



DESIGN CONCEPTS

Lecture # 19

Lecture by Engr. Sidra



OBJECTIVES

- When you have read the chapter you will:
 - Understand software design process
 - Know about the quality goals and guidelines for good software designs
 - Know about the principles of design
 - Understand design concepts



DESIGN

- Once the requirements document for the software to be developed is available, the software design phase begins.
- While the requirement specification activity deals entirely with the problem domain, design is the first phase of transforming the problem into a solution.
- In the design phase, the customer and business requirements and technical considerations all come together to formulate a product or a system.



DESIGN

- Software design is a phase in software engineering, in which a blueprint is developed to serve as a base for constructing the software system.
- **IEEE** defines software design as
 - 'both a process of defining, the architecture, components, interfaces, and other characteristics of a system or component and the result of that process.'
- In the design phase, many critical and strategic decisions are made to achieve the desired functionality and quality of the system.
- These decisions are taken into account to successfully develop the software and carry out its maintenance in a way that the quality of the end product is improved.



DESIGN

- Mitch Kapor, the creator of Lotus 1-2-3, presented a “software design manifesto” in Dr. Dobbs Journal. He said:
- Good software design should exhibit:
 - **Firmness:** A program should not have any bugs that inhibit its function.
 - **Commodity:** A program should be suitable for the purposes for which it was intended.
 - **Delight:** The experience of using the program should be pleasurable one.

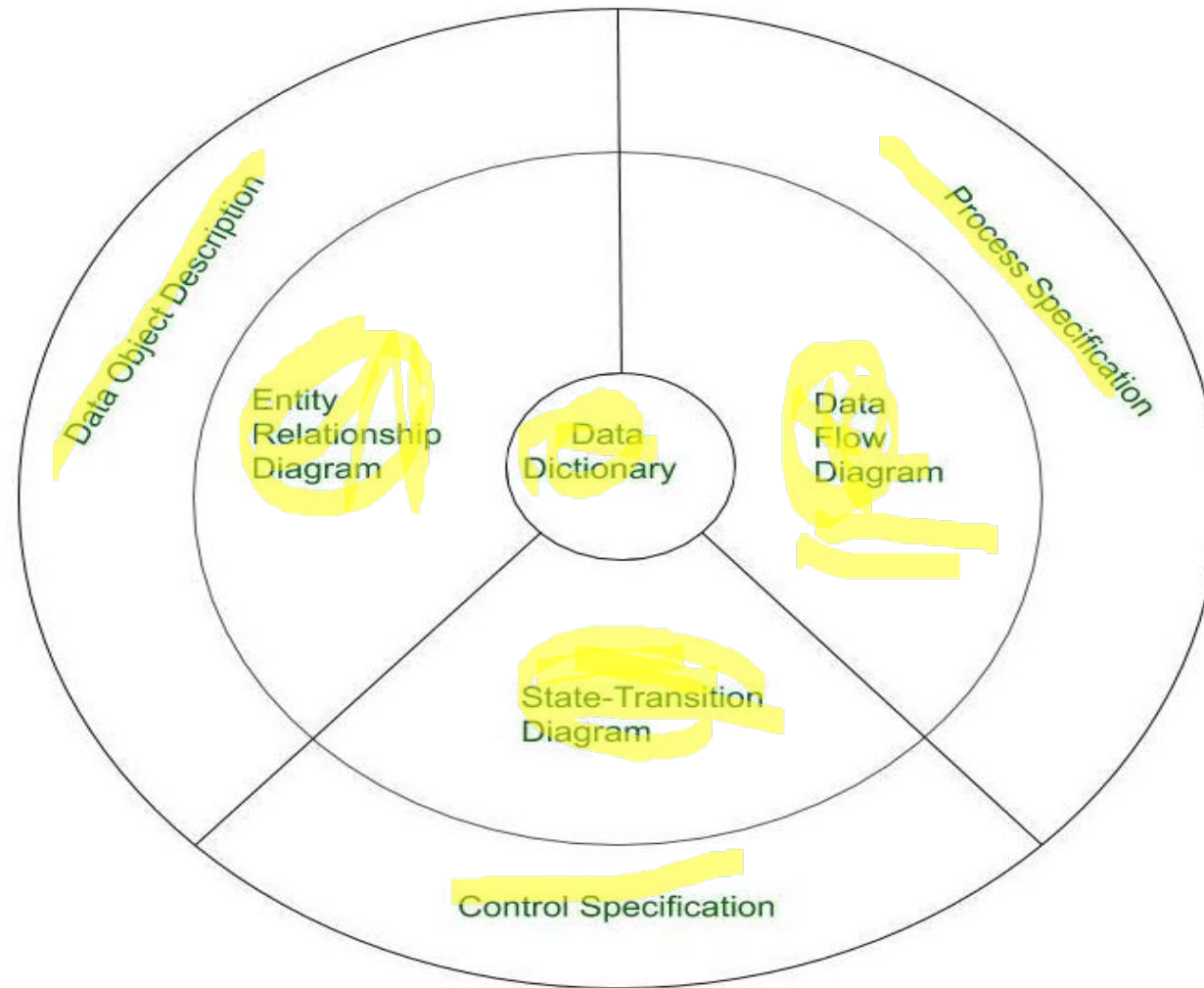


ANALYSIS MODELLING

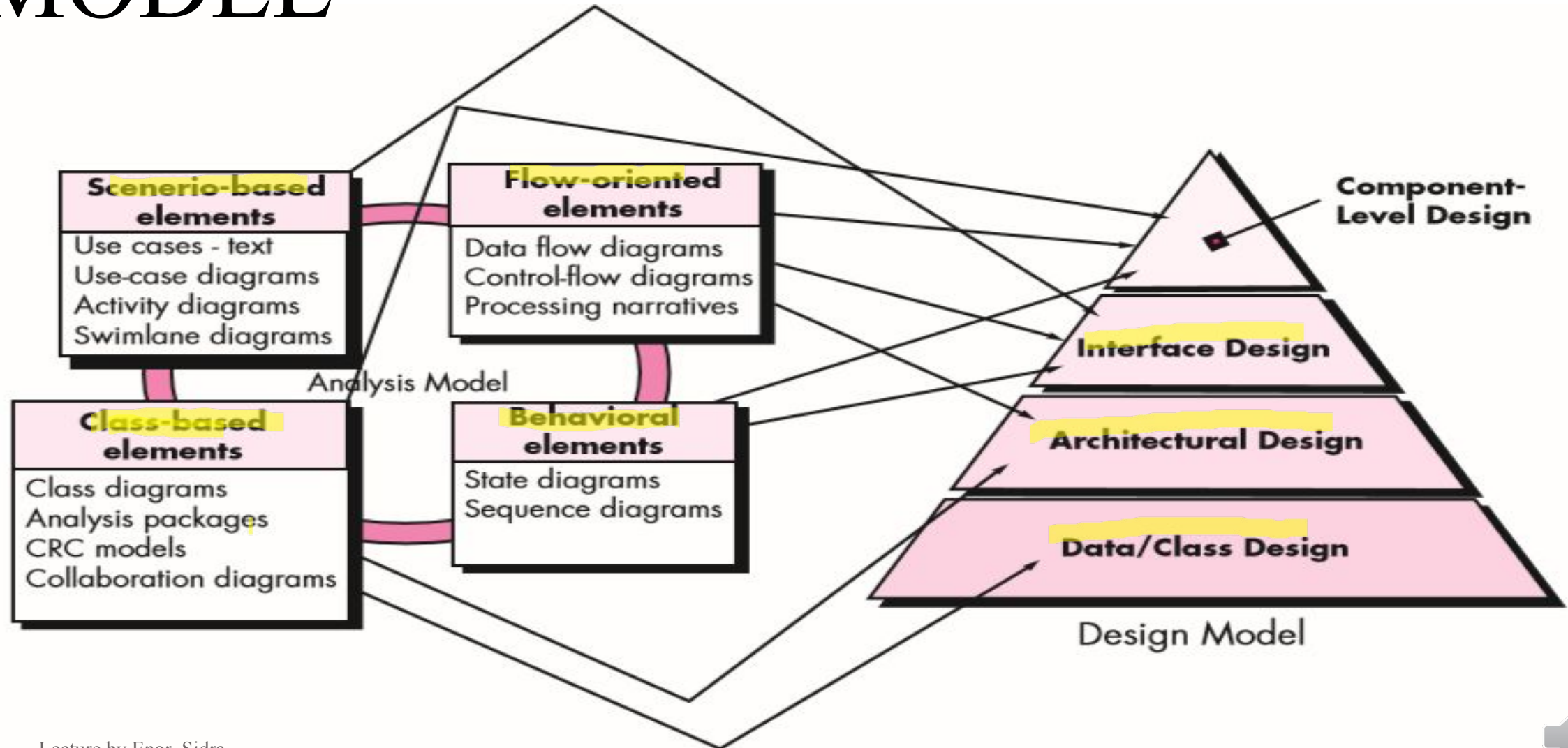
- **Analysis Model** is a technical representation of the system. It acts as a link between system description and design model.
- In Analysis Modelling, information, behavior and functions of the system is defined and translated into the architecture, component and interface level design in the design modeling.
- **Objectives of Analysis Modelling:**
 - It must establish a way of creation of software design.
 - It must describe requirements of customer.
 - It must define set of requirements which can be validated, once the software is built.



ELEMENTS OF ANALYSIS MODEL



ANALYSIS MODEL □ DESIGN MODEL



DESIGN PROCESS

- The design process comprises a set of principles, concepts and practices, which allow a software engineer to model the system or product that is to be built.
- This model, known as design model, is assessed for quality and reviewed before a code is generated and tests are conducted.
- The design model provides details about software data structures, architecture, interfaces and components which are required to implement the system.



DESIGN AND QUALITY GOALS

- **The design must implement all of the explicit requirements** contained in the analysis model, and it must accommodate all of the implicit requirements desired by the customer.
- **The design must be a readable, understandable guide** for those who generate code and for those who test and subsequently support the software.
- **The design should provide a complete picture of the software**, addressing the data, functional, and behavioral domains from an implementation perspective.



QUALITY GUIDELINES

- A design should exhibit an architecture that (1) has been created using recognizable architectural styles or patterns, (2) is composed of components that exhibit good design characteristics and (3) can be implemented in an evolutionary fashion
 - For smaller systems, design can sometimes be developed linearly.
- A design should be modular; that is, the software should be logically partitioned into elements or subsystems
- A design should contain distinct representations of data, architecture, interfaces, and components.
- A design should lead to data structures that are appropriate for the classes to be implemented and are drawn from recognizable data patterns.
- A design should lead to components that exhibit independent functional characteristics.
- A design should lead to interfaces that reduce the complexity of connections between components and with the external environment.
- A design should be derived using a repeatable method that is driven by information obtained during software requirements analysis.
- A design should be represented using a notation that effectively communicates its meaning.

