FARABI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Lecture 7.
UML Communication Diagrams

Participant links, numbered messages and responsibilities

Dr. Bazargul Matkerim, Professor-Researcher

Lecture goals and plan

Learning outcomes

Identify participants and communication paths.

Interpret numbered and nested messages.

Build a diagram from a scenario.

Compare communication and sequence views.

Lecture plan

1. Purpose and notation
2. Message numbering
3. Movie-system examples
4. Comparison and design review
5. Classroom practice

Purpose of a communication diagram

Questions it answers

Which participants collaborate?

Which paths carry messages?

Who sends each message?

In what order do the messages occur?

An interaction view

The layout emphasizes the participant network.

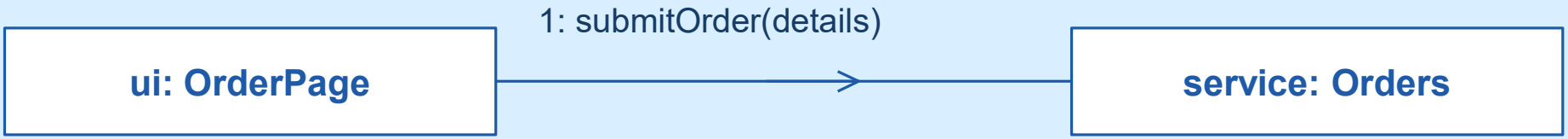
Message numbers express order and nesting.

UML 1.x called this a collaboration diagram.

Communication and sequence views

Aspect	Communication diagram	Sequence diagram
Main emphasis	Participant connections	Order of message events
Layout	Free arrangement of roles	Lifelines across the page
Ordering	Numbered message labels	Vertical event positions
Useful for	Responsibility review	Detailed scenario flow
Complex control	Can become hard to read	Frames for choices and loops

Participants, paths and messages



Participant

A role or instance taking part in this interaction.

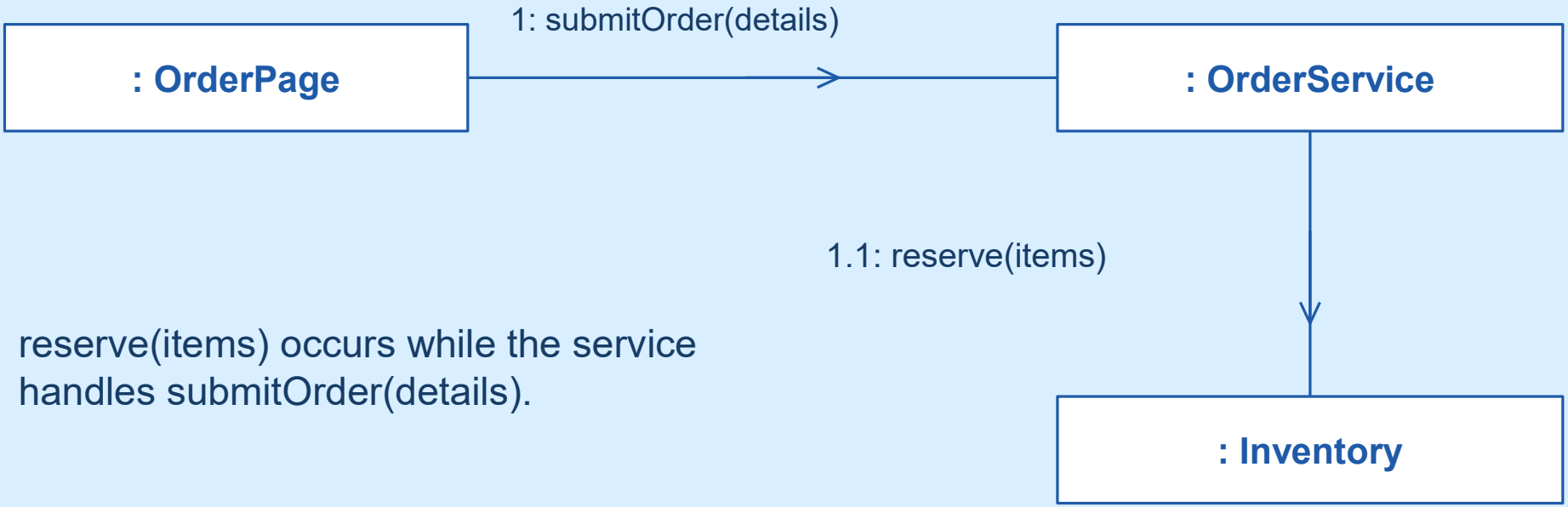
Communication path

A connection used to exchange the numbered message.

Message order and nesting

Label	Meaning
1: submitOrder()	The initial request
1.1: validate()	First call made while handling message 1
1.2: reserve()	Next call at the same nesting level
1.2.1: checkStock()	A call nested within handling message 1.2
2: displaySummary()	A later top-level message

Nested calls in an order interaction



reserve(items) occurs while the service handles submitOrder(details).

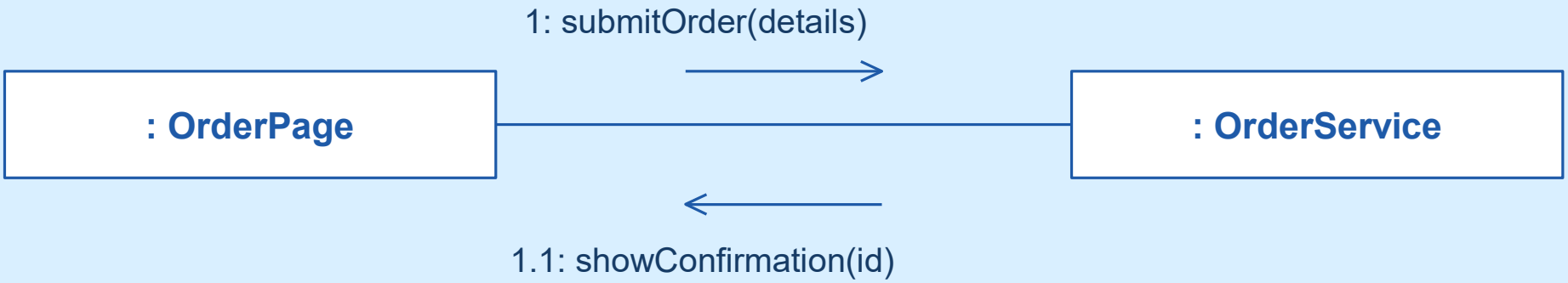
Moving the boxes does not change the message order.

Guards and repeated messages

Notation example	Interpretation
1.2 [available]: reserve(item)	Send only when the guard is true
1.2 [not available]: reject(item)	Alternative under the opposite guard
1.1 *[for each item]: check(item)	Repeat for the selected items

Guards state conditions. Iteration states what repeats.
Use a sequence diagram when complex control obscures the participant network.

Several messages on one path



The arrows identify the sender and receiver.
The shared path can carry messages in both directions.

Movie storage: scenario and responsibilities

Use cases in the source

Add a movie.

Show all movies.

Each use case needs its own focused interaction.

Participants

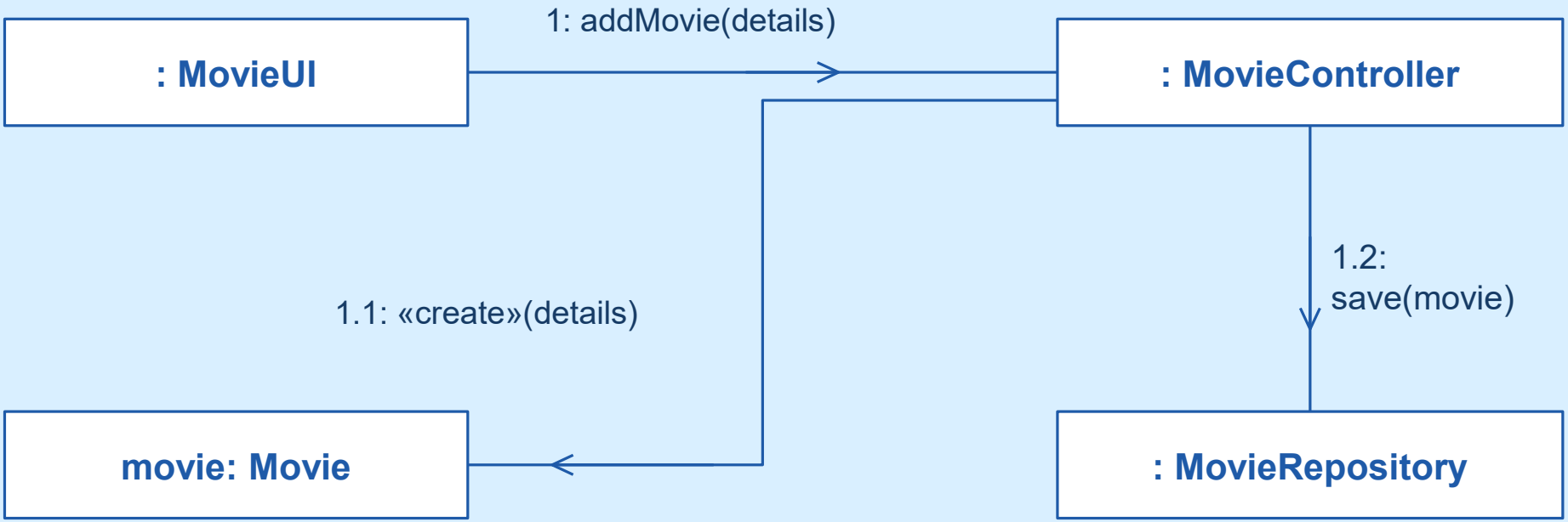
MovieUI collects input and displays results.

MovieController coordinates the request.

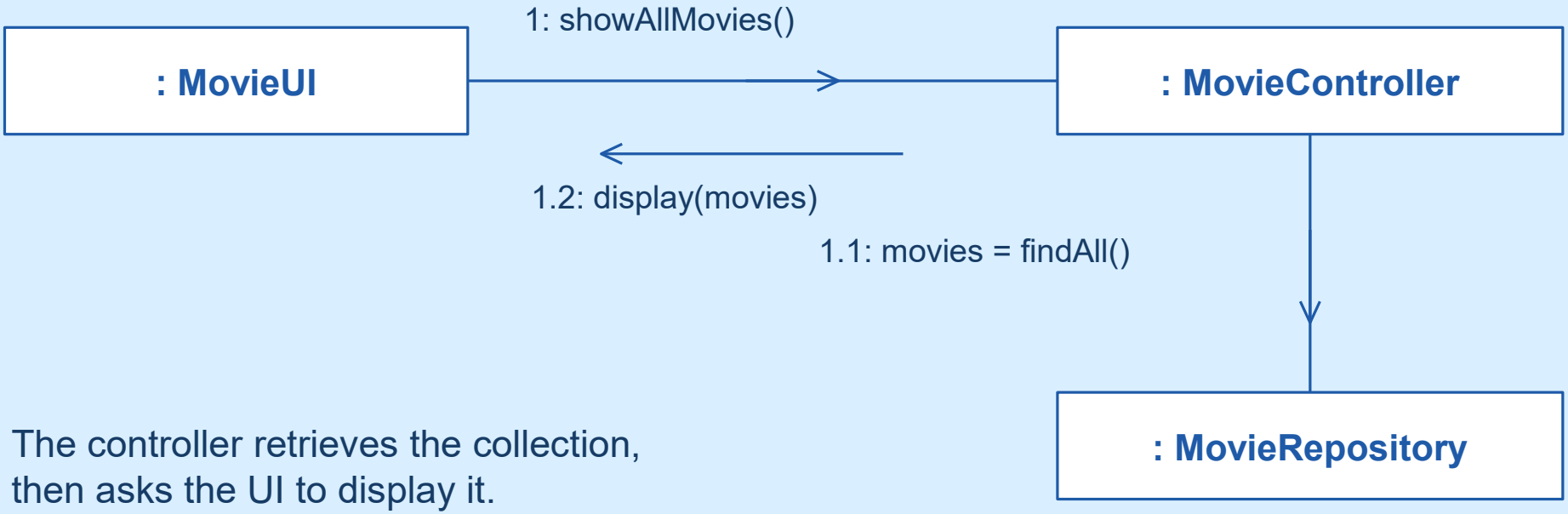
Movie represents domain data.

MovieRepository stores and retrieves it.

Add movie: communication diagram

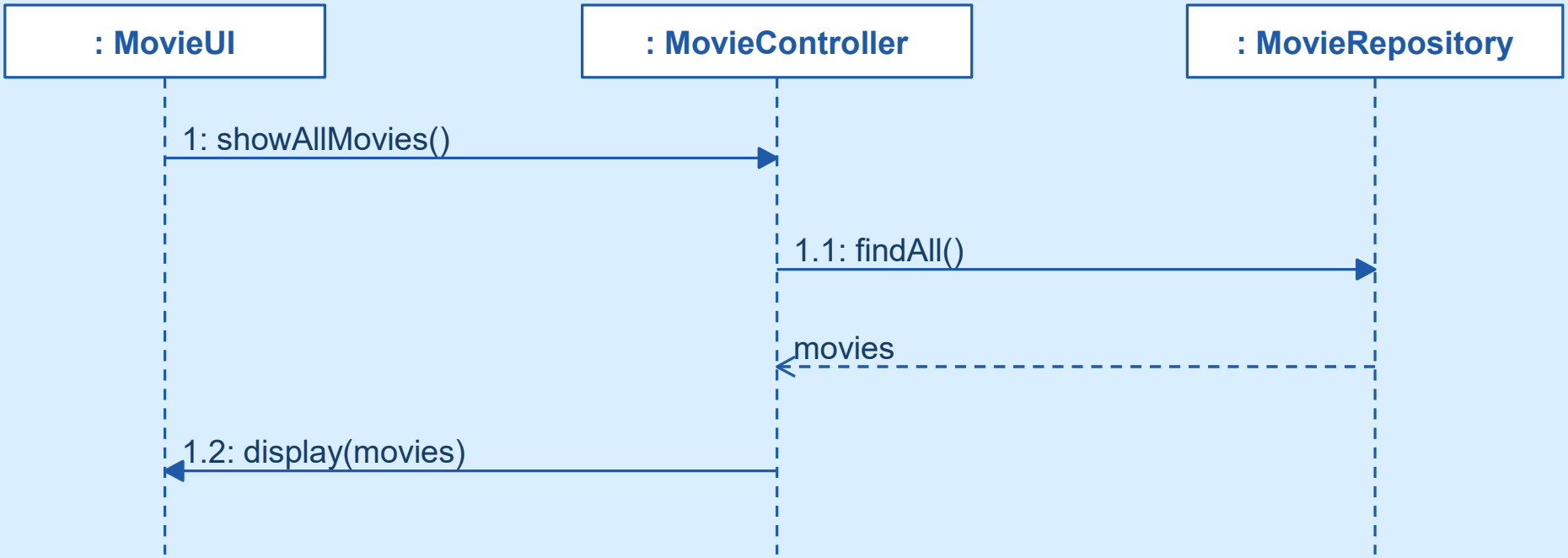


Show all movies: communication diagram



The controller retrieves the collection, then asks the UI to display it.

The same interaction as a sequence



The participants and calls match slide 12. Vertical placement makes their order visible.

Converting a simple interaction

Preserve the semantics

Keep the same participants.

Keep sender, receiver and message names.

Preserve call nesting and meaningful returns.

Carry over guards and repetition.

Change the presentation

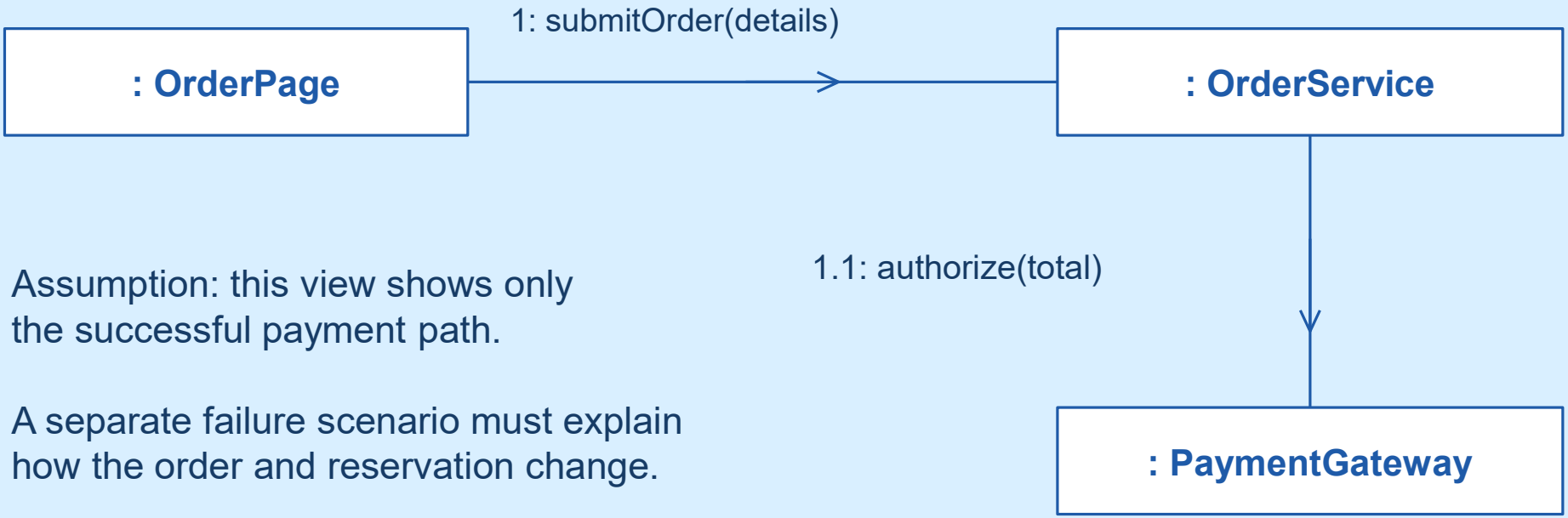
Replace vertical ordering with sequence labels.

Arrange participants to expose their connections.

Group messages on shared paths.

Check the scenario in both views.

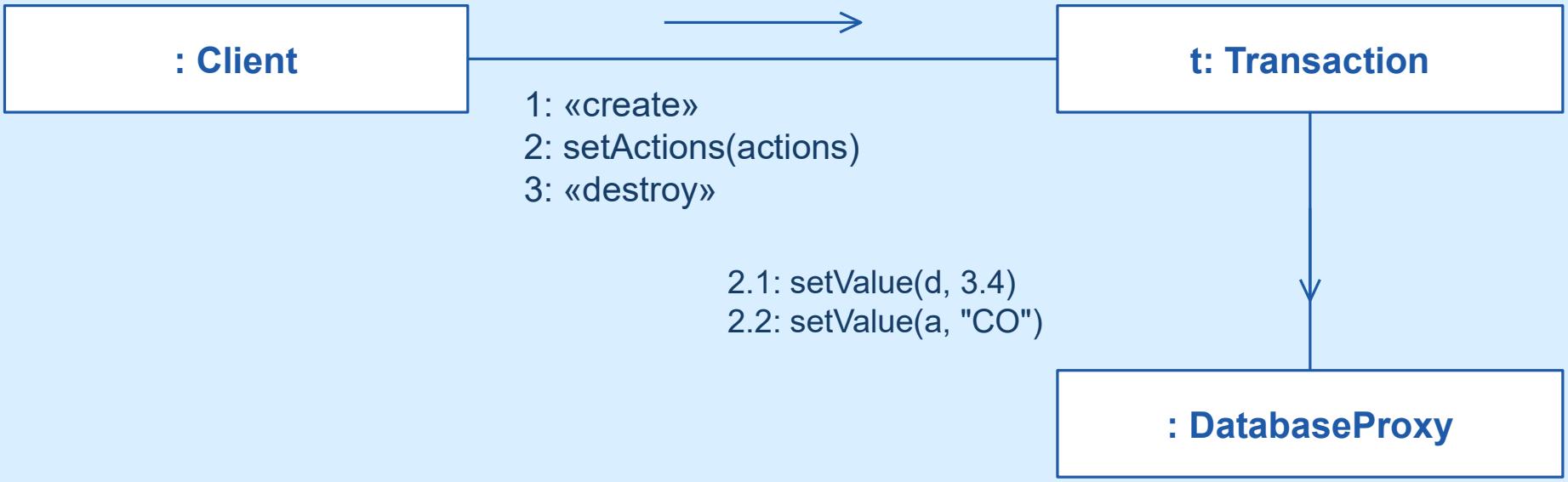
Order submission and payment



Assumption: this view shows only the successful payment path.

A separate failure scenario must explain how the order and reservation change.

Transaction example: calls and lifecycle



The two proxy calls occur within message 2, before message 3 destroys the transaction.

Responsibility and coupling review

Questions for each participant

Does this object own the requested behavior?

How does the sender obtain the receiver?

Does one object coordinate too many unrelated tasks?

Questions for each path

Is the connection needed for this scenario?

Does the interface expose the required operation?

Would a different responsibility assignment simplify the design?

A workflow for communication modeling

Step	Check
1. Select one scenario	State its trigger and outcome
2. Identify participants	Use consistent roles and types
3. Draw communication paths	Include the paths the scenario uses
4. Add directional messages	Identify sender, receiver and arguments
5. Number and qualify messages	Show nesting, guards and repetition
6. Trace and review	Check the complete scenario and design

Review and classroom practice

Common mistakes

Reading box position as time order.

Using ambiguous message numbers.

Confusing a callback with a returned value.

Drawing every implementation reference.

Practice: movie availability

Extend showAllMovies with available titles only.

Identify the participant that filters the results.

Number the calls and add any guard or iteration.

Draw the same interaction as a sequence.

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. Section 17.9: Communication Diagram.

Teaching examples adapted from the supplied communication diagrams lecture.