Information Systems

Lecture 11: State diagram

Mona Taghavi

LaSalle College
Montréal

The State Machine Diagram—Identifying Object Behavior

- Sometimes, it is important for a computer system to maintain information about the status of objects.
  - For example, a customer might want to know whether a particular sale has been shipped or a manager might want to know if a customer sale has been paid for.

- When defining requirements, analysts need to identify and document which domain objects require status checking and which business rules determine valid status conditions.
  - Referring back to RMO, an example of a business rule is that a customer sale shouldn’t be shipped until it has been paid for.

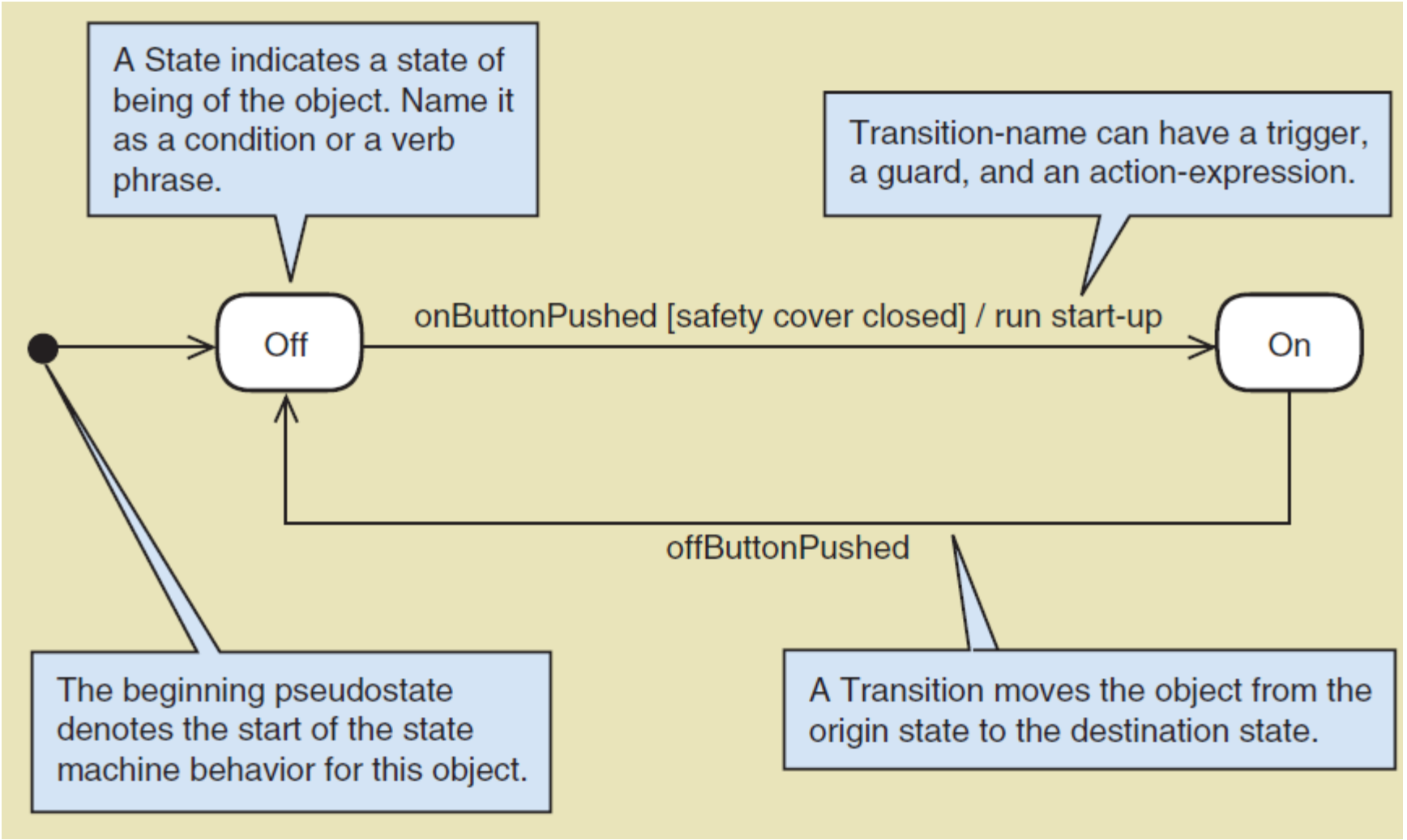
State and Transition - definition

- A **state** of an object is a condition that occurs during its life when it satisfies some criterion, performs some action, or waits for an event.
  - A state might have a name of a simple condition, such as *On* or *In repair*. Other states are more active, with names consisting of verb phrases, such as *Being shipped* or *Working*.

- An object remains in a state until some event causes it to move, or transition, to another state. A **transition**, then, is the movement of an object from one state to another state.

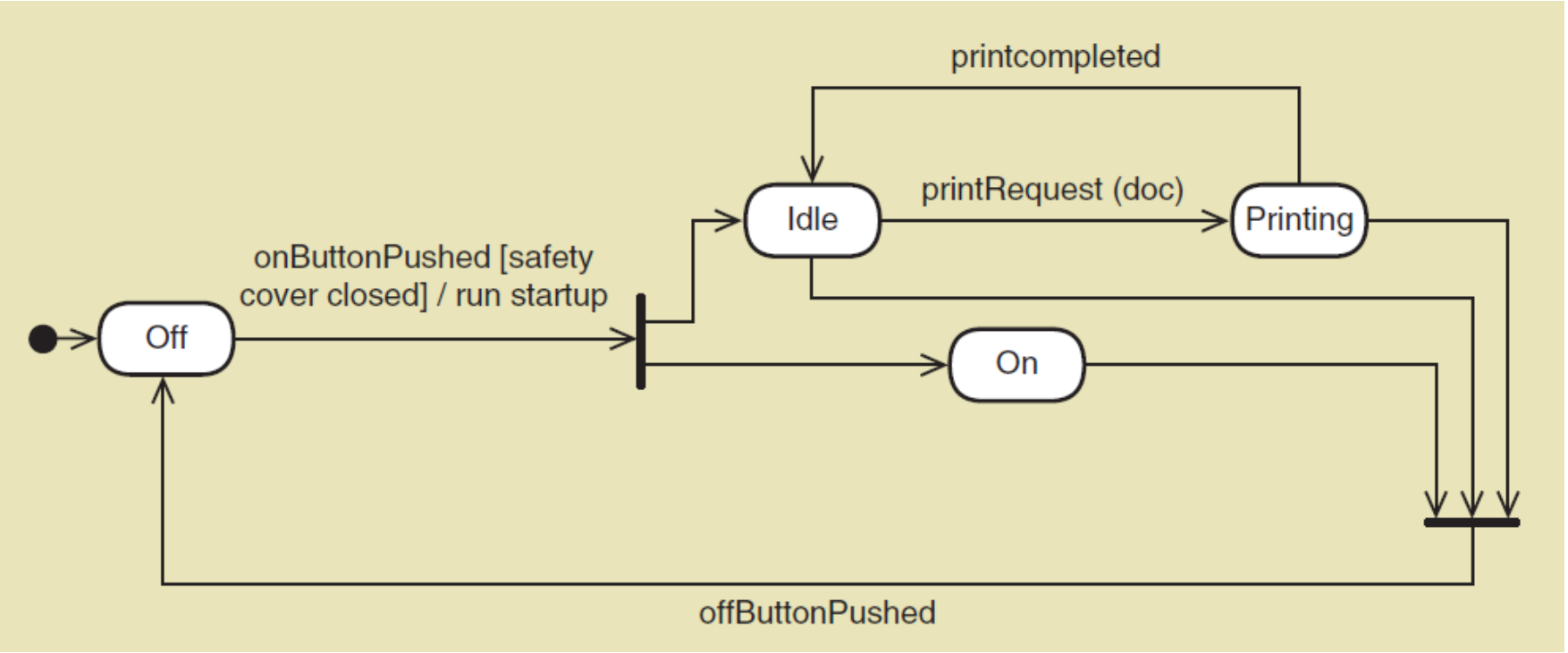
- The UML diagram that is used to describe the behavior of an object is called a **state machine diagram**.

Simple state machine diagram for a printer

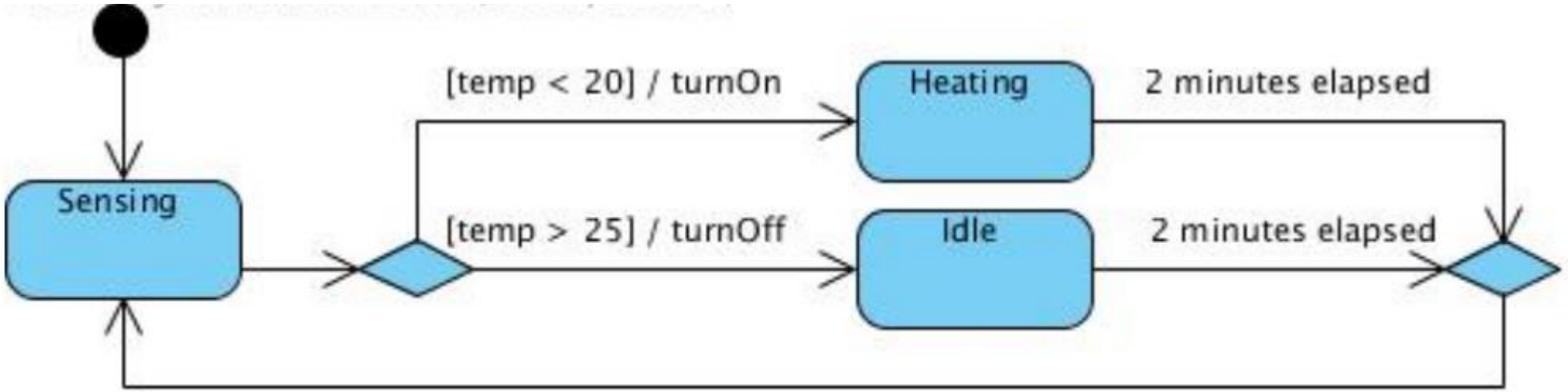


Concurrency and Concurrent States

- Sometimes an object will be in two states at the same time. This condition of being in more than one state at a time is called **concurrency**, or **concurrent states**.



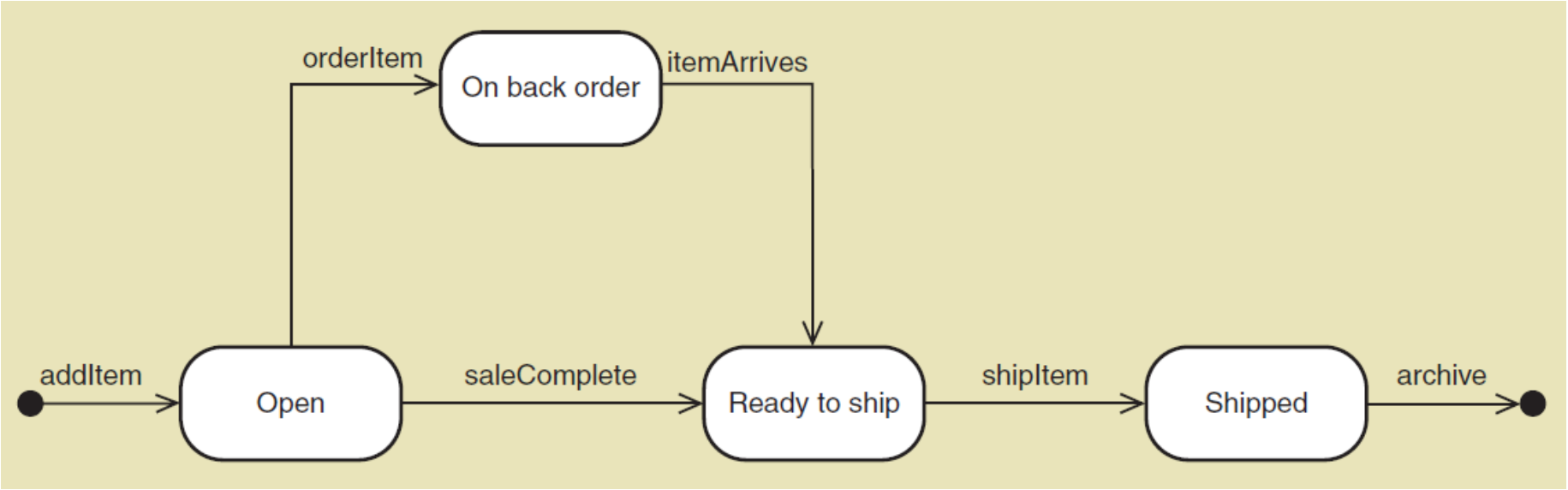
Heating device with alternate flows



Rules for Developing State Machine Diagrams

1. Review the class diagram and select the classes that might require state machine diagrams.
2. Begin building state machine diagram fragments by identifying the transitions that cause an object to leave the identified state.
3. Sequence these state-transition fragments in the correct order.
4. Review the paths and look for independent, concurrent paths.
5. Expand each transition with the appropriate message event, guard condition, and action-expression.

State machine diagram for SaleItem object



State machine diagram for InventoryItem object