

Elementi di UML (3)

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

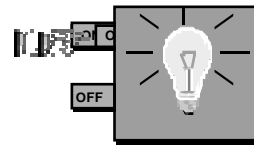


UML

1

Automa

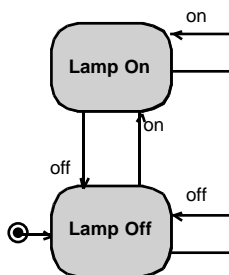
- Una macchina il cui output non è solo una conseguenza diretta dell'input attuale, ma basato sulla storia dei suoi input
- Caratterizzata da uno stato interno che rappresenta questa passata esperienza



UML

2

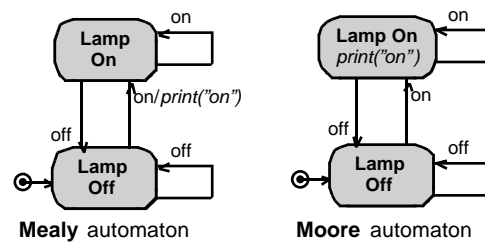
Rappresentazione grafica di un automa



UML

3

Automati di Mealy e di Moore (1/2)



Mealy automaton

Moore automaton

UML

4

Automati di Mealy e di Moore (2/2)

Automa di Mealy:

L'output è associato alla transizione di stato

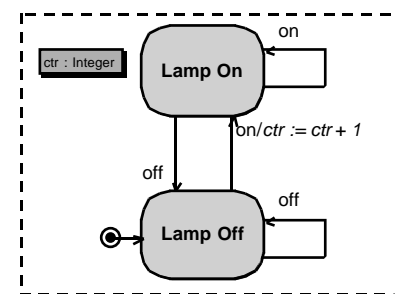
Automa di Moore:

L'output è associato allo stato di destinazione della transizione

UML

5

Estensione dello stato: variabili



UML

6

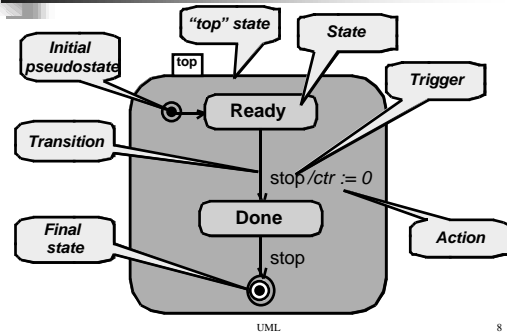
Automa di Mealy esteso

- Un insieme di segnali di input (alfabeto di input)
- Un insieme di segnali di output (alfabeto di output)
- Un insieme di stati
- Un insieme di transizioni
 - segnali di percezione
 - azioni
- Un insieme di variabili per l'estensione dello stato
- Uno stato iniziale
- Un insieme (eventualmente vuoto) di stati finali

UML

7





Statechart Diagram: mappatura sull'automa di Mealy esteso



UML

8

Entità

Construct	Description	Syntax
State	A condition during the life of an object or an interaction during which it satisfies some condition, perform some action, or waits for some event.	 Initial  Final 
State transition	A relationship between two states indicating that an instance in the first state will enter the second state and perform specific action when a specified event occurs provided that certain specified condition are satisfied.	

UML

9

Comportamento basato su eventi

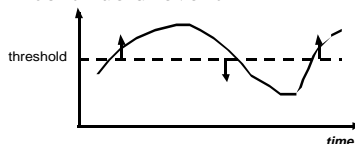
- Evento = Un qualsiasi tipo di occorrenza osservabile
 - interazioni:
 - invocazione di operazioni sincrone su oggetti (call event)
 - ricezione asincrona di un segnale (signal event)
 - occorrenza in determinati istanti di tempo (time event)
 - termine di un determinato intervallo temporale
 - occorrenza di una determinata data/ora
 - modifica del valore di qualche entità (change event)
- Istanza di un Evento:
 - Occorre in un particolare istante di tempo e non ha durata

UML

10

Impiego degli Statechart Diagram (1/2)

- Utili per descrivere comportamenti regolati da un insieme discreto di eventi
- Inappropriati per la modellazione di comportamenti basati su un insieme continuo di eventi



UML

11

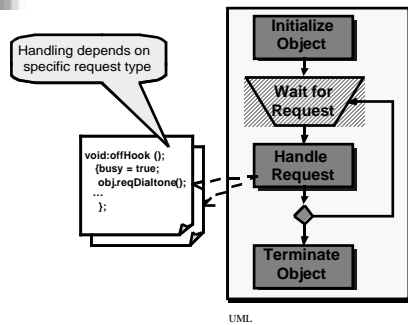
Impiego degli Statechart Diagram (2/2)

- Tutto ciò che manifesta un comportamento basato su eventi
- In pratica:
 - il comportamento di singoli oggetti
 - l'interazione tra oggetti
- Utili per la modellazione di oggetti attivi (vedi seguito)

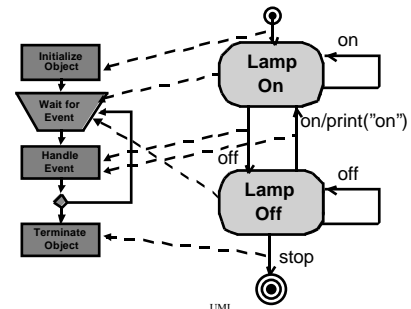
UML

12

Modello di comportamento di un server

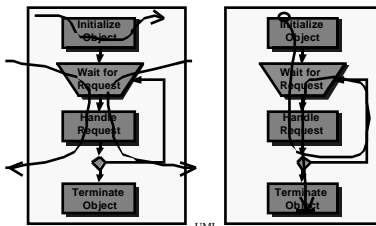


Server mappato su Statechart Diagram

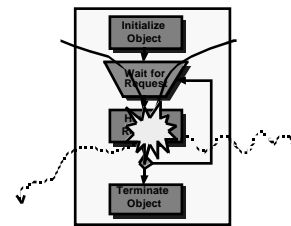


Oggetto passivo ed attivo

- **Oggetto passivo:** il suo comportamento dipende da stimoli esterni
- **Oggetto attivo:** il suo comportamento non dipende da stimoli esterni

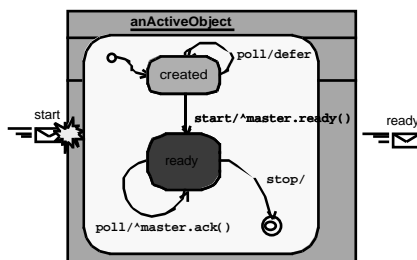


Oggetto passivo: comportamento

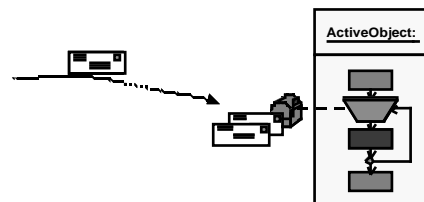


- L'incapsulamento non protegge l'oggetto da conflitti di concorrenza
- Un'esplicita sincronizzazione è ancora necessaria

Oggetto attivo: comportamento (1/3)



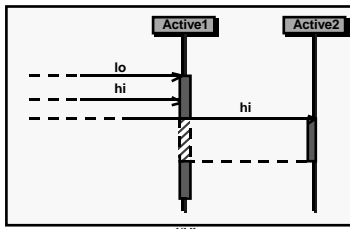
Oggetto attivo: comportamento (2/3)



- Gestione serializzata degli eventi
- Elimina la concorrenza interna

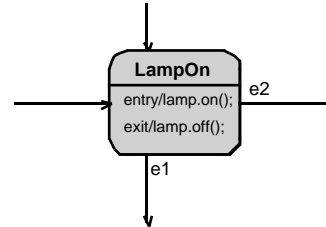
Oggetto attivo: comportamento (3/3)

- Un evento con maggiore priorità riguardante un altro oggetto attivo rende necessaria una preemption dell'oggetto attivo che gestisce l'evento a più bassa priorità



19

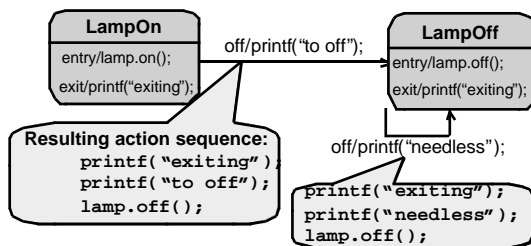
Stato: azioni di entrata e di uscita



UML

20

Ordinamento delle azioni: esempio

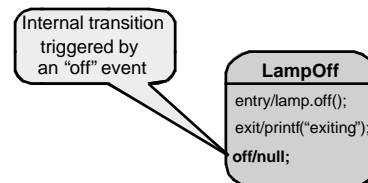


UML

21

Transizione interna

- Transizione interna che esclude azioni di entrata e di uscita

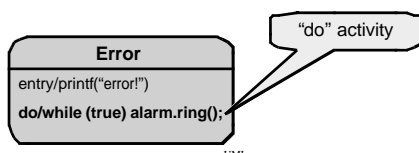


UML

22

Attività interna

- Generazione di un thread concorrente che resta in esecuzione finché:
 - il thread (eventualmente) termina
 - si esce dallo stato attraverso una transizione esterna

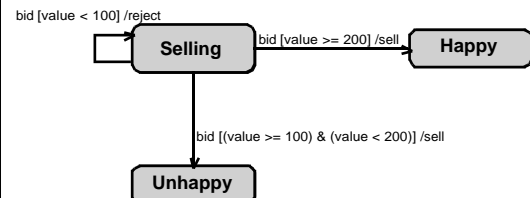


UML

23

Guardie

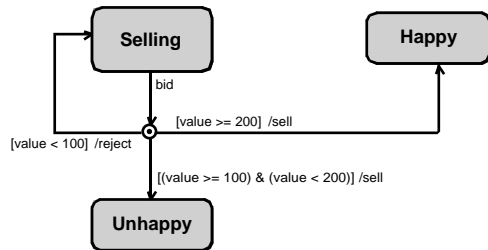
- Guardia:** Esecuzione di una transizione basata su un predicato booleano che non deve contenere side-effect



UML

24

Transizione condizionale statica

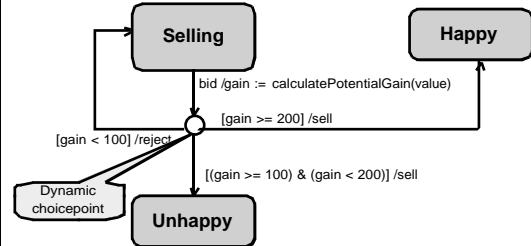


UML

25

Transizione condizionale dinamica

- Le guardie sono valutate nell'istante in cui è raggiunto il punto della decisione

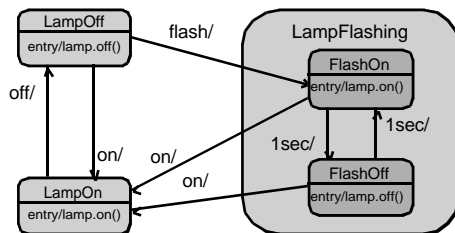


UML

26

Gerarchia di stati

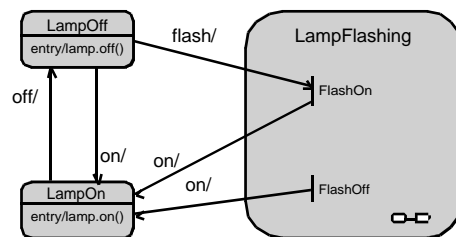
- Un modo per affrontare la complessità della modellazione



UML

27

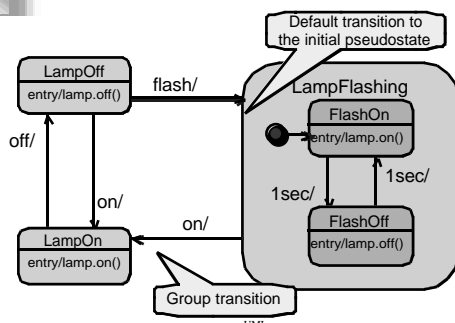
Gerarchia: notazione alternativa



UML

28

Gerarchia di transizioni : transizione di gruppo

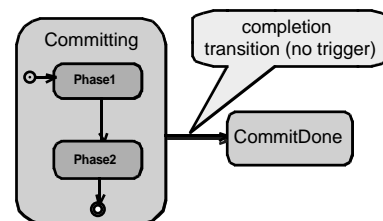


UML

29

Transizione di completamento

- Transizione generata automaticamente quando si giunge ad uno stato interno di terminazione

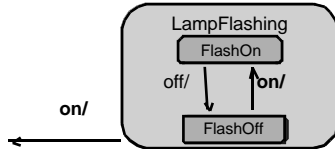


UML

30

Regole di Triggering

- Due o più transizioni possono avere gli stessi generatori:
 - la transizione più interna ha la precedenza
 - gli eventi sono considerati come svoltisi indipendentemente dal fatto che abbiano generato una transizione di stato

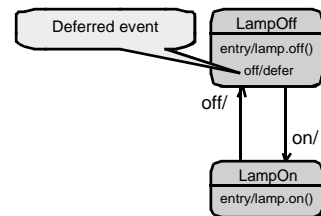


UML

31

Eventi differiti

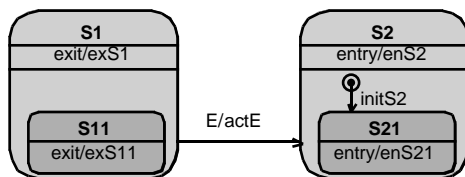
- Gli eventi possono essere memorizzati se non generano una transizione



UML

32

Ordinamento delle azioni: un'altro esempio



Sequenza di esecuzione delle azioni:

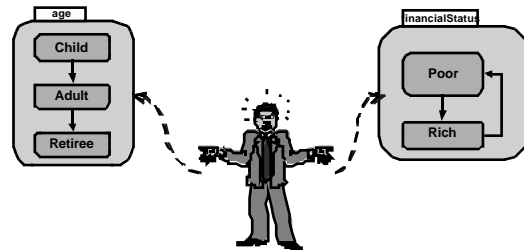
exS11 exS1 actE enS2 initS2 enS21

UML

33

Ortogonalità

- Molteplici simultanee prospettive riguardo la stessa entità

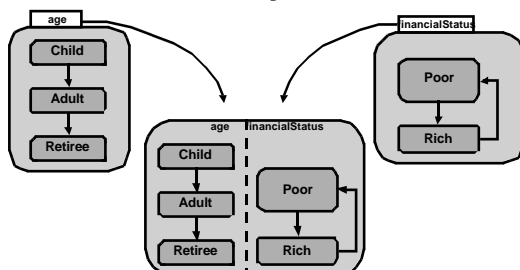


UML

34

Regioni Ortogonali (1/2)

- Per combinare molteplici descrizioni simultanee



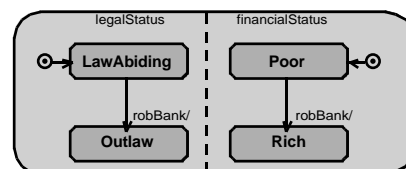
UML

35

Regioni Ortogonali (2/2)

- Semantica:**

Tutte le regioni **mutuamente ortogonali** gestiscono gli stessi eventi e reagiscono ad essi simultaneamente.

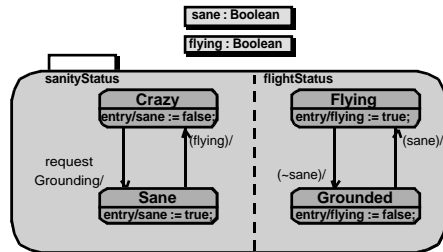


UML

36

Interazione tra regioni ortogonali

- Avviene tipicamente attraverso l'impiego di variabili condivise

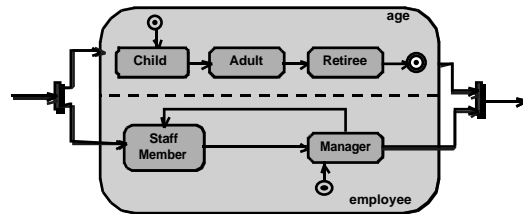


UML

37

Fork e Join di transizioni

- Per transizioni entranti/uscenti in/da regioni ortogonali:

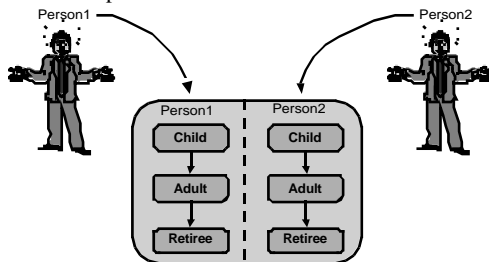


UML

38

Ortogonalità: errore frequente

- Utilizzare regioni per modellare oggetti indipendenti



UML

39

Conclusioni (1/2)

- Usa gli statechart diagram per modellare comportamenti basati sulla reazione ad eventi
 - particolarmente adatti al modello di server insito nel paradigma a oggetti
- Usa gli statechart diagram per modellare il comportamento di oggetti attivi guidati da eventi
 - sistemi modellati come reti di macchine a stati finiti collaboranti

UML

40

Conclusioni (2/2)

- Gli statechart diagram comprendono un certo numero di caratteristiche che mettono a disposizione i più comuni strumenti di utilizzo delle macchine a stati finiti:
 - azioni di entrata/uscita
 - l'attività svolta da uno stato
 - controllo di flusso
- Gli statechart diagram permettono una modellazione gerarchica utile per sistemi complessi:
 - gerarchia di stati
 - gerarchia di transizioni
 - ortogonalità

UML

41

Riferimenti

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

UML

42

Per ulteriori informazioni

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