Software Design and Architecture

Interaction Oriented Software Architectures

Introduction

- Software applications involve user input/output interactions and specific user interfaces to software systems.

- The primary objective of interaction oriented architecture is to separate the interaction of user from data abstraction and business data processing.

- The interaction oriented software architecture decomposes the system into three major partitions:

Data module

Control module

View presentation module

- **Data Module**

- The data module provides the data abstraction and all core business logic on data processing.
- The view presentation module is responsible for visual or audio data output presentation and it may also provide user input interface as well when necessary.

Control Module

- The control module determines the flow of control involving view selections, communications between modules, job dispatching, and certain data initialization and system configuration actions.
- The key point of this architecture is in the separation of user interactions from data abstraction and business data processing.

View Module

- Since there may be many view presentations in different formats multiple views may be supported for the same data set.

- Even for a specific view presentation, the interfaces or views may need to change often, so loose coupling between data abstractions and its presentations is helpful, and this is supported by this style.

- The loose coupling connections can be implemented in many different ways such as explicit method invocation or implicit registration/notification method invocation.

- The control module plays a central role that mediates the data module and view presentation modules.

- All three modules may be fully connected.

Styles of Interaction Oriented Software Architectures

- Model-View-Controller (MVC) architectures.
- Presentation-Abstraction-Control (PAC)

- These two models are very similar in the sense that they propose three component decompositions.

- The Presentation module of PAC is like the View module of MVC.
- The Abstraction module of PAC looks like the data (or Model) module of MVC.

- The Control module of PAC is like the Controller module of MVC.

- Both of MVC and PAC are used for interactive applications such as Web online applications, and distributed applications with multiple talks and user interactions.

- They are different in their flow of control and organization.

- The PAC is an agent based hierarchical architecture, whereas the MVC does not have a clear hierarchical structure and all three modules are connected together.

Summary

- Introduce Interaction Oriented Architecture
- Introduce MVC and PAC Architecture