

Giáo trình

# **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG HƯỚNG ĐỐI TƯỢNG SỬ DỤNG UML**

*Biên soạn*

ThS. Phạm Nguyễn Cương

TS. Hồ Tường Vinh

## **Giới thiệu**

Hệ thống phần mềm càng ngày càng trở nên phức tạp. Các ứng dụng hôm nay có những yêu cầu và kiến trúc đòi hỏi phức tạp hơn rất nhiều so với quá khứ. Các kỹ thuật, công cụ, và phương pháp luận phát triển hệ thống phần mềm đang thay đổi một cách nhanh chóng. Các phương pháp phát triển phần mềm chúng ta sẽ sử dụng trong tương lai có lẽ sẽ khác so với các phương pháp hiện hành đang sử dụng. Tuy nhiên, một điều hiển nhiên là phát triển hướng đối tượng và các khái niệm cơ bản của nó đang được sử dụng rộng rãi. Nhiều trường học đã nhận ra được điều này và đã tạo ra những khoá học phát triển hệ thống hướng đối tượng như một phần chính yếu của hệ thống thông tin tin học hoá và các chương trình khoa học máy tính. Giáo trình này dự kiến sẽ cung cấp một kiến thức nền tảng về phát triển các hệ thống hướng đối tượng cho các đối tượng sinh viên những năm cuối. Mục tiêu của giáo trình là cung cấp một mô tả rõ ràng về các khái niệm nền tảng phát triển hệ thống hướng đối tượng. Trong đó, nhấn mạnh đến tính đơn giản của tiếp cận giúp sinh viên có kiến thức về UML có thể dễ dàng nắm bắt để phát triển một hệ thống hướng đối tượng.

## **Mục tiêu**

Sau khi học xong môn học này sinh viên có thể:

- Hiểu các nguyên lý nền tảng của kỹ thuật hướng đối tượng và các khái niệm về sự trừu tượng, tính bao bọc, tính thừa kế, và tính đa hình.
- Hiểu về một số quy trình phát triển hệ thống, nội dung các giai đoạn cơ bản của một quy trình phát triển, và một số phương pháp phân tích thiết kế hướng đối tượng.
- Tiếp cận toàn bộ quy trình phát triển hệ thống sử dụng các kỹ thuật hướng đối tượng
- Sử dụng UML như là một công cụ mô hình hoá trong quá trình phát triển hệ thống
- Phát triển hệ thống từ các mô hình use case được xem như là một mô hình phân tích nhằm biểu diễn đầy đủ yêu cầu hệ thống.
- Áp dụng một quy trình lặp, tập trung vào kiến trúc để phát triển một mô hình thiết kế đủ chi tiết, đủ mạnh đáp ứng với các nhu cầu:
  - o Phù hợp với các yêu cầu hệ thống đã được thống nhất qua mô hình use case trong giai đoạn phân tích.
  - o Tái sử dụng.
  - o Dễ dàng để cài đặt hệ thống trong một ngôn ngữ và môi trường cụ thể.

## PHẦN 1: TỔNG QUAN

### Chương 1

#### GIỚI THIỆU VỀ PHƯƠNG PHÁP VÀ PHƯƠNG PHÁP LUẬN PHÁT TRIỂN HỆ THỐNG HƯỚNG ĐỐI TƯỢNG

##### ***Giới thiệu về phương pháp phát triển hướng chức năng***

Đây là phương pháp cận truyền thống của ngành công nghiệp phần mềm trong đó quan điểm về phần mềm như là một tập hợp các chương trình (hoặc chức năng) và dữ liệu giả lập. Vậy chương trình là gì? Theo Niklaus Wirth, người tạo ra ngôn ngữ lập trình Pascal thì: “Chương trình = thuật giải + cấu trúc dữ liệu”. Điều này có nghĩa rằng có hai khía cạnh khác nhau của hệ thống được tiếp cận, hoặc tập trung vào các chức năng của hệ thống hoặc tập trung vào dữ liệu. Chúng ta chia hướng tiếp cận này thành hai thời kỳ: thời kỳ vào những năm thập niên 70, tiếp cận phân tích và thiết kế hệ thống theo phương pháp gọi là Descartes. Ý tưởng chính trong cách tiếp cận này là một quá trình lặp phân rã hệ thống thành các chức năng và ứng dụng phương pháp lập trình cấu trúc đơn thể chương trình, việc phân rã kết thúc khi một chức năng được phân rã có thể lập trình được. Trong thời kỳ này, người ta chưa quan tâm đến các thành phần không được tin học hoá mà chỉ xoay quanh đến các vấn đề trong hệ thống để lập trình, tập trung vào chức năng và ít tập trung vào dữ liệu (vì thời kỳ này đang chuẩn hoá và phát triển về cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu)

Thời kỳ vào những thập niên 80, tiếp cận phân tích thiết kế theo phương pháp gọi là *hệ thống*. Quan điểm chính của phương pháp này là tiếp cận hệ thống theo 2 thành phần, thành phần xử lý (thành phần động) và thành phần dữ liệu (thành phần tĩnh) của hệ thống. Cách tiếp cận của các phương pháp trong giai đoạn này tuân theo hai tính chất: tính toàn thể: tiếp cận hệ thống qua việc mô tả các hệ thống con và sự tương tác giữa chúng; tính đúng đắn: tìm kiếm sự phân rã, kết hợp các hệ thống con sao cho hành vi của nó tiêu biểu nhất của hệ thống trong môi trường tác động lên hệ thống con đó. Cách tiếp cận hệ thống theo hai thành phần chính là tiền đề cho cách tiếp cận hướng đối tượng trong các giai đoạn sau. Tuy nhiên, việc tiếp cận chủ yếu là hướng xoay quanh dữ liệu để thu thập và tổ chức dữ liệu nhằm khai thác mặt đáp ứng nhu cầu thông tin. Hướng tiếp cận gây khó khăn trong những hệ thống lớn và thường xuyên thay đổi cũng như là trong việc thiết kế nhằm tái sử dụng một thành phần đã có.

##### ***Giới thiệu về phương pháp phát triển hướng đối tượng***

Vào thập niên 90, phương pháp tiếp cận phân tích thiết kế đối tượng là sự tổng hợp của phương pháp Descartes và phương pháp hệ thống. Trong khi các mô hình được đưa ra trong những thập niên trước thường đưa ra dữ liệu và xử lý theo hai hướng độc lập nhau. Khái niệm đối tượng là sự tổng hợp giữa khái niệm xử lý và khái niệm dữ liệu chung trong một cách tiếp cận, và một hệ thống là một tập hợp các đối tượng liên kết nội. Có nghĩa rằng việc xây dựng hệ thống chính là việc xác định các đối tượng đó bằng cách cố gắng ánh xạ các đối tượng của thế giới thực thành đối tượng hệ thống, thiết kế và xây dựng nó, và hệ thống hình thành chính là qua sự kết hợp của các đối tượng này. Phương pháp hướng đối tượng được xem là phương pháp phân tích thiết kế thể hệ thứ ba, các phương pháp tiêu biểu là OOD, HOOD, BON, OSA, ... và sau này là OOSA, OOA, OMT, CRC, OOM, OOAD, OOSE, RUP/UML

##### **Đặc trưng cơ bản**

- *Tính bao bọc (encapsulation)*: quan niệm mối quan hệ giữa đối tượng nhận và đối tượng cung cấp thông qua khái niệm hộp đen. Nghĩa là đối tượng nhận chỉ truy xuất đối tượng cung cấp qua giao diện được định nghĩa bởi đối tượng cung cấp, đối tượng nhận không được truy cập đến các đặc trưng được xem là “nội bộ” của đối tượng cung cấp.

- *Tính phân loại (classification)*: gom nhóm các đối tượng có cùng cấu trúc và hành vi vào một lớp (class).
- *Tính kết hợp (aggregation)*: kết hợp các đối tượng và các đối tượng cấu thành nó để mô tả cấu trúc cục bộ của đối tượng (ví dụ: toà nhà <-> phòng, xe <-> sườn xe, bánh xe,...) , hoặc sự liên kết phụ thuộc lẫn nhau giữa các đối tượng.
- *Tính thừa kế (heritage)*: phân loại tổng quát hoá và chuyên biệt hoá các đối tượng, và cho phép chia sẻ các đặc trưng của một đối tượng.

### Phân loại

Phương pháp lập trình hướng đối tượng được chia thành 2 hướng như sau:

- Hướng lập trình: từ lập trình đơn thể chuyển sang lập trình hướng đối tượng với lý thuyết cơ bản dựa trên việc trừu tượng hóa kiểu dữ liệu.
- Hướng hệ quản trị CSDL: phát triển thành CSDL hướng đối tượng

Có 2 cách tiếp cận riêng biệt:

- Phương pháp kỹ thuật: hướng công nghệ phần mềm như OOD, HOOD, BON, BOOCH, MECANO, OODA,...
- Phương pháp toàn cục: hướng về HTTT như OOA, OOSA, OOAD, OMT, OOM,...

### Ưu điểm

- Cấu trúc hoá được các cấu trúc phức tạp và sử dụng được cấu trúc đệ quy: các phương pháp đối tượng đều sử dụng các mô hình bao gồm nhiều khái niệm để biểu diễn nhiều ngữ nghĩa khác nhau của hệ thống. Ví dụ: trong mô hình lớp của OMT có khái niệm mỗi kết hợp thành phần cho phép mô tả một đối tượng là một thành phần của đối tượng khác, trong khi nếu dùng mô hình ER truyền thống không có khái niệm này do đó không thể biểu diễn được quan hệ thành phần.
- Xác định được đối tượng của hệ thống qua định danh đối tượng<sup>1</sup>
- Tính thừa kế được đưa ra tạo tiền đề cho việc tái sử dụng

### Mô hình

*Mô hình (model)* là một dạng thức trừu tượng về một hệ thống, được hình thành để hiểu hệ thống trước khi xây dựng hoặc thay đổi hệ thống đó. Theo Efraim Turban, mô hình là một dạng trình bày đơn giản hoá của thế giới thực. Bởi vì, hệ thống thực tế thì rất phức tạp và rộng lớn và khi tiếp cận hệ thống, có những chi tiết những mức độ phức tạp không cần thiết phải được mô tả và giải quyết. Mô hình cung cấp một phương tiện (các khái niệm) để quan niệm hoá vấn đề và giúp chúng ta có thể trao đổi các ý tưởng trong một hình thức cụ thể trực quan, không mơ hồ.

Các đặc điểm của một mô hình:

- Diễn đạt một mức độ trừu tượng hóa (ví dụ: mức quan niệm, mức tổ chức, mức vật lý,...)
- Tuân theo một quan điểm (quan điểm của người mô hình hoá)
- Có một hình thức biểu diễn (văn bản, đồ họa: sơ đồ, biểu đồ, đồ thị,...)

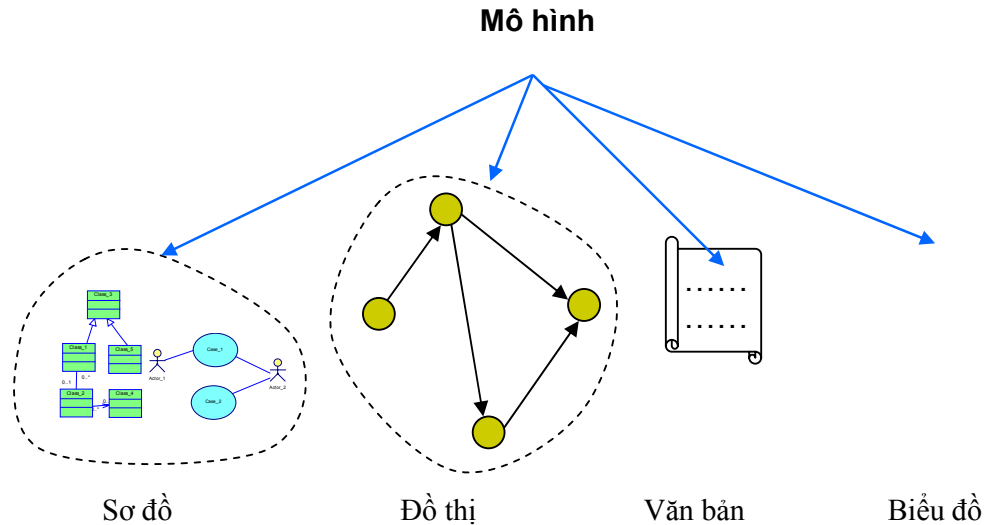
Hầu hết các kỹ thuật mô hình hóa sử dụng trong phân tích thiết kế là các ngôn ngữ đồ họa (đa số là sơ đồ - diagram), các ngôn ngữ này bao gồm một tập hợp các ký hiệu. Các ký hiệu này

---

<sup>1</sup> OID: Object Identifier

được dùng đi kèm theo các quy tắc của phương pháp luận giúp cho việc trao đổi các quan hệ thông tin phức tạp được rõ ràng hơn việc mô tả bằng văn bản.

Ví dụ :



### Mô hình tĩnh và mô hình động

**Mô hình tĩnh (static model):** được xem như là hình ảnh về thông số hệ thống tại một thời điểm xác định. Các mô hình tĩnh được dùng để trình bày cấu trúc hoặc những khía cạnh tĩnh của hệ thống.

**Mô hình động (dynamic model):** được xem như là một tập hợp các hành vi, thủ tục kết hợp với nhau để mô tả hành vi của hệ thống. Các mô hình động được dùng để biểu diễn sự tương tác của các đối tượng để thực hiện công việc hệ thống.

### Mục đích của mô hình hoá

Đứng trước sự gia tăng mức độ phức tạp của một hệ thống, việc trực quan hoá và mô hình hóa ngày càng trở nên chính yếu trong cách tiếp cận về một hệ thống, đặc biệt là cách tiếp cận hướng đối tượng. Việc sử dụng các ký hiệu để trình bày hoặc mô hình hóa bài toán có các mục đích sau:

- Làm sáng tỏ vấn đề: chúng ta có thể đưa ra được các lỗi hoặc các thiếu sót của hệ thống từ việc tiếp cận trực quan đồ họa hơn là các dạng trình bày khác như văn bản, đoạn mã,... Hơn nữa, việc mô hình hoá giúp chúng ta dễ dàng hiểu được hệ thống.
- Mô phỏng được hình ảnh tương tự của hệ thống: hình thức trình bày của mô hình có thể đưa ra được một hình ảnh giả lập như hoạt động thực sự của hệ thống thực tế, điều này giúp cho người tiếp cận cảm thấy thuận tiện khi làm việc với mô hình (là hình ảnh thu nhỏ của hệ thống thực tế)
- Gia tăng khả năng duy trì hệ thống: các ký hiệu trực quan có thể cải tiến khả năng duy trì hệ thống. Thay đổi các vị trí được xác định trực quan và việc xác nhận trực quan trên mô hình các thay đổi đó sẽ giảm đi các lỗi. Do đó, chúng ta có thể tạo ra các thay đổi nhanh hơn và các lỗi được kiểm soát hoặc xảy ra ít hơn.
- Làm đơn giản hóa vấn đề: mô hình hoá có thể biểu diễn hệ thống ở nhiều mức, từ mức tổng quát đến mức chi tiết, mức càng tổng quát thì ký hiệu sử dụng càng ít (do đó càng đơn giản hoá việc hiểu) và hệ thống được biểu diễn càng tổng quát.

## **Phương pháp luận phát triển hệ thống**

Phương pháp luận phát triển hệ thống bao gồm hai thành phần :

- Quy trình (process) : bao gồm các giai đoạn (phase) và tiến trình trong đó định nghĩa thứ tự các giai đoạn và các luật hình thành nên một quá trình phát triển hệ thống từ các công việc khởi tạo đến các công việc kết thúc của một dự án hệ thống.
- Các khái niệm (notation), phương pháp : các mô hình (bao gồm các phương pháp mô hình hoá của mô hình) cho phép mô hình hoá các kết quả của quá trình phát triển hệ thống.

### **Các giai đoạn cơ bản trong một quy trình :**

Để tự động hóa hoạt động xử lý, hệ thống phải trải qua một quá trình gồm nhiều bước được gọi là quá trình phát triển hệ thống. Cũng giống như nhiều tiến trình khác, phát triển hệ thống tự động cũng theo chu trình được gọi là vòng đời (Life cycle). Khái niệm vòng đời là một khái niệm rộng nó bắt đầu từ sự khởi đầu xây dựng cho đến kết thúc việc khai thác hệ thống. Nếu chúng ta chỉ chú trọng đến giai đoạn xây dựng và triển khai thì gọi là phát triển hệ thống. Vòng đời phát triển hệ thống - SDLC (Systems Development Life Cycle) là một phương pháp luận chung để phát triển nhiều loại hình hệ thống khác nhau. Tuy nhiên, các giai đoạn trong quá trình này cũng thay đổi khác nhau khoảng từ 3 cho đến 20 tùy theo qui mô và loại hình hệ thống chúng ta đang tiếp cận.

Phần sau đây sẽ giới thiệu các giai đoạn cơ bản làm nền tảng chung cho hầu hết quá trình phát triển hệ thống:

#### **Giai đoạn khởi tạo**

Hoạt động chính của giai đoạn này là khảo sát tổng quan hệ thống, vạch ra các vấn đề tồn tại trong hệ thống và các cơ hội của hệ thống, cũng như trình bày lý do tại sao hệ thống nên hoặc không nên được đầu tư phát triển tự động hóa. Một công việc quan trọng tại thời điểm này là xác định phạm vi của hệ thống đề xuất, trường dự án và nhóm phân tích viên ban đầu cũng lập một kế hoạch các hoạt động của nhóm trong các giai đoạn tiếp theo của dự án phát triển hệ thống. Kế hoạch này xác định thời gian và nguồn lực cần thiết. Đánh giá khả thi của dự án và nhất là phải xác định được chi phí cần phải đầu tư và lợi ích mang lại từ hệ thống. Kết quả của giai đoạn này là xác định được dự án hoặc được chấp nhận để phát triển, hoặc bị từ chối, hoặc phải định hướng lại.

#### **Giai đoạn phân tích**

Giai đoạn phân tích bao gồm các bước sau:

- Thu thập yêu cầu hệ thống: các phân tích viên làm việc với người sử dụng để xác định tất cả những gì mà người dùng mong muốn từ hệ thống đề xuất.
- Nguyên cứu các yêu cầu và cấu trúc hoá (mô hình hoá) để dễ dàng nhận biết và loại bỏ những yếu tố dư thừa.
- Phát sinh các phương án thiết kế chọn lựa phù hợp với yêu cầu và so sánh các phương án này để xác định giải pháp nào là đáp ứng tốt nhất các yêu cầu trong một mức độ cho phép về chi phí, nhân lực, và kỹ thuật của tổ chức. Kết quả của giai đoạn này là bản mô tả về phương án được chọn.

Trong phân tích hướng đối tượng giai đoạn này quan tâm đến mức độ trừu tượng hoá đầu tiên bằng cách xác định các lớp và các đối tượng đóng vai trò quan trọng nhằm diễn đạt các yêu cầu cũng như mục tiêu hệ thống. Để hiểu rõ các yêu cầu hệ thống chúng ta cần xác định ai là người dùng và là tác nhân hệ thống. Trong phương pháp phát triển hướng đối tượng cũng như phương pháp truyền thống, các mô tả kịch bản hoạt động được sử dụng để trợ giúp các phân tích viên hiểu được yêu cầu. Tuy nhiên, các kịch bản này có thể được mô tả không đầy đủ hoặc không theo một hình thức. Do đó, khái niệm use case được dùng trong giai đoạn này

nhằm biểu diễn chức năng hệ thống và sự tương tác người dùng hệ thống. Các kịch bản hoạt động lúc này sử dụng các mô hình động (dynamic diagram) nhằm mô tả nội dung của use case để làm rõ sự tương tác giữa các đối tượng, vai trò cũng như sự cộng tác của các đối tượng trong hoạt động của use case hệ thống. Trong giai đoạn phân tích, chỉ có các lớp tồn tại trong phạm vi hệ thống (ở thế giới thực) mới được mô hình hoá và như vậy thì kết quả mô hình hoá trong giai đoạn này sẽ phản ánh phạm vi của hệ thống. Các lớp về kỹ thuật, giao diện định nghĩa phần mềm cũng không quan tâm ở giai đoạn này.

### **Giai đoạn thiết kế**

Trong giai đoạn này kết quả của giai đoạn phân tích sẽ được chi tiết hoá để trở thành một giải pháp kỹ thuật để thực hiện. Các đối tượng và các lớp mới được xác định để bổ sung vào việc cài đặt yêu cầu và tạo ra một hạ tầng cơ sở kỹ thuật về kiến trúc. Ví dụ: các lớp mới này có thể là lớp giao diện (màn hình nhập liệu, màn hình hỏi đáp, màn hình duyệt,...). Các lớp thuộc phạm vi vấn đề có từ giai đoạn phân tích sẽ được "nhúng" vào hạ tầng cơ sở kỹ thuật này, tạo ra khả năng thay đổi trong cả hai phương diện: Phạm vi vấn đề và hạ tầng cơ sở. Giai đoạn thiết kế sẽ đưa ra kết quả là bản đặc tả chi tiết cho giai đoạn xây dựng hệ thống.

Về mức độ thiết kế thì có thể chia kết quả của giai đoạn này thành hai mức:

#### *Thiết kế luận lý*

Đặc tả hệ thống ở mức độ trừu tượng hóa dựa trên kết quả của giải pháp được chọn lựa từ giai đoạn phân tích. Các khái niệm và mô hình được dùng trong giai đoạn này độc lập với phần cứng, phần mềm sẽ sử dụng và sự chọn lựa cài đặt. Theo quan điểm lý thuyết, ở bước này hệ thống có thể cài đặt trên bất kỳ nền tảng phần cứng và hệ điều hành nào, điều này cho thấy giai đoạn này chỉ tập trung để biểu diễn khía cạnh hành vi và tính năng của đối tượng hệ thống.

#### *Thiết kế vật lý*

Chuyển đổi kết quả thiết kế luận lý sang các đặc tả trên phần cứng, phần mềm và kỹ thuật đã chọn để cài đặt hệ thống. Cụ thể là đặc tả trên hệ máy tính, hệ quản trị cơ sở dữ liệu, ngôn ngữ lập trình đã chọn,... Kết quả của bước này là các đặc tả hệ thống vật lý sẵn sàng chuyển cho các lập trình viên hoặc những người xây dựng hệ thống khác để lập trình xây dựng hệ thống.

### **Giai đoạn xây dựng**

Trong giai đoạn xây dựng (giai đoạn lập trình), các lớp của giai đoạn thiết kế sẽ được biến thành những dòng mã lệnh (code) cụ thể trong một ngôn ngữ lập trình hướng đối tượng (không nên dùng một ngôn ngữ lập trình hướng chức năng!). Phụ thuộc vào khả năng của ngôn ngữ được sử dụng, đây có thể là một công việc khó khăn hoặc dễ dàng. Khi tạo ra các mô hình phân tích và thiết kế trong UML, tốt nhất nên cố gắng né tránh việc ngay lập tức biến đổi các mô hình này thành các dòng mã lệnh. Trong những giai đoạn trước, mô hình được sử dụng để dễ hiểu, dễ giao tiếp và tạo nên cấu trúc của hệ thống; vì vậy, vội vàng đưa ra những kết luận về việc viết mã lệnh có thể sẽ thành một trở ngại cho việc tạo ra các mô hình chính xác và đơn giản. Giai đoạn xây dựng là một giai đoạn riêng biệt, nơi các mô hình được chuyển thành các mã lệnh.

### **Giai đoạn thử nghiệm**

Một hệ thống phần mềm thường được thử nghiệm qua nhiều giai đoạn và với nhiều nhóm thử nghiệm khác nhau. Các nhóm sử dụng nhiều loại biểu đồ UML khác nhau làm nền tảng cho công việc của mình: Thử nghiệm đơn vị sử dụng biểu đồ lớp (class diagram) và đặc tả lớp, thử nghiệm tích hợp thường sử dụng biểu đồ thành phần (component diagram) và biểu đồ cộng tác (collaboration diagram), và giai đoạn thử nghiệm hệ thống sử dụng biểu đồ Use case

(use case diagram) để đảm bảo hệ thống có phương thức hoạt động đúng như đã được định nghĩa từ ban đầu trong các biểu đồ này.

### Giai đoạn cài đặt và bảo trì

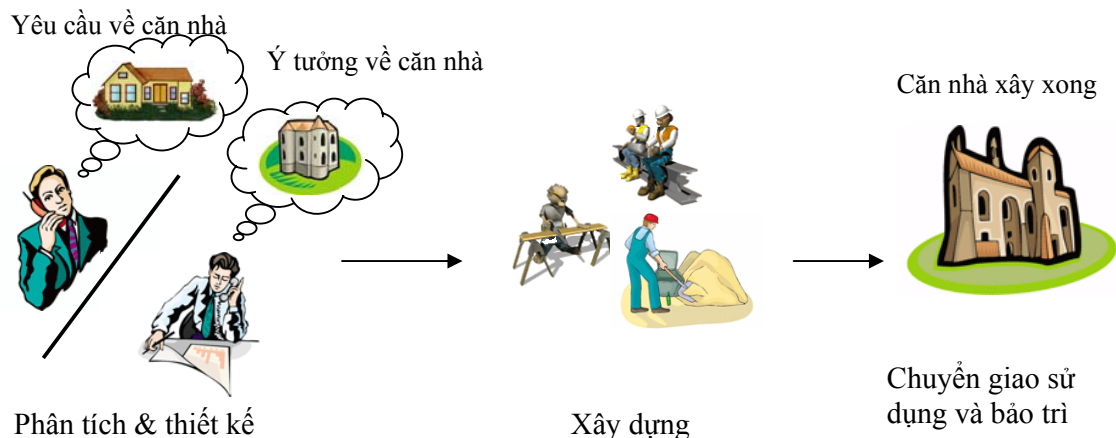
Điều chỉnh hệ thống phù hợp với nhu cầu sử dụng, các thay đổi phát sinh bao gồm:

- Chức năng sử dụng chưa phù hợp tốt nhất với người sử dụng hoặc khó sử dụng
- Các điều kiện và yêu cầu của người dùng hệ thống thay đổi, đòi hỏi phải chỉnh sửa sao cho hệ thống vẫn hữu dụng
- Các lỗi hệ thống phát sinh do quá trình kiểm tra còn sót lại
- Nâng cấp phiên bản mới của hệ thống

Bảo trì hệ thống không nên xem như là một giai đoạn tách rời mà nên xem như là một sự lặp lại chu trình của những giai đoạn trước đòi hỏi phải được nghiên cứu đánh giá và cài đặt. Tuy nhiên, nếu một hệ thống không còn hoạt động như mong muốn do có sự thay đổi quá lớn về hoạt động, hoặc nhu cầu mới đặt ra vượt quá sự giải quyết của hệ thống hiện tại, hoặc chi phí để bảo trì là quá lớn. Lúc này yêu cầu về hệ thống mới được xác lập để thay thế hệ thống hiện tại và một quy trình lại bắt đầu.

### Ví dụ về quy trình phát triển

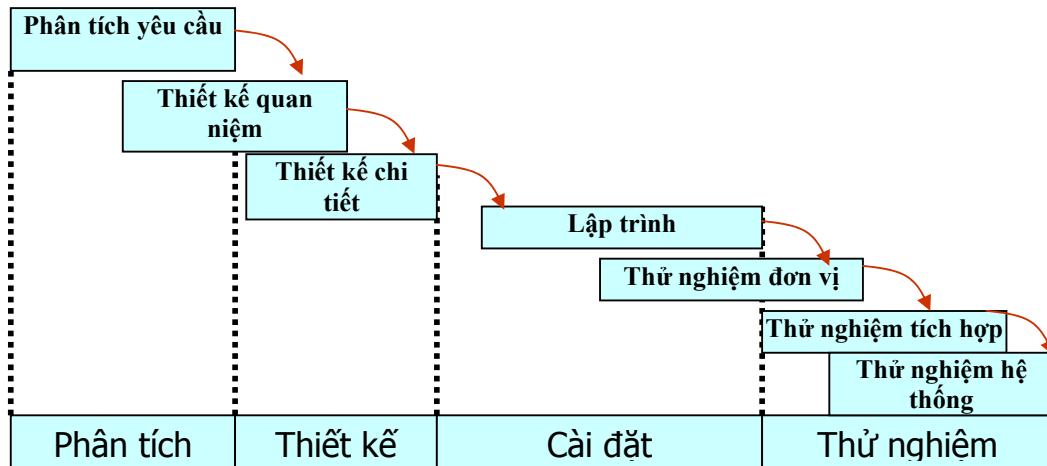
Chúng ta có thể hình dung rằng chúng ta muốn xây dựng một căn nhà. Công việc đầu tiên là chúng ta chắc chắn là chúng ta dự tính xem chúng ta sẽ bỏ số tiền bao nhiêu để xây dựng căn nhà này, dựa trên số tiền này chúng ta tìm kiếm và phác họa (có thể chỉ trong ý tưởng) căn nhà này phải như thế nào? Loại căn nhà theo kiểu gì, có mấy phòng, chiều rộng và chiều dài bao nhiêu, rồi nào đến nền nhà, màu sắc, tiện nghi?... Rồi sau đó, chúng ta sẽ chọn một đơn vị xây dựng (trong số nhiều đơn vị mà thỏa yêu cầu nhất). Tất cả các yêu trên sẽ phải trao đổi với đơn vị xây dựng này nhằm thống nhất về giá cả cũng như các điều khoản về yêu cầu xây dựng. Giai đoạn này được xem như là giai đoạn phân tích. Tiếp đó, đơn vị xây dựng sẽ thực hiện công việc thiết kế chi tiết của căn nhà, và từng đơn vị trong căn nhà (phòng, tường, trần, mái, phòng khách, phòng ăn, phòng ngủ,...). Giai đoạn này được xem là giai đoạn thiết kế. Sau đó các bản thiết kế chi tiết của căn nhà sẽ được bộ phận thi công dựa vào đó để tiến hành việc xây dựng. Giai đoạn này được xem là giai đoạn xây dựng. Căn nhà sau khi hoàn tất sẽ được chuyển giao để sử dụng, tất nhiên trong quá trình sử dụng nếu có các hư hỏng thì đơn vị xây dựng sẽ phải tiến hành bảo trì và sửa chữa.





## Mô số qui trình phát triển

### Qui trình thác nước (waterfall – Boehm 1970)

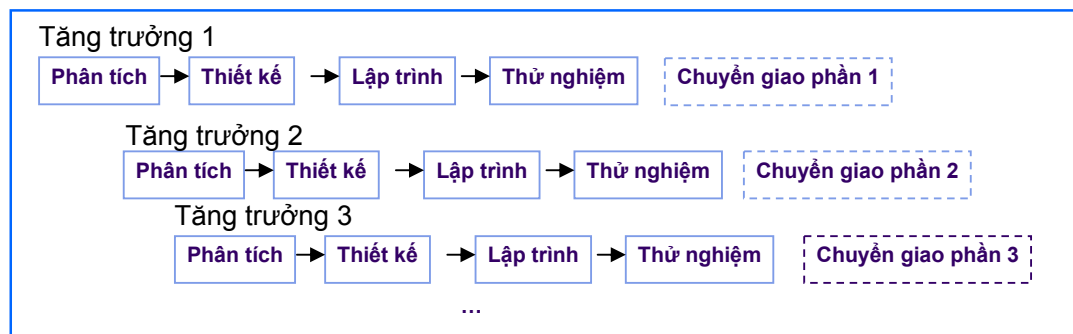


Đây là một qui trình đầu tiên được đề xuất và đã đưa ra được các giai đoạn căn bản nhất và đầy đủ cho một quá trình phát triển hệ thống, các giai đoạn bao gồm : phân tích, thiết kế, cài đặt và thử nghiệm hệ thống. Từ khi được đề xuất qui trình này nhanh chóng được phổ cập sử dụng rộng rãi trong giới công nghiệp và cho đến bây giờ đã có nhiều cải tiến hoàn thiện.

Nhược điểm :

- Qui trình là các giai đoạn tuần tự nối tiếp nhau, có nghĩa là giai đoạn phân tích phải được hoàn thành rồi đến giai đoạn thiết kế,... không cho phép sự quay lui và do đó, khi áp dụng qui trình này sẽ khó khăn khi ở giai đoạn trước có sự thay đổi (do sai sót, do nhu cầu người dùng thay đổi hoặc do có sự tiến hoá hệ thống,...).

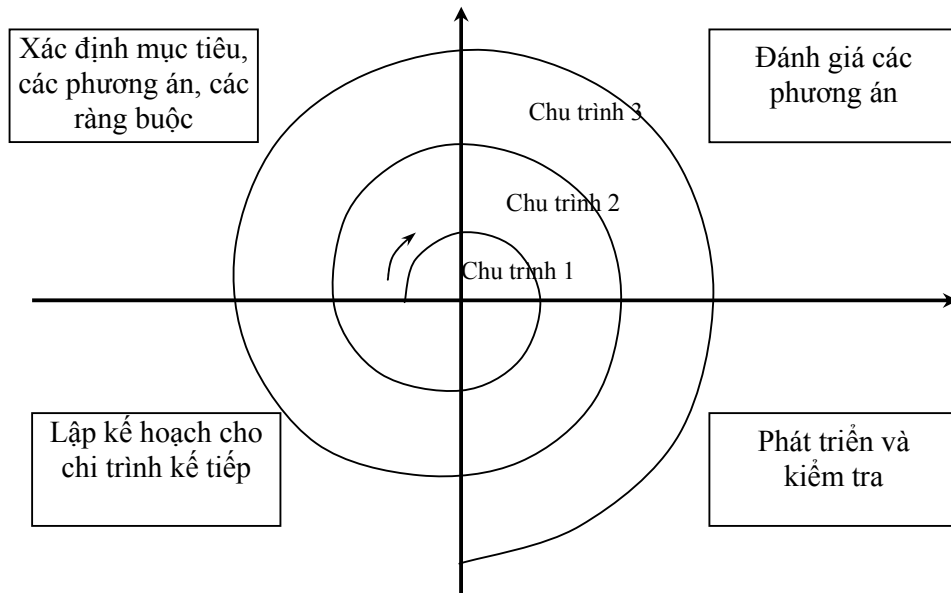
### Qui trình tăng trưởng (D.R. Graham 1988)



- Quan điểm chính của qui trình này là phát triển từng phần (phần hệ con) của hệ thống dùng qui trình thác nước.
- Lập : phân chia hệ thống thành những phần có thể phát triển một cách độc lập. Mỗi thành phần trong quá trình phát triển sẽ được áp dụng qui trình thác và được xem như một tăng trưởng của hệ thống. Khi thành phần cuối cùng hoàn tất thì quá trình phát triển toàn bộ hệ thống kết thúc.

Nhược điểm : qui trình này không thể áp dụng cho những hệ thống có sự phân chia không rõ ràng hoặc không thể phân chia thành những thành phần tác biệt.

### Qui trình xoắn ốc (Boehm 88)

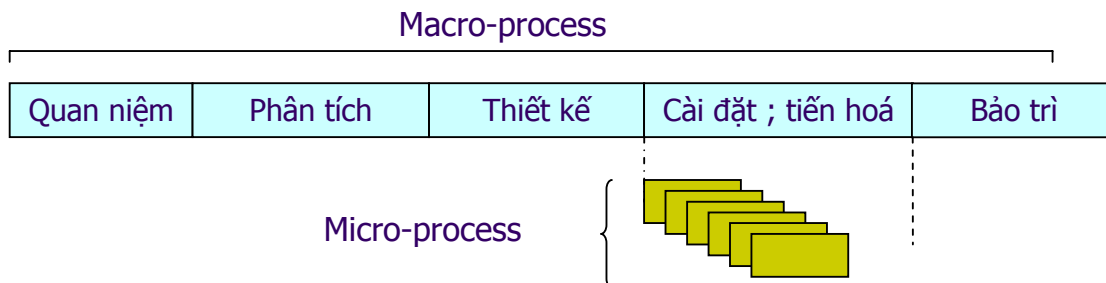


Là một quá trình gồm nhiều vòng lặp dựa trên bốn giai đoạn :

- Giai đoạn 1 :
  - o Đối với vòng lặp đầu tiên : phân tích yêu cầu
  - o Từ vòng lặp thứ hai trở đi : thiết lập mục tiêu cho vòng lặp, xác định các phương án để đạt mục tiêu đó ; các ràng buộc xuất phát từ các kết quả của các vòng lặp trước.
- Giai đoạn 2 :
  - o Đánh giá các phương án dựa trên các sản phẩm đạt được và tiến trình thực thi phương án.
  - o Xác định và giải quyết các rủi ro.
- Giai đoạn 3 :
  - o Phát triển và kiểm tra sản phẩm.
- Giai đoạn 4 :
  - o Lập kế hoạch cho vòng lặp tiếp theo.

Qui trình xoắn ốc cũng có thể áp dụng qui trình khác, ví dụ giai đoạn 3 có thể được thực hiện áp dụng qui trình thác nước.

### Qui trình Booch (1996)



Gồm hai tiến trình :

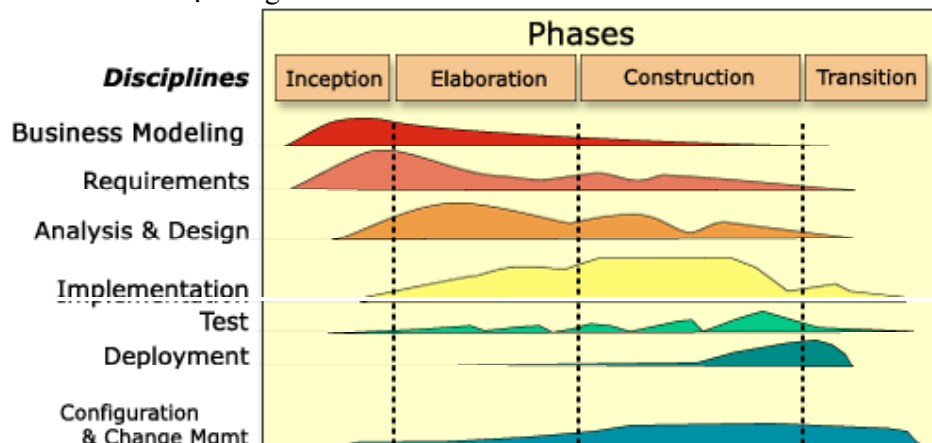
- Macro process : đóng vai trò như là bộ khung của micro process và bao phủ toàn bộ phạm vi dự án. Công việc chính của macro process là liên quan đến quản lý kỹ thuật của hệ thống trong việc chú trọng đến yêu cầu của người dùng và thời gian hoàn thành sản phẩm mà ít quan tâm đến chi tiết thiết kế hệ thống. Macro process gồm :

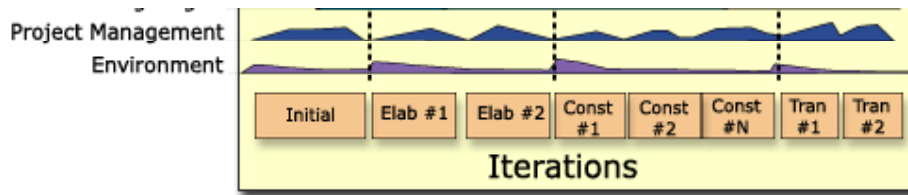
- Quan niệm hoá (conceptualization) : xác định yêu cầu căn bản, mục tiêu của hệ thống
- Phân tích và phát triển mô hình : sử dụng sơ đồ để mô hình hoá đối tượng hệ thống ; xác định vai trò và trách nhiệm của các đối tượng ; mô hình hoá hành vi của hệ thống thông qua các kịch bản mô tả hành vi.
- Thiết kế : thiết kế kiến trúc của hệ thống, các mối quan hệ giữa các lớp, các lớp sẽ được cài đặt, các vị trí định vị xử lý.
- Cài đặt, tiến hoá : tinh chế hệ thống thông qua nhiều vòng lặp. Lặp trình cài đặt phần mềm.
- Bảo trì : điều chỉnh lỗi phát sinh, cập nhật các yêu cầu mới
- Micro process : mô tả các hoạt động chi tiết của mỗi giai đoạn thông qua việc phân chia thành các hoạt động chi tiết theo từng nhóm phát triển hoặc theo từng đơn vị thời gian (giờ, ngày, tuần,...).

### RUP/UML (Rational Unified Process)

Quy trình bao gồm bốn giai đoạn chính và đan xen nhiều dòng hoạt động (activity flow) như là : mô hình hoá nghiệp vụ, phân tích yêu cầu, phân tích và thiết kế, cài đặt, thử nghiệm triển khai, ... Mỗi giai đoạn được hình thành từ những bước lặp (iteration).

- Khởi tạo (inception) :
  - Thiết lập phạm vi dự án, các điều kiện ràng buộc phạm vi, các kiến trúc đề xuất của hệ thống,
  - Xác định chi phí và thời gian của dự án,
  - Xác định độ rủi ro và môi trường hệ thống,
  - Xác định các thay đổi bổ sung, các tác động của các thay đổi này, các rủi ro nếu có,...
- Tinh chế (elaboration) :
  - Tinh chế kiến trúc hệ thống, yêu cầu hệ thống và đảm bảo kế hoạch sự ổn định của kế hoạch,
  - Đánh giá độ rủi ro, các thành phần sử dụng,
  - Xây dựng nền kiến trúc nền tảng hệ thống,...
- Xây dựng (construction) :
  - Quản lý tài nguyên, kiểm soát và thực hiện tối ưu hoá,
  - Hoàn thành việc phát triển các thành phần của sản phẩm, thử nghiệm sản phẩm,
  - Đánh giá sản phẩm cài đặt từ các tiêu chuẩn đã được thoả thuận,...
- Chuyển giao (transition) :
  - Thực hiện cài đặt hệ thống,
  - Thử nghiệm sản phẩm đã triển khai,
  - Thu thập các phản hồi từ phía người dùng,
  - Bảo trì hệ thống





### Phương pháp (method)

- Phương pháp là một quá trình tập trung vào một hoặc một vài giai đoạn của toàn bộ quy trình phát triển. Ví dụ :
  - o Phương pháp phân tích yêu cầu : mô tả cách thức và quy trình nhằm thu thập các yêu cầu của hệ thống,
  - o Phương pháp phân tích thiết kế : tập trung vào giai đoạn phân tích và thiết kế
  - o Phương pháp thử nghiệm : quy trình và cách thức cũng như các hoạt động thử nghiệm hệ thống
  - o ...
- Một phương pháp bao gồm một tập các ký hiệu đồ họa và văn bản, các luật sử dụng để mô tả các yếu tố hệ thống
- Một phương pháp thường được áp dụng trong một quy trình của phương pháp luận nhằm hướng dẫn cách thức thực hiện chi tiết của giai đoạn trong quy trình phát triển.

### Một số phương pháp

Phần sau đây sẽ tổng kết nội dung của một số phương pháp phát triển hệ thống hướng đối tượng:

#### OOD (Object Oriented Design - G.Booch 1991)

- Không đưa vào giai đoạn phân tích trong các phiên bản đầu tiên. Các bước phân tích hệ thống chuẩn bị cho giai đoạn thiết kế gồm :
  - o Xác định vấn đề thế giới thực
  - o Phát triển một chiến lược không hình thức hiện thức hoá từng phần đối với các vấn đề đã xác định
  - o Hình thức hoá chiến lược này
- Việc hình thức hoá chiến lược bao gồm một thứ tự các công việc sau :
  - o Xác định lớp và đối tượng ở mức trừu tượng hoá
  - o Xác định ngữ nghĩa cho các lớp và đối tượng
  - o Xác định mối quan hệ giữa các lớp và các đối tượng
  - o Cài đặt các lớp và đối tượng
- Đưa vào khái niệm gói (package) và dùng như một thành phần tổ chức của mô hình.
- Cài đặt các lớp và đối tượng thông qua việc đào sâu các chi tiết của lớp và đối tượng và cách thức cài đặt chúng trong một ngôn ngữ lập trình; cách thức tái sử dụng các thành phần và xây dựng các mô đun từ các lớp và đối tượng.
- Trong giai đoạn thiết kế, phương pháp này nhấn mạnh sự phân biệt giữa tầng luận lý (trong thuật ngữ lớp và đối tượng) và tầng vật lý (trong thuật ngữ mô đun và xử lý) và phân chia mô hình thành các mô hình động và mô hình tĩnh.
  - o Sơ đồ lớp (mô hình tĩnh)

- Sơ đồ đối tượng (mô hình tĩnh)
- Sơ đồ trạng thái (mô hình động)
- Sơ đồ thời gian (mô hình động)
- Sơ đồ mô-đun
- Sơ đồ xử lý

### ***HOOD (Hierarchical Object Oriented Design)***

- Khía cạnh tĩnh được biểu diễn qua sơ đồ đối tượng; văn bản hình thức cho phép hoàn thiện sơ đồ này thông qua việc chỉ dẫn các ràng buộc đồng bộ.
- Cấu trúc phân cấp được mô tả thông qua cấu trúc phân rã đối tượng
- Các giai đoạn cơ bản của giai đoạn thiết kế như sau :
  - Xác định vấn đề : xác định ngữ cảnh của đối tượng với mục đích tổ chức và cấu trúc hoá dữ liệu từ các yêu cầu của giai đoạn phân tích.
    - Diễn đạt vấn đề
    - Phân tích và cấu trúc hoá dữ liệu yêu cầu : thu thập và phân tích tất cả thông tin liên quan đến vấn đề, bao gồm môi trường mà hệ thống được thiết kế.
  - Phát triển chiến lược giải pháp: phác hoạ giải pháp vấn đề thông qua việc xác định các đối tượng ở mức trừu tượng hoá cao.
  - Hình thức hoá và mô hình hoá chiến lược : xác định các đối tượng và các toán tử. Phát sinh một giải pháp thiết kế dùng sơ đồ cho phép trực quan hoá các khái niệm, bao gồm năm bước :
    - Xác định đối tượng
    - Xác định các toán tử
    - Nhóm các đối tượng và các toán tử
    - Mô tả đồ hoạ
    - Điều chỉnh các quyết định thiết kế
  - Hình thức hoá giải pháp : giải pháp được hình thức hoá thông qua
    - Mô tả hình thức giao diện đối tượng
    - Mô tả hình thức đối tượng và các cấu trúc điều khiển toán tử

### ***OMT (Object modeling Technique)***

Cung cấp ba tập khái niệm diễn đạt ba cách nhìn về hệ thống. Sử dụng một phương pháp để dẫn dắt tới ba mô hình tương ứng với ba cách nhìn hệ thống. Các mô hình đó là :

- Mô hình đối tượng mô tả cấu trúc tĩnh của các đối tượng bên trong hệ thống và các quan hệ của chúng. Các khái niệm chính là :
  - Lớp
  - Thuộc tính
  - Toán tử
  - Thừa kế
  - Mối kết hợp (association)
  - Mối kết hợp thành phần (aggregation)
- Mô hình động hệ thống mô tả các khía cạnh của hệ thống có thể thay đổi theo thời gian. Mô hình này được sử dụng để xác định và cài đặt các khía cạnh điều khiển của một hệ thống. Các khái niệm đó là :

- Trạng thái
- Trạng thái con/ cha
- Sự kiện
- Hành động
- Hoạt động
- Mô hình chức năng mô tả việc chuyển đổi giá trị dữ liệu bên trong hệ thống. Các khái niệm đó là :
  - Xử lý
  - Kho dữ liệu
  - Dòng dữ liệu
  - Dòng điều khiển
  - Tác nhân (nguồn, đích)
- Phương pháp được phân chia thành bốn giai đoạn :
  - Phân tích : xây dựng một mô hình thể giới thực dựa vào việc mô tả vấn đề và yêu cầu hệ thống. Kết quả của giai đoạn này là :
    - Bản mô tả vấn đề
    - Mô hình đối tượng = sơ đồ lớp đối tượng + tự điển dữ liệu
    - Mô hình động = sơ đồ trạng thái + sơ đồ dòng sự kiện toàn cục
    - Mô hình chức năng = Sơ đồ dòng dữ liệu + các ràng buộc
  - Thiết kế hệ thống : phân chia hệ thống thành các hệ thống con dựa trên việc kết hợp kiến thức về lãnh vực vấn đề và kiến trúc đề xuất cho hệ thống. Kết quả của giai đoạn thiết kế là :
    - Sơ lược thiết kế hệ thống : kiến trúc hệ thống cơ sở và các quyết định chiến lược ở mức cao.
  - Thiết kế đối tượng : xây dựng một mô hình thiết kế dựa trên mô hình phân tích được làm giàu với các chi tiết cài đặt, bao gồm các lớp nền tảng các đối tượng cài đặt máy tính. Kết quả của giai đoạn này :
    - Mô hình đối tượng chi tiết
    - Mô hình động chi tiết
    - Mô hình chức năng chi tiết
  - Cài đặt : chuyển đổi các kết quả thiết kế vào một ngôn ngữ và phần cứng cụ thể. Đặc biệt nhấn mạnh trên các đặc điểm có thể truy vết, khả năng uyển chuyển và dễ mở rộng.

### **OOA (Object Oriented Analysis– Coad 90, 91)**

OOA sử dụng các nguyên lý cấu trúc hoá và kết hợp chúng với quan điểm hướng đối tượng tập trung vào giai đoạn phân tích. Phương pháp bao gồm năm bước :

- Tìm lớp và đối tượng : xác định cách thức tìm lớp và đối tượng. Tiếp cận đầu tiên bắt đầu với lãnh vực ứng dụng và xác định các lớp, các đối tượng hình thành nền tảng cho ứng dụng.
- Xác định cấu trúc : được thực hiện qua hai cách :
  - Xác định cấu trúc tổng quát hoá – chuyên biệt hoá và xác định sự phân cấp giữa các lớp đã tìm được

- Cấu trúc tổng thể - thành phần (whole – part) được dùng để mô hình hoá cách thức một đối tượng là một phần của đối tượng khác, và cách thức các đối tượng kết hợp thành các loại lớn hơn.
- Xác định các chủ đề : phân chia các mô hình lớp, đối tượng thành các đơn vị lớn hơn gọi là chủ đề.
- Xác định thuộc tính : xác định các thông tin và các mối liên kết cho mỗi thể hiện. Điều này bao gồm luôn việc xác định các thuộc tính cần thiết để đặc trưng hoá mỗi đối tượng. Các thuộc tính được tìm thấy sẽ được đưa vào đúng mức trong cấu trúc phân cấp.
- Xác định các dịch vụ : định nghĩa các toán tử cho lớp bằng cách xác định các trạng thái và các dịch vụ nhằm truy cập và thay đổi trạng thái đó.

Kết quả của giai đoạn phân tích là một mô hình gồm năm lớp:

- Lớp chủ đề
- Lớp các lớp và đối tượng
- Lớp cấu trúc (sự thừa kế, mối quan hệ,...)
- Lớp thuộc tính
- Lớp dịch vụ

Một mô hình thiết kế hướng đối tượng bao gồm các thành phần sau:

- Thành phần lãnh vực vấn đề (Problem Domain Component) : kết quả của phân tích hướng đối tượng đưa trực tiếp vào thành phần này.
- Thành phần tương tác (Human Interaction Component) : bao gồm các hoạt động như là : phân loại người dùng, mô tả kích bản nhiệm vụ, thiết kế cấu trúc lệnh, thiết kế tương tác chi tiết, lập bản mẫu giao diện tương tác người – máy, định nghĩa các lớp của thành phần tương tác.
- Thành phần quản lý nhiệm vụ (Task Management Component) : bao gồm việc xác định các nhiệm vụ (xử lý), các dịch vụ được cung cấp, mức độ ưu tiên, các sự kiện kích hoạt, và cách thức các xử lý trao đổi (với các xử lý khác và với bên ngoài hệ thống).
- Thành phần quản lý dữ liệu (Data Management Component) : phụ thuộc rất nhiều vào công nghệ lưu trữ sẵn có và dữ liệu yêu cầu.

## Chương 2

### CÁC KHÁI NIỆM CƠ BẢN VỀ HƯỚNG ĐỐI TƯỢNG

#### Đối tượng (object)

Đối tượng là thành phần trọng tâm của cách tiếp cận hướng đối tượng. Một đối tượng là một đại diện của bất kỳ sự vật nào cần được mô hình trong hệ thống và đóng một vai trò xác định trong lãnh vực ứng dụng.

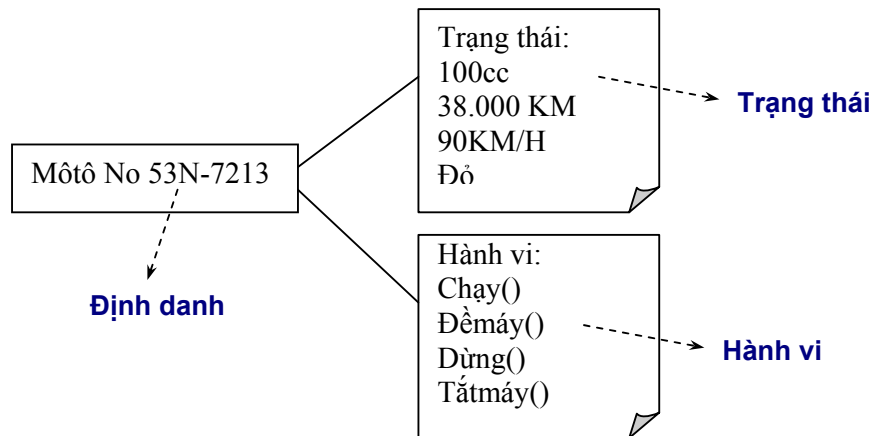
- Là một biểu diễn từ thế giới thực sang thể hiện của tin học (ví dụ : một chiếc xe ô tô trong thế giới thực được biểu diễn trong tin học dùng một khái niệm đối tượng xe ô tô).
- Là một sự trừu tượng hoá, một khái niệm có ý nghĩa trong lãnh vực ứng dụng.
- Diễn đạt một thực thể vật lý, hoặc một thực thể quan niệm, hoặc một thực thể phần mềm.
- Đối tượng có thể là một thực thể hữu hình trực quan (ví dụ : một con người, một vị trí, một sự vật,...) hoặc một khái niệm, một sự kiện (ví dụ : phòng ban, bộ phận, kết hôn, đăng ký, ...).

Một thực thể phải thoả ba nguyên lý :

- Phân biệt (distinction): đơn vị duy nhất (định danh)
- Thường xuyên (permanence) : quá trình sống (trạng thái)
- Hoạt động (activity) : vai trò, hành vi

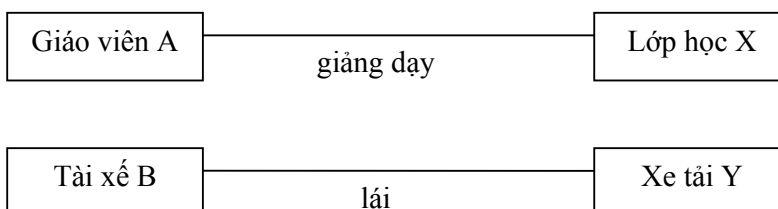
Đối tượng = định danh + trạng thái + hành vi

Ví dụ : một đối tượng xe mô tô



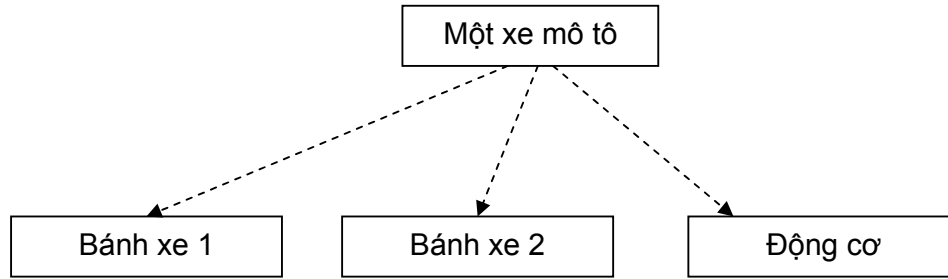
#### Liên kết giữa các đối tượng

- Mỗi kết hợp (association) – liên kết ngữ nghĩa :





- Phân cấp (hierarchy) – liên kết cấu trúc :

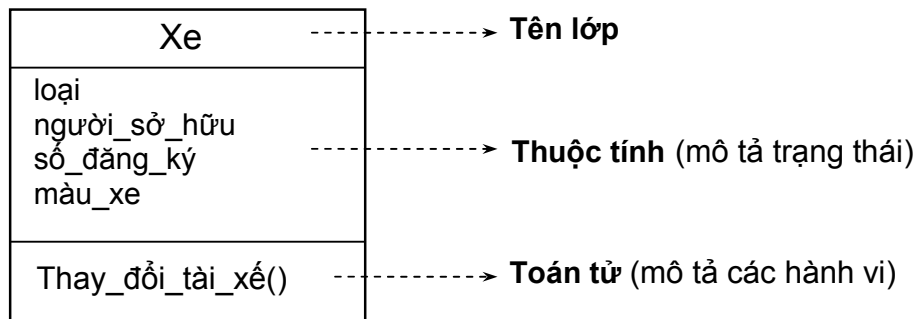


Đối tượng persistent/ transient

- Đối tượng transient : là đối tượng có quá trình sống tối đa tương ứng với quá trình chạy ứng dụng (đối tượng không được lưu trữ trạng thái)
- Đối tượng persistent : đối tượng có trạng thái được lưu trữ trong máy tính và có thể được thực thi bởi một ứng dụng khác ứng dụng tạo ra nó (quá trình sống của nó kéo dài và có thể từ ứng dụng này qua ứng dụng khác do trạng thái của nó được lưu trữ trong máy tính). Thông thường, trạng thái của đối tượng này sẽ được lưu trữ vào cơ sở dữ liệu trong quá trình sử dụng, và việc lưu trữ này sẽ duy trì được tình trạng của đối tượng và cung cấp tình trạng này cho những lần thực thi khác của ứng dụng hoặc cung cấp trạng thái của đối tượng cho những ứng dụng khác.

### Lớp (class)

- Một lớp là một mô tả của một tập hợp/ một loại các đối tượng có :
  - o Cùng cấu trúc (định danh, đặc trưng)
  - o Cùng hành vi (trạng thái, vai trò)
- Trình bày của lớp : là một hình chữ nhật bao gồm ba phần (không bắt buộc)



- Trong giai đoạn cài đặt, định danh của lớp được cài đặt từ một khoá. Khoá này cho phép phân biệt rõ các đối tượng của lớp một cách duy nhất. Khái niệm khoá có thể cho phép truy cập bởi người dùng một cách tường minh hoặc ngầm định. Một khoá tường minh có thể được khai báo chung với trạng thái của lớp trong khi đó khái niệm định danh là một khái niệm độc lập, và có các ý nghĩa sau :
  - o Xác định tính duy nhất của đối tượng
  - o Có ý nghĩa sử dụng đối với người dùng

Ví dụ : trong lớp Xe có thể khai báo số\_đăng\_ký là một khoá.

### Thể hiện của lớp (instance)

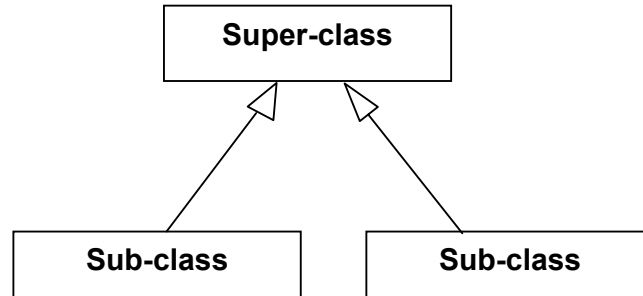
Thể hiện của lớp là một đối tượng cụ thể được tạo ra trên mô hình lớp :

- Các toán tử của lớp mô tả các hành vi chung của các thể hiện
- Tất cả các thể hiện của một lớp có chung các thuộc tính

## Phân cấp (hierarchy)

Là cơ chế hỗ trợ việc tổng quát hoá theo cách thức sau :

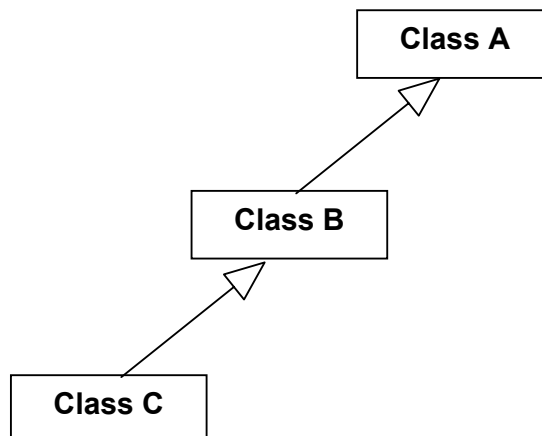
- Tổng quát hoá các đặc tính chung (định nghĩa các supper-class)
- Định nghĩa các đặc tính chuyên biệt nhất của các trường hợp cụ thể (định nghĩa các sub-class)



- Tổng quát hoá (generalisation) : xây dựng một lớp tổng quát từ các lớp khác cụ thể để đạt được một mức độ trừu tượng hoá có thể.
- Chuyên biệt hoá (specialisation) :
  - o Sự phân cấp của các lớp có phép mô tả các lớp chuyên biệt có thể từ các lớp trừu tượng
  - o Sự chuyên biệt hoá cũng có thể được tạo ra để :
    - Làm giàu thông tin : thêm mới thuộc tính hoặc toán tử vào lớp chuyên biệt so với các lớp trừu tượng
    - Có thể thay thế hoặc định nghĩa lại các thuộc tính, toán tử trong các lớp chuyên biệt từ thuộc tính, toán tử của các lớp trừu tượng

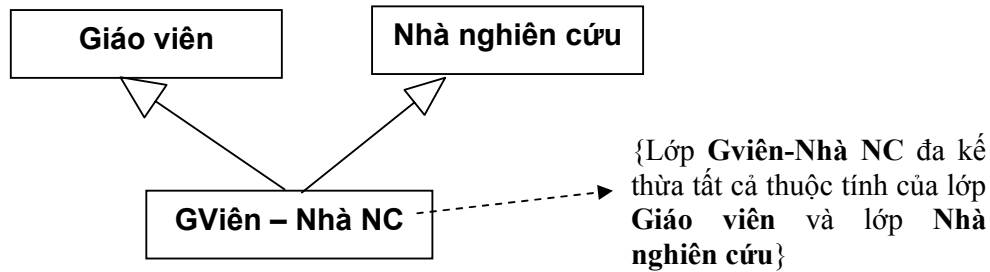
Trong quá trình phân tích hoặc thiết kế hệ thống hướng đối tượng, việc chuyên biệt hoá và tổng quát hoá cho phép định nghĩa các mối quan hệ tập con và làm sáng tỏ tính thừa kế. (Jacobson 1992). Nếu một lớp B thừa kế từ một lớp A, thì có nghĩa rằng tất cả các toán tử và các thuộc tính của lớp A trở thành toán tử và thuộc tính của lớp B.

- Quan hệ kế thừa là quan hệ:
  - o Có tính bắc cầu



{Nếu lớp B là một chuyên biệt hoá của lớp A và lớp C là một chuyên biệt hoá của lớp B thì lớp C cũng là một chuyên biệt hoá của lớp A}

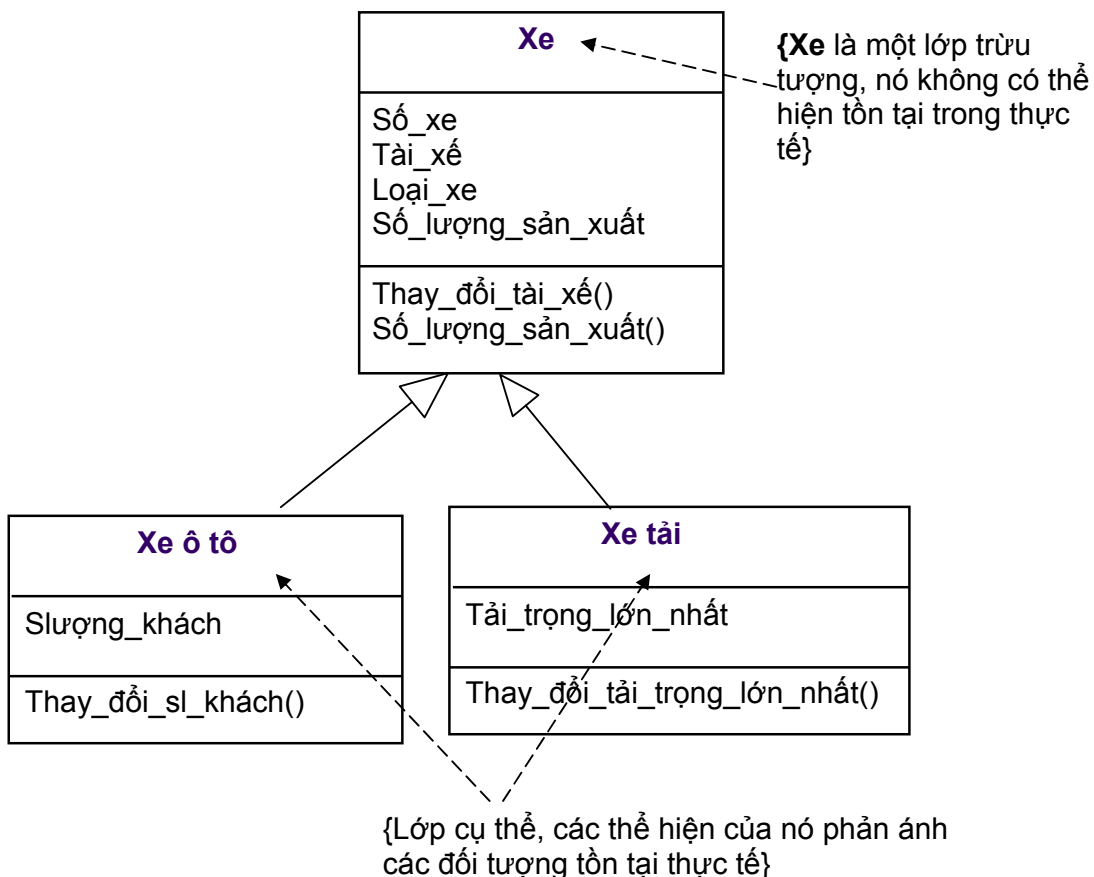
- o Có thể đa kế thừa



- Và được cài đặt với mục tiêu tái sử dụng

- Khái niệm lớp trừu tượng (abstract) – lớp cụ thể (concrete) :

Khi xây dựng một cấu trúc lớp phân cấp, chúng đã hình thành các lớp tổng quát và được gọi là lớp trừu tượng. Trong đó, tất cả các thể hiện đối tượng của một lớp trừu tượng đều xuất phát từ một trong những lớp cụ thể của nó. Một lớp trừu tượng không chứa đựng trực tiếp các đối tượng, các thể hiện của nó chỉ là sự xác định trừu tượng hơn của các thể hiện đối tượng trong các lớp cụ thể. Ngược lại, một lớp cụ thể thực sự chứa đựng các đối tượng và thể hiện. Trong ví dụ dưới đây, các lớp Xe tải và Xe ô tô là các lớp cụ thể bởi vì nó sẽ có các thể hiện xe ô tô và xe tải.



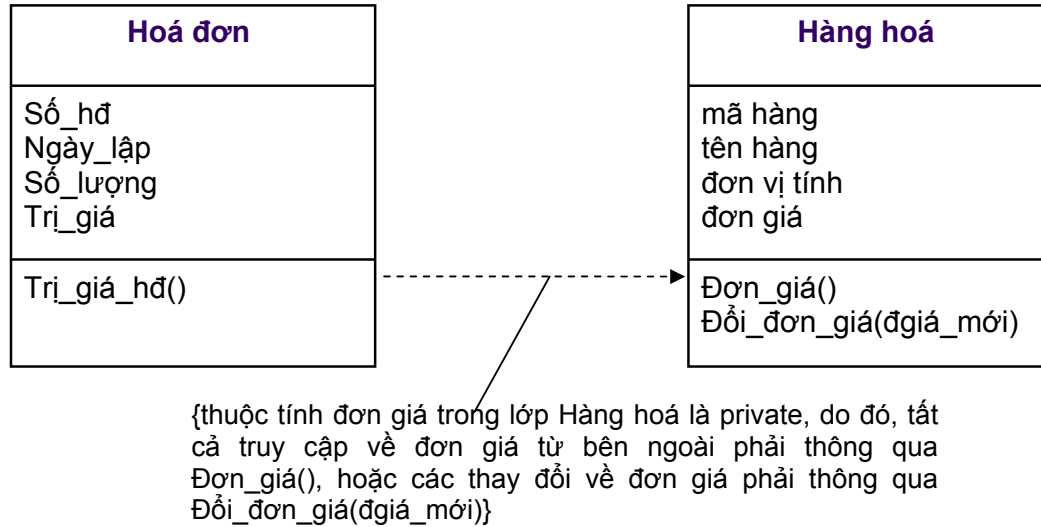
### Tính bao bọc(encapsulation)

Che dấu thông tin là nguyên lý để che dấu những dữ liệu và thủ tục bên trong của một đối tượng và cung cấp một giao diện tới mỗi đối tượng như là một cách để tiết lộ ít nhất có thể được về nội dung bên trong của đối tượng.

Các cơ chế bao bọc đối tượng tổng quan bao gồm : public, private, và protected

- public : thuộc tính và hành vi của đối tượng có thể được truy cập từ mọi nơi
- private : thuộc tính và hành vi của đối tượng chỉ được bên trong lớp
- protected : thuộc tính và hành vi của đối tượng chỉ được truy cập từ các lớp con

Tính bao bọc là một mục tiêu trong thiết kế hướng đối tượng. Thay vì cho phép một đối tượng truy cập trực tiếp đến dữ liệu của một đối tượng khác, thì đối tượng này sẽ yêu cầu dữ liệu đó thông qua việc gọi thì hành một hành vi đã được thiết kế cho việc cung cấp dữ liệu và một thông điệp sẽ được gửi tới đối tượng đích thông tin được yêu cầu. Điều này không chỉ đảm bảo rằng các lệnh đang hoạt động trong dữ liệu đúng mà còn không cho phép các đối tượng có thể thao tác trực tiếp lên dữ liệu của đối tượng khác.



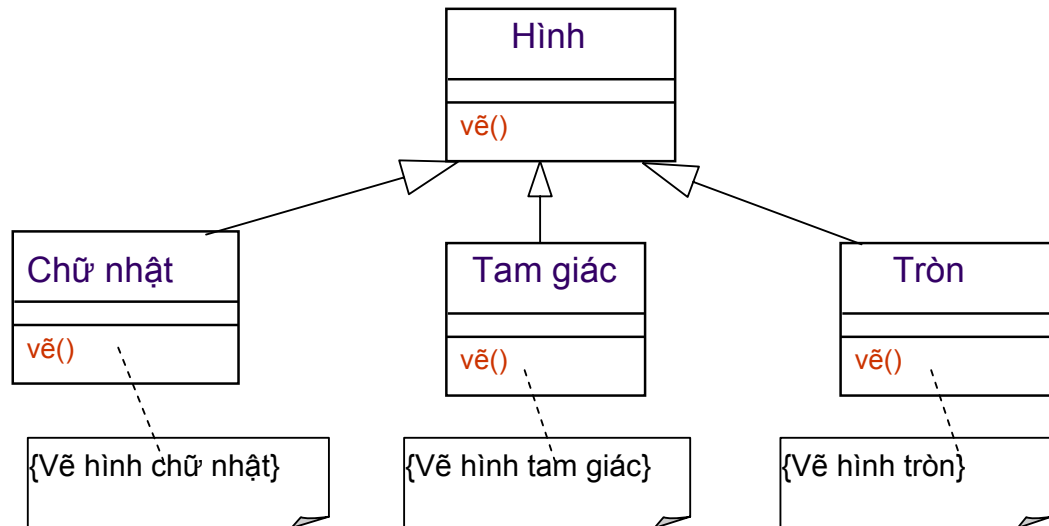
Một yếu tố quan trọng của tính bao bọc là việc thiết kế khác nhau của các đối tượng có thể sử dụng một phương thức (protocol) chung hoặc giao diện chung cho người dùng đối tượng. Điều này lý giải rằng nhiều đối tượng sẽ trả lời tới cùng thông điệp nhưng mỗi đối tượng sẽ thì hành thông điệp sử dụng các toán tử đã được biến đổi thích ứng tới lớp của nó. Bằng cách này, một chương trình có thể gửi một thông điệp tổng quát và để lại việc cài đặt cho đối tượng nhận. Điều này làm giảm sự phụ thuộc lẫn nhau và gia tăng số lượng trao đổi và tái sử dụng của đối tượng.

Ví dụ : các động cơ xe ô tô có thể khác nhau về cách cài đặt và vận hành cụ thể, giao diện giữa tài xế và xe ô tô là thông qua một phương thức chung : ví dụ, đạp cần gas để tăng lực và nhả cần gas để giảm lực của xe. Tất cả tài xế đều biết phương thức này và tất cả tài xế đều sử dụng phương thức này trong tất cả xe ô tô mà không qua tâm đến động cơ của xe ô tô được thực hiện như thế nào (tất nhiên, các động cơ khác nhau thì có cách vận hành khác nhau).

### Tính đa hình (polymorphism)

Trong hệ thống hướng đối tượng, thuật ngữ đa hình dùng để mô tả các đối tượng có nhiều dạng thức. Đa hình có nghĩa rằng cùng một toán tử có thể xử lý một cách khác nhau trong các lớp khác nhau có chung một (vài) lớp cha (superclass).

- Cùng toán tử có thể thì hành khác nhau trong các lớp khác nhau.
- Các phương thức khác nhau cùng cài đặt cho toán tử này trong các lớp khác nhau phải có cùng ký hiệu (tên, tham số và giá trị trả về)
- Cài đặt của toán tử được xác định bởi lớp đối tượng mà được sử dụng trực tiếp



Cấu trúc phân cấp trên cho thấy, lớp Hình là lớp tổng quát chung cho các lớp : Chữ nhật, Tam giác, Tròn. Vì cả ba lớp này đều có thể vẽ, do đó, có thể xác định một phương thức vẽ() chung trong lớp Hình. Tuy nhiên, các đối tượng trong các lớp chuyên biệt có thể được thực hiện trong một cách thức có thể khác so với phương thức vẽ() chung. Do đó, mỗi lớp chuyên biệt có thể cài đặt lại phương thức vẽ() chồng lên phương thức vẽ() của lớp tổng quát.

Nếu hệ thống xử lý một danh sách các đối tượng của lớp Hình, thì hệ thống sẽ dò tìm và thực hiện phương thức vẽ() phù hợp cho mỗi đối tượng. Nếu đối tượng đó là của lớp con Chữ nhật thì nội dung cài đặt của phương thức vẽ() trong lớp Chữ nhật sẽ được thực thi. Tương tự cho đối tượng của lớp con Tam giác và Tròn.

Đa hình ở đây là cho phép có nhiều hình thức cài đặt của cùng một hành vi (phương thức). Hệ thống sẽ tự động thực hiện phương thức thích hợp cho mỗi đối tượng.

### **Câu hỏi và bài tập**

## Chương 3

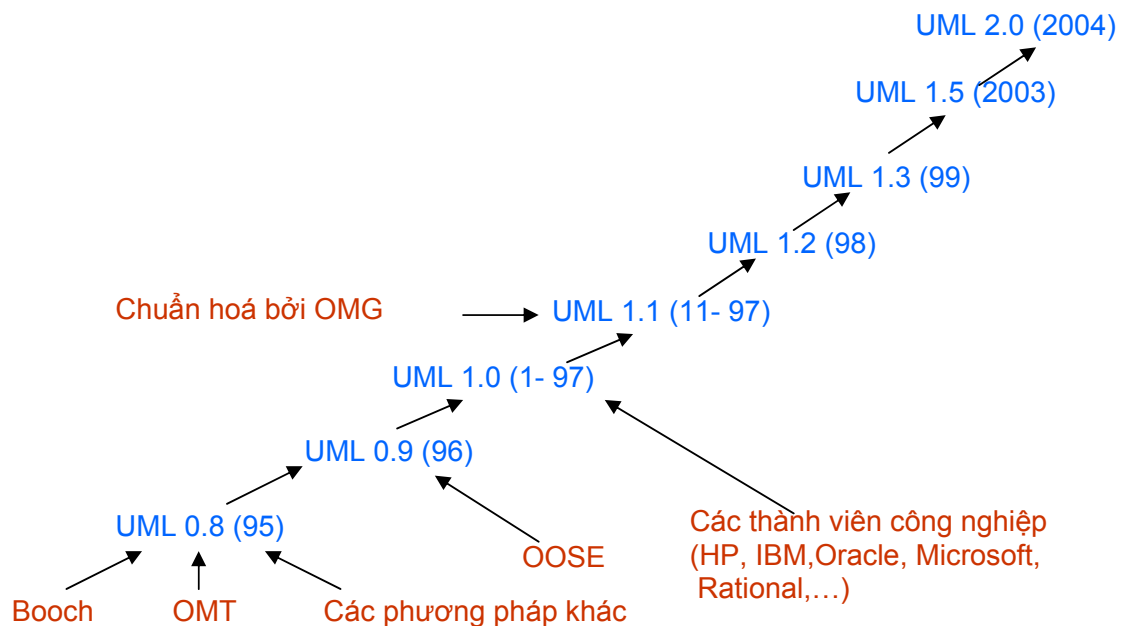
### UML (UNIFIED MODELING LANGUAGE)

#### Lịch sử của UML

- Số lượng các phương pháp luận hướng đối tượng gia tăng từ dưới 10 đến 50 trong khoảng những năm 1989 đến 1994, và do đó nảy sinh vấn đề là làm cho người phát triển khó tìm thấy một phương pháp luận duy nhất thoả mãn đầy đủ nhu cầu của họ.
- Vào tháng mười năm 1994, Rumbaugh đã liên kết với công ty Booch (Rational Software Corporation) để kết hợp phương pháp Booch và phương pháp OMT. Và cho ra một bản phác thảo về phương pháp có tên là Unified Process vào tháng mười năm 1995.
- Cũng trong năm 1995, Jacobson đã nỗ lực tích hợp phương pháp này với OOSE. Và những tài liệu đầu tiên về UML đã được trình làng vào trong năm 1996.
- Phiên bản 1.0 của UML đã được công bố vào tháng giêng 1997, bao gồm các công việc của các thành viên của UML consortium :

DEC	MCI Systemhouse
HP	Microsoft
i-Logix	Oracle
Intellicorp	Rational Software
IBM	TI
ICON	Computing Unisys

- Bản thảo về UML phiên bản 1.5 đã được tạo vào tháng ba năm 2003.
- Phiên bản UML 2.0 sẽ được tạo vào 2004.



1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
Rational	UML consortium	OMG							
U M L 0.8	U M L 0.9	U M L 1.0	U M L 1.1	U M L 1.2	U M L 1.3	U M L 1.4		U M L 1.5	U M L 2.0 ??

## UML ?

- UML được tạo ra nhằm chuẩn hoá ngôn ngữ mô hình hoá, UML không phải là một chuẩn về tiến trình và do đó, UML phải được sử dụng kết hợp với một tiến trình phương pháp luận.
- UML là một ngôn ngữ dùng để đặc tả, trực quan hoá, và tư liệu hoá phần mềm hướng đối tượng. Nó không mô tả một tiến trình hay một phương pháp mà trong đó chúng ta dùng nó để mô hình hoá. Ví dụ : Công ty Rational Software đề xuất một quy trình RUP (Rational Unified Process) được xem như là một phương pháp luận phát triển hệ thống và có ngôn ngữ mô hình hoá là UML.
- UML phủ tất cả các mức mô hình hoá khác nhau trong qui trình phát triển bao gồm chín loại sơ đồ, trong đó, năm sơ đồ dùng biểu diễn khía cạnh tĩnh và bốn sơ đồ biểu diễn khía cạnh động của hệ thống.

## Các đặc trưng của một tiến trình sử dụng UML

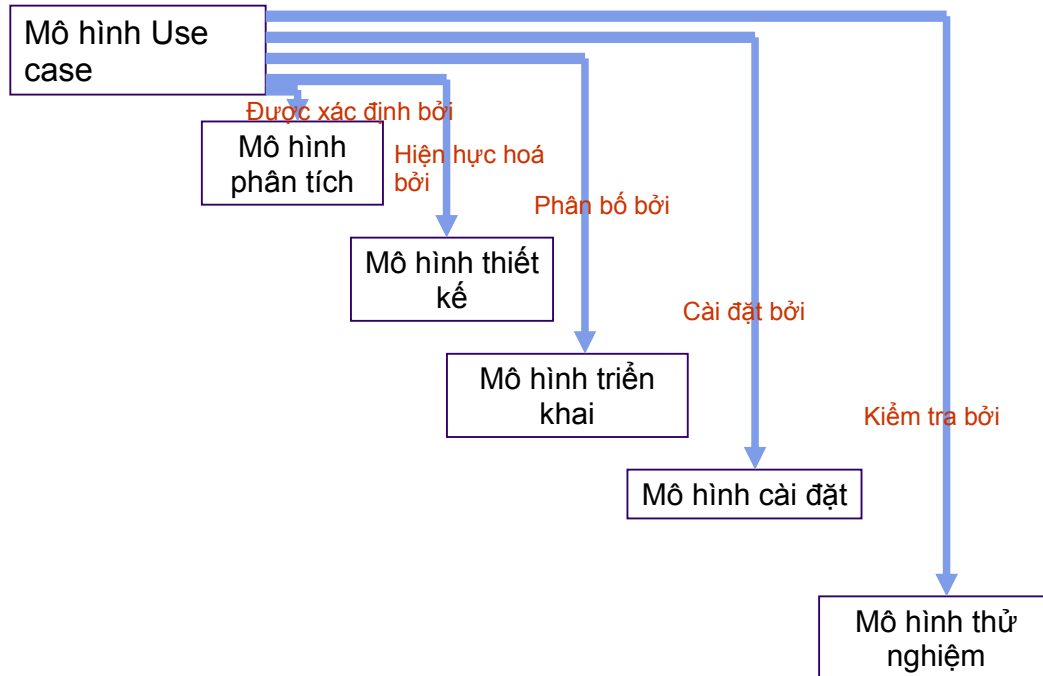
Thuở ban đầu, qui trình tuần tự được xem là phương pháp hợp lý nhất để phát triển hệ thống. Tuy nhiên, trải qua vài thập niên, đã cho thấy các dự án sử dụng qui trình tuần tự thường ít thành công, bởi những nguyên do sau đây:

- Sự giả định ban đầu có sai sót
- Thất bại trong việc kết hợp các nhân tố con người
- Các hệ thống ngày càng lớn và thường hay thay đổi
- Chúng ta vẫn còn đang trong giai đoạn thăm dò của công nghệ phần mềm, và không có nhiều kinh nghiệm. Đây là lý do chính.

Một phương pháp luận sử dụng UML phải kết hợp với một qui trình lặp và điều này sẽ làm giảm đi các hạn chế của qui trình tuần tự. Tính chất lặp gồm các đặc trưng cơ bản sau :

- Tính lặp (iterative)
  - o Thay vì nỗ lực xác định tất cả các chi tiết của mô hình trong một thời điểm, chúng ta chỉ xác định các chi tiết đã « đáp ứng » cho thời điểm đó để thực hiện, và
  - o Lặp lại một (hoặc nhiều) vòng lặp khác bổ sung thêm các chi tiết
- Gia tăng (incremental)

- Hệ thống tiến hoá thông qua một tập các sự gia tăng
- Mỗi sự gia tăng sẽ bù đắp thêm vào hệ thống các tính năng khác
- Tập trung vào người dùng (user – concentrated)
  - Phân tích viên xác định các tính năng của hệ thống thông qua các use case
  - Người dùng xác nhận các use case này
  - Thiết kế viên và người phát triển hiện thực hoá các use case
  - Người thử nghiệm kiểm tra hệ thống về việc thoả mãn các use case được đặt ra.



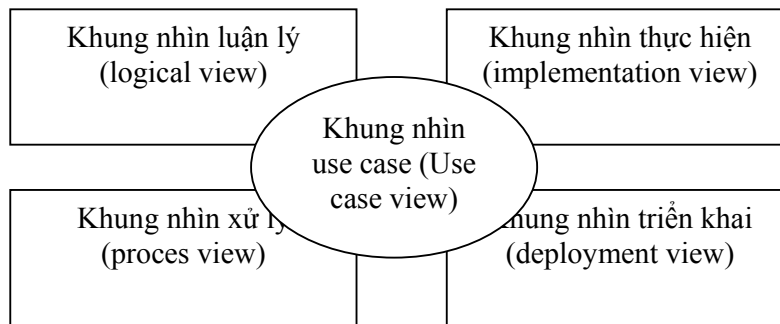
- Hướng kiến trúc (well-defined structure)
  - Hệ thống được phân chia thành các hệ thống con
  - Mức luận lý và vật lý phải được xác lập một cách tách biệt trong hệ thống
- Các khung nhìn về hệ thống :
  - Khung nhìn luận lý (logical view): mô tả các yêu cầu chức năng của hệ thống, tức những gì hệ thống nên làm cho người dùng cuối. Đó là sự trừu tượng của mô hình thiết kế và xác định các gói thiết kế chính, các subsystem và lớp chính. Trong UML khung nhìn này có thể được trình bày dùng sơ đồ lớp, sơ đồ đối tượng, sơ đồ mô tả các gói, hệ thống con.
  - Khung nhìn thực hiện (implementation view): mô tả tổ chức của các đơn thể (module) phần mềm tĩnh (như mã nguồn, tập tin dữ liệu, thành phần, tập tin thực thi, và các thành phần kèm theo khác) trong môi trường phát triển. Trong UML khung nhìn này có thể dùng sơ đồ thành phần để trình bày.
  - Khung nhìn xử lý (process view): mô tả các khía cạnh xảy ra đồng thời của hệ thống thời gian thực (run-time) (tasks, threads, processes cũng như sự tương tác giữa chúng). Khung nhìn này tập trung vào sự đồng hành, song song, khởi động và đóng hệ thống, khả năng chịu đựng hư hỏng, và sự phân tán các đối tượng.



- Khung nhìn triển khai (deployment): cho thấy các tập tin thực thi và các thành phần khác nhau được triển khai trên các hệ thống như thế nào. Nó giải quyết các vấn đề như triển khai, cài đặt, và tốc độ. Trong UML, khung nhìn này có thể sử dụng sơ đồ triển khai để mô tả.
- Khung nhìn use-case: đóng một vai trò đặc biệt đối với kiến trúc. Nó chứa một vài kịch bản hay use case chủ yếu. Ban đầu, chúng được dùng để khám phá và thiết kế kiến trúc trong các giai đoạn bắt đầu và giai đoạn đặc tả, nhưng sau đó chúng sẽ được dùng để xác nhận các khung nhìn khác nhau. Trong UML, khung nhìn này có thể sử dụng sơ đồ use case để minh họa.

**Người dùng**  
Chức năng

**Lập trình viên**  
Quản trị phần mềm



**Quản trị viên tích hợp hệ thống**  
Hiệu năng  
Tính co giãn  
Thông lượng

**Thiết kế viên hệ thống**  
Hình thái hệ thống  
Chuyên giao, cài đặt  
Truyền thông

Cần phân biệt khung nhìn kiến trúc với mô hình: mô hình là sự trình bày hoàn chỉnh về hệ thống, trong khi khung nhìn kiến trúc chỉ tập trung vào những gì có ý nghĩa về mặt kiến trúc, tức là những gì có tác động rộng lớn đến cấu trúc của hệ thống và lên tốc độ, sự hoàn thiện, tính tiên hóa của nó.

### Các sơ đồ trong UML

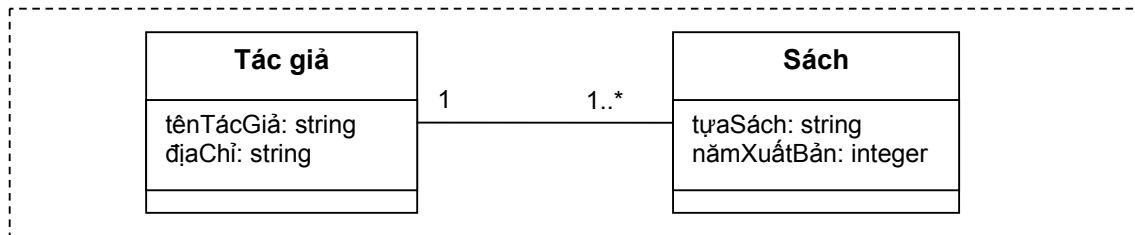
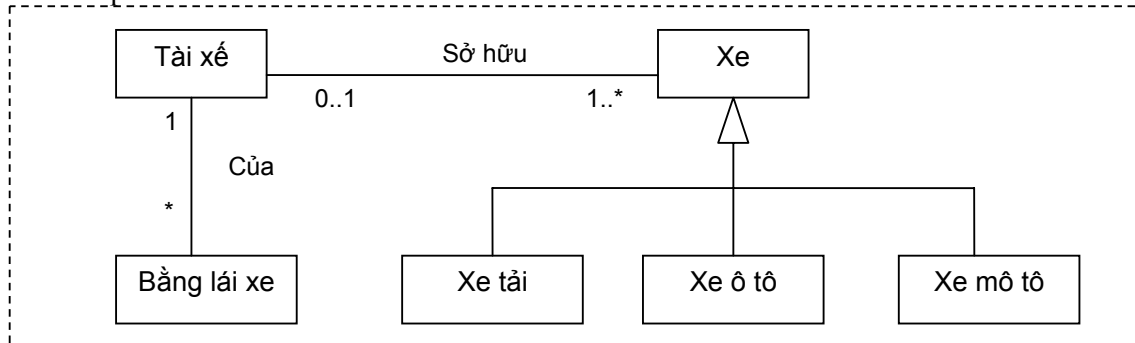
- Các sơ đồ mô tả khía cạnh tĩnh
  - Sơ đồ đối tượng (object diagram)
  - Sơ đồ lớp (class diagram)
  - Sơ đồ use case (use case diagram)
  - Sơ đồ thành phần (component diagram)
  - Sơ đồ triển khai (deployment diagram)
- Các sơ đồ mô tả khía cạnh động
  - Các sơ đồ tương tác (interaction diagram)
    - Sơ đồ tuần tự (sequence diagram)
    - Sơ đồ hợp tác (collaboration diagram)
  - Sơ đồ hoạt động (activity diagram)
  - Sơ đồ chuyển dịch trạng thái (state transition diagram)

**Sơ đồ lớp và đối tượng:** được sử dụng để mô hình hoá cấu trúc tĩnh của hệ thống trong quá trình phát triển. Mỗi sơ đồ chứa đựng các lớp và các mối quan hệ giữa chúng (quan hệ kế thừa (heritage), quan hệ kết hợp (association), quan hệ tập hợp (aggregation), quan hệ thành phần (composition)). Chúng ta cũng có thể mô tả các hoạt động của lớp (operation).

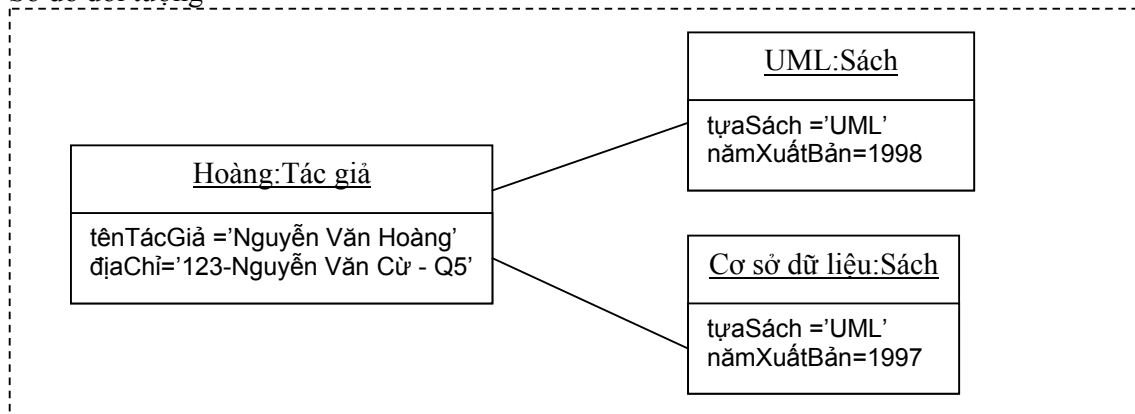
Sơ đồ đối tượng là một thể hiện của sơ đồ lớp. Nó mô tả trạng thái chi tiết của hệ thống tại một thời điểm cụ thể và là bức tranh của hệ thống tại một thời điểm, do đó, biểu đồ đối tượng được dùng để minh hoạ một trường hợp thực tế của sơ đồ lớp. Sơ đồ đối tượng có cùng ký hiệu với biểu đồ lớp. Sơ đồ đối tượng được dùng để minh hoạ một trường hợp phức tạp của bức tranh thực tế về hệ thống trong các thể hiện cụ thể.

Ví dụ :

Sơ đồ lớp

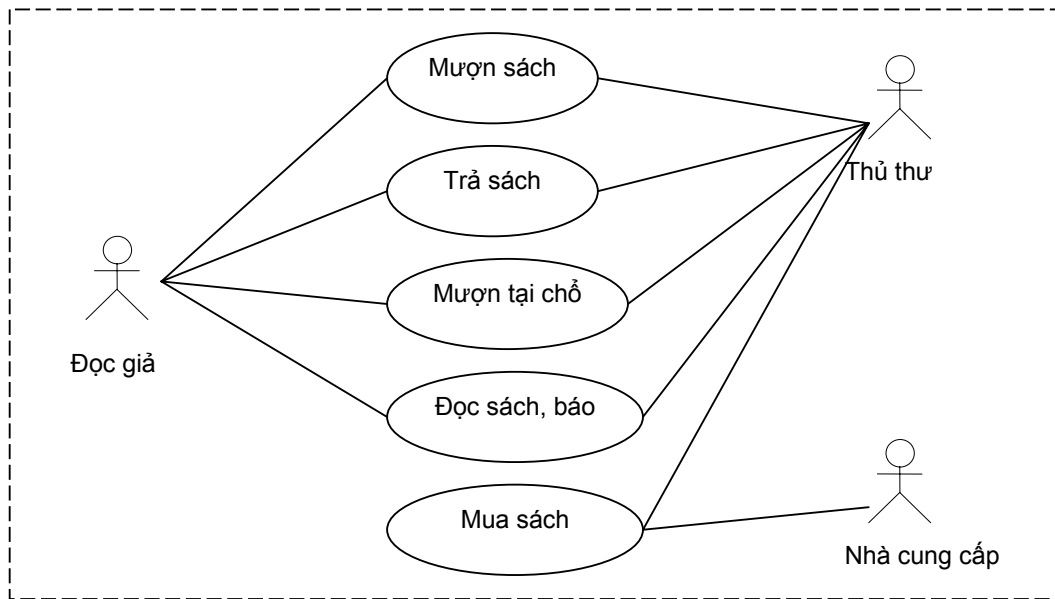


Sơ đồ đối tượng

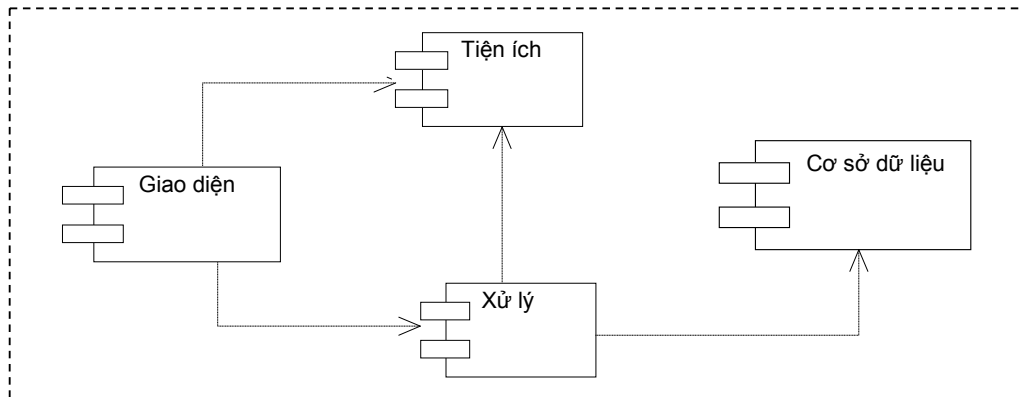


**Sơ đồ use case :** xuất phát từ các mô hình use case của phương pháp OOSE (Jacobson). Nó mô tả giao diện với một hệ thống từ quan điểm và cách nhìn của người sử dụng. Một sơ đồ use case mô tả các tình huống tiêu biểu của việc sử dụng một hệ thống. Nó biểu thị các trường hợp sử dụng (trong việc mô hình hoá các tính năng hệ thống) và các tác nhân (trong việc mô hình hoá các vai trò tham gia bởi các cá nhân tương tác với hệ thống), và mối quan hệ giữa các use case và các tác nhân.

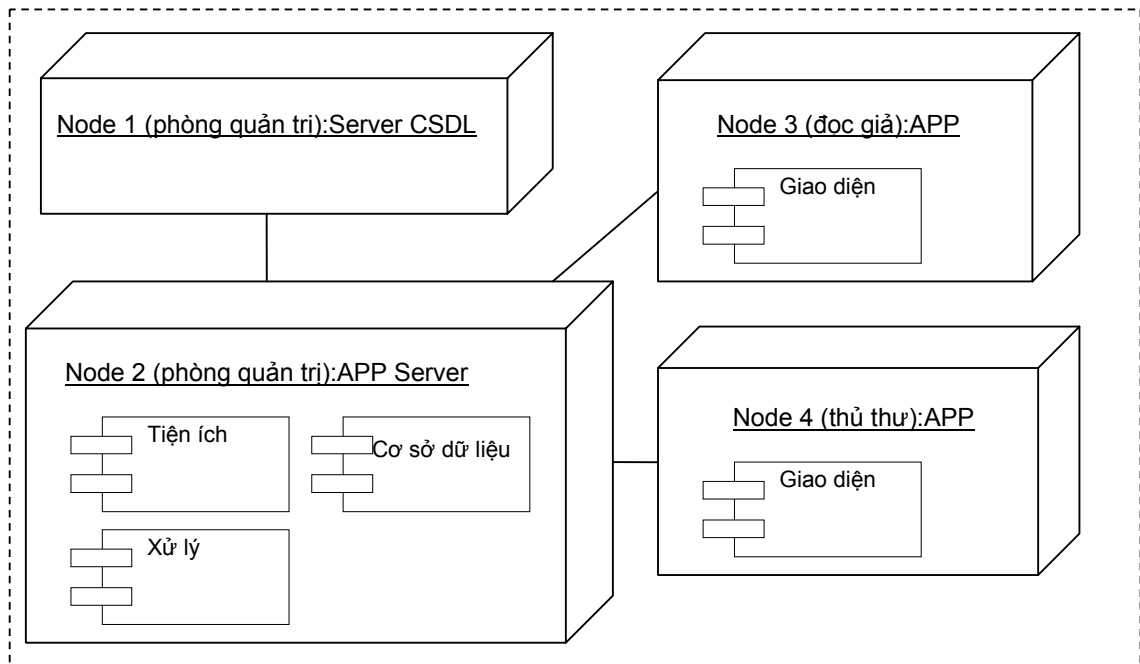
Ví dụ : sơ đồ use case một hệ thống quản lý một thư viện



**Sơ đồ thành phần** : được sử dụng để biểu thị các nhìn tĩnh trong việc cài đặt một hệ thống. Mỗi sơ đồ bao gồm các thành phần (component) và các mối quan hệ phụ thuộc giữa chúng trong môi trường cài đặt. Một thành phần đại diện cho một yếu tố cài đặt vật lý của môi trường (mã nguồn, mã thực thi, tập tin, cơ sở dữ liệu, một thư viện hàm,...). Ví dụ : sơ đồ thành phần của một hệ thống phần mềm quản lý thư viện

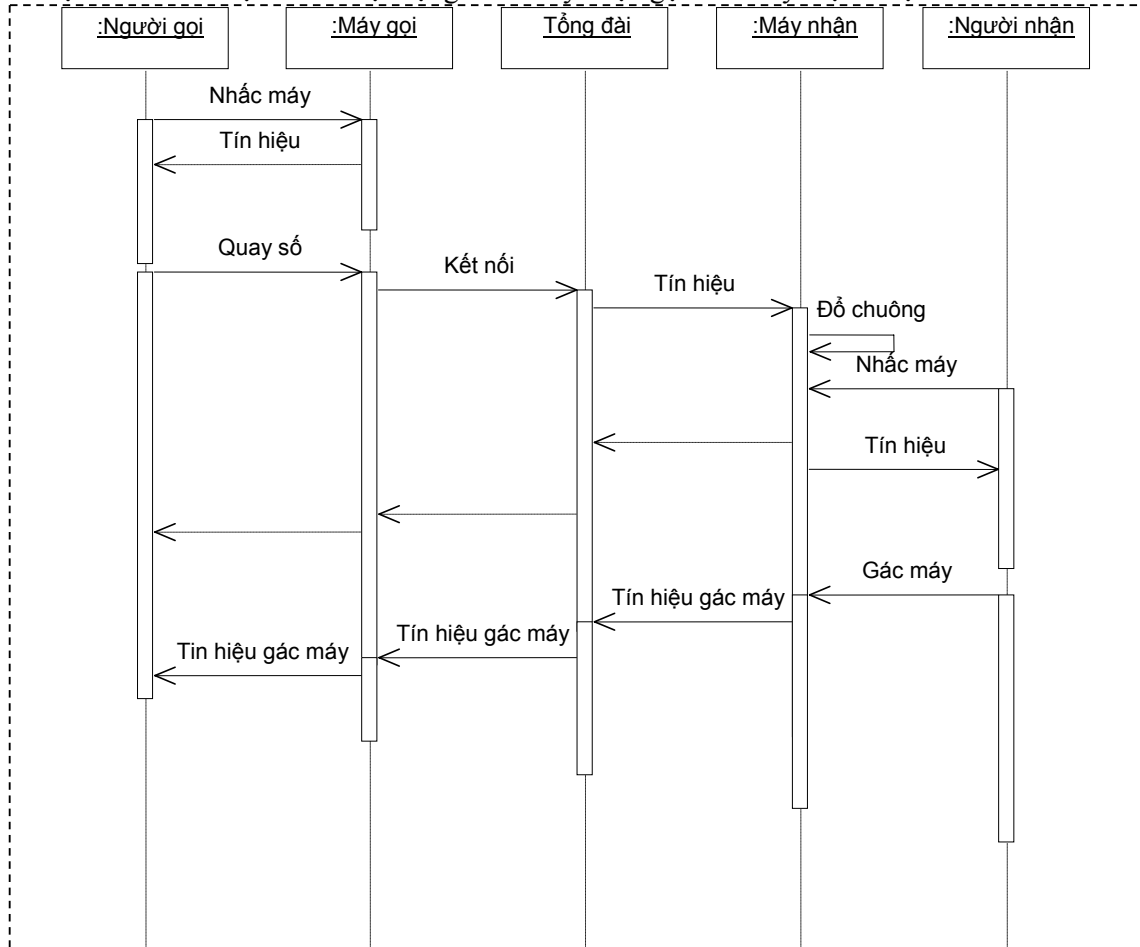


**Sơ đồ triển khai** : mô tả cách bố trí vật lý các thiết bị và sự phân phối các thành phần trú ngụ tại các thiết bị này. Một sơ đồ triển khai bao gồm các nút (node) đại diện cho các tài nguyên thiết bị và các thành phần được cài đặt trong thiết bị, các liên kết trong sơ đồ dùng để mô tả sự trao đổi giữa các nút. Sơ đồ triển khai biểu thị một sự tương ứng giữa cấu trúc phần mềm của một hệ thống và kiến trúc về bố trí thiết bị của nó. Ví dụ : sơ đồ triển khai của hệ thống quản lý thư viện



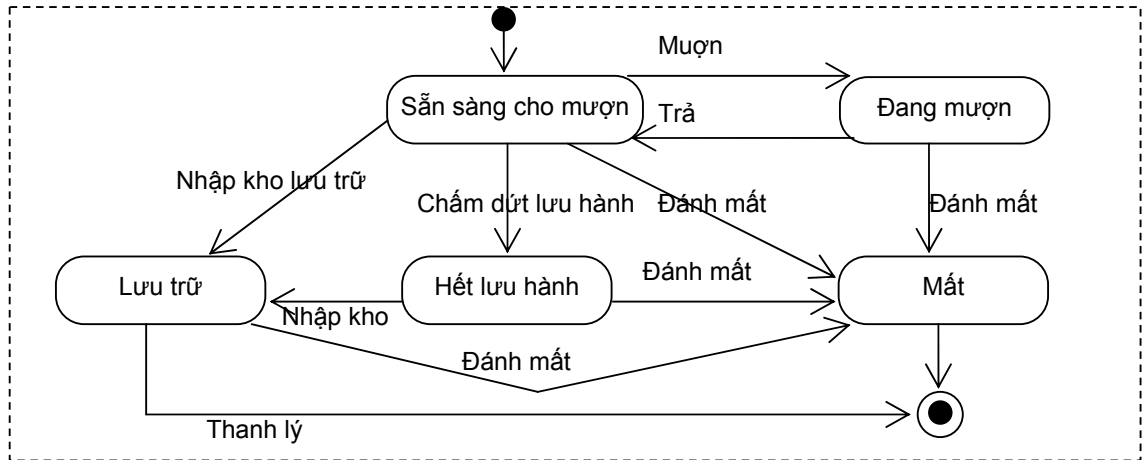
**Sơ đồ tuần tự và sơ đồ hợp tác** : trình bày các cách nhìn động về tương tác giữa các đối tượng của hệ thống trong quá trình phát triển. Sơ đồ hợp tác mô tả sự hợp tác giữa một nhóm các đối tượng trong hoạt động để đạt một mục tiêu cụ thể. Sơ đồ tuần tự thêm vào chiều thời gian nhằm thể hiện trực quan thứ tự trao đổi của các thông điệp (message).

Ví dụ : sơ đồ tuần tự mô tả hoạt động của xử lý cuộc gọi của máy điện thoại



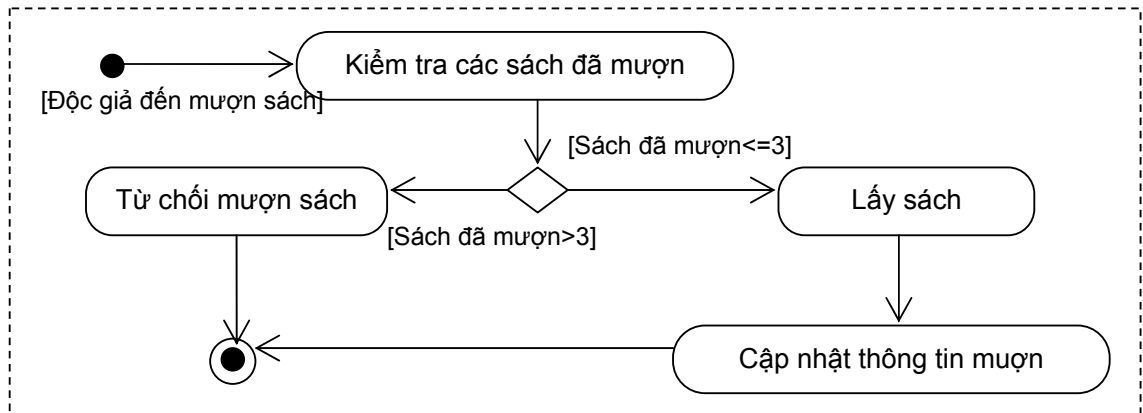
**Sơ đồ chuyển đổi trạng thái** : hình thành từ phương pháp OMT và Booch. Mỗi sơ đồ được dùng có liên quan đến một lớp để biểu thị các trạng thái khác nhau của đối tượng của lớp và các biến cố kích hoạt sự chuyển dịch giữa các trạng thái.

Ví dụ : sơ đồ trạng thái của sách trong thư viện



**Sơ đồ hoạt động** : dùng để mô hình hoá các dòng hoạt động liên kết tới các lớp như là trong trường hợp của một nhóm các lớp hợp tác cùng thực hiện trong một loại tiến trình. Mỗi lớp sẽ đảm nhiệm các hoạt động và các chuyển dịch như được mô tả trong sơ đồ chuyển dịch trạng thái. Tuy nhiên, một sơ đồ hoạt động có thể liên quan đến nhiều lớp hơn là một lớp. Mặt khác nó mô tả tiến trình tuần tự các hoạt động, sự đồng bộ hoá các dòng điều khiển song song, các điều kiện và quyết định, điểm bắt đầu và điểm kết thúc tiến trình.

Ví dụ : sơ đồ hoạt động đơn giản của hoạt động mượn sách thư viện



## Các hệ thống sử dụng UML trong việc mô hình hoá

Việc sử dụng UML trong quá trình mô hình hoá có thể áp dụng trong các loại hệ thống sau :

**Hệ thống nghiệp vụ (business system)** : mô tả tài nguyên (nhân lực, tài lực, tài sản,...), mục tiêu, luồng công việc, các quy tắc, ràng buộc trong hoạt động sản xuất kinh doanh của doanh nghiệp. Ví dụ : các công ty sản xuất, cửa hàng kinh doanh, y tế, giáo dục, ...

**Hệ thống thông tin (informaton system)** : thu thập và lưu trữ, biến đổi dữ liệu nhằm cung cấp thông tin đáp ứng nhu cầu người nhận trong các tổ chức hoạt động nghiệp vụ. Hệ thống thông tin cũng được chia thành nhiều loại tùy thuộc vào quy mô và độ phức tạp : hệ thống thông tin tác nghiệp : là hệ thống chuyên xử lý việc thu thập và truy tìm thông tin trong môi trường hoạt động nghiệp vụ. Hệ thống thông tin quản lý : xử lý tổng hợp dữ liệu thông qua các thống kê báo cáo nhằm đáp ứng thông tin cho các nhà quản lý theo dõi tình hình hoạt động. Hệ thống thông tin chuyên gia, hệ hỗ trợ ra quyết định : xử lý và tri thức hoá các dữ liệu hiện

tại nhằm đáp ứng các nhu cầu nâng cao về mặt thông tin như là hỗ trợ giải đáp tự động, hỗ trợ ra quyết định, dự báo tình hình tương lai,...

**Phần mềm hệ thống (System software)** : xây dựng các công cụ phần mềm cơ sở cho các phần mềm khác sử dụng như là hệ điều hành, hệ quản trị cơ sở dữ liệu, công cụ phát triển,....

**Hệ thống nhúng (embeded system)** : là một loại hệ thống phần mềm được xây dựng gắn trên một loại thiết bị như : điện thoại di động, thiết bị điều khiển,... Các hệ thống nhúng này thường được lập trình dùng ngôn ngữ cấp thấp hoặc chuyên dụng và có bộ nhớ lưu trữ cũng như màn hình.

**Hệ thống kỹ thuật (Technical system)** : xử lý và điều khiển các thiết bị kỹ thuật như hệ thống viễn thông, hệ thống quân sự, hay các quá trình xử lý kỹ thuật công nghiệp (dây chuyền sản xuất, vận hành máy móc,...). Đây là loại thiết bị phải xử lý các giao tiếp đặc biệt , không có phần mềm chuẩn và thường là các hệ thống thời gian thực (real time).

## PHẦN 2

### PHÂN TÍCH HỆ THỐNG

#### ***Mục tiêu***

Giúp cho người học nắm vững các nội dung về:

- Tiến trình phân tích hướng đối tượng
- Tiến trình, nội dung và các phương pháp khảo sát yêu cầu
- Phân tích chức năng hệ thống bằng mô hình hoá use case
- Hiểu về hệ thống nghiệp vụ và mô hình hoá hệ thống nghiệp vụ
- Phân loại và xác định đối tượng hệ thống bằng việc xây dựng sơ đồ lớp

#### ***Giới thiệu***

Bước đầu tiên để tìm ra một giải pháp thích hợp cho vấn đề hệ thống là hiểu vấn đề và lãnh vực của hệ thống đó. Mục tiêu chính của phân tích là nắm bắt một hình ảnh đầy đủ, không mơ hồ, và nhất quán về yêu cầu hệ thống và những gì hệ thống sẽ phải làm để đáp ứng đòi hỏi và nhu cầu người dùng. Điều này được thực hiện bằng cách xây dựng các mô hình của hệ thống với mục tiêu tập trung vào khía cạnh biểu diễn hệ thống về mặt nội dung (nghĩa là các mô hình tập trung vào làm rõ hệ thống có những gì) hơn là cách thức mà hệ thống thực hiện nội dung đó. Do đó, kết quả của giai đoạn phân tích là làm rõ các yếu tố hệ thống từ quan điểm và cách nhìn của người sử dụng mà không quan tâm đến cách thức chi tiết mà máy tính thực hiện nội dung đó.

Phân tích là một tiến trình chuyển đổi một định nghĩa vấn đề từ một tập mờ các sự kiện, các dự kiến mang tính tưởng tượng thành một diễn đạt chặt chẽ các yêu cầu hệ thống. Thực sự, phân tích là một hoạt động mang tính sáng tạo bao gồm việc hiểu vấn đề, các ràng buộc liên quan đến vấn đề đó và các phương pháp để khống chế hoặc giải quyết những ràng buộc. Đây là một tiến trình lặp cho đến khi các vấn đề phải được rõ ràng.

Phần này gồm ba chương bao gồm: xác định yêu cầu hệ thống; mô hình hoá use case; mô hình hoá nghiệp vụ; và xây dựng sơ đồ lớp.

## Chương 4

# XÁC ĐỊNH YÊU CẦU HỆ THỐNG

### **Mục tiêu**

Qua chương này, chúng ta có thể hiểu:

- Mục tiêu của việc khảo sát hệ thống
- Nội dung của việc khảo sát và các đối tượng để tiếp cận khảo sát
- Một số phương pháp khảo sát: phỏng vấn, bảng câu hỏi, phỏng vấn nhóm, phân tích tài liệu, thiết kế kết hợp người dùng, bản mẫu (prototype)
- Yêu cầu và việc phân loại các yêu cầu hệ thống

### **Mục đích khảo sát yêu cầu**

Khảo sát hệ thống là nhằm thu thập tốt nhất thông tin phản ánh về hệ thống hiện tại, để từ đó làm cơ sở cho việc phân tích và xây dựng hệ thống mới giải quyết tồn tại bất cập của hệ thống. Vậy khảo sát yêu cầu bao gồm những mục tiêu sau:

- Tiếp cận với nghiệp vụ chuyên môn, môi trường của hệ thống
- Tìm hiểu vai trò, chức năng, nhiệm vụ và cách thức hoạt động của hệ thống
- Nêu ra được các điểm hạn chế, bất cập của hệ thống cần phải thay đổi
- Đưa ra được những vấn đề của hệ thống cần phải được nghiên cứu thay đổi.

### **Nội dung khảo sát**

Nội dung khảo sát phải tìm hiểu được các nội dung của hệ thống như sau:

- Các mục tiêu hoạt động của đơn vị, các công việc và cách thức hoạt động để đạt những mục tiêu đó.
- Những thông tin cần để thực hiện công việc từng loại công việc
- Các nguồn dữ liệu (định nghĩa, cấu trúc, dung lượng, kích thước,...) bên trong và bên ngoài đơn vị. Có thể bao gồm:
  - o Các hồ sơ, sổ sách, tập tin
  - o Biểu mẫu, báo cáo, qui tắc, quy định, công thức.
  - o Các qui tắc, qui định ràng buộc lên dữ liệu
  - o Các sự kiện tác động lên dữ liệu
- Tìm hiểu khi nào, như thế nào, và bởi ai các dữ liệu đó được tạo ra, di chuyển, biến đổi và được lưu trữ. Ứng với mỗi xử lý thực hiện tìm hiểu:
  - o Phương pháp: cách thức thực hiện
  - o Tần suất: số lần thực hiện trong một đơn vị thời gian
  - o Khối lượng: độ lớn thông tin thực hiện
  - o Độ phức tạp: xử lý là một là một quá trình phức tạp liên quan đến nhiều loại dữ liệu hay chỉ là một tính toán đơn giản với một vài loại dữ liệu.
  - o Độ chính xác: độ chính xác của kết quả thực hiện
- Thứ tự và các phụ thuộc khác giữa các hoạt động truy xuất dữ liệu khác nhau



- Các chính sách, hướng dẫn mô tả hoạt động quản lý, thị trường và môi trường hệ thống
- Các phương tiện, tài nguyên có thể sử dụng
- Trình độ chuyên môn sử dụng và tính của các đối tượng xử lý thông tin hệ thống
- Môi trường hệ thống (kinh tế, xã hội, cơ quan chủ quản)
- Các đánh giá, phản nản về hệ thống hiện tại; các đề xuất giải quyết.

### ***Đối tượng khảo sát***

Có nhiều nguồn có thể cung cấp thông tin để đáp ứng nội dung khảo sát yêu cầu. Mỗi nguồn có một hình thức khác nhau do đó phải có một cách tiếp cận khảo sát khác nhau. Các đối tượng khảo sát đó là:

#### **Người dùng**

- *Các cán bộ lãnh đạo, cán bộ quản lý:* các đối tượng này sẽ giúp cho phân tích viên nắm bắt được tổng quan cấu trúc hệ thống, mục tiêu chung mà hệ thống mới mong muốn mang lại. Các thông tin mà đối tượng này mang lại thường là chiều rộng, mang tính tổng thể, chiến lược không mô tả chi tiết cách thức phải thực hiện.
- *Người sử dụng, nhân viên nghiệp vụ:* các đối tượng này sẽ cung cấp thông tin chi tiết cách thức mà họ đang thực hiện công việc gồm các bước cụ thể, các giấy tờ biểu mẫu liên quan. Các thông tin mà đối tượng mang lại thường là chiều sâu, chi tiết, cục bộ bỏ qua ý tưởng chiến lược mang tính tổng thể.
- *Nhân viên kỹ thuật:* các đối tượng này sẽ cung cấp thông tin về tình trạng công nghệ, trang thiết bị, phần mềm hiện hành đang sử dụng và khả năng, trình độ về kỹ thuật của họ. Các đối tượng này thường trợ giúp rất lớn trong việc huấn luyện, triển khai và bảo trì hệ thống mới.

#### **Tài liệu**

- *Tài liệu về sổ sách, biểu mẫu, tập tin:* nguồn cung cấp các thông tin về dữ liệu, luồng dữ liệu, giao dịch và xử lý giao dịch. Đặc biệt là các biểu mẫu đây chính là kết quả đầu ra của hệ thống.
- *Tài liệu về qui trình, thủ tục:* nguồn cung cấp thông tin về qui trình xử lý, vai trò xử lý của các nhân viên, chi tiết mô tả công việc của nhân viên, các qui định thủ tục.
- *Các thông báo:* các mẫu thông báo của hệ thống đối với môi trường ngoài, giữa các bộ phận trong hệ thống (ví dụ: thông báo hợp mặt khách hàng, thông báo mời thầu, thông báo từ chối đơn hàng, ... hoặc các thông báo nội bộ như là thông báo bổ nhiệm, thông báo nâng lương,...)

#### **Chương trình máy tính**

- Các chương trình phần mềm hệ thống đang sử dụng, các chương trình này giúp xác định được cấu trúc dữ liệu hệ thống, thói quen của người sử dụng, chức năng mà hệ thống mới chưa đáp ứng được, số liệu thử nghiệm hệ thống.

### ***Phương pháp xác định yêu cầu***

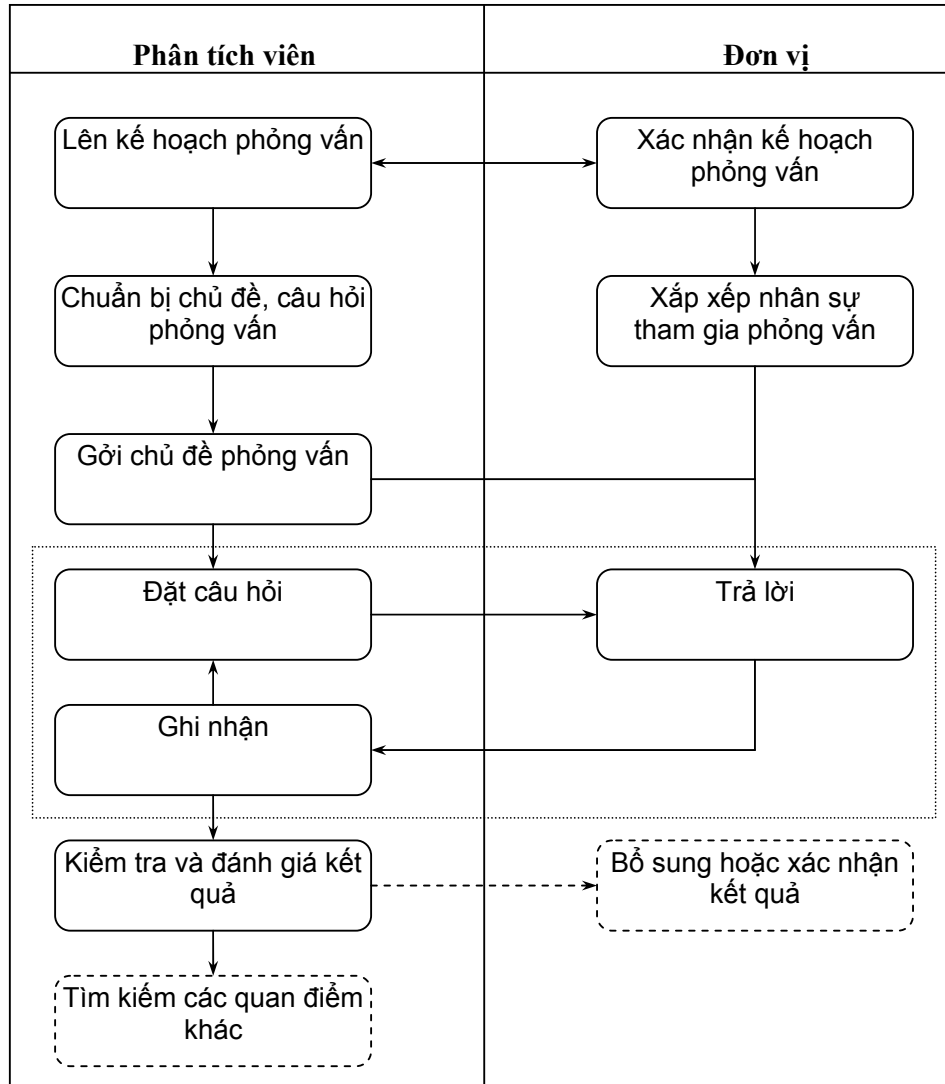
#### **Các phương pháp truyền thống xác định yêu cầu**

##### ***Phỏng vấn***

Phỏng vấn là một hình thức khảo sát thu thập thông tin trực tiếp từ các đối tượng sẽ sử dụng hệ thống. Vì mỗi người dùng sẽ có những hiểu biết nhất định về một phần công việc của mình trong hệ thống hiện tại và mong muốn hệ thống mới về những gì sẽ phục vụ và trợ giúp

cho công việc của họ. Ví dụ: một kế toán viên chi tiết thì biết được chi tiết các loại chứng từ, cách sắp xếp và xử lý chứng từ,... còn kế toán viên tổng hợp thì chỉ quan tâm đến những số liệu nào và cách thức để tổng hợp số liệu đó để tạo ra các báo cáo thống kê, tổng hợp,... Do đó, việc phỏng vấn phải được thực hiện trên nhiều người dùng khác nhau (ba loại người dùng) nhằm thu thập nhiều nhất yêu cầu hệ thống.

Phỏng vấn là một cách thức đối thoại trực tiếp trong đó, phân tích viên sẽ ra câu hỏi và đối tượng phỏng vấn sẽ trả lời câu hỏi. Quy trình các bước thực hiện như sau:



Hình 1. Sơ đồ mô phỏng quá trình phỏng vấn

Đầu tiên phân tích viên chuẩn bị một kế hoạch phỏng vấn tổng quát, kế hoạch này sẽ liệt kê tất cả các lãnh vực của hệ thống cần khảo sát và thời gian dự kiến cho từng lãnh vực. Mẫu kế hoạch như sau:

<b>Kế hoạch phỏng vấn tổng quan</b>				
Hệ thống: .....				
Người lập: .....			Ngày lập: ..../..../....	
STT	Chủ đề	Yêu cầu	Ngày bắt đầu	Ngày kết thúc



	<i>Kết quả quan sát:</i>
--	--------------------------

Ví dụ:

<i>Người được phỏng vấn: Trần Thị X...</i>	<i>Ngày: 05/08/2003</i>
<i>Câu hỏi</i>	<i>Ghi nhận</i>
<i>Câu hỏi 1:</i> Tất cả đơn hàng của khách hàng phải được thanh toán trước rồi mới giao hàng?	<i>Trả lời:</i> Phải thanh toán trước hoặc ngay khi giao. <i>Kết quả quan sát:</i> Thái độ không chắc chắn
<i>Câu hỏi 2:</i> Anh Chị muốn hệ thống mới sẽ giúp cho Anh Chị điều gì?	<i>Trả lời</i> Dữ liệu chỉ nhập một lần và các báo cáo tự động tính toán <i>Kết quả quan sát</i> Không tin tưởng lắm, hình như đã triển khai thất bại một lần

### **Loại câu hỏi phỏng vấn:**

**Câu hỏi mở:** là câu hỏi giúp cho việc trả lời được tự do trong phạm vi hệ thống. Kết quả trả lời không tuân theo một vài tình huống cố định. Mục đích của câu hỏi mở là khuyến khích người trả lời đưa ra được tất cả ý kiến có thể trong khuôn khổ câu hỏi. Do đó, câu hỏi mở dùng để thăm dò, gợi mở vấn đề và người trả lời phải có một kiến thức tương đối.

Ví dụ: “Anh (Chị) đang xử lý thông tin gì?” hoặc “Anh (Chị) có khó khăn gì khi quản lý dữ liệu của mình?”

**Câu hỏi đóng:** là câu hỏi mà sự trả lời là việc chọn lựa một hoặc nhiều trong những tình huống xác định trước. Do đó, câu hỏi đóng được dùng xác định một tình huống cụ thể.

Ví dụ: “Điều nào dưới đây là tốt nhất đối với HTTT Anh (Chị) đang sử dụng?”

- ☐ Dễ dàng truy cập đến tất cả dữ liệu cần
- ☐ Thời gian trả lời tốt nhất của hệ thống
- ☐ Khả năng chạy đồng thời với các ứng dụng khác

Câu hỏi đóng thường được thiết kế theo một trong những dạng sau:

- Đúng – sai
- Nhiều chọn lựa (có thể một trả lời hoặc trả lời tất cả chọn lựa)
- Tỉ lệ trả lời: từ xấu đến tốt, từ rất đồng ý đến hoàn toàn không đồng ý. Mỗi điểm trên tỉ lệ nên có một nghĩa rõ ràng và nhất quán và thường có một điểm trung lập ở giữa
- Xếp hạng các chọn lựa theo thứ tự mức độ quan trọng

**Xếp xếp câu hỏi:** thứ tự câu hỏi phải hợp lý, phù hợp với mục tiêu khảo sát và khả năng của người trả lời. Các thứ tự có thể là:

- Thu hẹp dần: ban đầu là những câu hỏi rộng, khái quát và càng về sau thì thu hẹp đến một mục tiêu.
- Mở rộng dần: ban đầu đề cập đến một điểm nào đó rồi mở rộng dần phạm vi đề cập

Câu hỏi mở	Câu hỏi đóng
<b>Ưu điểm</b> <ul style="list-style-type: none"> <li>- Không ràng buộc kết quả trả lời</li> <li>- Có thể phát sinh ý tưởng mới</li> </ul>	<ul style="list-style-type: none"> <li>- Thời gian trả lời ngắn</li> <li>- Nội dung trả lời tập trung, chi tiết</li> </ul>
<b>Khuyết điểm</b> <ul style="list-style-type: none"> <li>- Thời gian dễ kéo dài</li> <li>- Nội dung trả lời có thể vượt phạm vi câu hỏi</li> </ul>	<ul style="list-style-type: none"> <li>- Mất nhiều thời gian chuẩn bị câu hỏi</li> <li>- Không mở rộng được kết quả trả lời</li> </ul>

#### **Khảo sát dùng bảng câu hỏi (questionnaire)**

Phòng vấn là một phương pháp hiệu quả để trao đổi và thu thập được những thông tin quan trọng từ phía người dùng. Tuy nhiên, thực hiện phỏng vấn cũng rất tốn kém về thời gian và nguồn lực. Phương pháp khảo sát dùng bảng câu hỏi ít tốn kém hơn, thời gian trả lời lại nhanh hơn, kết quả xác định hơn và thu thập được thông tin từ nhiều đối tượng hơn trong cùng một thời gian ngắn. Tuy nhiên, phương pháp này lại thụ động và ít mang lại chiều sâu hơn phương pháp phỏng vấn.

Để thực hiện, các công việc phân tích viên cần phải làm rõ:

- Tập hợp câu hỏi thành từng nhóm
- Phân loại các đối tượng sử dụng thành nhóm và gởi nhóm câu hỏi nào đến nhóm người nào. Tổng quát, việc gộp nhóm sẽ được thực hiện bởi một hoặc sự kết hợp của bốn phương pháp sau:
  - o Đối tượng tích cực: đối tượng có vị trí thuận lợi, sẵn lòng để được khảo sát, hoặc những đối tượng có nhiều động lực trả lời nhất.
  - o Nhóm ngẫu nhiên: chọn ngẫu nhiên một nhóm người dùng trong danh sách để gởi câu hỏi.
  - o Theo chủ định: chọn những người thoả các tiêu chuẩn xác định nào đó. Ví dụ: những người đã làm việc với hệ thống 2 năm trở lên, những người thường xuyên sử dụng hệ thống,...
  - o Chọn theo loại: người dùng, quản lý, ...

Thông thường, người ta kết hợp các phương pháp lại. Trong bất kỳ trường hợp nào khi nhận được trả lời chúng ta nên kiểm tra lại các trường hợp không trả lời để tìm ra nguyên nhân và xem xét các kết quả trả lời là hợp lệ và đủ để được chấp nhận không.

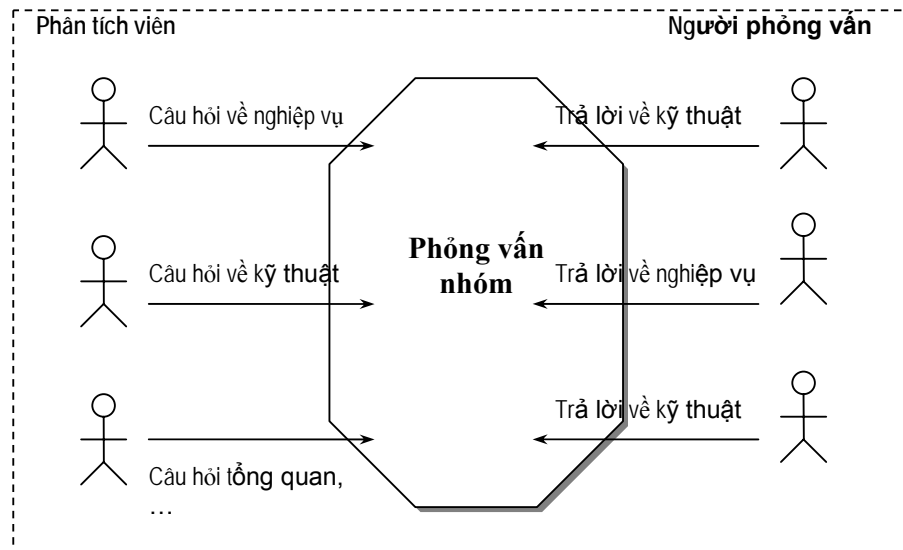
So sánh giữa phỏng vấn và bảng câu hỏi được liệt kê dưới đây

Đặc điểm	Phỏng vấn	Bảng câu hỏi
Sự phong phú thông tin	Cao (qua nhiều kênh: trả lời, cử chỉ,...)	Trung bình tới thấp (chỉ trả lời)
Thời gian	Có thể kéo dài	Thấp, vừa phải
Chi phí	Có thể cao	Vừa phải
Cơ hội nắm bắt và phát hiện	Tốt: việc phát hiện và chọn lọc các câu hỏi có thể được đặt ra bởi hoặc người phỏng vấn	Hạn chế: sau khi thu thập dữ liệu cơ sở

	hoặc người được phỏng vấn	
Tính bảo mật	Mọi người biết lẫn nhau	Không biết người trả lời
Vai trò tham gia	Người được phỏng vấn đóng một vai trò quan trọng và có thể quyết định kết quả	Trả lời thụ động, không chắc chắn quyết định kết quả

### Phỏng vấn nhóm

Các yêu cầu được thu thập có thể sử dụng phương pháp phỏng vấn hoặc điều tra dùng bảng câu hỏi. Tuy nhiên, các kết quả phỏng vấn các đối tượng khác nhau có thể dẫn đến sự không nhất quán thông tin về hệ thống hiện hành và yêu cầu về hệ thống mới. Do đó, chúng ta lại phải thực hiện việc kiểm tra, chọn lọc và quyết định chính xác đâu là thông tin đúng và được chấp nhận cuối cùng. Thông thường chúng ta tiếp tục thực hiện các trao đổi và gặp gỡ các nhân vật quan trọng có thể quyết định định và giới hạn được kết quả thông tin. Các cuộc phỏng vấn mới này thường tốn thời gian và có khi lại trả lời lại các câu hỏi mà chúng ta đã được trả lời trước đó bởi những người khác. Do đó, phương pháp phỏng vấn từng cá nhân riêng lẻ vẫn còn những hạn chế nhất định.



Phỏng vấn nhóm là một phương pháp tốt có thể giúp giải quyết được những yêu cầu trái ngược nhau. Các đặc điểm của phỏng vấn nhóm bao gồm:

- Nhiều phân tích viên phụ trách nhiều lãnh vực khác nhau
- Nhiều đối tượng phỏng vấn khác nhau mỗi đối tượng phụ trách một lãnh vực, có thể phân cấp từ quản lý đến nhân viên trực tiếp liên quan.
- Tổ chức một buổi phỏng vấn chung gồm các phân tích viên và các đối tượng phỏng vấn.
- Mỗi phân tích viên có thể đặt câu hỏi và các đối tượng đều có thể trả lời. Phân tích viên có thể ghi nhận lại chỉ những ý kiến liên quan đến lãnh vực của mình.

### Lợi điểm:

- Giảm thiểu thời gian phỏng vấn: tất cả các yêu cầu sẽ được thông suốt tại một thời điểm thay vì phải phỏng vấn từng đối tượng một tại những thời điểm khác nhau và thời gian sẽ kéo dài ra

- Cho phép các đối tượng phỏng vấn nghe được ý kiến chủ đạo của lãnh đạo trên những ý kiến bất đồng liên quan đến một vấn đề đặt ra. Đây là cơ hội làm cho các đối tượng thông suốt được ý kiến chủ đạo liên quan đến hệ thống mới.

**Nhược điểm:**

Nhược điểm chính là rất khó để tổ chức một buổi phỏng vấn nhóm vì khó để tìm được một thời gian và vị trí thích hợp cho tất cả mọi người. Ngày nay với công nghệ truyền thông phát triển cho phép tổ chức một buổi họp với các thành viên ở khoảng cách xa nhau (ví dụ: dùng Video conference).

***Quan sát trực tiếp***

Quan sát trực tiếp tại nơi làm việc, hiện trường nhằm thu thập chính xác cách thức và qui trình làm việc thực tế của hệ thống.

Ưu điểm:

- Đảm bảo tính trung thực của thông tin. Bởi vì các phương pháp phỏng vấn bị phụ thuộc vào cách thức mà người dùng trả lời, kiến thức và chủ quan của họ,
- Thu thập tốt về thông tin mô tả tổng quan về hệ thống.

Hạn chế

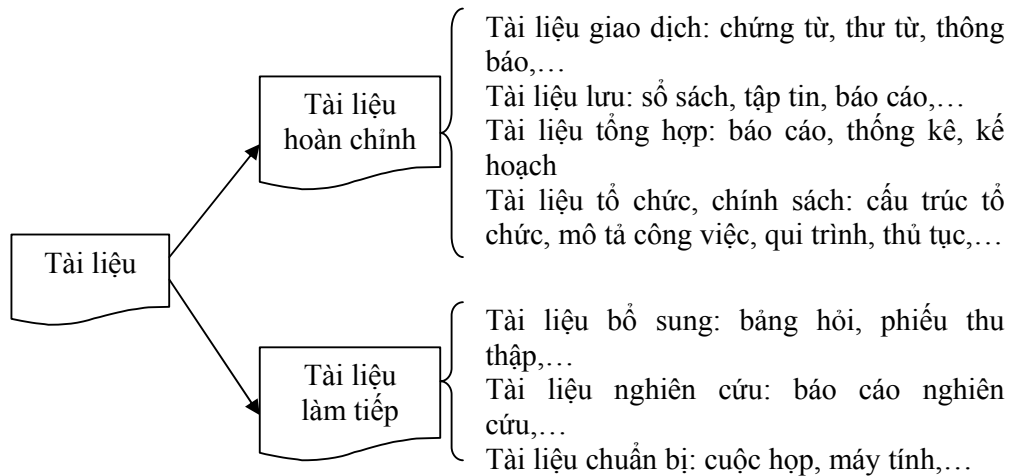
- Thời gian có thể kéo dài.
- Làm cho người dùng khó chịu khi thực hiện công việc, vì có cảm giác như bị theo dõi. Do đó, họ thường thay đổi cách thức làm việc không đúng với hiện trạng.

Thông thường, người ta kết hợp các phương pháp phỏng vấn với phương pháp quan sát để tiến hành khảo sát.

***Phân tích tài liệu và thủ tục***

Phương pháp quan sát hệ thống hoạt động là phương pháp trực tiếp thì phương pháp nghiên cứu tài liệu và thủ tục là phương pháp quan sát gián tiếp, bởi vì nó không nghiên cứu trực tiếp ở hiện trường hệ thống mà thông qua các văn bản, giấy tờ, tài liệu, tập tin máy tính,... mô tả hệ thống. Phương pháp này giúp xác định chi tiết về hệ thống hiện hành.

Có rất nhiều tài liệu liên mô tả hoạt động hệ thống, các yêu cầu của hệ thống trong tương lai: tài liệu mô tả nhiệm vụ, các kế hoạch kinh doanh, cấu trúc tổ chức, các tra cứu về chính sách, bản mô tả công việc, các thư tín bên trong và bên ngoài, các báo cáo nghiên cứu,... Chúng ta có thể thu thập được nhiều loại thông tin từ các hoạt động chung của đơn vị đến các dữ liệu cơ bản, dữ liệu cấu trúc. Thông thường phương pháp này kết hợp với phương pháp phỏng vấn ở mức thấp.



Phân tích tài liệu sẽ mang lại các thông tin sau:

- Các vấn đề tồn tại trong hệ thống (thiếu thông tin, các bước xử lý dư thừa)
- Các cơ hội để hệ thống đáp ứng nhu cầu mới (ví dụ: việc phân tích tài liệu cho thấy từ dữ liệu lưu trữ mà lâu nay không để ý có thể phân tích thông tin từng loại khách hàng, điều này tạo một cơ hội cho bộ phận bán hàng là có thể đánh giá và phân tích hoạt động bán hàng)
- Phương hướng tổ chức có thể tác động đến các yêu cầu của HTTT (ví dụ: một phương hướng mới của đơn vị là liên kết khách hàng và nhà cung cấp gần gũi hơn nữa với đơn vị mà trước đây chưa tính đến hoặc chưa thực hiện. Phương hướng này làm nảy sinh các nhu cầu mới về HTTT cần có để đáp ứng như là: hệ thống mới cần mở ra các kênh liên lạc thông tin cho khách hàng, xử lý các đánh giá dịch vụ khách hàng,...)
- Lý do tồn tại của hệ thống hiện hành, những chi tiết không được quản lý bởi hệ thống hiện hành và bây giờ thì cần thiết và khả thi trong hệ thống mới.
- Tìm ra tên và vị trí của những cá nhân có liên quan đến hệ thống. Giúp cho việc giao tiếp liên lạc đúng mục tiêu hơn.
- Giá trị của đơn vị, cá nhân có thể trợ giúp để xác định các ưu tiên đối với những khả năng khác nhau đến từ nhiều người dùng khác nhau.
- Các trường hợp xử lý thông tin đặc biệt không thường xuyên không thể được xác định bởi những phương pháp khác.
- Dữ liệu cấu trúc, qui tắc xử lý dữ liệu, các nguyên lý hoạt động được thực hiện bởi HTTT.

Một loại tài liệu hữu dụng khác là các thủ tục mô tả công việc của từng cá nhân hoặc nhóm. Các thủ tục này mô tả cách thức một công việc hoạt động, gồm dữ liệu và thông tin được sử dụng và được tạo ra trong quá trình thực hiện công việc.

Tuy nhiên, việc phân tích tài liệu thủ tục cũng có một số nhược điểm sau:

- Các thủ tục cũng là nguồn thông tin không đúng, trùng lặp
- Thiếu tài liệu
- Tài liệu hết hạn: dẫn đến việc phân tích tài liệu cho một kết quả không đúng với kết quả khi phỏng vấn.



## **Các phương pháp mới xác định yêu cầu**

### ***Thiết kế kết hợp người dùng (JAD - Join Application Design)***

Mục tiêu của JAD là một tiến trình xác định yêu cầu trong đó người dùng, nhà quản lý và các nhà phân tích làm việc với nhau trong một vài ngày diễn ra trong các buổi họp tập trung (trong một phòng) để xác định hoặc kiểm tra lại các yêu cầu hệ thống và các thiết kế chi tiết. Do đó, JAD có hình thức như là phương pháp phỏng vấn nhóm. Tuy nhiên, JAD đi theo một cấu trúc vai trò và chương trình đặc biệt hoàn toàn khác với phương pháp phỏng vấn nhóm đó là phân tích viên điều khiển thứ tự câu hỏi được trả lời bởi người dùng.

Mục đích chính của JAD trong giai đoạn phân tích là thu thập yêu cầu hệ thống một cách đồng thời từ nhiều đối tượng khác nhau, kết quả là một tiến trình tập trung, có cấu trúc nhưng có hiệu quả cao. Điểm giống nhau với phỏng vấn nhóm là JAD cũng cho phép các phân tích viên quan sát và xác định được ở đâu đồng ý và ở đâu có bất đồng trong các người dùng. Các cuộc gặp gỡ diễn ra trong vòng nhiều ngày tạo ra cơ hội để giải quyết bất đồng hoặc ít nhất cũng hiểu được tại sao có bất đồng.

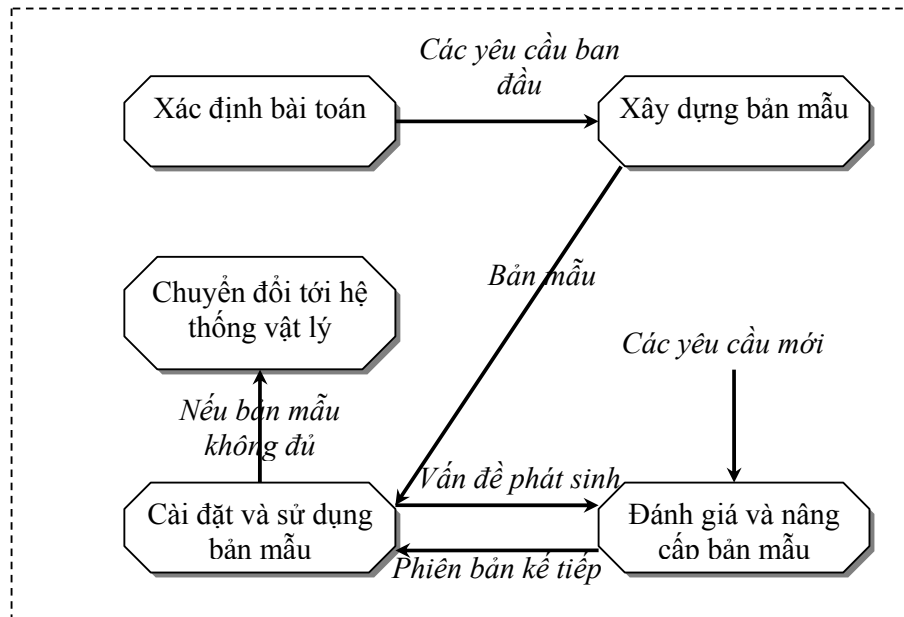
Thành phần của JAD bao gồm:

- Một địa điểm: một địa điểm (phòng họp) có đầy đủ các trang thiết bị hỗ trợ cho cuộc họp, mục đích là làm cho mọi người có tập trung cao trong việc phân tích hệ thống.
- Người tham dự bao gồm:
  - o Người chủ trì: điều hành cuộc họp, thiết lập chương trình, giữ thái độ trung lập, tập trung vào việc hướng cuộc họp vào đúng chương trình, giải quyết bất đồng.
  - o Người dùng: đại diện người sử dụng hệ thống
  - o Phân tích viên hệ thống: các phân tích viên đặt câu hỏi về hệ thống
  - o Người ghi chép: ghi chép tất cả các thông tin quá trình diễn ra JAD
  - o Nhân viên HTTT: ngoài các phân tích viên, bao gồm thêm các lập trình viên, phân tích viên CSDL.
- Chương trình: chương trình thể hiện nội dung của JAD bao gồm các bước và cuộc họp phải diễn ra đúng chương trình này
- Công cụ: các công cụ trợ giúp phân tích thiết kế (thiết kế bản mẫu, vẽ sơ đồ,...), đánh giá và trợ giúp cho các phân tích để nâng cao hiệu quả và giảm thiểu thời gian của JAD.

### ***Sử dụng bản mẫu (prototype) xác định yêu cầu***

Sử dụng bản mẫu như một kỹ thuật xác định yêu cầu, phân tích viên làm việc với người dùng để xác định các yêu cầu cơ bản và ban đầu của hệ thống. Sau đó, phân tích viên dựa trên yêu cầu này để xây dựng một bản mẫu ban đầu. Bản mẫu khi hoàn thành sẽ gửi đến người dùng để người dùng sử dụng thử và kiểm tra. Đặc biệt, việc trực quan hóa các mô tả yêu cầu bằng lời được chuyển đổi thành hệ thống vật lý sẽ nhắc nhở người dùng thay đổi những yêu cầu tồn tại không phù hợp và phát sinh những yêu cầu mới (ví dụ: trong buổi phỏng vấn ban đầu, người dùng muốn xây dựng một form nhập hóa đơn với tất cả thông tin về khách hàng, hoá đơn, dịch vụ, hàng hoá, quá trình thanh toán,... theo cách nghĩ của người dùng là tiện lợi. Tuy nhiên sau khi sử dụng bản mẫu, người dùng sẽ cảm thấy phức tạp, lộn xộn và sẽ thay đổi yêu cầu với nhiều form khác nhau và sự di chuyển hợp lý giữa các form). Kết quả thử nghiệm của người dùng sẽ phản hồi tới phân tích viên và phân tích viên sẽ dùng thông tin phản hồi này để cải tiến bản mẫu rồi tiếp tục gửi đến người dùng và vòng lặp này cứ tiếp tục như vậy cho đến khi bản mẫu thoả mãn người dùng.

Khi sử dụng phương pháp này, phân tích viên cũng phải sử dụng các phương pháp truyền thống để thu thập thông tin ban đầu.



Hình 2. Sơ đồ xác định yêu cầu dùng phương pháp bản mẫu (The New paradigm for Systems Development – J.D. Naumann & A.M. Jenkins)

Phương pháp bản mẫu sẽ rất hữu dụng để xác định yêu cầu trong các trường hợp sau:

- Yêu cầu chưa rõ ràng và thông suốt, thường là các trường hợp về hệ thống mới hoặc là các trường hợp về hệ hỗ trợ ra quyết định.
- Người dùng và các thành viên khác tham gia vào việc phát triển hệ thống.
- Việc thiết kế phức tạp và đòi hỏi phải có một hình thức cụ thể để đánh giá.
- Có những vấn đề giao tiếp đã tồn tại giữa phân tích viên và người dùng và tất cả đều mong muốn làm sáng tỏ.
- Công cụ (đặc biệt là công cụ phát sinh form và report) và dữ liệu sẵn sàng để xây dựng hệ thống.

Phương pháp này cũng có một số hạn chế:

- Tạo ra một xu hướng làm việc không theo chuẩn tài liệu hình thức về yêu cầu hệ thống, và điều này làm khó khăn hơn để phát triển một hệ thống đầy đủ cần phải có một chuẩn mực tuân theo.
- Các bản mẫu có thể trở thành rất đặc thù phong cách của người dùng ban đầu và khó để thích ứng với những người dùng tiềm năng khác.
- Các bản mẫu thường được xây dựng trên các hệ thống đơn. Do đó, nó bỏ qua các phát sinh về tương tác và chia sẻ dữ liệu với những hệ thống khác.

Ví dụ: mô tả khảo sát hoạt động của hệ thống máy ATM ngân hàng ABC

ATM là một loại máy rút tiền tự động, máy được các ngân hàng lắp đặt để hỗ trợ cho khách hàng có thể rút tiền ở các vị trí thuận tiện mà không phải đến ngân hàng. Hoạt động của máy được mô tả như sau:

Hệ thống được thiết kế để điều khiển một máy giao dịch tự động (ATM – Automated Teller

Machine) có một đầu đọc từ để đọc thẻ ATM, một màn hình giao tiếp (hiển thị và bàn phím), một khe nhỏ để chuyển tiền, một khay đựng tiền, một máy in để in hoá đơn và một công tắc cho phép nhân viên vận hành bật và tắt máy. Máy ATM sẽ giao tiếp với hệ thống ngân hàng thông qua một phương thức thích hợp.

Máy ATM sẽ phục vụ cho một khách hàng tại một thời điểm. Khách hàng của ngân hàng sẽ được lưu trữ thông tin về tên, số thẻ và PIN code (gồm 4 ký số) dùng để nhận dạng khách hàng, khách hàng có thể gửi và rút tiền từ tài khoản của mình tại máy ATM.

Một khách hàng phải có một tài khoản tại ngân hàng. Với tài khoản này, khách hàng có thể thực hiện các giao dịch được cung cấp bởi máy ATM của ngân hàng. Khi khách hàng đến máy ATM để sử dụng, khách hàng sẽ được yêu cầu đưa thẻ vào máy, hoặc nhập vào số thẻ và mã PIN kiểm tra. Sau khi kiểm tra thành công, khách hàng có thể thực hiện một số giao dịch trên máy như sau:

- Rút tiền: khách hàng nhập vào số tiền cần rút. Nếu số tiền dư trong tài khoản tiền gửi nhỏ hơn số tiền rút, hệ thống tự động tạo thêm một giao dịch rút tiền từ tài khoản tiết kiệm. Nếu số dư trong tài khoản vẫn không đủ hệ thống sẽ thông báo cho khách hàng và kết thúc giao dịch.
- Gửi tiền: khách hàng có thể thực hiện việc gửi tiền vào tài khoản tiền gửi hoặc tiết kiệm.
- Xem thông tin tài khoản: khách hàng có thể chọn xem thông tin về tài khoản của mình sau khi đăng nhập vào hệ thống.

Khách hàng cũng có thể huỷ bỏ thực hiện một dịch vụ bằng việc chọn “huỷ bỏ” hoặc “đóng” từ giao diện máy.

## **Yêu cầu**

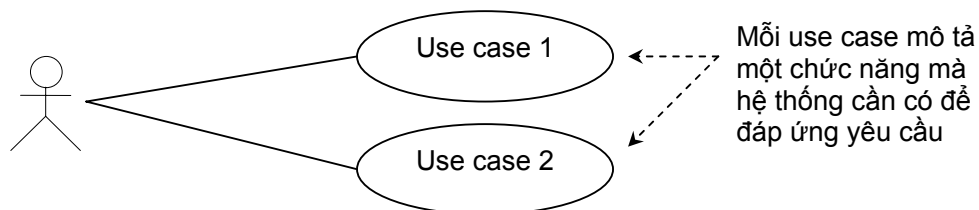
Yêu cầu là: “một điều kiện, hoặc khả năng mà hệ thống phải đáp ứng”. Yêu cầu có thể phân thành hai loại lớn là yêu cầu chức năng (functional requirement) và yêu cầu phi chức năng (nonfunctional requirement):

## **Yêu cầu chức năng**

Khi mô tả về một hệ thống, chúng ta nghĩ ngay đến hệ thống sẽ có những gì để thực hiện trên quan điểm người sử dụng. Những việc thực hiện này được xem như là các hành động mà hệ thống phải thi hành và được mô tả như là chức năng, hành vi, và yêu cầu.

Yêu cầu chức năng được dùng để diễn đạt hành vi của một hệ thống bằng việc xác định tất cả điều kiện đầu vào và đầu ra để đạt được một kết quả mong muốn.

Việc biểu diễn yêu cầu chức năng thường thông qua các sơ đồ. Trong UML chúng ta có thể dùng sơ đồ use case, sơ đồ hoạt động, sơ đồ tương tác. Ví dụ, trong một sơ đồ use case. Mỗi use case dùng để biểu diễn một chức năng của hệ thống cần có để cung cấp tới một đối tượng tác nhân.



## **Yêu cầu phi chức năng**

Là các đặc điểm chất lượng của chức năng mà hệ thống cần đáp ứng nhằm thỏa mãn nhu cầu người sử dụng. Các đặc điểm chất lượng này được gọi là các yêu cầu phi chức năng. Chúng ta phân loại yêu cầu phi chức năng như sau:

- Sự tiện lợi (usability): là các yêu cầu về yếu tố thẩm mỹ con người, tính dễ học, dễ sử dụng và sự nhất quán của giao diện, tài liệu sử dụng và các tài nguyên huấn luyện.
- Sự tin cậy (reliability): là các yêu cầu về tần suất và giới hạn về hỏng hóc, khả năng phục hồi, khả năng dự đoán và độ chính xác.
- Hiệu năng (performance): là các điều kiện áp đặt lên các yêu cầu chức năng. Ví dụ: tỉ lệ giao tác thực hiện, tốc độ thực hiện, tính sẵn sàng, độ chính xác, thời gian đáp ứng, thời gian phục hồi, dung lượng bộ nhớ sử dụng cho một hoạt động thi hành bởi hệ thống.
- Khả năng chịu đựng (supportability): là các yêu cầu về độ bền, khả năng duy trì, và các yêu cầu khác về chất lượng đòi hỏi hệ thống phải được cập nhật sau thời điểm triển khai.

## **Phân loại yêu cầu**

Với cách tiếp cận truyền thống, các yêu cầu được xem như là các đặc tả văn bản tương ứng với một trong hai loại trên, được diễn đạt qua hình thức: "Hệ thống sẽ ...". Tuy nhiên, để quản lý đầy đủ yêu cầu một cách có hiệu quả, các yêu cầu phải được mô tả dựa trên sự hiểu biết của người dùng và các đối tượng có liên quan. Sự hiểu biết này cung cấp cho nhóm phát triển lý do "tại sao?" cũng như "cái gì?" của hệ thống sẽ được phát triển. Vì hệ thống sẽ liên quan đến nhiều loại đối tượng khác nhau do đó, yêu cầu của hệ thống cũng có thể được phân loại theo nhiều cấp khác nhau:

### **Nhu cầu (need):**

Mô tả các yêu cầu ở mức cao thường là các đối tượng có liên quan đến dự án như là: người đầu tư, người hưởng lợi từ dự án, người dùng cuối, cũng như người mua, người thầu, người phát triển, người quản lý,... hoặc những đối tượng khác mà nhu cầu của họ hệ thống phải đáp ứng. Thu thập các nhu cầu này chúng ta phải khảo sát thông qua các phương pháp khảo sát như đã đề cập bên trên. Các nhu cầu được thu thập thông thường mô tả ở mức cao, không rõ ràng, nhọc nhằn và thường bắt đầu như là một "nhu cầu" hoặc "mong muốn".

Ví dụ các nhu cầu có thể là:

"Tôi cần gia tăng khả năng sản xuất",

"Tôi có nhu cầu mở rộng khả năng đáp ứng đơn hàng",

"Tôi có nhu cầu cải tiến hiệu năng hoạt động của hệ thống"

"Tôi muốn mở rộng việc khai thác số liệu của khách hàng"

...

Các nhu cầu này được xem như là một tập hợp rất quan trọng giúp chúng ta hiểu về các mong muốn thực sự của các đối tượng liên quan ở mức cao và nó sẽ cung cấp các đầu vào then chốt tới các yêu cầu chi tiết của hệ thống giúp chúng ta xác định các lý do và nội dung hành vi hệ thống.

### **Đặc điểm hệ thống (feature)**

Trong quá trình khảo sát hệ thống, nhu cầu và yêu cầu thường đi đôi với nhau. Trong khi nhu cầu là những cái mong muốn mang lại từ hệ thống trên quan điểm còn không rõ ràng thì yêu cầu ngược lại được mô tả mang tính giải pháp cho những nhu cầu đó.

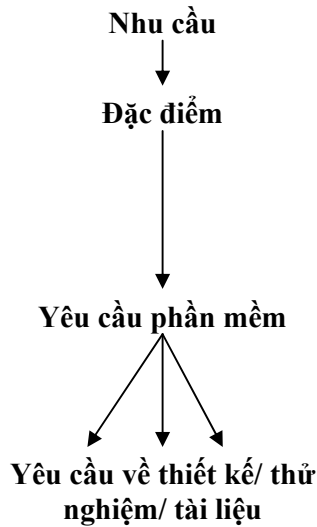
Ví dụ: một nhu cầu “Tôi muốn thông báo đến nhà cung cấp nhanh hơn” thì một yêu cầu là “hệ thống sẽ phát sinh thông báo tự động qua email đến nhà cung cấp”

Các yêu cầu này chính là các biểu thức ở mức cao về hành vi hệ thống – gọi là đặc điểm (feature). Xét về khía cạnh kỹ thuật, đặc điểm được xem như là “một dịch vụ được cung cấp bởi hệ thống để đáp ứng nhu cầu”. Như vậy, đặc điểm của hệ thống chính là sự chuyển đổi quan điểm về *cái gì* (“thông báo nhanh hơn”) thành *như thế nào* (“email tự động”).

Để xác định một đặc điểm chúng ta có thể thêm vào một số thuộc tính khác như là: rủi ro, độ ưu tiên, độ nỗ lực

### **Yêu cầu phần mềm**

Để thích hợp hơn trong quá trình trao đổi với người phát triển về chính xác những gì mà hệ thống sẽ làm. Chúng ta cần một đưa vào thêm một mức đặc tả để chuyển dịch những nhu cầu và đặc điểm thành một đặc tả mà chúng ta có thể thiết kế, cài đặt, thử nghiệm. Các đặc tả này gọi là yêu cầu phần mềm và có thể tiếp cận theo hai loại: yêu cầu chức năng và yêu cầu phi chức năng.



## **Câu hỏi và bài tập**

### **Câu hỏi**

1. Mục đích của khảo sát yêu cầu là gì?
2. Cần tiếp cận những đối tượng nào trong quá trình khảo sát?
3. Cần tu tập các nội dung gì trong quá trình khảo sát?
4. Các phương pháp khảo sát yêu cầu?
5. Ưu và khuyết điểm của phương pháp phỏng vấn?
6. Ưu và khuyết điểm của phương pháp dùng bảng câu hỏi?
7. Yêu cầu là gì? Như thế nào là yêu cầu chức năng, phi chức năng?
8. Yêu cầu được phân thành bao nhiêu loại? Ý nghĩa của từng loại?

## Chương 5 MÔ HÌNH HOÁ USE CASE

### **Mục tiêu**

Nội dung của chương này cung cấp cho sinh viên:

- Hiểu ý nghĩa của việc sử dụng sơ đồ use case trong biểu diễn yêu cầu hệ thống
- Xác định được các tác nhân và mối quan hệ giữa các tác nhân của một hệ thống phần mềm
- Xác định được các use case biểu diễn chức năng phần mềm hệ thống và mối quan hệ giữa tác nhân và use case nhằm xây dựng sơ đồ use case mô tả yêu cầu phần mềm hệ thống
- Tinh chế sơ đồ use case nhằm làm gia tăng tính diễn đạt, tính tái sử dụng qua việc sử dụng các liên kết <<extend>>, <<include>>

### **Giới thiệu**

Trong giai đoạn phân tích, kết quả của quá trình khảo sát yêu cầu phản ánh quá trình làm việc của người phát triển với người sử dụng. Các kết quả này phải nhắm đến yếu tố của người dùng. Có nghĩa là người phát triển trước tiên phải diễn đạt bức tranh của hệ thống tương lai theo cách nhìn của người sử dụng. Điều này sẽ giúp cho người dùng có thể thấy được hệ thống sẽ làm thỏa mãn các yêu cầu như thế nào và đó chính là chìa khoá đầu vào cho việc phát triển hệ thống trong các giai đoạn về sau. Một công cụ giúp diễn đạt điều này chính là mô hình use case.

Jacobson và cộng sự của ông (1992) là những người tiên phong trong việc sử dụng mô hình use case để phân tích yêu cầu hệ thống. Bởi vì mô hình use case đặt trọng tâm để biểu diễn hệ thống hiện tại làm gì, hệ thống mới sẽ làm gì và môi trường của nó. Nó giúp cho người phát triển có thể hiểu rõ về yêu cầu chức năng hệ thống mà không quan tâm đến chức năng này được cài đặt như thế nào.

Để hiểu yêu cầu của hệ thống, chúng ta phải tìm ra người dùng sẽ sử dụng hệ thống như thế nào. Do đó, từ quan điểm một người dùng, chúng ta phát hiện các tình huống sử dụng khác nhau của người dùng, các tình huống này được thiết lập bởi các use case. Tổng hợp các use case và tác nhân cùng với quan hệ giữa chúng sẽ cho ta một mô hình use case mô tả yêu cầu của hệ thống.

Trong chương 6, quá trình mô hình hoá nghiệp vụ được áp dụng đối với các hệ thống nghiệp vụ và kết quả của nó sẽ cung cấp sơ đồ use case từ việc thống nhất các yêu cầu hệ thống phần mềm để tự động hoá hoạt động của hệ thống nghiệp vụ đó. Tuy nhiên, trong những hệ thống mà không có hoạt động nghiệp vụ (ví dụ: hệ thống nhúng), hoặc các nghiệp vụ của hệ thống không quá phức tạp hoặc không quan tâm để mô hình hoá nghiệp vụ thì việc xây dựng mô hình use case phần mềm sẽ là bước tiếp cận mô hình hoá đầu tiên về hệ thống. Một tiến trình xây dựng sơ đồ use case bao gồm các bước sau:

- Xác định tác nhân hệ thống
  - o Ai đang sử dụng hệ thống?
  - o Hoặc trong trường hợp phát triển mới thì ai sẽ sử dụng hệ thống?
- Phát triển use case
  - o Người dùng (tác nhân) đang làm gì với hệ thống?
  - o Hoặc trong trường hợp hệ thống mới thì người dùng sẽ làm gì với hệ thống?
- Xây dựng sơ đồ use case

- Xác định mối quan hệ giữa tác nhân – use case
- Xác định mối quan hệ giữa các use case
- Phân chia sơ đồ use case thành các gói (package)

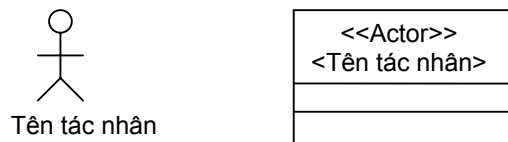
## Xác định tác nhân

### Tác nhân (actor)

Ý nghĩa: một tác nhân là một đối tượng bên ngoài hệ thống giao tiếp với hệ thống theo một trong những hình thức sau:

- Tương tác, trao đổi thông tin với hệ thống hoặc sử dụng chức năng hệ thống
- Cung cấp đầu vào hoặc nhận các đầu ra từ hệ thống
- Không điều khiển hoạt động của hệ thống

Ký hiệu

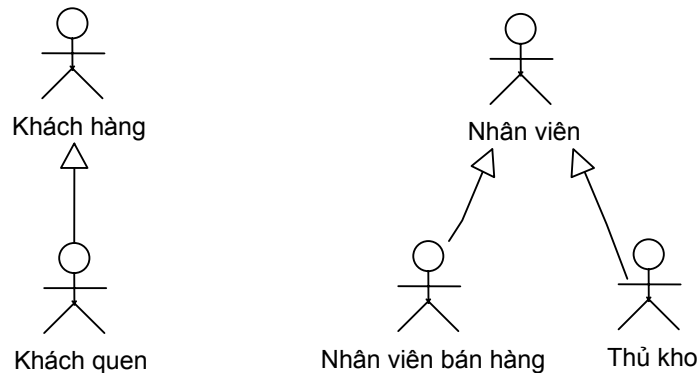


Tên tác nhân: tên tác nhân là một **danh từ**

### Quan hệ giữa các tác nhân:

Là quan hệ tổng quát hóa và chuyên biệt hoá

Ví dụ:

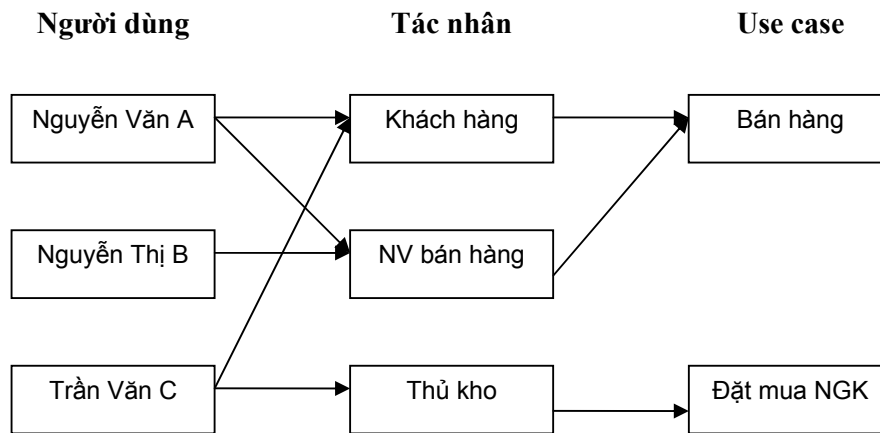


### Xác định tác nhân

Xác định tác nhân cũng được xem có tầm quan trọng như xác định class, use case, liên kết,....

Khi xác định người sử dụng phần mềm hệ thống, chúng ta đừng quan trọng vấn đề quan sát người nào đang sử dụng hệ thống mà chúng ta nên xác định xem vai trò chịu trách nhiệm trong việc sử dụng hệ thống. Nghĩa là tác động lên hệ thống theo nghĩa cung cấp thông tin cho hệ thống hoặc nhận kết quả xử lý từ hệ thống.

Tác nhân được hiểu là một vai trò tham gia vào hệ thống không giống như một con người cụ thể hoặc một công việc. Một đối tượng có thể tham gia vào một hoặc nhiều vai trò



Qua quá trình khảo sát và phân tích tài liệu hệ thống, chúng ta có thể nhận ra các tác nhân thông qua các câu hỏi sau:

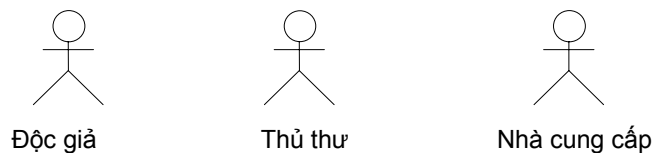
- Ai đang sử dụng hệ thống? Hoặc ai được tác động bởi hệ thống? Hoặc nhóm đối tượng nào cần hệ thống trợ giúp để làm công việc? (tác nhân chính)
- Ai tác động tới hệ thống? Những nhóm đối tượng nào hệ thống cần để thực hiện hoạt động của nó (hoạt động gồm chức năng chính và chức năng phụ, như là chức năng quản trị)?
- Những phần cứng hoặc hệ thống bên ngoài nào sử dụng hệ thống?

Ví dụ: trong hoạt động của máy ATM của một ngân hàng, các tác nhân được xác định là:



Trong đó, các tác nhân Khách hàng, Nhân viên ngân hàng là các tác nhân chính (primary actor) của hệ thống ATM. Bởi vì khách hàng là mục tiêu mà hệ thống tương tác; Nhân viên ngân hàng sử dụng hệ thống để trợ giúp công việc. Trong khi đó, Nhân viên vận hành là tác nhân phụ (secondary actor) bởi vì tác nhân này đảm nhận những chức năng phụ mà hệ thống cần có để thực hiện hoạt động của nó.

Hoặc trong một thư viện của trường đại học, các tác nhân của hệ thống phần mềm quản lý thư viện gồm:



## Xác định use case

### Use case

Một Use case được xem như một chức năng hệ thống từ quan điểm người dùng, như vậy tập hợp tất cả use case biểu diễn bộ mặt của hệ thống bao gồm các chức năng cần có để cung cấp cho các đối tượng tương tác làm việc với hệ thống. Như vậy, use case dùng để mô tả yêu cầu



của hệ thống mới về mặt chức năng, mỗi chức năng sẽ được biểu diễn như một hoặc nhiều use case.

Ví dụ: hệ thống cửa hàng NGK ta có một vài use case

Bán hàng, quản trị tồn kho,...

### Ký hiệu



### **Xác định use case**

Chúng ta bắt đầu từ tập các tác nhân đã xác định trong bước đầu tiên. Ứng với mỗi 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 sẽ đại diện một dòng sự kiện dẫn tới một mục tiêu rõ ràng (hoặc trong một vài trường hợp, dẫn tới một vài mục tiêu riêng biệt có thể là các phương án thay thế cho tác nhân hoặc cho hệ thống so với dòng sự kiện chính)
- Đặt tên cho use case: tên use case nên đặt nhằm phản ánh một mô tả tổng quan về chức năng của use case. Tên nên đến đạt những gì xảy ra khi một thể hiện của use case được thi hành. Một hình thức đặt tên use case phổ biến là : động từ (do) + danh từ (what).
- Mô tả use case một cách ngắn gọn bằng việc áp dụng các thuật ngữ gần gũi với người sử dụng. Điều này sẽ làm cho mô tả use case ít mơ hồ.

Ví dụ: trong hệ thống ATM

Tác nhân Khách hàng sẽ sử dụng hệ thống qua các chức năng:

- Gửi tiền
- Rút tiền
- Truy vấn thông tin về tài khoản

Tác nhân Nhân viên vận hành sẽ sử dụng các chức năng

- Khởi động hệ thống
- Đóng hệ thống



*Gửi tiền:* khách hàng đăng nhập vào hệ thống và yêu cầu gửi tiền vào tài khoản. Khách hàng sẽ xác định tài khoản và số tiền gửi, hệ thống sẽ tạo một giao tác gửi tiền và lưu vào hệ thống. Các bước như sau:

- Yêu cầu xác định tài khoản
- Hệ thống hỏi số tiền gửi

- Nhập vào số tiền gửi
- Khách hàng đưa tiền vào máy ATM

*Rút tiền:* khách hàng đăng nhập hệ thống và yêu cầu rút tiền từ tài khoản. Khách hàng xác định tài khoản và lượng tiền rút. Sau khi kiểm tra số dư tài khoản còn đủ, hệ thống sẽ tạo một giao tác rút tiền và lưu vào hệ thống. Các bước như sau:

- Yêu cầu xác định tài khoản
- Yêu cầu xác định số tiền cần rút
- Nhập số tiền rút
- Kiểm tra số dư tài khoản và số tiền hiện có ở máy có đủ không ?
- Chuyển tiền ra ngoài

*Truy vấn thông tin tài khoản:* khách hàng đăng nhập vào hệ thống và yêu cầu xem thông tin về các giao dịch của tài khoản. Hệ thống hiển thị các thông tin về các giao tác đã tạo lên màn hình cho khách hàng.

*Khởi động hệ thống:* hệ thống được khởi động khi nhân viên vận hành bật công tắc của máy. Nhân viên vận hành sẽ được yêu cầu nhập vào số tiền hiện hành của máy nằm trong két đựng tiền. Sau đó, hệ thống sẽ thiết lập một kết nối tới ngân hàng và các dịch vụ của máy ATM bắt đầu vận hành.

*Đóng hệ thống:* hệ thống được đóng lại khi nhân viên vận hành đảm bảo rằng không có khách hàng nào đang sử dụng máy. Khi đó, nhân viên vận hành sẽ lấy các bao tiền gửi ra, bổ sung lượng tiền, giấy,...

Trong hệ thống quản lý thư viện, các use case được xác định như sau:



## ***Xác định mối quan hệ***

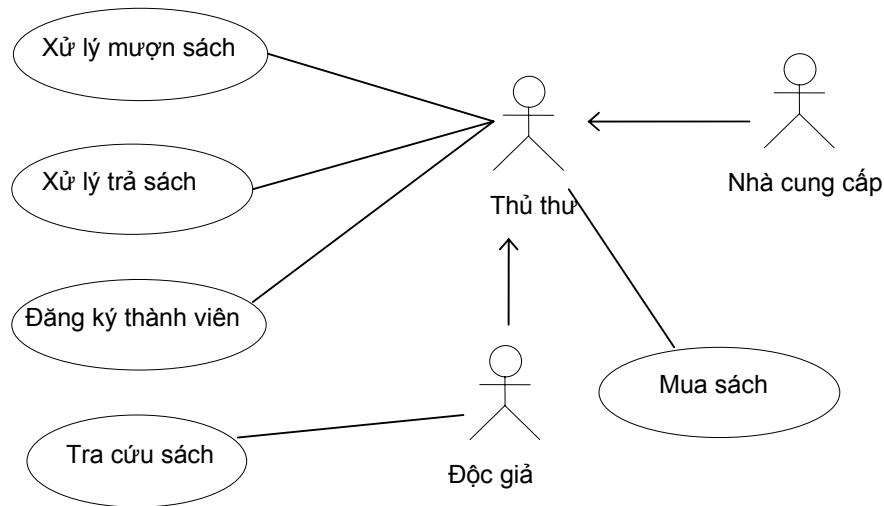
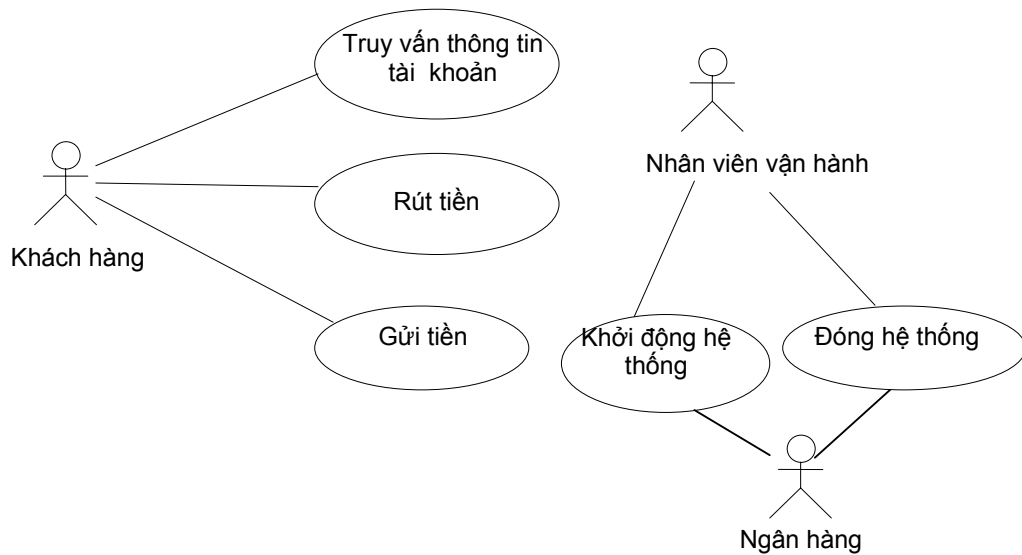
### **Quan hệ tác nhân – use case**

Quan hệ này cho biết tác nhân sẽ tương tác với use case. Một use case luôn luôn được khởi tạo bởi một tác nhân và có thể tương tác với nhiều tác nhân.

Ký hiệu



Ví dụ:



### Mối quan hệ giữa các use case

Việc mô tả use case có thể sẽ khó hiểu nếu use case này chứa đựng nhiều dòng phụ hoặc dòng ngoại lệ chỉ xử lý cho những sự kiện trong những điều kiện đặc biệt. Để làm đơn giản mô tả này chúng ta sử dụng thêm các mối kết hợp <<extend>> và <<include>>.

Liên kết mở rộng (<<extend>>): được dùng khi chúng ta có một use case tương tự như use case khác nhưng có nhiều hơn một vài xử lý đặc biệt. Giống như liên kết tổng quát - chuyên biệt, trong đó, use case chuyên biệt là một mở rộng của use case tổng quát bằng việc đưa thêm vào các hoạt động hoặc ngữ nghĩa mới vào use case tổng quát, hoặc bỏ qua hoạt động của use case tổng quát.

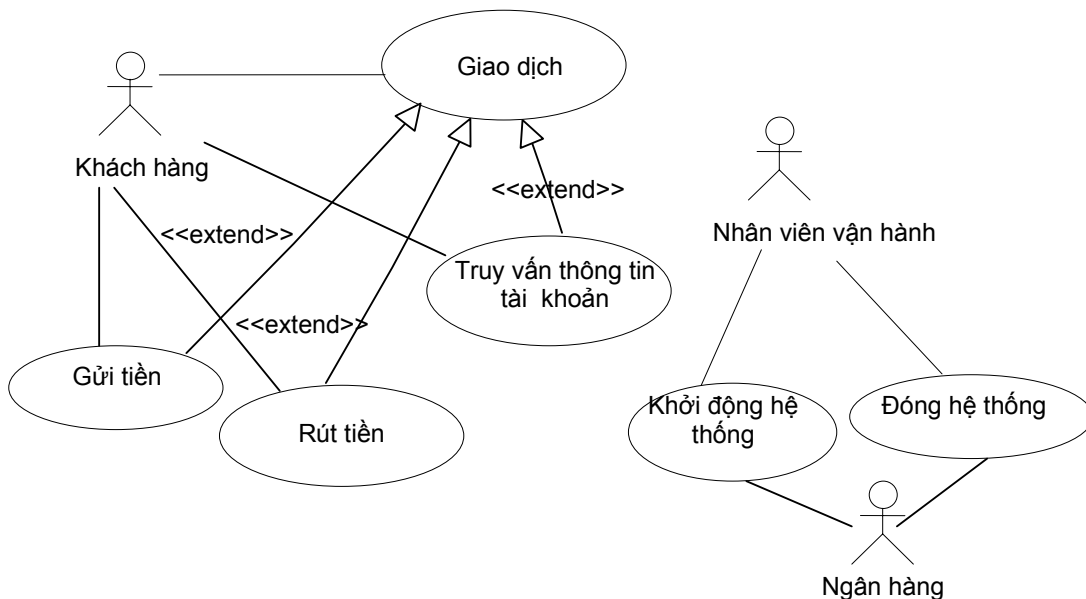
Ví dụ: giả sử Đăng nhập là một use case cơ bản. Use case này sẽ đại diện cho tất cả những gì được xem là thực hiện đăng nhập một cách xuyên suốt. Tuy nhiên, nhiều vấn đề có thể tác động đến dòng sự kiện chính. Ví dụ, mã số PIN không hợp lệ, hoặc thẻ không đọc được do bị hư,... Do đó, chúng ta không phải luôn luôn thi hành các hoạt động thường xuyên của một use case được cho và như vậy, cần thiết tạo ra các use case mới để giải quyết những tình huống mới. Tất nhiên, chúng ta có thể đưa vào use case cơ bản các nội dung xử lý đặc biệt

đó. Tuy nhiên, điều này có thể dẫn đến sự phức tạp với nhiều luận lý riêng biệt và sẽ làm giảm vai trò của dòng chính.

Để giải quyết vấn đề này chúng ta có thể sử dụng quan hệ <<extend>>. Ở đây chúng ta gom các xử lý cơ bản hoặc bình thường vào trong một use case (cơ bản). Các xử lý đặc biệt vào những use case (chuyên biệt) khác. Rồi tạo một liên kết <<extend>> giữa use case cơ bản tới các use case chuyên biệt để khai báo rằng: ngoài xử lý dòng chính (cơ bản), use case cơ bản có mở rộng đến các tình huống xử lý đặc biệt được giải quyết trong các use case chuyên biệt.



Tạo một use case tổng quát có tên là Giao dịch của các use case Rút tiền, Gửi tiền và Truy vấn thông tin tài khoản. Tạo các liên kết <<extend>> từ use case Giao dịch đến các use case này. Như vậy, một rút tiền, hoặc gửi tiền, hoặc truy vấn thông tin tài khoản là một loại giao dịch mà khách hàng có thể sử dụng trên máy ATM. Có nghĩa rằng, các xử lý trong use case Giao dịch sẽ cung cấp một dòng chung và khi khách hàng chọn một loại giao dịch đặc biệt nào đó thì use case này sẽ mở rộng việc giải quyết thông qua các use case chuyên biệt.

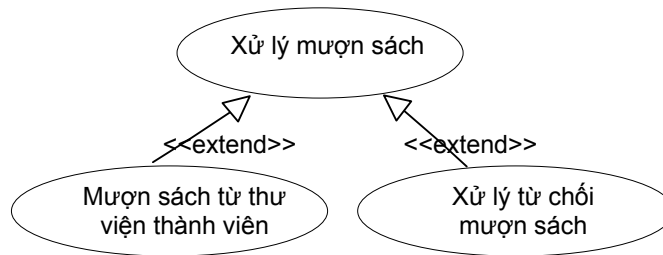


*Giao dịch*: khách hàng tương tác với hệ thống bắt đầu bằng việc đăng nhập hệ thống. Sau khi đăng nhập, khách hàng có thể thực hiện các giao dịch. Sau đây là các bước:

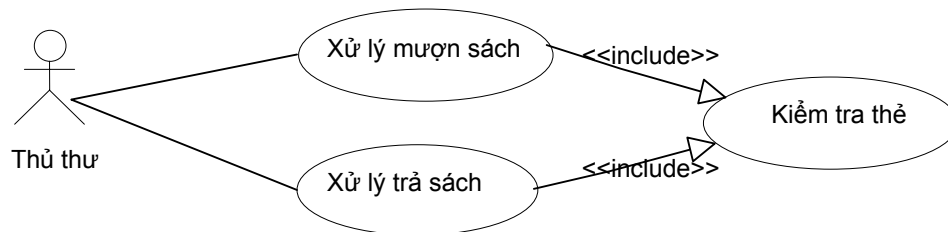
- Đưa thẻ vào máy
- Thực hiện đăng nhập
- Yêu cầu loại giao dịch
- Nhập loại giao dịch
- Thực hiện giao dịch
- Đẩy thẻ ra

- Yêu cầu lấy thẻ
- Lấy thẻ

Trong hệ thống quản lý thư viện, use case *Mượn sách ngoài dòng hoạt động chính* còn có các dòng phụ. Dòng phụ này sẽ được kích hoạt để giải quyết vấn đề khi một độc giả đến mượn tài liệu nhưng không có trong thư viện và thư viện sẽ mượn tài liệu đó từ những thư viện khác có liên kết. Hoặc do độc giả không thỏa các điều kiện để được mượn (mượn sách quá hạn chưa trả của lần mượn trước). Do đó, chúng ta tách dòng phụ này và use case “Mượn sách từ thư viện thành viên” và “” và tạo một liên kết <<extend>> từ use case này đến use case *Xử lý mượn sách*.

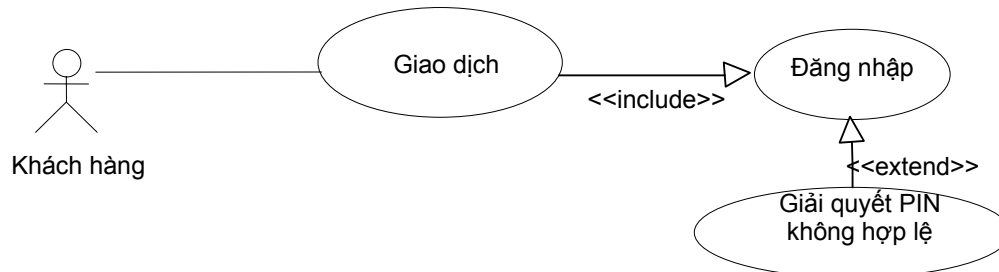


Liên kết sử dụng (<<include>>): được thành lập khi chúng ta có các use case mà tìm thấy một vài use case có những dòng hoạt động chung, và để tránh mô tả dòng hoạt động chung đó lặp lại trên những use case này, chúng ta có thể tách những dòng hoạt động chung đó ra thành một use case. Use case mới này có thể sử dụng bởi những use case khác. Quan hệ giữa những use case với use case được trích ra này gọi là quan hệ <<include>>. Quan hệ sử dụng giúp chúng ta tránh sự trùng lặp bằng cách cho phép một use case có thể được chia sẻ.



Trong ví dụ trên, use case *mượn sách* và *trả sách* đều phải thực hiện công việc kiểm tra thẻ thư viện của độc giả, do đó chúng ta phát sinh một use case mới là *kiểm tra thẻ* bằng cách trích ra hoạt động kiểm tra thẻ thư viện từ hai use case trên và tạo một liên kết <<include>> tới use case từ hai use case đó tới use case mới. Các use case *Xử lý mượn sách* và *Xử lý trả sách* đều thừa hưởng tất cả hoạt động của use case của use case *kiểm tra thẻ*.

Trong hệ thống ATM, use case *Giao dịch* sẽ có mối liên kết <<include>> với use case *Đăng nhập*.



*Đăng nhập*: khách hàng nhập vào mã số PIN gồm bốn ký số. Nếu mã số PIN hợp lệ, tài khoản của khách hàng sẽ sẵn sàng cho các giao dịch. Các bước như sau:

- Yêu cầu mật khẩu (mã số PIN)
- Nhập mật khẩu
- Kiểm tra mật khẩu

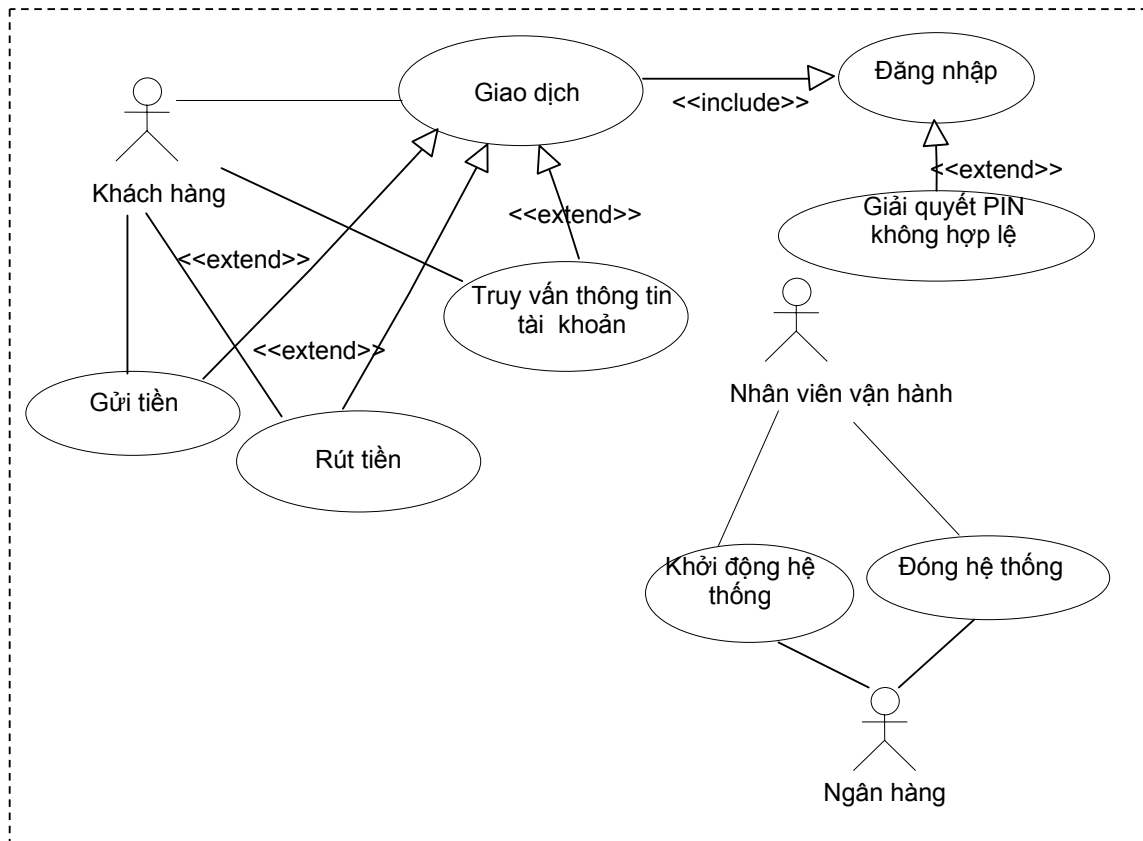
*Giải quyết PIN không hợp lệ:* nếu mã số PIN không hợp lệ, hệ thống sẽ hiển thị một thông báo tới khách hàng.

Sự giống nhau giữa liên kết <<extend>> và liên kết <<include>> là tất cả đều được xem như là một loại kế thừa. Khi chúng ta muốn chia sẻ một số hoạt động chung trong nhiều use case, dùng liên kết <<include>> bằng cách trích các hoạt động chia sẻ đó thành một use case mới. Khi chúng ta muốn thêm vào một ít khác biệt cho một use case để mô tả một tình huống đặc biệt trong một tình huống chung, chúng ta sẽ tạo một use case mới có liên kết <<extend>> với use case chung đó.

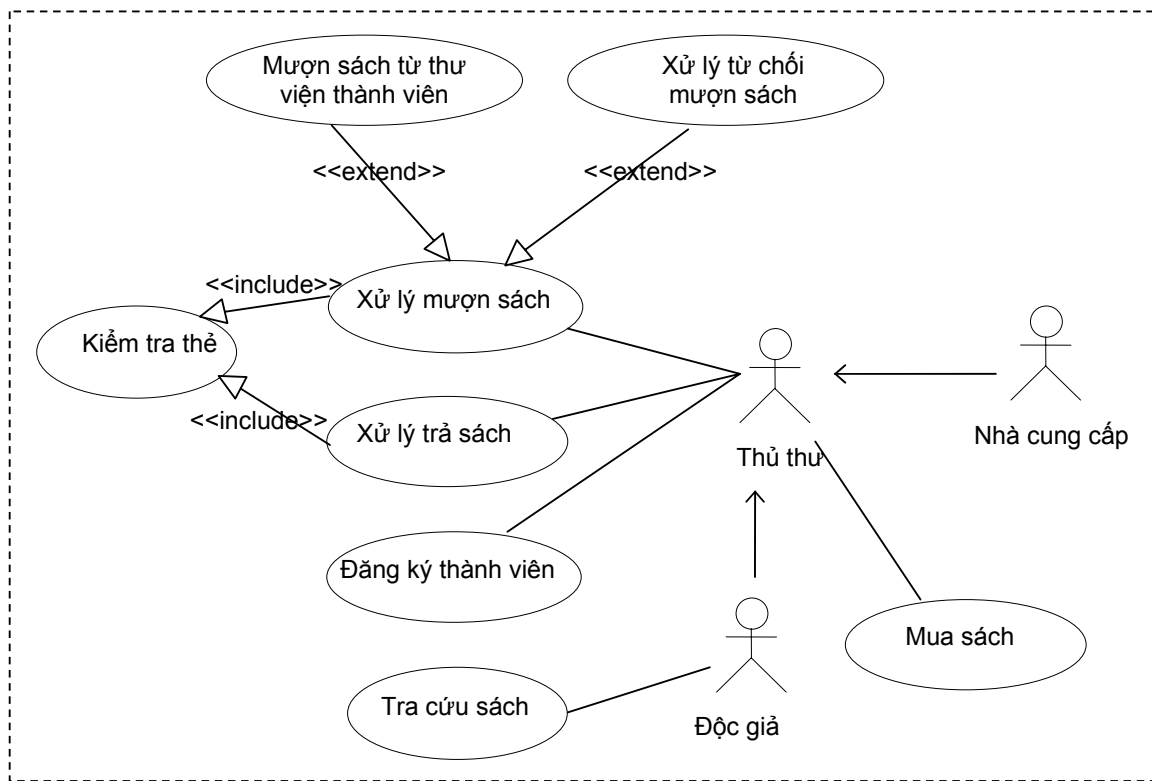
Dựa vào các liên kết được thiết lập cho các use case chúng ta phân use case thành hai loại:

*Use case trừu tượng:* là use case chưa hoàn hảo nghĩa là không tương tác với bất kỳ một tác nhân nào mà được sử dụng bởi một use case khác. Use case trừu tượng cũng có thể có liên kết <<extend>> hoặc liên kết <<include>> trong những mức độ khác. Ví dụ: các use case *Kiểm tra thẻ, Xử lý từ chối mượn sách, ...* là các use case trừu tượng.

*Use case cụ thể:* là use case có tương tác trực tiếp với một tác nhân. Ví dụ: các use case *Xử lý mượn sách, Xử lý trả sách, hoặc Khởi động máy,Đóng máy,....*



Mô hình use case của hệ thống máy ATM

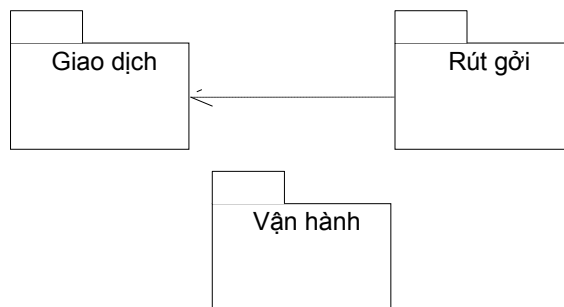


Mô hình use case hệ thống phần mềm quản lý thư viện

### Phân chia các use case thành các gói (package)

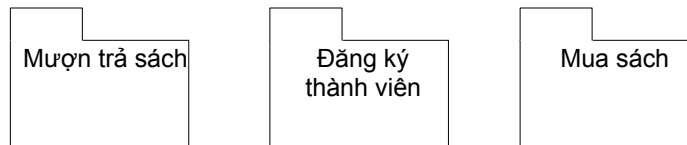
Mỗi use case minh họa một kịch bản trong hệ thống. Khi gặp những hệ thống tương đối phức tạp thì chúng ta nên thu hẹp tiêu điểm của các kịch bản trong hệ thống bằng cách phân chia thành các gói. Mỗi gói phản ánh một phạm vi của hệ thống mà chúng ta chỉ muốn quản lý nó khi chúng ta truy cập gói đó.

Ví dụ, có thể chia các se case của hệ thống máy ATM thành ba gói: Giao dịch, Rút gửi và Vận hành



Trong đó, gói Giao dịch gồm các use case: Giao dịch, Đăng nhập, Giải quyết PIN không hợp lệ; gói Rút gửi gồm các use case: Gởi tiền, Rút tiền, Truy vấn thông tin tài khoản; và gói Vận hành gồm các use case: Khởi động hệ thống, Đóng hệ thống

Hệ thống quản lý thư viện được chia thành ba gói như sau: Mượn trả sách, đăng ký thành viên, và Mua sách.



Trong đó, gói Mượn trả sách gồm các use case: Xử lý mượn sách, Xử lý trả sách, Kiểm tra thẻ, Mượn sách từ thư viện thành viên, Xử lý từ chối mượn sách; gói Đăng ký thành viên gồm use case: Đăng ký thành viên; gói mua sách gồm use case: Mua sách.



## **Câu hỏi và bài tập**

### **Câu hỏi**

9. Mô hình use case là gì?
10. Tại sao mô hình hoá use case là hữu dụng trong phân tích hệ thống?
11. Ai có thể là tác nhân?
12. Khi nào chúng ta sử dụng liên kết <<extend>> và <<include>> trong mô hình hoá use case?
13. Xác định tác nhân bằng cách nào?
14. Phân biệt sự khác nhau giữa người dùng và tác nhân?
15. Tại sao nên phân chia mô hình use case thành các gói?

### **Bài tập**

1. Hãy xây dựng sơ đồ use case mô tả một hệ thống diễn đàn trao đổi học tập của khoa Công Nghệ Thông Tin. Hoạt động của diễn đàn được mô tả như sau:
  - Tất cả mọi người khi truy cập vào địa chỉ của diễn đàn đều có thể xem được thông tin nội dung trao đổi của diễn đàn, tin tức trong diễn đàn.
  - Khi một thành viên muốn gửi thông tin lên diễn đàn. Thông tin đó có thể là một chủ đề cần trao đổi, hoặc một ý kiến liên quan đến một chủ đề đã được đưa ra, hoặc là một tin tức. Trước tiên, thành viên phải đăng nhập vào hệ thống với một tên và mã đăng nhập. Sau khi đăng nhập thành công, thành viên có thể sử dụng chức năng soạn thảo của hệ thống để soạn thảo và gửi thông tin lên diễn đàn. Thành viên có thể xem được các thông báo kết quả duyệt tin từ quản trị về những lần gửi trước.
  - Một người dùng khi truy cập thì có thể đăng ký thành viên của diễn đàn. Khi đăng ký, người dùng sẽ phải nhập các thông tin liên quan như: họ tên, tên đăng nhập, mã sinh viên, mã đăng nhập. Việc đăng ký có hiệu lực sau khi quản trị duyệt và chấp nhận.
  - Khi một thành viên đăng nhập vào quyền quản trị (quản trị hệ thống). Hệ thống sẽ hiển thị các thông tin được gửi bởi các thành viên. Ứng với mỗi thông tin, quản trị có thể chấp nhận hoặc từ chối đăng thông tin lên diễn đàn:
    - o Khi thông tin được đánh dấu là chấp nhận, hệ thống sẽ cập nhật lại trạng thái của thông tin và sẽ hiển thị thông tin này tới diễn đàn.
    - o Khi thông tin được đánh dấu là từ chối, hệ thống cũng sẽ cập nhật lại trạng thái thông tin và ghi chú về lý do từ chối thông tin.
  - Ngoài ra, người quản trị cũng có thể duyệt và chấp nhận hoặc từ chối một thành viên mới đăng ký. Huỷ một thành viên. Cập nhật thông tin của một thành viên.
2. Phân chia sơ đồ use case của hệ thống trên thành các gói. Mô tả sự liên kết giữa các gói (nếu có) và sơ đồ use case cho từng gói.

## Chương 6

# MÔ HÌNH HOÁ NGHIỆP VỤ (BUSINESS MODELING)

### Mục tiêu

Cung cấp cho người học các kiến thức về:

- Như thế nào là mô hình hoá nghiệp vụ, mục tiêu và quy trình của mô hình hoá nghiệp vụ
- Các hoạt động trong phân tích, thiết kế qui trình nghiệp vụ
- Áp dụng UML vào mô hình hoá nghiệp vụ. Đặc biệt, sử dụng sơ đồ use case biểu diễn nội dung của hệ thống nghiệp vụ trong giai đoạn phân tích. Sử dụng sơ đồ đối tượng trong việc thiết kế nghiệp vụ.
- Xác định các yêu cầu tự động hoá từ hệ việc phân tích và thiết kế thống nghiệp vụ.

### Giới thiệu

Mô hình hóa nghiệp vụ là một kỹ thuật để tìm hiểu quy trình nghiệp vụ của một tổ chức. Mô hình nghiệp vụ xác định các quy trình nghiệp vụ nào được hỗ trợ bởi hệ thống. Tóm lại, song song với quá trình khảo sát tìm hiểu về vấn đề hệ thống thì cách tiếp cận nghiệp vụ là phương pháp có hệ thống nhất để nắm bắt các yêu cầu của các ứng dụng nghiệp vụ.

Khi những hệ thống ngày càng phức tạp, việc mô hình hóa trực quan và cách vận dụng các kỹ thuật mô hình hóa ngày càng trở nên quan trọng hơn. Có nhiều nhân tố bổ sung cho sự thành công của một dự án, nhưng việc có một tiêu chuẩn ngôn ngữ mô hình hóa chặt chẽ là nhân tố quan trọng nhất. Một trong những mục đích đầu tiên của mô hình hoá nghiệp vụ là tạo ra các “đối tượng” (mô hình) nhằm để dễ hiểu hơn và để có thể thiết kế những chương trình máy tính bằng cách thông qua hiện tượng thể giới thực như: người, nguyên liệu làm việc và cách thức chúng thực hiện những nhiệm vụ của họ. Như vậy, việc mô hình hóa nghiệp vụ là lập mô hình những tổ chức thể giới thực.

Phạm vi ảnh hưởng của việc mô hình hóa nghiệp vụ có thể biến đổi tùy theo nhu cầu và hệ thống nghiệp vụ cụ thể. Có thể đơn giản chúng ta chỉ nhằm vào việc tăng năng suất bằng cách cải tiến những quy trình đã tồn tại, hoặc là đang tạo ra những sự cải tiến có ảnh hưởng lớn bằng cách thay đổi đáng kể những qui trình nghiệp vụ dựa trên sự phân tích kỹ lưỡng các mục tiêu và các khách hàng của tổ chức. Cho dù là bất kỳ trường hợp nào, những hệ thống thông tin hỗ trợ cho hệ thống nghiệp vụ đều bị ảnh hưởng bởi sự cải tiến của hoạt động nghiệp vụ.

### Tại sao phải mô hình hoá nghiệp vụ?

Trong quá trình phát triển hệ thống phần mềm, đặc biệt là trong các hệ thống phức tạp, một vấn đề tồn tại rất lớn cho thấy đội ngũ phát triển hệ thống thường hiếm khi có một kiến thức hiểu biết đầy đủ về nghiệp vụ của tổ chức mà chính họ là người xây dựng hệ thống phần mềm thực hiện tự động hoá xử lý thông tin trong môi trường nghiệp vụ đó. Trong khi đó, người sử dụng phần mềm chính là các đối tượng xử lý nghiệp vụ thường hiếm khi am hiểu tường tận về các công nghệ và các kỹ thuật của phần mềm nhằm chọn lựa và áp dụng nó một cách phù hợp và hiệu quả với nhu cầu của mình. Điều này luôn tạo ra một khoảng cách giữa người xây dựng và người sử dụng hệ thống. Khoảng cách này là một trở ngại dẫn đến nhiều sự thất bại hoặc không hiệu quả của nhiều dự án tin học hoá hệ thống. Do đó, làm thế nào để các đối tượng này có thể nắm bắt và thống nhất được tốt nhất về cách giải quyết hệ thống trong quá trình tin học hoá. Mô hình hoá nghiệp vụ là một nỗ lực nhằm đưa ra các cách thức diễn tả những qui trình nghiệp vụ dưới dạng những đối tượng và hành động tương tác giữa chúng.

Nếu không mô hình hóa nghiệp vụ thì ta có thể gặp nhiều rủi ro do những người phát triển không có thông tin đầy đủ về cách thức mà nghiệp vụ được thực hiện. Họ chỉ làm những gì mà họ hiểu rõ, như là thiết kế và tạo ra phần mềm, mà không quan tâm đến những gì nghiệp vụ thực thi. Điều này gây ra một sự lãng phí do trước đó đã xây dựng các quy trình nghiệp vụ tốn kém. Rủi ro do những hệ thống được xây dựng không hỗ trợ các nhu cầu thực sự của tổ chức cũng có thể xảy ra rất cao.

Việc hiểu rõ những quy trình nghiệp vụ là quan trọng để có thể xây dựng những hệ thống đúng. Việc mô hình hóa nghiệp vụ có mục tiêu chính là sự phát triển hệ thống, trong đó công việc thực sự là xác định đúng các yêu cầu hệ thống.

Cơ sở để xây dựng hệ thống là sử dụng những vai trò và trách nhiệm của con người cũng như định nghĩa các công việc được xử lý bởi nghiệp vụ. Điều này được thể hiện trong một mô hình đối tượng nghiệp vụ, mà qua đó có thể thấy các vai trò đối tượng sẽ được làm rõ.

Một khi xác định được các mô hình nghiệp vụ, chúng ta cần phải thiết lập những mối quan hệ giữa các use case hệ thống và những mô hình nghiệp vụ. Điều này sẽ cho phép các nhà phân tích được thông báo khi có những thay đổi ở trong hệ thống.

Tóm lại, mục đích của mô hình hóa nghiệp vụ là:

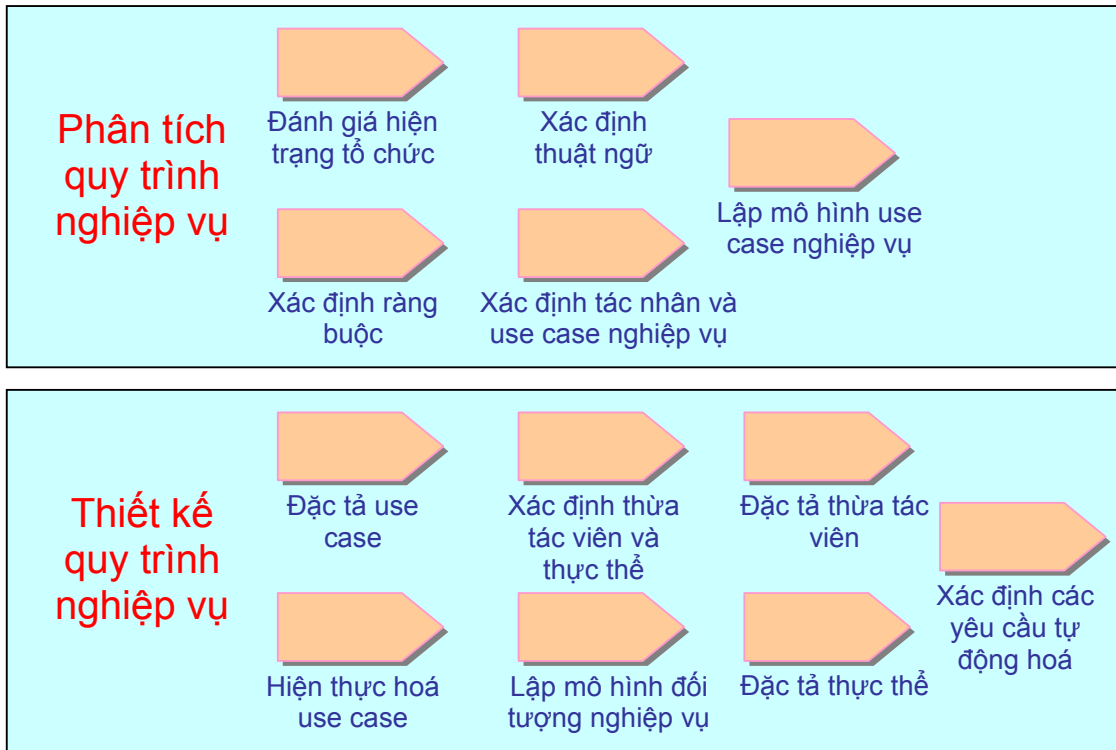
- Hiểu được cấu trúc và các hoạt động của tổ chức đang được hệ thống triển khai.
- Hiểu được các vấn đề hiện tại trong tổ chức và xác định các vấn đề cần cải tiến.
- Bảo đảm rằng các khách hàng, người dùng cuối, và các nhà phát triển có sự hiểu biết chung về tổ chức.
- Thiết lập các yêu cầu tự động hoá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

Để đạt được những mục đích trên, luồng công việc mô hình hóa nghiệp vụ mô tả một bức tranh tổng quát về tổ chức. Từ đó xác định các quy trình (process), các vai trò (role), và các trách nhiệm của tổ chức này trong mô hình use-case nghiệp vụ (business use-case model) và mô hình đối tượng nghiệp vụ (business object model).

## **Luồng công việc trong mô hình hoá nghiệp vụ**

Hệ thống nghiệp vụ là một loại hệ thống, do đó quá trình tiếp cận mô hình hoá cũng tuân theo quy trình chung qua nhiều giai đoạn. Tài liệu này sẽ giới thiệu hai giai đoạn mô hình hoá nghiệp vụ sử dụng UML.

- Phân tích quy trình nghiệp vụ: đây là giai đoạn đầu tiên của mô hình hóa nghiệp vụ giúp cho các nhà quản lý dự án hiểu rõ tình trạng tổ chức hiện tại và hoạt động của tổ chức, nắm bắt yêu cầu của người dùng và khách hàng từ đó phác thảo và giới hạn hệ thống phát triển.
- Thiết kế quy trình nghiệp vụ: đây là giai đoạn đặc tả chi tiết một bộ phận của tổ chức bằng cách mô tả luồng công việc của một hay nhiều nghiệp vụ, xác định các đối tượng làm việc và các thực thể nghiệp vụ trong biểu diễn hiện thực hóa nghiệp vụ và sắp xếp các hành vi của nghiệp vụ đồng thời xác định các trách nhiệm, thao tác, thuộc tính và mối quan hệ giữa các người làm việc và các thực thể trong nghiệp vụ.



## Phân tích quy trình nghiệp vụ

Các công việc của quy trình phân tích nghiệp vụ bao gồm:

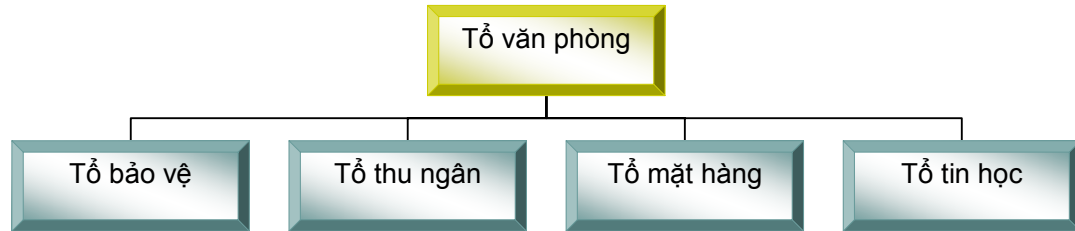
- Đánh giá và nắm bắt thông tin về tổ chức.
- Xác định các đối tượng liên quan (stakeholder) và khách hàng của hệ thống.
- Định nghĩa phạm vi của việc mô hình hóa nghiệp vụ.
- Tán thành những tiềm năng cải tiến và các mục tiêu mới của tổ chức.
- Mô tả những mục tiêu chính của tổ chức.

### Nắm bắt thông tin về tổ chức

Để thiết kế hệ thống phù hợp với nhu cầu của khách hàng thì việc hiểu rõ thông tin về cấu trúc tổ chức sẽ được triển khai hệ thống là điều quan trọng. Tất cả các thành viên trong dự án đều cần phải nắm bắt rõ ràng các thông tin này. Chúng ta có thể mô tả ngắn gọn các bộ phận cấu thành tổ chức và mối quan hệ giữa các bộ phận này thông qua các sơ đồ tổ chức và trình bày ngắn gọn các thông tin liên quan.

Để minh họa cho nội dung mô hình nghiệp vụ, giáo trình này đưa ra một hệ thống nghiệp vụ về hoạt động của một siêu thị XYZ. Tất cả các ví dụ sẽ được minh họa dựa trên việc phân tích các hoạt động nghiệp vụ của siêu thị này.

Sơ đồ tổ chức của siêu thị XYZ



**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.

**Tổ bảo vệ:** Kiểm tra, bảo vệ an ninh của Siêu Thị, ghi nhận Hàng Hóa đổi lại của khách hàng.

**Tổ thu ngân:** Thực hiện việc bán hàng và lập hóa đơn cho khách hàng đồng thời ghi nhận lại số hàng hoá bán được của mỗi loại để báo cáo cho tổ quản lý sau mỗi ca làm việc.

**Tổ mặt hàng:** Nhiệm vụ của tổ là kiểm tra chất lượng hàng hoá và nắm tình trạng hàng hoá của siêu thị, đảm bảo hàng hoá luôn ở trong tình trạng tốt nhất khi đến tay khách hàng. Khi phát hiện hàng hư hỏng phải kịp thời báo ngay cho tổ văn phòng để có biện pháp giải quyết và điều phối hàng. Ngoài ra, thường xuyên thống kê số lượng hàng tồn trên quầy, báo cáo về tổ văn phòng

**Tổ tin học:** Thực hiện việc nhập liệu, kết xuất các báo cáo cần thiết phục vụ cho tổ Văn Phòng.

### Xác định các đối tượng có liên quan và khách hàng

Việc tin học hóa công tác quản lý trong một tổ chức tạo sẽ ra những biến đổi, phần do việc tự động hóa công việc hành chính, phần do kiến thiết lại tổ chức và sự vận hành của hệ thống. Những thay đổi quan trọng phát sinh từ việc thiết kế hệ thống thông tin, chủ yếu tập trung vào việc tin học hóa, nếu không biết thực hiện dần dần sẽ có nguy cơ lớn chạm đến con người trong tổ chức dẫn đến thất bại ngay từ đầu. Bản thân công việc thiết kế hệ thống thông tin đã được thực hiện dưới nhiều góc độ khác nhau, thậm chí kể cả tâm lý, với những đặc thù riêng biệt và có độ phức tạp cao. Chính vì thế, cần phải tìm hiểu những đối tượng có liên quan và khách hàng của hệ thống là ai, đồng thời nắm bắt được nhu cầu của họ.

Nếu đánh giá tình trạng của tổ chức, ta nên xác định những đối tượng có liên quan trong nghiệp vụ. Nhưng khi xác định các mục tiêu của hệ thống thì cần xác định những đối tượng liên quan trong phạm vi dự án và điều đó cũng phụ thuộc vào phạm vi mô hình hóa nghiệp vụ, cũng như những phạm vi nào cần xác định đối với việc mô hình hóa.

Ví dụ: thể hiện thông tin các đối tượng liên quan và người dùng hệ thống qua bảng sau trong hệ thống siêu thị như sau:

Bảng mô tả tóm tắt các đối tượng liên quan:

Tên	Đại diện	Vai trò
Người quản lý	Giám đốc, người quản lý siêu thị	Theo dõi tiến trình phát triển của dự án và theo dõi tình hình hoạt động của siêu thị.
Nhân viên bán hàng	Người nhập các thông tin trong hệ thống.	Chịu trách nhiệm trong khâu bán hàng ở siêu thị, duy trì hoạt động của siêu thị.

Bảng mô tả tóm tắt các người dùng:

Tên	Mô tả	Đối tượng liên quan
Người quản lý	Đáp ứng các nhu cầu quản lý siêu thị như hàng hóa, khách hàng, doanh số.	Người quản lý
Nhân viên bán hàng	Đảm bảo rằng hệ thống sẽ đáp ứng các nhu cầu của công việc bán hàng.	Nhân viên bán hàng
Khách hàng	Đáp ứng nhu cầu tra cứu thông tin về hàng hóa có trong siêu thị.	

### Nắm bắt nhu cầu của các đối tượng liên quan

Nhiệm vụ trao cho họ là những công việc thực sự như là xử lý thông tin, chứ không chỉ đơn thuần là thao tác với máy tính và các thiết bị, vì vậy ta không được phép bỏ qua các ý kiến, nhu cầu của họ đối với hệ thống tin học tương lai. Hãy liệt kê danh sách các nhu cầu chính bằng cách điền đủ thông tin vào bảng sau:

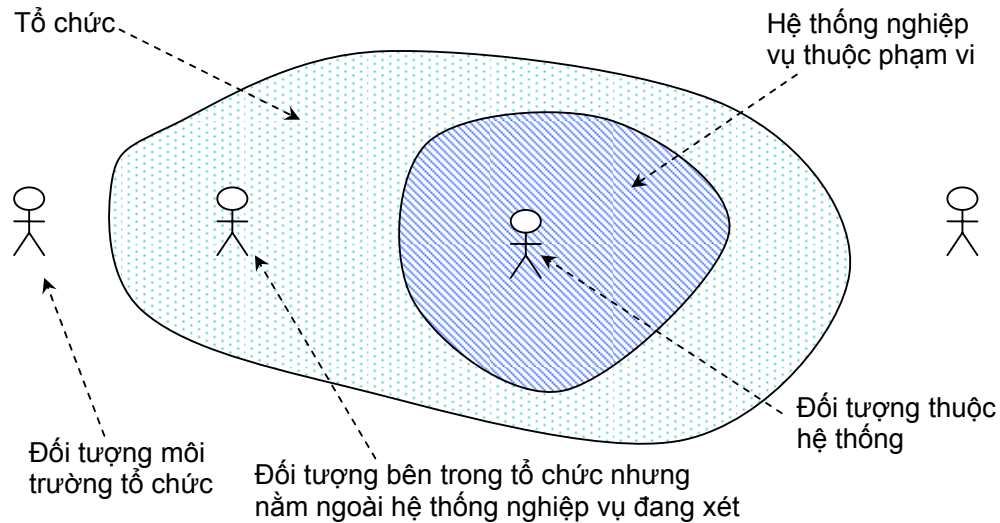
Tên đối tượng liên quan/ khách hàng	Độ ưu tiên	Nhu cầu	Giải pháp hiện hành	Giải pháp đề xuất

Ví dụ:

Tên đối tượng liên quan/ khách hàng	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	Hiện 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

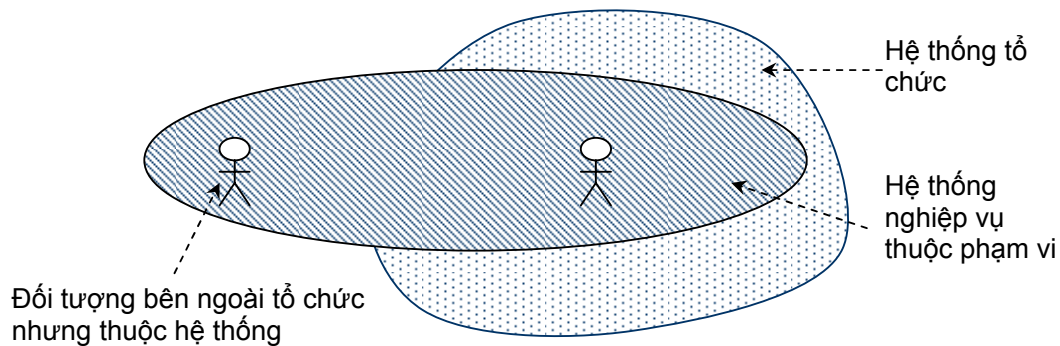
Cần phải đạt được sự thỏa thuận về những thực thể chính nằm ngoài hệ thống với các đối tượng liên quan và các thực thể này với nhau. Trong trường hợp mô hình hóa nghiệp vụ để xác định các yêu cầu cho một hệ thống cụ thể, có thể có những phần trong tổ chức sẽ không bị ảnh hưởng bởi hệ thống này, những phần đó có thể được xem như các thực thể nằm bên ngoài.



Những ranh giới đặt ra cho hệ thống có thể khác rất nhiều so với những gì có thể được xem là ranh giới của tổ chức.

Nếu mục đích là xây dựng một hệ thống mới là để hỗ trợ bán hàng, ta không cần quan tâm đến bất cứ việc gì trong kho hàng, nhưng cần xem kho hàng như là một tác nhân bởi vì chúng ta cần phải làm rõ ranh giới giữa chúng. Trong ví dụ này, các thực thể bên trong tổ chức được xem như là bên ngoài hệ thống đang xét và được mô hình hóa thành tác nhân nghiệp vụ.

Nếu mục tiêu xây dựng hệ thống là nhằm nâng cao khả năng trao đổi thông tin với các đối tác hay các nhà cung cấp (ứng dụng business-to-business) như quản lý đặt hàng thì các đối tác hay các nhà cung cấp này của tổ chức được mô hình hóa cần phải được quan tâm. Trong trường hợp này, các thực thể bên ngoài tổ chức sẽ nằm trong tổ chức. Điều này chỉ xảy ra khi sự cộng tác giữa các bên ảnh hưởng sâu sắc đến phương thức hoạt động của nhau. Nhưng nếu sự ảnh hưởng này không quá lớn hay nghiêm trọng thì các đối tác được xem như là các thực thể bên ngoài và được mô hình hóa thành tác nhân nghiệp vụ.



Nếu mục đích là xây dựng các ứng dụng tổng quát, tùy biến (như là ứng dụng kế toán tài chính, các ứng dụng đóng gói) thì chúng ta cần trình bày cách thức khách hàng sẽ sử dụng sản phẩm cuối như thế nào và nó là một thực thể trừu tượng.

**Xác định và trình bày các vấn đề của hệ thống**

Trong quá trình khảo sát hệ thống, có thể thu thập rất nhiều nhu cầu cần thay đổi của khách hàng. Đây được xem là các vấn đề của khách hàng cần chúng ta giải quyết trong hệ thống tương lai. Vì thế ta cần phải hiểu và trình bày rõ ràng các vấn đề này để mọi thành viên trong dự án nắm bắt tốt. Có thể áp dụng mẫu như sau:

Vấn đề	mô tả vấn đề
Đối tượng chịu tác động	các nhân vật bị ảnh hưởng bởi vấn đề
Ảnh hưởng của vấn đề	tác động ảnh hưởng của vấn đề
Một giải pháp thành công	liệt kê một vài lợi ích của một giải pháp thành công

Ví dụ:

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.

**Xác định những lãnh vực cần ưu tiên**

Cần phải thảo luận và đạt được sự nhất trí về những lãnh vực cần được ưu tiên trong mô hình hóa nghiệp vụ. Sự thảo luận này có thể theo nhiều hướng khác nhau, tùy vào mục tiêu của mô hình hóa nghiệp vụ:

- Nếu mục đích mô hình hóa nghiệp vụ là tạo một mô hình để thực hiện sự cải tiến đơn giản, thì chỉ cần mô tả nghiệp vụ hiện tại. Khi đó, những lãnh vực nào cần cải tiến phải xác định rõ.
- Nếu mục đích là tạo một nghiệp vụ mới hay thay đổi hoàn toàn nghiệp vụ hiện tại, thì phạm vi mô hình hóa sẽ lớn hơn. Lúc này, công việc tái cấu trúc các use case nghiệp vụ của một nghiệp vụ đã tồn tại hay thêm các use case nghiệp vụ mới - để tái thiết kế nghiệp vụ (business reengineering) hay thiết kế mới nghiệp vụ (business creation) là cần thiết.

Ví dụ: Trong Hệ thống quản lý nghiệp vụ bán hàng tại siêu thị, việc mô hình hóa nghiệp vụ nhằm mục đích để cải tiến nghiệp vụ nên chúng ta chỉ cần xác định những nghiệp vụ cần cải tiến.

Để cải tiến nghiệp vụ, một số câu hỏi được đặt ra như sau:



- Cấu trúc của tổ chức có thể được cải tiến không? Đó là cách thức tổ chức nhân viên làm việc trong các quy trình nghiệp vụ. Ta có thể xây dựng các nhóm nhân viên có nhiều năng lực khác nhau để thực hiện những công việc chính, giảm số lượng người giữ vai trò một công việc dẫn đến giảm chi phí, giảm các sai sót và để cho các nhân viên có nhiều trách nhiệm hơn, khi đó họ không phải chờ người khác quyết định.
- Có công việc nào không cần thiết không? Xác định những công việc không cần thiết trong tổ chức như: viết báo cáo mà không có ai đọc, lưu trữ những thông tin không bao giờ được sử dụng ....
- Có công việc nào giống hoặc tương tự nhau được thực hiện ở những nơi khác nhau không? Như công việc được làm lại, do người ta không tin tưởng vào kết quả hoặc không biết trước đó đã làm gì hay các kết quả được kiểm tra và chấp thuận nhiều lần.
- Có vấn đề nào về thời gian và chi phí không? Thời gian thực hiện có thể là một vấn đề thậm chí nếu mỗi thứ đều hoạt động tốt. Để xác định công việc nào có thời gian quá cấp bách, hãy phân tích mỗi use case nghiệp vụ sử dụng thời gian. Xác định mối quan hệ giữa thời gian sản xuất, thời gian chờ, và thời gian truyền.

Kết quả chính của hoạt động này là một bản mô tả tầm nhìn nghiệp vụ (Business Vision), trong đó mô tả tầm nhìn của hệ thống tương lai.

Bảng tầm nhìn nghiệp vụ xác định một tập hợp các mục tiêu của công việc mô hình hóa nghiệp vụ, cung cấp đầu vào cho quy trình kiểm chứng dự án, có liên quan mật thiết với các trường hợp nghiệp vụ (Business Case), cũng như tài liệu tầm nhìn của công nghệ phần mềm. Nó được sử dụng bởi các nhà quản lý, những người có thẩm quyền về ngân quỹ, những người làm việc trong mô hình hóa nghiệp vụ, và các nhà phát triển nói chung.

Sưu liệu này phải bảo đảm rằng:

- Nó phải được cập nhật và được phân phối.
- Nó phải giải quyết được đầu vào từ tất cả các đối tượng có liên quan.

### Xác định và mô tả các thuật ngữ nghiệp vụ

Một trong những khó khăn của dự án phần mềm quản lý hệ thống thông tin là sự bất đồng ngôn ngữ diễn đạt vấn đề giữa khách hàng và quản trị dự án hay giữa các thành viên tham gia trong dự án. Điều này gây ra các khó khăn trong việc tìm hiểu hay hiểu lầm các quy trình nghiệp vụ trong tổ chức của các thành viên trong dự án. Nhằm tránh những rủi ro này, chúng ta cần phải xác định và thống nhất những thuật ngữ trong các quy trình nghiệp vụ của tổ chức.

Sưu liệu thuật ngữ này chỉ thực sự hữu ích khi cần phân biệt rõ những từ chuyên môn của nghiệp vụ được dùng trong việc mô hình hóa nghiệp vụ với các từ chuyên môn của nghiệp vụ được dùng trong quá trình phát triển phần mềm.

Thông thường, mỗi thuật ngữ được mô tả như một danh từ với định nghĩa của nó. Tất cả các bên tham gia phải thống nhất với nhau về định nghĩa của các thuật ngữ này.

Ví dụ: Bảng thuật ngữ của hệ thống quản lý siêu thị như sau

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.
Tên đăng nhập	Tên đăng nhập của người sử dụng hệ thống. Mỗi nhân viên khi vào làm trong siêu thị sẽ được đăng ký một tên đăng nhập nhằm để quản lý. Khi đăng nhập vào hệ thống, nhân viên đó sẽ sử dụng tên này để đăng nhập. Người quản lý chịu trách nhiệm quản lý tên đăng nhập của nhân viên. Tồn tại duy nhất.
Mật khẩu	Mật khẩu đăng nhập của người sử dụng hệ thống. Mỗi nhân viên khi sử dụng tên đăng nhập sẽ được đăng ký kèm theo một mật khẩu đăng nhập. Mỗi nhân viên chỉ được biết duy nhất một mật khẩu của mình. Mật khẩu có thể rỗng.
Quyền đăng nhập	Quyền đăng nhập vào hệ thống. Tùy theo quyền và chức vụ trong công ty, nhân viên có quyền đăng nhập tương ứng.
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.
Điểm thưởng	Số điểm của khách hàng thân thiết trong siêu thị được thưởng do mua vượt mức thanh toán của siêu thị.
Ngày cấp thẻ	Ngày cấp thẻ khách hàng thân thiết cho khách hàng khi họ đăng ký chương trình khách hàng thân thiết của siêu thị.
Hóa đơn thanh toán	Hóa đơn tính tiền của siêu thị khi khách hàng mua hàng tại siêu thị
Chủng loại hàng	Chủng loại hàng hóa trong siêu thị, được phân chia tương ứng theo quầy hàng trưng bày trong siêu thị.
Loại hàng	Loại hàng trong siêu thị được phân chia theo tiêu chí công ty sản xuất, đơn vị tính....
Hàng hóa	Hàng hóa được bày bán trong siêu thị.
Hàng tồn	Số lượng hàng hóa còn lại trong siêu thị chưa bán được cho khách hàng.
Mức giảm	Tỉ lệ phần trăm giảm đối với khách hàng thân thiết
Thống kê doanh thu	Báo cáo thống kê tình hình kinh doanh của siêu thị theo tiêu chí nào đó như: hàng hóa, quý, khoảng thời gian....
Thống kê hàng hóa	Báo cáo thống kê số lượng hàng hóa của siêu thị theo tiêu chí nào đó như: hàng hóa, quý, khoảng thời gian....

### Xác định tác nhân và use case nghiệp vụ

Mục đích:

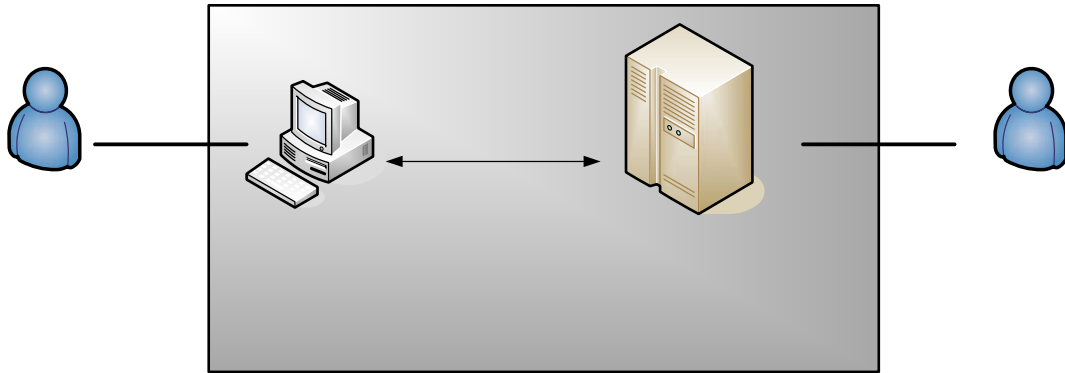
- Phác thảo các qui trình trong nghiệp vụ.
- Xác định ranh giới của nghiệp vụ cần được mô hình hóa.
- Xác định những gì sẽ tương tác với nghiệp vụ.
- Tạo ra các lược đồ của mô hình use-case nghiệp vụ.

### Tác nhân (actor) trong môi trường nghiệp vụ

Để hiểu rõ được mục tiêu của nghiệp vụ, cần phải biết nghiệp vụ tương tác với những ai; nghĩa là ai đang yêu cầu hay quan tâm đến đầu ra của nó. Những ai đó này được biểu diễn như là các tác nhân nghiệp vụ (business actor).

Thuật ngữ **tác nhân** trong trường hợp này ám chỉ vai trò mà một người hay một thứ gì đó nắm giữ trong khi tương tác với nghiệp vụ. Những loại người dùng nghiệp vụ sau đây có khả năng được xem là những tác nhân nghiệp vụ: khách hàng, nhà cung cấp, đối tác, đồng nghiệp ở những nghiệp vụ không được mô hình hóa ...

Như vậy, một tác nhân thường tương ứng với con người. Tuy nhiên, có những tình huống, chẳng hạn như một hệ thống thông tin đóng vai trò của một tác nhân. Ví dụ, ngân hàng có thể quản lý hầu hết các giao dịch trực tuyến từ một máy tính thì các use case của hệ thống sẽ tương tác với ngân hàng, khi đó ngân hàng được xem là một tác nhân, điều đó có nghĩa tác nhân lúc này là một hệ thống thông tin.



Một tác nhân biểu diễn một loại người dùng cụ thể hơn là một người dùng thực tế. Nhiều người dùng thực tế của một nghiệp vụ có thể chỉ giữ một vai trò của tác nhân; nghĩa là, họ được xem như là các thể hiện của cùng một tác nhân. Hoặc một người dùng có thể giữ nhiều vai trò tác nhân khác nhau; nghĩa là cùng một người có thể là thể hiện của các tác nhân khác nhau.

Cách thức đặt tên các tác nhân nghiệp vụ: tên của một tác nhân nghiệp vụ cần phản ánh vai trò nghiệp vụ của nó, đồng thời có thể áp dụng được với bất cứ ai - hay bất cứ hệ thống thông tin nào - đóng vai trò ấy.

#### Tiêu chí đánh giá những thừa tác viên chuẩn:

Mỗi sự vật tương tác trong môi trường nghiệp vụ - cả con người và máy móc - đều được mô hình hóa bởi các tác nhân. Không thể chắc chắn tìm thấy tất cả tác nhân cho đến khi tất cả use case được tìm ra và được mô tả đầy đủ.

Mỗi tác nhân "người" diễn tả một vai trò, chứ không phải một người cụ thể. Chúng ta phải chỉ rõ ít nhất hai người có thể có vai trò của mỗi tác nhân. Nếu không, ta có thể đang mô hình hóa một người, chứ không phải một vai trò. Dĩ nhiên là có những tình huống chỉ tìm thấy một người có thể đóng một vai trò.

Mỗi tác nhân mô hình hóa một sự vật ở bên ngoài nghiệp vụ.

Mỗi tác nhân có liên quan đến ít nhất một use case. Nếu một tác nhân không tương tác với ít nhất một use case, thì nên loại bỏ nó đi.

Một tác nhân cụ thể không tương tác với nghiệp vụ theo nhiều cách khác nhau hoàn toàn. Nếu một tác nhân tương tác theo nhiều cách khác nhau hoàn toàn, thì một tác nhân có thể có nhiều vai trò khác nhau. Trong trường hợp đó, tác nhân đó được chia thành nhiều actor, mỗi cái biểu diễn cho một vai trò khác nhau.

Mỗi tác nhân có một cái tên và mô tả rõ ràng. Tên của tác nhân cần trình bày vai trò nghiệp vụ của nó, tên này phải dễ hiểu cho những người không nằm trong nhóm mô hình hóa nghiệp vụ.

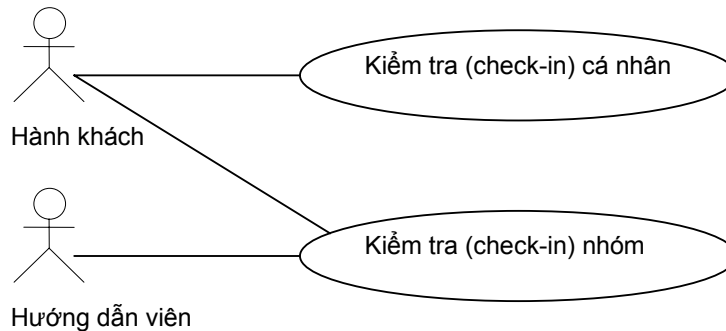
### Xác định use case nghiệp vụ

Các qui trình của một nghiệp vụ được xác định thành một số các use case nghiệp vụ khác nhau, mỗi cái biểu diễn một luồng công việc cụ thể trong nghiệp vụ. Một use case nghiệp vụ xác định những gì xảy ra trong nghiệp vụ khi nó được thực hiện; nó mô tả sự thực thi một chuỗi các hành động nhằm tạo ra một kết quả có giá trị cho một tác nhân cụ thể.

Tên của use case cần diễn tả những gì xảy ra khi một thể hiện use case được thực hiện. Do đó, tên cần ở dạng chủ động, thông thường là một động từ kết hợp với một danh từ.

Tên có thể mô tả các hoạt động trong use case từ góc nhìn bên ngoài hoặc bên trong, ví dụ: *đặt hàng* hay *nhận đặt hàng*. Cho dù một use case mô tả những gì xảy ra bên trong nghiệp vụ, cách tự nhiên nhất vẫn là đặt tên use case từ góc nhìn của tác nhân chủ chốt trong use case đó. Một khi đã quyết định theo phong cách nào, ta nên áp dụng cùng một quy tắc cho tất cả use case trong mô hình nghiệp vụ.

Ví dụ:



Một hành khách hoặc có thể đi du lịch riêng lẻ hoặc cùng với một nhóm. Khi đi du lịch cùng với một nhóm, sẽ có một hướng dẫn viên du lịch cùng đi và việc check-in có thể được thực hiện cho một đoàn bởi hướng dẫn viên hoặc bởi một hành khách đại diện.

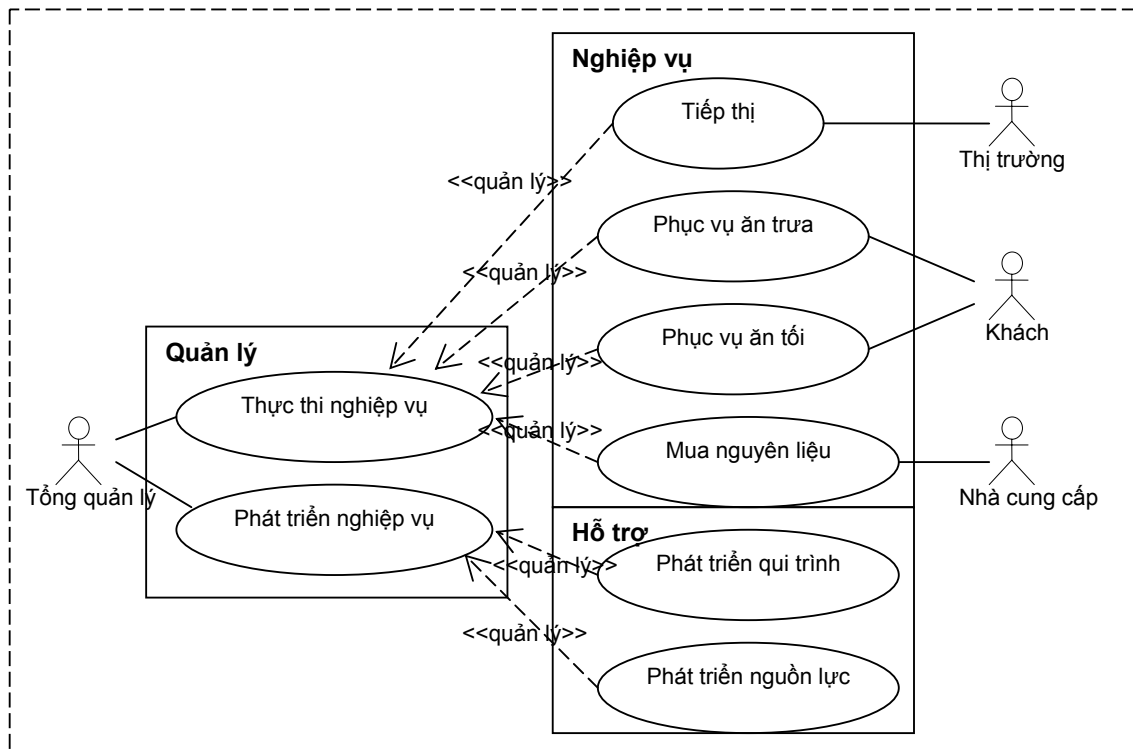
### Phân loại use case nghiệp vụ

Khi nhìn vào các hoạt động trong một nghiệp vụ, ta có thể xác định tối thiểu ba loại công việc tương ứng với ba loại use case sau:

- Các hoạt động liên quan đến công việc của tổ chức, thường được gọi là các qui trình nghiệp vụ.
- Nhiều hoạt động không liên quan đến công việc của tổ chức, nhưng phải được thực hiện theo một cách nào đó để làm cho nghiệp vụ hoạt động. Ví dụ như quản trị hệ thống, dọn dẹp, an ninh,... Các use case này mang đặc điểm hỗ trợ.
- Công việc quản lý: các use case có đặc điểm quản lý cho thấy những loại công việc ảnh hưởng đến cách thức quản lý các use case khác và các mối quan hệ của nghiệp vụ với những chủ nhân của nó.

Thông thường, một use case quản lý mô tả tổng quan về các mối quan hệ giữa nhà quản lý với những nhân viên làm việc trong các use case. Nó cũng mô tả cách thức phát triển và khởi tạo các use case.

Ví dụ: các loại use case nghiệp vụ của một tổ chức nhà hàng



Lưu ý rằng một use case nghiệp vụ quan trọng đôi khi có thể là một use case nghiệp vụ hỗ trợ trong một nghiệp vụ khác. Ví dụ: phát triển phần mềm là một use case nghiệp vụ quan trọng của một công ty phát triển phần mềm, trong khi đó nó được phân loại thành một use case nghiệp vụ hỗ trợ trong một ngân hàng hay một công ty bảo hiểm.

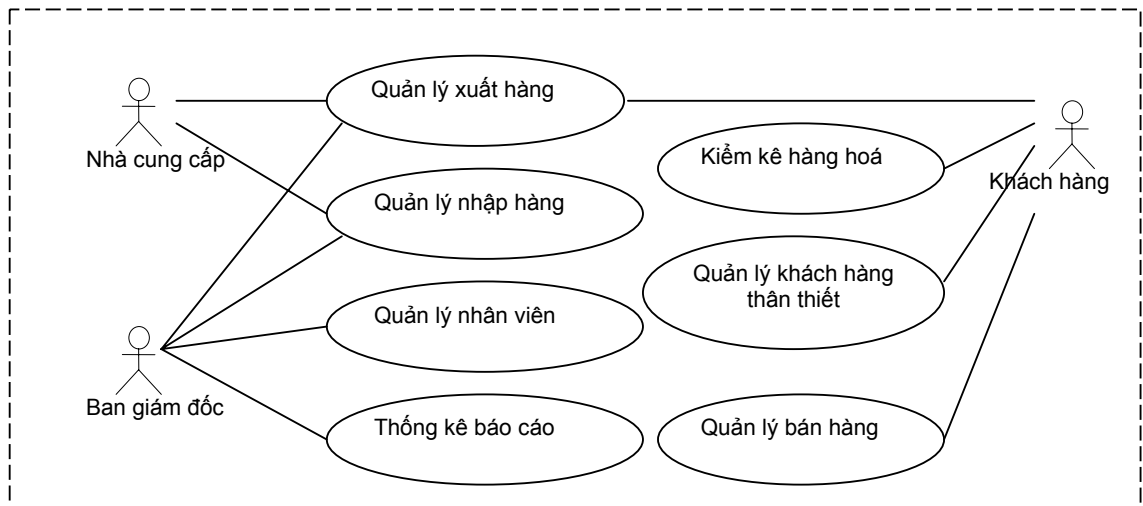
#### Qui mô của một use case nghiệp vụ

Đôi khi khó quyết định được một dịch vụ là một, hay nhiều use case nghiệp vụ. Áp dụng định nghĩa của một use case nghiệp vụ cho qui trình đăng ký chuyến bay. Một hành khách đưa vé và hành lý cho nhân viên đăng ký, nhân viên này sẽ tìm một chỗ ngồi cho hành khách, in ra thẻ lên máy bay và bắt đầu xử lý hành lý. Nếu hành khách có một hành lý thông thường, nhân viên đăng ký sẽ in ra thẻ đánh dấu hành lý và thẻ kiểm soát hành khách, cuối cùng kết thúc use case nghiệp vụ bằng cách gắn thẻ đánh dấu cho hành lý, đưa thẻ kiểm soát cùng với thẻ lên máy bay cho hành khách. Nếu hành lý là một dạng đặc biệt hay chứa những thứ đặc biệt không thể vận chuyển một cách bình thường, hành khách phải mang nó đến một quầy hành lý đặc biệt. Nếu hành lý quá nặng, hành khách phải tiếp tục đến văn phòng vé máy bay để trả tiền, bởi vì các nhân viên đăng ký không xử lý việc đóng tiền.

Câu hỏi đặt ra là có cần một use case nghiệp vụ tại quầy đăng ký, một use case nghiệp vụ khác tại quầy hành lý đặc biệt và cái thứ ba ở văn phòng vé? Hay là chỉ cần một use case nghiệp vụ duy nhất? Chắc chắn là sự giao dịch này có liên quan đến ba loại hành động khác nhau. Nhưng câu hỏi ở đây là có một hành động nào đó sẽ có ý nghĩa đối với hành khách mang hành lý đặc biệt nếu hành khách này không thực hiện những hành động còn lại? Câu trả lời là không có, nó chỉ là một thủ tục hoàn chỉnh - từ lúc hành khách đến quầy đăng ký đến khi ông ta trả thêm phí phụ thu (chỉ có giá trị hay có ý nghĩa đối với hành khách). Như vậy, thủ tục hoàn chỉnh có liên quan đến ba quầy khác nhau chính là một trường hợp sử dụng hoàn chỉnh, tức là một use case nghiệp vụ.

Ngoài tiêu chí này, điều quan trọng là cần giữ mô tả của các dịch vụ có liên quan mật thiết này cùng với nhau, để sau này có thể xem lại chúng cùng một lúc, điều chỉnh, kiểm tra và viết hướng dẫn cho chúng, và nói chung là quản lý chúng như một đơn vị.

Kết quả của quá trình tiếp cận phân tích nghiệp vụ là (các) sơ đồ use case nghiệp vụ và các mô tả của use case.



Mô hình use case mô tả nghiệp vụ của siêu thị

## Bài tập

### 1.1 Cho một hệ thống được mô tả như sau:

Government Solutions Company (GSC) là công ty chuyên bán các trang thiết bị cho các cơ quan chính phủ liên ban. Khi một cơ quan cần mua trang thiết bị từ GSC, cơ quan này sẽ phát sinh một đơn đặt hàng dựa trên một hợp đồng chuẩn trước đó đã được thoả thuận với công ty. GSC quản lý các hợp đồng với các cơ quan liên bang. Khi một đơn đặt hàng gửi tới nhân viên quản lý hợp đồng của GSC, nhân viên này xem xét lại các thông tin về các điều khoản và điều kiện của hợp đồng đã ký với cơ quan bằng cách sử dụng mã số hợp đồng được tham khảo trong đơn đặt hàng để tìm kiếm trong cơ sở dữ liệu của GSC rồi so với thông tin của đơn đặt hàng để xác định đơn đặt hàng có hợp lệ hay không. Đơn đặt hàng hợp lệ nếu hợp đồng chưa hết hạn, danh sách trang thiết bị phải thuộc các thiết bị của hợp đồng, và tổng chi phí phải không vượt quá giới hạn xác định trước. Nếu đơn đặt hàng không hợp lệ, nhân viên quản lý hợp đồng sẽ gửi trả lại đơn đặt hàng cho cơ quan kèm theo một lá thư diễn giải lý do đơn đặt hàng không hợp lệ và lưu lại một bản sao của lá thư.

Nếu đơn đặt hàng hợp lệ, nhân viên quản lý hợp đồng lưu vào cơ sở dữ liệu đơn đặt hàng đó cùng với trạng thái là “chưa giải quyết”. Sau đó, đơn đặt hàng sẽ được gửi đến bộ phận đáp ứng đơn hàng, bộ phận này sẽ kiểm tra tồn kho cho mỗi thiết bị dựa trên thông tin tồn kho lấy từ cơ sở dữ liệu. Nếu có bất kỳ một thiết bị nào không đủ số giao, bộ phận này sẽ tạo ra một báo cáo liệt kê tất cả các mặt hàng không đủ giao cùng với số lượng thiếu.

Tất cả các đơn đặt hàng sẽ được chuyển tới kho để thực hiện việc giao hàng cho cơ quan. Sau khi giao hàng, tại đây sẽ lập một hoá đơn giao hàng ghi nhận các thiết bị được giao gửi cho cơ quan, và đính kèm một bản sao hóa đơn này với đơn đặt hàng gửi trở lại cho nhân viên quản lý hợp đồng. Nhân viên này sẽ kiểm tra nếu đơn đặt hàng đã được giao hết thì cập nhật lại trạng thái đơn đặt hàng là “hoàn tất” và lưu thông tin hoá đơn giao hàng vào cơ sở dữ liệu.

Hàng tháng, nhân viên quản lý hợp đồng sẽ lập các báo cáo các thiết bị được đặt hàng trong tháng, các thiết bị đã được giao, các đơn đặt hàng đang “chưa giải quyết”, các hợp đồng đã hết hạn gửi cho giám đốc để giúp cho giám đốc nắm được tình hình hoạt động của công ty.

- Hãy xác định các tác nhân nghiệp vụ của hệ thống
- Xác định tất cả các use case nghiệp vụ
- Xây dựng sơ đồ use case nghiệp vụ

### 1.2 Mô tả hệ thống “Quản lý cho thuê văn phòng cao ốc” như sau:

Một công ty địa ốc muốn tin học hóa hoạt động cho thuê cao ốc của mình cho các công ty làm văn phòng hoạt động kinh doanh. Công ty có nhiều cao ốc ở trong thành phố, mỗi cao ốc được quản lý bởi một tên, có một địa chỉ, mô tả đặc điểm và tổng diện tích sử dụng. Một cao ốc sẽ có nhiều tầng, mỗi tầng có nhiều phòng, mỗi phòng có các thông tin cần quản lý là: mã phòng, diện tích sử dụng, số chỗ làm việc (theo tính toán của công ty), mô tả vị trí, giá cho thuê.

#### *Hoạt động thuê phòng*

Khách hàng muốn thuê phòng thì phải đến nơi quản lý tòa nhà để tham khảo vị trí, diện tích phòng, và tại đây khách hàng được nhân viên tiếp tân cung cấp thông tin tình trạng giá cả của phòng. Giá cả mỗi phòng được ấn định tùy theo độ cao, diện tích sử dụng,...

Khách hàng sau khi đồng ý thuê thì đến gửi và trình bày yêu cầu làm hợp đồng với bộ phận quản lý nhà, bộ phận này soạn thảo hợp đồng dựa trên yêu cầu thuê muốn được cung cấp bởi khách hàng, rồi tính toán giá thuê và lập hợp đồng. Hợp đồng sau khi được ký nhận bởi khách hàng và trưởng bộ phận quản lý nhà sẽ được lưu lại một bản, và một bản sẽ gửi cho khách hàng. Khách có thể làm hợp đồng thuê cùng lúc nhiều phòng. Nội dung của hợp đồng bao gồm: số hợp đồng, ngày hiệu lực hợp đồng, ngày thanh toán đầu tiên, khách hàng, thời gian thuê và chi tiết gồm các phòng cần thuê, giá thuê. Thời gian của đợt thuê ít nhất sáu tháng và sau đó có thể gia hạn thêm. Khách phải trả trước tiền thuê của sáu tháng đầu tiên, từ tháng thứ bảy nếu có thì phải trả vào đầu mỗi tháng. Giá thuê phòng không kể chi phí điện trong đó, do đó cuối tháng khách cũng phải thanh toán các chi phí điện.

Đầu mỗi tuần, phòng kế toán kiểm tra lại tất cả hợp đồng đến hạn phải thanh toán trong tuần. Sau đó lập thông báo gửi tới các khách hàng để chuẩn bị thanh toán. Mỗi lần thanh toán, khách hàng phải đến phòng kế toán của công ty để đề nghị thanh toán. Tại đây, phòng kế toán sẽ tìm kiếm hợp đồng từ hồ sơ lưu, kiểm tra thông tin đợt thanh toán của khách hàng. Sau đó lập hoá đơn thanh toán và lưu lại một bản, một bản gửi cho khách hàng. Các thông tin trên hoá đơn sẽ là: số phiếu, ngày thanh toán, lý do thanh toán (tiền phòng hay tiền điện nước), thanh toán cho hợp đồng nào và tổng số tiền thanh toán.

#### *Gia hạn hợp đồng*

Khi gần hết hạn hợp đồng, khách hàng muốn gia hạn thêm thì phải đến yêu cầu bộ phận quản lý xin gia hạn hợp đồng. Bộ phận này sẽ kiểm tra xem thực sự đã hết hạn chưa, đã thanh toán đầy đủ chưa. Nếu các điều kiện trên được đáp ứng thì sẽ lập một bản hợp đồng gia hạn. Nội dung bản gia hạn gồm có: số hợp đồng cần gia hạn, ngày ký gia hạn, mô tả và chi tiết gia hạn gồm các phòng gia hạn (các phòng này phải được thuê bởi hợp đồng) và thời gian gia hạn. Việc thanh toán gia hạn cũng sẽ được lập theo từng tháng và một hợp đồng có thể được gia hạn nhiều lần.

#### *Hoạt động truy xuất*

Các báo cáo hằng ngày được bộ phận quản lý lập:

- Danh sách phòng đang được thuê
- Danh sách phòng đang trống
- Danh sách nhân viên đang làm việc
- Danh sách công ty hết hạn thuê trong tháng

#### *Yêu cầu*

- Hãy xác định các tác nhân nghiệp vụ của hệ thống
- Xác định tất cả các use case nghiệp vụ
- Xây dựng sơ đồ use case nghiệp vụ

## Thiết kế quy trình nghiệp vụ

### Đặc tả các use case nghiệp vụ:

Bước đầu tiên trong giai đoạn thiết kế này chính là đặc tả các use case nghiệp vụ nhằm làm rõ nội dung của các use case này. Chú ý rằng chúng ta chỉ mô tả các nội dung xử lý thứ tự luận lý giữa các xử lý này, về kết quả, cách mô tả này độc lập tương đối với một trường thực tế xử lý nó có nghĩa rằng chúng ta chỉ làm rõ phần nội dung của use case mà chưa mô tả rõ các vai trò thực hiện và các đối tượng bị tác động bởi use case.

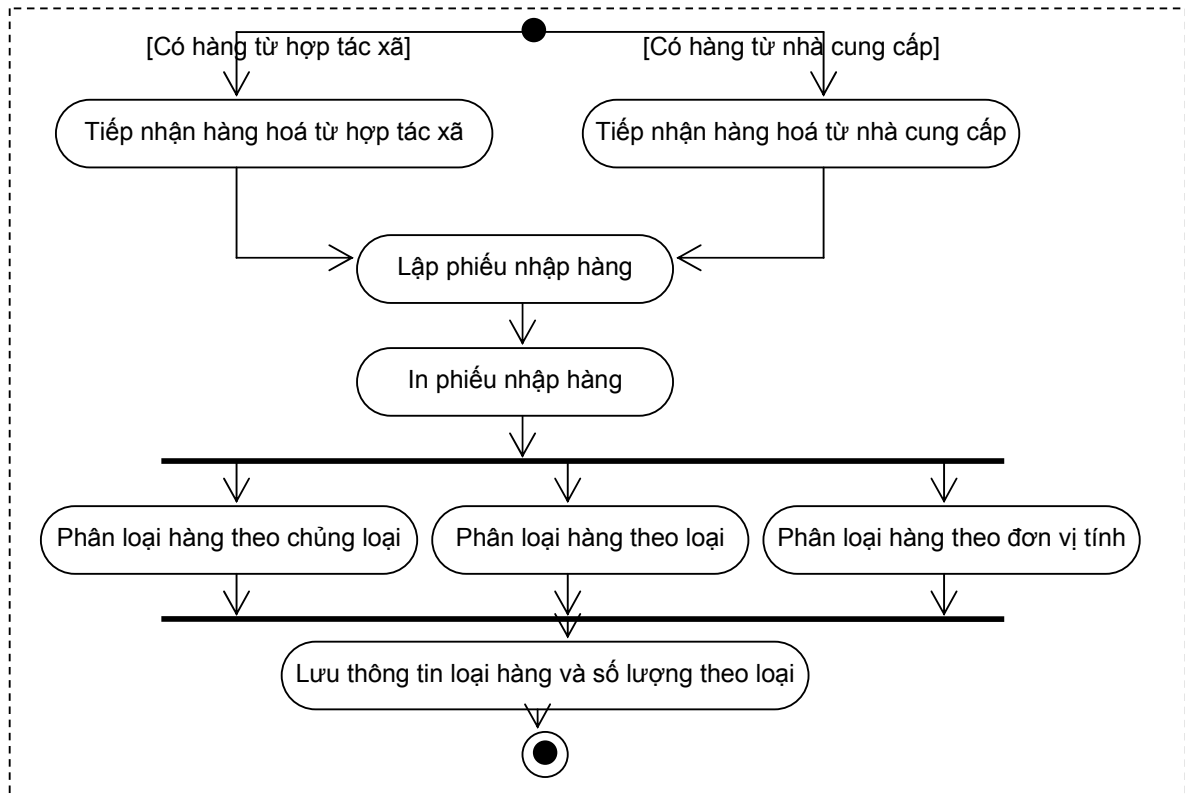
Hãy bắt đầu mô tả luồng công việc bình thường trong use case nghiệp vụ, xác định sự tương tác giữa các tác nhân và use case. Sau đó, khi luồng công việc bình thường ổn định, ta bắt đầu mô tả các luồng công việc thay thế khác. Một luồng công việc use-case nghiệp vụ được trình bày theo các cách thức đã nhất trí và tham khảo bảng chú giải chung khi viết những văn bản mô tả.

Mô tả tất cả những luồng sự kiện bất thường và luồng sự kiện tùy chọn. Mô tả một luồng sự kiện con trong phần bổ sung của luồng công việc đối với các trường hợp:

- Những luồng sự kiện con tham gia phần lớn luồng công việc chính.
- Những luồng công việc bất thường giúp luồng công việc chính rõ ràng hơn.
- Những luồng sự kiện con xảy ra ở những khoảng thời gian khác nhau trong cùng một luồng công việc và chúng có thể được thực thi.

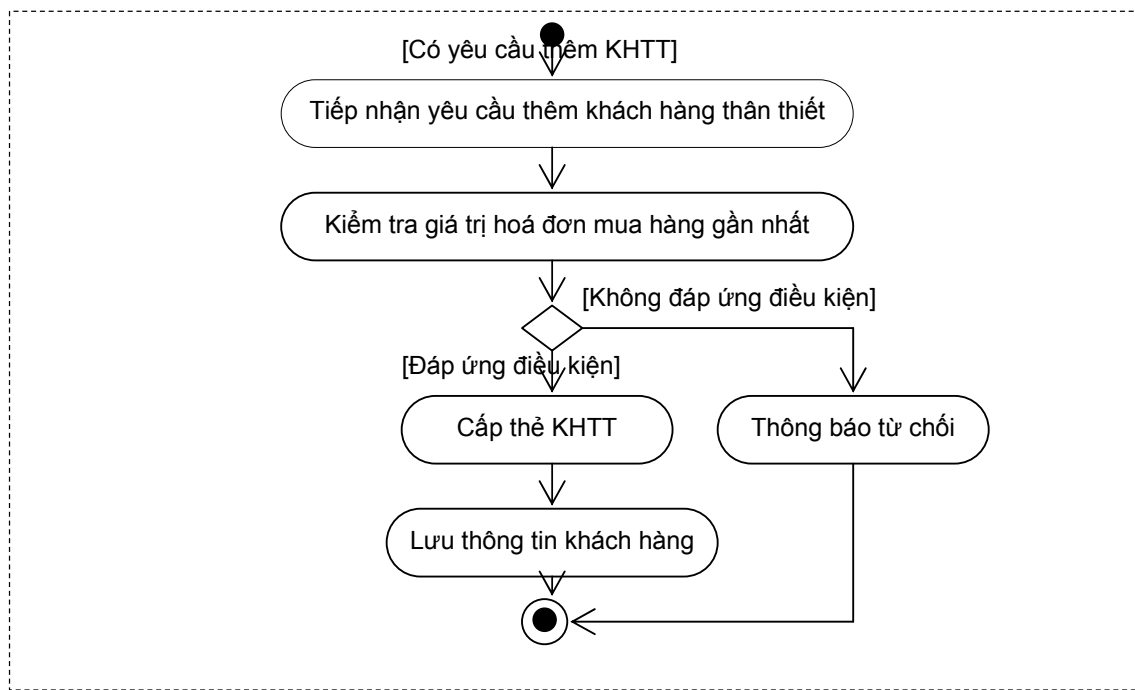
Ngoài ra, có thể đặc tả cấu trúc luồng công việc trong một sơ đồ hoạt động.

Ví dụ: sơ đồ hoạt động đặc tả use case nhập hàng



Ví dụ: sơ đồ hoạt động đặc tả use case Quản lý khách hàng thân thiết





Trong sơ đồ hoạt động trên, luồng công việc chính của use case bao gồm bốn hoạt động được vẽ trên một luồng thẳng đứng. Sau đó luồng công việc phụ như *Thông báo từ chối* được xác định bổ sung thành các nhánh.

### **Xác định các thừa tác viên nghiệp vụ (business worker) và các thực thể chịu tác động bởi nghiệp vụ (business entity)**

#### Xác định thừa tác viên nghiệp vụ

Một thừa tác viên biểu diễn sự trừu tượng của một người hoạt động trong nghiệp vụ. Một đối tượng thừa tác viên tương tác với các đối tượng thừa tác viên khác đồng thời thao tác với các đối tượng thực thể để hiện thực hóa một thể hiện use-case.

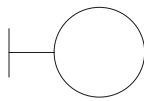
Một thừa tác viên được khởi tạo khi luồng công việc của thể hiện use-case tương ứng bắt đầu hay ngay vào lúc người có vai trò của thừa tác viên bắt đầu thực hiện vai trò đó trong thể hiện use-case. Một đối tượng thừa tác viên thường "sống" khi use case thực thi.

UML phân chia thừa tác viên thành hai loại: thừa tác viên thực hiện các công việc bên trong hệ thống và thừa tác viên tương tác trực tiếp với các tác nhân bên ngoài hệ thống. Thực sự việc phân chia này chỉ muốn chuyên biệt hơn nữa vai trò của các thừa tác viên trong việc giao tiếp với tác nhân bên ngoài.

Ký hiệu:



Thủ thư



Nhân viên bán hàng



Quản trị hệ thống

Trong các ký hiệu trên cho thấy, Nhân viên bán hàng được mô hình hoá như là một thừa tác viên giao tiếp trực tiếp với các tác nhân bên ngoài. Quản trị hệ thống được mô hình hoá là một thừa tác viên làm việc bên trong hệ thống. Thủ thư là một thừa tác viên có thể vừa làm việc bên trong và vừa chịu trách nhiệm giao tiếp với tác nhân bên ngoài, tuy nhiên, nếu được mô hình hoá theo như trên thì chúng ta muốn nhấn mạnh vai trò hoạt động bên trong hệ thống của Thủ thư hơn là vai trò giao tiếp với tác nhân bên ngoài.

#### Xác định các thực thể nghiệp vụ

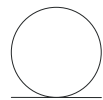
Các thực thể biểu diễn những thứ được xử lý hoặc sử dụng bởi các thừa tác viên khi chúng thực thi một use case nghiệp vụ. Một thực thể thường biểu diễn một sự vật có giá trị cho một số thể hiện use case hoặc thể hiện use case, vì vậy đối tượng thực thể sống lâu hơn. Nói chung, thực thể không nên giữ thông tin nào về cách thức nó được sử dụng bởi ai.

Một thực thể biểu diễn một tài liệu hoặc một phần thiết yếu của sản phẩm. Đôi khi nó là một thứ gì đó mơ hồ, như kiến thức về một thị trường hay về một khách hàng. Ví dụ về các thực thể tại nhà hàng là Thực đơn và Thức uống; tại phi trường là Vé và Thẻ lên máy bay (Boarding Pass) là những thực thể quan trọng.

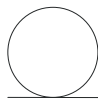
Mô hình hóa các hiện tượng thành những thực thể chỉ khi những lớp khác trong mô hình đối tượng phải tham chiếu đến các hiện tượng này. Những thứ khác có thể được mô hình hóa thành các thuộc tính của các lớp thích hợp, hay chỉ cần được mô tả bằng văn bản trong những lớp này.

Tất cả mỗi sự vật trong nghiệp vụ như: sản phẩm, tài liệu, hợp đồng, ... đều được mô hình hóa thành các thực thể nghiệp vụ, và nó tham gia vào tối thiểu một use case nghiệp vụ.

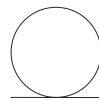
Ký hiệu:



Thực đơn



Thức uống



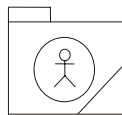
Thẻ lên máy bay

### Các khái niệm UML hỗ trợ thêm cho quá trình mô hình hoá nghiệp vụ

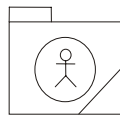
Ngoài ra, UML (phiên bản 1.5) còn bổ sung thêm một số stereotype cho phép mô hình hoá đầy đủ hơn về hệ thống nghiệp vụ:

*Đơn vị tổ chức (organization unit)*

Mô tả : tập hợp các thừa tác viên, thực thể, use case nghiệp vụ, các sơ đồ, và các đơn vị tổ chức khác. Đối tượng này được dùng để phân chia mô hình nghiệp vụ thành nhiều phần khác nhau.



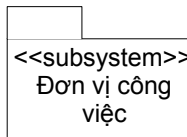
Nhập sách



Mượn sách

*Đơn vị công việc (WorkUnit)*

Mô tả : là một loại hệ thống con có thể chứa một hoặc nhiều thực thể. Nó là một tập đối tượng hướng nhiệm vụ nhằm hình thành một tổng thể có thể nhận thức được bởi người dùng cuối và có thể có một giao diện xác định cách nhìn các thực thể công việc thích hợp tới nhiệm vụ đó.



### Hiện thực hoá use case nghiệp vụ

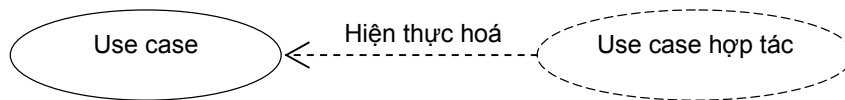
Trong một dự án mô hình hóa nghiệp vụ hướng use-case, hãy phát triển hai khung nhìn nghiệp vụ.

Use case nghiệp vụ trình bày khung nhìn bên ngoài của nghiệp vụ, qua đó xác định những gì thiết yếu cần thực hiện cho nghiệp vụ để phân phối các kết quả mong muốn cho tác nhân. Nó cũng xác định trong nghiệp vụ cần có những tương tác nào với tác nhân khi use case được

thực thi. Khung nhìn này được phát triển khi đang lựa chọn và nhất trí về những gì cần được thực hiện trong mỗi use case. Một tập hợp các use case cung cấp một cái nhìn tổng quan về nghiệp vụ, nó rất hữu ích để thông báo cho các nhân viên về những thay đổi, những điểm khác biệt của nghiệp vụ đang thực hiện, và những kết quả nào được mong muốn.

Mặt khác, một hiện thực hóa use-case cung cấp một khung nhìn bên trong về use case, qua đó xác định cách thức công việc cần được tổ chức và thực hiện như thế nào nhằm đạt được những kết quả mong muốn như trên. Một hiện thực hóa bao gồm các thừa tác viên và thực thể có liên quan đến sự thực thi một use case và các mối quan hệ giữa chúng. Những khung nhìn như vậy cần thiết cho công việc lựa chọn và thống nhất về cách thức tổ chức các công việc trong mỗi use case nhằm đạt được những kết quả mong muốn.

Cả hai khung nhìn của use case đều chủ yếu dành cho những nhân viên bên trong nghiệp vụ - khung nhìn bên ngoài dành cho những người hoạt động bên ngoài use case, khung nhìn bên trong dành cho những người hoạt động bên trong use case.



Mô hình hoá hiện use case hiện thực hoá qua việc lập cấu trúc mô hình đối tượng nghiệp vụ (business object)

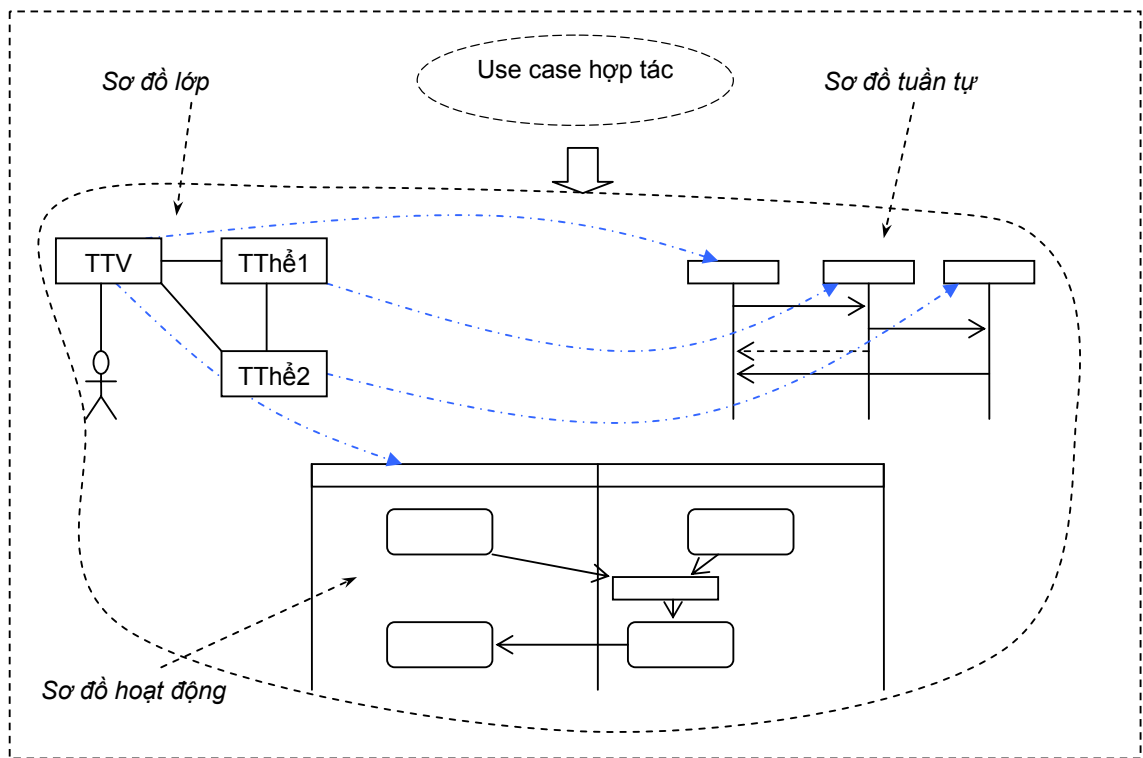
Sơ đồ đối tượng nghiệp vụ là một tập các sơ đồ nhằm trình bày sự hiện thực hóa của các use case nghiệp vụ. Nó mô tả trừu tượng cách thức các thừa tác viên và thực thể liên kết và cộng tác với nhau để thực hiện nghiệp vụ.

▪ Giải thích

Sơ đồ đối tượng xác định các use case từ góc nhìn bên trong của các thừa tác viên. Mô hình định nghĩa cách thức các nhân viên nghiệp vụ với những gì họ xử lý liên hệ với nhau để tạo ra các kết quả mong muốn. Nó nhấn mạnh vào các vai trò được thực hiện trong lĩnh vực nghiệp vụ và các trách nhiệm của nhân viên. Các đối tượng của các lớp trong mô hình cần có khả năng thực hiện tất cả use case nghiệp vụ.

Các thành phần chính của mô hình đối tượng nghiệp vụ là:

- Các thừa tác viên (worker): cho thấy các trách nhiệm của một nhân viên
- Các thực thể (entity): biểu diễn đầu ra, tài nguyên, sự vật được sử dụng
- Các hiện thực hóa use-case nghiệp vụ: cho thấy các thừa tác viên cộng tác và các thực thể thực hiện luồng công việc như thế nào. Các hiện thực hóa use-case nghiệp vụ được đặc tả với:
  - Các lược đồ lớp: là các thừa tác viên và thực thể tham gia
  - Các lược đồ hoạt động: trong đó các swimlane cho thấy các trách nhiệm của các thừa tác viên, các luồng đối tượng cho thấy cách sử dụng các thực thể trong luồng công việc.
  - Các lược đồ tuần tự: mô tả chi tiết sự tương tác giữa các thừa tác viên, tác nhân, và cách truy xuất các thực thể khi thực hiện một use case nghiệp vụ.



▪ Mục đích của mô hình đối tượng nghiệp vụ

- Là một thành phần trung gian để làm rõ các ý kiến về nghiệp vụ theo cách suy nghĩ của các nhà phát triển phần mềm, mà vẫn giữ được nội dung nghiệp vụ. Nó là sự thống nhất về lĩnh vực nghiệp vụ được mô tả dưới dạng các đối tượng, thuộc tính, trách nhiệm.
- Khảo sát bản chất của lĩnh vực nghiệp vụ nhằm chuyển tiếp lối tư duy về các vấn đề nghiệp vụ sang lối tư duy về các ứng dụng phần mềm.
- Làm rõ những yêu cầu được hỗ trợ bởi hệ thống thông tin đang xây dựng.
- Thống nhất các định nghĩa về đối tượng nghiệp vụ, các mối quan hệ giữa các đối tượng, tên các đối tượng và quan hệ. Qua đó, cho phép trình bày chính xác các kiến thức về lĩnh vực nghiệp vụ sao cho các chuyên gia về lĩnh vực nghiệp vụ có thể hiểu được.

▪ Lập cấu trúc mô hình đối tượng nghiệp vụ:

Phân tích chu kỳ sống của mỗi thực thể. Mỗi thực thể nên được tạo ra và hủy đi bởi một người nào đó trong đời sống của nghiệp vụ. Hãy bảo đảm rằng mỗi thực thể được truy xuất và sử dụng bởi một thừa tác viên hay một thực thể khác.

Cần giảm bớt số lượng các thừa tác viên. Khi phát triển các mô hình, có thể ta sẽ thấy có quá nhiều thừa tác viên. Hãy bảo đảm rằng mỗi thừa tác viên tương ứng với một tập hợp các tác vụ mà một người thường thực hiện.

Mỗi thực thể nên có một người chịu trách nhiệm cho nó. Điều này có thể được mô hình hóa bằng một mối kết hợp từ thừa tác viên đến các thực thể mà thừa tác viên đó chịu trách nhiệm. Một số thực thể có thể do những người ngoài nghiệp vụ chịu trách nhiệm. Mô tả điều này trong bản mô tả văn tắt của thực thể đó.

▪ Lược đồ lớp (class diagram)

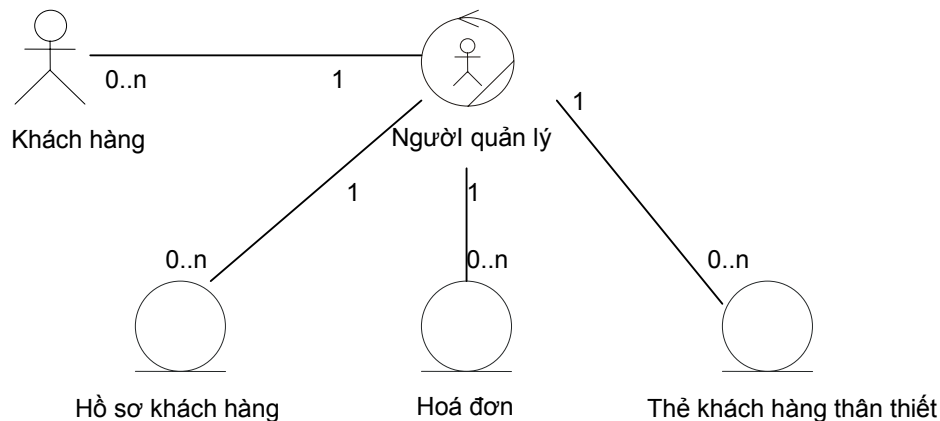
Một lược đồ lớp cho thấy một tập hợp các thành phần (tính) của mô hình, như lớp, gói, nội dung của chúng và các mối quan hệ.

Các lược đồ lớp cho thấy các mối kết hợp, kết tập và tổng quát hóa giữa thừa tác viên và thực thể. Những lược đồ lớp được quan tâm về:

- Sự phân cấp kế thừa
- Các mối kết tập của thừa tác viên và thực thể.
- Cách thức các thừa tác viên và thực thể liên quan đến nhau thông qua các mối kết hợp.

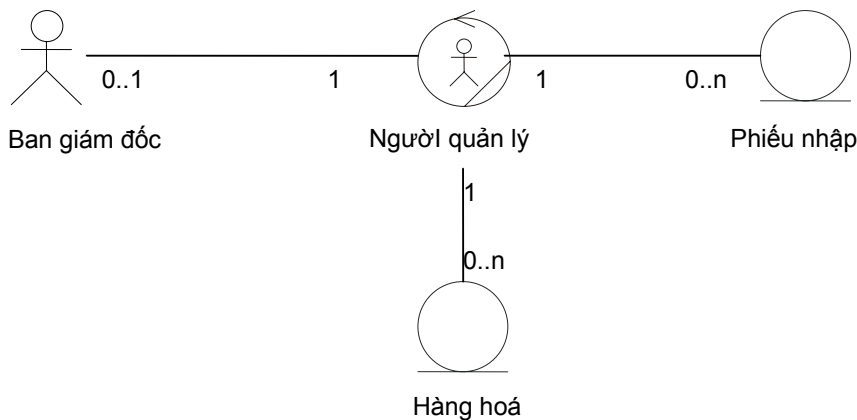
Các lược đồ lớp cho thấy các cấu trúc chung trong mô hình đối tượng nghiệp vụ, nhưng cũng có thể là một phần của tài liệu mô tả một hiện thực hóa use case bằng cách cho thấy các thừa tác viên và thực thể tham gia.

Ví dụ: sơ đồ lớp cho use case Quản lý khách hàng thân thiết cho biết các thừa tác viên, các thực thể và tác nhân liên kết với nhau trong việc thực hiện của use case này.



Trong đó, **Người quản lý** là thừa tác viên thực hiện use case. **Hồ sơ khách hàng**, **Thẻ khách hàng thân thiết** và **Hoá đơn** là ba thực thể được sử dụng trong use case này bởi thừa tác viên.

Hoặc sơ đồ lớp cho use case Quản lý nhập hàng



Trong đó, **Người quản lý** là thừa tác viên thực hiện use case. **Phiếu nhập** và **hàng hoá** là các thực thể bởi thừa tác viên này trong việc thực hiện hoạt động của use case.

### Đặc tả luồng công việc hiện thực hoá use case nghiệp vụ

#### Sử dụng sơ đồ hoạt động

Đầu tiên, để lập tài liệu hiện thực hóa cho một use case nghiệp vụ chính là vẽ một lược đồ hoạt động, trong đó các luồng (swimlane) biểu diễn các thừa tác viên tham gia. Đối với mỗi hiện thực hóa use-case, có thể có một hoặc nhiều lược đồ hoạt động để minh họa luồng công

việc. Một cách phổ biến là sử dụng một lược đồ tổng quan không có các swimlane để mô tả toàn bộ luồng công việc, trong đó trình bày các "hoạt động vĩ mô" ở mức cao. Sau đó, đối với mỗi hoạt động vĩ mô sẽ có một lược đồ hoạt động chi tiết, trình bày các luồng (swimlane) và các hoạt động ở cấp độ thừa tác viên. Mỗi lược đồ nên được gói gọn trong một trang giấy.

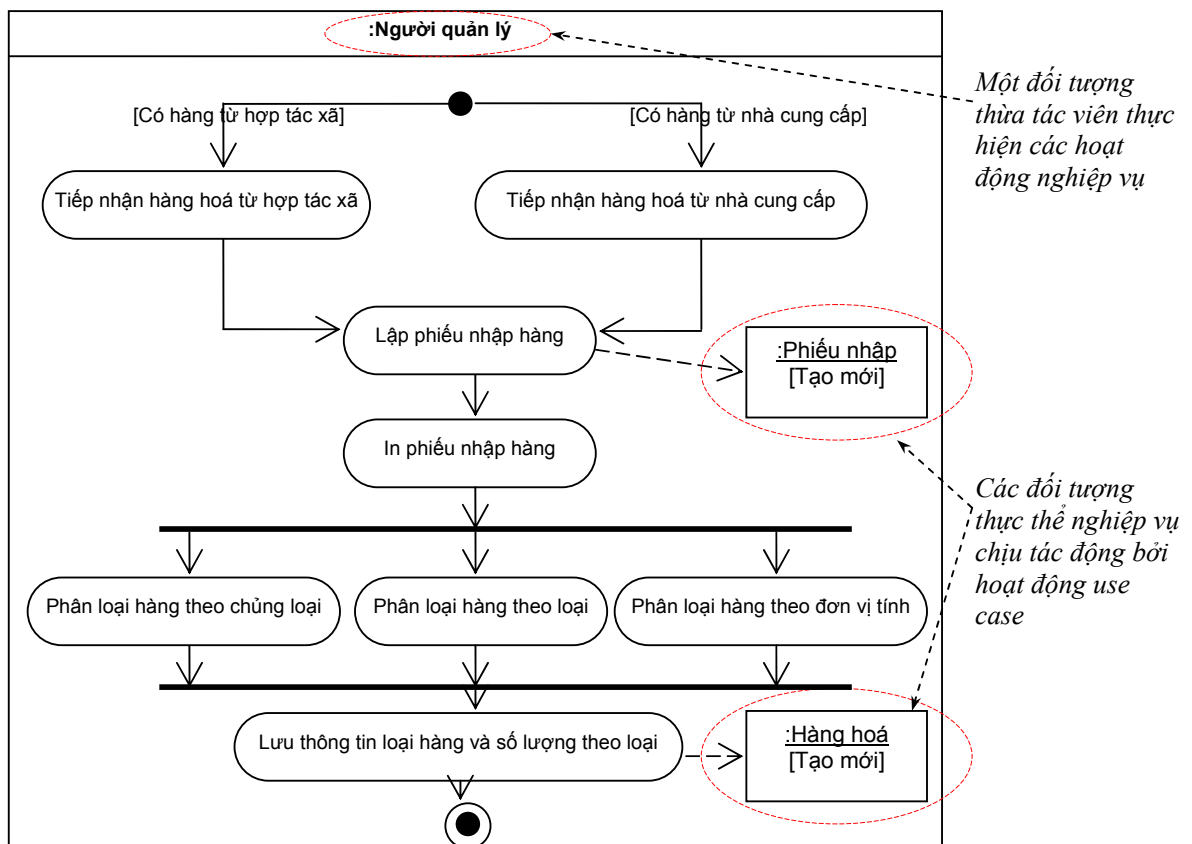
Lược đồ hoạt động trong mô hình đối tượng minh họa luồng công việc của một hiện thực hóa use-case nghiệp vụ. Lược đồ hoạt động của một hiện thực hóa use-case minh họa việc sắp xếp các công việc theo một thứ tự nhằm đạt được các mục tiêu của nghiệp vụ, cũng như thỏa mãn nhu cầu giữa các tác nhân bên ngoài và các thừa tác viên bên trong. Một hoạt động trong lược đồ hoạt động có thể là một công việc thủ công hoặc tự động hóa để hoàn thành một đơn vị công việc.

Các lược đồ hoạt động giúp:

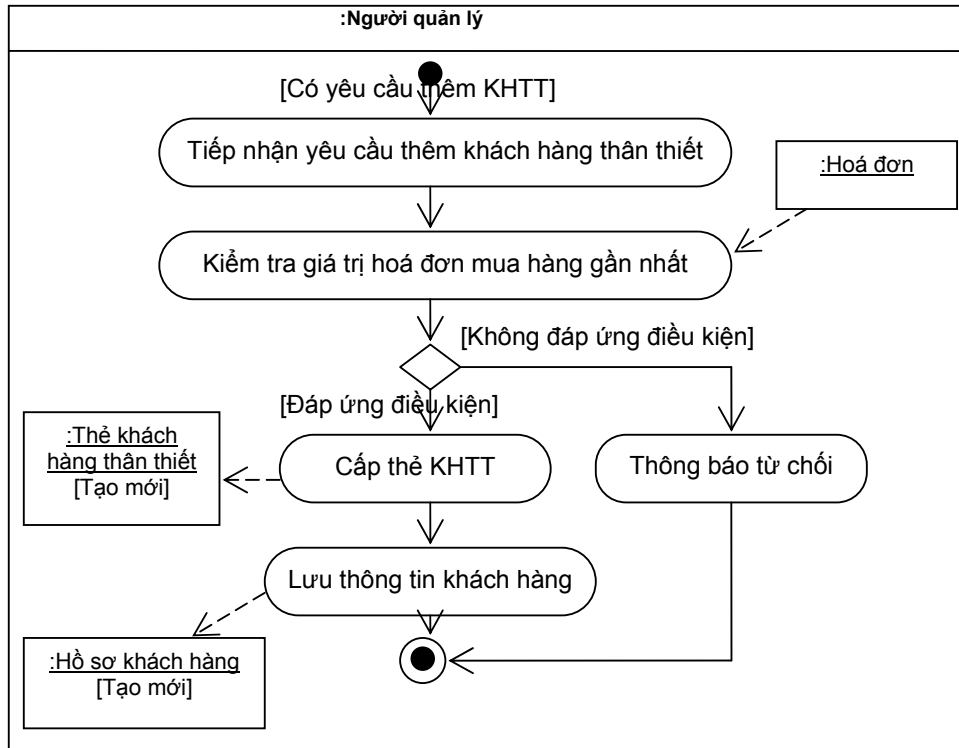
- Cung cấp cơ sở để giới thiệu các hệ thống thông tin đến doanh nghiệp một cách dễ hiểu hơn.
- Thiết lập các mục tiêu cho các dự án phát triển hệ thống nhằm cải tiến nghiệp vụ.
- Điều chỉnh mức độ đầu tư vào việc tự động hóa quy trình dựa trên các thông tin đo lường quy trình nghiệp vụ đó.

So sánh với lược đồ tuần tự có cùng mục đích thì lược đồ hoạt động tập trung mô tả cách thức phân chia trách nhiệm thành các lớp, trong khi đó, lược đồ tuần tự mô tả cách thức các đối tượng tương tác theo trình tự. Lược đồ hoạt động tập trung vào luồng công việc, trong khi lược đồ tuần tự tập trung vào việc xử lý các thực thể. Chúng bổ sung cho nhau, như lược đồ tuần tự cho thấy những gì xảy ra trong một trạng thái hoạt động.

Ví dụ: sơ đồ hoạt động hiện thực hoá use case Quản lý nhập hàng



## Sơ đồ hoạt động hiện thực hoá use case Quản lý khách hàng thân thiết



## Sử dụng các swimlane (làn bơi)

- Nếu các luồng (swimlane) được sử dụng và được nhóm thành các lớp (chủ yếu là các thừa tác viên) trong mô hình đối tượng, thì chúng ta đang sử dụng lược đồ hoạt động để trình bày các hiện thực hóa use-case nghiệp vụ.
- Lược đồ hoạt động cung cấp chi tiết về những gì xảy ra trong nghiệp vụ bằng cách khảo sát những người có các vai trò cụ thể (các thừa tác viên) và các hoạt động mà họ thực hiện. Đối với các dự án phát triển ứng dụng, các lược đồ này giúp ta hiểu một cách chi tiết về lĩnh vực nghiệp vụ sẽ được hỗ trợ hay chịu tác động của ứng dụng mới. Các lược đồ hoạt động giúp ta hình dung hệ thống mới được đề nghị rõ ràng hơn đồng thời xác định các use case của hệ thống đó.

## Sử dụng các luồng đối tượng

- Trong ngữ cảnh này, các luồng đối tượng được sử dụng để cho thấy cách thức các thực thể được tạo ra và sử dụng trong một luồng công việc. Các luồng đối tượng mô tả các đầu vào và đầu ra từ các trạng thái hoạt động trong một biểu đồ hoạt động. Có hai thành phần ký hiệu sau:
  - Trạng thái luồng đối tượng (object flow state): biểu diễn một đối tượng của một lớp tham gia vào luồng công việc được biểu diễn trong biểu đồ hoạt động. Đối tượng này có thể là đầu ra của một hoạt động và là đầu vào của nhiều hoạt động khác.
  - Luồng đối tượng (object flow) là một kiểu luồng điều khiển với một trạng thái luồng đối tượng làm đầu vào/đầu ra.
- Ký hiệu luồng đối tượng nói lên sự tồn tại của một đối tượng trong một trạng thái cụ thể, chứ không là chính đối tượng đó. Cùng một đối tượng này có thể được thao tác bởi một số các hoạt động kế tiếp nhau làm thay đổi trạng thái của đối tượng. Sau đó, nó có thể được hiển thị nhiều lần trong một biểu đồ hoạt động,

mỗi lần xuất hiện sẽ biểu diễn một trạng thái khác nhau trong đời sống của nó. Trạng thái của đối tượng tại mỗi thời điểm có thể được đặt trong ngoặc và viết thêm vào tên của lớp.

- Một trạng thái luồng đối tượng có thể xuất hiện như là trạng thái kết thúc của một luồng đối tượng (sự chuyển tiếp) và là trạng thái bắt đầu của nhiều luồng đối tượng (những sự chuyển tiếp).
- Các luồng đối tượng có thể được so sánh với các luồng dữ liệu bên trong luồng công việc của một use case. Không giống như các luồng dữ liệu truyền thống, các luồng đối tượng tồn tại ở một thời điểm xác định trong một biểu đồ hoạt động.

Sử dụng các lược đồ hợp tác (collaboration) và tuần tự (sequence)

Đối với mỗi hiện thực hóa use-case, có thể có một hoặc nhiều lược đồ tương tác để mô tả các thừa tác viên và thực thể tham gia, cùng với những tương tác của chúng. Có hai loại lược đồ tương tác là: lược đồ tuần tự và lược đồ hợp tác. Chúng diễn tả những thông tin tương tự nhau, nhưng trình bày những thông tin này theo những cách khác nhau:

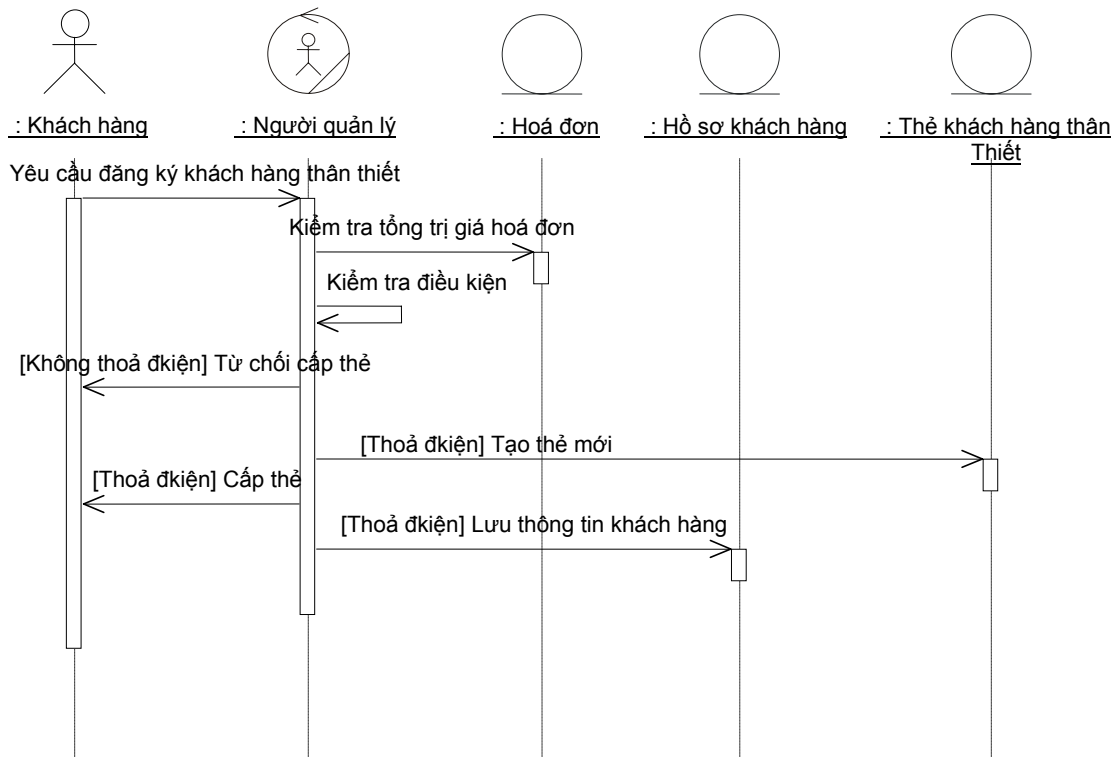
- Lược đồ tuần tự mô tả rõ ràng trình tự các sự kiện. Với các kịch bản phức tạp, thì lược đồ tuần tự thích hợp hơn so với các lược đồ hoạt động.
- Lược đồ hợp tác trình bày các mối liên kết giao tiếp và những thông điệp giữa các đối tượng. Chúng phù hợp hơn trong việc giúp ta hiểu được tất cả các tác động trên một đối tượng cho trước.
- Nếu có ít luồng rẽ nhánh, nhưng có nhiều thực thể tham gia, thì lược đồ tương tác thường là một sự lựa chọn tốt hơn so với lược đồ hoạt động nhằm để trình bày hiện thực hóa của luồng công việc.

Lược đồ tuần tự mô tả một mẫu tương tác giữa các đối tượng, được sắp xếp theo thứ tự thời gian; nó cho thấy các đối tượng tham gia vào sự tương tác theo những "lifeline" và những thông điệp mà chúng gửi cho nhau.

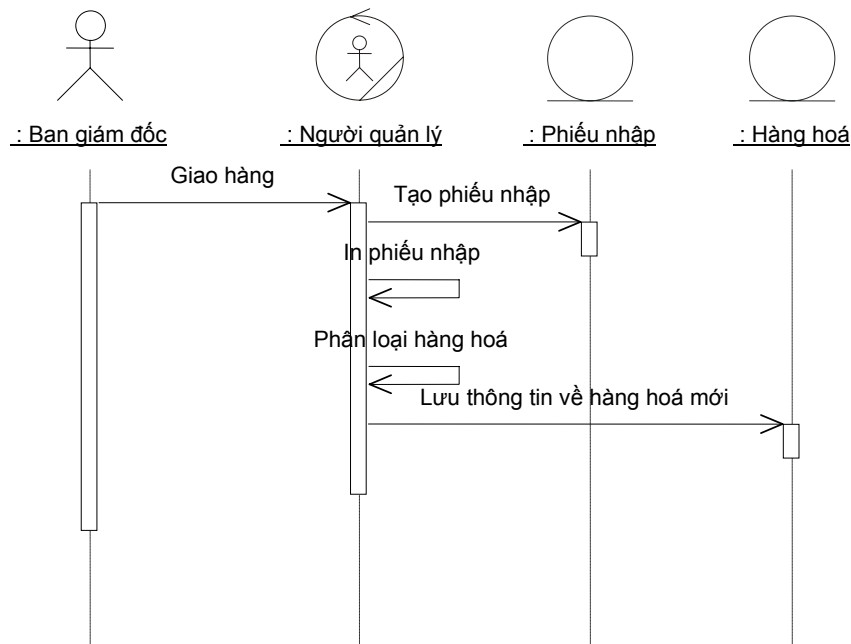
Về mặt đồ họa, một lược đồ tuần tự mô tả chi tiết sự tương tác giữa các thừa tác viên, tác nhân, và cách thức các thực thể được truy xuất khi một use case được thực thi. Một lược đồ tuần tự mô tả vắn tắt các thừa tác viên tham gia làm những gì, và cách thức các thực thể được thao tác thông qua những sự kích hoạt, và cách thức chúng giao tiếp bằng cách gửi thông điệp cho nhau.

Ví dụ: Sơ đồ tuần tự của use case Quản lý khách hàng thân thiết





Sơ đồ tuần tự của use case Quản lý nhập hàng

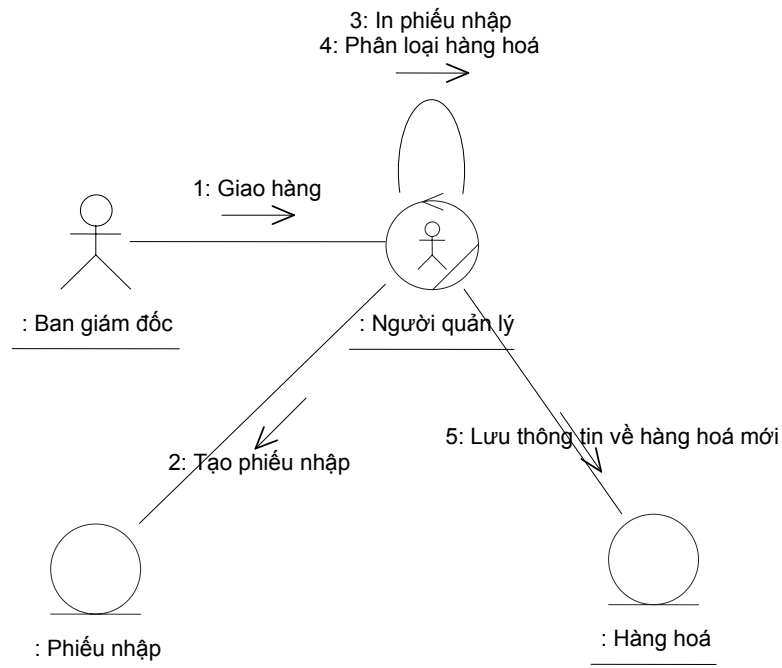
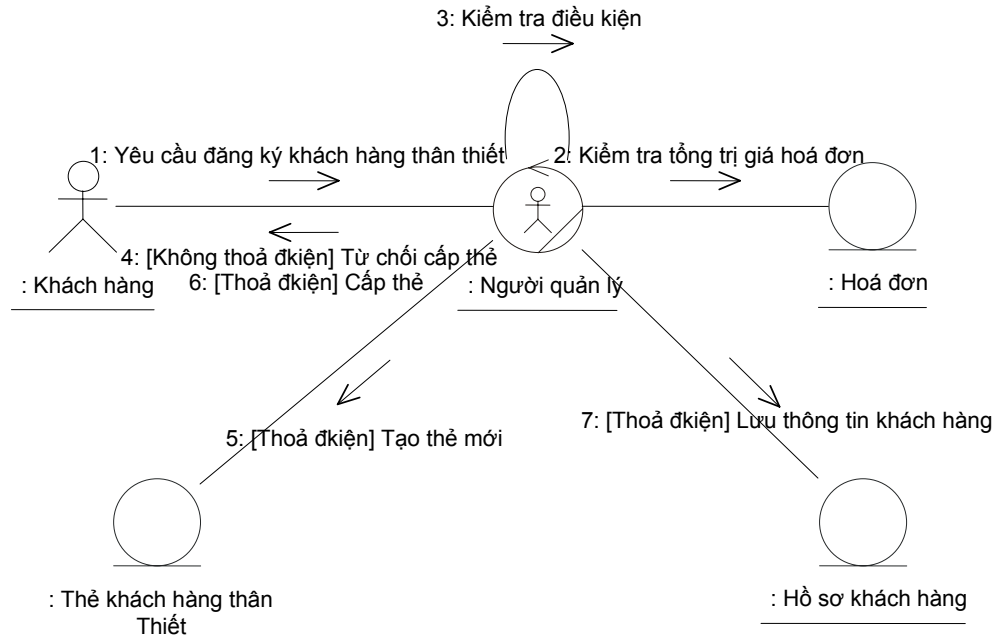


Những thông tin được tìm thấy trong một lược đồ tuần tự cũng có thể được biểu diễn trong một lược đồ hợp tác.

Một lược đồ hợp tác mô tả một mẫu tương tác giữa các đối tượng; nó cho thấy các đối tượng tham gia vào sự tương tác thông qua những mối liên kết giữa chúng và những thông điệp mà chúng gửi cho nhau.

Một lược đồ hợp tác về mặt ngữ nghĩa cũng tương tự như một lược đồ tuần tự, nhưng tập trung chủ yếu vào các đối tượng, trong khi lược đồ tuần tự tập trung vào các tương tác. Một lược đồ tuần tự trình bày một tập con các đối tượng có liên quan đến chuỗi công việc bị ảnh hưởng, bao gồm các mối liên kết giữa chúng, các thông điệp và các chuỗi thông điệp.

Ví dụ: lược đồ hợp tác tương ứng

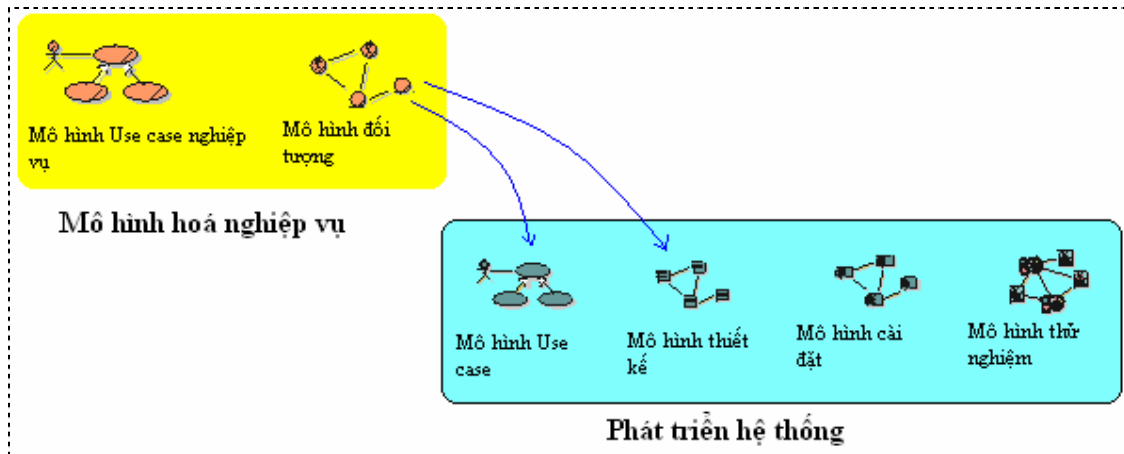


## Xác định yêu cầu tự động hoá

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.

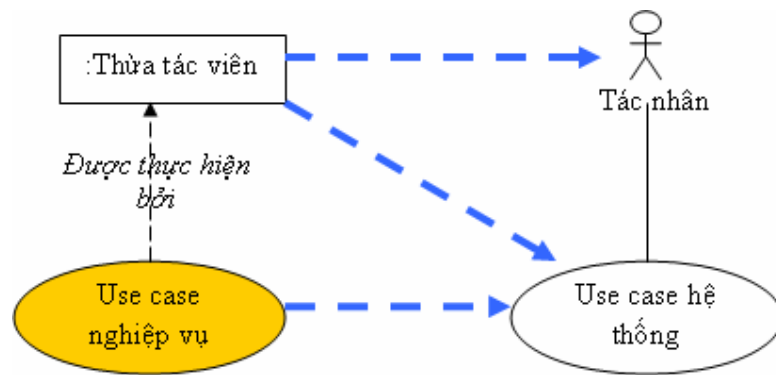
## Xác định tác nhân và use case hệ thống phần mềm



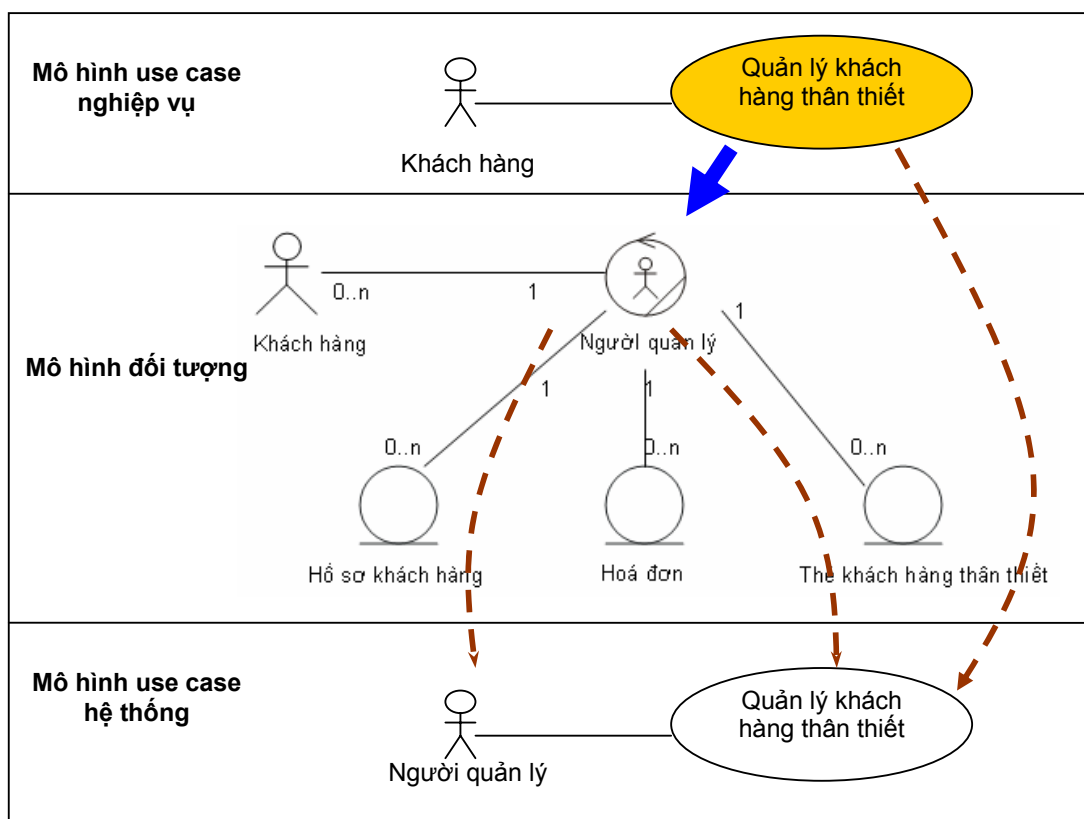
Để xây dựng các hệ thống, cần phải hiểu rõ các qui trình nghiệp vụ. Thậm chí sẽ hữu ích hơn nếu sử dụng các vai trò và trách nhiệm của nhân viên, cũng như những gì được xử lý bởi nghiệp vụ làm nền tảng để xây dựng hệ thống. Điều này được xác định từ góc nhìn bên trong nghiệp vụ dựa vào mô hình đối tượng, trong đó có thể thấy được mối liên kết chặt chẽ nhất đến hình thức thể hiện các mô hình của hệ thống.

Để xác định các use case trong hệ thống thông tin, hãy bắt đầu từ các thừa tác viên trong mô hình đối tượng nghiệp vụ. Đối với mỗi thừa tác viên, thực hiện những bước sau đây:

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



Ví dụ: Xác định tác nhân và use case hệ thống phần mềm cho use case nghiệp vụ Quản lý khách hàng thân thiết



Trong trường hợp này, khách hàng giao tiếp với người quản lý để giải quyết các yêu cầu về khách hàng thân thiết. Người quản lý sẽ sử dụng phần mềm như là một công cụ nhằm trợ giúp trong các xử lý đáp ứng cho yêu cầu của khách hàng. Do đó, người quản lý lúc này sẽ là tác nhân của hệ thống phần mềm và các chức năng được tự động hoá trong use case nghiệp vụ Quản lý khách hàng thân thiết sẽ trở thành use case mô tả chức năng phần mềm hệ thống.

Nếu mục đích của việc xây dựng một hệ thống là tự động hóa hoàn toàn các qui trình nghiệp vụ (chẳng hạn như việc xây dựng một ứng dụng thương mại điện tử) thì thừa tác viên sẽ không trở thành tác nhân hệ thống nữa. Thay vào đó, chính tác nhân của môi trường nghiệp vụ giao tiếp trực tiếp với hệ thống và hoạt động như một tác nhân hệ thống.

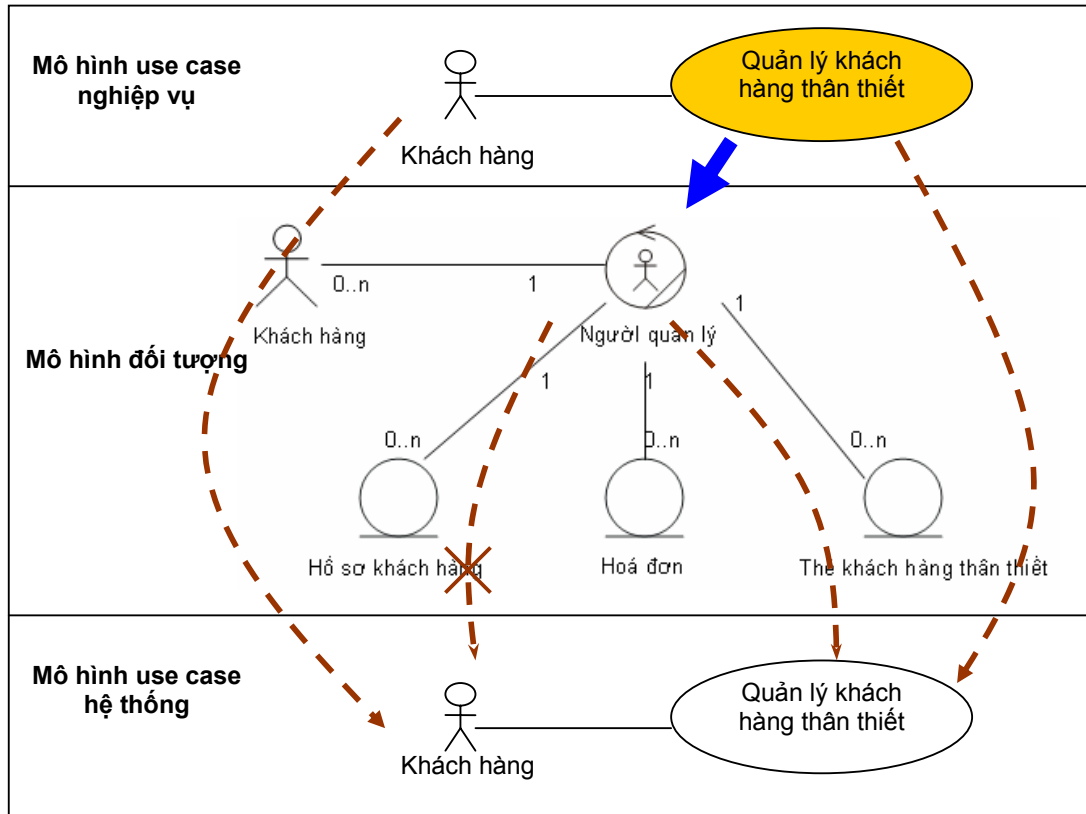
Khi đó, cách thức thực hiện nghiệp vụ sẽ bị thay đổi khi xây dựng một ứng dụng thuộc loại này. Các trách nhiệm của thừa tác viên sẽ chuyển sang tác nhân nghiệp vụ.

Ví dụ: giả sử xây dựng một site thương mại điện tử cho một việc quản lý khách hàng thân thiết, ta sẽ thay đổi cách thức mà qui trình được hiện thực hóa. Lúc này, Khách hàng sử dụng trực tiếp hệ thống thông qua việc truy cập ứng dụng web.

Các trách nhiệm của thừa tác viên Người quản lý sẽ chuyển sang tác nhân nghiệp vụ Khách hàng.

Tạo ra tác nhân hệ thống Khách hàng tương ứng với tác nhân nghiệp vụ Khách hàng.

Loại bỏ đi Người quản lý.



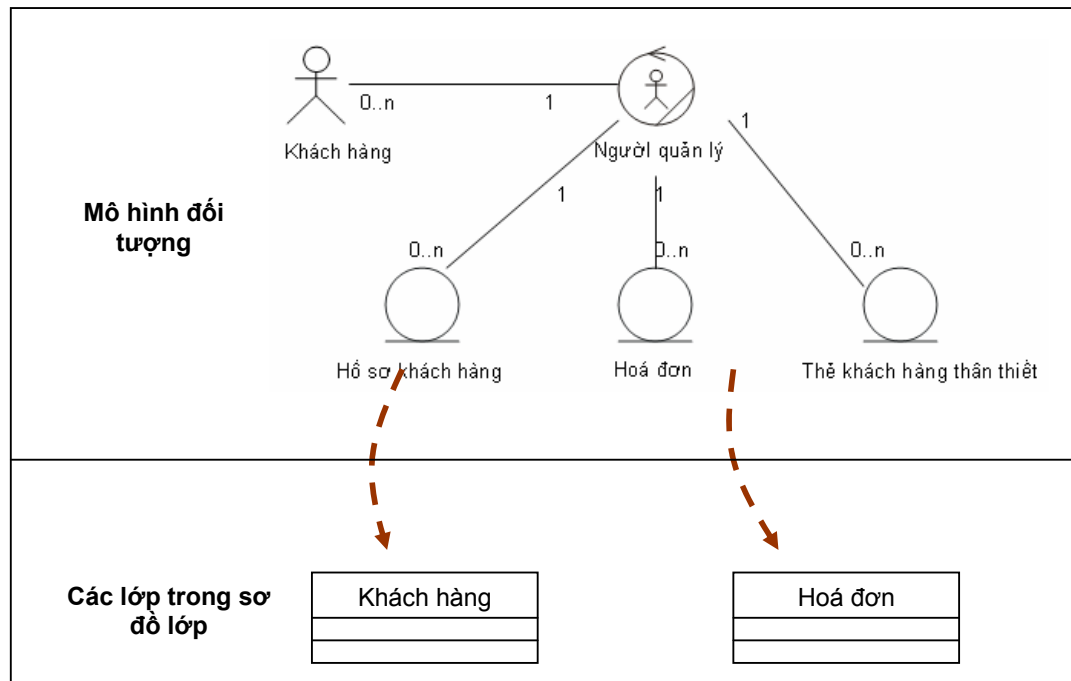
### Xác định các lớp đối tượng thông tin trong hệ thống phần mềm

Một thực thể nghiệp vụ được quản lý bởi hệ thống thông tin sẽ tương ứng với một thực thể trong mô hình phân tích của hệ thống thông tin. Tuy nhiên, trong một số trường hợp, sẽ thích hợp nếu để các thuộc tính của thực thể nghiệp vụ tương ứng với các thực thể trong mô hình hệ thống thông tin.

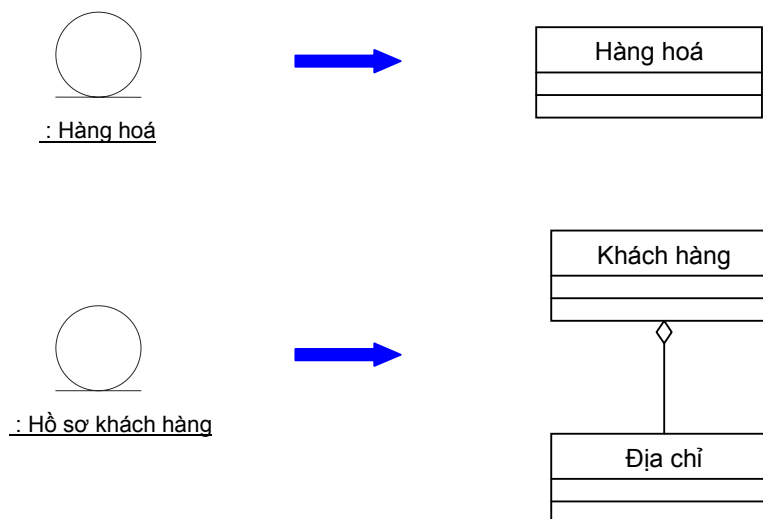
Nhiều thừa tác viên có thể truy xuất một thực thể nghiệp vụ. Do đó, các thực thể tương ứng trong hệ thống có thể tham gia vào một số use case hệ thống thông tin.



Ví dụ: các lớp đối tượng hệ thống phần mềm cho từ sơ đồ lớp của use case nghiệp vụ “Quản lý khách hàng thân thiết”



Việc xác định các lớp đối tượng phần mềm chính là việc xác định cấu trúc thông tin của đối tượng nghiệp vụ để quản lý trong hệ thống thông tin. Đây cũng có thể xem như công đoạn chi tiết hoá về mặt cấu trúc nhằm đạt được một cách nhìn luận lý về cấu trúc của một đối tượng nghiệp vụ theo góc độ tin học hoá cho đối tượng đó. Tùy theo độ phức tạp của cấu trúc thực thể mà chúng ta có những cách chuyển đổi đơn giản (một thực thể nghiệp vụ → một lớp) hoặc phức tạp (một thực thể nghiệp vụ → nhiều lớp). Cuối cùng, sau khi đã xác định được các lớp thì để đạt được một mô hình đầy đủ, chúng ta còn phải xác định các mối kết hợp giữa các lớp. Chi tiết của phần này được đề cập ở chương tiếp theo.



## Bài tập

Mô hình hoá việc hiện thực hoá các use case nghiệp vụ đã được tìm ra trong hệ thống “Quản lý thuê văn phòng cao ốc” bằng cách:

- Đặc tả nội dung use case dùng mô hình hoạt động (không sử dụng swimlane)
- Xác định các thừa tác viên và các thực thể
- Xây dựng sơ đồ lớp cho use case nghiệp vụ
- Thiết kế chi tiết use case dùng:
  - o Sơ đồ hoạt động (dùng swimlane để mô tả vai trò của thừa tác viên và các thực thể được tác động bởi các hoạt động)
  - o Xây dựng các sơ đồ tương tác (sơ đồ tuần tự, sơ đồ hợp tác, sơ đồ trạng thái)
- Đề xuất một sơ đồ use case mô tả chức năng hệ thống phần mềm giúp cho việc tự động hoá hệ thống nghiệp vụ trên.