



Miêu tả Use Case

Bởi:

duongkieuhoa

tonthathoan

Miêu tả Use Case

Như đã trình bày, lời miêu tả một Use Case thường được thực hiện trong văn bản. Đây là lời đặc tả đơn giản và nhất quán về việc các tác nhân và các Use Case (hệ thống) tương tác với nhau ra sao. Nó tập trung vào ứng xử đối ngoại của hệ thống và không đề cập tới việc thực hiện nội bộ bên trong hệ thống. Ngôn ngữ và các thuật ngữ được sử dụng trong lời miêu tả chính là ngôn ngữ và các thuật ngữ được sử dụng bởi khách hàng/người dùng.

Văn bản miêu tả cần phải bao gồm những điểm sau:

- Mục đích của Use Case: Mục đích chung cuộc của Use Case là gì? Cái gì cần phải được đạt tới? Use Case nói chung đều mang tính hướng mục đích và mục đích của mỗi Use Case cần phải rõ ràng.

- Use Case được khởi chạy như thế nào: Tác nhân nào gây ra sự thực hiện Use Case này? Trong hoàn cảnh nào?

- Chuỗi các thông điệp giữa tác nhân và Use Case: Use Case và các tác nhân trao đổi thông điệp hay sự kiện nào để thông báo lẫn cho nhau, cập nhật hoặc nhận thông tin và giúp đỡ nhau quyết định? Yếu tố nào sẽ miêu tả dòng chảy chính của các thông điệp giữa hệ thống và tác nhân, và những thực thể nào trong hệ thống được sử dụng hoặc là bị thay đổi?

- Dòng chảy thay thế trong một Use Case: Một Use Case có thể có những dòng thực thi thay thế tùy thuộc vào điều kiện. Hãy nhắc đến các yếu tố này, nhưng chú ý đừng miêu tả chúng quá chi tiết đến mức độ chúng có thể “che khuất” dòng chảy chính của các hoạt động trong trường hợp căn bản. Những động tác xử lý lỗi đặc biệt sẽ được miêu tả thành các Use Case khác.

- Use Case sẽ kết thúc với một giá trị đối với tác nhân như thế nào: Hãy miêu tả khi nào Use Case được coi là đã kết thúc, và loại giá trị mà nó cung cấp đến tác nhân.

Hãy nhớ rằng lời miêu tả này sẽ xác định những gì được thực thi có liên quan đến tác nhân bên ngoài, chứ không phải những sự việc được thực hiện bên trong hệ thống. Văn bản phải rõ ràng, nhất quán, khiến cho khách hàng có thể dễ dàng hiểu và thẩm tra chúng (đề rồi đồng ý rằng nó đại diện cho những gì mà anh/cô ta muốn từ phía hệ thống). Tránh dùng những câu văn phức tạp, khó diễn giải và dễ hiểu lầm.

Một Use Case cũng có thể được miêu tả qua một biểu đồ hoạt động. Biểu đồ hoạt động này chỉ ra chuỗi các hành động, thứ tự của chúng, các quyết định chọn lựa để xác định xem hành động nào sau đó sẽ được thực hiện.

Để bổ sung cho lời miêu tả một Use Case, người ta thường đưa ra một loạt các cảnh kịch cụ thể để minh họa điều gì sẽ xảy ra một khi Use Case này được thực thể hóa. Lời miêu tả cảnh kịch minh họa một trường hợp đặc biệt, khi cả tác nhân lẫn Use Case đều được coi là một thực thể cụ thể. Khách hàng có thể dễ dàng hiểu hơn toàn bộ một Use Case phức tạp nếu có những cảnh kịch được miêu tả thực tiễn hơn, minh họa lại lỗi ứng xử và phương thức hoạt động của hệ thống. Nhưng xin nhớ rằng, một lời miêu tả cảnh kịch chỉ là một sự bổ sung chứ không phải là ứng cử viên để thay thế cho lời miêu tả Use Case.

Sau khi các Use Case đã được miêu tả, một hoạt động và một công việc đặc biệt cần phải thực hiện là thẩm tra xem các mối quan hệ (đã đề cập tới trong phần 2.7) có được nhận diện không. Trước khi tất cả các Use Case được miêu tả, nhà phát triển chưa thể có được những kiến thức hoàn tất và tổng thể để xác định các mối quan hệ thích hợp, thử nghiệm làm theo phương thức đó có thể sẽ dẫn đến một tình huống nguy hiểm. Trong thời gian thực hiện công việc này, hãy trả lời các câu hỏi sau:

- Tất cả các tác nhân liên quan đến một Use Case có mối liên kết giao tiếp với Use Case đó không?
- Có tồn tại những sự tương tự giữa một loạt các tác nhân minh họa một vai trò chung và nhóm này liệu có thể được miêu tả là một lớp tác nhân căn bản (base class)?
- Có tồn tại những sự tương tự giữa một loạt các Use Case, minh họa một dòng chảy hành động chung? Nếu có, liệu điều này có thể được miêu tả là một mối quan hệ sử dụng đến với một Use Case khác?
- Có tồn tại những trường hợp đặc biệt của một Use Case có thể được miêu tả là một mối quan hệ mở rộng?
- Có tồn tại một tác nhân nào hay một Use Case nào không có mối liên kết giao tiếp? Nếu có, chắc chắn ở đây đã có chuyện làm lạc, sai trái: Tại sao lại xuất hiện tác nhân này?
- Có lời yêu cầu nào về chức năng đã được xác định, nhưng lại không được bắt kỳ một Use Case nào xử lý? Nếu thế, hãy tạo một Use Case cho yêu cầu đó.

Miêu tả Use Case

Văn bản miêu tả một Use Case đơn giản:

Ví dụ Use Case "Cung Cấp Thông Tin Về Một Tài Khoản Tại Nhà Băng ABC": Sau khi phân tích hệ thống, ta nhận thấy cần có một Use Case để in lên màn hình của nhân viên nhà băng tất cả những chi tiết về một tài khoản của một khách hàng.

Đặc tả Use Case:

Chi tiết tài khoản: // tên Use Case

Số Use Case: UCSEC35

Miêu tả ngắn: // miêu tả ngắn gọn Use Case

Use Case "chi tiết tài khoản" cho phép nhân viên nhà băng xem các chi tiết của một tài khoản mà anh ta định tìm hiểu.

Dòng chảy các sự kiện: // dòng logic chung

Nhân viên chọn Chi Tiết Tài Khoản trên menu. Một con đường khác để chỉ ra các thông tin chi tiết của tài khoản là gọi từ Màn Hình Tóm Tắt Thông Tin Về Tài Khoản (xem Use Case số UCSEC99).

Dòng hành động chính: // dòng logic chi tiết.

Mỗi khách hàng sẽ có một số định danh gọi là CustomerId. Một khách hàng có thể có nhiều tài khoản. Sau khi nhân viên nhập CustomerId vào hệ thống, màn hình phải in ra tất cả những tài khoản thuộc về khách hàng này và thuộc về nhà băng ABC, rải rác tại tất cả các chi nhánh. Khi chọn tiếp loại tài khoản và số tài khoản, các chi tiết của tài khoản mong muốn sẽ được in ra.

Loại tài khoản tiết kiệm: Nếu loại tài khoản được chọn là tài khoản tiết kiệm, thì theo Use Case số UCSEC45, các chi tiết sau đây sẽ được in ra:

- Mức tiền hiện có
- Các tờ séc chưa thanh toán
- Lượng tiền tín dụng được phép
- Lượng tiền lãi cho tới ngày hôm nay
- Lượng tiền tối thiểu cần phải có trong tài khoản

Miêu tả Use Case

Loại tài khoản đầu tư: Nếu loại tài khoản được chọn là loại đầu tư, thì theo Use Case số UCSEC46, các chi tiết sau đây sẽ được in ra.

- Hạn đầu tư
- Số tiền đầu tư
- Ngày đầu tư
- Lượng tiền cuối hạn
- Ngày cuối hạn
- Tỷ lệ lời

Dòng hành động thay thế: // chuỗi logic thay thế

Không tìm thấy chi tiết: Khi chọn một số tài khoản không thích hợp (không có tài khoản tương ứng) dù vì lý do chức năng hay kỹ thuật, theo Use Case số UCSEC12, hệ thống sẽ đưa ra một màn hình báo lỗi.

Điều kiện thoát: // Use Case kết thúc như thế nào?

Nút Thoát: Khi chọn nút thoát, người sử dụng sẽ quay trở lại màn hình chính.

Nút Xem Thêm: Khi chọn nút này, người sử dụng sẽ được yêu cầu chọn loại tài khoản từ một danh sách đổ xuống.

Nút Xem Giao Dịch: Khi chọn nút này, người sử dụng sẽ được chuyển sang màn hình "Giao dịch" và theo Use Case số UCSEC91, màn hình sẽ chỉ ra những giao dịch đã xảy ra đối với tài khoản, bên cạnh những chi tiết chính của tài khoản.

Nút Yêu Cầu In Kết Quả: Khi chọn phần thực đơn này, kết quả giao dịch theo Use Case số UCSEC70 sẽ được in ra ở một máy in địa phương nối trực tiếp với máy tính của nhân viên.

Các yêu cầu đặc biệt: // các yêu cầu đặc biệt

Theo Use Case số UCSEC110, hệ thống có khả năng in lên màn hình bằng những ngôn ngữ khác. Chức năng này sẽ được kích hoạt khi người sử dụng chọn mục Ngoại Ngữ trên menu.

Điều kiện trước đó: // điều xảy ra trước khi Use Case được thực hiện

Bảo an: Người sử dụng (nhân viên tiếp khách) được cung cấp một số định danh riêng biệt để truy nhập vào hệ thống.

Dịch chuyển: Người sử dụng chỉ đến được màn hình Chi Tiết Tài Khoản sau khi đã truy nhập thành công và Identify thành công.

Điều kiện sau đó: // điều gì xảy ra sau khi Use Case được thực hiện?

Hệ thống sẽ không lưu trữ lại bất kỳ một thông tin nào liên quan tới khách hàng lên đĩa cứng cục bộ.

Thử Use Case

Một trong các mục đích chính của Use Case là thử nghiệm (testing). Có hai loại thử nghiệm khác nhau được thực hiện ở đây: **kiểm tra** (verification) và **phê duyệt xác nhận** (validation). Kiểm tra đảm bảo là hệ thống đã được phát triển đúng đắn và phù hợp với các đặc tả đã được tạo ra. Phê duyệt xác nhận đảm bảo rằng hệ thống sẽ được phát triển chính là thứ mà khách hàng hoặc người sử dụng cuối thật sự cần đến.

Công việc **phê duyệt xác nhận** được thực hiện kể trước giai đoạn phát triển. Ngay khi một mô hình Use Case được hoàn tất (hay thậm chí có thể đang trong giai đoạn phát triển), mô hình này phải được trình bày và thảo luận với khách hàng cũng như người sử dụng. Họ cần phải xác nhận rằng mô hình này là đúng đắn, hoàn tất và thỏa mãn sự mong đợi của họ đối với hệ thống; đặc biệt là phương cách mà hệ thống cung cấp chức năng cho họ. Để làm điều đó, nhà phát triển phải đảm bảo rằng khách hàng thật sự hiểu được mô hình và ý nghĩa của chúng, để tránh trường hợp tạo ra những thứ không thể chấp nhận nổi. Trong giai đoạn này, rõ ràng là các câu hỏi và các ý tưởng sẽ xuất hiện và chúng cần phải được bổ sung thêm vào mô hình Use Case trước khi đến giai đoạn phê duyệt chung cuộc. Giai đoạn xác nhận cũng có thể được thực hiện trong thời kỳ thử nghiệm hệ thống, nhưng điểm yếu của phương thức làm này là nếu hệ thống không thỏa mãn những yêu cầu cụ thể của người sử dụng thì toàn bộ dự án rất có thể sẽ phải làm lại từ đầu.

Kiểm tra hệ thống là để đảm bảo nó hoạt động đúng như đặc tả. Điều này không thể được thực hiện trước khi đã có những thành phần của hệ thống được tạo ra. Chỉ sau đó người ta mới có thể thử xem hệ thống có hoạt động đúng như đặc tả mà người sử dụng đã đưa ra, rằng các Use Case thực hiện đúng theo như những lời đã miêu tả trong mô hình, rằng chúng hoạt động theo đúng phương thức đã được miêu tả trong văn bản miêu tả Use Case.

Đi bộ dọc Use Case.

Một trong những kỹ thuật hữu dụng được dùng trong cả giai đoạn định nghĩa lẫn thử nghiệm Use Case gọi là "Đi Bộ Dọc Use Case". Theo kỹ thuật này, nhiều người khác nhau trong nhóm làm mô hình sẽ đóng vai các tác nhân cũng như hệ thống trong một Use Case cụ thể. Người đảm nhận vai tác nhân sẽ bắt đầu bằng việc nói ra tác nhân làm gì với hệ thống. Kết quả của công việc này là hệ thống sẽ khởi chạy một Use Case cụ thể được bắt đầu từ hành động trên. Người đóng vai hệ thống sau đó sẽ nói anh ta làm gì khi Use Case được thực hiện. Nhà phát triển đứng ngoài trò chơi diễn kịch sẽ ghi chép và tìm cách phát hiện ra các điểm yếu trong các Use Case được miêu tả bằng các diễn viên. Trong trường hợp đặc thù, bạn sẽ tìm thấy rằng có một vài chuỗi hành động bổ sung không được miêu tả cũng như một vài hành động không được miêu tả với đầy đủ chi tiết.

Các "diễn viên" càng hiểu thấu đáo khía cạnh sử dụng của hệ thống bao nhiêu thì công việc thử Use Case sẽ càng hiệu quả bấy nhiêu. Việc thay đổi các diễn viên để đóng những vai trò khác nhau sẽ dẫn tới những thay đổi trong miêu tả và hướng nhìn, cung cấp dữ liệu đầu vào cho các nhà tạo mô hình để họ biết được làm cách nào có thể đưa ra những lời miêu tả Use Case rõ ràng hơn, minh bạch hơn, và chỉ ra những điểm còn thiếu. Một khi vai trò của tất cả các tác nhân đã được diễn và thực thi, và tất cả các Use Case đã được thực thi theo kiểu này, đó là thời điểm mà người ta nói một quá trình thử nghiệm của mô hình Use Case đã hoàn tất.

Thực hiện các Use Case

Use Case là những lời miêu tả độc lập với sự thực thi các chức năng của hệ thống. Điều đó có nghĩa là Use Case sẽ được thực hiện (thực thể hóa) trong hệ thống, vậy nên trách nhiệm để thực thi các hành động được miêu tả trong tài liệu Use Case đều được phân bổ về cho các đối tượng cộng tác thực thi chức năng đó.

Các nguyên tắc của UML cho việc thực hiện các Use Case là: Một Use Case sẽ được thực hiện trong một sự cộng tác (collaboration): Một sự cộng tác chỉ ra một giải pháp (phụ thuộc vào sự thực thi nội bộ) của một Use Case sử dụng các khái niệm lớp/đối tượng và mối quan hệ giữa chúng đối với nhau (gọi là *ngữ cảnh* – *context* của sự cộng tác) cũng như sự tương tác giữa chúng để đạt tới chức năng mong muốn (gọi là chuỗi tương tác của sự cộng tác). Kí hiệu cho sự cộng tác là một hình ellipse có chứa tên của sự cộng tác đó.

Một sự cộng tác được trình bày trong UML qua một loạt các biểu đồ chỉ ra cả ngữ cảnh lẫn chuỗi tương tác giữa các thành phần tham gia: thành phần tham gia trong một sự cộng tác là một loạt các lớp (và trong một thực thể cộng tác: các đối tượng). Các biểu đồ sử dụng ở đây là biểu đồ cộng tác, biểu đồ chuỗi và biểu đồ hoạt động. Cần phải sử dụng loại biểu đồ nào để tạo ra một bức tranh bao quát về sự cộng tác là quyết định tùy thuộc vào từng trường hợp cụ thể. Trong một vài trường hợp, chỉ một biểu đồ cộng tác

đã có thể là đủ; nhưng trong các trường hợp khác, người ta nhất thiết cần tới sự kết hợp của nhiều loại biểu đồ khác nhau.

Một cảnh kịch (Scenario) là một thực thể (instance) của một Use Case hay là một sự cộng tác: một cảnh kịch là một chuỗi thực thi cụ thể (một dòng chảy cụ thể của các sự kiện) trình bày một sự thực thể hóa của một Use Case (tức là một lần sử dụng hệ thống). Khi một cảnh kịch được quan sát trong tư cách một Use Case, người ta chỉ miêu tả những ứng xử bên ngoài hướng về phía tác nhân. Khi quan sát một cảnh kịch trong tư cách là một thực thể của sự cộng tác, người ta sẽ miêu tả cả sự thực thi nội tại (các dòng lệnh code) của các lớp tham gia ở đây, thuật toán cũng như thủ tục của chúng cùng sự giao tiếp giữa chúng với nhau.

Tác vụ thực hiện một Use Case là chuyển các bước và hành động khác nhau trong lời miêu tả Use Case thành lớp (Class), thủ tục trong những lớp này cũng như quan hệ giữa chúng với nhau. Nó được miêu tả là động tác phân bổ trách nhiệm của mỗi bước đi trong Use Case vào các lớp tham gia sự cộng tác thực hiện Use Case đó. Tại giai đoạn này, người ta phải tìm ra một giải pháp cung cấp những hành vi hướng ngoại đã được xác định của Use Case; nó được miêu tả trong những thuật ngữ của một sự cộng tác nội bộ trong hệ thống.

Mỗi bước hành động trong Use Case sẽ được chuyển thành thủ tục (operation) trong các lớp tham gia. Một bước trong Use Case sẽ được chuyển thành một loạt các thủ tục tại nhiều lớp; rất hiếm khi xảy ra ánh xạ 1-1 giữa các hành động trong Use Case và các thủ tục được thực thi trong tương tác giữa các đối tượng của các lớp tham gia. Cũng xin nhớ rằng một lớp có thể tham gia nhiều Use Case khác nhau và trách nhiệm cao nhất của lớp nằm chính trong việc kết tập tất cả các vai trò mà lớp này đảm nhận trong các Use Case khác nhau.

Mối quan hệ giữa một Use Case và sự thực thi nó theo khái niệm cộng tác được chỉ ra hoặc qua một mối quan hệ nâng cao (refinement relationship) – biểu thị bằng đoạn thẳng chấm chấm với mũi tên - - -> hay một *hyperlink* ngầm trong một công cụ nào đó. Một hyperlink trong một công cụ sẽ tạo điều kiện chuyển từ việc quan sát một Use Case trong một biểu đồ Use Case sang ngay sự cộng tác thực thi Use Case đó. Các hyperlink cũng được sử dụng để chuyển từ Use Case này sang một cảnh kịch (thường là một mô hình động – biểu đồ hoạt động, biểu đồ chuỗi hay biểu đồ cộng tác) miêu tả một sự thực hiện cụ thể nào đó của Use Case.

Phân bổ trách nhiệm cho các lớp một cách thành công là một tác vụ đòi hỏi kinh nghiệm. Cũng giống như mọi công đoạn hướng đối tượng khác, công việc này mang tính vòng lặp (iterative). Nhà phát triển thử nghiệm với nhiều sự phân bổ trách nhiệm khác nhau và dần dần nâng cấp chúng trong giải pháp của mình cho tới khi tạo ra được một mô hình thực hiện chức năng đó, đồng thời lại đủ mức độ năng động để cho phép tiến hành các sự thay đổi trong tương lai.

Jacobson sử dụng phương pháp định nghĩa ba loại đối tượng căn bản (có nghĩa là ba loại lớp): các đối tượng biên (boundary objects), đối tượng chỉ huy (control objects) và đối tượng thực thể (entity objects). Đối với mỗi Use Case, các loại đối tượng này được sử dụng để miêu tả một sự cộng tác thực thi Use Case. Trách nhiệm của các loại đối tượng kể trên như sau:

- *Đối tượng thực thể*: loại đối tượng này đại diện cho các thực thể của bài toán nằm trong phạm vi mà hệ thống xử lý. Thường chúng mang tính thụ động, theo khái niệm là chúng không tự gây nên các tương tác đối với chúng. Trong một hệ thống thông tin, các đối tượng thực thể thường mang tính trường tồn (persistent) và được lưu trữ trong một hệ ngân hàng dữ liệu. Các đối tượng thực thể thường tham gia vào nhiều Use Case khác nhau.

- *Đối tượng biên*: loại đối tượng này nằm gần đường ranh giới của hệ thống (mặc dù vẫn nằm bên trong hệ thống). Chúng tương tác với các tác nhân nằm bên ngoài hệ thống và nhận thông điệp cũng như gửi thông điệp đến các loại đối tượng khác nằm bên trong hệ thống.

- *Đối tượng chỉ huy*: loại đối tượng này chỉ huy sự tương tác giữa các nhóm đối tượng. Một đối tượng như thế có thể đóng vai trò "bộ phận điều khiển" cho toàn bộ một Use Case hoàn tất, hay nó có thể thực thi một chuỗi hành động chung của nhiều Use Case. Thường thì một đối tượng như vậy chỉ tồn tại trong quá trình thực thi Use Case.

Ba loại đối tượng này có ba kí hiệu khác nhau và có thể được sử dụng khi vẽ các loại biểu đồ miêu tả cộng tác hoặc biểu đồ lớp. Sau khi đã định nghĩa nhiều loại đối tượng khác nhau và xác nhận các cộng tác, người ta có thể dễ dàng đi tìm sự tương tự giữa chúng để một số lớp có thể được sử dụng trong một loạt các Use Case khác nhau. Sử dụng các Use Case theo phương thức này ta có thể tạo nên nền tảng cho việc phân tích và thiết kế hệ thống; qui trình phát triển được Ivar Jacobson gọi là "Qui Trình Phát Triển Theo Use Case" (Use case – driven).

Nhìn chung có nhiều phương pháp khác nhau để phân bổ trách nhiệm từ Use Case về cho các lớp. Có phương pháp đề nghị đầu tiên phải tiến hành phân tích phạm vi bài toán, chỉ ra tất cả các lớp thực thể (thuộc phạm vi bài toán) với mối quan hệ của chúng với nhau. Sau đó nhà phát triển sẽ phân tích từng Use Case và phân bổ trách nhiệm cho các lớp trong mô hình phân tích (analysis model), nhiều khi sẽ thay đổi chúng hoặc bổ sung thêm các lớp mới. Một phương pháp khác lại đề nghị là nên lấy các Use Case làm nền tảng để tìm các lớp, làm sao trong quá trình phân bổ trách nhiệm thì mô hình phân tích của phạm vi bài toán sẽ từng bước từng bước được thiết lập.

Một điểm quan trọng cần phải nhắc lại là công việc ở đây mang tính vòng lặp. Khi phân bổ trách nhiệm cho các lớp, ta có thể phát hiện ra sự thiếu đồng bộ hoặc lỗi trong các biểu đồ lớp và qua đó, dẫn đến việc sửa chữa trong biểu đồ lớp. Những lớp mới sẽ được

nhận dạng và tìm ra nhằm mục đích hỗ trợ cho các Use Case. Trong một số trường hợp, thậm chí có thể xảy ra chuyện phải thay đổi hoặc sửa chữa cả biểu đồ Use Case vì khi hiểu hệ thống một cách sâu sắc hơn, nhà phát triển sẽ nhận ra rằng có một Use Case nào đó đã không được miêu tả chính xác và đúng đắn. Các Use Case giúp chúng ta tập trung vào khía cạnh chức năng của hệ thống, làm sao phải đảm bảo cho nó được miêu tả chính xác và được xây dựng chính xác trong hệ thống. Một trong những vấn đề xảy ra với nhiều phương pháp hướng đối tượng mà không sử dụng đến khái niệm Use Case là chúng tập trung quá nhiều vào cấu trúc tĩnh của các lớp và các đối tượng (nhiều khi người ta gọi là phương pháp mô hình hóa khái niệm – conceptual modeling) nhưng lại bỏ qua các khía cạnh chức năng và khía cạnh động của hệ thống.