

490 Phần B Quy trình làm việc của Vòng đời Phần mềm

kĩ năng. Bởi vì hầu hết mọi kỹ thuật thiết kế cho các hệ thống thời gian thực đều tốt hơn là không có kỹ thuật nào, nên một số kỹ thuật thiết kế thời gian thực được sử dụng trong thực tế. Tuy nhiên, vẫn còn một quãng đường dài trước khi có thể thiết kế các hệ thống thời gian thực như những hệ thống đã được mô tả trước đó và chắc chắn rằng, trước khi hệ thống được triển khai, mọi ràng buộc thời gian thực sẽ câu trả lời được giải đáp và các vấn đề đồng bộ hóa sẽ không thể xảy ra. .

Các kỹ thuật thiết kế thời gian thực cũ hơn là phần mở rộng của các kỹ thuật phi thời gian thực đối với miền thời gian thực. Ví dụ: sự phát triển có cấu trúc cho các hệ thống thời gian thực (SDRTS)

[Ward và Mellor, 1985] về cơ bản là phần mở rộng của hệ thống phân tích có cấu trúc (Phần 12.3), phân tích luồng dữ liệu (Phần 14.3) và phân tích giao dịch (Phần 14.4) cho phần mềm thời gian thực. Bao gồm kỹ thuật phát triển kỹ thuật bao gồm một phần thành phần cho thời gian thực của thiết kế. Các kỹ thuật mới hơn được mô tả trong [Liu, 2000] và [Gomaa, 2000].

Như đã nêu trước đây, sự thật có thể không phải là trình độ kỹ thuật của thiết kế thời gian kế tiếp không phải là tiến trình đầu tiên như người ta mong muốn. Tuy nhiên, các nỗ lực đang được tiến hành để cải thiện tình hình.

14.14 CASE Tool for design

Như đã nêu trong Phần 14.10, một khía cạnh quan trọng của thiết kế là kiểm tra xem các tác nhân tạo thiết kế có kết hợp chính xác tất cả các khía cạnh của phân tích hay không. Do đó, điều cần thiết là một công cụ CASE có thể được sử dụng cho cả các công cụ tạo phân tích và tạo sản phẩm thiết kế, cái được gọi là công cụ front-end hoặc upperCASE (ngược với công cụ back-end hoặc back-end hoặc upperCASE). LowerCASE, support khai thác các sản phẩm tạo ra).

Một số công cụ chữ hoa có trên thị trường. Một số công việc phổ biến hơn bao gồm Nhà phân tích/ Nhà thiết kế, Phần mềm thông qua Hình ảnh và Kiến trúc sư Hệ thống. Công cụ UpperCASE thường được xây dựng xung quanh một từ điển dữ liệu. Công cụ CASE có thể kiểm tra xem mọi trường của mọi bản ghi trong từ điển có được đề cập ở đâu trong thiết kế hay mọi mục trong thiết kế đều bị phản ánh trong sơ đồ luồng dữ liệu. Ngoài ra, nhiều công cụ viết hoa kết hợp một trình kiểm tra tính quán nhất sử dụng từ điển dữ liệu để xác minh rằng mọi mục trong thiết kế đã được khai báo trong thông số kỹ thuật và đảo ngược rằng mọi mục trong thông số kỹ thuật đều xuất hiện trong design.

Hơn nữa, nhiều công cụ viết hoa kết hợp trình tạo màn hình và báo cáo. Nghĩa là, khách hàng có thể chỉ định bất kỳ mục nào sẽ xuất hiện trong báo cáo hoặc trên màn hình nhập liệu cũ ng như vị trí và cách thức xuất hiện của từng mục. Do chi tiết đầy đủ về mọi mục đều có trong từ điển dữ liệu nên công cụ CASE có thể dễ dàng tạo mã để đưa vào báo cáo hoặc hiển thị màn hình dữ liệu theo ý muốn của khách hàng. Một số sản phẩm chữ hoa cũ ng kết hợp các công cụ quản lý tỷ lệ hợp lệ để ước tính và lập kế hoạch kế hoạch.

Đối với thiết kế hướng đối tượng, IBM Rational Rose và Phần mềm thông qua Hình ảnh giống nhau cung cấp hỗ trợ cho quy trình công việc này trong bối cảnh của vòng đời hướng đối tượng hoàn chỉnh. Các công cụ CASE mã nguồn mở thuộc loại này bao gồm ArgoUML.

14.15 Number for design

Một loạt các số liệu có thể được sử dụng để mô tả các khía cạnh của thiết kế. Ví dụ: số lượng tạo mã sản phẩm (mô-đun hoặc lớp) là thước đo thô về kích thước của sản phẩm mục tiêu.

Sự gắn kết và khớp nối là thước đo chất lượng của thiết kế, cũ ng như lỗi thống kê.

Như với tất cả các loại kiểm tra khác, điều quan trọng là phải lưu giữ hồ sơ về số lượng và loại

lỗi thiết kế được phát hiện trong quá trình kiểm tra thiết kế. Thông báo này được sử dụng trong quá trình kiểm tra mã sản phẩm và kiểm tra thiết kế của các sản phẩm tiếp theo.

Độ phức tạp của chu trình M của một thiết kế chi tiết là số quyết định nhị phân (vị trí từ) cộng với 1 [McCabe, 1976] hoặc tương đương, số nhánh trong tác vụ tạo mã. Có ý kiến cho rằng độ phức tạp của chu trình là thước đo chất lượng thiết kế; giá trị của M càng thấp càng tốt. Một điểm mạnh của số liệu này là nó rất dễ tính toán của độ phức tạp của một vấn đề hữu ích. Độ phức tạp theo chu kỳ hoàn toàn là thước đo được mức độ phức tạp của một mã sản phẩm tạo được điều khiển bởi dữ liệu, thực tế là các giới hạn bởi các giá trị trong một bảng. Ví dụ: giả sử một nhà thiết kế không biết về hàm thư viện C++ toascii và thiết kế tạo mã từ đầu để đọc một ký tự do người dùng nhập và trả về mã ASCII tương ứng (một số nguyên từ 0 đến 127). Một cách để thiết kế điều này là sử dụng nhánh 128 chiều được khai thác bằng câu lệnh chuyển đổi. Cách thứ hai là có một mảng chứa 128 ký tự theo thứ tự mã ASCII và sử dụng một vòng lặp để so sánh các ký tự do người dùng nhập với từng phần tử của ký tự mảng; vòng lặp bị thoát khi thu được kết quả khớp. Giá trị hiện tại của biến vòng lặp khi đó là mã ASCII tương ứng. Hai thiết kế tương đương về chức năng nhưng có độ phức tạp theo chu kỳ như 128 và 1.

Khi mô hình cổ điển được sử dụng, một lớp đo lường liên quan cho giai đoạn thiết kế dựa trên công việc biểu diễn thiết kế kiến trúc dưới dạng biểu đồ có hướng với các mô-đun được biểu thị bằng các nút và luồng giữa các nút mô-đun (các lệnh gọi thủ tục và chức năng) được biểu thị bằng các cung cấp. Fan **-in** của một mô-đun có thể được định nghĩa là lượng luồng vào mô-đun cộng với lượng cấu trúc dữ liệu toàn cầu được truy cập bởi mô-đun. Tương tự như vậy, fan **-out** là lượng luồng ra khỏi mô-đun cộng với lượng cấu trúc dữ liệu toàn cầu được mô-đun cập nhật cập nhật. Khi đó, kích thước đo độ phức tạp của mô-đun được đưa ra bằng chiều dài \times (quạt vào \times quạt ra)

² [Henry và Kafura, 1981], in that the length is the dimensions size of the module (Phần 9.2.1). Do các định nghĩa về phân phối được đưa vào và xuất ra kết quả toàn bộ dữ liệu nên số liệu này có các phần phụ thuộc vào dữ liệu. Tuy nhiên, các thí nghiệm đã chỉ ra rằng thước đo này không phải là thước đo mức độ phức tạp tốt hơn so với thước đo đơn giản hơn, chẳng hạn như mức độ phức tạp chu trình [Kitchenham, Pickard và Linkman, 1990; Shepperd, 1990].

Các vấn đề về mức độ thiết kế thậm chí còn phức tạp hơn khi mô hình định hướng đối tượng được sử dụng. Ví dụ, độ phức tạp của chu trình của một lớp thường thấp, bởi vì nhiều lớp thường bao gồm một số lượng lớn các phương thức nhỏ, đơn giản. Hơn nữa, như đã chỉ ra trước đó, độ phức tạp theo chu trình bỏ qua độ phức tạp của dữ liệu. Bởi vì dữ liệu và hoạt động là các đối tượng bình đẳng trong mô hình định hướng đối tượng, nên độ phức tạp theo chu kỳ bỏ qua một thành phần chính có thể đóng góp phần tạo nên độ phức tạp của một đối tượng. Do đó, các số liệu cho các lớp kết hợp phức tạp của chu trình thường ít được sử dụng.

Ví dụ, một số lượng đo thiết kế hướng đối tượng đã được đưa ra trong [Chidamber và Kemerer, 1994]. Những số liệu này và những số liệu khác đã được đặt câu hỏi trên cả cơ sở lý thuyết và thực tế trải nghiệm [Binkley và Schach, 1996; 1997; 1998].

14.16 The thể thức của quy trình thiết kế

Như đã chỉ ra trong Phần 12.16 và 13.22, điều quan trọng là không làm quá nhiều trong quy trình phân tích; nghĩa là, nhóm phân tích không được bắt đầu sớm các phần của quy trình thiết kế. Trong quy trình thiết kế, nhóm thiết kế có thể mắc sai lầm theo hai cách: làm quá nhiều và làm quá ít.

Xem xét thiết kế chi tiết PDL (mã giả) của Hình 14.7 . Sự cám dỗ mạnh mẽ chống lại một nhà thiết kế thích lập trình để viết thiết kế chi tiết bằng C ++ hoặc Java, thay vì PDL. Nghĩa là, thay vì phác thảo thiết kế chi tiết bằng mã giả, người thiết kế có thể hoàn thành toàn bộ việc viết mã cho lớp. Điều này mất nhiều thời gian để viết hơn là chỉ thảo luận lớp và lâu hơn để chỉnh sửa x nếu một lỗi được phát hiện trong thiết kế (xem Hình 1.6). Giống như nhóm phân tích, các thành viên của nhóm thiết kế phải kiên quyết chống lại sự thôi thúc phải làm nhiều hơn những gì được yêu cầu.

Đồng thời, nhóm thiết kế phải cẩn thận để không làm quá ít. Xem xét bảng thiết kế chi tiết dạng bảng của Nếu Hình 14.6.

nhóm thiết kế đang hoạt động, họ có thể quyết định thu nhỏ thiết bị kế chi tiết thành hộp tường thuật. Nhóm thậm chí có thể quyết định rằng những người lập trình phải tự mình thiết kế chi tiết. Một trong những quyết định này sẽ là một sai lầm. Lý do chính cho thiết kế chi tiết là để đảm bảo rằng tất cả các giao diện đều chính xác.

Bản thân hộp tường thuật là không đủ cho mục đích này; không có thiết kế chi tiết rõ ràng thậm chí còn ít hữu ích hơn. Do đó, một chế độ của quy trình thiết kế là các nhà thiết kế chỉ thực hiện đúng khối work volume.

Ngoài ra, có một số thứ không thể quan trọng hơn nhiều. Trong “No Silver Bullet” (xem Just in Case You Wanted to Know Box 3.4), Brooks [1986] chỉ trích dẫn sự thiếu sót của cái mà ông gọi là những nhà thiết kế kế công so với những nhà thiết kế phần mềm. Brooks nói rằng các nhà thiết kế phần mềm giỏi nhất là những người được dạy; thiết kế tuyệt vời chỉ được tạo ra bởi những nhà thiết kế tuyệt vời và họ “cực kỳ” .

Sau đó, tri thức là sự phát triển của các nhà thiết kế tuyệt vời. Họ nên được xác định càng sớm càng tốt (những nhà thiết kế giỏi nhất không nhất thiết phải là người có kinh nghiệm nhất), chỉ xác định một người cố vấn, cung cấp nền tảng giáo dục chính thức cũ ng như học nghề cho các nhà thiết kế giỏi và được phép tương tác với các nhà thiết kế khác. Một con đường sự nghiệp cụ thể nên có sẵn cho những nhà thiết kế này và phần thưởng mà họ nhận được phải tương xứng với sự đóng góp mà chỉ một nhà thiết kế vĩ đại mới có thể thực hiện cho một dự án phát triển phần mềm .

On file
program

Quy trình thiết kế được giới thiệu trong Phần 14.1. Có ba cách tiếp cận cơ bản để thiết kế: thiết kế hướng hoạt động (Phần 14.2), thiết kế hướng dữ liệu (Phần 14.5) và thiết kế hướng đối tượng (Phần 14.6). Hai trường hợp của hướng hoạt động thiết kế được mô tả, phân tích luồng dữ liệu (Phần 14.3) và phân tích giao dịch (Phần 14.4). Thiết kế hướng đối tượng được ứng dụng cho bài thang toán máy nghiên cứu tình chất trong Phần 14.7 và cho bài nghiên cứu tình chất của Tổ chức MSG trong Phần 14.8. Quy định thiết kế được trình bày trong Phần 14.9. Các khu vực cạnh thiết kế của quy trình thử nghiệm đã được mô tả trong Phần 14.10 và được ứng dụng để nghiên cứu sự thật của Tổ chức MSG trong Phần 14.11. Các kỹ thuật chính thức cho thiết kế chi tiết được thảo luận trong Phần 14.12. Thời gian thiết kế hệ thống được mô tả trong Phần 14.13. Các công cụ và số liệu CASE cho quy trình thiết kế được trình bày lần cuối trong Phần 14.14 và 14.15. Chương trình này kết thúc với một cuộc thảo luận về các thức thức của quy trình thiết kế (Phần 14.16).

Tổng quan về tình hình nghiên cứu của Tổ chức MSG cho Chương 14 được thể hiện trong Hình 14.18 và cho vấn đề về thang máy trong Hình 14.19 .

HÌNH 14.18

Tổng quan về

Root root

nghiên cứu tình

lỗi cho Chương 14.

Thiết kế hướng đối tượng	Mục 14.8
Sơ đồ lớp tổng thể	Hình 14.13
Một phần của lớp sơ đồ có thể tổng hợp được các định dạng thuộc tính được thêm vào	Hình 14.14
Design chi tiết	Phụ lục G

HÌNH 14.19 Tổng quan về nghiên cứu tình huống bài toán thang máy cho Chương 14.

Thiết kế hướng đối tượng	Mục 14.7
Layer sơ đồ chi tiết	Hình 14.11

Vì
More nữa
Độc

Phân luồng dữ liệu và phân tích giao dịch được mô tả trong các cuốn sách như [Gane và Sarsen, 1979] và [Yourdon và Constantine, 1979].

Số tháng 3-tháng 4 năm 2005 của IEEE Software có chứa một số bài viết về thiết kế. Design to restore, mean is essential kể phần mềm để phát hiện, phản hồi ứng dụng và phục hồi từ các điều kiện ngoại lệ, được mô tả trong [Wirfs-Brock, 2006].

Briand, Bunse và Daly [2001] thảo luận về khả năng bảo đảm của các thiết kế hướng đối tượng. Một so sánh của cả kỹ năng thuật thiết kế hướng đối tượng và thiết kế cổ điển xuất hiện trong [Fichman và Kemerer, 1992]. Design design back system kiểm soát không lưu được mô tả trong [Jackson và Chapin, 2000].

Kỹ thuật thiết kế cho hiệu năng cao, hệ thống đáng tin đáng tin cậy được đưa ra trong [Stolper, 1999]. Một cách tiếp cận xác định để ước tính xu hướng thay đổi của một đối tượng hướng thiết kế xuất hiện trong [Tsantalis, Chatzigeorgiou, và Stephanides, 2005]. Một cuộc thảo luận về việc liệu thiết kế hướng đối tượng có trực quan xuất hiện trong [Hadar và Leron, 2008].

Các kỹ thuật thiết kế hình thức được mô tả trong [Hoare, 1987]. Vai trò quan trọng của kiến trúc sư được mô tả trong [McBride, 2007]. Tương tự như lập trình cấp, thiết kế cấp và hiệu quả của nó được mô tả trong [Lui, Chan, và Nosek, 2008].

Liên quan đến các đánh giá trong quá trình thiết kế, tài liệu gốc về kiểm tra thiết kế là [Fagan, 1976]; Information chi tiết có thể được lấy từ giấy đó. Những tiến bộ sau này trong các kỹ thuật đánh giá được mô tả trong [Fagan, 1986]. Đánh giá về kiến trúc được thảo luận trong [Maranzano et al., 2005].

Đối với thiết kế thời gian thực, các kỹ thuật có thể được tìm thấy trong [Liu, 2000] và [Gomaa, 2000]. So sánh bốn kỹ thuật thiết kế thời gian thực được tìm thấy trong [Kelly và Sherif, 1992]. Cách tiếp cận dựa trên tài liệu để thiết bị kế hoạch các hệ thống thời gian thực sự phức tạp được mô tả trong [Luqi, Zhang, Berzins và Qiao, 2004]. Design of the system đồng thời được mô tả trong [Magee và Kramer, 1999].

Các thước đo cho thiết kế được mô tả trong [Henry và Kafura, 1981] và [Zage và Zage, 1993]. Các thước đo cho thiết bị hướng đối tượng được thảo luận trong [Chidamber và Kemerer, 1994] và trong [Binkley và Schach, 1996]. One model for chất lượng hướng đối tượng được trình bày trong [Bansiya và Davis, 2002].

Kỷ yếu của Hội thảo Quốc tế về Đặc tả và Thiết kế Phần mềm là một nguồn thông tin toàn diện về các kỹ thuật design.

Main language design type data display 476

bộ truy cập 482

kiến trúc sư 486

thiết kế kiến trúc 466 sơ đồ

layer 476 phase complex

491 data stream data (DFA)

467 design direction data 465

quy trình thiết kế 483

thiết kế chi tiết 466

fan to 491

fan ra 491

design chung 466

thiết kế cao cấp 466 chiều

long 491 design logic 466

thiết kế cấp thấp 466

thiết kế mô-đun 466 bộ

biến đổi 482

thiết kế hướng đối tượng (OOD) 476

thiết kế hướng hoạt động 465 gói 486

thiết kế vật lý 466