**12.2 ORGANIZING DETAILED REQUIREMENTS  
-----**

Yêu cầu thay đổi liên tục, do đó việc viết yêu cầu phải được tổ chức tốt và dễ dàng để cập nhật. Để đánh giá đúng giá trị các yêu cầu chi tiết tổ chức một cách cẩn thận, ta hãy xem xét văn bản yêu cầu chi tiết cho trò chơi Encounter. Lưu ý rằng các yêu cầu này vẫn còn nguyên và không được kiểm tra.

Mỗi nhân vật trong game có một tên.

Mỗi nhân vật game có cùng một tập các thuộc tính, đều có một giá trị kiểu dấu chấm động.

Mất ít hơn một giây để tính toán các kết quả của một sự trận chiến.

Mỗi khu vực có một tập hợp cụ thể gọi là "những thuộc tính phẩm chất cần thiết." Ví dụ, khu vực chiến đấu đòi hỏi sức mạnh và sức chịu đựng; phòng sinh hoạt đòi hỏi sự nhạy cảm và trí tuệ.

Khi hai nhân vật trong Encounter đang ở cùng khu vực cùng một lúc họ có thể thể chọn hoặc buộc phải đánh nhau.

Mỗi nhân vật trò chơi có các số điểm sống.

Tổng các giá trị của các thuộc tính phẩm chất của một nhân vật có liên quan đến các khu vực sẽ được gọi như giá trị khu vực của nhân vật. Trong một trận chiến, hệ thống so sánh giá trị khu vực của nhân vật và tính toán kết quả của sự tham gia nhân vật đó.

Tên của bất kỳ nhân vật nào trong Encounter sẽ không có nhiều hơn 15 chữ cái.

Khi nó phát triển, một danh sách chưa được tổ chức như ở trên nhanh chóng trở nên không thể quản lý.

* Vấn đề kích thước của nó làm cho nó khó hiểu ngay cả trên một đơn vị, đó là chưa tính đến nó phát triển thành hàng trăm, hàng ngàn.

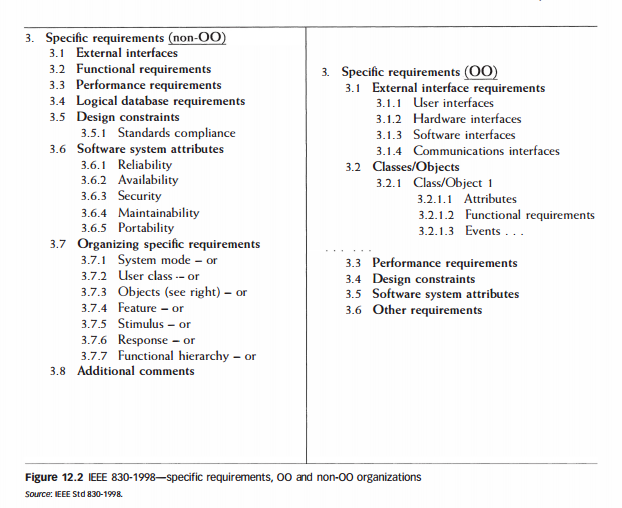
• Các yêu cầu của các loại hỗn hợp: yêu cầu thực hiện phải được xử lý khác nhau.

• Một số yêu cầu tự nhiên đi liền với những yêu cầu liên quan.

• Rất khó để xác định vị trí một yêu cầu cụ thể.

Yêu cầu chi tiết chức năng có thể được tổ chức theo nhiều cách phân loại, bao gồm cả theo tính năng, trường hợp sử dụng, CUI, giai đoạn, và lớp. Chúng tôi mô tả mỗi phương pháp chi tiết hơn trong các phần tiếp theo. Những công cụ cho yêu cầu quản lý giúp đỡ rất nhiều. Tuy nhiên, quyết định như thế nào để tổ chức yêu cầu chi tiết ở nơi đầu tiên là quan trọng bởi vì các nhóm giới thiệu cho họ liên tục nếu tài liệu được thực hiện tốt.

Các tiêu chuẩn IEEE 830- 1998 cung cấp tài liệu mẫu cho một số cách để phân loại các yêu cầu chi tiết. Hình 12. 2 cho thấy các đối tượng của IEEE thông thường và 830- 1 998 tiêu chuẩn. Việc phân loại đối tượng sử dụng các lớp / đối tượng như một phương pháp tổ chức yêu cầu chức năng. SRS thường được thiết kế riêng cho nhu cầu của công ty hoặc nhóm bằng cách thêm hoặc sửa đổi phần cho phù hợp. Ví dụ, tổ chức 00 thiếu một phần tương đương với 3. 4 trong không-OO tổ chức "các yêu cầu cơ sở dữ liệu Iogical." Nghiên cứu trường hợp Encounter sử dụng một hình thức sửa đổi của IEEE 00 phong cách và bao gồm một phần cho các trường hợp sử dụng. Hình 1 2. 3 bản đồ các yêu cầu chi tiết các phần của mục 3 để thể loại yêu cầu nó mô tả.

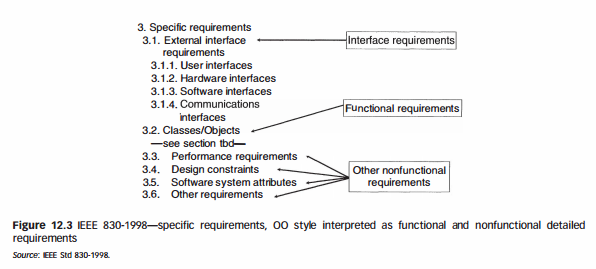


Nó có thể được khuyến khích để tổ chức các yêu cầu chi tiết bằng một sự kết hợp của các phân loại. Ví dụ, tổ chức dựa trên tính năng có thể được sử dụng trong các cấu hình, thực hiện, và sao lưu trạng thái của một ứng dụng kế toán. Các yêu cầu cho một hệ thống nhà máy tự động hóa có thể được tổ chức ở cấp cao nhất của chức năng (một phần sản xuất và lắp ráp); sau đó họ có thể được tổ chức bởi trong mỗi lớp.

Điều này có nghĩa là phương thức tổ chức yêu cầu chi tiết đôi khi có thể xảy ra liên quan đến kiến trúc hoặc triển khai các ứng dụng. Ví dụ, nếu thiết kế theo hướng đối tượng, chi tiết yêu cầu bởi trường hợp sử dụng hoặc theo lớp tổ chức nên được xem xét bởi vì họ tạo điều kiện truy xuất nguồn gốc. Nếu ứng dụng vay chính nó để một phân tích chức năng rõ ràng, sau đó tổ chức các yêu cầu của hệ thống phân cấp tính năng có thể thích hợp.

**12.2.1 Yêu cầu chi tiết tổ chức theo tính năng**

Cách thức cổ điển nhất của tổ chức yêu cầu chi tiết là bởi tính năng. Hình sau cung cấp một danh sách đơn giản các chức năng như sau đối với trò chơi Encounter. Nhiều tính năng được xác định bởi một cặp stimulusand phản ứng, đối với yêu cầu 127.



Sẽ có một nhóm vị trí siêu liên kết ở mọi lối ra vào mọi lĩnh vực.

126. Các nhân vật ngoài di chuyển từ vùng này sang vùng lân cận vào thời điểm ngẫu nhiên, với trung bình ba giây.

127. Khi người dùng nhấp chuột vào nút "Đặt chất lượng", cửa sổ trong hình xx xuất hiện trên màn hình.

Sắp xếp các yêu cầu bởi tính năng có lợi thế là đơn giản nhưng bất lợi là việc không tiếp cận; bởi vì nó cho phép nhảy từ một tính năng trong một phần của ứng dụng đến một tính năng trong một phần khác hoàn toàn. Ví dụ, nếu chúng ta muốn thay đổi cách thức mà các nhân vật ngoài di chuyển về, chúng ta sẽ phải xác định rằng các yêu cầu có liên quan là số 126. Làm thế nào chúng ta sẽ biết những yêu cầu khác bị ảnh hưởng? Công cụ tìm kiếm có thể giúp đỡ rất nhiều. Một nhược điểm nữa là việc thiếu truy xuất nguồn gốc. Một cách để áp đặt trật tự trên danh sách tính năng chức năng là để sắp xếp chúng trong một hệ thống phân cấp chức năng (tức là bằng phân hủy các ứng dụng vào một tập hợp các chức năng cao cấp, và sau đó các thành subfunctions, vv. ). Ví dụ, các yêu cầu cho một chương trình ngân sách tại nhà có thể được phân tích thành (1) kiểm tra chức năng, (2) chức năng tiết kiệm, và (3) các chức năng đầu tư. Các chức năng kiểm tra yêu cầu có thể được phân hủy thêm vào chức năng séc, hòa giải, và báo cáo v.v.

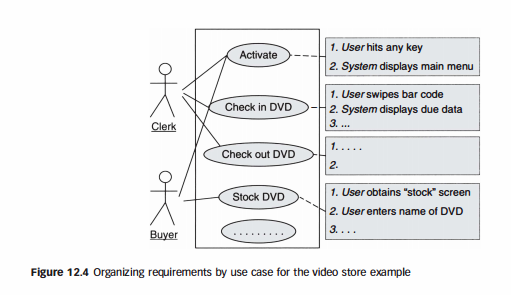
**12.2.2 Yêu cầu chi tiết tổ chức theo trường hợp**

Sử dụng trường hợp (giới thiệu trong chương 11) được nhận xét rằng xảy ra nhiều trong tự nhiên trình tự hoạt động. Ví dụ, một yêu cầu cá nhân như "một ứng dụng cửa hàng video sẽ cho phép nhập vào tiêu đề của video mới" thường diễn ra như là một phần của một chuỗi các giao dịch. Một sơ đồ trường hợp sử dụng cho thấy một bộ sưu tập các trường hợp sử dụng cho một cửa hàng ứng dụng video được minh họa trong hình 12 .4.

Các tác giả chủ trương cung cấp các trường hợp sử dụng cho các yêu cầu cao cấp. Một người có thể có tùy chọn sử dụng nguyên tắc tổ chức cùng này cho các yêu cầu chi tiết. Trong trường hợp này, chúng ta sẽ nhóm các chi tiết yêu cầu theo từng trường hợp sử dụng. Sau đây là một ví dụ về một nhóm các yêu cầu sẽ xuất hiện trong SRS.

3 .7. 1 Kiểm tra trong DVD

Đây là một phiên bản chi tiết của phần xx. (Lưu ý 1, xem "Ghi chú" giải thích trên trang tiếp theo)



Ứng dụng này sẽ cung cấp các khả năng sau để tương tác với các nhân viên cửa hàng. Những

yêu cầu của màn hình chính. (Lưu ý 2 )

3. 7. 1. 1 Bước 1: Kiểm tra trong nút nhấn (Lưu ý 3)

Các nhân viên bán hàng thì có thể nhấn nút Check In, sau đó, màn hình Check In sẽ xuất hiện (Mô tả trong phần xx trên).

3 .7. 1 .2 Bước 2: Điền in các trường (Lưu ý 4)

Các nhân viên bán hàng sẽ có thể điền vào các trường văn bản sau đây trong màn hình Check In

3 .7. 1 .2. 1 Video trường tên

Trường này sẽ cho phép 30 ký tự chữ và số, kể cả space.

Lưu ý 1: đoạn này tương ứng với phân đoạn Kiểm tra trong trường hợp sử dụng đĩa DVD được mô tả trong các cấp cao các yêu cầu.

Lưu ý 2: Phần này giới thiệu các trường hợp sử dụng và nêu điều kiện tiên quyết, nếu có. Một điều kiện tiên quyết là một thực tế giả định đúng tại thời điểm khởi trường hợp sử dụng.

Lưu ý 3: Điều này tương ứng với bước đầu tiên của các trường hợp sử dụng.

Lưu ý 4: Vì đây là những yêu cầu chi tiết, họ phải xác định các yêu cầu hoàn toàn. Các chi tiết sẽ không được quy định ở nơi khác.

Quá trình phát triển phần mềm thống nhất ủng hộ việc tổ chức tất cả các yêu cầu của các trường hợp sử dụng.

**12.2.3 Yêu cầu chi tiết tổ chức theo giao diện người dùng**

Ứng dụng hiển thị GUI là những phương tiện mà người dùng đánh giá chúng, vì vậy chúng có thể được tự nhiên cung cấp các yêu cầu tổ chức theo cách này. Sử dụng cách này, các yêu cầu đối với các trò chơi Encounter sẽ như sau:

1. Khu vực GUI

Khu vực GUI sẽ có sự xuất hiện trong hình xx. Khi người dùng nhấp chuột vào nút "Đặt chất lượng ", cửa sổ trong hình xx xuất hiện trên màn hình. Khi người dùng nhấp vào ...

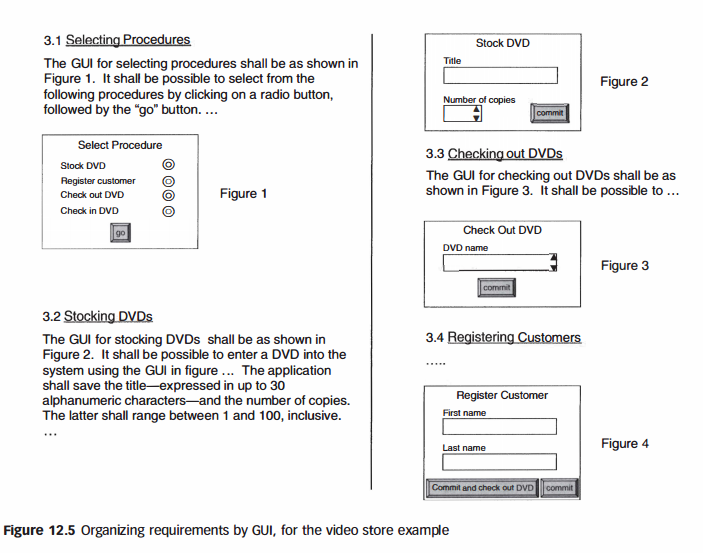
2. Dungeon GUI ...

3. Phòng khách GUI ...

4. Thiết lập chất lượng GUI ...

5. Xem Phẩm chất Window ...

Lợi thế của tổ chức yêu cầu bởi GUI là kết nối trực tiếp với việc sử dụng các ứng dụng. Một lợi thế khác là truy xuất nguồn gốc: chúng ta có một cơ hội tốt để theo dõi các yêu cầu liên quan đến một lớp GUI. Một bất lợi của phương tiện này của tổ chức là nó lỗi bao phủ tất cả các yêu cầu. Ví dụ Encounter, chúng ta cần phải mô tả các yêu cầu cho sự tương tác của nhân vật của người chơi và các nhân vật ngoài. Như một ví dụ, hình 12.5 minh họa một tổ chức của các yêu cầu lưu trữ video bằng GUI. Đặc điểm kỹ thuật của GUI được thảo luận thêm tại mục 12.3.



**12.2.4 Yêu cầu chi tiết theo giai đoạn:**

Cách này là thu thập yêu cầu chi tiết áp dụng cho mỗi giai đoạn. Ví dụ, các yêu cầu cho một ứng dụng điều khiển một quá trình hóa học có thể được phân loại theo quá trình của chính nó (bắt đầu lên, phản ứng, làm mát, vv). Trong mỗi loại, các sự kiện ảnh hưởng đến các ứng dụng trong trạng thái đã được liệt kê. Phân loại theo trạng thái có thể thích hợp khi yêu cầu cho mỗi trạng thái là khá khác biệt. Ví dụ, một hệ thống kế toán có thể được yêu cầu để xử lí hoàn toàn khác nhau tùy thuộc vào việc nó là đang trong giai đoạn cấu hình, thực hiện, hoặc sao lưu trạng thái.

**12.2.5 Yêu cầu chi tiết theo lớp:**

Cách này thực hiện theo cách hướng đối tượng để tổ chức yêu cầu, một phân loại được trình diễn lần đầu tương đương với việc lựa chọn các lớp; sau đó các yêu cầu cá nhân được đặt vào các lớp kết quả. Các lớp sử dụng để phân loại các yêu cầu được biết đến như là các lớp miền. Các lớp miền đại diện cho đối tượng hiện thực trên thế giới hoặc các khái niệm trong ứng dụng. Ví dụ, một ứng dụng ngân hàng sẽ có các lớp miền như khách hàng, kiểm tra tài khoản, và tiết kiệm. Bước đầu tiên phổ biến trong việc xác định các lớp miền là để thu thập các định nghĩa hoặc tương đương được sử dụng trong các yêu cầu cao cấp. Sau đó chúng tôi làm cho mỗi chức năng chi tiết yêu cầu tương ứng với một chức năng trong ngôn ngữ đích.

Một bất lợi của tổ chức yêu cầu theo lớp là rủi ro mà sau này chúng ta thay đổi các lớp, do đó phá vỡ sự tương ứng giữa các yêu cầu và thiết kế. Điều này được thảo luận bởi Jordan, Smilan, và Wilkinson trong [2]. Trong thực tế, một số nhà phát triển sử dụng theo lớp để tổ chức các yêu cầu không nghiêm trọng mục đích để sử dụng các lớp cho các thiết kế. Trong trường hợp này, truy xuất nguồn gốc bị tổn thất, nhưng có ít áp lực xác định các lớp kéo dài từ rất sớm trong quá trình này. Một bất lợi của việc phân loại này là nó đòi hỏi chúng ta chọn lớp rất sớm trong chu kỳ phát triển, và nhiều người cho rằng chúng tôi đang thực hiện một cách hiệu quả thiết kế khi làm như vậy. Hãy xem xét các trường hợp nghiên cứu Encounter như là một ví dụ. Chọn lớp như PlayerCharacter và các yêu cầu thời gian là vô hại khi sử dụng các các lớp. Mặt khác, nó có thể được lập luận hợp lý rằng khi các đối tượng AreaConnection tham chiếu

đối tượng khu vực mà chúng kết nối là một quyết định thiết kế.

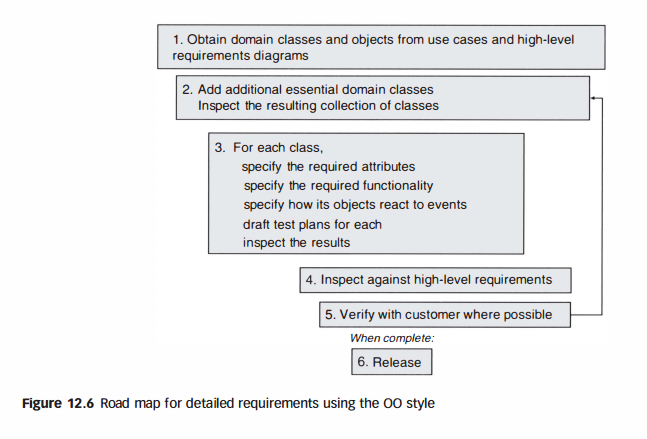
Những lợi thế lớn để tổ chức yêu cầu bởi các lớp sẽ được sử dụng trong thiết kế là nó thúc đẩy sự tương ứng chặt chẽ giữa yêu cầu, thiết kế và triển khai thực hiện. Đây là một lợi ích quan trọng cho cách sử dụng mô hình 00 trong mọi trường hợp.

Một danh sách điển hình cho việc thu thập yêu cầu chi tiết chức năng bằng cách sử dụng mô hình 00 như sau:

1. Liệt kê các khái niệm được đề cập trong trường hợp sử dụng và các yêu cầu cao cấp khác.

2. Bộ sưu tập kết quả của các lớp thường là không đầy đủ.

3. Đối với mỗi lớp thu được, viết xuống tất cả các chức năng cần thiết của ứng dụng chủ yếu liên quan đến lớp đó. Điều này được thực hiện dưới hình thức các thuộc tính và chức năng. Ví dụ, "mỗi khách hàng sẽ có một tên" (một thuộc tính được liệt kê theo khoản khách hàng)



và "các ứng dụng sẽ có thể tính tổng tài sản của từng khách hàng" (một chức năng được liệt kê dưới Khách hàng). Trong (yêu cầu) tài liệu mà bạn đang viết, tránh sử dụng thuật ngữ "c1ass." Sử dụng bình thường, không kỹ thuật bằng tiếng Anh.

4. Kiểm tra các yêu cầu chi tiết như quá trình phát triển.

5. Một cách lý tưởng, kế hoạch kiểm tra đối với từng yêu cầu chi tiết cần được thiết kế cùng một lúc, như được giải thích dưới đây.

6. Thanh tra các yêu cầu chi tiết đối với các yêu cầu cao cấp.

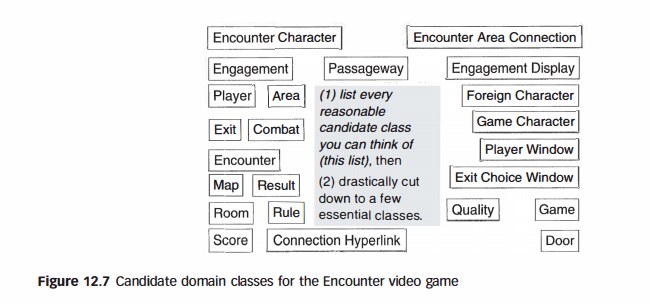
7. Xác nhận các yêu cầu chi tiết với khách hàng.

8. Quay lại Bước I để yêu cầu nhiều hơn.

Nhớ lại rằng các đối tượng chính cho các yêu cầu chi tiết bao gồm các nhà phát triển. Tuy nhiên, khách hàng là cực kỳ quan tâm đến các chi tiết, quá. Hình 12. 6 tóm tắt các bước sau.

Đó là một lỗi phổ biến khi phân loại các yêu cầu của lớp học để sử dụng ngôn ngữ thiết kế thay vì

đồng bằng tiếng Anh. Ví dụ, các ngôn ngữ sau đây được chấp nhận.



**12.2.6 Phương pháp Phân loại: Ưu điểm và nhược điểm**

Hình 12.09 tóm tắt những cách khác nhau để tổ chức yêu cầu chi tiết chức năng mà chúng ta đã thảo luận trong các phần trước, cùng với lợi thế và bất lợi tương đối của chúng.

