

Biểu đồ Use Case

Bởi:

duongkieuhoa

tonthathoan

Biểu đồ Use Case

Use Case được mô tả trong ngôn ngữ UML qua biểu đồ Use Case (Use Case Diagram), và một mô hình Use Case có thể được chia thành một số lượng lớn các biểu đồ như thế. Một biểu đồ Use Case chứa các phần tử mô hình biểu thị hệ thống, tác nhân cũng như Use Case và chỉ ra các mối quan hệ giữa các Use Case.

Lời mô tả nội dung Use Case thường được cung cấp dưới dạng văn bản. Trong UML, lời mô tả đó được coi là thuộc tính "văn bản" (document) của Use Case. Lời mô tả này bao chứa những thông tin quan trọng, định nghĩa các yêu cầu và chức năng cụ thể. Thay cho việc mô tả Use Case bằng văn bản, bạn cũng có thể vẽ một biểu đồ hoạt động (activity diagram). Mặc dầu vậy, nên nhớ rằng một Use Case cần phải được mô tả sao cho dễ hiểu và dễ giao tiếp đối với người sử dụng, mà những cấu trúc phức tạp như một biểu đồ hoạt động có thể gây cảm giác xa lạ đối với những người không quen sử dụng.

Tóm tắt: Một biểu đồ Use Case thể hiện:

- Hệ thống
- Tác nhân
- Use Case.

Ví dụ biểu đồ Use Case trong UML:



Trong đó :

- Hệ thống được thể hiện qua hình chữ nhật với tên hệ thống ở bên trên
- Tác nhân được thể hiện qua kí hiệu hình nhân
- Use Case được thể hiện qua hình ellipse

Hệ thống:

Vì hệ thống là một thành phần của mô hình Use Case nên ranh giới của hệ thống mà ta muốn phát triển cần phải được định nghĩa rõ ràng. Xin nhớ rằng một hệ thống không phải bao giờ cũng nhất thiết là một hệ thống phần mềm; nó có thể là một chiếc máy, hoặc là một doanh nghiệp. Định nghĩa các ranh giới và trách nhiệm của hệ thống không phải bao giờ cũng là việc dễ dàng, bởi không phải bao giờ người ta cũng rõ ràng nhìn ra tác vụ nào có khả năng được tự động hóa tốt nhất ở hệ thống này và tác vụ nào thì tốt nhất nên thực hiện thủ công hoặc dành cho các hệ thống khác. Một khía cạnh khác cần chú ý là hệ thống cần phải lớn tới mức độ nào trong phiên bản đầu tiên của nó. Cố gắng tối đa cho phiên bản đầu tiên của hệ thống thường là cách mà người ta hay thực hiện, thế nhưng những mục tiêu quá tầm như vậy có thể khiến cho hệ thống trở nên quá lớn và thời gian để cung cấp hệ thống quá lâu. Một sáng kiến tốt hơn là xác nhận cho rõ các chức năng căn bản và tập trung vào việc định nghĩa một kiến trúc hệ thống thích hợp, rõ ràng, có nền tảng rộng mở để nhiều chức năng hơn có thể được bổ sung vào hệ thống này trong các phiên bản sau.

Yếu tố quan trọng là bạn phải tạo dựng được một bản catalog của các khái niệm (các thực thể) trung tâm cùng với các thuật ngữ và định nghĩa thích hợp trong những giai đoạn đầu của thời kỳ phân tích. Đây chưa phải mô hình phạm vi đối tượng, mà đúng hơn là một cố gắng để mô tả các thuật ngữ của hệ thống hoặc doanh nghiệp mà chúng ta cần mô hình hóa. Các thuật ngữ sau đó sẽ được dùng để mô tả Use Case. Phương thức cụ thể của catalog này có thể rất khác nhau; nó có thể là một mô hình khái niệm chỉ ra các mối quan hệ đơn giản hoặc chỉ là một văn bản chứa các thuật ngữ cùng lời mô tả vắn tắt những thuật ngữ này trong thế giới thực.

Tác nhân:

Một tác nhân là một người hoặc một vật nào đó tương tác với hệ thống, sử dụng hệ thống. Trong khái niệm "tương tác với hệ thống", ý chúng ta muốn nói rằng tác nhân sẽ gửi thông điệp đến hệ thống hoặc là nhận thông điệp xuất phát từ hệ thống, hoặc là thay đổi các thông tin cùng với hệ thống. Nói một cách ngắn gọn, tác nhân thực hiện các Use Case. Thêm một điều nữa, một tác nhân có thể là người mà cũng có thể là một hệ thống

khác (ví dụ như là một chiếc máy tính khác được nối kết với hệ thống của chúng ta hoặc một loại trang thiết bị phần cứng nào đó tương tác với hệ thống).

Một tác nhân là một dạng thực thể (một lớp), chứ không phải một thực thể. Tác nhân mô tả và đại diện cho một vai trò, chứ không phải là một người sử dụng thật sự và cụ thể của hệ thống. Nếu một anh chàng John nào đó muốn mua hợp đồng bảo hiểm từ một hãng bảo hiểm, thì vai trò của anh ta sẽ là người mua hợp đồng bảo hiểm, và đây mới là thứ mà chúng ta muốn mô hình hóa, chứ không phải bản thân anh chàng John. Trong sự thực, một con người cụ thể có thể đóng vai trò làm nhiều tác nhân trong một hệ thống: một nhân viên ngân hàng đồng thời cũng có thể là khách hàng của chính ngân hàng đó. Mặt khác, số lượng các vai trò mà một con người cụ thể được phép đảm trách trong một hệ thống cũng có thể bị hạn chế, ví dụ cùng một người không được phép vừa soạn hóa đơn vừa phê duyệt hóa đơn đó. Một tác nhân sẽ có một tên, và cái tên này cần phải phản ánh lại vai trò của tác nhân. Cái tên đó không được phản ánh lại một thực thể riêng biệt của một tác nhân, mà cũng không phản ánh chức năng của tác nhân đó.

Một tác nhân giao tiếp với hệ thống bằng cách gửi hoặc là nhận thông điệp, giống như khái niệm chúng ta đã quen biết trong lập trình hướng đối tượng. Một Use Case bao giờ cũng được kích hoạt bởi một tác nhân gửi thông điệp đến cho nó. Khi một Use Case được thực hiện, Use Case có thể gửi thông điệp đến một hay là nhiều tác nhân. Những thông điệp này cũng có thể đến với các tác nhân khác, bên cạnh chính tác nhân đã kích hoạt và gây ra Use Case.

Tác nhân cũng có thể được xếp loại. Một tác nhân chính (Primary Actor) là tác nhân sử dụng những chức năng căn bản của hệ thống, tức là các chức năng chính. Ví dụ, trong một hệ thống bảo hiểm, một tác nhân căn bản có thể là tác nhân xử lý việc ghi danh và quản lý các hợp đồng bảo hiểm. Một tác nhân phụ (secondary actor) là tác nhân sử dụng các chức năng phụ của hệ thống, ví dụ như các chức năng bảo trì hệ thống như quản trị ngân hàng dữ liệu, giao tiếp, back-up và các tác vụ quản trị khác. Một ví dụ cho tác nhân phụ có thể là nhà quản trị hoặc là một nhân viên sử dụng chức năng trong hệ thống để rút ra các thông tin thống kê về doanh nghiệp. Cả hai loại tác nhân này đều được mô hình hóa để đảm bảo mô tả đầy đủ các chức năng của hệ thống, mặc dù các chức năng chính mới thật sự nằm trong mối quan tâm chủ yếu của khách hàng.

Tác nhân còn có thể được định nghĩa theo dạng tác nhân chủ động (active actor) hay tác nhân thụ động (passive actor). Một tác nhân chủ động là tác nhân gây ra Use Case, trong khi tác nhân thụ động không bao giờ gây ra Use Case mà chỉ tham gia vào một hoặc là nhiều Use Case.

Tìm tác nhân:

Khi nhận diện tác nhân, có nghĩa là chúng ta lọc ra các thực thể đáng quan tâm theo khía cạnh sử dụng và tương tác với hệ thống. Sau đó chúng ta có thể thử đặt mình vào vị trí

Biểu đồ Use Case

của tác nhân để cố gắng nhận ra các yêu cầu và đòi hỏi của tác nhân đối với hệ thống và xác định tác nhân cần những Use Case nào. Có thể nhận diện ra các tác nhân qua việc trả lời một số các câu hỏi như sau:

- Ai sẽ sử dụng những chức năng chính của hệ thống (tác nhân chính)?
- Ai sẽ cần sự hỗ trợ của hệ thống để thực hiện những tác vụ hàng ngày của họ?
- Ai sẽ cần bảo trì, quản trị và đảm bảo cho hệ thống hoạt động (tác nhân phụ)?
- Hệ thống sẽ phải xử lý và làm việc với những trang thiết bị phần cứng nào?
- Hệ thống cần phải tương tác với các hệ thống khác nào? Nhóm các hệ thống này được chia ra làm hai nhóm, nhóm kích hoạt cho mối quan hệ với hệ thống, và nhóm mà hệ thống cần phải xây dựng của chúng ta sẽ thiết lập quan hệ. Khái niệm hệ thống bao gồm cả các hệ thống máy tính khác cũng như các ứng dụng khác trong chính chiếc máy tính mà hệ thống này sẽ hoạt động.
- Ai hay cái gì quan tâm đến kết quả (giá trị) mà hệ thống sẽ sản sinh ra?

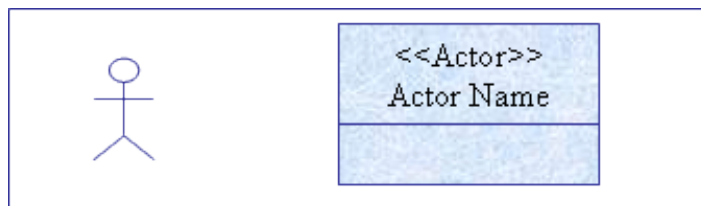
Khi đi tìm những người sử dụng hệ thống, đừng quan sát những người đang ngồi ở trước màn hình máy tính. Nên nhớ rằng, người sử dụng có thể là bất kỳ người nào hay bất kỳ vật nào tương tác hoặc trực tiếp hoặc gián tiếp với hệ thống và sử dụng các dịch vụ của hệ thống này để đạt đến một kết quả nào đó. Đừng quên rằng mô hình hóa Use Case được thực hiện để mô hình hóa một doanh nghiệp, vì thế tác nhân thường thường là khách hàng của doanh nghiệp đó. Từ đó suy ra họ không phải là người sử dụng theo nghĩa đơn giản và trực tiếp là người ngồi trước màn hình máy tính và thao tác với máy tính.

Để có thể nhận dạng được tốt nhiều tác nhân khác nhau, hãy tiến hành nghiên cứu những người sử dụng của hệ thống hiện thời (một hệ thống thủ công hoặc một hệ thống đang tồn tại), hỏi xem họ đóng những vai trò nào khi thực thi công việc hàng ngày của họ với hệ thống. Cũng người sử dụng đó có thể thực thi nhiều vai trò khác nhau tại nhiều thời điểm khác nhau, tùy thuộc vào việc chức năng nào trong hệ thống đang được sử dụng.

Xin nhắc lại, một tác nhân là một vai trò (một lớp), chứ không phải một thực thể riêng lẻ. Mặc dù vậy, khi cung cấp ví dụ là một vài các thực thể của một tác nhân, bạn có thể đảm bảo rằng tác nhân đó thật sự tồn tại. Một tác nhân phải có một sự liên kết (Association) nào đó với một hoặc là nhiều Use Case. Mặc dù có những tác nhân có thể không kích hoạt nên một Use Case nào, nhưng tác nhân đó sẽ giao tiếp ít nhất với một Use Case tại một thời điểm nào đó. Cần phải đặt tên cho tác nhân làm sao để tên phản ánh đúng vai trò của tác nhân đó trong hệ thống.

Biểu diễn tác nhân trong ngôn ngữ UML:

Tác nhân trong UML là một lớp với biệt ngữ "Actor" (Tác nhân) và tên của lớp này là tên của tác nhân (phản ánh vai trò của tác nhân). Một lớp tác nhân có thể vừa có thuộc tính (attribute) lẫn hành vi (method) cũng như một thuộc tính tài liệu (document) mô tả tác nhân đó. Một lớp tác nhân có một biểu tượng chuẩn hóa, biểu tượng "hình nhân":



biểu tượng tác nhân trong UML

Use Case:

Một Use Case là đại diện cho một chức năng nguyên vẹn mà một tác nhân nhận được. Một Use Case trong ngôn ngữ UML được định nghĩa là một tập hợp của các chuỗi hành động mà một hệ thống thực hiện để tạo ra một kết quả có thể quan sát được, tức là một giá trị đến với một tác nhân cụ thể. Những hành động này có thể bao gồm việc giao tiếp với một loạt các tác nhân cũng như thực hiện tính toán và công việc nội bộ bên trong hệ thống.

Các tính chất tiêu biểu của một Use Case là:

- Một Use Case bao giờ cũng được gây ra bởi một tác nhân, được thực hiện nhân danh một tác nhân nào đó. Tác nhân phải ra lệnh cho hệ thống để thực hiện Use Case đó, dù là trực tiếp hay gián tiếp. Hiếm khi có tác nhân không liên quan đến việc gây ra một Use Case nào đó.
- Một Use Case phải cung cấp một giá trị cho một tác nhân. Giá trị đó không phải bao giờ cũng cần thiết phải nổi trội ra ngoài, nhưng luôn phải được thấy rõ.
- Một Use Case là phải hoàn tất. Một trong những lỗi thường gặp là sẽ chia một Use Case thành các Use Case nhỏ hơn, và các Use Case này thực thi lẫn nhau giống như việc gọi hàm cho một ngôn ngữ lập trình. Một Use Case sẽ không được coi là hoàn tất chừng nào mà giá trị cuối cùng của nó chưa được sản sinh ra, thậm chí ngay cả khi đã xảy ra nhiều động tác giao tiếp (ví dụ như đối thoại với người sử dụng).

Use Case được nối với tác nhân qua liên kết (association). Đường liên kết chỉ ra những tác nhân nào giao tiếp với Use Case nào. Mỗi liên kết bình thường ra là một mối quan hệ 1-1 và không có hướng. Điều đó muốn nói lên rằng một thực thể của lớp tác nhân sẽ giao tiếp với một thực thể của một Use Case và cả hai có thể giao tiếp với nhau trong cả hai chiều. Một Use Case sẽ được đặt tên theo một thực thể mà Use Case sẽ thực hiện, ví

Biểu đồ Use Case

dụ như ký hợp đồng bảo hiểm, cập nhật danh sách, v.v..., và thường là một cụm từ hơn là chỉ một từ riêng lẻ.

Một Use Case là một lớp, chứ không phải một thực thể. Nó mô tả trọn vẹn một chức năng, kể cả các giải pháp bổ sung và thay thế có thể có, các lỗi có thể xảy ra cũng như những ngoại lệ có thể xảy ra trong quá trình thực thi. Một kết quả của sự thực thể hóa một Use Case được gọi là một cảnh kịch (scenario) và nó đại diện cho một sự sử dụng cụ thể của hệ thống (một đường dẫn thực thi riêng biệt qua hệ thống). Ví dụ một cảnh kịch của Use Case "Ký hợp đồng bảo hiểm" có thể là "John liên hệ với hệ thống qua điện thoại rồi sau đó ký hợp đồng bảo hiểm ô tô cho chiếc xe Toyota Carolla mà anh ta vừa mua."

Tìm Use Case:

Quá trình tìm các Use Case bắt đầu với các tác nhân đã được xác định ở phần trước. Đối với mỗi tác nhân, hãy hỏi các câu hỏi sau:

a. Tác nhân này cần những chức năng nào từ hệ thống? Hành động chính của tác nhân là gì ?

Ví dụ cho một giao dịch rút tiền bên máy ATM trong một nhà băng lẻ, các hành động chính của khách hàng (tác nhân) có thể là :

- Đút thẻ vào máy ATM
- Nhập password
- Nhập loại chuyển dịch
- Nhập số tiền mặt muốn rút ra
- Yêu cầu về loại tiền
- Nhặt tiền ra từ máy
- Rút thẻ và tờ in kết quả giao dịch

b. Tác nhân có cần phải đọc, phải tạo, phải hủy bỏ, phải sửa chữa, hay là lưu trữ một loại thông tin nào đó trong hệ thống?

Ví dụ :

- Nhân viên nhà băng liệu có quyền truy xuất hay thay đổi mức tiền lãi?

Biểu đồ Use Case

- Khách hàng có thể thay đổi password của mình.

c. Tác nhân có cần phải báo cho hệ thống biết về những sự kiện nào đó? Những sự kiện như thế sẽ đại diện cho những chức năng nào?

Ví dụ :

- Khách hàng kết thúc tài khoản, nhân viên cung cấp những thông tin này cho hệ thống.

- Có một chương trình đầu tư mới, các chi tiết của chương trình này sẽ phải được nhân viên nhà băng nhập vào hệ thống.

d. Hệ thống có cần phải thông báo cho Actor về những thay đổi bất ngờ trong nội bộ hệ thống?

- Trong tài khoản còn quá ít tiền.

- Ba kỳ liên tiếp tiền lương chưa đổ về tài khoản.

e. Công việc hàng ngày của tác nhân có thể được đơn giản hóa hoặc hữu hiệu hóa qua các chức năng mới trong hệ thống (thường đây là những chức năng tiêu biểu chưa được tự động hóa trong hệ thống)?

f. Các câu hỏi khác:

- Use Case có thể được gây ra bởi các sự kiện nào khác?

Ví dụ :

Sự kiện thời gian: Cuối tháng, hết hạn đầu tư.

Sự kiện bình thường của hệ thống: Tự động chuyển tiền theo các lệnh xác định trước.

Các sự kiện bất bình thường: Hợp đồng đầu tư kết thúc trước thời hạn.

- Hệ thống cần những thông tin đầu vào/đầu ra nào? Những thông tin đầu vào/đầu ra đó từ đâu tới và sẽ đi đâu?

- Khó khăn và thiếu hụt chính trong hệ thống hiện thời nằm ở đâu (thủ công /tự động hóa)?

Đối với nhóm câu hỏi cuối không có nghĩa là Use Case ở đây không có tác nhân, mà tác nhân sẽ được nhận ra chỉ khi chúng ta nhận diện ra các Use Case này và sau đó xác định

Biểu đồ Use Case

tác nhân dựa trên cơ sở là Use Case. Xin nhắc lại, một Use Case bao giờ cũng phải được liên kết với ít nhất một tác nhân.

Ví dụ tìm Use Case:

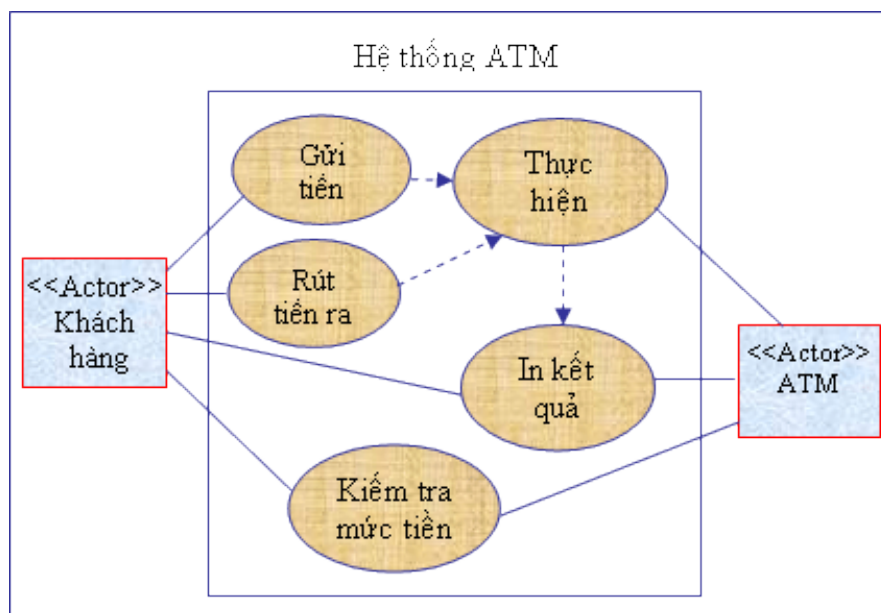
Nhà băng ABC đưa ra các yêu cầu sau:

- Một khách hàng có thể muốn gửi tiền vào, rút tiền ra hoặc đơn giản kiểm tra lại số tiền trong tài khoản của anh ta qua máy tự động rút tiền (ATM). Khi đưa tiền vào hoặc rút tiền ra, cần phải ghi ra giấy kết quả những chuyển dịch đã thực hiện và trao tờ giấy này cho khách hàng.

Quan sát các chức năng căn bản và các thành phần tham gia, ta thấy có hai tác nhân dễ nhận ra nhất là khách hàng và nhân viên thu ngân.

Qua đó, có thể nhận dạng các Use Case sau:

- Gửi tiền vào.
- Rút tiền ra.
- Kiểm tra mức tiền trong tài khoản
- Thực hiện các chuyển dịch nội bộ hệ thống
- In kết quả các chuyển dịch đã thực hiện.



Các Use case trong hệ thống ATM

Biểu đồ Use Case

Use Case gửi tiền vào và rút tiền ra phụ thuộc vào Use Case thực hiện các chuyển dịch trong nội bộ hệ thống, việc thực hiện này về phần nó lại phụ thuộc vào chức năng in ra các công việc đã được thực hiện. Kiểm tra mức tiền trong tài khoản là một Use Case độc lập, không phụ thuộc vào các Use Case khác.