



Biểu đồ hoạt động (Activity Diagram)

Bởi:

duongkieuhoa

tonthathoan

Biểu đồ hoạt động (Activity Diagram)

Biểu đồ hoạt động nắm bắt hành động và các kết quả của chúng. Biểu đồ hoạt động tập trung vào công việc được thực hiện trong khi thực thi một thủ tục (hàm), các hoạt động trong một lần thực thi một trường hợp sử dụng hoặc trong một đối tượng. Biểu đồ hoạt động là một biến thể của biểu đồ trạng thái và có một mục tiêu tương đối khác, đó là nắm bắt hành động (công việc và những hoạt động phải được thực hiện) cũng như kết quả của chúng theo sự biến đổi trạng thái. Các trạng thái trong biểu đồ hoạt động (được gọi là các trạng thái hành động) sẽ chuyển sang giai đoạn kế tiếp khi hành động trong trạng thái này đã được thực hiện xong (mà không xác định bất kỳ một sự kiện nào theo như nội dung của biểu đồ trạng thái). Một sự điểm phân biệt khác giữa biểu đồ hoạt động và biểu đồ trạng thái là các hành động của nó được định vị trong các *luồng* (*swimlane*). Một luồng sẽ gom nhóm các hoạt động, chú ý tới khái niệm người chịu trách nhiệm cho chúng hoặc chúng nằm ở đâu trong một tổ chức. Một biểu đồ hoạt động là một phương pháp bổ sung cho việc miêu tả tương tác, đi kèm với trách nhiệm thể hiện rõ các hành động xảy ra như thế nào, chúng làm gì (thay đổi trạng thái đối tượng), chúng xảy ra khi nào (chuỗi hành động), và chúng xảy ra ở đâu (luồng hành động).

Biểu đồ hoạt động có thể được sử dụng cho nhiều mục đích khác nhau, ví dụ như:

- Để nắm bắt công việc (hành động) sẽ phải được thực thi khi một thủ tục được thực hiện. Đây là tác dụng thường gặp nhất và quan trọng nhất của biểu đồ hoạt động.
- Để nắm bắt công việc nội bộ trong một đối tượng.
- Để chỉ ra một nhóm hành động liên quan có thể được thực thi ra sao, và chúng sẽ ảnh hưởng đến những đối tượng nằm xung quanh chúng như thế nào.
- Để chỉ ra một trường hợp sử dụng có thể được thực thể hóa như thế nào, theo khái niệm hành động và các sự biến đổi trạng thái của đối tượng.

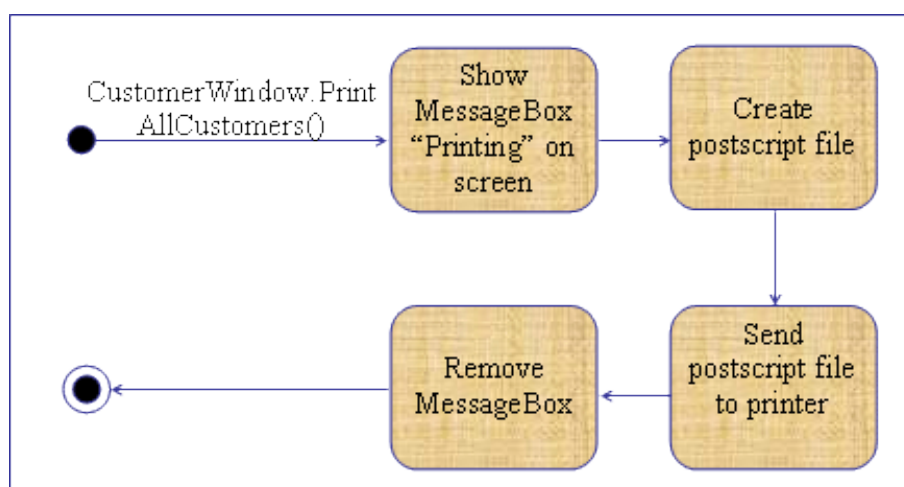
Biểu đồ hoạt động (Activity Diagram)

- Để chỉ ra một doanh nghiệp hoạt động như thế nào theo các khái niệm công nhân (tác nhân), qui trình nghiệp vụ (workflow), hoặc tổ chức và đối tượng (các khía cạnh vật lý cũng như tri thức được sử dụng trong doanh nghiệp).

Biểu đồ hoạt động có thể được coi là một loại Flow chart. Điểm khác biệt là Flow Chart bình thường ra chỉ được áp dụng đối với các qui trình tuần tự, biểu đồ hoạt động có thể xử lý cả các qui trình song song.

Hành động và sự thay đổi trạng thái

Một hành động được thực hiện để sản sinh ra một kết quả. Việc thực thi của thủ tục có thể được miêu tả dưới dạng một tập hợp của các hành động liên quan, sau này chúng sẽ được chuyển thành các dòng code thật sự. Theo như định nghĩa ở phần trước, một biểu đồ hoạt động chỉ ra các hành động cùng mối quan hệ giữa chúng và có thể có một điểm bắt đầu và một điểm kết thúc. Biểu đồ hoạt động sử dụng cũng cùng những ký hiệu như trong biểu đồ trạng thái bình thường.



Khi một người gọi tác vụ PrintAllCustomer (trong lớp CustomerWindow), các hành động khởi động. hành động đầu tiên là hiện một hộp thông báo lên màn hình; hành động thứ hai là tạo một tập tin postscript; hành động thứ ba là gửi file postscript đến máy in; và hành động thứ tư là xóa hộp thông báo trên màn hình. Các hành động được chuyển tiếp tự động; chúng xảy ra ngay khi hành động trong giai đoạn nguồn được thực hiện.

Các sự thay đổi có thể được bảo vệ bởi các điều kiện canh giữ (Guard-condition), các điều kiện này phải được thỏa mãn thì sự thay đổi mới nổ ra. Một ký hiệu hình thoi được sử dụng để thể hiện một quyết định. Ký hiệu quyết định có thể có một hoặc nhiều sự thay đổi đi vào và một hoặc nhiều sự thay đổi đi ra được dán nhãn đi kèm các điều kiện bảo vệ. Bình thường ra, một trong số các sự thay đổi đi ra bao giờ cũng được thỏa mãn (là đúng).

Một sự thay đổi được chia thành hai hay nhiều sự thay đổi khác sẽ dẫn đến các hành động xảy ra song song. Các hành động được thực hiện đồng thời, mặc dù chúng cũng

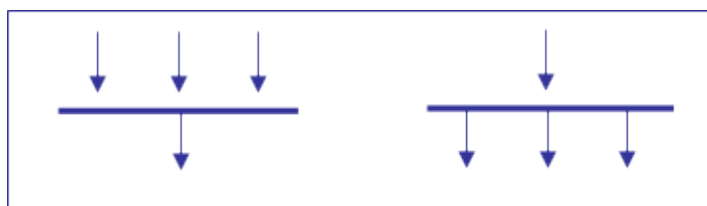
Biểu đồ hoạt động (Activity Diagram)

có thể được thực hiện lần lượt từng cái một. Yếu tố quan trọng ở đây là tất cả các thay đổi đồng thời phải được thực hiện trước khi chúng được thống nhất lại với nhau (nếu có). Một đường thẳng nằm ngang kẻ đậm (còn được gọi là thanh đồng bộ hóa – Synchronisation Bar) chỉ rằng một sự thay đổi được chia thành nhiều nhánh khác nhau và chỉ ra một sự chia sẻ thành các hành động song song. Cũng đường thẳng đó được sử dụng để chỉ ra sự thống nhất các nhánh.

Kí hiệu UML cho các thành phần **căn bản của biểu đồ hoạt động**:

- *Hoạt động (Activity)*: là một qui trình được định nghĩa rõ ràng, có thể được thực thi qua một hàm hoặc một nhóm đối tượng. Hoạt động được thể hiện bằng hình chữ nhật bo tròn cạnh.

- Thanh đồng bộ hóa (Synchronisation bar): chúng cho phép ta mở ra hoặc là đóng lại các nhánh chạy song song nội bộ trong tiến trình.



Thanh đồng bộ hóa

- *Điều kiện canh giữ (Guard Condition)*: các biểu thức logic có giá trị hoặc đúng hoặc sai. Điều kiện canh giữ được thể hiện trong ngoặc vuông, ví dụ:

[Customer existing].

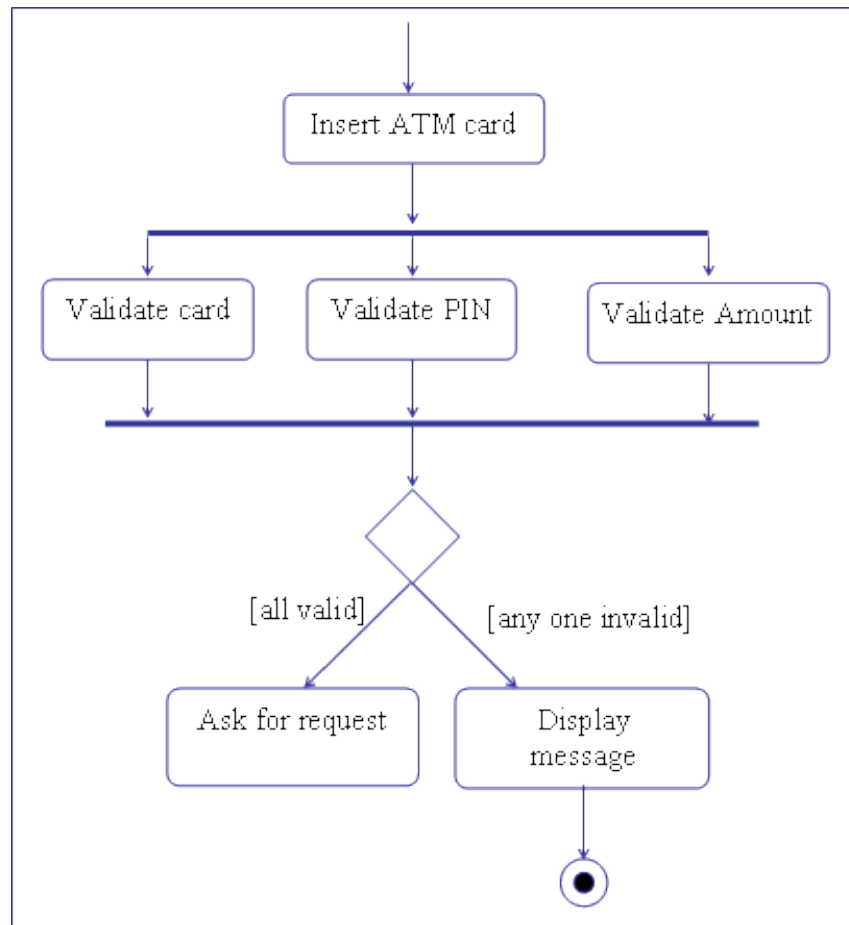
- *Điểm quyết định (Decision Point)*: được sử dụng để chỉ ra các sự thay đổi khả thi. Kí hiệu là hình thoi.

Hình sau đây miêu tả một đoạn biểu đồ hoạt động của máy ATM. Sau khi thẻ được đưa vào máy, ta thấy có ba hoạt động song song:

- Xác nhận thẻ
- Xác nhận mã số PIN
- Xác nhận số tiền yêu cầu được rút

Chỉ khi sử dụng biểu đồ hoạt động, các hoạt động song song như vậy mới có thể được miêu tả. Mỗi một hoạt động xác nhận bản thân nó cũng đã có thể là một quá trình riêng biệt.

Biểu đồ hoạt động (Activity Diagram)



Biểu đồ hoạt động của máy ATM

Vòng đời của đối tượng (Object lifecycle)

Vòng đời mà một đối tượng đi qua phụ thuộc vào loại đối tượng. Có hai loại vòng đời khác nhau đối với một đối tượng: vòng đời sinh ra rồi chết đi; và vòng đời vòng lặp.

Vòng đời sinh ra và chết đi:

Trong một vòng đời sinh ra rồi chết đi:

- Sẽ có một hay nhiều trạng thái nơi đối tượng bắt đầu tồn tại. Những trạng thái này được gọi là trạng thái tạo ra đối tượng.
- Sẽ có một hay nhiều trạng thái đóng tư cách là điểm kết thúc cho vòng đời của một đối tượng. Những trạng thái này được gọi là trạng thái kết thúc.

Có hai loại trạng thái kết thúc. Một dạng trạng thái kết thúc là loại nơi đối tượng bị hủy và không tiếp tục tồn tại nữa. Loại thứ hai là dạng trạng thái kết thúc mà sau đó đối tượng sẽ được lưu trữ lại hoặc chuyển sang trạng thái im lặng. Đối tượng tiếp tục tồn tại nhưng không tiếp tục thể hiện ứng xử động.

Biểu đồ hoạt động (Activity Diagram)

Sau trạng thái khởi tạo và trước trạng thái kết thúc, đối tượng có thể đi qua một hoặc là nhiều trạng thái trung gian. Tại mỗi một thời điểm, đối tượng phải ở một trạng thái hiện thời.

Không có một điểm nào sau trạng thái khởi tạo và trước trạng thái kết thúc mà đối tượng lại không có trạng thái.

Ví dụ cho đối tượng có vòng đời sinh ra và chết đi: khách hàng, tài khoản.

Vòng đời lặp

Khác với trường hợp sinh ra và chết đi, trong vòng đời vòng lặp:

- Đối tượng được coi là đã luôn luôn tồn tại ở đây và sẽ còn mãi mãi tiếp tục tồn tại.

- Không có trạng thái khởi tạo cũng không có trạng thái kết thúc.

Mặc dù thật ra đối tượng đã được tạo ra tại một thời điểm nào đó và cũng sẽ thật sự bị hủy diệt tại một thời điểm nào đó, nhưng nó vẫn được coi là luôn luôn tồn tại và có mặt. Thường thì những đối tượng tỏ ra có một vòng đời vòng lặp sẽ được tạo ra tại thời điểm cài đặt hệ thống và sẽ chết đi khi hệ thống kết thúc.

Một đối tượng với vòng đời vòng lặp sẽ có một hoặc là nhiều trạng thái "ngủ yên". Đó là những trạng thái nơi đối tượng nằm chờ một sự kiện xảy ra. Bên cạnh đó, đối tượng cũng luôn luôn ở trạng thái hiện thời.

Ví dụ cho đối tượng có vòng đời lặp lại: máy rút tiền tự động (ATM), nhân viên thu ngân.

Xem xét lại mô hình động

Thăm vấn biểu đồ trạng thái

Sau khi đã hoàn tất các thành phần căn bản của mô hình động như các biểu đồ tuần tự, biểu đồ cộng tác, biểu đồ trạng thái và biểu đồ hoạt động, nhóm phát triển có thể phác thảo biểu đồ thành phần và biểu đồ triển khai. Biểu đồ triển khai có thể được coi là biểu đồ cuối cùng trong mô hình động. Tới thời điểm này, có thể coi là ta đã hoàn tất một phiên bản của mô hình động.

Phần quan trọng nhất trong mô hình này là biểu đồ trạng thái. Hãy tìm câu trả lời cho một loạt các câu hỏi để xác định xem biểu đồ trạng thái đã đúng đắn và có một mức độ chi tiết thích hợp hay chưa. Công việc này cần nhắm tới hai mục đích:

- Kiểm tra tính trọn vẹn của mô hình

Biểu đồ hoạt động (Activity Diagram)

- Đảm bảo mọi điều kiện gây lỗi đã được xử lý

Trong giai đoạn này, có thể sẽ có các cảnh kịch (scenario) mới xuất hiện và gia nhập phạm vi quan sát của chúng ta, nếu trước đó có một số trạng thái chưa được xử lý. Những tình huống loại này là loại vấn đề có thể được giải quyết, song có thể né tránh qua việc xác định thật đầy đủ các sự kiện và trạng thái.

Phối hợp sự kiện

Bước cuối cùng là một vòng kiểm tra bổ sung nhằm đảm bảo tính đúng đắn của mô hình động:

- Kiểm tra để đảm bảo mỗi thông điệp đều có đối tượng gửi và đối tượng nhận. Trong một số trường hợp, số liệu chính xác của những đối tượng nhận sự kiện có thể không được biết tới, nhưng chúng ta phải đảm bảo rằng chúng ta biết những lớp nào sẽ xử lý những sự kiện này.

- Hãy nghiên cứu mô hình theo khía cạnh trạng thái, tìm ra những trạng thái không có trạng thái dẫn trước và không có trạng thái tiếp theo. Những trạng thái này rất có thể là trạng thái khởi đầu hoặc trạng thái kết thúc. Mặc dù vậy, nếu trạng thái đó không thuộc về một trong hai loại trạng thái kia, rất có thể đây là một triệu chứng cho thấy mô hình còn thiếu điều gì đó.

- Nhìn chung, tất cả các trạng thái thường đều có trạng thái dẫn trước và trạng thái tiếp sau.

- Hãy lần theo hiệu ứng của các sự kiện đi vào (entry) để đảm bảo là chúng tương thích với các trường hợp sử dụng nơi chúng xuất phát. Để làm điều này, hãy lần theo một sự kiện từ một đối tượng này đến đối tượng khác, kiểm tra xem mỗi sự kiện có phù hợp với trường hợp sử dụng hay không. Trong trường hợp có mâu thuẫn, hãy sửa lại biểu đồ trạng thái hoặc trường hợp sử dụng để đảm bảo sự nhất quán.

- Kiểm tra lại những lỗi đồng bộ, có thể chúng là kết quả của một sự kiện không chờ đợi.

Bao giờ thì sử dụng biểu đồ nào

Không cần phải vẽ tất cả các loại biểu đồ động cho tất cả các loại hệ thống. Mặc dù vậy, trong một số trường hợp khác nhau chúng ta nhất thiết phải cần đến một số loại biểu đồ động nhất định. Sau đây là một vài lời mách bảo có thể giúp giải thích một vài điều còn chưa thông tỏ về việc sử dụng các loại biểu đồ động.

Biểu đồ tuần tự và biểu đồ cộng tác được vẽ khi chúng ta muốn xem xét ứng xử động của nhiều đối tượng/ lớp trong nội bộ một cảnh kịch của một trường hợp sử dụng. Biểu

Biểu đồ hoạt động (Activity Diagram)

đồ tuần tự và biểu đồ cộng tác rất hữu dụng trong việc chỉ ra sự cộng tác giữa các đối tượng, nhưng chúng lại không hữu dụng khi muốn miêu tả ứng xử chính xác của một đối tượng.

Biểu đồ trạng thái được sử dụng để thể hiện ứng xử chính xác của một đối tượng.

Biểu đồ hoạt động được sử dụng để thể hiện lối ứng xử xuyên suốt nhiều trường hợp sử dụng hoặc các tiểu trình xảy ra song song của một lần thực thi.

Biểu đồ thành phần và biểu đồ triển khai được sử dụng để chỉ ra mối quan hệ vật lý giữa phần mềm và các thành phần phần cứng trong hệ thống.

Lớp con và biểu đồ trạng thái

Tất cả các lớp con đều thừa kế cả thuộc tính cũng như các thủ tục của lớp cha. Vì vậy, một lớp con cũng sẽ thừa kế cả mô hình động của lớp cha.

Ngoài biểu đồ trạng thái được thừa kế, lớp con cũng có biểu đồ trạng thái riêng của nó. Biểu đồ trạng thái của một lớp cha sẽ được mở rộng để bao chứa lối ứng xử chuyên biệt của lớp con.

Biểu đồ trạng thái của lớp con và biểu đồ trạng thái của lớp cha phải được bảo trì riêng biệt và độc lập. Biểu đồ trạng thái của lớp con cần phải được định nghĩa sử dụng các thuộc tính của lớp con chứ không phải chỉ bằng các thuộc tính của lớp cha. Mặt khác, vẫn có một sự móc nối ngoài ý muốn của lớp cha đến với lớp con thông qua các thuộc tính mà chúng sử dụng chung, ví dụ chỉ nên xem xét biểu đồ trạng thái cho các tài khoản có kỳ hạn theo phương diện sự thay đổi của chính các thuộc tính của chúng, chứ không phải là thuộc tính của lớp cha. Ta phải thực hiện như vậy để né tránh trường hợp trộn lẫn thuộc tính của lớp con và lớp cha.

Việc tuân thủ quy tắc kể trên trong quá trình vẽ biểu đồ trạng thái cho một lớp con sẽ đảm bảo tính môđun cho động tác mở rộng của bạn.