

# **System Development Methods**

## **CT00046-3-2**

### **System Design – Part 2**

# Topic & Structure of the Lesson

- ❑ Unified Modeling Language (UML) and examples of UML diagrams.
- ❑ Concept of user interface design and Human-Computer Interaction (HCI), including basic principles of user-centered design.

# Learning Outcomes

At the end of the module, you should be able to:

1. Describe the Unified Modeling Language (UML) and examples of UML diagrams.
2. Explain the concept of user interface design and human-computer interaction, including basic principles of user-centered design.

# Key Terms you must be able to use

If you have mastered this topic, you should be able to use the following terms correctly in your assignment and exam:

- ☐ UML and UML diagrams.
- ☐ Concept of user interface design, human-computer interaction, and basic principles of user-centered design.

# Influence of Software Design

## ❑ Process Centered Design

- Designing software based on automation of the process
- Examples: Ticketing Machine, Manufacturing process, Hotel booking, etc.

## ❑ Data Centered Design

- Designing software based on processing large data.
- Examples: Big Data, Shopping, Immigration system, flight management.

## ❑ User Centered Design

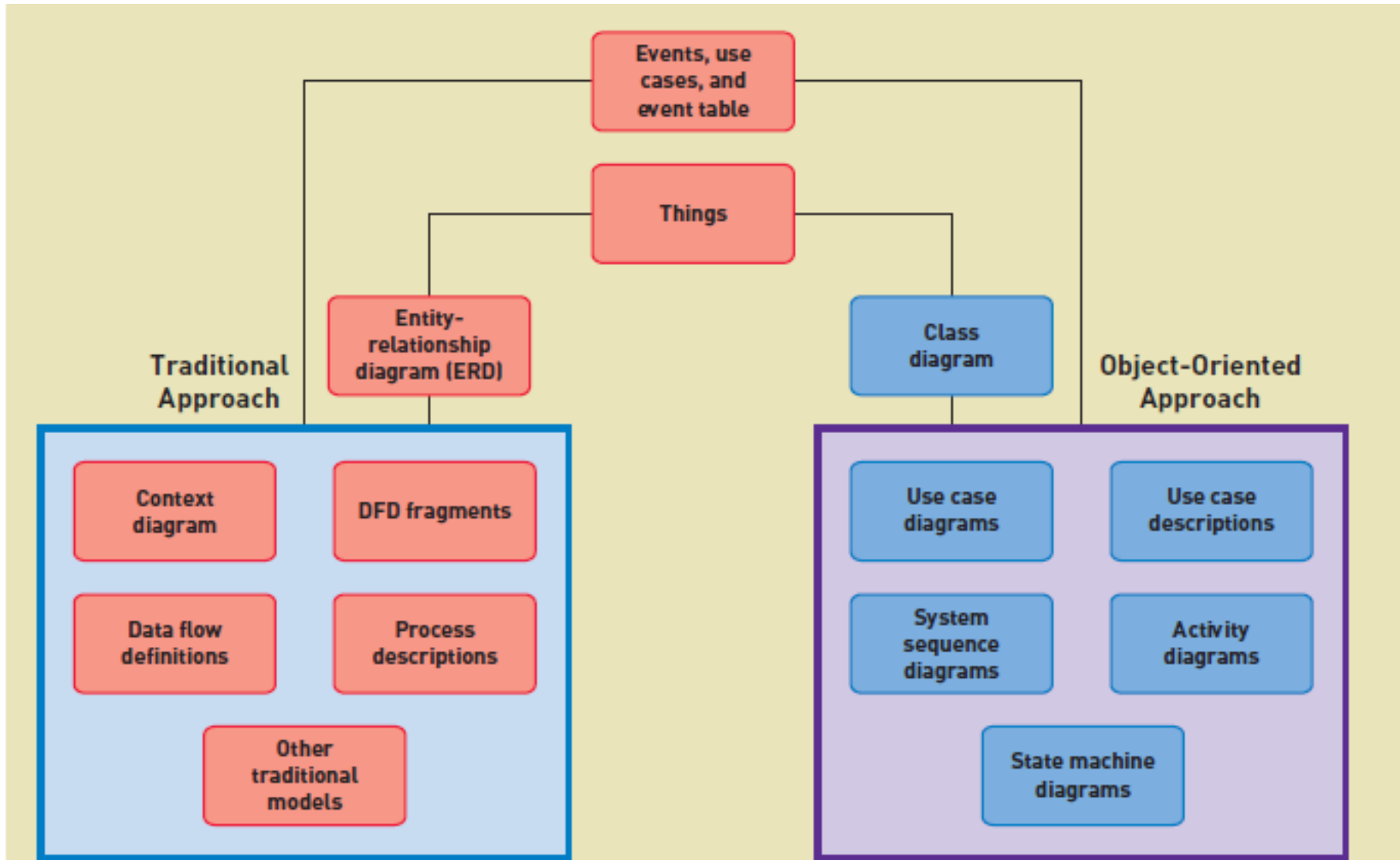
- Designing software based on human behavior
- Examples: AI, games, social sites, navigation applications, etc.

# Unified Modeling Language (UML)

- ❑ **Unified Modeling Language (UML)** is a widely used method of visualizing and documenting software systems design.
- ❑ UML uses object-oriented design concepts, but it is independent of any specific programming language and can be used to describe business processes and requirements generally.
- ❑ UML can be used to design, discuss, present and even test a software component before its built.

# Unified Modeling Language (UML)

## Traditional versus Object-Oriented Approaches



# Unified Modeling Language (UML)

## TWO different views of UML diagrams

### ❑ Static / Structural view

- Emphasizes the static structure of the system using objects, attributes, operations and relationships.
- Used for data and process-oriented development approach.
- Presented through - class and entity relationship diagrams.

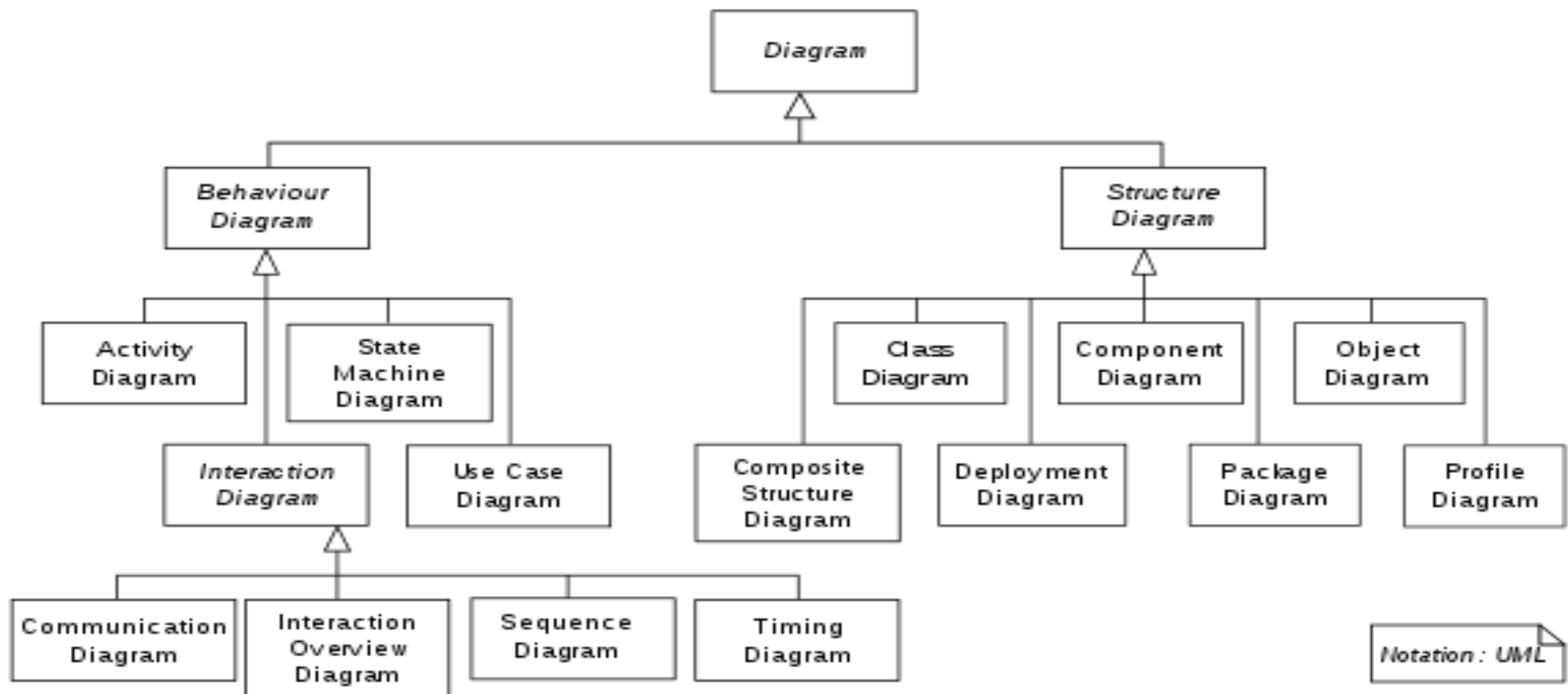
### ❑ Dynamic / Behavioral view

- Emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects.
- Used for a user and some process-oriented development approach.
- Presented through sequence diagrams, activity diagrams and state machine diagrams.



# Unified Modeling Language (UML)

## TWO different views of UML modeling



# Examples of UML Diagrams

## Behaviour Diagrams

- ❑ An Activity diagram represents workflows in a graphical way.
  - Describes the business workflow or the operational workflow of any component in a system.
- ❑ A State machine diagram is similar to an activity diagram.
  - Known as a state diagram or state chart diagram.
  - Describes the behavior of objects that act differently according to the state they are in now.
- ❑ A Use case diagram gives a graphic overview of the actors involved in a system,
  - Shows different functions needed by those actors and how these different functions interact.
- ❑ A Sequence diagram shows how objects interact with each other and the order in which those interactions occur.

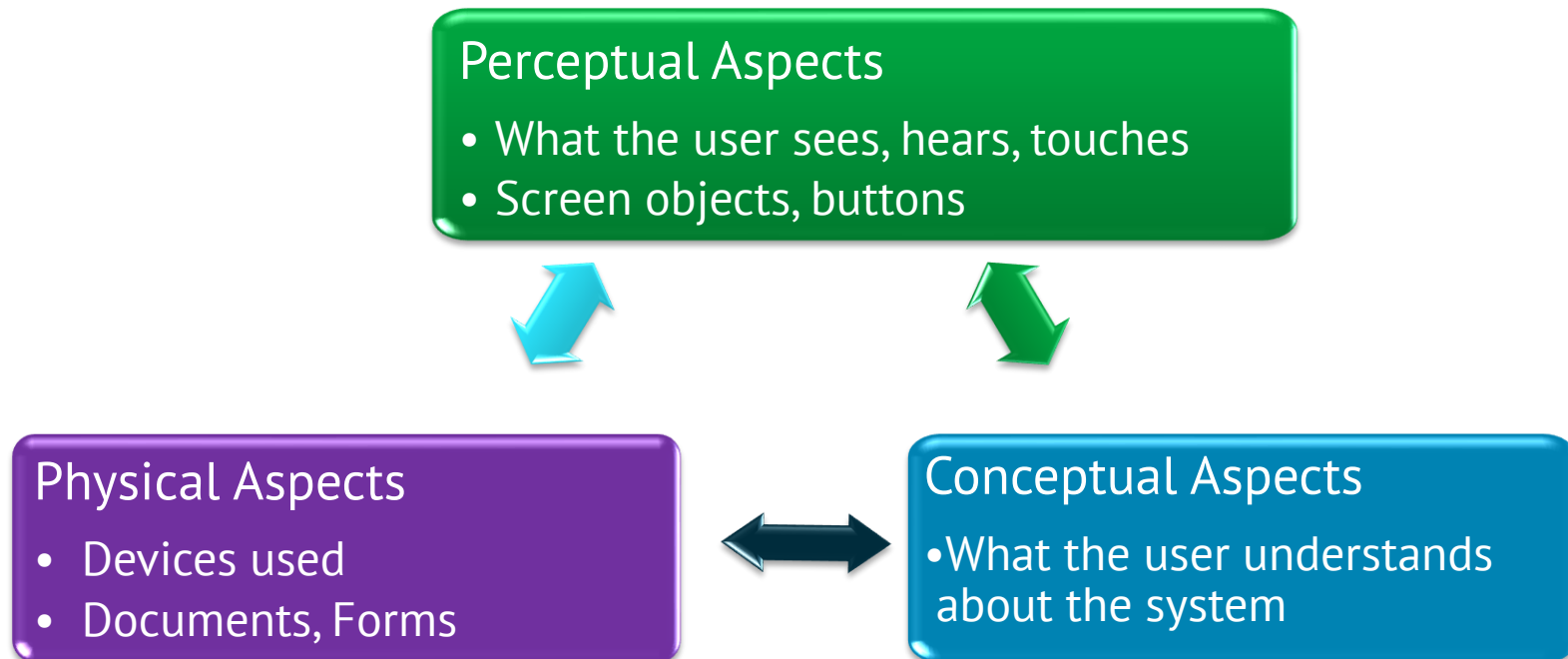
# Examples of UML Diagrams

## Structure Diagrams

- ❑ A class diagram is the main building block of any object-oriented solution.
  - It shows the classes in a system, attributes, and operations of each class and the relationship between each class.
- ❑ A component diagram displays the structural relationship of components of a software system.
  - Mostly used when working with complex systems with many components
- ❑ A deployment diagram shows the hardware of your system and the software in that hardware.
  - Useful when your software solution is deployed across multiple machines with each having a unique configuration.
- ❑ Object Diagram referred to as Instance diagram - very similar to class diagrams.
  - Show the relationship between objects, but they use real-world examples.

# User Interface Design

- ❑ A **user interface (UI)** describes how users interact with a computer system, and consists of all the hardware, software, screens, menus, functions, output, and features that affect two-way communications between the user and the computer.
- ❑ Three aspects of the user interface:



# User-Centered Design Principles

## Understand the Business

- Understand the underlying business functions and how the system supports individual, departmental, and enterprise goals.
- The overall objective is to design an interface that helps users to perform their jobs.

## Maximize Graphical Effectiveness

- A well-designed interface can help users learn a new system rapidly and be more productive.

## Think Like a User

- The interface should be flexible enough to accommodate novices as well as experienced users.

## Use Models and Prototypes

- Construct models and prototypes for user approval.
- An interface designer should obtain as much feedback as possible, as early as possible.

# User-Centered Design Principles (continued)

## Focus on Usability

- Include all tasks, commands, and communications between users and the information system.

## Invite Feedback

- Even after the system is operational, it is important to monitor system usage and solicit user suggestions.
- You can determine if system features are being used as intended by observing and surveying users.

## Document Everything

- Document all screen designs for later use by programmers.
- If you are using a CASE tool or screen generator, number the screen designs and save them in a hierarchy like a menu tree.
- User-approved sketches and storyboards also can be used to document the user interface.

# Human Computer Interaction (HCI)

- ❑ A user interface is based on basic principles of human-computer interaction. **Human-computer interaction (HCI)** describes the relationship between computers and people who use them to perform their jobs.
- ❑ HCI concepts apply to everything from PC desktops to global networks. In its broadest sense, a user interface includes all the communications and instructions necessary to enter input to the system and to obtain output in the form of screen displays or printed reports.
- ❑ Concepts of creating software that is safe and pleasant to use.

# Human Computer Interaction (HCI) (continued)

- ❑ As a systems analyst, you will design user interfaces for in-house developed software and customize interfaces for various commercial packages and user productivity applications. Your main objective is to create a user-friendly design that is easy to learn and use.
- ❑ Things to consider in HCI:
  - Users' abilities/ disabilities
  - User's age group (interest, legal contents, etc.)
  - User's computer literacy (Novice, Intermediate, Expert)
  - User's Socio-culture (perception of color, language, religious sensitivity, etc.)
  - Purpose of Software (Command line, GUI, Interactive, 3D, etc.)



# Summary

- ❑ Unified Modeling Language (UML) is a method of visualizing and documenting software systems design. UML uses object-oriented design concepts, but it is independent of any specific programming language.
- ❑ UML static diagrams emphasize the static structure of the system using objects, attributes, operations, and relationships. Used for data and process-oriented development approach.
- ❑ UML behavioral diagrams emphasize the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. Used for a user and some process-oriented development approach.
- ❑ System user interface is based on basic principles of human-computer interaction that describe the relationship between computers and people who use them to perform their jobs.
- ❑ The user-centered design aim is to design an interface that helps users to perform their jobs and based on perceptual, physical, and conceptual aspects.

# Question & Answer

# Next session

☐ System Implementation

# References

- ❑ Tilley, S. (2019). Systems Analysis and Design 12th Edition. Cengage Learning. ISBN-13: 978-0357117811. ISBN-10: 0357117816
- ❑ Pressman, R., & Maxim, B. (2019). Software Engineering: A Practitioner's Approach 9th Edition. McGraw Hill. ISBN-13: 978-1259872976. ISBN-10: 1259872971
- ❑ Dennis, A., Wixom, B., & Roth, R.M. (2021). Systems Analysis and Design 8th Edition. Wiley. ISBN: 978-1-119-80378-2