



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY



Phân tích thiết kế hệ thống

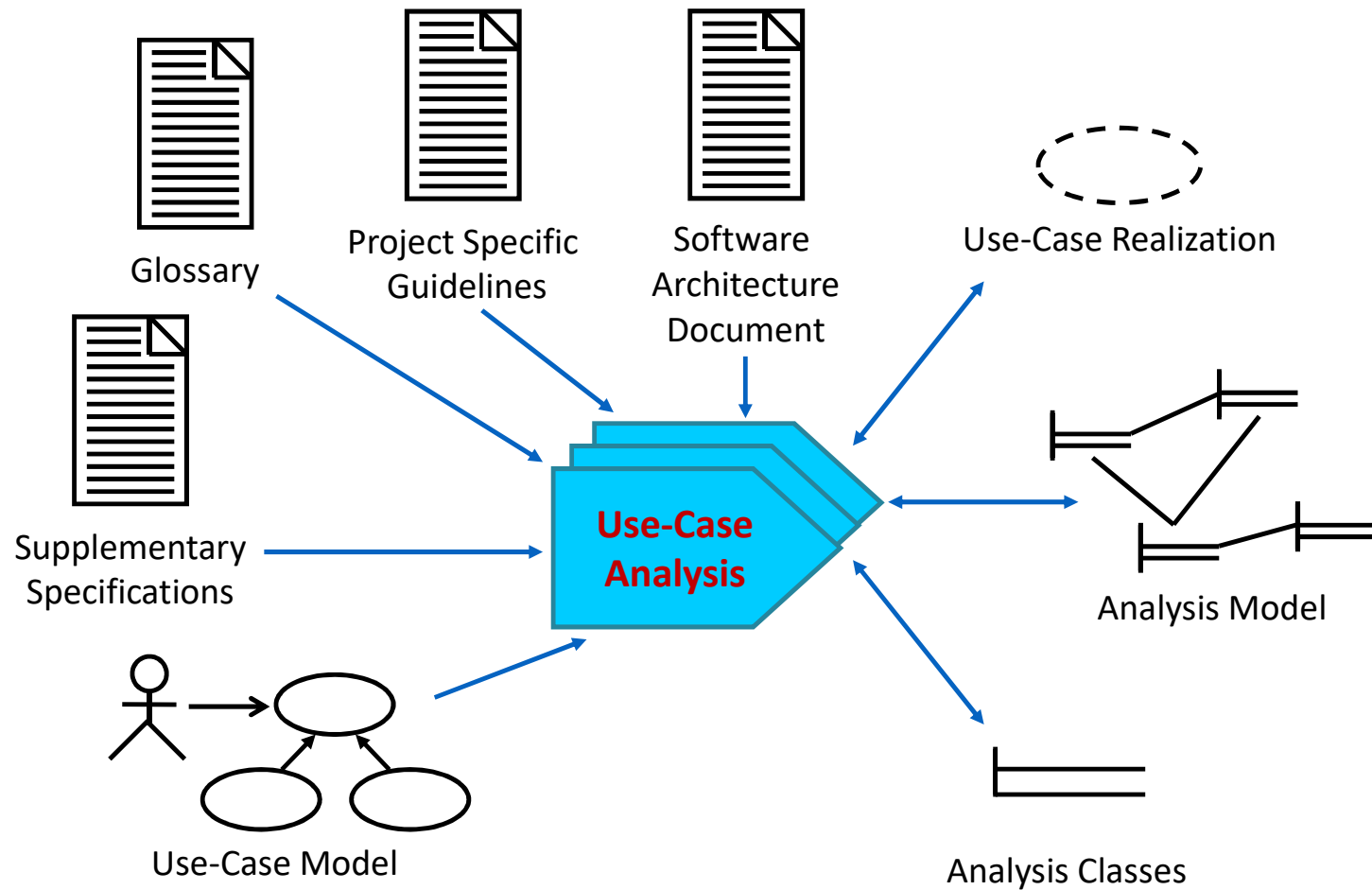
\$10 Phân tích trường hợp sử dụng



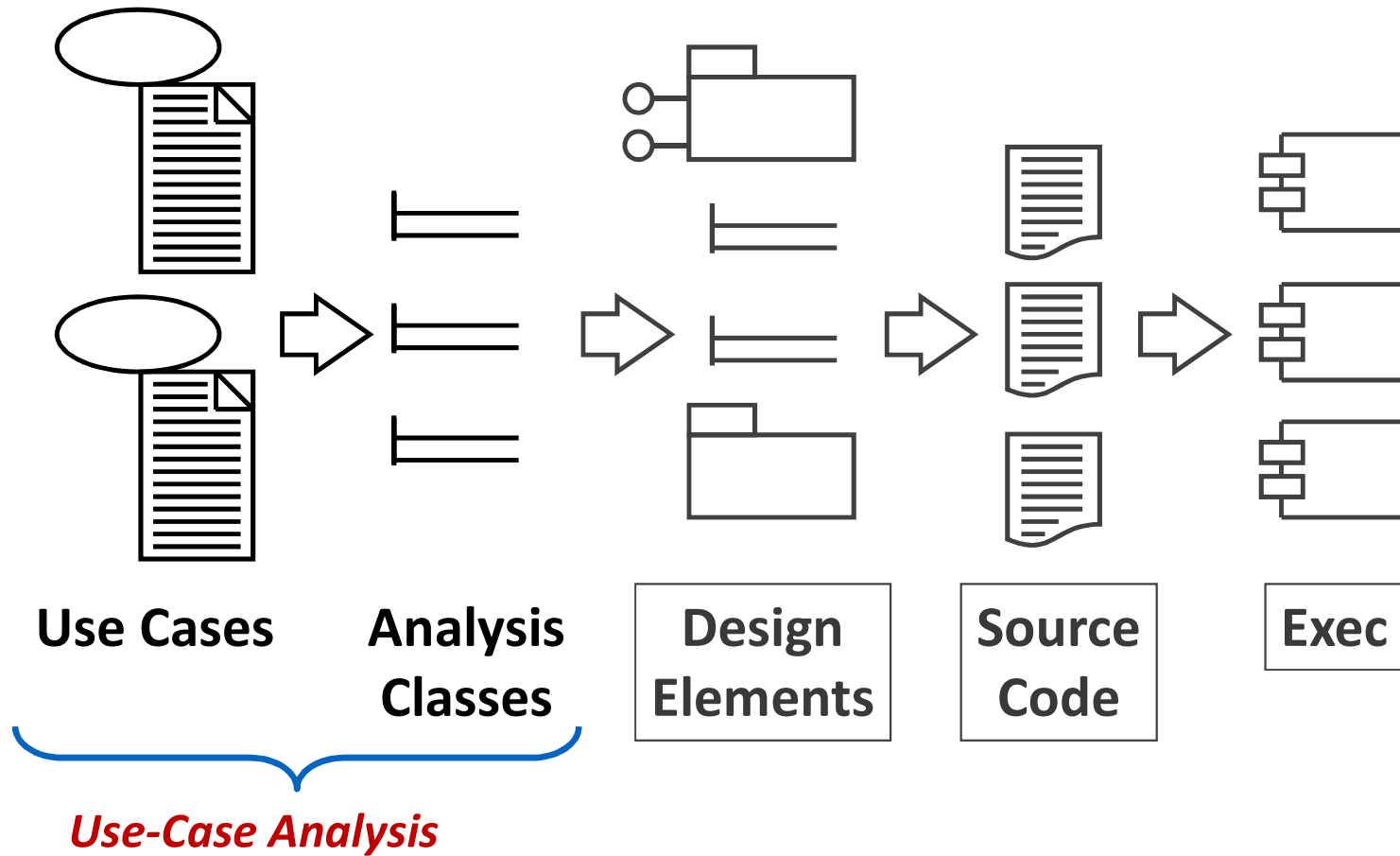
Nội dung

- Giới thiệu
- Bổ sung mô tả
- Tìm kiếm các lớp
- Phân phối hành vi tới lớp
- Mô tả trách nhiệm lớp
- Mô tả thuộc tính và liên kết
- Mô tả tính chất cơ chế phân tích
- Hợp nhất các lớp phân tích

Phân tích usecase

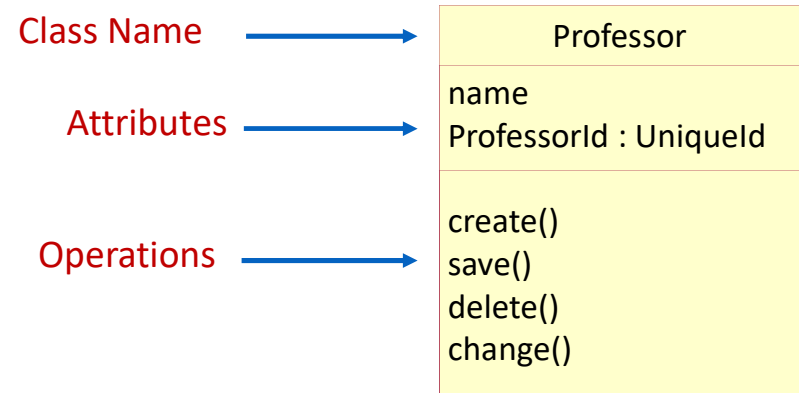


Lớp phân tích



Lớp

- Một sự trừu tượng hóa cụ thể
- Mô tả nhóm các đối tượng cùng:
 - Thuộc tính
 - Hành vi
 - Quan hệ
 - Ngữ nghĩa



Hiện thực hóa usecase

Use-Case Model

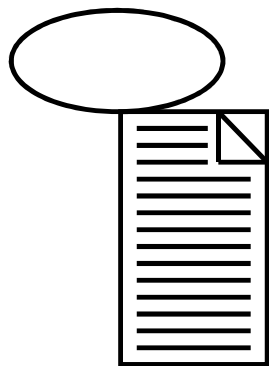


Use Case

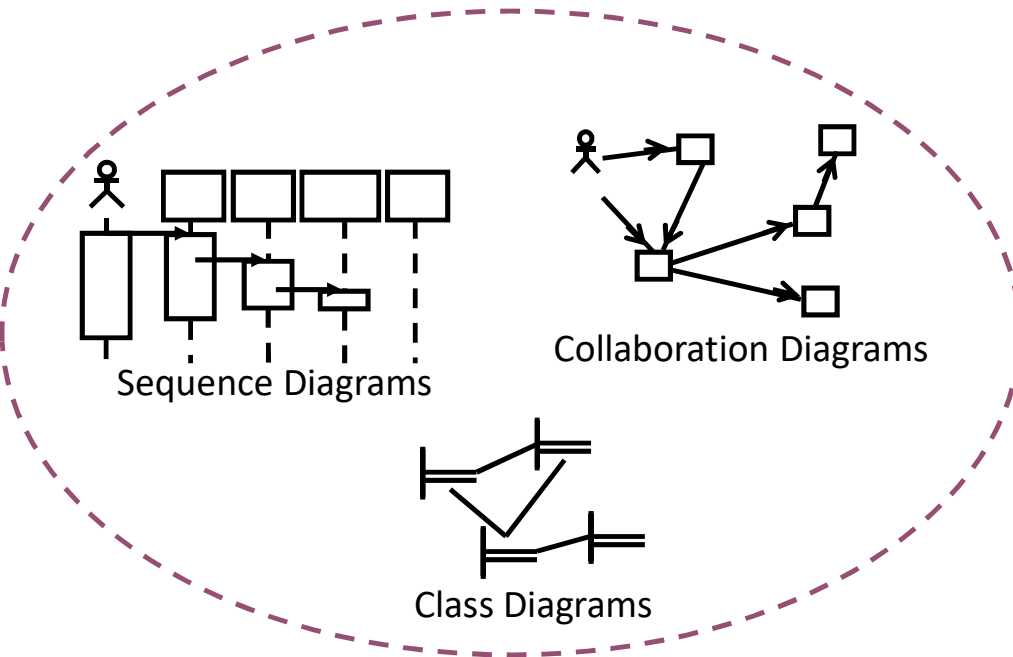
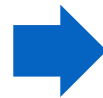
Design Model



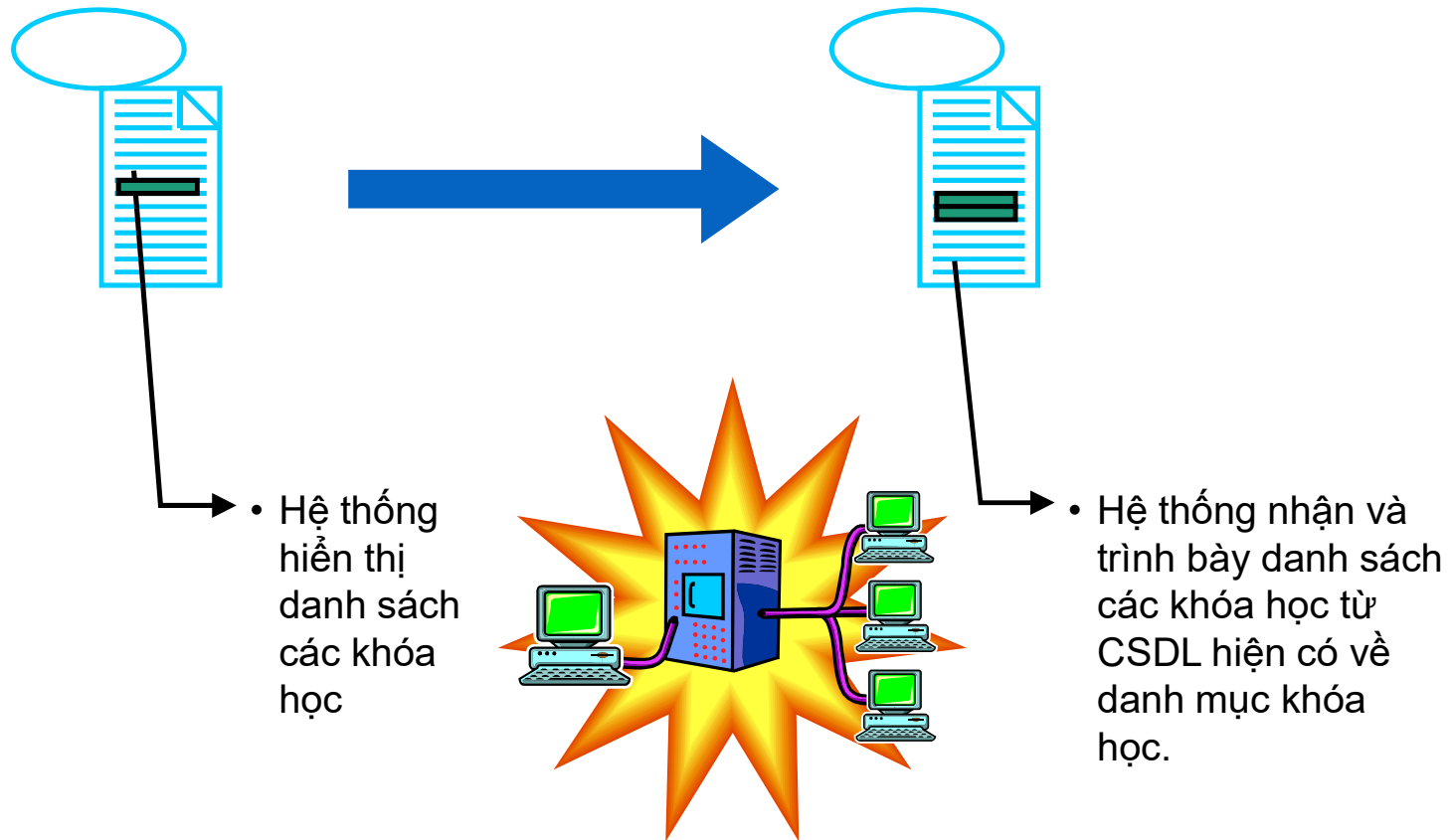
Use-Case Realization



Use Case

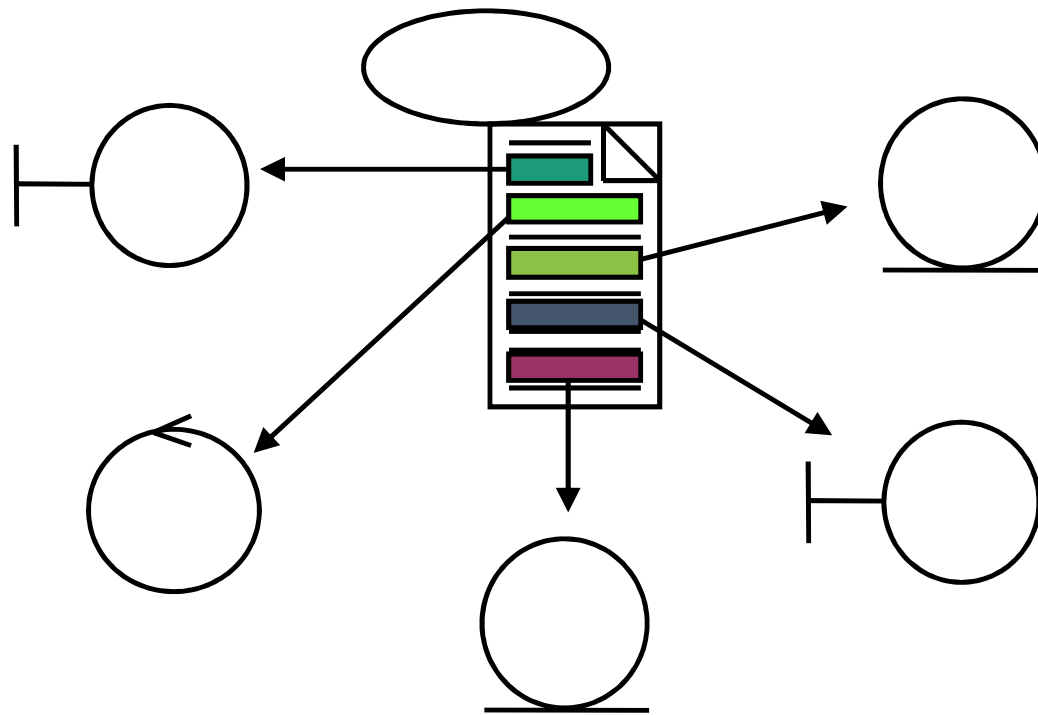


Bổ sung mô tả usecase

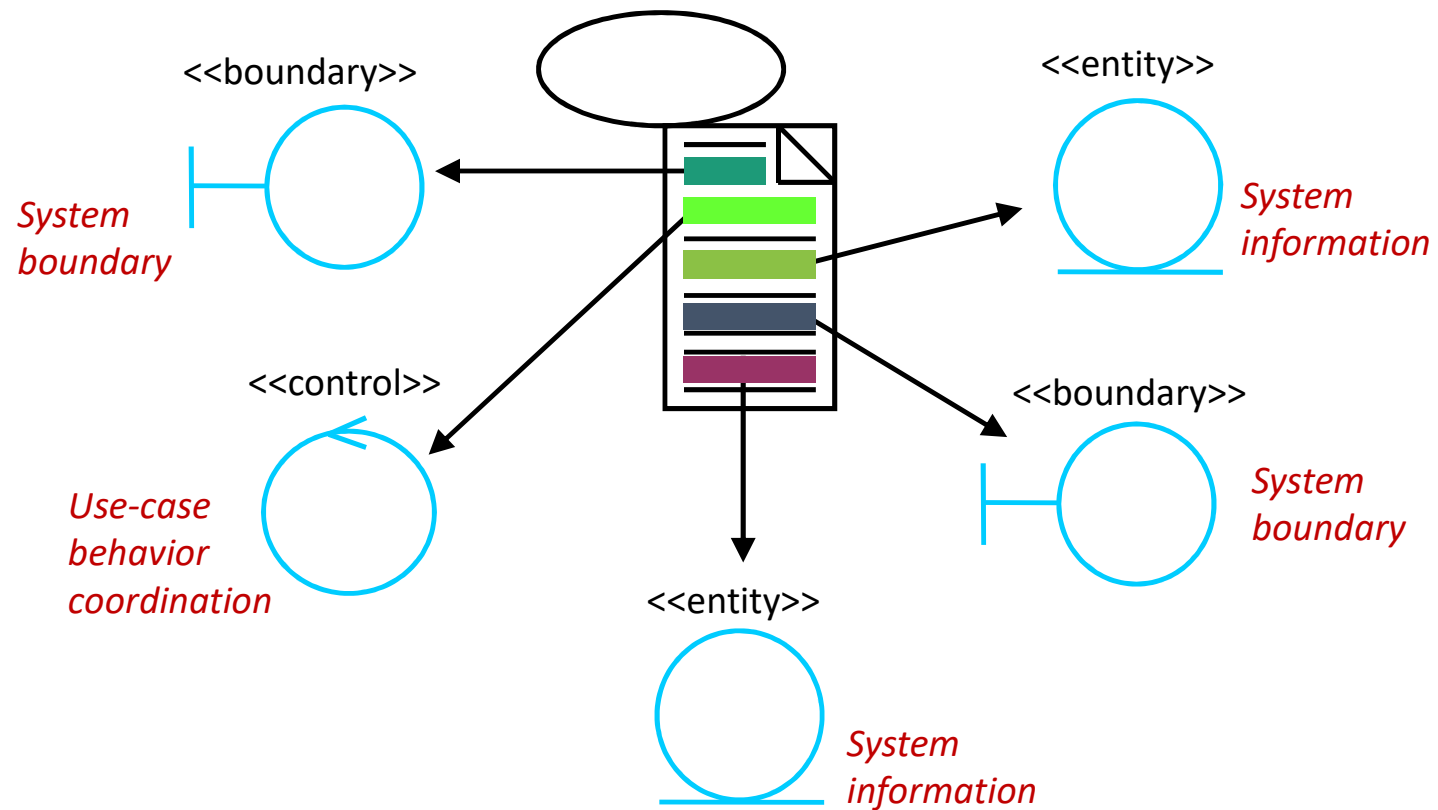


Phát hiện các lớp từ hành vi của usecase

- Hành vi của một usecase phải được phân bổ vào các lớp phân tích



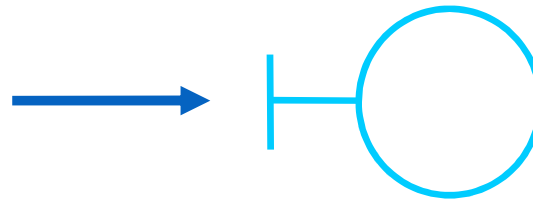
Lớp phân tích



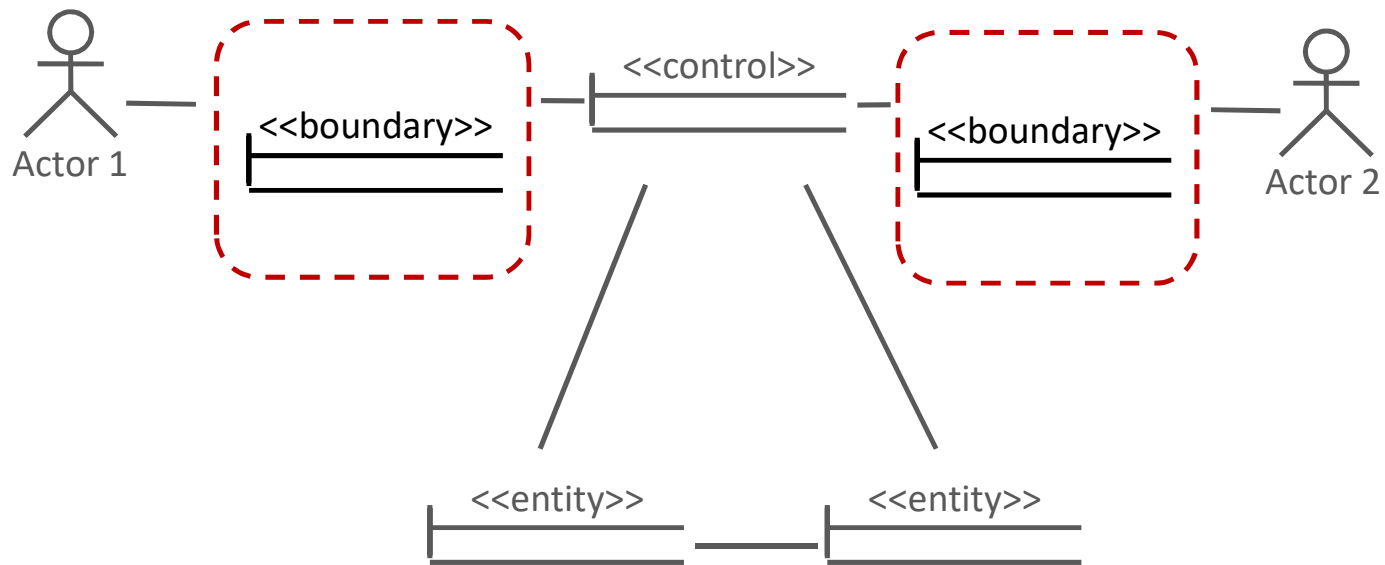
Lớp biên

- Trung gian giữa giao diện và đối tượng ngoài hệ thống
- Có thể thuộc các dạng:
 - Lớp giao diện người dùng
 - Lớp giao diện hệ thống
 - Lớp giao diện thiết bị
- Một lớp biên cho mỗi cặp tác nhân/usecase

*Analysis class
stereotype*



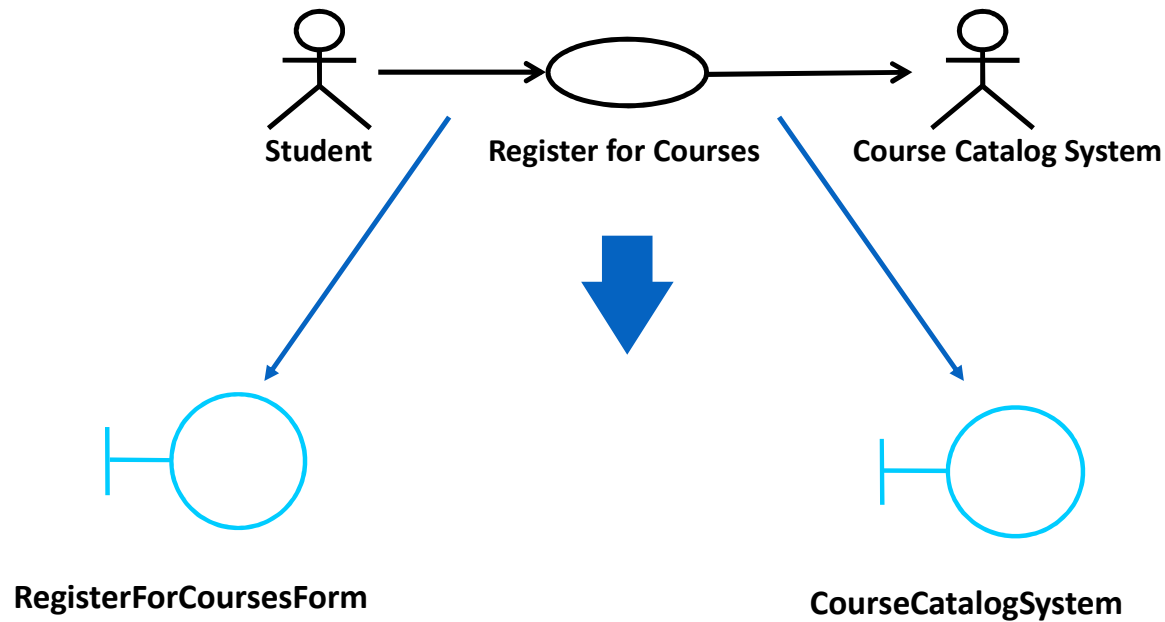
Vai trò của lớp biên



Mô hình tương tác giữa hệ thống và môi trường

Thí dụ tìm lớp biên

- Một lớp biên cho mỗi cặp tác nhân/usecase

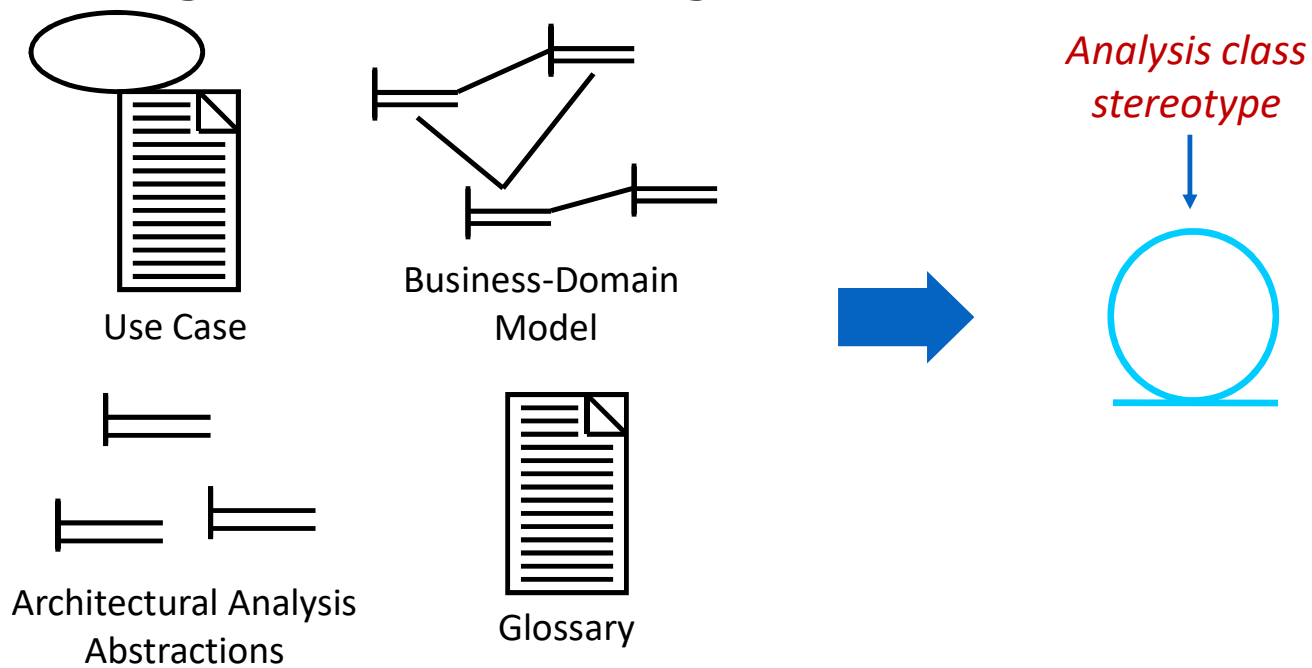


Chỉ dẫn đối với lớp biên

- Lớp giao diện người dùng
 - Tập trung vào thông tin được thể hiện, không vào chi tiết GUI
- Lớp giao diện hệ thống/thiết bị
 - Tập trung vào xác định các giao thức
 - Không tập trung vào giao thức được thể hiện thế nào

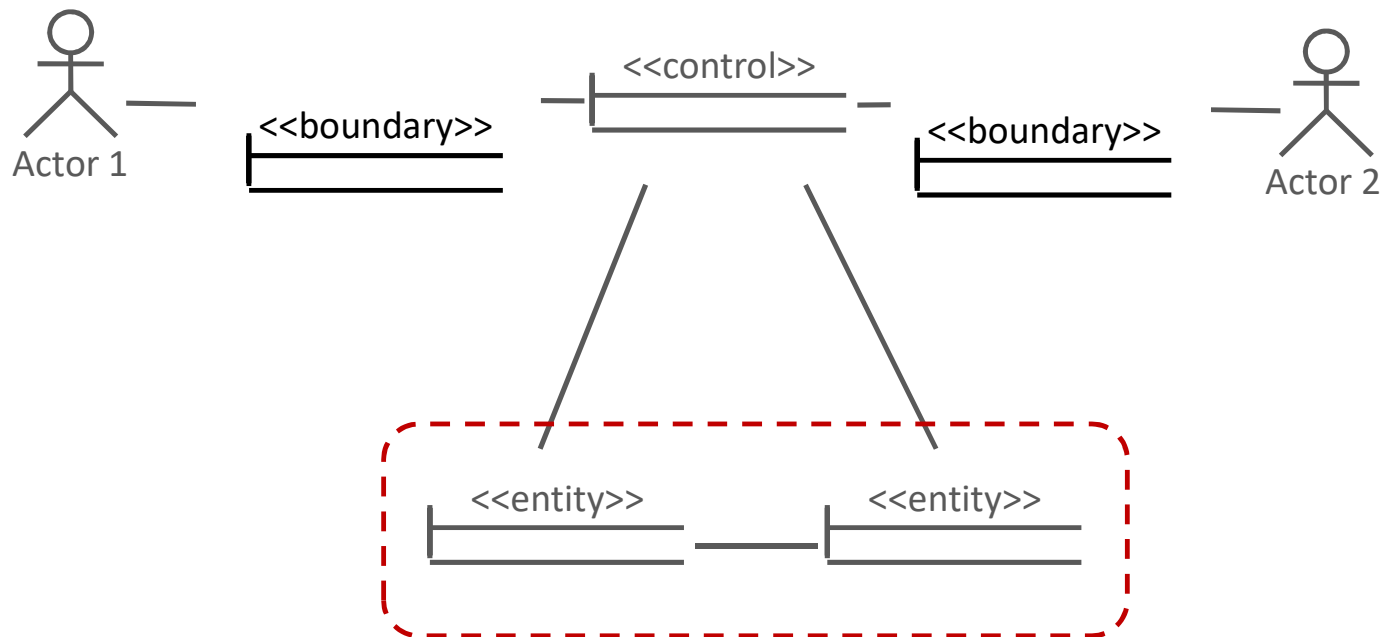
Lớp thực thể - entity

- Các trườ tượng chính của hệ thống



Độc lập với môi trường

Vai trò của lớp thực thể

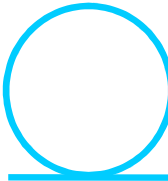


Lưu trữ và quản lý thông tin trong hệ thống

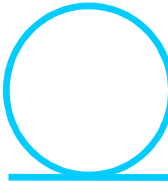
Thí dụ lớp thực thể



CourseOffering



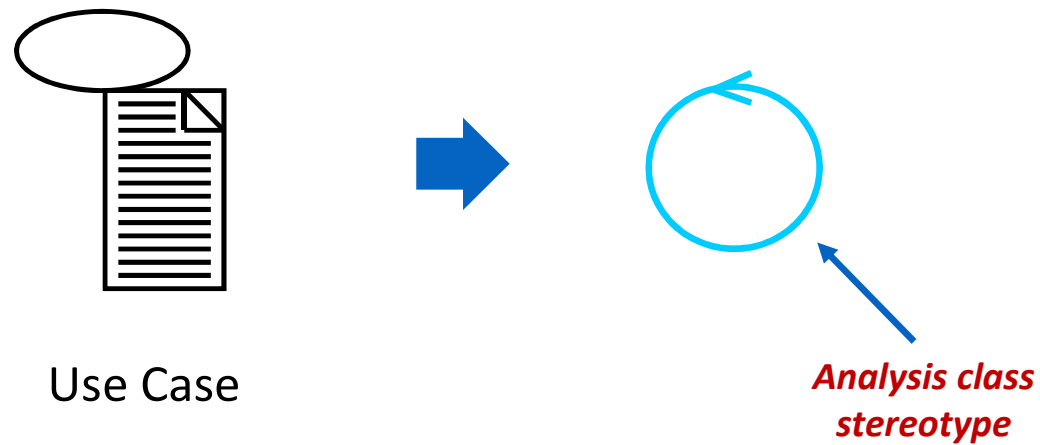
Schedule



Student

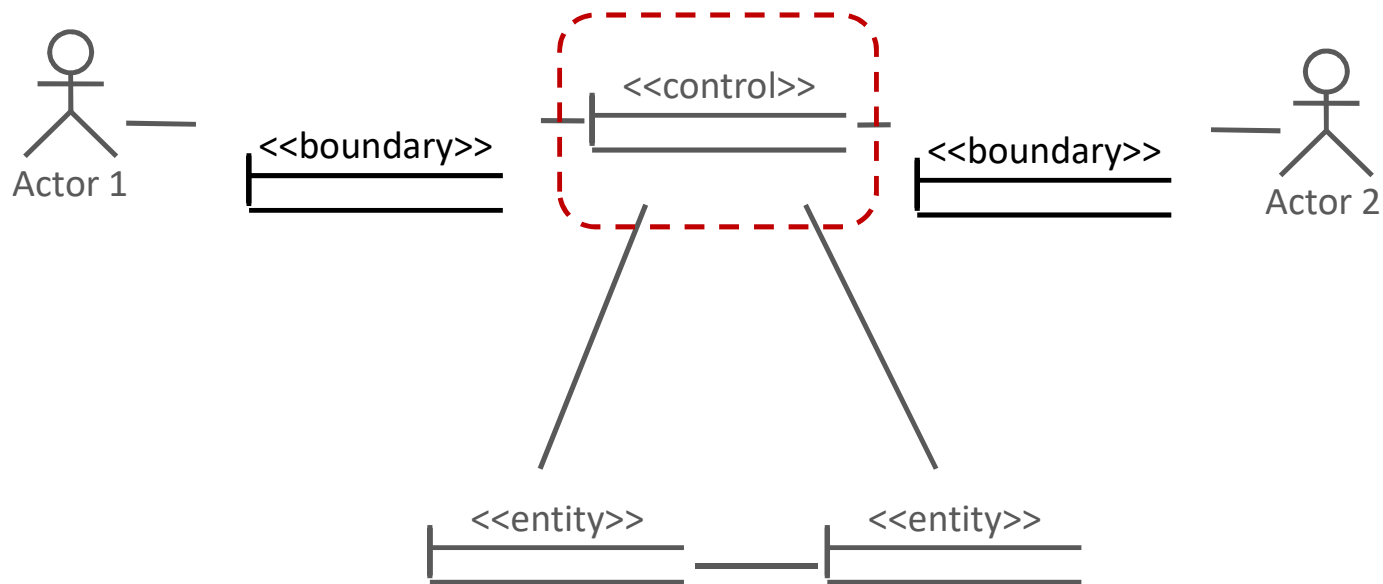
Lớp điều khiển

- Phối hợp hành vi của usecase
 - Các usecase phức tạp thường đòi hỏi nhiều lớp điều khiển



Phụ thuộc usecase, độc lập môi trường

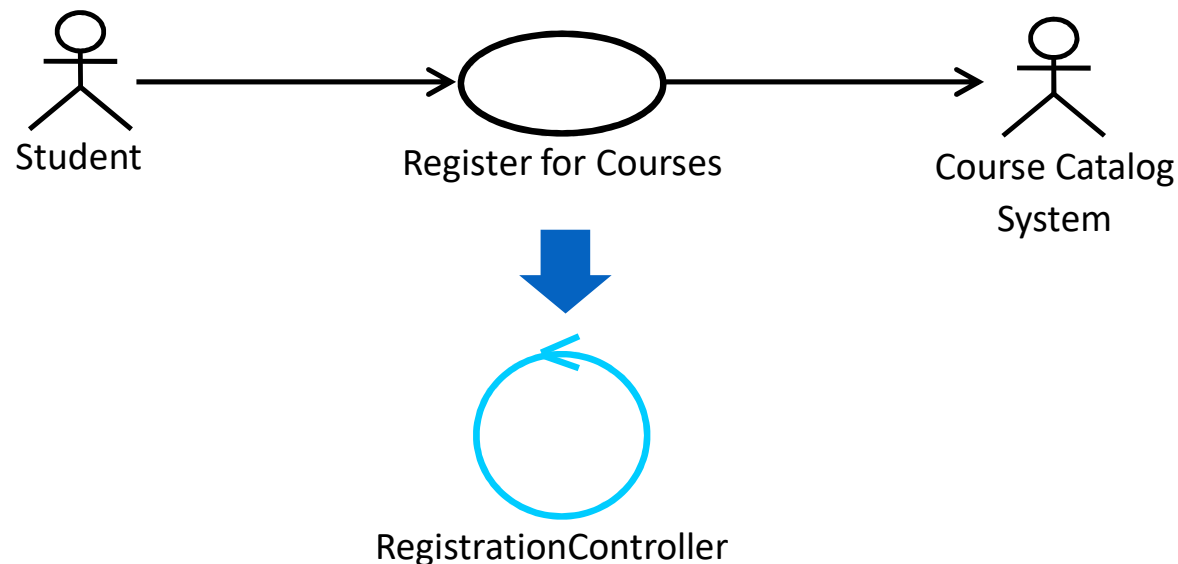
Vai trò của lớp điều khiển



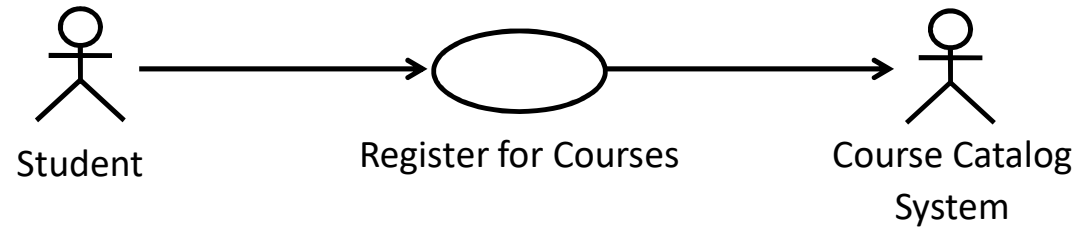
Phối hợp các hành vi của usecase

Thí dụ lớp điều khiển

- Nói chung, xác định mỗi lớp điều khiển cho 1 usecase
 - Tiếp tục phân tích, nếu lớp điều khiển phức tạp, có thể tách nó thành nhiều lớp

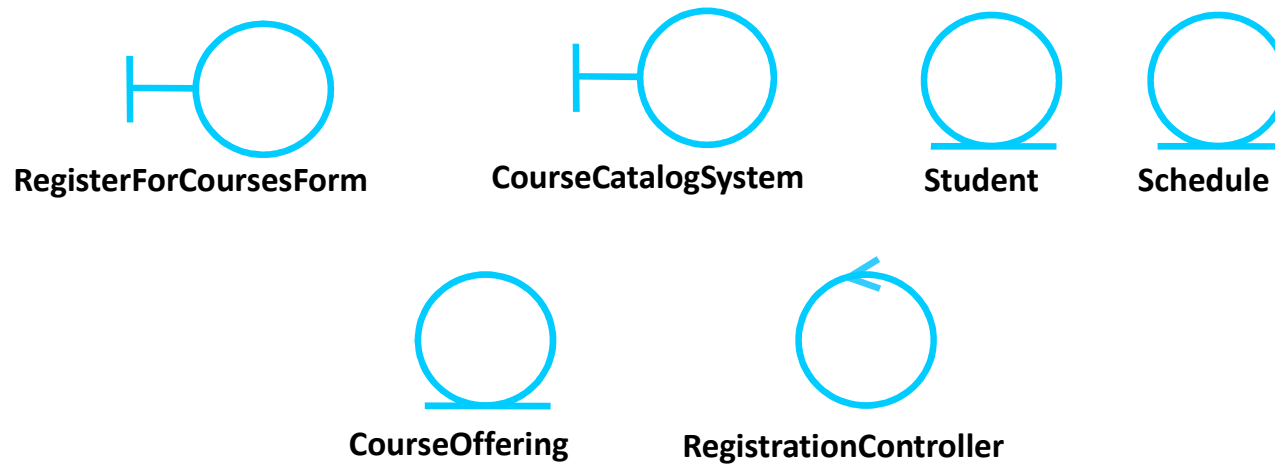


Lớp phân tích



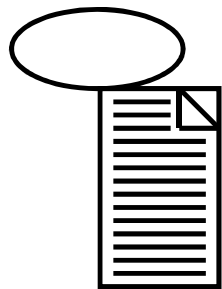
Use-Case Model

Design Model

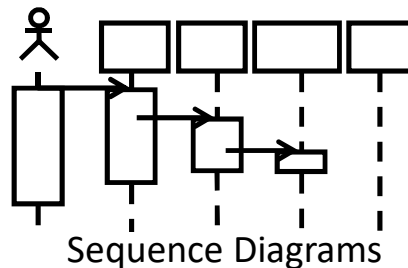


Phân bổ hành vi usecase tới các lớp

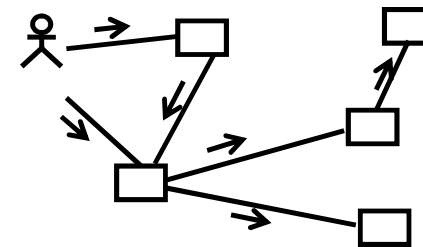
- Đối với mỗi luồng sự kiện của usecase
 - Nhận diện các lớp phân tích
 - Phân bổ trách nhiệm của usecase tới các lớp
 - Mô hình tương tác của các lớp phân tích trong các biểu đồ tương tác



Use Case



Sequence Diagrams



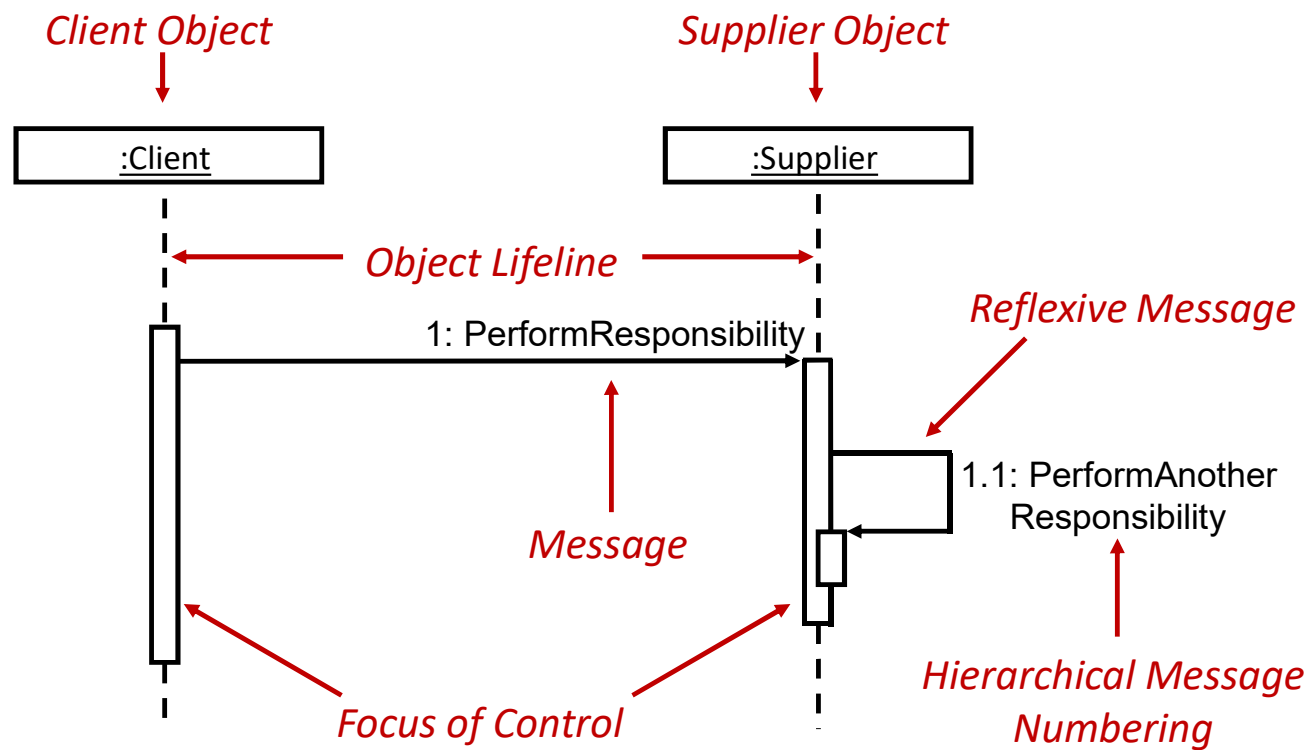
Collaboration Diagrams

Use-Case Realization

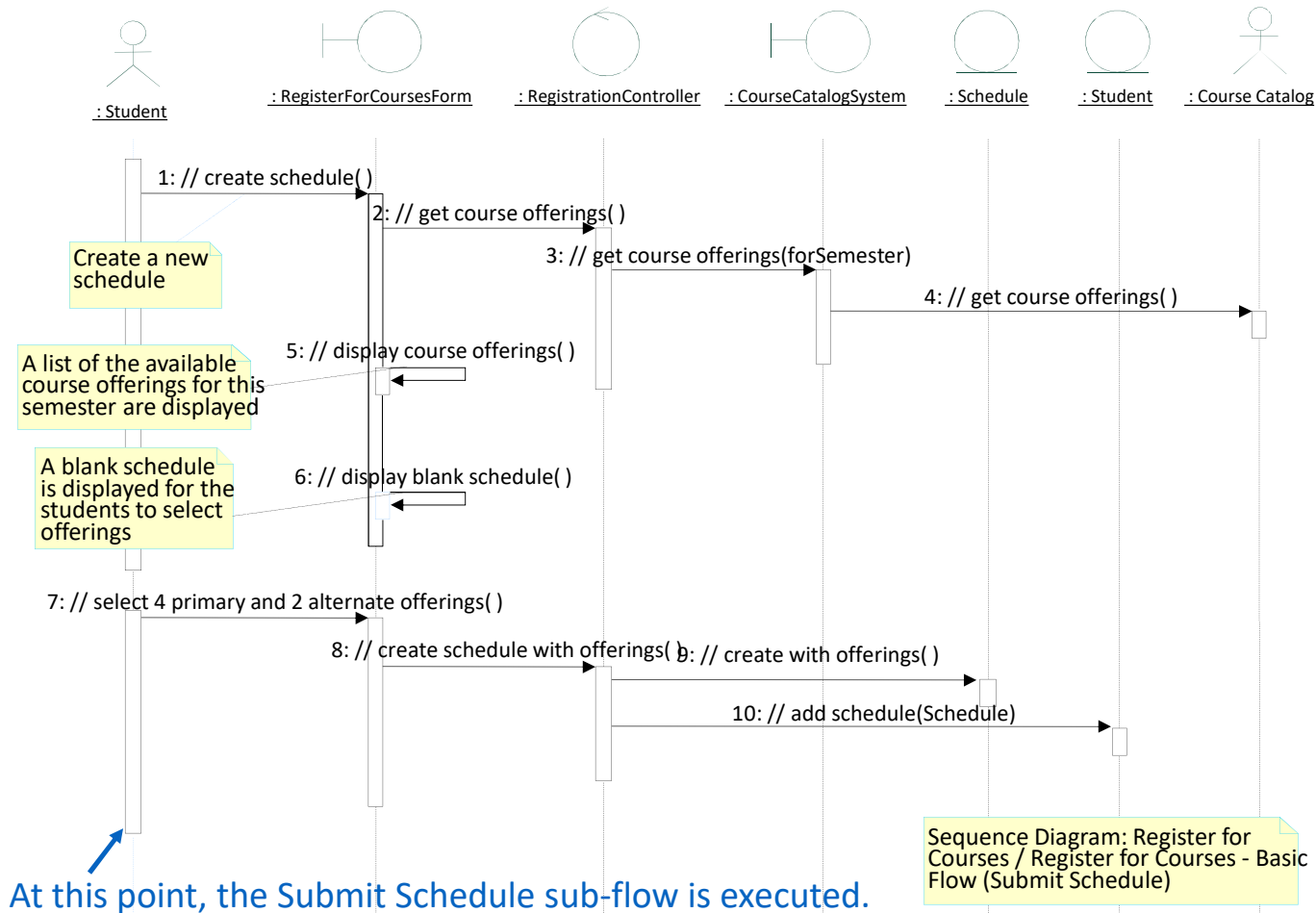
Hướng dẫn phân bố

- Lớp biên
 - Hành vi trao đổi thông tin với tác nhân
- Lớp thực thể
 - Hành vi đóng gói dữ liệu trong các trừu tượng
- Lớp điều khiển
 - Hành vi điều khiển luồng sự kiện
- Dữ liệu nên đi cùng chức năng

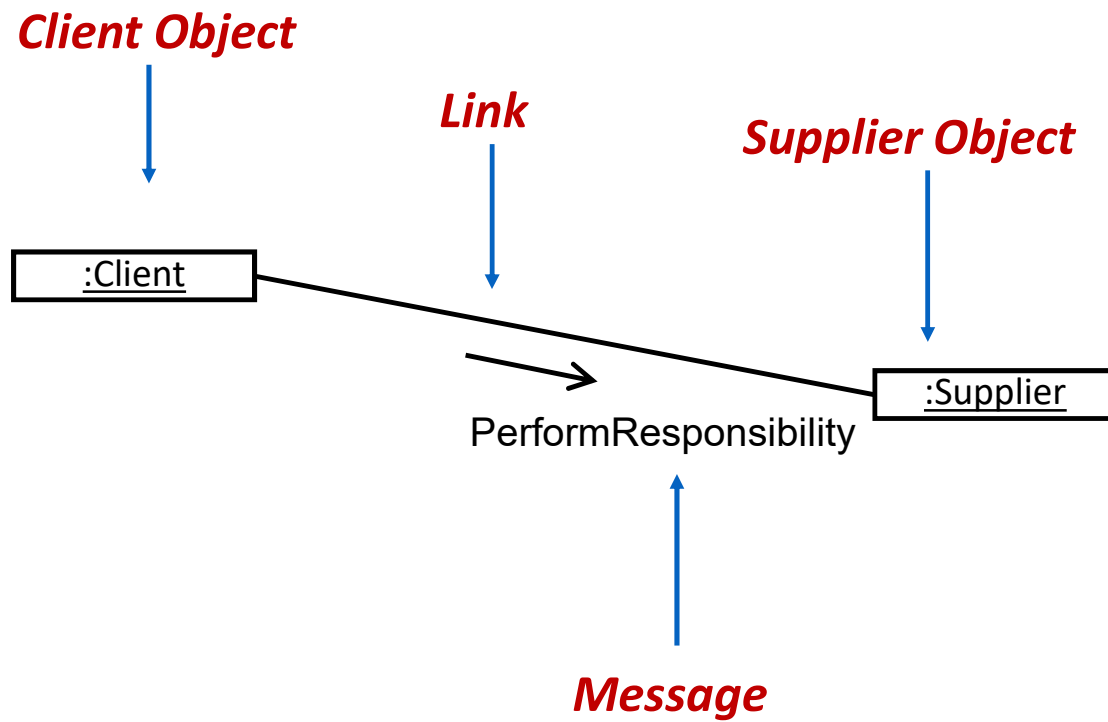
Biểu đồ tuần tự



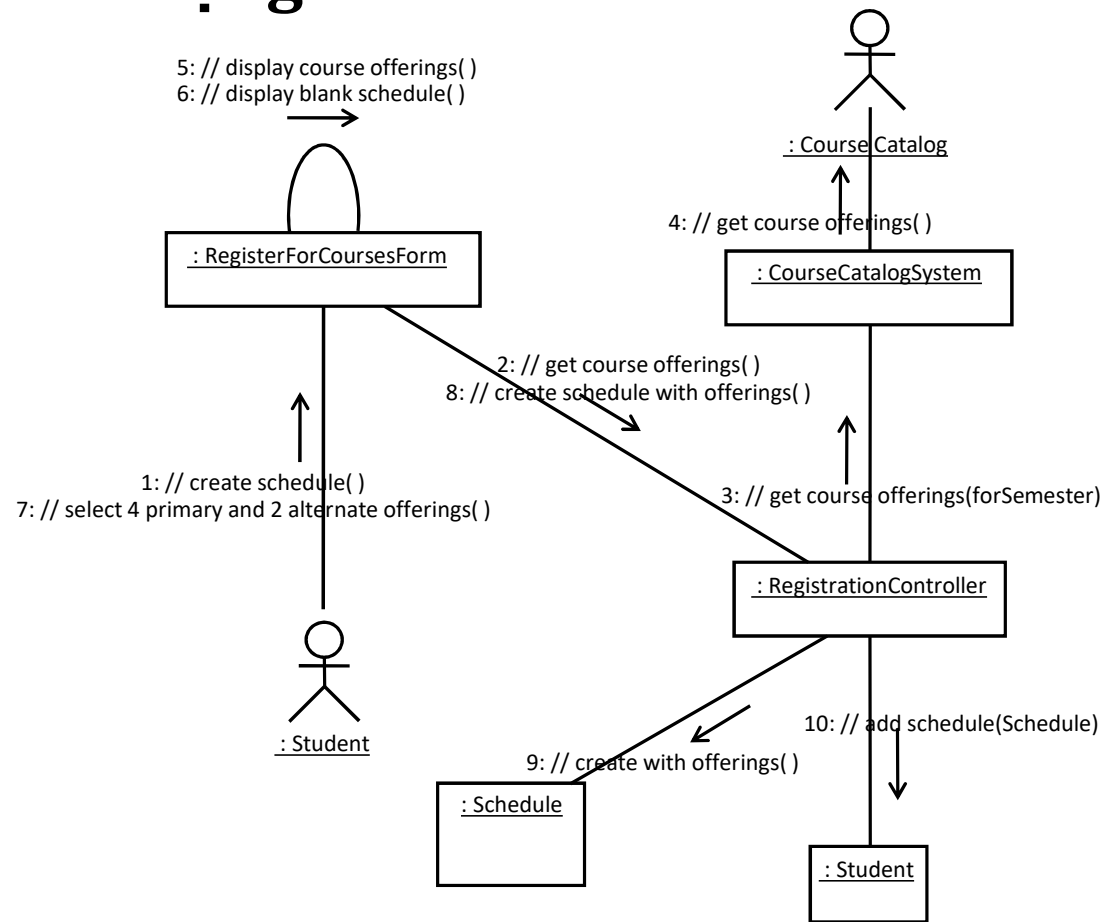
Thí dụ



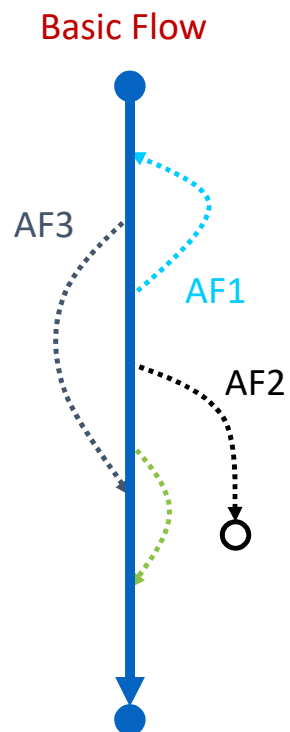
Biểu đồ cộng tác



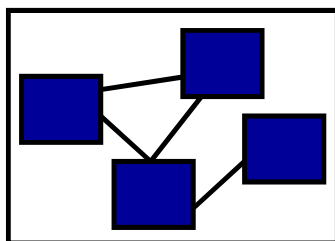
Thí dụ biểu đồ cộng tác



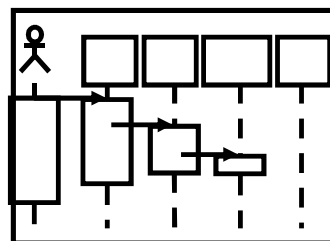
Có thể cần nhiều biểu đồ tương tác



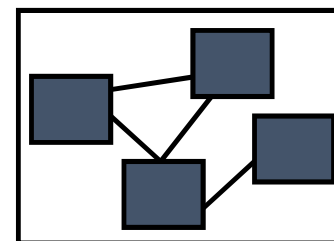
Alternate Flow 1



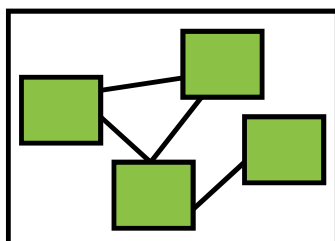
Alternate Flow 2



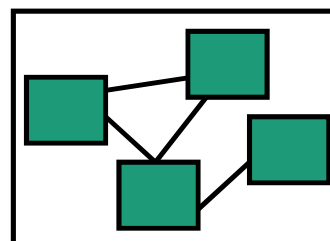
Alternate Flow 3



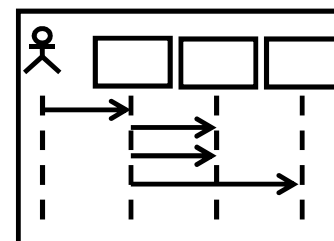
Alternate Flow 4



Alternate Flow 5



Alternate Flow n

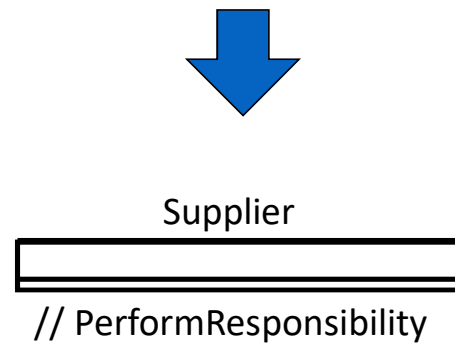


Mô tả trách nhiệm của lớp

Interaction Diagram



Class Diagram



Thí dụ

<<entity>> Student
// get tuition() // add schedule() // get schedule() // delete schedule() // has pre-requisites()

<<control>> RegistrationController
// get course offerings() // get current schedule() // delete current schedule() // submit schedule() // is registration open?() // save schedule() // create schedule with offerings() // update schedule with new selections()

<<entity>> Schedule
// commit() // select alternate() // remove offering() // level() // cancel() // get cost() // delete() // submit() // save() // any conflicts?() // create with offerings() // update with new selections()

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

<<boundary>> RegisterForCoursesForm
// display course offerings() // display blank schedule() // update offering selections()

Mô tả thuộc tính và liên kết

<<stereotype>>
ClassName

Attribute : Type = InitValue
Attribute : Type = InitValue
Attribute : Type = InitValue

*Trong giai đoạn phân tích,
không tập trung vào kiểu
thuộc tính*

attribute

<<entity>>
CourseOffering

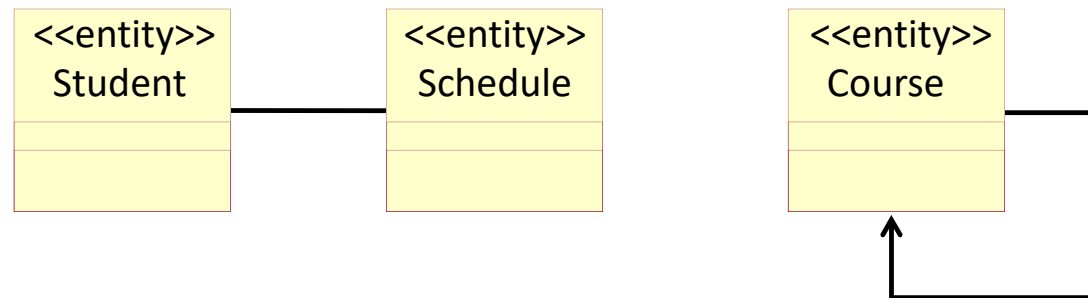
number : String = "100"
startTime : Time
endTime : Time
days : Enum
numStudents : Int

Tìm thuộc tính

- Tính chất của lớp xác định
- Thông tin duy trì bởi lớp
- Các “danh từ” mà không chuyển thành lớp

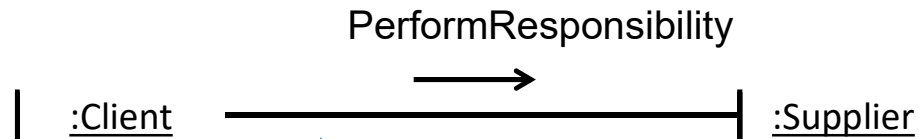
Liên kết

- Quan hệ ngữ nghĩa giữa 2 hay nhiều lớp
- Quan hệ cấu trúc xác định liên kết giữa các đối tượng của một lớp với các đối tượng của các lớp khác

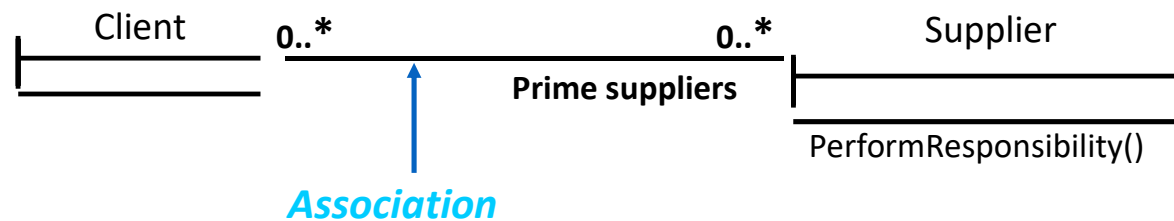


Tìm liên kết

**Collaboration
Diagram**



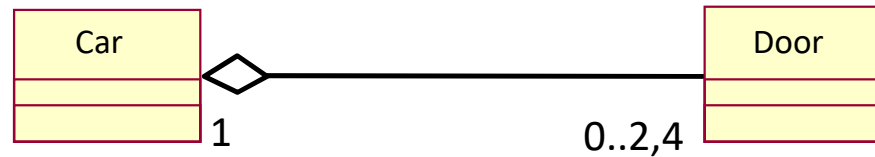
**Class
Diagram**



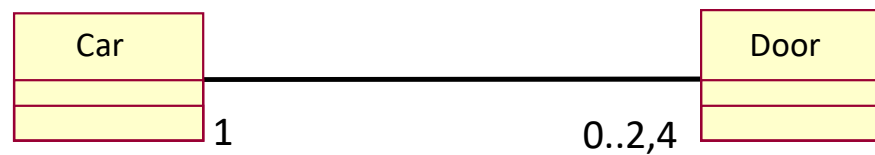
Relationship for every link!

Liên kết và tổ hợp

- Tổ hợp



- Liên kết



Mô tả tính chất cơ chế phân tích

- Thí dụ
- Cơ chế lưu trữ của lớp Schedule:
 - Granularity: 1 to 10 Kbytes per product
 - Volume: up to 2,000 schedules
 - Access frequency
 - Create: 500 per day
 - Read: 2,000 access per hour
 - Update: 1,000 per day
 - Delete: 50 per day
 - Các tính chất khác

Hợp nhất các lớp phân tích

