

# Phân tích Thiết kế HTTP

Phân tích ca sử dụng

Ths. Nguyễn Văn Nam

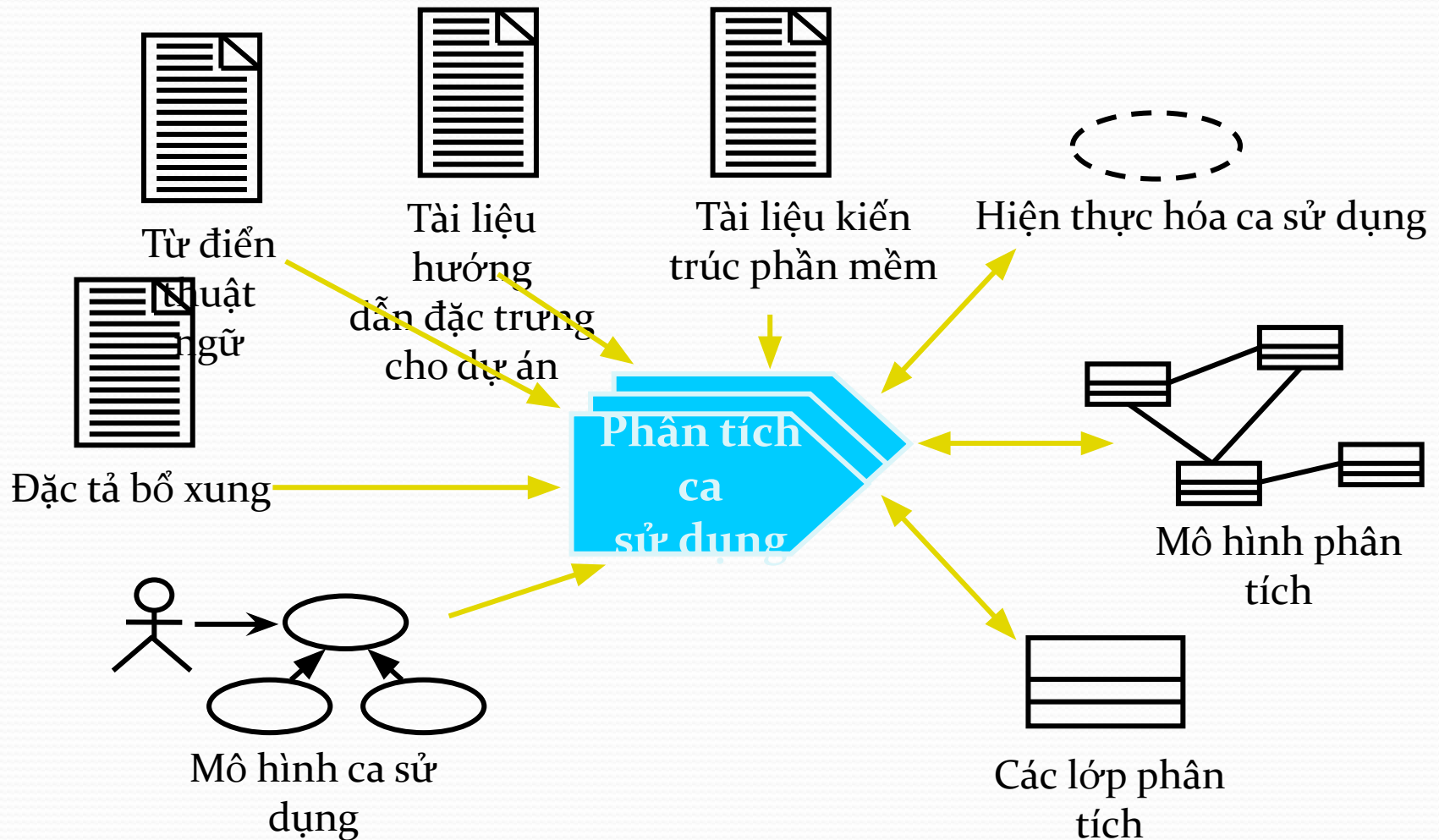
Email: [namnv@tlu.edu.vn](mailto:namnv@tlu.edu.vn)

Website: [namvannguyen.blogspot.com](http://namvannguyen.blogspot.com)

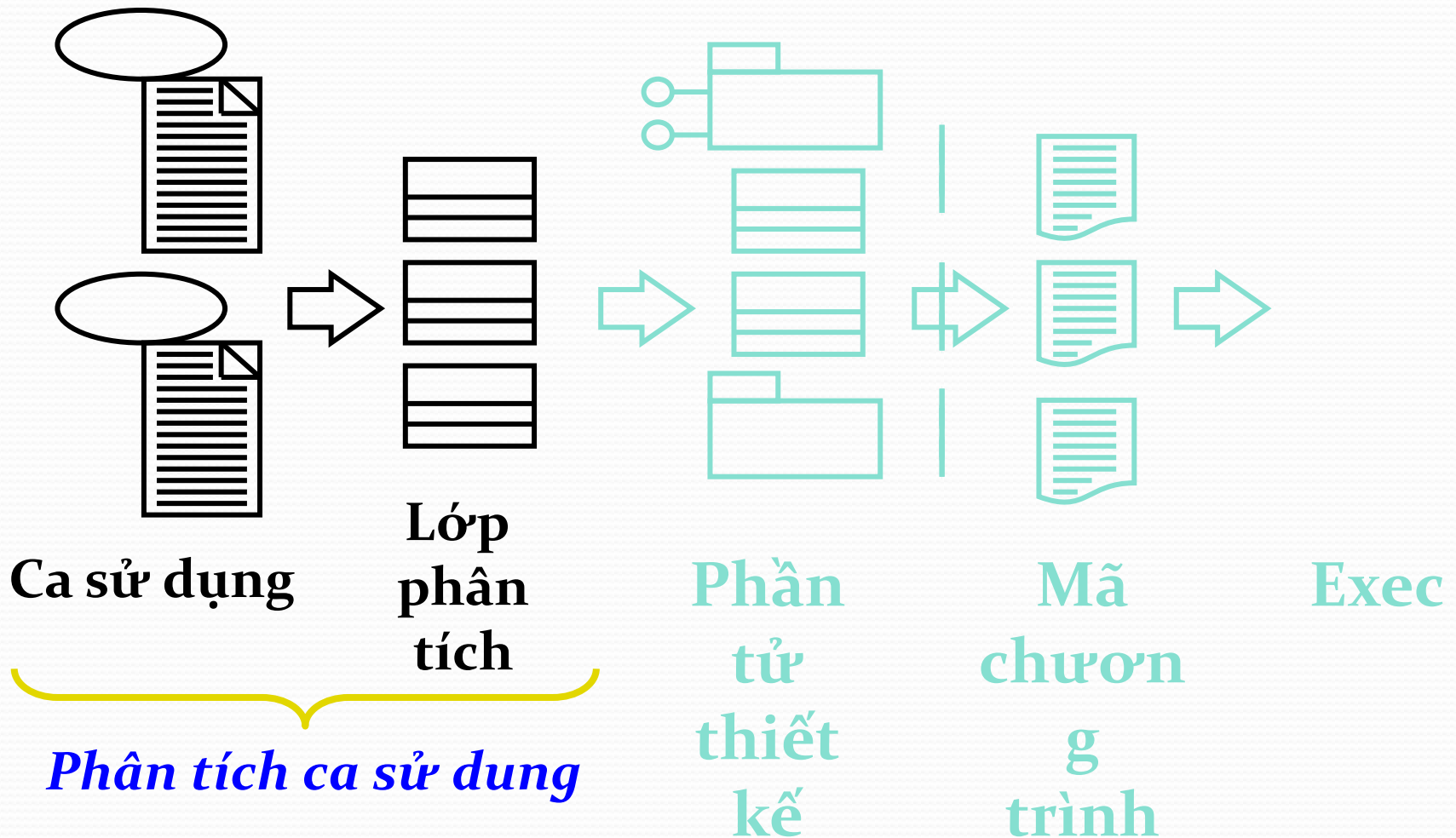
# Nội dung

- Tổng quan
- Xác định lớp phân tích

# Tổng quan phân tích ca sử dụng

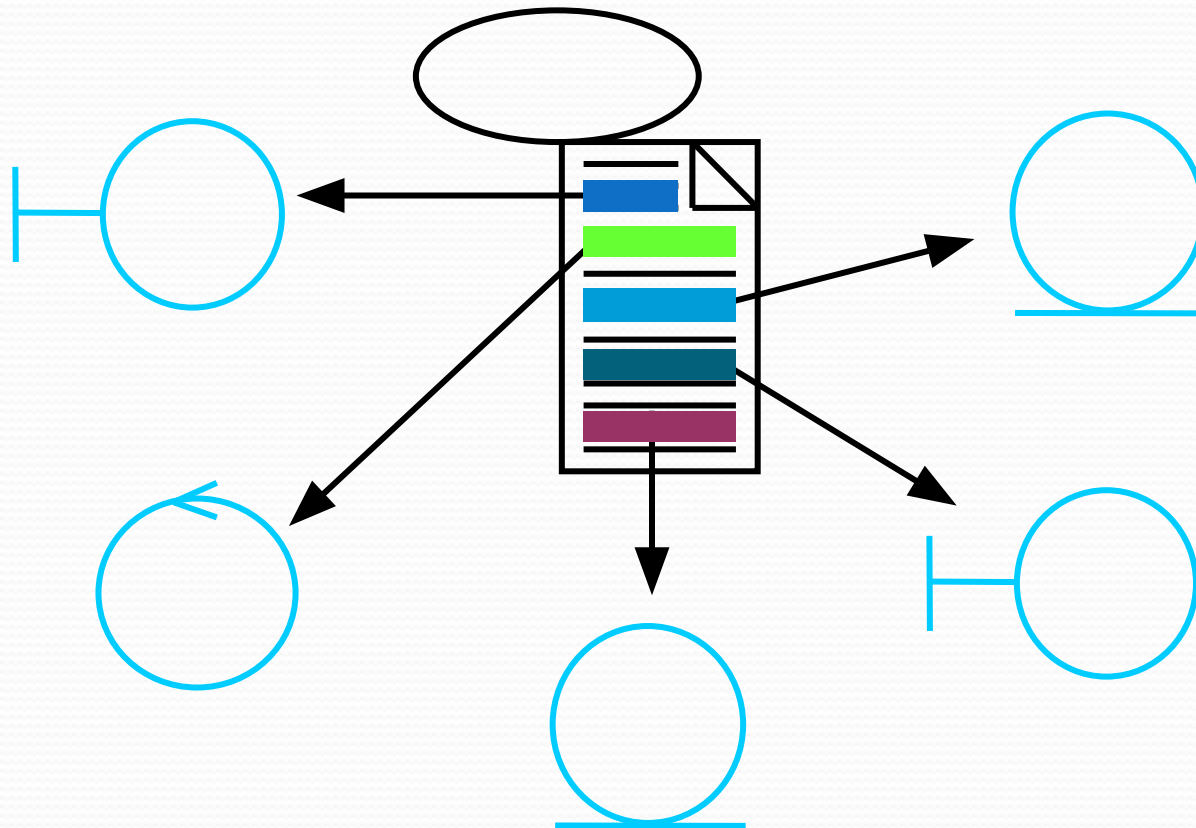


# Lớp phân tích: Bước đầu tiến tới các phần tử có thể thực thi

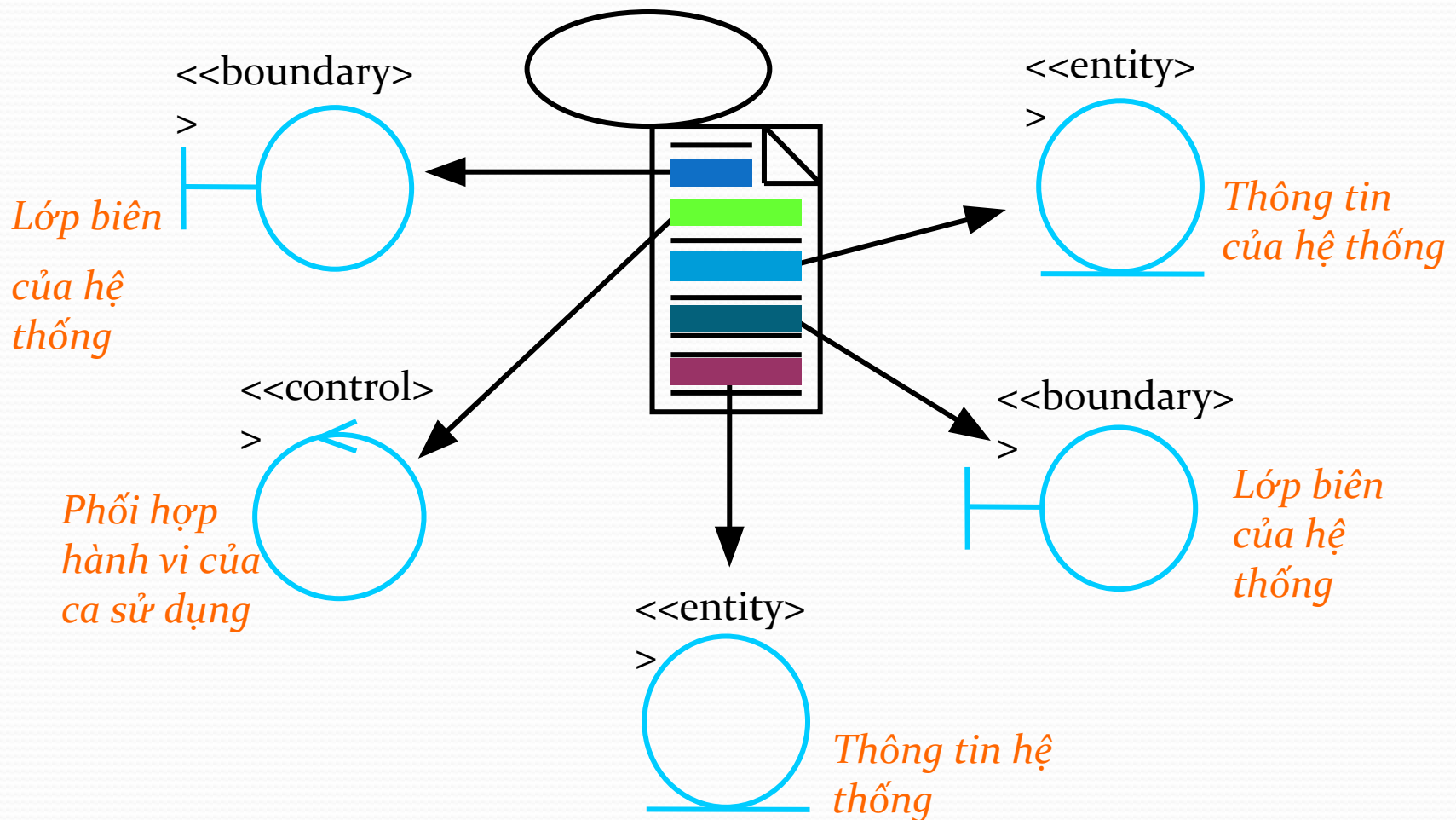


# Tìm các lớp trong hành của ca sử dụng

- Toàn bộ hành vi của ca sử dụng cần phải được phân phối vào các lớp phân tích



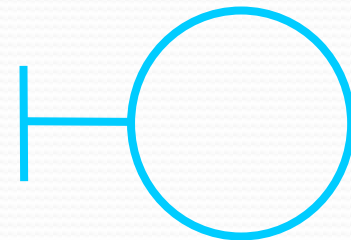
# Lớp phân tích là gì?



# Lớp biên là gì?

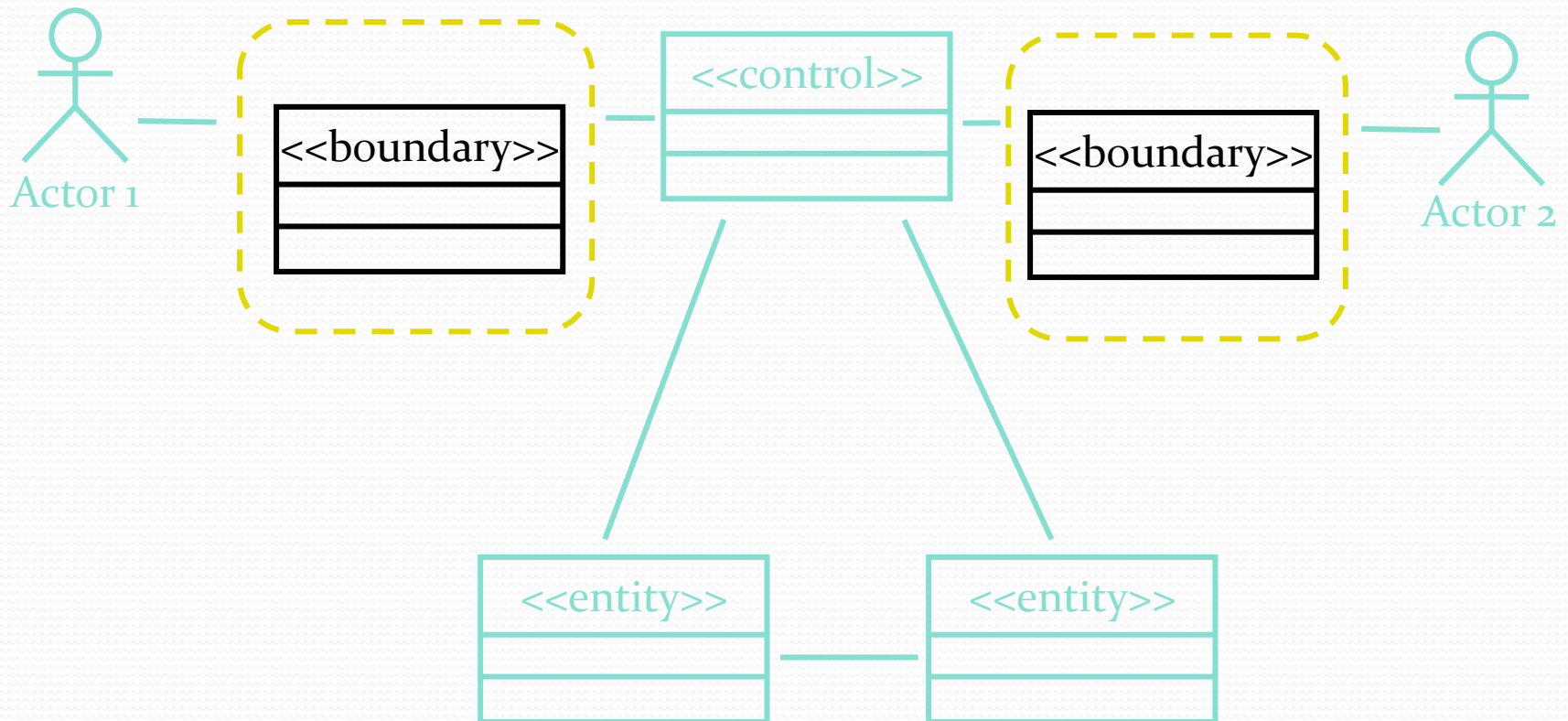
- Là cầu nối giữa giao diện và những thứ bên ngoài hệ thống
- Một số kiểu lớp biên
  - Lớp giao diện người dùng
  - Lớp giao diện hệ thống
  - Lớp giao diện thiết bị

*Analysis class  
stereotype*



*Phụ thuộc môi trường*

# Vai trò của các lớp biên

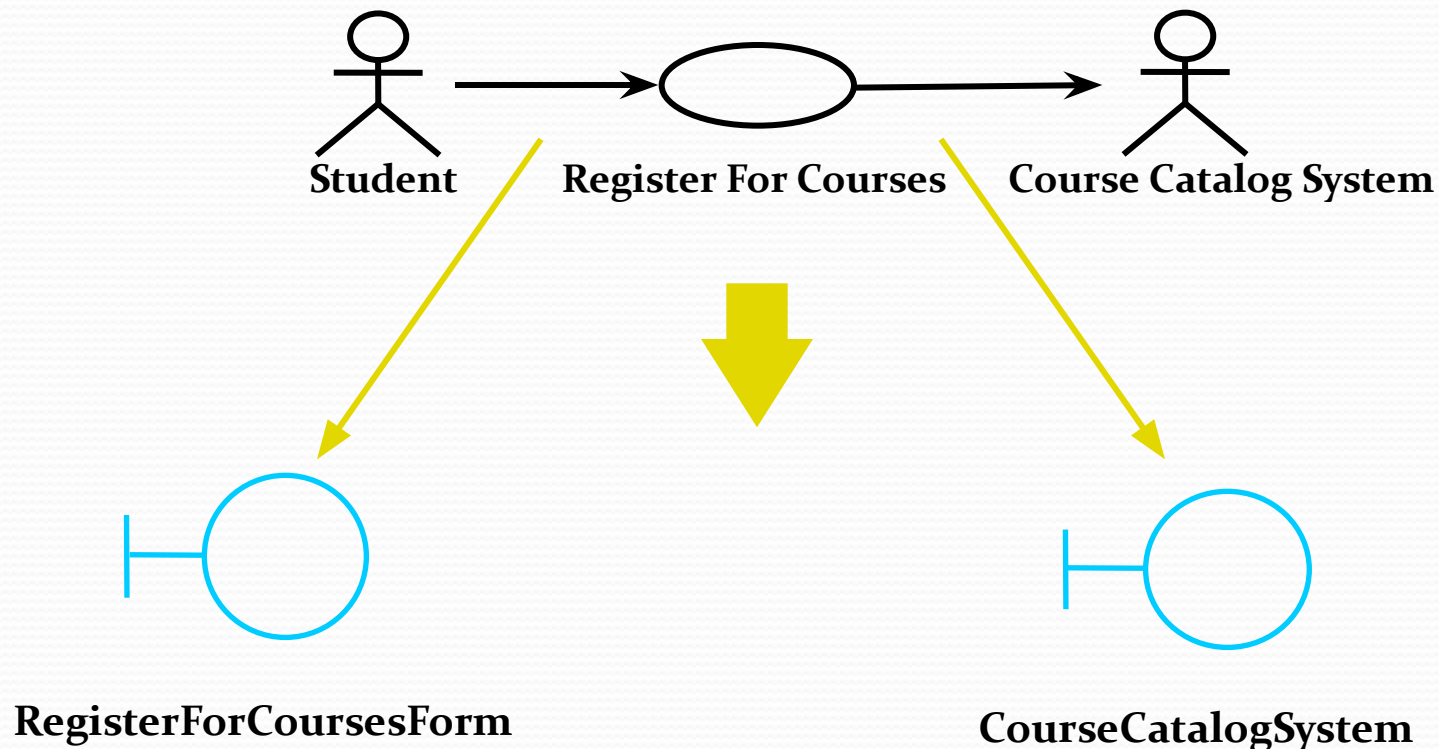


*Mô hình hóa tương tác giữa hệ thống với môi trường của nó*



# Ví dụ: Tìm kiếm các lớp biên

- Thông thường, xác định được một lớp biên trên mỗi cặp tác nhân/ca sử dụng



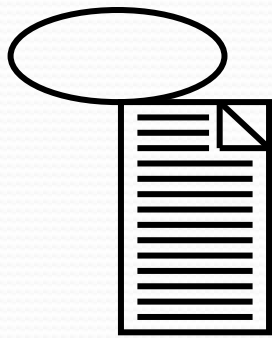
# Lớp Biên

- Lớp giao diện người dùng (User Interface Classes)
  - Tập trung vào thông tin nào được biểu diễn cho người dùng
  - KHÔNG tập trung vào chi tiết của giao diện người dùng
- Lớp giao diện hệ thống và thiết bị
  - Tập trung vào giao thức nào cần được xác định
  - KHÔNG tập trung vào việc các giao thức được cài đặt ra sao

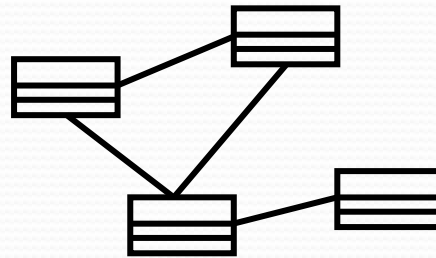
*Tập trung vào vai trò, trách nhiệm, không tập trung vào tiểu tiết!*

# Lớp thực thể là gì?

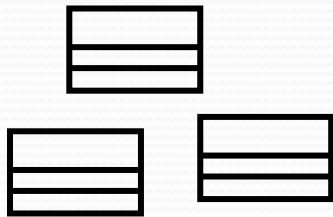
Những trừu tượng hóa chính của hệ thống



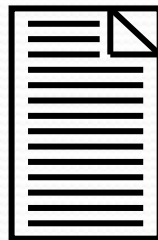
Ca sử dụng



Mô hình miền nghiệp vụ



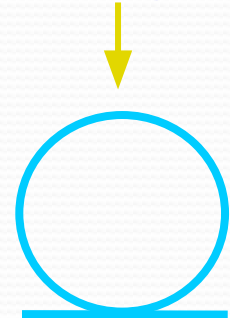
Những trừu tượng hóa  
trong quá trình phân  
tích kiến trúc



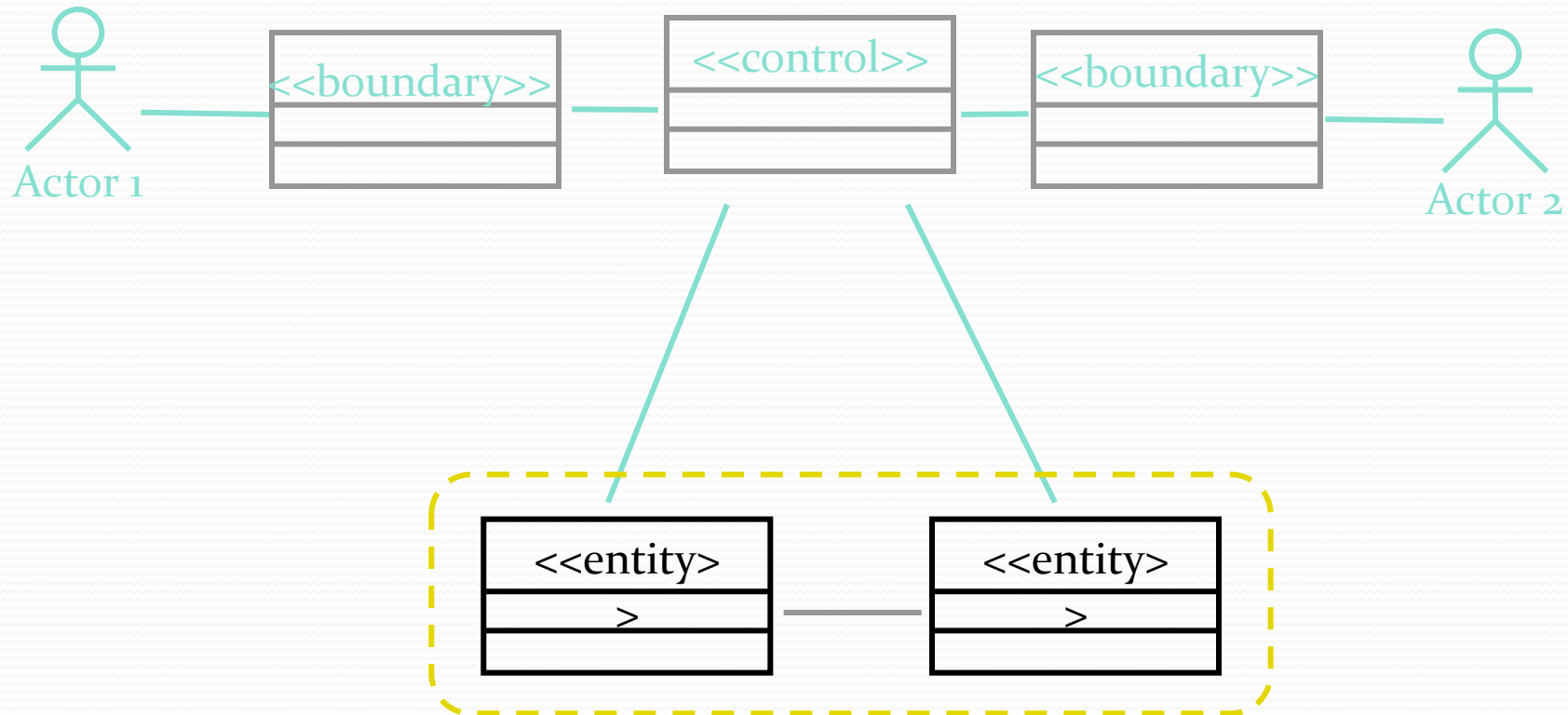
Từ điển thuật ngữ



*Analysis class  
stereotype*



# Vai trò của lớp thực thể



*Lưu trữ và quản lý thông tin trong hệ thống*

# Cách tìm các lớp thực thể

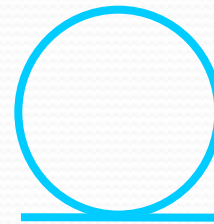
- Sử dụng luồng sự kiện của ca sử dụng như đầu vào
- Xác định các trừu tượng hóa chính của ca sử dụng
  - Gạch chân các cụm danh từ
- Thông thường, sử dụng phương pháp lọc danh từ
  - Gạch chân các mệnh đề danh từ trong luồng sự kiện của ca sử dụng
  - Xóa bỏ những cụm danh từ dư thừa
  - Xóa bỏ những cụm mơ hồ, không rõ ràng
  - Xóa bỏ tác nhân (nằm ngoài phạm vi hệ thống)
  - Xóa bỏ những cấu trúc cài đặt
  - Xóa bỏ thuộc tính (sử dụng trong giai đoạn sau)
  - Xóa bỏ hoạt động (operations)

# Ví dụ: Các lớp thực thể

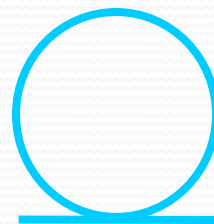
- Ca sử dụng: Đăng ký môn học
  - Register for Courses (Create Schedule)



CourseOffering



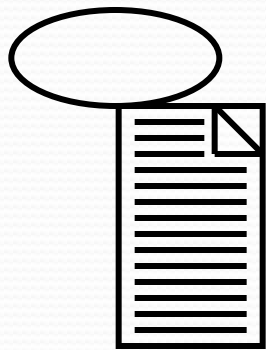
Schedule



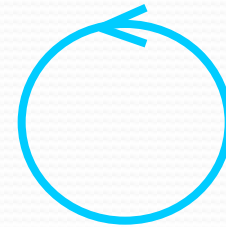
Student

# Lớp điều khiển là gì?

- Phối hợp hành vi của ca sử dụng
  - Các ca sử dụng phức tạp thường đòi hỏi một hoặc nhiều lớp điều khiển

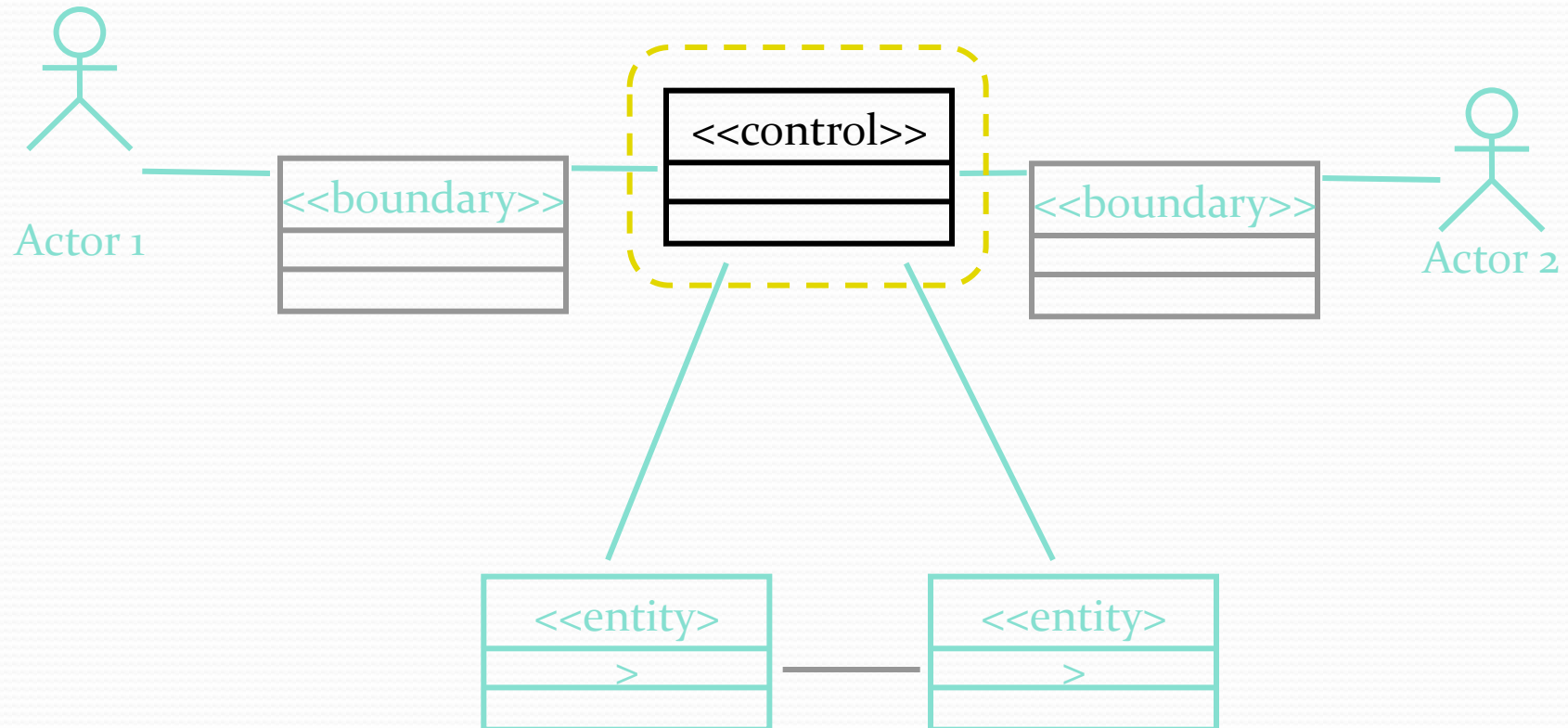


Ca sử dụng



*Analysis class  
stereotype*

# Vai trò của một lớp điều khiển

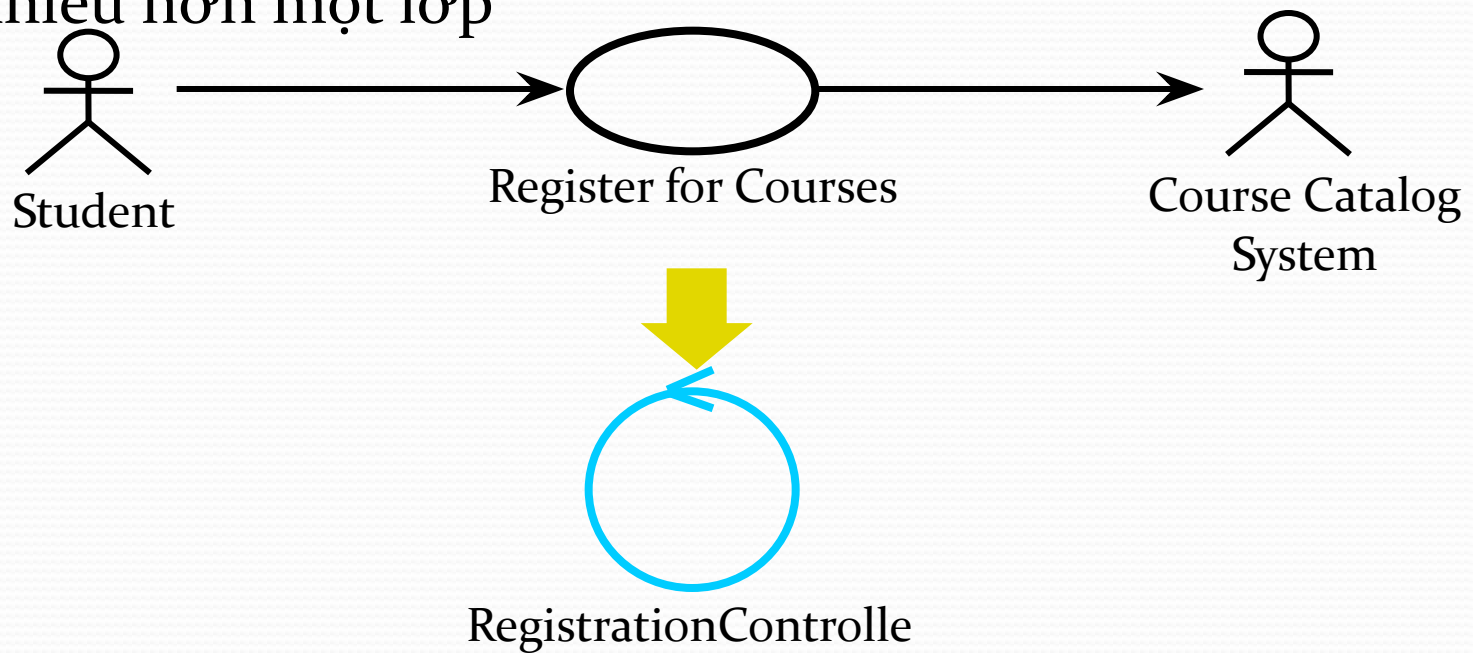


*Phối hợp hành vi của ca sử dụng*



# Ví dụ: Tìm các lớp điều khiển

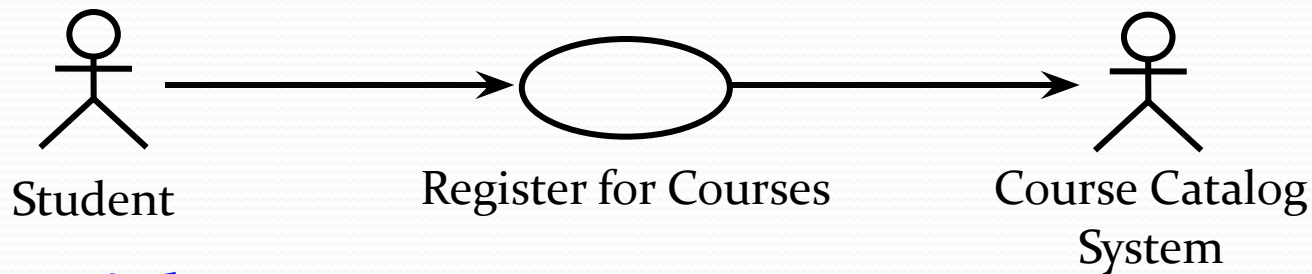
- Thông thường, xác định một lớp điều khiển trên mỗi ca sử dụng
- Khi việc phân tích vẫn tiếp tục, một lớp điều khiển của một ca sử dụng phức tạp có thể được phát triển thành nhiều hơn một lớp



# Đặc điểm các hành vi của lớp điều khiển

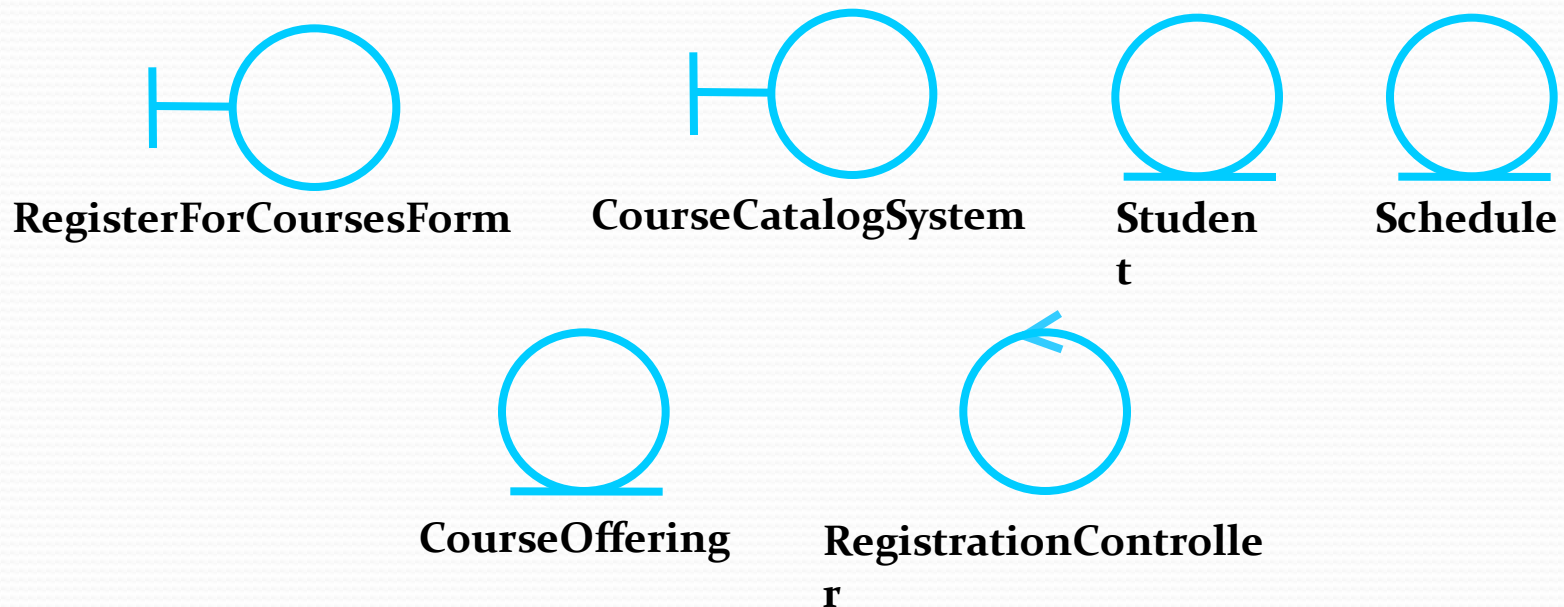
- Hành vi thường độc lập với bên ngoài (môi trường hệ thống)
- Hành vi xác định logic điều khiển và tổ chức các giao dịch trong ca sử dụng
- Hành vi ít bị thay đổi khi cấu trúc và hành vi bên trong lớp thực thể thay đổi.
- Hành vi sử dụng hay thiết lập nội dung của các lớp thực thể.
- Hành vi thường không là duy nhất, tùy theo từng tình huống, kịch bản ca sử dụng.

# Ví dụ: Tóm tắt: Các lớp phân tích



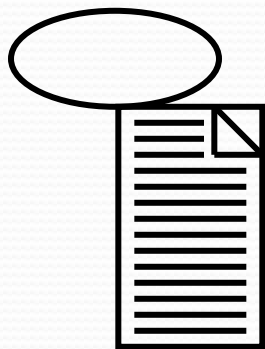
## Mô hình ca sử dụng

## Mô hình thiết kế

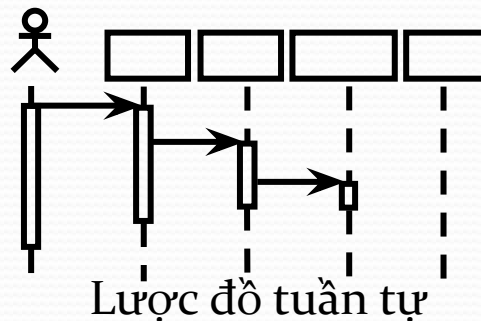
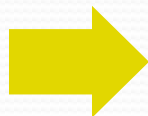


# Phân phối hành vi của ca sử dụng vào các lớp phân tích

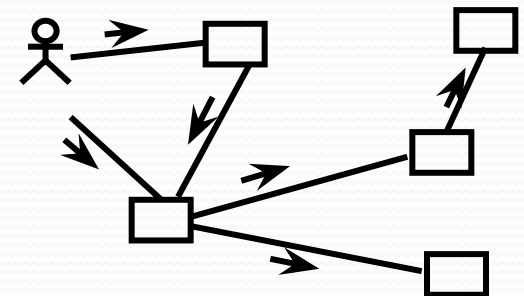
- Với mỗi luồng sự kiện của ca sử dụng
  - Xác định các lớp phân tích
  - Phân phối trách nhiệm ca sử dụng cho các lớp phân tích
  - Mô hình tương tác của các lớp phân tích trong các lược đồ tương tác



Ca sử dụng



Lược đồ tuần tự



Lược đồ cộng tác

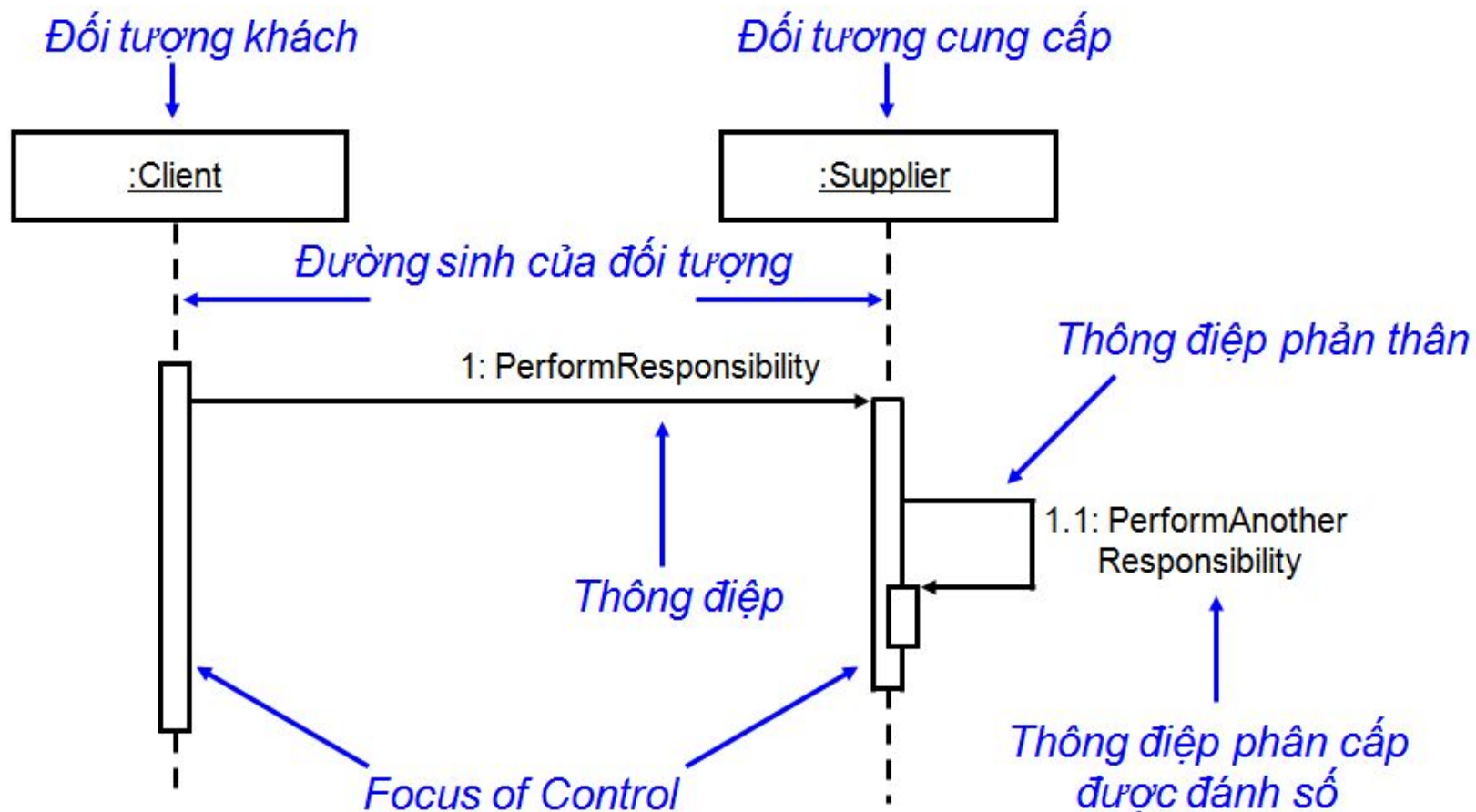
Hiện thực hóa ca sử dụng

# Hướng dẫn: Phân phối các trách nhiệm cho các lớp phân tích

- Sử dụng các kiểu mở rộng (stereotypes) để định hướng
  - Lớp biên (Boundary Classes)
    - Hành vi liên quan tới giao tiếp với tác nhân
  - Lớp thực thể (Entity Classes)
    - Hành vi liên quan đến đóng gói dữ liệu bên trong các trừu tượng
  - Lớp điều khiển (Control Classes)
    - Hành vi đặc trưng cho một ca sử dụng hoặc một phần của một luồng sự kiện rất quan trọng

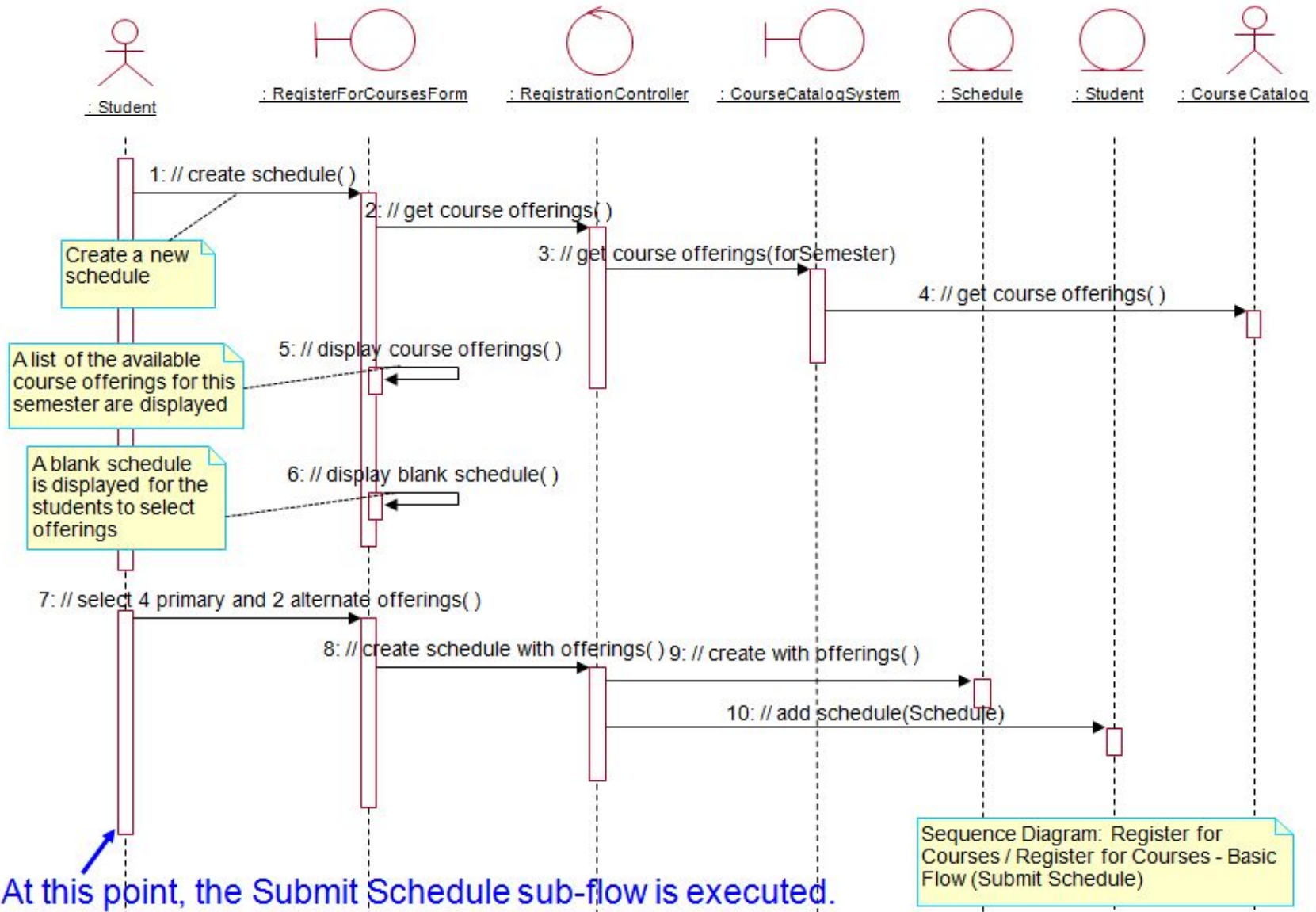
- Ai là người có dữ liệu cần thiết để thực hiện các trách nhiệm (responsibility)?
  - Nếu một lớp có dữ liệu, đặt trách nhiệm đi kèm với dữ liệu
    - Nếu nhiều lớp có dữ liệu
  - Gán trách nhiệm với một lớp và thêm vào một quan hệ với lớp khác
    - Tạo một lớp mới, đặt trách nhiệm vào trong lớp mới, thêm vào các quan hệ với các lớp cần thiết để thực hiện trách nhiệm đó
    - Đặt trách nhiệm vào trong lớp điều khiển, và thêm vào các quan hệ với các lớp mà cần thiết để thực hiện trách nhiệm

# Phân tích thành phần cấu thành lược đồ tuần tự



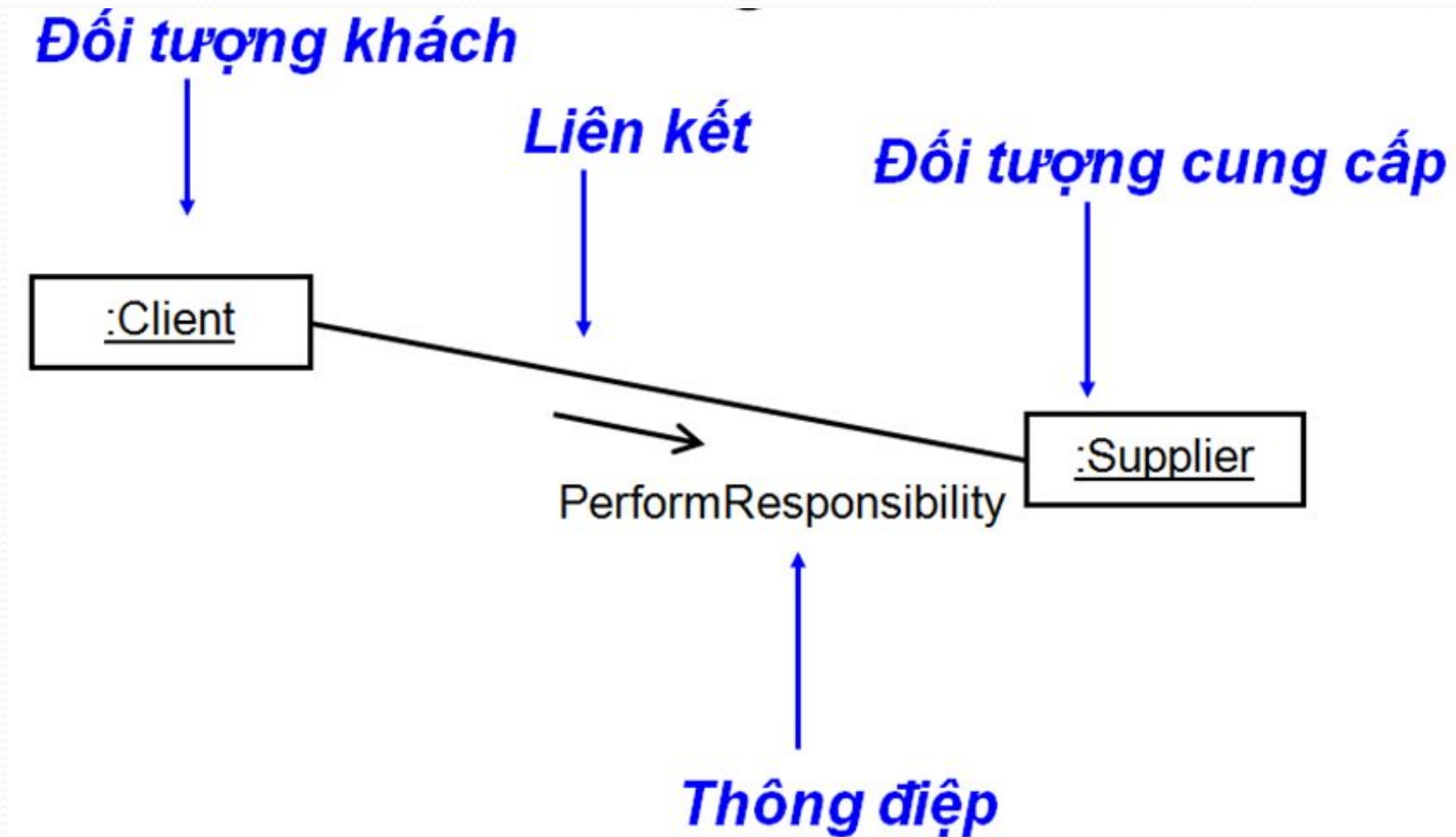


# Ví dụ: lược đồ tuần tự của HT đăng ký học

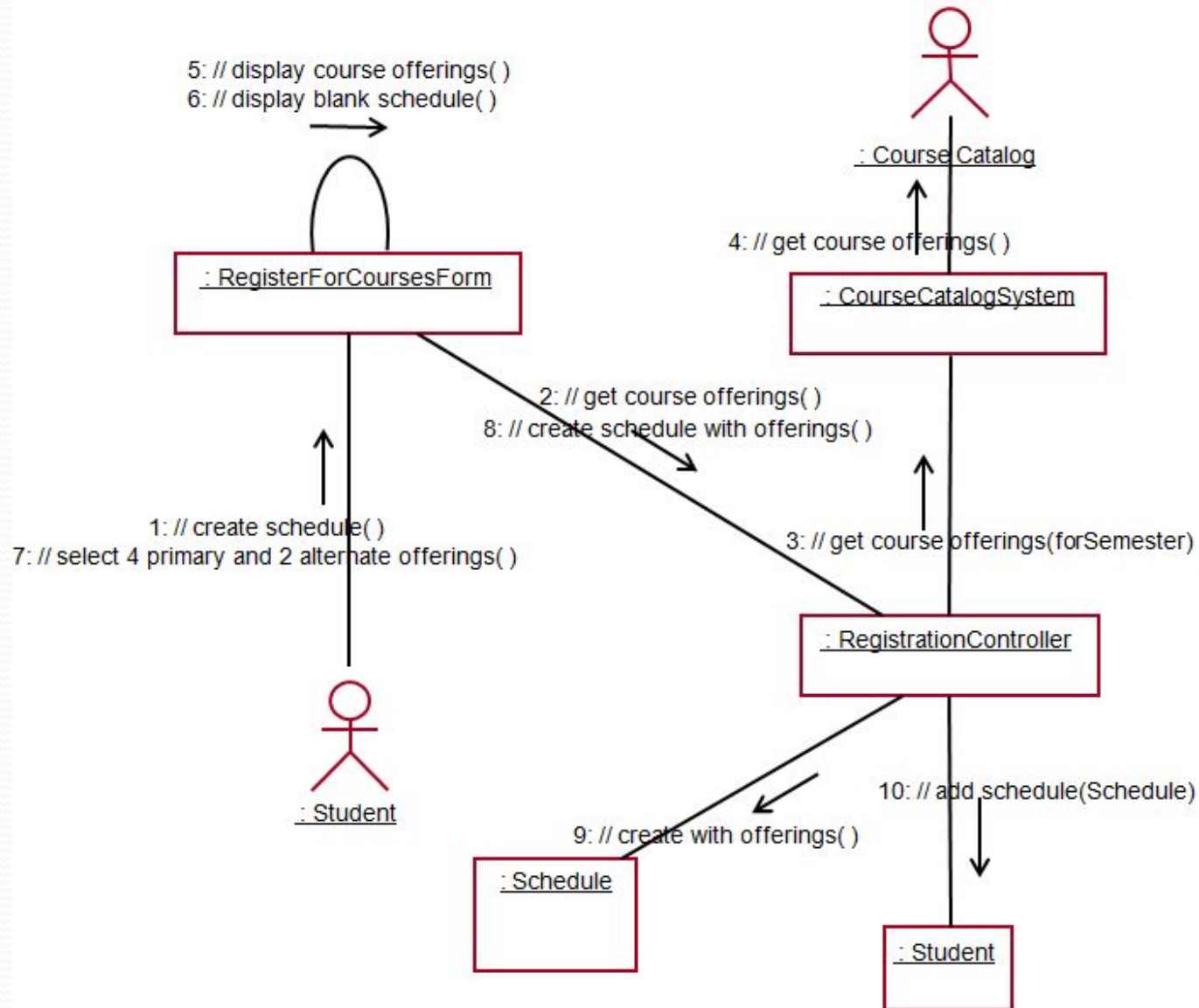




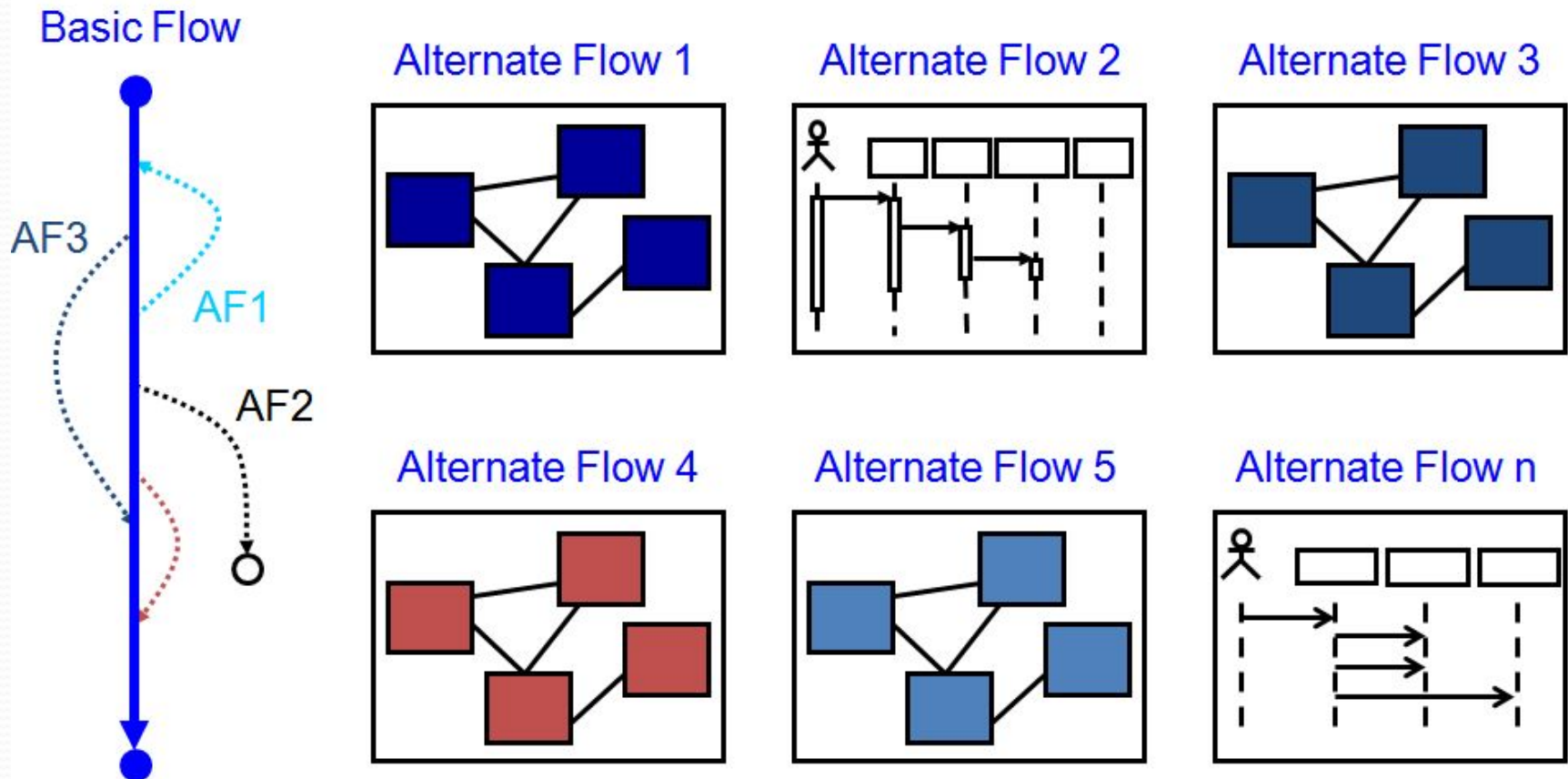
# Phân tích thành phần cấu thành lược đồ cộng tác



# Ví dụ: Biểu đồ hợp tác



# Lược đồ tương tác cần nhiều khía cạnh khác nhau



# Lược đồ cộng tác và Lược đồ tuần tự

## ● Lược đồ cộng tác

- Thể hiện mối quan hệ bổ xung cho các tương tác
- Tốt cho việc mô tả trực quan các mẫu cộng tác
- Tốt cho việc mô tả trực quan tất cả những ảnh hưởng trên một đối tượng nào đó
- Dễ sử dụng trong các giai đoạn động não

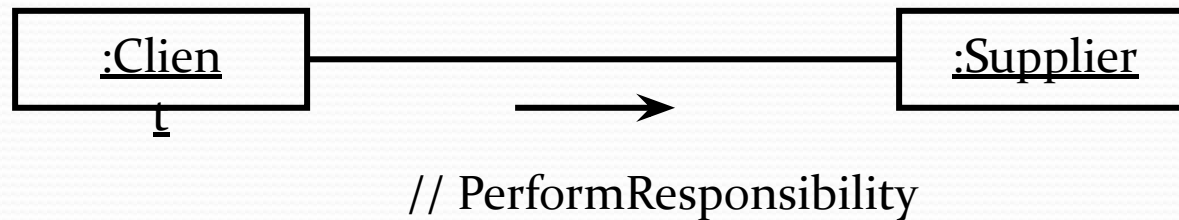
## • Lược đồ tuần tự

- Thể hiện chuỗi thông điệp tương minh
- Tốt cho việc mô tả trực quan toàn bộ luồng sự kiện
- Tốt cho đặc tả thời gian thực và các kịch bản phức tạp

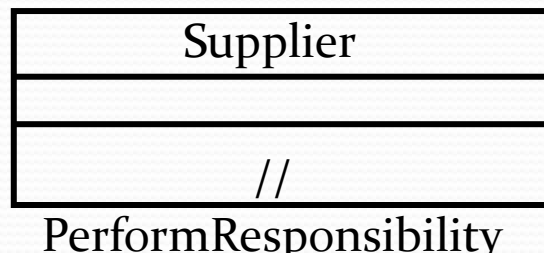
# Mô tả trách nhiệm – Hành vi

- Trách nhiệm là gì (responsibilities)?
- Làm sao để tìm ra trách nhiệm?

Lược đồ tương tác



Lược đồ lớp



# Ví dụ: Lược đồ các lớp tham gia

<<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

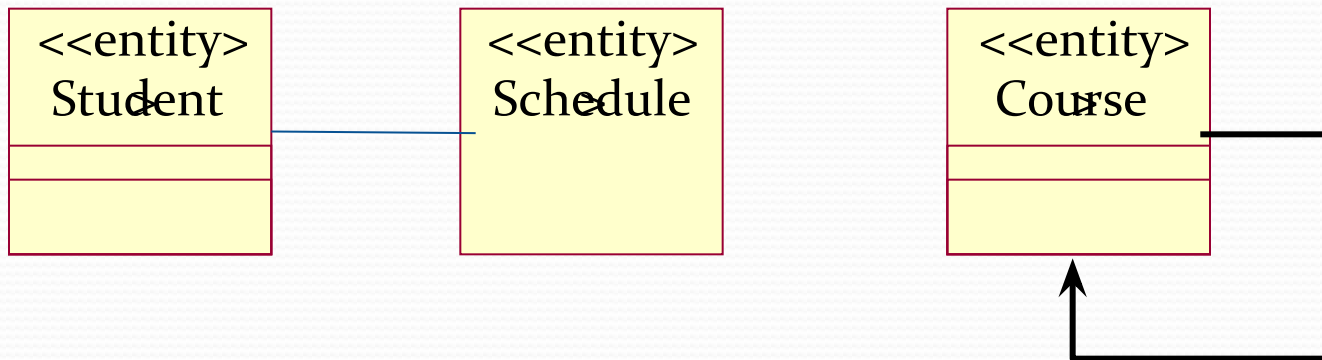
```
// display course offerings()
// display blank schedule()
// update offering selections()
```

# Tìm kiếm thuộc tính

- Thuộc tính/đặc điểm của các lớp được xác định
- Thông tin được lưu giữ bởi các lớp được xác định
- “Danh từ” mà không trở thành lớp
  - Thông tin mà giá trị của nó là những thứ quan trọng
  - Thông tin mà được “sở hữu” duy nhất bởi một đối tượng
  - Thông tin mà không có hành vi

# Thuộc tính và mối liên kết

- Quan hệ kết hợp (Association): Một quan hệ cấu trúc xác định việc các đối tượng của một lớp được kết nối với các đối tượng của lớp khác

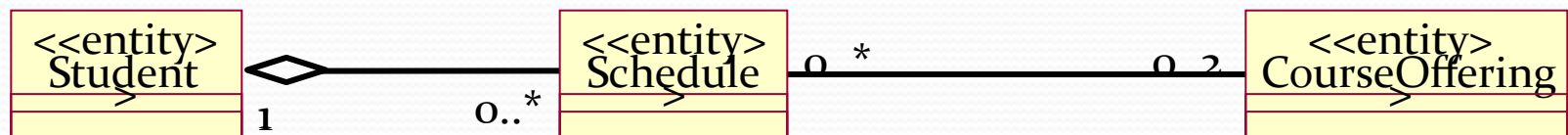




- Quan hệ kết tập (aggregation): Là một dạng kết hợp (association) mô hình hóa quan hệ toàn bộ - bộ phận giữa một toàn bộ và các bộ phận của nó

*Whole/aggregate*

*Part*

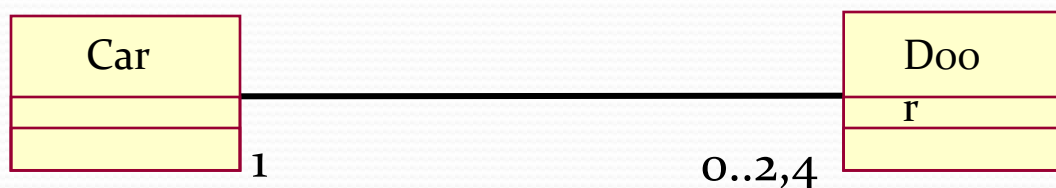


# Kết hợp (association) - kết tập (aggregation)

- Nếu hai đối tượng ràng buộc bởi một quan hệ toàn bộ - bộ phận
  - Quan hệ sẽ là một kết tập



- Nếu hai đối tượng thường được xem là độc lập mặc dù đôi khi chúng được liên kết với nhau
  - Quan hệ khi đó là một kết hợp

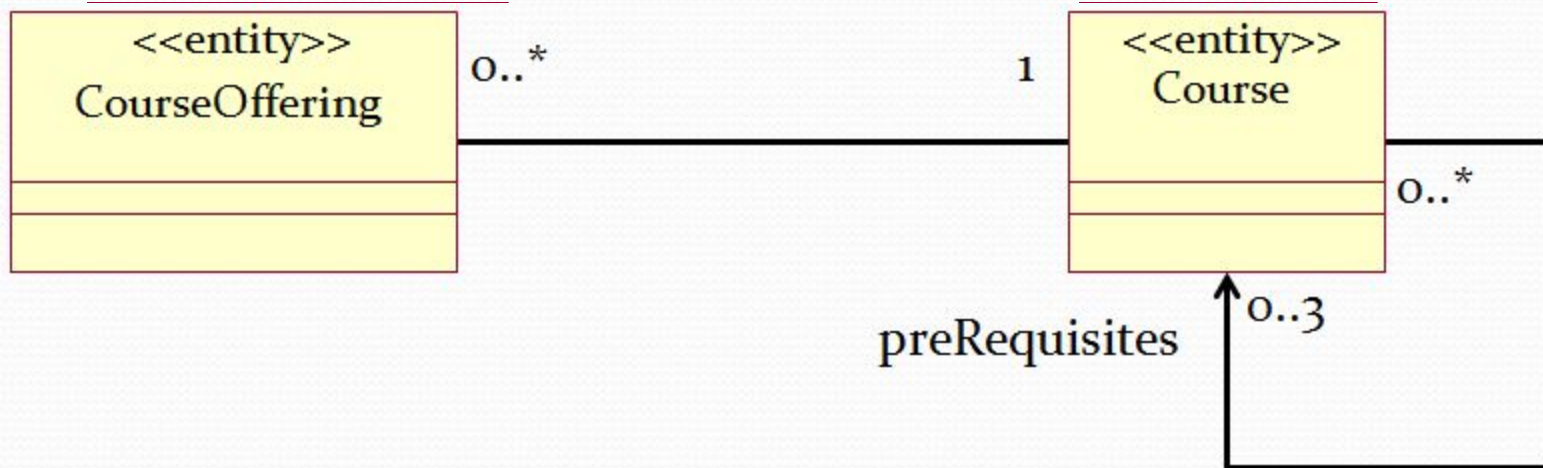


# Multiplicity – Tính phức tạp

Unspecified	
Exactly	1
One or Zero or	0..
More or	*
Zero or	*
More or	1..
One or	*
Zero or One (optional scalar role)	0..
Specified	1
Range	2..
Multiple, Disjoint	4
	2,
	4..6

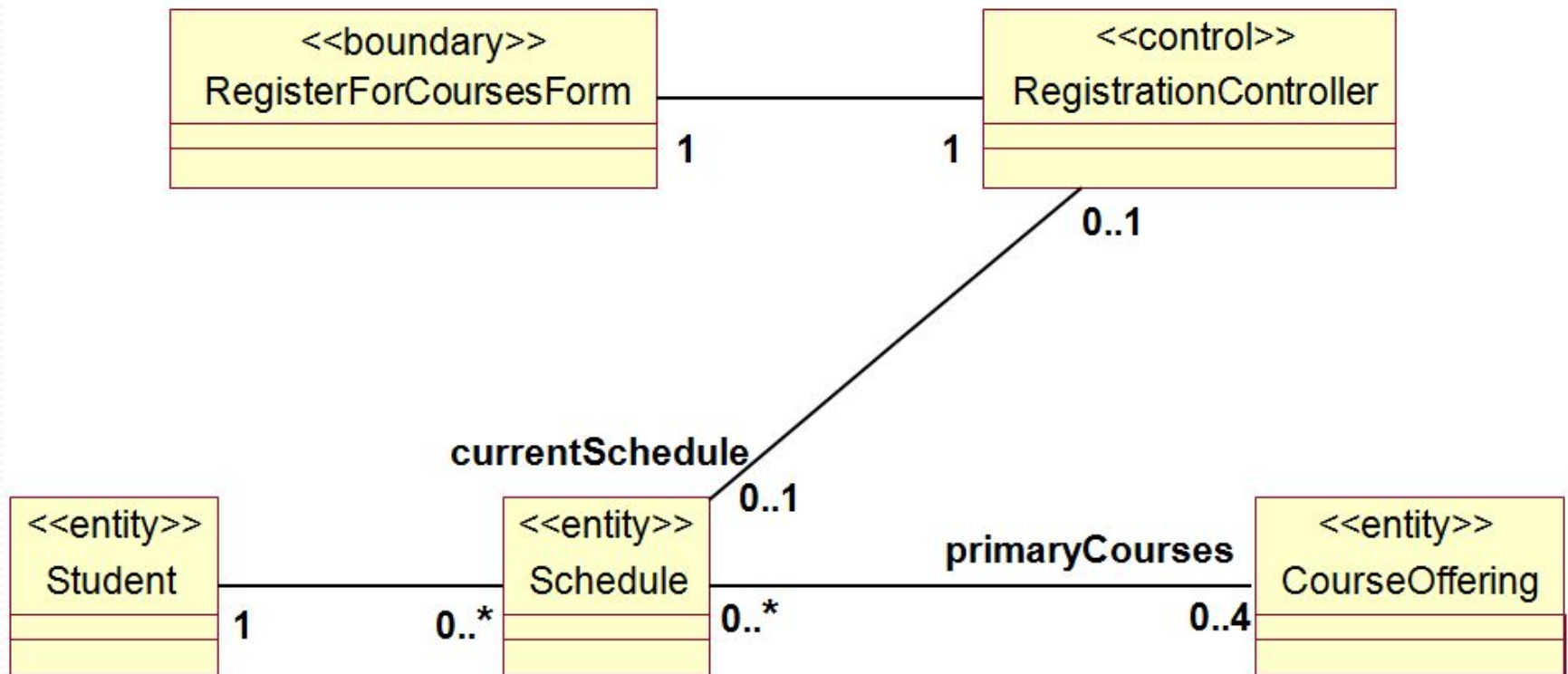
Ranges

- Multiplicity trả lời hai câu hỏi:
  - Kết hợp là thiết yếu hay tùy chọn:
  - Số thể hiện tối đa và tối thiểu có thể được liên kết với một thể hiện?



# VOPC – Tìm kiếm các liên kết giữa các lớp

- VOPC – View of Participating Classes

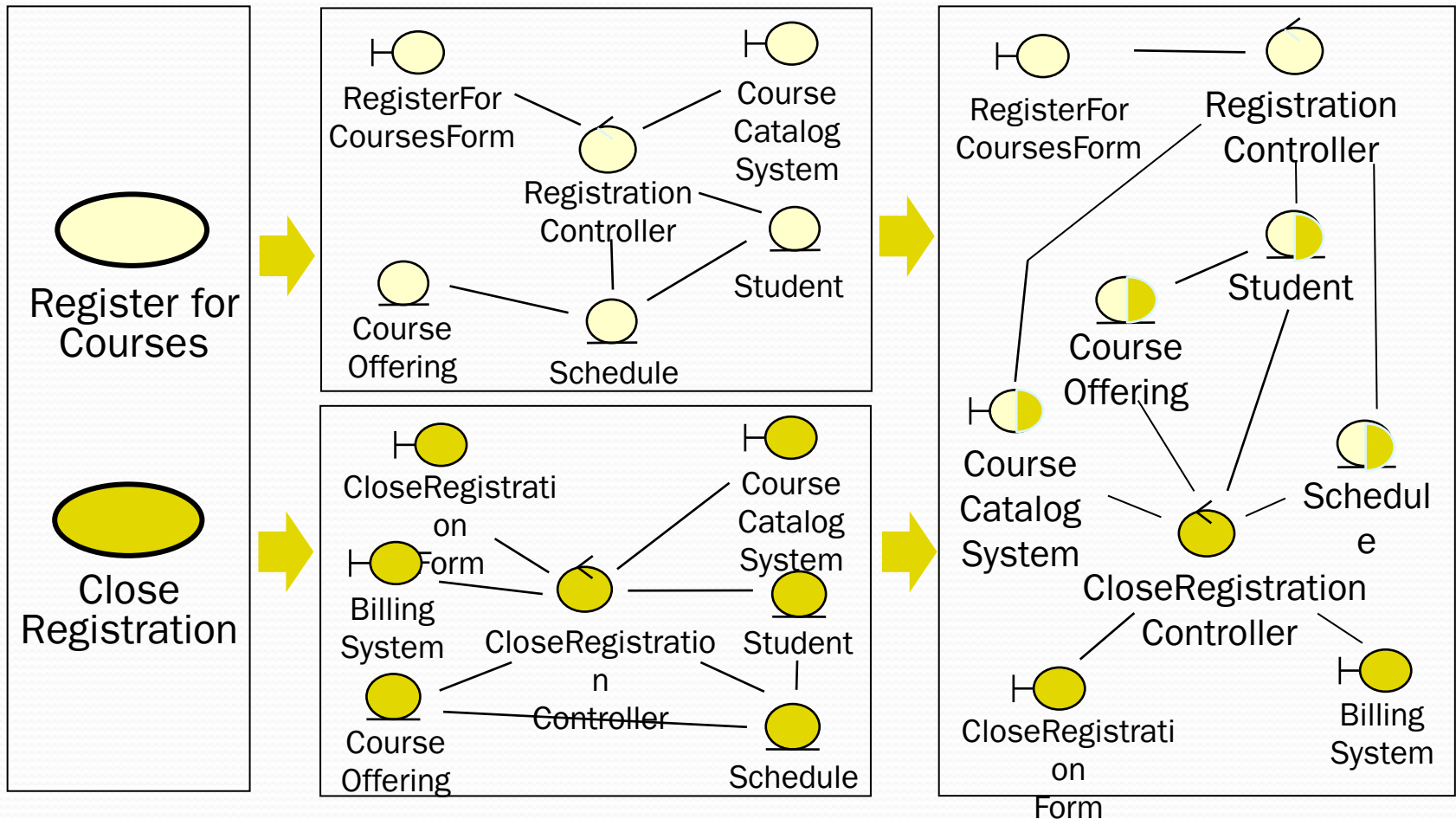


# Cơ chế phân tích

- Xác định giữa lớp phân tích vào các cơ chế phân tích

Lớp phân tích	Các cơ chế phân tích
Student	Lưu trữ bền vững, An ninh
Schedule	Lưu trữ bền vững, An ninh
CourseOffering	Lưu trữ bền vững, Kế thừa giao diện
Course	Lưu trữ bền vững, Kế thừa giao diện
RegistrationController	Phân tán

# Hợp nhất các lớp phân tích





The End.