

UML1.5

Dynamic Modeling (State Machine)

Koichiro OCHIMIZU

School of Information Science

JAIST

Schedule(2/3)





- Feb. 27th
 - 13:00 Introduction to Java Programming
 - 14:30 Outline of UML: Static Modeling
(usecase modeling, details of class definition)
- Feb. 28th
 - 13:00 **Outline of UML: Dynamic Modeling
(state machine)**
 - 14:30 Outline of UML: Dynamic Modeling
(communication diagram, sequence diagram)

Dynamic Model

- **State Diagram**
 - Describe which states an object can have during its life cycle
- **Sequence Diagram**
 - Describe how objects interact and communicate with each other
 - The primary focus in sequence diagrams is time
- **Collaboration Diagram**
 - Describe how objects interact
 - But the focus in a collaboration diagram is space

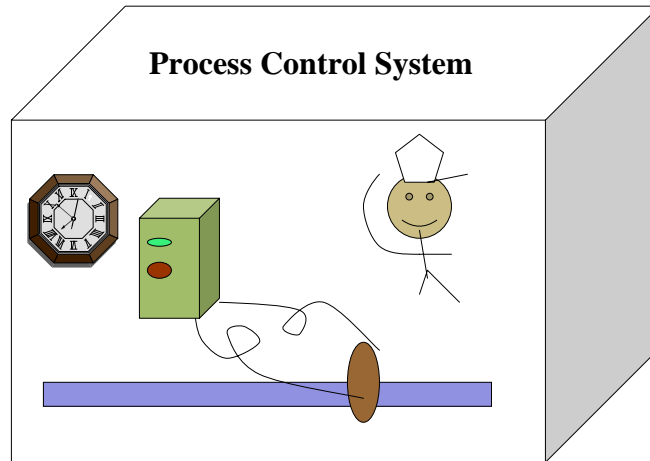
H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

Object Interaction (Message)

- **Synchronous** 
 - procedure call or other nested flow of control
- **Return** 
 - return from a procedure call
- **Simple** 
 - Flat flow of control
- **Asynchronous** 
 - Asynchronous flow of control.

H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

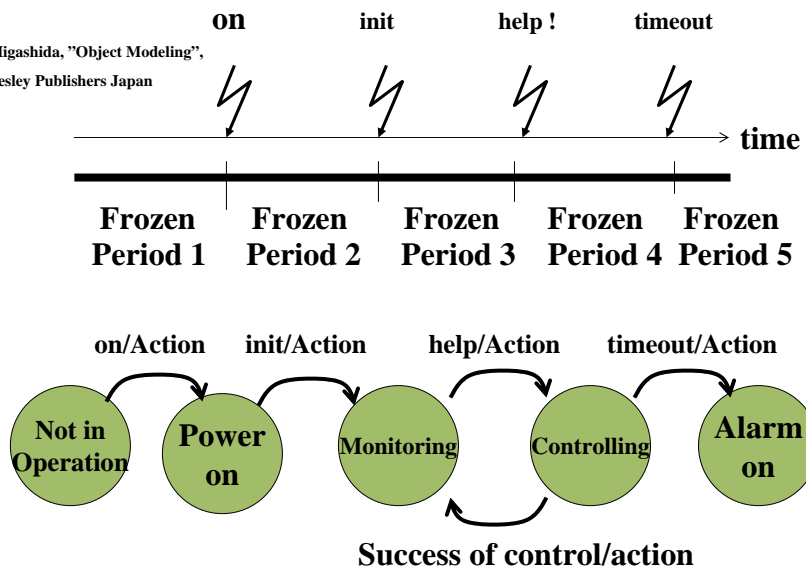
Defining behavior of the system by a state diagram



Ochimizu, Higashida, "Object Modeling", Addison-Wesley Publishers Japan

What is a State ?

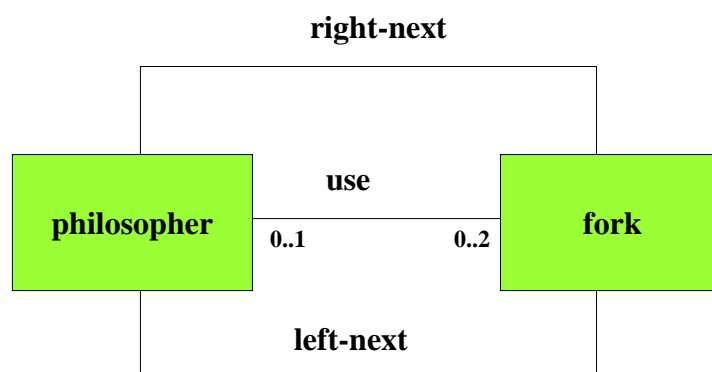
Ochimizu, Higashida, "Object Modeling",
Addison-Wesley Publishers Japan



State Model

- Represent the behavior of an object by a state transition diagram.

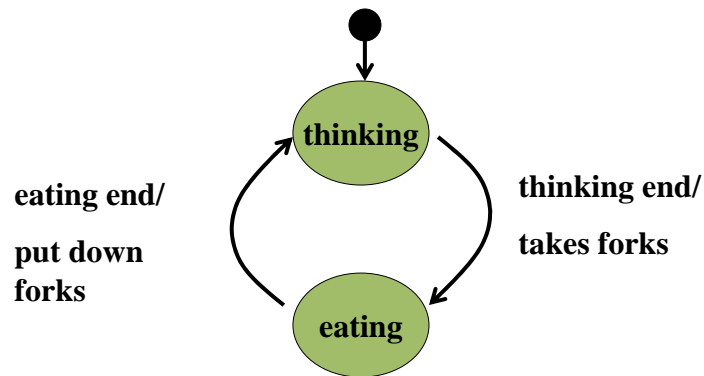
Defining the behavior of philosopher and fork



Each philosopher must use right next fork and left next one not used

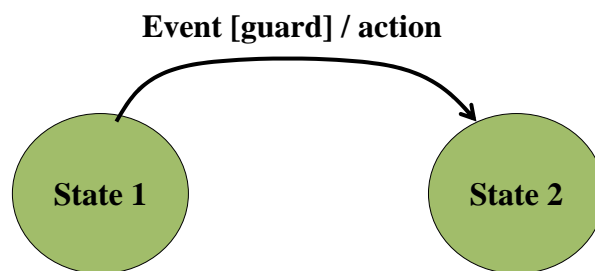
Ochimizu, Higashida, "Object Modeling", Addison-Wesley Publishers Japan

Behavior of a Philosopher

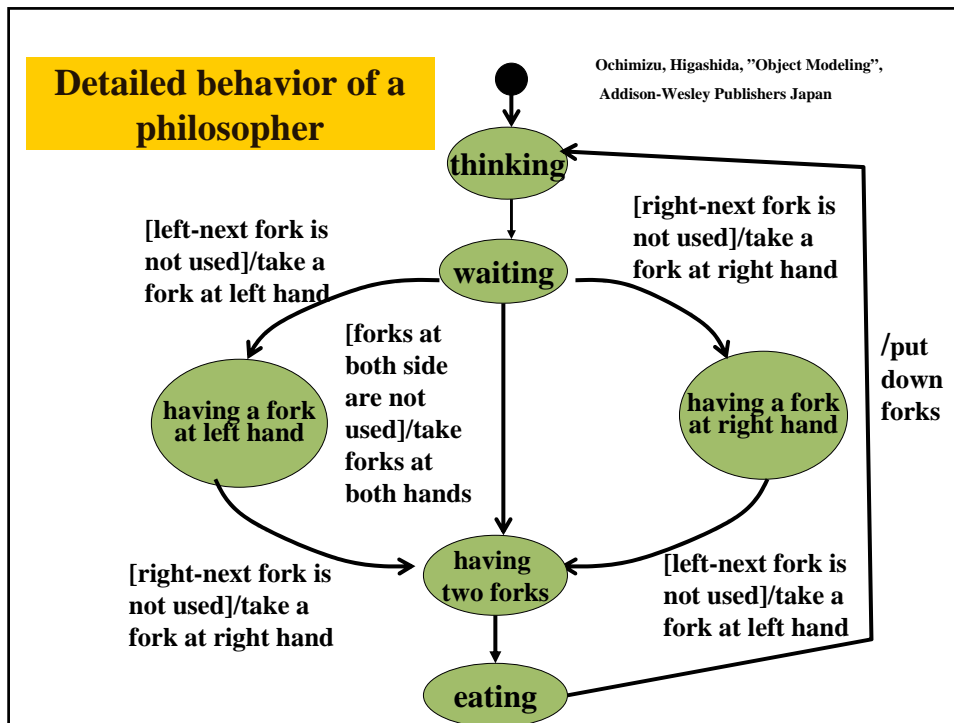


Ochimizu, Higashida, "Object Modeling", Addison-Wesley Publishers Japan

State transition depends on the state of forks both side



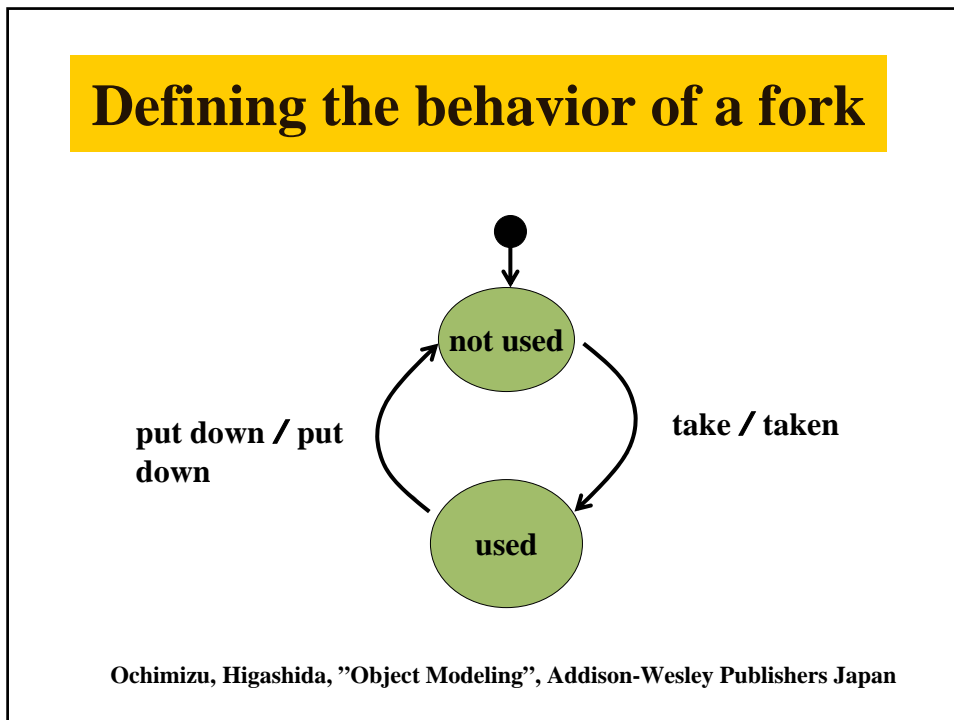
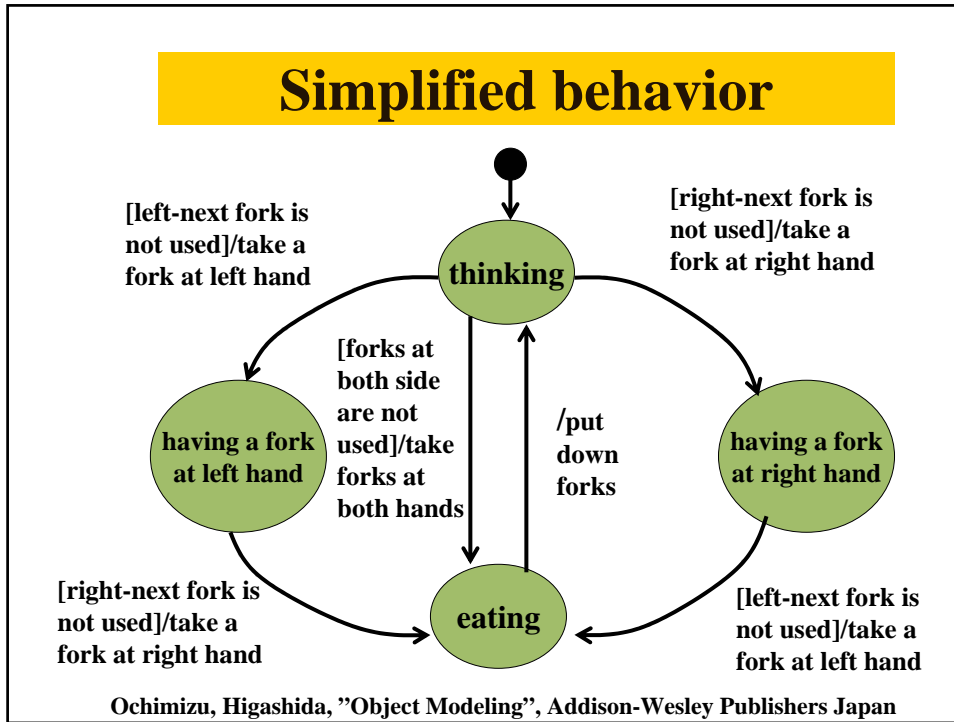
Ochimizu, Higashida, "Object Modeling", Addison-Wesley Publishers Japan

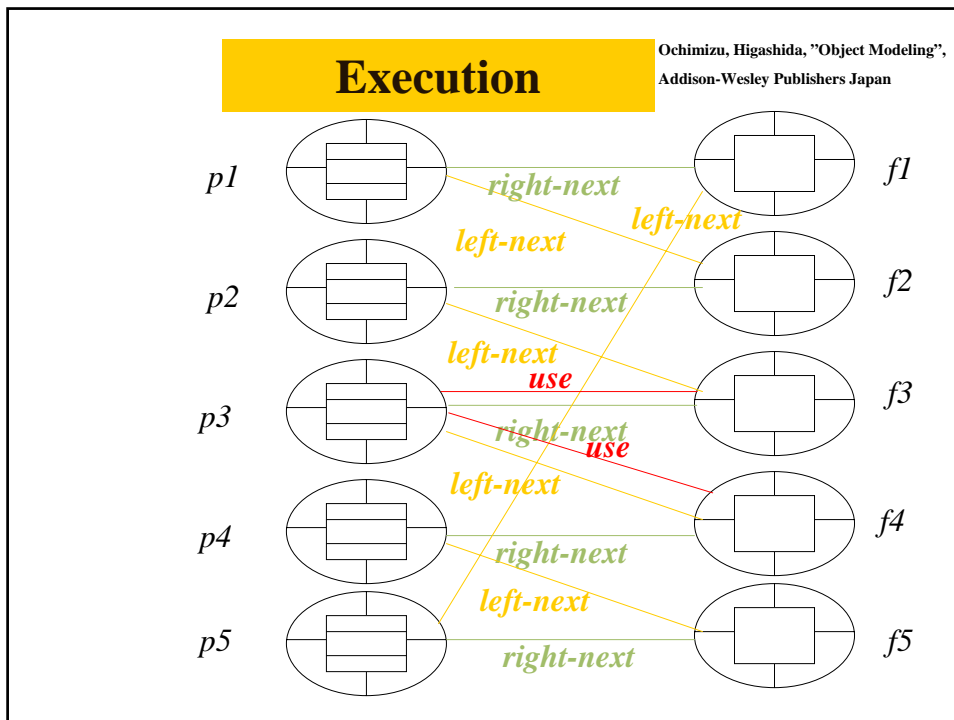
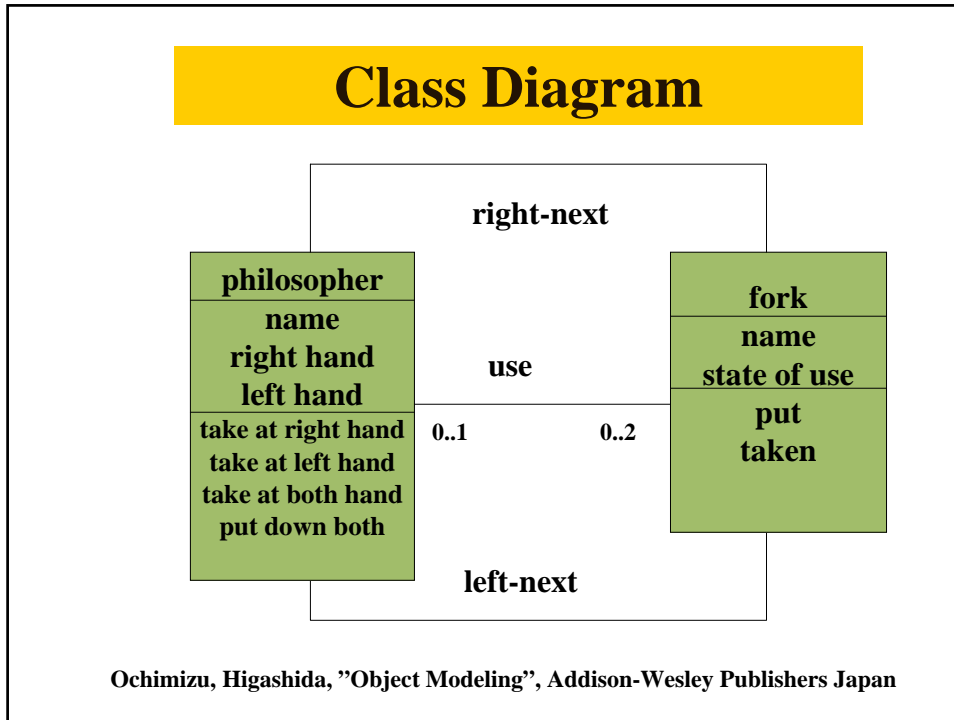


Passive Object and Active Object

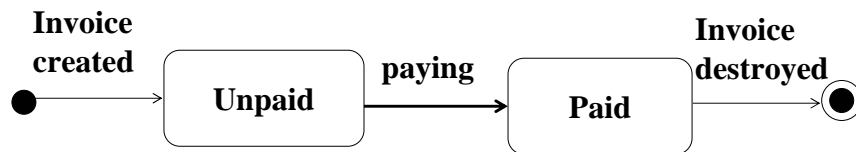
- **Passive Object**
 - It is activated when it receives a message from other object (work when being hit)
- **Active Object**
 - It can change its state by itself and send a message to other object if necessary
- **In the case of philosopher**
 - **Philosopher's state of hunger changes autonomously during thinking and eating.**

Ochimizu, Higashida, "Object Modeling", Addison-Wesley Publishers Japan



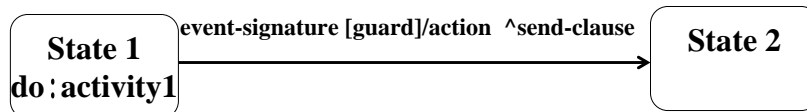


Initial State and Final State



H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

State Transition



event-signature: event-name (parameter list)

draw (f: Figure, C: color)

draw()

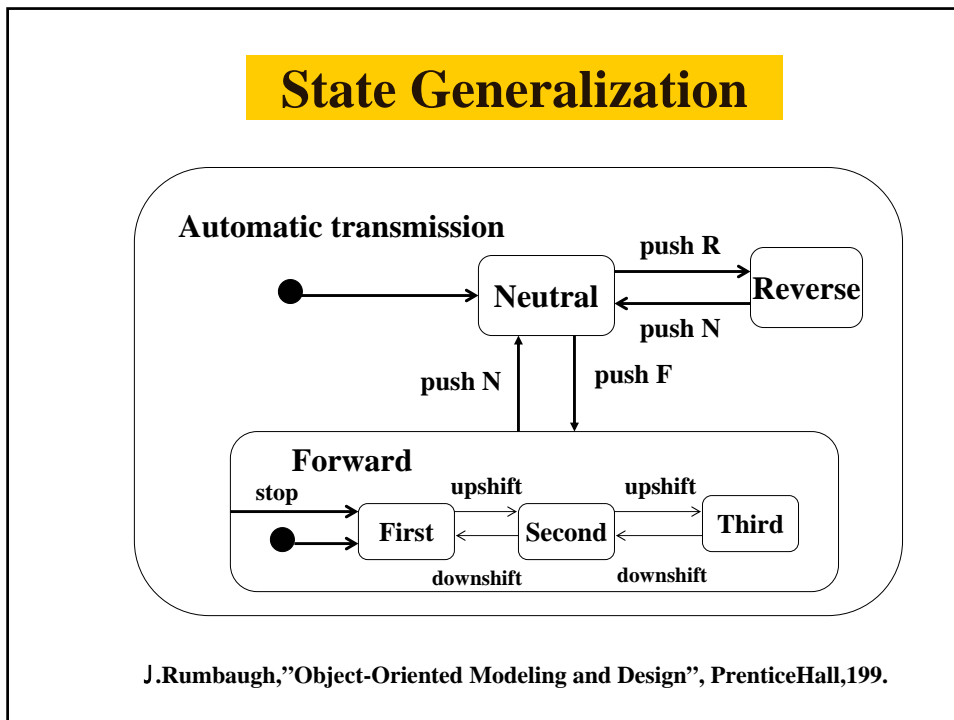
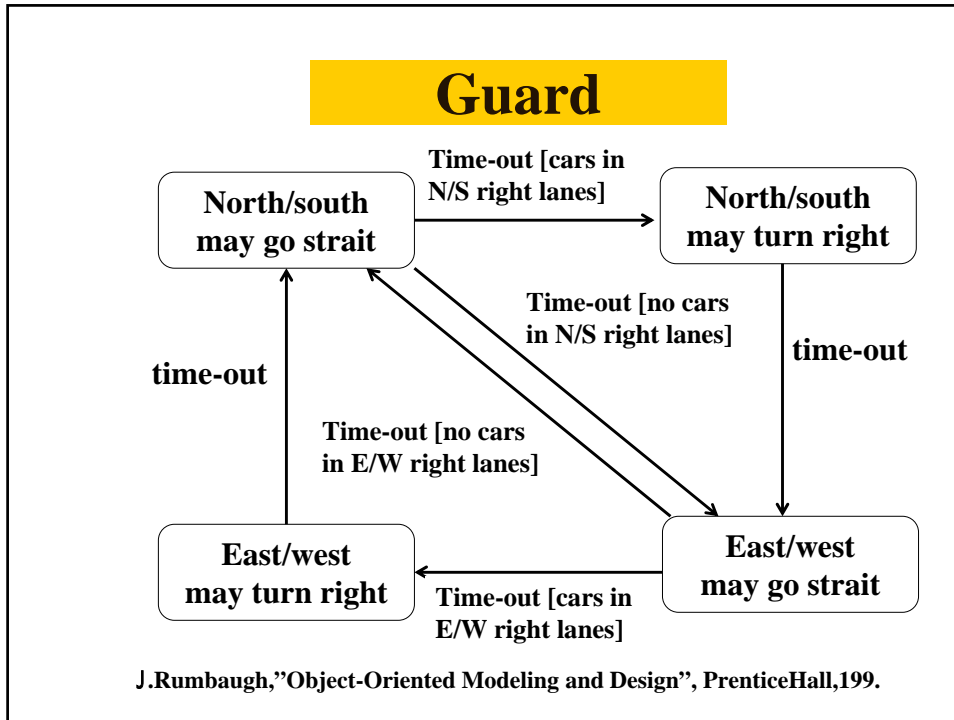
send-clause: destination-expression . Destination-event-name (argument list)

message sent during transition

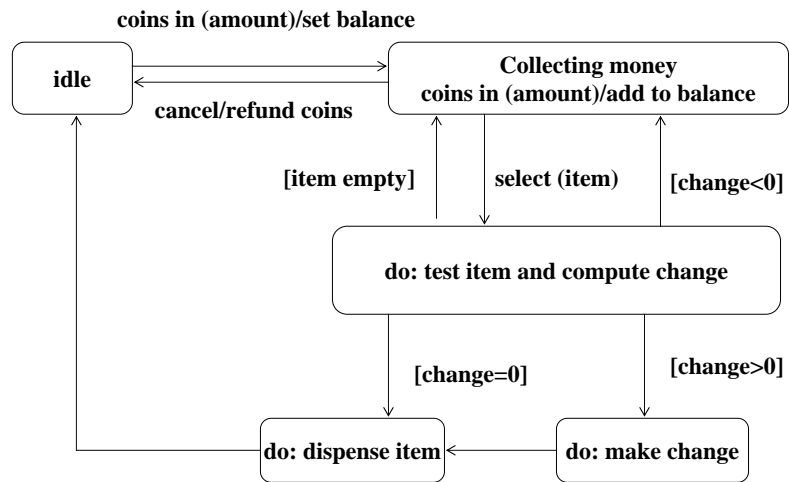
[timer=time-out] ^ self.go down (first floor)

destination-expression: object or a series of objects

H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

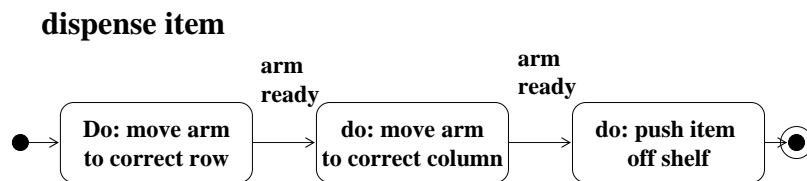


Automatic Transition



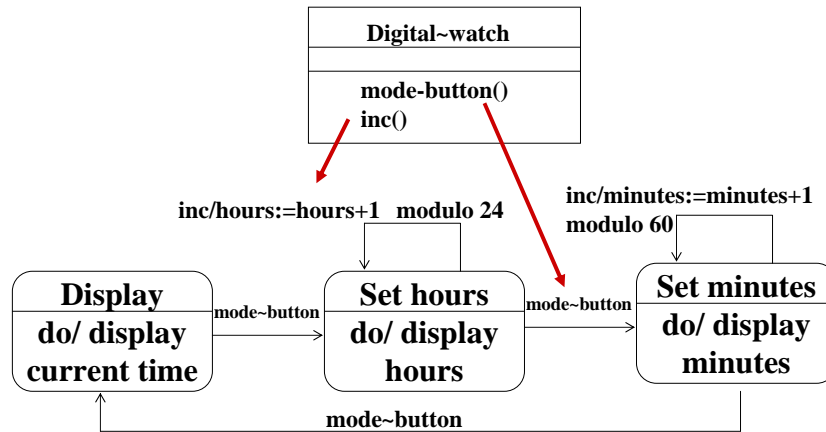
J.Rumbaugh,"Object-Oriented Modeling and Design", PrenticeHall,199.

Sub diagram



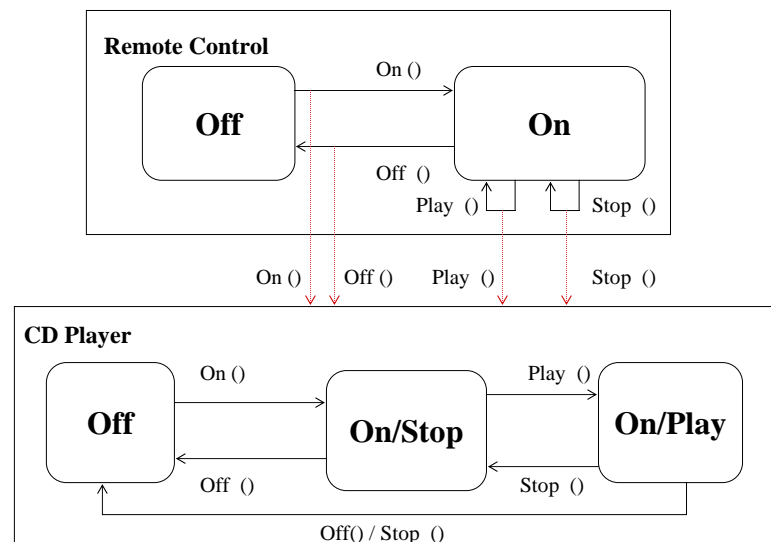
J.Rumbaugh,"Object-Oriented Modeling and Design", PrenticeHall,199.

Events in the state diagram correspond with operations within the class



H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

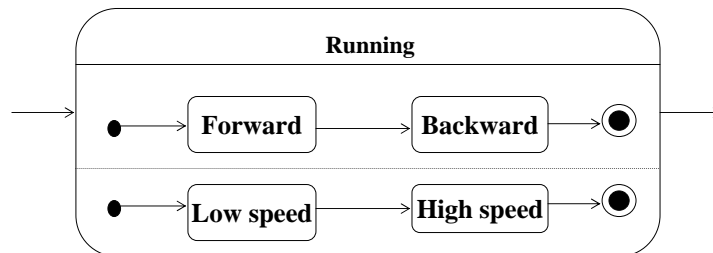
Message passing between two state diagram



H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

And-substate

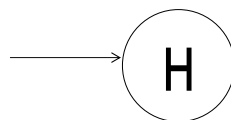
And-substate



H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.

History Indicator

- is used to memorize internal state
- Support roll-back
- If a transition to the indicator fires, the object resumes the state it had last within that region
- Shown as a circle with an H



H.E. Eriksson and M. Penker, "UML Toolkit" John Wiley & Sons, Inc.