

Progettazione del Software

Diagramma delle Attività

Giuseppe De Giacomo, Massimo Mecella

Dipartimento di Informatica e Sistemistica

SAPIENZA Università di Roma

Processi

Accanto a rappresentare quali sono le informazioni di interesse nel dominio della nostra applicazione, dobbiamo anche rappresentare come la nostra applicazione accede, modifica, usa tali informazioni. Cioè dobbiamo analizzare i **processi** che sono di interesse per la nostra applicazione. La descrizione delle processi si ottiene attraverso l'interazione tra analista ed utente in fase di analisi.

In UML per descrivere i processi si usa il **diagramma delle attività**.

Il **diagramma delle attività** descrive le attività che il sistema deve supportare, in termini di attività atomiche realizzate nel sistema e del flusso lavoro (workflow) in cui sono coinvolte.

Diagramma delle Attività

Le **attività** rappresentano le operazioni e le processi (organizzativi, tecnici, ecc.) che il sistema deve supportare

Le attività agiscono, accedono e aggiornano, l'istanziamento del diagramma delle classi, modificandone lo stato corrente del sistema, cioè le informazioni memorizzate nel sistema stesso.

Inoltre le attività accedono all'esterno del sistema per fornire e ricevere informazioni dai suoi utilizzatori.

Si noti che anche un'attività è **trasversale** rispetto alle classi, cioè **coinvolge più classi**.

3

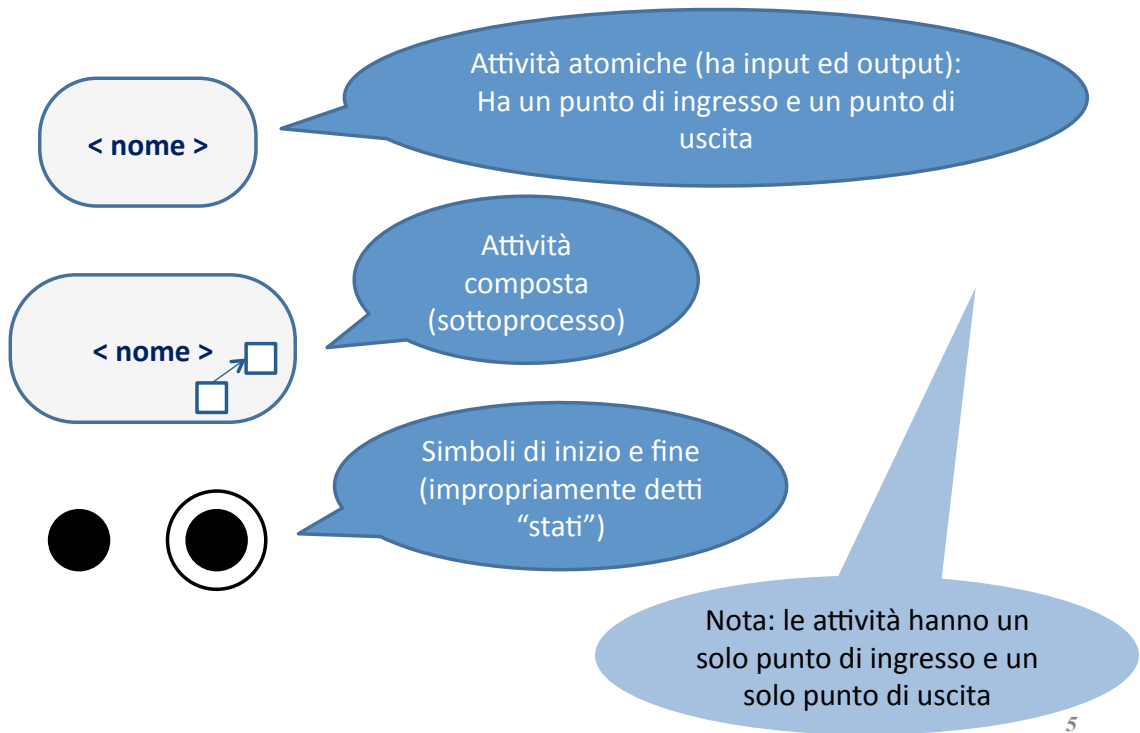
Diagramma delle Attività

Un diagramma delle attività è composto:

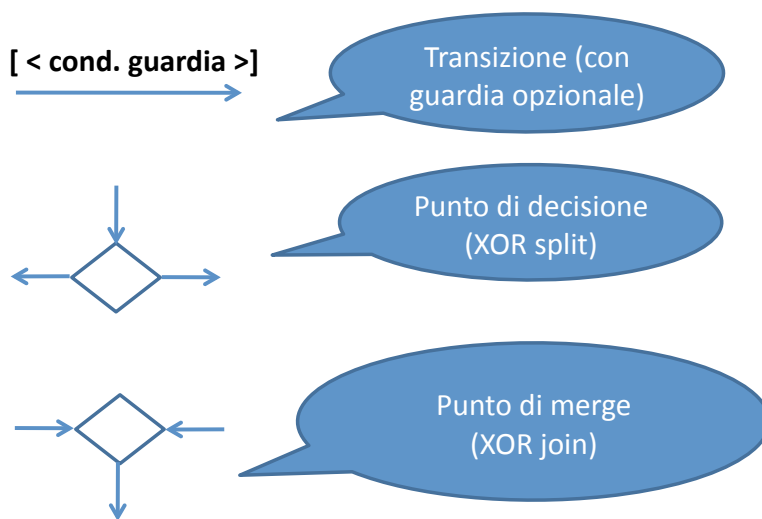
- da diverse **attività atomiche**, che rappresentano
 - Operazioni (che chiameremo **task**) sul dominio dell'applicazione, così come l'abbiamo modellato nel diagramma delle classi
 - **Operazioni di I/O** (input/output) che realizzano l'interfacciamento (attraverso una qualche interfaccia utente testuale, grafica, ecc.) del sistema con l'utente
- dal **flusso di controllo** tra queste attività atomiche.

4

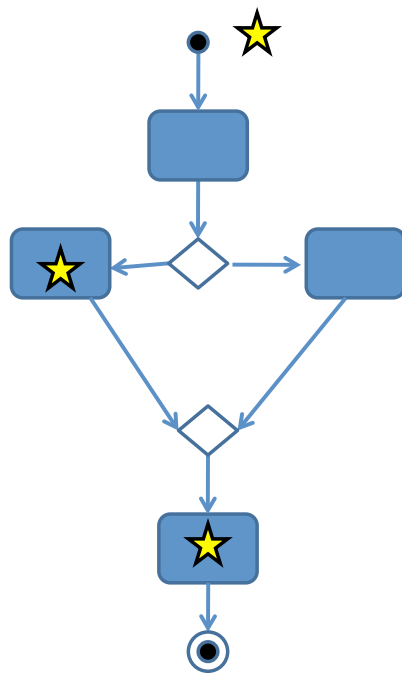
Elementi di Base: Atomici (1)



Elementi di Base: Sequenziali (2)



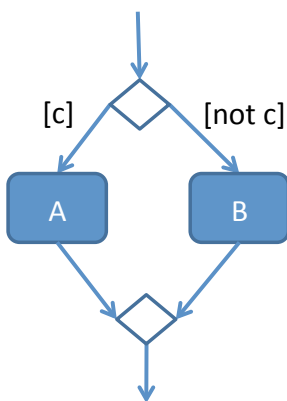
Esempio



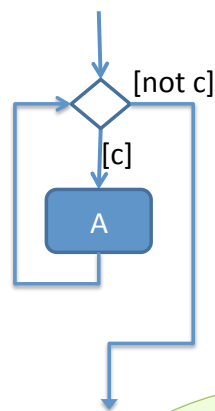
7

Altri esempi

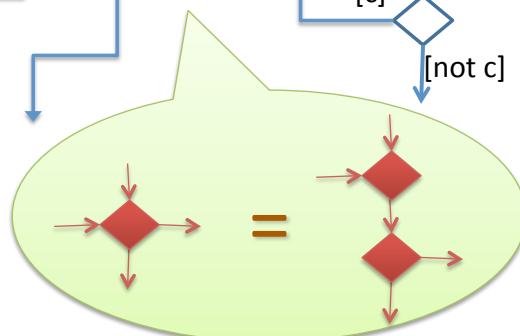
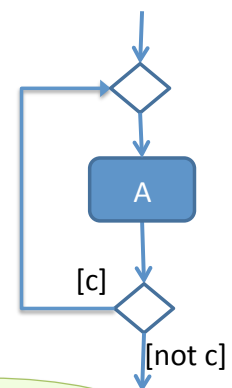
if (c) A else B



while (c) A

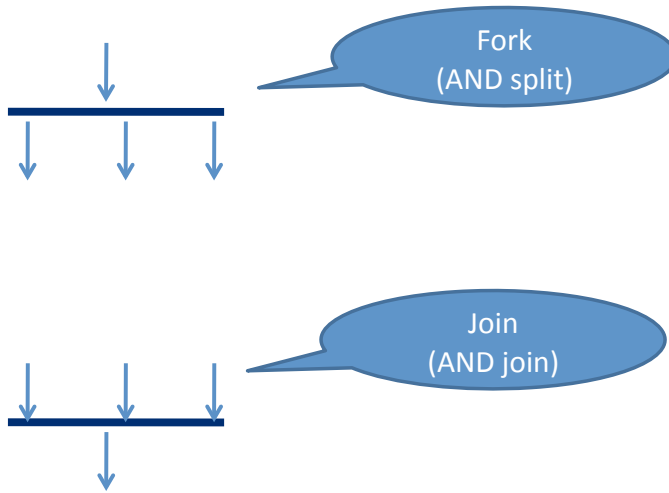


do A while (c)



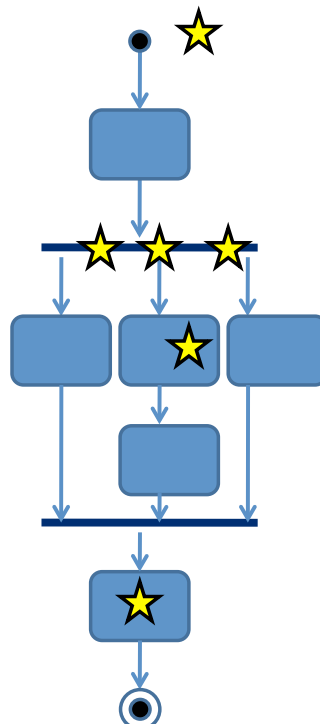
8

Elementi di Base: per Concorrenza (3)



9

Esempio



10

Nota sul significato del Fork/Join

- Flusso (*thread*) di esecuzione
 - “Esecutore” che è in carico di gestire l’avanzamento del processo
 - Nel fork, il numero di tali flussi/thread/executori viene aumentato, in quanto prima era unico e dopo ce ne è uno per ogni transizione in uscita
 - Nel join il numero viene diminuito, in quanto prima ce ne erano tanti (uno per ogni transizione in entrata) e dopo ce ne è di nuovo uno solo
 - Questo può partire solo dopo che sono terminati tutti quelli che devono fare join
 - Sincronizzazione dei executori

11

Differenza tra Fork/Join e Punto di Decisione/Merge

Attenzione, alla differenza tra il fork (and split) e il punto di decisione (xor split). In quest’ultimo caso, solo una delle transizioni in uscita viene eseguita, mentre nel fork tutte le transizioni di uscita vengono eseguite contemporaneamente.

Una simile differenza sussiste anche tra il join (and join) e il merge (xor join). Nel join tutte le transizioni entranti devono essere completate per proseguire, mentre nel merge, solo una transizione entrante è significativa.

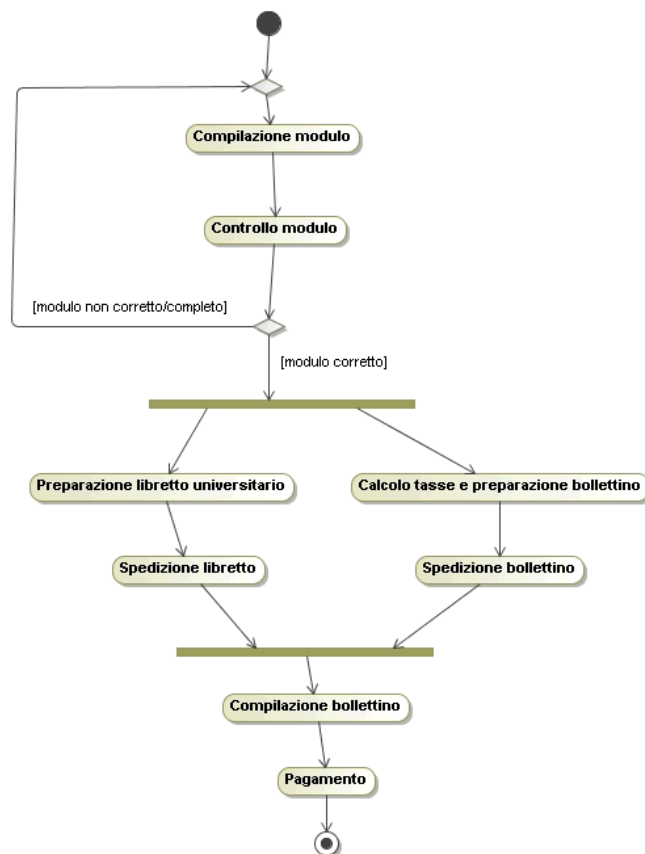
12

Esempio

- Per iscriversi all'Università viene compilato un modulo, che viene ispezionato al fine di verificarne la correttezza, ed eventualmente fatto ricompilare fino a quando non è corretto e completo. Successivamente, viene preparato il libretto universitario e spedito a casa dello studente, e contemporaneamente vengono calcolate le tasse di iscrizione e spedito l'apposito bollettino. Alla ricezione di entrambi, si può procedere alla compilazione del bollettino (infatti bisogna inserire il numero del libretto) ed al pagamento del bollettino

13

Esempio



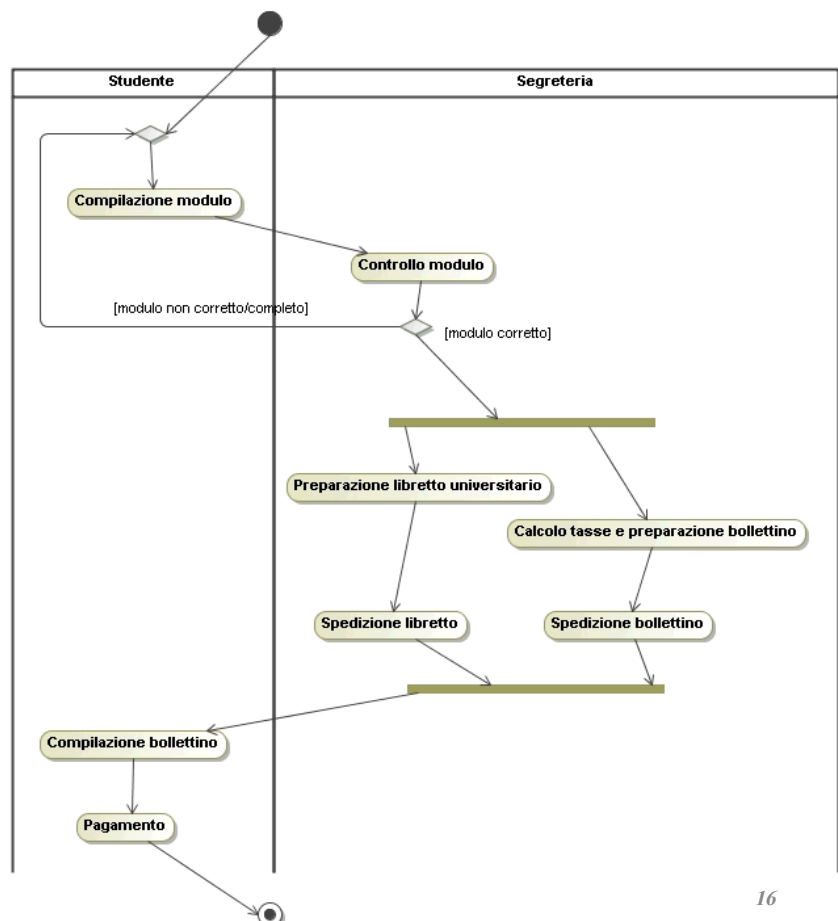
14

Swimlane

- Problema: il diagramma non evidenzia chi deve fare cosa (ovvero gli attori che eseguono le attività).
- Se si vuole mettere in evidenza chi fa cosa si introducono le swimlane.
- Una swimlane (“corsia di nuoto”) permette di evidenziare quali azioni sono eseguite da un dato attore
 - Graficamente è una linea verticale che identifica la “corsia”

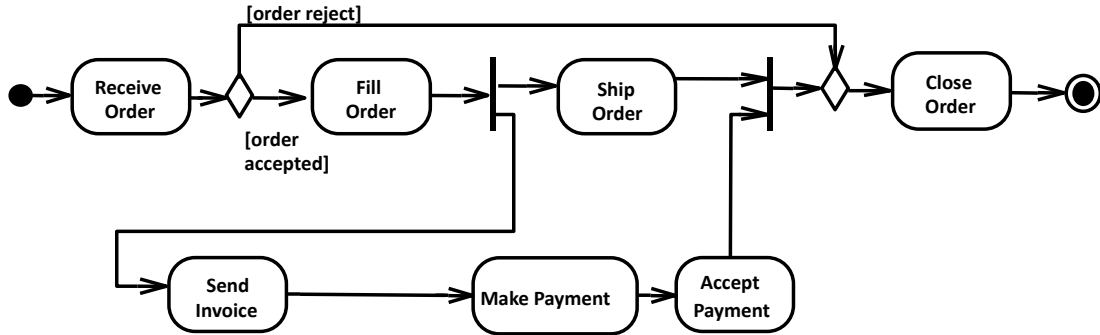
15

Esempio con Swimlane



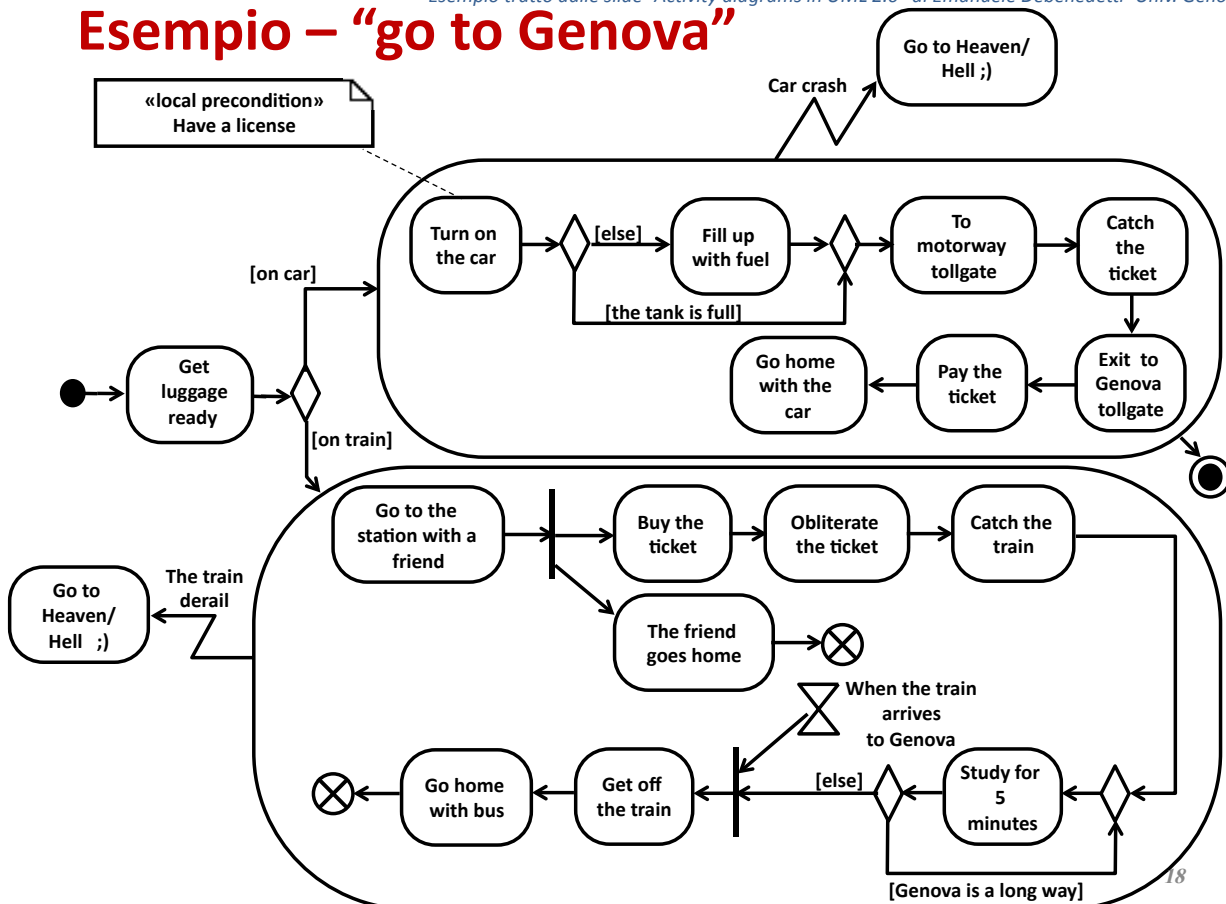
16

Esempio – "orders"



17

Esempio – "go to Genova"



18

FlowFinal

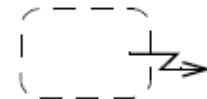
- A volte serve segnalare che un ramo concorrente della computazione va a chiudersi.
- Per fare ciò si usa il simbolo di terminazione di un ramo concorrente: FlowFinal



19

Costrutti per gli eventi

- Regione interrompibile
 - Raggruppamento di attività e transizioni che supporta la terminazione del flusso di esecuzione
 - Se la terminazione avviene, tutto il flusso ed il comportamento dovuto alle attività interne alla regione è terminato



- Evento
 - genera un evento per sincronizzarsi con una condizione esterna (eg temporale)

at() indica la condizione che triggera l'evento (e la relativa transizione):



20

Costrutti per gli eventi



Attività “Invia Segnale”



Attività “Ricevi Segnale”



Evento esterno (eg. Temporale)

at() indica la condizione che triggera l'evento (e la relativa transizione)

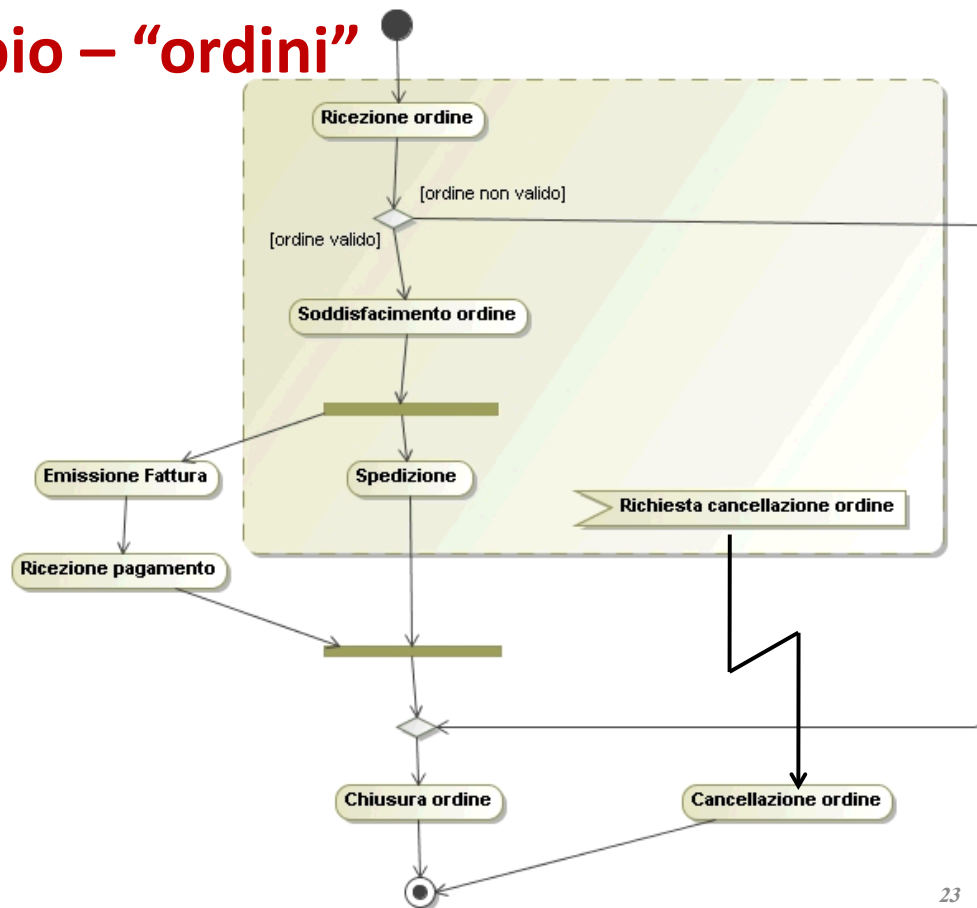
21

Esempio – “ordini”

- Rappresentare il processo di gestione degli ordini. Dopo la ricezione di un ordine, se esso viene accettato si procede al soddisfacimento dello stesso, e quindi si attivano in parallelo la spedizione dell'ordine e la preparazione della fattura. A pagamento ricevuto, l'ordine può essere chiuso. In un qualsiasi momento prima dell'emissione della fattura, l'ordine può essere cancellato, e quindi chiuso.

22

Esempio – “ordini”



23

Relazione Diagramma delle Attività con gli Altri (Classi e Stati/Transizioni)

- Il diagramma delle attività rappresenta la specifica del processo che il sistema supporta
- Ogni attività potrà
 - Manipolare gli oggetti (cfr. Diagramma delle Classi e Realizzazione): aggiungere/togliere/modificare oggetti e link
 - Scatenare degli eventi che daranno luogo ad una interazione tra gli oggetti (cfr. Diagrammi degli Stati e Transizioni e Realizzazione): lanciare un evento/messaggio

27