

Phân tích thiết kế hướng đối tượng

Bài 12: Từ phân tích đến thiết kế

TS. Nguyễn Hiếu Cường

Bộ môn CNPM, Khoa CNTT

Trường ĐH GTVT

cuonggt@gmail.com

Phân tích và Thiết kế

- Phân tích
 - Những yêu cầu phải thực hiện – What?
 - Chủ yếu quan tâm đến các yêu cầu chức năng
 - Độc lập với cài đặt (implementation independent)
- Thiết kế
 - Làm sao thực hiện được các yêu cầu chỉ ra ở bước phân tích – How?
 - Quan tâm cả yêu cầu chức năng và phi chức năng
 - Hướng tới cài đặt: architecture, user interface, database...

Thiết kế

- Lựa chọn chiến lược thiết kế
 - Tự phát triển (custom development)
 - Mua một phần (packaged software)
 - Thuê ngoài (outsourcing)
- Thiết kế tổng thể
 - Kiến trúc hệ thống
- Thiết kế chi tiết
 - Giao diện, dữ liệu, và chi tiết các lớp

Tự phát triển

- Ưu điểm
 - Đáp ứng những nhu cầu chuyên biệt
 - Linh hoạt trong giải quyết vấn đề và dễ dàng thay đổi
 - Bảo mật về công nghệ và chiến lược kinh doanh
 - Nâng cao kỹ năng cho nhóm phát triển
- Nhược điểm
 - Có thể vượt quá nguồn lực cho phép
 - Có thể không hiệu quả: “re-invent the wheel”...

Mua một phần

- Ưu điểm
 - Có thể hoàn thiện nhanh hơn
 - Sử dụng các thành phần đã có sẵn
 - Components (ActiveX, Java beans...)
 - Tools (shopping cart program...)
- Nhược điểm
 - Phải chấp nhận những gì đã được cung cấp
 - Không linh hoạt khi cần thay đổi
 - Vấn đề tích hợp hệ thống (có thể từ nhiều nguồn khác nhau)

Thuê ngoài

- Ưu điểm
 - Có thể đảm bảo thời gian và chất lượng hơn
 - Có thể thực hiện khi không có đủ kỹ thuật, kinh nghiệm...
 - Đa dạng hóa sản phẩm
- Nhược điểm
 - Lựa chọn nhà cung cấp
 - Tích hợp hệ thống từ nhiều nhà cung cấp
 - Bảo mật thông tin: công nghệ, chiến lược kinh doanh...

Lựa chọn chiến lược thiết kế

	Use Custom Development When...	Use a Packaged System When...	Use Outsourcing When...
Business Need	The business need is unique.	The business need is common.	The business need is not core to the business.
In-house Experience	In-house functional and technical experience exists.	In-house functional experience exists.	In-house functional or technical experience does not exist.
Project Skills	There is a desire to build in-house skills.	The skills are not strategic.	The decision to outsource is a strategic decision.
Project Management	The project has a highly skilled project manager and a proven methodology.	The project has a project manager who can coordinate the vendor's efforts.	The project has a highly skilled project manager at the level of the organization that matches the scope of the outsourcing deal.
Time frame	The time frame is flexible.	The time frame is short.	The time frame is short or flexible.

Kiến trúc hệ thống

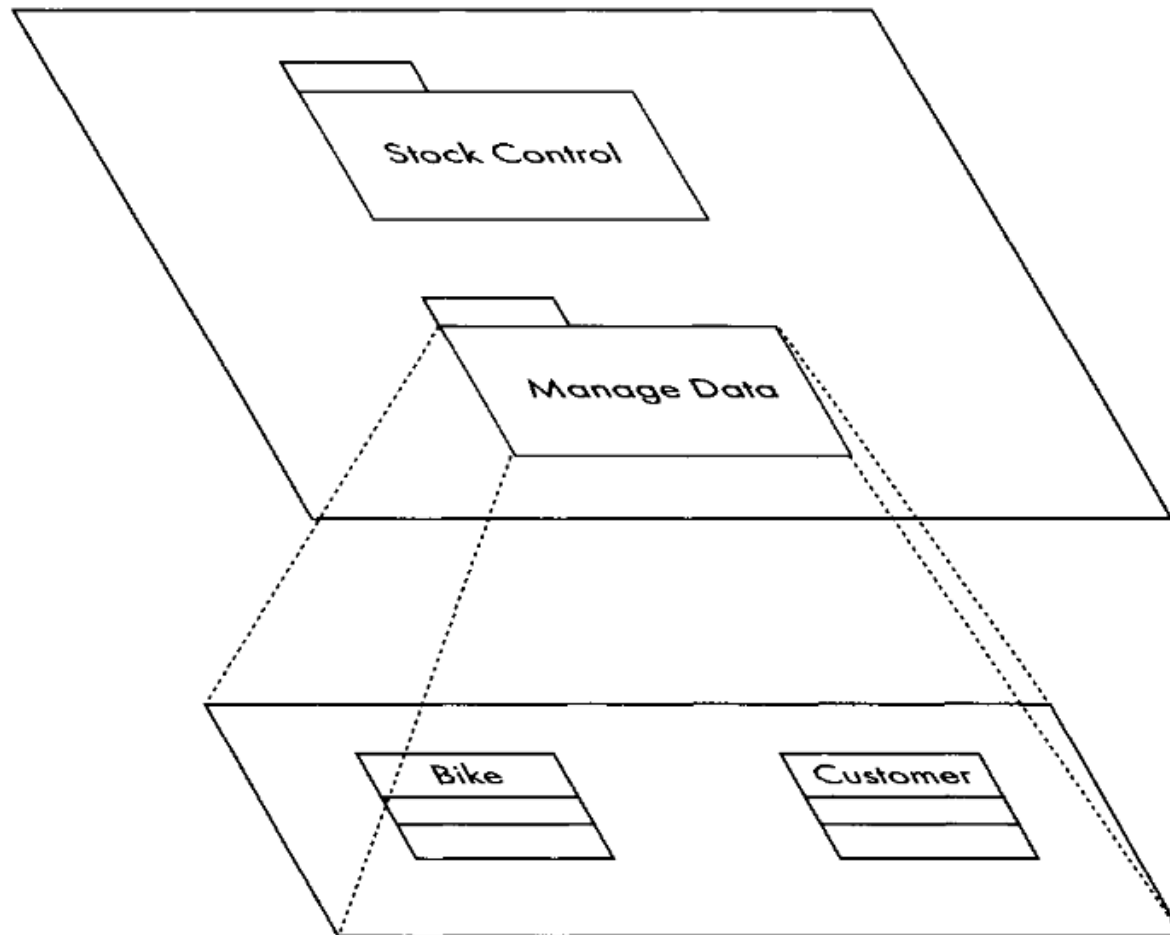
- Kiến trúc phần mềm
 - Kiến trúc logic (Logical software architecture)
 - Class diagram
 - Package diagram
 - Kiến trúc vật lý (Physical software architecture)
 - Component diagram
- Kiến trúc phần cứng (Hardware architecture)
 - Deployment diagram

Phân hoạch (partition)

- Hệ thống có thể phân hoạch thành các gói (package)
- Gói dùng để nhóm các phần tử mô hình:
 - Các ca sử dụng (trong biểu đồ ca sử dụng)
 - Các lớp (trong biểu đồ lớp)
 - Các đối tượng (tương tác với nhau trong biểu đồ cộng tác)
- Ví dụ
 - Phân tách hệ thống Wheels thành 2 gói, mỗi gói có các lớp liên quan:
 - Gói Stock Control: Hire, Payment...
 - Gói Manage Data: Bike, Customer...

Ví dụ (phân hoạch thành các gói)

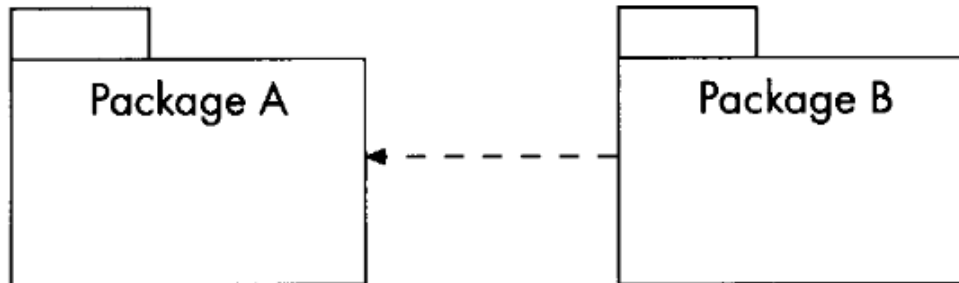
Level 1 Package
diagram of
subsystem



Level 2 class
diagram for
Manage Data
subsystem

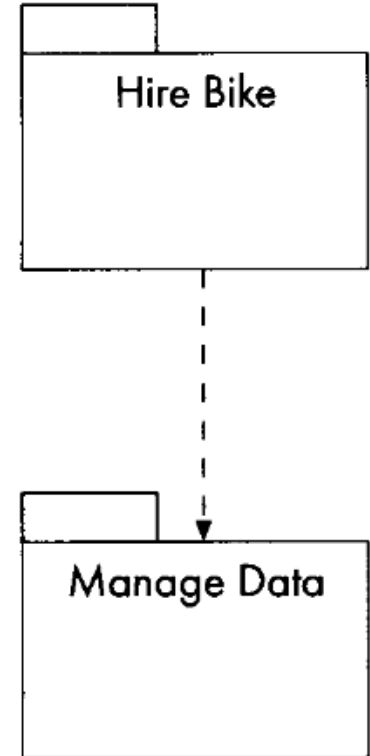
Phụ thuộc giữa các gói

- Phụ thuộc giữa hai gói (dependency) nếu sự thay đổi ở gói này có ảnh hưởng đến gói kia
- Gói B phụ thuộc vào gói A
 - Nếu sự thay đổi ở gói A ảnh hưởng đến gói B
 - Ví dụ: Nếu có một lớp của gói B phụ thuộc vào một lớp của gói A thì gói B phụ thuộc gói A

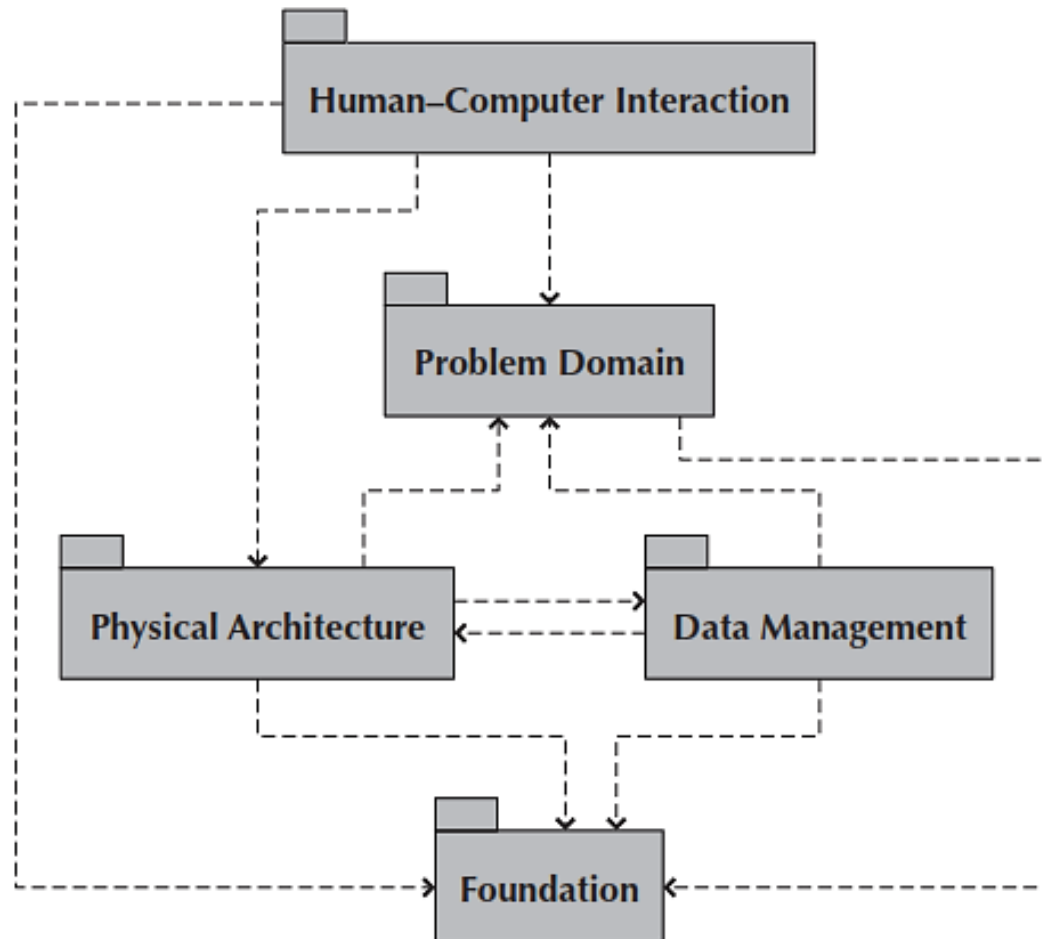


Ví dụ (phụ thuộc)

- Gói Hire Bike có các lớp: Hire và Payment
 - Một số lớp trong gói Hire Bike cần sử dụng các dịch vụ của lớp Bike
 - Bike là một lớp nằm trong gói Manage Data
- ➔ Gói Hike Bike phụ thuộc gói Manage Data



Ví dụ (biểu đồ gói)

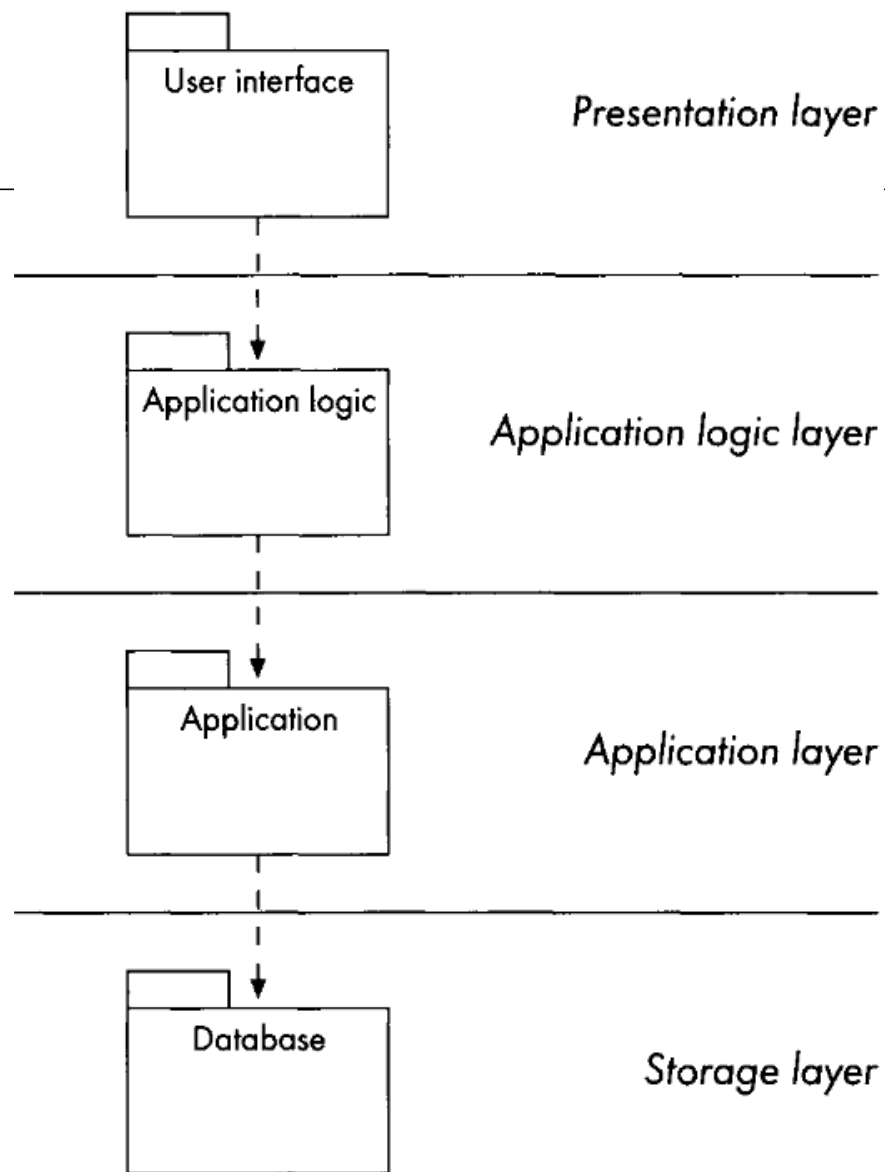


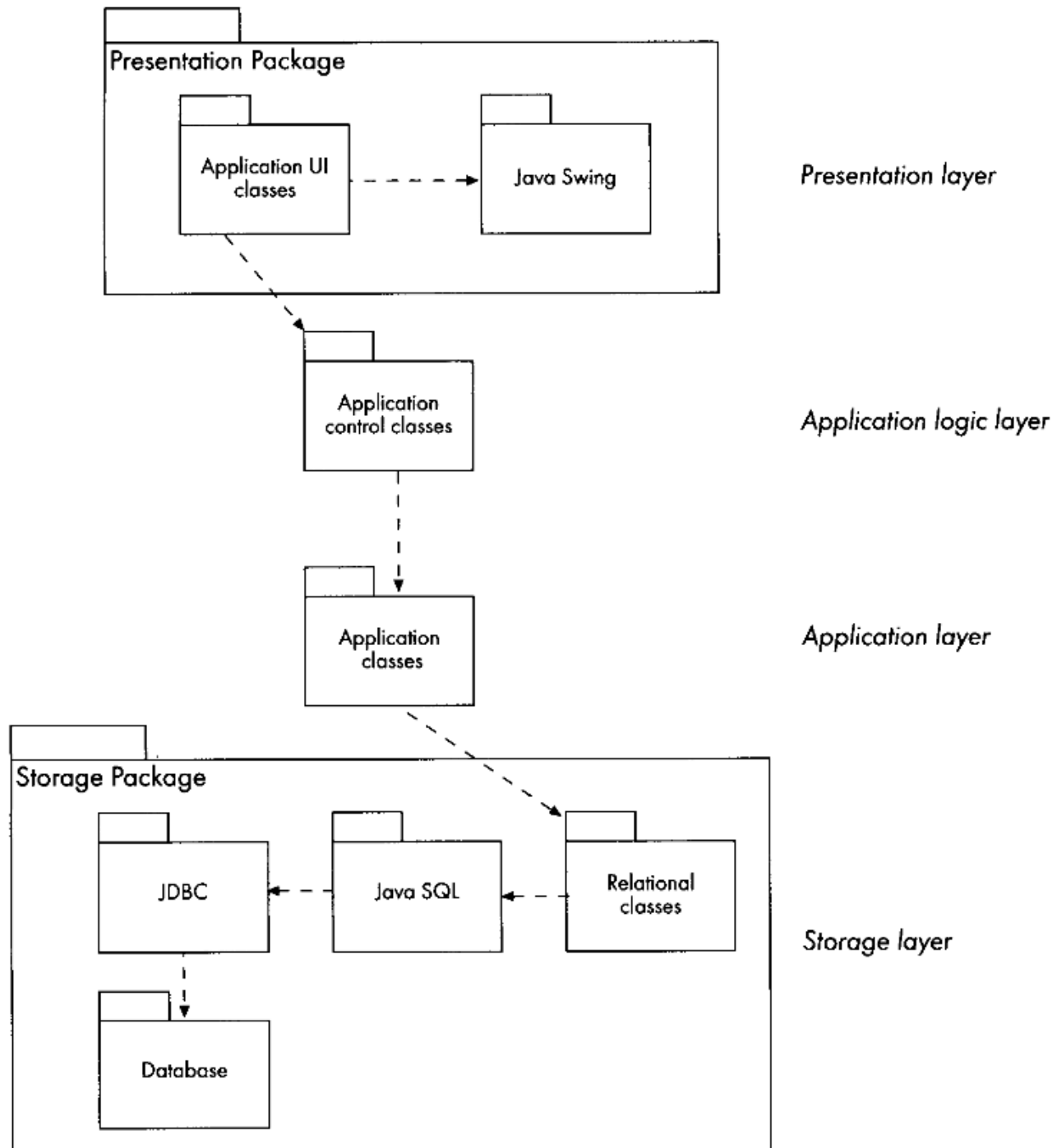
Phân tầng (layering)

- Hệ thống có thể phân hoạch thành các tầng (layers)
- Phân tích: tập trung vào nội dung bài toán (problem domain)
- Thiết kế: bổ sung thêm thông tin về môi trường hệ thống
 - User interface, Data management ...
 - Có thể phân chia các thành phần kiến trúc trên thành các tầng
 - Ý tưởng phân tầng bắt nguồn từ ngôn ngữ Smalltalk với kiến trúc MVC (Model – View – Controller)

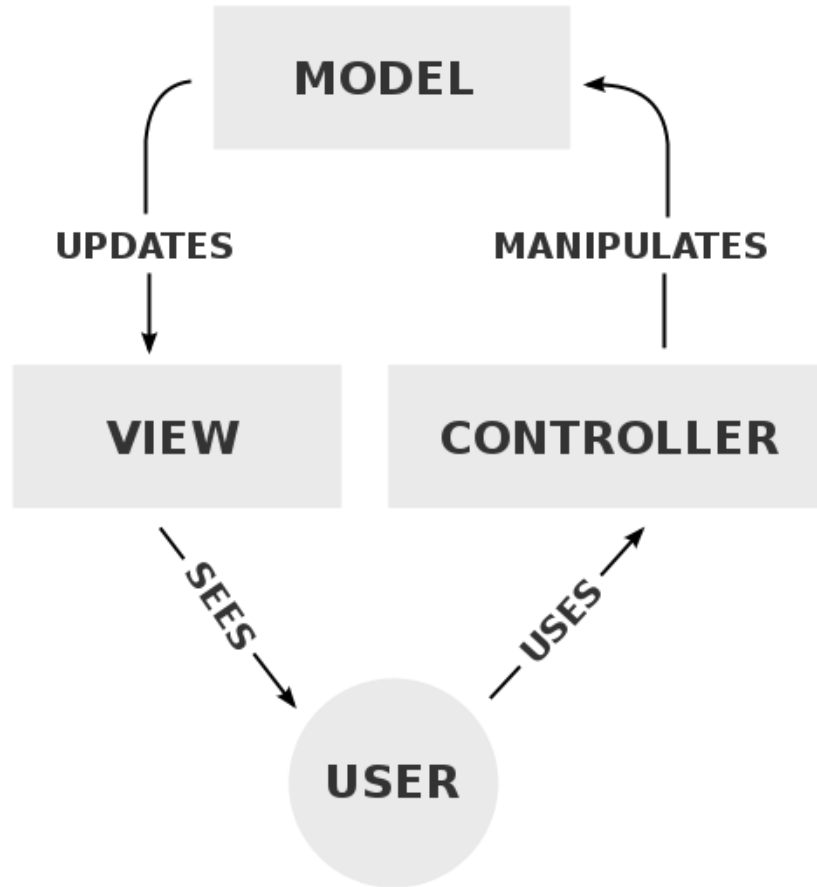
Ví dụ

Một kiến trúc 4-tầng





Kiến trúc MVC

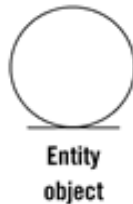
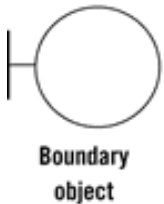


Các loại lớp

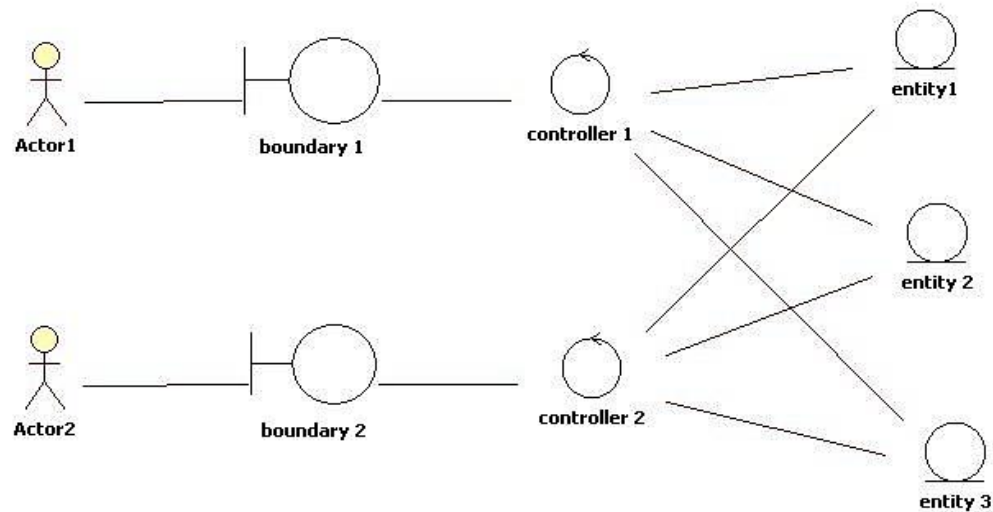
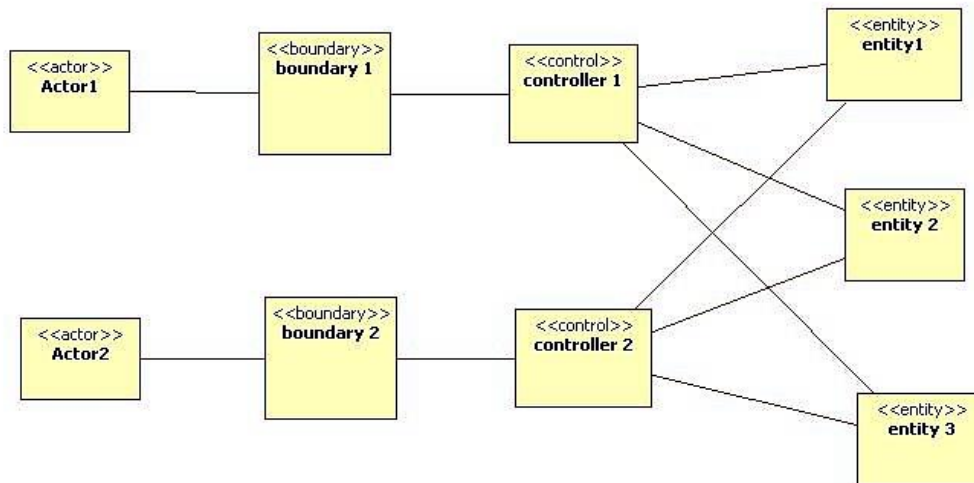
- Phân tích: làm rõ hệ thống cần những gì (what?)
 - Lớp được phát hiện là các lớp thực thể (entity)
- Thiết kế: làm sao để hoàn thành những yêu cầu trên (how?)
 - Bổ sung các lớp biên (boundary), lớp điều khiển (control)
- Cần xác định các lớp biên và lớp điều khiển cho mỗi ca sử dụng
 - Lớp thực thể: thực thể của hệ thống (Customer, Bike...)
 - Lớp biên: giao diện của hệ thống (màn hình, form, webpage...)
 - Lớp điều khiển: thực hiện các thao tác kết nối giữa các lớp biên và các lớp thực thể.

Biểu diễn các loại lớp

- Dùng khuôn mẫu (stereotype)
 - Lớp biên: <<boundary>>
 - Lớp thực thể: <<entity>>
 - Lớp điều khiển: <<control>>
- Dùng ký hiệu



Ví dụ

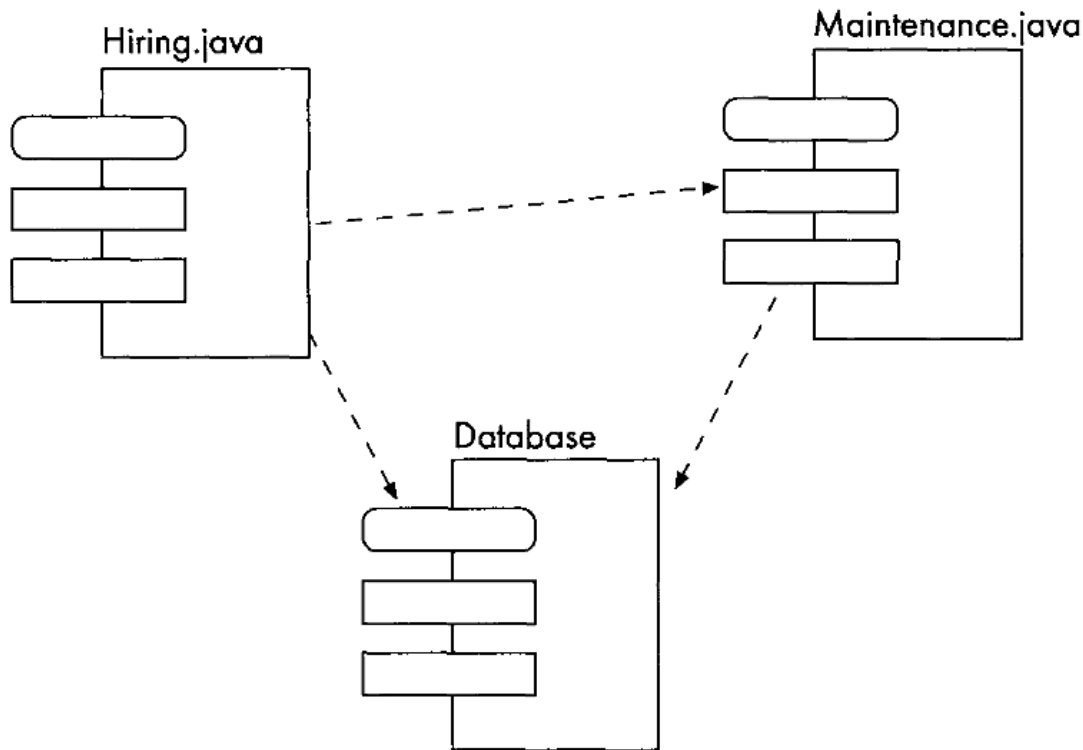


Quy tắc giao tiếp giữa các đối tượng

- Tác nhân chỉ có thể giao tiếp với các đối tượng biên
- Đối tượng biên chỉ có thể giao tiếp với các đối tượng điều khiển và các tác nhân
- Đối tượng thực thể chỉ có thể giao tiếp với đối tượng điều khiển
- Đối tượng điều khiển có thể giao tiếp với các đối tượng biên và thực thể, và các đối tượng điều khiển khác, nhưng không được với các tác nhân

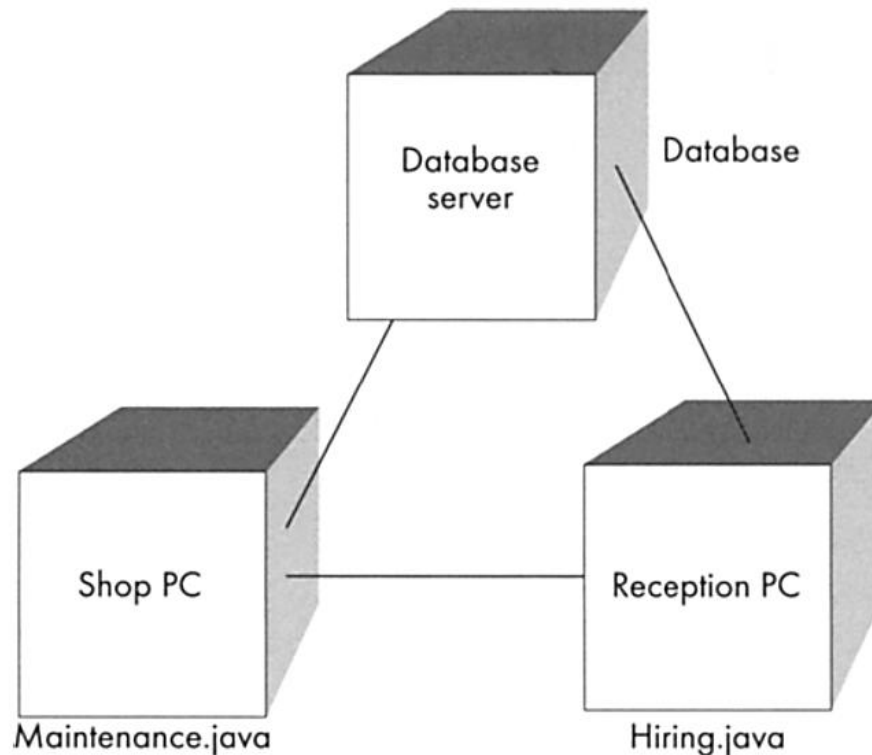
Biểu đồ thành phần (Component diagram)

- Mô tả các thành phần phần mềm và sự phụ thuộc giữa chúng
- Thành phần có thể là các gói hoặc các file (source, binary, data)



Biểu đồ triển khai (Deployment diagram)

- Mô tả sự bố trí các thành phần phần cứng của hệ thống
- Thành phần: Máy chủ, máy trạm, máy in...



Câu hỏi

1. Nêu khác biệt giữa mô hình phân tích và mô hình thiết kế?
2. Kiến trúc phân tầng (layered architecture) là gì?
3. Biểu đồ gói (package diagram) dùng làm gì?
4. Sự phụ thuộc (dependency) giữa các gói là gì?
5. Biểu đồ thành phần (component diagram) dùng làm gì?
6. Biểu đồ triển khai (deployment diagram) dùng làm gì?

Bài tập

1. We have grouped classes into a Stock package and an Ordering package as follows:

<i>Stock package</i>	<i>Ordering package</i>
Product	Customer
Supplier	Order
	OrderLine

Each :Product knows its price. Each :OrderLine has a numberOfItems attribute and is linked to a :Product. Each :OrderLine must work out its line cost by asking for the price from the appropriate :Product and multiplying this by the numberOfItems. What is the dependency between the two packages?