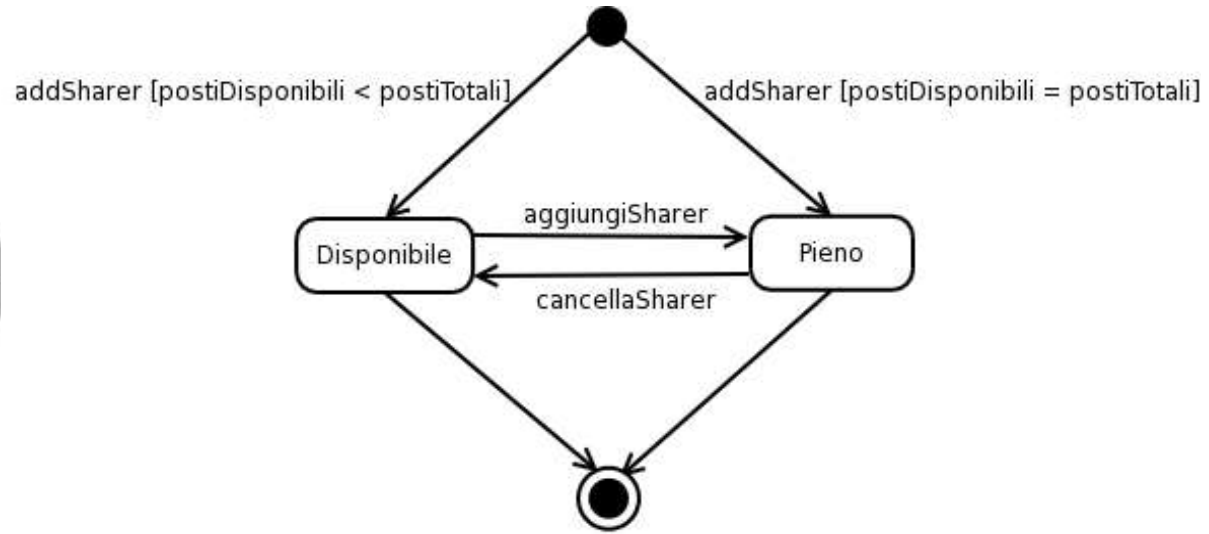
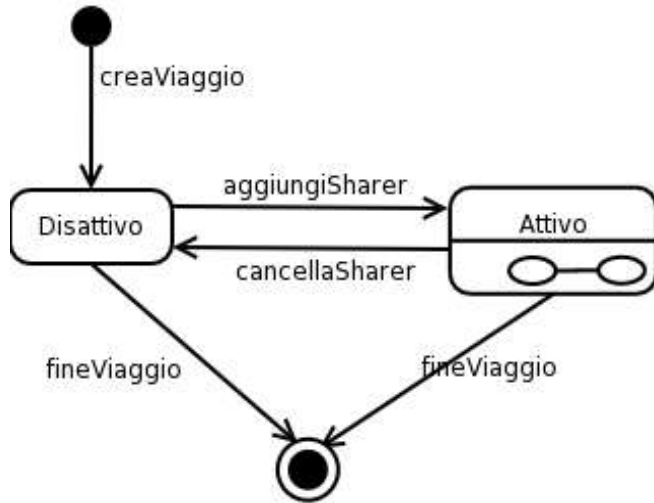


Stati composti (1/2)

- Uno stato composto è uno stato che può essere ulteriormente spezzato in sottostati
 - questi possono essere rappresentati all'interno del medesimo diagramma o in uno a parte
- Esempio di *state machine* della classe *Viaggio*



Stati composti (2/2)

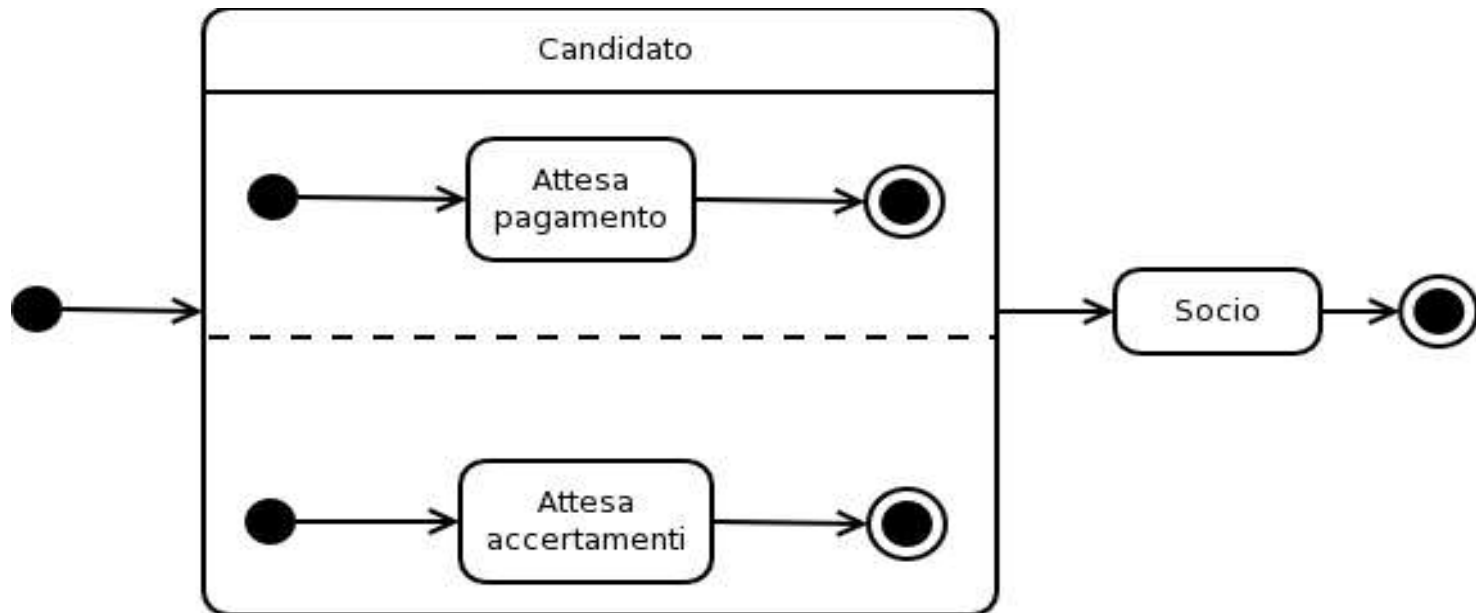


Sottoflusso dello stato Attivo

Notiamo che un'istanza di *Viaggio* che si trova nello stato **Attivo** deve anche trovarsi nello stato **Disponibile** oppure nello stato **Pieno**

Stati concorrenti

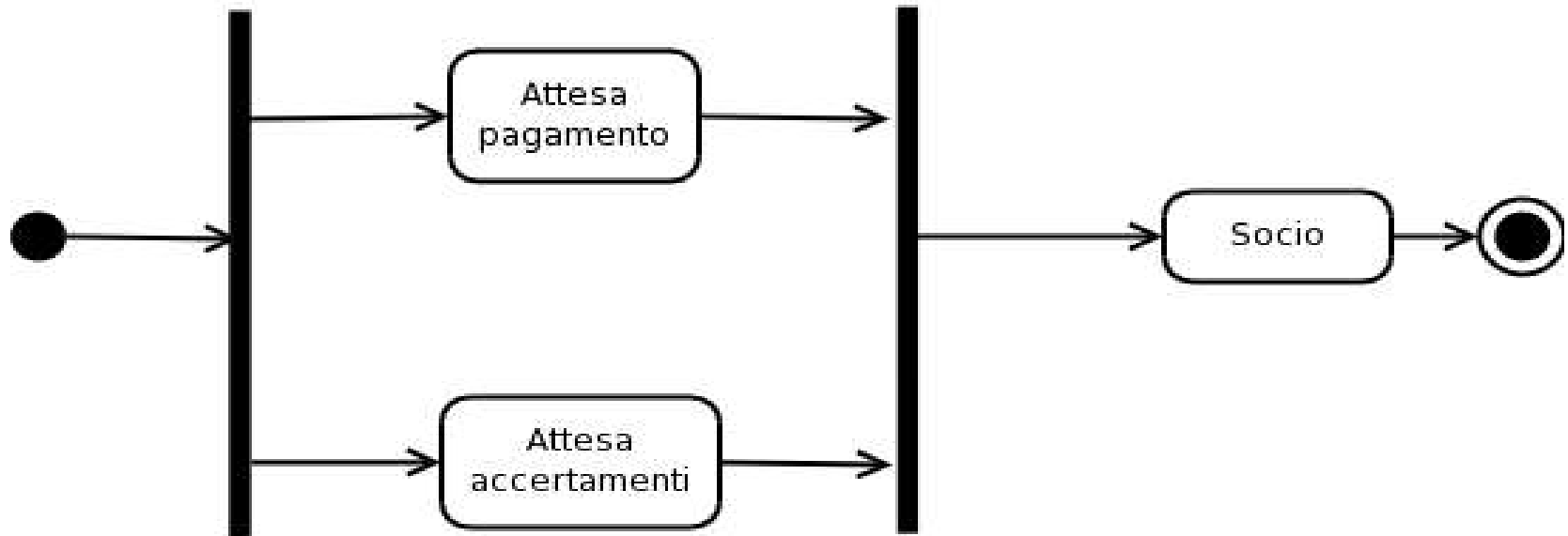
- È possibile che uno stato composto possieda più stati **concorrenti**
- Esempio di stati concorrenti per la classe *Cliente* in fase di registrazione



- Un'istanza di *Cliente* sarà sia nello stato *Attesa Pagamento* che in quello *Attesa accertamenti*

Linee di sincronizzazione

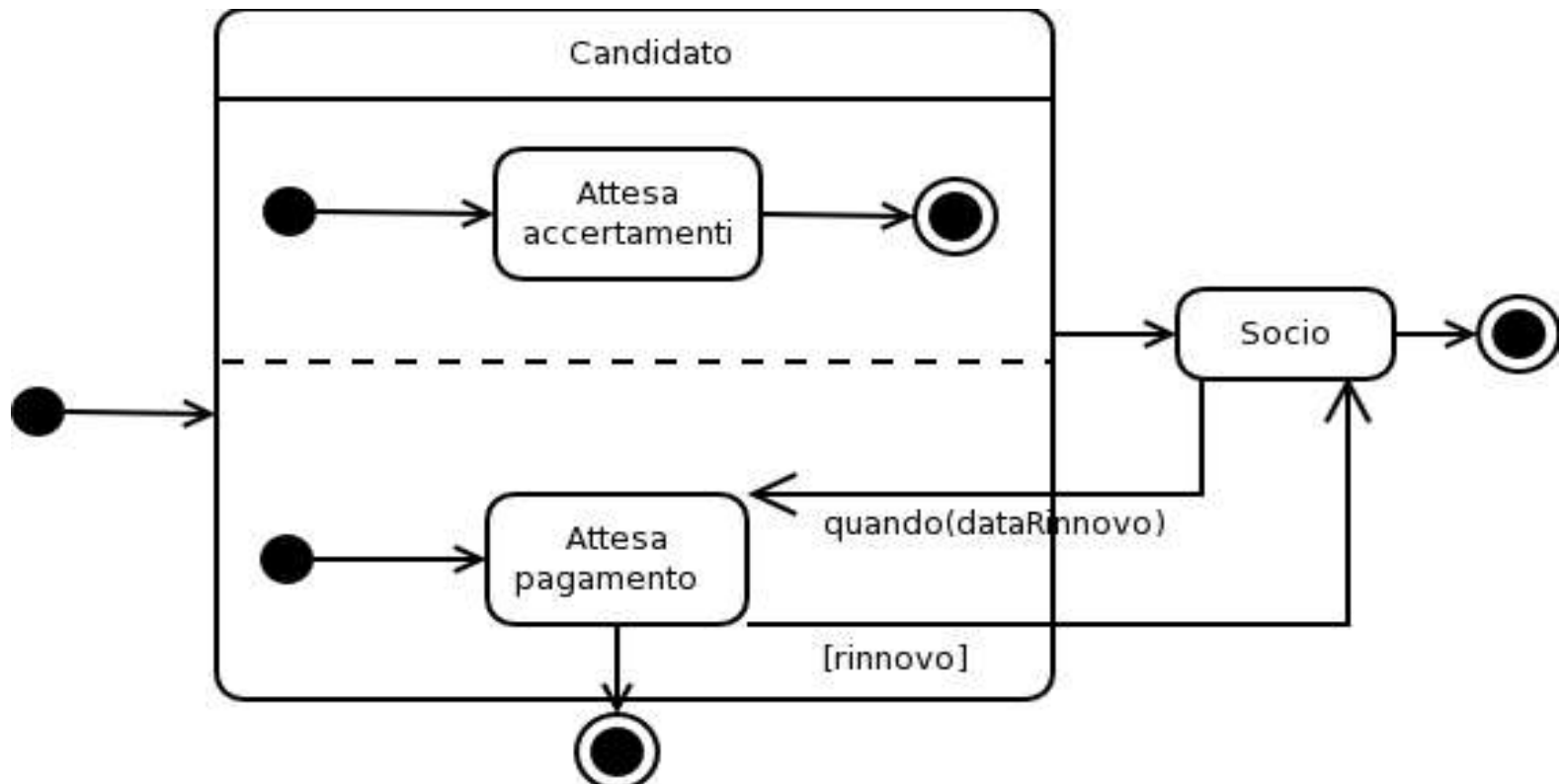
- Nell'esempio precedente è possibile eliminare del tutto lo stato *Candidato* e promuovere i due sottostanti



- Si presenta lo stato *Socio* solo quando *Attesa pagamento* e *Attesa accertamenti* vengono portati a termine
- Quando lo stato di una classe si spezza in stati paralleli, questi si devono ricombinare nel medesimo diagramma

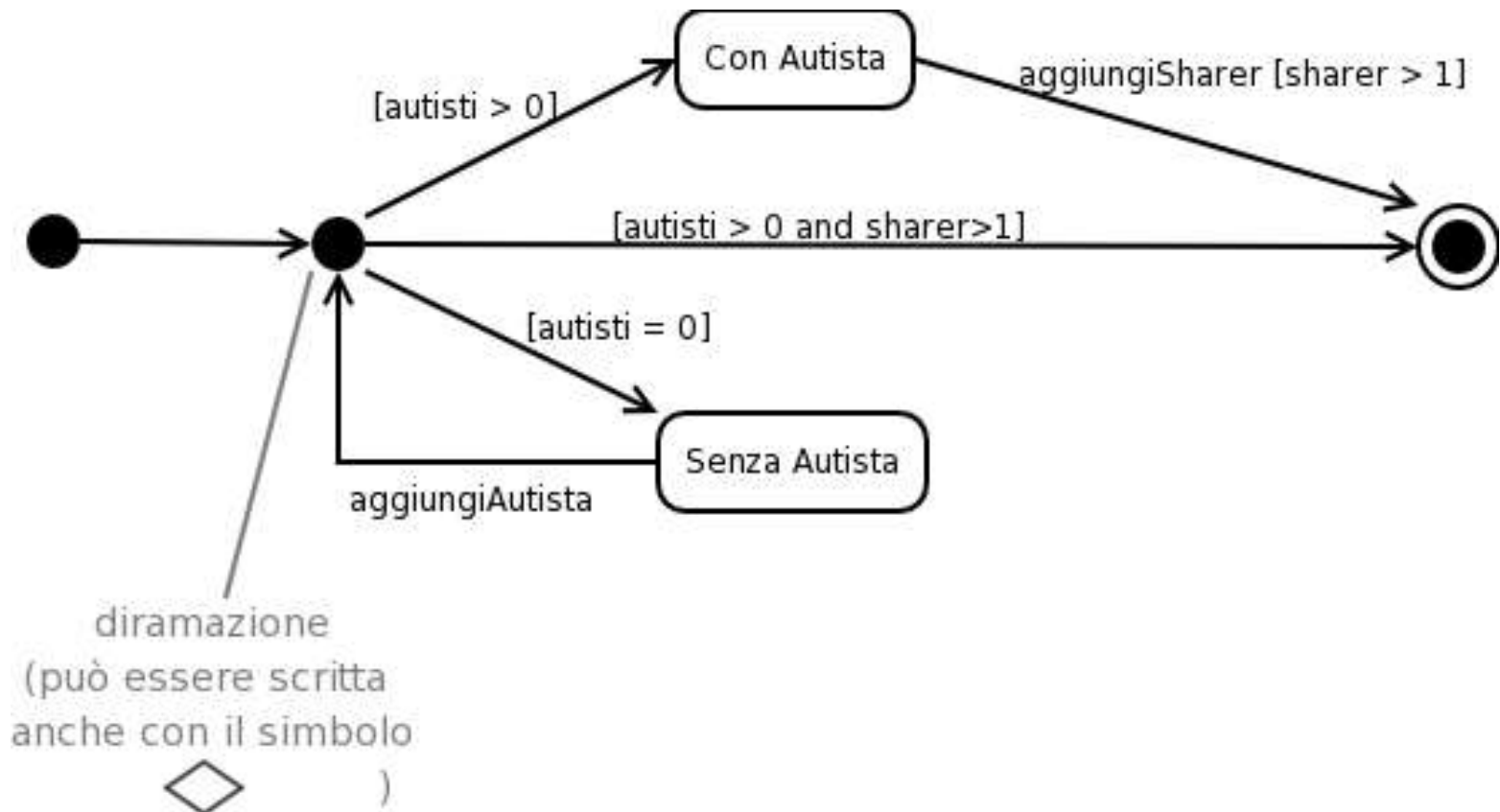
Transizioni verso e da stati composti

- Per indicare che, in fase di rinnovo, non si ha la necessità di ulteriori accertamenti legali, si modella una transizione con condizione verso lo stato interno relativo al pagamento



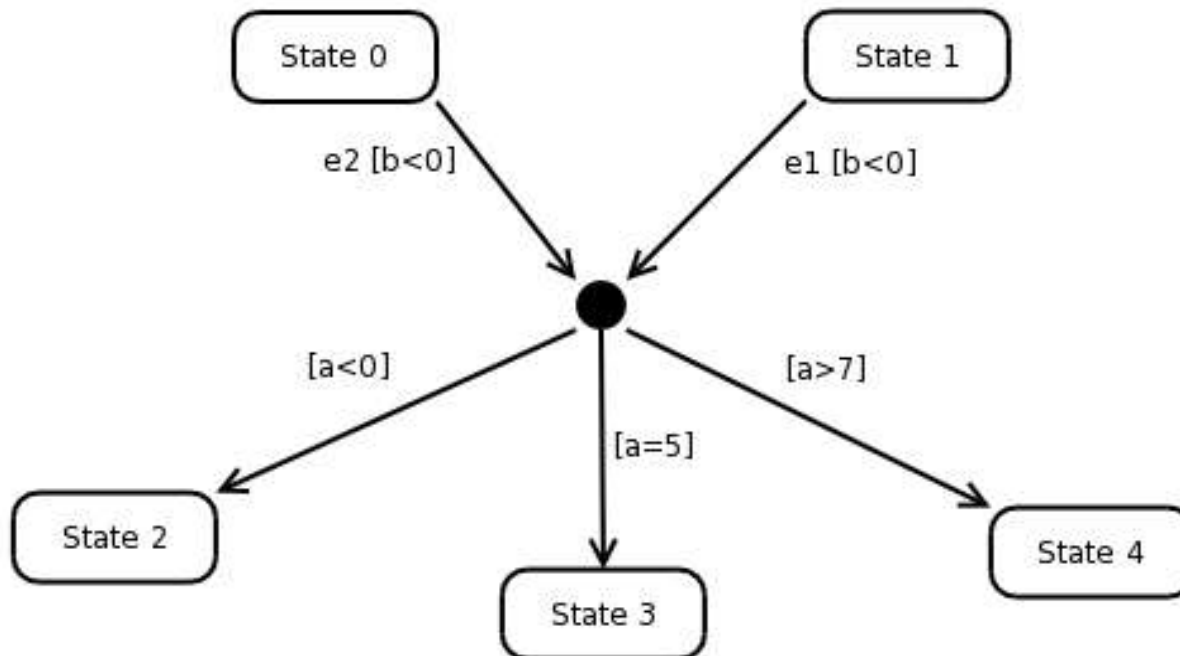
Punti di diramazione

- Un *punto di diramazione* permette di suddividere una transazione sulla base di una condizione.
- Ad esempio:



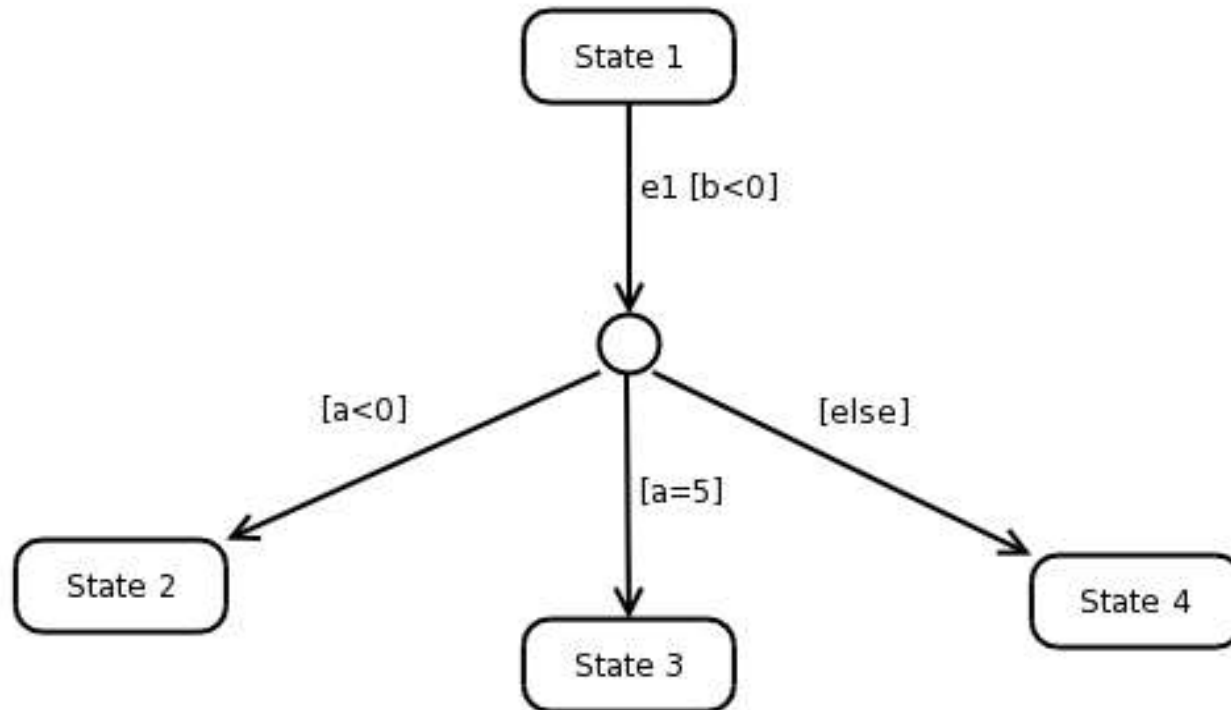
Transizione condizionale statica

- Nella **diramazione statica** tutte le *guardie* sono valutate prima che qualsiasi transizione sia percorsa
- Ad esempio:



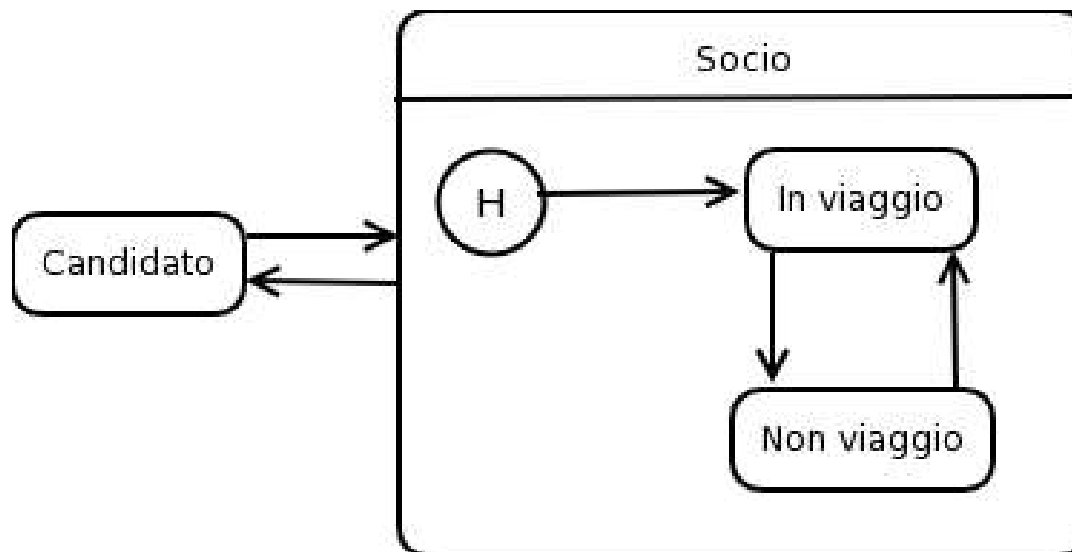
Transizione condizionale dinamica

- Nella **diramazione dinamica** le *guardie* sono valutate nell'istante in cui è raggiunto il punto della decisione
- Ad esempio:




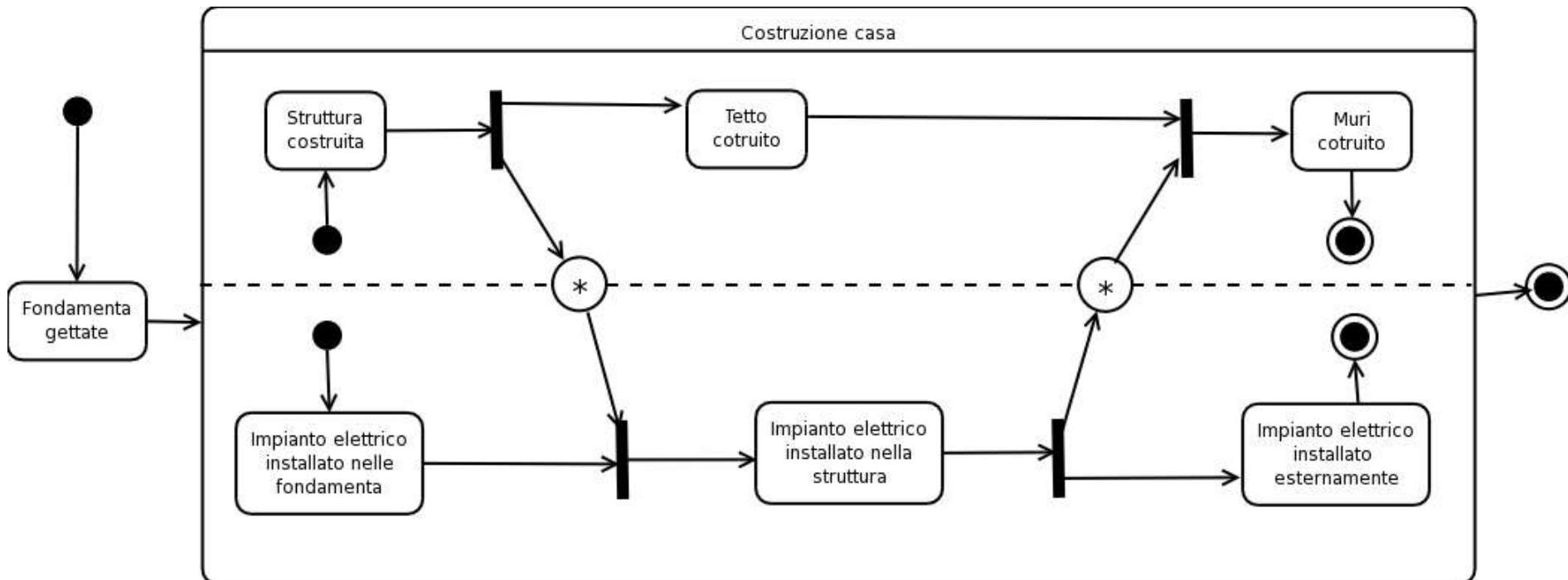
Stati con memoria

- Normalmente, quando si entra in uno stato composto la transizione parte dallo stato iniziale
- A volte è utile poter rientrare in uno stato composto nello stesso punto dal quale si è usciti:
- Questo viene fatto aggiungendo uno **stato con memoria**, indicato con un cerchio con la lettera *H* al suo interno (da *History*).



Stati di synch

- A volte è necessario sincronizzare i flussi paralleli di uno stato composto
- Per questo si usa la notazione degli **stati di synch**: 
- Ad esempio:



Diagrammi di attività

Diagrammi di attività

- I diagrammi di attività
 - sono un tipo speciale di diagramma di stato in cui ogni stato ha un'azione di ingresso che specifica una procedura o funzione da eseguire quando si entra nello stato
 - contengono stati di azione e stati di sottoattività
- Gli stati di azione rappresentano attività che non possono essere scomposte in sottoattività
- È possibile associare un diagramma di attività a qualunque elemento di modellazione, al solo fine di modellarne il comportamento

Diagrammi di attività

- Ci sono quattro modi di attivare un'azione:
 - **Entry**: le azioni sono attivate nel momento in cui ha inizio l'attività
 - **Do**: le azioni si svolgono durante il ciclo di vita dell'attività
 - **Evento**: le azioni avvengono in risposta a un particolare evento
 - **Exit**: le azioni si svolgono appena prima che l'attività termini



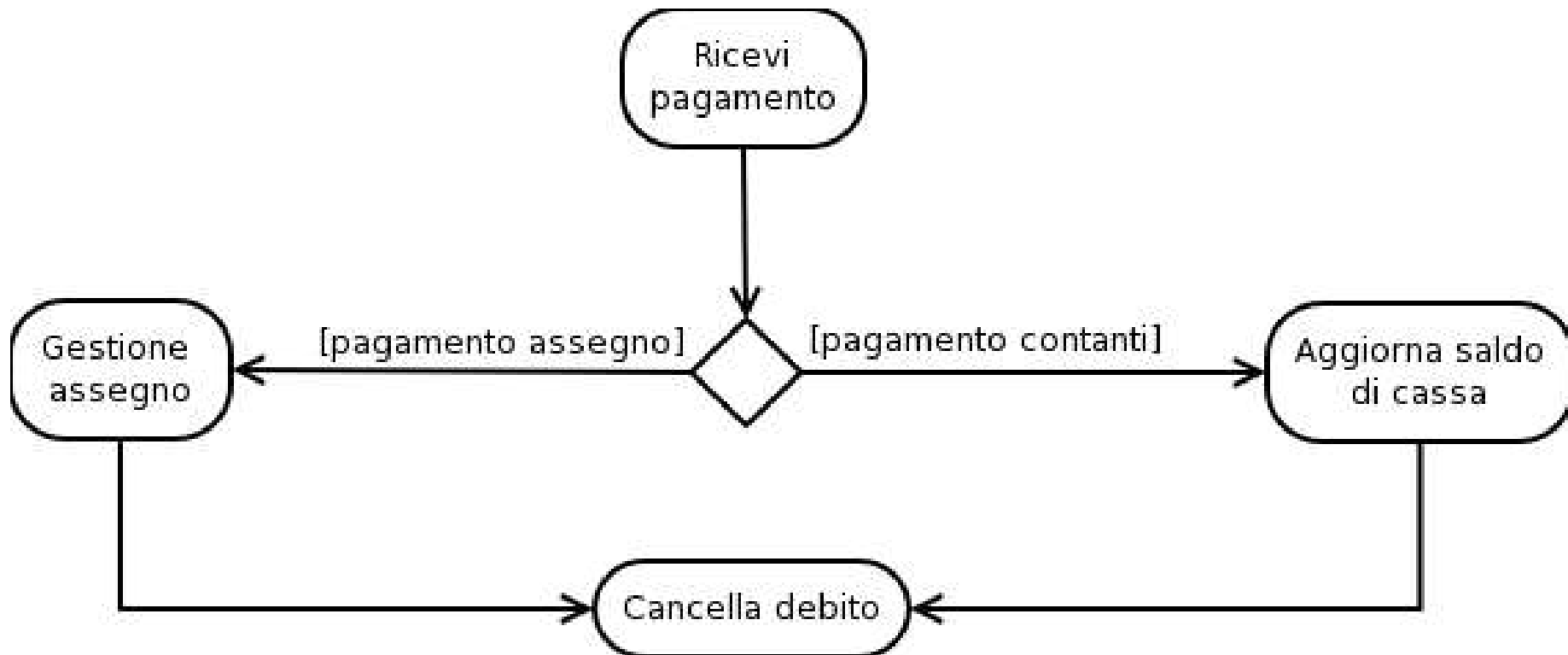
Transizioni

- Quando uno ***stato di azione*** o di ***sottoattività*** ha terminato il proprio lavoro avviene una ***transizione automatica*** in uscita dallo *stato di azione* e un passaggio allo *stato di azione* successivo.



Decisioni

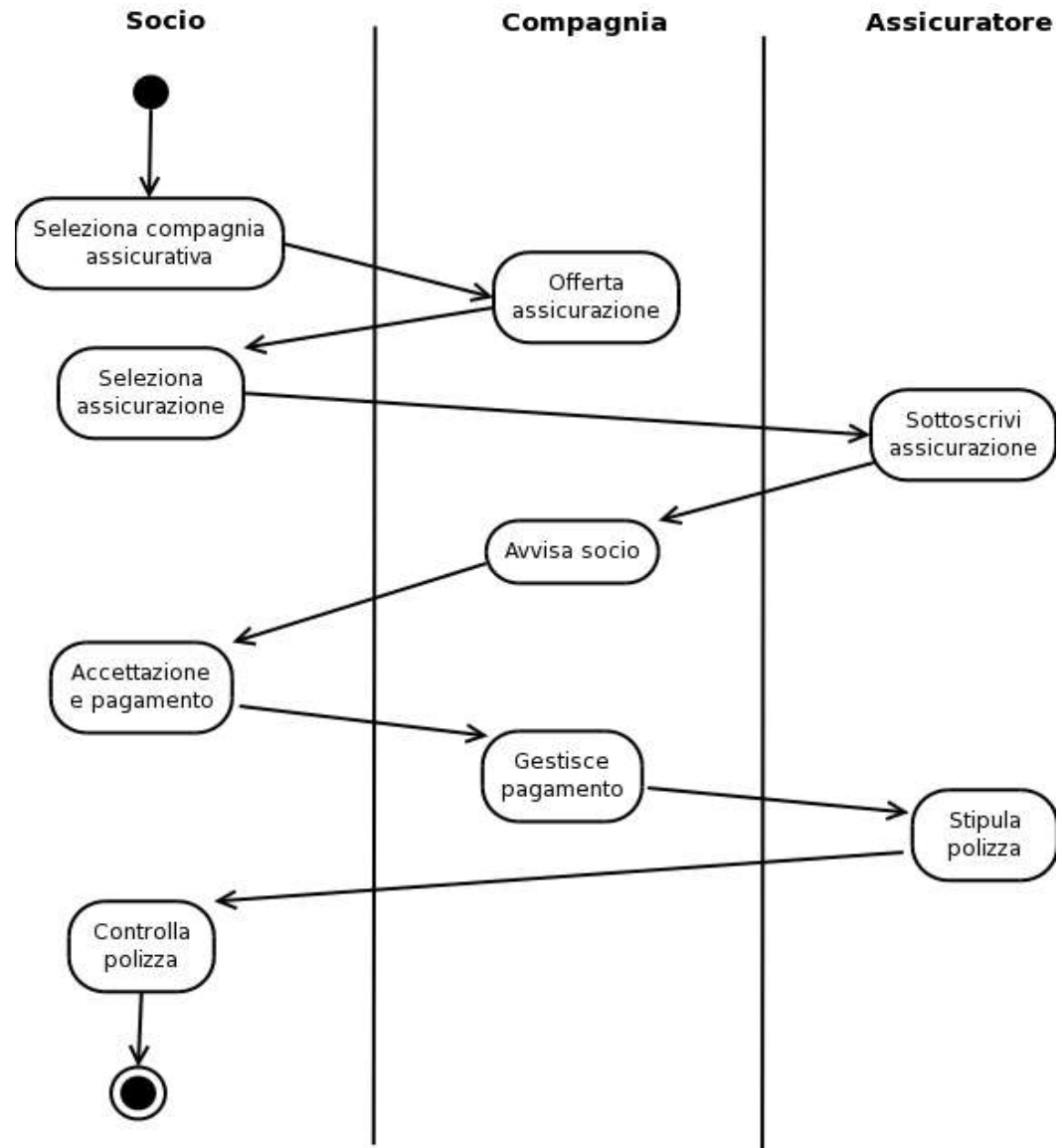
- Un **punto di decisione** (o di **branch**) si presenta quando la transizione in uscita da un'attività porta in direzioni diverse a seconda di una condizione



Corsie di marcia

Le ***corsie di marcia*** o ***swimlane***

costituiscono una notazione atta ad indicare le aree nelle quali vengono svolte le varie attività



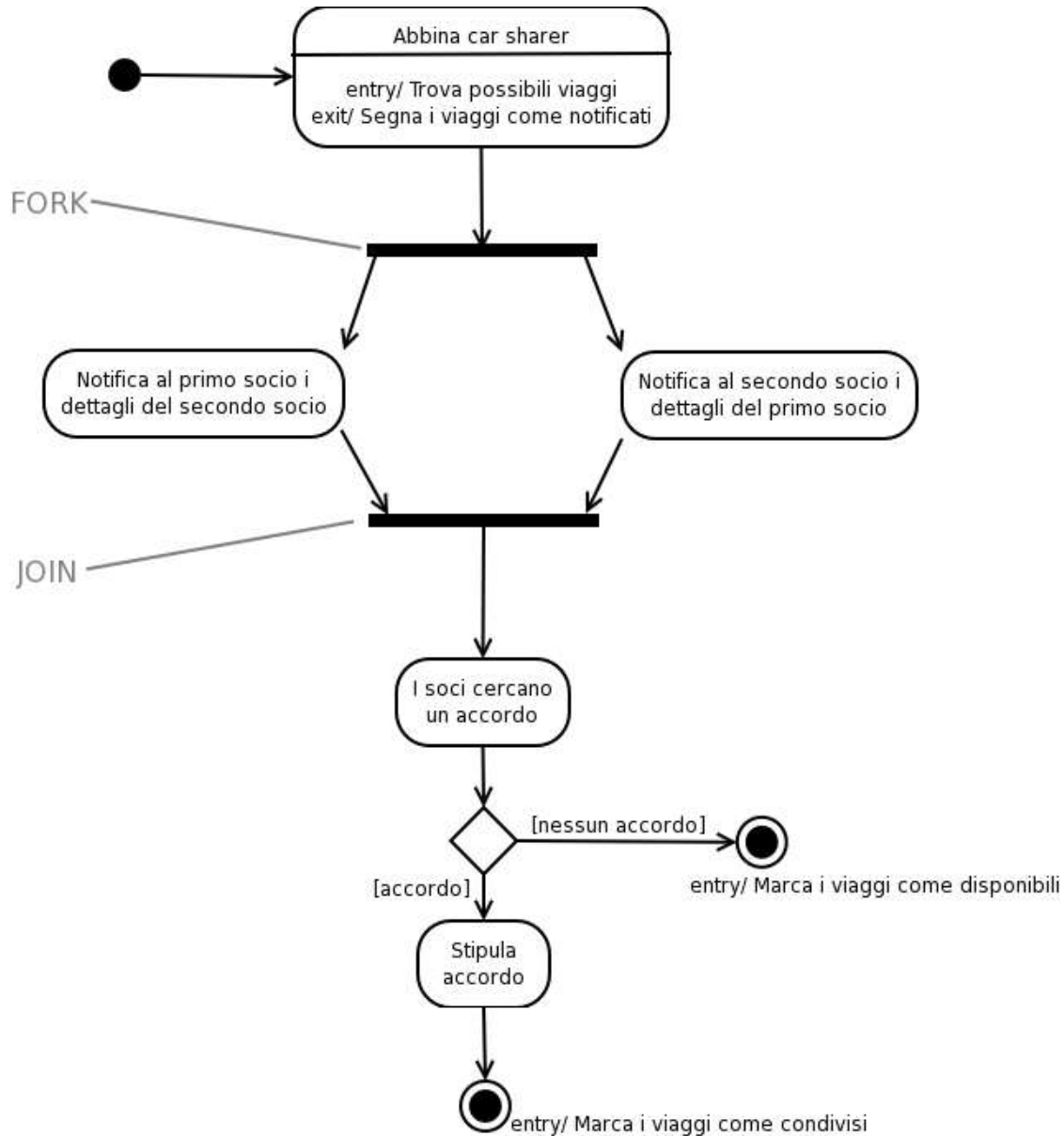
Fork e Join

- A volte può essere utile che un certo numero di attività si svolgano in parallelo.
- Una transizione può essere spezzata in più cammini e più cammini possono essere raggruppati in un'unica transizione utilizzando una ***linea di sincronizzazione***
- La separazione dei cammini prende il nome di **fork** (divisione)
- L'unione di più cammini prende il nome di **join** (unione)

Fork e Join (esempio 1/2)

- Nell'esempio CarMatch (visto per i casi d'uso), il caso d'uso Abbinare car sharer può essere modellato come processo di azioni:
 - una volta che i viaggi sono stati abbinati, vengono avvertiti i potenziali car sharer nello stesso momento, ossia in parallelo
 - successivamente i soci dovranno mettersi d'accordo fra loro.

Fork e Join (esempio 2/2)



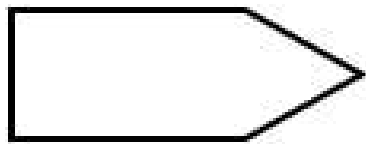
Oggetti nei diagrammi delle attività

- È utile specificare il punto in cui il flusso coinvolge un oggetto, collocando l'oggetto sul diagramma e collegandolo all'attività che lo riguarda con una relazione di *dipendenza* (*in input e in output*).
- Tali *dipendenze* sono conosciute come **flussi per gli oggetti**
 - Infatti indica come viene utilizzato un oggetto all'interno di un flusso di controllo
- Il flusso modifica lo stato di un oggetto
 - Ogni oggetto ha uno stato ad esso associato

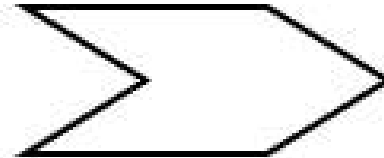


Segnali

- Il **segnale** consente di rappresentare un pacchetto di informazioni che viene comunicato in modo **asincrono** tra due oggetti
- Un **evento di segnale** avviene quando un oggetto riceve un segnale
- La ricezione di un evento può provocare una *transizione di stato*
- Fatta eccezione per gli *eventi di segnale*, normalmente non si riportano eventi sui *diagrammi di attività*, ma solo sui *diagrammi di stato*.



Invio di segnale



Ricezione di segnale

Segnali (esempio)

Rilevatore autovelox



Monitoraggio del traffico



Riferimenti

- [UML 1.5] *OMG UML Specification v. 1.5.*
- [AN02] Arlow, Neustadt, *UML e Unified Process*, McGraw-Hill, 2002
- [BSL02] Bennett, Skelton, Lunn, *Introduzione a UML*, McGraw-Hill collana Schaum's, 2002
- I diagrammi sono stati realizzati con *Dia* ed *Eclipse*
<http://www.gnome.org/projects/dia/>
<http://www.eclipseuml.com>