Creating State Diagrams

- What are State Diagrams?
- State Diagram Symbols
- State Diagram Guidelines

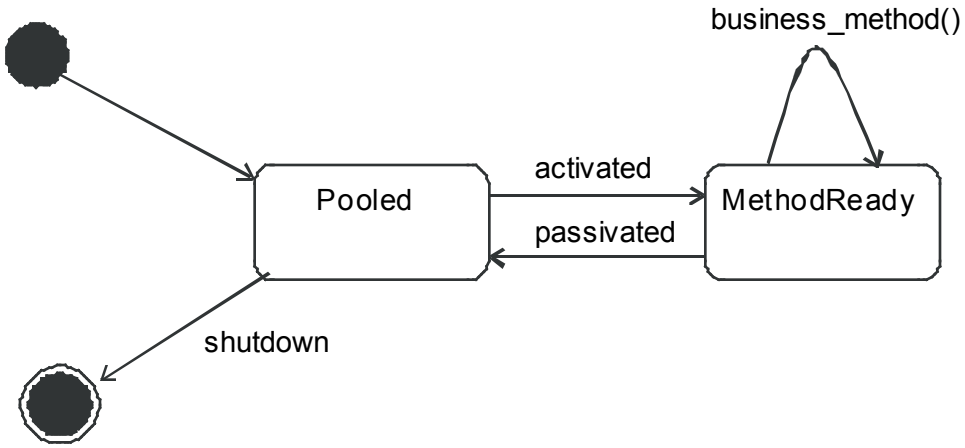
State diagrams are sometimes referred to as "statechart" diagrams or "state machines".

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

What is a State Diagram?

- A state diagram models the lifecycle and/or modes of a single, complex object
- You typically don't need state diagrams for all objects in a system, just the complicated, modal ones



2 - 2

State diagrams are different from the other UML diagrams (e.g. interaction diagrams), since they model an individual object (or class). The UML state diagram borrows symbols from older diagrams for non-OO systems (e.g. finite-state machines).

Note that objects that always react in the same way to a given event do not typically need a diagram. These modeless objects generally have a simple enough lifecycle so that there's not much benefit in creating a state diagram.

Common types of objects where state diagrams are useful include:

- Objects that participate in transactions (have states like "modified", "uncommitted").

- Objects that act as Controllers.

- Objects that are software models of devices (e.g. WinModems)

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Why Use State Diagrams?

- Complex object states are difficult to describe narratively
- State diagrams lead naturally into coding the Gang of Four State pattern
- In an iterative process, you can use state diagrams both during analysis and design



Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

State Diagram Symbols



- Start state



- End state (can have multiple)



- State

event[guard]/action



- Transition (fully dressed)

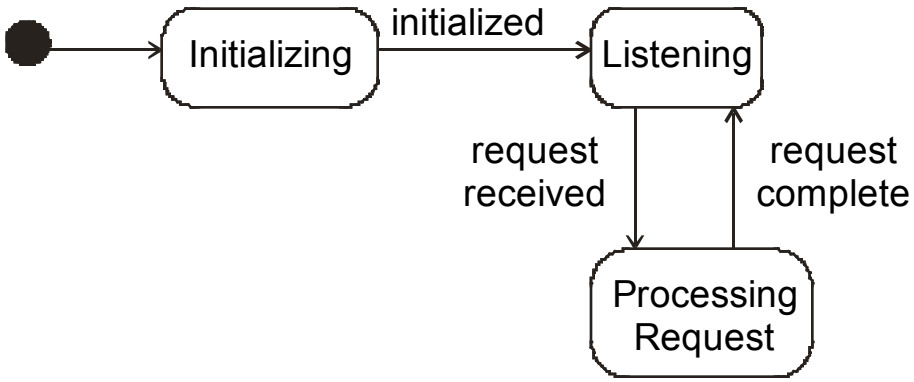
A diagram typically has a single Start State, but it's possible to have many End States. There normally is no transition required from the Start State to the first general state.

The transitions can be labeled with the event that triggered the transaction, a conditional guard and an "action". More on transitions in a moment.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Defining States

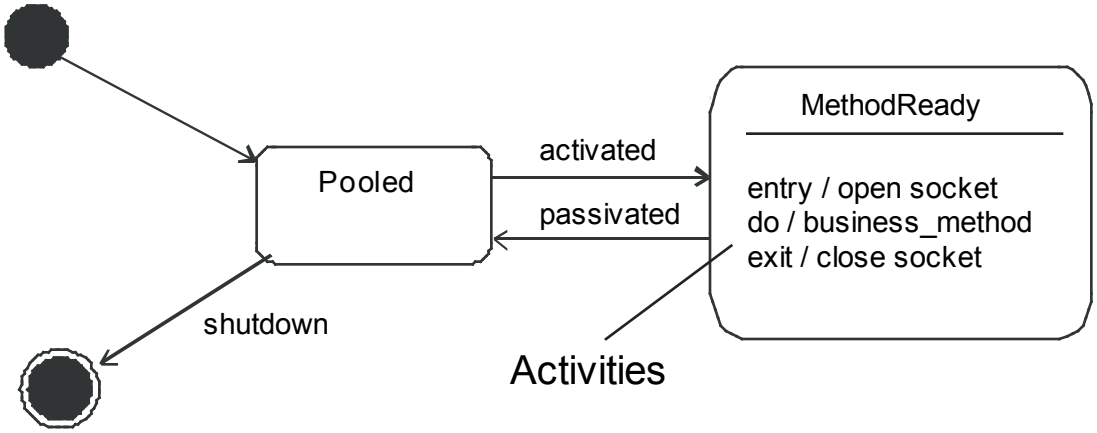
- State names should be a present-tense noun (e.g. "ready" or "waiting")



Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

States with Activities

- **Activities** are interruptible, perhaps on-going tasks that occur during a state



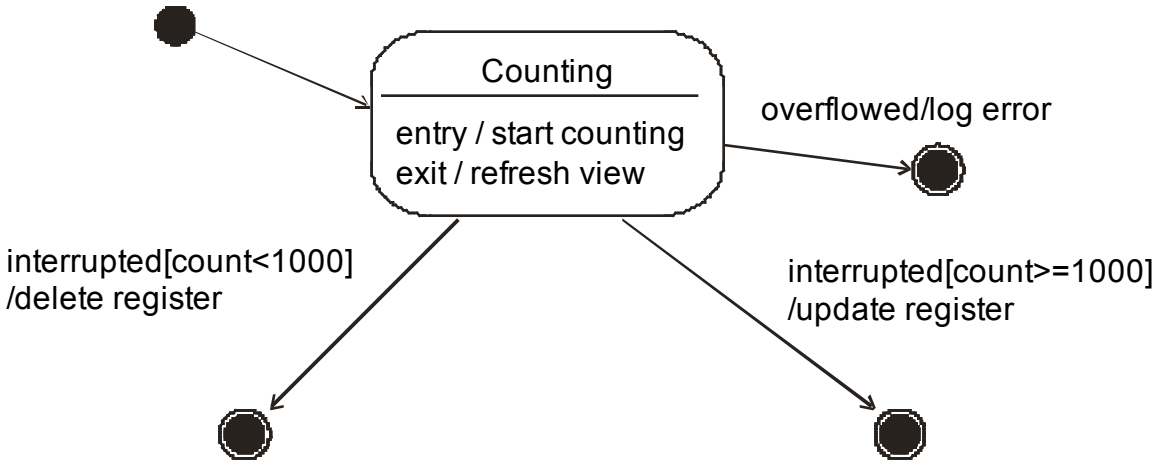
This diagram is similar to the one shown earlier, but models the self-transition as an activity. It also shows using "entry" and "exit" activities, which are common.

Activities differ from the "actions" on a transition since actions are supposed to be short and not interruptible.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Transitions

- Transitions have three optional parts: **event trigger**, **guard condition** and an **action**



Transitions (except for the initial transition) generally have an event that triggers them, and can optionally specify a conditional guard (within square brackets) and an action that's executed as part of the transition.

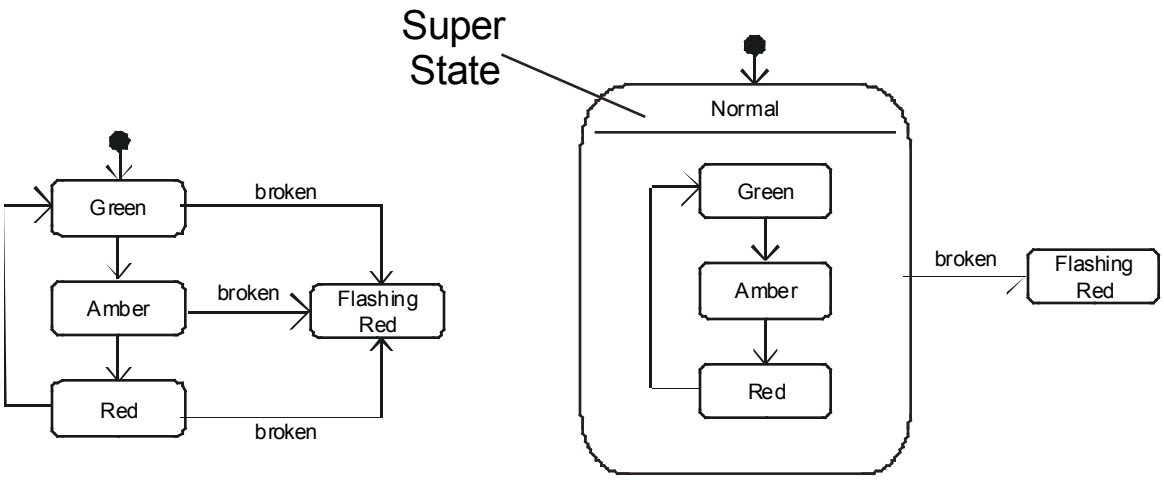
The actions should be relatively quickly executing, non-interruptible methods.

Note that by using mutually exclusive guard conditions, you can depict branching.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Superstates

- Sometimes a state diagram can be simplified using **superstates**(a.k.a composite states)



These two diagrams model the same object in the same way.

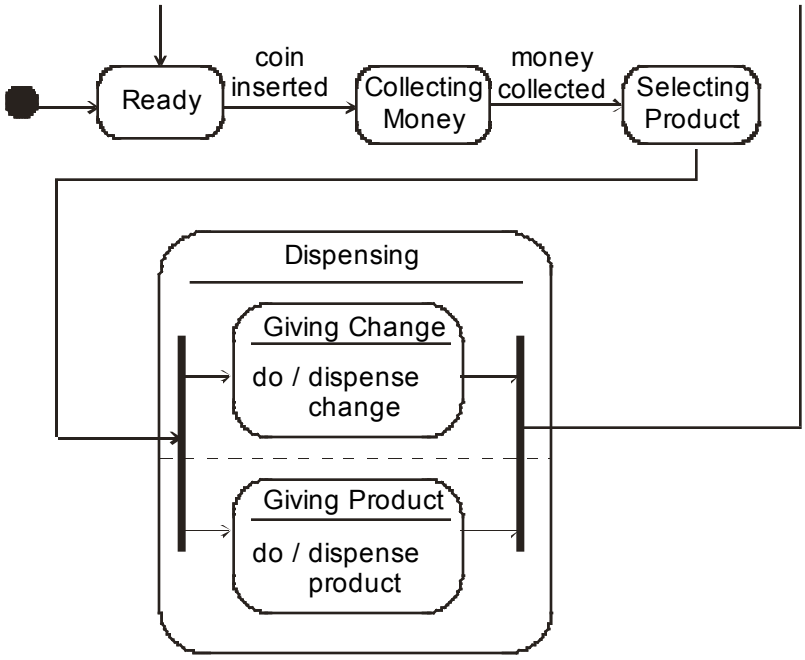
Superstates act as a container for sub-states.

Using superstates can simplify diagrams if a given state has many transitions into or out of it.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Concurrency in State Diagrams

- You can depict an object that can be in multiple states at the same time using concurrency



2 - 9

Some objects or systems can be doing multiple things at the same time. To model concurrency, you use a dashed line to separate the concurrently running states.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

State Diagram Guidelines

1. Create state diagrams when behavior differs based on state
2. Place start state in upper-left corner and final state in lower-right corner
3. State names should be simple but descriptive
4. Place transition labels near the source state
5. Guard conditions should not overlap and all cases should be accounted for

---

These guidelines are adapted from:

http://www.agilemodeling.com/style/stateChartDiagram.htm

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Review Questions

- Explain why state diagrams are not required for all of the objects in a software system.
- What's the difference between an activity and an action?

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Chapter Summary

In this chapter, you learned:

- When state diagrams are appropriate
- How to use state diagrams to model objects with complex modes or lifecycles

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum