

SWE 760

# Lecture 4: State Machines for Real-Time Embedded Systems

Hassan Gomaa  
Department of Computer Science  
George Mason University  
Email: hgomaa@gmu.edu

References:

H. Gomaa, Chapter 7 - *Real-Time Software Design for Embedded Systems*,  
Cambridge University Press, 2016

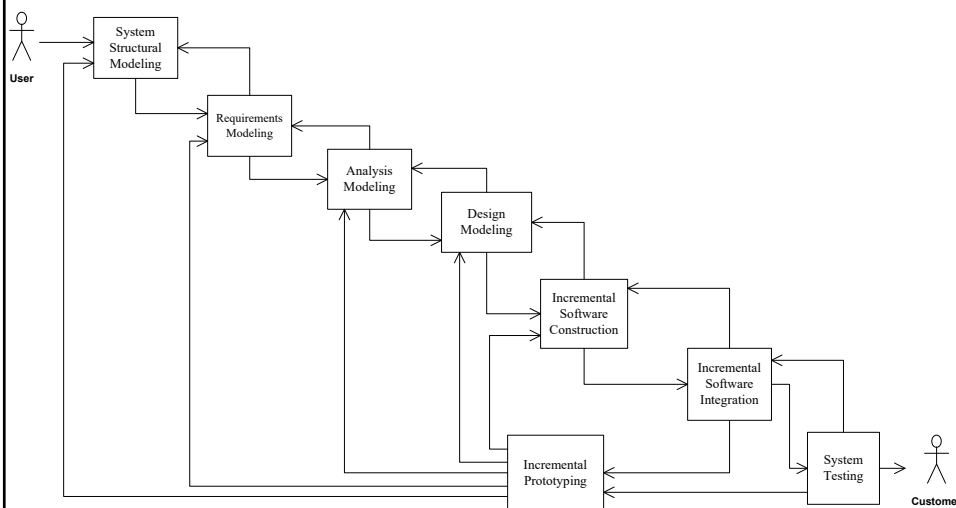
Copyright © 2016 Hassan Gomaa

All rights reserved. No part of this document may be reproduced in any form  
or by any means, without the prior written permission of the author.

Copyright 2016 H. Gomaa

1

Figure 4.1 COMET/RTE life cycle model



Copyright © 2016 Hassan Gomaa

2

## Analysis Modeling

- Static Modeling
  - Define entity classes and relationships
- **Dynamic State Machine Modeling**
  - **Real-time systems are highly state dependent**
  - **Actions depend on input event AND current state**
- Object Structuring
  - Determine objects that realize each use case
- Dynamic Interaction Modeling
  - Determine sequence of interactions among objects

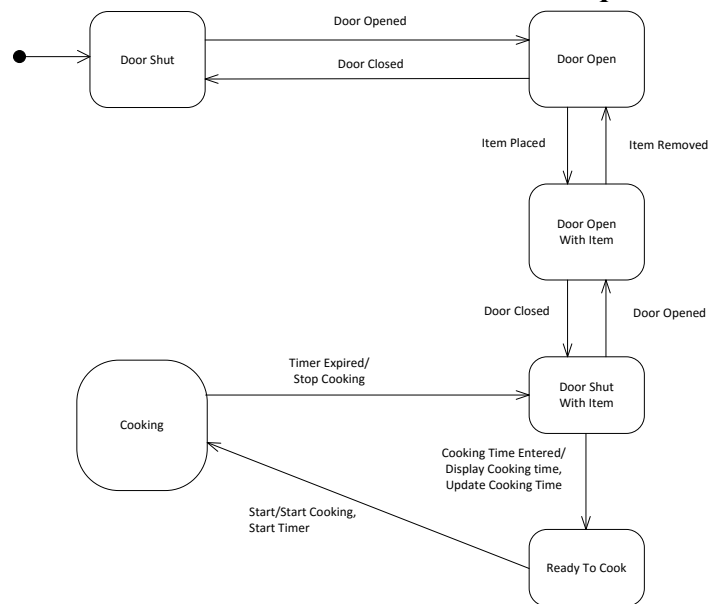
## State Machines

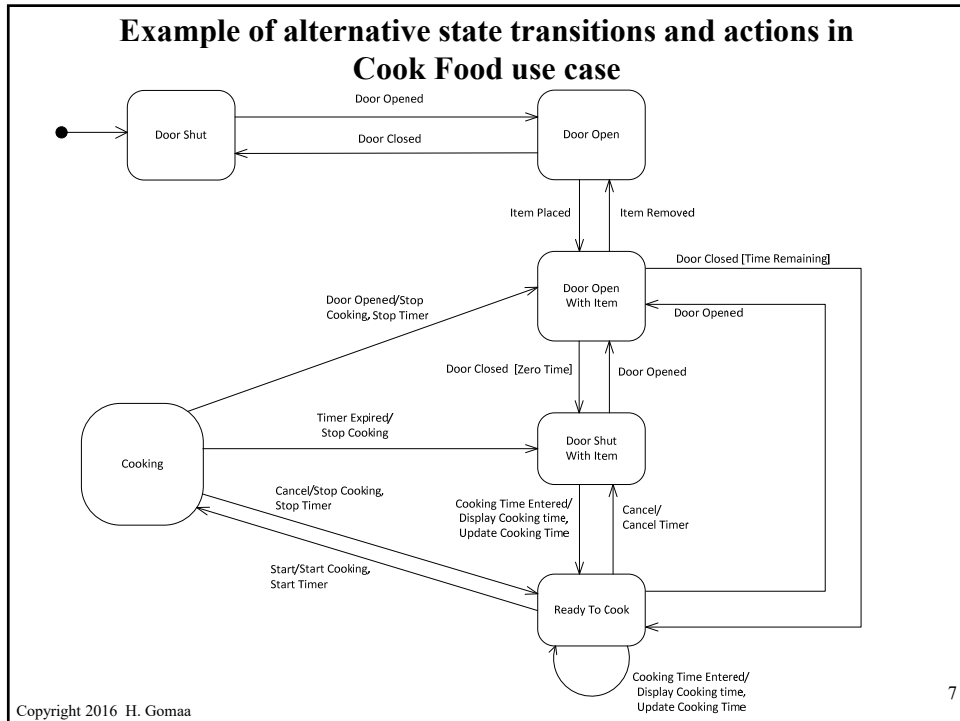
- Conceptual machine with finite number of states
  - a.k.a. finite state machine
- Graphical representation of State Machine
  - States are rounded boxes
  - Transitions are arcs
- State Machine
  - Relates events and states
- Event
  - Causes change of state
    - Referred to as state transition
- State
  - A recognizable situation
  - Exists over an interval of time
  - Represents an interval between successive events

## Events and Actions

- State transition label
  - Event [condition] / action(s)
- Event
  - When event occurs, condition must be *true* for state transition to occur.
  - If condition is *false*, state transition does not occur
- Action
  - Executed as a result of state transition
  - Executes instantaneously at state transition
  - Terminates itself

### Example of events and actions Microwave Control - Cook Food main sequence





## Entry and Exit Actions

- Entry action
  - Action executed on entry into state
    - Entry / action
    - E.g., Start Cooking
- Exit action
  - Action executed on exit from state
    - Exit / action
    - E.g, Stop Cooking

Copyright 2012 H. Goma

Copyright 2016 H. Goma

fsm-8

### Example of entry action

Fig. 10.11a: Actions on state transitions

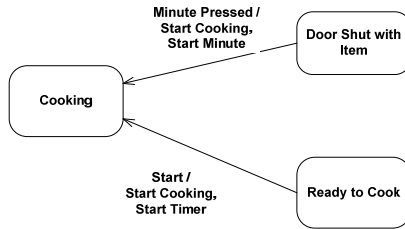
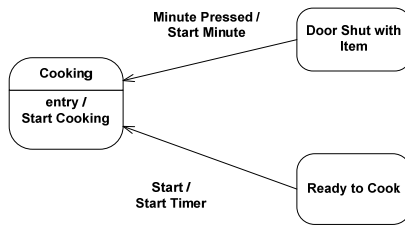
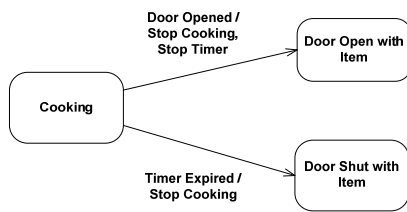


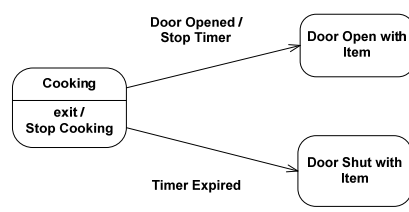
Fig. 10.11b: Entry action



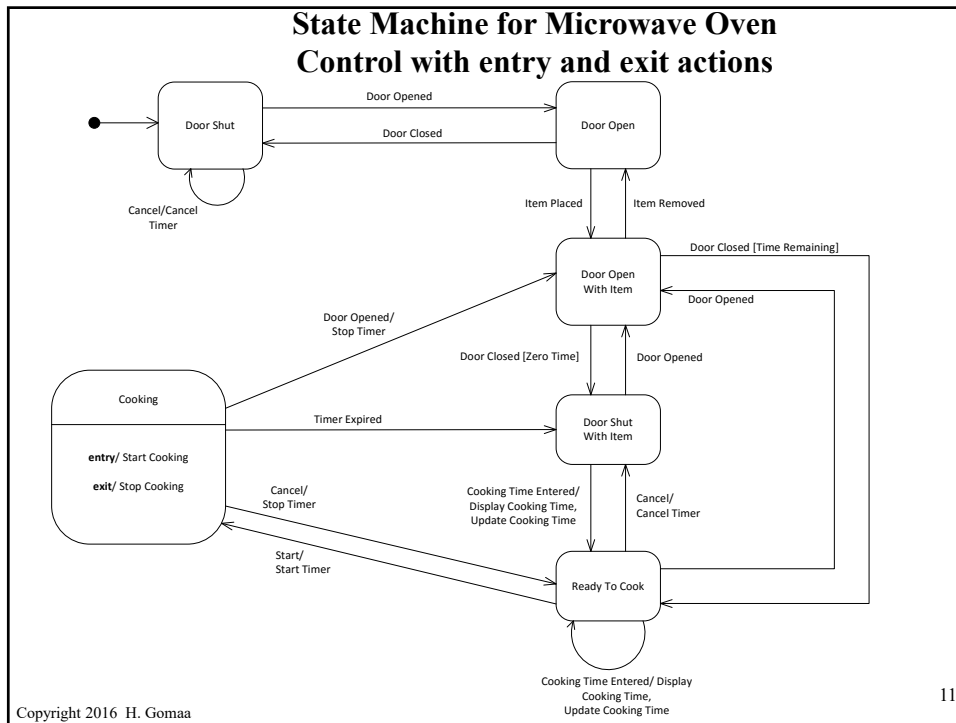
### Example of exit action



Actions on state transitions



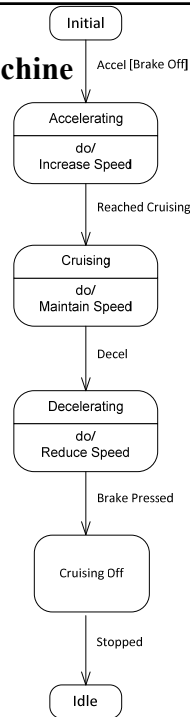
Exit action



## Activities

- Activity
  - Executes for duration of state
    - do / Activity in state
- Examples of activities in Cruise Control State Machine
  - Increase Speed
    - Executes for duration of Accelerating state
  - Maintain Speed
    - Executes for duration of Cruising state
  - Reduce Speed
    - Executes for duration of Decelerating state
  - Stop Train
    - Executes for duration of Stopping state

## Cruise Control State Machine with activities



Copyright 2016 H. Goma

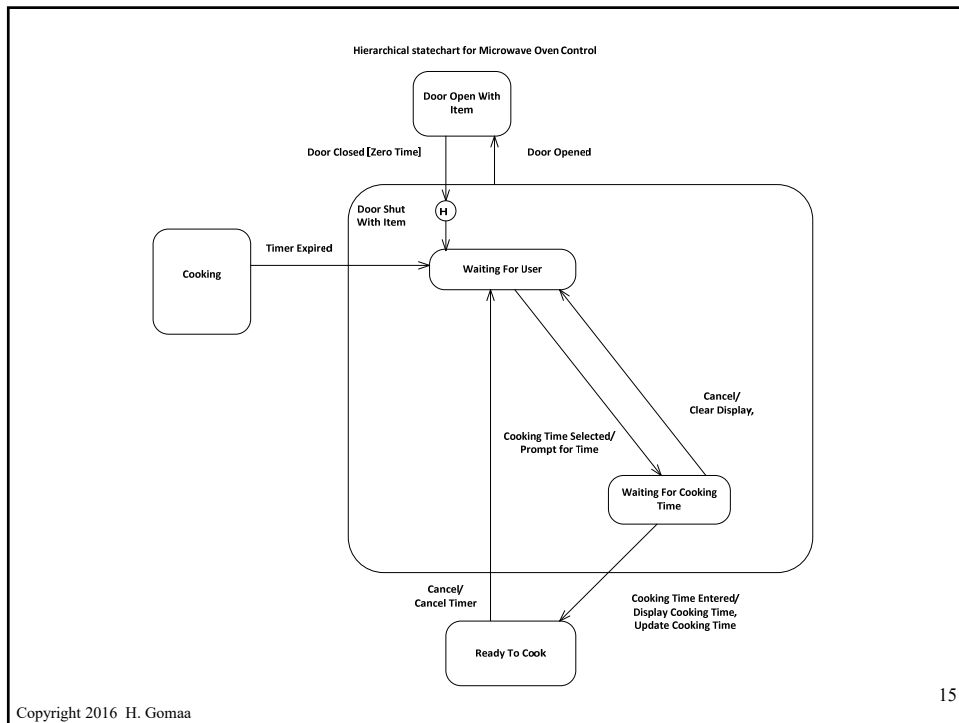
13

## Hierarchical State Machines

- Sequential State (OR) decomposition
  - When object is in composite state (superstate)
    - It is in one and only one of substates
  - Transition into composite state
    - Must be to one and only one of substates
- Aggregation of state transitions
  - If same event causes transition out of every substate
  - Then aggregate into transition out of composite state
- History state
  - When composite state is re-entered
  - Enter substate that was previously active
    - Prior to previous exit from composite state
- Example: Microwave Oven Control

Copyright 2016 H. Goma

14



## Orthogonal State Machines

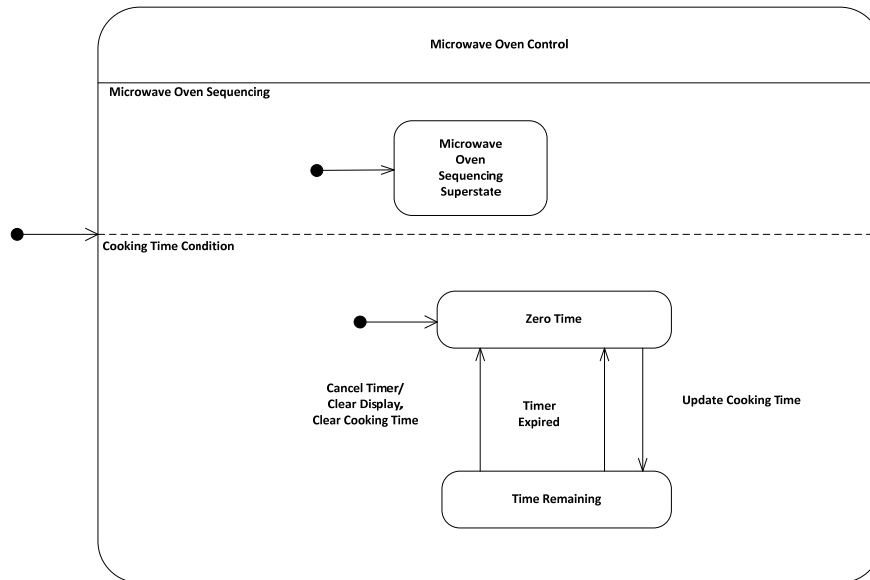
- Concurrent State Machines
  - State of an object described by more than one state machine
- Orthogonal State Machine
  - Used to depict states of different aspects of object
  - Uses concurrent state machine
- Concurrent state (AND) decomposition
  - Object is in one substate on each lower level state machine
  - Object's state is union of all substates
- Same event
  - May cause transitions on more than one state machine
- Output event on one state machine
  - May be input event on other state machine
- Substate on one state machine
  - May be condition on other state machine
- Example: Microwave Oven Control

16

Copyright 2016 H. Goma



## Orthogonal State Machines for Microwave Oven Control



Copyright 2016 H. Goma

17

## Variability in State Machines - Inherited State Machines

- When a state machine is specialized
- Child state machine inherits properties of parent state machine
- Child state machine can:
  - Add new states
    - New states at same level of hierarchy as inherited states
    - New substates of new or inherited states
    - New orthogonal (concurrent) states
  - Add new events and transitions
  - Add or remove actions and activities
- Child state machine
  - Must not delete states or events defined in parent

Copyright 2016 H. Goma

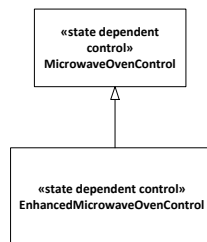
18

## Inherited State Machines

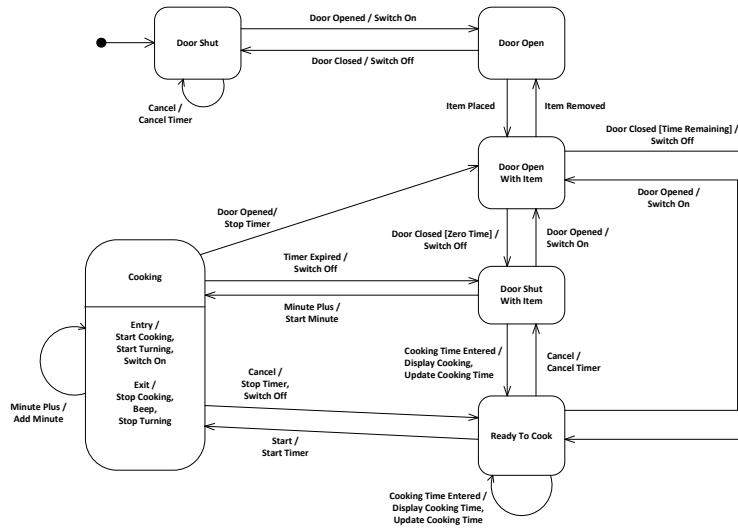
### Examples from Microwave Oven SPL

- Superclass: Microwave Oven Control
- Subclass with all features: Enhanced Microwave Oven Control
- Added functionality
  - Light, Turntable, Beeper, Minute Plus, Time of Day (TOD)
- Example of new states added
  - TOD
    - 3 substates of inherited Door Shut state
- Example of new transitions added
  - Minute Plus
- Example of new actions added
  - Turntable, Beeper

## Inheritance of State Dependent Control Class



## Inherited State Machine for Enhanced Microwave Oven Control



Copyright 2016 H. Goma

21

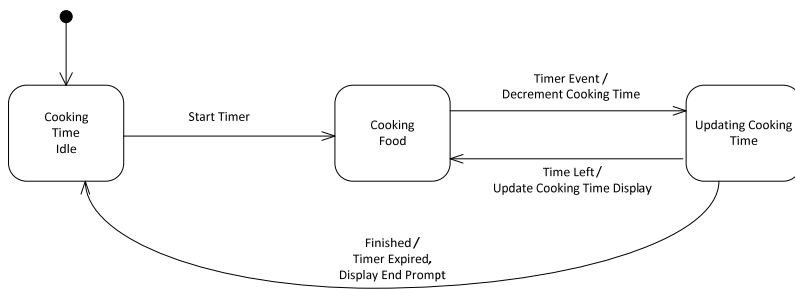
## Cooperating State Machines

- Cooperating state-dependent control objects
- Each control object executes a state machine
- Cooperating state machines
  - Action on one state machine corresponds to event on another state machine
  - E.g., Microwave Oven Control, Oven Timer state machines
- Microwave Oven Control state machine
  - Action: Start Timer -> Oven Timer state machine
  - Event: Timer Expired
- Oven Timer state machine
  - Event: Start Timer
  - Action: Timer Expired -> Microwave Oven Control state machine

Copyright 2016 H. Goma

22

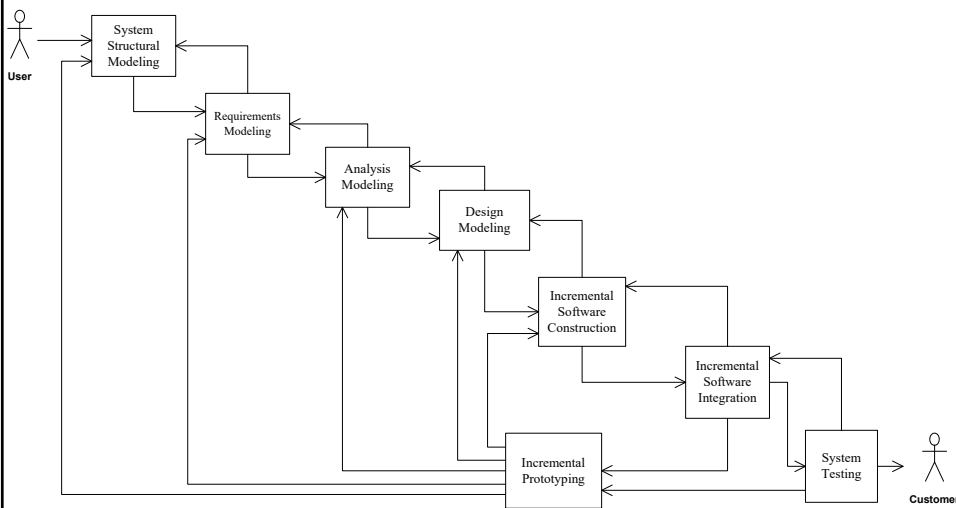
## Cooperating State Machines: Oven Timer State Machine – Cook Food use case



Copyright 2016 H. Goma

23

### Figure 4.1 COMET/RTE life cycle model



Copyright © 2016 Hassan Goma

24

## Analysis Modeling

- Static Modeling
  - Define entity classes and relationships
- **Dynamic State Machine Modeling**
  - **Real-time systems are highly state dependent**
- Object Structuring
  - Determine objects that realize each use case
- Dynamic Interaction Modeling
  - Determine sequence of interactions among objects