

Lecture 9:

Mô hình hóa tương tác hệ thống

Tương tác với hệ thống mới

- ↪ Con người sẽ tương tác với hệ thống như thế nào?
- ↪ Khi nào/Vì sao họ tương tác với hệ thống?

Use Cases

- ↪ Giới thiệu mô hình hoạt vụ (use cases)
- ↪ Định nghĩa tác nhân (actors)
- ↪ Định nghĩa tình huống (cases)
- ↪ Các đặc tính mở rộng

Biểu đồ tuần tự (Sequence Diagrams)

- ↪ Trình tự thời gian của các sự kiện bao gồm trong hoạt vụ (use case)

Mô tả về sự chuyển dịch

Hệ thống Use case sẽ cung cấp những chức năng gì ?

- ⇒ Con người sẽ tương tác với nó như thế nào?
- ⇒ Mô tả các chức năng từ hướng nhìn của người dùng

UML Use Cases

- ⇒ Dùng để chỉ:
 - Các chức năng (functions) được cung cấp bởi hệ thống
 - Tác nhân (actors) nào sẽ dùng những chức năng này
- ⇒ Mỗi Use Case là:
 - Một mẫu ứng xử mà hệ thống mới được yêu cầu biểu diễn
 - Một trình tự của các hành động có liên quan thực thi bởi một tác nhân và hệ thống thông qua việc đối thoại.

Một tác nhân (actor) là :

- ⇒ Mọi thứ cần tương tác với hệ thống: Một người (a person); Một vai trò (role) mà những người khác nhau có thể đóng; Một hệ thống khác (bên ngoài)

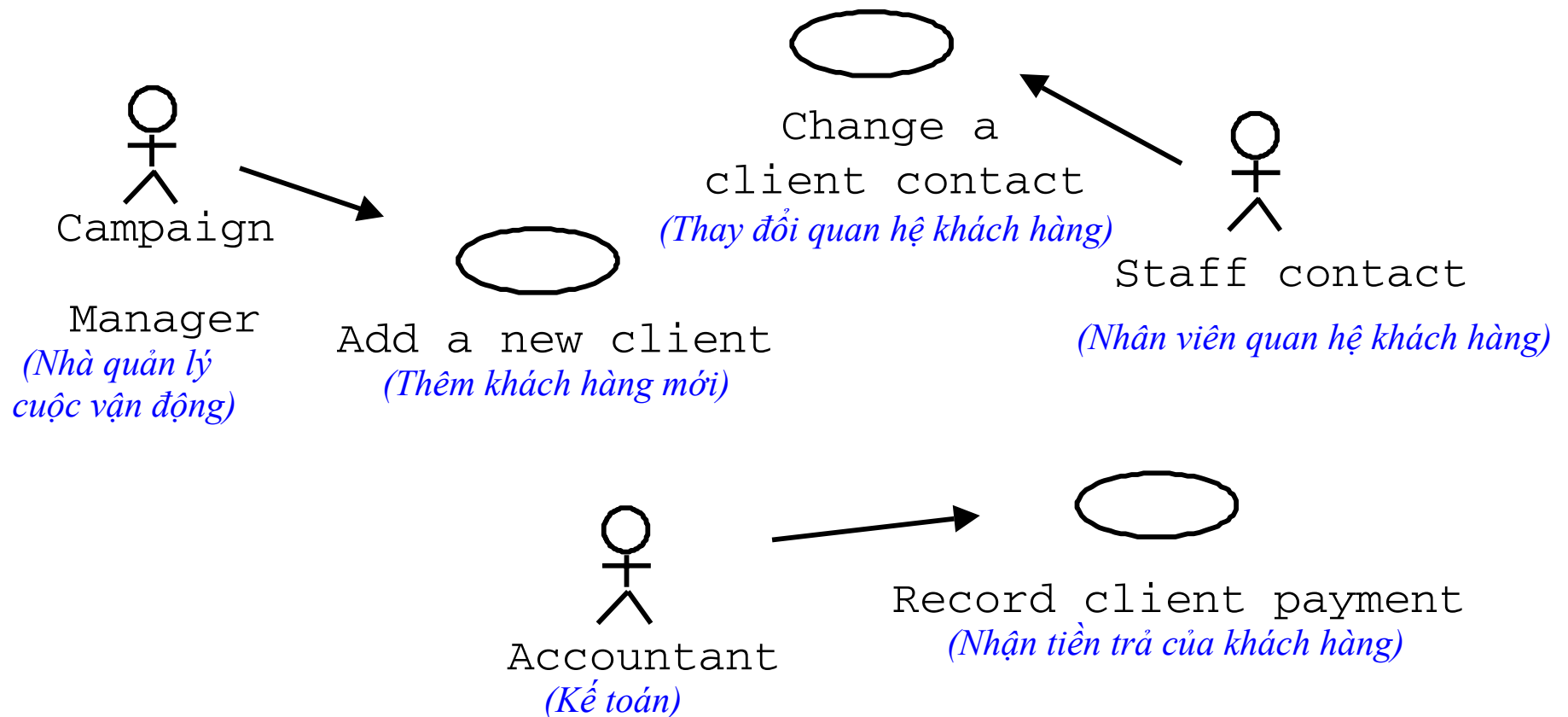
Ranh giới hệ thống (System Boundary) biểu diễn :

- ⇒ Như một cái hộp chứa tất cả các use case có liên quan (vẽ dưới dạng khung chữ nhật)
- ⇒ Lưu ý : các tác nhân nằm bên ngoài ranh giới hệ thống.

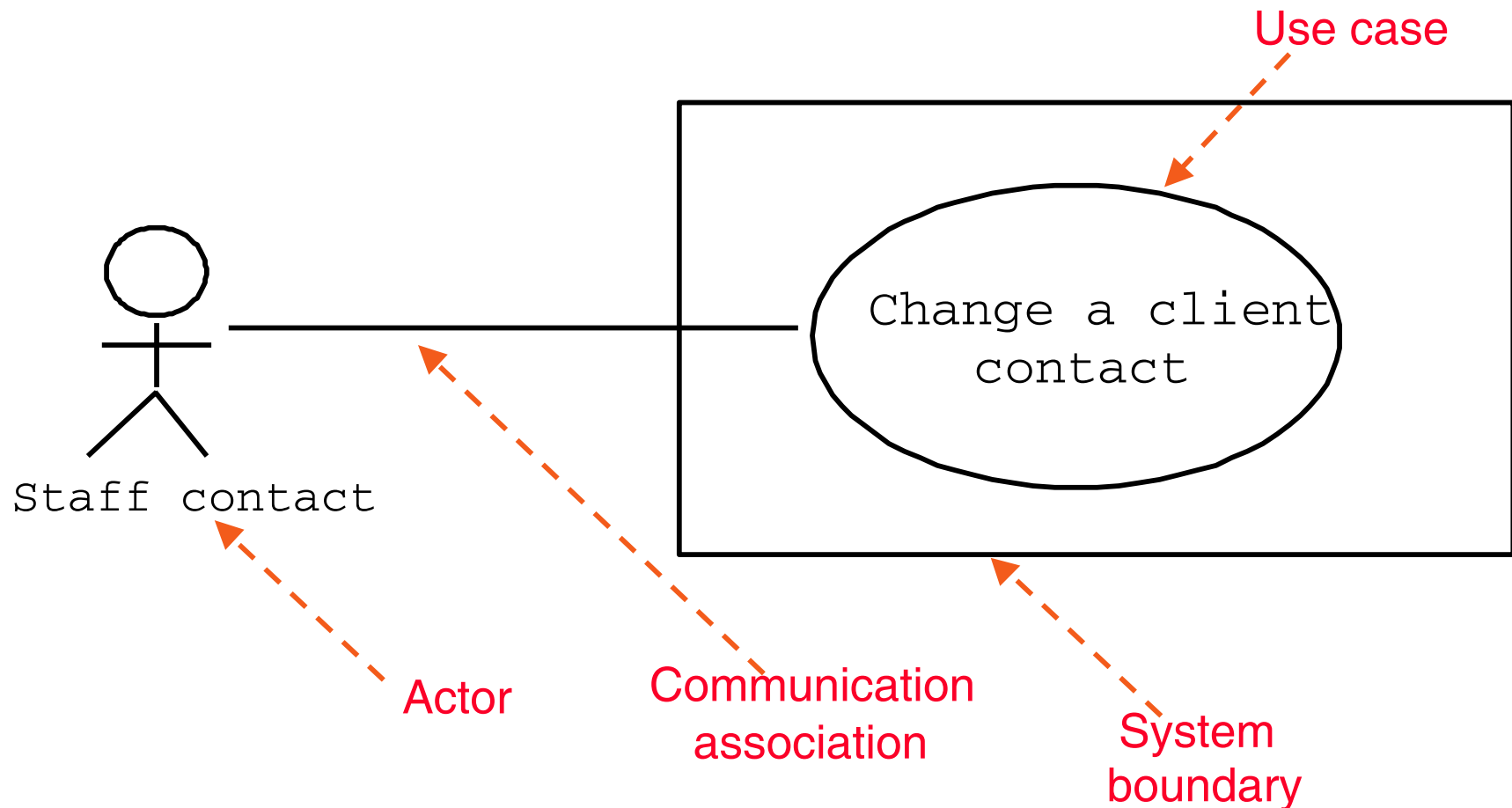
Đường nối kết (Communication association) : nối giữa tác nhân và các usecase.

Sơ đồ hoạt vụ (Use Case Diagrams)

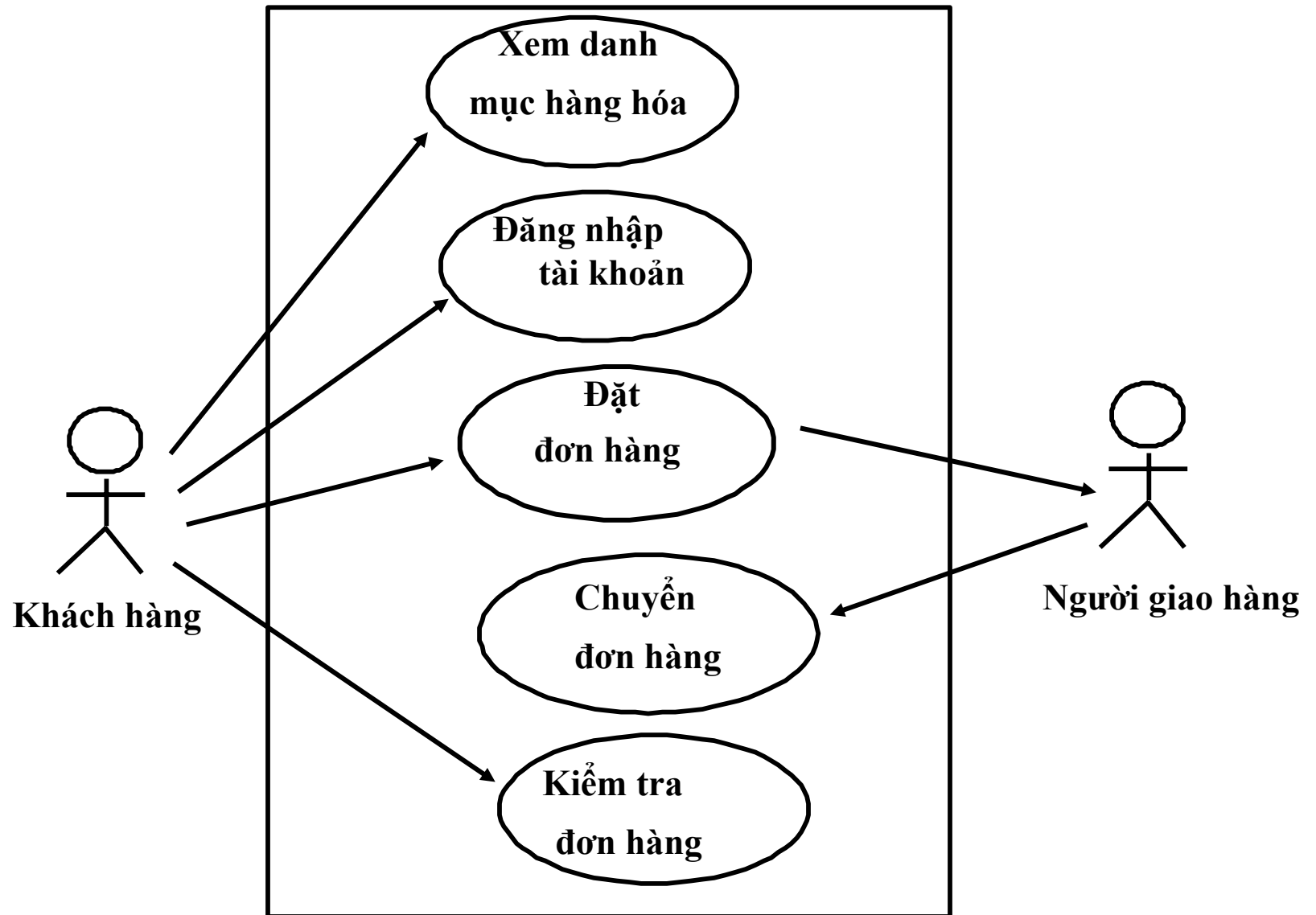
Nắm bắt mối quan hệ giữa các nhân tố và Use Cases



Các ký hiệu trong Sơ đồ hoạt vụ



Ví dụ



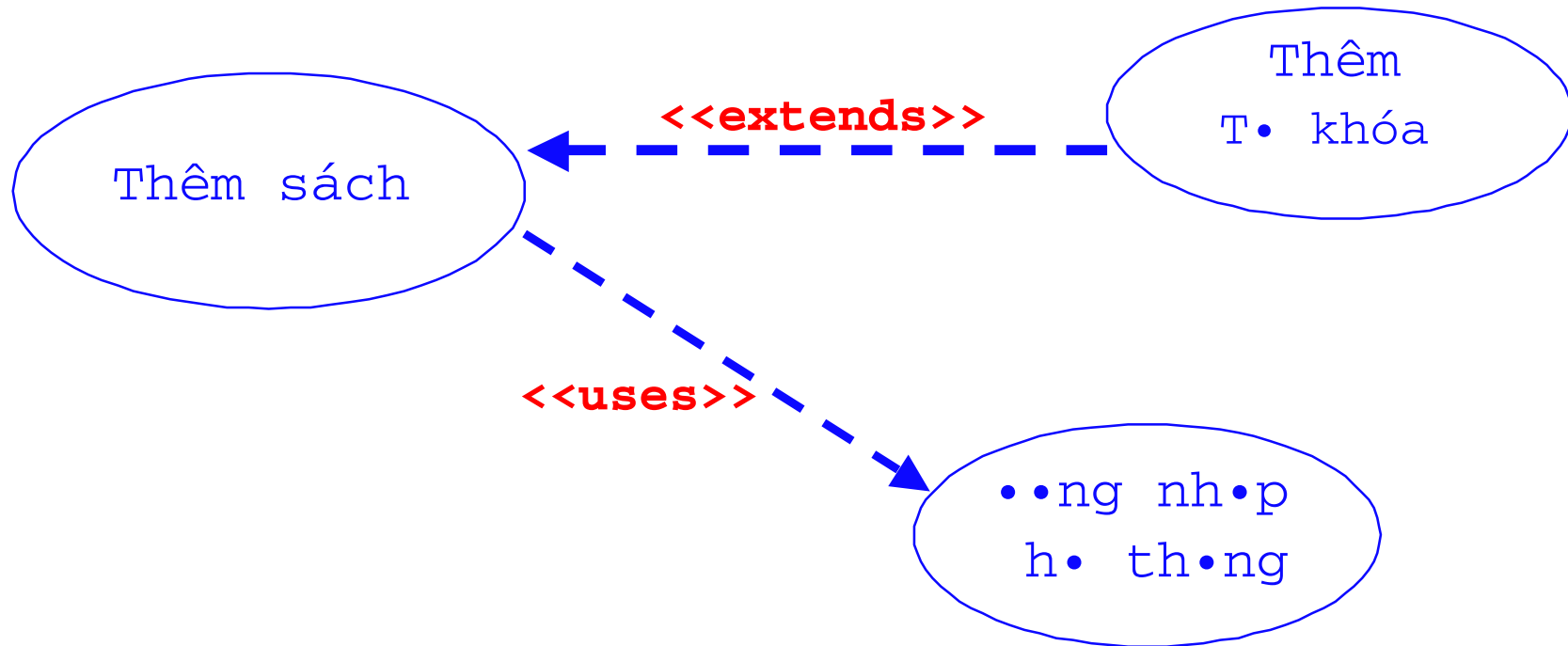
Quan hệ <<extends>> và <<uses>>

<<extends>> : khi một uses case thêm hành vi ứng xử vào base case

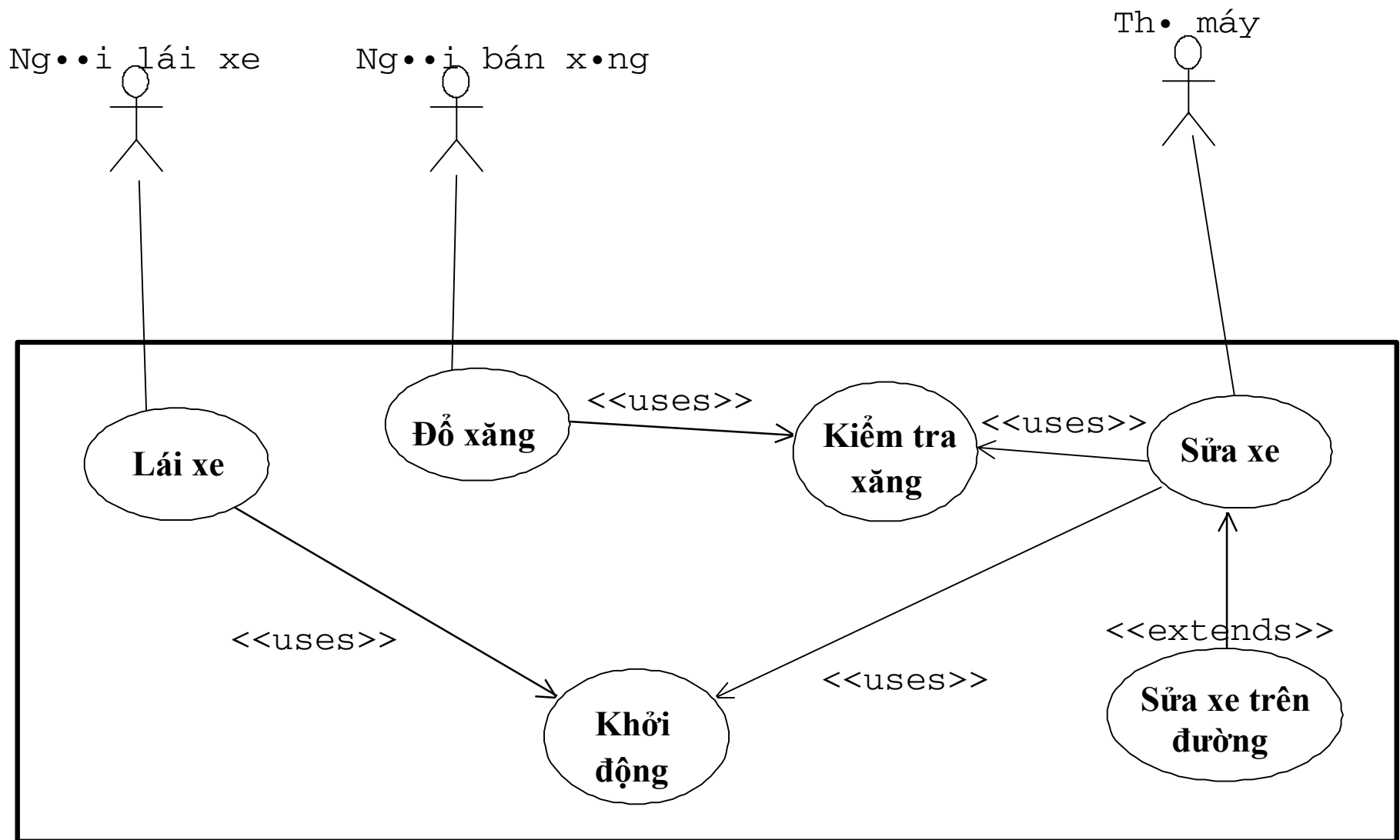
- ⇒ Dùng để mô hình một phần của use case mà người dùng có thể nhìn thấy như **hành vi tùy chọn** của hệ thống ;
- ⇒ Cũng mô hình một sub-case riêng lẻ mà nó chỉ thực thi trong một số điều kiện.

<<uses>>: khi một use case gọi đến một cái khác (giống như gọi thủ tục)

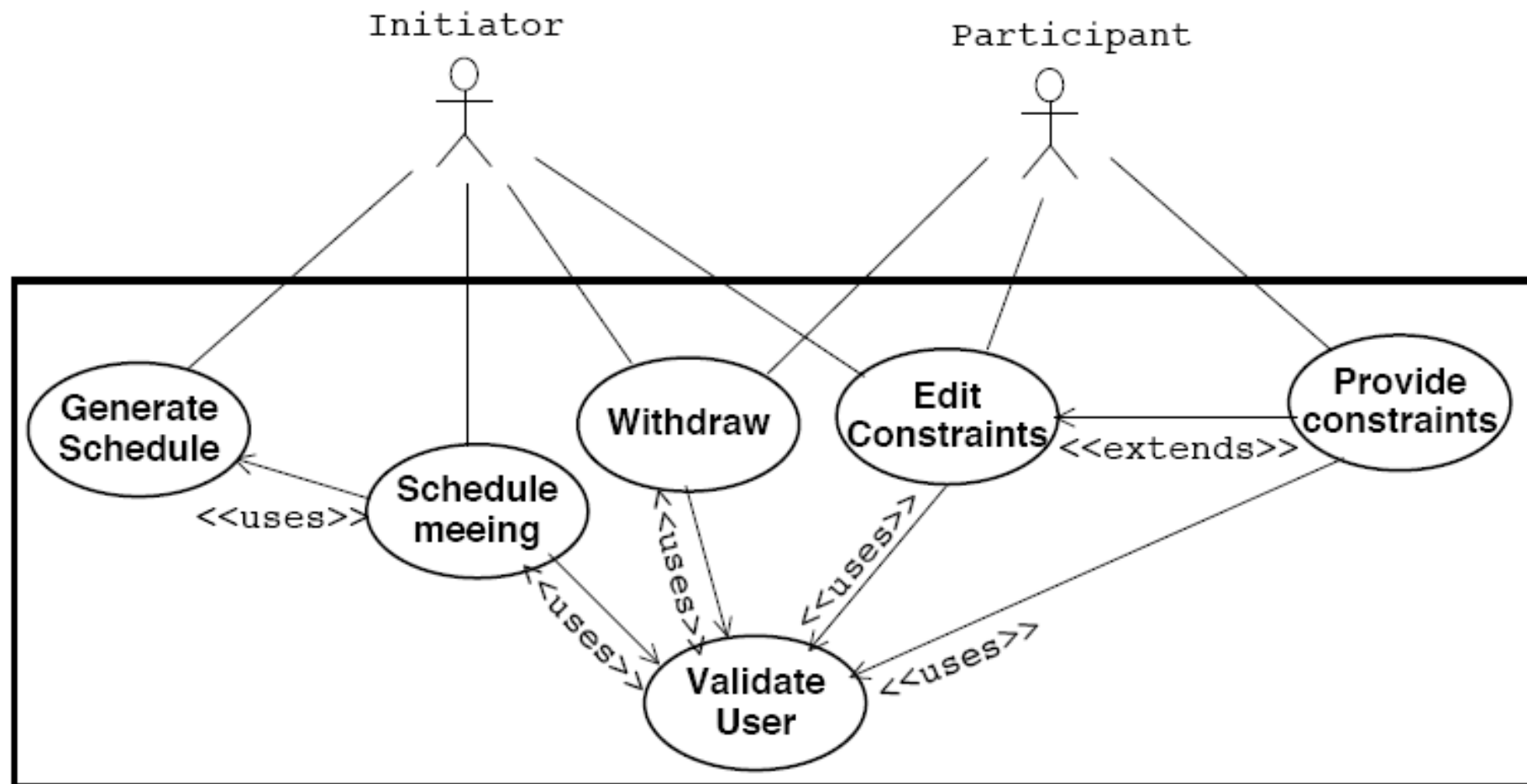
- ⇒ Dùng để tránh việc mô tả cùng một dòng sự kiện một vài lần
- ⇒ Đặt hành vi chung trong một use case của cái sở hữu nó.



Mẫu use cases cho một chiếc xe



Ví dụ sắp xếp lịch họp



Quan hệ kế thừa

Lớp tác nhân (Actor classes)

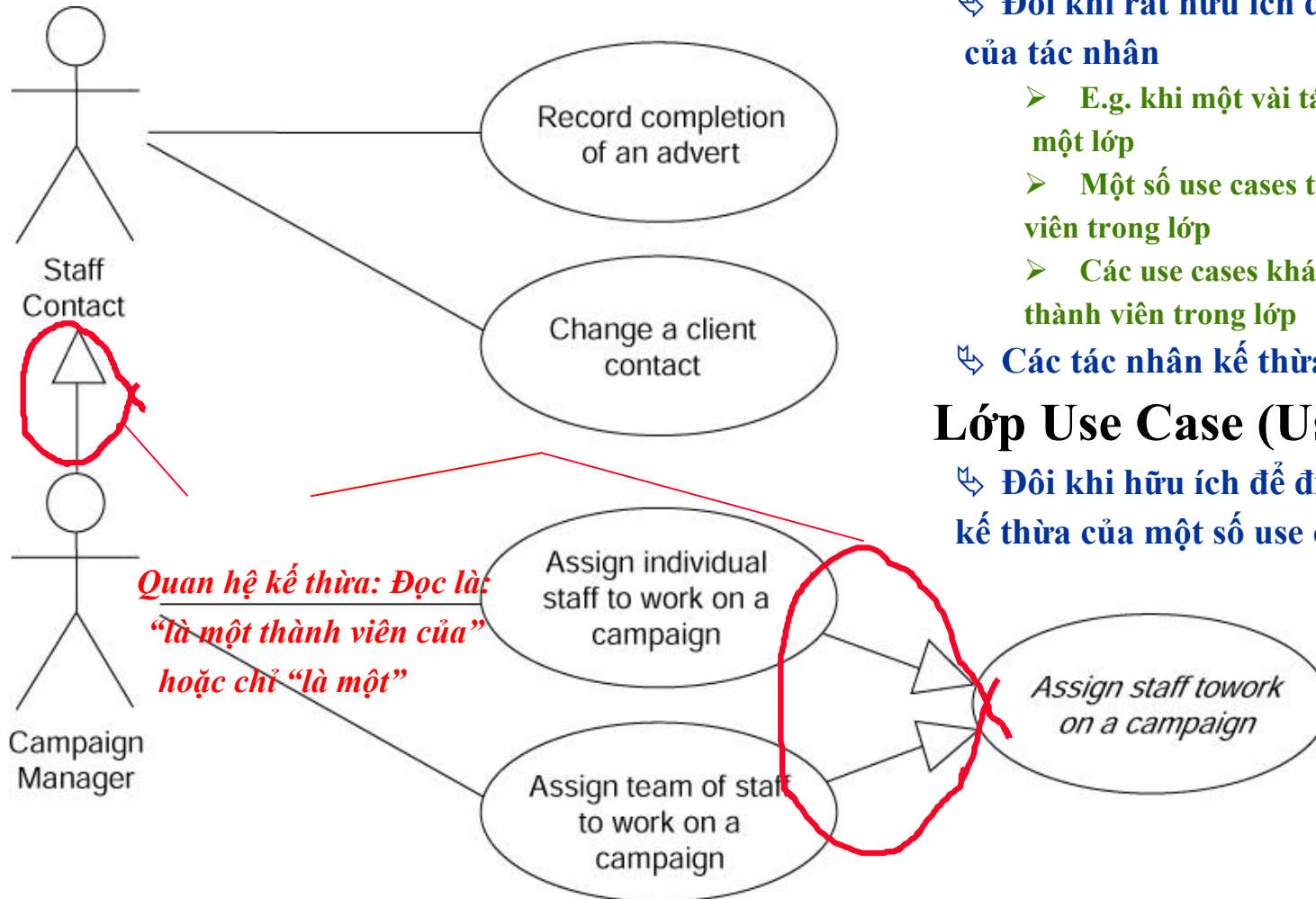
⇒ Đôi khi rất hữu ích để định nghĩa các lớp của tác nhân

- E.g. khi một vài tác nhân cùng thuộc về cùng một lớp
- Một số use cases thì cần bởi tất cả các thành viên trong lớp
- Các use cases khác thì chỉ cần bởi một số thành viên trong lớp

⇒ Các tác nhân kế thừa use cases từ lớp

Lớp Use Case (Use Case classes)

⇒ Đôi khi hữu ích để định nghĩa một quan hệ kế thừa của một số use case



Nhận biết các tác nhân (Actors))

Hỏi những câu hỏi sau:

- ⇒ Ai sẽ là người dùng chính của hệ thống? (tác nhân chính)
 - Ai sẽ cần sự hỗ trợ từ hệ thống để làm các công việc hàng ngày của họ?
 - Ai hoặc điều gì sẽ quan tâm đến những kết quả mà hệ thống đưa ra ?
- ⇒ Ai sẽ bảo trì, quản lý, điều hành hoạt động của hệ thống ? (tác nhân phụ)
- ⇒ Hệ thống cần các thiết bị phần cứng nào ?
- ⇒ Hệ thống cần tương tác với những hệ thống nào khác ?

Tìm kiếm:

- ⇒ Những người trực tiếp sử dụng hệ thống
- ⇒ Và những người dùng khác – những người cần các dịch vụ từ hệ thống

Có 3 loại Actor chính:

- ⇒ Người dùng. ➢ Ví dụ: sinh viên, nhân viên, khách hàng...
- ⇒ Hệ thống khác.
- ⇒ Sự kiện thời gian. ➢ Ví dụ: Kết thúc tháng, đến hạn...

Tìm kiếm Use Cases

Đối với mỗi tác nhân, hỏi các câu hỏi sau:

- ⇒ Những chức năng nào mà tác nhân đòi hỏi từ hệ thống ?
- ⇒ Tác nhân nào cần làm ?
- ⇒ Tác nhân có cần đọc (read), tạo ra (create), phá hủy (destroy), sửa đổi (modify) hoặc lưu trữ (store) một số dạng thông tin trong hệ thống ?
- ⇒ Tác nhân có phải thông báo về các sự kiện trong hệ thống ?
- ⇒ Tác nhân thông báo những gì với hệ thống ?
- ⇒ Các sự kiện gì thì cần thiết trong môi trường chức năng của hệ thống?
- ⇒ Hoạt động thông thường của tác nhân thì quá đơn giản hoặc quá hiệu quả đối với các chức năng mới được cung cấp bởi hệ thống ?

Lập tài liệu Use Cases

Cho mỗi use case:

- ↪ Chuẩn bị tài liệu “luồng sự kiện” (“flow of events”), được viết từ hướng nhìn của một tác nhân.
- ↪ Mô tả chi tiết những cái mà hệ thống cần phải làm chứ không chỉ ra hệ thống sẽ làm nó như thế nào?

Các nội dung đặc trưng :

- ↪ Use case bắt đầu và kết thúc như thế nào;
- ↪ Tiến trình bình thường của các sự kiện;
- ↪ Tiến trình thay phiên của các sự kiện;
- ↪ Tiến trình ngoại lệ của các sự kiện;

Kiểu tài liệu:

- ↪ Chọn lựa cách hiển thị use case:
 - Biểu đồ hoạt động (Activity Diagrams) – tốt cho tiến trình công việc
 - Biểu đồ hợp tác (Collaboration Diagrams) – tốt cho thiết kế cấp cao
 - Biểu đồ trình tự (Sequence Diagrams) – tốt cho thiết kế chi tiết

Mẫu tài liệu Use Cases

Có thể dùng theo mẫu đơn giản như sau:

- **X. Luồng sự kiện cho Use case ABC**
- **X1. Điều kiện bắt đầu:** danh sách những điều kiện phải thỏa mãn trước khi Use case được thực hiện.

Ví dụ như: một Use case khác phải thực hiện trước khi Use case này được thực hiện hay người dùng phải có đủ quyền để thực hiện Use case này. Không nhất thiết mọi Use case đều phải có điều kiện bắt đầu.
- **X2. Luồng chính:** mô tả những bước chính sẽ xảy ra khi thực hiện Use case.
- **X3. Các luồng phụ (luồng con).**
- **X4. Các luồng rẽ nhánh.**

Trong đó X là số thự tự của Use case trong hệ thống

Ví dụ : Luồng sự kiện mô tả Use Case

Luồng sự kiện mô tả Use case cho hệ thống rút tiền tự động như sau:

1.1 Điều kiện bắt đầu.

1.2 Luồng chính:

1.2.1 Người dùng đưa thẻ vào máy.

1.2.2. Máy hiển thông báo chào mừng và yêu cầu nhập mã số

1.2.3 Người dùng nhập mã số

1.2.4 Máy xác nhận mã số đúng. **Nếu nhập sai mã số, luồng rẽ nhánh E-1 được thực hiện.**

1.2.5 Máy hiện ra ba lựa chọn:

Rút tiền: luồng con A-1

Chuyển tiền: luồng con A-2

Thêm tiền vào tài khoản: luồng con A-3

1.2.6 Người dùng chọn rút tiền

1.3. Luồng con:

1.3.1 Luồng con A-1:

1.3.1.1 Máy hỏi số lượng tiền cần rút

1.3.1.2 Người dùng nhập số tiền cần rút : **Máy kiểm tra trong tài khoản có đủ tiền không. Nếu không đủ luồng rẽ nhánh E-2 được thực hiện**

....

1.4. Luồng rẽ nhánh:

1.4.1 E-1: Người dùng nhập sai mã số :**Máy thông báo là người dùng đã nhập sai mã số yêu cầu người dùng nhập lại hoặc hủy bỏ giao dịch.**

1.4.2 E-2: Không đủ tiền trong tài khoản...

Mô hình hóa trình tự của các sự kiện

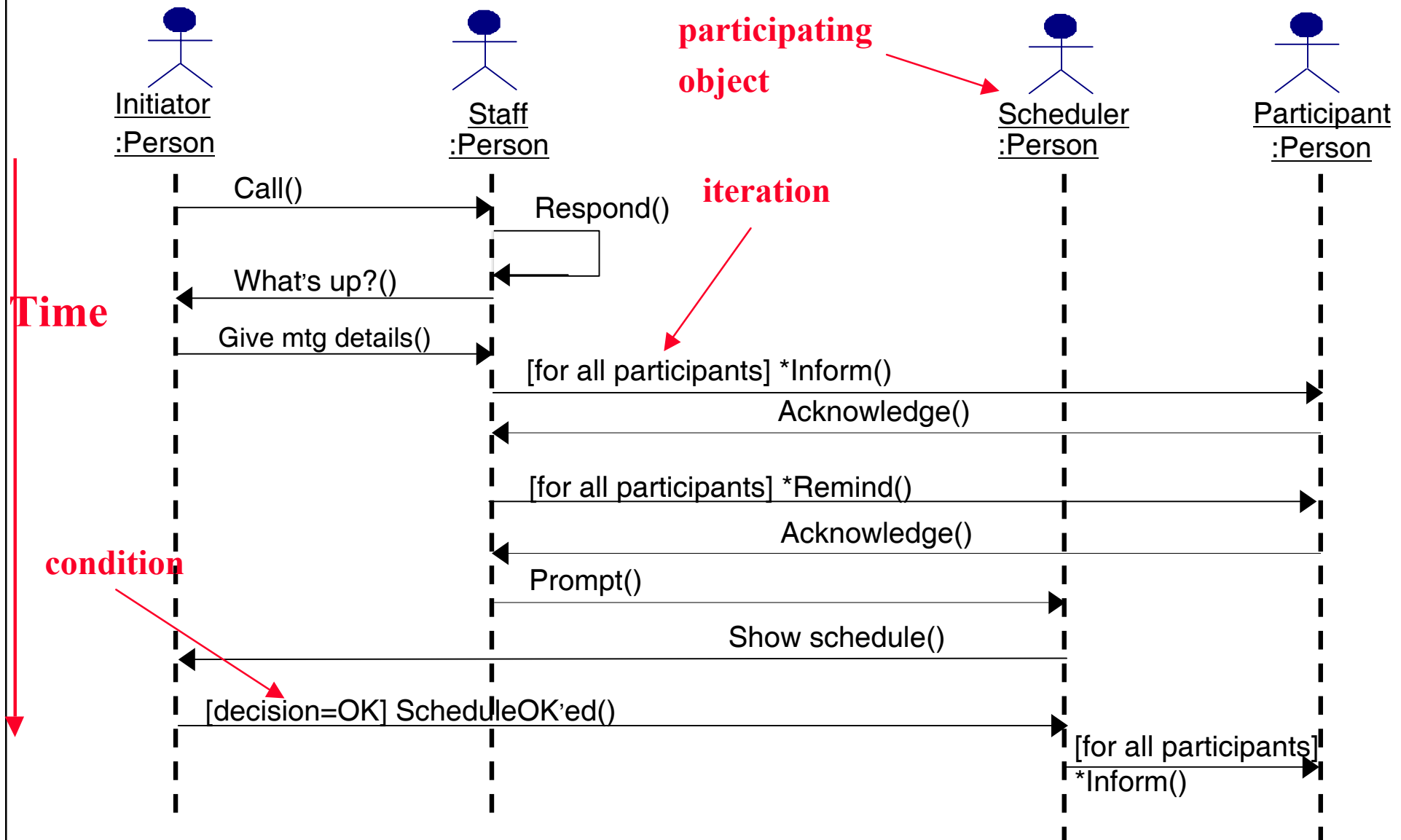
Các đối tượng “làm chủ” thông tin và hành vi

- ⇒ Chúng có các thuộc tính và phương thức liên quan đến **trách nhiệm** của chúng.
- ⇒ Chúng không “biết” thông tin về các đối tượng khác, nhưng có thể gọi đến.
- ⇒ Để thực hiện tiến trình công việc, các đối tượng phải hợp tác.
 - ...bằng cách gửi thông báo đến một cái khác để gọi những phương thức từ chúng
- ⇒ Các đối tượng chỉ có thể gửi thông báo đến cái khác nếu chúng “biết” cái khác
 - I.e. nếu có một quan hệ kết hợp giữa chúng.

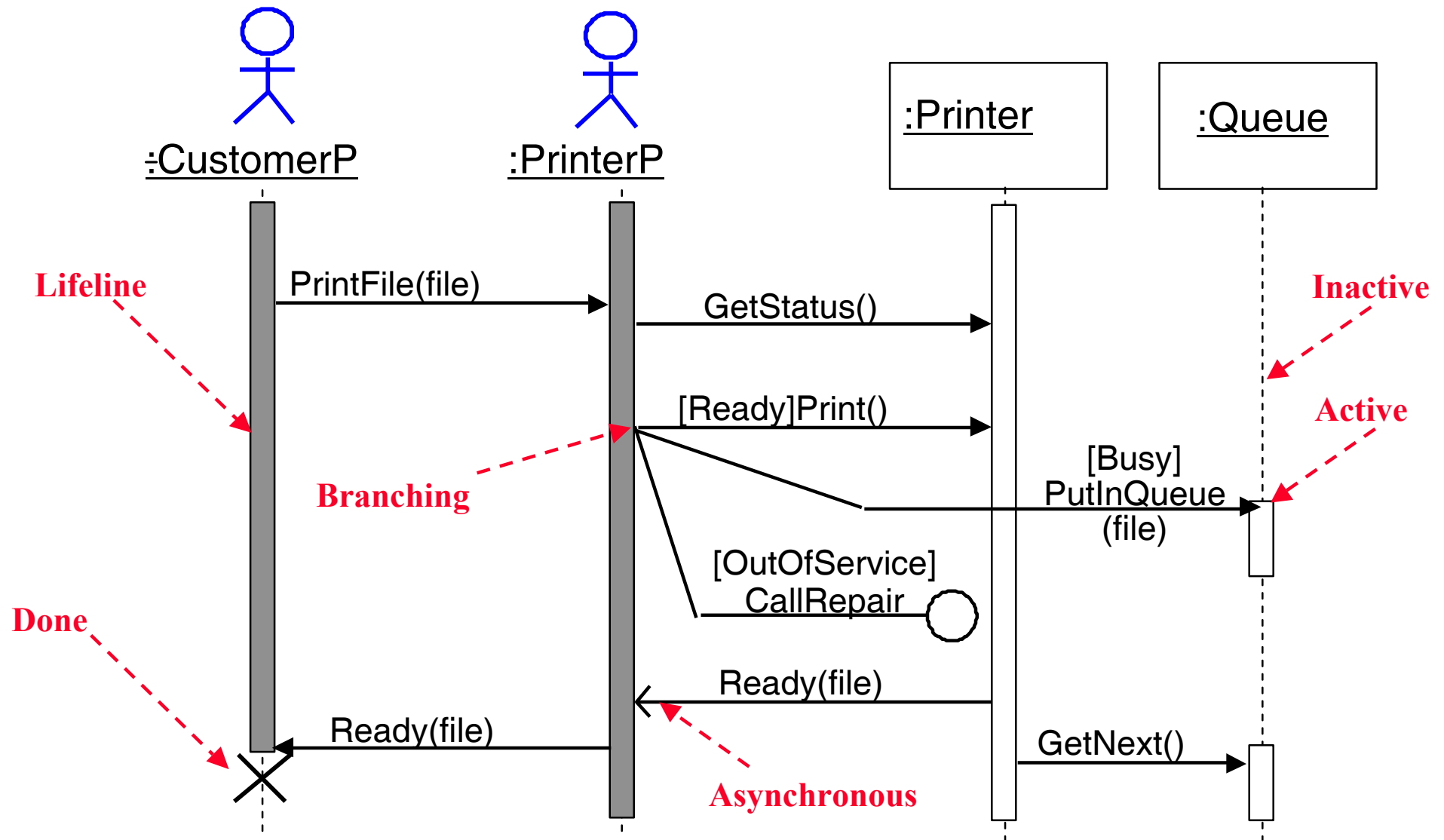
Mô tả một Use Case dùng Biểu đồ trình tự

- ⇒ Biểu đồ trình tự chỉ ra từng bước những cái bao gồm trong một use case
 - Những đối tượng nào liên quan tới use case
 - Các đối tượng đó tham gia như thế nào trong chức năng
- ⇒ Có thể cần một vài biểu đồ trình tự để mô tả duy nhất một use case.
 - Mỗi biểu đồ trình tự mô tả một kịch bản có thể cho use case
- ⇒ Biểu đồ trình tự ...
 - ... sẽ giữ dễ dàng để đọc và hiểu.
 - ... không bao gồm những kiểm soát logic phức tạp

Ví dụ về Biểu đồ trình tự



Branching messages, etc



Đừng quên cái gì đang được lập mô hình !

Trong giai đoạn phân tích

- ↳ Chúng ta muốn biết về lĩnh vực ứng dụng và các yêu cầu
- ↳ ... Vì thế, chúng ta xây dựng mô hình phác họa quá trình diễn tiến (course-grained model) để mô tả các nhiệm vụ sẽ nằm ở đâu, và các đối tượng sẽ tương tác như thế nào
 - Mô hình này sẽ chỉ rõ một thông báo được chuyển đi, nhưng không quan tâm quá nhiều về nội dung mỗi thông báo.
 - Để giữ mọi thứ rõ ràng, dùng các biểu tượng (icons) mô tả các đối tượng và tác nhân bên ngoài, và các khối hộp (boxes) để biểu diễn các đối tượng của hệ thống.

Trong giai đoạn thiết kế

- ↳ Chúng ta muốn quyết định về việc phần mềm sẽ hoạt động như thế nào
- ↳ ... Vì thế, chúng ta phát triển các mô hình phác họa cầu kỳ (fine-grained models) để chỉ ra chính xác cái gì sẽ xảy ra khi hệ thống thực thi
 - E.g. chỉ ra những chi tiết chính xác của mỗi cách truyền thông báo.