National University
Of Computer and Emerging Sciences

Software Design and Architecture

Logical Models: Interaction Models
Sequence Diagram

Sajid Anwer

Department of Software Engineering,
FAST-NUCES, CFD Campus

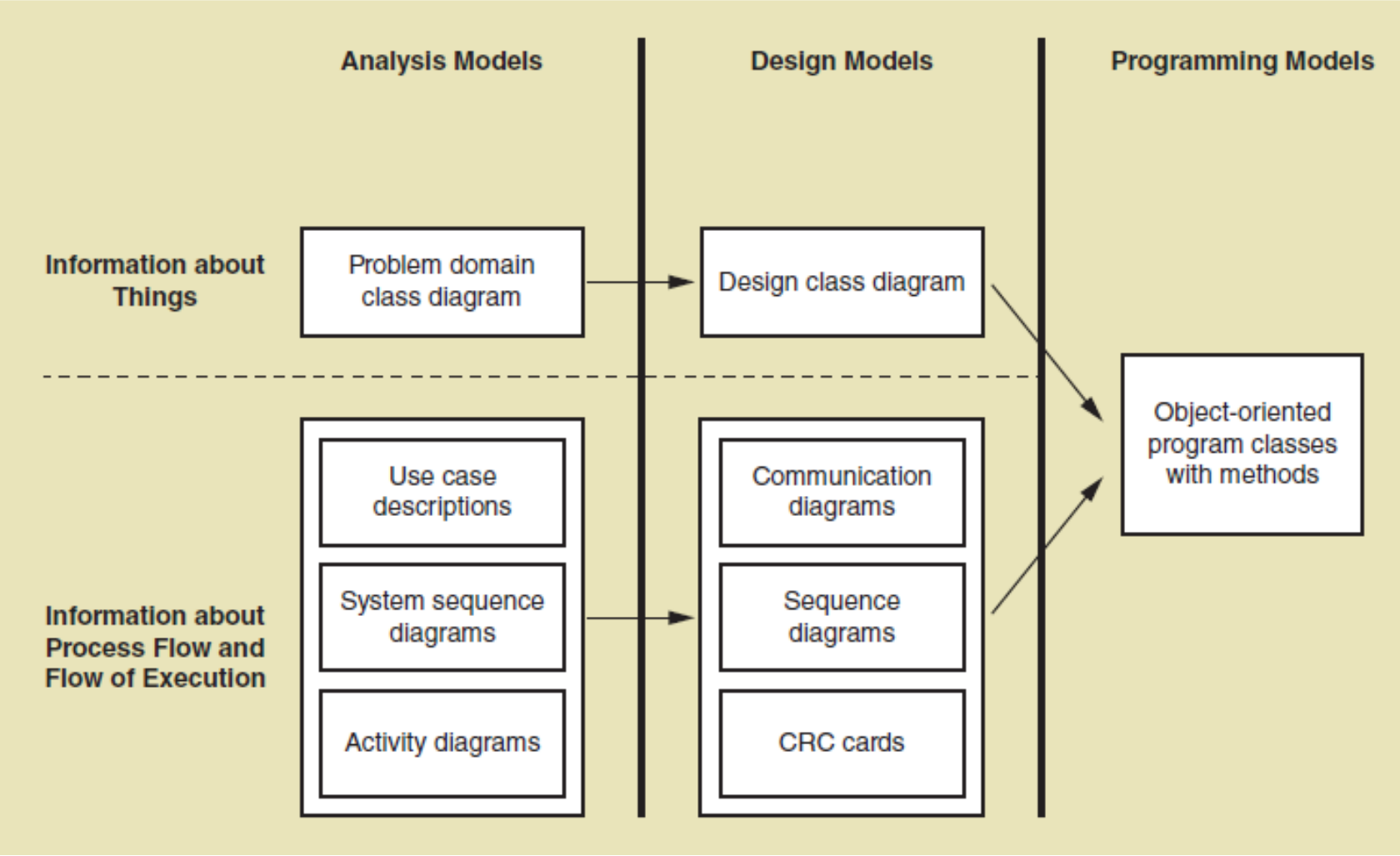
Lecture Material

- System Analysis and Design in a Changing World (Chapter 13)

Lecture Outline

- Interaction diagrams fundamentals
- Sequence diagram fundamentals
- SD design guidelines
- Example
- Collaboration Diagram

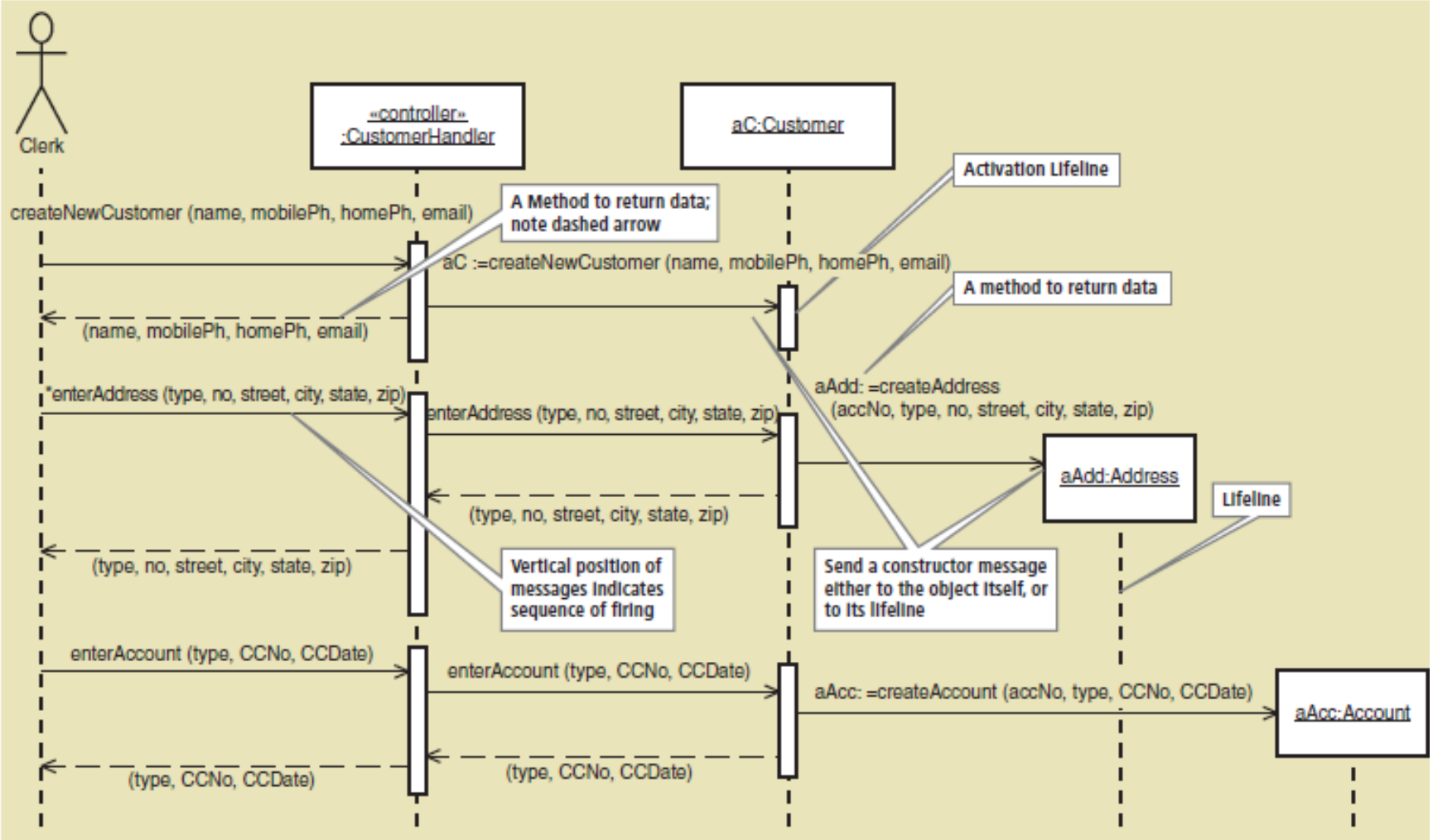
Class Diagram Fundamentals



Interaction Diagram Fundamentals

- Why we need interaction diagrams?
  - » Illustrate how user interacts with system.
  - » Illustrate how objects interacts with each other.
  - » Emphasizes time ordering of messages.
- There are two types of interaction diagrams:
  - » sequence diagram
  - » communication diagram (called collaboration diagram in UML 1.4).
- Can model simple sequential flow, branching, iteration (loop) , …etc
- The two diagrams are equivalent in terms of semantics

SD Fundamentals

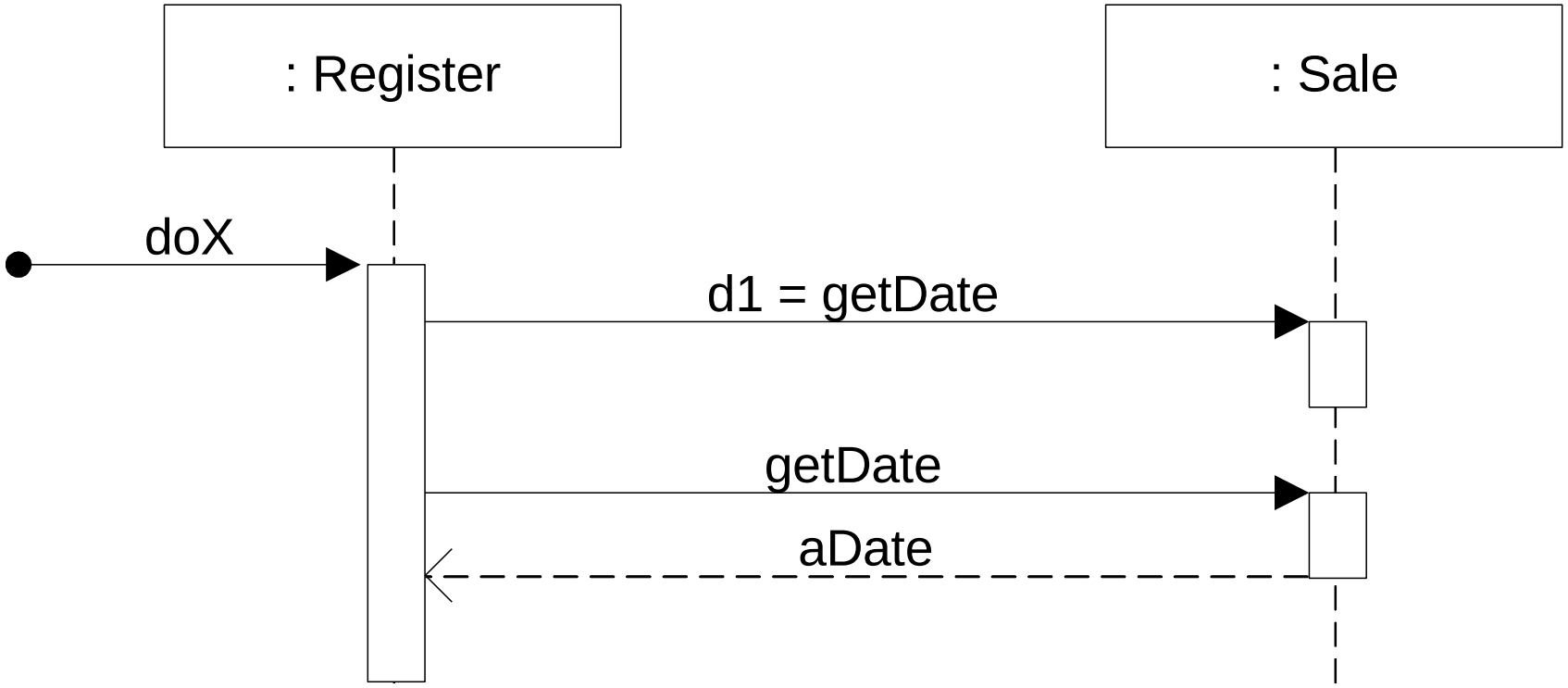


SD Fundamentals

Message	Description
⟶	**Synchronous:** A synchronous message between active objects indicates wait semantics; the sender waits for the message to be handled before it continues. This typically shows a method call.
→	**Asynchronous:** With an asynchronous flow of control, there is no explicit return message to the caller. An asynchronous message between objects indicates no-wait semantics; the sender does not wait for the message before it continues. This allows objects to execute concurrently.
<- - - - - - - - - - -	**Reply:** This shows the return message from another message.

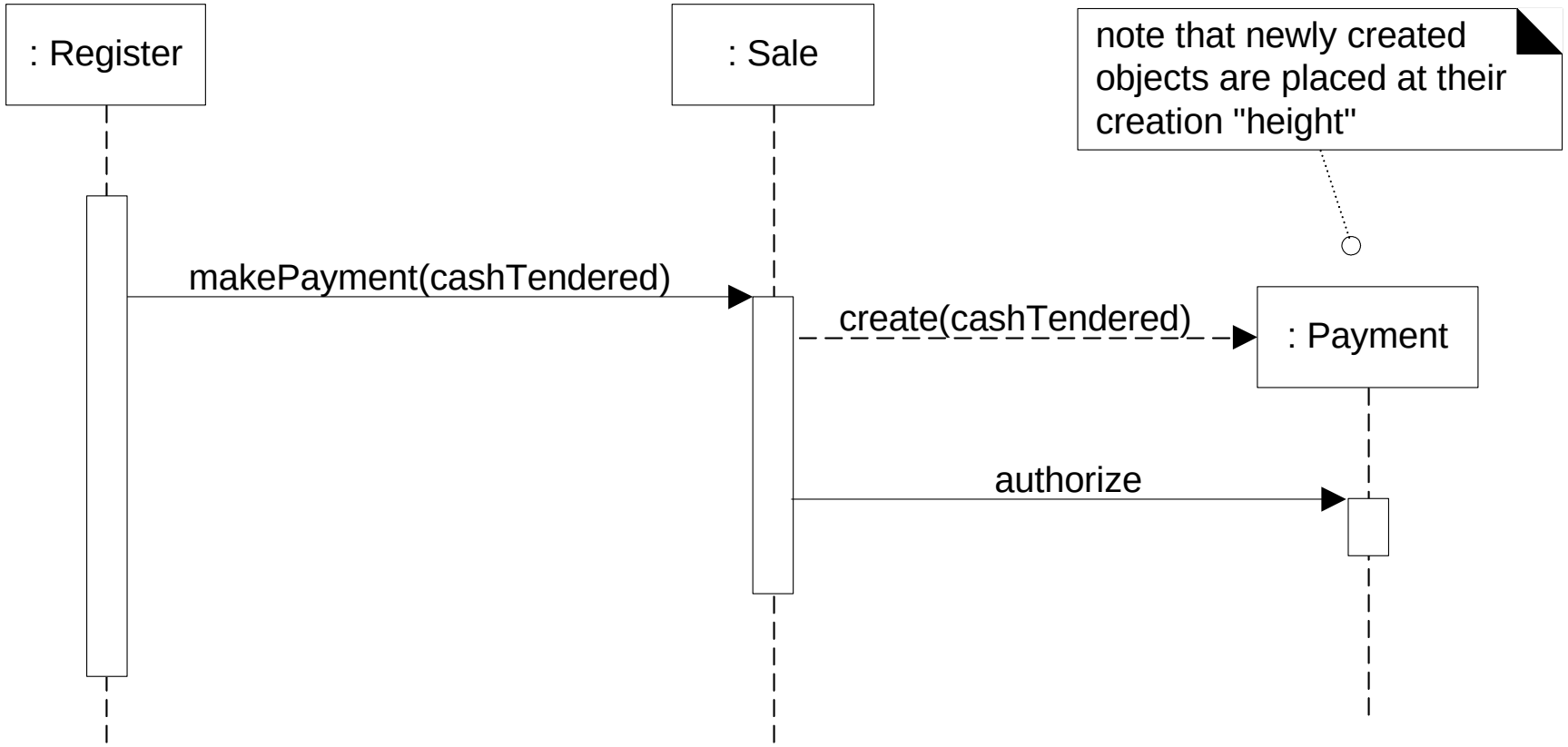
SD Fundamentals

- Return message



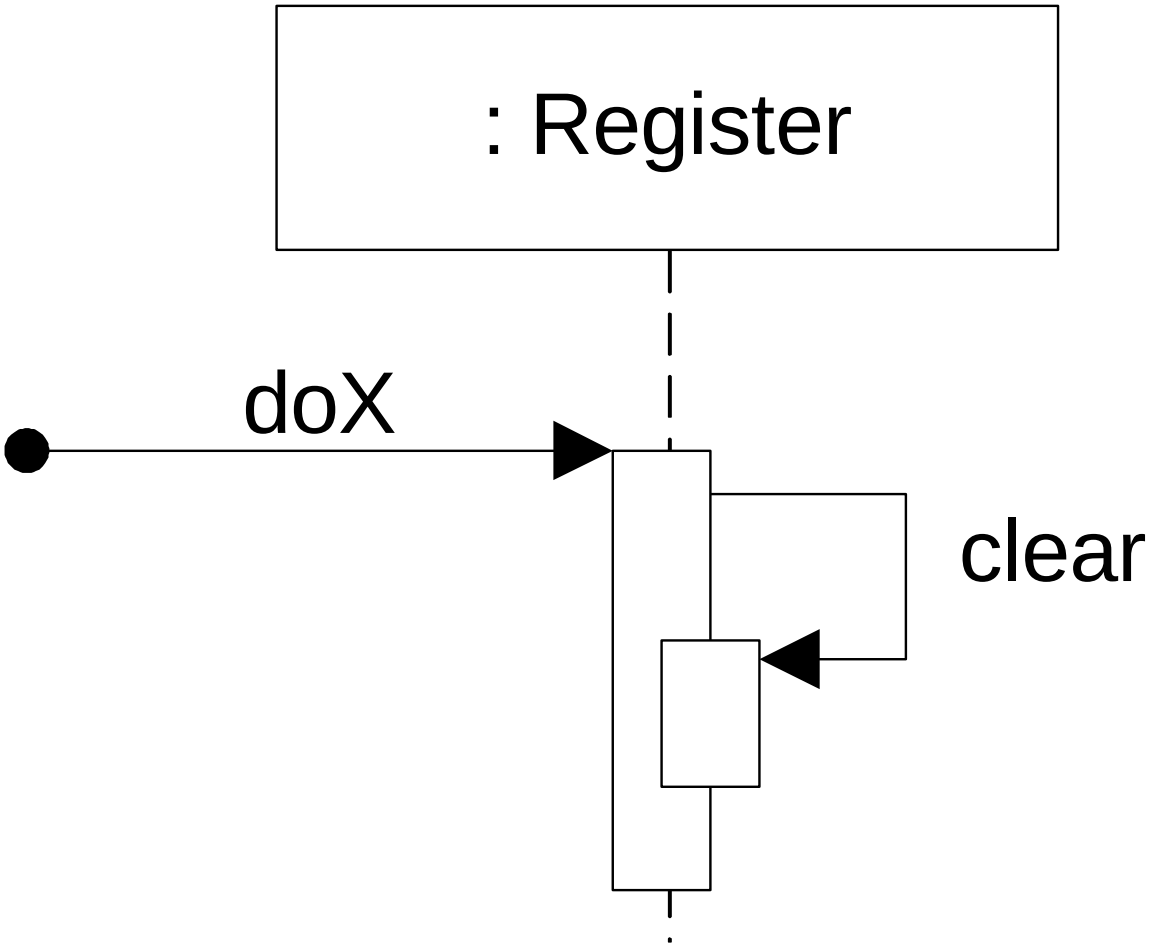
SD Fundamentals

- Creation of instance



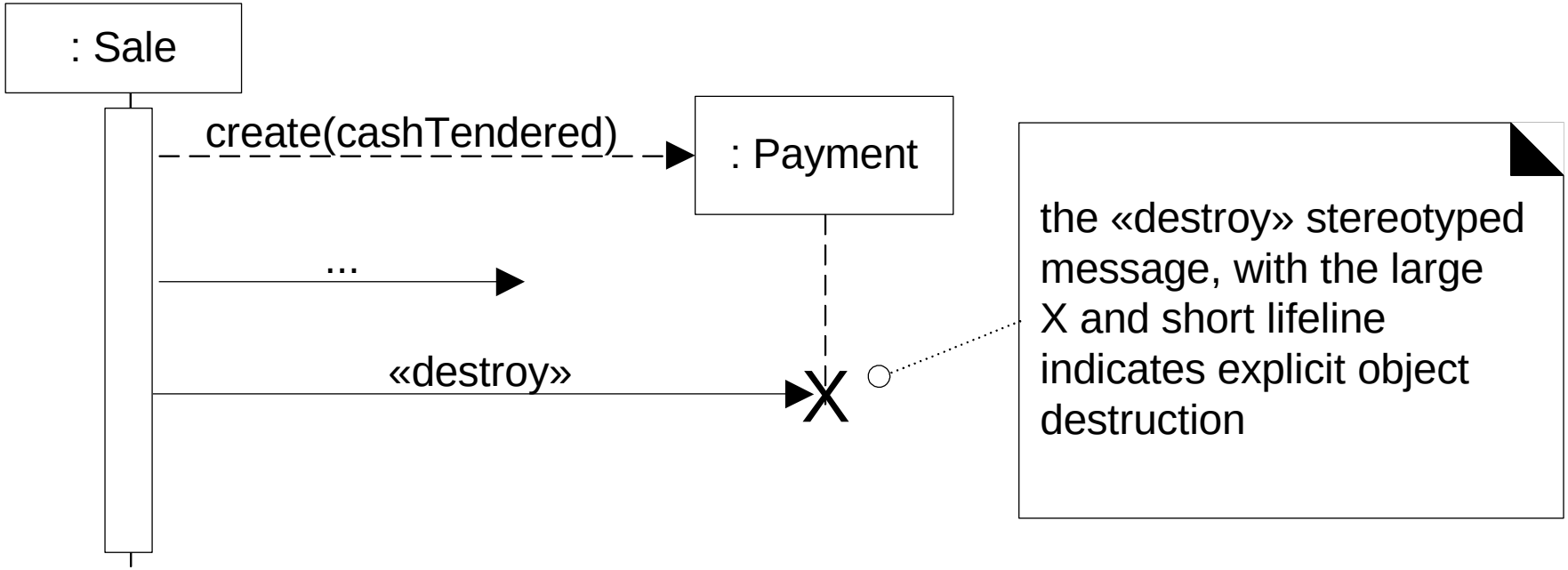
SD Fundamentals

- Self message



SD Fundamentals

- Destroy objects



- Use only if you need to show the destruction of an object explicitly (e.g., C++)

SD Fundamentals

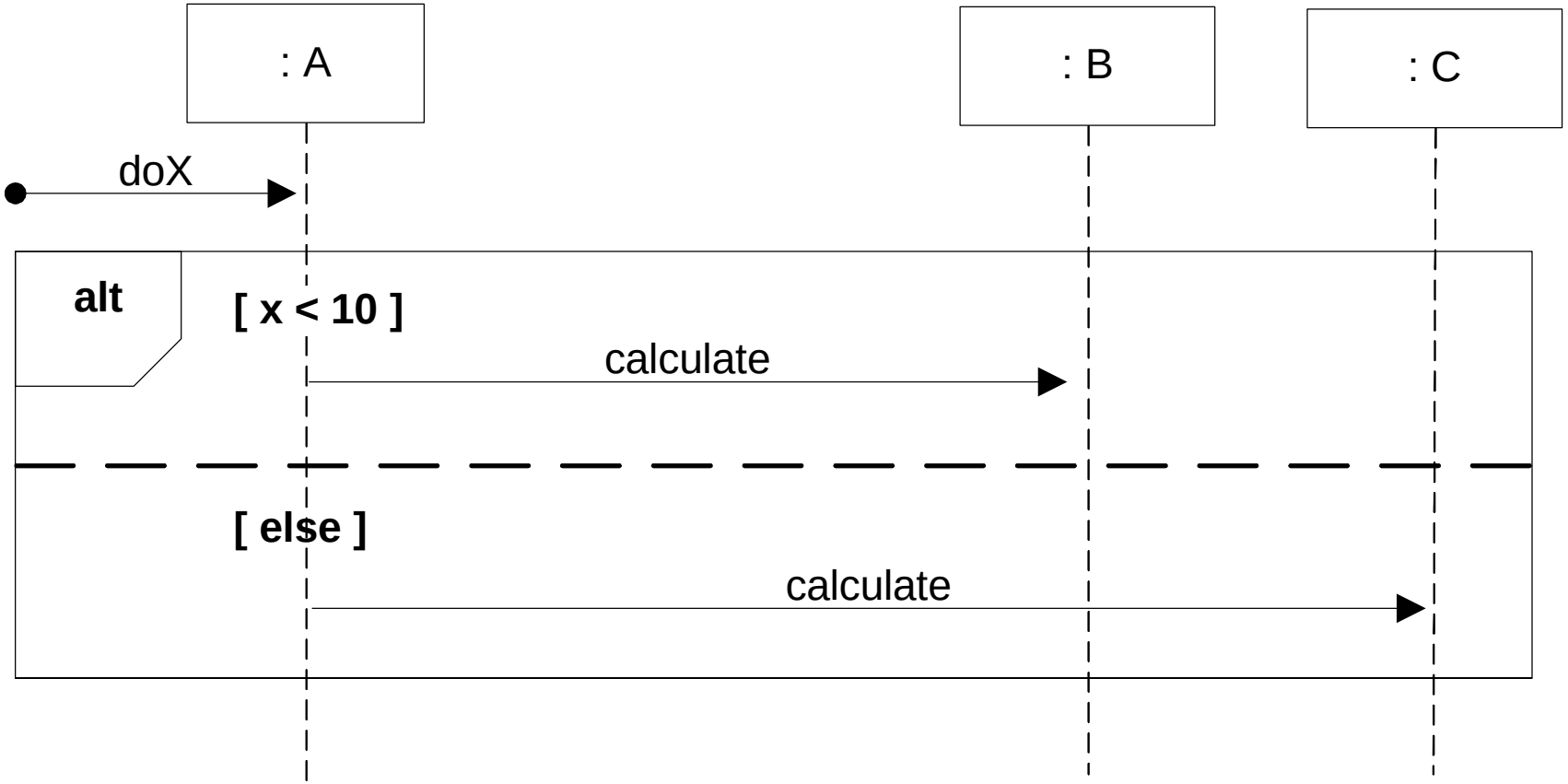
- Advanced constructs

The following table summarizes some common frame operators:

Frame Operator	Meaning
alt	Alternative fragment for mutual exclusion conditional logic expressed in the guards.
loop	Loop fragment while guard is true. Can also write *loop(n)* to indicate looping n times. There is discussion that the specification will be enhanced to define a *FOR* loop, such as *loop(i, 1, 10)*
opt	Optional fragment that executes if guard is true.
par	Parallel fragments that execute in parallel.
region	Critical region within which only one thread can run.

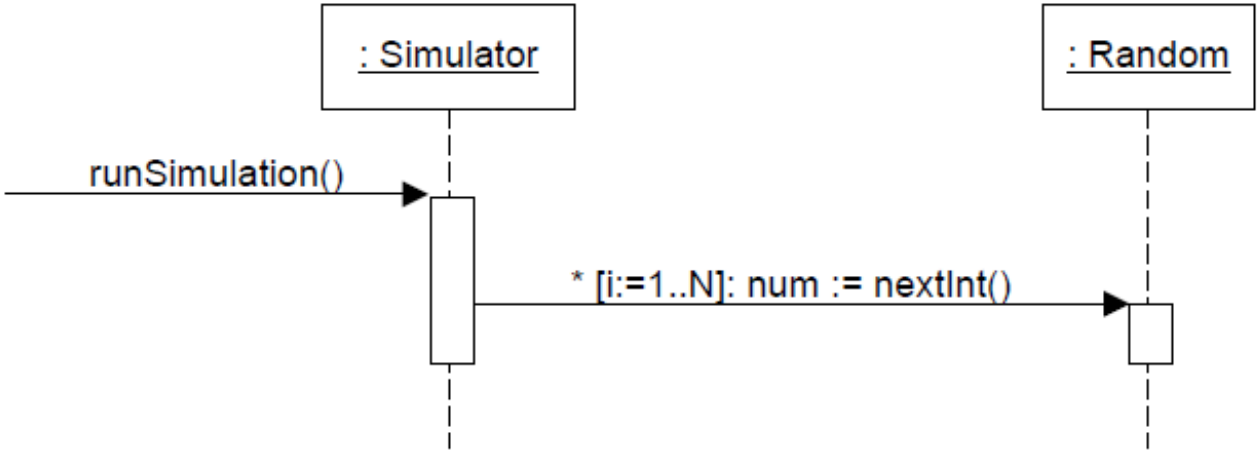
SD Fundamentals

- Alter frame (mutually exclusive)



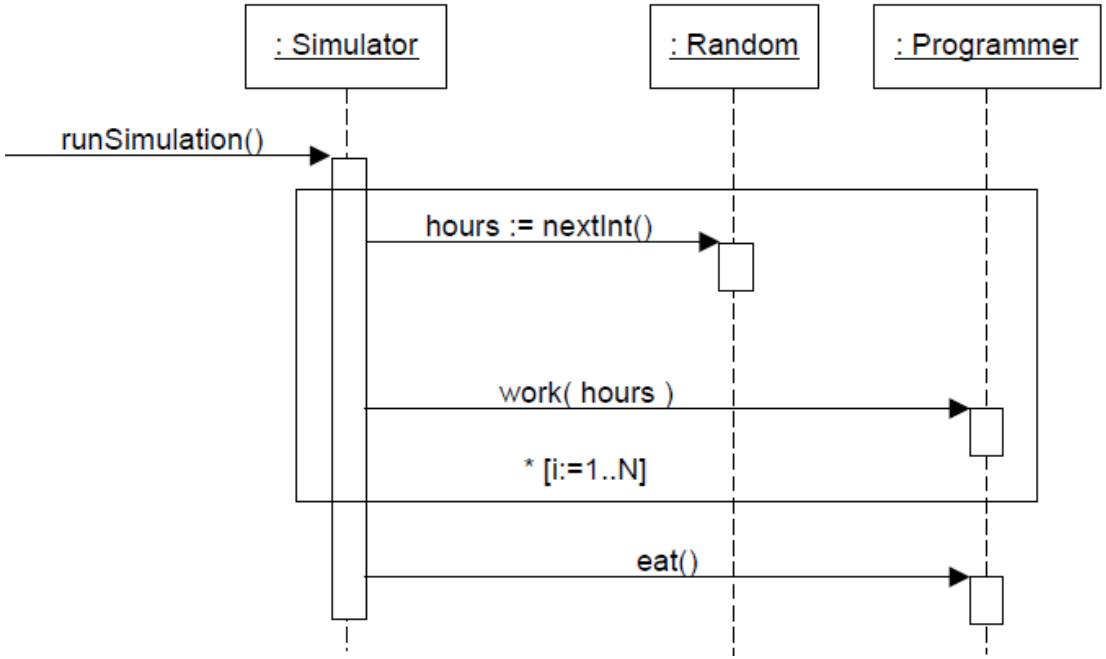
SD Fundamentals

- Iteration frames
  - » Iteration over a single message.
  - » Used when you have to perform one task for known multiple time.



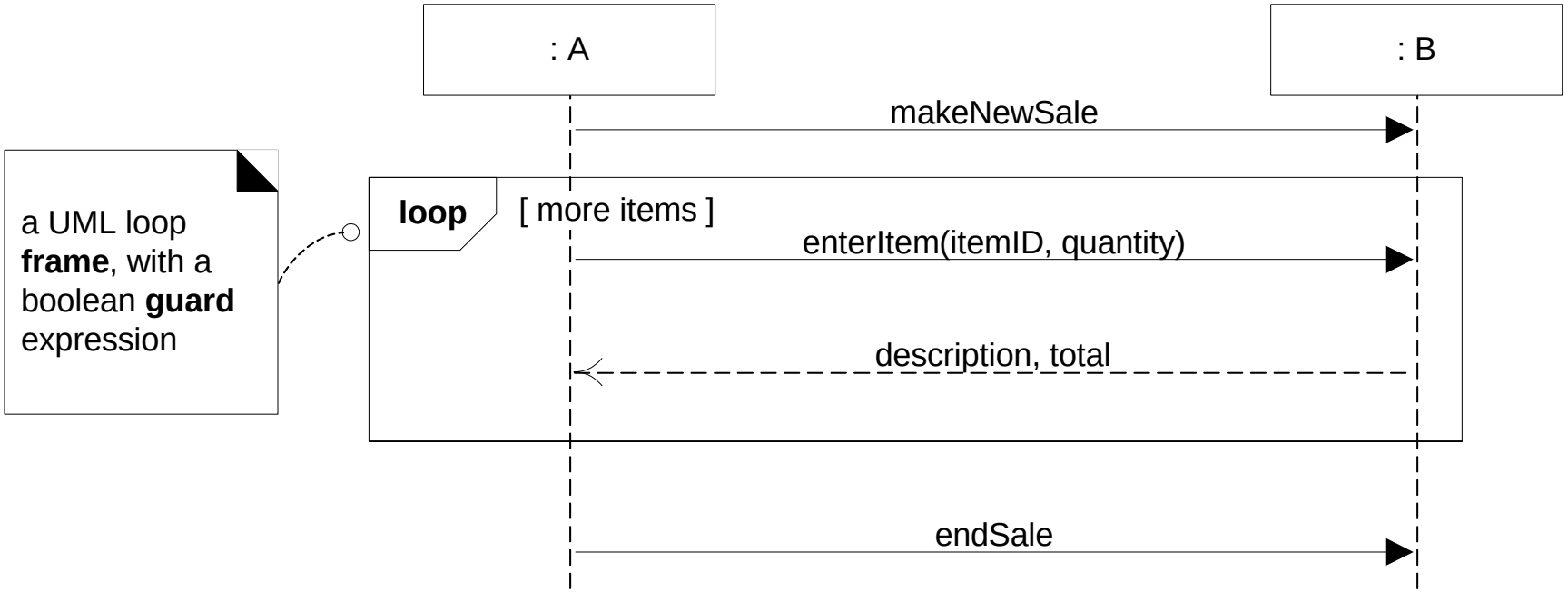
SD Fundamentals

- Iteration frames
  - » Iteration over a series of messages.
  - » Used when you have to perform more than one task for known multiple time.



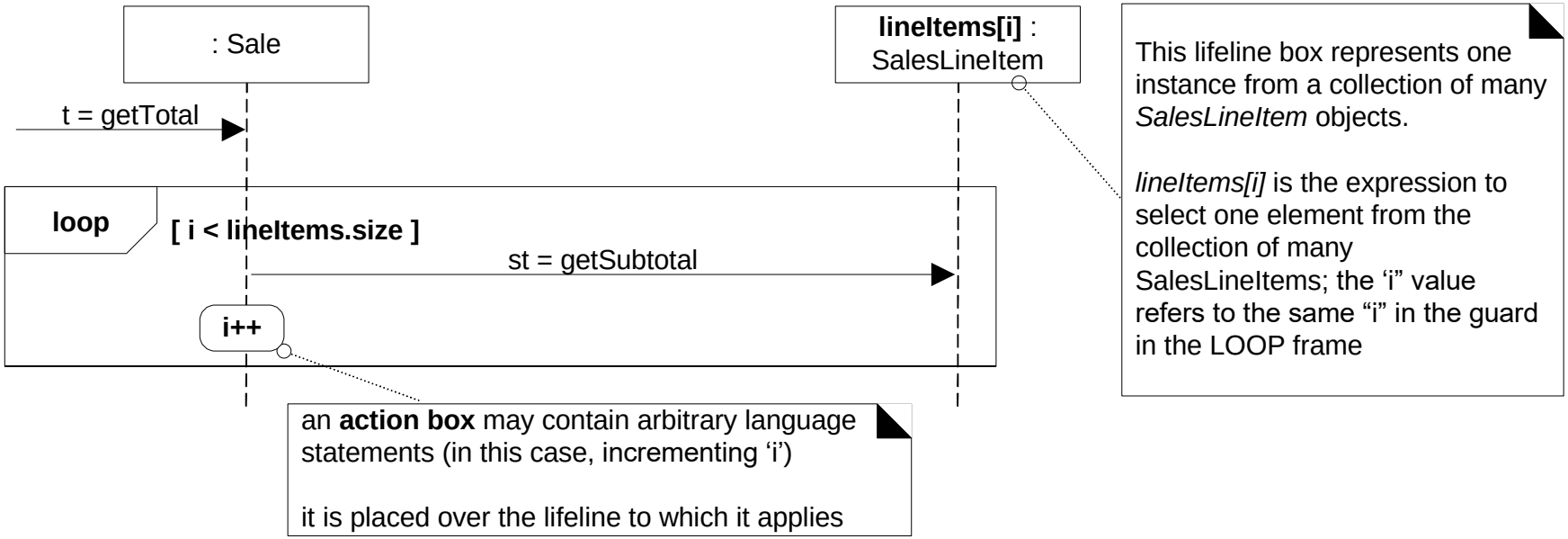
SD Fundamentals

- Iteration frames
  - » Logical Iteration.
  - » Can be used as an iteration construct.



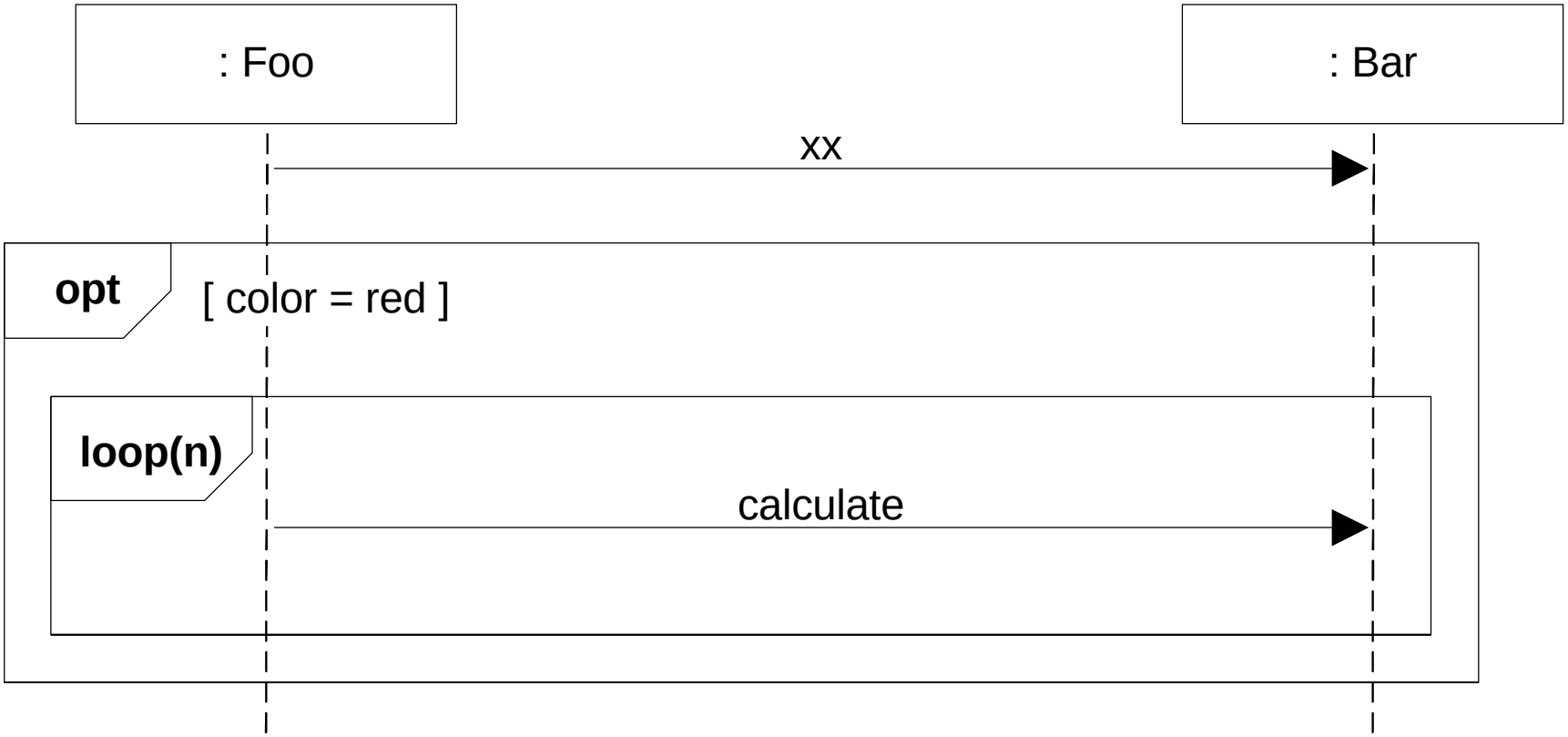
SD Fundamentals

- Iteration frames
  - » Iteration over a list items



SD Fundamentals

- Nesting of frames



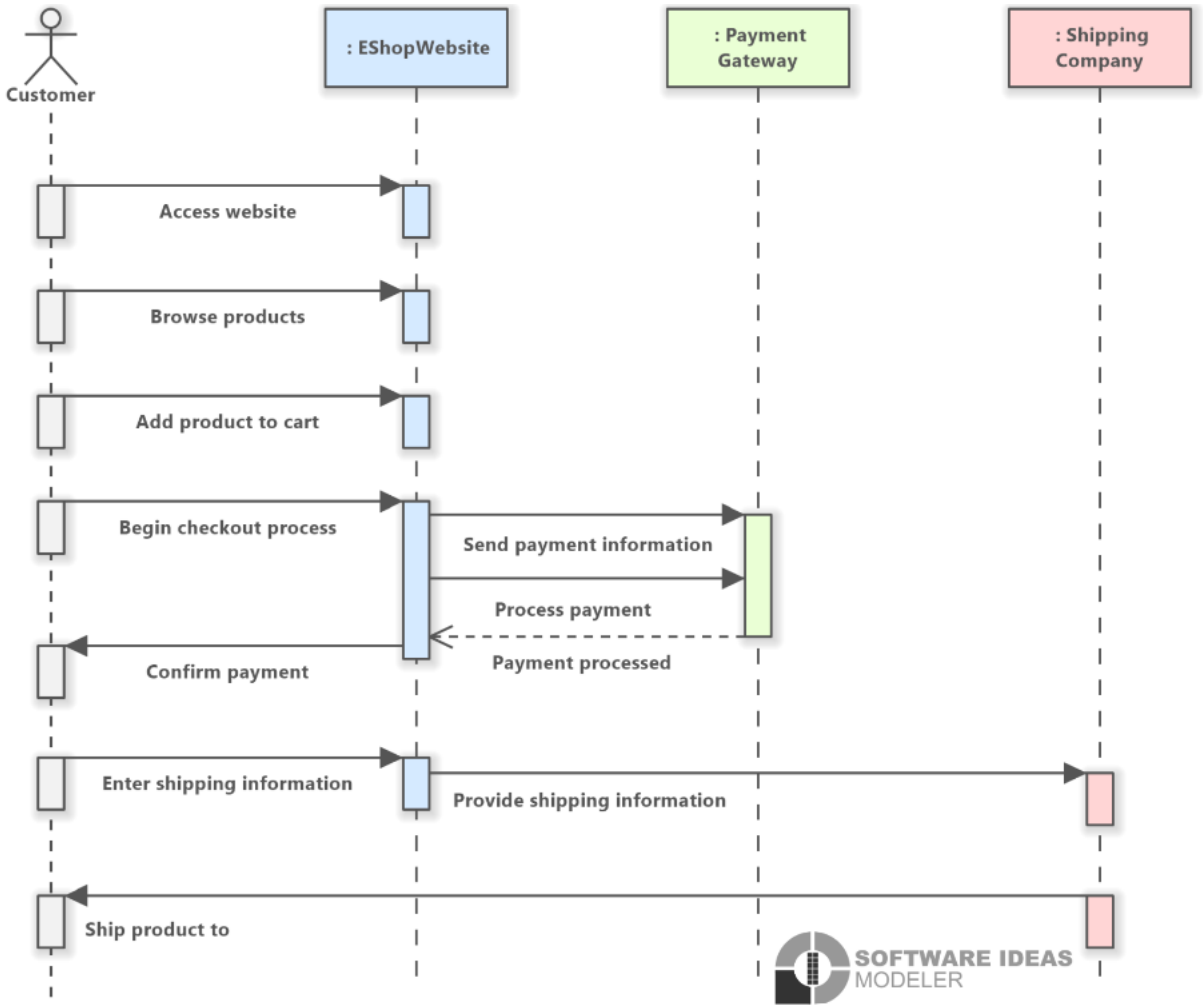
SD Example

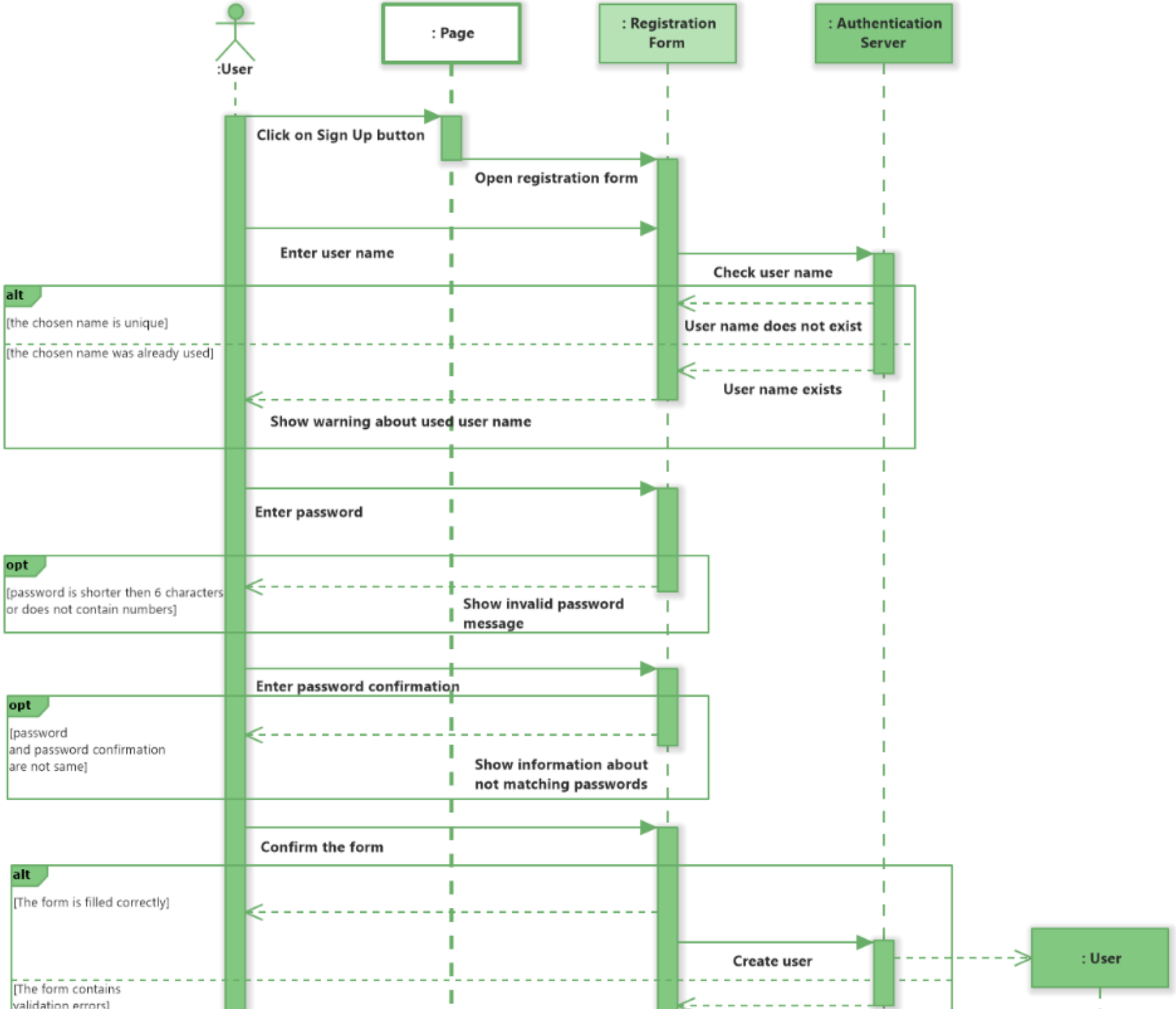
- The Customer initiates the process by accessing the E-shop website and browsing the available products. Once they have found a product they wish to purchase, they add it to their cart and begin the checkout process.

- During the checkout process, the E-shop website retrieves the Customer's payment information and sends it to the Payment gateway to be processed. Once the Payment gateway has confirmed the payment, the E-shop website sends a confirmation to the Customer.

- The E-shop website then provides the necessary shipping information to the Shipping company, which ships the product to the Customer.

SD Example


Customer
: EShopWebsite
: Payment Gateway
: Shipping Company
Access website
Browse products
Add product to cart
Begin checkout process
Send payment information
Process payment
Payment processed
Confirm payment
Enter shipping information
Provide shipping information
Ship product to
SOFTWARE IDEAS MODELER

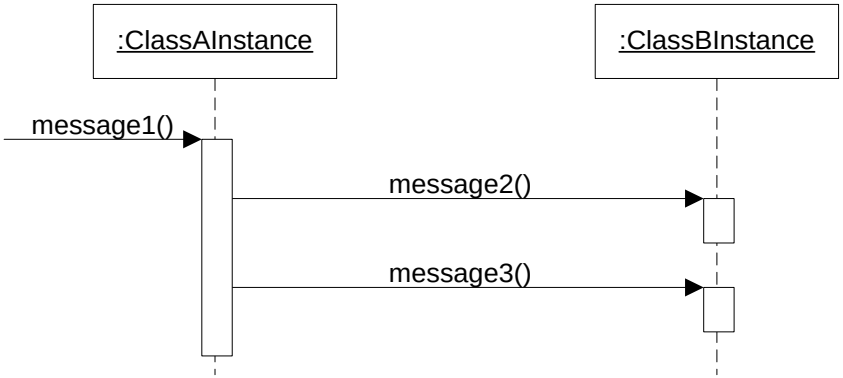

:User
: Page
: Registration Form
: Authentication Server
Click on Sign Up button
Open registration form
Enter user name
Check user name
alt
[the chosen name is unique]
User name does not exist
[the chosen name was already used]
User name exists
Show warning about used user name
Enter password
opt
[password is shorter then 6 characters or does not contain numbers]
Show invalid password message
Enter password confirmation
opt
[password and password confirmation are not same]
Show information about not matching passwords
Confirm the form
alt
[The form is filled correctly]
Create user
: User
[The form contains validation errors]

Collaboration/Communication Diagram

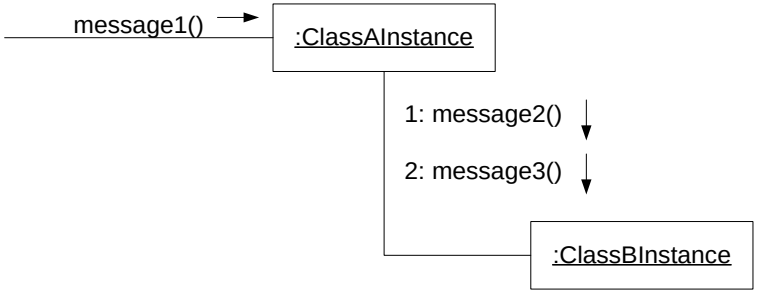
- A Collaboration is a collection of named objects and actors with links connecting them.
- A Collaboration between objects working together provides emergent desirable functionalities in Object-Oriented systems.
- Objects collaborate by communicating (passing messages) with one another in order to work together

Interaction Diagram Fundamentals

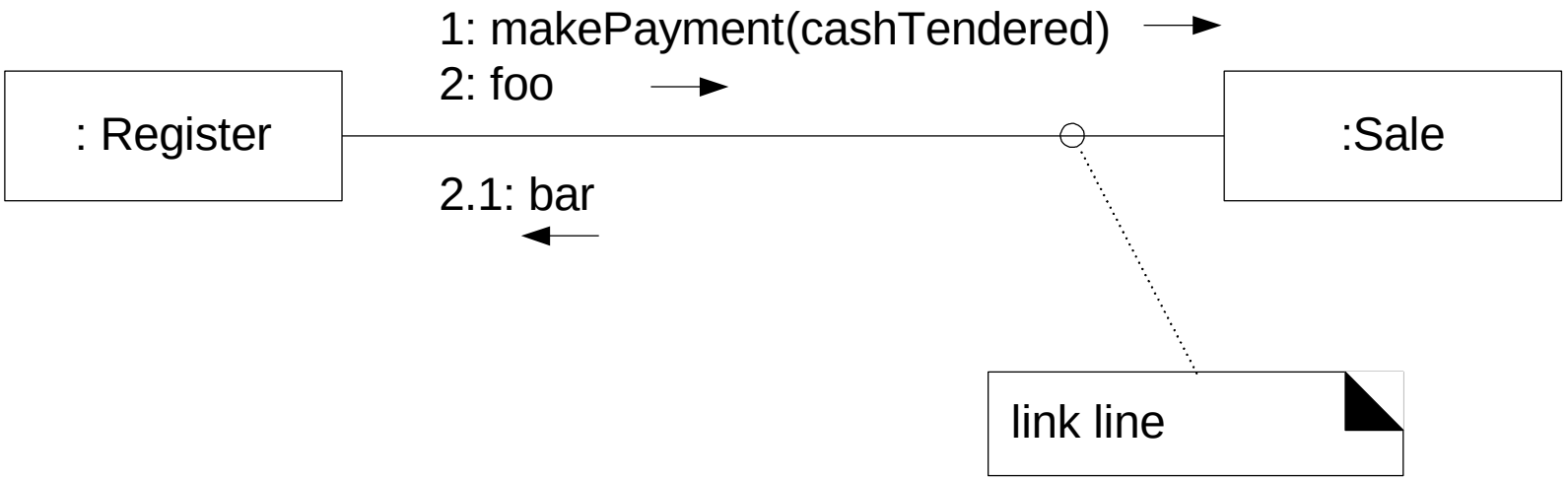
- Sequence diagram



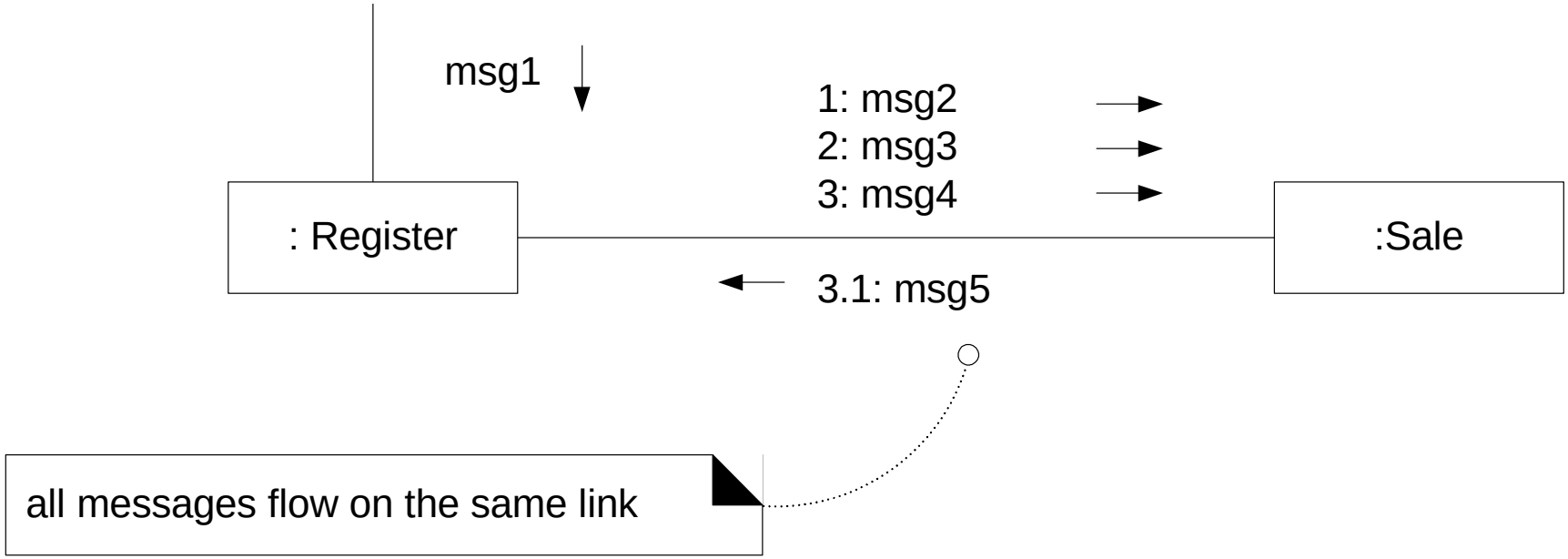
- Communication diagram



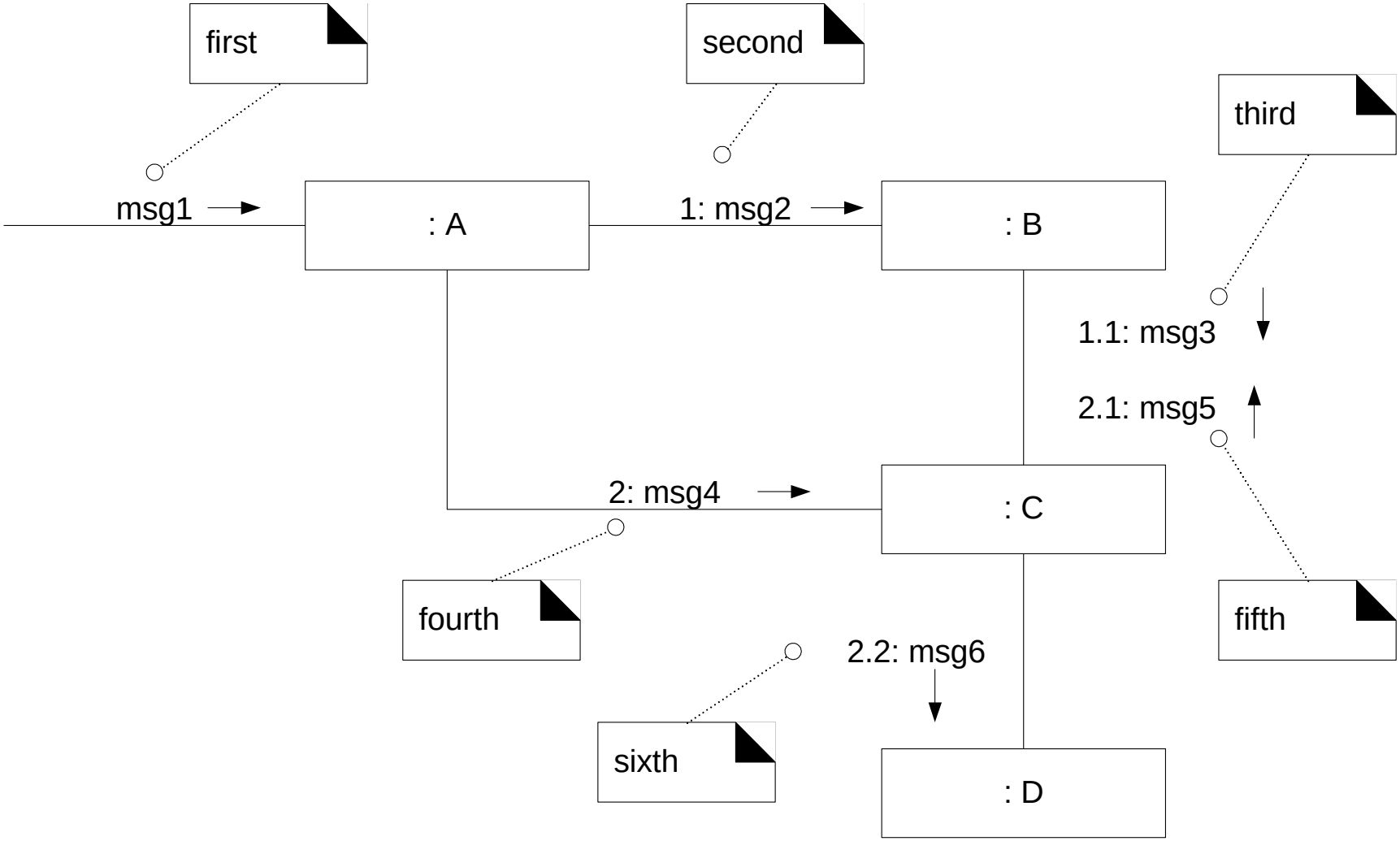
Collaboration/Communication Diagram



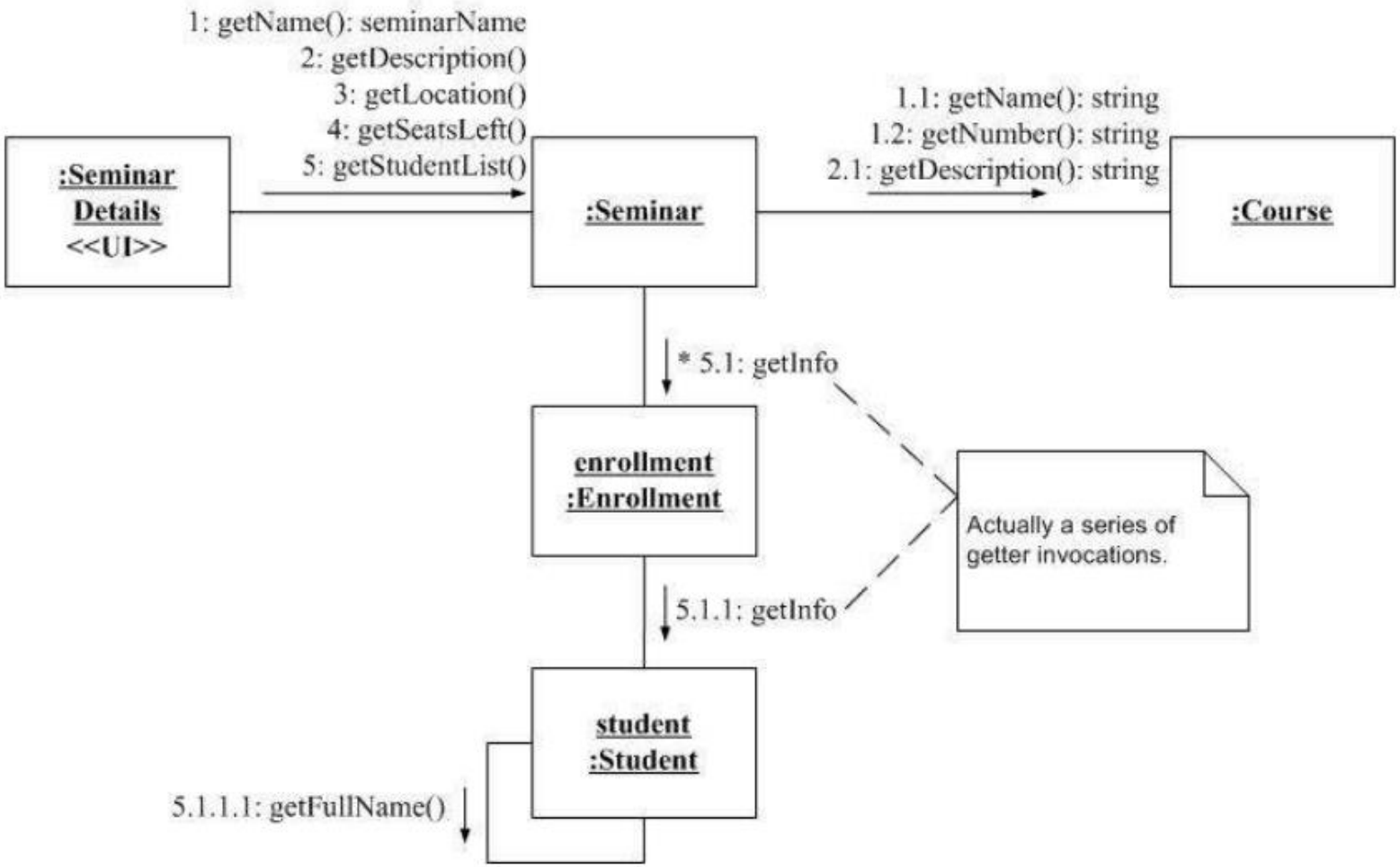
Collaboration/Communication Diagram



Collaboration/Communication Diagram – Message numbering

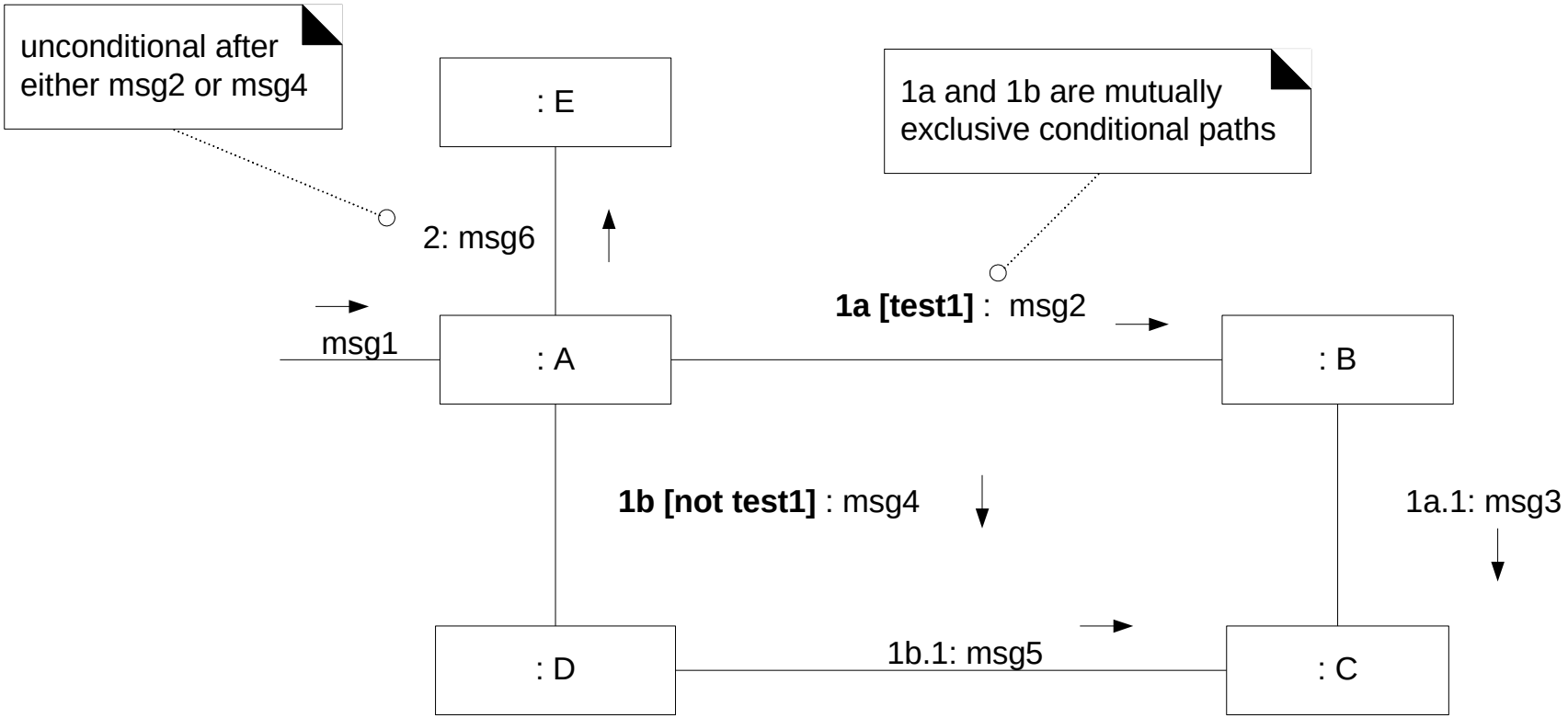


Collaboration/Communication Diagram



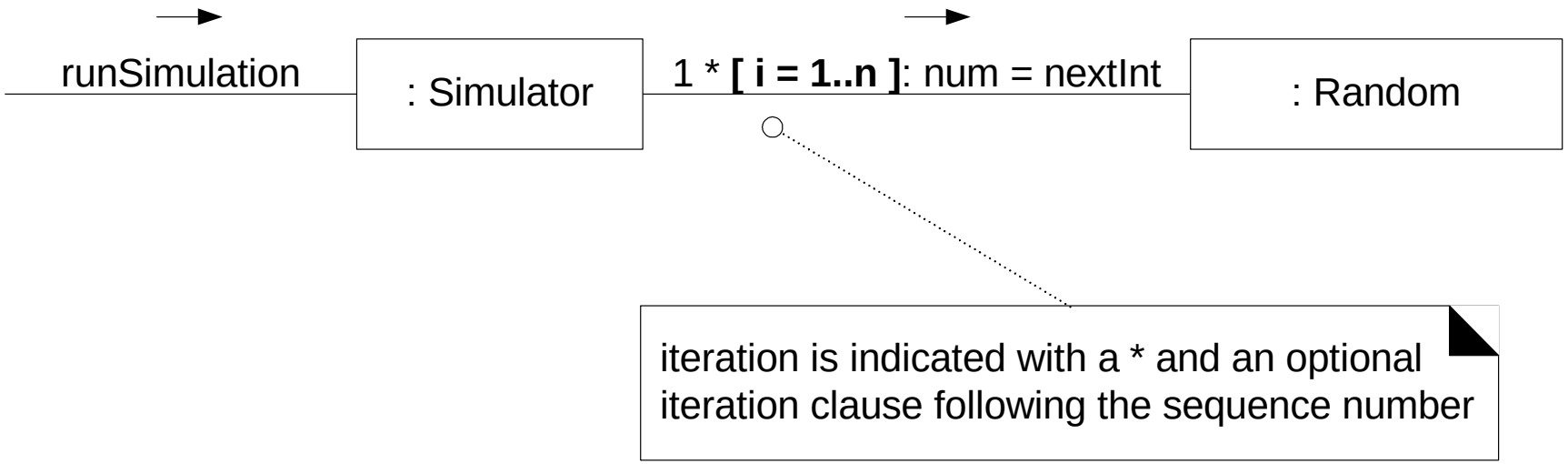
Collaboration/Communication Diagram

- Mutually Exclusive Conditional Paths

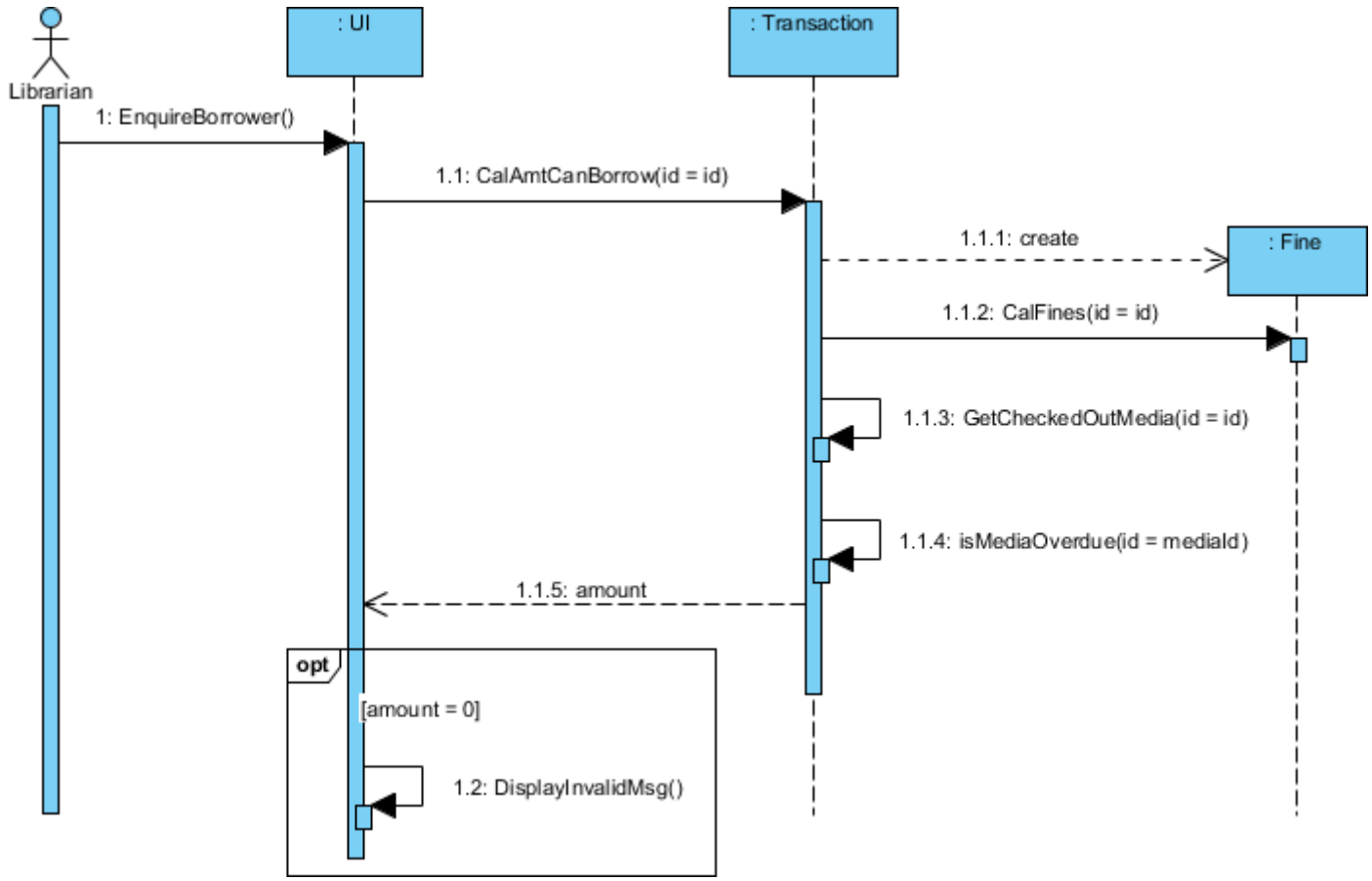


Collaboration/Communication Diagram

- Iterations



Collaboration/Communication Diagram -- Example



Collaboration/Communication Diagram -- Example