Today’s Lecture

• Introduction and Background of
  – Software Architecture
    • concepts
    • styles
    • domains

Software Architecture

• The principled study of software components, including their properties, relationships, and patterns of combination

• Also, a particular set of software components as combined in a particular software system
The Role of Architecture

Internal decomposition, functions, and data structures

Buy/build strategy for each component

Major components and interfaces

Problem definition, rationale, and financial plan

System features, functions, and usage scenarios

Low-level Design

High-level Design

Architecture

Requirements

Business Case

The Central Planning Phase

Good Architecture Lowers Cost

Planning Phases

Low-level Design

High-level Design

Architecture

Requirements

Business Case

Planning Phases

Low-level Design

High-level Design

Architecture

Requirements

Business Case

Cost of Error Repair

Phase of Error Discovery
Addresses Integration Issues

- Error Count
  - System Test Errors
  - Integration Test Errors
  - Unit Test Errors

Architectural Domains

- Application Domain
  - Translation
  - Data Processing
  - Transaction Processing
  - Telephone Call Processing
  - Flight Dynamics Control

- Computer Science Domain
  - Bus
  - Pipes and Filters
  - Layered Abstract Machines
  - Shared Repository
  - Clients and Servers
Problems ↔ Solutions

Application Domain
A program is compiled by successive application of lexical analysis, syntactic analysis, semantic analysis, and code generation.

Computer Science Domain
The output of one component is the input to another component.

Problems ↔ Solutions

Application Domain
Each agent in the telephone network is a separate computational entity.

Computer Science Domain
Each operating system process is a separate computational entity.

Problems ↔ Solutions

Application Domain
Network services are organized as layers, where each layer adds value to services of lower layers. Services at a layer are defined independently of how they are performed.

Computer Science Domain
Components are organized as layers, such that each layer provides services only to next higher layer and uses services only of next lower layer.

Architectural Style

• Generic set of components and arrangement of those components
• Constraint on components and their interconnections
• Often given a name...convenient for quickly conveying essential information
Example: DBMS Architecture

- Traditional Business Data Processing
  - Processing steps are independent programs
  - Each runs to completion before next step starts
  - Data stored and passed through magnetic tapes

- Centralized Database
  - Monolithic query/update/store engine
  - Applications as serializable transactions

Example: DBMS Architecture

- Traditional Business Data Processing
  - Processing steps are independent programs
  - Each runs to completion before next step starts
  - Data stored and passed through magnetic tapes

- Centralized Database
  - Monolithic query/update/store engine
  - Applications as serializable transactions
Example: DBMS Architecture

- Modern Database Toolkits
  - Discrete, interchangeable components
  - Applications as distributed clients

![DBMS Architecture Diagram]

Analogy: Chemical Engineering

Chemical engineering evolved from a mixture of craft, mysticism, wrong theories, and empirical guesses. Improvements were very slow until the Scientific Revolution. Only then were mystical interpretations replaced by scientific theories: though the early theories were often wrong, they played a leading role in stimulating thought.

- J.T. Davies

Interesting Architectural Points
- Theory ignored by engineers
- “Hacking” worked until problems with scale
- Scale problems solved by development of relatively small number of unit operations
- Strong emphasis on relationship to process
- Handbook of chemical engineering eventually developed and used
Unit Operations of Chemical Engineering

- Distillation
- Evaporation
- Drying
- Filtration
- Absorption
- Extraction

Analogy: Computer Engineering

- Interesting Architectural Points
  - Relatively small number of component kinds
  - Properties constrained by physical laws
  - Scale by replicating components
- Compared to Software Architecture
  - Very large number of component kinds
  - Constraints hard to determine
  - Scale by adding new component kinds

Analogy: Civil Engineering

- Architectural Styles
  - Colonial, Victorian, Ranch, etc.
  - Pipes and filters, layers, client/server, etc.
- Building Codes
  - Electrical, structural, zoning, etc.
  - Formal specifications
- Special Expertise
  - Slate roofs, post and beam, logs, etc.
  - Domain-specific architectures

Unit Operations of Software Engineering

- Input Validation
- Status Monitoring
- Load Balancing
- Translation
- Filtering
- Multicasting
- …