SLIIT

Discover Your Future

Software Engineering (IT2020)
2022
Lecture 3 - Communication Diagram

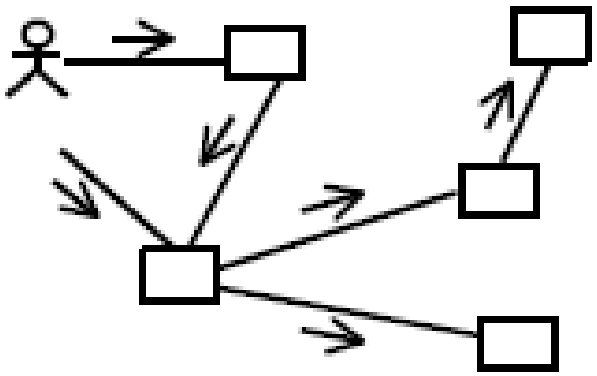
SLIIT
FACULTY OF COMPUTING

Session Outcomes

- Symbols of communication diagrams
  - Objects
  - Links
  - Messages and directions
  - Message sequence numbers
- Iteration and Looping
- Guard Expressions
- Parallel Activities

What Is a Communication Diagram?

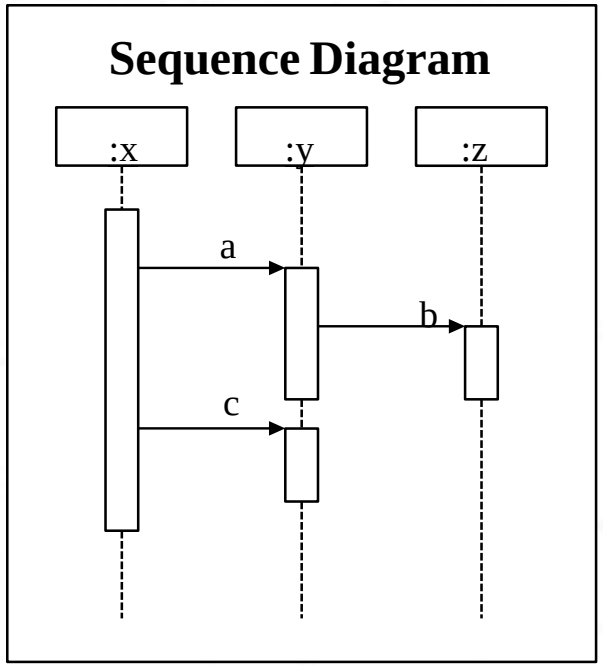
- A communication diagram emphasizes the organization of the objects that participate in an interaction.
- The communication diagram shows:
  - The objects participating in the interaction.
  - Links between the objects.
  - Messages passed between the objects.

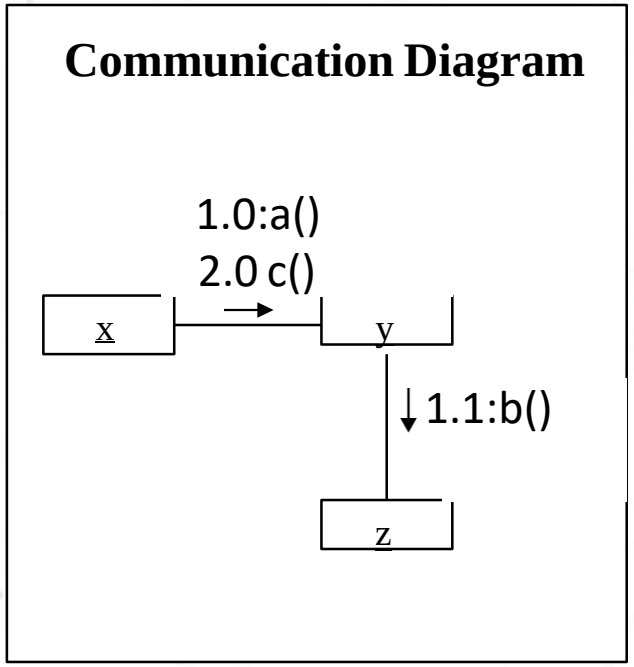


Communication Diagrams

Sequence and Communication Diagrams

- Interaction diagrams
  - Sequence diagram (temporal focus)
  - Communication diagram (structural focus)





Symbols of Communication Diagram

Actors : Each Actor is named and has a role

Object

Placed anywhere

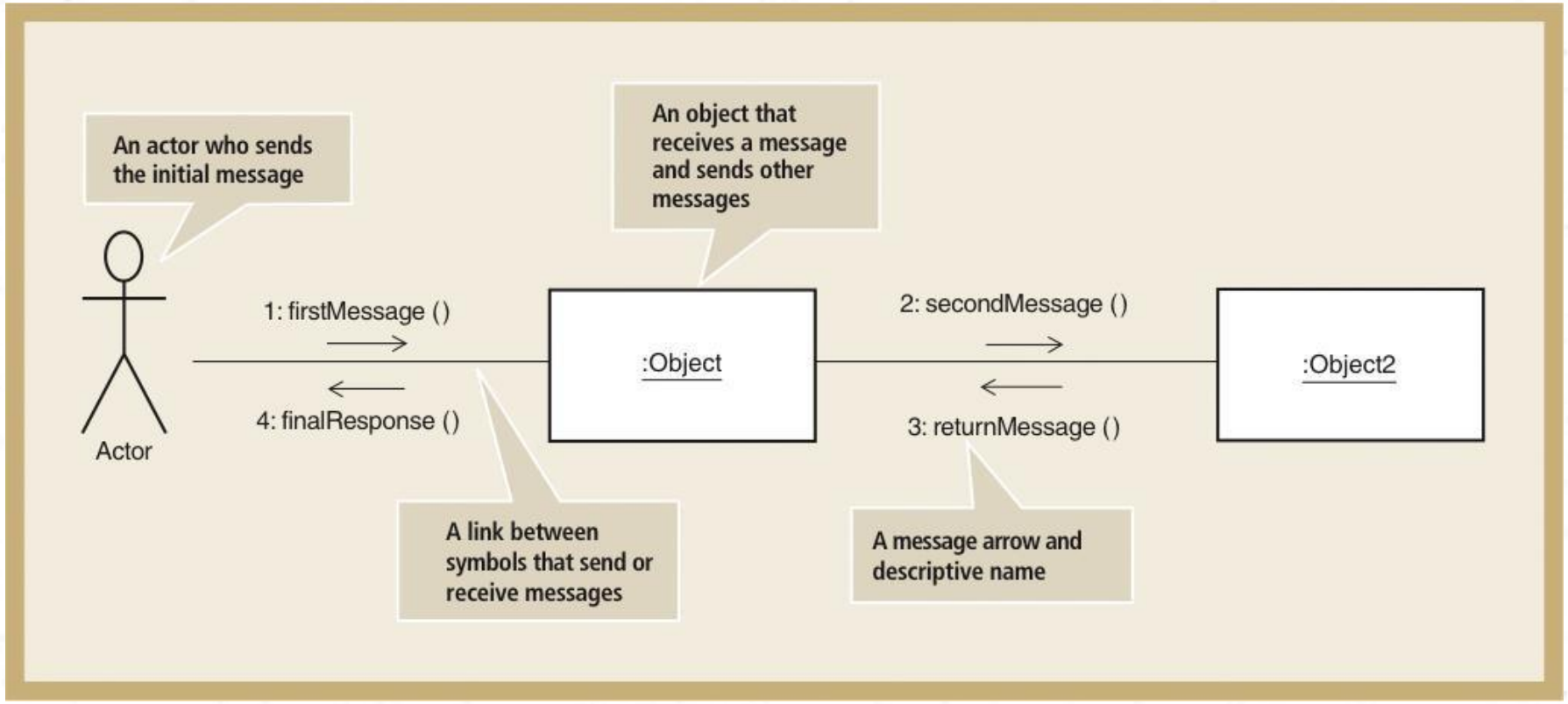
Links between objects

Direction of messages from one object to another object.

1, 2, 2.1, 2.2

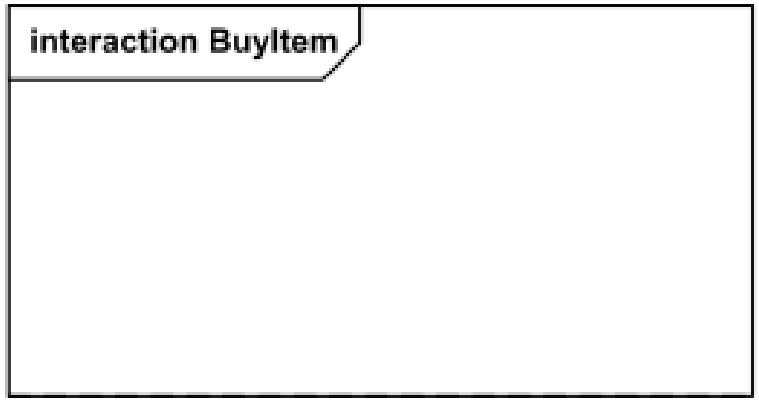
Message sequence numbers.

Communication Diagram - Example



Frame

- Communication diagrams could be shown within a rectangular frame with the diagram name in the name box preceding with the “interaction” keyword.



Interaction ***Frame*** *for* ***Communication Diagram*** *BuyItem*

Objects and Links

- Objects : Similar to Sequence Diagram.
- The connecting lines drawn between objects are links.
- They enable you to see the relationships between objects.
- This symbolizes the ability of objects to send messages to each other.
- A single link can support one or more messages sent between objects

Messages

- The message types in a Communication diagram are the same as in a Sequence diagram.

Message Syntax

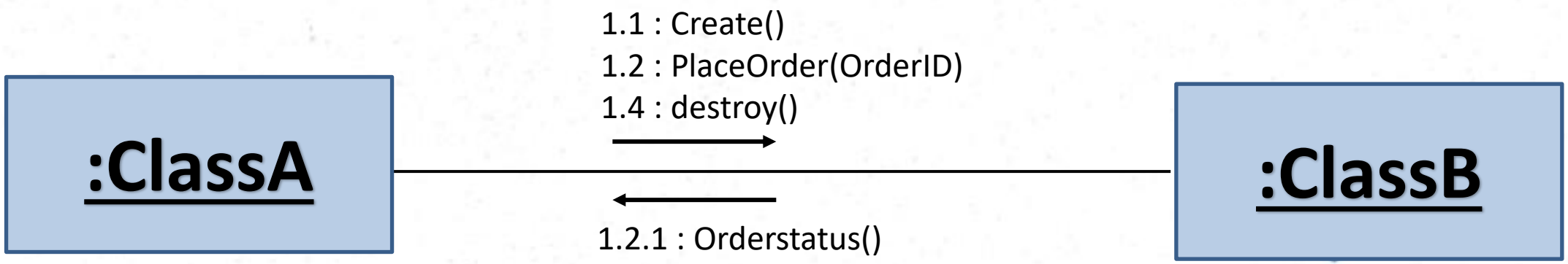
Message Sequence Number : Message signature

e.g. 1.0 : Login (UserName, Pwd)

3.1.1 : getPerformance ( )

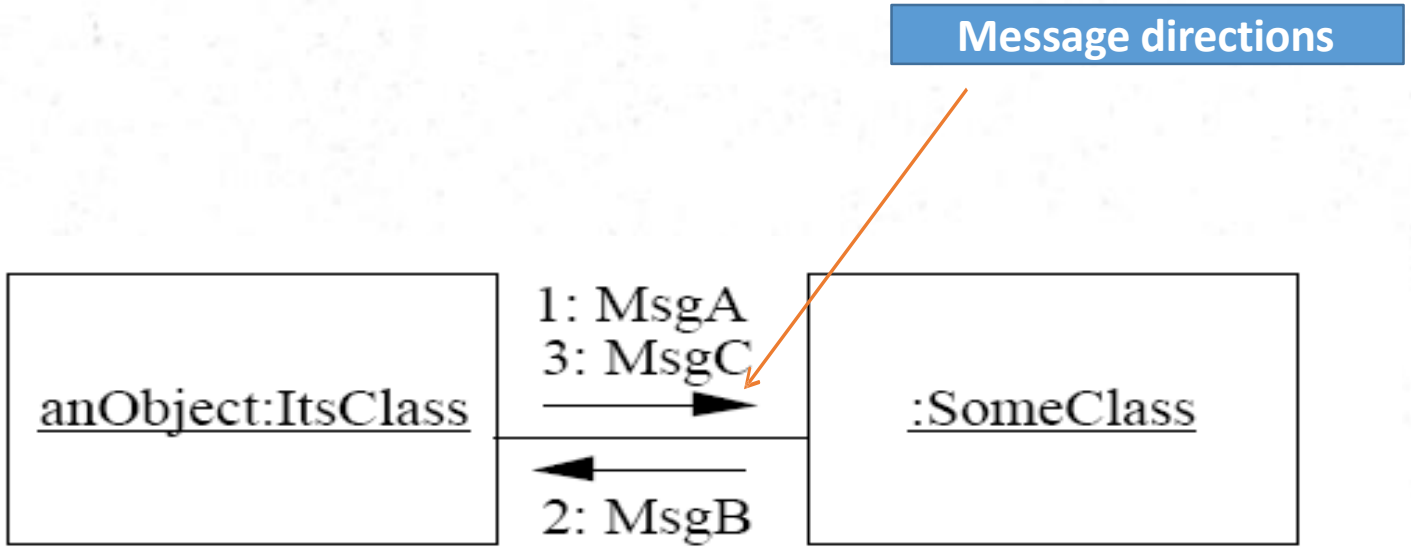
Message Types

In a communication diagram all the message types (Synchronous, Asynchronous, Create, Destroy and Reply) indicate in the same way.

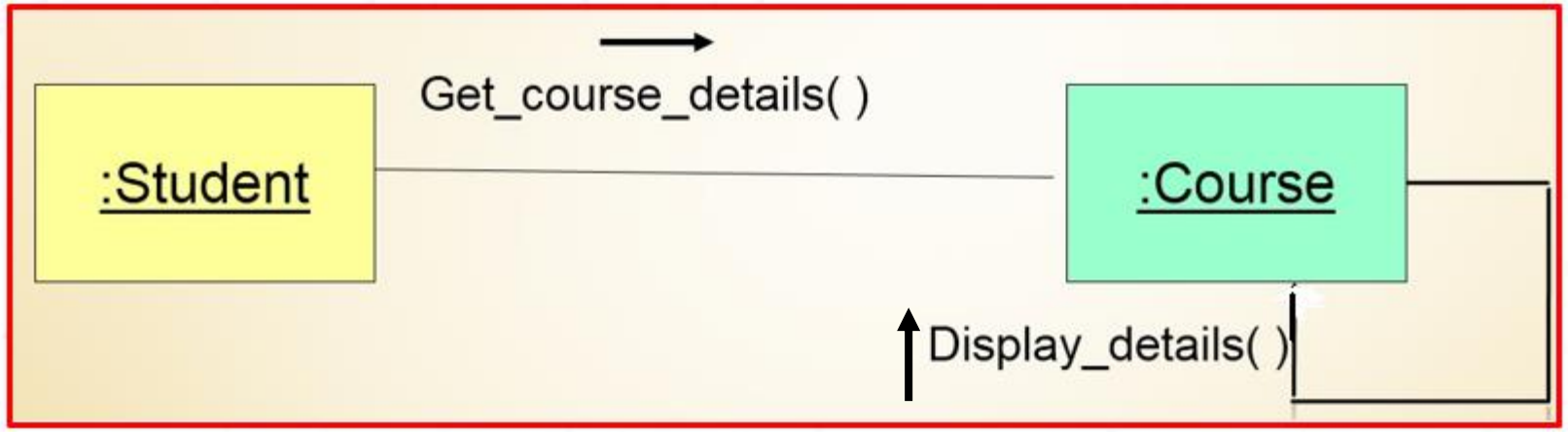


Message Directions

- A message on a communication diagram is shown using an arrow from the message sender to the message receiver.



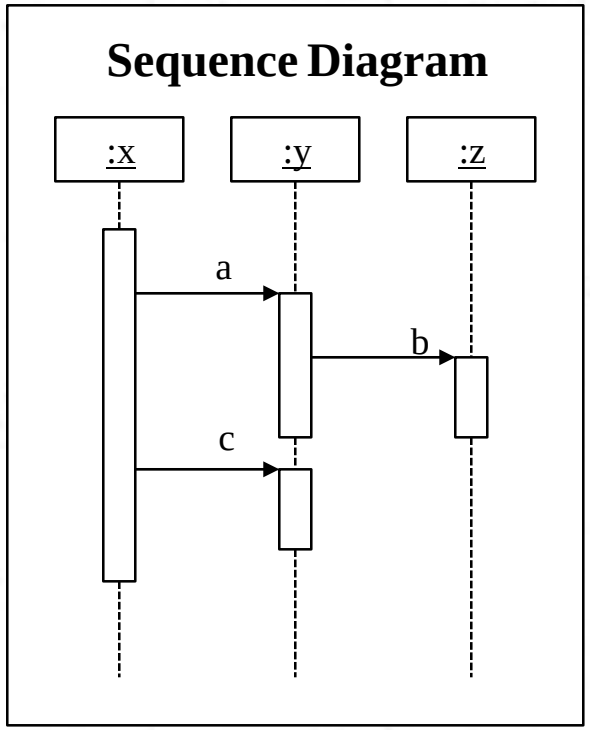
Self Calls

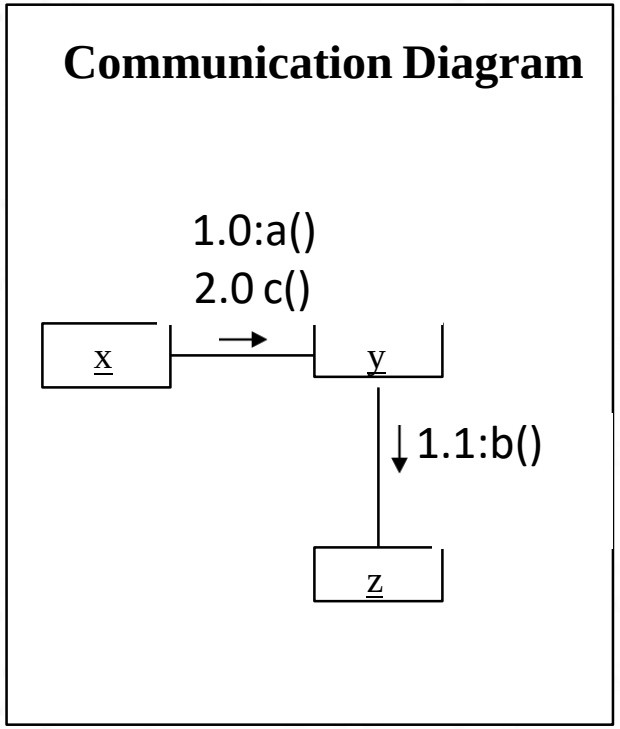


Message Sequence Numbers

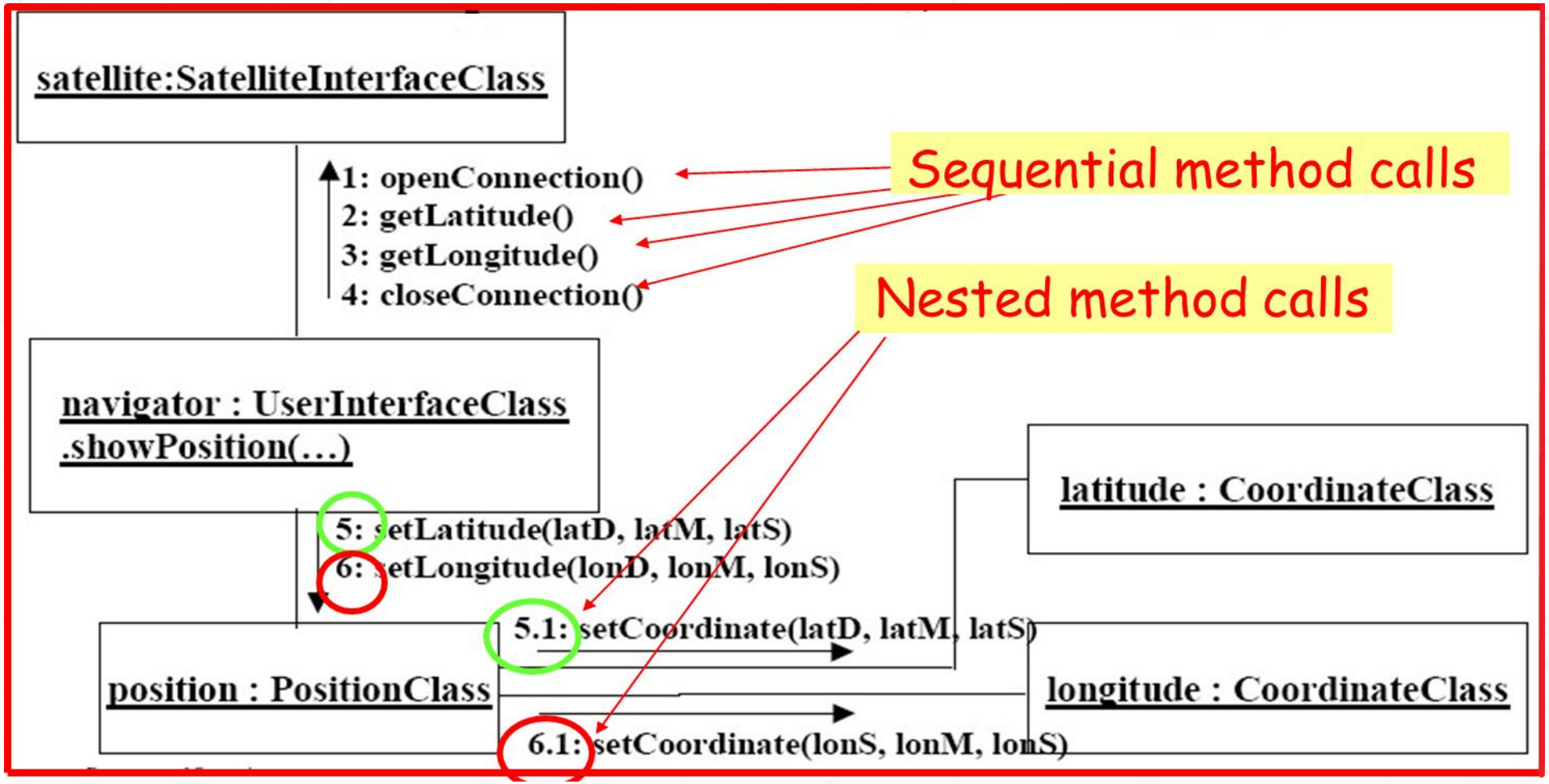
- "Message Sequence number" is the integer represents the sequential order of the message.
- Each sequence term represents a level of procedural nesting.
- If message sequence numbers are at the same dot-level such as 1.1 and 1.2, those messages are considered to be sequential.
- If the model adds steps 1.1.1 and 1.1.2, then these new steps are understood to execute after step 1.1 and before step 1.2.
- In other words, they are nested beneath/within step 1.1.

Message Numbering – example 1



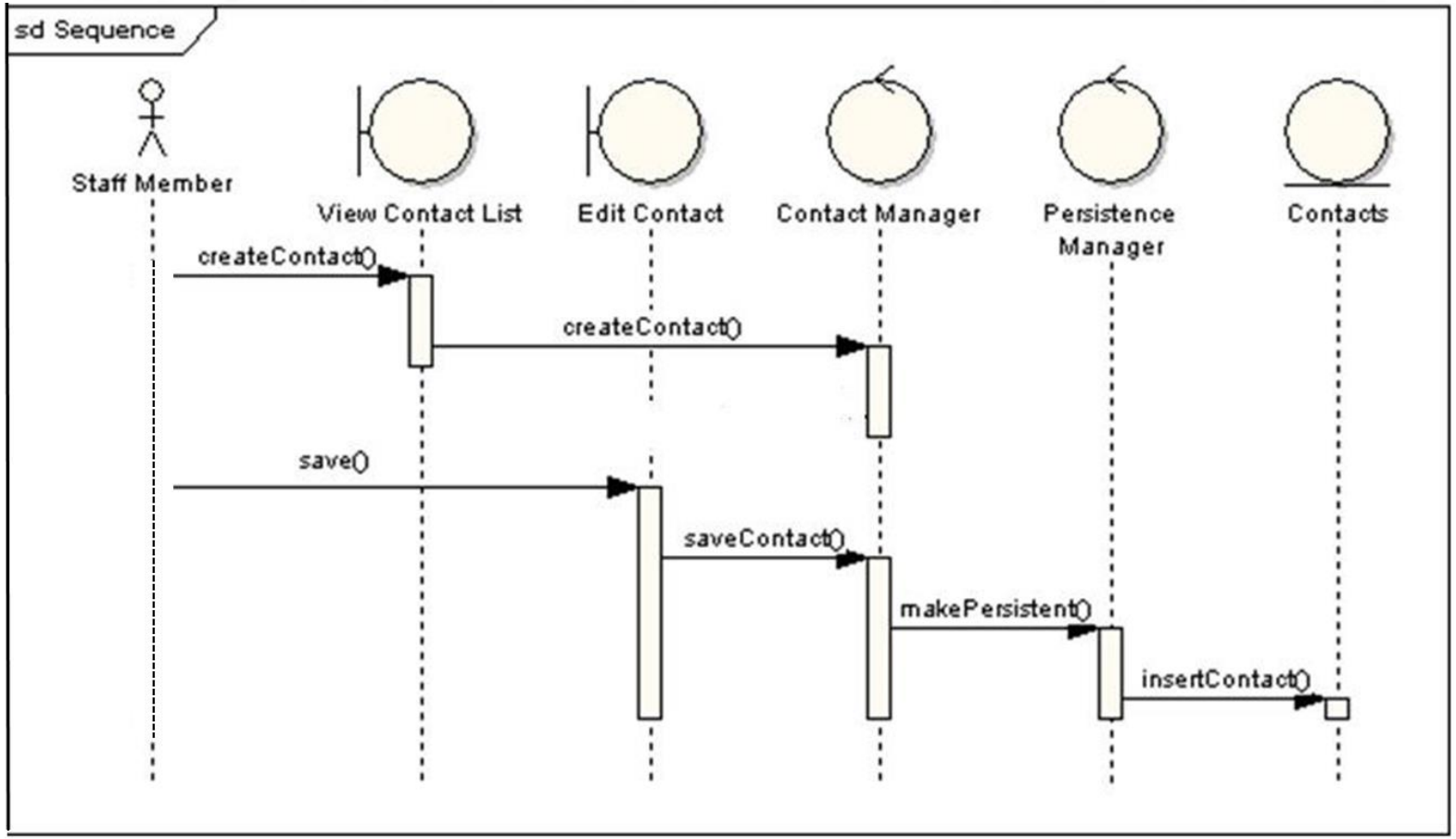


Message Numbering - example 2



Activity 1

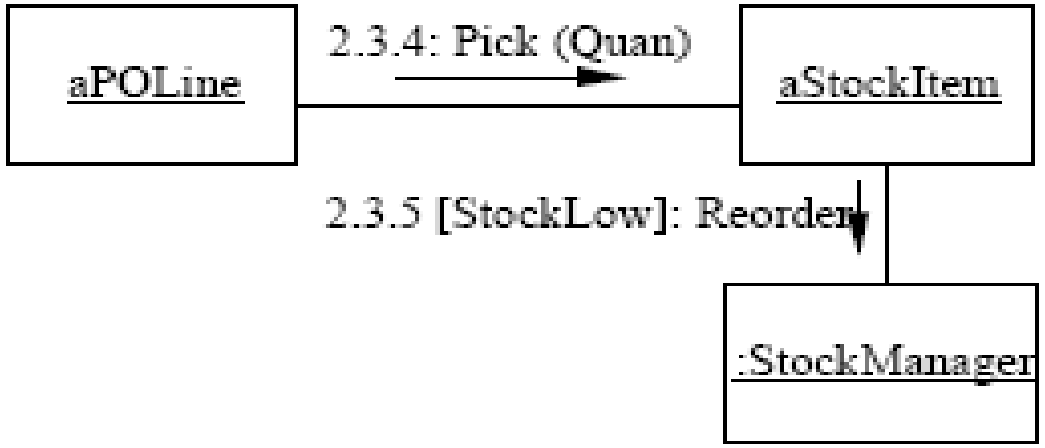
Convert following sequence diagram to a communication diagram



Guard Expressions

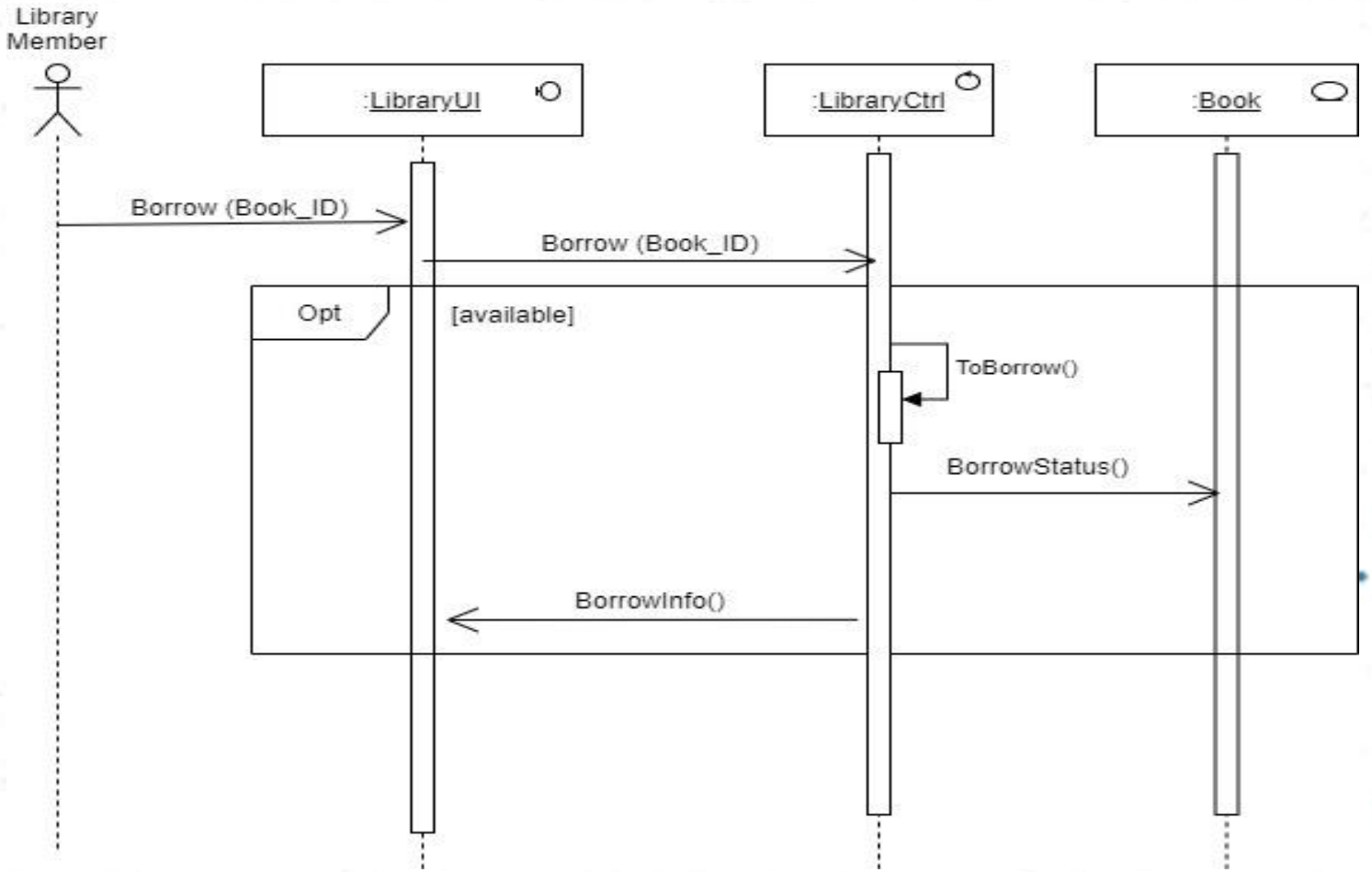
- Use to indicate messages which send under a certain condition.
- The message will be send only if the condition in the square bracket is true.

Syntax:- **message sequence number [condition] : Message**



Activity 2

Convert following sequence diagram into a communication diagram



Iteration and Looping

- A message may be executed repeatedly.
- The message repeats while the condition in the square brackets is true.

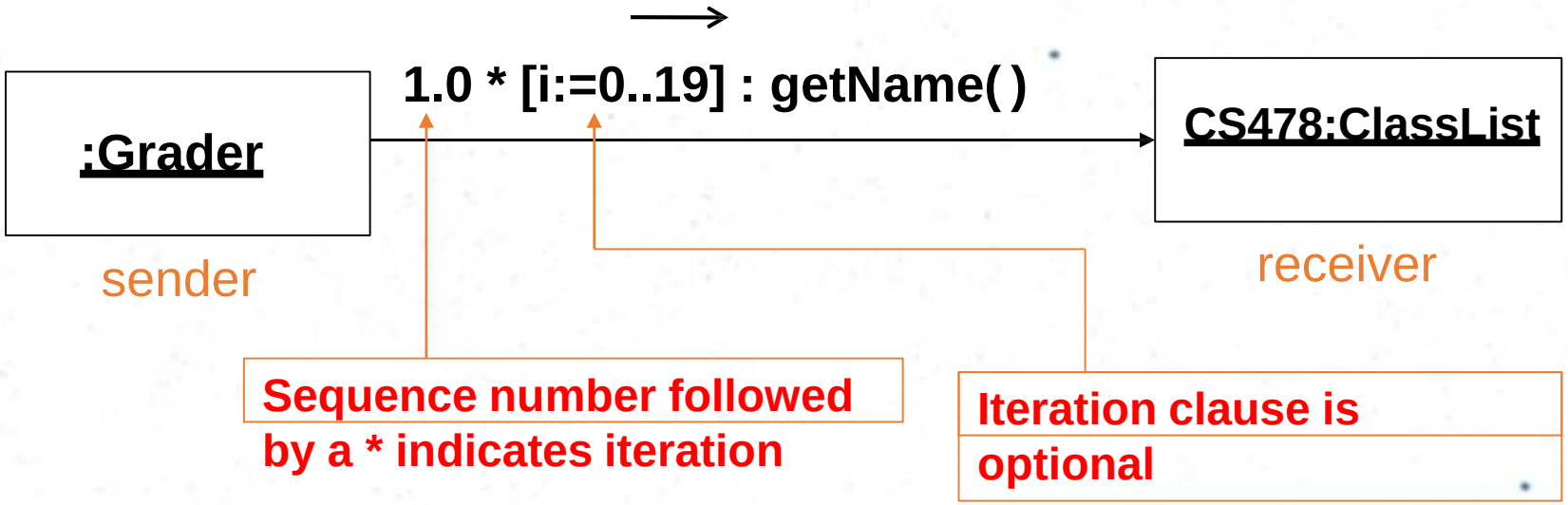
Syntax:

Message Sequence Number *[Condition] : Message signature
Message Sequence Number *[Condition][iterative clause] : Message signature

"The asterisk (*) indicates that the message is repeating"

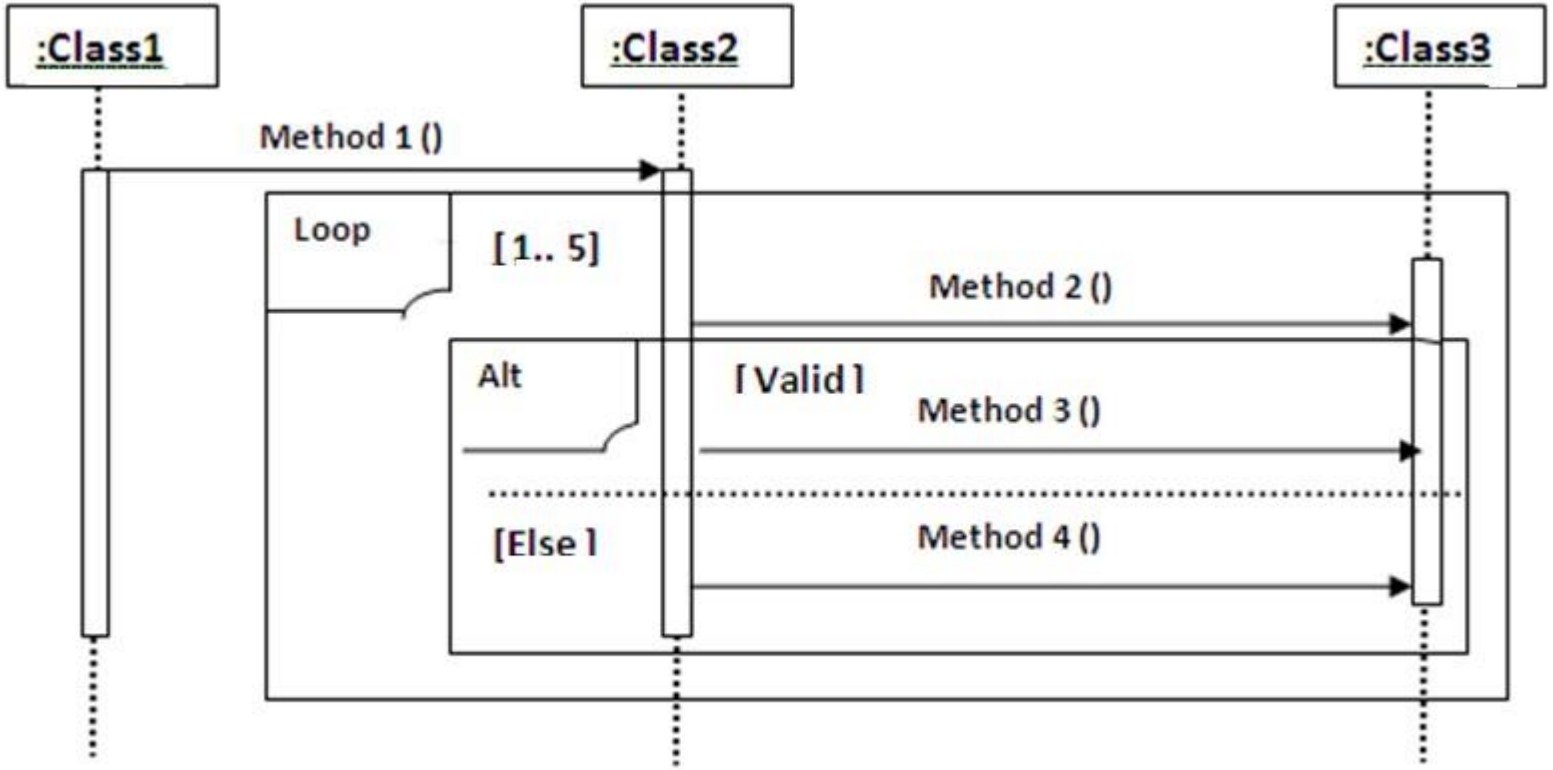
Example:
1.2 * [amount > 50,000] : Withdraw()
1.3 * [incorrect password] [i:=1..3] : Relogging()

Iteration and Looping - Example



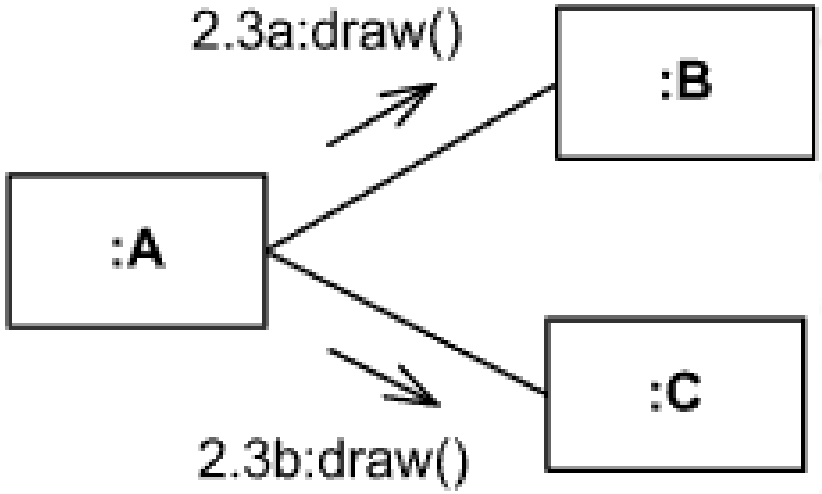
Activity 3

Convert following sequence diagram into a communication diagram



Parallel Activities

Indicate concurrent threads of execution in a UML communication diagram by having letters precede the sequence numbers on messages.



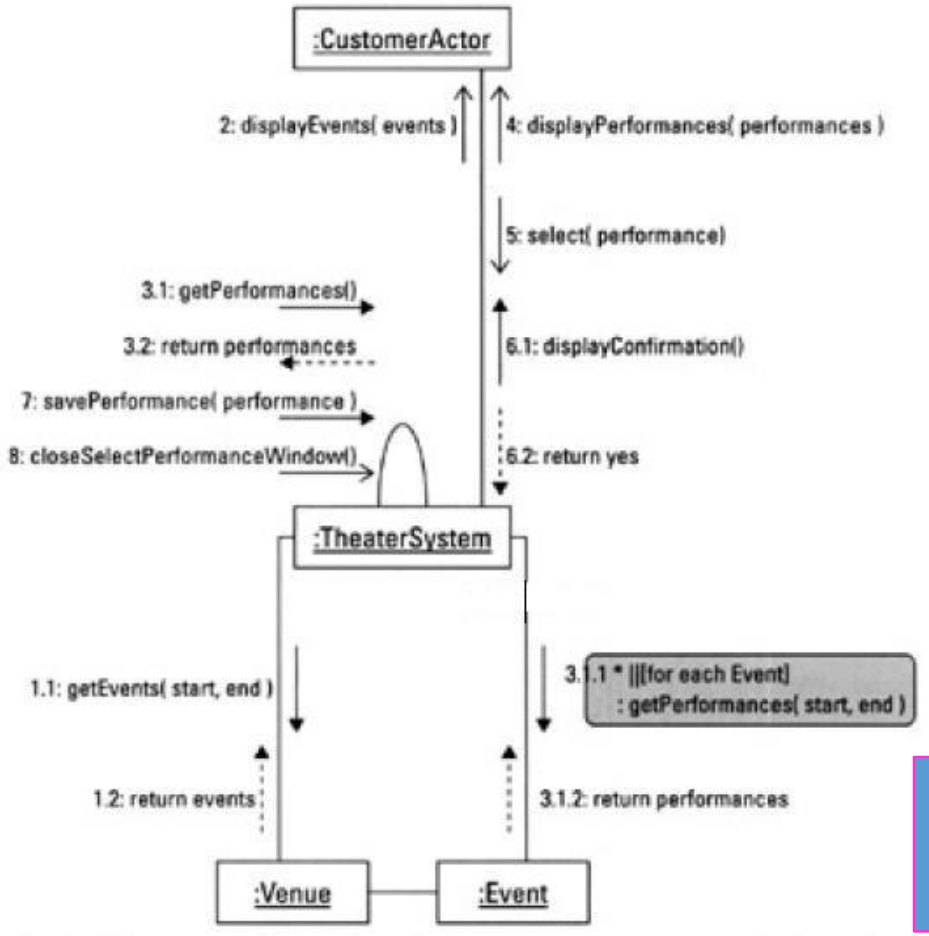
Instance of class A sends draw() messages concurrently to instance of class B and to instance of class C

Iteration and Parallel activities

- The iteration expression assumes that the messages in the iteration will be executed sequentially. But this is not always true.

- To model the fact that the messages may execute concurrently (in parallel), use a pair of vertical lines (||) after the iteration indicator (*).

Iteration and Parallel activities example



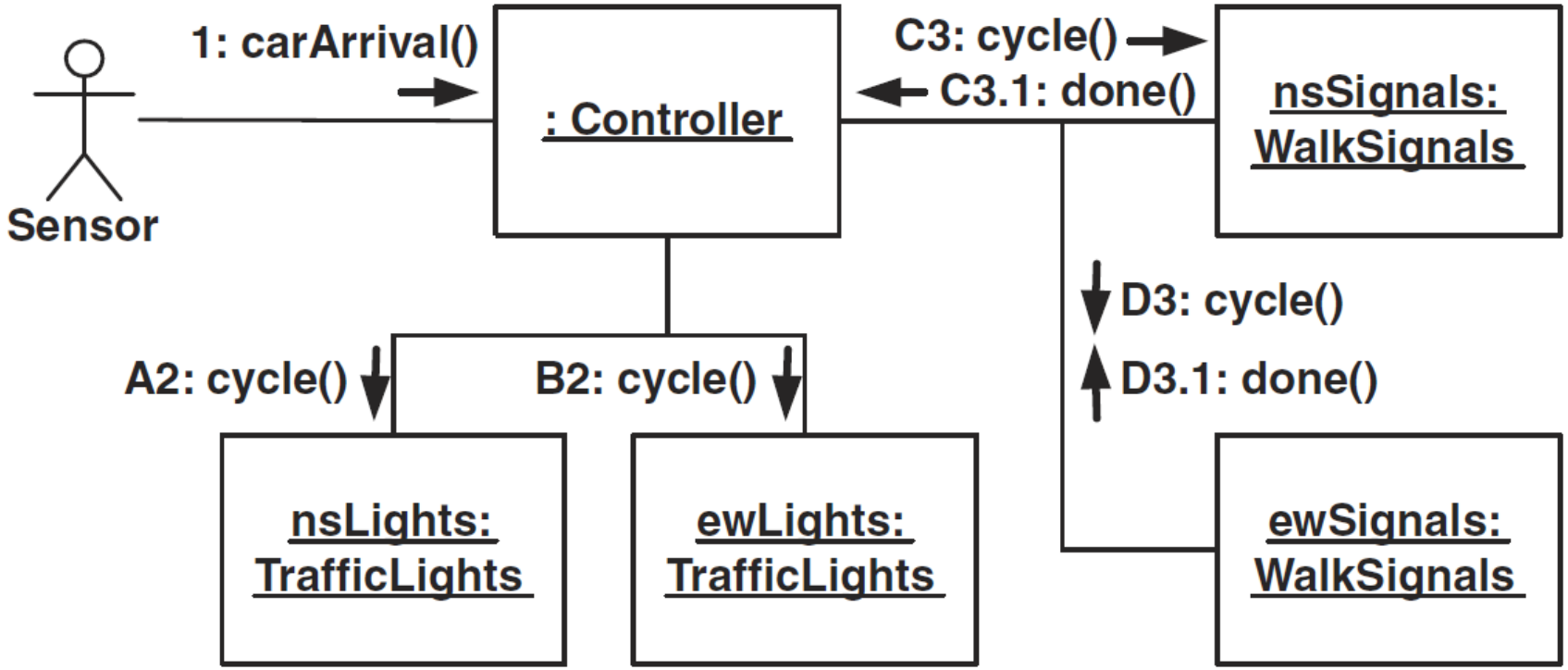
message 3.1.1
retrieves the performances for each event, one at a time.

But we could change it to retrieve the performances for all events concurrently by adding the concurrency notation to the sequence term.

3.1.1 * | | [For each Event] : getPerformances (start, end)

Activity 4

Find the concurrent activities in the following communication diagram.



Rules of Thumb

- **Avoid crossing links** and **crowded** diagrams.
- **Do not show all interactions** on an interaction diagram - only what is important for the scenario.
- Do Not Model **Obvious Return Values**.
- Model a return value **only when you need to refer to it elsewhere** in a diagram.

Sequence Diagram vs. Communication Diagram

- Sequence diagrams emphasis the **sequences of events** well.
- Communication diagrams show the **relationships between the classes** well.
- Keep both types of diagrams simple.

Strengths and Weaknesses

Type	Strengths	Weaknesses
Sequence	Show sequence or **time** order	Forced to extend to the right when adding new objects
Communication	Flexibility to add new objects in two dimensions. Better to illustrate complex branching, iteration and concurrent behavior	Difficult to see sequence of messages

Sequence and Communication Diagram Similarities

- Semantically equivalent.
- Can convert one diagram to the other without losing most of the information.
- Model the dynamic aspects of a system.
- Model the implementation of a use-case scenario.

References

- UML 2 Bible
  - Chapters 8 & 9
- Applying UML and Patterns by Craig Larman
  - Chapter 15
- TheElementsofUML2Style
  - Chapter 7

Thank you