

Unified Modeling Language

State Diagrams

1

State (o Statechart) Diagrams

- ◆ Specifica il ciclo di vita di un oggetto (istanza di classe, sistema)
- ◆ Rappresentano il comportamento dei singoli oggetti in termini di
 - Eventi a cui gli oggetti (la classe) sono sensibili
 - Azioni prodotte
 - Transizioni di stato
 - » Identificazione degli stati interni degli oggetti
- ◆ Possibilità di descrivere evoluzioni parallele
- ◆ Sintassi mutuata da StateChart (D. Harel)

2

Il diagramma degli stati e delle transizioni

Il diagramma degli stati e delle transizioni viene definito **per una classe**, ed intende descrivere l'**evoluzione tipica di un oggetto** generico di quella classe.

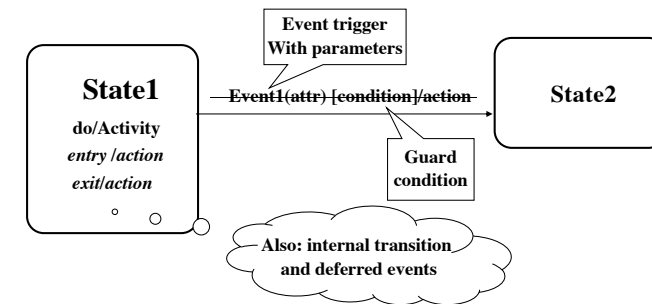
Uno **stato** rappresenta una situazione in cui un oggetto ha un insieme di proprietà considerate stabili

Una **transizione** modella un cambiamento di stato ed è denotata da:

Evento [Condizione] / Azione

3

UML Statechart Diagram Notation



- ◆ Notation based on work by Harel
 - Added are a few object-oriented modifications
- ◆ A UML statechart diagram can be mapped into a finite state machine

4

- State Diagram

Statechart Diagrams

- ◆ Graph whose nodes are states and whose directed arcs are transitions labeled by event names.
- ◆ We distinguish between two types of operations in statecharts:
 - Activity: Operation that takes time to complete
 - » associated with states
 - Action: Instantaneous operation
 - » associated with events
 - » associated with states (reduces drawing complexity): Entry, Exit Action
- ◆ A statechart diagram relates events and states for *one class*
 - An object model with a set of objects has a set of state diagrams

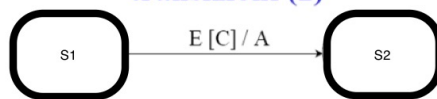
5

State

- ◆ An abstraction of the attributes of a class
 - State is the aggregation of several attributes a class
- ◆ Basically an equivalence class of all those attribute values and links that do not need to be distinguished as far as the control structure of the system is concerned
 - Example: State of a bank
 - » A bank is either solvent or insolvent
- ◆ State has duration

6

Il diagramma degli stati e delle transizioni (2)



Il significato di una transizione del tipo di quella qui mostrata è:

- Se l'oggetto si trova nello stato S1, e
- Se si verifica l'evento E, e
- Se la condizione C è verificata
- Allora viene eseguita l'azione A e l'oggetto passa nello stato S2.

7

7

Stato iniziale e finale

- Una freccia non etichettata che parte dal “vuoto” ed entra in uno stato indica che lo stato è **iniziale**
- Una freccia non etichettata che esce da uno stato e finisce nel “vuoto” indica che lo stato è **finale**
- Stato iniziale e finale possono anche essere denotati da appositi simboli

stato iniziale

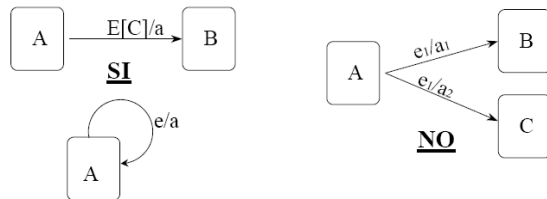
stato finale

8

8

Transizione

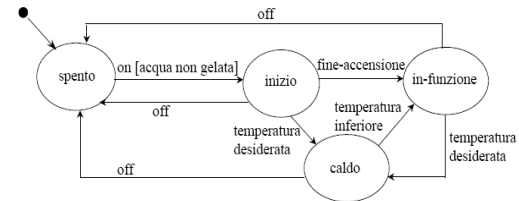
- Ogni transizione connette due stati
- Il diagramma corrisponde ad un automa **deterministico** (transizioni dallo stesso stato hanno eventi diversi), in cui un evento è un input, mentre un'azione è un output
- La condizione è detta anche "guardia" (*guard*)
- L'evento è (quasi) sempre presente (condizione e azione sono opzionali)



9

Esempio di diagramma degli stati e delle transizioni per la classe Caldaia

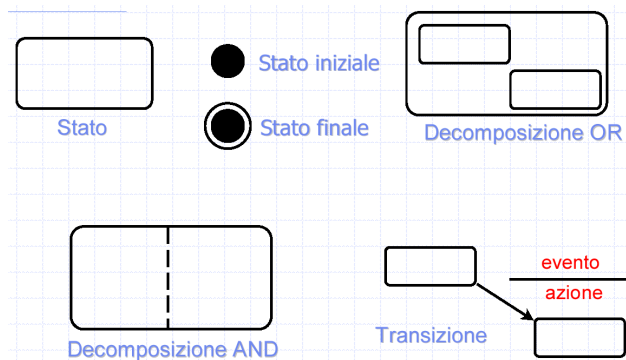
Descriviamo il diagramma degli stati e delle transizioni relativa ad una classe "Caldaia". In questo diagramma ogni transizione è caratterizzata solamente da eventi e condizioni (i cambiamenti di stato non hanno bisogno di azioni perché sono automatici)



Nota: gli stati dovrebbero essere rettangoli arrotondati

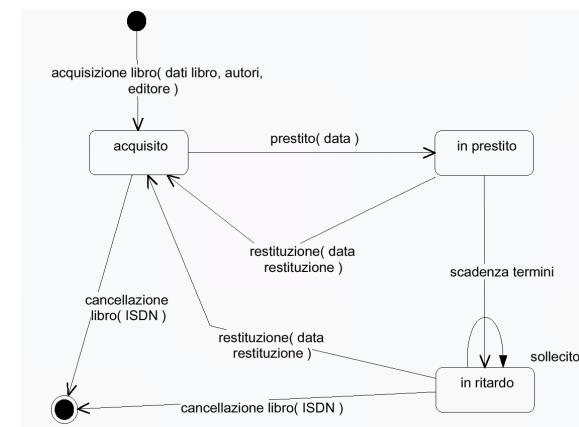
10

Elementi grafici



11

Esempio: Libro



12

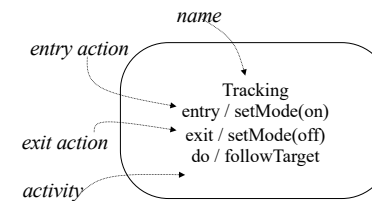
Stato: Altre caratteristiche

- ◆ Stato: Situazione in cui l'oggetto soddisfa qualche condizione, esegue qualche attività o aspetta qualche condizione
- ◆ Attributi (opzionali):
 - **Nome:** una stringa; uno stato può essere anonimo
 - **Entry / Exit actions:** eseguite all'ingresso / uscita dallo stato (non interrompibili, durata istantanea)
 - **Transizioni interne:** non causano un cambiamento di stato
 - **Attività** dello stato (interrompibile, durata significativa)
 - **Sottostati:** struttura innestata di stati; disgiunti (sequenzialmente attivi) o concorrenti (concorrentemente attivi)

14

14

Stato



15

15

Transizioni: altre caratteristiche

- ◆ Cambiamento da uno stato iniziale a uno finale
 - transizione esterna (stato finale diverso da stato iniziale)
 - interna (stato finale uguale a stato iniziale)
- ◆ Attributi (opzionali): evento [guardia] / azione
 - Evento
 - » segnale, messaggio da altri oggetti, passaggio del tempo, cambiamento
 - Condizione di guardia
 - » La transizione occorre se l'evento accade e la condizione di guardia è vera
 - Azione
 - » E' eseguita durante la transizione e non è interrompibile (durata istantanea)
 - Transizioni senza eventi (triggerless) scattano
 - » con guardia: se la condizione di guardia diventa vera
 - » senza guardia: se l'attività interna allo stato sorgente della transizione è completata

16

16

Esempio di diagramma degli stati e delle transizioni per la classe Motore

L'analisi dei requisiti ha evidenziato l'esistenza, nel diagramma delle classi, di una classe "Motore".

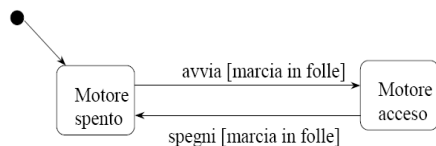
Tracciare il diagramma degli stati e delle transizioni a partire da questi requisiti.

Un motore di automobile può essere spento o acceso, ma può essere avviato o spento solo se la marcia è in folle

17

17

Esempio di diagramma degli stati e delle transizioni per la classe Motore (2)



18

18

Esercizio

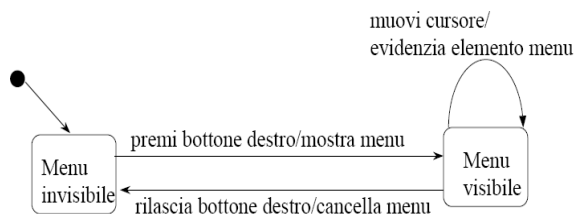
Supponiamo che nel diagramma delle classi abbiamo rappresentato la classe "Menu a tendina". Tracciare il diagramma degli stati e delle transizioni per tale classe, tenendo conto delle seguenti specifiche.

Un menu a tendina può essere visibile oppure no. Viene reso visibile a seguito della pressione del tasto destro del mouse, e viene reso invisibile quando tale tasto viene sollevato. Se si muove il cursore quando il menu è visibile, si evidenzia il corrispondente elemento del menu.

19

19

Esercizio 1: soluzione



20

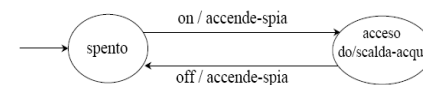
20

Il diagramma degli stati e delle transizioni (3)

- Alcune volte vogliamo rappresentare dei processi che l'oggetto esegue senza cambiare stato. Questi processi si chiamano **attività**, e si mostrano negli stati con la notazione:

do / attività

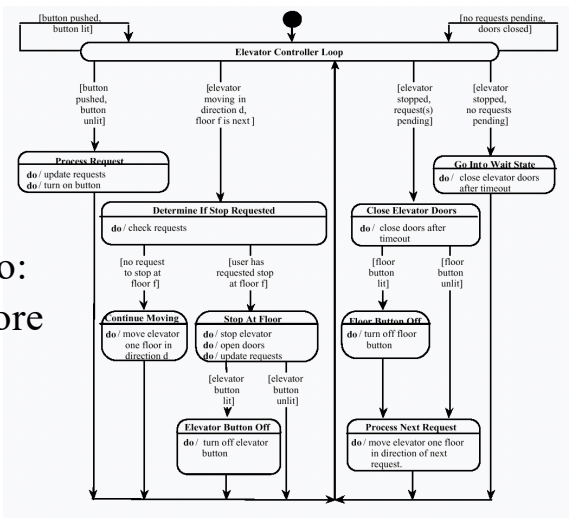
Esempio:



21

21

Esempio: Ascensore



22

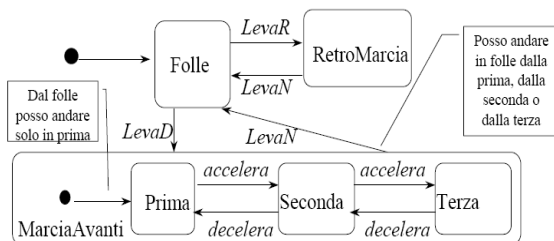
Nested State Diagram

- ◆ Activities in states are composite items denoting other lower-level state diagrams
- ◆ A lower-level state diagram corresponds to a sequence of lower-level states and events that are invisible in the higher-level diagram.
- ◆ Sets of substates in a nested state diagram denote a **superstate** are enclosed by a large rounded box, also called contour.

23

Stato composto

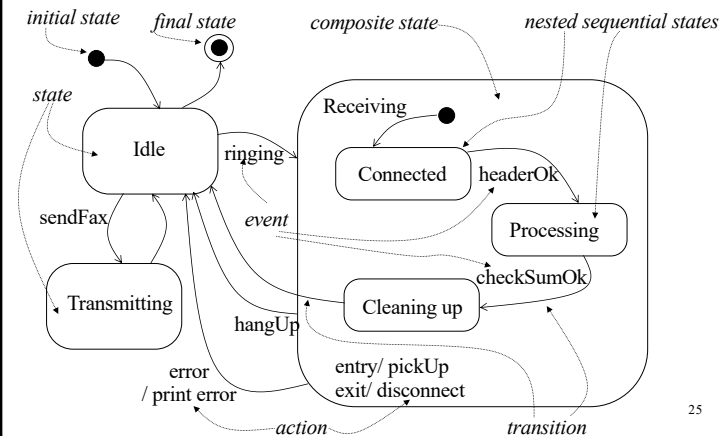
- Uno stato composto (o macro-stato) è uno stato che ha un nome, e che contiene a sua volta un diagramma
- Esiste uno stato iniziale del macro-stato
- I sottostati **ereditano** le transizioni in uscita del macro-stato



24

24

Esempio: Fax



25

25

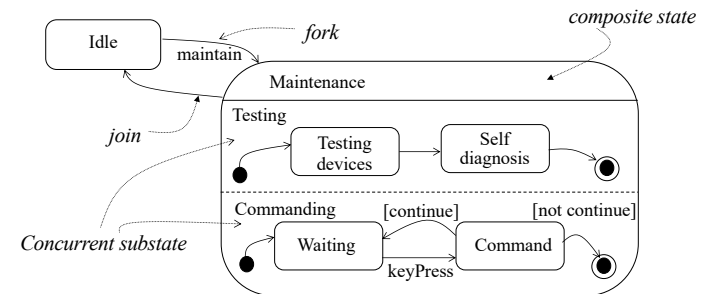
- State Diagram

Concurrent substates (decomposizione AND)

- ◆ Più state machine sono eseguite in parallelo entro lo stato che li racchiude
- ◆ Se un substate machine raggiunge lo stato finale prima dell'altro, il controllo aspetta lo stato finale dell'altro
- ◆ Quando avviene una transizione in uno stato con concurrent substate, il flusso di controllo subisce un fork per ciascun concurrent substate; alla fine esso si ricompone in un unico flusso con un join

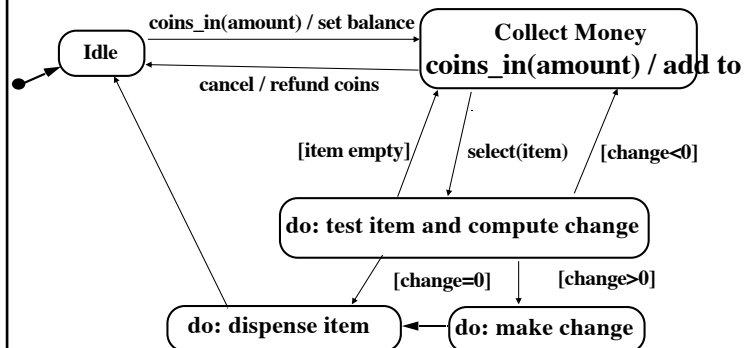
26

Esempio



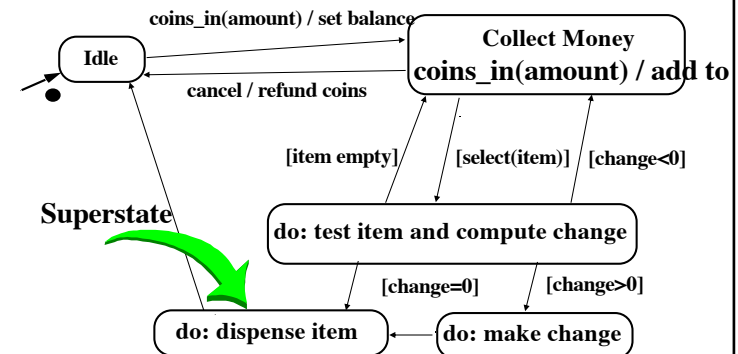
27

Example of a StateChart Diagram



28

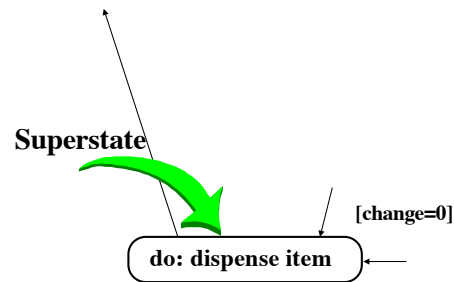
Example of a Nested Statechart Diagram



29

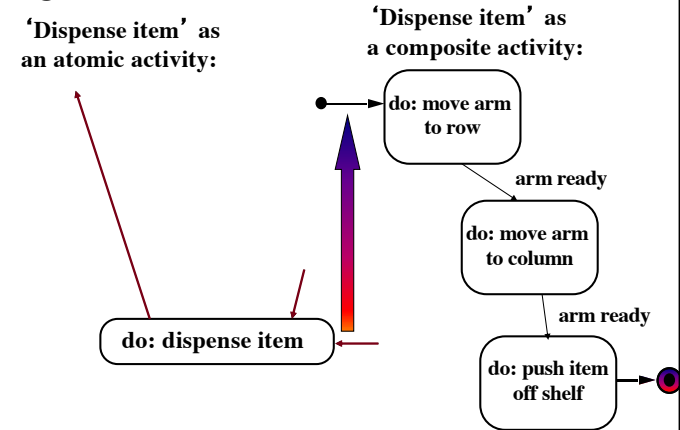
- State Diagram

Example of a Nested Statechart Diagram



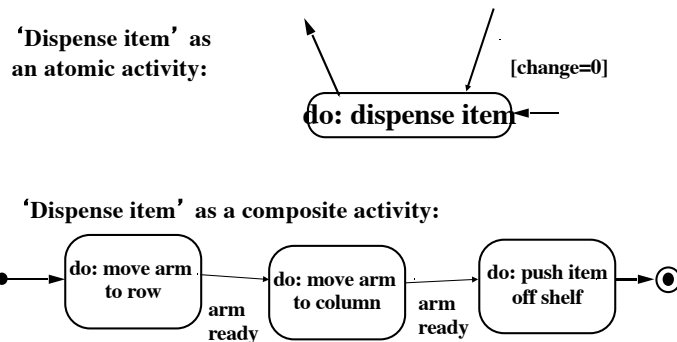
30

Example of a Nested Statechart Diagram



31

Expanding activity "do:dispense item"



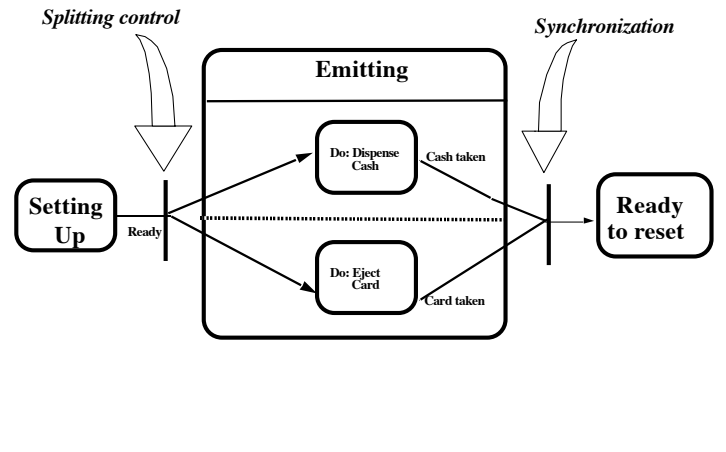
32

Superstates

- ◆ Goal:
 - Avoid spaghetti models
 - Reduce the number of lines in a state diagram
- ◆ Transitions from other states to the superstate enter the first substate of the superstate.
- ◆ Transitions to other states from a superstate are inherited by all the substates (state inheritance)
 - ... except in the case of transitions that have to be executed at the end of the activities associated with the state (like in the previous example)

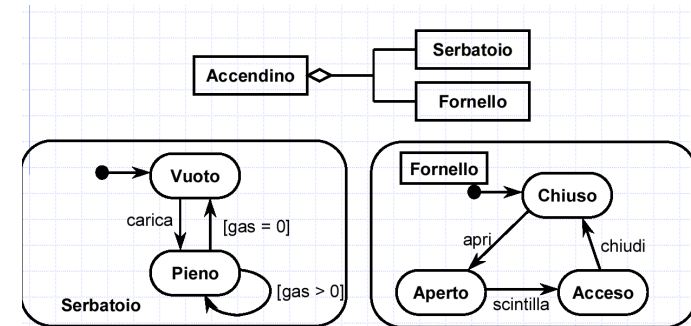
33

Example of Concurrency within an Object



35

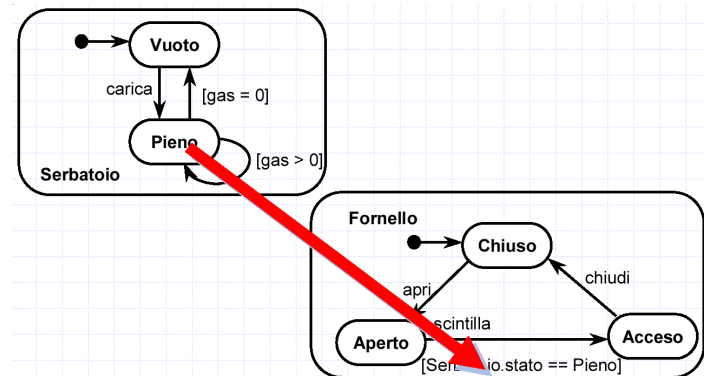
Oggetti Compositi



36

36

Interazioni tra Componenti



37

37

State Chart Diagram vs Sequence Diagram

- ◆ State chart diagrams help to identify:
 - *Changes* to an individual object over time
- ◆ Sequence diagrams help to identify
 - The *temporal relationship* of between objects over time
 - *Sequence of operations* as a response to one ore more events

38

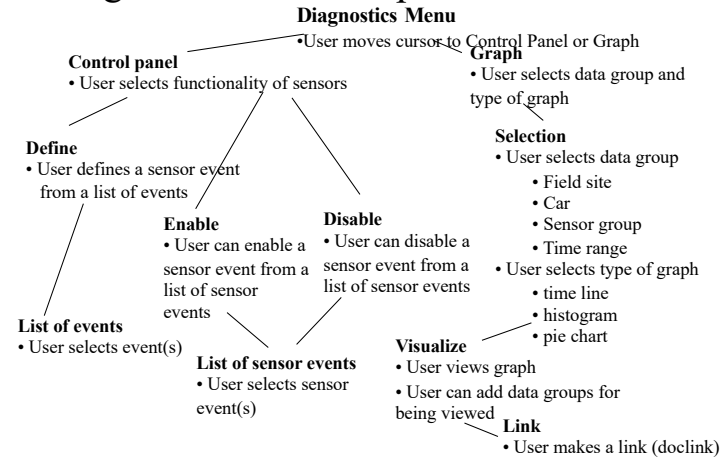
- State Diagram

Dynamic Modeling of User Interfaces

- ◆ Statechart diagrams can be used for the design of user interfaces
 - Also called Navigation Path
- ◆ States: Name of screens
 - Graphical layout of the screens associated with the states helps when presenting the dynamic model of a user interface
- ◆ actions are shown as bullets under screen name
 - Often only the action is shown
- ◆ State transitions: Result of human action
 - Button click
 - Menu selection
 - Cursor movements
- ◆ Good for web-based user interface design

42

Navigation Path Example



NOT a good UML diagram! Syntax not respected

43