

Trong chương này, trước tiên chúng tôi giải thích cách đọc DFD và mô tả cú pháp cơ bản của chúng. Sau đó, chúng tôi mô tả quy trình được sử dụng để xây dựng DFD lấy thông tin từ các trường hợp sử dụng và từ thông tin yêu cầu bổ sung được thu thập từ

người dùng.

SƠ ĐỒ DÒNG DỮ LIỆU

Đọc sơ đồ luồng dữ liệu

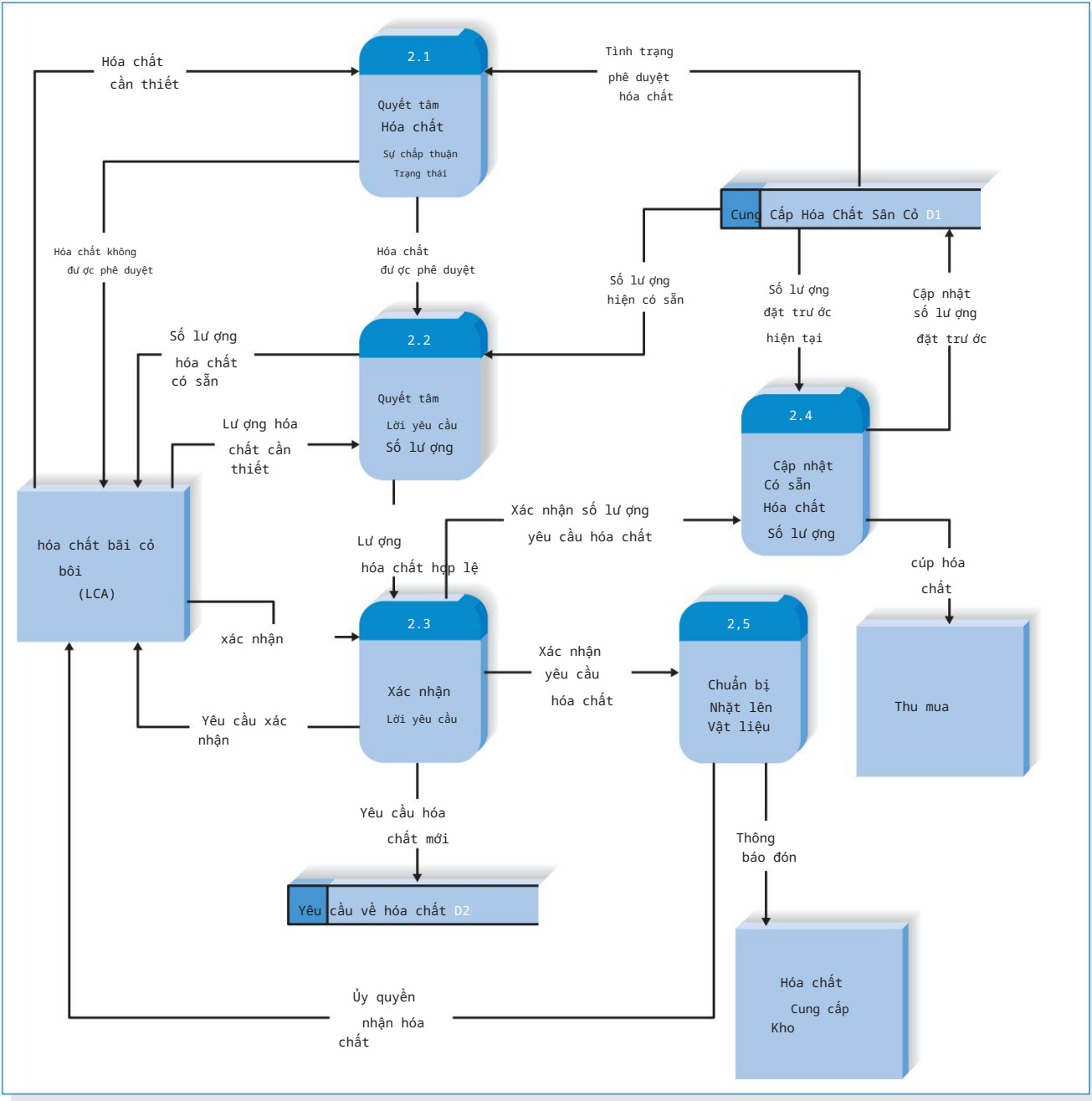
Hình 5-1 cho thấy một DFD cho sự kiện mà chúng tôi đã giới thiệu trong Chương 4, đó là sự kiện Máy phun hóa chất cho cỏ (LCA) yêu cầu hóa chất cho cỏ. Bằng cách kiểm tra DFD, nhà phân tích có thể hiểu quy trình LCA yêu cầu hóa chất cho cỏ. Hãy dành một chút thời gian để kiểm tra sơ đồ trước khi đọc tiếp. Làm thế nào nhiều làm bạn dư dãi đứng? Trước khi tiếp tục, bạn có thể xem lại trường hợp sử dụng của sự kiện này trong chương trước (Hình 4-1) và các yêu cầu chức năng (Hình 4-4).

Hầu hết mọi người từ các nền văn hóa phương Tây bắt đầu đọc sơ đồ từ trái sang phải, từ trên xuống dưới. Vì vậy, bất cứ khi nào có thể, đây là nơi hầu hết các nhà phân tích cố gắng bắt đầu DFD. Mục đầu tiên ở phía bên trái của Hình 5-1 là thực thể bên ngoài "Lawn Chemical Applicator (LCA)", là một hình chữ nhật đại diện cho từng nhân viên phải yêu cầu hóa chất mà họ sẽ sử dụng cho nhiệm vụ chăm sóc bãi cỏ của mình. Biểu tượng này có ba mũi tên hướng từ nó sang các biểu tượng hình chữ nhật tròn. Các mũi tên này biểu thị các luồng dữ liệu và cho thấy rằng thực thể bên ngoài (LCA) cung cấp ba "gói" dữ liệu cho các quy trình sử dụng dữ liệu. Bây giờ hãy xem lại Hình 4-1 và bạn sẽ thấy rằng các gói dữ liệu tương tự này được liệt kê là Đầu vào chính trong trường hợp sử dụng, với nguồn được liệt kê là LCA. Ngoài ra, có một số mũi tên đến thực thể bên ngoài LCA từ các hình chữ nhật được làm tròn, biểu thị các nhóm dữ liệu mà các quy trình tạo ra để truyền trở lại LCA. Các gói dữ liệu này được liệt kê trong Các kết quả đầu ra chính trong trường hợp sử dụng (Hình 4-1), với điểm đến được liệt kê là LCA.

Bây giờ hãy nhìn vào mũi tên chảy vào quy trình "Xác định Trạng thái Phê duyệt Hóa chất" từ phía bên phải. Để xác định xem hóa chất được yêu cầu có được phép sử dụng hay không, quy trình phải truy xuất một số thông tin từ kho lưu trữ.

Hình chữ nhật kết thúc mở có nhãn "Cung cấp hóa chất cho cỏ" được gọi là kho lưu trữ dữ liệu và nó đại diện cho một tập hợp dữ liệu được lưu trữ. Quy trình "Xác định trạng thái phê duyệt hóa chất" sử dụng mã định danh cho hóa chất được yêu cầu để tìm hóa chất được yêu cầu và để xác định xem đó là hóa chất được phê duyệt hay hóa chất không được phê duyệt. Lưu ý rằng "Danh sách các hóa chất được phê duyệt" được liệt kê là Đầu vào chính trong trường hợp sử dụng (Hình 4-1), với nguồn được liệt kê là kho lưu trữ dữ liệu Cung cấp hóa chất cho cỏ. Bây giờ, vẫn tham khảo Hình 4-1, lưu ý rằng mọi Đầu vào chính được liệt kê trong trường hợp sử dụng đều chảy vào một quy trình từ một thực thể bên ngoài hoặc dữ liệu được lưu trữ (được ghi chú bởi nguồn). Cũng lưu ý rằng mọi Đầu ra chính được liệt kê trong trường hợp sử dụng đều chảy ra đích (một thực thể bên ngoài hoặc bộ lưu trữ dữ liệu) trên sơ đồ luồng dữ liệu.

Bây giờ hãy xem xét kỹ hơn phần Các bước chính đã thực hiện của trường hợp sử dụng. Bạn có thể thấy rằng một số bước được liệt kê trong trường hợp sử dụng. Trên sơ đồ luồng dữ liệu, các bước này đã được tổ chức thành năm quy trình chính, mỗi quy trình thực hiện một thành phần chính của các tương tác được nêu chi tiết trong trường hợp sử dụng. Trên DFD (Hình 5-1), khi bạn đi theo các mũi tên bắt đầu với "Hóa chất cần thiết" từ LCA đến quy trình "Xác định tình trạng phê duyệt hóa chất", hãy tương ứng với LCA chỉ định hóa chất mà anh ta cần cho công việc. Hệ thống tra cứu hóa chất và phản hồi bằng một thông báo xác minh đó là hóa chất được phê duyệt hoặc thông báo cho



HÌNH 5-1

Yêu cầu DFD hóa chất cấp 1

LCA rằng hóa chất không thể được sử dụng. Đối với một hóa chất đã được phê duyệt, hệ thống sẽ tra cứu lượng hóa chất hiện có và thông báo cho LCA. LCA cho biết số lượng anh ta muốn. Bây giờ hãy xem mô tả về ca sử dụng (Hình 4-1) cho các bước 1-4 và chú ý cách ca sử dụng mô tả các quy trình đó bằng từ ngữ. Cũng lưu ý cách phần "Thông tin cho các bước" của ca sử dụng liệt kê các thành phần dữ liệu được sử dụng hoặc tạo ra bởi mỗi bước, tương ứng với các luồng vào và luồng ra từ các ký hiệu quy trình (2.1, 2.2) trên sơ đồ luồng dữ liệu (Hình 5-1).

Nhìn vào ba biểu tượng quy trình khác trong DFD và kiểm tra các luồng vào và ra của mỗi quy trình. Trên cơ sở dữ liệu chảy vào và chảy ra, hãy cố gắng hiểu quy trình đang làm gì. Kiểm tra sự hiểu biết của bạn bằng cách xem Các bước chính đã thực hiện và Thông tin cho các bước trong trữ ứng hợp sử dụng.

Bạn có thể nhận ra rằng quy trình "Xác nhận Yêu cầu" (2.3) nhận xác nhận yêu cầu hóa chất của LCA, đồng thời tạo và lưu trữ Yêu cầu Hóa chất mới. Bạn cũng có thể thấy rằng hai quy trình bổ sung được thực hiện bởi hệ thống. Số lượng hóa chất có sẵn được sửa đổi bằng cách đánh dấu số lượng được yêu cầu là "dành riêng" (và không còn có sẵn cho LCA khác). Cuối cùng, ủy quyền lấy hàng được cung cấp cho LCA và Kho cung cấp hóa chất được thông báo về việc lấy hàng đã được phê duyệt. Bạn có thể thấy các luồng từ quy trình 2.3 đến quy trình 2.4 và 2.5 mà đôi khi một quy trình gửi một luồng dữ liệu trực tiếp đến một quy trình khác.

Trữ ứng hợp sử dụng trong Hình 4-1 và sơ đồ luồng dữ liệu trong Hình 5-1 hoàn toàn có liên quan với nhau. Một trữ ứng hợp sử dụng được xây dựng tốt làm cho việc phát triển sơ đồ luồng dữ liệu trở nên khá đơn giản, mặc dù nhà phân tích sẽ phải đưa ra một số quyết định về mức độ chi tiết cần mô tả trong DFD. Các bước được nêu trong trữ ứng hợp sử dụng có thể được tổ chức thành các quy trình hợp lý trên DFD. Đầu vào chính và Đầu ra chính được liệt kê trong trữ ứng hợp sử dụng cung cấp danh sách các nguồn và đích tương ứng của các luồng vào và luồng ra của các quy trình. Phần Thông tin cho các bước hiển thị dữ liệu được đưa vào hoặc tạo ra theo từng bước của ca sử dụng và những dữ liệu này tương ứng với các luồng dữ liệu đi vào hoặc rời khỏi mỗi quy trình trên sơ đồ luồng dữ liệu.



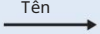
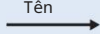




Các thành phần của sơ đồ luồng dữ liệu Bây

giờ bạn đã có cái nhìn thoáng qua về DFD, chúng tôi sẽ trình bày ngôn ngữ của DFD, ngôn ngữ này bao gồm một tập hợp các ký hiệu, quy ước đặt tên và quy tắc cú pháp. Có bốn ký hiệu trong ngôn ngữ DFD (quy trình, luồng dữ liệu, kho lưu trữ dữ liệu và các thực thể bên ngoài), mỗi ký hiệu được biểu thị bằng một ký hiệu đồ họa khác nhau. Có hai kiểu ký hiệu được sử dụng phổ biến, một kiểu do Chris Gane và Trish Sarson phát triển và kiểu kia do Tom DeMarco và Ed Yourdon² phát triển (Hình 5-2). Cái này không tốt hơn cái kia; một số tổ chức sử dụng kiểu biểu tượng Gane và Sarson, và những tổ chức khác sử dụng kiểu DeMarco/Yourdon. Chúng tôi sẽ sử dụng phong cách Gane và Sarson trong cuốn sách này.

Quy trình Một quy trình là một hoạt động hoặc một chức năng được thực hiện vì một số lý do kinh doanh cụ thể. Các quy trình có thể được thực hiện thủ công hoặc được vi tính hóa. Mọi quy trình nên được đặt tên bắt đầu bằng một động từ và kết thúc bằng một danh từ (ví dụ: "Xác định số lượng yêu cầu"). Tên phải ngắn gọn nhưng chứa đủ thông tin để người đọc có thể dễ dàng hiểu chính xác những gì họ làm. Nói chung, mỗi quy trình chỉ thực hiện một hoạt động, vì vậy hầu hết các nhà phân tích hệ thống tránh sử dụng từ "và" trong tên quy trình vì nó gợi ý rằng quy trình thực hiện một số hoạt động. Ngoài ra, mọi quy trình phải có ít nhất một luồng dữ liệu đầu vào và ít nhất một luồng dữ liệu đầu ra.

Hình 5-2 cho thấy các yếu tố cơ bản của một quy trình và cách chúng thường được đặt tên trong các công cụ CASE. Mỗi quy trình có một số nhận dạng duy nhất, tên và mô tả, tất cả đều được ghi chú trong kho lưu trữ CASE. Mô tả rõ ràng và

² Xem Chris Gane và Trish Sarson, *Structured Systems Analysis: Tools and Techniques*, Englewood Cliffs, NJ: Prentice Hall, 1979; Tom DeMarco, *Phân tích Cấu trúc và Đặc tả Hệ thống*, Englewood Cliffs, NJ: Prentice-Hall, 1979; và E. Yourdon và Larry L. Constantine, *Structured Design: Fundamentals of a Discipline of Computer Programme and Systems Design*, Englewood Cliffs, NJ: Prentice-Hall, 1979.

Sơ đồ luồng dữ liệu Yếu tố	Máy tính hỗ trợ điển hình Kỹ thuật phần mềm Lĩnh vực	Gane và Biểu tượng Sarson	DeMarco và bạn đồng hành Biểu tượng
Mỗi quy trình có một số một tên (pha động tử) một mô tả ít nhất một luồng dữ liệu đầu ra ít nhất một luồng dữ liệu đầu vào	Nhãn (tên) Loại (quy trình) Mô tả (nó là gì) số quy trình Mô tả quy trình (cấu trúc tiếng Anh) ghi chú		
Mỗi luồng dữ liệu có một tên (một danh từ) một mô tả một hoặc nhiều kết nối với một quá trình	Nhãn (tên) Loại (dòng chảy) Sự miêu tả Bi danh (tên khác) Thành phần (mô tả các yếu tố dữ liệu) ghi chú		
Mỗi kho lưu trữ dữ liệu có một số một tên (danh từ) một mô tả một hoặc nhiều luồng dữ liệu đầu vào một hoặc nhiều luồng dữ liệu đầu ra	Nhãn (tên) Loại (cửa hàng) Sự miêu tả Bi danh (tên khác) Thành phần (mô tả các yếu tố dữ liệu) ghi chú		
Mỗi thực thể bên ngoài có một tên (một danh từ) một mô tả	Nhãn (tên) Loại (thực thể) Sự miêu tả Bi danh (tên khác) Mô tả thực thể ghi chú		

HÌNH 5-2
Các yếu tố sơ đồ luồng dữ liệu

mô tả chính xác các bước và chi tiết các quy trình; cuối cùng, chúng được sử dụng để hướng dẫn các lập trình viên, những người cần tin học hóa các quy trình (hoặc những người viết sổ tay hướng dẫn chính sách cho các quy trình không được vi tính hóa). Các mô tả quy trình trở nên chi tiết hơn khi thông tin được biết về quy trình thông qua giai đoạn phân tích. Nhiều mô tả quy trình được viết dưới dạng câu văn bản đơn giản về những gì xảy ra. Các quy trình phức tạp hơn sử dụng các kỹ thuật chính thức hơn như tiếng Anh có cấu trúc, bảng quyết định hoặc cây quyết định sẽ được thảo luận trong phần sau.

Luồng dữ liệu Luồng dữ liệu là một mẫu dữ liệu đơn lẻ (ví dụ: số lượng có sẵn) (đôi khi được gọi là phần tử dữ liệu) hoặc một tập hợp logic của một số mẫu thông tin (ví dụ: yêu cầu hóa chất mới). Mỗi luồng dữ liệu nên được đặt tên bằng một danh từ. Mô tả của luồng dữ liệu liệt kê chính xác những phần tử dữ liệu mà luồng đó chứa. Ví dụ: luồng dữ liệu thông báo lấy hàng có thể liệt kê tên LCA, hóa chất và số lượng được yêu cầu dưới dạng các thành phần dữ liệu của nó.

Luồng dữ liệu là chất keo giữ các quy trình lại với nhau. Một đầu của mọi luồng dữ liệu sẽ luôn đến hoặc đi đến một quy trình, với mũi tên hiển thị hướng vào hoặc ra khỏi quy trình. Luồng dữ liệu cho biết đầu vào nào đi vào mỗi quy trình và đầu ra nào mà mỗi quy trình tạo ra. Mọi quy trình phải tạo ra ít nhất

một luồng dữ liệu đầu ra, vì nếu không có đầu ra, tiến trình sẽ không làm gì cả.

Tương tự như vậy, mỗi quá trình có ít nhất một luồng dữ liệu đầu vào, bởi vì rất khó, nếu không muốn nói là không thể, để tạo ra một đầu ra mà không có đầu vào.

Kho dữ liệu Kho dữ liệu là một tập hợp dữ liệu được lưu trữ theo một cách nào đó (được xác định sau khi tạo mô hình vật lý). Mỗi kho lưu trữ dữ liệu được đặt tên bằng một danh từ và được gán một số nhận dạng và mô tả. Các kho lưu trữ dữ liệu tạo thành điểm bắt đầu cho mô hình dữ liệu (được thảo luận trong chương tiếp theo) và là liên kết chính giữa mô hình quy trình và mô hình dữ liệu.

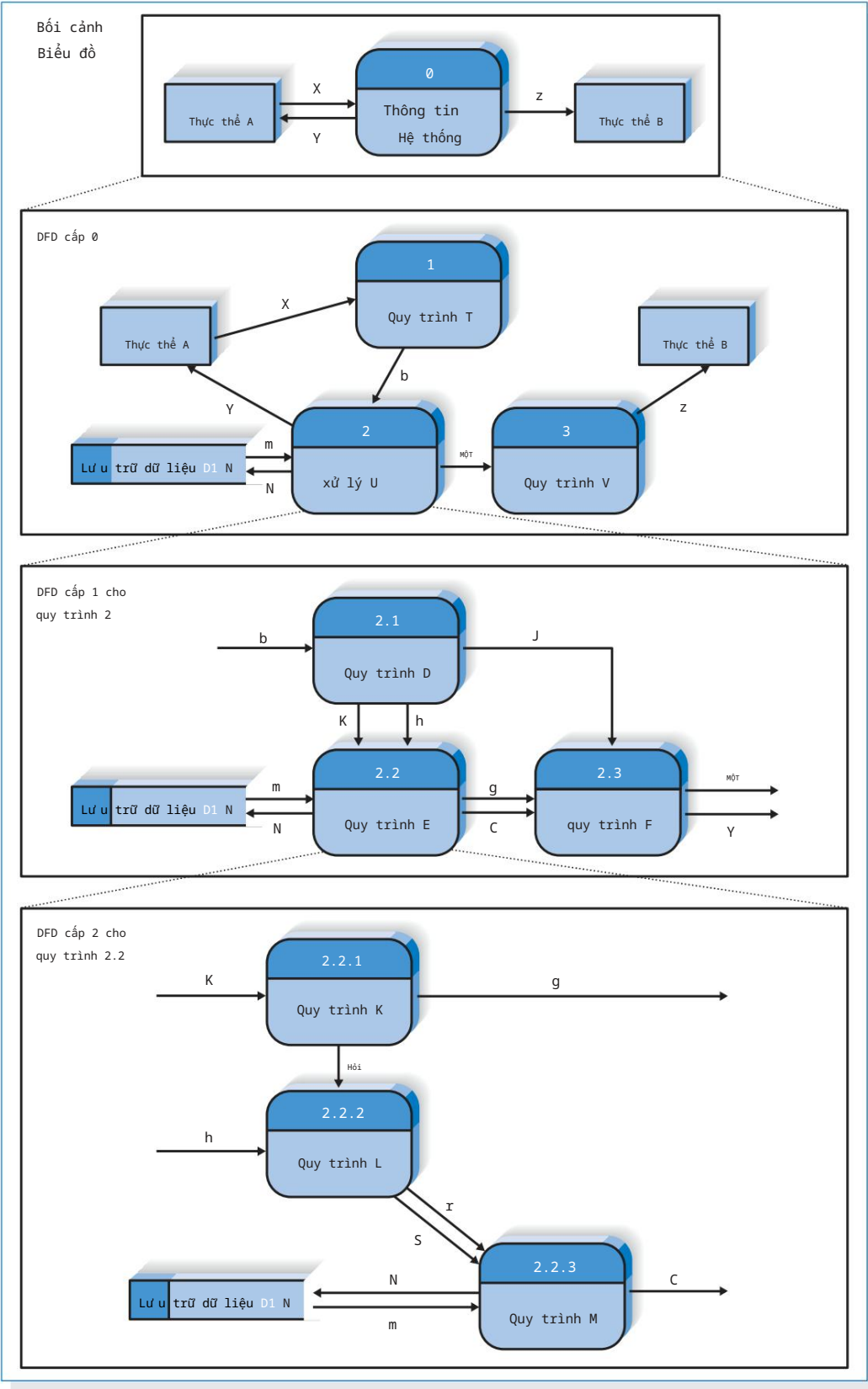
Luồng dữ liệu đi ra khỏi kho lưu trữ dữ liệu cho biết thông tin được lấy từ kho lưu trữ dữ liệu. Nhìn vào Hình 5-1, bạn có thể thấy quy trình 2.1 (Xác định Trạng thái Phê duyệt Hóa chất) truy xuất luồng dữ liệu Trạng thái Phê duyệt Hóa chất từ kho lưu trữ dữ liệu Cung cấp Hóa chất cho Có. Tương tự, Quy trình 2.2 (Xác định Số lượng Yêu cầu) truy xuất luồng dữ liệu Số lượng Có sẵn từ kho dữ liệu Cung cấp Hóa chất cho Có. Luồng dữ liệu đi vào kho lưu trữ dữ liệu cho biết thông tin được thêm vào kho lưu trữ dữ liệu. Ví dụ: quy trình 2.3 thêm luồng dữ liệu Yêu cầu Hóa chất Mới vào kho lưu trữ dữ liệu Yêu cầu Hóa chất. Cuối cùng, các luồng dữ liệu đi vào và ra khỏi kho lưu trữ dữ liệu chỉ ra rằng thông tin trong kho lưu trữ dữ liệu đã bị thay đổi (ví dụ: bằng cách truy xuất dữ liệu từ kho lưu trữ dữ liệu, thay đổi và lưu trữ lại). Trong Hình 5-1, chúng ta có thể thấy quy trình 2.4 (Cập nhật Số lượng Hóa chất Có sẵn) truy xuất số lượng dự trữ hiện tại từ kho dữ liệu Cung cấp Hóa chất Lawn, sửa đổi nó và ghi dữ liệu đã cập nhật trở lại kho dữ liệu.

Tất cả các kho lưu trữ dữ liệu phải có ít nhất một luồng dữ liệu đầu vào (hoặc nếu không thì chúng không bao giờ chứa bất kỳ dữ liệu nào), trừ khi chúng được tạo và duy trì bởi một hệ thống thông tin khác hoặc trên một trang khác của DFD. Tương tự như vậy, chúng có ít nhất một luồng dữ liệu đầu ra trên một số trang của DFD. (Tại sao phải lưu trữ dữ liệu nếu bạn không bao giờ sử dụng nó?) Trong trường hợp cùng một quy trình vừa lưu trữ dữ liệu vừa truy xuất dữ liệu từ kho lưu trữ dữ liệu, sẽ có xu hướng vẽ một luồng dữ liệu bằng một mũi tên ở cả hai đầu. Thực hành này là không chính xác, tuy nhiên. Luồng dữ liệu lưu trữ dữ liệu và luồng dữ liệu truy xuất dữ liệu phải luôn được hiển thị dưới dạng hai luồng dữ liệu riêng biệt.

Thực thể bên ngoài Một thực thể bên ngoài là một người, tổ chức, đơn vị tổ chức hoặc hệ thống bên ngoài hệ thống, như ng tưng tác với nó (ví dụ: khách hàng, cơ quan thanh toán bù trừ, tổ chức chính phủ, hệ thống kế toán). Thực thể bên ngoài thường tương ứng với tác nhân chính được xác định trong trường hợp sử dụng. Các thực thể bên ngoài cung cấp dữ liệu cho hệ thống hoặc nhận dữ liệu từ hệ thống và dùng để thiết lập ranh giới hệ thống. Mọi thực thể bên ngoài đều có tên và mô tả. Điểm mấu chốt cần nhớ về một thực thể bên ngoài là nó ở bên ngoài hệ thống, nhưng có thể là một phần của tổ chức hoặc không. Những người sử dụng thông tin từ hệ thống để thực hiện các quy trình khác hoặc những người quyết định thông tin nào sẽ đi vào hệ thống được ghi lại dưới dạng các thực thể bên ngoài (ví dụ: người quản lý, nhân viên).

Sử dụng sơ đồ luồng dữ liệu để xác định quy trình kinh doanh Hầu hết các

quy trình kinh doanh đều quá phức tạp để giải thích trong một DFD. Do đó, hầu hết các mô hình quy trình đều bao gồm một tập hợp các DFD. DFD đầu tiên cung cấp một bản tóm tắt về toàn bộ hệ thống, với các DFD bổ sung cung cấp ngày càng nhiều chi tiết hơn về từng phần của quy trình kinh doanh tổng thể. Do đó, một nguyên tắc quan trọng trong mô hình hóa quy trình với DFD là phân tách quy trình kinh doanh thành một hệ thống phân cấp DFD, với mỗi cấp độ xuống dưới hệ thống phân cấp đại diện cho phạm vi ít hơn như chỉ tiết hơn. Hình 5-3 cho thấy cách một quy trình nghiệp vụ có thể được phân tách thành nhiều cấp DFD.



HÌNH 5-3

Mối quan hệ giữa các mức của Sơ đồ luồng dữ liệu (DFD)

Sơ đồ ngữ cảnh DFD đầu tiên trong mọi mô hình quy trình kinh doanh, dù là hệ thống thủ công hay hệ thống máy tính, là sơ đồ ngữ cảnh (xem Hình 5-3). Như tên gợi ý, sơ đồ ngữ cảnh hiển thị toàn bộ hệ thống trong ngữ cảnh với môi trường của nó. Tất cả các mô hình quy trình đều có một sơ đồ ngữ cảnh.

Biểu đồ ngữ cảnh hiển thị quy trình kinh doanh tổng thể chỉ là một quy trình (nghĩa là bản thân hệ thống) và hiển thị các luồng dữ liệu đến và từ các thực thể bên ngoài. Các kho lưu trữ dữ liệu thư ờng không đư ợc đư a vào sơ đồ ngữ cảnh, trừ khi chúng đư ợc “sở hữu” bởi các hệ thống hoặc quy trình không phải là hệ thống đư ợc ghi lại. Ví dụ, một hệ thống thông tin đư ợc thư viện trư ờng đại học sử dụng để ghi lại những ngư ời đã mượn sách có thể sẽ kiểm tra cơ sở dữ liệu thông tin sinh viên của cơ quan đăng ký để xem liệu một sinh viên hiện có đăng ký tại trư ờng đại học hay không. Trong sơ đồ ngữ cảnh này, kho lưu trữ dữ liệu thông tin sinh viên của nhà đăng ký có thể đư ợc hiển thị trên sơ đồ ngữ cảnh vì nó nằm ngoài hệ thống thư viện, nhưng đư ợc sử dụng bởi nó. Tuy nhiên, nhiều tổ chức sẽ hiển thị đây là một thực thể bên ngoài đư ợc gọi là “Hệ thống thông tin sinh viên của công ty đăng ký”, chứ không phải là một kho lưu trữ dữ liệu.

Sơ đồ mức 0 DFD tiếp theo đư ợc gọi là sơ đồ mức 0 hoặc DFD mức 0. (Xem Hình 5-3.) Biểu đồ mức 0 hiển thị tất cả các quy trình ở mức đánh số đầu tiên (nghĩa là các quy trình đư ợc đánh số từ 1 đến 3), các kho lưu trữ dữ liệu, các thực thể bên ngoài và các luồng dữ liệu giữa chúng. Mục đích của DFD mức 0 là hiển thị tất cả các quy trình cấp cao chính của hệ thống và cách chúng liên quan với nhau. Tất cả các mô hình quy trình đều có một và chỉ một DFD cấp 0.

Một nguyên tắc quan trọng khác trong việc tạo các bộ DFD là cân bằng. Cân bằng có nghĩa là đảm bảo rằng tất cả thông tin đư ợc trình bày trong DFD ở một cấp độ đư ợc thể hiện chính xác trong DFD cấp độ tiếp theo. Điều này không có nghĩa là thông tin giống hệt nhau, nhưng thông tin đó đư ợc hiển thị phù hợp. Có một sự khác biệt tinh tế về ý nghĩa giữa hai từ này sẽ sớm trở nên rõ ràng, nhưng hiện tại, hãy so sánh biểu đồ ngữ cảnh với DFD cấp 0 trong Hình 5-3 để xem hai từ này cân bằng như thế nào. Trong trư ờng hợp này, chúng ta thấy rằng các thực thể bên ngoài (A, B) giống hệt nhau giữa hai sơ đồ và luồng dữ liệu đến và từ các thực thể bên ngoài trong sơ đồ ngữ cảnh (X, Y, Z) cũng xuất hiện trong DFD mức 0. DFD mức 0 thay thế một quy trình đơn lẻ của sơ đồ ngữ cảnh (luôn đư ợc đánh số 0) bằng ba quy trình (1, 2, 3), thêm kho lưu trữ dữ liệu (D1) và bao gồm hai luồng dữ liệu bổ sung không có trong sơ đồ ngữ cảnh (luồng dữ liệu B từ quy trình 1 đến quy trình 2; luồng dữ liệu A từ quy trình 2 đến quy trình 3).

Ba quy trình này và hai luồng dữ liệu đư ợc chứa trong quy trình 0. Chúng không đư ợc hiển thị trên sơ đồ ngữ cảnh vì chúng là các thành phần bên trong của quy trình 0. Sơ đồ ngữ cảnh cố tình che giấu một số sự phức tạp của hệ thống để giúp ngư ời đọc dễ hiểu hơn. Chỉ sau khi ngư ời đọc hiểu sơ đồ ngữ cảnh, nhà phân tích mới “mở” quy trình 0 để hiển thị các hoạt động bên trong của nó bằng cách phân tách sơ đồ ngữ cảnh thành DFD cấp 0, hiển thị chi tiết hơn về các quy trình và luồng dữ liệu bên trong hệ thống.

Sơ đồ mức 1 Cũng giống như cách mà sơ đồ ngữ cảnh cố tình che giấu một số sự phức tạp của hệ thống, thì DFD mức 0 cũng vậy. DFD cấp 0 chỉ hiển thị cách các quy trình cấp cao chính trong hệ thống tương tác với nhau. Mỗi quy trình trên DFD cấp 0 có thể đư ợc phân tách thành một DFD rõ ràng hơn, đư ợc gọi là sơ đồ cấp 1 hoặc DFD cấp 1, biểu thị cách thức hoạt động của nó một cách chi tiết hơn. DFD đư ợc minh họa trong Hình 5-1 là DFD cấp 1.

Nói chung, tất cả các mô hình quy trình đều có số sơ đồ mức 1 bằng với số quy trình trên sơ đồ cấp 0; mọi quy trình trong DFD cấp 0 sẽ được phân tách thành DFD cấp 1 của chính nó, vì vậy DFD cấp 0 trong Hình 5-3 sẽ có ba DFD cấp 1 (một cho quy trình 1, một cho quy trình 2, một cho quy trình 3). Để đơn giản, chúng tôi đã chọn chỉ hiển thị một DFD cấp 1 trong hình này, DFD cho quy trình 2. Các quy trình trong DFD cấp 1 được đánh số trên cơ sở quy trình được phân tách.

Trong ví dụ này, chúng tôi đang phân tách quy trình 2, vì vậy các quy trình trong DFD cấp 1 này được đánh số 2.1, 2.2 và 2.3.

Các quy trình 2.1, 2.2 và 2.3 là con của quy trình 2 và quy trình 2 là cha của các quy trình 2.1, 2.2 và 2.3. Ba tiến trình con này hoàn toàn tạo nên tiến trình 2. Tập hợp con và cha giống hệt nhau; chúng chỉ đơn giản là những cách khác nhau để nhìn vào cùng một thứ. Khi một tiến trình cha được phân tách thành các tiến trình con, các tiến trình con của nó phải thực hiện đầy đủ tất cả các chức năng của nó, giống như cách cắt một chiếc bánh sẽ tạo ra một tập hợp các lát tạo nên toàn bộ và đầy đủ chiếc bánh. Mặc dù các lát có thể không có cùng kích thước, nhưng tập hợp các lát giống hệt với toàn bộ chiếc bánh; không có gì được bỏ qua bằng cách cắt chiếc bánh.

Một lần nữa, điều rất quan trọng là phải đảm bảo rằng các DFD cấp 0 và cấp 1 được cân bằng. DFD mức 0 cho thấy quá trình 2 truy cập kho dữ liệu D1, có hai luồng dữ liệu đầu vào (B, M) và có ba luồng dữ liệu đầu ra (A, N và Y). Kiểm tra DFD cấp 1 cho thấy lưu trữ dữ liệu và luồng dữ liệu giống nhau. Một lần nữa, chúng ta thấy rằng năm luồng dữ liệu mới đã được thêm vào (C, G, H, J, K) ở cấp độ này. Các luồng dữ liệu này được chứa trong quy trình 2 và do đó không được ghi lại trong DFD cấp 0. Chỉ khi chúng ta phân tách hoặc mở tiến trình 2 thông qua DFD cấp 1, chúng ta mới thấy chúng tồn tại.

DFD cấp 1 hiển thị chính xác hơn quy trình nào sử dụng luồng dữ liệu đầu vào B (quy trình 2.1) và quy trình nào tạo ra luồng dữ liệu đầu ra A và Y (quy trình 2.3). Tuy nhiên, lưu ý rằng DFD cấp 1 không hiển thị các luồng dữ liệu này đến từ đâu hoặc đi đến đâu. Ví dụ, để tìm nguồn của luồng dữ liệu B, chúng ta phải di chuyển lên DFD cấp 0, hiển thị luồng dữ liệu B đến từ thực thể bên ngoài B. Tương tự như vậy, nếu chúng ta theo dõi luồng dữ liệu từ A lên đến cấp 0 DFD thì ta thấy nó chuyển sang process 3 nhưng vẫn chưa biết chính xác process nào trong process 3 sử dụng nó (ví dụ process 3.1, 3.2). Để xác định nguồn chính xác, chúng tôi sẽ phải kiểm tra DFD cấp 1 cho quy trình 3.

Ví dụ này cho thấy một nhược điểm của việc phân tách DFD trên nhiều trang. Để tìm nguồn và đích chính xác của luồng dữ liệu, người ta thường phải theo dõi luồng dữ liệu qua một số DFD trên các trang khác nhau. Một số giải pháp thay thế cho phương pháp này để phân tách DFD đã được đề xuất, nhưng không có phương án nào được sử dụng phổ biến như phương pháp “truyền thống”. Cách thay thế phổ biến nhất là hiển thị nguồn và đích của các luồng dữ liệu đến và đi từ các thực thể bên ngoài (cũng như các kho lưu trữ dữ liệu) ở các DFD cấp thấp hơn. Thực tế là hầu hết các luồng dữ liệu đến hoặc từ các kho lưu trữ dữ liệu và các thực thể bên ngoài, chứ không phải các quy trình trên các trang DFD khác, có thể đơn giản hóa đáng kể việc đọc các DFD nhiều trang. Chúng tôi tin rằng đây là một cách tiếp cận tốt hơn, vì vậy khi giảng dạy các khóa học của mình, chúng tôi hiển thị các thực thể bên ngoài trên tất cả các DFD, bao gồm các DFD cấp 1 trở xuống.

Biểu đồ mức 2 Phần dư của Hình 5-3 cho thấy mức độ phân tách tiếp theo: biểu đồ mức 2, hoặc DFD mức 2, cho quy trình 2.2. DFD này cho thấy quy trình 2.2 được phân tách thành ba quy trình (2.2.1, 2.2.2 và 2.2.3). Biểu đồ mức 1 cho quy trình 2.2 hiển thị các tương tác với kho lưu trữ dữ liệu D1 mà chúng ta thấy trong DFD cấp 2 như xảy ra trong quy trình 2.2.3. Tương tự, DFD cấp 2

đối với 2.2 hiển thị hai luồng dữ liệu đầu vào (H, K) và hai luồng dữ liệu đầu ra (C, G), mà chúng ta cũng thấy trên sơ đồ mức 2, cùng với một số luồng dữ liệu mới (Q, R, S). Do đó, hai DFD được cân bằng.

Đôi khi rất khó để nhớ mức DFD nào. Có thể hữu ích khi nhớ rằng các số cấp đề cập đến số điểm thập phân trong các số quy trình trên DFD. DFD cấp 0 có số quy trình không có dấu thập phân (ví dụ: 1, 2), trong khi DFD cấp 1 có số quy trình có một dấu thập phân (ví dụ: 2.3, 5.1), DFD cấp 2 có số có hai dấu thập phân (ví dụ: 1.2.5, 3.3.2), v.v.

Luồng dữ liệu thay thế Giả sử rằng một quy trình tạo ra hai luồng dữ liệu khác nhau trong các trường hợp khác nhau. Ví dụ: một quy trình kiểm soát chất lượng có thể tạo ra một vật dụng được phê duyệt chất lượng hoặc một vật dụng bị lỗi, hoặc quá trình tìm kiếm một loại hóa chất của chúng tôi có thể tìm thấy nó được phê duyệt hoặc không được phê duyệt để sử dụng. Làm cách nào để hiển thị các trường hợp thay thế này trong DFD? Câu trả lời là chúng tôi hiển thị cả hai luồng dữ liệu và sử dụng mô tả quy trình để giải thích rằng chúng là các lựa chọn thay thế. Bản thân DFD không có gì cho thấy các luồng dữ liệu là loại trừ lẫn nhau. Ví dụ, quy trình 2.1 trên DFD mức 1 tạo ra ba luồng dữ liệu đầu ra (H, J, K). Nếu không đọc mô tả văn bản của quy trình 2.1, chúng tôi không biết liệu chúng được sản xuất đồng thời hay chúng loại trừ lẫn nhau.

Mô tả quy trình Mục

Mục đích của các mô tả quy trình là để giải thích quy trình làm gì và cung cấp thông tin bổ sung mà DFD không cung cấp. Khi chúng tôi chuyển qua SDLC, chúng tôi dần dần chuyển từ các mô tả văn bản chung về các yêu cầu sang các mô tả ngày càng chính xác hơn, cuối cùng được dịch sang các ngôn ngữ lập trình rất chính xác. Trong hầu hết các trường hợp, quy trình đủ đơn giản để định nghĩa yêu cầu, trường hợp sử dụng và DFD với mô tả văn bản đơn giản cùng nhau cung cấp đủ chi tiết để hỗ trợ các hoạt động trong giai đoạn thiết kế. Tuy nhiên, đôi khi, quy trình đủ phức tạp để có thể hưởng lợi từ một mô tả quy trình chi tiết hơn giải thích logic diễn ra bên trong quy trình.

Ba kỹ thuật thường được sử dụng để mô tả logic xử lý phức tạp hơn: tiếng Anh có cấu trúc, cây quyết định và bảng quyết định. Các quy trình rất phức tạp có thể sử dụng kết hợp tiếng Anh có cấu trúc và cây quyết định hoặc bảng quyết định.

Tiếng Anh có cấu trúc sử dụng các câu ngắn để mô tả công việc mà một quy trình thực hiện. Cây quyết định hiển thị logic quyết định (câu lệnh IF) dưới dạng tập hợp các nút (câu hỏi) và nhánh (câu trả lời). Các bảng quyết định biểu thị các quyết định chính sách phức tạp dưới dạng các quy tắc liên kết các điều kiện khác nhau với các hành động. Vì những kỹ thuật này thường được thảo luận trong các tài liệu lập trình nên chúng tôi sẽ không giải thích chi tiết về chúng ở đây. Chúng hữu ích cho nhà phân tích hệ thống trong việc truyền đạt sự hiểu biết đúng đắn về những gì diễn ra “bên trong” một quy trình.

TẠO SƠ ĐỒ LUỒNG DỮ LIỆU

Sơ đồ luồng dữ liệu bắt đầu với thông tin trong các trường hợp sử dụng và định nghĩa yêu cầu. Mặc dù các trường hợp sử dụng được tạo bởi người dùng và nhóm dự án làm việc cùng nhau, DFD thường được tạo bởi nhóm dự án và sau đó được người dùng xem xét. Nói chung, tập hợp các DFD tạo nên mô hình quy trình

chỉ cần tích hợp các trường hợp sử dụng riêng lẻ (và thêm vào bất kỳ quy trình nào trong định nghĩa yêu cầu không được chọn làm trường hợp sử dụng). Nhóm dự án lấy các trường hợp sử dụng và viết lại chúng dưới dạng DFD. Tuy nhiên, vì DFD có các quy tắc chính thức về ký hiệu và cú pháp mà các trường hợp sử dụng không có, nhóm dự án đôi khi phải sửa đổi một số thông tin trong các trường hợp sử dụng để làm cho chúng phù hợp với các quy tắc DFD. Các loại thay đổi phổ biến nhất là tên của các trường hợp sử dụng trở thành quy trình và đầu vào và đầu ra trở thành luồng dữ liệu. Loại thay đổi phổ biến thứ hai là kết hợp một số đầu vào và đầu ra nhỏ trong các trường hợp sử dụng thành các luồng dữ liệu lớn hơn trong DFD (ví dụ: kết hợp ba đầu vào riêng biệt, chẳng hạn như “tên khách hàng”, “địa chỉ khách hàng” và “số điện thoại của khách hàng số,” vào một luồng dữ liệu, chẳng hạn như “thông tin khách hàng”).

Các nhóm dự án thường sử dụng các công cụ mô hình hóa quy trình hoặc công cụ CASE để vẽ các mô hình quy trình. Các công cụ đơn giản như Visio chứa các bộ biểu tượng DFD và cho phép tạo và sửa đổi sơ đồ dễ dàng. Các công cụ mô hình hóa quy trình khác chẳng hạn như BPWin hiểu DFD và có thể thực hiện kiểm tra cú pháp đơn giản để đảm bảo rằng DFD ít nhất là đúng ở một mức độ nào đó. Một công cụ CASE đầy đủ, chẳng hạn như Visible Analyst Workbench, cung cấp nhiều khả năng ngoài việc mô hình hóa quy trình (ví dụ: mô hình hóa dữ liệu như được thảo luận trong chương tiếp theo). Các công cụ CASE có xu hướng phức tạp và mặc dù chúng có giá trị đối với các dự án lớn và phức tạp, nhưng chúng thường có giá cao hơn so với các dự án đơn giản. Hình 5-4 cho thấy một màn hình mẫu từ công cụ CASE Visible Analyst.

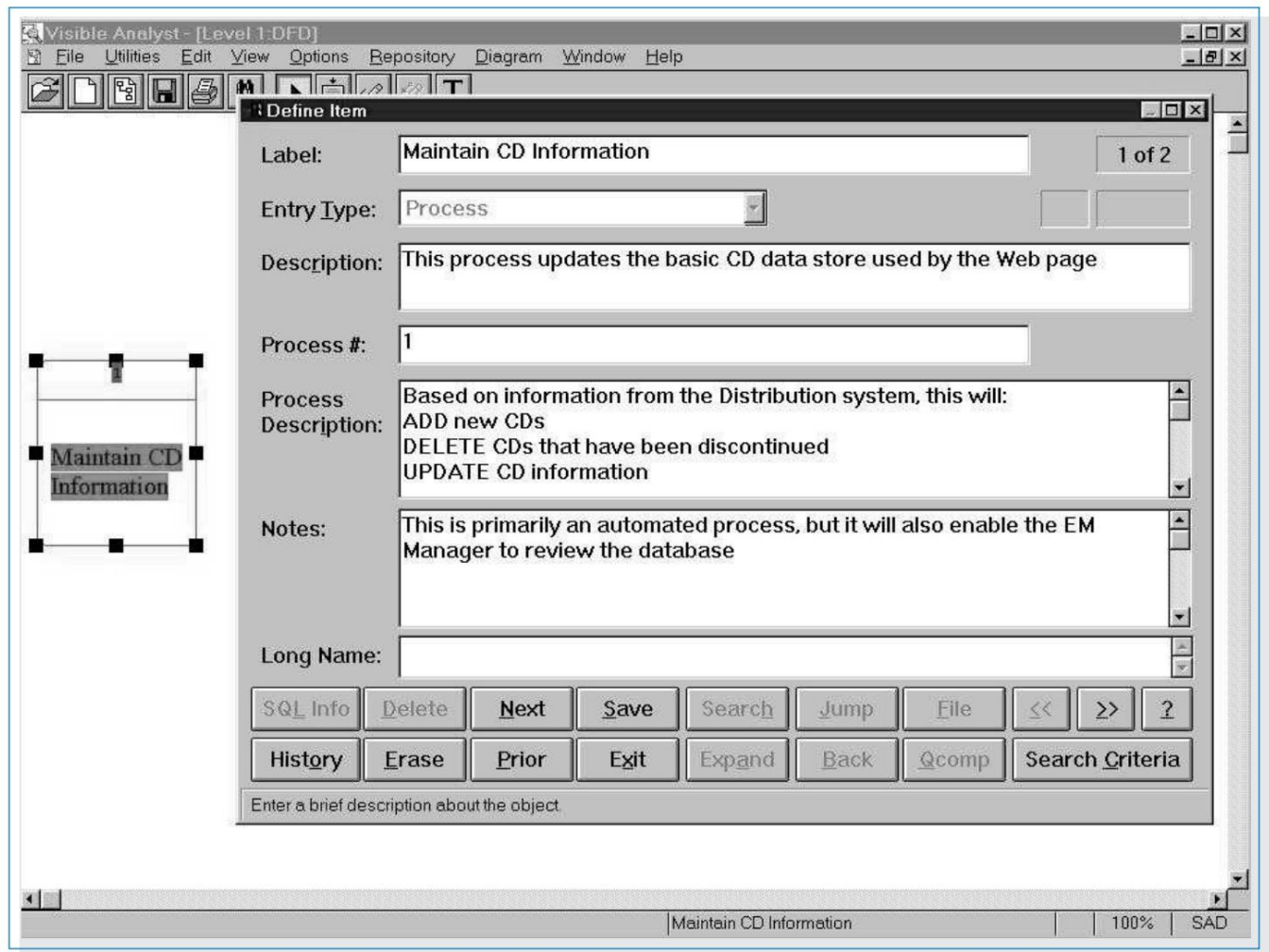
Việc xây dựng một mô hình quy trình có nhiều mức DFD thường bao gồm một số bước. Một số nhà phân tích thích bắt đầu mô hình hóa quy trình bằng cách tập trung đầu tiên vào biểu đồ mức 0. Chúng tôi thấy hữu ích khi xây dựng sơ đồ ngữ cảnh trước tiên hiển thị tất cả các thực thể bên ngoài và các luồng dữ liệu bắt nguồn từ hoặc kết thúc trong chúng. Thứ hai, nhóm tạo một đoạn DFD cho mỗi trường hợp sử dụng cho biết cách trường hợp sử dụng trao đổi luồng dữ liệu với các thực thể bên ngoài và kho lưu trữ dữ liệu. Thứ ba, các mảnh DFD này được tổ chức thành một DFD cấp 0. Thứ tư, nhóm phát triển các DFD cấp 1, dựa trên các bước trong từng trường hợp sử dụng, để giải thích rõ hơn cách chúng hoạt động. Trong một số trường hợp, các DFD cấp 1 này được phân tách thành DFD cấp 2, DFD cấp 3, DFD cấp 4, v.v. Thứ năm, nhóm xác thực bộ DFD để đảm bảo rằng chúng hoàn chỉnh và chính xác.

Trong các phần sau, quá trình mô hình hóa được minh họa với Holiday Hệ thống thông tin phục vụ du lịch.

Tạo sơ đồ ngữ cảnh

Sơ đồ ngữ cảnh xác định cách quy trình kinh doanh hoặc hệ thống máy tính tương tác với môi trường của nó—chủ yếu là các thực thể bên ngoài. Để tạo biểu đồ ngữ cảnh, bạn chỉ cần vẽ một ký hiệu quy trình cho quy trình kinh doanh hoặc hệ thống đang được mô hình hóa (được đánh số 0 và đặt tên cho quy trình hoặc hệ thống). Bạn đọc qua các trường hợp sử dụng và thêm đầu vào và đầu ra được liệt kê trên biểu mẫu, cũng như nguồn và đích của chúng. Thông thường, tất cả các đầu vào và đầu ra sẽ đến từ hoặc đi đến các thực thể bên ngoài như một người, tổ chức hoặc hệ thống thông tin khác.

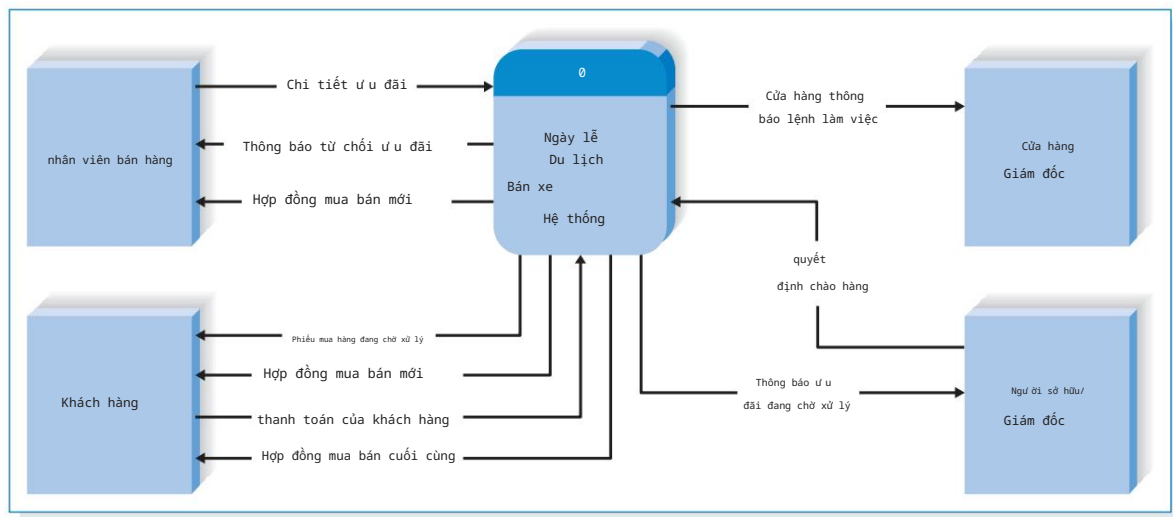
Nếu bất kỳ đầu vào và đầu ra nào kết nối trực tiếp với các kho lưu trữ dữ liệu trong một hệ thống bên ngoài, thì cách tốt nhất là tạo một thực thể bên ngoài được đặt tên theo hệ thống sở hữu kho lưu trữ dữ liệu. Không có kho lưu trữ dữ liệu nào bên trong quy trình/hệ thống do chính quy trình hoặc hệ thống tạo ra được đưa vào sơ đồ ngữ cảnh, bởi vì chúng nằm “bên trong” hệ thống. Vì đôi khi có quá nhiều đầu vào và đầu ra nên chúng ta thường kết hợp một số luồng dữ liệu nhỏ thành các luồng dữ liệu lớn hơn.



HÌNH 5-4

Nhập quy trình sơ đồ luồng dữ liệu trong công cụ kỹ thuật phần mềm hỗ trợ máy tính

Hình 5-5 cho thấy sơ đồ ngữ cảnh của hệ thống Phươg tiện Du lịch Kỳ nghỉ tập trung vào doanh số bán xe. Hãy dành một chút thời gian để xem lại hệ thống này như được mô tả trong Chương 4 và xem xét các trường hợp sử dụng trong Hình 4-11. Bạn có thể thấy từ các phần Đầu vào và Đầu ra chính trong các trường hợp sử dụng trong Hình 4-11 rằng hệ thống có nhiều tương tác với thực thể bên ngoài nhân viên bán hàng. Chúng tôi đã đơn giản hóa các luồng vào và luồng ra này thành ba luồng dữ liệu chính trên biểu đồ ngữ cảnh. Nếu chúng ta đưa vào từng luồng dữ liệu nhỏ, sơ đồ ngữ cảnh sẽ trở nên quá lộn xộn. Các luồng dữ liệu nhỏ hơn sẽ trở nên rõ ràng khi chúng ta phân tách biểu đồ ngữ cảnh thành các mức chi tiết hơn. Lưu ý rằng chúng tôi đã thành lập ba thực thể bên ngoài để đại diện cho các bộ phận của tổ chức Xe Du lịch Kỳ nghỉ nhận thông tin từ hoặc cung cấp thông tin cho hệ thống này. Nhân viên bán hàng cung cấp thông tin đầu vào chính cho hệ thống và các đơn đặt hàng công việc tại cửa hàng được chuyển đến người quản lý cửa hàng. Chủ sở hữu hoặc người quản lý công ty cung cấp thông tin cho hệ thống.



HÌNH 5-5

Sơ đồ ngữ cảnh hệ thống bán xe du lịch ngày lễ

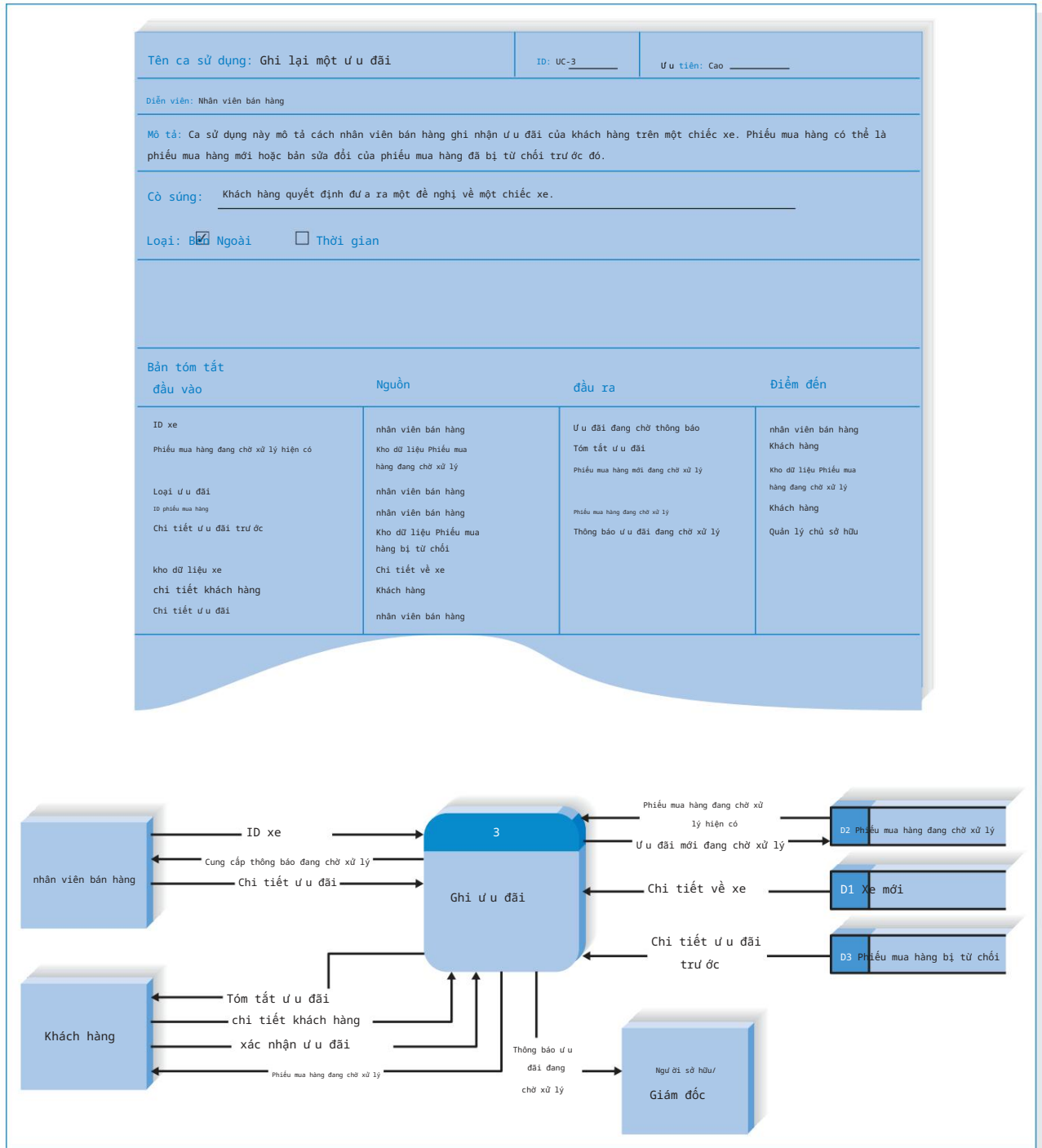
Tạo các đoạn sơ đồ luồng dữ liệu Một đoạn DFD là

một phần của DFD mà cuối cùng sẽ được kết hợp với các đoạn DFD khác để tạo thành một sơ đồ DFD. Trong bước này, mỗi trừu tượng hợp sử dụng được chuyển đổi thành một đoạn DFD. Bạn bắt đầu bằng cách lấy từng trừu tượng hợp sử dụng và vẽ một đoạn DFD, sử dụng thông tin được cung cấp ở đầu trừu tượng hợp sử dụng: tên, số ID và các đầu vào và đầu ra chính. Thông tin về các bước chính tạo nên mỗi trừu tượng hợp sử dụng bị bỏ qua tại thời điểm này; nó sẽ được sử dụng trong một bước sau. Hình 5-6 cho thấy một trừu tượng hợp sử dụng và đoạn DFD được tạo từ trừu tượng hợp đó.

Một lần nữa, một số thay đổi nhỏ nhưng quan trọng thường được thực hiện khi chuyển đổi trừu tượng hợp sử dụng thành DFD. Hai thay đổi phổ biến nhất là sửa đổi tên quy trình và bổ sung các luồng dữ liệu. Không có quy tắc chính thức nào cho tên trừu tượng hợp sử dụng, nhưng có các quy tắc chính thức để đặt tên quy trình trên DFD. Tất cả các tên quy trình phải là một cụm động từ—chúng phải bắt đầu bằng một động từ và bao gồm một danh từ. (Xem Hình 5-2.) Không phải tất cả tên ca sử dụng của chúng ta đều được cấu trúc theo cách này, vì vậy đôi khi chúng ta cần thay đổi chúng. Điều quan trọng là phải có quan điểm nhất quán khi đặt tên cho các quy trình. Ví dụ, DFD trong Hình 5-6 được viết từ quan điểm của đại lý, không phải của khách hàng. Tất cả các tên và mô tả quy trình được viết dưới dạng các hoạt động mà nhân viên thực hiện. Theo truyền thống, thiết kế các quy trình theo quan điểm của tổ chức vận hành hệ thống, vì vậy điều này đôi khi yêu cầu một số thay đổi bổ sung về tên.

Thay đổi phổ biến thứ hai là việc bổ sung các luồng dữ liệu. Các trừu tượng hợp sử dụng được viết để mô tả cách hệ thống và người dùng tương tác. Thông thường, chúng không mô tả cách hệ thống lấy dữ liệu, vì vậy trừu tượng hợp sử dụng thường bỏ qua các luồng dữ liệu được đọc từ kho lưu trữ dữ liệu. Khi tạo các phân đoạn DFD, điều quan trọng là đảm bảo rằng mọi thông tin được cung cấp cho người dùng đều được lấy từ kho lưu trữ dữ liệu. Cách dễ nhất để thực hiện việc này là trước tiên tạo đoạn DFD với các đầu vào và đầu ra chính được liệt kê trong trừu tượng hợp sử dụng, sau đó xác minh rằng tất cả các đầu ra đều có đủ đầu vào để tạo chúng.

Không có quy tắc chính thức nào về cách bố trí các quy trình, luồng dữ liệu, kho lưu trữ dữ liệu và các thực thể bên ngoài trong DFD. Hầu hết các nhà phân tích hệ thống cố gắng đặt quy trình ở giữa đoạn DFD, với các đầu vào chính bắt đầu từ



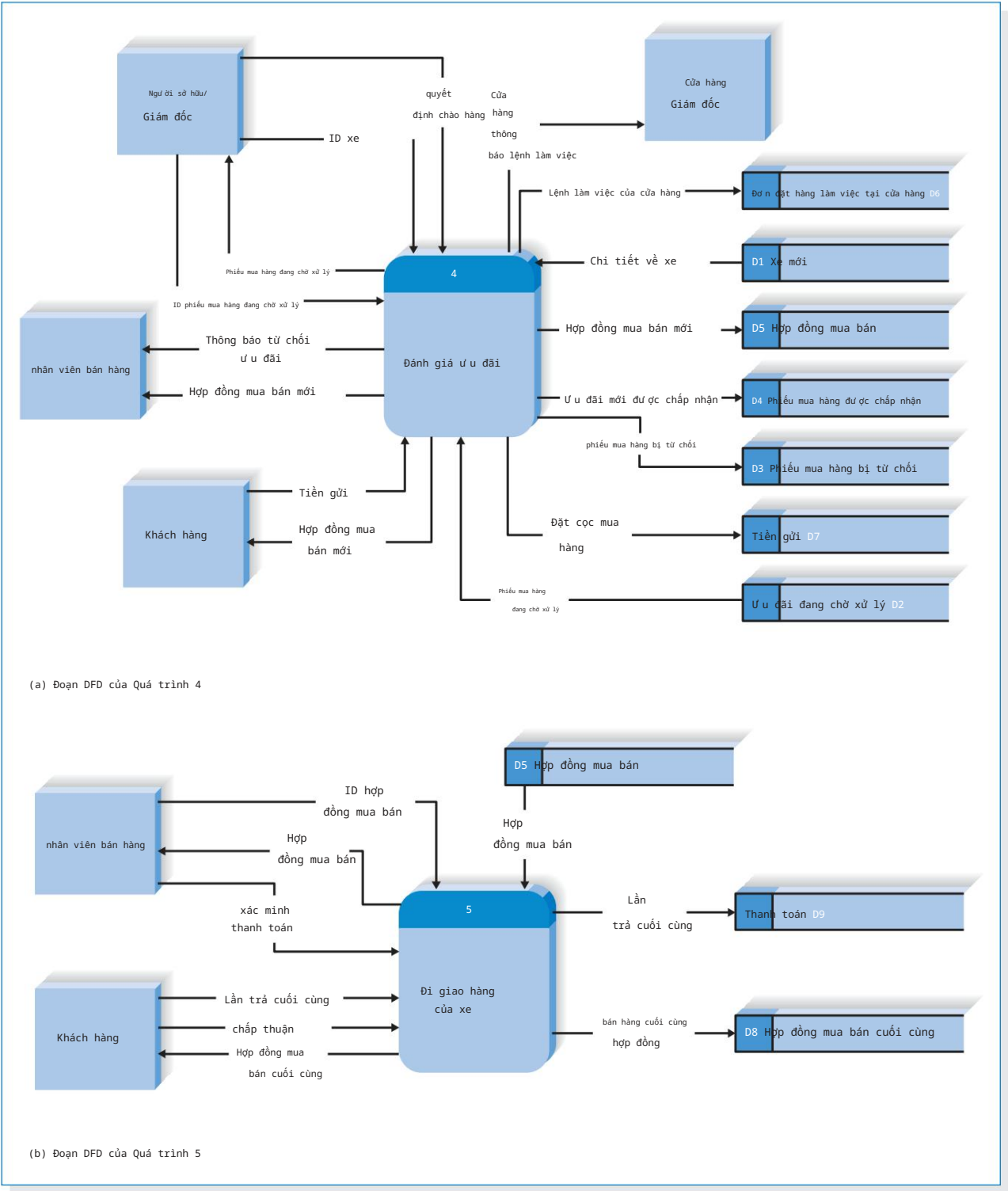
HÌNH 5-6

Phước ng tiện đi lại trong kỳ nghỉ Quy trình 3 (Ưu đãi kỷ lục) Đoạn DFD

bên trái hoặc trên cùng vào quy trình và đầu ra rời khỏi bên phải hoặc dưới cùng.

Kho lưu trữ dữ liệu thường được viết bên dưới quy trình.

Hãy dành một chút thời gian và vẽ một đoạn DFD cho hai trường hợp sử dụng khác được hiển thị trong Hình 4-11 (Đánh giá Ưu Đãi và Nhận Giao xe). Chúng tôi đã bao gồm các cách khả thi để vẽ các mảnh này trong Hình 5-7. (Đừng nhìn cho đến khi bạn đã tự mình vẽ thử!)



HÌNH 5-7

Các mảnh DFD bổ sung cho các phụ dòng tiện du lịch trong kỳ nghỉ

Tạo sơ đồ luồng dữ liệu mức 0 Sau khi bạn có tập

hợp các đoạn DFD (một đoạn cho mỗi trừu tượng hợp sử dụng chính), bạn kết hợp chúng thành một bản vẽ DFD trở thành DFD mức 0. Như đã đề cập trước đó, không có quy tắc bố cục chính thức nào cho DFD. Hầu hết các nhà phân tích hệ thống cố gắng đặt quy trình đầu tiên theo trình tự thời gian ở góc trên bên trái của sơ đồ và thực hiện theo cách của chúng từ trên xuống dưới, từ trái sang phải (ví dụ: Hình 5-1). Nói chung, hầu hết các nhà phân tích cố gắng giảm số lần các dòng dữ liệu giao nhau hoặc để đảm bảo rằng khi chúng giao nhau, chúng giao nhau ở các góc vuông để ít nhầm lẫn hơn. (Nhiều người cho một dòng hơi i "gù" để ám chỉ rằng một luồng dữ liệu nhảy qua dòng kia mà không chạm vào nó.) Việc giảm thiểu số lượng luồng dữ liệu đi qua là một thách thức.

Lập đi lập lại là nền tảng của thiết kế DFD tốt. Ngay cả những nhà phân tích có kinh nghiệm cũng hiếm khi vẽ DFD một cách hoàn hảo ngay lần đầu tiên. Trong hầu hết các trừu tượng hợp, họ vẽ nó một lần để hiểu mô hình quy trình, luồng dữ liệu, kho dữ liệu và các thực thể bên ngoài, sau đó vẽ nó lần thứ hai trên một tờ giấy mới (hoặc trong một tệp mới) để dễ hiểu hơn và để giảm số luồng dữ liệu đi qua. Thông thường, một DFD được vẽ nhiều lần trước khi hoàn thành.

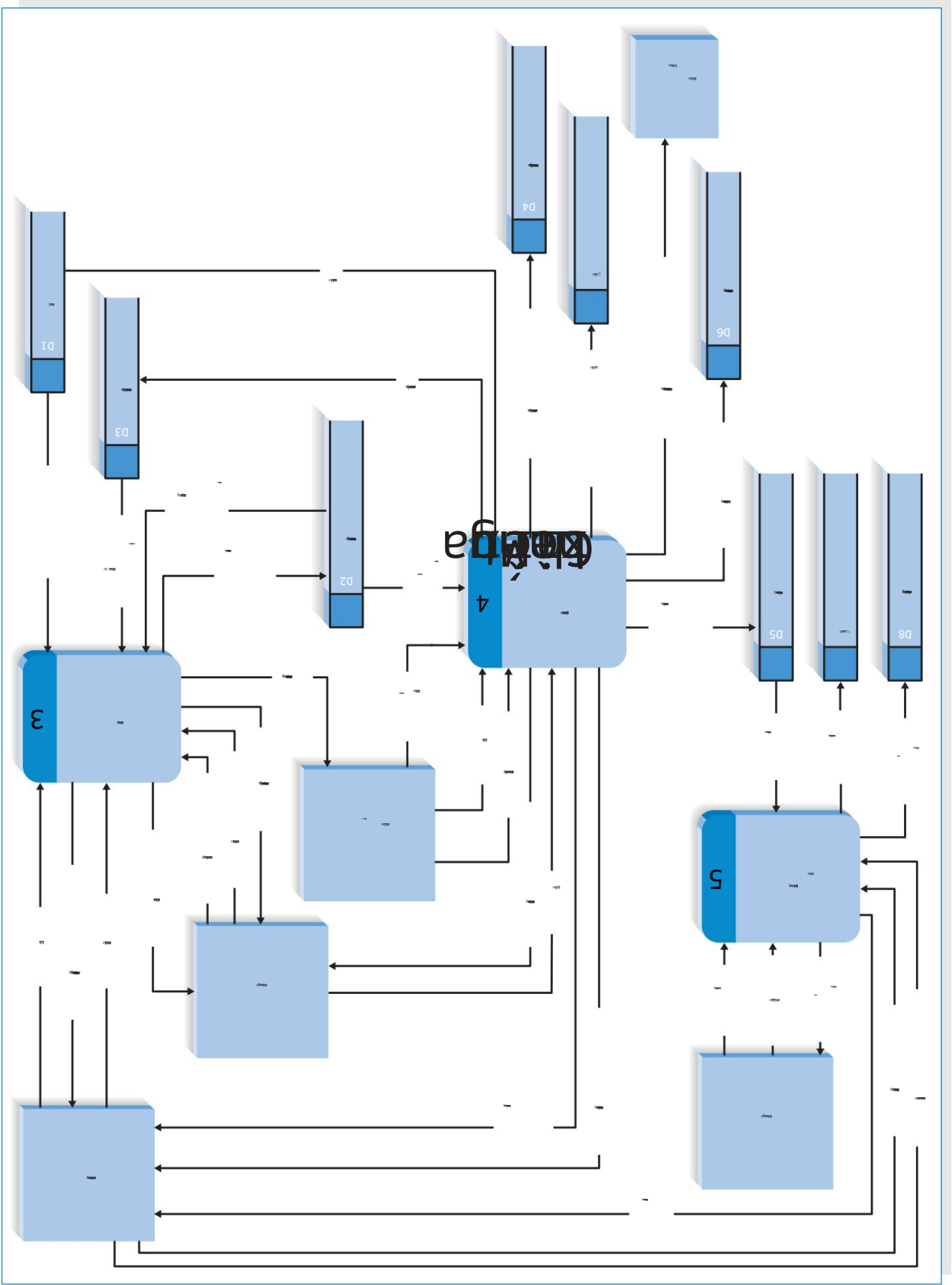
Hình 5-8 kết hợp các đoạn DFD trong Hình 5-6 và 5-7. Hãy dành một chút thời gian để kiểm tra Hình 5-8 và tìm các mảnh DFD từ Hình 5-6 và 5-7 chứa trong đó.

Tạo sơ đồ luồng dữ liệu cấp 1 (và bên dưới)

Bây giờ, nhóm bắt đầu tạo DFD cấp thấp hơn cho từng quy trình trong DFD cấp 0 của DFD cấp 1. Mỗi một trong các trừu tượng hợp sử dụng được chuyển thành DFD của chính nó. Quá trình tạo DFD cấp 1 là thực hiện các bước như được viết trong các trừu tượng hợp sử dụng và chuyển đổi chúng thành DFD theo cách tương tự như đối với DFD cấp 0. Thông thường, mỗi bước chính trong ca sử dụng trở thành một quy trình trên DFD cấp 1, với đầu vào và đầu ra trở thành luồng dữ liệu đầu vào và đầu ra. Tuy nhiên, một lần nữa, đôi khi cần phải có những thay đổi nhỏ để chuyển từ các mô tả không chính thức trong trừu tượng hợp sử dụng sang mô hình quy trình chính thức hơn, chẳng hạn như thêm các luồng dữ liệu đầu vào không có trong trừu tượng hợp sử dụng. Và bởi vì các nhà phân tích hiện đang bắt đầu suy nghĩ sâu hơn về cách các quy trình sẽ được hệ thống thông tin hỗ trợ, nên đôi khi họ thay đổi một chút các bước của ca sử dụng để làm cho quy trình dễ sử dụng hơn.

Trong một số cách tiếp cận để tạo DFD, không có nguồn và đích nào được cung cấp trên DFD cấp 1 (và cấp thấp hơn) cho các đầu vào đến và đi giữa các thực thể bên ngoài (hoặc các quy trình khác bên ngoài quy trình này). Tuy nhiên, nguồn và đích của các luồng dữ liệu cho các kho lưu trữ dữ liệu và các luồng dữ liệu đi đến các quy trình trong DFD này đều được bao gồm (nghĩa là từ bước này sang bước khác trong cùng một trừu tượng hợp sử dụng, chẳng hạn như "Yêu cầu hóa chất đã xác nhận" từ quy trình 2.3 đến 2.5 trong Hình 5-1). Điều này là do thông tin là dư thừa; bạn có thể thấy điểm đến của luồng dữ liệu bằng cách đọc DFD cấp 0.

Vấn đề với các cách tiếp cận này là để thực sự hiểu DFD cấp 1, bạn phải tham khảo lại DFD cấp 0. Đối với các hệ thống nhỏ chỉ có một hoặc hai DFD cấp 1, đây không phải là vấn đề lớn. Nhưng đối với các hệ thống lớn có nhiều cấp độ DFD, vấn đề sẽ tăng lên; để hiểu đích đến của luồng dữ liệu trên DFD cấp 3, bạn phải đọc DFD cấp 2, DFD cấp 1 và DFD cấp 0—và nếu đích đến là một hoạt động khác, thì bạn phải theo dõi các DFD cấp thấp hơn trong quy trình khác.



Chúng tôi tin rằng việc bao gồm các thực thể bên ngoài trong DFD cấp 1 và thấp hơn sẽ đơn giản hóa đáng kể khả năng đọc của DFD, với rất ít nhược điểm. Trong công việc của chúng tôi trong hàng chục dự án với Bộ Quốc phòng Hoa Kỳ, một số cơ quan liên bang khác và quân đội của hai quốc gia khác, chúng tôi đã hiểu được giá trị của phương pháp này và chuyển đổi quyền hạn theo quan điểm của chúng tôi.

Do DFD không được tiêu chuẩn hóa hoàn toàn nên mỗi tổ chức sử dụng chúng hơi khác nhau. Vì vậy, quyết định cuối cùng về việc có đưa các thực thể bên ngoài vào DFD cấp 1 hay không là của bạn—hoặc của người hướng dẫn của bạn! Trong cuốn sách này, chúng tôi sẽ bao gồm chúng.

Lý tưởng nhất là chúng tôi cố gắng giữ các kho lưu trữ dữ liệu ở cùng một vị trí chung trên trang trong DFD cấp 1 giống như trong DFD cấp 0, nhưng điều này không phải lúc nào cũng có thể thực hiện được. Chúng tôi cố gắng vẽ các luồng dữ liệu đầu vào đến từ mép trái của trang và các luồng dữ liệu đầu ra rời khỏi mép phải. Ví dụ, xem DFD mức 1 trong Hình 5-1.

Một trong những câu hỏi thiết kế khó khăn nhất là khi nào thì phân tách DFD mức 1 thành các mức thấp hơn. Việc phân tách DFD có thể được thực hiện ở hầu hết mọi cấp độ, vì vậy, ví dụ: chúng ta có thể phân tách quy trình 1.2 trên DFD cấp độ 1 thành các quy trình 1.2.1, 1.2.2, 1.2.3, v.v. trong DFD cấp độ 2. Điều này có thể được lặp lại với bất kỳ mức độ chi tiết nào, vì vậy người ta có thể có DFD cấp 4 hoặc thậm chí cấp 5.

Không có câu trả lời đơn giản nào cho mức độ phân tách “lý tưởng”, bởi vì nó phụ thuộc vào mức độ phức tạp của hệ thống hoặc quy trình kinh doanh được mô hình hóa. Nói chung, bạn phân tách một quy trình thành một DFD cấp thấp hơn bất cứ khi nào quy trình đó đủ phức tạp để việc phân tách bổ sung có thể giúp giải thích quy trình. Hầu hết các chuyên gia tin rằng nên có ít nhất ba và không nhiều hơn bảy đến chín quy trình trên mỗi DFD, vì vậy nếu bạn bắt đầu phân tách một quy trình và kết thúc chỉ với hai quy trình trên DFD cấp thấp hơn, thì có thể bạn sẽ không cần phải phân hủy nó. Dường như có rất ít điểm trong việc phân tách một quy trình và tạo một DFD cấp thấp hơn chỉ cho hai quy trình; tốt hơn hết là bạn chỉ cần hiển thị hai quy trình trên DFD cấp cao hơn ban đầu. Tư duy tự nhiên vậy, một DFD với hơn chín quy trình trở nên khó đọc và hiểu đối với người dùng vì nó rất phức tạp và đông đúc. Một số quy trình này nên được kết hợp và giải thích trên DFD cấp thấp hơn.

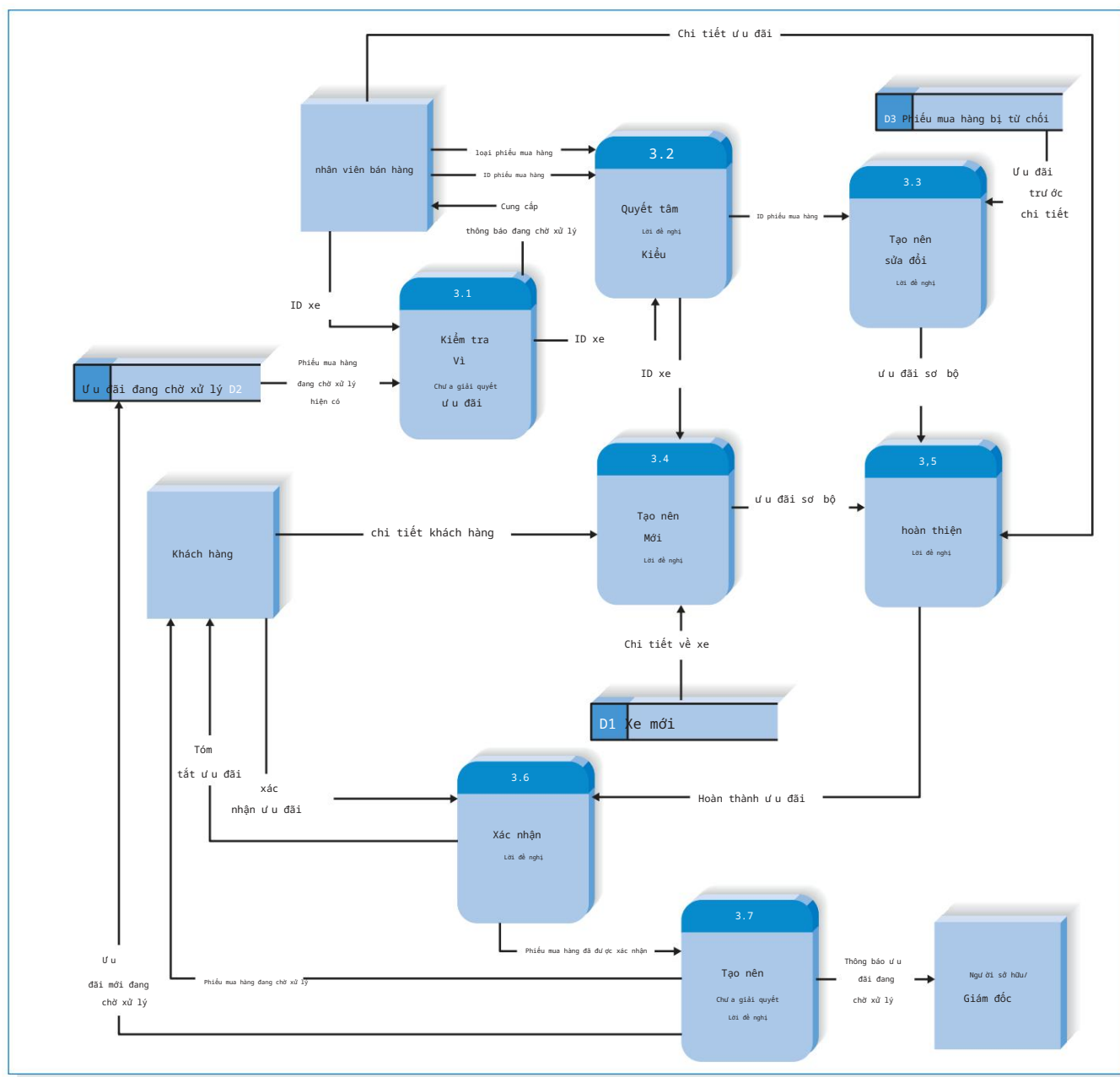
Một hướng dẫn để đạt được mức phân tách lý tưởng là phân tách cho đến khi bạn có thể cung cấp mô tả chi tiết về quy trình trong không quá một trang mô tả quy trình: tiếng Anh có cấu trúc, cây quyết định hoặc bảng quyết định.

Một nguyên tắc ngón tay cái hữu ích khác là mỗi quy trình cấp thấp nhất không được phức tạp hơn những gì có thể nhận ra trong khoảng 25-50 dòng mã.

Trong Hình 5-9, 5-10 và 5-12, chúng tôi đã cung cấp DFD cấp 1 cho các quy trình bán xe mà chúng tôi đã tập trung vào hệ thống Xe du lịch nghỉ dưỡng. Khi chúng tôi mô tả từng hình, hãy dành một chút thời gian để quay lại phần Các bước chính trong các trường hợp sử dụng tư duy ứng của chúng trong Hình 4-11.

Hình 5-9 mô tả DFD cấp 1 cho quy trình Ghi lại Đơn hàng (quy trình 3). Trước tiên, nhân viên bán hàng kiểm tra mọi Ưu đãi đang chờ xử lý hiện có cho xe (3.1). Nếu tìm thấy một phiếu mua hàng đang chờ xử lý, nhân viên bán hàng sẽ được thông báo và quá trình kết thúc. Mặt khác, nhân viên bán hàng được chỉ định nếu ưu đãi là ưu đãi mới hoặc bản sửa đổi của ưu đãi (đã bị từ chối) trước đó (3.2). Nếu đó là phiếu mua hàng đã sửa đổi, thì phiếu mua hàng trước đó sẽ được truy xuất từ kho dữ liệu Phiếu mua hàng bị từ chối (3.3). Nếu đó là một ưu đãi mới, thông tin về phương tiện và khách hàng sẽ được thu thập (3.4). Nhân viên bán hàng hoàn thành các chi tiết cụ thể của ưu đãi (3.5) và ưu đãi phải được khách hàng (3.6) xác nhận.

Sau khi được xác nhận, Ưu đãi đang chờ xử lý được lưu trữ, một bản sao sẽ được cung cấp cho khách hàng và Chủ sở hữu/Người quản lý được thông báo về Ưu đãi đang chờ xử lý mới (3.7). Khi xem xét trường hợp sử dụng cho sự kiện này, bạn sẽ thấy rằng các bước được phác thảo đã được ghi lại trong các

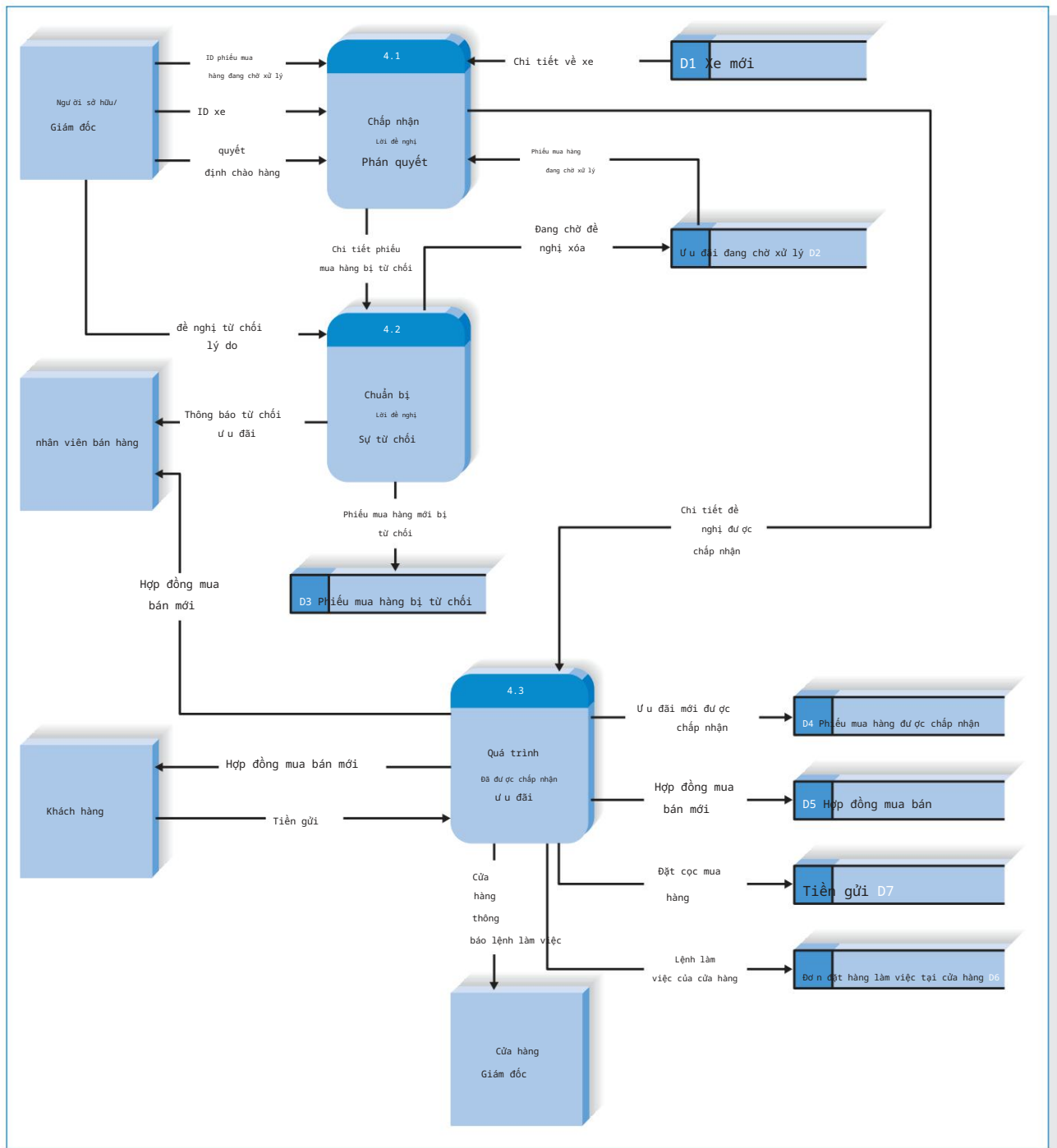


HÌNH 5-9

Quy trình Phục vụ Khách hàng Kỳ nghỉ 3 (Ưu đãi Kỳ nghỉ) Cấp độ 1 DFD

bảy quy trình trong Hình 5-9. Cũng lưu ý rằng chúng tôi đã cố gắng sắp xếp các bước của quy trình hợp sử dụng thành các quy trình tập trung vào một nhiệm vụ chính.

Hình 5-10 mô tả quá trình Đánh giá một Phiếu mua hàng. Chúng tôi đã thực hiện một cách tiếp cận hơi khác để tạo DFD cấp 1 này. Chủ sở hữu/Người quản lý có thể xem Ưu đãi đang chờ xử lý và thông tin chi tiết về chương trình và đưa ra quyết định chấp nhận hoặc từ chối ưu đãi (4.1). Đối với các phiếu mua hàng bị từ chối, Chủ sở hữu/Người quản lý nhập lý do từ chối, phiếu mua hàng bị từ chối được ghi lại và nhân viên bán hàng được thông báo về việc từ chối phiếu mua hàng (4.2). Đối với các đề nghị được chấp nhận, một loạt các nhiệm vụ được



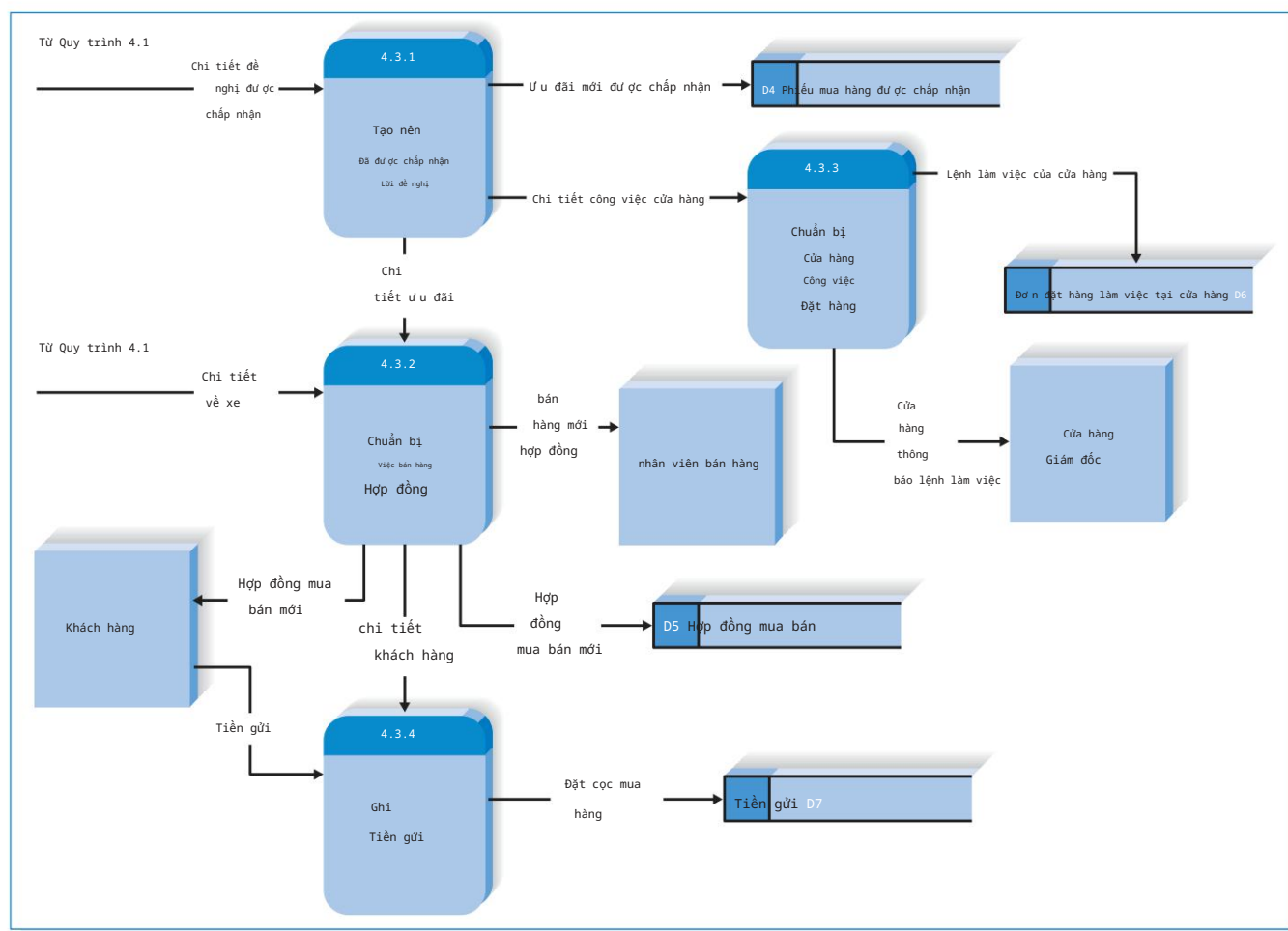
HÌNH 5-10

Phươ ng tiện du lịch nghỉ lễ Quy trình 4 (Đánh giá ưu đãi) DFD cấp 1

được thực hiện để hoàn thành đề nghị được chấp nhận (4.3). Chúng tôi đã cố tình thu thập các nhiệm vụ này vào quy trình 4.3 để chúng tôi có thể chứng minh quy trình “bùng nổ” quy trình đó thành DFD cấp 2.

Quy trình 4.3 trong Hình 5-10 rõ ràng là một quy trình đang thực hiện nhiều nhiệm vụ. Khi một quy trình có nhiều luồng vào và luồng ra, nó là một ứng cử viên sáng giá cho

204 Chương 5 Mô hình hóa quy trình



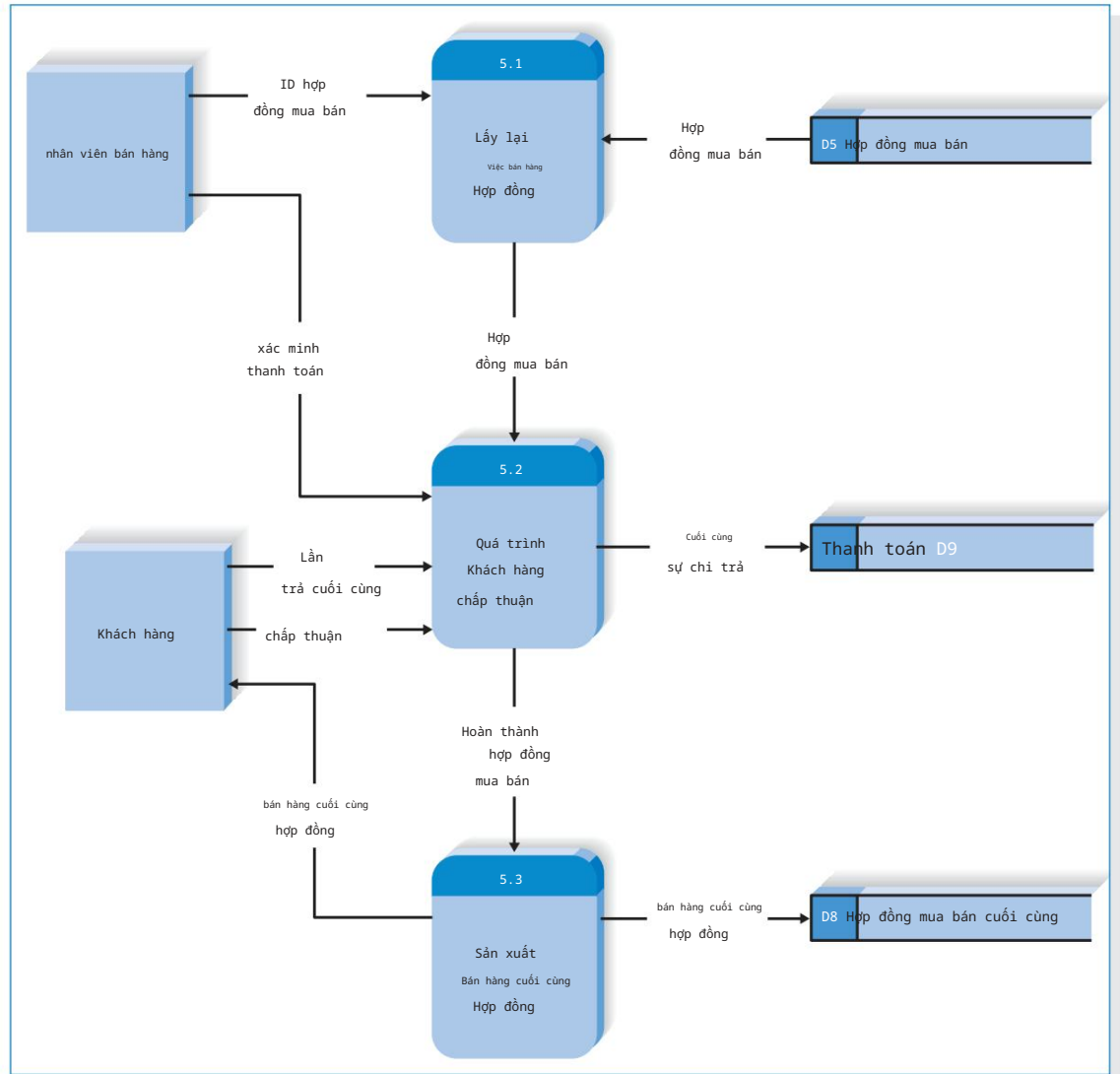
HÌNH 5-11 Quy trình

Phụ lục 4.3 (Quy trình Đề xuất được Chấp nhận) DFD Cấp 2

phân tách thành một DFD cấp thấp hơn. Như chúng tôi đã đề cập trước đây, các nhà phân tích thường cố tình "ẩn" các chi tiết trên sơ đồ cấp cao hơn để loại bỏ sự nhầm lẫn và sử dụng sơ đồ cấp thấp hơn để "tiết lộ" các chi tiết. Hình 5-11 chứa DFD cấp 2 cho quy trình 4.3, Ưu đãi được chấp nhận quy trình. Như bạn có thể thấy khi nghiên cứu Hình 5-11, quy trình gốc được chia thành bốn quy trình con, mỗi quy trình tập trung vào một nhiệm vụ chính. Quy trình 4.3.1 ghi lại Ưu đãi được chấp nhận mới và cung cấp dữ liệu cho hai quy trình khác. Quy trình 4.3.3 sử dụng các chi tiết từ đề nghị được chấp nhận liên quan đến các tùy chọn mà khách hàng muốn chuẩn bị Đơn đặt hàng làm việc tại cửa hàng và thông báo cho người quản lý cửa hàng. Quy trình 4.3.2 sử dụng chi tiết ưu đãi và chi tiết phụ trợ để chuẩn bị hợp đồng mua bán cuối cùng. Cuối cùng, tiền gửi của khách hàng được ghi lại (4.3.4).

Hình 5-12 mô tả trường hợp sử dụng thứ ba của chúng tôi từ Hình 4-11 (Nhận phụ trợ giao hàng). Nhân viên bán hàng lấy Hợp đồng mua bán (5.1). Nhân viên bán hàng xác nhận việc nhận khoản thanh toán cuối cùng của khách hàng, khách hàng ký xác nhận đã chấp nhận phụ trợ tiền và khoản thanh toán được ghi lại (5.2). Cuối cùng, Hợp đồng mua bán cuối cùng được ghi lại và một bản sao được cung cấp cho khách hàng (5.3).

Mô hình quy trình có nhiều khả năng được rút ra ở mức độ chi tiết thấp nhất cho mô hình tương lai nếu sử dụng quy trình phát triển truyền thống (tức là không nhanh chóng).



HÌNH 5-12

Phụ trợ tiện dụng lịch nghỉ lễ Quy trình 5 (Nhận phụ trợ tiện) DFD cấp 1

phát triển ứng dụng [RAD]; xem Chương 2) hoặc nếu hệ thống sẽ được xây dựng bởi một nhà thầu bên ngoài. Nếu không có mức độ chi tiết đầy đủ, có thể khó xác định chính xác trong hợp đồng những gì hệ thống nên làm. Nếu cách tiếp cận RAD, bao gồm nhiều tương tác với người dùng và thử nghiệm là các nguyên mẫu, đang được sử dụng, chúng tôi sẽ ít có khả năng đi đến mức độ chi tiết thấp, bởi vì thiết kế sẽ phát triển thông qua tương tác với người dùng. Theo kinh nghiệm của chúng tôi, hầu hết các hệ thống chỉ ở cấp độ 2 nhiều nhất.

Không có yêu cầu rằng tất cả các phần của hệ thống phải được phân tách thành cùng một mức DFD. Một số phần của hệ thống có thể rất phức tạp và yêu cầu nhiều cấp độ, trong khi các phần khác của hệ thống có thể đơn giản hơn và yêu cầu ít hơn.

CÁC KHÁI NIỆM	5-A QUÂN ĐỘI HOA KỲ VÀ QUÂN ĐOÀN Linh thủy đánh bộ hậu cần chiến trường
TRONG HÀNH ĐỘNG	<div><div><p>Ngay sau Chiến tranh vùng Vịnh năm 1991 (Bão táp sa mạc), Bộ Quốc phòng Hoa Kỳ nhận ra rằng có những vấn đề nghiêm trọng trong hệ thống hậu cần chiến trường cung cấp tiếp tế cho quân đội từ cấp sư đoàn trở xuống. Trong Chiến tranh vùng Vịnh, các đơn vị lục quân và thủy quân lục chiến cùng nhau chiến đấu đã gặp khó khăn trong việc chia sẻ nguồn cung cấp qua lại vì hệ thống máy tính hậu cần của họ không dễ dàng liên lạc. Mục tiêu của hệ thống mới là kết hợp các hệ thống hậu cần của quân đội và thủy quân lục chiến thành một hệ thống để cho phép các đơn vị chia sẻ nguồn cung cấp trong điều kiện chiến trường.</p><p>Quân đội và thủy quân lục chiến đã xây dựng các mô hình quy trình nguyên trạng riêng biệt của các hệ thống hậu cần hiện có của họ, có 165 quy trình cho hệ thống lục quân và 76 quy trình cho thủy quân lục chiến. Cả hai mô hình quy trình đều được phát triển trong khoảng thời gian 3 tháng và tiêu tốn vài triệu đô la để xây dựng, mặc dù chúng không nhằm mục đích toàn diện.</p></div><div><p>Tôi đã giúp họ phát triển một mô hình cho hệ thống hậu cần chiến trường tích hợp mới sẽ được sử dụng bởi cả hai quân chủng (tức là mô hình tư vấn lai). Mô hình quy trình ban đầu chứa 1.500 quy trình và giảm xuống DFD cấp 6 ở nhiều nơi. Phải mất 3.300 trang để in. Họ nhận ra rằng mô hình này quá lớn để có ích. Người lãnh đạo dự án đã quyết định rằng DFD cấp độ 4 là hết mức có thể của mô hình, với thông tin bổ sung có trong các mô tả quy trình. Điều này làm giảm mô hình xuống còn 375 quy trình (800 trang) và làm cho nó hữu ích hơn nhiều.</p><p>Alan Dennis</p></div><div><p>CÂU HỎI:</p><ol style="list-style-type: none">Ưu điểm và nhược điểm của việc đặt giới hạn độ sâu tối đa cho DFD là gì?DFD cấp 4 có phải là giới hạn phù hợp không?</div></div>

Xác thực sơ đồ luồng dữ liệu Sau khi

bạn đã tạo một bộ DFD, điều quan trọng là phải kiểm tra chất lượng của chúng. Hình 5-13 cung cấp một danh sách kiểm tra nhanh để xác định các lỗi phổ biến nhất. Có hai loại vấn đề cơ bản khác nhau có thể xảy ra trong DFD: lỗi cú pháp và lỗi ngữ nghĩa. "Cú pháp" đề cập đến cấu trúc của DFD và liệu DFD có tuân theo các quy tắc của ngôn ngữ DFD hay không. Lỗi cú pháp có thể được coi là lỗi ngữ pháp do nhà phân tích mắc phải khi họ tạo DFD. "Ngữ nghĩa" đề cập đến ý nghĩa của DFD và liệu chúng có mô tả chính xác quy trình kinh doanh được mô hình hóa hay không. Lỗi ngữ nghĩa có thể được coi là sự hiểu lầm của nhà phân tích trong việc thu thập, phân tích và báo cáo thông tin về hệ thống.

Nói chung, các lỗi cú pháp dễ tìm và sửa hơn các lỗi ngữ nghĩa, bởi vì có các quy tắc rõ ràng có thể được sử dụng để xác định chúng (ví dụ: một quy trình phải có tên). Hầu hết các công cụ CASE đều có trình kiểm tra cú pháp sẽ phát hiện lỗi trong một trang của DFD giống như cách mà trình xử lý văn bản có trình kiểm tra chính tả và trình kiểm tra ngữ pháp. Tìm các lỗi cú pháp kéo dài nhiều trang của DFD (ví dụ: từ DFD cấp 1 đến DFD cấp 2) khó hơn một chút, đặc biệt đối với quan điểm nhất quán, phân tách và cân bằng. Một số công cụ CASE có thể phát hiện lỗi số dư, nhưng đó chỉ là tất cả. Trong hầu hết các trường hợp, các nhà phân tích phải xem xét cẩn thận và tỉ mỉ mọi quy trình, thực thể bên ngoài, luồng dữ liệu và lưu trữ dữ liệu trên tất cả các DFD để đảm bảo rằng họ có quan điểm nhất quán và sự phân tách và cân bằng là phù hợp.

Mỗi kho lưu trữ dữ liệu được yêu cầu phải có ít nhất một đầu vào và một đầu ra trên một số trang của DFD. Trong Hình 5-10, kho dữ liệu D1, Xe mới, chỉ có đầu ra và một số kho dữ liệu chỉ có đầu vào. Tình huống này không nhất thiết là một lỗi. Nhà phân tích nên kiểm tra những nơi khác trong DFD để tìm nơi dữ liệu được ghi vào dữ liệu

cú pháp	
Trong DFD	
Quá trình	<ul style="list-style-type: none">• Mọi quy trình đều có một tên duy nhất là một cụm động từ hướng hành động, một con số và một mô tả.• Mọi tiến trình đều có ít nhất một luồng dữ liệu đầu vào.• Mọi tiến trình đều có ít nhất một luồng dữ liệu đầu ra.• Luồng dữ liệu đầu ra thường có tên khác với luồng dữ liệu đầu vào do quá trình thay đổi đầu vào thành một đầu ra khác theo một cách nào đó.• Có từ ba đến bảy quy trình trên mỗi DFD.
Dòng dữ liệu	<ul style="list-style-type: none">• Mỗi luồng dữ liệu có một tên duy nhất là danh từ và mô tả.• Mọi luồng dữ liệu đều kết nối với ít nhất một tiến trình.• Dữ liệu chỉ chảy theo một hướng (không có mũi tên hai đầu).• Số lượng tối thiểu các đường luồng dữ liệu giao nhau.
Kho dữ liệu	<ul style="list-style-type: none">• Mỗi kho dữ liệu có một tên duy nhất là danh từ và mô tả.• Mỗi kho dữ liệu có ít nhất một luồng dữ liệu đầu vào (có nghĩa là thêm dữ liệu mới hoặc thay đổi dữ liệu hiện có trong kho dữ liệu) trên một số trang của DFD.• Mỗi kho dữ liệu có ít nhất một luồng dữ liệu đầu ra (có nghĩa là đọc dữ liệu từ kho dữ liệu) trên một số trang của DFD.• Mọi thực thể bên ngoài đều có một tên riêng là danh từ và mô tả.• Mỗi thực thể bên ngoài có ít nhất một luồng dữ liệu đầu vào hoặc đầu ra.
Thực thể bên ngoài	
Trên các DFD	
Sơ đồ ngữ cảnh	<ul style="list-style-type: none">• Mỗi bộ DFD phải có một sơ đồ ngữ cảnh.• Có quan
Quan điểm	<ul style="list-style-type: none">điểm nhất quán đối với toàn bộ các DFD.• Mọi quy trình được
phân hủy	<ul style="list-style-type: none">mô tả toàn bộ và đầy đủ bởi các quy trình trên các DFD con của nó.• Mọi luồng dữ liệu, kho
Sự cân bằng	<ul style="list-style-type: none">lưu trữ dữ liệu và thực thể bên ngoài trên DFD cấp cao hơn được hiển thị trên DFD cấp thấp hơn sẽ phân tách nó.
ngữ nghĩa	
Đại diện phù hợp	<ul style="list-style-type: none">• Xác thực ngữ cảnh• Quy
	<ul style="list-style-type: none">trình nhập vai• Kiểm
phân hủy nhất quán	<ul style="list-style-type: none">tra DFD cấp thấp nhất• Kiểm
Thuật ngữ nhất quán	<ul style="list-style-type: none">tra tên cẩn thận

HÌNH 5-13

Sơ đồ luồng dữ liệu Danh sách kiểm tra chất lượng

Lưu trữ D2 hoặc đọc từ các kho lưu trữ dữ liệu khác. Tất cả các kho lưu trữ dữ liệu nên có ít nhất một luồng vào và một luồng ra, nhưng các luồng có thể không nằm trên cùng một sơ đồ, vì vậy hãy kiểm tra các phần khác của hệ thống. Một vấn đề khác phát sinh là khi kho lưu trữ dữ liệu được sử dụng bởi các hệ thống khác. Trong trường hợp đó, dữ liệu có thể được thêm vào hoặc sử dụng bởi một hệ thống riêng biệt. Điều này hoàn toàn ổn, nhưng nhà phân tích nên điều tra để xác minh rằng các luồng vào và ra khỏi kho lưu trữ dữ liệu cần thiết tồn tại ở đâu đó.

Theo kinh nghiệm của chúng tôi, lỗi cú pháp phổ biến nhất mà các nhà phân tích mới làm quen mắc phải khi tạo DFD là vi phạm luật bảo toàn dữ liệu.3 Phần đầu tiên của luật nêu rõ như sau:

1. Dữ liệu ở trạng thái nghỉ sẽ ở trạng thái nghỉ cho đến khi được di chuyển bởi một tiến trình.

3 Luật này được phát triển bởi Giáo sư Dale Goodhue tại Đại học Georgia.

Nói cách khác, dữ liệu không thể di chuyển mà không có quá trình. Dữ liệu không thể đi đến hoặc đến từ một kho lưu trữ dữ liệu hoặc một thực thể bên ngoài mà không có một quy trình để đẩy hoặc kéo nó. Phần thứ hai của luật quy định như sau:

2. Các quy trình không thể sử dụng hoặc tạo dữ liệu.

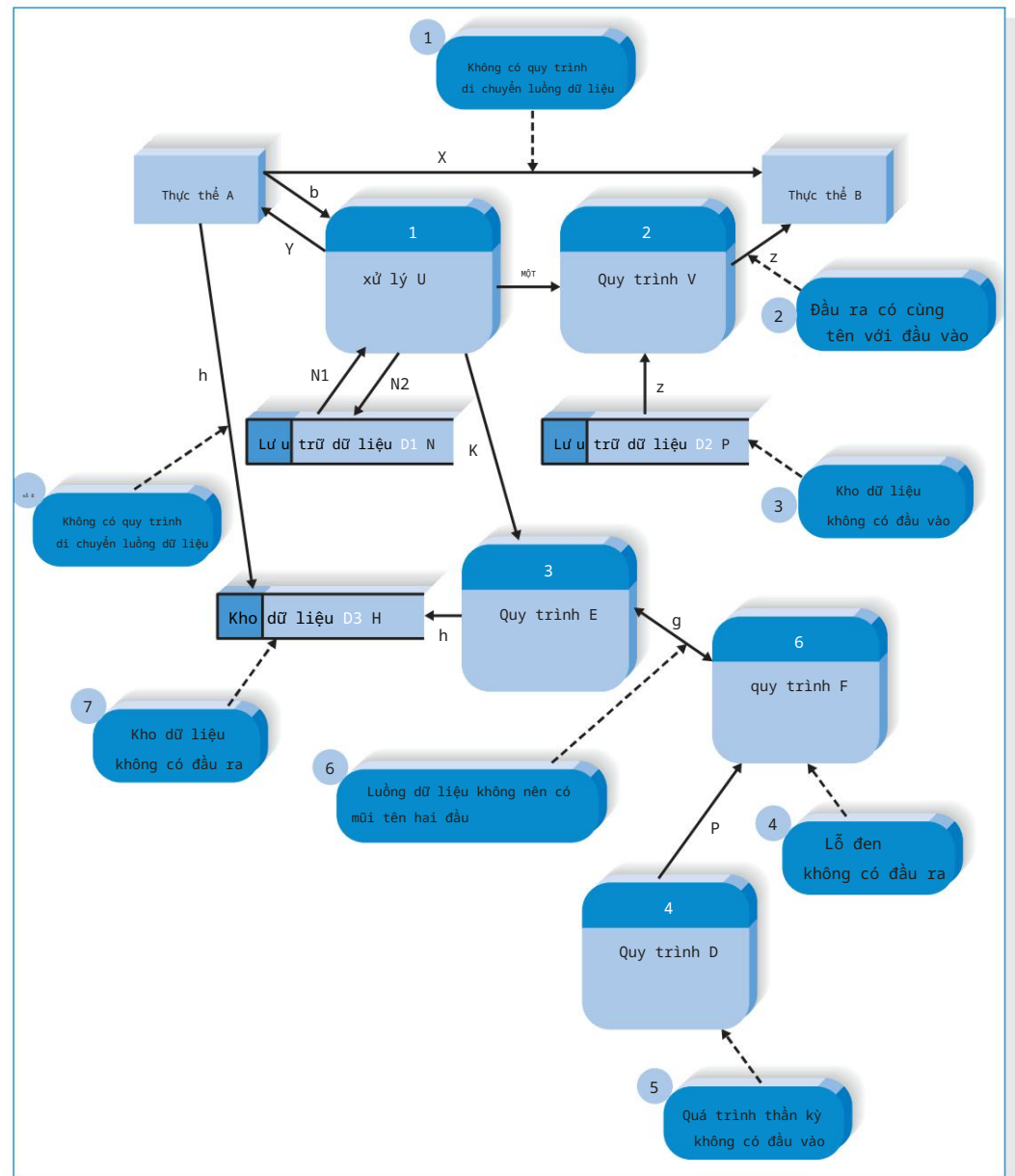
Nói cách khác, dữ liệu chỉ vào hoặc ra khỏi hệ thống thông qua các thực thể bên ngoài. Một quy trình không thể hủy dữ liệu đầu vào; tất cả các quá trình phải có đầu ra. Về một quy trình không có đầu ra đôi khi được gọi là lỗi “lỗ đen”. Tư duy tự nhiên như vậy, một quy trình không thể tạo dữ liệu mới; nó có thể biến đổi dữ liệu từ dạng này sang dạng khác, nhưng nó không thể tạo ra dữ liệu đầu ra nếu không có đầu vào. Về một quy trình không có đầu vào đôi khi được gọi là lỗi “kỳ diệu” (vì dữ liệu đầu ra xuất hiện một cách kỳ diệu). Có một ngoại lệ đối với phần luật yêu cầu đầu vào, nhưng nó hiếm đến mức hầu hết các bộ phân tích phân tích không bao giờ gặp phải nó. Hình 5-14 cho thấy một số lỗi cú pháp phổ biến.

Nhìn vào Hình 5-14, chúng ta sẽ thảo luận lần lượt từng lỗi. Đầu tiên, chúng ta có thể thấy luồng dữ liệu X được vẽ trực tiếp từ Thực thể A đến Thực thể B. Hãy nhớ rằng luồng dữ liệu phải bắt đầu hoặc kết thúc tại một quy trình; do đó, một quá trình là cần thiết. Thứ hai, chúng ta thấy luồng dữ liệu Z được lấy từ Kho dữ liệu P và được gửi đến Thực thể B. Các quy trình nên tồn tại để chuyển đổi dữ liệu theo một cách nào đó, vì vậy chúng tôi thường sửa đổi tên luồng dữ liệu để phản ánh những thay đổi được thực hiện trong quy trình. Thứ ba, chúng ta thấy rằng Data Store P có đầu ra nhưng không có đầu vào. Đây không nhất thiết là một lỗi, nhưng đáng để nhà phân tích điều tra. Chúng ta nên đảm bảo rằng một quy trình bổ sung dữ liệu vào Kho lưu trữ dữ liệu P tồn tại ở đâu đó trong sơ đồ của toàn bộ mô hình quy trình. Thứ tư, chúng ta có thể thấy rằng Process F nhận dữ liệu nhưng không có đầu ra. Đây được coi là một lỗ đen vì dữ liệu được nhận nhưng không có gì được tạo ra. Thứ năm, Quy trình D được hiển thị tạo ra một luồng dữ liệu nhưng không có đầu vào. Đây được gọi là một quá trình kỳ diệu. Thứ sáu, chúng ta thấy một mũi tên hai đầu mô tả Luồng dữ liệu G giữa Quy trình E và Quy trình F. Các luồng dữ liệu không nên được vẽ theo cách này mà chỉ nên chảy theo một hướng. Thứ bảy, Kho lưu trữ dữ liệu H nhận Luồng dữ liệu H làm đầu vào, nhưng không có đầu ra. Vấn đề này có thể không phải là một lỗi, nhưng nhà phân tích nên theo dõi để đảm bảo rằng dữ liệu được lưu trữ trong Kho lưu trữ dữ liệu H được sử dụng ở một nơi nào đó trong mô hình quy trình; nếu không, không có lý do gì để lưu trữ nó. Cuối cùng, chúng tôi thấy rằng một quy trình nên được tham gia giữa Thực thể A và Kho lưu trữ dữ liệu H.

Nói chung, lỗi ngữ nghĩa gây ra nhiều vấn đề nhất trong quá trình phát triển hệ thống. Lỗi ngữ nghĩa khó tìm và sửa hơn nhiều vì làm như vậy đòi hỏi sự hiểu biết tốt về quy trình kinh doanh. Và thậm chí sau đó, những gì có thể được xác định là lỗi thực sự có thể là sự hiểu lầm của người xem xét mô hình. Có ba kiểm tra hữu ích để giúp đảm bảo rằng các mô hình về mặt ngữ nghĩa là chính xác. (Xem Hình 5-13.)

Kiểm tra đầu tiên để đảm bảo rằng mô hình là một đại diện phù hợp là yêu cầu người dùng xác nhận tính hợp lệ của mô hình khi xem qua (nghĩa là mô hình được trình bày cho người dùng và họ kiểm tra độ chính xác của mô hình). Một kỹ thuật hiệu quả hơn là để người dùng nhập vai quy trình từ DFD giống như cách mà họ đóng vai trong trường hợp sử dụng. Người dùng giả vờ thực hiện quy trình chính xác như được mô tả trong DFD. Họ bắt đầu ở quy trình đầu tiên và cố gắng thực hiện nó bằng cách chỉ sử dụng các đầu vào được chỉ định và chỉ tạo ra các đầu ra được chỉ định. Sau đó, họ chuyển sang quy trình thứ hai, v.v.

4 Ngoại lệ là một quy trình tạm thời đưa ra đầu ra kích hoạt dựa trên đồng hồ thời gian bên trong. Bất cứ khi nào một khoảng thời gian xác định trôi qua, quy trình sẽ tạo ra một đầu ra. Quy trình chấm công không có đầu vào vì đồng hồ nằm trong quy trình.



HÌNH 5-14

Một số lỗi thường gặp

Một trong những dạng lỗi ngữ nghĩa tinh vi nhất xảy ra khi một quá trình tạo ra một đầu ra, nhưng không có đủ đầu vào để tạo ra nó. Ví dụ, để tạo ra nước (H_2O), chúng ta cần có cả hydro (H) và oxy (O). Điều này cũng đúng với các hệ thống máy tính, trong đó đầu ra của một quy trình chỉ có thể là sự kết hợp và biến đổi của các đầu vào của nó. Ví dụ, giả sử chúng ta muốn ghi lại một đơn đặt hàng; chúng tôi cần tên khách hàng và địa chỉ gửi thư cũng như số lượng và giá cả cho các mặt hàng mà khách hàng đang đặt hàng. Chúng tôi cần thông tin từ kho dữ liệu khách hàng (ví dụ: địa chỉ) và thông tin từ kho dữ liệu mặt hàng (ví dụ: giá). Chúng ta không thể vẽ một quy trình tạo ra luồng dữ liệu thứ tự đầu ra mà không có đầu vào từ hai kho dữ liệu này. Nhập vai với sự tuân thủ nghiêm ngặt các đầu vào và đầu ra trong một mô hình là một trong những cách tốt nhất để phát hiện loại lỗi này.

Kiểm tra lỗi ngữ nghĩa thứ hai là để đảm bảo phân tách nhất quán, có thể được kiểm tra bằng cách kiểm tra các quy trình cấp thấp nhất trong DFD. Trong hầu hết các trường hợp, tất cả các quy trình nên được phân tách thành cùng một mức độ chi tiết—điều này không giống với việc nói cùng một số cấp độ. Ví dụ: giả sử rằng chúng ta đang lập mô hình quá trình lái xe đi làm vào buổi sáng. Một mức độ chi tiết sẽ là nói như sau: (1) Nhập ô tô; (2) khởi động xe; (3) lái xe đi.

Một mức độ chi tiết khác có thể nói như sau: (1) Mở khóa xe; (2) ngồi trên xe; (3) thắt dây an toàn; và như thế. Vẫn còn một cấp độ khác là nói như sau: (1) Lấy chìa khóa ra khỏi túi; (2) tra chìa khóa vào ổ khóa cửa; (3) chìa khóa xoay; và như thế.

Không cái nào trong số này tốt hơn cái nào, nhưng trừ những trường hợp bắt buộc, tốt nhất là đảm bảo rằng tất cả các quy trình ở dư ới cùng của mô hình đều cung cấp cùng một mức độ chi tiết nhất quán.

Tương tự như vậy, điều quan trọng là phải đảm bảo rằng thuật ngữ nhất quán trong suốt mô hình. Cùng một mặt hàng có thể có các tên khác nhau ở các bộ phận khác nhau của tổ chức, vì vậy “đơn đặt hàng” của một người có thể là “đơn đặt hàng của khách hàng” của một người khác. Tương tự như vậy, cùng một thuật ngữ có thể có nghĩa khác nhau; ví dụ: “ngày giao hàng” có thể có nghĩa là một điều đối với đại diện bán hàng nhận đơn đặt hàng (ví dụ: ngày đã hứa) và một điều gì đó khác đối với nhà kho (ví dụ: ngày giao hàng thực tế). Việc giải quyết những khác biệt này trước khi mô hình được hoàn thiện là rất quan trọng để đảm bảo rằng tất cả những người đọc mô hình hoặc những người sử dụng hệ thống thông tin được xây dựng từ mô hình đều có sự hiểu biết chung.

ÁP DỤNG CÁC KHÁI NIỆM TẠI TUNE NGUỒN

Tạo sơ đồ ngữ cảnh Nhóm dự án bắt đầu

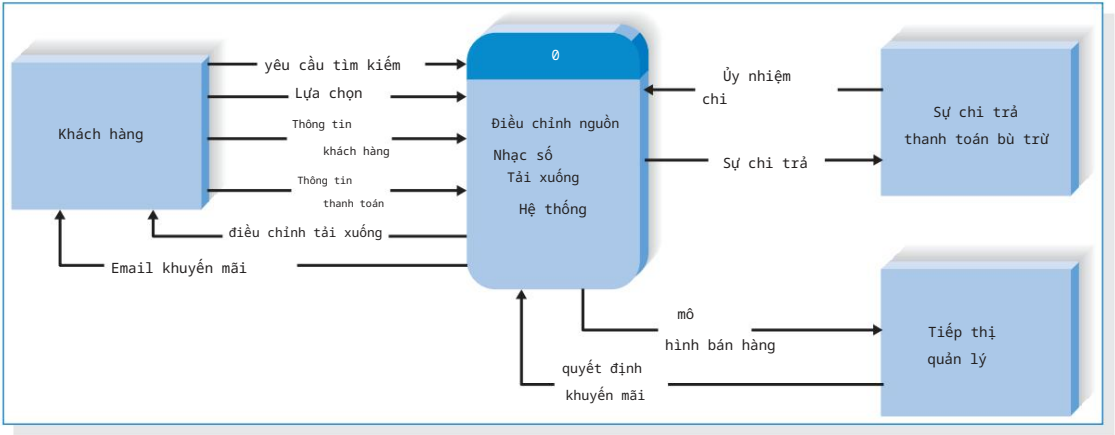
bằng cách tạo sơ đồ ngữ cảnh. Họ đọc qua khu vực tổng hợp của ba trường hợp sử dụng chính trong Hình 4-14 để tìm các đầu vào và đầu ra chính.

Nhóm nhận thấy rằng phần lớn các tương tác luồng dữ liệu là với những khách hàng đang sử dụng trang Web để duyệt qua các lựa chọn âm nhạc và thực hiện các giao dịch mua tải xuống. Sẽ có sự tương tác với tổ chức thanh toán bù trừ sẽ xử lý việc xác minh thanh toán và xử lý các giao dịch mua. Cuối cùng, mặc dù không rõ ràng từ các trường hợp sử dụng, nhưng các nhà quản lý tiếp thị sẽ sử dụng thông tin bán hàng từ hệ thống để thiết kế và thực hiện các chiến dịch quảng cáo. Nhóm đã sử dụng các luồng vào và luồng ra chính từ các trường hợp sử dụng và phát triển sơ đồ ngữ cảnh như trong Hình 5-15.

Tạo các đoạn sơ đồ luồng dữ liệu Bức tiếp theo

là tạo một đoạn DFD cho mỗi trường hợp sử dụng. Điều này được thực hiện bằng cách vẽ quy trình ở giữa trang, đảm bảo rằng số và tên quy trình là phù hợp, đồng thời kết nối tất cả các luồng dữ liệu đầu vào và đầu ra với nó. Không giống như sơ đồ ngữ cảnh, đoạn DFD bao gồm các luồng dữ liệu đến các thực thể bên ngoài và các kho lưu trữ dữ liệu bên trong.

Các đoạn DFD đã hoàn thành được hiển thị trong Hình 5-16. Trước khi nhìn vào hình, hãy dành một phút và tự vẽ chúng. Có nhiều cách tốt để vẽ những mảnh này. Trên thực tế, có nhiều cách “đúng” để tạo các trường hợp sử dụng và DFD. Lưu ý rằng trên đoạn DFD cho quy trình 3, chúng tôi đã hiển thị một luồng vào theo đúng chấm có nhãn “Thời gian để xác định các khuyến mãi” vào quy trình. Nhớ lại rằng chúng ta



HÌNH 5-15

Điều chỉnh sơ đồ ngữ cảnh nguồn

đã chỉ định rằng trữ ứng hợp sử dụng, Quảng cáo Giai điệu, là trữ ứng hợp sử dụng tạm thời, được kích hoạt khi đến lúc cập nhật các chương trình khuyến mãi và ưu đãi đặc biệt. Luồng dữ liệu chậm chạp vào quy trình 3 trong Hình 5-16 đôi khi được gọi là luồng điều khiển và trữ ứng hợp sử dụng để biểu thị trình kích hoạt dựa trên thời gian cho một sự kiện.

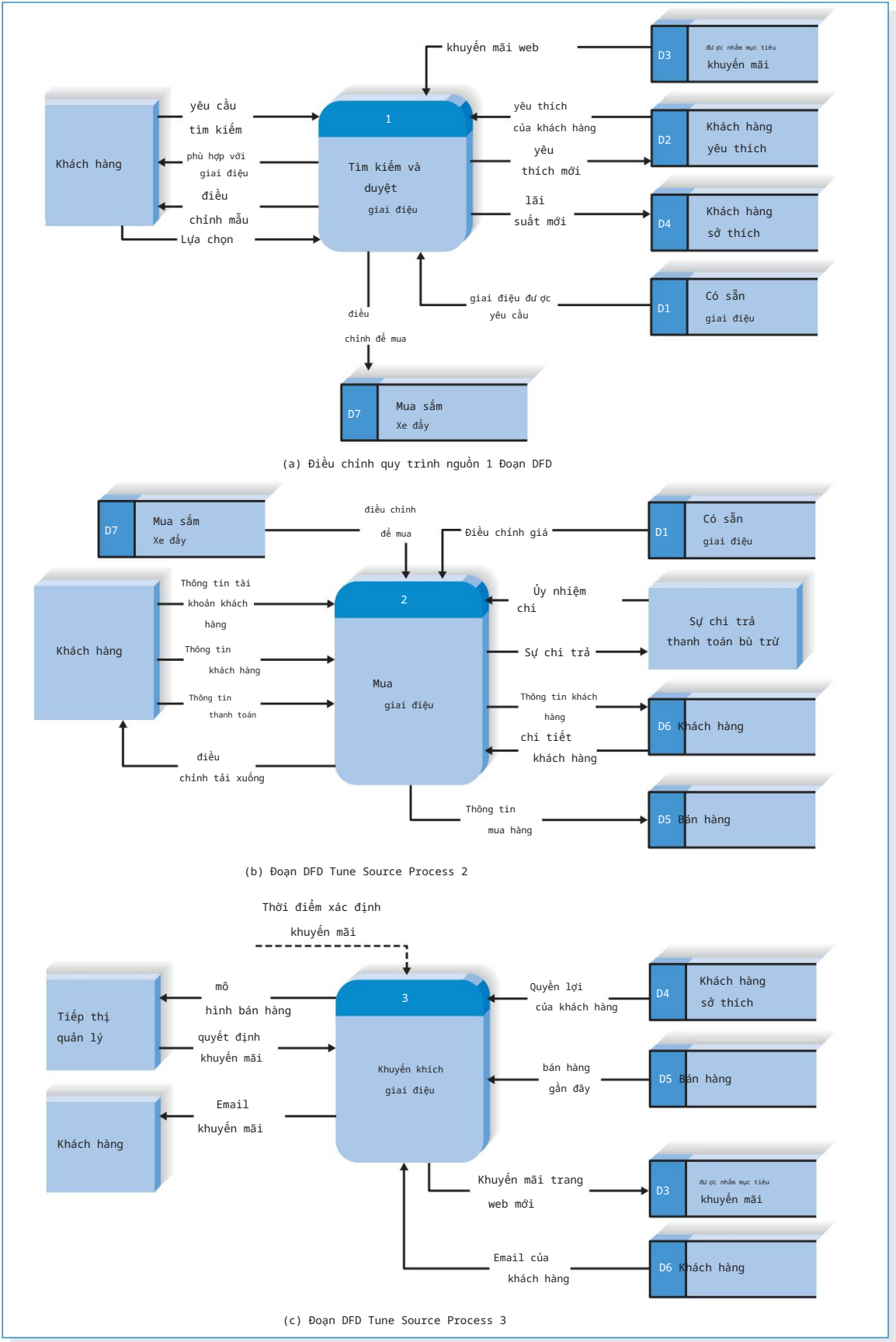
Tạo sơ đồ luồng dữ liệu cấp 0

Bước tiếp theo là tạo DFD cấp 0 bằng cách tích hợp các mảnh DFD, điều này đã được chứng minh là phản khi hậu. Nhóm chỉ cần lấy các mảnh DFD và vẽ chúng lại với nhau trên một tờ giấy. Mặc dù đôi khi rất khó để sắp xếp tất cả các mảnh DFD trên một tờ giấy, nhưng về cơ bản, đó là một bài tập cơ học (Hình 5-17). So sánh sơ đồ mức 0 với sơ đồ ngữ cảnh trong Hình 5-15. Hai DFD có cân bằng không? Lưu ý các chi tiết bổ sung có trong sơ đồ mức 0.

Việc xem xét cẩn thận các kho lưu trữ dữ liệu trong Hình 5-17 cho thấy rằng mọi kho dữ liệu đều có cả luồng vào và luồng ra, với một ngoại lệ, D1: Giai điệu có sẵn. D1:Các giai điệu khả dụng được đọc bởi hai quy trình, nhưng không được ghi vào bởi bất kỳ quy trình nào được hiển thị. Việc vi phạm cú pháp DFD này cần được nhóm điều tra vì đây có thể là một sự giám sát nghiêm trọng. Như chúng tôi sẽ giải thích sau, trong tình huống này, chúng tôi cần tạo một quy trình cụ thể để thêm, sửa đổi và xóa dữ liệu trong D1: Giai điệu có sẵn. Các quy trình "Quản trị" như thế này trữ ứng bị bỏ qua, vì ban đầu chúng tôi chỉ tập trung vào các yêu cầu nghiệp vụ, nhưng sẽ cần được bổ sung trữ ứng khi hệ thống hoàn thiện.

Tạo sơ đồ luồng dữ liệu cấp 1 (và bên dư ới)

Bước tiếp theo là tạo các DFD cấp 1 cho những quy trình có thể hưởng lợi từ chúng. Các nhà phân tích bắt đầu với trữ ứng hợp sử dụng đầu tiên (tìm kiếm và duyệt các giai điệu) và bắt đầu vẽ DFD cho các bước riêng lẻ trong đó. Các bước trong trữ ứng hợp sử dụng rất đơn giản, nhưng thông trữ ứng, nhóm phải chọn tên và số cho các quy trình và thêm các luồng dữ liệu đầu vào từ các kho lưu trữ dữ liệu không có.



HÌNH 5-16

Điều chỉnh nguồn DFD

mảnh vỡ



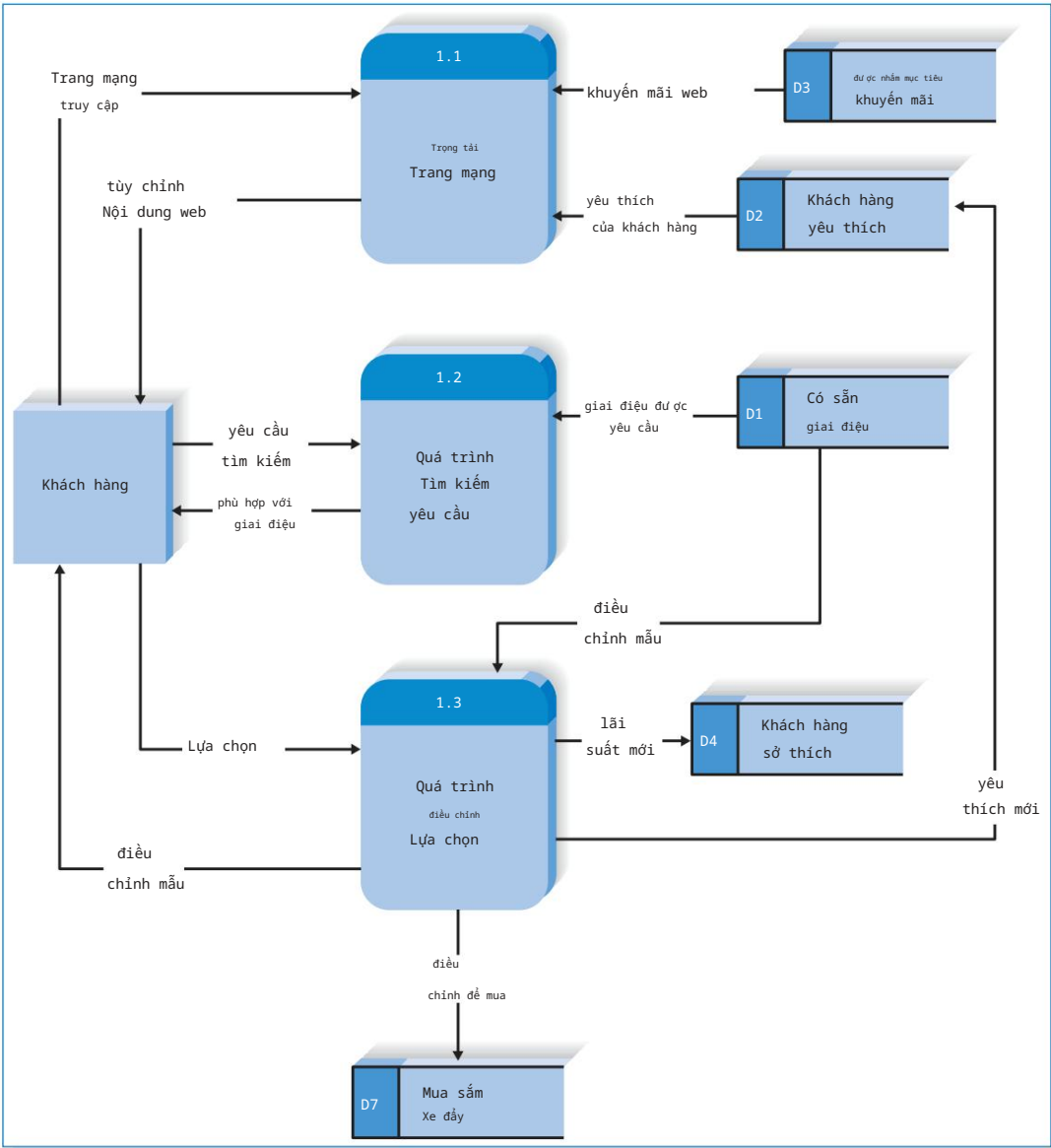
Nhóm cũng đã phát triển sơ đồ cấp 1 cho các quy trình khác, sử dụng các bước chính được nêu trong các trụ đồng hợp sử dụng tư duy ứng. Một số điều chỉnh đã được thực hiện từ các bước như được hiển thị trong các trụ đồng hợp sử dụng, như nhóm đã tuân theo các bước khá chặt chẽ. Xem Hình 5-19 và 5-20 và so sánh chúng với các trụ đồng hợp sử dụng được minh họa trong Hình 4-14.

BẠN R

5-1 KHU VỰC NHÀ Ở

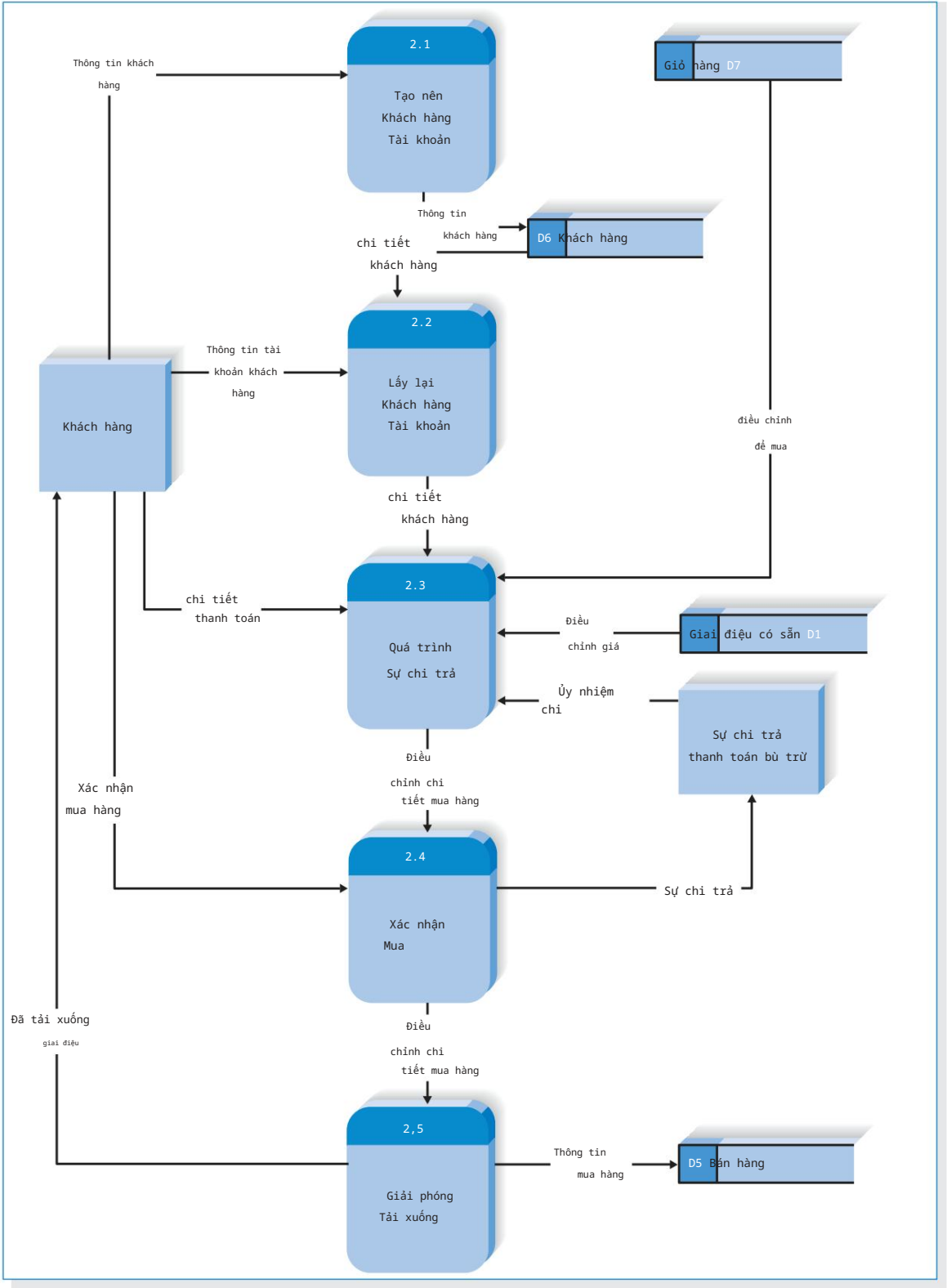
XOAY

Vẽ một sơ đồ ngữ cảnh, một DFD cấp 0 và một tập hợp các DFD cấp 1 (nếu cần) cho các trường hợp sử dụng nhà ở campus mà bạn đã phát triển cho hộp Lướt 4-1 của bạn trong Chương 4.

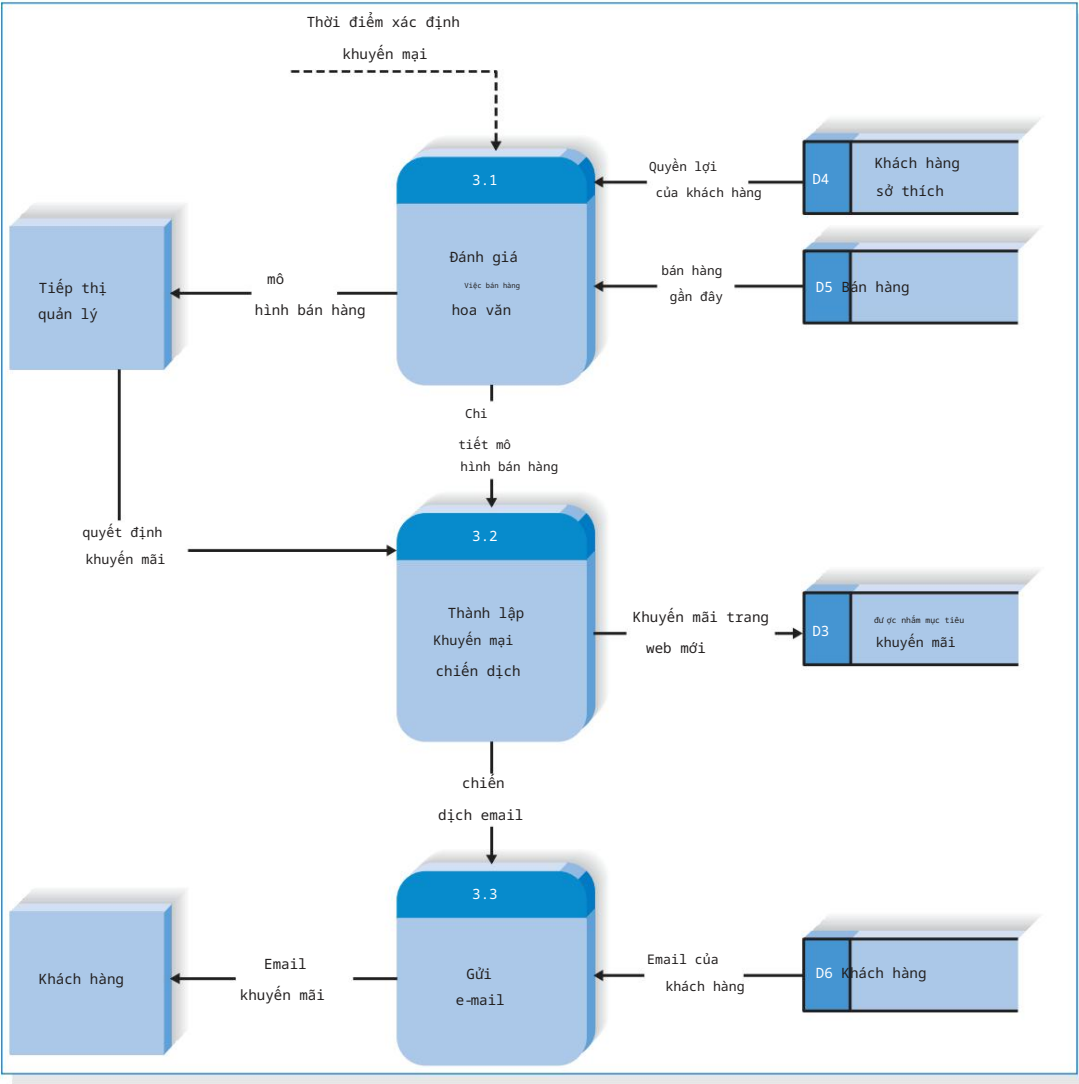


HÌNH 5-18

DFD cấp 1 cho nguồn giai điệu Quá trình 1: Tìm kiếm và duyệt các giai điệu



HÌNH 5-19
DFD cấp 1 cho nguồn giai điệu Quy trình 2: Mua giai điệu



HÌNH 5-20
DFD cấp 1 cho nguồn giai điệu Quy trình 3: Quảng cáo giai điệu

Như chúng tôi đã chỉ ra trong Hình 5-2, mỗi kho lưu trữ dữ liệu nên có một hoặc nhiều luồng dữ liệu đầu vào và một hoặc nhiều luồng dữ liệu đầu ra. Tuy nhiên, khi xem xét cẩn thận Hình 5-18, bạn sẽ thấy rằng D1: Giai điệu khả dụng có các luồng dữ liệu đầu ra, nhưng không có luồng dữ liệu đầu vào nào. Kho lưu trữ dữ liệu này là kho lưu trữ chính cho thư viện nhạc kỹ thuật số, vì vậy, rõ ràng, nó có vai trò trung tâm trong hệ thống của chúng tôi. Nhóm sẽ cần đảm bảo tạo quy trình quản trị để duy trì kho dữ liệu này: thêm thông tin mới vào kho, sửa đổi nội dung hiện có và xóa thông tin khỏi kho. Các tác vụ quản trị này đôi khi bị bỏ qua trong các tác vụ định hướng kinh doanh được liệt kê trong các trụ sở hợp sử dụng, do đó, nhóm phải đảm bảo rằng các quy trình được đưa vào để thêm, sửa đổi và xóa nội dung của các kho lưu trữ dữ liệu ở đâu đó trong hệ thống là tùy thuộc vào họ. Chỉ cần kiểm tra cú pháp DFD (tất cả các kho lưu trữ dữ liệu phải có ít nhất một luồng dữ liệu đầu vào và ít nhất một luồng dữ liệu đầu ra) đã giúp phát hiện ra thiếu sót này trong mô hình quy trình.

Mặc dù có thể phân tách một số quy trình trên DFD cấp 1 thành chi tiết hơn n trên sơ đồ cấp 2, nhóm dự án đã quyết định rằng bước đó là không cần thiết. Thay vào đó, họ đảm bảo rằng các mô tả quy trình mà họ đã tạo trong kho lưu trữ CASE cho mỗi quy trình đều rất chi tiết. Chi tiết này sẽ rất quan trọng trong việc phát triển các mô hình dữ liệu và thiết kế giao diện người dùng và các chương trình trong giai đoạn thiết kế.

Xác thực sơ đồ luồng dữ liệu

Bộ DFD cuối cùng đã được xác nhận bởi nhóm dự án và sau đó bởi Carly và nhóm tiếp thị của cô ấy trong cuộc họp JAD cuối cùng. Đã có sự chấp thuận chung về các quy trình hướng tới khách hàng (quy trình 1 và quy trình 2) và thảo luận đáng kể về thông tin cụ thể sẽ được yêu cầu để giúp nhóm tiếp thị đưa ra các quyết định về chiến dịch quảng cáo của mình (quy trình 3). Thông tin này đã được ghi lại trong kho lưu trữ CASE để có thể gọi lại trong quá trình phát triển mô hình dữ liệu của hệ thống, chủ đề của chương tiếp theo của chúng ta.

BẢN TÓM TẮT

Sơ đồ luồng dữ liệu Cú pháp

pháp Bốn ký hiệu được sử dụng trên sơ đồ luồng dữ liệu (các quy trình, luồng dữ liệu, kho lưu trữ dữ liệu và các thực thể bên ngoài). Một quá trình là một hoạt động làm một cái gì đó. Mỗi quy trình có một tên (cụm động từ), mô tả và một số cho biết vị trí của nó trong mối quan hệ với các quy trình khác và với các quy trình con của nó. Mỗi quá trình phải có ít nhất một đầu ra và thường có ít nhất một đầu vào. Luồng dữ liệu là một phần dữ liệu hoặc một đối tượng và có tên (danh từ) và mô tả và bắt đầu hoặc kết thúc tại một quy trình (hoặc cả hai). Kho lưu trữ dữ liệu là tệp thủ công hoặc tệp máy tính và nó có số, tên (danh từ) và ít nhất một luồng dữ liệu đầu vào và một luồng dữ liệu đầu ra (trừ khi kho lưu trữ dữ liệu được tạo bởi một quy trình bên ngoài dữ liệu lưu đồ [DFD]). Thực thể bên ngoài là một người, tổ chức hoặc hệ thống bên ngoài phạm vi của hệ thống và có tên (danh từ) và mô tả. Mỗi bộ DFD bắt đầu bằng một sơ đồ ngữ cảnh và DFD cấp 0 và có nhiều DFD cấp 1, DFD cấp 2, v.v. Mọi phần tử trên DFD cấp cao hơn (nghĩa là luồng dữ liệu, lưu trữ dữ liệu và thực thể bên ngoài) phải xuất hiện trên DFD cấp thấp hơn, nếu không chúng sẽ không cân bằng.

Tạo biểu đồ luồng dữ liệu

Các DFD được tạo từ các trường hợp sử dụng. Đầu tiên, nhóm xây dựng biểu đồ ngữ cảnh hiển thị tất cả các thực thể bên ngoài và dữ liệu vào và ra khỏi hệ thống từ chúng. Thứ hai, nhóm tạo các đoạn DFD cho từng trường hợp sử dụng để chỉ ra cách trường hợp sử dụng trao đổi luồng dữ liệu với các thực thể bên ngoài và kho lưu trữ dữ liệu. Thứ ba, các mảnh DFD này được tổ chức thành một DFD cấp 0. Thứ tư, nhóm phát triển DFD cấp 1 trên cơ sở các bước trong mỗi trường hợp sử dụng để giải thích rõ hơn cách chúng hoạt động. Thứ năm, nhóm xác thực bộ DFD để đảm bảo rằng chúng hoàn chỉnh, chính xác và không có lỗi cú pháp hoặc ngữ nghĩa. Các nhà phân tích hiếm khi tạo DFD hoàn hảo ngay lần đầu tiên, do đó, việc lặp lại rất quan trọng để đảm bảo rằng cả DFD một trang và nhiều trang đều rõ ràng và dễ đọc.

ĐIỀU KHOẢN QUAN TRỌNG

tuyên bố hành động	phân hủy	Cha mẹ
cân bằng	đoạn DFD	mô hình vật lý
Bó	Thực thể bên ngoài	mô hình quy trình
tuyên bố trừu tượng hợp	Đối với tuyên bố	Quá trình
Những đứa trẻ	câu lệnh nếu	lỗi ngữ nghĩa
Sơ đồ ngữ cảnh	lặp lại	Tiếng Anh có cấu trúc
Dòng dữ liệu	Cách trình bày	Lỗi cú pháp
Sơ đồ luồng dữ liệu (DFD)	DFD cấp 0	Quan điểm
Kho dữ liệu	DFD cấp 1	câu lệnh While
bảng quyết định	DFD cấp 2	
cây quyết định	Mô hình quy trình logic	

CÂU HỎI

1. Mô hình quy trình là gì? Sơ đồ luồng dữ liệu là gì? Là hai liên quan? Nếu vậy, làm thế nào?

2. Phân biệt giữa mô hình quy trình logic và mô hình quy trình vật lý.

3. Định nghĩa quy trình trong mô hình quy trình có nghĩa là gì. Làm thế nào một quá trình nên được đặt tên? Thông tin nào về một quy trình nên được lưu trữ trong kho lưu trữ CASE?

4. Định nghĩa ý nghĩa của luồng dữ liệu trong mô hình quy trình. Luồng dữ liệu nên được đặt tên như thế nào? Thông tin nào về luồng dữ liệu nên được lưu trữ trong kho lưu trữ CASE?

5. Định nghĩa kho lưu trữ dữ liệu trong mô hình quy trình nghĩa là gì. Một kho lưu trữ dữ liệu nên được đặt tên như thế nào? Thông tin nào về kho lưu trữ dữ liệu nên được lưu trữ trong kho lưu trữ CASE?

6. Xác định ý nghĩa của một thực thể bên ngoài trong một mô hình quy trình. Thực thể bên ngoài nên được đặt tên như thế nào? Thông tin nào về một thực thể bên ngoài sẽ được lưu trữ trong kho lưu trữ CASE?

7. Tại sao một mô hình quy trình thư ờng bao gồm một tập hợp các DFD? Sự phân rã của một quy trình kinh doanh có nghĩa là gì?

8. Giải thích mối quan hệ giữa sơ đồ ngữ cảnh DFD và sơ đồ DFD mức 0.

9. Giải thích mối quan hệ giữa sơ đồ DFD cấp 0 và (các) sơ đồ DFD cấp 1.

10. Thảo luận về cách nhà phân tích biết cách ngừng phân tách mô hình quy trình thành các cấp độ ngày càng chi tiết hơn.
11. Giả sử rằng một tiến trình trên DFD được đánh số 4.3.2. Biểu đồ cấp độ nào chứa quy trình này? Quá trình mẹ của quá trình này là gì?

12. Giải thích việc sử dụng cấu trúc tiếng Anh trong mô tả quy trình.

13. Tại sao ngư ời ta lại sử dụng cây quyết định và/hoặc bảng quyết định trong mô tả quy trình?

14. Giải thích quá trình cân bằng một bộ DFD.

15. Các luồng dữ liệu loại trừ lẫn nhau (nghĩa là thay đổi đư ờng dẫn gốc thông qua một quy trình) đư ợc mô tả như thế nào trong DFD?

16. Thảo luận về một số cách để xác minh tính đúng đắn của một mô hình quy trình.

17. Xác định 3 lỗi cú pháp điển hình thư ờng gặp đư ợc tìm thấy trong DFD.

18. Lỗi ngữ nghĩa DFD nghĩa là gì? Cung cấp một ví dụ.

19. Tạo các trừu tượng hợp sử dụng khi làm việc với ngư ời dùng là một bư ớc phát triển gần đây trong thực hành phân tích hệ thống. Tại sao ngày nay xu hướ ng sử dụng các trừu tượng hợp sử dụng trong các cuộc phỏng vấn ngư ời dùng hoặc các phiên JAD?

20. Bạn có thể làm cho DFD dễ hiểu hơn bằng cách nào? (Trư ớc tiên hãy nghĩ về cách làm cho một ngư ời khó hiểu.)

21. Giả sử rằng mục tiêu của bạn là tạo một tập hợp các DFD. Làm thế nào bạn sẽ bắt đầu một cuộc phỏng vấn với một ngư ời dùng có hiểu biết? Làm thế nào bạn sẽ bắt đầu một phiên JAD?

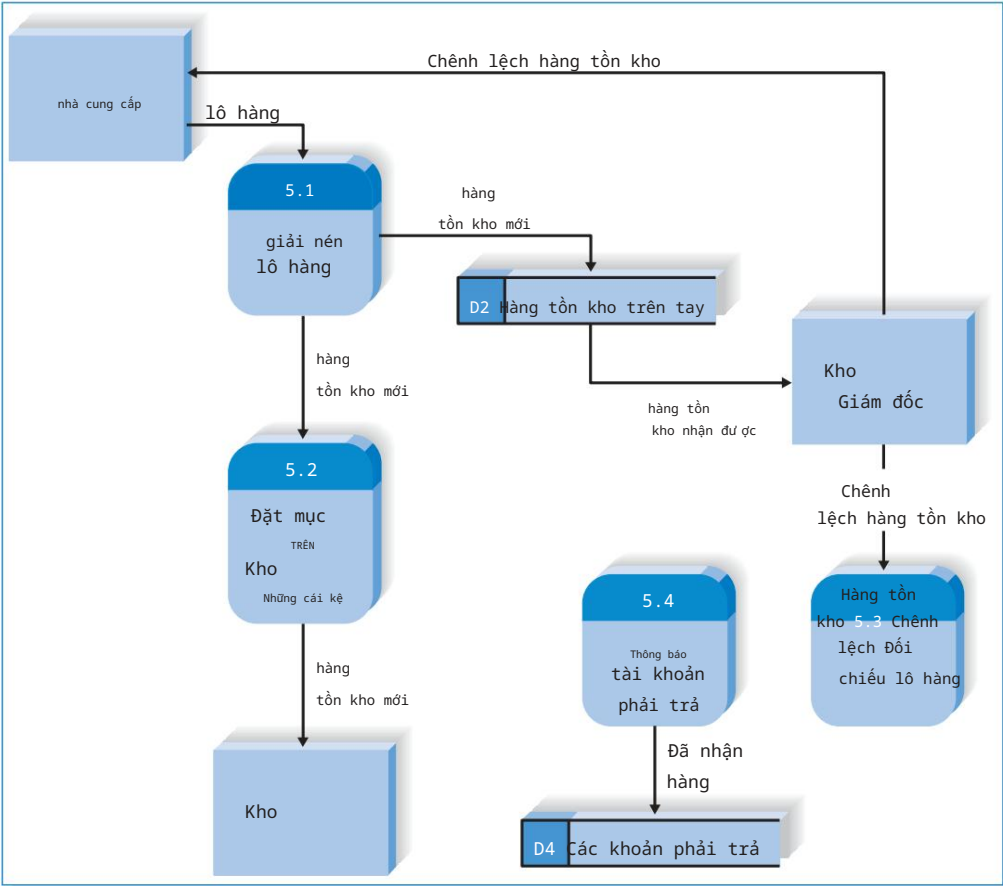
BÀI TẬP

- A. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho quá trình mua kính trong Bài tập A, Chương 4.
- B. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống văn phòng nha sĩ trong Bài tập B, Chương 4.
- C. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống đại học ở bài tập D chương 4.
- D. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống bắt động sản trong Bài tập E, Chương 4.
- E. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống lưu trữ video trong Bài tập F, Chương 4.
- F. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống câu lạc bộ sức khỏe trong Bài tập G, Chương 4.
- G. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống Picnics R Us trong Bài tập H, Chương 4.
- H. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống Câu lạc bộ tháng trong Bài tập I, Chương 4.
- I. Vẽ sơ đồ luồng dữ liệu mức 0 (DFD) cho hệ thống thư viện trường đại học trong bài tập J, chương 4.

CỬA SỔ NHỎ

1. Công ty Hatcher đang trong quá trình phát triển một hệ thống quản lý hàng tồn kho mới. Một trong những quy trình xử lý sự kiện trong hệ thống đó là Nhận lô hàng của nhà cung cấp. Nhà phân tích hệ thống (chưa có kinh nghiệm) của dự án đã dành thời gian ở trong kho để quan sát quá trình này và phát triển danh sách các hoạt động được thực hiện sau đây: nhận đơn đặt hàng mới trong kho, mở hộp, đảm bảo rằng tất cả các mặt hàng đã đặt đã thực sự được chuyển đến. đã nhận, đặt các mặt hàng lên kệ chính xác, giao dịch với nhà cung cấp để điều chỉnh bất kỳ sự khác biệt nào, điều chỉnh số lượng hàng tồn kho có sẵn và chuyển thông tin vận chuyển đến văn phòng tài khoản phải trả. Anh ấy cũng đã tạo sơ đồ luồng dữ liệu Cấp 1 đi kèm cho quá trình này. Thật không may, DFD này có nhiều lỗi cú pháp và ngữ nghĩa. Xác định các lỗi. Vẽ lại DFD để thể hiện chính xác hơn quy trình Nhận lô hàng của nhà cung cấp.
 2. Quản lý nhân viên khoa học và chuyên nghiệp (PSSM) là một loại cơ quan nhân sự tạm thời duy nhất. Nhiều tổ chức ngày nay thuê nhân viên kỹ thuật có tay nghề cao trong thời gian ngắn, tạm thời để hỗ trợ các dự án đặc biệt hoặc cung cấp kỹ năng kỹ thuật cần thiết. PSSM đàm phán hợp đồng với các công ty khách hàng của mình, trong đó PSSM đồng ý cung cấp nhân viên tạm thời trong các hạng mục công việc cụ thể với một mức chi phí cụ thể. Ví dụ, PSSM có một hợp đồng với một công ty thăm dò dầu khí, theo đó công ty này đồng ý cung cấp cho các nhà địa chất có ít nhất bằng thạc sĩ với mức lương 5000 đô la mỗi tuần. PSSM có hợp đồng với nhiều công ty và có thể tuyển dụng hầu hết mọi loại nhân viên chuyên nghiệp hoặc khoa học, từ lập trình viên máy tính đến nhà địa chất đến nhà vật lý thiên văn.
- Khi một công ty khách hàng của PSSM xác định rằng họ sẽ cần một nhân viên khoa học hoặc chuyên gia tạm thời, họ sẽ đưa ra một yêu cầu nhân sự so với hợp đồng mà họ đã ký trước đó.
- đã thu được lợi nhuận với PSSM. Khi người quản lý hợp đồng của PSSM nhận được yêu cầu nhân sự, số hợp đồng được tham chiếu trong yêu cầu nhân sự sẽ được nhập vào cơ sở dữ liệu hợp đồng. Sử dụng thông tin từ cơ sở dữ liệu, người quản lý hợp đồng xem xét các điều khoản và điều kiện của hợp đồng và xác định xem yêu cầu nhân sự có hợp lệ hay không. Yêu cầu nhân sự hợp lệ nếu hợp đồng chưa hết hạn, loại nhân viên chuyên môn hoặc khoa học được yêu cầu được liệt kê trong hợp đồng ban đầu và phí yêu cầu nằm trong phạm vi phí đã thu được lợi nhuận. Nếu yêu cầu nhân sự không hợp lệ, người quản lý hợp đồng sẽ gửi lại yêu cầu nhân sự cho khách hàng kèm theo một lá thư nêu rõ lý do tại sao không thể nộp yêu cầu nhân sự và một bản sao của lá thư đó sẽ được nộp. Nếu yêu cầu nhân sự hợp lệ, người quản lý hợp đồng sẽ nhập yêu cầu nhân sự vào cơ sở dữ liệu yêu cầu nhân sự, dưới dạng yêu cầu nhân sự xuất sắc. Yêu cầu nhân sự sau đó được gửi đến bộ phận sắp xếp PSSM.
- Trong bộ phận sắp xếp, loại nhân viên, kinh nghiệm và trình độ được yêu cầu trong yêu cầu nhân sự được kiểm tra dựa trên cơ sở dữ liệu về các nhân viên khoa học và chuyên nghiệp hiện có. Nếu một cá nhân đủ tiêu chuẩn được tìm thấy, người đó sẽ được đánh dấu là "dành riêng" trong cơ sở dữ liệu nhân viên. Nếu không thể tìm thấy một cá nhân đủ điều kiện trong cơ sở dữ liệu hoặc không có sẵn ngay lập tức, bộ phận sắp xếp sẽ tạo một bản ghi nhớ giải thích việc không thể đáp ứng yêu cầu nhân sự và đính kèm nó vào yêu cầu nhân sự. Tất cả các yêu cầu nhân sự sau đó được gửi đến bộ phận sắp xếp.
- Trong bộ phận sắp xếp, nhân viên tạm thời tiềm năng được liên hệ và yêu cầu đồng ý với vị trí. Sau khi các chi tiết về vị trí đã được tính toán và đồng ý, nhân viên được đánh dấu là "được bổ nhiệm" trong cơ sở dữ liệu nhân viên. Một bản sao của nhân sự

Hàng tồn kho của công ty Hatcher
Hệ thống quản lý DFD cấp 1



yêu cầu và một hóa đơn cho phí vị trí được gửi cho khách hàng. Cuối cùng, yêu cầu nhân sự, bản ghi nhớ “không thể điền” (nếu có) và một bản sao hóa đơn phí sắp xếp được gửi đến người quản lý hợp đồng. Nếu yêu cầu nhân sự đã được điền, người quản lý hợp đồng sẽ đóng yêu cầu nhân sự đang mở trong cơ sở dữ liệu yêu cầu nhân sự. Nếu không thể đáp ứng yêu cầu nhân sự, khách hàng sẽ được thông báo. Sau đó, yêu cầu nhân sự, hóa đơn phí sắp xếp và bản ghi nhớ “không thể điền” sẽ được nộp tại văn phòng hợp đồng.

Một. Phát triển một tư duy hợp sử dụng cho từng quy trình chính vừa được mô tả. b. Tạo biểu đồ ngữ cảnh cho hệ thống vừa được mô tả. c. Tạo các đoạn DFD cho từng tư duy hợp trong số bốn tư duy hợp sử dụng được nêu trong phần a, sau đó kết hợp chúng thành DFD cấp 0.

đ. Tạo DFD cấp 1 cho mục đích sử dụng phức tạp nhất tư duy hợp.

Trang này đư ợc chủ ý để trống bên trái

LẬP KẾ HOẠCH

PHÂN TÍCH

- ☐ Sử dụng các kỹ thuật khơi gợi yêu cầu (phỏng vấn, phiên JAD, bảng câu hỏi, phân tích tài liệu và quan sát).
- ☐ Áp dụng các chiến lược phân tích yêu cầu khi cần thiết để khám phá các yêu cầu cơ bản.
- ☐ Phát triển định nghĩa yêu cầu.
- ☐ Phát triển các trường hợp sử dụng.
- ☐ Xây dựng sơ đồ luồng dữ liệu.
- ☐ Xây dựng mô hình quan hệ thực thể
- ☐ Chuẩn hóa mô hình quan hệ thực thể

DANH SÁCH KIỂM TRA NHIỆM VỤ

LẬP KẾ HOẠCH

PHÂN TÍCH S

THIẾT KẾ

CHƯƠNG 6

DỮ LIỆU

MÔ HÌNH

MỘT

mô hình dữ liệu mô tả dữ liệu chảy qua các quy trình kinh doanh trong một tổ chức. Trong giai đoạn phân tích, mô hình dữ liệu trình bày cách tổ chức dữ liệu hợp lý mà không chỉ ra cách dữ liệu được lưu trữ, tạo hoặc thao tác để các nhà phân tích có thể tập trung vào hoạt động kinh doanh mà không bị phân tâm bởi các chi tiết kỹ thuật.

Sau đó, trong giai đoạn thiết kế, mô hình dữ liệu được thay đổi để phản ánh chính xác cách dữ liệu sẽ được lưu trữ trong cơ sở dữ liệu và tệp. Chương này mô tả việc kết hợp sơ đồ mối quan hệ thực thể, một trong những kỹ thuật lập mô hình dữ liệu phổ biến nhất được sử dụng trong công nghiệp.

MỤC TIÊU

- Giải thích các quy tắc và hướng dẫn phong cách để tạo sơ đồ mối quan hệ thực thể.
- Tạo sơ đồ mối quan hệ thực thể.
- Mô tả việc sử dụng từ điển dữ liệu và siêu dữ liệu.
- Giải thích cách cân bằng sơ đồ mối quan hệ thực thể và sơ đồ luồng dữ liệu.
- Mô tả quá trình bình thường hóa.

TỔNG KẾT CHƯƠNG

Giới thiệu	Cú pháp nâng cao
Sơ đồ mối quan hệ thực thể	Áp dụng các khái niệm tại Tune Source
Đọc một mối quan hệ thực thể	Xác nhận ERD
Biểu đồ	Nguyên tắc thiết kế bình thường hóa
Các yếu tố của một mối quan hệ thực thể	Mối quan hệ thực thể cân bằng
Biểu đồ	Sơ đồ với Sơ đồ luồng dữ liệu
Từ điển dữ liệu và siêu dữ liệu	Bản tóm tắt
Tạo sơ đồ mối quan hệ thực thể	Phụ lục 6A: Chuẩn hóa dữ liệu
Xây dựng mối quan hệ thực thể sơ đồ	Ngữ ời mẫu



GIỚI THIỆU

Trong giai đoạn phân tích, các nhà phân tích tạo ra các mô hình quy trình để thể hiện cách thức hoạt động của hệ thống kinh doanh. Đồng thời, các nhà phân tích cần hiểu các thông tin được sử dụng và tạo ra bởi hệ thống kinh doanh (ví dụ: thông tin khách hàng, thông tin đặt hàng). Trong chương này, chúng ta thảo luận về cách dữ liệu chảy qua các quy trình được tổ chức và trình bày.

Mô hình dữ liệu là một cách chính thức để biểu diễn dữ liệu được sử dụng và tạo ra bởi một hệ thống kinh doanh; nó minh họa con người, địa điểm hoặc sự vật mà thông tin được nắm bắt và chúng liên quan với nhau như thế nào. Mô hình dữ liệu được vẽ bởi một quy trình lặp đi lặp lại, trong đó mô hình trở nên chi tiết hơn và ít khái niệm hơn theo thời gian. Trong quá trình phân tích, các nhà phân tích vẽ một mô hình dữ liệu logic, mô hình này cho thấy tổ chức dữ liệu logic mà không chỉ ra cách dữ liệu được lưu trữ, tạo hoặc thao tác. Bởi vì mô hình này không có bất kỳ chi tiết triển khai hoặc kỹ thuật nào, các nhà phân tích có thể tập trung để dàng hơn vào việc khớp sơ đồ với các yêu cầu kinh doanh thực tế của hệ thống.

Trong giai đoạn thiết kế, các nhà phân tích vẽ một mô hình dữ liệu vật lý để phản ánh cách dữ liệu sẽ được lưu trữ vật lý trong cơ sở dữ liệu và tệp. Tại thời điểm này, các nhà phân tích nghiên cứu các cách để lưu trữ dữ liệu một cách hiệu quả và làm cho dữ liệu dễ dàng truy xuất. Mô hình dữ liệu vật lý và điều chỉnh hiệu suất được thảo luận trong Chương 11.

Các nhóm dự án thường sử dụng các công cụ CASE để vẽ các mô hình dữ liệu. Một số công cụ CASE là các gói mô hình hóa dữ liệu, chẳng hạn như ERwin của Platinum Technology, giúp các nhà phân tích tạo và duy trì các mô hình dữ liệu logic và vật lý; chúng có nhiều khả năng để hỗ trợ người lập mô hình và chúng có thể tự động tạo nhiều loại cơ sở dữ liệu khác nhau từ các mô hình được tạo. Các công cụ CASE khác (ví dụ: Oracle Designer) đi kèm với các hệ thống quản lý cơ sở dữ liệu (ví dụ: Oracle) và chúng đặc biệt tốt cho việc lập mô hình cơ sở dữ liệu sẽ được xây dựng trong các sản phẩm cơ sở dữ liệu đồng hành của chúng. Tùy chọn cuối cùng là sử dụng công cụ CASE đầy đủ dịch vụ, chẳng hạn như Bàn làm việc của nhà phân tích có thể nhìn thấy, trong đó mô hình hóa dữ liệu là một trong nhiều khả năng và công cụ này có thể được sử dụng với nhiều cơ sở dữ liệu khác nhau. Lợi ích của công cụ CASE đầy đủ dịch vụ là nó tích hợp thông tin mô hình dữ liệu với các phần liên quan khác của dự án.

Trong chương này, chúng ta tập trung vào việc tạo một mô hình dữ liệu logic. Mặc dù có một số cách để lập mô hình dữ liệu, nhưng chúng tôi sẽ trình bày một trong những kỹ thuật được sử dụng phổ biến nhất: lập sơ đồ mối quan hệ thực thể, một kỹ thuật vẽ đồ họa do Peter Chen¹ phát triển để hiển thị tất cả các thành phần dữ liệu của một hệ thống kinh doanh. Trước tiên, chúng tôi sẽ mô tả cách tạo sơ đồ mối quan hệ thực thể (ERD) và thảo luận về một số nguyên tắc về phong cách. Sau đó, chúng tôi sẽ trình bày một kỹ thuật gọi là chuẩn hóa giúp phân tích phân tích xác thực các mô hình dữ liệu mà chúng vẽ. Chương này kết thúc với phần thảo luận về cách các mô hình dữ liệu cân bằng hoặc tương quan với các mô hình quy trình mà bạn đã học trong Chương 5.

SƠ ĐỒ MỐI QUAN HỆ THỰC THỂ

Sơ đồ mối quan hệ thực thể (ERD) là một bức tranh hiển thị thông tin được tạo, lưu trữ và sử dụng bởi một hệ thống kinh doanh. Một nhà phân tích có thể đọc ERD để khám phá các mẫu thông tin riêng lẻ trong một hệ thống và cách chúng được tổ chức

¹ P. Chen, "The Entity-Relationship Model—Toward a Unified View of Data," ACM Transactions on Database Systems, 1976, 1:9-36.

và liên quan đến nhau. Trên ERD, các loại thông tin tự động tự được liệt kê cùng nhau và đặt bên trong các hộp được gọi là thực thể. Các đường được vẽ giữa các thực thể để thể hiện mối quan hệ giữa dữ liệu và các ký hiệu đặc biệt được thêm vào sơ đồ để truyền đạt các quy tắc kinh doanh cấp cao cần được hệ thống hỗ trợ. ERD ngụ ý không có trật tự, mặc dù các thực thể có liên quan đến nhau thường được đặt gần nhau.

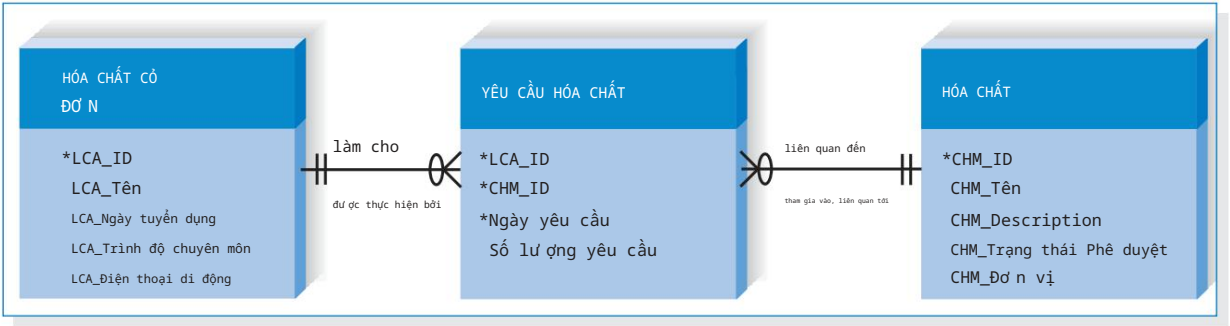
Ví dụ, hãy xem xét hệ thống Yêu cầu Hóa chất cho Cỏ được mô tả trong Chương 5. Mặc dù hệ thống này chỉ là một phần nhỏ của hệ thống thông tin dành cho doanh nghiệp chăm sóc cỏ, nhưng chúng ta sẽ sử dụng nó để thảo luận về cách đọc sơ đồ mối quan hệ thực thể. Trước tiên, hãy quay lại và xem DFD mẫu cho quy trình yêu cầu hóa chất trong Hình 5-1. Mặc dù chúng ta hiểu cách thức hoạt động của hệ thống từ việc nghiên cứu sơ đồ luồng dữ liệu, nhưng chúng ta có rất ít hiểu biết chi tiết về bản thân thông tin chảy qua hệ thống. Chính xác thì “yêu cầu hóa chất mới” là gì? Những phần dữ liệu nào được thu thập trong “ủy quyền nhận hóa chất

Đọc sơ đồ mối quan hệ thực thể

Nhà phân tích có thể trả lời những câu hỏi này và hơn thế nữa bằng cách sử dụng sơ đồ mối quan hệ thực thể. Chúng tôi đã bao gồm ERD một phần cho kịch bản yêu cầu hóa chất trong Hình 6-1. Đầu tiên, chúng tôi đã sắp xếp dữ liệu thành ba loại chính: Người sử dụng hóa chất cho cỏ, Yêu cầu về hóa chất và Hóa chất. Dữ liệu của Lawn Chemical Applicator mô tả những nhân viên sử dụng hóa chất cho bãi cỏ. Dữ liệu Yêu cầu Hóa chất thu thập thông tin về mọi sự kiện yêu cầu hóa chất và dữ liệu hóa chất mô tả các hóa chất được sử dụng để chăm sóc cỏ.

Chúng ta cũng có thể thấy các sự kiện cụ thể mô tả từng loại trong ba loại. Ví dụ: một hóa chất được mô tả bằng số ID, tên, mô tả, trạng thái phê duyệt và đơn vị đo lường. Chúng ta cũng có thể xem những gì có thể được sử dụng để xác định duy nhất một hóa chất, yêu cầu hóa chất và LCA bằng cách tìm kiếm các dấu hoa thị bên cạnh các thành phần dữ liệu. Một ID duy nhất đã được tạo để xác định mọi LCA và mọi hóa chất. Một yêu cầu hóa chất được xác định duy nhất bằng sự kết hợp của ID LCA, ID hóa chất và ngày yêu cầu.

Các đường kết nối ba loại thông tin truyền đạt các mối quan hệ mà các loại chia sẻ. Bằng cách đọc các đường quan hệ, nhà phân tích hiểu rằng LCA đưa ra các yêu cầu hóa chất và các yêu cầu hóa chất liên quan đến hóa chất.



HÌNH 6-1
Yêu cầu hóa chất ERD

ERD cũng truyền đạt các quy tắc kinh doanh cấp cao. Quy tắc kinh doanh là các ràng buộc hoặc hướng dẫn được tuân theo trong quá trình vận hành hệ thống; đó là các quy tắc như “Một khoản thanh toán có thể bằng tiền mặt, séc, thẻ ghi nợ, thẻ tín dụng, (các) phiếu giảm giá hoặc phiếu thực phẩm,” “Một vụ mua bán được trả bằng một hoặc nhiều khoản thanh toán,” hoặc “Một khách hàng có thể đặt nhiều mệnh lệnh.” Trong suốt một ngày làm việc, mọi người liên tục áp dụng các quy tắc kinh doanh để thực hiện công việc của họ và họ biết các quy tắc thông qua đào tạo hoặc biết nơi để tra cứu chúng. Nếu xảy ra tình huống không biết các quy tắc, người lao động có thể phải tham khảo hướng dẫn chính sách hoặc thủ tục bằng văn bản để xác định các quy tắc kinh doanh phù hợp.

Trên một mô hình dữ liệu, các quy tắc kinh doanh được truyền đạt bằng các loại tàu quan hệ mà các thực thể chia sẻ. Ví dụ: từ ERD, chúng tôi biết từ “vết chân chim” được đặt trên đường dây gần Yêu cầu Hóa chất nhất rằng một LCA có thể đưa ra nhiều Yêu cầu Hóa chất. Chúng ta có thể thấy bằng hai thanh được đặt trên đường gần LCA nhất rằng một Yêu cầu Hóa chất được thực hiện bởi chính xác một LCA. Cuối cùng, hệ thống mới sẽ hỗ trợ các quy tắc kinh doanh mà chúng tôi vừa mô tả và nó phải đảm bảo rằng người dùng không vi phạm các quy tắc khi thực hiện các quy trình của hệ thống.

Do đó, trong ví dụ của chúng tôi, hệ thống không được cho phép thực hiện yêu cầu hóa chất không liên quan đến LCA. Tương tự, hệ thống không được cho phép một yêu cầu hóa chất liên quan đến nhiều hơn một LCA.

Bây giờ bạn đã thấy một ERD, hãy lùi lại và tìm hiểu những điều cơ bản về ERD. Trong các phần tiếp theo, trước tiên chúng ta sẽ mô tả cú pháp của ERD, sử dụng biểu đồ trong Hình 6-1. Sau đó, chúng tôi sẽ hướng dẫn bạn cách tạo ERD bằng cách sử dụng một ví dụ từ Tune Source.

Các yếu tố của một sơ đồ mối quan hệ thực thể

Có ba yếu tố cơ bản trong ngôn ngữ mô hình hóa dữ liệu (thực thể, thuộc tính và mối quan hệ), mỗi yếu tố được biểu thị bằng một ký hiệu đồ họa khác nhau.

Có nhiều bộ biểu tượng khác nhau có thể được sử dụng trên ERD. Không một bộ ký hiệu nào thống trị việc sử dụng trong ngành và không có bộ ký hiệu nào nhất thiết phải tốt hơn bộ ký hiệu nào. Chúng tôi sẽ sử dụng vết chân chim trong cuốn sách này. Hình 6-2 tóm tắt ba yếu tố cơ bản của ERD và các ký hiệu mà chúng ta sẽ sử dụng.

Thực thể Thực thể là khối xây dựng cơ bản cho một mô hình dữ liệu. Đó là một người, địa điểm, sự kiện hoặc sự vật mà dữ liệu được thu thập—ví dụ: nhân viên, đơn đặt hàng hoặc sản phẩm. Một thực thể được mô tả bằng một hình chữ nhật và nó được mô tả bằng một danh từ số ít được đánh vần bằng chữ in hoa. Tất cả các thực thể đều có tên, mô tả ngắn giải thích chúng là gì và mã định danh là cách định vị thông tin trong thực thể (sẽ được thảo luận sau). Trong Hình 6-1, các thực thể là Lawn Chemical Applicator, Chemical Request và Chemical.

Các thực thể đại diện cho một cái gì đó tồn tại nhiều phiên bản hoặc lần xuất hiện. Ví dụ, John Smith và Susan Jones có thể là các thể hiện của thực thể khách hàng (Hình 6-3). Chúng tôi mong đợi thực thể khách hàng đại diện cho tất cả những người mà chúng tôi đã hợp tác kinh doanh và mỗi người trong số họ sẽ là một thực thể trong thực thể khách hàng. Nếu chỉ có một trường hợp hoặc sự xuất hiện của một người, địa điểm, sự kiện hoặc sự vật thì không nên đưa trường hợp đó vào làm thực thể trong mô hình dữ liệu. Ví dụ, hãy nghĩ rộng hơn một chút về hệ thống thông tin của doanh nghiệp chăm sóc bãi cỏ. Hình 6-1 chỉ tập trung vào một phần nhỏ của hệ thống thông tin đó. Chúng tôi giả định rằng công ty bao gồm nhiều hơn một Lawn Chemical Applicator, bởi vì chúng tôi đã bao gồm một thực thể LCA để nắm bắt thông tin cụ thể về từng loại.

Tuy nhiên, nếu công ty được sở hữu và điều hành bởi một người duy nhất, thì sẽ có

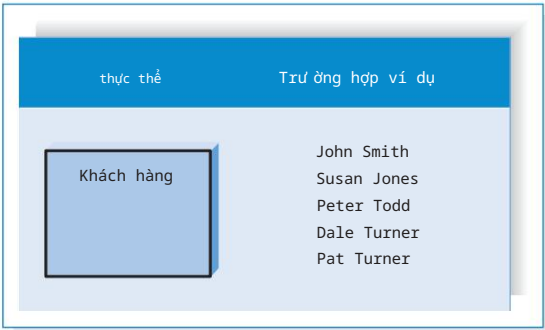
	IDEFIX	Trần	con quạ Chân
<div>ENTITY là</div> <div><div>✓ một người, địa điểm hoặc sự vật. có một tên riêng đánh vần bằng tất cả các chữ in hoa. có một định danh. nên chứa nhiều hơn một phiên bản dữ liệu.</div></div>	<div>ENTITY-NAME</div> <div><div>định danh</div></div>	<div>ENTITY-NAME ENTITY-NAME</div> <div></div>	<div>*Số nhận dạng</div>
<div>ATTRIBUTE là thuộc</div> <div><div>✓ tính của một thực thể. nên được sử dụng bởi ít nhất một quy trình nghiệp vụ. được chia nhỏ đến mức độ chi tiết hữu ích nhất của nó.</div></div>	<div>ENTITY-NAME</div> <div><div>tên thuộc tính</div><div>tên thuộc tính</div><div>tên thuộc tính</div></div>	<div>tên thuộc tính</div> <div></div>	<div>ENTITY-NAME</div> <div><div>tên thuộc tính</div><div>tên thuộc tính</div><div>tên thuộc tính</div></div>
<div>MỘT MỐI QUAN HỆ</div> <div><div>✓ thể hiện sự liên kết giữa hai thực thể. có một thực thể mẹ và một thực thể con. được miêu tả bằng một cụm động từ. có lực lượng (1 : 1, 1 : N, hoặc M : N). có phụ thuộc (null, không null). là phụ thuộc hoặc độc lập.</div></div>	<div>Tên mối quan hệ</div>	<div>Mối quan hệ tên</div>	<div>Tên mối quan hệ</div>

HÌNH 6-2
Bộ ký hiệu mô hình hóa dữ liệu

không cần thiết lập một thực thể LCA trong mô hình dữ liệu tổng thể. Không cần phải thu thập dữ liệu trong hệ thống về thứ gì đó chỉ có một phiên bản duy nhất.

Thuộc tính Một thuộc tính là một số loại thông tin được nắm bắt về một thực thể. Ví dụ: họ, địa chỉ nhà và địa chỉ e-mail đều là các thuộc tính của một khách hàng. Có thể dễ dàng đưa ra hàng trăm thuộc tính cho một thực thể (ví dụ: khách hàng có màu mắt, sở thích yêu thích, tôn giáo), nhưng chỉ những thuộc tính thực sự sẽ được sử dụng bởi quy trình kinh doanh mới được đưa vào mô hình .

Thuộc tính là danh từ được liệt kê trong một thực thể. Thông thường, một số dạng của tên thực thể được thêm vào phần đầu của mỗi thuộc tính để làm cho nó rõ ràng như



HÌNH 6-3
Thực thể và trụ sở hợp

nó thuộc về thực thể nào (ví dụ: CUS_lastname, CUS_address). Nếu không làm điều này, bạn có thể bị nhầm lẫn bởi nhiều thực thể có cùng thuộc tính—ví dụ: cả khách hàng và nhân viên đều có thể có thuộc tính gọi là “họ”. CUS_lastname và EMP_lastname là những cách rõ ràng hơn nhiều để đặt tên cho các thuộc tính trên mô hình dữ liệu.

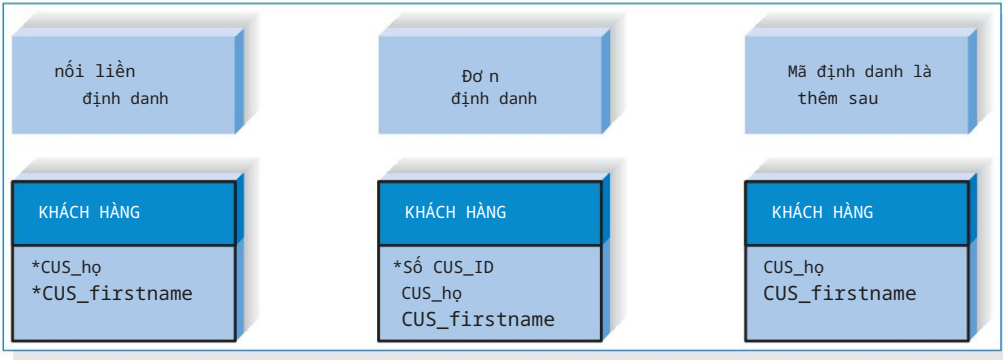
Một hoặc nhiều thuộc tính có thể đóng vai trò là số nhận dạng—(các) thuộc tính có thể xác định duy nhất một phiên bản của thực thể—và các thuộc tính đóng vai trò là số nhận dạng được đánh dấu bằng dấu hoa thị bên cạnh tên thuộc tính. Nếu không có khách hàng nào có cùng họ, thì họ có thể được sử dụng làm định danh của thực thể khách hàng. Trong trường hợp này, nếu chúng ta cần xác định vị trí của John Brown, thì cái tên Brown sẽ đủ để xác định một trường hợp duy nhất của họ Brown.

Giả sử rằng chúng tôi thêm một khách hàng tên là Sarah Brown. Bây giờ chúng ta có một vấn đề: Việc sử dụng tên Brown sẽ không chỉ dẫn đến một trường hợp duy nhất mà sẽ dẫn đến hai trường hợp (tức là John Brown và Sarah Brown). Bạn có ba lựa chọn vào thời điểm này và tất cả đều là giải pháp chấp nhận được. Đầu tiên, bạn có thể sử dụng kết hợp nhiều trường để làm mã định danh (họ và tên). Đây được gọi là mã định danh nối tiếp vì một số trường được kết hợp hoặc nối với nhau để xác định duy nhất một thể hiện. Thứ hai, bạn có thể tìm thấy một trường duy nhất cho từng phiên bản, chẳng hạn như số ID khách hàng. Thứ ba, bạn có thể đợi để gán một số nhận dạng (chẳng hạn như số được tạo ngẫu nhiên mà hệ thống sẽ tạo) cho đến giai đoạn thiết kế của SDLC (Hình 6-4). Nhiều người lập mô hình dữ liệu không tin rằng các số nhận dạng được tạo ngẫu nhiên thuộc về một mô hình dữ liệu logic, bởi vì chúng không tồn tại một cách logic trong quy trình kinh doanh.

Mối quan hệ Mối quan hệ là sự liên kết giữa các thực thể và chúng được thể hiện bằng các đường kết nối các thực thể với nhau. Mọi mối quan hệ đều có một thực thể mẹ và một thực thể con, thực thể mẹ là thực thể đầu tiên trong mối quan hệ và con là thực thể thứ hai.

Các mối quan hệ nên được dán nhãn rõ ràng bằng các động từ chủ động để có thể hiểu được các kết nối giữa các thực thể. Nếu một động từ được đưa ra cho mỗi mối quan hệ, nó được đọc theo hai hướng. Ví dụ: chúng ta có thể viết động từ *make* bên cạnh mối quan hệ của các thực thể LCA và Yêu cầu hóa chất, và động từ này sẽ được đọc là “một LCA đưa ra yêu cầu hóa chất” và “một LCA đưa ra yêu cầu hóa chất”. Trong Hình 6.1, chúng tôi đã bao gồm các từ cho cả hai hướng của đường quan hệ; các từ trên cùng được đọc từ cha mẹ sang con và các từ dưới cùng được đọc từ con sang cha mẹ. Lưu ý rằng thực thể LCA là thực thể mẹ trong mối quan hệ LCA-Yêu cầu hóa chất. Ngoài ra, một số công cụ CASE yêu cầu mỗi tên mối quan hệ phải là duy nhất trên ERD, vì vậy chúng tôi chọn các động từ mô tả duy nhất cho mỗi mối quan hệ.

HÌNH 6-4
Lựa chọn cho định danh



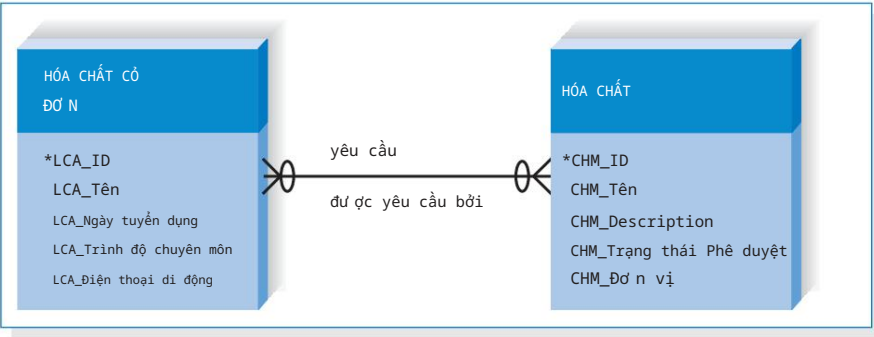
Mối quan hệ Cardinality có hai thuộc tính. Đầu tiên, một mối quan hệ có số lượng, là tỷ lệ giữa các phiên bản cha và các phiên bản con. Để xác định bản chất của một mối quan hệ, chúng tôi tự hỏi: “Có bao nhiêu phiên bản của một thực thể được liên kết với một phiên bản của thực thể kia?” (Hãy nhớ rằng một phiên bản là một lần xuất hiện của một thực thể, chẳng hạn như LCA John Brown hoặc Chemical Orthene™.) Ví dụ: một LCA đưa ra bao nhiêu yêu cầu về hóa chất? Số lượng đối với các mối quan hệ nhị phân (tức là, mỗi quan hệ giữa hai thực thể) là 1:1, 1:N hoặc M:N, và chúng ta sẽ thảo luận lần lượt từng vấn đề.

Mối quan hệ 1:1 (được đọc là “một đối một”) có nghĩa là một phiên bản của thực thể mẹ được liên kết với một phiên bản của thực thể con. Không có ví dụ nào về mối quan hệ 1:1 trong Hình 6-1. Vì vậy, hãy tưởng tượng trong giây lát rằng, như một phần thưởng, một công ty chỉ định một chỗ đậu xe dành riêng cụ thể cho mỗi nhân viên được vinh danh là “nhân viên của tháng”. Một chỗ đậu xe dành riêng được chỉ định cho mỗi nhân viên được vinh danh và mỗi nhân viên được vinh danh được chỉ định một chỗ đậu xe dành riêng. Nếu chúng ta vẽ hai thực thể này, chúng ta sẽ đặt một thanh bên cạnh thực thể Nhân viên và một thanh bên cạnh thực thể Chỗ đỗ xe dành riêng. Số lượng rõ ràng là 1:1 trong trường hợp này, bởi vì mỗi nhân viên được vinh danh được chỉ định chính xác một chỗ đậu xe dành riêng và một chỗ đậu xe dành riêng được chỉ định cho chính xác một nhân viên.

Trường hợp xuyên hơn, các mối quan hệ là 1:N (được đọc là “một đối với nhiều”). Trong loại quan hệ này, một thể hiện của thực thể mẹ được liên kết với nhiều thể hiện của thực thể con; tuy nhiên, thể hiện của thực thể con chỉ liên quan đến một thể hiện của thực thể cha. Ví dụ: một LCA (thực thể mẹ) có thể đưa ra nhiều Yêu cầu Hóa chất (thực thể con), nhưng một Yêu cầu Hóa chất cụ thể chỉ được thực hiện bởi một LCA, gợi ý về mối quan hệ 1:N giữa LCA và Yêu cầu Hóa chất. Một ký tự giống vết chân chim được đặt gần thực thể Yêu cầu Hóa chất nhất để hiển thị kết thúc “nhiều” của mối quan hệ. Thực thể mẹ luôn ở phía “1” của mối quan hệ; do đó, một thanh được đặt bên cạnh thực thể LCA. Bạn có thể xác định các mối quan hệ 1:N khác trong Hình 6-1 không? Xác định các thực thể cha và con cho mỗi mối quan hệ.

Loại quan hệ thứ ba là quan hệ M:N (đọc là “nhiều với nhiều”). Trong trường hợp này, nhiều phiên bản của thực thể mẹ có thể liên quan đến nhiều phiên bản của thực thể con. Không có mối quan hệ M:N nào được hiển thị trong Hình 6-1, nhưng hãy xem Hình 6-5. Hình này cho thấy một phiên bản dự thảo ban đầu của ERD Yêu cầu Hóa chất. Trong phiên bản này, tồn tại mối quan hệ M:N giữa LCA và Chemical. Như chúng ta có thể thấy, một LCA (thực thể mẹ) có thể yêu cầu nhiều Hóa chất (ví dụ: Orthene™, Roundup™ và 2, 4-D.) và một Hóa chất (thực thể con) có thể được nhiều LCA yêu cầu. Các mối quan hệ M:N được mô tả trên ERD bằng cách có vết chân chim ở cả hai đầu của đường quan hệ. Như chúng ta sẽ tìm hiểu sau, có những lợi thế khi loại bỏ các mối quan hệ M:N khỏi ERD, vì vậy đó là lý do tại sao nó bị xóa khỏi Hình 6-1 bằng cách tạo thực thể Yêu cầu Hóa chất giữa LCA và Hóa chất. Quá trình “giải quyết” mối quan hệ M:N sẽ được giải thích ở phần sau của chương.

HÌNH 6-5
Mối quan hệ M:N



BẠN R

6-1 TÌM HIỂU CÁC YẾU TỐ CỦA MỘT ERD

XOAY

Một doanh nhân giàu có sở hữu một số lượng lớn tranh mà ông cho các viện bảo tàng trên toàn thế giới mượn. Anh ấy quan tâm đến việc thiết lập một hệ thống ghi lại những gì anh ấy cho ai vay để anh ấy không mất đầu các khoản đầu tư của mình. Anh xin được giữ thông tin về những bức tranh anh sở hữu cũng như họa sĩ đã vẽ những bức tranh đó. Anh ấy cũng muốn theo dõi các năng thơ khác nhau đã bảo lưu tác phẩm của mình, cùng với những người bảo lưu thực tế. Rõ ràng, nghệ sĩ gắn liền với tranh vẽ, tranh vẽ gắn liền với đặt chỗ trưng bày, và đặt chỗ gắn liền với bảo tàng.

CÂU HỎI:

- 1. Vẽ bốn thực thể thuộc mô hình dữ liệu này.

- 2. Cung cấp một số thuộc tính cơ bản cho từng thực thể và chọn một mã định danh, nếu có thể.
- 3. Vẽ các mối quan hệ thích hợp giữa các thực thể và đặt tên cho chúng.
- 4. Sức mạnh của mỗi mối quan hệ là gì? Mô tả điều này trên bản vẽ của bạn.
- 5. Phức tạp thức cho mỗi mối quan hệ là gì? mô tả điều này trên bản vẽ của bạn.
- 6. Liệt kê hai quy tắc kinh doanh được truyền đạt bởi bạn ERD.

Phức tạp thức Thứ hai, các mối quan hệ có phức tạp thức null hoặc không null, nghĩa là liệu một thể hiện của thực thể con có thể tồn tại mà không có một thể hiện liên quan trong thực thể mẹ hay không. Về cơ bản, phức tạp thức của một mối quan hệ cho biết liệu thực thể con có bắt buộc phải tham gia vào mối quan hệ hay không. Nó buộc bạn phải đặt những câu hỏi như, Bạn có thể yêu cầu Hóa chất mà không có Hóa chất không? và Bạn có thể có Hóa chất mà không cần Yêu cầu Hóa chất không? Phức tạp thức được mô tả bằng cách đặt số 0 trên đường quan hệ bên cạnh thực thể mẹ nếu giá trị null được cho phép. Một thanh được đặt trên đường quan hệ bên cạnh thực thể mẹ nếu null không được phép.

Trong hai câu hỏi chúng tôi vừa hỏi, câu trả lời đầu tiên là không: bạn cần một hóa chất để có yêu cầu về hóa chất. Tuy nhiên, bạn có thể có một hóa chất mà không cần yêu cầu hóa chất cho hóa chất đó. Phức tạp thức là “không rỗng” hoặc “bắt buộc” đối với mối quan hệ đầu tiên trong Hình 6-1. Tuy nhiên, hãy lưu ý rằng số 0 đã được đặt trên đường quan hệ giữa Hóa chất và Yêu cầu Hóa chất bên cạnh thực thể Yêu cầu Hóa chất. Điều này có nghĩa là hóa chất có thể tồn tại trong hệ thống của chúng tôi mà không cần phải có yêu cầu hóa chất. Nói cách khác, các trưng bày yêu cầu hóa chất là tùy chọn đối với một hóa chất. Phức tạp thức là "null."

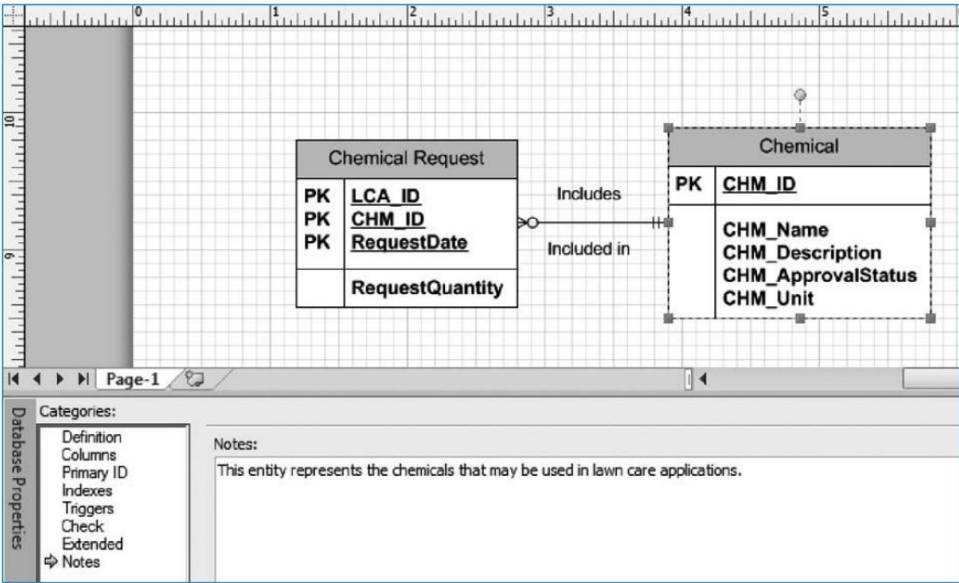
Từ điển dữ liệu và siêu dữ liệu

Như chúng tôi đã mô tả trưng bày đó, một công cụ CASE được sử dụng để giúp xây dựng ERD. Mọi công cụ CASE đều có một thứ gọi là từ điển dữ liệu, theo đúng nghĩa đen là nơi nhà phân tích đến để xác định hoặc tra cứu thông tin về các thực thể, thuộc tính và mối quan hệ trên ERD. Ngay cả Visio 2010, chủ yếu được biết đến như một công cụ vẽ, cũng có một số khả năng từ điển dữ liệu cơ bản. Hình 6-6, 6-7 và 6-8 minh họa các mục từ điển dữ liệu phổ biến cho một thực thể, một thuộc tính và một mối quan hệ; lưu ý các loại thông tin mà từ điển dữ liệu nắm bắt về từng phần tử.

Thông tin bạn nhìn thấy trong từ điển dữ liệu được gọi là siêu dữ liệu, khá đơn giản, là dữ liệu về dữ liệu. Siêu dữ liệu là bất kỳ thứ gì mô tả một thực thể, thuộc tính hoặc mối quan hệ, chẳng hạn như tên thực thể, mô tả thuộc tính và mối quan hệ

HÌNH 6-6

Mục nhập từ điển dữ liệu cho thực thể hóa học
(trong Visio 2010)



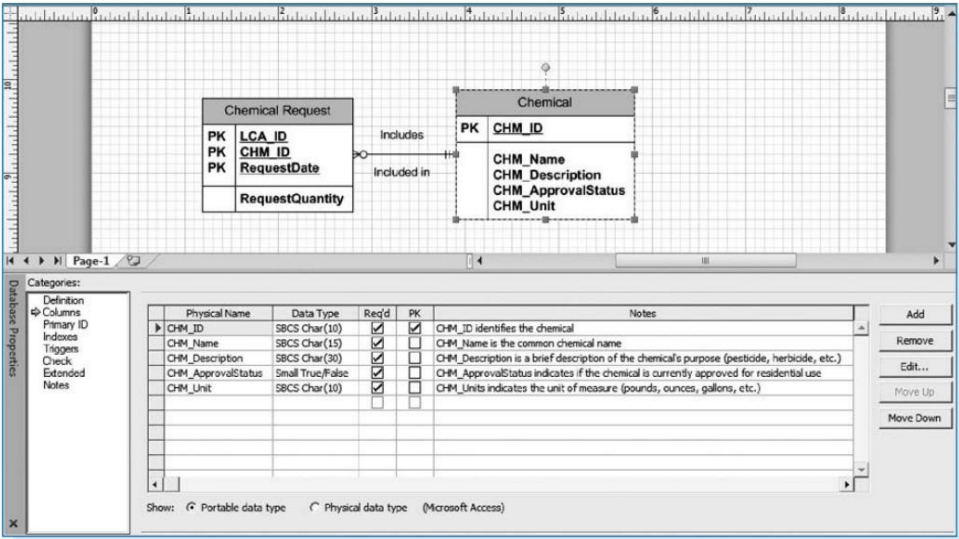
cardinality và nó được nắm bắt để giúp các nhà thiết kế hiểu rõ hơn về hệ thống mà họ đang xây dựng và để giúp người dùng hiểu rõ hơn về hệ thống mà họ sẽ sử dụng.

Hình 6-9 liệt kê các siêu dữ liệu điển hình được tìm thấy trong từ điển dữ liệu. Lưu ý rằng siêu dữ liệu có thể mô tả một thành phần ERD (như tên thực thể) và cả thông tin hữu ích cho nhóm dự án (như liên hệ người dùng, liên hệ nhà phân tích và các ghi chú đặc biệt).

Siêu dữ liệu được lưu trữ trong từ điển dữ liệu để các nhà phát triển và người dùng có thể chia sẻ và truy cập chúng trong toàn bộ SDLC. Từ điển dữ liệu cho phép bạn ghi lại các mẫu thông tin tiêu chuẩn về các yếu tố của mình ở một nơi và giúp thông tin đó có thể truy cập được ở nhiều phần của dự án. Ví dụ: các thuộc tính dữ liệu trong mô hình dữ liệu cũng xuất hiện trên các mô hình quy trình dự đoán dạng các thành phần của kho lưu trữ dữ liệu và luồng dữ liệu và trên giao diện người dùng dự đoán dạng các trường trên màn hình nhập liệu.

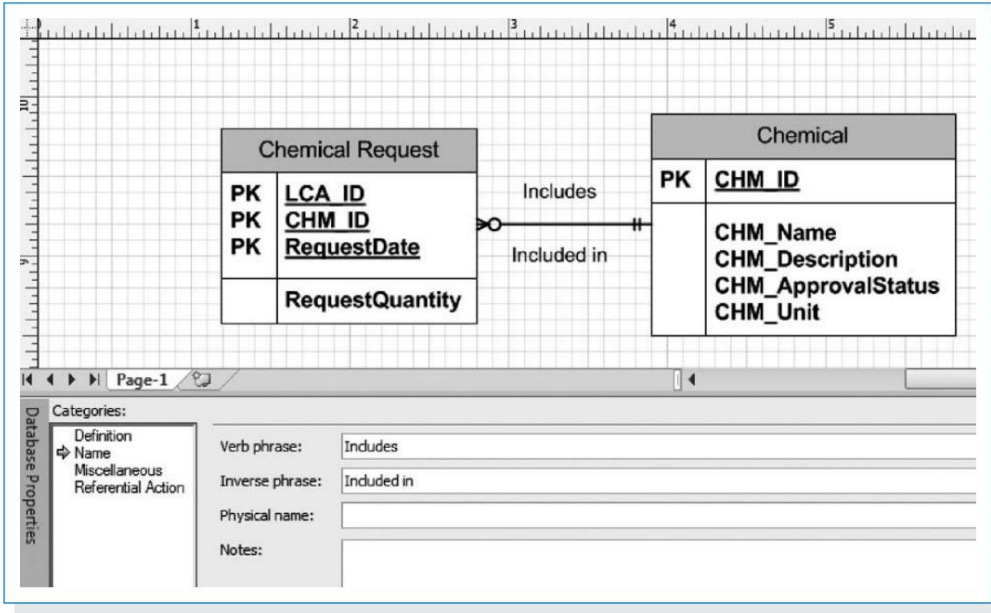
HÌNH 6-7

Nhập từ điển dữ liệu cho hóa chất
Thuộc tính (trong Visio 2010)



HÌNH 6-8

Mục từ điển dữ liệu cho một
Mối quan hệ (trong Visio 2010)



Yếu tố ERD	Các loại siêu dữ liệu	Ví dụ
thực thể	Tên	Mục
	Sự định nghĩa	Đại diện cho bất kỳ mặt hàng nào có trong kho trong siêu thị
	Ghi chú đặc biệt	Bao gồm các mặt hàng nông sản, bánh mì và đồ nguội
	Liên hệ người dùng	Nancy Keller (x6755) đứng đầu bộ phận mã hóa mặt hàng
	liên hệ phân tích	John Michaels là nhà phân tích được chỉ định cho thực thể này
Thuộc tính	Tên	Mục_UPC
	Sự định nghĩa	Mã sản phẩm chung tiêu chuẩn cho mặt hàng dựa trên Thư ơng phẩm toàn cầu
		Số do GS1 phát triển
	bí danh	Mã vạch vật phẩm
	giá trị mẫu	036000291452; 034000126453 Bất kỳ bộ số
	Giá trị chấp nhận được	12 chữ số nào Chỉ có 12 chữ số, chữ số
	Định dạng	Được lưu trữ dưới dạng giá trị chữ và
	Kiểu	số Các giá trị có chữ số đầu tiên là 2
Mối quan hệ	Ghi chú đặc biệt	được chỉ định cục bộ, đại diện cho các mặt hàng được đóng gói trong cửa hàng, chẳng hạn như thịt, bánh, sản phẩm hoặc đồ nguội. Xem Nancy Keller để biết thêm thông tin.
	Cu m đồ ng tư	bao gồm trong
	Thực thể mẹ	Mục
	Thực thể con	mặt hàng đã bán
	Sự định nghĩa	Một mặt hàng được bao gồm trong số 0 hoặc nhiều mặt hàng đã bán. Một mặt hàng đã bán bao gồm một và chỉ một mặt hàng.
	hồng y	1:N
	phư ơ ng thức	Vô giá trị
	Ghi chú đặc biệt	

HÌNH 6-9

Các loại siêu dữ liệu được thu thập bởi từ điển dữ liệu

Khi bạn thực hiện thay đổi trong từ điển dữ liệu, thay đổi sẽ ảnh hưởng đến các phần có liên quan của dự án bị ảnh hưởng.

Khi siêu dữ liệu hoàn chỉnh, rõ ràng và có thể chia sẻ, thông tin có thể được sử dụng để tích hợp các phần khác nhau của giai đoạn phân tích và cuối cùng dẫn đến một thiết kế tốt hơn nhiều. Nó trở nên chi tiết hơn nhiều khi dự án phát triển thông qua SDLC.

TAO SƠ ĐỒ MỐI QUAN HỆ THỰC THỂ

Vẽ ERD là một quá trình thử nghiệm và sửa đổi lặp đi lặp lại. Nó thường mất thực hành đáng kể. ERD có thể trở nên khá phức tạp—trên thực tế, có những hệ thống có ERD chứa hàng trăm hoặc hàng nghìn thực thể. Các bước cơ bản trong việc xây dựng một ERD là: (1) Xác định các thực thể, (2) thêm các thuộc tính thích hợp cho từng thực thể và sau đó (3) vẽ mối quan hệ giữa các thực thể để chỉ ra cách chúng được liên kết với nhau. Đầu tiên, chúng tôi sẽ mô tả ba bước trong việc tạo ERD, sử dụng ví dụ về mô hình dữ liệu từ Hình 6-1. Sau đó chúng ta sẽ thảo luận về một số khái niệm nâng cao về ERD's. Cuối cùng, chúng tôi sẽ trình bày ERD cho Nguồn điều chỉnh.

Xây dựng sơ đồ mối quan hệ thực thể Bước 1: Xác

định các thực thể Như chúng tôi đã giải thích, cách phổ biến nhất để bắt đầu ERD là truy cập tiên xác định các thực thể cho mô hình và các thuộc tính của chúng. Các thực thể phải đại diện cho các loại thông tin chính mà bạn cần lưu trữ trong hệ thống của mình. Nếu bạn bắt đầu mô hình dữ liệu của mình bằng cách sử dụng một ca sử dụng, hãy xem các đầu vào chính của ca sử dụng, các đầu ra chính và thông tin được sử dụng cho các bước của ca sử dụng. Nếu các mô hình quy trình (ví dụ: DFD) đã được chuẩn bị, thì cách dễ nhất để bắt đầu là bắt đầu với chúng: Dữ liệu lưu trữ trên DFD, các thực thể bên ngoài và luồng dữ liệu cho biết các loại thông tin được thu thập và truyền qua hệ thống .

Yêu cầu Hóa chất đóng một vai trò quan trọng trong ví dụ về hệ thống yêu cầu hóa chất của chúng tôi và do đó được bao gồm dưới dạng thực thể dữ liệu. Ngoài ra, bản thân Hóa chất sẽ cần được mô tả bằng dữ liệu và do đó cũng sẽ được đưa vào dưới dạng thực thể dữ liệu. Cuối cùng, chúng tôi sẽ cần nắm bắt thông tin về những người sử dụng dịch vụ chăm sóc cò (LCA), vì những cá nhân này là những tác nhân chính trong hệ thống.

Bước 2: Thêm thuộc tính và gán định danh Thông tin mô tả mỗi thực thể trở thành thuộc tính của nó. Có khả năng là bạn đã xác định được một vài thuộc tính nếu bạn đọc các truy cập hợp sử dụng hệ thống yêu cầu hóa chất và chú ý đến các luồng thông tin trên DFD của chúng. Ví dụ: LCA có tên và hóa chất có tên

BẠN R

6-2 ĐÁNH GIÁ CÔNG CỤ TÌNH HUỐNG CỦA BẠN

XOAY

Kiểm tra công cụ CASE mà bạn sẽ sử dụng cho dự án của mình hoặc tìm công cụ CASE trên Web mà bạn muốn tìm hiểu. Loại nào

siêu dữ liệu có nắm bắt được từ điển dữ liệu của nó không? Công cụ CASE có tích hợp thông tin mô hình dữ liệu với các phần khác của dự án không? Làm sao?

HÌNH 6-10

Các yếu tố của yêu cầu hóa chất mới
Dòng dữ liệu

Tên luồng dữ liệu: Yêu cầu hóa chất mới				
Các yếu tố dữ liệu:	ID LCA	ID hóa chất	Ngày yêu cầu	Số lưu trữ

và mô tả. Thật không may, nhiều thông tin từ các mô hình quy trình và tư vấn hợp sử dụng không bao gồm đủ chi tiết để xác định các thuộc tính chính xác nên tồn tại trong mỗi thực thể của chúng tôi.

Trong một dự án thực tế, có một số nơi bạn có thể truy cập để tìm ra thuộc tính nào thuộc về thực thể của mình. Ví dụ, bạn có thể kiểm tra trong công cụ CASE-thông thư ờng, một nhà phân tích sẽ mô tả chi tiết luồng dữ liệu của mô hình quy trình khi họ nhập luồng dữ liệu vào kho lưu trữ CASE. Ví dụ: một nhà phân tích có thể tạo một mục nhập cho luồng dữ liệu thông tin yêu cầu hóa chất giống như mục được hiển thị trong Hình 6-10, trong đó liệt kê bốn thành phần dữ liệu tạo nên thông tin yêu cầu hóa chất. Các thành phần của luồng dữ liệu phải được thêm vào ERD dưới dạng thuộc tính trong các thực thể của bạn. Cách tiếp cận thứ hai là kiểm tra định nghĩa yêu cầu. Thông thư ờng, có một phần dưới yêu cầu chức năng được gọi là yêu cầu dữ liệu. Phần này mô tả các nhu cầu dữ liệu cho hệ thống đã được xác định trong khi các yêu cầu được thu thập. Cách tiếp cận cuối cùng để xác định các thuộc tính là sử dụng các kỹ thuật khơi gợi yêu cầu. Các kỹ thuật hiệu quả nhất sẽ là phỏng vấn (ví dụ: hỏi những người tạo và sử dụng báo cáo về nhu cầu dữ liệu của họ) hoặc phân tích tài liệu (ví dụ: kiểm tra các biểu mẫu, báo cáo hoặc màn hình nhập hiện có).

Khi các thuộc tính được xác định, một hoặc nhiều trong số chúng sẽ trở thành định danh của thực thể. Mã định danh phải là (các) thuộc tính có thể xác định duy nhất một phiên bản duy nhất của thực thể. Nhìn vào Hình 6-1 và chú ý các mã định danh đã được chọn cho mỗi thực thể.

Bước 3: Xác định các mối quan hệ Bước cuối cùng trong việc tạo ERD là xác định xem các thực thể có liên quan với nhau như thế nào. Các đường được vẽ giữa các thực thể có mối quan hệ, mỗi mối quan hệ được gắn nhãn, và số lượng và phức tạp được chỉ định. Cách tiếp cận đơn giản nhất là bắt đầu với một thực thể và xác định tất cả các thực thể mà nó chia sẻ mối quan hệ. Trong ví dụ của chúng ta ở Hình 6-1, chúng ta có thể thấy rằng một LCA đưa ra các yêu cầu về hóa chất và một hóa chất được bao gồm trong một yêu cầu về hóa chất.

Khi bạn tìm thấy một mối quan hệ để đưa vào mô hình, bạn cần xác định tính chính xác và phức tạp của nó. Đối với số lượng, hãy hỏi có bao nhiêu phiên bản của mỗi thực thể tham gia vào mối quan hệ. Bạn biết rằng một LCA có thể đưa ra nhiều yêu cầu hóa chất, nhưng một yêu cầu hóa chất cụ thể chỉ được thực hiện bởi một LCA. Do đó, chúng tôi đặt một vết chân chim bên cạnh thực thể yêu cầu hóa chất và một thanh duy nhất gần thực thể LCA nhất. Điều này cho thấy rằng có mối quan hệ 1:N trong đó LCA là thực thể mẹ ("1") và yêu cầu hóa chất là thực thể con ("nhiều").

Tiếp theo, chúng tôi kiểm tra phức tạp của mối quan hệ. LCA có thể tồn tại mà không có yêu cầu hóa chất liên quan không? Trong ví dụ của chúng tôi, câu trả lời là "có", vì vậy phức tạp là "null" hoặc không bắt buộc. Số 0 được đặt cạnh vết chân chim gần yêu cầu hóa chất. Bây giờ, liệu một yêu cầu hóa chất có thể tồn tại mà không có LCA liên quan không? Câu trả lời này là "không", vì vậy phức tạp là "không rỗng": hoặc bắt buộc, và chúng tôi đặt một thanh duy nhất bên cạnh thực thể LCA. Kiểu suy nghĩ tư duy được áp

dụng để xác định bản chất và phức tạp của mối quan hệ giữa hóa chất và yêu cầu hóa chất. Một hóa chất (gốc) có thể được đưa vào nhiều yêu cầu hóa chất (đứa trẻ), vì vậy mối quan hệ là 1:N. Một hóa chất có thể tồn tại mà không cần yêu cầu hóa chất, vì vậy phức tạp là "null";

tuy nhiên, một yêu cầu hóa học đòi hỏi phải có sự tồn tại của một hóa chất, vì vậy phương thức này là “không vô hiệu”.

Một lần nữa, hãy nhớ rằng mô hình hóa dữ liệu là một quá trình lặp đi lặp lại. Thông thường, các giả định bạn đưa ra và các quyết định bạn đưa ra sẽ thay đổi khi bạn tìm hiểu thêm về các yêu cầu nghiệp vụ và khi các trường hợp sử dụng và mô hình quy trình thay đổi. Nhưng bạn phải bắt đầu từ đâu đó—vì vậy hãy làm tốt nhất có thể với ba bước chúng tôi vừa mô tả và tiếp tục lặp đi lặp lại cho đến khi bạn có một mô hình hoạt động. Ở phần sau của chương này, chúng tôi sẽ chỉ cho bạn một số cách để xác thực các ERD mà bạn vẽ.

Cú pháp nâng cao

Bây giờ chúng ta đã tạo một mô hình dữ liệu theo cú pháp cơ bản đã được trình bày trước đó, chúng ta có thể chuyển sang một số khái niệm nâng cao. Chúng tôi sẽ giải thích ba loại thực thể đặc biệt và chỉ ra cách chúng có thể được sử dụng trong hệ thống Yêu cầu Hóa chất của chúng tôi.

Thực thể độc lập Một thực thể độc lập là một thực thể có thể tồn tại mà không cần sự trợ giúp của một thực thể khác, chẳng hạn như Lawn Chemical Applicator và Chemical. Tất cả các thực thể này đều có số nhận dạng được tạo từ các thuộc tính của chính chúng. Các thuộc tính từ các thực thể khác không cần thiết để xác định duy nhất các thể hiện của các thực thể này.

Các thực thể độc lập được vẽ với dư ới dạng hình chữ nhật với một dư ờng viền duy nhất.

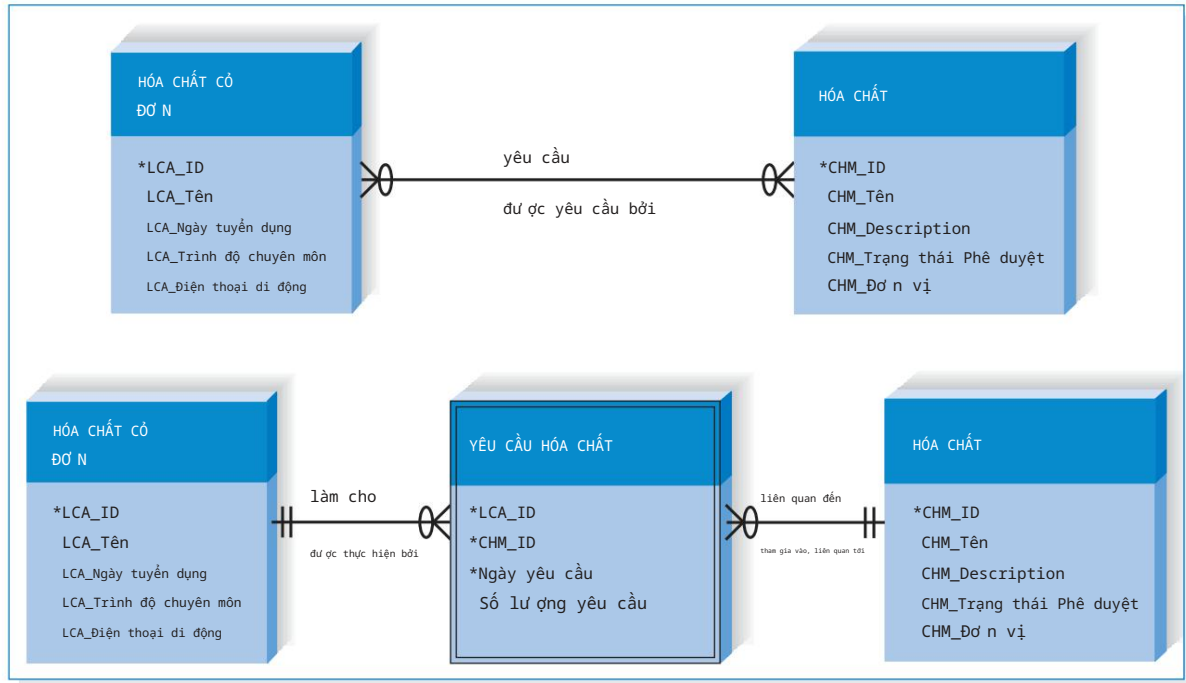
Khi một mối quan hệ bao gồm một thực thể con độc lập, nó được gọi là mối quan hệ không xác định. Tên này bắt nguồn từ thực tế là các thuộc tính của thực thể mẹ không cần thiết như một phần của mã định danh của thực thể con.

Thực thể phụ thuộc Có những tình huống khi một thực thể con yêu cầu các thuộc tính từ thực thể mẹ để xác định duy nhất một thực thể. Trong những trường hợp này, thực thể con được gọi là thực thể phụ thuộc và mã định danh của nó bao gồm ít nhất một thuộc tính từ thực thể mẹ.

Một ví dụ điển hình về thực thể phụ thuộc là thực thể Yêu cầu hóa chất được hiển thị trong Hình 6-1. Yêu cầu Hóa chất được thực hiện bởi một LCA cụ thể và bao gồm một hóa chất cụ thể. Chúng tôi bao gồm LCA_ID và Chemical_ID cộng với ngày yêu cầu để xác định đầy đủ từng Yêu cầu Hóa chất, Yêu cầu Hóa chất được coi là một thực thể phụ thuộc và được hiển thị dư ới dạng hình chữ nhật có dư ờng viền kép.

Khi các mối quan hệ có một thực thể con phụ thuộc, chúng được gọi là các mối quan hệ xác định. Tên này bắt nguồn từ thực tế là các thuộc tính của thực thể mẹ là cần thiết như một phần của mã định danh của thực thể con.

Thực thể Giao lộ Loại thực thể thứ ba là thực thể giao lộ. Nó tồn tại để nắm bắt một số thông tin về mối quan hệ giữa hai thực thể khác. Thông thường, các thực thể giao điểm được thêm vào mô hình dữ liệu để lưu trữ thông tin về hai thực thể chia sẻ mối quan hệ M:N. Các thực thể này còn được gọi là các thực thể liên kết. Hãy nhớ lại mối quan hệ M:N giữa LCA và Hóa chất được minh họa trong Hình 6-5. Trong hình đó, một phiên bản LCA có thể yêu cầu nhiều Hóa chất và một Hóa chất có thể được yêu cầu bởi nhiều LCA. Một khó khăn phát sinh nếu chúng tôi muốn nắm bắt ngày mà một hóa chất cụ thể được yêu cầu bởi một LCA cụ thể. Chúng tôi không thể đặt ngày trong thực thể Hóa chất, vì Hóa chất được yêu cầu bởi nhiều LCA. Chúng tôi cũng không thể đưa ngày vào thực thể LCA vì có nhiều Hóa chất được LCA yêu cầu. Do đó, chúng tôi cần một thực thể khác cho phép chúng tôi liên kết một hóa chất cụ thể với một LCA cụ thể vào một ngày cụ thể.



HÌNH 6-11

Giải quyết mối quan hệ M:N

Quá trình thêm một thực thể giao nhau được gọi là "giải quyết mối quan hệ M:N" vì nó loại bỏ mối quan hệ M:N và các vấn đề liên quan của nó khỏi mô hình dữ liệu. Có ba bước liên quan đến việc thêm một thực thể giao lộ.

Bước 1: Xóa đường quan hệ M:N và chèn một thực thể mới vào giữa hai thực thể hiện có.

Bước 2: Thêm hai mối quan hệ 1:N vào mô hình. Hai thực thể ban đầu sẽ đóng vai trò là thực thể mẹ cho mỗi 1:N và thực thể giao điểm mới trở thành thực thể con trong cả hai mối quan hệ. Bước 3: Đặt tên cho thực thể giao lộ.

Các thực thể giao lộ thường được đặt tên bằng cách ghép hai thực thể đã tạo ra nó (ví dụ: Yêu cầu hóa học), làm cho ý nghĩa của nó trở nên rõ ràng. Ngoài ra, thực thể có thể được đặt tên thích hợp khác. Hình 6-11 cho thấy mối quan hệ M:N LCA-Hóa chất và cách nó được giải quyết với việc sử dụng một thực thể giao nhau.

Các thực thể giao lộ phụ thuộc hay độc lập? Trên thực tế, nó phụ thuộc.

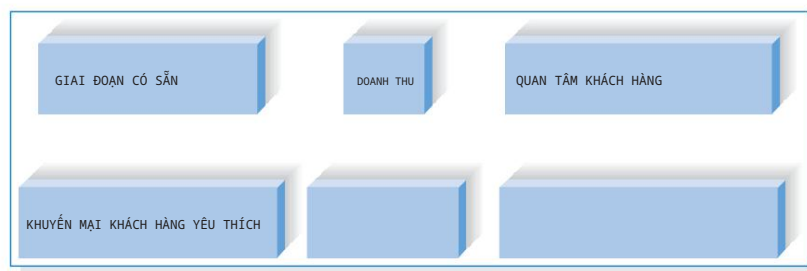
Đôi khi một thực thể giao điểm có một mã định danh logic có thể xác định duy nhất các thực thể của nó. Ví dụ, một thực thể giao nhau giữa một sinh viên và một khóa học (một sinh viên có thể học nhiều khóa học và một khóa học được nhiều sinh viên tham gia) có thể được gọi là bảng điểm. Nếu bảng điểm có số bảng điểm duy nhất, thì thực thể sẽ được coi là độc lập. Ngược lại, thực thể giao điểm Yêu cầu Hóa chất trong Hình 6-11 yêu cầu các mã định danh từ cả LCA và Hóa chất cho một phiên bản phải được xác định duy nhất. Do đó, Yêu cầu hóa chất là một thực thể phụ thuộc.

Áp dụng các khái niệm tại Tune Source

Hãy xem thêm một ví dụ về việc tạo mô hình dữ liệu bằng cách sử dụng ngữ cảnh của Nguồn điều chỉnh. Hiện tại, hãy xem lại các trường hợp sử dụng được trình bày trong Hình 4-14 và mô hình quy trình cấp 0 cuối cùng được trình bày trong Hình 5-17.

HÌNH 6-12

Các thực thể cho Nguồn điều chỉnh ERD



Xác định các thực thể Khi bạn kiểm tra DFD nguồn giai điệu 0, bạn thấy rằng có sáu kho lưu trữ dữ liệu: khách hàng, giảm giá, giai điệu có sẵn, sở thích của khách hàng, mục yêu thích của khách hàng và khuyến mãi được nhắm mục tiêu. Mỗi loại dữ liệu duy nhất này có khả năng sẽ được đại diện bởi các thực thể trên một mô hình dữ liệu.

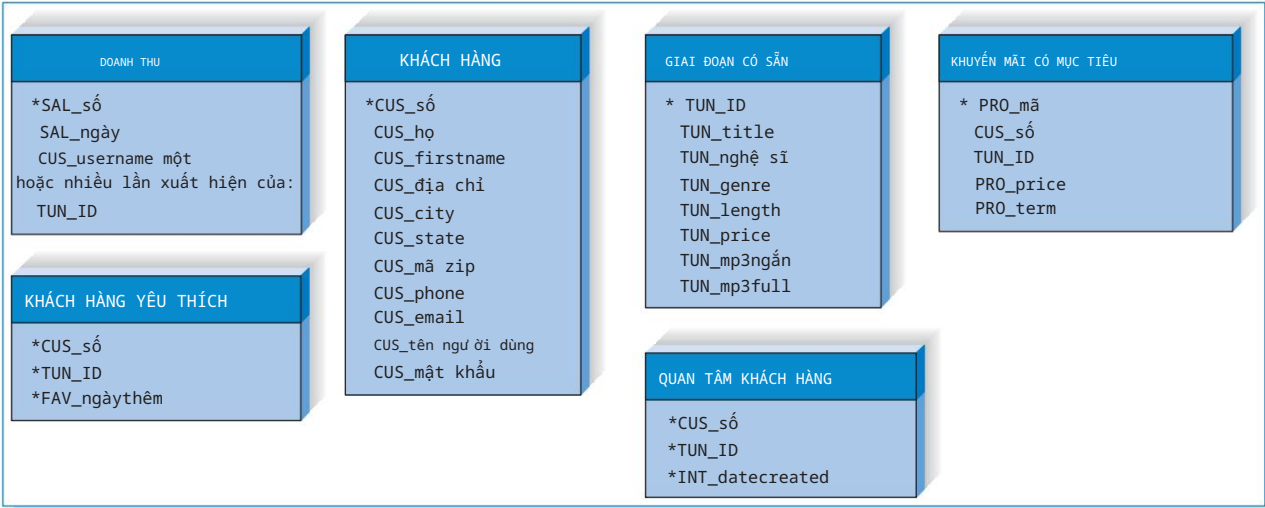
Bước tiếp theo, bạn nên kiểm tra các thực thể bên ngoài và tự hỏi: “Liệu hệ thống có cần thu thập thông tin về bất kỳ thực thể nào trong số này không?” Bạn có thể muốn bao gồm các nhà quản lý tiếp thị, nhưng thực sự không cần phải theo dõi thông tin về những người này trong hệ thống của chúng tôi. Sau đó, chúng tôi có thể muốn theo dõi người dùng hệ thống, chuyển từ và đặc quyền truy cập dữ liệu, nhưng thông tin này liên quan đến việc sử dụng hệ thống mới và sẽ không được thêm cho đến khi mô hình dữ liệu vật lý được tạo trong giai đoạn thiết kế.

Bạn cũng nên xem xét các luồng dữ liệu trên mô hình quy trình của mình và đảm bảo rằng tất cả thông tin chảy qua hệ thống đã được bao phủ bởi ERD của bạn. Có vẻ như các thực thể chính cho Nguồn điều chỉnh đã được xác định sau khi kiểm tra các kho lưu trữ dữ liệu và các thực thể bên ngoài. Xem Hình 6-12 để bắt đầu mô hình dữ liệu của chúng tôi.

Xác định các thuộc tính Bước tiếp theo là chọn những thuộc tính nào sẽ được sử dụng để mô tả mỗi thực thể. Có khả năng là bạn đã xác định được một số thuộc tính nếu bạn đọc các từ vựng hợp sử dụng Nguồn điều chỉnh và kiểm tra các DFD. Ví dụ: một giai điệu có sẵn có nghệ sĩ, tiêu đề, thể loại và độ dài, đồng thời một số thuộc tính của khách hàng là tên và thông tin liên hệ, có thể bao gồm địa chỉ, số điện thoại và địa chỉ email.

Thoạt nhìn, hai thực thể khách hàng yêu thích và sở thích của khách hàng có vẻ giống nhau, nhưng chúng được sử dụng để nắm bắt các loại thông tin khác nhau về sở thích âm nhạc của khách hàng. Yêu thích của khách hàng là một giai điệu mà khách hàng thêm cụ thể vào danh sách Yêu thích của mình để theo dõi các giai điệu khi trang Web được tìm kiếm và duyệt qua. Theo một nghĩa nào đó, nó giống như một danh sách mua sắm trong tương lai, vì vậy chúng tôi chỉ ghi lại ID của khách hàng, ID của giai điệu và ngày giai điệu được thêm vào danh sách. Mục Yêu thích của khách hàng luôn có sẵn mỗi khi khách hàng truy cập lại trang web để giúp nhớ lại các giai điệu đã khám phá trước đó và (hy vọng) mua chúng. Mặt khác, sự quan tâm của khách hàng được tạo tự động khi khách hàng đầu tư vào giai điệu và nghe mẫu. Sở thích của khách hàng được bộ phận tiếp thị sử dụng để giúp thiết kế các chương trình khuyến mãi cho khách hàng sẽ phù hợp với các loại nhạc mà khách hàng đã khám phá. Các thuộc tính hơi khác nhau được liên kết với hai thực thể này do các mục đích khác nhau của chúng trong hệ thống.

Các chương trình khuyến mãi được nhắm mục tiêu là các ưu đãi đặc biệt sẽ được tạo cho khách hàng trên cơ sở sở thích của họ và liên quan đến mô hình bán hàng. Một chương trình khuyến mãi sẽ bao gồm giá bán cho một giai điệu cụ thể nếu nó được mua trong một giai điệu cụ thể.



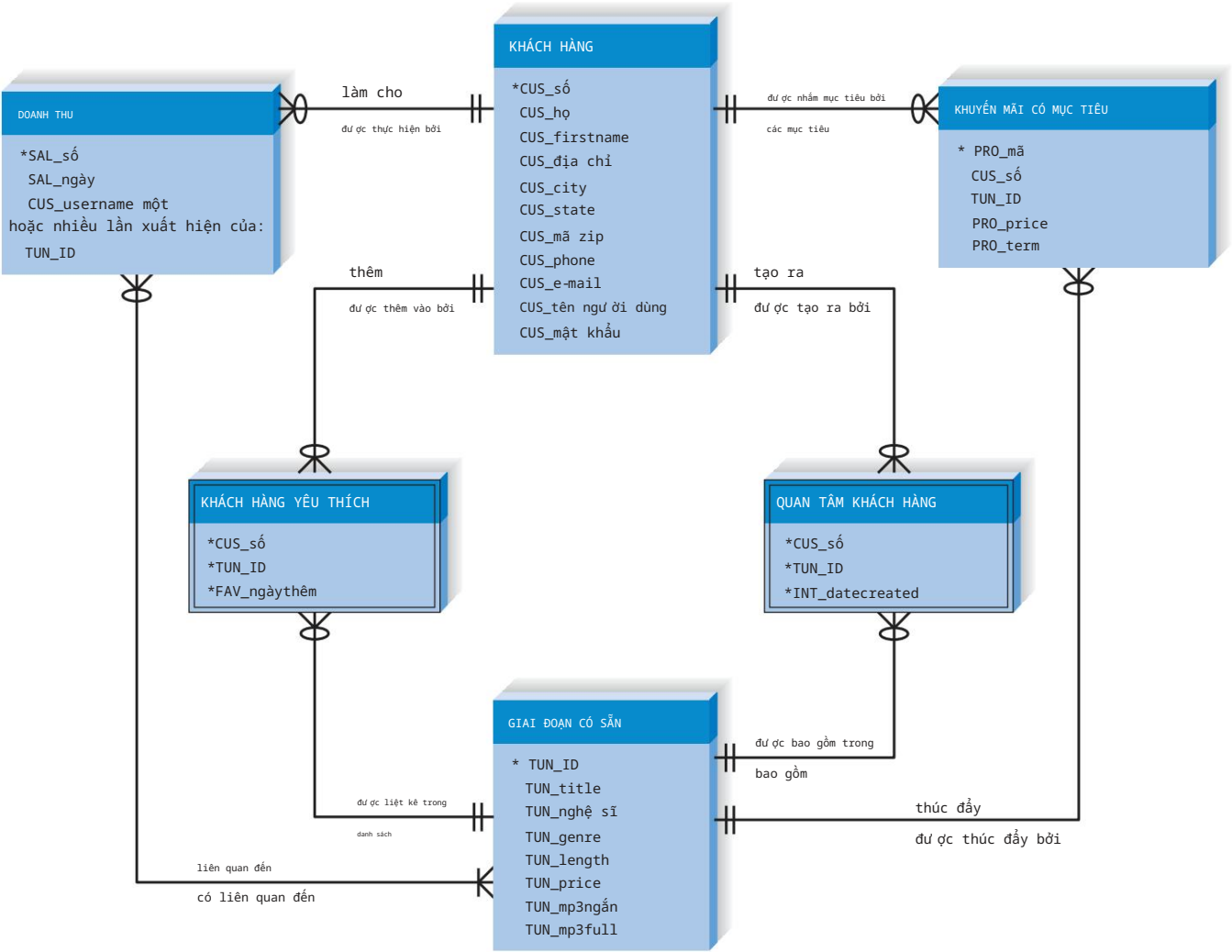
HÌNH 6-13 Thuộc tính và định danh cho nguồn điều chỉnh ERD

khung thời gian. Các thuộc tính cho các chương trình khuyến mãi được nhắm mục tiêu được liệt kê trong Hình 6-13. Cuối cùng, chúng tôi cũng thấy rằng một số thuộc tính liên quan đến việc bán giai điệu đã được liệt kê trong ERD.

Để xác định các số nhận dạng thực thể, chúng tôi xem xét thuộc tính hoặc các thuộc tính sẽ xác định duy nhất từng thực thể. Chúng tôi sẽ thiết lập mã số khách hàng cho từng khách hàng trong hệ thống. Mỗi giai điệu có sẵn mà chúng tôi có sẽ được chỉ định một ID giai điệu duy nhất. Mỗi khuyến mãi được nhắm mục tiêu sẽ được cấp một mã khuyến mãi duy nhất và mỗi lần bán hàng sẽ được cấp một số bán hàng duy nhất. Cuối cùng, mỗi mục yêu thích của khách hàng và mỗi mối quan tâm của khách hàng có thể được xác định duy nhất bằng mã số khách hàng, ID giai điệu và ngày tạo.

Các thực thể khách hàng, bán hàng, giai điệu có sẵn và quảng cáo được nhắm mục tiêu là các thực thể độc lập; các thuộc tính từ các thực thể khác không cần thiết để xác định duy nhất các thể hiện. Tuy nhiên, các mã định danh cho sở thích của khách hàng và mục yêu thích của khách hàng dựa trên các thuộc tính từ các thực thể mẹ của chúng: khách hàng và giai điệu có sẵn. Điều này là do khách hàng yêu thích (hoặc sở thích của khách hàng) được xác định duy nhất bởi khách hàng đã tạo ra nó, giai điệu liên quan và ngày tạo ra nó. Do đó, vì hai thực thể này lấy một phần khóa chính của chúng từ các thực thể mẹ nên chúng được coi là các thực thể phụ thuộc.

Xác định các mối quan hệ Được cuối cùng trong việc tạo ERD là xác định cách các thực thể có quan hệ với nhau. Các đường được vẽ giữa các thực thể có mối quan hệ và mỗi mối quan hệ được gắn nhãn và gán một số lượng và phương thức. Như thể hiện trong Hình 6-14, một khách hàng có thể thực hiện nhiều lần bán hàng, nhưng một lần bán hàng được thực hiện bởi một khách hàng. Việc bán hàng là không bắt buộc đối với một trữ ứng hợp khách hàng cụ thể, nhưng một khách hàng là bắt buộc để bán hàng. Một khách hàng có thể được nhắm mục tiêu bởi nhiều chương trình khuyến mãi được nhắm mục tiêu, nhưng một chương trình khuyến mãi được nhắm mục tiêu là dành cho một khách hàng. Không bắt buộc phải có khuyến mãi tar got cho khách hàng, nhưng phải có khách hàng cho khuyến mãi được nhắm mục tiêu. Một khách hàng tạo nhiều mục yêu thích, nhưng một mục yêu thích chỉ được tạo bởi một khách hàng. Một yêu thích là không cần thiết cho một khách hàng, nhưng một khách hàng được yêu cầu cho một yêu thích. Một giai điệu có sẵn có thể được bao gồm trong nhiều mục yêu thích của khách hàng, nhưng một mục yêu thích chỉ bao gồm một giai điệu. Yêu thích là không cần thiết cho một



HÌNH 6-14
Mối quan hệ cho Nguồn điều chỉnh ERD

giai điệu có sẵn, như ng một giai điệu là cần thiết cho một mục yêu thích. (Các mối quan hệ tư ơng tự áp dụng cho giai điệu sẵn có của khách hàng-quan tâm). Chúng ta cũng có thể thấy rằng một giai điệu có sẵn có thể đư ợc quảng cáo bởi nhiều quảng cáo đư ợc nhắm mục tiêu, như ng một quảng cáo đư ợc nhắm mục tiêu chỉ quảng bá một giai điệu. Giai điệu có sẵn không bắt buộc phải có quảng cáo đư ợc nhắm mục tiêu, như ng quảng cáo đư ợc nhắm mục tiêu phải đư ợc liên kết với giai điệu có sẵn. Cuối cùng, chúng tôi thấy rằng khách hàng có thể đặt nhiều lần bán hàng, như ng không bắt buộc phải bán hàng cho một khách hàng. Một giao dịch bán thuộc về một và chỉ một khách hàng. Một đợt giảm giá có thể bao gồm nhiều giai điệu và một giai điệu có thể đư ợc đư a vào nhiều đợt giảm giá. Một giai điệu là bắt buộc khi bán hàng, như ng một giai điệu thì không cần phải bán hàng.

Các mối quan hệ giữa khách hàng-khách hàng yêu thích, khách hàng yêu thích-giai điệu có sẵn, khách hàng-sự quan tâm của khách hàng và mối quan hệ giữa sở thích của khách hàng-giai điệu có sẵn là các mối quan hệ xác định. Tất cả các mối quan hệ khác là mối quan hệ không xác định.

Là bư ớc cuối cùng trong quá trình tạo Tune Source ERD, chúng ta nên giải quyết mọi mối quan hệ M:N trong mô hình dữ liệu. Nhìn vào Hình 6-14 cho thấy một mối quan hệ như vậy, giữa việc bán hàng và giai điệu có sẵn. Xem Lư ợt của bạn 6-6 và tự mình giải quyết mối quan hệ này.

XÁC NHẬN MỘT ERD

Như bạn có thể đoán từ phần trước, việc tạo ERD khá khó khăn. Cần rất nhiều kinh nghiệm để vẽ ERD tốt và không có nhiều quy tắc đen trắng để giúp hướng dẫn bạn. May mắn thay, có một số hướng dẫn thiết kế chung mà bạn có thể ghi nhớ khi xây dựng ERD và sau khi ERD được vẽ, bạn có thể sử dụng một kỹ thuật gọi là chuẩn hóa để xác thực rằng các mô hình của bạn đã được định dạng tốt. Một kỹ thuật khác là kiểm tra ERD của bạn với các mô hình quy trình của bạn để đảm bảo rằng cả hai mô hình cân bằng lẫn nhau.

Hướng dẫn thiết kế

Hướng dẫn thiết kế không phải là quy tắc phải tuân theo; đúng hơn, chúng là “những phương pháp hay nhất” thường dẫn đến những sơ đồ có chất lượng tốt hơn. Ví dụ: nhãn và quy ước đặt tên rất quan trọng để tạo ERD rõ ràng. Tên không nên mơ hồ (ví dụ: tên, số); thay vào đó, họ nên truyền đạt rõ ràng thành phần mô hình đại diện cho điều gì. Những tên này phải nhất quán trong toàn bộ mô hình và phản ánh thuật ngữ mà doanh nghiệp sử dụng. Nếu Tune Source đề cập đến những người đặt hàng nhạc với tư cách là khách hàng, thì mô hình dữ liệu phải bao gồm một thực thể được gọi là khách hàng, không phải khách hàng hoặc bên liên quan. Không có quy tắc nào về cách bố trí các thành phần ERD. Chúng có thể được đặt ở bất cứ đâu bạn thích trên trang, mặc dù hầu hết các nhà phân tích hệ thống đều cố gắng đặt

CÁC KHÁI NIỆM	6-A VAI TRÒ CỦA NGƯỜI DÙNG TRONG MÔ HÌNH DỮ LIỆU
TRONG HÀNH ĐỘNG	<div><p>Tôi có hai câu chuyện rất khác nhau về mô hình dữ liệu. Đầu tiên, khi tôi làm việc với First American Corporation, người đứng đầu bộ phận Tiếp thị đã treo một mô hình dữ liệu cho các hệ thống tiếp thị trên tường trong văn phòng của cô ấy. Tôi nghĩ điều này hơi bất thường đối với một giám đốc điều hành cấp cao, nhưng cô ấy giải thích với tôi rằng dữ liệu rất quan trọng đối với hầu hết các sáng kiến mà cô ấy đưa ra. Trước khi cô ấy có thể phê duyệt một chiến dịch tiếp thị hoặc chiến lược mới, cô ấy muốn xác nhận rằng dữ liệu tồn tại trong hệ thống và các nhà phân tích của cô ấy có thể truy cập dữ liệu đó. Cô ấy đã hiểu rất rõ về ERD trong những năm qua vì chúng là một công cụ giao tiếp quan trọng để cô ấy sử dụng với người của mình và với CNTT.</p><p>Ở một lưu ý rất khác, đây là câu chuyện mà tôi nhận được từ một người bạn của tôi, người đứng đầu bộ phận CNTT: “Chúng tôi đang làm việc với nỗ lực phát triển kinh doanh quan trọng, phụ thuộc vào thời gian và RẤT NHIỀU quản lý cấp cao đã quyết định rằng cách để đảm bảo thành công là để các nhóm khác nhau hướng dẫn thiết kế kỹ thuật cho ban quản lý cấp cao hàng tuần. Nhóm của tôi chịu trách nhiệm về kiến trúc dữ liệu và thiết kế cơ sở dữ liệu. Làm thế nào quản lý cấp cao có thể, không ai trong số họ</p><p>có lẽ đã từng thiết kế một kiến trúc Oracle, hãy đánh giá tính hợp lý trong công việc của chúng tôi?</p><p>Vì vậy, tôi đã yêu cầu nhân viên của mình chuẩn bị những điều sau đây cho hướng dẫn thiết kế một (và duy nhất) mà nhóm của chúng tôi được yêu cầu thực hiện. Đầu tiên, chúng tôi đã hợp nhất một số mô hình dữ liệu hiện có và saogđĩa là, chép từng tệp vào mỗi quan hệ được in hai lần (bất cứ lúc nào, nếu được hỏi, kiến trúc dự phòng). Sau đó, chúng tôi đánh mã màu mô hình một cách phức tạp và in mô hình ra trên máy vẽ và in một bản sao của mỗi inch tài liệu mô hình mà chúng tôi có.</p><p>Vào ngày đánh giá, tôi chỉ cần lật tài liệu và kéo căng mô hình đã vẽ trên bàn của phòng giám đốc điều hành. 'Có câu hỏi nào không,' tôi hỏi?</p><p>"Rất ấn tượng," họ trả lời. Điều đó là vậy đó! Các thiết kế của tôi không bao giờ bị nghi ngờ nữa." Barbara Wixom</p><p>CÂU HỎI:</p><ol style="list-style-type: none">Từ hai câu chuyện này, bạn nghĩ vai trò của người dùng trong mô hình hóa dữ liệu là gì?Khi nào thì thích hợp để người dùng tham gia vào quá trình tạo ERD?Người dùng có thể giúp nhà phân tích tạo ERD tốt hơn bằng cách nào?</div>

BẠN R

6-3 HỆ THỐNG KHU VỰC NHÀ Ở

XOAY

Hãy xem xét hệ thống đi kèm, đư ợc mô tả trong Chương 4. Sử dụng các trư ờng hợp sử dụng và mô hình quy trình mà bạn đã tạo trong Chương 4 và 5 để giúp bạn trả lời các câu hỏi tiếp theo.

Dịch vụ nhà ở tại trư ờng giúp sinh viên tìm căn hộ. Chủ sở hữu căn hộ điền vào các mẫu thông tin về các đ ơn vị cho thuê mà họ có sẵn (ví dụ: vị trí, số phòng ngủ, tiền thuê hàng tháng). Sinh viên đăng ký dịch vụ có thể tìm kiếm thông tin cho thuê để tìm căn hộ đáp ứng nhu cầu của họ (ví dụ: căn hộ hai phòng ngủ với giá 800 đô la trở xuống mỗi tháng trong bán kính 1/2 dặm từ khuôn viên trư ờng). Sau đó, họ liên hệ với căn hộ

chủ sở hữu trực tiếp để xem căn hộ và có thể thuê nó. Chủ sở hữu căn hộ gọi cho dịch vụ để xóa danh sách của họ khi họ đã thuê (các) căn hộ của mình.

CÂU HỎI:

1. Bạn sẽ đ ưa những thực thể nào vào một mô hình dữ liệu?
2. Bạn sẽ liệt kê những thuộc tính nào cho mỗi thực thể? Chọn một mã định danh cho mỗi thực thể, nếu có thể.
3. Mỗi quan hệ nào tồn tại giữa các thực thể mà bạn đã xác định? Gắn nhãn các mối quan hệ một cách thích hợp và biểu thị số lượng và phư ơ ng thức của từng mối quan hệ.

các thực thể có quan hệ với nhau. Nếu mô hình trở nên quá phức tạp hoặc bận rộn (một số công ty có hàng trăm thực thể trên một mô hình dữ liệu), thì mô hình có thể đư ợc chia thành các lĩnh vực chủ đề. Mỗi lĩnh vực chủ đề sẽ chứa các thực thể và mối quan hệ liên quan và nhà phân tích có thể làm việc với một nhóm thực thể tại một thời điểm để làm cho quá trình mô hình hóa bớt khó hiểu hơn.

Nói chung, mô hình hóa dữ liệu có thể khá phức tạp, chủ yếu là do mô hình dữ liệu chủ yếu dựa trên diễn giải; do đó, khi các quy tắc kinh doanh thay đổi, các mối quan hệ hoặc các thành phần mô hình dữ liệu khác sẽ phải thay đổi. Các giả định là một phần quan trọng của mô hình hóa dữ liệu. Điều quan trọng là chúng tôi xác minh tất cả các giả định về quy tắc kinh doanh để mô hình dữ liệu của chúng tôi là chính xác.

BẠN R

6-4 CÁC THỰC THỂ ĐỘC LẬP

XOAY

Xác định vị trí các thực thể độc lập trên Hình 6-14. Làm thế nào để bạn biết những thực thể nào là độc lập? Xác định vị trí các mối quan hệ không xác định.

Làm thế nào mà bạn tìm thấy chúng? Bạn có thể tạo một quy tắc mô tả mối liên kết giữa các thực thể độc lập và các mối quan hệ không xác định không?

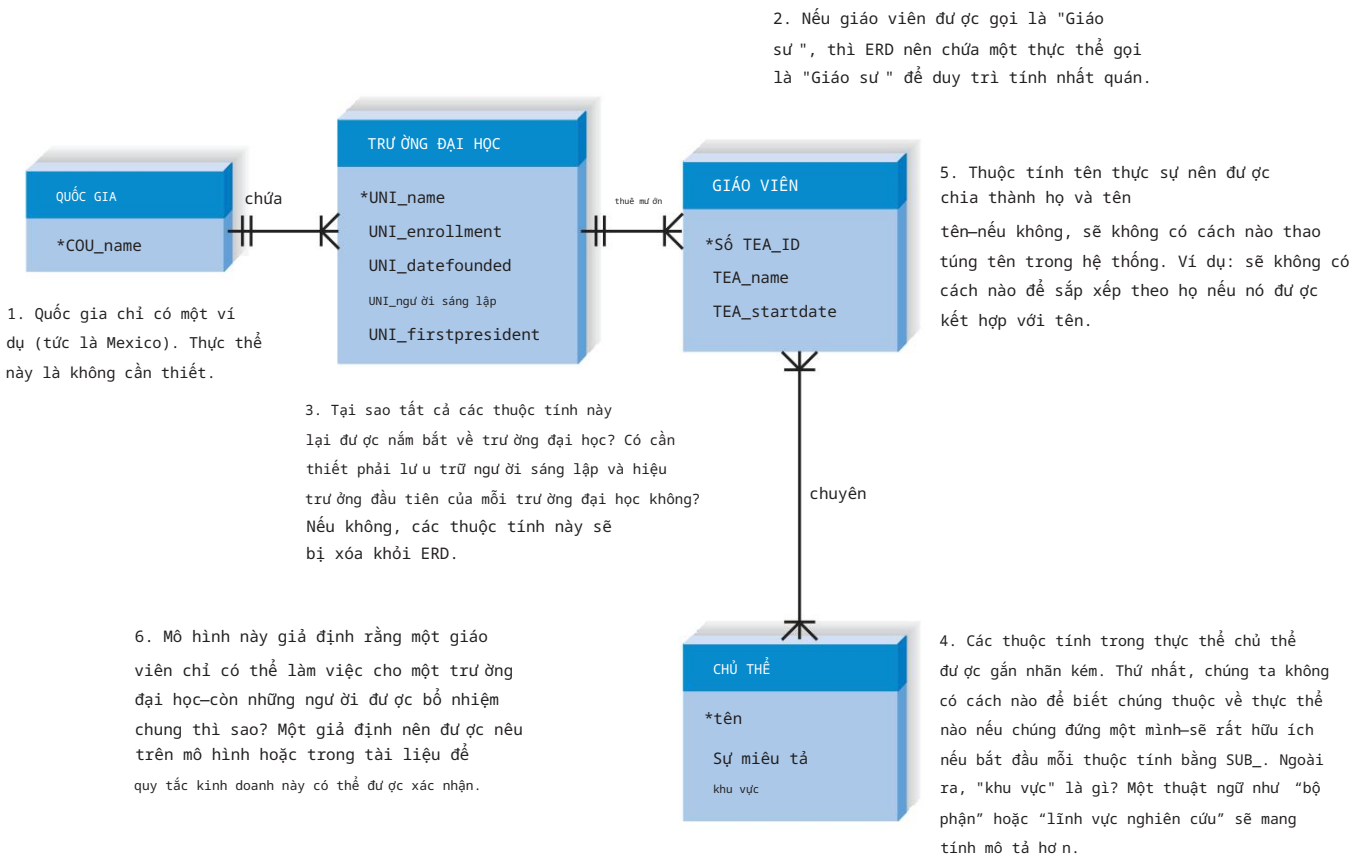
BẠN R

6-5 CÁC THỰC THỂ PHỤ THUỘC

XOAY

Xác định vị trí các thực thể phụ thuộc trên Hình 6-14. Xác định vị trí các mối quan hệ xác định. Làm thế nào mà bạn tìm thấy chúng? Bạn có thể tạo một quy tắc mô tả

sự liên kết giữa các thực thể phụ thuộc và xác định các mối quan hệ?



HÌNH 6-15

Tóm tắt nguyên tắc lập mô hình dữ liệu

Do đó, khi bạn lập mô hình dữ liệu, đừng hoảng sợ hoặc bị choáng ngợp bởi các chi tiết. Thay vào đó, hãy thêm các thành phần vào sơ đồ một cách từ từ, biết rằng chúng sẽ được thay đổi và sắp xếp lại nhiều lần. Đưa ra các giả định trên đường đi và sau đó xác nhận các giả định này với người dùng doanh nghiệp. Làm việc lặp đi lặp lại và liên tục thách thức mô hình dữ liệu với các quy tắc và ngoại lệ kinh doanh để xem liệu sơ đồ có giao tiếp với hệ thống kinh doanh một cách thích hợp hay không. Hình 6-15 tóm tắt các hướng dẫn được trình bày trong chương này để giúp bạn đánh giá mô hình dữ liệu của mình.

BẠN R

6-6 ĐỐI TƯ ỢNG GIAO TIẾP

XOAY

Giải quyết mối quan hệ M:N giữa đợt giảm giá và giai điệu có sẵn được minh họa trong Hình 6-14. Những loại thông tin bạn có thể nắm bắt về mối quan hệ này? ERD mới sẽ như thế nào? Thực thể giao lộ sẽ được coi là phụ thuộc hay độc lập?

Bạn có thể nghĩ ra các loại mối quan hệ M:N khác tồn tại trong thế giới thực không? Bạn sẽ giải quyết các mối quan hệ M:N này như thế nào nếu bạn đưa chúng vào ERD?

CÁC KHÁI NIỆM

6-B TRIỂN KHAI HỆ THỐNG EIM

TRONG HÀNH ĐỘNG

Một nhà cung cấp dịch vụ y tế bảo hiểm và sức khỏe trực tiếp lớn cần có hệ thống quản lý thông tin doanh nghiệp (EIM) để cho phép quản lý thông tin trên toàn doanh nghiệp và hỗ trợ việc sử dụng dữ liệu hiệu quả cho việc ra quyết định liên chức năng quan trọng. Ngoài ra, công ty cần giải quyết các vấn đề liên quan đến dư thừa dữ liệu, tính không nhất quán và chi phí không cần thiết. Công ty phải đối mặt với một số thách thức về thông tin: Dữ liệu của công ty nằm ở nhiều địa điểm, dữ liệu được phát triển để sử dụng cho bộ phận cụ thể và quyền truy cập của doanh nghiệp bị hạn chế. Ngoài ra, định nghĩa dữ liệu

được tạo bởi các bộ phận riêng lẻ và không được chuẩn hóa và dữ liệu được quản lý bởi nhiều bộ phận trong công ty.

Nguồn: http://www.deloitte.com/dtt/case_study/o,1005,sid%253D26562%2526cid%253D132760,00.html

CÂU HỎI:

1. Bạn sẽ đề xuất giải pháp nào cho công ty này?
2. Thảo luận về vai trò của mô hình hóa dữ liệu trong một án để giải quyết vấn đề này.

biến thứ ờng hóa

Khi bạn đã tạo ERD của mình, có một kỹ thuật gọi là chuẩn hóa có thể giúp các nhà phân tích xác thực các mô hình mà họ đã vẽ. Đó là một quá trình theo đó một loạt các quy tắc được áp dụng cho một mô hình dữ liệu logic hoặc một tệp để xác định mức độ hình thành của nó. Các quy tắc chuẩn hóa giúp nhà phân tích xác định các thực thể không được gửi chính xác trong mô hình dữ liệu logic hoặc các thực thể có thể bị tách ra khỏi tệp. Kết quả của quá trình chuẩn hóa là các thuộc tính dữ liệu được sắp xếp để tạo thành các quan hệ ổn định như ng linh hoạt cho mô hình dữ liệu. Trong Phụ lục 6A, chúng tôi mô tả ba quy tắc chuẩn hóa được áp dụng thứ ờng xuyên trong thực tế.

Cân bằng sơ đồ quan hệ thực thể với sơ đồ luồng dữ

liệu Tất cả các hoạt động phân tích của nhà phân tích

hệ thống đều có liên quan với nhau. Ví dụ, các kỹ thuật phân tích yêu cầu được sử dụng để xác định cách vẽ cả hai

BẠN R

CÔNG TY THUÊ TÀU 6-7

XOAY

Một công ty thuê tàu sở hữu những chiếc thuyền được sử dụng cho các chuyến đi thuê đến các đảo. Công ty đã tạo ra một hệ thống máy tính để theo dõi những chiếc thuyền mà họ sở hữu, bao gồm số ID, tên và sức chứa của mỗi chiếc thuyền. Công ty cũng theo dõi thông tin về các hòn đảo khác nhau, chẳng hạn như tên và dân số của chúng. Mỗi khi thuê thuyền, điều quan trọng là phải biết ngày chuyến đi sẽ diễn ra và số lượng người trong chuyến đi. Công ty cũng lưu giữ thông tin về từng thuyền trư ờng, chẳng hạn như số An sinh xã hội,

tên, ngày sinh và thông tin liên lạc của người thân. Thuyền chỉ đi đến một hòn đảo mỗi lần.

CÂU HỎI:

1. Tạo mô hình dữ liệu. Bao gồm các thực thể, thuộc tính, định danh và các mối quan hệ.
2. Những chủ thể nào phụ thuộc? Đó là độc lập?
3. [Tùy chọn] Sử dụng các bước chuẩn hóa để đặt mô hình dữ liệu của bạn ở dạng 3NF. Mô tả cách bạn biết rằng nó ở dạng 3NF.

mô hình quy trình và mô hình dữ liệu, và kho lưu trữ CASE được sử dụng để thu thập thông tin được lưu trữ và cập nhật trong toàn bộ giai đoạn phân tích. Bây giờ chúng ta sẽ xem các mô hình quy trình và mô hình dữ liệu có mối quan hệ với nhau như thế nào.

Mặc dù mô hình quy trình tập trung vào các quy trình của hệ thống kinh doanh, nhưng nó chứa hai thành phần dữ liệu—luồng dữ liệu (bao gồm các phần tử dữ liệu) và kho lưu trữ dữ liệu. Mục đích của những điều này là để minh họa dữ liệu nào được sử dụng và tạo ra bởi các quy trình và nơi lưu giữ những dữ liệu đó. Các thành phần này của DFD cần phải cân bằng với ERD. Nói cách khác, các thành phần dữ liệu DFD cần phải tương ứng với các kho lưu trữ dữ liệu của ERD (nghĩa là các thực thể) và các thành phần dữ liệu bao gồm các luồng dữ liệu (tức là các thuộc tính) được mô tả trên mô hình dữ liệu.

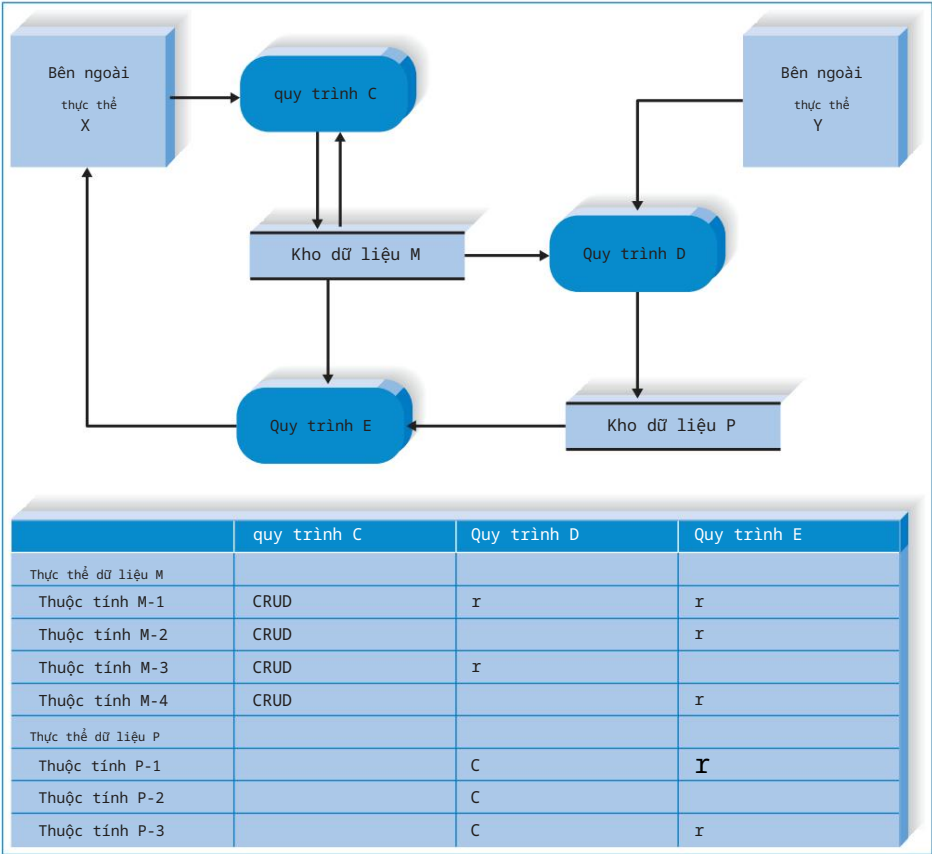
Nhiều công cụ CASE cung cấp tính năng xác định các vấn đề về sự cân bằng giữa DFD và ERD; tuy nhiên, bạn nên hiểu cách tự xác định vấn đề. Điều này liên quan đến việc kiểm tra mô hình dữ liệu bạn đã tạo và so sánh nó với các mô hình quy trình đã được tạo cho hệ thống.

Kiểm tra mô hình dữ liệu của bạn và xem liệu có bất kỳ thực thể nào bạn đã tạo không xuất hiện dưới dạng lưu trữ dữ liệu trên các mô hình quy trình của bạn hay không. Nếu có, bạn nên thêm chúng vào các mô hình quy trình của mình để phản ánh quyết định lưu trữ thông tin về thực thể đó trong hệ thống của bạn.

Tương tự, các bit thông tin có trong luồng dữ liệu (những bit này thường được xác định trong mục nhập CASE cho luồng dữ liệu) phải khớp với các thuộc tính được tìm thấy trong các thực thể trong mô hình dữ liệu. Ví dụ: nếu luồng dữ liệu thông tin khách hàng đi từ thực thể khách hàng đến quy trình giai đoạn mua hàng được xác định là có tên khách hàng, địa chỉ email và địa chỉ nhà, thì mỗi mẫu thông tin này sẽ được ghi lại dưới dạng thuộc tính trong thực thể khách hàng trên mô hình dữ liệu. Chúng ta phải xác minh rằng tất cả các mục dữ liệu có trong kho lưu trữ dữ liệu và luồng dữ liệu trong mô hình quy trình đã được đưa vào đâu đó dưới dạng thuộc tính thực thể trong mô hình dữ liệu. Chúng tôi muốn đảm bảo rằng mô hình dữ liệu kết hợp đầy đủ tất cả dữ liệu được xác định trong mô hình quy trình. Nếu không, thì mô hình dữ liệu không đầy đủ. Ngoài ra, tất cả các phần tử dữ liệu trong mô hình dữ liệu sẽ xuất hiện như một phần của kho lưu trữ dữ liệu và (các) luồng dữ liệu trong mô hình quy trình. Nếu một số yếu tố dữ liệu đã bị bỏ qua khỏi mô hình quy trình, thì chúng ta cần điều tra xem liệu các mục dữ liệu đó có thực sự cần thiết trong quá trình xử lý của hệ thống hay không. Nếu cần, chúng phải được thêm vào kho lưu trữ dữ liệu và luồng dữ liệu của mô hình quy trình; nếu không, chúng sẽ bị xóa khỏi mô hình dữ liệu dưới dạng các mục dữ liệu không liên quan.

Một công cụ hữu ích để mô tả rõ ràng mối quan hệ qua lại giữa các mô hình quy trình và dữ liệu là ma trận CRUD. Ma trận CRUD (tạo, đọc, cập nhật, xóa) là một bảng mô tả cách các quy trình của hệ thống sử dụng dữ liệu trong hệ thống. Sẽ rất hữu ích nếu phát triển ma trận CRUD trên cơ sở các mô hình dữ liệu và quy trình logic, sau đó sửa đổi nó sau trong giai đoạn thiết kế. Ma trận cũng cung cấp thông tin quan trọng cho các thông số kỹ thuật của chương trình, bởi vì nó cho thấy chính xác cách dữ liệu được tạo và sử dụng bởi các quy trình chính trong hệ thống.

Để tạo ma trận CRUD, một bảng được vẽ liệt kê tất cả các quy trình của hệ thống ở phía trên cùng và tất cả các thực thể dữ liệu (và thuộc tính thực thể) ở phía bên trái của bảng. Sau đó, từ thông tin được trình bày trong mô hình quy trình, nhà phân tích điền vào mỗi ô một C, R, U, D, (hoặc không có gì) để mô tả sự tương tác của quy trình với từng thực thể dữ liệu (và các thuộc tính của nó). Hình 6-16 cho thấy một phần của sơ đồ luồng dữ liệu và ma trận CRUD có thể được suy ra từ đó. Như bạn có thể thấy, nếu một tiến trình đọc thông tin từ kho lưu trữ dữ liệu, nhưng không cập nhật thông tin đó, thì sẽ chỉ có một luồng dữ liệu đi ra khỏi kho lưu trữ dữ liệu. Khi một quá trình



HÌNH 6-16
Mô hình quy trình một phần và ma trận CRUD

cập nhật kho lưu trữ dữ liệu theo một cách nào đó, sẽ có một luồng dữ liệu đi từ quy trình đến kho lưu trữ dữ liệu.

Suy nghĩ cẩn thận về nội dung của các luồng dữ liệu trong các mô hình quy trình, chúng ta có thể xác định những nơi mà các thuộc tính có thể đã bị bỏ qua khỏi các kho lưu trữ/thực thể dữ liệu. Ngoài ra, chúng tôi có thể xác minh rằng mọi thuộc tính được tạo, đọc, cập nhật và xóa ở đâu đó trong mô hình quy trình. Nếu nó không được đọc bởi một số quy trình, thì thuộc tính có thể không cần thiết. Nếu nó không được tạo hoặc cập nhật, thuộc tính có thể cần được thêm vào (các) luồng dữ liệu trong mô hình quy trình.

BẢN TÓM TẮT

Sơ đồ mối quan hệ thực thể cơ bản Cú

pháp Sơ đồ mối quan hệ thực thể (ERD) là kỹ thuật phổ biến nhất để vẽ một mô hình dữ liệu, một cách chính thức để biểu diễn dữ liệu được sử dụng và tạo bởi một hệ thống kinh doanh. Có ba yếu tố cơ bản trong ngôn ngữ mô hình hóa dữ liệu, mỗi yếu tố được biểu thị bằng một ký hiệu đồ họa khác nhau. Thực thể là khối xây dựng cơ bản cho một mô hình dữ liệu. Đó là một người, địa điểm hoặc sự vật mà dữ liệu được thu thập. Một thuộc tính là một số loại thông tin được nắm bắt về một thực thể.

Thuộc tính có thể xác định duy nhất một thể hiện của một thực thể được gọi là mã định danh. Thành phần mô hình dữ liệu thứ ba là mối quan hệ, truyền tải các mối liên kết giữa các thực thể. Mối quan hệ có cardinality (tỷ lệ

của các thể hiện cha mẹ đến các thể hiện con) và phụ ngữ thức (cha mẹ cần tồn tại nếu một đứa trẻ tồn tại). Thông tin về tất cả các thành phần được ghi lại bằng siêu dữ liệu trong từ điển dữ liệu.

[Tạo Sơ đồ Mối quan hệ Thực thể](#) Các bước

cơ bản trong việc xây dựng ERD là (1) xác định các thực thể, (2) thêm các thuộc tính thích hợp cho từng thực thể và (3) vẽ các mối quan hệ giữa các thực thể để chỉ ra cách chúng được liên kết với nhau. Có ba loại thực thể đặc biệt mà ERD chứa. Hầu hết các thực thể là độc lập, bởi vì một (hoặc nhiều) thuộc tính có thể được sử dụng để xác định duy nhất một thực thể. Các thực thể dựa vào các thuộc tính từ các thực thể khác để xác định một thể hiện là phụ thuộc. Một thực thể giao lộ được đặt giữa hai thực thể để nắm bắt thông tin về mối quan hệ của chúng. Nói chung, các mô hình dữ liệu dựa trên diễn giải; do đó, điều quan trọng là phải nêu rõ các giả định phản ánh các quy tắc kinh doanh.

[Xác thực Chuẩn hóa sơ đồ mối quan hệ thực](#)

[thể](#), quy trình theo đó một loạt các quy tắc được áp dụng cho mô hình dữ liệu logic để xác định mức độ hình thành của nó, được mô tả trong Phụ lục của Chương 6. Một mô hình dữ liệu logic ở dạng chuẩn đầu tiên (1NF) nếu nó không chứa các thuộc tính lặp lại, là các thuộc tính nắm bắt nhiều giá trị cho một phiên bản. Dạng chuẩn thứ hai (2NF) yêu cầu tất cả các thực thể đều ở dạng 1NF và chỉ chứa các thuộc tính có giá trị phụ thuộc vào toàn bộ định danh (nghĩa là không phụ thuộc một phần). Dạng chuẩn thứ ba (3NF) xảy ra khi một mô hình ở cả 1NF và 2NF và không có thuộc tính kết quả nào phụ thuộc vào thuộc tính không định danh (nghĩa là không phụ thuộc bắc cầu). Với mỗi lần vi phạm, các thực thể bổ sung sẽ được tạo để loại bỏ các thuộc tính lặp lại hoặc các thành phần phụ thuộc không phù hợp khỏi các thực thể hiện có. Cuối cùng, ERD nên được cân bằng với sơ đồ luồng dữ liệu (DFD)—đã được trình bày trong Chương 5—bằng cách đảm bảo rằng các thực thể và thuộc tính của mô hình dữ liệu tư ngữ ứng với các kho lưu trữ dữ liệu và luồng dữ liệu trên mô hình quy trình. Ma trận CRUD là một công cụ có giá trị để sử dụng khi cân bằng các mô hình dữ liệu và quy trình.

[ĐIỀU KHOẢN QUAN TRỌNG](#)

mối quan hệ 1:1	Thuộc tính dẫn xuất	Mối quan hệ không xác định
mối quan hệ 1:N	thực thể	bình thư ờng hóa
Giả thiết	Sơ đồ mối quan hệ thực thể (ERD)	Thực thể mẹ
Thuộc tính	Dạng chuẩn đầu tiên (1NF)	phụ thuộc một phần
Sự cân bằng	IDEF1X	Mô hình dữ liệu vật lý
Quy tắc kinh doanh	định danh	Mối quan hệ
hồng y	Xác định mối quan hệ	Thuộc tính lặp lại
Thực thể con	pháp nhân độc lập	Nhóm lặp lại
số nhận dạng được nối	Ví dụ	Dạng chuẩn thứ hai (2NF)
ma trận CRUD	thực thể giao lộ	lĩnh vực chủ đề
Từ điển dữ liệu	Mô hình dữ liệu logic	Dạng chuẩn thứ ba (3NF)
Mô hình dữ liệu	Mối quan hệ M:N	Bảng điểm
Sự phụ thuộc	metadata	phụ thuộc bắc cầu
pháp nhân phụ thuộc	phụ ngữ thức	