

## Modeling with UML

### 1) *Actors of an ATM*

Consider an ATM system. Identify at least three different actors that interact with this system.

### 2) *Model an ATM*

Draw a use case diagram for an ATM (cash terminal). The system includes two actors: a customer, who draws money from his account either on the money chip on his bank card or in cash, and a security man, who fills money into the ATM.

Use cases should include: DrawCash, LoadMoneyChip, CheckAccountBalance, FillATM. Also include the following exceptional cases: OutOfMoney, TransactionAborted (i.e., customer selected the cancel button without completing the transaction) and MoneyChipOutOfOrder.

Remark: It is possible to use inheritance between use cases!

Write the flow of events and specify all fields for the use case CheckAccountBalance that you drew before. Specify relationships.

### 3) *System actors*

Can the system under consideration be represented as an actor? Justify your answer.

### 4) *Scenarios and Use Cases*

What is the difference between a scenario and a use case? When do you use each construct?

### 5) *Ticket Distributor*

Draw a use case diagram for a ticket distributor for a train system. The system includes two actors: a traveler, who purchases different types of tickets, and a central computer system, which maintains a reference database for the tariff. Use cases should include:

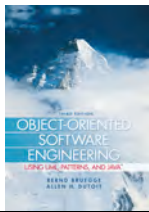
BuyOneWayTicket, BuyWeeklyCard, BuyMonthlyCard, UpdateTariff. Also include the following exceptional cases: Time-Out (i.e., traveler took too long to insert the right amount), TransactionAborted (i.e., traveler selected the cancel button without completing the transaction), DistributorOutOfChange, and DistributorOutOfPaper.

### 6) *Update Tariff*

Write the flow of events and specify all fields for the use case UpdateTariff that you drew in Exercise 4. Do not forget to specify any relationships.

### 7) *Model a Sudoku puzzle*

Draw a class diagram for the following statement: "A Sudoku puzzle is a grid, that consists of 3x3 boxes which in turn consist of 3x3 fields. A field can contain numbers from 1 to 9. Persons solve Sudoku puzzles."



## 8) Modeling a book

### a) Class Diagram

Draw a class diagram representing a book defined by the following statement: “A book is composed of a number of parts, which in turn are composed of a number of chapters. Chapters are composed of sections.” Focus only on classes and associations.

### b) Multiplicities

Add multiplicity to the class diagram you produced in a).

## 9) Modeling the OOSE Book

Draw an object diagram representing the first part of the OOSE book containing the following content:

Book: OOSE

Table of Contents:

Part 1: Getting Started

Chapter 1: Introduction of Software Engineering

1.1 Introduction: Software Engineering Failures

1.2 What is Software Engineering

1.3 Software Engineering Concepts

1.4 Software Engineering Development Activities

1.5 Managing Software Development

1.6 Exercises

References

Chapter 2: Modeling with UML

2.1 Introduction

2.2 An Overview of UML

2.3 Modeling Concepts

2.4 A Deeper View into UML

2.5 Exercises

References

Chapter 3: Project Communication

3.1 Introduction

3.2 An Overview of Project Communication

3.3 Models of Communication

3.4 Mechanisms of Communication

3.5 Project Communication Activities

3.6 Exercises

References

Make sure that the object diagram you draw is consistent with the class diagram of Exercise 8).

## 10) Refining the book

### a) Attributes of the book

Extend the class diagram of Exercise 6 to include the following attributes:

- book includes a publisher, publication date, and an ISBN
- a part includes a title and a number
- a chapter includes a title, a number, and an abstract
- a section includes a title and a number



### **b) Generalization**

Consider the class diagram of a). Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an class and an inheritance association to factor out these two attributes into the new class.

## **11) Modeling a watch**

### **a) Class diagram**

Draw a class diagram for the following statement: “Persons wear watches: Some wear digital watches, others wear traditional watches. All watches have batteries, digital watches show the time on a LCD display, whereas traditional watches use two watch hands.”

### **b) Instance diagram**

Draw an UML instance diagram for the class diagram you developed in a) showing two persons: Bob and Alice. Bob wears a traditional watch and Alice a digital watch. Make sure, that your diagram is consistent with the diagram in a).

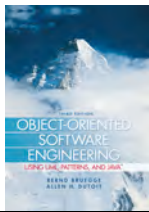
## **12) Parents and Children**

Draw a class diagram representing the relationship between parents and children. Take into account that a person can have both a parent and a child. Annotate associations with roles and multiplicities.

## **13) Warehouse on Fire**

Draw a sequence diagram for the WarehouseOnFire scenario in Figure 1. Include the objects bob, alice, john, FRIEND, and instances of other classes you may need. Draw only the first five messages sent.

Scenario name	WarehouseOnFire
Participating actor instances	<u>bob, alice: FieldOfficer</u> <u>john: Dispatcher</u>
Flow of events	<ol style="list-style-type: none"><li>1. Bob, driving down main street in his patrol car, notices smoke coming out of a warehouse. His partner, Alice, activates the “Report Emergency” function of her FRIEND laptop.</li><li>2. Alice enters the address of the building, a brief description of its location (i.e., northwest corner), and an emergency level. In addition to a fire unit, she requests several paramedic units given that area appears to be quite busy. She confirms her input and waits for an acknowledgment.</li><li>3. John, the Dispatcher, is alerted to the emergency by a beep of his workstation. He reviews the information</li></ol>



	<p>submitted by Alice and acknowledges the report. He allocates a fire unit and two paramedic units to the incident site and sends their estimated arrival time (ETA) to Alice.</p> <p>4. Alice receives the acknowledgment and the ETAs.</p>
--	---

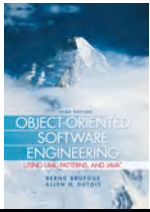
**Figure 1**

### 14) Report Emergency

Draw a sequence diagram for the `ReportEmergency` use case of Figure 2. Draw only the first five messages sent. Make sure it is consistent with the sequence diagram of Exercise 1.

Use case name	ReportEmergency	
Participating Actors	Initiated by <code>FieldOfficer</code> Communicates with <code>Dispatcher</code>	
Flow of events	1. The <code>FieldOfficer</code> activates the “Report Emergency” function on her terminal.	
		2. FRIEND responds by presenting a form to the <code>FieldOfficer</code> .
	3. The <code>FieldOfficer</code> fills in the form by selecting the emergency level, type and location and entering a brief description of the situation. The <code>FieldOfficer</code> also describes possible responses to the emergency situation. Once the form is completed, the <code>FieldOfficer</code> submits the form.	
		4. FRIEND receives the form and notifies the <code>Dispatcher</code> .
	5. The <code>Dispatcher</code> reviews the submitted information and creates an <code>Incident</code> in the database by invoking the <code>OpenIncident</code> use case. The <code>Dispatcher</code> selects a response and acknowledges the report.	
		6. FRIEND displays the acknowledgment and the selected response to the <code>FieldOfficer</code> .
Entry condition	<ul style="list-style-type: none"> <li>The <code>FieldOfficer</code> is logged into FRIEND.</li> </ul>	
Exit condition	<ul style="list-style-type: none"> <li>The <code>FieldOfficer</code> has received an acknowledgment and the selected response from the <code>Dispatcher</code>, OR</li> <li>The <code>FieldOfficer</code> has received an explanation indicating why the transaction could not be processed.</li> </ul>	
Quality requirements	<ul style="list-style-type: none"> <li>The report of the <code>FieldOfficer</code> report is acknowledged within 30 seconds.</li> <li>The selected response arrives no later than 30 seconds after it is sent by the <code>Dispatcher</code>.</li> </ul>	

**Figure 2**



## **15) Pizza Service**

### **a) Ordering a pizza**

Consider the process of ordering a pizza by phone. Draw an activity diagram representing each step of the process, from the moment you pick up the phone to the point when you start eating the pizza. Do not represent any exceptions. Include activities that others need to perform.

### **b) Exceptions**

Add exception handling to the activity diagram you developed in a). Consider at least three exceptions (e.g., operator wrote down wrong address, delivery man delivers wrong pizza, store is out of anchovies).