

Fowler, Chapter 10

State Diagrams



Announcements

- Homework 2 due tomorrow at Noon
- Homework 3 on implementation and testing will be handed out next week
- Homework 3 due date extended by one week to November 18th
- Tutorial 3 starts next week.

We Will Cover

- State Machines
 - An alternate way of capturing scenarios
 - Large classes of scenarios
- Syntax and Semantics
- When to use state machines

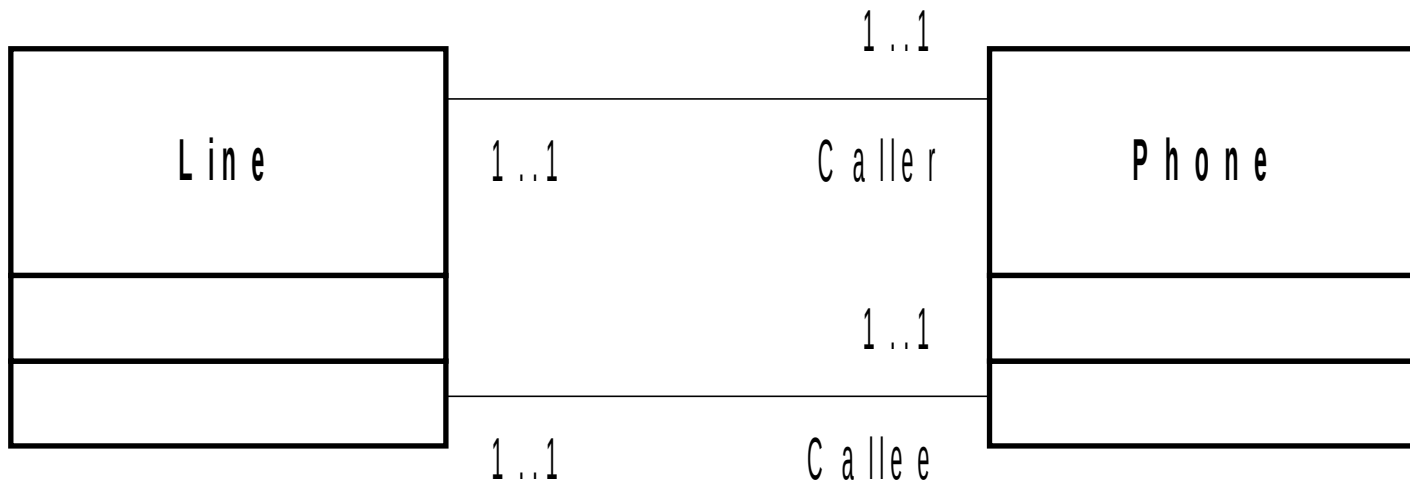
Events, Conditions, and States

- Event
 - Something that happens at a point in time
 - Operator presses self-test button
 - The alarm goes off
- Condition
 - Something that has a duration
 - The fuel level is high
 - The alarm is on
- State
 - An abstraction of the attributes and links of an object (or entire system)
 - The controller is in the state self-test after the self-test button has been pressed and the rest-button has not yet been pressed
 - The tank is in the state too-low when the fuel level has been below level-low for alarm-threshold seconds

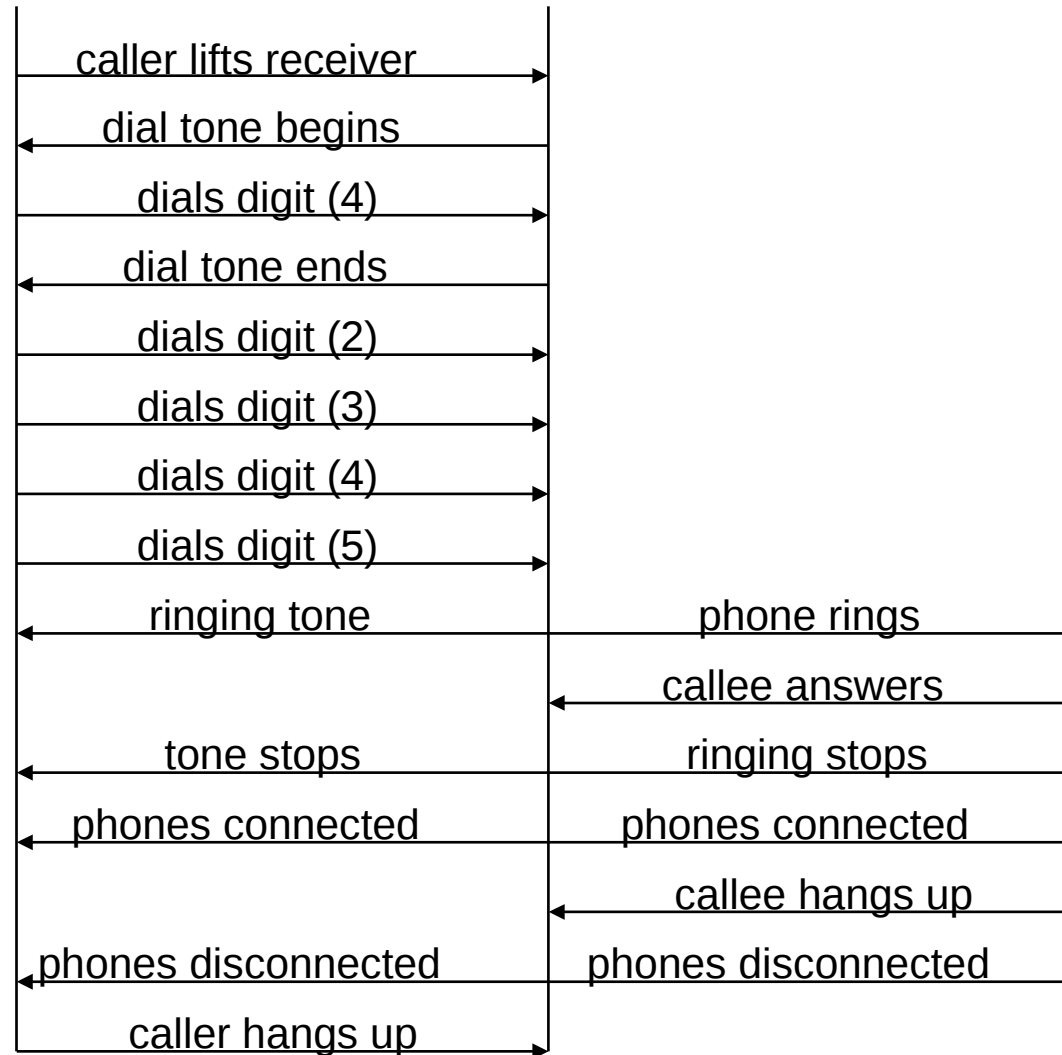
Making a Phone Call Scenario

- To make a call, the caller lifts receiver. The caller gets a dial tone and the caller dials digit (x). The dial tone ends. the caller completes dialing the number. The callee phone begins ringing at the same time a ringing begins in caller phone. When the callee answers the called phone stops ringing and ringing ends in caller phone. The phones are now connected. The caller hangs up and the phones are disconnected. The callee hangs up.

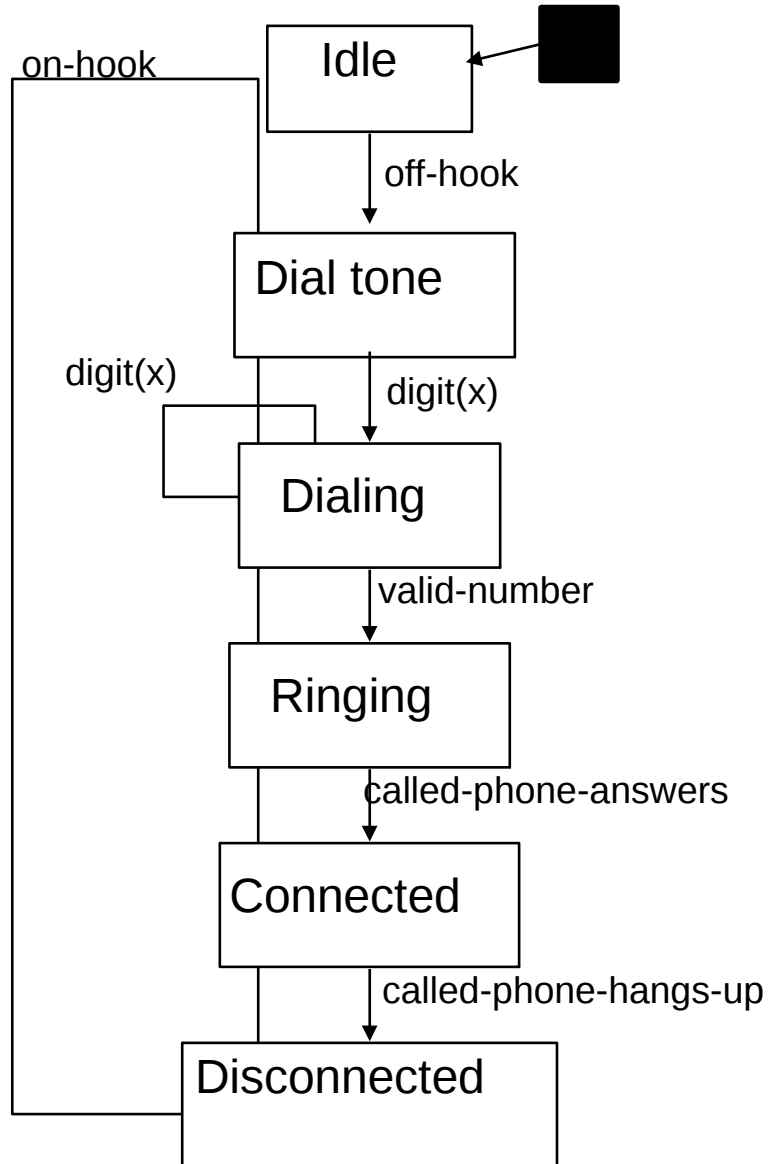
Partial Class Diagram



Event Trace



State Diagram for Scenario

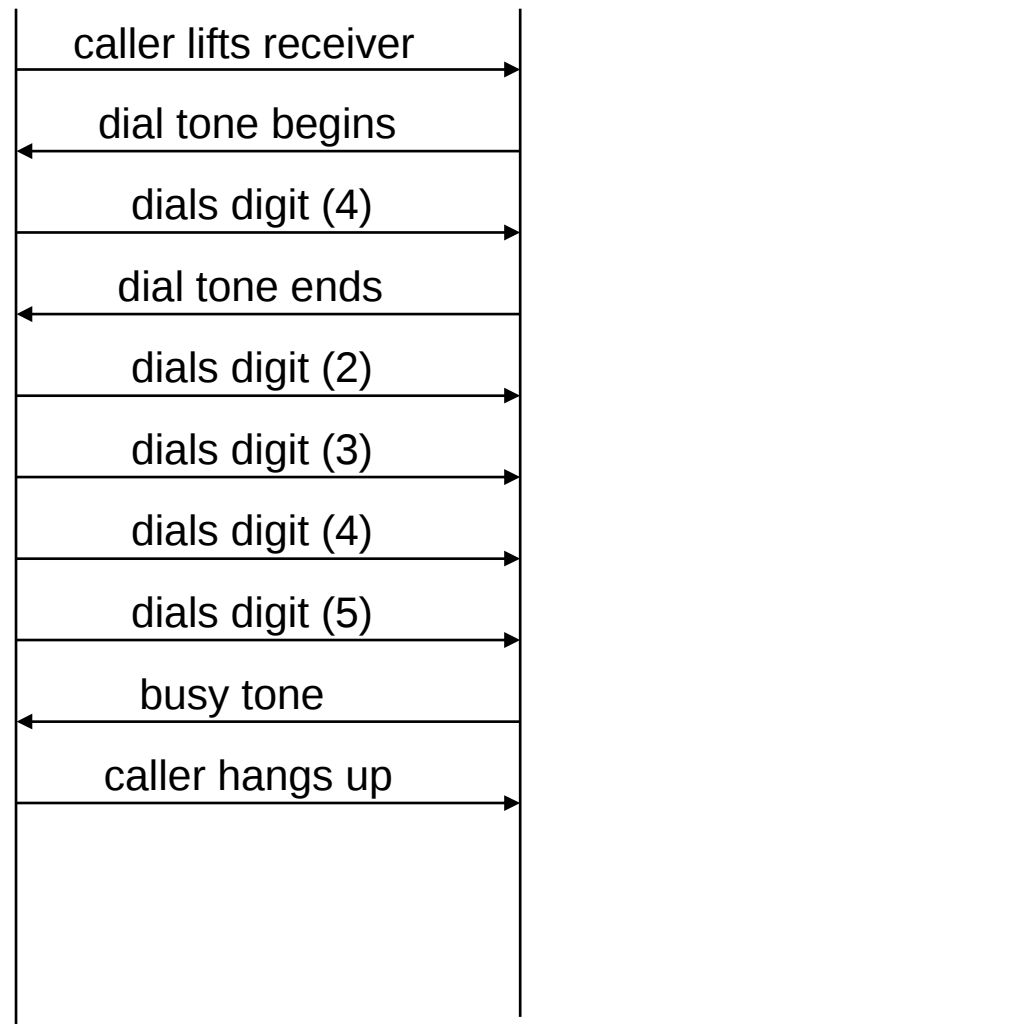


Scenario 2

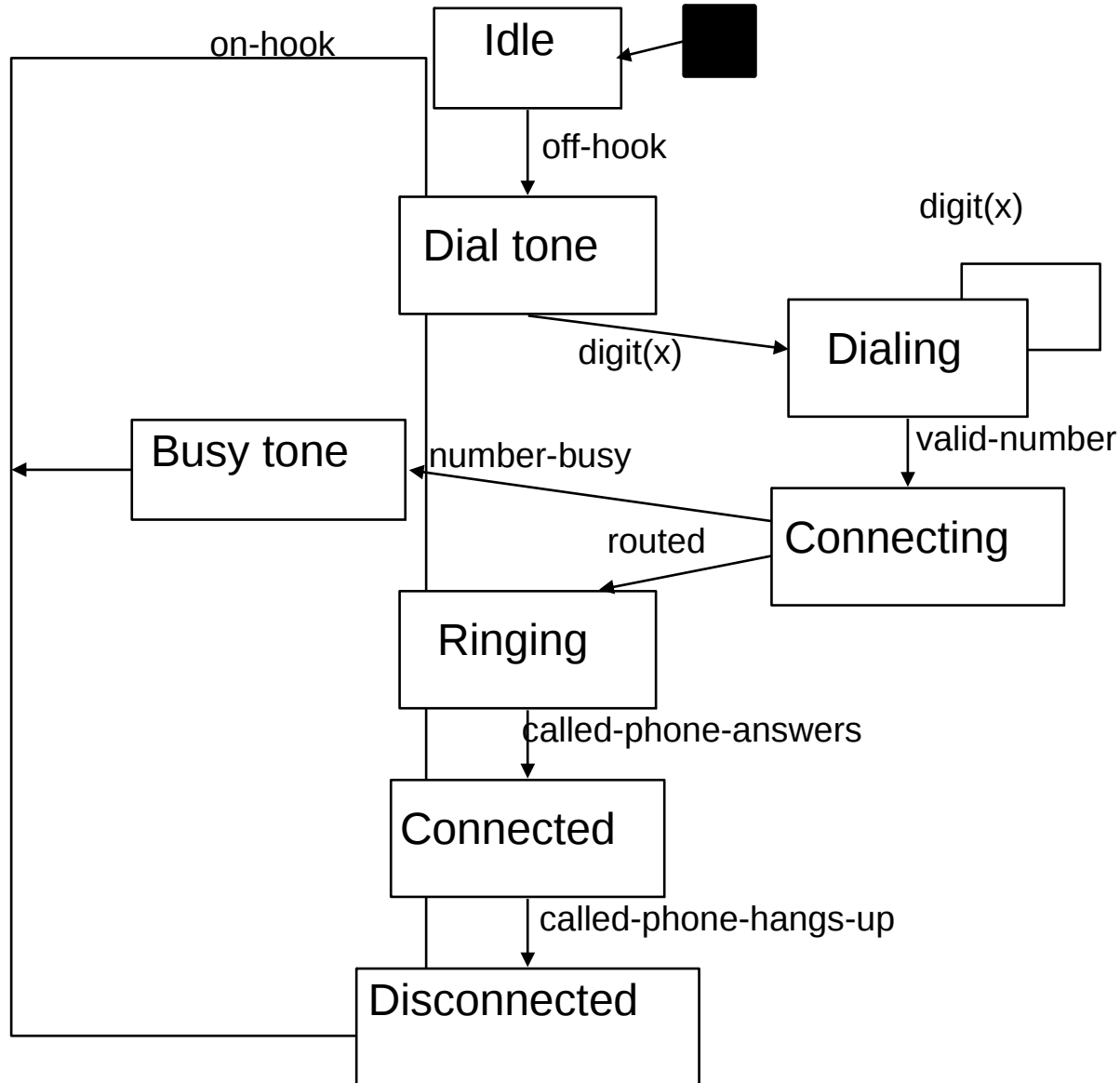
Caller

Line

Callee

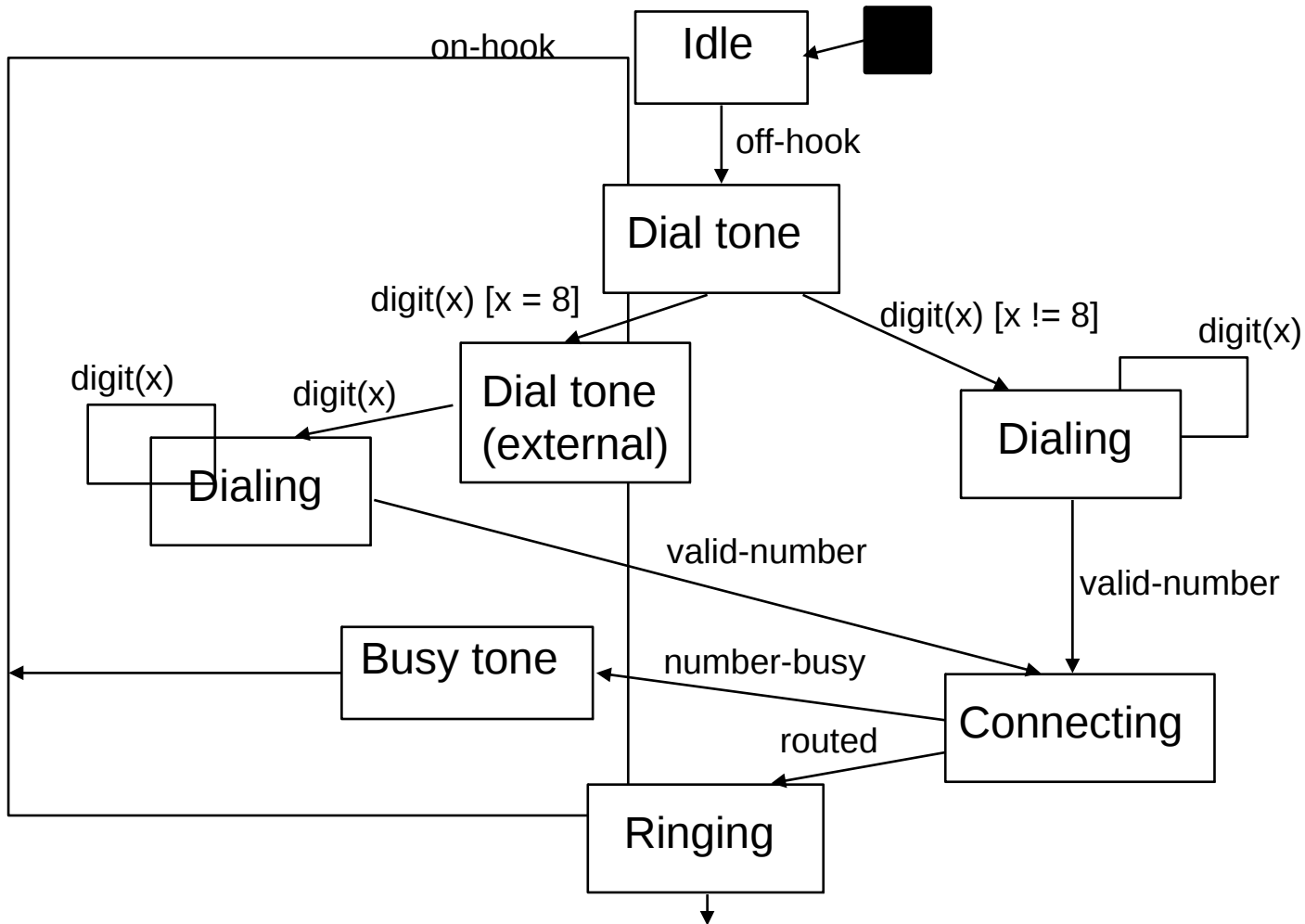


Modified State Machine



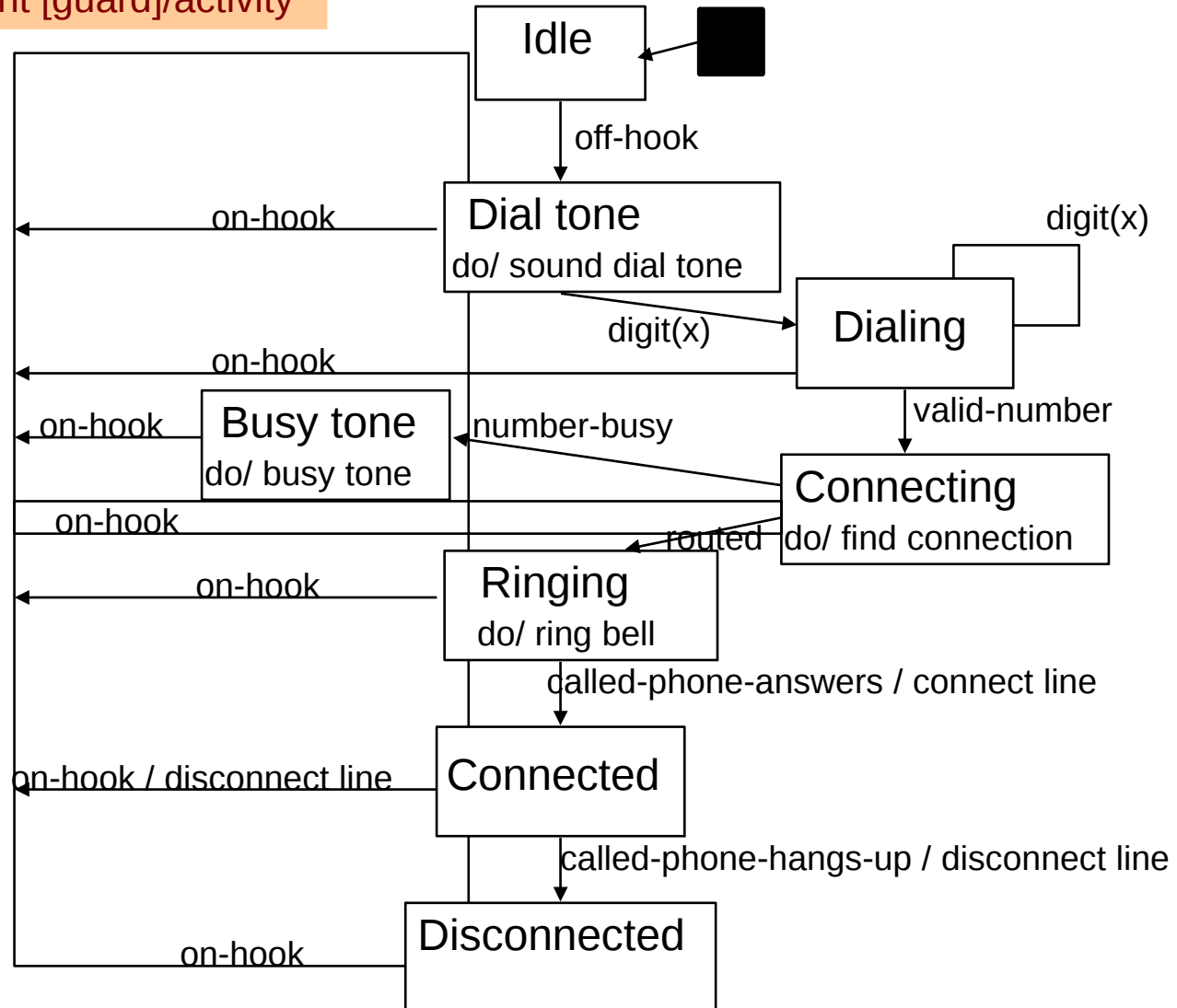
Conditions

- Sometimes the state transitions are conditional



Operations (AKA Actions)

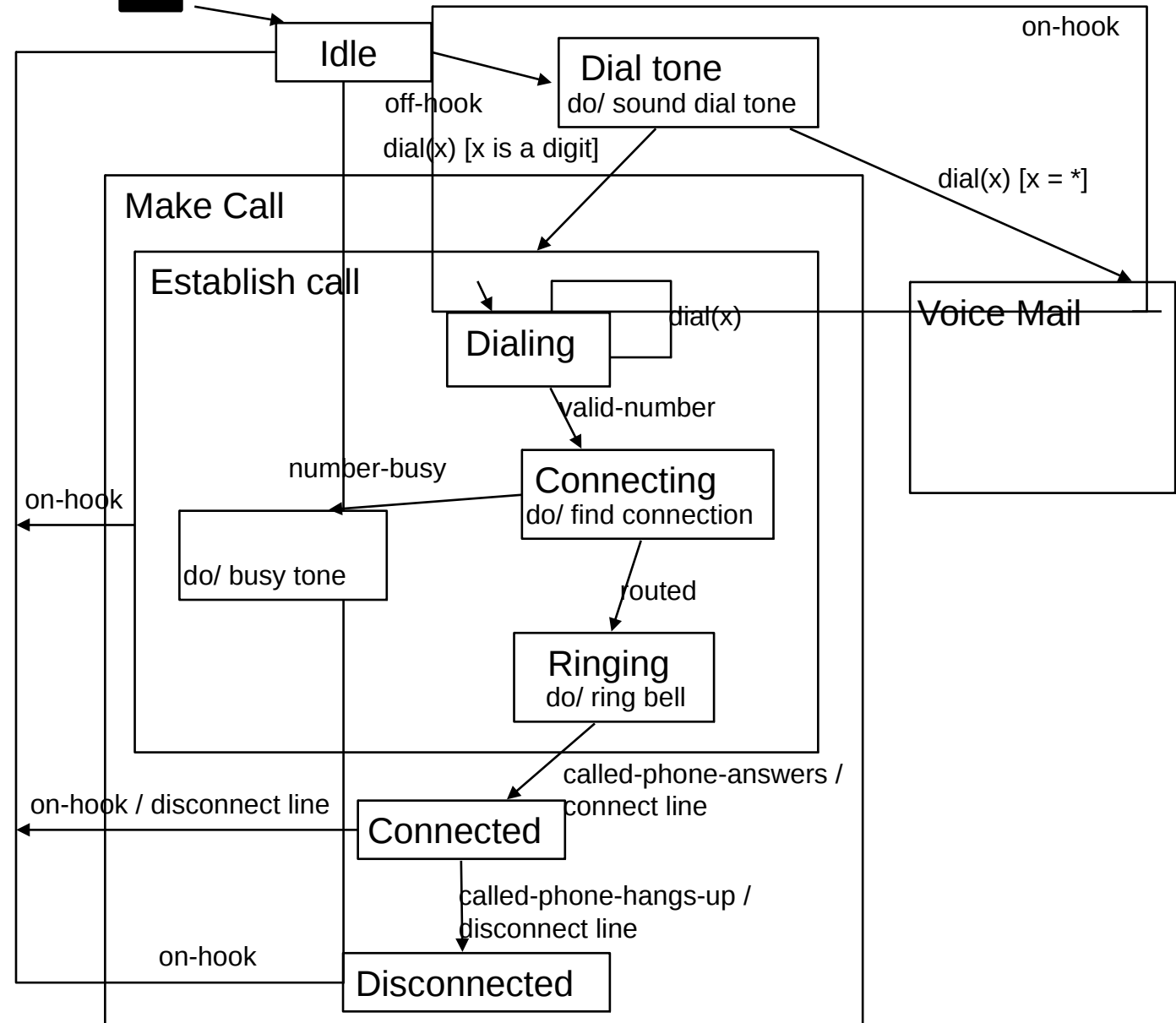
Transition label: trigger-event [guard]/activity



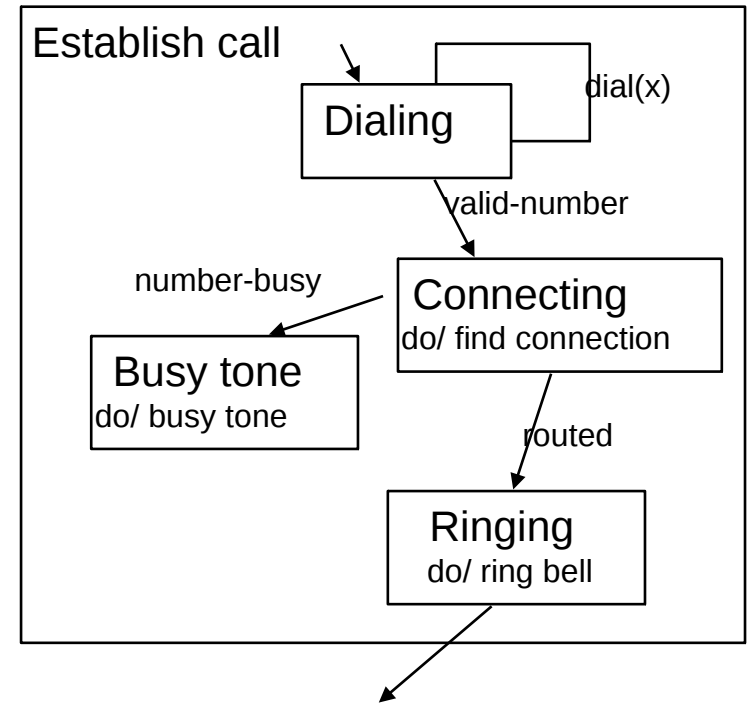
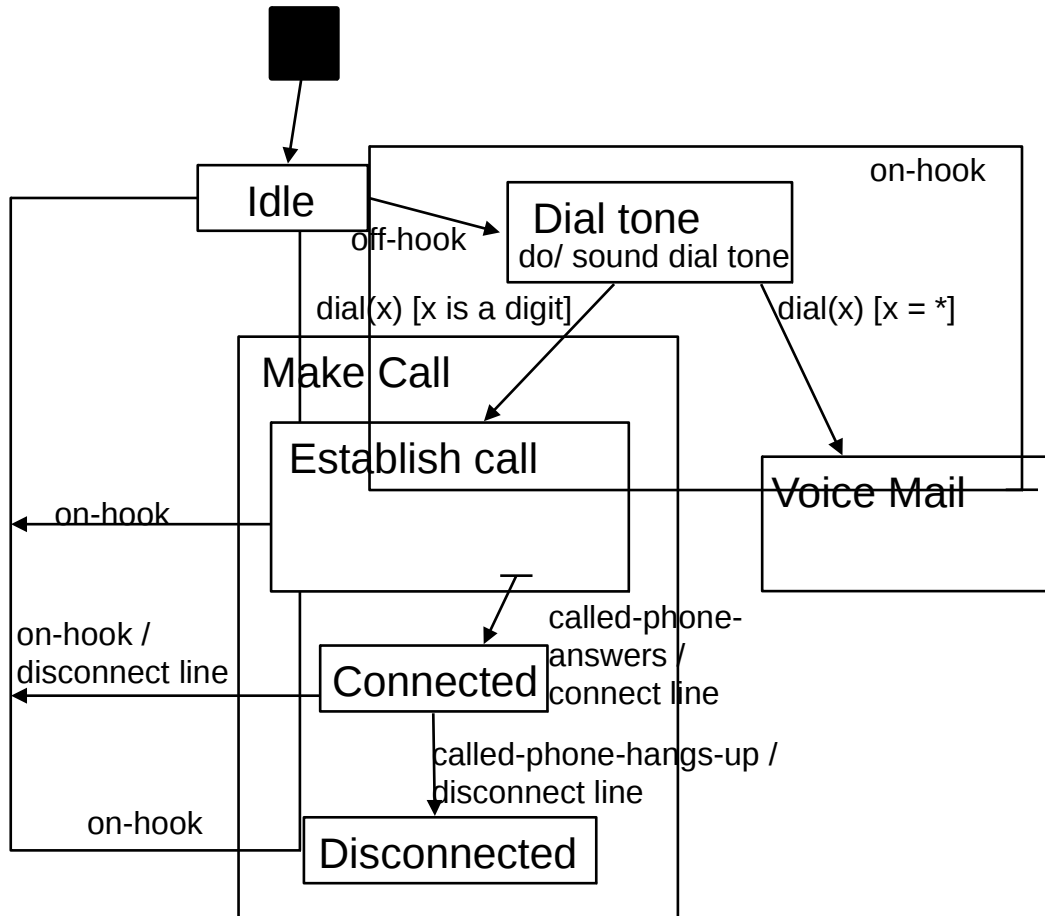
- Actions are performed when a transition is taken or performed while in a state
- Actions are terminated when leaving the state

Hierarchical State Machines

- Group states with similar characteristics
- Enables information hiding
- Simplifies the diagrams

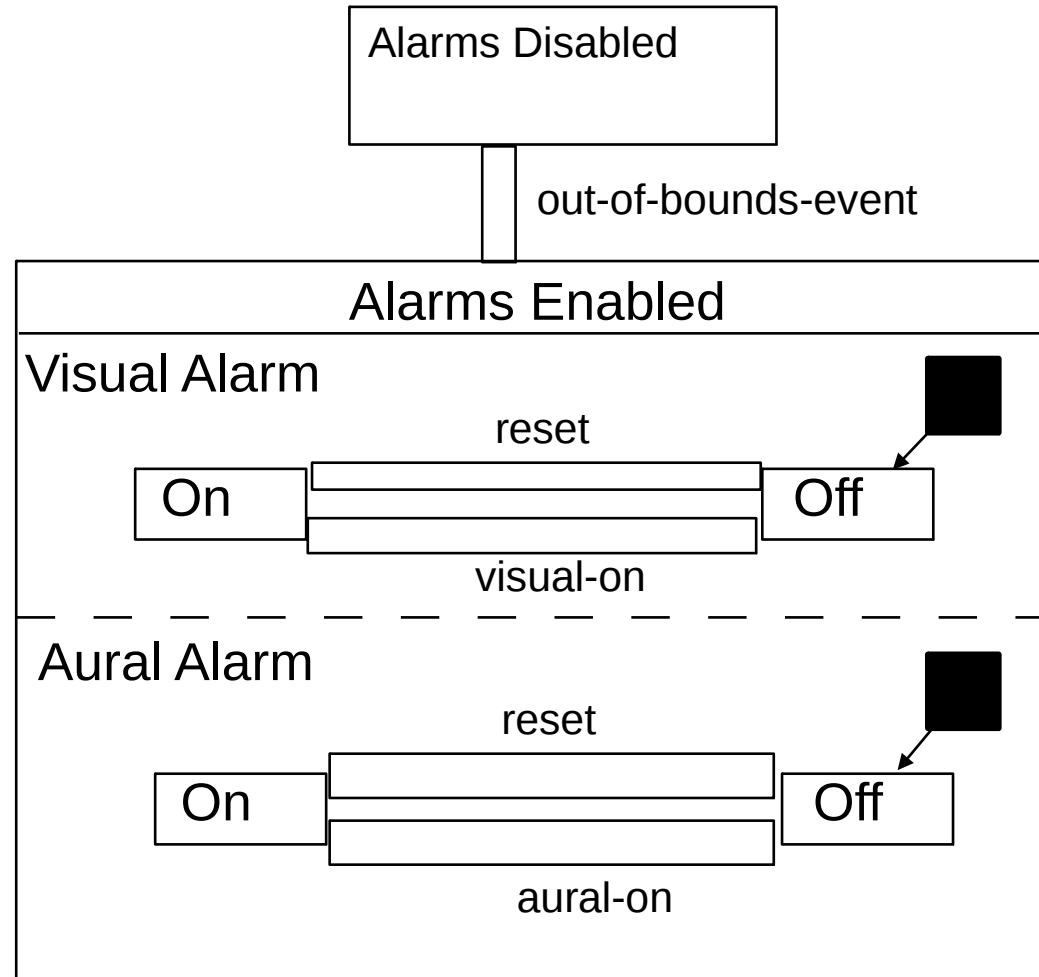


Information Hiding



Concurrency

- Some states represent several concurrent concepts
- Concurrency is supported by the state machines
- Concurrent state machines are separated by dashed lines



State Machines - Summary

■ Events

- instances in time

■ Conditions

- conditions over time

■ States

- abstraction of the attributes and associations

■ Transitions

- Takes the state machine from one state to the next
- Triggered by events
- Guarded by conditions
- Cause actions to happen

■ Internal actions

- something performed in a state

■ Hierarchies

- allows abstraction and information hiding

■ Parallelism

- models concurrent concepts

When to use State Machines

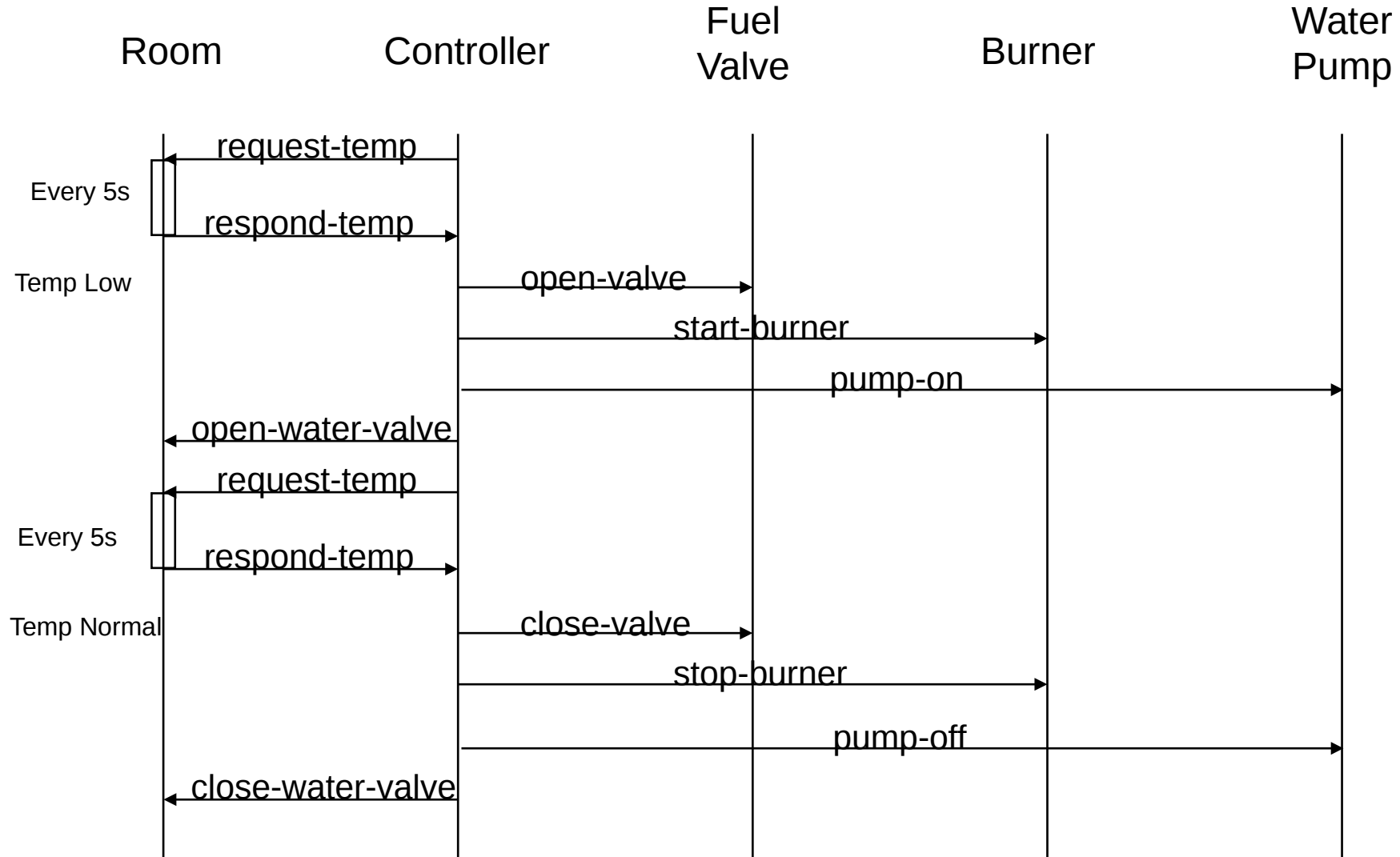
- When you want to describe the behavior of one object for all (or at least many) scenarios that effect that object
- Not good at showing the interaction between objects
 - Use interaction diagrams or activity diagrams
- Do not use them for all classes
 - Some methods prescribe this (I would)
 - Very time consuming and questionable benefit

Coming up with the State Diagrams

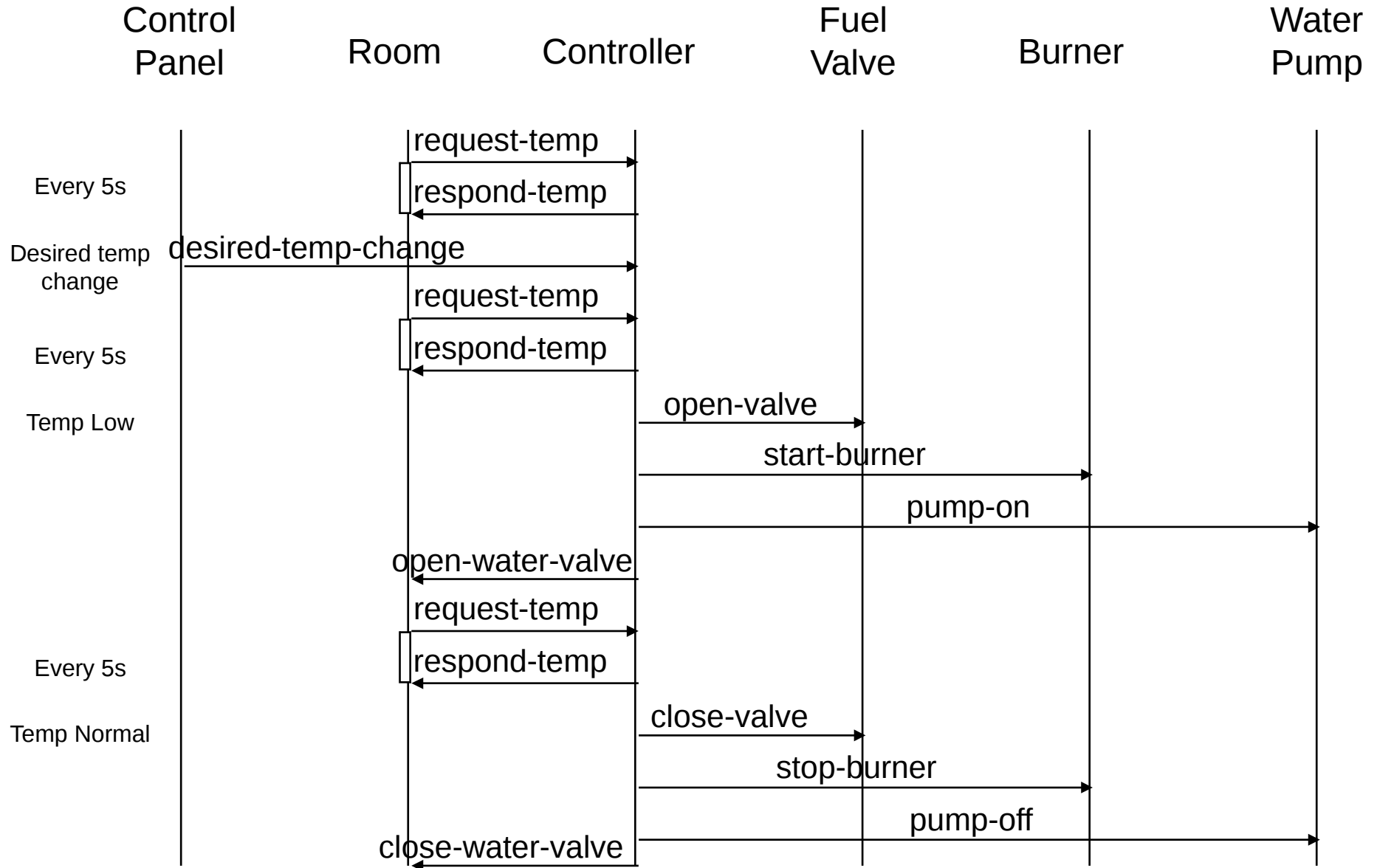
Modeling Approach

- Prepare scenarios
 - Work with the customer
 - Start with normal scenarios
 - Add abnormal scenarios
- Identify events (often messages)
 - Group into event classes
- Draw some sequence diagrams
 - Find objects with complex functionality you want to understand better
- Build a state diagram for the complex classes

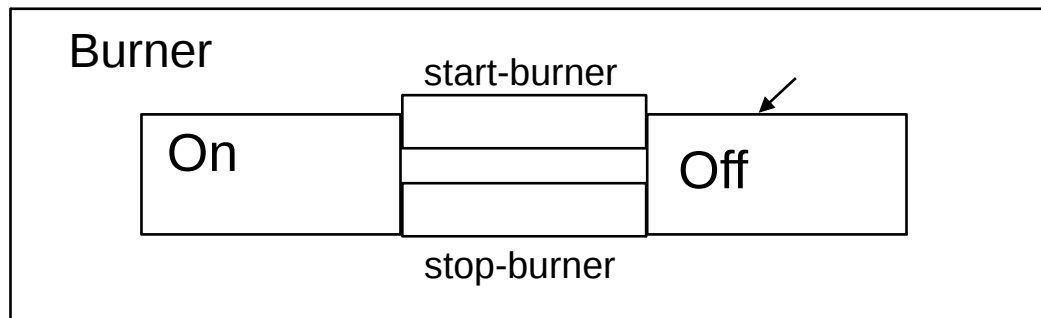
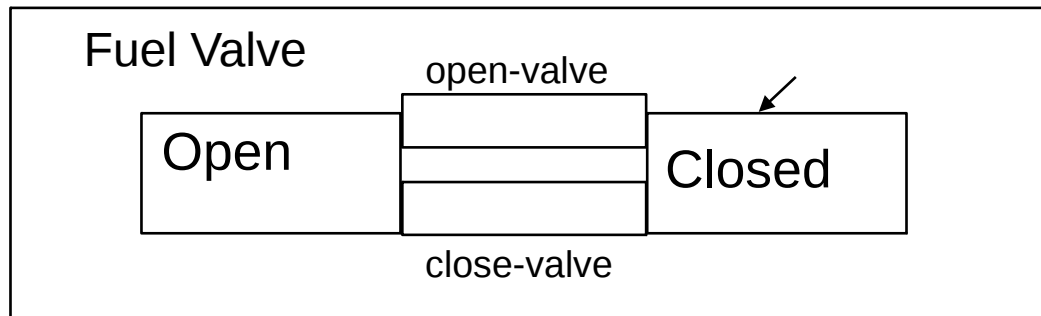
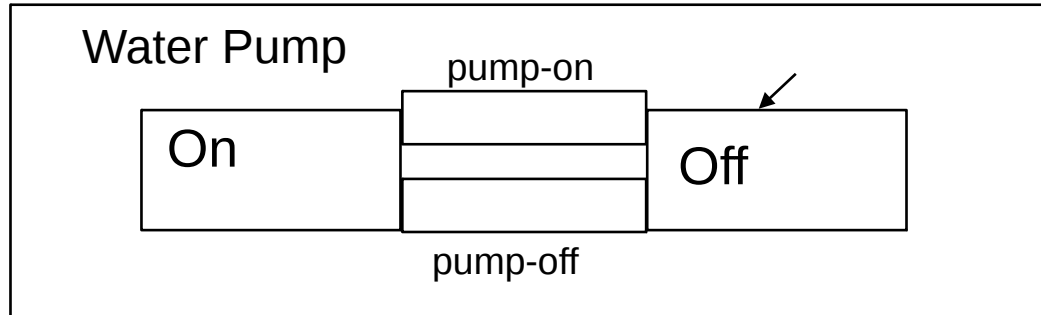
Scenario-1



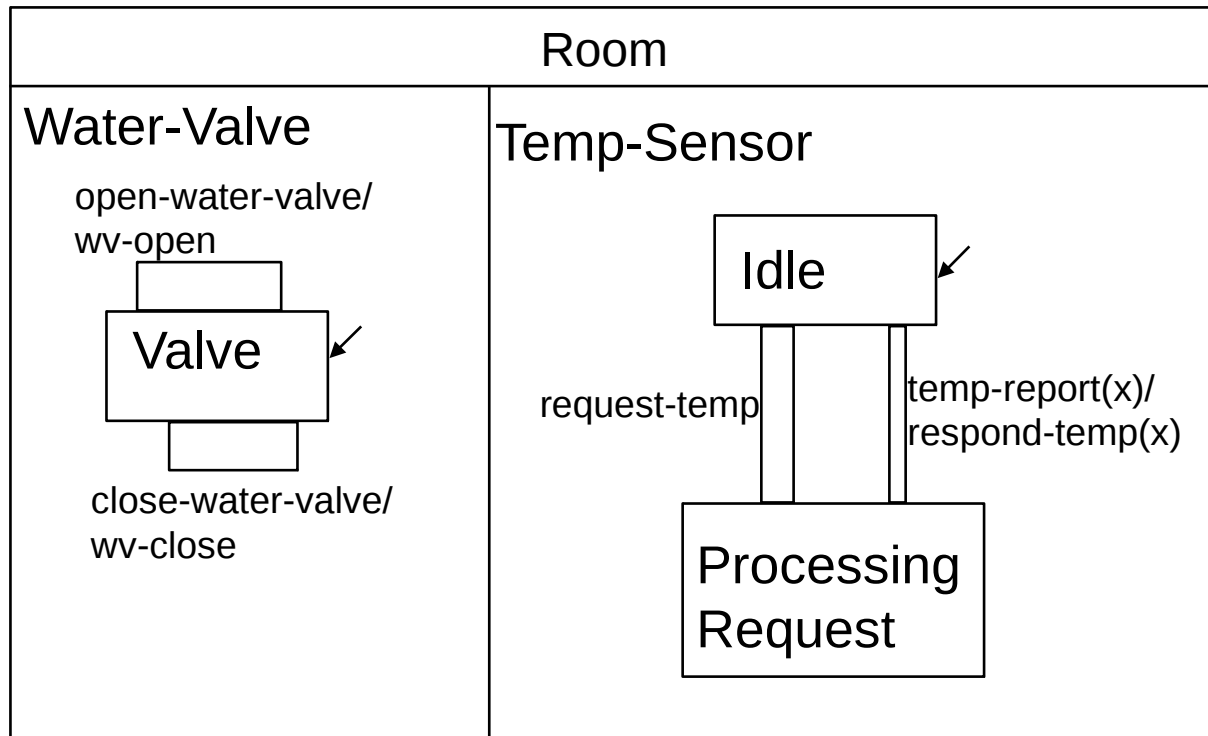
Scenario-2



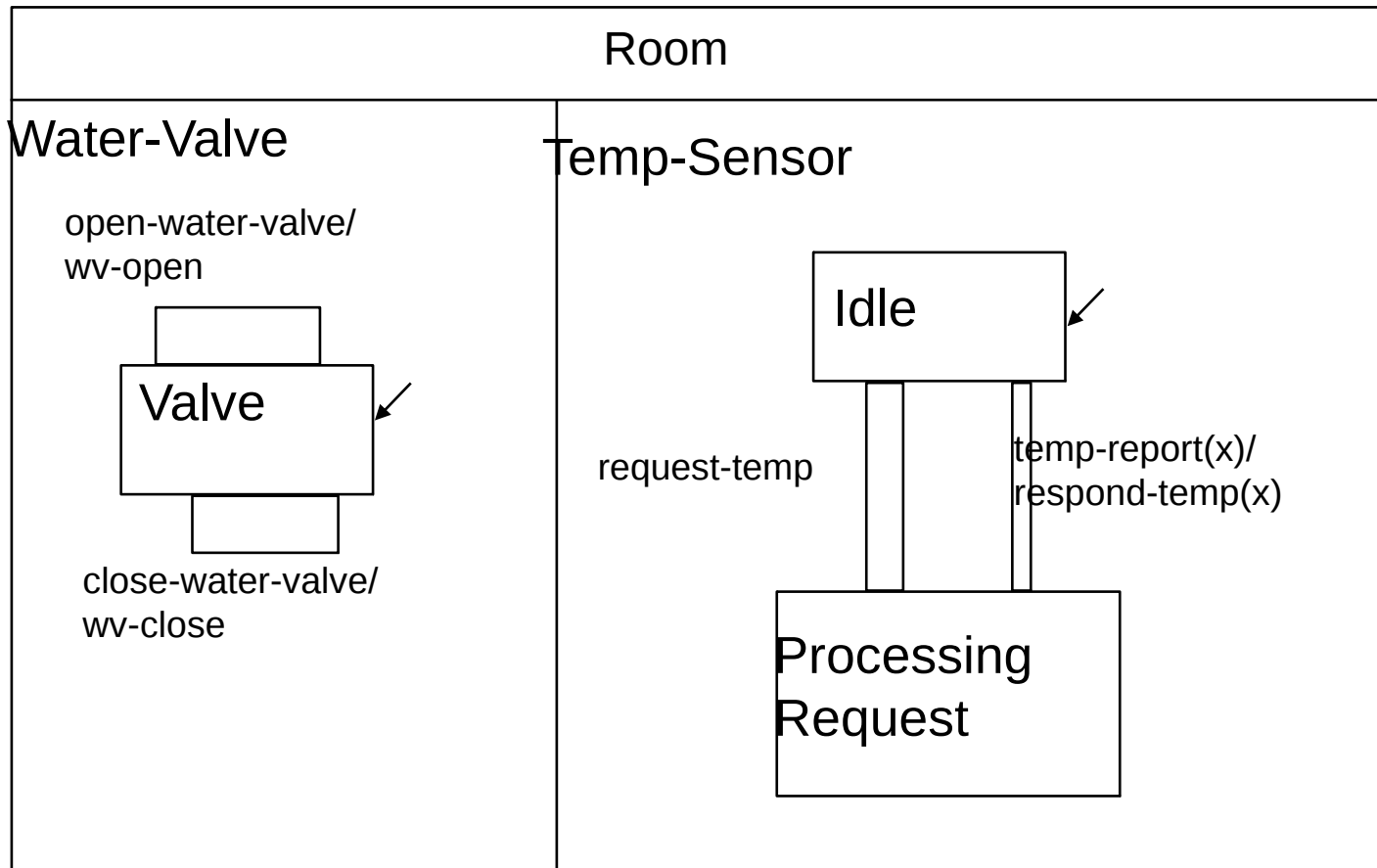
Dynamic Model



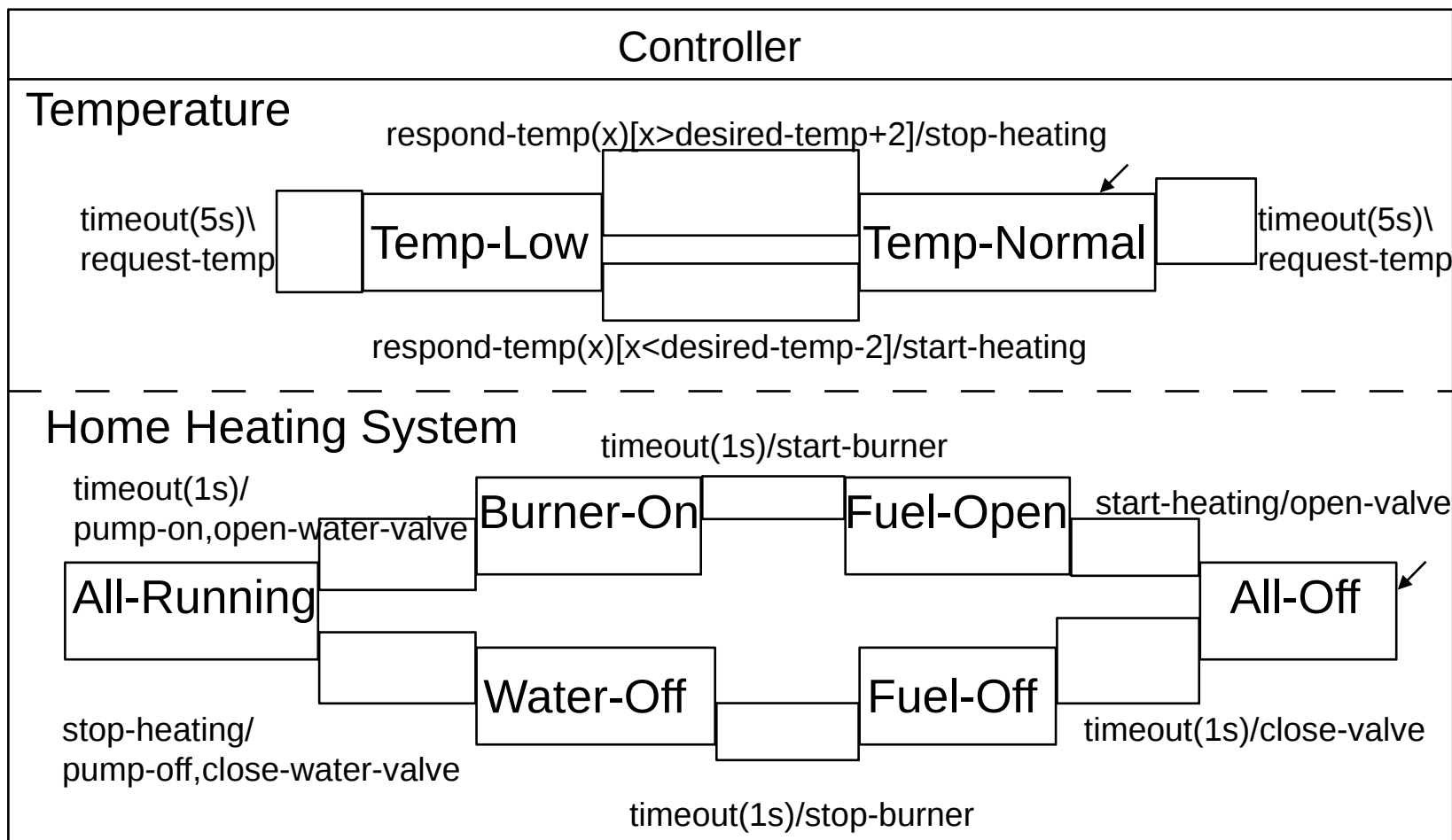
More Dynamic Model



More Dynamic Model



Even More Dynamic Model



Identify Key Operations

Operations from the object model

- Accessing and setting attributes and associations (often not shown)

Operations from events

- All events represent some operation
 - Operations from actions and activities

Actions and activities represent some processing activity within some object

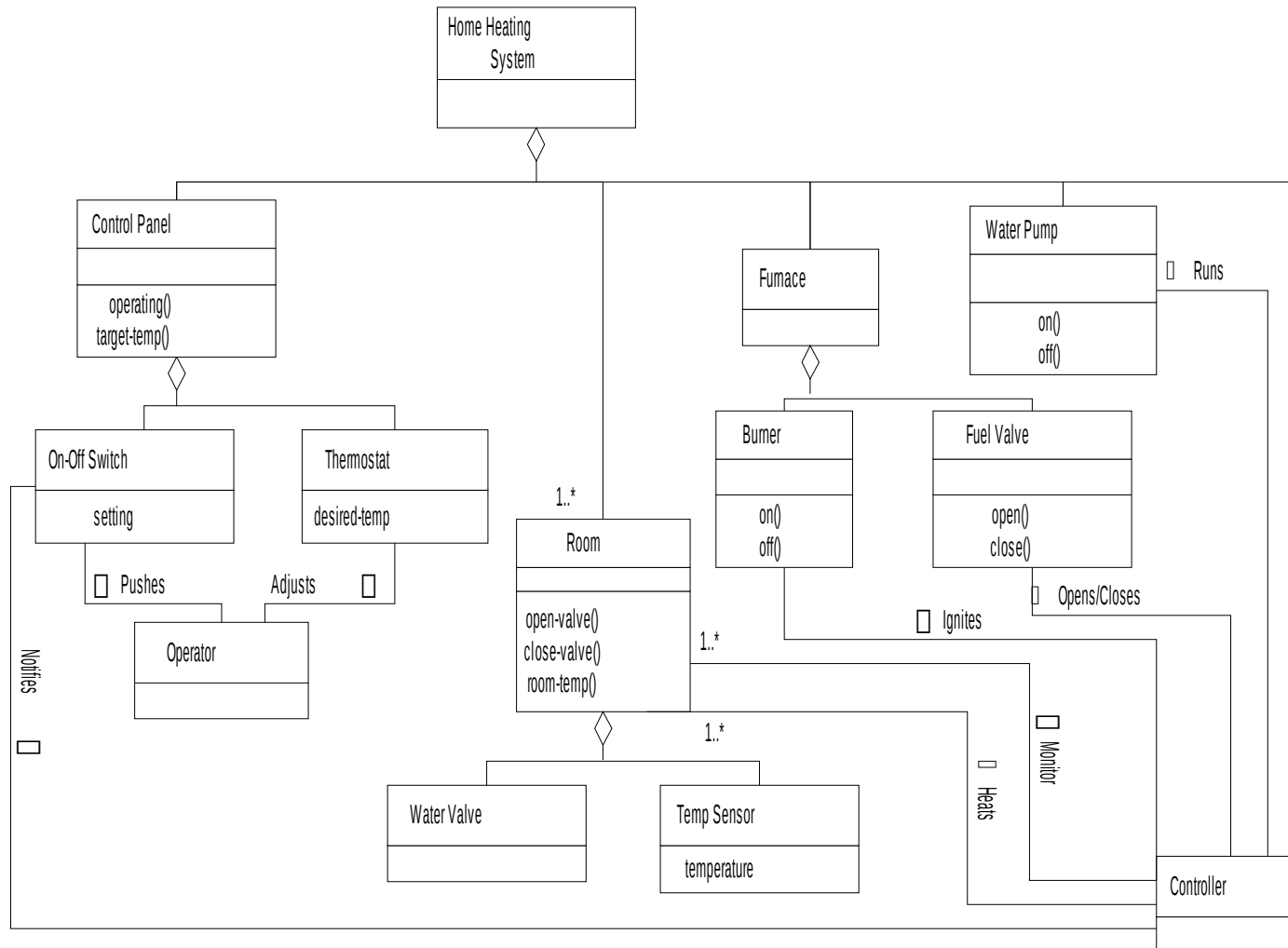
Operations from functions

- Each function typically represent one or more operations

Shopping list operations

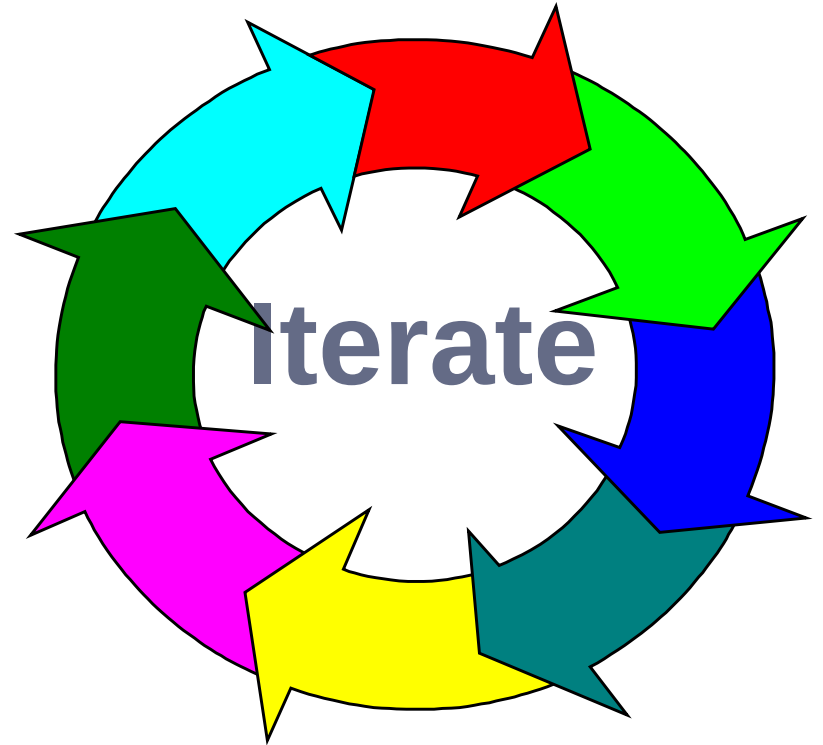
- Inherent operations (what should be there)

Complete OO Model



Iterate the Model

- Keep on doing this until you, your customer, and your engineers are happy with the model



We Have Learned

- State machines are used to capture classes of scenarios
 - They consist of states, transitions, events, and guarding conditions
 - Activities can be performed in a state
- How to find operations (methods) from the various models