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 3

SPECIFICATION OF SYSTEM COMPONENTS

STUDY GOALS

- Describe the concept of a component and the relationship between components and systems.
- Explain how UML component diagrams can be used to specify structures.
- Know the internal behavior of components and how they can be specified.

EXPLAIN SIMPLY

1. Name which parts a complex software system consists of.
2. What can be considered the smallest component of a software system?
3. How can complex business rules be represented clearly?

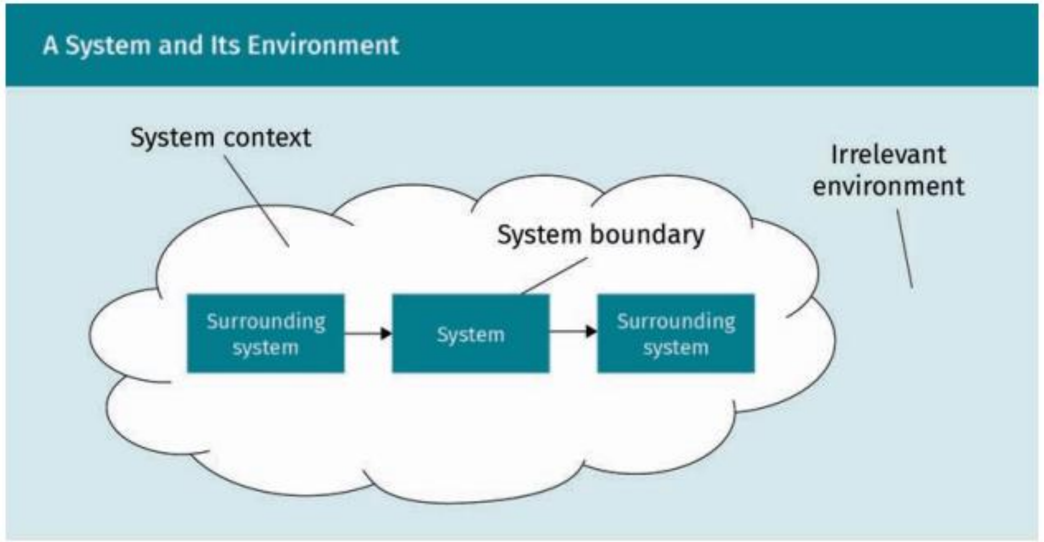
SYSTEM VS. COMPONENT

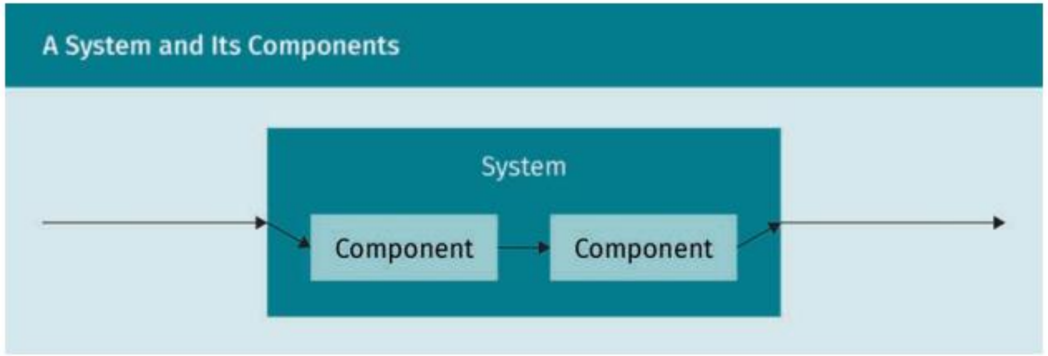
A system represents an **interaction context** that is distinct from its environment.

A specification does not always describe the SW system to be created holistically, but rather specific parts. Complex SW systems are divided into **components**.

In object-oriented systems **classes** are the smallest possible form of components of a system.

A component can consist again of **two or more compound components**.
As with classes, components make **interfaces** available.


A System and Its Environment


A System and Its Components

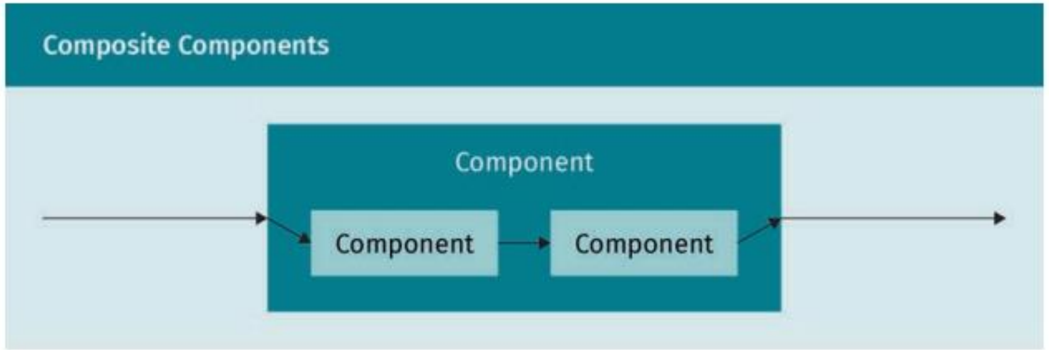

Composite Components

Image source: Coursebook.

SYSTEM VS. COMPONENT

A **system** consists of at least one **component**.

A component can contain different **subcomponents**.

The smallest possible structural elements are **classes**.

By using components in particular these quality goals can be realized:

- Simple expandability
- Better testability
- Better maintainability

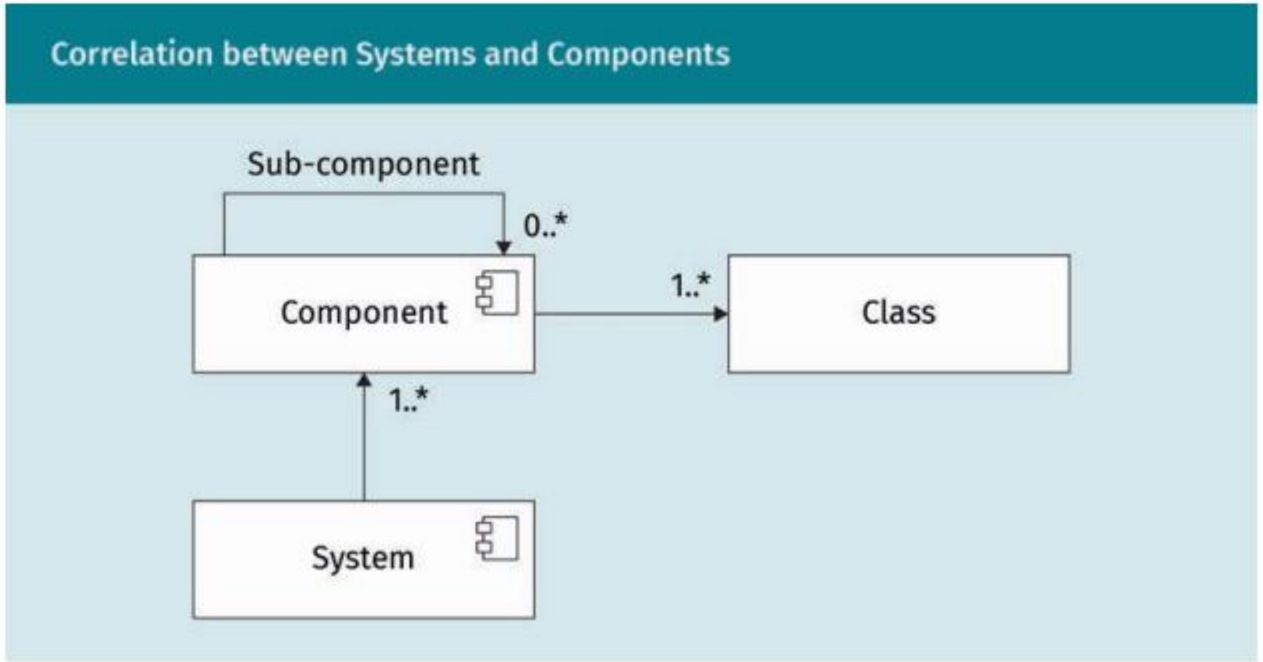
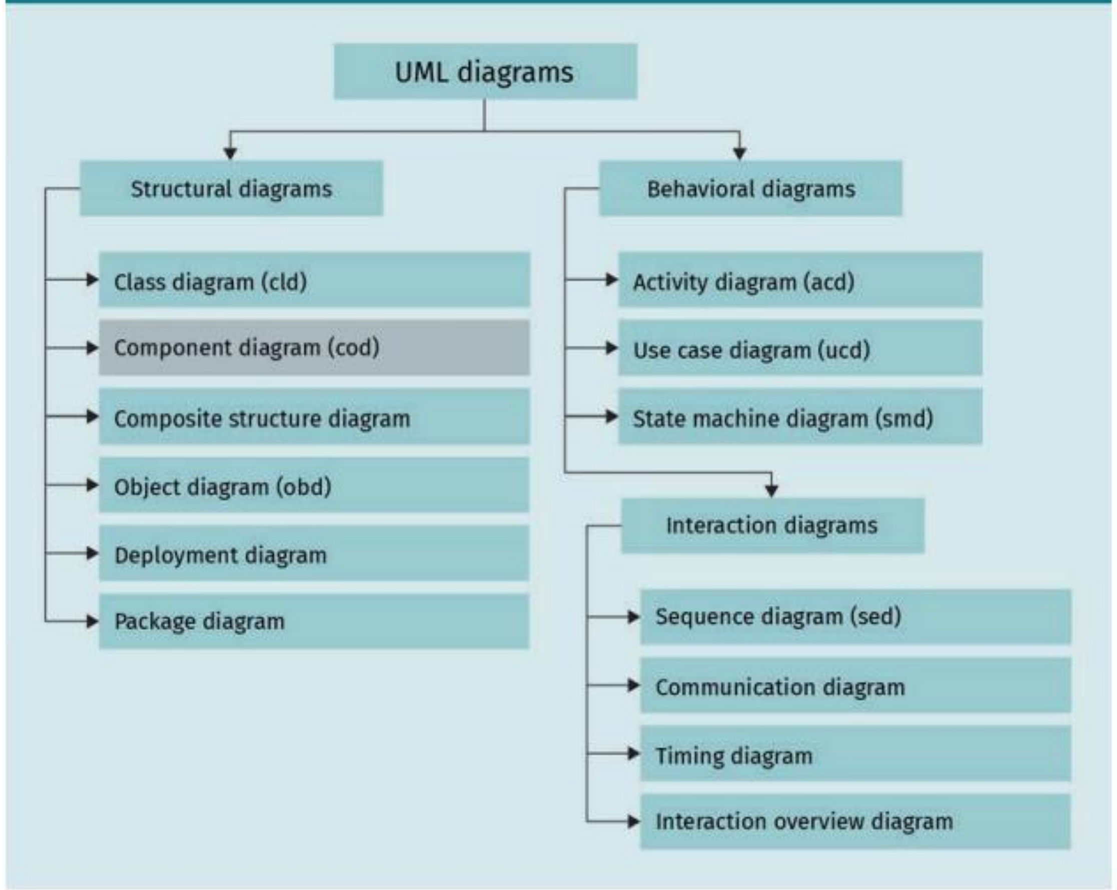


Image source: Coursebook.

SPECIFY THE STRUCTURE OF SYSTEMS AND COMPONENTS

Object Diagrams in the UML Diagram Family



Example of a Component Diagram

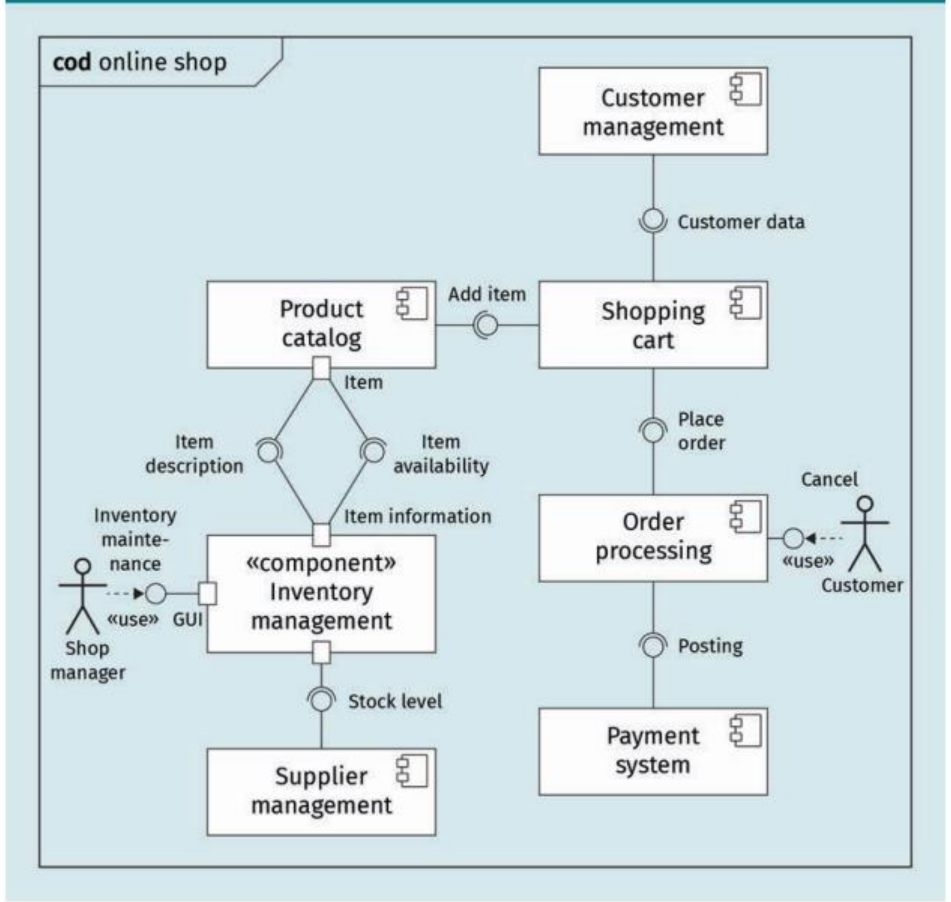
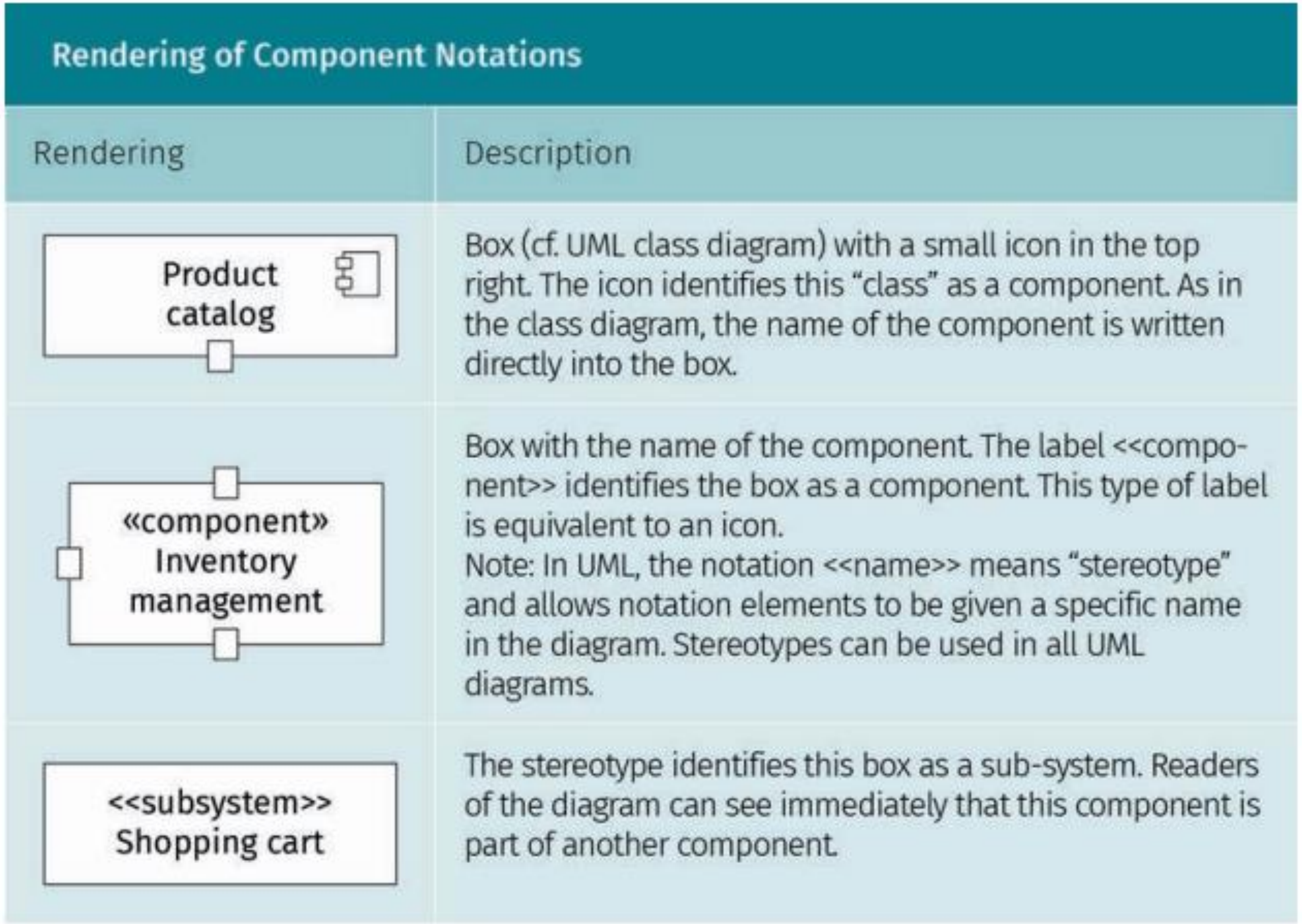


Image source: Coursebook.

SPECIFY THE STRUCTURE OF SYSTEMS AND COMPONENTS

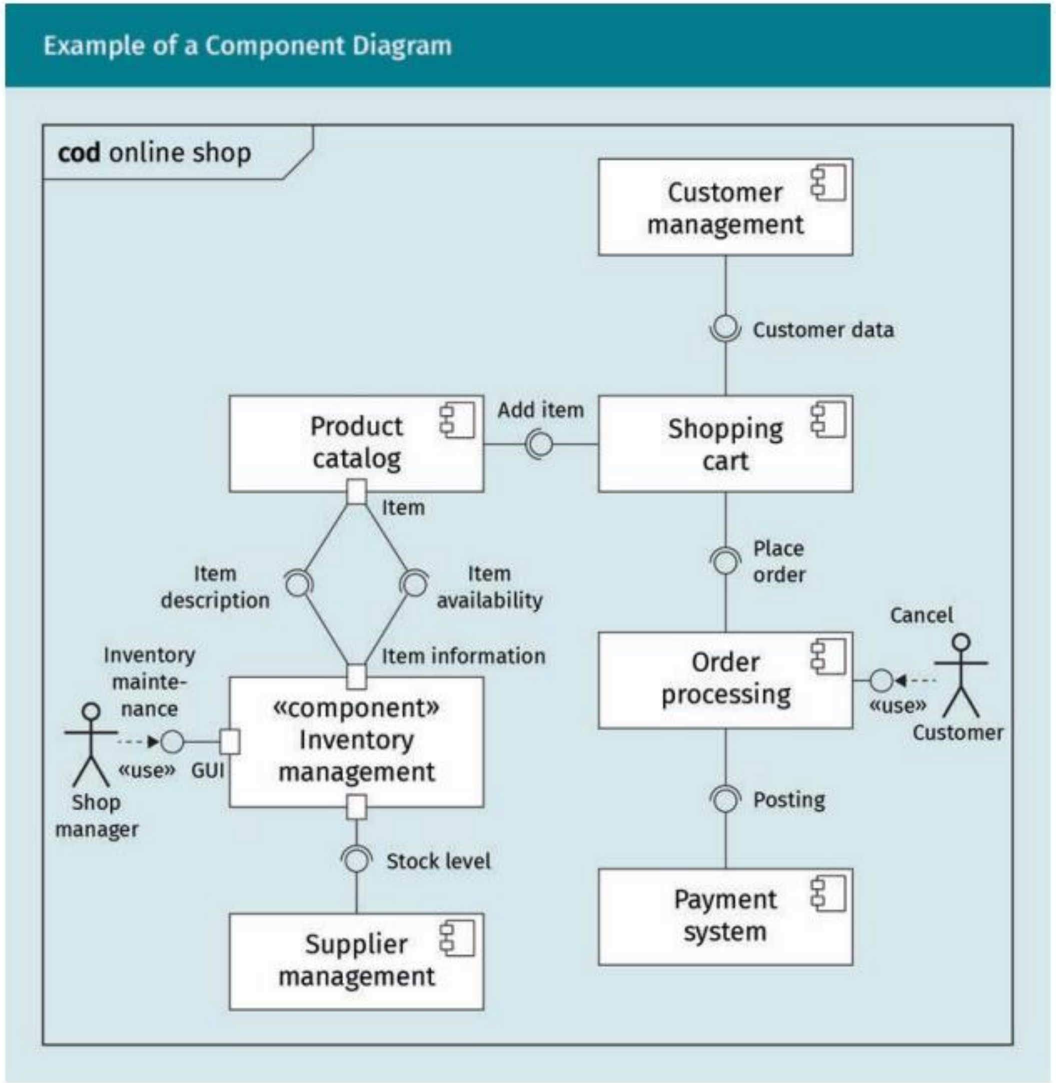


Rendering of Component Notations

Rendering	Description
Product catalog	Box (cf. UML class diagram) with a small icon in the top right. The icon identifies this “class” as a component. As in the class diagram, the name of the component is written directly into the box.
«component» Inventory management	Box with the name of the component. The label <<component>> identifies the box as a component. This type of label is equivalent to an icon. Note: In UML, the notation <<name>> means “stereotype” and allows notation elements to be given a specific name in the diagram. Stereotypes can be used in all UML diagrams.
<<subsystem>> Shopping cart	The stereotype identifies this box as a sub-system. Readers of the diagram can see immediately that this component is part of another component.

Image source: Coursebook.

SPECIFYING THE BEHAVIOR OF COMPONENTS


Example of a Component Diagram

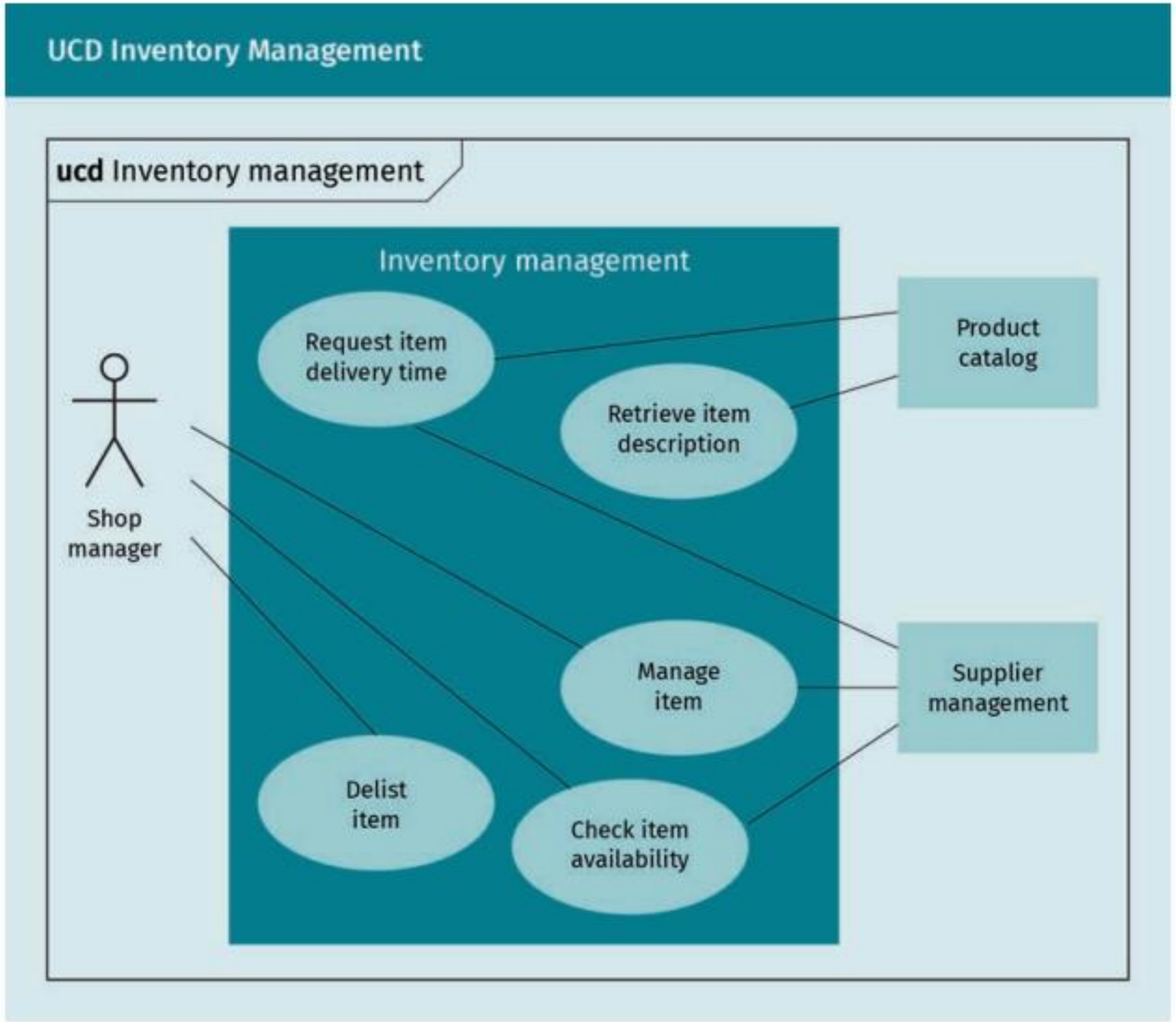
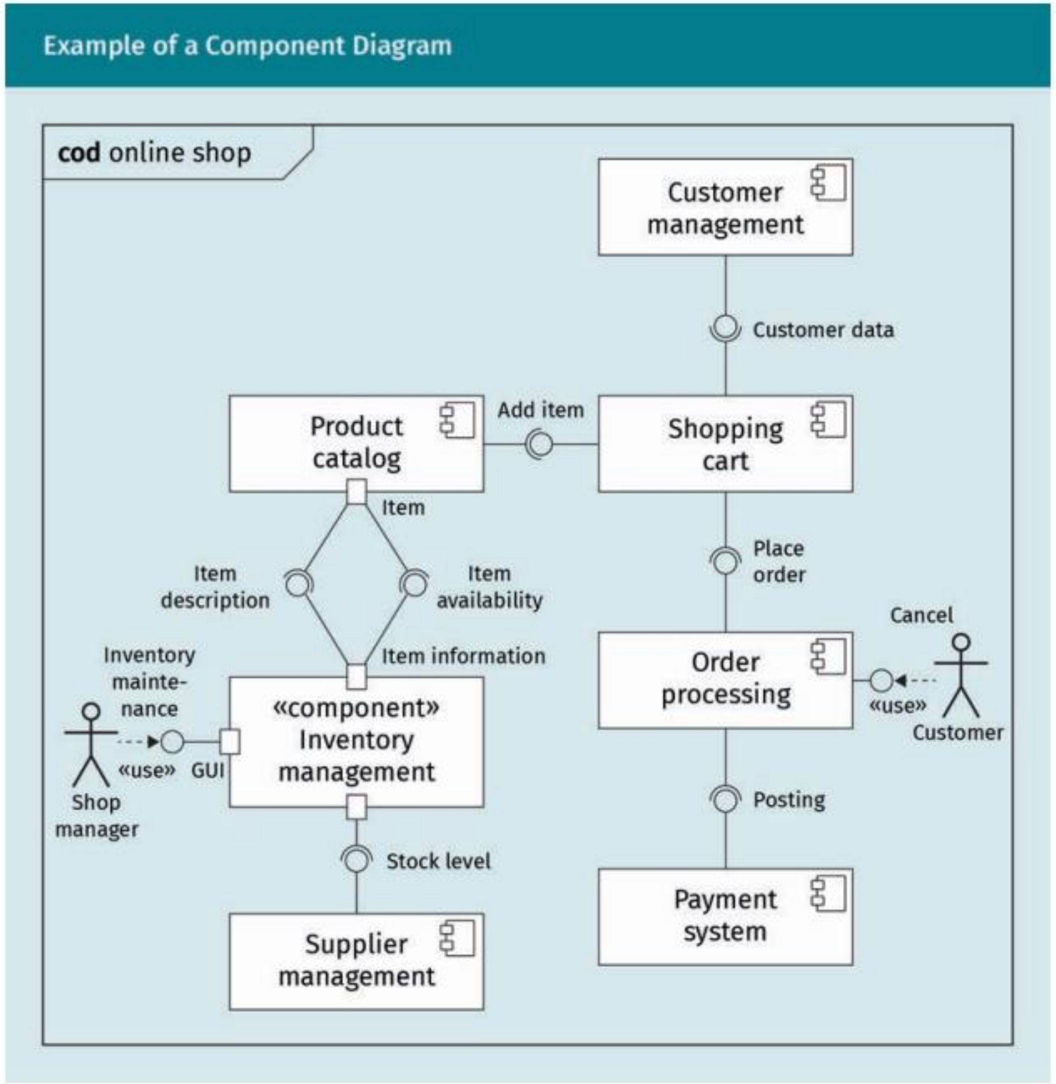

UCD Inventory Management

Image source: Coursebook.

SPECIFYING THE BEHAVIOR OF COMPONENTS



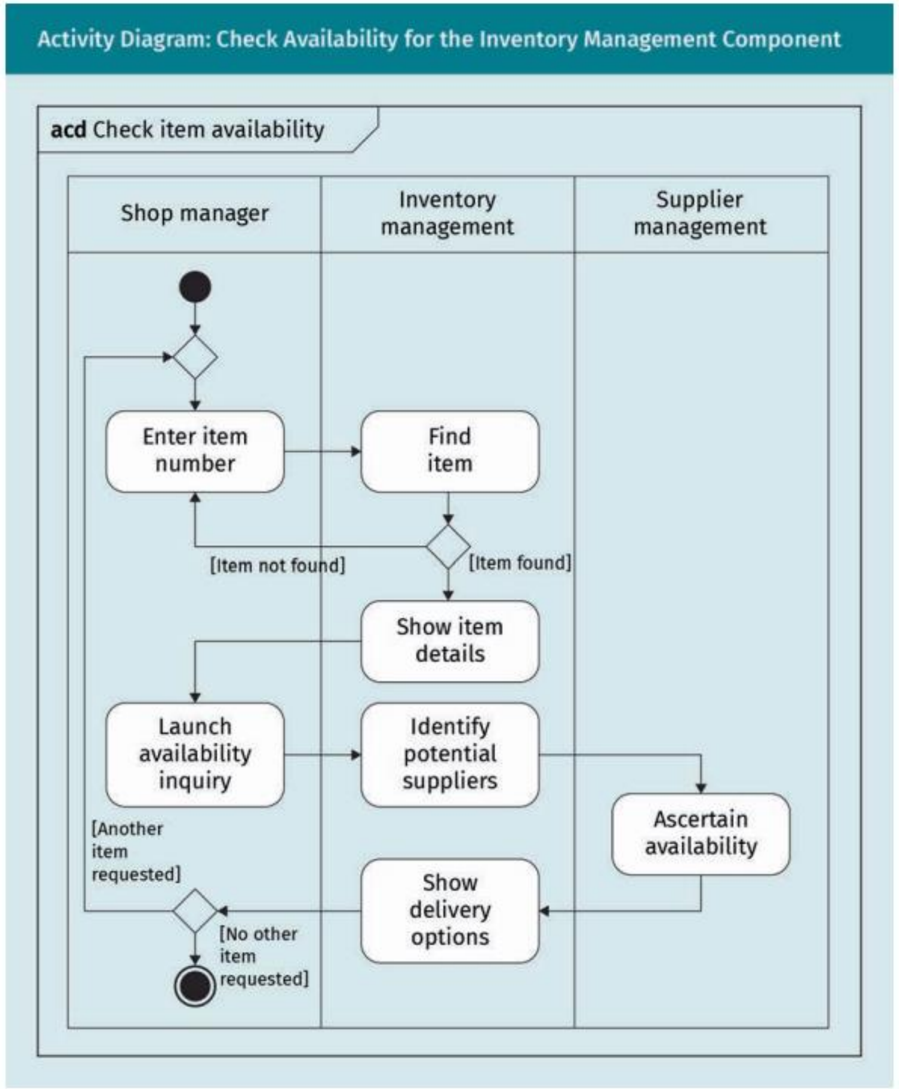


Image source: Coursebook.

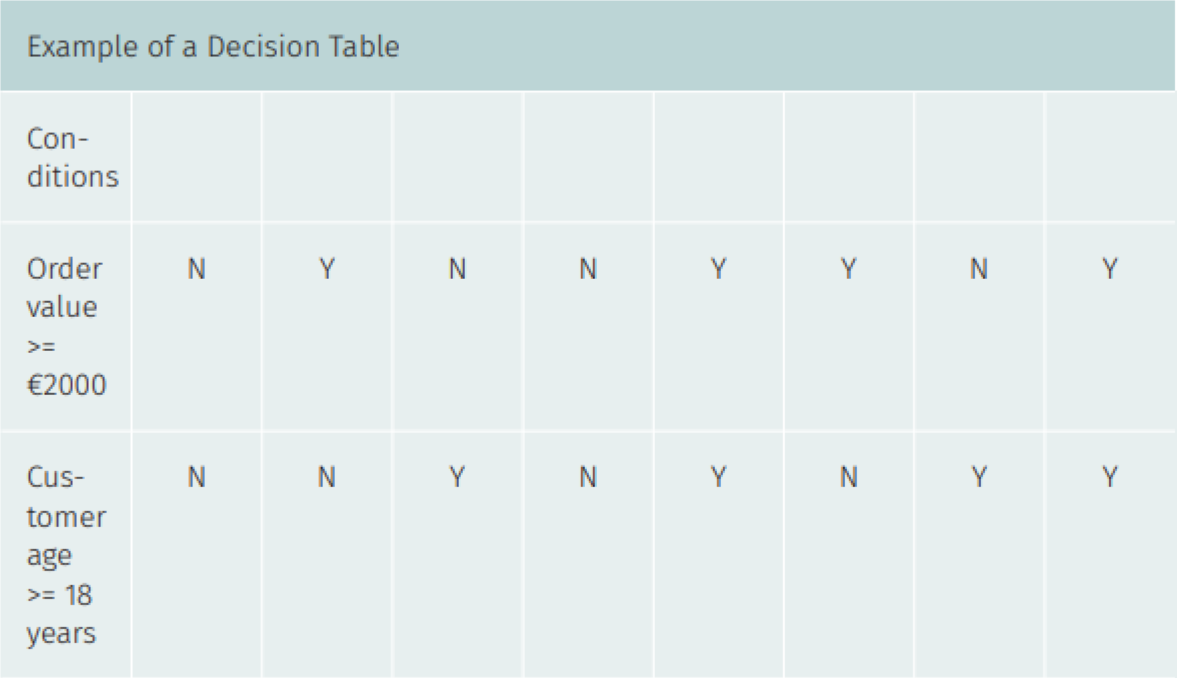
Decision tables offer the possibility to document **complex business** rules clearly.

Four different areas can be distinguished in decision tables:

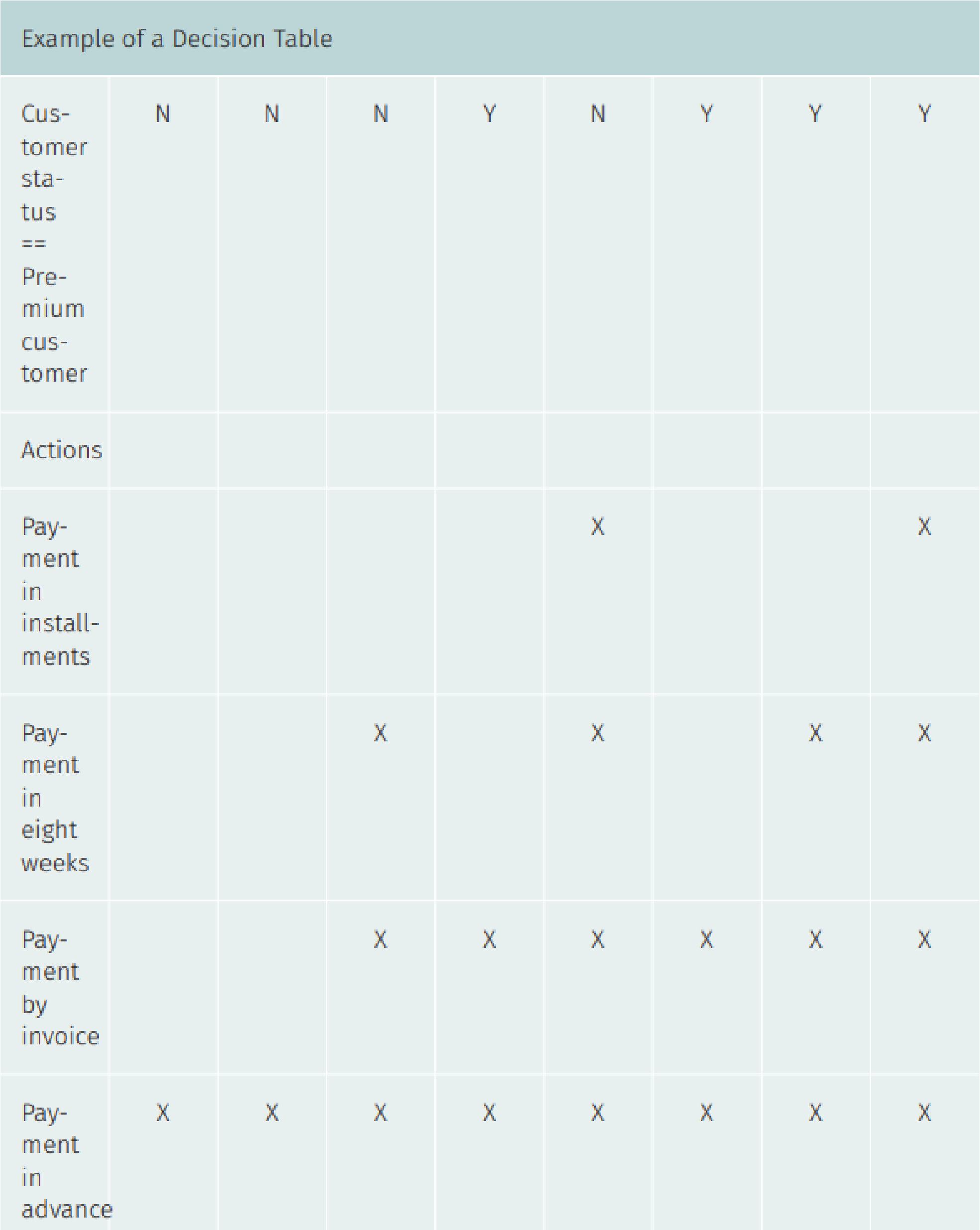
- Definition of conditions.
- Definition of the actions to be executed.
- Combination of the results of the evaluation of the conditions.
- Assignment of the evaluation results to the actions to be executed.

See next slide for example.

MODELING BUSINESS RULES WITH DECISION TABLES



Example of a Decision Table								
Conditions								
Order value >= €2000	N	Y	N	N	Y	Y	N	Y
Customer age >= 18 years	N	N	Y	N	Y	N	Y	Y



Example of a Decision Table								
Customer status == Premium customer	N	N	N	Y	N	Y	Y	Y
Actions								
Payment in installments					X			X
Payment in eight weeks			X		X		X	X
Payment by invoice			X	X	X	X	X	X
Payment in advance	X	X	X	X	X	X	X	X

Image source: Coursebook.

REVIEW STUDY GOALS

- Describe the concept of a component and the relationship between components and systems.
- Explain how UML component diagrams can be used to specify structures.
- Know the internal behavior of components and how they can be specified.

SESSION 3

TRANSFER TASK

TRANSFER TASKS

In the Unit 2 transfer task, you depicted the Deutsche Bahn booking system as a UML state diagram.

1. Build on this and represent the booking system as a component diagram.
2. Think about a meaningful activity diagram and present it.

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

LEARNING CONTROL QUESTIONS

1. Software components …

   a) … are independent software units which interact with one another via precisely one interface.

   b) … are viewed in the specification as a white box. Only their internally visible behavior must be specified.

   c) … are the elements which make up complex software systems and are therefore included in the specification.

   d) … cannot contain any further software components themselves.

LEARNING CONTROL QUESTIONS

2. The UML component diagram ...

a) … has interfaces and human actors as its principal notation elements.

b) … is suitable for denoting operations between different components.

c) … is a UML structural diagram suitable for modeling the internal structure of systems.

d) … is unsuitable for modeling conceptual dependencies.

LEARNING CONTROL QUESTIONS

3. Decision tables for specifying component behavior ...

a) … comprise four zones: one for the conditions, one for their evaluations, one for the actions, and one for the allocation of evaluations to actions.

b) … enable the clear representation of complex information structures.

c) … can verify that a set of rules is free from contradictions, but do not check for completeness.

d) … may alternatively be modeled by UML state diagrams.

LIST OF SOURCES