

Adapted for a textbook by Blaha M. and Rumbaugh J.

# Object Oriented Modeling and Design

Pearson Prentice Hall, 2005

## STATE MODELING

Remigijus GUSTAS

Phone: +46-54 700 17 65

E-mail: [Remigijus.Gustas@kau.se](mailto:Remigijus.Gustas@kau.se)

<http://www.cs.kau.se/~gustas/>

# Events

- ✓ Event is noteworthy change in object state.
- ✓ Event is an occurrence at a point in time.
- ✓ Events are happening instantaneously with regards to a time scale of an application.
- ✓ Events can be
  - causally unrelated (concurrent),
  - precede each other,
  - follow each other.
- ✓ Concurrent events can occur in any order.

# Event Type and Event Instance

- ✓ Event type is a classification event
  - Flight Departure
  - SAS flight SK64 from Stockholm on 2006.01.12.
- ✓ The most common events:
  - Signal event,
  - Change event,
  - Temporal event.

# Signal Event

«signal» <b>FlightDeparture</b>
airline flightNumber city date

«signal» <b>MouseButtonPushed</b>
button location

«signal» <b>StringEntered</b>
text

«signal» <b>ReceiverLifted</b>

«signal» <b>DigitDialed</b>
digit

- ✓ A signal event is event of sending or receiving information.
- ✓ A signal is a one way message from one object to another.
- ✓ Most signal classes have attributes indicating values they convey.

# Time Event

- ✓ A time event is an event caused by the elapse of a time interval or by matching an absolute time point.
  - Expression is continuously tested,
  - Whenever it changes from false to true, the event happens
- ✓ UML notation for a change event (***after*** or ***when*** followed by expression that evaluates a condition)

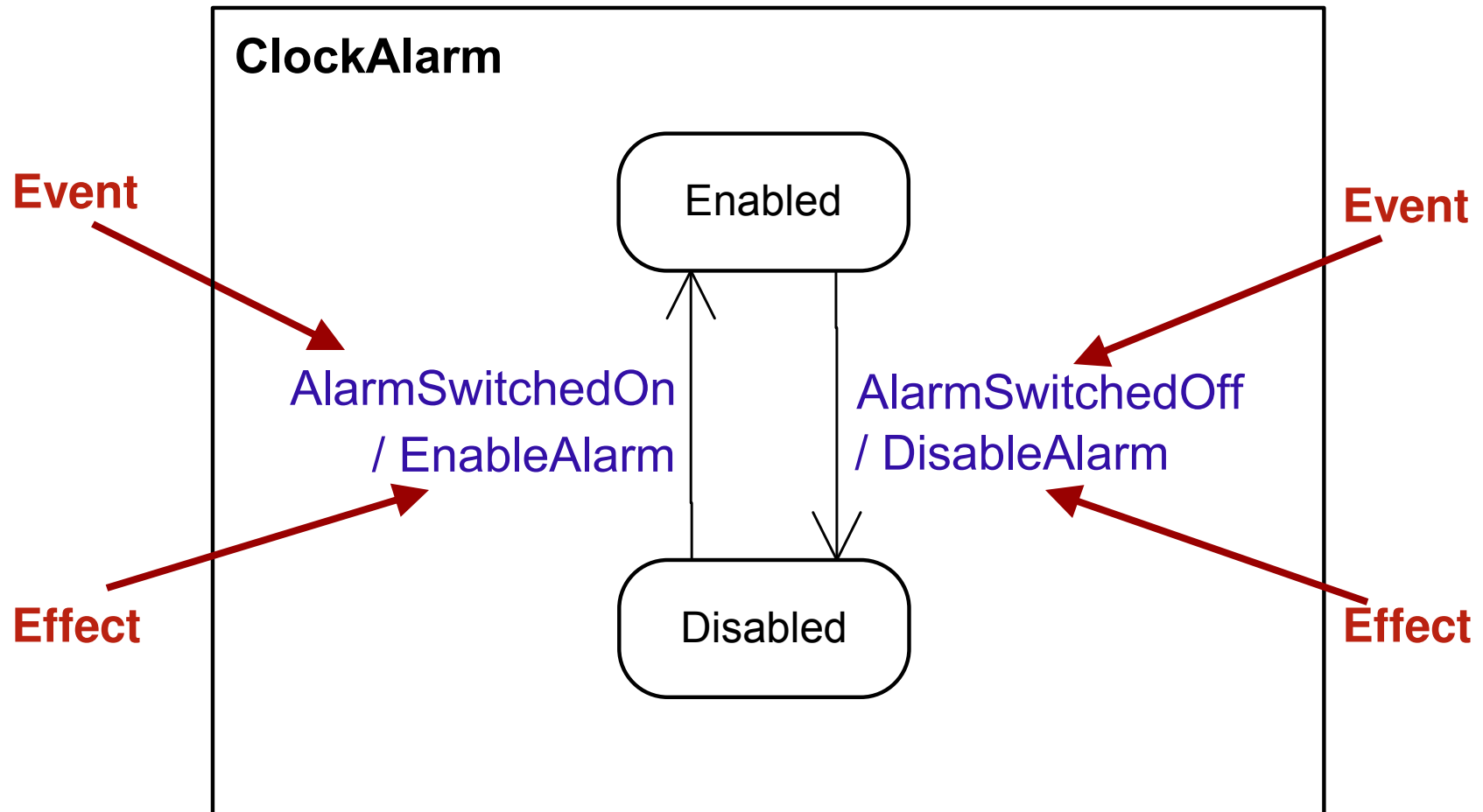
- when (date = January 1, 2000)
- after (10 seconds)

# Change Event

- ✓ A change event is an event that is caused by satisfaction of a boolean expression.
  - The condition must be checked often enough so that it seems continuous from the application perspective
- ✓ UML notation for a change event (***when*** followed by boolean expression)

- when (room temperature < heating set point)
- when (room temperature > cooling set point)
- when (battery power < lower limit)
- when (tire pressure < minimum pressure)

# Finite-State Machine (FSM)



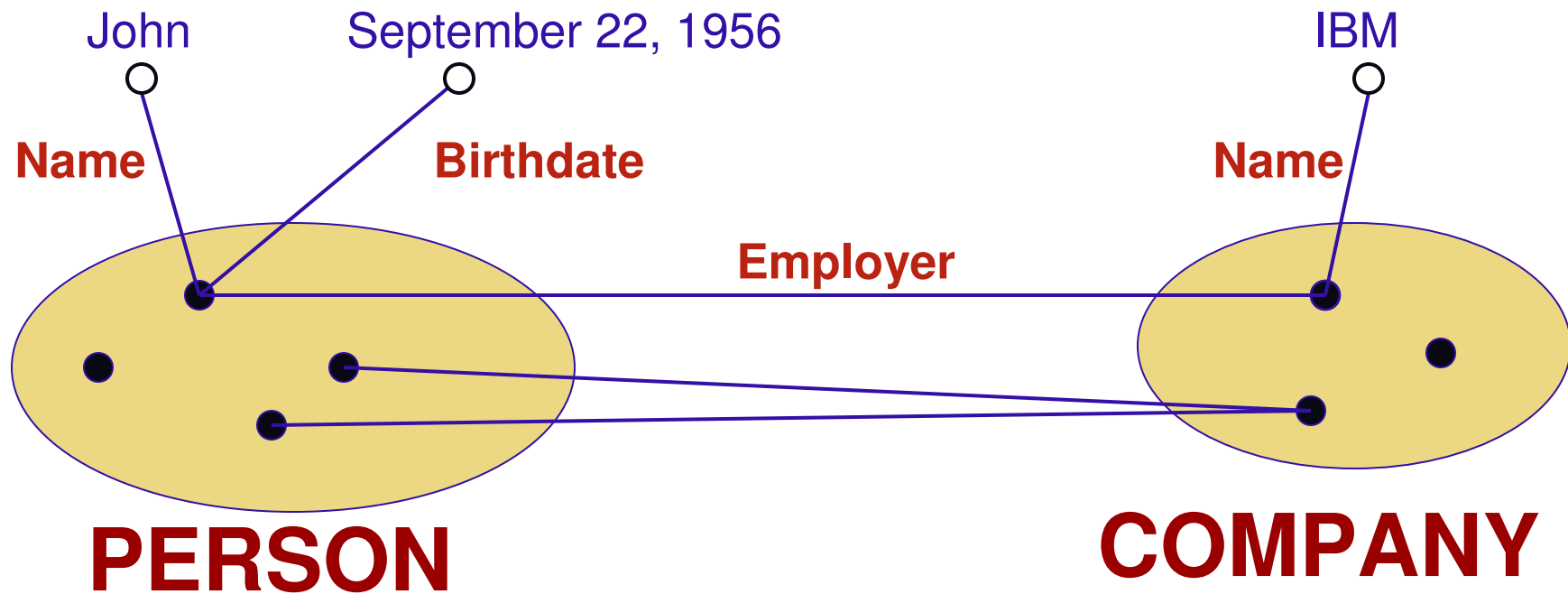
# Finite-State Machine (cont.)

- ✓ FSM is a hypothetical machine that can exist in a finite number of states in a given moment.
- ✓ Each FSM proceeds in discrete steps from one state to another.
- ✓ Each FSM has a finite set of events that can reach it and trigger state change.
- ✓ When operation is completed successfully, two things occur: state change of object and generation of one or more activity effects.



# What is Object State ?

- ✓ The collection of all relations an object has with attribute values and other objects.

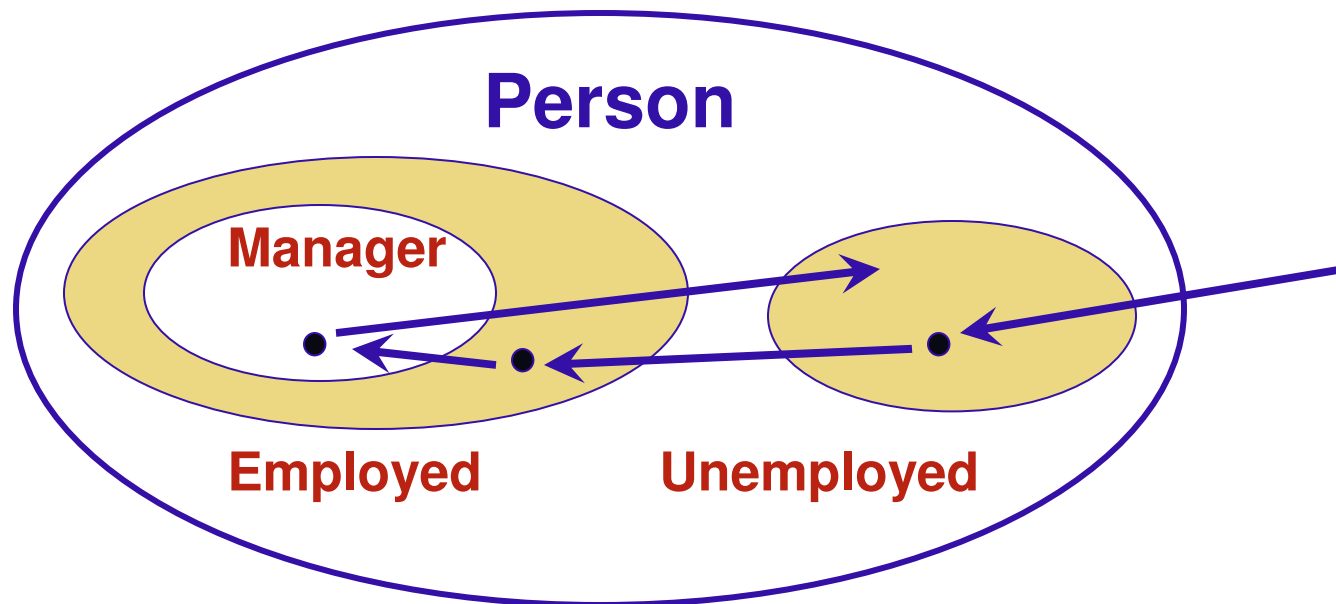


# States

- ✓ A state is an abstraction of the values and links (association instances) of an object.
- ✓ A state can be defined by a set of attribute values and links to other objects.
- ✓ In defining states, we ignore attributes that do not affect the object behavior.
- ✓ Objects have a finite number of possible states.
- ✓ Each object can only be in one state at a time.

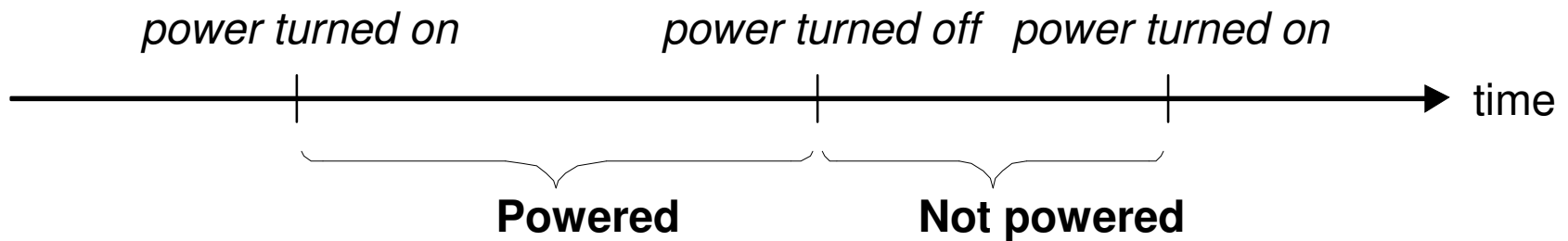
# Primary purpose is to access or change state of object by:

- ✓ Creation/Termination operation,
- ✓ Access operation,
- ✓ Modification operation.



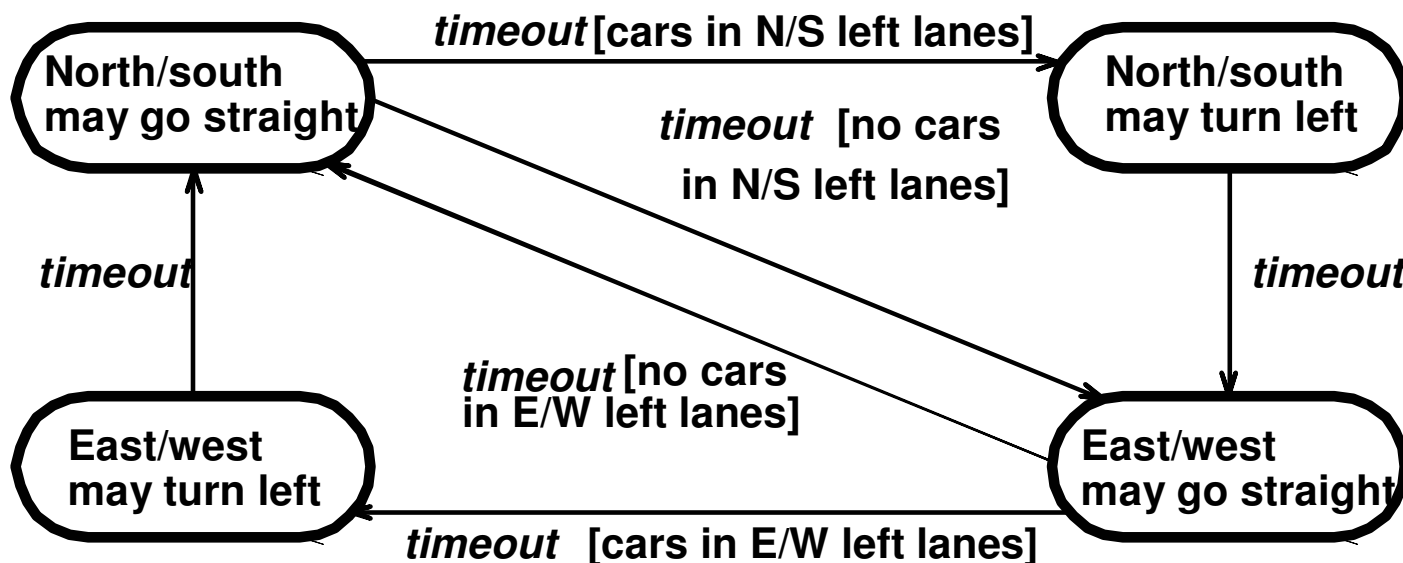
# States and Events

- ✓ There is a certain symmetry between events and states:
  - Events represent points in time
  - States represent intervals of time



# Transitions and Conditions

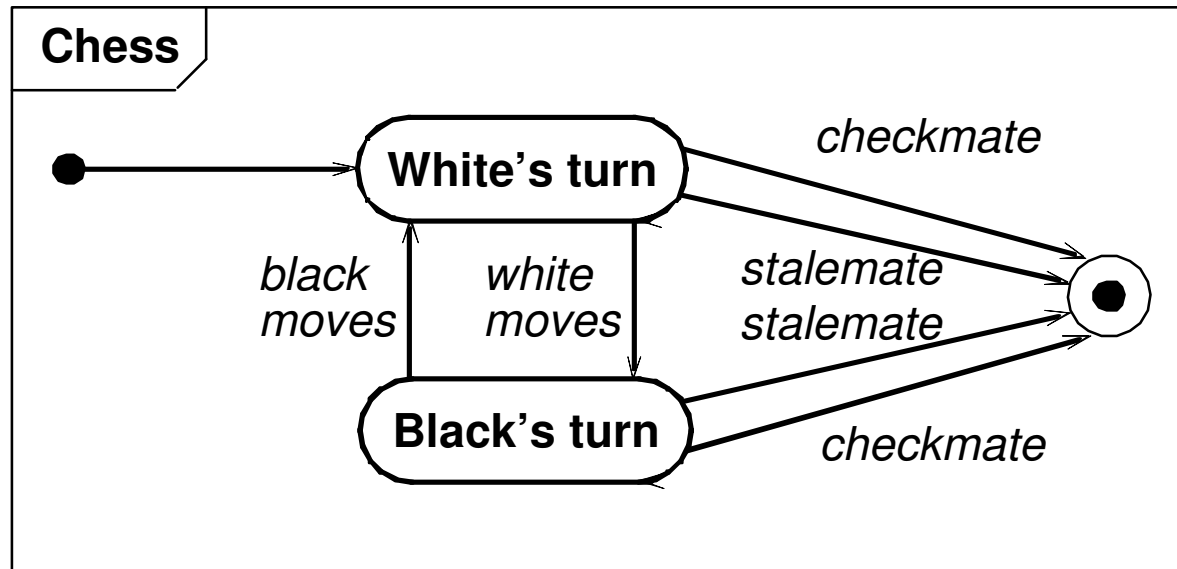
- ✓ Transition is an instantaneous change from one state to another.
- ✓ The transition is fired upon change from the source state to the target state.
- ✓ A guard condition is boolean expression that must be true for a transition to occur.



# State Diagrams

- ✓ A class with more than one state has important temporal behavior.
- ✓ State diagram is defined for one class of objects.
- ✓ All objects in a class execute state diagram for that class, which represents their common behavior.
- ✓ State model consists of multiple state diagrams.
- ✓ State diagrams can be directly interpreted or implemented into the programming code.

# State Diagrams (cont.)



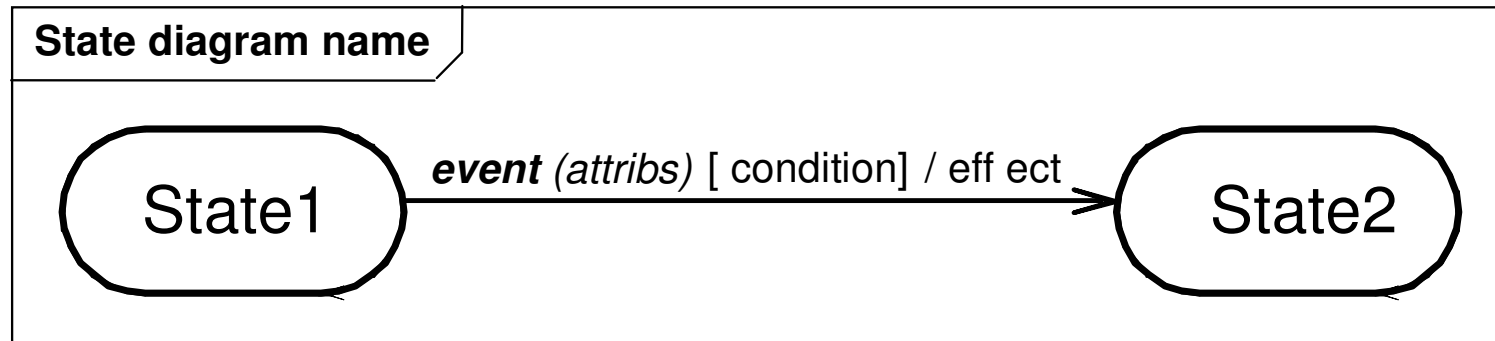
- ✓ State diagrams can represent continuous behavior with or without life cycle of objects.
- ✓ One short diagrams represent object finite lifecycle. It has initial (creation of object) and final states (destruction of object).

# Activity Effects

- ✓ An activity is the behavior that is invoked by any number of effects.
- ✓ Activities can be represented by operations, such as setting attribute values or generating other events.
- ✓ An object can perform the activity of sending signal to another object. A signal can be directed at a set of objects or a single object.
- ✓ If the target is a set of objects, each of them independently process the signal and determines whether to fire a transition.



# Basic Notation for State Diagram

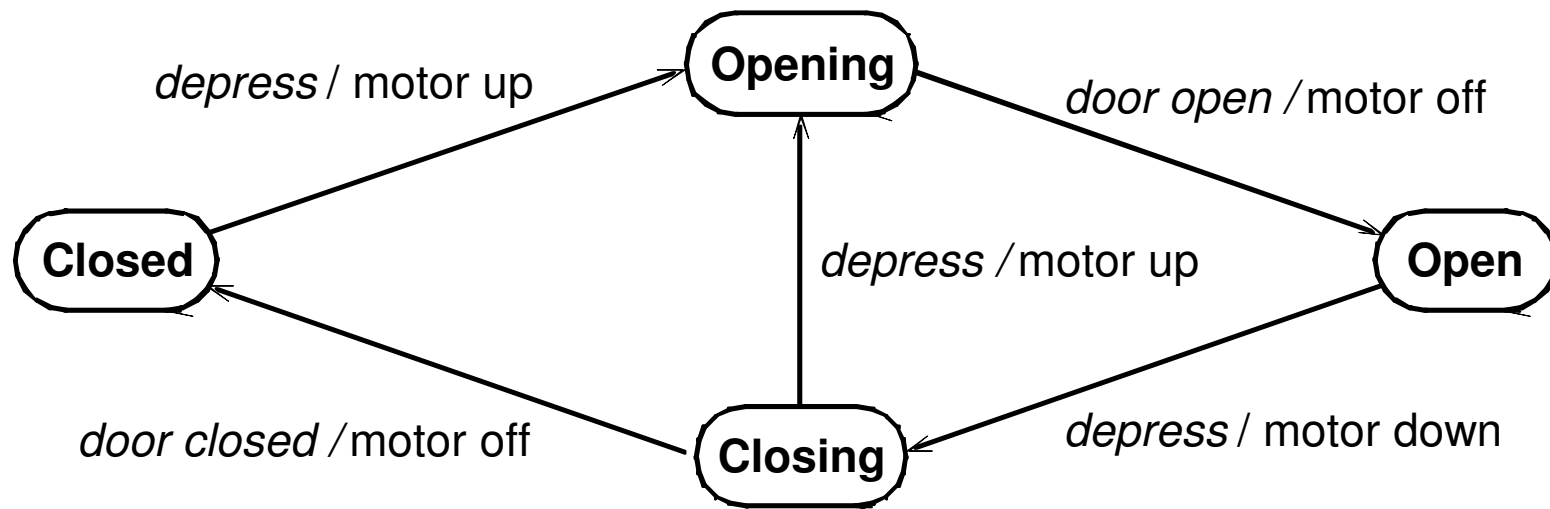
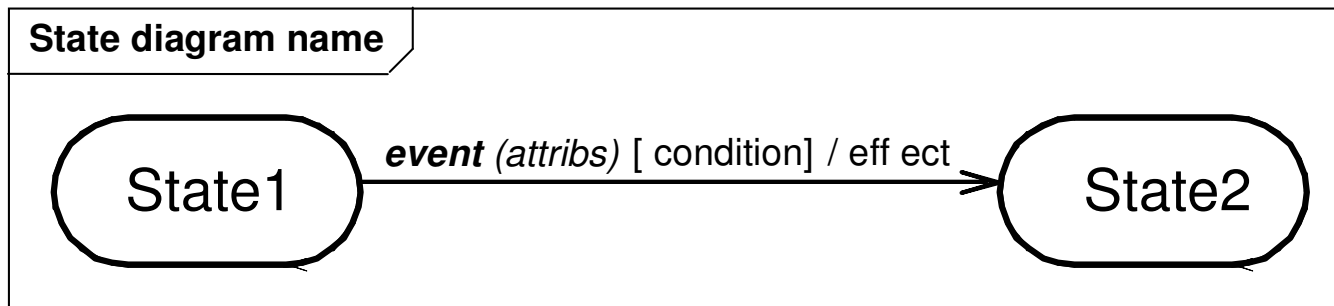


- ✓ State diagram is enclosed in a rectangular frame with the diagram name in a pentagonal tag in the left corner.
- ✓ Effect is a reference to an activity that is executed in response to an event. It can be attached to a transition or state after a slash.

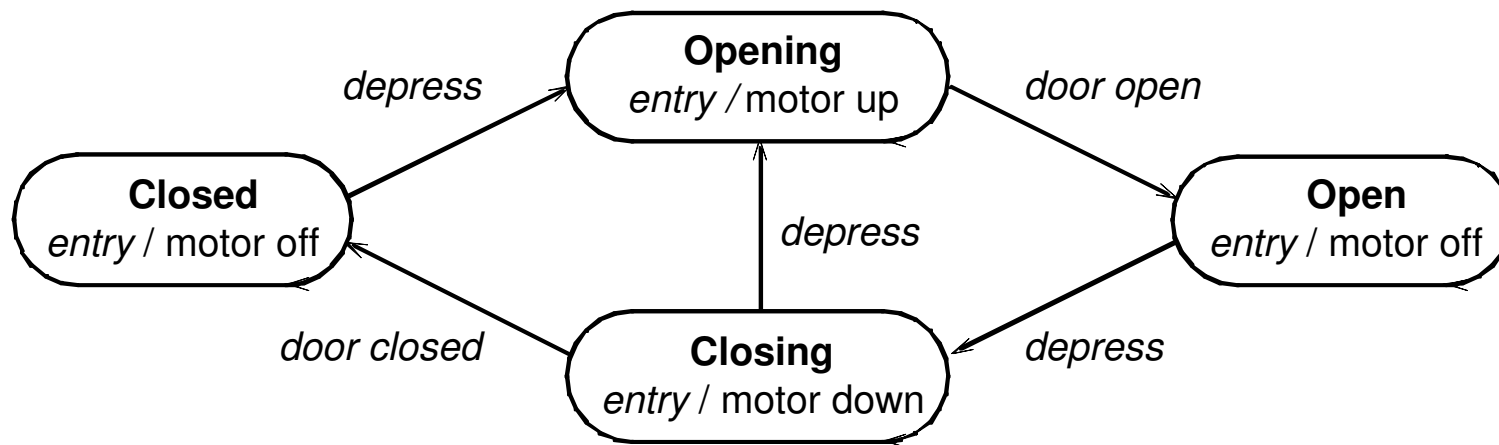
# Varieties of State Models

- ✓ **Mealy Model** (operations are associated with the transition).
- ✓ **Moore Model** (operations are associated with the state).
- ✓ **Combination Model** (operations are associated with the transition and states).
- ✓ There is no difference in expressive power between the two notations.

# Mealy Model (operation is associated with the transition)



# Moore Model (operation is associated with the state)



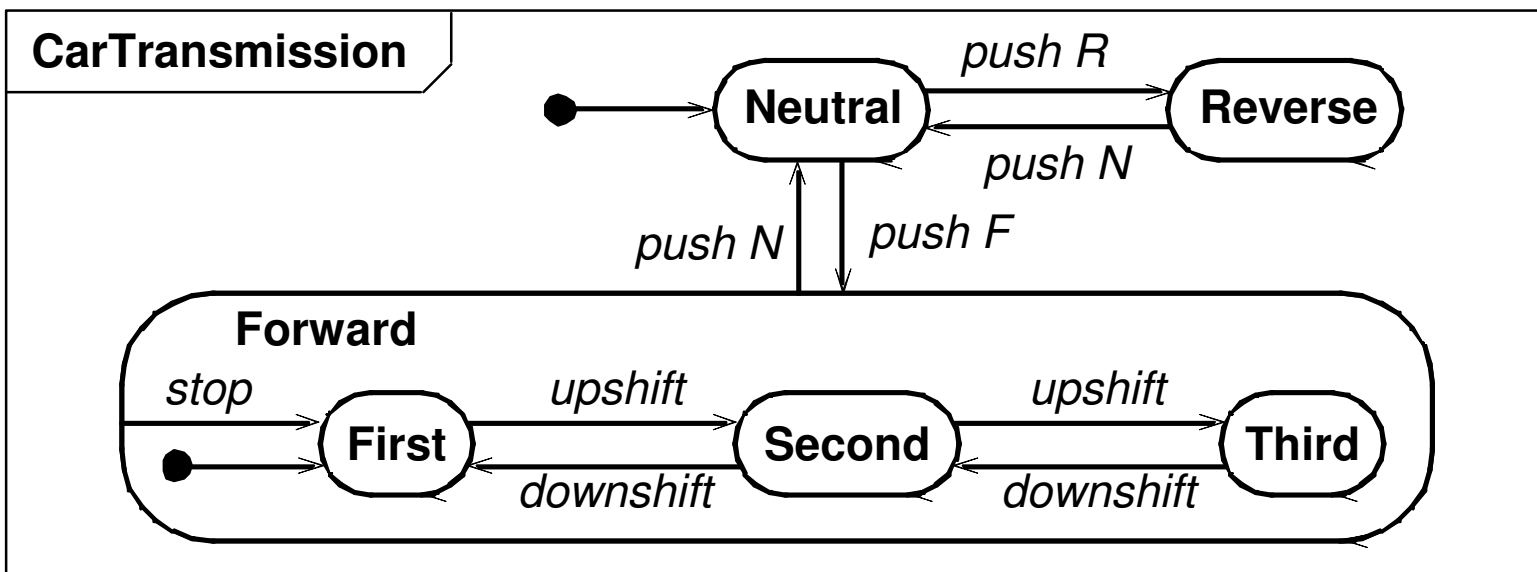
- ✓ Operations are bound to **entry** or **exit** from state.
- ✓ Only self-transition causes the entry and exit activities to be executed.

# Completion Transition

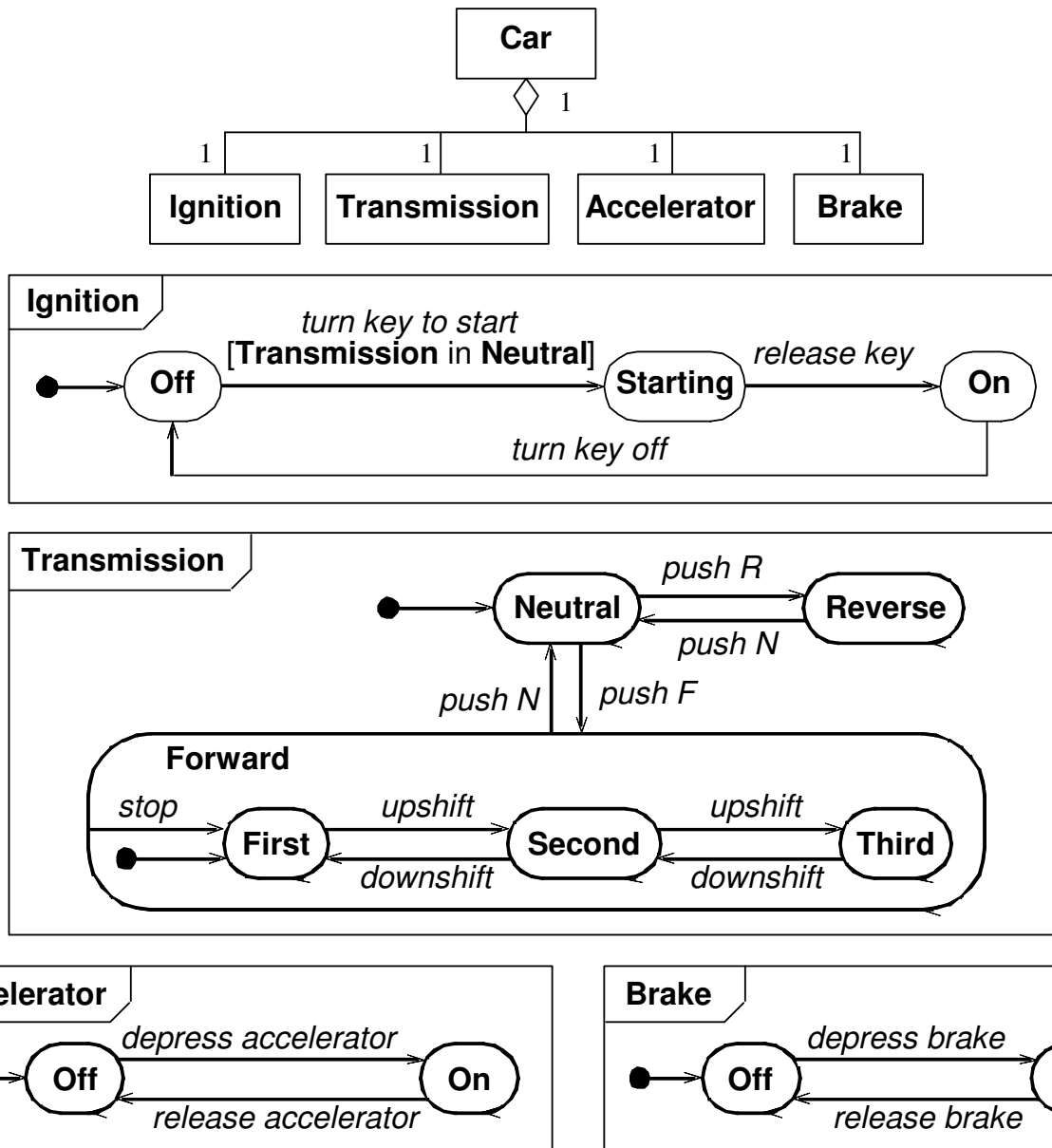
- ✓ When activity is completed, a transition to another state fires.
- ✓ An arrow without an event name indicates automatic transition that fires (if there is an operation associated with the source state, it must be completed).
- ✓ A guard condition is tested only once, when event occurs. If state has a completion transition, but none of the guard conditions are satisfied, when it remains active (may become 'stuck').

# Nested State Diagrams

- ✓ Any state can be replaced by a submachine.
- ✓ The composite state encloses the nested states.



# Aggregation Concurrency



The aggregate state corresponds to the combined states of all the parts.

# Class and State Models

- ✓ An object can have different states, but it cannot have different classes.
- ✓ Temporal differences of objects are defined as different states of the same class.
- ✓ The state model of a class is inherited by its subclasses.
- ✓ The state diagram of the subclass must be a refinement of the superclass state diagram.
- ✓ Any state from the parent class can be elaborated by using nested substates.