LECTURER: JOHN DOE

SPECIFICATION

SPECIFICATION
TOPIC OUTLINE

Introduction to Software Requirements Specification (SRS)	1
Specification of User Interfaces (GUIs)	2
Specification of System Components	3
Specification of Technical System Interfaces	4
Specification of Detailed Conceptual Data Models	5
Using Structured Text in the Specification of Data Interfaces Specification of Quality Requirements	6

UNIT 5

SPECIFICATION OF DETAILED CONCEPTUAL DATA MODELS

STUDY GOALS

- Explain where conceptual data models can be used in the specification process.
- Outline the characteristics by which conceptual data models can be extended and refined.
- Describe how data models can be tested.

EXPLAIN SIMPLY

1. Explain the relationship between requirements engineering and a conceptual data model.
2. Why is the class diagram not suitable for the representation of processes?
3. Which elements are contained in conceptual data models?

APPLICATION AREAS OF CONCEPTUAL DATA MODELS

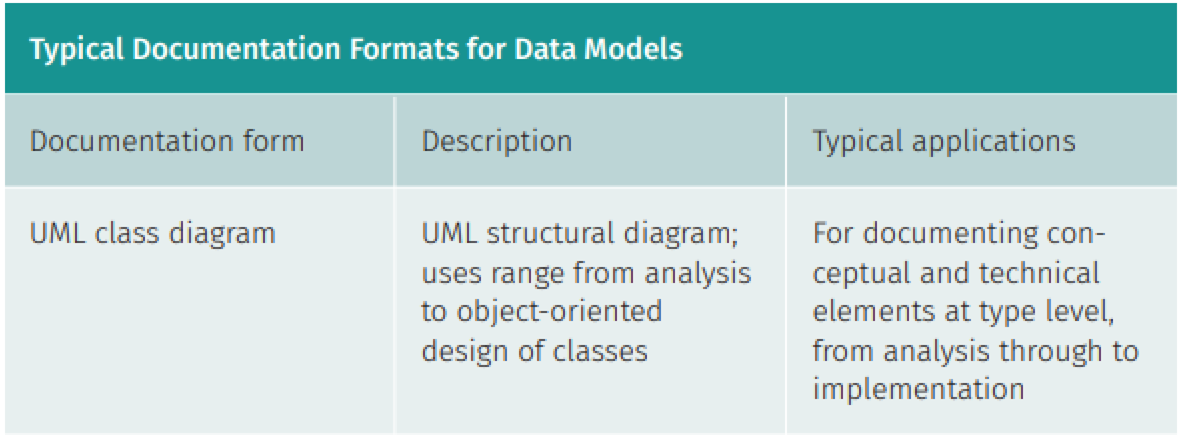
Conceptual data models specify conceptual classes (business objects) and their relationships.

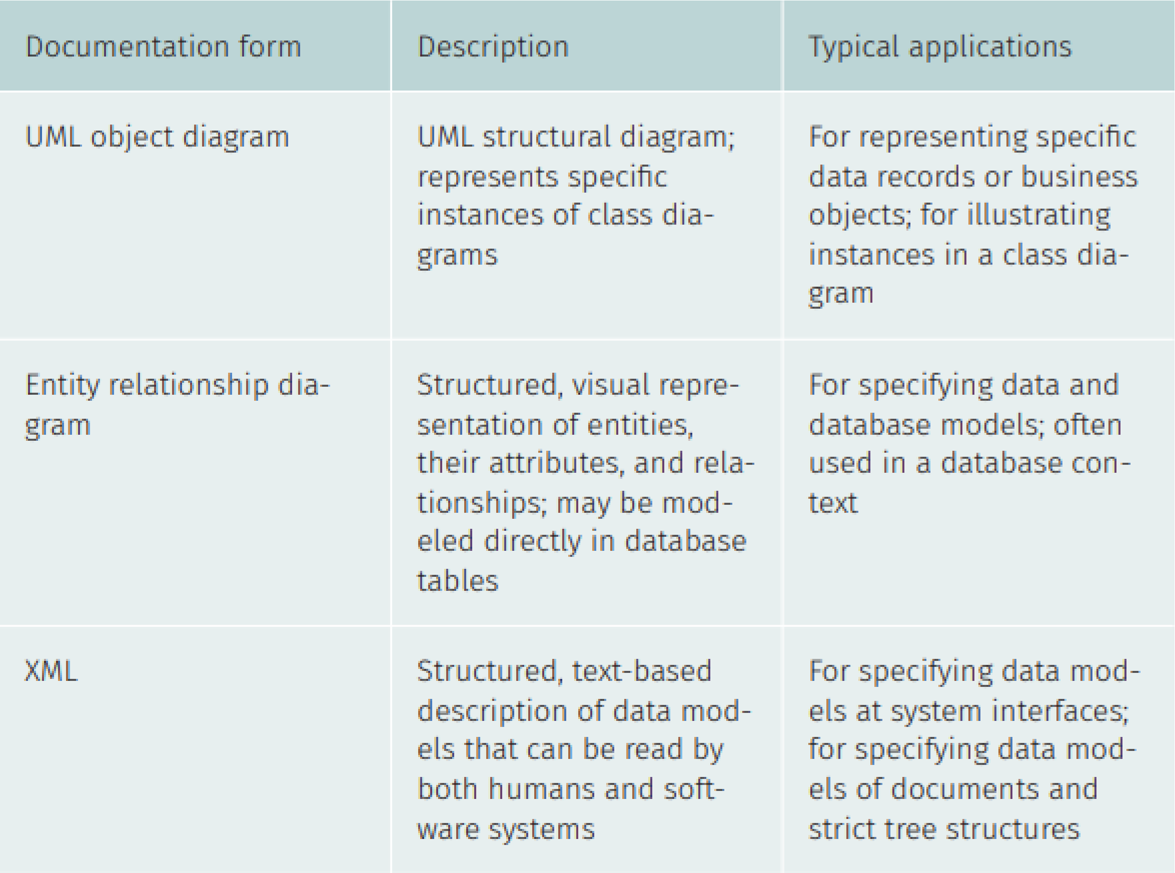
Different types of data models can be distinguished:

- Data models for **component and system behavior** (conceptual model): **Business objects** whose properties and interrelationships are documented. Modeled classes give a compact overview. Process models specify the HOW of a system, data models specify the WHAT.
- Data models for **GUI specification**: Considers all relevant elements from the data model when **constructing the GUI**. Examples include the number of input or output elements, specifications for validation, set of values, etc.
- Data model for **technical interfaces**: Specification of technical interfaces, i.e., direct exchange of messages. Coordination of the structure of the message between participating systems and **definition of a standard**.

DOCUMENTATION FORMS FOR DATA MODELS

Strengths and weaknesses of UML diagrams for behavior specification:





Typical Documentation Formats for Data Models

Documentation form	Description	Typical applications
UML class diagram	UML structural diagram; uses range from analysis to object-oriented design of classes	For documenting conceptual and technical elements at type level, from analysis through to implementation
UML object diagram	UML structural diagram; represents specific instances of class diagrams	For representing specific data records or business objects; for illustrating instances in a class diagram
Entity relationship diagram	Structured, visual representation of entities, their attributes, and relationships; may be modeled directly in database tables	For specifying data and database models; often used in a database context
XML	Structured, text-based description of data models that can be read by both humans and software systems	For specifying data models at system interfaces; for specifying data models of documents and strict tree structures

Image source: Coursebook.

KEY DETAILS OF THE UML CLASS DIAGRAM

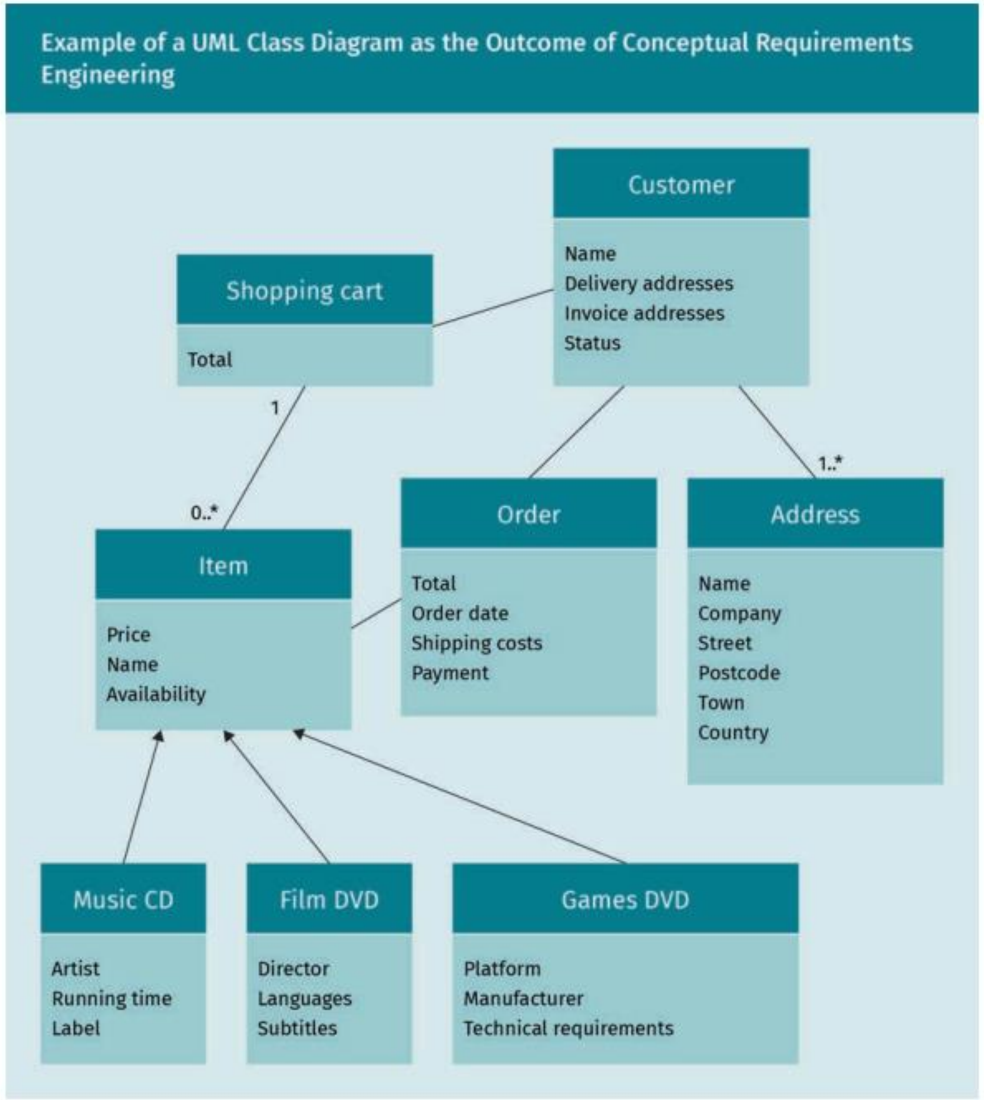


Image source: Coursebook.

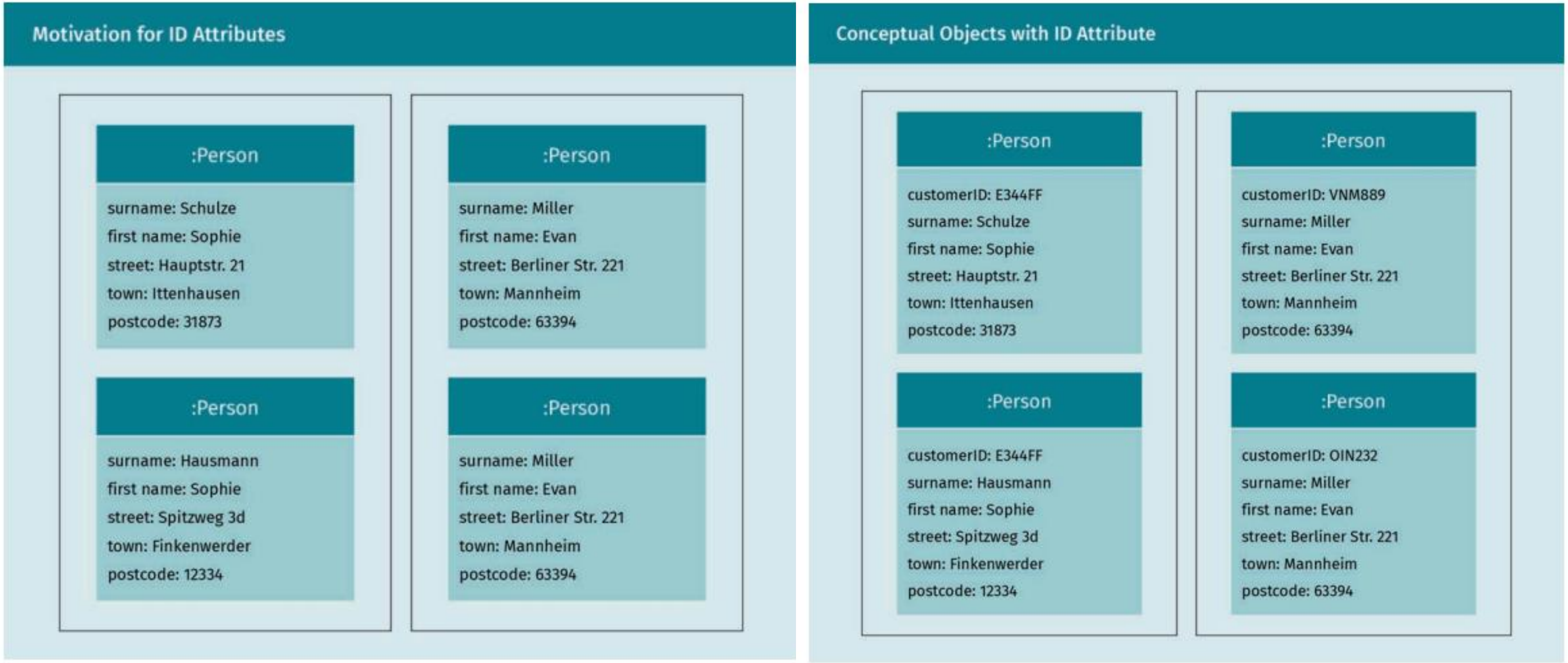
Requirements engineering uses the class diagram to document static concepts of an application area.

A UML class corresponds to a conceptual concept, i.e., a set of objects with the same properties, e.g., customer, article or order.

The class diagram is not suitable for the representation of behavior or processes.

ID ATTRIBUTE FOR UNIQUE IDENTIFICATION

Which box uniquely identifies an object of the class: Person?



Motivation for ID Attributes

:Person

surname: Schulze
first name: Sophie
street: Hauptstr. 21
town: Ittenhausen
postcode: 31873

:Person

surname: Hausmann
first name: Sophie
street: Spitzweg 3d
town: Finkenwerder
postcode: 12334

:Person

surname: Miller
first name: Evan
street: Berliner Str. 221
town: Mannheim
postcode: 63394

:Person

surname: Miller
first name: Evan
street: Berliner Str. 221
town: Mannheim
postcode: 63394

Conceptual Objects with ID Attribute

:Person

customerID: E344FF
surname: Schulze
first name: Sophie
street: Hauptstr. 21
town: Ittenhausen
postcode: 31873

:Person

customerID: E344FF
surname: Hausmann
first name: Sophie
street: Spitzweg 3d
town: Finkenwerder
postcode: 12334

:Person

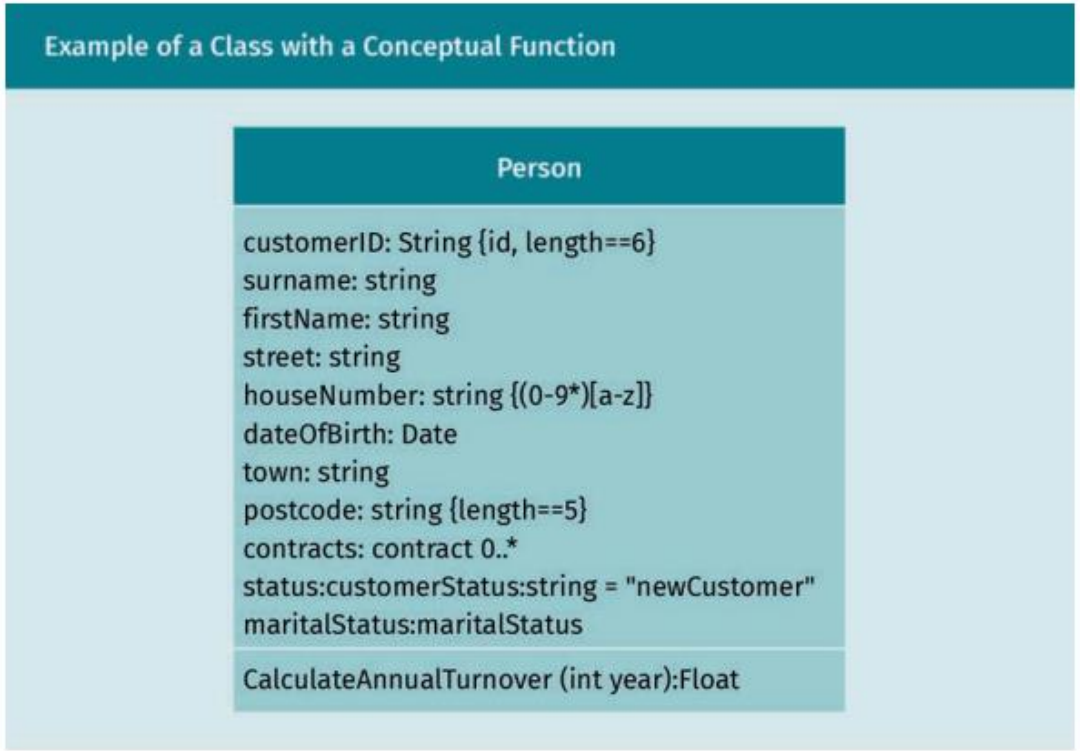
customerID: VNM889
surname: Miller
first name: Evan
street: Berliner Str. 221
town: Mannheim
postcode: 63394

:Person

customerID: OIN232
surname: Miller
first name: Evan
street: Berliner Str. 221
town: Mannheim
postcode: 63394

Image source: Coursebook.

COMPLETENESS OF ALL ATTRIBUTES



Example of a Class with a Conceptual Function

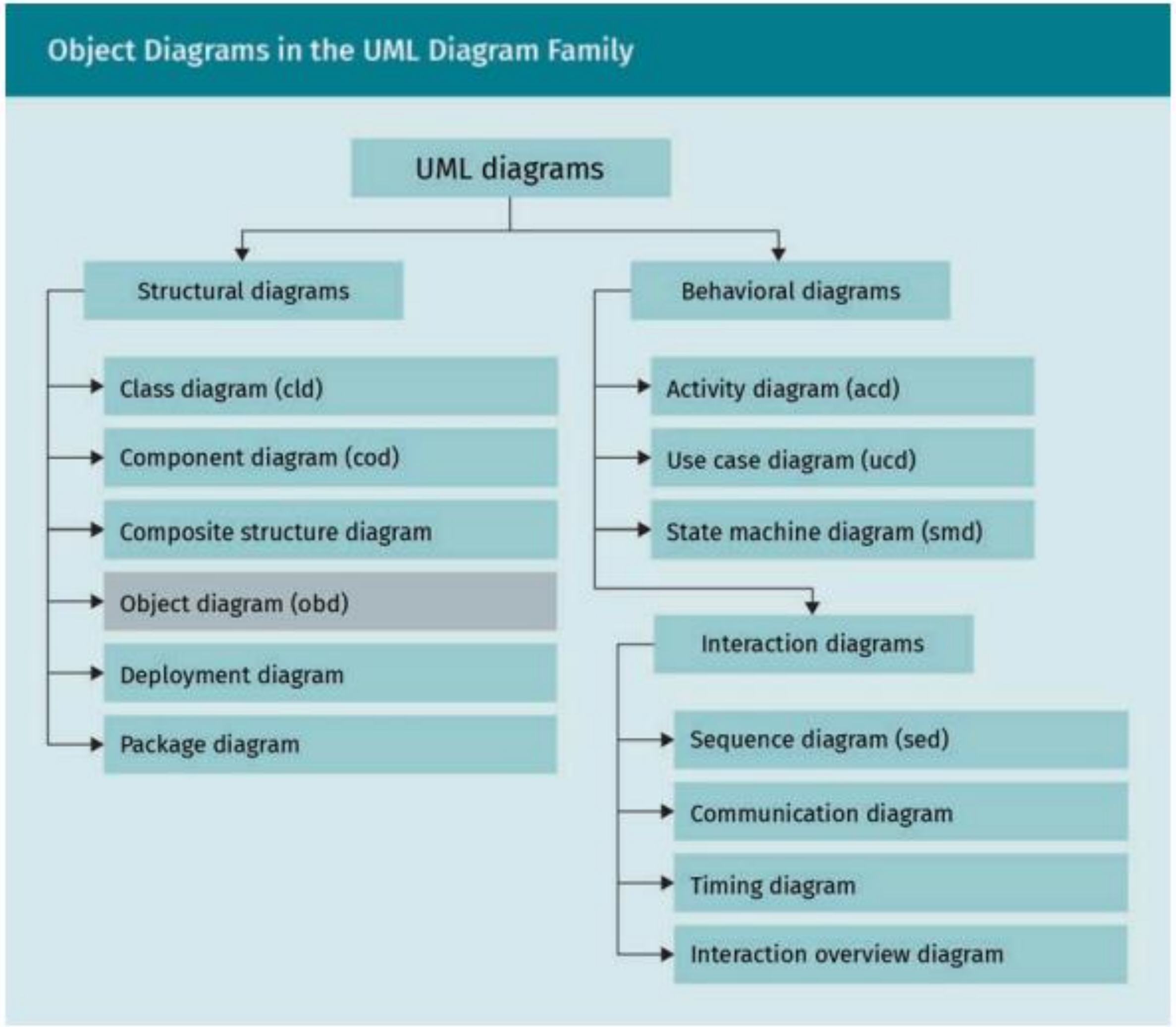
```
Person
customerID: String {id, length==6}
surname: string
firstName: string
street: string
houseNumber: string {(0-9*)[a-z]}
dateOfBirth: Date
town: string
postcode: string {length==5}
contracts: contract 0..*
status:customerStatus:string = "newCustomer"
maritalStatus:maritalStatus
CalculateAnnualTurnover (int year):Float
```

Image source: Coursebook.

The name of an attribute is often sufficient for **functional requirements engineering**, but the **technical specification requires further information**, e.g., the data type. This is because the data type influences the GUI, selection elements, validation rules, etc., for example. The set of attributes must be **complete**:

- **Data type**: See unit 1, e.g., a string. If there are already business or technical reasons for a special data type, it should be specified (e.g., integer for house numbers).
- **Multiplicities**: Specification of the number of values that can belong to an attribute.
- **Default value**: Initial value with which the attribute is automatically preassigned (if it makes sense for the subject matter).
- **Property values**: Optional properties such as read permissions, uniqueness, sorting, etc.
- Constraints: Rules for allowed attribute values.

CHECKING CLASS DIAGRAMS WITH UML OBJECT DIAGRAMS



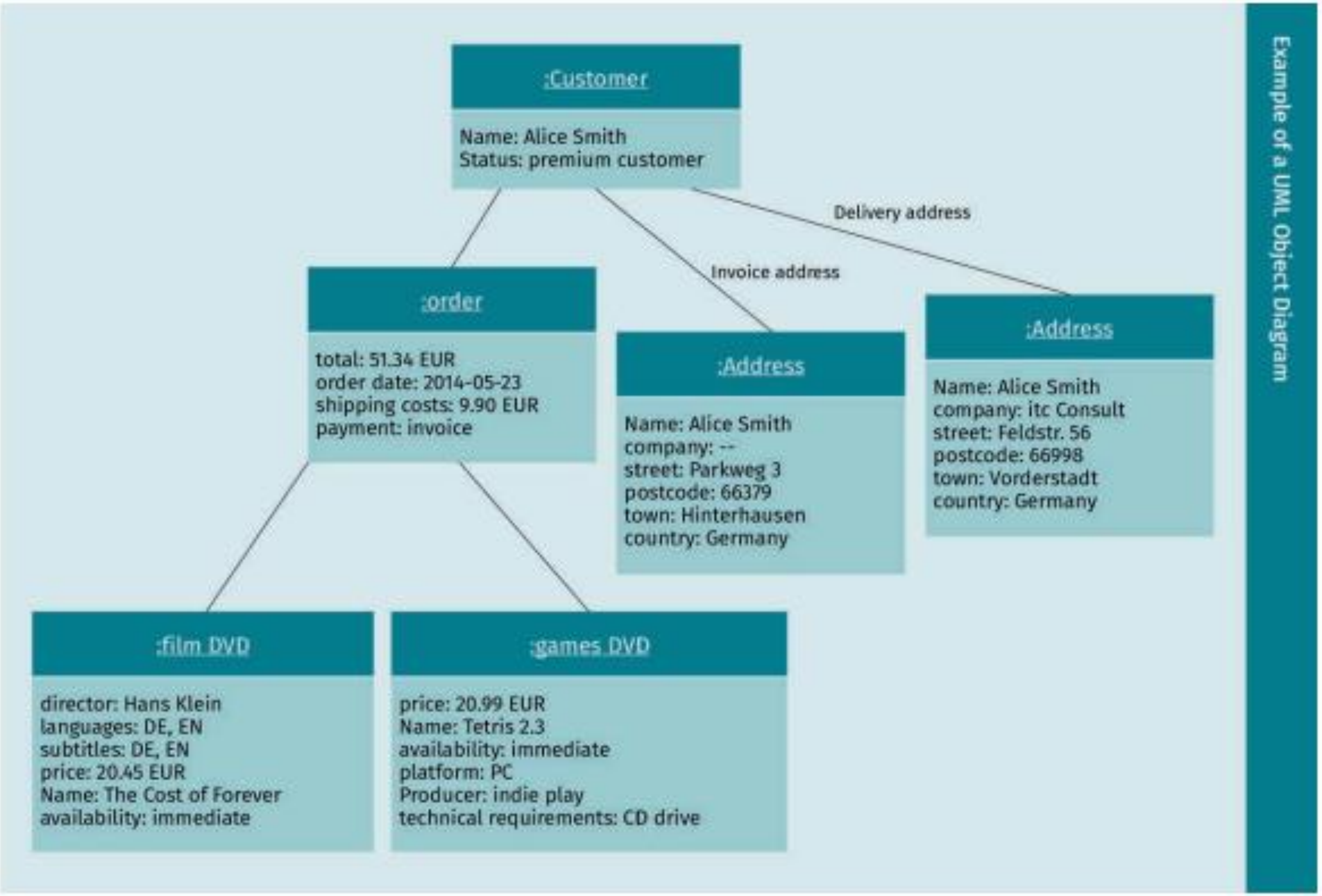
Object diagrams can be used to represent complex system states.

This allows the **technical correctness of class diagrams** and the **current state of data sets** to be checked and evaluated.

Due to the representation of data sets on the **instance level** (class diagram: Type level), no great capacity for abstraction is required for understanding.

Image source: Coursebook.

CHECKING CLASS DIAGRAMS WITH UML OBJECT DIAGRAMS



Example of a UML Object Diagram

:Customer

Name: Alice Smith
Status: premium customer

:order

total: 51.34 EUR
order date: 2014-05-23
shipping costs: 9.90 EUR
payment: invoice

:Address

Name: Alice Smith
company: --
street: Parkweg 3
postcode: 66379
town: Hinterhausen
country: Germany

:Address

Name: Alice Smith
company: itc Consult
street: Feldstr. 56
postcode: 66998
town: Vorderstadt
country: Germany

:film DVD

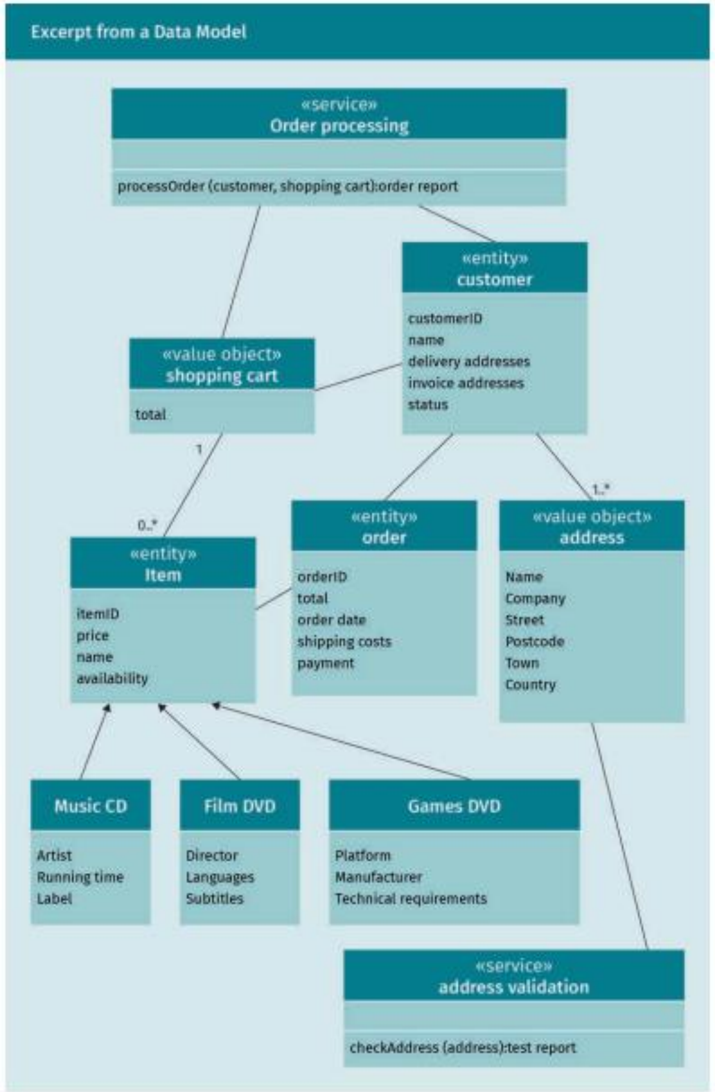
director: Hans Klein
languages: DE, EN
subtitles: DE, EN
price: 20.45 EUR
Name: The Cost of Forever
availability: immediate

:games DVD

price: 20.99 EUR
Name: Tetris 2.3
availability: immediate
platform: PC
Producer: indie play
technical requirements: CD drive

Image source: Coursebook.

TYPICAL ELEMENTS IN CONCEPTUAL DATA MODELS



- **Entities**: Elements of the data model that have a **conceptual framework**. Entities are often subject to a **life cycle**. Example: Insurance application (application > contract).

- **Value objects**: Elements of the data model that **do not** have a **conceptual framework**. Only the **data stored** in the objects is business-relevant. Value objects are used to store **additional information about entities**.

- **Services**: **Stateless business functions** that cannot be directly assigned to entities or value objects. The service itself has **no attributes and no internal state**. A service is described by its behavior.

Image source: Coursebook.

REVIEW STUDY GOALS

- Explain where conceptual data models can be used in the specification process.
- Outline the characteristics by which conceptual data models can be extended and refined.
- Describe how data models can be tested.

SESSION 5

TRANSFER TASK

TRANSFER TASKS

In the transfer task of unit 2, you found data types for the Deutsche Bahn booking system.

1. Represent the booking system as a UML object diagram.
2. Design the data model.

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

LEARNING CONTROL QUESTIONS

1. ID attributes in data models ...

   a) … are used to distinguish objects technically but not conceptually from one another.

   b) … are used to distinguish objects conceptually but not technically from one another.

   c) … are only needed to distinguish objects but are not used to identify them.

   d) … are used to distinguish objects both conceptually and technically from one another.

LEARNING CONTROL QUESTIONS

2. Checking data models for completeness …

a) … can be omitted if the project is already behind schedule.

b) … is a pre-requisite for specifying detailed conceptual operations with the UML use case diagram.

c) … is far more important for GUI specification than for the specification of technical interfaces.

d) … can be supported with the use of UML object diagrams.

3. The UML object diagram …

a) … is a type of diagram always used in preference over class diagrams.

b) … can be used to denote specific attribute values in classes.

c) … allows the selective representation of class instances but not the relationships between them.

d) … is a behavioral diagram whose structure is based on the UML class diagram.

LIST OF SOURCES