

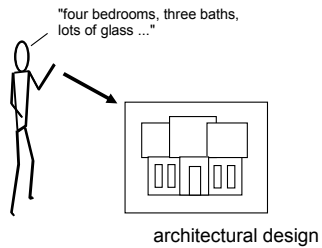
Architectural Design III

Suradet Jitrapaikulsarn

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

An Architectural Design Method

customer requirements

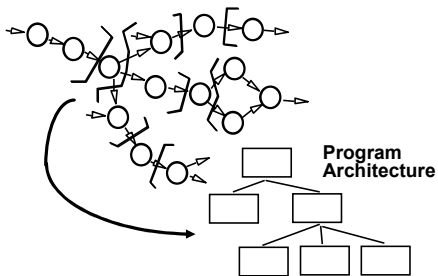


มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ภาควิชาวิศวกรรมคอมพิวเตอร์

2

Deriving Program Architecture

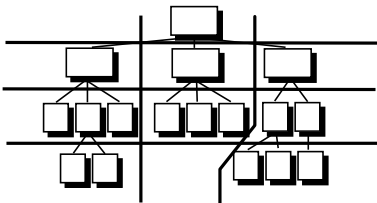


มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ภาควิชาวิศวกรรมคอมพิวเตอร์

3

Partitioning the Architecture



ศาสตราจารย์ ดร. ปิยะพรพรรณ ๒๕๕๘

ศูนย์วิจัยเทคโนโลยี การเกษตร ๒๕๕๘

4

Why Partitioning

- results in software that is easier to test
- leads to software that is easier to maintain
- results in propagation of fewer side effects
- results in software that is easier to extend

ศาสตราจารย์ ดร. ปิยะพรพรรณ ๒๕๕๘

ศูนย์วิจัยเทคโนโลยี การเกษตร ๒๕๕๘

5

Horizontal Partitioning

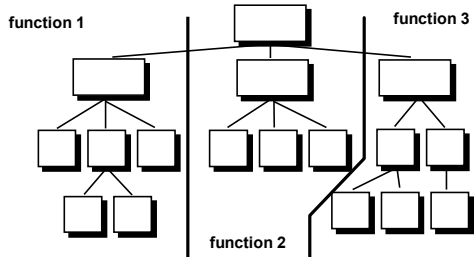
- define separate branches of the module hierarchy for each major function
- use control modules to coordinate communication between functions

ศาสตราจารย์ ดร. ปิยะพรพรรณ ๒๕๕๘

ศูนย์วิจัยเทคโนโลยี การเกษตร ๒๕๕๘

6

Horizontal Partitioning



ศาสตราจารย์ ดร. ปิยะสิทธิ์ ๒๕๕๘

ศูนย์วิจัยระบบ ฐานข้อมูลและปัญญาประดิษฐ์

7

Vertical Partitioning

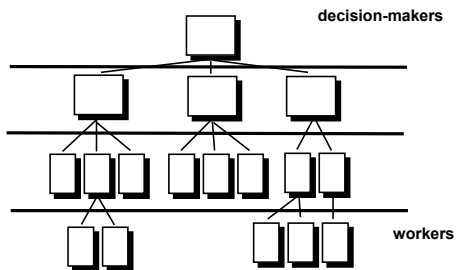
- design so that decision making and work are stratified
- decision making modules should reside at the top of the architecture

ศาสตราจารย์ ดร. ปิยะสิทธิ์ ๒๕๕๘

ศูนย์วิจัยระบบ ฐานข้อมูลและปัญญาประดิษฐ์

8

Vertical Partitioning



ศาสตราจารย์ ดร. ปิยะสิทธิ์ ๒๕๕๘

ศูนย์วิจัยระบบ ฐานข้อมูลและปัญญาประดิษฐ์

9

Design Quality

- Performance
- Availability
- Modifiability
- Safety
- Security
- etc.

Evaluating Architectural Design

- Software Architecture Analysis Method (SAAM)
- Architecture Tradeoff Analysis Method (ATAM)
- Active Design Reviews (ADR)
- Active Reviews for Intermediate Designs (ARID)

Software Architecture Analysis Method (SAAM)

- Three perspectives to understand and describe architectures
 - Functionality
 - Structure
 - Allocation

SAAM

1. Define functional partitioning
2. Allocate the functional to the components
3. Select the quality attributes
4. Develop quality tests
5. Assess the results

Architecture Trade-off Analysis Method (ATAM)

1. Collect scenarios
2. Gather requirements, constraints and environment
3. Describe architectural styles/patterns
4. Evaluate quality attributes
5. Identify sensitivity
6. Critique candidate architectures

Active Design Reviews (ADR)

- Evaluate detailed design of coherent units of software
- Quality and completeness
- Sufficiency, fitness, and suitability

Active Reviews for Intermediate Designs (ARID)

- Hybrid of ADR and ATAM
- 3 Groups of participants
 - ARID review team
 - Lead designer
 - Reviewers

ARID

1. Identify reviewers
2. Prepare design presentation
3. Prepare seed scenarios
4. Prepare for the review meeting
5. Present the ARID method
6. Present design
7. Brainstorm and prioritize scenarios
8. Perform review
9. Present conclusions
