CS 320
Fundamentals of Software Engineering

Lecture 11: Architecture Design
Software Architecture

- The design process for identifying the sub-systems 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 the design processes
- Often carried out in parallel with some specification activities
- It involves identifying major system components and their communications
Architectural Representations

- Simple, informal block diagrams showing entities and relationships
- "Box and line" diagrams
- Very abstract, but useful for communication with stakeholders and for project planning
A Packing Robot Control System

Vision system

Object identification system

Arm controller

Gripper controller

Packaging selection system

Packing system

Conveyor controller
System in the same domain often have similar architectures that reflect domain concepts.

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

- Patterns are a means of representing, sharing and reusing knowledge

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

- Patterns should include information about when they are (not) useful

- Patterns may be represented using tabular and graphical descriptions
Model-View-Controller

Separates presentation and interaction from the system data
Web Application Architecture using MVC Pattern

Tabular representation of MVC pattern:

<table>
<thead>
<tr>
<th>Name</th>
<th>MVC (Model-View-Controller)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
<tr>
<td>Example</td>
<td>Figure 6.4 shows the architecture of a web-based application system organized using the MVC pattern.</td>
</tr>
<tr>
<td>When used</td>
<td>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.</td>
</tr>
<tr>
<td>Advantages</td>
<td>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.</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>Can involve additional code and code complexity when the data model and interactions are simple.</td>
</tr>
</tbody>
</table>
Layered Architecture

- Used to model the interfacing of sub-systems
- Organizes the system into a set of layers, each of which provide a set of services
- Supports incremental development of sub-system in different layers
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

- Web browser interface
- LIBSYS login
- Forms and query manager
- Print manager
- Distributed search
- Document retrieval
- Rights manager
- Accounting
- Library index
- DB1
- DB2
- DB3
- DB4
- DBn
Repository Architecture

- Sub-systems must exchange data
  - Shared data is held in a central database or repository
  - Each sub-system maintains its own database and passes data explicitly to other sub-systems
- When large amounts of data are to be shared, the repository model of sharing is most commonly used
A Repository Architecture for an IDE
Client-Server Architecture

- Distributed system model which shows how data and processing is distributed across a range of components
- Set of stand-alone servers which provide specific services
- Set of clients which call on these services
- Network which allows clients to access servers
A Client-Server Architecture for a Film Library

Client 1

Catalog server
Library catalogue

Client 2

Video server
Film store

Client 3

Picture server
Photo store

Client 4

Web server
Film and photo info.

Internet
Pipe and Filter Architecture

- Functional transformations process their inputs to produce outputs
- May be referred to as a pipe and filter model
- Not really suitable for interactive systems
A Billing System Using Pipe and Filter Architecture
Transaction System

• ATM

**Input**
- Get customer account id
- Validate card
- Select service

**Process**
- Query account
- Update account

**Output**
- Print details
- Return card
- Dispense cash
Examples of Application Types

✤ Information systems
  ✤ have a generic architecture that can be organized as a layered architecture
  ✤ These are transaction-based systems as interaction with these systems generally involves database transactions

✤ Layers include
  ✤ The user interface
  ✤ User communications
  ✤ Information retrieval
  ✤ System database
Layered Information System Architecture

- User interface
- User communications
- Authentication and authorization
- Information retrieval and modification
- Transaction management
- Database
The Architecture of the MHC-PMS

- Web browser
- Login
- Role checking
- Form and menu manager
- Data validation
- Security management
- Patient info. manager
- Data import and export
- Report generation
- Transaction management
- Patient database