

Architectural Design 1

Suradet Jitprapaikulsarn

Derived from Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill 2005

What is Software Architecture?

The structure of a system and the manner in which data and procedural components collaborate with one another

Pressman, R., 2005

Why Architecture?

- Help analyzing the design effectiveness
- Help considering the architectural alternatives
- Help reducing the risks associated with software construction

Why is Architecture important?

- Architecture enables communication between all stakeholders
- Architecture highlights early design decisions
- Architecture manifests how the system is structured and how its components work together

Design of Software Architecture

- Data Design
- Architecture Design

Data Design

- data structure at component-level
- database at application level
- data warehouse at business level

Data Design at Architecture Level

- Design databases for application architecture
- Design the data warehouse for accessing multiple databases
- Design the data mining for extracting knowledge from existing databases

Data Design at Component Level

- refine data objects and develop a set of data abstractions
- implement data object attributes as one or more data structures
- review data structures to ensure that appropriate relationships have been established
- simplify data structures as required

Principles for data specification

1. Apply systematic analysis principle
2. Take into account the operation to be performed
3. Establish a mechanism for defining content and operations applied to it
4. Defer low-level design decisions
5. Only modules that directly use the data containing within a data structure should know the representation
6. Develop a library of data and its operations
7. Select the software design and programming language that can realize the abstract data types

Architecture Styles

- Components
- Connectors
- Constraints
- Semantic

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
10

Architectural Styles

- Data-centered architecture
- Data-flow architecture
- Call and return architecture
- Object-oriented architecture
- Layered architecture

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
11

Data-centered architecture

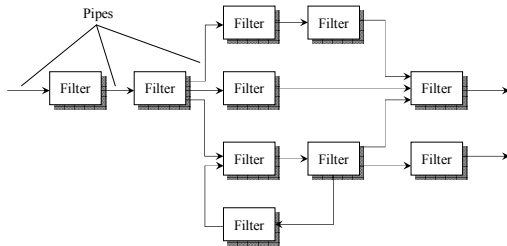
```

    graph TD
      CS1[Client software] --> DS((Data store repository or blackboard))
      CS2[Client software] --> DS
      CS3[Client software] --> DS
      CS4[Client software] --> DS
      CS5[Client software] --> DS
  
```

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
12

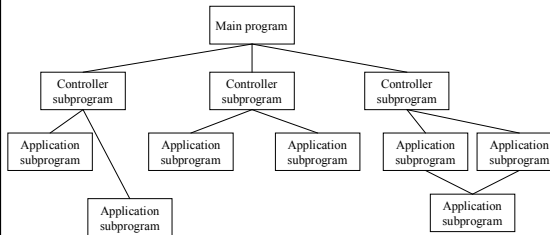
Data-flow architecture



Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005
 បានមកពីសៀវភៅ ៖ បច្ចេកវិទ្យា គណនេយ្យ ទំព័រ ២៤៤៤
 ចំណុចទី១៣៧២ ក្នុង វិស័យ គណនេយ្យ

13

Call and return architecture



Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005
 បានមកពីសៀវភៅ ៖ បច្ចេកវិទ្យា គណនេយ្យ ទំព័រ ២៤៤៤
 ចំណុចទី១៣៧២ ក្នុង វិស័យ គណនេយ្យ

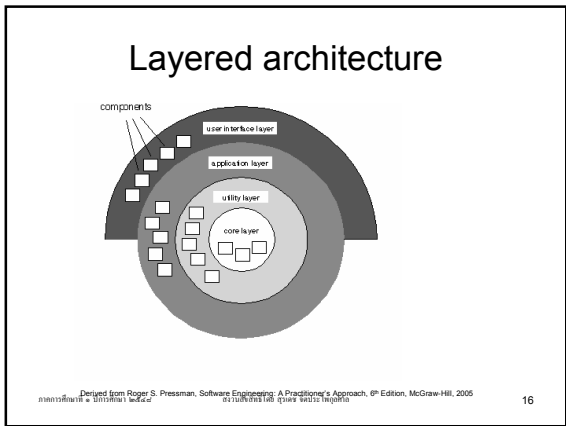
14

Object-oriented architecture

- The components encapsulate data and operations
- Communication and coordination are via message passing

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005
 បានមកពីសៀវភៅ ៖ បច្ចេកវិទ្យា គណនេយ្យ ទំព័រ ២៤៤៤
 ចំណុចទី១៣៧២ ក្នុង វិស័យ គណនេយ្យ

15



Architectural Pattern

- Focus on one aspect of architecture
- Impose a rule on the architecture
- Address specific behavioral issues

ศึกษาระบบสถาปัตยกรรมแบบเลเยอร์ ศึกษาระบบสถาปัตยกรรมแบบเลเยอร์

17

Architectural Patterns

- Concurrency—simulate parallelism
 - operating system management pattern
 - task scheduler pattern
- Persistence—data are stored for later use
 - database management system pattern
 - application level persistence pattern
- Distribution—communication in a distributed environment
 - broker pattern

ศึกษาระบบสถาปัตยกรรมแบบเลเยอร์ ศึกษาระบบสถาปัตยกรรมแบบเลเยอร์

18

Organization and Refinement

- Control
 - How is control managed within the architecture?
 - Does a distinct control hierarchy exist?
 - What is the role of components within the control hierarchy?
 - How do components transfer control within the system?
 - How is control shared among components?
 - What is the control topology?
 - Is control synchronized or do operate asynchronously?

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005

19

Organization and Refinement

- Data
 - How are data communicated between component?
 - Is the flow of data continuous, or are data objects passed to the system sporadically?
 - What is the model of data transfer?
 - Do data components exist?
 - What is the role of data components?
 - How do functional components interact with data components?
 - Are data components passive or active?
 - How do data and control interact within the system?

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005

20

Architectural Design

- Software must be put into context
 - Define the external entities and the nature of the interaction between entities and software
 - Specify the structure of the system

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005

21
