Chapter 6 – Architectural Design

Topics covered

✧ Architectural design decisions

✧ Architectural views

✧ Architectural patterns
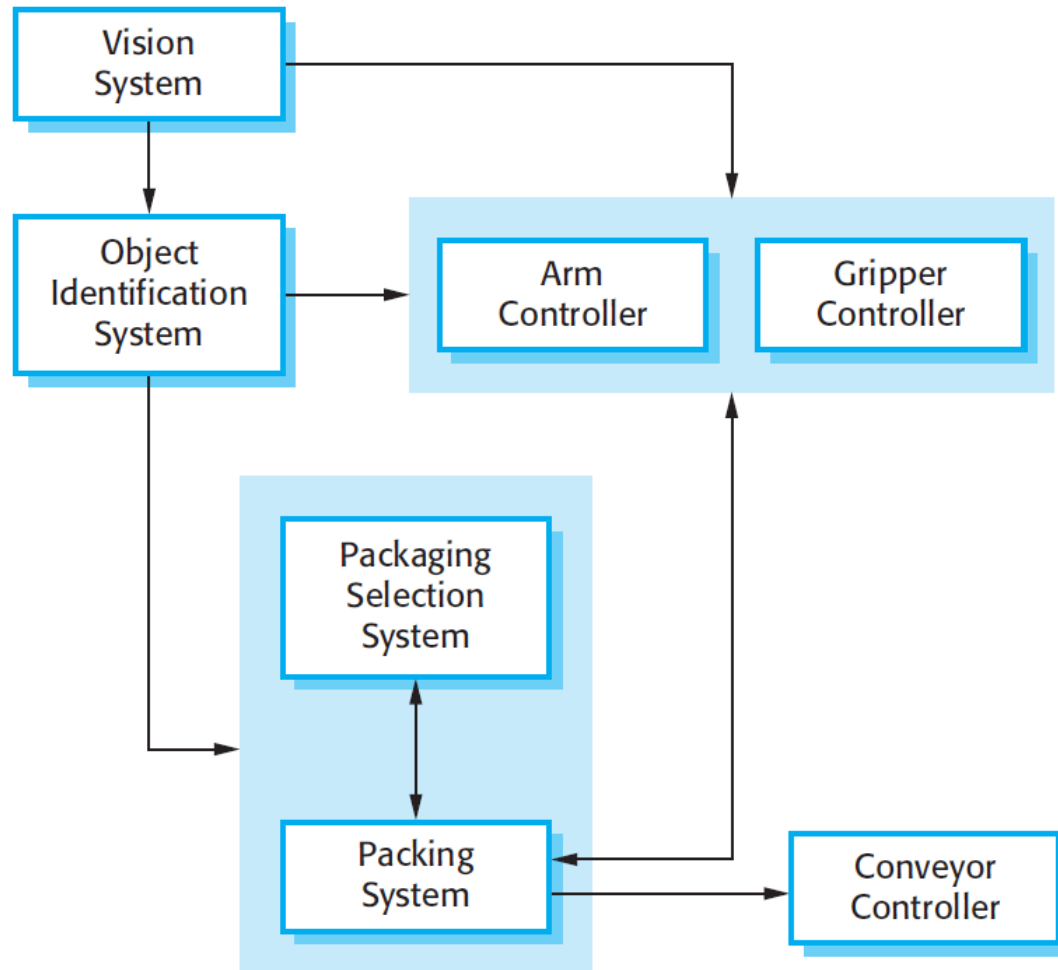
✧ Application architectures

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.

The architecture of a packing robot control system

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.

Advantages of explicit architecture

✧ Stakeholder communication

- Architecture may be used as a focus of discussion by system stakeholders.

✧ System analysis

- Means that analysis of whether the system can meet its non-functional requirements (performance, reliability, maintainability) is possible.

✧ Large-scale reuse

- The architecture may be reusable across a range of systems
  - Large-scale software reuse
- Product-line architectures may be developed.

Architectural representations

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

✧ But these have been criticized 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 depends on how the models are used.

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; Also, 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 across a number of cores or processors?

- ✧ What architectural styles are appropriate?

- ✧ How will the system be decomposed into modules?

- ✧ What component control strategy should be used?

- ✧ How will the architectural design be evaluated?

- ✧ How should the architecture be documented?

Architecture reuse

✧ Systems in the same domain often have similar architectures that reflect domain concepts.

✧ Application product lines are built around a core architecture with variants that satisfy particular customer requirements.

✧ The architecture of a system may be designed around one of more architectural patterns or 'styles'.

  ▪ These capture the essence of an architecture and can be instantiated in different ways.

  ▪ Discussed later in this lecture.

Architecture and system characteristics

- ✧ Performance
  - ▪ Localize critical operations and minimize communications. Use large rather than fine-grain components. Or allow replications.
- ✧ Security
  - ▪ Use a layered architecture with critical assets in the inner layers.
- ✧ Safety
  - ▪ Localize 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 together using use cases or scenarios (+1)

Architectural patterns

✧ Patterns are a means of representing, sharing and reusing knowledge.

✧ An architectural pattern is a stylized description of good design practice, which has been tried and tested in different environments.

✧ Patterns should include information about when they are and when the are not useful.

✧ Patterns may be represented using tabular and graphical descriptions.

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

| Name	Layered architecture
Description	Organizes the system into layers with related functionality associated with each layer. A layer provides services to the layer above it so the lowest-level layers represent core services that are likely to be used throughout the system. See Figure 6.6.
Example	A layered model of a system for sharing copyright documents held in different libraries, as shown in Figure 6.7.
When used	Used when building new facilities on top of existing systems; when the development is spread across several teams with each team responsibility for a layer of functionality; when there is a requirement for multi-level security.
Advantages	Allows replacement of entire layers so long as the interface is maintained. Redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system.
Disadvantages	In practice, providing a clean separation between layers is often difficult and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.

A generic layered architecture

User Interface

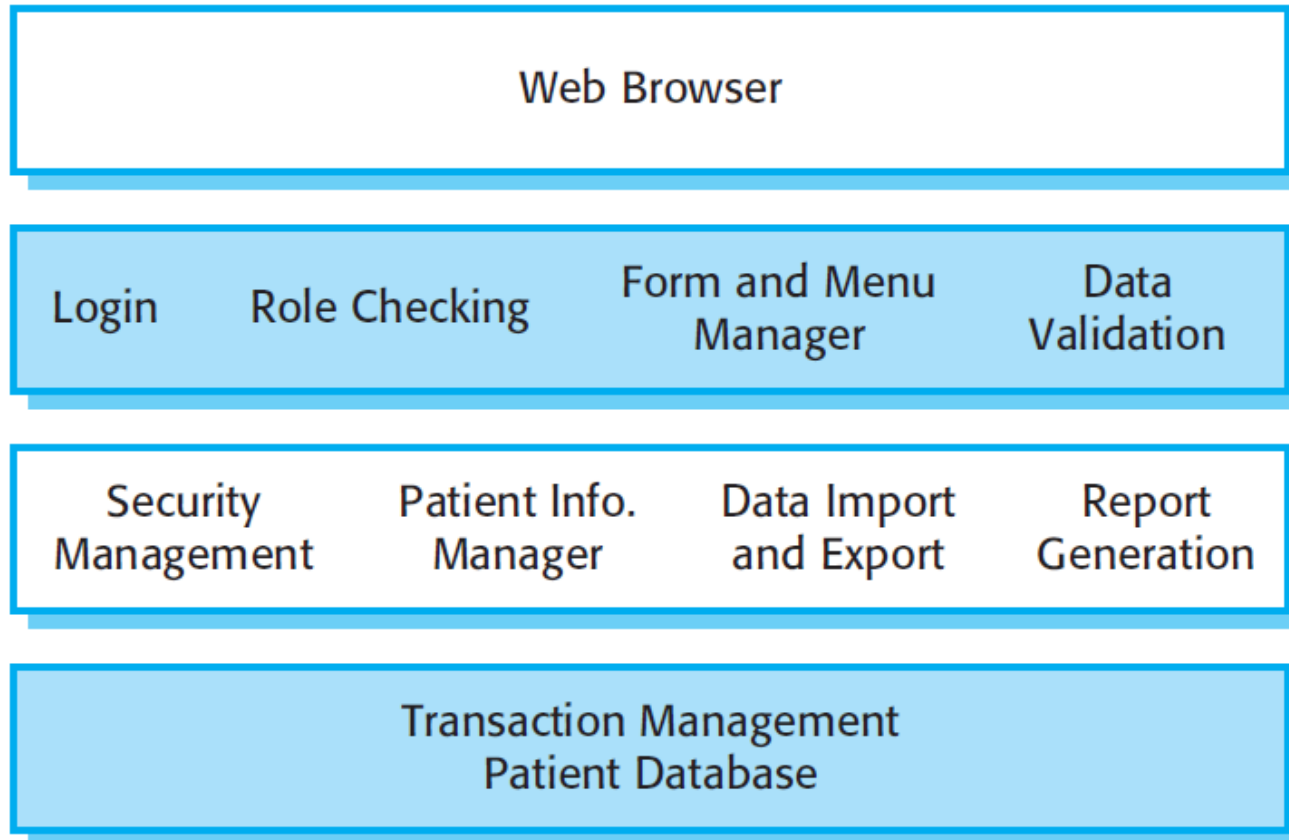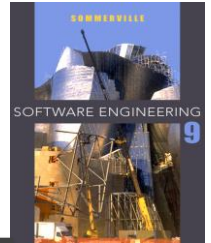| User Interface Management
Authentication and Authorization |
|---|

Core Business Logic/Application Functionality System Utilities

System Support (OS, Database etc.)

The architecture of the LIBSYS system
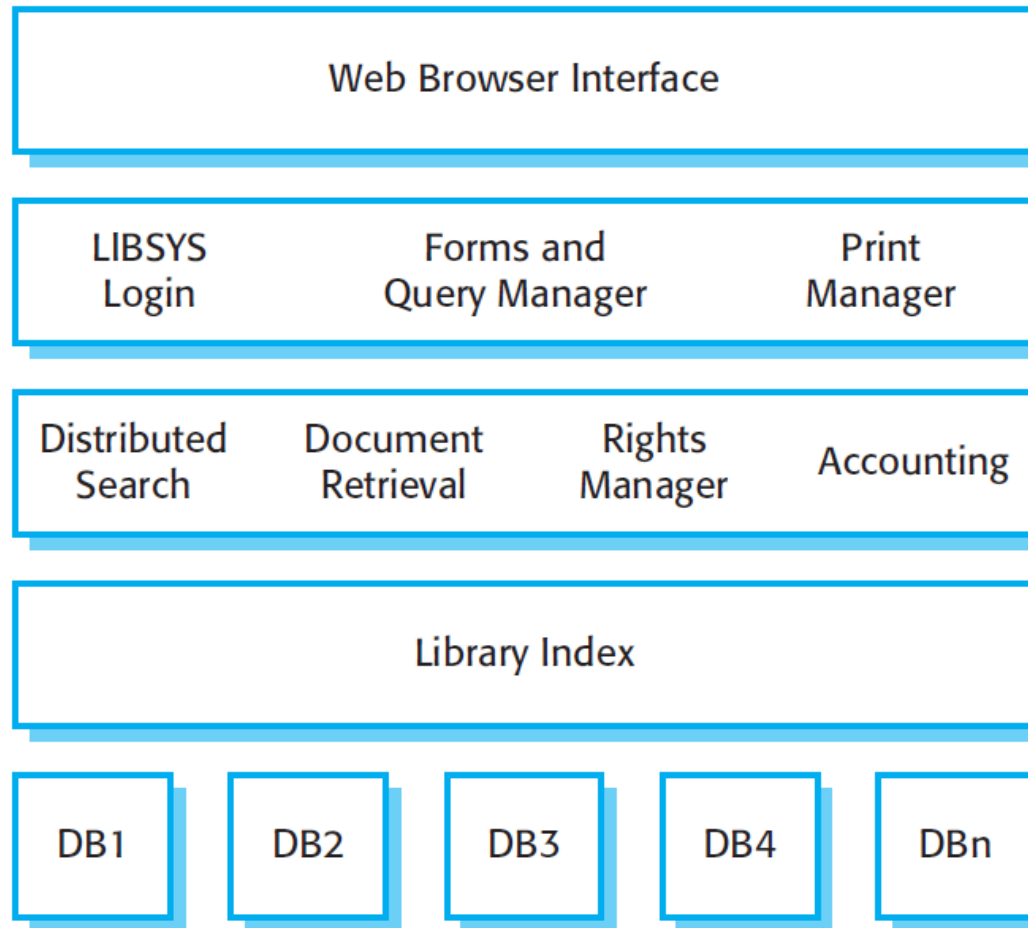
The architecture of the MHC-PMS

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.