FARABI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Lecture 9.
UML State Machine Diagrams

Object lifecycles and event-driven behavior

Dr. Bazargul Matkerim, Professor-Researcher

Lecture goals and plan

Learning outcomes

Identify meaningful lifecycle states.

Read triggers, guards and effects.

Explain nested and concurrent states.

Derive tests from a state model.

Lecture plan

1. States and transitions
2. Events and state behaviors
3. Composite and orthogonal states
4. Worked examples
5. Review and practice

State-dependent behavior

The modeling question

How does this object respond to an event in its current condition?

A BookCopy can be Available or On loan.

A borrow request has different outcomes in these states.

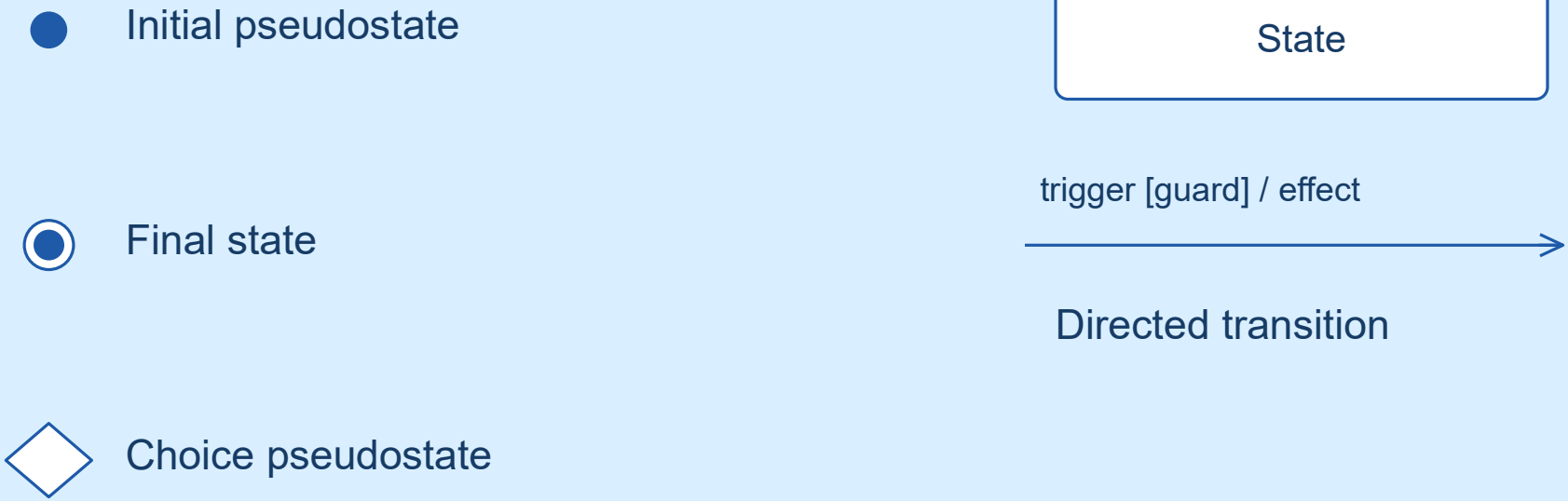
Scope of the model

Choose one entity or behavioral context.

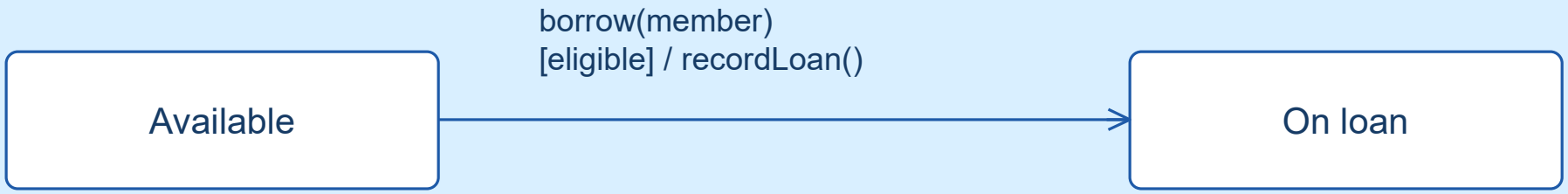
Model conditions that change allowed behavior.

Use sequence diagrams to show interactions that produce events.

Core state machine notation



Transition labels: trigger, guard and effect



Trigger and guard

The borrow event supplies a member.
The guard must evaluate to true.

Effect

recordLoan() runs as part of the transition.

Common event types

Event type	Example and meaning
Signal	paymentConfirmed(id): asynchronous signal arrives
Call	borrow(member): an operation is invoked
Time	after(5 min): a relative interval expires
Change	when(temperature > limit): condition becomes true

Event parameters provide data for guards and effects.

Initial, final and completion semantics

Entering a region

An initial pseudostate defines default entry.

A region permits at most one initial pseudostate.

Its outgoing transition has no trigger or guard.

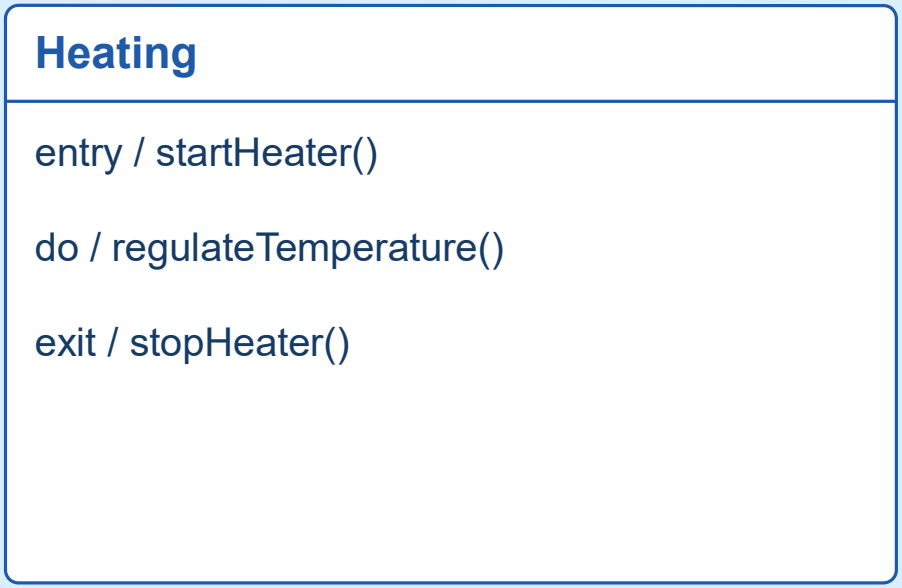
Completing a region

A final state completes its region.

A region can have more than one final state.

Completion of all top-level regions completes the state machine.

Entry, do and exit behaviors



entry

Runs when the state is entered.

do

May continue while the state is active.

exit

Runs when the state is exited.

External transitions and internal reactions

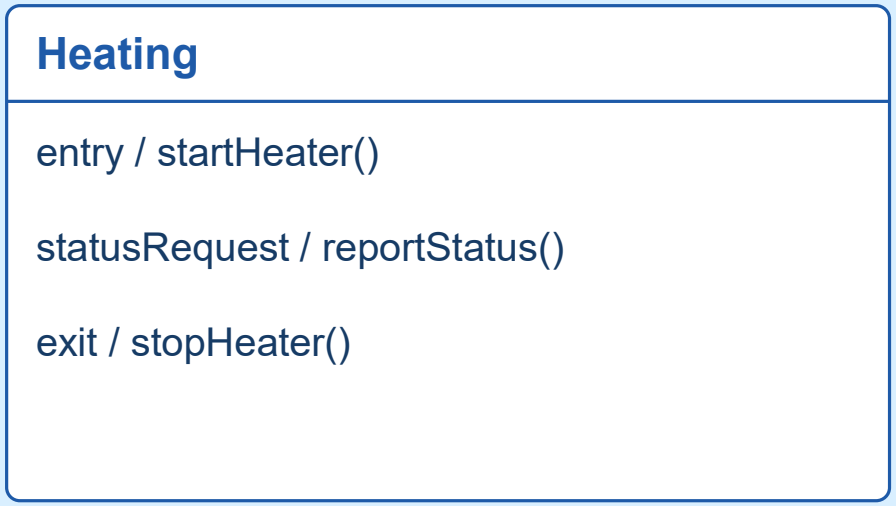
External transition

Exit the source state.

Execute the transition effect.

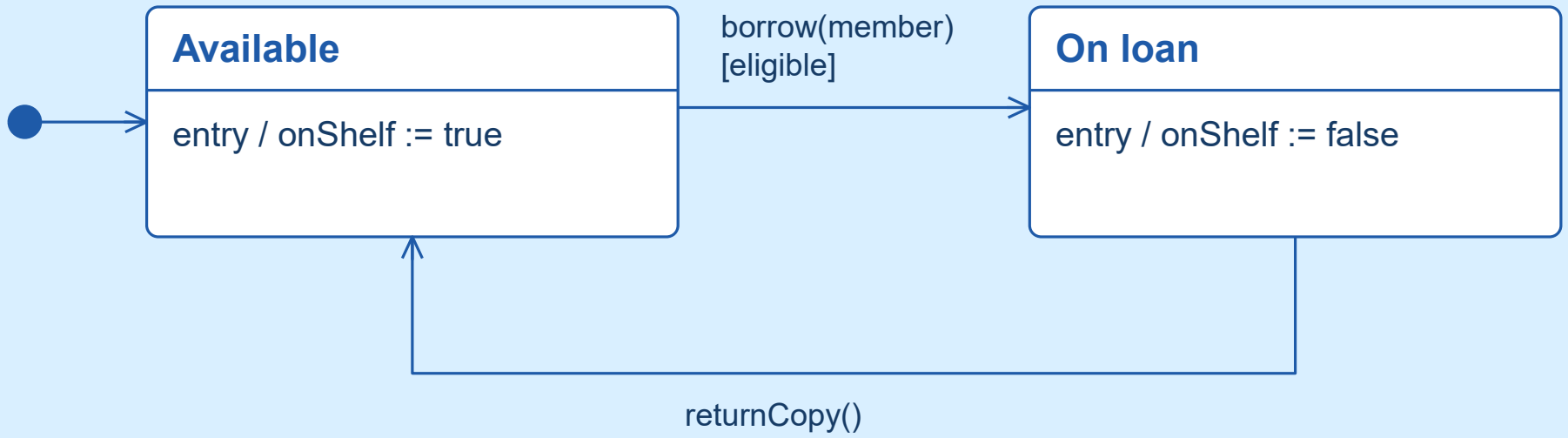
Enter the target state.

An external self-transition exits and re-enters the same state.



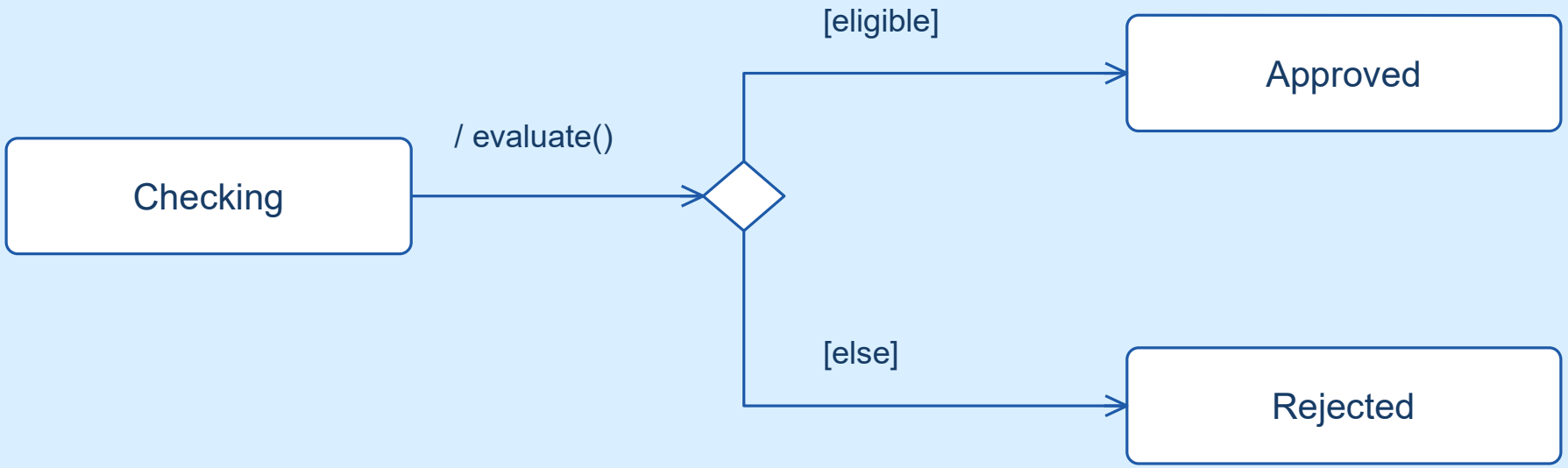
The internal statusRequest reaction preserves the active state and skips exit and entry.

BookCopy lifecycle



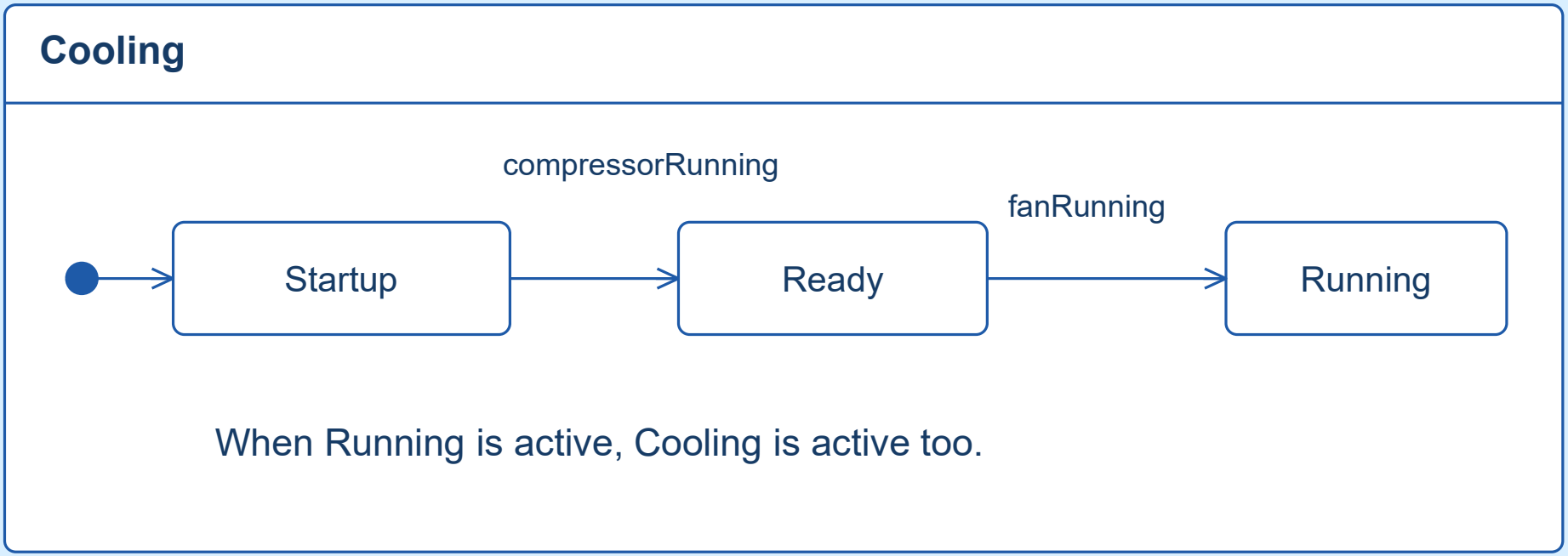
The state machine describes one copy, not every copy of the same book.

Guards and choice paths

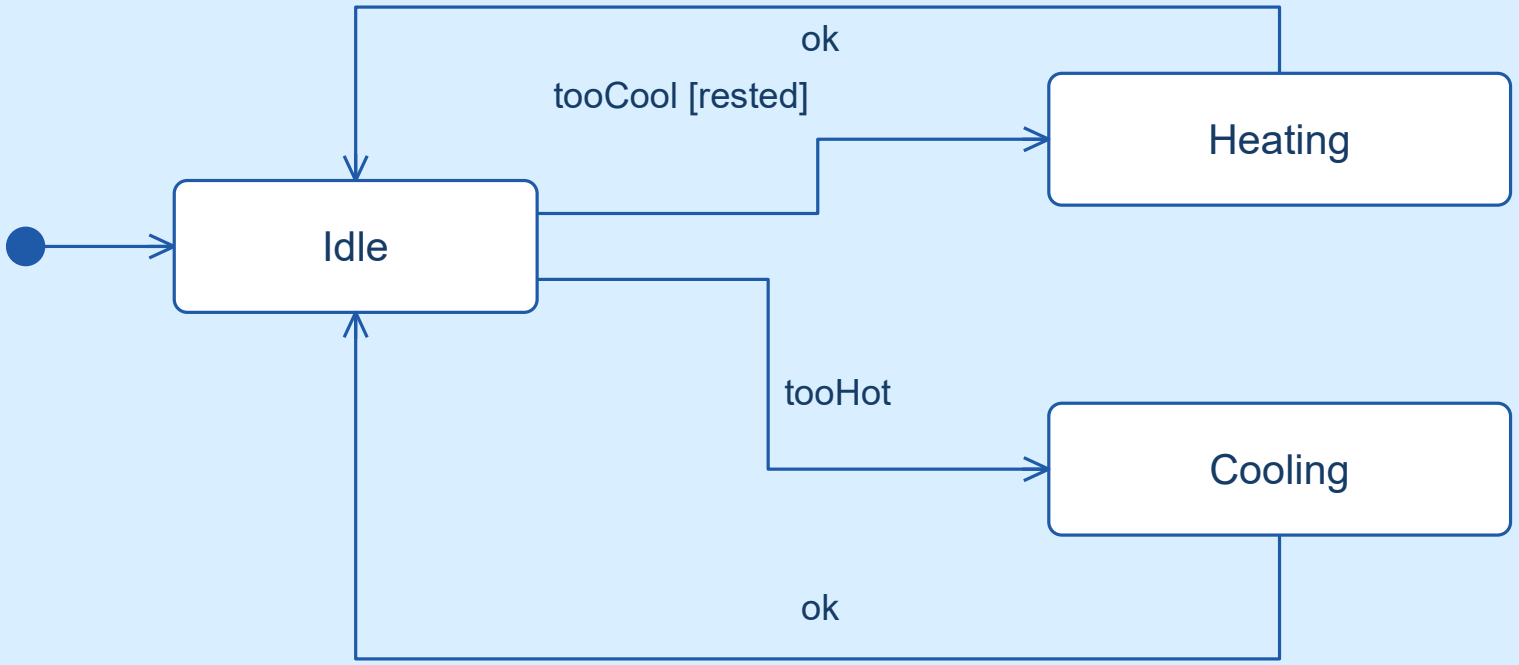


Use complete, unambiguous conditions for the intended alternatives.

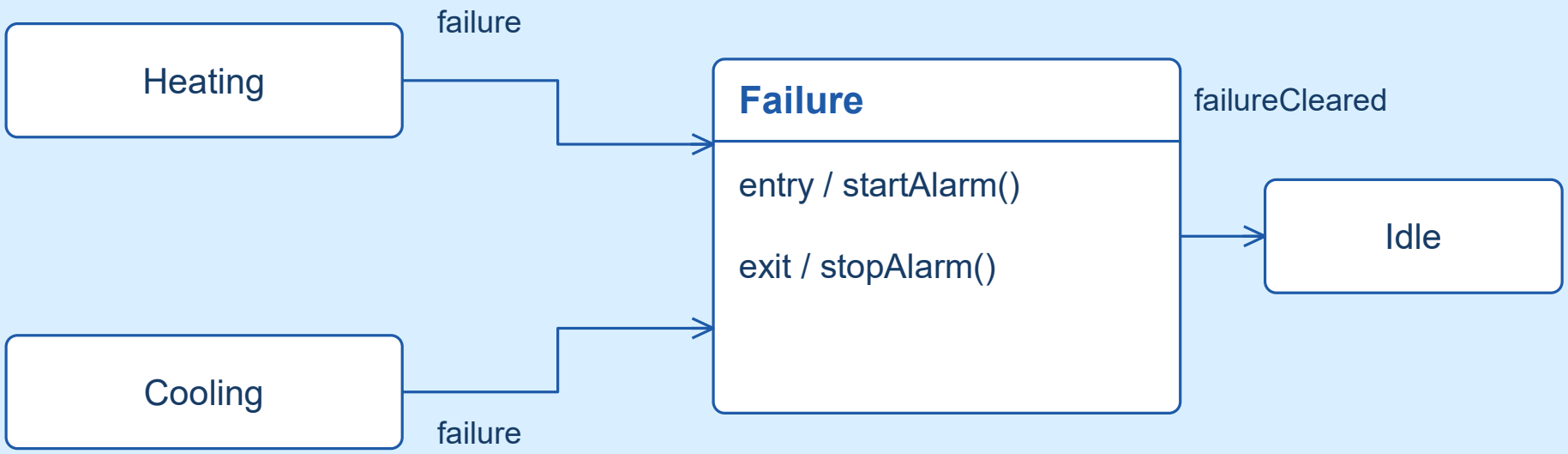
Composite states and nested behavior



Heater controller: outer lifecycle

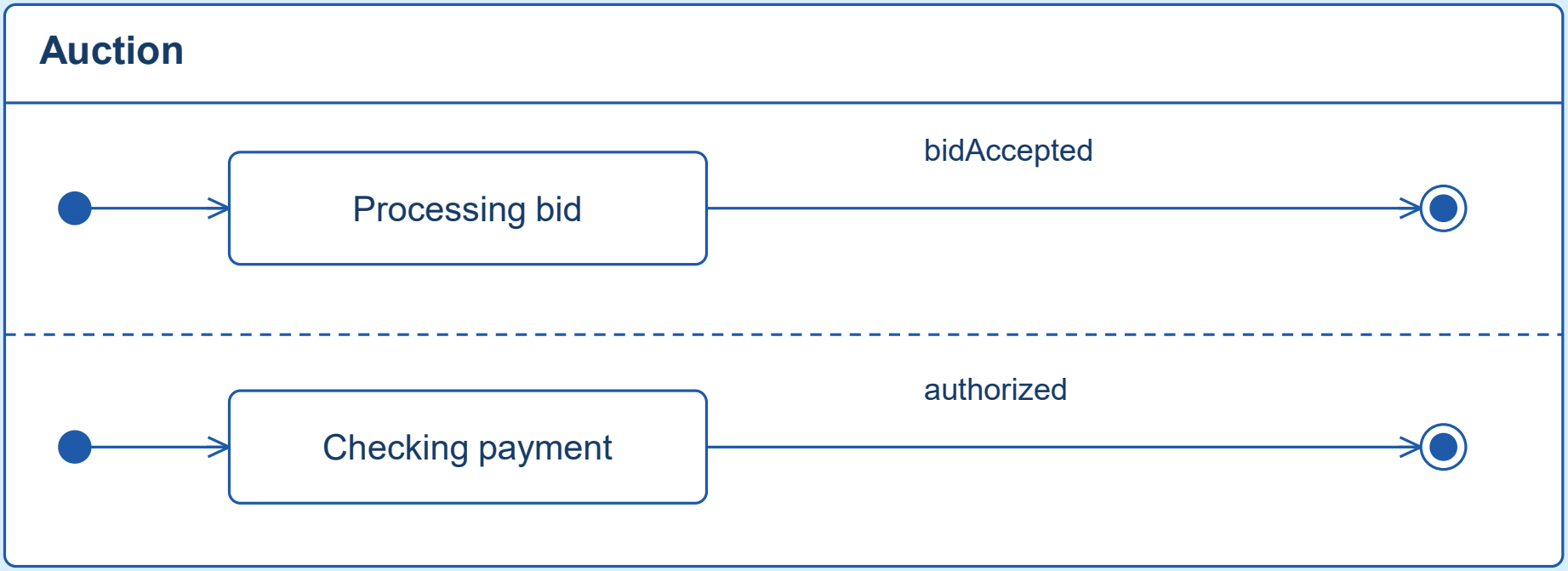


Heater failures and recovery



A transition out of Cooling exits its active nested state as well.

Orthogonal regions: auction processing



Each region progresses independently within the same composite state.

Auction completion and cancellation

Normal completion

The bid region reaches its final state.

The payment region reaches its final state.

Auction can then take a completion transition to Purchased.

Abnormal exit

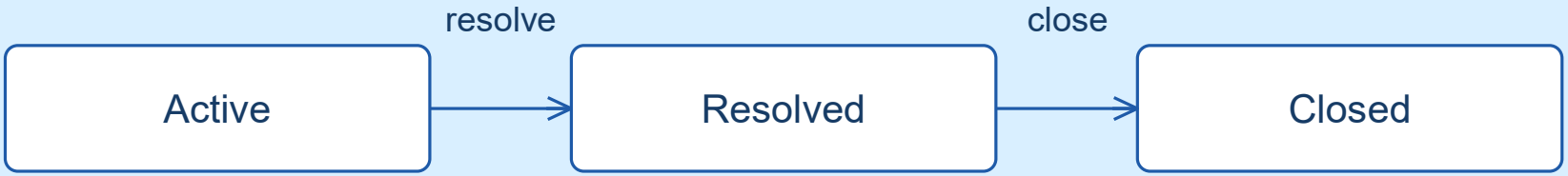
A cancel event can leave Auction.

A payment rejection can leave Auction.

Leaving the composite exits all its active regions.

Incident lifecycle and workflow

State machine: conditions of an incident



Activity diagram: work performed



State machines and activity diagrams

Aspect	State machine	Activity diagram
Main emphasis	State-dependent response	Flow of work and data
Main nodes	States and pseudostates	Actions and control/object nodes
Progress	Events and completion	Token flow and action semantics
Concurrency	Orthogonal regions	Forks, joins and parallel flows
Typical use	Object or system lifecycle	Business or software workflow

Activity diagrams can also accept events and send signals.

Review and classroom practice

Model review

Are states meaningful conditions?

Are triggers and guards distinct?

Do entry and exit effects match intent?

Are completion and failure paths clear?

Practice: test the heater

Trace Idle to Cooling.Running.

Inject failure during Startup.

Check alarm start and recovery.

Test tooCool with the rest guard false, then true.

Reading and references

Core reading

Dennis, A., Wixom, B. H., & Tegarden, D. (2021).
Systems Analysis and Design: An Object-Oriented Approach
with UML, 6th ed. Wiley. ISBN 9781119559917.

UML notation reference

Object Management Group (2017). Unified Modeling Language,
Version 2.5.1. Chapter 14: State Machines.

Examples adapted from the supplied lecture and Visual Paradigm’s state machine guides.