

## Elementi di UML (4)

Università degli Studi di Bologna  
Facoltà di Scienze MM. FF. NN.  
Corso di Laurea in Informatica  
Anno Accademico 2001-2002

Corso di:  
Laboratorio di Ingegneria del Software

Rocco Moretti    moretti@cs.unibo.it



UML

1

## Attività vs. Stato

### n Attività:

Un insieme di azioni che deve essere necessariamente ed interamente completato prima di potersi considerare terminato.

### n Stato:

Un punto ben preciso nel quale è necessario che determinati eventi abbiano luogo prima che una certa attività possa proseguire.

Sebbene gli stati possano svolgere azioni esattamente come le attività, questi sottointendono un concetto di attesa.

UML

2

## Activity diagram

- n I diagrammi delle attività (activity diagram) sono costituiti da *attività*, *stati*, e *transizioni* tra attività e stati
- n Descrivono flussi tra aree di lavoro
- n ... mentre gli statechart diagram descrivono le modifiche dello stato degli elementi modellati

UML

3

## Activity Diagrams basati su Statechart diagram

- n Attualmente gli activity diagram sono modellati come un particolare tipo di statechart diagram
- n Gli activity diagram diventeranno indipendenti dagli statechart diagram in UML 2.0

UML

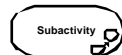
4

## Activity diagram: notazione (1/3)

### n Attività:



### n Sottoattività:



- n Le transizioni uscenti avvengono quando le attività indicate sono state completate, invece di avvenire in seguito ad un evento esterno
- n Le attività possono essere concorrenti

UML

5

## Activity diagram: notazione (2/3)

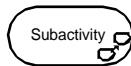


- n Un'attività è da utilizzarsi quando questa non sottointende una sottoattività (ad es., l'invocazione di un'operazione su un oggetto)
- n Tuttavia, un'attività può invocare un'operazione che ha un'altra attività come metodo (vedi esempio)

UML

6

### Activity diagram: notazione (3/3)

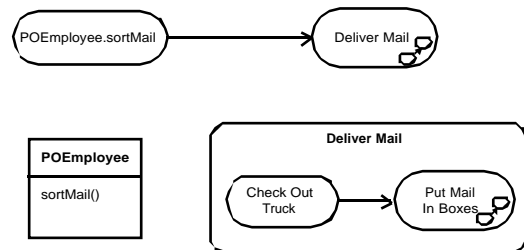


- Una sottoattività sottointende un activity diagram interno non dipendente dall'utilizzo di un'operazione
- Utilizzato per ottenere decomposizione funzionale

UML

7

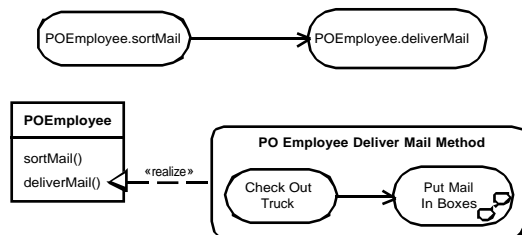
### Attività e Sottoattività: esempio



UML

8

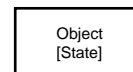
### Attività come metodo: esempio



UML

9

### Oggetti negli activity diagram

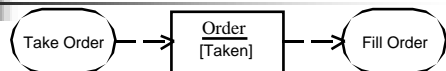


- Può essere utile specificare un punto in cui delle attività coinvolgono un oggetto collegandolo all'attività che lo riguarda con una relazione di dipendenza
- Ogni oggetto coinvolto in un'attività si troverà per questo in un determinato stato

UML

10

### Oggetti negli activity diagram: esempio



- L'attività *Take Order* deve avere un parametro di output rappresentato dall'oggetto Order nello stato *Taken*
- L'attività *Fill Order* deve avere un parametro di input rappresentato dall'oggetto Order nello stato *Taken*
- Il collegamento tra attività ed oggetti, oltre ad esprimere una relazione tra l'oggetto e le attività, indica un effettivo ordine nelle transizioni

UML

11

### Strumenti di coordinazione

Eredità degli statechart diagram:

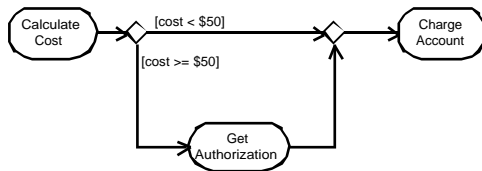
- Stato iniziale:
- Stato finale:
- Fork e Join:
- Decisione e Riunione:
- Stato di sincronizzazione:

UML

12

## Decisioni e Riunioni: esempio

- Per modellare decisioni di controllo di flusso:



UML

13

## Stato di sincronizzazione



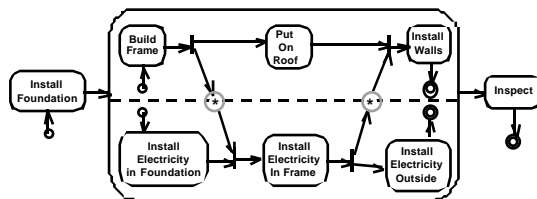
- Finalizzato alla sincronizzazione di regioni concorrenti
- Utilizzato in congiunzione con Fork e Join per assicurare che una determinata attività/(stato) di una regione termini/(sia abbandonato) prima che in una seconda regione una nuova attività/(stato) venga iniziata/(sia raggiunto)
- Utilizzato principalmente negli activity diagram

UML

14

## Stato di sincronizzazione: esempio (1/2)

- Fornisce una possibilità di sincronizzazione tra processi paralleli

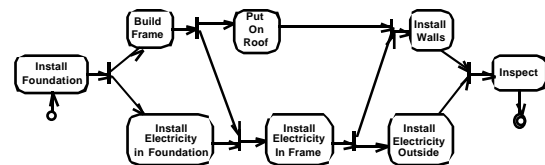


UML

15

## Stato di sincronizzazione: esempio (2/2)

- Fork e Join non necessitano di stati composti
- Lo stato di sincronizzazione può essere omesso nei casi banali (una sola transizione entrante ed una sola uscente)

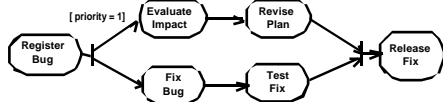


UML

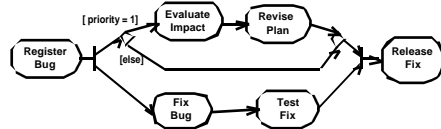
16

## Guardie

- Le transizioni di Fork possono avere guardie:



- Equivalente a:

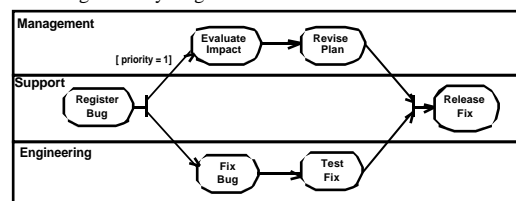


UML

17

## Swimlane

- Le corsie di marcia (swimlane) costituiscono una notazione utile per indicare dove un'attività ha luogo
- Non forniscono una semantica specifica per il dominio degli activity diagram



UML

18

## Icone di controllo

- Icona d'invio di un segnale:

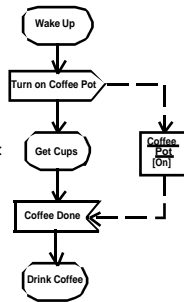


- Si converte in una transizione con un'azione d'invio

- Icona di ricezione di un segnale:



- Si converte in uno stato d'attesa (uno stato senza azioni e con eventi legati alla percezione di segnali)



UML

19

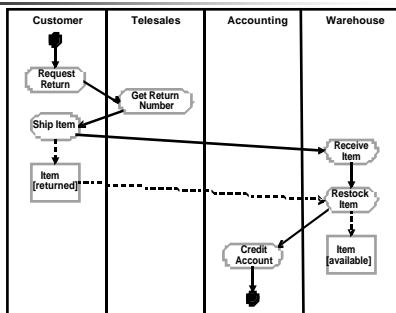
## Modellazione degli Activity Diagram (1/6)

- Il controllo del flusso delle attività e quello degli oggetti non è separato: entrambi sono modellati attraverso transizioni
- È possibile inserire costrutti per il controllo di flusso e la sincronizzazione che coinvolgono: stati, attività, ed oggetti

UML

20

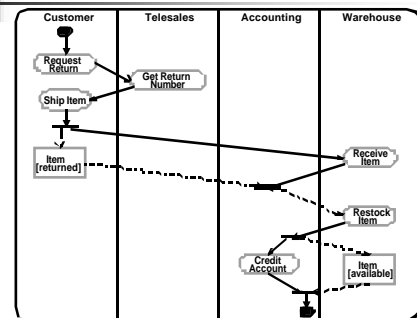
## Modellazione degli Activity Diagram (2/6)



UML

21

## Modellazione degli Activity Diagram (3/6)



UML

22

## Modellazione degli Activity Diagram (4/6)

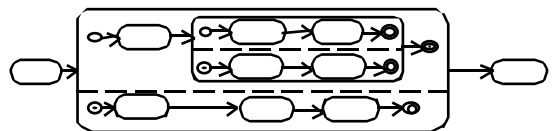
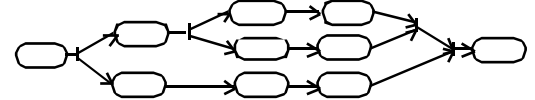
- Gli Activity diagram ereditano dagli statechart diagram il requisito di corretta strutturazione dell'annidamento delle attività e degli stati
- Occorre dunque che:
  - Ad ogni *Fork/Decisione* corrisponda un *Join/Merge*
  - L'annidamento degli stati sia ben strutturato
- Ciò consentirà a tali diagrammi di essere interpretati con una semantica analoga a quella di un automa a stati finiti

UML

23

## Modellazione degli Activity Diagram (5/6)

Strutturazione corretta:

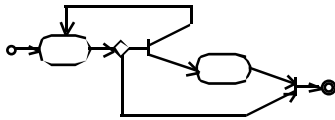


UML

24

## Modellazione degli Activity Diagram (6/6)

Strutturazione scorretta:



È opportuno applicare i principi della programmazione strutturata e rivedere la modellazione proposta.

UML

25

## Quando utilizzare gli Activity Diagram?

- n Utilizza gli activity diagram quando il comportamento che si vuole modellare:
  - n non dipende sostanzialmente da eventi esterni
  - n è caratterizzato da attività che si svolgono fino alla loro conclusione, con scarse interruzioni dovute ad eventi
  - n richiede un flusso di oggetti/dati nel corso delle varie attività
  - n è in una fase di modellazione nella quale si è più interessati a *quali attività* debbano svolgersi, piuttosto che a *quali oggetti* siano responsabili di tali attività

UML

26

## Riferimenti

- n [UML 1.3] *OMG UML Specification v. 1.3*, OMG doc# ad/06-08-99
- n [UML 1.4] *OMG UML Specification v. 1.4*, UML Revision Task Force recommended final draft, OMG doc# ad/01-02-13.

UML

27

## Per ulteriori informazioni

- n Web:
  - n UML 1.4 RTF: [www.celigent.com/omg/umlrtf](http://www.celigent.com/omg/umlrtf)
  - n OMG UML Tutorials: [www.celigent.com/omg/umlrtf/tutorials.htm](http://www.celigent.com/omg/umlrtf/tutorials.htm)
  - n UML 2.0 Working Group: [www.celigent.com/omg/adptf/wgs/uml2wg.htm](http://www.celigent.com/omg/adptf/wgs/uml2wg.htm)
  - n OMG UML Resources: [www.omg.org/uml/](http://www.omg.org/uml/)
- n Email
  - n [uml-rtf@omg.org](mailto:uml-rtf@omg.org)

UML

28