FARABI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Lecture 8.
UML Activity Diagrams

Workflow, decisions, concurrency and responsibility

Dr. Bazargul Matkerim, Professor-Researcher

Lecture goals and plan

Learning outcomes

Read actions and activity edges.

Distinguish alternatives from concurrency.

Model data and responsibility.

Check a workflow for missing paths or synchronization.

Lecture plan

1. Core notation and tokens
2. Decisions, forks and joins
3. Object flows and partitions
4. Worked workflow examples
5. Review and practice

Purpose and scope of an activity model

Useful modeling questions

What work must happen?

Which conditions choose a path?

Which actions may proceed concurrently?

What data and responsibilities cross boundaries?

Relationship to other diagrams

Use cases describe actor goals and scenarios.

Activities describe workflow behavior.

Sequence diagrams focus on participant messages.

State machines describe state-based behavior.

Core activity notation



Initial node

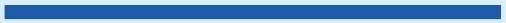


Decision or merge



Activity final

Action



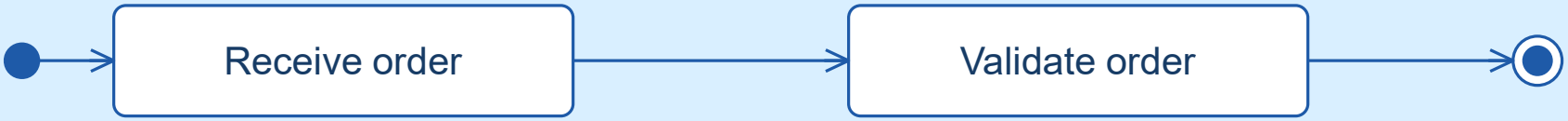
Fork or join



Flow final

The same shape can serve different roles. Read its incoming and outgoing edges.

Actions and control flows



Action

An executable step at the chosen level of detail.

Control flow

A directed edge that sequences execution using control tokens.

Starting and ending an activity



Initial node

Offers control when the activity starts. More than one is permitted.



Activity final

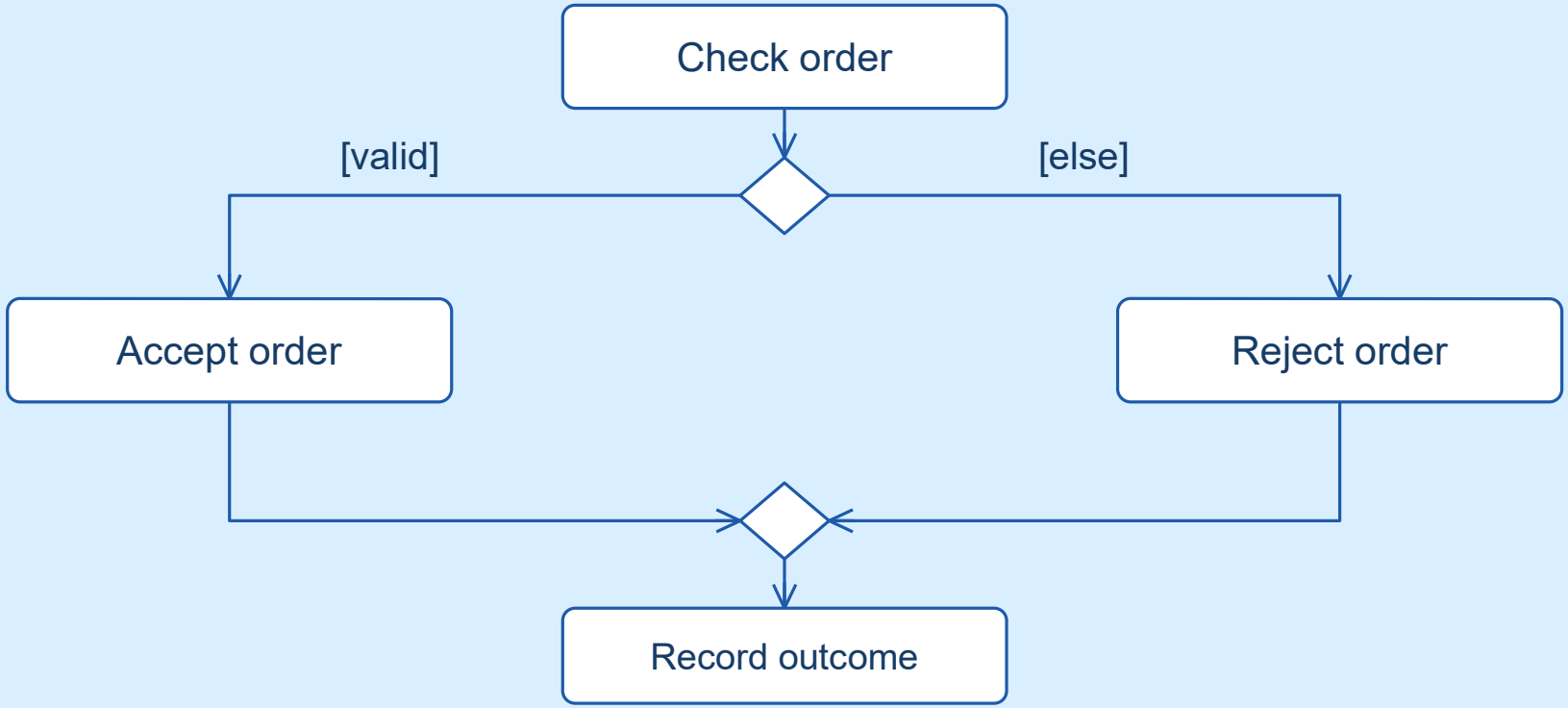
Ends the entire activity execution, including other active flows.



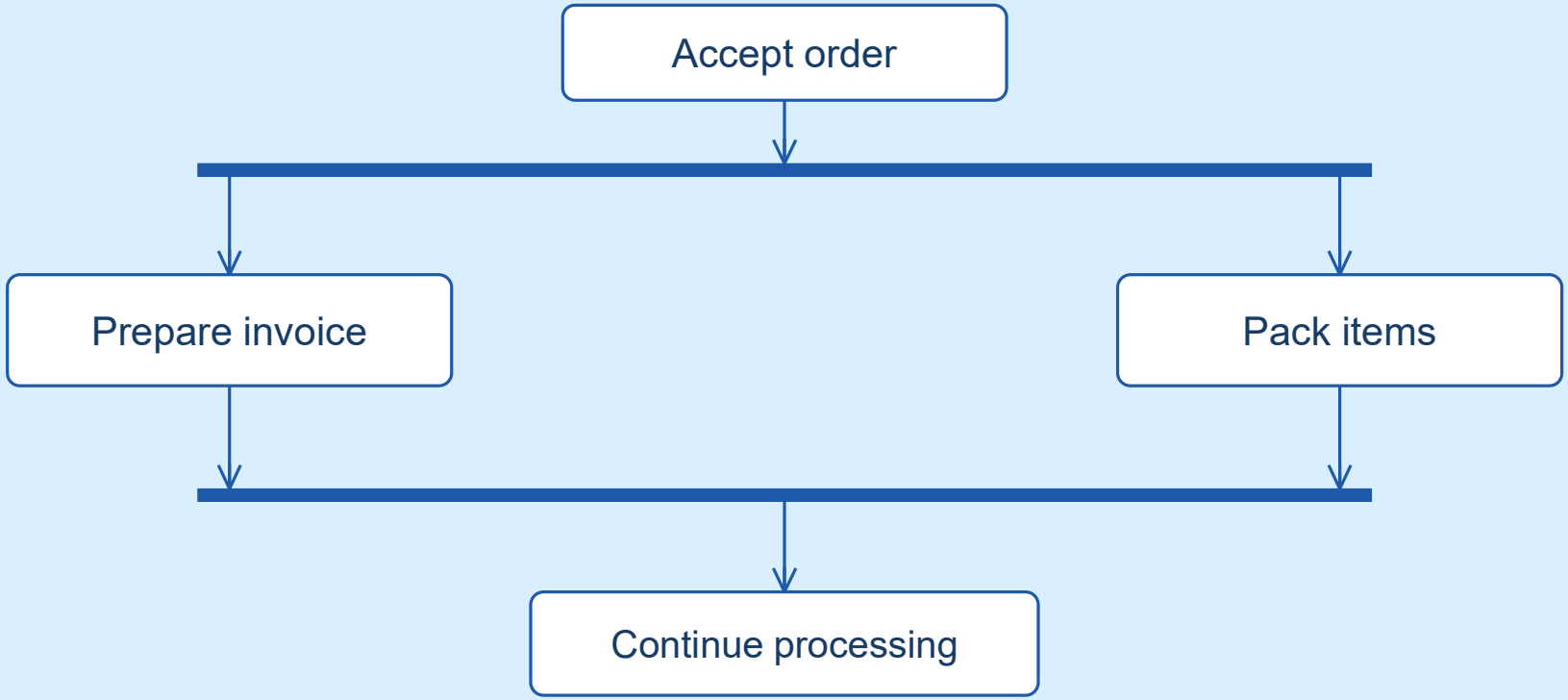
Flow final

Consumes the arriving token without ending other flows.

Decision and merge: alternative paths



Fork and join: concurrent paths



Control nodes and token behavior

Node	Effect on flow
Decision	Chooses one eligible outgoing path for a token
Merge	Passes arriving alternative flows onward
Fork	Offers flow to multiple concurrent branches
Join	Synchronizes incoming flows, normally all branches
Flow final	Ends the arriving flow
Activity final	Ends the whole activity execution

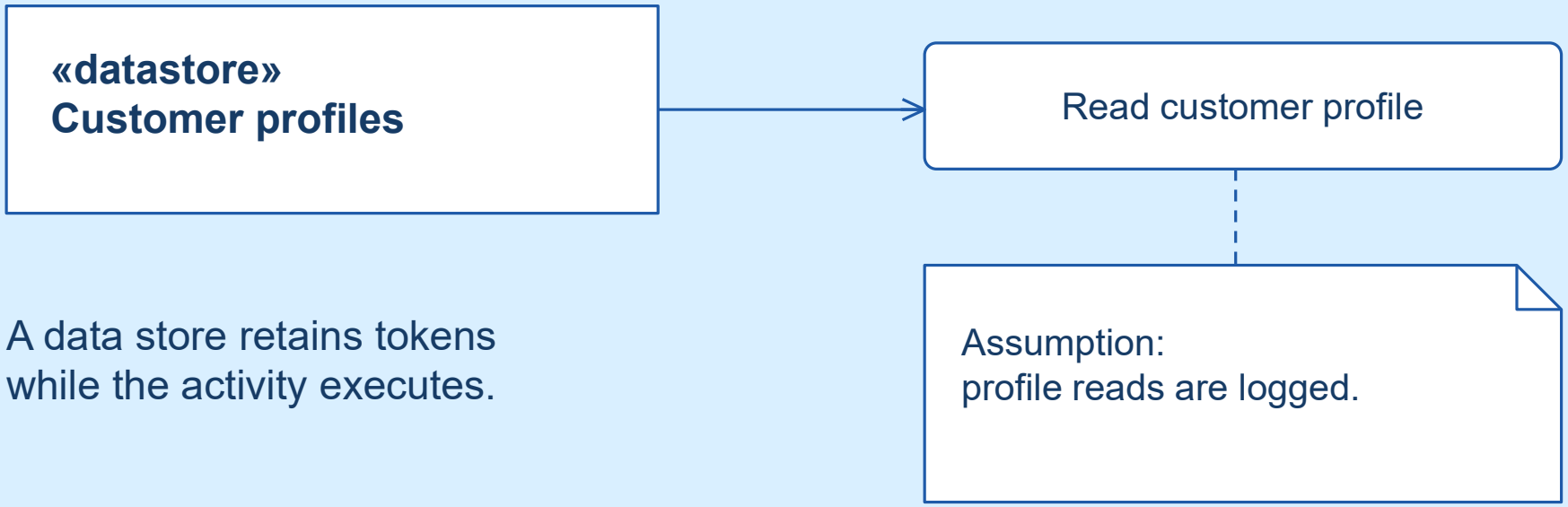
Object nodes and object flow



Object flow carries values or objects between actions.
The object node identifies the data type and optional state.

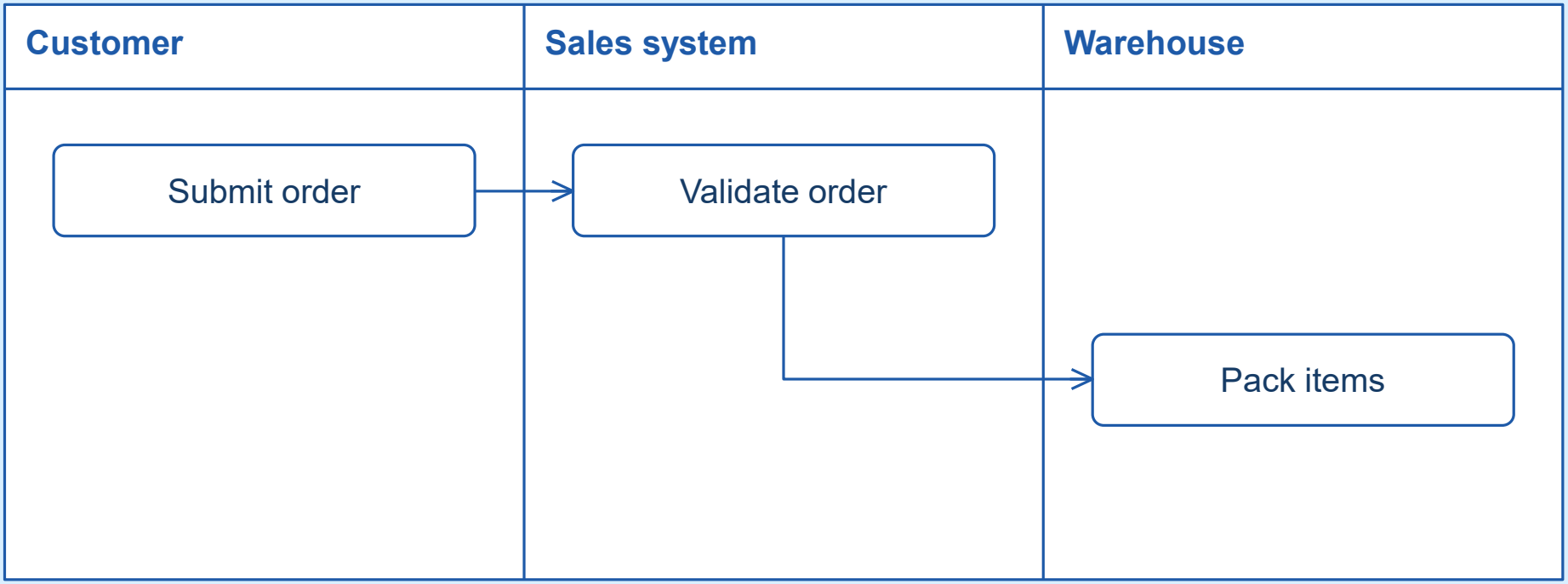
UML 2 object flows use solid arrows. Pins may show action inputs and outputs.

Data stores and comments



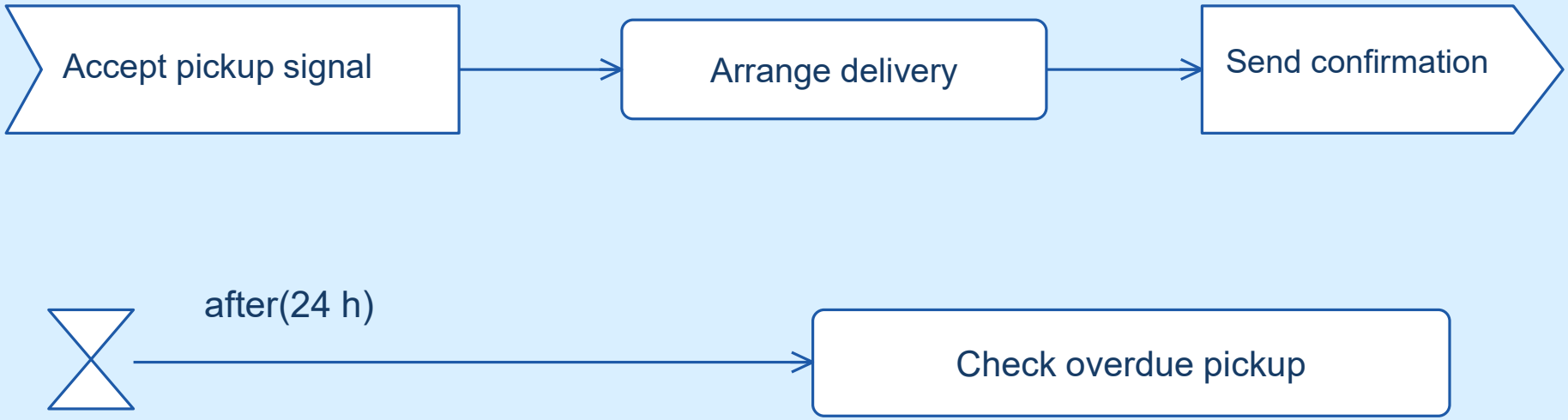
A data store retains tokens while the activity executes.

Swimlanes show responsibility

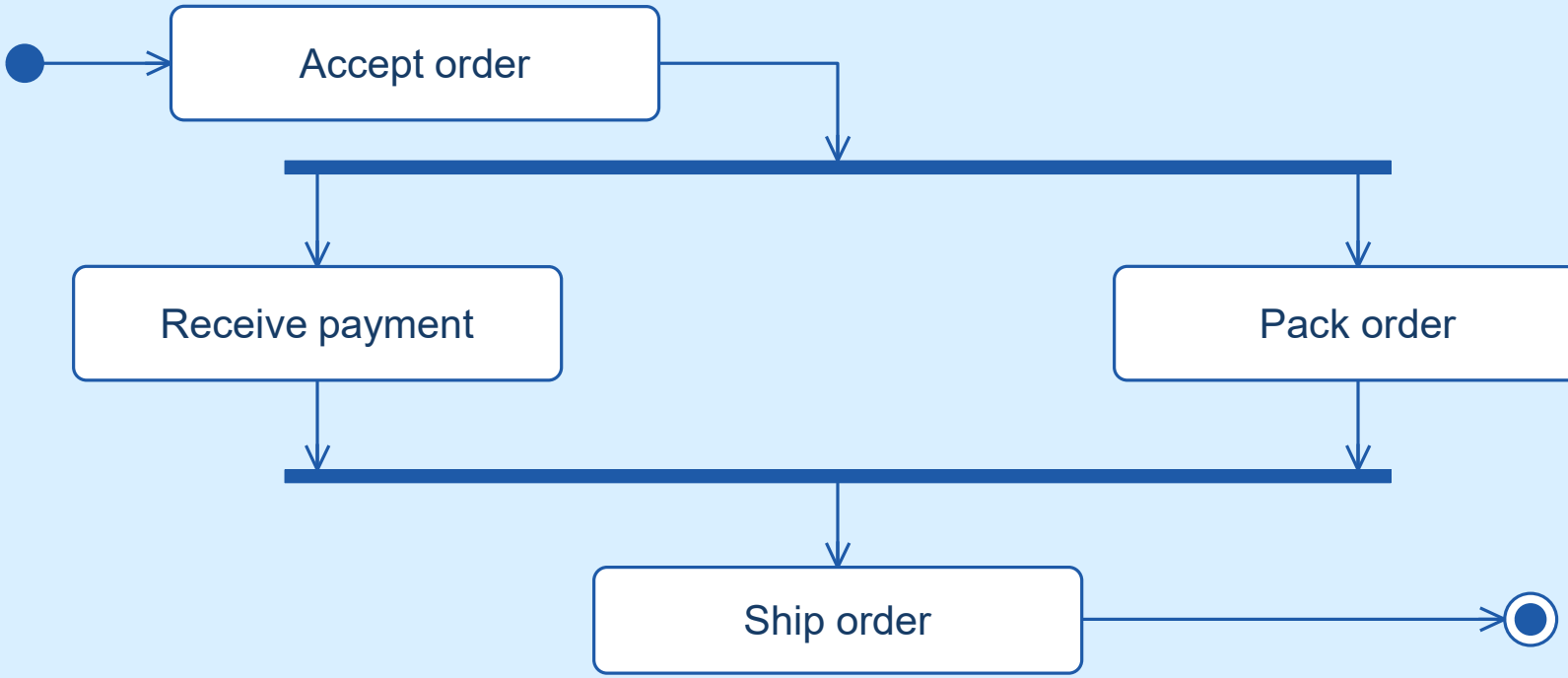


Partitions group actions. Crossing a lane boundary does not itself send a message.

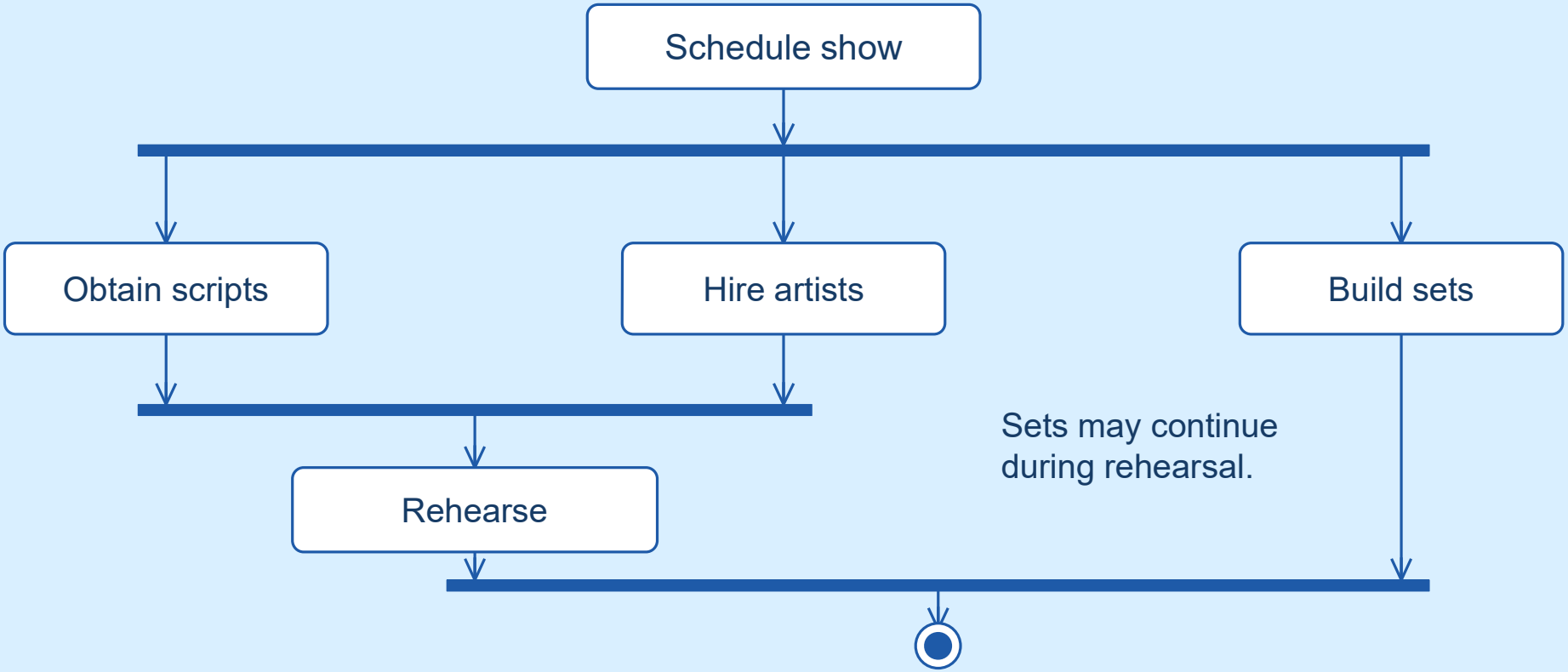
Signals and time events



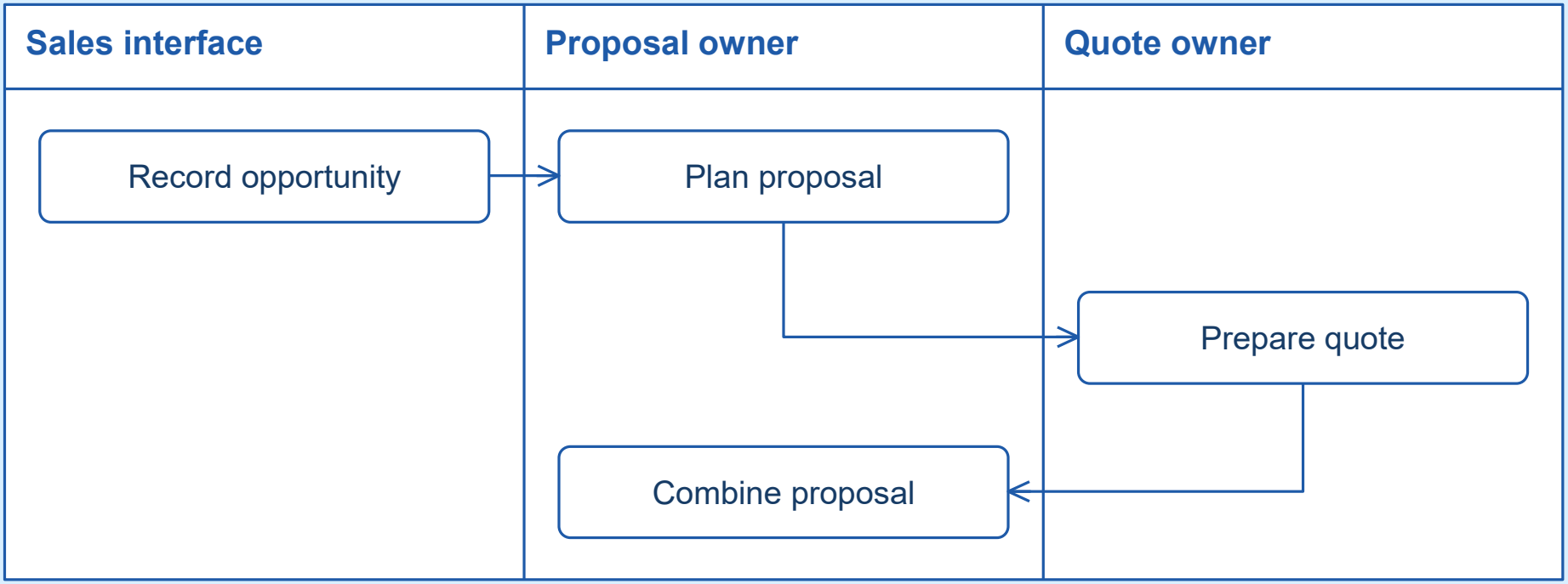
Shipping: synchronize payment and packing



Theatre planning: rehearsal dependencies

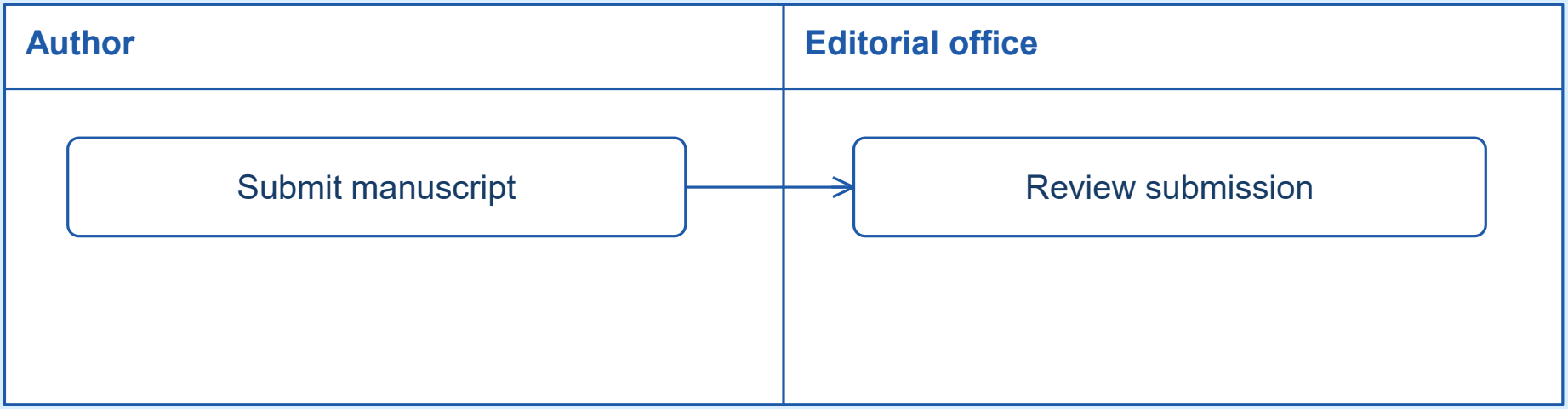


Proposal process across responsibilities



A handoff becomes visible when a flow crosses a responsibility boundary.

Article submission and partition dimensions



Responsibility: Author / Editorial office
Location: University / Publisher

Multidimensional partitions can classify the same action by role and location.

A workflow for building an activity model

Step	Question
1. Define the boundary	What starts and completes this activity?
2. Identify actions	What work transforms the inputs?
3. Add dependencies	Which actions require earlier work or data?
4. Add alternatives	Are guards complete and unambiguous?
5. Add concurrency	Which branches need synchronization?
6. Assign responsibility	Which partitions and handoffs matter?

Review and classroom practice

Checks before completion

Decision paths have meaningful guards.

Merges combine alternatives.

Joins wait only for required concurrent work.

Final nodes have the intended scope.

Practice: manuscript review

Model submission and an editorial check.

Request two reviews concurrently.

Wait for both reports before deciding.

Add acceptance and revision paths with author and editor lanes.

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. Chapters 15–16: Activities and Actions.

Examples adapted from the supplied lecture and Visual Paradigm’s Activity Diagram Tutorial.