Unit: II
Design Concepts and Principles
Lecture:1

Software Design Process:

Software design is an iterative process through which requirements are translated into a blueprint for constructing the software.

- Initially, the design is represented at a high level of abstraction i.e. a level that can be directly traced to the specific system objective and more detailed data, functional and behavioral requirements.
- As design iterations occur, subsequent refinement leads to design representations at much lower levels of abstraction.

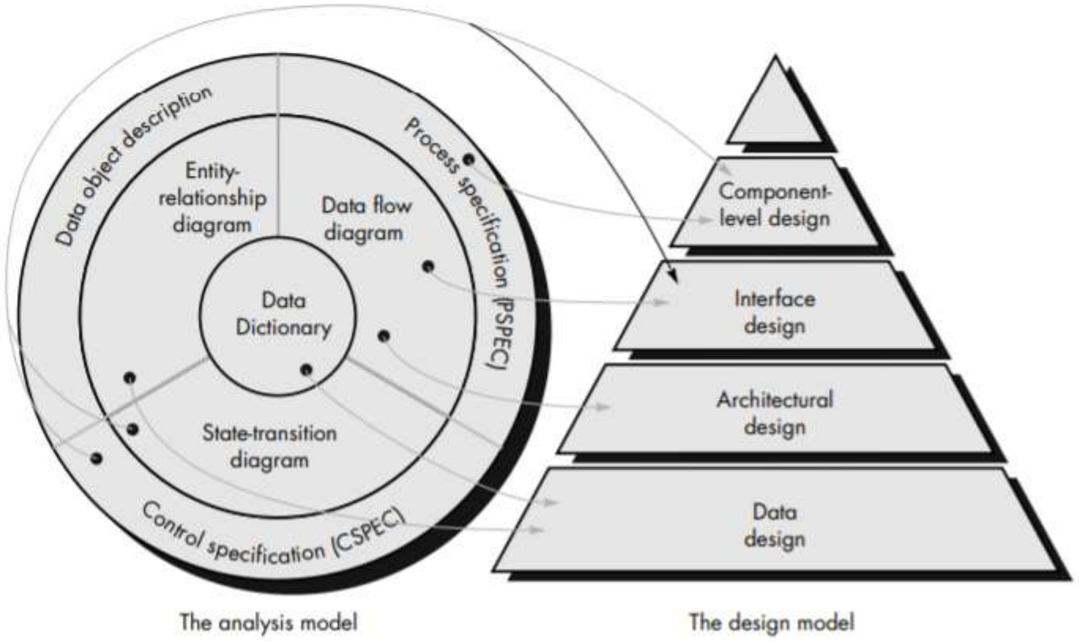

Fig: Translating the analysis model into a software design

The flow of information during software design is illustrated in above fig.

Software requirements, manifested by the data, functional and behavioral models, feed the design task.

The design task produces a data design, an architectural design, an interface design and a component design.

- **The Data Design:** It transforms the information domain model created during analysis into the data structures that will be required to implement the software.
    - The data objects and relationships defined in the ER diagram and the detailed data content depicted in the data dictionary provide the basis for the data design activity.

- **The Architectural Design:** defines the relationship between major structural elements of the software, the design patterns that can be used to achieve the requirements that have been defined for the system and the constraints that affect the way in which architectural design patters can be applied.
- **The Interface Design:** describes how the software communicates within itself, with systems that interoperate with it and with humans who use it. An interface implies a flow of information and a specific type of behavior.
- **The component-level design:** transforms structural elements of the software architecture into a procedural description of software components.

Importance of Software Design:

"Quality"

Design is the place where quality is fostered in software engineering. Design provides us with representations of software that can be assessed for quality.

- Design is the only way that we can accurately translate a customer's requirements into a finished software product or system.

Without design, we risk building an unstable system:

- one that will fail when small changes are made
- one that may be difficult to test
- one whose quality cannot be assessed until late in the software process, when time is short and many dollars have already been spent.

Three characteristics (suggested by McGlaughlin) that serve as a guide for evaluation of a good design/ Goal of the Design Process:

1. 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.
2. The design must be a readable, understandable guide for those who generate code and for those who test and subsequently support the software.
3. The design should provide a complete picture of the software, addressing the data, functional and behavioral domains from an implementation perspective.

Software Design Process vs Model:

Software design is both a process and a model.

The design process is a sequence of steps that enable the designer to describe all aspects of the software to be built.

The design model provides a variety of different views of the computer software. e.g. the design model is the equivalent of an architect's plans for a house. It begins by representing the totality of the thing to be built and slowly refines the thing to provide guidance for constructing each detail.

Principles for Software Design:

- The design should be traceable to the analysis model
- The design should exhibit uniformity and integration.
- The design should be structured to accommodate change.
- The design should be assessed for quality as it is being created, not after the fact.
- The design should be designed to accommodate unusual circumstances.