

**Design Engineering 2**  
Suradet Jitprapaikulsarn

---

---

---

---

---

---

---

---

**Functional Independence**

**COHESION** - the degree to which a module performs one and only one function.

**COUPLING** - the degree to which a module is "connected" to other modules in the system.

Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6<sup>th</sup> Edition, McGraw-Hill, 2005  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ๒๕๕๘      สถาบันวิจัยระบบสารสนเทศ

2

---

---

---

---

---

---

---

---

**Law of Demeter**

- Discovered at Northeastern University in 1987
- "Only talk to your friends" motto
- A subsystem should have only limited knowledge of other subsystems

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ๒๕๕๘      สถาบันวิจัยระบบสารสนเทศ

3

---

---

---

---

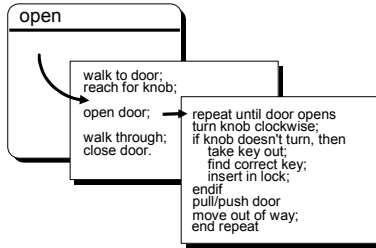
---

---

---

---

## Refinement



Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005  
 ភាសាកម្ពុជា • បច្ចេកវិទ្យា ២០០៥ ទំព័រទី២៧៤ គ្នានេក្នុងតារាង គ្មាន ទំព័រ២៧៤កន្លែង

4

---

---

---

---

---

---

---

---

## Refactoring

The process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure

Fowler, M. 1999

ភាសាកម្ពុជា • បច្ចេកវិទ្យា ២០០៥

ទំព័រទី២៧៤ គ្នានេក្នុងតារាង គ្មាន ទំព័រ២៧៤កន្លែង

5

---

---

---

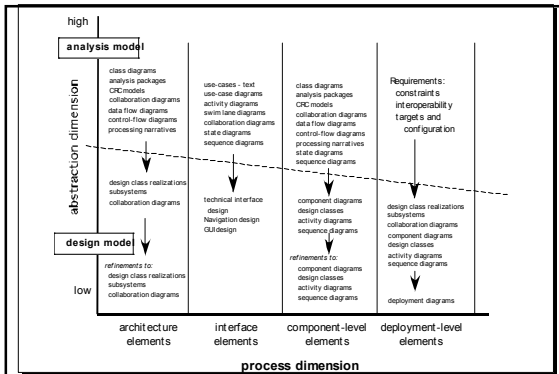
---

---

---

---

---



Derived from Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005  
 ភាសាកម្ពុជា • បច្ចេកវិទ្យា ២០០៥ ទំព័រទី២៧៤ គ្នានេក្នុងតារាង គ្មាន ទំព័រ២៧៤កន្លែង

6

---

---

---

---

---

---

---

---

## Design Model Elements

- **Data elements**
  - Data model -> data structures
  - Data model -> database architecture
- **Architectural elements**
  - Application domain
  - Analysis classes, their relationships, collaborations and behaviors are transformed into design realizations
  - Patterns and "styles" (Chapter 10)
- **Interface elements**
  - User interface (UI)
  - External interfaces to other systems, devices, networks or other producers or consumers of information
  - Internal interfaces between various design components.
- **Component elements**
- **Deployment elements**

สถาปัตยกรรมศาสตร์ • ปีการศึกษา ๒๕๕๗

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

7

---

---

---

---

---

---

---

---

## Designing Classes

- Well-formed design class
  - Complete and sufficient
  - Primitiveness
  - High cohesion
  - Low coupling

สถาปัตยกรรมศาสตร์ • ปีการศึกษา ๒๕๕๗

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

8

---

---

---

---

---

---

---

---

## A Class Design Methodology

- Identifying classes
- Identifying associations and attributes
- Identifying generalizations and interfaces
- Allocating responsibilities to classes
- Identifying operations
- Refine the model using design patterns

สถาปัตยกรรมศาสตร์ • ปีการศึกษา ๒๕๕๗

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

9

---

---

---

---

---

---

---

---

**Identifying Classes**

- Refine the analysis classes → Business domain classes
- Create a new set of design classes that support the solution
  - All abstractions necessary for HCI → User interface class
  - Lower-level business abstractions → Process classes
  - Data stores → Persistent classes
  - System management and control → System classes

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี      สถาบันวิจัยระบบสารสนเทศ      10

---

---

---

---

---

---

---

---

**Identifying associations and attributes**

- Start from a most central and important set of classes
- Determine the relationships to other classes
  - possess
  - control
  - is connected to
  - is related to
  - is a part of/has as parts
  - is a member of/has as members
- Avoid using associations to represent actions
- Promote similar attributes to classes

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี      สถาบันวิจัยระบบสารสนเทศ      11

---

---

---

---

---

---

---

---

**Identifying generalizations and interfaces**

- Bottom-up: groups together similar classes, creating a new superclass
- Top-down: divides up a complex class, creating new subclasses

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี      สถาบันวิจัยระบบสารสนเทศ      12

---

---

---

---

---

---

---

---

### Allocating responsibilities to classes

- Setting/Getting
- Creating and Initializing/Destroying
- Storing and Retrieving
- Adding/Deleting
- Transforming and Transmitting
- Computing
- Navigating and Searching
- Others

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
สถาบันวิจัยระบบสารสนเทศ
13

---

---

---

---

---

---

---

---

### Identifying Operations

- Realize responsibilities with operations
- There can be multiple operations for one responsibility but one will be in charge

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
สถาบันวิจัยระบบสารสนเทศ
14

---

---

---

---

---

---

---

---

### Design Modeling Principles

1. Traceability
2. Think about architecture
3. Pay equal attention to data and operation
4. Watch out the interfaces
5. Tailor UI to users
6. Loose coupling, high cohesion
7. Make the model easy to comprehend
8. Iterative design
9. Make it simple but not simpler

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
สถาบันวิจัยระบบสารสนเทศ
15

---

---

---

---

---

---

---

---