

Software Engineering

Week # 8

LECTURER: ABDULRAHMAN QAIM

Software architecture

The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is **architectural design**.

The output of this design process is a description of the **software architecture**.

Architectural design

An early stage of the system design process.

Represents the link between specification and design processes.

Often carried out in parallel with some specification activities.

It involves identifying major system components and their communications.

Architectural abstraction

Architecture in the small is concerned with the architecture of individual programs. At this level, we are concerned with the way that an individual program is decomposed into components.

Architecture in the large is concerned with the architecture of complex enterprise systems that include other systems, programs, and program components. These enterprise systems are distributed over different computers, which may be owned and managed by different companies.

Architectural representations

Simple, informal block diagrams showing entities and relationships are the most frequently used method for documenting software architectures.

But these have been criticised because they lack semantics, do not show the types of relationships between entities nor the visible properties of entities in the architecture.

Depends on the use of architectural models. The requirements for model semantics depend on how the models are used.

Box and line diagrams

Very abstract - they do not show the nature of component relationships nor the externally visible properties of the sub-systems.

However, useful for communication with stakeholders and for project planning.

Use of architectural models

As a way of facilitating discussion about the system design

- A high-level architectural view of a system is useful for communication with system stakeholders and project planning because it is not cluttered with detail. Stakeholders can relate to it and understand an abstract view of the system. They can then discuss the system as a whole without being confused by detail.

As a way of documenting an architecture that has been designed

- The aim here is to produce a complete system model that shows the different components in a system, their interfaces and their connections.

Architectural design decisions

Architectural design is a creative process so the process differs depending on the type of system being developed.

However, a number of common decisions span all design processes and these decisions affect the non-functional characteristics of the system.

Architectural design decisions

Is there a generic application architecture that can be used?

How will the system be distributed?

What architectural styles are appropriate?

What approach will be used to structure the system?

How will the system be decomposed into modules?

What control strategy should be used?

How will the architectural design be evaluated?

How should the architecture be documented?

Architecture and system characteristics

Performance

- Localise critical operations and minimise communications. Use large rather than fine-grain components.

Security

- Use a layered architecture with critical assets in the inner layers.

Safety

- Localise safety-critical features in a small number of sub-systems.

Availability

- Include redundant components and mechanisms for fault tolerance.

Maintainability

- Use fine-grain, replaceable components.

Architectural views

What views or perspectives are useful when designing and documenting a system's architecture?

What notations should be used for describing architectural models?

Each architectural model only shows one view or perspective of the system.

- It might show how a system is decomposed into modules, how the run-time processes interact or the different ways in which system components are distributed across a network. For both design and documentation, you usually need to present multiple views of the software architecture.

4 + 1 view model of software architecture

A logical view, which shows the key abstractions in the system as objects or object classes.

A process view, which shows how, at run-time, the system is composed of interacting processes.

A development view, which shows how the software is decomposed for development.

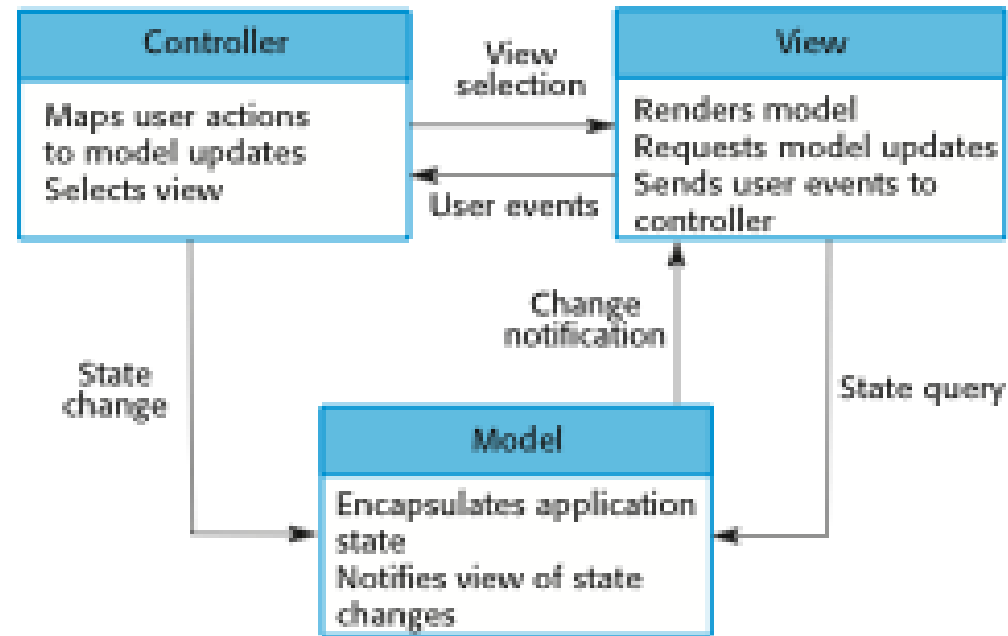
A physical view, which shows the system hardware and how software components are distributed across the processors in the system.

Related using use cases or scenarios (+1)

The Model-View-Controller (MVC) pattern

Name	MVC (Model-View-Controller)
Description	Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model. See Figure 6.3.
Example	Figure 6.4 shows the architecture of a web-based application system organized using the MVC pattern.
When used	Used when there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown.
Advantages	Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways with changes made in one representation shown in all of them.
Disadvantages	Can involve additional code and code complexity when the data model and interactions are simple.

The organization of the Model-View-Controller



Layered architecture

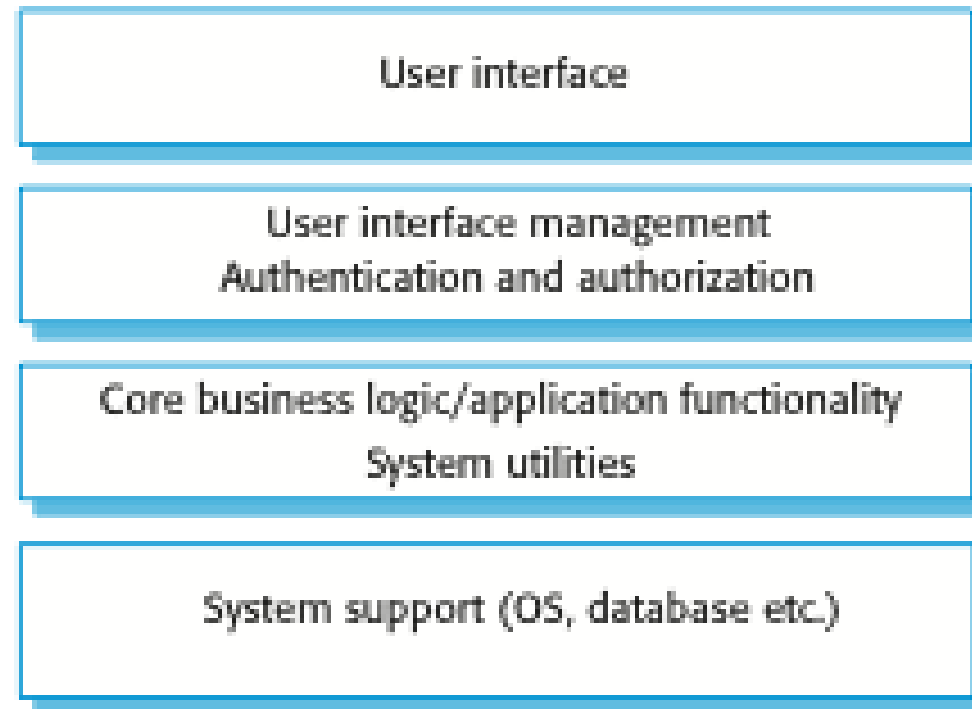
Used to model the interfacing of sub-systems.

Organises the system into a set of layers (or abstract machines) each of which provide a set of services.

Supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.

However, often artificial to structure systems in this way.

The Layered architecture pattern



Key points

A software architecture is a description of how a software system is organized.

Architectural design decisions include decisions on the type of application, the distribution of the system, the architectural styles to be used.

Architectures may be documented from several different perspectives or views such as a conceptual view, a logical view, a process view, and a development view.

Architectural patterns are a means of reusing knowledge about generic system architectures. They describe the architecture, explain when it may be used and describe its advantages and disadvantages.