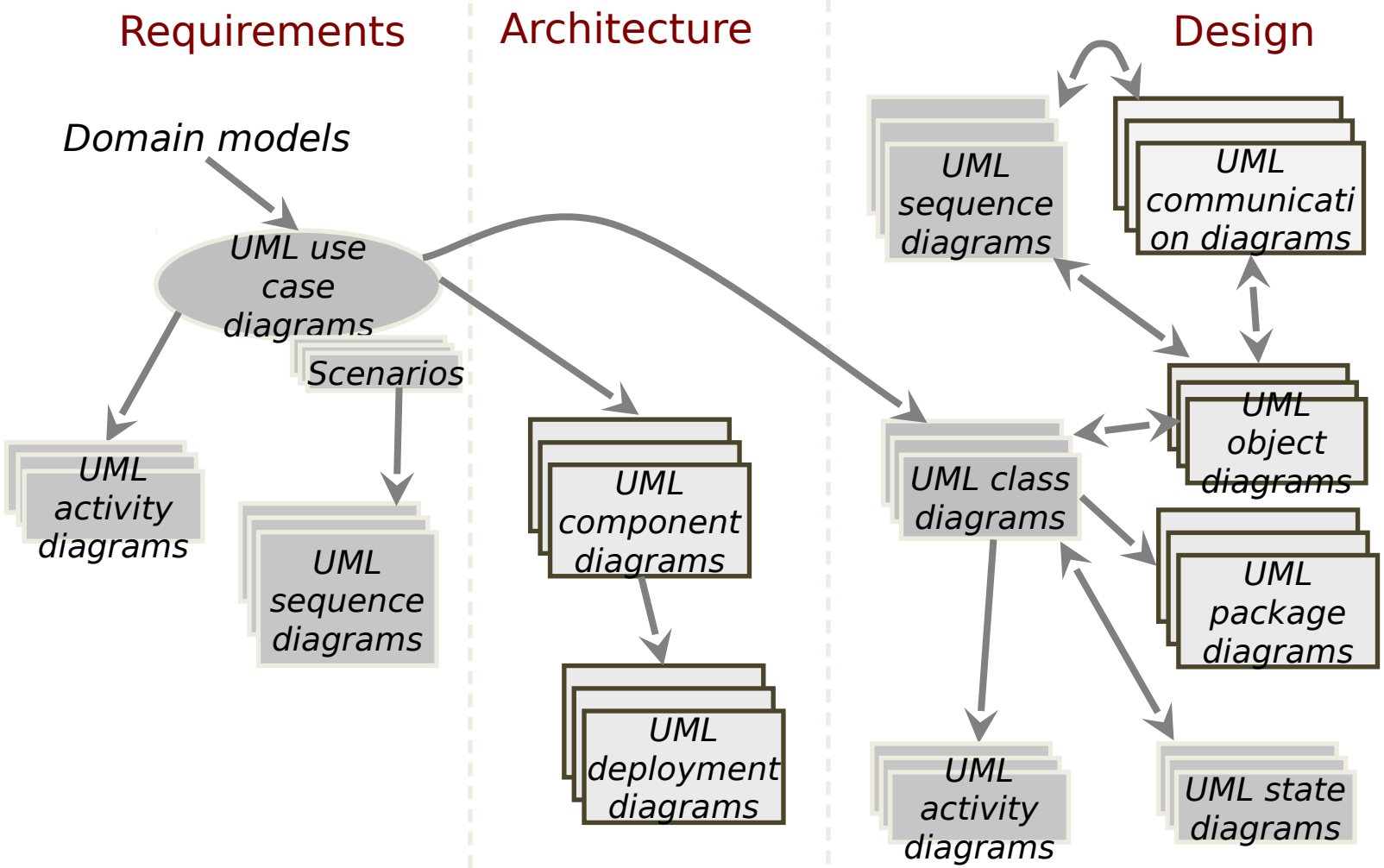
Software Engineering

Rogério de Lemos

- Sequence diagrams

OO Designs in the UML

How UML is used in the development process



Lecture Outline

Some of the topics

- Static/dynamic aspect of models
- Interaction diagrams
- Sequence diagrams basics
- Sequence diagrams and use cases
- Example

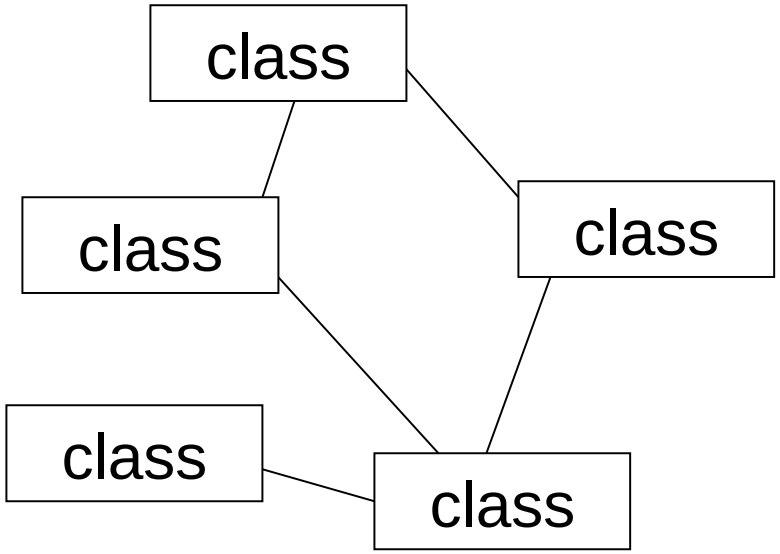
Types of UML Diagrams

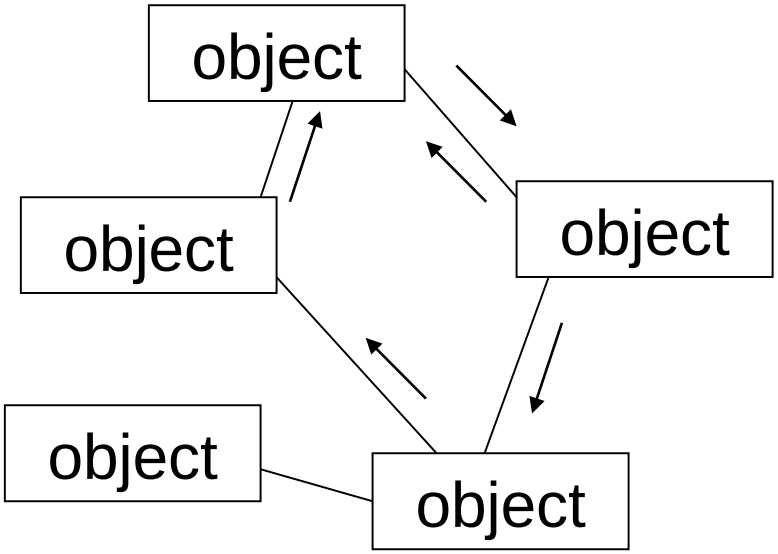
Static - emphasizes structure

- Structural description of the participants

Dynamic – emphasizes behaviour

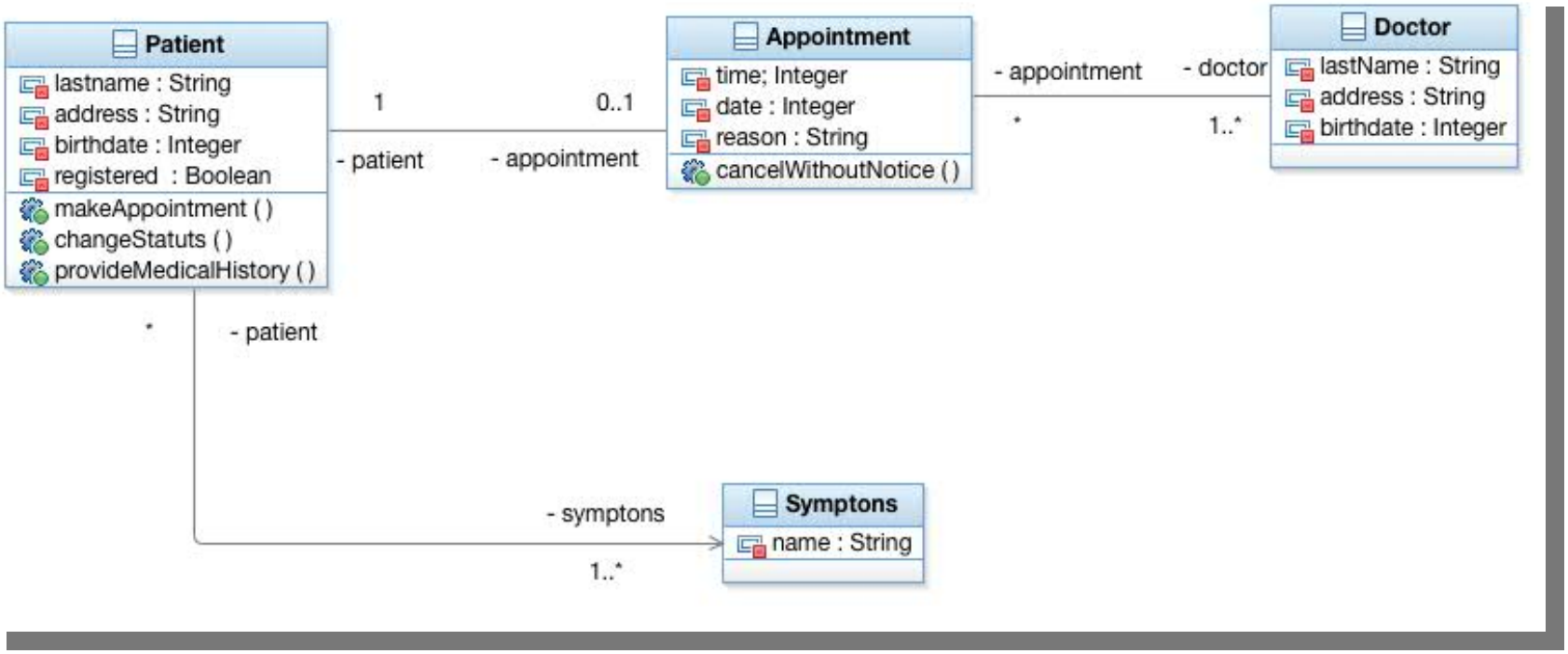
- Interaction between participants





Example: UML Class Diagram

A quite simple medical practice

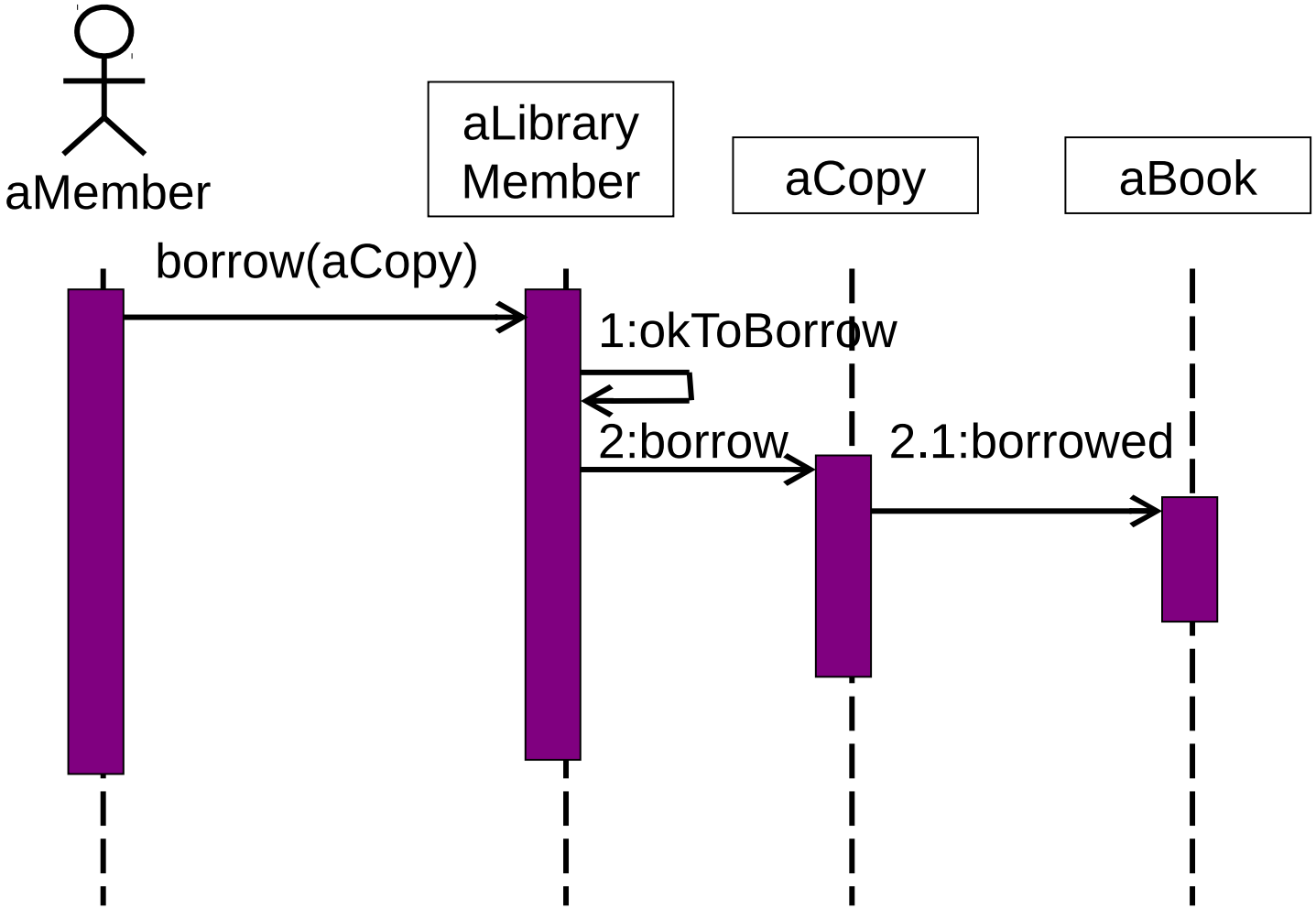


Interaction Diagrams

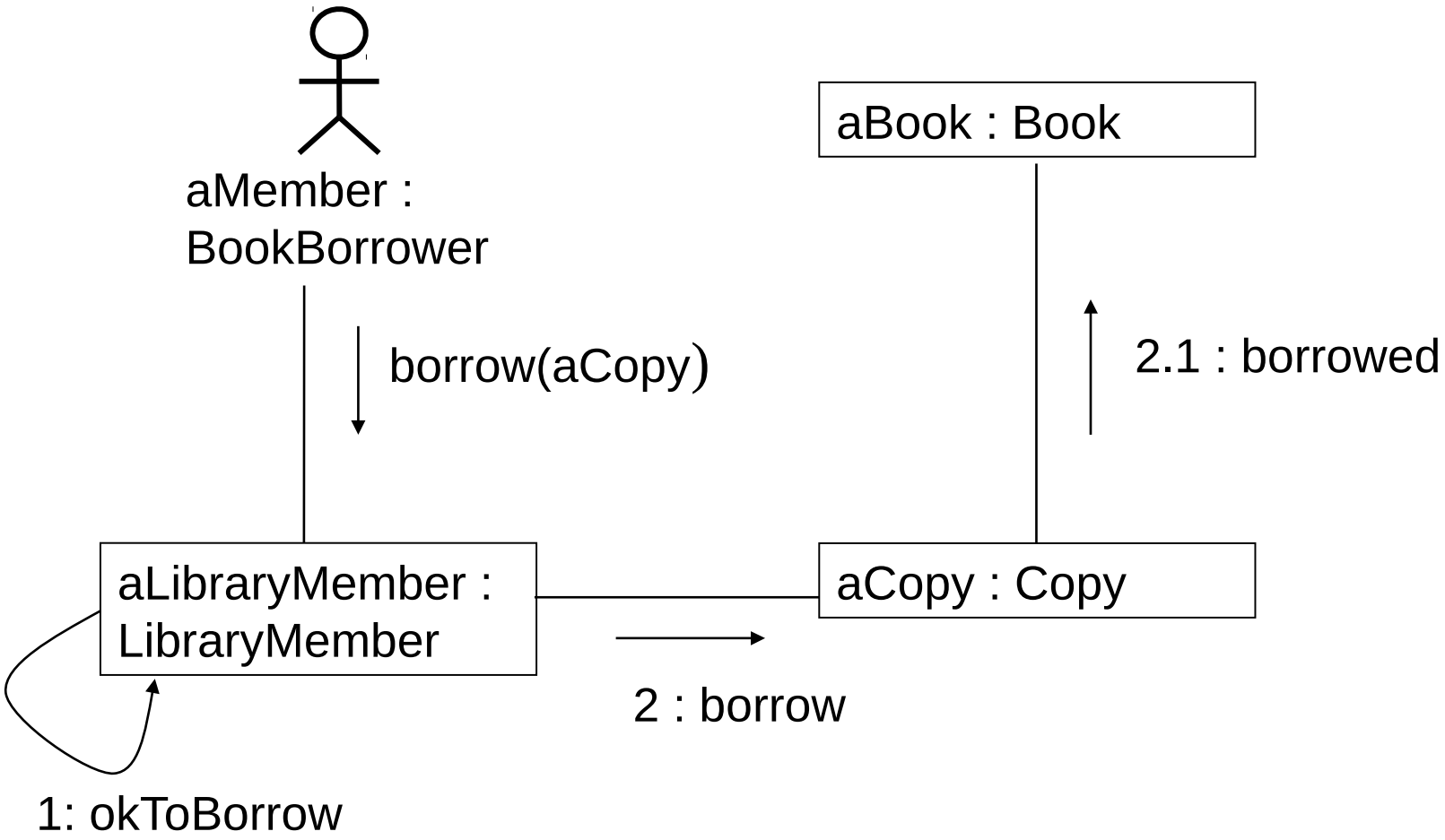
Interaction diagrams – describe how groups of objects collaborate in some behaviour

- **sequence diagrams** - explicit sequence of communications for complex scenarios
  - emphasis the time ordering of messages
- **collaboration diagrams** - emphasises the structural organisation of the objects that send and receive messages

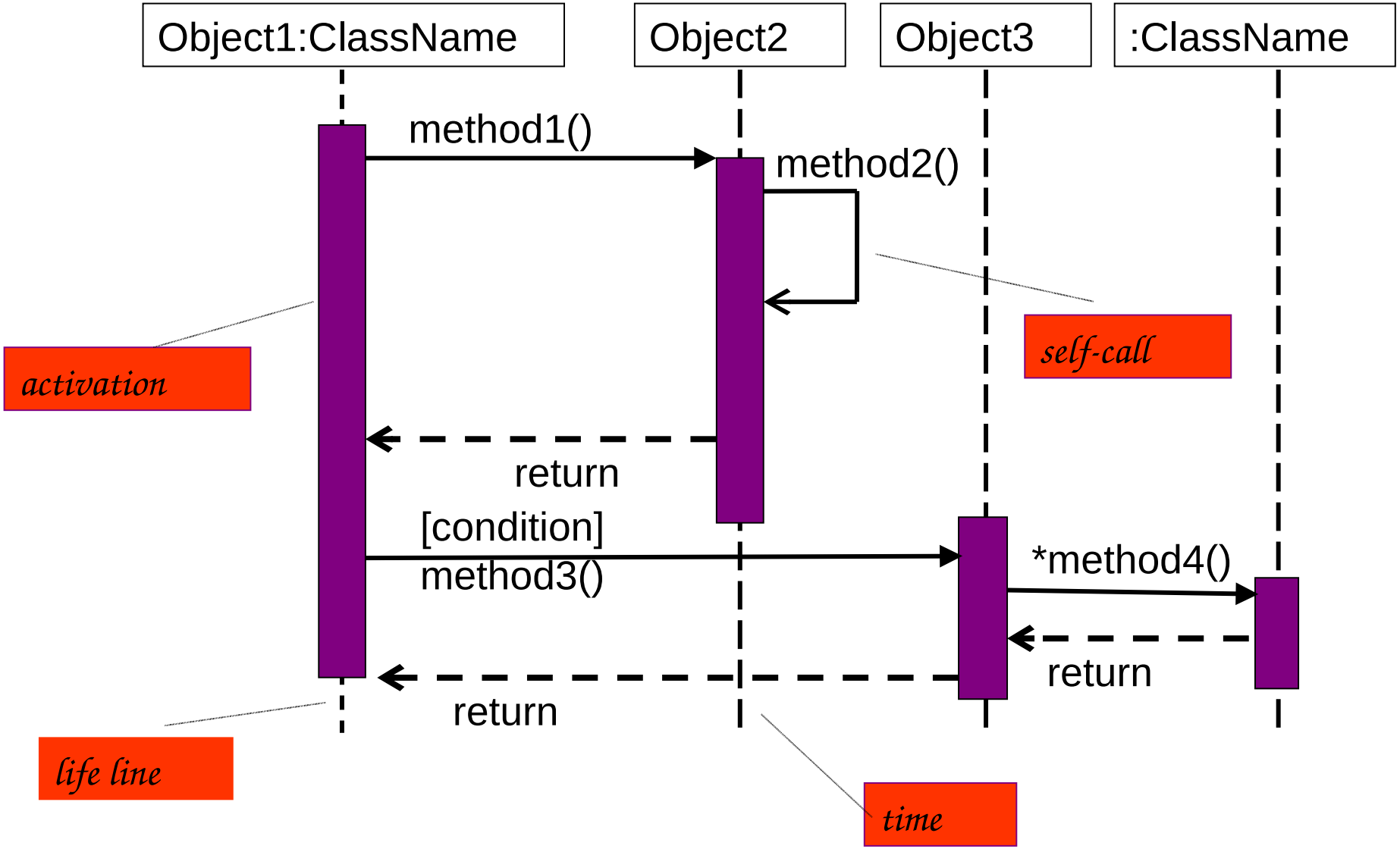
Sequence Diagrams



Collaboration Diagrams



Notation of Sequence Diagrams



Notation of Sequence Diagrams

Life line

- object's life during interaction

Self-call

- message sent to itself

Condition syntax

- [some condition] method

Iteration marker

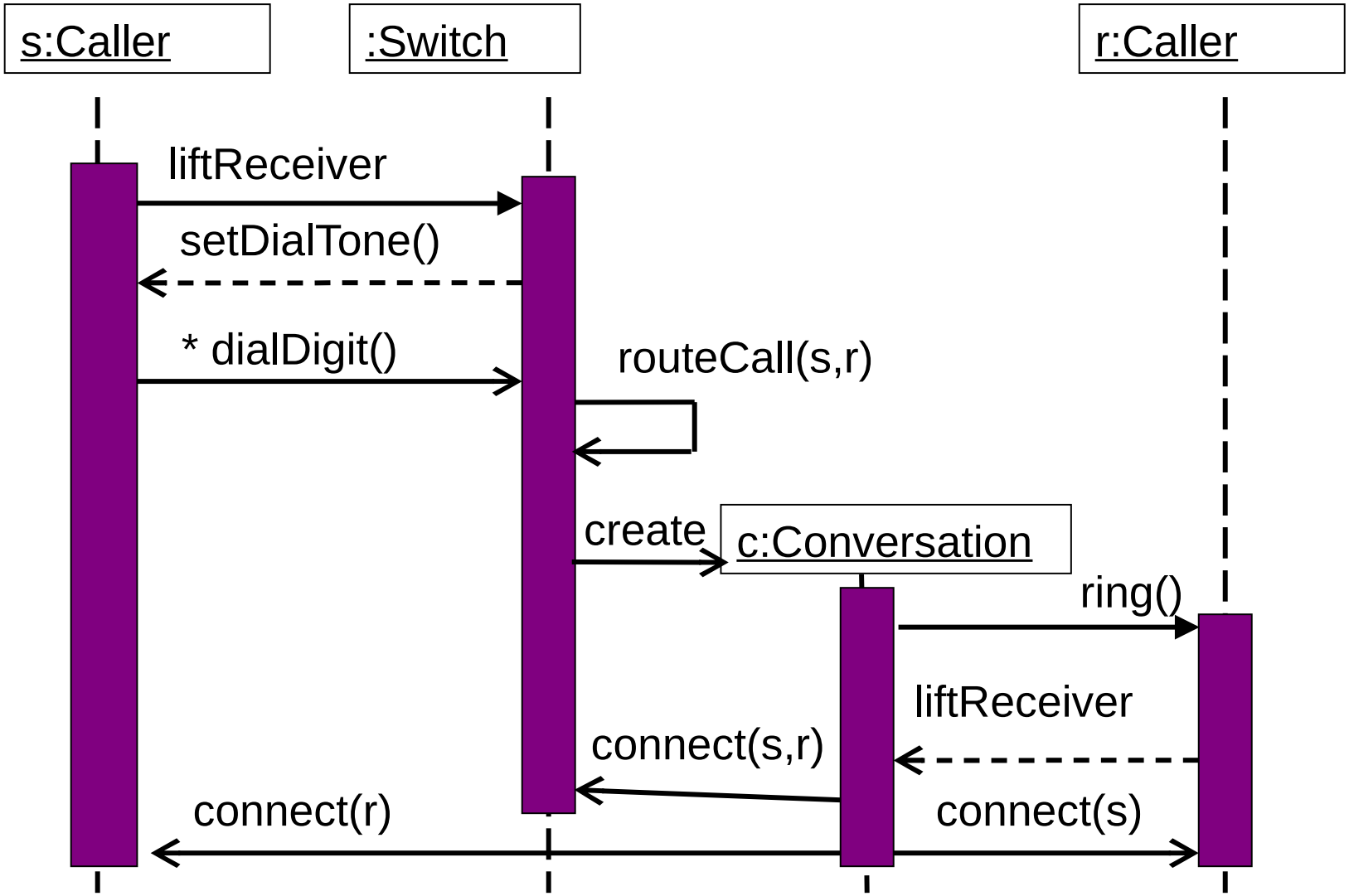
- message sent many times

However other UML diagrams such as activity diagrams are better for modelling *loops, if* …

Example: Two-Party Phone Call

- One Caller picks up the phone which results in dispatching a signal *liftReceiver* to the Switch.
- In turn, the Switch calls *setDialTone* on the Caller, and the Caller iterates on the message *dialDigit*
- The Switch then calls itself with the message *routecall* which is to call the other caller
- It then creates a Conversation, to which it delegates the rest of the work. The Conversation rings the other Caller, who asynchronously sends the message *liftreceiver*
- The Conversation object then tells the Switch to *connect* the call, then tells both Caller objects to *connect*, after which they may exchange information

Example: Two-Party Phone Call



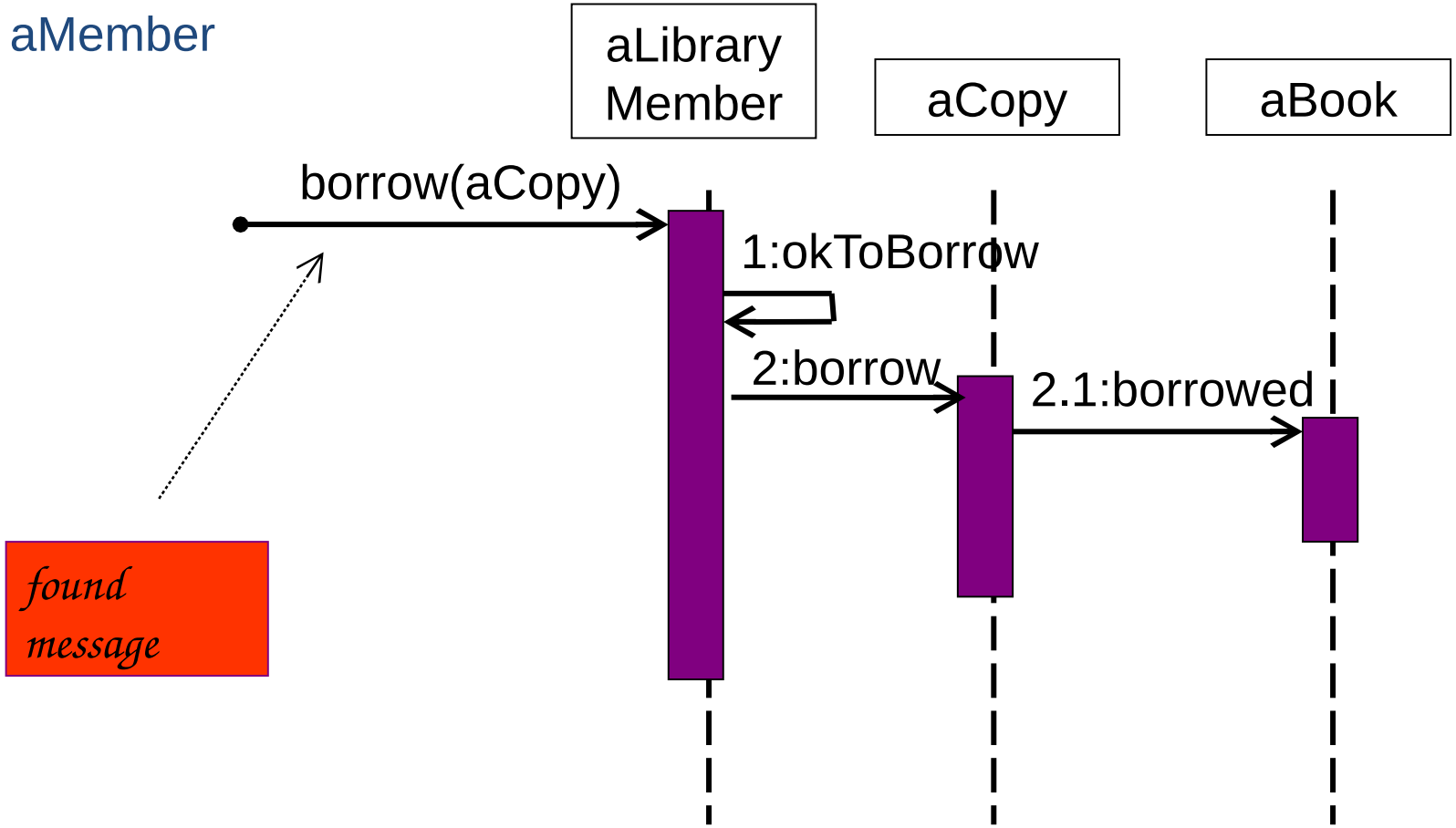
Notation of Sequence Diagrams

Arrows

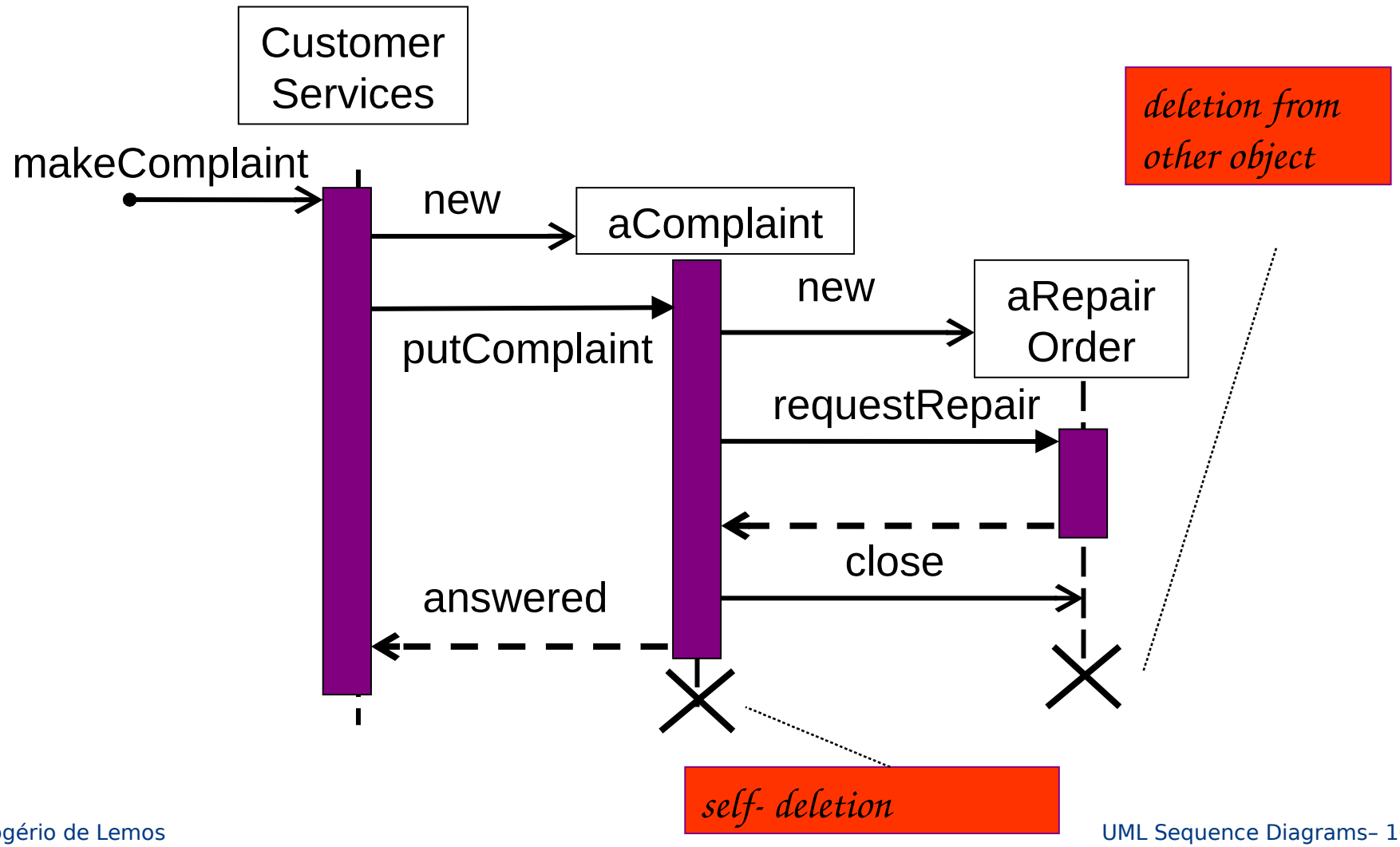
- synchronous calls: ⟶
  - it must wait until the message is done, such as invoking a subroutine
    - procedure call, invoke of methods nested flow control
      - inner nested sequence completes first
- asynchronous calls: ⟶
  - it can continue processing and does not have to wait for a response
    - no nested control, sender dispatches the signal
    - Example: create new thread, create new object, sending signals, concurrent communication with a running thread

Sequence Diagrams
(asynchronous calls)

- Instead of actor aMember



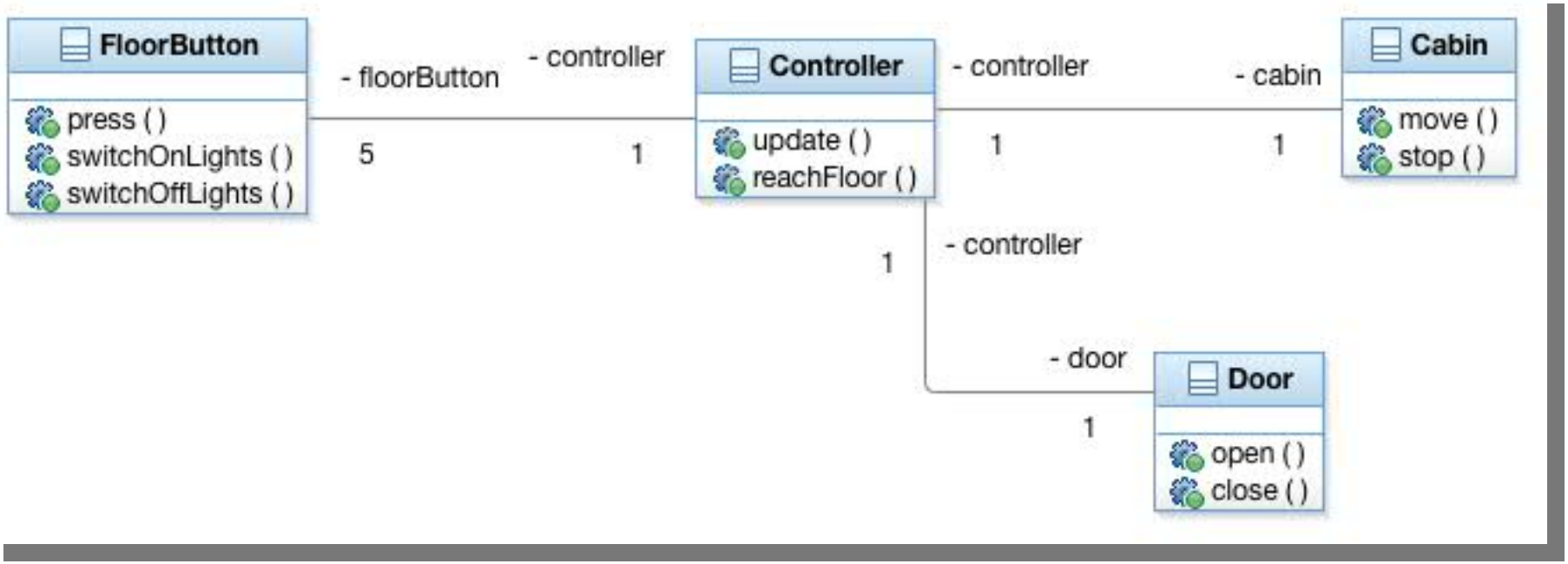
Sequence Diagrams
(synchronous calls)



From UML Class to Sequence Diagrams

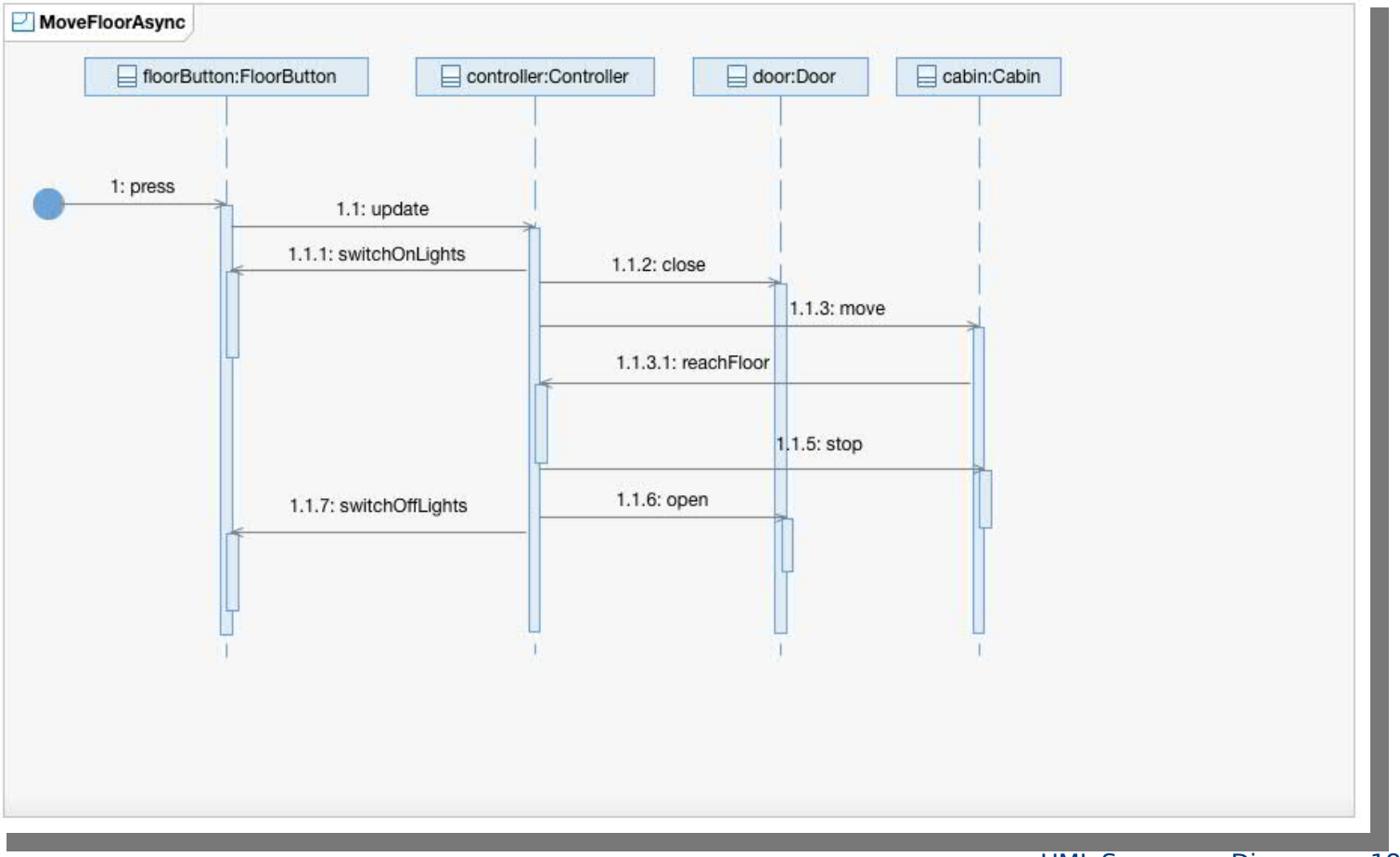
Consider the following UML class diagram of a lift

- what operation comes first?
- what is the sequence of operations?



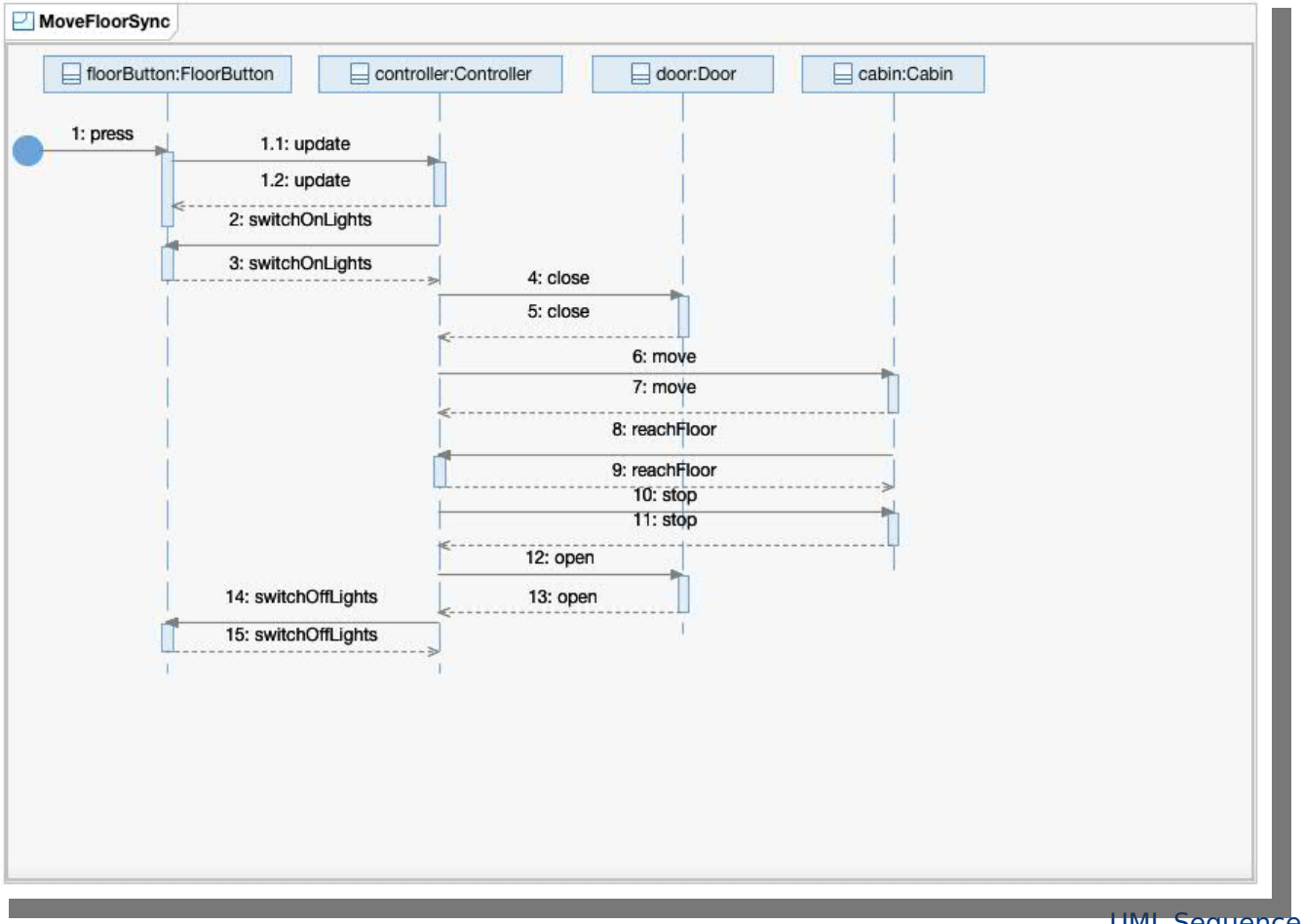
From UML Class to Sequence Diagrams

This sequence diagram represents a possible behaviour of calling a lift using asynchronous messaging



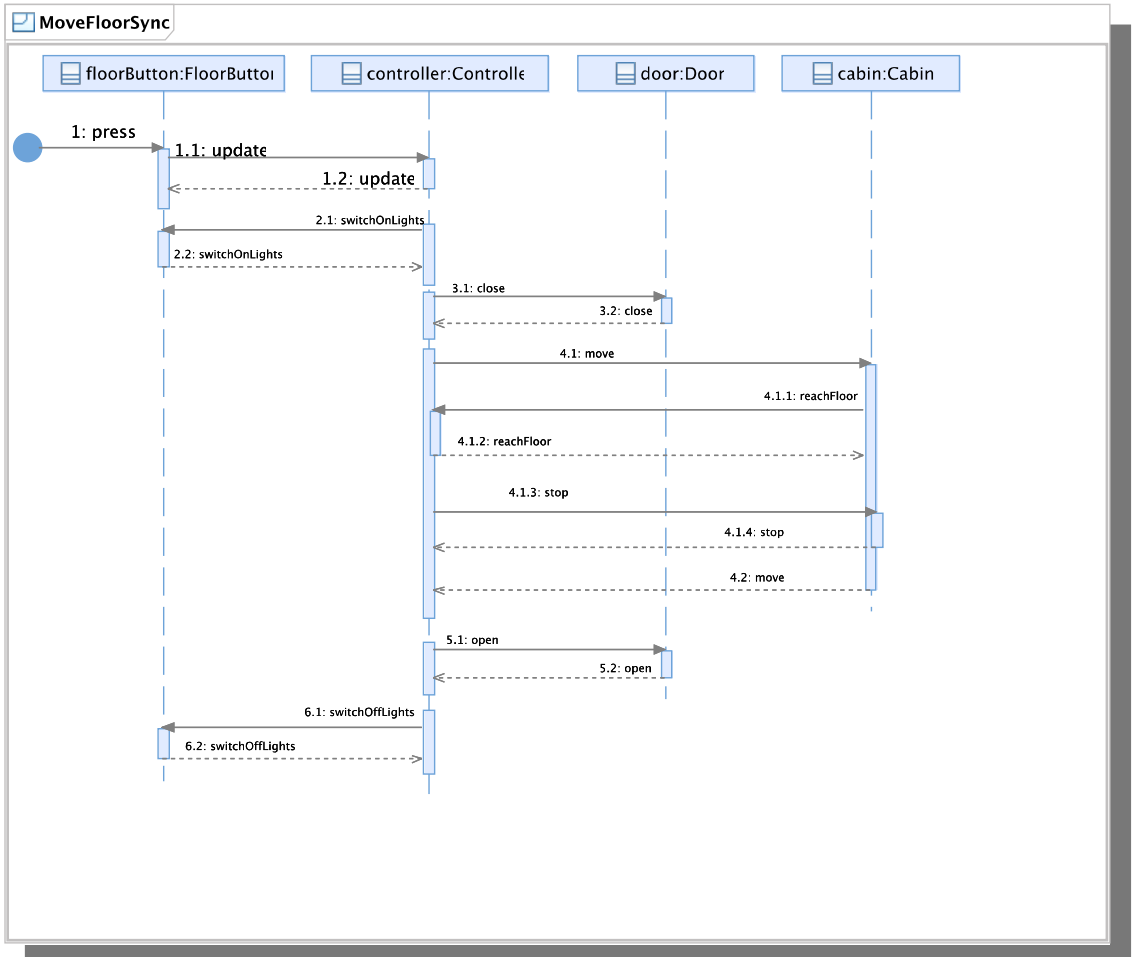
From UML Class to Sequence Diagrams

This sequence diagram represents a possible behaviour of calling a lift using synchronous messaging



From UML Class to Sequence Diagrams

A much better sequence diagram represents a possible behaviour of calling a lift using synchronous messaging



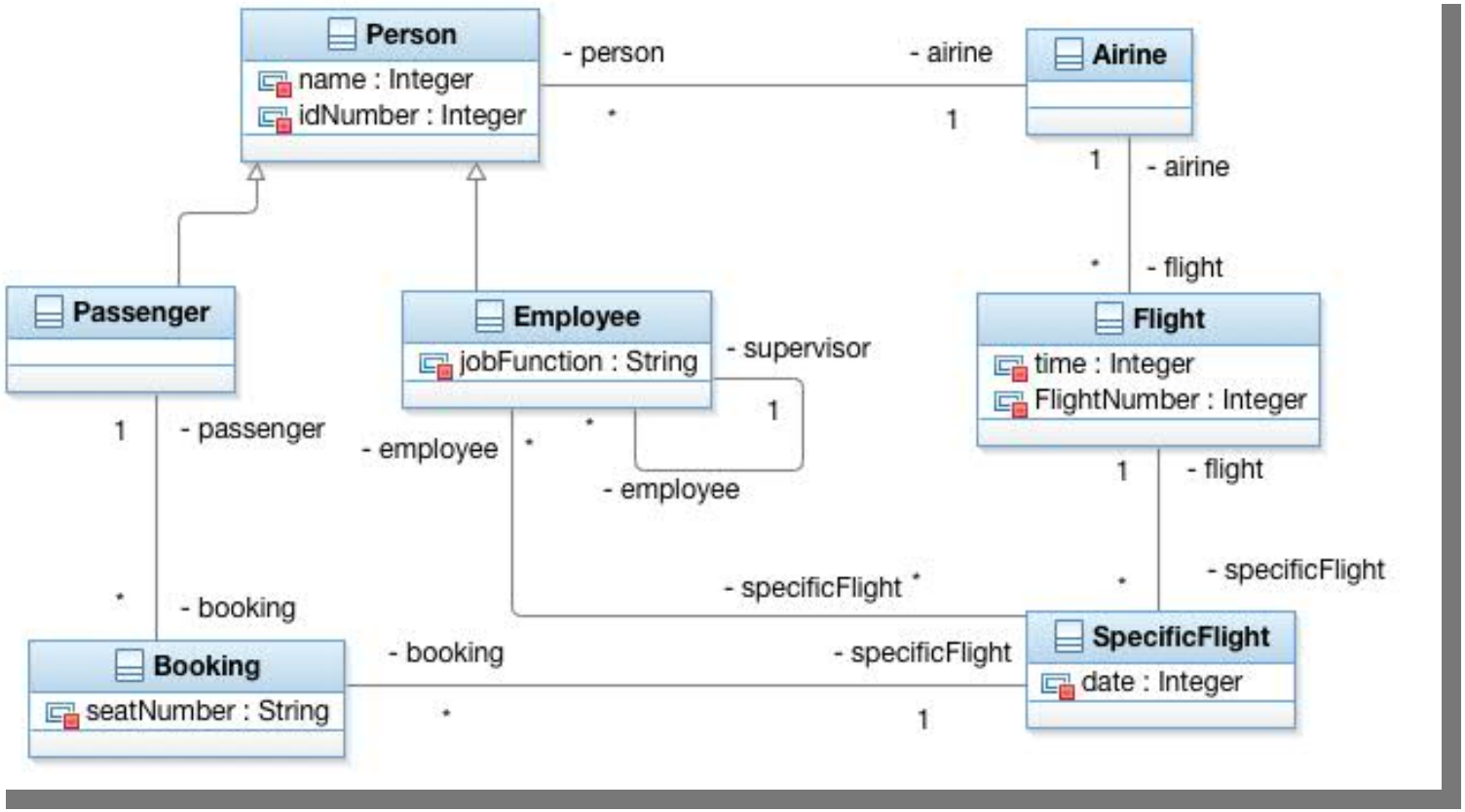
Example Domain Modelling

Airline reservation system

- Koffee Airlines runs a sightseeing flights from Java Valley, the capital of Koffee. The reservation system keeps track of passengers who will be flying in specific seats on various flights, as well as people who will form the crew. For the crew, the system needs to track what everyone does, and who supervises whom. Koffee Airlines runs several daily numbered flights on a regular schedule. Koffee Airlines expects to expand in the future, therefore the system needs to be flexible; in particular will be adding a frequent-flier program.

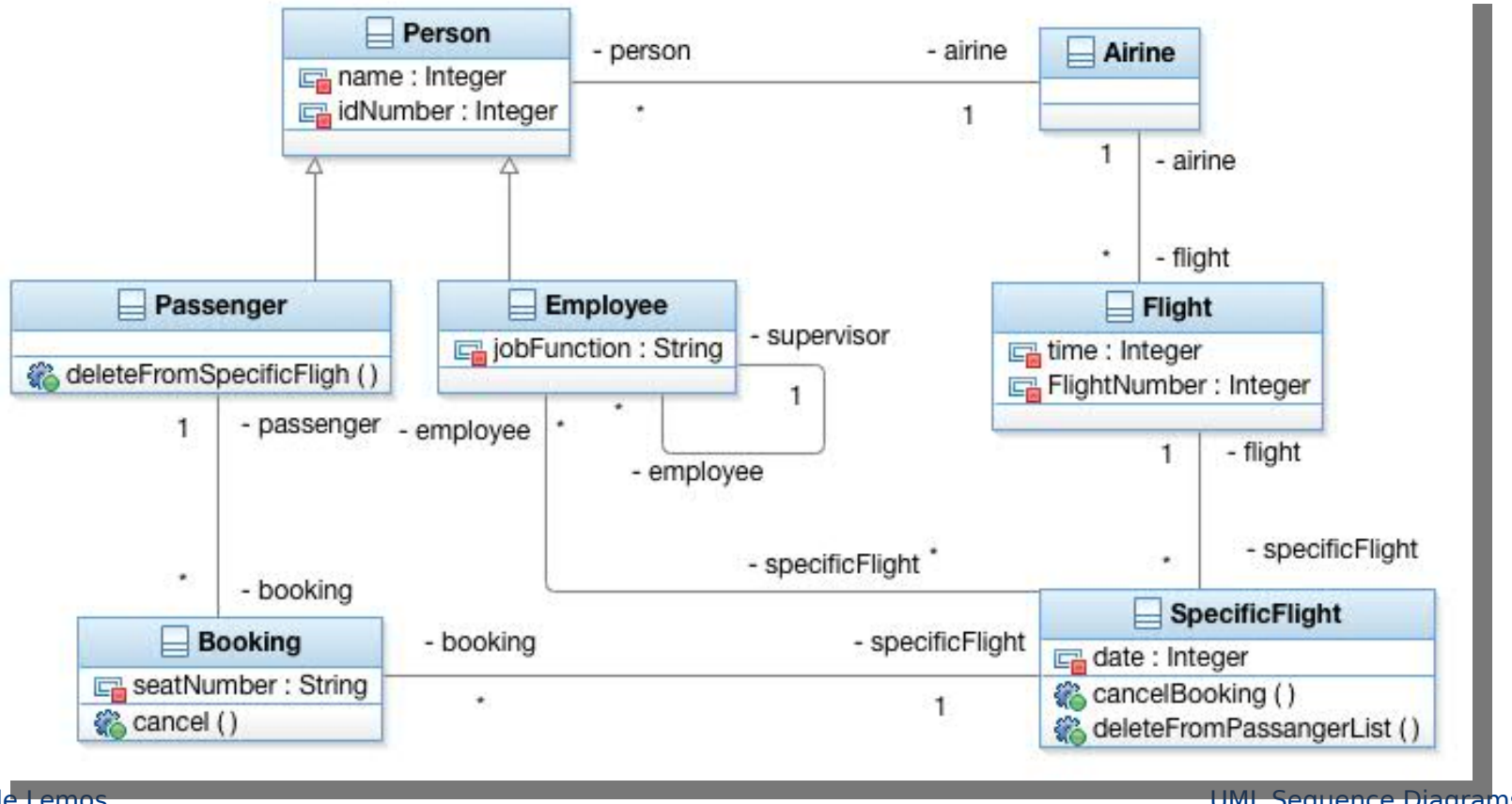
From UML Class to Sequence Diagrams

Does the following UML class diagram representing the Koffee Airline reservation system is able to represent cancel a booking?



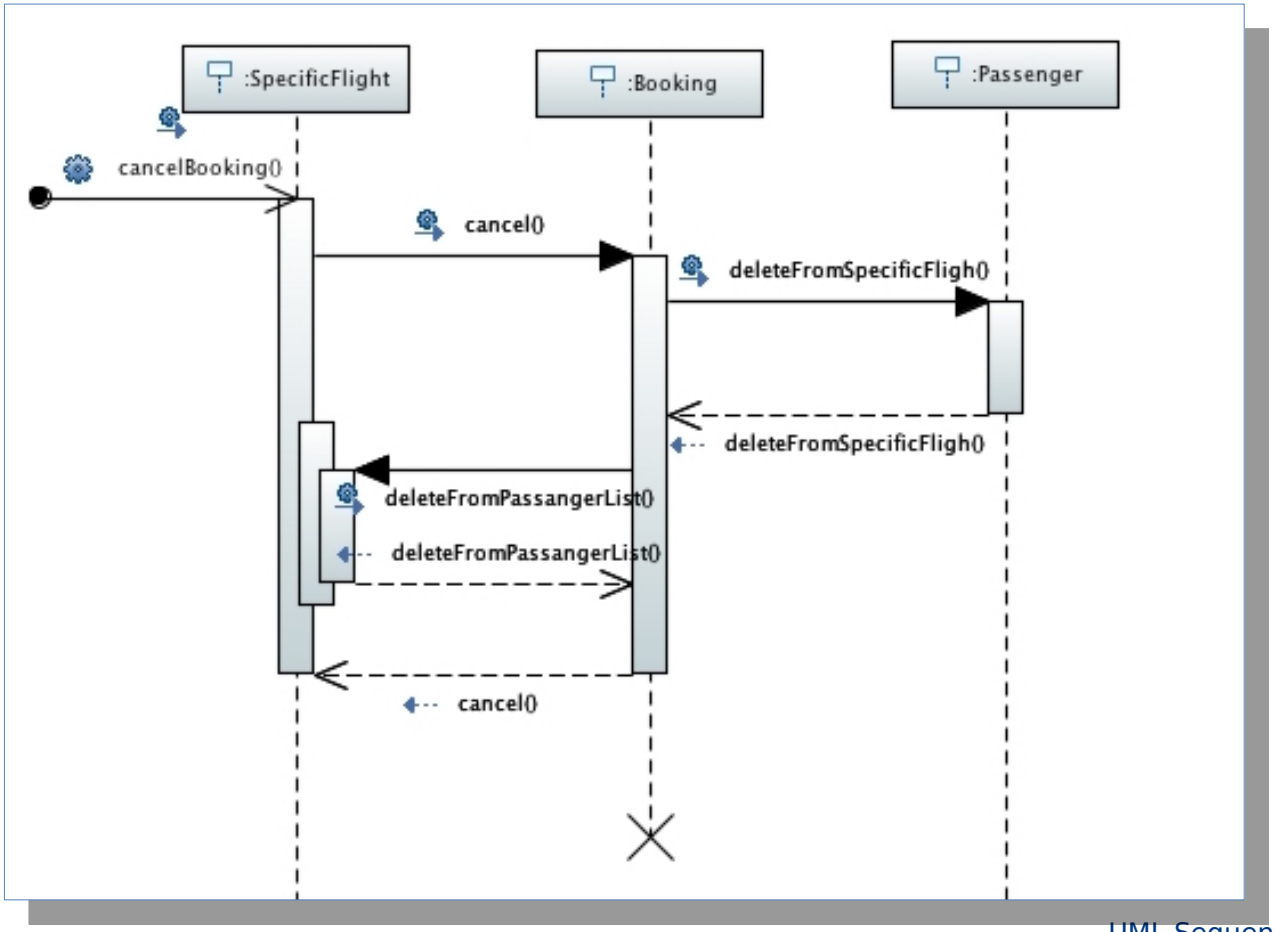
From UML Class to Sequence Diagrams

The UML class diagram of Koffee Airline reservation system with additional operations for capturing cancel a booking



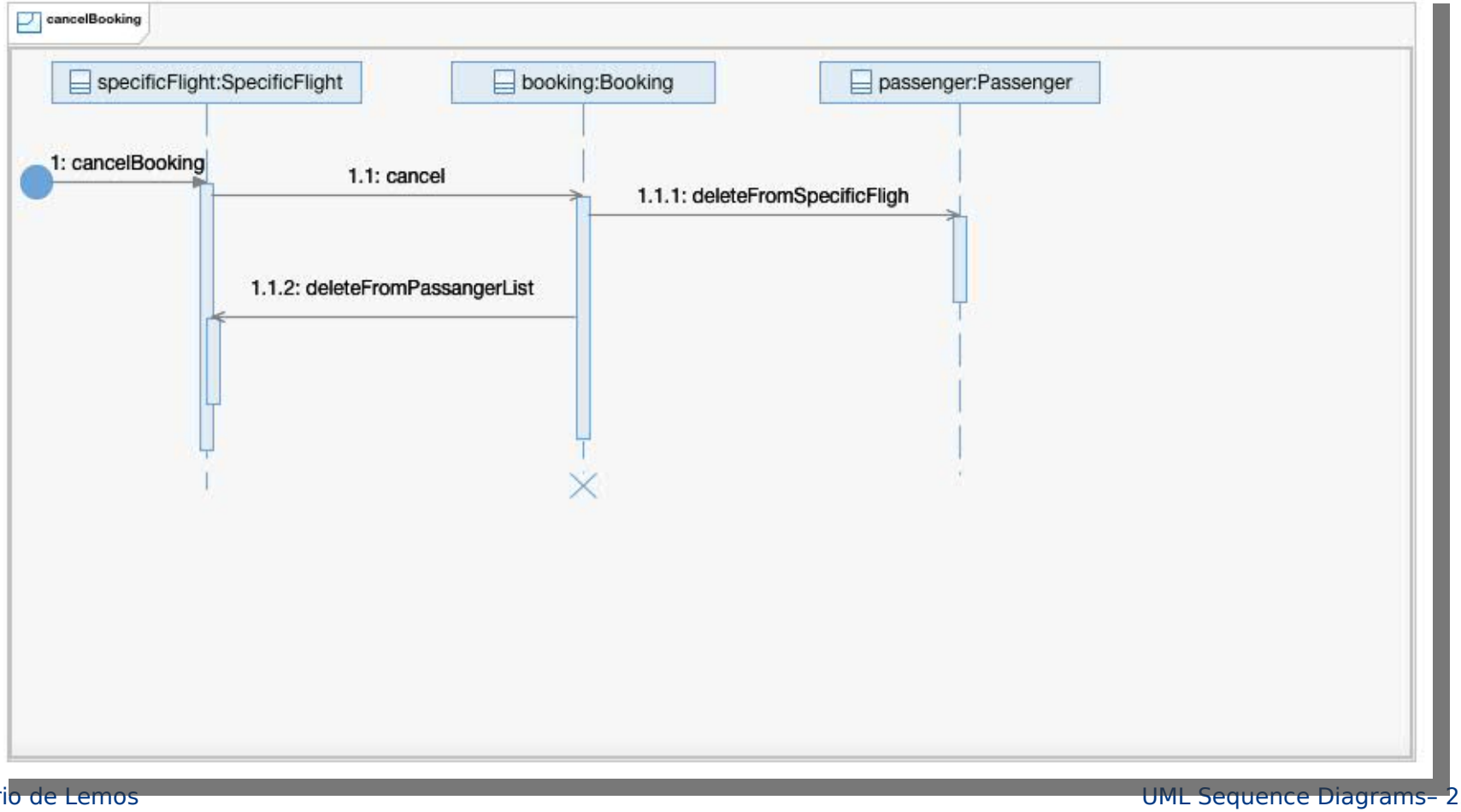
From UML Class to Sequence Diagrams

The sequence diagram capturing the cancellation of a booking with the destruction of an object (synchronous)



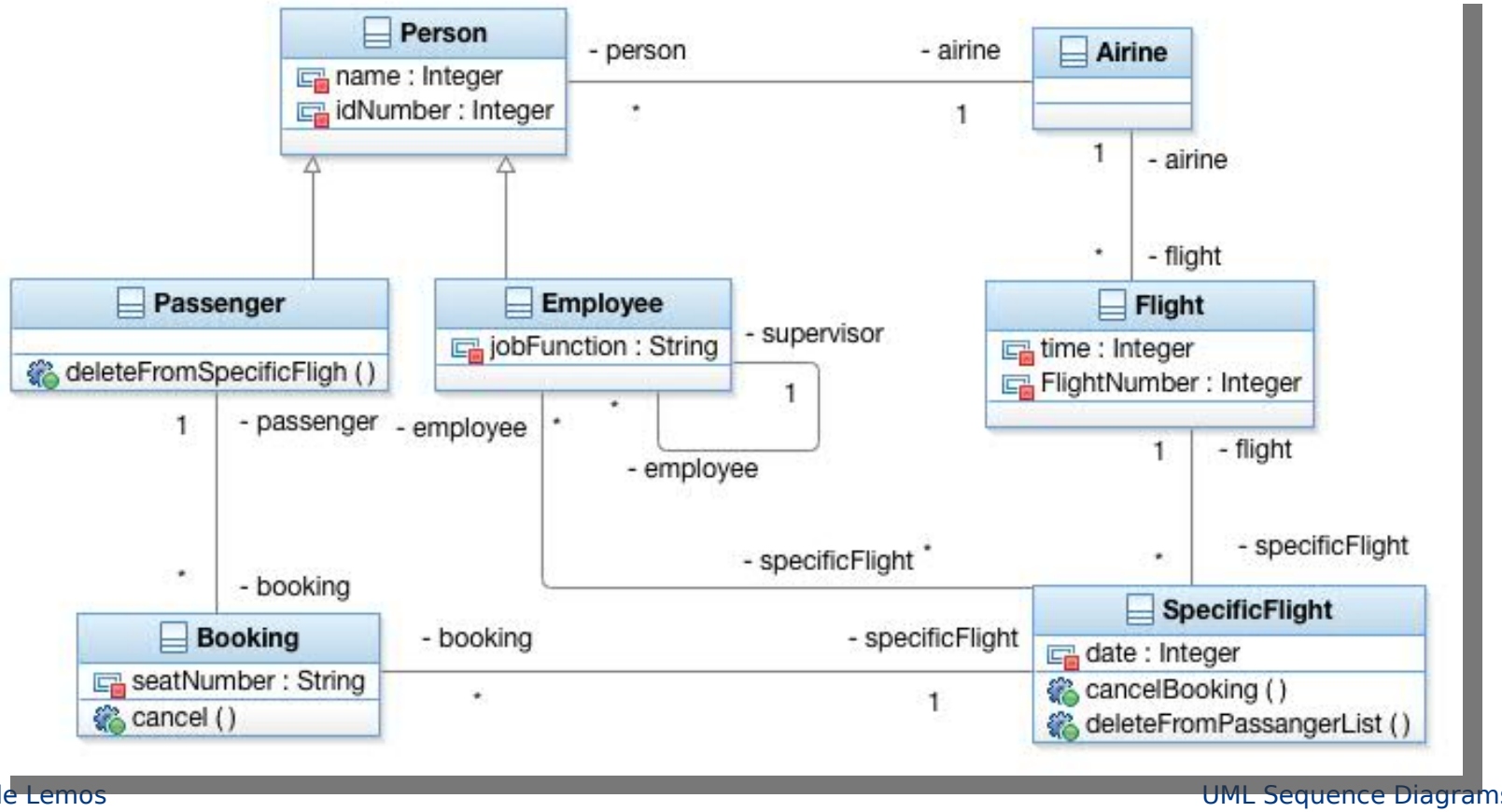
From UML Class to Sequence Diagrams

The sequence diagram capturing the cancellation of a booking with the destruction of an object (as



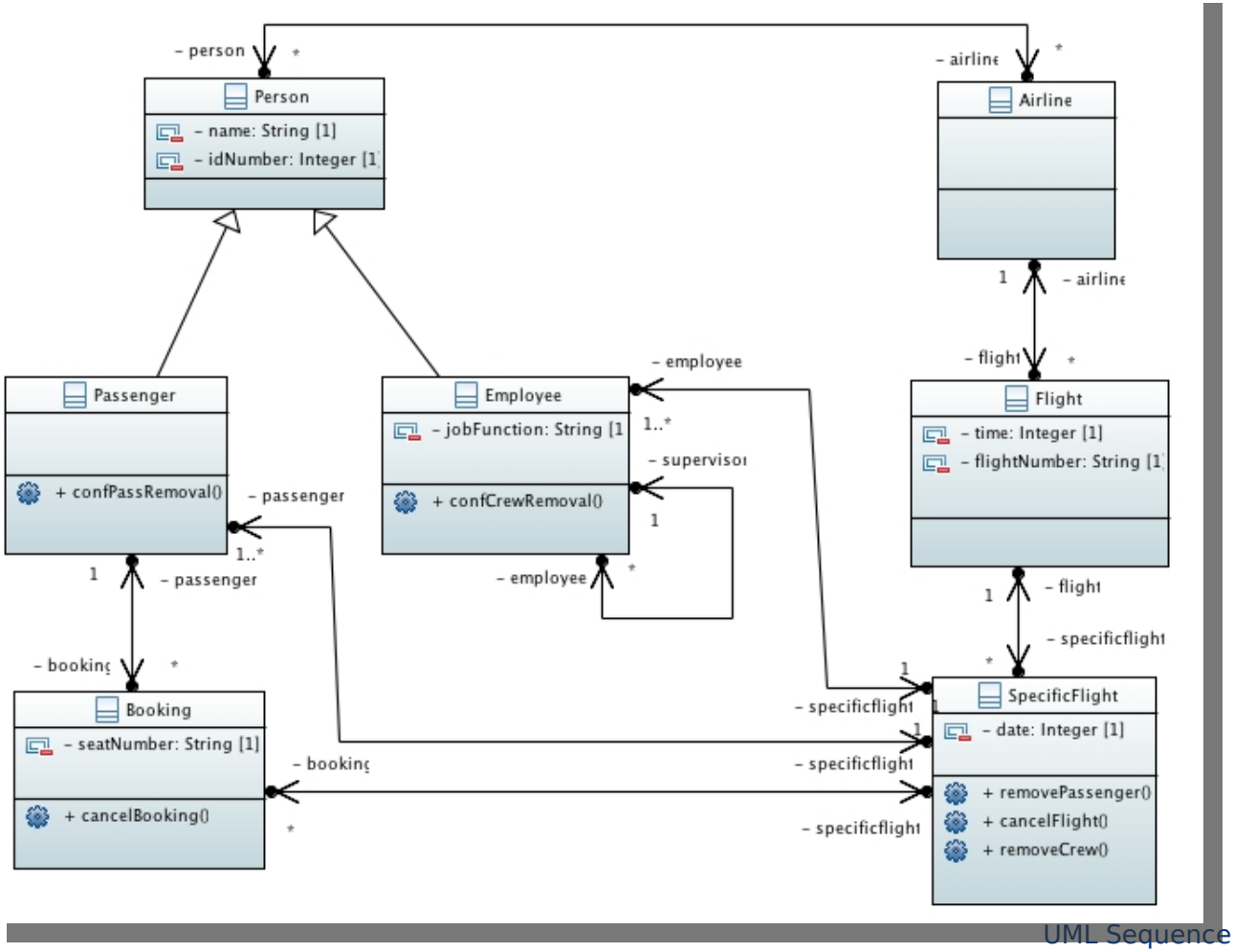
From UML Class to Sequence Diagrams

Does the following UML class diagram representing the Koffee Airline reservation system is able to represent cancel a flight?



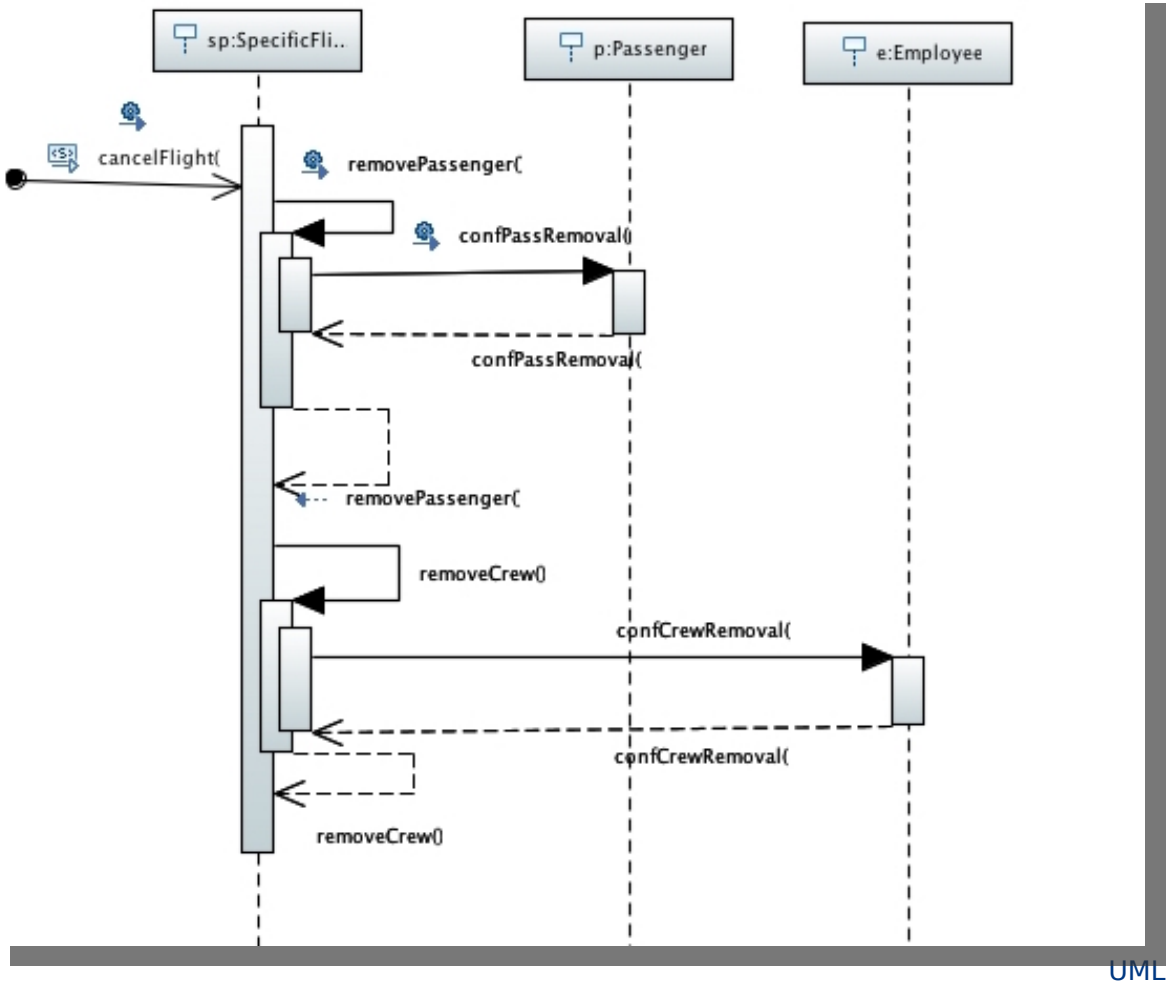
From UML Class to Sequence Diagrams

The UML class diagram of Koffee Airline reservation system with additional operations for capturing cancel a flight



From UML Class to Sequence Diagrams

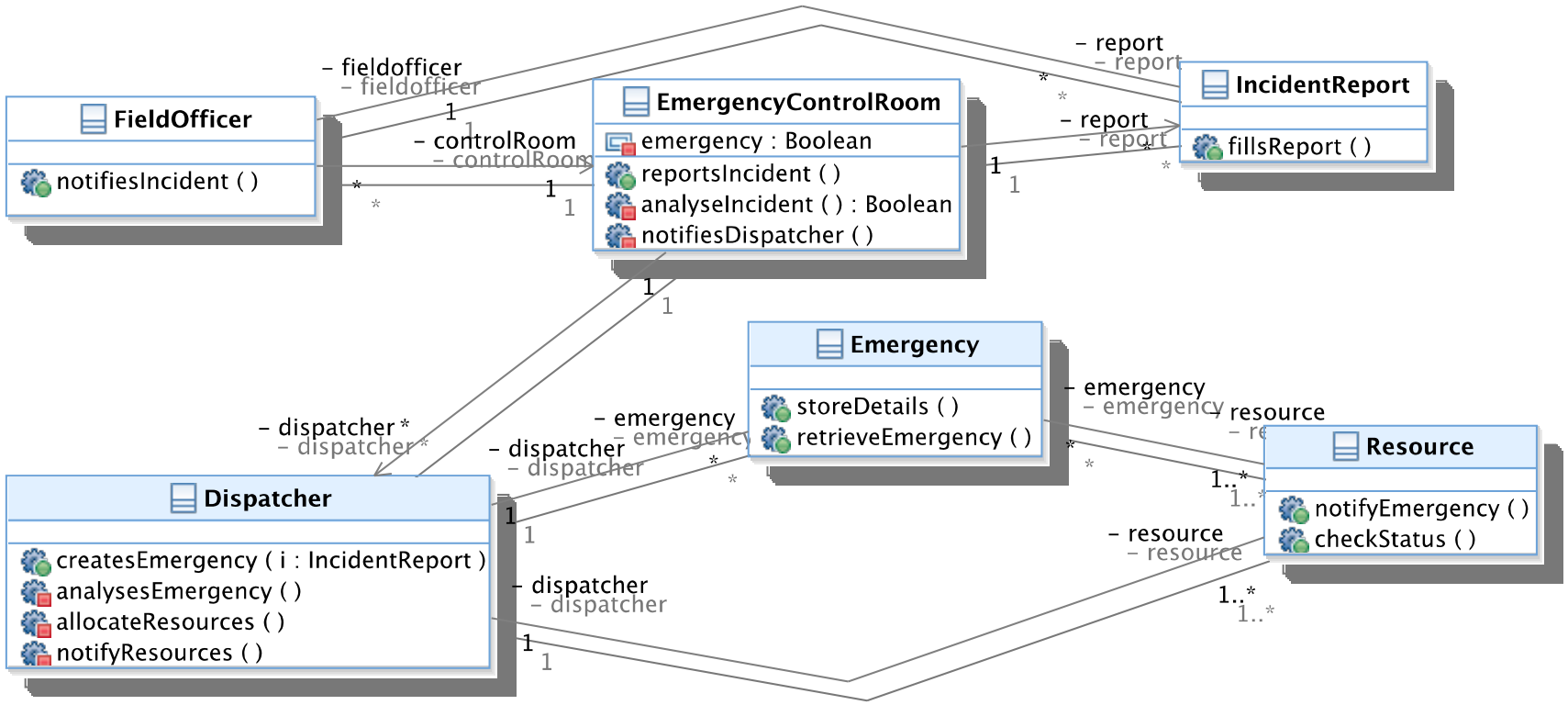
The sequence diagram capturing the cancellation of a flight (synchronous)



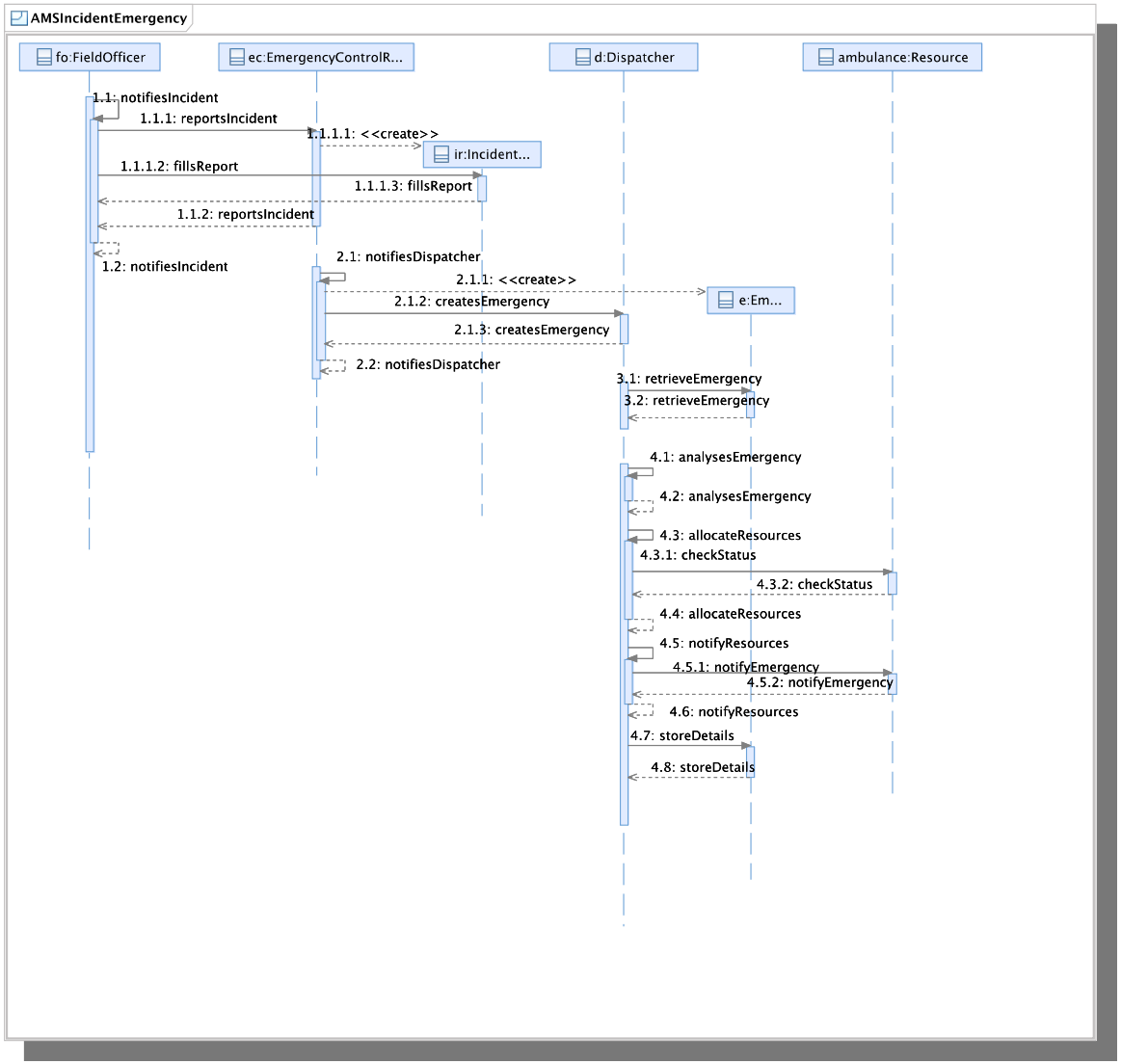
Example: Accident Management System

- field officers, like police officer or fire fighter, have access to an application that enable them to contact a dispatcher for dealing with emergencies
- the dispatcher can visualise the status of all the resources (ambulances, fire trucks and police cars), and dispatch appropriate resources to the emergency

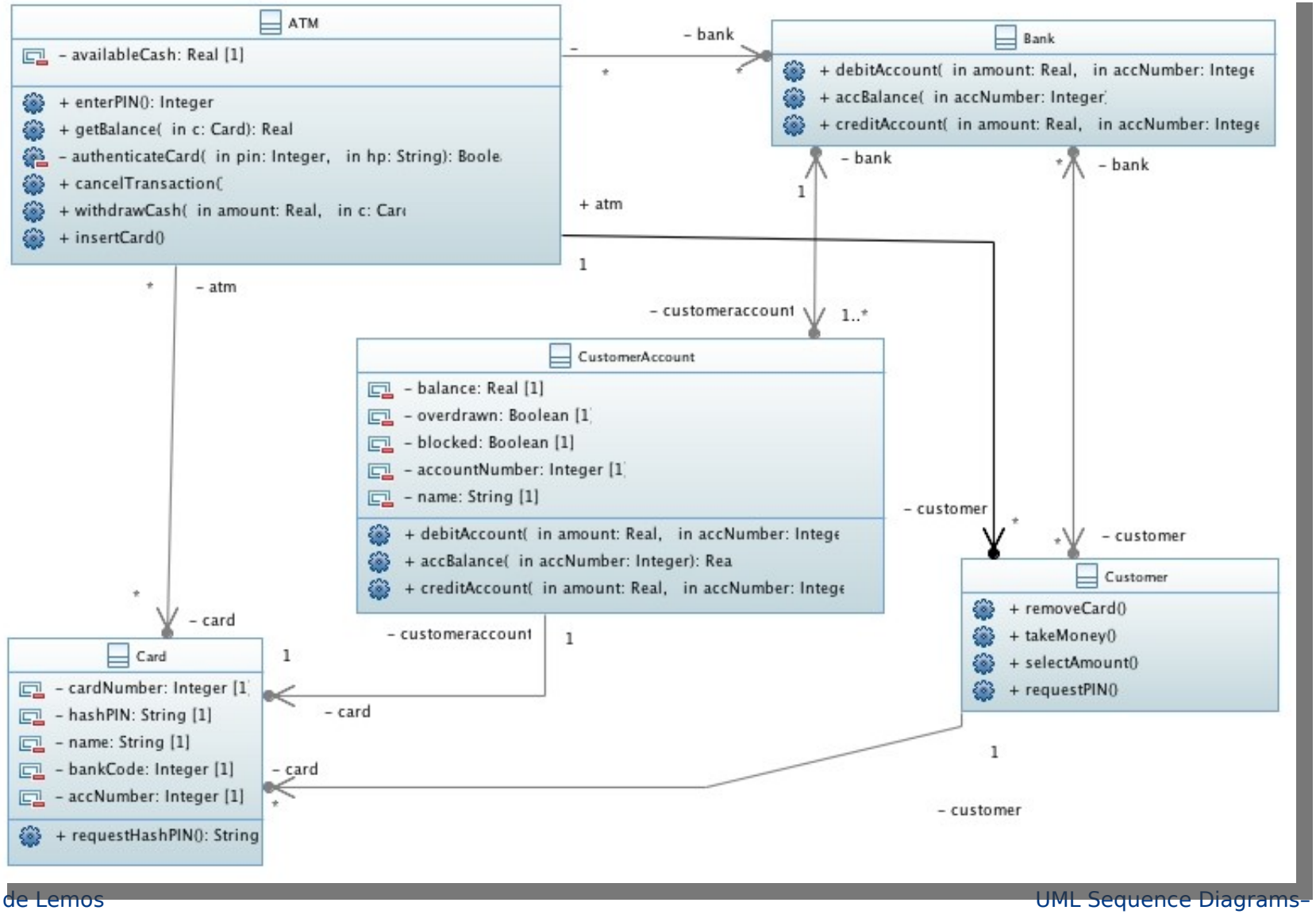
Accident Management System
UML Class Diagram



Accident Management System UML Sequence Diagram

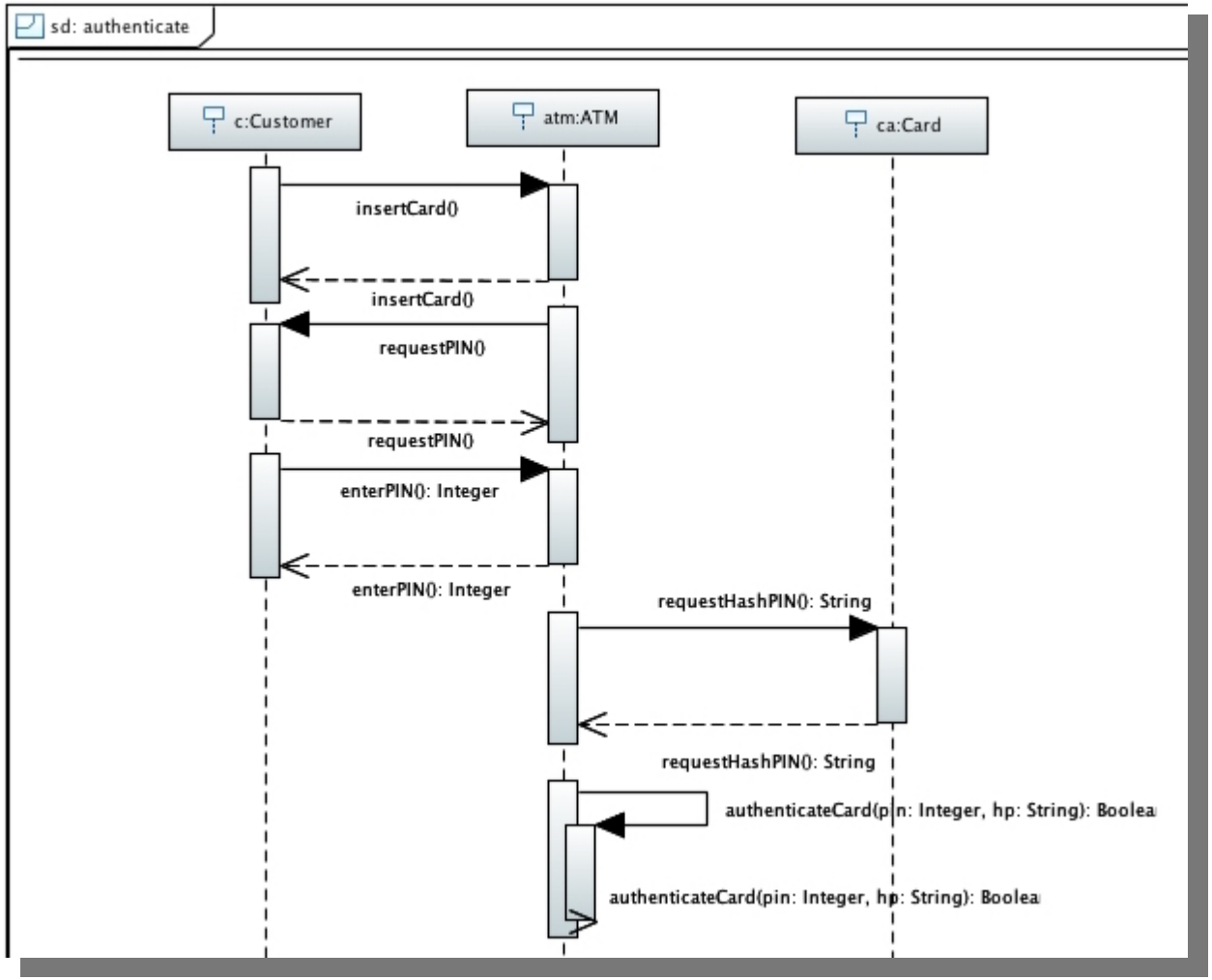


Example: ATM


ATM
- availableCash: Real [1]
+ enterPIN(): Integer
+ getBalance(in c: Card): Real
- authenticateCard(in pin: Integer, in hp: String): Boole
+ cancelTransaction(
+ withdrawCash(in amount: Real, in c: Car
+ insertCard()
Bank
+ debitAccount(in amount: Real, in accNumber: Intege
+ accBalance(in accNumber: Integer
+ creditAccount(in amount: Real, in accNumber: Intege
- bank
- bank
- bank
+ atm
- atm
- customeraccount
1..*
CustomerAccount
- balance: Real [1]
- overdrawn: Boolean [1
- blocked: Boolean [1]
- accountNumber: Integer [1
- name: String [1]
+ debitAccount(in amount: Real, in accNumber: Intege
+ accBalance(in accNumber: Integer): Rea
+ creditAccount(in amount: Real, in accNumber: Intege
- customer
- customer
Customer
+ removeCard()
+ takeMoney()
+ selectAmount()
+ requestPIN()
- card
- customeraccount
- card
- card
- customer
Card
- cardNumber: Integer [1
- hashPIN: String [1]
- name: String [1]
- bankCode: Integer [1]
- accNumber: Integer [1]
+ requestHashPIN(): String

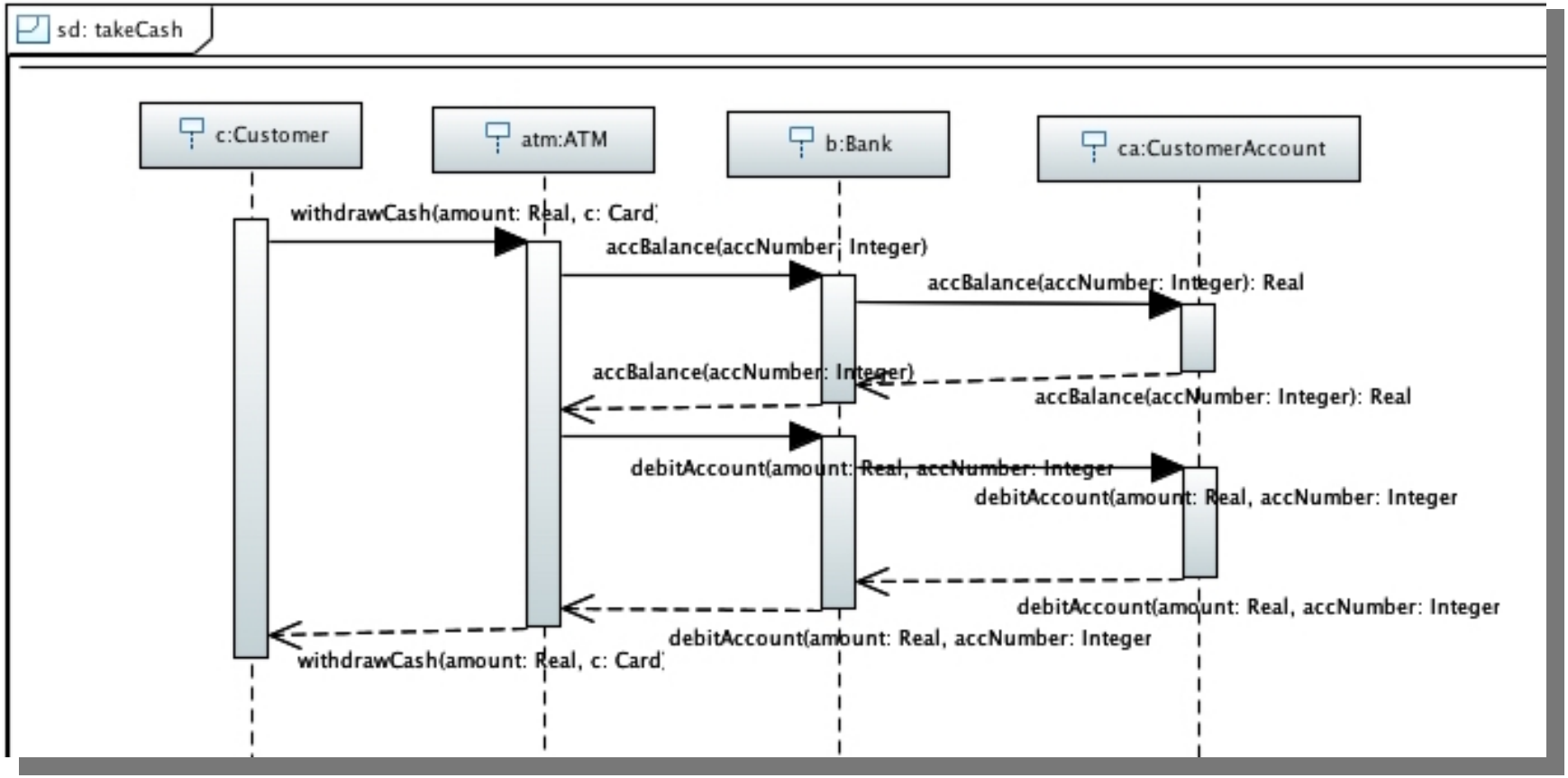
ATM
UML Sequence Diagram

The sequence diagram capturing authentication



ATM
UML Sequence Diagram

The sequence diagram of taking cash



Combined Fragments

A **combined fragment** is used to group sets of messages together to show conditional flow in a sequence diagram

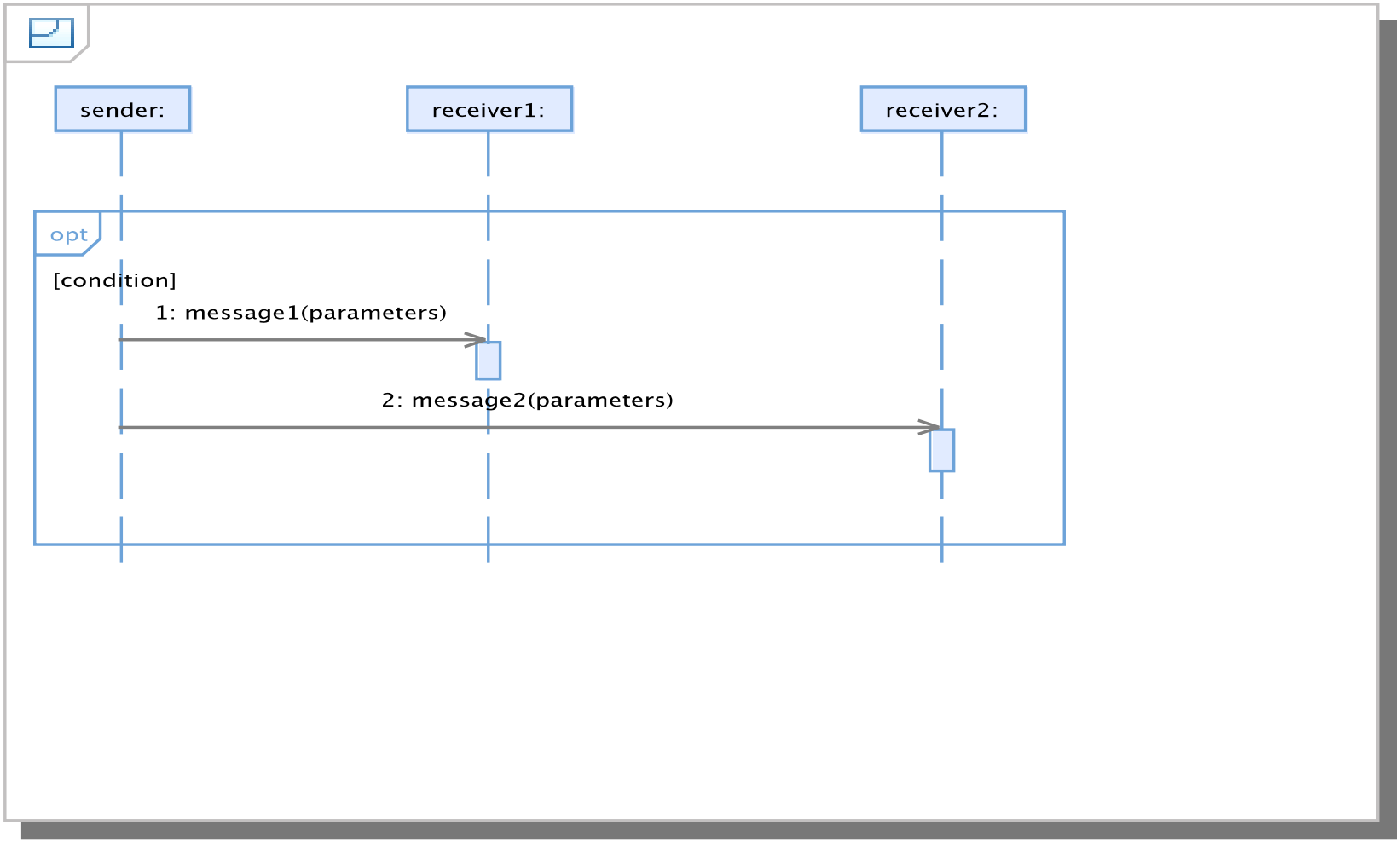
- there are 11 interaction types for combined fragments
  - option
  - alternatives
  - loops

Option

Option combination fragment is used to model a sequence

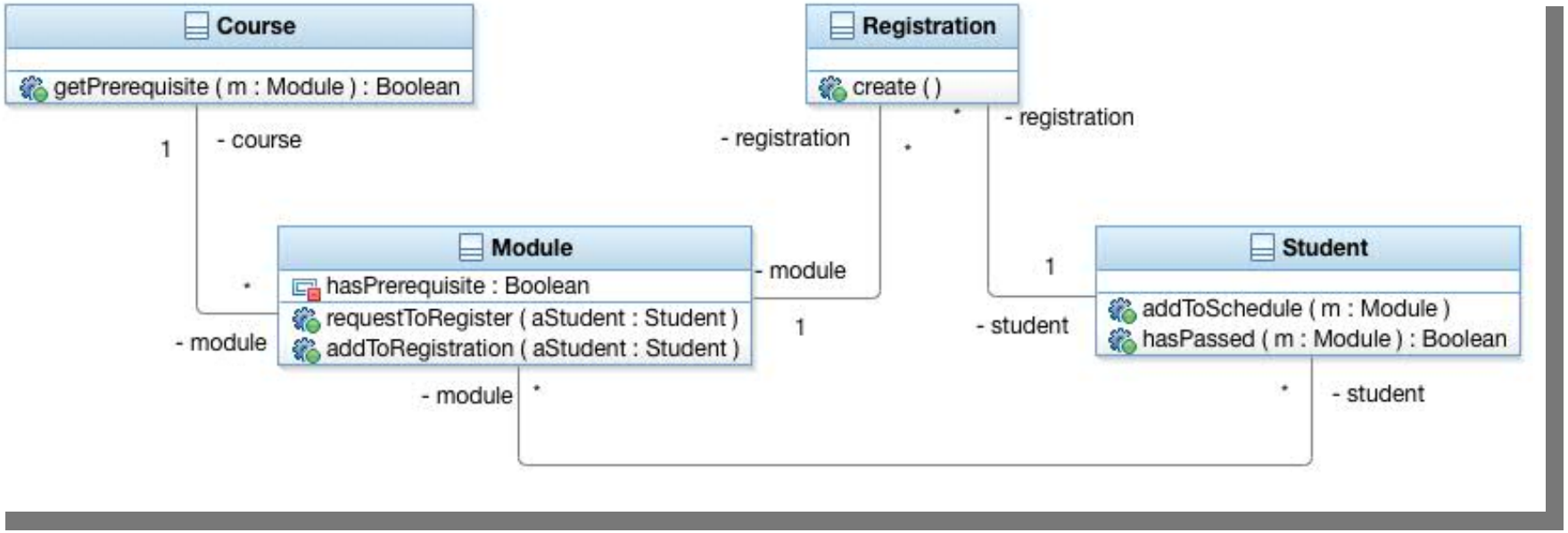
- given a certain condition, will occur; otherwise, the sequence does not occur
- it is used to model a simple "if then" statement
- the text "`opt`" is placed inside the frame's *namebox*

Option: Example



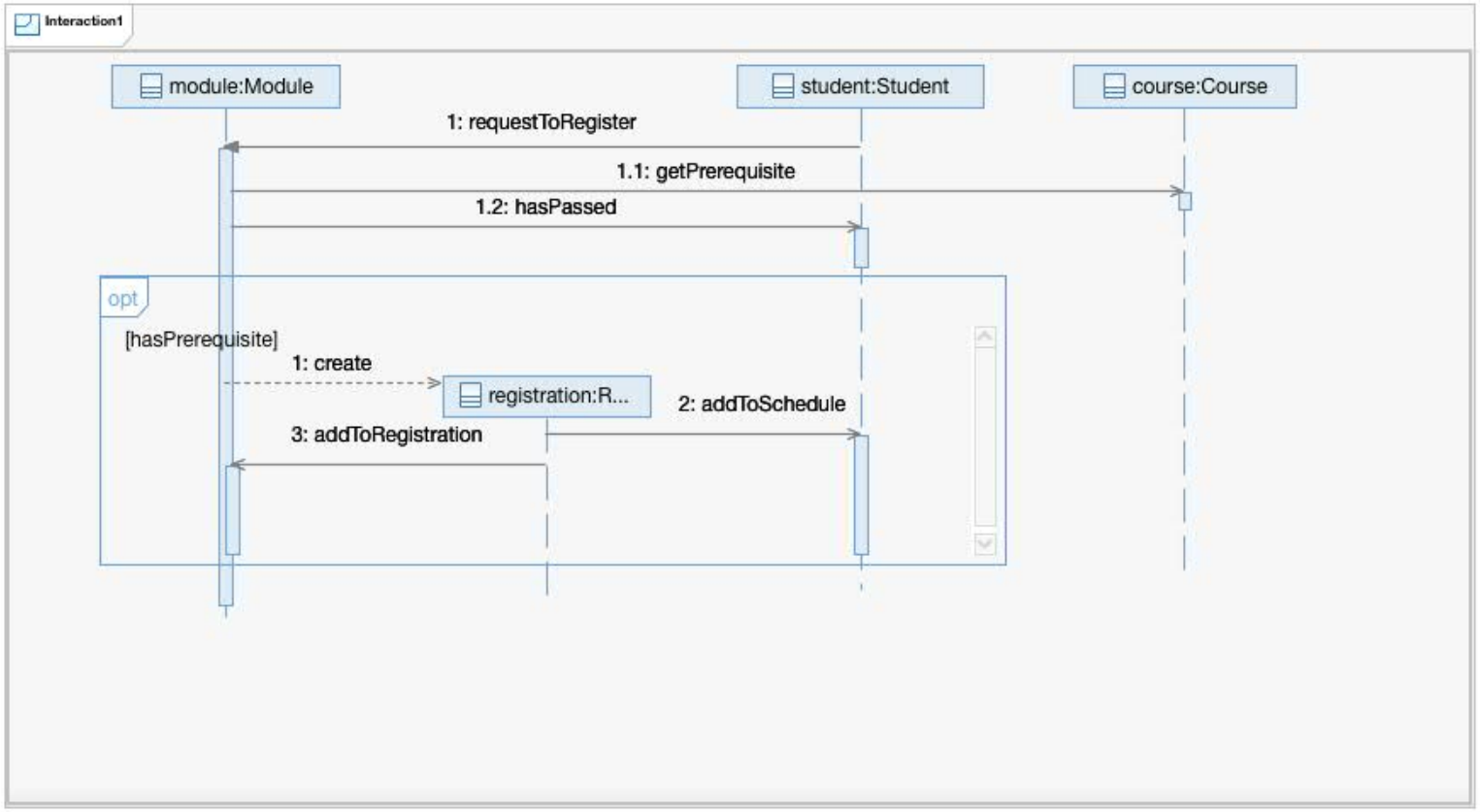
Example of an Option

Consider the following example, which captures the process of a student registering into a module



Example of an Option

The sequence diagram captures the condition that a student is able to register into a module if she/he has the prerequisite

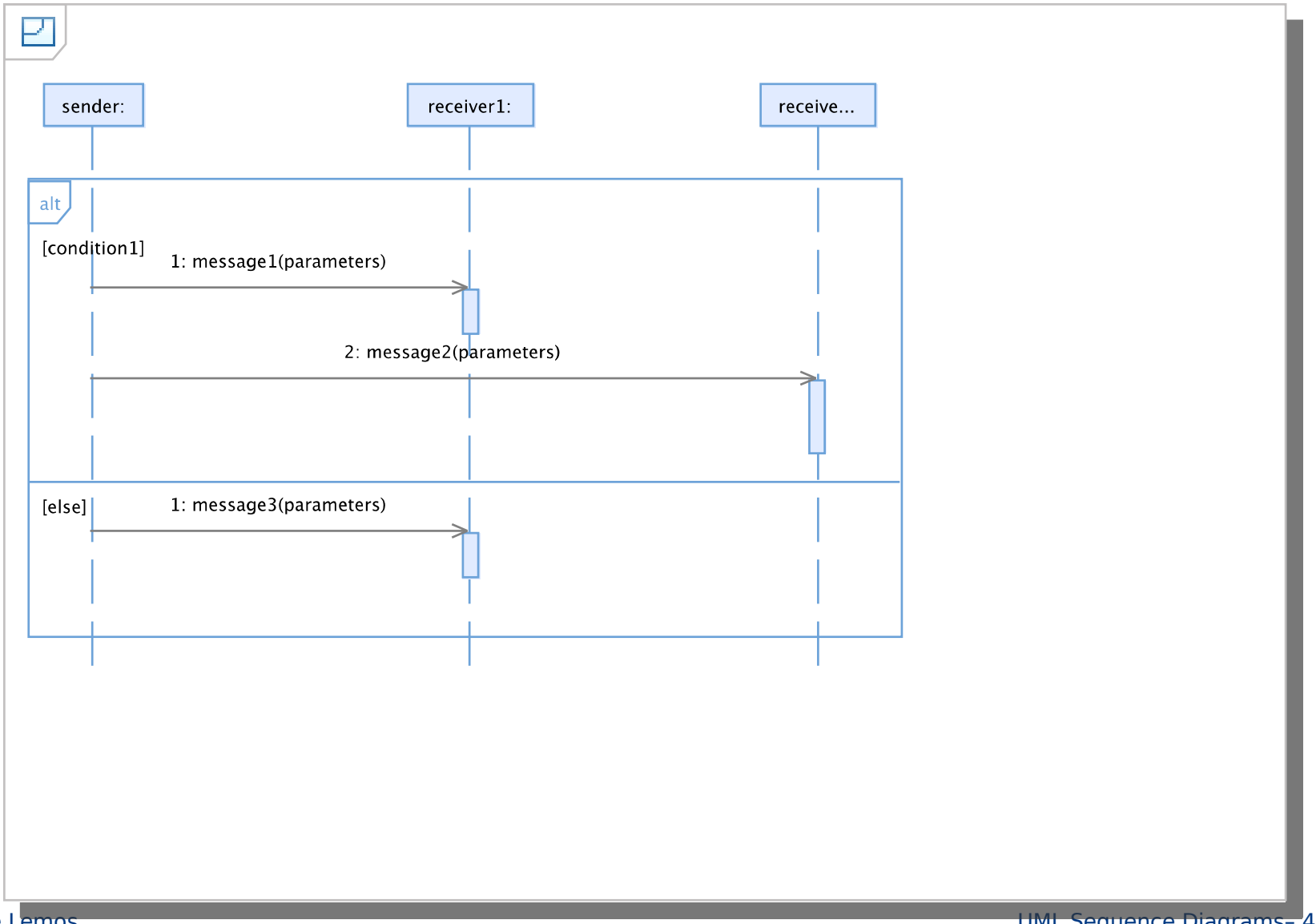


Alternatives

Alternatives are used to designate a mutually exclusive choice between two or more message sequences

- allow the modelling of the classic "if then else" logic
- the text "`alt`" is placed inside the frame's *namebox*
- the lifeline to which the guard is attached is the lifeline that owns the variable that is included in the guard expression

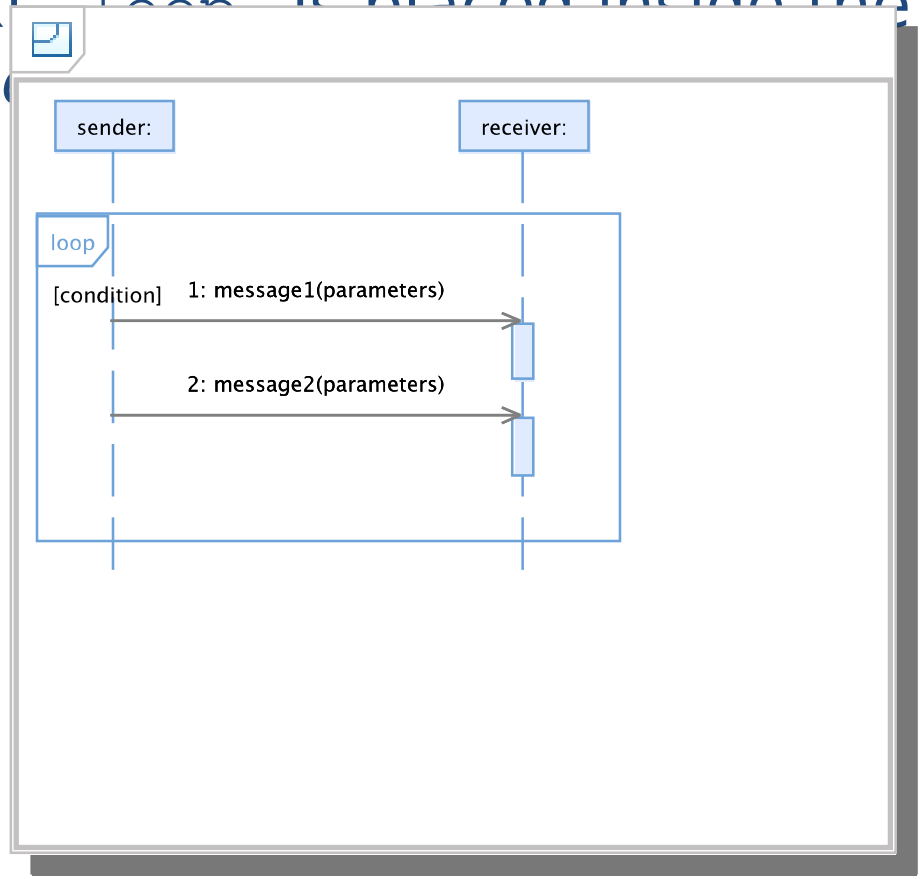
Alternatives: Example



Loops

Loop combination fragment model a repetitive sequence

- the text "loop" is placed inside the frame's *nameb*



Example of a Loop

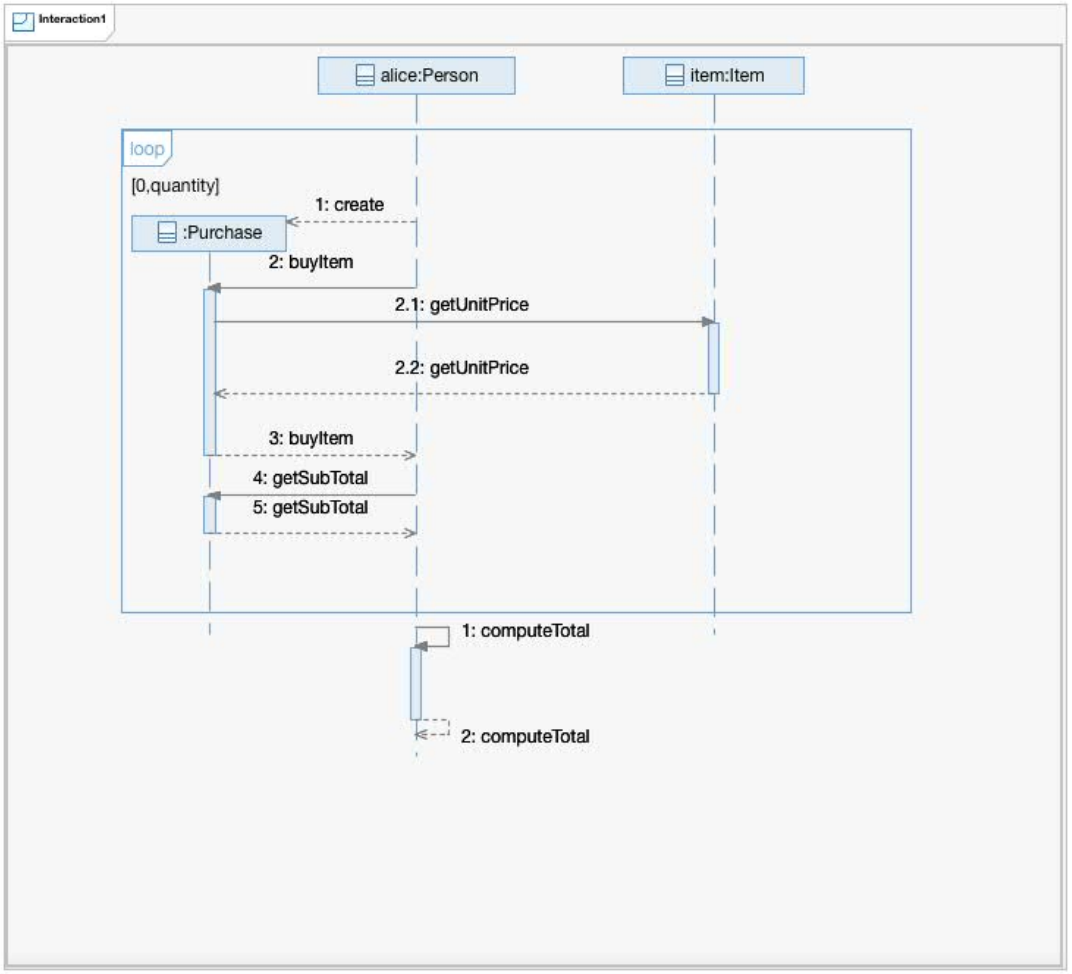
Consider the following example, which captures a customer making several purchases

The sequence diagram represents the iteration as the number of messages need to be repeated

- depending on the `quantity` we need to calculate the subtotal depending on the unit price

Example of a Loop

The number of times the loop needs to execute is specified by `quantity` (synchronous messaging)

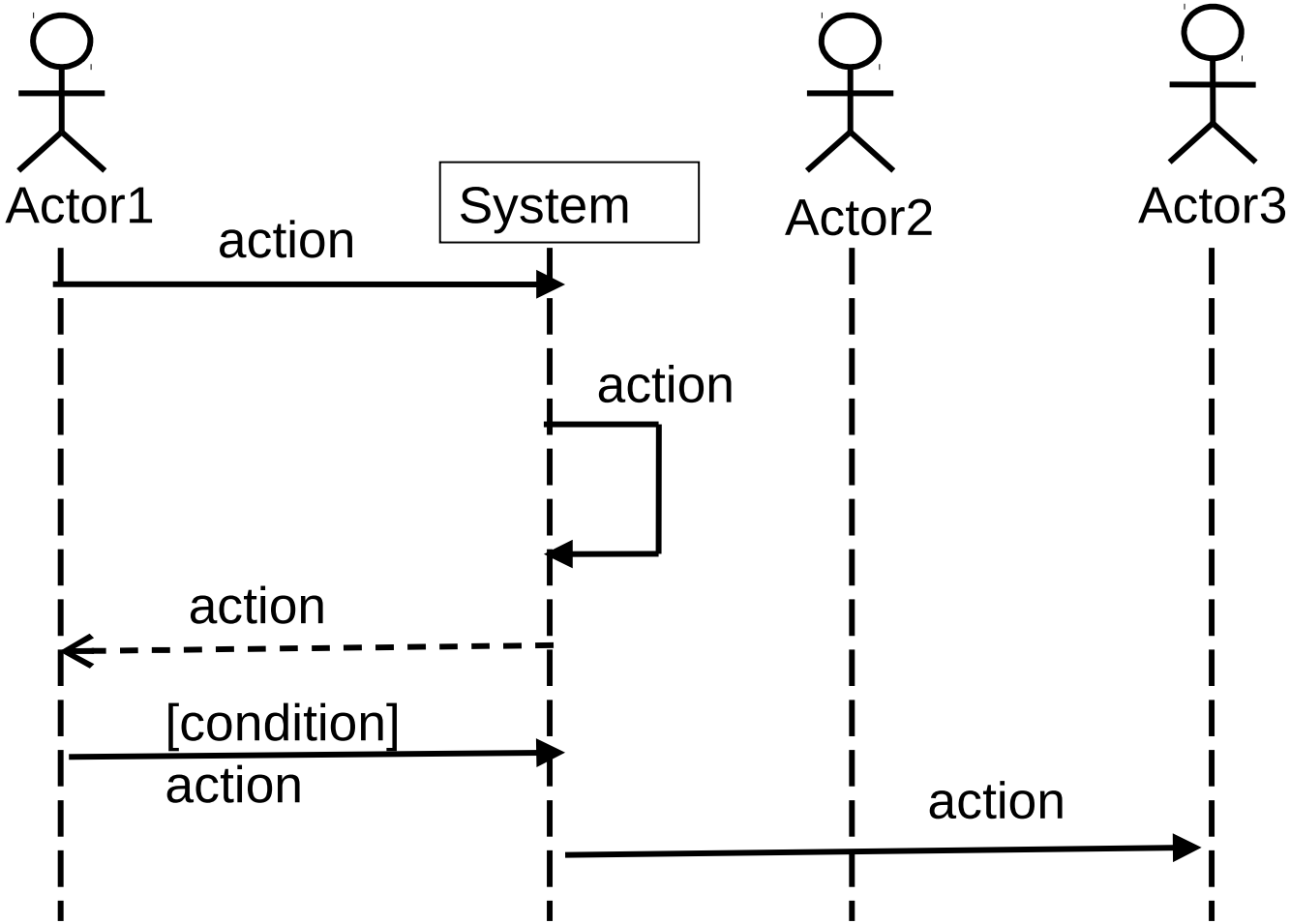


Use Case and Sequence Diagrams

Sequence diagrams can elaborate use cases

- can be used to depict use case steps
  - we could also use an activity diagram
- they can be seen as a simplified form of diagram notation
- actors and the system are objects
- actions can assume an informal description
  - rather than a method call

Use Case Sequence Diagram



Use Case: Place Order

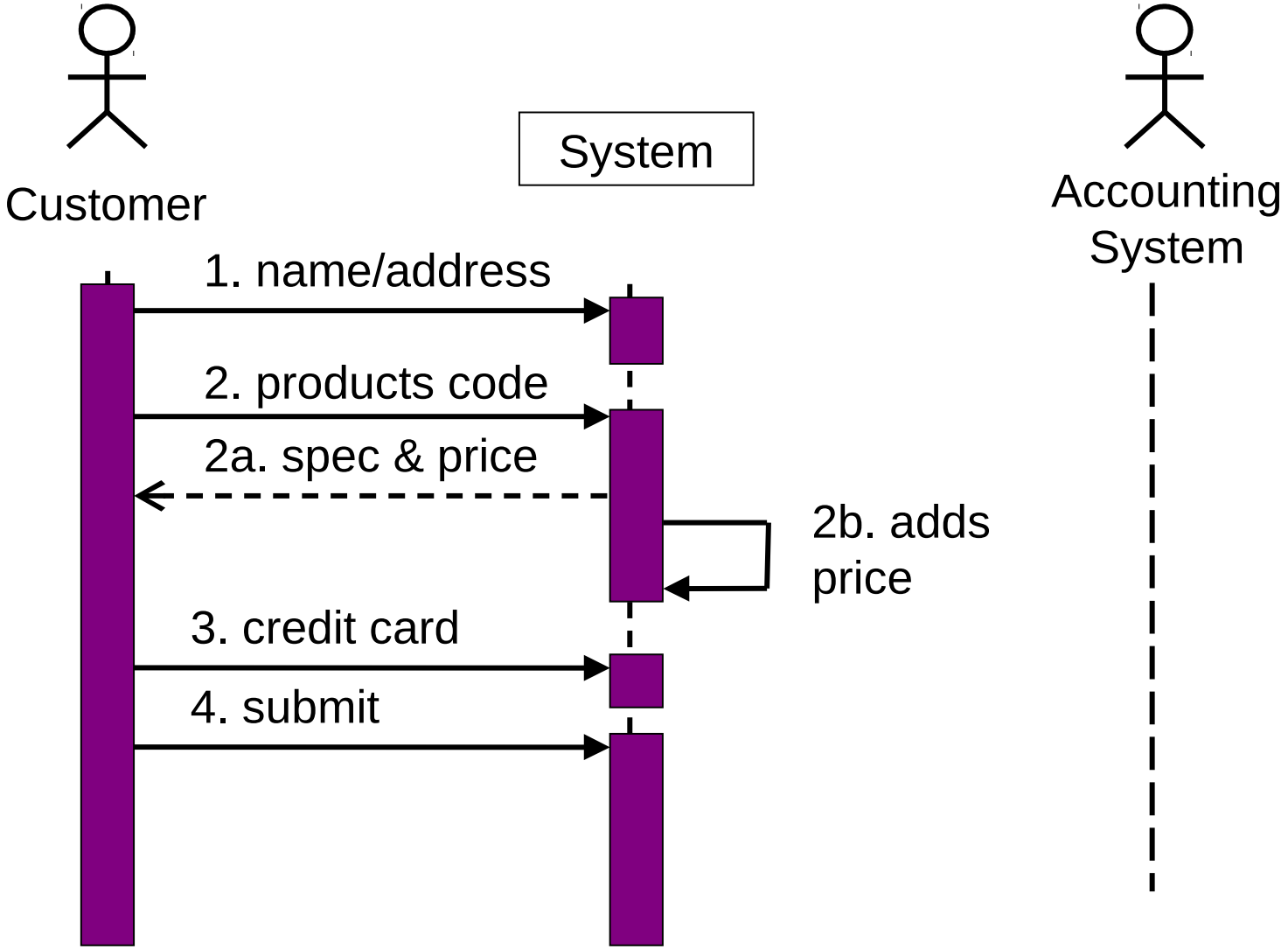
Idealised scenario

- only the normal path
  - no branching or alternatives in your basic path

Example: **Place order**

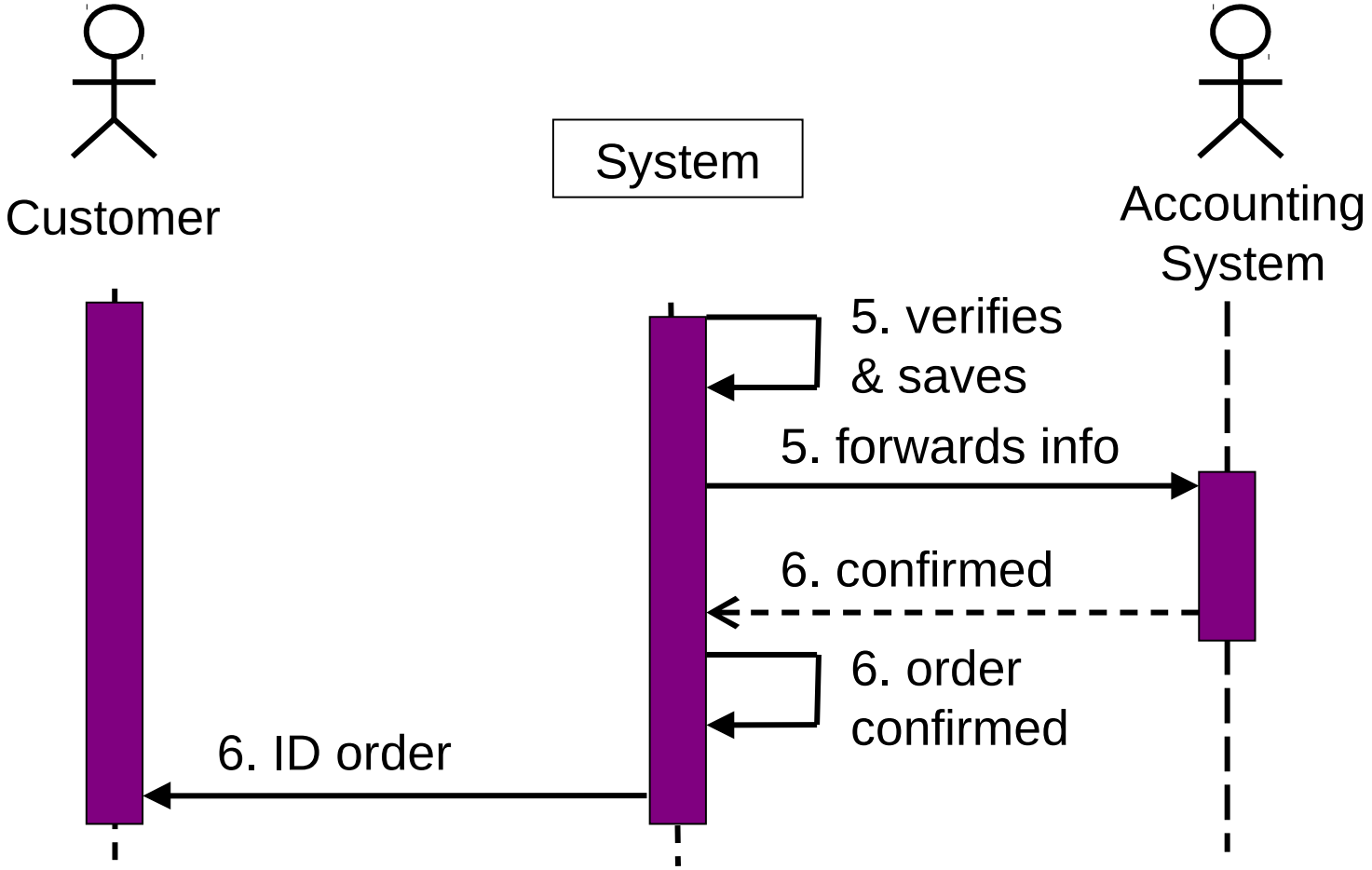
1. Customer enters his/her name and address
2. While customer enters product code
   a. The system provides product spec and price
   b. The system adds up the price of items
3. The customer enters credit card info
4. The customer selects submit
5. The system verifies the info, saves the order, and forwards the info to accounting system
6. When payment is confirmed, the order is marked confirmed, an order ID is issued and returned to customer

Sequence Diagram: Place Order



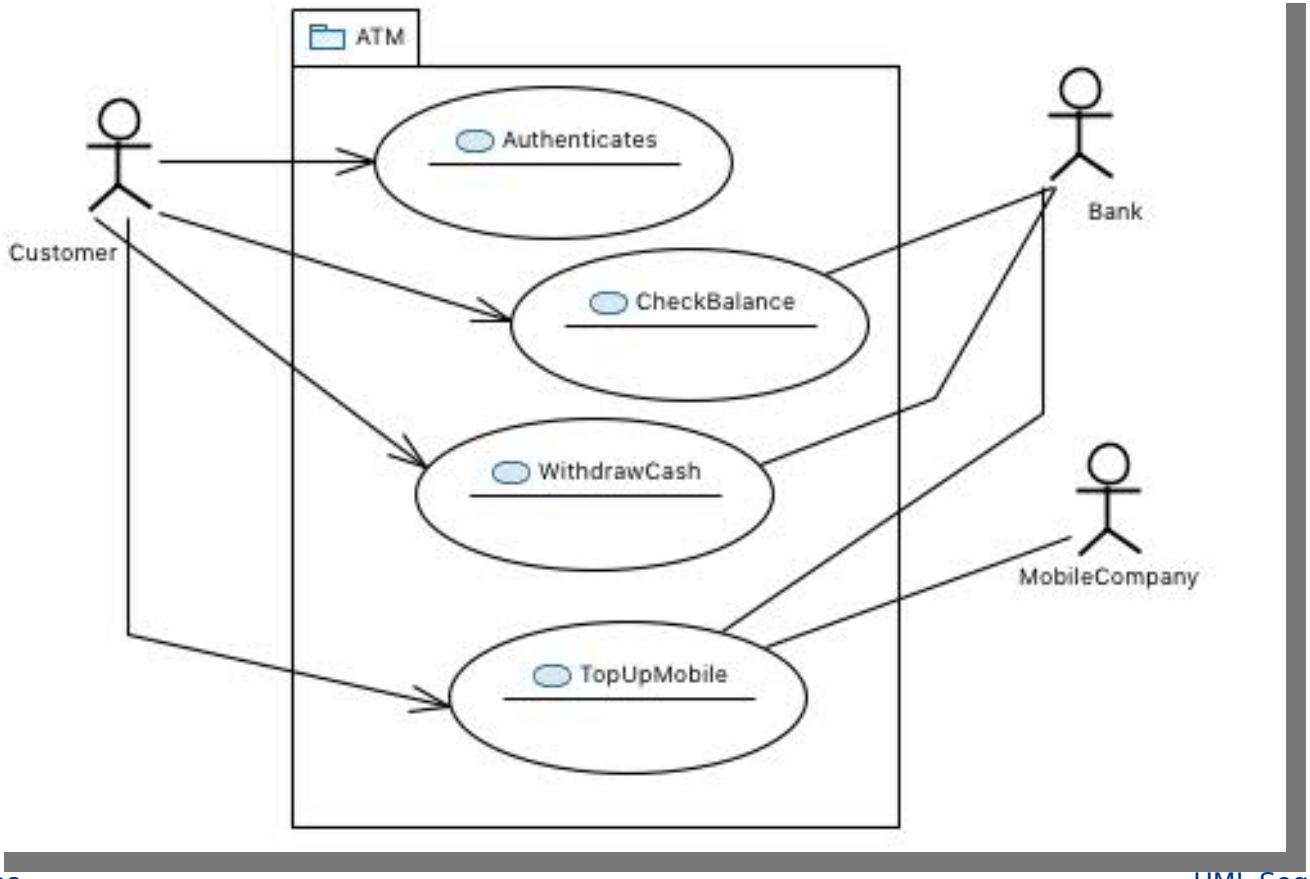
(cont.)

Sequence Diagram: Place Order



Example: ATM

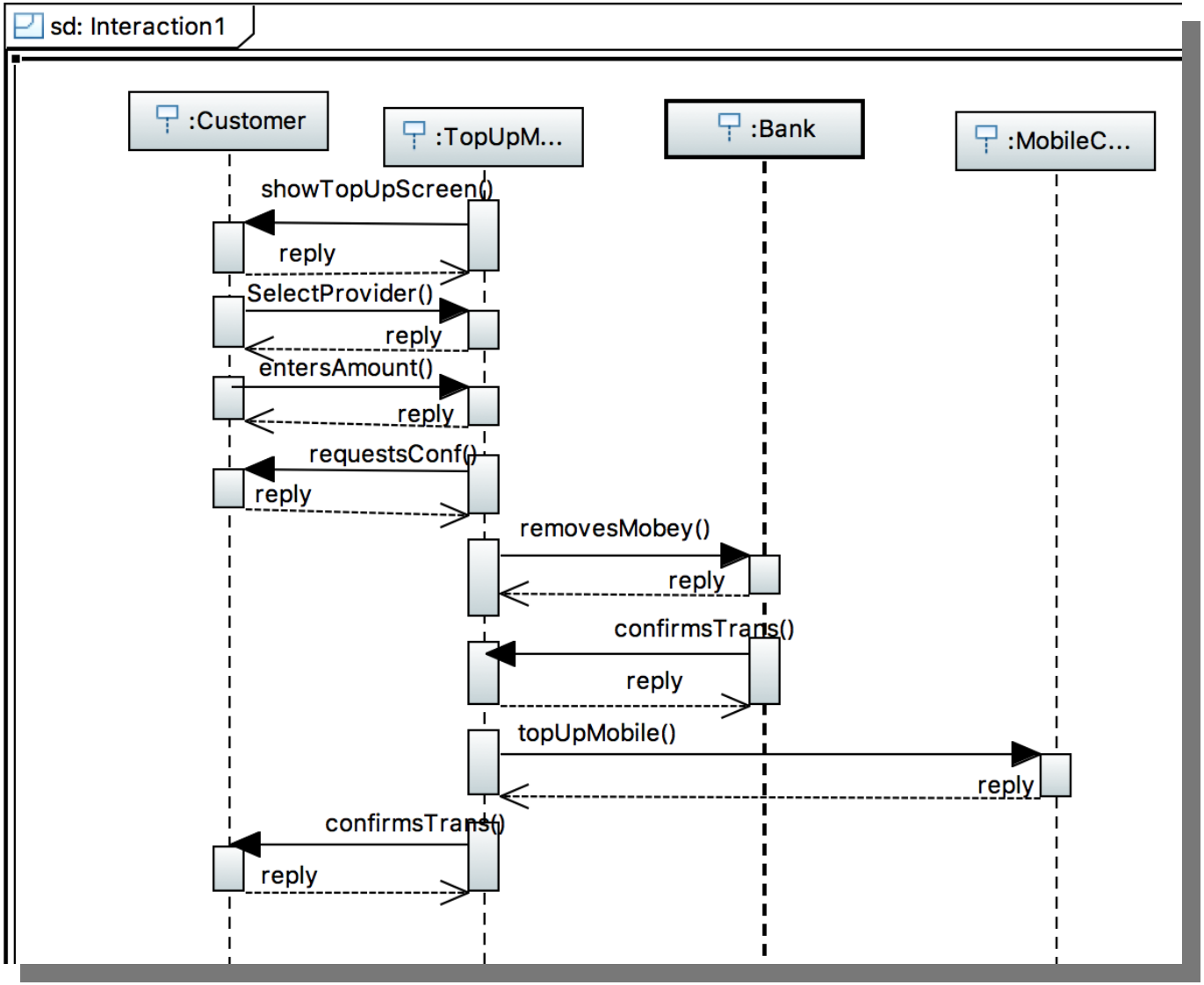
- Consider a cash machine (ATM) that provides the following services to a customer: authentication, check balance, withdraws cash, and top-up her/his mobile phone.



Example: ATM

Use case name	`TopUpMobile`
Participating actors	`Customer, Bank, MobileCompany`
Flow of events: Normal flow	Normal path 1. `ATM` prompts the top up screen 2. `Customer` choses service provider 3. `Customer` enters the amount to top up 4. `ATM` requests confirmation from `customer` 5. `ATM` contacts `Bank` 6. `Bank` removes top up amount from `customer` account 7. `Bank` confirms transaction to `ATM` 8. `ATM` inform `MobileCompany` to top up `customer` mobile phone 9. `ATM` confirms to `customer` success of transaction
Flow of events: Alternative flow	`Customer` doesn't have enough funds 1. … `Customer` provides wrong number 2. …
Pre-condition	• `Customer is` authenticated
Post-condition	• `Customer` topped his mobile and less money on the bank account • `Customer` not successful in topping up mobile

Example: ATM


sd: Interaction1
:Customer
:TopUpM...
:Bank
:MobileC...
showTopUpScreen()
reply
SelectProvider()
reply
entersAmount()
reply
requestsConf()
reply
removesMobey()
reply
confirmsTrans()
reply
topUpMobile()
reply
confirmsTrans()
reply

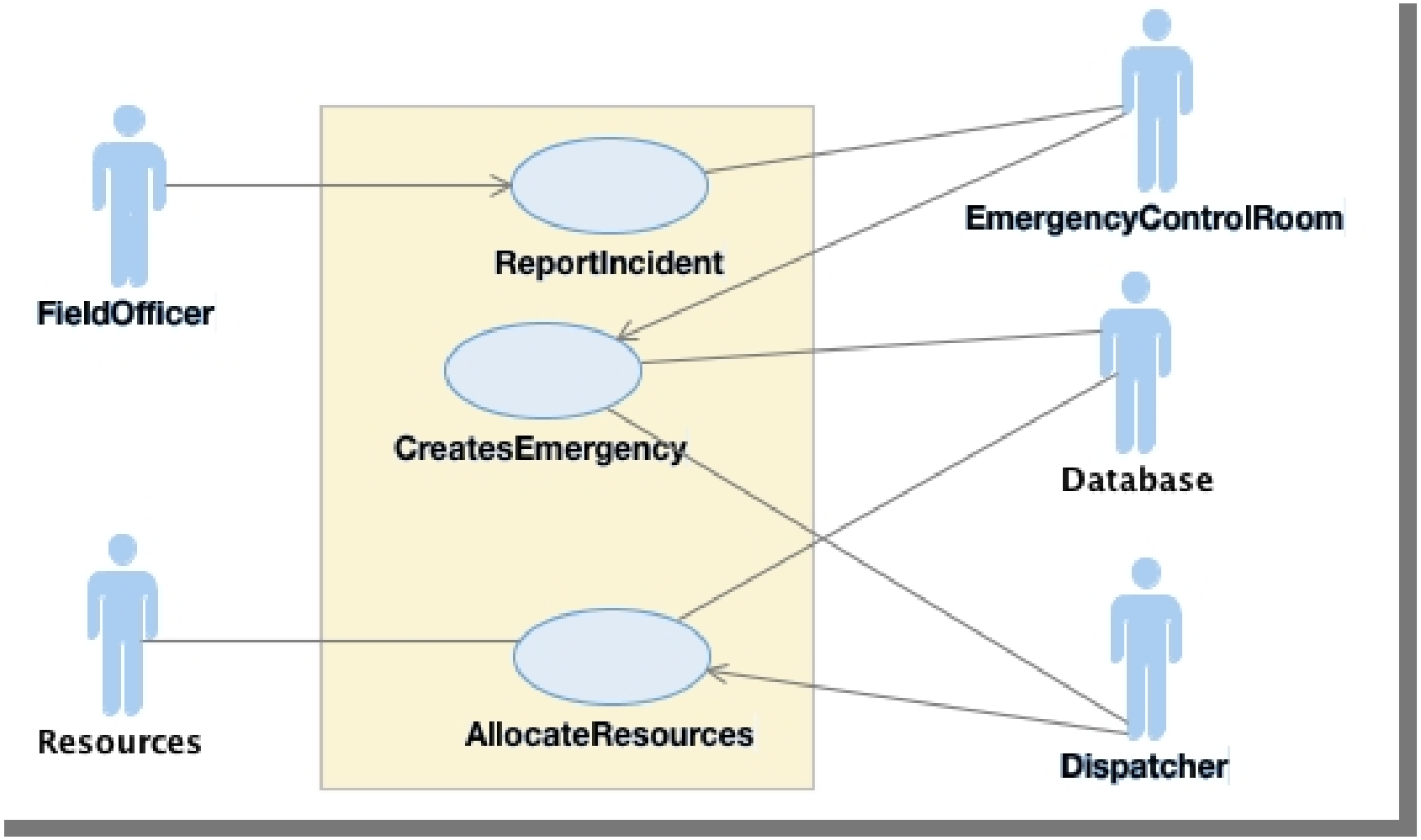
Sequence Diagrams: Design Validation

- Use case diagram
  - draw the sequence diagram for a particular use
- Class diagram
  - based on a scenario from the use case draw a sequence diagram
- Compare the two sequence diagrams
  - the sequence diagram for the class diagram should capture the behaviour specified in the use case (sequence of actions)
- Example: Accident Management System
  - the class diagram will be modified

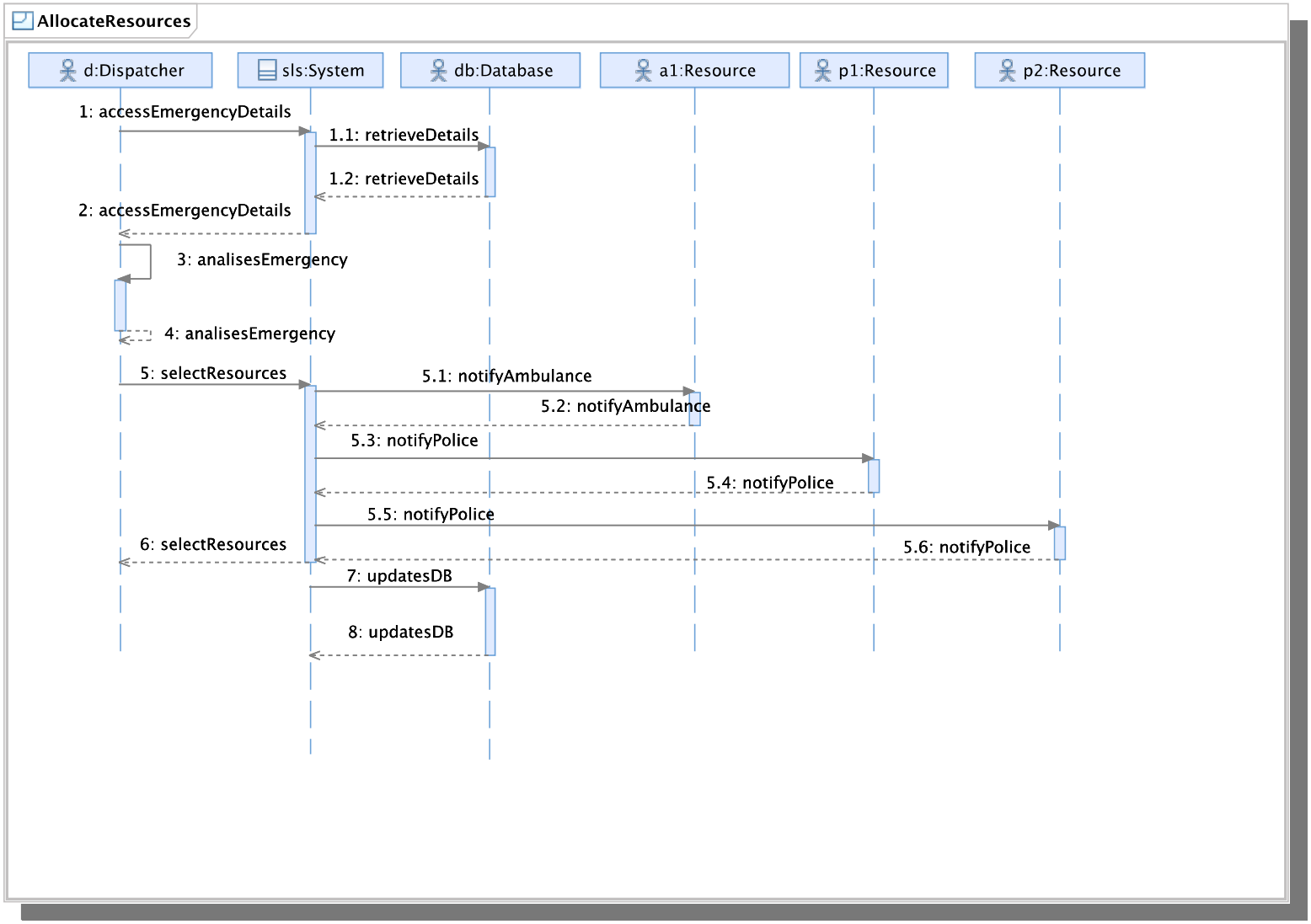
Example: Accident Management System

- field officers, like police officer or fire fighter, have access to an application that enable them to contact a dispatcher for dealing with emergencies
- the dispatcher can visualise the status of all the resources (ambulances, fire trucks and police cars), and dispatch appropriate resources to the emergency

Example: Accident Management System

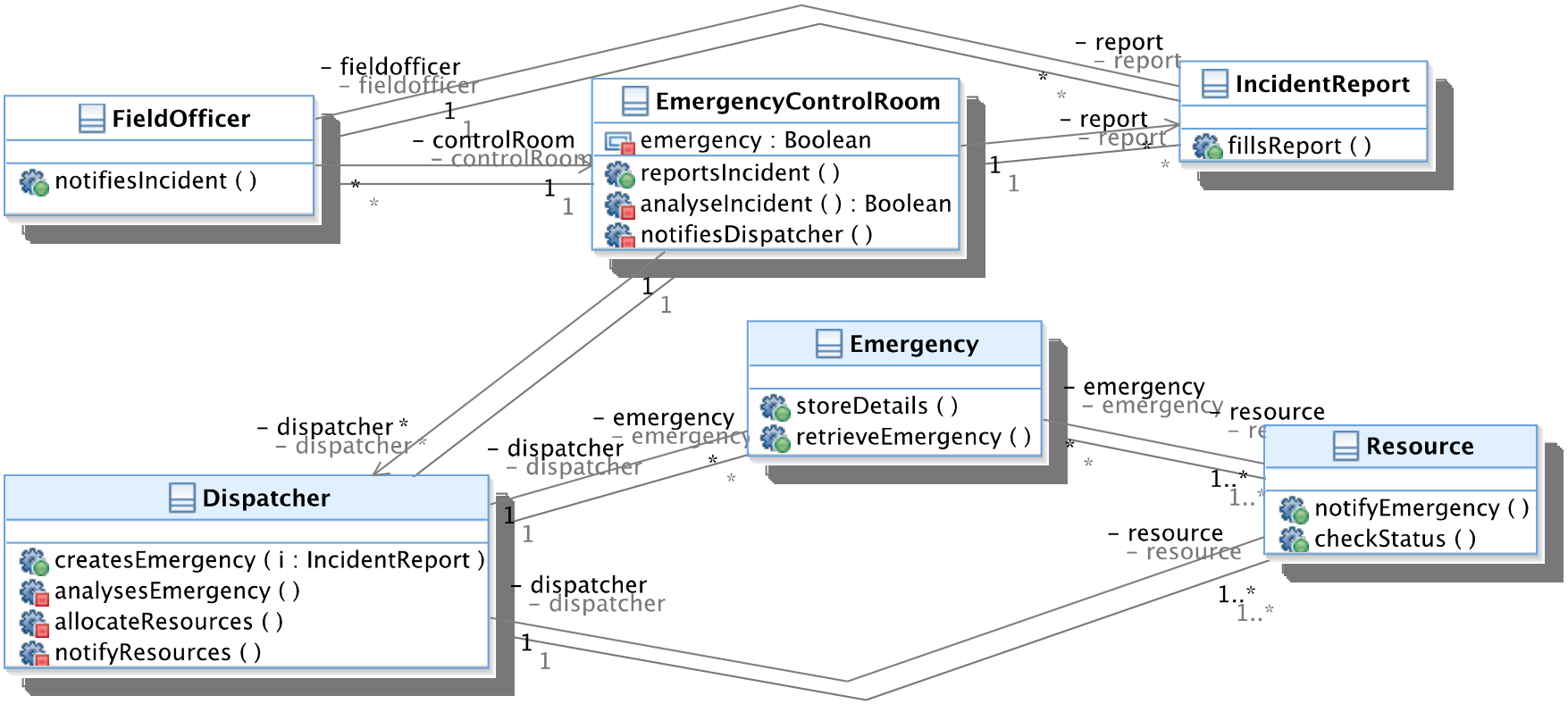


Example: Accident Management System

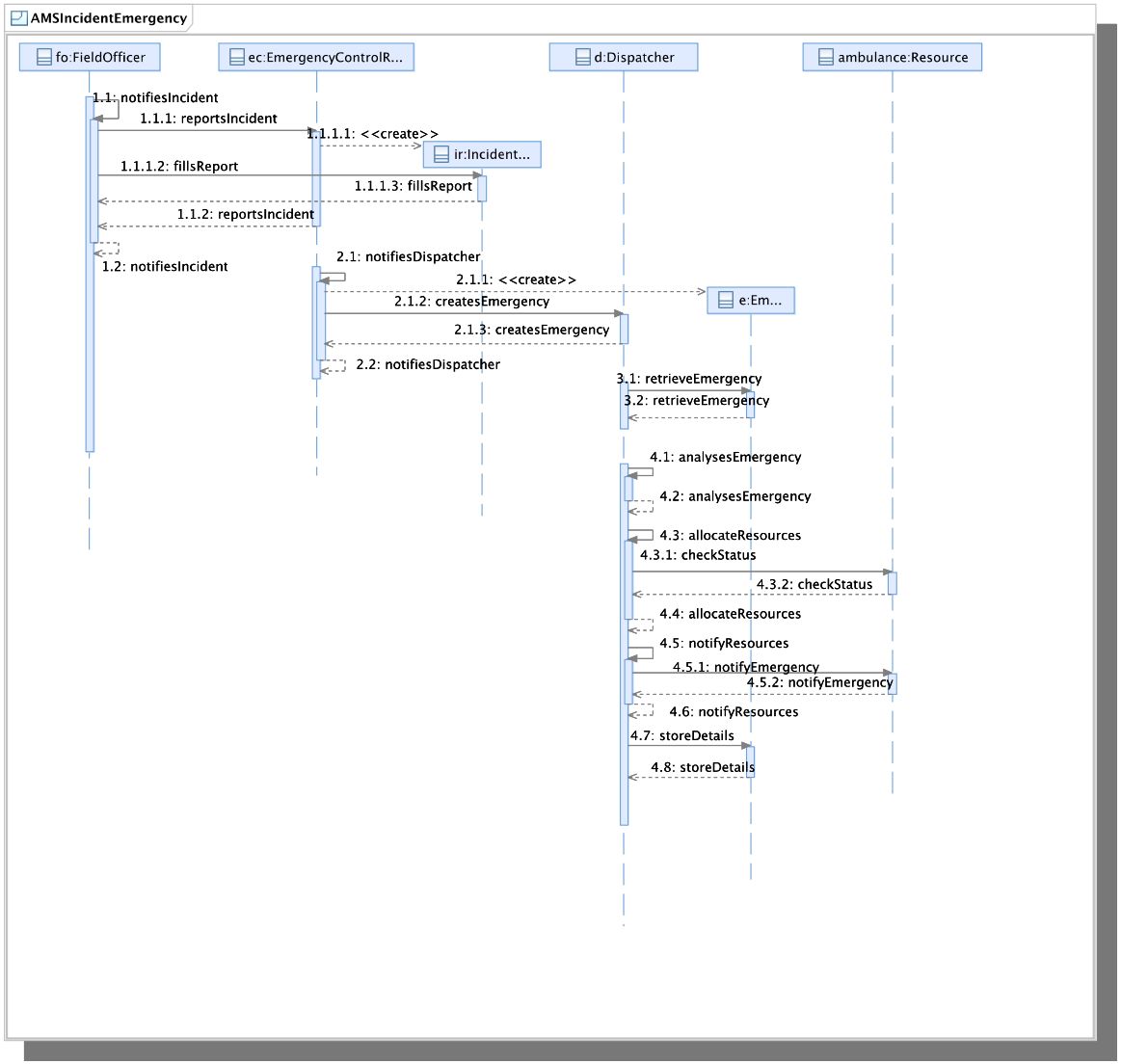


Accident Management System UML Class Diagram

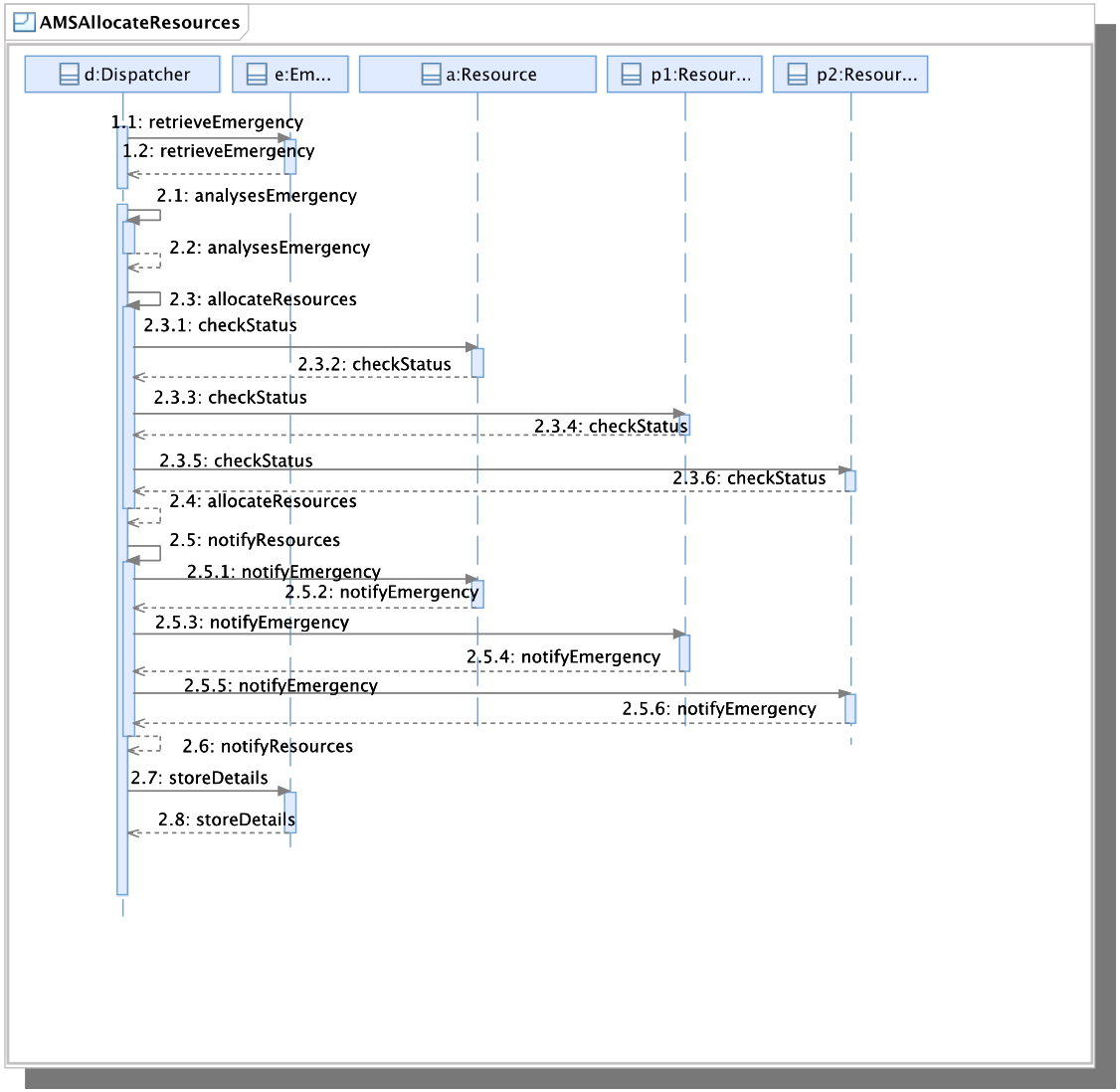
- How do you validate this design against requirements specifications?



Accident Management System UML Sequence Diagram



Accident Management System UML Sequence Diagram



Summary: Sequence Diagrams

- Interaction diagrams
  - modelling dynamic aspects of a system
  - collaboration and sequence diagrams
- Sequence diagrams
  - the interaction logic between the objects in the system in the time order that the interactions take place
  - can be use at multiple levels of abstraction
    - high level
      - to show use case steps
    - low level
      - to show trace of method calls

Summary: Sequence Diagrams

Conceptually

- help with design, understanding the flow of control

Specification

- lots of small methods in different classes
- provides the overall behaviour

Implementation

- creation of code