

PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

Bài 6. Use case

Giáo viên: TS. Trần Mạnh Tuấn

Bộ môn: Hệ thống thông tin

Khoa: Công nghệ thông tin

Email: tmtuan@tlu.edu.vn

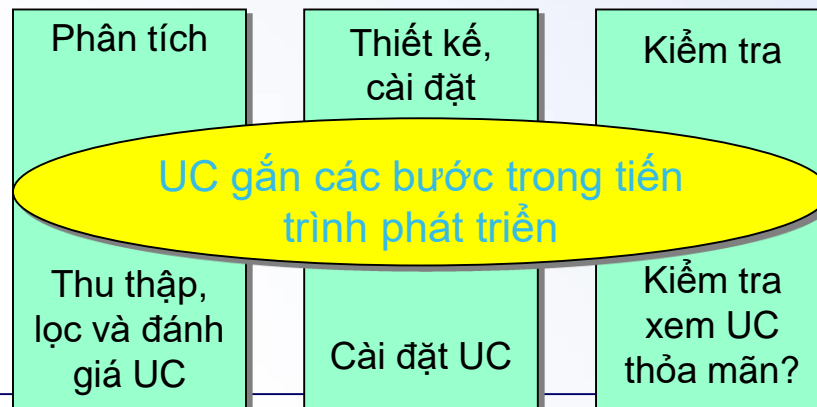
Điện thoại: 0983.668.841

Nội dung

- 1. Các thành phần Use-case**
- 2. Các quan hệ của Use-case**
- 3. Cách biểu diễn một usecase**

Các thành phần Use-case

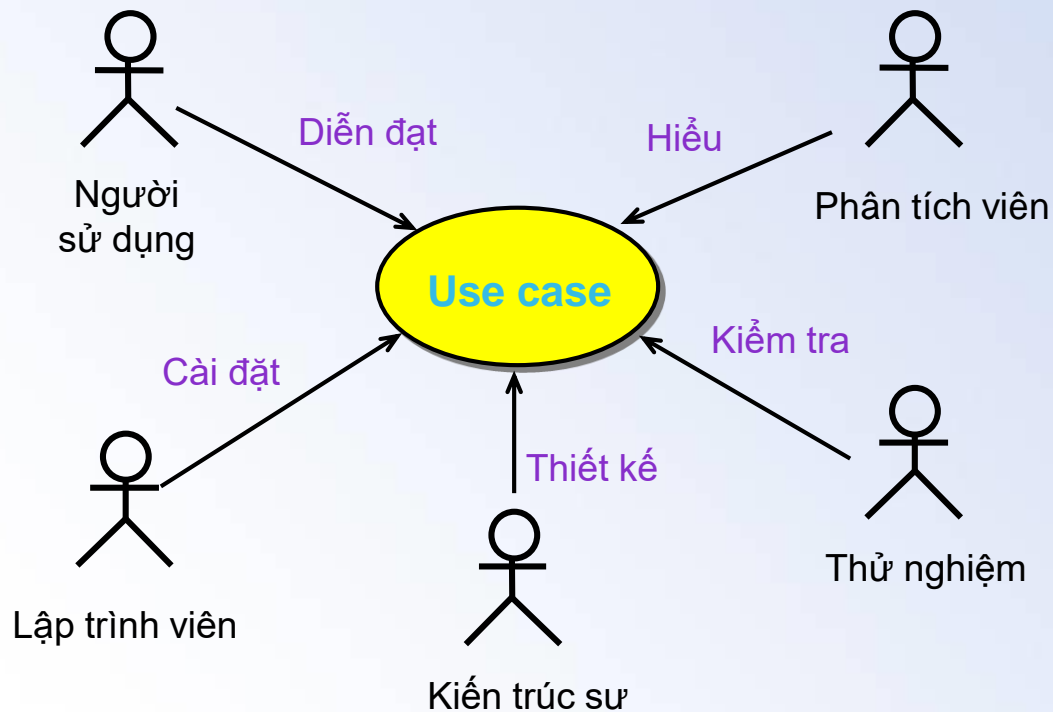
- ❖ **Hình thành và mô tả yêu cầu chức năng hệ thống**
 - Là kết quả thỏa thuận giữa khách hàng và người phát triển hệ thống phần mềm
- ❖ **Cho phép mô tả rõ ràng và nhất quán cái hệ thống sẽ làm**
 - Mô hình có khả năng được sử dụng xuyên suốt quá trình phát triển
- ❖ **Cung cấp cơ sở để kiểm tra, thử nghiệm hệ thống**
- ❖ **Cho khả năng dễ thay đổi hay mở rộng yêu cầu hệ thống**



UC và tiến trình phát triển

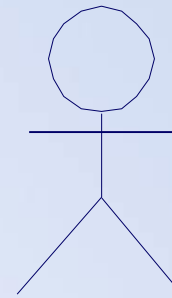
Các thành phần Use-case

Ai quan tâm đến UC?



Các thành phần Use-case

- ❖ Một tác nhân (actor) đại diện cho bất cứ thứ gì tương tác với hệ thống.
- ❖ Một use-case mô tả một chuỗi các sự kiện mà được thực hiện bởi hệ thống, và nó sinh ra một kết quả có thể quan sát được đối với một tác nhân cụ thể nào đó



Actor



Use Case

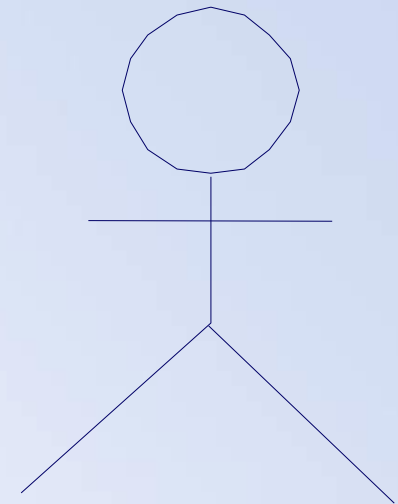
Các thành phần Use-case

	Mô hình hóa nghiệp vụ	Mô hình hóa hệ thống
Use case	Mô tả cái nghiệp vụ làm	Mô tả cái mà hệ thống bên trong nghiệp vụ làm
Actor	Bên ngoài tổ chức	Bên ngoài hệ thống (có thể bên trong tổ chức)
Business worker	Bên trong tổ chức	Không sử dụng

Các thành phần Use-case

Actor

- Tác nhân (actor) biểu diễn vai trò có thể được đảm nhiệm của một người dùng đối với hệ thống.
- Tác nhân có thể biểu diễn cho một con người, máy, hoặc hệ thống khác.
- Tác nhân có thể chủ động trao đổi thông tin với hệ thống.
- Tác nhân có thể là đối tượng cung cấp thông tin.
- Tác nhân có thể bị động nhận thông tin.
- Tác nhân không phải là một phần của hệ thống.
 - Tác nhân là thành phần bên ngoài.

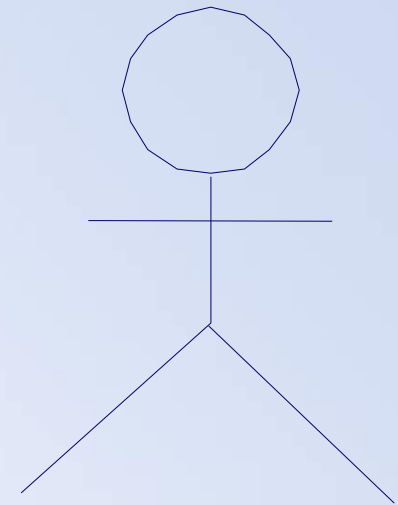


Actor

Các thành phần Use-case

Actor

- Mô tả ai, cái gì tương tác với hệ thống
- Ba loại:
 - Ai: con người sử dụng trực tiếp hệ thống
 - Cái gì: hệ thống khác tương tác với hệ thống đang xây dựng
 - Thời gian: khi đồng hồ khởi sự sự kiện của hệ thống
- Đặt tên: theo vai trò, không theo tên cụ thể vì nó là lớp



Actor

Các thành phần Use-case

Tìm kiếm Actor

❖ Hãy trả lời các câu hỏi sau để tìm ra tác nhân hệ thống (6)

- Ai sẽ sử dụng chức năng chính của hệ thống?
- Ai giúp hệ thống làm việc hàng ngày?
- Ai quản trị, bảo dưỡng để hệ thống làm việc liên tục?
- Hệ thống quản lý thiết bị phần cứng nào?
- Hệ thống đang xây dựng tương tác với hệ thống khác nào?
- Ai hay cái gì quan tâm đến kết quả hệ thống cho lại?

Các thành phần Use-case

Use-case

- ❖ Ca sử dụng (use-case) là đơn vị chức năng có thể quan sát từ bên ngoài, được cung cấp bởi một phân lớp (được gọi là chủ thể).
- ❖ Ca sử dụng được diễn đạt bằng chuỗi các thông điệp giữa chủ thể và một hay nhiều tác nhân của chủ thể.
- ❖ Tiêu chí phân biệt một use-case với chức năng của hệ thống là kết thúc ca sử dụng, tác nhân có thể thu được một kết quả có thể quan sát được.

Các thành phần Use-case

Use-case

- ❖ Một use-case được sử dụng để mô hình hóa một cuộc hội thoại giữa một hay nhiều actor với hệ thống.
- ❖ Một use-case mô tả các hành động (action) mà hệ thống thực hiện để tạo ra một kết quả cho tác nhân.



Use Case

Các thành phần Use-case

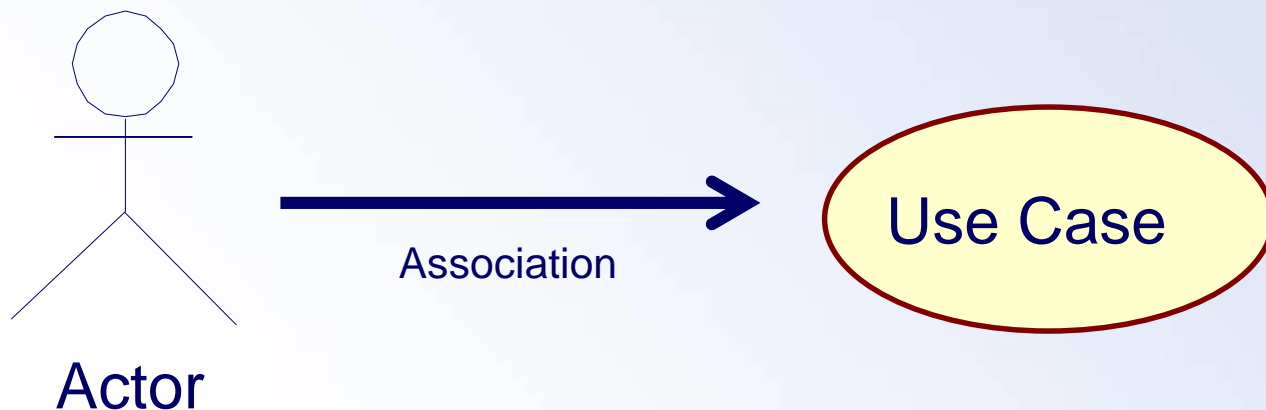
Tìm kiếm Use-case

- ❖ **Với mỗi tác nhân đã tìm ra, hãy trả lời các câu hỏi sau để tìm ra các Use case hệ thống**
 - Tác nhân yêu cầu hệ thống thực hiện chức năng nào?
 - Tác nhân cần đọc, tạo lập, bãi bỏ, lưu trữ, sửa đổi các thông tin nào trong hệ thống?
 - Tác nhân cần thông báo cho hệ thống sự kiện xảy ra trong nó?
 - Hệ thống cần thông báo cái gì đó cho tác nhân?
 - Hệ thống cần vào/ra nào? Vào/ra đi đến đâu hay từ đâu?
- ❖ **Đặt tên UC hệ thống**
 - Theo khái niệm nghiệp vụ của tổ chức
 - Không sử dụng từ kỹ thuật, chuyên môn
 - Sử dụng các động từ, cụm từ ngắn gọn
- ❖ **Tùy theo tầm cỡ dự án mà mỗi hệ thống có từ 20-70 UC**

Quan hệ trong biểu đồ Use-case

Quan hệ kết hợp giữa Use-case với Actor

- ❖ Một use-case mô hình hóa cuộc hội thoại giữa tác nhân và hệ thống.
- ❖ Một use-case được khởi tạo bởi một tác nhân để yêu cầu một chức năng cụ thể nào đó trong hệ thống.

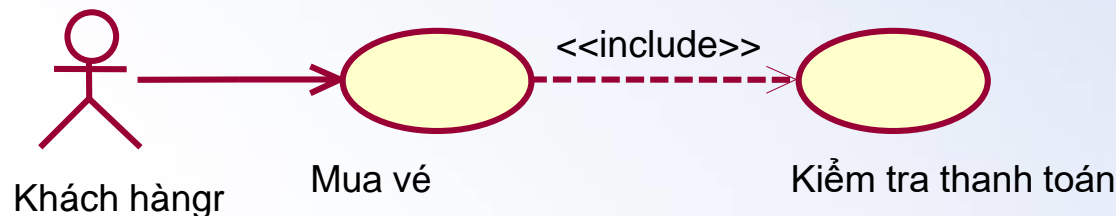
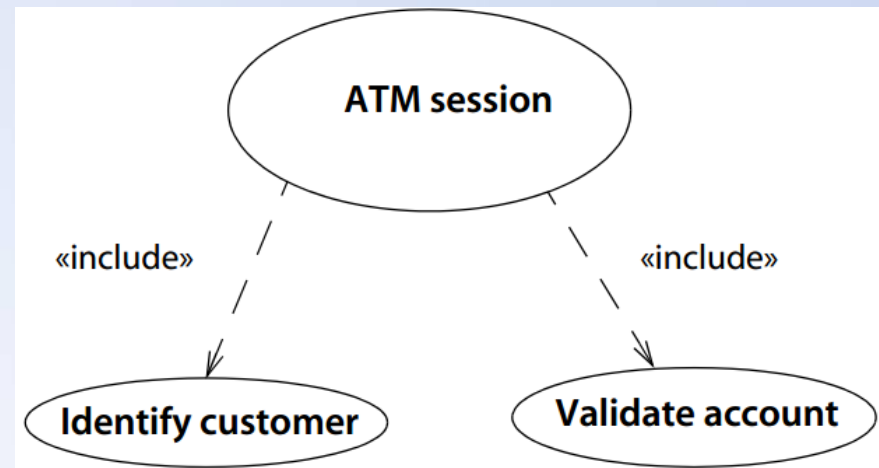


Quan hệ trong biểu đồ Use-case

Quan hệ gộp của Use-case

❖ Include:

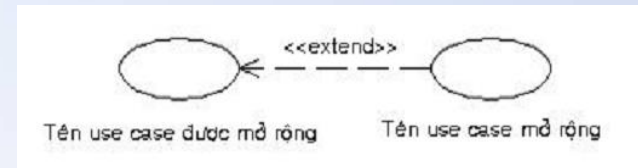
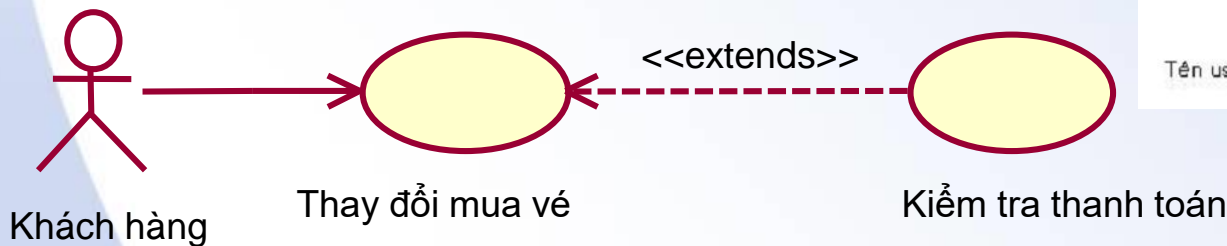
- Include có nghĩa là bao gộp một chuỗi các hành động cụ thể trong use-case cơ sở.
- Ca sử dụng được bao gộp (included use-case) có thể được sử dụng độc lập với các ca sử dụng cơ sở.



Quan hệ trong biểu đồ Use-case

Quan hệ mở rộng của Use-case

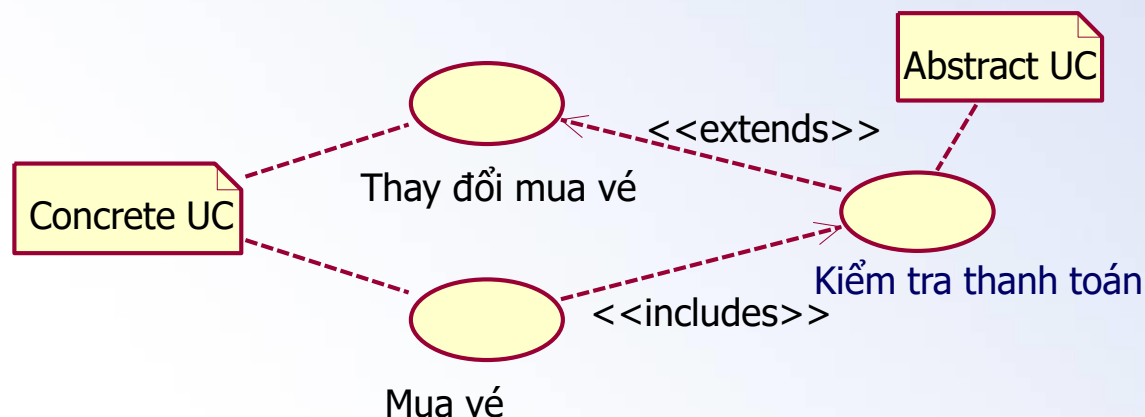
- Một ca sử dụng có thể được định nghĩa như là một sự mở rộng tăng dần của một lớp ca sử dụng cơ sở (base use case). Quan hệ này được gọi là quan hệ mở rộng.
- Là một quan hệ từ một use-case mở rộng (extension use-case) tới một use-case cơ sở (extended use-case – base use-case), xác định việc các hành vi trong use-case mở rộng có thể được thêm vào hành vi của use-case cơ sở như thế nào.
- Use-case mở rộng bổ xung gia tăng cho use-case cơ sở theo cách mô đun hóa.



Quan hệ trong biểu đồ Use-case

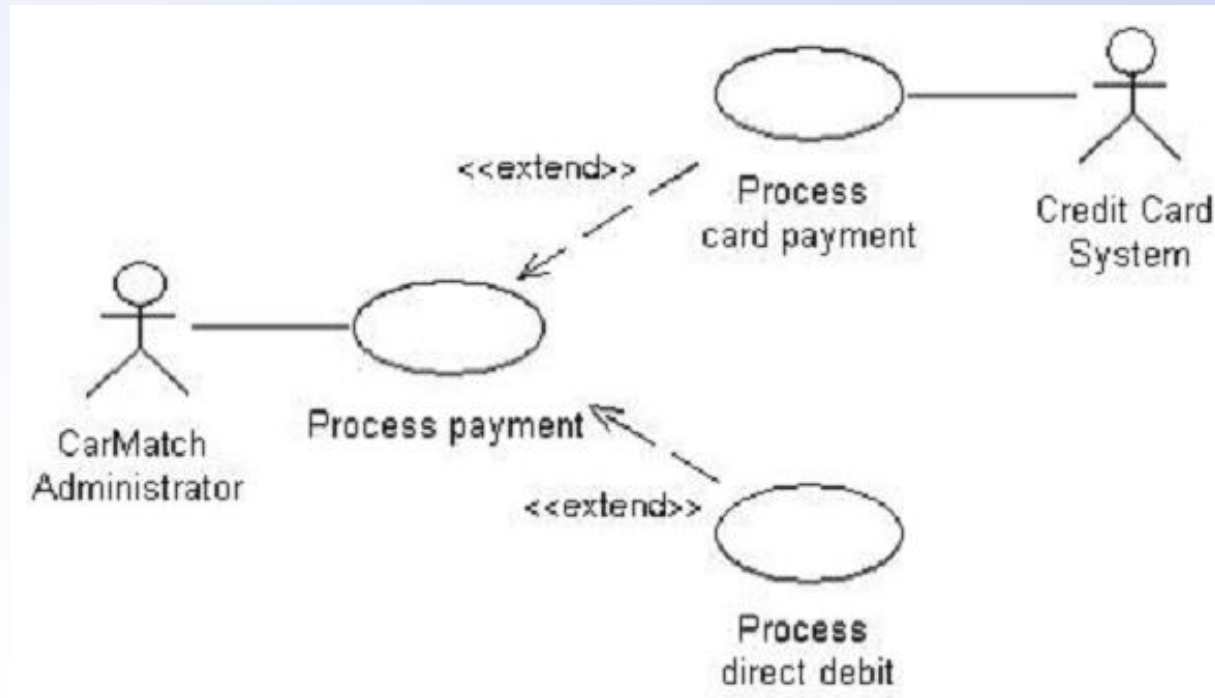
Quan hệ trừu tượng của Use-case

- Quan hệ **includes** và **extends** đều có tính chất chung là cùng sử dụng chức năng do UC khác cung cấp
- Phần chức năng sử dụng chung có thể để trong UC mới – UC trừu tượng
 - UC trừu tượng không bị tác nhân kích hoạt giao tiếp

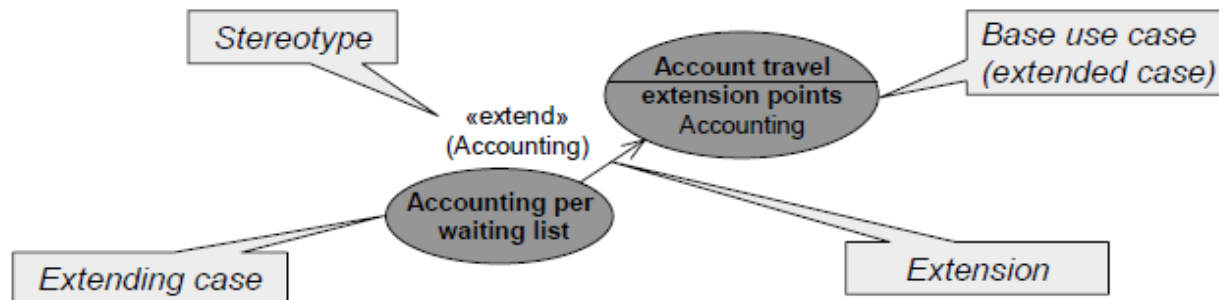


Quan hệ trong biểu đồ Use-case

Quan hệ mở rộng của Use-case



Notation:



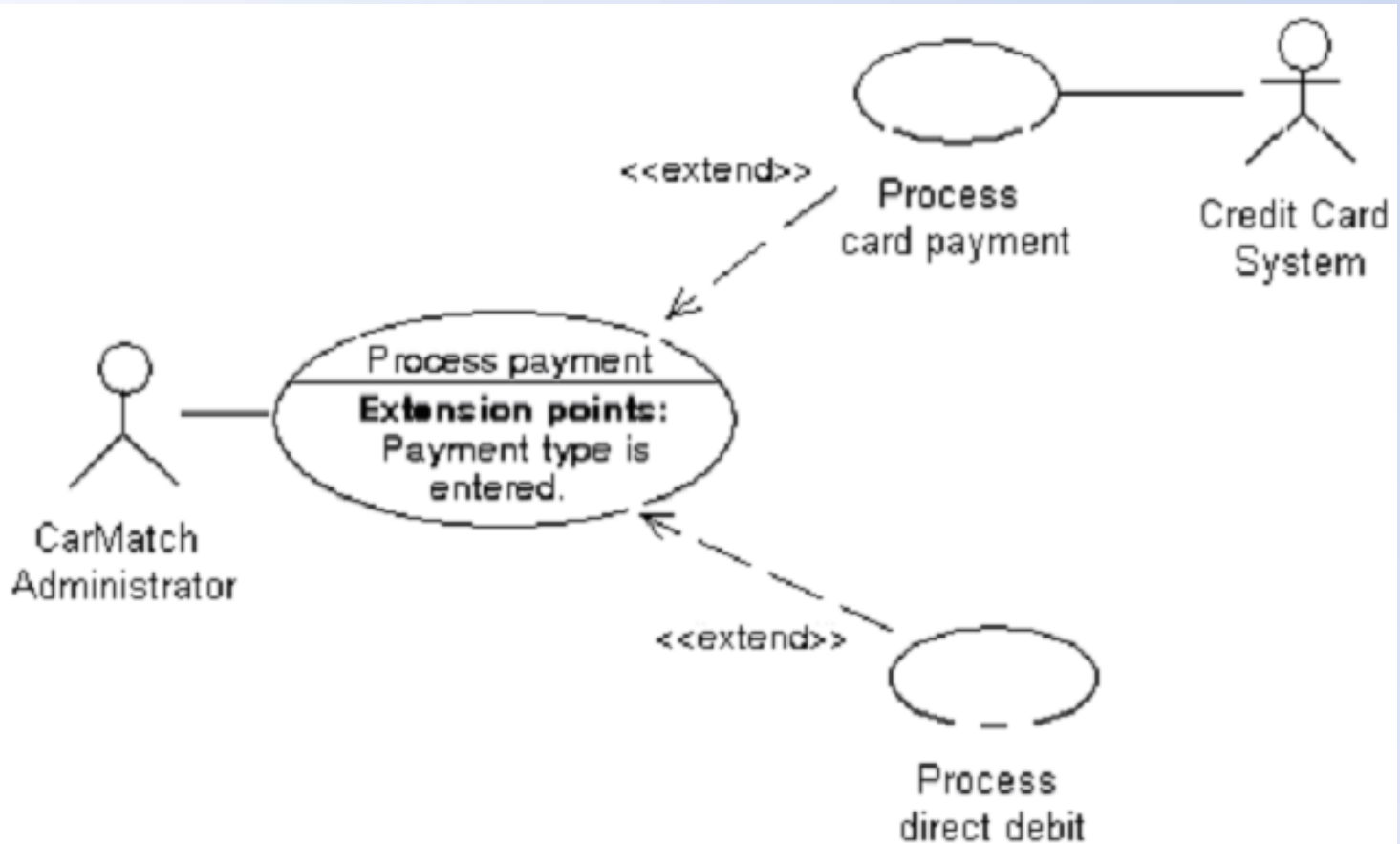
Quan hệ trong biểu đồ Use-case

Usecase với Extension Point

- ❖ **Chi tiết của các điểm trong use-case mà tại đó sự mở rộng xảy ra được mô tả ở phần bên dưới nằm trong biểu tượng use-case.**
 - Phần này có tên là Extension points (các điểm mở rộng)
- ❖ **Trong quan hệ extend, một số điều kiện phải thỏa mãn thì use case mở rộng mới được thực hiện**
 - `paymentType = Credit card` hoặc `debit card`
 - `paymentType = cash` hoặc `cheque`
 - `paymentType = direct debit`

Quan hệ trong biểu đồ Use-case

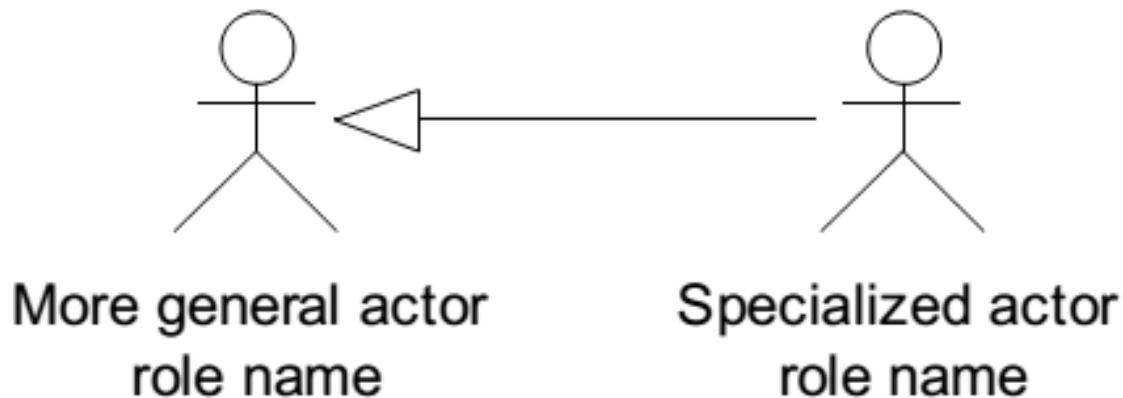
Usecase với Extension Point



Quan hệ trong biểu đồ Use-case

Quan hệ giữa các Actor – Generalization

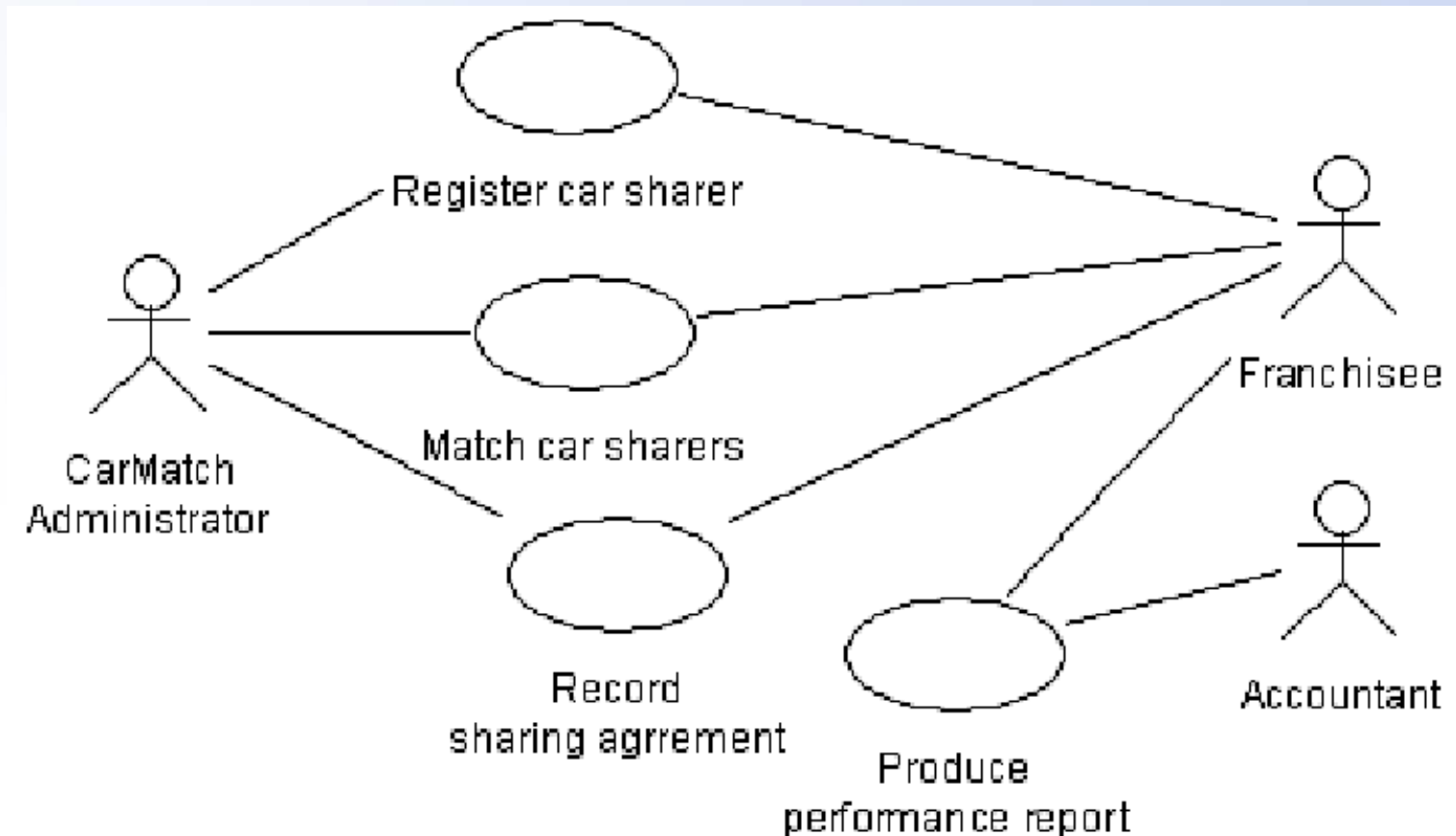
- ❖ Giữa các actors có thể tồn tại kết hợp tổng quát (generalization)



Quan hệ trong biểu đồ Use-case

Quan hệ giữa các Actor – Generalization

- ❖ Trong hệ thống CarMatch, Franchisee quản lý cả việc đăng ký thành viên mới và cũng là người nhận báo cáo.

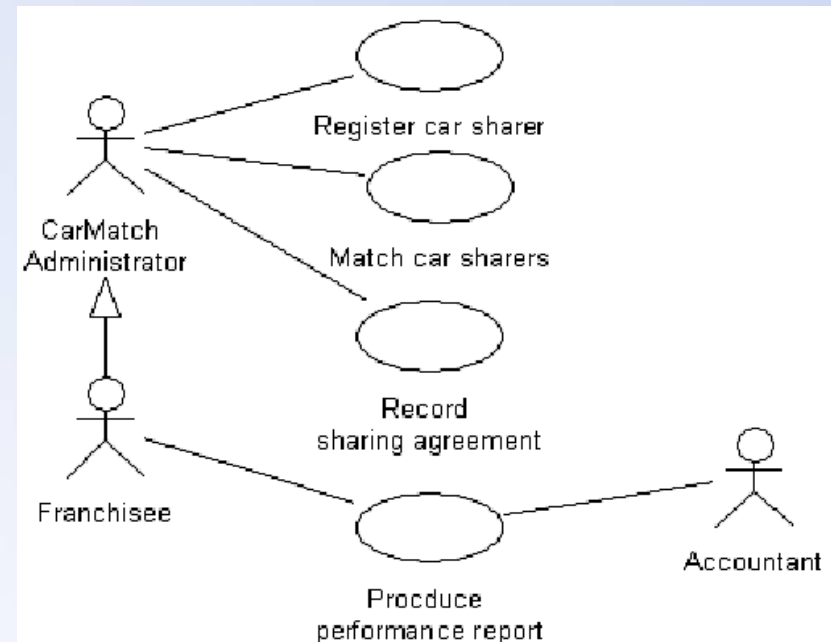


Quan hệ trong biểu đồ Use-case

Quan hệ giữa các Actor – Generalization

❖ Thay vì biểu diễn như trên, ta có thể biểu diễn Franchisee là một trường hợp đặc biệt của CarMatch Administrator.

- Nghĩa là Franchisee có thể thực hiện mọi hành động của CarMatch Administrator, ngoài ra nó còn có riêng một số hoạt động khác.



Biểu diễn Use-case

Làm tài liệu UC

❖ Mô tả UC bao gồm các thông tin sau

- Khởi đầu UC - sự kiện khởi động UC
 - "UC bắt đầu khi X xảy ra"
- Kết thúc UC - sự kiện dừng UC
 - "Khi Y xảy ra thì UC kết thúc"
- Tương tác giữa UC và tác nhân
- Trao đổi thông tin
 - "Người sử dụng làm việc với hệ thống và nhập tên, mật khẩu"
- Niên đại và nguồn gốc của thông tin
 - khi nào hệ thống đòi hỏi thông tin và khi nào hệ thống lưu trữ chúng
- Lặp hành vi trong UC
 - có thể được mô tả bằng pseudo-code, biểu đồ activity
- Tình thế phụ

Biểu diễn Use-case

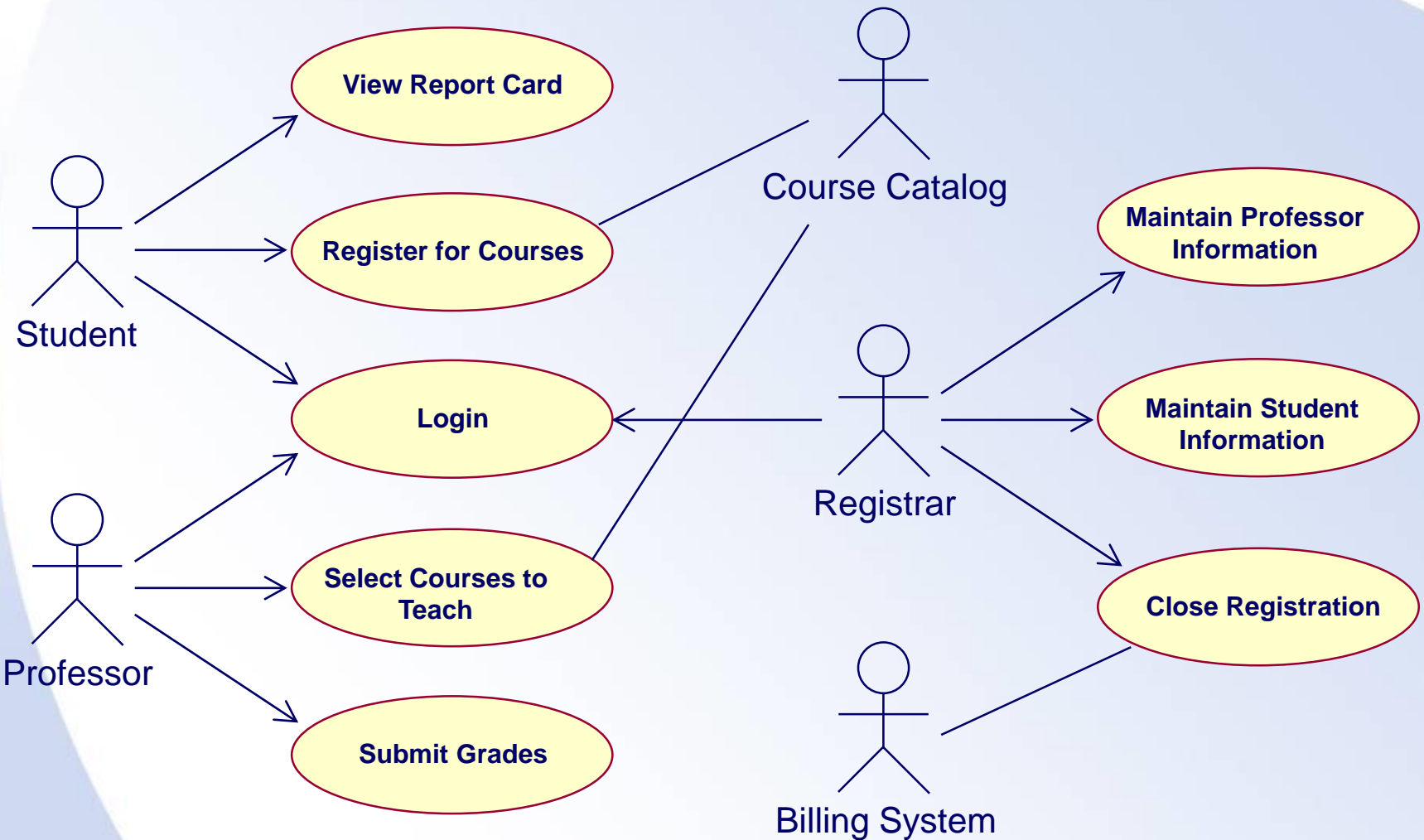
Đã tìm đầy đủ UC cho hệ thống?

❖ Các câu hỏi sau giúp xác định đã tìm đầy đủ UC?

- Mỗi yêu cầu chức năng ở trong ít nhất một UC?
 - Nếu yêu cầu chức năng không ở trong UC nào thì nó sẽ không được cài đặt sau này.
- Đã khảo sát mọi tác nhân tương tác với hệ thống?
- Tác nhân cung cấp cho hệ thống thông tin nào?
- Tác nhân nhận thông tin nào từ hệ thống?
- Đã nhận biết mọi hệ thống bên ngoài tương tác với hệ thống đang xây dựng?
- Thông tin nào hệ thống bên ngoài nhận và gửi cho hệ thống đang xây dựng?

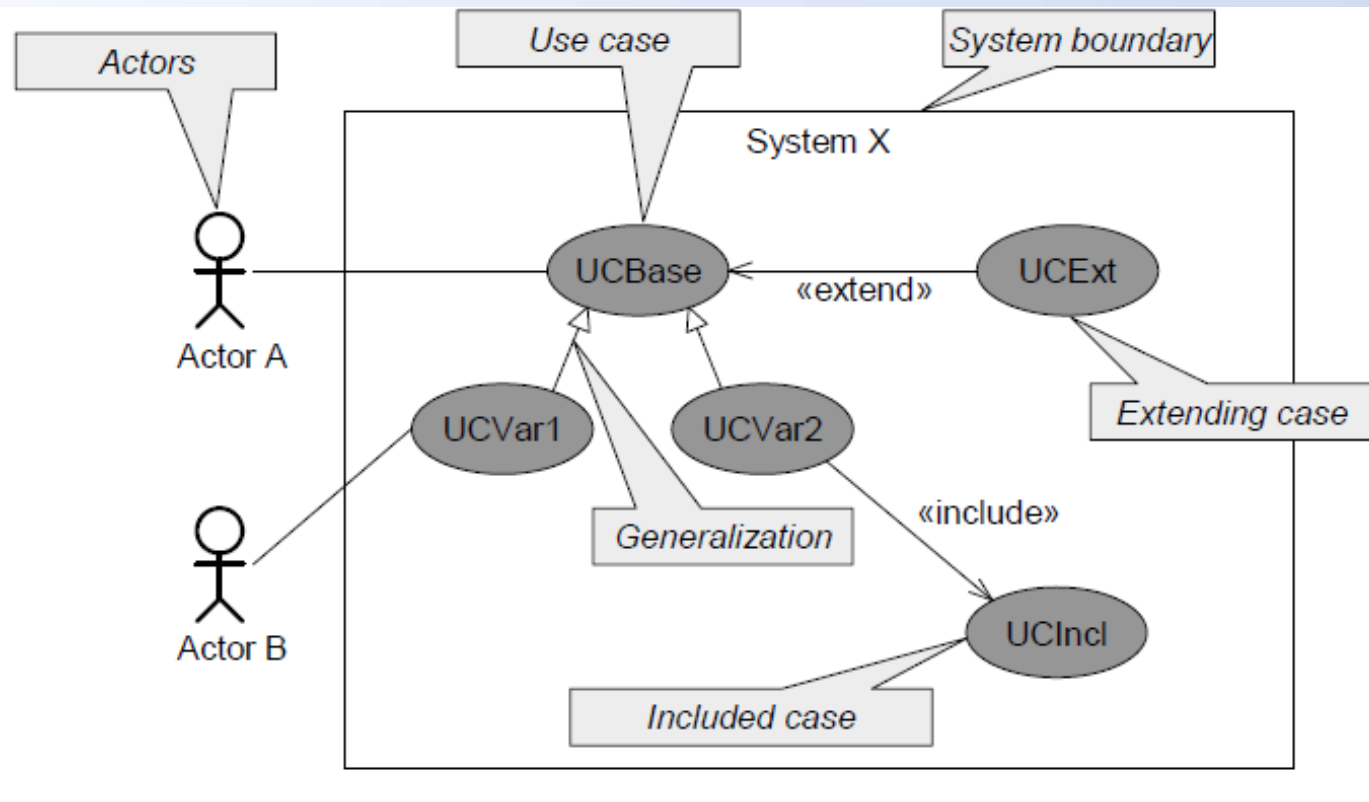
Biểu diễn Use-case

Cách đọc sơ đồ Use case



Biểu diễn Use-case

Cách đọc sơ đồ Use case



Biểu diễn Use-case

❖ Các chú ý khi xây dựng biểu đồ UC

- Không nên mô hình hóa quan hệ kết hợp giữa tác nhân với tác nhân -> vì giao tiếp giữa các tác nhân là ở bên ngoài hệ thống
 - Hãy sử dụng biểu đồ luồng công việc để khảo sát quan hệ giữa các tác nhân
- Không hình thành quan hệ Association giữa các UC
 - Biểu đồ chỉ ra có các UC nào nhưng không chỉ ra trật tự thực hiện chúng
- Mỗi UC phải có tác nhân kích hoạt (trừ UC trong quan hệ **extends** và quan hệ **includes**)
 - Nên vẽ mũi tên thể hiện association đi từ tác nhân đến UC
- Có thể xem CSDL là lớp ở dưới biểu đồ UC
 - Có thể nhập tin vào CSDL ở UC này và xâm nhập dữ liệu trong CSDL ở UC khác
 - Không vẽ association giữa các UC để chỉ ra luồng thông tin

Trao đổi, câu hỏi?