IT4490 - SOFTWARE DESIGN AND CONSTRUCTION
6. INTERFACE DESIGN

1

Interface design

1. Graphical user interface design
2. System/Device interface design

2

References

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

3

1. Graphical user interface design

1.1. Standardizing the screen configuration
1.2. Creating screen images
1.3. Creating a screen transition diagram
1.4. Creating screen specifications

4

Standardizing

◆**Display**
- Physical size, resolution, and number of colors supported by displays

◆**Screen:** divided into displayed objects called windows (Window)
- Location of standard buttons (e.g., OK, Cancel, Register, Search)
- Display location of messages, etc.
- Display of screen title and menus
- Consistency in expression of alphanumeric characters
- Expression of sentences and detailed items
- Color coordination

5

Standardizing

- **Control**
  - Style, size, color, and characters displayed
  - Input check process
  - Sequence of moving the focus (e.g., defining the tab sequence)
- **Menu**
  - Design menus with consideration of the standard specification (common client area) of the screen
- **Direct input from a keyboard**
  - Maintain consistency in the assignment of shortcut keys

6

Standardizing

- **Messages**
  - Determine how messages are displayed when a time-consuming process is executed (busy).
- **Error**
  - Execute standardized processing if an error occurs
- **Help**
  - Develop detailed Help information in accordance with the manual, and maintain consistency in terminology, descriptions, and explanations of methods.

7

1. Graphical user interface design

1.1. Standardizing the screen configuration

1.2. Creating screen images

1.3. Creating a screen transition diagram

1.4. Creating screen specifications

8

From use case

- Based on use case and boundary classes which interact with users
  - Map these boundary classes to screens
- Based on input/output description in use case specification/scenario
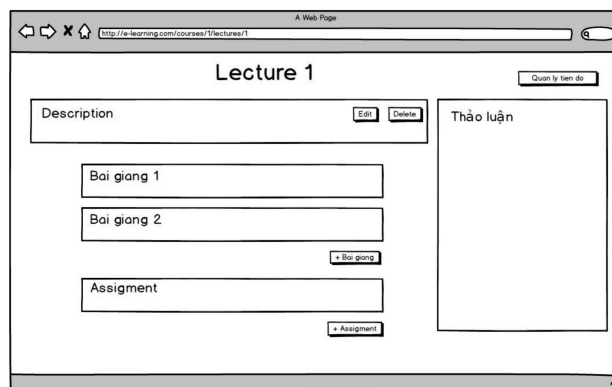
=> Design screen using tools

GUI Design tools

- Simple tools
  - Notepad
  - Microsoft Excel/Powerpoint/Word/FrontPage
- Professional tools
  - Free
    - InVision
    - IDEs: Eclipse, NetBean
  - Commercial
    - Adobe Dreamweaver
    - Axure RP
    - Photoshop
    - IDEs: Visual Studio

1. Graphical user interface design

1.1. Standardizing the screen configuration

1.2. Creating screen images

1.3. Creating a screen transition diagram

1.4. Creating screen specifications

Display transition diagram

- Summarize the correlation of screens in the screen transition diagram
  - Classify the screens into the four patterns by focusing on the transition pattern
  - Link the screens in accordance with the classifications

13

Four transition patterns

- ◆ *1. Simple screen transition:*
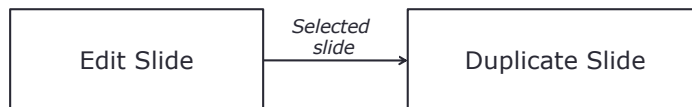 - A conventional simple transition to an independent screen

| Edit Slide | ----→ | View Slide Show (From Start) |
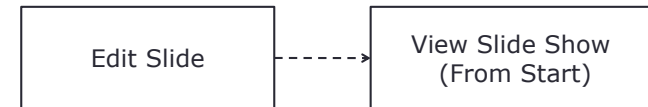
14

Four transition patterns (4)

- *2. Transition to a dependent screen:*
  - Start a dependent new screen with data

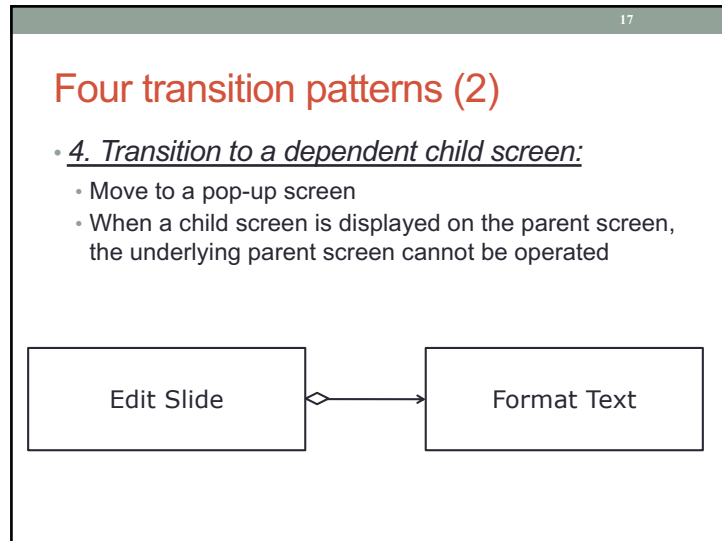| Edit Slide | — *Selected slide* → | Duplicate Slide |

15

Four transition patterns (3)

- *3. Transition to an independent child screen:*
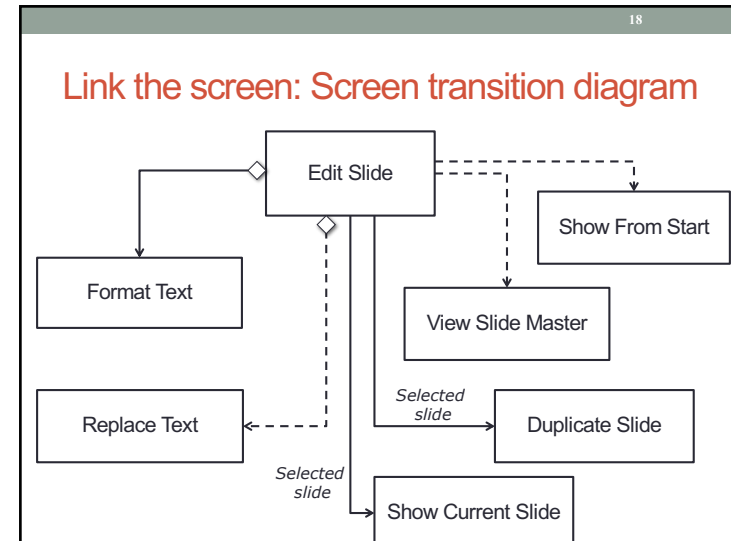 - Move to a pop-up screen,
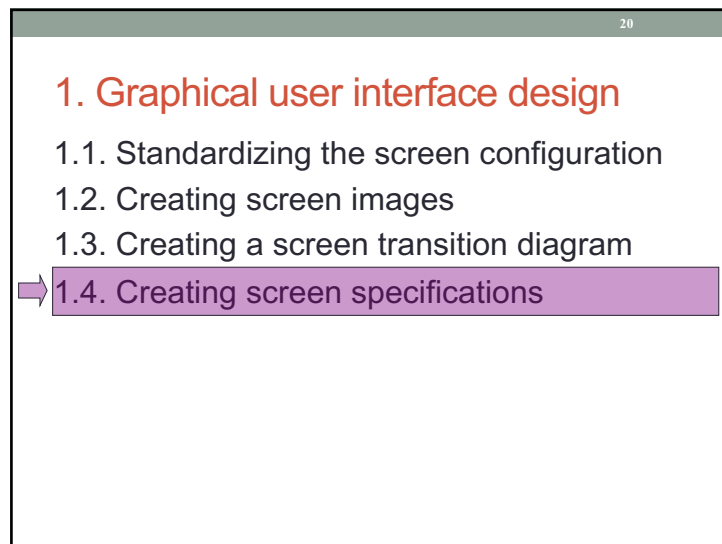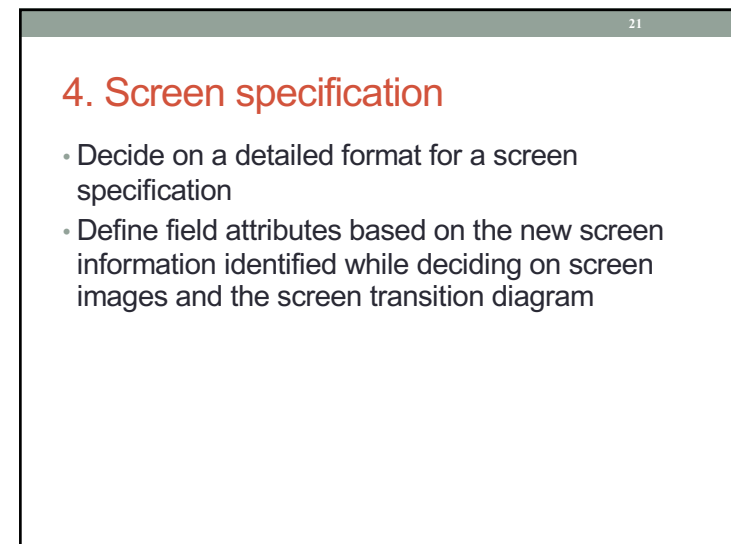 - Parent screen and other screens can be operated while the child screen is displayed.

| Edit Slide | ◇----→ | Replace Text |

16

4

Four transition patterns (2)

- *4. Transition to a dependent child screen:*
  - Move to a pop-up screen
  - When a child screen is displayed on the parent screen, the underlying parent screen cannot be operated

17

Link the screen: Screen transition diagram

18

1. Graphical user interface design

1.1. Standardizing the screen configuration

1.2. Creating screen images

1.3. Creating a screen transition diagram

⇨ 1.4. Creating screen specifications

20

4. Screen specification

- Decide on a detailed format for a screen specification
- Define field attributes based on the new screen information identified while deciding on screen images and the screen transition diagram
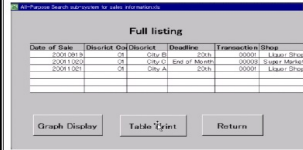
21

Slide 22

Screen specification

- Screen image
  - This is the screen image to be displayed. If screen images are created in advance with the screen design tool, attach a hardcopy.
- List of functions
  - Defines the names of parts such as the buttons on the screen, and summarizes their functions.
  - Provide descriptions of events for individual screens, attributes of parts, input check specifications and output specifications, etc.
- Defining the field attributes

22

Slide 23

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				

Screen specification example

Full listing

	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

Graph Display | Table Print | Return

23

Slide 24

Defining the field attributes

- Decide on the field attributes of input and output items
- Summarize them in descriptions of items for screen display.
- The screen consists of multiple fields.
- Each field consists of a one-byte (equivalent to a single character) attribute at the beginning and a variable item

24

Slide 25

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

25

6

Interface design

1. Graphical user interface design
2. System/Device interface design

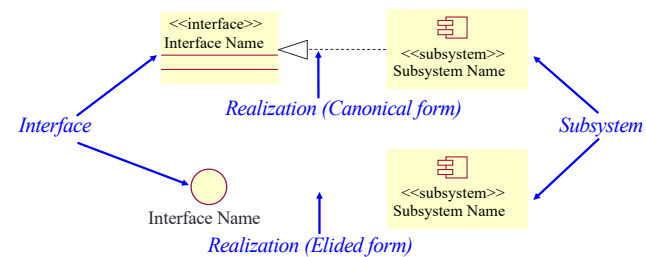# 2. System/Device interface design

2.1. Identify subsystem
2.2. Identify subsystem interfaces
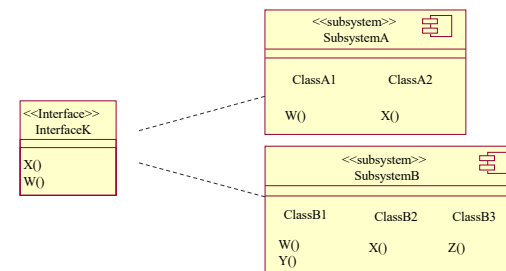2.3. Subsystem design

Review: Subsystems and Interfaces

- Realizes one or more interfaces that define its behavior

Subsystems and Interfaces (continued)

- Subsystems :
  - Completely encapsulate behavior
  - Represent an independent capability with clear interfaces (potential for reuse)
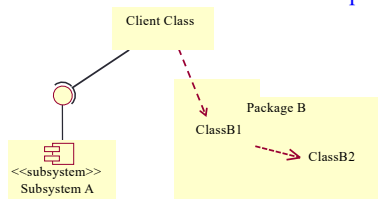 - Model multiple implementation variants

Packages versus Subsystems

Subsystems
- Provide behavior
- Completely encapsulate their contents
- Are easily replaced

Packages
- Don't provide behavior
- Don't completely encapsulate their contents
- May not be easily replaced

Client Class

Package B

ClassB1

ClassB2

<<subsystem>>
Subsystem A

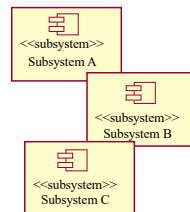Encapsulation is the key!

30

Subsystem Usage

- Subsystems can be used to partition the system into parts that can be independently:
  - ordered, configured, or delivered
  - developed, as long as the interfaces remain unchanged
  - deployed across a set of distributed computational nodes
  - changed without breaking other parts of the systems
- Subsystems can also be used to:
  - partition the system into units which can provide restricted security over key resources
  - represent existing products or external systems in the design (e.g. components)

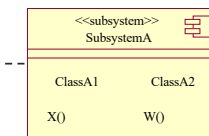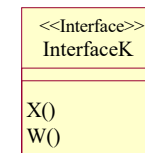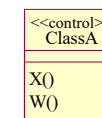 Subsystems raise the level of abstraction.

31

Candidate Subsystems

- Analysis classes which may evolve into subsystems:
  - Classes providing complex services and/or utilities
  - Boundary classes (user interfaces and external system interfaces)
- Existing products or external systems in the design (e.g., components):
  - Communication software
  - Database access support
  - Types and data structures
  - Common utilities
  - Application-specific products

<<subsystem>>
Subsystem A

<<subsystem>>
Subsystem B

<<subsystem>>
Subsystem C

33

Identifying Subsystems

"Superman Class"

<<control>>
ClassA

X()
W()

<<Interface>>
InterfaceK

X()
W()

<<subsystem>>
SubsystemA

ClassA1   ClassA2

X()   W()

34

8

2. System/Device interface design

2.1. Identify subsystem

2.2. Identify subsystem interfaces

2.3. Subsystem design

35

Identifying Interfaces

- Purpose
  - To identify the interfaces of the subsystems based on their responsibilities
- Steps
  - Identify a set of candidate interfaces for all subsystems.
  - Look for similarities between interfaces.
  - Define interface dependencies.
  - Map the interfaces to subsystems.
  - Define the behavior specified by the interfaces.
  - Package the interfaces.

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

36

Interface Guidelines

- Interface name
  - Reflects role in system
- Interface description
  - Conveys responsibilities
- Operation definition
  - Name should reflect operation result
  - Describes what operation does, all parameters and result
- Interface documentation
  - Package supporting info: sequence and state diagrams, test plans, etc.
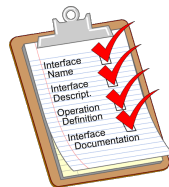
37

Example: Design Subsystems and Interfaces

Analysis   *Design*

<<boundary>>
BillingSystem

//submit bill()

<<subsystem>>
BillingSystem

IBillingSystem
submitBill(forTuition : Double, forStudent : Student)

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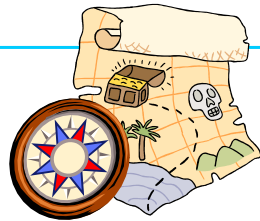
//get course offerings()

<<subsystem>>
CourseCatalogSystem

ICourseCatalogSystem
getCourseOfferings(forSemester : Semester, forStudent : Student) : CourseOfferingList
initialize()

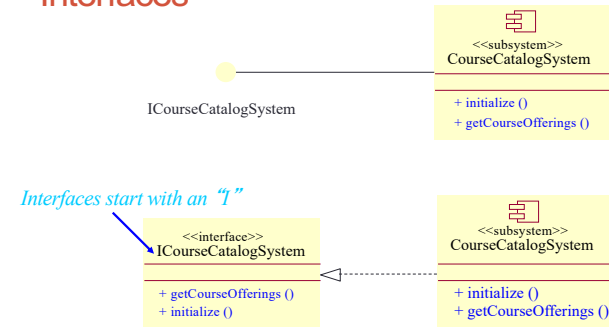All other analysis classes map directly to design classes.

38

Example: Analysis-Class-To-Design-Element Map

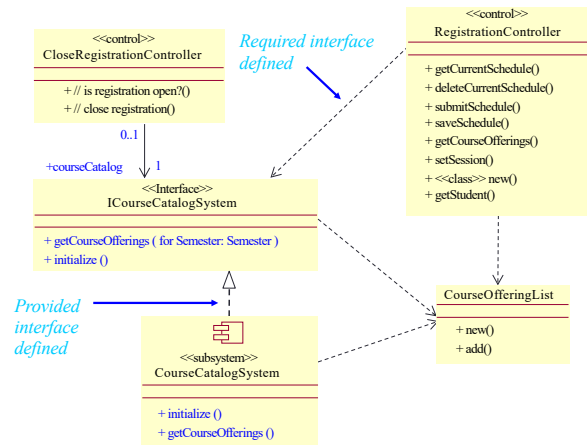| Analysis Class	Design Element
CourseCatalogSystem	CourseCatalogSystem Subsystem
BillingSystem	BillingSystem Subsystem
All other analysis classes map directly to design classes	

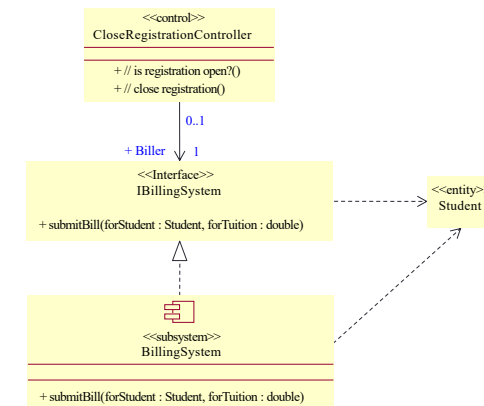---

Modeling Convention: Subsystems and Interfaces

ICourseCatalogSystem

<<subsystem>>
CourseCatalogSystem
+ initialize ()
+ getCourseOfferings ()

Interfaces start with an "I"

<<interface>>
ICourseCatalogSystem
+ getCourseOfferings ()
+ initialize ()

<<subsystem>>
CourseCatalogSystem
+ initialize ()
+ getCourseOfferings ()

Example: Subsystem Context: CourseCatalogSystem

<<control>>
CloseRegistrationController
+ // is registration open?()
+ // close registration()

Required interface defined

<<control>>
RegistrationController
+ getCurrentSchedule()
+ deleteCurrentSchedule()
+ submitSchedule()
+ saveSchedule()
+ getCourseOfferings()
+ setSession()
+ <<class>> new()
+ getStudent()

0..1
+courseCatalog   1

<<Interface>>
ICourseCatalogSystem
+ getCourseOfferings (for Semester: Semester)
+ initialize ()

Provided interface defined

<<subsystem>>
CourseCatalogSystem
+ initialize ()
+ getCourseOfferings ()

CourseOfferingList
+ new()
+ add()

Example: Subsystem Context: Billing System

<<control>>
CloseRegistrationController
+ // is registration open?()
+ // close registration()

0..1
+ Biller   1

<<Interface>>
IBillingSystem
+ submitBill(forStudent : Student, forTuition : double)

<<entity>>
Student

<<subsystem>>
BillingSystem
+ submitBill(forStudent : Student, forTuition : double)

2. System/Device interface design

2.1. Identify subsystem

2.2. Identify subsystem interfaces

2.3. Subsystem design

43

Subsytem design

• Refer to 12-Subsystem Design (IBM).pdf

44

Question?

45

Bài tập tuần

• Tất cả các use case
  • Tìm các lớp phân tích cho từng use case
  • Gom nhóm thành các package, giải thích lý do
  • Vẽ biểu đồ lớp (không cần thuộc tính, hành vi) cho từng package, vẽ biểu đồ phụ thuộc giữa các package
    • Chú ý với các use case có người phụ trách cần có hành vi cho lớp phân tích
  • Vẽ biểu đồ dịch chuyển màn hình của các lớp GUI

• Với 2 use case của cá nhân
  • Biểu đồ tương tác (1 trong 2 hoặc cả hai loại: Trình tự / Giao tiếp)
  • Biểu đồ lớp phân tích cho từng use case (có hành vi)
  • Thiết kế, đặc tả màn hình cho các lớp GUI

46