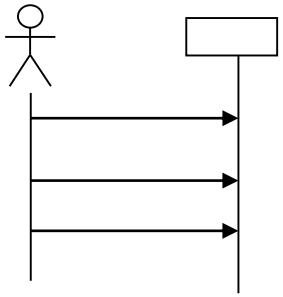
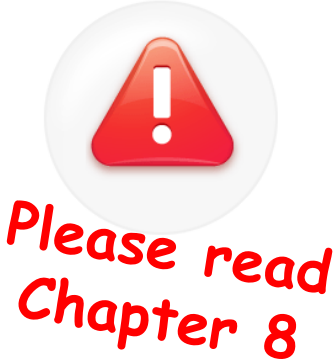
CMPS 411

Interaction Modeling





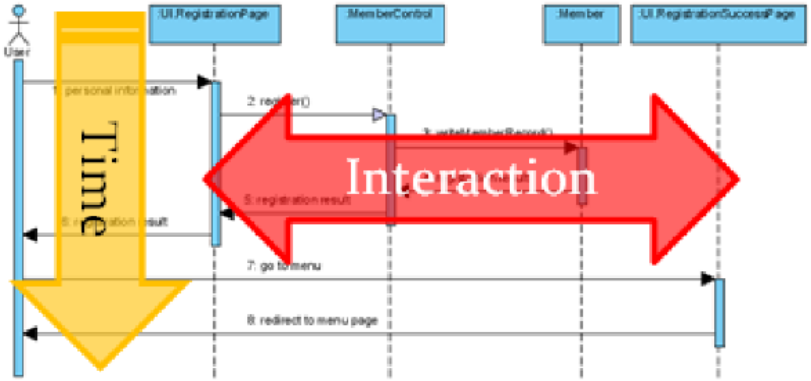
Dr. Abdelkarim Erradi

Dept. of Computer Science & Engineering

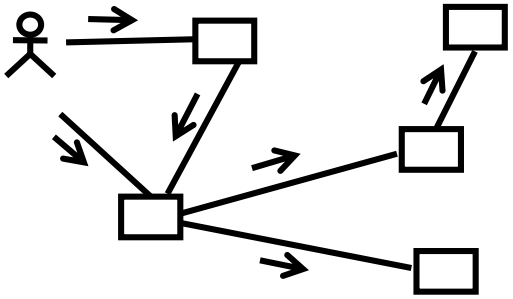
QU

Outline

- Sequence Diagram



- Communication Diagram



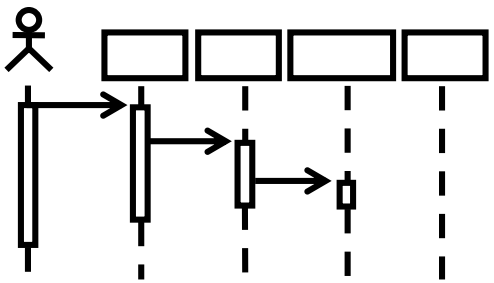
Interaction Diagrams

- Interaction diagrams are used to model the **dynamic aspects** of a system
- Dynamic aspects of the system
  - Messages moving among objects
  - Flow of control among objects
  - Sequences of events

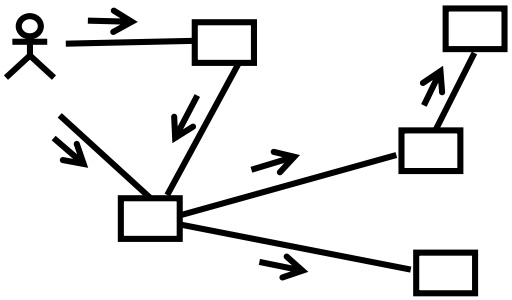
- The main UML diagram to model interactions is the Sequence Diagram

Interaction Diagrams

- Design Sequence Diagram
  - **Time oriented view** emphasize the time ordering of the interactions. The diagram shows:
    - The **objects participating** in the interaction.
    - The **sequence of messages exchanged**.
- Communication Diagram
  - **Shows how the objects related to each other**
  - Emphasize the **structural organization** of the objects participating in interactions:
    - The objects participating in the interaction.
    - **Links between the objects**.
    - Messages passed between the objects.



Sequence Diagrams



Communication Diagrams

System-Level Sequence Diagrams

Sequence Diagrams

- Visualize the set of messages exchanged between the actor and the system to perform the steps of a use case

- **Time oriented view** emphasize the time ordering of the interactions.

Emphasis on time ordering!

System Sequence Diagrams

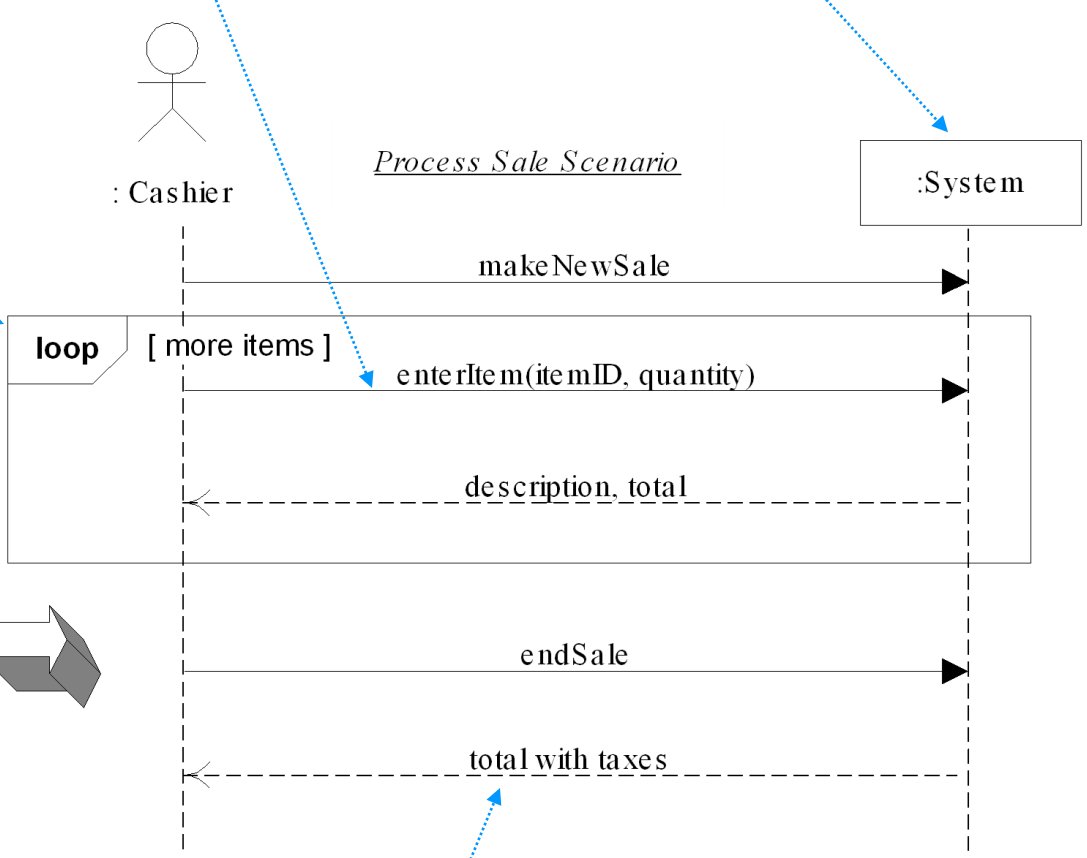
- Use case scenarios describe how external actors interact with the system…
  - The actor generates <u>system events</u> to a system, requesting some <u>system operation</u> to handle the event.
- A <u>System Sequence Diagram</u> (SSD) is a diagram that shows, *for one particular scenario of a use case*:
  - the **events that the actor generates**,
  - **their order**
  - the **system response** to such events.
- The **system is regarded as a black box** and the functionalities are expressed from a user's perspective

SSD Example – Process Sale Scenario

Message with parameters to pass data to the system

System as a black box

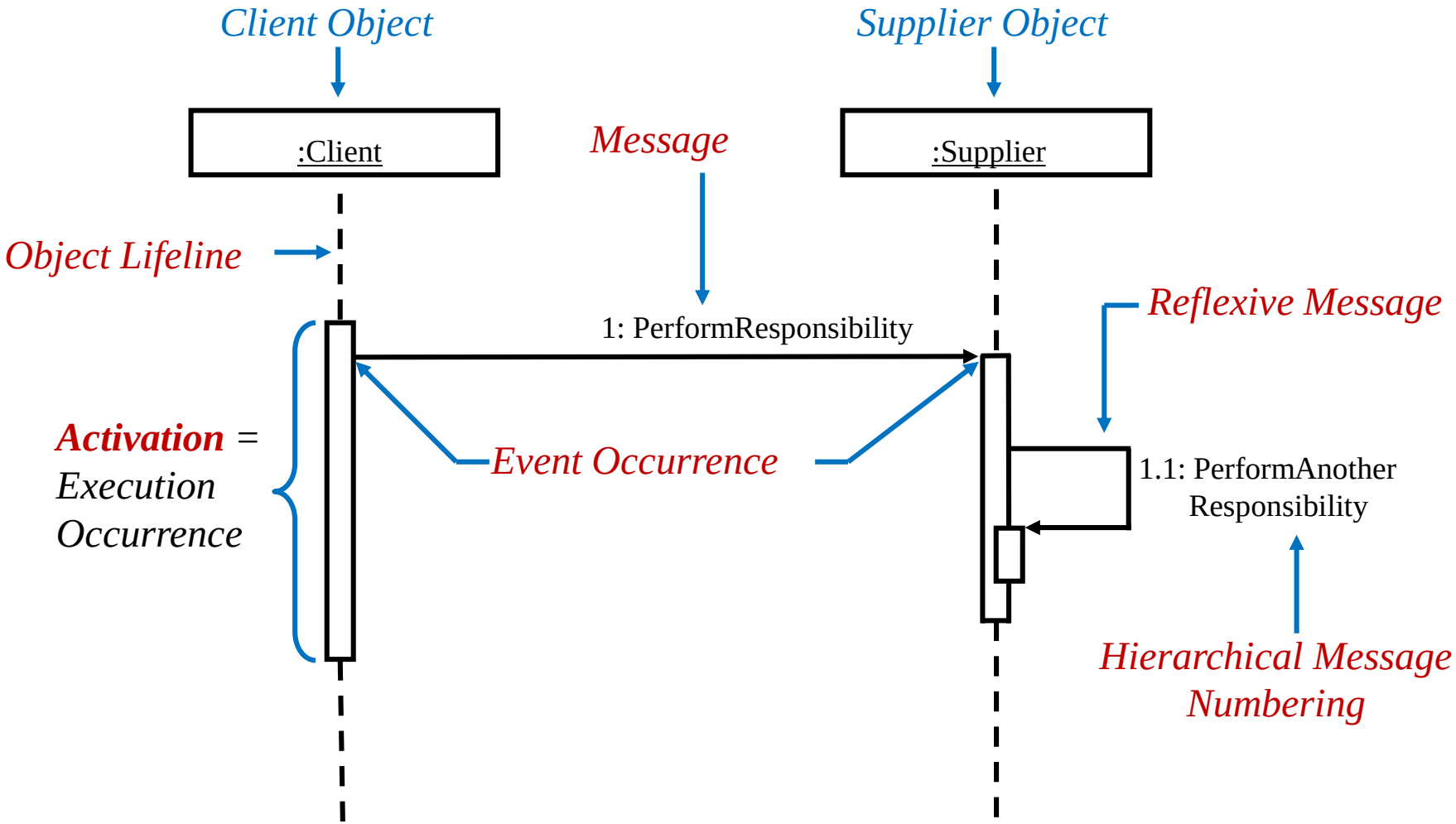
A loop indicates any recurring events



Actor (cashier)	System
Cashier starts a new sale	Displays a transaction entry area.
Cashier enters item identifier	System creates a line item record and retrieves and presents item description (including price) and running total for the transaction.
The above step is repeated until the cashier signals that the transaction is complete.	The system calculates total with taxes and presents the results.

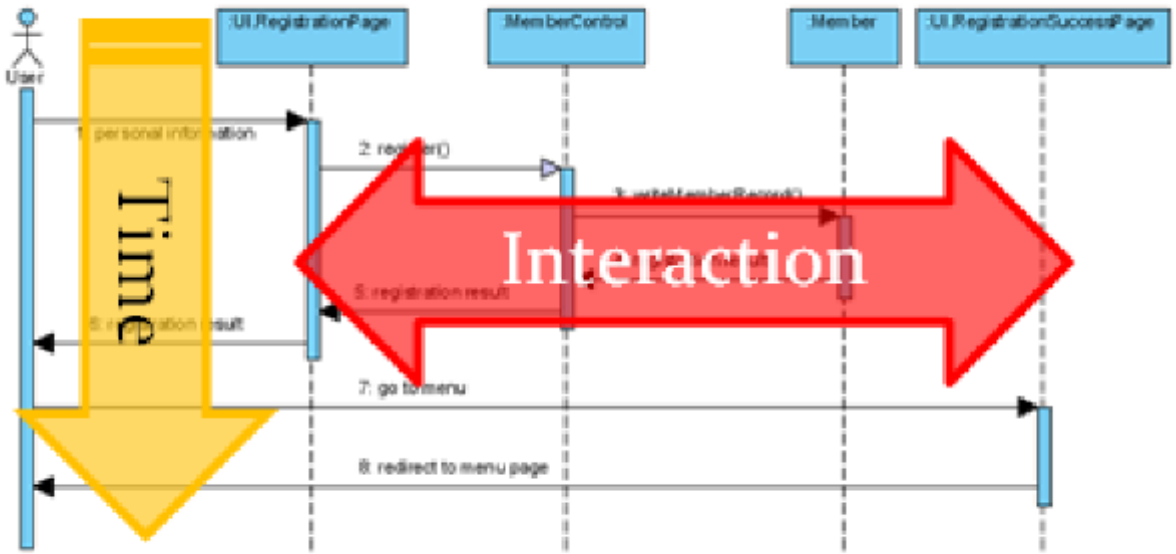
Return Value associated with previous message; The return line is optional if nothing is returned

The Anatomy of Sequence Diagrams

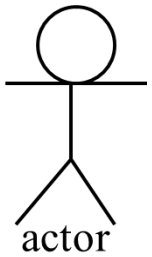


The Anatomy of Sequence Diagrams (cont.)

- SD Visualize interactions between objects
  - Interactions shown in horizontal direction
  - Time in shown in vertical direction
- Activation is a rectangle on the lifeline, it indicate the time where execution takes place on that object.



SSD Elements



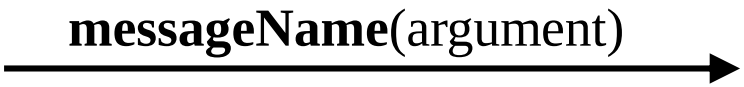
- **Actor:** An Actor is modeled using the ubiquitous symbol, the stick figure.

- **Lifeline:** The Lifeline identifies the existence of the object over time. The notation for a Lifeline is a vertical dotted line extending from an object.

- **System** behaves as "Black Box".

  – **In the design phase we will simply opened up the black box** to show details of the object interactions to handle the actor requests

- **Message:** Messages, modeled as horizontal arrows indicating the interactions between the actor and the system.



Message Types

- Message, often a method call, carry information from the actor to the system (and vise versa)
  - A message is numbered and labelled and can have an argument list and a return value.

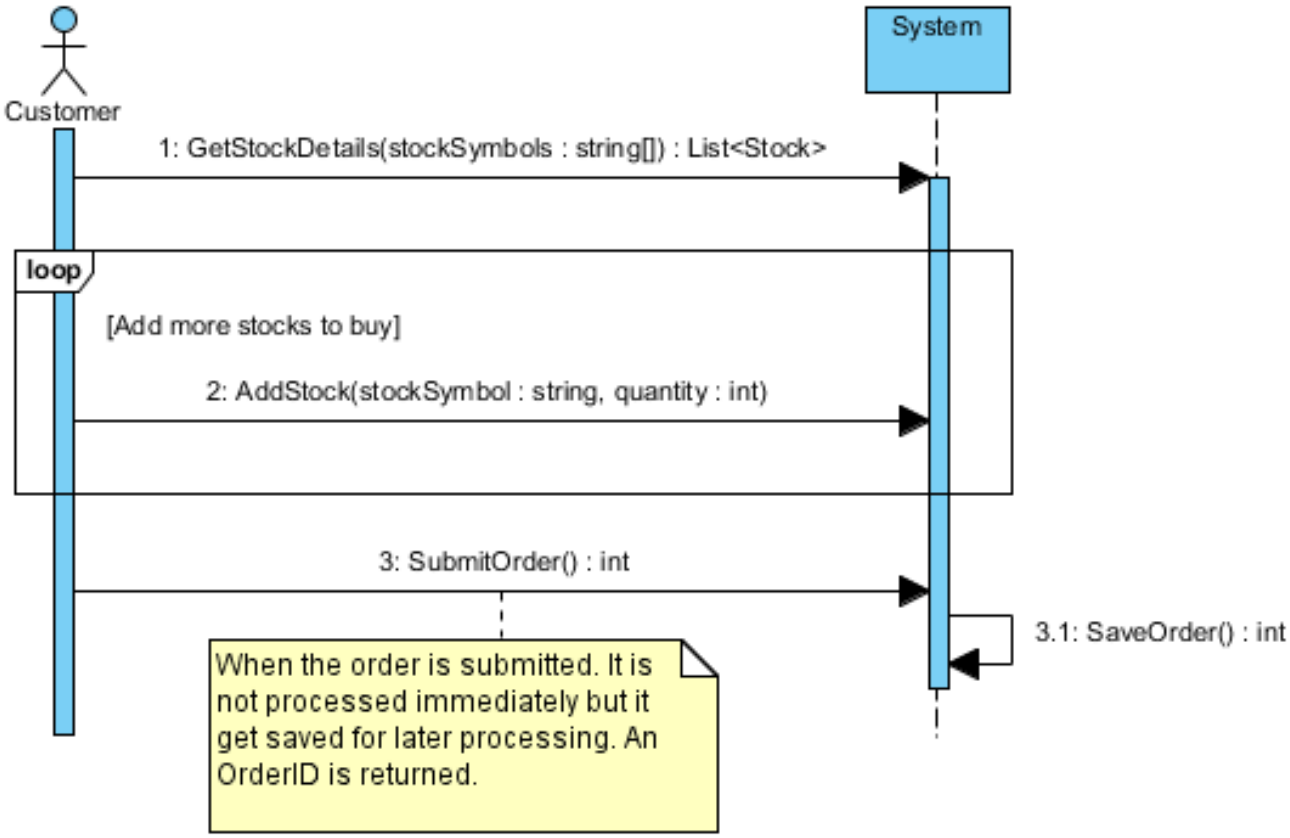
=> Ignore the UI as they are often designed separately. UI is a simply the visual representation of the exchanged objects

Message Type	Notation
Asynchronous	⟶
Synchronous	⟶
Return	⟵ - - - - - - - - - - - - - - - -

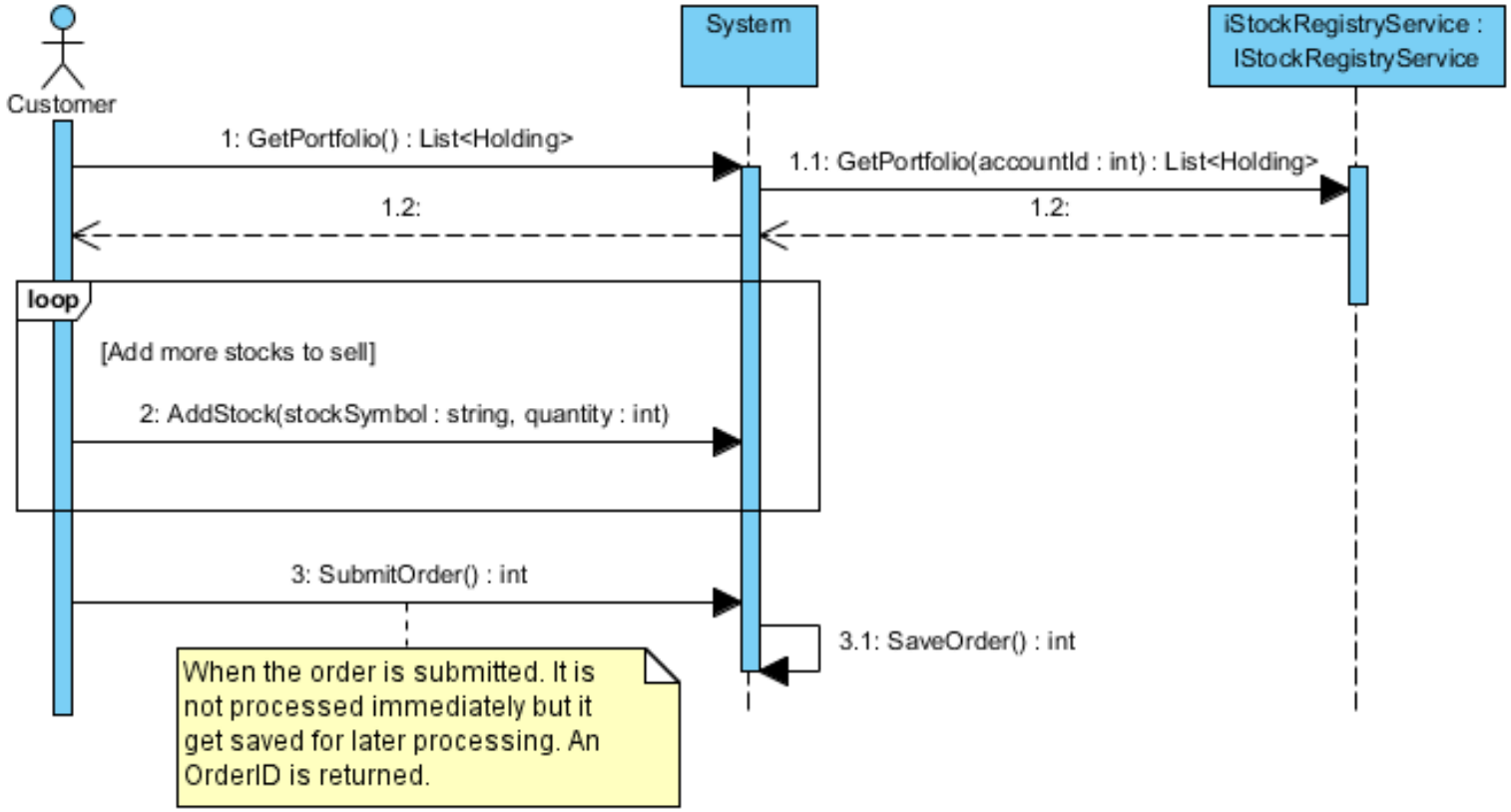
Drawing System Sequence Diagrams

- Select a Use Case scenario to model
  - One diagram per scenario
- Add lifelines for the System and each involved Actor
- Add interactions between the Actors and the System using arrows
- Name the arrows using message names and parameters
  - Keep it simple, no need to provide all parameters at this point

Buy Stocks SSD



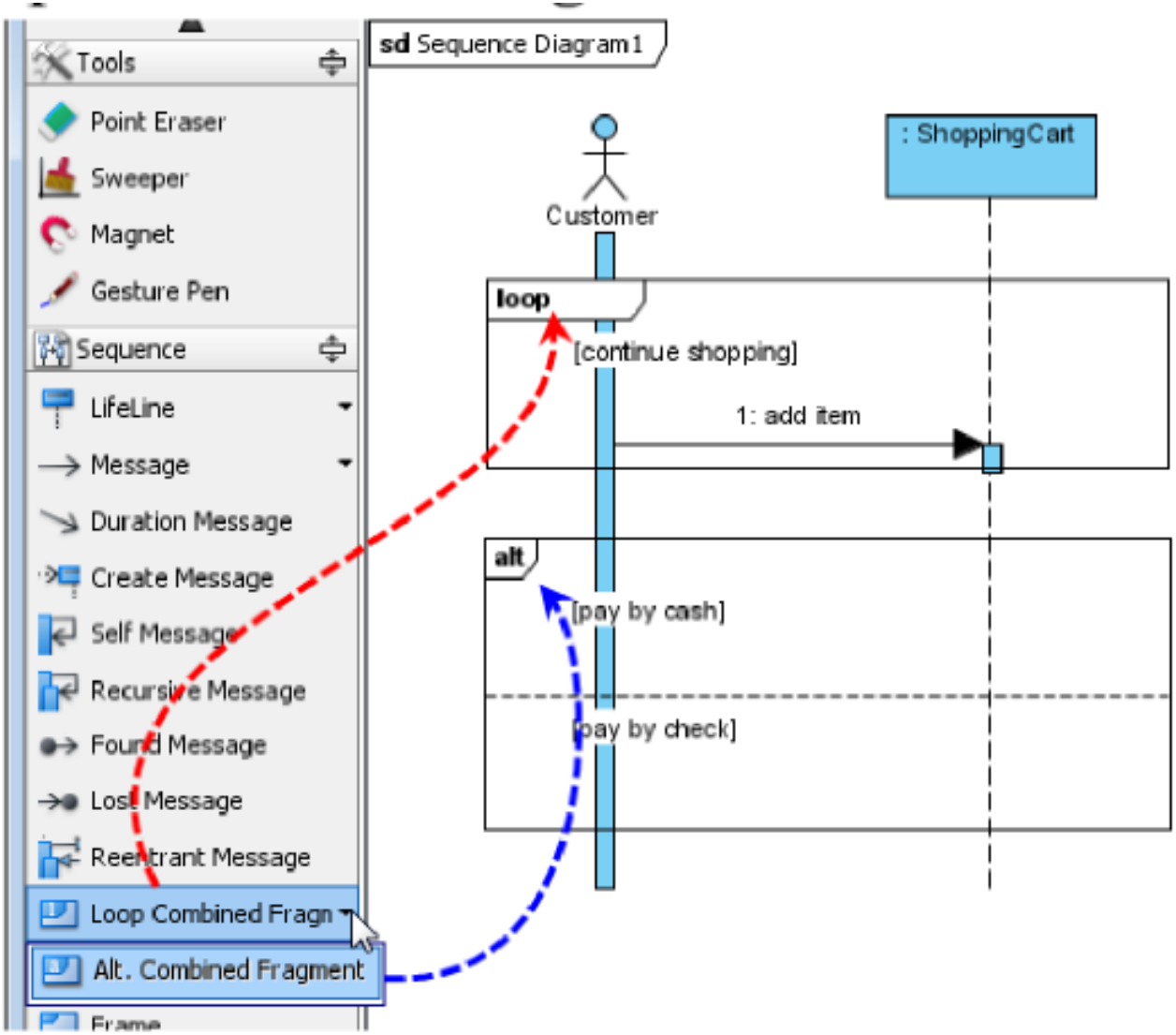
Sell Stocks SSD



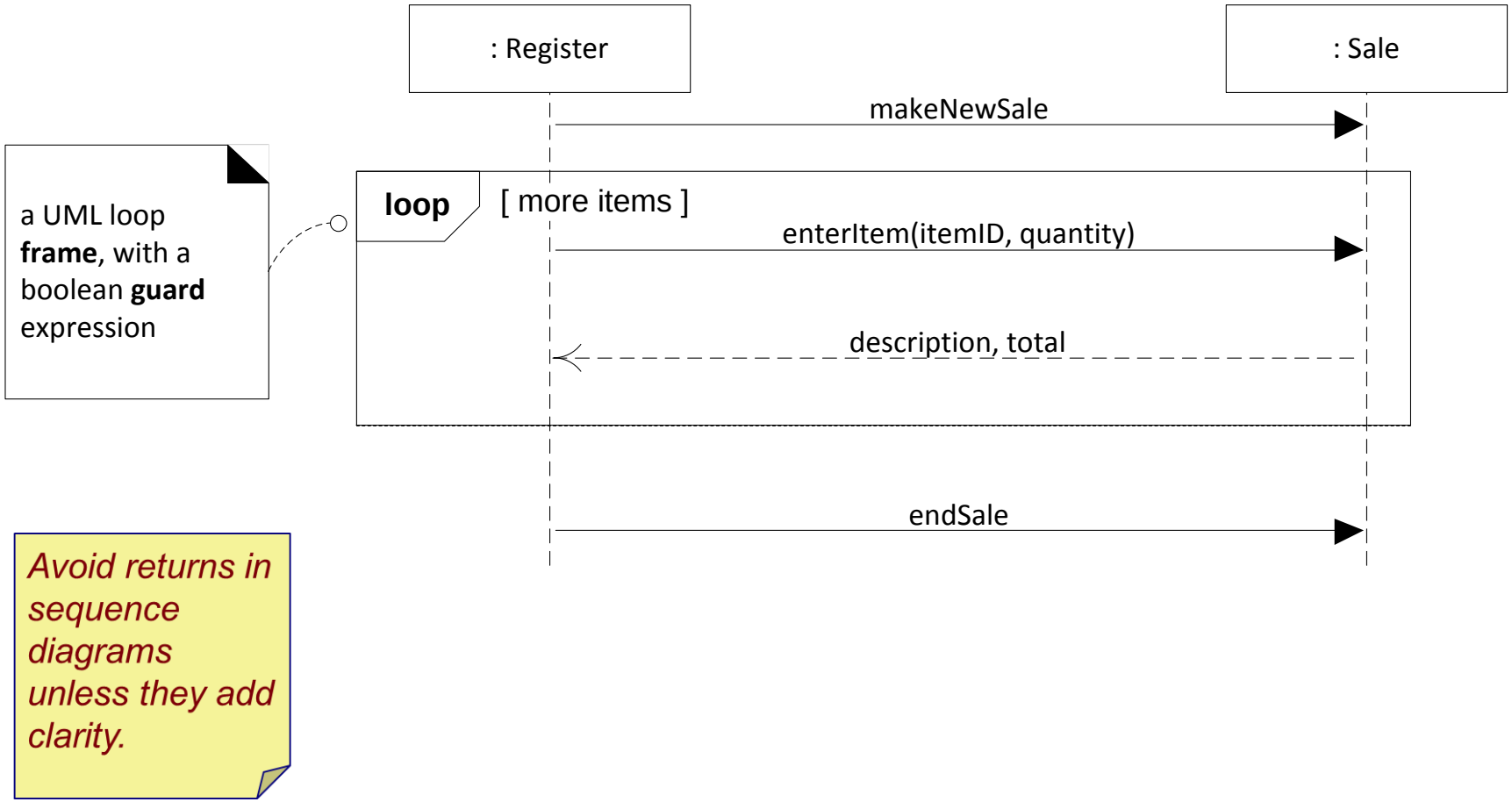
Sequence Diagram Common Operators

- Alternative fragment (denoted "**alt**") models **if…then…else constructs.**
  - Guard condition specify the true case for the execution of the interaction
- Option fragment (denoted "**opt**") models **switch constructs**.
  - Guard condition specified for each case.
- **Loop** fragment encloses a series of messages which are repeated.
  - Guard condition specify the lower and upper limit of the loop.
- **"ref"** refers to an interaction defined on another diagram.
- Parallel fragment (denoted "**par**") models concurrent processing.

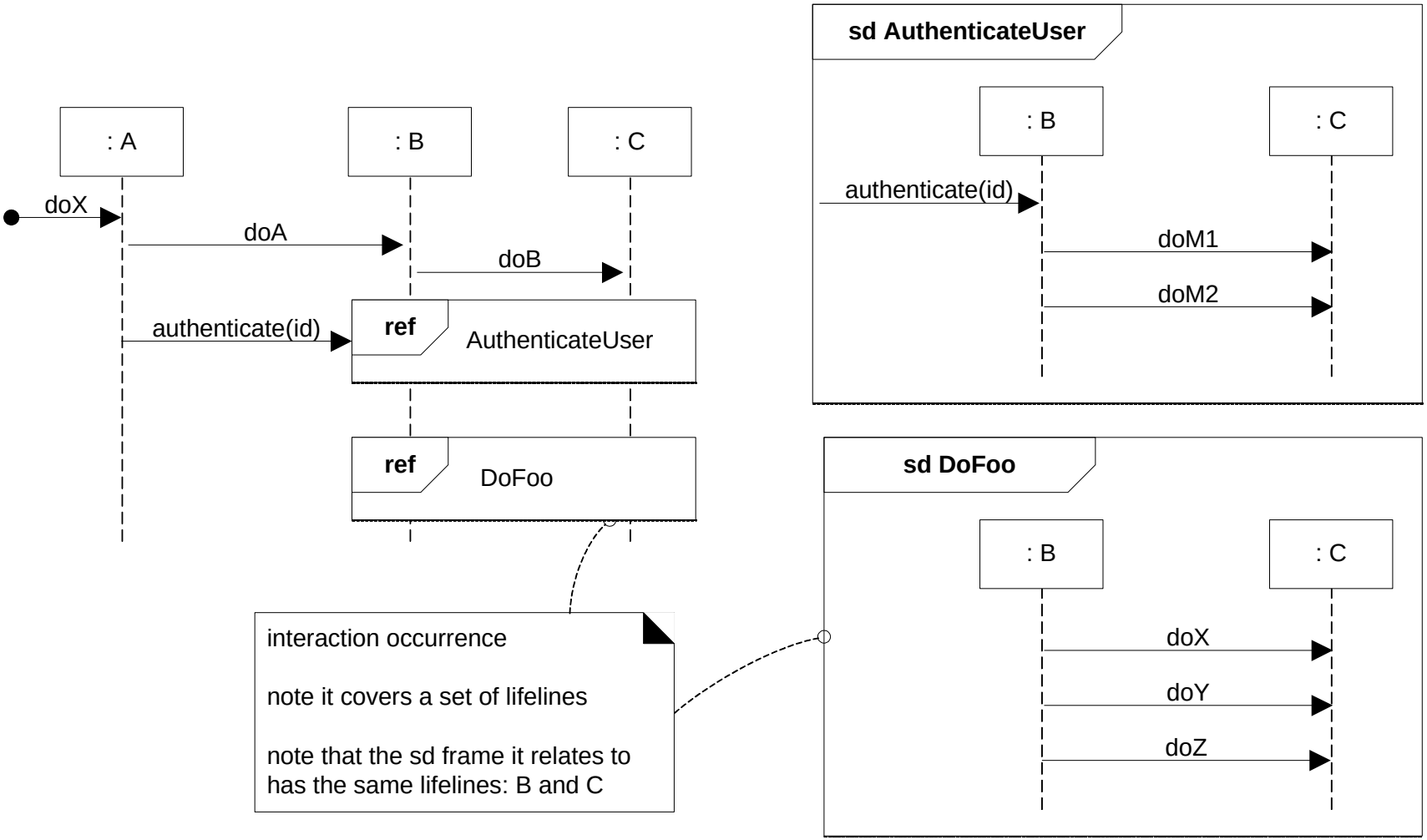
Example of using Alt and Loop



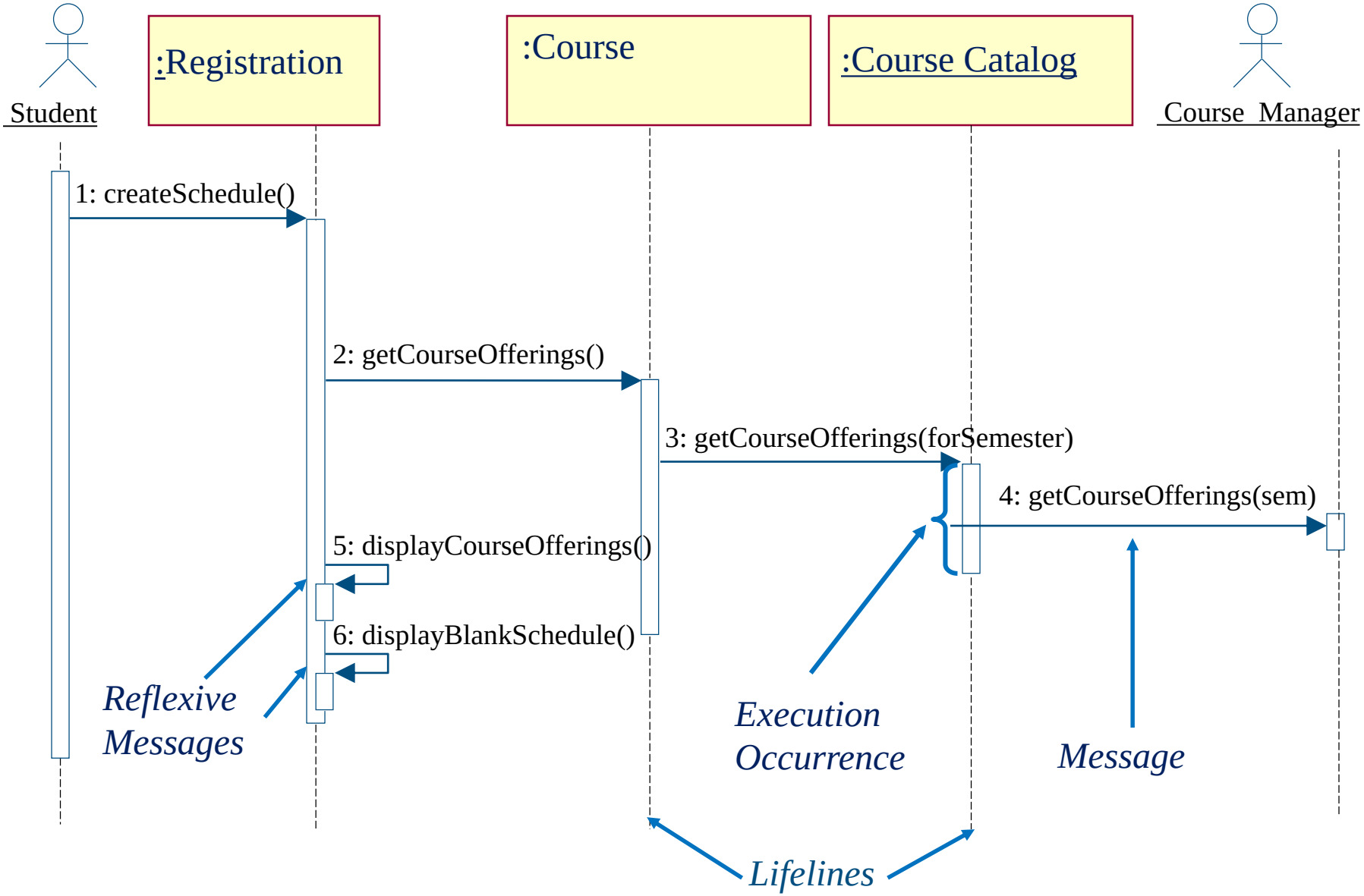
Sequence diagram with loop



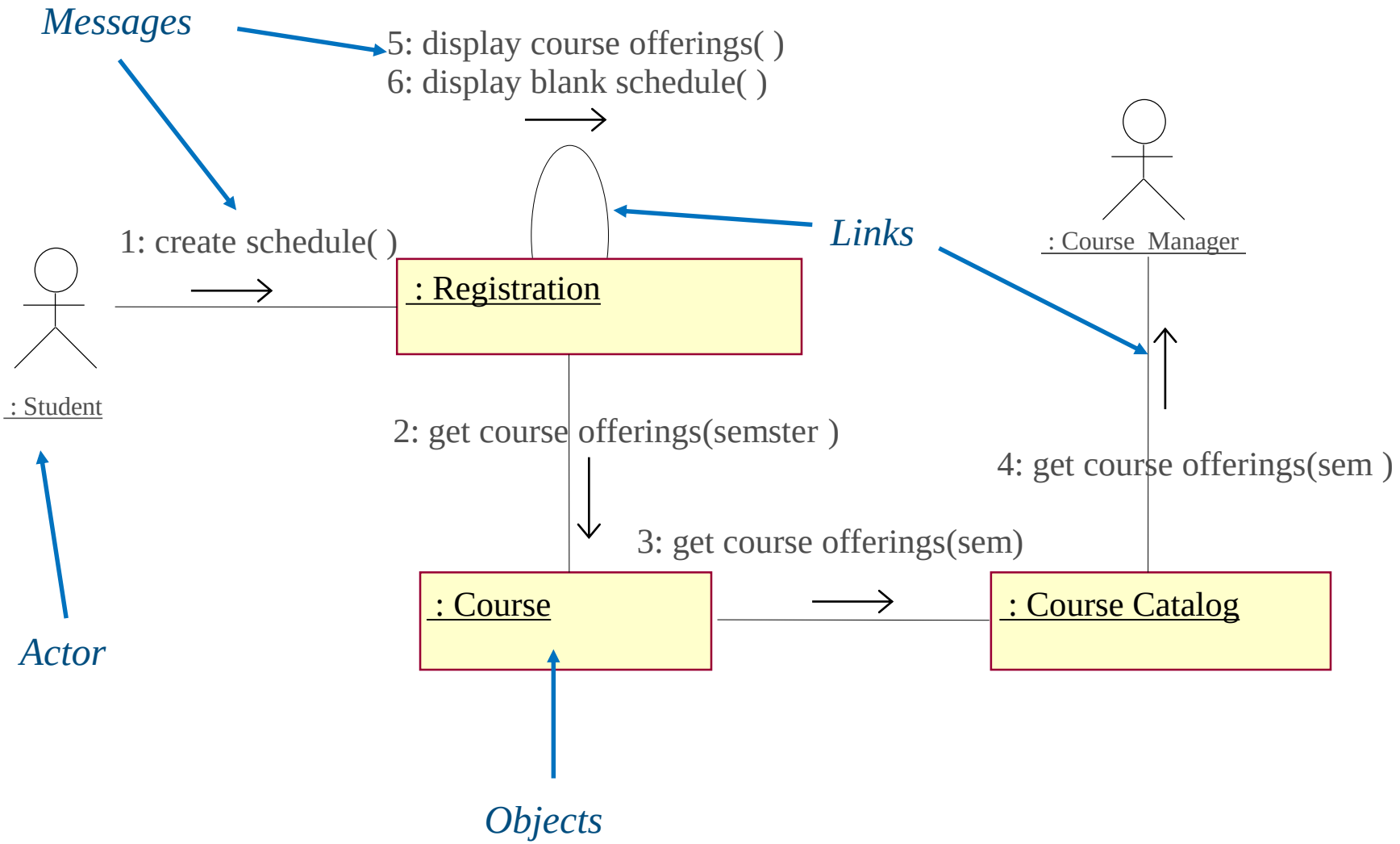
Referencing another SD



Sequence Diagram for Register for Courses Use Case



Communication Diagram Contents: Links and Messages



Design Sequence Diagram vs. Communication Diagram

- Semantically equivalent: Can convert one diagram to the other without losing any information
  - Model the dynamic aspects of a system
  - Model a use-case scenario
- Sequence diagrams
  - **Time-oriented**: better for visualizing overall flow
- Communication diagrams
  - **Message-oriented**: **useful for *validating* class diagrams**
  - Better for visualizing all of the effects on a given object

Conclusion

- After class diagrams, sequence diagrams are the most widely used diagrams in UML.
- It is impossible to model all possible interactions within a system.
- Only model those interactions that are interesting or shed light on important aspects of the system.
- A system of even modest complexity may require several interaction diagrams.