

Trường Đại Học Thủy Lợi
Bộ Môn Công Nghệ Phần Mềm – Khoa CNTT

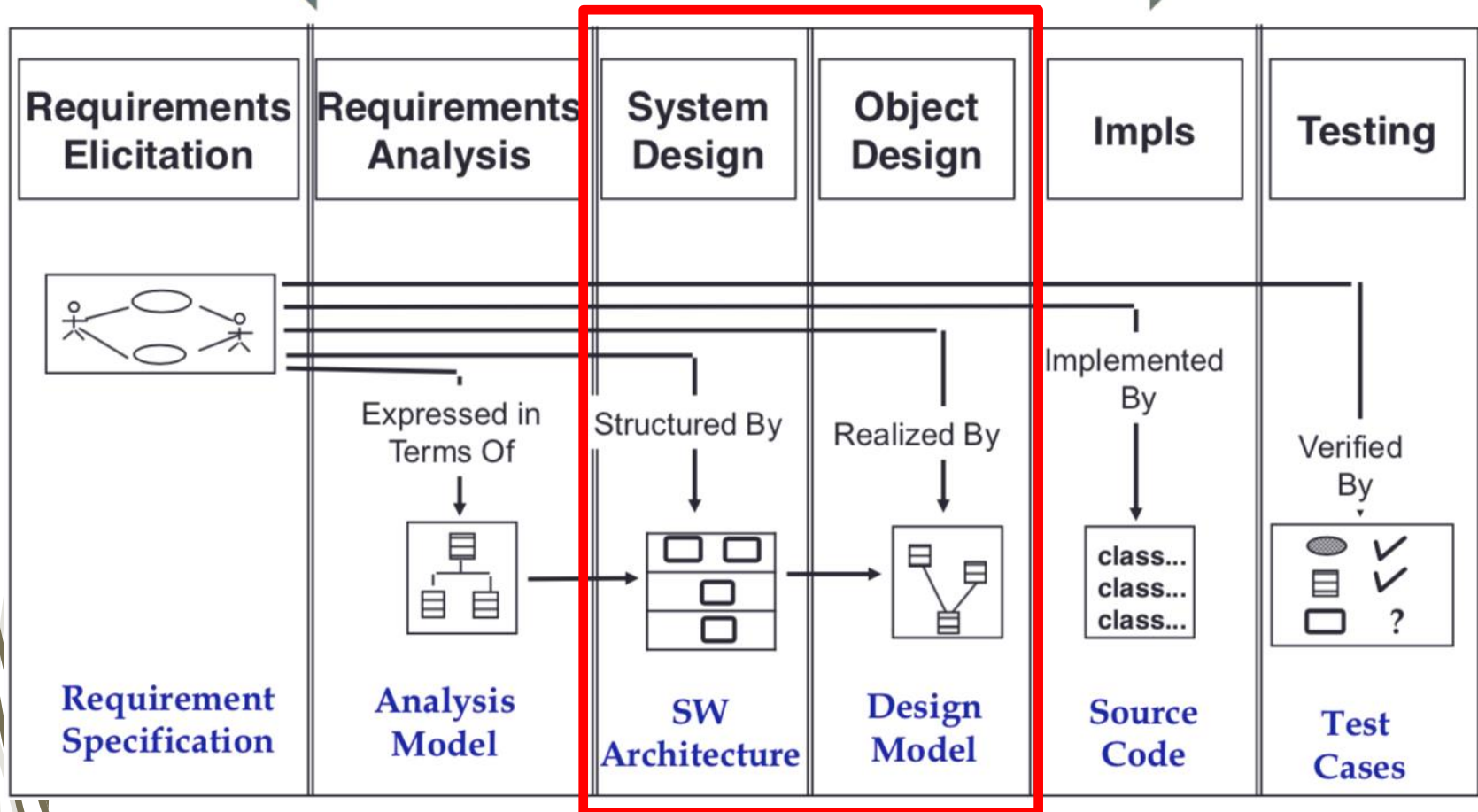
CÔNG NGHỆ PHẦN MỀM

BÀI 3
THIẾT KẾ PHẦN MỀM

Giảng viên: ThS. Nguyễn Thị Thu Hương
Email: huongnt@tlu.edu.vn

Problem

Solution



Nội dung

1. Thiết kế kiến trúc
2. Thiết kế hướng đối tượng
3. Thiết kế dữ liệu
4. Thiết kế giao diện



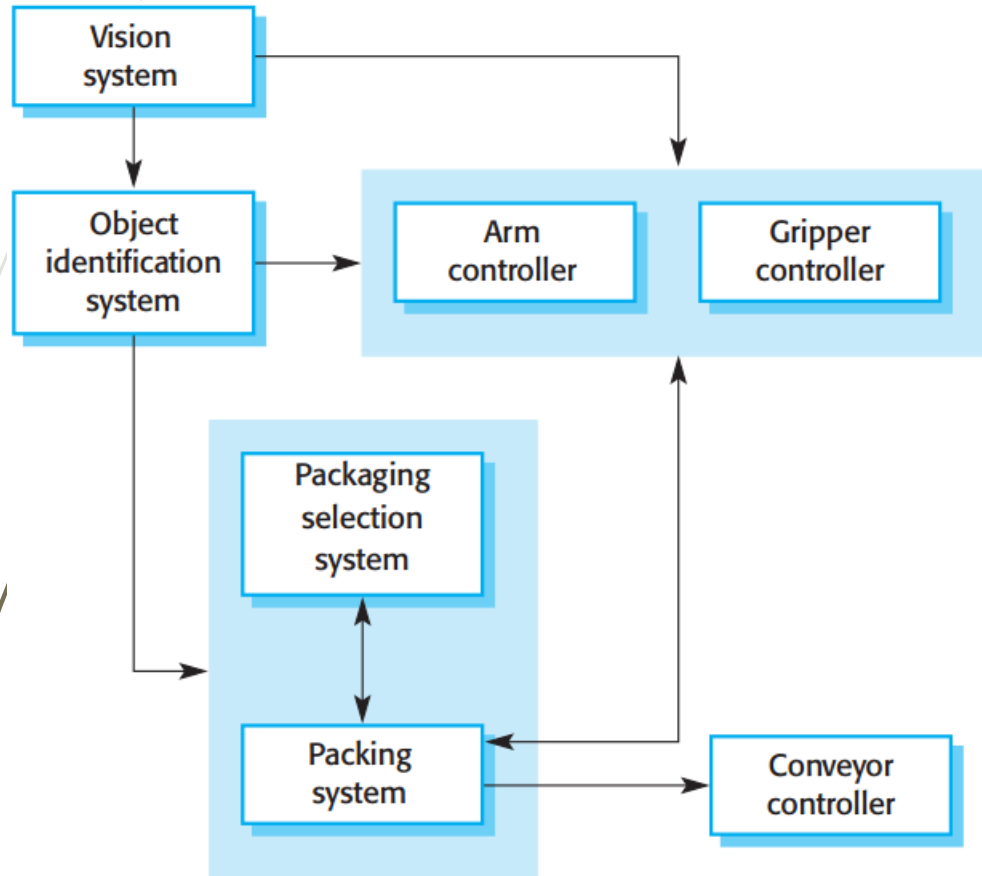
4

Phần 1. Thiết kế kiến trúc

Architectural design

Giới thiệu

- Thiết kế kiến trúc phần mềm là giai đoạn đầu của quá trình thiết kế hệ thống
- Là quá trình xác định:
 1. **những hệ thống con** hoặc **mô-đun** và
 2. **một khung làm việc** giúp kiểm soát và giao tiếp giữa các hệ thống con/mô-đun này.
- Đầu ra của quá trình thiết kế này là *một tài liệu mô tả về kiến trúc phần mềm.*



Applications (Ứng dụng)

Trang chính

Danh sách
liên lạc

Điện thoại

Trình duyệt

...

Applications Framework (Khung tổ chức ứng dụng)

Trình quản lý
ActivityTrình quản lý
WindowTrình quản lý
Content ProvidersTrình quản lý
View SystemTrình quản lý
đóng góiTrình quản lý
thoạiTrình quản lý
tài nguyênTrình quản lý
vị tríTrình quản lý
thông báo

Libraries (Thư viện)

Surface
ManagerMedia
Framework

SQLite

OpenGL | ES

FreeType

Webkit

SGL

SSL

libc

Android Runtime (Thời gian chạy Android)

Thư viện lỗi

Máy ảo
Dalvik

Linux Kernel (Nhân Linux)

Trình điều khiển
màn hìnhTrình điều khiển
cameraTrình điều khiển
bộ nhớ FlashTrình điều khiển
IPCTrình điều khiển
bàn phímTrình điều khiển
Wi-FiTrình điều khiển
âm thanhTrình điều khiển
nguồn điện

CÁC PHONG CÁCH THIẾT KẾ KIẾN TRÚC

Các phong cách thiết kế kiến trúc

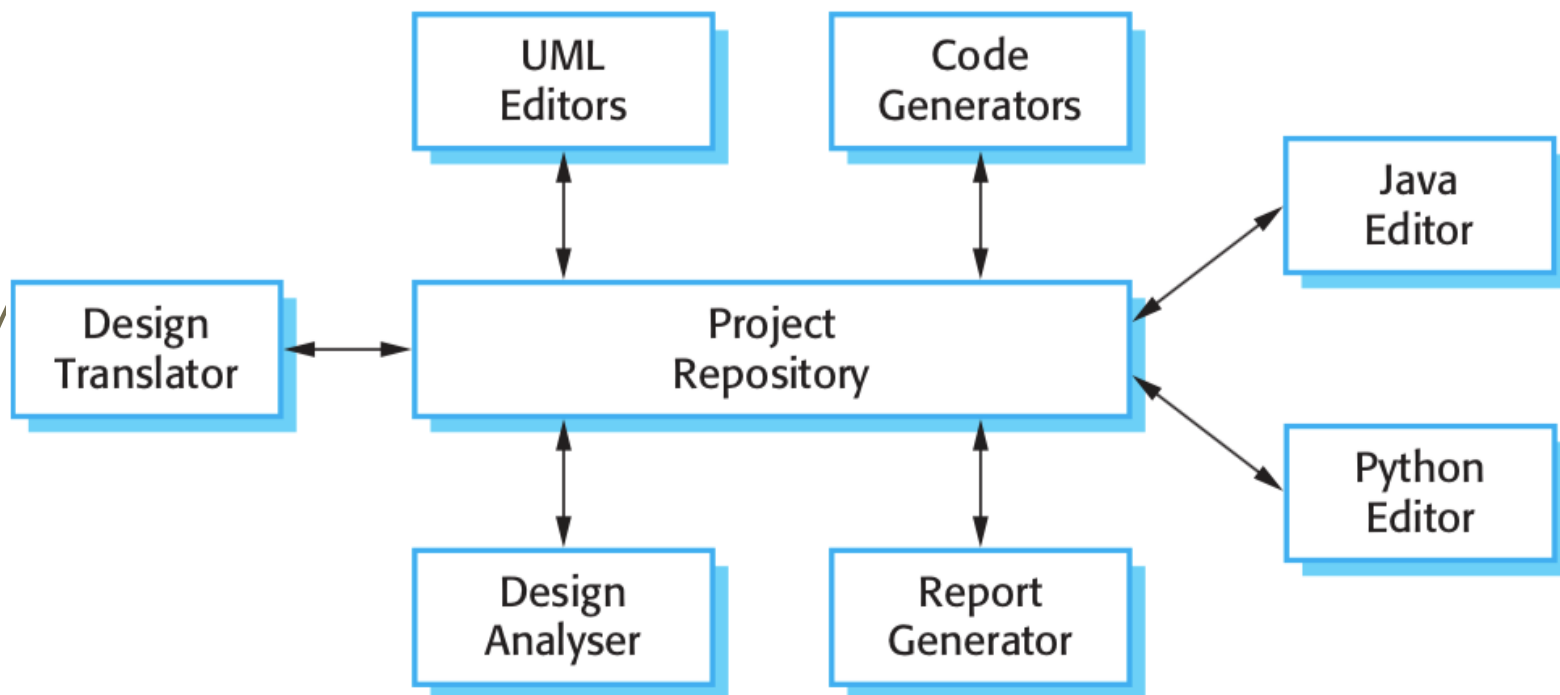
1. Phong cách tổ chức
2. Phong cách chia nhỏ mô-đun
3. Phong cách điều khiển

Phong cách tổ chức

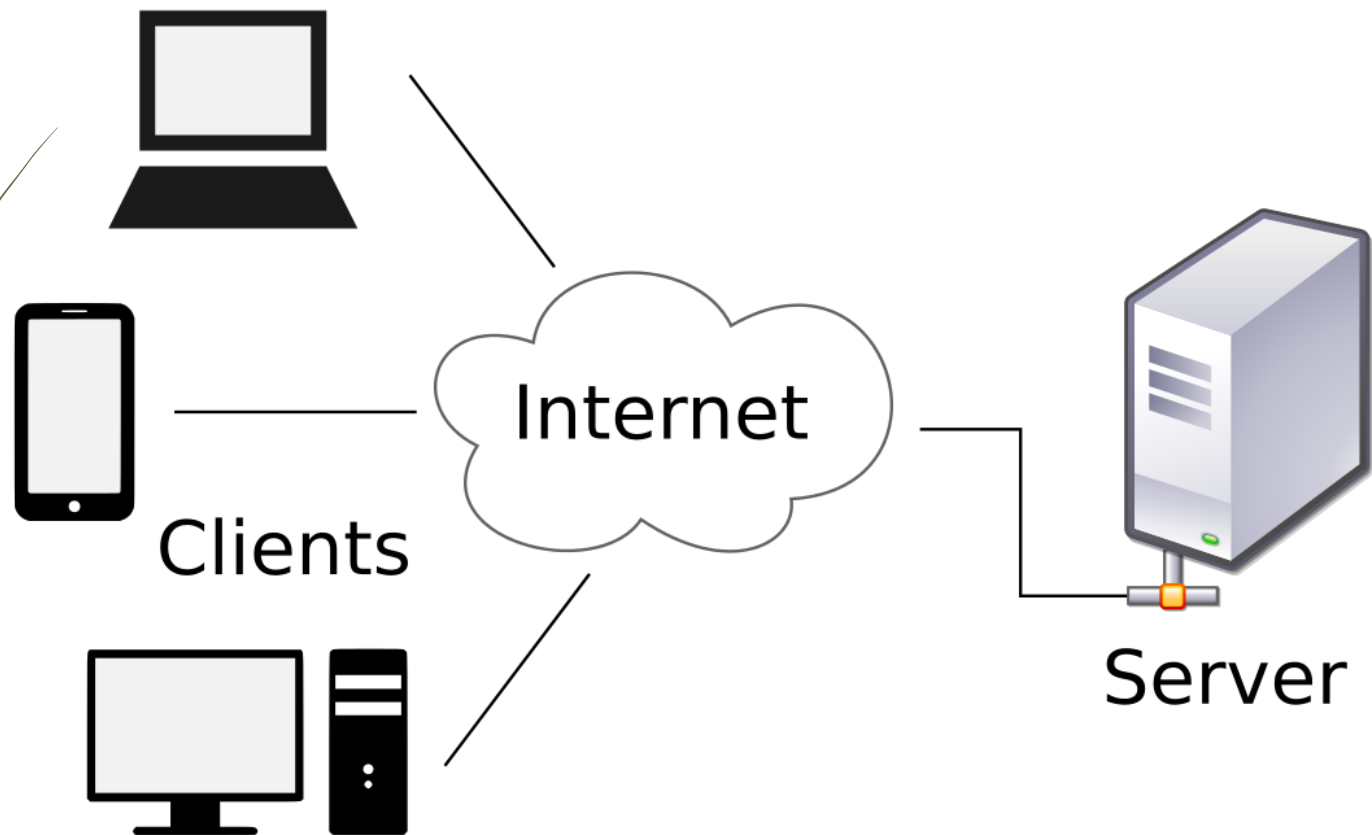
➡ Một số mô hình tổ chức:

1. Mô hình kho lưu trữ chia sẻ
2. Mô hình máy chủ - máy khách
3. Mô hình P2P
4. Mô hình phân tầng

M1-1: Mô hình kho lưu trữ chia sẻ

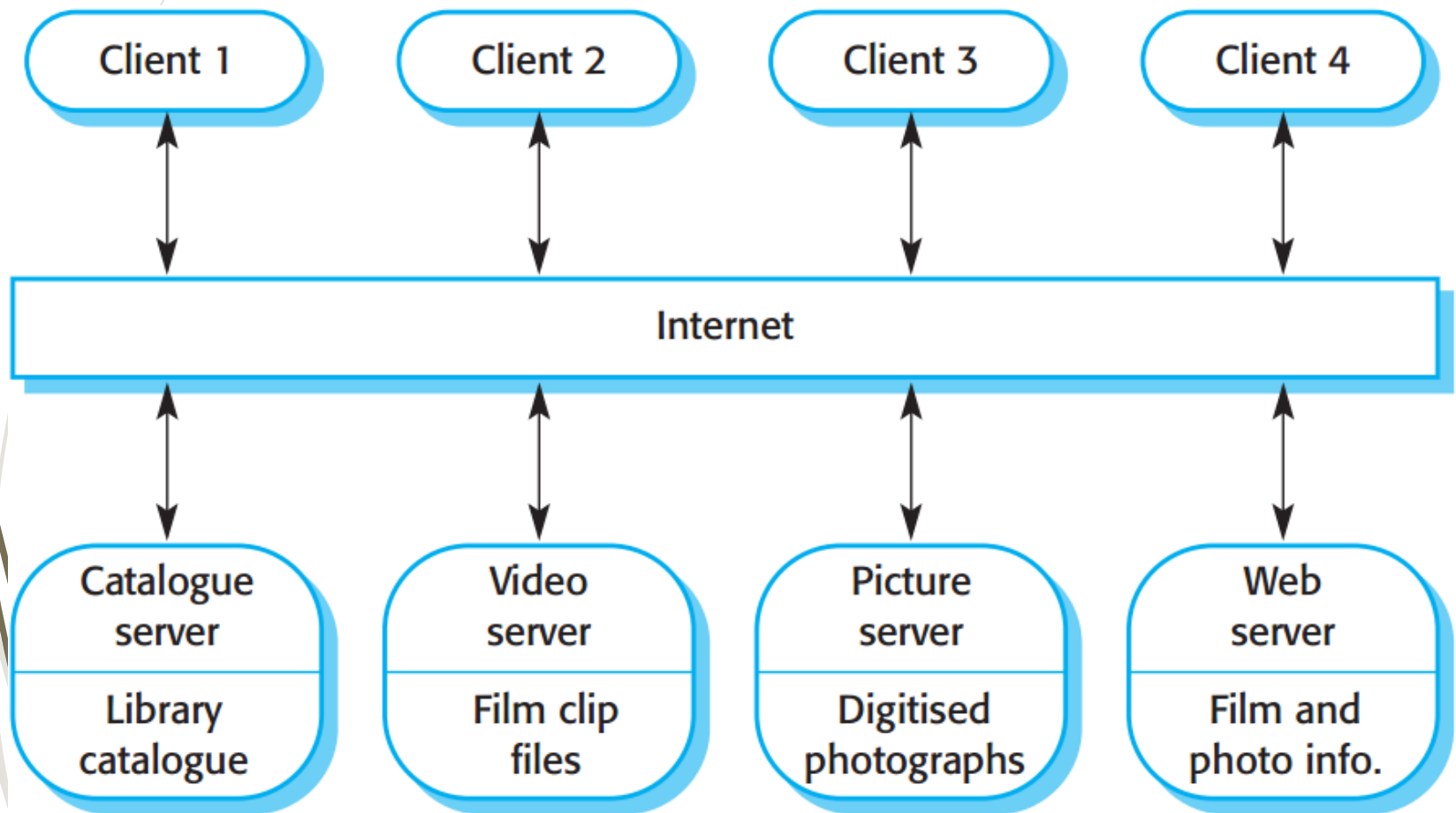


M1-2: Mô hình Máy khách - Máy chủ

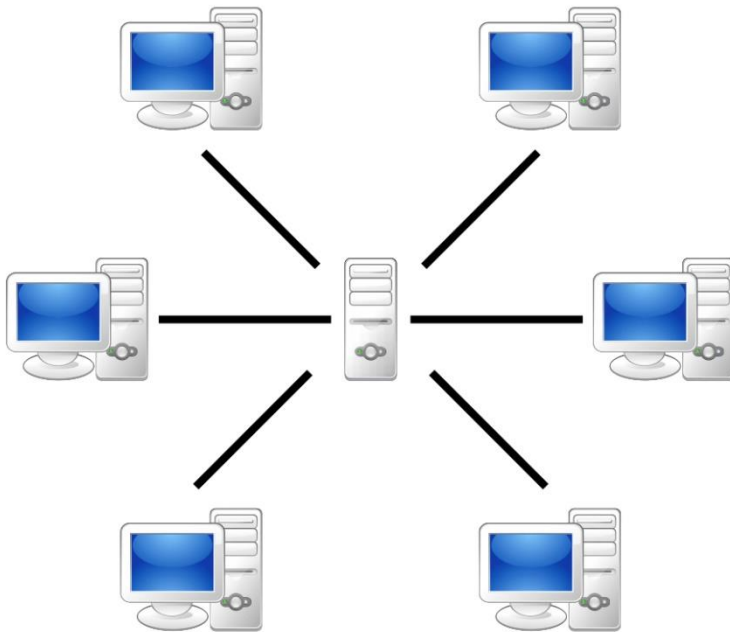


13

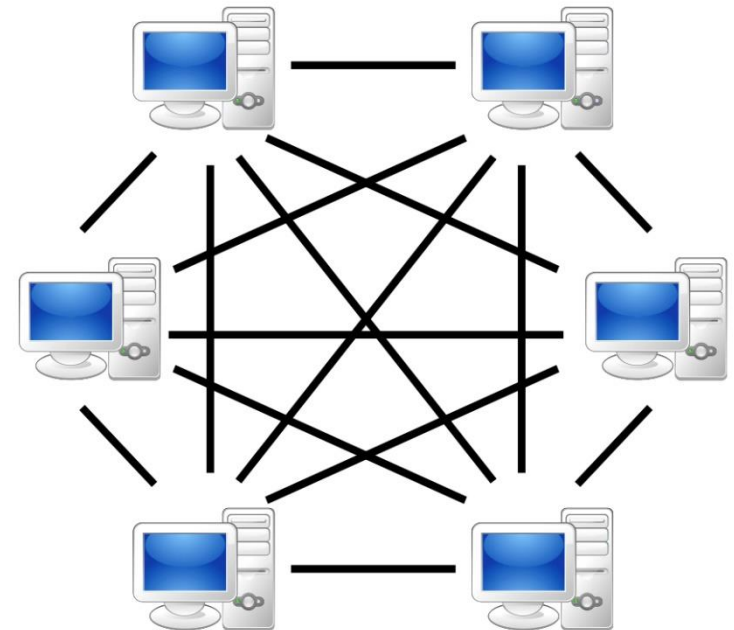
Ví dụ: Film & picture library



M1-3. Mô hình Mạng ngang hàng (Peer-to-Peer hay P2P)



Server-based



P2P-network

M1-4: Mô hình phân tầng

- Tổ chức hệ thống thành một tập các tầng
 - Mỗi tầng cung cấp một tập các dịch vụ cho tầng phía trên.
- Khi một tầng giao diện thay đổi, chỉ có các tầng lân cận bị ảnh hưởng.

Kiến trúc phân tầng tổng quát

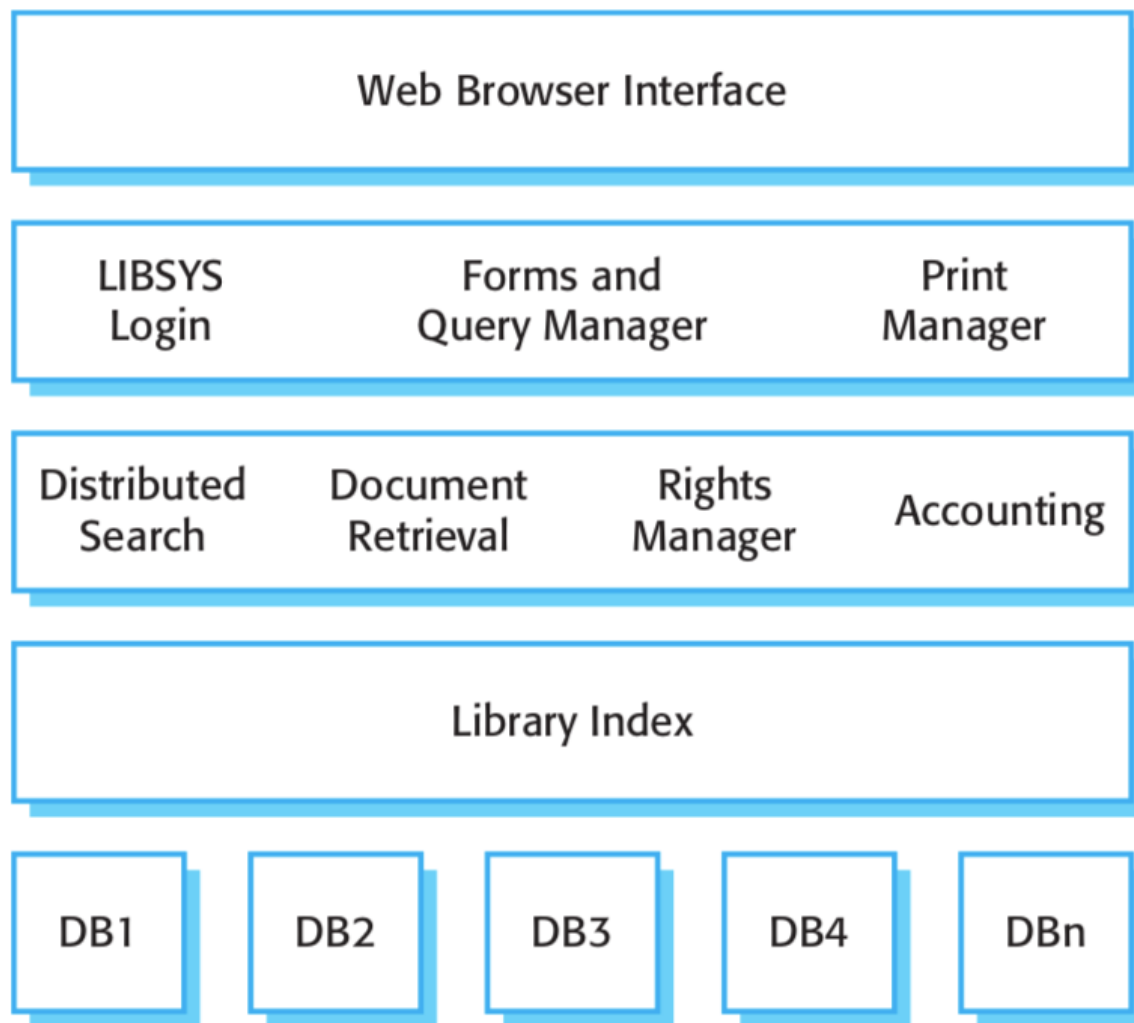
User Interface

User Interface Management
Authentication and Authorization

Core Business Logic/Application Functionality
System Utilities

System Support (OS, Database etc.)

Kiến trúc phân tầng của LIBSYS



2. Phong cách chia nhỏ mô-đun (1)

Hệ thống con

- Là một hệ thống có quyền riêng của nó
 - Hoạt động độc lập với các dịch vụ được cung cấp bởi các hệ thống con khác.

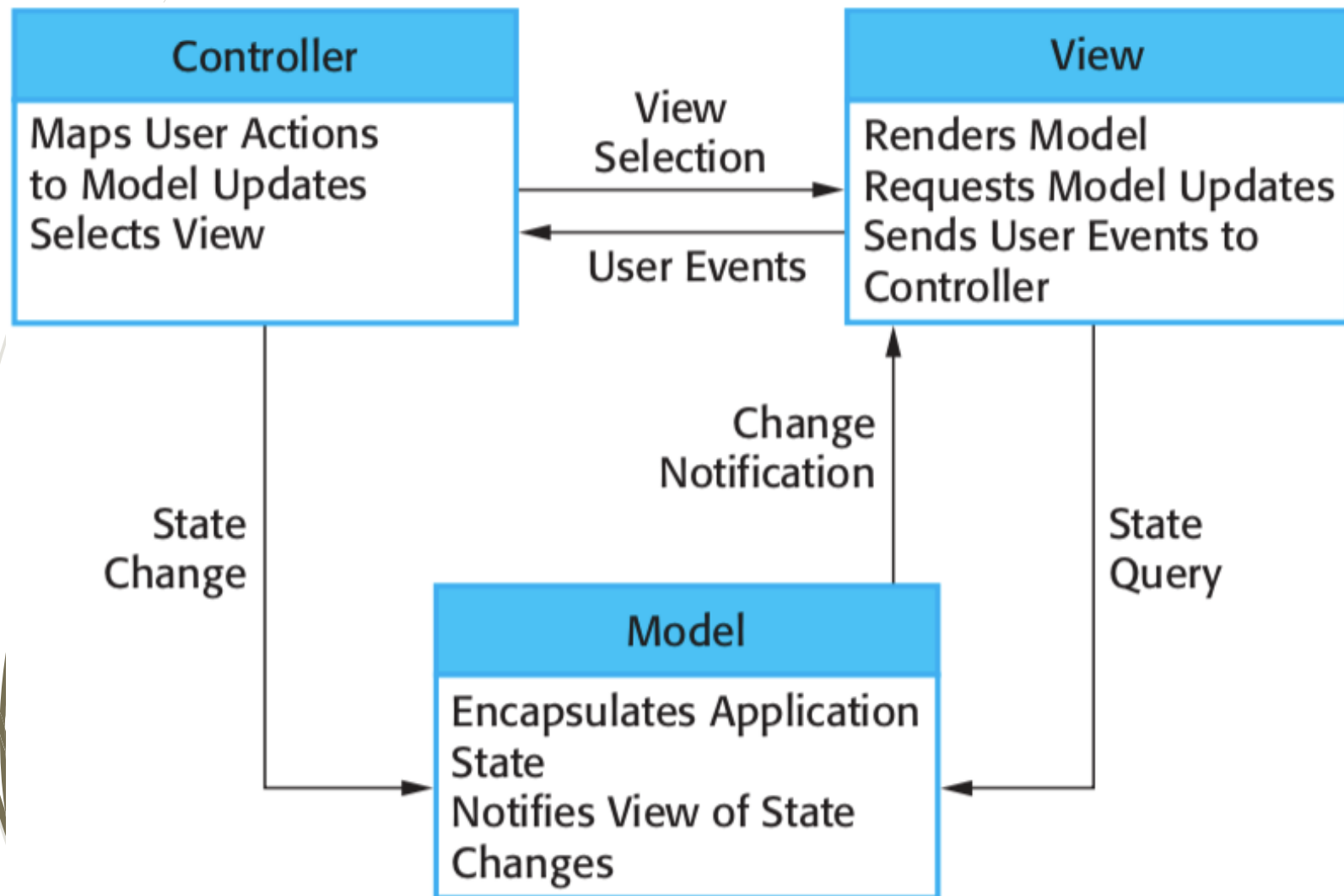
Mô-đun

- Là một thành phần hệ thống cung cấp các dịch vụ cho các thành phần khác
 - Thường sẽ không được xem như một hệ thống riêng biệt.

2. Phong cách chia nhỏ mô-đun (2)

- ➡ Một số mô hình chia nhỏ mô-đun:
 1. Mô hình MVC
 2. Mô hình đường ống

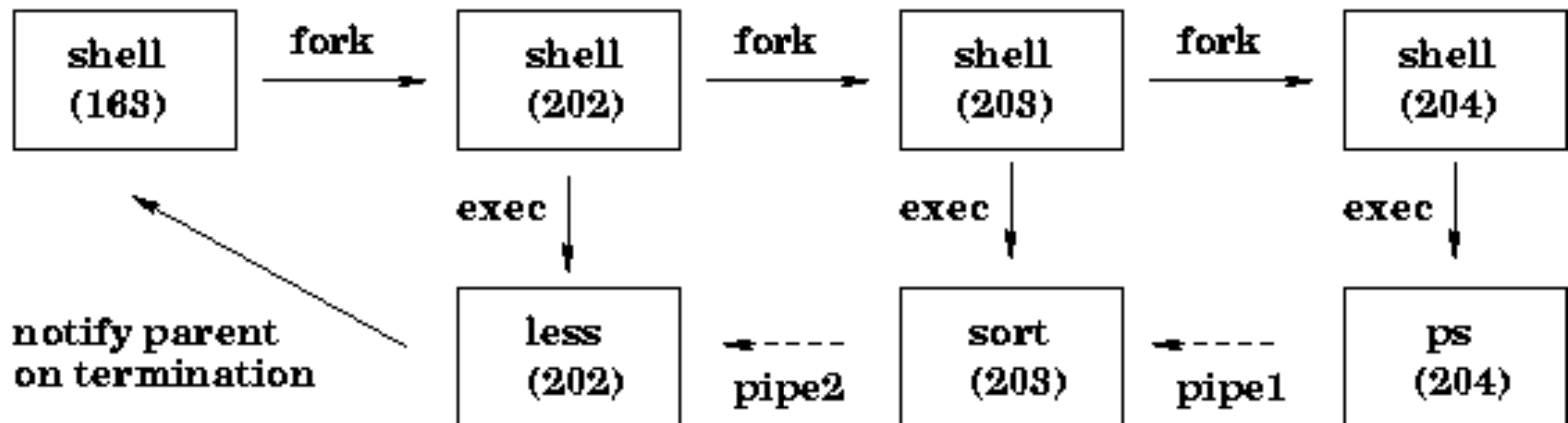
M2-1: Mô hình MVC



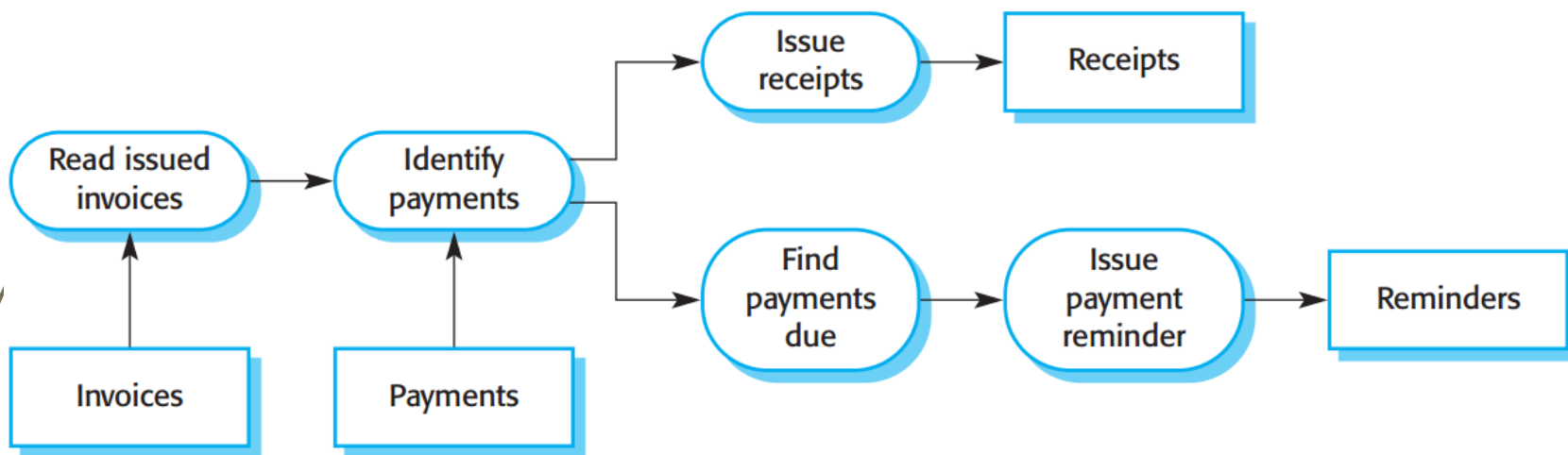
M2-2: Mô hình đường ống chức năng

- Miêu tả quá trình **biến đổi chức năng**, xử lý đầu vào để tạo ra kết quả đầu ra.
- Còn được gọi là mô hình đường ống và bộ lọc (giống như shell UNIX).

ps | sort | less



Ví dụ: Hệ thống xử lý hoá đơn



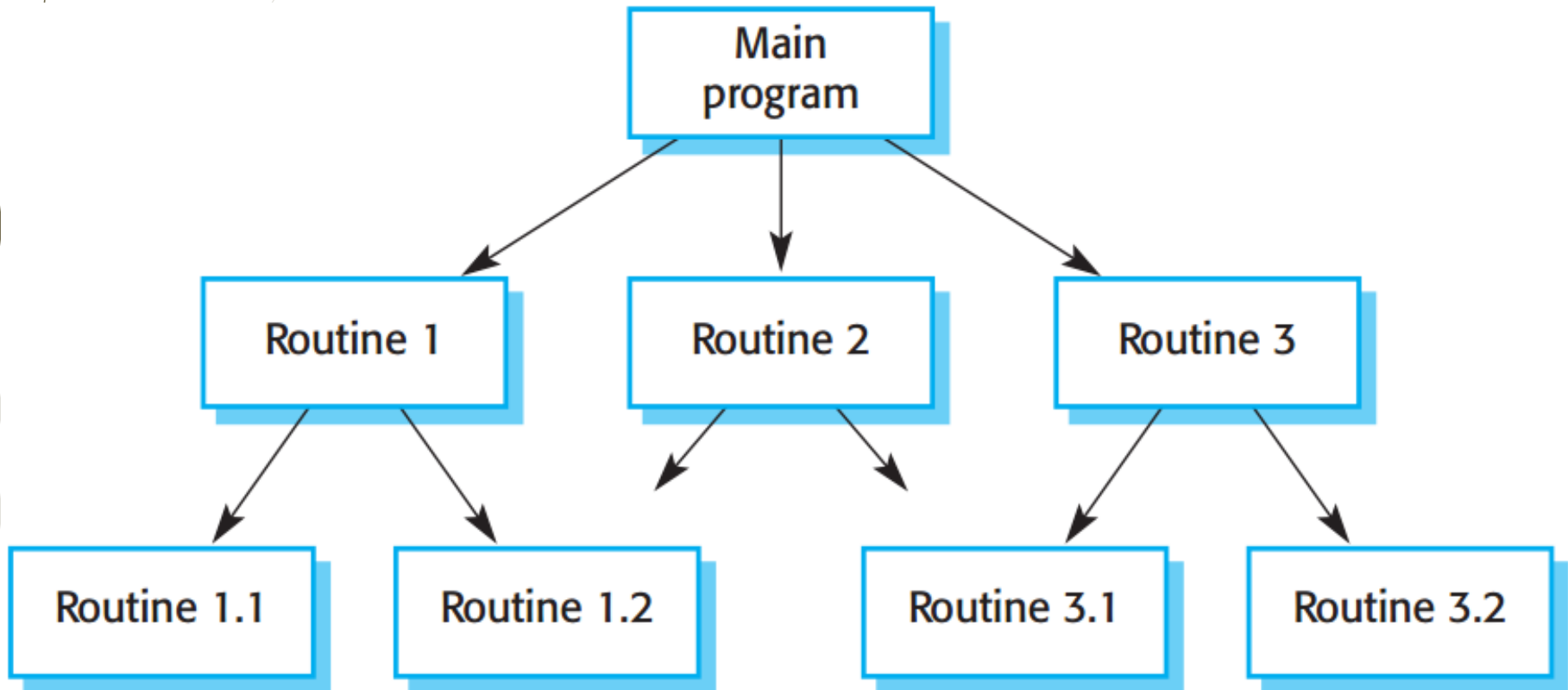
Phong cách điều khiển

- Liên quan đến **luồng điều khiển** giữa các hệ thống con.
- **M3-1: Điều khiển tập trung**
 - Một hệ thống con có trách nhiệm kiểm soát chung, có quyền khởi động hoặc dừng các hệ thống con khác.
- **M3-2: Điều khiển dựa trên sự kiện**
 - Mỗi hệ thống con có thể phản ứng các sự kiện bên ngoài từ các hệ thống con khác hoặc từ môi trường.

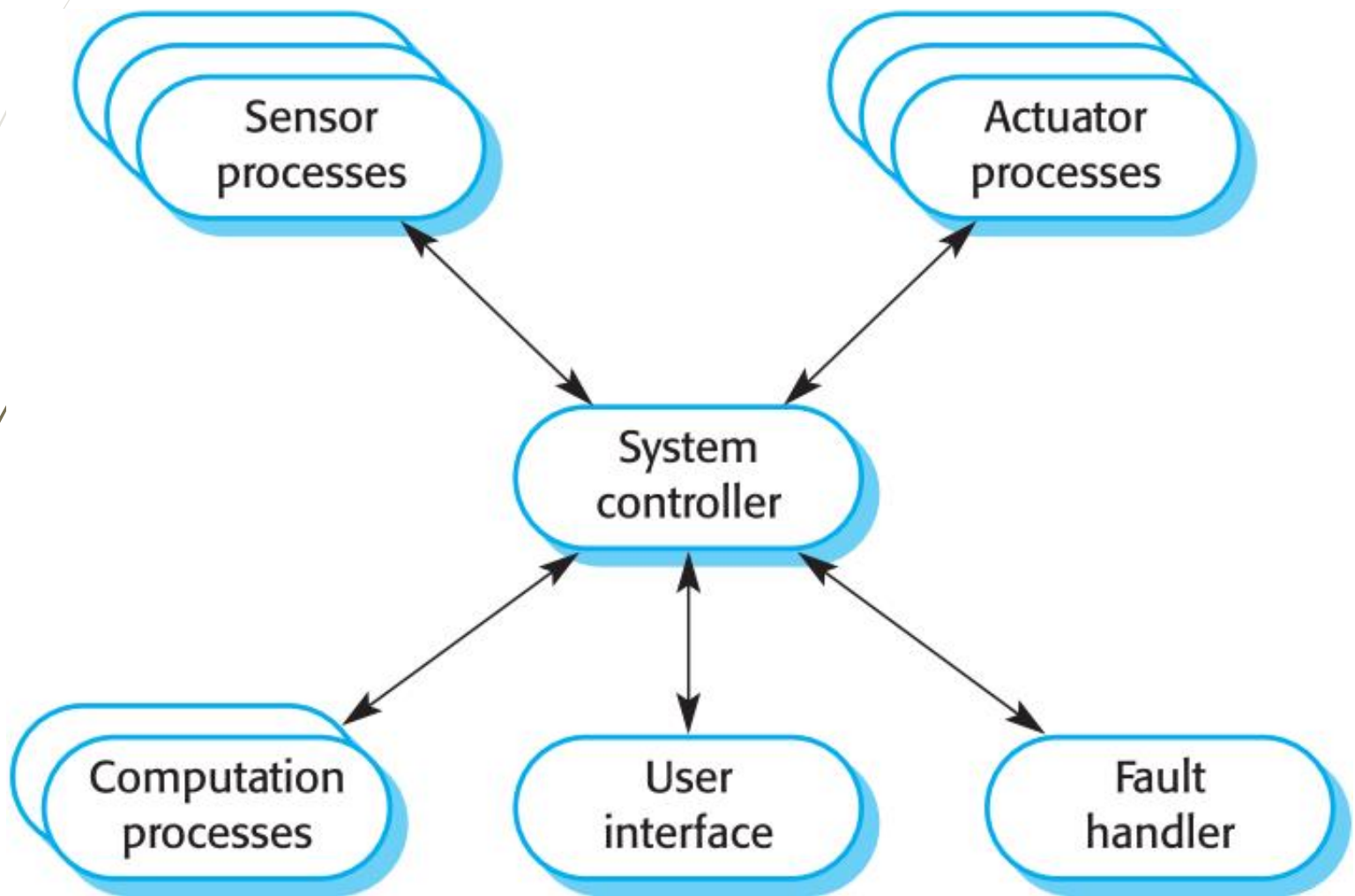
M3-1: Mô hình điều khiển tập trung

- Mô hình **Gọi-Trả_về** (Call-return model)
 - Mô hình hệ thống con điều khiển từ trên xuống dưới.
- Mô hình **Nhà quản lý** (Manager model)
 - Một hệ thống con điều khiển việc dừng, khởi động và điều phối các hệ thống khác.

Mô hình Gọi-Trả_về



Mô hình nhà quản lý cho hệ thống thời gian thực



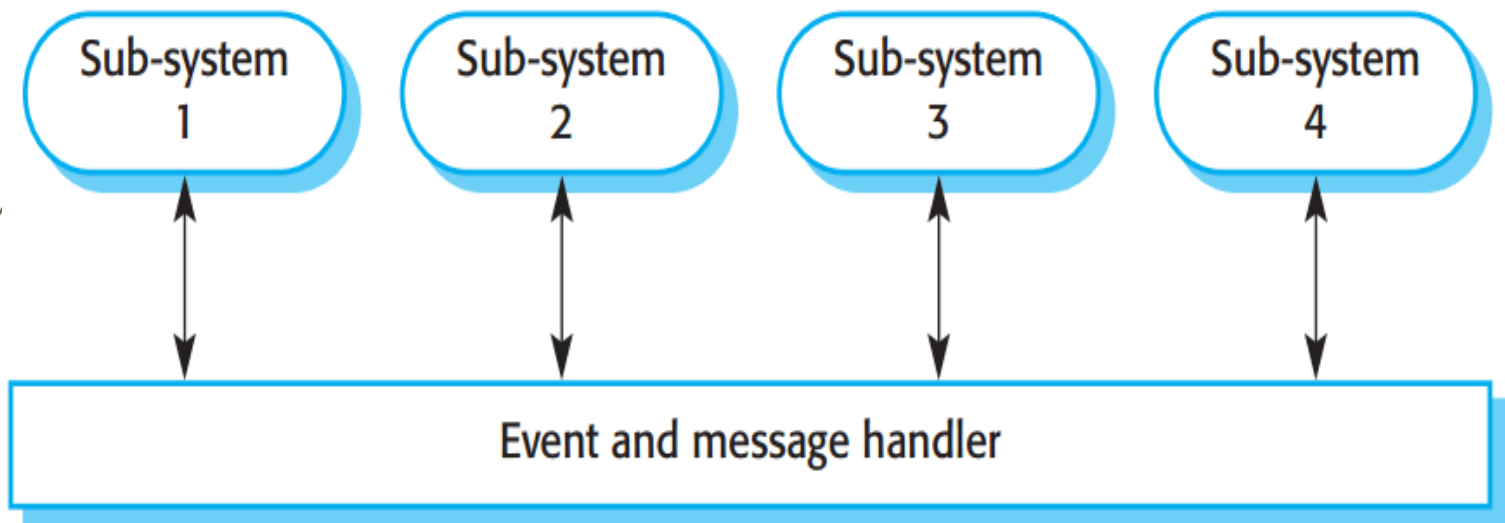
M3-2: Mô hình điều khiển dựa trên hướng sự kiện

- Các hệ thống con được kích hoạt bởi các sự kiện bên ngoài
- Hai mô hình hướng sự kiện chính:
 1. **Mô hình quảng bá** (broadcast models)
 - Sự kiện được truyền tới tất cả các hệ thống con. Bất kỳ hệ thống con nào có thể xử lý sự kiện này đều có thể thực hiện nó;
 2. **Mô hình hướng ngắt** (interrupt-driven models)
 - Được sử dụng trong các hệ thống *thời gian thực* nơi mà các ngắt được phát hiện bởi một trình xử lý ngắt và được truyền cho một số thành phần khác để xử lý.

Mô hình quảng bá

- Các hệ thống con đăng ký sự quan tâm của mình đến các sự kiện cụ thể.
- Khi những sự kiện này xảy ra, quyền kiểm soát được chuyển cho hệ thống con có thể xử lý sự kiện này.
- Tuy nhiên, các hệ thống con không biết liệu hay khi nào một sự kiện sẽ được xử lý.

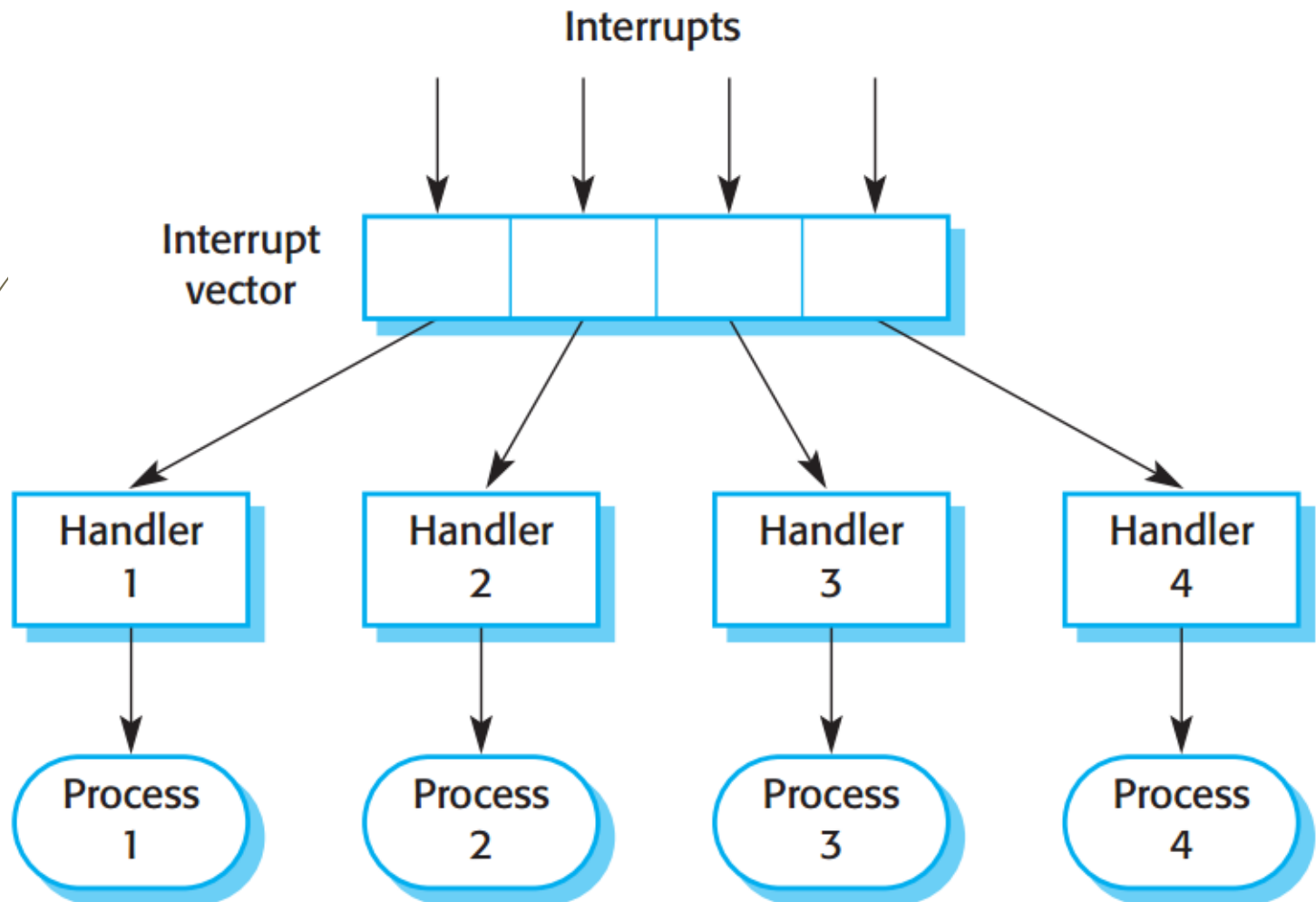
Minh họa: Mô hình quảng bá



Mô hình hướng ngắt

- Thường được sử dụng trong *các hệ thống thời gian thực* khi mà việc phản ứng nhanh với một sự kiện là điều tối cần thiết.
- Mỗi loại ngắt được liên kết với một vị trí bộ nhớ và một chuyển đổi phần cứng gây ra việc chuyển giao cho trình xử lý của nó.

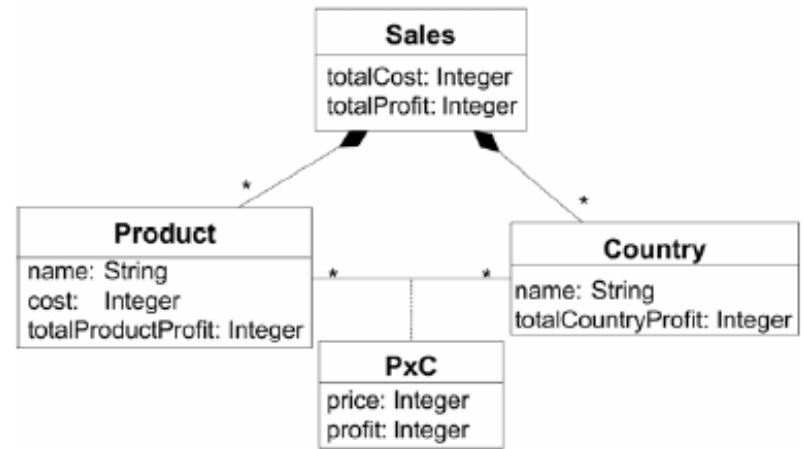
Minh hoạ Mô hình hướng ngắt



Bài tập: Thiết kế kiến trúc

Đề xuất dự án
phần mềm

Thiết kế kiến
trúc cho phần
mềm đó



Phần 2.

33

Thiết kế hướng đối tượng

Object-oriented design

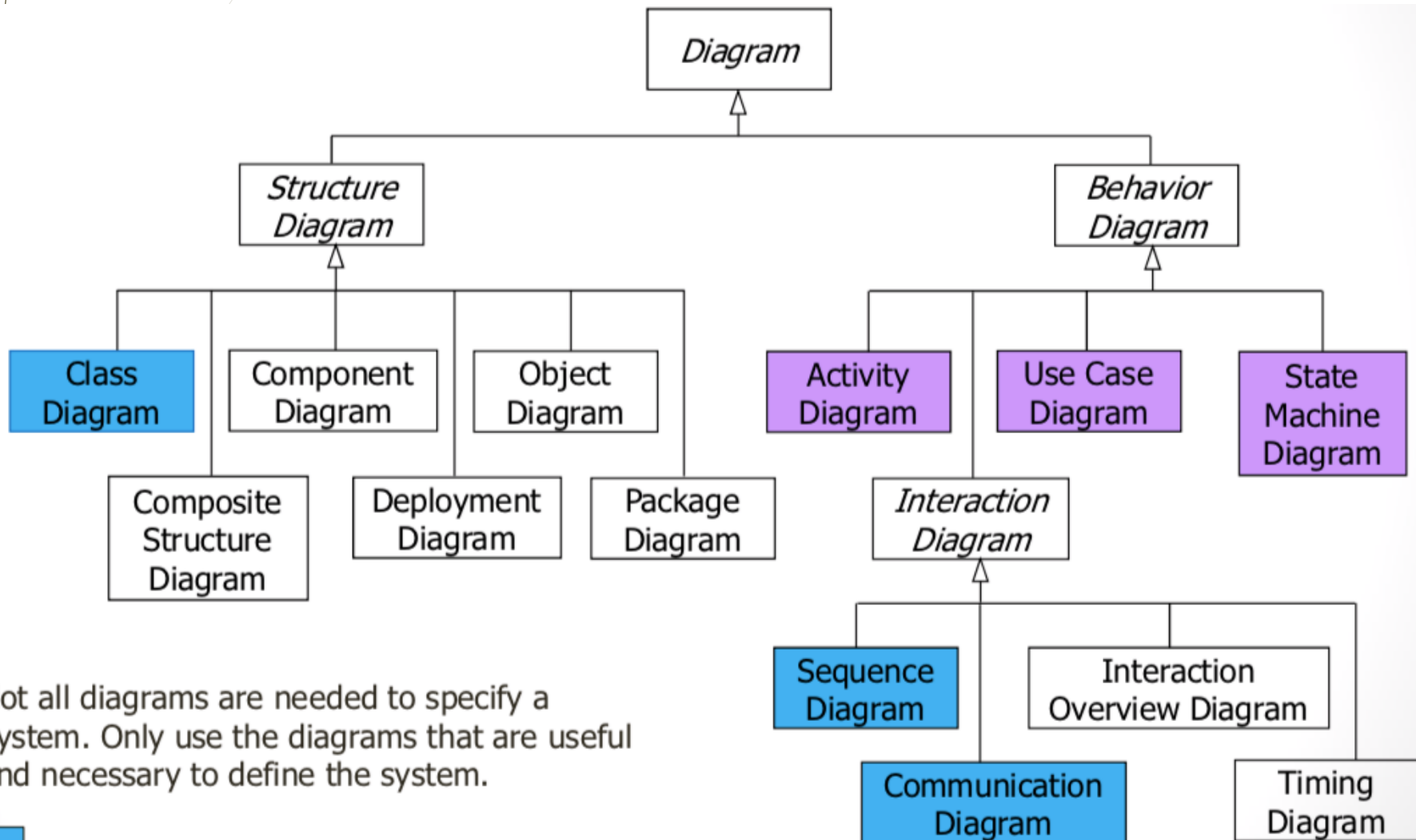
Phát triển phần mềm hướng đối tượng

- ➡ **OOA** liên quan đến việc phát triển một mô hình đối tượng của miền ứng dụng.
- ➡ **OOD** liên quan đến việc phát triển một mô hình hệ thống hướng đối tượng để cài đặt các yêu cầu.
- ➡ **OOP** liên quan đến việc thực hiện một OOD bằng cách sử dụng một ngôn ngữ lập trình OO như Java hay C ++.

Đối tượng và Lớp đối tượng

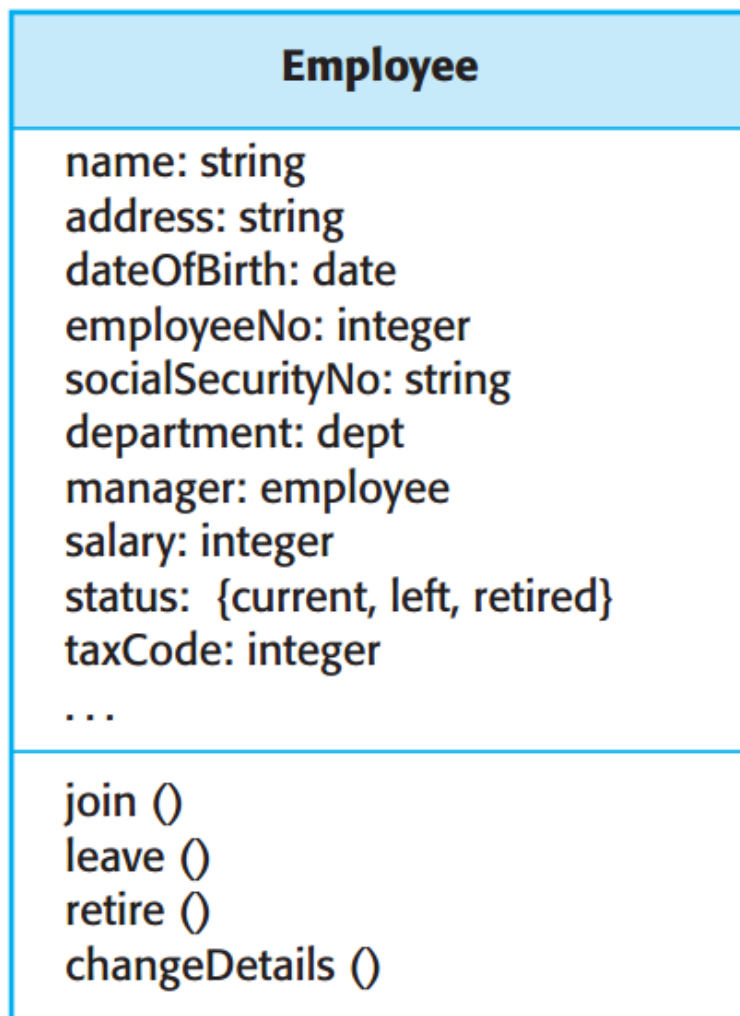
- **Đối tượng** là một *thực thể* trong hệ thống phần mềm để biểu diễn các trường hợp cụ thể trong thế giới thực
- **Lớp đối tượng** là các *khuôn mẫu* cho các đối tượng.
 - Có thể được sử dụng để tạo ra đối tượng.

UML



Not all diagrams are needed to specify a System. Only use the diagrams that are useful and necessary to define the system.

Minh hoạ lớp đối tượng Employee sử dụng UML



Giao tiếp giữa (lớp) đối tượng

- Về mặt lý thuyết, các đối tượng giao tiếp bằng cách truyền thông điệp.
- **Thông điệp**
 - Tên của dịch vụ được yêu cầu bởi đối tượng gọi
 - Bản sao của thông tin được yêu cầu để thực hiện dịch vụ và tên của đối tượng giữ cho kết quả của dịch vụ.
- Trong thực tế, thông điệp thường được cài đặt bằng các lời gọi hàm

Các liên kết giữa các lớp đối tượng trong UML

- Đối tượng và lớp đối tượng có thể có **các mối quan hệ** với đối tượng và lớp đối tượng khác.
- Trong UML, một mối quan hệ tổng quát được biểu diễn bằng một liên kết
- Các liên kết có thể được chú thích với thông tin miêu tả về liên kết đó.

Các mối quan hệ giữa các lớp đối tượng

- Association



- Aggregation



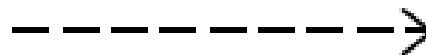
- Composition



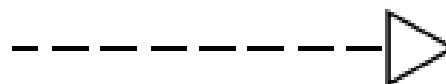
- Generalization

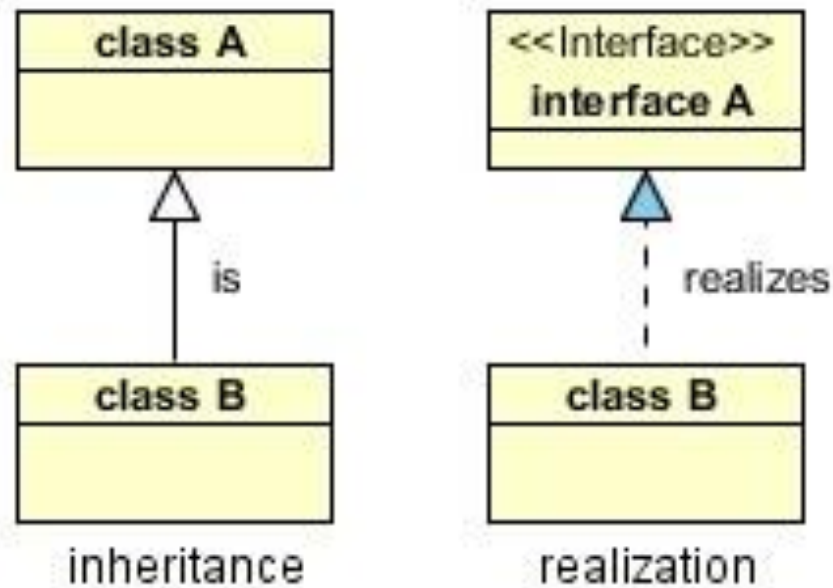
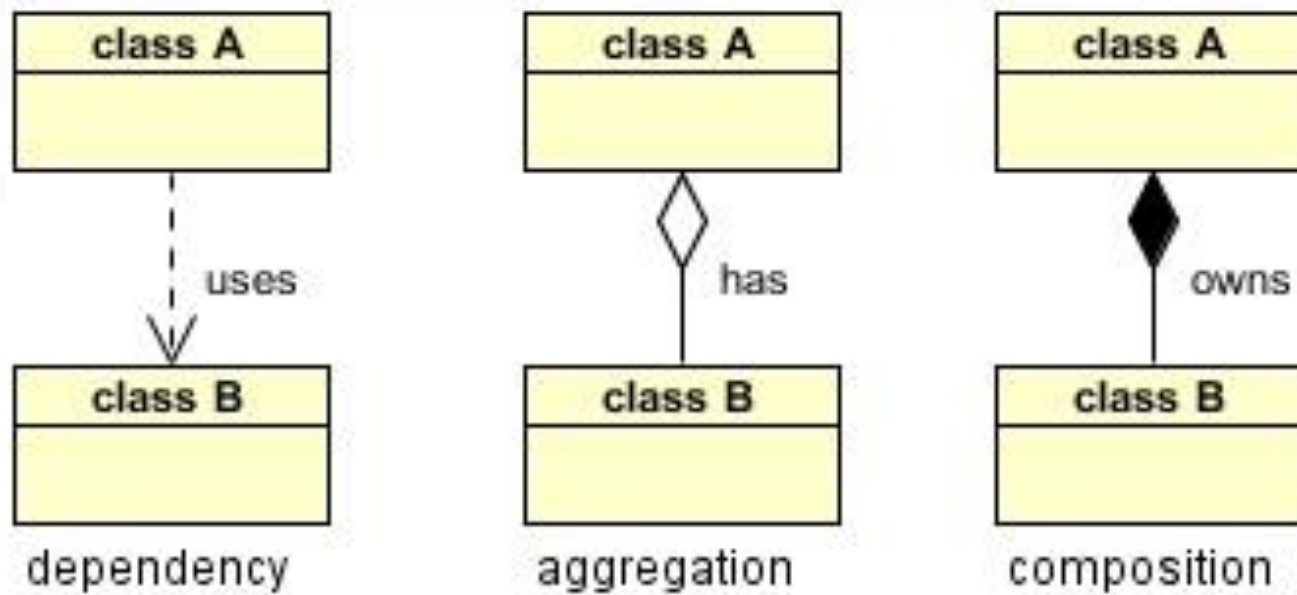


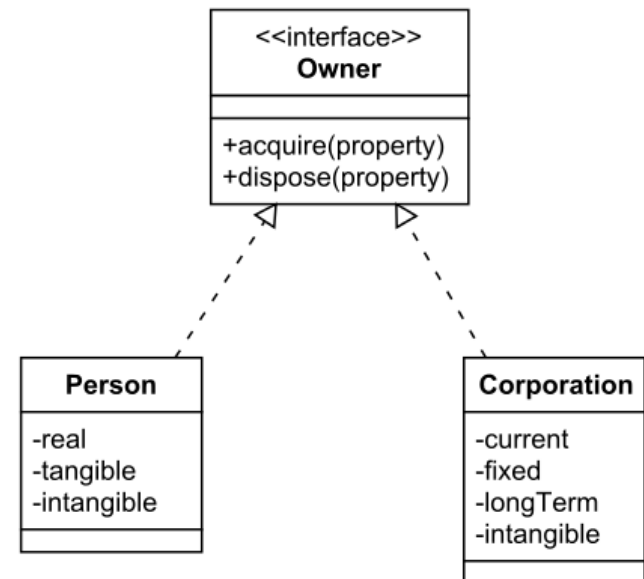
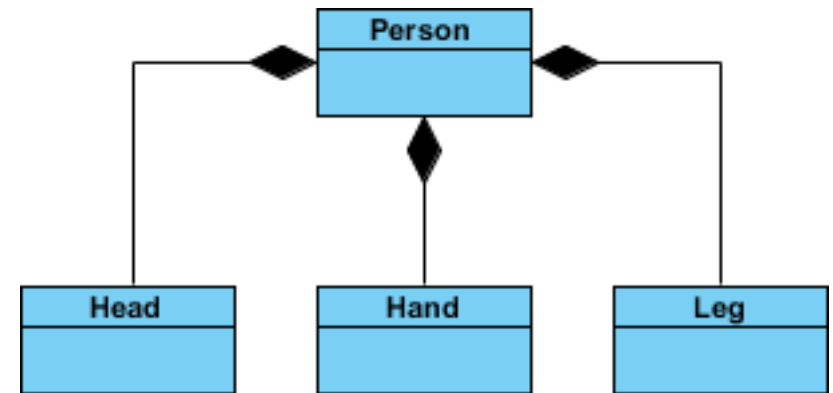
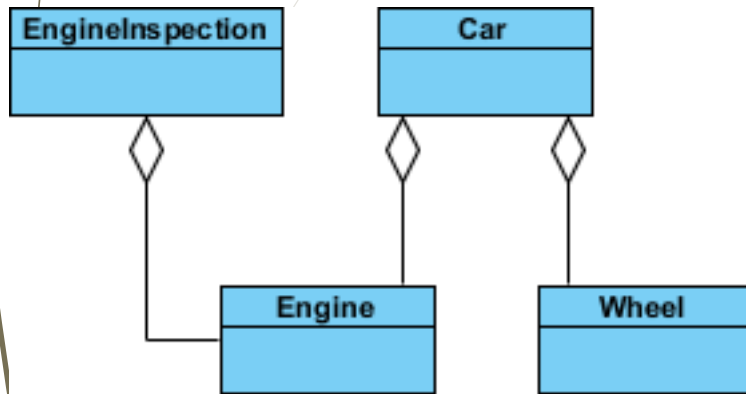
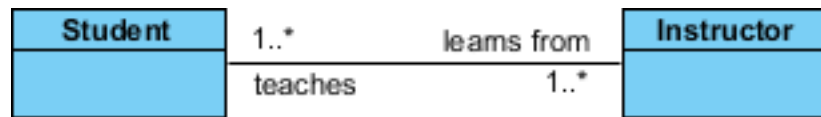
- Dependency



- Realization







Bài tập

- ➡ Đề xuất một dự án phần mềm
- ➡ Thiết kế **sơ đồ lớp** chi tiết sử dụng UML
 1. Liệt kê các lớp trong hệ thống
 2. Sử dụng các ký hiệu để biểu thị mối quan hệ giữa 2 lớp

DESIGN PATTERNS

ENDER AYDIN ORAK @ 2014

Mẫu thiết kế

- ➡ Một mẫu thiết kế là một cách tái sử dụng kiến thức trừu tượng về một vấn đề và giải pháp của nó.
- ➡ Một mẫu là một mô tả về vấn đề và bản chất giải pháp của nó.

Phân loại mẫu thiết kế

- ***Design Patterns: Elements of Reusable Object Oriented Software***, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 1995 (Gang of Four)
- **Creational Patterns**
 - Abstract Factory, Builder, Factory Method, Prototype, Singleton
- **Structural Patterns**
 - Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy
- **Behavioral Patterns**
 - Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, **Observer**, State, Strategy, Template Method, Visitor

Các thành phần của một mẫu thiết kế

1. Tên mẫu thiết kế

- ➡ Là một tên gọi có ý nghĩa, ví dụ: *Observer*

2. Miêu tả vấn đề

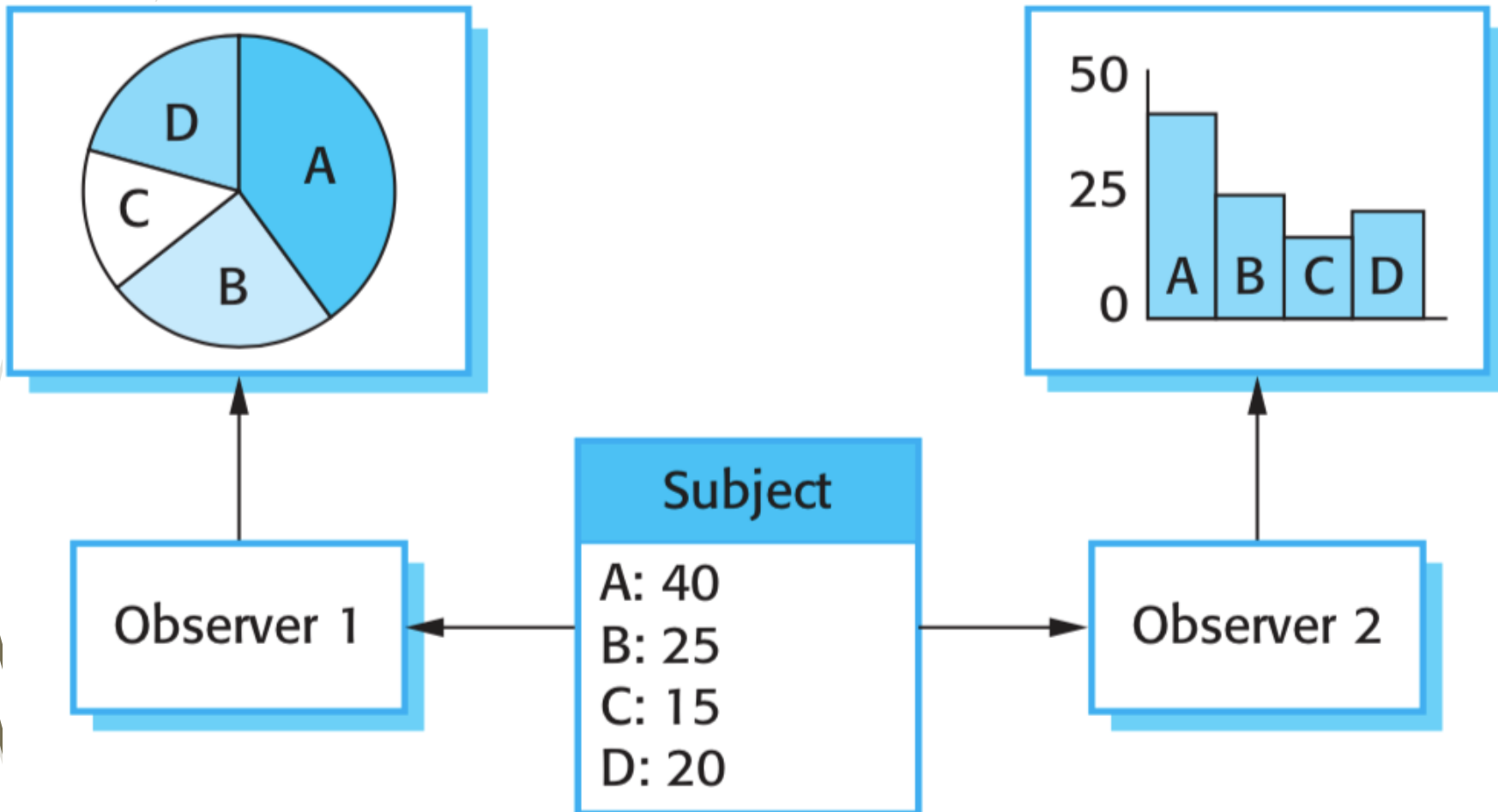
3. Miêu tả giải pháp

- ➡ Không phải là một thiết kế cụ thể mà là một khuôn mẫu cho một giải pháp thiết kế có thể được khởi tạo theo nhiều cách khác nhau.

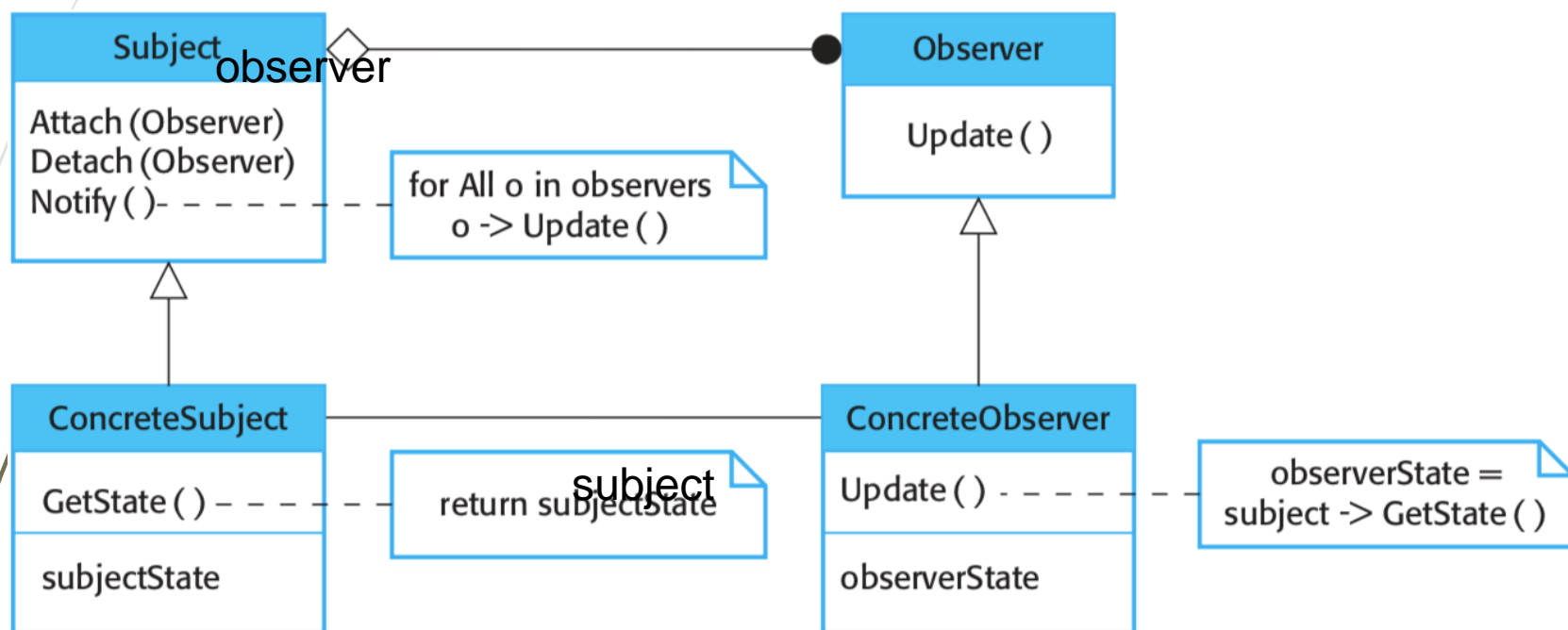
4. Các tác động

- ➡ Những kết quả và sự đánh đổi (trade-offs) trong việc áp dụng mẫu.

Mẫu Observer: Miêu tả vấn đề

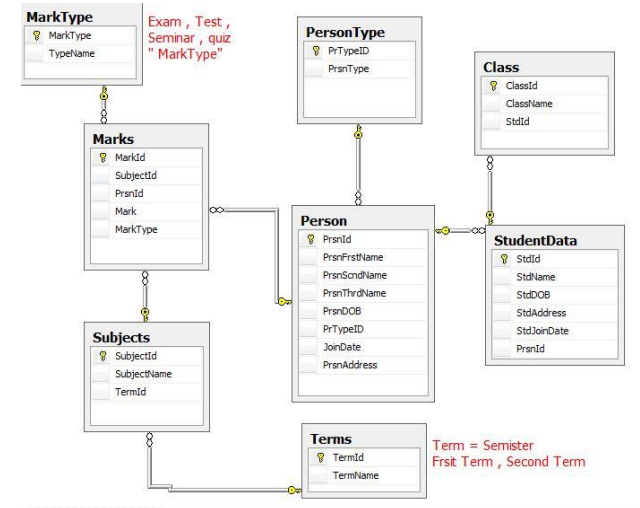


Mẫu Observer: Giải pháp



Phần 3. Thiết kế dữ liệu

Database Design



Thiết kế dữ liệu

- Mục tiêu: mô tả cách thức tổ chức lưu trữ dữ liệu của phần mềm bên trong máy tính
- Ba yêu cầu:
 1. Thiết kế dữ liệu với tính đúng đắn
 2. Thiết kế dữ liệu với tính tiến hóa
 3. Thiết kế dữ liệu hiệu quả về truy suất và lưu trữ

Sơ đồ logic (1)

- Bao gồm các bảng dữ liệu và mối quan hệ giữa chúng
- Các ký hiệu

Tên Table

Bảng dữ liệu



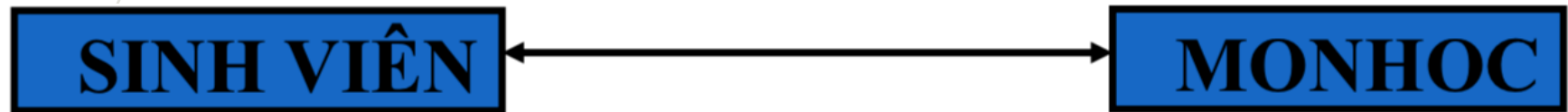
Liên kết



Sơ đồ logic (2)

- ➡ Nếu quan hệ giữa A và B là quan hệ 1-1 thì có thể gộp hai bảng A và B lại thành 1 bảng duy nhất chứa tất cả thuộc tính của A và B
- ➡ Nếu quan hệ giữa A và B là quan hệ 1-N thì giữ nguyên
- ➡ Nếu quan hệ giữa A và B là quan hệ N-N thì tách quan hệ này thành 2 quan hệ 1-N bằng cách thêm vào 1 bảng trung gian chứa khóa chính của cả hai bảng A và B

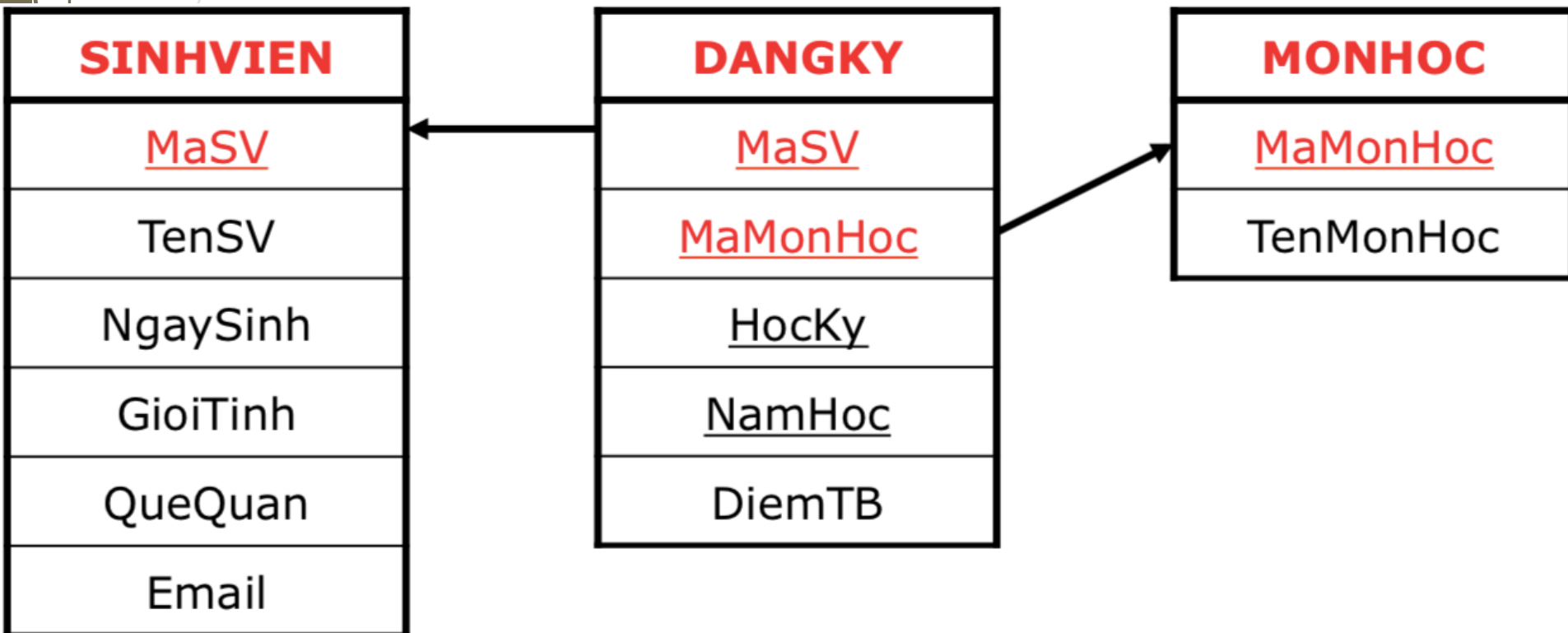
Minh hoạ Quan hệ N-N (1)



Chuyển thành:



Minh hoạ Quan hệ N-N (2)



Xác định khoá chính

➤ Ba tính chất của khoá chính:

1. Tối thiểu
2. Không trùng lặp
3. Không thay đổi theo thời gian

➤ Thuộc tính trừu tượng:

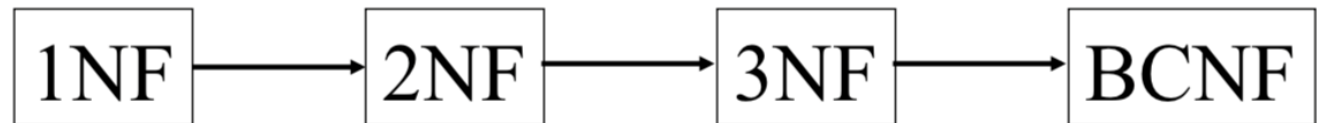
- Là thuộc tính không xuất hiện trong thế giới thực.
- Sử dụng thuộc tính trừu tượng để làm khoá chính cho bảng
- Ví dụ: *MaDaiLy*, *MaLoaiDaiLy*...

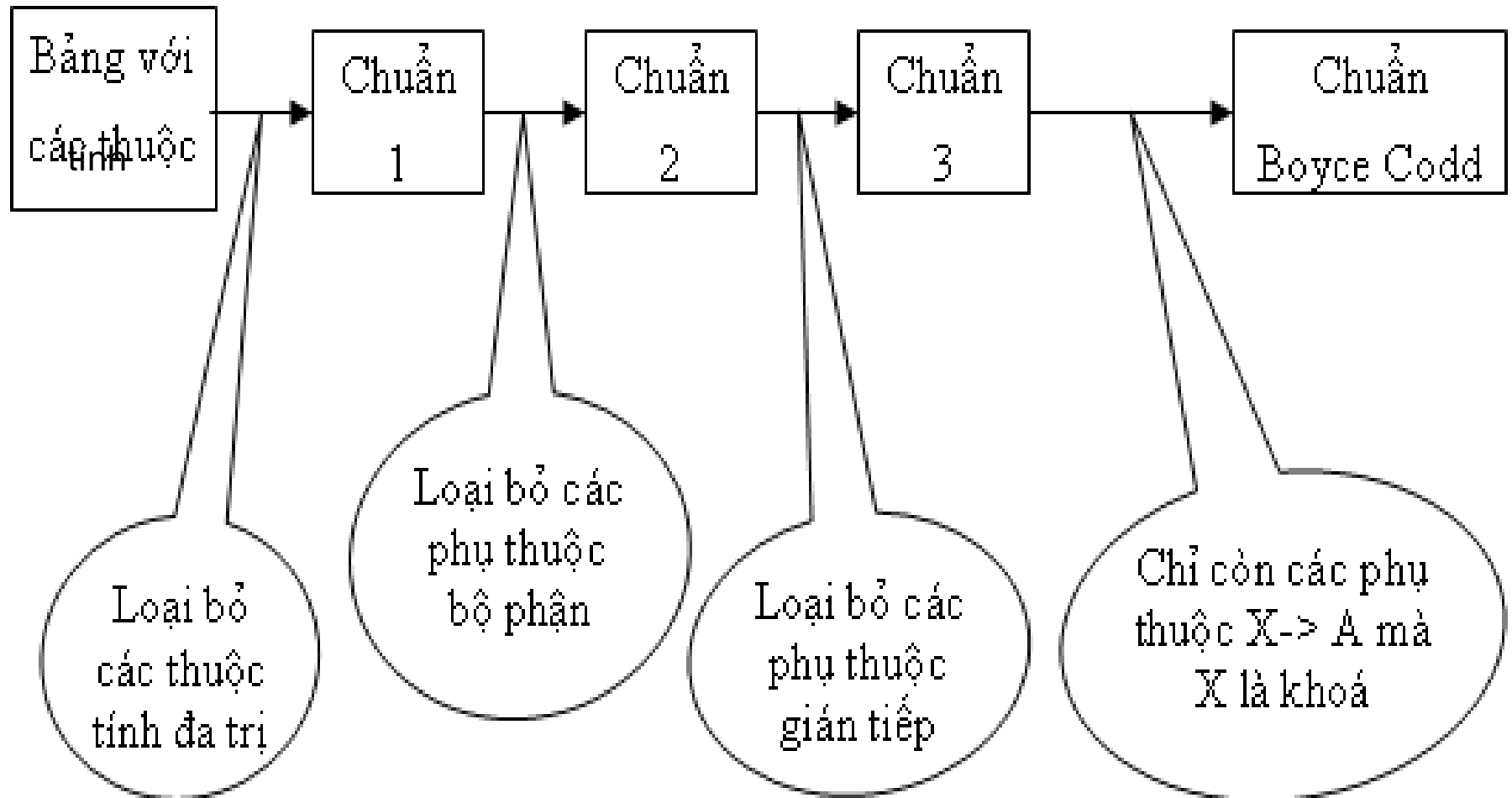
Một số lưu ý khi đặt tên

- **Tên bảng:** viết bằng chữ IN HOA, không dấu, không có khoảng cách giữa các từ
 - Ví dụ: NHANVIEN, KHACHHANG...
- **Tên thuộc tính:** viết hoa các ký tự đầu của mỗi từ, không dấu, không có khoảng cách giữa các từ
 - Ví dụ: HoTen, NgaySinh, DiaChi...

Chuẩn hoá dữ liệu

- Mục đích:
 - Giảm thiểu dư thừa dữ liệu
 - Loại bỏ các bất thường khi cập nhật cơ sở dữ liệu
- Các dạng chuẩn


















Dạng chuẩn 1





1 Normal Form – 1NF

- ➡ Một quan hệ được gọi là ở dạng chuẩn 1 khi và chỉ khi:
 - ➡ Toàn bộ miền giá trị của các thuộc tính trong quan hệ đều chỉ chứa các giá trị nguyên tố
 - ➡ Giá trị của các thuộc tính trên các hàng phải là đơn trị, không chứa nhóm lặp
 - ➡ Không có một thuộc tính nào có giá trị có thể được tính toán từ các thuộc tính khác

customer_id	customer_name	phone	address	order_id	order_date	product_id	description	quantity	unit_price	total_amount
1	Chu Quynh Trang	123456789	Ha Noi, Hung Yen	1001	2017-06-26	MH370	may bay	1	321000	321000
2	Nguyen Quynh Thao	987654321	Ho Chi Minh, CampuChia	1002	2017-06-25	AK47	sung tieu lien	10	50000	500000
3	Chu Quynh Trang	123456789	Ha Noi, Hung Yen	1004	2017-06-27	BA001	truyen tranh	3	41000	123000
4	Nguyen Van Thuy	343419993	Hai Duong	1005	2017-06-27	BD008	bong da	2	940000	1880000
5	Nguyen Van Thuy	343419993	Hai Duong	1006	2017-06-25	HI345	ao phong	9	111000	999000
6	Nguyen Van Thuy	343419993	Hai Duong	1007	2017-06-28	NA000	giay the thao	1	515000	515000

  vidu addresses
 customer_id : int(11)
 address : varchar(255)

  vidu orders
 customer_id : int(11)
 order_id : int(11)
 order_date : int(11)
 product_id : int(11)
 description : int(11)
 quantity : int(11)
 unit_price : int(11)

  vidu customers
 customer_id : int(11)
 customer_name : varchar(255)
 phone : int(11)

Dạng chuẩn 2

2 Normal Form – 2NF

- Một quan hệ ở dạng chuẩn 2 nếu:
 - Thoả mãn 1NF
 - Các thuộc tính không phải là khoá **phụ thuộc đầy đủ** vào **tập các thuộc tính tham gia làm khoá chính**
 - Ví dụ:
 - $R = (\underline{A}, \underline{B}, C, D, E)$
 - $F = \{B \rightarrow D, C\}$
- =>
- $R1(\underline{B}, C, D)$
 - $R2(\underline{A}, \underline{B}, E)$

Dạng chuẩn 3

3 Normal Form – 3NF

- Một quan hệ ở dạng chuẩn 3 nếu:
 - Thoả mãn 2NF
 - Các thuộc tính không phải là khoá phụ thuộc trực tiếp vào khoá chính
 - Ví dụ:
 - $R = (\underline{A}, \underline{B}, C, D, G, H)$
 - $F = \{A, B \rightarrow C; A, B \rightarrow D; A, B \rightarrow G, H; G \rightarrow D, H\}$
- =>
- $R_1(\underline{G}, D, H)$
 - $R_2(\underline{A}, \underline{B}, C, G)$

Dạng chuẩn Boyce Codd - BCNF

- Một quan hệ ở dạng chuẩn Boyce Codd nếu:
 - Thoả mãn 3 NF
 - Không có thuộc tính khoá nào phụ thuộc vào thuộc tính không khoá
 - Ví dụ:
 - $R = (\underline{A}, \underline{B}, C, D, G, H)$
 - $F = \{A, B \rightarrow C; A, B \rightarrow D; A, B \rightarrow G, H; H \rightarrow B\}$
- =>
- $R_1(\underline{H}, B)$
 - $R_2(\underline{A}, \underline{H}, C, D, G)$



Phần 4.

Thiết kế giao diện người dùng

User Interface design

65

A problem has been detected and Windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: kbdhid.sys

MANUALLY_INITIATED_CRASH

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use safe mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x000000e2 (0x00000000, 0x00000000, 0x00000000, 0x00000000)

*** kbdhid.sys - Address 0x94efd1aa base at 0x94efb000 DateStamp 0x4a5bc705

Giao diện người dùng

- Giao diện người dùng nên được thiết kế phù hợp với kỹ năng, kinh nghiệm và mong muốn của người dùng dự kiến.
- Người dùng thường đánh giá một hệ thống qua giao diện hơn là chức năng của nó.
 - Một giao diện được thiết kế tồi có thể khiến người dùng tạo ra một lỗi thảm họa.
 - Thiết kế giao diện người dùng kém là lý do tại sao rất nhiều hệ thống phần mềm không bao giờ được sử dụng.

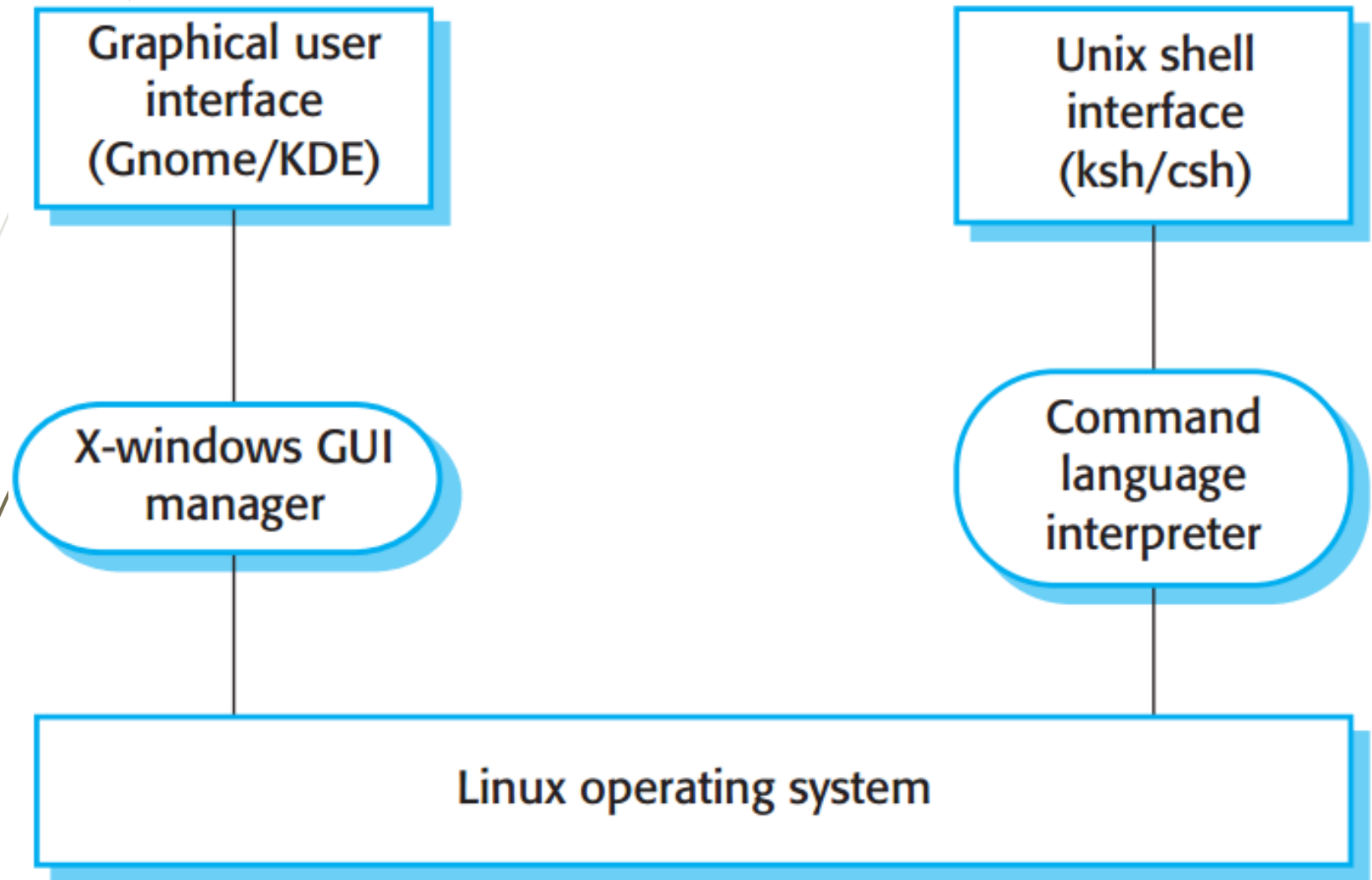
Yếu tố con người trong thiết kế giao diện

- **Bộ nhớ ngắn hạn**
 - Khoảng 7 thông tin
- **Dễ mắc sai lầm**
- **Cách tương tác, sở thích khác nhau**

Nguyên tắc thiết kế giao diện người dùng

1. Thân thiện người dùng
2. Nhất quán
3. Ít gây ngạc nhiên
4. Khả năng phục hồi
5. Hướng dẫn người dùng
6. Đa dạng người dùng

Giao diện đa người dùng

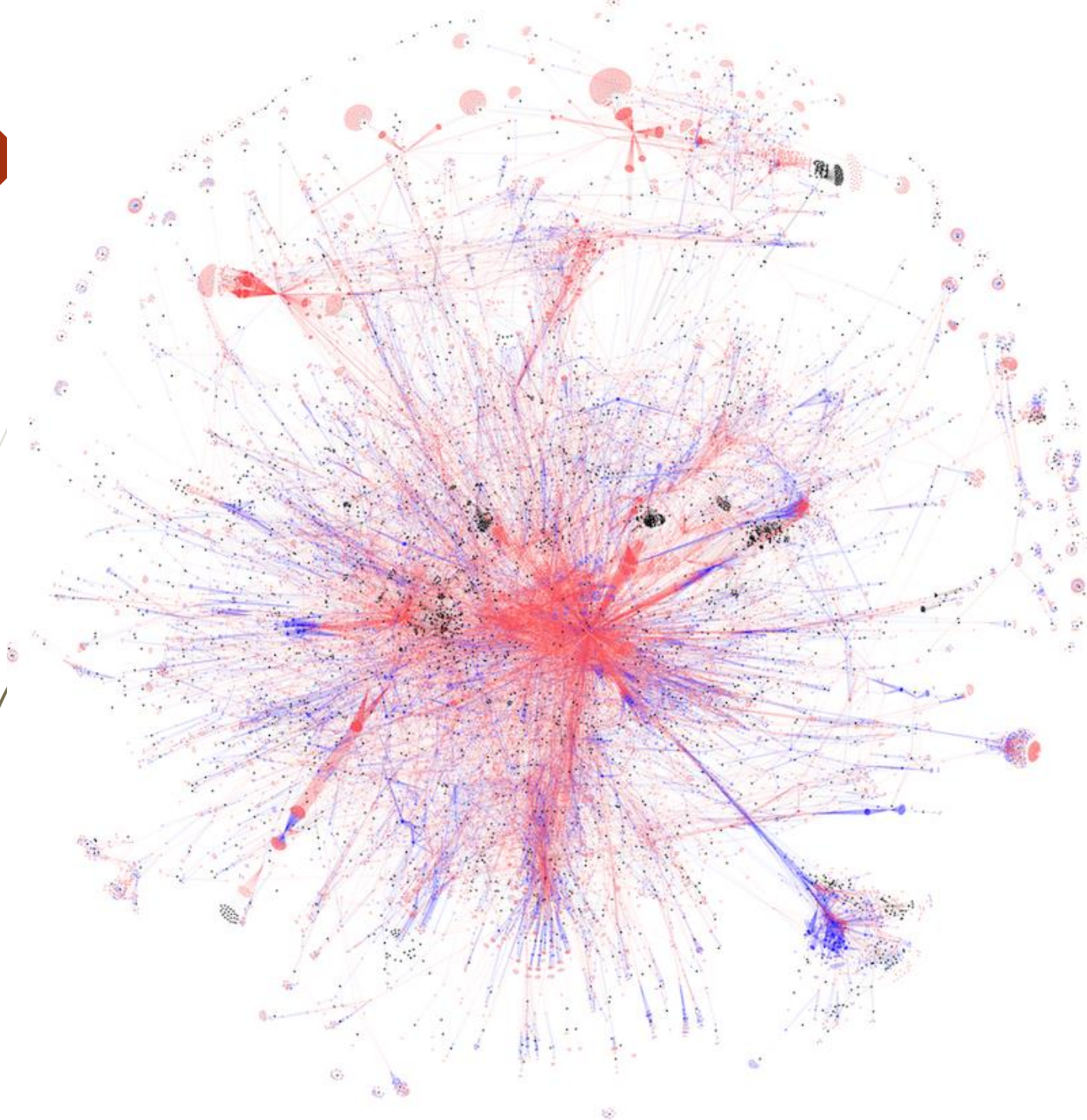


Phong cách tương tác

Phong cách	Ví dụ
Thao tác trực tiếp	Trò chơi điện tử
Lựa chọn menu	Hầu hết các hệ thống mục đích chung
Điền vào biểu mẫu	Kiểm soát chứng khoán, xử lý khoản vay cá nhân
Ngôn ngữ mệnh lệnh	Hệ điều hành, Hệ thống chỉ huy và điều khiển

Hiện thị màu sắc

- Màu sắc bổ sung thêm thông tin cho giao diện
- **Sai lầm phổ biến** trong việc sử dụng màu sắc khi thiết kế giao diện: *sử dụng quá mức màu sắc trong hiển thị*
- Hướng dẫn sử dụng màu sắc:
 - Hạn chế số lượng màu được sử dụng.
 - Sử dụng sự thay đổi màu sắc để hiển thị thay đổi trạng thái hệ thống.



Thông báo lỗi

- Thiết kế thông báo lỗi rất quan trọng.
- Thông báo phải: ***lịch sự, súc tích, nhất quán và xây dựng.***
- Các nhân tố khi diễn đạt thông báo lỗi:
 - Ngữ cảnh
 - Kinh nghiệm
 - Kỹ năng
 - Phong cách
 - Văn hóa

Ví dụ: Lỗi người dùng

- ➡ Giả sử một y tá đánh sai tên của một bệnh nhân khi tìm hồ sơ

Please type the patient name in the box then click on OK

Patient name

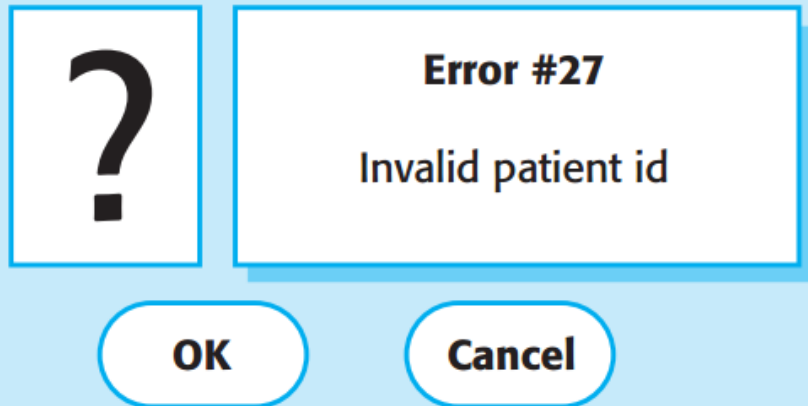
MacDonald, R.

OK

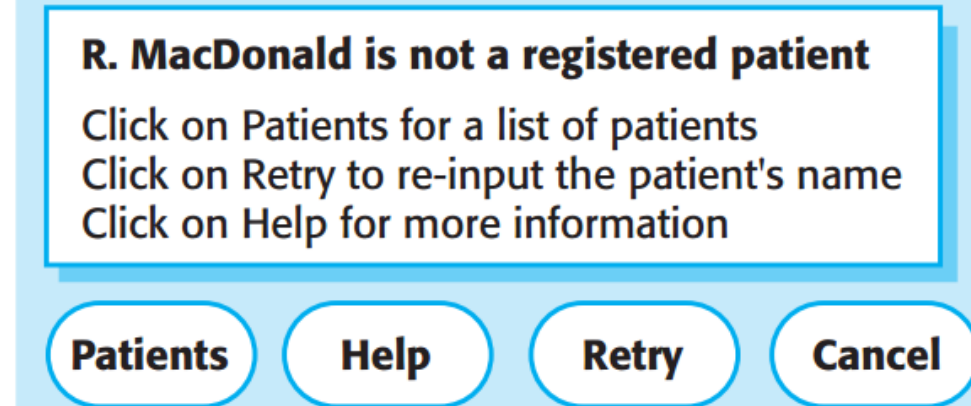
Cancel

Thông báo lỗi tồi và tốt?

System-oriented error message



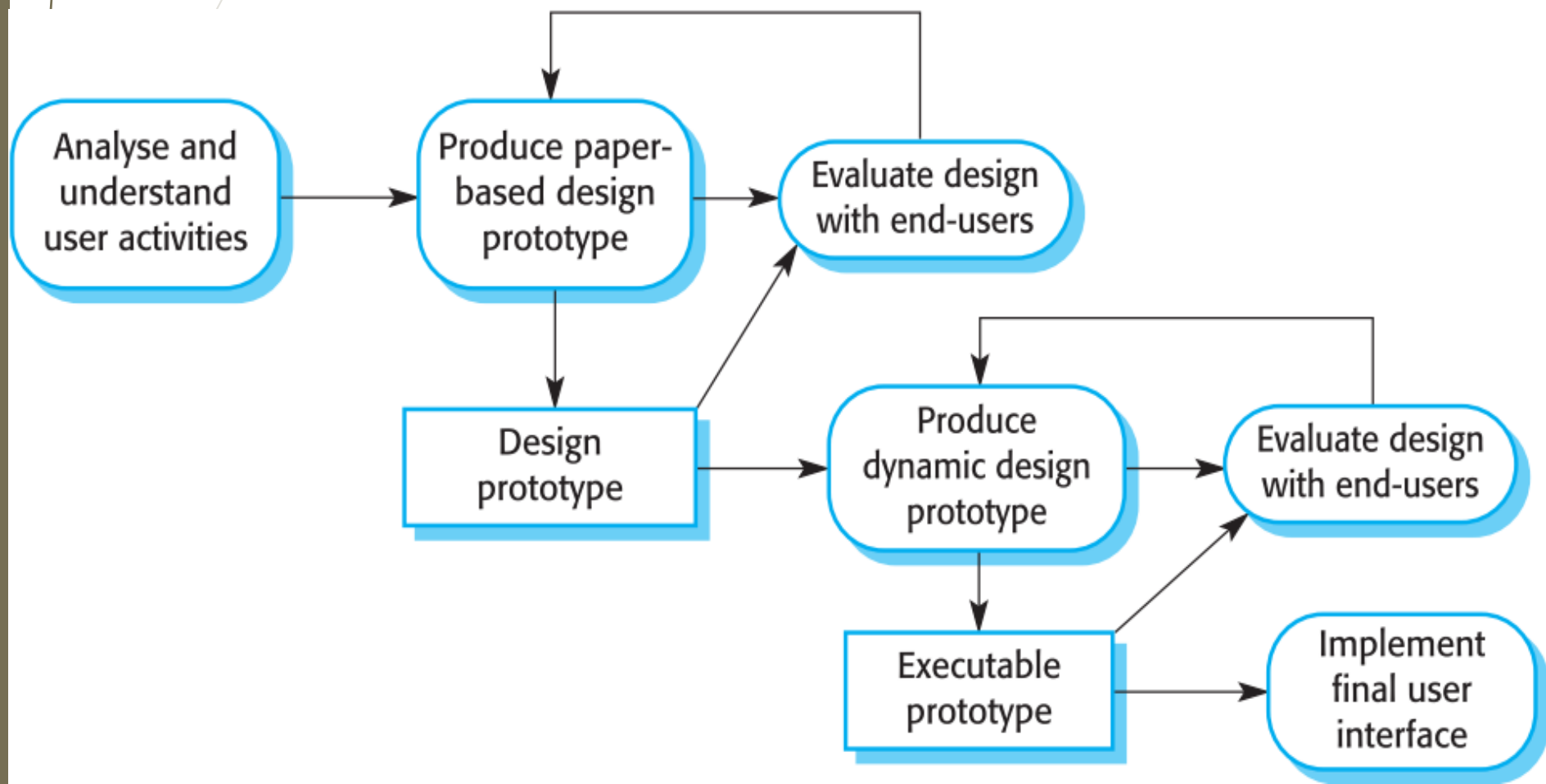
User-oriented error message



Quy trình thiết kế giao diện người dùng (1)

- Thiết kế giao diện người dùng là một quá trình lập liên quan chặt chẽ giữa người dùng và nhà thiết kế.
- Ba hoạt động cốt lõi trong quy trình này là:
 1. Phân tích người dùng
 2. Tạo nguyên mẫu giao diện
 3. Đánh giá nguyên mẫu giao diện

Quy trình thiết kế giao diện người dùng (2)



Tài liệu tham khảo

- ***Software engineering: A practitioner's approach***, Part 1 & Part 2, Roger S. Pressman, McGraw-Hill Higher Education, 2010. (#000021579)
- ***Software Engineering***, Ian Sommerville, 10th Edition, 2016
- ***Kỹ nghệ Phần mềm***, TS Lê Văn Phùng, Nhà xuất bản thông tin và truyền thông, 2014
- ***Nhập Môn Công Nghệ Phần Mềm***, Phạm Thị Quỳnh