

## THIẾT KẾ VÀ XÂY DỰNG PHẦN MỀM

### Bài 5. Thiết kế giao diện

1

1

### Thiết kế giao diện

- ⇒ 1. Thiết kế giao diện đồ họa người dùng
- 2. Thiết kế giao diện hệ thống/thiết bị

2

### Tài liệu tham khảo

- [1] Textbook for Software Design & Development Engineers, No. 3 – *System Development, Operations and Maintenance, 2<sup>nd</sup> Edition*; Japan Information Processing Development Corporation, Japan Information-Technology Engineers Examination Center.

3

3

### 1. Thiết kế giao diện đồ họa người dùng

- ⇒ 1.1. Chuẩn hóa cấu hình màn hình
- 1.2. Tạo hình ảnh màn hình
- 1.3. Tạo biểu đồ chuyển tiếp màn hình
- 1.4. Tạo đặc tả màn hình

4

## Chuẩn hóa (Standardizing)

### ◆ Hiển thị (Display)

- Kích thước vật lý, độ phân giải, số lượng màu hỗ trợ

### ◆ Màn hình (Screen): chia thành các đối tượng được hiển thị, gọi là cửa sổ (Window)

- Vị trí của các nút chuẩn (v.d., OK, Cancel, Register, Search)
- Vị trí hiển thị của các thông điệp (message), v.v.
- Hiện thị tiêu đề màn hình và các menu
- Sự nhất quán trong việc biểu thị các ký tự chữ và số (alphanumeric characters)
- Biểu thị của câu văn và chi tiết các item
- Phối màu (color coordination)

## Chuẩn hóa (Standardizing)

### ❖ Điều khiển (control)

- Kiểu (style), kích thước, màu sắc, và các ký tự được hiển thị
- Quy trình kiểm tra đầu vào
- Trình tự di chuyển của tiêu điểm (sequence of moving the focus) (v.d., định nghĩa trình tự các tab)

### ❖ Menu

- Thiết kế các menu với sự cân nhắc về chuẩn chung của màn hình

### ❖ Dữ liệu đầu vào nhập từ bàn phím

- Duy trì sự nhất quán của các phím tắt giữa các màn hình

## Chuẩn hóa (Standardizing)

### ❖ Thông điệp (Messages)

- Xác định cách hiển thị thông điệp khi một tiến trình tốn thời gian được thực thi

### ❖ Lỗi (Error)

- Thực thi quy trình xử lý chuẩn khi có lỗi

### ❖ Hỗ trợ (Help)

- Phát triển thông tin hỗ trợ chi tiết theo hướng dẫn sử dụng, và duy trì sự nhất quán về các thuật ngữ, mô tả, và cách diễn giải các hàm.

## 1. Thiết kế giao diện đồ họa người dùng

### 1.1. Chuẩn hóa cấu hình màn hình

### ⇒ 1.2. Tạo hình ảnh màn hình

### 1.3. Tạo biểu đồ chuyển tiếp màn hình

### 1.4. Tạo đặc tả màn hình

## Từ use case

- ❖ Dựa vào use case và các lớp biên (boundary classes) mà tương tác với người dùng
  - Ảnh xạ các lớp biên này thành các màn hình
- ❖ Dựa vào mô tả đầu ra, đầu vào trong đặc tả use case

=> Thiết kế màn hình sử dụng các công cụ



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

9

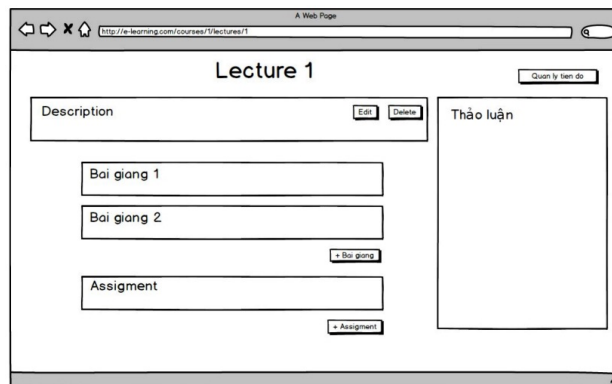
## Công cụ thiết kế GUI

- ❖ Công cụ đơn giản
  - Notepad
  - Microsoft Excel/Powerpoint/Word/FrontPage
- ❖ Công cụ chuyên nghiệp
  - Free
    - InVision
    - IDEs: Eclipse, NetBeans
  - Commercial
    - Adobe Dreamweaver
    - Axure RP
    - Photoshop
    - IDEs: Visual Studio



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

10



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

11

## 1. Thiết kế giao diện đồ họa người dùng

- 1.1. Chuẩn hóa cấu hình màn hình
- 1.2. Tạo hình ảnh màn hình
- 1.3. Tạo biểu đồ chuyển tiếp màn hình
- 1.4. Tạo đặc tả màn hình



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

12

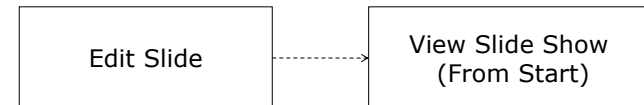
## Biểu đồ chuyển tiếp màn hình

- ❖ Tổng quan mối tương quan giữa các màn hình trong biểu đồ chuyển tiếp màn hình
  - Phân loại các màn hình thành 4 khuôn mẫu (patterns) bằng cách tập trung vào khuôn mẫu chuyển tiếp (transition pattern)
  - Liên kết các màn hình phù hợp với sự phân loại

13

## Bốn khuôn mẫu chuyển tiếp

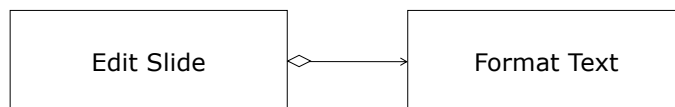
- ❖ 1. Chuyển tiếp màn hình đơn giản:
  - Chuyển tiếp đơn giản thông thường tới một màn hình độc lập



14

## Bốn khuôn mẫu chuyển tiếp (2)

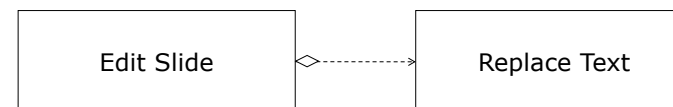
- ❖ 2. Chuyển tiếp tới một màn hình con phụ thuộc:
  - Chuyển tới một màn hình pop-up
  - Khi màn hình con được hiển thị trên màn hình cha, màn hình cha ở dưới không thể thao tác được



15

## Bốn khuôn mẫu chuyển tiếp (3)

- ❖ 3. Chuyển tiếp tới một màn hình con độc lập:
  - Chuyển tiếp tới một màn hình pop-up,
  - Màn hình cha và các màn hình khác có thể thao tác được khi màn hình con được hiển thị

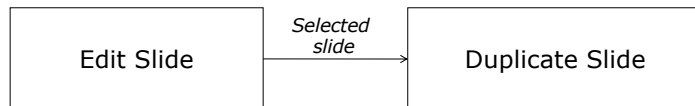


16

## Four transition patterns (4)

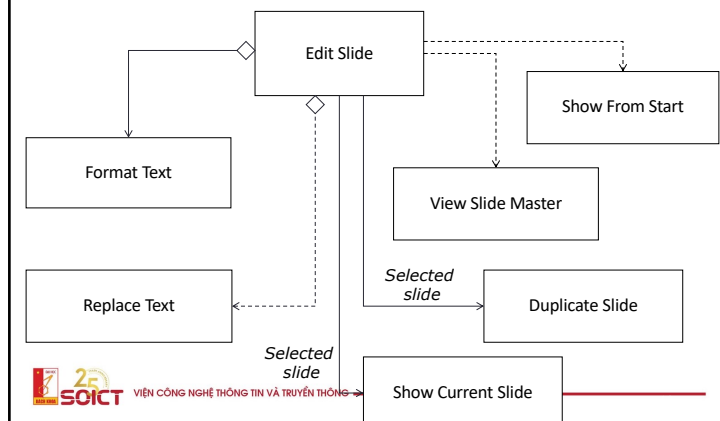
### ❖ 4. Chuyển tiếp tới một màn hình phụ thuộc:

- Bắt đầu một màn hình phụ thuộc với dữ liệu



17

## Kết nối màn hình: Biểu đồ chuyển tiếp màn hình



18

## 1. Thiết kế giao diện đồ họa người dùng

1.1. Chuẩn hóa cấu hình màn hình

1.2. Tạo hình ảnh màn hình

1.3. Tạo biểu đồ chuyển tiếp màn hình

⇒ 1.4. Tạo đặc tả màn hình

19

## Đặc tả màn hình (Screen specification)

- ❖ Xác định một định dạng (format) chi tiết cho đặc tả màn hình
- ❖ Định nghĩa thuộc tính (attributes) của các trường (field) dựa vào các thông tin màn hình (screen information) được xác định khi tạo ảnh màn hình và biểu đồ chuyển tiếp màn hình

20

## Đặc tả màn hình

### ❖ Ảnh màn hình (screen image)

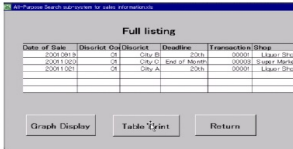
- Ảnh màn hình để hiển thị
- Nếu ảnh màn hình được tạo từ trước với công cụ thiết kế màn hình, đính kèm một bản cứng (hardcopy).

### ❖ Danh sách các tính năng (List of functions)

- Xác định tên của các thành phần như nút trên màn hình và tóm tắt tính năng của chúng.
- Cung cấp mô tả cho các sự kiện của từng màn hình, thuộc tính của từng phần, đặc tả của kiểm tra đầu vào và đầu ra, v.v.

### ❖ Định nghĩa thuộc tính của các trường

21

Liquor sales basic system (general-purpose search subsystem for sales information)		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Displaying detail table				
<b>Screen specification example</b>  		Control	Operation	Function	
		Area for displaying detail table	Initial	-Displays in a table information meeting the conditions defined in the search specification screen. -This follows the setting specified in the display settings screen for display items and sequence of display.	
		Graph display button	Click	Displays the graph display screen	
		Table print button	Click	Displays the print preview screen	
		Return button	Click	Displays the search specification screen	

[1]: Section 3.2.1, pp 3-54

22

## Định nghĩa thuộc tính của các trường

- ❖ Xác định thuộc tính của các trường đầu vào và đầu ra
- ❖ Khái quát chúng trong mô tả của các item trong màn hình hiển thị
- ❖ Màn hình gồm có nhiều trường
- ❖ Mỗi trường gồm một nhóm các thuộc tính (tương đương với một đặc tính) lúc ban đầu và một item có thể thay đổi (variable item)

23

## Example: Defining the field attributes

Screen name	Order entry		[1]	
Item name	Number of digits (bytes)	Type	Field attribute	Remarks
Transaction category	3	Numeral	Green (blink)	Error items blink.
Customer code	5	Numeral	Green (blink)	Error items blink.
Customer name	30	Character	White	15 characters, left-justified
Product code	8	Numeral	Green (blink)	Error items blink.
Product name	22	Character	White	11 characters, left-justified
Quantity	6	Numeral	Green (blink)	Error items blink.
Unit price	7	Numeral	White	
Amount	9	Numeral	White	
Quantity in stock	10	Numeral, special character	White	Displayed in the format of ZZZ, ZZZ, ZZ9

[1]: Section 3.2.1, pp 3-57

24

## Thiết kế giao diện

1. Thiết kế giao diện đồ họa người dùng

⇒ 2. Thiết kế giao diện hệ thống/thiết bị



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

25

## 2. Thiết kế giao diện hệ thống/thiết bị

⇒ 2.1. Xác định hệ thống con

2.2. Xác định giao diện hệ thống con

2.3. Thiết kế hệ thống con

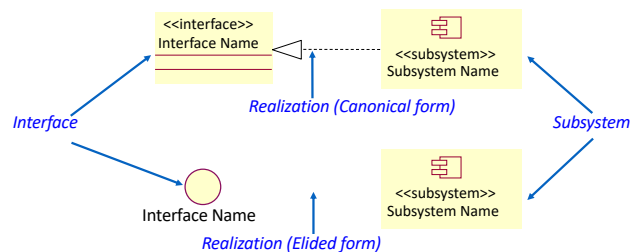


VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

26

## Hệ thống con và giao diện

❖ Tạo một hoặc nhiều giao diện mà định nghĩa các hành vi của hệ thống con (subsystem)



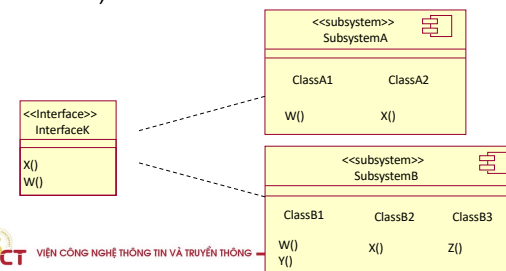
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

27

## Hệ thống con và giao diện (tiếp)

❖ Hệ thống con :

- Đóng gói hành vi hoàn toàn
- Đại diện cho khả năng độc lập với các giao diện rõ ràng (có tiềm năng để tái sử dụng)
- Mô hình hóa các biến thể thực thi (implementation variants)



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

28

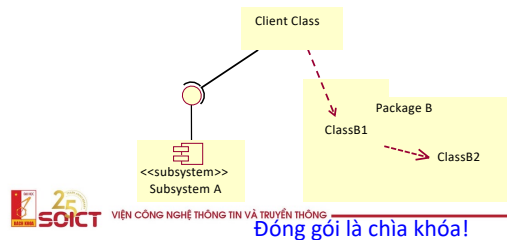
## Packages và Subsystems

### Subsystems

- Cung cấp hành vi
- Đóng gói hoàn toàn
- Dễ thay thế

### Packages

- Không cung cấp hành vi
- Không đóng gói hoàn toàn
- Có thể không dễ thay thế



29

## Sử dụng hệ thống con

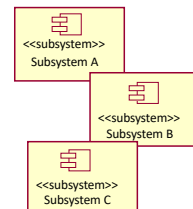
- ❖ Hệ thống con có thể dùng để chia hệ thống thành các phần có thể độc lập:
  - Được gọi (ordered), cấu hình (configured), hoặc cung cấp (delivered)
  - Được phát triển (developed), miễn là các giao diện không đổi
  - Được triển khai (deployed) giữa tập hợp các nút tính toán phân tán
  - Được thay đổi (changed) mà không ảnh hưởng các phần khác của hệ thống
- ❖ Hệ thống con còn được dùng để:
  - Phân vùng hệ thống thành các đơn vị mà có thể cung cấp các giới hạn về bảo mật (restricted security) đối với các tài nguyên quan trọng
  - Đại diện các sản phẩm đã tồn tại hoặc các hệ thống bên ngoài (v.d. components)

Các hệ thống con tang mức trừu tượng

30

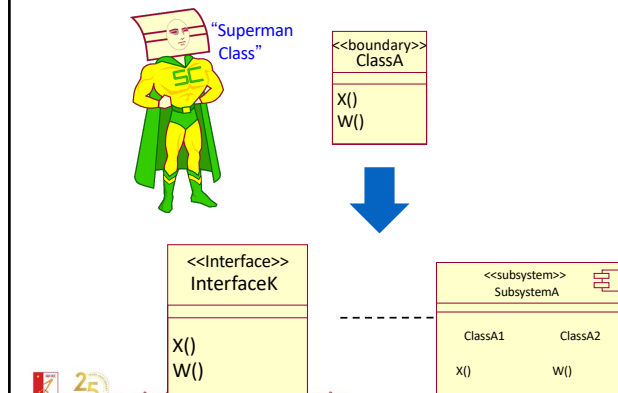
## Ứng viên cho hệ thống con

- ❖ Các lớp phân tích (analysis classes) có thể được “tiến hóa” (evolve) thành các hệ thống con:
  - Các lớp cung cấp các dịch vụ (services) phức tạp và/hoặc các tiện ích
  - Các lớp biên (boundary classes) (giao diện người dùng và giao diện với hệ thống bên ngoài)
- ❖ Các sản phẩm đã tồn tại hoặc các hệ thống bên ngoài (e.g., components):
  - Phần mềm tượng tác
  - Hỗ trợ truy cập cơ sở dữ liệu
  - Kiểu và cấu trúc dữ liệu
  - Các tiện ích chung
  - Sản phẩm có ứng dụng cụ thể



32

## Xác định các hệ thống con



33



## 2. Thiết kế giao diện hệ thống/thiết bị

### 2.1. Xác định hệ thống con

### 2.2. Xác định giao diện hệ thống con

### 2.3. Thiết kế hệ thống con

## Xác định giao diện

### ❖ Mục đích

- Để xác định giao diện của hệ thống con, cần dựa trên trách nhiệm của chúng

### ❖ Các bước tiến hành

- Xác định một tập hợp các ứng viên cho các giao diện của tất cả các hệ thống con
- Tìm điểm chung giữa các giao diện
- Tìm các phụ thuộc của giao diện
- Ánh xạ các giao diện tới các hệ thống con
- Định nghĩa hành vi được xác định bởi các giao diện
- Gói (package) các giao diện này

Stable, well-defined interfaces are key to a stable, resilient architecture.

## Interface Guidelines

- ❖ Tên giao diện (interface name)
  - Phản ánh vai trò trong hệ thống
- ❖ Mô tả giao diện (interface description)
  - Bao quát các trách nhiệm
- ❖ Định nghĩa hoạt động (operation definition)
  - Tên gọi cần phản ánh kết quả của hoạt động
  - Mô tả hoạt động làm gì, các tham số và kết quả
- ❖ Tài liệu giao diện (interface documentation)
  - Gói hỗ trợ thông tin: biểu đồ trình tự, biểu đồ trạng thái, kế hoạch kiểm thử(test plans), v.v.



## Example: Design Subsystems and Interfaces

### Analysis

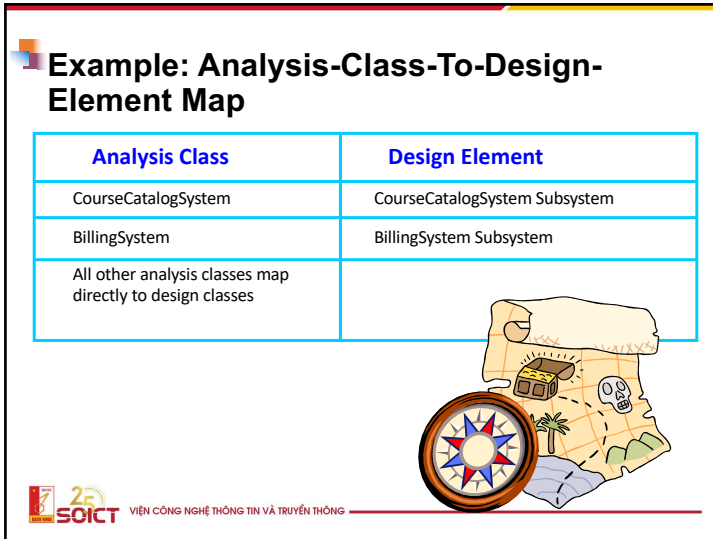
```
<<boundary>>  
BillingSystem  
  
//submit bill()
```

```
<<boundary>>  
CourseCatalogSystem  
  
//get course offerings()
```

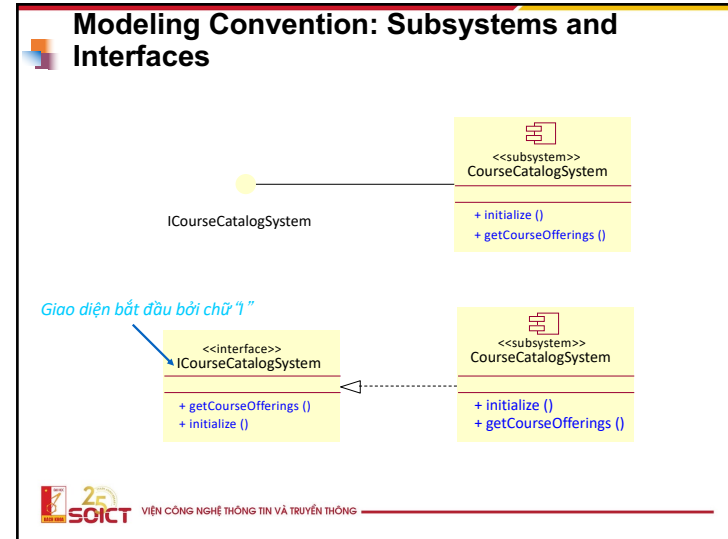
### Design

```
<<subsystem>>  
Billing System  
  
IBillingSystem  
submitBill(forTuition : Double, forStudent : Student)
```

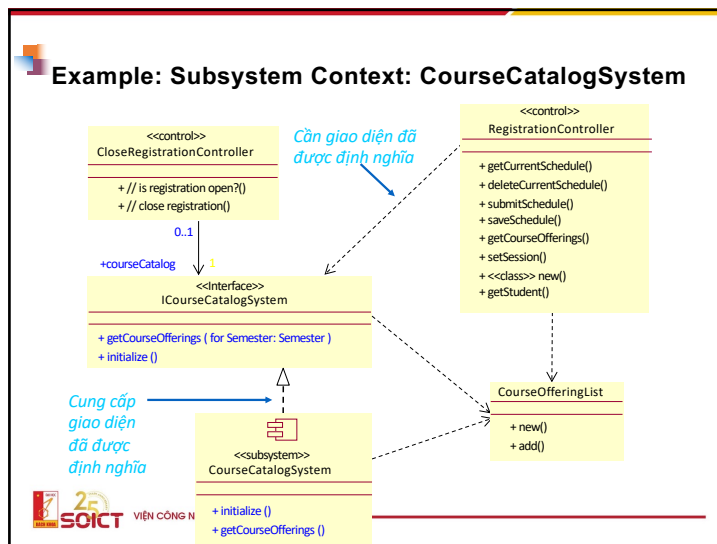
```
<<subsystem>>  
Course Catalog System  
  
ICourseCatalogSystem  
getCourseOfferings(forSemester : Semester, forStudent : Student) : CourseOfferingList  
initialize()
```



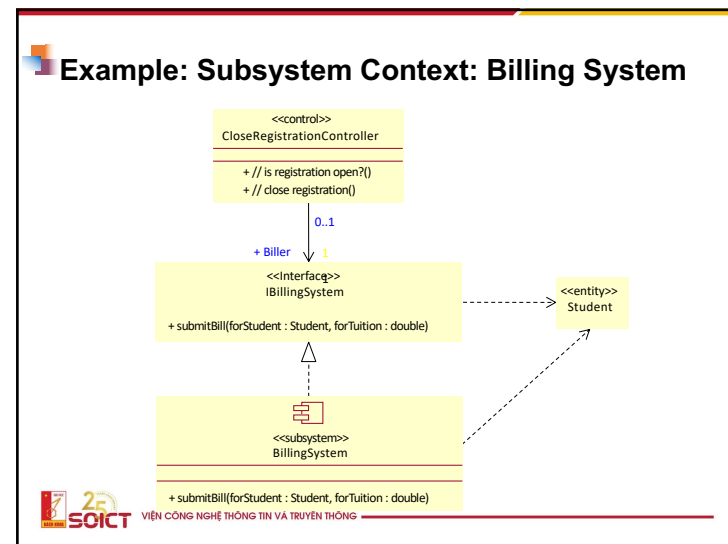
38



39



40



41

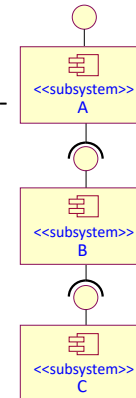
## 2. Thiết kế giao diện hệ thống/thiết bị

- 2.1. Xác định hệ thống con
- 2.2. Xác định giao diện hệ thống con
- ⇒ 2.3. Thiết kế hệ thống con

42

## Subsystem Guidelines

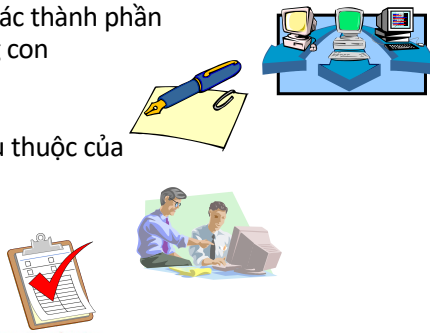
- ❖ Mục tiêu
  - Giảm ràng buộc (Loose coupling)
  - Di động (portability), sự tương thích cắm-và-chạy (plug-and-play compatibility)
  - Tránh khỏi thay đổi (Insulation from change)
  - Tiến hóa độc lập (Independent evolution)
- ❖ Gợi ý mạnh mẽ (strong suggestions)
  - Không để lộ chi tiết, chỉ giao diện
  - Chỉ phụ thuộc vào các giao diện khác



43

## Các bước thiết kế hệ thống con

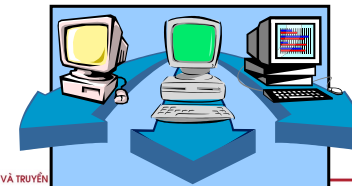
- ❖ Phân chia các hành vi của hệ thống con tới các thành phần của hệ thống
- ❖ Viết tài liệu các thành phần của hệ thống con
- ❖ Mô tả sự phụ thuộc của hệ thống con
- ❖ Checkpoints



44

## Các bước thiết kế hệ thống con

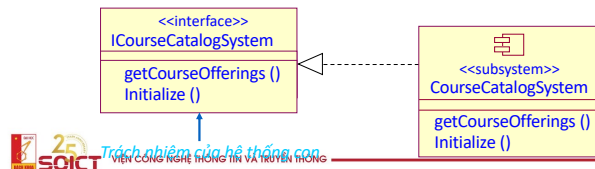
- ★ ❖ Phân chia các hành vi của hệ thống con tới các thành phần của hệ thống
- ❖ Viết tài liệu các thành phần của hệ thống con
- ❖ Mô tả sự phụ thuộc của hệ thống con
- ❖ Checkpoints



45

## Các trách nhiệm của hệ thống con

- ❖ Các trách nhiệm của hệ thống con được định nghĩa bởi các hoạt động (operations) của giao diện
  - Mô hình hóa các hiện thực giao diện (interface realizations)
- ❖ Giao diện có thể được hiện thực bởi
  - Hành vi của lớp bên trong
  - Hành vi của hệ thống con



46

## Phân phối trách nhiệm của hệ thống con

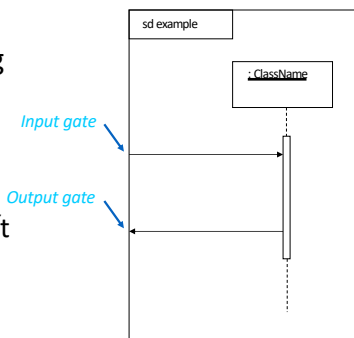
- ❖ Xác định các thành phần mới, hoặc tái sử dụng các thành phần thiết kế (v.d., lớp và/hoặc hệ thống con)
- ❖ Phân bổ các trách nhiệm của hệ thống con cho các thành phần thiết kế
- ❖ Kết hợp các cơ chế có thể áp dụng được (v.d., persistence, distribution)
- ❖ Viết tài liệu về sự phối hợp giữa các thành phần thiết kế (design element collaborations) trong các hiện thực của giao diện ("interface realizations")
  - Một biểu đồ tương tác hoặc nhiều hơn cho mỗi hoạt động của giao diện (interface operation)
  - (Các) biểu đồ lớp chứa mối quan hệ giữa các thành phần thiết kế cần thiết
- ❖ Xem lại "Identify Design Elements"
  - Chính sửa các lớp biên (boundaries) và sự phụ thuộc của hệ thống con nếu cần



47

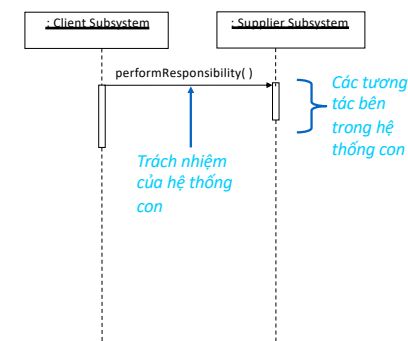
## What Are Gates?

- ❖ Một điểm kết nối trong một sự tương tác của một thông điệp mà đến từ bên ngoài hoặc đi từ bên trong ra ngoài
  - Một điểm trên cạnh của biểu đồ trình tự
  - Tên của thông điệp tiếp kết nối chính là tên của cổng (gate)



48

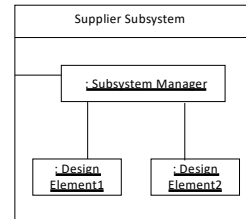
## Subsystem Interaction Diagrams



49

## Cấu trúc bên trong của Supplier Subsystem

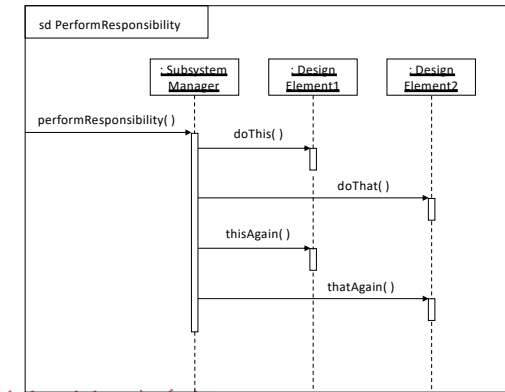
- ❖ Subsystem Manager điều phối hành vi bên trong của hệ thống con.



- ❖ Hành vi hoàn thiện của hệ thống con được phân bổ giữa các lớp thành phần thiết kế bên trong (the internal Design Element classes)

50

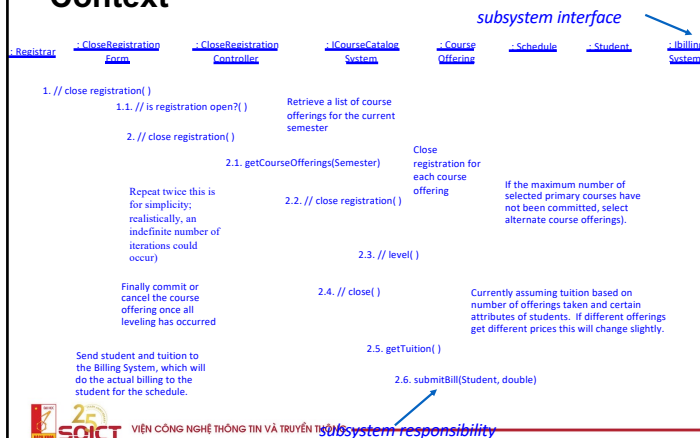
## Modeling Convention: Internal Subsystem Interaction



Độc nhìn hộp trắng (white box view) của Supplier Subsystem

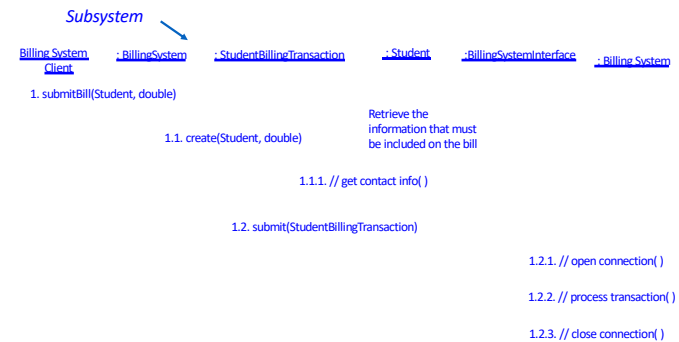
51

## Example: Billing System Subsystem In Context



52

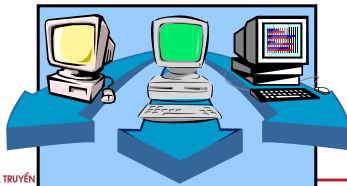
## Example: Local BillingSystem Subsystem Interaction



53

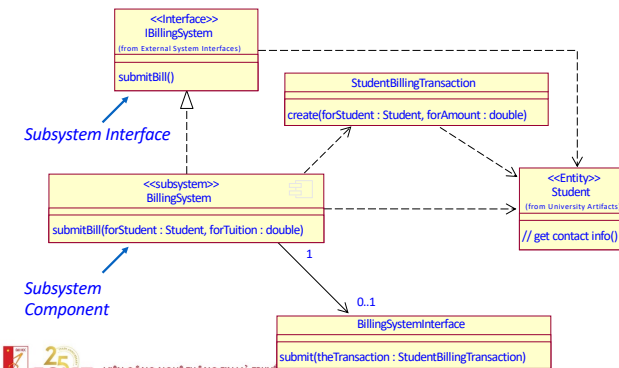
## Các bước thiết kế hệ thống con

- ❖ Phân chia các hành vi của hệ thống con tới các thành phần của hệ thống
- ★ ❖ Viết tài liệu các thành phần của hệ thống con
- ❖ Mô tả sự phụ thuộc của hệ thống con
- ❖ Checkpoints



54

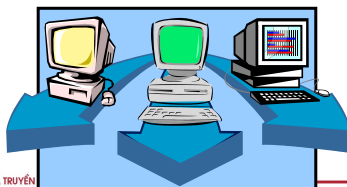
## Example: Billing System Subsystem Elements



55

## Các bước thiết kế hệ thống con

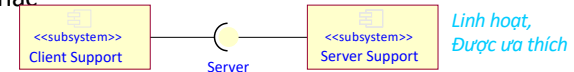
- ❖ Phân chia các hành vi của hệ thống con tới các thành phần của hệ thống
- ❖ Viết tài liệu các thành phần của hệ thống con
- ★ ❖ Mô tả sự phụ thuộc của hệ thống con
- ❖ Checkpoints



56

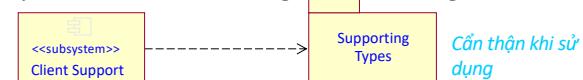
## Subsystem Dependencies: Guidelines

- ❖ Sự phụ thuộc của hệ thống con vào một hệ thống con khác



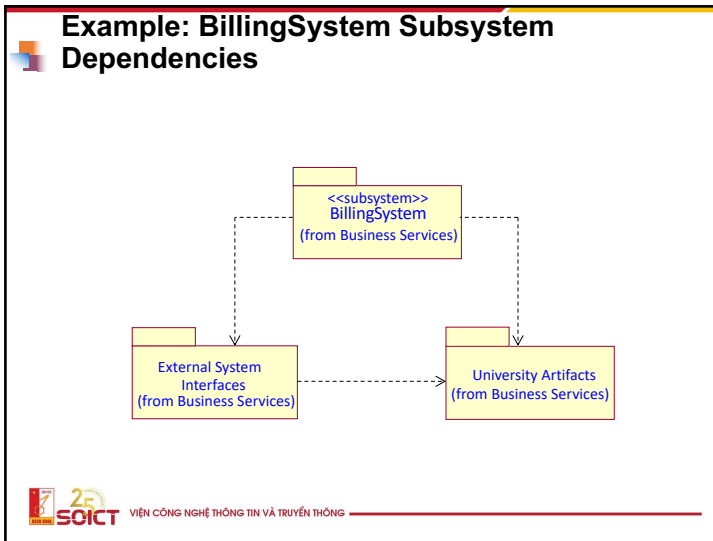
Linh hoạt,  
Được ưa thích

- ❖ Sự phụ thuộc của hệ thống con vào một gói



Cẩn thận khi sử dụng

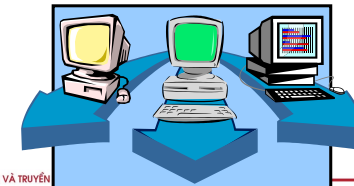
57



58

### Các bước thiết kế hệ thống con

- ❖ Phân chia các hành vi của hệ thống con tới các thành phần của hệ thống
- ❖ Viết tài liệu các thành phần của hệ thống con
- ❖ Mô tả sự phụ thuộc của hệ thống con
- ★ ❖ Checkpoints



SOICT VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

59

### Checkpoints: Design Subsystems

- ❖ Một liên kết hiện thực liệu có được định nghĩa cho mỗi giao diện mà được cung cấp bởi hệ thống con không (Is a realization association defined for each interface offered by the subsystem) ?
- ❖ Một liên kết phụ thuộc liệu có được định nghĩa cho mỗi giao diện mà được sử dụng bởi hệ thống con không (Is a dependency association defined for each interface used by the subsystem) ?
- ❖ Bạn có chắc rằng không có thành phần nào trong hệ thống con bị công khai (Are you sure that none of the elements within the subsystem have public visibility)?



SOICT VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

60

### Checkpoints: Design Subsystems

- ❖ Mỗi hoạt động trong một giao diện liệu đã được hiện thực hóa bởi hệ thống con mà được viết thành tài liệu trong một biểu đồ tương tác? Nếu chưa, hoạt động đó liệu có được hiện thực hóa bởi một lớp duy nhất để có thể thấy ánh xạ 1-1 giữa hoạt động của lớp với hoạt động của giao diện? (Is each operation on an interface realized by the subsystem documented in a interaction diagram? If not, is the operation realized by a single class, so that it is easy to see that there is a simple 1:1 mapping between the class operation and the interface operation?)



SOICT VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

61

## Review: Subsystem Design

- ❖ Mục đích của thiết kế hệ thống con (Subsystem Design) là gì?
- ❖ Cổng (Gates) là gì?
- ❖ Tại sao sự phụ thuộc vào một hệ thống con cần phải ở giao diện (Why should dependencies on a subsystem be on the subsystem interface) ?



## Question?

