

# System Modelling using UML Class Diagram and UML Sequence Diagram

Erik Perjons



# Questions to answer

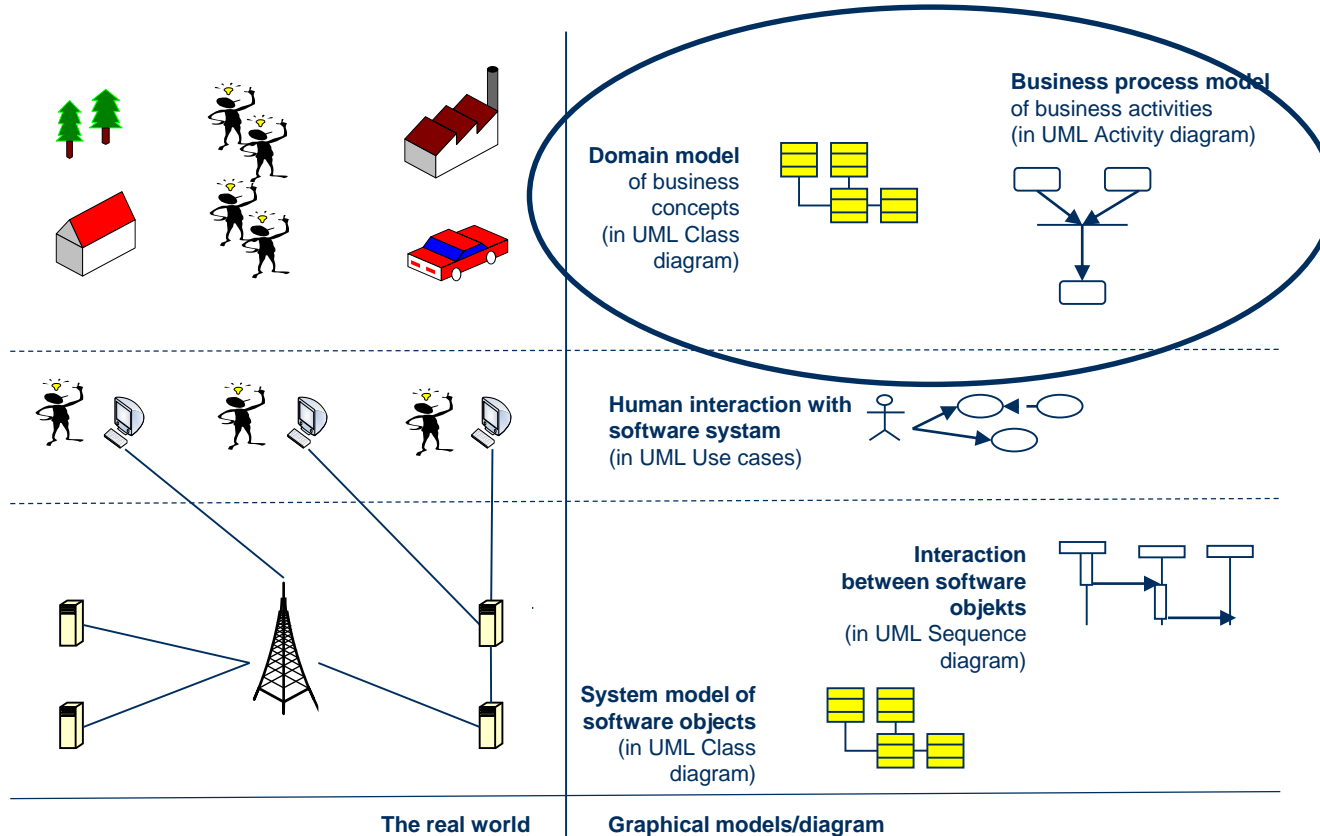
- How can you carry out system modelling?
- How can you identify the software objects and their relationships?
- How can you identify the interactions between software objects?



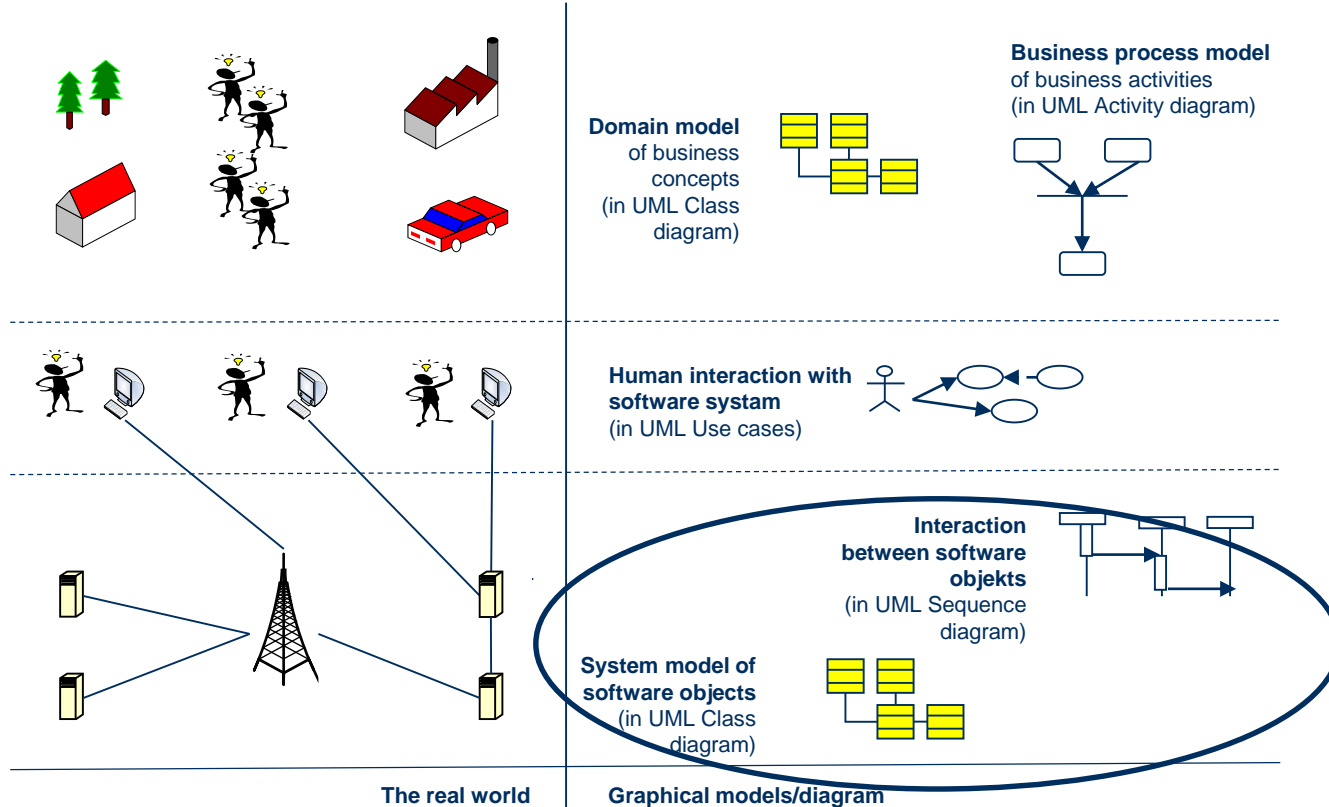
# Enterprise and System Modelling



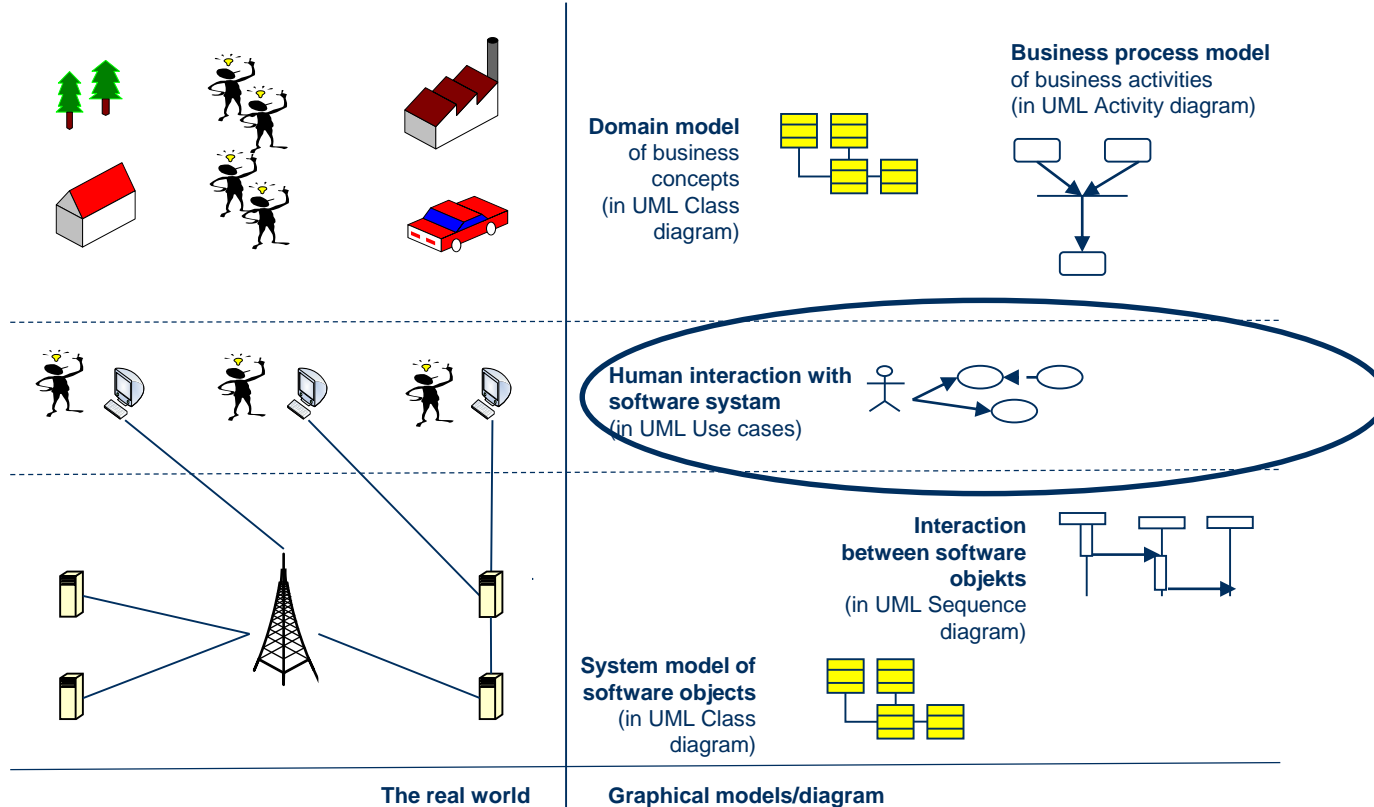
# Real World and Models



# Real World and Models



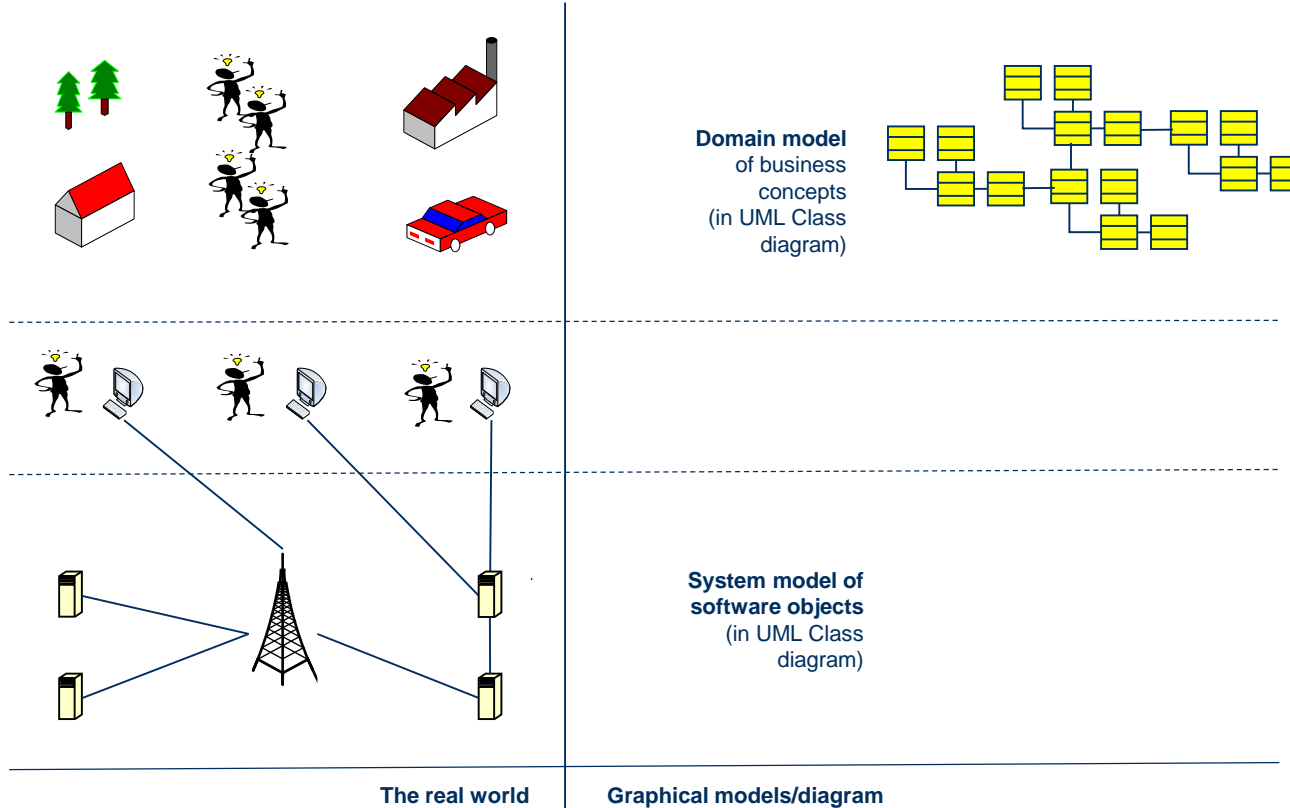
# Real World and Models



# System Modelling – System Model of Software Objects

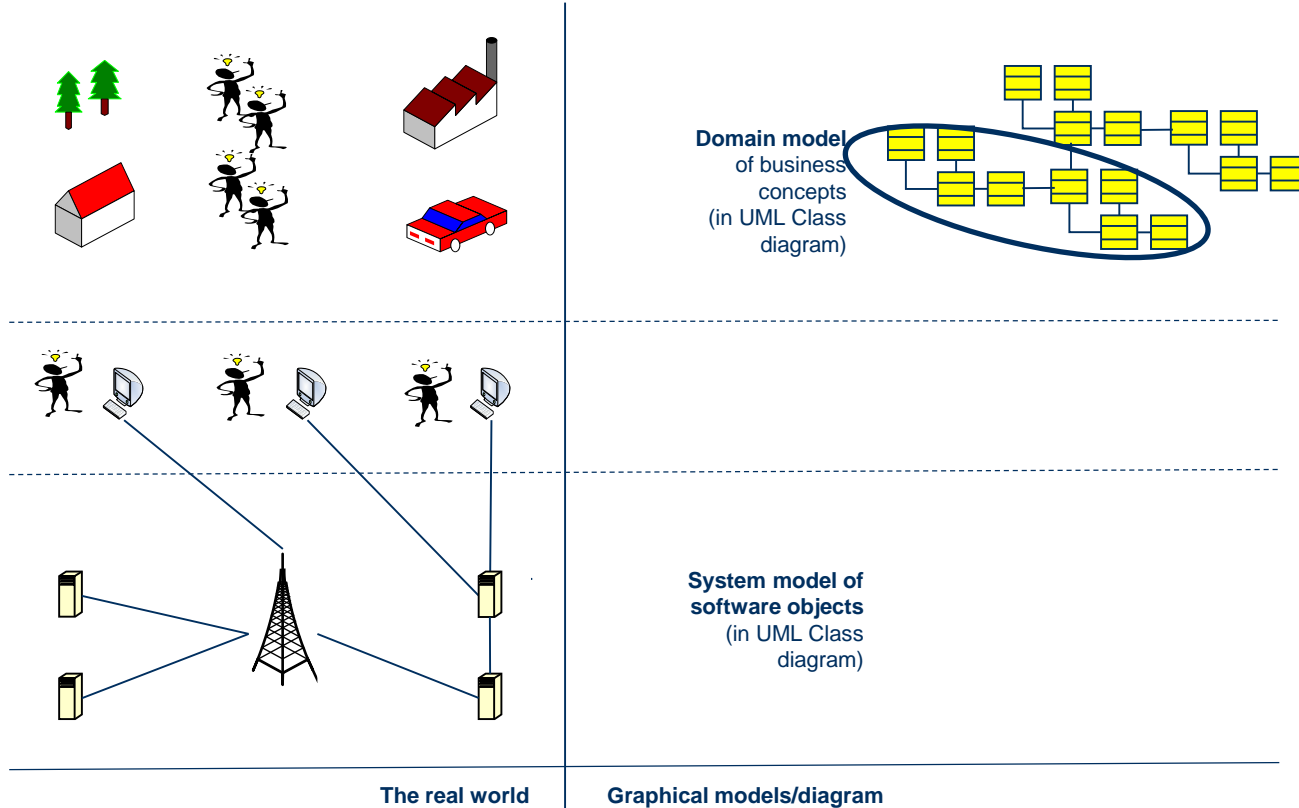


# Real World and Models

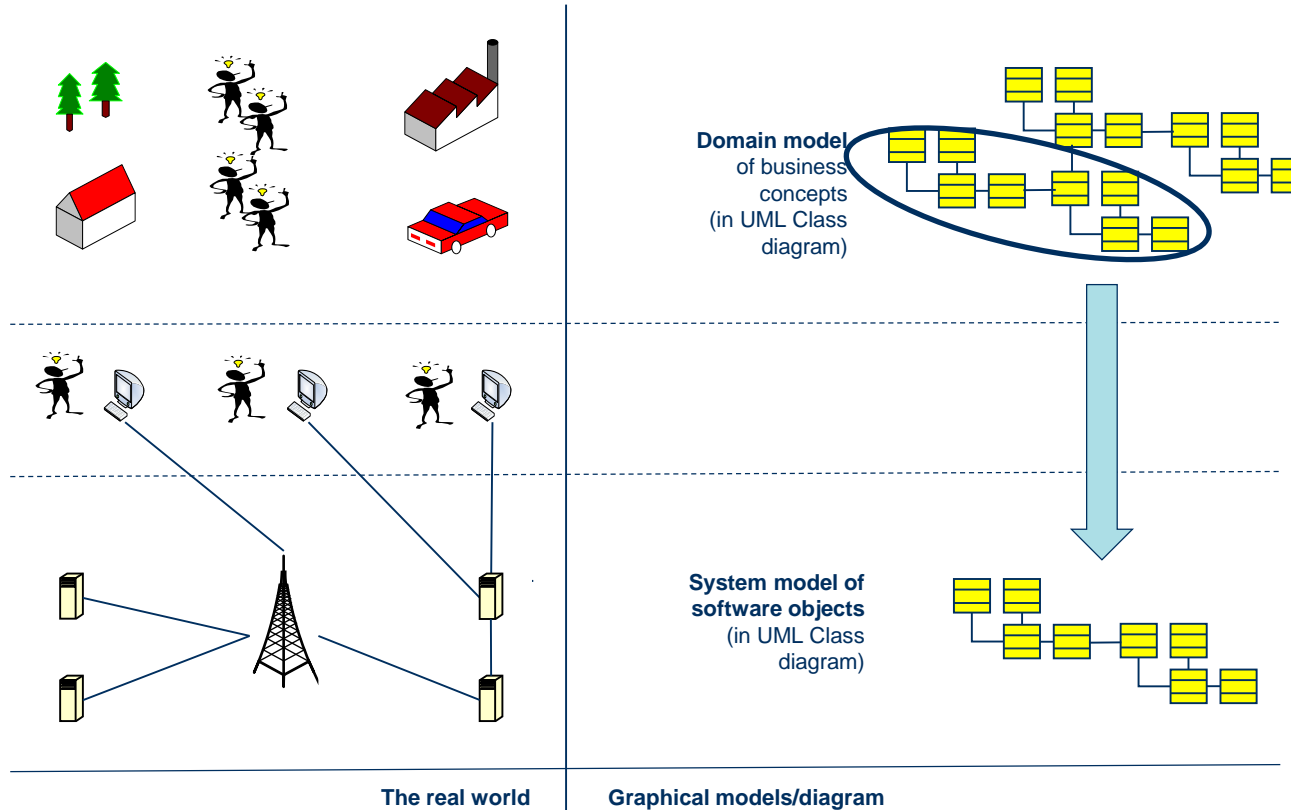




# Real World and Models



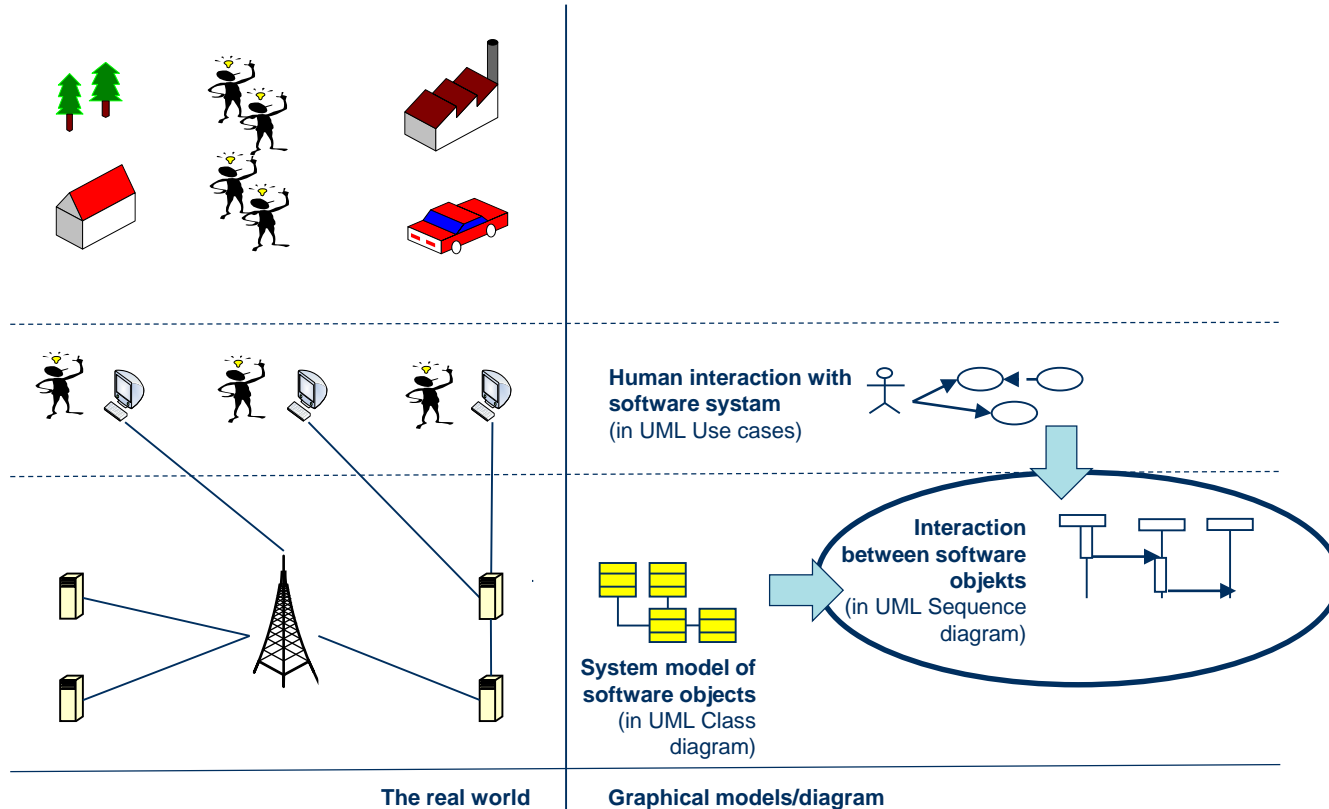
# Real World and Models



# System Modelling – Interaction between software objects



# Real World and Models



# Why show interactions between objects?

- An information system's main tasks are to:
  - store/save information in the system
  - retrieve/read information from the system (to display to the user)
  - perform calculations
- Interaction diagrams (e.g. UML sequence diagrams) represent these tasks in an effective way



# Why show interaction between objects?

- Interaction diagrams (e.g. UML sequence diagrams) provide a good descriptions of how an information systems works
- Therefore, they can be used as one of the first steps to develop an information system
- More precisely, interaction diagrams (e.g. UML sequence diagrams) can be used for identifying which operations (methods) are needed in a system, and in which classes they should belong to



# UML Sequence Diagrams



# Why UML Sequence Diagram?

- Sequence diagram can be used for identifying which operations (methods) are needed in a system, and in which classes (objects) they should belong to





# How do you model a Sequence Diagram?

- In order to design a sequence diagram, you need to decide:
  - which design principles to follow
  - how close the diagram should be to the implementation in a certain programming language

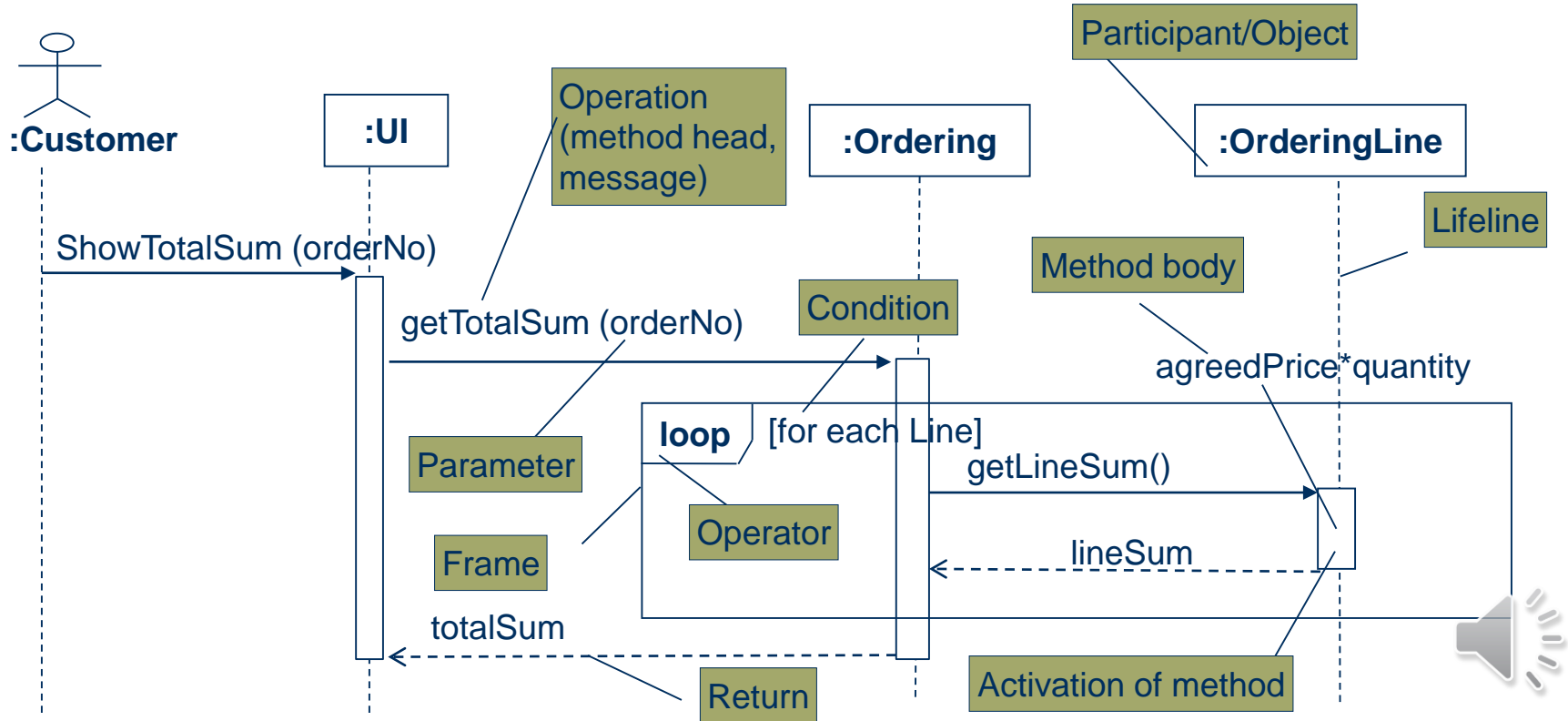


# How do you model a Sequence Diagram?

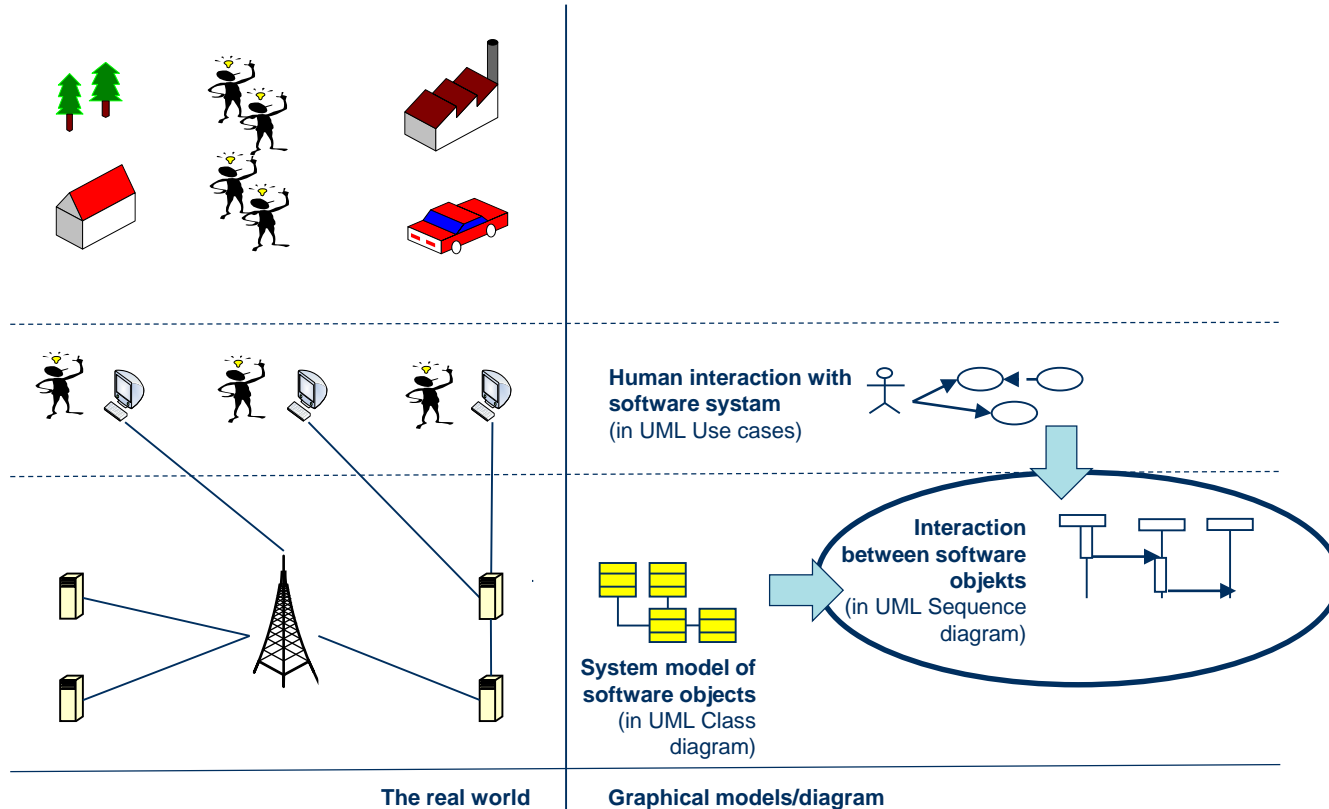
- For the sequence diagrams in this presentation, we follow simple design principles and we will not be close to any implementation/programming language
- Our version of sequence diagrams aims:
  - to identify in which objects that data is store/saved (set)
  - to identify in which objects that data is retrieve/read from (get)
  - to identify in which objects calculations are made



# Sequence Diagram - Notation



# Modelling a Sequence Diagram



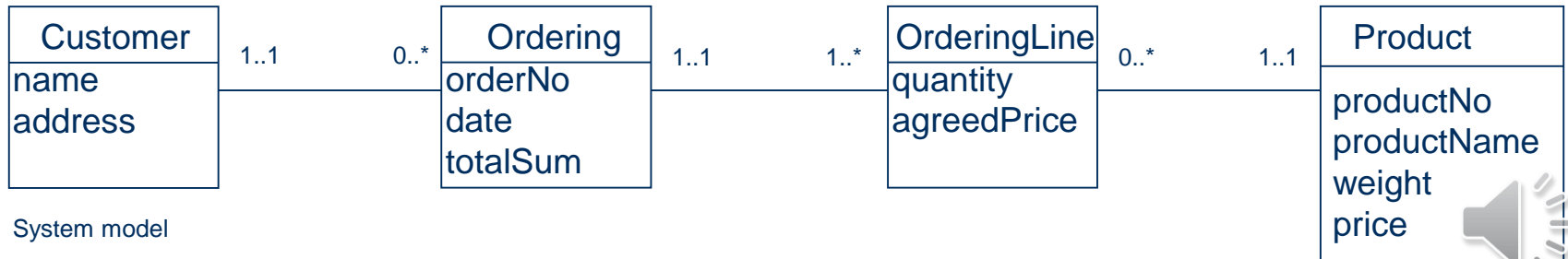
Use Case: Show total sum of order  
 Actor: Customer  
 Goal: Customer shall be provided with the total sum of the order upon request

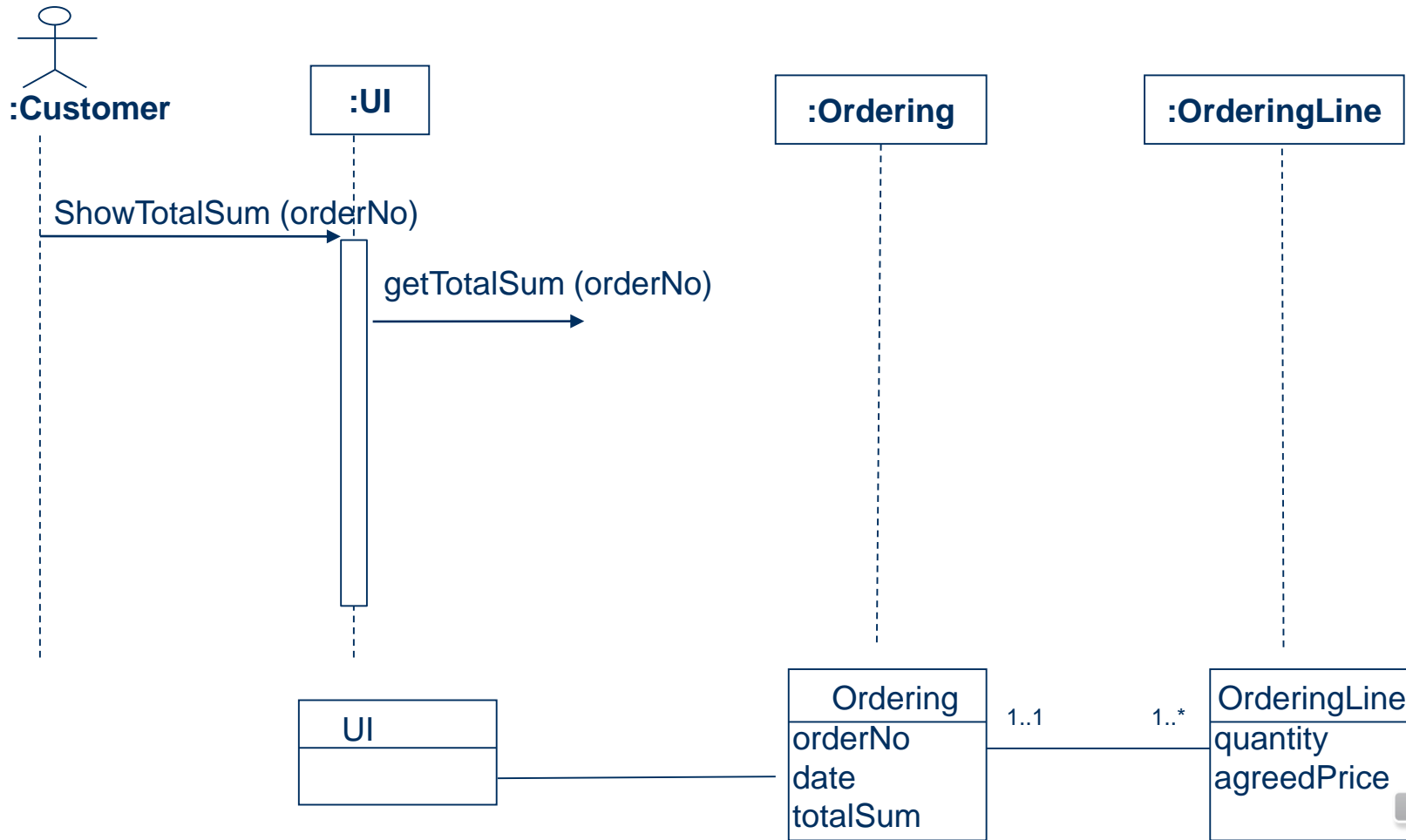
Main Scenario:  
 1) Customer wants to see the total sum of the order  
 2) System presents the total sum of the order

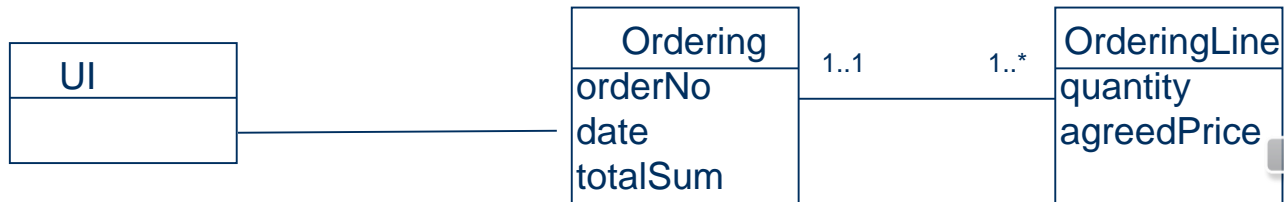
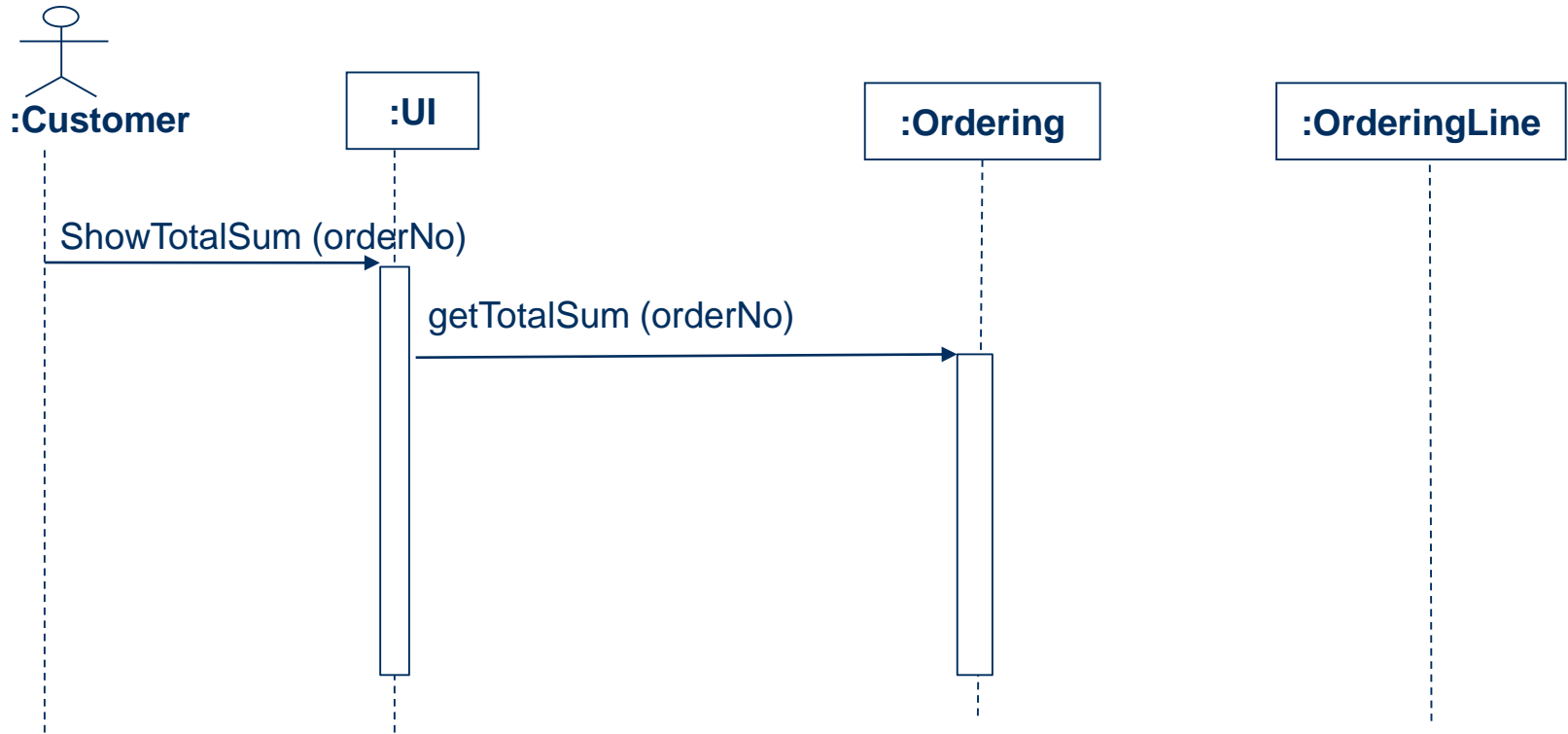


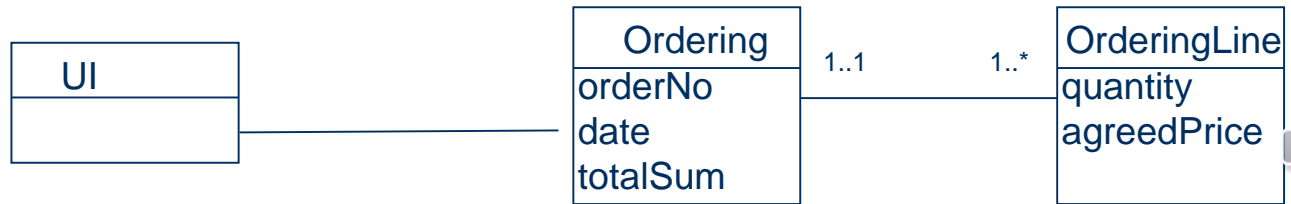
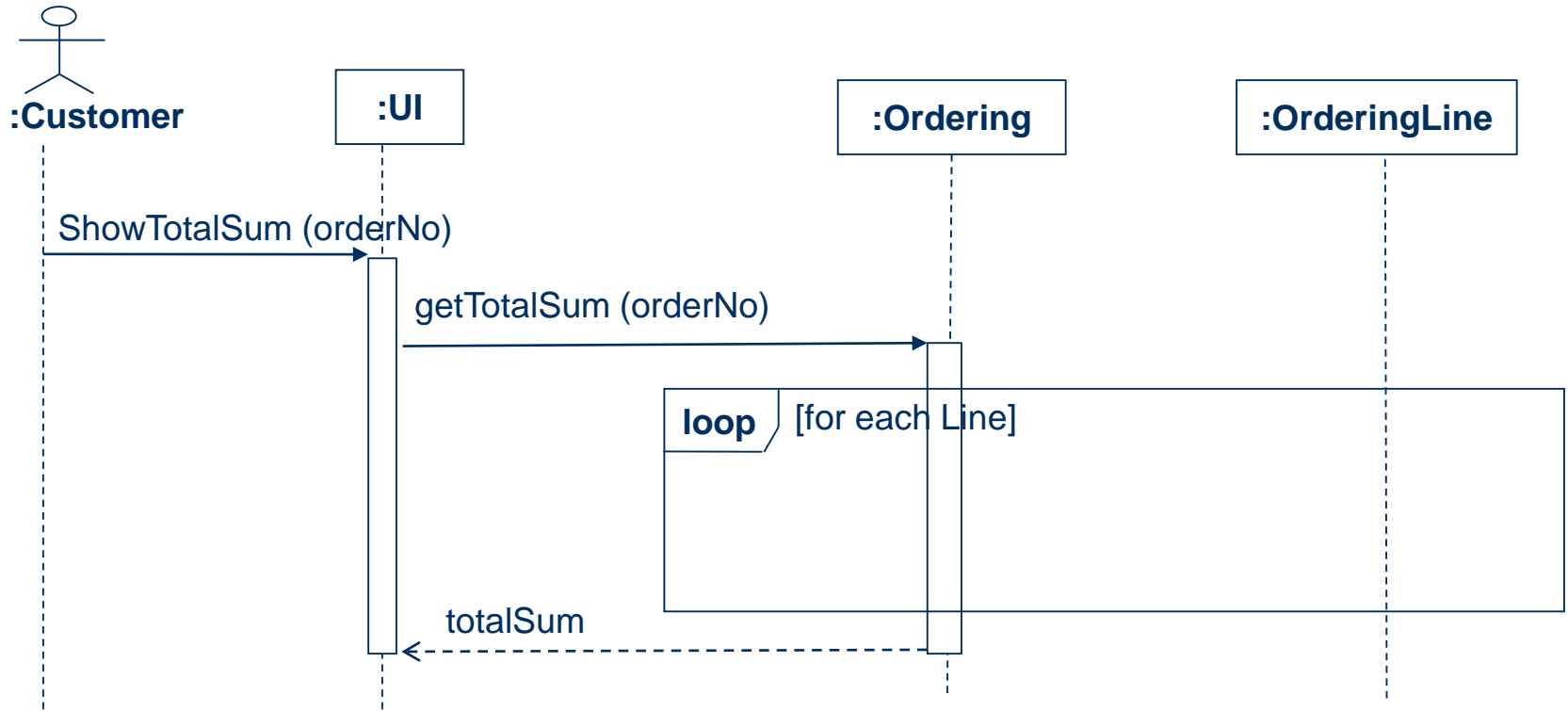
Use Case Description

Use Case Diagram

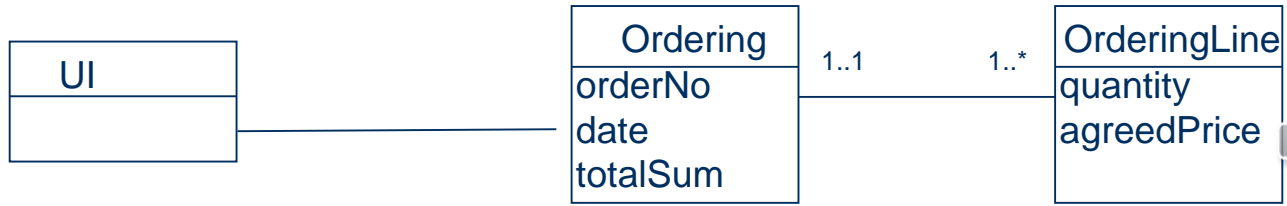
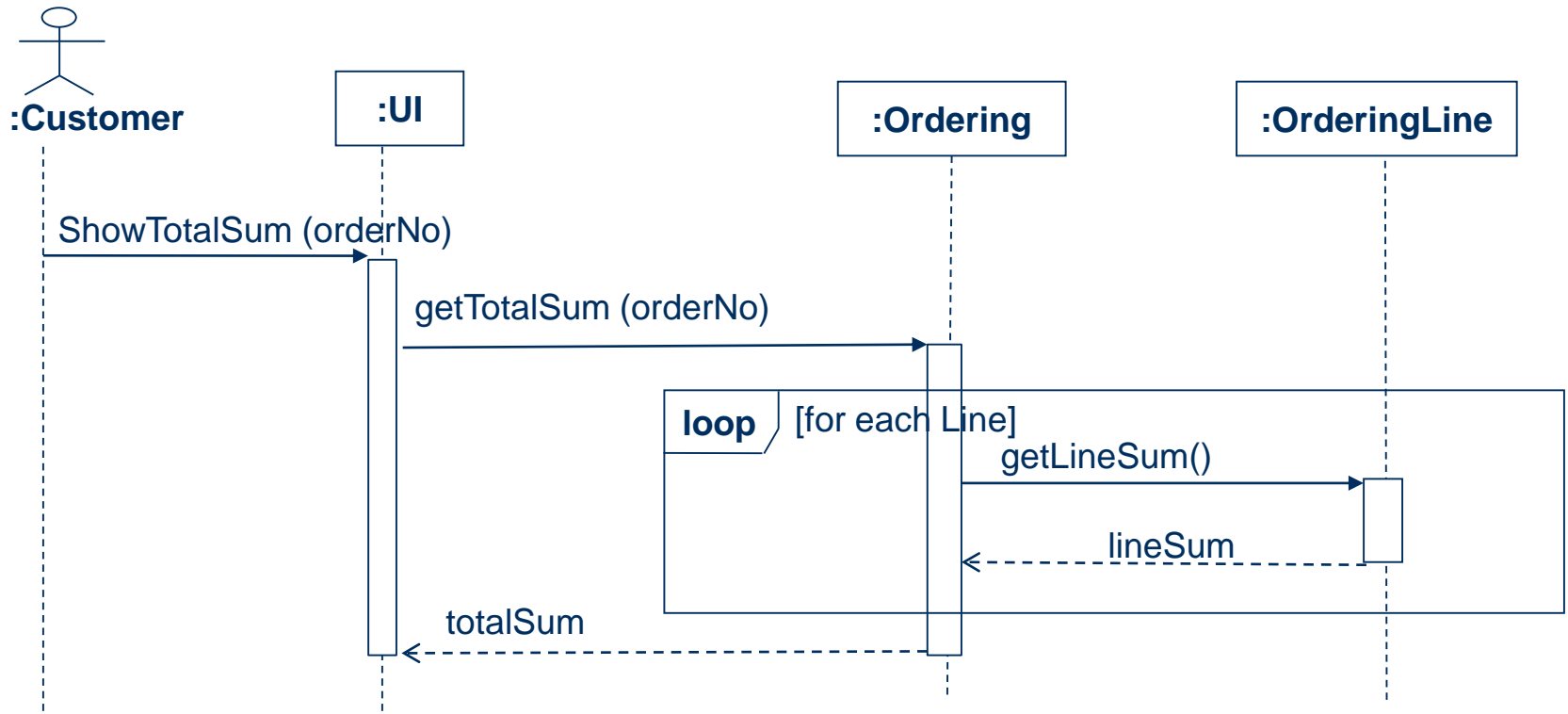


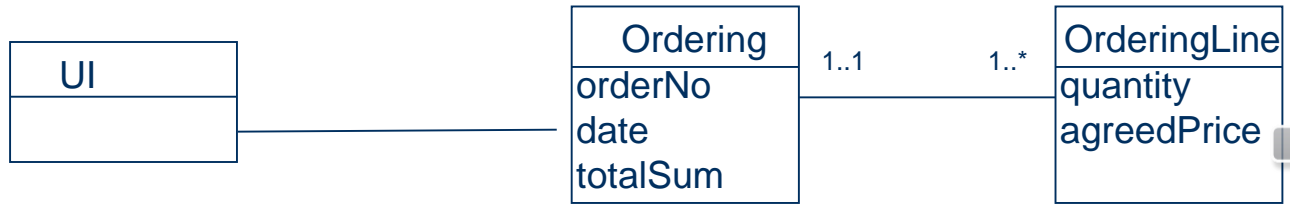
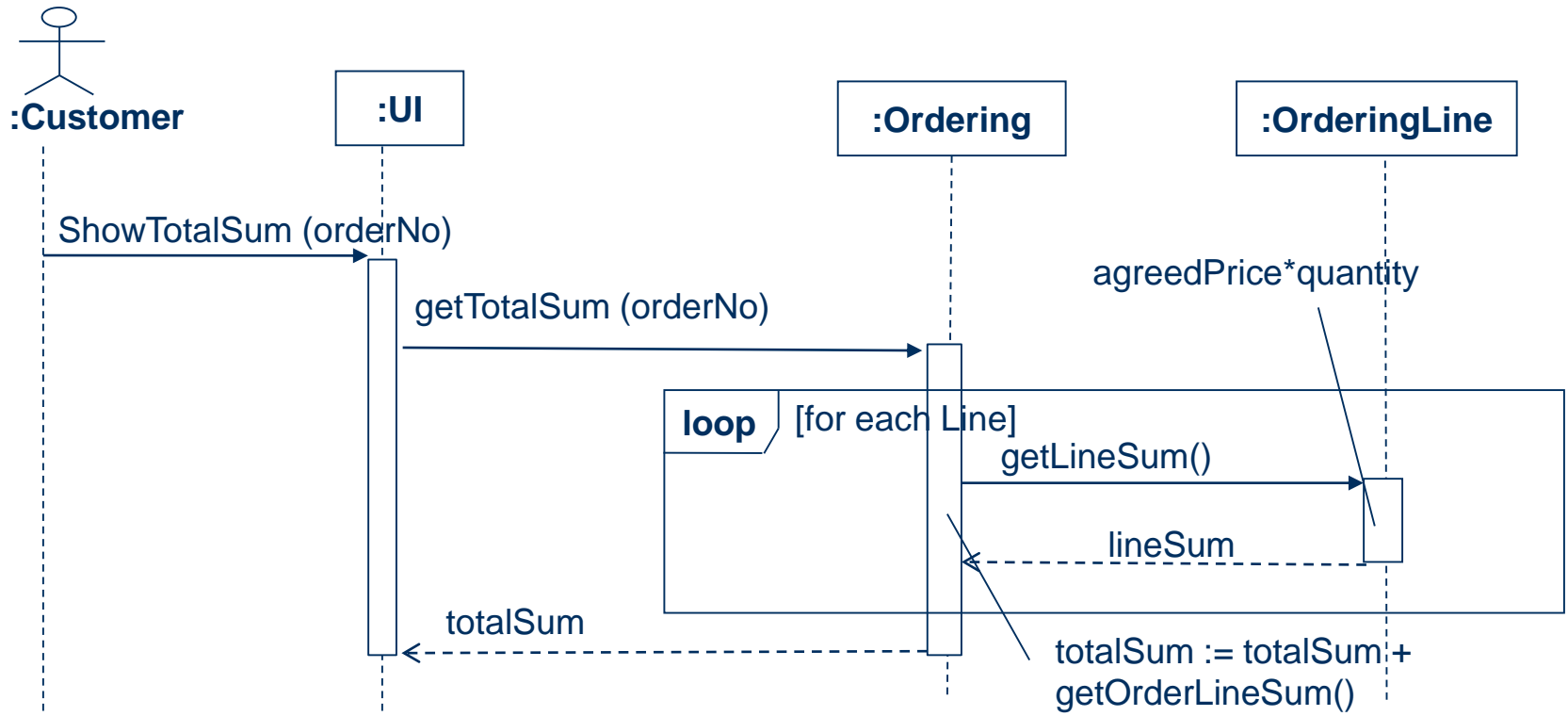


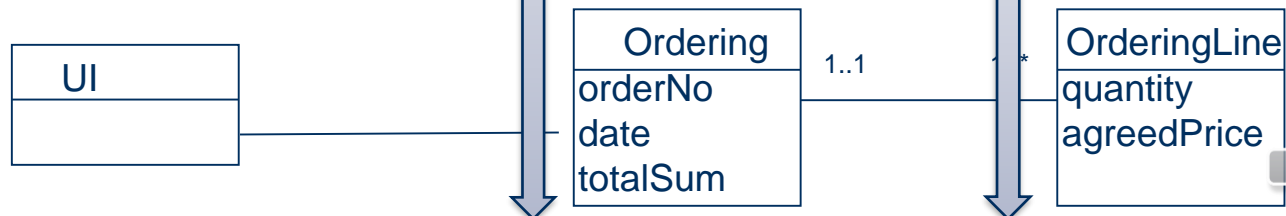
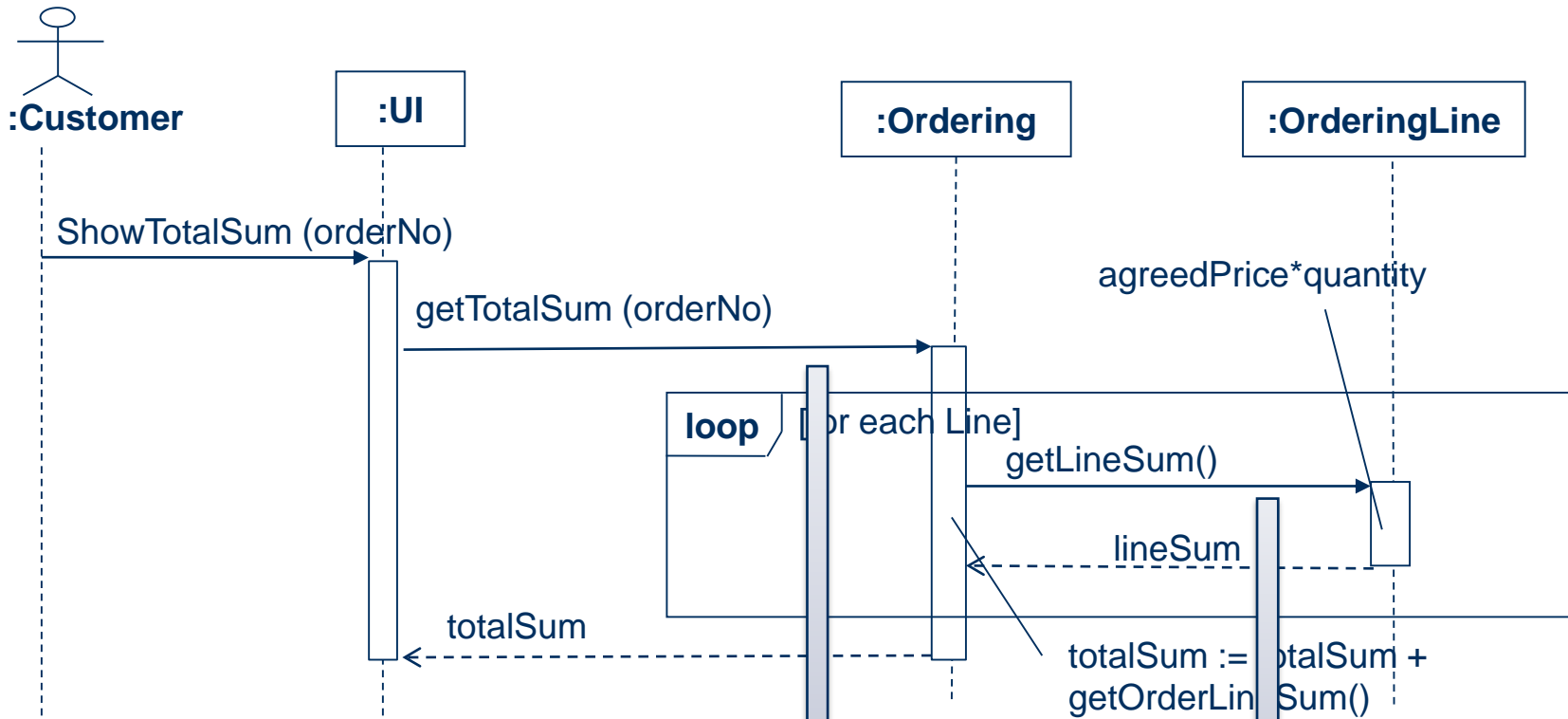






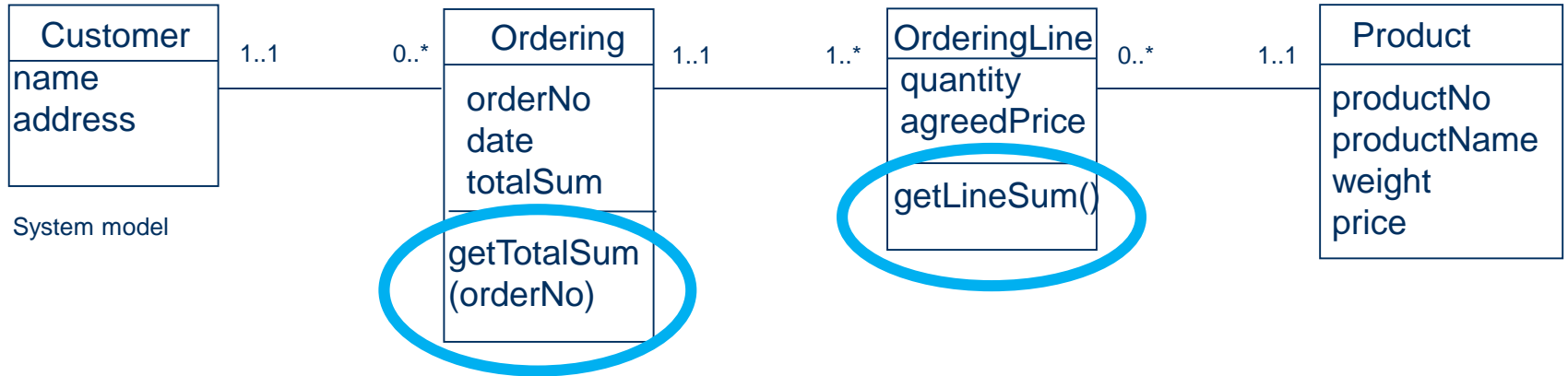






getTotalSum (orderNo)      getLineSum()





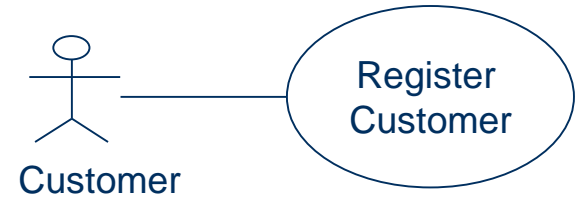
Use Case: Register Customer

Actor: Customer

Goal: Customer shall be registered which is necessary for shopping at the companies web site

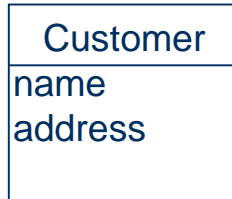
Main Scenario:

- 1) Customer register for company
- 2) System confirm registration



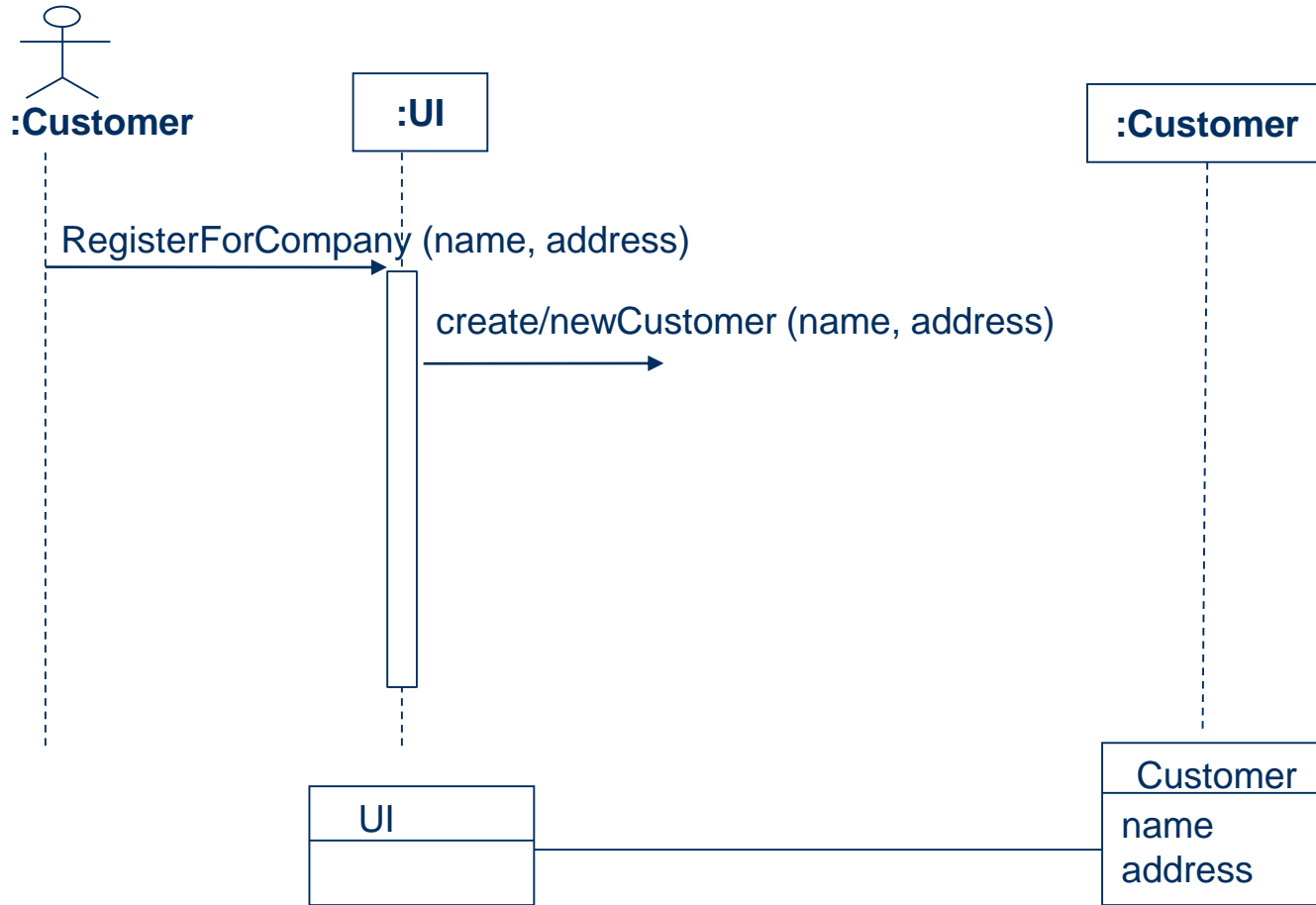
Use Case Description

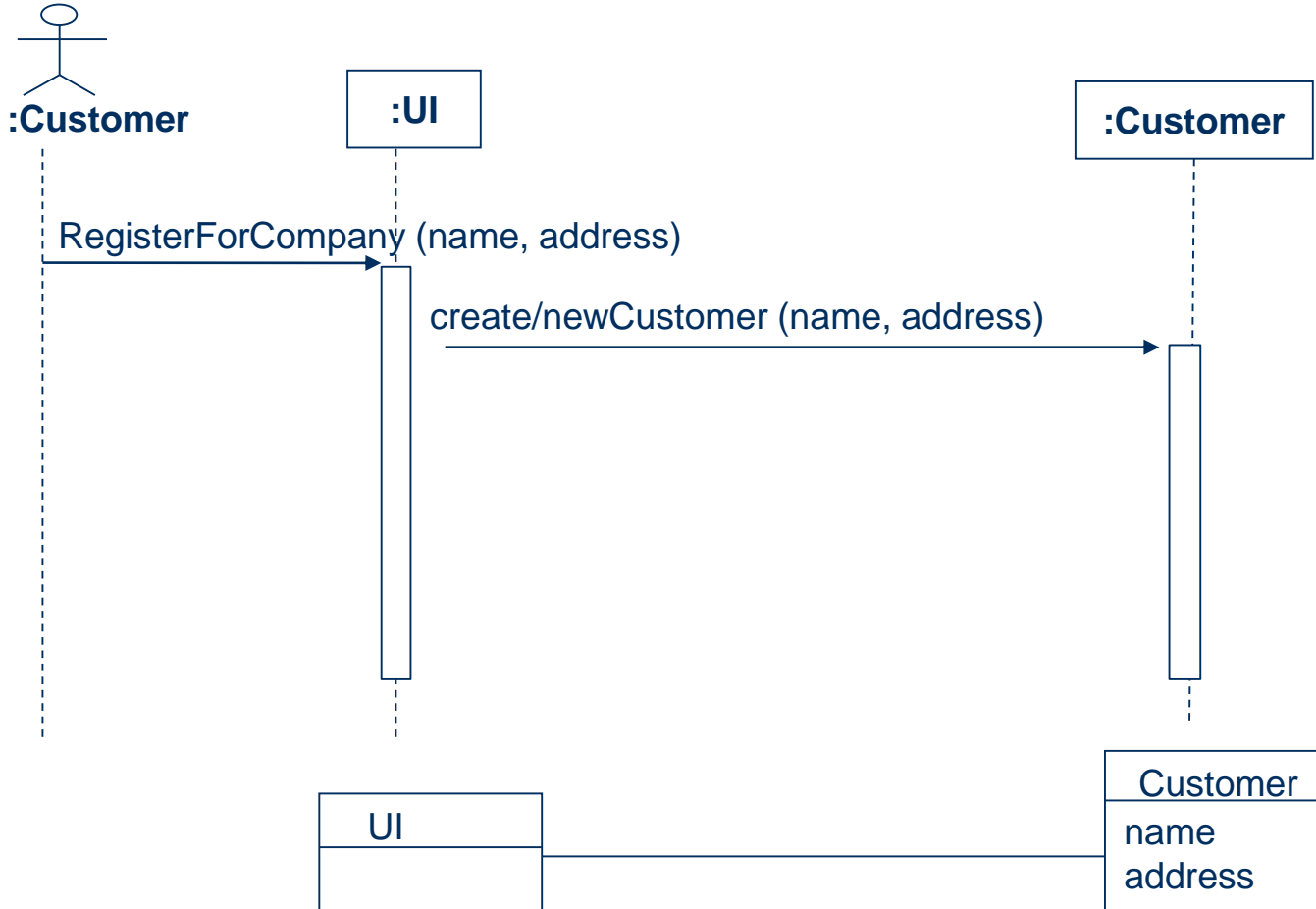
Use Case Diagram

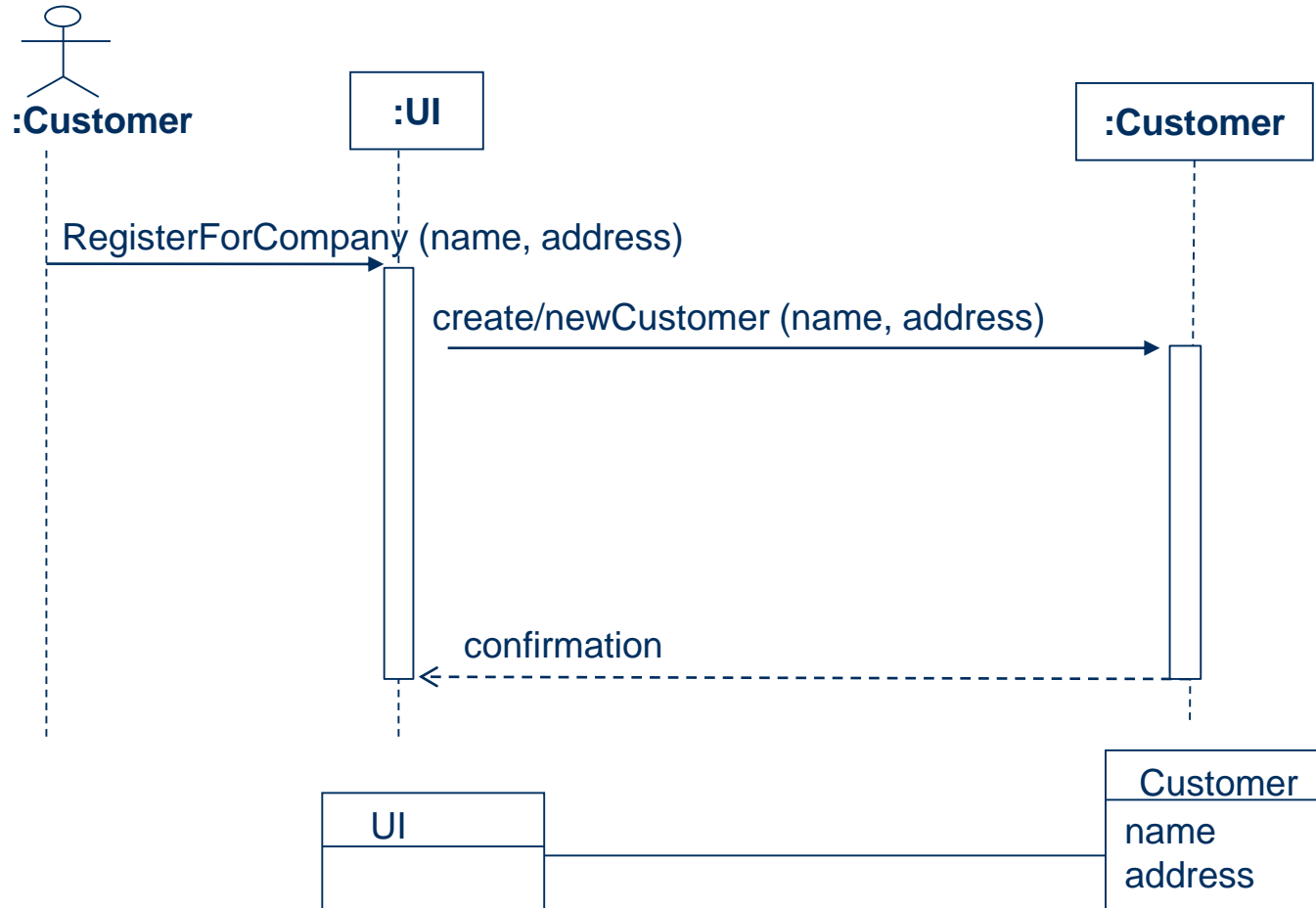


System model

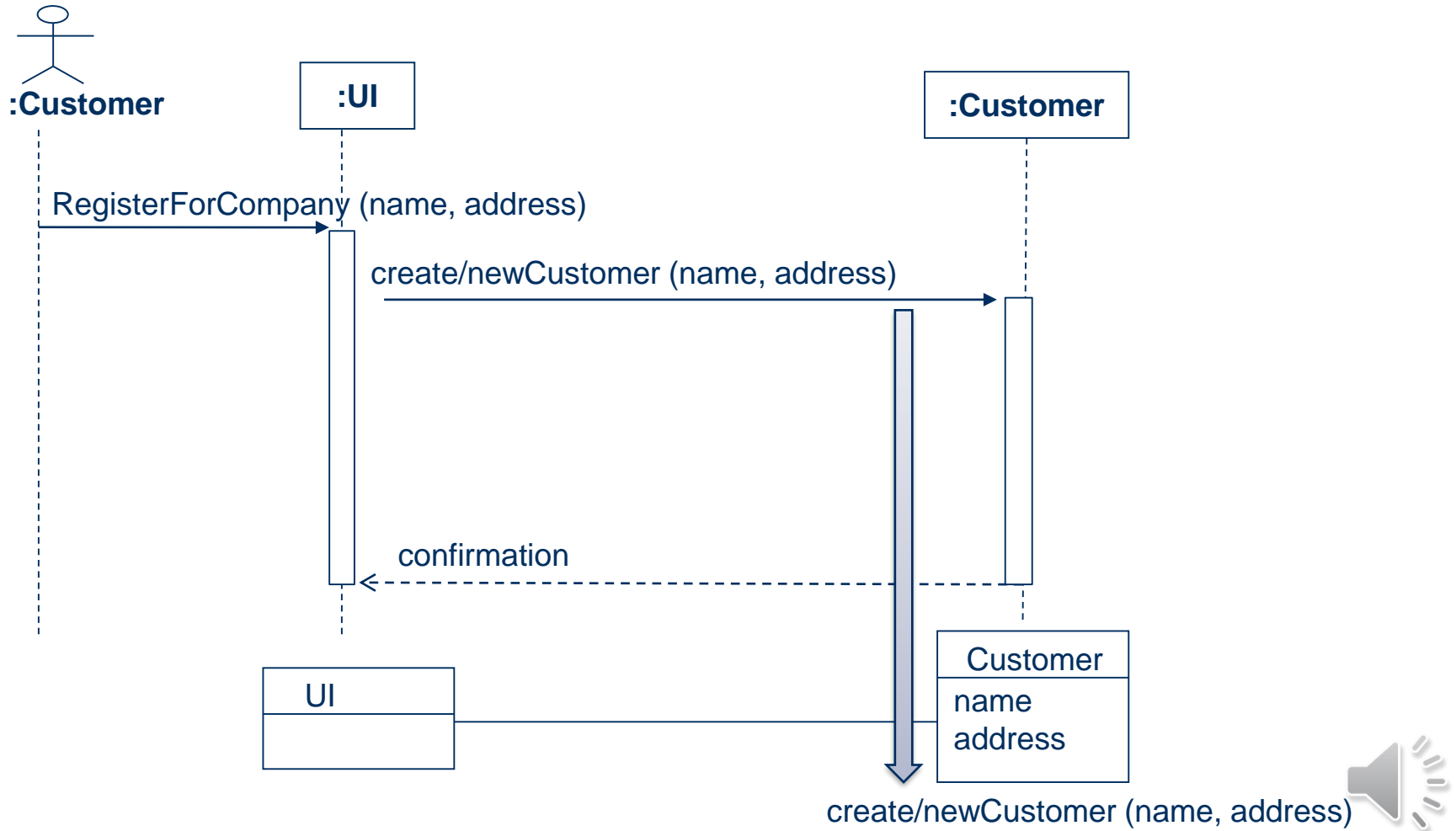


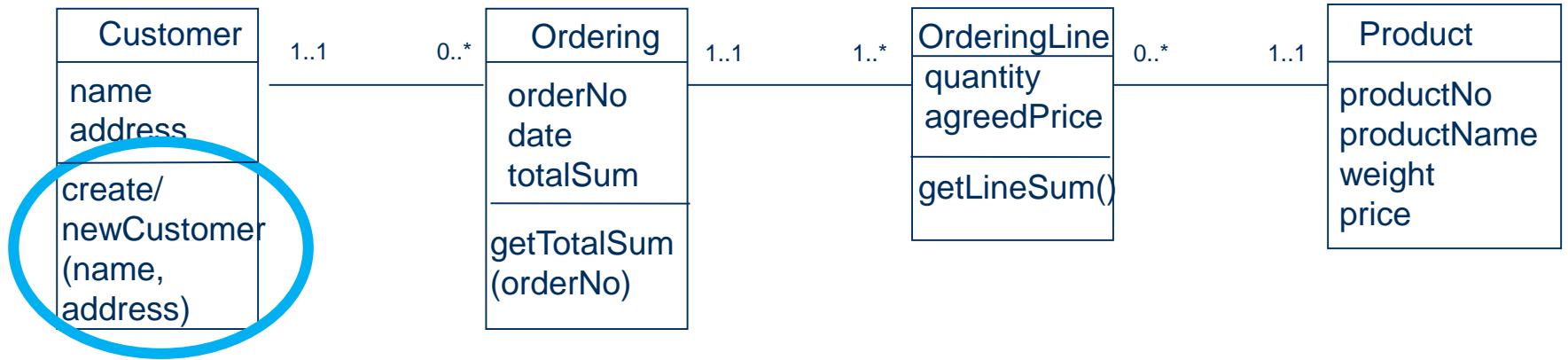






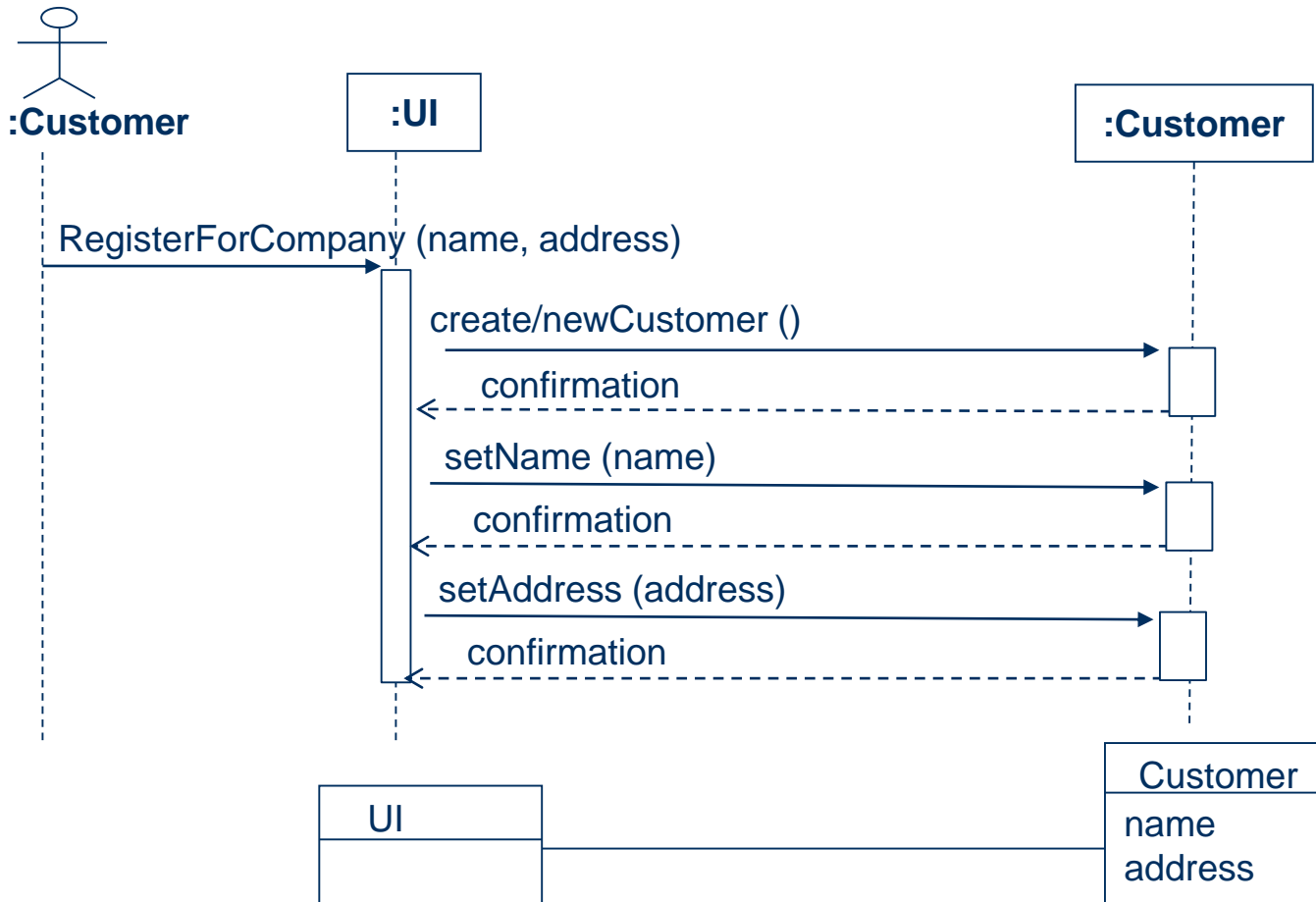


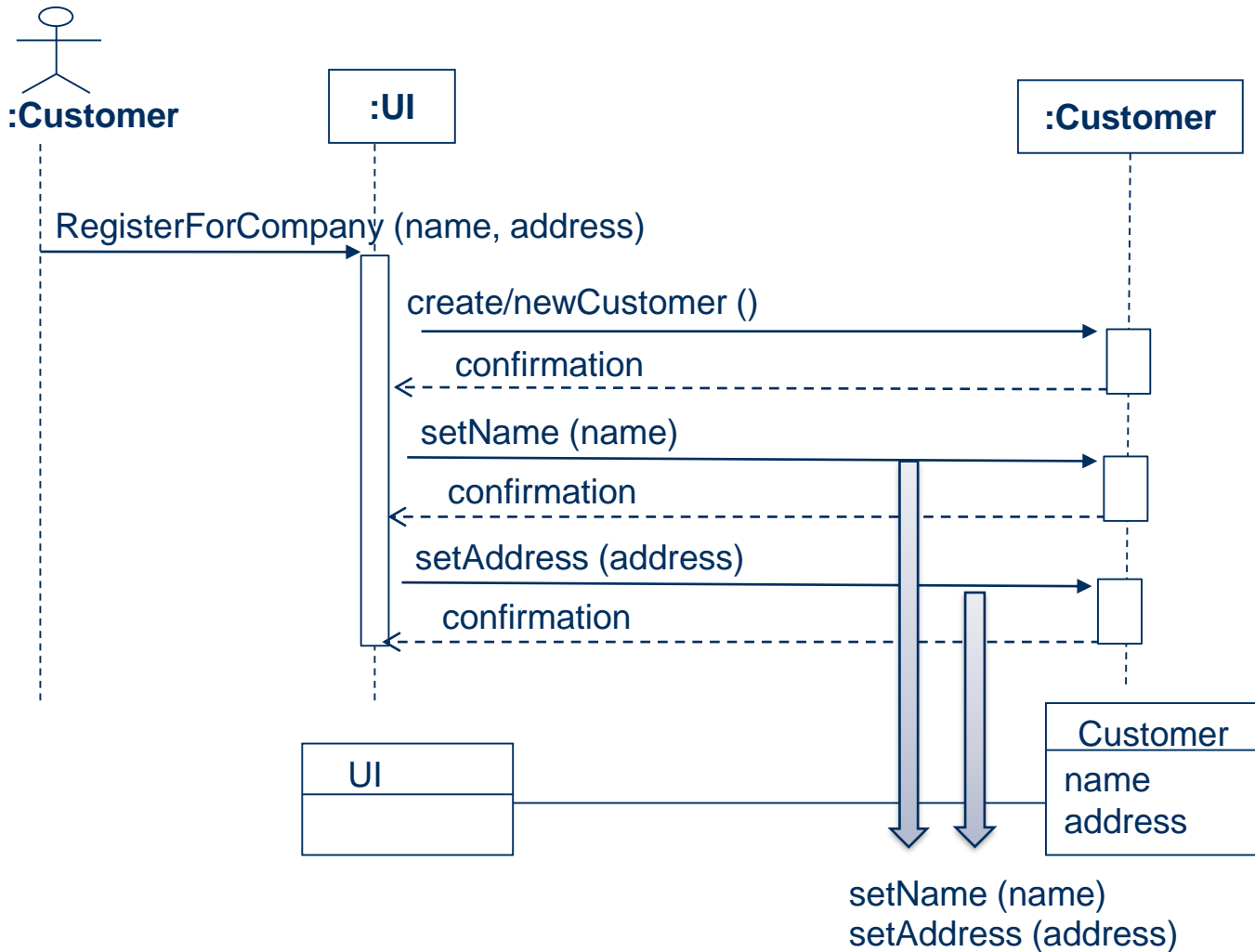




System model







System model

