



ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

Chương 2

XÁC ĐỊNH YÊU CẦU NGƯỜI DÙNG

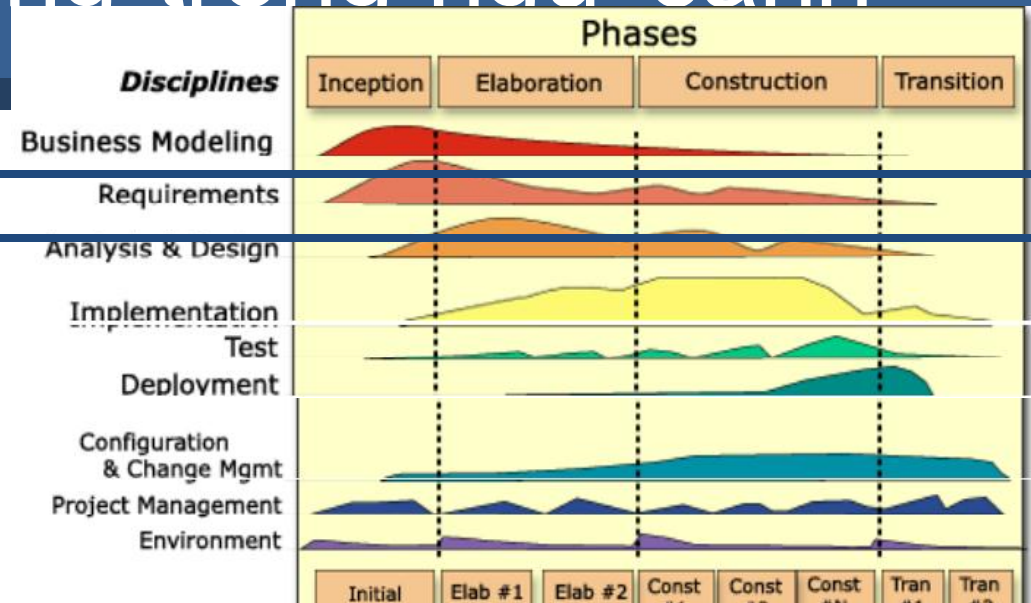
Mục tiêu

- Tìm hiểu các khái niệm cơ bản về xác định yêu cầu người dùng và tác dụng của chúng lên Phân tích và Thiết kế
- Tìm hiểu cách ghi nhận và diễn dịch các yêu cầu của người dùng, là những thông tin được dùng để bắt đầu việc phân tích và thiết kế

Nội dung

- **Giới thiệu**
- Phát biểu bài toán
- Bảng chú giải
- Các khái niệm chính
 - Actor
 - Use Case
 - Mô hình use case
- Mối quan hệ
- Đặc tả use case
- Các đặc tả bổ sung
- Checkpoints

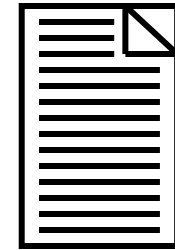
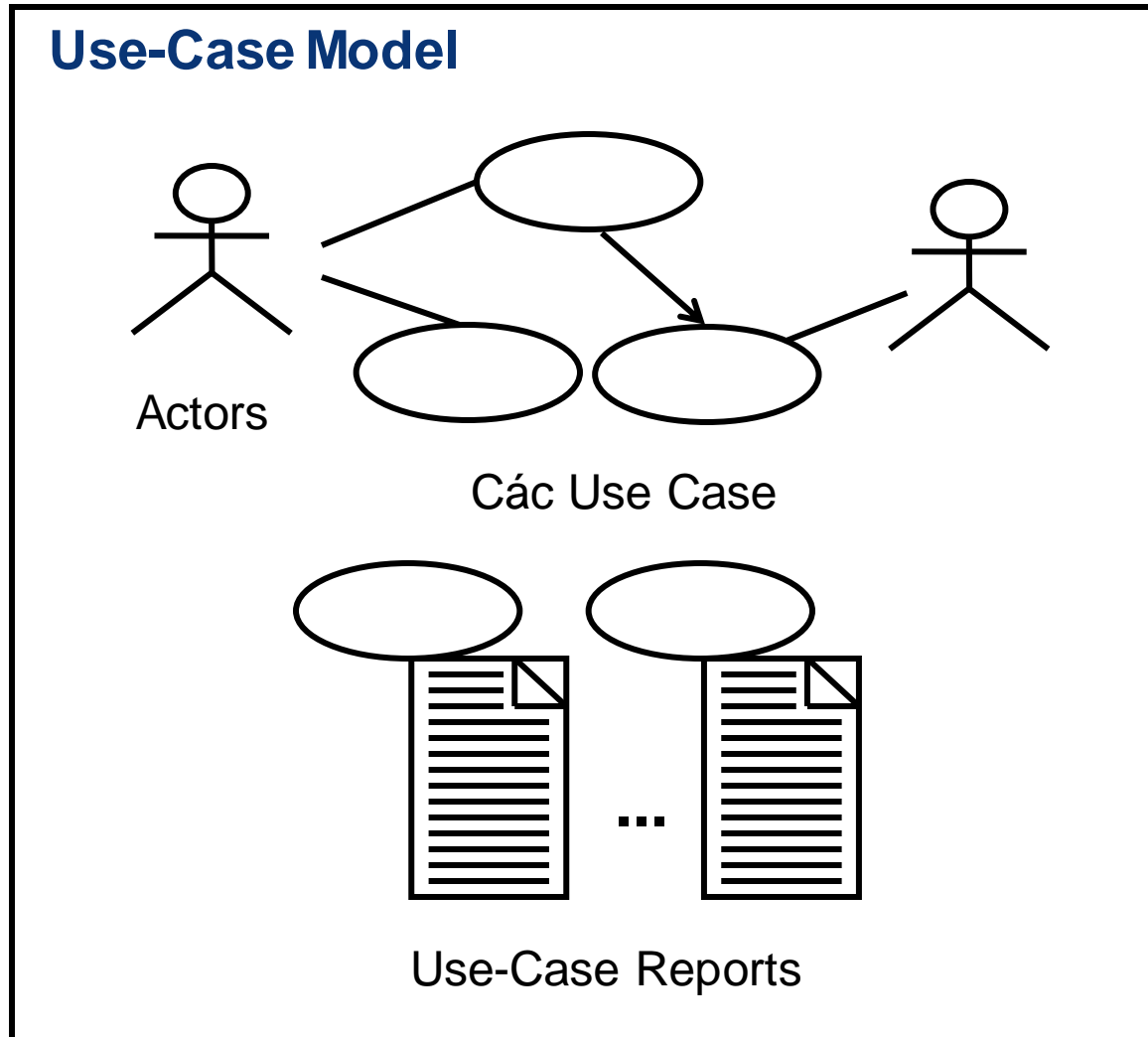
Yêu cầu người dùng trong môi trường



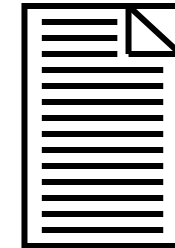
Mục đích của bước xác định yêu cầu người dùng là:

- Đi đến thỏa thuận với khách hàng và người dùng về các chức năng của hệ thống (những gì hệ thống phải thực hiện).
- Cho phép các nhà phát triển hệ thống (system developer) hiểu rõ hơn các yêu cầu đối với hệ thống.
- Phân định các ranh giới của hệ thống.
- Cung cấp cơ sở để hoạch định nội dung kỹ thuật của các vòng lặp.
- Xác định giao diện người dùng cho hệ thống.

Các dạng thông tin về yêu cầu người dùng



Bảng chú giải



Các đặc tả bổ sung

Nội dung

- Giới thiệu
- **Phát biểu bài toán**
- Bảng chú giải
- Các khái niệm chính
 - Actor
 - Use Case
 - Use Case Model
- Mối quan hệ
- Đặc tả use case
- Các đặc tả bổ sung
- Checkpoints

Ví dụ: Phát biểu bài toán Đăng ký học phần

- Hệ thống mới cho phép sinh viên đăng ký học phần và xem phiếu điểm từ một máy tính cá nhân được kết nối vào mạng nội bộ của trường. Các giáo sư cũng có thể truy cập hệ thống này để đăng ký lớp dạy và nhập điểm cho các môn học.
- Trường sẽ giữ lại cơ sở dữ liệu sẵn có về danh mục học phần mà lưu trữ toàn bộ thông tin về học phần. Hệ thống mới sẽ đọc các thông tin học phần trong CSDL cũ nhưng sẽ không cập nhật chúng. Phòng Đào tạo sẽ tiếp tục duy trì các thông tin học phần thông qua một hệ thống khác.
- Đầu mỗi học kỳ, sinh viên có thể yêu cầu danh sách các học phần được mở trong học kỳ đó. Thông tin về mỗi học phần, như tên giáo sư, khoa, và các môn học phần tiên quyết sẽ được cung cấp để giúp sinh viên chọn lựa.

Ví dụ: Phát biểu bài toán Đăng ký học phần (tt)

- Hệ thống mới cho phép sinh viên chọn bốn học phần được mở trong học kỳ tới. Thêm vào đó mỗi sinh viên có thể đưa ra hai môn học thay thế trong trường hợp không thể đăng ký theo nguyện vọng chính. Các học phần được mở có tối đa là 100 và tối thiểu là 30 sinh viên. Các học phần có ít hơn 30 sinh viên sẽ bị hủy. Đầu mỗi học kỳ, sinh viên có một khoảng thời gian để thay đổi lịch các học phần đã đăng ký. Sinh viên chỉ có thể thêm hoặc hủy học phần đã đăng ký trong khoảng thời gian này. Khi quá trình đăng ký hoàn tất cho một sinh viên, hệ thống đăng ký sẽ gửi thông tin tới hệ thống thanh toán (billing system) để sinh viên có thể đóng học phí. Nếu một lớp bị hết chỗ trong quá trình đăng ký, sinh viên phải được thông báo về sự thay đổi trước khi xác nhận lịch các học phần đã đăng ký.
- Ở cuối học kỳ, sinh viên có thể truy cập vào hệ thống để xem phiếu điểm. Bởi vì điểm của sinh viên là thông tin nhạy cảm cần được giữ kín, nên hệ thống phải có cơ chế bảo mật để ngăn chặn những truy cập không hợp lệ.
- Các giáo sư có thể truy cập vào hệ thống để đăng ký những học phần mà họ sẽ dạy. Họ cũng có thể xem danh sách các sinh viên đã đăng ký vào lớp của họ, và cũng có thể nhập điểm sau mỗi khóa học.

Nội dung

- Giới thiệu
- Phát biểu bài toán
- **Bảng chú giải**
- Các khái niệm chính
 - Actor
 - Use Case
 - Use Case Model
- Mối quan hệ
- Đặc tả use case
- Các đặc tả bổ sung
- Checkpoints

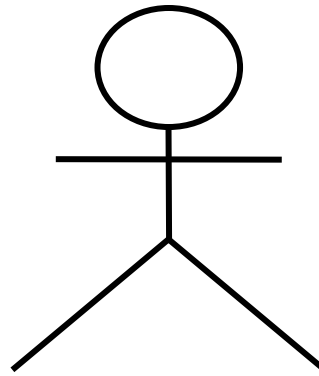
Bảng chú giải – Từ điển thuật ngữ

- **Bảng chú giải của ứng dụng “Course Registration”:**
- Course (Học phần): Một môn học được dạy trong trường.
- Course Offering (Lớp): Một lớp học cụ thể được mở trong một học kỳ cụ thể – cùng một học phần có thể được mở song song nhiều lớp trong một học kỳ. Thông tin gồm cả ngày học trong tuần và giờ học.
- Course Catalog (Danh mục học phần): Danh mục đầy đủ của tất cả các học phần được dạy trong trường.
- Faculty: Toàn bộ cán bộ giảng dạy của trường..
- Finance System (Hệ thống thanh toán): Hệ thống dùng để xử lý các thông tin thanh toán học phí.
- Grade (Điểm số): Sự đánh giá cho một sinh viên cụ thể trong một lớp cụ thể.
- Professor (Giáo sư): Người giảng dạy trong trường.
- Report Card (Phiếu điểm): Toàn bộ điểm số cho tất cả học phần một sinh viên đã học trong một học kỳ xác định.
- Roster (Danh sách sinh viên đăng ký): Tất cả sinh viên đăng ký vào một lớp học cụ thể.
- Student (Sinh viên): Người đăng ký vào học các lớp của trường.
- Schedule (Lịch học): Các học phần mà một sinh viên đã chọn học trong học kỳ hiện tại.
- Transcript (Bản sao học bạ): Bản sao tất cả điểm số cho tất cả các học phần của một sinh viên cụ thể được chuyển cho hệ thống thanh toán để hệ thống này lập hóa đơn cho sinh viên.

Nội dung

- Giới thiệu
- Phát biểu bài toán
- Bảng chú giải
- **Các khái niệm chính**
 - Actor
 - Use Case
 - Use Case Model
- Mối quan hệ
- Đặc tả use case
- Các đặc tả bổ sung
- Checkpoints

Khái niệm - Actor



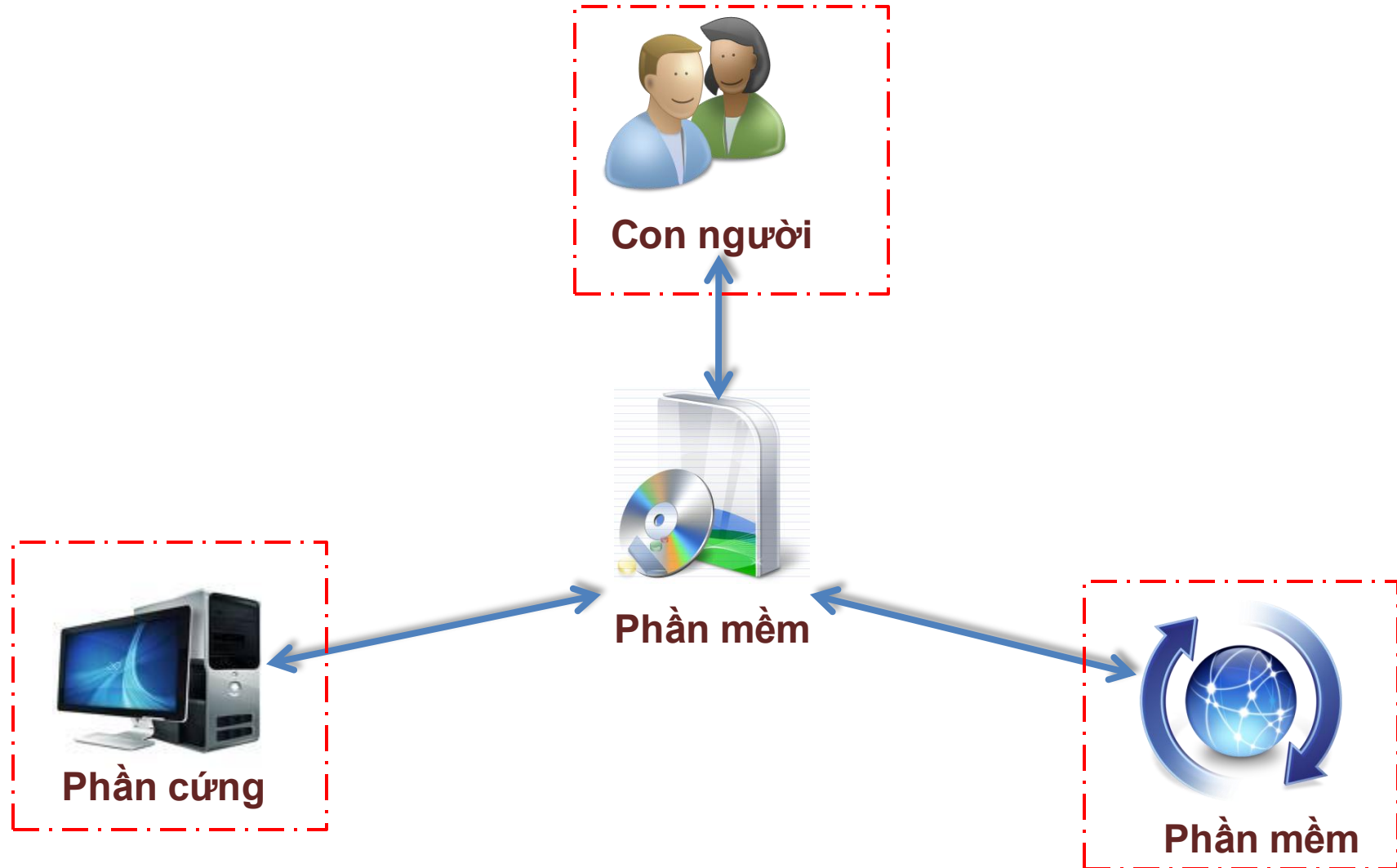
Actor (Tác nhân)

Các Actor nằm BÊN NGOÀI hệ thống

Actor

- Một actor xác định một **tập các vai trò** khi người sử dụng **tương tác** với hệ thống. Người sử dụng có thể là một **cá nhân** hay **một hệ thống khác**
- Để có thể hiểu một cách đầy đủ hệ thống cần xây dựng, bạn cần phải biết hệ thống **phục vụ cho ai**, có nghĩa là ai sẽ là người sử dụng hệ thống. Những loại người dùng khác nhau sẽ được biểu diễn bởi các tác nhân trong mô hình.
- Một tác nhân là một cái gì đó **trao đổi dữ liệu với hệ thống**. Tác nhân có thể là **người sử dụng**, một **thiết bị phần cứng bên ngoài**, hoặc có thể là một **hệ thống khác**.

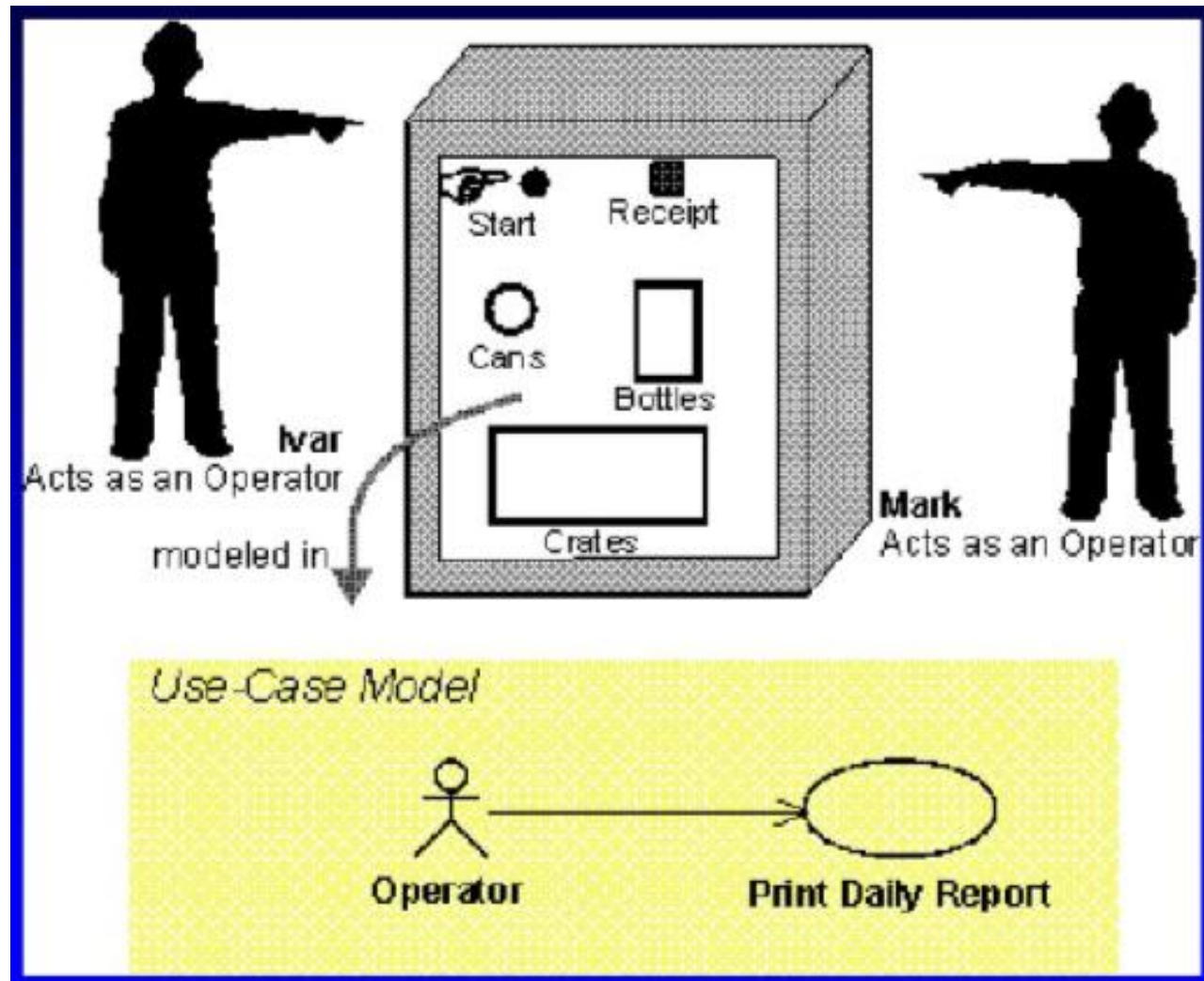
Actor



Phân biệt Actor – Instant Actor ?

- Sự khác biệt giữa một tác nhân và một người sử dụng độc lập trong hệ thống là tác nhân **biểu diễn một lớp (một tập) người sử dụng** chứ không phải là một cá nhân cụ thể nào.
- Một vài người sử dụng có thể đóng cùng một vai trò đối với hệ thống → chỉ thiết kế một tác nhân biểu diễn cho các người dùng trên
→ **Mỗi người dùng** cụ thể là **một thể hiện** của tác nhân (Instant Actor)

Actor – Ví dụ



Actor – Ví dụ

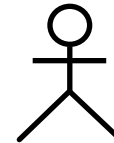
- Một hệ thống quản lý thư viện, cho phép người dùng có thể tra cứu thông tin của các quyển sách có trong thư viện.
- Hai sinh viên A và B sử dụng hệ thống để tra cứu thông tin
 - Chỉ có một tác nhân là “Người sử dụng”
 - A và B là hai thể hiện của tác nhân này.

Vai trò (Role)



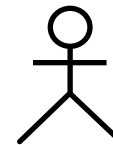
Charlie

**Charlie có vai trò như
một sinh viên**



Student

**Charlie có vai trò như
một giáo sư**



Professor

=> Một User có thể có nhiều vai trò

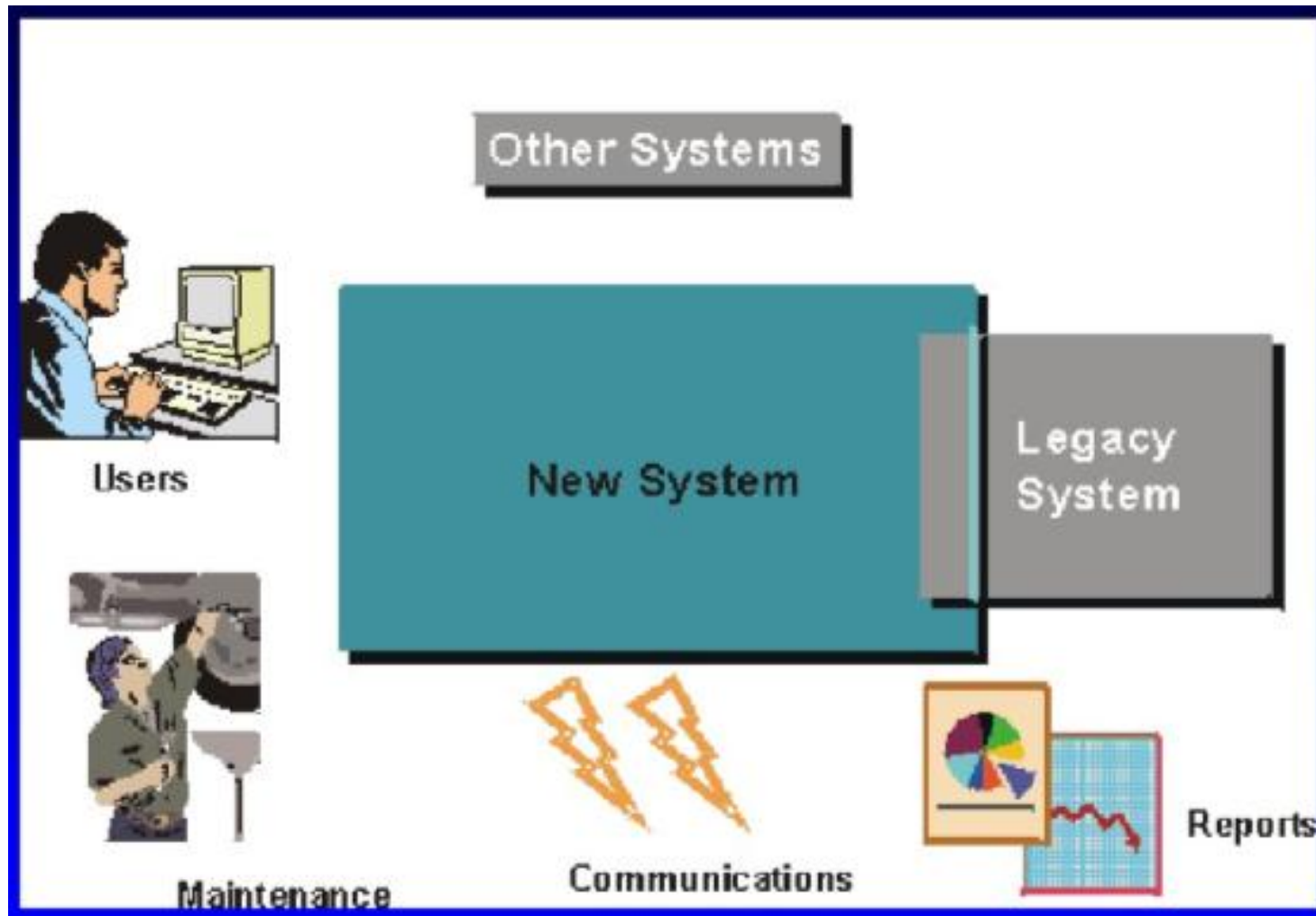
Vai trò (Role)

- Trong một vài tình huống, một người đóng một vai trò nào đó được mô hình hóa thành một actor trong hệ thống. Ví dụ như quản trị hệ thống.
- Cũng có trường hợp cùng một người dùng nhưng là thể hiện của nhiều tác nhân (trong trường hợp một cá nhân có nhiều vai trò).
- Ví dụ: người thủ thư tên A có thể có hai vai trò khác nhau trong hệ thống quản lý thư viện
 - Tác nhân “Người sử dụng” bình thường
 - Tác nhân “Người thủ thư”.

Làm thế nào để xác định Actor?

- Những gì xung quanh hệ thống sẽ trở thành tác nhân của hệ thống?
 - Những cá nhân độc lập sẽ sử dụng hệ thống.
 - Phân loại?
- nghĩ tới một vài cá nhân nào đó và đảm bảo rằng các tác nhân được thiết kế đáp ứng hầu hết các nhu cầu của họ.

Làm thế nào để xác định Actor?



Làm thế nào để xác định Actor?

- Các câu hỏi

1. Ai là người cung cấp, sử dụng hoặc lấy thông tin từ hệ thống ?
2. Ai sẽ sử dụng các tính năng của chương trình ?
3. Người quan tâm tới một yêu cầu nào đó ?
4. Nơi nào trong tổ chức (phòng ban, công ty) sẽ sử dụng hệ thống?
5. Ai là người duy trì, bảo dưỡng và quản lý hệ thống?
6. Những tài nguyên bên ngoài hệ thống là gì ?
7. Có những hệ thống nào khác tương tác với hệ thống này không?

Ví dụ

- Người dùng những chức năng chính của hệ thống
- Người dùng những chức năng phụ, như là quản trị hệ thống.
- Những thiết bị phần cứng bên ngoài
- Những hệ thống khác có tương tác trao đổi thông tin với hệ thống.
- Nếu xây dựng ứng dụng trên nền internet, có thể có tác nhân “vô danh”

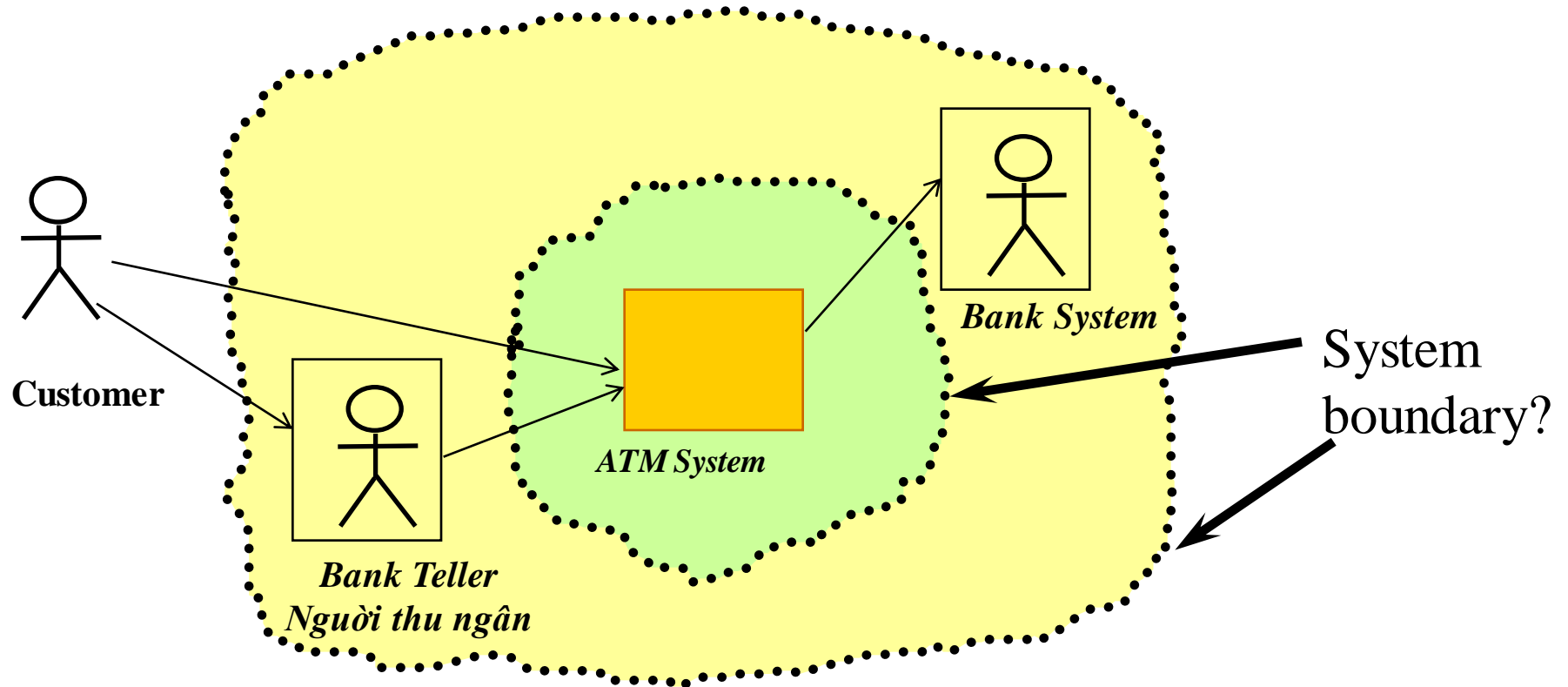
Ví dụ: Actor – phần cứng

- Phần mềm quản lý Siêu thị:
 - Đọc thông tin từ thiết bị đọc mã vạch
- Phần mềm quản lý cửa tự động:
 - Đọc thông tin từ camera
 - Phát lệnh điều khiển mở cửa
- Phần mềm quản lý ra vào các phòng trong công sở
 - Đọc tín hiệu từ đầu đọc thẻ từ
 - Phát lệnh điều khiển mở cửa
- Phần mềm chống trộm
 - Đọc tín hiệu từ camera, sensor
 - Phát lệnh điều khiển ra loa, đèn, điện thoại...

Ví dụ: Actor – phần mềm khác

- Kết xuất/nạp dữ liệu từ Excel
- Kết xuất dữ liệu báo cáo ra phần mềm gửi email (Microsoft Outlook, Outlook Express...)
- ...

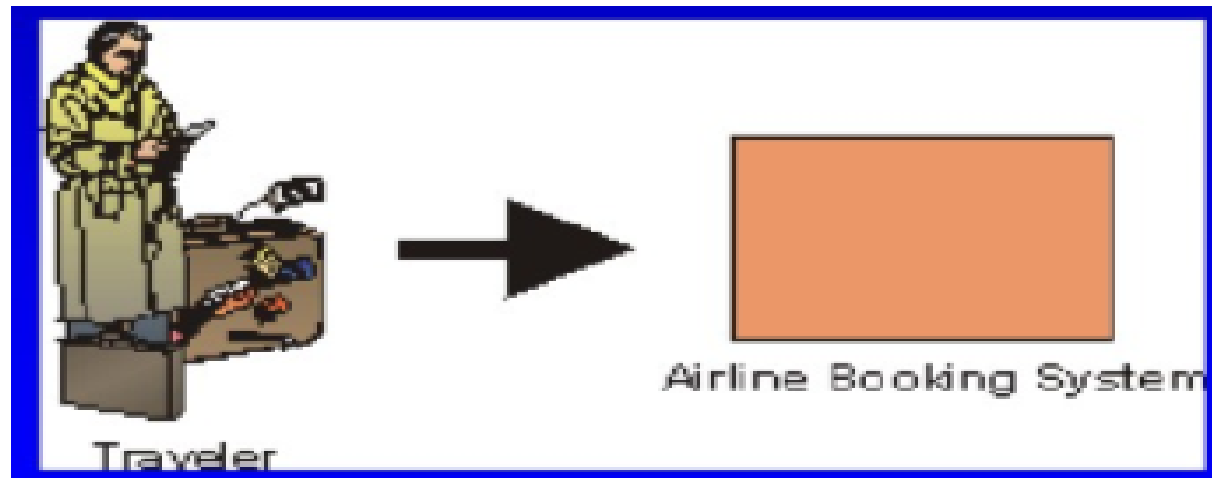
Actors và giới hạn hệ thống (System Boundary)



Actors và giới hạn hệ thống

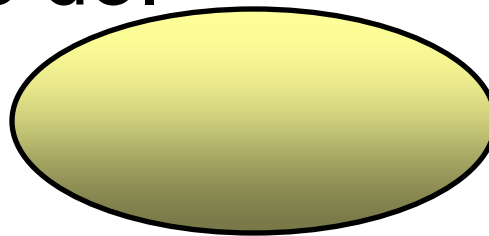
- Tìm kiếm tác nhân cũng có nghĩa là xác định phạm vi của hệ thống, giúp ta xác định mục đích và qui mô của hệ thống cần xây dựng.
- Chỉ những người nào có tương tác trực tiếp với hệ thống mới được xem là tác nhân
- Ví dụ: trong hệ thống đăng ký vé, cần xét các trường hợp
 - Khách hàng mua vé thông qua nhân viên du lịch (travel agent) → **không là tác nhân** của hệ thống.
 - Khách hàng có thể đăng ký vé trực tiếp thông qua internet → **tác nhân**.

Ví dụ



Khái niệm – Use Case

- Một use case xác định một tập các thể hiện use case
- Trong đó **mỗi thể hiện** là một chuỗi các hành động được hệ thống thực hiện và đem lại một kết quả thấy được có ý nghĩa đối với một actor cụ thể nào đó.



Use-Case

Giải thích

- Có thể hiểu một Use-Case là một chức năng của hệ thống, mang một ý nghĩa nhất định đối với người dùng
- Thể hiện Use Case và Use Case
 - Hệ thống thực hiện thao tác đăng nhập của nhân viên A
 - Hệ thống thực hiện thao tác đăng nhập của nhân viên
 - => “đăng nhập” là một Use Case

Làm thế nào để xác định Use Case?

- Đối với mỗi Actor, xác định những công việc nào liên quan đến Actor này?
- Hệ thống hỗ trợ những nghiệp vụ nào trong thế giới thực?
- Những thông tin nào cần được quản lý trong hệ thống?
- Những thông tin nào cần được kết xuất ra khỏi hệ thống?

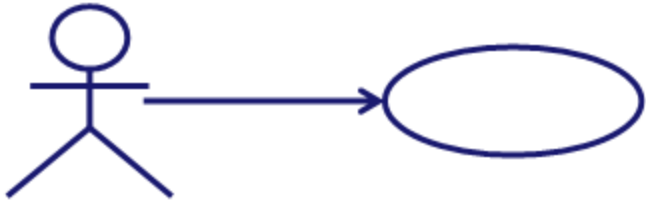
Các bước thực hiện

- Xác định qui trình nghiệp vụ được hỗ trợ
- Xác định các đối tượng thông tin cần quản lý
 - Các Use Case dạng quản lý, tra cứu, kết xuất liên quan đến các đối tượng thông tin này
- Các nghiệp vụ, các xử lý chính
- Các báo cáo, kết xuất
- Các nghiệp vụ liên quan đến quản lý, duy trì thông tin hệ thống
- Các chức năng liên quan đến các yêu cầu đặc biệt (an toàn, bảo mật, thay đổi giao diện, màu sắc...)

Ví dụ

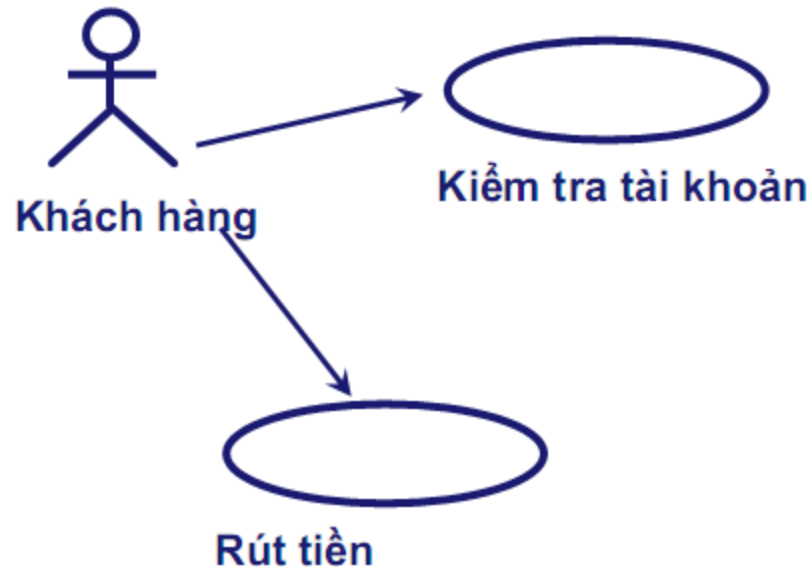
- Xét phần mềm Quản lý thư viện
 - Lập thẻ độc giả
 - Nhận sách mới
 - Tra cứu sách
 - Lập phiếu mượn
 - Nhận trả sách
 - Thay đổi quy định
 - ...

Sơ đồ Use case (Use case diagram)



Sự tương tác giữa Actor và Use-case

Chiều của mũi tên thể hiện vai trò chủ động trong sự tương tác



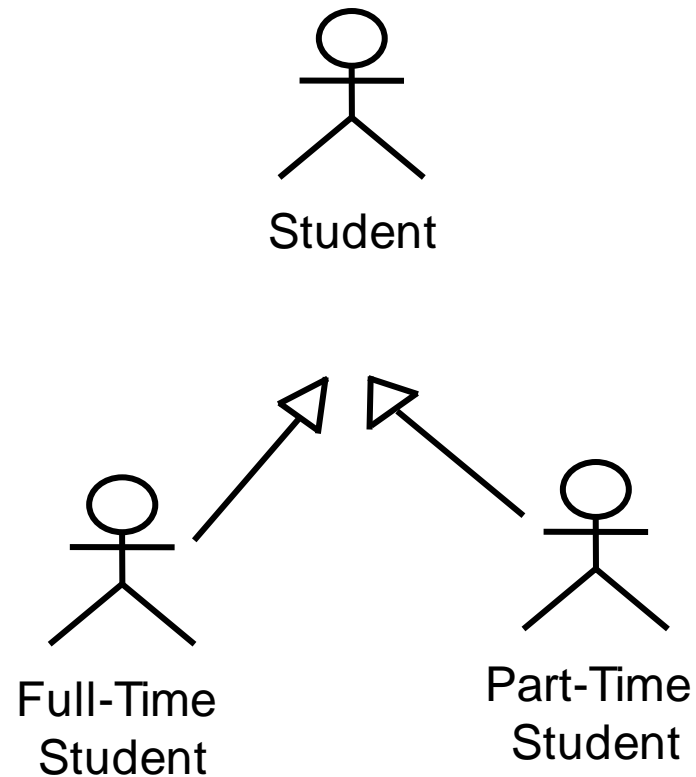
Nội dung

- Giới thiệu
- Phát biểu bài toán
- Bảng chú giải
- Các khái niệm chính
 - Actor
 - Use Case
 - Use Case Model
- **Mối quan hệ**
- Đặc tả use case
- Các đặc tả bổ sung
- Checkpoints

Các mối quan hệ

- Quan hệ giữa các Actor
- Quan hệ giữa Actor và Use Case
- Quan hệ giữa các Use Case

Actor Generalization (Tổng quát hóa)



Quan hệ giữa Actor & Use Case



Ví dụ: Hệ thống đăng ký học phần

- Hệ thống mới cho phép sinh viên đăng ký học phần và xem phiếu điểm từ một máy tính cá nhân được kết nối vào mạng nội bộ của trường. Các giáo sư cũng có thể truy cập hệ thống này để đăng ký lớp dạy và nhập điểm cho các môn học.
- Trường sẽ giữ lại cơ sở dữ liệu sẵn có về danh mục học phần mà lưu trữ toàn bộ thông tin về học phần. Hệ thống mới sẽ đọc các thông tin học phần trong CSDL cũ nhưng sẽ không cập nhật chúng. Phòng Đào tạo sẽ tiếp tục duy trì các thông tin học phần thông qua một hệ thống khác.
- Đầu mỗi học kỳ, sinh viên có thể yêu cầu danh sách các học phần được mở trong học kỳ đó. Thông tin về mỗi học phần, như tên giáo sư, khoa, và các môn học phần tiên quyết sẽ được cung cấp để giúp sinh viên chọn lựa.

Ví dụ (tt)

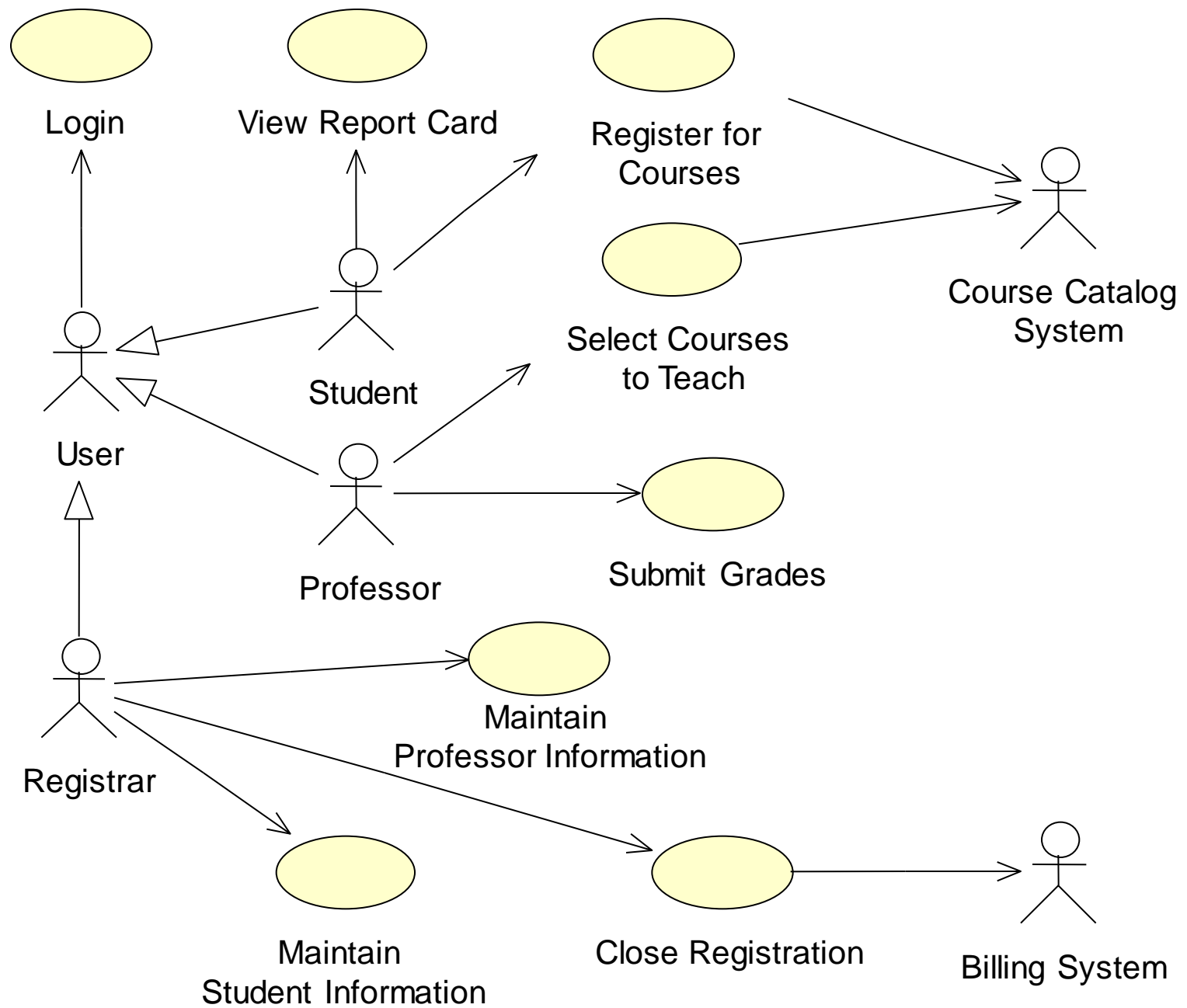
- Hệ thống mới cho phép sinh viên chọn bốn học phần được mở trong học kỳ tới. Thêm vào đó mỗi sinh viên có thể đưa ra hai môn học thay thế trong trường hợp không thể đăng ký theo nguyện vọng chính. Các học phần được mở có tối đa là 100 và tối thiểu là 30 sinh viên. Các học phần có ít hơn 30 sinh viên sẽ bị hủy. Đầu mỗi học kỳ, sinh viên có một khoảng thời gian để thay đổi lịch các học phần đã đăng ký. Sinh viên chỉ có thể thêm hoặc hủy học phần đã đăng ký trong khoảng thời gian này. Khi quá trình đăng ký hoàn tất cho một sinh viên, hệ thống đăng ký sẽ gửi thông tin tới hệ thống thanh toán (billing system) để sinh viên có thể đóng học phí. Nếu một lớp bị hết chỗ trong quá trình đăng ký, sinh viên phải được thông báo về sự thay đổi trước khi xác nhận lịch các học phần đã đăng ký.
- Ở cuối học kỳ, sinh viên có thể truy cập vào hệ thống để xem phiếu điểm. Bởi vì điểm của sinh viên là thông tin nhạy cảm cần được giữ kín, nên hệ thống phải có cơ chế bảo mật để ngăn chặn những truy cập không hợp lệ.
- Các giáo sư có thể truy cập vào hệ thống để đăng ký những học phần mà họ sẽ dạy. Họ cũng có thể xem danh sách các sinh viên đã đăng ký vào lớp của họ, và cũng có thể nhập điểm sau mỗi khóa học.

Ví dụ: Xác định các Actor

- Người dùng:
 - Sinh viên (Student)
 - Giáo sư (Professor)
 - Nhân viên giáo vụ (Registrar)
- Hệ thống khác:
 - Hệ thống quản lý danh mục học phần (Course Catalog System)
 - Hệ thống thanh toán (Billing System)

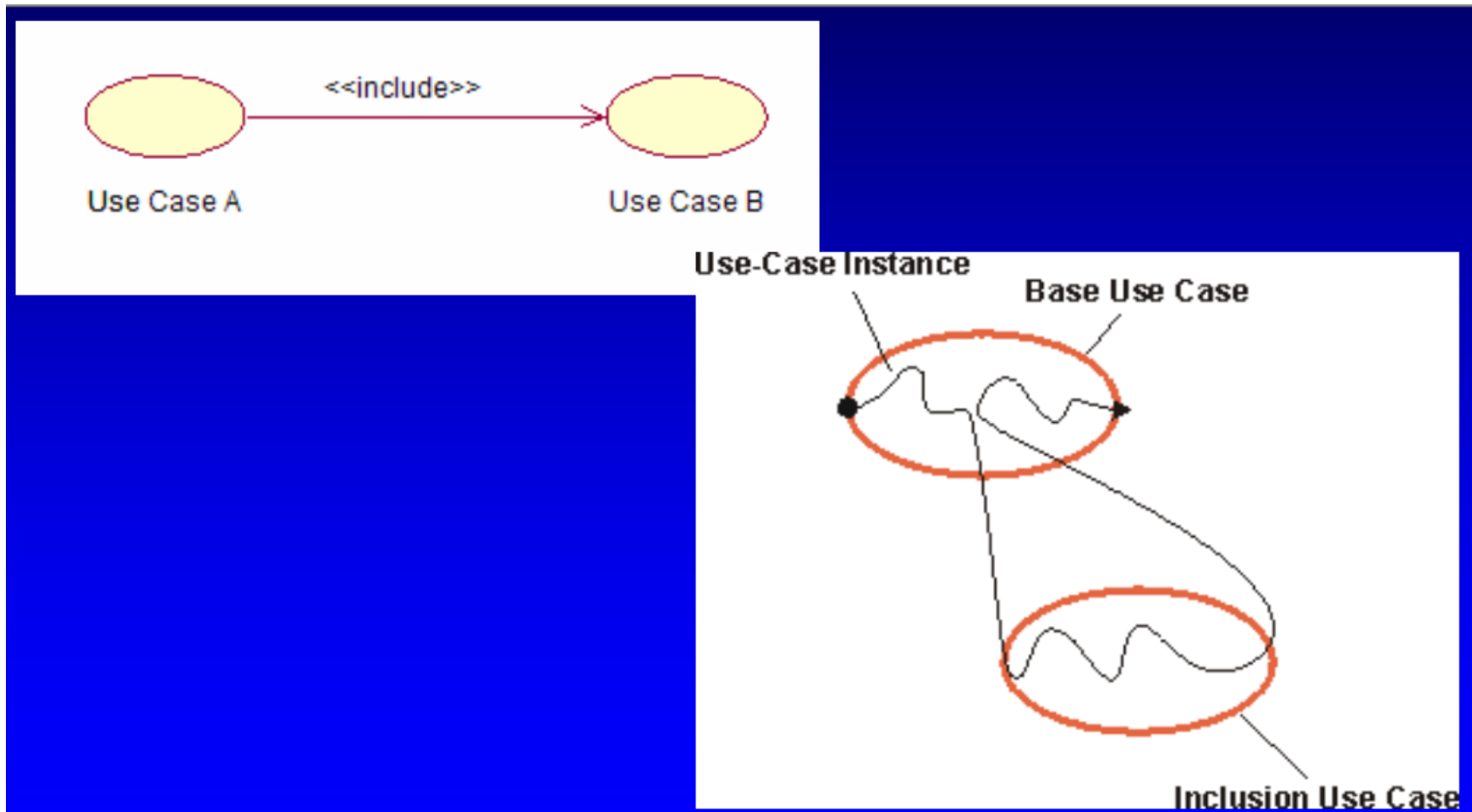
Ví dụ: Xác định các use case

- Chức năng cho mọi actor:
 - Đăng nhập hệ thống (Login)
- Các chức năng sử dụng bởi Student:
 - Đăng ký học phần (Register for Course)
 - Xem phiếu điểm (View Report Card)
- Các chức năng sử dụng bởi Professor:
 - Đăng ký môn dạy (Select Courses to Teach)
 - Nộp điểm (Submit Grades)
- Nhiệm vụ của Registrar:
 - Kết thúc đăng ký (Close Registration)
 - Cập nhật thông tin giáo sư (Maintain Professor Information)
 - Cập nhật thông tin sinh viên (Maintain Student Information)



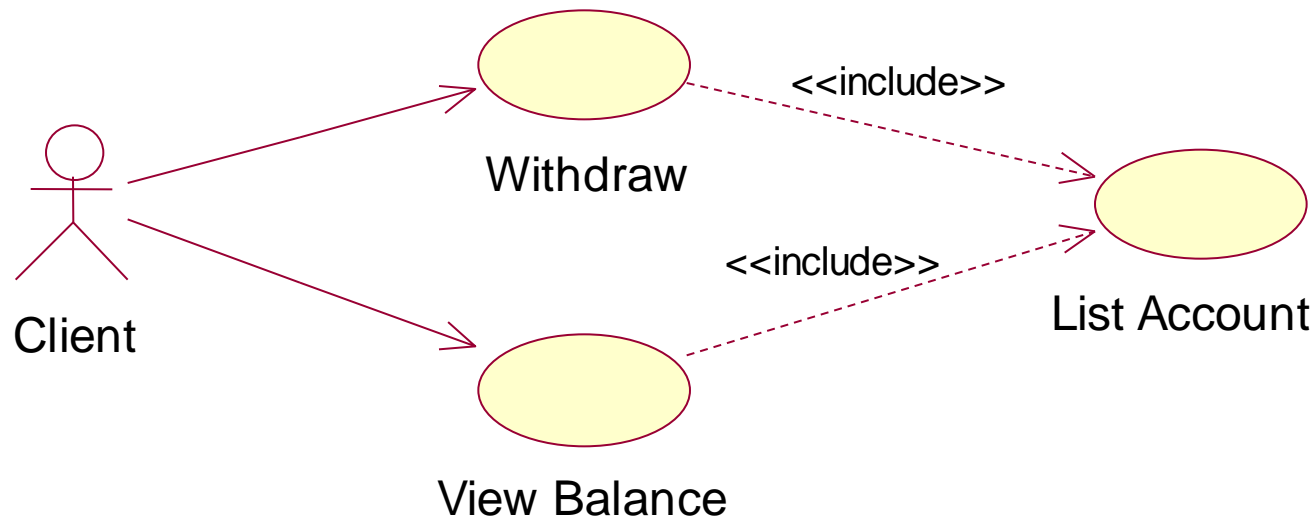
Quan hệ giữa các Use Case - Include

- <<Include>> - bao hàm, sử dụng

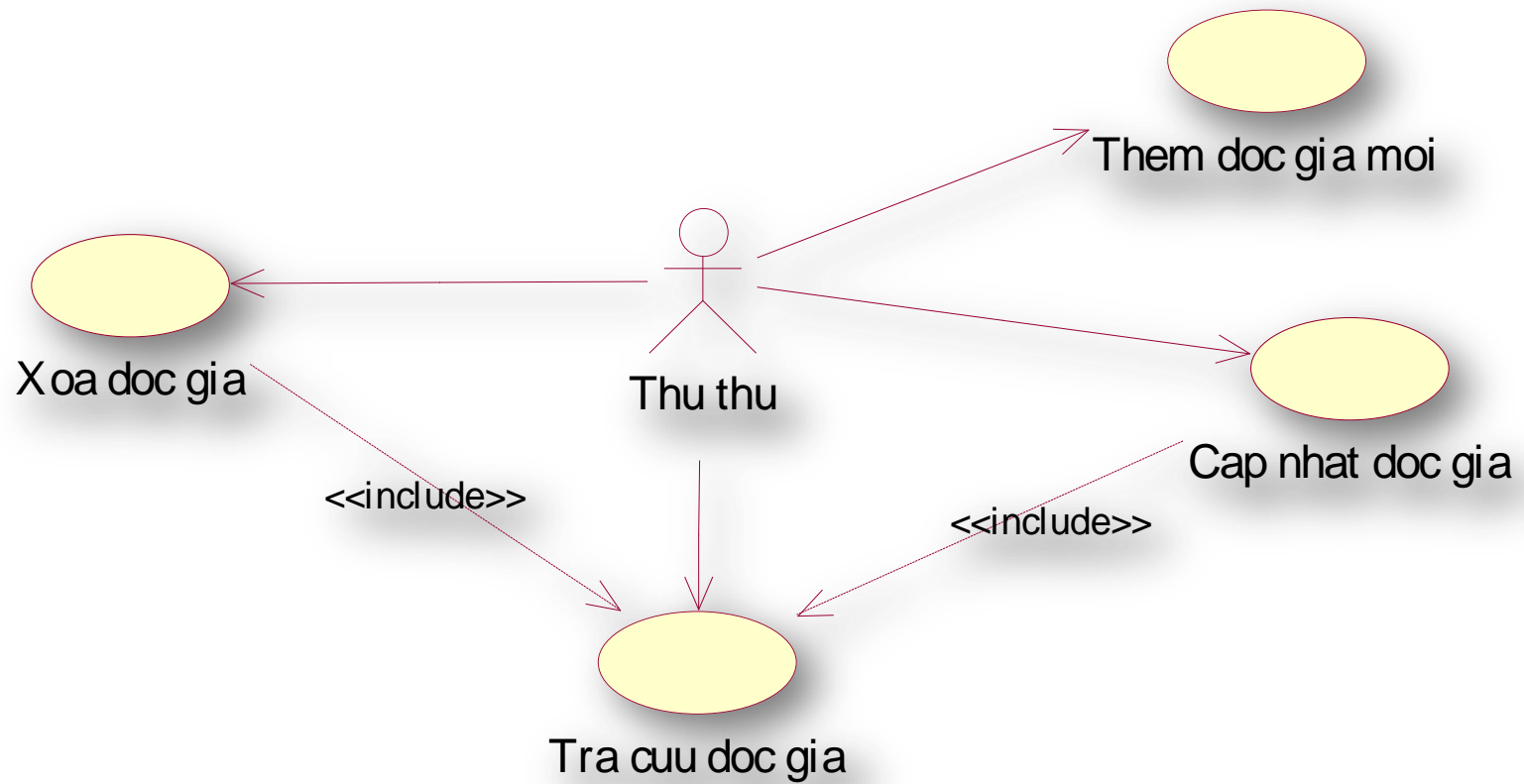


Quan hệ Include

- Một use case **luôn luôn** bao gồm dãy các ứng xử của một use case khác
- Được dùng để tách một dãy các ứng xử mà được dùng bởi nhiều use case

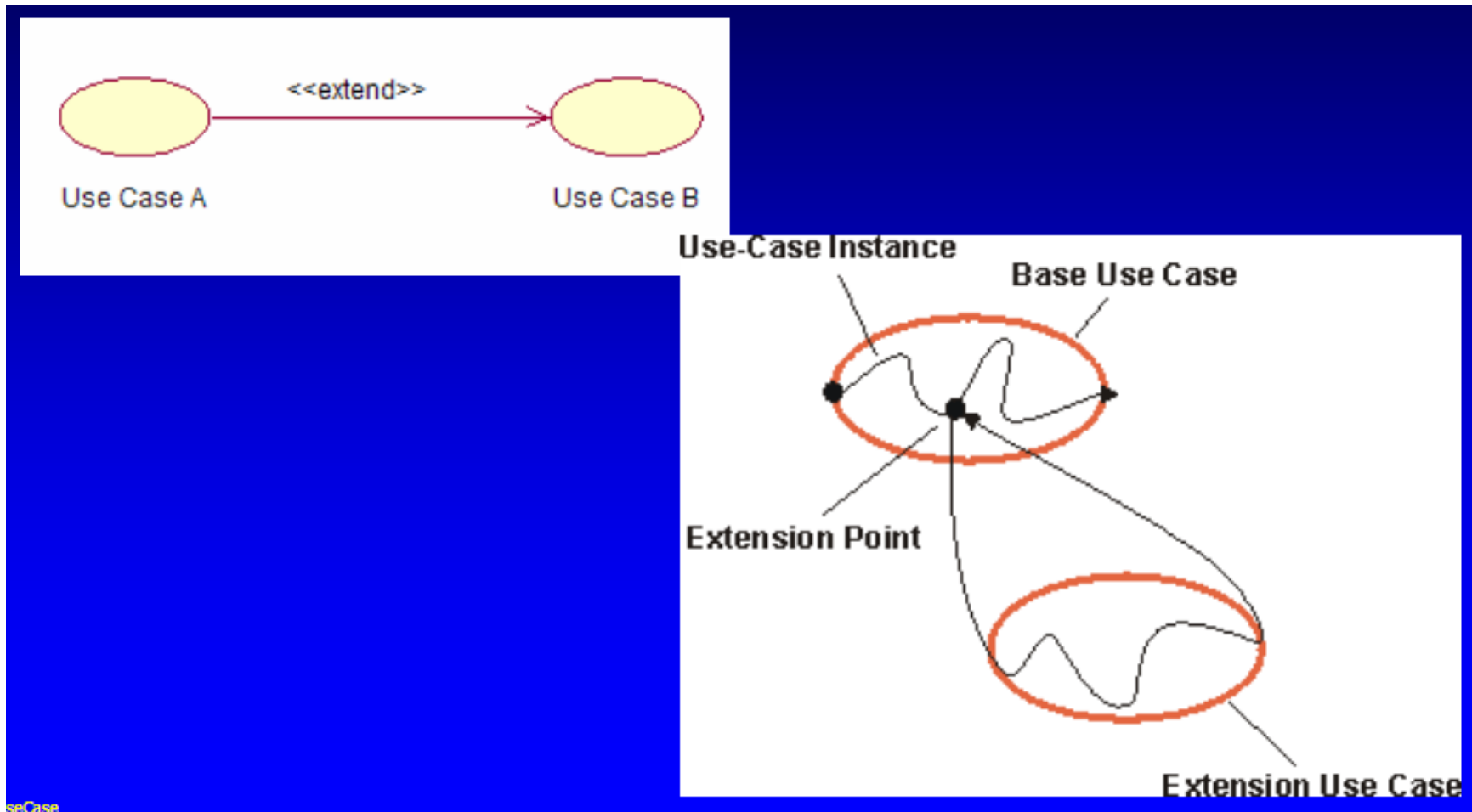


Ví dụ



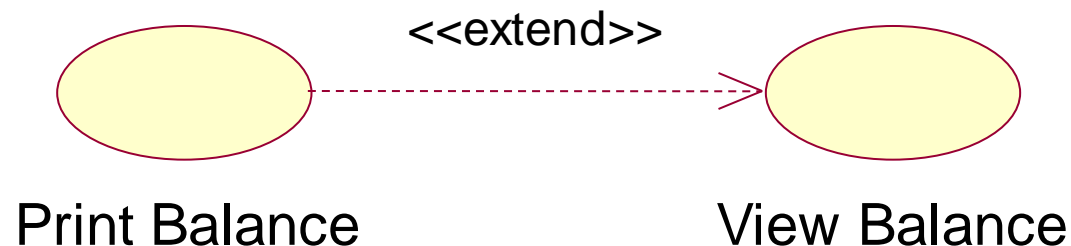
Quan hệ giữa các Use Case - Extend

- <<Extend>> - mở rộng

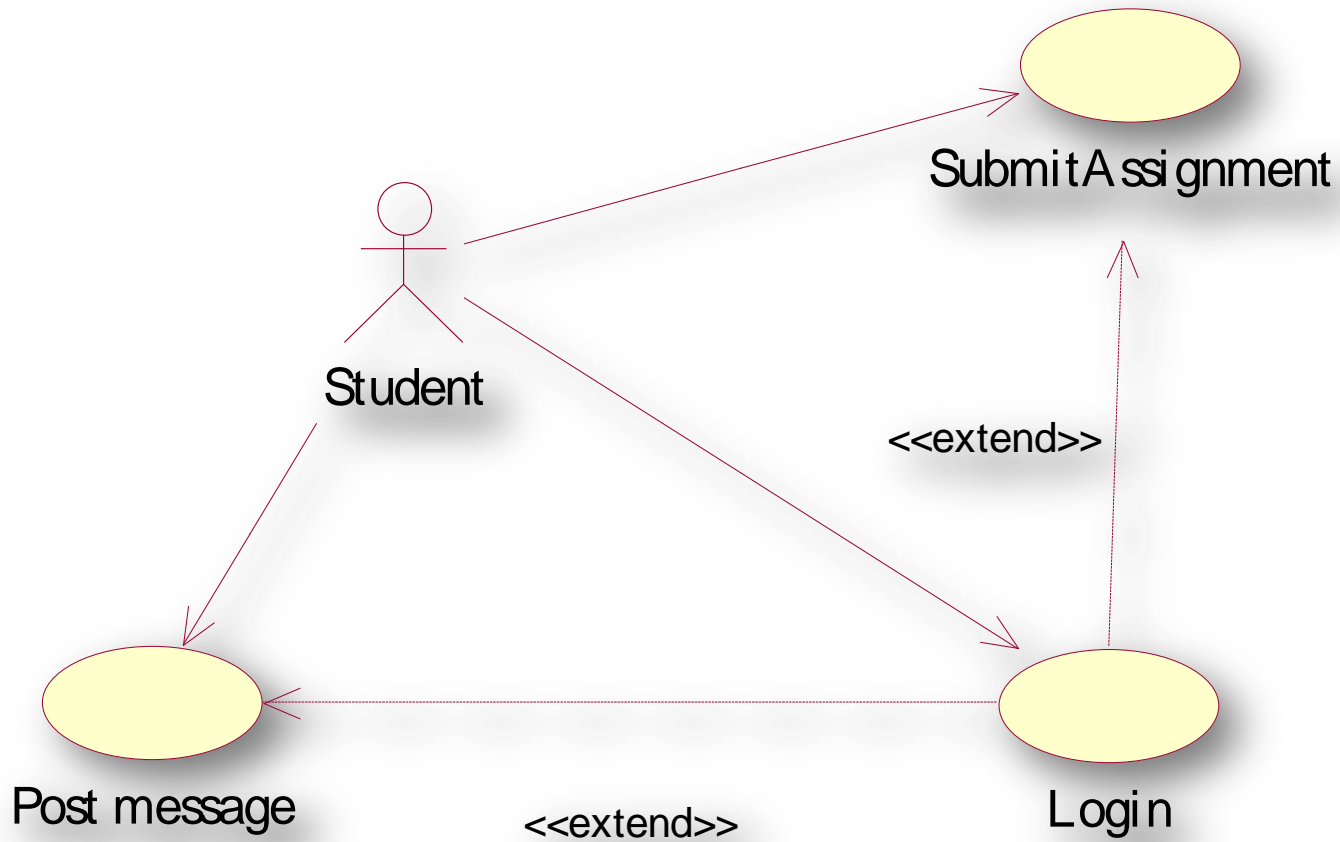


Quan hệ Extend

- Một use case cung cấp thêm chức năng cho một use case khác
 - Biểu diễn ứng xử **tùy chọn**, nhiều công việc khác nhau được thực hiện dựa vào việc chọn lựa của actor.
 - Điểm mở rộng (extension points) trình bày điều kiện khi nào việc mở rộng xảy ra.



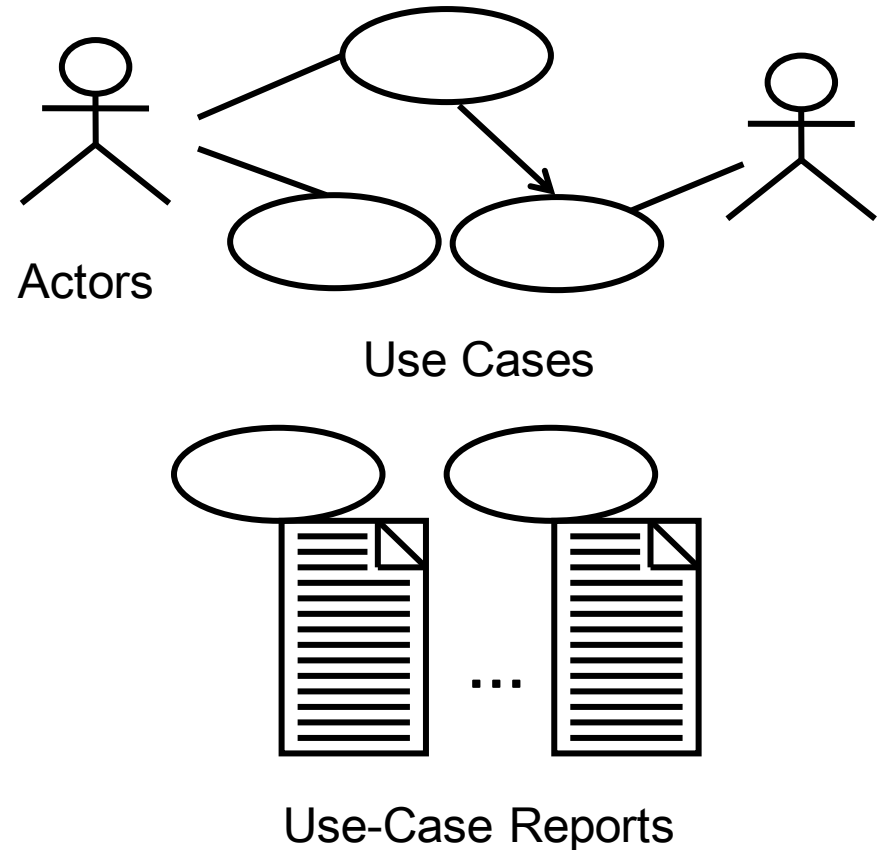
Ví dụ



Use-Case Model

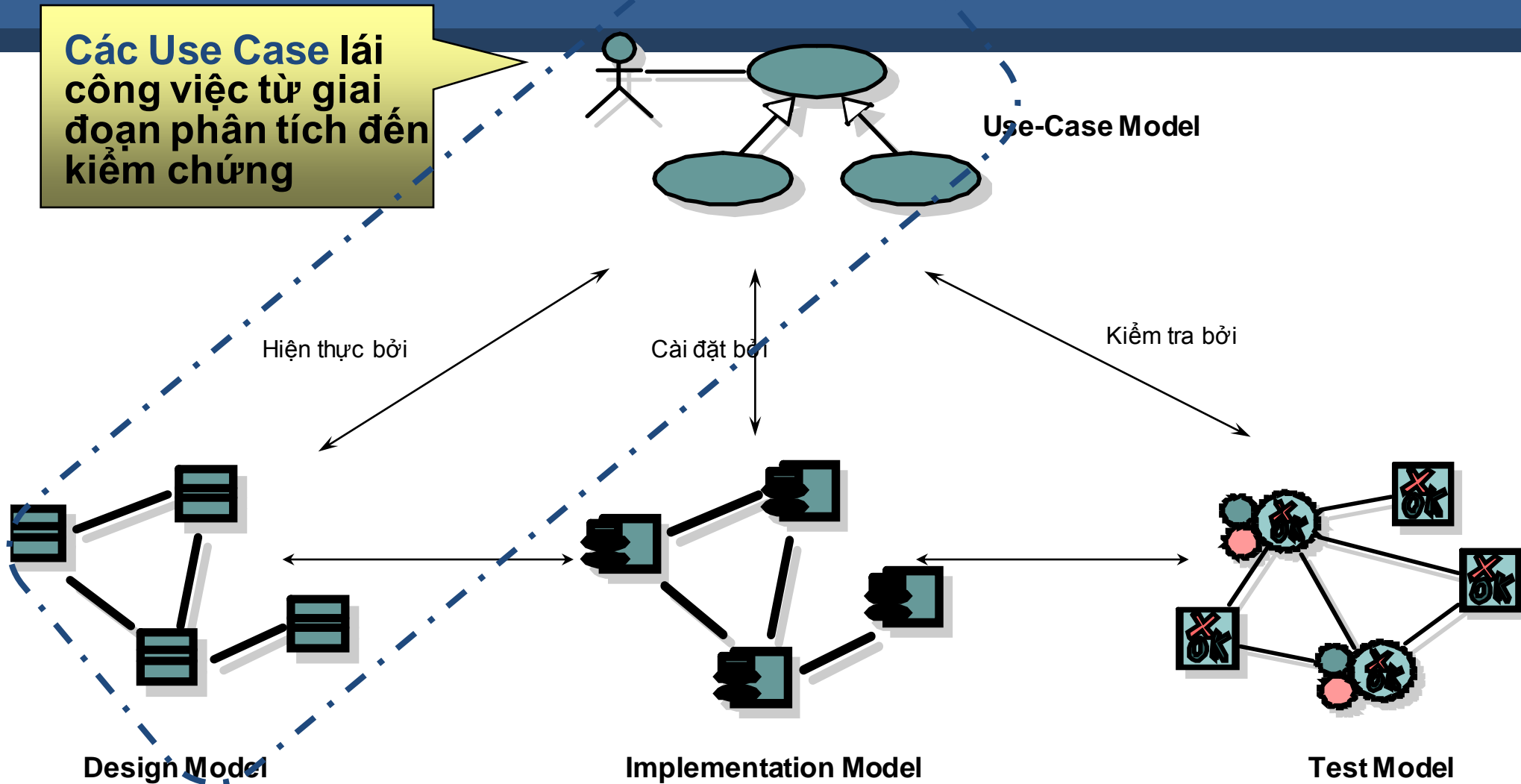
- Giới thiệu
- Survey Description
- Use-Case Packages
- Use Cases
- Actors
- Relationships
- Diagrams
- Use-Case View

Use-Case Model



Use-Case Model

Các Use Case lái công việc từ giai đoạn phân tích đến kiểm chứng

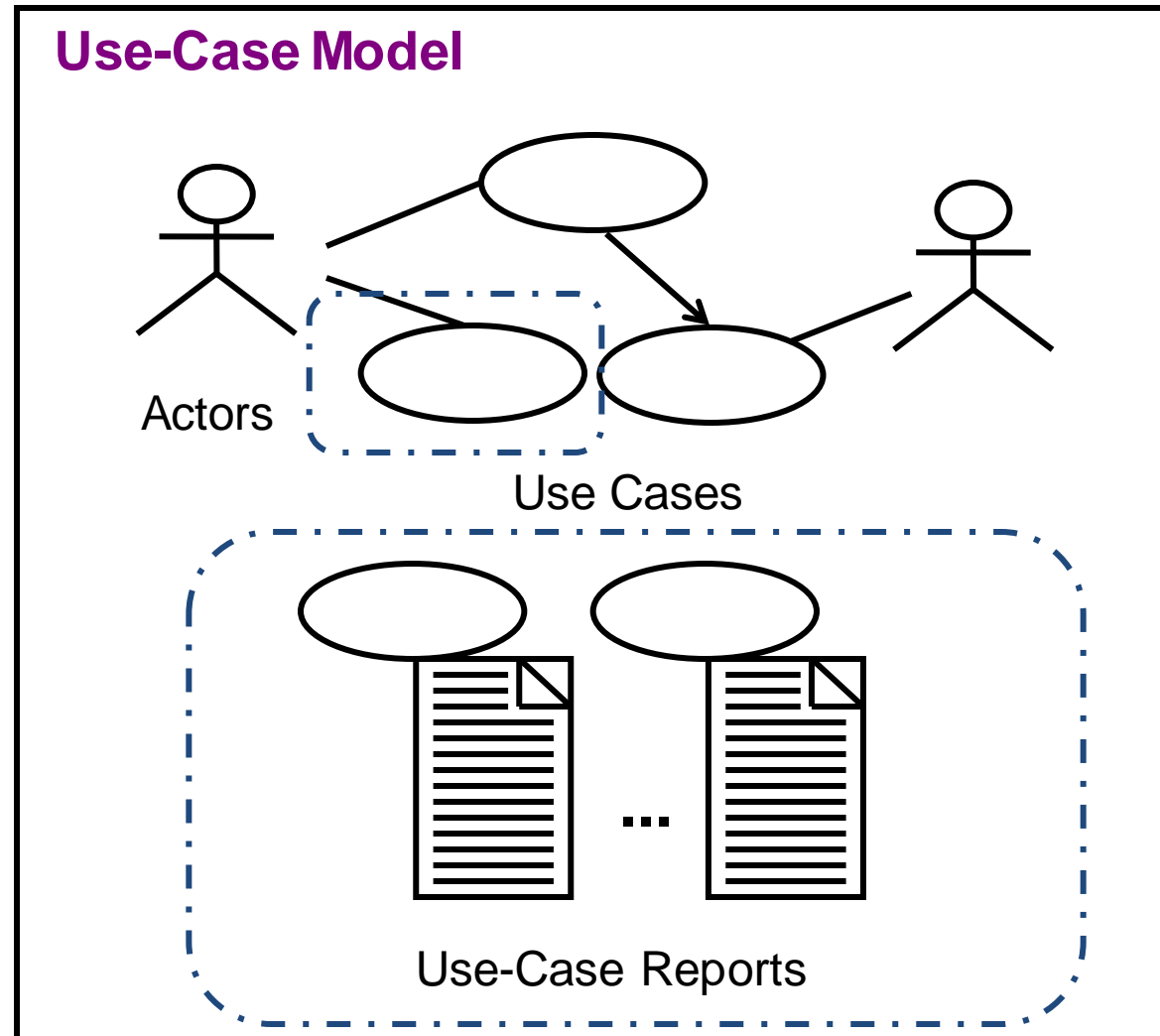


Nội dung

- Giới thiệu
- Phát biểu bài toán
- Bảng chú giải
- Các khái niệm chính
 - Actor
 - Use Case
 - Use Case Model
- Mối quan hệ
- **Đặc tả use case**
- Các đặc tả bổ sung
- Checkpoints

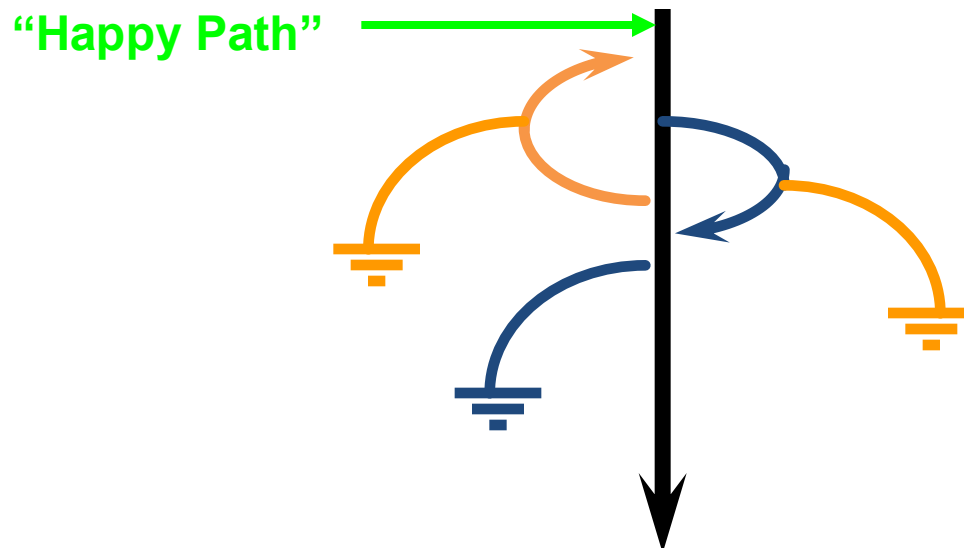
Đặc tả Use Case

- Tên
- Tóm tắt
- Luồng các sự kiện
- Các yêu cầu đặc biệt
- Pre-conditions
- Post-conditions
- Điểm mở rộng



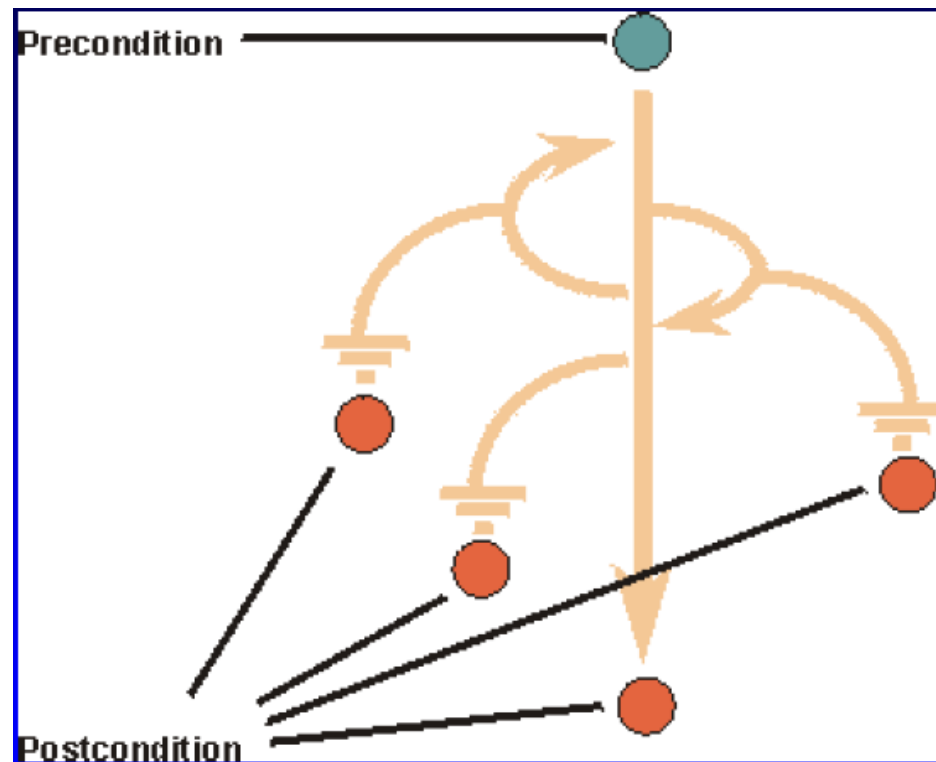
Luồng sự kiện (Use-Case Flows of Events)

- Của một *basic flow* (“Happy Path”)
- Một số *alternative flows*
 - Các biến thể thường gặp (Regular variants)
 - Các trường hợp bất thường (Odd cases)
 - **Exceptional flows** xử lý các tình huống lỗi



Pre và Post Conditions

- Pre-Conditions: điều kiện cần để use case xảy ra
- Post-Conditions: kết quả, trạng thái hệ thống sau khi kết thúc use case



Ví dụ: Luồng sự kiện use case Rút tiền

1. Use-Case bắt đầu khi khách hàng đưa thẻ tín dụng vào. Hệ thống đọc và thẩm tra thông tin của thẻ.
2. Hệ thống nhắc nhập số PIN. Hệ thống kiểm tra số PIN.
3. Hệ thống hỏi tác vụ nào khách hàng muốn thực hiện. Khách hàng chọn “Rút tiền.”
4. Hệ thống hỏi số lượng. Khách hàng nhập số lượng.
5. Hệ thống yêu cầu nhập kiểu tài khoản. Khách hàng chọn checking hoặc savings.
6. Hệ thống liên lạc với ATM network . . .

Ví dụ: Đặc tả use case mượn sách

- Tóm tắt:
 - Use case bắt đầu khi một có độc giả đến mượn sách. Mục tiêu của use case nhằm xử lý mượn sách cho độc giả
- Điều kiện tiên quyết
 - User đã đăng nhập vào hệ thống.
- Post-Conditions
 - Nếu thành công, cơ sở dữ liệu được cập nhật tương ứng
 - Nếu thất bại, trạng thái hệ thống không thay đổi

Ví dụ: Đặc tả use case mượn sách (tt)

- Dòng sự kiện chính
 - Hệ thống yêu cầu người dùng nhập thông tin độc giả. Nhân viên cung cấp thông tin thẻ độc giả
 - Hệ thống kiểm tra thông tin thẻ độc giả và thông tin các sách đang nợ của độc giả
 - Nếu hợp lệ, hệ thống yêu cầu nhập thông tin các sách độc giả muốn mượn.
 - Nhân viên nhập danh sách sách mượn và ngày thuê tương ứng.
 - Hệ thống kiểm tra và hiển thị ngày trả cho từng sách
 - Hệ thống cập nhật thông tin sách mượn và số lần mượn của độc giả

Ví dụ: Đặc tả use case mượn sách (tt)

- Dòng sự kiện khác:
 - Thẻ đọc giả hết hạn hoặc độc giả có sách quá hạn
 - Hệ thống hiển thị thông báo lỗi, từ chối cho mượn sách và use case kết thúc.
 - Độc giả chưa có thẻ đọc giả
 - Nhân viên kích hoạt use case Lập thẻ đọc giả để đăng ký và cấp thẻ cho độc giả.
 - Nếu sách mượn đã hết
 - Hệ thống hiển thị thông báo nhắc nhở.
 - Xử lý không cho mượn sách
 - Nếu số sách độc giả đang mượn > 3 , hệ thống từ chối lần mượn của độc giả

Ví dụ: Đặc tả use case Select Courses to Teach

- Tóm tắt
 - Use case này cho phép một giáo sư chọn từ danh mục học phần các lớp học mà mình có thể dạy được và muốn dạy trong học kỳ sắp tới.
- Điều kiện tiên quyết
 - Giáo sư phải đăng nhập vào hệ thống trước khi use case bắt đầu.
- Post-Conditions
 - Nếu use case thành công, các lớp mà giáo sư chọn dạy sẽ được cập nhật. Ngược lại, trạng thái của hệ thống không đổi.

Ví dụ: Đặc tả use case Select Courses to Teach (tt)

- Dòng sự kiện chính

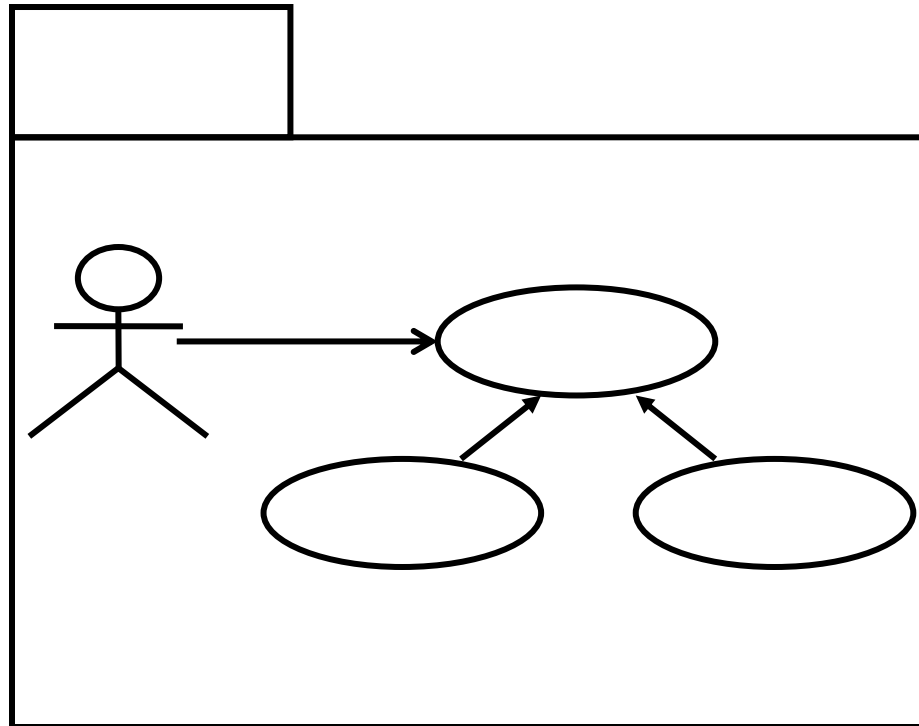
Use case này bắt đầu khi một giáo sư muốn đăng ký dạy một số lớp trong học kỳ sắp tới.

1. Hệ thống truy xuất và hiển thị danh sách các lớp mà giáo sư có thể dạy trong học kỳ hiện tại. Hệ thống cũng truy xuất và hiển thị các lớp học mà giáo sư này đã đăng ký dạy.
2. Giáo sư chọn thêm/bỏ bớt các lớp mà mình muốn dạy trong học kỳ sắp tới.
3. Hệ thống xóa giáo sư ra khỏi những lớp bị giáo sư bỏ bớt.
4. Hệ thống kiểm tra các lớp học được chọn xem có mâu thuẫn với nhau hay không (ví dụ như có cùng ngày, giờ dạy). Nếu không có mâu thuẫn, hệ thống cập nhật thông tin lớp học cho mỗi lớp học được giáo sư chọn (ví dụ như ghi nhận giáo sư là người giảng dạy cho lớp này).

Ví dụ: Đặc tả use case Select Courses to Teach (tt)

- Dòng sự kiện khác
 - Không có lớp học nào
Nếu trong Dòng sự kiện chính, giáo sư không chọn dạy bất cứ môn nào được mở trong học kỳ sắp tới hệ thống sẽ hiển thị thông báo lỗi. Giáo sư xác nhận thông báo này và use case kết thúc.
 - Mâu thuẫn trong lịch dạy
Nếu hệ thống tìm thấy mâu thuẫn trong lịch dạy khi giáo sư đăng ký lớp dạy, hệ thống sẽ hiển thị một thông báo lỗi. Hệ thống cũng chỉ ra các lớp học gây mâu thuẫn. Giáo sư có thể hoặc giải quyết mâu thuẫn này (ví dụ như hủy dạy một số lớp) hoặc hủy thao tác. Trong trường hợp này, chọn lựa của giáo sư sẽ bị mất và use case kết thúc.
 - Hệ thống Danh mục học phần không sẵn sàng
Nếu hệ thống không thể kết nối được với Hệ thống Danh mục học phần, hệ thống sẽ hiển thị một thông báo lỗi. Giáo sư xác nhận thông báo lỗi và use case kết thúc.
 - Đăng ký học phần đã bị đóng
Nếu khi use case bắt đầu, nó xác định được quá trình đăng ký cho học kỳ này đã bị đóng, một thông báo sẽ được hiển thị cho giáo sư và use case kết thúc. Giáo sư không thể thay đổi lớp dạy sau khi quá trình đăng ký cho học kỳ này đã bị đóng. Nếu một thay đổi giáo sư cần thiết xảy ra sau khi quá trình đăng ký bị đóng, nó sẽ được xử lý bên ngoài phạm vi của hệ thống.

Package trong Use-Case Model

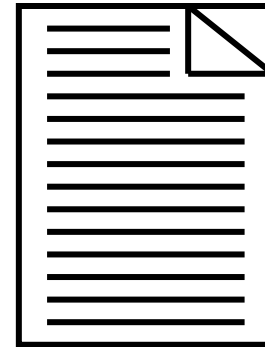


Nội dung

- Giới thiệu
- Phát biểu bài toán
- Bảng chú giải
- Các khái niệm chính
 - Actor
 - Use Case
 - Use Case Model
- Mối quan hệ
- Đặc tả use case
- **Các đặc tả bổ sung**
- Checkpoints

Các đặc tả bổ sung

- Functionality
- Tính khả dụng (Usability)
- Tính tin cậy (Reliability)
- Tính hiệu năng (Performance)
- Tính hỗ trợ (Supportability)
- Các ràng buộc thiết kế



Supplementary
Specification

Ví dụ: Các đặc tả bổ sung

- **Chức năng**

- Hỗ trợ nhiều người dùng làm việc đồng thời.
- Nếu một lớp bị hết chỗ khi một sinh viên đang đăng ký học của lớp đó thì sinh viên này phải được thông báo.

- **Tính khả dụng**

- Giao diện người dùng tương thích Windows 7/8.

- **Tính ổn định**

- Hệ thống phải hoạt động liên tục 24 giờ/ngày, 7 ngày/tuần, với thời gian ngừng hoạt động không quá **10%**.

- **Hiệu suất**

- Hệ thống phải hỗ trợ đến 2000 người dùng truy xuất CSDL trung tâm đồng thời bất kỳ lúc nào, và đến 500 người dùng truy xuất các server cục bộ.
- Hệ thống phải cho phép truy xuất đến CSDL danh mục học phần cũ với độ trễ không quá 10 giây.
- Hệ thống phải có khả năng hoàn tất 80% giao dịch trong vòng 2 phút.

Ví dụ: Các đặc tả bổ sung (tt)

- **Sự hỗ trợ**
 - Không có.
- **Tính bảo mật**
 - Hệ thống phải ngăn chặn sinh viên thay đổi lịch học của người khác, và ngăn các giáo sư thay đổi lớp dạy của giáo sư khác.
 - Chỉ có giáo sư mới có thể nhập điểm cho sinh viên.
 - Chỉ có cán bộ đào tạo mới được phép thay đổi thông tin của sinh viên.
- **Các ràng buộc thiết kế**
 - Hệ thống phải tích hợp với hệ thống có sẵn, Hệ thống danh mục học phần, một CSDL RDBMS.
 - Hệ thống phải cung cấp giao diện dựa trên Windows.

Nội dung

- Giới thiệu
- Các khái niệm chính
 - Actor
 - Use Case
 - Use Case Model
- Mối quan hệ
- Đặc tả use case
- Phát biểu bài toán
- Bảng chú giải
- Các đặc tả bổ sung
- **Checkpoints**

Checkpoints: Use-Case Model

- Use-case model có dễ hiểu không?
- Sau khi nghiên cứu use-case model, bạn có hình thành được một ý tưởng rõ ràng về các chức năng của hệ thống và cách thức mà chúng liên hệ với nhau ?
- Đã xác định hết tất cả các actor? Tất cả các yêu cầu chức năng được thỏa hay chưa?
- Use-case model có chứa các hành vi vô dụng nào không?
- Việc chia model thành các use-case package có xác đáng?

Checkpoints: Actors

- Đã xác định hết tất cả các actor?
- Mỗi actor có tham gia vào ít nhất một use case?
- Mỗi actor thật sự có một vai trò (role)? Có cần ghép hoặc tách các actor không?
- Có tồn tại 2 actor dùng cùng một vai trò đối với 1 use case không?
- Tên của các actor có gợi nhớ không? Users và customers có hiểu tên của chúng?

Checkpoints: Requirements: Use Cases

- Mỗi use case có ít nhất một actor tương tác?
- Các use case có độc lập với nhau?
- Tồn tại các use case có các luồng sự kiện và các hành vi tương tự nhau không?
- Liệu các use case có tên duy nhất, gợi nhớ, và dễ hiểu để chúng không bị nhầm lẫn trong các giai đoạn sau?
- Các khách hàng và người dùng có hiểu tên và mô tả của các use case không?

Checkpoints: Đặc tả Use-Case

- Use case có đủ rõ ràng đối với những người muốn hiện thực?
- Mục đích của use-case có rõ ràng?
- Mô tả sơ lược (Brief description) có cho ta hình ảnh trung thực của use-case?
- Có xác định rõ luồng sự kiện của use-case như thế nào và khi nào bắt đầu và kết thúc?
- Chuỗi các giao tiếp giữa actor và use-case có tiện nghi không (từ góc nhìn của người dùng)?
- Các tương tác và các thông tin trao đổi của actor có rõ ràng?
- Có use-case nào quá phức tạp không?
- Các luồng sự kiện (basic và alternative) được mô hình đúng đắn?

Checkpoints: Bảng chú giải

- Các thuật ngữ có định nghĩa rõ ràng và súc tích?
- Mỗi thuật ngữ có dùng đâu đó trong các mô tả use-case?
- Các thuật ngữ có được sử dụng hợp lý trong các mô tả ngắn về các actor và use case?

Review

- Các thành phần chính của Đặc tả yêu cầu (Requirements)?
- Các thành phần của đặc tả yêu cầu được dùng để làm gì?
- Use-case model là gì?
- Actor là gì?
- Use case là gì? Liệt kê một số quan hệ trong mô hình.
- Đặc tả bổ sung là gì và nó chứa những gì?
- Bảng chú giải là gì và nó chứa những gì?