THÊM KÝ HIỆU CHO ĐẶC TẢ YÊU CẦU ĐỂ DỄ DÀNG SINH BỘ KIỂM THỬ VÀ BẢN MẪU GIAO DIỆN

● VŨ DIỆU HƯƠNG - BÙI QUANG TRƯỜNG - TRẦN THỊ NHUNG

TÓM TẮT:

Sinh bộ kiểm thử và bản mẫu giao diện cần được thực hiện sớm và nhanh chóng để sử dụng trong việc trao đổi, xác nhận yêu cầu giữa nhà phát triển và khách hàng. Hoạt động này cũng thường được lặp đi lặp lại nhiều lần mỗi khi có sự thay đổi nhỏ về yêu cầu phần mềm. Nếu được thực hiện thủ công thì với số ca kiểm thử lớn và sự đòi hỏi tạo ra các bản mẫu nhanh chóng, tốn ít chi phí sẽ rất khó khăn cho phía phát triển. Tự động hóa hoạt động sinh bộ kiểm thử và bản mẫu giao diện người dùng là một giải pháp cho tình huống này. Bài báo đề xuất việc thêm vào các chú thích (notations) cho đặc tả ca sử dụng UML, để từ đó dễ dàng sinh bộ kiểm thử và các bản mẫu giao diện người dùng, giúp tiết kiệm thời gian, cũng như nâng cao hiệu năng của các dự án công nghệ thông tin từ khâu ban đầu đến khâu người sử dụng cuối.

Từ khóa: sinh ca kiểm thử, sinh bản mẫu giao diện, đặc tả yêu cầu phần mềm, ngôn ngữ UML.

1. Đặt vấn đề

Trong lĩnh vực phát triển phần mềm, đặc tả yêu cầu (requirement specifications) mô tả hành vi bên ngoài của phần mềm [7]. Chúng mô tả các yêu cầu chức năng - là các dịch vụ mà hệ thống cung cấp và mô tả các yêu cầu phi chức năng - là các ràng buộc thiết kế và các yếu tố khác, tạo nên một bản mô tả đầy đủ về các yêu cầu cho phần mềm. Đặc tả yêu cầu là chế tác đầu tiên của quy trình phát triển phần mềm. Mọi hoạt động tiếp theo trong quy trình như thiết kế, lập trình, kiểm thử và bảo trì đều tham chiếu đến đặc tả yêu cầu. Có nhiều cách để mô tả yêu cầu: ngôn ngữ tự nhiên, ngôn ngữ mô hình [7] và ngôn ngữ hình thức [7]. Trong đó, ngôn ngữ mô hình là ngôn ngữ được sử dụng phổ biến nhất trong công nghiệp vì tính trực quan, dễ hiểu, dễ mô tả của nó cũng như vì sự phát triển của phương pháp phát triển hướng đối tượng từ năm 2000 cho đến nay.

Trong phát triển phần mềm hướng đối tượng, đặc tả yêu cầu bao gồm các biểu đồ ca sử dụng và các bản đặc tả chi tiết các ca sử dụng (use case specifications) [6]. Chúng là những chế tác có vai trò quan trọng thể hiện mong muốn của người dùng về hành vi bên ngoài của hệ thống. Cụ thể, chúng mô tả tương tác giữa người dùng và hệ thống gồm những sự kiện mà người dùng đưa vào hệ thống (input) và những kết quả mà hệ thống trả về cho người dùng (output). Những tương tác này được mô tả dưới dạng các đoạn văn bản và được tách thành các luồng gồm một luồng chính và một vài luồng phụ. Những sự kiện mà người dùng đưa vào hệ thống có thể là nhập dữ liệu dạng văn bản, bấm

chọn một phương án trong danh sách các lựa chọn mà hệ thống cung cấp, thao tác kéo thả, sự kiện nhấn nút bấm, sự kiện nhấn vào đường liên kết. Phản hồi của hệ thống có thể là một thông báo dạng văn bản, bảng dữ liệu cho người dùng xem hay một trang thông tin mà người dùng mong đợi.

Ca kiểm thử (test cases) là mô tả của dữ liệu đầu vào, điều kiện thực thi, quy trình thực nghiệm và kết quả mong đợi nhằm kiểm tra từng chức năng của hệ thống phần mềm hoạt động có đúng hay không. Việc tạo các bản mô tả ca kiểm thử thường tốn rất nhiều thời gian và công sức, cũng như dễ bỏ sót các ca kiểm thử nếu được thực hiện thủ công. Bản mẫu giao diện người dùng (UI prototypes) - công cụ giao tiếp giữa người dùng, nhà phân tích yêu cầu, nhà thiết kế - xác nhận yêu cầu người dùng, UI prototypes phải được tạo ra nhanh, sớm để có thể sử dụng để trao đổi và xác nhận về yêu cầu ở giai đoạn nắm bắt yêu cầu. Bản mẫu giao diện người dùng chứa các đối tượng trên giao diện để nhận dữ liệu (inputs) từ người dùng đưa vào và chứa các đối tượng mà hiển thị kết quả hệ thống xuất ra cho người dùng [7]. Mục đích chung để tạo ra các bản mẫu là: (i) để đảm bảo nhà thiết kế và nhà lập trình hiểu đúng yêu cầu [1]; (ii) để minh họa cho khách hàng/người dùng xác nhận yêu cầu [1]; (iii) hỗ trợ cho hoạt động kiểm thử. Một tiêu chí quan trọng trong việc tạo các bản mẫu là phải tốn ít thời gian và công sức để tiết kiệm chi phí vì nó có thể chỉ được sử dụng để trao đổi và xác nhận yêu cầu giữa người dùng, nhà thiết kế và nhà lập trình hoặc cũng có thể trở thành một phần của sản phẩm bàn giao cho khách hàng.

Việc sinh ca kiểm thử và bản mẫu giao diện phải làm đi làm lại nhiều lần, tốn nhiều thời gian, công sức, chi phí nếu làm bằng tay. Nếu có một sự thay đổi nhỏ trong yêu cầu phải xây dựng lại bộ kiểm thử và bản mẫu giao diện, thực hiện lại hoạt động kiểm thử với bộ kiểm thử mới. Tự động hóa bước sinh bộ kiểm thử và bản mẫu giao diện là thiết thực để giảm chi phí và tránh sự nhàm chán khi phải thực hiện lặp đi lặp lại bước này. Đầu vào để sinh ca kiểm thử có thể là tài liệu đặc tả yêu cầu, tài liệu thiết kế hoặc tài liệu mô tả chương trình. Trong bài này, chúng tôi sử dụng tài liệu đặc tả yêu cầu để sinh bộ kiểm thử và bản mẫu giao diện người dùng, bởi vì tài liệu yêu cầu thể hiện mong muốn của người dùng về phần mềm, nó là cơ sở để ký kết hợp đồng phát triển phần mềm, ngoài ra, nó được xây dựng từ các bước sớm của quy trình phát triển phần mềm. Do đó, chúng ta có thể sinh bộ kiểm thử ngay từ giai đoạn đầu của quy trình sau khi có tài liệu yêu cầu.

Các nghiên cứu liên quan đến việc sinh ca kiểm thử và bản mẫu giao diện với đầu vào là đặc tả yêu cầu tập trung theo một số hướng sau đây: (i) sinh dữ liệu kiểm thử cho các ca kiểm thử [4]; (ii) áp dụng các phương pháp xử lý ngôn ngữ tự nhiên để xử lý từ vựng, cấu trúc câu trong đặc tả ca sử dụng, từ đó, sinh tự động các ca kiểm thử [8,9]; (iii) sinh bản mẫu giao diện từ đặc tả ca sử dụng được mô tả theo form (semi-formal specification) dựa trên thư viện các mẫu sự kiện tương tác và thư viện các mẫu giao diện được định nghĩa sẵn [10]; (iv) sinh bản mẫu giao diện từ mô hình hình thức của yêu cầu bao gồm thể hiện hình thức của đặc tả ca sử dụng và mô hình miền [13]; (v) sinh ca kiểm thử từ đặc tả ca sử dụng được viết theo template của Cockburn [2] và tập trung vào tùy biến các đường đi để xác định các ca kiểm thử [12]; và (vi) sinh bản mẫu giao diện web từ mô hình yêu cầu bao gồm biểu đồ ca sử dụng, biểu đồ hoạt động, biểu đồ lớp, biểu đồ đối tượng [2,5,11]. Các nghiên cứu này sử dụng đầu vào khởi đầu là đặc tả hình thức để sinh bộ kiểm thử hoặc đầu vào là đặc tả yêu cầu được mô tả bằng ngôn ngữ tự nhiên hoặc ngôn ngữ mô hình như UML, rồi đặc tả này được chuyển sang đặc tả hình thức để làm đầu vào cho bước sinh bộ kiểm thử. Như vậy, các nghiên cứu hiện tại đều cần hình thức hóa yêu cầu. Các phương pháp này tốn chi phí, khó hiểu, khó áp dụng trong môi trường phát triển phần mềm công nghiệp - nơi mà đa phần các kỹ sư không thành thạo ngôn ngữ hình thức mà thường sử dụng ngôn ngữ tự nhiên và ngôn ngữ mô hình. Trong bài báo này, chúng tôi đề xuất phương pháp sinh tự động bộ kiểm thử, bản mô tả các ca kiểm thử và các bản mẫu giao diện người dùng từ đặc tả yêu cầu được mô tả bằng ngôn ngữ mô hình (UML). Phương pháp của chúng tôi tập trung vào đặc tả bằng ngôn ngữ mô hình để phù hợp với môi trường công

nghiệp và bổ sung thêm các chú thích (notation) - một cách đơn giản, dễ hiểu, cho cả người sử dụng cuối (end-users) và các kỹ sư phát triển phần mềm mà vẫn đem lại hiệu quả mong muốn. Các chú thích đều có khả năng mở rộng bởi người dùng nhờ đó có thể tùy biến theo từng ứng dụng hoặc cá nhân người phân tích yêu cầu. Chúng tôi sử dụng tài liệu yêu cầu được mô tả bằng ngôn ngữ mô hình để sinh bộ kiểm thử và bản mẫu giao diện, những việc mà thực tế đòi hỏi phải thực hiện lặp đi lặp lại trong quy trình phát triển phần mềm.

Phần còn lại của bài báo được tổ chức như sau. Mục 2 trình bày tổng quan về phương pháp sinh bộ kiểm thử và bản mẫu giao diện từ đặc tả ca sử dụng. Mục 3 và 4 mô tả chi tiết hai bước chính của quy trình là bước thêm chú thích cho đặc tả và bước sinh các kết quả đầu ra. Mục 5 chúng tôi đưa ra một số ý kiến thảo luận và kết luận.

2. Quy trình sinh bộ kiểm thử và các bản mẫu giao diện

Chúng tôi sẽ giảng giải cách tiếp cận của chúng tôi để sinh bộ kiểm thử và các bản mẫu giao diện người dùng qua mô tả một ví dụ đơn giản. Đầu vào là các bản đặc tả ca sử dụng - chế tác của giai đoạn đặc tả yêu cầu. Đầu ra là bộ kiểm thử, gồm các ca kiểm thử đi kèm các bản mô tả ca kiểm thử, và các bản mẫu giao diện người dùng. Giả sử chúng ta có đặc tả cho ca sử dụng Đăng nhập (Login) của một hệ thống bán hàng như được mô tả trong Bảng 1. Chúng ta cần sinh bản mẫu giao diện người dùng (được minh họa trong Hình 1) từ đặc tả này để sử dụng cho việc giao tiếp giữa nhà phân tích yêu cầu, người dùng và nhà thiết kế đảm bảo cho các bên liên quan hiểu đúng và xác nhận yêu cầu. Chúng ta cũng sinh tất cả các ca kiểm thử điển hình tương ứng với tất cả các kịch bản hành vi điển hình từ đặc tả ca sử dụng; sau đó, chúng ta sinh các bản mô tả cho các ca kiểm thử tương ứng (được minh họa trong Hình 2), giúp tiết kiệm thời gian, công sức và đảm bảo chất lượng của các ca kiểm thử vì chúng khớp với yêu cầu được mô tả.

Trong Bảng 1, người dùng đưa vào hệ thống dữ liệu dạng văn bản gồm có tên người dùng và mật khẩu. Đồng thời, người dùng nhấn một nút bấm để yêu cầu hệ thống thực hiện đăng nhập. Hệ thống phản hồi bằng một thông báo dạng văn bản cho người dùng biết kết quả đăng nhập.

Trong đặc tả ca sử dụng như chúng ta thấy ở Hình 1 có 3 luồng sự kiện vào ra giữa tác nhân và hệ thống, mỗi luồng đại diện cho một kịch bản hành vi điển hình. Tập hợp tất cả các luồng sự kiện được mô tả trong đặc tả ca sử dụng sẽ là nguồn để sinh ra tất cả các ca kiểm thử điển hình và các bản mô tả các ca kiểm thử này. Sau đây, chúng tôi gọi chung các tập hợp các ca kiểm thử và tập hợp các bản mô tả các ca kiểm thử là bộ ca kiểm thử cho ngắn gọn.

Các sự kiện mà tác nhân đưa vào hệ thống được mô tả trong mỗi luồng là inputs mà người dùng đưa vào hệ thống dưới các dạng khác nhau như là text input, selection input, button click input, file input. Cũng vậy, phản hồi của hệ thống cho tác nhân chính là outputs mà hệ thống hiển thị cho người dùng thấy dưới các dạng như text output, alert box,... Các inputs, outputs này là các ứng cử viên phù hợp để nhận ra các phần tử trên giao diện người dùng - nơi để người dùng đưa dữ liệu vào và nơi để hệ thống xuất kết quả trả về cho người dùng.

Bảng 1. Đặc tả ca sử dụng Login (đầu vào của quy trình)

Luồng chính:	1. Người dùng nhập vào tên người dùng và mật khẩu 2. Người dùng yêu cầu hệ thống đăng nhập với tên người dùng và mật khẩu 3. [If valid password then] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thành công
Luồng phụ 1:	3'. [If invalid password then] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thất bại do sai mật khẩu
Luồng phụ 2:	3''. [If empty password then] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thất bại do chưa nhập mật khẩu

Hình 1: Bản mẫu giao diện cho ca sử dụng Login (đầu ra của quy trình)

Quy trình thực hiện sinh bộ kiểm thử và các bản mẫu giao diện người dùng từ các bản đặc tả ca sử dụng gồm các bước: (1) Thêm chú thích cho đặc tả ca sử dụng; (2) Sinh đồ thị luồng; (3) Sinh các ca kiểm thử và sinh bản mô tả các ca kiểm thử; (4) Sinh bản mẫu giao diện người dùng. (Hình 2)

Để tạo nhãn cho dữ liệu vào và dữ liệu ra xuất hiện trên giao diện người dùng cũng như nhãn cho dữ liệu vào cần nhập trong bản mô tả kiểm thử (như được minh họa trong Hình 1, Hình 2), chúng ta cần bổ sung chú thích là tên của dữ liệu, như trong Bảng 4.

Nhà phân tích yêu cầu và người dùng sẽ thống nhất đặt tên cho dữ liệu vào và ra, do đó nhãn tương ứng xuất hiện trong bản mô tả kiểm thử và nhãn xuất hiện trên giao diện người dùng là tùy vào mong muốn của người dùng, thường là ngắn gọn và khớp với tên của dữ liệu, ngôn ngữ là tùy theo yêu cầu từ người dùng.

Kết quả sau khi thêm các chú thích cần thiết cho đặc tả ca sử dụng Login được minh họa trong Hình 3.

Hình 2: Mô tả các ca kiểm thử (đầu ra của quy trình)

	A	B	C	D	E	F	G	H
		STT	input element	input label	input data	expected output element	expected output label	expected output state
0		1	text input	user-name		text output	pw	valid
1			text input	password				
2			button onclick input	login				
3		2	text input	user-name		text output	pw	invalid
4			text input	password				
5			button onclick input	login				
6		3	text input	user-name		text output	pw	empty
7			text input	password				
8			button onclick input	login				

Sheet1

3. Thêm chú thích cho đặc tả yêu cầu

Thông tin chính trên bản mẫu giao diện là các loại dữ liệu mà tác nhân đưa vào và các loại dữ liệu mà hệ thống xuất cho tác nhân. Để sinh tự động các bản mẫu giao diện, công cụ cần nhận ra các loại dữ liệu vào và ra này từ bản đặc tả ca sử dụng tương ứng với các sự kiện mà tác nhân đưa vào và các kết quả mà hệ thống xuất ra cho tác nhân. Như được nhắc đến ở trên, các luồng chính và luồng phụ của ca sử dụng đều được mô tả bằng văn bản (ngôn ngữ tự nhiên). Để công cụ có thể nhận ra mỗi loại dữ liệu vào và ra này, chúng ta cần bổ sung các chú thích cho mỗi loại dữ liệu được mô tả trong đặc tả ca sử dụng. Bảng 2 dưới đây liệt kê mỗi loại dữ liệu đầu vào và chú thích tương ứng cho chúng.

Để phân biệt luồng sự kiện chính và luồng sự kiện phụ, chúng ta bổ sung chú thích tại mỗi điểm rẽ nhánh của luồng, như trong Bảng 3.

Đặc tả ca sử dụng sau khi được thêm các chú thích như trên cũng rất có ích cho việc trao đổi để xác nhận yêu cầu giữa bên phát triển và khách hàng trước khi sử dụng chúng như là nguồn tin cậy để sinh bộ kiểm thử và bản mẫu giao diện.

4. Sinh bộ kiểm thử và các bản mẫu giao diện

Để sinh bộ kiểm thử từ đặc tả yêu cầu, chúng tôi phát triển một công cụ thực hiện tự động các bước (2) - sinh đồ thị luồng, (3) - sinh bộ kiểm thử và (4) - sinh các giao diện người dùng. Công cụ được cài đặt bằng ngôn ngữ Python [3]. Đầu vào là đặc tả ca sử dụng được lưu trong tệp văn bản thông thường (.txt). Đầu ra ở mỗi bước sẽ được mô tả sau đây:

4.1. Sinh đồ thị luồng từ đặc tả ca sử dụng đã được thêm chú thích

Đồ thị luồng là đồ thị có hướng gồm một tập các đỉnh và các đường nối có hướng giữa các đỉnh. Trong số các đỉnh có 1 đỉnh đầu tương ứng với sự

Bảng 2. Các chú thích được thêm vào đặc tả ca sử dụng để nhận ra dữ liệu vào ra

STT	Dữ liệu vào	Chú thích	Ý nghĩa
1	Nhập vào đoạn văn bản	#ti (text input)	Loại dữ liệu: text Hành động: nhập vào
2	Lựa chọn phương án	#si (selection input)	Loại dữ liệu: lựa chọn Hành động: nhập vào
3	Nhập vào một tệp	#fi (file input)	Loại dữ liệu: file Hành động: nhập vào
4	Bấm vào nút bấm	#bi (button onclick input)	Loại dữ liệu: nút bấm Hành động: nhập vào
5	Bấm vào liên kết	#li (link onclick input)	Loại dữ liệu: liên kết Hành động: nhập vào
6	Hiển thị kết quả dạng văn bản	#to (text output)	Loại dữ liệu: text Hành động: xuất ra
7	Hiển thị hộp thoại thông báo	#ao (Alert box output)	Loại dữ liệu: hộp thoại Hành động: xuất ra
8	Hiển thị một giao diện khác	#uio	Loại dữ liệu: giao diện Hành động: xuất ra

Bảng 3. Các chú thích được thêm vào đặc tả ca sử dụng để nhận ra các luồng sự kiện

1	Text input is invalid	#invalid #ti
2	Text input is empty	#empty #ti

Bảng 4. Các chú thích được thêm vào đặc tả ca sử dụng để tạo nhãn

STT	Dữ liệu	Nhãn	Chú thích
1	Tên người dùng	User name	#user-name
2	Mật khẩu	Password	#password

Hình 3: Đặc tả ca sử dụng Login sau khi được thêm các chú thích cần thiết

```
Use case Name: Login
Behavior Scenarios:
Basic flow:
1. Người dùng nhập vào tên người dùng [#ti][#user-name] và mật khẩu [#ti][#password]
2. Người dùng yêu cầu hệ thống đăng nhập với tên người dùng và mật khẩu đã nhập [#bi][#login].
3. [#valid] [#pw] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thành công [#to]

Alternate flow 1:
1. Người dùng nhập vào tên người dùng (#ti) và mật khẩu (#ti)
2. Người dùng yêu cầu hệ thống đăng nhập với tên người dùng và mật khẩu đã nhập (#bi).
3'. [#invalid] [#pw] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thất bại do sai mật khẩu [#to]

Alternate flow 2:
1. Người dùng nhập vào tên người dùng (#ti) và mật khẩu (#ti)
2. Người dùng yêu cầu hệ thống đăng nhập với tên người dùng và mật khẩu đã nhập (#bi).
3''. [#empty] [#pw] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thất bại do chưa nhập mật khẩu [#to]
```

kiện bắt đầu ca sử dụng và có một hoặc một vài đỉnh kết thúc tương ứng với các sự kiện kết thúc ca sử dụng. Đồ thị có thể có các đỉnh điều kiện tương ứng với điều kiện rẽ nhánh ở ca sử dụng. Hình 4 là một thể hiện đồ họa của đồ thị luồng gồm có các đỉnh *1, 2, 3, 4, 5*, đỉnh rẽ nhánh C và các đường mũi

tên nối các đỉnh. Trong đó đỉnh số 1 là đỉnh đầu, đỉnh số *3, 4, 5* là các đỉnh kết thúc.

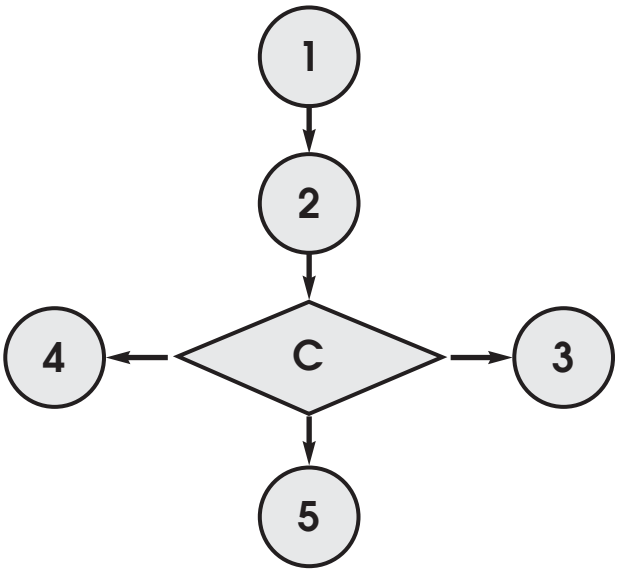
Mỗi luồng sự kiện chính hoặc phụ trong đặc tả ca sử dụng bao gồm một số bước tương tác giữa tác nhân và hệ thống để hoàn thành hành vi của ca sử dụng. Mỗi luồng này là một kịch bản hành vi gồm nhiều bước. Mỗi bước có thể được thể hiện bằng một nút trong đồ thị luồng. Đường nối các bước trong cùng một luồng sự kiện chính hoặc phụ thể hiện một đường đi trong đồ thị luồng. Hình 4 là thể hiện dưới dạng đồ họa của đồ thị luồng được sinh ra từ đặc tả ca sử dụng Login. Trong đó, nút số 1, 2, 3, 4, 5 tương ứng với các sự kiện 1, 2, 3, 3, 3 trong đặc tả ca sử dụng.

4.2. Sinh các ca kiểm thử và mô tả các ca kiểm thử

Công cụ sinh các ca kiểm thử bằng cách tìm tất cả các đường đi trong đồ thị luồng. Mỗi đường đi được bắt đầu từ đỉnh đầu đến một trong các đỉnh kết thúc. Mỗi đường đi là một trường hợp điển hình cần kiểm thử. Vậy tập hợp tất cả các đường đi được nhận ra từ đồ thị sẽ là tập hợp các trường hợp cần kiểm thử.

Với đồ thị ở Hình 4, chúng ta có thể nhận ra các đường đi như sau: 1-2-3; 1-2-4; và 1-2-5. Như vậy, có 3 ca kiểm thử được sinh ra trong trường hợp này cùng với các bản mô tả tương ứng cho mỗi ca kiểm thử. Hình 5 minh họa các ca kiểm thử được sinh ra, mỗi dòng thể hiện một ca kiểm thử tương ứng với 1 đường đi trong đồ thị luồng - được lưu dưới dạng tệp văn bản (.txt). Hình 6 minh họa bản mô tả các ca kiểm thử - được lưu dưới dạng tệp bảng tính của MS.Excel (.xlsx) - với khuôn dạng gồm các nhãn cho dữ liệu đầu vào và đầu ra, các ô nhập dữ liệu vào và ra cho mỗi ca kiểm thử đang để trống để người thiết kế kiểm thử nhập vào.

Hình 4: Đồ thị luồng được sinh từ đặc tả yêu cầu đã bổ sung chú thích



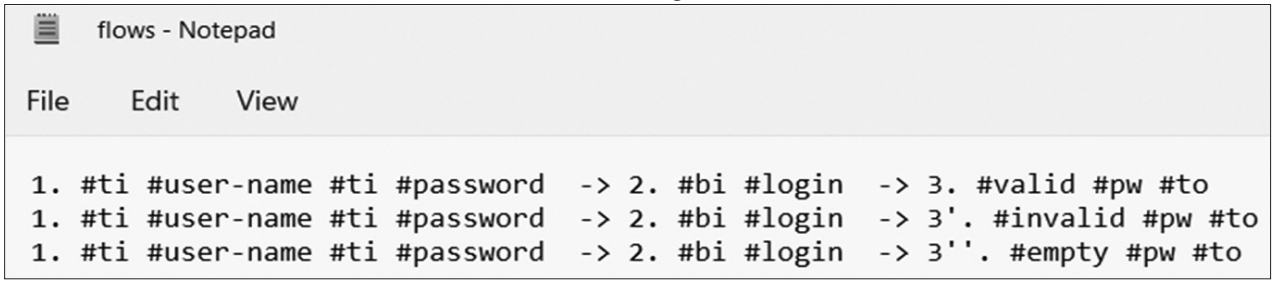
Các bản mô tả kiểm thử còn thiếu dữ liệu kiểm thử cần được bổ sung theo gợi ý của từng trường hợp kiểm thử điển hình.

4.3. Sinh bản mẫu giao diện người dùng từ đặc tả ca sử dụng đã được thêm chú thích

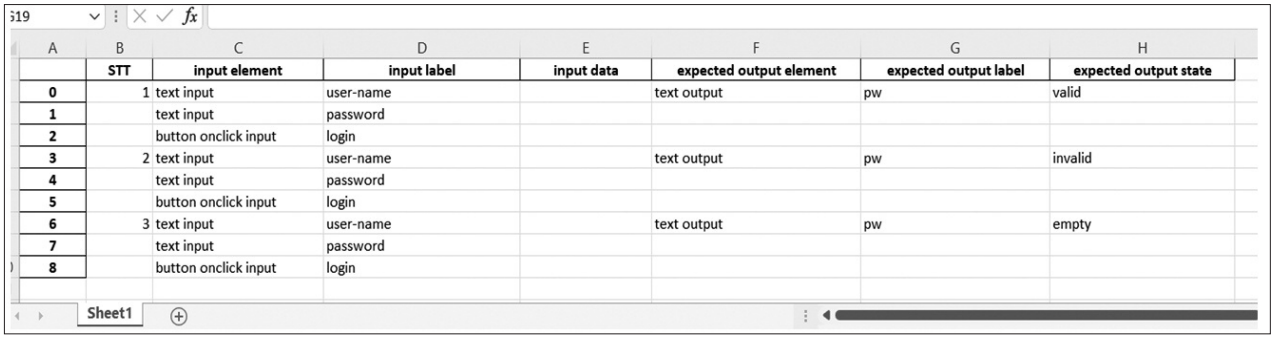
Trong bài này chúng tôi minh họa bằng giao diện được sinh trên web, do đó, cần sinh mã nguồn được viết bằng ngôn ngữ HTML để khai báo ra các đối tượng trên giao diện bao gồm cả các đối tượng nhận dữ liệu từ người dùng và các đối tượng xuất kết quả cho người dùng, nói cách khác, gồm cả inputs và outputs của hệ thống.

HTML cung cấp các thẻ để khai báo ra các đối tượng trên giao diện trang web, trong đó có thẻ đôi (gồm thẻ mở và thẻ đóng) và thẻ đơn (chỉ có 1 thẻ duy nhất). Chúng tôi sử dụng bảng ánh xạ từ các chú thích có thể được nhận ra từ đặc tả ca sử dụng tới các thẻ HTML tương ứng như được minh họa

Hình 5: Các ca kiểm thử được sinh ra từ đồ thị luồng

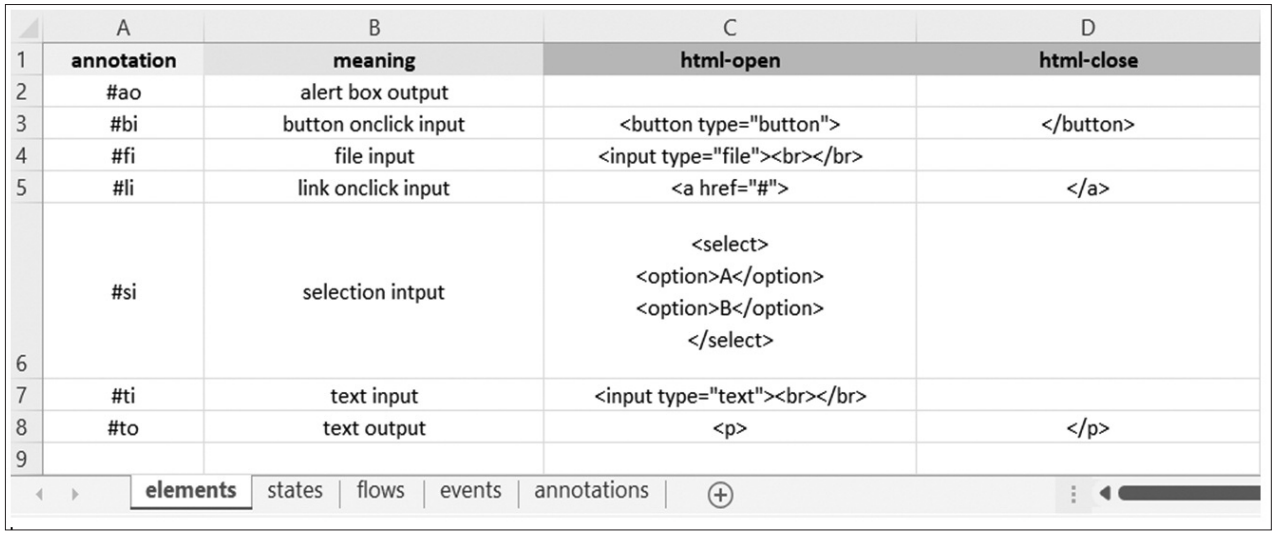


Hình 6: Các bản mô tả ca kiểm thử được sinh từ đặc tả yêu cầu và đồ thị luồng



	STT	input element	input label	input data	expected output element	expected output label	expected output state
0	1	text input	user-name		text output	pw	valid
1		text input	password				
2		button onclick input	login				
3	2	text input	user-name		text output	pw	invalid
4		text input	password				
5		button onclick input	login				
6	3	text input	user-name		text output	pw	empty
7		text input	password				
8		button onclick input	login				

Hình 7: Bảng ánh xạ được sử dụng để sinh bản mẫu giao diện



	annotation	meaning	html-open	html-close
2	#ao	alert box output		
3	#bi	button onclick input	<button type="button">	</button>
4	#fi	file input	<input type="file"> </br>	
5	#li	link onclick input		
6	#si	selection intput	<select> <option>A</option> <option>B</option> </select>	
7	#ti	text input	<input type="text"> </br>	
8	#to	text output		
9				

elements | states | flows | events | annotations

trong Hình 7 cho việc sinh ra bản mẫu giao diện người dùng từ đặc tả ca sử dụng tương ứng.

Đầu ra của bước này là các tệp mã nguồn html, trong đó có các thẻ html được định nghĩa để khai báo các phần tử trên giao diện của người dùng, bao gồm cả các phần tử nhận dữ liệu vào từ người dùng và các phần tử xuất dữ liệu ra cho người dùng.

4.4. Đánh giá và thảo luận

Các bước được thực hiện bởi công cụ cho hiệu quả cao nhờ vào tốc độ thực hiện nhanh, quy trình tự động khép kín. Khuôn dạng của dữ liệu đầu vào, đầu ra có thể linh hoạt điều chỉnh theo yêu cầu của người sử dụng. Do đó, quy trình dễ được chấp nhận từ người dùng.

5. Kết luận

Trong bài báo này, chúng tôi đề xuất phương pháp sinh tự động bộ kiểm thử, bản mô tả các ca kiểm thử và các bản mẫu giao diện người dùng từ đặc tả yêu cầu được mô tả bằng ngôn ngữ mô hình (UML). Đặc tả yêu cầu (ở đây là use case specifications) là nguồn tin cậy để sinh bộ kiểm thử, bản mô tả các ca kiểm thử và bản mẫu giao diện người dùng. Nhờ đó, chúng ta có thể tiết kiệm được chi phí tạo ra các chế tác này và hơn hết chúng ta có thể đảm bảo các ca kiểm thử và các giao diện khớp với yêu cầu của người dùng. Thêm vào đó, việc thay đổi yêu cầu là không thể tránh khỏi ở bất kỳ dự án phát triển phần mềm nào và sự thay đổi này có thể diễn ra nhiều lần. Mỗi khi có yêu cầu thay đổi, sinh tự động bộ kiểm thử và bản mẫu giao diện từ bản mô tả các yêu cầu mới sẽ giúp giảm đáng kể chi phí cho sự thay đổi và chất lượng sản phẩm luôn được đảm bảo vì bộ kiểm thử và bản mẫu giao diện được sinh ra nhất quán với mong muốn của người dùng.

Đã có rất nhiều nghiên cứu liên quan chủ đề

này, mỗi nghiên cứu tập trung một khía cạnh của tự động hóa kiểm thử, ví dụ, tùy biến các đường kiểm thử hay sinh dữ liệu kiểm thử. Trong bài này, chúng tôi tập trung vào độ phủ (coverage) của bộ kiểm thử và tính thực hành cao của cách tiếp cận. Công cụ sinh tự động các ca kiểm thử dưới dạng tệp MS Excel để hỗ trợ một phần của bước mô tả ca kiểm thử, người kiểm thử chỉ cần thêm dữ liệu vào nữa là đầy đủ và sinh tự động UI prototype dưới dạng các tệp html để hỗ trợ bước xác nhận yêu cầu với người dùng và hỗ trợ một phần cho bước thiết kế giao diện.

Phương pháp của chúng tôi tập trung vào đặc tả bằng ngôn ngữ mô hình để phù hợp với môi trường công nghiệp và bổ sung thêm các chú thích - một cách đơn giản, dễ hiểu cho cả người sử dụng cuối và các kỹ sư phát triển phần mềm. Các chú thích và bảng ánh xạ đều có khả năng mở rộng bởi người dùng và phù hợp với môi trường phát triển phần mềm công nghiệp - nơi mà hầu hết các kỹ sư và khách hàng thường sử dụng ngôn ngữ mô hình và các bản mẫu giao diện đồ họa làm công cụ giao tiếp để nắm bắt, phân tích và xác nhận yêu cầu cho phần mềm.

Hướng phát triển của đề tài là mở rộng các chú thích để diễn tả được đa số các trường hợp (ví dụ: điền dữ liệu kiểm thử) thông qua việc áp dụng vào các bài toán phân tích thực tế để tìm ra bộ chú thích chuẩn và tổng quát cho đa số các trường hợp ■

TÀI LIỆU THAM KHẢO:

1. A. O. J. Sabriye, W. M. N. Zainon (2017). "A Framework for Detecting Ambiguity in Software Requirement Specification" in 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, 2017, 209-2013.

2. A. Cockburn, (1999). Structuring use cases with goals. *Journal of Object-Oriented Programming,* vol. 35, 40, 56-62.

3. Nexcod Publishing (2019). *Python Programming: The Beginners Guide, Independently published,* United State, Traverse City.

4. C. Wang, F. Pastore, A. Goknil et al. (2015). *Automatic generation of system test cases from use case specifications.* Proceedings of the 2015 International Symposium on Software Testing and Analysis.

5. E V. Sunitha (2021). *Improving Requirement Specification by Prototype Generation from Requirement Models.* ICCIDT - 2021 Conference Proceedings.

6. G. Booch,et al. (2007). *Object-Oriented Analysis and Design with Applications3rd Edition,* Addison-Wesley, Boston.

7. I. Sommerville (2011). *Software Engineering 9th Edition.* Addison-Wesley, Boston.

8. J. H. Sneed,. (2007). Testing against natural language requirements, in IEEE Quality Software, 2007. QSIC'07. *Seventh International Conference on,* 380-387.

9. Marko (2020). Automatic generation of test cases from use-case specification using natural language processing. 33rd bled econference enabling technology for a sustainable society.

10. M. Kamalrudin, J. C. Grundy (2011). Generating essential user interface prototypes to validate requirements. 2011 26th IEEE/ACM International Conference on Automated Software Engineering.

11. S. Ogata, S. Matsuura (2010). Evaluation of a use-case-driven requirements analysis tool employing web UI prototype generation. *WSEAS Trans. Info. Sci. and App.,* 7(2), 273-282.

12. T. A. Alrawashed, A. Almomani, A. Althunibat, (2019). An Automated Approach to Generate Test Cases From Use Case Description Model. *CMES-Computer Modeling in Engineering & Sciences,* 119(3), 409–425.

13. Y. Yang, X. Li, W. Ke et al (2020). Automated Prototype Generation from Formal Requirements Model. *IEEE Trans. Reliab*. 69(2):632-656.

Ngày nhận bài: 5/2/2023
Ngày phản biện đánh giá và sửa chữa: 5/3/2023
Ngày chấp nhận đăng bài: 18/3/2023

Thông tin tác giả:
1. VŨ DIỆU HƯƠNG
2. BÙI QUANG TRƯỜNG
3. TRẦN THỊ NHUNG
Khoa Hệ thống thông tin Kinh tế và Thương mại điện tử
Trường Đại học Thương mại

ADDING NOTATION TO UML USE CASE SPECIFICATION TO FACILITATE THE GENERATION OF TEST CASES AND UI PROTOTYPES

● VU DIEU HUONG[1]
● BUI QUANG TRUONG[1]
● TRAN THI NHUNG[1]
[1]Faculty of Economic Information Systems and E-commerce,
ThuongMai University

ABSTRACT:

Generating test cases and user interface (UI) prototypes should be done in the beginning phase of software development processes because they are useful to validate and confirm software requirements between the developer and the client. These activities are often repeated many times whenever there is a change in software requirements. If these activities are done manually, it is very difficult for the developer to create a large number of test cases and make prototypes quickly at low costs. Automating test suite generation and UI prototyping generation is a solution for this problem. This paper proposes an approach to add notations to the UML Use Case specification to automatically generate test cases and UI prototypes. This approach is highly practical and effective in the software development process of companies.

Keywords: test case generation, UI prototype generation, requirement specification, UML language.