

## Unified Modeling Language

Tổng quan về Phân tích và Thiết kế

Giảng viên: Mai Thúy Nga



### Nội dung môn học

Giới thiệu về PTTK hướng đối tượng với UML 1	Khái niệm về Hướng đối tượng trong UML 2	Đặc tả Yêu cầu với mô hình Use Case I 3
Đặc tả Yêu cầu với mô hình Use Case II 4	Tổng quan về Phân tích và Thiết kế 5	Phân tích Use Case I 6
Phân tích Use Case II 7	Mô hình hóa Thiết kế 8	Ôn tập 9

Analysis and Design Overview

5 - 3

### Nội dung trình bày

- ☆ Tổng quan về phân tích và thiết kế
  - ♦ Khái niệm về kiến trúc phần mềm
  - ♦ Mức độ trừu tượng hóa trong phân tích

Analysis and Design Overview

5 - 4

### Mục đích của Phân tích và Thiết kế

- ♦ Chuyển các yêu cầu của bài toán thành một bản thiết kế của hệ thống sẽ được xây dựng
- ♦ Xây dựng kiến trúc ở mức độ cơ bản cho hệ thống
  - Kiến trúc thể hiện sự tương tác giữa các hệ thống con, thành phần có trong hệ thống
- ♦ Chọn lựa thiết kế để phù hợp với môi trường phát triển
  - Là thiết kế khả thi cho từng hệ thống con, thành phần của kiến trúc
  - Ở mức chi tiết, thiết kế sẽ phụ thuộc vào nền tảng, ngôn ngữ lập trình, hay cơ sở dữ liệu

Analysis and Design Overview

5 - 5

---

---

---

---

---

---

---

### Phân tích và Thiết kế hướng đối tượng (OOAD) (1)

- ♦ Tập trung vào quá trình phân tích các YÊU CẦU của hệ thống và thiết kế các MÔ HÌNH cho hệ thống đó trước giai đoạn lập trình
- ♦ Được thực hiện nhằm đảm bảo mục đích và yêu cầu của hệ thống được ghi lại một cách hợp lý trước khi hệ thống được xây dựng
- ♦ Cung cấp các mô hình một cách chi tiết thể hiện hệ thống sẽ được xây dựng như thế nào
- ♦ Các mô hình này là quá trình trừu tượng hóa của từng phần có trong hệ thống
- ♦ Cung cấp cho người dùng, khách hàng, kỹ sư phân tích, thiết kế nhiều cái nhìn khác nhau về cùng một hệ thống

Analysis and Design Overview

5 - 6

---

---

---

---

---

---

---

### Phân tích và Thiết kế hướng đối tượng (OOAD) (2)

- ♦ OOAD được chia thành 2 giai đoạn
  - Phân tích hướng đối tượng (OOA)
  - Thiết kế hướng đối tượng (OOD)
- ♦ OOA là giai đoạn nhằm tạo ra các mô hình cơ bản (mô hình khái niệm) của hệ thống dựa theo những gì khách hàng yêu cầu về hệ thống của họ
  - Đứng nhiều trên quan điểm của khách hàng
- ♦ OOD sẽ bổ sung thêm các thông tin thiết kế chi tiết cho các mô hình nói trên
  - Đứng trên quan điểm của nhóm phát triển

Analysis and Design Overview

5 - 7

---

---

---

---

---

---

---

## Sự liên quan của PTTK với các nguyên lý khác

- ♦ Nguyên lý PTTK có quan hệ mật thiết đến các nguyên lý khác trong quá trình phát triển phần mềm:
  - “Mô hình hoá bài toán” cung cấp một phương pháp để thể hiện hệ thống
  - “Khảo sát yêu cầu” cung cấp đầu vào chính cho PTTK
  - “Cài đặt” sẽ thực thi các thiết kế
  - “Kiểm thử” đảm bảo kiểm tra hệ thống theo thiết kế đặt ra

Analysis and Design Overview

5 - 8

---

---

---

---

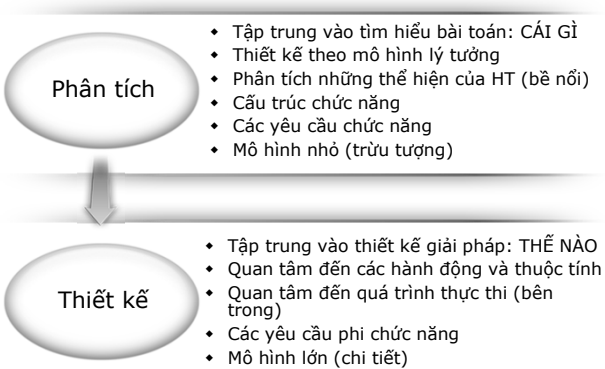
---

---

---

---

## Phân biệt giữa Phân tích và Thiết kế



Analysis and Design Overview

5 - 9

---

---

---

---

---

---

---

---

## Quá trình chuyển đổi mô hình PTTK (1)

### Sự chuyển đổi các mô hình

Yêu cầu	Phân tích	Thiết kế	Lập trình	Kiểm thử
<ul style="list-style-type: none"> <li>• Các tính năng</li> <li>• Mô hình nghiệp vụ</li> <li>• Use cases</li> <li>• Thể hiện bên ngoài</li> <li>• Quan điểm KH</li> </ul>	<ul style="list-style-type: none"> <li>• Kiến trúc tổng thể</li> <li>• Mô hình rõ ràng</li> <li>• Use Case thống nhất</li> <li>• Phân tích bên trong</li> <li>• Thực thể chính</li> <li>• Quan điểm LTV</li> <li>• ...</li> </ul>	<ul style="list-style-type: none"> <li>• Kiến trúc chi tiết mức thành phần</li> <li>• Mô hình chi tiết cho từng UC</li> <li>• Công nghệ</li> <li>• Chi tiết các thực thể</li> <li>• Chi tiết &amp; Rõ ràng</li> </ul>	<ul style="list-style-type: none"> <li>• Mã nguồn</li> <li>• Kiểm thử đơn vị</li> </ul>	<ul style="list-style-type: none"> <li>• Các thủ tục kiểm thử</li> <li>• Kiểm thử tích hợp, hệ thống</li> <li>• ...</li> </ul>

- Sự khác nhau giữa PT và TK là mức độ quan tâm và mức độ nhấn mạnh
- Không có ranh giới rõ rệt để phân chia quá trình PT và TK

Analysis and Design Overview

5 - 10

---

---

---

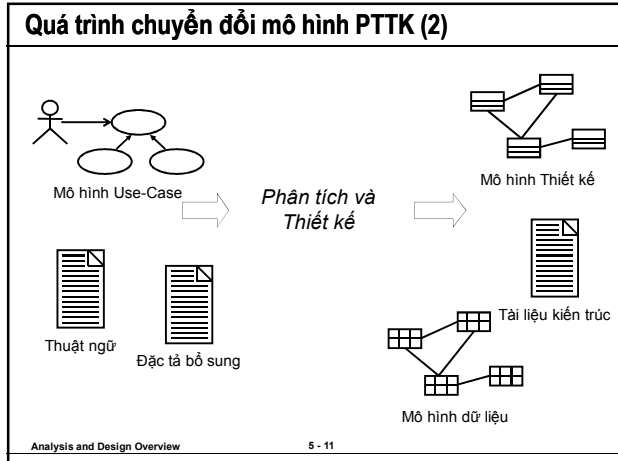
---

---

---

---

---




---

---

---

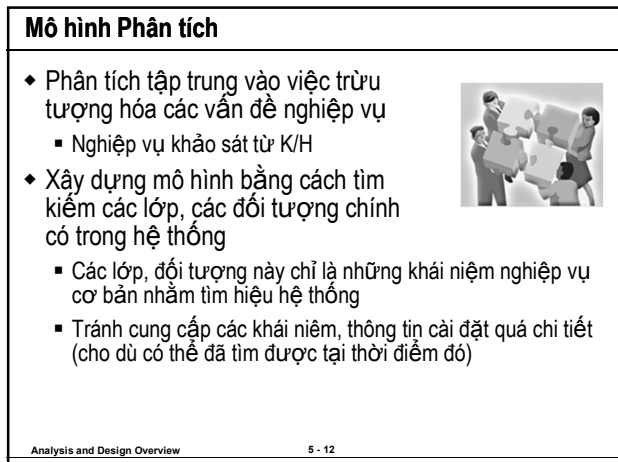
---

---

---

---

---




---

---

---

---

---

---

---

---




---

---

---

---

---

---

---

---

### Nội dung trình bày

- ♦ Tổng quan về phân tích và thiết kế
- ☆ Khái niệm về kiến trúc phần mềm
- ♦ Mức độ trừu tượng hóa trong phân tích

Analysis and Design Overview

5 - 14

---

---

---

---

---

---

---

---

### Kiến trúc phần mềm

- ♦ Kiến trúc phần mềm là một tổ chức hay cơ cấu của các thành phần chính có trong hệ thống
  - Sự lựa chọn các thành phần cấu trúc và giao diện thành phần của nó tương tác với nhau như thế nào
  - Chỉ ra các hành vi của từng thành phần
  - Kết hợp các thành phần nói trên dựa trên cấu trúc và hành vi của chúng để tạo ra hệ thống lớn hơn
  - Thường được kế thừa từ một mẫu kiến trúc nào đó

Analysis and Design Overview

5 - 15

---

---

---

---

---

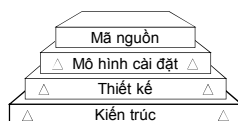
---

---

---

### Tầm quan trọng của mô hình kiến trúc

- ♦ Đưa ra một mô hình mô tả một cách rõ ràng:
  - Hệ thống được phân bố thể nào
  - Từng thành phần của nó hoạt động với nhau như thế nào
- ♦ Là bản mô tả kiến trúc phần mềm dùng để giao tiếp giữa các thành viên phát triển dự án
- ♦ Đưa ra những quyết định về mặt thiết kế rất sớm để dẫn dắt các tác vụ tiếp theo của giai đoạn này



Analysis and Design Overview

5 - 16

---

---

---

---

---

---

---

---

## Các khung nhìn trong quá trình phát triển

- ♦ Quá trình phát triển có sự tham gia của nhiều thành viên
- ♦ Mỗi thành viên có sự quan tâm và cái nhìn khác nhau về hệ thống
- ♦ OOAD xây dựng nhiều loại sơ đồ khác nhau
  - Mỗi loại sơ đồ có một vai trò khác nhau, phản ánh phần mềm ở một khía cạnh nào đó (bỏ qua những thành phần mà nó không quan tâm)
  - Cung cấp cái nhìn cho một, hoặc một nhóm thành viên cụ thể nào đó (chứ không phải tất cả)

Analysis and Design Overview

5 - 17

---

---

---

---

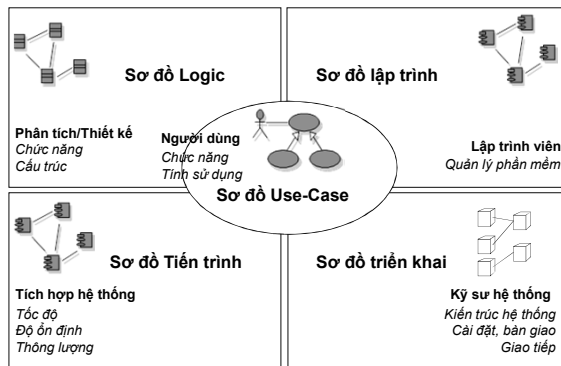
---

---

---

---

## Mô hình “khung nhìn 4+1”



Analysis and Design Overview

5 - 18

---

---

---

---

---

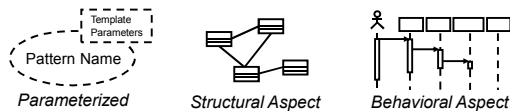
---

---

---

## Mẫu thiết kế (Design Pattern)

- ♦ Mẫu thiết kế là 1 giải pháp cho 1 vấn đề thiết kế chung
  - Mô tả một vấn đề thiết kế chung
  - Mô tả giải pháp cho vấn đề đó
  - Thảo luận về kết quả và sự kết hợp các yếu tố khác nhau khi áp dụng mẫu này
- ♦ Các mẫu thiết kế cung cấp khả năng sử dụng lại các thiết kế sẵn có một cách thích hợp
- ♦ Các mẫu thiết kế nên được mô tả bằng những sơ đồ UML thông thường với một vài tham số nào đó để tùy biến



Analysis and Design Overview

5 - 19

---

---

---

---

---

---

---

---

## Mẫu kiến trúc (Architectural Pattern)

- ♦ Là một cách tổ chức mang tính cấu trúc nền tảng cho một hệ thống phần mềm
- ♦ Cung cấp 1 tập các thành phần bên trong được định nghĩa trước
  - Chỉ rõ vai trò của chúng
  - Chỉ rõ các nguyên tắc và tổ chức các mối quan hệ giữa chúng
- ♦ Một số mẫu thiết kế chung
  - Phân tầng (Layers)
  - Dữ liệu – Thể hiện - Điều khiển (Model-View-Controller)
  - Phân luồng (Pipes and filters)

Analysis and Design Overview

5 - 20

---

---

---

---

---

---

---

## Hướng tiếp cận phân tầng

Chức năng  
cụ thể

Tầng ứng dụng	Các phần mềm ứng dụng cụ thể phục vụ một mục đích nghiệp vụ nào đó
Tầng nghiệp vụ	Cung cấp một loạt các hệ thống cho phép sử dụng lại được trong một nghiệp vụ cụ thể nào đó
Tầng thư viện	Là các hệ thống con sử dụng các lớp thư viện ứng dụng hoặc các dịch vụ khác nhằm phục vụ trong các môi trường tính toán khác nhau
Tầng hệ thống	Là các phần mềm làm nền tảng ứng dụng cho các hệ thống khác như hệ điều hành, các driver cho phần cứng...

Chức năng  
mức hệ thống

Analysis and Design Overview

5 - 21

---

---

---

---

---

---

---

## Các vấn đề trong quá trình phân tầng

- ♦ Mức độ trừu tượng hoá
  - Tìm cách nhóm các thành phần ở cùng 1 mức trừu tượng hoá
- ♦ Phân chia các vấn đề thành từng nhóm
  - Nhóm các thành phần giống nhau vào cùng một nhóm (tầng)
- ♦ Đảm bảo tính cơ động
  - Tính kết hợp tối thiểu
  - Che dấu thông tin
    - Đảm bảo khi bị thay đổi không ảnh hưởng đến thành phần khác
  - Giao diện và các thành phần nghiệp vụ là những nơi có khả năng thay đổi cao nhất

Analysis and Design Overview

5 - 22

---

---

---

---

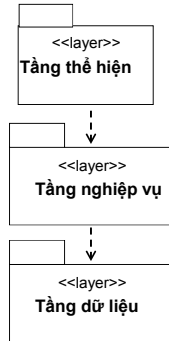
---

---

---

## Mô hình hoá các tầng kiến trúc trong UML

- ♦ Kiến trúc của hệ thống được phân chia thành các tầng riêng biệt
- ♦ Mỗi tầng làm một nhiệm vụ xác định trong hệ thống
- ♦ Các tầng kiến trúc thường được mô hình hoá bằng cách sử dụng các gói
- ♦ Mỗi một gói đại diện cho một tầng và chứa các thành phần bên trong như các lớp, các sơ đồ hoạt động cụ thể cho tầng đó



Analysis and Design Overview

5 - 23

---

---

---

---

---

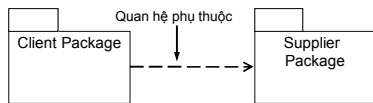
---

---

---

## Các quan hệ gói: Sự phụ thuộc

- ♦ Các gói có thể liên hệ với nhau bằng cách sử dụng mối quan hệ phụ thuộc



- ♦ Sự phụ thuộc:
  - Thay đổi trong gói Supplier sẽ ảnh hưởng đến gói Client
  - Không thể sử dụng gói Client một cách độc lập vì sự phụ thuộc này

Analysis and Design Overview

5 - 24

---

---

---

---

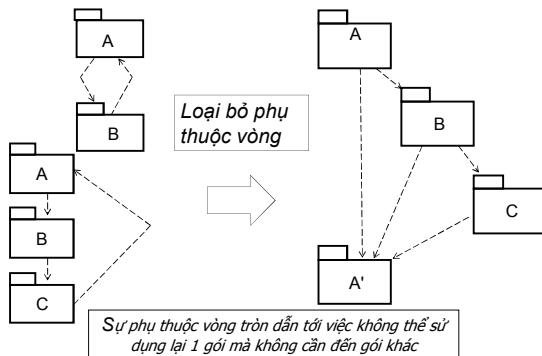
---

---

---

---

## Tránh sự phụ thuộc vòng tròn



Analysis and Design Overview

5 - 25

---

---

---

---

---

---

---

---



### Nội dung trình bày

- ♦ Tổng quan về phân tích và thiết kế
- ♦ Khái niệm về kiến trúc phần mềm
- ☆ Mức độ trừu tượng hóa trong phân tích

Analysis and Design Overview

5 - 26

---

---

---

---

---

---

---

---

### Trừu tượng hoá các vấn đề

- ♦ Trừu tượng hóa một vấn đề trong quá trình phân tích yêu cầu cho phép cung cấp một khái niệm về vấn đề cần được giải quyết
  - Khi mà ta chưa thể biết cụ thể về nó tại giai đoạn đó
- ♦ Các vấn đề chính có thể phải trừu tượng hoá
  - Thông tin, khái niệm nghiệp vụ
  - Các yêu cầu
  - Các thuật ngữ nghiệp vụ
  - Các mô hình nghiệp vụ

Analysis and Design Overview

5 - 27

---

---

---

---

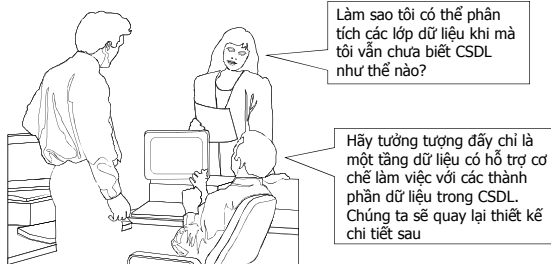
---

---

---

---

### Trừu tượng hóa trong quá trình phân tích



*Trừu tượng hóa trong quá trình phân tích được sử dụng để giảm thiểu sự phức tạp của một hệ thống bằng cách chỉ đưa ra phần thể hiện của vấn đề phức tạp đó và bỏ qua một số chi tiết bên trong nó*

Analysis and Design Overview

5 - 28

---

---

---

---

---

---

---

---

## Trừu tượng hoá với lớp phân tích

- ♦ Tìm cách trừu tượng hoá các thực thể chính (lớp phân tích)
  - Có thể phân loại các lớp phân tích thành các nhóm riêng biệt
- ♦ Chỉ ra các quan hệ giữa các lớp phân tích này
- ♦ Mô hình hoá các lớp phân tích và quan hệ giữa chúng bằng sơ đồ lớp phân tích (Analysis class diagram)
  - Nên mô tả ngắn gọn về các lớp phân tích
- ♦ Có thể coi một lớp phân tích như là một cơ chế phân tích bằng cách trừu tượng hoá một vấn đề phức tạp bằng một lớp phân tích

Analysis and Design Overview

5 - 29

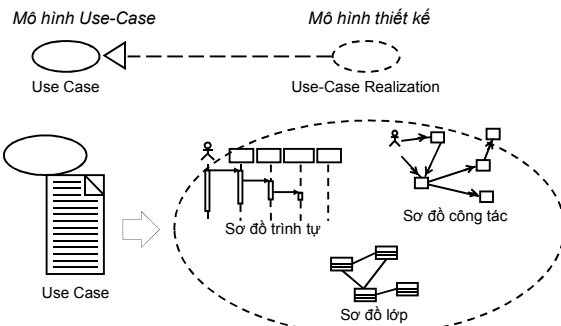
## Các mức độ trừu tượng hóa trong phân tích

- ♦ Cơ chế phân tích (mức khái niệm)
  - Ví dụ: Thành phần lưu trữ dữ liệu
- ♦ Cơ chế thiết kế (mức giải pháp)
  - Ví dụ: Bản vẽ công nghệ: RDBMS hoặc ODBMS
- ♦ Cơ chế cài đặt (mức kỹ thuật)
  - Ví dụ: Công nghệ cụ thể: Oracle, SQL Server

Analysis and Design Overview

5 - 30

## Mức độ chi tiết của mô hình Use-Case



Analysis and Design Overview

5 - 31

### Tổng kết

- ♦ Tổng quan về phân tích và thiết kế
  - Thể hiện ở mức độ tập trung và nhấn mạnh của vấn đề, trả lời câu hỏi CÁI GÌ và NHƯ THẾ NÀO
  - Không có sự phân chia rõ ràng về phân tích và thiết kế
- ♦ Kiến trúc phần mềm
  - Nền tảng cho quá trình xây dựng hệ thống
  - Sử dụng các mẫu kiến trúc có sẵn
- ♦ Mức độ trừu tượng hóa trong phân tích
  - Mức độ tùy thuộc vào ngữ cảnh của quá trình phân tích

Analysis and Design Overview

5 - 32

---

---

---

---

---

---

---