8. Architectural Design

Yan Shi

SE 2730 Lecture Notes

Part of this notes are from Summerville’s Textbook Notes Chapter 11
Software architecture

- Architectural Design is the design process for identifying
  - the sub-systems making up a system;
  - the framework for sub-system control and communication

- 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.
  - Reuse is common: architectural styles.
The architectural model of a system may conform to a generic architectural model or style.

An awareness of these styles can simplify the problem of defining system architectures.

However, most large systems are heterogeneous and do not follow a single architectural style.
How to choose architecture styles

The particular architecture style chosen for an application may depend on the non-functional system requirements:

- **Performance**
  - Localize critical operations and minimize communications.
  - Use large rather than fine-grain components.

- **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, self-contained, replaceable components.
Architectural conflicts

- Using large-grain components
  - improves performance but reduces maintainability.

- Introducing redundant data
  - improves availability but makes security more difficult.

- Localizing safety-related features usually means
  - more communication so degraded performance.

Compromise solution must be found:

*Use different architectural styles for different parts of a system!*
System structuring

- Concerned with decomposing the system into interacting sub-systems.
- The architectural design is normally expressed as a block diagram presenting an overview of the system structure.
- More specific models showing how sub-systems share data, are distributed and interface with each other may also be developed.
Packing robot control system

- Vision system
- Object identification system
- Arm controller
- Gripper controller
- Packaging selection system
- Packing system
- Conveyor controller
Static Structural Architecture Styles

- repository model
- client-server model
- layer model
The repository model

- Sub-systems must exchange data. This may be done in two ways:
  - Shared data is held in a central database or repository and may be accessed by all sub-systems;
  - 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.
  - MIS: management information system
  - CAD
  - CASE
CASE toolset architecture

- Design translator
- Design editor
- Code generator
- Project repository
- Design analyser
- Report generator
- Program editor
Repository model characteristics

- **Advantages**
  - Efficient way to share large amounts of data;
  - Sub-systems need not be concerned with how data is used or produced;
  - Centralised management e.g. backup, security, etc.
  - Sharing model is published as the repository schema.

- **Disadvantages**
  - Sub-systems must agree on a repository data model: inevitably a compromise;
  - Data evolution is difficult and expensive;
  - No scope for specific management policies;
  - Difficult to distribute efficiently.
Client-server model

- Distributed system model which shows how data and processing is distributed across a range of components.

- Set of **servers** which provide specific services such as printing, data management, etc.

- Set of **clients** which call on these services.

- **Network** which allows clients to access servers.

- Examples:
  - web browser based application
  - Email, FTP
  - UWPClass
Film and picture library
Client-server characteristics

- Advantages
  - Distribution of data is straightforward;
  - Makes effective use of networked systems. May require cheaper hardware;
  - Easy to add new servers or upgrade existing servers.

- Disadvantages
  - No shared data model so sub-systems use different data organisation. Data interchange may be inefficient;
  - Redundant management in each server;
  - No central register of names and services - it may be hard to find out what servers and services are available.
Layered model

- Used to model the interfacing of sub-systems.
- Organises the system into a set of layers (or abstract machines)
- Each layer
  - provide a set of services.
  - has its own interface
  - only “talks” to its immediate neighbours (usually)

- OSI models, TCP/IP
- A typical 3-layered model:
  - Interface ↔ Business Logic ↔ Data
OSI Layers

The Seven Layers of OSI

GNU/Linux OS Architecture

Typical 3-Layered Model

http://www.hanselman.com/blog/AReminderOnThreeMultiTierLayerArchitectureDesignBroughtToYouByMyLateNightFrustrations.aspx
Layer model characteristic

- **Advantage:**
  - Supports the incremental development of sub-systems in different layers.
  - When a layer’s interface changes, only the adjacent layer is affected.
  - Easier to provide multi-platform implementations of an application system: only the inner, machine-dependent layer need to be re-implemented.

- **Disadvantage:**
  - Structuring systems can be difficult.
  - Multiple levels of command interpretation may jeopardise system performance.
Combining Architecture Styles

- Most software do NOT use only one architecture style
  - client/server & repository
  - client/server & layered
  - ...

Summary

- Software architecture
- Architectural design
- Architectural models
- System organization model:
  - repository
  - client-server
  - layer

They are often combined!