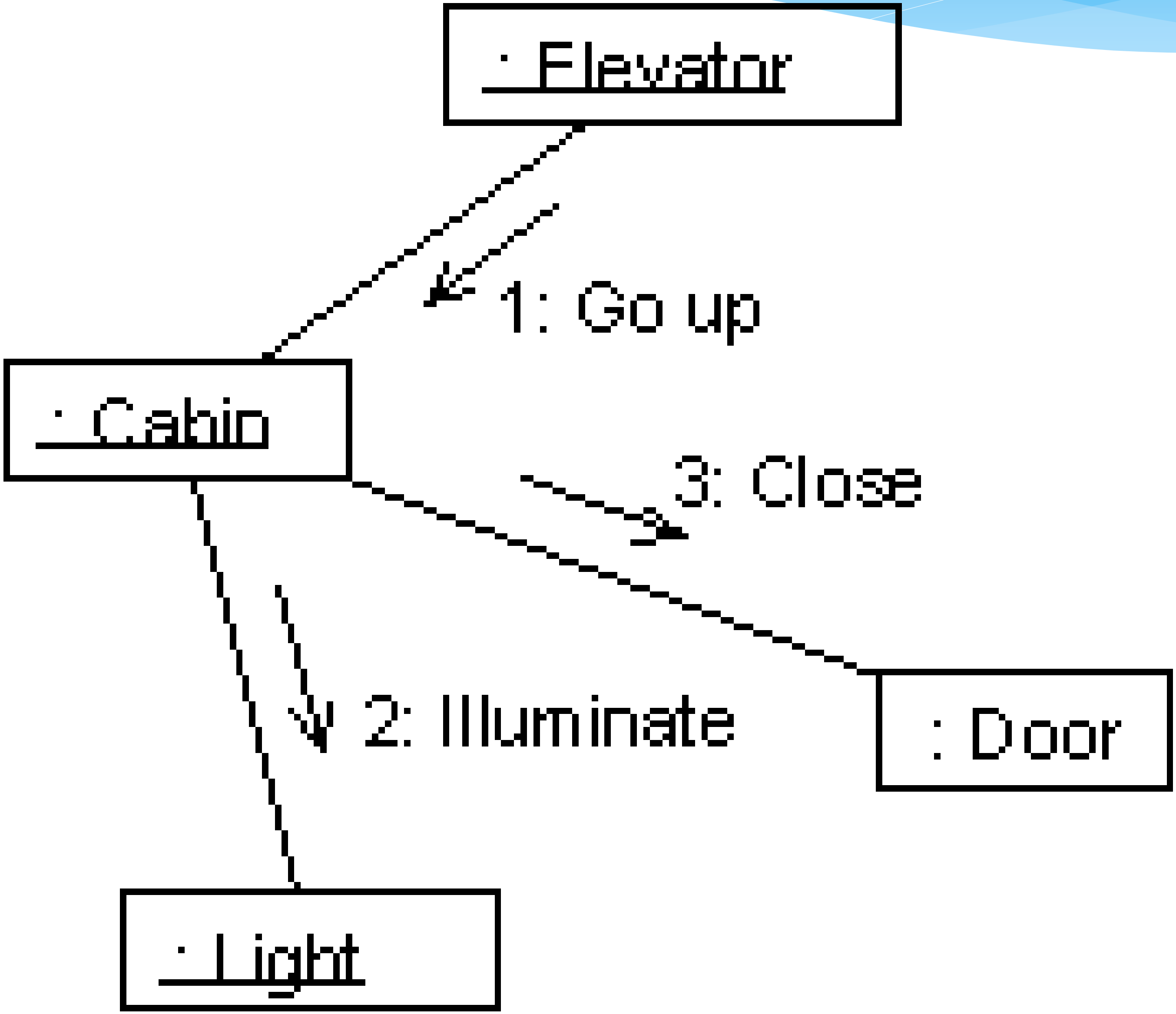
Communication Diagrams

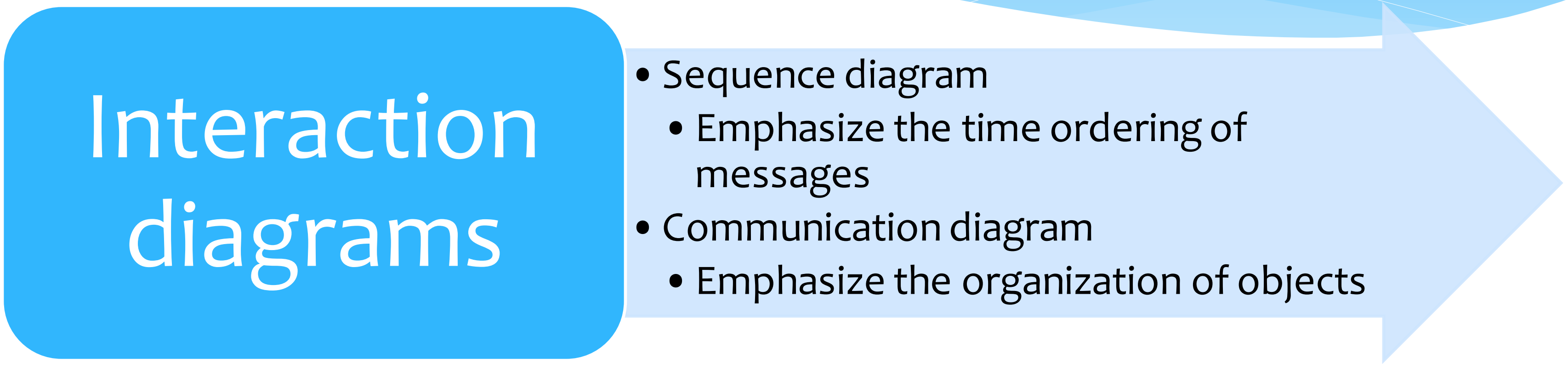
Lecture 11


: Elevator
1: Go up
: Cabin
3: Close
2: Illuminate
: Door
: Light

Introduction

- Interaction Diagrams are used to model system dynamics
  - How do objects change state?
  - How do objects interact (message passing)?

Sequence and communication Diagrams

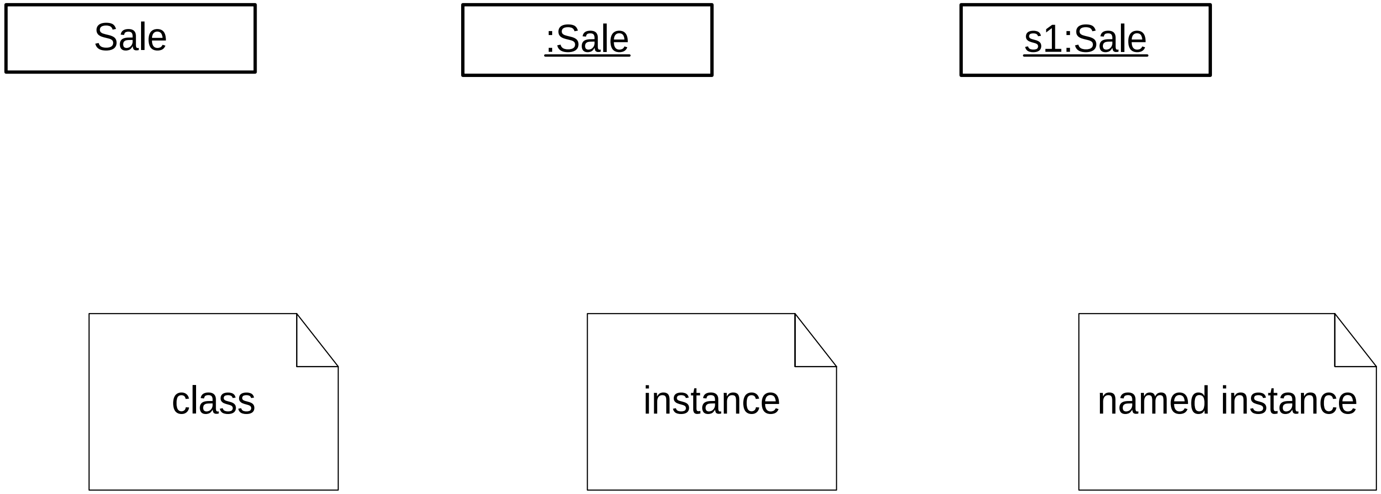


Interaction diagrams

- Sequence diagram
  - Emphasize the time ordering of messages
- Communication diagram
  - Emphasize the organization of objects

- Can be used interchangeably
- Communication Diagrams explicitly show object linkages, while links are implied in Sequence Diagrams
- Communication diagram introduced in UML 2.0. It original name in UML 1.x is Collaboration Diagram.

Common Interaction Diagram Notation



Sale

:Sale

s1:Sale

class

instance

named instance

- A Collaboration is a collection of named objects and actors with links connecting them

- Objects collaborate by communicating (passing messages) with one another in order to work together

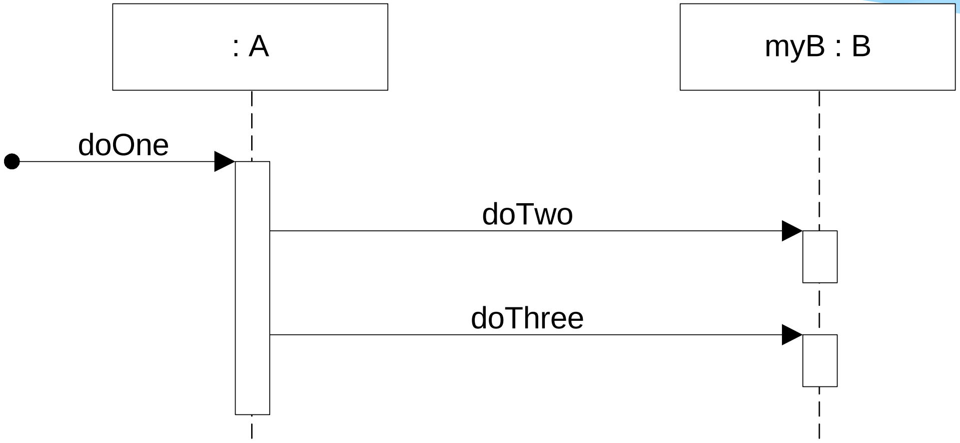
Communication Diagrams

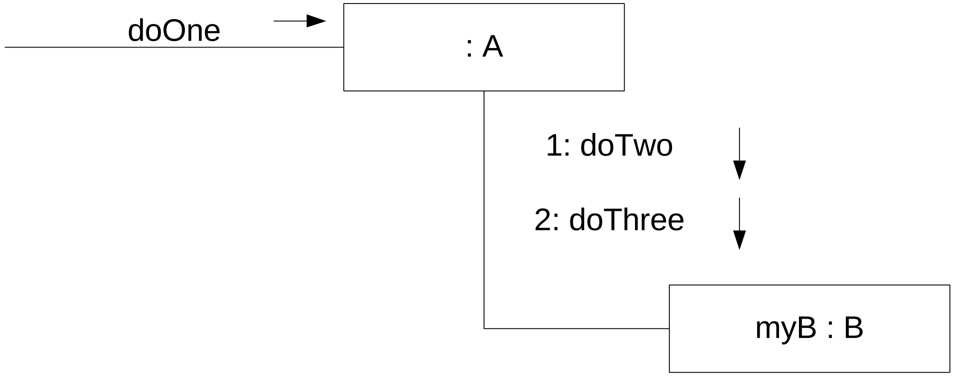
- Actors
  - Each Actor is named and has a role
  - One actor will be the initiator of the use case
- Objects
  - Each object in the collaboration is named and has its class specified
  - Not all classes need to appear
  - There may be more than one object of a class
- Links
  - Links connect objects and actors and are instances of associations
  - Each link corresponds to an association in the class diagram

Sequence Number

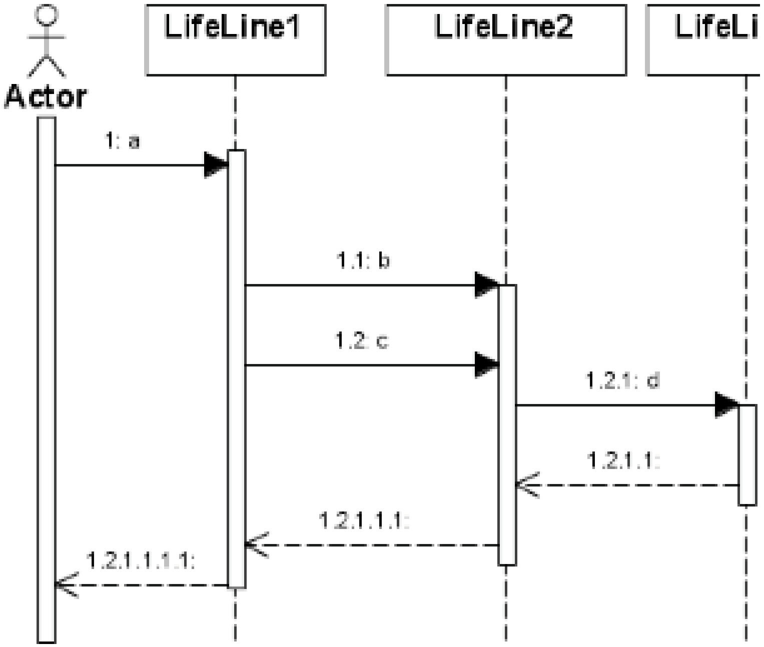
  - Represents the order in which the flows are used

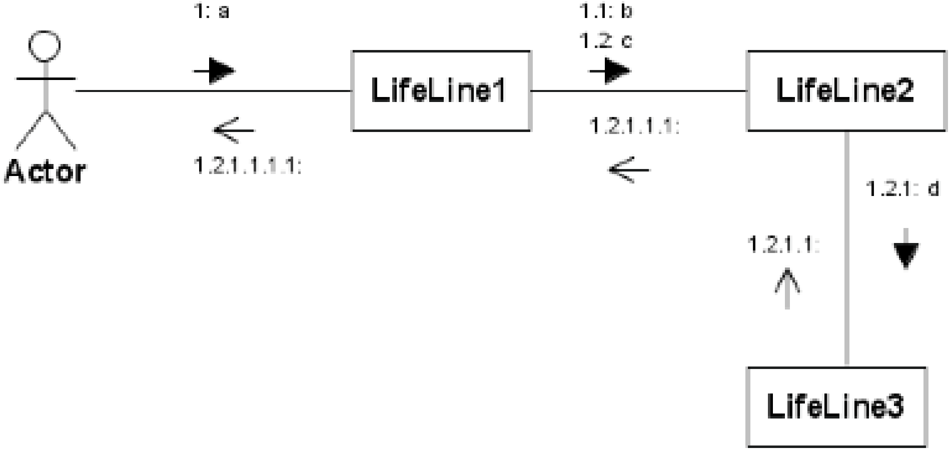
Communication vs. Sequence Diagram





Communication vs. Sequence Diagram





Messages

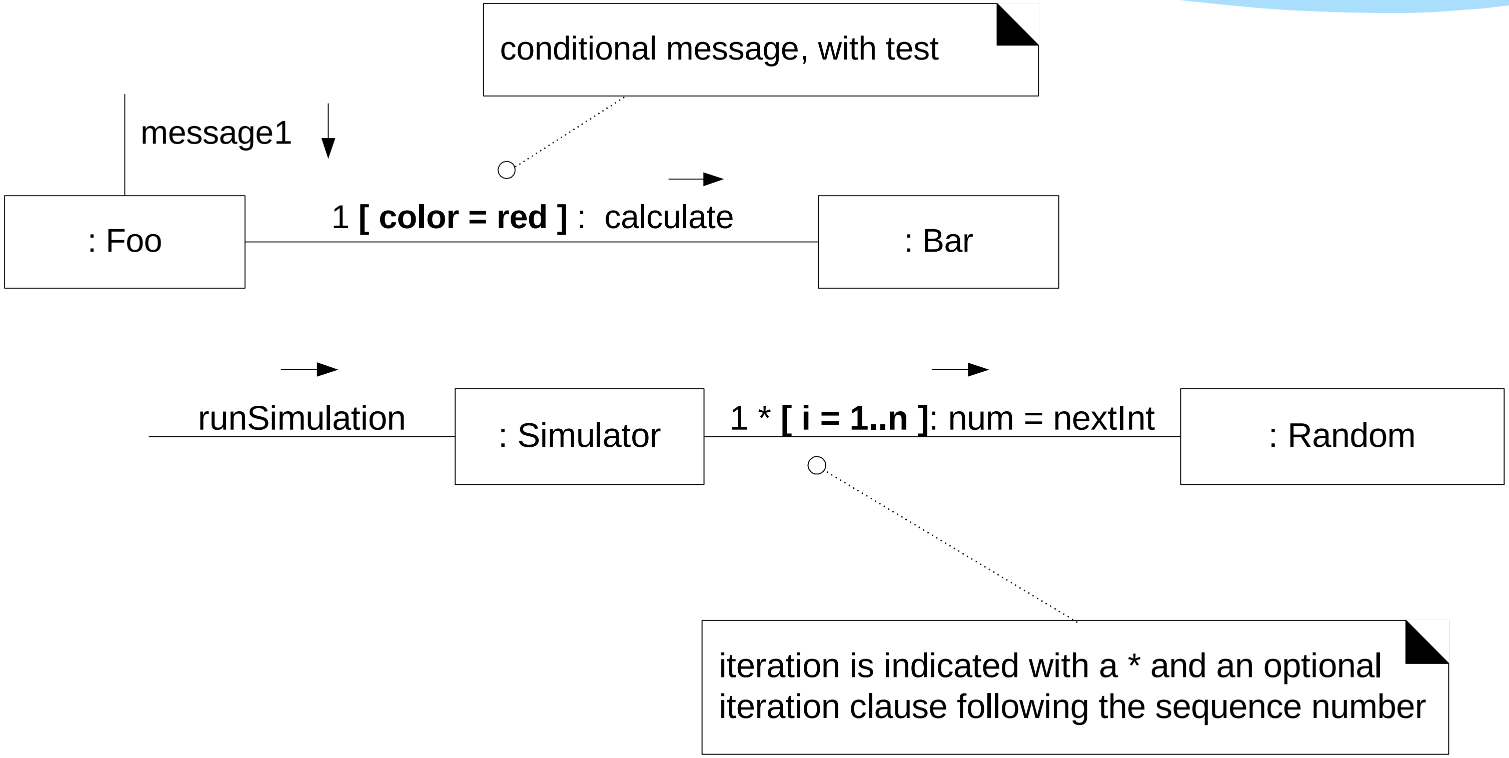
- A message on a communication diagram is shown using an arrow from the message sender to the message receiver
- Message Signature: return-value, message-name, argument-list
- Each message in a collaboration diagram has a sequence number. The top-level message is numbered 1. Messages sent during the same call have the same decimal prefix but suffixes of 1, 2, etc. according to when they occur.

Basic Communication Diagram Notation

- Conditional Message
  - Seq. Number *[variable = value]* : message()
  - Message is sent only if clause evaluates to *true*

- Iteration (Looping)
  - Seq. Number * *[i := 1..N]*: message()
  - "*" is required; [...] clause is optional

Messages – Conditional and Looping

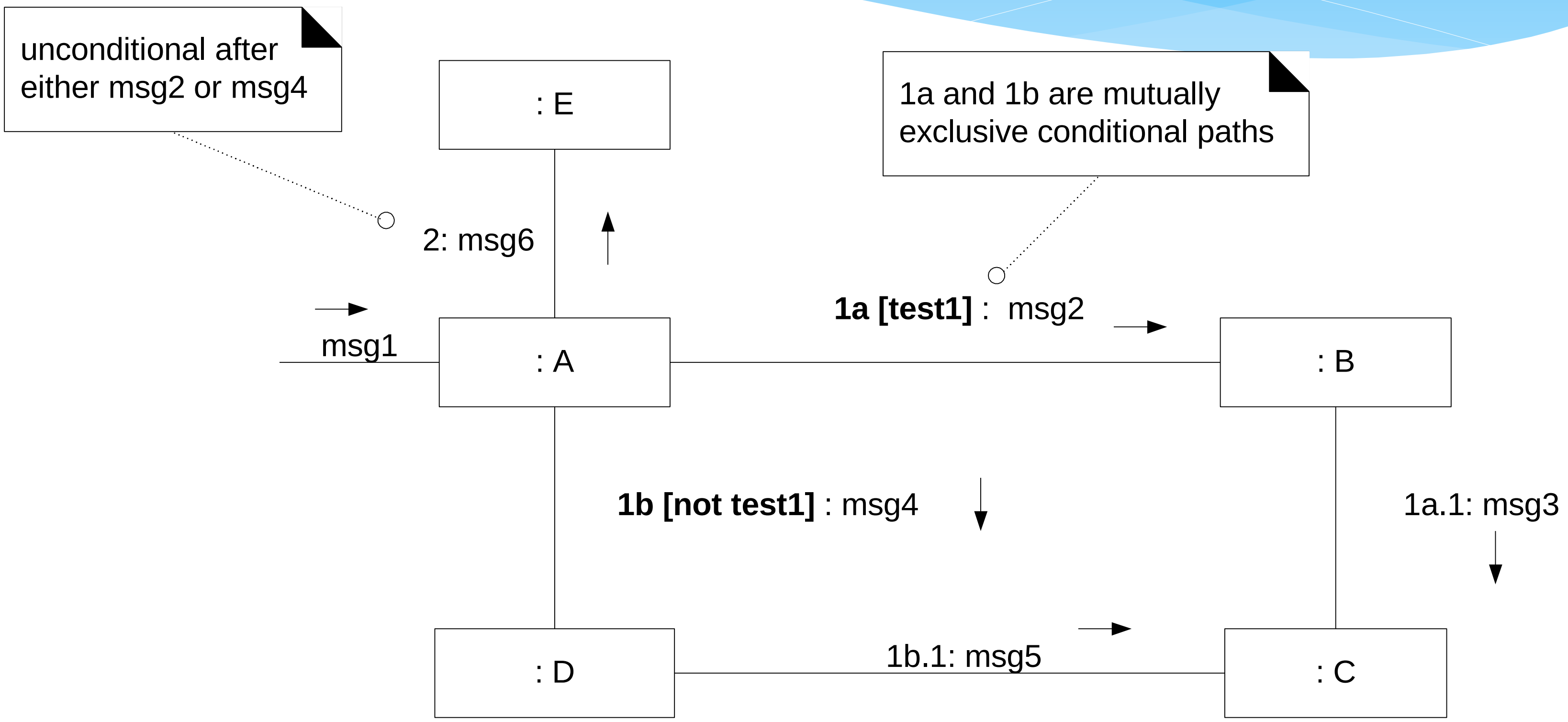


conditional message, with test

iteration is indicated with a * and an optional iteration clause following the sequence number

message1
: Foo
1 **[color = red]** : calculate
: Bar
runSimulation
: Simulator
1 * **[i = 1..n]**: num = nextInt
: Random

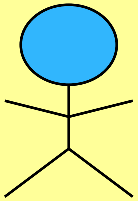
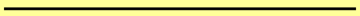
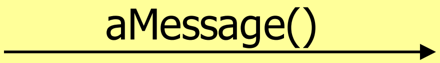
Messages – Conditional (Mutually Exclusive)



unconditional after either msg2 or msg4

1a and 1b are mutually exclusive conditional paths

Collaboration Diagram Syntax

AN ACTOR	
AN OBJECT	anObject:aClass
AN ASSOCIATION	
A MESSAGE	

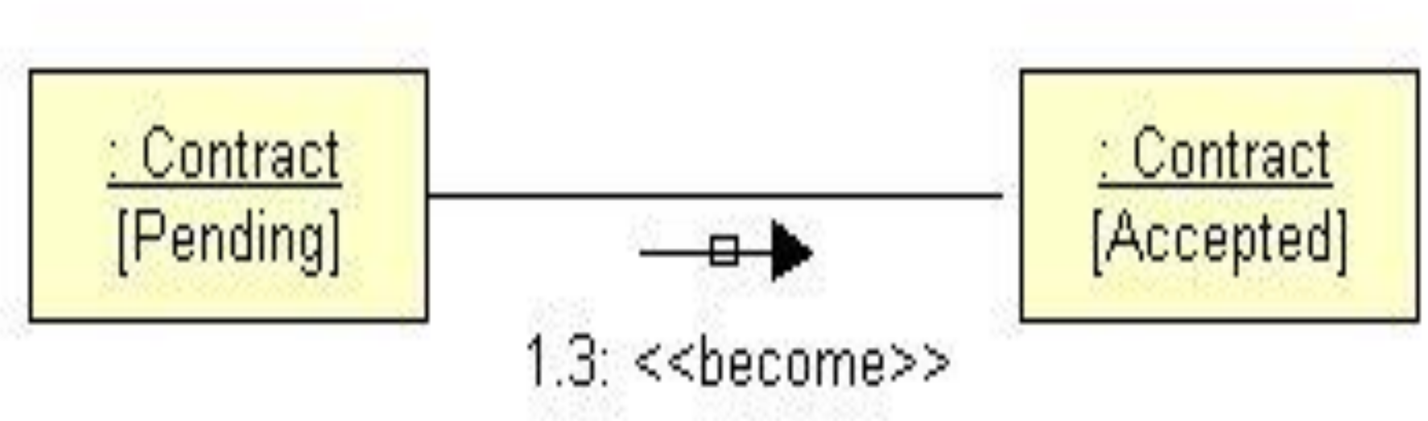
Creation and Deletion

- Unlike sequence diagrams, you don't show an object's lifeline in a collaboration diagram
- If you want to indicate the lifespan of an object in a collaboration diagram, you can use create and destroy messages to show when an object is instantiated and destroyed

Objects Changing State

- State of on object can be indicated
- Initial state is indicated with <<create>>
- If an object changes significantly during an interaction, you can add a new instance of the object to the diagram, draw a link between them and add a message with the stereotype <<become>>

Change State of an Object



Collaboration Diagram

Message Signature

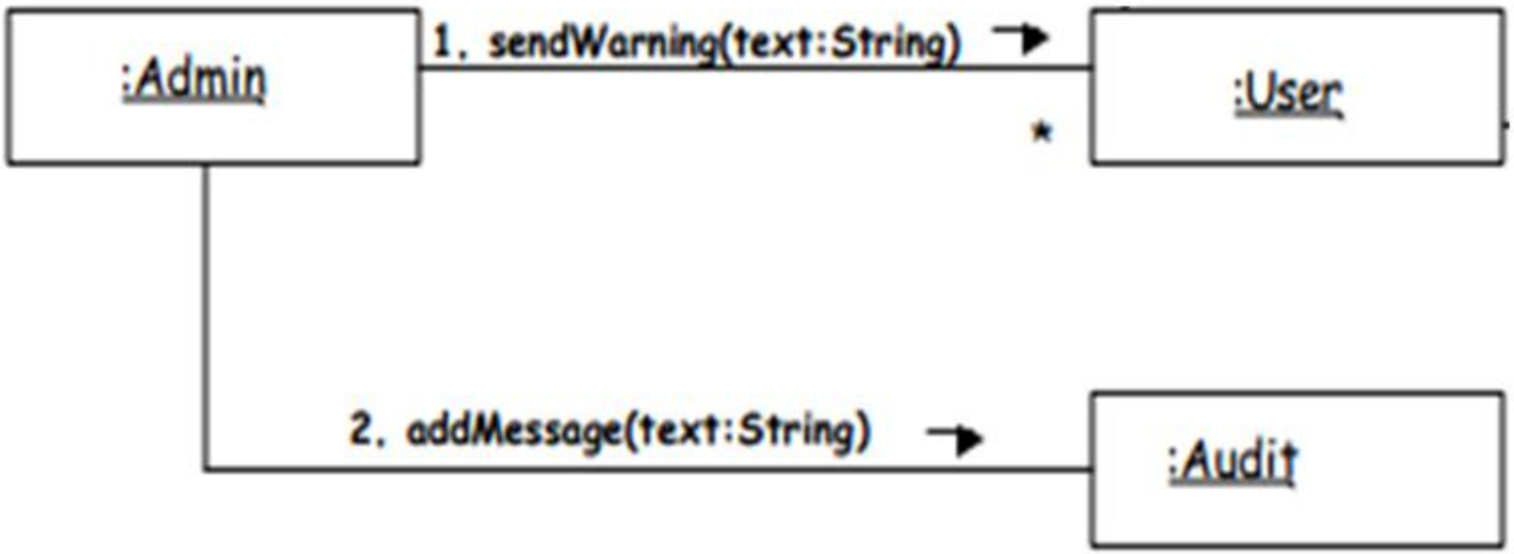
- **Guard**
  - condition applied to the message
  - in square brackets at the start of the signature
- **Sequence number**
  - numbers separated by dots, ending in a colon
- **Return value**
  - name followed by :=

Collaboration Diagram

- **Operation name**
- **Argument list**
  - names separated by commas, within round brackets
  - Types of message flows
- **Synchronous, asynchronous, simple**
  - Multiplicity of objects

Multiplicity in Collaboration Diagrams

- An admin object needs to send warning to each user logged in, and add message to the audit object (which keeps a track of messages sent)



Communication Diagrams

- The communication is implicit in a Sequence Diagram, rather than explicitly represented as in a Communication Diagram
- There is some redundancy between Communication and Sequence Diagrams
- They differently show how elements interact over time

Communication Diagrams

- Communication Diagrams show relationship between objects

- Sequence Diagrams focus on the time in which events occur

- Communication Diagrams, formerly called Collaboration Diagrams

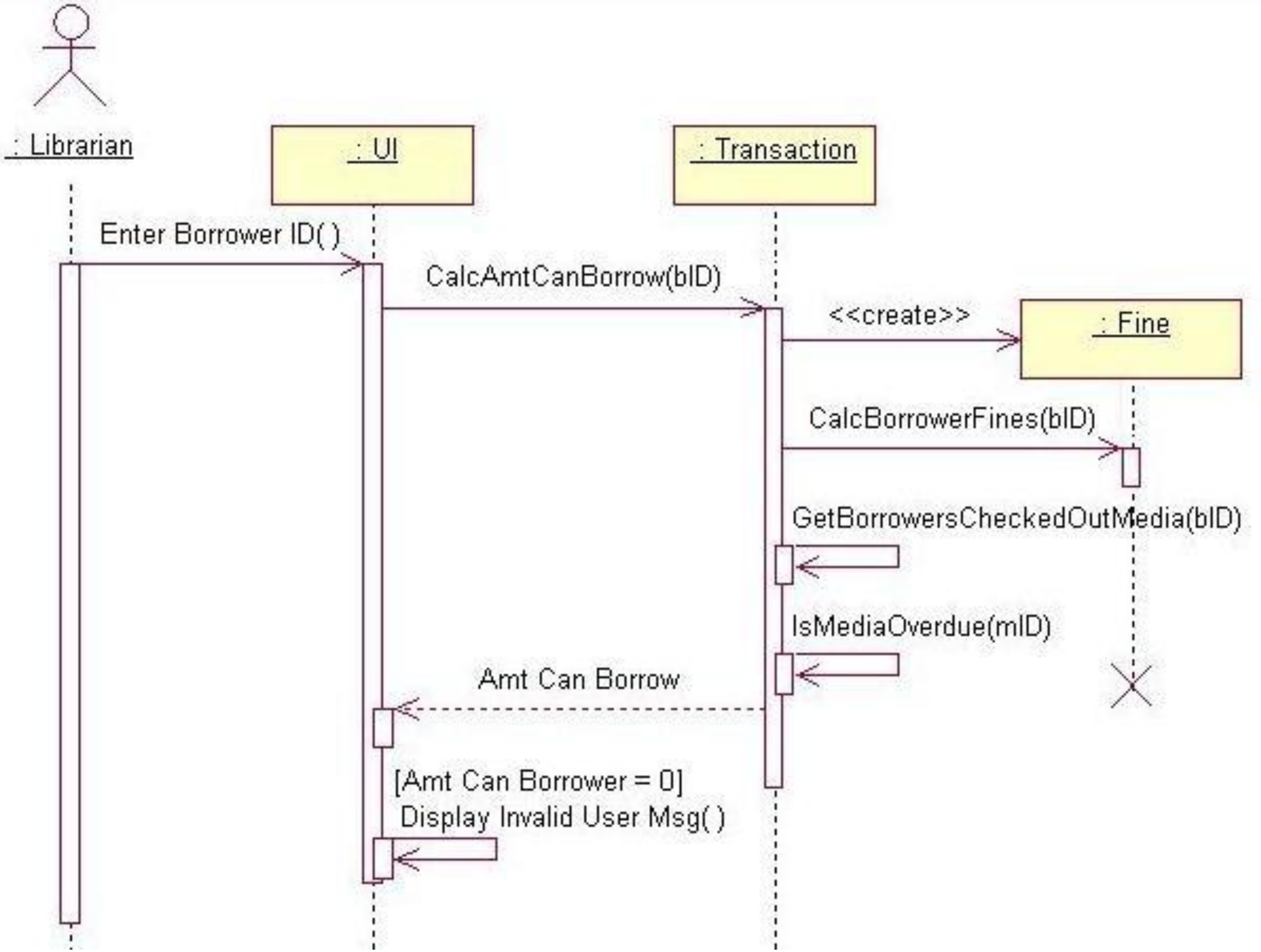
Interaction Diagram Strengths

- Communication Diagram
  - Space Economical - flexibility to add new objects in two dimensions
  - Better to illustrate complex branching, iteration, and concurrent behavior

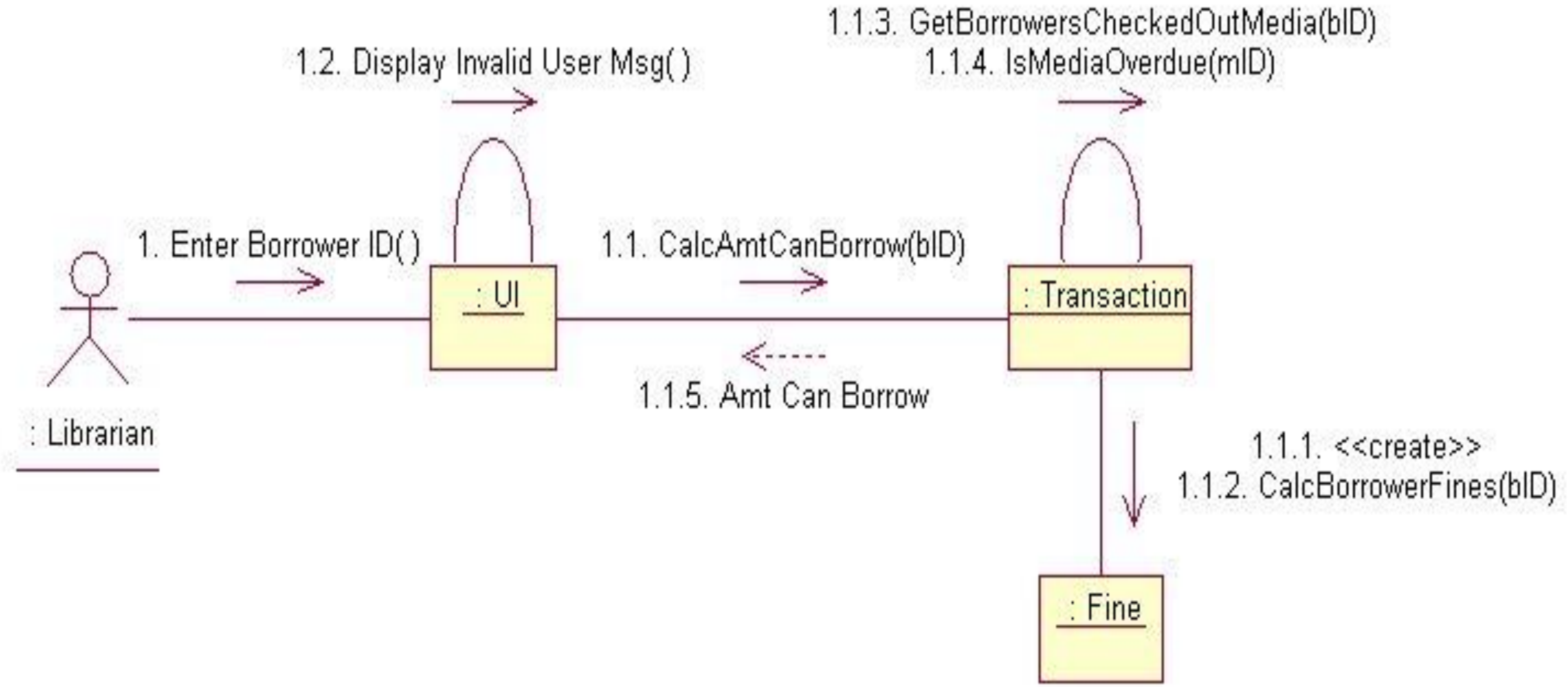
- Sequence Diagram
  - Clearly shows sequence or time ordering of messages
  - Simple notation

Interaction Diagram Weaknesses

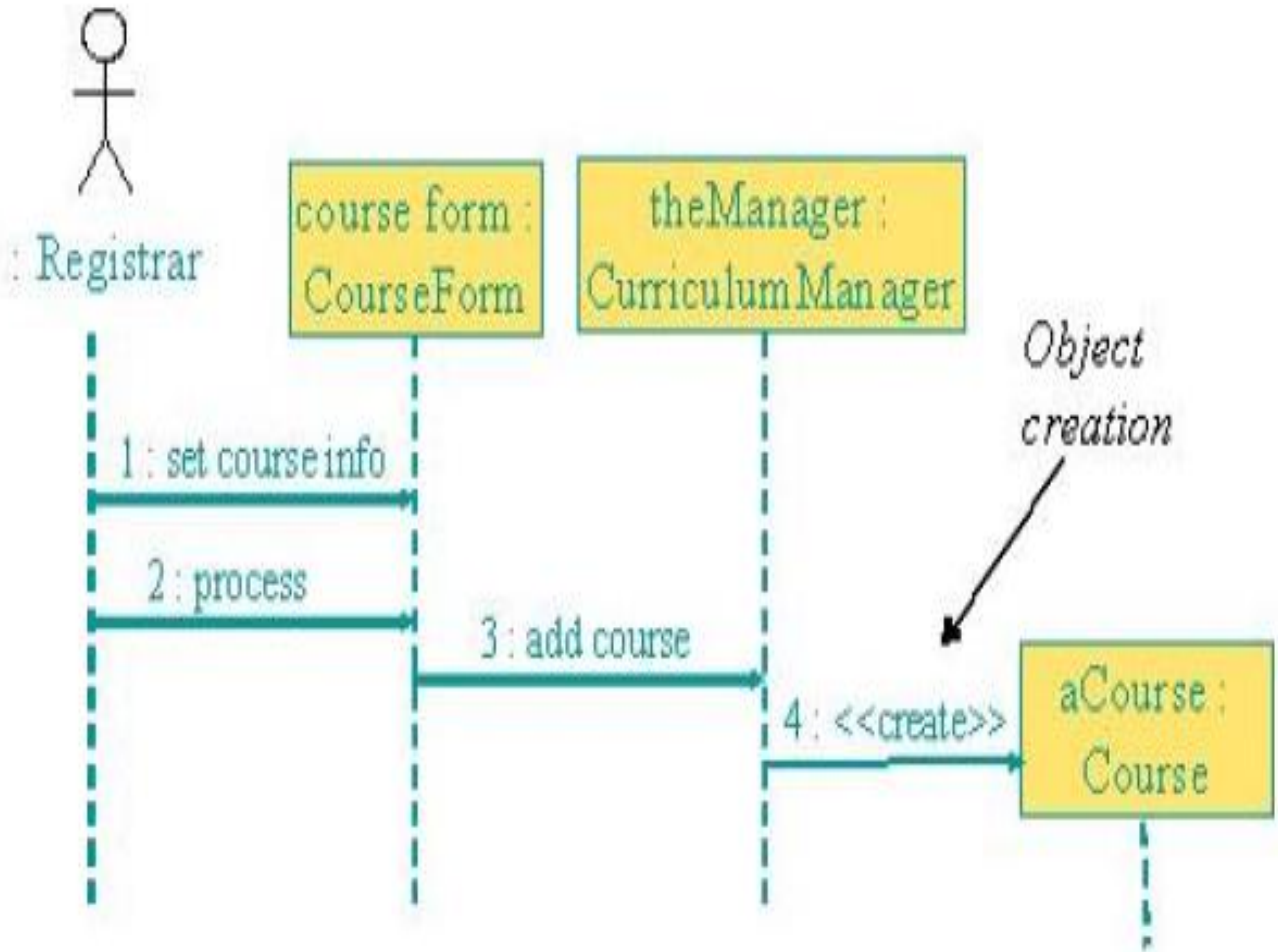
- Communication Diagram
  - Difficult to see sequence of messages
  - More complex notation
- Sequence Diagram
  - Forced to extend to the right when adding new objects; consumes horizontal space



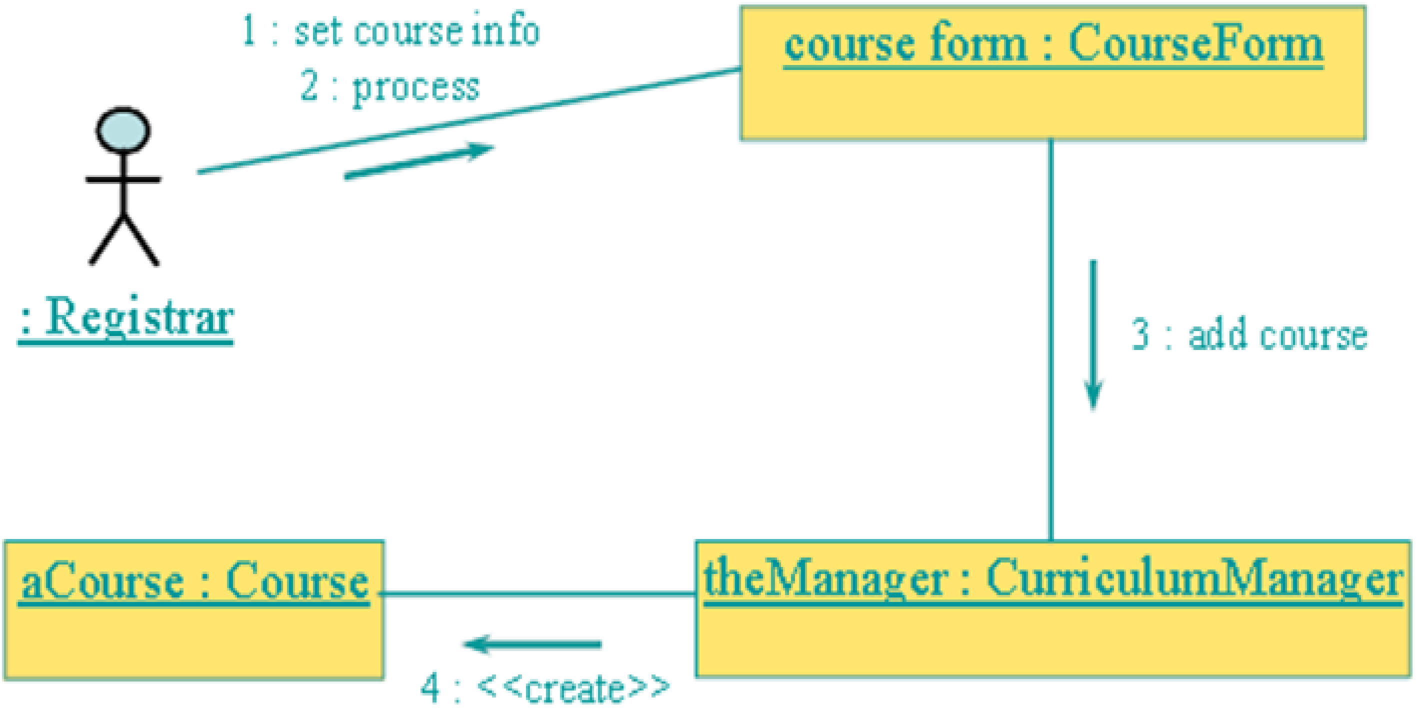
Sequence diagram is better at ‘time ordering’


1.1.3. GetBorrowersCheckedOutMedia(bID)
1.1.4. IsMediaOverdue(mID)
1.2. Display Invalid User Msg( )
1. Enter Borrower ID( )
1.1. CalcAmtCanBorrow(bID)
: UI
: Transaction
1.1.5. Amt Can Borrow
: Librarian
1.1.1. <<create>>
1.1.2. CalcBorrowerFines(bID)
: Fine

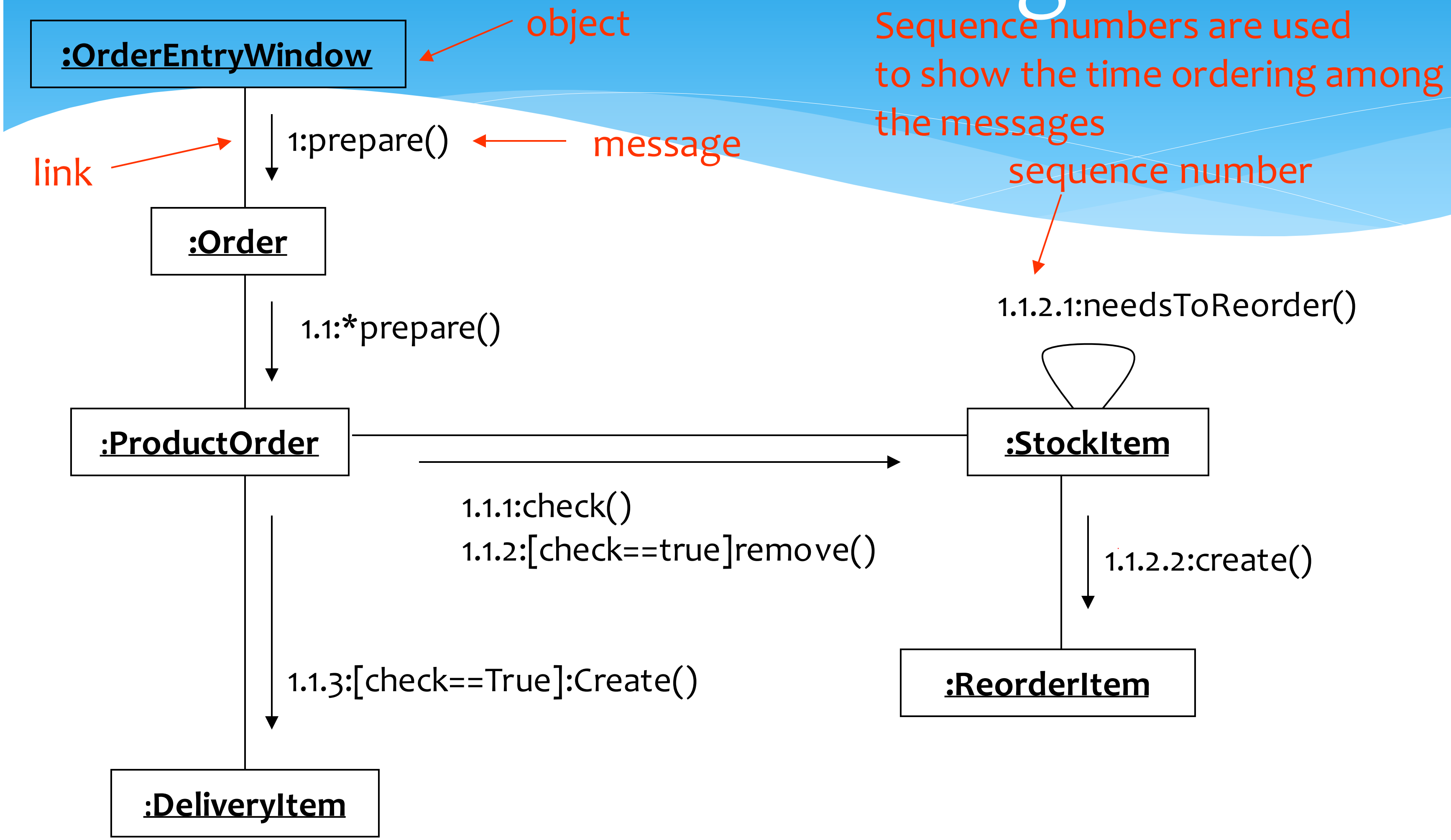
Collaboration diagram is better at showing the relationship between objects


: Registrar
course form : CourseForm
theManager : CurriculumManager
Object creation
1 : set course info
2 : process
3 : add course
4 : <<create>>
aCourse : Course

Example


1 : set course info
2 : process
course form : CourseForm
: Registrar
3 : add course
aCourse : Course
theManager : CurriculumManager
4 : <<create>>

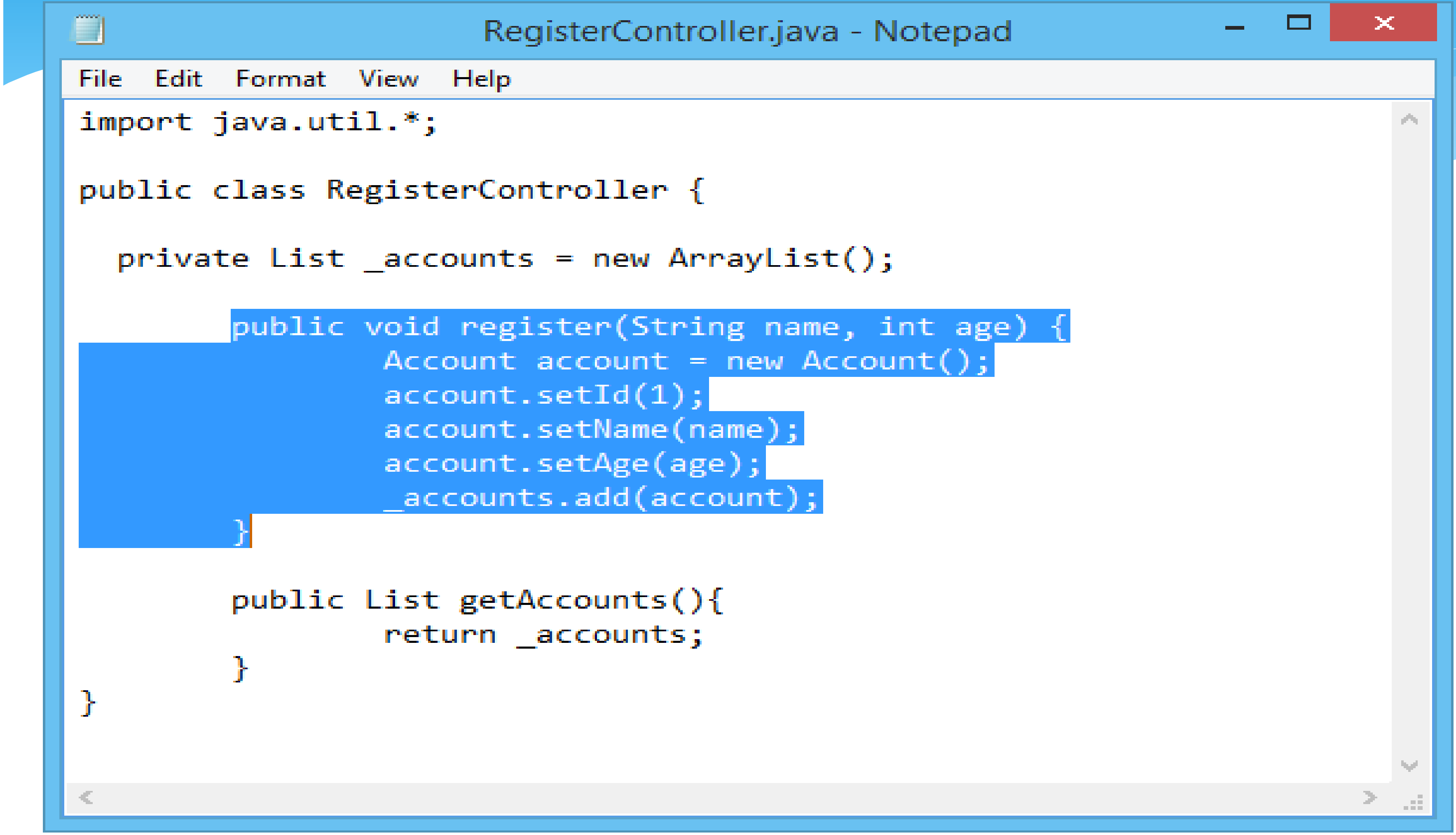
Collaboration Diagram



Class Activity

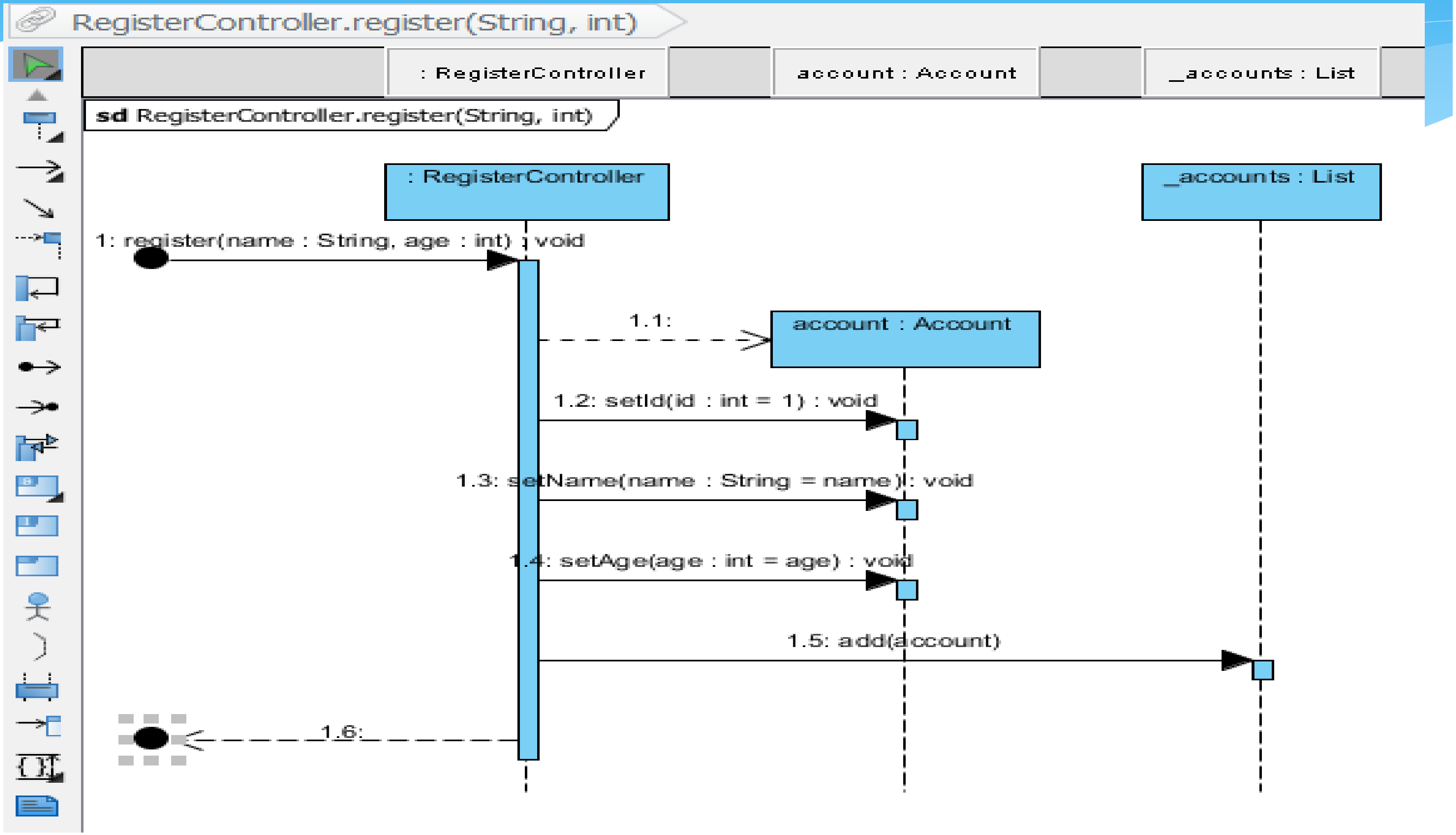
- Consider the software that controls a very simple cellular telephone. Such a phone has buttons for dialing digits, and a “send” button for initiating a call. It has “dialer” hardware and software that gathers the digits to be dialed and emits the appropriate tones. It has a cellular radio that deals with the connection to the cellular network using a dialer. It has a microphone, a speaker, and a display

Your Turn



```
import java.util.*;

public class RegisterController {

  private List _accounts = new ArrayList();

        public void register(String name, int age) {
                Account account = new Account();
                account.setId(1);
                account.setName(name);
                account.setAge(age);
                _accounts.add(account);
        }

        public List getAccounts(){
                return _accounts;
        }
}
```

Solution



RegisterController.register(String, int)
: RegisterController
account : Account
_accounts : List
sd RegisterController.register(String, int)
: RegisterController
_accounts : List
1: register(name : String, age : int) : void
1.1:
account : Account
1.2: setId(id : int = 1) : void
1.3: setName(name : String = name) : void
1.4: setAge(age : int = age) : void
1.5: add(account)
1.6:

- The train details like train no, scheduled time, expected time to arrive, scheduled time for departure, expected time to start, platform number and current number is given as input to calculate the time delay by an administrator.
- The entered details are updated and verified with the database and administrator calculate the time delay and also checks whether there is any clash with the next train timing
- Based on the delay and arrival of the next train the platform assignment is done.

Example for Withdraw

- Joe withdraws $20 from the ATM (flow of events)
  - The process begins when Joe inserts his card into the card reader. The card reader reads the number on Joe's card, then tells the ATM screen to initialize itself
  - The ATM verifies the card against account and prompts Joe for his PIN.
  - Joe enters PIN.
  - Joe's PIN is validated and the ATM prompts him for a transaction
  - Joe selects Withdraw Money
  - The ATM prompts Joe for an amount.
  - Joe enters $ 20.
  - The ATM verifies that Joe's account has sufficient funds and subtracts $ 20 from his account.
  - The ATM dispenses $ 20 and ejects Joe's card

- For customer communication of online Book shop starts with find books that iteratively repeats the message number of times. Client searches inventory of books, and if he/she is interested in some book he/she can view description of the book. If client decides to buy, he/she can add book to the shopping cart.
- Customer can also checkout from the online book store. Initially he/she will get the books from the shopping cart and make the order. Online Bookstore will make sure that shopping cart is not empty before placing an order. After the order is paced, online Bookstore update the inventory.

Practice Exercise – DVD Rental (SSD)

- UC1: Customer rents videos
  - Preconditions: Customer has a membership, has selected videos they want, and made system aware of their choices.
  - Actor: Customer (self-service/remote), or store associate (in store)
- Main flow:
  1. Actor indicates to rent first item (e.g., clicking "rent" on a networked device, or scanning it physically in a store)
  2. System verifies immediate availability, and waits to make next option
  3. Actor indicates they are done selecting
  4. System shows total, prompts for payment
  5. Actor selects method of payment, entering additional data if needed (e.g., credit card number)
  6. System verifies the payment has gone through, schedules the goods for rental (e.g., sets up a window to click on to view the video remotely, or tells the store clerk where to find the DVD)