



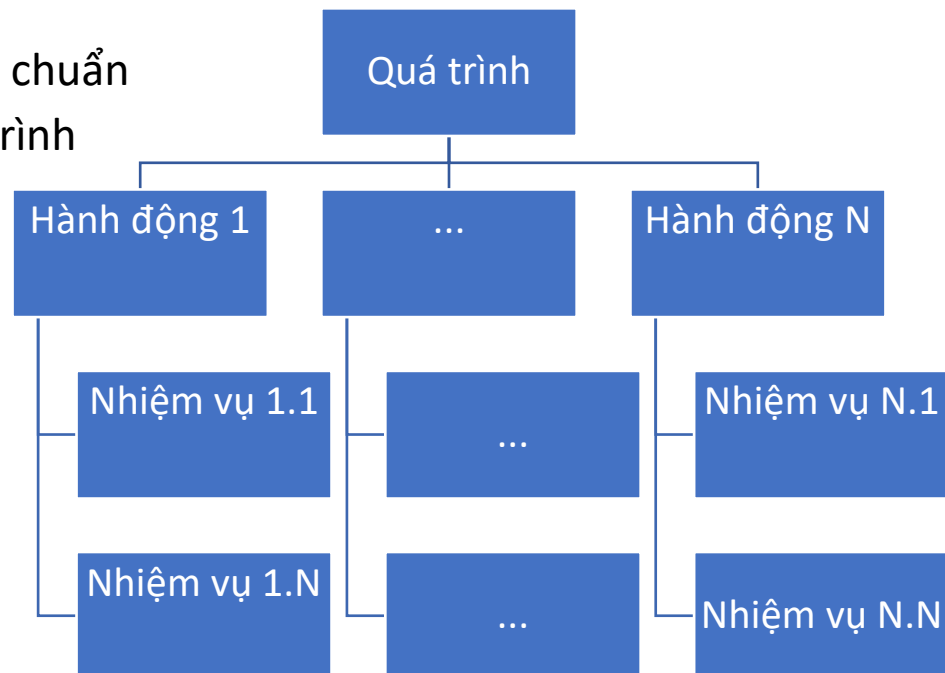
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Phát triển phần mềm theo chuẩn ITSS

Các nội dung chính

Quy trình vòng đời phần mềm

- ❖ Theo “ISO / IEC 12207: 2008, Hệ thống và kỹ thuật phần mềm – Quy trình vòng đời phần mềm”
 - Quy trình phát triển phần mềm tiêu chuẩn quốc tế và mới nhất
- ❖ “Vòng đời bắt đầu với một ý tưởng hoặc một nhu cầu có thể được phần mềm đáp ứng toàn bộ hoặc một phần và kết thúc bằng việc phần mềm ngừng hoạt động”.
- ❖ Các công việc thực hiện tiêu chuẩn
 - Phân cấp như các quy trình



Quy trình vòng đời

❖ “Tiêu chuẩn này nhóm các hoạt động có thể được thực hiện trong vòng đời của hệ thống phần mềm thành 7 nhóm quá trình” [1]*:

1. Các quy trình thỏa thuận: 2 quy trình
2. Các quy trình hỗ trợ dự án của tổ chức: 5 quy trình
3. Quy trình dự án: 7 quy trình
4. Quy trình kỹ thuật: 11 quy trình
5. Quy trình triển khai: 6 quy trình

Mục đích: “để tạo ra một phần tử hệ thống cụ thể được triển khai như một sản phẩm hoặc dịch vụ phần mềm” [1]**.

6. Quy trình hỗ trợ phần mềm: 8 quy trình
7. Quy trình tái sử dụng phần mềm: 3 quy trình

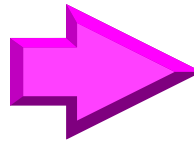
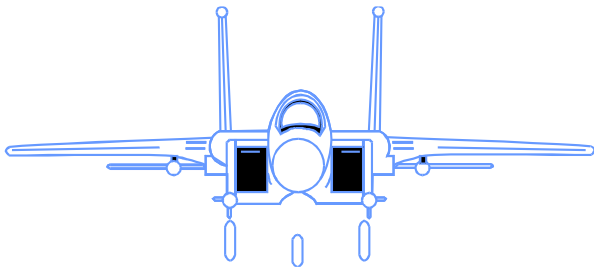
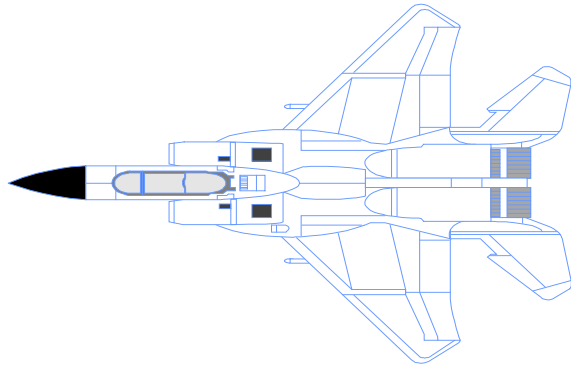
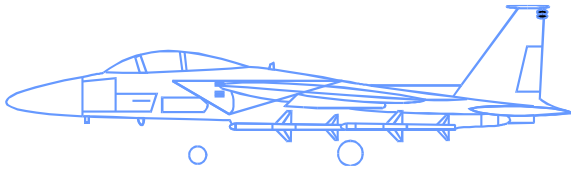
[1]*: clause 5.2.1; pp. 13, [1]**: clause 7.1.1.1; pp. 57,

Tổng kết

- ❖ “Quy trình Vòng đời Phần mềm – SLCP” là các quy trình tiêu chuẩn quốc tế tập trung vào việc phát triển và hỗ trợ Phần mềm Ứng dụng.
- ❖ SLCP có thể được sử dụng như một ngôn ngữ chung giữa các bên liên quan như bên mua và nhà cung cấp. Họ có thể giao tiếp hoặc ra lệnh phát triển phần mềm bằng SLCP. Ví dụ, chúng ta có thể nói “Để đặt hàng quy trình thiết kế chi tiết phần mềm hoặc các quy trình triển khai phần mềm sau này của hệ thống thư viện mới”.

Mô hình hóa

❖ Mô hình là sự đơn giản hóa hệ thống.



Ngôn ngữ mô hình thống nhất (UML)

- ❖ “UML là ngôn ngữ dành cho
 - Hình dung
 - Xác định
 - Cấu tạo
- ❖ Lập hồ sơ tạo tác của một hệ thống sử dụng nhiều phần mềm ”[1].

[1]: Chapter 2, Section 2.1

Lịch sử của UML



UML
Partners'
Expertise

UML 2.0
(2004)

UML 1.5
(March, '03)

UML 1.1
(Sept. '97)

UML 1.0
(Jan. '97)

UML 0.9 and UML 0.91
(June '96) (Oct. '96)

Unified Method 0.8
(OOPSLA '95)

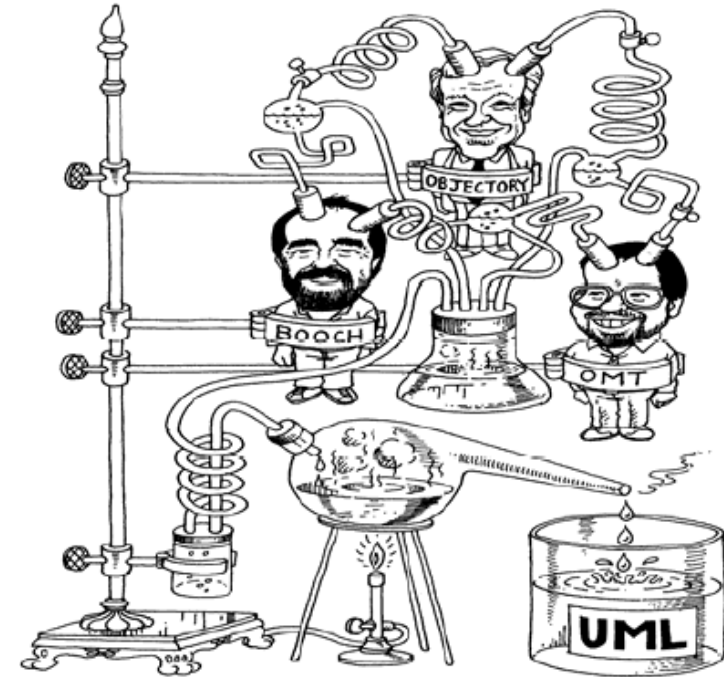
Booch '93

Booch '91

OMT - 1

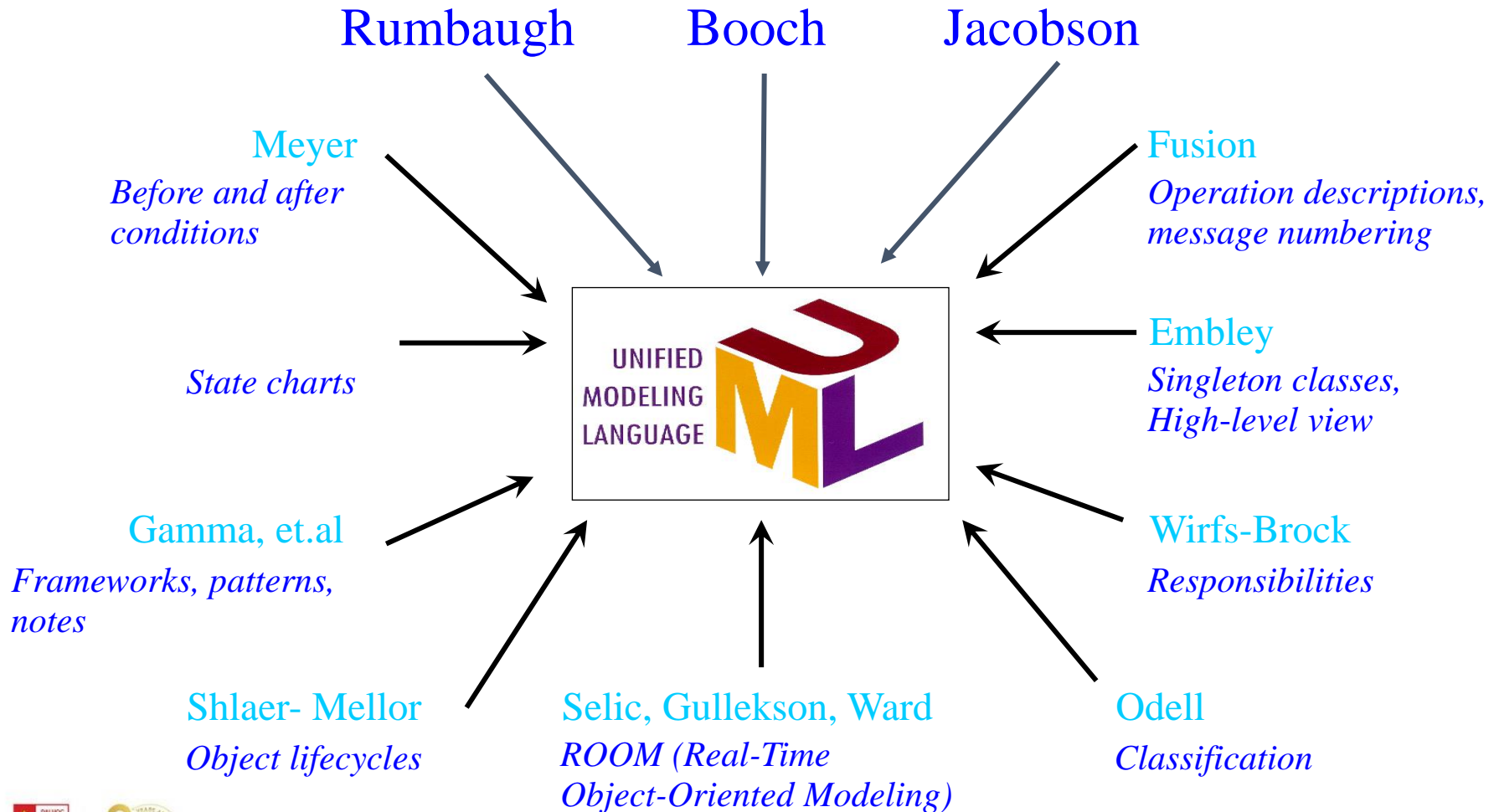
Other
Methods

OOSE



Booch91 (Grady Booch):
Conception, Architecture
OOSE (Ivar Jacobson): Use cases
OMT (Jim Rumbaugh): Analysis

Đầu vào cho UML



Chế độ xem tĩnh so với chế độ xem động

Vai trò của các bộ phận thành phần và cách chúng liên quan với nhau



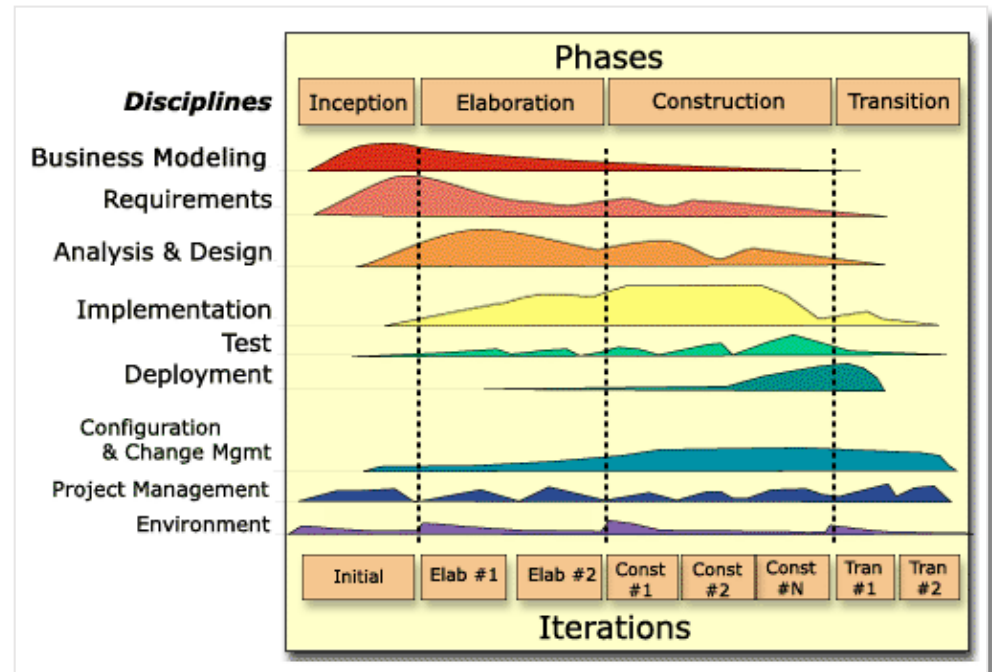
Cách các thành phần tương tác với nhau và / hoặc thay đổi trạng thái nội bộ theo thời gian



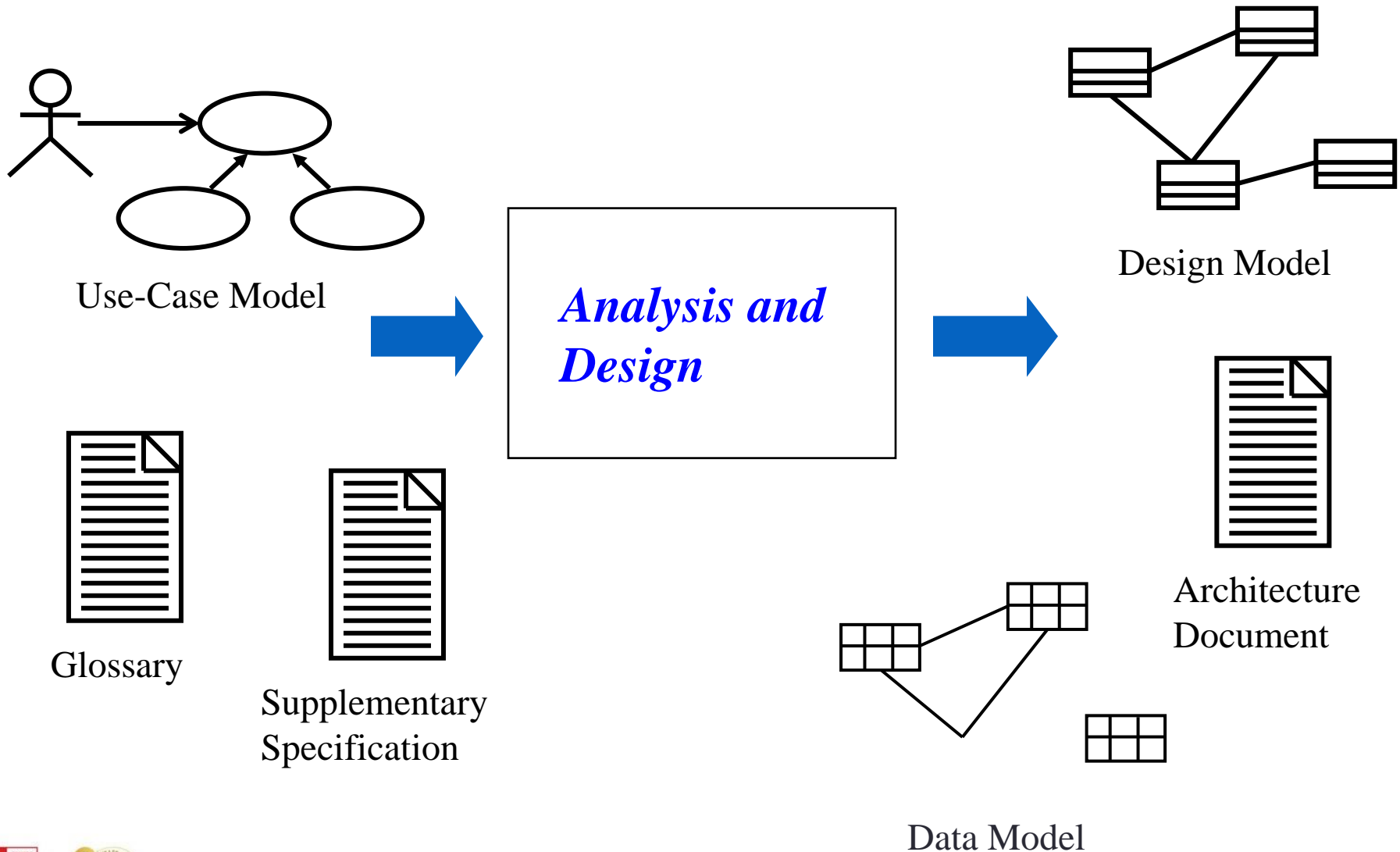
Phân tích và thiết kế hệ thống

Mục đích của Phân tích và Thiết kế là:

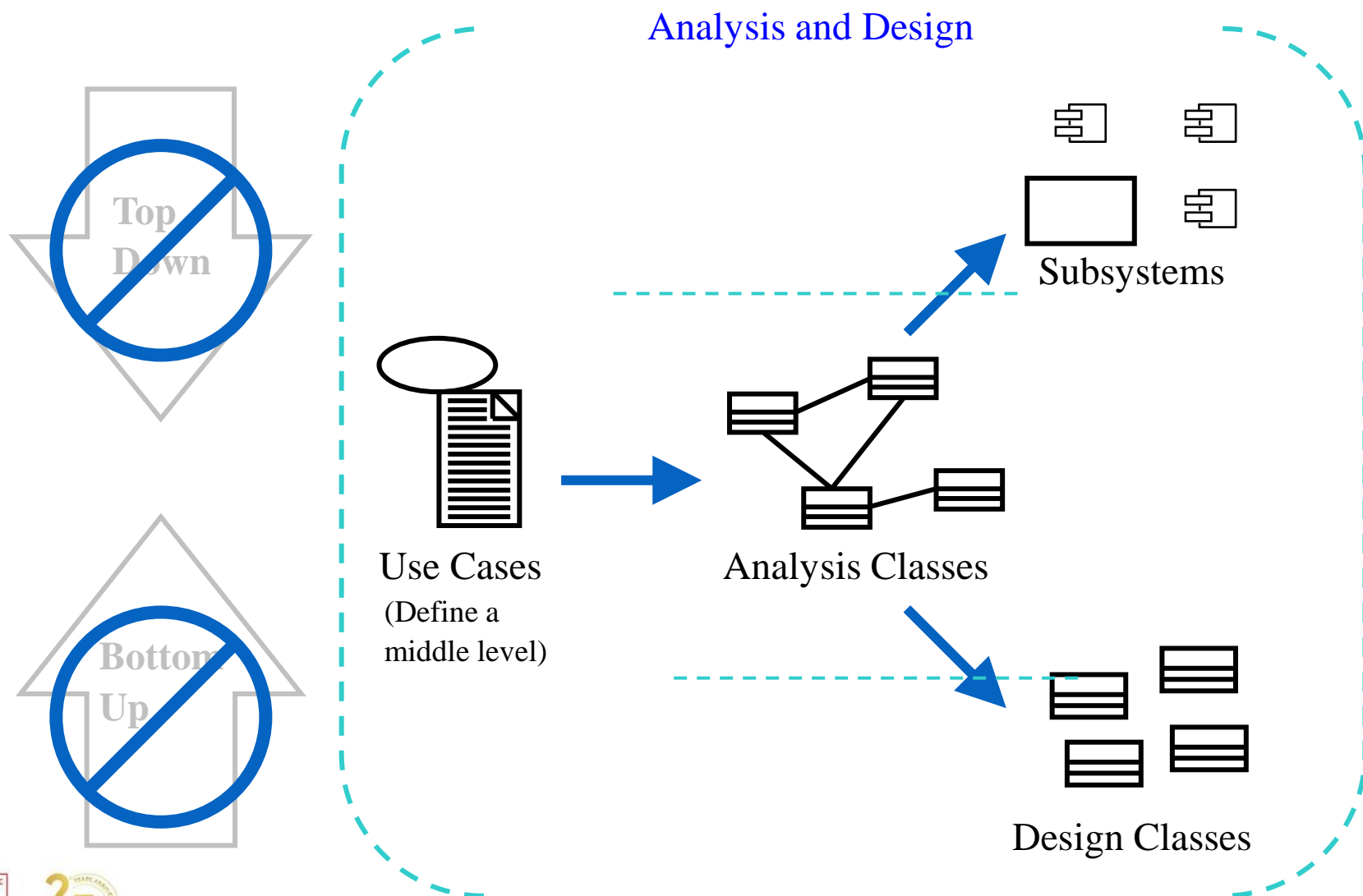
- Chuyển đổi các yêu cầu thành một thiết kế của hệ thống tương lai.
- Phát triển một kiến trúc mạnh mẽ cho hệ thống.
- Điều chỉnh thiết kế để phù hợp với môi trường thực hiện, thiết kế để thực hiện.



Phân tích và thiết kế hướng đối tượng



Phân tích và thiết kế không từ trên xuống hoặc từ dưới lên

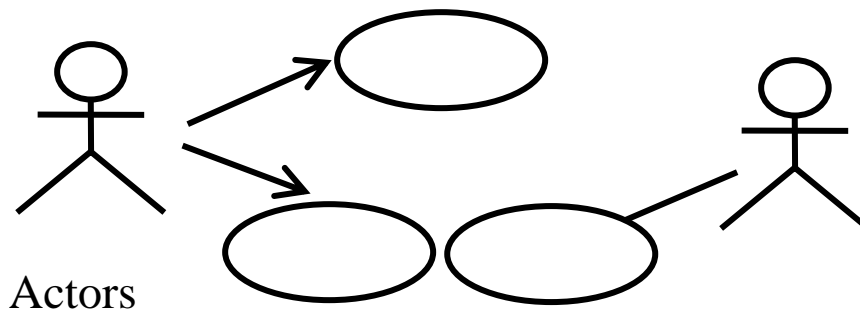


Các bước phân tích và thiết kế

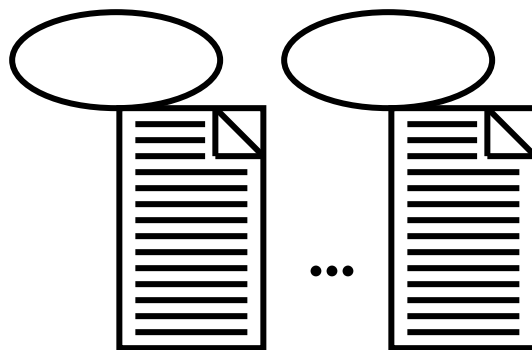
Hoạt động	Bước	Mô tả	Người thực hiện
Xác định kiến trúc ứng viên	1. Phân tích kiến trúc	<ul style="list-style-type: none"> Một lần ở giai đoạn đầu Bỏ qua nếu rủi ro kiến trúc thấp 	Kiến trúc sư
Phân tích hành vi	2. Phân tích ca sử dụng	<ul style="list-style-type: none"> Mỗi trường hợp ca sử dụng 	Nhà thiết kế
Tinh chỉnh kiến trúc	3. Xác định các yếu tố thiết kế	<ul style="list-style-type: none"> Khớp nối và gắn kết Khả năng tái sử dụng 	Kiến trúc sư
	4. Xác định cơ chế thiết kế	<ul style="list-style-type: none"> Mẫu thiết kế 	
	5. Mô tả kiến trúc thời gian chạy	<ul style="list-style-type: none"> Bỏ qua nếu không phải đa luồng Chế độ xem 	
	6. Mô tả phân phối	<ul style="list-style-type: none"> Kiến trúc vật lý 	
Thành phần thiết kế	7. Thiết kế ca sử dụng	<ul style="list-style-type: none"> Mỗi trường hợp sử dụng 	Nhà thiết kế
	8. Thiết kế hệ thống con		
	9. Thiết kế lớp		
Thiết kế DB	10. Thiết kế CSDL		

Mô hình hoá Yêu cầu

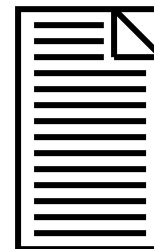
Use-Case Model



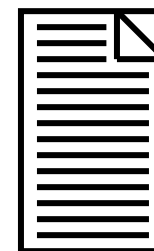
Use Cases



Use-Case Specifications

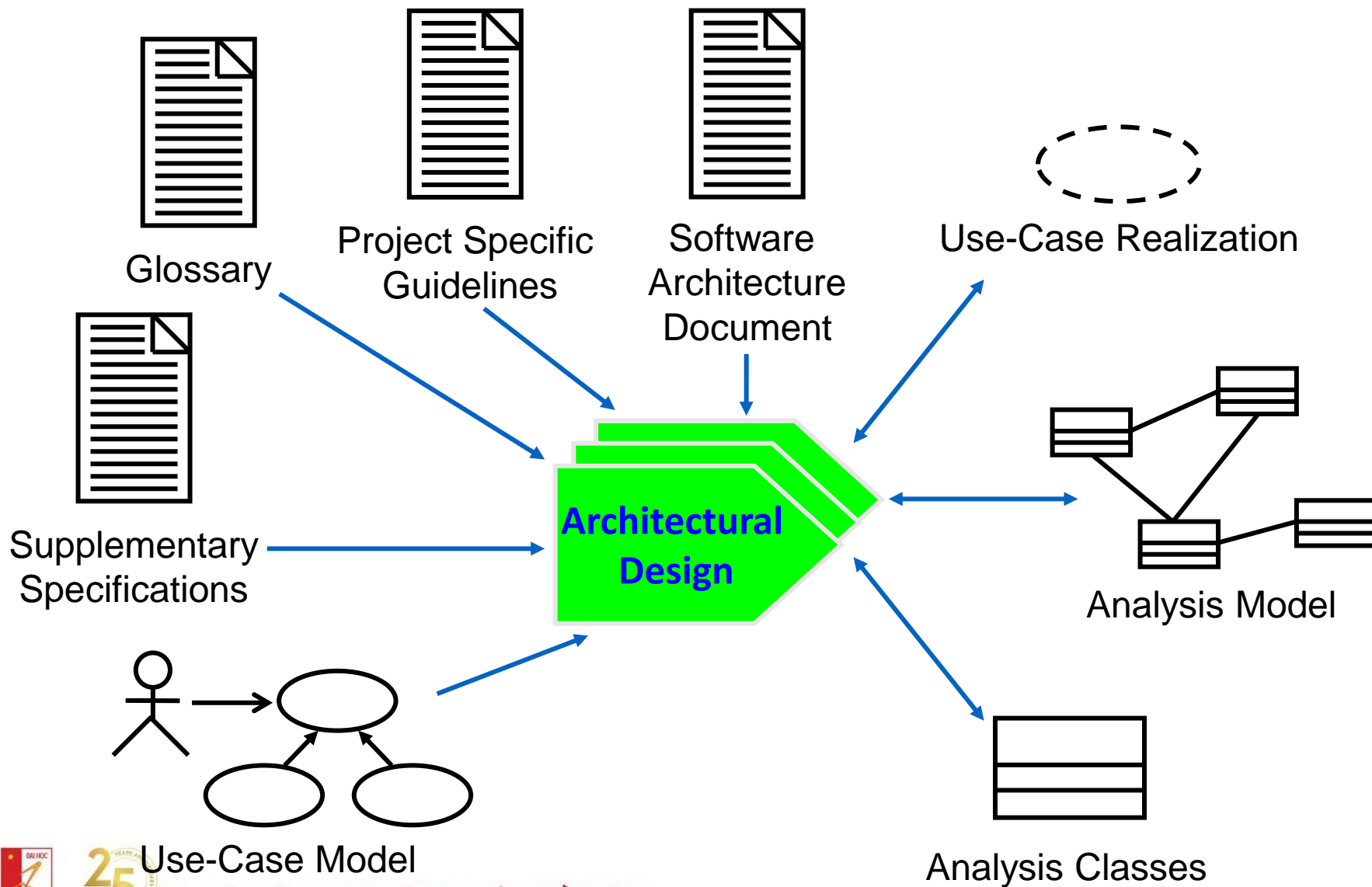


Glossary

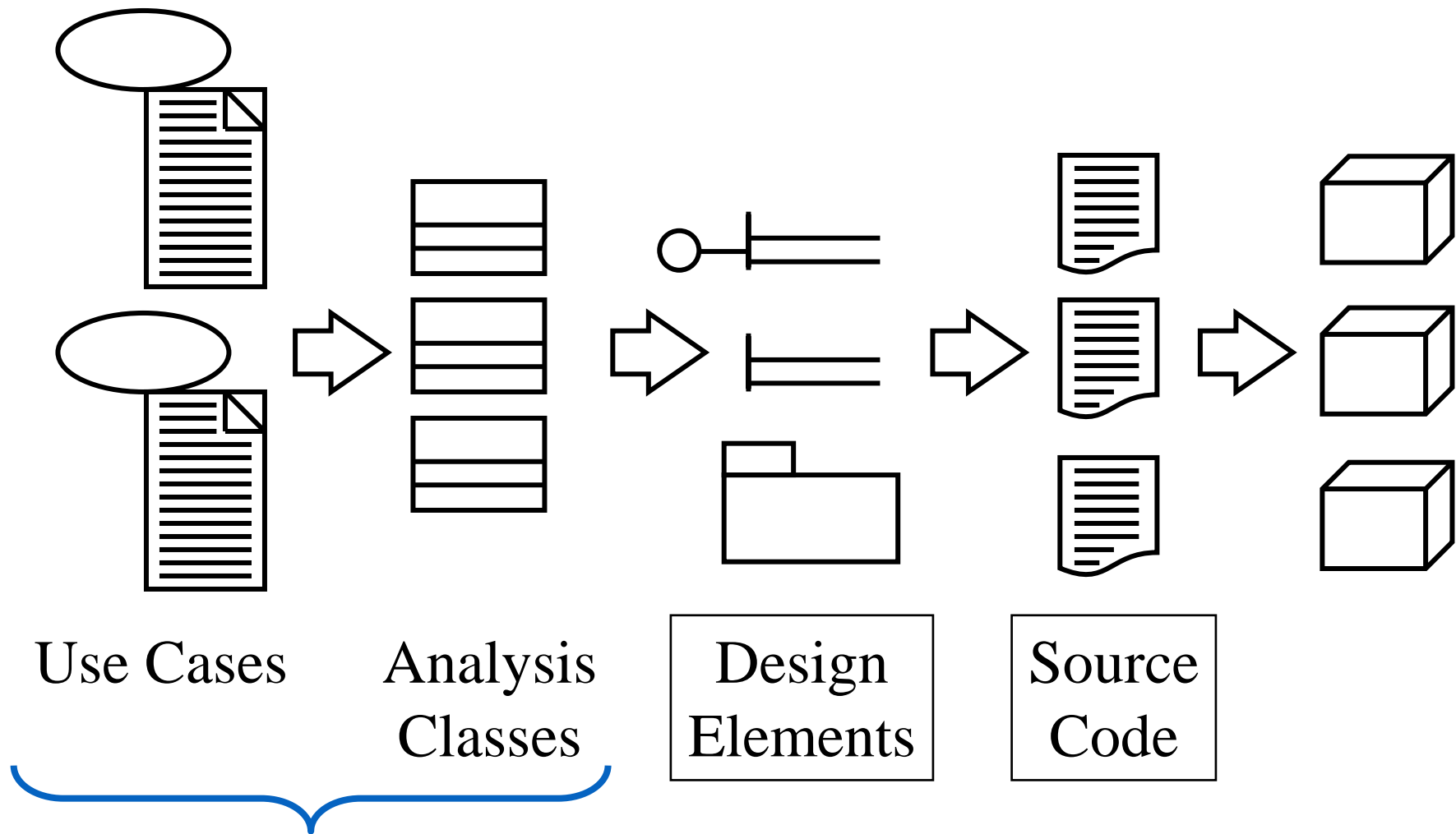


Supplementary
Specification

Thiết kế kiến trúc

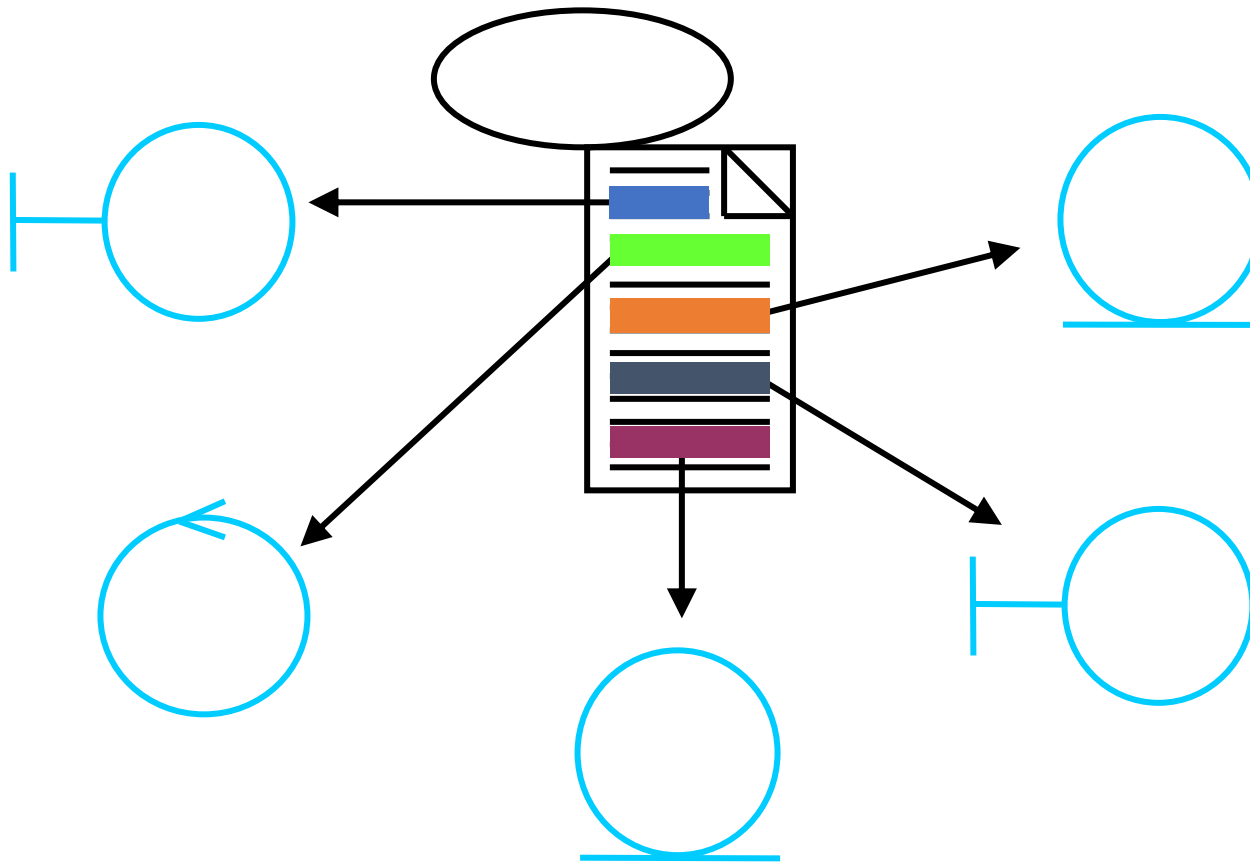


Các lớp phân tích: Bước đầu tiên hướng tới thực thi

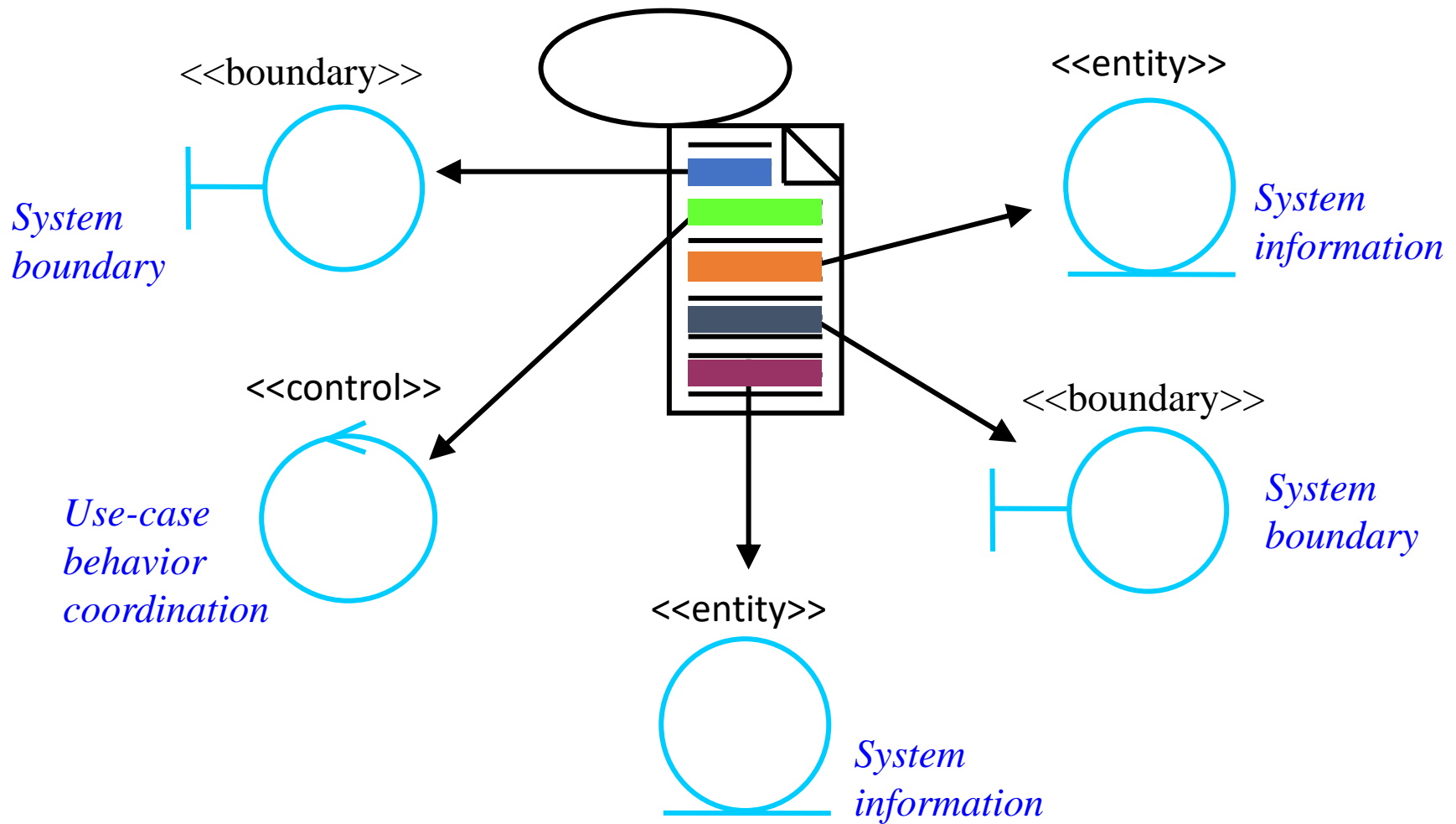


Tìm lớp từ hành vi sử dụng trường hợp

- ❖ Hành vi đầy đủ của một trường hợp sử dụng đã được phân phối cho các lớp học phân tích

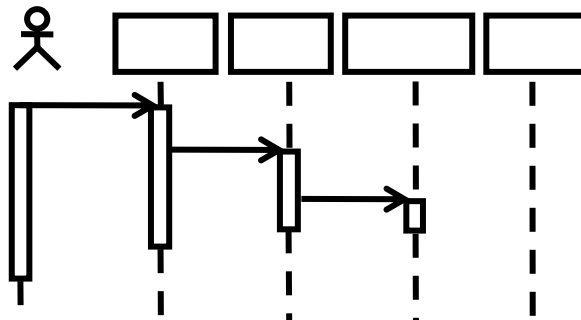


Các loại lớp phân tích



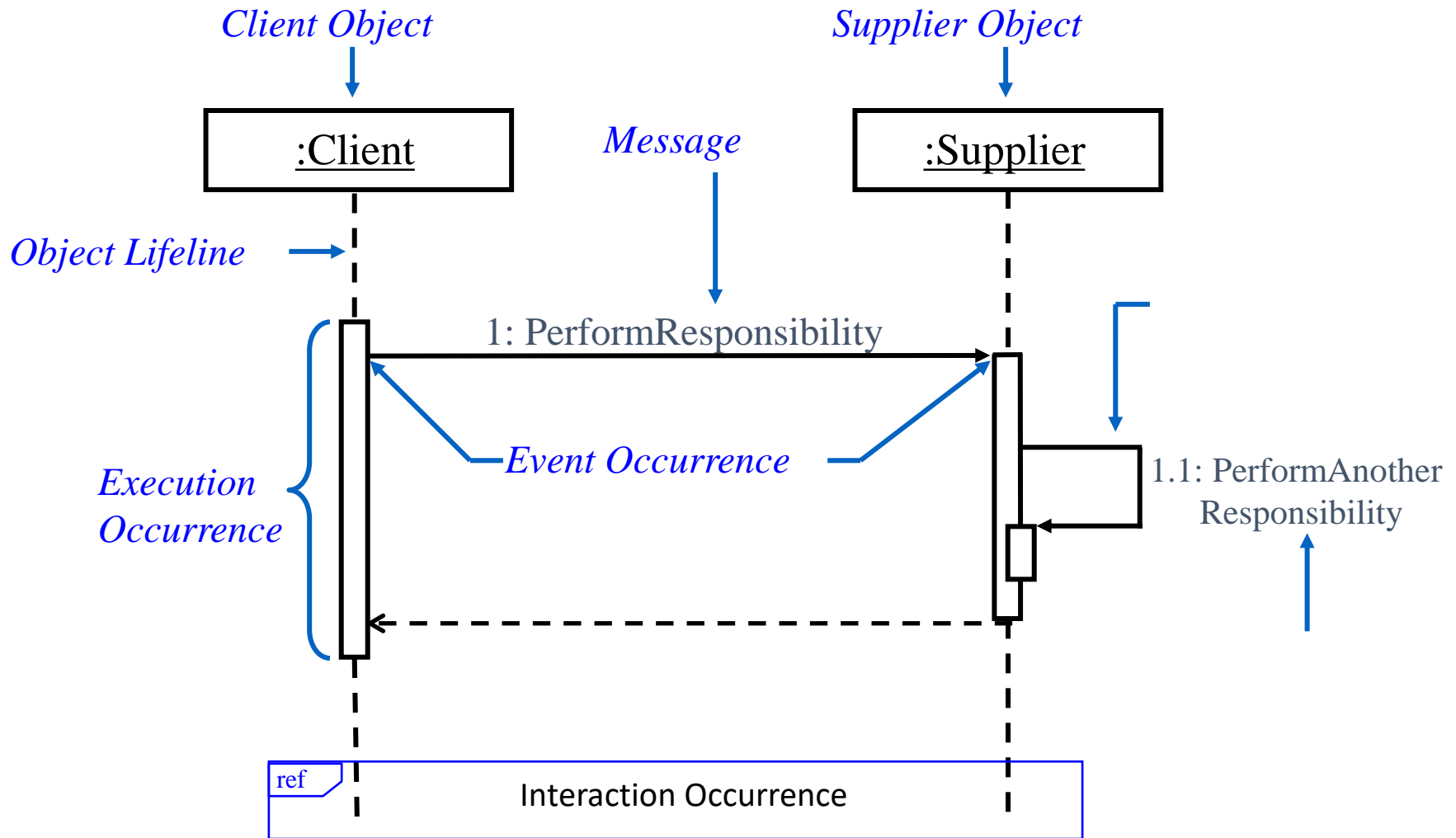
Sơ đồ trình tự

- ❖ Sơ đồ trình tự là một sơ đồ tương tác nhấn mạnh thứ tự thời gian của thư.
- ❖ Sơ đồ hiển thị:
- ❖ Các đối tượng tham gia vào sự tương tác.
- ❖ Chuỗi tin nhắn được trao đổi.



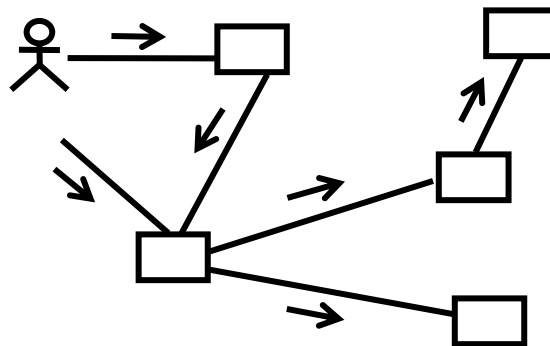
Sequence Diagram

Giải thích sơ đồ trình tự



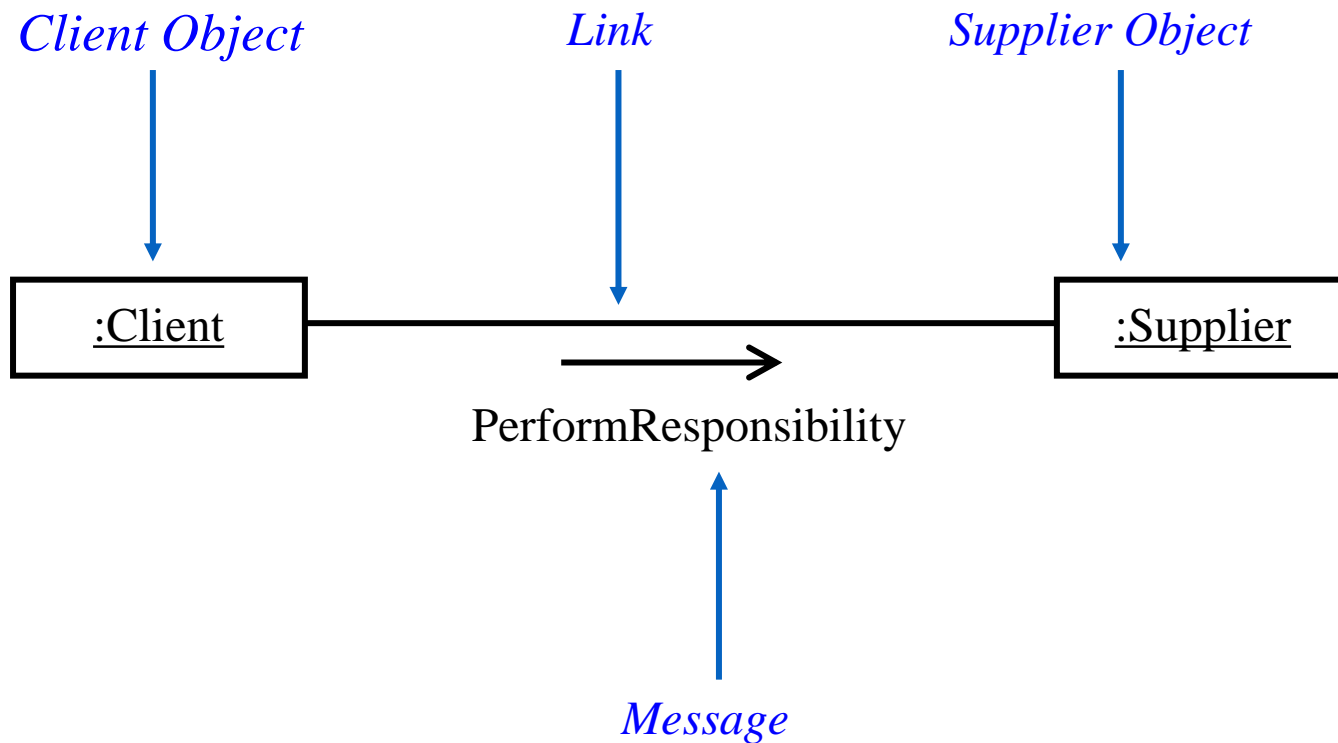
Sơ đồ giao tiếp

- ❖ Sơ đồ giao tiếp nhấn mạnh tổ chức của các đối tượng tham gia vào một tương tác.
- ❖ Sơ đồ giao tiếp cho thấy:
- ❖ Các đối tượng tham gia vào sự tương tác.
- ❖ Liên kết giữa các đối tượng.
- ❖ Thư được chuyển giữa các đối tượng.

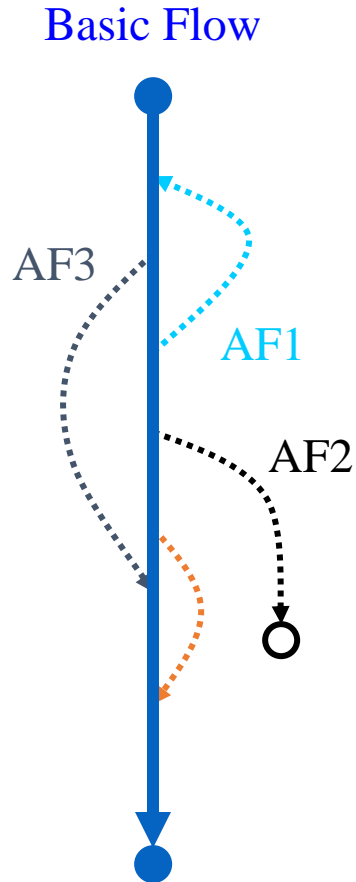


Communication Diagrams

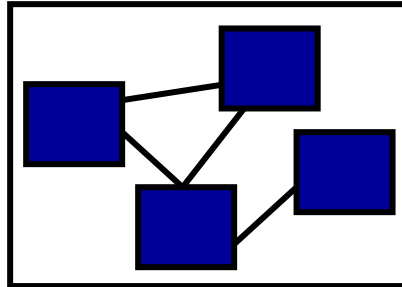
Giải thích sơ đồ giao tiếp



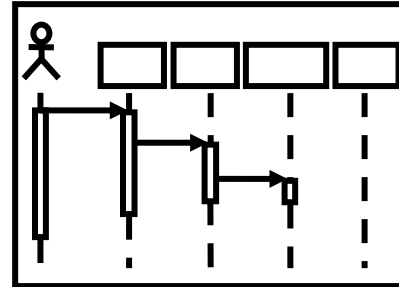
Một sơ đồ tương tác có thể không đủ tốt



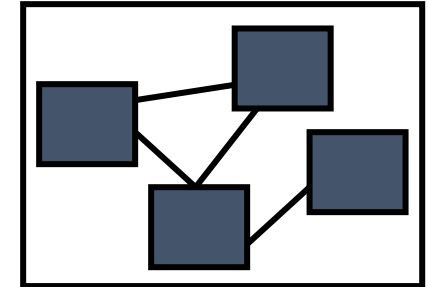
Alternate Flow 1



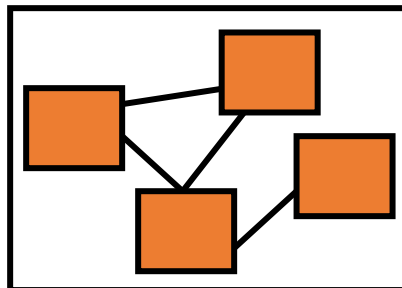
Alternate Flow 2



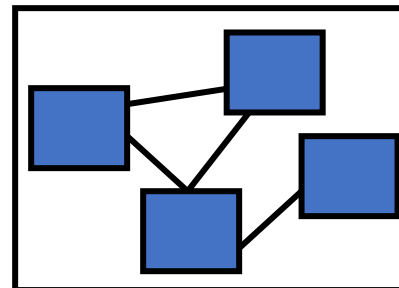
Alternate Flow 3



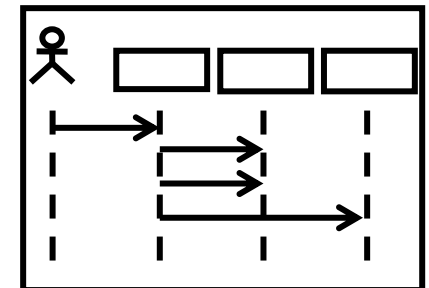
Alternate Flow 4



Alternate Flow 5

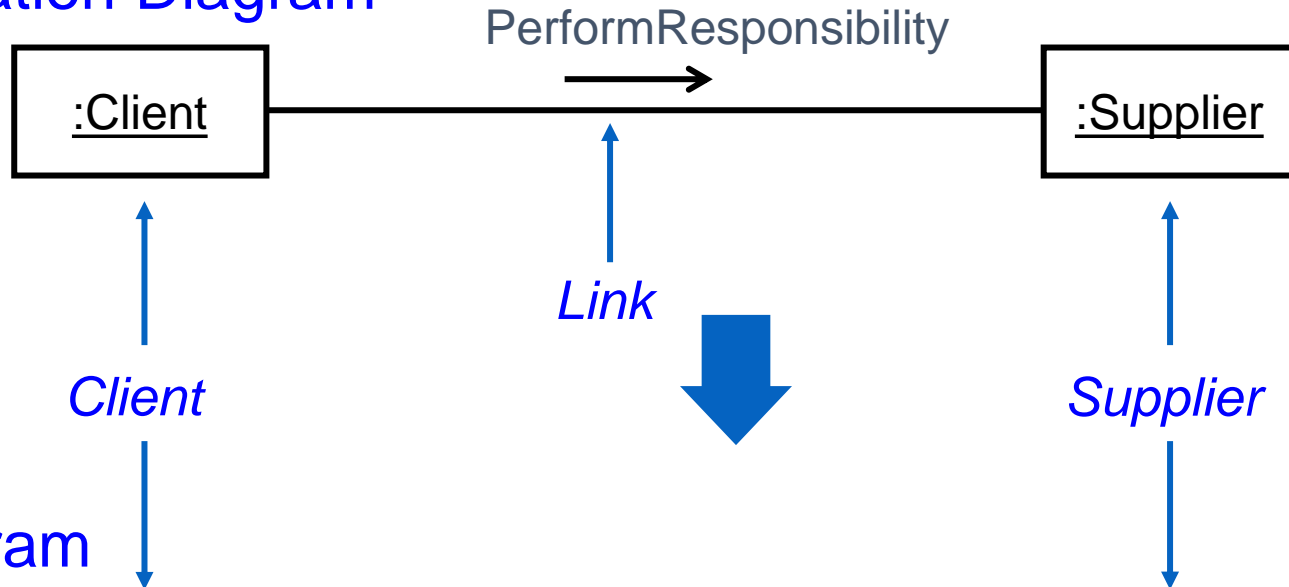


Alternate Flow n

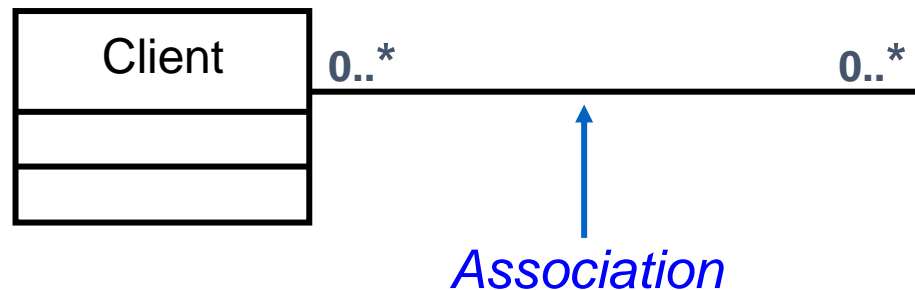


Tìm mối quan hệ

Communication Diagram



Class Diagram



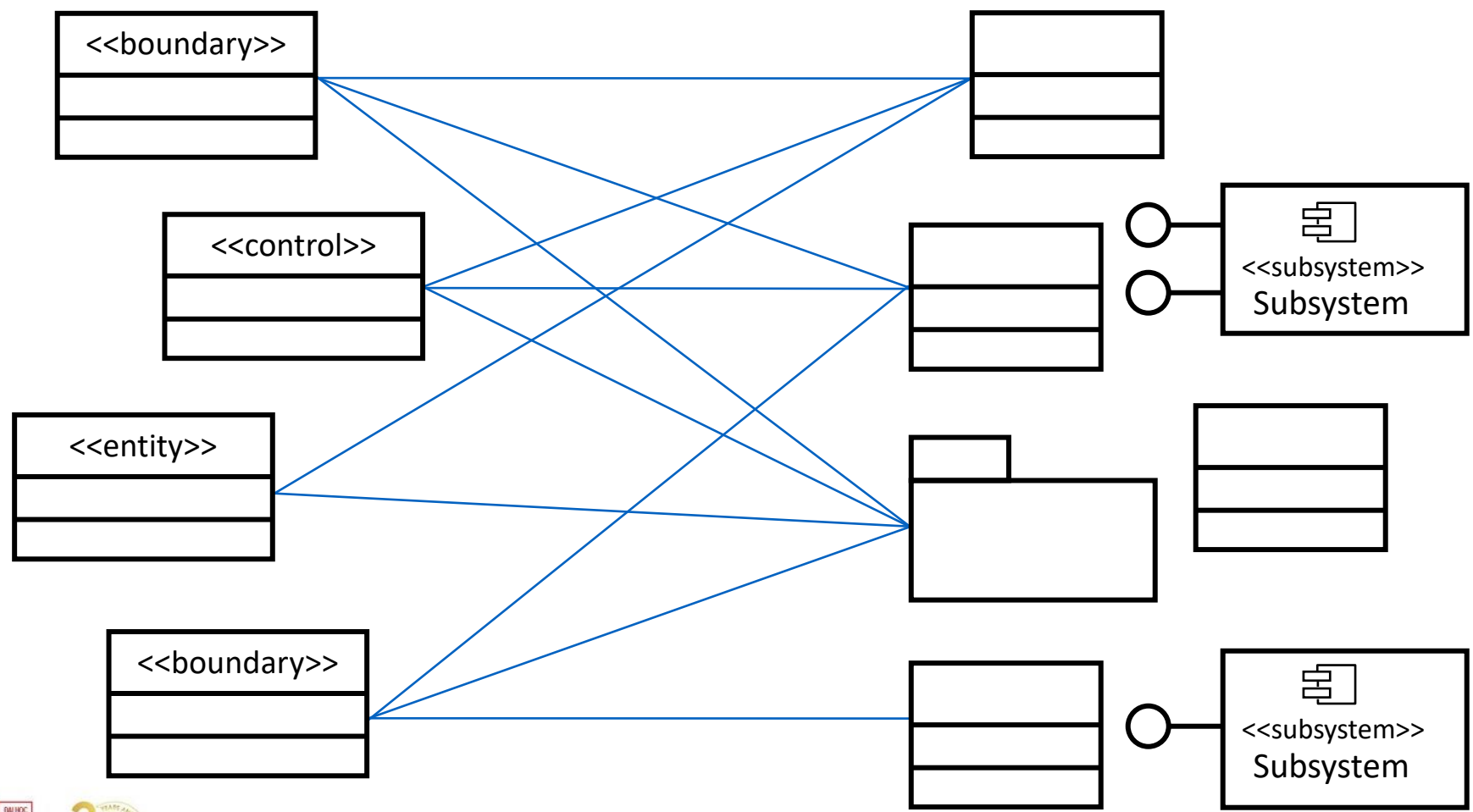
Mục tiêu: Xác định các phần tử thiết kế

- ❖ Định nghĩa mục đích của các phần tử thiết kế và làm rõ vòng đời của chúng được thực hiện khi nào
- ❖ Phân tích sự tương tác của các lớp phân tích và xác định các phần tử mô hình thiết kế => Thiết kế lớp

Từ Lớp phân tích tới Các phần tử thiết kế

Các lớp phân tích

Các phần tử thiết kế



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

Thiết kế giao diện hệ thống/thiết bị

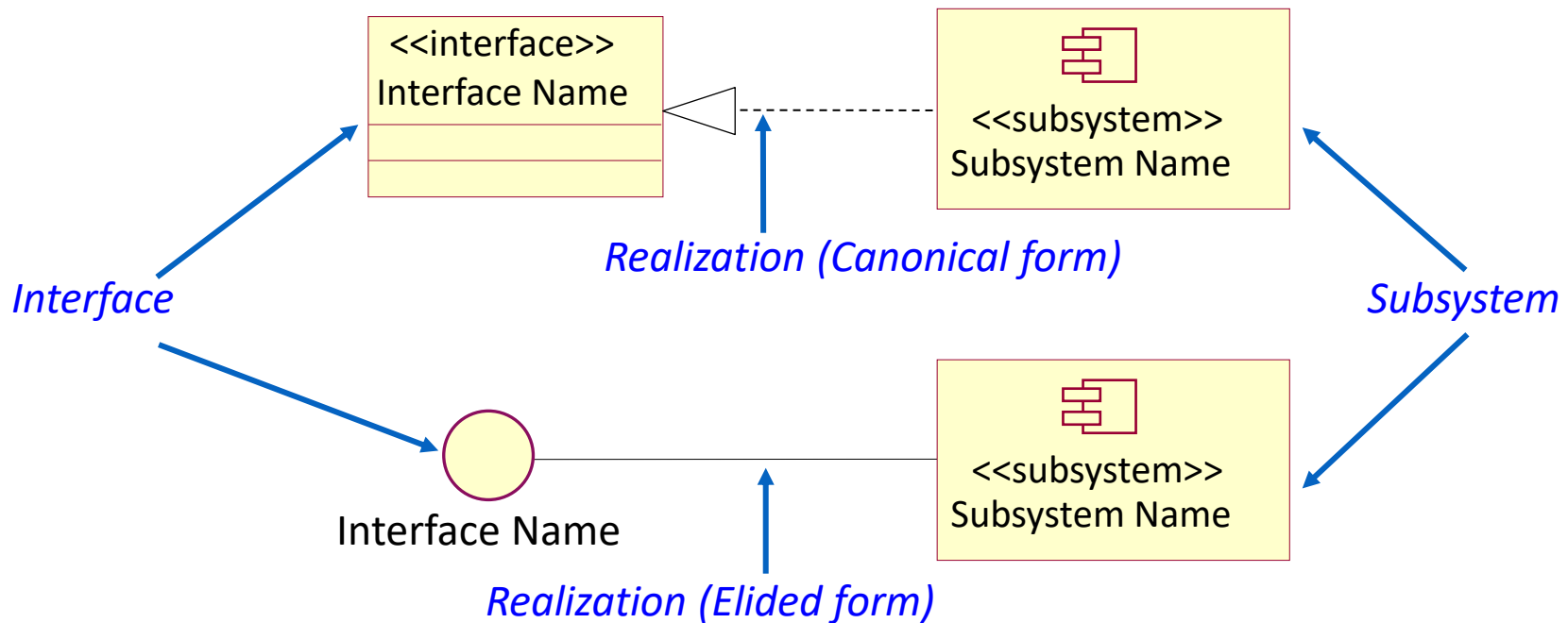
2.1. Xác định hệ thống con

2.2. Xác định giao diện hệ thống con

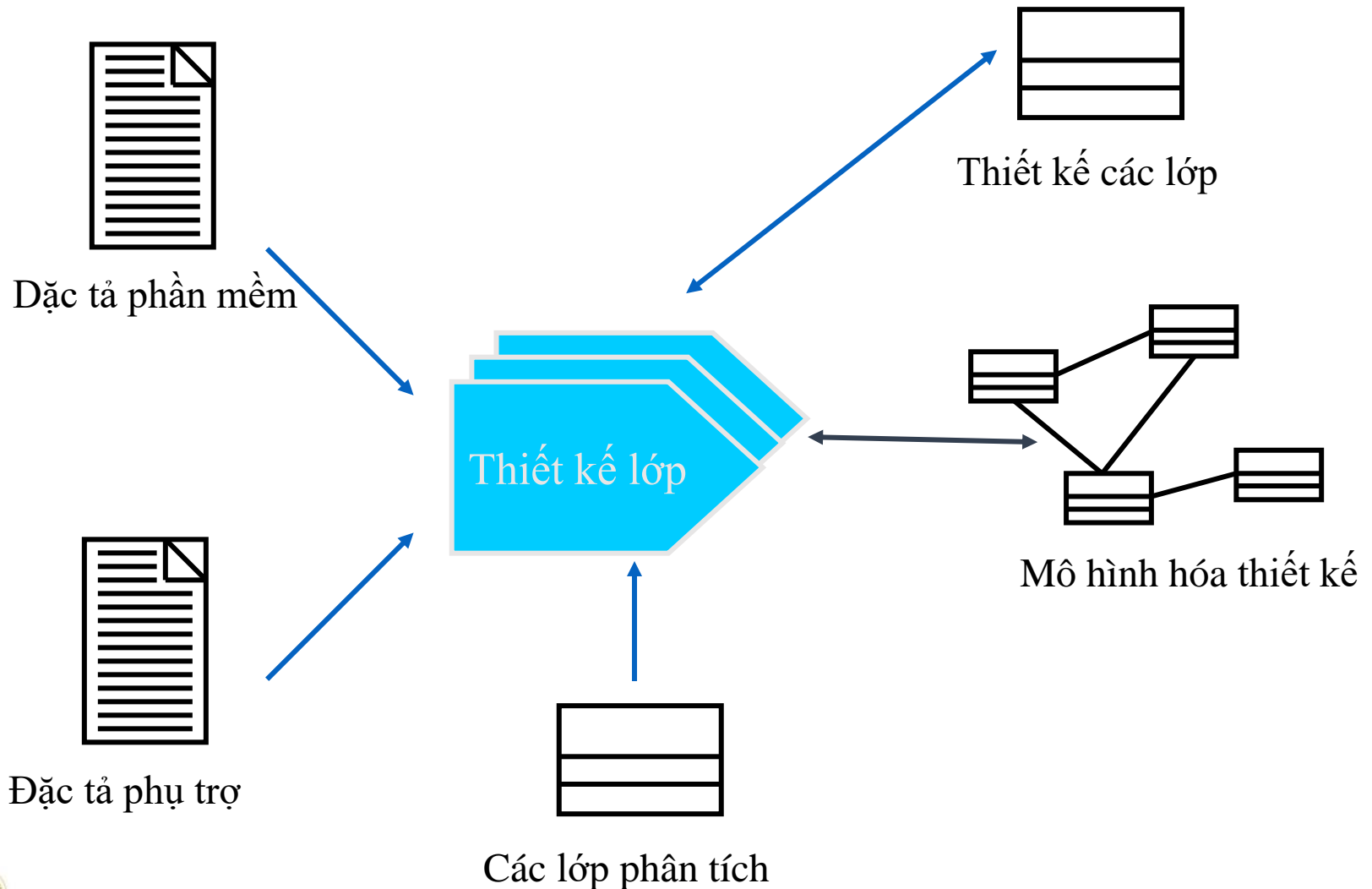
2.3. Thiết kế hệ thống con

Hệ thống con và giao diện

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



Thiết kế lớp

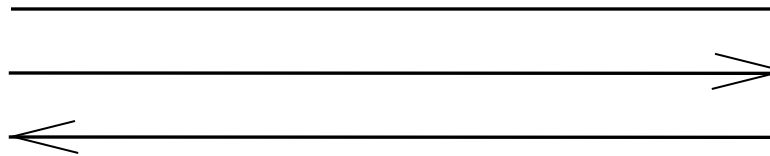


Thiết kế lớp

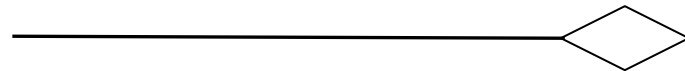
1. Tạo các lớp khởi tạo
2. Định nghĩa ra các thao tác/phương thức
3. Định nghĩa ra mối quan hệ giữa các lớp
4. Định nghĩa ra các trạng thái
5. Định nghĩa ra các thuộc tính
6. Sơ đồ lớp

Mỗi quan hệ giữa các class

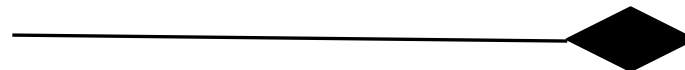
- Association
(kết hợp)



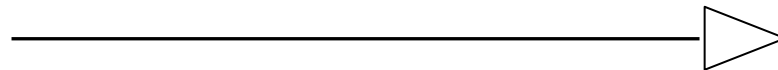
- Aggregation (kết tập)



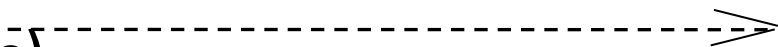
- Composition (hợp thành)



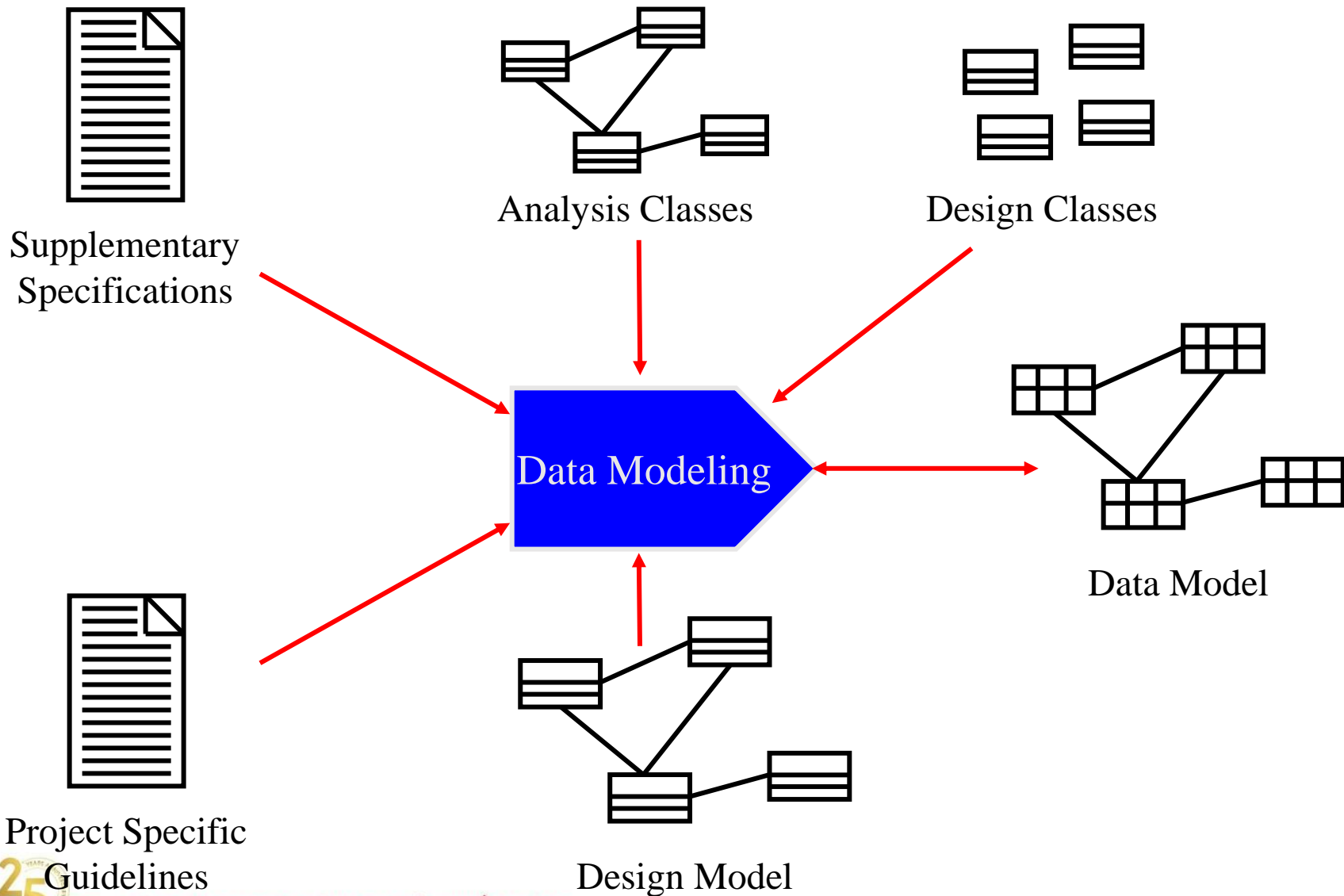
- Inheritance (Kế thừa)



- Dependence (Phụ thuộc)



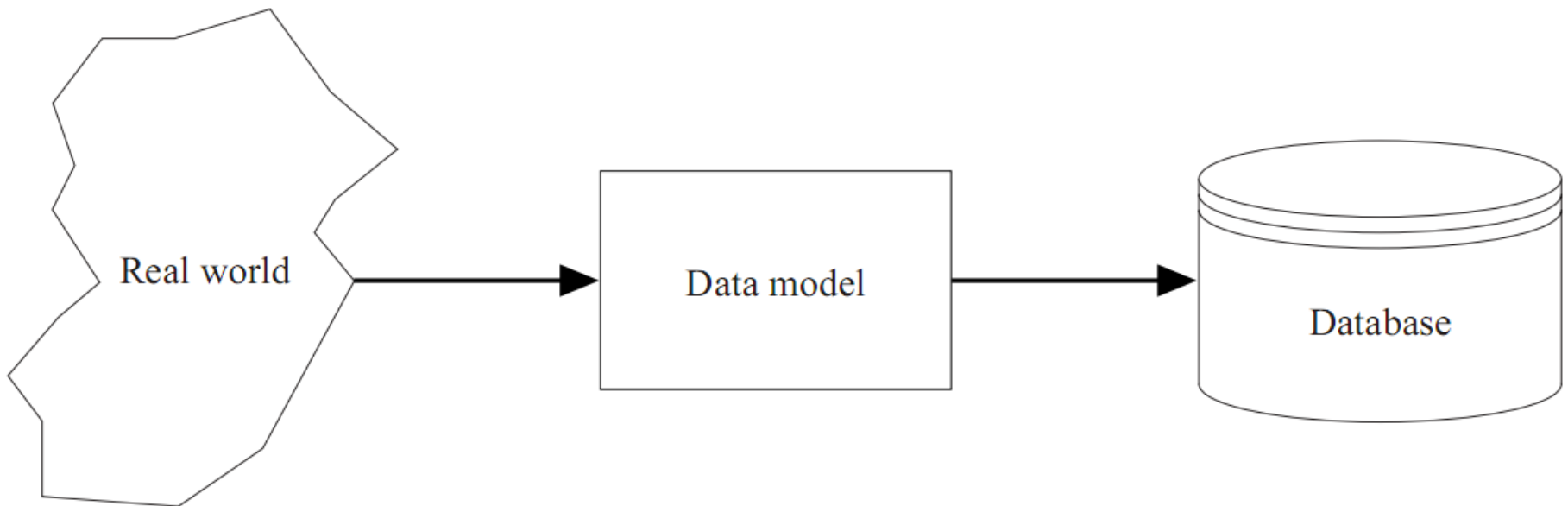
Mô hình hoá dữ liệu



Mô hình dữ liệu

◆ Mô hình hoá dữ liệu:

- Quá trình trừu tượng hoá và tổ chức cấu trúc của thông tin trong thế giới thực, là đối tượng tạo thành cơ sở dữ liệu và sau đó thể hiện nó



Chuẩn hoá

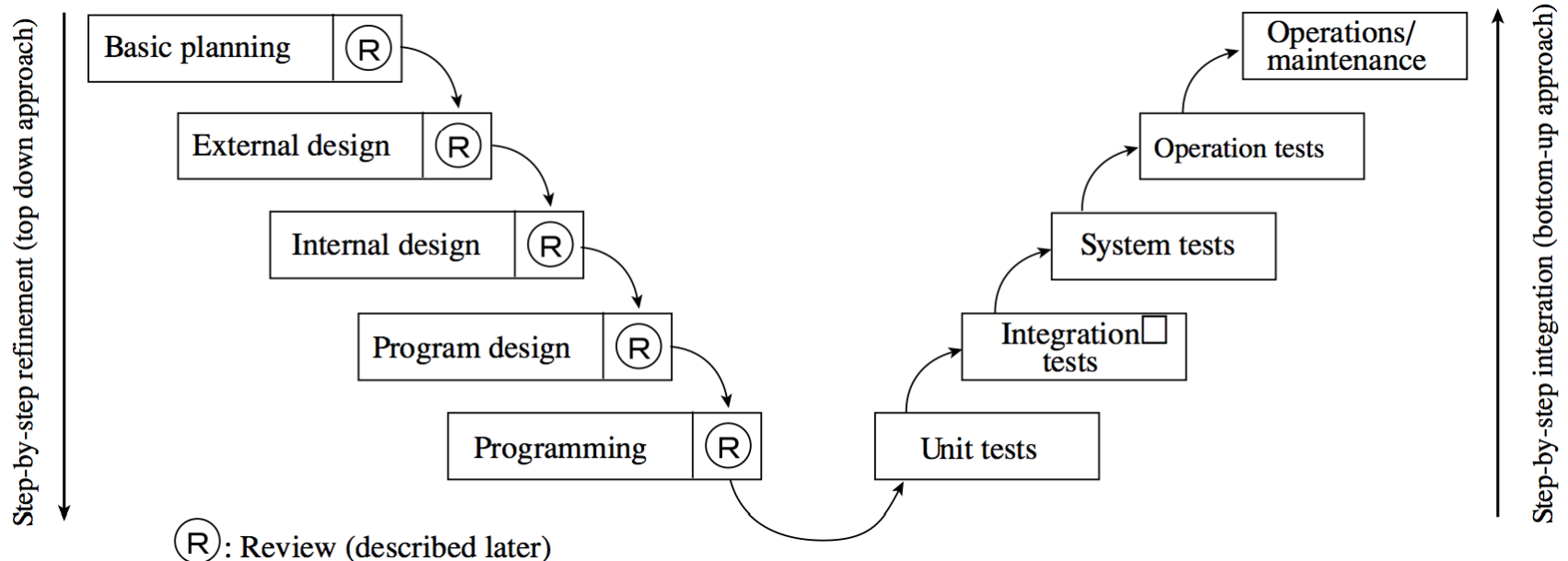
- ❖ Chuẩn hoá (Normalization): quy trình các bước sẽ xác định và loại bỏ các dư thừa trong thiết kế cơ sở dữ liệu.
- ❖ Mục đích của chuẩn hoá: để cải thiện
 - hiệu quả lưu trữ
 - toàn vẹn dữ liệu
 - và khả năng mở rộng

Chuẩn hoá (2)

- ❖ Trong mô hình quan hệ, tồn tại các phương pháp để định lượng mức độ hiệu quả của cơ sở dữ liệu.
- ❖ Các phân loại này được gọi là **các dạng chuẩn (normal forms hoặc NF)**, và có các thuật toán để chuyển đổi một cơ sở dữ liệu đã cho giữa chúng.
- ❖ Chuẩn hóa thường liên quan đến việc tách các bảng hiện có thành nhiều bảng, các bảng này phải được nối lại hoặc liên kết mỗi khi truy vấn được đưa ra.

Mô hình chữ V (V Model) – Các mức kiểm thử khác nhau

- ❖ Kiểm thử đơn vị (Unit test): riêng từng module một (ONE module at a time)
- ❖ Kiểm thử tích hợp (Integration test): liên kết các modules
- ❖ Kiểm thử hệ thống (System test): tổng thể (toàn bộ) hệ thống



Các kỹ thuật kiểm thử đơn vị

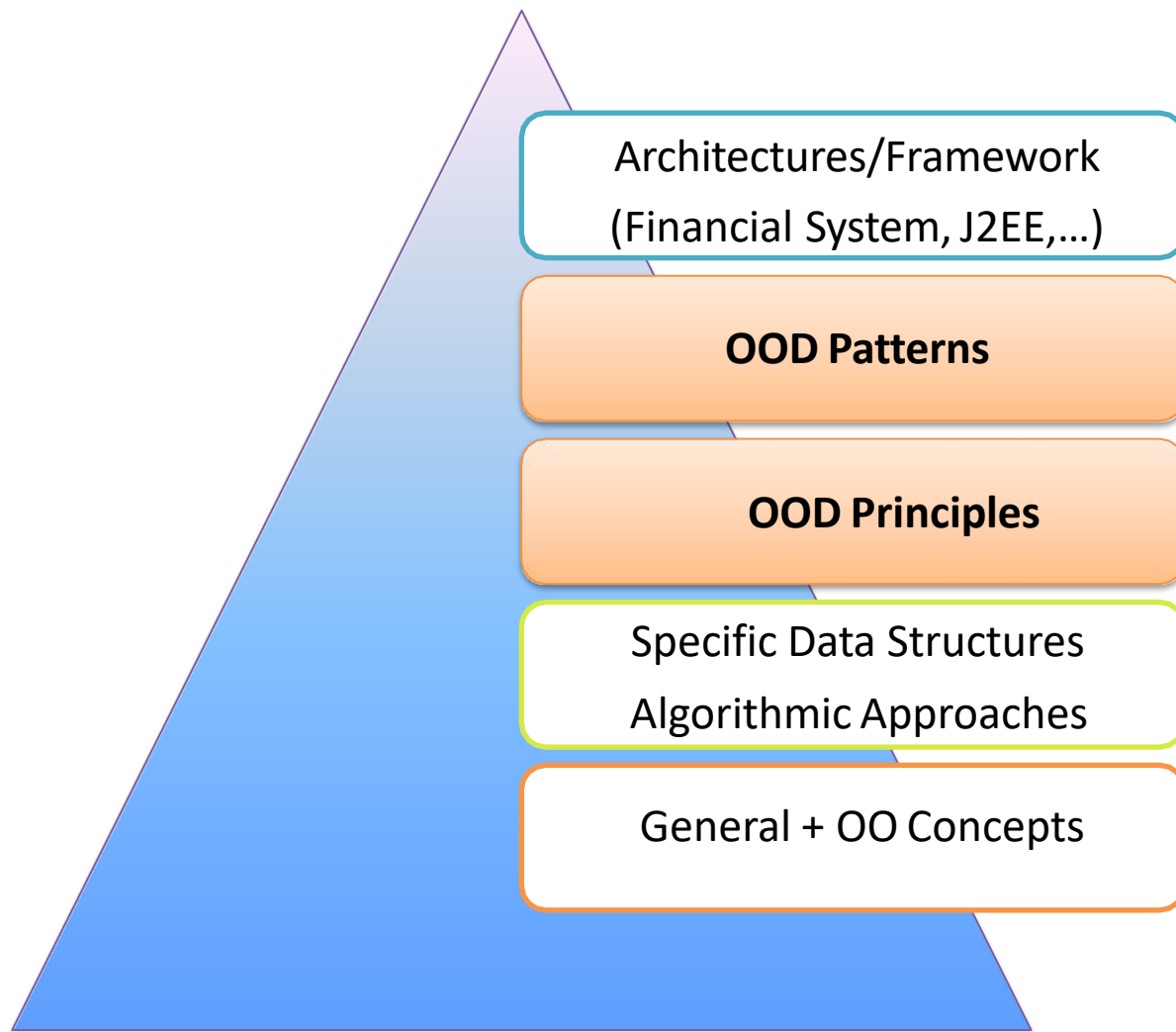
- ❖ Dành cho thiết kế trường hợp thử
- ❖ (2.2) Các kỹ thuật kiểm thử cho kiểm thử hộp đen (Black Box Test)
 - Phân tích phân vùng tương đương (Equivalence Partitioning Analysis)
 - Phân tích giá trị biên (Boundary-value Analysis)
 - Bảng quyết định (Decision Table)
 - Kiểm thử dựa trên ca sử dụng (Use Case-based Test)
- ❖ (2.3) Các kỹ thuật kiểm thử cho kiểm thử hộp trắng (White Box Test)
 - Kiểm thử luồng điều khiển với phủ C0, C1 (Control Flow Test with C0, C1 coverage)
 - Kiểm thử phủ biểu đồ tuần tự (Sequence chart coverage test)



Xây dựng chương trình

1. Phong cách lập trình
2. Tinh chỉnh / tối ưu mã (tuning / optimization)
3. Tái cấu trúc mã (refactoring)
4. Debugging

Các nguyên lý thiết kế





Key design concepts

General

- Cohesion
- Coupling
- Information hiding
 - Encapsulation
 - Creation
- Binding time

OO Specific

- Behaviors follow data
- Class vs. Interface Inheritance
 - Class = implementation
 - Interface = type
- Inheritance / composition / delegation

Good Design

❖ What's a design?

- Express a idea to resolve a problem
- Use for communications in the team members

❖ What's a good design?

- Easy for Developing, Reading & Understanding
- Easy for Communication
- Easy for Extending (add new features)
- Easy for Maintenance

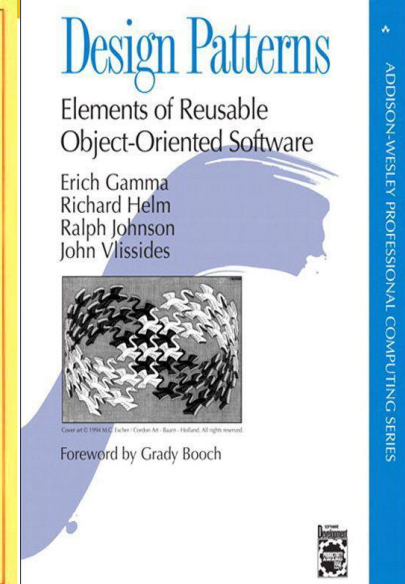
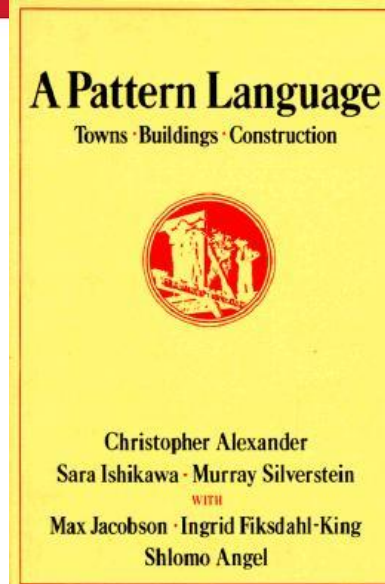
Two general design issues

- ❖ *Cohesion* – why are sub-modules (like methods) placed in the same module? Usually to collectively form an ADT
- ❖ *Coupling* – what is the dependence between modules? Reducing the dependences (which come in many forms) is desirable

S.O.L.I.D Principles of OOD

- ❖ SRP: The Single Responsibility Principle
- ❖ OCP: The Open Closed Principle
- ❖ LSP: The Liskov Substitution Principle
- ❖ ISP: The Interface Segregation Principle
- ❖ DIP: The Dependency Inversion Principle

Design Patterns



- ❖ Published in 1994
- ❖ “Each pattern describes a **problem** which occurs over and over **again** in our environment, and then describes the **core of the solution** to that problem, in such a way that you can use this solution **a million times over**, without ever doing it the same way twice”
 - Christopher Alexander
- ❖ Today's amazon.com stats

Amazon Best Sellers Rank: #2,069 in Books ([See Top 100 in Books](#))

#1 in [Books](#) > [Computers & Internet](#) > [Computer Science](#) > [Software Engineering](#) > [Design Tools & Techniques](#)

#1 in [Books](#) > [Computers & Internet](#) > [Programming](#) > [Software Design, Testing & Engineering](#) > [Software Reuse](#)

#3 in [Books](#) > [Nonfiction](#) > [Foreign Language Nonfiction](#) > [French](#)

Đánh giá môn học

❖ Hình thức: thi trực tuyến

- Thi viết với đề thi dạng mở và nộp bài qua Assignment trên Team

❖ Thời gian (dự kiến):

- Kíp 2 thứ 2 ngày 14/02/2022

Đánh giá môn học

- ❖ Đề bài sẽ mô tả một case study sau đó các em sẽ thực hiện các bước phân tích, thiết kế,... theo các chủ đề trong môn học:
 - Xác định yêu cầu phần mềm (xác định tác nhân, usecase, xây dựng biểu đồ usecase, đặc tả usecase, ràng buộc đầu vào,...)
 - Phân tích và thiết kế phần mềm (các sơ đồ UML: biểu đồ trình tự, giao tiếp, hoạt động), xây dựng và mô tả biểu đồ lớp thiết kế, thiết kế dữ liệu, thiết kế giao diện màn hình.
 - Kiểm thử (hộp đen và hộp trắng)
 - Nguyên lý thiết kế hướng đối tượng,
 - ...