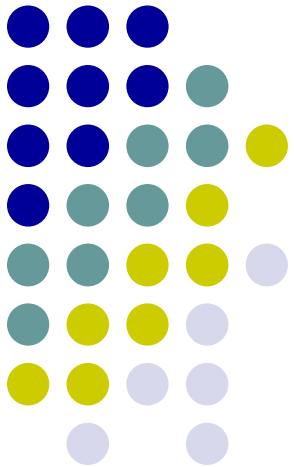
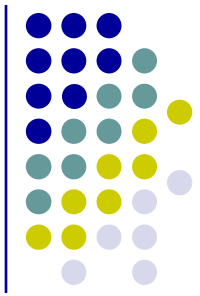
Phân tích hướng đối tượng UML

Giáo viên: Đỗ Thị Mai Hường

Bộ môn : Các hệ thống thông tin

Khoa : CNTT - Học viện kỹ thuật quân sự

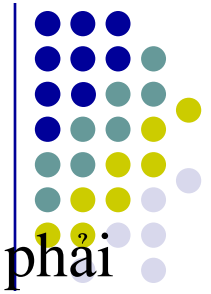




Bài 4

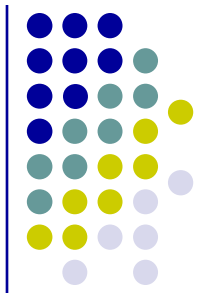
Mô hình hóa ca sử dụng

Giới thiệu mô hình hóa UC



- Trong pha thu thập yêu cầu và phân tích hệ thống thường phải xây dựng các biểu đồ cho
  - Mô hình nghiệp vụ
  - Mô hình ca sử dụng
  - Mô hình giao diện người sử dụng
- Mô hình ca sử dụng mô tả hệ thống được sử dụng như thế nào
  - Use case (UC) hệ thống và tác nhân hệ thống xác định phạm vi hệ thống
    - UC là những gì bên trong hệ thống
    - Actor là những gì bên ngoài hệ thống
  - Biểu đồ UC mô tả tương tác giữa các UC và tác nhân để hình thành chức năng hệ thống
- Sự khác nhau giữa mô hình hóa nghiệp vụ và mô hình hóa ca sử dụng
  - Mô hình hóa nghiệp vụ tập trung vào tổ chức của cơ quan
  - Mô hình hóa hệ thống tập trung vào hệ thống đang xây dựng

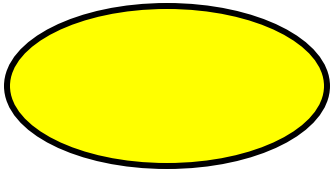
Các khái niệm mô hình hóa UC



- Các khái niệm cơ bản
  - Ca sử dụng (Use case-UC)
  - Tác nhân (Actor)
  - Quan hệ (Relationship)
  - Biểu đồ ca sử dụng (Use case Diagram)

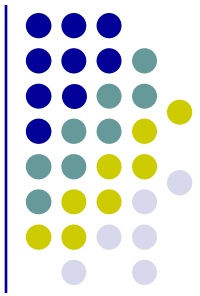
Use case, tác nhân là gì?

- Use case?
  - UC được xem là chức năng của hệ thống cung cấp từ quan điểm của người dùng.
  - UC dùng để mô tả hệ thống mới về mặt chức năng, mỗi một chức năng sẽ được biểu diễn như một hoặc nhiều UC.
  - Không phải là thiết kế, cài đặt mà là một phần của vấn đề cần giải quyết
- Kí hiệu

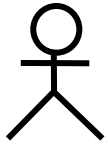


Purchase Ticket

Use case, tác nhân là gì?...



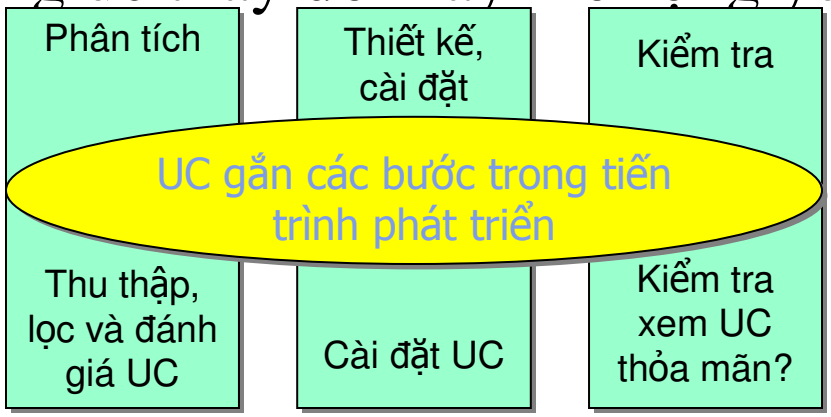
- Tác nhân?
  - Là đối tượng bên ngoài tương tác với hệ thống theo 3 hình thức:
    - Tương tác trao đổi thông tin với hệ thống hoặc sử dụng chức năng.
    - Cung cấp đầu vào hoặc nhận thông tin đầu ra từ hệ thống.
    - Không điều khiển hoạt động của hệ thống.
  - Đặt tên: theo vai trò, không theo tên cụ thể vì nó là lớp
- Kí hiệu:



Customer

Xây dựng UC để làm gì?

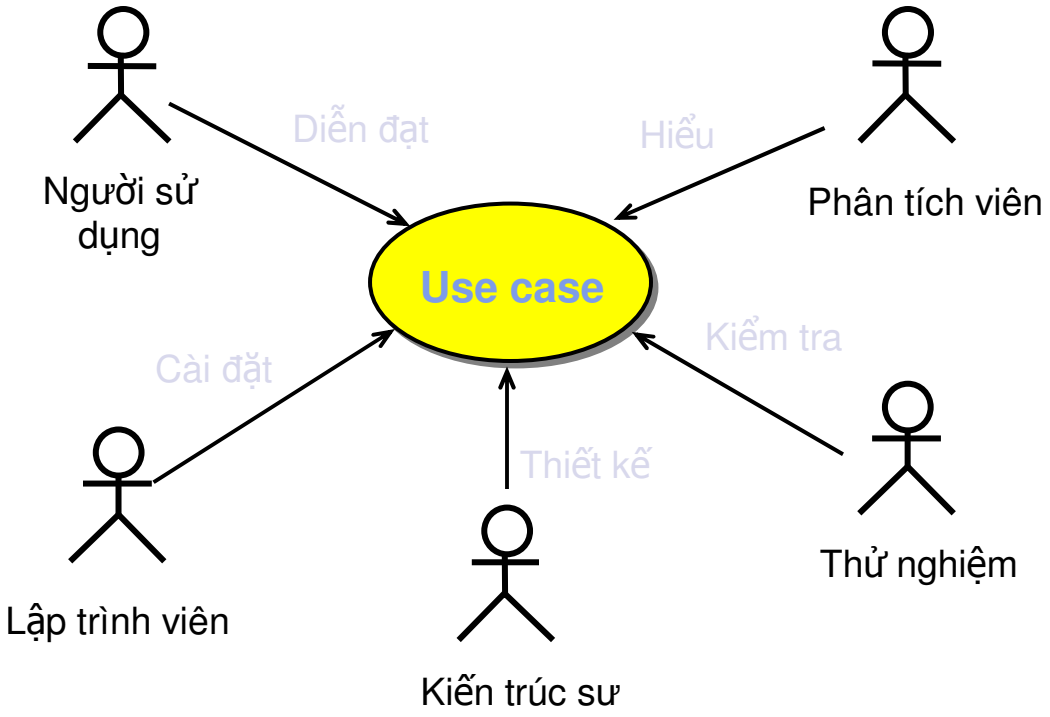
- Hình thành và mô tả yêu cầu chức năng hệ thống
  - Là kết quả thỏa thuận giữa khách hàng và người phát triển hệ thống phần mềm
- Cho phép mô tả rõ ràng và nhất quán cái hệ thống sẽ làm
  - Mô hình có khả năng được sử dụng xuyên suốt quá trình phát triển
- Cung cấp cơ sở để kiểm tra, thử nghiệm hệ thống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



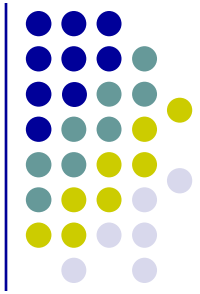
UC và tiến trình phát triển

Xây dựng UC để làm gì?

- Ai quan tâm đến UC?

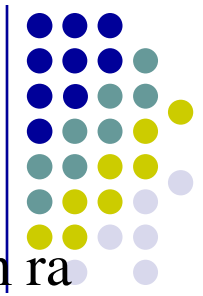


Tìm kiếm tác nhân như thế nào?



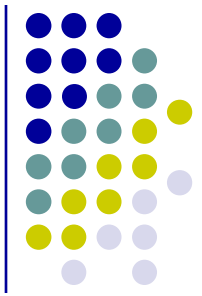
- Hãy trả lời các câu hỏi sau để tìm ra tác nhân hệ thống
  - Ai sẽ sử dụng chức năng chính của hệ thống?
  - Ai giúp hệ thống làm việc hàng ngày?
  - Ai quản trị, bảo dưỡng để hệ thống làm việc liên tục?
  - Hệ thống quản lý thiết bị phần cứng nào?
  - Hệ thống đang xây dựng tương tác với hệ thống khác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

Tìm kiếm UC như thế nào?



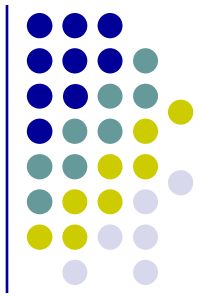
- Với mỗi tác nhân đã tìm ra, hãy trả lời các câu hỏi sau để tìm ra các Use case hệ thống
  - Tác nhân yêu cầu hệ thống thực hiện chức năng nào?
  - Tác nhân cần đọc, tạo lập, bãi bỏ, lưu trữ, sửa đổi các thông tin nào trong hệ thống?
  - Tác nhân cần thông báo cho hệ thống sự kiện xảy ra trong nó?
  - 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

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?

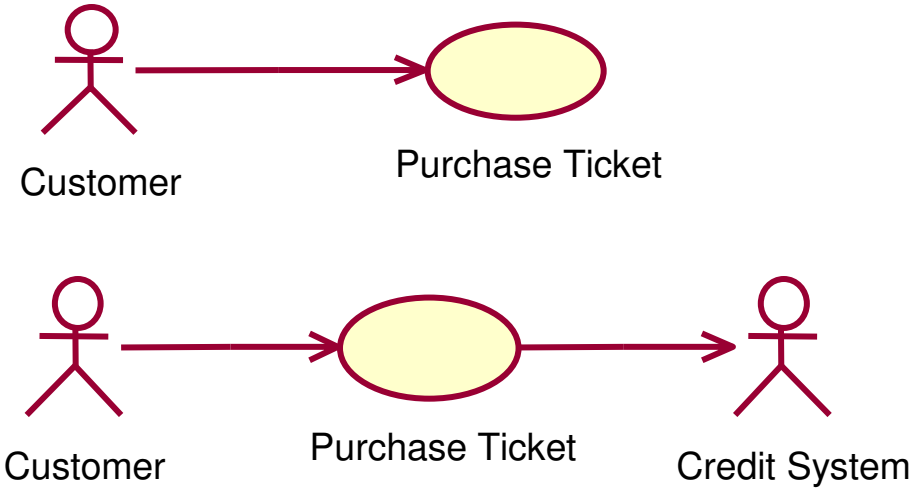
Các quan hệ



- Quan hệ kết hợp (Association)
- Quan hệ gộp (Includes)
- Quan hệ mở rộng (Extends)
- Quan hệ tổng quát hóa (Generalization)

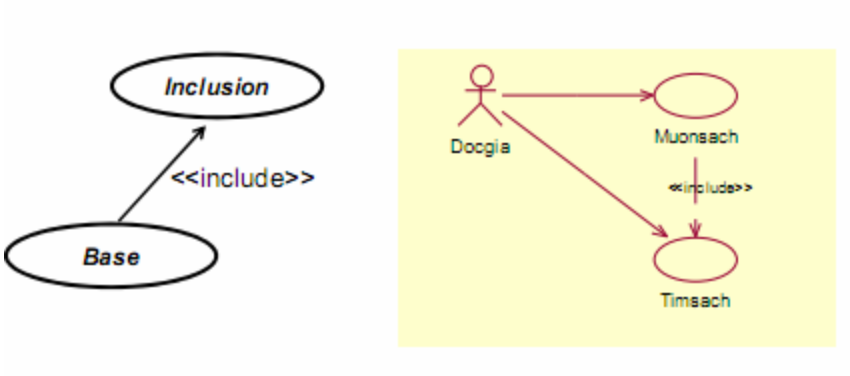
Các quan hệ

- Quan hệ kết hợp (Association)
  - Là loại quan hệ giữa tác nhân và UC
  - Mũi tên cho biết ai là người khởi xướng giao tiếp

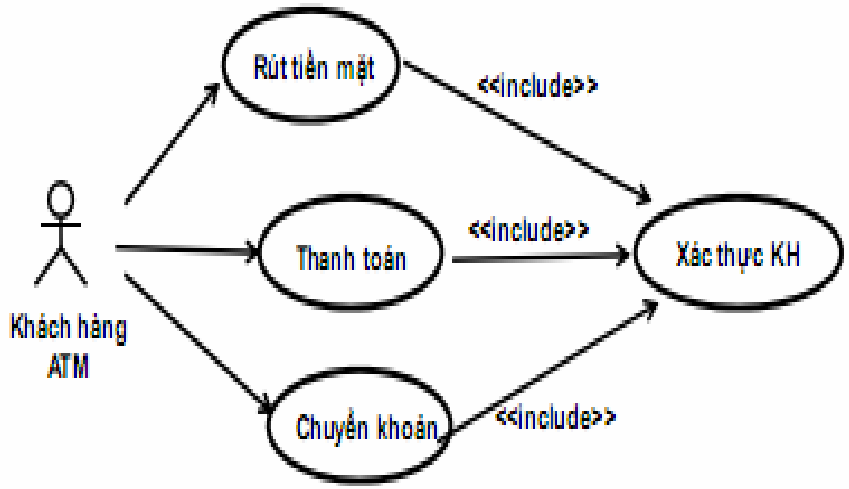


Các quan hệ

- Quan hệ gộp (Includes)
  - Trước phiên bản UML 1.3 quan hệ <<includes>> có tên là <<uses>>
  - Cho phép một UC sử dụng chức năng của UC khác
  - Chức năng của UC include sẽ được gọi trong UC cơ bản.



- Ví dụ, UC rút tiền



UC rút tiền

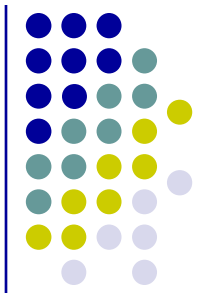
1. Gọi UC xác thực KH
2. Hiển thị menu
3. KH chọn chức năng rút tiền

UC xác thực KH

1. Đưa thẻ vào máy
2. Kiểm tra thẻ
3. KH nhập pin
4. Hệ thống kiểm tra pin

   E1: Thẻ sai.

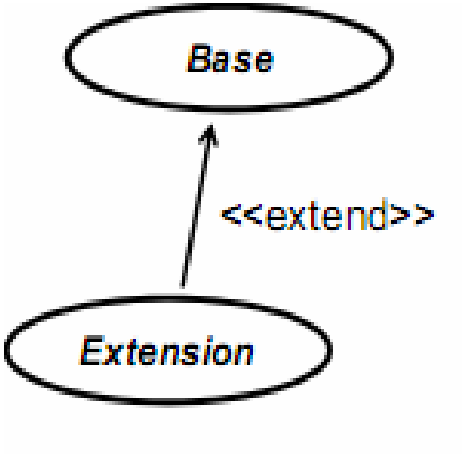
   E2: sai pin

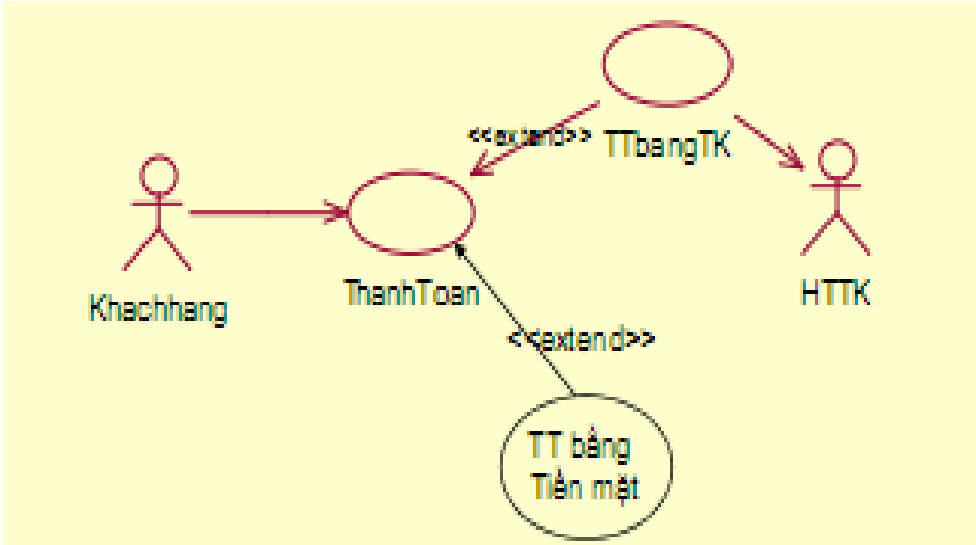


- Khi nào dùng quan hệ include
  - Tách ra hành vi(chức năng) chung của 2 hoặc nhiều UC.
    - Tránh mô tả hành vi đó nhiều lần trong các UC.
    - Đảm bảo những hành vi chung đó được thống nhất.
  - Tách ra hành vi của UC cơ sở nên được đóng gói riêng.
    - Tách ra hành vi không phải là chính của UC đó (hành vi ít quan trọng)
    - Giảm thiểu sự phức tạp của luồng sự kiện

Các quan hệ

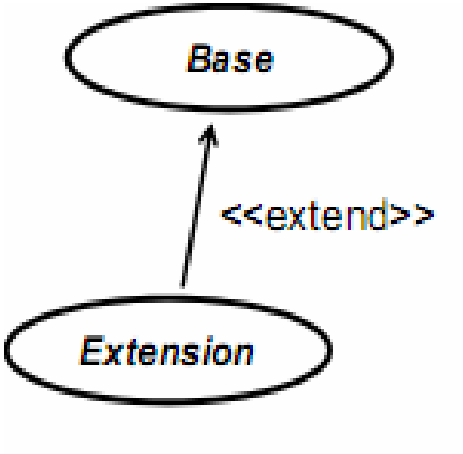
- Quan hệ mở rộng (Extends)
  - Cho phép mở rộng chức năng của một UC
  - Chèn hành vi của UC extend vào UC cơ sở.
  - Chỉ chèn khi điều kiện extend đúng
  - Chèn vào lớp cơ sở tại điểm phát sinh

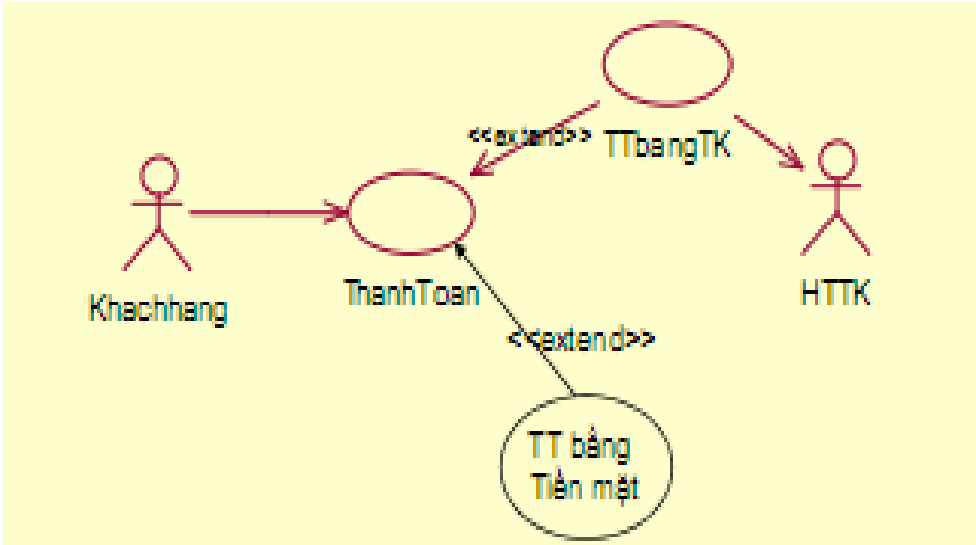




Các quan hệ

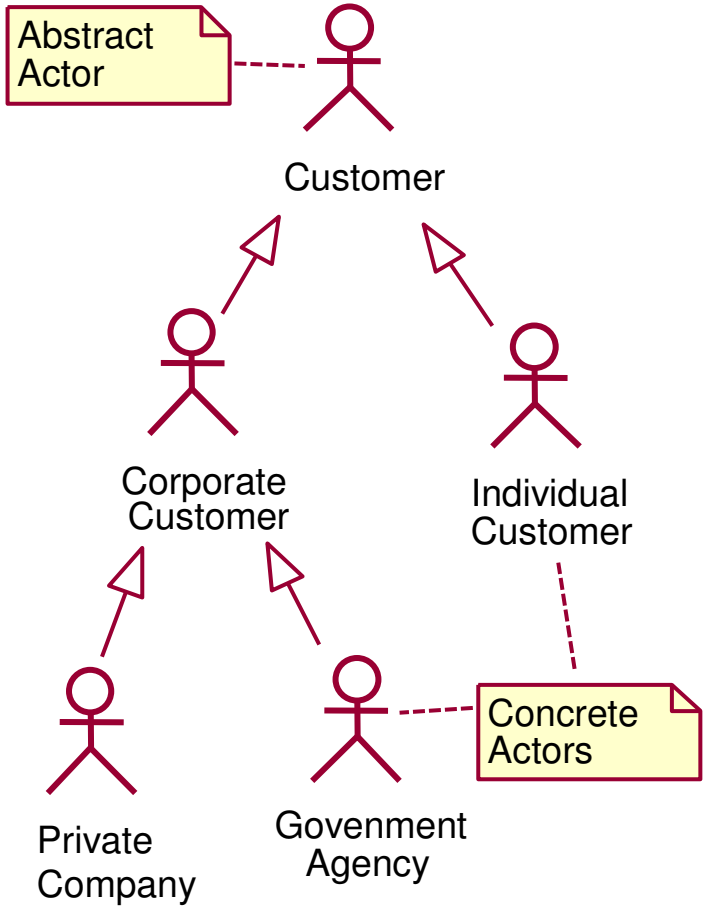
- Khi nào dùng quan hệ mở rộng (Extends)
  - Tách ra hành vi ngoại lệ, đặc biệt hoặc không bắt buộc
    - Chỉ được thực thi trong điều kiện cụ thể
    - Tách ra để làm đơn giản luồng chính
  - Thêm một hành vi mở rộng đối với UC cơ sở.
    - Phát triển hành vi đó độc lập



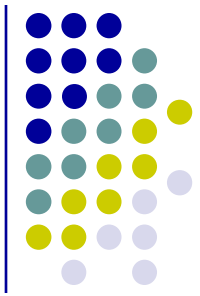


Các quan hệ

- Quan hệ tổng quát hóa (Generalization)
  - Chỉ ra một vài tác nhân hay UC có một số cái chung, giống nhau
  - Không nhất thiết hình thành quan hệ này cho các tác nhân
    - Khi một loại tác nhân kích hoạt một hay vài UC mà loại tác tác nhân khác không kích hoạt -> nên hình thành quan hệ khái quát hóa
    - Khi cả hai loại tác nhân cùng sử dụng các UC -> không cần mô hình hóa quan hệ khái quát hóa

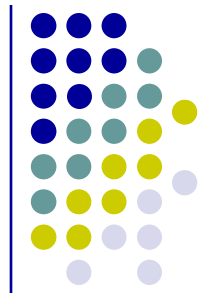


Tạo các gói



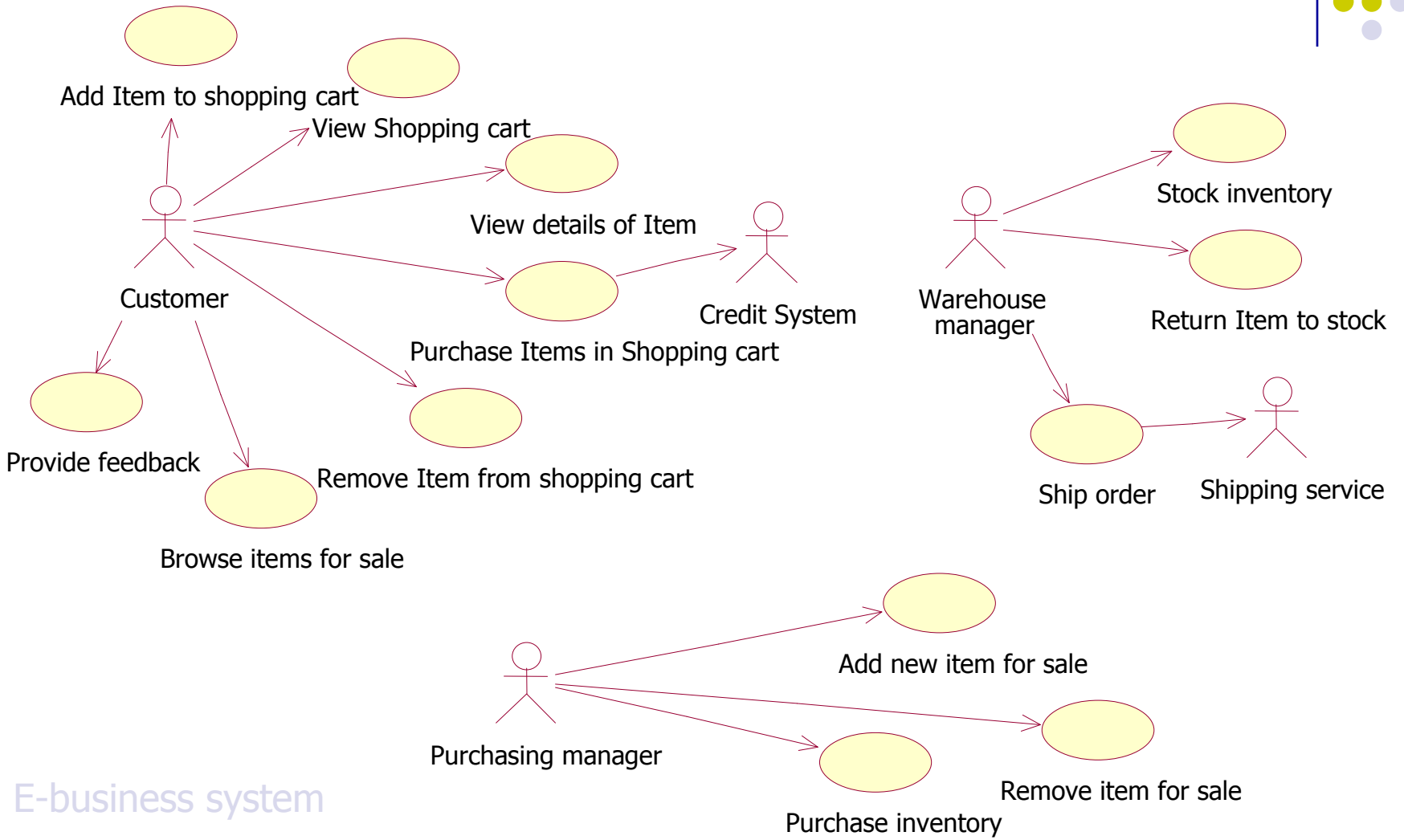
- Có thể nhóm các thành phần thành một nhóm chung
- Nếu số lượng UC quá lớn có thể chia chúng vào các nhóm
  - Dễ hiểu mô hình tổng thể hơn
  - Dễ bảo trì mô hình UC
  - Dễ giao việc cho các thành viên
- Xem xét khả năng gộp nhóm
  - Tương tác với cùng một tác nhân
  - Nhóm UC hợp thành một module tương đối hoàn thiện

Biểu đồ Use Case

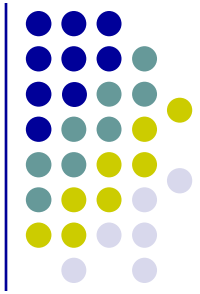


- Mô hình UC được mô tả bởi một hay nhiều biểu đồ UC
  - Số lượng biểu đồ UC cho một dự án là tùy ý
    - Không quá nhiều làm rối loạn
    - Phải đảm bảo đầy đủ để biểu diễn đầy đủ thông tin của hệ thống
- Nó là công cụ mạnh giúp thu thập yêu cầu chức năng hệ thống
- Nó chỉ ra quan hệ giữa UC và tác nhân và giữa UC với nhau
- Sử dụng biểu đồ để làm tài liệu UC, tác nhân và các quan hệ giữa chúng
- Lợi ích chính của biểu đồ UC là làm giao tiếp
  - Khi quan sát các UC, customer biết hệ thống có các chức năng nào
  - Khi quan sát các tác nhân, customer biết ai giao tiếp với hệ thống
  - Khi quan sát cả UC và tác nhân, customer biết phạm vi dự án

Thí dụ biểu đồ Use Case

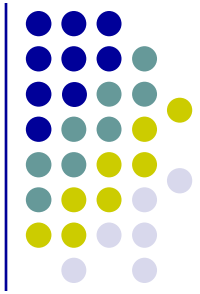


Biểu đồ Use Case



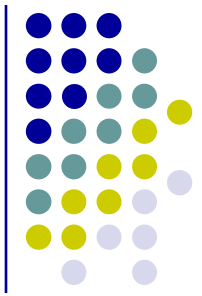
- Các chú ý khi xây dựng biểu đồ UC
  - Không nên mô hình hóa quan hệ kết hợp giữa tác nhân với tác nhân -> vì giao tiếp giữa các tác nhân là ở bên ngoài hệ thống
    - Hãy sử dụng biểu đồ luồng công việc để khảo sát quan hệ giữa các tác nhân
  - Không hình thành quan hệ Association giữa các UC
    - 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

Luồng sự kiện trong UC



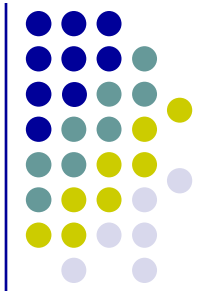
- Tài liệu luồng sự kiện (flow of events) mô tả hành vi của UC
  - mô tả luồng logíc đi qua UC
  - mô tả người sử dụng làm gì, hệ thống làm gì
  - Trong một UC có nhiều luồng sự kiện: luồng chính, luồng phụ
- Kịch bản (Scenario)
  - Một luồng sự kiện trong một hiện thực của UC
  - Là trình tự hành động cụ thể để mô tả hành vi
  - Kịch bản đi xuyên suốt UC theo nhánh chính, nhánh phụ, nhánh đặc biệt

Tài liệu luồng sự kiện



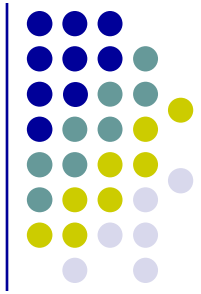
- Tài liệu luồng sự kiện bao gồm
  - Mô tả vắn tắt UC
    - Mô tả ngắn gọn UC làm gì?
    - Những ai sử dụng UC?
    - Nó cho lại kết quả gì?
  - Tiền điều kiện (pre-condition)
    - Điều kiện cần thực hiện trước khi UC khởi động
    - Không phải UC nào cũng có tiền điều kiện
  - Luồng sự kiện chính và luồng sự kiện rẽ nhánh
  - Hậu điều kiện (post-condition)

Tài liệu luồng sự kiện



- Tài liệu luồng sự kiện bao gồm
  - Mô tả vắn tắt UC
  - Tiền điều kiện (pre-condition): khi nào UC được kích hoạt?
  - Luồng sự kiện chính và luồng sự kiện rẽ nhánh
    - chi tiết về UC được mô tả trong hai luồng sự kiện này
    - mô tả cái gì sẽ xảy ra để thực hiện chức năng của UC
    - Nội dung tài liệu
      - UC khởi động như thế nào?
      - Các đường đi xuyên qua các UC
      - Luồng chính thông qua UC
      - Luồng rẽ nhánh thông qua UC
      - Các luồng lỗi
      - UC kết thúc thế nào.
  - Hậu điều kiện (post-condition)
    - Là điều kiện được thực hiện ngay sau khi kết thúc UC

Thí dụ tài liệu luồng sự kiện



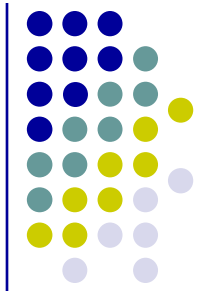
- Ca sử dụng: Cập nhật từ điển môn học
- Tác nhân: Nhân viên phòng đào tạo
- Mục đích: Cập nhật các môn học trong từ điển môn học
- Mô tả: Sau khi đăng nhập vào hệ thống, nhân viên phòng đào tạo cập nhật thông tin môn học vào biểu mẫu hoặc sửa môn học có sẵn và ghi lại.
- Luồng sự kiện chính:

Hành động tác nhân	Phản ứng hệ thống	Dữ liệu liên quan
1. Chọn chức năng cập nhật từ điển môn học	2. Hiển thị biểu mẫu để nhập thông tin về các môn học mới và hiển thị danh sách môn học để chọn môn học tiên quyết cho môn học.	Từ điển môn học
3. Nhập thông tin môn học, chọn môn học tiên quyết, đồng ý nhập mới.	4. Cập nhật môn học vào từ điển môn học	Môn học

Luồng sự kiện phụ: sửa thông tin

Ngoại lệ: Bước 4: nếu thông tin nhập ko chính xác thì yêu cầu nhập lại hoặc dừng ca sử dụng.

Thí dụ tài liệu luồng sự kiện



- Ca sử dụng: Sửa đổi thông tin môn học
- Tác nhân: Nhân viên phòng đào tạo
- Mục đích: Sửa các thông tin về một môn học đang tồn tại trong hệ thống
- Mô tả khái quát: Tìm đến môn học cần sửa đổi, xóa các thông tin cũ và nhập các thông tin mới về môn học này. Cuối cùng, yêu cầu hệ thống ghi nhận các thông tin mới.

Thí dụ tài liệu luồng sự kiện

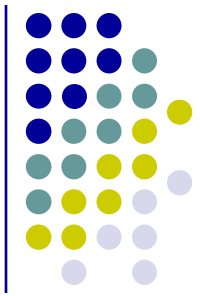
Hành động của tác nhân	Hồi đáp của hệ thống
1. Yêu cầu sửa thông tin môn học	**2.** Hiện form nhập điều kiện tìm kiếm.
3. Nhập thông tin về môn học cần sửa, yêu cầu tìm	**4.** Tìm kiếm và hiển thị danh sách môn học tìm được.
5. Chọn môn học cần sửa trong danh sách kết quả tìm kiếm	**6.** Hiển thị thông tin về môn học đã chọn
7. Tiến hành sửa, yêu cầu ghi lại	**8.** Kiểm tra, ghi lại thông tin mới và thông báo kết quả ghi nhận.

- Ngoại lệ:

  - Bước 4: Không có môn học nào thỏa mãn điều kiện tìm kiếm thì thông báo không tìm được và yêu cầu tìm lại hoặc dừng.

  - Bước 8: Nếu thông tin sửa không chính xác thì yêu cầu sửa lại hoặc dừng ca sử dụng.

Tóm tắt



- Bài này đã xem xét các vấn đề sau
  - Biểu đồ UC là gì?
  - Quan hệ giữa biểu đồ UC và biểu đồ nghiệp vụ
  - Các khái niệm của mô hình UC
  - Cách tìm kiếm UC, tác nhân, quan hệ trong mô hình UC
  - Các phần tử đồ họa xây dựng biểu đồ UC
  - Cách mô tả luồng sự kiện
    - văn bản
    - biểu đồ hoạt động