

Software Architecture and Architectural styles

Software Architecture

Bass, Clements, and Kazman defines the Software architecture as “The software architecture is structure or structures of the system, which consists of software components, the externally visible properties of those components, and the relationships among them”.

The architecture is not the operational software, it is a representation that is used to

- Meet its stated requirements effectively.
- Look for architectural alternatives which makes design changes relatively easy.
- Reduce the risks in the construction of the software.

In the architectural representation software component can be a module, class, database or middle ware configuration. Here the focus is on external relationships among components.

Importance of software architecture

Three reasons that describe the importance of software architecture.

1. The representation of software architecture enables the communication among all the stakeholders
2. The representation of software architecture enables the communication among all the stakeholders
The architecture focuses on the early design decisions that impact on all software engineering work and that leads to ultimate success of the system.

The Architecture represents a relatively small, intellectually graspable model of how the system is structured and how its components work together”.

Architectural Descriptions

The architectural description is actually a set of work products that reflect different views of the system.

Architecture can be viewed by different stake holders as follows

- The developer views the architecture as guide to proceed with design.
- Customer wants clear understanding of environment and check to see whether it meets business needs.
- Other stakeholders see the architecture in different aspects.

The IEEE standard defines an architectural description (AD) as “a collection of products to document an architecture.” The description consists of multiple views, where each view is “a representation of a whole system from the perspective of stakeholder.”

Architectural Decisions

Architect must look for variety of alternatives to to represent architecture in different views. so architectural decisions themselves can be considered to be one view of the architecture. We must document each major decision.

Architectural styles

The software to be built exhibit various architecture styles. These styles describe the system as follows.

1. A set of components perform the function required by the system.
 2. A set of connectors used for the communication, coordination and cooperation among the components.
 3. The constraints defined for the integration of components to form the system.
- Semantic models used to understand the overall properties of a system.

Taxonomy of Architectural Styles

computer systems are categorized into one of the following architectural styles.

Data-centered architecture

The data store that is file or database is located at the center of the architecture, and is accessed by other components that modify, update, add, delete data at data store.

The following figure illustrates typical data-centered architecture

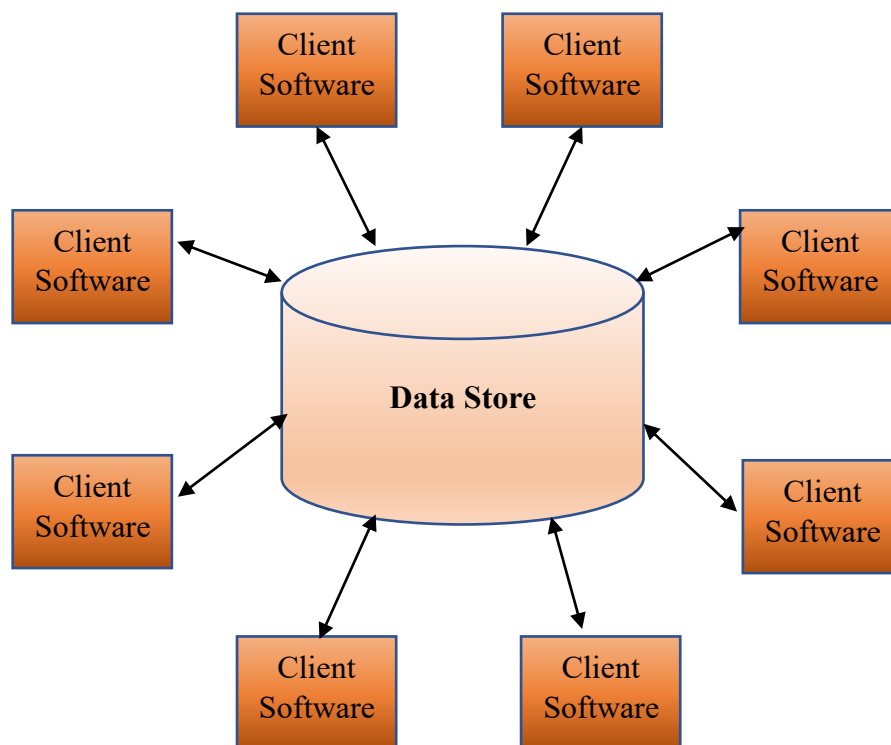


Figure: Data-centered Architecture

Data-flow architecture

This architecture style is applied when input data are to be transformed into output data through a series of computational components.

Pipe and filter pattern shown in the following figure is best example for this architecture style.

It contains set of components called as filters, connected by pipes transforms data from component to next. Each filter works independently and takes data of one form and transform it into another form.

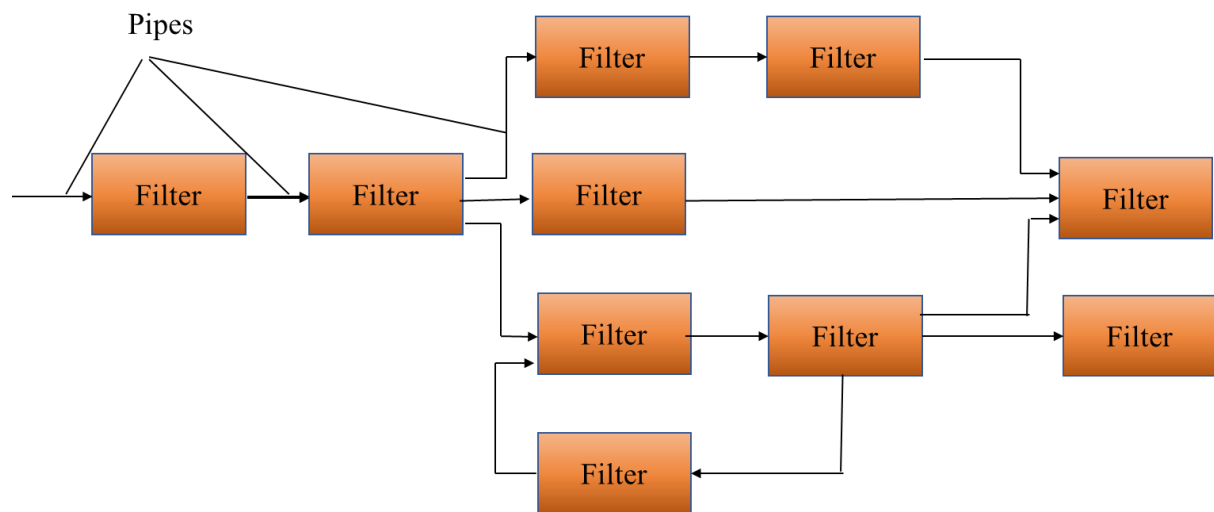


Figure: Data-flow Architecture

3. Call and return architectures

This architecture style allows to easily modify and scale the structure.

Following are the sub styles exist in this category.

1. Main program or subprogram architecture.

The program is divided into smaller pieces hierarchically, The main program invokes many of program components, they in turn may invoke other components.

2. Remote procedure call architecture

The components of main program or subprogram are distributed across the in network, to increase system performance.

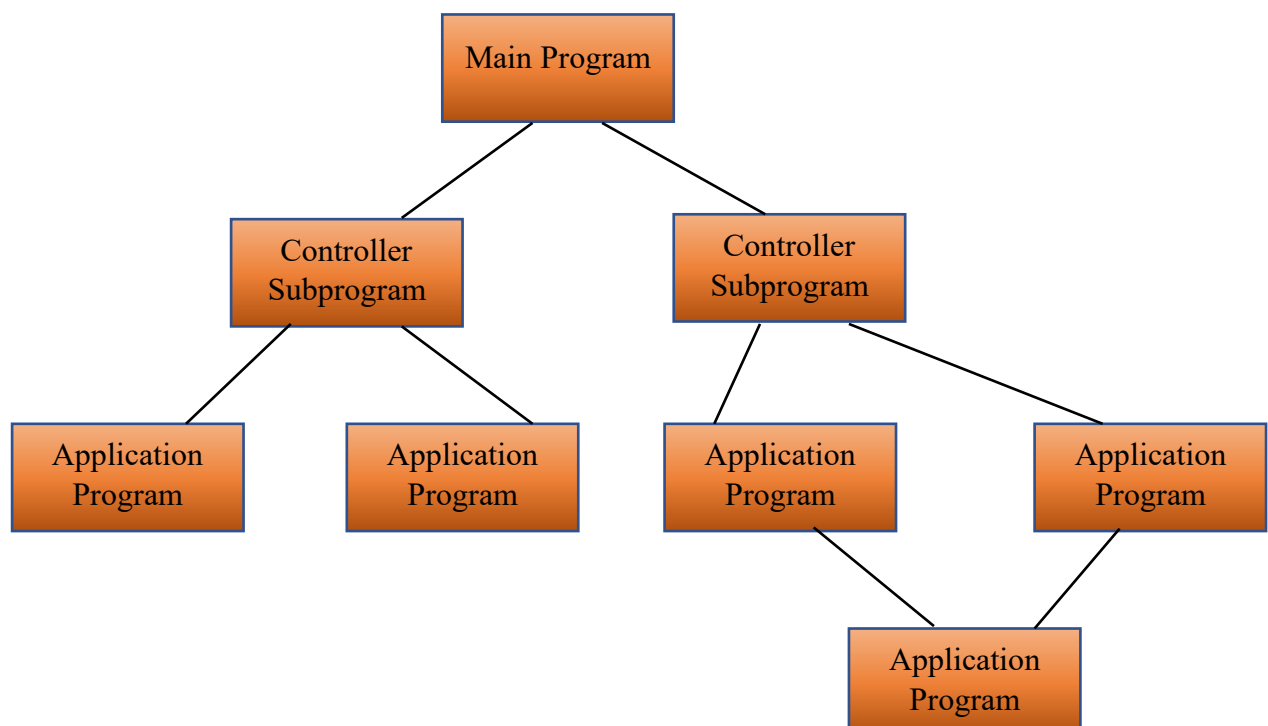


Figure: Main Program/Subprogram Architecture

4. Object-oriented architectures

The components of the system encapsulate both data and operations. The communication and coordination among the components take place via message passing.

5. Layered architectures

The basic structure of layered architecture is defined in the following figure

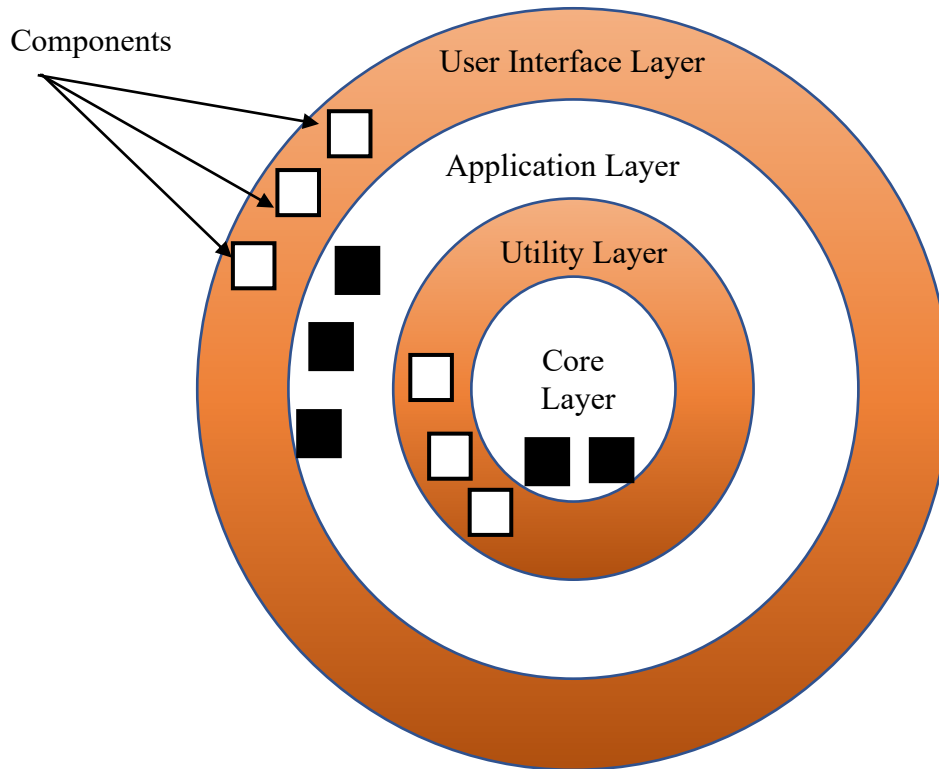


Figure: Layered Architecture

Different layers are defined, each perform their own operation and are progressively close to machine instructions.

Outer layer deals with interface operations, intermediate layer provide application program and utility services. Inner layer deals with operating system functions.

Text Books

1. Roger Pressman S., "Software Engineering: A Practitioner's Approach", 7th Edition, McGraw Hill, 2010.
2. Sommerville, "Software Engineering", Eighth Edition, Pearson Education, 2007

Web links

1. https://cdn.shopify.com/s/files/1/0457/4009/7694/files/software_engineering_pdf_pressman_7th_edition.pdf