



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

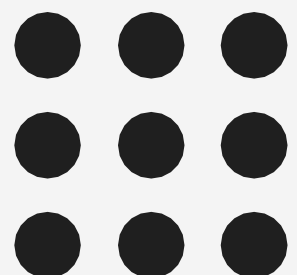
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Course Name – Software Engineering

II Year / III Semester

DESIGN CONCEPTS AND PRINCIPLES





Architectural Design



- The software needs the architectural design to represent the design of software.
- IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.”
- The software that is built for computer-based systems can exhibit one of these many architectural styles.

Architectural Styles



Each style describes a system category that encompasses:

- (1) a **set of components** (e.g., a database, computational modules) that perform a function required by a system,
- (2) a **set of connectors** that enable "communication, coordination and cooperation" among components,
- (3) **constraints** that define how components can be integrated to form the system,
- (4) **semantic models** that enable a designer to understand the overall properties of a system by analyzing the known properties of its constituent parts.

- Data-centered architectures
- Data flow architectures
- Call and return architectures
- Object-oriented architectures
- Layered architectures

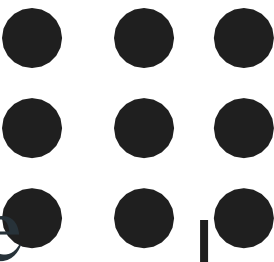


1.Data centered architectures:

- 1.A data store will reside at the center of this architecture and is accessed frequently by the other components that update, add, delete or modify the data present within the store.
- 2.This data-centered architecture will promote integrability. This means that the existing components can be changed and new client components can be added to the architecture without the permission or concern of other clients.
- 3.Data can be passed among clients using blackboard mechanism.

2. Data flow architectures:

1. This kind of architecture is used when input data to be transformed into output data through a series of computational manipulative components.
2. The figure represents pipe-and-filter architecture since it uses both pipe and filter and it has a set of components called filters connected by pipes.
3. Pipes are used to transmit data from one component to the next.
4. Each filter will work independently and is designed to take data input of a certain form and produces data output to the next filter of a specified form. The filters don't require any knowledge of the working of neighboring filters.
5. If the data flow degenerates into a single line of transforms, then it is termed as batch sequential. This structure accepts the batch of data and then applies a series of sequential components to transform it.



3.Call and Return architectures: It is used to create a program that is easy to scale and modify. Many sub-styles exist within this category. Two of them are explained below.

- **Remote procedure call architecture:** This components is used to present in a main program or sub program architecture distributed among multiple computers on a network.
- **Main program or Subprogram architectures:** The main program structure decomposes into number of subprograms or function into a control hierarchy. Main program contains number of subprograms that can invoke other components.



4. Object Oriented architecture: The components of a system encapsulate data and the operations that must be applied to manipulate the data. The coordination and communication between the components are established via the message passing.

5. Layered architecture:

1. A number of different layers are defined with each layer performing a well-defined set of operations. Each layer will do some operations that becomes closer to machine instruction set progressively.
2. At the outer layer, components will receive the user interface operations and at the inner layers, components will perform the operating system interfacing (communication and coordination with OS)
3. Intermediate layers to utility services and application software functions.



THANK YOU

