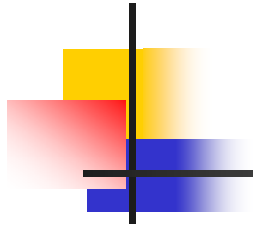




## Chương 3

---

# **Biểu đồ use case, vấn đề nghiệp vụ**

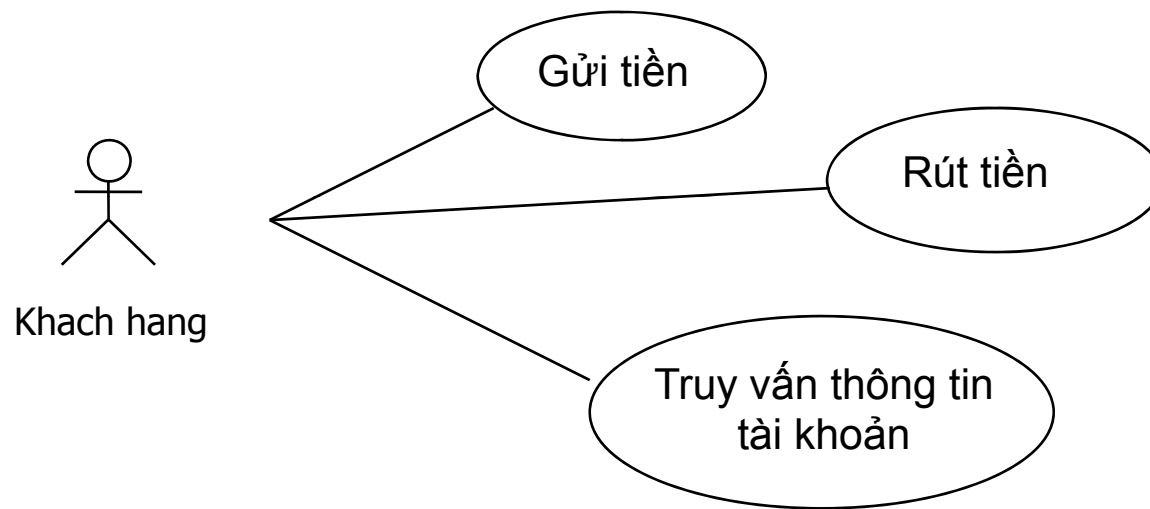


# Nội dung

---

- 3.1 Biểu đồ use-case (tình huống sử dụng)
- 3.2 Vấn đề nghiệp vụ (business)

## 3.1 Biểu đồ use case (hệ thống)



- Actor
- Use case
- Association (liên kết)



## Biểu đồ use case

---

- Được dùng cho phân tích, thiết kế (ng nghiệp vụ (business), hệ thống cũ, hệ thống mới)
- Mỗi use case sẽ thể hiện một tình huống sử dụng hệ thống khác nhau của người dùng.
- Tập hợp các use case và các tác nhân (actor) cùng với quan hệ kết hợp (association) giữa chúng sẽ cho ta 1 biểu đồ use case dùng để mô tả yêu cầu hệ thống (mô hình nghiệp vụ trình bày sau)



## Mục đích

---

- Cung cấp cái nhìn tổng thể trực quan về những gì hệ thống sẽ làm và ai sẽ dùng nó
- Cơ sở để xác định giao tiếp người và máy
- Dùng làm cơ sở cho những thiết kế kiểm thử hệ thống
- Biểu đồ use-case là mô hình hệ thống theo hướng nhìn chức năng người dùng cuối



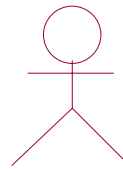
# **Xây dựng biểu đồ use case**

---

1. Xác định tác nhân (Actor)
2. Xác định Use case
3. Xây dựng biểu đồ Use case
4. Chia các biểu đồ use case thành các package

# 1. Tác nhân - actor

- Một tác nhân là một đối tượng bên ngoài hệ thống giao tiếp với hệ thống theo 1 trong các hình thức sau:
  - Tương tác, trao đổi thông tin
  - Cung cấp thông tin đầu vào hoặc nhận thông tin đầu ra
- Tác nhân không phải là điều khiển hoạt động của hệ thống
- Ký hiệu:



Khách hàng

- Tên: danh từ
- Tác nhân và Vai trò (role)



## Ba loại tác nhân

---

- Người dùng của hệ thống
- Những hệ thống khác tương tác với hệ thống đang xét
- Thời gian (Time)





## Tác nhân chính và tác nhân hỗ trợ

---

- Tác nhân chính (primary actor): Ai đang sử dụng hệ thống? Ai cung cấp thông tin cho hệ thống? Hệ thống cung cấp thông tin cho ai?
- Tác nhân hỗ trợ (secondary actor): những nhóm đối tượng nào hệ thống cần để thực hiện hoạt động của nó (vd: quản trị, backup,...)



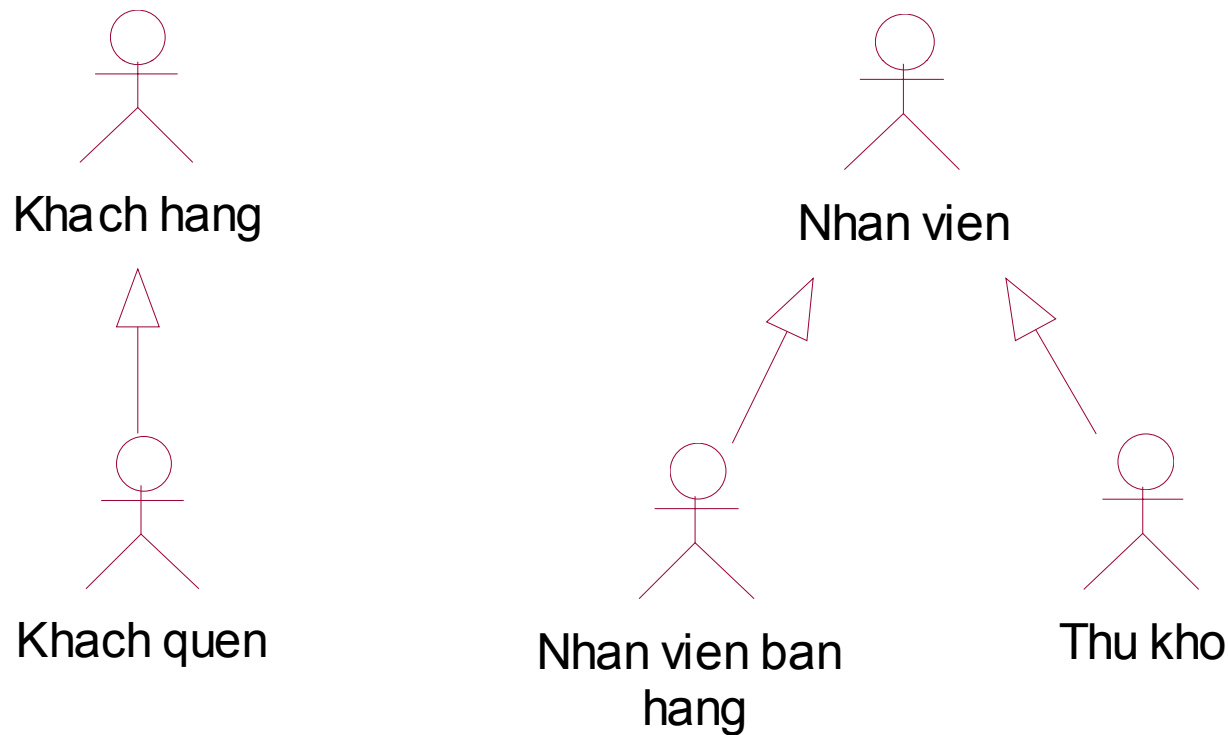
# Tác nhân – những câu hỏi

---

- Ai sẽ sử dụng những chức năng chính của hệ thống (tác nhân chính)?
- Ai sẽ cần sự hỗ trợ của hệ thống để thực hiện những tác vụ hàng ngày của họ?
- Ai sẽ cần bảo trì, quản trị và đảm bảo cho hệ thống hoạt động (tác nhân phụ)?
- Hệ thống cần phải tương tác với các hệ thống khác nào?
  - Hệ thống kích hoạt mối quan hệ với hệ thống chúng ta
  - Hệ thống mà hệ thống chúng ta sẽ thiết lập quan hệ
  - Hệ thống máy tính khác
  - Các ứng dụng khác trong chính chiếc máy tính mà hệ thống này sẽ hoạt động
- Ai hay cái gì quan tâm đến kết quả (giá trị) mà hệ thống sẽ sản sinh ra?

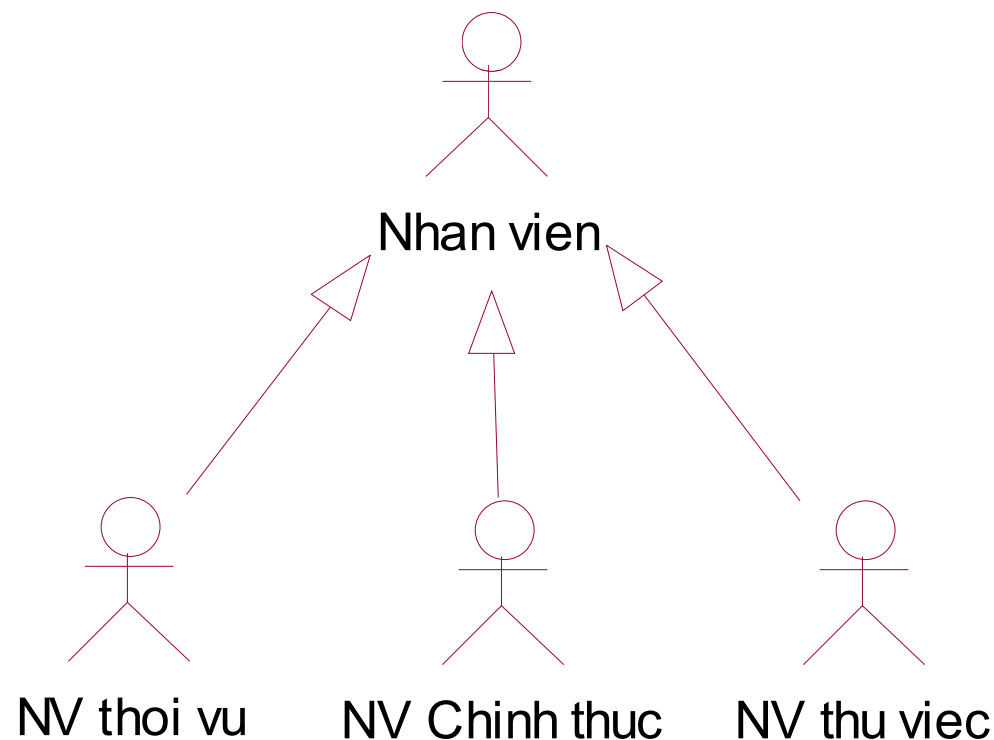
# Quan hệ giữa các tác nhân

- Tổng quát hoá (generalization) và chuyên biệt hoá (specialization)



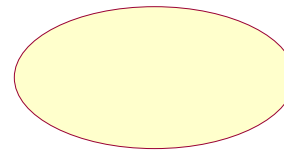
# Tác nhân trừu tượng (Abstract actor)

- Là actor mà không có 1 thể hiện (instance) nào



## 2. Use case

- Use case mô tả một chuỗi các hoạt động mà hệ thống sẽ thực hiện để đạt được một kết quả có ý nghĩa đối với một tác nhân. Use case được xem như 1 chức năng của hệ thống theo quan điểm của người dùng
- Mỗi chức năng sẽ được biểu diễn bởi 1 use case (có thể phân rã)
- Biểu diễn:



Dat hang

- Tên: động từ + danh từ



## ...Use case

---

- Không phụ thuộc vào việc thực thi hệ thống
  - Tập trung vào What
  - Không quan tâm How
- Mức trừu tượng cao
  - Dễ xem
  - Biểu diễn số use case hợp lý
- Mỗi use case:
  - Nên biểu diễn một giao dịch (transaction) hoàn chỉnh giữa user và hệ thống
  - Nên đặt tên theo thuật ngữ của user không nên dùng từ ngữ kỹ thuật

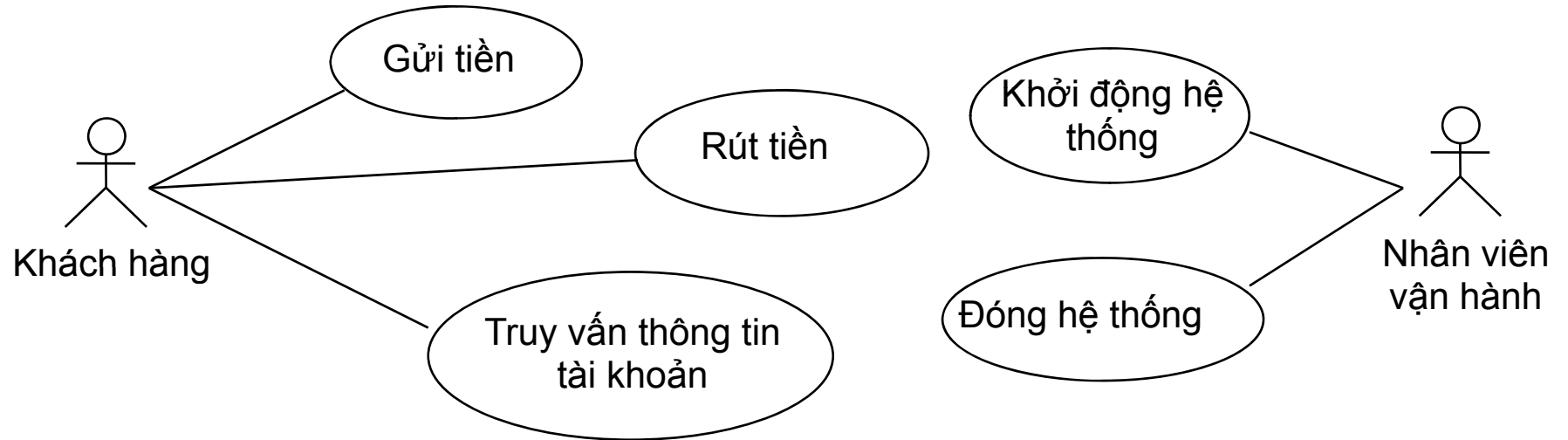


## Xác định use case

---

- Xác định actor hay use case trước hay sau không quan trọng. Hai bước này có thể tiến hành cùng lúc
- Hệ thống sẽ phải làm gì để cung cấp giá trị cho bên ngoài hệ thống?
- Xác định use case thông qua tác nhân
  - Tìm các nhiệm vụ và chức năng mà tác nhân sẽ thi hành hoặc hệ thống cần tác nhân để thi hành và mô hình hoá nó như là use case

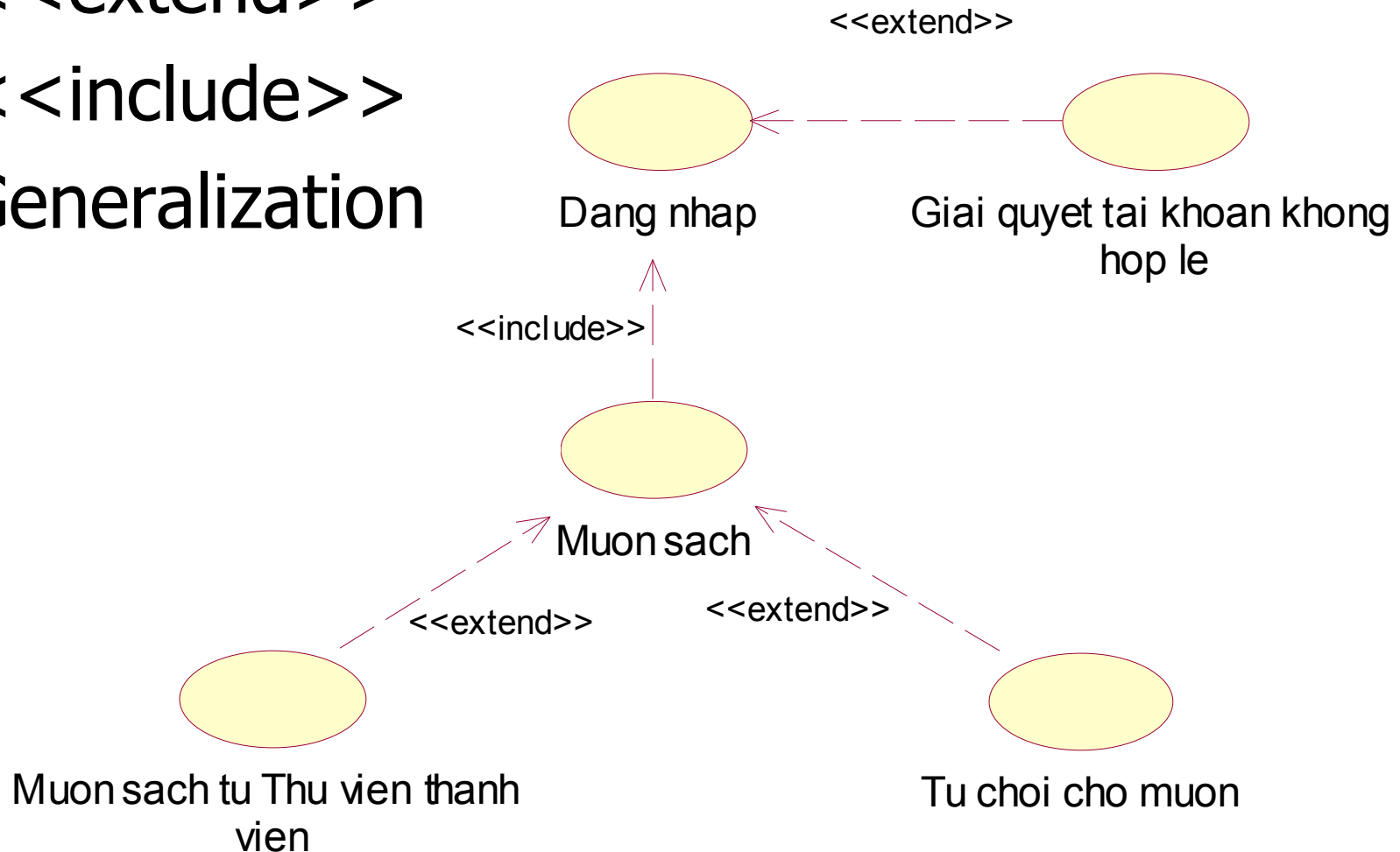
# Use case (ATM)





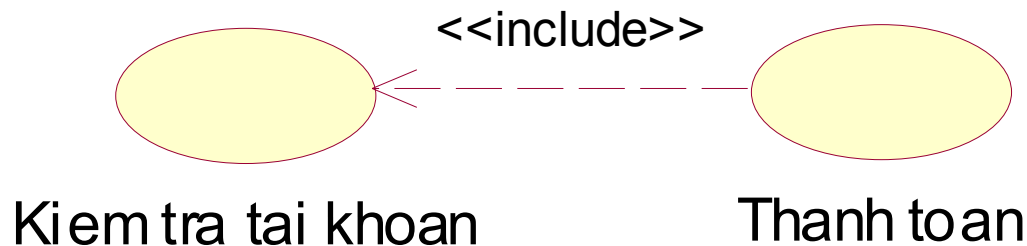
# Quan hệ giữa các use case

- <<extend>>
- <<include>>
- Generalization



# Use case trừu tượng (Abstract Use Case)

- Là một use case mà không được bắt đầu trực tiếp bởi 1 actor.
- Là use case chỉ tham gia vào lược đồ chỉ với các quan hệ include hay extend
- Use case trừu tượng được dùng để cung cấp 1 số chức năng phụ để dùng cho các use case khác.





## 3. Xây dựng biểu đồ use case

---

- Tạo biểu đồ use case ban đầu
- Tinh chế và nâng cấp sơ đồ use case
  - Bổ sung các use case mô tả chức năng đặc trưng phần mềm
  - Tách những dòng xử lý đặc biệt → use case <<extend>>
  - Tổng quát hoá các use case có dòng xử lý chung → use case tổng quát
  - Phát triển các liên kết <<include>>

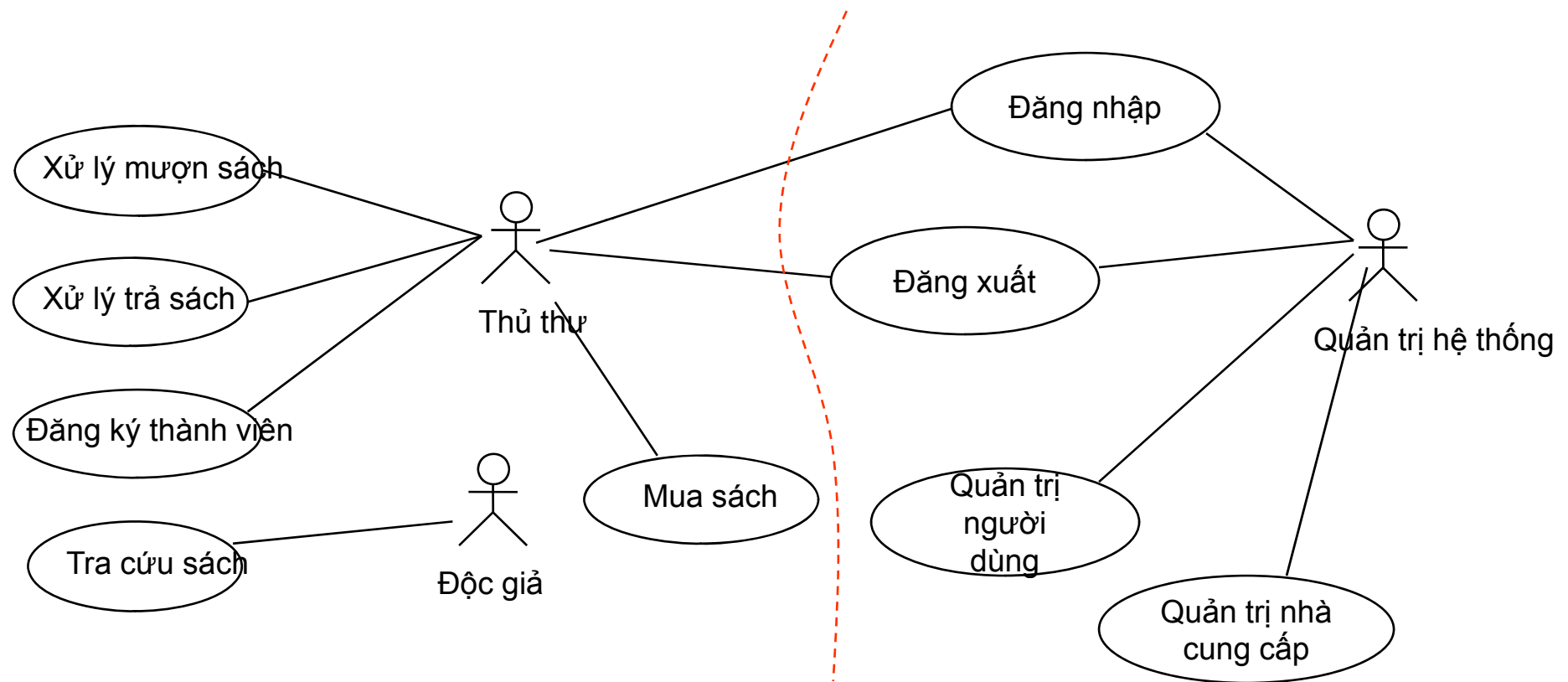


## Bổ sung các use case

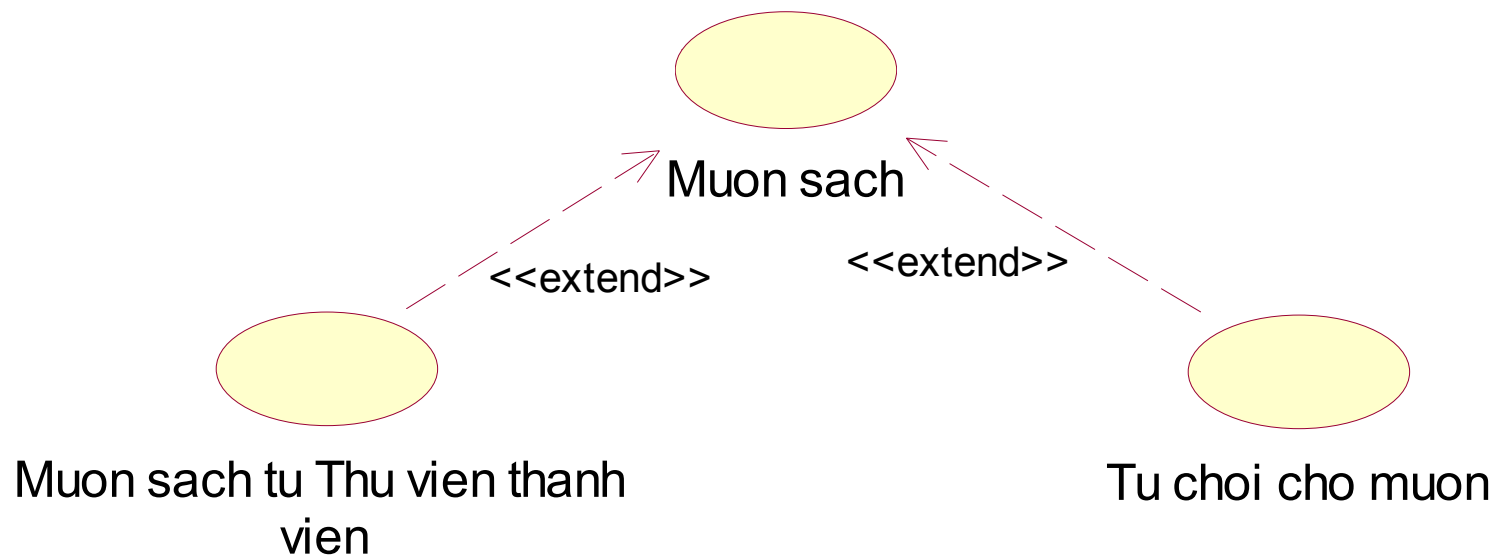
---

- Các use case quản trị hệ thống:
  - Quản trị người dùng
  - Quản trị thông số hệ thống
- Các use case quản trị dữ liệu
  - Quản trị danh mục dữ liệu
  - Quản trị an toàn (backup/restore)
- ...

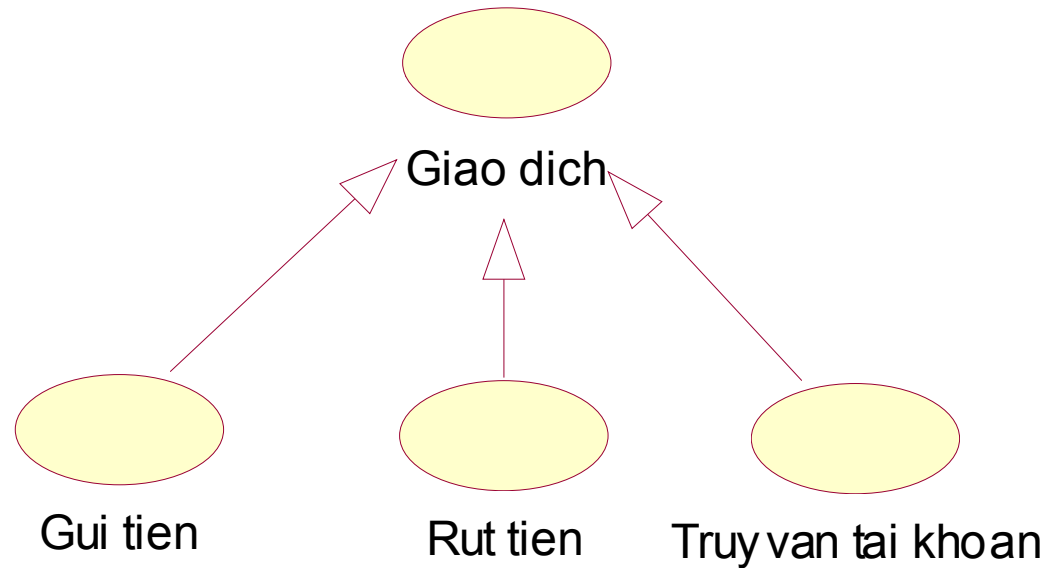
# Bổ sung (thư viện)



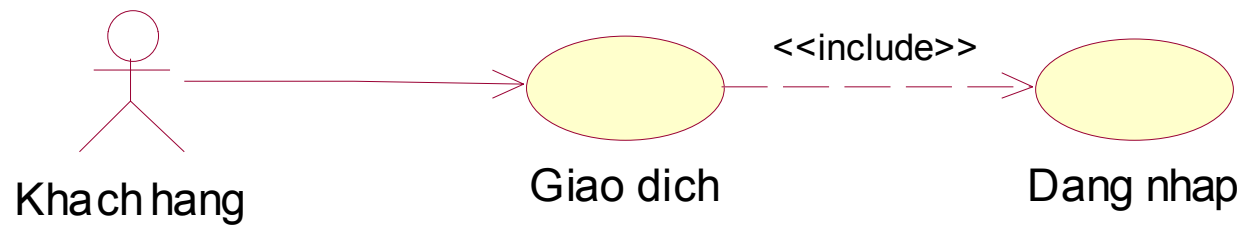
# Tách những dòng xử lý đặc biệt



# Tổng quát hoá các use case

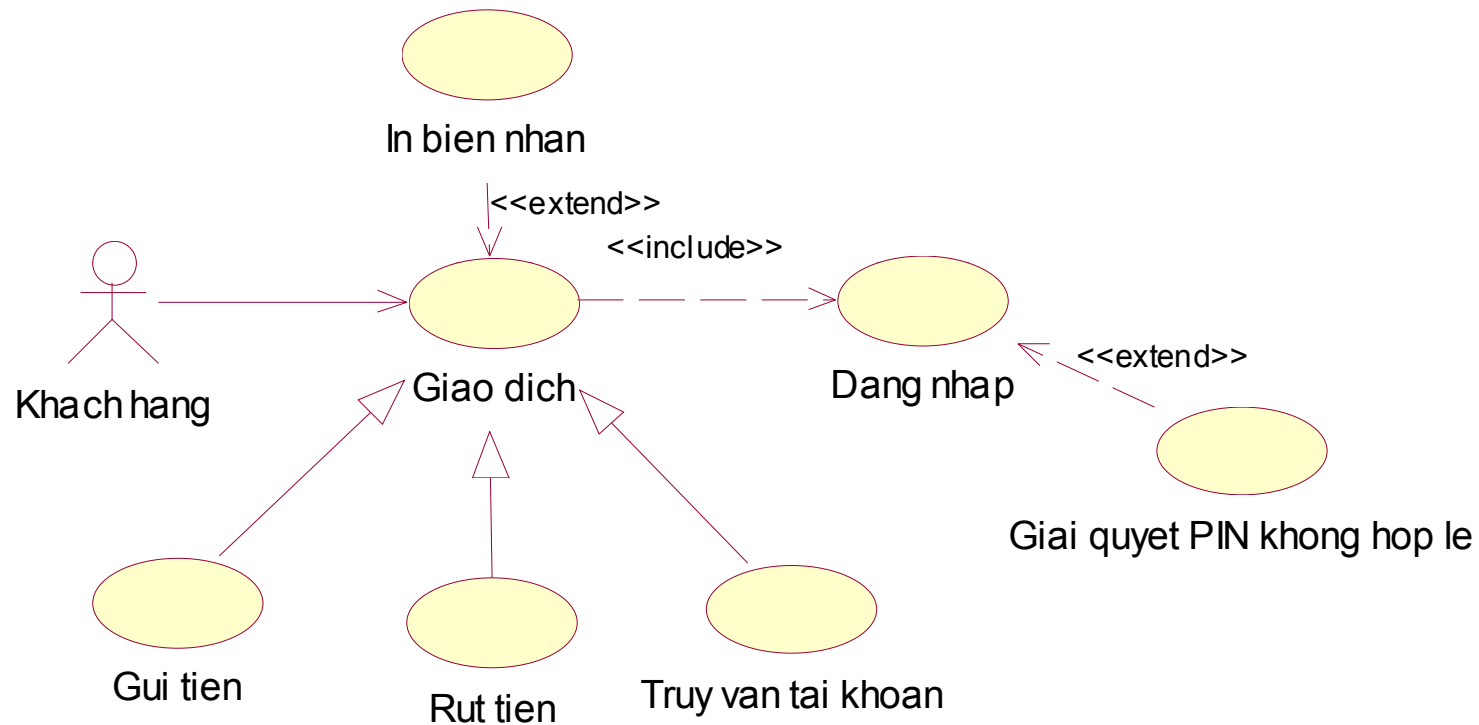


# Phát triển các liên kết <<include>>

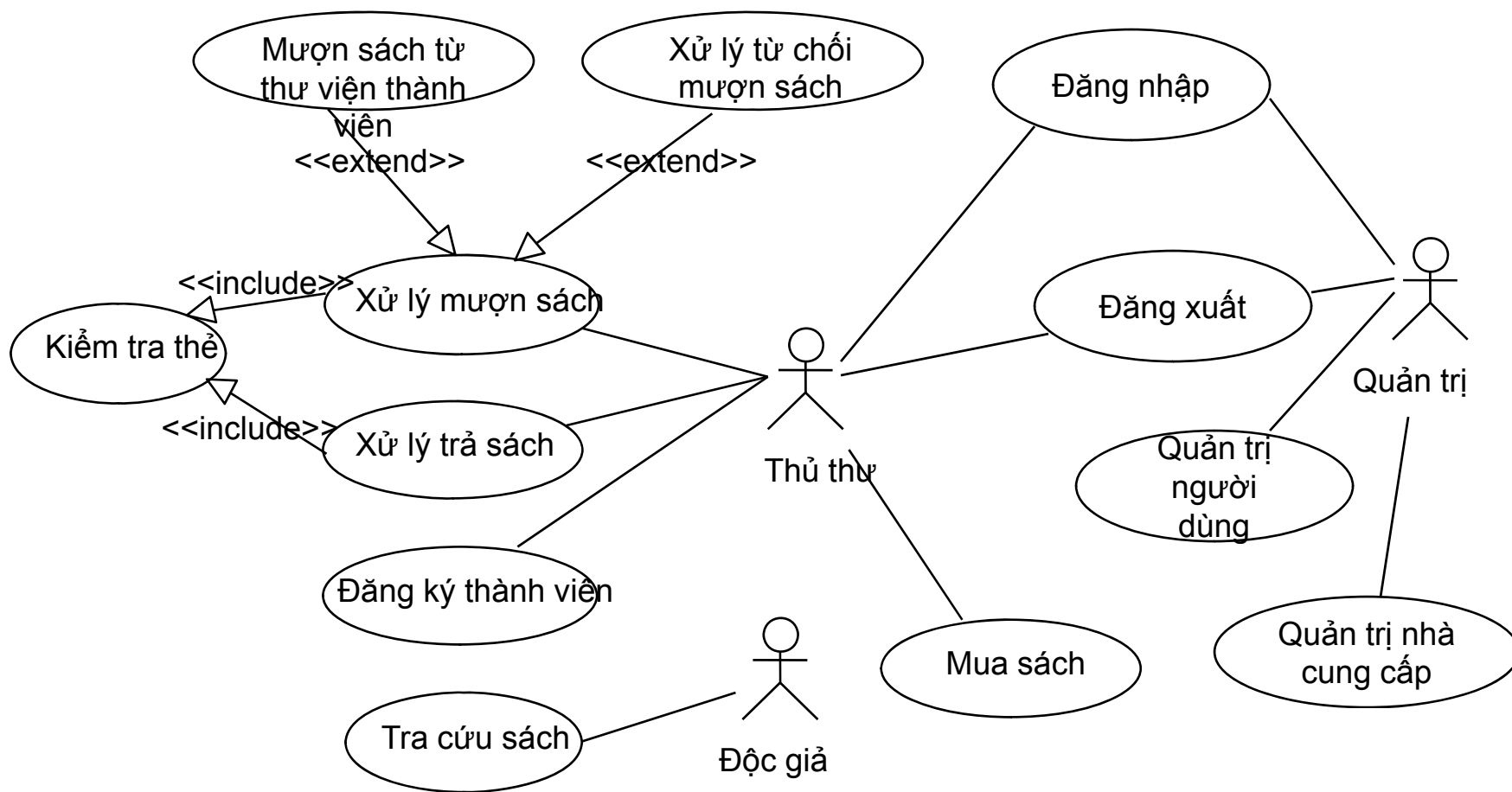




# Biểu đồ use case (ATM)



# Biểu đồ use case (thư viện)





# Mô tả use case

**Tóm tắt:** tóm tắt nội dung hoạt động của use case

**Dòng sự kiện chính:** các sự kiện và hoạt động xảy ra bình thường của use case

**Dòng sự kiện phụ:** các sự kiện và những hoạt động bất thường của use case ngoài những hoạt động chính

**Tiền điều kiện (pre-condition):** mô tả trạng thái của hệ thống phải đạt được để use case có thể bắt đầu

**Hậu điều kiện (post-condition):** liệt kê các trạng thái có thể của hệ thống tại cuối use case. Hệ thống phải thuộc một trong những trạng thái đó khi use case kết thúc



# Mô tả use case Giao dịch

Tóm tắt: use case bắt đầu khi có khách hàng đưa thẻ vào máy ATM. Hệ thống sẽ cho phép khách hàng thực hiện các giao dịch

Dòng sự kiện chính:

1. Xác nhận thẻ và kiểm tra đăng nhập của khách hàng (thực hiện bởi use case đăng nhập)
2. Hiển thị menu yêu cầu khách hàng chọn dịch vụ thực hiện
3. Khách hàng chọn dịch vụ thực hiện
4. Thi hành dịch vụ tương ứng khi khách hàng chọn (rút tiền, gửi tiền, xem thông tin tài khoản: được mô tả trong từng use case tương ứng)
5. Đóng giao diện thực đơn và thông báo khách hàng rút thẻ
6. Khách hàng rút thẻ và hệ thống trở về trạng thái sẵn sàng



# Mô tả use case Giao dịch

Dòng sự kiện phụ:

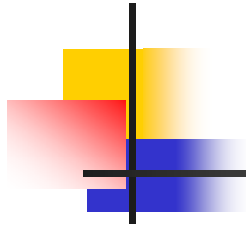
1. Xử lý thẻ không hợp lệ và đăng nhập không thành công

Tiền điều kiện:

1. ATM phải thông kết nối với mạng ngân hàng
2. ATM phải có đủ giấy để in hoá đơn

Hậu điều kiện:

1. ATM phải ở trạng thái sẵn sàng cho một khách hàng mới

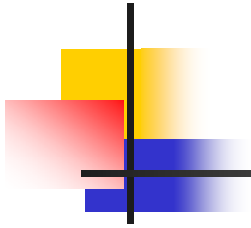


## ... Mô tả use case Rút tiền

Tóm tắt: use case bắt đầu khi khách hàng chọn thực hiện rút tiền từ menu chính. Hệ thống sẽ xử lý yêu cầu rút tiền của khách hàng

Dòng sự kiện chính:

1. Yêu cầu xác định tài khoản
2. Yêu cầu nhập số tiền cần rút
3. Khách hàng nhập số tiền
4. Kiểm tra số tiền nhập với số dư tài khoản
5. Chuyển tiền ra cho khách hàng
6. In hoá đơn rút tiền cho khách hàng



## ... Mô tả use case Rút tiền

Dòng sự kiện phụ:

1. Nếu số tiền vượt quá số dư thì thông báo cho khách hàng điều chỉnh hoặc huỷ bỏ giao dịch

Tiền điều kiện:

1. ATM phải có đủ tiền mặt trong máy
2. ATM phải có đủ giấy để in hoá đơn

Hậu điều kiện:

1. ATM phải ở trạng thái cho thực hiện một giao dịch mới



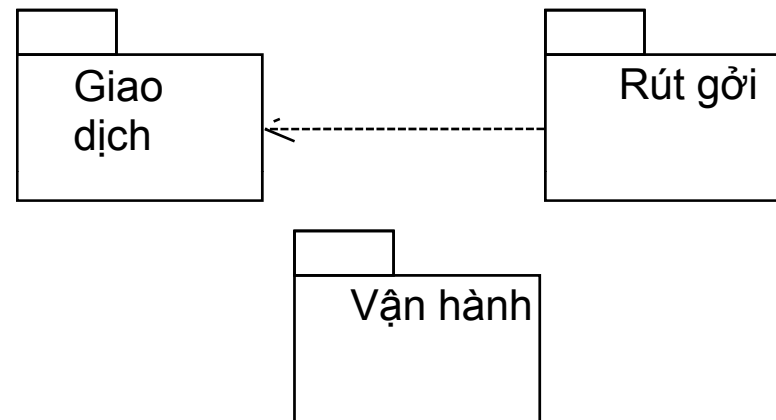
## 4. Phân chia sơ đồ use case thành gói

- Phân chia sơ đồ use case thành các gói nhằm:
  - Thu hẹp tiêu kích bản khi gặp các hệ thống lớn với sơ đồ use case phức tạp
  - Mỗi gói phản ánh một phạm vi của hệ thống mà chúng ta chỉ muốn quản lý nó khi chúng ta truy cập gói đó
- Mỗi gói phản ánh một chức năng chính của hệ thống
- Nên tạo 1 lược đồ chính chỉ chứa các package và nhóm các use case.

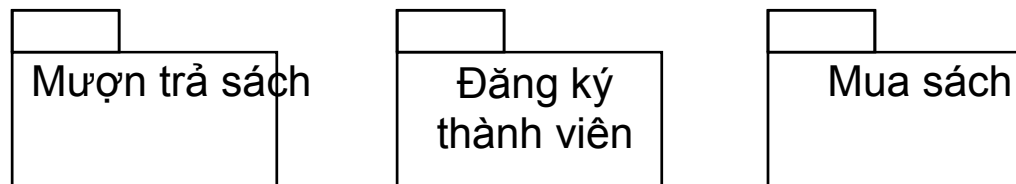


# Phân chia thành các gói

- Ví dụ: hệ thống ATM



- Hệ thống thư viện





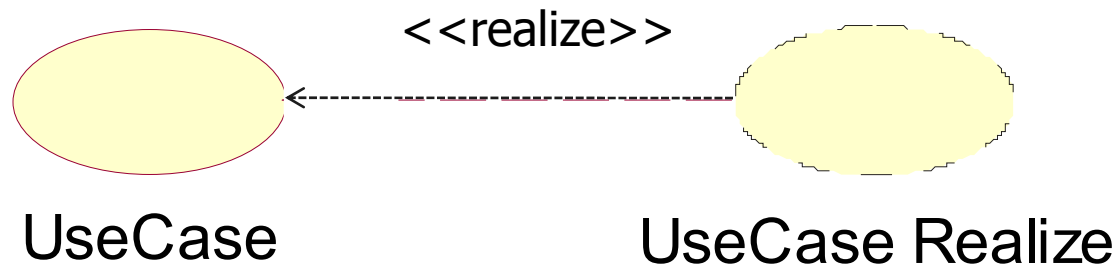
## Hiện thực use case (Use case Realization)

---

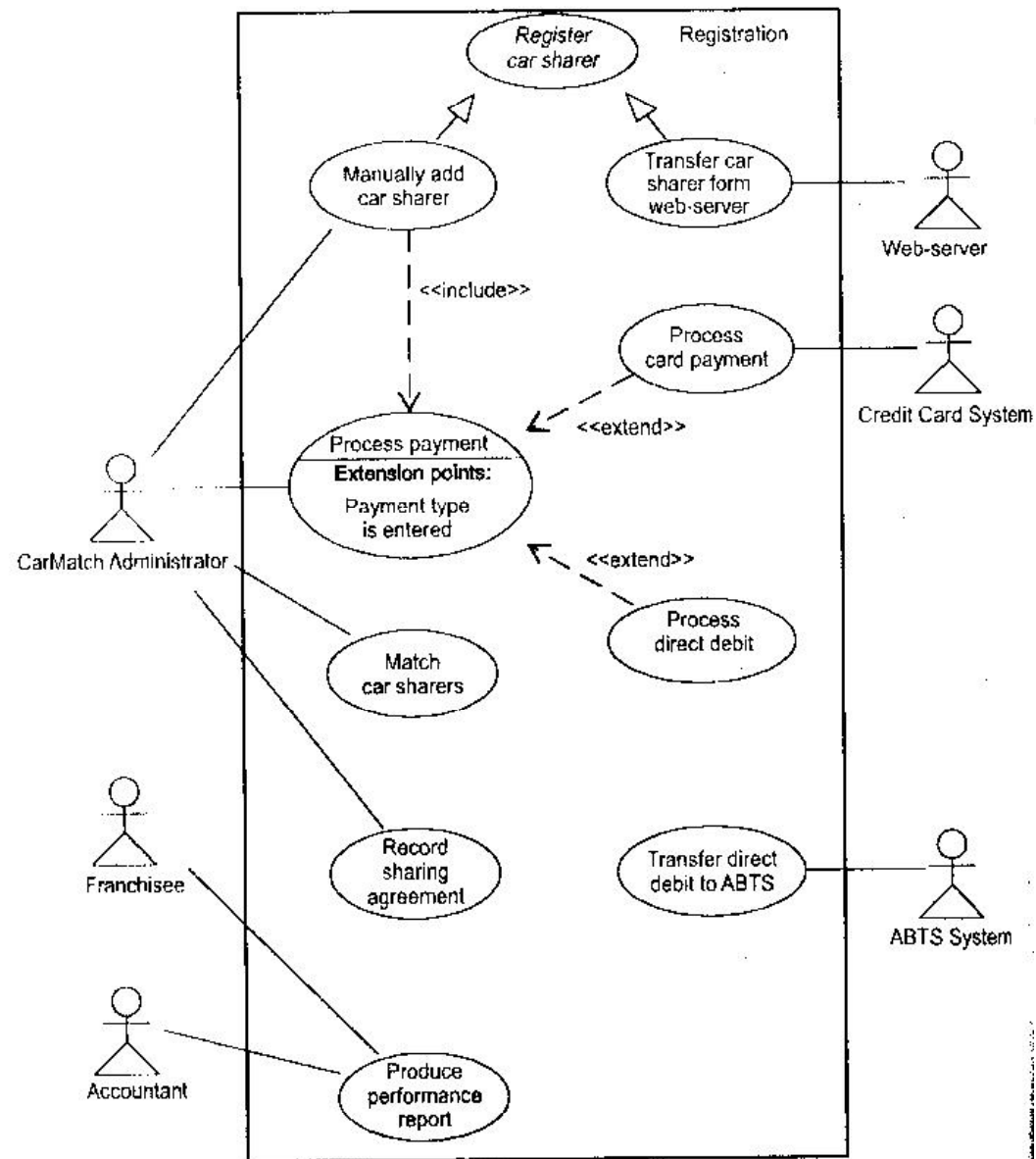
- Nó biểu diễn việc thiết kế cho use case.
- Cần tách riêng use case và use-case hiện thực nhờ đó có thể quản lý chúng 1 cách riêng biệt → có thể thay đổi use case thiết kế mà không ảnh hưởng đến use case gốc

# Biểu diễn

- 2 loại use case được kết nối với nhau bằng 1 quan hệ phụ thuộc (dependency) với phân loại là «realize».



Ví dụ 3.6



Hình 3.23: Phiên bản cấu trúc của biểu đồ use case



## 3.2 Vấn đề nghiệp vụ

---

- Mô hình hóa nghiệp vụ
- Phân tích qui trình nghiệp vụ
- Mô tả ràng buộc
- Xác định worker (thừa tác viên) và thực thể nghiệp vụ
- Mô hình nghiệp vụ và thiết kế hệ thống

# Mô hình hóa nghiệp vụ

- Thu thập yêu cầu?





# Yêu cầu chức năng

---

- Yêu cầu chức năng chỉ ra những gì hệ thống làm, chúng thường quan hệ với những nguồn đặc trưng, thường là các use-case hay những qui tắc nghiệp vụ (business rule)
- Một số yêu cầu chức năng
  - Chức năng tính toán
  - Chức năng lưu trữ
  - Chức năng tìm kiếm
  - Chức năng kết xuất
  - Chức năng backup, restore
  - Chức năng đa người dùng
  - Chức năng đa phương tiện...



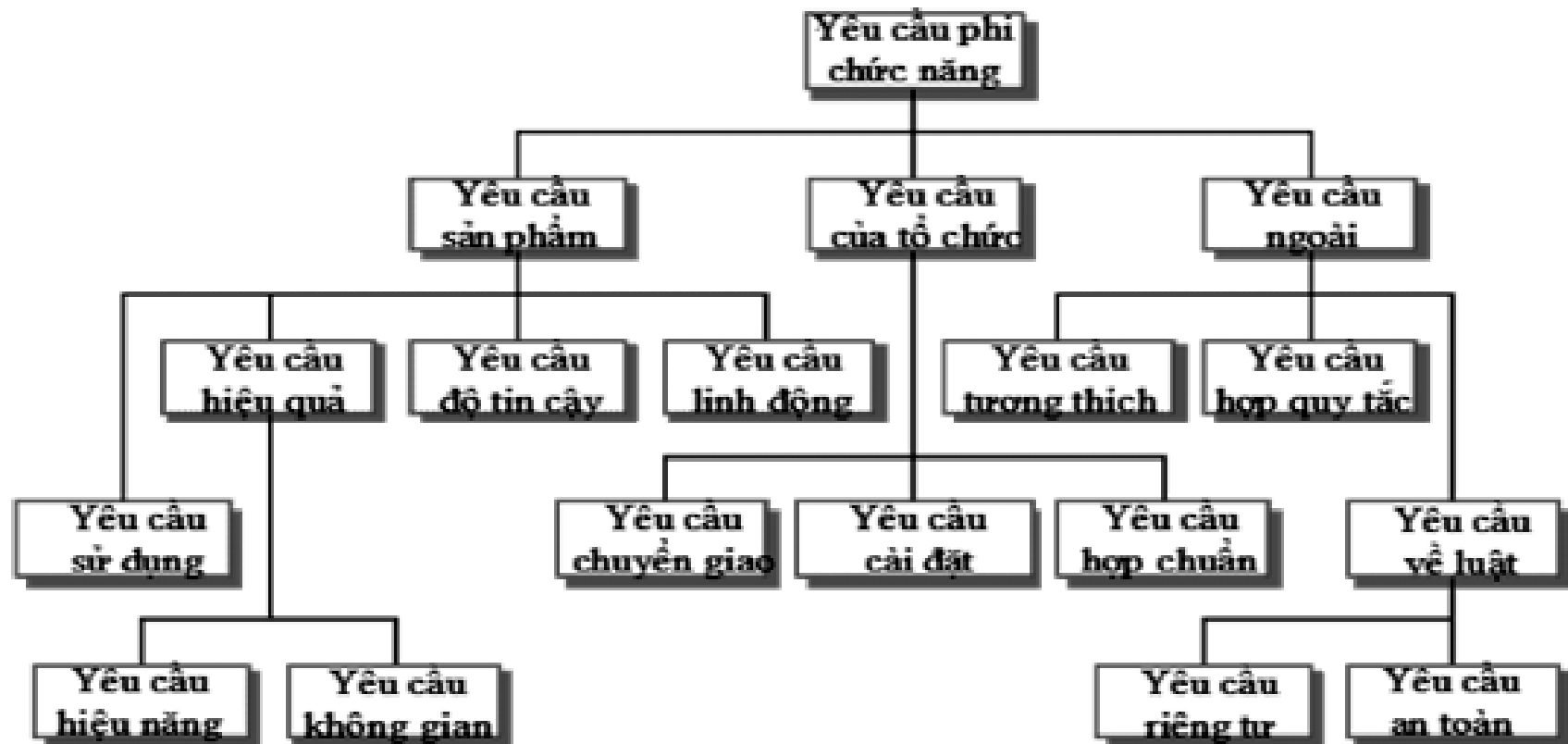
# Yêu cầu phi chức năng

---

- Một số yêu cầu phi chức năng
  - Độ tin cậy, thời gian đáp ứng, các yêu cầu về lưu trữ...
  - Các chuẩn được sử dụng, các công cụ CASE, ngôn ngữ lập trình...
  - Yêu cầu của người sử dụng: dễ sử dụng, thân thiện
  - Ràng buộc về ngân sách
  - Phù hợp với các chính sách của tổ chức sử dụng hệ thống
  - Yêu cầu tương thích giữa phần cứng và phần mềm
  - Các yêu cầu từ các tác nhân ngoài khác...



# Yêu cầu phi chức năng





# Mô hình hóa nghiệp

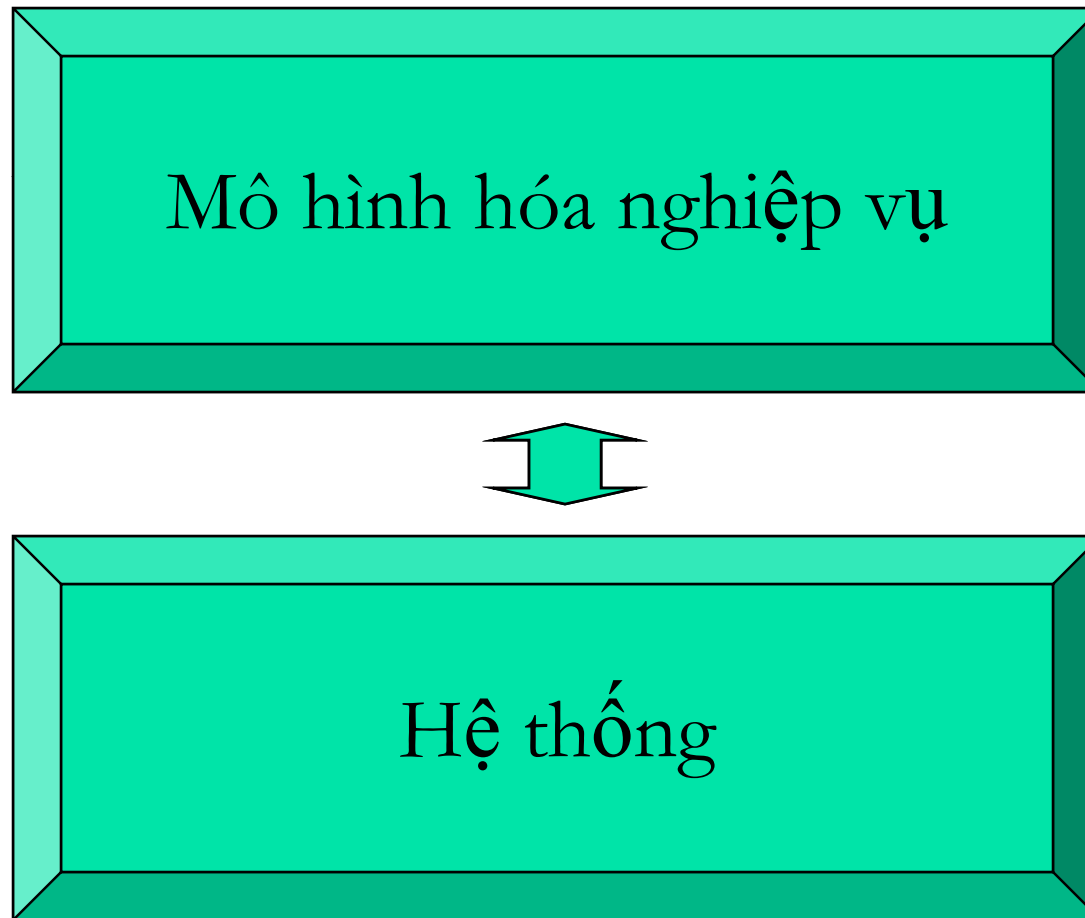
---

- Nghiệp vụ (business)
- Trực quan hóa những hệ thống phức tạp
  - Dễ giao tiếp, dễ truyền đạt
  - Giúp thực hiện những giải pháp của dễ dàng hơn. Chúng ta có thể so sánh và tối ưu hóa
- Nắm bắt được các yêu cầu nghiệp vụ
- Xác định được phạm vi hệ thống
- Biểu diễn sự thay đổi, cải tiến qui trình đã tồn tại, hoặc xây dựng qui trình mới, hoặc nâng cấp môi trường, ...



# Mô hình hóa nghiệp vụ

---





# Tại sao mô hình hóa nghiệp vụ?

---

- Nhằm đảm bảo những giải pháp, những hệ thống cần xây dựng đáp ứng thực sự nhu cầu khách hàng.
- Giảm thiểu rủi ro do những người phát triển không có thông tin đầy đủ về cách thức mà nghiệp vụ được thực hiện
- Xác định đúng vai trò trách nhiệm của con người cũng như xác định những gì được xử lý bởi nghiệp vụ trong việc phát triển hệ thống

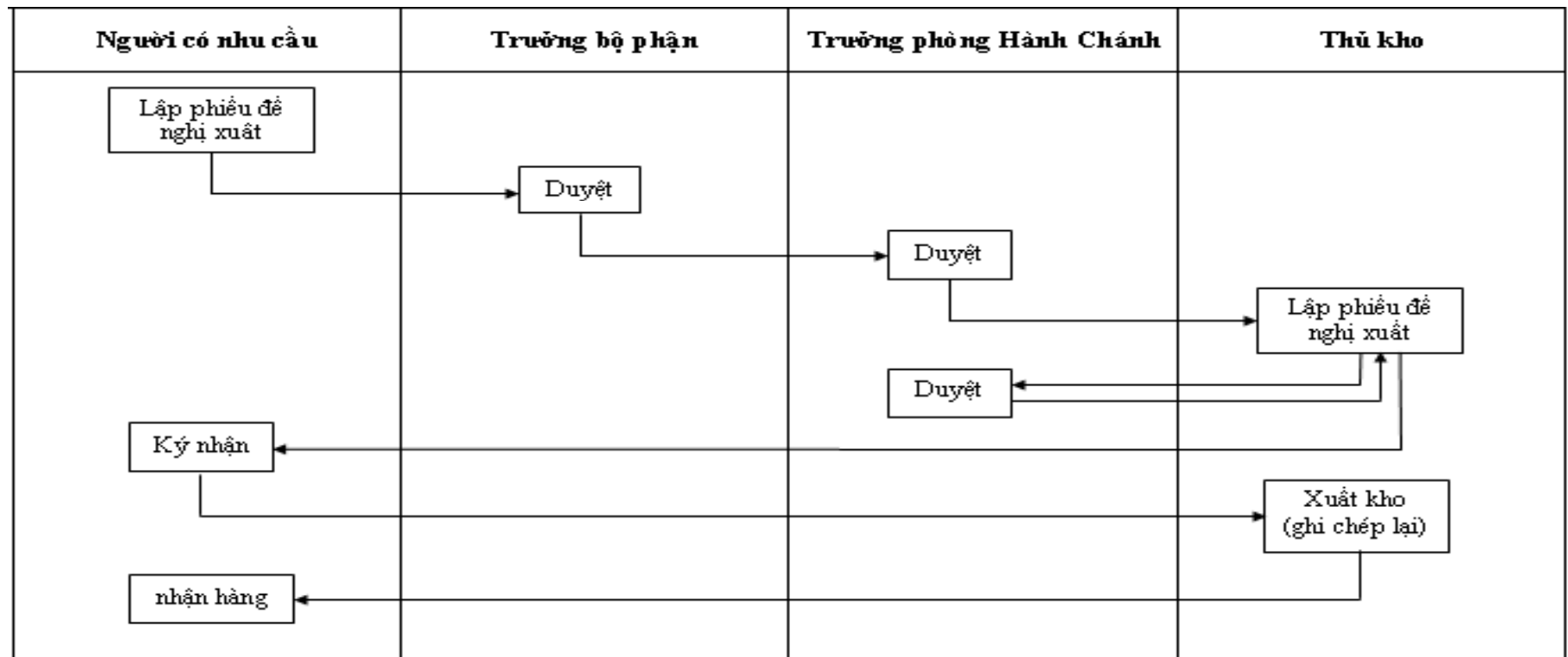


## Mục đích mô hình hóa nghiệp vụ?

---

- Hiểu được cấu trúc và các hoạt động của tổ chức được triển khai hệ thống.
- Hiểu được các vấn đề hiện tại trong tổ chức và xác định các vấn đề cần cải tiến.
- Bảo đảm rằng các khách hàng, người dùng cuối, và các nhà phát triển có sự hiểu biết chung về tổ chức.
- Thiết lập các yêu cầu hệ thống nhằm hỗ trợ cho tổ chức

# Mô hình yêu cầu cung cấp





## 3.2 Vấn đề nghiệp vụ

---

- Mô hình hóa nghiệp vụ
- **Phân tích qui trình nghiệp vụ**
- Mô tả ràng buộc
- Xác định worker (thừa tác viên) và thực thể nghiệp vụ
- Mô hình nghiệp vụ và thiết kế hệ thống



# Phân tích qui trình nghiệp vụ

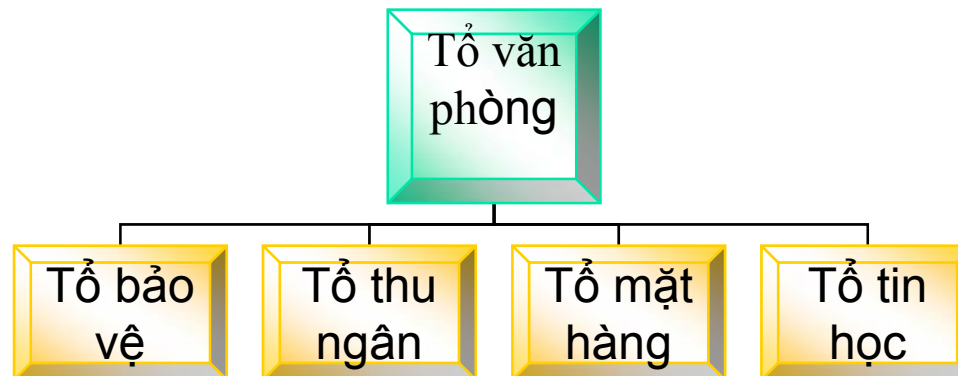
---

- Đánh giá và nắm bắt thông tin về tổ chức.
- Xác định các đối tượng liên quan (stakeholder) và người dùng cuối của hệ thống.
- Xác định phạm vi trong mô hình hóa nghiệp vụ.
- Đưa ra những mục tiêu chính của tổ chức.
- Xác định những khả năng cải tiến và các mục tiêu mới của tổ chức.



# Nắm bắt thông tin về tổ chức

- Cơ cấu tổ chức, phân cấp và các vai trò trong hệ thống.
- Mô tả ngắn gọn các thành phần và mối quan hệ này thông qua sơ đồ tổ chức



Sơ đồ tổ chức của siêu thị



# Mô tả hoạt động

---

**Tổ văn phòng**: Gồm 1 Giám Đốc và 2 phó Giám Đốc có nhiệm vụ điều phối toàn bộ hoạt động của siêu thị. Tổ phải nắm được tình hình mua bán, doanh thu của siêu thị để báo cáo lại cho ban giám đốc. Việc báo cáo được thực hiện hàng tháng, hàng quý hoặc cũng có khi báo cáo đột xuất theo yêu cầu



# Các yếu tố cần xem xét

---

- Hỗ trợ ra quyết định
- Ưu thế cạnh tranh
- Hoàn vốn đầu tư: chỉ ra các lợi ích kinh tế rõ ràng (phân tích lợi nhuận)
- Giảm chi phí (phân tích chi phí)
- Hỗ trợ cho việc quản lý nghiệp vụ
- Khả năng thực hiện công việc phải nhanh hơn và tốt hơn



# Stakeholder và End user

---

- Đối tượng liên quan (stakeholder): là những liên quan trực tiếp tới hệ thống
- End user: người dùng hệ thống, có thể là các stakeholder



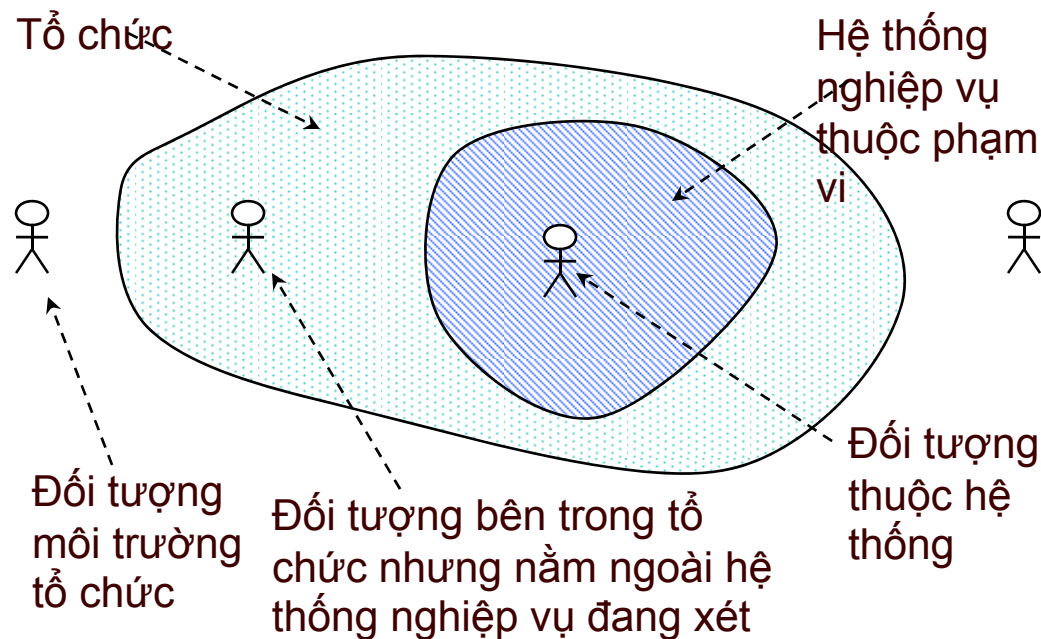
# Mô tả

---

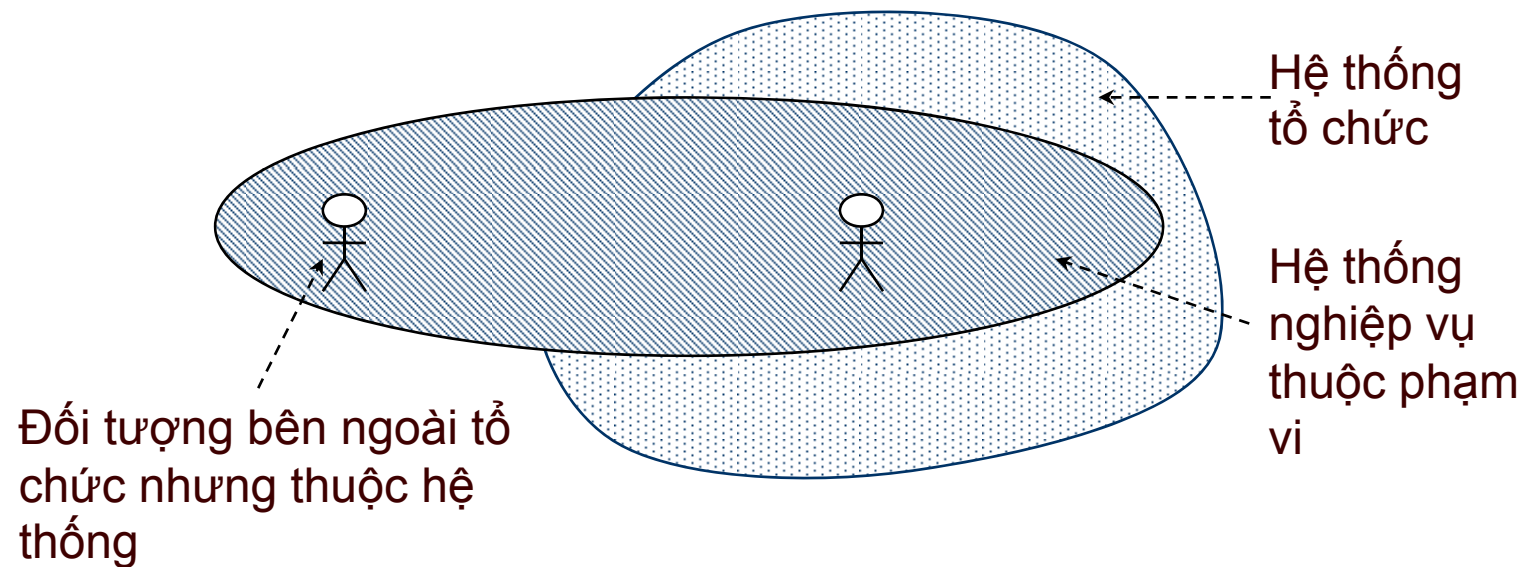
Tên đối tượng liên quan/ người dùng	Độ ưu tiên	Nhu cầu	Giải pháp hiện hành	Giải pháp đề xuất
Người quản lý	Cao	Xem các báo cáo thống kê theo các yêu cầu khác nhau	Báo cáo thống kê doanh thu	Hiển thị báo cáo theo nhiều tiêu chí khác nhau, thông tin bố trí dễ nhìn và đơn giản nhưng đầy đủ.

# Giới hạn hệ thống phát triển

- Chỉ ra những thực thể nằm ngoài hệ thống
- Chỉ ra những thực thể bên trong tổ chức nhưng nằm ngoài hệ thống



# Hệ thống thương mại điện tử e-Business



Các hệ thống thương mại điện tử e-Business, e-Commerce



# Phát biểu vấn đề (cho cải tiến)

Vấn đề	Cơ sở dữ liệu của các khách hàng thân thiết được lưu trữ ở nhiều nơi và không có sự đồng bộ .
Đối tượng chịu tác động	Khách hàng, người quản lý
Ảnh hưởng của vấn đề	Dịch vụ khách hàng thân thiết chỉ thiết lập được ở từng siêu thị. Điều này là bất hợp lý, làm rắc rối trong việc nâng cao dịch vụ khách hàng, làm giảm khả năng cạnh tranh của siêu thị.
Một giải pháp thành công	Nhân viên có thể sử dụng chung một tài khoản (account) cấp cho mỗi khách hàng được dùng ở tất cả siêu thị. Nâng cao khả năng chăm sóc khách hàng của siêu thị tốt hơn từ đó thu hút được khách hàng nhiều hơn, tăng doanh thu của siêu thị. Giúp người quản lý có thể làm tốt công tác quản lý khách hàng, theo dõi tình hình phục vụ khách hàng một cách dễ dàng.





# Thuật ngữ

Thuật ngữ	Diễn giải
Người quản lý	Người quản lý siêu thị và cũng là người quản trị hệ thống. Nguoiquanly được gọi chung cho những người được cấp quyền là "Quản lý", có thể bao gồm giám đốc, phó giám đốc, kế toán, nhân viên tin học, ...
Nhân viên bán hàng	Là nhân viên làm việc trong siêu thị. Nhân viên bán hàng, đứng ở quầy thu tiền và tính tiền cho khách hàng. Thông qua các mã vạch quản lý trên từng mặt hàng được nhân viên bán hàng nhập vào hệ thống thông qua một đầu đọc mã vạch.
Khách hàng thân thiết	Khách hàng thân thiết của siêu thị hay khách hàng đăng ký tham gia chương trình khách hàng thân thiết của siêu thị.



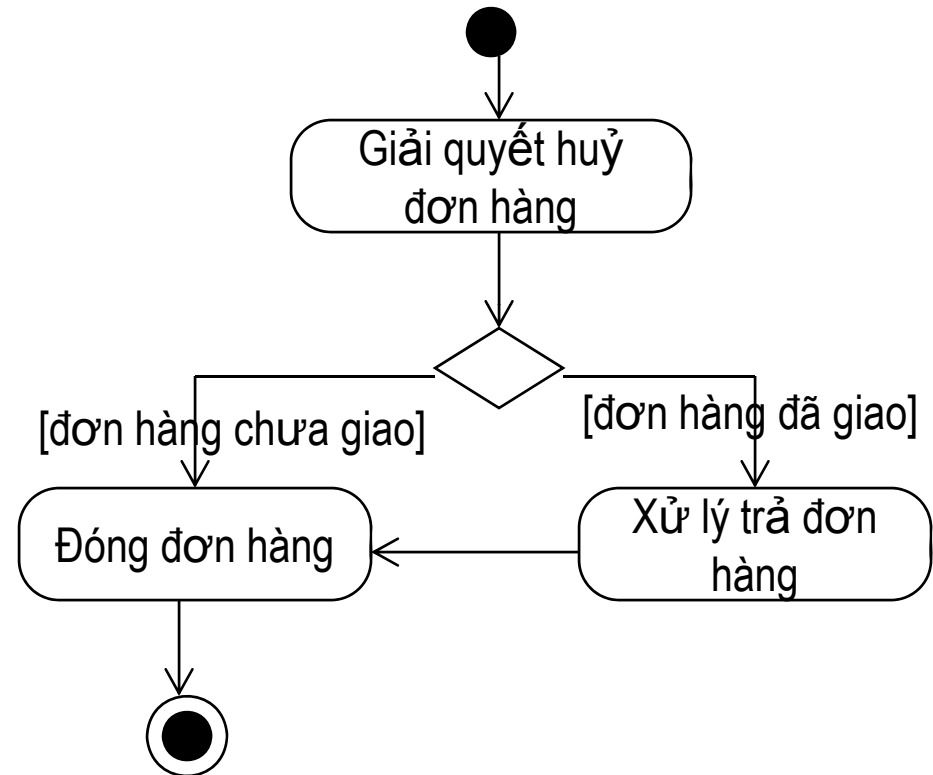
# Vấn đề nghiệp vụ

---

- Mô hình hóa nghiệp vụ
- Phân tích qui trình nghiệp vụ
- **Mô tả ràng buộc**
- Xác định worker (thừa tác viên) và thực thể nghiệp vụ
- Mô hình nghiệp vụ và thiết kế hệ thống

# Nguyên tắc ràng buộc 1

- Nguyên tắc kích hoạt và đáp ứng (điều kiện):
  - Là những ràng buộc hay điều kiện xác định khi nào một hành động xảy ra
  - Ví dụ: ràng buộc trong hoạt động hủy một đơn hàng

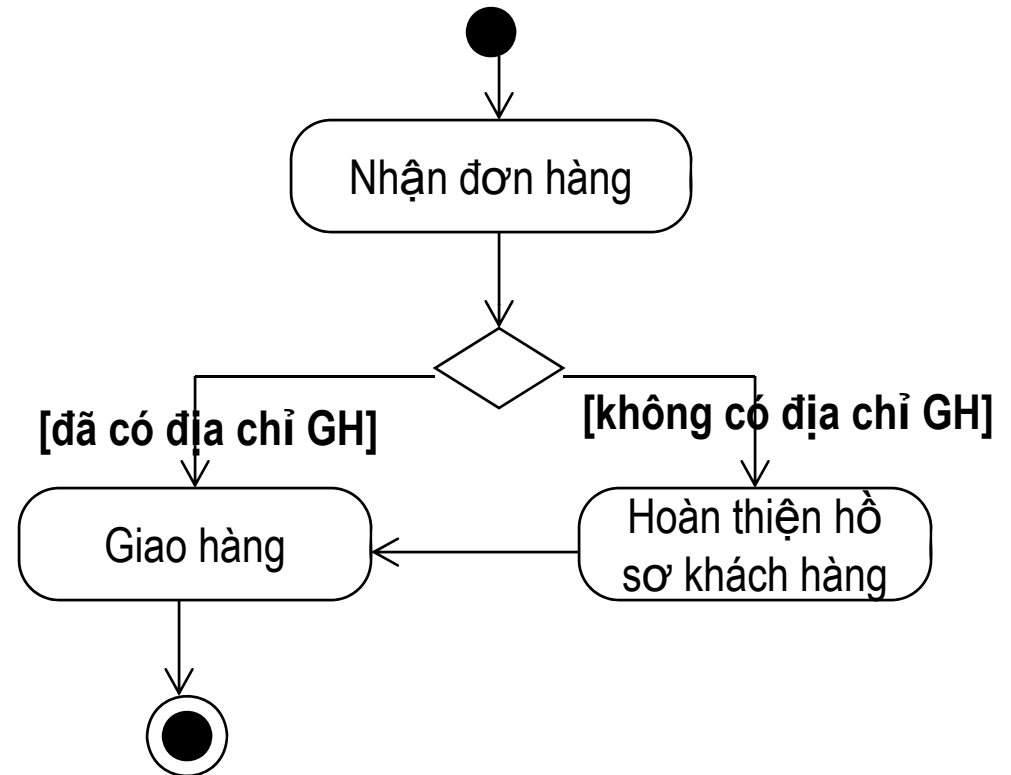


WHEN một đơn đặt hàng bị hủy bỏ  
IF hàng chưa được vận chuyển  
THEN kết thúc đơn đặt hàng

# Nguyên tắc ràng buộc 2

## ■ Nguyên tắc ràng buộc tác vụ (trước sau)

- Đây là những điều kiện phải thỏa trước và sau tác vụ để bảo đảm tác vụ đó hoạt động đúng
- Ví dụ: Trong một tổ chức quản lý đặt hàng, nguyên tắc sau đây có thể xảy ra

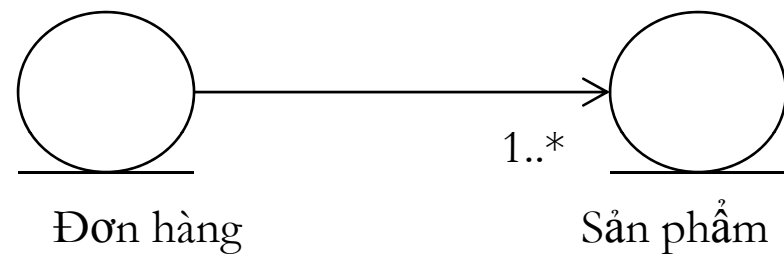


Vận chuyển Hàng hóa đến chỗ Khách hàng  
ONLY IF Khách hàng có địa chỉ

# Nguyên tắc ràng buộc 3

## ■ Nguyên tắc ràng buộc cấu trúc

- Xác định các quy định và điều kiện về các lớp, đối tượng, và các mối quan hệ giữa chúng
- Ví dụ:

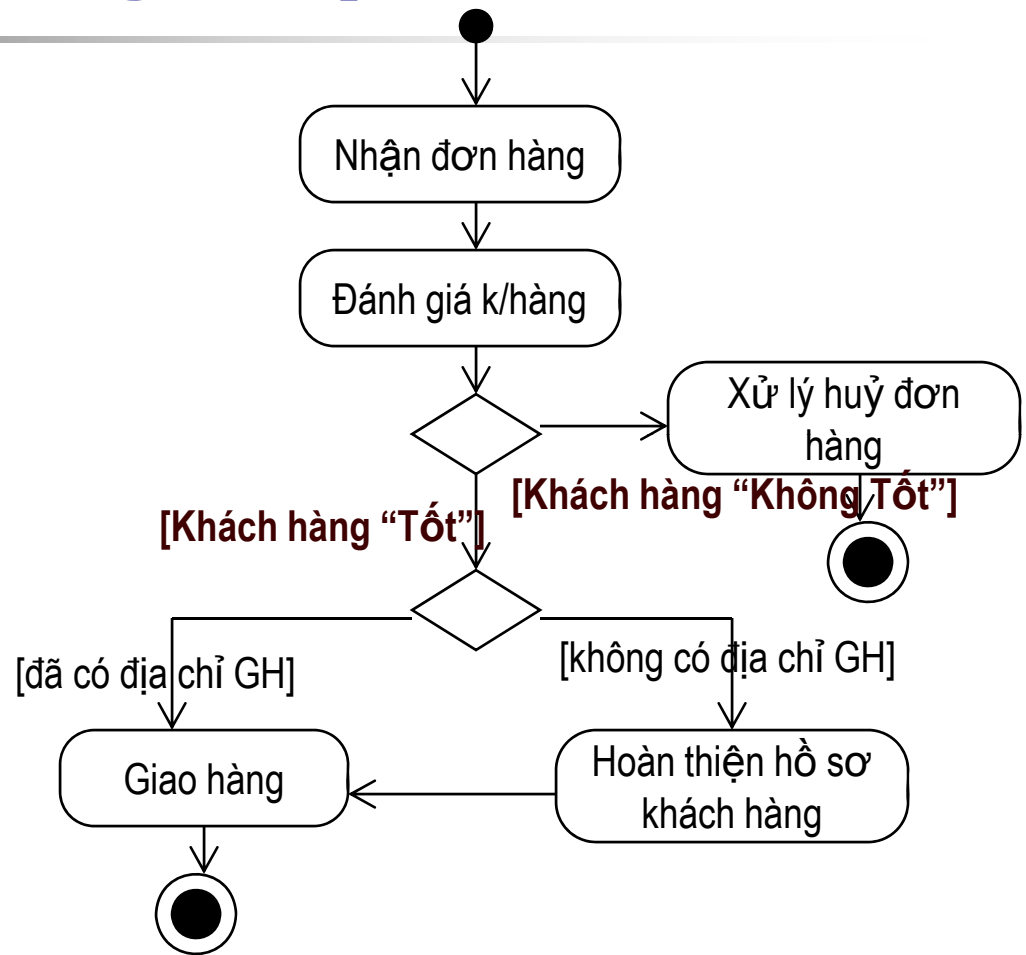


IT MUST ALWAYS HOLD THAT

Một đơn đặt hàng tham chiếu đến tối thiểu một sản phẩm

# Nguyên tắc ràng buộc 4

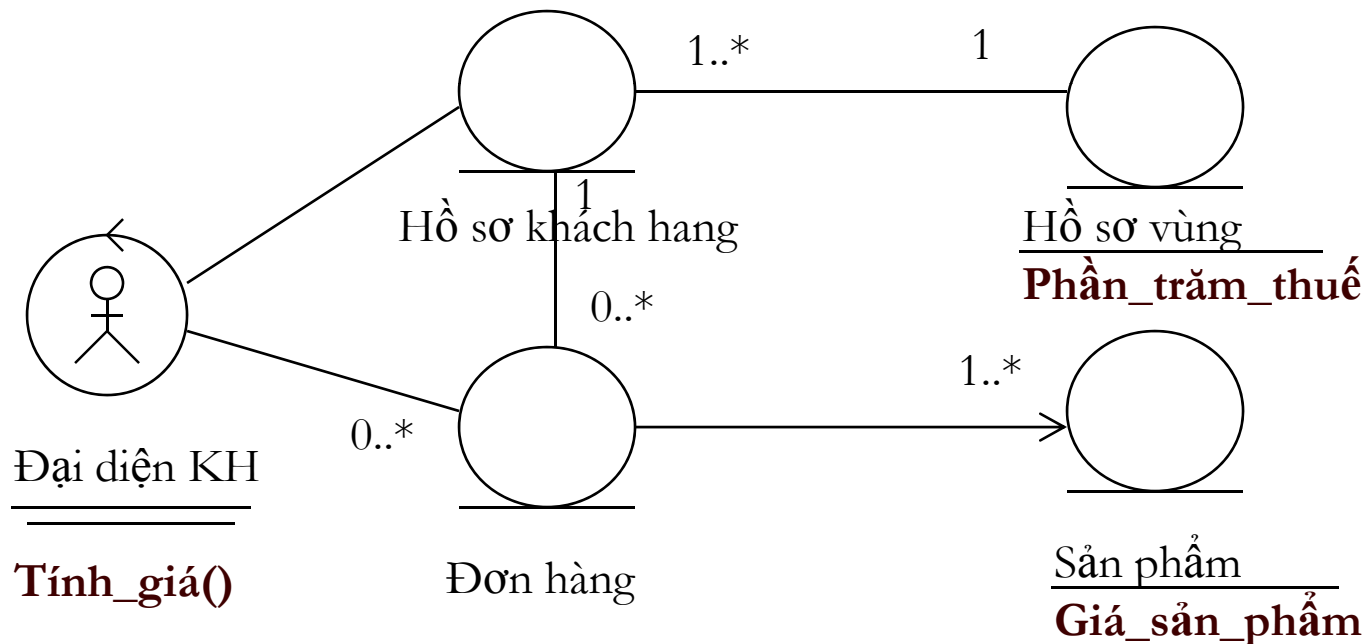
- **Các nguyên tắc suy diễn:** Xác định rằng nếu một số sự kiện nhất định là đúng, một kết luận có thể được suy ra
- Ví dụ: Thiết lập nguyên tắc sau đây để xác định trạng thái của một khách hàng



Một Khách hàng là một Khách hàng Tốt IF AND ONLY IF  
Những hóa đơn chưa thanh toán gửi đến Khách hàng đều ít hơn 30 ngày

# Nguyên tắc ràng buộc 5

## ■ Nguyên tắc tính toán:



Giá một sản phẩm được tính toán như sau:  
 $\text{giá sản phẩm} * (1 + \text{phần trăm thuế} / 100)$



# Vấn đề nghiệp vụ

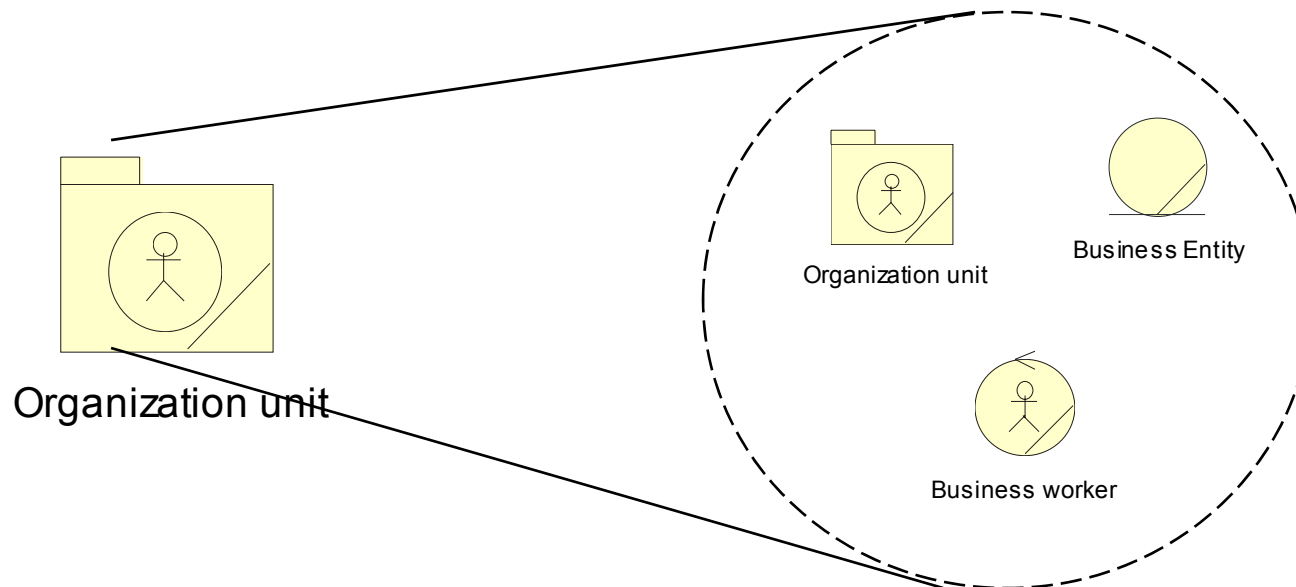
---

- Mô hình hóa nghiệp vụ
- Phân tích qui trình nghiệp vụ
- Mô tả ràng buộc
- **Xác định worker (thừa tác viên) và thực thể nghiệp vụ**
- Xác định yêu cầu tự động

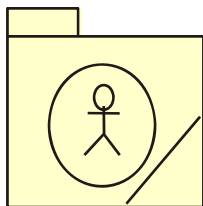


# Xác định đơn vị tổ chức

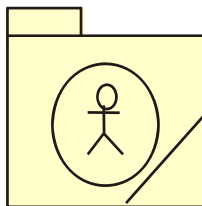
- Một đơn vị tổ chức bao gồm các worker, thực thể, và các đơn vị tổ chức khác có liên quan đến nhau theo một số tiêu chí nào đó
- Ký hiệu:



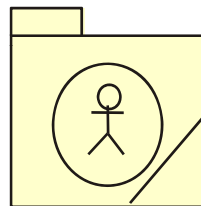
# Xác định đơn vị tổ chức



Bo phan ban hang



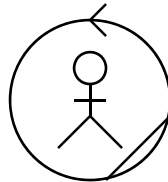
Bộ phận Kho



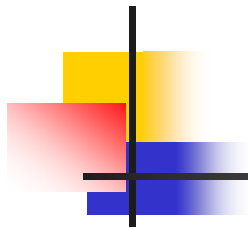
Bo phan quan ly

# Xác định worker nghiệp vụ

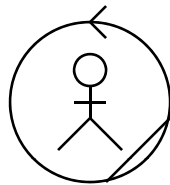
- Một worker (bên trong tổ chức) biểu diễn một vai trò hay một tập các vai trò trong nghiệp vụ
- Tương tác với các worker khác và các thực thể nghiệp vụ trong khi tham gia hoạt động của use case
- Ký hiệu:



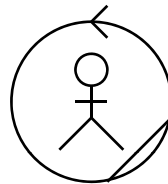
Worker name



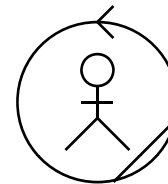
# Các worker



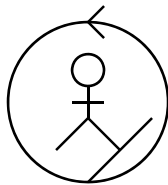
Nhân viên bán hàng



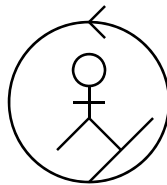
Nhân viên quản lý



Thủ kho

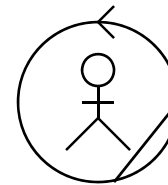


Nhân viên bán hàng

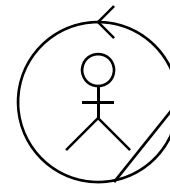


Thủ thư

**Giao tiếp với môi trường**



Quản trị hệ thống

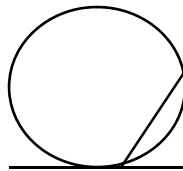


Thủ kho

**Làm việc bên trong**

# Xác thực thể nghiệp vụ

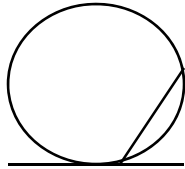
- Một thực thể biểu diễn một sự vật được xử lý hoặc sử dụng bởi các thừa tác viên
- Ký hiệu



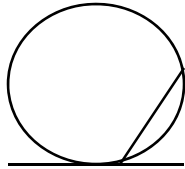
Tên thực thể nghiệp vụ

# Các loại thực thể nghiệp vụ

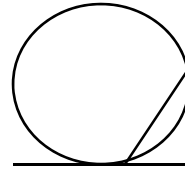
- **Đối tượng thông tin:** các đối tượng dùng để chứa thông tin dữ liệu hệ thống như là: sổ sách, chứng từ, hồ sơ, giấy tờ, thẻ, báo cáo, tập tin, CSDL,...



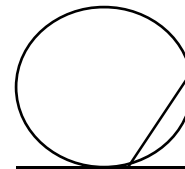
Thẻ thư viện



Hồ sơ khách hàng



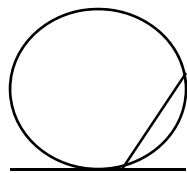
Hoá đơn



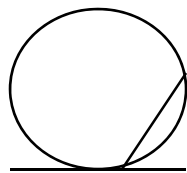
Sổ Nký bán hàng

# Các loại thực thể nghiệp vụ

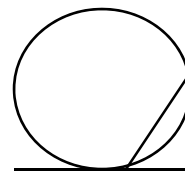
- **Đối tượng sự vật:** các đối tượng mô tả các sự vật trong hoạt động nghiệp vụ như là: các đối tượng trong quá trình sản xuất, các trang thiết bị, ...



Hàng hoá

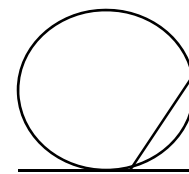


Nguyên vật liệu

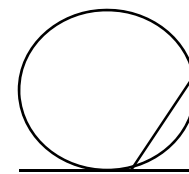


Sách

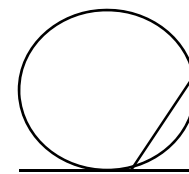
# Ví dụ



Thực đơn

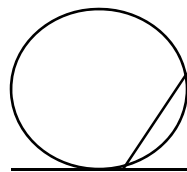


Thức ăn

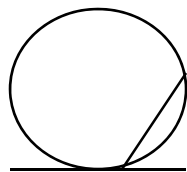


Thức uống

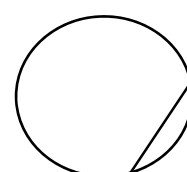
## Tại nhà hàng



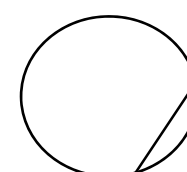
Sách



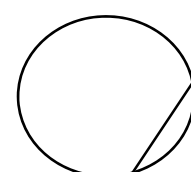
Hồ sơ độc giả



Vé máy bay



Thẻ lên máy bay



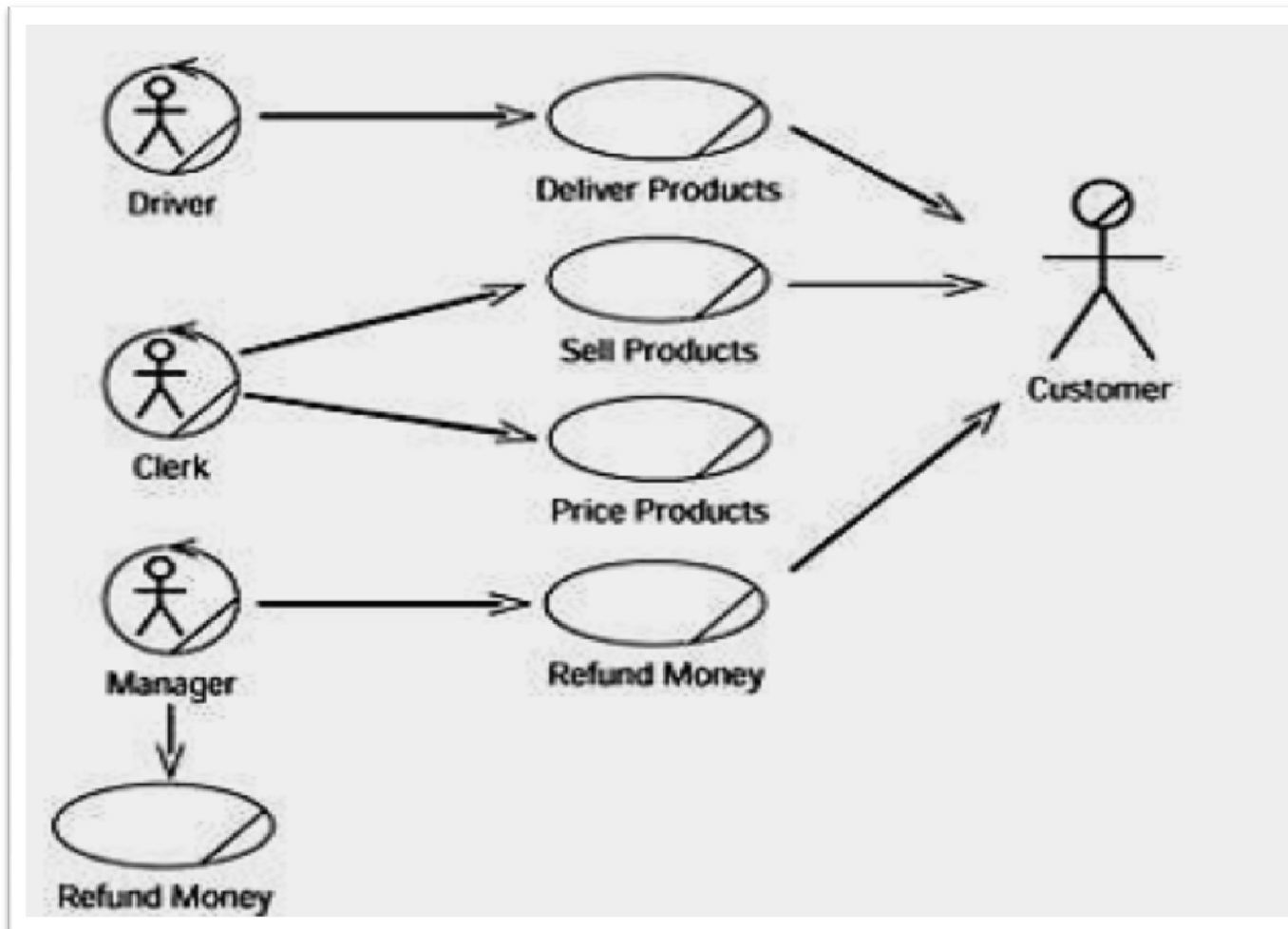
Hành lý

## Tại thư viện

## Tại sân bay



# Cửa hàng bán lẻ





# Vấn đề nghiệp vụ

---

- Mô hình hóa nghiệp vụ
- Phân tích qui trình nghiệp vụ
- Mô tả ràng buộc
- Xác định thừa tác viên và thực thể nghiệp vụ
- **Mô hình nghiệp vụ và thiết kế hệ thống**



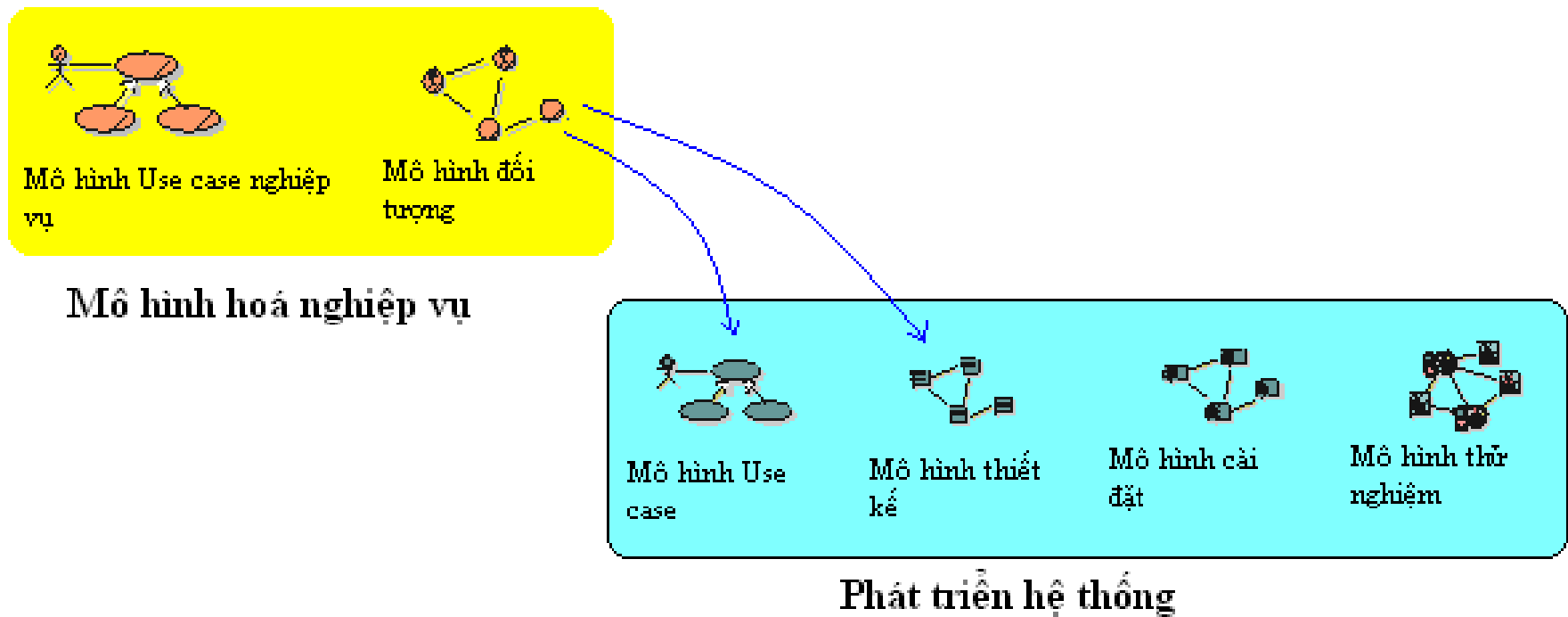
# Xác định các yêu cầu tự động hóa

---

## ■ Mục đích

- Hiểu được cách thức sử dụng các công nghệ mới cải thiện hoạt động hiệu quả của tổ chức.
- Xác định mức độ tự động hóa trong tổ chức.
- Thiết lập các yêu cầu hệ thống từ những kết quả mô hình hóa nghiệp vụ.

# Chuyển từ nghiệp vụ sang thiết kế

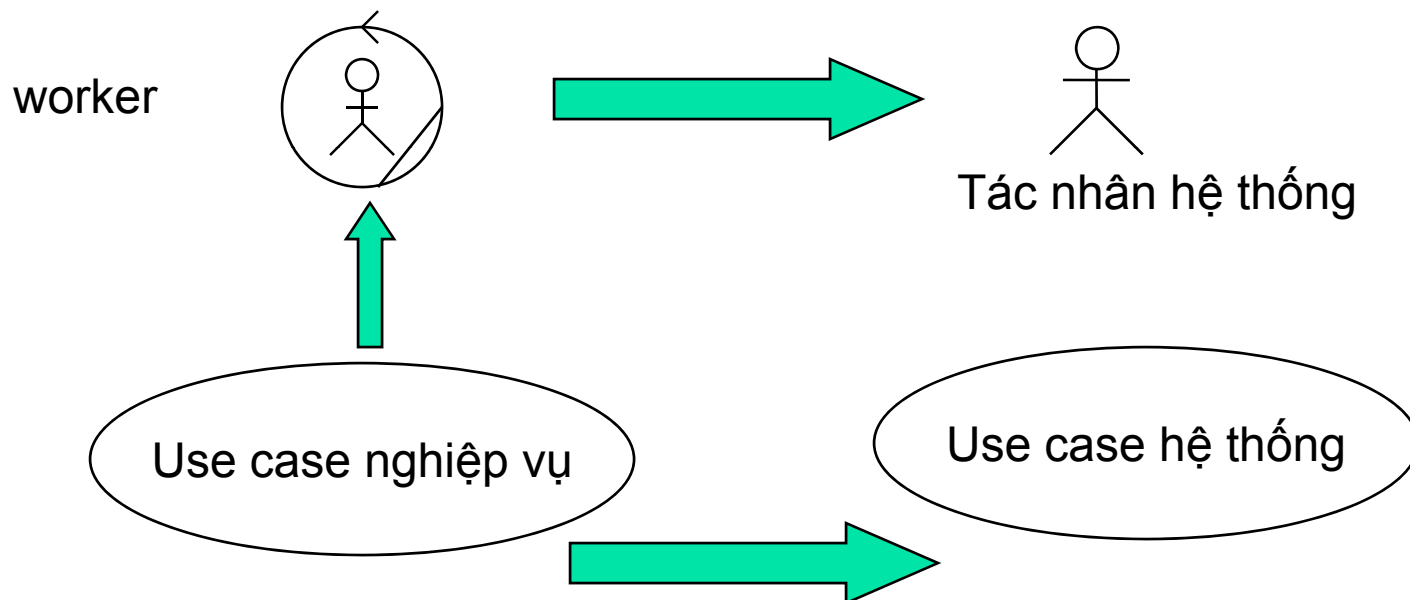




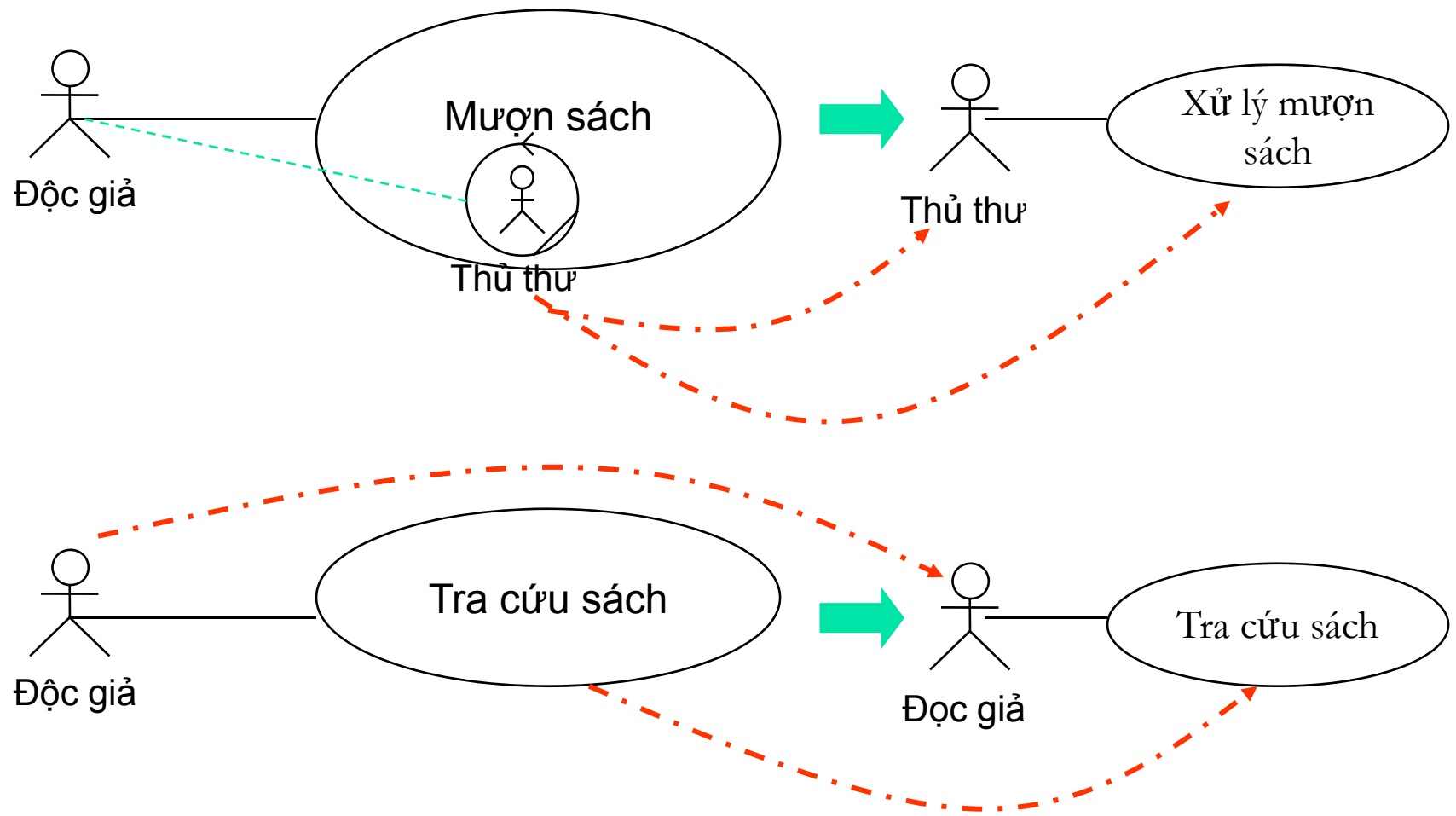
## Xác định actor và use case hệ thống

- Xác định xem worker nào sẽ sử dụng hệ thống thông tin?
  - Nếu có, xác định một tác nhân cho hệ thống thông tin trong mô hình use-case của hệ thống thông tin. Đặt tên tác nhân với tên của thừa tác viên hoặc tác nhân nghiệp vụ.
  - Đối với mỗi use case nghiệp vụ mà worker tham gia, tạo một use case hệ thống (actor hệ thống) và mô tả vắn tắt.
- Xem xét các mục tiêu về tốc độ thực thi hay những thông tin bổ sung về worker cần được chú thích như là một yêu cầu đặc biệt của use case hệ thống.

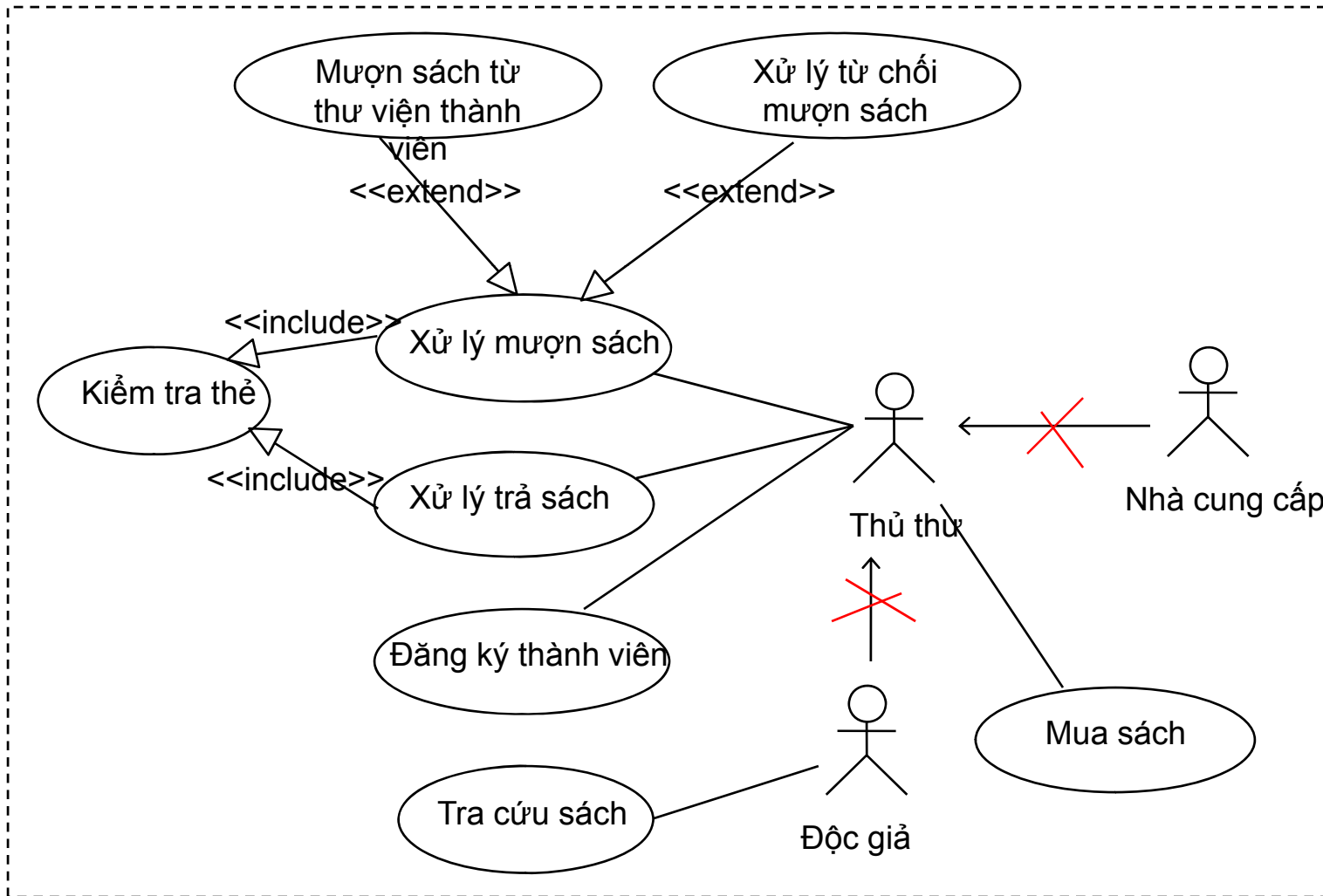
# Actor và use case hệ thống



# Hệ thống thư viện (từ nghiệp vụ)

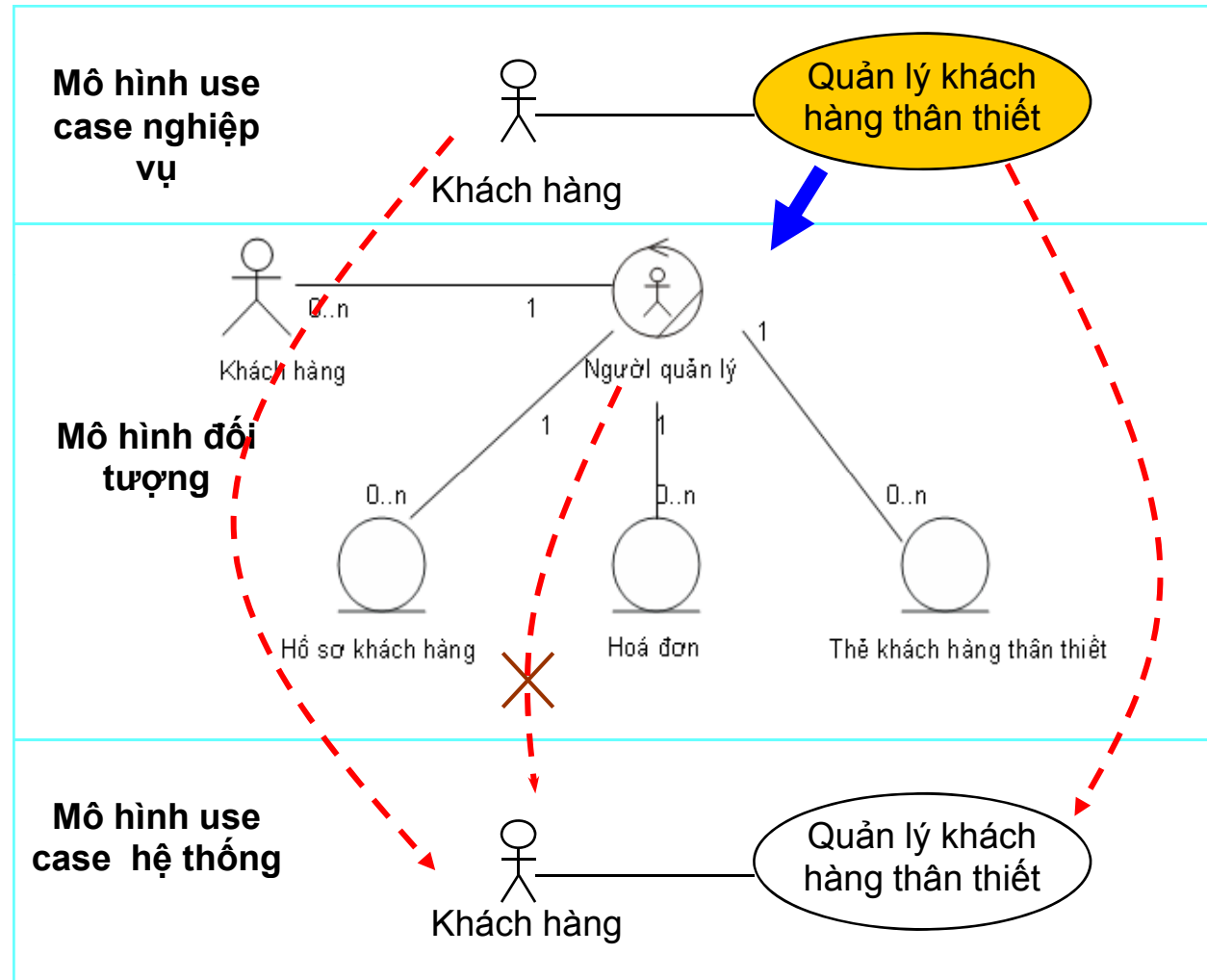


# Phần mềm Quản lý thư viện

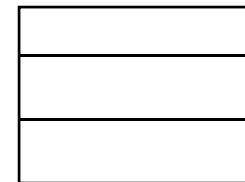
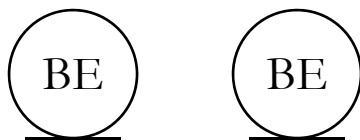
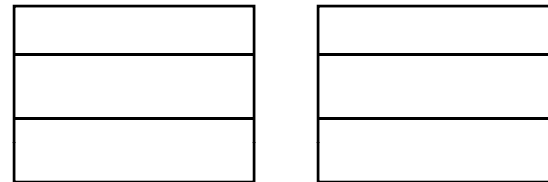
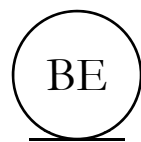




# Hệ thống bán hàng



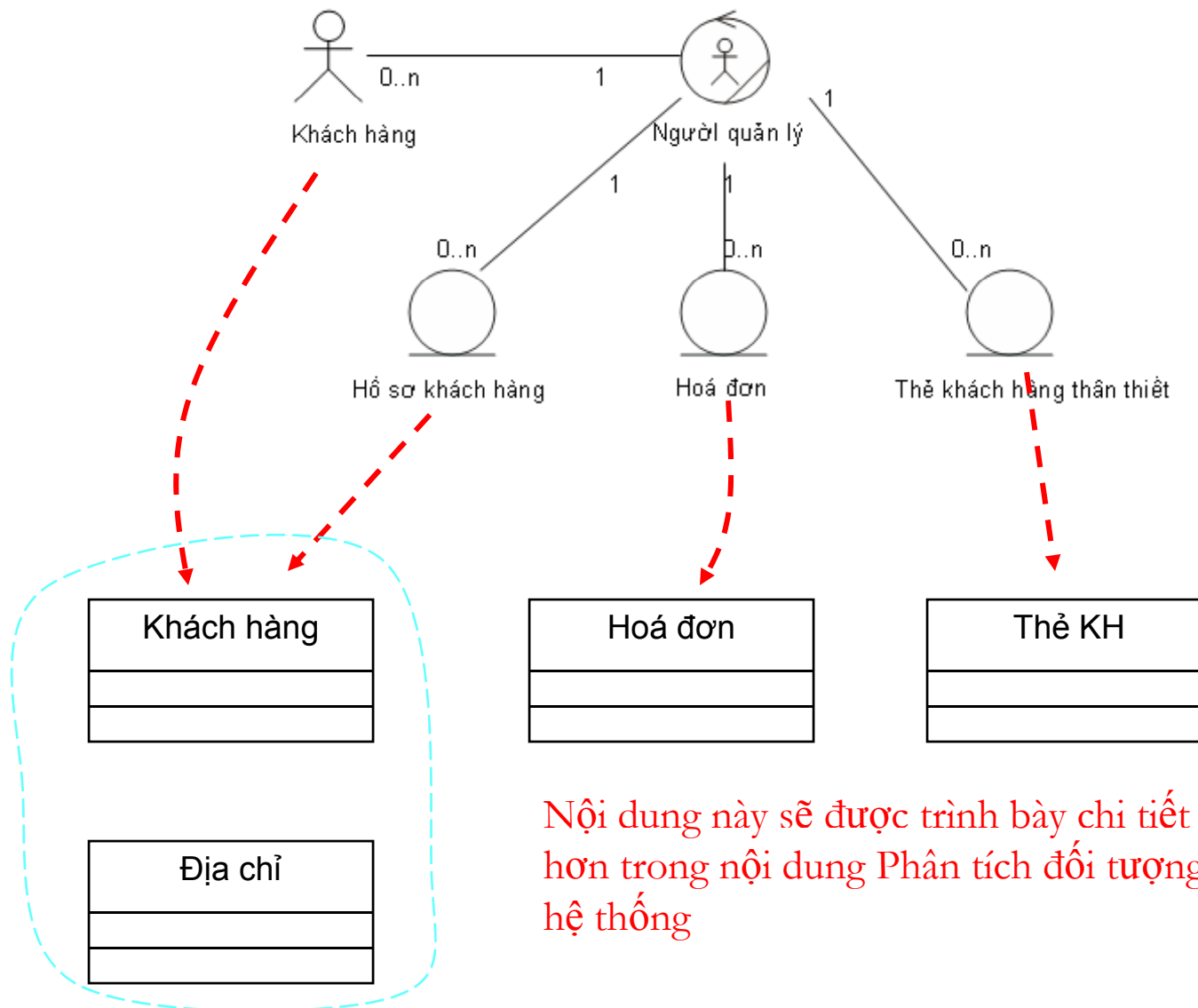
# Chuyển thực thể



Thực thể

Analysis class

# Chuyển thực thể trong Hệ thống bán hàng

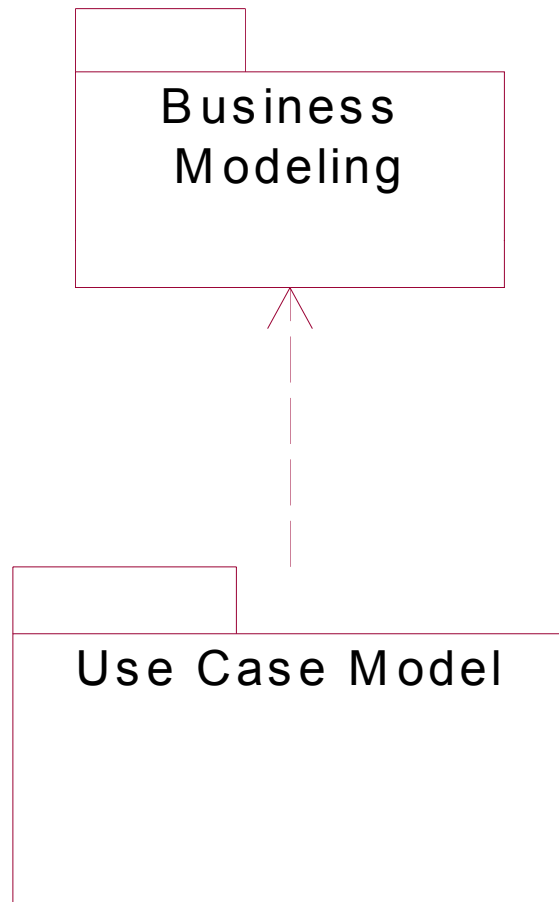




# Mô hình hoá hệ thống và nghiệp vụ

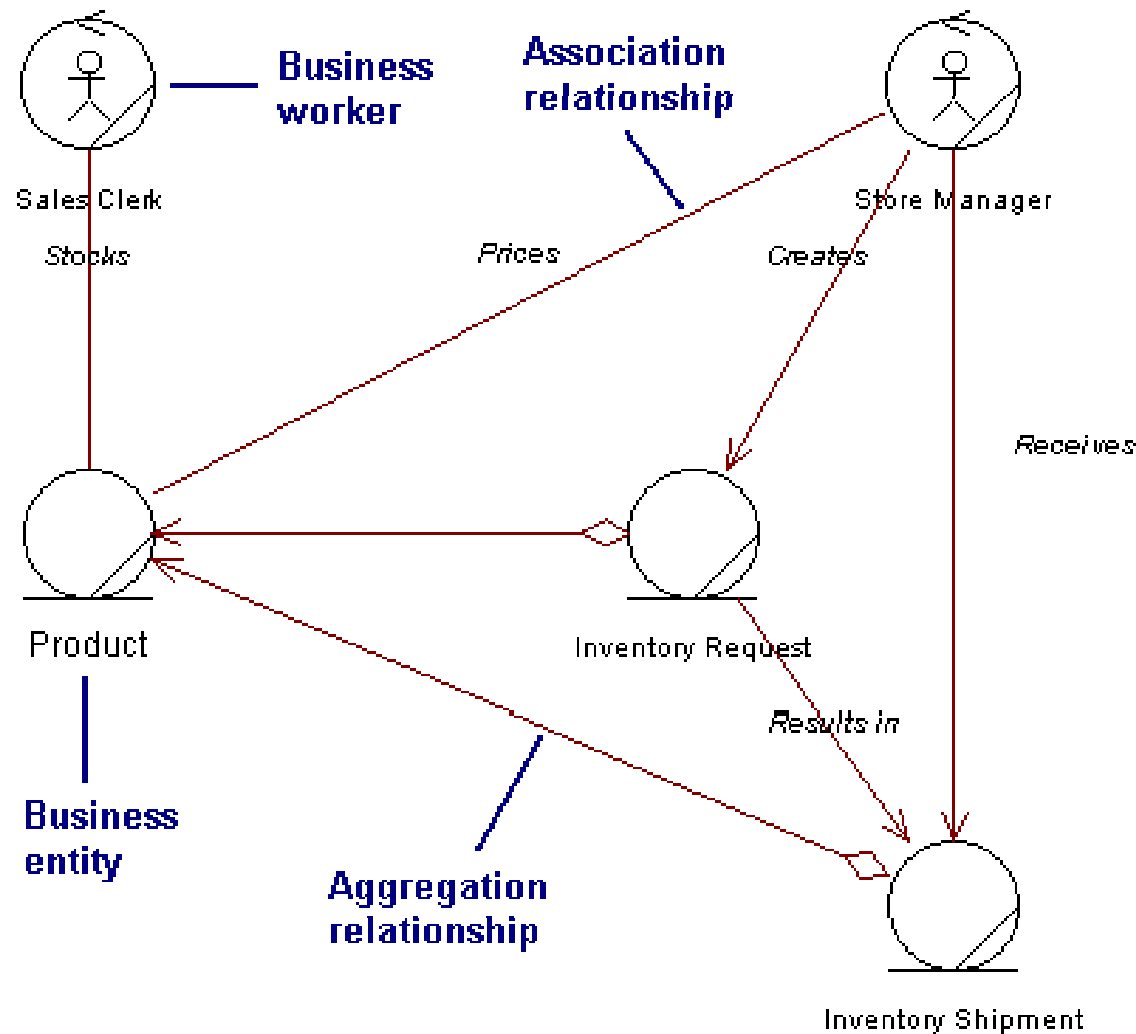
Item	Business Modeling	System Modeling
Use case	Describes what the business does	Describes what a system within the business does
Actor	External to the organization	External to the system (may be internal to the organization)
Business worker	Internal to the organization	Not used

# Hướng nhìn use case (Use case View)



- Mục đích của việc mô hình hoá nghiệp vụ:
  - Hiểu cấu trúc và hoạt động của 1 tổ chức
  - Để bảo đảm customer, end user và developer có sự hiểu biết chung về tổ chức
  - Để rút ra được các yêu cầu cho hệ thống cần xây dựng
- Mô hình use case là 1 mô hình các chức năng dự kiến của hệ thống và môi trường của nó, được dùng như 1 sự thoả thuận giữa customer và developer. Mô hình use case được dùng như điểm khởi đầu cho các giai đoạn tiếp theo: phân tích, thiết kế và thử nghiệm

# Lược đồ business entities





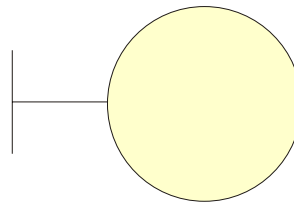
# Analysis object

---

- Có 3 loại đối tượng phân tích:
  - Boundary ·
  - Control ·
  - Entity

# Boundary object

- Nằm trên phạm vi giữa hệ thống và thế giới bên ngoài.
- Dùng để mô hình mối tương tác giữa các actor với hệ thống.
- Dùng để nắm bắt được các yêu cầu của giao diện người dùng.

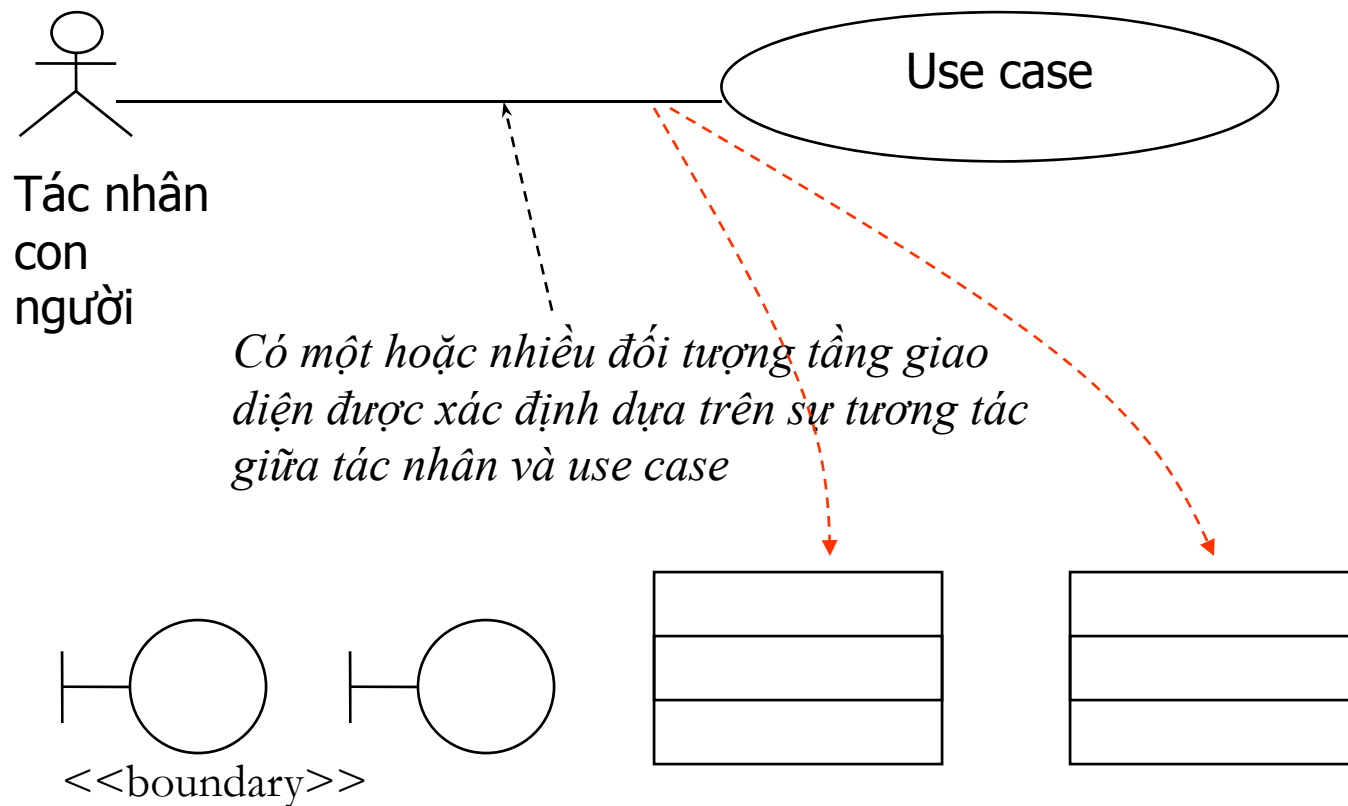


Form dat hang



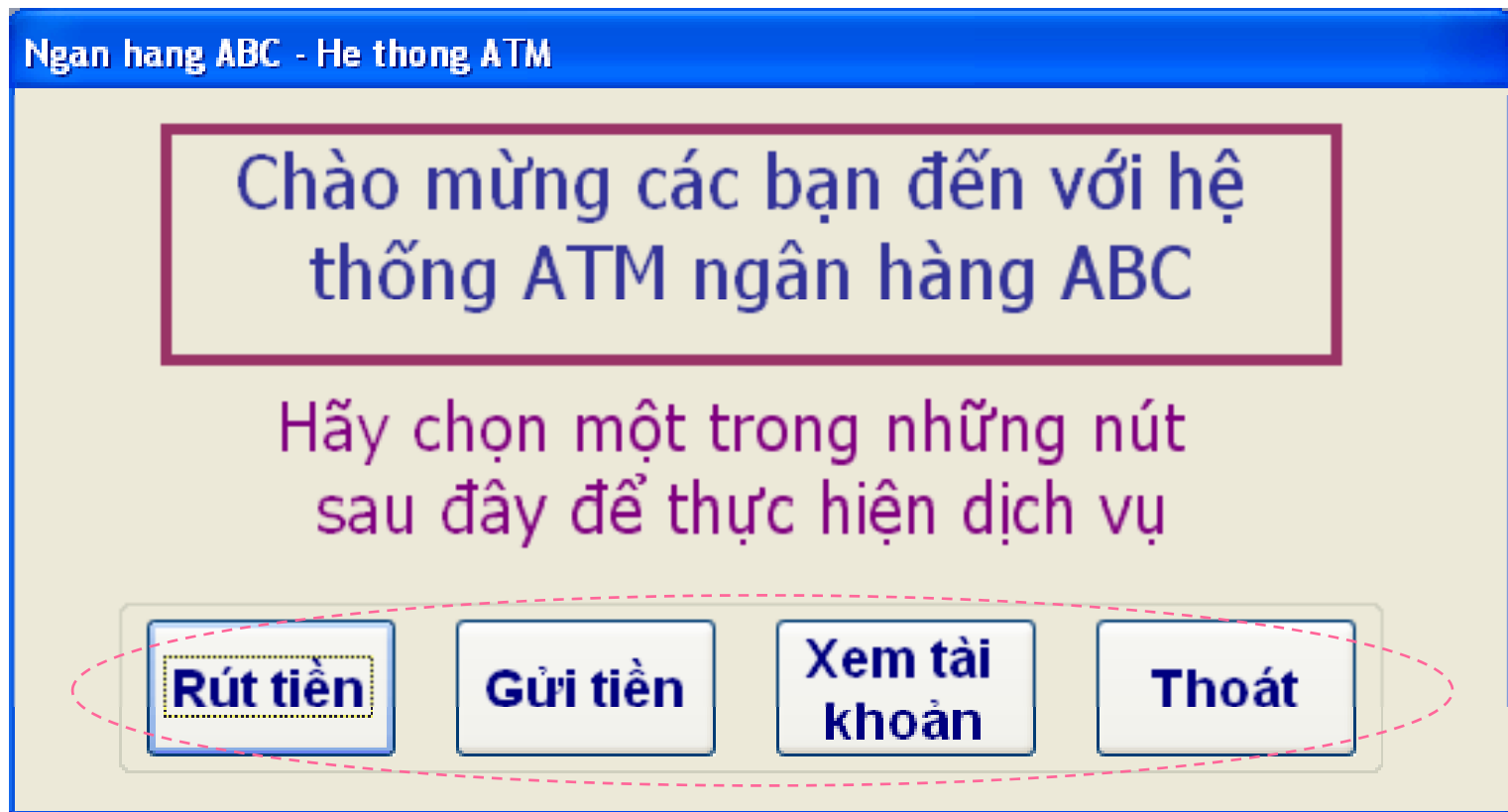
# Thiết kế lớp tầng giao diện

- Xác định các đối tượng ở tầng giao diện:



# Thiết kế tầng giao diện hệ thống ATM

- Thiết kế mẫu giao diện:
  - MáyATM\_GD



# Thiết kế tầng giao diện hệ thống ATM

- Thiết kế mẫu giao diện:
  - GiaoDichGD (Rút tiền)

The image shows a software interface for an ATM transaction. The title bar is blue and says 'Thực hiện giao dịch'. Below it, there are two tabs: 'Rút tiền' (Withdraw) and 'Gửi tiền' (Deposit). The 'Rút tiền' tab is active. The interface displays the current account balance as 'Số dư tài khoản hiện nay: 80755000' and the amount to be withdrawn as 'Nhập số tiền cần rút: 2500000'. There are two buttons at the bottom: 'Rút tiền' (Withdraw) and 'Đóng' (Close). To the right of the input fields is a numeric keypad with buttons for digits 0-9, a backspace button, and a zero button.

# Thiết kế tầng giao diện hệ thống ATM

## ■ Use case Rút tiền

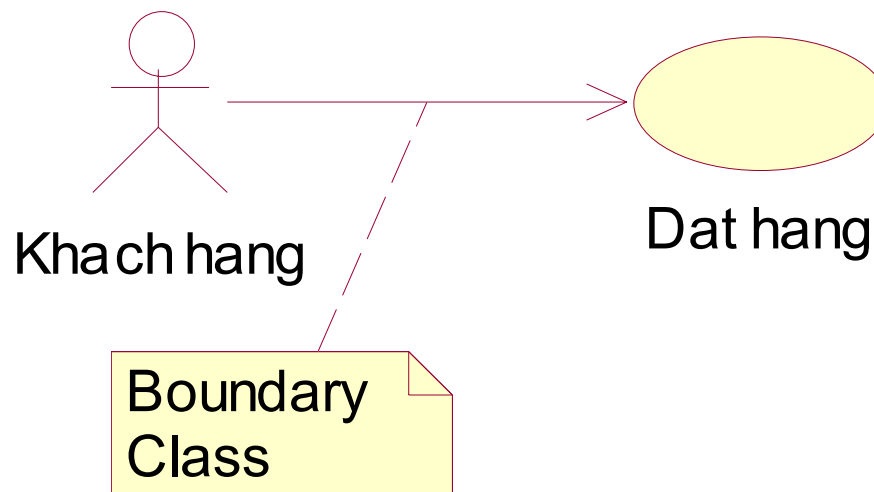
The screenshot shows an ATM transaction screen titled "Thực hiện giao dịch". It has two tabs: "Rút tiền" (selected) and "Gửi tiền". The "Số dư tài khoản hiện nay:" (Current account balance) is 80755000. The "Nhập số tiền cần rút:" (Enter amount to withdraw) field shows 2500000, with red circles highlighting the input fields and the amount. There are two buttons: "Rút tiền" and "Đóng". To the right is a numeric keypad with digits 0-9 and a backspace key.

Sự kiện	Hành động
Khách hàng chọn rút tiền (1)	-thực hiện rút tiền (TàiKhoản) -thông báo kết quả (GiaoDichGD) -in hoá đơn rút (GiaoDichGD) -đóng giao diện rút tiền (GiaoDichGD)
Khách hàng chọn đóng (2)	Đóng giao diện rút tiền (GiaoDichGD)

GiaoDichGD::-thôngBáo(thôngBáo:String)  
GiaoDichGD::-inHoáĐơn()  
GiaoDichGD::-+đóng()

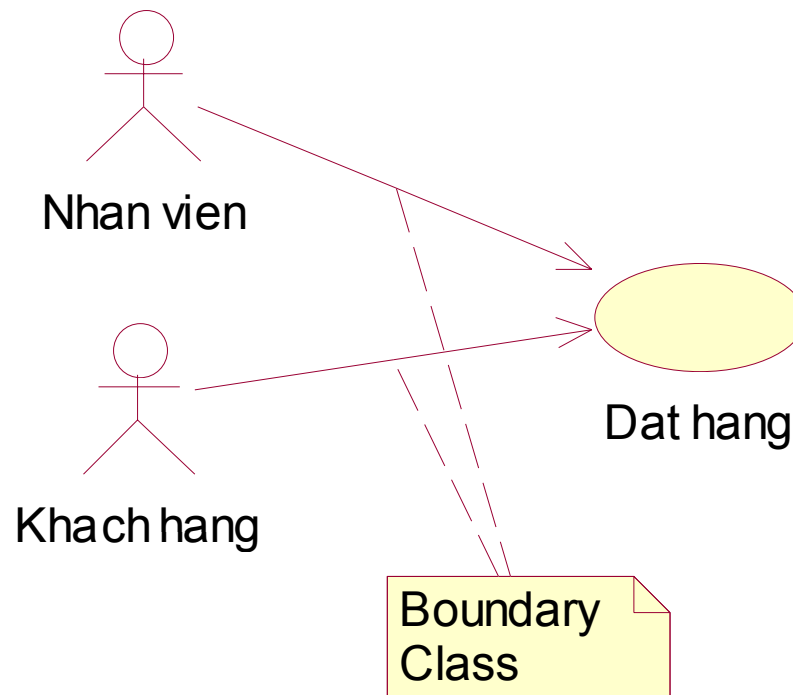
# Tìm boundary class

- Có ít nhất 1 boundary class cho mỗi tương tác giữa actor–use case. Nó là cái mà cho phép actor tiếp xúc với hệ thống.



# Tìm boundary class

- Không nhất thiết phải tạo class riêng biệt cho mỗi cặp actor- use case.
- Ví dụ: 2 actor có cùng 1 boundary class để giao tiếp với hệ thống

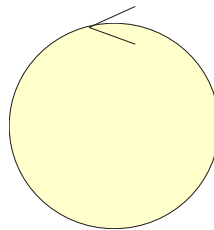




## Control object

---

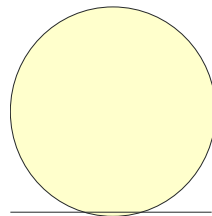
- Đối tượng điều khiển là các đối tượng không thực thi một chức năng nghiệp vụ nào cả, mà điều phối giám sát các đối tượng khác.



Xu ly dat hang

# Entity object

- Đối tượng thực thể được dùng để mô hình thông tin cần được lưu trữ bởi hệ thống và các hành vi có liên quan đến nó.
- Có đặc tính “persistent” thường được sử dụng lại trong các system use case khác.
- Chỉ ra cấu trúc dữ liệu có tính logical của hệ thống.



Don hang



# Đối tượng phân tích

