

NGÔ TRUNG VIỆT

PHÁT TRIỂN HỆ THỐNG THÔNG TIN

GÓC NHÌN
CỦA NGƯỜI
QUẢN LÝ



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

NGÔ TRUNG VIỆT

Biên soạn

PHÁT TRIỂN HỆ THỐNG THÔNG TIN

GÓC NHÌN CỦA NGƯỜI QUẢN LÝ



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI

60-6T7.3
———— 123-209-01
KHKT-01

LỜI NÓI ĐẦU

Cuốn sách này được soạn thảo với mục tiêu cung cấp một số kiến thức cơ bản và cần thiết cho người quản lý và các sinh viên sẽ tham gia vào công tác quản lý và phát triển hệ thống tin. Cuốn sách cũng có thể là một tài liệu học tập tốt cho sinh viên năm cuối khoa công nghệ thông tin các trường đại học. Trọng tâm của giáo trình nhằm vào việc làm sáng tỏ các vấn đề chính yếu của quản lý và cách sử dụng hữu hiệu CNTT trong thực tế hàng ngày của những người làm công tác quản lý. Do đó những vấn đề có tính chất chi tiết kỹ thuật đã được loại bỏ bớt và nội dung chính của cuốn sách này xoay quanh các phương pháp luận suy nghĩ, các quan niệm và cách quản lý mới có sử dụng kỹ thuật hiện đại.

Độc giả được làm quen với cách tư duy hệ thống, cách xem xét toàn diện vấn đề quản lý. Từ chỗ nắm được bản chất các vấn đề quản lý độc giả sẽ có một hình dung rõ rệt hơn về vai trò của thông tin trong quản lý và cách tổ chức thông tin trong các hệ thống tin. Độc giả sẽ làm quen với việc sử dụng các hệ thống tin để phục vụ cho công tác quản lý của mình, đặc biệt các vấn đề về quản lý và tổ chức các hệ thống tin, một trong những nhiệm vụ quản lý mới phát sinh. Cuốn sách này bao gồm các nội dung chính như sau:

Giới thiệu chung: Một tổng quan ngắn gọn về tình hình phát triển công nghệ thông tin trên thế giới và ứng dụng của nó vào mọi mặt cuộc sống, đặc biệt vào quản lý các tổ chức hiện nay.

Phần I: Nền tảng tổ chức của hệ thống tin: Đi sâu giới thiệu về các khía cạnh tổ chức, quản lý nói chung, vốn là những tri thức cơ bản mà người ứng dụng các kỹ thuật vào các hệ thống kinh tế xã hội cần phải nắm được. Việc sử dụng công nghệ

thông tin trong các tổ chức không phải đơn thuần chỉ là vấn đề kỹ thuật, nó còn bao hàm rất nhiều yếu tố về con người, cách làm việc, cách tổ chức. Các khía cạnh phi kỹ thuật đó cần phải được xem xét chu đáo cẩn thận thì việc đưa kỹ thuật vào các tổ chức kinh tế xã hội mới có hiệu quả thực sự.

Phần 2: Nền tảng kỹ thuật của hệ thống tin: Phần này đi sâu giới thiệu về các nguyên lý làm việc của máy tính cũng như việc kết mạng máy tính và việc phát triển các phần mềm trên máy tính, mạng máy tính để giúp cho các tổ chức quản lý thông tin của mình. Phần này mang nhiều tính chất kỹ thuật nhưng được giới thiệu một cách tổng quát và không đi vào chi tiết để những độc giả dù chưa có hiểu biết nhiều về máy tính và công nghệ thông tin vẫn có thể nắm được.

Phần 3: Xây dựng hệ thống thông tin: Phần này tập trung giới thiệu về các cách thức và phương pháp luận xây dựng hệ thống thông tin dựa trên máy tính. Các khía cạnh phát triển hệ thống cả về kỹ thuật lẫn phi kỹ thuật đều được khảo sát kỹ lưỡng để cung cấp cho người đọc cái hiểu đại cương về việc ứng dụng công nghệ thông tin trong các hoạt động quản lý là như thế nào.

Phụ lục A: Phương pháp phân tích thiết kế có cấu trúc: Phần này trình bày tóm tắt về một số khía cạnh kỹ thuật và các mô hình được dùng trong phương pháp luận có cấu trúc để xây dựng và phát triển hệ thống. Phần này có mục đích giới thiệu sâu hơn một chút về tổng quan đối với phương pháp luận này.

Phụ lục B: Ngôn ngữ mô hình hoá thống nhất (UML): Phần này trình bày một tổng quan ngắn gọn về ngôn ngữ mô hình hoá thống nhất, vốn dựa trên cách tiếp cận hướng sự vật để làm công cụ cho việc phân tích và thiết kế cho các hệ thống thực tế.

Với nội dung được lựa chọn như vậy, hi vọng rằng cuốn sách có thể đem tới cho bạn đọc nói chung những thông tin bổ ích và những hướng dẫn bước đầu về việc phát triển các hệ thống thông tin phục vụ quản lý. Do vậy cuốn sách này sẽ tập trung tốt hơn vào đối tượng phục vụ chính của nó là các cán bộ lãnh đạo các tổ chức, cơ quan, người muốn tìm hiểu các vấn đề của công nghệ thông tin để có thể ứng dụng nó một cách có hiệu quả. Các chi tiết kỹ thuật về phân tích và thiết kế hệ thống đã được chủ định không đưa vào quá tiết trong các phần chính của cuốn sách. Tuy nhiên các tài liệu được trình bày trong Phụ lục A và B lại là những chủ đề hơi chuyên sâu, được chủ định dành cho những người quản lý về công nghệ thông tin, có am hiểu về tin học và muốn tìm hiểu sâu thêm về các phương pháp luận và mô hình hoá dùng trong phân tích thiết kế hệ thống.

Cuốn sách này mặc dầu được chuẩn bị cẩn thận, nhưng vẫn còn nhiều thiếu sót do việc nhiều tư liệu còn chưa được sử dụng nhuần nhuyễn và phối hợp chặt chẽ. Hi vọng rằng nó sẽ nhận được nhiều ý kiến đóng góp của các độc giả xa gần để được hoàn chỉnh hơn.

Mục lục

Lời nói đầu.....	3
Giới thiệu chung	11
Cuộc cách mạng số	11
Môi trường kinh tế kĩ thuật hiện nay.....	13
Phát triển hệ thông tin và thương mại điện tử.....	16
 Phần 1 - Nền tảng tổ chức của hệ thông tin	21
1.1 Tổ chức	22
1.1.1 Định nghĩa	22
1.1.2 Các tính chất của tổ chức.....	23
1.1.3 Cấu trúc tổ chức: các dạng cơ bản.....	32
1.2 Hệ thống.....	41
1.2.1 Định nghĩa	41
1.2.2 Quan niệm điều khiển học các tổ chức	45
1.2.3 Kiểm soát qua "phản hồi"	50
1.2.4 Kiểm soát và điều chỉnh hệ thống	52
1.3 Quản lí.....	53
1.3.1 Vạch kế hoạch	55
1.3.2 Tổ chức	57
1.3.3 Bố trí cán bộ	68
1.3.4 Chỉ huy	59
1.3.5 Kiểm soát	60
1.4 Người quản lí	61

1.5 Ra quyết định.....	65
1.5.1 Tiến trình ra quyết định	65
1.5.2 Các giai đoạn ra quyết định	67
1.6 Hệ thống tin	72
1.6.1 Định nghĩa.....	81
1.6.2 Các loại Hệ thống tin.....	84
1.7. Công nghệ hệ thống tin.....	101
 Phần 2 - Nền tảng kĩ thuật của hệ thống tin.....	103
2.1. Máy tính và xử lí thông tin	104
2.1.1 Hệ thống máy tính là gì	104
2.1.2 Bộ nhớ chính.....	107
2.1.3 Đơn vị xử lí trung tâm	109
2.1.4 Bộ vi xử lí là gì? Chip là gì?	112
2.1.5 Bộ nhớ phụ	117
2.1.6 Thiết bị vào ra	121
2.1.7 Tiến hoá của phần cứng máy tính.....	124
2.1.8 Xu hướng công nghệ thông tin.....	130
2.2 Phần mềm hệ thống tin.....	132
2.2.1 Phần mềm là gì.....	132
2.2.2 Phần mềm hệ thống.....	134
2.2.3 Phần mềm ứng dụng.....	149
2.2.4 Các dụng cụ phần mềm mới và cách tiếp cận	157
2.3 Quản lí tài nguyên dữ liệu	167
2.3.1 Tổ chức dữ liệu trong môi trường tệp truyền thống.....	167
2.3.2 Môi trường cơ sở dữ liệu hiện đại	174
2.3.3 Thiết kế cơ sở dữ liệu	178
2.3.4 Các yêu cầu quản lí cho hệ thống cơ sở dữ liệu..	189
2.4 Viễn thông và Mạng máy tính	191
2.4.1 Cuộc cách mạng viễn thông	191
2.4.2 Các kiểu mạng truyền thông	199

2.4.3 Tổ chức sử dụng viễn thông	204
2.4.4 Tính nối và chuẩn	206
Phần 3 - Xây dựng hệ thống tin	217
3.1 Hệ thống xem như thay đổi tổ chức có kế hoạch ...	218
3.2 Tổng quan về phát triển hệ thống	224
3.3 Các phương pháp xây dựng hệ thống	238
3.4 Xây dựng hệ thống thành công: Chất lượng và thực hiện	260
3.4.1 Hệ thống tin thành công là gì?	260
3.4.2 Giải pháp cho vấn đề chất lượng phần mềm	269
3.4.3 Quản lí việc thực hiện hệ thống	288
Phụ lục A - Phân tích thiết kế có cấu trúc	301
A.1 Giai đoạn phân tích nghiệp vụ	303
A.2 Giai đoạn thiết kế hệ thống	306
A.3 Các khía cạnh phụ của phương pháp luận	318
Phụ lục B - Ngôn ngữ mô hình hoá thống nhất (UML)	343
B.1 Mô thức hướng sự vật	344
B.2 Mô hình hoá phân tích hướng sự vật	353

GIỚI THIỆU CHUNG

CUỘC CÁCH MẠNG SỐ

Cuộc cách mạng công nghiệp đã được thúc đẩy bởi động cơ hơi nước, phát minh năm 1792, và điện, được khai thác lần đầu tiên năm 1831. Việc khai thác năng lực của hơi nước mang lại việc giảm bớt lao động thủ công. Vì cần có một mạng lưới để chứa và truyền năng lượng, nên tiềm năng của điện phải đợi mãi tới 50 năm sau mới lần đầu tiên được sử dụng trước khi trạm năng lượng đầu tiên được xây dựng năm 1882. Phải mất thêm 50 năm nữa trước khi điện trở thành nguồn năng lượng cho 80 phần trăm các nhà máy và nhà ở. Chỉ mãi đến khi các nhà máy thay thế hệ thống năng lượng cũ bằng các mô tơ điện thì thay đổi cơ bản trong sản xuất mới xuất hiện. Cấu trúc nhà máy được sắp xếp hợp lý lại, và các tiến trình chủ chốt, như xử lý vật tư và dây chuyền chế tạo, đã được thực hiện hiệu quả hơn. Và chúng ta thấy các xã hội đã được biến đổi thành xã hội công nghiệp như hiện nay, khác hẳn hình thái xã hội nông nghiệp.

Cuộc cách mạng công nghệ thông tin (CNTT) đang xảy ra nhanh chóng hơn nhiều. Việc khai thác ánh sáng để truyền thông tức thời và khả năng dùng vi mạch để xử lý và cất giữ khối lượng thông tin khổng lồ đang làm cho việc biến đổi kinh tế hiện nay thành cực kỳ nhanh chóng.

Năm 1946, máy tính lập trình được đầu tiên trên thế giới ENIAC, cao hơn ba mét, rộng hơn 450 mét, giá hàng triệu đô la, và có thể thực hiện được 5000 lệnh một giây. Hai mươi lăm năm sau, năm 1971, Intel đóng gói một năng lực xử lý gấp 12 lần ENIAC vào trong một chip 12mm^2 với giá bán 200 \$. Máy tính cá nhân ngày nay (phần cứng) với bộ xử lý Pentium thực hiện quá 400 triệu lệnh trong một giây (MIPS). Với nhịp độ phát triển hiện tại, đến trước năm 2012, phần cứng sẽ có thể xử lý 100 000 triệu lệnh trong một giây.

Mãi đến năm 1980, các cuộc nói chuyện điện thoại chỉ truyền qua dây đồng, chuyển tải ít hơn một trang thông tin mỗi giây. Ngày nay, một sợi quang mỏng như sợi tóc con người có thể truyền trong một giây khối lượng thông tin tương đương 90 000 tập bách khoa toàn thư. Đến trước năm 2002, người ta dự kiến chùm hàng trăm vệ tinh địa tĩnh cách xa trái đất 36 000 km đem tới việc truyền thông bằng thông rộng cho doanh nghiệp, trường học và các cá nhân tại bất kì đâu trên hành tinh này.

Mạng máy tính toàn cầu dùng công nghệ chuyển mạch gói mới tổ hợp năng lực của những cải tiến đáng để ý này trong tính toán và truyền thông. Internet gắn với năng lực tính toán để bàn, trong nhà máy và trong văn phòng trên khắp thế giới qua kết cấu nền truyền thông tốc độ cao. Hơn 100 triệu người trên khắp thế giới, phần lớn là những người chưa bao giờ nghe thấy nói về Internet bốn năm trước đây, bây giờ dùng nó để nghiên cứu, gửi thư điện tử cho bạn bè, đặt hàng nhà cung cấp, và đi chợ điện tử.

Mặc dù các khuynh hướng rất ấn tượng này, cuộc cách mạng số cũng mới chỉ là bắt đầu. Sự tăng trưởng còn có thể được tăng tốc trong những năm sắp tới không chỉ trong chính bản thân lĩnh vực CNTT mà còn xuyên qua mọi lĩnh vực khác của nền kinh tế khi số người được nối với Internet tăng lên và khi việc sử dụng thương mại của nó được mở rộng.

Tất cả những điều này là chỉ báo cho thấy một sự biến đổi sâu xa nữa trong xã hội loại người đang xảy ra để tiến từ xã hội công nghiệp sang xã hội thông tin. Nhìn lại lịch sử vài chục năm qua của CNTT, có thể thấy rõ sự phát triển dần của xu hướng này. Tiêu biểu nhất là việc phát triển và mở rộng dần của các hệ thống tin, từ qui mô nội bộ trong tổ chức kinh tế, công nghiệp tiến tới qui mô toàn xã hội.

MÔI TRƯỜNG KINH TẾ KỸ THUẬT HIỆN NAY

Môi trường kinh doanh cạnh tranh của những năm 1990

Hai thay đổi toàn cầu rất mạnh mẽ đã làm biến đổi môi trường kinh doanh. Thay đổi đầu tiên là việc nổi lên và mạnh thêm của nền kinh tế toàn cầu. Thay đổi thứ hai là sự biến đổi của nền kinh tế và xã hội công nghiệp thành nền kinh tế dịch vụ dựa trên tri thức và thông tin. Những thay đổi này trong môi trường và khí hậu kinh doanh được tóm tắt trong bảng sau, đặt ra một số thách thức mới cho các tổ chức và sự quản lí của họ.

Môi trường kinh doanh thay đổi của những năm 1990

Toàn cầu hoá

- Quản lí và điều khiển trong thị trường toàn cầu
- Cạnh tranh trong thị trường thế giới
- Nhóm công tác toàn cầu
- Hệ thống giao hàng toàn cầu

Biến đổi của nền kinh tế công nghiệp

- Nền kinh tế dựa trên tri thức và thông tin
- Hiệu năng
- Sản phẩm và dịch vụ mới

- Quyền lãnh đạo
 - Cạnh tranh theo thời gian
 - Đời sống sản phẩm ngắn hơn
 - Môi trường biến động
 - Cơ sở tri thức người lao động có hạn chế
-

Sự nổi lên của nền kinh tế toàn cầu

Tỉ lệ phần trăm ngày càng tăng của nền kinh tế Mỹ - và các nền kinh tế công nghiệp tiên tiến khác ở Châu Âu và Châu Á - tùy thuộc vào nhập khẩu và xuất khẩu. Ngoại thương - cả xuất khẩu và nhập khẩu - đã vượt lên chút ít trên 25 phần trăm về hàng hoá và dịch vụ được tạo ra ở Mỹ, và thậm chí còn hơn trong các quốc gia như Nhật và Đức. Số phần trăm này sẽ tăng lên nữa trong tương lai. Sự thành công của các công ti trong những năm 1990 tùy thuộc vào khả năng của họ vận hành một cách toàn cầu.

Toàn cầu hoá nền kinh tế công nghiệp của thế giới làm nâng cao rất nhiều giá trị của thông tin cho công ti và đem đến nhiều cơ hội mới cho việc kinh doanh. Ngày nay, hệ thống tin cung cấp năng lực truyền thông và phân tích mà các tổ chức cần để tiến hành thương mại và quản lí kinh doanh trên qui mô toàn cầu.

Toàn cầu hoá và công nghệ thông tin cũng đem lại những nhân tố mới cho các công ti kinh doanh nội địa.

Biến đổi của nền kinh tế công nghiệp

Cuộc cách mạng tri thức và thông tin bắt đầu từ cuối thế kỉ hai mươi và đang dần dần tăng tốc. Trước năm 1976, số lượng công nhân cổ trắng trong các văn phòng đã vượt quá số lượng công nhân trong trang trại, công nhân dịch vụ và công nhân cổ xanh trong chế tạo. Ngày nay, phần lớn mọi người không còn làm việc trong trang trại hay nhà máy mà thay vì thế họ làm việc trong lĩnh vực bán

hàng, giáo dục, y tế, ngân hàng, bảo hiểm, luật pháp; họ cũng còn cung cấp các dịch vụ như sao chép, phần mềm máy tính hay giao hàng. Những công việc này chủ yếu là làm việc với, phân phối hay tạo ra thông tin và tri thức mới. Trong thực tế, công tác tri thức và thông tin bây giờ chiếm tới 75 phần trăm tổng sản phẩm quốc gia Mỹ, gần 70 phần trăm lực lượng lao động.

Trong một nền kinh tế dựa trên tri thức và thông tin, công nghệ thông tin và hệ thống chiếm một phần rất quan trọng. Chẳng hạn, công nghệ thông tin chiếm hơn 70 phần trăm vốn đầu tư trong ngành công nghiệp dịch vụ tựa như tài chính, bảo hiểm và bất động sản. Điều này có nghĩa là đối với nhiều nhà quản lí - có lẽ là phần lớn - các quyết định về công nghệ thông tin sẽ là những quyết định đầu tư thông dụng nhất.

Bởi vì hiệu suất của nhân viên sẽ phụ thuộc vào chất lượng của hệ thống phục vụ cho họ nên các quyết định quản lí về công nghệ thông tin có tầm quan trọng chủ chốt cho tương lai và sự tồn tại của một tổ chức.

Tất cả những điều trên cho thấy rằng chúng ta đang đứng trong một thời kì biến đổi sâu rộng trong xã hội và tổ chức để đi vào thời đại thông tin. Việc hiểu rõ và thích ứng được với những biến động lớn lao này luôn là vấn đề đặt ra trước mọi nhà quản lí các cấp, và thực tế đang dần trở thành một yêu cầu cấp thiết để mọi người tự thay đổi mình, để nâng mình lên một tầm cao tri thức mới.

Những tiến bộ kĩ thuật, một khi được xã hội chấp nhận, thì trở thành động lực thúc đẩy gây ra những biến đổi to lớn trong cách nghĩ, cách làm của con người và từ đó làm thay đổi cách thức hoạt động và sinh hoạt của xã hội. Để hiểu được những biến đổi lớn lao đó, chúng ta cần nhìn lại các vấn đề cơ bản của tổ chức, quản lí và từ đó thấy được vị trí của thông tin trong hệ thống điều khiển phức tạp của tổ chức, từ đó thấy được vai trò và vị trí của việc sử dụng CNTT trong quản lí. Chỉ trên cơ sở đó thì việc tham gia tích cực vào phát triển các hệ thống thông tin quản lí mới trở thành nhu cầu

cấp thiết và các tri thức về xây dựng hệ thống mới mới trở thành hữu hiệu.

PHÁT TRIỂN HỆ THÔNG TIN VÀ THƯƠNG MẠI ĐIỆN TỬ

Có thể nói, sự phổ cập sử dụng máy móc xử lý thông tin trong mọi mặt hoạt động xã hội là con đường đưa tới những biến đổi về xã hội thông tin. Điều này ban đầu được thực hiện từ việc làm biến đổi các đơn vị cơ sở trong nền kinh tế, xã hội, biến đổi của chính việc điều khiển và quản lý các công ti, cơ quan. Quá trình biến đổi này mở rộng ra toàn xã hội với việc mọi người đều dùng phổ cập công nghệ thông tin để xử lý công việc của mình và trao đổi với nhau, với các cơ quan khác.

Những thay đổi lớn lao trong công nghệ và kinh tế đang làm cho việc triển khai nhanh chóng kết cấu nền tính toán và truyền thông thành khả hiện trên toàn thế giới. Các nhà nước và đối tác thương mại của nó có được ích lợi chủ yếu từ khả năng xử lý thông tin nhanh hơn và chính xác hơn. Dần dần nhà nước sẽ bắt đầu một cách làm việc mới và phải theo kịp với sự bùng nổ thông tin đang xuất hiện trên toàn thế giới và là một người ủng hộ tích cực cho nó.

Về mặt lịch sử trong ngành công nghiệp thông tin, thời kì đầu, kiến trúc cho các hệ thống thông tin phần lớn do nhà cung cấp đưa ra. Khi người dùng chọn nhà cung cấp, thì họ cũng đã ngầm chọn một kiến trúc cho hệ thống tin của mình. Mỗi nhà cung cấp đều có một kiến trúc khác nhau và nhiều kiến trúc khác nhau lại không tương hợp nhau. Trong thập kỉ vừa qua cách tiếp cận đó đã được thay đổi với việc người sử dụng đứng ra xác định và thực hiện trên qui mô rộng về các chuẩn áp dụng cho tổ chức của mình. Một số chuẩn đã được các tổ chức chuẩn chính thức phát triển trong khi số khác đã được phát triển qua công nghệ được triển khai riêng tư nhưng được dùng phổ biến. Trong mọi trường hợp bây giờ người dùng có thể chọn lựa sản phẩm từ nhiều nhà cung cấp có khả năng và có thể được lợi từ sự cạnh tranh ngày càng tăng và một nền kinh

tế qui mô công nghiệp. Tuy nhiên, người dùng bây giờ là “điểm tích hợp” cho hệ thống tin và kết quả là trách nhiệm về tính toàn vẹn kiến trúc và sự phối hợp làm việc lẫn nhau của các phần tử hệ thống đang dịch chuyển về phía người dùng.

Khi thiết lập kiến trúc hệ thống qui mô công ti, các tổ chức cần xác định một chiến lược CNTT tổng thể. Chiến lược này phải bao gồm nhiều thành phần được dàn hàng với việc đạt tới mục tiêu của tổ chức một cách hiệu quả và hiệu suất nhất. Kết quả là ở chỗ tất cả các thành viên của bất kì cơ quan nào cũng đều truy nhập đúng lúc và dễ dàng vào đúng thông tin được trình bày trong một dạng có hiệu quả để cho họ có thể thực hiện công việc của mình theo cách tốt nhất. Bên cạnh đó tổ chức sẽ có khả năng quản lí các tài nguyên thông tin để đảm bảo sự tin cậy và tính sẵn có của toàn bộ hệ thống, cung cấp cơ chế chuyển giao thông tin.

Các hệ thống tin đã tiến triển trên các máy tính lớn thế hệ đầu tiên trong những năm 1970, sang các máy tính cá nhân PC thế hệ thứ hai trong những năm 1980 và tới thế hệ thứ ba của không gian điều khiển mạng mở. Môi trường có dùng mạng mở này đang biểu lộ dần khi chúng ta hướng tới thế kỉ thứ 21, tạo ra một hình ảnh hệ thống tin mới. Trong hình ảnh này, có cả các xí nghiệp ảo dựa trên các hệ thống tin xí nghiệp mới tạo khả năng cho việc xử lí kinh doanh trực tuyến (xử lí giao tác) và xử lí hỗ trợ quyết định (nhà kho dữ liệu) không có hạn chế gì về không gian và thời gian cả. Bên cạnh đó, thư điện tử và hội nghị điện tử đang bành trướng ra ngoài lĩnh vực kinh doanh tạo ra khả năng tính toán di động và sự hợp tác phi biên giới. Hơn nữa, việc truyền thông hai chiều giữa các hệ thống tin xí nghiệp, các trung tâm dịch vụ môi giới và tại nhà đã tạo ra khả năng làm việc ở nhà, giáo dục từ xa và chữa bệnh từ xa. Trong thế hệ thứ ba này, người ta trông đợi rằng sẽ có tiến bộ trong việc phát triển xã hội dựa trên việc nối mạng điện tử tập trung vào thị trường nơi làm việc và cộng đồng điện tử.

Kinh doanh điện tử đặt tất cả những trao đổi thông tin có trong các giao tác kinh doanh vào dạng điện tử trên mạng mở. Các tính năng cơ bản của thương mại điện tử là như sau:

- Tính mở, cho phép thương mại điện tử tiến hành các giao tác kinh doanh giữa các đối tác xác định, và cả những đối tác không xác định, cho phép người dùng chọn sản phẩm và kinh doanh.
- Truyền thông phi biên giới, để cho các giao tác kinh doanh có thể được thực hiện ở bất kì đâu bất kể với khu vực hay quốc gia.
- Tương tác, vốn cho phép truyền thông hai chiều được thực hiện giữa nhiều người tiêu thụ/doanh nghiệp do vậy tạo ra phong cách kinh doanh mới,
- Có khả năng truyền thông đa phương tiện do vậy tạo ra thông tin sản phẩm phong phú hơn có sẵn cho người mua hàng trong giao tác kinh doanh

Điều này làm thay đổi vai trò của của những định chế vốn là trung gian trong kinh doanh như phân phối và tài chính. Nhà nước được gắn trực tiếp với doanh nghiệp, doanh nghiệp gắn với doanh nghiệp, và doanh nghiệp gắn với người tiêu thụ, đem tới những thay đổi sau:

- Cách mạng trong thương mại: Việc kinh doanh có thể được tiến hành trên toàn thế giới, bất kể tới vị trí của công ti, với chi phí thấp.
- Cách mạng trong phân phối hàng hoá: luồng hàng hoá được gắn với thông tin, và các nội dung số được bàn giao trực tiếp, nhanh chóng và với chi phí thấp.
- Cách mạng trong tài chính: các tài khoản có thể được thanh toán nhanh chóng, chính xác và an toàn, với chi phí thấp.

- Cách mạng trong phân phối thông tin: người tiêu thụ có thể gửi thông tin từ phía mình, cho phép một phong cách kinh doanh mới có sự tham gia của người tiêu thụ.

Sự xuất hiện của các ứng dụng mới

Kinh doanh theo thương mại điện tử đã được khởi động mới gần đây và có thể được phân loại thành (a) giao tác giữa các doanh nghiệp và người tiêu thụ; (b) giao tác giữa các doanh nghiệp; (c) các dịch vụ công cộng. Điều chúng mang tính chất chung là ở chỗ chúng cung cấp và dùng chung thông tin, các dịch vụ trung gian và các hệ thống thực hành được xây dựng, cho việc chi trả điện tử tại ngân hàng và các công ti thẻ tín dụng.

(1) Giao tác giữa các doanh nghiệp và người tiêu thụ

Ứng dụng chủ yếu là mua bán trên Internet và ngân hàng Internet. Trong lĩnh vực này, hiện nhiều thực nghiệm đang được tiến hành, nhiều hệ thống thực tế đang được xây dựng cho hệ thống chi trả, như tiền điện tử dựa trên thẻ IC, chi trả theo thẻ tín dụng và giao tác ngân hàng trên Internet, vân vân. Phong trào này không chỉ tạo ra một cuộc cách mạng lớn trong kinh doanh tài chính, mà còn tạo ra một mô hình thương mại mới, bao gồm nhiều ngành công nghiệp, như đại lí và phân phối.

(2) Giao tác giữa các doanh nghiệp

Các ứng dụng đã được đề nghị cho giao tác giữa các doanh nghiệp, tương ứng với các tiến trình kinh doanh, bao gồm các dịch vụ cung cấp thông tin, các hệ thống thầu và đấu giá, các hệ thống xác nhận để xác nhận cho nội dung hợp đồng, và việc nhận và đặt đơn đặt hàng, các hợp đồng, dùng việc trao đổi dữ liệu điện tử. Trong mua bán của công ti, đã có tiến bộ từ hệ thống truyền thống của các đối tác giao tác đặc biệt sang thị trường mở. Khi mạng mở được sử dụng thì có nhiều yêu cầu về an toàn và khả năng đáp ứng. Bên cạnh đó, vì các giao thức giao dịch, vốn được giải quyết khác nhau đối với mỗi doanh nghiệp, cho nên việc chuẩn hoá là cần

thiết. Hiện tại, một hệ thống thực hành nhằm vào việc giải quyết các vấn đề này đang được phát triển.

(3) Các dịch vụ công cộng

Hiện nay, việc các cơ quan nhà nước dùng Internet bao gồm chủ yếu là việc tạo ra các trang nhà làm phương tiện công bố và phát tán thông tin, nhưng trong tương lai có thể trông đợi là sẽ phát triển thêm các phương tiện để dùng mạng mở cho nhà nước để cung cấp các dịch vụ công cộng cho dân chúng và để thu được vật liệu từ công nghiệp.

Vấn đề chúng ta phải đối diện

Từ những tiến bộ công nghệ trên đây, một thực tế mới hiện đang dần dần nổi lên và dù muốn hay không chúng ta cũng phải đương đầu với nó. Các tiến bộ công nghệ đã đang và sẽ được khuếch trương và phổ cập trên toàn cầu. Những gì thế giới ngày nay phát minh ra đều được phổ cập trên toàn thế giới, nói riêng chúng ta có mọi khả năng tiếp cận với những tiến bộ công nghệ mới nhất, nếu chúng ta có tiền bỏ ra mua máy móc mới. Và giá thành của các thiết bị xử lý thông tin này cũng ngày một hạ hơn, vào mức chúng ta có thể đảm đương nổi.

Nhưng ngược lại, để dùng được những công nghệ mới đó, chúng ta cần phải trải qua quá trình đào tạo và học hỏi thường xuyên, lâu dài. Một lực lượng lao động mới với những tri thức mới về dụng cụ lao động mới và cách làm việc mới cần được hình thành. Trong khi đó những nhân viên cũ cũng phải tích cực học hỏi các kỹ thuật mới để khẳng định vị trí của mình trong tiến trình ứng dụng công nghệ vào thực tế.

Do vậy việc am hiểu các vấn đề ứng dụng tiến bộ công nghệ vào công tác nghiệp vụ và quản lý là rất cần thiết và cấp bách. Các phần sau của cuốn sách này tập trung vào việc cung cấp cho độc giả những tri thức như vậy.

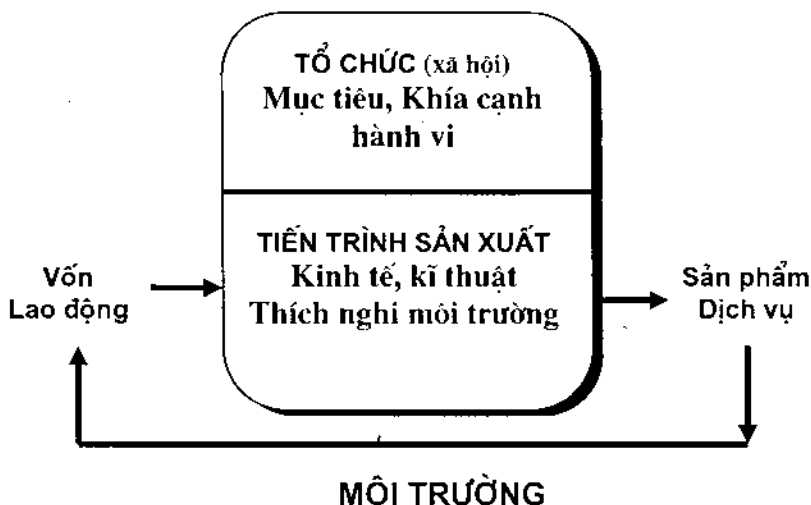
Phần **1**

NỀN TẢNG TỔ CHỨC CỦA HỆ THỐNG TIN

1.1 TỔ CHỨC

1.1.1 ĐỊNH NGHĨA

Tổ chức là một cấu trúc xã hội chính thức, ổn định nhận tài nguyên từ môi trường và xử lý chúng để tạo ra cái ra. Định nghĩa kỹ thuật này tập trung vào ba yếu tố của tổ chức. *Vốn* và *lao động* là hai nhân tố sản xuất chủ yếu do môi trường cung cấp. Tổ chức (cơ quan, công ti) biến những cái vào này thành sản phẩm và dịch vụ trong *chức năng sản xuất*. Sản phẩm và dịch vụ được *môi trường tiêu thụ* để cung cấp tiếp cái vào cho tổ chức (Xem hình 1.1) Một tổ chức là ổn định hơn một nhóm không chính thức dưới dạng độ dài lâu và thường lệ. Tổ chức là các thực thể hợp pháp chính thức, với các qui tắc và thủ tục nội bộ, cái phải được luật pháp hỗ trợ. Tổ chức cũng còn là một *cấu trúc xã hội* bởi vì chúng là tập hợp các phần tử xã hội.



Hình 1.1 Định nghĩa tổ chức về khía cạnh kỹ thuật và xã hội.

Quan điểm kĩ thuật về tổ chức cổ vũ chúng ta tập trung vào cách tổ hợp các cái vào thành cái ra khi sự thay đổi công nghệ được đưa vào tổ chức. Tổ chức được xem như để uốn nắn, với việc dễ dàng thay thế vốn và lao động. Nhưng định nghĩa hành vi thực tế hơn về tổ chức lại gợi ý rằng tổ chức bao gồm nhiều vấn đề hơn chỉ là bố trí máy móc và công nhân - đến mức một số hệ thống tin làm thay đổi sự cân bằng của tổ chức về quyền lợi, đặc quyền, nghĩa vụ, trách nhiệm và những cảm giác đã được thiết lập qua một thời kì dài.

Một định nghĩa có tính hành vi và hiện thực về tổ chức là một *cấu trúc xã hội*, đó là một tập hợp các quyền lợi, đặc quyền, nghĩa vụ, và trách nhiệm được giữ cân bằng tĩnh tế qua từng thời kì giữa xung khắc và giải quyết xung khắc. Tổ chức có **mục tiêu** của riêng nó.

Tổ chức tồn tại và phát triển trong môi trường.

Sự thay đổi kĩ thuật đòi hỏi có những thay đổi trong chính bản thân thông tin và cách kiểm soát thông tin; người có quyền truy nhập và cập nhật vào thông tin; và người ra quyết định về ai làm, khi nào làm và làm thế nào.

Các định nghĩa về kĩ thuật và hành vi của tổ chức là không mâu thuẫn nhau. Thực tế, chúng bổ sung lẫn cho nhau: định nghĩa kĩ thuật cho ta biết cách thức hàng nghìn công ti trong thị trường cạnh tranh tổ hợp vốn, lao động và công nghệ thông tin, trong khi mô hình hành vi cho chúng ta nhìn vào bên trong từng công ti để xem cách thức công nghệ ảnh hưởng tới sự làm việc bên trong của tổ chức.

1.1.2 CÁC TÍNH CHẤT CỦA TỔ CHỨC

Như đã nêu ở trên, tổ chức phục vụ cho các chức năng sau:

- Làm cho các thành viên tổ chức hiểu được các hoạt động ứng với việc phân chia lao động xác định ra chuyên môn hoá, chuẩn hoá và phân phòng ban cho các nhiệm vụ và chức năng
- Làm cho các thành viên tổ chức phối hợp hoạt động của họ qua cơ chế tích hợp như giám sát theo cấp bậc, qui tắc và thủ tục chính thức, huấn luyện và xã hội hoá.
- Xác định biên giới của tổ chức và giao diện của nó với môi trường hay các tổ chức khác mà thành viên của nó phải tương tác.

Trên một số khía cạnh thì tất cả các tổ chức hiện đại đều giống nhau bởi vì chúng có chung những đặc trưng đã được nêu trong Bảng 1.1 sau. Một nhà xã hội học người Đức, Max Weber, là người đầu tiên mô tả những đặc trưng "lí tưởng" của tổ chức vào năm 1911. Ông gọi các tổ chức là **quan liêu** nếu có một số "đặc trưng" nào đó. Theo Weber tất cả các cơ cấu quan liêu đều có

Bảng 1.1 Đặc trưng cấu trúc của mọi tổ chức hiện đại

Phân chia lao động rõ rệt
 Phân cấp bậc
 Qui tắc và thủ tục tường minh
 Đánh giá công bằng
 Phẩm chất kĩ thuật theo vị trí
 Hiệu quả tổ chức tối đa

- Có sự phân chia rõ rệt giữa *lao động* và *chuyên môn*
- Sắp xếp các chuyên gia theo *cấp bậc* quyền hành.
- Giới hạn quyền hành và hành động theo các *qui tắc* hay *thủ tục* trừu tượng (các thủ tục tác nghiệp chuẩn hay SOP)
- Tạo ra một hệ thống *ra quyết định công bằng* và *vận năng*
- Thuê và đề bạt nhân viên trên cơ sở *phẩm chất* và *chuyên môn kĩ thuật* (không liên quan tới cá nhân)

- **Hướng theo nguyên tắc hiệu quả:** tối đa cái ra bằng cách dùng cái vào có giới hạn.

Cấu trúc tổ chức có nhiều vấn đề hơn chỉ là các hộp và đường vẽ trong sơ đồ. Trong khi suy nghĩ về cấu trúc, các vấn đề phải được đề cập tới bao gồm việc phân chia lao động và quyền quyết định, cơ chế điều phối, biên giới tổ chức và các cấu trúc cùng tiến trình không chính thức.

Phân chia lao động. Điều này bao gồm việc quyết định về điều sau đây:

- **Sự khác biệt:** Qui mô của việc chuyên biệt hoá theo chiều ngang và chiều đứng của công việc; chẳng hạn, chiều ngang công việc theo số các hoạt động khác biệt cần phải được thực hiện, chiều sâu công việc theo qui mô tổ hợp các hoạt động quan niệm, thực hiện và quản trị.
- **Phân ban hoá:** Việc gộp nhóm các hoạt động theo hàng hoá và dịch vụ, việc cung cấp chúng cho ai hay các chức năng, kĩ năng và tri thức được dùng để tạo ra chúng.

Phân chia quyền quyết định. Ai ra quyết định nào? Quyền quyết định bao gồm chủ quyền để khởi xướng, chấp thuận, thực hiện và kiểm soát nhiều kiểu quyết định chiến lược và chiến thuật khác nhau. Những quyền này được phân bố theo cả các chiều đứng và ngang. Chiều đứng trải rộng qua các mức độ khác nhau của tổ chức, dẫn tới vấn đề xác định mức độ thẩm quyền tập trung hay phi tập trung thích hợp. Chiều ngang liên quan tới các hoạt động tại cùng một mức của tổ chức. Chẳng hạn các phòng ban tiếp thị và kĩ nghệ có quyền quyết định về đặc tả sản phẩm mới không?

Một cách lí tưởng, quyền quyết định được trao cho những người có sự truy nhập lớn vào thông tin có liên quan. Chẳng hạn người quản lí chế tạo thường có thông tin rõ nhất liên quan tới khả năng sản xuất, trong khi người quản lí bán hàng lại biết nhiều nhất về khách hàng. Tuy nhiên, không phải việc cho phép người quản lí

chế tạo ra quyết định công việc cho nhà máy là tối ưu, bởi vì họ có thể không có thông tin về các nhân tố có liên quan. Thậm chí nếu từng người có mọi thông tin cần thiết, thì họ vẫn có thể ra quyết định không tối ưu nếu mối quan tâm của họ là xa lạ với tổ chức xem như một toàn thể. Chẳng hạn người quản lý chế tạo có thể bị nghiêng về sản xuất nếu người đó được quyền ra quyết định; người quản lý bán hàng có thể có khuynh hướng đưa ra những giảm giá cao để đáp ứng hạn ngạch đơn vị bán.

Cơ chế điều phối. Bởi vì lao động và quyền quyết định bị phân chia giữa nhiều thành viên của tổ chức, nên sự khác biệt phải được làm cân bằng qua việc tích hợp. Việc tích hợp có thể được đạt tới qua cơ chế điều phối vốn bao gồm các phương tiện chính thức như giám sát trực tiếp, việc dùng các qui tắc và thủ tục chính thức, các tiến trình và cái ra công việc đã chuẩn hoá, chuẩn hoá kỹ năng, và cũng qua các cơ chế không thể thức hay không chính thức như tổ đặc nhiệm. Việc chọn lựa cơ chế điều phối phụ thuộc một phần vào mức độ mà với nó vấn đề là *thường lệ* chứ không *ngoại lệ*:

- Việc điều phối *thường lệ* bao gồm các vấn đề như tích hợp ngang qua các vị trí trong một dây chuyền lắp ráp hay xử lý hàng tháng về séc chi trả. Các thủ tục vận hành chuẩn và việc giám sát trực tiếp hoàn thành sự điều phối này về cơ bản.
- Việc điều phối *ngoại lệ* bao gồm giải quyết các vấn đề bất thường hay không thường xuyên hay việc phân bổ các tài nguyên có giới hạn. Kế hoạch và ngân sách, thương lượng, tổ chức theo uỷ ban, và lực lượng đặc nhiệm thường được dùng cho dạng điều phối này.

Mặc dù quan trọng nhưng việc điều phối là tốn kém. Chẳng hạn, các cuộc họp điều phối, trong khi có ích, thì có thể tốn thời gian.

Biên giới tổ chức. Việc cấu trúc một tổ chức cũng bao hàm việc quyết định phải làm gì bên trong và phải làm gì bên ngoài biên

giới của tổ chức, điều này đến lượt nó lại dẫn tới quyết định làm hay mua và chọn đối tác chiến lược. Kho phải lưu giữ bao nhiêu vật tư và linh kiện để đảm bảo việc cung ứng thích hợp, khi so với việc quản lý mối quan hệ với các nhà cung cấp? Kho thành phẩm cuối cùng phải giữ ở mức nào để làm bộ đệm cho tổ chức khi nhu cầu khách hàng trời nổi?

Cấu trúc không chính thức. Vai trò và mối quan hệ chính thức không phải là cơ sở duy nhất của mối tương tác xã hội trong tổ chức. Mặc dầu cấp bậc và chức danh được tính tới, chúng không đưa ra một bức tranh đầy đủ về ai có quyền trong tổ chức. Thảm quyền cũng có thể phát sinh từ địa vị xã hội, tri thức chuyên gia và việc ảnh hưởng tới người khác. Quan trọng chẳng kém gì các vai trò chính thức là các mối quan hệ không chính thức, thường dựa trên sự gần gũi, tình bạn, chung quyền lợi và các nhân tố khác. Điều cũng quan trọng là cần thừa nhận rằng mọi tổ chức đều mang tính chính trị ở mức độ nào đó, vì bao giờ cũng có liên minh với những quyền lợi cạnh tranh. Đường lối chính trị có thể được vẽ ra ngang qua các đường biên phòng ban, chức năng, hay chúng có thể cắt qua biên giới chính thức. Những nhà quản lý có hiệu quả đều thừa nhận mạng lưới không chính thức và đều tài nghệ khi ước định khung cảnh chính trị.

Các nhà quản lý chung có thể thao tác và thay đổi một cách có chủ ý việc phân chia lao động và quyền quyết định, cơ chế điều phối và biên giới tổ chức. Cấu trúc không chính thức ít tuân theo ảnh hưởng quản lý, nhưng lại không kém phần quan trọng.

Chính trị trong tổ chức

Mọi người trong tổ chức giữ những cương vị khác nhau với các chuyên môn, mối quan tâm và viễn cảnh khác nhau. Kết quả là họ tự nhiên có những quan điểm khác nhau về việc nên phân bổ tài nguyên, thưởng và phạt. Những sự khác biệt này trở thành quan trọng cho các thành viên trong tổ chức, cả với người quản lý lẫn

nhân viên, và kết quả là làm phát sinh đấu tranh chính trị, cạnh tranh và xung khắc bên trong mọi tổ chức. Người ta dùng chính trị để thu được mọi thứ đáng có trong công việc: lương, chức, điều kiện công việc, sự kính trọng, uy thế và chung cuộc cả nghề nghiệp. Bởi nguyên tắc này người ta tham dự trò chơi rất nghiêm chỉnh. Sự chống đối chính trị là một trong những khó khăn nhất của việc đưa thay đổi vào tổ chức - đặc biệt với sự phát triển của hệ thống tin mới. Thực tế mọi hệ thống tin đem đến những thay đổi có ý nghĩa về mục tiêu, thủ tục, hiệu năng, nhân sự thì đều tấn công vào mặt chính trị và sẽ khơi ra sự đối lập chính trị trầm trọng.

Văn hoá trong tổ chức

Mọi tổ chức đều có những giả thiết không thể bác bỏ, không thể nghi vấn để xác định ra mục tiêu và sản phẩm của chúng. Văn hoá trong tổ chức là tập các giả thiết cơ bản này, các giả thiết về việc tổ chức nên sản xuất ra sản phẩm nào; nên sản xuất chúng như thế nào, ở đâu và cho ai. Nói chung những giả thuyết văn hoá này được công nhận hoàn toàn và hiếm khi được công bố hay nói ra cho mọi người. Chúng đơn thuần là những giả thuyết mà chỉ vài người, nếu có, mới hỏi tới.

Mọi thứ khác - công nghệ, giá trị, chuẩn mực, công báo văn văn - đều tuân theo giả thuyết này. Văn hoá trong tổ chức là một lực thống nhất mạnh, kìm hãm lại những xung khắc chính trị và thúc đẩy hiểu biết chung, thoả thuận về thủ tục và thực hành chung. Nếu chúng ta cùng có chung các giả thuyết văn hoá cơ bản thì thoả thuận trên các vấn đề khác sẽ thuận tiện hơn.

Đồng thời cấu trúc tổ chức cũng là một sự kìm hãm mạnh với thay đổi, đặc biệt những thay đổi về kỹ thuật. Phần lớn các tổ chức sẽ làm gần như mọi điều để tránh bị thay đổi các giả thuyết cơ bản. Ban đầu các công nghệ mới gần như bao giờ cũng được dùng để hỗ trợ cho văn hoá hiện có, và bất kỳ sự thay đổi công nghệ nào đe dọa

tới các giả thiết văn hoá đã được thừa nhận chung thì đều gặp phải sự chống đối rất lớn.

Cấu trúc tổ chức có thể thay đổi dưới dạng của năm chiều hướng sau:

Phân chia lao động

Mức độ khác biệt

Gộp nhóm các hoạt động

Phân chia quyền quyết định

Thẩm quyền chiều đứng

Thẩm quyền chiều ngang

Cơ chế điều phối

Cơ chế chính thức

Cơ chế không chính thức

Thường lệ so với ngoại lệ

Biên giới tổ chức

Tích hợp chiều ngang và chiều đứng

Mối quan hệ liên tổ chức

Cấu trúc không chính thức

Mạng xã hội và tiến trình ảnh hưởng

Liên minh chính trị

Tại sao các tổ chức lại giống nhau: những nét chung

Thủ tục tác nghiệp chuẩn

Tổ chức đã tồn tại qua thời gian sẽ trở nên rất hiệu quả, tạo ra một số giới hạn các sản phẩm và dịch vụ theo các qui trình chuẩn. Các qui trình chuẩn này được soạn thành điều luật với những qui định, thủ tục và thực hành chuẩn còn được gọi là các thủ tục tác nghiệp chuẩn (SOP), được thiết lập để bao quát thực tế mọi tình

huống có thể dự kiến. Một số các qui tắc và thủ tục này được viết thành các thủ tục chính thức. Phần lớn các qui tắc là qui định cần phải theo trong một số tình huống.

Tại sao các tổ chức lại khác nhau: các nét đặc thù

Mặc dầu mọi tổ chức đều có những đặc trưng chung nhưng không có hai tổ chức nào là trùng nhau. Các tổ chức đều có cấu trúc, mục tiêu, thể chế, phong cách lãnh đạo, nhiệm vụ và môi trường bao quanh khác nhau.

Các kiểu tổ chức khác nhau

Sự khác biệt giữa các tổ chức thể hiện một phần quan trọng ở cấu trúc hay hình thái của chúng. Sự khác biệt giữa các cấu trúc tổ chức được đặc trưng theo nhiều cách.

- Cấu trúc xí nghiệp
- Bộ máy quan liêu
- Bộ máy quan liêu chuyên nghiệp
- Bộ máy quan liêu phân nhánh
- Tổ chức theo lực lượng đặc nhiệm

Môi trường

Các tổ chức đều có những môi trường khác nhau, mỗi môi trường thực thi ảnh hưởng quyền lực lên cấu trúc của tổ chức. Nói chung, các tổ chức thành công trong những môi trường phát triển nhanh đều có tổ chức theo kiểu lực lượng đặc nhiệm - chúng ít cấp bậc hơn, với nhiều mức uỷ quyền cho thuộc cấp hơn. Tổ chức trong môi trường ổn định có khuynh hướng phát triển thành bộ máy quan liêu.

Phần lớn mọi người đều không hiểu thực tế các tổ chức chính thức có đời sống ngắn ngủi và mỏng manh làm sao. Ta cứ xét việc có ít hơn 10 phần trăm trong số 500 công ti có tên trong Fortune từ năm 1918 còn lại sau 50 năm; ít hơn 4 phần trăm của tất cả các tổ chức chính quyền liên bang từ khi được thành lập là còn tồn tại; 50 phần trăm của tất cả các tổ chức tư nhân mới chấm dứt kinh doanh trong vòng năm năm.

Lí do chính cho sự thất bại tổ chức là ở việc không có khả năng thích nghi với môi trường thay đổi nhanh chóng và thiếu tài nguyên - đặc biệt trong các công ti trẻ - để chống đỡ những thời kì trục trặc ngắn. Những công nghệ thay đổi nhanh, sản phẩm mới và thay đổi thị hiếu và giá trị trong dân chúng đặt sự kìm hãm lên văn hoá, chính trị, thủ tục tác nghiệp và con người của bất kì tổ chức nào. Những công nghệ thay đổi nhanh như CNTT đặt ra sự đe dọa đặc biệt cho tổ chức. Hiện nay những thay đổi công nghệ xuất hiện quá nhanh chóng đã tạo nên "sự không liên tục công nghệ", một sự hụt hẫng trong thực hành công nghiệp tạo ra hoặc việc nâng cao hoặc việc phá huỷ các công ti trong công nghiệp. Khi sự không liên tục công nghệ xảy ra, phần lớn các tổ chức không thích nghi nổi và nhiều tổ chức biến mất, giải phóng nguồn tài nguyên cho những tổ chức mới, trẻ hơn.

Sự khác biệt giữa các tổ chức

Tổ chức có cấu trúc và chính thức khác nhau bởi nhiều lí do. Chúng khác nhau ở mục tiêu cuối cùng và kiểu năng lực để đạt tới các mục tiêu đó. Một số tổ chức có mục tiêu bị bó buộc, số khác có mục tiêu thiết thực.

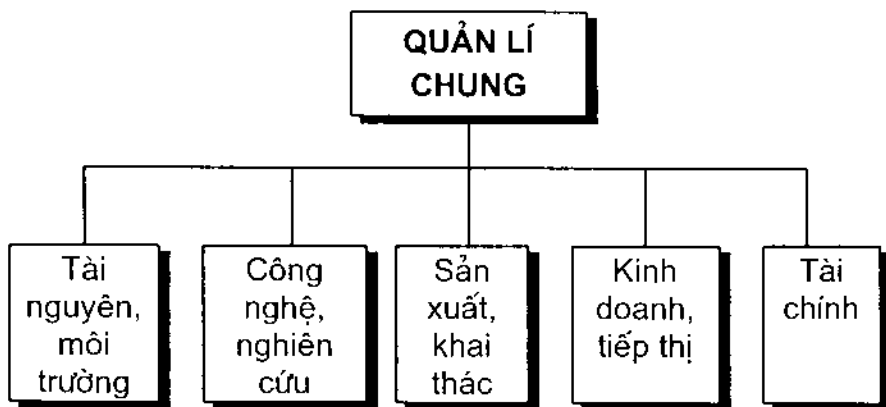
Rõ ràng bản chất của lãnh đạo làm cho tổ chức nọ khác biệt rất lớn với tổ chức kia. Một số phong cách lãnh đạo chính là dân chủ, phân quyền, bỏ mặc, kĩ nghệ hay quan liêu.

Các tổ chức còn khác biệt theo nhiệm vụ họ thực hiện và công nghệ họ dùng.

1.1.3 CẤU TRÚC TỔ CHỨC: CÁC DẠNG CƠ BẢN

Về mặt lí thuyết, có thể có một số lớn các cấu trúc tổ chức. Trong thực tế, ba dạng cơ bản - chức năng, phân khu vực và ma trận - mô tả cho phần lớn các tổ chức đương đại.

Dạng chức năng. Khi tổ chức lớn lên vượt ra ngoài những vấn đề có thể được một nhóm người và một ông chủ giải quyết, thì cấp quản lí thường chọn lựa việc hình thành cấu trúc chức năng (Hình 1.2). Điều này tạo ra một sự phân chia lao động ban đầu dưới dạng các chức năng chính, như kĩ nghệ, sản xuất, bán hàng và tài chính. Khi tổ chức lớn lên, các chức năng có thể được phân chia nhỏ thêm và những chức năng mới có thể được thêm vào.



Hình 1.2 Cấu trúc chức năng.

Trong cấu trúc chức năng, các hoạt động chung được gộp nhóm lại với nhau. Chẳng hạn các kĩ sư được bố trí trong phòng kĩ nghệ do một phó chủ tịch công nghệ đứng đầu. Hoạt động của họ được điều phối theo chiều đứng theo cách giám sát cấp bậc. Các nghề nghiệp thông thường được xác định dựa trên cơ sở tri thức chuyên gia chức năng, và nhân viên bên trong một chức năng chấp nhận các giá trị, mục tiêu và khuynh hướng tương tự. Sự tương tự

này động viên sự hợp tác, tính hiệu quả và chất lượng bên trong chức năng đó, nhưng làm cho việc điều phối và hợp tác với các phòng ban khác khó khăn hơn.

Vì hiệu năng của tổ chức phụ thuộc vào tất cả các chức năng làm việc với nhau theo cách thức có điều phối, nên cấu trúc chức năng đòi hỏi sự trao đổi thông tin mạnh mẽ giữa các chức năng. Các thủ tục chéo chức năng phải được tạo ra để xử lý đơn hàng, quản lý kho và tiền mặt, điều phối việc phát triển sản phẩm và thay đổi thiết kế, và thực hiện các tiến trình kinh doanh. Trách nhiệm xử lý thông tin chéo chức năng chủ yếu rơi vào người quản lý chung, người cũng làm hoà giải mọi xung khắc phát sinh giữa các chức năng. Người quản lý chung có thể bổ sung một nhân viên trung tâm và tạo ra các cơ chế chính thức cho việc điều phối - như hệ thống lập kế hoạch và lên ngân sách - để quản lý trách nhiệm tương tác này. Với việc dùng các cơ chế như vậy, một số tổ chức đã có khả năng tăng qui mô lên nhiều trong khi vẫn duy trì cấu trúc chức năng cơ bản.

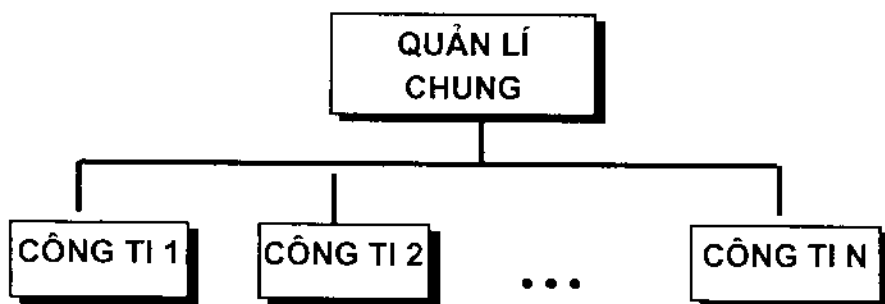
Dạng chức năng có tác dụng tốt khi các vấn đề cạnh tranh chủ yếu của tổ chức nhấn mạnh vào tri thức chuyên gia, tính hiệu quả và chất lượng, và khi môi trường của nó tương đối ổn định. Điều này là vì dạng chức năng thúc đẩy cơ cấu tổ chức theo qui mô. Các nhà máy chế tạo hợp nhất có thể làm cho tổ chức thu được máy móc đắt tiền và hiệu quả và giảm bớt sự trùng lặp và lãng phí. Cấu trúc này cũng thúc đẩy việc phát triển kỹ năng chức năng của các nhân viên bằng việc cung cấp một thang nghề nghiệp xác định rõ.

Điểm yếu của cấu trúc chức năng là ở tính không có khả năng đáp ứng với môi trường khác biệt hay môi trường mà trong đó tổ chức phải đáp ứng khác đi trong mỗi loại sản phẩm hay sự phân chia khách hàng. Cấu trúc này cũng yếu khi tổ chức phải đáp ứng nhanh chóng với những thay đổi của môi trường vốn cần tới sự điều phối giữa các chức năng. Nơi thay đổi (và do đó tính không chắc chắn) là cao, thì cơ chế điều phối chéo chức năng có thể trở thành

bị quá tải. Các cấp quyết định chồng chất và cấp quản lí thượng đỉnh không thể nào đáp ứng lại một cách nhanh chóng. Tổ chức theo chức năng cũng hạn chế cái nhìn của từng nhân viên về những mục tiêu tổng thể và sự vận hành của tổ chức, dẫn tới việc tối ưu về chi phí cho mục tiêu của tổ chức.

Dạng phân khu vực. Cấu trúc phân khu vực, được minh hoạ trong hình 1.3, gộp nhóm nhiều chức năng như chế tạo, nghiên cứu và phát triển, và tiếp thị vào từng khu vực. Trong khi cấu trúc chức năng được tổ chức theo *cái vào*, thì cấu trúc phân khu vực được tổ chức theo *cái ra*. Mỗi khu vực có thể chịu trách nhiệm cho một tập hợp các sản phẩm khác nhau, thị trường hay phân chia theo vùng địa lí, hay khách hàng. Vậy người ta có thể có cấu trúc phân khu vực dựa theo sản phẩm, vùng, phân chia thị trường hay khách hàng.

Bên trong một khu vực, việc điều phối chéo chức năng còn mạnh hơn việc điều phối cho dạng chức năng. Các nhân viên đồng nhất với khu vực của mình hơn là với chuyên môn chức năng. Những quyết định thúc đẩy người quản lí về cơ bản dựa rất nhiều vào các kĩ năng tích hợp - như khả năng trao đổi và động viên các cá nhân với nền đào tạo khác nhau, và hiểu biết mối liên hệ giữa các chức năng - hơn là trên cơ sở tri thức chuyên gia chức năng.



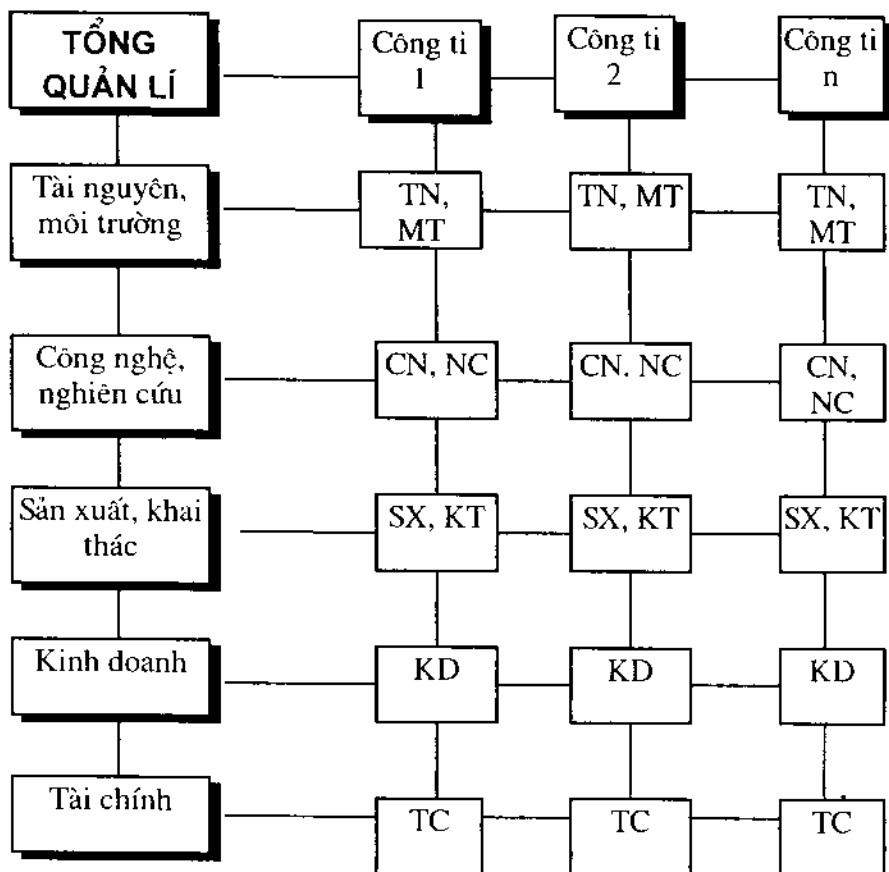
Hình 1.3 Dạng khu vực.

Mỗi khu vực có thể được giữ trách nhiệm về hiệu năng của mình bởi việc xác định nó như một trung tâm lợi tức, trung tâm chi phí, trung tâm lợi nhuận, hay trung tâm đầu tư. Vấn đề mấu chốt trong cấu trúc này là mức độ tự trị được ban cho các khu vực trong việc ra quyết định vốn là chiến lược và bao hàm những cam kết tài nguyên có ý nghĩa. Việc điều phối ngang qua các khu vực được cai quản bởi một nhóm các nhà quản lí tại tổng hành dinh công ti, người chịu trách nhiệm phân bổ tài nguyên giữa các khu vực và định ra chiến lược dài hạn.

Cấu trúc khu vực có tác dụng tốt khi hành động điều phối được cần tới để phát triển sản phẩm đổi mới, thoả mãn trông đợi của khách hàng, hay duy trì phân mảnh thị trường. Nó thường được thấy nhất trong các tổ chức trung bình hay lớn đã chấp nhận chiến lược đa dạng hoá việc sản xuất ra nhiều sản phẩm, vận hành trong các doanh nghiệp và thị trường khác nhau, phục vụ cho nhiều loại khách hàng, và/hoặc bán sản phẩm trong các vùng địa lí khác nhau. Vì mỗi khu vực có đầy đủ các tài nguyên chức năng nên nó có thể đáp ứng cho yêu cầu về sản phẩm cá nhân, thị trường, khách hàng, hay miễn, và thích nghi nhanh chóng khi những nhu cầu này thay đổi.

Một nhược điểm tiềm năng của dạng này là làm giảm tính hiệu quả do mất cơ cấu tổ chức theo tỉ lệ. Chẳng hạn, thay vì 50 kĩ sư dùng chung một tiện nghi chung trong tổ chức theo chức năng, thì 10 kĩ sư có thể được phân công cho mỗi một trong năm khu vực. Sự trộn lẫn kĩ năng cần thiết được cần tới cho việc nghiên cứu theo chiều sâu có thể không xuất hiện, và các tiện nghi vật lí phải được nhân bản lên cho từng khu vực. Kĩ năng và chuyên môn kĩ thuật theo chiều sâu cũng có thể bị yếu đi, bởi vì các nhân viên đồng nhất với và đầu tư vào khu vực hơn là vào một chuyên môn đặc biệt. Việc điều phối ngang qua các khu vực cũng có thể là khác. Chẳng hạn, người bán hàng từ các khu vực khác nhau có thể cạnh tranh để hấp dẫn khách hàng chung.

Dạng ma trận. Một số tổ chức có nhu cầu cạnh tranh vì lợi nhuận cả theo cấu trúc chức năng và khu vực. Những tổ chức này cần cả tri thức chuyên gia kỹ thuật bên trong chức năng và sự phối hợp chiều ngang chặt chẽ xuyên qua các chức năng. Chẳng hạn, một tổ chức đa quốc gia có thể cần phối hợp xuyên chức năng, sản phẩm và vị trí địa lý.



Hình 1.4 Dạng ma trận.

Tổ chức kiểu ma trận, với cấu trúc được vẽ trong Hình 1.4, để cập tới tình thế khó khăn do việc đồng thời thực hiện cả hai cấu trúc này. Người quản lý vùng và người quản lý chức năng có quyền ngang nhau bên trong tổ chức, và nhân viên phải báo cáo lại cho cả hai. Phần lớn nhân viên đều được phân việc kép. Chẳng hạn một kỹ sư có thể được trao quản lý một dự án và phòng công nghệ. Khi hoàn thành một nhiệm vụ đặc biệt, người kỹ sư này lại trở về với phòng công nghệ và được trao cho một dự án mới. Người kỹ sư này có một cơ sở nhà, nhưng làm việc cả thời gian cho các dự án khác. Người đó báo cáo cho cả người quản lý dự án và người quản lý khu vực. Vì tài nguyên cần thiết trong tổ chức ma trận là được phân bố giữa các chức năng và khu vực, nên việc phân bổ tài nguyên dưới dạng lịch biểu và ưu tiên phải được thương lượng giữa các nhóm này.

Ma trận là thích hợp khi có những điều kiện sau. Thứ nhất là, sức ép của môi trường là từ hai hay nhiều chiều chủ chốt, như chức năng và sản phẩm hay chức năng và miễn. Những sức ép này có nghĩa là một cấu trúc quyền lực nhị nguyên là cần thiết để đạt tới cân bằng quyền lực trong việc phản ứng với sức ép của môi trường. Thứ hai là, môi trường nhiệm vụ của tổ chức vừa không chắc chắn vừa phức tạp. Những thay đổi thường xuyên bên ngoài và sự liên phụ thuộc cao độ giữa các đơn vị đòi hỏi việc móc nối cực kỳ hiệu quả trong cả chiều đứng và chiều ngang. Thứ ba là, cần sự tiết kiệm về qui mô trong việc dùng các tài nguyên nội bộ (con người, trang bị, thông tin, tiền mặt).

Cấu trúc ma trận làm cho tổ chức có khả năng đáp ứng với nhiều yêu cầu từ môi trường. Các nguồn tài nguyên có thể được phân bổ linh hoạt và tổ chức có thể thích nghi với những điều kiện cạnh tranh thay đổi, đều đặn và điều kiện tài nguyên. Nó cũng đưa ra một cơ hội cho các nhân viên để thu được hoặc các kỹ năng quản lý chung và chức năng, tùy theo mỗi quan tâm của họ.

Một vấn đề cơ bản đối với cấu trúc ma trận là xác định trách nhiệm và mối quan hệ quyền lực giữa người quản lý chức năng và dự án. Thường thiếu sự rõ ràng pháp lý, và mỗi người báo cáo cho hai ông chủ thì đôi khi phải đối mặt với các chiều hướng xung khắc. Những người trong cơ cấu ma trận này thường phải dành rất nhiều thời gian cho hội họp. Một số xung khắc dừng nên trong dạng ma trận là lành mạnh, vì nó buộc phải thảo luận và phối hợp để giải quyết các vấn đề đi đôi với cả chức năng và dự án. Nó cũng cung cấp cho quản lý cấp cao nhất khả năng gây ảnh hưởng mà chiều hướng này thì lại ít nhiều thu được sự chú ý trong các tình huống khác nhau. Tuy nhiên mỗi quan hệ báo cáo kép và việc bổ nhiệm cũng có thể gây ra sự mơ hồ về vai trò, gây cản trở cho việc phát triển nghề nghiệp, và làm yếu mối ràng buộc của nhân viên với nhóm chuyên nghiệp.

Nếu người quản lý không thích nghi với thông tin và quyền chia sẻ thông tin cần cho cấu trúc ma trận này, thì cấu trúc này sẽ không làm việc. Họ phải hợp tác với người khác, thay vì dựa trên chủ quyền chiều đứng trong việc ra quyết định. Dạng này đã được thấy là khó thiết lập và duy trì trong nhiều tổ chức. Khi nó thất bại, đây thường là vì một bên của cấu trúc quyền lực chi phối hay bởi vì các nhân viên còn chưa biết làm việc theo cách hợp tác.

Tiến hoá của các dạng tổ chức

Cấu trúc tổ chức là hữu cơ và tiến hoá. Ba dạng được mô tả ở trên biểu thị cho cái nhìn đơn giản hoá về những tính năng chủ chốt được lộ ra trong cấu trúc tổ chức. Tuy nhiên trong thực tế, cấu trúc tổ chức được hiểu rõ nhất như hình mẫu của việc thu xếp cho tổ chức vốn tiến hoá theo thời gian.

Phần lớn các tổ chức bắt đầu với cấu trúc chức năng. Khi chúng trở nên lớn và phức tạp hơn dưới dạng sản phẩm hay dịch vụ thì chúng được tổ chức lại theo cấu trúc phân khu vực, có lẽ với vài bộ phận chức năng tập trung. Trong quá trình phát triển, tổ chức có

thể thay đổi cấu trúc chức năng hay cấu trúc phân khu vực thuận tiện. Hai kiểu cấu trúc lại thường xuất hiện tiếp đó. Một dạng lại là tổ dự án hay sản phẩm chèn lắp lên cấu trúc chức năng để thuận tiện cho việc điều phối ngang qua các chức năng. Dạng lại kia là, với một vài chức năng chủ chốt, cần tiết kiệm về qui mô và chuyên dụng - như chế tạo hay bán hàng - có thể được tập trung lại và định vị tại trụ sở chính, vậy đặt chồng một cấu trúc chức năng lên trên một cấu trúc phân khu vực. Bằng việc tổ hợp các đặc tính của cả các cấu trúc chức năng và khu vực, các cấu trúc lại có thể đạt được ưu thế của cả hai dạng cấu trúc và tránh được một số nhược điểm của chúng.

Các cấu trúc chức năng, phân khu vực và ma trận đều có chung sự kế thừa cấp bậc của mối quan hệ báo cáo theo tầng. Các cơ chế chính thức - quyền quyết định, qui tắc làm việc, tiến trình vạch kế hoạch và lên ngân sách v.v.. đều nhấn mạnh vào các tiến trình cơ sở và không chính thức. Tuy nhiên, về sau đôi điều đang xảy ra cho khái niệm truyền thống của chúng ta về cấu trúc tổ chức. Một số cơ chế chính thức, như hệ thống cấp bậc địa vị chính thức, dường như giảm hay mất đi tầm quan trọng. Tổ chức không chính thức dường như xác định một cách có ý nghĩa hơn về tính hiệu quả của tổ chức, một phần bởi công ti đang tổ chức và tổ chức lại thường xuyên hơn.

Việc tổ chức lại thường dẫn đến việc xóa bỏ toàn bộ các tầng nhân công, đặc biệt là quản lý cấp trung. Những nhà quản lý còn lại có phạm vi kiểm soát rộng hơn, mà điều này lại đưa tới việc uỷ quyền ra các quyết định có ý nghĩa cho cấp dưới trong tổ chức. CNTT được dùng để phân phát thông tin cho việc ra quyết định trong toàn tổ chức, thay thế cho người quản lý cấp trung, thường trước đây vẫn được dùng như bộ lọc và điểm nối thông tin.

Biên giới giữa chức năng và các đơn vị tổ chức dường như mờ đi khi nhiều công việc được thực hiện theo “dự án” bởi các “tổ” và dùng các nhân viên “tuyển tạm”. Biên giới bên ngoài của tổ chức

cũng dần khó xác định hơn. Đã đến lúc các tiến trình sản xuất gắn chặt tổ chức với nhà cung cấp, làm cho họ trở thành một phần của tổ chức. Các liên minh khác với người bán, khách hàng và thậm chí với đối thủ cạnh tranh cũng làm mờ dần biên giới của tổ chức.

Mặc dầu không hiện tượng nào trong những điều này, lấy một cách riêng rẽ, làm nảy sinh ra một cấu trúc tổ chức mới, nhưng lấy một cách tập thể lại thì chúng quả có tạo ra cấu trúc tổ chức mới thật. Hiện đang có sự nổi lên của một cấu trúc tổ chức mới đó. Chúng ta sẽ gọi cấu trúc tổ chức mới đang nổi lên đó là cấu trúc mạng. Dạng này dường như có vẻ "phẳng hơn", bởi vì có ít tầng "chỉ huy và kiểm soát." Nó có thể bao gồm một số lớn các chuyên gia, với ít người quản lí chung hơn - và ít nhân viên thường xuyên hơn. Nó được đặc trưng bởi các tổ vốn hình thành và giải tán khi được yêu cầu. Điều quan trọng nhất là dạng mới này đánh dấu bước chuyển về chất từ tổ chức "chỉ huy và kiểm soát" truyền thống sang tổ chức "dựa trên thông tin". Dạng phân cấp bậc truyền thống và các hệ thống lập kế hoạch, kiểm soát và tài nguyên con người ăn khớp với việc dự đoán và quản lí các biến cố. Tương phản lại, dạng mạng lưới mới ăn khớp với việc khai thác tính không chắc chắn, thay vì làm giảm nó, nhấn mạnh vào sự thích nghi linh động trong một môi trường thay đổi thường xuyên.

Tính thích nghi này là ưu thế chính của cấu trúc mạng đang nổi lên. Bởi vì con người, quyền quyết định, vai trò, và mối quan hệ báo cáo còn tùy thuộc vào dự án hay biến cố đặc thù, chúng có thể bị thay đổi khi được yêu cầu. Đến lượt mình, điều này lại cho cấu trúc mạng một ưu thế là nhanh chóng và đáp ứng với các biến cố thay đổi, như các đột phá công nghệ bất ngờ, nước đi của kẻ cạnh tranh, hay các hiện tượng khác. Mặt khác cấu trúc mạng có thể dẫn tới việc nhân bản tài nguyên và khó lường hành động. Cấu trúc mạng dường như phù hợp nhất với môi trường linh động thay đổi nhanh chóng, và khi đổi mới là ưu thế chiến lược chủ yếu.

Từ các vấn đề về tổ chức này, chúng ta cần tìm hiểu một khái niệm tổng quát: hệ thống và một phương pháp luận làm việc khoa học: cách tiếp cận hệ thống.

1.2 HỆ THỐNG

1.2.1 ĐỊNH NGHĨA

Nếu có tồn tại một số ý định để định nghĩa từ hệ thống, thì phần lớn các ý định đó đều mô tả hệ thống như một tập hợp các phần tử, có liên hệ với nhau theo cách để đạt tới một mục đích chung. Chúng ta xem xét sau đây về các thành phần của hệ thống

PHẦN TỬ

Các phần tử, dù thuộc loại gì, cũng đều tạo nên chất liệu cơ bản của hệ thống. Các phần tử được định tính bởi: các vật, các bộ phận, đơn vị, các thành viên, các thành phần, v.v.. Một biến toán học, một quá trình, một thủ tục, một phương pháp, một thể thức, một kĩ thuật, một biến cố, thậm chí có thể còn là một nguyên tử, một bóng bán dẫn, một máy tính điện tử, một trung tâm nhiệt năng, một ngôi sao, tất cả đều có thể coi là phần tử của một hệ thống.

THUỘC TÍNH CỦA CÁC PHẦN TỬ

Tất cả các phần tử, dù bản chất gì: đồ vật, con người, tiến trình v.v.. đều có các tính chất hoặc các đặc trưng, gọi là **thuộc tính**. Mọi thuộc tính đều là biểu hiện bề ngoài cho sự tồn tại của một phần tử, để người ta có thể quan sát và biết được nó. Phần tử gây ảnh hưởng ít nhiều lên sự vận hành của hệ thống dưới các khía cạnh khác nhau: cách phản ứng, tốc độ, độ chính xác, độ chắc chắn, khả năng, tiến hoá... Sự vận hành của hệ thống gần như bao giờ cũng tạo ra việc thay đổi các đặc trưng của các phần tử của nó.

Chọn lựa phần tử là một việc quan trọng. Trong hệ thống xí nghiệp, việc lựa chọn phần tử là lựa chọn giữa việc sử dụng nhân công hoặc sử dụng máy móc, hoặc còn giữa các loại máy khác nhau và việc chọn lựa này dựa trên cơ sở các thuộc tính và giá.

QUAN HỆ GIỮA CÁC PHẦN TỬ

Hiện tại ta sẽ không biết là có một hệ thống thực nếu như các phần tử chỉ được đặt cạnh nhau và rời nhau, không có mối liên hệ gì với nhau cả. Liên kết giữa các phần tử có tầm quan trọng sống còn và tạo ra cho hệ thống các đặc trưng và các điều kiện vận hành. Các quan hệ giữa những phần tử này có thể có bản chất vật lí, bản chất thông tin (truyền tin, lệnh v.v...) hoặc mọi bản chất khác (lực hấp dẫn vũ trụ, sự tương đồng về văn hoá v.v...) Vị trí hoặc chức năng của một phần tử trong hệ thống tùy thuộc vào cả chất lượng riêng của nó (thuộc tính) lẫn quan hệ của nó với các phần tử khác. Việc đưa một phần tử bổ sung vào một hệ thống không chỉ làm xuất hiện các quan hệ mới với các phần tử cũ mà còn làm thay đổi cả các quan hệ tương hỗ giữa các phần tử đã có sẵn. Gốc rễ của việc phân tích hệ thống phần lớn nằm ở sự kiện là nó tập trung sự chú ý vào các quan hệ giữa các phần tử.

THỰC THỂ

Thừa nhận sự tồn tại của những liên hệ giữa các phần tử dẫn tới suy nghĩ rằng một hệ thống có cấu trúc thì cấu trúc đó sẽ làm cho nó trở thành một đơn vị hoàn chỉnh.

Theo một số tác giả, hệ thống được tạo nên từ các đơn vị được phối hợp qua bản chất hoặc nghệ thuật. Một số khác ưa sử dụng các thuật ngữ: hợp nhất, tập hợp, tổ hợp, kết hợp hoặc các cụm từ: rất phức tạp, rất hòa hợp, hợp các bộ phận, thực thể khái niệm và cụ thể. Để được coi là một hệ thống, một tổ hợp phải có liên hệ với môi trường bên ngoài qua các lối vào ra có ý nghĩa đối với các phần tử của nó. Chẳng hạn, con vật không thể tách rời khỏi

nguồn thức ăn, cây không thể tách rời khỏi đất và các tia nắng mặt trời, xí nghiệp không thể tách rời thị trường và nguồn lao động.

MỤC TIÊU CỦA HỆ THỐNG

Không thể hình dung được một thực thể mà lại không có mục đích gắn với hoạt động hoặc sự tồn tại của nó. Để diễn đạt tư tưởng này, nhiều tác giả đã không sử dụng từ cấu trúc (quá chung chung so với những gì liên quan tới khái niệm về tính mục đích) để sử dụng các cách diễn đạt rõ ý hơn như: "vận động đi tới sự thống nhất", "đạt mục tiêu", "chức năng sống cơ bản", "tuân theo kế hoạch khi phục vụ mục đích chung", hoặc các thuật ngữ: mục đích, mục tiêu, đề án. Do vậy đơn vị được chọn, gắn liền cố hữu với khái niệm hệ thống, không chỉ đơn thuần là một đơn vị tích tụ. Tổng thể phải hướng về một mục đích chung cho tất cả các phần tử. Chính vì hệ thống có mục đích chung nên nó khác biệt với một tập hợp các phần tử hoặc các quan hệ. Chẳng hạn, với mọi hệ thống sống, cây cỏ, con vật, con người, cơ cấu hoặc tổ chức, mục đích (cuối cùng) là thích nghi với môi trường, với hoàn cảnh.

KIỂM SOÁT

Để đảm bảo tính thống nhất và để theo đuổi mục tiêu của mình, hệ thống phải có các liên hệ đặc biệt về liên lạc. Mỗi khi một phần của đầu ra được sử dụng làm đầu vào thì tức là đã thiết lập được sự kiểm soát. Dù sơ khai hay phức tạp, kiểu kiểm soát này đều cho phép hệ thống tiến hoá trong những giới hạn khá rộng rãi. Thuật ngữ "kiểm soát", hiểu theo nghĩa của tự động hoá, chỉ ra một cách vận hành thuộc về hệ máy nội cân bằng và tự điều chỉnh.

GIỚI HẠN CỦA MỘT HỆ THỐNG

Để phân biệt hệ thống với môi trường, rõ ràng cần phải xác định giới hạn (hoặc biên giới) của hệ thống đó về phương diện vật lí hoặc khái niệm. Không phải bao giờ cũng dễ dàng làm được việc

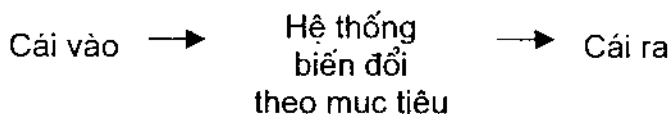
đó mỗi khi phải phân biệt chẳng hạn như bức tranh vẽ chiếc lọ cắm hoa hay khuôn mặt hai người đang nhìn nhau. Cách thức hiện thực hơn cả để phân giới hạn hệ thống là: thiết lập ra danh sách tất cả các phần tử của hệ thống và chính xác hoá các điểm nối hệ thống với môi trường. Một cách định biên hợp lý sẽ làm dễ dàng cho việc xác định các vấn đề cần khảo cứu. Nhưng giới hạn một hệ thống, đó là tự nguyện chấp nhận việc bỏ qua một số phần tử và một số quan hệ nào đó, bởi vì chúng dường như không có ảnh hưởng gì tới mục đích cuối cùng. Do vậy, trong thực tế, tất cả mọi sự giới hạn đều phụ thuộc chặt chẽ vào các mục tiêu. Muốn phủ nhận hoặc bỏ qua mối quan hệ này, đó chính là biểu hiện của cách xem xét hệ thống phiến diện.

MÔI TRƯỜNG

Cho tới nay, ta vẫn cố gắng định nghĩa hệ thống từ bên trong, tức là làm chính xác hoá khái niệm hệ thống. Cũng có thể đạt được cùng mục đích đó nếu xuất phát từ bên ngoài. Do vậy, cần phải tạo ra khái niệm về môi trường. Khái niệm về hệ thống và về môi trường giống như hai mặt của một tấm huân chương, giống như mặt trên và mặt dưới của một tờ giấy, theo nghĩa là chúng không thể tách rời và không thể đặt riêng biệt nhau được.

Môi trường của một hệ thống là tập hợp các phần tử không thuộc vào hệ thống nhưng có thể tác động vào hệ thống hoặc bị tác động bởi hệ thống. Cần phải làm chính xác hoá thêm hai điểm cho định nghĩa này. Thứ nhất, đó là việc các phần tử (của môi trường) phải ở bên ngoài hệ thống vì hệ thống chỉ có thể có được một ảnh hưởng hạn chế lên sự tồn tại, các thuộc tính hoặc quan hệ của chúng. Thứ hai là việc môi trường tập hợp các vật và người làm nên "dữ liệu", khuôn khổ, ràng buộc cho hệ thống hoạt động. Do vậy, đối với một xí nghiệp công nghiệp hoặc kinh doanh, khách hàng thuộc vào môi trường của hệ thống bởi vì họ được xác định một cách bất kì, theo một cách thức bên ngoài đối với hệ thống. Xí

ng nghiệp cũng có thể ảnh hưởng tới nhu cầu (theo một cách thức rất hạn chế) qua việc xuất bản, qua phương thức bán hàng v.v...



Môi trường

Hình 1.5 Khái niệm vận hành tổng quát của hệ thống.

Độc lập với độ phức tạp của hệ thống đã cho, các phân tử cơ sở là giống nhau về mặt hoạt động trong mọi hệ thống. Theo cách nhìn này, hệ thống, hiểu một cách đơn giản, là nơi diễn ra sự biến đổi từ cái vào của môi trường thành cái ra của hệ thống.

Mỗi hệ thống cũng có thể được phân tích thành các hệ con biểu diễn được qua cùng sơ đồ trên. Vậy một hệ thống phức tạp sẽ được phân rã dần ra thành từng phần ngày một sơ cấp hơn. Phương pháp đơn giản này đôi khi cho phép mô tả mọi thứ mà ta không thể giải thích ngay vào một lúc. Vậy nó có ưu điểm giải phóng được cách suy nghĩ và mở ra những viễn cảnh mới.

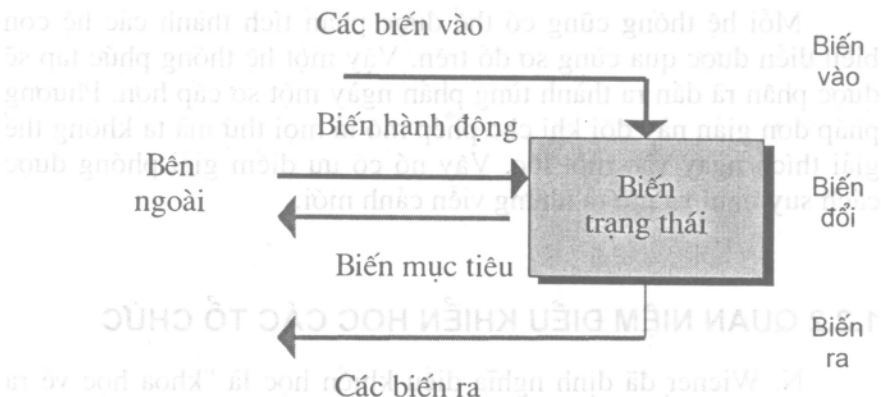
1.2.2 QUAN NIỆM ĐIỀU KHIỂN HỌC CÁC TỔ CHỨC

N. Wiener đã định nghĩa điều khiển học là "khoa học về ra lệnh và truyền thông báo tới người và máy". Điều khiển học mang các khái niệm và các luật cho phép làm rõ ra những nguyên lý chung của kiến trúc và của việc kiểm soát các hệ thống.

Ba phần tử cơ sở của một hệ thống điều khiển

Được định nghĩa theo quan điểm điều khiển học, hệ thống là một tập hợp các biến nhận biết được, lấy các giá trị khác nhau. Chính qua các biến đó, qua quan hệ giữa các đại lượng vào ra (đặc trưng cho các biến hoạt động) mà ta đánh giá được sự thực hiện của hệ thống. Đồng thời định nghĩa này chuyển sự chú ý vào bộ phận bị che khuất : cấu trúc của hệ thống, được giả thiết trước là tạo thành từ các phần tử bất biến - hoặc biến thiên chậm - (các biến trạng thái) để bảo đảm tính liên tục của một hoạt động nào đó.

Một hệ thống điều khiển học về cơ bản là một hệ mở, những trao đổi của nó với môi trường có tầm quan trọng lớn. Điều lưu ý này có ích trong việc phân biệt ba kiểu phần tử tạo nên một hệ thống điều khiển học : các biến vào, các biến ra, và phép biến đổi giữa hai kiểu biến đó. Việc sửa đổi lại sơ đồ cổ điển của một hệ thống dẫn tới cách biểu diễn ở Hình 1.6.



Hình 1.6 Các phần tử của một hệ điều khiển học.

Biến vào. Các nhân tố môi trường của hệ thống đòi hỏi hệ thống, áp đặt lên hệ thống những hành động xác định. Đối với hệ thống, đó là các ràng buộc mà nó phải tuân thủ. Người ta gọi chúng

là các biến vào. Giữa các biến vào có thể có một sự phân biệt quan trọng nếu đứng ở vị trí của thao tác viên ngoài hệ. Trong thực tế một số các biến vào đặc biệt nhạy cảm với sự thay đổi do thao tác viên tác động. Ta gọi chúng là các biến hành động (hoặc tham biến). Qua trung gian các biến này, thao tác viên có thể yêu cầu hệ thống hoạt động theo mục tiêu của nó.

Biến ra. Nếu một hệ thống phải chịu các ràng buộc, các hành động mà môi trường áp đặt lên nó thì nó cũng phản ứng lại môi trường thông qua các hoạt động của nó. Đó là các hành động được gọi là các biến ra. Theo quan điểm của thao tác viên, một số biến ra biểu thị mối quan tâm đặc biệt vì chúng có liên quan trực tiếp tới việc thực hiện mục tiêu của hệ thống: đó là các biến mục tiêu (cũng còn được gọi là các biến chủ chốt hoặc tiêu chuẩn).

Phép biến đổi. Các biến vào ở bên trong hệ thống, chịu tác động của các nhân tố bên trong (biến trạng thái) và chuyển qua một loạt các trạng thái kế tiếp. Tập hợp các giai đoạn chuyển hoá từ các biến vào thành các biến ra được gọi là một phép biến đổi và tương ứng với sự hoạt động của hệ thống. Phép biến đổi của một hệ thống điều khiển học có thể được tóm tắt theo nhiều cách :

- Thông qua đặc tả. Thường có thể phân rã một hệ thống phức tạp thành một loạt các hệ con thực hiện từng thao tác đơn giản. Trong trường hợp này, cơ cấu của những hệ con sẽ càng quen thuộc hơn nếu đã có trên thị trường các thiết bị thực hiện phép biến đổi đó.
- Thông qua tương tự. Để hiểu rõ hơn cách thức hoạt động của một tổ chức, điều đáng quan tâm và đáng gợi ý là hãy phân tích sự giống nhau về chức năng với các cơ chế khác đã được nghiên cứu kỹ hơn. Do vậy, việc xích lại gần nhau giữa người và máy đặc biệt góp phần vào sự tiến bộ của khoa học về con người.
- Thông qua thực nghiệm. Bằng cách chọn một vài biến vào để làm thí nghiệm, ta thử tìm hiểu cái ra tương ứng.

Khái niệm "hộp đen"

NGUỒN GỐC CỦA KHÁI NIỆM HỘP ĐEN

Khái niệm "hộp đen" nảy sinh trong lĩnh vực xây dựng các thiết bị điện tử. Trong thời kì chiến tranh thế giới lần thứ hai, thiết bị này đã trở thành phức tạp cực kì, và các chuyên gia, chỉ có thể duy trì sự hoạt động bình thường của chúng nếu có rất ít các thiết bị đó. Người ta có thói quen thay thế cả cụm bị hỏng bằng một cụm mới tương đương, chứ không thực hiện sửa chữa tận nơi nữa (tại chiến trường). Trong trường hợp đó, vậy chỉ cần dùng các kĩ thuật viên kiêm người sửa chữa là đủ và người thiết kế chỉ cần nêu rõ các đặc tả vào - ra, cho các hướng dẫn về cách ghép nối, chính xác hoá các phép thử chức năng cần thực hiện đối với mỗi "hộp đen" phải thay thế. Kĩ thuật viên không cần phải tháo máy (hộp) để biết rõ xem nó hoạt động thế nào. Chữ đen là do, trong thời kì chiến tranh này, màu đen thường được dùng cho các bảng hướng dẫn, do vậy khái niệm "hộp đen" được đặt ra. Mỗi "hộp đen" có một chức năng nào đó, hoàn thành một phép biến đổi xác định. Mọi phép biến đổi đều có thể được xem như một "hộp đen".

GIẢ THIẾT VỀ SỰ VẬN HÀNH

Khái niệm "hộp đen" cho phép không cần biết chính xác cách thức thực hiện một phép biến đổi, cách máy vận hành. Chỉ cần các cái vào và ra đủ ổn định để có thể quan sát được, từ đó rút ra được các giả thuyết về sự vận hành của nó.

Giả thuyết đầu tiên là giả thuyết về tính ổn định liên tục của phép biến đổi. Để cho ta có thể đoán trước được về các cái ra, do vậy sử dụng được "hộp đen", thì "hộp đen" cần phải vận hành trong tương lai giống như nó đã vận hành trong quá khứ. Một ví dụ cổ điển đặc biệt cho phép hoài nghi giá trị tuyệt đối của giả thuyết này là về quả mìn chống người hoặc chống chiến xa. Quả mìn được điều chỉnh để chỉ nổ sau khi đã bị s bắn đề lên. Nếu người thực

nghiệm suy luận rằng mình đã bị giảm lên (s- 1) lần mà không nỗ thì ở lần thứ s anh ta sẽ bị bắt ngờ.

Giả thuyết thứ hai là giả thuyết về tính độc lập của "hộp đen". Giả thuyết này rất thực tế để xây dựng một "hộp đen" mới, phức tạp hơn, từ các "hộp đen" khác, bằng cách có thể dự kiến trước các đặc trưng. Nhưng những liên hệ của chúng đôi khi có thể làm xuất hiện các tính chất mới không lường trước được.

SỬ DỤNG NGUYÊN LÝ "HỘP ĐEN" NHƯ MỘT KỸ THUẬT XÂY DỰNG

Trong thực tế, một "hộp đen" sẽ thực hiện tất cả những gì mà ta muốn nó thực hiện. Chính đặc thù rất thực tiễn này là cơ sở cho một kỹ thuật tổng quát để thiết kế. Phương pháp này thường được các kỹ sư sử dụng trong việc thiết kế các mạch điện tử phức tạp. Nó cũng có thể được sử dụng để thiết kế đủ loại hệ thống... trong đó có hệ thống tin.

Nguyên lý cơ bản của việc thiết kế qua "hộp đen" là như sau : thay vì xuất phát từ các phần tử và các tính chất của chúng để làm ra một thiết bị thực hiện một phép biến đổi nào đó, ta sẽ xuất phát từ chức năng của thiết bị và xác định dần các phần tử của nó. Và sau đây là các bước chính :

Tương hết, ta hình dung sự hoạt động của "hộp đen" khi nó nằm trong môi trường bằng cách xác định các cái vào, cái ra. Từ đó ta rút ra những điều nó phải làm, không cần biết nó thực hiện như thế nào. Rồi giai đoạn kế tiếp của quá trình là tự hỏi xem liệu nó có thể hoàn thành phép biến đổi đã dự kiến không và nó sẽ hoạt động thế nào. Để vượt qua các giai đoạn khó khăn nảy sinh do việc nghiên cứu này, "hộp đen" lớn đến lượt nó lại được phân rã rã thành các "hộp đen" nhỏ hơn, thực hiện một phần phép biến đổi đã dự kiến. Chắc chắn là, quá trình phân rã từ một "hộp đen" thành các "hộp đen" nhỏ hơn (tức là ít phức tạp hơn) được thực hiện cho tới khi ta biết rằng đã tồn tại (hoặc có thể xây dựng được) các thiết bị có thể thực hiện các chức năng đơn giản đã nêu. Nhiều khi quá

trình này dẫn đến các công việc không thể thực hiện được. Vậy phải thay đổi lại điểm xuất phát và bắt đầu lại việc phân rã. Cách tiếp cận thông qua "hộp đen" này đã có từ rất lâu trước khi N. Wiener sử dụng nó. Thực tế, các nhà khoa học thường phải đương đầu với các "hộp đen" mà họ phải cố gắng phát hiện ra bản chất của cơ chế, của sự biến đổi. Do vậy, ta có thể coi thói quen sử dụng khái niệm "hộp đen" như một kỹ thuật quan trọng nhất của điều khiển học.

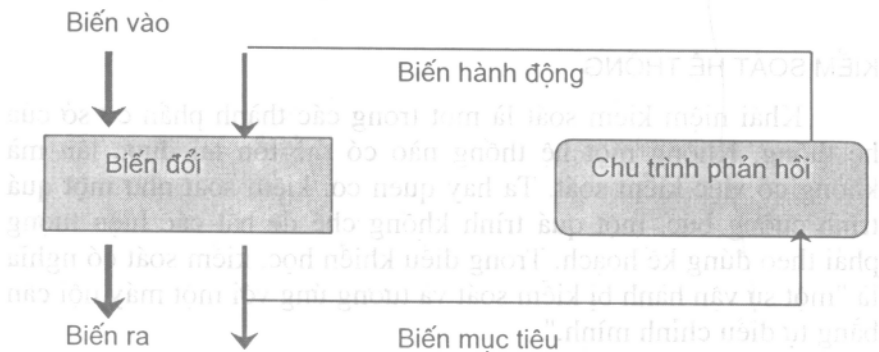
1.2.3 KIỂM SOÁT QUA "PHẢN HỒI"

"Phản hồi" là việc điều khiển một hệ thống nhờ đưa lại vào trong hệ thống này các kết quả của sự hoạt động của nó. Vậy đây không phải là cái gì khác hơn khả năng xác định hoạt động tương tự của hệ thống từ những hoạt động quá khứ của nó.

CHU TRÌNH "PHẢN HỒI"

Hình 1.7 minh họa nguyên lý hoạt động của hệ thống có "phản hồi". Thí dụ về "phản hồi", vừa đơn giản nhất vừa cổ điển nhất vừa hay được trích dẫn nhiều nhất, là nguyên lý bộ điều chỉnh của Watt.

Khái niệm "phản hồi" liên quan tới quá trình thông tin bởi vì nó sử dụng các kết quả của phép biến đổi của hệ thống, thông qua việc so sánh với sự thực hiện mong muốn để thay đổi các biến hoạt động. Trong một cơ cấu hoặc tổ chức, "phản hồi" được chuyển thành sự tồn tại của một mạng liên lạc nội bộ luân chuyển những kết quả của hành động hiện tại để xác định các hành động khác trong tương lai. Mỗi khi các kết quả này chỉ được sử dụng như dữ liệu số để điều chỉnh hệ thống (không thay đổi cơ bản về phép biến đổi) thì ta chỉ có một "phản hồi" đơn giản, cơ chế rất chung đối với các kỹ sư ở vị trí chỉ huy. Ngược lại, nếu thông tin được truyền cũng tạo ra sự thay đổi về bản chất của phép biến đổi thì vấn đề bây giờ trở thành một quá trình học tập.



Hình 1.7 Nguyên lí "phản hồi" .

"PHẢN HỒI" DƯƠNG VÀ "PHẢN HỒI" ÂM

Tùy theo quan điểm về hậu quả trực tiếp của chúng, các "phản hồi" được chia làm hai loại : "phản hồi" âm và "phản hồi" dương.

"Phản hồi" âm. "Phản hồi" được đặc trưng là âm tính nếu hoạt động của nó tạo cho hệ thống đi vào một tình huống đặc biệt, hướng về vị trí cân bằng, về một trạng thái xác định. Bộ điều chỉnh của Watt là một minh họa. Mục đích của nó là tạo ra một tình trạng nội cân bằng (tốc độ quay không đổi của trục mô tơ). Hiệu quả của "phản hồi" là ngăn cản những sai lệch thực tế làm sự vận hành của hệ thống xa với các điều kiện mong muốn.

"Phản hồi" dương. "Phản hồi" được đặc trưng là dương tính nếu hoạt động của nó tạo cho hệ thống xa rời với vị trí ban đầu. Trong trường hợp này, chức năng kiểm soát của cơ chế sẽ khuếch đại những sai lệch đo được. Chẳng hạn như việc phanh một chiếc xe lại. Hệ thống "phản hồi" phát hiện mọi chuyển động nhỏ nhất do người lái xe thực hiện và nó khuếch đại chúng lên thành lực đủ để dừng xe lại.

1.2.4 KIỂM SOÁT VÀ ĐIỀU CHỈNH HỆ THỐNG

KIỂM SOÁT HỆ THỐNG

Khái niệm kiểm soát là một trong các thành phần cơ sở của hệ thống. Không một hệ thống nào có thể tồn tại được lâu mà không có việc kiểm soát. Ta hay quen coi kiểm soát như một quá trình cưỡng bức, một quá trình khống chế để bắt các hiện tượng phải theo đúng kế hoạch. Trong điều khiển học, kiểm soát có nghĩa là "một sự vận hành bị kiểm soát và tương ứng với một máy nội cân bằng tự điều chỉnh mình."

Mỗi máy, mỗi hệ thống đều thực hiện một cái gì đó có thể xem như mục đích của nó. Cũng vậy, ta có thể nói rằng kiểm soát là chiến lược của máy để đạt tới mục đích. Nếu bằng cách tự đặt mình vào góc độ của thao tác viên thì kiểm soát, theo một nghĩa nào đó là việc đặt hệ thống vào trạng thái điều khiển. Việc điều khiển này giả thiết rằng ta có thể

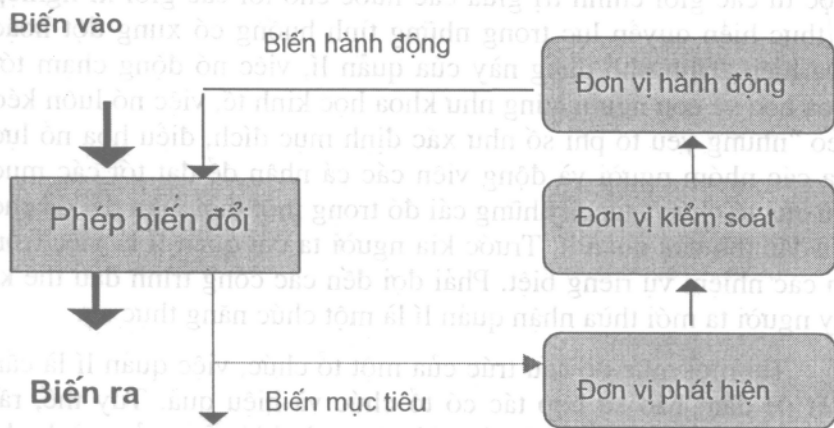
- lựa chọn được các biến mục tiêu có ích cho việc kiểm soát,
- đánh giá được những sai lệch của các biến mục tiêu theo các tiêu chuẩn, các chuẩn, và xác định các vùng sai lệch có thể chấp nhận được, - am hiểu phép biến đổi của hệ thống, tức là biết rõ mối quan hệ vào - ra để biết được cần can thiệp vào các biến hành động nào,
- xác định được các giá trị của các biến hành động để cho phép đưa các biến mục đích vào vùng dự kiến.

CÁC PHẦN TỬ CỦA HỆ THỐNG KIỂM SOÁT

Mọi hệ thống kiểm soát đều bao gồm bốn phần tử cơ bản

1. Một điều kiện hoặc đặc trưng (tức là một biến mục đích) được lựa chọn làm cách biểu diễn đặc biệt cho sự vận hành của hệ thống theo mục đích của nó.

2. Một bộ phận thích hợp để đo hoặc phát hiện các giá trị của biến mục đích.
3. Một đơn vị kiểm soát để so sánh các giá trị thu được với sự thực hiện vạch trong kế hoạch.
4. Một đơn vị can thiệp để biến đổi thông tin tới từ đơn vị kiểm soát thành hành động trên hệ thống.



Hình 1.8 Sơ đồ hệ thống kiểm soát .

1.3 QUẢN LÝ

Mục đích của ta là xây dựng một công cụ có khả năng phục vụ sự vận hành và đảm bảo sự tiến hoá cho các tổ chức. Do đó, đầu tiên cần xác định ý nghĩa của từ quản lý để ấn định được chính xác các mục tiêu cần theo đuổi.

Được đặc trưng bởi việc ra quyết định hướng dẫn hành động, quản lý từ xưa tới nay vẫn dựa trên nền tảng thông tin được sử dụng và hiểu một cách truyền thống theo kinh nghiệm mà thiếu một tư duy rõ ràng về tầm quan trọng và nhược điểm của nó. Để sửa chữa những thiếu sót này, và để đưa những phương pháp đã được áp

dụng rất thành công trong các lĩnh vực khác vào trong lãnh đạo và quản lí, các chuyên gia về quản lí càng ngày càng hướng về lí thuyết hệ thống. Vậy thế nào là quản lí?

Một chức năng phức tạp

Thuật ngữ "quản lí" bao hàm các nguyên lí chung áp dụng được từ các giới chính trị giữa các nước cho tới các giới xí nghiệp để thực hiện quyền lực trong những tình huống có xung đột hoặc thỏa hiệp. Tính phổ dụng này của quản lí, việc nó động chạm tới khoa học về con người cũng như khoa học kinh tế, việc nó luôn kéo theo "những yếu tố phi số như xác định mục đích, điều hòa nỗ lực của các nhóm người và động viên các cá nhân để đạt tới các mục tiêu của tổ chức", tất cả những cái đó trong một thời gian dài đã che giấu đặc thù của quản lí. Trước kia người ta coi quản lí là việc trộn lẫn các nhiệm vụ riêng biệt. Phải đợi đến các công trình đầu thế kỉ này người ta mới thừa nhận quản lí là một chức năng thực sự.

Tại mọi mức độ cấu trúc của một tổ chức, việc quản lí là cần thiết để đảm bảo sự hợp tác có tổ chức và hiệu quả. Tuy thế, rất hiếm người dành được toàn bộ tài năng và thời gian của mình chỉ để đáp ứng cho các nhiệm vụ quản lí. Hầu như ở mỗi cấp bậc, còn có các quy tắc khác trùm lên trên các hoạt động quản lí và lãnh đạo. Từ quản lí (management) - một từ tràn ngập trong mọi ngôn ngữ có nguồn gốc Italia "maneggiare". Bản thân từ này lại được rút ra từ chữ Latinh "*manus*" nghĩa là bàn tay. Theo nghĩa gốc, thực hiện quản lí là "nắm vững trong tay", "điều khiển vững tay". Theo một nghĩa nào đó, quản lí là một nghệ thuật khiến người khác phải làm việc. Tư tưởng này cũng đã được diễn đạt bằng cách ít thô bạo hơn: đạt được một việc gì đó thông qua một ai đó. Nhưng đó chẳng phải đó là một tập hợp các qui tắc hoặc nguyên lí hành động, các chuẩn mực về tính hiệu quả được thừa nhận một cách khách quan và được truyền bá qua học tập đó sao? J. W. Forrester xem quản lí "... như một quá trình biến đổi thông tin thành hành động, một quá trình tương đương với việc ra quyết định". Theo F. Kast, và J.

Rosenweig, "quản lí bao gồm việc điều hòa các nguồn tài nguyên về người và vật chất để đạt tới mục đích..."

Có thể chỉ ra bốn yếu tố cơ bản sau của quản lí :

1. hướng tới các mục đích,
2. thông qua con người,
3. với các kĩ thuật,
4. bên trong một tổ chức.

Trước khi đạt được các nghiên cứu sâu sắc trong lĩnh vực này, nghệ thuật quản lí thường được xem như cách ứng xử riêng của mỗi người lãnh đạo tổ chức, cách ứng xử phụ thuộc chặt chẽ vào hoàn cảnh kinh tế và xã hội, vào tính đa dạng của môi trường và nhất là vào tính cách của người chịu trách nhiệm.

Có lẽ cũng nên nhấn mạnh thêm ở đây rằng quản lí là cách biểu hiện một ý chí muốn biến đổi và thuận phục một tổng thể các hiện tượng. Đó là việc tạo ra sự kiện, thay vì để cho sự kiện xảy ra bột phát. Đây không phải là sự lắp ráp lại các nhiệm vụ rời rạc mà là một hoạt động phân biệt rõ ràng với các chức năng lớn khác của tổ chức. Sau khi đã nhấn mạnh vào khía cạnh tổng thể này, một nghiên cứu phân tích cho phép chia nó ra năm chức năng con: vạch kế hoạch, tổ chức, bố trí cán bộ, chỉ huy và kiểm soát.

1.3.1 VẠCH KẾ HOẠCH

Vạch kế hoạch, đó là dự kiến và lên chương trình cho hoạt động tương lai của tổ chức, trên quan điểm đạt tới các mục đích đã xác định. Là chức năng đặc trưng cao nhất của tổ chức, việc vạch kế hoạch được thực hiện bên trong các phòng, ban và các đơn vị đủ mọi tầm cỡ, giống như một quá trình liên tục để chọn lựa giữa các khả năng khác nhau dẫn tới các quyết định. Là chức năng mấu chốt, bao gồm việc tìm ra các phương tiện thích hợp nhất để thực hiện điều gì đó, vạch kế hoạch trước hết bao hàm ý muốn "chuẩn bị cho việc gặp gỡ với tương lai" và tiếp đó là dự kiến, "để quyết định

trước xem sẽ phải làm gì, làm như thế nào và ai sẽ phải làm... Nó cũng thiết lập một bước chuyển từ nơi ta đang ở tới nơi ta muốn đến". Là nỗ lực có hệ thống để thăm dò tương lai từ các sự kiện thực tế đã qua, dự kiến thực chất bao hàm việc tranh thủ các hoàn cảnh thuận lợi và việc thích nghi có hiệu quả với những thay đổi đã tạo ra hoặc đang trông đợi. Nó qui định chặt chẽ sự phát triển của các giai đoạn khác trong cả quá trình.

Việc xác định mục tiêu là giai đoạn đầu tiên trong các giai đoạn kế tiếp và tất nhiên là thiết yếu nhất. Các mục tiêu, đó là những cái đích được thiết lập để hướng mọi nỗ lực của cả một tổ chức hoặc của các phòng ban của nó. "Việc quản lí có hiệu quả bao giờ cũng là việc quản lí theo mục tiêu". Tổ chức chỉ có thể lớn lên và phát triển một cách điều hòa khi nó đã ấn định được các mục tiêu cụ thể.

Thiếu các mục tiêu được thiết lập trước, một tổ chức có nguy cơ thực hiện những đầu tư không thích hợp, sử dụng kém khả năng của cán bộ và sau một thời gian dài thực sự thu được những kết quả tồi tệ. Giống như các nguyên lí hoặc qui tắc hành động, các chính sách xuất phát từ những quyết định có giá trị lâu dài và áp dụng được vào những tình huống lặp lại. Chẳng hạn, mỗi khi lãnh đạo xí nghiệp quyết định các điều kiện mở tín dụng cho một khách hàng thì đó không phải là một chính sách nhưng việc xác định những điều kiện chung để mở tín dụng áp dụng được cho các khách hàng, thì lại có thể được xem là chính sách.

Trong một tổ chức, việc xác định và hình thành rõ ràng các chính sách cho phép từng người có trách nhiệm biết được mình phải làm gì vào mỗi lúc và bao giờ thì làm. Không cần phải bất cứ lúc nào cũng thỉnh thị cấp trên để giải quyết các vấn đề hợp với ý muốn của lãnh đạo.

Việc ấn định các mục tiêu và hình thành ra chính sách phải đi trước việc lập kế hoạch, chương trình và ngân sách ở mọi cấp bậc của tổ chức. Một kế hoạch, xác định ra trước một hành động, được

đặc trưng bởi ba yếu tố : liên quan tới tương lai, bao gồm một hành động và bao hàm sự can thiệp của con người để đảm bảo thực hiện kế hoạch. Còn chương trình hành động, đó là việc áp dụng các nguyên tắc đã được xác định thông qua các mục tiêu và chính sách vào một tình huống cụ thể. Mỗi chương trình hành động đều là một loạt các hoạt động được đặt ra để hướng dẫn thực hiện hành động. Nó trả lời cho câu hỏi : Làm gì ? Phân hoạch về thời gian hoàn tất việc xác định bản chất các công việc cần thực hiện. Đi kèm với các chương trình hành động (hoặc nằm trong chương trình), ngân sách là những ước lượng về nguồn tài nguyên cần thiết để hoàn thành một công việc trong chương trình và để thu được kết quả mong đợi.

Vạch kế hoạch không chỉ bao hàm có việc dự kiến (tất nhiên đó là khía cạnh chủ yếu được đề cập tới trong các đoạn sau) mà còn là quyết định. Do vậy, vạch kế hoạch, đó là " sự chọn lựa về cơ bản (và)... nếu có một vấn đề về việc vạch kế hoạch, thì vấn đề đó chỉ xuất hiện khi phát hiện ra được một cách hành động khác nữa" (Goetz).

1.3.2 TỔ CHỨC

Tổ chức, cánh cửa thứ hai của quản lí, tạo ra một tập hợp các phương tiện hoàn toàn hướng về phía thực hiện một chức năng. Đi từ chữ Hy Lạp "organon" nghĩa là "hài hòa", từ tổ chức nói lên một quan điểm rất tổng quát, "đó là cái đem lại bản chất thích nghi với sự sống". Ảnh hưởng của các khoa học về cách ứng xử có khuynh hướng thu hẹp nghĩa của từ này về những mối quan hệ con người phát sinh từ các hoạt động của các nhóm người. Trong trường hợp này, tổ chức bao gồm các quan hệ chính thức và không chính thức bên trong xí nghiệp và các quan hệ giữa xí nghiệp với môi trường.

Việc xem tổ chức như một hệ thống cấu trúc đường như có thể là một thái độ thực tế. Tùy theo quan điểm, tổ chức có thể được coi là việc tạo ra một khung cảnh lao động trong một môi trường nào đó, để vượt qua những khó khăn có thể nảy sinh ra bởi sự

không chắc chắn của những yếu tố tham dự. Cuối cùng, cũng có thể coi tổ chức như một quá trình logic, tự phát triển qua từng giai đoạn, để tạo nên một tổng thể thuần nhất các phương tiện con người và vật chất, hoàn toàn vận hành một cách hài hòa, nhằm đạt tới các mục đích. Thông qua quá trình này, "... nhà quản lí phải làm cho rõ ra những sự lộn xộn có thể xuất hiện theo thứ tự nào, tránh xa những xung đột giữa các cá nhân có liên quan tới công việc hoặc trách nhiệm của họ, và thiết lập được một môi trường thích hợp cho mọi người."

1.3.3 BỐ TRÍ CÁN BỘ

Đã từ lâu người ta phải thừa nhận rằng khi tổ chức có được người lãnh đạo giỏi thì đó sẽ là nơi mà chức năng quản lí được thực hiện đầy đủ nhất và mỗi người trong đó đều phát huy được khả năng của mình để tham gia tích cực vào công việc chung. "... Các xí nghiệp thành công hay thất bại trong thời kì dài, không phải vì lí do các vấn đề về thị trường, tư bản, trang thiết bị mà là do yếu tố con người ..." Trong phạm vi chức trách của nhà quản lí về bố trí cán bộ, xin nêu ra ba đặc trưng tiêu biểu nhất: chọn lựa, đào tạo và đánh giá cán bộ.

Liên quan tới việc chọn lựa nhà quản lí có ba yếu tố chung gây khó khăn cho việc đánh giá các ứng cử viên. Thứ nhất là việc các tình huống hoàn toàn mới sẽ xảy ra nhưng hiện tại chưa được xác định. Các tình huống này càng thay đổi nhiều khi càng đi lên cao theo bậc thang cấp bậc, và những phẩm chất cần có đối với mỗi bậc thang cấp bậc lại không phải là như nhau. Yếu tố thứ hai là sự trưởng thành của ứng cử viên và của những hiểu biết của ta về kinh nghiệm quá khứ của anh ta. Cuối cùng, mọi sự lựa chọn đều ít nhiều là việc của một cá nhân và phụ thuộc chặt chẽ vào người đương quyền. Các nhân tố chủ quan thường đóng vai trò ít ra cũng quan trọng bằng nhân tố khách quan.

Sau vấn đề lựa chọn là tới vấn đề đào tạo. Các cách tiếp cận thường gặp nhất lại khá đa dạng và bao gồm: lên kế hoạch nghề nghiệp, luân chuyển chức vụ, đặt vào vị trí trợ giúp, đào tạo tâm lý học qua việc đóng vai hoặc qua những thảo luận không hướng dẫn, thăng cấp tạm thời, tham dự các ủy ban, tổ chức hội thảo, đào tạo đại học. Ngày nay ta biết rằng việc đảm bảo sự phát triển của nhà quản lý là một quá trình lâu dài và khó khăn, không chỉ bao gồm có đào tạo sơ cấp vào lúc bắt đầu nghề chuyên nghiệp mà còn cả việc thích nghi và đổi mới các kiến thức và các tập quán thông qua việc đào tạo thường xuyên và nếu cần thì qua cả đào tạo nâng cao.

Trong hoạt động chuyên nghiệp của mình, mọi người có trách nhiệm đều phải đánh giá các cán bộ dưới quyền :

- để đảm bảo họ đáp ứng được vai trò của mình,
- để xác định cơ sở cho sự đãi ngộ,
- để đề bạt đúng vị trí.

1.3.4 CHỈ HUY

Nhà quản lý làm việc với con người. Những con người này không chỉ là một nhân tố sản xuất và không chỉ quan tâm tới các mục đích của tổ chức. Họ có những mục đích riêng và có những nhu cầu luôn nâng cao. Tính được tới cả những điều kiện này, việc chỉ huy trở thành một chức năng phức tạp có mục tiêu động viên người dưới quyền làm việc thật hiệu quả để thực hiện các mục tiêu từ ngắn hạn đến dài hạn của tổ chức. Do vậy để lãnh đạo người dưới quyền nhà quản lý phải động viên họ, giao thiệp với họ và phải là một người hướng dẫn mọi người cùng làm.

Việc động viên là làm cho con người bảo đảm được chức năng của mình trong những điều kiện hứng thú về mặt tâm lý đối với họ và có lợi cho tổ chức (thường là về mặt tài chính). Mọi nhà quản lý, bằng cách áp dụng các chính sách nhân sự riêng, và bằng

cách đảm bảo sự lãnh đạo có hiệu quả, đều muốn duy trì một mức độ tinh thần thỏa đáng cho từng nhóm hoặc từng cá nhân. Người quản lý phải giữ được niềm tin và bảo đảm cho mỗi người một sự an tâm đầy đủ.

Mặc dầu có mặt trong mọi khâu của quản lý, việc giao tiếp tạo nên một yếu tố đặc biệt quan trọng của chỉ huy. Ta có thể định nghĩa giao tiếp là sự trao đổi thông tin để đảm bảo hiểu biết tương hỗ và tạo ra niềm tin, cơ sở của mối quan hệ thân thiện giữa con người.

Vai trò lãnh đạo là một thành phần khác của chỉ huy. Có thể định nghĩa nó "như một ảnh hưởng trên nhiều người, được thực hiện theo tình huống và được chỉ đạo, thông qua quá trình giao tiếp, hướng về việc thực hiện một hoặc nhiều mục đích đã thiết lập." Vai trò lãnh đạo còn là "tập quán của nhà quản lý làm cho người dưới quyền làm việc với lòng tin cậy và năng nổ."

1.3.5 KIỂM SOÁT

Chức năng cuối cùng nổi bật này vừa là sự bổ sung vừa là sự tổng hợp của mọi chức năng khác. Kiểm soát có mục tiêu chủ yếu để đảm bảo rằng hoạt động của một tổ chức được phù hợp với kế hoạch nhằm đạt tới các mục đích đã lựa chọn. Kiểm soát không chỉ là ý thức đơn thuần tới các kết quả mà nó còn đòi hỏi thường xuyên phải sửa đổi kế hoạch, thích nghi lại tổ chức, và làm tiến hoá các phương pháp lãnh đạo. Dù cho được áp dụng vào đồ vật, con người hay hành động, quá trình kiểm soát bao giờ cũng gồm bốn bước :

1. Thiết lập ra các chuẩn, tức là các chỉ tiêu xác định để lấy làm cơ sở cho việc so sánh. Các chuẩn này biểu thị cho việc diễn đạt các đích của việc vạch kế hoạch của tổ chức dưới dạng có tính kỹ thuật và tính thực tiễn nhiều hơn.

2. Đo việc thực hiện. Khuynh hướng thực tế là thử phát hiện nhanh nhất các sai lệch để đảm bảo việc điều khiển mềm dẻo nhất.
3. Đánh giá các kết quả bằng cách so sánh với chuẩn.
4. Sửa chữa các sai lệch bằng cách tác động vào các nguyên nhân gây ra sai lệch.

1.4 NGƯỜI QUẢN LÝ

Mô tả phân loại về quản lý

MÔ HÌNH CỔ ĐIỂN VỀ QUẢN LÝ.

Mô hình cổ điển về quản lý mô tả nhà quản lý làm gì, đã là vấn đề không được bàn cãi nữa trong hơn 70 năm qua từ những năm 1920. Henri Fayol và những người khác lần đầu tiên đã mô tả năm chức năng của nhà quản lý là *lập kế hoạch, tổ chức, bố trí cán bộ, chỉ huy hay ra quyết định và kiểm soát*. Mô tả về các hoạt động quản lý này đã chi phối việc quản lý trong một thời gian dài và còn phổ biến ngày nay.

Nhưng những thuật ngữ này thực tế đã mô tả các chức năng quản lý và lại không thoả đáng khi mô tả thực tại điều các nhà quản lý phải thực hiện. Những thuật ngữ này không đề cập tới điều nhà quản lý phải làm khi họ lập kế hoạch, quyết định công việc hay kiểm soát công việc của người khác. Chúng ta cần một sự hiểu biết mịn hơn về cách thức nhà quản lý thực tế hành xử.

MÔ HÌNH HÀNH VI

Các nhà khoa học hành vi đương đại đã phát hiện từ quan sát rằng các nhà quản lý không hành xử như mô hình quản lý cổ điển đã

làm cho chúng ta tin như vậy. Kotter (1982), chẳng hạn, đã mô tả hoạt động buổi sáng của bà chủ tịch một tổ chức quản lý đầu tư.

- 7:35 Richardson tới làm việc, mở tủ, lấy cà phê và bắt đầu viết danh sách các hoạt động trong ngày.
- 7:45 Bradshaw (phụ tá) và Richardson thảo luận về một số chủ đề và cho nhau xem ảnh chụp kì nghỉ hè vừa rồi.
- 8:00 Họ nói chuyện về lịch biểu công việc trong ngày.
- 8:20 Wilson (phụ tá) và Richardson nói chuyện về một số vấn đề nhân sự, nói chuyện phiếm trong quá trình.
- 8:45 Thư kí của Richardson tới và họ thảo luận về căn hộ mới của cô ấy và bố trí cuộc họp muộn trong sáng.
- 8:55 Richardson dự cuộc họp do một trong các phụ tá chủ trì. Có 30 người dự họp và Richardson đọc bài nói trong buổi họp.
- 11:05 Richardson và phụ tá trở về văn phòng và thảo luận một vấn đề khó khăn. Họ cố gắng xác định vấn đề và vạch đại cương các phương án có thể. Bà ấy để cho việc thảo luận lúc thì chạy gần lúc chạy xa chủ đề. Cuối cùng họ đồng ý về bước tiếp.

Trong thí dụ này, khó xác định được hoạt động nào tạo nên việc lập kế hoạch, điều phối và ra quyết định của Richardson. Mô hình hành vi nói rằng hành vi thực tế của nhà quản lý dường như ít tính hệ thống mà nhiều phi hình thức, ít có suy nghĩ mà nhiều phản ứng, ít được tổ chức tốt mà nhiều phù phiếm hơn là những gì sinh viên về hệ thống tin và ra quyết định nói chung trông đợi ở họ.

Các nhà quan sát thấy rằng hành vi quản lý thực tại có năm thuộc tính khác khá xa với mô tả cổ điển: Trước hết, các nhà quản lý thực hiện khá nhiều việc với nhịp điệu không giảm - các khảo cứu đã cho thấy rằng nhà quản lý tham dự vào hơn 600 hoạt động khác nhau mỗi ngày, mà không dứt đoạn nhịp độ. Thứ hai là, các hoạt động quản lý bị đứt đoạn; phần lớn các hoạt động kéo dài không quá chín phút; chỉ có 10 phần trăm các hoạt động là vượt

quá một giờ. Thứ ba, các nhà quản lí thích lời bàn tán, lời đồn đại, chuyện tâm phào - họ cần thông tin hiện thời, xác định và không theo thể thức (các thông tin in ra thường quá cũ). Thứ tư, họ thích dạng trao đổi miệng hơn dạng viết bởi vì trao đổi trực tiếp cung cấp độ mềm dẻo lớn hơn, ít đòi hỏi nỗ lực hơn và đem đến câu trả lời nhanh hơn. Thứ năm, các nhà quản lí đặt ưu tiên cao cho việc duy trì sự tiếp xúc phức tạp và đa dạng vốn hành động như một hệ thống tin không chính thức.

Các nhà quản lí có hiệu quả thường dành rất nhiều thời gian xây dựng một mạng lưới người trung gian bao gồm những người thực tế ở mọi cấp bậc tổ chức, từ nhân viên kho tới thư kí nhân sự, tới các nhà quản lí khác và cấp quản lí cao. Các nhà quản lí dùng mạng lưới của họ để thực hiện chương trình nghị sự cá nhân, và để hoàn thành mục tiêu riêng của họ.

Phân tích hành vi thường ngày của nhà quản lí, Mintzberg thấy rằng có thể phân loại chúng thành mười vai trò quản lí. Vai trò quản lí là những kì vọng về các hoạt động mà nhà quản lí nên thực hiện trong tổ chức. Mintzberg thấy rằng những vai trò quản lí này rơi vào ba phạm trù: liên con người, thông tin và quyết định.

Vai trò liên con người. Các nhà quản lí hành động như cái bung xung cho tổ chức khi họ đại diện cho công ti mình với thế giới bên ngoài và thực hiện bốn phận tượng trưng như trao phần thưởng cho nhân viên. Các nhà quản lí hành động như các vị lãnh đạo, cố gắng động viên, cố vấn và hỗ trợ cho người dưới quyền. Các nhà quản lí cũng hành động như một mối liên hệ giữa nhiều mức trong tổ chức; trong từng mức này, họ tựa như mối liên hệ trong các thành viên của nhóm quản lí. Các nhà quản lí cung cấp thời cơ và ưu đãi, điều họ trông đợi được đền đáp lại.

Vai trò thông tin. Các nhà quản lí hành động như các trung tâm thần kinh trong tổ chức của mình, nhận hầu hết những thông tin cụ thể, mới nhất và phân bố lại nó cho những người cần biết. Do

đó các nhà quản lí cũng là những người phán tán thông tin và là người phát ngôn cho tổ chức của họ.

Vai trò ra quyết định. Các nhà quản lí ra quyết định. Họ hành động như một doanh nhân bằng cách khởi đầu các loại hoạt động mới; họ xử lí những rắc rối nảy sinh trong tổ chức; họ cấp phát tài nguyên cho nhân viên cần tới; và họ thương lượng về các xung đột và làm trung gian giữa các nhóm xung khắc trong tổ chức.

Vai trò của nhà quản lí được tóm tắt như sau:

Vai trò	Hành vi	Hệ thống hỗ trợ
Vai trò liên con người		Không tồn tại
Bung xung	liên con	Không tồn tại
Lãnh đạo	người	Hệ thống liên lạc điện tử
Liên hệ		
Vai trò thông tin		
Trung tâm thần kinh		hệ thông tin quản lí
Người phán tán	xử lí	Thư tín, hệ văn phòng
Người phát ngôn	thông tin	Trạm làm việc văn phòng
Vai trò ra quyết định		
Doanh nhân		
người xử lí rắc rối		Không tồn tại
người cấp phát tài nguyên	ra quyết định	Không tồn tại
người thương lượng		Hệ trợ giúp quyết định
		Không tồn tại

Nhà quản lí đóng vai trò chủ chốt trong tổ chức. Trách nhiệm của họ bao quát một phạm vi từ ra quyết định, tới viết báo cáo, dự họp, thu xếp các buổi tiệc mừng sinh nhật. Để xác định hệ thông tin có thể đem lại ích lợi gì cho nhà quản lí, trước hết chúng ta phải kiểm tra lại xem nhà quản lí làm gì và họ cần những thông tin nào để ra quyết định và các chức năng khác của họ. Chúng ta cũng phải

hiểu cách ra quyết định và loại quyết định nào mà hệ thống tin chính thức có thể hỗ trợ được.

1.5 RA QUYẾT ĐỊNH

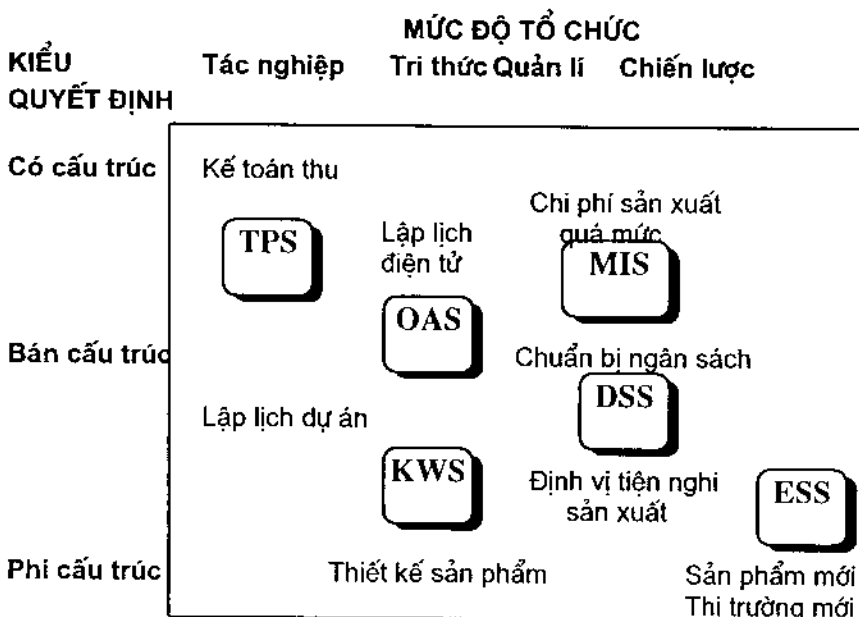
Các nhà lí thuyết quản lí cổ điển coi việc ra quyết định là trung tâm của hoạt động quản lí. Mặc dầu chúng ta không biết rằng điều này có thực là như vậy không thì việc ra quyết định vẫn còn là một trong những vai trò thách thức cho các nhà quản lí. Hệ thống tin đã giúp các nhà quản lí trao đổi và phân bố thông tin; tuy nhiên, chúng chỉ cung cấp sự trợ giúp có giới hạn cho việc ra quyết định quản lí. Bởi vì việc ra quyết định là một lĩnh vực mà người thiết kế hệ thống đã tìm kiếm nhiều nhất nên chúng ta sẽ chuyển sự chú ý sang lĩnh vực này.

1.5.1 TIẾN TRÌNH RA QUYẾT ĐỊNH

Việc ra quyết định có thể được phân loại theo mức độ tổ chức, tương ứng với các mức chiến lược, quản lí, tri thức và vận hành, của tổ chức. Việc **ra quyết định chiến lược** xác định ra các mục tiêu, tài nguyên và chính trị của tổ chức. Ra quyết định để **điều khiển quản lí** chủ yếu liên quan tới cách sử dụng có hiệu quả và hiệu suất các nguồn tài nguyên và cách thức các đơn vị thực hiện công việc tốt đến đâu. Ra quyết định **kiểm soát tác nghiệp** xác định cách thức tiến hành các nhiệm vụ xác định do những người ra quyết định cấp chiến lược và quản lí cấp trung gian ấn định. **Ra quyết định mức tri thức** xử lí với việc đánh giá những ý tưởng mới về sản phẩm và dịch vụ, cách thức trao đổi các tri thức mới và cách phân phối thông tin trong toàn tổ chức.

Bên trong các mức ra quyết định này, các nhà nghiên cứu còn phân chia quyết định thành loại *có cấu trúc* và *phi cấu trúc*. Các

quyết định phi cấu trúc là những quyết định trong đó người ra quyết định phải đưa ra đánh giá, ước lượng và cách nhìn riêng vào việc xác định vấn đề. Mỗi một trong các quyết định này đều là mới, quan trọng và không thường lệ, và không có sẵn thủ tục đã được hiểu rõ hay chấp thuận để tạo ra chúng. Các **quyết định có cấu trúc** thì ngược lại, có tính chất lặp lại, thường lệ và bao gồm một thủ tục xác định để xử lý sao cho người ta không phải xử lý mới mỗi khi chúng tới. Một số quyết định thuộc loại bán cấu trúc; trong những trường hợp đó, chỉ có phần vấn đề đã có câu trả lời rõ ràng mới được cung cấp bằng thủ tục đã được chấp thuận.



TPS = hệ thống xử lý giao tác
OAS = hệ tự động văn phòng
KWS = hệ công tác tri thức

MIS = hệ thông tin quản lí
DSS = hệ trợ giúp quyết định
ESS = hệ hỗ trợ điều hành

Hình 1.9 Các loại hệ thống tin tại các mức tổ chức khác .

Tổ hợp hai quan điểm về thủ tục ra quyết định này tạo ra phân hình xám vẽ trong Hình 1.9. Nói chung, nhân viên kiểm soát tác nghiệp thường hay phải đương đầu với các vấn đề có cấu trúc rõ. Ngược lại, những người làm kế hoạch chiến lược phải giải quyết các vấn đề phi cấu trúc cao độ. Nhiều vấn đề mà các công nhân tri thức gặp phải thì rất thường là phi cấu trúc. Tuy thế, mỗi mức của tổ chức đều chứa cả các vấn đề có cấu trúc và phi cấu trúc.

Trong quá khứ, phần lớn sự thành công trong hệ thống tin hiện đại đều do xử lí với các vấn đề có cấu trúc, tác nghiệp và kiểm soát quản lí. Nhưng bây giờ phần lớn những ứng dụng sôi động đều xuất hiện trong lĩnh vực quản lí, tri thức và lập kế hoạch chiến lược, chỗ có các vấn đề hoặc là bán cấu trúc hoặc hoàn toàn phi cấu trúc.

1.5.2 CÁC GIAI ĐOẠN RA QUYẾT ĐỊNH

Ra quyết định không phải là một hoạt động riêng lẻ xảy ra ngay một lúc. Tiến trình này bao gồm nhiều hoạt động khác nhau xảy ra tại nhiều thời điểm. Simon (1960) đã mô tả bốn giai đoạn trong việc ra quyết định: hiểu biết, phác hoạ, chọn lựa và thực hiện.

Hiểu biết bao gồm việc *xác định* và *hiểu thấu* vấn đề xuất hiện trong tổ chức - tại sao có vấn đề và với ảnh hưởng nào. Các hệ thống tin quản lí MIS truyền thống cung cấp rất nhiều thông tin chi tiết có thể giúp cho việc xác định vấn đề, đặc biệt nếu hệ thống này báo cáo lại những ngoại lệ.

Trong lúc **phác hoạ** giải pháp, cá nhân *phác hoạ* ra các giải pháp có thể cho vấn đề. Các hệ trợ giúp quyết định DSS nhỏ là lí tưởng cho giai đoạn ra quyết định này bởi vì chúng vận hành trên các mô hình đơn giản, có thể được xây dựng nhanh chóng, và có thể được vận hành với những dữ liệu hạn chế.

Chọn lựa bao gồm việc *chọn* trong số các phương án giải pháp. Tại đây người ra quyết định có thể cần một hệ thống trợ giúp

quyết định DSS rộng hơn để từ nhiều dữ liệu hơn xây dựng ra nhiều phương án và dùng các mô hình phân tích phức tạp để tính mọi chi phí, hậu quả và cơ hội.

Trong khi **thực hiện** giải pháp, nhà quản lí có thể dùng một hệ thống báo cáo để cung cấp các báo cáo đều đặn về tiến độ của giải pháp đặc biệt. Hệ trợ giúp quyết định có thể bao gồm từ các hệ thông tin quản lí MIS đầy đủ cho tới các hệ thống nhỏ hơn nhiều cũng như phần mềm lập kế hoạch dự án vận hành trên máy vi tính.

Nói chung, các giai đoạn của việc ra quyết định không nhất thiết tuân theo đường tuyến tính. Tại bất kì điểm nào trong tiến trình làm quyết định bạn cũng có thể quay vòng lại các giai đoạn trước. Chẳng hạn, người ta thường đi tới nhiều phác thảo nhưng không thể chắc chắn được về phác thảo nào đáp ứng với các yêu cầu cho một vấn đề đặc biệt. Tình huống này đòi hỏi có thêm công việc thu thập thông tin. Một cách khác là khi đang trong tiến trình thực hiện một quyết định người ta mới phát hiện ra rằng nó không làm việc. Trong trường hợp như thế người ta bị buộc phải lặp lại các giai đoạn phác thảo hay chọn lựa.

MÔ HÌNH CÁ NHÂN RA QUYẾT ĐỊNH

Người ta định xây dựng ra một số mô hình mô tả cách thức con người ra quyết định. Một số trong những mô hình này tập trung vào việc ra quyết định cá nhân, trong khi số khác tập trung vào ra quyết định theo nhóm.

Giả thiết cơ bản nằm sau mô hình ra quyết định cá nhân là ở chỗ con người có một cảm giác lí trí nào đó. **Mô hình lí trí** về hành vi con người được xây dựng trên ý tưởng là sự tham dự của con người về cơ bản là nhất quán, hợp lí, có tính toán tối đa giá trị. Theo mô hình này, một cá nhân xác định mục tiêu, phân hạng mọi hành động có thể có theo đóng góp của họ vào những mục tiêu đó, và chọn lựa các phương án để đóng góp được nhiều nhất cho các

mục tiêu đó. Cá nhân có thể hoàn toàn nhận biết mọi phương án và hậu quả.

Những phê phán về mô hình này chỉ ra rằng trong thực tế người ta không thể xác định được mọi phương án và rằng phần lớn các cá nhân không có một mục tiêu riêng lẻ và do đó không thể nào phân hạng được mọi phương án và hậu quả. Có lẽ phê phán cơ bản nhất là ở chỗ nhiều quyết định thì phức tạp đến mức việc tính toán sự chọn lựa (cho dù được máy tính thực hiện) cũng thực tế là không làm được (chỉ trong trò chơi cờ "đơn giản" cũng đã phải mất 10^{95} năm tính toán với tốc độ 1 triệu lệnh một giây (MIPS) để kiểm thử mọi nước đi, nước đối lại, rồi nước đối của nước đối lại từ lúc bắt đầu cho tới khi kết thúc chỉ cho nước đi đầu tiên!) Một thay đổi cho mô hình lí trí nói rằng thay vì tìm qua tất cả các phương án thì người ta thực tế chỉ chọn phương án có sẵn đầu tiên rồi chuyển chúng tới mục tiêu tối hậu. Một sự thay đổi khác thì đổi mô hình lí trí bằng cách gợi ý rằng trong việc ra quyết định chính sách, người ta thường cố kiểm chế các thay đổi qui mô rộng và chọn những chính sách giống chính sách trước đó. Cuối cùng, một số học giả chỉ ra rằng người ta lại không chọn lựa, nhưng việc ra quyết định đó là một tiến trình liên tục trong đó những quyết định cuối cùng bao giờ cũng được sửa đổi.

Các nhà tâm lí hiện đại còn định lượng thêm cho mô hình lí trí bằng nghiên cứu phát hiện ra rằng con người khác biệt về *cách họ làm tới đa giá trị của mình* và trong khuôn khổ tham khảo họ quen dùng chúng để hiểu thông tin và chọn lựa. **Phong cách nhận thức** mô tả khuynh hướng cá nhân nền tảng hướng tới việc xử lí thông tin, chọn lựa phương án và đánh giá hậu quả. McKenney và Keen mô tả về hai phong cách nhận thức: kiểu hệ thống và kiểu trực giác. **Những người ra quyết định kiểu hệ thống** tiếp cận tới vấn đề bằng cách cấu trúc nó theo một phương pháp chính thức. Họ đánh giá và thu thập thông tin theo phương pháp có cấu trúc. **Những người ra quyết định kiểu trực giác** tiếp cận tới vấn đề bằng nhiều phương pháp, dùng cách thử và sai lầm để tìm ra giải

pháp. Họ có khuynh hướng không cấu trúc việc thu thập hay đánh giá thông tin. Cả hai kiểu này đều không trội hơn nhau. Còn có những cách khác của kiểu lí trí.

Nghiên cứu gần đây hơn của tâm lí thực sự thách thức giả thiết nền tảng về các mô hình lí trí bằng cách chỉ ra rằng con người có khuynh hướng sẵn có là có thể bóp méo việc ra quyết định. Tôi tệ hơn nữa, con người có thể bị thao túng trong việc chọn các phương án mà họ có thể loại bỏ đơn thuần bởi việc thay đổi *khôn khổ tham chiếu*.

MÔ HÌNH TỔ CHỨC RA QUYẾT ĐỊNH

Việc ra quyết định thường không được thực hiện chỉ bởi một cá nhân mà bởi toàn nhóm hay tổ chức. **Mô hình tổ chức** cho việc ra quyết định tính tới những đặc trưng cấu trúc và chính trị của một tổ chức. Các mô hình quan liêu, chính trị và thậm chí "hộp rác" đã được đề nghị để mô tả cách ra quyết định xảy ra trong tổ chức. Ta sẽ xem xét từng mô hình này.

MÔ HÌNH QUAN LIÊU

Theo **mô hình quan liêu**, mục tiêu quan trọng nhất tổ chức là gìn giữ bản thân tổ chức ấy (như duy trì ngân sách, nhân lực và lãnh thổ). Việc rút bớt độ bất trắc là một mục tiêu chính khác. Chính trị có khuynh hướng tăng lên, chỉ khác biệt chút ít với quá khứ, bởi vì những sự bắt đầu chính trị hợp lí kéo theo quá nhiều bất trắc. Các mô hình này miêu tả tổ chức nói chung không "chọn lựa" hay "quyết định" theo một nghĩa hợp lí. Thay vì thế, theo mô hình quan liêu, bất kì cái gì tổ chức làm cũng đều là kết quả các thủ tục tác nghiệp chuẩn (SOP) được mài giũa qua nhiều năm sử dụng tích cực.

Vấn đề mà bất kì tổ chức lớn nào cũng phải đương đầu bao giờ cũng quá lớn lao và quá phức tạp không thể được toàn bộ tổ chức chú trọng tới hết. Những hành động đặc biệt thường được tổ chức chọn lựa do vậy thường thuộc vào một hay một số đơn vị con

thay vì toàn bộ (như bộ phận tiếp thị, sản xuất, tài chính, nhân lực). Từng đơn vị con này đều có một số thủ tục tác nghiệp chuẩn - các kỹ thuật đã được thử và chứng minh - rằng nó đưa đến việc giải quyết vấn đề. Các tổ chức hiếm khi thay đổi những SOP này bởi vì điều đó có thể dẫn đến thay đổi nhân sự và đưa tới rủi ro (ai biết được liệu một kỹ thuật mới có làm việc tốt hơn kỹ thuật cũ không?)

Được gắn với nhau, các vấn đề của các đơn vị con lập nên một phạm vi những hành động hiệu quả ngắn hạn mà người lãnh đạo tổ chức có thể tiến hành. Mặc dầu giới quản lý cấp cao và người lãnh đạo được thuê để điều phối và lãnh đạo tổ chức, họ thực tế bị mắc vào bẫy của các đơn vị con cục bộ vốn chuyển giao thông tin cho họ và cung cấp các giải pháp chuẩn. Giới quản lý cấp cao không thể quyết định hành động theo cách mà các đơn vị con cấp dưới không thể ủng hộ được.

Một số tổ chức tất nhiên cứ tiến hành thay đổi; họ học cách hành xử mới, và họ có thể đứng đầu. Nhưng tất cả những thay đổi này đều đòi hỏi thời gian dài. Bạn hãy nhìn quanh và bạn sẽ thấy nhiều tổ chức đang làm những cái gần giống như cái họ đã làm mười, hai mươi hay thậm chí ba mươi năm trước.

MÔ HÌNH CHÍNH TRỊ CHO SỰ CHỌN LỰA CỦA TỔ CHỨC

Quyền lực trong tổ chức được chia sẻ; ngay cả người công nhân mức thấp nhất cũng có một quyền nào đó. Trên đỉnh, quyền lực được tập trung vào tay một vài người. Với nhiều lý do, các nhà lãnh đạo mang ý kiến khác nhau về việc tổ chức phải làm gì, và, như ta đã làm rõ từ đầu chương này, những khác biệt đó làm nảy sinh vấn đề.

Mỗi cá nhân trong tổ chức, đặc biệt tại đỉnh, là một diễn viên chính trong trò chơi chính trị. Trong các **mô hình chính trị** về ra quyết định, điều tổ chức làm là kết quả của sự thoả thuận đạt được giữa các nhà lãnh đạo chủ chốt và các nhóm quyền lợi. Các hành động không nhất thiết hợp lý ngoại trừ theo nghĩa chính trị. Hậu quả

không phải là điều bất kì cá nhân nào cũng mong muốn. Thay vì thế, hành động chính trị-tổ chức là một sự thỏa hiệp, một sự trộn lẫn các khuynh hướng xung khắc. Tổ chức không đi tới "giải pháp" "được chọn" để giải quyết một "vấn đề" nào đó. Họ chỉ đạt tới sự thỏa thuận phản ánh mỗi xung khắc, những người đặt cọc chính, những quyền lợi khác nhau, quyền lực bất bình đẳng và sự lẫn lộn tạo nên chính trị.

1.6 HỆ THÔNG TIN

Với một nền tảng vững chắc về bản chất của tổ chức, bạn nay đã sẵn sàng nhìn gần hơn vào mối quan hệ giữa hệ thông tin và tổ chức. Ta sẽ bắt đầu bằng việc mô tả việc phát triển vai trò của hệ thông tin bên trong tổ chức trước khi ta xem xét đến ảnh hưởng của công nghệ hệ thông tin lên tổ chức.

Tiến hoá của hệ thông tin

Tổ chức có một tác động trực tiếp lên công nghệ thông tin bằng việc ra quyết định về cách sử dụng công nghệ thông tin và quyết định về vai trò của nó trong tổ chức. Hệ thông tin bắt đầu từ những năm 1950 như một hệ thống mức tác nghiệp được thiết kế để xử lí các giao tác sơ cấp nhưng sống còn như kiểm tra thanh toán. Vào cuối những năm 1960, các hệ thống quản lí đã được dùng để điều phối và kiểm soát; và vào những năm 1970 chúng đã được dùng cho lập kế hoạch và mô phỏng. Đến những năm 1980, hệ thông tin đã bành trướng thành các hệ thống trợ giúp quyết định và các hệ thống lập kế hoạch chiến lược xây dựng theo đặt hàng. Vào những năm 1990, hệ thông tin đang giúp tạo ra và phát tán tri thức và thông tin trong toàn bộ tổ chức qua những hệ thống công tác tri thức, những ứng dụng cung cấp việc truy nhập dữ liệu qui mô toàn công ti, và mạng truyền thông toàn công ti.

Tổ chức hiện nay phụ thuộc chủ yếu vào hệ thống và không thể tồn tại được ngay cả với những ngắt quãng bất thường. Trong những tổ chức lớn toàn bộ luồng tiền mặt đều được nối với hệ thống tin. Người thiết kế hệ thống đã bắt đầu xây dựng những khả năng dư thừa để tính tới tình huống mới này.

Song song với việc bành trướng chức năng của hệ thống tin, những thay đổi xuất hiện trong cấu hình tổ chức và kĩ thuật của hệ thống đã đem năng lực tính toán và dữ liệu đến gần người sử dụng cuối cùng hơn. Các "máy kế toán điện tử" cô lập với những chức năng giới hạn trong những năm 1950 đã phải nhường bước cho các máy tính lớn tập trung phục vụ cho tổng hành dinh công ti và một vài trạm từ xa trong những năm 1960. Trong những năm 1970, các máy tính mini cỡ trung bình, đặt trong các công sở hay phân xưởng riêng biệt của tổ chức, đã được nối mạng với các máy tính tập trung lớn. Máy vi tính để bàn ban đầu được dùng độc lập và rồi được nối lại với máy tính mini và máy tính lớn trong những năm 1980.

Trong những năm 1990 đã nổi lên kiến trúc cho một tổ chức được nối mạng hoàn toàn. Trong kiến trúc mới này, máy tính điều phối luồng thông tin qua các máy tính để bàn; giữa máy để bàn, máy tính mini và máy tính lớn; và có lẽ còn giữa hàng trăm mạng cục bộ nhỏ. Hệ thống tin đã trở thành các công cụ tích hợp, trực tuyến, tương tác dùng trong các công việc từng phút và ra quyết định của tổ chức lớn.

Vị trí và vai trò của các chuyên gia hệ thống tin cũng đã tiến hoá qua thời gian. Đơn vị hay chức năng tổ chức chính thức mới nổi lên đó được gọi là **bộ phận hệ thống tin**. Trong những năm đầu, nhóm hệ thống tin hầu như chỉ bao gồm những **người lập trình**, các chuyên gia kĩ thuật được đào tạo cẩn thận, người viết các lệnh phần mềm cho máy tính. Ngày nay một thành phần đang ngày càng tăng trong số nhân viên hệ thống tin là những **người phân tích hệ thống**, người thiết lập nên mối quan hệ giữa nhóm hệ thống tin và phần còn lại của tổ chức. Chính công việc của người phân

tích hệ thống là dịch các vấn đề và yêu cầu nghiệp vụ thành yêu cầu và hệ thống thông tin. **Những người quản lí hệ thống tin** là những người lãnh đạo các nhóm người lập trình và phân tích; những người quản lí dự án; những người quản lí tiện nghi vật lí; người quản lí viễn thông; trưởng các nhóm tự động hoá văn phòng; và cuối cùng, những người quản lí việc vận hành máy tính và nhân viên vào dữ liệu. **Người dùng cuối cùng** là đại diện cho các phòng ban bên ngoài nhóm hệ thống tin, những người sử dụng các ứng dụng do nhóm hệ thống tin làm ra. Những người sử dụng này đóng vai trò ngày một lớn trong việc thiết kế và phát triển hệ thống tin.

Kích cỡ của bộ phận hệ thống tin có thể thay đổi rất lớn, tùy theo vai trò của hệ thống tin trong tổ chức và tùy theo kích cỡ của tổ chức. Ngày nay nhóm hệ thống tin thường hoạt động như những tác nhân gây thay đổi mạnh mẽ trong tổ chức, gợi ý những chiến lược kinh doanh mới và những sản phẩm dựa trên thông tin mới, và phối hợp cả việc phát triển của công nghệ lẫn những thay đổi có kế hoạch trong tổ chức.

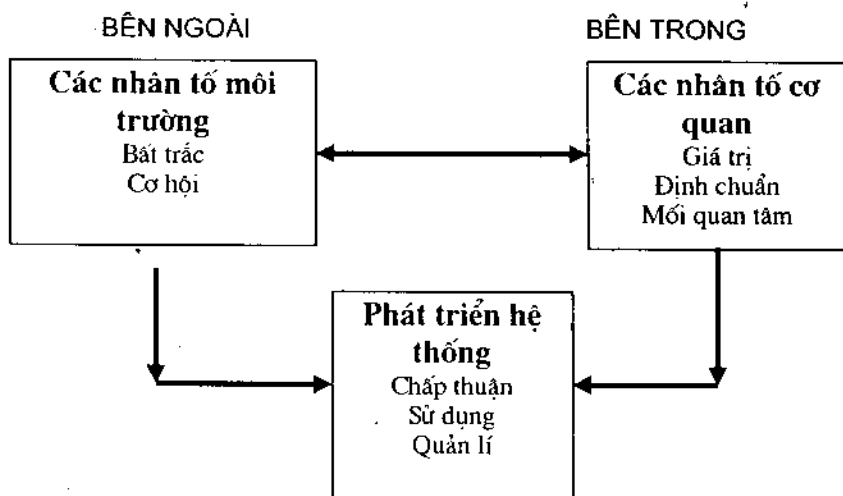
Tại sao tổ chức cần xây dựng hệ thống tin

Câu trả lời truyền thống cho câu hỏi, "Tại sao tổ chức lại chấp nhận hệ thống tin?" có thể dường như rất đơn giản. Hiển nhiên tổ chức chấp nhận hệ thống tin để trở nên hiệu quả hơn, để tiết kiệm tiền bạc và để giảm bớt lực lượng lao động. Mặc dầu câu trả lời này nói chung là đúng trong quá khứ, nó không còn diễn tả được những lí do chủ yếu cho việc chấp nhận hệ thống hiện tại.

Hệ thống tin đã trở thành có tầm quan trọng sống còn đơn thuần chỉ để cho tổ chức trụ lại trong kinh doanh. Những cải tiến trong việc ra quyết định (tốc độ, độ chính xác, toàn diện), phục vụ khách hàng tốt hơn và đáp ứng sự trông đợi của khách, điều phối các nhóm khác nhau trong một tổ chức, hoàn thành các qui tắc báo cáo của chính phủ và tiến hành kiểm soát chặt hơn về nhân sự và

chỉ tiêu đã trở thành những lí do quan trọng cho việc xây dựng hệ thống.

Hình 1.10 sau đây minh hoạ cho mô hình của tiến trình phát triển hệ thống bao gồm nhiều xem xét khác ngoài vấn đề kinh tế. Mô hình này chia sự giải thích cho lí do tại sao tổ chức lại chấp nhận hệ thống thành hai nhóm: *các nhân tố môi trường ngoài* và *các nhân tố nội bộ cơ quan*.



Hình 1.10 Các nhân tố tác động lên tổ chức.

Nhân tố môi trường là các nhân tố bên ngoài tổ chức nhưng ảnh hưởng tới việc chấp nhận và thiết kế hệ thống tin. Một số nhân tố môi trường ngoài đang làm tăng chi phí về lao động hay các tài nguyên khác, hành động cạnh tranh của các tổ chức khác và những thay đổi trong các qui định của chính phủ. Nói chung có thể coi những nhân tố này như những *ràng buộc môi trường*. Đồng thời môi trường cũng cung cấp cho tổ chức những *cơ hội*: công nghệ mới, nguồn vốn mới, sự phát triển của các tiến trình sản xuất mới,

sự chết của các đối thủ cạnh tranh hay chương trình mới của chính phủ làm tăng thêm nhu cầu về loại sản phẩm nào đó.

Nhân tố cơ quan là những nhân tố bên trong của tổ chức có ảnh hưởng tới việc chấp nhận và thiết kế hệ thống tin. Chúng ảnh hưởng tới các giá trị, các định chuẩn và mối quan tâm sống còn vẫn chi phối các vấn đề có tầm quan trọng chiến lược cho tổ chức. Chẳng hạn, cấp lãnh đạo cao nhất của công ti có thể quyết định cần thực hiện việc kiểm soát chặt hơn tiến trình quản lí kho và do đó quyết định xây dựng hệ thống thông tin vật tư.

Hệ thống tin ảnh hưởng tới tổ chức

Sau khi đã mô tả tiến hoá của hệ thống tin và cách hệ thống có thể được tổ chức hình thành bây giờ ta xem xét câu hỏi: Hệ thống tin ảnh hưởng tới tổ chức như thế nào? Để trả lời câu hỏi này, chúng ta cần thu tóm một khối lượng lớn các nghiên cứu và lí thuyết.

LÍ THUYẾT KINH TẾ

Theo quan điểm kinh tế, công nghệ hệ thống tin có thể được xem xét như một nhân tố sản xuất có thể thay thế tự do cho vốn và lao động. Khi chi phí cho công nghệ hệ thống tin giảm xuống, thì nó sẽ được dùng thay thế cho lao động mà về mặt lịch sử luôn có chi phí tăng lên. Qua thời gian, càng ngày càng ít vốn và lao động cần cho một cái vào đã cho. Do đó, công nghệ thông tin phải nghiêng về số các nhà quản lí cấp trung và các thư kí khi công nghệ thông tin thay thế cho lao động của họ.

Công nghệ thông tin cũng giúp cho công ti thu nhỏ kích cỡ vì nó có thể làm giảm chi phí giao tác - chi phí công ti phải gánh chịu khi mua trên thị trường những cái mà nó không tự làm ra được. Về mặt truyền thống, công ti phải giảm chi phí giao tác cứ ngày một

lớn hơn: thuê nhiều nhân viên hơn, tích hợp chiều đứng, mua nhà cung cấp và người phân phối riêng, mua các công ti nhỏ hơn. Công nghệ thông tin, bằng cách giúp cho công ti hạ thấp chi phí về phần tham gia thị trường (chi phí giao tác) làm cho nó hợp đồng với các nhà cung cấp ngoài thay vì dùng tài nguyên nội bộ để phát triển. Kết quả là kích cỡ của công ti (đo theo số nhân viên) có thể giữ không đổi hay thậm chí chỉ cần hợp đồng với họ để làm tăng lợi nhuận. Cuối cùng tổ chức thu nhỏ kích cỡ bởi vì công nghệ thông tin làm giảm chi phí quản lí chung, hạ thấp chi phí mua và phân tích thông tin cần để điều hành và giám sát nhân viên.

LÍ THUYẾT HÀNH VI

Trong khi lí thuyết kinh tế cổ giải thích phần lớn các công ti hoạt động trong thị trường thì lí thuyết hành vi dựa trên xã hội, tâm lí và khoa học chính trị lại có ích nhiều hơn trong việc mô tả hành vi của riêng từng công ti. Nghiên cứu hành vi tìm ra rằng hầu như không có bằng chứng là hệ thông tin tự động làm biến đổi tổ chức. mặc dầu hệ thống có thể là công cụ để thực hiện mục tiêu này một khi lãnh đạo cao cấp quyết định theo đuổi mục đích này. Thay vào đó các nhà nghiên cứu đã quan sát mối quan hệ bản chất trong đó tổ chức và công nghệ thông tin ảnh hưởng tương hỗ lẫn nhau.

Các nhà nghiên cứu hành vi đã lí thuyết hoá rằng công nghệ thông tin có thể làm thay đổi cấp bậc ra quyết định của tổ chức bằng cách hạ thấp chi phí thu nhận thông tin và phổ biến rộng rãi thông tin. Công nghệ thông tin có thể đưa thông tin trực tiếp từ đơn vị tác nghiệp tới nhà quản lí cấp cao, do đó khử bỏ những người quản lí trung gian và các nhân viên thư kí giúp việc. Công nghệ thông tin có thể cho phép nhà quản lí cấp cao hợp đồng với các đơn vị tác nghiệp cấp dưới một cách trực tiếp thông qua việc dùng đường viễn thông và máy tính đã nối mạng, xoá bỏ các cấp quản lí trung gian. Một phương án khác, công nghệ thông tin có thể đóng góp trực tiếp thông tin cho công nhân cấp dưới, làm cho họ có thể ra quyết định riêng dựa trên tri thức và thông tin của mình mà

không cần có sự can thiệp của cấp quản lí. Các nghiên cứu khác gợi ý rằng việc tin học hoá đem lại nhiều thông tin cho cấp quản lí trung gian, làm mạnh cho họ để đưa ra những quyết định quan trọng hơn trong quá khứ, do đó làm giảm nhu cầu dùng số lớn nhân viên mức dưới.

Trong xã hội hậu công nghiệp, có đầu ra kinh tế chủ yếu là dịch vụ và thông tin thì những người có chức trách phải dựa nhiều vào tri thức và tài năng hơn là vị trí chính thức. Hình dáng của tổ chức nên "phẳng ra", vì các công nhân chuyên nghiệp có khuynh hướng tự quản lí; còn việc ra quyết định trở nên phi tập trung hơn khi tri thức và thông tin trở thành phổ biến rộng rãi hơn. Công nghệ thông tin nên dẫn tới các tổ chức có kết mạng theo kiểu "lực lượng đặc nhiệm" trong đó các nhóm chuyên môn lại gần nhau - mặt đối mặt hay qua trao đổi điện tử - để trong một thời kì ngắn hoàn thành một nhiệm vụ xác định (như thiết kế xe ô tô mới); một khi nhiệm vụ đã hoàn thành thì các cá nhân lại tham dự vào các lực lượng đặc nhiệm khác.

Các nhà tư vấn và các tạp chí kinh doanh phổ thông đã gợi ý rằng tổ chức sẽ trở nên phẳng hơn và trải theo chiều ngang hơn bằng cách định dạng lại chính chúng xung quanh tiến trình kinh doanh thay vì là các phòng ban chức năng rời rạc, truyền thống. Hệ thông tin có thể giúp cho tổ chức đạt tới tính hiệu quả lớn bằng cách tự động hoá những thành phần của các tiến trình này hay bằng cách giúp cho các tổ chức suy nghĩ lại và vạch lại các tiến trình này.

Các cách tiếp cận hành vi khác coi hệ thông tin như kết quả của cạnh tranh chính trị giữa các nhóm con trong tổ chức để ảnh hưởng tới chính trị, thủ tục và tài nguyên của tổ chức. Hệ thông tin tất yếu trở thành gắn bó với chính trị của tổ chức bởi vì chúng ảnh hưởng tới việc truy nhập vào một tài nguyên chủ chốt - nghĩa là thông tin. Hệ thông tin có thể ảnh hưởng tới việc ai làm gì cho ai, ở đâu, khi nào và như thế nào trong tổ chức.

Bởi vì hệ thông tin có tiềm năng làm thay đổi cấu trúc, văn hoá, chính trị và công việc của tổ chức nên thường có sự chống cự đáng kể khi nó được đưa vào.

Ảnh hưởng đối với thiết kế và hiểu biết về hệ thông tin

Mối quan hệ giữa tổ chức và hệ thông tin có nhiều vấn đề hơn là chỉ có mua và cài đặt công nghệ. Việc cải tiến tổ chức bao giờ cũng khó khăn và phức tạp và thường dẫn tới thất bại hệ thống. Các nhà quan sát hệ thống có kinh nghiệm và các nhà quản lý tiếp cận tới sự thay đổi hệ thống rất thận trọng. Để thu được ích lợi của công nghệ, những đổi mới của tổ chức - thay đổi về văn hoá, giá trị, chuẩn và dần trải quyền lợi nhóm - phải được quản lý với nhiều nỗ lực và việc lập kế hoạch cũng như thay đổi về công nghệ.

Không có công thức nào tính tới được những nhân tố tổ chức này. Là người quản lý hay người thiết kế, bạn phải xây dựng một bản đối chiếu những nhân tố cần xét tới trong kế hoạch hệ thống của mình. Theo kinh nghiệm chúng tôi, các nhân tố tổ chức trung tâm sắp theo trật tự đại thể là như sau:

- *Môi trường mà tổ chức vận hành trong đó.*
- *Cấu trúc của tổ chức: cấp bậc, chuyên dụng, thủ tục tác nghiệp chuẩn.*
- *Văn hoá và chính trị trong tổ chức.*
- *Kiểu tổ chức.*
- *Bản chất và phong cách lãnh đạo.*
- *Sự hỗ trợ và hiểu biết của cấp lãnh đạo cao nhất.*
- *Mức của tổ chức mà hệ thống nằm tại đó.*
- *Các nhóm quyền lợi chính bị ảnh hưởng bởi hệ thống.*
- *Các loại nhiệm vụ và quyết định mà hệ thông tin được thiết kế trợ giúp cho.*

- *Tình cảm và thái độ* của nhân viên trong tổ chức, những người sẽ dùng hệ thống tin.
- *Lịch sử của tổ chức*: đầu tư quá khứ trong công nghệ thông tin, kỹ năng hiện có, chương trình quan trọng và nhân lực.

Quan hệ mật thiết với thiết kế hệ thống

Đâu là mối quan hệ mật thiết với việc dùng hệ thống tin của tổ chức? Trong khi nhà quản lý thực sự có dùng hệ thống tin chính thức của tổ chức để lập kế hoạch, tổ chức và điều phối thì những hệ thống này có thể còn bị giới hạn trong ảnh hưởng của chúng tới các nhà quản lý hơn là người ta vẫn tưởng cho tới giờ. Các nhà quản lý đánh giá cao các nguồn thông tin miệng, thời sự, mềm dẻo. Họ muốn dùng các hệ thống tin cho các nhiệm vụ sống còn (đối với công việc của họ) như một sự trao đổi giữa con người, đặt ra và thực thi chương trình nghị sự cá nhân, và thiết lập một mạng toàn tổ chức. Các hệ thống tin cơ sở (ít chính thức) có thể được xây dựng nhanh chóng, dùng nhiều thông tin hiện thời và mới nhất, và có thể được điều chỉnh theo các tình huống duy nhất của nhóm các nhà quản lý sẽ được các nhà quản lý hiện đại đánh giá cao.

Những người thiết kế và xây dựng hệ thống phải đánh giá được tầm quan trọng của việc tạo ra các hệ thống có thể liên lạc với các nguồn thông tin khác bên ngoài tổ chức và cung cấp những phương tiện có hiệu quả để liên lạc giữa các nhà quản lý và nhân viên trong tổ chức.

Bảng 1.2, dựa trên phân loại vai trò của Mintzberg là một cái nhìn vào chỗ hệ thống có thể giúp cho các nhà quản lý và chỗ chúng không thể giúp ích gì được. Bảng này chỉ ra rằng hệ thống tin còn chưa là một đóng góp thật lớn cho nhiều lĩnh vực của quản lý. Những lĩnh vực này chắc chắn đưa ra những cơ hội lớn cho hệ thống tương lai và người thiết kế hệ thống.

Bảng 1.2 VAI TRÒ QUẢN LÝ VÀ HỆ THỐNG TIN HỖ TRỢ

Vai trò	Hành vi	Hệ thống hỗ trợ
Vai trò liên con người		Không tồn tại
Bung xung	liên con	Không tồn tại
Lãnh đạo	người	Hệ thống liên lạc điện tử
Liên hệ		
Vai trò thông tin		hệ thông tin quản lý
Trung tâm thần kinh		Thư tín, hệ văn phòng
Người phát tán	xử lý	Trạm làm việc văn phòng
Người phát ngôn	thông tin	
Vai trò ra quyết định		Không tồn tại
Doanh nhân		Không tồn tại
người xử lý rắc rối	ra quyết	Hệ trợ giúp quyết định
người cấp phát tài nguyên	định	Không tồn tại
người thương lượng		

1.6.1 ĐỊNH NGHĨA

Về mặt kỹ thuật, HTT có thể được định nghĩa như một tập hợp của nhiều thành phần liên hệ nhau có chức năng thu thập, quản lý, xử lý, lưu trữ, và phân phối thông tin để hỗ trợ cho việc làm quyết định và điều hành trong một tổ chức. Bên cạnh việc hỗ trợ làm quyết định, phối hợp, và điều hành, HTT còn có thể sản xuất ra các thông tin “chiến lược” để giúp những người quản lý và lao động phân tích vấn đề, hình dung ra các đối tượng phức tạp, và làm ra các sản phẩm mới.

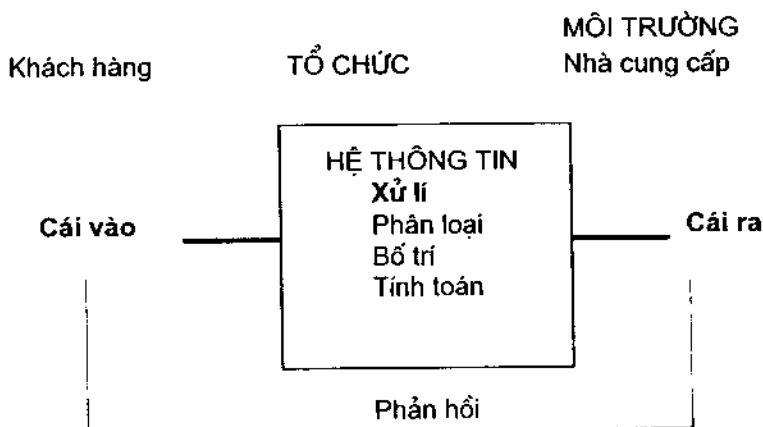
Hệ thống tin có thể chứa thông tin về những người, nơi chốn và những vật quan trọng bên trong tổ chức hay trong môi trường xung quanh tổ chức.

Nói như thế, ta thấy ngay rằng một HTT có thể được thực hiện bằng các công nghệ thủ công như giấy và bút (mà ở một chừng mực nào đó ta có thể tìm gặp ở mọi nơi), bộ xử lý cơ bản là bộ óc con người; hoặc nó được thực hiện bằng các công cụ hiện đại của CNTT và được gọi bằng thuật ngữ Hệ thống tin dựa trên máy tính.

Về mặt kinh doanh và quản lý, HTT có vai trò sâu rộng hơn là những mớ máy móc xử lý thông tin nằm giữa đầu vào và đầu ra. HTT là một giải pháp có tính tổ chức và quản lý để đáp ứng lại một thử thách mà môi trường kinh doanh đặt ra cho doanh nghiệp. Vì lẽ đó, người làm thiết kế HTTQL giỏi trước hết phải thấu đáo về môi trường, cấu trúc, chức năng và cả chính trị-văn hóa của các tổ chức, cũng như vai trò của quản lý và làm quyết định về quản lý. Việc thấu hiểu khả năng và các cơ hội ứng dụng công nghệ hiện đại cho phù hợp với yêu cầu tìm ra giải pháp tốt nhất là một loại yêu cầu khác nữa. Nói nôm na, người thợ giỏi sẽ giỏi hơn nữa nếu biết lựa chọn và sử dụng công cụ trong công tác của mình.

Trong cách mô tả này, chỉ có mặt tổ chức và sử dụng của thông tin được nhấn mạnh, cái nền công nghệ của HTT, mặc dù có thể bao gồm cả một tập thể chuyên môn quan trọng trong xí nghiệp phụ trách, và là một tài sản lớn lao và cực kỳ quan trọng của xí nghiệp, sẽ chỉ đóng vai trò hỗ trợ của nó. Việc phân biệt giữa tri thức nghiệp vụ quản lý, thể hiện qua mô hình thông tin, và tri thức về các công nghệ có vai trò công cụ, giúp nhấn mạnh sự quan trọng của thông tin (nghiệp vụ) trong mục tiêu phát triển CNTT để phục vụ kinh tế xã hội. Trong một phần sau, ý niệm này sẽ được trở lại để triển khai thêm về vấn đề xây dựng các cơ sở thông tin, cũng như các yêu cầu về giáo dục đào tạo ra những con người giúp cho cả bộ máy thông tin làm việc và phát huy chức năng.

Ba hoạt động trong một hệ thống tin tạo ra thông tin mà tổ chức cần để làm quyết định, điều khiển tác nghiệp, phân tích vấn đề và tạo ra các sản phẩm và dịch vụ mới. Ba hoạt động đó là đưa vào, xử lý và đưa ra. **Đưa vào** thu thập hay thu thập dữ liệu thô lấy từ bên trong tổ chức hay từ môi trường bên ngoài. **Xử lý** chuyển đổi cái vào thô này thành một dạng có nghĩa nhiều hơn. **Đưa ra** chuyển thông tin đã xử lý cho những người hay hoạt động sẽ dùng nó. Hệ thống tin cũng cần **phản hồi**, tức là cái ra được đưa trở lại cho thành viên thích hợp của tổ chức để giúp họ đánh giá hay sửa chữa giai đoạn vào (Hình 1.11).



Hình 1.11. Các chức năng của một hệ thống tin.

SỰ KHÁC BIỆT GIỮA MÁY TÍNH VÀ HỆ THỐNG TIN

Mặc dầu hệ thống tin dựa trên máy tính dùng công nghệ máy tính để xử lý dữ liệu thô thành thông tin có nghĩa, có sự khác biệt sắc nét giữa máy tính và chương trình máy tính một bên, và hệ thống tin bên kia. Máy tính điện tử và các chương trình phần mềm có liên quan là nền tảng kỹ thuật, công cụ và vật liệu, của hệ thống tin hiện đại. Máy tính cung cấp phương tiện để lưu trữ và xử lý

thông tin. Chương trình máy tính, hay phần mềm, là tập các lệnh điều hành, chỉ huy và điều khiển việc xử lý máy tính. Hiểu biết về cách máy tính và chương trình máy tính làm việc là quan trọng trong việc vạch ra giải pháp cho các vấn đề tổ chức, nhưng lí do dùng chúng lại bắt nguồn từ hệ thống tin có máy tính.

1.6.2 CÁC LOẠI HỆ THÔNG TIN

Để phục vụ cho mọi hoạt động có tính hướng mục tiêu của con người nói chung, người ta cần phải có đầy đủ thông tin. Thông tin ở đây mang nghĩa nhưng tri thức, hiểu biết của con người đối với các vấn đề thực tế, phản ánh tình trạng thực tại của sự vật và các cách thức, phương pháp tác động vào thực tế để con người đạt tới mục tiêu của mình.

Hình 1.12 sau đây minh họa cho một cách mô tả các loại hệ thống có trong tổ chức sản xuất. Trong minh họa này tổ chức được chia thành các mức chiến lược, quản lí, tri thức và tác nghiệp và rồi được chia thêm thành các miền chức năng như Tài nguyên và môi trường, Con người, Vốn, Công nghệ, Sản xuất, Kinh doanh. Các hệ thống được xây dựng để phục vụ cho các mối quan tâm khác nhau của tổ chức.

Về mặt tổ chức và cấu trúc, một cách tổng quát, ta có thể hình dung một HTT, tùy chức năng, bao gồm một số hoặc tất cả các cấp thông tin sau đây:

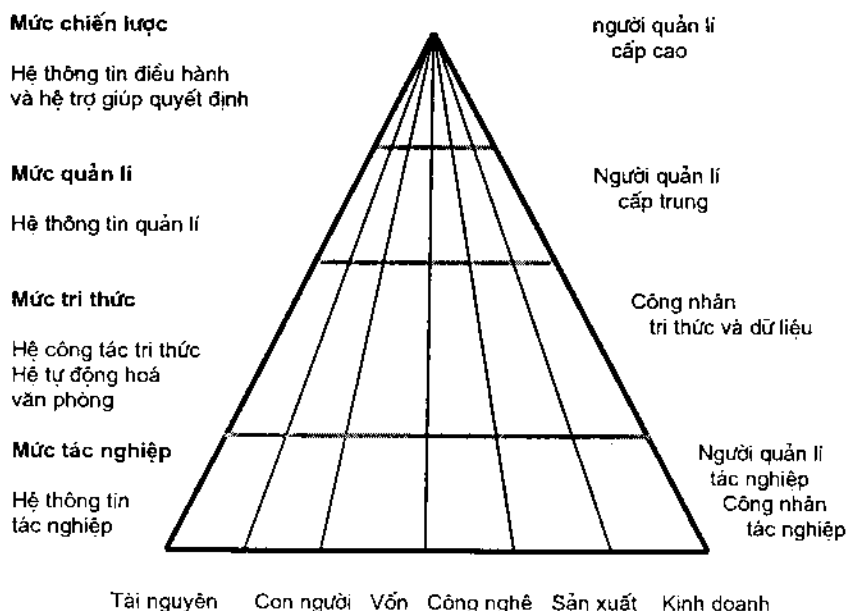
- **cấp công tác hay tác nghiệp** : dùng theo dõi các hoạt động và các giao tác sơ cấp của tổ chức;
- **cấp tri thức** : hỗ trợ cho những lao động tri thức và dữ liệu của tổ chức;
- **cấp quản lí**: hỗ trợ việc theo dõi, kiểm tra, quyết định, và các công việc quản lí dành cho lớp quản lí trung cấp;

- **cấp chiến lược:** hỗ trợ công tác hoạch định dài hạn dành cho lớp quản lý cao cấp.

Nhìn hàng ngang, mỗi cấp thông tin nói trên đều phải bao gồm tất cả các khu vực hoạt động của tổ chức, thí dụ, đối với một doanh nghiệp, sẽ có những khu vực sau đây: Tài nguyên và môi trường, Con người, Vốn, Công nghệ, Sản xuất, Kinh doanh.

LOẠI HỆ THỐNG TIN

NHÓM ĐƯỢC PHỤC VỤ



Hình 1.12. Các kiểu hệ thống tin. Tổ chức và hệ thống tin có thể được chia thành các mức chiến lược, quản lý, tri thức và tác nghiệp. Chúng còn có thể được chia thêm thành sáu lĩnh vực chức năng : Tài nguyên và môi trường, Con người, Vốn, Công nghệ, Sản xuất, Kinh doanh.

Hệ thống mức tác nghiệp hỗ trợ cho những người quản lý tác nghiệp bằng cách lưu giữ dấu vết của các hoạt động sơ cấp và giao tác của tổ chức như bán hàng, hoá đơn, tiền mặt, tiền lương, quyết định tín dụng và luồng vật tư trong nhà máy. Mục đích chính của hệ thống tại mức này là trả lời cho các câu hỏi thường lệ và theo dõi luồng giao tác qua tổ chức. Còn bao nhiêu trong kho? Khối lượng trả lương tháng này là bao nhiêu? Để trả lời cho câu hỏi này, thông tin nói chung phải có sẵn, hiện thời và chính xác.

Hệ thống mức tri thức hỗ trợ cho các công nhân tri thức và dữ liệu trong một tổ chức. Mục đích của hệ thống mức tri thức là giúp cho công ti kinh doanh tích hợp tri thức mới vào trong kinh doanh và giúp cho tổ chức kiểm soát được luồng công việc giấy tờ. Hệ thống mức tri thức, đặc biệt dưới dạng trạm làm việc và hệ văn phòng là những ứng dụng phát triển nhanh nhất trong kinh doanh ngày nay.

Hệ thống mức quản lý được thiết kế phục vụ cho các hoạt động điều hành, kiểm soát, ra quyết định và hành chính của người quản lý cấp trung. Câu hỏi chính mà hệ thống này đề cập đến là: mọi việc có vận hành tốt không? Các hệ thống này so sánh kết quả hàng ngày với kết quả tháng trước hay năm trước. Hệ thống mức quản lý cung cấp điển hình các báo cáo theo kì hạn chứ không phải là thông tin hay tác nghiệp hiện thời.

Một số hệ thống mức quản lý hỗ trợ cho việc ra quyết định bất thường. Chúng có khuynh hướng ít tập trung vào các quyết định có cấu trúc mà các yêu cầu thông tin thường không rõ. Những hệ thống này thường trả lời cho câu hỏi "cái gì xảy ra nếu": Cái gì sẽ ảnh hưởng tới lịch sản xuất nếu chúng ta tăng gấp đôi lương tháng mười hai? Điều gì sẽ xảy ra cho lợi tức trên đầu tư nếu lịch nhà máy bị chậm sáu tháng? Trả lời cho những câu hỏi này thường đòi hỏi dữ liệu mới từ bên ngoài tổ chức, cũng như dữ liệu từ bên trong mà không thể rút ra được từ các hệ thống mức tác nghiệp.

Hệ thống mức chiến lược giúp cho giới quản lí cấp cao giải quyết và đề cập đến các vấn đề chiến lược và khuynh hướng dài hạn, cả trong công ti và trong môi trường bên ngoài. Mối quan tâm chính của họ là so sánh những thay đổi trong môi trường ngoài với khả năng hiện tại của tổ chức. Mức độ nhân viên trong năm năm nữa sẽ thế nào? Khuynh hướng chi phí công nghiệp dài hạn là gì, và công ti sẽ giữ vị trí nào? Chúng ta cần đưa ra sản phẩm nào trong năm năm nữa?

Hệ thông tin cũng có thể bị khác biệt nhau bởi chuyên dụng chức năng. Các chức năng tổ chức chính, như Tài nguyên và môi trường, Con người, Vốn, Công nghệ, Sản xuất, Kinh doanh, đều có hệ thống tin riêng của mình. Trong các tổ chức lớn, các chức năng con của từng chức năng chính này cũng lại có hệ thống tin riêng của nó. Chẳng hạn chức năng sản xuất có thể có hệ thống tin để quản lí kho, kiểm soát tiến trình, bảo trì nhà máy, kĩ nghệ với sự trợ giúp của máy tính và lập kế hoạch yêu cầu vật tư.

Một tổ chức điển hình có các hệ thống mức tác nghiệp, quản lí, tri thức và chiến lược cho từng lĩnh vực chức năng. Chẳng hạn, chức năng bán hàng nói chung có hệ thống bán hàng ở mức tác nghiệp để ghi lại con số bán hàng ngày và xử lí đơn hàng. Một hệ thống mức tri thức thiết kế ra các giới thiệu hàng mẫu cho sản phẩm công ti. Hệ thống mức quản lí theo dõi con số bán hàng tháng theo vùng lãnh thổ bán hàng và báo cáo theo vùng lãnh thổ những số bán tăng cao hay xuống thấp một mức nào đó. Hệ thống dự báo xu hướng bán hàng kì năm năm phục vụ cho mức chiến lược.

Không một hệ thống riêng biệt nào cung cấp được mọi thông tin mà tổ chức cần. Tổ chức có nhiều hệ thống tin phục vụ cho các chức năng và mức tổ chức khác nhau. Do vậy, các hệ thống điển hình có trong các tổ chức thường được thiết kế để trợ giúp cho các nhân viên hay người quản lí tại mỗi mức và theo các chức năng Tài nguyên và môi trường, Con người, Vốn, Công nghệ, Sản xuất, Kinh doanh. Trong mục này chúng tôi mô tả một phân loại đặc biệt về

các hệ thống phục vụ cho từng mức tổ chức. Các thí dụ chúng tôi nêu cho từng loại sẽ minh hoạ cho giá trị mà mỗi loại đó đóng góp cho tổ chức.

Phân loại hệ thống					
Hệ thống mức chiến lược					
Hệ hỗ trợ điều hành (ESS)	Dự kiến khuynh hướng 5 năm	Kế hoạch tác nghiệp 5 năm	Dự kiến ngân sách 5 năm	Lập kế hoạch lợi nhuận	Lập kế hoạch nhân lực
Hệ thống mức quản lí					
Hệ thống tin quản lí (MIS)	Quản lí bán	Quản lí kho	Lập ngân sách năm	Phân tích đầu tư vốn	Phân tích bố trí
Hệ trợ giúp quyết định (DSS)	Phân tích vùng bán hàng	Lập lịch sản xuất	Phân tích chi phí	Phân tích giá cả/lợi nhuận	Phân tích giá hợp đồng
Hệ thống mức tri thức					
Hệ công tác tri thức (KWS)	Trạm làm việc kĩ nghệ		Trạm làm việc đồ hoạ		Trạm làm việc quản lí
Hệ tự động hoá văn phòng (OAS)	Xử lí văn bản		Lưu trữ ảnh		Lịch văn phòng
Hệ thống mức tác nghiệp					
Hệ xử lí giao tác (TPS)	Theo dõi đơn hàng	Điều khiển máy móc	Tính lương	Kiểm toán	Đền bù
	Xử lí đơn hàng	Lập lịch nhà máy	Kế toán chi	Báo thuế	Huấn luyện, phát triển
		Kiểm soát luân chuyển vật tư	Kế toán thu	Quản lí tiền mặt	Quản lí nhân sự

Hình 1.13 Sáu kiểu hệ thống tin chính cần cho bốn mức của một tổ chức.

Sáu kiểu hệ thống chính

Hình 1.13 chỉ ra các kiểu hệ thống tin đặc biệt tương ứng với từng mức tổ chức. Tổ chức có hệ thống hỗ trợ điều hành (ESS) tại mức chiến lược; hệ thống tin quản lí (MIS) và hệ trợ giúp quyết định (DSS) tại mức quản lí; hệ công tác tri thức (KWS) và hệ tự động hoá văn phòng (OAS) ở mức tri thức; và hệ xử lí giao tác (TPS) ở mức tác nghiệp. Hệ thống tại mỗi mức đến lượt nó lại được chuyên môn hoá để phục vụ cho từng lĩnh vực chức năng chính.

Bảng 1.3 Các đặc trưng của hệ xử lí thông tin

Kiểu hệ thống	Thông tin vào	Xử lí	Thông tin ra	Người dùng
ESS	Dữ liệu thu thập; bên ngoài, bên trong	Đồ hoạ; mô phỏng; tương tác	Dự phóng; trả lời câu hỏi	Người quản lí cấp cao
DSS	Dữ liệu khối lượng nhỏ; mô hình phân tích	Tương tác, mô phỏng	Báo cáo đặc biệt; phân tích quyết định; trả lời câu hỏi	Chuyên gia; nhân viên quản lí
MIS	Dữ liệu giao tác tóm tắt; dữ liệu khối lượng lớn;	Báo cáo thường kì; mô hình đơn giản; phân tích mức thấp	Báo cáo tóm tắt và ngoại lệ	Người quản lí cấp trung
KWS	Mô hình đơn giản	Mô hình hoá; mô phỏng	Mô hình; đồ hoạ	Chuyên gia; nhân viên kĩ thuật
OAS	Đặc tả thiết kế; cơ sở tri thức; Tư liệu; lịch biểu	Quản lí tài liệu; lập lịch; liên lạc	Tài liệu; lịch biểu; thư	Nhân viên thư kí
TPS	Giao tác; sự kiện	Sắp xếp; in ; gộp; cập nhật	Báo cáo chi tiết; danh sách; tóm tắt	Nhân viên vận hành; đốc công

Bảng 1.3 tóm tắt các tính năng của sáu kiểu hệ thống tin. Cần phải chú ý rằng từng loại hệ thống khác nhau có thể có các thành phần được dùng bởi các mức và nhóm tổ chức khác hơn các thể chế chính của chúng. Một thư kí có thể tìm thấy thông tin trong MIS hay một người quản lí cấp trung có thể cần trích dữ liệu từ TPS.

Hệ xử lí giao tác

Hệ thống xử lí giao tác (TPS) là hệ thống nghiệp vụ cơ bản phục vụ cho mức tác nghiệp của tổ chức. Một hệ thống xử lí giao tác là một hệ thống tin học hoá thực hiện và ghi lại các giao tác thường lệ hàng ngày cần cho việc tiến hành kinh doanh. Thí dụ như vào hoá đơn bán hàng, hệ thống giữ chỗ khách sạn, thông tin khách hàng (với các công ti công cộng), tính lương, lưu trữ hồ sơ nhân viên và vận đơn.

Tại mức tác nghiệp, các nhiệm vụ, tài nguyên và mục tiêu để được xác định trước và có cấu trúc cao độ. Quyết định cấp tín dụng cho khách hàng chẳng hạn là do giám sát viên cấp thấp ấn định theo tiêu chuẩn đã đặt sẵn. Quyết định, theo nghĩa đó, đã được "lập chương trình". Tất cả những điều phải được xác định là liệu khách hàng có đáp ứng đúng tiêu chuẩn đó hay không.

Các ứng dụng hệ xử lí giao tác TPS được xác định như trong Hình 1.14. Hình này minh hoạ cho năm loại chức năng của TPS: bán hàng/thị trường, chế tạo/sản xuất, tài chính/kế toán, nhân lực và các kiểu TPS khác duy nhất cho một ngành công nghiệp đặc thù.

Mọi tổ chức đều có năm loại TPS (cho dù là hệ thống thủ công). Thật khó mà hình dung được một tổ chức hiện đại lại không có hệ thống xử lí giao tác. Các hệ thống này thường là trung tâm cho một doanh nghiệp trong những năm 1960 đến độ người ta ước lượng rằng tổ chức có thể chỉ tồn tại được một ngày nếu hệ thống máy tính không vận hành.

Hai tính năng của TPS rất đáng lưu ý. Thứ nhất nhiều TPS trải ngang qua biên giới giữa tổ chức và môi trường. Chúng nối khách hàng với nhà kho, nhà máy và cấp quản lý của công ti. Nếu TPS không làm việc tốt thì tổ chức sẽ không thể nhận được cái vào từ môi trường bên ngoài (đơn hàng) hay không thể bàn giao cái ra (hàng hoá đã lắp ráp). Thứ hai là, TPS là người sản xuất chính ra thông tin cho các kiểu hệ thống khác. Bởi vì TPS giữ dấu vết mối quan hệ với môi trường nên chúng là nơi duy nhất mà người quản lý có thể thu được sự định giá đến từng phút về độ hoàn thiện của tổ chức và các bản ghi lâu dài của sự thực hiện trong quá khứ. TPS có thể được xem như "hệ thống xử lý thông báo của tổ chức", thông báo cho người quản lý về trạng thái sự vận hành bên trong và về mối quan hệ của công ti với môi trường bên ngoài, và hỗ trợ cho các hệ thống tin khác làm thuận tiện cho việc ra quyết định quản lý.

Hệ công tác tri thức và tự động hoá văn phòng

Hệ công tác tri thức (KWS) và tự động hoá văn phòng (OAS) phục vụ cho nhu cầu thông tin ở mức tri thức của tổ chức.

Hệ công tác tri thức giúp đỡ cho các công nhân tri thức, trong khi hệ tự động hoá văn phòng thì chủ yếu giúp cho công nhân dữ liệu (mặc dầu chúng cũng được công nhân tri thức sử dụng rất nhiều) (Hình 1.14).

Nói chung, *công nhân tri thức* là những người có bằng cấp đại học chính thức và thường là thành viên của các chuyên môn đã được thừa nhận, như kỹ sư, bác sĩ, luật sư và nhà khoa học. Công việc của họ chủ yếu là tạo ra thông tin và tri thức mới. Hệ thống công tác tri thức, như các trạm thiết kế kỹ nghệ và khoa học, thúc đẩy việc tạo ra tri thức mới và đảm bảo rằng tri thức và tính chuyên gia kỹ thuật được tích hợp đúng đắn vào trong kinh doanh.

Các kiểu hệ thống xử lý giao tác TPS

	Hệ thống bán hàng/ tiếp thị	Hệ thống chế tạo/ sản xuất	Hệ thống tài chính / kế toán	Hệ thống nhân lực	Các kiểu khác (như cho đại học)
Các chức năng chính của hệ thống	Quản lý bán hàng	Lập lịch	Lập ngân sách	Hồ sơ lý lịch	Nhận vào
	Nghiên cứu thị trường	Mua hàng	Sổ cái	Phúc trợ	Hồ sơ bằng cấp
	Khuyến mại	Vận chuyển/ nhận hàng	Làm hoá đơn	Bù	Hồ sơ giáo trình Sinh viên
	Làm giá Sản phẩm mới	Kĩ nghệ Tác nghiệp	Kế toán chi phí	Quan hệ lao động Huấn luyện	
Hệ thống ứng dụng chính	Hệ thống tin đơn hàng	Hệ thống lập kế hoạch tài nguyên vật tư	Sổ cái Tài khoản thu / chi	Trả lương Hồ sơ lý lịch	Hệ thống đăng kí
	Hệ nghiên cứu thị trường	Hệ thống kiểm soát đơn mua hàng	Ngân sách	Hệ thống phúc lợi Hệ thống đường	Hệ thống ghi sinh viên
	Hệ thống làm giá	Hệ thống kĩ nghệ	Hệ thống quản lý	nghe nghiệp	Hệ thống kiểm soát chương trình học
		Hệ thống kiểm tra chất lượng	quĩ	Hệ thống kế hoạch nhân sự	

Hình 1.14 Những ứng dụng điển hình của TPS.

Công nhân dữ liệu về cơ bản ít có bằng cấp giáo dục cao cấp hay chính thức và có khuynh hướng xử lý chứ không tạo ra thông tin. Họ bao gồm chủ yếu là các thư kí, kế toán viên, nhân viên văn thư hay người quản lý có công việc chính là sử dụng, thao tác hay phát tán thông tin. Hệ tự động hoá văn phòng là ứng dụng công

nghe thông tin được thiết kế để làm tăng tính hiệu suất của công nhân dữ liệu trong văn phòng bằng cách hỗ trợ cho các hoạt động điều phối và truyền thông của văn phòng. Hệ tự động hoá văn phòng phối hợp các công nhân thông tin khác nhau, các đơn vị theo vùng địa lí và các lĩnh vực chức năng. Hệ thống này liên lạc với khách hàng, nhà cung cấp và các tổ chức khác bên ngoài công ti và phục vụ như một ngân hàng cho các luồng thông tin và tri thức.

Hệ thống tự động hoá văn phòng điển hình xử lí và quản lí các tài liệu (qua việc xử lí văn bản, xuất bản tại chỗ, sắp xếp số), lập lịch (qua lịch điện tử) và truyền thông (qua thư điện tử, thư tiếng nói hay họp từ xa). **Xử lí văn bản** nói tới phần mềm và phần cứng để tạo ra, sửa đổi, định dạng và in tài liệu. Hệ xử lí văn bản biểu thị cho một ứng dụng thông thường nhất của công nghệ thông tin đối với công việc văn phòng, một phần bởi vì việc tạo ra các tài liệu chính là việc các văn phòng đều làm. **Xuất bản tại bàn** tạo ra các tài liệu với chất lượng như xuất bản bằng cách tổ hợp kết quả của phần mềm xử lí văn bản với các phần tử thiết kế, đồ hoạ và các tính năng bố trí đặc biệt.

Một trong những trở ngại lớn nhất cho việc tự động hoá văn phòng là sự tràn ngập tài liệu giấy tờ cần phải cất giữ. Hàng trăm tỉ tài liệu hàng năm được các doanh nghiệp tạo ra. Nhiều tài liệu, như đăng kí kết hôn, giấy khai sinh và hợp đồng các loại, đều yêu cầu có chữ kí và sự giữ gìn chữ kí đó trong thời gian dài. Nhiều tài liệu cơ sở khác, như đơn mua hàng, hoá đơn, giấy chấp nhận, giấy rút tiền và những thứ giấy tờ khác, tất cả đều được lưu trữ thủ công trong các tủ hồ sơ. Một giải pháp cho vấn đề quản lí công việc giấy tờ là **xử lí hình ảnh số**, chuyển đổi các tài liệu và hình ảnh thành dạng tin học hoá để cho chúng có thể được lưu trữ và truy nhập trong hệ thống máy tính.

Các công nhân tri thức, người tạo ra và sản xuất tri thức, về truyền thống cũng đã sử dụng công nghệ tự động hoá văn phòng. Tuy nhiên ngày nay họ còn có cả hệ thống và công nghệ về công

tác tri thức để hỗ trợ cho vai trò của họ trong công ti. Các máy tính để bàn mạnh, gọi là **trạm làm việc**, với những khả năng đồ hoạ, phân tích, quản lí tài liệu và truyền thông có thể gán bó thông tin từ các viễn cảnh và nguồn khác nhau cả bên trong lẫn bên ngoài công ti. Trong lĩnh vực kĩ nghệ, hệ thống công tác tri thức có thể dùng những dụng cụ như thế để chạy hàng ngàn tính toán trước khi người thiết kế thấy thoả mãn rằng một bộ phận nào đó là an toàn. Người thiết kế và các chuyên gia phác thảo có thể muốn dùng các trạm làm việc với phần mềm 3 chiều để hình dung một mô hình sản phẩm một cách đầy đủ hơn. Thí dụ về những hệ thống công tác tri thức chủ yếu là **hệ thống tin địa lí (GIS)** và các **hệ thống tự động hoá mô phỏng công tác mỏ**. Hệ thống tin địa lí lưu giữ tất cả các thông tin liên quan tới tình hình tài nguyên khoáng sản và môi trường trên cơ sở thông tin địa lí. Các hệ thống tự động hoá mô phỏng công tác mỏ là những dụng cụ hiện đại giúp đỡ cho các cấp lãnh đạo có cái nhìn trước về những khả năng khai thác mỏ trước khi ra quyết định sản xuất. Đây là hai loại hình hệ thống tin tri thức rất quan trọng cho các cấp quản lí tri thức khi ra những quyết định đầu tư và sản xuất quan trọng.

Hệ thống tin địa lí (GIS) (Bảng 1.4) là tập hợp công nghệ thông tin, dữ liệu và thủ tục để thu thập, ghi nhớ, thao tác, phân tích và trình bày các bản đồ và thông tin mô tả về những tính năng có thể được biểu diễn trên bản đồ. Nói một cách chính xác hơn, GIS giải quyết các ứng dụng của CNTT, nguyên tắc quản trị dữ liệu và lí thuyết tổ chức để đáp ứng các nhu cầu thông tin địa lí. Bốn yếu tố được mô tả trong Bảng 1.4 có quan hệ chặt chẽ với nhau.

Vai trò của các hệ thống công tác tri thức và tự động hoá văn phòng trong công ti không nên bị đánh giá sai lạc. Khi nền kinh tế dịch chuyển từ việc dựa trên hàng hoá chế tạo sang việc tạo ra dịch vụ, tri thức và thông tin thì hiệu suất của từng công ti và toàn bộ nền kinh tế sẽ càng phụ thuộc nhiều vào hệ thống mức tri thức. Đây là một lí do cho các hệ thống mức tri thức đã là lĩnh vực ứng dụng phát triển nhanh nhất trong thập kỉ qua và có thể còn phát triển

trong tương lai. Các hệ thống mức tri thức cũng đã trở nên gắn chặt hơn với các hệ thống khác của công ti.

BẢNG 1.4 **ĐẶC TRƯNG CỦA CÁC HỆ THỐNG TIN ĐỊA LÍ GIS**

1. *Hệ thống tin địa lí GIS* dựa trên nền tảng quan niệm về việc dùng thông tin địa lí để đưa ra một cơ sở chung để tham khảo hay tập trung vào ba phần tử sau đây.
2. *Nguyên lí quản trị dữ liệu.* Cấu trúc logic và quản lí của các CSDL lớn có chứa bản đồ và những dữ liệu khác có thể có liên quan tới mỗi quan tâm về địa lí của tổ chức.
3. *Công nghệ.* Việc tổ hợp có hiệu quả của nhiều thành phần phần cứng và phần mềm khác nhau làm tự động hoá nhiều chức năng xử lí dữ liệu địa lí.
4. *Thiết đặt tổ chức.* Một môi trường quản lí cung cấp các tài nguyên và làm cho những thay đổi có thể được tổ hợp và trong việc sử dụng GIS trong toàn tổ chức.

Hệ thống tin quản lí

Hệ thống tin quản lí (MIS) phục vụ cho mức quản lí của tổ chức, cung cấp cho các nhà quản lí các báo cáo và, trong một số trường hợp, truy nhập trực tuyến vào các bản ghi lịch sử và hiện tại của tổ chức. Một cách điển hình, chúng gần hướng vào các sự kiện bên trong tổ chức chứ không hướng ra môi trường hay bên ngoài. MIS chủ yếu phục vụ cho các chức năng lập kế hoạch, kiểm soát và ra quyết định tại mức quản lí. Nói chung, chúng phụ thuộc vào hệ thống xử lí giao tác nền tảng để lấy dữ liệu.

MIS tổng hợp và báo cáo về các vận hành cơ bản của công ti. Dữ liệu giao tác cơ sở từ TPS được cô đọng lại và thường được trình bày trong những báo cáo dài theo định kì. MIS thường phục vụ cho những người quản lí về các kết quả đáng quan tâm trong tuần, tháng và năm – không phải là các hoạt động thường ngày. MIS đề cập đến các vấn đề có cấu trúc nói chung được biết rõ từ trước. Các

hệ thống này nói chung không mềm dẻo và ít có khả năng phân tích. Bảng 1.5 mô tả các đặc trưng của hệ thống tin quản lý điển hình.

Các MIS mới hơn thì thường linh hoạt hơn và có thể chứa các phần mềm cho phép người quản lý cấu trúc lấy báo cáo của riêng mình và tổ hợp dữ liệu từ các tệp tách biệt và từ TPS.

BẢNG 1.5 ĐẶC TRƯNG CỦA CÁC HỆ THỐNG TIN QUẢN LÝ

1. MIS hỗ trợ cho các quyết định có cấu trúc và nửa cấu trúc ở mức kiểm soát tác nghiệp và quản lý. Tuy nhiên chúng cũng có ích cho các mục đích lập kế hoạch của các nhân viên quản lý cấp cao.
 2. MIS nói chung có khuynh hướng báo cáo và kiểm soát. Chúng được thiết kế để báo cáo về các hoạt động hiện có và do đó giúp cung cấp việc kiểm soát thường ngày về các hoạt động.
 3. MIS dựa trên dữ liệu công ti và luồng dữ liệu hiện có.
 4. MIS ít có khả năng phân tích
 5. MIS nói chung giúp đỡ cho việc ra quyết định bằng cách dùng dữ liệu quá khứ và hiện tại
 6. MIS là tương đối không linh hoạt
 7. MIS hướng nội bộ hơn là hướng ngoại
 8. Các yêu cầu thông tin được xem như ổn định
 9. MIS yêu cầu một tiến trình thiết kế và phân tích dài (độ một hay hai năm).
-

Hệ trợ giúp quyết định

Bất kỳ hệ thống nào hỗ trợ cho một quyết định cũng đều là **hệ thống hỗ trợ quyết định (DSS)**. Giống như MIS, DSS phục vụ cho mức quản lý của tổ chức. Hệ thống tin hỗ trợ quyết định theo rất nhiều cách khác nhau, và DSS là một lớp các hệ thống hỗ trợ cho quyết định theo cách thức duy nhất (ít nhất là khi so sánh với quá

khứ). Bảng 1.6 chỉ ra đặc trưng của các hệ thống hỗ trợ quyết định đương đại.

DSS giúp cho những người quản lí ra quyết định dạng nửa cấu trúc, duy nhất hay thay đổi nhanh chóng và không dễ xác định trước. Theo những cách này chúng khác với các hệ thống MIS và TPS. DSS nhanh nhẹn, tương tác, hướng mô hình và hướng hành động trong khi các hệ thống MIS có khuynh hướng chậm chạp, hướng xử lí theo lô và hướng dữ liệu. DSS phải đáp ứng đủ nhanh để có thể chạy nhiều lần trong ngày tương ứng với những điều kiện thay đổi. Trong khi DSS dùng thông tin nội bộ từ TPS và MIS thì chúng thường lấy thêm thông tin từ các nguồn ngoài, như giá cổ phiếu hiện tại hay giá sản phẩm của các đối thủ cạnh tranh.

BẢNG 1.6 ĐẶC TRƯNG CỦA CÁC HỆ THỐNG HỖ TRỢ QUYẾT ĐỊNH

1. DSS cung cấp cho người dùng câu trả lời linh hoạt, thích ứng và nhanh chóng
2. DSS cho phép người dùng khởi đầu và kiểm soát cái vào và cái ra.
3. DSS vận hành với rất ít hay không cần sự trợ giúp của người lập trình chuyên nghiệp.
4. DSS cung cấp sự hỗ trợ cho các quyết định và vấn đề mà giải pháp không thể được mô tả trước.
5. DSS dùng các dụng cụ mô hình và phân tích tình vi

Rõ ràng, theo thiết kế, DSS có nhiều năng lực phân tích hơn các hệ thống khác; chúng cũng được xây dựng tương minh với rất nhiều mô hình để phân tích dữ liệu. Thứ hai là, DSS được thiết kế sao cho người dùng có thể tự làm việc trực tiếp; các hệ thống này chứa cả giao diện thân thiện người dùng. Yêu cầu này bắt nguồn cả từ mục đích của chúng (để thông báo việc ra quyết định cá nhân bởi các tác nhân chủ chốt) và từ phương pháp thiết kế. Thứ ba là những

hệ thống này là tương tác; người dùng có thể thay đổi giả thiết và đưa vào dữ liệu mới.

Cấp quản lí bậc cao và trung muốn có một hệ thống nhiều tương tác mà tự họ có thể kiểm soát và cho chạy được, với nhiều sự thay đổi trong dữ liệu và mô hình mà họ cần và ít phải phiền tới các nhà chuyên môn hệ thông tin. Giới quản lí cũng cần thông tin ngay tức khắc để đáp ứng cho các cơ hội đầu thầu.

Hệ hỗ trợ điều hành

Những người quản lí cấp cao dùng một loại hệ thông tin gọi là **hệ hỗ trợ điều hành (ESS)** để ra quyết định. ESS phục vụ cho mức chiến lược của tổ chức. Chúng đề cập đến các quyết định phi cấu trúc và tạo ra một môi trường tính toán và truyền thông tổng quát chứ không cung cấp bất kì ứng dụng cố định hay khả năng đặc biệt nào. ESS được thiết kế để tổ hợp dữ liệu về các sự kiện bên ngoài, như các luật thuế mới hay đối thủ cạnh tranh mới, nhưng chúng cũng rút ra những thông tin tóm tắt từ MIS và DSS bên trong. Chúng lọc, nén và theo dõi những dữ liệu chủ chốt, nhấn mạnh vào việc rút ngắn thời gian và nỗ lực cần để thu được thông tin có ích cho điều hành. Mặc dầu chúng có những khả năng phân tích giới hạn, ESS sử dụng hầu hết các phần mềm đồ hoạ cao cấp và có thể trình bày ngay các hình đồ hoạ và dữ liệu từ nhiều nguồn cho các quan chức điều hành cấp cao hay cho phòng họp ban giám đốc.

Không giống như các kiểu hệ thông tin khác, ESS không được thiết kế chủ yếu để giải quyết các vấn đề đặc thù. Thay vì thế, ESS cung cấp khả năng tính toán và truyền thông tổng quát có thể được áp dụng cho hàng loạt vấn đề thay đổi. Trong khi DSS được thiết kế nặng về phân tích thì ESS có khuynh hướng ít dùng mô hình phân tích. Thay vào đó, chúng trao thông tin cho người quản lí theo yêu cầu và trên cơ sở tương tác cao độ.

Các vấn đề mà ESS phải trợ giúp trong việc trả lời bao gồm những vấn đề sau: Chúng ta nên tiến hành các hoạt động kinh

doanh nào? Các đối thủ cạnh tranh đang làm gì? Những thu nhập mới nào sẽ bảo vệ chúng ta khỏi sự chao đảo trong kinh doanh? Chúng ta nên bán những đơn vị nào để nâng cao tiền mặt? Đây là tác động lên lương từ những thay đổi được đề nghị trong khoản tín dụng thuế đầu tư?

Những nhân viên điều hành cấp cao đều khác biệt nhau trong phong cách cá nhân và tất cả đều phải đương đầu với những thay đổi lớn lao của môi trường và vấn đề. Hệ thống phải được xây dựng để có thể thích ứng với những điều kiện mới đó. ESS là một đáp ứng cho thách thức này. Vì ESS được thiết kế cho những người quản lý cấp cao sử dụng, những người thường ít có tiếp xúc trực tiếp hay kinh nghiệm làm việc với các hệ thống tin dựa trên máy tính, cho nên chúng thường tổ hợp các giao diện đồ họa dễ dùng.

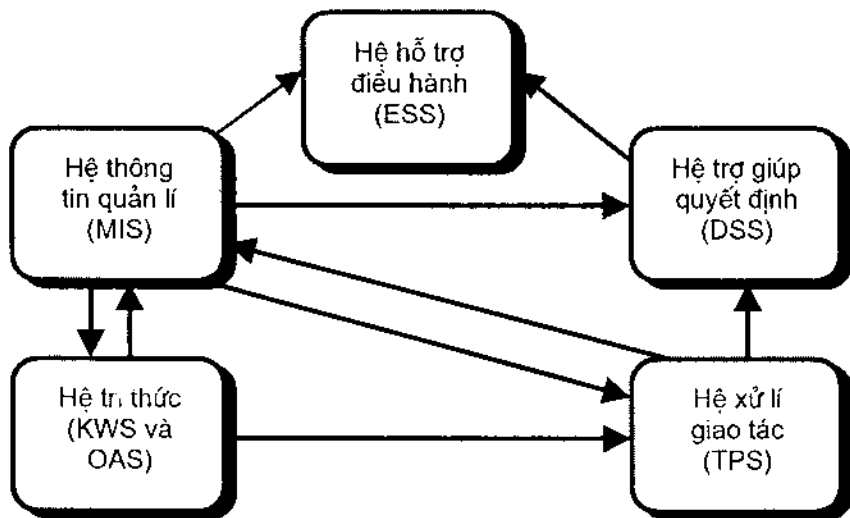
Quan hệ giữa các hệ này: tích hợp

Hình 1.15 sau minh họa cho cách thức các kiểu hệ thống khác nhau trong tổ chức có quan hệ lẫn nhau. TPS điển hình là nguồn dữ liệu chính cho các hệ thống khác, trong khi ESS lại chủ yếu là nơi nhận dữ liệu từ các hệ thống mức thấp hơn. Các kiểu hệ thống khác cũng trao đổi thông tin lẫn nhau.

Nhưng các hệ thống này nên tích hợp với nhau như thế nào? Các tổ chức liệu có nên có một hệ thống tin duy nhất phục vụ cho toàn bộ tổ chức và điều phối mọi hệ chuyên dụng đã nêu đại cương trước đây không? Liệu việc có một hệ thống toàn bộ, duy nhất như thế có phải là tốt nhất không để đảm bảo rằng thông tin có thể luân chuyển đến bất kì đâu cần tới, rằng nó là thống nhất và rằng tất cả những hệ thống mới đều được phối hợp không?

Quan điểm hệ thống toàn bộ này đang nhanh chóng trở thành lạc hậu. Trong thực tế, không tổ chức nào lại xây dựng hệ thống theo cách này. Quan điểm hệ thống toàn bộ giả thiết rằng một

chuyên gia nào đó tồn tại ở đâu đó, có thể hiểu được “toàn bộ nhu cầu thông tin” của tổ chức. Nhưng các hoạt động của thậm chí một công ti chế tạo nhỏ cũng đã đa dạng đến độ phải cần tới nhiều chuyên gia khác nhau để xây dựng các hệ thống khác nhau phục vụ cho các mục đích khác nhau.



Hình 1.15 Mối quan hệ lẫn nhau giữa các hệ thống.

Một quan điểm hiện đại hơn là ở chỗ các hệ thống nên được tích hợp lẫn nhau – tức là, chúng nên cung cấp luồng thông tin hệ thống giữa các hệ thống khác nhau. Cách tiếp cận tích hợp này có nhiều ưu điểm. Nhưng việc tích hợp lại tốn tiền.

Trong thế giới thực, người quản lí cung cấp mức độ tích hợp cần để vận hành việc kinh doanh. Mối nối giữa các hệ thống tiến hoá theo thời gian. Phần lớn các hệ thống đều được xây dựng cô lập với các hệ thống khác (trừ phi lí do kinh doanh nào đó gợi ý ra một cách tiếp cận khác). Các tổ chức đều không xây dựng tất cả mọi hệ thống ngay một lúc; tài nguyên cần để tiến hành như vậy sẽ cực kì

lớn, các vấn đề quản lý hầu như không thể vượt qua nổi. Với sự phát triển của các phần cứng và phần mềm mới - đặc biệt các hệ quản trị cơ sở dữ liệu và hệ viễn thông riêng - việc xây dựng cầu giữa các hệ thống mới và hệ thống cũ trở nên đỡ tốn kém hơn và tin cậy hơn.

Tổ chức phải gánh chịu mọi hậu quả về cách tiếp cận tiến hoá tới hệ thống. Hệ thống thường không tích hợp được như chúng đôi khi cần phải như thế. Tình huống này tạo ra các chỗ cổ chai và tính phi hiệu quả trong các hoạt động kinh doanh của công ti. Đôi khi tổ chức phải đưa ra một nỗ lực lớn lao để phát triển các hệ thống xử lý và viễn thông để tổ hợp các hệ thống.

1.7. CÔNG NGHỆ HỆ THÔNG TIN

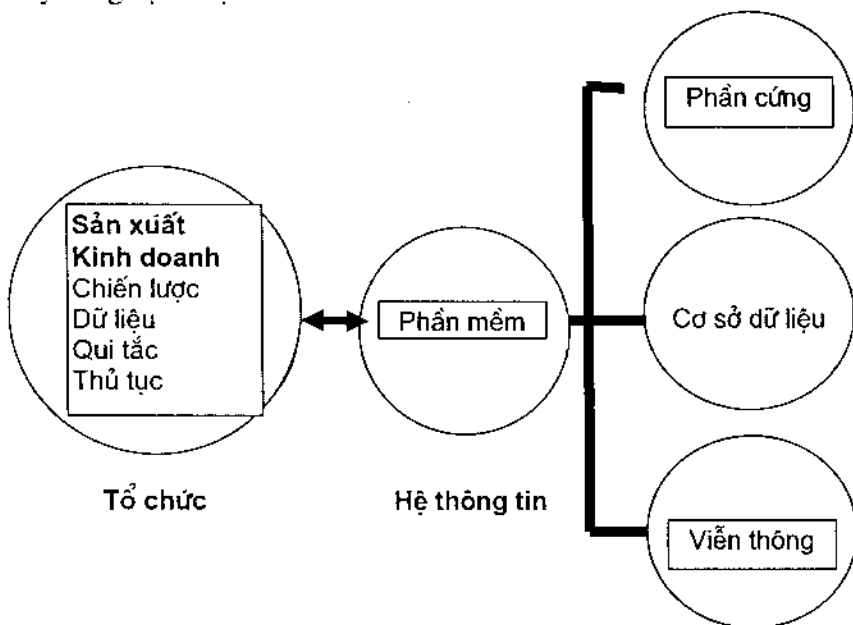
Công nghệ hệ thông tin là một trong nhiều công cụ có sẵn cho các nhà quản lý để đối phó với thay đổi. Ngày nay còn quan trọng hơn nữa là việc hệ thông tin là chất keo gắn tổ chức với nhau. Nó là công cụ để qua đó cấp quản lý thực hiện điều khiển và sáng tạo. Công nghệ hệ thông tin tích hợp nhiều tiến bộ của công nghệ thông tin áp dụng vào lĩnh vực hệ thống. Đó là sự phối hợp của các công nghệ phần cứng máy tính, phần mềm máy tính, công nghệ lưu trữ thông tin và công nghệ viễn thông (Hình 1.16).

Phần cứng máy tính là thiết bị vật lý được dùng cho các hoạt động đưa vào, xử lý và đưa ra trong một hệ thông tin. Nó bao gồm những thành phần sau: đơn vị xử lý máy tính; thiết bị đưa vào, đưa ra và lưu trữ; và phương tiện vật lý để nối các thiết bị này với nhau.

Phần mềm máy tính bao gồm các lệnh lập trình sẵn chỉ tiết để điều khiển và phối hợp các thành phần phần cứng máy tính trong một hệ thông tin.

Công nghệ lưu trữ bao gồm cả phương tiện vật lí để lưu trữ dữ liệu, như đĩa hoặc băng từ, và phần mềm cai quản việc tổ chức dữ liệu trên các phương tiện vật lí.

Công nghệ viễn thông, bao gồm cả thiết bị vật lí và phần mềm, nối các phần cứng khác nhau và truyền dữ liệu từ vị trí vật lí này sang vị trí vật lí khác..



Hình 1.16 Sự phụ thuộc lẫn nhau giữa tổ chức và hệ thống tin.

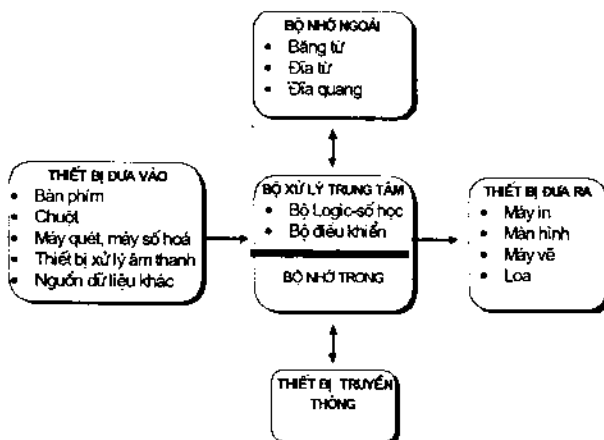
Phần **2**

NỀN TẢNG KỸ THUẬT CỦA HỆ THỐNG TIN

2.1 MÁY TÍNH VÀ XỬ LÝ THÔNG TIN

2.1.1 HỆ THỐNG MÁY TÍNH LÀ GÌ

Để tìm hiểu cách máy tính xử lý dữ liệu, cần phải hiểu các thành phần của hệ thống máy tính và cách máy tính làm việc. Dù kích cỡ thế nào, các máy tính đều biểu thị và xử lý dữ liệu theo cùng một cách, tuân theo nguyên lý công nghệ Von Neumann.



Hình 2.1 Thành phần phần cứng của hệ thống máy tính.

Cấu hình hệ thống

Hệ thống máy tính hiện nay gồm có bộ xử lý trung tâm và bốn loại thiết bị phần cứng khác : bộ nhớ chính, thiết bị đưa vào, thiết bị đưa ra và thiết bị truyền thông (xem Hình 2.1). Bộ xử lý trung tâm chứa các thành phần thao tác dữ liệu thô thành dạng hữu ích hơn và điều khiển các phần khác của hệ thống máy tính. Bộ nhớ phụ (đĩa từ, đĩa quang, băng từ...) nạp dữ liệu và chương trình vào bộ xử lý trung tâm, ghi nhớ chúng để sử dụng sau. Thiết bị đưa vào, như bàn phím, “con chuột” máy tính, máy quét... chuyển dữ liệu và

lệnh thành dạng điện tử để đưa vào vào máy tính. Thiết bị đưa ra, như màn hình, máy in, chuyển dữ liệu điện tử do máy tính tạo ra và hiển thị nó dưới dạng con người có thể hiểu được. Thiết bị truyền thông giúp kiểm soát truyền thông giữa bộ xử lý trung tâm, các thiết bị vào và ra, và người dùng cuối cùng.

BIT VÀ BYTE: CÁCH MÁY TÍNH BIỂU DIỄN DỮ LIỆU

Để cho thông tin chạy qua một hệ thống máy tính và ở dạng thích hợp cho xử lý, thì tất cả các kí hiệu, hình ảnh hay từ đều phải được thu lại thành một xâu các chữ số nhị phân. Một chữ số nhị phân được gọi là một **bit** và biểu thị cho hoặc một số không hoặc một số một. Trong máy tính, sự hiện diện của một kí hiệu điện hay từ có nghĩa là *một* và việc thiếu vắng nó có nghĩa là *không*. Máy tính số vận hành trực tiếp với chữ số nhị phân, hoặc từng chữ số đó hoặc theo chùm để tạo thành byte. Một xâu gồm 8 bit được gọi là **byte**. Mỗi byte có thể được dùng để biểu diễn cho một số thập phân, một kí hiệu, hay một phần của một hình ảnh.

THỜI GIAN VÀ KÍCH THƯỚC TRONG HỆ THỐNG MÁY TÍNH

Sau đây là vài mức đơn vị thời gian và kích thước đánh giá tốc độ và dung lượng của các hệ thống máy tính hiện đại.

Tốc độ xử lý :

Mili giây :	Một phần nghìn (1/1.000) giây.
Micro giây :	Một phần triệu (1/1.000.000) giây.
Nano giây :	Một phần tỷ (1/1.000.000.000) giây.

Bộ nhớ nói chung vận hành với tốc độ mili giây (một phần nghìn giây). Chẳng hạn, một máy tính có thể tìm thấy bản ghi thông tin về một sinh viên trong đĩa từ trong một khoảng thời gian 15 mili giây. Nó cũng làm trong vài giây để tìm tên của bạn trên một hệ thống băng từ chậm hơn. Một máy vi tính hạng trung có thể xử lý xấp xỉ 10 triệu lệnh chương trình trong 1 giây hoặc một micro

giây cho mỗi lệnh. Bộ xử lý trung tâm trong máy tính lớn hiện đại có thể xử lý trên 200 triệu lệnh trong 1 giây (200 MIPS). Với tốc độ này, bộ xử lý trung tâm vận hành với tốc độ nano giây (một phần tỉ của giây) hoặc một lệnh xử lý trong vòng 15 nano giây.

Dung lượng bộ nhớ :

Kilobyte :	$1 \text{ Kb} = 2^{10} \text{ byte} = 1024 \text{ byte}$
Megabyte :	$1 \text{ Mb} = 2^{10} \text{ Kb} = 1024 \text{ Kb}$
Gigabyte:	$1 \text{ Gb} = 2^{10} \text{ Mb} = 1024 \text{ Mb}$

Kích thước, cũng như tốc độ, là xem xét quan trọng trong hệ thống. Thông tin được ghi nhớ trong máy tính dưới dạng các con số 0 và 1 (chữ số nhị phân, hay bit), được xâu chuỗi lại để tạo nên byte. Một byte có thể được dùng để ghi nhớ một ký tự, như chữ A. Một nghìn byte (thực tế là 1024 vị trí bộ nhớ) được gọi là 1 **kilobyte**. Các máy vi tính nhỏ có bộ nhớ chính bên trong trên 600 kilobyte. Một máy tính lớn ngày nay có thể chứa 32 megabyte thông tin trong bộ nhớ chính. Mỗi megabyte xấp xỉ một triệu byte. Điều này có nghĩa là, về lý thuyết, máy có thể chứa tới 32 triệu chữ cái hay chữ số. Các thiết bị bộ nhớ phụ, như ổ đĩa cứng trong máy vi tính hay đĩa mềm trong máy tính lớn, có thể chứa đến hàng triệu byte thông tin. Máy vi tính có thể có đĩa cứng hàng trăm megabyte cho tới vài gigabyte, trong khi, một máy tính lớn có thể có nhiều ổ đĩa, mỗi cái có thể giữ đến 8 gigabyte. Một gigabyte xấp xỉ một tỷ byte.

Sự khác biệt lớn về kích cỡ và tốc độ của các phần tử chính trong hệ thống máy tính tạo ra vấn đề phối hợp. Chẳng hạn, trong khi bộ xử lý trung tâm vận hành với mức độ micro giây, và trong một vài trường hợp với mức nano giây, thì máy in thông thường vận hành ở mức vài trăm cho tới vài nghìn ký tự trong một giây. Điều này nghĩa là bộ xử lý trung tâm có thể xử lý thông tin nhanh hơn máy in có thể in ra rất nhiều. Với lý do này, bộ nhớ chính và các thiết bị nhớ phải được đặt vào giữa bộ xử lý trung tâm và máy in để

cho bộ xử lý trung tâm không bị kìm lại trong việc xử lý nhiều thông tin vì nó phải chờ máy in in ra.

Một trong các chức năng của các thiết bị truyền thông và các loại vùng nhớ khác trong hệ thống là để phân giai đoạn cho luồng thông tin vào và ra khỏi máy theo cách tận dụng tối đa việc sử dụng đơn vị xử lý trung tâm. Một phát triển chính trong ba mươi năm qua trong các hệ thống thông tin là tạo ra phần mềm hệ điều hành và những thiết bị đệm và ghi nhớ khác, tất cả chúng đều tổ hợp vào để làm tăng cường việc sử dụng toàn bộ của đơn vị xử lý trung tâm.

2.1.2 BỘ NHỚ CHÍNH

Bộ nhớ chính có ba chức năng. Nó ghi nhớ tất cả các phần của chương trình đang được thực hiện. Bộ nhớ chính cũng chứa chương trình hệ điều hành vốn quản lý việc vận hành của máy tính. Cuối cùng, vùng bộ nhớ chính giữ dữ liệu đang được chương trình sử dụng. Dữ liệu và chương trình được đặt vào trong bộ nhớ chính trước khi xử lý, giữa các bước xử lý và sau khi việc xử lý đã kết thúc, trước khi được trả về bộ nhớ phụ hay đưa ra như cái ra.

Hình 2.2 biểu thị rằng bộ nhớ chính được chia thành các ô nhớ. Mỗi ô chứa một tập hợp tám khoá hay thiết bị nhị phân, mỗi khoá trong đó có thể ghi nhớ một bit thông tin. Tập hợp tám bit trong mỗi ô nhớ đủ để ghi nhớ một byte, bằng việc dùng cách mã EBCDIC hoặc ACSII. Do đó, trong mỗi ô nhớ, chúng ta có thể ghi nhớ một chữ cái, một chữ số hay một kí hiệu đặc biệt (như \$). Mỗi ô nhớ có một địa chỉ riêng biệt chỉ ra chỗ nó được định vị trong RAM. Máy tính có thể nhớ chỗ mà tất cả các byte dữ liệu được cất giữ đơn giản bằng việc lưu giữ dấu vết địa chỉ của chúng. Mỗi vị trí bộ nhớ trong bộ nhớ chính có một địa chỉ đặc biệt- tương tự như một hộp thư. Trong hộp thư này, một byte thông tin được ghi nhớ.

	1	2	3	4	5	6	7	8	n
1	0								
2	0								
3	0								
4	0								
5	0								
6	0								
7	0								
8	0								
9	0								
n	0								

Mỗi byte trong một hộp thư riêng

Mỗi hộp thư chứa 8 khoá (0/1) hoặc transistor biểu diễn cho 8 bit :

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

 =0 theo EBCDIC

Hình 2.2 Bộ nhớ chính trong máy tính.

Hầu hết thông tin được sử dụng trong các ứng dụng máy tính đều được ghi nhớ trong các thiết bị nhớ phụ như các đĩa, băng từ, đặt bên ngoài vùng bộ nhớ chính. Để cho máy tính làm việc trên thông tin, thì thông tin phải được truyền vào bộ nhớ chính để xử lý. Do đó, dữ liệu liên tục được đọc vào và ghi ra vùng bộ nhớ chính trong khi thực hiện chương trình.

Các kiểu bộ nhớ bán dẫn

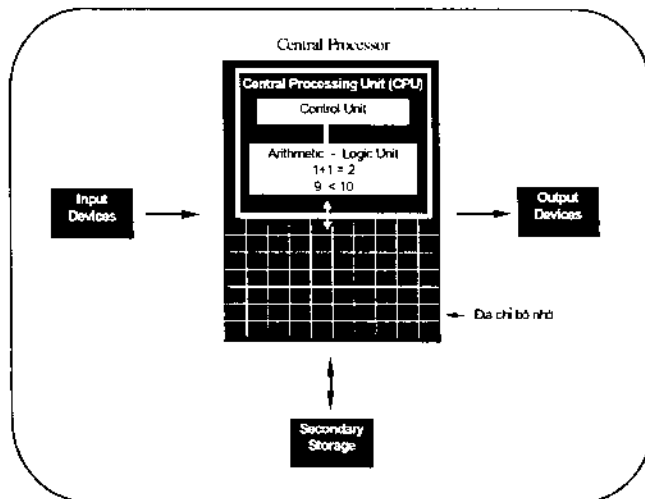
Bộ nhớ chính thực tế bao gồm các chất bán dẫn. Một chất bán dẫn là một mạch tích hợp được làm ra bằng cách in hàng nghìn hay thậm chí hàng triệu transistors tí hon lên một miếng silic nhỏ. Có nhiều loại chip bán dẫn khác nhau đang được sử dụng trong bộ nhớ chính. **RAM** (bộ nhớ truy nhập ngẫu nhiên), được sử dụng để ghi nhớ ngắn hạn dữ liệu hay lệnh chương trình. RAM hay thay đổi: nội dung của nó sẽ bị mất khi nguồn cấp điện cho máy tính bị mất hay khi máy tính bị tắt đi. **ROM**, hay bộ nhớ chỉ đọc, chỉ có thể

được đọc ra; nó không thể bị ghi vào. Chip ROM do nhà chế tạo cung cấp cùng với các chương trình đã được “đốt sẵn” hay ghi nhớ sẵn vào đó. ROM được sử dụng trong máy tính vạn năng để ghi nhớ các chương trình quan trọng hay thường được sử dụng (như các trình tính toán căn bậc hai của một số). Cách dùng khác cho chip ROM là ghi nhớ các vi chương trình của riêng nhà chế tạo như chip Hệ thống vào ra cơ bản (BIOS) được dùng trong các máy vi tính IBM Personal System/2, vốn kiểm soát việc giải quyết dữ liệu bên trong máy này.

Có hai lớp con ROM khác : PROM, hay bộ nhớ chỉ đọc lập trình được, và EPROM, hay bộ nhớ chỉ đọc lập trình được, xoá được. Chip PROM được nhà chế tạo dùng làm thiết bị điều khiển trong sản phẩm của họ. Chúng có thể được lập chương trình một lần. Bằng cách này, nhà chế tạo tránh được phí tổn chế tạo chip chuyên dụng để điều khiển các động cơ nhỏ chẳng hạn; thay vì vậy, chúng có thể được lập trình trong PROM với một chương trình chuyên dụng cho sản phẩm của họ. Do đó chip PROM có thể được sản xuất phục vụ nhiều nhà chế tạo trong loạt sản phẩm lớn. Chip EPROM được sử dụng cho việc điều khiển thiết bị, chẳng hạn như trong rô bốt, nơi mà chương trình có thể phải được thay đổi trên cơ sở thường lệ. Với chip EPROM này, chương trình có thể được xoá đi và được lập trình lại

2.1.3 ĐƠN VỊ XỬ LÝ TRUNG TÂM

Đơn vị xử lý trung tâm, hay bộ xử lý trung tâm (CPU) là chỗ của hệ thống máy tính thực hiện việc thao tác các kí hiệu, số và chữ. Nó cũng điều khiển các bộ phận khác của hệ thống máy tính. CPU bao gồm một **đơn vị logic-số học** và **đơn vị điều khiển** (xem Hình 2.3). Được nối chặt chẽ với CPU trong bộ xử lý trung tâm là **bộ nhớ chính**, nơi dữ liệu và các lệnh chương trình được ghi nhớ tạm thời trong khi xử lí. Đặc tính của CPU là rất quan trọng trong việc xác định tốc độ và khả năng của máy tính.



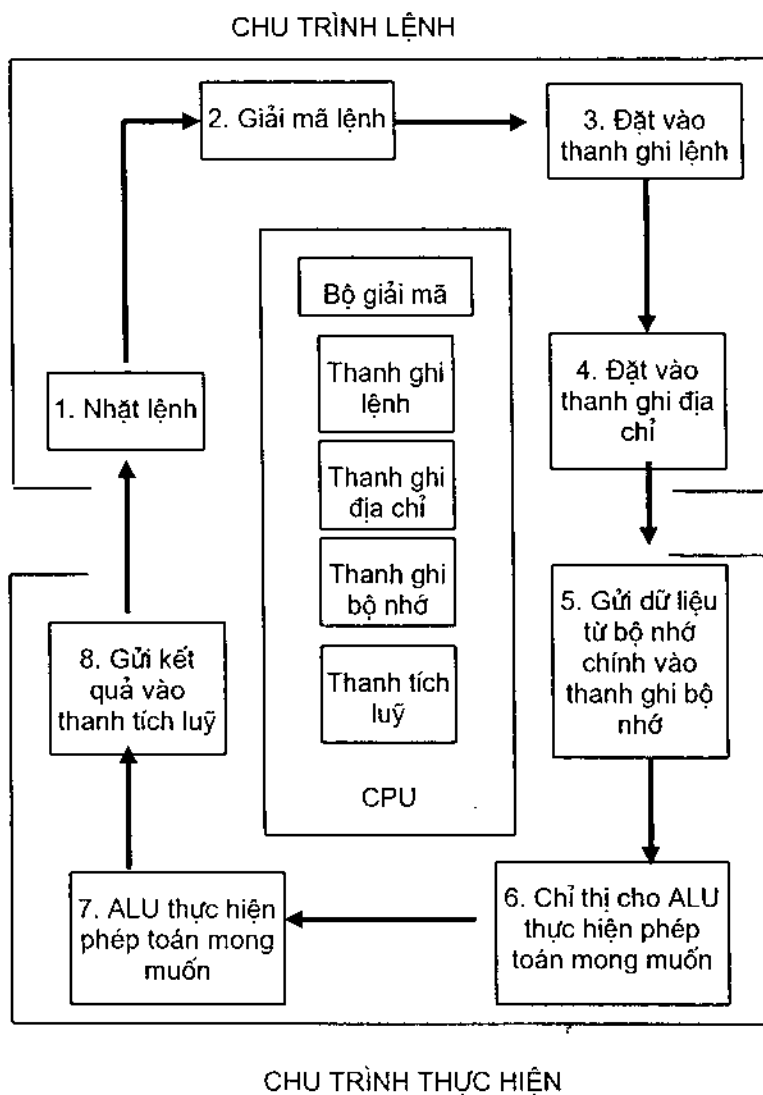
Hình 2.3 CPU và bộ nhớ chính.

ĐƠN VỊ SỐ HỌC - LOGIC

Đơn vị số học và logic (ALU) thực hiện các phép toán số học và logic chính của máy tính. ALU chỉ cần cộng và trừ các số để thực hiện các phép nhân và chia, cũng như các phép cộng và trừ. Phép nhân đơn giản là phép cộng nhanh, phép chia là phép trừ nhanh. Bên cạnh việc thực hiện các phép toán số học này, ALU phải có khả năng xác định khi nào một lượng này lớn hơn hay nhỏ hơn lượng khác và khi nào thì chúng bằng nhau. ALU có thể thực hiện các phép toán logic trên các mã nhị phân cho các chữ cũng như cho các số.

ĐƠN VỊ ĐIỀU KHIỂN

Đơn vị điều khiển phối hợp và kiểm soát các bộ phận khác của hệ thống máy tính. Nó đọc một chương trình đã ghi nhớ, mỗi



Hình 2.4 Các bước khác nhau trong chu trình máy. Chu trình máy có hai thao tác: chu trình lệnh và chu trình thực hiện.

lúc một lệnh, và chỉ thị cho các thành phần khác của hệ thống máy tính thực hiện các nhiệm vụ do chương trình yêu cầu. Chuỗi các thao tác cần để xử lý một lệnh máy được gọi là **chu trình máy**. Như minh họa trong Hình 2.4, chu trình máy gồm hai phần : chu trình lệnh và chu trình thực hiện.

Trong chu trình lệnh, đơn vị điều khiển tìm kiếm một lệnh chương trình trong bộ nhớ chính và giải mã nó. Nó đặt phần lệnh báo cho ALU phải làm gì tiếp vào thanh ghi lệnh đặc biệt và đặt phần xác định địa chỉ của dữ liệu cần dùng trong phép toán vào thanh ghi địa chỉ (Thanh ghi là một vị trí bộ nhớ tạm thời trong ALU hoặc trong đơn vị điều khiển, nó hoạt động giống như một vùng trung chuyển tốc độ cao để truyền dữ liệu và lệnh chương trình từ bộ nhớ chính đến CPU để chờ xử lý.)

Trong chu trình thực hiện, đơn vị điều khiển định vị dữ liệu cần thiết trong bộ nhớ chính, đặt nó vào thanh ghi bộ nhớ, chỉ dẫn cho ALU thực hiện thao tác cần thiết, ghi nhớ tạm thời các kết quả của thao tác trong thanh tích lũy và cuối cùng đặt kết quả vào bộ nhớ chính. Sau khi việc thực hiện của từng lệnh được hoàn tất, đơn vị điều khiển tiến thêm và đọc lệnh tiếp của chương trình.

2.1.4 BỘ VI XỬ LÝ LÀ GÌ? CHIP LÀ GÌ?

Công nghệ mạch tích hợp qui mô cực lớn VLSIC với hàng trăm nghìn (hay thậm chí hàng triệu) transistor trên một chip, tích hợp cả bộ nhớ, bộ logic và điều khiển của máy tính vào một chip; do đó mới có tên **bộ vi xử lý** hay máy tính trên một chip. Bộ vi xử lý mạnh hiện đang được sử dụng rộng rãi trong máy tính cá nhân là chip 32bit-330 MHz như Intel Pentium II.

Một vài dạng chip thông dụng của Intel được giới thiệu trong mục sau đây. Bạn sẽ thường thấy các chip được gắn nhãn như các thiết bị 8 bit, 16 bit hoặc 32 bit. Các nhãn đó nói đến **độ dài từ** hoặc số bit có thể được máy xử lý một lần. Chip 8 bit có thể xử lý 8

bit hoặc 1 byte thông tin trong một chu trình máy. Chip 32 bit có thể xử lý 32 bit hoặc 4 byte trong một chu trình. Độ dài từ càng lớn thì tốc độ của máy tính càng nhanh.

Nhân tố thứ hai ảnh hưởng tới chip là tốc độ chu trình (hay còn gọi là xung đồng hồ). Mỗi sự kiện trong máy tính phải được làm tuần tự để cho bước nọ nối một cách logic với bước kia. Đơn vị điều khiển đặt nhịp cho chip. Nhịp này được thiết lập bằng đồng hồ bên trong và được đo theo megahertz (viết tắt là MHz, kí hiệu cho hàng triệu chu trình trong một giây).

Nhân tố thứ ba ảnh hưởng tới tốc độ là độ rộng bus dữ liệu. Bus dữ liệu hoạt động như một xa lộ giữa CPU, bộ nhớ chính và với các thiết bị khác, xác định bao nhiêu dữ liệu có thể được truyền vào từng lúc. Chip 8088 sử dụng trong máy tính cá nhân PC gốc IBM, có độ dài từ là 16 bit nhưng chỉ có độ rộng bus dữ liệu 8 bit. Điều này có nghĩa là dữ liệu được xử lý trong bản thân chip CPU là theo từng khối 16 bit nhưng chỉ có thể truyền 8 bit một lần giữa CPU với bộ nhớ chính và thiết bị ngoài. Mặt khác, chip 40486, được sử dụng trong máy IBM Personal System/2 và các chip Motorola 68030 và chip 68040 sử dụng trong máy vi tính Macintosh, cả hai đều có độ dài từ 32 bit và độ rộng bus dữ liệu 32 bit. Hiển nhiên, để làm cho máy tính thực hiện nhiều lệnh hơn trong một giây thì cần phải tăng độ dài từ của bộ xử lý, tăng độ rộng bus dữ liệu hoặc tốc độ chu trình - hoặc phải tăng cả ba. Có một cách khác để làm cho các bộ vi xử lý mạnh hơn, được gọi là **tập lệnh tính toán thu gọn RISC**. RISC tăng tốc độ cho bộ vi xử lý bằng cách chỉ cấy vào chip các lệnh thường dùng nhất.

Có một số công ti chính trên thế giới chuyên tập trung vào việc chế tạo và sản xuất các vi mạch và bộ xử lý cho máy tính như Intel, Motorola,... Sau đây là mô tả về một số bộ vi xử lý do công ti Intel đã đưa ra thị trường trong những năm qua. Họ bộ vi xử lý của hãng Intel (Mỹ), nền tảng của thế giới vi tính tương thích IBM bắt

dầu từ bộ vi xử lý 8080 ra đời năm 1974 đến các bộ vi xử lý Pentium hiện đang dùng. Sự phát triển của công nghệ vi xử lý Intel đã xác nhận sự đúng đắn của dự báo do chủ tịch hãng này đưa ra : cứ sau 18 tháng các tính năng của vi xử lý lại tăng lên gấp đôi.

Intel 8080 : bộ vi xử lý tiền thân của họ 80x86 với cách lập địa chỉ và bus dữ liệu 8 bit là một trong những chip đầu tiên được dùng trong máy vi tính.

Intel 8086 : ra đời năm 1978, hậu thân của 8080 với các thanh ghi và bus dữ liệu 16 bit, lập địa chỉ 20 bit, điều khiển 1 megabyte bộ nhớ. Tốc độ 4,77Mhz, 8Mhz và 10Mhz. Vào thời điểm này, giá các thiết bị ngoại vi 16 bit còn đắt nên ít máy tính dùng chip 8086, trừ các máy tính PS/2 model 25 và 30 nhưng ít thành công.

Intel 8088 : ra đời năm 1978 có bus dữ liệu ngoài 8 bit, bus dữ liệu trong 16 bit, tốc độ 4,77 và 8 Mhz, được dùng trong các máy vi tính IBM đầu tiên (IBM PC, IBM XT, PCjunior) và các máy tương thích. Do có bus dữ liệu ngoài 8 bit nên vẫn tận dụng được các ngoại vi và chip 8 bit rẻ tiền.

Intel 8087: bộ đồng xử lý toán học làm việc với 8086/8088 và 80186/80188, có thêm các lệnh số học, lượng giác, hàm mũ và loga vốn không có trong tập lệnh của 8086/8088 và 80186/80188. Có khả năng tính toán với các số nguyên 16,32 và 64 bit, các số có dấu phẩy động 32,64 và 80 bit và các số thập phân mã hóa nhị phân 18 chữ số.

Intel 80286: còn gọi là 286, ra đời năm 1982, được dùng trong các máy IBM PC/AT và các máy tương thích năm 1984. Có các thanh ghi 16 bit, bus dữ liệu trong và ngoài 16 bit, dùng 24 bit để lập địa chỉ bộ nhớ. 286 hoạt động ở hai chế độ : chế độ thực và chế độ bảo vệ. Chế độ thực tương thích với MS-DOS và 8086/8088 nhưng chỉ có thể lập địa chỉ cho 1 megabyte bộ nhớ. Ở chế độ bảo

vệ, 286 có thể truy nhập trực tiếp tới 16 megabyte bộ nhớ, đồng thời bảo vệ hệ điều hành tránh những trục trặc có thể làm treo máy.

Intel 80386DX: còn gọi là 80386, 386, 386DX, bộ vi xử lý 32 bit ra đời năm 1985 và được dùng trong các máy IBM PC, IBM PS/2 model 80 và các máy tương thích. 386 là bộ vi xử lý 32 bit đầy đủ (thanh ghi 32 bit, bus dữ liệu 32 bit, dùng 32 bit để lập địa chỉ bộ nhớ). Tổng bộ nhớ có thể lập địa chỉ trực tiếp là 4 gigabyte. Cũng như 80286, 386 hoạt động ở hai chế độ : chế độ thực và chế độ bảo vệ. Ngoài ra, còn chế độ 386 ảo, cho phép chia 386 thành một số bộ vi xử lý 8086, mỗi bộ có không gian nhớ riêng 1 megabyte và có thể chạy chương trình riêng của mình.

Intel 386SX: biến thể rẻ tiền của 386DX ra đời năm 1988, với bus dữ liệu chỉ có 16 bit.

Intel 386SL: bộ vi xử lý dựa trên cơ sở chip 386DX, được thiết kế với các mạch quản lý năng lượng đặc biệt, nên làm việc ít tốn điện, thường dùng trong các máy xách tay (laptop và notebook).

i486DX: còn gọi là 80486 hoặc 486, ra đời năm 1989 có cấu trúc bus dữ liệu đủ 32 bit và khả năng lập địa chỉ cho 64 gigabyte bộ nhớ. Chứa hơn 1 triệu tranzito, 486 được cài sẵn bộ nhớ nhanh 8K , bộ đồng xử lý toán học 80387 và các khả năng dự phòng cho đa xử lý. Ngoài ra 486 còn sử dụng một phần cứng đặc biệt gọi là "ống dẫn" cho phép thực hiện một lệnh chỉ trong một chu kỳ nhịp đồng hồ, do đó 486 có tốc độ nhanh hơn nhiều so với 386.

i486DX2: còn gọi là 486DX2, 486D/2 hoặc đơn giản D2, ra đời năm 1992 cập nhật tính năng cao hơn đối với một số bộ VXL i486DX. Sử dụng công nghệ đồng hồ kép cho phép tốc độ làm việc bên trong bộ VXL gấp đôi tốc độ của hệ thống. i486DX2 nâng cao các tính năng chung của hệ thống mặc dù tốc độ chung của toàn bộ hệ máy tính không nhân gấp đôi vì đồng hồ kép chỉ nhân đôi tốc độ bên trong bộ VXL không đụng đến phần còn lại của hệ thống. Ví dụ phiên bản 66 Mhz của i486DX2 có tốc độ bên trong là 66Mhz và phần còn lại của hệ thống chỉ chạy ở 33Mhz. Do làm việc với

tốc độ gấp đôi nên các chip i486DX2 tạo ra nhiều nhiệt hơn i486DX, do đó cần lắp thêm bộ phận tiêu nhiệt trên chip.

i486SX, i486SX2: còn gọi là 80486SX hoặc 486SX (tương ứng 80486SX2 hoặc 486SX2), là các phiên bản rẻ tiền của i486DX và không có bộ đồng xử lý toán học cài sẵn. Các tính năng khác như i486DX. Như DX2, bộ VXL i486SX2 có đồng hồ gấp đôi.

i486SL: Phiên bản dùng ít năng lượng của i486DX tiện dùng cho các máy xách tay.

Intel 80287, 80387, 80487SX: các bộ đồng xử lý toán học làm việc với 80286, 80386 và 80486.

Intel Pentium: ra đời tháng 3-1993, xem như kế tiếp họ VXL i486. Trước khi phát hành, nó được Intel mã tên là P5, nhưng giới công nghiệp và người dùng gọi là 80586 hoặc 586. Sau này Intel chính thức đặt tên là Pentium. Pentium là bộ vi xử lý siêu vô hướng, dựa trên CISC, chứa 3,1 triệu tranzito. Các phiên bản đầu tiên chạy ở 60 và 66 Mhz, tiếp sau là 100, 133, 166, 200, 233 Mhz với công nghệ MMX và bus hệ thống 66 Mhz. Nó có bus địa chỉ 32 bit, bus dữ liệu 64 bit, bộ đồng xử lý toán học gài sẵn, bộ quản lý bộ nhớ, hai bộ nhớ nhanh mỗi bộ 8K và bộ quản lý hệ thống SMM sở hữu riêng của Intel. Pentium sử dụng dự báo rẽ nhánh *if-then*, nạp tức thời vào ống dẫn những lệnh hoặc đo nhánh *if* hoặc đo nhánh *then* điều khiển, khiến cho tính năng của hệ thống nhanh hơn. Pentium cũng chấp nhận kiểm tra dư thừa chức năng (FRC), dùng một Pentium tách riêng để so sánh với Pentium chính nhằm bảo đảm không xảy ra lỗi.

Intel Pentium II : còn gọi là P6, loại VXL có tốc độ nhanh nhất hiện nay, những kiểu đầu tiên ra đời năm 1995. Chứa trên 6 triệu tranzito và tốc độ xử lý tới 250 triệu lệnh trong 1 giây (250 MIPS), thực hiện hơn 2 lệnh trong mỗi chu kỳ đồng hồ. Các kiểu Pentium II 233, 266, 300, 330 Mhz chạy với bus hệ thống 66Mhz,

các kiểu 400 và 450 Mhz chạy với bus hệ thống 100Mhz được đưa ra thị trường trong năm 1998.

Intel Pentium OverDrive : chip nâng cấp bộ xử lí do người dùng tự lắp trên các máy Intel 80487SX và DX có sẵn để cắm OverDrive trên bo mẹ, nhằm tăng gấp đôi tốc độ xử lí bên trong bộ xử lí. Một số hãng sản xuất máy tính gọi các máy tính 486 có khả năng nâng cấp là máy Pentium OverDrive.

2.1.5 BỘ NHỚ PHỤ

Bên cạnh bộ nhớ chính, nơi thông tin và chương trình được ghi nhớ cho việc xử lí ngay, các hệ thống máy tính hiện đại còn dùng các kiểu ghi nhớ khác để hoàn thành nhiệm vụ của mình. Các hệ thống tin cần ghi nhớ thông tin bên ngoài máy tính trong trạng thái không biến động (không yêu cầu nguồn điện) và ghi nhớ khối lượng dữ liệu lớn hơn kích cỡ bộ nhớ trong của máy tính. Bộ nhớ dữ liệu tương đối dài hạn bên ngoài CPU và bộ nhớ chính được gọi là **bộ nhớ phụ**.

Bộ nhớ chính là nơi sử dụng công nghệ nhanh nhất và đắt nhất. Như được biểu thị trong Bảng 2.1, thực tế có ba loại bộ nhớ chính: thanh ghi, bộ nhớ ẩn và RAM. **Thanh ghi** là bộ nhớ nhanh nhất và đắt nhất, nơi một lượng nhỏ dữ liệu và lệnh thường trú trong phần nghìn giây ngay trước khi sử dụng, tiếp theo đó là bộ nhớ **ẩn** (bộ nhớ tốc độ cao cho các lệnh và dữ liệu thường dùng) và bộ nhớ RAM cho khối lượng lớn dữ liệu. Việc truy nhập vào thông tin được ghi nhớ trong bộ nhớ chính là theo dạng điện tử và xuất hiện gần ngang tốc độ ánh sáng. Bộ nhớ phụ thì không biến động và duy trì dữ liệu ngay cả khi máy tính đã bị tắt đi. Có nhiều loại bộ nhớ phụ: thông dụng nhất là băng từ, đĩa từ và đĩa quang. Các phương tiện này có thể truyền khối lượng lớn dữ liệu vào CPU một cách nhanh chóng. Nhưng vì bộ nhớ phụ cần chuyển động cơ khí

để thu được việc truy nhập vào dữ liệu, nên nó tương đối chậm so với bộ nhớ chính.

BẢNG 2.1 THIẾT BỊ GHI NHỚ DỮ LIỆU TRONG MÁY VI TÍNH

Kiểu bộ nhớ	Dung lượng toàn bộ	Thời gian truy nhập
Bộ nhớ chính		
Thanh ghi	1 kilobyte	0.01 micro giây
Bộ nhớ ẩn	1 - 256 kilobyte	0,1 micro giây
RAM	16 - 512 megabyte	0,5 micro giây
Bộ nhớ phụ		
Đĩa cứng	800 - 8000 megabyte	15-20 mili giây
Đĩa mềm mật độ cao	2,8 megabyte	200 mili giây
Đĩa quang	640 megabyte	200-500 mili giây
Băng từ	40 megabyte	1-2 giây

Băng từ là loại thiết bị cổ nhưng vẫn quan trọng đối với bộ nhớ phụ khối lượng thông tin lớn. Nó được dùng chủ yếu trong các ứng dụng theo lô trên máy tính lớn và cho việc lưu trữ dữ liệu. Nói chung, băng từ cho các hệ thống lớn bao gồm các cuộn băng 14 inch, dài tới 3600 feet và rộng 0,5 inch. Nó tương tự như băng ghi âm cassette gia đình nhưng chất lượng cao hơn. Bên cạnh băng từ cho các hệ thống lớn, máy vi tính và một số máy tính mini cũng dùng băng cassette

Ưu điểm chính của băng từ là nó khá rẻ, tương đối ổn định và có thể ghi nhớ khối lượng thông tin rất lớn. Nhược điểm chính của băng từ là ở chỗ nó ghi nhớ dữ liệu theo kiểu tuần tự và tốc độ

tương đối chậm nếu so với tốc độ của các phương tiện nhớ phụ khác. Để tìm một bản ghi cá nhân được ghi trên băng từ, thì phải đọc từ đầu băng cho tới vị trí của bản ghi mong muốn. Điều này có nghĩa là CPU phải đọc duyệt qua lần lượt từng bản ghi từ đầu băng tới vị trí mong muốn.

Đĩa từ. Phương tiện nhớ phụ được sử dụng rộng rãi nhất hiện nay là đĩa từ. Có hai loại đĩa từ: đĩa mềm và đĩa cứng. **Đĩa cứng** là các đĩa phẳng bằng thép mỏng có phủ oxit kim loại. Các đĩa này được gắn với nhau theo trục tâm thẳng đứng. Thông tin được ghi trên các mặt đĩa từ mặt đầu và cuối, trên mỗi mặt dữ liệu được ghi theo từng rãnh. Dàn đĩa nói chung được bọc kín với môi trường ngoài và quay với tốc độ từ 3500 - 5000 vòng / phút, tạo ra tốc độ luồng khí quãng 80 km/h trên bề mặt đĩa.

Thông tin được ghi hay đọc từ đĩa qua các đầu đọc/ghi, thực sự bay trên các đĩa đang quay. Khác với các đầu kim máy hát, các đầu đọc ghi này không bao giờ chạm vào mặt đĩa mà bay lơ lửng cách một vài phần nghìn inch trên bề mặt đĩa. Một hạt khối hay sợi tóc con người cũng đủ làm cho đầu đọc ghi kẹt vào đĩa.

Tốc độ truy nhập dữ liệu trên đĩa là một hàm của tốc độ quay của đĩa và tốc độ của tay truy nhập. Các đầu đọc ghi phải tự định vị và dàn đĩa phải quay cho tới khi định vị được đúng thông tin. Các đĩa tiên tiến và đắt có tốc độ truy nhập từ 1,5 đến 10 mili giây và dung lượng lên tới 8-9 gigabyte. Mỗi rãnh đĩa chứa nhiều chùm, mỗi chùm chứa nhiều sector. Như vậy dung lượng đĩa được tính qua số lượng mặt đĩa, số đầu từ, số lượng rãnh trên đĩa, số lượng sector trên đĩa và số lượng byte trong mỗi sector.

Đĩa mềm là đĩa phẳng, 5,25 inch hay 3,5 inch làm bằng phim nhựa có phủ từ tính. Các đĩa này có dung lượng thay đổi từ 360 Kb cho tới 2,8 megabyte và có tỉ lệ truy nhập chậm hơn đĩa cứng. Đĩa mềm cùng dàn nhiều đĩa dùng phương pháp chia sector để ghi nhớ dữ liệu. Trong hầu hết các kiểu đĩa mềm, mỗi sector có cùng dung lượng nhớ (dữ liệu được ghi dày đặc hơn ở các rãnh bên trong). Mỗi

sector được gán cho một số hiệu duy nhất. Dữ liệu có thể được định vị bằng việc dùng địa chỉ có chứa số hiệu của sector và số hiệu bản ghi dữ liệu cá nhân.

Đĩa từ nói chung có nhiều ưu thế hơn băng từ. Trước hết chúng cho phép truy nhập trực tiếp vào từng bản ghi. Mỗi bản ghi có thể được cho một địa chỉ vật lý chính xác dưới dạng số hiệu trụ và rãnh, và đầu đọc / ghi có thể đi trực tiếp tới địa chỉ đó và truy nhập vào thông tin quăng 10 đến 60 mili giây. Bộ nhớ đĩa thường được nói tới như **thiết bị nhớ truy nhập trực tiếp (DASD)**. Tuy thế DASD lại khá đắt nếu so với băng từ. Hơn nữa, việc cập nhật thông tin được ghi trên đĩa sẽ phá hủy thông tin cũ và đĩa cũng có thể bị hỏng cho nên cần có môi trường làm việc sạch sẽ.

Đĩa quang, cũng còn được gọi là đĩa compact hay đĩa lade, ghi nhớ dữ liệu với mật độ nhiều lần lớn hơn mật độ của đĩa từ. Dữ liệu được ghi lên đĩa quang khi thiết bị lade đốt một vi lỗ trong tầng phản xạ của rãnh xoắn ốc. Thông tin nhị phân được mã hoá theo chiều dài của những lỗ này và khoảng cách giữa chúng. Đĩa quang cũng có thể ghi nhớ khối lượng lớn dữ liệu, không chỉ văn bản mà còn âm thanh và video chuyển động, dưới dạng rất gọn gàng. Đĩa quang được đọc bằng việc chiếu tia lade năng lượng thấp từ đầu quang quét trên đĩa.

Hệ thống đĩa quang thông dụng nhất được dùng với máy vi tính được gọi là **CD-ROM (bộ nhớ chỉ đọc đĩa compact)**. Đĩa compact 4,75 inch có thể chứa tới 660 megabyte, gần gấp 300 lần đĩa mềm mật độ cao. CD-ROM là bộ nhớ chỉ đọc, không ghi được dữ liệu mới lên nó. Hệ thống đĩa quang **WORM** (ghi một lần, đọc nhiều lần) cho phép người dùng ghi dữ liệu chỉ một lần lên đĩa quang. Các **đĩa quang từ** ghi lại được đang bắt đầu trở thành công cụ cho lưu trữ dữ liệu. Bề mặt đĩa được phủ một chất liệu từ mà có thể được đảo cực từ khi bị làm nóng. Để ghi dữ liệu, một chùm tia lade năng lượng cao đốt nóng một chỗ tí hon trong môi giới từ để cho phép nó chấp nhận hình mẫu từ. Dữ liệu có thể được đọc ra

bằng việc chiếu một chùm tia lade năng lượng thấp vào tầng từ và đọc ánh sáng phản xạ. Đĩa quang từ là xoá được và có thể ghi lại hàng triệu lần. Tốc độ truy nhập đĩa quang, trong khi chậm hơn đĩa từ, vẫn tiếp tục được nâng cao làm cho đĩa quang thành công nghệ bộ nhớ rất hấp dẫn cho những năm 1990.

2.1.6 THIẾT BỊ VÀO RA

Con người tương tác với hệ thống máy tính chủ yếu qua các thiết bị vào và ra. Tiến bộ trong các hệ thống tin không chỉ dựa vào tốc độ và khả năng của CPU mà còn cả vào tốc độ, khả năng và thiết kế của các thiết bị vào và ra. Các thiết bị vào/ra thường được gọi là thiết bị ngoại vi.

Thiết bị đưa vào

Phương pháp vào dữ liệu truyền thống là gõ bàn phím. Việc gõ vào máy đọc bìa thuộc vào thời kì đầu của tin học. Các nhân viên vào dữ liệu dùng máy đọc bìa để mã hoá các kí tự trên bìa 80 cột, dành cho mỗi kí tự một cách đọc lỗ duy nhất trên bìa. Máy đọc bìa cơ điện cảm nhận các lỗ hổng hay kín trên bìa. Một bìa có thể nhớ được tới 80 byte thông tin (80 cột). Máy gõ lên băng và gõ vào đĩa cũng cho phép dữ liệu được gõ trực tiếp lên băng từ hay đĩa từ để máy tính xử lí về sau. Các phương pháp này nay đã được thay thế bởi những phương pháp đưa vào hiệu quả và trực tiếp hơn. Ngày nay phần lớn dữ liệu đều được đưa trực tiếp vào máy tính bằng việc dùng thiết bị cuối vào dữ liệu và chúng được xử lí trực tuyến.

Chuột máy tính. Các hành động “trò và bấm” của chuột máy tính đã làm cho chuột thành ngày càng phổ cập thay cho việc gõ bàn phím và chỉ lệnh văn bản. Chuột là một thiết bị dùng tay thường được nối với máy tính qua dây cáp. Người dùng máy tính di chuyển chuột trên bàn để điều khiển vị trí của con chạy trên màn

hình hiển thị. Một khi chuột đã vào vị trí mong muốn thì người dùng có thể nhấn nút trên chuột để chọn chỉ lệnh. Chuột cũng có thể được dùng để “vẽ” ảnh trên màn hình.

Màn hình chạm. Màn hình chạm rất dễ dùng và hấp dẫn với những người không thể dùng bàn phím truyền thống. Người dùng có thể đưa vào một lượng hạn chế dữ liệu bằng việc chạm ngón tay vào bề mặt của màn hình hiển thị nhạy cảm. Với đồ hoạ mẫu, âm thanh và đơn chọn đơn giản, màn hình chạm cho phép người dùng chọn lựa bằng việc chạm vào những phần đặc biệt của màn hình.

Tự động vào dữ liệu tại nguồn. Việc tự động vào dữ liệu tại nguồn nắm bắt ngay dữ liệu dưới dạng máy tính đọc được tại lúc và nơi chúng được tạo ra. Các thiết bị **nhận dạng kí tự quang học (OCR)** dịch các dấu hiệu, kí tự và mã đặc biệt thành dạng số. Ưu điểm của vào dữ liệu tại nguồn là giảm thiểu sai sót do việc gõ vào.

Công nghệ **nhận dạng mực từ (MIRC)** được dùng chủ yếu trong việc xử lí séc cho công nghiệp ngân hàng. Phần đáy của mỗi séc đều có chứa các kí tự được in bằng việc dùng một loại mực đặc biệt về tên ngân hàng, tài khoản séc và số séc. Máy đọc MIRC dịch các kí tự này trên séc đã chi tiêu và gửi về ngân hàng để xử lí dưới dạng số trong máy tính. Số tiền trên séc, được ghi tay bằng mực thường, thì phải gõ tay vào máy.

Mã được dùng phổ biến nhất là **mã vạch**, được dùng tại các điểm bán hàng trong siêu thị và cửa hàng đại lí, trong bệnh viện, thư viện, phương tiện vận tải, hoạt động quân sự... Mã này có chứa thêm dữ liệu thời gian, ngày tháng và vị trí để xác định dữ liệu.

Máy quét số được dùng để chuyển các hình ảnh hay tài liệu thành dạng số và là thành phần chủ chốt của hệ thống xử lí ảnh.

Thiết bị đưa vào tiếng nói chuyển lời thành dạng số. Phần mềm nhận dạng tiếng nói so sánh các mẫu điện do tiếng nói của người nói tạo ra với tập các mẫu đã ghi sẵn. Nếu sánh đúng thì cái vào được chấp nhận. Phần lớn các hệ thống tiếng nói mới chỉ có

vốn “tù vụng” giới hạn cỡ vài trăm tới vài nghìn từ và chỉ có thể chấp nhận các chỉ lệnh rất đơn giản.

Thiết bị đưa ra

Thiết bị đưa dữ liệu ra chính là **thiết bị cuối hiển thị**, hay VDT, đôi khi cũng còn được gọi là thiết bị đèn hình (CRT) và **máy in**.

CRT có lẽ là phương tiện phổ biến nhất để đưa thông tin ra trong hệ thống máy tính hiện đại. Nó làm việc giống như đèn hình ti vi, với “súng” điện tử bắn ra chùm tia điện tử để làm sáng các chấm điểm pixel trên màn hình. Màn hình càng nhiều pixel thì độ phân giải càng cao. Có thể phân loại màn hình CRT như loại đơn sắc hay màu hoặc theo khả năng hiển thị của chúng. Một số màn hình chỉ hiển thị văn bản, trong khi các màn hình khác hiển thị cả văn bản và đồ họa. CRT điển hình hiển thị 80 cột và 24 dòng dữ liệu văn bản. Màn hình đồ họa thường dùng tương ứng bit. Tương ứng bit cho phép từng pixel trên màn hình được máy tính định địa chỉ và thao tác (khác với cả khối pixel trong màn hình hiển thị kí tự). Điều này đòi hỏi nhiều bộ nhớ máy tính hơn nhưng cho phép các chi tiết mịn hơn và khả năng tạo ra bất kì loại ảnh nào trên màn hình.

Máy in tạo ra bản in trên giấy về thông tin đưa ra. Chúng bao gồm máy in có va đập (kiểu máy chữ chuẩn hay máy in ma trận chấm) và máy in không va đập (máy in lade, phun mực và truyền nhiệt). Một số máy in in từng kí tự mỗi lúc, nhưng số khác có thể in cả dòng hay cả trang một lúc. Máy in dòng có khả năng in cả dòng trong một bước in và in được tới 3000 dòng một phút. Máy in trang in cả trang một lúc, đưa ra 20 000 dòng một phút. Máy in kim in được từ 60 đến 600 kí tự một phút. Nói chung máy in có va đập thì chậm hơn máy in không va đập. Máy in lade cho máy vi tính in được từ bốn đến tám trang một phút. Máy in trong các trung tâm máy tính lớn có thể in trên 100 trang một phút.

Các thiết bị khác. Microfilm và microfiche đã được dùng để ghi nhớ gọn các ảnh vi phim. Thiết bị đưa ra tiếng nói chuyển dữ liệu ra dạng số trở lại thành tiếng nói. Âm thanh được ghi âm sẵn, được mã hoá và ghi nhớ trên đĩa, được dịch ngược lại thành lời.

2.1.7 TIẾN HOÁ CỦA PHẦN CỨNG MÁY TÍNH

Đã có bốn giai đoạn chính, hay **thế hệ máy tính**, trong tiến hoá của phần cứng máy tính, mỗi thế hệ đều được phân biệt bởi một công nghệ khác nhau cho các thành phần thực hiện công việc xử lý của máy tính. Mỗi thế hệ đều mở rộng đáng kể năng lực xử lý của máy tính và khả năng ghi nhớ trong khi đồng thời lại giảm giá thành. Chẳng hạn, giá của việc thực hiện 100 000 tính toán tụt từ vài đô la trong những năm 1950 xuống ít hơn 0,025 đô la trong những năm 1980. Những thay đổi thế hệ này trong phần cứng máy tính đã đi kèm với những thay đổi thế hệ trong phần mềm máy tính làm cho máy tính ngày một mạnh hơn, không đắt và dễ dùng.

Các thế hệ phần cứng máy tính

Thế hệ thứ nhất và thế hệ thứ hai của phần cứng máy tính đều dựa trên công nghệ đèn điện tử và transistor, trong khi các thế hệ thứ ba và thứ tư lại dựa trên công nghệ bán dẫn.

THẾ HỆ THỨ NHẤT : ĐÈN ĐIỆN TỬ, 1946-1956

Thế hệ máy tính thứ nhất dựa vào đèn điện tử để ghi nhớ và xử lý thông tin. Các đèn này tiêu thụ rất nhiều năng lượng, đòi hỏi lại ngắn, và sinh ra rất nhiều nhiệt. Với kích thước khổng lồ, máy tính thế hệ thứ nhất có bộ nhớ và khả năng xử lý cực kì giới hạn và đã được dùng cho công việc khoa học và kĩ nghệ rất giới hạn. Kích cỡ bộ nhớ chính tối đa là 2000 byte (2 kilobyte), với tốc độ 10 000 lệnh trong 1 giây. Trống từ được dùng làm bộ nhớ trong và bìa đục

lỗi được dùng làm bộ nhớ ngoài. Các việc như cho chạy chương trình hay in cái ra phải được điều phối thủ công.

THẾ HỆ THỨ HAI TRANSISTOR, 1957-1963

Trong thế hệ thứ hai, bóng bán dẫn transistor đã thay thế cho đèn điện tử làm thiết bị ghi nhớ và xử lý thông tin. Transistor ổn định và tin cậy hơn nhiều so với đèn điện tử, chúng sinh ra ít nhiệt hơn và tiêu thụ ít năng lượng hơn. Tuy nhiên, từng transistor phải được chế tạo riêng biệt và được nối dây trong bo mạch in, một tiến trình chậm chạp và tốn kém. Bộ nhớ lõi từ là công nghệ bộ nhớ chính cho thời kỳ này. Nó bao gồm các xuyên từ nhỏ (cỡ 1 mm đường kính), mà có thể được phân cực theo 1 hay 2 hướng để biểu diễn cho một bit dữ liệu. Dây nối được xâu qua các lõi này cho cả việc đọc và ghi dữ liệu. Hệ thống này phải được lắp ráp thủ công và do đó rất đắt. Máy tính thế hệ thứ hai này có tới 32 KB bộ nhớ RAM và tốc độ đạt tới từ 200 000-300 000 lệnh một giây. Năng lực xử lý và bộ nhớ được nâng cao của các máy tính thế hệ thứ hai làm cho chúng được sử dụng rộng rãi hơn trong công tác khoa học và cho các nhiệm vụ kinh doanh như tự động tính lương làm hoá đơn.

THẾ HỆ THỨ BA : MẠCH TÍCH HỢP (IC), 1964-1979

Máy tính thế hệ thứ ba dựa trên các mạch tích hợp, được làm ra bằng cách in hàng trăm và sau đó hàng nghìn transistor tí hon lên một tấm silic nhỏ. Những thiết bị này được gọi là thiết bị bán dẫn. Bộ nhớ máy tính đã mở rộng tới 2 megabyte và tốc độ tăng tới 5 MIPS. Việc tăng năng lực xử lý cho phép phát triển những phần mềm đặc biệt gọi là hệ điều hành vốn tự động cai quản các chương trình và việc truyền thông giữa CPU, máy in và các thiết bị khác. Công nghệ máy tính thế hệ thứ ba đã đưa vào các phần mềm có thể dùng được máy, làm cho máy tính mở rộng vai trò của chúng trong kinh doanh.

THẾ HỆ THỨ TƯ : MẠCH TÍCH HỢP CỖ LỚN, 1980 - NAY

Thế hệ thứ 4 trải rộng từ năm 1980 đến nay. Máy tính trong thời kì này sử dụng mạch tích hợp cỡ rất lớn (VLSIC), được đóng gói với cỡ từ 200 000 đến trên 3 000 000 transistor trên một chip. Giá cả đã tụt xuống điểm mà máy tính để bàn là không đắt và có sẵn để dùng trong kinh doanh và đời sống hàng ngày. Năng lực của máy tính, có thời chiếm cả một phòng lớn, bây giờ nằm trên một chiếc bàn nhỏ. Kích cỡ bộ nhớ tăng nhanh như nắm vượt quá một gigabyte trong những máy tính thương mại lớn; tốc độ xử lý đã vượt quá 200 MIPS.

Công nghệ VLSIC đã tiếp nhiên liệu cho một phong trào tăng trưởng hướng tới việc giảm thiểu kích cỡ xuống micro - sự phát triển nhanh chóng của máy tính nhỏ, nhanh và rẻ đến mức chúng đã trở thành phổ biến ở đâu cũng có. Chẳng hạn, nhiều tính năng “thông minh” đã được đưa vào ô tô, máy nghe stereo, đồ chơi, đồng hồ, máy ảnh và các thiết bị khác làm cho dễ dùng hơn, đều dựa trên các bộ vi xử lý.

Các loại máy tính: Máy tính lớn, máy tính mini, máy vi tính, trạm làm việc và siêu máy tính

Máy tính biểu diễn và xử lý dữ liệu theo cùng cách, nhưng có các phân loại khác nhau. Chúng ta có thể sử dụng kích cỡ và tốc độ xử lý để phân loại các máy tính hiện nay thành máy tính lớn, máy tính mini, trạm làm việc, máy vi tính và siêu máy tính.

MÁY TÍNH LỚN, MÁY TÍNH MINI, MÁY VI TÍNH

Máy tính lớn là loại máy tính lớn nhất, để đầy một nhà với bộ nhớ đồ sộ và năng lực xử lý cực kì nhanh. Nó được sử dụng cho các ứng dụng thương mại, khoa học, quân sự rất lớn trong đó máy

tính phải thao tác với khối lượng dữ liệu lớn hoặc nhiều tiến trình phức tạp.

Máy tính mini là máy tính hạng trung có kích cỡ bàn làm việc, thường thường sử dụng cho các trường đại học, xí nghiệp hay phòng thí nghiệm nghiên cứu.

Máy vi tính là máy có thể để được trên bàn hay mang từ phòng nọ sang phòng kia. Máy vi tính được dùng như máy cá nhân cũng như máy trong kinh doanh.

Trạm làm việc cũng có thể đặt vừa trên mặt bàn nhưng có khả năng xử lý đồ họa và toán học mạnh hơn máy vi tính và có thể thực hiện nhiều nhiệm vụ phức tạp trong cùng thời gian mà máy vi tính có thể làm. Trạm làm việc được dùng cho công việc khoa học, kỹ nghệ và thiết kế, vốn đòi hỏi khả năng tính toán và đồ họa mạnh.

Siêu máy tính là máy siêu mạnh và phức tạp thường được dùng cho các nhiệm vụ đòi hỏi các tính toán cực nhanh chóng và phức tạp với hàng trăm nghìn nhân tố biến thiên, nhưng chúng cũng đang được dùng trong kinh doanh nữa. Máy tính đại diện cho mỗi loại được liệt kê trong bảng 2.2.

Tuy vậy nói chung, máy tính lớn vẫn có thể được coi là loại có RAM từ 50 megabyte đến 1 gigabyte, máy tính mini, từ 10 tới 650 megabyte RAM; trạm làm việc, 8 tới 300 megabyte RAM; còn máy vi tính: 640 kilobyte tới 64 megabyte RAM. Bảng 2.2 biểu diễn khả năng của các máy tính thương mại trong mỗi loại hiện nay.

Thuật ngữ *máy vi tính* đôi khi được sử dụng đồng nghĩa với máy tính cá nhân, vì vi tính ban đầu được dự định làm công cụ cá nhân, một người dùng. Tuy nhiên, máy vi tính đã trở thành mạnh đến mức chúng không còn bị giới hạn vào các hệ thống tin cá nhân nữa. Máy vi tính có thể vận hành hoặc như máy riêng lẻ cá nhân với năng lực xử lý biệt lập hoặc như một phần của mạng các thiết bị thông minh qui mô công ti hay phòng ban. Chúng có thể được kết

nối với các máy vi tính khác, các thiết bị viễn thông, trạm làm việc hay máy tính lớn.

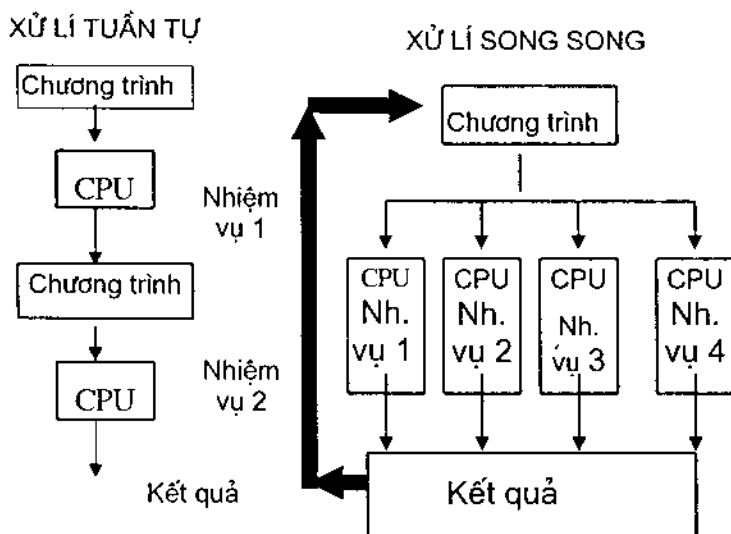
Bảng 2.2 MÔ TẢ CÁC MÁY TÍNH

Kiểu	Ví dụ	Bộ nhớ (MB)	Hiệu năng (tốc độ)	Độ dài từ (bit)	Giá (1000\$)
Máy tính lớn	IBM Enterprise System 9000	1000	240 MIPS	32	55.000
Máy tính nhỏ	DEC VAX 7000 Model 600	64- 2350	124-508 transaction per second	32	165- 355
Trạm làm việc	SUN SPARC station	32- 512	50.2 SPEC int92 marks	32	17 và hơn nữa
Máy tính nhỏ	Compaq Prolinea Model 4633	4-32	10.8 MIPS	32	1,3-1,7
Siêu máy tính	Cray C90	256- 8.000	1-16 gigaflop	64	3000- 35.000

Ghi chú : Các nhà chế tạo máy tính sử dụng cách đo khác nhau về hiệu năng máy tính. Nhiều công ti dùng MIPS, nhưng vài hãng thích đo hiệu năng theo giao tác trên giây. Hiệu năng của siêu máy tính được đo theo flops (số phép toán dấu phẩy động trong một giây).

SIÊU MÁY TÍNH VÀ XỬ LÝ SONG SONG

Siêu máy tính là một kiểu máy tính đặc biệt phức tạp và mạnh được dùng chủ yếu cho các tính toán nhanh chóng và phức tạp với hàng trăm hay hàng nghìn nhân tố biến thiên. Siêu máy tính về truyền thống đã được dùng cho nghiên cứu vũ khí, dự báo thời tiết, các ứng dụng dầu khí và công nghệ, tất cả đều dùng những mô hình và mô phỏng toán học phức tạp. Mặc dầu cực kì đắt tiền, siêu máy tính đang bắt đầu được sử dụng trong kinh doanh.



Hình 2.5 Xử lý tuần tự và xử lý song song.

Siêu máy tính có thể thực hiện các tính toán phức tạp và khối lượng lớn gần như tức thời bởi vì chúng có thể thực hiện hàng tỉ và thậm chí hàng trăm tỉ tính toán trong một giây - nhanh hơn máy tính lớn lớn nhất nhiều lần. Siêu máy tính không xử lý từng lệnh mỗi lúc mà thay vì thế lại dựa trên **xử lý song song**. Như minh họa trong hình 2.5, nhiều đơn vị xử lý (CPU) chia một bài toán thành

nhiều phần nhỏ hơn và làm việc đồng thời trên chúng. Một số siêu máy tính thử nghiệm dùng tới 64 000 bộ xử lí. Nói tới việc lấy một nhóm bộ xử lí để giải một bài toán ngay một lúc thì dễ hơn là thực hiện nó. Việc này đòi hỏi phải nghĩ lại vấn đề và phần mềm đặc biệt để có thể phân chia bài toán cho các bộ xử lí khác nhau theo cách hiệu quả nhất, cung cấp dữ liệu cần thiết và lắp ráp nhiều nhiệm vụ còn lại để đạt tới giải pháp thích hợp. Phần lớn phần mềm hiện đang được dùng đều được viết cho các máy xử lí tuần tự.

Vấn đề với việc phân loại này là ở chỗ khả năng của máy thay đổi quá nhanh. Máy vi tính hiện nay có năng lực tính toán của máy tính lớn trong những năm 80 hay máy tính mini vài năm trước đây. Máy vi tính mạnh có khả năng đồ họa phức tạp và khả năng xử lí tương tự như một trạm làm việc. Máy vi tính vẫn không thể thực hiện nhiều nhiệm vụ một lúc như máy tính lớn, máy tính mini hoặc trạm làm việc; mà chúng cũng không thể được nhiều người dùng sử dụng đồng thời như máy tính lớn. Tuy nhiên, những khác biệt này sẽ trở thành không rõ ràng trong tương lai. Vài thập kỷ sau, máy tính để bàn rất có thể sẽ có năng lực và tốc độ xử lí của siêu máy tính thời nay.

2.1.8 XU HƯỚNG CÔNG NGHỆ THÔNG TIN

Những tiến bộ trong khoa học vật liệu, chế tạo và khái niệm tin học hứa hẹn duy trì hình mẫu tăng trưởng lịch sử trong phần cứng. Trong ba mươi năm qua, chi phí tính toán đã giảm theo hệ số 10 cứ mỗi thập kỉ và khả năng lại tăng lên theo hệ số ít nhất là 100 cho mỗi thập kỉ. Cái đà này rất có thể sẽ được duy trì. Các bộ vi xử lí ngày nay có thể xem như máy tính lớn ngày xưa được đặt lên bàn hay cuối cùng cho vào cặp hay túi áo, máy tính trên một chip sẽ đi vào tận ô tô, vũ khí quân sự, người máy và các vật dụng gia đình. Máy tính và công nghệ thông tin có liên quan sẽ nhào nặn dữ liệu, hình ảnh và âm thanh, gửi chúng qua mạng máy tính bao la mà có thể xử lí tất cả chúng một cách dễ dàng.

Multimedia được định nghĩa là công nghệ làm thuận tiện cho việc tích hợp hai hay nhiều kiểu môi giới như văn bản, đồ họa, âm thanh, tiếng nói, video chuyển động, video tĩnh hay hoạt hình trong ứng dụng dựa trên máy tính. Từ những năm 1990 cho tới thế kỉ hai mươi mốt, multimedia sẽ là nền tảng cho các sản phẩm và dịch vụ tiêu thụ mới, như sách báo điện tử, công nghệ trình bày lớp học điện tử, hội thảo video, dụng cụ thiết kế ảnh đồ họa, thư nói và video điện tử.

Các hệ thống multimedia tổ hợp các yếu tố của máy tính cá nhân ngày nay (máy tính, máy in, bàn phím và chuột) với hai yếu tố mới: âm thanh và hình ảnh.

Siêu chip. Các nghiên cứu viên về chất bán dẫn vẫn tiếp tục tìm các phương tiện đóng gói các mạch dây đặc hơn, để cho hàng triệu transistor có thể được đóng gói trong chất silic cỡ móng tay. Cách chính để các bộ vi xử lí được chế tạo ra thực hiện nhanh hơn là co lại khoảng cách giữa các transistor. Tiến trình này làm cho dòng điện ít phải đi xa hơn. Một bước trong việc chế tạo bộ vi xử lí là khắc các đường dẫn vào trong chất silic để tạo nên đường viền của mạch điện. Càng làm hẹp lại các đường tạo nên transistor thì càng làm tăng số transistor có thể được co lại trong một chip và những mạch này sẽ vận hành nhanh hơn. Vì số các transistor có thể được co sát lại trên một chip silic cứ gấp đôi lên sau mỗi 18 tháng, nên có thể hình dung một bộ vi xử lí vào năm 2000 có có khoảng từ 50 đến 100 triệu transistor. Giới hạn cho cách tiếp cận này có thể sớm sẽ đạt tới, nhưng các nghiên cứu viên vẫn đang thử nghiệm chất liệu mới để làm tăng tốc độ bộ vi xử lí.

Máy tính thế hệ năm. Máy tính qui ước dựa trên nguyên lí kiến trúc Von Neumann, xử lí thông tin một cách tuần tự, mỗi lúc một lệnh. Trong tương lai, nhiều máy tính sẽ dùng cách xử lí song song và xử lí song song cực lớn để trộn lẫn tiếng nói, hình ảnh và dữ liệu từ các nguồn khác nhau, bằng việc dùng trí tuệ nhân tạo và các mô hình toán học phức tạp.

Máy tính xử lí song song cực lớn có các mạng khổng lồ các chip xử lí đan dệt với nhau theo cách linh hoạt và phức tạp. Đối lập với xử lí song song, nơi mà một số nhỏ các chip chuyên dụng mạnh nhưng đắt được nối với nhau, các máy song song cực lớn móc xích hàng trăm hay thậm chí hàng nghìn chip thường dùng, không đắt để tấn công vào những vấn đề tính toán lớn, đạt tới tốc độ siêu máy tính.

2.2 PHẦN MỀM HỆ THÔNG TIN

2.2.1 PHẦN MỀM LÀ GÌ

Phần mềm là các lệnh chi tiết kiểm soát sự vận hành của hệ thống máy tính. Không có phần mềm thì phần cứng máy tính không thể thực hiện được nhiệm vụ mà chúng ta trao cho máy tính. Các chức năng của phần mềm là: 1) quản lí tài nguyên máy tính của tổ chức; 2) cung cấp công cụ cho con người tận dụng ưu thế của những tài nguyên này; và 3) hành động như một trung gian giữa tổ chức và thông tin được lưu giữ.

Chương trình phần mềm là một chuỗi các câu lệnh hay lệnh cho máy tính. Việc xử lí viết ra hay mã hoá chương trình được gọi là lập trình, và người chuyên môn làm việc này được gọi là người lập trình.

Khái niệm chương trình được lưu trữ có nghĩa là chương trình phải được lưu giữ trong bộ nhớ chính của máy tính cùng với dữ liệu cần thiết để thực hiện, hay có các lệnh của nó được máy tính thực hiện. Một khi chương trình đã kết thúc thực hiện, thì phần cứng máy tính có thể được dùng cho nhiệm vụ khác khi một chương trình mới được nạp vào trong bộ nhớ.

Các kiểu phần mềm chính

Có ba kiểu phần mềm chính: phần mềm hệ thống, phần mềm ứng dụng và phần mềm người dùng cuối. Mỗi loại đều thực hiện một chức năng khác nhau. **Phần mềm hệ thống** là tập các chương trình tổng quát quản lý tài nguyên của máy tính, như đơn vị xử lý trung tâm, các móc nối truyền thông, và các thiết bị ngoại vi. Người lập trình viết phần mềm hệ thống được gọi là người lập trình hệ thống.

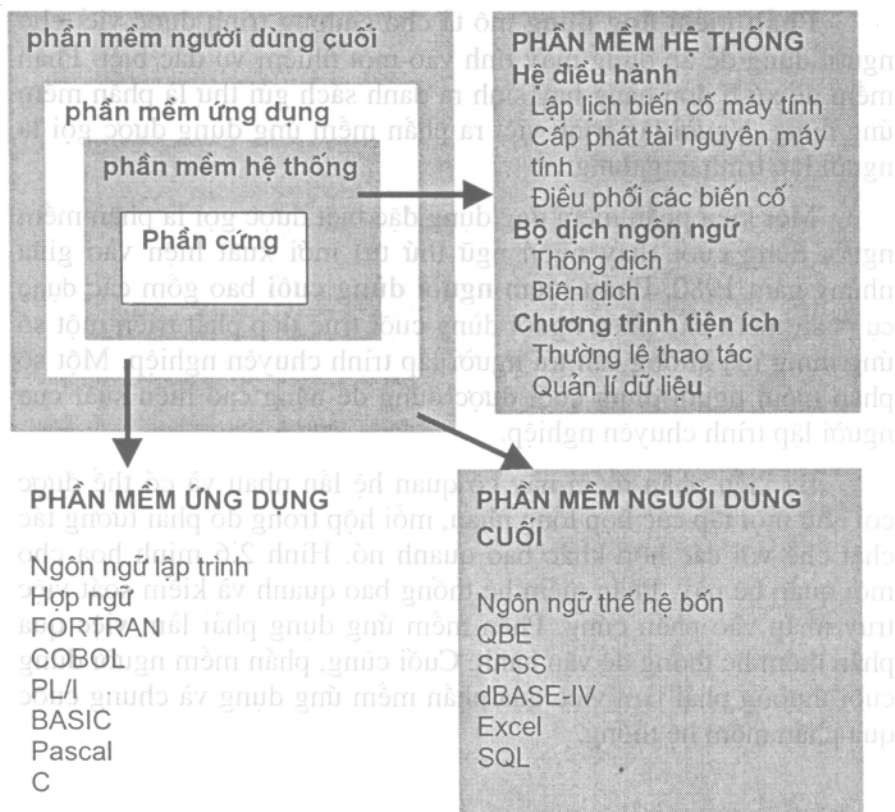
Phần mềm ứng dụng mô tả cho chương trình được viết cho người dùng để áp dụng máy tính vào một nhiệm vụ đặc biệt. Phần mềm để xử lý đơn hàng hay sinh ra danh sách gửi thư là phần mềm ứng dụng. Người lập trình viết ra phần mềm ứng dụng được gọi là người lập trình ứng dụng.

Một kiểu phần mềm ứng dụng đặc biệt được gọi là phần mềm người dùng cuối (hay ngôn ngữ thứ tư) mới xuất hiện vào giữa những năm 1980. **Phần mềm người dùng cuối** bao gồm các dụng cụ phần mềm cho phép người dùng cuối trực tiếp phát triển một số ứng dụng mà không cần tới người lập trình chuyên nghiệp. Một số phần mềm người dùng cuối được dùng để nâng cao hiệu suất của người lập trình chuyên nghiệp.

Ba kiểu phần mềm này có quan hệ lẫn nhau và có thể được coi như một tập các hộp lồng nhau, mỗi hộp trong đó phải tương tác chặt chẽ với các hộp khác bao quanh nó. Hình 2.6 minh họa cho mối quan hệ này. Phần mềm hệ thống bao quanh và kiểm soát việc truy nhập vào phần cứng. Phần mềm ứng dụng phải làm việc qua phần mềm hệ thống để vận hành. Cuối cùng, phần mềm người dùng cuối thường phải làm việc qua phần mềm ứng dụng và chung cuộc qua phần mềm hệ thống.

2.2.2 PHẦN MỀM HỆ THỐNG

Phần mềm hệ thống phối hợp những bộ phận khác nhau của hệ thống máy tính và làm trung gian giữa phần mềm ứng dụng và phần cứng máy tính. Phần mềm hệ thống, vốn quản lý và kiểm soát các hoạt động của máy tính được gọi là **hệ điều hành**. Các phần mềm hệ thống khác bao gồm các chương trình dịch ngôn ngữ máy tính, vốn chuyển các ngôn ngữ lập trình thành ngôn ngữ máy và các chương trình tiện ích, thực hiện các nhiệm vụ xử lý chung.



Hình 2.6 Ba kiểu phần mềm chính.

Các chức năng của hệ điều hành

Một cách dễ nhìn vào hệ điều hành là như người quản lý chính của hệ thống. Phần mềm hệ điều hành quyết định tài nguyên máy tính nào sẽ được dùng, chương trình nào sẽ được cho chạy, và trình tự mà theo đó các hoạt động sẽ xảy ra .

Hệ điều hành thực hiện ba chức năng. Nó cấp phát và phân bổ tài nguyên hệ thống; nó lập lịch dùng các tài nguyên máy tính và công việc máy tính; và nó điều phối hoạt động của hệ thống máy tính.

Cấp phát và phân bổ

Hệ điều hành cấp phát tài nguyên cho công việc quản lý theo hàng đợi thực hiện. Nó cung cấp vị trí trong bộ nhớ chính cho dữ liệu và chương trình và kiểm soát các thiết bị vào và ra như máy in, thiết bị cuối, và móc nối truyền thông.

Lập lịch

Hàng nghìn mẫu công việc có thể được tiến hành đồng thời trên máy tính. Hệ điều hành quyết định khi nào phải lập lịch cho những việc đã được đệ trình và khi nào thì điều phối việc lập lịch trong các vùng khác nhau của máy tính để cho các phần khác nhau của các việc khác nhau có thể được làm việc vào cùng lúc. Chẳng hạn, trong khi một chương trình đang thực hiện, thì hệ điều hành lại lập lịch để dùng các thiết bị vào ra. Không phải tất cả mọi việc đều được thực hiện theo thứ tự chúng được đệ trình; hệ điều hành phải lập lịch cho những việc này tương ứng với ưu tiên tổ chức. Việc xử lý đơn hàng trực tuyến có thể có quyền ưu tiên hơn việc sinh ra danh sách gửi thư và nhận.

Điều phối

Hệ điều hành điều phối các hoạt động của hệ thống máy tính. Nó giữ dấu vết của mỗi công việc máy tính và có thể cũng giữ dấu vết về người dùng máy tính, về chương trình nào đã được cho chạy, và về bất kì dự định không được phép nào để truy nhập vào hệ thống.

Hiển nhiên hệ điều hành của máy tính lớn bản thân nó cũng đã là một chương trình rất lớn. Bởi lí do này, chỉ một phần của hệ điều hành mới thực tế được lưu giữ trong bộ nhớ chính. Phần lớn hệ điều hành được lưu giữ trong bản sao trên đĩa, mà với nó bộ nhớ chính truy nhập vào rất nhanh. Bất kì khi nào các phần của hệ điều hành được chương trình ứng dụng cần tới, thì chúng đều được truyền từ đĩa và nạp vào bộ nhớ chính. Thiết bị mà toàn bộ hệ điều hành được lưu giữ được gọi là **thiết bị thường trú hệ thống**.

Đa lập trình, phân thời, bộ nhớ ảo và đa xử lí

Làm sao mà đến hàng nghìn người hay hơn ngồi tại các thiết bị cuối đồng thời dùng được hệ thống tin máy tính khi mà như chúng ta đã nói trong các phần trước, máy tính chỉ có thể thực hiện một lệnh từ một chương trình vào mỗi lúc? Làm sao máy tính có thể chạy hàng nghìn chương trình được? Câu trả lời là ở chỗ máy tính có hàng loạt khả năng hệ điều hành chuyên dụng.

ĐA LẬP TRÌNH

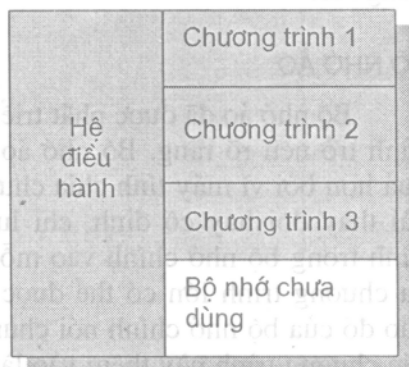
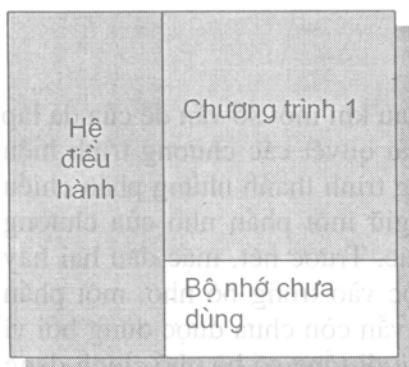
Khả năng hệ điều hành quan trọng nhất cho việc dùng chung tài nguyên máy tính là **đa lập trình**. Đa lập trình cho phép nhiều chương trình dùng chung tài nguyên của hệ thống máy tính vào bất kì lúc nào thông qua việc dùng tương tranh CPU. Với sử dụng tương tranh, chúng ta ngụ ý rằng chỉ một chương trình thực tế dùng CPU vào bất kì lúc nào nhưng chúng ta cũng ngụ ý rằng các nhu cầu vào/ra của chương trình khác có thể được thực hiện vào cùng

lúc. Hai hay nhiều chương trình đang kích hoạt vào cùng lúc, nhưng chúng không dùng cùng tài nguyên máy tính một cách đồng thời. Với đa lập trình, một nhóm chương trình lần lượt nắm quyền dùng bộ xử lý.

Hình 2.7 chỉ ra cách ba chương trình trong môi trường đa lập trình có thể được lưu giữ trong bộ nhớ chính. Chương trình thứ nhất thực hiện cho tới khi một biến cố vào/ra được đọc vào trong chương trình. Thế rồi hệ điều hành chỉ thị cho một kênh (một bộ xử lý nhỏ được giới hạn cho các chức năng vào và ra) đọc cái vào và chuyển cái ra tới thiết bị ra. CPU chuyển sang chương trình thứ hai cho tới khi câu lệnh vào/ra xuất hiện. Tại điểm này, CPU chuyển sang việc thực hiện chương trình thứ ba, và cứ thế, cho tới khi cuối cùng cả ba chương trình đã được thực hiện. Lưu ý rằng việc ngắt trong xử lý bị gây ra bởi các biến cố xảy ra trong bản thân chương trình. Theo cách này, nhiều chương trình khác nhau có thể thực hiện vào cùng lúc, mặc dầu các tài nguyên khác nhau bên trong máy tính thực tế được dùng.

Hệ thống truyền thống
không có đa lập trình

Môi trường
đa lập trình



Hình 2.7 Thực hiện một chương trình so với đa lập trình.

Bạn có thể quan sát ưu thế của đa lập trình bằng việc so sánh hệ thống đa lập trình với hệ điều hành đầu tiên, vốn thực hiện một chương trình mỗi lần. Trước đa lập trình, bất kì khi nào chương trình đọc dữ liệu từ băng từ hay đĩa từ hay ghi dữ liệu ra máy in, thì toàn bộ CPU đều dừng lại. Đây là cách dùng máy tính rất không hiệu quả. Với đa lập trình, tỉ lệ dùng CPU cao hơn nhiều.

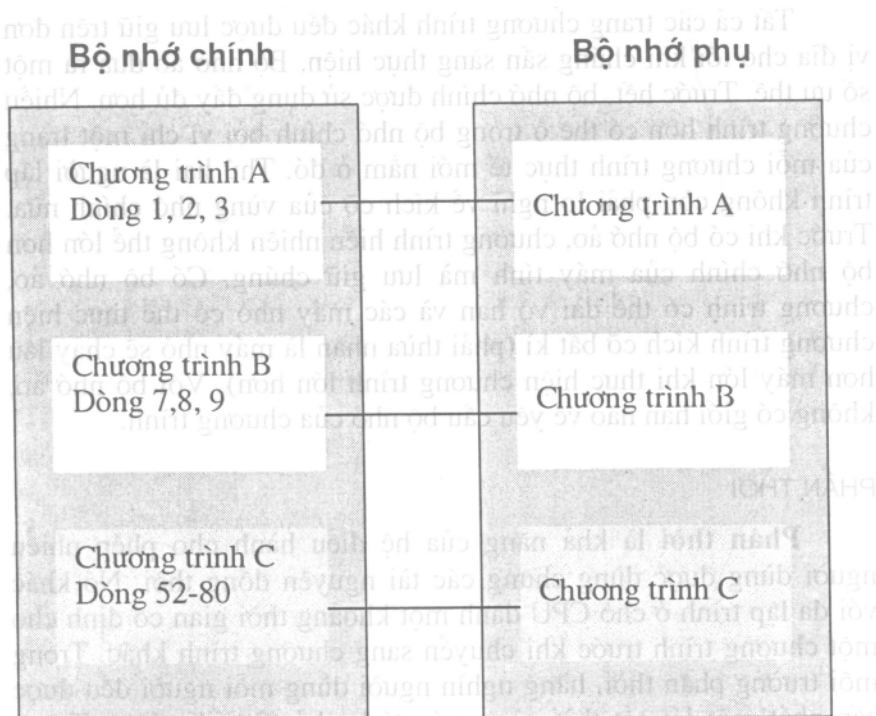
ĐA NHIỆM

Đa nhiệm nói tới đa lập trình trên hệ điều hành một người dùng như hệ điều hành trên máy vi tính. Một người có thể cho chạy hai hay nhiều chương trình tương tranh trên một máy tính đơn lẻ. Chẳng hạn, một đại diện bán hàng có thể viết bức thư cho các khách hàng tiềm năng bằng chương trình xử lí văn bản trong khi đồng thời dùng chương trình cơ sở dữ liệu để tìm tất cả các tiếp xúc bán hàng trong một thành phố đặc biệt hay vùng địa lí. Thay vì kết thúc phiên làm việc của mình bằng chương trình xử lí văn bản, trở về hệ điều hành, và rồi khởi đầu phiên làm việc với chương trình cơ sở dữ liệu, thì đa nhiệm cho phép người đại diện bán hàng hiển thị cả hai chương trình trên màn hình máy tính và làm việc với chúng cùng lúc.

BỘ NHỚ ẢO

Bộ nhớ ảo đã được phát triển sau khi một số vấn đề của đa lập trình trở nên rõ ràng. Bộ nhớ ảo giải quyết các chương trình hiệu quả hơn bởi vì máy tính chia chương trình thành những phần chiều dài thay đổi hay cố định, chỉ lưu giữ một phần nhỏ của chương trình trong bộ nhớ chính vào mỗi lúc. Trước hết, mặc dầu hai hay ba chương trình lớn có thể được đọc vào trong bộ nhớ, một phần nào đó của bộ nhớ chính nói chung vẫn còn chưa được dùng bởi vì các chương trình này thêm vào làm ít đi tổng số bộ nhớ chính đang có. Thứ hai là với kích thước cố giới hạn của bộ nhớ chính, chỉ một phần nhỏ của chương trình mới có thể thường trú trong bộ nhớ chính vào bất kì lúc nào. Chẳng hạn, nhiều chương trình nghiệp vụ

có thể đòi hỏi tới 200Kb bộ nhớ, mà máy tính chỉ có 1 Mb bộ nhớ chính. Do đó, chỉ vài chương trình mới có thể thường trú trong bộ nhớ vào bất kì lúc nào.



Hình 2.8 Bộ nhớ ảo.

Chỉ vài lệnh của chương trình mới thực tế thực hiện vào bất kì lúc nào. Bộ nhớ ảo tận dụng ưu thế này của xử lí. Bộ nhớ ảo chia chương trình thành một số phần chiều dài cố định được gọi là **trang**, hay thành các phần có chiều dài biến thiên được gọi là **đoạn**. Mỗi một trong những phần này đều tương đối nhỏ (một trang xấp xỉ 2-4 Kb). Điểm ngắt thực tế giữa các đoạn có thể được xác

định hoặc do người lập trình hoặc do hệ điều hành. Điều này cho phép một số rất lớn chương trình thường trú trong bộ nhớ chính, chỉ như một trang hay đoạn của mỗi chương trình thực tế được đặt ở đó (xem Hình 2.8)

Tất cả các trang chương trình khác đều được lưu giữ trên đơn vị đĩa cho tới khi chúng sẵn sàng thực hiện. Bộ nhớ ảo đưa ra một số ưu thế. Trước hết, bộ nhớ chính được sử dụng đầy đủ hơn. Nhiều chương trình hơn có thể ở trong bộ nhớ chính bởi vì chỉ một trang của mỗi chương trình thực tế mới nằm ở đó. Thứ hai là người lập trình không còn phải lo nghĩ về kích cỡ của vùng nhớ chính nữa. Trước khi có bộ nhớ ảo, chương trình hiển nhiên không thể lớn hơn bộ nhớ chính của máy tính mà lưu giữ chúng. Có bộ nhớ ảo, chương trình có thể dài vô hạn và các máy nhỏ có thể thực hiện chương trình kích cỡ bất kỳ (phải thừa nhận là máy nhỏ sẽ chạy lâu hơn máy lớn khi thực hiện chương trình lớn hơn). Với bộ nhớ ảo, không có giới hạn nào về yêu cầu bộ nhớ của chương trình.

PHÂN THỜI

Phân thời là khả năng của hệ điều hành cho phép nhiều người dùng được dùng chung các tài nguyên đồng thời. Nó khác với đa lập trình ở chỗ CPU dành một khoảng thời gian cố định cho một chương trình trước khi chuyển sang chương trình khác. Trong môi trường phân thời, hàng nghìn người dùng mỗi người đều được cấp phát một lát cắt thời gian máy tính nhỏ (2 mili giây). Trong khoảng thời gian này, mỗi người dùng được tự do thực hiện bất kỳ thao tác được yêu cầu nào; vào cuối chu kỳ này, người dùng khác được trao cho lát cắt thời gian 2 mili giây của CPU. Việc bố trí này cho phép nhiều người dùng được nối với một CPU đồng thời, mỗi người chỉ nhận được một khoảng nhỏ thời gian CPU. Nhưng vì CPU vận hành tại mức độ nano giây, nên CPU có thể hoàn thành nhiều công việc trong hai mili giây.

ĐA XỬ LÝ

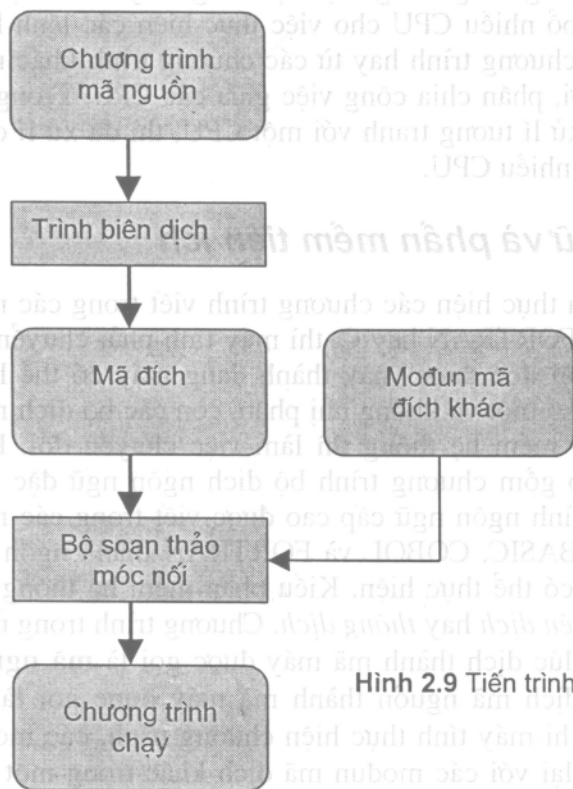
Đa xử lý là khả năng của hệ điều hành móc nối hai hay nhiều CPU để làm việc song song trong một hệ thống máy tính. Hệ điều hành có thể phân bố nhiều CPU cho việc thực hiện các lệnh khác nhau từ cùng một chương trình hay từ các chương trình khác nhau một cách đồng thời, phân chia công việc giữa các CPU. Trong khi đa lập trình dùng xử lý tương tranh với một CPU, thì đa xử lý dùng xử lý đồng thời với nhiều CPU.

Dịch ngôn ngữ và phần mềm tiện ích

Khi máy tính thực hiện các chương trình viết trong các ngôn ngữ như COBOL, FORTRAN hay C, thì máy tính phải chuyển đổi các lệnh "con người đọc được" này thành dạng máy có thể hiểu. Máy tính hiểu các số một và không nhị phân, còn các bộ dịch ngôn ngữ có trong phần mềm hệ thống thì làm việc chuyển đổi. Phần mềm hệ thống bao gồm chương trình bộ dịch ngôn ngữ đặc biệt, dịch các chương trình ngôn ngữ cấp cao được viết trong các ngôn ngữ lập trình như BASIC, COBOL và FORTRAN thành ngôn ngữ máy mà máy tính có thể thực hiện. Kiểu phần mềm hệ thống này được gọi là trình *biên dịch* hay *thông dịch*. Chương trình trong ngôn ngữ cấp cao trước lúc dịch thành mã máy được gọi là **mã nguồn**. Trình **biên dịch** dịch mã nguồn thành mã máy được gọi là **mã đích**. Ngay trước khi máy tính thực hiện chương trình, các modul mã đích được gắn lại với các modul mã đích khác trong một tiến trình được gọi là soạn thảo móc nối. Modul kết quả nạp vào là điều máy tính thực tế thực hiện. Hình 2.9 minh họa cho tiến trình dịch ngôn ngữ.

Một số ngôn ngữ lập trình, như BASIC, không dùng trình biên dịch mà dùng **trình thông dịch**, vốn dịch từng câu lệnh mã nguồn mỗi lúc thành mã máy và thực hiện nó. Các ngôn ngữ thông dịch như BASIC đưa ra phản hồi ngay lập tức cho người lập trình

nếu như có sai lầm phạm phải, nhưng chúng thực hiện rất chậm bởi vì chúng được dịch từng câu lệnh một mỗi lúc.



Hình 2.9 Tiến trình dịch ngôn ngữ.

Trình hợp dịch cũng tương tự như trình biên dịch nhưng được dùng để dịch chỉ cho hợp ngữ thành mã máy.

Phần mềm hệ thống bao gồm **chương trình tiện ích** cho các nhiệm vụ thường lệ, lặp lại, như sao chép, xóa bộ nhớ chính, tính căn bậc hai, hay sắp xếp. Nếu bạn đã làm việc trên máy tính và đã thực hiện những chức năng như tạo lập tệp mới, xóa tệp cũ, hay

định dạng đĩa, thì bạn đã làm việc với các chương trình tiện ích. Chương trình tiện ích là những chương trình đã viết sẵn, được lưu giữ để cho chúng có thể được mọi người dùng hệ thống máy tính dùng chung và có thể được dùng nhanh chóng trong nhiều ứng dụng hệ thống tin khác nhau khi được yêu cầu.

Hệ điều hành máy vi tính

Giống như bất kì phần mềm nào khác, phần mềm máy vi tính dựa trên hệ điều hành và phần cứng máy tính riêng. Gói phần mềm được viết cho hệ điều hành máy vi tính này không thể chạy trên hệ khác được. Bản thân hệ điều hành máy vi tính cũng có những tính năng phân biệt - như liệu chúng có hỗ trợ cho đa nhiệm hay công việc đồ họa hay không - vốn xác định ra kiểu ứng dụng mà chúng thích hợp với.

Đa nhiệm là một trong những sức mạnh chính của hệ điều hành như OS/2 của IBM đối với dòng máy vi tính IBM Personal Sysytem/2 và máy tương hợp IBM hay UNIX, PC-DOS và MS-DOS, hệ điều hành cũ hơn cho máy tính cá nhân của IBM và IBM PC, không cho phép đa nhiệm, mặc dầu Microsoft Corporation đã tiếp thị phần mềm Windows để tạo ra môi trường đa nhiệm cho các chương trình DOS.

Bảng 2.3 so sánh các hệ điều hành vi tính hàng đầu - OS/2, UNIX, DOS, Windows NT và Macintosh. DOS là hệ điều hành phổ cập nhất cho máy vi tính 16 bit. Nó vẫn còn được dùng rộng rãi ngày nay với nhiều máy vi tính mạnh dựa trên chuẩn máy vi tính IBM bởi vì rất nhiều phần mềm ứng dụng có sẵn đã được viết cho các hệ thống dùng DOS. (PC-DOS được dùng chỉ cho các máy vi tính IBM, MS-DOS được dùng với các máy vi tính 16 bit khác mà vận hành giống như máy vi tính IBM.) DOS bản thân nó không hỗ trợ cho đa nhiệm và giới hạn kích cỡ chương trình trong bộ nhớ 640Kb.

Bảng 2.3 SO SÁNH CÁC HỆ ĐIỀU HÀNH MÁY VI TÍNH

Hệ điều hành	Tính năng
OS/2	Hệ điều hành cho dòng máy vi tính IBM System/2 và IBM tương hợp. Có thể lấy ưu thế của bộ vi xử lý 32 bit. Hỗ trợ cho đa nhiệm và kết mạng và có thể chạy các chương trình cần nhiều hơn 640Kb bộ nhớ
Windows NT	Hệ điều hành 32 bit cho máy vi tính và trạm làm việc không bị giới hạn vào các bộ vi xử lý của Intel. Hỗ trợ đa nhiệm, đa xử lý và kết mạng và có thể chạy các chương trình cần nhiều hơn 640 Kb bộ nhớ
UNIX (XENIX)	Được dùng cho máy vi tính mạnh, trạm làm việc và máy tính mini. Hỗ trợ cho đa nhiệm, xử lý đa người dùng và kết mạng. Là khả chuyển với các mô hình phần cứng khác nhau.
System 7	Hệ điều hành cho máy tính Macintosh. Hỗ trợ đa nhiệm và có khả năng đồ hoạ mạnh và khả năng đa mê đĩa.
PC-DOS	Hệ điều hành cho máy tính cá nhân IBM. Giới hạn chương trình dùng bộ nhớ tối 640Kb
MS-DOS	Chứa các tính năng tương tự như PC-DOS nhưng là hệ điều hành cho máy nhái IBM PC (máy vi tính vận hành như máy tính cá nhân IBM)

OS/2 là hệ điều hành vững chãi được dùng cho máy vi tính IBM Personal System/2 và máy tương hợp IBM. OS/2 có thể chạy nhanh hơn DOS (vốn chỉ đánh địa chỉ dữ liệu theo từng chùm 16-

bit) bởi vì nó có thể đánh địa chỉ dữ liệu theo các chùm 32 bit. OS/2 được dùng cho các ứng dụng tiêu thụ nhiều bộ nhớ phức tạp hay những ứng dụng đòi hỏi kết mạng hay đa nhiệm. OS/2 hỗ trợ cho đa nhiệm, thích ứng cho các ứng dụng lớn, cho phép các ứng dụng chạy đồng thời, hỗ trợ các ứng dụng mạng, và là hệ điều hành được bảo vệ hơn nhiều. Một ứng dụng hỏng thì ít có khả năng làm cho toàn bộ hệ thống và các ứng dụng khác sập theo với nó. Hệ điều hành này đòi hỏi phần cứng máy tính mạnh- tối thiểu là bộ vi xử lý 80386, 4 Mb RAM và 60 Mb đĩa cứng. OS/2 cung cấp cho các máy tính để bàn khả năng của hệ điều hành máy lớn, như đa nhiệm và hỗ trợ đa người dùng trong mạng.

Windows NT là một hệ điều hành mới do Microsoft phát triển với tính năng làm cho nó thích hợp cho các ứng dụng nghiệp vụ trong môi trường kết mạng hơn là DOS hay DOS với lớp vỏ giao diện người dùng đồ họa Windows. Windows NT dùng cùng giao diện đồ họa như Windows nhưng nó có khả năng đa nhiệm mạnh và khả năng quản lý bộ nhớ. Windows NT có thể hỗ trợ cho các phần mềm hiện có được viết cho DOS và nó có thể cung cấp khả năng tính toán tựa máy tính lớn cho các ứng dụng mới với yêu cầu bộ nhớ và tệp lớn. Nó có thể đánh địa chỉ theo các chùm 32 bit nếu cần và thậm chí có thể hỗ trợ cho đa xử lý với nhiều CPU. Khác với OS/2, Windows NT không bị gán với phần cứng máy tính dựa trên bộ vi xử lý của Intel. Nó có thể chạy trên các máy vi tính và trạm làm việc dùng các bộ vi xử lý của Mips Computer Systems Inc. hay chip Alpha mới của DEC.

UNIX đã được phát triển tại Bell Laboratories vào năm 1969 để giúp các nhà nghiên cứu khoa học dùng chung dữ liệu và chương trình trong khi vẫn giữ các thông tin khác riêng tư. Nó là hệ điều hành tương tác, đa người dùng, đa nhiệm. Nhiều người có thể dùng UNIX đồng thời để thực hiện cùng kiểu nhiệm vụ, hay một người dùng có thể chạy nhiều nhiệm vụ trên UNIX theo kiểu tương tranh. UNIX đã được phát triển để nối nhiều máy khác nhau lại và hỗ trợ cao cho truyền thông và kết mạng.

UNIX ban đầu được thiết kế cho máy mini nhưng bây giờ có phiên bản cho máy vi tính và máy tính lớn. UNIX có thể chạy trên nhiều loại loại máy tính khác nhau và có thể dễ dàng được chuyên biệt hoá. Nó cũng có thể lưu trữ và quản lí một số lớn các tệp. Hiện tại, UNIX chủ yếu được dùng cho trạm làm việc, máy tính mini và các môi trường đa người dùng không đắt trong các nghiệp vụ nhỏ, nhưng việc dùng của nó trong các nghiệp vụ lớn đang tăng lên bởi vì tính độc lập máy của nó. Các chương trình ứng dụng chạy dưới UNIX có thể được chuyển từ máy nọ sang máy kia với ít thay đổi.

UNIX bị buộc tội là không thân thiện với người dùng. Nó mạnh nhưng rất phức tạp. Nó có những chỉ lệnh, một số trong chúng rất bí ẩn và cộc lốc. Một lỗi gõ dòng lệnh cũng có thể dễ dàng xoá đi các tệp quan trọng. UNIX không thể đáp ứng tốt với các vấn đề gây ra bởi việc lạm dụng tài nguyên hệ thống như công việc và không gian đĩa. UNIX cũng đặt ra một số vấn đề an toàn, vì đa việc và đa người dùng có thể truy nhập vào cùng tệp đồng thời. Cuối cùng UNIX đòi hỏi khối lượng truy nhập bộ nhớ và đĩa lớn, giới hạn sự hữu dụng của nó cho các máy vi tính kém mạnh hơn.

Linux là một hệ điều hành được giới đại học phát triển từ việc viết lại và cho không mã nguồn của hệ điều hành UNIX. Linux hoàn toàn mang tất cả những ưu điểm của UNIX nhưng đồng thời đang trở thành một hệ điều hành được ngày càng nhiều người chú ý vì tính mở của nó để đối lập lại tính đóng kín của hệ điều hành Windows. Linux hiện nay được gắn liền với việc phát triển kết cụm máy tính và xử lí song song trên cơ sở nhiều bộ xử lí.

System 7, nay đã có **System 10**, là phần mềm hệ thống cho máy Macintosh, mang tính năng đa nhiệm cũng như các khả năng đồ hoạ mạnh và giao diện đồ hoạ người dùng điều khiển bằng chuột. Một mở rộng của hệ điều hành này được gọi là QuickTime cho phép người dùng Macintosh tích hợp các mảnh video, âm thanh stereo và các đoạn hoạt hình với phần mềm đồ hoạ và văn bản qui ước. System 7 có một số tính năng làm cho nó thành hấp dẫn cho

các ứng dụng toàn cầu. Chẳng hạn, nó cung cấp sự hỗ trợ mức hệ thống cho các ngôn ngữ Châu Á với tập kí tự lớn.

Giao diện đồ hoạ

Bất kì khi nào người dùng tương tác với máy tính, cho dù là máy vi tính, thì tương tác đều được kiểm soát bởi hệ điều hành. Giao diện người dùng là một phần của hệ thống tin mà người dùng tương tác với. Người dùng trao đổi với hệ điều hành qua giao diện người dùng của hệ điều hành đó. Các hệ điều hành máy vi tính ban đầu được điều khiển theo dòng lệnh, nhưng giao diện đồ hoạ, đầu tiên được hệ điều hành Macintosh phổ biến, sử dụng rất nhiều các icon - biểu tượng, nút, thanh và hộp để thực hiện cùng nhiệm vụ. Nó bắt đầu trở thành mô hình thống trị cho giao diện người dùng của hệ điều hành máy vi tính.

Những người ủng hộ giao diện đồ hoạ nói rằng các giao diện người dùng đồ hoạ tiết kiệm thời gian học bởi vì người mới học máy tính không phải học các chỉ lệnh bí mật cho từng ứng dụng. Các chức năng chung như lấy lời giúp đỡ, ghi tệp, hay in ra được thực hiện theo cùng cách qua tất cả các chương trình. Một chuỗi các chỉ lệnh phức tạp có thể được ban ra chỉ đơn giản bằng việc nhấn vào biểu tượng. Các chỉ lệnh đã được chuẩn hoá từ chương trình này sang chương trình khác, cho nên việc dùng chương trình mới thường là có thể được mà không cần huấn luyện gì thêm hay dùng tài liệu tham khảo. Chẳng hạn, các bước trong việc in thư do chương trình xử lí văn bản tạo ra hay một báo cáo tài chính do chương trình trang tính tạo ra nên là như nhau. Giao diện người dùng đồ hoạ cũng giúp thúc đẩy màn hình tốt và in ra - đặt mọi sự được truyền thông qua đồ hoạ.

Windows của Microsoft là lớp vỏ giao diện người dùng đồ hoạ phổ biến cao chạy nối tiếp với hệ điều hành DOS. Windows hỗ trợ cho đa nhiệm có giới hạn và một số dạng kết mạng nhưng có chung những giới hạn bộ nhớ của DOS. Nó bị coi là chạy không

hiệu quả máy trong môi trường đa nhiệm. Các phiên bản đầu của Windows có một số vấn đề với ứng dụng bị hỏng khi nhiều chương trình tranh nhau cùng không gian bộ nhớ. OS/2 có giao diện người dùng đồ họa riêng của nó được gọi là Lớp vỏ chỗ làm việc, giống với giao diện người dùng đồ họa cho máy Macintosh. Nó cung cấp cho người dùng một giao diện người dùng đồ họa nhất quán qua các ứng dụng. OS/2 hỗ trợ cho các ứng dụng DOS và có thể chạy các ứng dụng Windows và DOS đồng thời trong các cửa sổ chỉnh kích cỡ được.

Chọn hệ điều hành máy vi tính

Tổ chức nên chọn hệ điều hành cho các ứng dụng dựa trên máy vi tính của mình như thế nào? Quyết định này nên được đưa ra trên cơ sở lợi ích kỹ thuật nào? Tổ chức có nên nhìn vào các vấn đề khác như dễ sử dụng, huấn luyện, và giá của phần cứng và phần mềm trong việc dùng hệ điều hành không? Tổng quan ngắn gọn này gợi ý rằng có nhiều nhân tố cần tính tới.

Nếu tổ chức muốn có một hệ điều hành cho những ứng dụng nghiệp vụ chính thì nó cần hệ điều hành tương hợp với phần mềm mà những ứng dụng đó cần tới. Hệ điều hành nên dễ dùng và thiết đặt. Các tính năng giao diện người dùng của hệ điều hành nên dễ học. Những ứng dụng chủ chốt có các yêu cầu hệ điều hành đặc biệt, vì nghiệp vụ phụ thuộc vào chúng cho sự làm việc liên tục và sống còn. Với những ứng dụng như vậy, một hệ điều hành cung cấp sự hỗ trợ tin cậy cho đa nhiệm và quản lý bộ nhớ là điều bản chất. Hệ điều hành nên có khả năng chạy nhiều ứng dụng một cách nhanh chóng mà không gây sập hệ thống bởi vì các ứng dụng tranh giành cùng không gian bộ nhớ. Các ứng dụng chủ chốt điển hình có khối lượng giao tác lớn phải xử lý và yêu cầu hệ điều hành có thể giải quyết các chương trình phần mềm lớn phức tạp và khối lượng tệp lớn.

Khi lựa một hệ điều hành máy vi tính, vài câu hỏi chốt cần phải hỏi là:

- Phần mềm ứng dụng nào chạy trên hệ điều hành?
- Loại phần cứng máy tính nào mà hệ điều hành chạy trên đó?
- Hệ điều hành chạy nhanh như thế nào?
- Hệ điều hành này dễ học và dùng như thế nào?
- Hệ điều hành này được thiết kế cho một người dùng hay cho nhiều người dùng trên mạng?
- Hệ điều hành này có khả năng đa nhiệm mạnh không?
- Hệ điều hành này có độ tin cậy thế nào?
- Tốn bao nhiêu tiền để thiết đặt hệ điều hành?
- Sự hỗ trợ và trợ giúp kỹ thuật nào là cần có để thiết lập và cho chạy hệ điều hành? Sự hỗ trợ này có sẵn ở đâu?

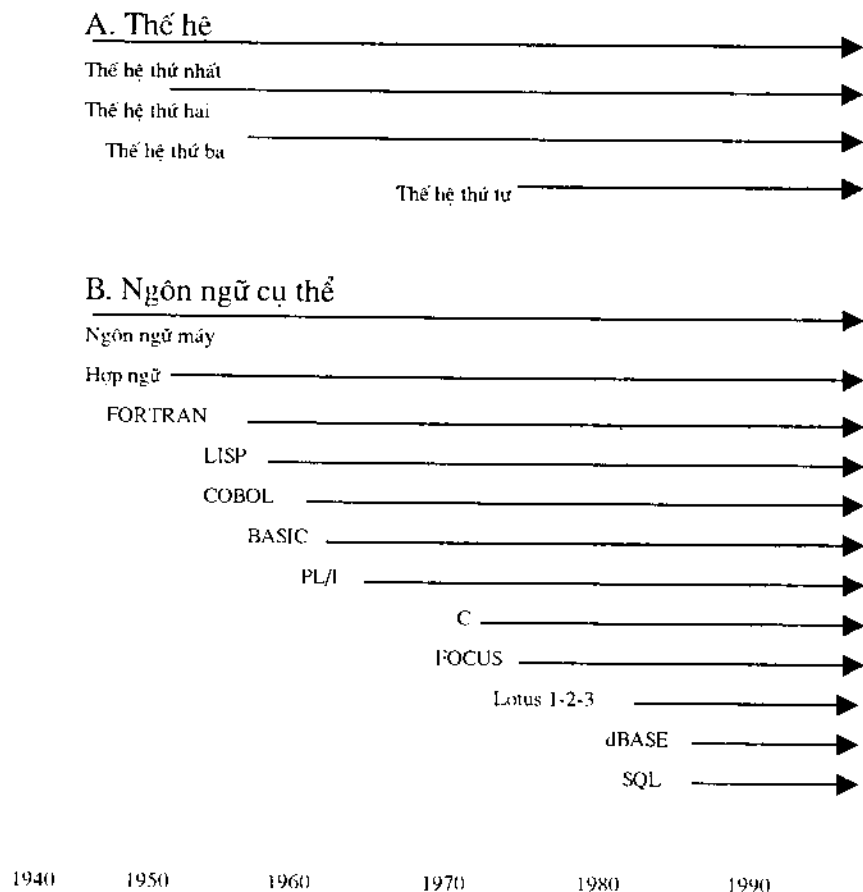
2.2.3 PHẦN MỀM ỨNG DỤNG

Phần mềm ứng dụng chủ yếu liên quan tới việc hoàn thành các nhiệm vụ của người dùng cuối. Nhiều ngôn ngữ lập trình khác nhau có thể được dùng để phát triển phần mềm ứng dụng. Mỗi ngôn ngữ đều có điểm mạnh và điểm yếu khác nhau.

Các thế hệ ngôn ngữ lập trình

Để trao đổi với thế hệ máy tính thứ nhất, người lập trình phải viết các chương trình trong **ngôn ngữ máy** - các số 0 và 1 theo mã nhị phân. Người dùng cuối cùng, người cần các ứng dụng phải làm việc với những người lập trình chuyên dụng, người có thể hiểu, nghĩ và làm việc trực tiếp trong ngôn ngữ máy của một máy tính đặc thù. Việc lập trình theo 0 và 1, đưa mọi câu lệnh như cộng, trừ,

và chia thành một chuỗi các số 0 và 1, đã làm việc lập trình thời kì đầu thành tiến trình chậm chạp rất vất vả.



Hình 2.10 Các thế hệ ngôn ngữ lập trình.

Khi phần cứng máy tính được cải tiến và tốc độ xử lý cùng kích cỡ bộ nhớ tăng lên, thì ngôn ngữ máy tính đã chuyển đổi từ ngôn ngữ máy sang ngôn ngữ dễ hơn cho con người hiểu. Các thế hệ ngôn ngữ lập trình đã phát triển tương ứng với các thế hệ phần

cứng máy tính. Hình 2.10 vẽ ra sự phát triển của các ngôn ngữ lập trình trong suốt 50 năm qua khi khả năng phần cứng đã tăng lên.

Ngôn ngữ máy là ngôn ngữ lập trình thế hệ thứ nhất. Thế hệ thứ hai của các ngôn ngữ lập trình xuất hiện vào đầu những năm 1950 với việc phát triển hợp ngữ. Thay vì dùng các số 0 và 1, người lập trình bây giờ có thể thể vào các chữ viết tắt và từ giống ngôn ngữ như add (cộng), sub (trừ) và load vào trong các câu lệnh lập trình. Một bộ dịch ngôn ngữ được gọi là trình biên dịch chuyển các câu lệnh như trong tiếng Anh vào ngôn ngữ máy.

Khi thế hệ phần cứng thứ ba đang được phát triển thì các ngôn ngữ lập trình cũng đi vào thế hệ thứ ba. Từ giữa những năm 1950 cho đến những năm 1970, các ngôn ngữ cấp cao hơn đầu tiên đã nổi lên. Các ngôn ngữ này cho phép các nhà toán học lần đầu tiên làm việc với máy tính qua việc dùng các ngôn ngữ như FORTRAN (FORMula TRANslator). Các nhà toán học bây giờ có khả năng định nghĩa các biến bằng câu lệnh như $Z = A + B$. Phần mềm dịch các định nghĩa và phát biểu toán học này thành một chuỗi các số 0 và 1. COBOL (COMmon Business Oriented Language) cho phép người lập trình dùng các câu tiếng Anh như print và sort, người ta không phải nghĩ dưới dạng 0 và 1.

Ngôn ngữ cấp cao được gọi như vậy bởi vì mỗi câu lệnh trong COBOL, hay FORTRAN đều sinh ra nhiều câu lệnh tại mức mã máy. Việc dùng các ngôn ngữ cấp cao này đòi hỏi trình biên dịch nhanh hơn, hiệu quả hơn để dịch các ngôn ngữ cấp cao thành mã máy.

Các ngôn ngữ máy tính thế hệ bốn nổi lên vào cuối những năm 1970, và sự phát triển của chúng vẫn đang tiếp diễn. Các ngôn ngữ này làm giảm thời gian lập trình cực kì lớn và cho nhiệm vụ phần mềm thành dễ dàng đến mức người dùng không có kĩ thuật về máy tính vẫn có thể phát triển các ứng dụng mà không cần sự trợ giúp của người lập trình chuyên nghiệp. Các dụng cụ thế hệ bốn cũng bao gồm các gói phần mềm ứng dụng viết sẵn mà có thể được

người dùng cuối cùng sử dụng một cách trực tiếp. Chẳng hạn việc dùng gói phần mềm Lotus 1-2-3, người dùng có thể tạo ra trang tính tài chính riêng của mình và thao tác dữ liệu mà không có sự can thiệp của người lập trình.

Tất nhiên có mối quan hệ rõ ràng giữa việc tăng khả năng của phần cứng và phần mềm. Mỗi thế hệ phần mềm mới lại đòi hỏi ngày càng nhiều vùng nhớ chính, trình biên dịch nhanh hơn, và bộ nhớ phụ lớn hơn. Các chương trình ngôn ngữ cấp cao đòi hỏi khối lượng bộ nhớ khổng lồ. Chẳng hạn, để nạp bộ chương trình SPSS (Statistic Package for the Social Science), một mình chương trình này có thể chiếm hơn 200Kb bộ nhớ chính.

Các ngôn ngữ lập trình phổ biến

Phần lớn những người quản lý không cần phải là người lập trình chuyên gia, nhưng họ nên hiểu cách đánh giá các ứng dụng phần mềm và lựa chọn ngôn ngữ lập trình thích hợp cho các mục tiêu tổ chức của họ. Chúng ta bây giờ sẽ mô tả ngắn gọn các ngôn ngữ cấp cao phổ biến hơn và những ngôn ngữ cấp rất cao.

HỢP NGỮ

Nhiều người lập trình vẫn còn ưa thích viết chương trình trong hợp ngữ bởi vì ngôn ngữ này cho họ việc kiểm soát chặt chẽ đối với phần cứng và thực hiện rất hiệu quả. Giống như ngôn ngữ máy, hợp ngữ được thiết kế cho máy riêng biệt và bộ xử lý riêng biệt. Chẳng hạn, có hợp ngữ riêng cho chip Intel 80386 được dùng trong máy vi tính IBM. Nói chung có tương ứng một một giữa ngôn ngữ máy và hợp ngữ. Mỗi phép toán trong hợp ngữ đều tương ứng với một phép toán của máy. Mặt khác, hợp ngữ có dùng một số kí hiệu tượng trưng và gán địa chỉ và vị trí bộ nhớ một cách tự động. Trong khi hợp ngữ cho người lập trình việc kiểm soát lớn lao, thì nó gây tốn kém dưới dạng thời gian của người lập trình, khó đọc và

khó gỡ lỗi, và khó học. Hợp ngữ ngày nay được dùng chủ yếu trong phần mềm hệ thống.

FORTRAN

FORTRAN (FORMula TRANslator) đã được phát triển từ năm 1956 để cung cấp một cách dễ dàng hơn cho việc viết các ứng dụng khoa học và kỹ nghệ. FORTRAN đặc biệt có ích trong xử lý dữ liệu số. Nhiều loại ứng dụng kinh doanh có thể được viết trong FORTRAN, nó tương đối dễ học, và các phiên bản hiện đại (như FORTRAN 77) cung cấp các cấu trúc phức tạp để kiểm soát logic chương trình. FORTRAN thì không tốt trong việc cung cấp sự hiệu quả đưa vào/đưa ra hay trong việc in ấn và làm việc với danh sách. Cú pháp thì rất nghiêm ngặt và các lỗi chủ chốt là phổ biến, làm cho chương trình khó gỡ lỗi.

COBOL

COBOL (COmmon Business Oriented Language) đi vào sử dụng vào đầu những năm 1960, Nó ban đầu được phát triển bởi vì Bộ Quốc phòng Mỹ muốn tạo ra một ngôn ngữ hành chính chung cho các phần mềm bên trong và bên ngoài. COBOL đã được thiết kế với việc quản trị kinh doanh trong đầu, để xử lý các tệp dữ liệu lớn với các ký tự chữ số (trộn lẫn chữ và dữ liệu số) và để thực hiện các nhiệm vụ lặp lại như tính lương. Cấu trúc dữ liệu chủ yếu của nó là bản ghi, tệp, bảng và danh sách. COBOL dễ học cho người phân tích nghiệp vụ. Như một ngôn ngữ lập trình được dùng rộng rãi nhất, nó được nhiều nhóm bên ngoài hỗ trợ, và có khá nhiều sự trợ giúp có hiệu suất. Điểm yếu của COBOL là do cái tốt của nó. Nó nghèo nàn về các tính toán toán học phức tạp. Có nhiều phiên bản COBOL và không phải tất cả đều tương hợp với nhau. Chương trình COBOL dài dòng - dài đến hàng trăm nghìn dòng với chương trình tính lương chính, chẳng hạn - có thể trở thành phức tạp đến mức thực tế không thể nào hiểu nổi.

BASIC

BASIC (Beginner All-purposr Symbolic Instruction Code) đã được John Kemeny và Thomas Kurtz phát triển từ năm 1964 để dạy cho sinh viên tại đại học Dartmouth College cách dùng máy tính. Ngày nay nó là ngôn ngữ lập trình phổ biến nhất cho các khu đại học và về máy vi tính. BASIC có thể làm hầu hết các nhiệm vụ xử lý máy tính từ quản lý kho cho tới các tính toán toán học. Nó dễ dùng, biểu lộ tốt khả năng của máy tính, và chỉ yêu cầu bộ thông dịch nhỏ. Nhược điểm của BASIC là ở chỗ nó quá cố thực hiện tốt vài nhiệm vụ mặc dầu nó không làm tốt tất cả. Nó không có cấu trúc điều khiển logic chương trình hay cấu trúc dữ liệu phức tạp, mà điều này thì làm cho nó thành khó dùng trong việc dạy thực hành lập trình tốt. Trong khi BASIC chỉ có vài chỉ lệnh và dễ học, thì các phiên bản kế sau của ngôn ngữ này đã cố gắng bổ sung vào cú pháp ban đầu của nó các phiên bản mới của BASIC không tương hợp với phiên bản cũ. Do đó, các chương trình BASIC thường không thể nào chuyển được từ máy nọ sang máy kia.

PL/I

PL/I (Programming Language 1) đã được IBM phát triển vào năm 1964. Nó là ngôn ngữ lập trình vạn năng mạnh nhất bởi vì nó có thể giải quyết các ứng dụng toán học và kinh doanh một cách dễ dàng, có hiệu quả cao trong các hoạt động vào/ra, và có thể giải quyết khối lượng dữ liệu lớn.

Điều không may là do khối lượng khổng lồ của các chương trình COBOL và FORTRAN đã được viết trong khu vực tư nhân với chi phí lớn không thể đơn giản bị vứt bỏ khi một ngôn ngữ mới, mạnh hơn ra đời. Người ta ước lượng đã có đến 12 tỉ dòng lệnh mã COBOL dùng trong sản xuất ở Mỹ. Điều này biểu thị cho một đầu tư trên 52 nghìn tỉ đô la. PL/I đã không thành công phần lớn bởi vì người lập trình được huấn luyện trong COBOL không thể bị thuyết phục để học một ngôn ngữ hoàn toàn mới; và các tổ chức kinh

doanh không thể bị thuyết phục để chi hàng triệu đô la cho việc viết lại phần mềm của mình. Tuy nhiên PL/I là cái gì đó khó học về toàn bộ của nó.

PASCAL

Được mang tên Blaise Pascal, nhà toán học và triết học thế kỷ mười bảy, **Pascal** đã được giáo sư tin học người Thụy Sĩ Niklaus Wirth tại Zurich phát triển vào cuối những năm 1960. Các chương trình Pascal có thể được biên dịch bằng việc dùng bộ nhớ máy tính tối thiểu, cho nên chúng có thể được dùng trong máy vi tính. Với các cấu trúc phức tạp để điều khiển logic chương trình và một tập đơn giản nhưng mạnh các chỉ lệnh, Pascal được dùng chủ yếu trong các học trình tin học để dạy việc thực hành lập trình tốt. Ngôn ngữ này yếu trong việc giải quyết với tệp và vào/ra và không dễ cho người bắt đầu dùng.

ADA

Ada được phát triển vào năm 1980 để cung cấp cho Bộ Quốc phòng Mỹ ngôn ngữ lập trình có cấu trúc lấy làm chuẩn cho tất cả các ứng dụng của nó. Ngôn ngữ này ban đầu được quan niệm dùng cho các hệ thống vũ khí nơi phần mềm được phát triển trên một bộ xử lý và rồi được cấy vào trong vũ khí. Nó đã được thiết kế tường minh để cho nó có thể thực hiện đều trong các môi trường phần cứng đa dạng. Ngôn ngữ này cũng thúc đẩy thiết kế phần mềm có cấu trúc. Các chuyên gia của chính phủ Mỹ hi vọng Ada sẽ tạo ra phần mềm với chi phí-hiệu quả tốt hơn bởi vì nó làm thuận tiện cho mã có cấu trúc rõ ràng hơn COBOL.

Nghị viện đã thông qua điều luật bắt buộc dùng Ada cho tất cả mọi việc phát triển hệ thống cho Bộ Quốc phòng, và Ada được dùng trong cả những ứng dụng công quyền phi quân sự nữa. Ngôn ngữ này cũng có thể được dùng cho các ứng dụng kinh doanh nói chung vì nó có thể làm việc trên máy vi tính và là khả chuyển qua

các nhãn hiệu phần cứng máy tính khác nhau. Liệu Ada có nổi lên như một chuẩn phần mềm cho thế giới kinh doanh không? Nhiều tổ chức không nghĩ như vậy, bởi cùng lí do họ đã không chào đón PL/I. Họ không bị thuyết phục rằng đáng phải đầu tư vào nó và sẽ bị mạo hiểm để bỏ COBOL xem như một chuẩn kinh doanh.

C

C đã được phát triển tại Bell Labs của AT&T vào đầu những năm 1970 và là ngôn ngữ mà trong đó nhiều phần của hệ điều hành UNIX được viết ra. C tổ hợp việc kiểm soát chặt và tính năng thực hiện hiệu quả của hợp ngữ với tính khả chuyển máy. Nói cách khác, nó có thể làm việc trên nhiều máy tính thay vì chỉ một máy. Nhiều phần mềm máy vi tính thương mại đã được viết trong C, nhưng C đang thu được sự hỗ trợ cho các ứng dụng máy tính mini và máy tính lớn. C không thể đánh bật được COBOL đối với các ứng dụng kinh doanh trên máy lớn, nhưng nó sẽ được dùng ngày càng tăng cho phần mềm máy vi tính thương mại và cho các ứng dụng khoa học và kĩ thuật.

LISP VÀ PROLOG

LISP (viết tắt cho LISt Processor) và Prolog được dùng rộng rãi trong trí năng nhân tạo. LISP được tạo ra vào cuối những năm 1950 nhờ nhà toán học John Mc-Carthy tại MIT và có khuynh hướng đặt các kí hiệu như phép toán, biến và giá trị dữ liệu vào các danh sách có nghĩa. LISP là tốt trong việc thao tác kí hiệu hơn là việc nghiên cứu thông thường.

Prolog, được đưa ra quãng năm 1970, cũng rất thích hợp cho thao tác kí hiệu và có thể chạy trên các máy tính vạn năng, trong khi LISP thường chạy tốt nhất trên các máy có cấu hình đặc biệt để chạy chương trình LISP.

2.2.4 CÁC DỤNG CỤ PHẦN MỀM MỚI VÀ CÁCH TIẾP CẬN

Ngôn ngữ thế hệ bốn bao gồm rất nhiều dụng cụ phần mềm làm cho người dùng cuối có khả năng phát triển các ứng dụng phần mềm với sự trợ giúp kỹ thuật tối thiểu hay không có sự trợ giúp nào cả, làm tăng hiệu suất của người lập trình chuyên nghiệp. Các ngôn ngữ thế hệ bốn có khuynh hướng là phi thủ tục hay ít tính thủ tục hơn các ngôn ngữ lập trình quy ước. Các ngôn ngữ thủ tục yêu cầu đặc tả về trình tự các bước, hay thủ tục, vốn bảo cho máy tính điều phải làm và cách làm nó. Các ngôn ngữ phi thủ tục chỉ cần xác định cái gì cần phải được thực hiện thay vì đưa ra chi tiết về cách tiến hành nhiệm vụ. Vậy, ngôn ngữ phi thủ tục có thể thực hiện cùng nhiệm vụ với số bước ít hơn và số dòng chương trình ít hơn là ngôn ngữ thủ tục.

Ngôn ngữ thế hệ bốn

Có nhiều loại ngôn ngữ thế hệ bốn: ngôn ngữ vấn đáp, bộ sinh báo cáo, ngôn ngữ đồ họa, bộ sinh ứng dụng, gói phần mềm ứng dụng, các ngôn ngữ lập trình cấp rất cao, và các dụng cụ máy vi tính. Hình 2.11 minh họa cho phổ các dụng cụ này và một số sản phẩm có sẵn trong thương mại cho mỗi loại.

NGÔN NGỮ VẤN ĐÁP

Ngôn ngữ vấn đáp là các ngôn ngữ cấp cao cho việc lục tìm dữ liệu được lưu giữ trong các cơ sở dữ liệu hay các tệp. Chúng thường là tương tác, trực tuyến và có khả năng hỗ trợ cho các yêu cầu thông tin vốn không được xác định trước. Chúng thường được gắn với hệ quản trị cơ sở dữ liệu và dụng cụ máy vi tính. Ngôn ngữ vấn đáp có thể dò tìm cơ sở dữ liệu hay tệp, dùng các tiêu chí tuyển lựa đơn giản hay phức tạp để hiển thị thông tin có liên quan tới

nhiều bản ghi. Các dụng cụ ngôn ngữ vấn đáp có sẵn mang nhiều loại cú pháp và cấu trúc khác nhau, một số gần với ngôn ngữ tự nhiên hơn số khác. Một số hỗ trợ cho việc cập nhật dữ liệu cũng như dò tìm. Một thí dụ về câu hỏi cơ sở là "Hãy liệt kê tất cả các nhân viên trong phòng tính lương."

hướng người sử dụng cuối			hướng người chuyên môn		
Dụng cụ máy vi tính	Ngôn ngữ vấn đáp/ sinh cáo	Ngôn ngữ đồ hoạ	Bộ sinh ứng dụng	Gói phần ứng dụng	Ngôn ngữ lập trình cấp rất cao
Lotus 1-2-3 dBASE IV Low Charting II+	Easytrieve Intellect Query-By-Example SQL RPG-III Inquire	Harvard Graphic SAS Graph Systat	FOCUS DMS SAS Mapper Ideal Natural CSP	PeopleSoft Human Resource CA-Warehouse BOSS AVP Sales/Use Tax AMAPS	APL Nomad

Hình 2.11 Ngôn ngữ thế hệ bốn.

BỘ SINH BÁO CÁO

Bộ sinh báo cáo là tiện nghi để tạo ra các báo cáo chuyên biệt hoá. Chúng trích dữ liệu từ tệp hay cơ sở dữ liệu và tạo ra báo cáo theo nhiều định dạng. Bộ sinh báo cáo nói chung cung cấp nhiều điều khiển trên cách dữ liệu được định dạng, tổ chức và hiển thị hơn là ngôn ngữ vấn đáp. Bộ sinh báo cáo mạnh có thể thao tác dữ liệu với các tính toán và logic phức tạp trước khi chúng được đưa ra. Một số bộ sinh báo cáo là việc mở rộng của cơ sở dữ liệu hay ngôn ngữ vấn đáp. Bộ sinh báo cáo càng mạnh và phức tạp thì

không thể thích hợp cho người dùng cuối mà không có sự trợ giúp từ các chuyên gia hệ thống tin chuyên nghiệp.

NGÔN NGỮ ĐỒ HOẠ

Ngôn ngữ đồ hoạ dò tìm dữ liệu từ các tệp và cơ sở dữ liệu rồi hiển thị chúng dưới định dạng đồ hoạ. Người dùng có thể yêu cầu dữ liệu và xác định cách chúng được vẽ sơ đồ. Một số phần mềm đồ hoạ có thể thực hiện các phép toán số học và logic trên dữ liệu nữa. SAS, Harvard Graphics và Graph Plan là các dụng cụ đồ hoạ phổ biến.

BỘ SINH ỨNG DỤNG

Bộ sinh ứng dụng chứa các modul đã được lập trình sẵn mà có thể sinh ra toàn bộ ứng dụng, phát triển với tốc độ lớn. Người sử dụng có thể xác định điều gì cần được làm và bộ sinh ứng dụng sẽ tạo ra mã thích hợp cho việc đưa vào, làm hợp lệ, cập nhật, xử lý và làm báo cáo. Phần lớn các bộ sinh ứng dụng đầy đủ chức năng đều bao gồm một tập tích hợp, toàn diện các dụng cụ phát triển: hệ quản trị cơ sở dữ liệu, từ điển dữ liệu, ngôn ngữ vấn đáp, bộ vẽ màn hình, bộ sinh đồ hoạ, bộ sinh báo cáo, các dụng cụ mô hình hoá/hỗ trợ quyết định, tiện nghi an toàn, và ngôn ngữ lập trình cấp cao. Với các yêu cầu duy nhất mà không thể được đáp ứng bằng các modul tổng quát, thì phần lớn các bộ sinh ứng dụng đều chứa *lối ra cho người dùng*, nơi các trình đã được lập trình để làm việc chuyên biệt hoá có thể được chèn thêm vào. Một số bộ sinh ứng dụng là tương tác, làm cho người dùng có khả năng ngồi tại thiết bị cuối để định nghĩa cái vào, tệp, xử lý và báo cáo bằng việc đáp ứng cho các câu hỏi một cách trực tuyến.

CÁC NGÔN NGỮ LẬP TRÌNH CẤP RẤT CAO

Các ngôn ngữ lập trình cấp rất cao được thiết kế để sinh ra mã chương trình với ít lệnh hơn là các ngôn ngữ qui ước như

PHẦN MỀM XỬ LÝ VĂN BẢN

Phần mềm xử lý văn bản lưu giữ dữ liệu văn bản theo cách điện tử như tệp máy tính chứ không trên giấy. Phần mềm xử lý văn bản cho phép người dùng tạo những thay đổi trong tài liệu một cách điện tử trong bộ nhớ. Việc này khử bỏ đi nhu cầu gõ lại toàn bộ trang để đưa thêm vào phần sửa chữa. Phần mềm này có các tùy chọn định dạng để tạo ra thay đổi trong khoảng cách dòng, lề, kích cỡ kí tự và chiều rộng cột. Microsoft Word và WordPerfect là các gói phần mềm xử lý văn bản phổ biến.

Phần lớn các phần mềm xử lý văn bản đều nâng cao các tính năng tự động hoá các nhiệm vụ viết khác: bộ kiểm tra chính tả, bộ kiểm tra phong cách (để phân tích văn phạm và ngắt câu), chương trình cho từ đồng nghĩa, và chương trình gửi thư cho nhiều người (thường là móc nối thư hay văn bản khác với các tên và địa chỉ trong danh sách gửi thư).

TRANG TÍNH

Phần mềm **trang tính** điện tử cung cấp phiên bản máy tính hoá của dụng cụ mô hình hoá tài chính truyền thống như tập giấy kẻ ô của kế toán viên, bút chì và máy tính tay. Trang tính điện tử được tổ chức thành lưới các cột và hàng. Sức mạnh của trang tính điện tử là hiển nhiên khi người ta thay đổi một hay nhiều giá trị, bởi vì tất cả các giá trị có liên quan trên trang tính sẽ được tự động tính toán lại.

Trang tính là có giá trị cho các ứng dụng nơi mà các tính toán số với các mẫu dữ liệu phải có liên hệ lẫn nhau. Trang tính cũng có ích cho các ứng dụng yêu cầu mô hình hoá và phân tích "cái gì xảy ra nếu". Sau khi người dùng đã xây dựng một tập các mối quan hệ toán học, thì trang tính có thể được tính toán lại ngay lập tức bằng việc dùng tập các giả thuyết khác. Một số phương án có thể dễ dàng được tính bằng việc thay đổi một hay vài dữ liệu mà không phải gõ lại vào phần còn lại của trang tính. Nhiều gói trang tính bao

gồm cả các chức năng đồ họa mà có thể trình bày dữ liệu dưới dạng các đồ thị đường, đồ thị thanh hay đồ thị múi. Các gói trang tính phổ biến nhất là Lotus 1-2-3, Quattro Pro và Microsoft Excel.

PHẦN MỀM QUẢN TRỊ DỮ LIỆU

Trong khi các chương trình trang tính là những dụng cụ mạnh để thao tác dữ liệu định lượng, thì **phần mềm quản trị dữ liệu** là thích hợp hơn cho việc tạo ra và thao tác các danh sách và cho việc tổ hợp thông tin từ các tệp khác. Các gói quản trị cơ sở dữ liệu máy vi tính có những tính năng lập trình và đơn giản để học làm cho những người không chuyên môn cũng xây dựng được các hệ thống tin nhỏ.

Phần mềm quản trị dữ liệu về cơ bản có các tiện nghi để tạo ra tệp và cơ sở dữ liệu và để lưu giữ, sửa đổi và thao tác dữ liệu để làm ra báo cáo và trả lời câu hỏi. Phần mềm quản trị cơ sở dữ liệu phổ biến cho máy tính cá nhân bao gồm dBASE IV, dBASE III PLUS (R), Paradox, Rbase và Foxbase.

GÓI PHẦN MỀM TÍCH HỢP

Gói phần mềm tích hợp tổ hợp các chức năng của các gói phần mềm máy vi tính quan trọng nhất, như trang tính, xử lý văn bản, đồ họa và quản trị dữ liệu. Việc tích hợp này cung cấp một dụng cụ phần mềm vạn năng hơn và xoá bỏ việc vào dữ liệu dư thừa và việc bảo trì dữ liệu. Trong khi chúng có thể làm nhiều điều tốt, thì nói chung chúng không có cùng sức mạnh và chiều sâu như các gói chuyên dụng.

Lập trình hướng sự vật

Việc ùn đồng ngày càng tăng các dự án phần mềm và nhu cầu cho kinh doanh để làm cho hệ thống mềm dẻo và xây dựng nhanh chóng đã mở ra một cách tiếp cận mới tới việc phát triển phần mềm

qua các dụng cụ lập trình "hướng sự vật". Các khái niệm hướng sự vật quay trở ngược lại những năm 1970, nhưng chỉ mới gần đây các phần cứng khả năng cao và các phần mềm cửa sổ, đồ hoạ, biểu tượng đã làm cho chúng thành một giải pháp có thể đứng vững.

Các phương pháp phát triển phần mềm truyền thống đã đối xử dữ liệu và thủ tục như các cấu phần độc lập. Một thủ tục lập trình tách biệt phải được viết ra mọi lúc ai đó muốn tiến hành một hành động trên một mẫu dữ liệu đặc biệt. Thủ tục hành động trên dữ liệu mà chương trình truyền cho chúng.

CÁI GÌ LÀM CHO LẬP TRÌNH HƯỚNG SỰ VẬT THÀNH KHÁC BIỆT?

Lập trình hướng sự vật tổ hợp dữ liệu và các thủ tục đặc thù vốn tác động trên những dữ liệu đó vào trong một "sự vật". Sự vật tổ hợp dữ liệu và mã chương trình. Thay vì truyền dữ liệu cho thủ tục thì chương trình gửi một thông báo cho sự vật để thực hiện một thủ tục mà nó đã được cụ thể hoá trong đó. (Thủ tục còn được gọi là "phương pháp" theo ngôn ngữ hướng sự vật.) Cùng thông báo có thể được gửi cho nhiều sự vật khác nhau, nhưng mỗi sự vật sẽ thực hiện thông báo đó một cách khác nhau.

Chẳng hạn, một ứng dụng tài chính hướng sự vật có thể có các sự vật Khách hàng gửi các thông báo nợ và có tới các sự vật Tài khoản. Sự vật Tài khoản đến lượt nó lại có thể duy trì các sự vật Tiền-trong-tay, Tài-khoản-trả và Tài-khoản-nhận.

Dữ liệu của sự vật bị đầu kín với các phần khác của chương trình và chỉ có thể bị thao tác từ bên trong sự vật. Phương pháp để thao tác các dữ liệu của sự vật có thể bị thay đổi ở bên trong mà không ảnh hưởng tới các phần khác của chương trình. Người lập trình có thể lập trung vào điều họ muốn sự vật làm, và sự vật quyết định cách làm điều đó.

Bởi vì dữ liệu của sự vật được bao bọc tách khỏi các phần khác của hệ thống, nên từng sự vật là một khối xây dựng phần mềm

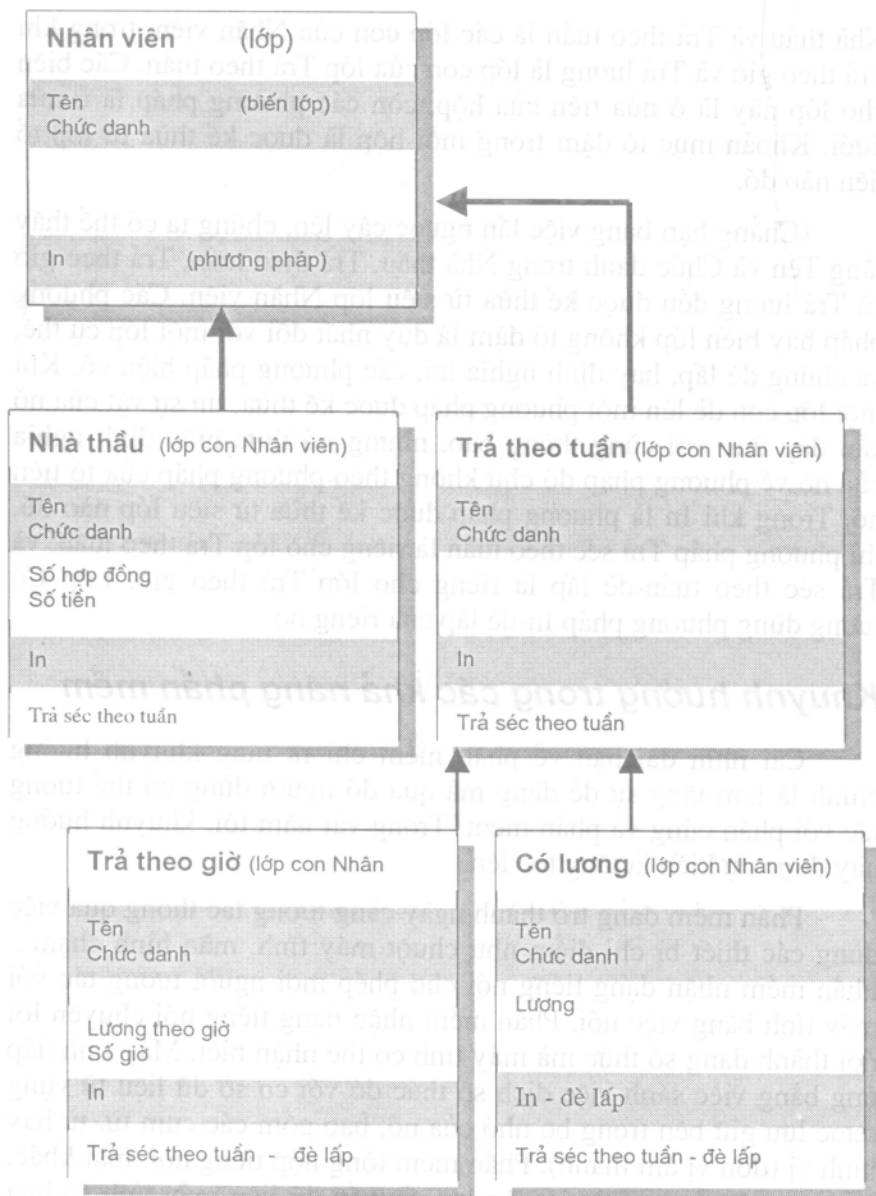
độc lập vốn có thể được dùng trong nhiều hệ thống khác nhau mà không làm thay đổi mã chương trình. Vậy, lập trình hướng sự vật được trông đợi để làm giảm thời gian và chi phí của việc viết phần mềm bằng việc tạo ra mã chương trình dùng lại được hay "chip" phần mềm vốn có thể được dùng lại trong các hệ thống có liên quan khác. Công việc phần mềm tương lai có thể rút ra từ thư viện các sự vật dùng lại được, và hiệu suất thu được từ công nghệ hướng sự vật có thể được khuếch đại lên nếu các sự vật được lưu giữ trong các thư viện phần mềm dùng lại được.

CÁC KHÁI NIỆM CỦA LẬP TRÌNH HƯỚNG SỰ VẬT

Lập trình hướng sự vật dựa trên các khái niệm về **lớp** và **kế thừa**. Mã chương trình không được viết một cách tách biệt cho mọi sự vật mà cho các lớp, hay các loại chung cho các sự vật tương tự. Sự vật thuộc vào một lớp nào đó có các tính năng của lớp đó. Các lớp sự vật đến lượt nó lại có thể kế thừa tất cả các cấu trúc và hành vi của một lớp tổng quát hơn và rồi bổ sung thêm các biến và hành vi duy nhất cho từng sự vật. Các lớp sự vật mới được tạo ra bằng việc chọn một lớp hiện có và xác định cách lớp mới này khác biệt với lớp hiện có, thay vì bắt đầu từ đầu mỗi lần.

Các lớp được tổ chức một cách có hệ thống cấp bậc thành siêu lớp và lớp con. Chẳng hạn, lớp "ô tô" có thể có lớp "xe cộ" là siêu lớp, cho nên nó sẽ kế thừa tất cả các phương pháp và dữ liệu được định nghĩa trước đó cho "xe cộ". Thiết kế của lớp ô tô sẽ chỉ cần mô tả cách ô tô khác với xe cộ. Một ứng dụng ngân hàng có thể định nghĩa một sự vật Tài-khoản-tiết-kiệm mà rất giống sự vật Tài-khoản-Ngân-hàng nhưng với vài khác biệt nhỏ. Tài-khoản-tiết-kiệm kế thừa tất cả các trạng thái và phương pháp của Tài-khoản-Ngân-hàng và rồi bổ sung thêm và điều phụ.

Chúng ta có thể thấy các lớp và kế thừa làm việc trong Hình 2.12, vốn minh họa cho cây các lớp liên quan tới nhân viên và cách họ được trả lương. Nhân viên là một tổ tiên chung của bốn lớp khác.



Hình 2.12 Lớp, lớp con, kế thừa và đề lấp.

Nhà thầu và Trả theo tuần là các lớp con của Nhân viên, trong khi Trả theo giờ và Trả lương là lớp con của lớp Trả theo tuần. Các biến cho lớp này là ở nửa trên của hộp, còn các phương pháp là ở nửa dưới. Khoản mục tô đậm trong mỗi hộp là được kế thừa từ lớp tổ tiên nào đó.

(Chẳng hạn bằng việc lần ngược cây lên, chúng ta có thể thấy rằng Tên và Chức danh trong Nhà thầu, Trả theo tuần, Trả theo giờ và Trả lương đều được kế thừa từ siêu lớp Nhân viên. Các phương pháp hay biến lớp không tô đậm là duy nhất đối với một lớp cụ thể, và chúng đề lập, hay định nghĩa lại, các phương pháp hiện có. Khi một lớp con đề lên một phương pháp được kế thừa, thì sự vật của nó vẫn đáp ứng với cùng thông báo, nhưng nó thực hiện định nghĩa của nó về phương pháp đó chứ không theo phương pháp của tổ tiên nó. Trong khi In là phương pháp được kế thừa từ siêu lớp nào đó, thì phương pháp Trả séc theo tuần là riêng cho lớp Trả theo tuần, và Trả séc theo tuần-đề lập là riêng cho lớp Trả theo giờ. Lớp Có lương dùng phương pháp In-đề lập của riêng nó

Khuynh hướng trong các khả năng phần mềm

Cái nhìn dài hạn về phần mềm chỉ ra rằng khuynh hướng chính là làm tăng sự dễ dàng mà qua đó người dùng có thể tương tác với phần cứng và phần mềm. Trong vài năm tới, khuynh hướng này được dự kiến là tăng tốc lên.

Phần mềm đang trở thành ngày càng tương tác thông qua việc dùng các thiết bị chỉ điểm như chuột máy tính, màn hình chạm... Phần mềm nhận dạng tiếng nói cho phép mọi người tương tác với máy tính bằng việc nói. Phần mềm nhận dạng tiếng nói chuyển lời nói thành dạng số thức mà máy tính có thể nhận biết. Máy tính đáp ứng bằng việc sánh việc dịch số thức đó với cơ sở dữ liệu từ vựng được lưu giữ bên trong bộ nhớ của nó, bao gồm các cụm từ, từ hay hình vị (đơn vị âm thanh). Phần mềm tổng hợp tiếng nói, mặt khác, lại làm việc theo cách đối lập lại, chuyển dữ liệu máy tính số hoá

trở lại thành lời nói. Các hệ thống nhận dạng tiếng nói và tổng hợp tiếng nói vẫn làm việc với từ vựng có giới hạn. Chúng đã được dùng chủ yếu cho các nhiệm vụ thường lệ, đơn giản trong việc kiểm soát dây chuyền sản xuất và tiếp thị từ xa dựa trên máy tính nhưng vẫn chưa đủ phức tạp đối với phần lớn các hệ thống thông tin văn phòng.

2.3 QUẢN LÝ TÀI NGUYÊN DỮ LIỆU

2.3.1 TỔ CHỨC DỮ LIỆU TRONG MÔI TRƯỜNG TẬP TRUYỀN THÔNG

Hệ thông tin có hiệu quả cung cấp cho người dùng những thông tin đúng lúc, chính xác và có liên quan. Thông tin này được lưu giữ trong các tệp máy tính. Khi các tệp được sắp xếp và duy trì gọn gàng thì người dùng có thể dễ dàng truy nhập và tìm kiếm thông tin mình cần. Các tệp được sắp xếp cẩn thận, được quản lý tốt thì làm dễ dàng cho việc có được dữ liệu cho các quyết định nghiệp vụ, trong khi các tệp bị quản lý dở lại dẫn tới hỗn loạn trong xử lý thông tin, chi phí cao, hiệu năng kém, và ít nếu không nói là không mềm dẻo. Mặc cho việc dùng phần cứng và phần mềm tốt đến đâu, nhiều tổ chức vẫn có hệ thông tin kém hiệu quả bởi vì việc quản lý tệp tồi.

Thuật ngữ và khái niệm tổ chức tệp

Hệ thống máy tính tổ chức dữ liệu theo hệ thống cấp bậc, bắt đầu từ bit và byte rồi tiến tới trường, bản ghi, tệp và cơ sở dữ liệu (xem Hình 2.13).

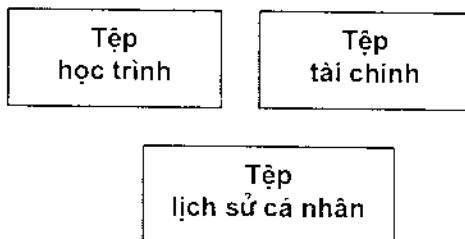
Bit biểu diễn cho đơn vị dữ liệu nhỏ nhất mà máy tính có thể xử lý. Một nhóm các bit, được gọi là byte, biểu diễn cho một kí tự, mà có thể là chữ, số hay kí hiệu khác. Một nhóm các kí tự trong một

Hệ thống cấp bậc

Thí dụ

Cơ sở dữ liệu sinh viên

Cơ sở dữ liệu



	Tập học trình			
	Tên	Học trình	Ngày	Bậc
Tập	NV A	Hệ TT 01	F20	B+
	Lê B.	Hệ TT 01	F20	A
	H.v. C	Hệ TT 01	F20	C
	Tên	Học trình	Ngày	Bậc
Bản ghi	NV A	Hệ TT 01	F20	B+
Trường	Nguyễn Văn A (trường Tên)			
Byte	1010 1010 (chữ J trong ASCII)			
Bit	0			

Hình 2.13 Hệ thống cấp bậc dữ liệu.

từ, một nhóm các từ, hay số hoàn chỉnh (như tên người hay tuổi), được gọi là một **trường**. Một nhóm các trường có liên quan tạo thành **bản ghi**; một nhóm các bản ghi thuộc cùng một kiểu được

gọi là **tệp**. Chẳng hạn, tất cả các bản ghi sinh viên trong Hình 2.13 có thể tạo nên một tệp học trình. Một nhóm các tệp có liên hệ tạo thành cơ sở dữ liệu. Tệp học trình sinh viên được minh hoạ trong Hình 2.13 có thể được nhóm với các tệp về lịch sử sinh viên và nền tảng tài chính để tạo ra cơ sở dữ liệu sinh viên.

Bản ghi mô tả cho một thực thể. **Thực thể** là một người, nơi chốn, vật, hay biến cố mà trên đó chúng ta duy trì thông tin. Đơn hàng là một thực thể điển hình trong tệp đơn hàng bán, vốn duy trì thông tin về các đơn bán hàng của công ti. Mỗi đặc trưng hay phẩm chất mô tả cho một thực thể đặc thù được gọi là một **thuộc tính**. Chẳng hạn, số hiệu đơn, ngày đơn, khối lượng đặt hàng, số hiệu khoản mục, và số lượng khoản mục sẽ là thuộc tính của thực thể *đơn hàng*. Các giá trị đặc biệt mà những thuộc tính này có thể có thì đều có thể tìm được trong các trường của bản ghi mô tả cho thực thể đơn hàng.

Mọi bản ghi trong một tệp đều cần phải chứa ít nhất một trường mà duy nhất xác định ra bản ghi đó để cho bản ghi đó có thể được dò tìm lại, được cập nhật hay sắp xếp. Trường định danh này được gọi là **trường khoá**. Thí dụ về trường khoá là số hiệu đơn hàng cho bản ghi đơn hàng hay số hiệu nhân viên hay số hiệu căn cước cho bản ghi nhan sự (chứa dữ liệu nhân viên như tên, tuổi, địa chỉ, nghề nghiệp v.v.v của nhân viên).

Truy nhập các bản ghi từ tệp máy tính

Hệ thống máy tính lưu giữ tệp trên thiết bị nhớ phụ. Các bản ghi có thể được bố trí theo nhiều cách trên thiết bị nhớ, và việc sắp xếp xác định cách thức để từng bản ghi có thể được truy nhập hay dò tìm. Một cách tổ chức các bản ghi là tuần tự. Trong **tổ chức tệp tuần tự**, các bản ghi dữ liệu phải được dò tìm theo cùng trình tự vật lí mà chúng đã được lưu giữ. Tương phản lại, **tổ chức tệp ngẫu nhiên** cho phép người dùng truy nhập các bản ghi theo bất kì trình

tự nào họ muốn, không để ý tới thứ tự vật lí thực tế trên phương tiện nhớ.

Tổ chức tệp tuần tự là cách tổ chức tệp duy nhất mà có thể được dùng trên băng từ. Phương pháp tổ chức tệp này không còn được phổ biến nữa, nhưng một số tổ chức vẫn dùng nó cho việc xử lí theo lô mà theo đó họ truy nhập và xử lí từng bản ghi một cách tuần tự.

Phương pháp truy nhập tệp trực tiếp được dùng với tổ chức tệp trực tiếp. Phương pháp này sử dụng trường khoá để định vị địa chỉ vật lí của bản ghi. Tuy nhiên, tiến trình này được thực hiện không có chỉ số. Thay vì thế, một công thức toán học gọi là **thuật toán biến đổi** được dùng để dịch trường khoá thẳng thành vị trí lưu giữ vật lí của bản ghi trên đĩa. Thuật toán này thực hiện tính toán toán học nào đó trên khoá của bản ghi, và kết quả của tính toán đó là địa chỉ vật lí của bản ghi.

Vấn đề đối với môi trường tệp truyền thống

Phần lớn các tổ chức bắt đầu việc xử lí thông tin trên qui mô nhỏ, tự động hoá một ứng dụng mỗi lúc. Hệ thống có khuynh hướng trở nên độc lập, và không tương ứng với kế hoạch lớn nào đó. Một cách điển hình, mỗi phòng ban của công ti nhiều phòng ban đều phát triển các ứng dụng riêng của mình. Bên trong mỗi phòng ban, mỗi vùng chức năng lại có khuynh hướng phát triển các hệ thống một cách cô lập với các vùng chức năng khác. Kế toán, tài chính, chế tạo, và tiếp thị tất cả đều phát triển các hệ thống và các tệp dữ liệu riêng của mình.

Tất nhiên, mỗi ứng dụng yêu cầu các tệp riêng của nó và chương trình máy tính riêng của nó để vận hành. Nói chung, các tệp được dùng trong một ứng dụng đều là phiên bản của một tệp chủ cho vùng chức năng. Chẳng hạn, có một tệp nhân sự chủ rất lớn chứa hầu hết các thông tin cơ bản về mọi nhân viên trong tổ chức,

kể cả chức vụ và lương hiện tại. Tuy nhiên, qua thời gian, một số tệp nhỏ hơn được trích từ tệp chủ lớn hơn đã được chắt ra để xử lý hiệu quả cũng như để dành cho các ứng dụng chuyên biệt. Do đó, tệp nhân sự chủ để ra tệp lương, tệp bảo hiểm y tế, tệp lương hưu, tệp danh sách gửi thư, danh sách các nhân viên gia nhập công ti trước khi có chương trình quản lí (những người đã được trả lương bằng chương trình tính lương khác), và do vậy có đến hàng chục, có lẽ hàng trăm tệp và chương trình tồn tại.

Trong tổ chức xem như một tổng thể, tiến trình này dẫn tới việc nhân bản các tệp chủ được tạo ra, được duy trì và vận hành bởi nhiều phòng ban tách biệt. Tất nhiên mọi khoản mục dữ liệu trong các tệp khác nhau lại đòi hỏi một tập tài liệu để hỗ trợ cho tệp và giúp thu thập thông tin. Thường thì cùng một khoản mục dữ liệu, như mã sản phẩm, được thu thập trên nhiều tài liệu bởi những phòng ban khác nhau. Trong lúc đó, cấu trúc tệp của tổ chức đã trở thành phức tạp đến mức người lập trình phát triển những đặc sản bằng việc tập trung vào một tập con các tệp và chương trình. Cuối cùng, các chương trình này trở thành hoàn toàn phụ thuộc vào vài người lập trình, người hiểu các chương trình và tệp đó. Nếu những người lập trình này bị ốm hay ra khỏi tổ chức, thì các ứng dụng chủ chốt bị hỏng.

Có những cái tên cho tình huống này: **môi trường tệp truyền thống**; tổ chức tệp đet (bởi vì phần lớn dữ liệu có nguồn gốc từ các tệp đet); và cách tiếp cận tệp dữ liệu đó (vì dữ liệu và logic nghiệp vụ được gắn với các tệp riêng biệt và các chương trình có liên quan). Dù với cái tên nào thì tình huống này cũng làm tăng tính không hiệu quả và độ phức tạp.

Khi tiến trình này cứ tiếp diễn trong năm hay mười năm, thì tổ chức trở nên bị trói buộc sự sáng tạo riêng của mình trong cái nút. Tổ chức bị chững chất bởi hàng trăm chương trình và ứng dụng, chẳng ai biết chúng làm gì, chúng dùng dữ liệu gì, và ai dùng dữ liệu ấy. Không có bản in trung tâm về các tệp dữ liệu, phần từ

dữ liệu hay định nghĩa dữ liệu. Tổ chức cứ thu thập cùng thông tin trên quá nhiều tài liệu. Các vấn đề phát sinh là sự dư thừa dữ liệu, sự phụ thuộc chương trình-dữ liệu, không linh hoạt, kém an toàn dữ liệu và không có khả năng dùng chung dữ liệu giữa các ứng dụng.

DƯ THỪA DỮ LIỆU VÀ LẤN LỘN

Dư thừa dữ liệu là sự hiện diện của việc nhân bản dữ liệu trong nhiều tệp dữ liệu. Dư thừa dữ liệu xuất hiện khi các phòng ban, vùng chức năng, và các nhóm trong tổ chức thu thập cùng mẫu thông tin một cách độc lập. Chẳng hạn, bên trong phòng cho vay thương mại của ngân hàng, các chức năng tiếp thị và thông tin vay nợ có thể thu thập cùng thông tin khách hàng. Bởi vì nó được thu thập và duy trì trong nhiều chỗ khác nhau quá, nên cùng khoản mục dữ liệu này có thể có nhiều nghĩa khác nhau trong các phần khác nhau của tổ chức. Các khoản mục dữ liệu đơn giản như năm tài khoá, căn cước nhân viên, và mã sản phẩm có thể mang những nghĩa khác nhau khi người lập trình và phân tích làm việc trên những ứng dụng khác nhau một cách cô lập.

SỰ PHỤ THUỘC CHƯƠNG TRÌNH-DỮ LIỆU

Phụ thuộc chương trình-dữ liệu là mối quan hệ chặt chẽ giữa dữ liệu được lưu trữ trong tệp và những chương trình đặc thù cần để cập nhật và duy trì các tệp đó. Mọi chương trình máy tính đều phải mô tả vị trí và bản chất của dữ liệu mà nó làm việc với. Những khai báo dữ liệu này có thể dài hơn phần bản chất của chương trình. Trong môi trường tệp truyền thống, bất kì thay đổi nào trong dữ liệu cũng yêu cầu thay đổi trong mọi chương trình truy nhập vào dữ liệu đó. Những thay đổi lập trình như vậy có thể tốn tới hàng triệu đô la để cài đặt trong từng chương trình cần tới dữ liệu đã sửa đổi đó.

Việc phát triển của những ứng dụng mới kết quả là phải tốn thời gian và tiền bạc nhiều hơn. Người lập trình phải viết các

kể cả chức vụ và lương hiện tại. Tuy nhiên, qua thời gian, một số tệp nhỏ hơn được trích từ tệp chủ lớn hơn đã được chắt ra để xử lý hiệu quả cũng như để dành cho các ứng dụng chuyên biệt. Do đó, tệp nhân sự chủ để ra tệp lương, tệp bảo hiểm y tế, tệp lương hưu, tệp danh sách gửi thư, danh sách các nhân viên gia nhập công ti trước khi có chương trình quản lý (những người đã được trả lương bằng chương trình tính lương khác), và do vậy có đến hàng chục, có lẽ hàng trăm tệp và chương trình tồn tại.

Trong tổ chức xem như một tổng thể, tiến trình này dẫn tới việc nhân bản các tệp chủ được tạo ra, được duy trì và vận hành bởi nhiều phòng ban tách biệt. Tất nhiên mọi khoản mục dữ liệu trong các tệp khác nhau lại đòi hỏi một tập tài liệu để hỗ trợ cho tệp và giúp thu thập thông tin. Thường thì cùng một khoản mục dữ liệu, như mã sản phẩm, được thu thập trên nhiều tài liệu bởi những phòng ban khác nhau. Trong lúc đó, cấu trúc tệp của tổ chức đã trở thành phức tạp đến mức người lập trình phát triển những đặc sản bằng việc tập trung vào một tập con các tệp và chương trình. Cuối cùng, các chương trình này trở thành hoàn toàn phụ thuộc vào vài người lập trình, người hiểu các chương trình và tệp đó. Nếu những người lập trình này bị ốm hay ra khỏi tổ chức, thì các ứng dụng chủ chốt bị hỏng.

Có những cái tên cho tình huống này: **môi trường tệp truyền thống**; tổ chức tệp dẹt (bởi vì phần lớn dữ liệu có nguồn gốc từ các tệp dẹt); và cách tiếp cận tệp dữ liệu đó (vì dữ liệu và logic nghiệp vụ được gán với các tệp riêng biệt và các chương trình có liên quan). Dù với cái tên nào thì tình huống này cũng làm tăng tính không hiệu quả và độ phức tạp.

Khi tiến trình này cứ tiếp diễn trong năm hay mười năm, thì tổ chức trở nên bị trói buộc sự sáng tạo riêng của mình trong cái nút. Tổ chức bị chông chốt bởi hàng trăm chương trình và ứng dụng, chẳng ai biết chúng làm gì, chúng dùng dữ liệu gì, và ai dùng dữ liệu ấy. Không có bản in trung tâm về các tệp dữ liệu, phần tử

chương trình phức tạp, tách các khoản mục dữ liệu từ bản ghi trong nhiều tệp để tạo ra các tệp mới. Chương trình mới đòi hỏi cách bố trí dữ liệu mới. Phần lớn nỗ lực lập trình của tổ chức bao gồm việc cập nhật các phần tử dữ liệu vốn bị rải rác trong hàng trăm tệp. Trong nhiều trường hợp, các ứng dụng làm việc với dữ liệu lạc hậu đơn giản bởi vì khó tiến hành cập nhật.

THIẾU LINH HOẠT

Hệ thống tệp truyền thống có thể chuyển giao các báo cáo đã lên lịch thường lệ sau những nỗ lực lập trình lớn, nhưng nó không thể chuyển giao các báo cáo không theo thể thức hay đáp ứng cho các yêu cầu thông tin không dự kiến trước một cách đúng hạn. Thông tin được yêu cầu từ các yêu cầu không theo thể thức là "ở đâu đó trong hệ thống" nhưng việc dò tìm lại quá tốn kém. Nhiều người lập trình sẽ phải làm việc nhiều tuần để gắn các khoản mục dữ liệu được yêu cầu vào trong tệp mới. Người dùng - đặc biệt là giới quản lý cấp cao - bắt đầu tự hỏi tại điểm này là tại sao họ lại có mặt lính.

KÉM AN TOÀN

Vì ít có kiểm soát hay quản lý dữ liệu, việc truy nhập và phát tán thông tin thực tế là ngoài sự kiểm soát. Điều giới hạn lên việc truy nhập đang tồn tại có khuynh hướng là kết quả của thói quen và truyền thống, cũng như của khó khăn cực kì trong việc tìm thông tin.

THIẾU VIỆC DÙNG CHUNG DỮ LIỆU VÀ TÍNH SẴN CÓ

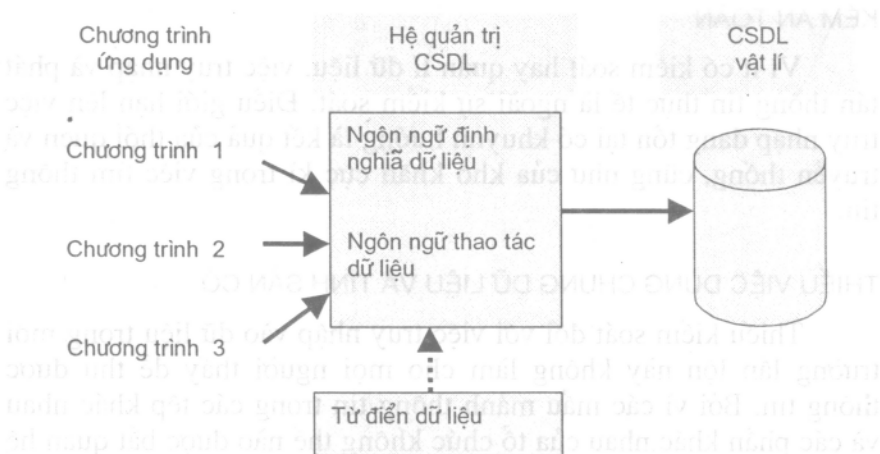
Thiếu kiểm soát đối với việc truy nhập vào dữ liệu trong môi trường lẫn lộn này không làm cho mọi người thấy dễ thu được thông tin. Bởi vì các mẫu mảnh thông tin trong các tệp khác nhau và các phần khác nhau của tổ chức không thể nào được bất quan hệ

với nhau, nên thực tế không thể nào dùng chung hay truy nhập vào thông tin theo cách thức kịp thời được.

2.3.2 MÔI TRƯỜNG CƠ SỞ DỮ LIỆU HIỆN ĐẠI

Công nghệ cơ sở dữ liệu có thể xoá bỏ đi những vấn đề do tổ chức tệp truyền thống tạo ra. Một định nghĩa chặt chẽ hơn về **cơ sở dữ liệu** là một tuyển tập các dữ liệu được tổ chức để phục vụ nhiều ứng dụng một cách có hiệu quả bằng việc tập trung dữ liệu và tối thiểu hoá dữ liệu dư thừa. Thay vì lưu giữ dữ liệu trong các tệp tách biệt cho từng ứng dụng, thì dữ liệu được lưu giữ về mặt vật lí để xuất hiện cho người dùng cứ như là được lưu giữ chỉ trong một chỗ. Một cơ sở dữ liệu riêng phục vụ cho nhiều ứng dụng. Chẳng hạn, thay vì một công ti lưu giữ dữ liệu nhân viên trong các hệ thống tin tách biệt và các tệp tách biệt cho nhân sự, tính lương và lợi ích, thì công ti có thể tạo ra một cơ sở dữ liệu Tài nguyên nhân lực chung duy nhất.

Hệ quản trị cơ sở dữ liệu (DBMS)



Hình 2.14 Hệ quản trị cơ sở dữ liệu.

Hệ quản trị cơ sở dữ liệu (DBMS) đơn giản là phần mềm cho phép tổ chức tập trung hoá dữ liệu, quản lý chúng một cách có hiệu quả, và cung cấp việc truy nhập vào dữ liệu đã lưu giữ bởi các chương trình ứng dụng. Như được minh hoạ trong Hình 2.14, DBMS hành động như một giao diện giữa chương trình ứng dụng và tệp dữ liệu vật lý. Khi chương trình ứng dụng cần tới một khoản mục dữ liệu như tổng số lương, thì DBMS tìm khoản mục này trong cơ sở dữ liệu rồi trình bày nó cho chương trình ứng dụng. Dùng các tệp dữ liệu truyền thống, người lập trình sẽ phải định nghĩa dữ liệu này và rồi bảo máy tính chúng ở đâu. DBMS khử bỏ phần lớn những câu lệnh định nghĩa dữ liệu tìm thấy trong chương trình truyền thống.

- Ngôn ngữ định nghĩa dữ liệu.
- Ngôn ngữ thao tác dữ liệu.
- Từ điển dữ liệu

Ngôn ngữ định nghĩa dữ liệu là ngôn ngữ hình thức do người lập trình dùng để xác định nội dung và cấu trúc của cơ sở dữ liệu. Ngôn ngữ định nghĩa dữ liệu xác định từng phần tử dữ liệu như nó xuất hiện trong cơ sở dữ liệu trước khi phần tử dữ liệu đó được dịch thành dạng mà chương trình ứng dụng yêu cầu.

Phần lớn DBMS đều có một ngôn ngữ chuyên dụng gọi là **ngôn ngữ thao tác dữ liệu**, vốn được dùng tiếp nối với một số ngôn ngữ lập trình qui ước thế hệ ba hay bốn để thao tác dữ liệu trong cơ sở dữ liệu. Ngôn ngữ này chứa các chỉ lệnh cho phép người dùng cuối và các chuyên viên lập trình trích rút dữ liệu từ cơ sở dữ liệu để thoả mãn yêu cầu thông tin và phát triển ứng dụng. Ngôn ngữ thao tác dữ liệu có triển vọng nhất ngày nay là **SQL**, hay Ngôn ngữ vấn đáp có cấu trúc Structured Query Language. Các nhiệm vụ lập trình phức tạp không thể nào được thực hiện một cách có hiệu quả bằng các ngôn ngữ thao tác dữ liệu điển hình. Tuy nhiên phần lớn DBMS trên máy tính lớn đều tương hợp với COBOL

và FORTRAN, và các ngôn ngữ lập trình thế hệ thứ ba khác, cho phép có hiệu quả xử lý lớn hơn và linh hoạt hơn.

Phần tử thứ ba của DBMS là **từ điển dữ liệu**. Đây là một tệp tự động hoá hay thủ công vốn lưu giữ các định nghĩa về các phần tử dữ liệu và các đặc trưng dữ liệu như việc sử dụng, biểu diễn vật lý, quyền sở hữu (ai trong tổ chức chịu trách nhiệm duy trì dữ liệu này), chủ quyền và an toàn. Nhiều từ điển dữ liệu có thể tạo ra các danh sách và báo cáo về sử dụng dữ liệu, gộp nhóm, vị trí chương trình v.v. **Phần tử dữ liệu** biểu diễn cho một trường. Bên cạnh danh sách các tên chuẩn, từ điển còn liệt kê các tên tham chiếu tới phần tử này trong một hệ thống riêng biệt và xác định các cá nhân, chức năng nghiệp vụ, chương trình, và báo cáo việc dùng về phần tử dữ liệu này.

Bằng việc tạo ra một kho tất cả các mẫu mảnh dữ liệu có chứa trong cơ sở dữ liệu, từ điển dữ liệu phục vụ như một dụng cụ quản lý dữ liệu quan trọng. Người dùng nghiệp vụ có thể tra cứu từ điển này để tìm ra đích xác mẫu dữ liệu nào được duy trì cho chức năng bán hàng hay tiếp thị hay thậm chí xác định tất cả những thông tin được toàn bộ tổ chức duy trì. Từ điển này cũng cung cấp cho người dùng nghiệp vụ về tên, định dạng và đặc tả cần để truy nhập vào dữ liệu cho báo cáo. Các nhân viên kỹ thuật có thể dùng từ điển này để xác định phần tử dữ liệu và tệp nào phải bị thay đổi nếu chương trình thay đổi.

Phần lớn các từ điển dữ liệu là hoàn toàn thụ động; chúng đơn giản báo cáo. Các kiểu nâng cao hơn thì chủ động; những thay đổi trong từ điển có thể được tự động sử dụng bởi các chương trình có liên quan. Chẳng hạn, thay đổi số điện thoại từ sáu lên bảy con số thì người ta có thể đơn giản đưa thay đổi đó vào trong từ điển mà không phải sửa đổi và dịch lại tất cả chương trình của người dùng số điện thoại đó.

Trong một môi trường cơ sở dữ liệu lý tưởng, dữ liệu trong cơ sở dữ liệu được định nghĩa một lần và nhất quán và được dùng cho

mọi ứng dụng có dữ liệu nằm trong cơ sở dữ liệu này. Các chương trình ứng dụng (thường được viết bằng việc dùng một tổ hợp của ngôn ngữ thao tác dữ liệu của DBMS và ngôn ngữ qui ước như COBOL) yêu cầu các phần tử dữ liệu từ cơ sở dữ liệu. Các phần tử dữ liệu này được gọi bởi chương trình ứng dụng do DBMS tìm thấy và chuyển giao. Người lập trình không phải xác định chi tiết dữ liệu đó được tìm thế nào và ở đâu.

Cái nhìn logic và vật lí về dữ liệu

Có lẽ sự khác biệt lớn nhất giữa một DBMS và tổ chức tệp truyền thống là ở chỗ DBMS tách bạch các cái nhìn logic và vật lí về dữ liệu, làm giảm nhẹ cho người lập trình hay người dùng cuối khỏi nhiệm vụ hiểu dữ liệu thực tại được lưu giữ ở đâu và thế nào.

Khái niệm cơ sở dữ liệu phân biệt giữa cái nhìn *logic* và *vật lí* về dữ liệu. **Cái nhìn logic** trình bày dữ liệu như chúng sẽ được người dùng cuối hay chuyên viên nghiệp vụ cảm nhận, trong khi **cái nhìn vật lí** chỉ ra cách dữ liệu thực tế được tổ chức và cấu trúc trên môi giới lưu giữ vật lí.

Mô tả logic về toàn bộ cơ sở dữ liệu, liệt kê tất cả các khoản mục dữ liệu và mối quan hệ giữa chúng, được gọi theo thuật ngữ là **lược đồ**. Tập riêng các dữ liệu từ cơ sở dữ liệu mà được từng chương trình ứng dụng yêu cầu được gọi theo thuật ngữ là **lược đồ con**.

Ưu điểm của hệ quản trị cơ sở dữ liệu

Thảo luận trên đây minh hoạ cho các ưu điểm của DBMS:

- Độ phức tạp của môi trường hệ thông tin của tổ chức có thể được thu lại bằng việc quản lí tập trung dữ liệu, truy nhập, sử dụng và an toàn.

- Việc dư thừa và không nhất quán dữ liệu có thể được rút gọn bằng việc khử bỏ mọi tệp cô lập mà trong đó cùng phần tử dữ liệu được lặp lại.
- Sự lộn xộn dữ liệu có thể được xóa bỏ bằng việc cung cấp kiểm soát trung tâm về việc tạo ra và định nghĩa dữ liệu.
- Sự phụ thuộc chương trình - dữ liệu có thể được rút gọn bằng việc tách bạch cái nhìn logic với việc bố trí vật lý.
- Chi phí cho việc phát triển và bảo trì có thể được rút lại đáng kể
- Tính linh hoạt của hệ thống tin có thể được nâng cao lớn lao bằng việc cho phép các câu hỏi không tốn kém, nhanh chóng và không theo thể thức về khối lượng lớn thông tin.
- Việc truy nhập và tính sẵn có của thông tin có thể được tăng lên.

Với tất cả những ích lợi này của DBMS, người ta có thể trông đợi tất cả các tổ chức sẽ thay đổi ngay sang dạng quản lý thông tin theo cơ sở dữ liệu. Nhưng điều đó không phải dễ dàng như chúng ta sẽ thấy sau đây.

2.3.3 THIẾT KẾ CƠ SỞ DỮ LIỆU

Có nhiều phương án tổ chức dữ liệu và biểu diễn mối quan hệ giữa dữ liệu trong cơ sở dữ liệu. Các DBMS qui ước thường dùng một trong ba mô hình cơ sở dữ liệu logic chính để giữ dấu vết của các thực thể, thuộc tính và mối quan hệ. Ba mô hình cơ sở dữ liệu logic chính là phân cấp, mạng và quan hệ. Mỗi mô hình logic lại có những ưu điểm xử lý nào đó và những ưu thế nghiệp vụ nào đó.

Mô hình dữ liệu phân cấp

Các DBMS sớm nhất đều là phân cấp. **Mô hình dữ liệu phân cấp** trình bày dữ liệu cho người dùng trong một cấu trúc tựa cây. DBMS phân cấp chung nhất là IMS (Information Management

System) của IBM. Bên trong mỗi bản ghi, các phần tử dữ liệu được tổ chức thành từng mẫu mảnh bản ghi gọi là đoạn. Với người dùng, mỗi bản ghi trông như một sơ đồ tổ chức với một đoạn ở mức đỉnh được gọi là gốc. Đoạn phía trên được nối về mặt logic với đoạn phía dưới trong mỗi quan hệ cha-con. Đoạn cha có thể có nhiều con, nhưng một con chỉ có thể có một cha.

Đằng sau cái nhìn logic về dữ liệu là một số móc nối vật lý và thiết bị để buộc thông tin lại trong một toàn thể logic. Trong DBMD phân cấp dữ liệu được móc nối về mặt vật lý với nhau thông qua một loạt con trỏ vốn tạo nên dây chuyền các đoạn dữ liệu có liên quan. Con trỏ là các yếu tố dữ liệu được gắn vào cuối các đoạn bản ghi trên đĩa để hướng dẫn hệ thống tới bản ghi có liên quan.

Mô hình dữ liệu mạng

Mô hình dữ liệu mạng là một biến thể của mô hình dữ liệu phân cấp. Quả vậy, các cơ sở dữ liệu có thể được dịch từ phân cấp sang mạng và ngược lại để tối ưu tốc độ xử lý và để thuận tiện. Trong khi các cấu trúc phân cấp mô tả các mối quan hệ một-nhiều, thì cấu trúc mạng lại mô tả cho dữ liệu về mặt logic như mối quan hệ nhiều-nhiều.

Mối quan hệ nhiều-nhiều điển hình trong đó DBMS mạng trội hơn về hiệu năng là mối quan hệ sinh viên - học trình. Có nhiều học trình trong đại học và nhiều sinh viên, và sinh viên thường đăng kí theo nhiều học trình. Nếu các mối quan hệ này được cấu trúc theo kiểu phân cấp thì sẽ gây ra những dư thừa đáng kể và làm chậm đáp ứng với kiểu câu hỏi thông tin nào đó. Cấu trúc mạng làm giảm sự dư thừa và trong những tình huống nào đó (khi mối quan hệ nhiều-nhiều tham dự vào), thì việc đáp ứng được nhanh chóng hơn. Tuy nhiên cái giá cho việc giảm dư thừa này và tăng tốc độ là: số các con trỏ trong cấu trúc mạng tăng lên nhanh chóng, làm cho việc bảo trì và vận hành thành tốn kém.

Mô hình dữ liệu quan hệ

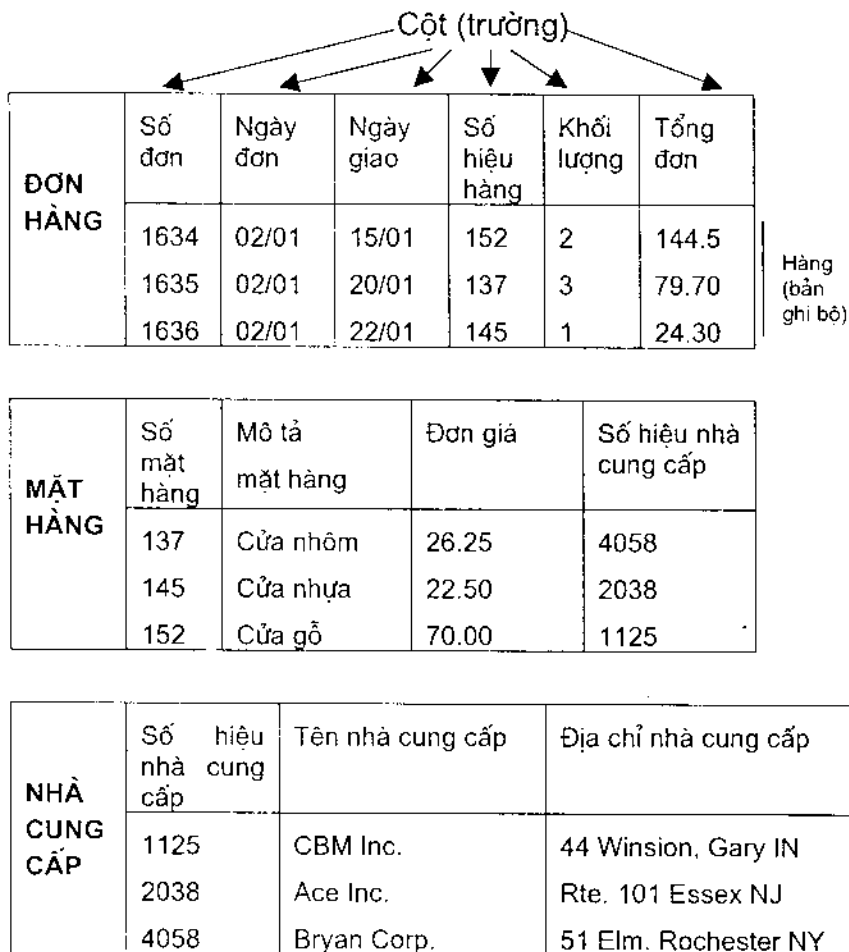
Mô hình dữ liệu quan hệ, mô hình mới nhất trong ba mô hình dữ liệu này, vượt qua một số những giới hạn của hai mô hình trước. Mô hình quan hệ biểu diễn cho tất cả các dữ liệu trong cơ sở dữ liệu như các bảng hai chiều đơn giản được gọi là quan hệ. Các bảng này giống như các tệp dữ liệu, nhưng thông tin trong nhiều tệp có thể dễ dàng được trích ra và tổ hợp lại. Đôi khi các bảng còn được gọi là các tệp.

Hình 2.15 chỉ ra bảng nhà cung cấp, một bảng mặt hàng và một bảng đơn hàng. Trong mỗi bảng, các hàng là các bản ghi duy nhất còn các cột là các trường. Một thuật ngữ khác cho hàng hay bản ghi trong quan hệ là một bộ. Thường người dùng cần thông tin từ một số quan hệ để tạo ra báo cáo. Đây là sức mạnh của mô hình quan hệ: Nó có thể lập quan hệ dữ liệu trong bất kì tệp nào hay bảng nào với dữ liệu trong tệp hay bảng khác *chẳng hạn mà cả hai bảng đều có chung một phần tử dữ liệu*.

Để trình bày, giả sử chúng ta muốn tìm trong cơ sở dữ liệu quan hệ trong Hình 2.15 tên và địa chỉ của nhà cung cấp, người có thể cung cấp cho chúng ta mặt hàng số hiệu 137 hay số hiệu 152. Chúng ta sẽ cần thông tin từ hai bảng: bảng nhà cung cấp và bảng mặt hàng. Chú ý rằng hai tệp này có một phần tử dữ liệu chung : Số hiệu Nhà cung cấp.

Trong cơ sở dữ liệu quan hệ, ba phép toán cơ bản được dùng để phát triển các tập dữ liệu có ích: chọn, chiếu và nối. Phép toán *select* (chọn) tạo ra một tập con bao gồm tất cả các bản ghi trong tệp vốn đáp ứng cho tiêu chuẩn đã thiết lập. Nói cách khác "select" tạo ra một tập con các hàng đáp ứng cho tiêu chuẩn nào đó. Trong thí dụ của ta, chúng ta muốn chọn các bản ghi (hàng) từ số hiệu mặt hàng bằng với 137 hay 152.

Bảng (quan hệ)



Hình 2.15 Mô hình dữ liệu quan hệ.

Phép toán *nối join* tổ hợp các bảng quan hệ lại để cung cấp cho người dùng với nhiều thông tin hơn hiện có trong từng bảng

riêng lẻ. Trong thí dụ của ta, chúng ta muốn nối bảng mặt hàng ngắn (chỉ những mặt hàng mang số hiệu 137 và 152 mới được nêu ra) và bảng nhà cung cấp thành một bảng kết quả mới.

Phép toán *chiếu project* tạo ra một tập con bao gồm các cột trong một bảng, cho phép người dùng tạo ra các bảng mới có chứa chỉ những thông tin yêu cầu. Trong thí dụ này, chúng ta muốn trích ra từ bảng kết quả mới chỉ những cột sau: Số hiệu mặt hàng, số hiệu nhà cung cấp, tên nhà cung cấp và địa chỉ nhà cung cấp.

Các hệ quản trị cơ sở dữ liệu quan hệ trên máy tính lớn hàng đầu bao gồm DB2 của IBM và Oracle của Oracle Corporation. FoxBase Plus của Fox Software Inc., và dBASE IV PLUS và Paradox của Borland International Inc. đều là những thí dụ về các hệ quản trị cơ sở dữ liệu quan hệ trên máy vi tính.

Ưu và nhược điểm của ba mô hình cơ sở dữ liệu

Ưu điểm chính của các mô hình cơ sở dữ liệu mạng và phân cấp là ở việc xử lý hiệu quả. Chẳng hạn, mô hình phân cấp là thích hợp cho hệ thống xử lý giao tác giữ chỗ hàng không, vốn phải giải quyết cho hàng triệu yêu cầu thường lệ có cấu trúc mỗi ngày về thông tin đặt chỗ.

Các cấu trúc phân cấp và mạng có nhiều nhược điểm. Tất cả mọi đường truy nhập, danh mục, và chỉ số đều phải được xác định trước. Một khi được xác định, thì chúng không dễ bị thay đổi mà không có nỗ lực lập trình chính. Do đó, những thiết kế này có sự linh hoạt thấp. Chẳng hạn, nếu bạn gọi điện đến hãng hàng không và nói, "Bố tôi cho tôi chiếc vé đi nghỉ mát Điện Biên từ sân bay Phú Bài, nhưng không nói rõ đi vào lúc nào và ở đâu. Vậy tôi đi đâu?" thì bạn sẽ phát hiện ra rằng chẳng có cách nào để hệ thống có thể tìm ra câu trả lời trong thời gian thoả đáng.

Cả hai hệ thống phân cấp và mạng đều cần rất nhiều việc lập trình tốn thời gian, khó cài đặt và khó chữa chạy nếu xuất hiện lỗi

thiết kế. Chúng không hỗ trợ cho các đòi hỏi không theo thể thức, kiểu ngôn ngữ tự nhiên về thông tin.

Sức mạnh của DBMS quan hệ là ở sự linh hoạt lớn đối với các câu hỏi không theo thể thức, mạnh về tổ hợp thông tin từ các nguồn khác nhau, sự đơn giản của thiết kế và bảo trì, và khả năng bổ sung dữ liệu và bản ghi mới mà không làm rối loạn các chương trình và ứng dụng hiện có. Nhược điểm của DBMS quan hệ là tính hiệu quả xử lý tương đối thấp. Các hệ thống này là cái gì đó hơi chậm hơn bởi vì chúng diễn hình yêu cầu nhiều truy nhập vào dữ liệu đã được lưu giữ trên đĩa để tiến hành các chỉ lệnh chọn, nối và chiếu. Việc chọn một số hiệu mặt hàng từ trong hàng triệu mặt hàng, mỗi bản ghi một lúc, có thể là tốn thời gian. Tất nhiên cơ sở dữ liệu có thể được làm chỉ số và làm tăng tốc với những câu hỏi đã mô tả trước. Các hệ thống quan hệ không có một số lớn các con trỏ như trong hệ thống phân cấp.

Các cơ sở dữ liệu lớn có thể được thiết kế để có một số dư thừa dữ liệu để làm cho việc dò tìm dữ liệu được hiệu quả hơn. Cùng một phần tử dữ liệu có thể được lưu giữ trong nhiều bảng. Việc cập nhật các phần tử dữ liệu dư thừa không tự động hoá trong nhiều DBMS quan hệ. Chẳng hạn, việc thay đổi trường trạng thái nhân viên trong bảng này sẽ không tự động thay đổi nó trong mọi bảng. Những cách bố trí đặc biệt được yêu cầu để đảm bảo rằng mọi bản sao của cùng phần tử dữ liệu là cùng được cập nhật.

Các cơ sở dữ liệu phân cấp vẫn còn là con ngựa chiến cho việc xử lý giao tác khối lượng lớn. Ngân hàng, các công ti bảo hiểm, và những người dùng khối lượng lớn dữ liệu vẫn tiếp tục dùng các cơ sở dữ liệu phân cấp như IMS của IBM, được phát triển từ năm 1969. Để làm ứng dụng chương trình trong môi trường quan hệ, nhưng nhiều tổ chức không muốn chi ra hàng triệu đô la để chuyển đổi phần mềm từ hệ quản trị cơ sở dữ liệu phân cấp sang quan hệ. Nhiều tổ chức đã chuyển sang DB2, một DBMS quan hệ của IBM

cho các ứng dụng mới, trong khi vẫn duy trì IMS cho việc xử lý giao tác truyền thống.

Tạo ra cơ sở dữ liệu

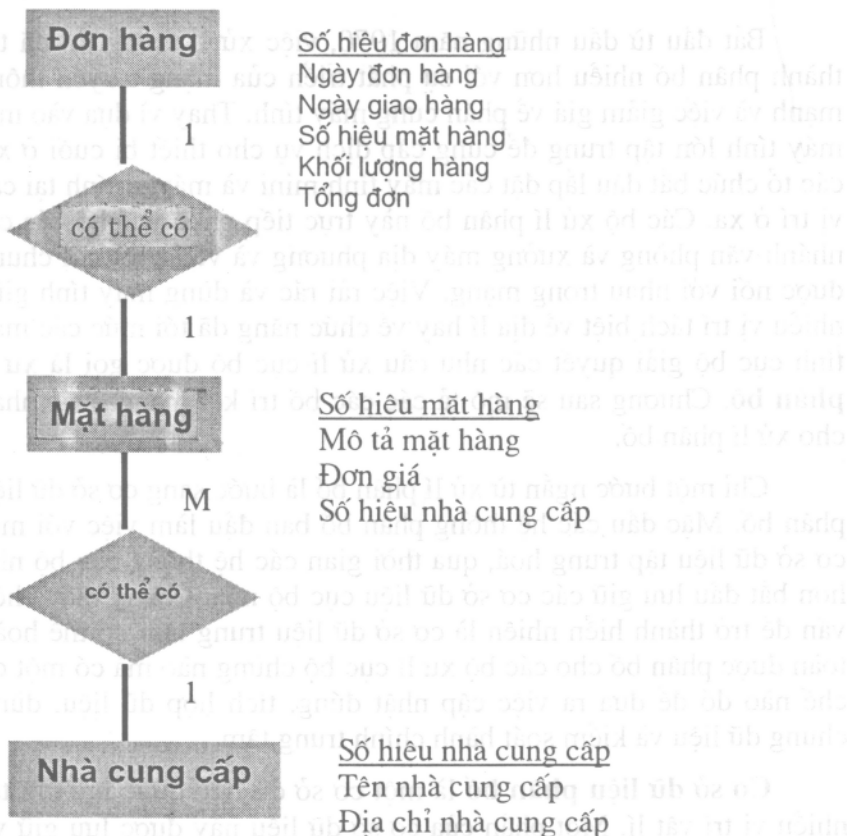
Để tạo ra cơ sở dữ liệu, người ta phải đi qua hai lần làm thiết kế, thiết kế quan niệm và thiết kế vật lý. Thiết kế quan niệm về cơ sở dữ liệu là một mô hình trừu tượng về cơ sở dữ liệu theo viễn cảnh nghiệp vụ, trong khi thiết kế vật lý lại chỉ ra cách cơ sở dữ liệu thực tế được bố trí trên thiết bị lưu giữ truy nhập trực tiếp. Thiết kế cơ sở dữ liệu vật lý được các chuyên viên cơ sở dữ liệu thực hiện, trong khi thiết kế logic, yêu cầu một mô tả chi tiết về thông tin nghiệp vụ, lại cần người dùng cuối thực tại của cơ sở dữ liệu. Một cách lí tưởng, thiết kế cơ sở dữ liệu sẽ là một phần của nỗ lực lập kế hoạch dữ liệu của tổ chức toàn thể.

Thiết kế cơ sở dữ liệu quan niệm mô tả cho cách các phân tử dữ liệu trong cơ sở dữ liệu được gộp nhóm lại. Tiến trình thiết kế xác định ra mối quan hệ trong các phân tử dữ liệu và cách hiệu quả nhất để gộp nhóm các phân tử dữ liệu với nhau để đáp ứng cho yêu cầu thông tin. Tiến trình này cũng xác định các phân tử dữ liệu dư thừa và các việc gộp nhóm phân tử dữ liệu cần cho các chương trình ứng dụng đặc biệt. Các nhóm dữ liệu được tổ chức, làm mịn và sắp xếp hợp lí cho tới khi một cái nhìn logic toàn thể về các mối quan hệ giữa tất cả các phân tử dữ liệu trong cơ sở dữ liệu nổi lên.

Những người thiết kế cơ sở dữ liệu làm tài liệu cho mô hình dữ liệu quan niệm bằng **biểu đồ thực thể-quan hệ**, được minh hoạ trong Hình 2.16. Các hộp biểu diễn cho các thực thể còn các hình thoi biểu diễn cho mối quan hệ. Đầu 1 hay M của hình thoi biểu thị cho mối quan hệ giữa các thực thể là một-một, một-nhiều hay nhiều-nhiều. Hình 2.16 chỉ rằng thực thể Đơn hàng có chủ thể có một Mặt hàng và một Mặt hàng chỉ có thể có một Nhà cung cấp. Nhiều mặt hàng có thể do một nhà cung cấp đưa ra. Các thuộc tính

cho từng thực thể được liệt kê ngay bên cạnh thực thể đó và trường khóa được gạch dưới.

Thuộc tính thực thể



Hình 2.16 Biểu đồ Thực thể- Quan hệ.

Khuynh hướng cơ sở dữ liệu mới đây bao gồm việc phát triển của cơ sở dữ liệu phân bố và việc nổi lên của các cơ sở dữ liệu siêu media và hướng sự vật.

Xử lý phân bố và cơ sở dữ liệu phân bố

Bắt đầu từ đầu những năm 1970, việc xử lý thông tin đã trở thành phân bố nhiều hơn với sự phát triển của mạng truyền thông mạnh và việc giảm giá về phần cứng máy tính. Thay vì dựa vào một máy tính lớn tập trung để cung cấp dịch vụ cho thiết bị cuối ở xa, các tổ chức bắt đầu lắp đặt các máy tính mini và máy vi tính tại các vị trí ở xa. Các bộ xử lý phân bố này trực tiếp phục vụ cho các chi nhánh văn phòng và xưởng máy địa phương và vùng, và nói chung được nối với nhau trong mạng. Việc rải rác và dùng máy tính giữa nhiều vị trí tách biệt về địa lý hay về chức năng đã tới mức các máy tính cục bộ giải quyết các nhu cầu xử lý cục bộ được gọi là **xử lý phân bố**. Chương sau sẽ mô tả các các bố trí kết mạng khác nhau cho xử lý phân bố.

Chỉ một bước ngắn từ xử lý phân bố là bước sang cơ sở dữ liệu phân bố. Mặc dầu các hệ thống phân bố ban đầu làm việc với một cơ sở dữ liệu tập trung hoá, qua thời gian các hệ thống cục bộ nhỏ hơn bắt đầu lưu giữ các cơ sở dữ liệu cục bộ nữa. Chẳng mấy chốc vấn đề trở thành hiển nhiên là cơ sở dữ liệu trung tâm có thể hoàn toàn được phân bố cho các bộ xử lý cục bộ chừng nào mà có một cơ chế nào đó để đưa ra việc cập nhật đúng, tích hợp dữ liệu, dùng chung dữ liệu và kiểm soát hành chính trung tâm.

Cơ sở dữ liệu phân bố là một cơ sở dữ liệu được lưu giữ tại nhiều vị trí vật lý. Một phần của cơ sở dữ liệu này được lưu giữ về mặt vật lý tại một vị trí còn các phần khác được lưu giữ và duy trì ở các vị trí khác. Có nhiều cách để phân bố cơ sở dữ liệu. Cơ sở dữ liệu trung tâm có thể được phân hoạch ra để cho từng bộ xử lý ở xa có dữ liệu cần thiết về khách hàng của mình để phục vụ cho vùng địa phương của mình. Những thay đổi ở các tệp cục bộ có thể được

cập nhật với cơ sở dữ liệu trung tâm trên cơ sở theo lô, thường vào đêm. Một chiến lược khác là nhân bản cơ sở dữ liệu trung tâm tại tất cả các vị trí ở xa. Chiến lược này cũng đòi hỏi cập nhật cơ sở dữ liệu trung tâm vào những giờ nghỉ.

Vẫn còn khả năng khác - khả năng thường được các cơ sở dữ liệu rất lớn sử dụng - là duy trì chỉ một chỉ mục tên trung tâm và lưu giữ đầy đủ các bản ghi một cách cục bộ. Việc hỏi đáp chỉ mục tên trung tâm xác định ra vị trí mà có thể tìm thấy đầy đủ bản ghi. Tại đây không có cơ sở dữ liệu trung tâm và không tốn chi phí cập nhật. National Westminster Bank ở London dùng cách tiếp cận tương tự để duy trì tất cả thông tin tài khoản khách hàng trong hai cơ sở dữ liệu DB2 lớn được chia đoạn. Phần mềm được phát triển đặc biệt cho phép mỗi một trong 22 000 người dùng của nó truy nhập vào dữ liệu trên cả hai cơ sở dữ liệu với bảng danh mục toàn cục về nơi dữ liệu được lưu giữ. Một biến thể khác là một sơ đồ "hỏi-mạng". Không có chỉ mục trung tâm cho các tên trong thiết kế này. Thay vì thế, tất cả các bộ xử lý ở xa đều được điếm để tìm ra bản ghi đầy đủ. Bản ghi đầy đủ tiếp đó được truyền tới bất kì bộ xử lý nào yêu cầu nó.

Cả việc xử lý phân bố và cơ sở dữ liệu phân bố đều có ích lợi và nhược điểm. Các hệ phân bố làm giảm tính mong manh của một vị trí trung tâm lớn. Chúng cho phép làm tăng sức mạnh của hệ thống bằng việc mua các máy tính mini nhỏ hơn và đỡ tốn kém hơn. Cuối cùng, chúng làm tăng dịch vụ và tính đáp ứng cho người dùng cục bộ. Tuy nhiên các hệ phân bố lại phụ thuộc vào đường viễn thông chất lượng cao, mà bản thân chúng thì cũng mong manh. Hơn nữa, cơ sở dữ liệu cục bộ đôi khi có thể xuất phát từ các chuẩn và định nghĩa dữ liệu trung tâm và làm lộ ra các vấn đề an toàn bởi việc truy nhập phân bố rộng rãi vào dữ liệu nhạy cảm. Tính kinh tế của phân bố có thể bị mất khi vị trí ở xa mua nhiều năng lực tính toán hơn chúng cần.

Mặc cho những nhược điểm này, xử lí phân bố vẫn đang phát triển nhanh chóng. Với tiến bộ của máy vi tính và hệ thống viễn thông mạnh, nhiều dịch vụ thông tin sẽ được phân bố. Đối với các tổ chức quốc gia lớn đang làm việc trong nhiều vùng, vấn đề không còn là liệu có phân bố hay không mà là cách phân bố thế nào để làm tối thiểu chi phí và nâng cao tính đáp ứng mà không phải hi sinh tính toàn vẹn dữ liệu và hệ thống.

Cơ sở dữ liệu hướng sự vật và siêu media

Hệ quản trị cơ sở dữ liệu qui ước đã được thiết kế cho các dữ liệu thuần nhất mà có thể dễ dàng được cấu trúc lại trong các trường và bản ghi dữ liệu định trước. Nhưng nhiều ứng dụng từ những năm 1990 sẽ yêu cầu các cơ sở dữ liệu có thể lưu giữ và dò tìm không chỉ các số và kí tự có cấu trúc mà còn cả tranh, ảnh, ảnh chụp, tiếng nói, và video chuyển động. Các DBMS qui ước không thích hợp để giải quyết cho các ứng dụng dựa trên đồ hoạ hay multimedia. Chẳng hạn, dữ liệu thiết kế trong cơ sở dữ liệu CAD bao gồm các mối quan hệ phức tạp giữa nhiều kiểu dữ liệu. Việc thao tác những kiểu dữ liệu này trong hệ thống quan hệ đòi hỏi lập trình rất nhiều để dịch những cấu trúc dữ liệu phức tạp này thành các bảng và hàng. **Cơ sở dữ liệu hướng sự vật**, mặt khác lại lưu giữ dữ liệu và thủ tục như các sự vật mà có thể được tự động dò tìm và dùng chung.

Cách tiếp cận **cơ sở dữ liệu siêu media** tới việc quản lí thông tin vượt lên trên một số giới hạn của phương pháp cơ sở dữ liệu truyền thống bằng việc lưu giữ cả chùm thông tin dưới dạng các nút được nối bởi các móc nối do người dùng thiết lập. Các nút này có thể chứa văn bản, đồ hoạ, âm thanh, video chuyển động, hay chương trình máy tính thực hiện được. Việc truy tìm thông tin không phải đi theo một sơ đồ tổ chức định sẵn. Thay vì vậy, người ta có thể nhảy lập tức tới thông tin có liên quan trong bất kì loại

quan hệ nào được tác giả thiết lập. Mỗi quan hệ giữa các bản ghi ít tính cấu trúc hơn trong DBMS truyền thống.

2.3.4 CÁC YÊU CẦU QUẢN LÝ CHO HỆ THỐNG CƠ SỞ DỮ LIỆU

Nhiều điều hơn nữa còn được yêu cầu cho việc phát triển hệ thống cơ sở dữ liệu so với việc chọn lựa đơn giản một mô hình cơ sở dữ liệu logic. Quả thực việc lựa chọn này có thể là nằm trong những quyết định cuối cùng. Cơ sở dữ liệu là một bộ môn, một phương pháp mang tính tổ chức, chứ không chỉ là một dụng cụ hay công nghệ. Nó đòi hỏi sự thay đổi tổ chức và quan niệm.

Không có sự hỗ trợ quản lý và hiểu biết, thì nỗ lực cơ sở dữ liệu sẽ thất bại. Các phần tử chủ chốt trong môi trường cơ sở dữ liệu là 1) quản trị dữ liệu, 2) lập kế hoạch dữ liệu và phương pháp luận mô hình hoá, 3) công nghệ và quản lý cơ sở dữ liệu, và 4) người dùng.

Quản trị dữ liệu

Hệ thống cơ sở dữ liệu yêu cầu rằng tổ chức nhận ra vai trò chiến lược của thông tin và bắt đầu tích cực quản lý và lập kế hoạch cho thông tin như tài nguyên của tổ chức. Điều này có nghĩa là tổ chức phải phát triển chức năng quản trị dữ liệu với quyền định nghĩa ra yêu cầu thông tin cho toàn bộ tổ chức và tiếp cận trực tiếp với cấp quản lý cao. Viên chức phụ trách thông tin (CIO) hay phó chủ tịch thông tin trở thành người phụ trách chủ yếu trong tổ chức về hệ thống cơ sở dữ liệu.

Quản trị dữ liệu là việc thực thi các chính sách và thủ tục đặc biệt, qua đó dữ liệu có thể được quản lý như một nguồn tài nguyên của tổ chức. Những trách nhiệm này bao gồm việc phát triển chính sách thông tin, lập kế hoạch về dữ liệu, dự kiến trước thiết kế cơ sở

dữ liệu logic và phát triển từ điển dữ liệu, rồi điều phối việc sử dụng dữ liệu qua các chuyên viên hệ thống tin và nhóm người dùng cuối.

Nguyên lý nền tảng của quản trị dữ liệu là ở chỗ tất cả các dữ liệu đều là tài sản của tổ chức xem như một toàn thể. Dữ liệu không thể là của riêng bất kì miền nghiệp vụ nào hay đơn vị tổ chức nào. Tất cả các dữ liệu đều phải luôn có sẵn cho bất kì nhóm nào cần tới chúng để hoàn thành sứ mệnh của mình. Tổ chức cần phát biểu có hệ thống về **chính sách thông tin**, vốn xác định ra các qui tắc về việc dùng chung, phát tán, thu nhận, chuẩn hoá, phân loại và lưu trữ thông tin trong khắp tổ chức. Chính sách thông tin ấn định ra các thủ tục và trách nhiệm riêng, xác định các đơn vị tổ chức nào dùng chung thông tin; nơi nào thông tin có thể được phân bố; và ai có trách nhiệm cập nhật và duy trì thông tin.

Lập kế hoạch dữ liệu và phương pháp luận mô hình hoá

Bởi vì, mối quan tâm của tổ chức có dùng DBMS là rộng hơn nhiều so với mối quan tâm trong môi trường tệp truyền thống, nên tổ chức yêu cầu việc lập kế hoạch dữ liệu trên qui mô toàn tổ chức. Việc phân tích tổ chức, vốn đề cập tới các yêu cầu thông tin của toàn bộ tổ chức (như đối lập với các yêu cầu của ứng dụng cá nhân), là cần thiết để phát triển cơ sở dữ liệu. Mục đích của việc phân tích tổ chức là để xác định ra các thực thể chính, các thuộc tính, và mối quan hệ vốn thiết lập nên dữ liệu của tổ chức.

Công nghệ và quản lí cơ sở dữ liệu

Các cơ sở dữ liệu yêu cầu phần mềm mới và nhân viên mới được huấn luyện đặc biệt về các kĩ thuật DBMS cũng như các cấu trúc quản lí mới. Phần lớn các công ti đều phát triển một thiết kế cơ sở dữ liệu và nhóm quản lí bên trong bộ phận hệ thống tin tổ chức, vốn chịu trách nhiệm nhiều về các khía cạnh kĩ thuật và vận hành

của quản lí dữ liệu. Các chức năng đó được gọi là **quản trị cơ sở dữ liệu**. Nhóm này làm những việc sau:

- Định nghĩa và tổ chức cấu trúc và nội dung cơ sở dữ liệu.
- Phát triển các thủ tục an toàn để gìn giữ cơ sở dữ liệu
- Phát triển tài liệu cơ sở dữ liệu.
- Duy trì phần mềm quản trị cơ sở dữ liệu

Trong hợp tác gần gũi với người dùng, nhóm thiết kế thiết lập nên cơ sở dữ liệu vật lí, các quan hệ logic giữa các phần tử và các qui tắc và thủ tục truy nhập.

Người dùng

Cơ sở dữ liệu phục vụ cho một cộng đồng rộng những người dùng hơn là các hệ thống truyền thống. Các hệ thống quan hệ với các ngôn ngữ hỏi đáp thể hệ bốn cho phép các nhân viên vốn không phải là chuyên viên máy tính truy nhập vào các cơ sở dữ liệu lớn. Bên cạnh đó, người dùng bao gồm cả các chuyên viên máy tính được huấn luyện. Để làm tối ưu việc truy nhập đối với những người không phải là chuyên viên, nhiều tài nguyên hơn cần phải được dồn cho việc huấn luyện người dùng cuối. Các công nhân hệ thống chuyên nghiệp phải được huấn luyện lại theo ngôn ngữ DBMS, các thủ tục phát triển ứng dụng DBMS, và thực hành phần mềm mới.

2.4 VIỄN THÔNG VÀ MẠNG MÁY TÍNH

2.4.1 CUỘC CÁCH MẠNG VIỄN THÔNG

Viễn thông có thể được định nghĩa như việc truyền thông tin theo phương tiện điện tử, thông thường qua khoảng cách xa nào đó. Chúng ta hiện đang ở giữa cuộc cách mạng viễn thông gồm ba thành phần: những thay đổi nhanh chóng về công nghệ truyền thông; những thay đổi về quyền sở hữu, kiểm soát và tiếp thị về

dịch vụ viễn thông; và tầm nhìn khác hẳn về cách dùng viễn thông trong tổ chức.

Trong suốt 125 năm qua, từ khi Alexander Bell phát minh ra “điện tín hát” vào năm 1876, viễn thông đã là sự độc quyền hoặc của quốc gia hoặc của các tổ chức tư nhân. Tại Mỹ, Công ti Điện tín và Điện thoại Mỹ (AT&T) là công ti độc quyền lớn nhất, thực sự cung cấp mọi dịch vụ viễn thông. Tại Châu Âu và phần còn lại của thế giới, có các cơ quan bưu điện, điện thoại và điện tín nhà nước. Tại Mỹ, độc quyền chấm dứt vào năm 1984, khi Bộ Tư pháp buộc AT&T từ bỏ độc quyền của mình để cho phép các công ti cạnh tranh bán dịch vụ và trang bị viễn thông.

Việc chấm dứt độc quyền đã mở rộng thị trường cho các công nghệ và dịch vụ viễn thông mới, từ dịch vụ đường dài giá rẻ tới thiết bị trả lời điện thoại, điện thoại di động và dịch vụ truyền thông vệ tinh. Những thay đổi này trong ngành công nghiệp viễn thông đã đi kèm với những thay đổi trong công nghệ viễn thông. Trước đây, viễn thông có nghĩa là truyền tiếng nói qua đường điện thoại. Ngày nay, nhiều việc truyền viễn thông là truyền dữ liệu số, dùng máy tính để truyền dữ liệu từ chỗ này sang chỗ khác. Các công ti điện thoại cũng đã mở rộng từ việc truyền thông tiếng nói truyền thống sang dịch vụ thông tin mới.

Người quản lí ngày nay cần hiểu cách dùng công nghệ viễn thông và cách tận dụng tối đa ích lợi của nó cho tổ chức của mình. Gần đây cuộc cách mạng viễn thông đã mở ra một tầm nhìn tính toán mới được biết tới như *kiến trúc thông tin mới*, dự kiến thu được ưu thế cạnh tranh lớn cho những công ti biết tận dụng công nghệ viễn thông.

Kiến trúc thông tin là một hình thái công nghệ thông tin đặc biệt trong tổ chức để đạt tới các mục tiêu đã chọn lựa. Kiến trúc thông tin của một tổ chức bao gồm phần cứng và phần mềm của nó, các mối nối viễn thông và các tệp dữ liệu. Trong kiến trúc thông tin

mới, các thành phần này được bố trí để đạt nhiều năng lực tính toán của tổ chức lên bàn làm việc và tạo ra việc móc nối toàn bộ công ti.

Các thành phần và chức năng của hệ thống viễn thông

Kiến trúc thông tin mới dựa trên các hệ thống viễn thông. **Hệ thống viễn thông** là tập hợp phần cứng và phần mềm tương hợp được bố trí để truyền thông tin từ vị trí nọ sang vị trí kia. Hệ thống viễn thông có thể truyền văn bản, hình ảnh đồ hoạ, tiếng nói, hay thông tin video. Hệ thống viễn thông truyền đi thông tin, thiết lập giao diện giữa nơi gửi và nơi nhận, chọn đường gửi các thông báo một cách hiệu quả nhất, và thực hiện việc xử lý sơ đẳng cho thông tin để đảm bảo rằng thông báo đúng tới người nhận đúng. Hình 2.17 minh hoạ cho các thành phần của hệ thống viễn thông điển hình. Các thành phần chủ chốt của hệ thống viễn thông là:

1. Máy tính để xử lý thông tin
2. Thiết bị cuối hay các thiết bị vào/ra để gửi và nhận dữ liệu
3. Các kênh truyền thông, các đường nối mà qua đó dữ liệu hay tiếng nói được truyền đi giữa các thiết bị gửi và nhận trong một mạng lưới.
4. Các bộ xử lý truyền thông, như modem, bộ dồn kênh và các bộ xử lý đầu cuối, cung cấp các chức năng hỗ trợ cho việc truyền và nhận dữ liệu.
5. Phần mềm truyền thông, để kiểm soát các hoạt động vào và ra và quản lý các chức năng khác của mạng truyền thông.

Kiểu tín hiệu: tương tự và số

Thông tin đi qua hệ thống viễn thông dưới dạng tín hiệu điện từ. Tín hiệu được biểu thị theo hai cách: tương tự và số. **Tín hiệu tương tự** được biểu diễn bởi một dạng sóng liên tục đi qua phương

tiện truyền thông. Tín hiệu tương tự được dùng để xử lý việc truyền tiếng nói (như điện thoại). **Tín hiệu số** là tín hiệu rời rạc chứ không phải là dạng sóng liên tục. Nó truyền dữ liệu đã mã hoá theo hai trạng thái rời rạc: bit 1 và bit 0, được biểu thị như các xung điện bật-tắt. Phần lớn các máy tính đều truyền tín hiệu số, như nhiều công ti điện thoại địa phương và một số mạng lớn.

Nếu máy tính gửi và nhận tín hiệu số qua hệ thống viễn thông tương tự, như mạng điện thoại truyền thông, thì các tín hiệu số không thể được hệ thống tương tự mang đi chình nào mà chúng còn chưa được chuyển thành tín hiệu tương tự. Thiết bị thực hiện việc chuyển đổi này được gọi là **modem** (điều chế/giải điều chế). Modem không chỉ chuyển tín hiệu số của máy tính thành dạng tương tự mà còn chuyển tín hiệu tương tự ngược trở lại dạng số cho máy tính.

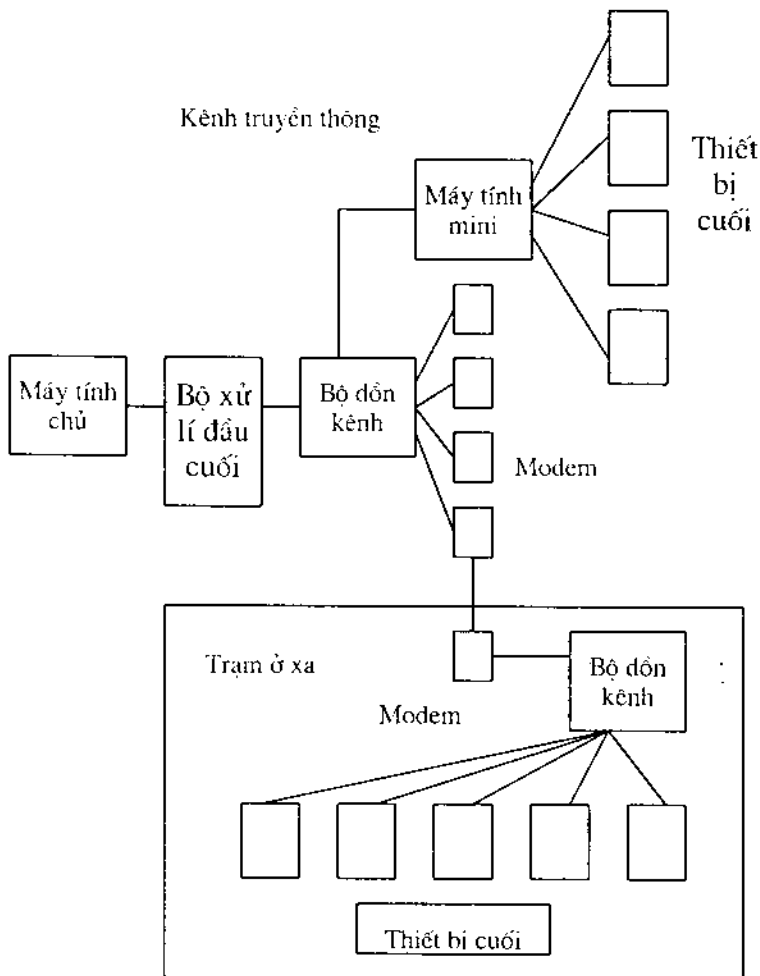
Kênh truyền thông

Kênh truyền thông là phương tiện mà qua đó dữ liệu được truyền từ một thiết bị này trong mạng sang thiết bị khác. Một kênh có thể dùng nhiều loại môi giới truyền khác nhau: dây xoắn, cáp đồng trục, sợi quang, và việc truyền không dây. Mỗi môi giới có những ưu điểm và giới hạn nhất định.

Dây xoắn bao gồm các sợi dây đồng xoắn lại thành cặp và là môi giới truyền cổ nhất. Phần lớn hệ thống điện thoại trong nhà đều dùng dây xoắn cho việc truyền tương tự. Mặc dầu dây xoắn thì giá rẻ và đã có sẵn nhưng nó lại tương đối chậm cho việc truyền dữ liệu, và việc truyền tốc độ cao gây ra nhiễu gọi là *nói chuyện xen lẫn*. Mặt khác khả năng hiện tại của cáp dây xoắn chỉ truyền được tới 10 megabit một giây.

Cáp đồng trục, giống như cáp được dùng cho ti vi, bao gồm dây đồng được cách điện dây và có thể truyền một khối lượng dữ liệu lớn hơn khả năng của dây xoắn. Cáp đồng trục chống nhiễu tốt

hơn, tốc độ nhanh hơn, thường tới 200 megabit một giây. Tuy nhiên cáp đồng trục lại dày, khó đi dây trong nhiều toà nhà và không thể hỗ trợ được cho đường thoại.



Hình 2.17 Các thành phần của hệ thống viễn thông.

Cáp sợi quang bao gồm hàng nghìn sợi thủy tinh trong, dây cỡ sợi tóc người, được đóng trong các sợi cáp. Dữ liệu được truyền theo xung ánh sáng, được gửi đi qua cáp sợi quang bằng thiết bị lade với tỉ lệ từ 500 kilobit tới vài tỉ bit trên giây. Cáp sợi quang nhanh hơn, nhẹ hơn và tuổi thọ lớn hơn đáng kể so với dây xoắn và thích hợp cho các hệ thống cần truyền những khối lượng dữ liệu lớn.

Việc truyền dẫn không dây gửi tín hiệu qua không trung hay không gian mà không dùng bất kì cách nối vật lí nào và có thể được thực hiện bằng việc dùng sóng vi ba mặt đất, vệ tinh, điện thoại di động, hay tia hồng ngoại. Hệ thống vi ba mặt đất truyền tín hiệu radio tần số cao qua bầu khí quyển và được dùng rộng rãi cho việc truyền thông điểm tới điểm, đường dài, khối lượng lớn. Vì tín hiệu sóng vi ba đi theo đường thẳng và không uốn cong theo chiều cong mặt đất nên các trạm truyền phát phải được đặt cách nhau quãng 40- 50 km, điều này làm tăng thêm chi phí cho sóng vi ba. Điều này có thể được khắc phục bằng cách dùng truyền thông sóng vi ba với vệ tinh.

Vệ tinh được sử dụng điển hình cho việc truyền thông trong các tổ chức cỡ lớn, rải rác về địa lí, thường không thể gần với nhau qua đường cáp hay sóng vi ba mặt đất. Vệ tinh di chuyển theo quỹ đạo tĩnh trên cao cách xa trái đất khoảng 36 000 km. Vệ tinh có thể được dùng làm trạm chuyển tiếp cho các tín hiệu vi ba được truyền từ các trạm mặt đất.

Các công nghệ truyền không dây khác gần đây đã được phát triển và được dùng trong những tình huống có yêu cầu tới năng lực tính toán di động. Các công nghệ này bao gồm các sóng radio tần số cao hay thấp hoặc sóng hồng ngoại. Trong các hệ không dây này, một máy tính laptop hay thiết bị cuối có trang bị đặc biệt có thể truyền dữ liệu đã mã hoá qua “địa chỉ” điện tử duy nhất của người gửi tới hàng loạt các trung tâm địa phương, miền hay quốc

gia, mà cuối cùng được móc nối với hệ thống máy tính nội bộ của người gửi.

Đặc trưng của kênh truyền thông

Các đặc trưng của kênh truyền thông giúp xác định tính hiệu quả và khả năng của hệ thống viễn thông. Những đặc trưng này bao gồm tốc độ truyền, hướng tín hiệu đi và một truyền.

Bảng 2.5 TỐC ĐỘ VÀ CHI PHÍ CỦA MÔI GIỚI TRUYỀN VIỄN THÔNG

Môi giới	Tốc độ	Chi phí
Dây xoắn	300 BPS - 10 MBPS	Thấp
Sóng vi ba	256KBPS-100MBPS	
Vệ tinh	256KBPS-100MBPS	
Cáp đồng trục	56KBPS-200 MBPS	
Cáp sợi quang	500 KBPS - 10 GBPS	Cao

BPS = bit trên giây

KBPS = kilobit trên giây

MBPS = megabit trên giây

GBPS = gigabit trên giây

Tốc độ truyền - tổng khối lượng thông tin có thể được truyền qua bất kì kênh viễn thông nào - được đo theo bit trên giây (bps). Đôi khi điều này còn được gọi là tỉ lệ baud. Baud là một sự kiện nhị phân biểu diễn cho việc thay đổi tín hiệu từ dương sang âm hay ngược lại. Vì một sự thay đổi tín hiệu, hay chu kì, là cần để truyền đi một hay vài bit trong mỗi giây, nên dung lượng truyền cho mỗi kiểu môi giới viễn thông là một hàm của tần số của nó - số các chu kì trên giây có thể được gửi đi qua môi giới đó được đo bằng hertz. Phạm vi các tần số có thể được cung cấp trên một kênh viễn thông đặc biệt được gọi là **dải thông** hay **băng thông** của nó. Băng thông chính là độ sai khác giữa các tần số cao nhất và thấp nhất mà một kênh có thể cung cấp. Phạm vi tần số càng lớn thì băng thông càng lớn và dung lượng truyền viễn thông của kênh đó càng lớn. Băng

2.5 so sánh tốc độ truyền và chi phí tương đối cho các kiểu môi giới truyền chính.

Một truyền là qui ước cho việc truyền tín hiệu. Các qui ước này là cần thiết cho các thiết bị để truyền thông khi một kí tự bắt đầu hay kết thúc. Việc **truyền dị bộ** (thường được gọi là truyền bắt đầu - dừng) truyền một kí tự mỗi lúc trên đường, mỗi kí tự được đóng khung bởi bit điều khiển - một bit bắt đầu, một hay hai bit kết thúc, và bit chẵn lẻ. Việc truyền dị bộ được dùng cho việc truyền tốc độ thấp. Việc **truyền đồng bộ** truyền đi cả nhóm kí tự đồng thời, với việc bắt đầu và kết thúc khối kí tự được xác định theo mạch thời gian của thiết bị nhận và gửi. Việc truyền đồng bộ được dùng để truyền khối lượng lớn dữ liệu với tốc độ cao.

Hướng truyền - hướng của luồng dữ liệu qua mạng viễn thông - cũng phải được xem xét tới. Trong việc truyền đơn, dữ liệu chỉ có thể đi theo một hướng vào mọi lúc. Trong việc truyền bán công (simple duplex), dữ liệu có thể đi theo hai đường nhưng mỗi lúc chỉ theo một hướng. Trong việc truyền song công (double duplex), dữ liệu có thể được truyền theo cả hai hướng đồng thời.

Các bộ xử lí truyền thông

Các bộ xử lí truyền thông như bộ xử lí đầu cuối, bộ tập trung, bộ điều khiển, bộ đón kênh và modem đều hỗ trợ cho việc truyền và nhận dữ liệu trong mạng viễn thông. **Bộ xử lí đầu cuối** là một máy tính nhỏ chuyên dụng cho việc quản lí truyền thông và được gắn với máy tính chủ, máy chính trong mạng. **Bộ xử lí đầu trước** thực hiện việc xử lí đặc biệt liên quan tới truyền thông như kiểm soát lỗi, định dạng, hiệu chỉnh, kiểm soát, chọn đường và chuyển đổi tốc độ cũng như tín hiệu. **Bộ tập trung** là một máy tính viễn thông lập trình được, thực hiện việc thu thập và cất giữ tạm thời các thông báo gửi từ thiết bị cuối chừng nào còn chưa đủ thông báo sẵn

sàng để gửi đi một cách kinh tế. **Bộ điều khiển**, thường là một máy tính mini chuyên dụng, giám sát lưu thông truyền giữa CPU và thiết bị ngoại vi như thiết bị cuối và máy in. **Bộ dồn kênh** là một thiết bị làm cho một kênh truyền thông mang việc truyền dữ liệu từ nhiều nguồn đồng thời. Bộ dồn kênh chia kênh truyền thông ra để cho kênh có thể được dùng chung cho nhiều thiết bị truyền. Bộ dồn kênh có thể chia một kênh tốc độ cao thành nhiều kênh tốc độ chậm hơn hay có thể gán cho từng nguồn truyền một lát cắt thời gian rất nhỏ để dùng kênh tốc độ cao.

Phần mềm và giao thức viễn thông

Cần có **phần mềm viễn thông** đặc biệt để điều khiển và hỗ trợ cho các hoạt động của mạng truyền thông. Phần mềm này thường trú trong máy tính chủ, bộ xử lý đầu trước và các bộ xử lý khác trong mạng. Chức năng chính của phần mềm viễn thông là kiểm soát mạng, kiểm soát truy nhập, kiểm soát việc truyền, phát hiện/sửa lỗi và an toàn.

Mạng viễn thông chứa rất nhiều thành phần phần cứng và phần mềm đa dạng cần làm việc với nhau để truyền thông tin. Các thành phần khác nhau trong mạng có thể trao đổi với nhau được bởi việc thừa nhận một tập các qui tắc chung làm cho chúng “nói chuyện” được với nhau. Tập các qui tắc và thủ tục cai quản việc truyền giữa hai điểm trong mạng được gọi là một **giao thức**. Mỗi thiết bị trong mạng phải có khả năng diễn giải giao thức của các thiết bị khác.

2.4.2 CÁC KIỂU MẠNG TRUYỀN THÔNG

Hiện có một số cách thức khác nhau để tổ chức các thành phần viễn thông tạo nên một mạng lưới và do đó cung cấp nhiều cách phân loại mạng. Mạng có thể được phân loại theo hình dáng

của chúng hay còn gọi là **tô pô mạng**. Mạng cũng có thể được phân loại theo phạm vi địa lý và kiểu dịch vụ được cung cấp. Mục này sẽ mô tả những cách khác nhau để nhìn vào mạng.

Tô pô mạng. Mạng hình sao bao gồm một máy tính chủ được nối với một số máy tính nhỏ hay thiết bị cuối. Hình thái này có ích cho những ứng dụng mà việc xử lý nào đó thì phải tập trung và xử lý khác có thể được thực hiện cục bộ. Một vấn đề với mạng hình sao là tính mong manh của nó. Nếu máy tính chủ trục trặc thì cả mạng bị ảnh hưởng vì tất cả các điểm trong mạng đều phải truyền về máy chủ.

Mạng đường bus nối một số máy tính bằng một mạch làm bằng dây xoắn, cáp đồng trục hay cáp sợi quang. Tất cả các tín hiệu đều được phát đi theo cả hai hướng tới toàn bộ mạng, cùng với phần mềm đặc biệt để xác định thành phần nào nhận từng thông báo. Không có máy chủ trung tâm để kiểm soát mạng. Thông báo có thể đi theo cả hai hướng dọc theo cáp. Nếu một máy tính trong mạng hỏng thì các thành phần khác không bị ảnh hưởng. Hình thái này thường được dùng cho mạng cục bộ (LAN).

Mạng vòng, cũng giống như mạng đường bus, không dựa vào máy tính chủ trung tâm và không bị sập nếu một trong các máy tính thành phần bị trục trặc. Mỗi máy tính trong mạng có thể trao đổi trực tiếp với bất kỳ máy tính nào khác, và từng máy xử lý ứng dụng của riêng mình một cách độc lập. Tuy nhiên, trong hình thái vòng, việc nối dây, cáp hay cáp quang nối lại thành vòng tròn đóng. Dữ liệu được truyền dọc theo vòng từ máy tính nọ sang máy tính kia và bao giờ cũng đi theo một hướng.

Tổng đài nội bộ và mạng cục bộ LAN

Tổng đài nội bộ (PBX) là một máy tính chuyên dụng được thiết kế để giải quyết và chuyển mạch những lời gọi điện thoại văn phòng tại vị trí công ti. Trong khi các PBX đầu tiên chỉ thực hiện

chức năng chuyển mạch có giới hạn, thì các PBX ngày nay có thể tải cả tiếng nói và dữ liệu để tạo ra mạng cục bộ. Bây giờ chúng có thể ghi nhớ, truyền, giữ và quay số lại các lời gọi điện thoại. PBX cũng có thể được dùng để chuyển mạch thông tin số giữa máy tính và các thiết bị khác. Ưu điểm của PBX so với các tùy chọn nối mạng cục bộ khác là ở chỗ chúng sử dụng các đường điện thoại hiện có và không cần dây đặc biệt.

Mạng cục bộ (LAN) đòi hỏi các kênh truyền thông riêng của nó. Mạng cục bộ bị giới hạn trong phạm vi địa lý trong bán kính vài trăm mét, mặc dầu chúng thường có thể mở rộng hơn bằng việc nối với các mạng cục bộ khác. Chúng có dải thông cao, cho nên dung lượng truyền cao hơn PBX, đạt tới 100 megabit trên giây. LAN hoàn toàn do người dùng cuối kiểm soát, duy trì và vận hành. LAN cho phép tổ chức dùng chung phần cứng và phần mềm đắt tiền. Cách dùng thông dụng nhất của LAN là để nối các máy vi tính bên trong một toà nhà hay văn phòng để dùng chung thông tin và các thiết bị ngoại vi đắt như máy in laser.

Bộ phục vụ tệp hành động như một thủ thư, cất giữ các chương trình và tệp dữ liệu khác nhau cho người dùng trên mạng. Bộ phục vụ xác định ai có quyền truy nhập vào cái gì và theo trình tự nào. Bộ phục vụ có thể là máy vi tính mạnh với dung lượng đĩa cứng lớn, trạm làm việc, máy tính mini hay máy tính lớn, mặc dầu máy tính chuyên dụng cũng có sẵn cho mục đích này. Bộ phục vụ điển hình có chứa **hệ điều hành mạng** quản lý bộ phục vụ và chọn đường và quản lý truyền thông trên mạng.

Máy cửa ngõ mạng nối LAN với mạng công cộng, như mạng điện thoại, hoặc tới mạng công ti để cho LAN có thể trao đổi thông tin với mạng bên ngoài nó. Máy cửa ngõ nối chung là bộ xử lý truyền thông (phần cứng và phần mềm) mà có thể nối các mạng không tương tự bằng việc dịch từ tập giao thức này sang tập giao thức khác. (Cầu nối hai mạng thuộc cùng kiểu. *Bộ chọn đường*

được dùng để chọn đường thông báo qua nhiều LAN nối với nhau hay qua mạng diện rộng.)

Mạng diện rộng (WAN)

Mạng diện rộng (WAN) trải trên khoảng cách địa lý rộng, từ nhiều cây số tới cả lục địa. Các công ti chuyển phát (các công ti được chính phủ cấp phép cung cấp dịch vụ truyền thông cho công cộng) xác định ra tỉ lệ truyền hay liên nối giữa hai đường, nhưng khách hàng chịu trách nhiệm về nội dung viễn thông và quản lý. Các tổ chức tự lo liệu việc thiết lập đường thông báo hiệu quả nhất và xử lý lỗi, hiệu chỉnh, giao thức và quản lý viễn thông.

WAN có thể bao gồm việc tổ hợp các đường chuyển mạch và dành riêng, truyền thông vì ba và vệ tinh. **Đường chuyển mạch** là đường điện thoại mà người ta có thể truy nhập tới từ thiết bị cuối của mình để truyền dữ liệu cho máy tính khác, cuộc gọi được chọn đường hay chuyển mạch qua đường đi tới nơi nhận đã định. **Đường dành riêng**, hay không chuyển mạch, là liên tục có sẵn cho việc truyền, và người thuê bao trả tiền theo tỉ lệ cố định cho toàn bộ việc truy nhập vào đường này. Các đường này có thể được thuê bao hay mua từ các công ti chuyển phát hay người bán môi giới truyền thông. Đường dành riêng thường được dùng để truyền dữ liệu với tốc độ cao hơn đường chuyển mạch và thích hợp hơn với việc truyền khối lượng cao. Đường chuyển mạch, mặt khác, lại rẻ và thích hợp hơn với các ứng dụng khối lượng thấp và chỉ cần thỉnh thoảng mới truyền.

Mạng giá trị gia tăng (VAN)

Mạng giá trị gia tăng là một phương án cho các tổ chức thiết kế và quản lý mạng riêng của mình. **Mạng giá trị gia tăng (VAN)** là mạng tư nhân, nhiều đường, chỉ truyền dữ liệu, do bên thứ ba quản lý, mà có thể cung cấp việc tiết kiệm về chi phí dịch vụ và

quản lí mạng vì chúng được dùng cho nhiều tổ chức. Tổ chức thiết lập và quản lí VAN bán việc thuê bao cho các tổ chức khác muốn dùng nó. Người thuê bao chỉ phải trả khoản tiền cho phần dữ liệu họ truyền cộng thêm phí thuê bao.

Thuật ngữ *giá trị gia tăng* nói tới “giá trị” phụ được thêm vào cho kênh truyền thông bởi những dịch vụ viễn thông và tính toán mà những mạng này cung cấp cho khách hàng. Người thuê bao tiết kiệm được bởi vì họ không phải đầu tư vào trang bị mạng và phần mềm hay thực hiện việc kiểm tra lỗi riêng, hiệu chỉnh, chọn đường và chuyển đổi giao thức. Bên cạnh đó, chi phí cho việc dùng mạng được chia sẻ cho nhiều người dùng. Chi phí thực có thể thấp hơn nếu khách hàng phải thuê đường riêng hay dịch vụ vệ tinh.

Một cách khác mà mạng giá trị gia tăng cho phép tiết kiệm là qua **chuyển mạch gói**. Chuyển mạch gói chia một khối văn bản dài thành những bó dữ liệu nhỏ, chiều dài cố định gọi là **gói**. VAN tiếp tục dùng nhiều kênh truyền thông để thu thập thông tin và gửi các gói. Mỗi gói đều độc lập đi qua mạng (điều này tương phản với một tổ chức dùng đường thuê bao, chẳng hạn, trong một giờ và rồi không dùng nó trong ba hay bốn giờ). Các gói dữ liệu nhằm vào một nguồn có thể được chọn đường qua nhiều con đường khác nhau trong mạng, và rồi được lắp ráp lại trong thông báo gốc khi chúng đạt tới nơi nhận. Vậy, chuyển mạch gói làm cho tiện nghi truyền thông được sử dụng đầy đủ hơn bởi nhiều người hơn. **Role khung** là một biến thể nhanh hơn và rẻ hơn của chuyển mạch gói. Nó là một dịch vụ mạng dùng chung, gói các dữ liệu vào trong các “khung”, cũng tương tự như gói. Tốc độ truyền role khung có thể đạt tới 1.544 megabit trên giây. Tuy nhiên role khung không thực hiện sửa lỗi. Điều này là vì nhiều đường số thực hiện nay trong và rõ hơn các đường truyền quá khứ và mạng cũng khéo sửa lỗi khi truyền. Role khung chỉ làm việc thành công trên đường truyền tin cậy mà không yêu cầu thường phải truyền lại vì lỗi.

Phần lớn các công ti hiện nay vẫn dùng các mạng tách rời cho tiếng nói, dịch vụ đường tư, và dữ liệu vốn được nhiều công nghệ khác nhau hỗ trợ. Một công nghệ mạng đang nổi lên được gọi là **một truyền dữ bộ (ATM)** có thể khắc phục được một số vấn đề này vì nó có thể chuyển mạch không ngừng và linh động giữa tiếng nói, dữ liệu, hình ảnh và video cho các người dùng. Công nghệ ATM gói thông tin thành các “ngăn” đều nhau, mỗi ngăn có 53 nhóm tám bit, loại bỏ nhu cầu chuyển đổi giao thức. Nó có thể truyền dữ liệu giữa các máy tính của các nhà sản xuất khác nhau và cho phép dữ liệu được truyền với bất kì tốc độ nào mà mạng giải quyết được. ATM hiện yêu cầu cấp sợ quang, nhưng nó có thể đạt tới tốc độ vượt quá 2.5 GBPS.

2.4.3 TỔ CHỨC SỬ DỤNG VIỄN THÔNG

Làm thuận tiện các ứng dụng: thư điện tử, thư tiếng nói, FAX và dịch vụ điện tử

Thư điện tử, hay email (đọc là i meo), là việc trao đổi thông báo giữa các máy tính. Một người có thể dùng máy vi tính có gắn modem để gửi đi các ý kiến hay văn bản dài hơn chỉ bằng việc gõ tên của người nhận thông báo. Nhiều tổ chức vận hành hệ thống thư điện tử riêng của mình, nhưng một số công ti lớn cũng cung cấp dịch vụ này. Email loại bỏ các chi phí điện thoại đường dài đắt đỏ, mở rộng truyền thông giữa các bộ phận khác nhau của tổ chức.

Hệ thống **thư tiếng nói** số hoá thông báo lời nói của người nhận, truyền nó qua mạng, và cất giữ thông báo này trên đĩa cho việc tìm kiếm về sau. Khi người nhận đã sẵn sàng nghe thì thông báo được chuyển sang dạng tiếng nói.

Máy **FAX** có thể truyền các tài liệu có chứa cả văn bản lẫn đồ hoạ qua đường điện thoại thông thường. Máy gửi FAX quét và

số hoá ảnh tài liệu. Tài liệu được số hoá tiếp đó được truyền qua mạng và được tái tạo lại dưới dạng in ra tại máy FAX nhận. Kết quả xử lý là một bản sao, hay bản fax, của bản gốc.

Hội nghị từ xa và hội nghị video

Con người có thể gặp nhau theo dạng điện tử - cho dù họ ở xa nhau hàng nghìn cây số - bằng việc dùng hội nghị từ xa hay hội nghị video. **Hội nghị từ xa** cho phép một nhóm người tổ chức “hội nghị” đồng thời nhờ có phần mềm truyền thông cho nhóm điện thoại hay thư điện tử. Hội nghị từ xa có khả năng để người tham dự thấy lẫn nhau qua màn hình video có tên là hội nghị video từ xa hay **hội nghị video**. Hội nghị video thường đòi hỏi phòng hội nghị đặc biệt với các máy quay video, microphone, màn hình tivi và máy tính được trang bị thiết bị giải mã chuyển đổi tín hiệu ảnh video và sóng âm thanh tương tự thành tín hiệu số rồi nén chúng để truyền qua kênh truyền thông. Một thiết bị giải mã khác ở đầu nhận sẽ chuyển tín hiệu số lại thành tín hiệu tương tự cho màn hình nhận.

Trao đổi dữ liệu điện tử

Trao đổi dữ liệu điện tử (EDI) là việc trao đổi trực tiếp từ máy tính sang máy tính, giữa hai tổ chức, về các tài liệu giao dịch kinh doanh như hoá đơn, vận đơn hay đơn mua hàng. EDI tiết kiệm tiền của và thời gian vì các giao dịch có thể được truyền từ hệ thống tin này sang hệ thống tin khác qua mạng viễn thông, xoá bỏ việc in ra và xử lý giấy tờ tại đầu này và việc vào dữ liệu tại đầu kia. EDI khác với thư điện tử ở chỗ nó truyền một giao dịch có cấu trúc (với các trường phân biệt như ngày giao dịch, khối lượng giao dịch, tên người gửi, tên người nhận) ngược với thông báo văn bản phi cấu trúc như bức thư.

2.4.4 TÍNH NỐI VÀ CHUẨN

Viễn thông và mạng rất có thể làm tăng năng suất và tăng ưu thế cạnh tranh khi thông tin số hoá có thể chuyển qua một mạng nhện các mạng điện tử, nối các loại máy móc, con người, cảm biến, cơ sở dữ liệu, các bộ phận chức năng, phòng ban và nhóm làm việc. Mặc cho tất cả những tiến bộ trong xử lý tại bàn, phần mềm thân thiện người dùng và công nghệ viễn thông, mục tiêu này khó mà thực hiện được bởi vì nhiều loại khác nhau về các hệ phần cứng, phần mềm và truyền thông làm cho chúng không có khả năng làm việc với nhau. Chúng thiếu **tính nối** - khả năng của máy tính và các thiết bị dựa trên máy tính để trao đổi lẫn với nhau và "chia sẻ" thông tin theo cách thức có nghĩa mà không có sự can thiệp của con người.

Vấn đề về tính nối

Sau đây là một số thí dụ về việc thiếu tính nối:

- Các máy tính mini để bàn thường không thể dùng dữ liệu từ máy tính lớn của công ti, thường không thể dùng chung thông tin giữa các máy vì tính nhãn hiệu khác nhau, và nhiều lúc không thể dùng chung thông tin một cách có ý nghĩa ngay giữa các phần khác nhau của cùng hệ điều hành trên cùng máy vi tính.
- Một số công ti có nhiều hệ thống thư điện tử khác nhau bên trong mà lại không thể nào trao đổi lẫn nhau được. Việc trao đổi giữa các tổ chức bị phức tạp lên bởi việc thiếu những sản phẩm hỗ trợ cho các chuẩn thư điện tử.
- IBM, cũng như các nhà cung cấp phần cứng khác, bán máy và phần mềm nhưng lại không thể trao đổi với các máy và phần mềm khác với cùng nhãn hiệu bởi vì các thiết kế phần cứng và hệ điều hành khác nhau.

- Các công ti hoạt động ở nước ngoài gặp rất nhiều khó khăn khi xây dựng mạng toàn cầu vốn gắn liền mạnh các tổ chức của họ với nhau. Các quốc gia khác nhau có các hạ tầng viễn thông khác nhau, nhiều khi thuộc vào các tổ chức độc quyền về điện tín, điện thoại, bưu điện quốc gia, vốn dùng các chuẩn kết mạng khác nhau.

Có nhiều lí do tại sao máy tính và hệ thống tin lại đạt tới những tầm cao chóng mặt như vậy về tính không tương hợp. Các máy tính cá nhân đã được thiết kế từ lâu trước khi mạng máy tính được xây dựng. Trước năm 1980, chẳng có chuẩn nào cho các nhà chế tạo phần cứng hay phần mềm: người mua trang thiết bị thiếu sức mạnh thị trường để yêu cầu các chuẩn; còn bản thân các nhà cung ứng phần cứng và phần mềm thì lại cố vũ cho sự phân biệt sản phẩm. Hơn nữa, tiến trình thiết lập chuẩn phần lớn là chính trị và bao gồm nhiều nhóm lợi ích mạnh, như các hiệp hội trang thiết bị công nghiệp thuộc khu vực tư nhân, Chính phủ liên bang Mỹ, và các nhóm chuyên môn như Viện các Kỹ sư Điện và Điện tử (IEEE), Viện Chuẩn Quốc gia Mỹ (ANSI), Tổ chức Tiêu chuẩn Quốc tế (ISO) và Ủy ban Tư vấn Quốc tế về Điện tín và Điện thoại (CCITT).

Các khía cạnh của tính nối

Tính nối bao gồm nhiều vấn đề hơn chỉ là kết mạng. Có nhiều phẩm chất khác nhau mà một hệ thống tin mang tính nối phải có. Chúng ta hãy khám phá một số thuật ngữ được dùng để mô tả cho các khía cạnh khác nhau của tính nối.

Khả chuyển ứng dụng là khả năng vận hành cùng chương trình phần mềm trên các kiểu phần cứng máy tính khác nhau. Chẳng hạn như dùng phần mềm trang tính Lotus1-2-3 trên máy vi tính tại chỗ làm việc, ghi kết quả lên tệp, đem nó về nhà và dùng máy Macintosh để hoàn thành còn gviệc đó.

Di trú là khả năng chuyển phần mềm từ thế hệ phần cứng này sang thế hệ phần cứng khác mạnh hơn. Phần lớn phần mềm máy vi tính đều đưa ra khả năng tương hợp về sau. Chẳng hạn bạn có thể chạy Word 6 trên các môi trường Windows 3.x hoặc 95, 98. Tương hợp về trước có vấn đề nhiều hơn. Nhiều phần mềm máy vi tính được thiết kế để vận hành dưới DOS không thể vận hành trên Windows.

Xử lý hợp tác phân chia các nhiệm vụ tính toán giữa các máy tính lớn, máy mini, máy vi tính hay trạm làm việc để giải quyết một vấn đề chung. Xử lý hợp tác là vấn đề nổi bật vì các máy khác nhau phải được nối mạng và lập trình để cho chúng có thể làm việc với nhau trên một ứng dụng.

Tính khả chuyển thông tin là việc dùng chung các tệp máy tính giữa các kiểu phần cứng khác nhau và các ứng dụng phần mềm khác nhau. Hiện thời, việc dùng chung tệp là có thể giữa các ứng dụng như xử lý văn bản vốn tạo ra một tệp chuẩn như tệp văn bản ASCII. Những tệp văn bản này có thể được truyền vào trong trang tính hay ứng dụng cơ sở dữ liệu, nhưng việc truyền có thể xảy ra chỉ trên cùng nền phần cứng hay lớp máy. Bởi vì các máy khác nhau và các lớp máy dùng các sơ đồ mã hoá khác nhau (chẳng hạn, máy tính lớn dùng EBCDIC, một số máy nhỏ dùng ASCII), cho nên thậm chí tính khả chuyển thông tin có thể cũng thành khó khăn.

Liên tác là khả năng của phần mềm vận hành trên hai loại phần cứng khác nhau, nhưng lại biểu lộ cho người dùng một giao diện đồng nhất, và thực hiện cùng các nhiệm vụ. Chẳng hạn, phần mềm xử lý văn bản Microsoft Word có chức năng tương tự và trông cũng như nhau cả trên máy vi tính Macintosh và IBM, nhưng các phiên bản khác nhau của phần mềm này là cần để chạy trên từng kiểu máy.

Liên tác, cũng như tính khả chuyển thông tin và ứng dụng, đòi hỏi hệ thống mở. **Hệ thống mở** được xây dựng từ các chuẩn hệ điều hành công cộng, không thuộc sở hữu riêng, chuẩn giao diện

người dùng, chuẩn ứng dụng, và các giao thức kết mạng. Trong hệ thống mở, phần mềm có thể vận hành trên các nền phần cứng khác nhau và theo nghĩa đó có thể là "khả chuyển". Sẽ rất đắt đỏ để đạt tới tính khả chuyển ứng dụng nếu không có hệ điều hành được dùng chung.

Trong thế giới máy vi tính, phần lớn các máy vi tính tương hợp IBM dùng Microsoft/IBM DOS hay Microsoft Windows. Các nguyên lý vận hành của chúng đã được phổ biến, và chúng có thể được những người làm máy vi tính khác IBM dùng với một phí sử dụng tối thiểu. Nhưng ngược lại, phần lớn các máy tính mini và máy tính lớn đều có hệ điều hành tài sản riêng mà không thể nào được dùng cho các lớp máy khác vốn có nguyên lý vận hành bị dấu kín với công luận và được bảo vệ chặt bởi phí sử dụng cao.

Có lẽ chìa khoá cho hệ thống mở thực là hệ điều hành UNIX. UNIX bắt nguồn từ nghiên cứu phòng thí nghiệm của AT&T trong những năm 1970 thông qua các kĩ sư muốn có một hệ điều hành phân thời mạnh mà có thể vận hành trên nhiều loại phần cứng máy tính. Từ đó, UNIX đã đi tới thống trị thị trường kĩ nghệ cho trạm làm việc và bắt đầu được dùng cho một số kiểu máy vi tính. Tuy nhiên, có những phiên bản khác nhau của UNIX, và không phiên bản nào đã được chấp nhận như chuẩn hệ thống mở.

Vậy tính nối đúng đắn đòi hỏi rất nhiều điều hơn đơn giản chỉ có nối dây các máy tính khác nhau hay cung cấp việc truy nhập có giới hạn vào nhiều máy tính khác nhau. Tính nối yêu cầu các hệ điều hành chung, các chuẩn viễn thông chung, và thậm chí các chuẩn giao diện người dùng chung, một màn hình trông và cảm thấy tương tự qua các ứng dụng phần mềm khác nhau.

Chuẩn để đạt tới tính nối

Việc đạt tới tính nối yêu cầu các chuẩn về kết mạng, hệ điều hành và giao diện người dùng. Cho tới nay, vẫn không có các chuẩn thống nhất đảm bảo rằng các hệ thông tin có thể đạt tới các thuộc

tính về tính nối đã được mô tả ở trên, nhưng một vài chuẩn quả có tồn tại và các chuẩn khác đang được khởi xướng. Mục này mô tả cho những chuẩn quan trọng nhất đang được sử dụng ngày nay.

MÔ HÌNH VỀ TÍNH NỐI CHO CÁC MẠNG

Bởi vì nhiều mối quan tâm được bao hàm trong tính nối và việc lập chuẩn nên có các mô hình khác nhau để đạt tới tính nối trong các mạng viễn thông. **Mô hình tham chiếu** là một khuôn khổ chung cho việc suy nghĩ về một vấn đề. Nó là việc chia ra một cách logic một hoạt động nào đó (như truyền thông) thành một số bước hay bộ phận phân biệt. Các giao thức đặc biệt được cần tới để thiết lập mô hình tham chiếu. **Giao thức** là một trình bày vốn giải thích cách một nhiệm vụ đặc biệt, như truyền dữ liệu, sẽ được thực hiện. Các mô hình tham chiếu và giao thức trở thành **chuẩn** khi chúng được các nhóm lập chuẩn quan trọng chấp nhận hay khi công nghiệp xây dựng ra các sản phẩm hỗ trợ cho các mô hình và giao thức đó.

Tính nối mạng cũng có thể được đạt tới mà không có mô hình tham chiếu hay giao thức bằng cách dùng các máy cửa ngõ. Các công ti phát triển ra các máy cửa ngõ giữa hai mạng khác nhau khi không thể hay quá đắt không tích hợp chúng lại được bằng việc tuân thủ theo các mô hình tham chiếu hay chuẩn. Tuy nhiên, cửa ngõ thì đắt trong việc xây dựng và bảo trì, và chúng có thể chậm và kém hiệu quả. Bây giờ chúng ta mô tả cho các mô hình nối mạng quan trọng nhất.

Mô hình **Liên nối hệ thống mở (OSI)** là một mô hình tham chiếu quốc tế được tổ chức tiêu chuẩn quốc tế phát triển để nối các kiểu máy tính và mạng khác nhau. Nó là một khuôn khổ để định nghĩa các chức năng cần tới trong các phiên viễn thông giữa hai hay nhiều máy tính. Nó được thiết kế để hỗ trợ cho việc nối mạng toàn cầu với khối lượng lớn các xử lý giao tác.

Máy chủ A
Người dùng A

Máy chủ B
Người dùng B

Tầng OSI

Tầng OSI

7	Ứng dụng Các chức năng người dùng chuyên dụng như hệ điều hành mạng, truyền tệp, thư điện tử
6	Trình bày Định dạng dữ liệu cho việc trình bày, cung cấp chuyển đổi mã
5	Phiên Thiết lập truyền thông giữa các trạm trên mạng
4	Vận tải Đảm bảo chuyển giao dữ liệu đầu nọ tới đầu kia tin cậy
3	Mạng Chọn đường và chuyển tiếp các gói thông tin qua mạng diện rộng
2	Móc nối dữ liệu Đóng gói và truyền các gói thông tin, kiểm tra lỗi
1	Vật lí Truyền dữ liệu thô qua phương tiện truyền thông

7	Ứng dụng Các chức năng người dùng chuyên dụng như hệ điều hành mạng, truyền tệp, thư điện tử
6	Trình bày Định dạng dữ liệu cho việc trình bày, cung cấp chuyển đổi mã
5	Phiên Thiết lập truyền thông giữa các trạm trên mạng
4	Vận tải Đảm bảo chuyển giao dữ liệu đầu nọ tới đầu kia tin cậy
3	Mạng Chọn đường và chuyển tiếp các gói thông tin qua mạng diện rộng
2	Móc nối dữ liệu Đóng gói và truyền các gói thông tin, kiểm tra lỗi
1	Vật lí Truyền dữ liệu thô qua phương tiện truyền thông

Thông tin

Hình 2.18 Mô hình tham chiếu Liên nối hệ thống mở (OSI).

OSI làm cho máy tính được nối với mạng để trao đổi với bất kì máy tính nào khác trên cùng mạng hay khác mạng, bất kể tới nhà chế tạo. Để thiết lập tính nối này, mô hình OSI chia tiến trình viễn thông ra thành bảy tầng (xem Hình 2.18). Mỗi tầng trong mô hình OSI đều được xác định bởi các chức năng truyền thông và giải quyết cho một khía cạnh đặc biệt của tiến trình truyền thông. Mỗi tầng đều có một hay vài giao thức liên kết. Một giao thức nhiều tầng có ưu thế là mỗi tầng đều độc lập với các tầng khác cho nên nó có thể được thay đổi mà không làm ảnh hưởng tới các tầng khác.

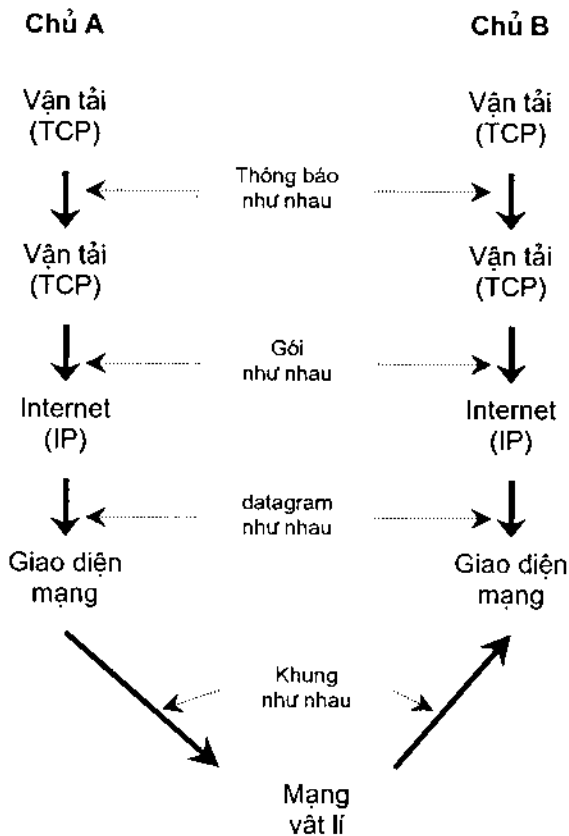
Hai máy tính khác nhau cùng dùng chuẩn OSI sẽ có phần mềm và phần cứng tương ứng với từng tầng của mô hình OSI. Một thông báo được gửi từ máy tính này sang máy tính khác sẽ đi qua cả bảy tầng, bắt đầu với tầng ứng dụng của máy tính gửi và truyền tới tầng vật lí của máy tính gửi. Thế rồi nó sẽ du hành qua kênh truyền thông và đi vào máy tính nhận, đi lên qua bảy tầng trong máy đó. Tiến trình này được đảo ngược khi máy tính nhận đáp ứng.

Chẳng hạn, nếu một viên chức tại một ngân hàng địa phương cần thông tin về tài khoản của một khách hàng đặc biệt được lưu giữ trong máy tính trung tâm của ngân hàng, thì người ấy sẽ đưa vào các chỉ thị để tìm kiếm bản ghi tài khoản khách hàng trên thiết bị cuối của mình đang kiểm soát ở tầng 7, tầng ứng dụng.

Tầng trình bày (tầng 6) sẽ thay đổi dữ liệu vào này thành định dạng cho việc truyền. Tầng 5 (tầng phiên) khởi đầu cho phiên làm việc. Tầng 4 (tầng vận chuyển) kiểm tra chất lượng của thông tin đi từ người dùng sang nút chủ. Tầng 3 và 2 (tầng mạng và móc nối dữ liệu) truyền dữ liệu qua tầng 1 (tầng vật lí). Khi thông báo đạt tới máy tính chủ, thì điều khiển chuyển lên các tầng ngược lại tới người dùng, đảo ngược lại trình tự.

Mô hình **Transmission Control Protocol/Internet Protocol (TCP/IP)** là mô hình tham chiếu do Bộ Quốc phòng Mỹ phát triển. Được đưa ra vào năm 1972 tiếp nối với nghiên cứu và phát triển mạng do Cơ quan Dự án Nghiên cứu Quốc phòng Năng cao

(DARPA) tiến hành, mục đích của nó là giúp cho các nhà khoa học móc nối các máy tính tách biệt. Bởi vì nó là một trong những mô hình tham chiếu truyền thông cổ nhất, nên TCP/IP vẫn được sử dụng rộng rãi, đặc biệt ở Mỹ (Châu Âu có khuynh hướng thích OSI). Hình 2.19 chỉ ra rằng TCP/IP có mô hình tham chiếu năm tầng.



Hình 2.19 Mô hình tham chiếu TCP/IP.

1. Tầng lưới: Định nghĩa các đặc trưng truyền điện cơ bản được sinh ra trong khi truyền
2. Giao diện mạng: Giải quyết vấn đề định địa chỉ, thường trong hệ điều hành, cũng như giao diện giữa máy tính khởi đầu và mạng.
3. Internet (IP): Giải quyết truyền thông hệ thống sang hệ thống. Tầng này là tiến trình chuyển giao các nút dữ liệu (datagram) không nối, tự lập vốn không phụ thuộc gì vào mạng để báo nhận thông báo. Datagram là một đơn vị thông tin bao gồm đoạn đầu và đoạn văn bản. Internet Protocole nhận các datagram từ TCP và truyền chúng qua Internet
4. Giao thức kiểm soát truyền (TCP): Thực hiện việc chuyển tải. TCP hỗ trợ việc truyền thông từ chương trình sang chương trình tại mức người dùng cuối. Giao thức kiểm soát truyền hỗ trợ cho việc truyền thông tin tin cậy độc lập với công việc máy tính và các tầng cao hơn (như email hay log-on)
5. Ứng dụng: Cung cấp chức năng người dùng cuối bởi việc dịch các thông báo thành phần mềm người dùng/máy chủ cho việc trình bày trên màn hình

CÁC CHUẨN KẾT MẠNG KHÁC

Các chuẩn cũng đã được phát triển cho việc truyền dữ liệu số qua mạng chuyển mạch công cộng. **Mạng số thức dịch vụ tích hợp Integrated Services Digital Network (ISDN)** là một chuẩn mạng chuyển mạch công cộng số thức quốc tế cho việc truyền tiếng nói, video, và dữ liệu qua đường điện thoại. Nó là một kế hoạch đã được sự chấp thuận quốc tế để di chuyển mạng điện thoại công cộng sang việc dùng chuẩn công nghệ số thức. CCITT (Ủy ban Tư vấn về Điện tín và Điện thoại Quốc tế), vốn chịu trách nhiệm cho việc đề ra các chuẩn cho việc thực hiện ISDN, định nghĩa ISDN như

một mạng cung cấp việc nối số thức đầu cuối tới đầu cuối để hỗ trợ cho một phạm vi rộng các dịch vụ, kể cả tiếng nói và dịch vụ phi tiếng nói. Theo lời của nhiều người quan sát công nghiệp thì ISDN là phát triển mới lớn nhất trong truyền thông điện thoại.

Ta hãy xét những giới hạn của LAN và PBX. Ngay bây giờ, bạn không thể gọi điện thoại trên mạng cục bộ được. Bạn không thể gửi FAX hay ảnh video thật thuận tiện trên LAN của mình. Nếu bạn mua PBX để cải thiện cả dịch vụ thoại và tính nối máy tính trong tổ chức của mình, thì bạn phải chọn và duy trì thiết bị, và dịch vụ chấm dứt ngay khi bạn rời khỏi toà nhà.

Một cách khác, ta hãy hình dung việc tổ hợp vào một dịch vụ duy nhất các tính năng sau:

- Nối ghép toàn bộ tiếng nói, dữ liệu và video với bất kì điểm nào trên thế giới.
- Nối ghép số thức hoàn toàn với bất kì thiết bị số thức nào khác trên thế giới.
- Sử dụng đồng thời thiết bị tiếng nói, video và thiết bị số thức.
- Tự do hoàn toàn để di chuyển thiết bị và con người mà không phải nối dây lại các toà nhà hay cáp ddặc biệt, và với một chuẩn vật lí duy nhất.
- Việc định nghĩa do người dùng kiểm soát về đường video, số thức và dữ liệu. Bạn có thể dùng một đường cho dịch vụ số thức cá nhân giờ này, rồi đặt cấu hình lại cùng đường điện thoại đó như đường 800 wat đi tới vào giờ tiếp, rồi giờ tiếp nữa định nghĩa lại đường đó như đường video mang hình ảnh của phiên hội nghị video theo nhóm.

Tóm lại, ISDN là mạng cho mọi thứ. Mục đích và hứa hẹn của ISDN là cung cấp một mạng chức năng hơn để chuyển vận mọi loại thông tin số thức, bất kể tới nguồn hay nơi nhận của nó. Viễn cảnh tương lai của mạng điện thoại chuyển mạch là biến thành đường siêu cao tốc số thức. ISDN cung cấp tính nối dữ liệu phổ dụng dưới dạng số thức. Các modem không cần tới nữa với ISDN

ngoại trừ khi cần nối với người dùng mạng phi ISDN. ISDN dùng cáp dây đồng xoắn chuẩn để nối các thiết bị để bàn với bộ tập trung ở mức cả toà nhà. Một thiết bị chuyển mạch văn phòng trung tâm tạo ra mọi ghép nối với thế giới bên ngoài. Với một số thay đổi công nghệ đơn giản, dây điện thoại đồng xoắn thông thường có thể được sử dụng đầy đủ hơn mà không phải nối cáp tốn kém.

CHUẨN TRUYỀN SỢI QUANG: FDDI

Giao diện dữ liệu phân bố sợi (FDDI) là chuẩn cho việc truyền dữ liệu với tốc độ 100 MBPS qua cáp sợi quang trong cấu hình vòng. FDDI LAN dùng một cấu hình vòng sợi quang kép. Dùng hai vòng thay vì một vòng làm tăng độ tin cậy.

CHUẨN HỆ ĐIỀU HÀNH: PHONG TRÀO HỆ THỐNG MỞ

Trận chiến về các chuẩn hệ điều hành tập trung quanh UNIX. UNIX là hệ điều hành không sở hữu riêng duy nhất, và bất luận người bán nào. Tuy nhiên, nỗ lực để làm cho UNIX thành chuẩn hệ điều hành đã gặp phải sự chống đối đáng kể. Các hệ điều hành và mớic nối truyền thông chuẩn hoá sẽ khừ bỏ đi sự phụ thuộc của tổ chức vào một nhà cung cấp phần cứng đặc biệt, như IBM hay DEC. Hơn nữa, UNIX đi ngược trực tiếp lại các kế hoạch tiếp thị của nhiều nhà cung cấp lớn. Chẳng hạn, nếu UNIX trở thành chuẩn, thì ít có nhu cầu dùng hệ điều hành Windows NT của Microsoft Corporation hay OS/2 của IBM. UNIX có thể chuyển giao tính khả chuyển ứng dụng, phân thời, truyền tệp và giao diện người dùng chung.

Phần **3**

XÂY DỰNG HỆ THỐNG TIN

3.1 HỆ THỐNG XEM NHƯ SỰ THAY ĐỔI TỔ CHỨC CÓ KẾ HOẠCH

Trong cuốn sách này chúng ta đã nhấn mạnh tới quan điểm về hệ thống tin như một thực thể xã hội kỹ thuật, một sự thu xếp cho cả các phần tử xã hội và kỹ thuật. Việc đưa một hệ thống tin mới vào bao gồm nhiều vấn đề hơn là chỉ có phần cứng và phần mềm. Nó cũng bao gồm cả những thay đổi về công việc, kỹ năng, quản lý và tổ chức. Theo triết lý về xã hội kỹ thuật, người ta không thể thiết lập công nghệ mới mà không xem xét tới con người, những người phải làm việc với nó. Các hệ thống tin mới thường có nghĩa là cách thức mới để tiến hành nghiệp vụ và làm việc cùng nhau. Bản chất của nhiệm vụ, tốc độ hoàn thành nhiệm vụ, bản chất của giám sát (tính thường xuyên và độ mạnh mẽ của nó), và người nào có thông tin nào về ai sẽ giữ vai trò quyết định trong tiến trình xây dựng hệ thống tin. Điều này đặc biệt đúng trong các hệ thống hiện đại, vốn ảnh hưởng sâu sắc tới nhiều phần của tổ chức. Khi chúng ta thiết kế một hệ thống tin mới, thì chúng ta đang thiết kế lại tổ chức.

Cần phải lập luận rằng việc quyết định hệ thống mới nào cần xây dựng nên là cấu phần chủ yếu của tiến trình lập kế hoạch tổ chức. Tổ chức cần phát triển một kế hoạch hệ thống tin hỗ trợ cho kế hoạch nghiệp vụ tổng thể của mình. Một khi đã có bản kế hoạch nghiệp vụ chiến lược tổng thể, thì **bản kế hoạch hệ thống tin** có thể được phát triển bên trong hoàn cảnh đó. Một kế hoạch như vậy sẽ phục vụ làm bản lộ trình chỉ ra hướng phát triển hệ thống, cơ sở logic, hiện trạng, chiến lược quản lý, kế hoạch thực hiện, và ngân sách (xem Bảng 3.1).

Bản kế hoạch này bao gồm một phát biểu về mục tiêu của tổ chức và xác định công nghệ thông tin hỗ trợ thế nào cho việc đạt tới những mục tiêu này. Nó chỉ ra cách thức các mục tiêu tổng quát sẽ được đạt tới bằng việc xác định các dự án hệ thống, và nó bố trí các ngày tháng đích riêng và các cột mốc mà có thể được dùng về sau

để đo và đánh giá tiến trình kế hoạch dưới dạng bao nhiêu mục tiêu thực tế đã đạt được trong khuôn khổ thời gian được xác định theo kế hoạch.

Bảng 3.1 KẾ HOẠCH HỆ THỐNG TIN

1. **Mục đích của bản kế hoạch**
 - Tổng quan về nội dung kế hoạch
 - Thay đổi trong hiện trạng của tổ chức
 - Kế hoạch chiến lược của tổ chức
 - Tổ chức nghiệp vụ hiện thời
 - Chiến lược quản lí
2. **Kế hoạch nghiệp vụ chiến lược**
 - Hiện trạng
 - Tổ chức nghiệp vụ hiện trạng
 - Thay đổi môi trường
 - Các mục tiêu chính của kế hoạch nghiệp vụ
3. **Hệ thống hiện tại**
 - Hệ thống chính hỗ trợ cho chức năng nghiệp vụ
 - Khả năng chính hiện tại
 - Phần cứng
 - Phần mềm
 - Cơ sở dữ liệu
 - Viễn thông
 - Những khó khăn khi đáp ứng yêu cầu kinh doanh
 - Nhu cầu tương lai dự kiến
4. **Những phát triển mới**
 - Dự án hệ thống mới
 - Mô tả dự án
 - Cơ sở logic nghiệp vụ
 - Những khả năng mới cần tới
 - Phần cứng
 - Phần mềm
 - Cơ sở dữ liệu
 - Viễn thông

5. Chiến lược quản lí

Kế hoạch mua sắm
Cột mốc và định thời gian
Bố trí lại tổ chức
Tổ chức lại nội bộ
Điều khiển quản lí
Sàng kiến huấn luyện chính
Chiến lược nhân sự

6. Kế hoạch thực hiện

Kế hoạch thực hiện chi tiết
Những khó khăn dự kiến trong thực hiện
Báo cáo tiến độ

7. Yêu cầu ngân sách

Yêu cầu
Tiết kiệm tiềm năng
Tài chính
Chu trình mua

Một phần quan trọng của kế hoạch hệ thống tin là chiến lược quản lí để chuyển từ tình trạng hiện thời tới tương lai. Nói chung, điều này sẽ chỉ ra các quyết định mấu chốt do người quản lí đưa ra liên quan tới việc mua phần cứng, viễn thông, tập trung hoá/phi tập trung hoá quyền lực, dữ liệu và phần cứng. Những thay đổi tổ chức cũng thường được mô tả, kể cả việc quản lí và huấn luyện nhân viên; các nỗ lực tuyển mộ; và thay đổi trong quyền lực, cấu trúc hay thực hành quản lí.

Thiết lập các yêu cầu thông tin của tổ chức

Để phát triển kế hoạch hệ thống tin có hiệu quả, tổ chức phải có một hiểu biết rõ ràng về cả các yêu cầu thông tin dài hạn và ngắn hạn của mình. Hai phương pháp luận chính cho việc thiết lập các yêu cầu hệ thống tin chủ chốt của tổ chức xem như một toàn thể là phân tích tổ chức và các nhân tố thành công chủ yếu.

PHÂN TÍCH TỔ CHỨC (LẬP KẾ HOẠCH HỆ THỐNG NGHIỆP VỤ)

Phân tích tổ chức (cũng còn được gọi là lập kế hoạch hệ thống nghiệp vụ) biện minh rằng các yêu cầu thông tin của tổ chức chỉ có thể được hiểu bằng việc nhìn vào toàn bộ tổ chức dưới dạng các đơn vị tổ chức, chức năng, tiến trình và các phần tử dữ liệu. Phân tích tổ chức có thể giúp xác định ra các thực thể chủ chốt và các thuộc tính của dữ liệu của tổ chức. Phương pháp này bắt đầu với ý niệm rằng các yêu cầu thông tin của tổ chức hay phòng ban có thể được xác định chỉ bằng việc hiểu thấu đáo về toàn bộ tổ chức. Phương pháp này đã được IBM phát triển từ những năm 1960 một cách tường minh để thiết lập mối quan hệ giữa các dự án phát triển hệ thống lớn.

Phương pháp trung tâm được dùng trong cách tiếp cận phân tích tổ chức là lấy mẫu lớn về các nhà quản lí và hỏi họ cách họ dùng thông tin, nơi họ lấy thông tin, môi trường của họ như thế nào, mục đích của họ là gì, họ ra quyết định như thế nào, và họ cần dữ liệu gì.

Kết quả của cuộc điều tra lớn này về các nhà quản lí được tập hợp lại trong các đơn vị con, các chức năng, tiến trình, và ma trận dữ liệu. Các phần tử dữ liệu được tổ chức thành các *nhóm ứng dụng logic* - nhóm các phần tử dữ liệu hỗ trợ cho tập các tiến trình có liên quan của tổ chức. Một điểm mạnh của phân tích tổ chức là ở chỗ nó cho cái nhìn thấu đáo về tổ chức và việc dùng hệ thống/dữ liệu và lỗi hỏng.

Phân tích tổ chức đặc biệt thích hợp cho những tình huống mới bắt đầu hay có những thay đổi lớn lao. Chẳng hạn, đây là một phương pháp được Cơ quan Quản trị Bảo hiểm xã hội Mỹ sử dụng để đem sự thay đổi chiến lược dài hạn vào trong các hoạt động xử lí thông tin. Tổ chức này chưa bao giờ thực hiện một phân tích thấu đáo như vậy trước đây về các yêu cầu thông tin của mình. Thay vì thế, nó đã dựa trên phương pháp từ dưới lên để đáp ứng cho bất kì cái gì người dùng yêu cầu, cũng như bởi cách tiếp cận sản phẩm

phụ, chỗ mà sự nhấn mạnh nhiều nhất được đặt vào việc thực hiện đơn giản xử lý giao tác sơ cấp. Phân tích tổ chức đã được dùng để phát triển một cái nhìn thấu đáo về cách Cơ quan Quản trị Bảo hiểm xã hội sử dụng thông tin.

Nhược điểm của phân tích tổ chức là ở chỗ nó tạo ra khối lượng dữ liệu khổng lồ vốn rất tốn kém để thu thập và khó phân tích. Hơn nữa, các câu hỏi thường xuyên tập trung không phải vào các mục tiêu chủ chốt của quản lý và nơi thông tin được cần tới, mà thay vì thế lại tập trung vào thông tin *hiện có* đang được dùng. Kết quả là có khuynh hướng tự động hoá bất kỳ cái gì đang có. Theo cách này, các hệ thống thủ công có thể được tự động hoá trong khi, trong nhiều trường hợp, toàn bộ cách tiếp cận mới tới cách tiến hành nghiệp vụ lại là cần thiết, và nhu cầu này có thể không được đề cập tới.

PHÂN TÍCH CHIẾN LƯỢC: CÁC NHÂN TỐ THÀNH CÔNG CHỦ CHỐT

Plan tích chiến lược hay cách tiếp cận nhân tố thành công chủ chốt biện minh rằng các yêu cầu thông tin của tổ chức được xác định bởi một số nhỏ các **nhân tố thành công chủ chốt (CSF)** của người quản lý. CSF là các mục tiêu vận hành. Nếu những mục tiêu này có thể được đạt tới, thì sự thành công của tổ chức được đảm bảo.

CSF được hình thành bởi công nghiệp, công ti, người quản lý và môi trường rộng hơn. Sự tập trung rộng hơn này, trong so sánh với phương pháp trên đây, giải thích cho mô tả về kỹ thuật này là "chiến lược". Một tiền đề quan trọng của cách tiếp cận phân tích chiến lược là ở chỗ có một số nhỏ các mục tiêu mà người quản lý có thể dễ dàng xác định và hệ thống tin có thể tập trung vào đó.

Phương pháp chính được dùng trong phân tích CSF là phỏng vấn cá nhân - ba hay bốn người - những người quản lý cao nhất để xác định mục đích của họ và CSF kết quả. Những CSF cá nhân này được tổ hợp lại để phát triển một bức tranh về CSF của tổ chức. Thế

rời hệ thống được xây dựng để chuyển giao thông tin về những CSF này.

Một điểm mạnh của phương pháp CSF là ở chỗ nó tạo ra một tập dữ liệu nhỏ để phân tích hơn là cách phân tích tổ chức. Chỉ những người quản lý cấp cao nhất mới được phỏng vấn, và các câu hỏi hội tụ vào một số nhỏ các CSF thay vì việc hỏi rộng rãi thông tin nào được dùng hay cần tới. Phương pháp này có thể được may đo theo cấu trúc của từng ngành công nghiệp, với những chiến lược cạnh tranh khác nhau tạo ra các hệ thống tin khác nhau. Phương pháp CSF cũng phụ thuộc vào vị trí ngành công nghiệp và thậm chí vào vị trí địa lý. Do đó, phương pháp này tạo ra các hệ thống nhiều tính may đo hơn đối với tổ chức. Phương pháp CSF đặc biệt thích hợp cho các hệ thống phục vụ cho cấp quản lý cao nhất và cho việc phát triển các hệ thống hỗ trợ quyết định và hệ thống hỗ trợ điều hành.

Nhược điểm chính của phương pháp này là ở chỗ tiến trình tổ hợp và phân tích dữ liệu là dạng nghệ thuật. Không có cách thức chặt chẽ đặc biệt nào để theo dõi các CSF cá nhân có thể được tổ hợp lại thành một hình mẫu tổ chức rõ ràng. Thứ hai là, thường có sự lẫn lộn giữa những người được phỏng vấn (và người phỏng vấn) giữa CSF *cá nhân* và *tổ chức*. Chúng không nhất thiết là như nhau. Điều có thể là chủ chốt đối với người quản lý thì lại có thể không quan trọng đối với tổ chức. Hơn nữa, phương pháp này rõ ràng nghiêng về phía nhà quản lý cấp cao nhất bởi vì họ là người (nói chung duy nhất) được phỏng vấn.

Thiết kế lại các tiến trình nghiệp vụ

Hệ thông tin mới có thể là công cụ mạnh mẽ cho việc thay đổi tổ chức. Chúng không chỉ giúp làm hợp lý hoá các thủ tục và luồng công việc của tổ chức, mà chúng còn thực tế được dùng để định hình lại cách tổ chức tiến hành nghiệp vụ của mình hay thậm chí cả bản chất của bản thân nghiệp vụ.

Tiến trình nghiệp vụ là tập hợp các nhiệm vụ có quan hệ logic được thực hiện để đạt tới ích lợi nghiệp vụ đã xác định. Một số thí dụ về các tiến trình nghiệp vụ là phát triển một sản phẩm mới, đặt hàng từ nhà cung cấp, hay xử lý và thanh toán yêu cầu bảo hiểm.

Hệ thông tin mới có thể thiết kế lại triệt để các tiến trình nghiệp vụ để tăng tốc độ, dịch vụ và chất lượng. Nếu tiến trình nghiệp vụ đầu tiên được thiết kế lại trước khi năng lực tính toán được áp dụng, thì các tổ chức có thể thu được phần lợi lớn từ đầu tư của họ vào công nghệ thông tin. Việc thiết kế lại tiến trình nghiệp vụ phục vụ cho việc tổ chức lại luồng công việc, tổ hợp các bước để cắt bỏ lãng phí và xoá bỏ các nhiệm vụ lặp lại, nặng gấy tờ (đôi khi thiết kế mới còn xoá bỏ cả công việc nữa). Thuật ngữ khác cho việc thiết kế lại tiến trình nghiệp vụ là *tái kỹ nghệ nghiệp vụ*.

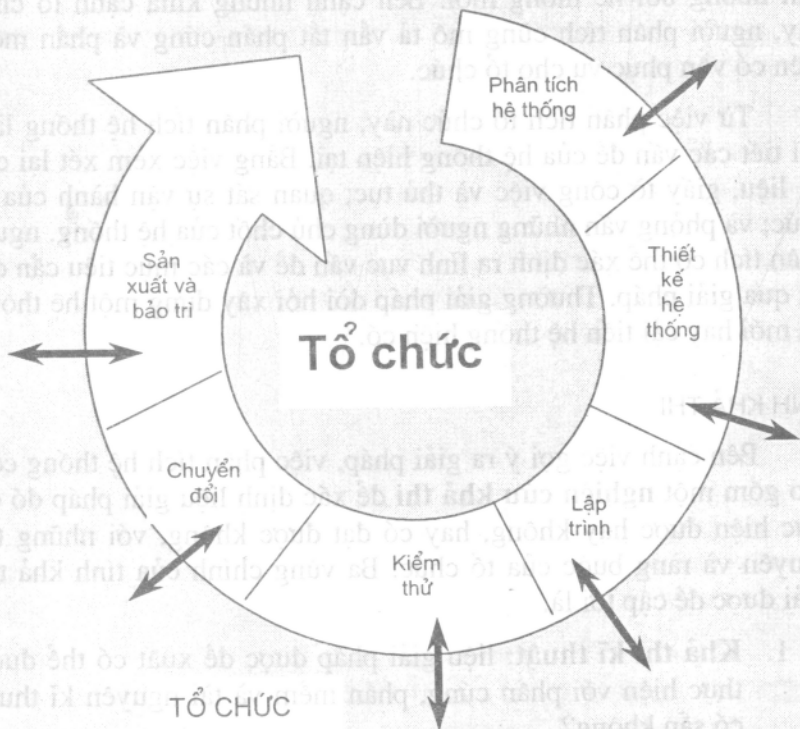
3.2 TỔNG QUAN VỀ PHÁT TRIỂN HỆ THỐNG

Dù nguồn gốc của chúng ở đâu thì các hệ thông tin mới cũng đang phát triển ra từ tiến trình giải quyết vấn đề của tổ chức. Hệ thông tin mới được xây dựng xem như một giải pháp cho kiểu vấn đề nào đó hay tập các vấn đề nào đó mà tổ chức cảm nhận mình đang phải đối diện. Vấn đề có thể là ở chỗ các nhà quản lý và nhân viên hiểu ra rằng tổ chức đang không thực hiện tốt như được trông đợi, hay nó có thể tới từ việc hiểu ra rằng tổ chức nên tận dụng ưu thế của những cơ hội mới để thực hiện thành công hơn.

Hệ thông tin là sản phẩm của một loạt những hoạt động được gọi là phát triển hệ thống. **Phát triển hệ thống** nói tới mọi hoạt động đi vào việc tạo ra giải pháp hệ thống tin cho một vấn đề hay cơ hội của tổ chức. Phát triển hệ thống là một loại có cấu trúc của việc giải quyết vấn đề bằng các hoạt động phân biệt. Các hoạt động

này bao gồm phân tích hệ thống, thiết kế hệ thống, lập trình, kiểm thử, chuyển đổi, sản xuất và bảo trì.

Hình 3.1 minh họa cho tiến trình phát triển hệ thống. Các hoạt động phát triển hệ thống được vẽ ra ở đây thường xảy ra theo trật tự tuần tự. Nhưng một số hoạt động có thể cần được lặp lại hay một số có thể xảy ra đồng thời, tùy theo cách tiếp cận tới việc xây dựng hệ thống được sử dụng. Cũng nên lưu ý rằng từng hoạt động lại bao gồm tương tác với tổ chức. Các thành viên của tổ chức tham gia vào những hoạt động này, và tiến trình phát triển hệ thống tạo ra những thay đổi tổ chức.



Hình 3.1. Tiến trình phát triển hệ thống

Phân tích hệ thống

Phân tích hệ thống là việc phân tích vấn đề mà tổ chức sẽ cố gắng giải quyết với hệ thống tin. Nó bao gồm việc định nghĩa vấn đề, định danh nguyên nhân của nó, xác định giải pháp và định danh các yêu cầu thông tin cần phải được giải pháp hệ thống thoả mãn.

Điều mấu chốt để xây dựng bất kì hệ thống tin lớn nào là hiểu biết thấu đáo về tổ chức và hệ thống hiện tại. Vậy, người phân tích hệ thống tạo ra một bản lộ trình về tổ chức và hệ thống hiện tại, xác định người chủ chính và người dùng dữ liệu trong tổ chức. Những người bảo trợ này có mối quan tâm trực tiếp về thông tin bị ảnh hưởng bởi hệ thống mới. Bên cạnh những khía cạnh tổ chức này, người phân tích cũng mô tả vấn đề phân cứng và phần mềm hiện có vốn phục vụ cho tổ chức.

Từ việc phân tích tổ chức này, người phân tích hệ thống làm chi tiết các vấn đề của hệ thống hiện tại. Bằng việc xem xét lại các tài liệu, giấy tờ công việc và thủ tục; quan sát sự vận hành của tổ chức; và phỏng vấn những người dùng chủ chốt của hệ thống, người phân tích có thể xác định ra lĩnh vực vấn đề và các mục tiêu cần đạt tới qua giải pháp. Thường giải pháp đòi hỏi xây dựng một hệ thống tin mới hay cải tiến hệ thống hiện có.

TÍNH KHẢ THI

Bên cạnh việc gợi ý ra giải pháp, việc phân tích hệ thống còn bao gồm một **nguyên cứu khả thi** để xác định liệu giải pháp đó có thực hiện được hay không, hay có đạt được không, với những tài nguyên và ràng buộc của tổ chức. Ba vùng chính của tính khả thi phải được đề cập tới là:

1. **Khả thi kĩ thuật:** liệu giải pháp được đề xuất có thể được thực hiện với phần cứng, phần mềm và tài nguyên kĩ thuật có sẵn không?

2. **Khả thi kinh tế:** liệu ích lợi của giải pháp được đề nghị có cân xứng với chi phí không?
3. **Khả thi vận hành:** liệu giải pháp được đề nghị có là điều mong muốn bên trong khuôn khổ quản lí và tổ chức hiện tại không.

Thông thường tiến trình phân tích hệ thống sẽ xác định nhiều giải pháp mà có thể được tổ chức theo đuổi. Thế rồi tiến trình này sẽ định giá tính khả thi của từng giải pháp. Ba phương án giải pháp tồn tại cho mọi vấn đề hệ thống:

- Không làm gì cả, để hệ thống hiện tại không đổi
- Sửa đổi hay nâng cao hệ thống hiện có
- Phát triển hệ thống mới

Để thiết lập tính khả thi kinh tế, người phân tích sẽ phải cân nhắc chi phí và ích lợi của từng giải pháp được đề nghị. Phân tích chi phí/ích lợi bao gồm việc so sánh tất cả các chi phí của một dự án và mọi ích lợi của nó. Những chi phí và ích lợi này được làm chi tiết trong Bảng 3.2. Một giải pháp tối ưu sẽ làm cực đại ích lợi với chi phí đã cho hay chi phí tối thiểu cho một khối lượng ích lợi đặc biệt.

Nhưng vấn đề lại hoá ra là không phải dễ xác định được tất cả mọi ích lợi và chi phí một cách chính xác đối với nhiều dự án hệ thống tin. Nhiều ích lợi của các hệ thống phức tạp, như việc làm quyết định được nâng cao hay sự thoả mãn hơn của khách hàng, là không thấy được. **Ích lợi không thấy được** như nâng cao việc làm quyết định hay phục vụ khách hàng tốt hơn không thể nào định lượng được. (Tương phản lại, **ích lợi thấy được**, như giảm nhân viên thư kí hay chi phí máy tính thấp, là những ích lợi có thể định lượng được.) Khó khăn của việc đo đạc ích lợi không thấy được có thể tạo ra "khuyh hướng ứng dụng", bởi vì hệ xử lí giao tác TPS vốn tiết kiệm lao động thì tạo ra nhiều ích lợi đo được, thấy được hơn là hệ thống tin quản lí MIS, hệ trợ giúp quyết định DSS hay hệ công tác tri thức. Vậy mà những hệ thống tạo ra chủ yếu các ích lợi

không thấy được lại có thể rất có giá trị và có lẽ thậm chí còn là bản chất cho tổ chức.

Bảng 3.2 CHI PHÍ VÀ ÍCH LỢI CỦA HỆ THỐNG

Chi phí hệ thống	Ích lợi hệ thống
Chi phí thực hiện	Thấy được
Thời gian máy tính	Tiết kiệm chi phí
Nhân viên kỹ thuật	Năng suất tăng
Thời gian người dùng	Chi phí vận hành thấp
Chi phí đóng gói	Lực lượng lao động giảm
	Chi phí máy tính thấp
Chi phí vận hành	Chi phí nhà cung cấp ngoài thấp
Thời gian máy tính	
Trang bị phụ	Tránh được chi phí
Bảo trì phần mềm	Chi phí thư ký và chuyên môn
Nhân viên vận hành	Giảm tỉ lệ tăng chi phí
Thời gian người dùng	Chi phí tiện nghi giảm
Nhà cung cấp	
Chi phí huấn luyện tiếp tục	Không thấy được
Chi phí tiện nghi	Việc sử dụng tài sản cải thiện
	Kiểm soát tài nguyên cải tiến
	Hoạch định tổ chức cải tiến
	Linh hoạt tổ chức cải tiến
	Thông tin đúng hạn hơn
	Nhiều thông tin hơn
	Việc học của tổ chức tăng
	Yêu cầu pháp lý được đạt tới
	Thiện chí nhân viên tăng lên
	Thoả mãn công việc tăng lên
	Ra quyết định tốt hơn
	Vận hành tốt hơn
	Thoả mãn khách hàng cao
	Hình ảnh tổ chức tốt hơn

Một báo cáo đề xuất hệ thống được viết ra sẽ mô tả cho chi phí và ích lợi, ưu và nhược điểm của từng phương án. Thế rồi tùy cấp quản lý xác định việc trộn lẫn chi phí, ích lợi, tính năng kỹ thuật

và tác động tổ chức nào biểu thị cho phương án đáng mong muốn nhất.

THIẾT LẬP YÊU CẦU THÔNG TIN

Có lẽ nhiệm vụ khó khăn nhất của người phân tích hệ thống là định nghĩa ra các yêu cầu thông tin cụ thể mà phải được giải pháp đã chọn lựa đáp ứng cho. Đây là vùng mà nhiều nỗ lực hệ thống lớn đi sai và là vùng đặt ra những khó khăn lớn nhất cho người phân tích. Tại mức cơ sở nhất, các yêu cầu thông tin về hệ thống mới bao gồm việc xác định ai cần thông tin gì, ở đâu, khi nào và thế nào. Việc phân tích yêu cầu xác định cẩn thận các mục tiêu của hệ thống mới hay hệ thống được sửa đổi và phát triển một mô tả chi tiết về chức năng mà hệ thống mới phải thực hiện. Các yêu cầu phải xem xét những ràng buộc về kinh tế, kỹ thuật và thời gian, cũng như các mục tiêu, thủ tục và tiến trình quyết định của tổ chức. Việc phân tích yêu cầu sai dẫn tới nguyên nhân hệ thống thất bại và chi phí phát triển hệ thống cao. Một hệ thống được thiết kế quanh một tập các yêu cầu sai sẽ hoặc phải bị bỏ đi bởi vì hiệu năng kém hay sẽ cần được sửa đổi lại rất nhiều. Do đó, tầm quan trọng của phân tích yêu cầu không thể nào bị đánh giá thấp được.

Trong nhiều trường hợp, việc xây dựng một hệ thống mới tạo ra cơ hội để định nghĩa lại cách thức tổ chức tiến hành nghiệp vụ thường ngày của nó. Một số vấn đề thậm chí không yêu cầu tới giải pháp hệ thống tin, nhưng thay vì thế lại cần sự điều chỉnh trong quản lý, việc huấn luyện thêm, hay làm hoàn chỉnh thêm các thủ tục tổ chức hiện có. Nếu vấn đề có liên quan tới thông tin, thì người phân tích hệ thống có thể vẫn được yêu cầu chẩn đoán vấn đề và đi tới một giải pháp đúng.

Thiết kế hệ thống

Trong khi phân tích hệ thống mô tả điều hệ thống phải làm để đáp ứng các yêu cầu thông tin, thì **thiết kế hệ thống** chỉ ra cách

thức hệ thống sẽ hoàn thành mục đích này. Thiết kế một hệ thống tin là bản kế hoạch tổng thể hay mô hình cho hệ thống đó. Giống như bản kế hoạch toàn diện để xây dựng một toà nhà hay ngôi nhà, nó bao gồm tất cả những đặc tả đem lại cho hệ thống hình hài và cấu trúc. Thiết kế hệ thống tin là một nhiệm vụ sáng tạo và đòi hỏi cố gắng, đòi hỏi nhiều tưởng tượng, nhạy cảm với chi tiết, và kỹ năng chuyên gia.

Người thiết kế hệ thống làm chi tiết hoá các đặc tả vốn chuyển giao các chức năng được xác định trong việc phân tích hệ thống. Những đặc tả này nên đề cập tới tất cả các cấu phần quản lý, tổ chức và công nghệ của giải pháp hệ thống. Bảng 3.3 liệt kê các kiểu đặc tả sẽ được tạo ra trong thiết kế hệ thống.

Bảng 3.3 ĐẶC TẢ THIẾT KẾ

Cái ra	Kiểm soát
Môi giới	Kiểm soát cái vào (kí tự, giới hạn, tính hợp lý)
Nội dung	Kiểm soát xử lý (tính nhất quán, đếm bản ghi)
Thời gian	Kiểm soát cái ra (tổng số, mẫu cái ra)
	Kiểm soát thủ tục (mật hiệu, dạng đặc biệt)
Cái vào	An toàn
Nguồn gốc	Kiểm soát truy nhập
Luồng	Kế hoạch tai biến
Vào dữ liệu	Vết kiểm toán
Giao diện người dùng	Làm tài liệu
Đơn giản	Tài liệu vận hành
Hiệu quả	Tài liệu hệ thống
Logic	Tài liệu người dùng
Phản hồi	
Lỗi	
Thiết kế cơ sở dữ liệu	Chuyển đổi
Quan hệ dữ liệu logic	Truyền tệp

Bảng 3.3 (tiếp theo)

Yêu cầu khối lượng và tốc độ	Khởi đầu thủ tục mới Chọn phương pháp kiểm thử Chuyển qua hệ thống mới
Xử lí	Huấn luyện
Tính toán	Chọn kĩ thuật huấn luyện
Modun chương trình	Phát triển các modun huấn luyện
Báo cáo được yêu cầu	Xác định tiện nghi huấn luyện
Thời gian đưa ra	
Thủ tục thủ công	Thay đổi tổ chức
Hoạt động gì	Thiết kế lại nhiệm vụ
Ái thực hiện chúng	Thiết kế công việc
Khi nào	Thiết kế văn phòng và cấu trúc tổ chức
Thế nào	Mối quan hệ báo cáo
ở đâu	

THIẾT KẾ LOGIC VÀ VẬT LÍ

Thiết kế hệ thống tin có thể được chia ra thành các đặc tả thiết kế logic và vật lí. **Thiết kế logic** bố trí các cấu phần của hệ thống và mối quan hệ của chúng với nhau khi chúng xuất hiện cho người dùng. Nó biểu lộ điều thiết kế hệ thống sẽ thực hiện đối lập với cách nó thực tế được thực hiện về mặt vật lí. Nó mô tả cái vào và cái ra, các chức năng xử lí cần được thực hiện, các thủ tục nghiệp vụ, các mô hình dữ liệu và các kiểm soát. (Kiểm soát xác định các chuẩn về hiệu năng chấp nhận được và các phương pháp để đo lường hiệu năng thực tế trong mối quan hệ với các chuẩn này.

Thiết kế vật lí là tiến trình dịch mô hình logic trừu tượng thành thiết kế kĩ thuật đặc thù cho hệ thống mới. Nó tạo ra các đặc tả thực tế về phần cứng, phần mềm, cơ sở dữ liệu thực tế, môi giới vào/ra, thủ tục thủ công, và các kiểm soát đặc biệt. Thiết kế vật lí đưa ra các đặc tả làm biến đổi bản kế hoạch thiết kế logic thành hệ thống vận hành gồm người và máy.

Giống như xây nhà hay toà nhà, hệ thống tin có thể có nhiều thiết kế. Chúng có thể là tập trung hoá hay phi tập trung hoá, trực tuyến hay theo lô, thủ công một phần hay tự động hoá toàn bộ. Mỗi thiết kế biểu thị cho một sự pha trộn duy nhất của tất cả các nhân tố kỹ thuật và tổ chức vốn hình thành nên hệ thống tin. Điều làm cho thiết kế này tốt hơn thiết kế kia là sự dễ dàng và tính hiệu quả mà qua đó nó đáp ứng cho yêu cầu của người dùng bên trong một tập hợp xác định các ràng buộc kỹ thuật, tổ chức, tài chính và thời gian. Trước khi thiết kế của hệ thống tin được hoàn tất, người phân tích sẽ đánh giá nhiều phương án thiết kế khác nhau và chọn ra phương án biểu thị cho giải pháp tốt nhất cho tổ chức.

VAI TRÒ CỦA NGƯỜI DÙNG CUỐI

Các yêu cầu thông tin người dùng hướng dẫn cho toàn thể nỗ lực xây dựng hệ thống, và do vậy thiết kế hệ thống tin yêu cầu một mức độ tham dự và kiểm soát rất cao của người dùng. Người dùng phải có đủ kiểm soát với tiến trình thiết kế để đảm bảo rằng hệ thống phản ánh các ưu tiên nghiệp vụ của họ và nhu cầu thông tin, không nghiêng về phía nhân viên kỹ thuật.

Làm việc trên thiết kế làm tăng hiểu biết và sự chấp nhận của người dùng đối với hệ thống, làm giảm các vấn đề gây ra bởi việc chuyển quyền lực, xung khắc giữa các nhóm và sự không quen thuộc với các chức năng và thủ tục hệ thống mới.

Bản chất và mức độ của sự tham gia của người dùng vào thiết kế thay đổi từ hệ thống nọ sang hệ thống kia. Ít có nhu cầu về sự tham gia của người dùng trong các hệ thống với những yêu cầu đơn giản và trực tiếp hơn là trong những hệ với yêu cầu phức tạp hay còn xác định mung lung. Các hệ thống kém cấu trúc, như ứng dụng hỗ trợ quyết định, cần sự tham gia của người dùng nhiều hơn để định nghĩa các yêu cầu. Các mức độ khác nhau của sự tham gia của người dùng trong thiết kế được phản ánh trong nhiều cách tiếp cận phát triển hệ thống sẽ được thảo luận sau.

Hoàn chỉnh tiến trình phát triển hệ thống

Các bước còn lại trong tiến trình phát triển hệ thống dịch các đặc tả giải pháp đã được thiết lập trong phân tích và thiết kế hệ thống thành hệ thống thông tin vận hành hoàn toàn. Các bước kết thúc này bao gồm việc lập trình, kiểm thử, chuyển đổi và sản xuất cùng bảo trì.

LẬP TRÌNH

Tiến trình dịch các đặc tả thiết kế thành phần mềm cho máy tính tạo nên một phần nhỏ hơn của chu trình phát triển hệ thống so với thiết kế và có lẽ cả các hoạt động kiểm thử. Nhưng chính ở đây, trong khi đưa ra các lệnh thực tế cho máy, thì chính trái tim của hệ thống cũng hình thành. Trong **lập trình**, các đặc tả hệ thống đã được chuẩn bị trong giai đoạn thiết kế nay được dịch thành mã chương trình. Trên cơ sở các tài liệu thiết kế chi tiết về tệp, giao tác và bố trí báo cáo cùng các chi tiết thiết kế khác, các đặc tả cho từng chương trình trong hệ thống sẽ được chuẩn bị.

KIỂM THỬ

Kiểm thử thấu đáo và toàn diện phải được tiến hành để chắc chắn xem liệu hệ thống có tạo ra kết quả đúng hay không. Kiểm thử trả lời câu hỏi, "Hệ thống có tạo ra kết quả mong muốn theo những điều kiện đã biết hay không?"

Khối lượng thời gian cần thiết để trả lời cho câu hỏi này về truyền thống thường bị coi nhẹ trong việc lập kế hoạch dự án. Nhiều nhất thì 50 phần trăm của toàn bộ ngân sách phát triển phần mềm có thể dành cho chi tiêu vào việc kiểm thử. Kiểm thử cũng tốn thời gian. Dữ liệu kiểm thử phải được chuẩn bị cẩn thận, các kết quả cần được xét duyệt, và sửa đổi được tiến hành trong hệ thống. Trong một số thể nghiệm, các bộ phận của hệ thống có thể cần được thiết kế lại. Vậy mà rủi ro bao trùm bước này lại rất lớn.

Kiểm thử hệ thông tin có thể được chia thành ba kiểu hoạt động:

Kiểm thử đơn vị, hay kiểm thử chương trình, bao gồm việc kiểm thử cho từng chương trình trong hệ thống một cách tách biệt. Trong khi người ta tin tưởng rộng rãi rằng việc kiểm thử như vậy là để đảm bảo cho chương trình hết lỗi, thì mục tiêu này thực tế là không thể thực hiện được. Thay vì thế kiểm thử được coi là phương tiện để định vị lỗi trong chương trình, hội tụ vào việc tìm ra tất cả mọi cách để làm cho chương trình bị hỏng. Một khi đã được chỉ rõ ra, thì vấn đề có thể được sửa chữa.

Kiểm thử hệ thống kiểm thử cho việc vận hành của hệ thống tin xem như một tổng thể. Nó cố gắng xác định ra liệu các modul rời rạc có làm việc cùng nhau như đã vạch kế hoạch không và liệu có tồn tại sự khác biệt giữa cách hệ thống thực tế làm việc và cách nó được quan niệm không. Trong số những lĩnh vực được xem xét có xem xét về thời gian hiệu năng, khả năng lưu giữ tệp và giải quyết tải cao điểm, khả năng khôi phục và chạy lại, và các thủ tục thủ công.

Kiểm thử chấp nhận nảy sinh trong việc xác nhận cuối cùng của người dùng và cấp quản lí rằng hệ thống sẵn sàng được dùng trong môi trường sản xuất - khi tất cả các bên đều thoả mãn rằng hệ thống mới đáp ứng cho các chuẩn của họ, thì hệ thống được chấp một cách chính thức cho việc thiết đặt.

Điều bản chất là mọi khía cạnh của kiểm thử phải được suy tính cẩn thận và chúng phải thấu đáo nhất có thể được. Để đảm bảo điều này, tổ phát triển làm việc với người dùng để tạo ra bản kế hoạch kiểm thử hệ thống. **Bản kế hoạch kiểm thử** bao gồm tất cả mọi sự chuẩn bị cho hàng loạt các phép kiểm thử đã được mô tả trước đây.

Người dùng đóng vai trò chủ chốt trong tiến trình kiểm thử. Họ hiểu đầy đủ phạm vi dữ liệu và điều kiện xử lí mà có thể xuất

hiện bên trong hệ thống của mình. Hơn nữa, người lập trình có khuynh hướng chỉ nhận biết về điều kiện được xử lý trong chương trình của mình; dữ liệu kiểm thử mà họ phát minh ra thường quá giới hạn. Do đó, cái vào từ các thành viên tổ khác và người dùng sẽ giúp đảm bảo rằng miền điều kiện được chứa trong dữ liệu kiểm thử là đầy đủ. Người dùng có thể xác định ra các giao tác thường xuyên hay ít thông dụng, các điều kiện bất thường cần dự kiến trước, và phần lớn các kiểu lỗi mà có thể xuất hiện khi hệ thống được sử dụng. Cái vào của người dùng cũng là quyết định trong việc kiểm chứng các thủ tục thủ công đối với hệ thống.

CHUYỂN ĐỔI

Chuyển đổi là tiến trình thay đổi từ hệ thống cũ sang hệ thống mới. Nó trả lời cho câu hỏi, "Hệ thống mới có làm việc trong điều kiện thực hay không?" Bốn chiến lược chuyển đổi chính có thể được sử dụng: chiến lược song song, chiến lược chuyển trực tiếp, chiến lược nghiên cứu thử nghiệm và chiến lược tiếp cận theo pha.

Trong **chiến lược song song**, cả hệ thống cũ và hệ thống tiềm năng thay thế cho nó đều được cùng cho chạy trong một thời gian cho tới khi mọi người đều được đảm bảo rằng hệ thống mới vận hành đúng đắn. Đây là cách tiếp cận chuyển đổi an toàn nhất bởi vì, trong biến cố có lỗi hay gián đoạn xử lý, thì hệ thống cũ có thể vẫn được dùng như bản dự phòng. Tuy nhiên, cách tiếp cận này lại rất tốn kém, và nhân viên hay tài nguyên phụ có thể cần phải có để cho chạy hệ thống phụ.

Chiến lược **chuyển thẳng trực tiếp** thay thế toàn bộ hệ thống cũ bằng hệ thống mới vào một ngày đã chỉ định. Mới thoáng nhìn, chiến lược này dường như không tốn kém hơn chiến lược chuyển đổi song song. Tuy nhiên, đây là cách tiếp cận rất mạo hiểm mà tiềm năng có thể còn tốn kém hơn là các hoạt động song song. Nếu những vấn đề nghiêm trọng với hệ thống mới được tìm ra, thì

không có hệ thống khác để dùng tới. Trục trặc, ngắt quãng, và cái giá cho việc sửa chữa có thể cực kì lớn.

Chiến lược **nghiên cứu thử nghiệm** đưa hệ thống mới vào chỉ từng vùng có giới hạn của tổ chức, như một phòng ban hay đơn vị vận hành. Khi phiên bản thử nghiệm này hoàn chỉnh và làm việc trơn tru, thì nó được thiết đặt trong toàn bộ phần còn lại của tổ chức, hoặc đồng thời hoặc theo từng giai đoạn.

Chiến lược **theo pha** đưa hệ thống mới vào theo từng giai đoạn, hoặc theo chức năng hoặc theo đơn vị tổ chức. Nếu chẳng hạn hệ thống được đưa vào theo chức năng, thì một hệ thống trả lương mới có thể bắt đầu làm việc cho người làm việc theo giờ, người được trả theo tuần, rồi tiếp sau đó sáu tháng sau mới làm việc thêm cho các nhân viên ăn lương, người được trả lương theo tháng. Nếu hệ thống được đưa vào theo các đơn vị tổ chức, thì tổng hành dinh công ti có thể được chuyển đổi trước, tiếp đó là các đơn vị vận hành xa trung tâm bốn tháng sau đó.

Kế hoạch chuyển đổi chính thức cung cấp một lịch biểu cho tất cả các hoạt động cần thiết cho việc thiết đặt hệ thống mới. Hoạt động tốn thời gian nhất thường là chuyển đổi dữ liệu. Dữ liệu từ hệ thống cũ phải được truyền sang cho hệ thống mới, hoặc là thủ công hoặc qua chương trình phần mềm chuyển đổi đặc biệt. Rồi dữ liệu đã chuyển đổi đó phải được kiểm chứng cẩn thận lại về độ chính xác và tính đầy đủ.

Việc chuyển từ hệ thống cũ sang hệ thống mới đòi hỏi rằng người dùng cuối cần được huấn luyện để dùng hệ thống mới. **Tài liệu** chi tiết chỉ ra cách hệ thống làm việc theo cả quan điểm kĩ thuật lẫn người dùng cuối cần được hoàn thành trong thời gian chuyển đổi để dùng trong huấn luyện và vận hành hàng ngày. Thiếu huấn luyện và tài liệu đúng đắn cũng đóng góp cho sự thất bại của hệ thống, cho nên phần này của tiến trình phát triển hệ thống là rất quan trọng.

SẢN XUẤT VÀ BẢO TRÌ

Sau khi hệ thống mới được thiết đặt và việc chuyển đổi được hoàn tất, thì hệ thống được coi là vào **sản xuất**. Trong giai đoạn này, hệ thống sẽ được xem xét bởi cả người dùng và các chuyên viên kỹ thuật để xác định nó đáp ứng tốt đến đâu cho các mục tiêu khởi thủy và để quyết định xem liệu có những cải tiến hay sửa đổi nào là hợp lệ. Những thay đổi về phần cứng, phần mềm, tài liệu hay thủ tục đối với hệ thống sản xuất để sửa lỗi, đáp ứng yêu cầu mới hay cải tiến hiệu quả xử lý được gọi là **bảo trì**.

Bảng 3.4 PHÁT TRIỂN HỆ THỐNG

Hoạt động lõi	Mô tả
Phân tích hệ thống	Xác định vấn đề Đặc tả giải pháp Thiết lập yêu cầu thông tin
Thiết kế hệ thống	Tạo ra đặc tả thiết kế logic Tạo ra đặc tả thiết kế vật lý Quản lý việc thực hiện kỹ thuật về hệ thống
Lập trình	Dịch các đặc tả thiết kế thành mã chương trình
Kiểm thử	Kiểm thử đơn vị Kiểm thử hệ thống Kiểm thử chấp nhận
Chuyển đổi	Lập kế hoạch chuyển đổi Chuẩn bị tài liệu Huấn luyện người dùng và nhân viên kỹ thuật
Sản xuất và bảo trì	Vận hành hệ thống Đánh giá hệ thống Sửa đổi hệ thống

Các khảo cứu về bảo trì đã xem xét khối lượng thời gian cần cho nhiều nhiệm vụ bảo trì khác nhau. Xấp xỉ quãng 20 phần trăm

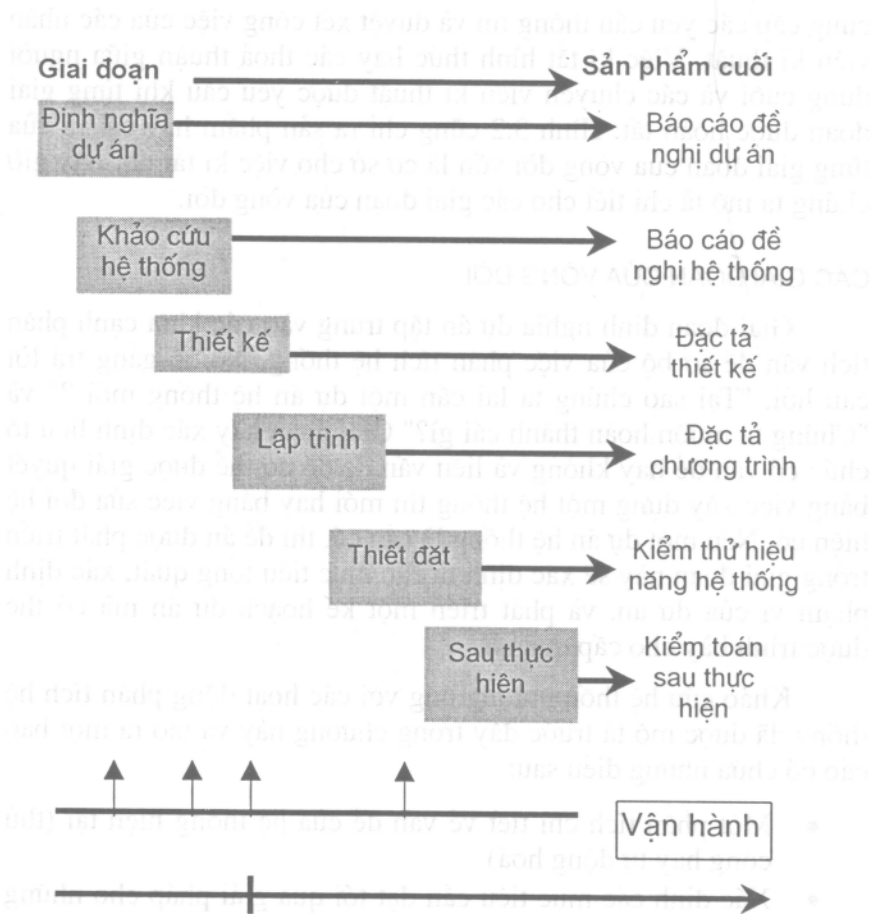
thời gian được dành cho gỡ lỗi hay sửa các vấn đề sản xuất khẩn cấp; 20 phần trăm khác liên quan tới những thay đổi trong dữ liệu, tệp, báo cáo, phần cứng hay phần mềm hệ thống. Nhưng 60 phần trăm của tất cả công việc bảo trì bao gồm việc nâng cao cho người dùng, cải tiến tài liệu, và mã hoá lại các cấu phần của hệ thống để có hiệu quả xử lý lớn hơn. Khối lượng công việc trong loại vấn đề bảo trì thứ ba có thể được giảm đi đáng kể qua việc thực hành phân tích và thiết kế tốt hơn. Bảng 3.4 tóm tắt các hoạt động phát triển hệ thống.

3.3 CÁC PHƯƠNG PHÁP XÂY DỰNG HỆ THỐNG

Tiến trình phát triển hệ thống có thể được tiến hành theo một số cách khác nhau. Phần còn lại của chương này mô tả cho các cách tiếp cận chính được dùng cho việc xây dựng hệ thống: vòng đời hệ thống truyền thống, làm bản mẫu, dùng các gói phần mềm ứng dụng mua sẵn, phát triển của người dùng cuối và khoán ngoài.

Vòng đời hệ thống truyền thống

Vòng đời hệ thống là phương pháp cổ nhất để xây dựng hệ thông tin và vẫn còn được dùng ngày nay cho các dự án hệ thống phức tạp lớn hay trung bình. Phương pháp luận này giả thiết rằng hệ thông tin có vòng đời tương tự như vòng đời của bất kì tổ chức sống nào, với việc bắt đầu, phân giữa và kết thúc. Vòng đời là cách tiếp cận rất chính thức và tuần tự để xây dựng hệ thông tin. Phương pháp này phân hoạch tiến trình xây dựng ra làm sáu giai đoạn: định nghĩa dự án, nghiên cứu hệ thống, thiết kế, lập trình, cài đặt và sau cài đặt. Hình 3.2 minh hoạ cho các giai đoạn này. Mỗi giai đoạn lại bao gồm các hoạt động cơ sở được mô tả ở trên mà phải được thực hiện trước khi giai đoạn tiếp có thể bắt đầu.



Hình 3.2 Phương pháp luận vòng đời cho phát triển hệ thống.

Phương pháp luận vòng đời cũng có sự phân chia lao động rất hình thức giữa người dùng cuối và chuyên viên hệ thông tin. Các chuyên viên kỹ thuật như người phân tích hệ thống và người lập trình đều có trách nhiệm phân nhiều cho việc phân tích hệ thống, thiết kế, và thực hiện; người dùng cuối cùng bị giới hạn vào việc

cung cấp các yêu cầu thông tin và duyệt xét công việc của các nhân viên kỹ thuật. Việc ký tắt hình thức hay các thoả thuận giữa người dùng cuối và các chuyên viên kỹ thuật được yêu cầu khi từng giai đoạn được hoàn tất. Hình 3.2 cũng chỉ ra sản phẩm hay cái ra của từng giai đoạn của vòng đời vốn là cơ sở cho việc ký tắt đó. Bây giờ chúng ta mô tả chi tiết cho các giai đoạn của vòng đời.

CÁC GIAI ĐOẠN CỦA VÒNG ĐỜI

Giai đoạn định nghĩa dự án tập trung vào các khía cạnh phân tích vấn đề sơ bộ của việc phân tích hệ thống. Nó cố gắng trả lời câu hỏi, "Tại sao chúng ta lại cần một dự án hệ thống mới?" và "Chúng ta muốn hoàn thành cái gì?" Giai đoạn này xác định liệu tổ chức có vấn đề hay không và liệu vấn đề đó có thể được giải quyết bằng việc xây dựng một hệ thống tin mới hay bằng việc sửa đổi hệ hiện có. Nếu một dự án hệ thống là cần có, thì đề án được phát triển trong giai đoạn này sẽ xác định ra các mục tiêu tổng quát, xác định phạm vi của dự án, và phát triển một kế hoạch dự án mà có thể được trình bày cho cấp quản lý.

Khảo cứu hệ thống tương ứng với các hoạt động phân tích hệ thống đã được mô tả trước đây trong chương này và tạo ra một báo cáo có chứa những điều sau:

- Một phân tích chi tiết về vấn đề của hệ thống hiện tại (thủ công hay tự động hoá)
- Xác định các mục tiêu cần đạt tới qua giải pháp cho những vấn đề này.
- Các giải pháp phương án.
- Một khảo cứu khả thi cho từng phương án giải pháp.
- Các yêu cầu hệ thống tin mới
- Một kế hoạch dự án mô tả chi tiết cho các hoạt động vòng đời còn lại và các nhiệm vụ cho từng giai đoạn.

Giai đoạn **thiết kế** tạo ra các phương án đặc tả thiết kế logic và vật lí cho giải pháp. Bởi vì phương pháp vòng đời nhấn mạnh vào đặc tả hình thức, nên nhiều dụng cụ thiết kế và làm tài liệu như biểu đồ luồng dữ liệu hay sơ đồ cấu trúc chương trình đều có thể được sử dụng.

Giai đoạn **lập trình** dịch các đặc tả thiết kế đã được tạo ra trong giai đoạn thiết kế thành mã chương trình phần mềm. Người phân tích hệ thống làm việc với người lập trình để chuẩn bị các đặc tả cho từng chương trình trong hệ thống và chuẩn bị kế hoạch kiểm thử cho các modun khác nhau. Những đặc tả chương trình này mô tả cho điều từng chương trình sẽ làm, kiểu của ngôn ngữ lập trình được sử dụng, cái vào và cái ra, logic xử lí, lịch xử lí và các câu lệnh điều khiển như câu lệnh cho việc tuần tự dữ liệu vào. Người lập trình viết mã chương trình đã chuyên biệt hoá điển hình là dùng ngôn ngữ thế hệ ba qui ước như COBOL hay FORTRAN hay ngôn ngữ thế hệ bốn có năng suất cao. Vì các hệ thống lớn có nhiều chương trình với hàng trăm nghìn dòng mã lệnh nên có thể phải cần tới cả toàn bộ tổ lập trình viên.

Giai đoạn **thiết đặt** bao gồm các bước cuối cùng để đưa hệ thống vào vận hành - kiểm thử chấp nhận, huấn luyện và chuyển đổi. Phần mềm được kiểm thử để đảm bảo nó thực hiện đúng từ cả quan điểm kĩ thuật và nghiệp vụ chức năng. Các chuyên gia nghiệp vụ và kĩ thuật được huấn luyện để dùng hệ thống mới. Một chiến lược chuyển đổi chính thức được lựa chọn và một kế hoạch được phát triển để đưa ra một lịch biểu chi tiết cho tất cả các hoạt động cần để thiết lập hệ thống mới. Với việc dùng kế hoạch đó, hệ thống cũ được chuyển đổi sang hệ thống mới.

Giai đoạn **sau thực hiện** bao gồm chủ yếu các hoạt động bảo trì đã mô tả trước đây. Nó cũng bao gồm việc kiểm toán sau thực hiện chính thức được tiến hành bởi người dùng và các chuyên viên kĩ thuật để xác định hệ thống mới đáp ứng tốt đến đâu với các mục tiêu nguyên thủy và liệu có cần bất kì sự xem xét lại hay sửa đổi

không. Qua thời gian, hệ thống có thể yêu cầu quá nhiều bảo trì để vẫn còn hiệu quả và đáp ứng các mục tiêu của người dùng, điều sẽ dẫn tới chấm dứt khoảng sống có ích của nó. Một khi vòng đời hệ thống đi tới chấm dứt, thì một hệ thống hoàn toàn mới là cần tới và chu kỳ có thể bắt đầu lại.

CÁC GIỚI HẠN CỦA CÁCH TIẾP CẬN VÒNG ĐỜI

Vòng đời hệ thống vẫn còn được dùng cho việc xây dựng các hệ thống xử lý giao tác lớn (TPS) và hệ thống tin quản lí (MIS) nơi các yêu cầu mang tính cấu trúc cao và xác định rõ. Nó cũng sẽ vẫn còn thích hợp cho các hệ thống kĩ thuật phức tạp như phóng tàu không gian, kiểm soát không lưu và hoạt động tình luyện. Những ứng dụng như vậy cần một phân tích yêu cầu hình thức và chặt chẽ, các đặc tả định trước, và việc kiểm tra chặt chẽ về tiến trình xây dựng hệ thống. Tuy nhiên, phương pháp luận vòng đời hệ thống lại tốn kém, tốn thời gian và không linh hoạt. Khối lượng tài liệu mới phải được sinh ra và các bước cần được lập lại nếu yêu cầu và đặc tả bị sửa lại. Phương pháp luận này cổ vũ cho việc làm đông cứng đặc tả hệ thống ngay từ đầu của tiến trình phát triển. Nó không thích hợp cho hầu hết các hệ thống để bản vốn chiếm ưu thế trong những năm 1990 và thế kỉ thứ hai mươi mốt, có khuynh hướng kém cấu trúc và nhiều tính cá nhân hơn. Những vấn đề này có thể được giải quyết bằng các chiến lược khác cho việc xây dựng hệ thống sẽ được mô tả sau đây.

Làm bản mẫu

Làm bản mẫu thì kém hình thức hơn phương pháp vòng đời. Thay vì sinh ra khối lượng các đặc tả chi tiết và tài liệu kĩ thuật, thì việc làm bản mẫu lại nhanh chóng sinh ra một mô hình làm việc thực nghiệm của hệ thống cho người dùng cuối để đánh giá. Bằng việc tương tác với bản mẫu, người dùng có thể phát triển một ý tưởng rõ ràng về yêu cầu thông tin của mình. Bản mẫu được người

dùng chấp nhận có thể được dùng như tiêu bản để tạo ra hệ thống cuối cùng.

Bản mẫu là phiên bản làm việc của hệ thống tin hay một phần của hệ thống được ngụ ý chỉ là một mô hình sơ bộ. Một khi làm việc được, thì bản mẫu sẽ được hoàn chỉnh thêm nữa cho tới khi nó tuân thủ chính xác yêu cầu của người dùng. Với nhiều ứng dụng, bản mẫu sẽ được kéo dài và nâng cao lặp đi lặp lại trước khi thiết kế cuối cùng được chấp nhận. Một khi thiết kế đã được hoàn tất, thì bản mẫu có thể được chuyển đổi thành hệ thống sản xuất thành nhả.

Tiến trình xây dựng thiết kế sơ bộ, thử nó, làm mịn nó và lại thử nữa đã được gọi là tiến trình **lập** của việc phát triển hệ thống bởi vì các bước cần cho việc xây dựng hệ thống có thể được lặp đi lặp lại. Mặc dầu cách tiếp cận vòng đời truyền thống bao gồm biện pháp nào đó để làm lại và làm mịn, việc làm bản mẫu thường là lặp lại tường minh và tích cực thúc đẩy thay đổi thiết kế hệ thống. Người ta nói rằng việc làm bản mẫu thay thế việc làm lại không có kế hoạch bằng việc lập có kế hoạch, với mỗi phiên bản lại phản ánh chính xác hơn các yêu cầu của người dùng.

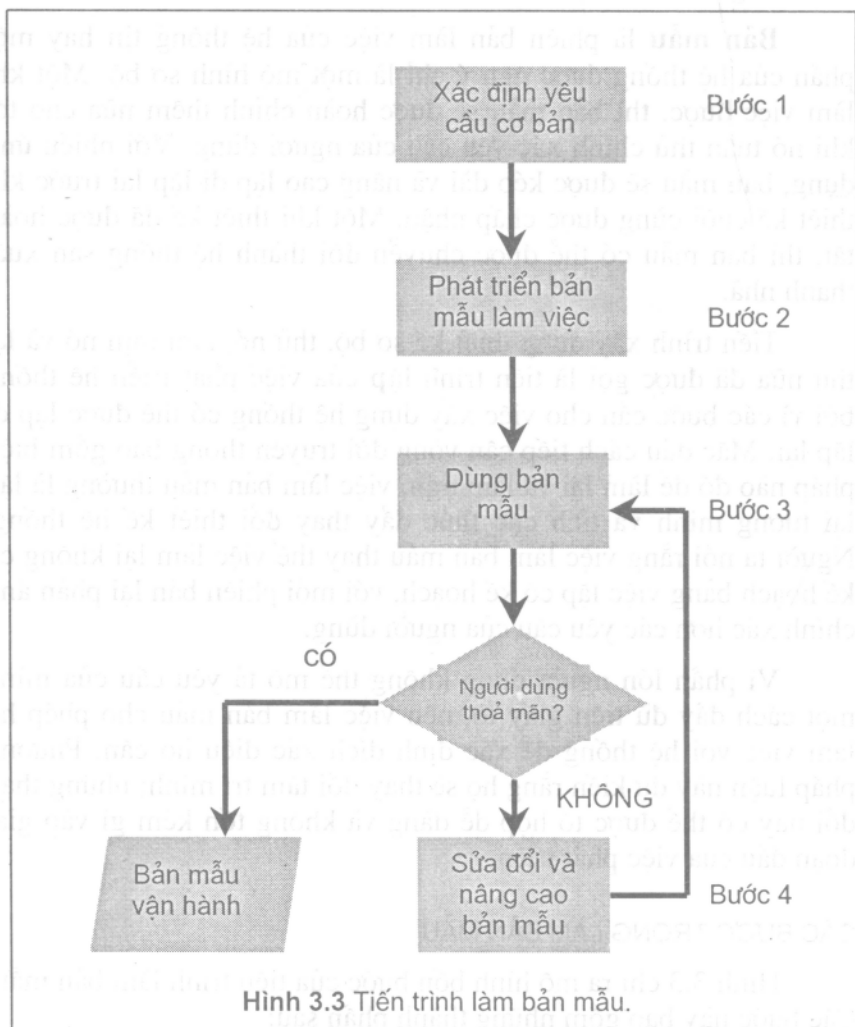
Vì phần lớn người dùng không thể mô tả yêu cầu của mình một cách đầy đủ trên giấy tờ, nên việc làm bản mẫu cho phép họ làm việc với hệ thống để xác định đích xác điều họ cần. Phương pháp luận này dự kiến rằng họ sẽ thay đổi tâm trí mình; những thay đổi này có thể được tổ hợp dễ dàng và không tốn kém gì vào giai đoạn đầu của việc phát triển.

CÁC BƯỚC TRONG LÀM BẢN MẪU

Hình 3.3 chỉ ra mô hình bốn bước của tiến trình làm bản mẫu. Các bước này bao gồm những thành phần sau:

Bước 1. *Xác định yêu cầu cơ bản của người dùng.* Người thiết kế hệ thống (thông thường là một chuyên viên hệ thông tin) làm

việc cùng với người dùng đủ lâu để nắm bắt đủ nhu cầu thông tin của người đó phục vụ cho việc phát triển bản mẫu ban đầu.



Bước 2. Phát triển bản mẫu ban đầu. Người thiết kế hệ thống tạo ra một bản mẫu làm việc nhanh chóng, khả dĩ nhất là dùng các

dụng cụ phần mềm thế hệ bốn để làm tăng tốc việc phát triển. (Một số tính năng của các dụng cụ CASE cũng có thể được dùng cho việc làm bản mẫu.) Bản mẫu có thể chỉ thực hiện những chức năng quan trọng nhất của hệ thống được đề đạt, hay nó có thể chứa toàn bộ hệ thống với một tệp hạn chế.

Bước 3. *Dùng bản mẫu.* Người dùng được cổ vũ làm việc với hệ thống để xác định bản mẫu đáp ứng tốt đến đâu với nhu cầu của mình và đề gợi ý cho việc cải tiến bản mẫu.

Bước 4. *Sửa đổi và nâng cao bản mẫu.* Người xây dựng hệ thống ghi lại tất cả những yêu cầu mà người dùng đề nghị rồi hoàn thiện bản mẫu tương ứng. Sau khi bản mẫu đã được sửa đổi thì chu trình trở về bước 3. Bước 3 và 4 được lặp lại cho tới khi người dùng thoả mãn.

Khi không còn cần lập lại nữa, thì bản mẫu được chấp thuận sẽ cung cấp đặc tả cuối cùng cho ứng dụng. Đôi khi bản thân bản mẫu lại được chấp nhận như phiên bản sản xuất của hệ thống. Việc làm bản mẫu thì nhanh hơn, và không hình thức bằng phương pháp vòng đời hệ thống.

ƯU VÀ NHƯỢC ĐIỂM CỦA LÀM BẢN MẪU

Làm bản mẫu có ích nhất khi có sự không chắc chắn nào đó về các yêu cầu hay giải pháp thiết kế. Chẳng hạn, một hãng bảo hiểm yêu cầu tổng hợp thông tin để phân tích hiệu năng điều hành tài khoản của mình. Nhưng việc đo hiệu năng nên là thế nào? Liệu thông tin có thể được trích ra từ riêng hệ thống nhân sự, hay dữ liệu phải lấy từ hoá đơn khách hàng tổ hợp vào nữa? Khoản mục nào cần được so sánh trong báo cáo? Có cần đưa vào việc xử lý trung gian dựa trên một dạng phân tích thống kê nào đó không? Người quản lý có thể không có khả năng nhìn thấy trước hệ thống sẽ làm việc như thế nào.

Việc làm bản mẫu đặc biệt có giá trị cho **thiết kế giao diện** người dùng cuối của hệ thống tin (phần của hệ thống mà người dùng cuối tương tác với, như màn hình hiển thị trực tuyến, màn hình vào dữ liệu và báo cáo). Bản mẫu làm cho người dùng có khả năng phản ứng ngay lập tức với những phần của hệ thống mà họ cần giải quyết.

Việc làm bản mẫu đã được hoan nghênh rộng rãi như liều thuốc bách bệnh cho các vấn đề cố hữu trong tiến trình phát triển hệ thống truyền thống. Nó cổ vũ cho sự tham dự của người dùng cuối ngay từ đầu và mạnh mẽ vào trong nỗ lực thiết kế và trong toàn bộ vòng đời phát triển hệ thống. Việc làm bản mẫu có thể tạo ra những hệ thống đáp ứng cho yêu cầu của người dùng. Việc làm bản mẫu hứa hẹn xoá bỏ chi phí phát triển quá đáng và lỗi thiết kế. Việc thoả mãn và tinh thần của người dùng thường được nâng cao bởi vì người dùng được giới thiệu với một hệ thống làm việc sơ bộ trong một thời kì rất ngắn.

Việc làm bản mẫu là thích hợp nhất cho những ứng dụng nhỏ - các hệ thống lớn sẽ phải được phân chia ra để cho bản mẫu có thể được xây dựng cho từng phần vào mỗi lúc, điều này là không thể được nếu không có việc phân tích yêu cầu thấu đáo bằng việc dùng cách tiếp cận qui ước.

Việc làm bản mẫu nhanh cũng có thể che đậy các bước bản chất trong phát triển hệ thống. Sự hấp dẫn của bản mẫu được phát triển nhanh chóng và dễ dàng có thể cổ vũ tổ phát triển đi quá nhanh hướng tới mô hình làm việc mà không thấu tóm được tập hợp cơ bản các yêu cầu.

Bước cuối cùng để chuyển bản mẫu thành hệ thống sản xuất thanh nhã có thể không được tiến hành. Một khi được hoàn thành xong, nếu bản mẫu làm việc một cách hợp lý tốt, thì cấp quản lí có thể không thấy nhu cầu cần lập trình lại, thiết kế lại hay kiểm thử đầy đủ, kết quả là hệ thống được xây dựng vội vàng và đó là điều có thể khó khăn cho việc bảo trì và hỗ trợ. Vì bản mẫu thường

không được xây dựng cẩn thận nên hiệu năng kỹ thuật của chúng có thể rất không hiệu quả. Chúng có thể không dễ dàng điều tiết được theo khối lượng lớn dữ liệu hay một số lớn người dùng trong môi trường sản xuất. Hệ thống được làm bản mẫu cũng vẫn cần được làm tài liệu và kiểm thử đầy đủ, nhưng thường những bước này nữa cũng bị lừa dối. Bởi vì bản mẫu được xây dựng với ít nỗ lực, nên người quản lý có thể cho rằng việc kiểm thử là được giải quyết qua người dùng theo cách của họ; bất kỳ sơ suất nào trong kiểm thử cũng đều có thể được sửa chữa thêm về sau. Bởi vì hệ thống dễ dàng được thay đổi thế, nên việc làm tài liệu có thể không giữ thường xuyên được cập nhật.

Phát triển hệ thống với gói phần mềm ứng dụng

Một chiến lược khác là phát triển một hệ thống tin bằng việc mua gói phần mềm ứng dụng. **Gói phần mềm ứng dụng** là một tập các chương trình phần mềm ứng dụng đã viết sẵn, đã được mã trước vốn có sẵn trong thương mại để bán hay cho thuê. Các gói phần mềm ứng dụng có thể biến thiên từ một nhiệm vụ đơn giản (như in nhãn có địa chỉ từ cơ sở dữ liệu trên máy vi tính) cho tới hơn 400 modul chương trình với 500 000 dòng mã lệnh cho hệ thống tính lương máy lớn phức tạp. Các gói phần mềm đang nở hoa bởi vì nhiều ứng dụng là chung cho mọi tổ chức nghiệp vụ. Với những chức năng phổ biến như vậy trong thực hành chuẩn kế toán, một hệ thống tổng quát sẽ đáp ứng cho các yêu cầu của nhiều tổ chức.

Khi một gói phần mềm thích hợp là có sẵn, thì không nhất thiết công ti phải viết lại chương trình của riêng mình. Một gói phần mềm viết sẵn, thiết kế sẵn, đã kiểm thử thông thì có thể đáp ứng cho các yêu cầu và có thể được thay thế nữa. Nó làm giảm khối lượng công việc thiết kế, kiểm thử, thiết đặt và bảo trì. Vì người bán gói phần mềm đã làm rất nhiều việc nên khuôn khổ thời gian và chi phí cho việc phát triển hệ thống mới có thể được rút gọn đáng kể. Việc dùng phần mềm đóng gói cũng giải phóng cho các

nhân viên hệ thống tin nội bộ của tổ chức để làm các việc khác hay phép tổ chức phát triển ra một hệ thống trong khi các nhân viên của nó vẫn còn thiếu tri thức chuyên gia kỹ thuật cần thiết.

LƯU ĐIỂM CỦA GÓI

Dùng các chiến lược phát triển khác, các hoạt động thiết kế có thể dễ dàng tiêu tốn 50 phần trăm hay hơn nỗ lực phát triển. Tuy nhiên với gói phần mềm, các đặc tả thiết kế, cấu trúc tệp, mối quan hệ xử lý, giao tác, và báo cáo đã được người bán gói đó giải quyết để cho phần lớn công việc thiết kế là đã hoàn tất. Chương trình gói phần mềm đã được kiểm thử kỹ càng trước khi đưa ra thị trường cho nên việc kiểm thử của người mua có thể được hoàn tất trong một thời kỳ tương đối ngắn. Người bán cung cấp nhiều việc bảo trì đang tiếp diễn và hỗ trợ cho hệ thống và cung cấp việc nâng cao để giữ cho hệ thống sống hàng với sự phát triển công nghệ. Họ cũng đảm nhận trách nhiệm tiến hành những thay đổi để giữ cho hệ thống tuân thủ theo việc thay đổi những qui định của chính phủ.

Vì chi phí bảo trì có thể chiếm từ 50 đến 80 phần trăm ngân sách của hệ thống tin nên giải pháp gói là một cách để cắt giảm những chi phí này và giải phóng nhân viên nội bộ cho các ứng dụng khác. Người bán gói duy trì nhân viên hỗ trợ thường xuyên với tri thức chuyên gia về gói ứng dụng đặc biệt. Người bán cũng giữ cho hệ thống và tài liệu người dùng hệ thống luôn được cập nhật.

NHƯỢC ĐIỂM CỦA GÓI

Nhược điểm của gói có thể là khá nhiều, thậm chí hầu khắp, đối với hệ thống phức tạp. Phần mềm thương mại còn chưa đạt được mức độ tinh tế và phẩm chất kỹ thuật cần để tạo ra các gói đa năng vốn có thể làm mọi việc tốt mà người dùng muốn trong một ứng dụng đặc thù. Để làm cực đại sự hấp dẫn thị trường, các gói được làm ăn khớp với phần lớn các yêu cầu chung của mọi tổ chức.

Điều gì xảy ra cho các yêu cầu tổ chức duy nhất mà gói không đề cập tới? Với những mức độ thay đổi, người phát triển phần mềm gói thấy trước vấn đề này bằng việc đưa ra các tính năng cho việc chuyên biệt hoá mà không làm thay thế phần mềm cơ sở. Tính năng **chuyên biệt hoá** cho phép gói phần mềm được thay đổi để đáp ứng cho một số yêu cầu duy nhất của tổ chức mà không phá huỷ tính toàn vẹn của phần mềm gói. Chẳng hạn, các gói có thể cấp phát các bộ phận của tệp hay cơ sở dữ liệu của chúng để duy trì những phần dữ liệu duy nhất riêng của tổ chức; số khác có thể có thiết kế modun vốn cho phép khách hàng chọn ra chỉ những chức năng phần mềm họ cần từ một bảng các tùy chọn.

Giá mua ban đầu của gói có thể là một sự lừa dối bởi vì điều này che dấu đi chi phí cài đặt. Một khảo cứu nội bộ của một công ti về chi phí và thời gian cần thiết để thiết đặt sáu gói ứng dụng chính (kể cả lập kế hoạch tài nguyên chế tạo, sổ cái, tài khoản nhận và tài sản cố định) chỉ ra rằng chi phí cài đặt tổng cộng biến thiên từ 1.5 đến 11 lần giá mua gói. Tỷ lệ này là cao nhất đối với các gói có nhiều giao diện với các hệ thống khác. Cùng khảo cứu đó đã chỉ ra rằng chi phí hỗ trợ và quản lí cho năm đầu sau cài đặt trung bình là gấp hai giá mua gói.

CHỌN GÓI PHẦN MỀM

Các gói phần mềm ứng dụng phải được đánh giá kĩ càng trước khi chúng có thể được dùng làm nền tảng cho hệ thống tin mới. Tiêu chuẩn đánh giá quan trọng nhất là các chức năng do gói cung cấp, tính mềm dẻo, thân thiện người dùng, các yêu cầu kĩ thuật, nỗ lực cài đặt và bảo trì, tài liệu, chất lượng người bán, và giá cả. Tiến trình đánh giá gói thường dựa trên **Yêu cầu đề án Request for Proposal (RFP)** vốn là một danh sách chi tiết các vấn đề được đề trình cho người bán phần mềm đóng gói đó. RFP rất có thể bao gồm các vấn đề như sau:

- *Các chức năng được bao hàm:* Bao nhiêu yêu cầu chức năng sẽ được gói đáp ứng?
- *Mềm dẻo:* Gói dễ dàng sửa đổi đến đâu? Tính năng chuyên biệt hoá nào được bao hàm (modun tùy chọn, vùng dữ liệu người dùng?)
- *Thân thiện người dùng:* Gói dễ dùng và dễ kiểm soát đến đâu theo quan điểm phi kĩ thuật? Cần có bao nhiêu huấn luyện để hiểu hệ thống gói?
- *Yêu cầu kĩ thuật:* Gói có thể chạy trên máy tính theo mô hình nào? Hệ điều hành của gói là gì, các yêu cầu lưu giữ đĩa và băng? Loại cơ sở dữ liệu / cấu trúc tệp nào mà gói dùng?
- *Thiết đặt và bảo trì:* Loại tài liệu người dùng và hệ thống nào mà người bán cung cấp cùng gói? Nó có đầy đủ, dễ hiểu và dễ dùng không?
- *Phẩm chất người bán:* Người bán có kinh nghiệm trong lĩnh vực này không? Người bán có số bán lớn và kỉ lục tài chính không? Người bán cung cấp tiện nghi hỗ trợ loại gì? Người bán có nhóm người dùng tích cực không?
- *Giá:* Giá mua hay thuê phần mềm cơ bản là gì? Giá mua bao gồm những gì (modun phụ thêm; tiện nghi trực tuyến, dò tìm hay bộ sinh màn hình; thời gian tư vấn; huấn luyện; hỗ trợ cài đặt)? Có phí bảo trì hàng năm không? Giá vận hành thường năm là bao nhiêu cho khối lượng xử lí đã ước lượng được trông đợi từ gói này?

Khi hệ thống được phát triển bằng việc dùng gói phần mềm ứng dụng, thì việc phân tích hệ thống sẽ bao gồm nỗ lực đánh giá gói. Các hoạt động thiết kế sẽ hội tụ vào việc sánh các yêu cầu với tính năng gói. Thay vì may đo theo các đặc tả thiết kế hệ thống lấy trực tiếp từ yêu cầu người dùng, thì nỗ lực thiết kế sẽ bao gồm việc cố gắng uốn các yêu cầu người dùng tuân thủ theo tính năng của gói.

Khi giải pháp gói phần mềm được lựa chọn, thì tổ chức không còn kiểm soát được toàn bộ vào tiến trình thiết kế hệ thống nữa. Ngay cả với gói chuyên biệt hoá dễ dàng và mềm dẻo nhất thì cũng có những giới hạn về khối lượng may đo được phép. Các tổ chức có kinh nghiệm trong việc dùng phần mềm gói cho các ứng dụng nghiệp vụ chính của mình đã lưu ý rằng ngay cả gói tốt nhất cũng không thể được trông đợi đáp ứng nhiều hơn 70 phần trăm các yêu cầu của tổ chức. Thế 30 phần trăm còn lại thì sao? Chúng sẽ không được thoả mãn bằng gói và phải được thoả mãn bằng phương tiện khác. Nếu gói không thể thích ứng được cho tổ chức, thì tổ chức sẽ phải thích ứng theo gói và thay đổi thủ tục của mình.

Phát triển của người dùng cuối

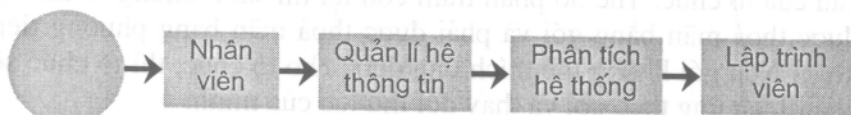
Trong nhiều tổ chức, người dùng cuối đang tự phát triển hệ thông tin ngày càng nhiều với rất ít hay không có sự trợ giúp chính thức từ các chuyên viên kỹ thuật. Hiện tượng này, được gọi là **phát triển của người dùng cuối**, đã được các dụng cụ phần mềm thế hệ bốn làm thành khả dĩ. Với các ngôn ngữ thế hệ bốn, ngôn ngữ đồ hoạ, và dụng cụ máy vi tính, người dùng cuối thường có thể truy nhập vào dữ liệu, tạo ra báo cáo và phát triển toàn bộ hệ thống tin theo cách riêng của mình, ít hay không có sự hỗ trợ của người phân tích hệ thống và người lập trình chuyên nghiệp. Nhiều trong những hệ thống do người dùng cuối này phát triển có thể được tạo ra nhanh chóng hơn nhiều so với vòng đời hệ thống truyền thống. Hình 3.4 minh hoạ cho khái niệm về phát triển của người dùng cuối trong so sánh với phát triển (vòng đời) hệ thống truyền thống.

DỤNG CỤ TÍNH TOÁN NGƯỜI DÙNG CUỐI: SỨC MẠNH VÀ GIỚI HẠN

Nhiều tổ chức đã báo cáo về những cái lợi thấy rõ được trong năng suất phát triển ứng dụng bằng việc dùng các dụng cụ thế hệ bốn. Việc nâng năng suất dựa trên các ngôn ngữ lập trình qui ước, như lập trình có cấu trúc, chỉ tạo ra kết quả cải tiến năng suất tối đa

quãng 25 phần trăm. Tương phản lại, một số nghiên cứu về các tổ chức phát triển ứng dụng bằng các dụng cụ thể hệ bốn đã báo cáo cái lợi năng suất từ 300 đến 500 phần trăm. Các dụng cụ thể hệ bốn cũng có những khả năng mới, như đồ hoạ, trang tính, mô hình hoá và dò tìm thông tin không theo thể thức, điều đáp ứng cho nhu cầu nghiệp vụ quan trọng.

Phát triển (vòng đời) hệ thống truyền thống



Thiết kế Lập trình Kiểm thử

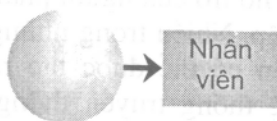
Quản lí cấp cao
và cấp trung

Hàng tuần hay tháng

Phát triển của
người dùng cuối

Dụng cụ tính toán người dùng cuối

Ngôn ngữ vấn đáp
Ngôn ngữ đồ hoạ
Bộ sinh báo cáo
Bộ sinh ứng dụng
Ngôn ngữ cấp rất cao
Dụng cụ máy vi tính



Quản lí cấp cao và
cấp trung

Hàng phút hay ngày

Hình 3.4 Phát triển của người dùng cuối so với phát triển vòng đời.

Nhưng không may các dụng cụ thể hệ bốn vẫn không thể thay thế được các dụng cụ qui ước cho một số ứng dụng nghiệp vụ bởi vì khả năng của chúng vẫn còn giới hạn. Việc xử lý thể hệ bốn còn chưa thật hiệu quả, việc xử lý các giao dịch riêng lẻ quá chậm và giá thành thì quá cao để làm cho các hệ thống này thích hợp với hệ xử lý giao tác lớn.

Phần lớn các dụng cụ thể hệ bốn đều là phi thủ tục và không thể dễ dàng giải quyết cho các ứng dụng với các yêu cầu cập nhật và logic thủ tục nhiều, như hệ thống được dùng để lập lịch biểu sản xuất tối ưu, hay để ghi dấu vết thương mại hàng ngày của kho, trái phiếu và các chứng khoán khác. Logic thủ tục phải được dùng để xác định các chức năng xử lý, chức năng tiện ích, điều kiện giải quyết lỗi, giao diện đặc biệt, và báo cáo chuyên biệt hoá cao độ. Cuối cùng các dụng cụ thể hệ bốn có đóng góp lớn nhất cho việc lập trình và thiết kế chi tiết và có ít tác động lên các hoạt động xây dựng hệ thống khác. Năng suất trong việc phân tích hệ thống, thay đổi thủ tục, chuyển đổi và các khía cạnh khác của thiết kế chủ yếu phụ thuộc vào việc chọn dụng cụ lập trình.

ÍCH LỢI VÀ VẤN ĐỀ QUẢN LÝ

Việc phát triển của người dùng cuối đưa ra nhiều ích lợi cho tổ chức. Những ích lợi này bao gồm những điều sau:

- *Việc xác định yêu cầu được cải tiến* khi người dùng xác định nhu cầu nghiệp vụ riêng của mình.
- *Sự tham dự và thoả mãn của người dùng tăng lên.* Khi người dùng tự phát triển hệ thống của mình và kiểm soát tiến trình phát triển hệ thống thì họ rất có thể dùng hệ thống.
- *Sự ỳn tắc ứng dụng giảm xuống* khi người dùng không còn dựa hoàn toàn vào các chuyên viên hệ thống tin chuyên nghiệp đã quá bận rộn.

Đồng thời, việc phát triển của người dùng cuối độc lập đã tạo ra vấn đề cho tổ chức bởi vì hệ thống được tạo ra theo cách này ở ngoài cơ chế truyền thống cho việc quản lý và kiểm soát hệ thống tin. Phần lớn các tổ chức còn chưa xây dựng các chiến lược để đảm bảo rằng các ứng dụng do người dùng cuối phát triển đáp ứng cho các mục tiêu của tổ chức hay đáp ứng chuẩn đảm bảo chất lượng thích hợp cho chức năng của họ. Khi hệ thống được tạo ra nhanh chóng, không có phương pháp luận phát triển truyền thống, thì việc kiểm thử và tài liệu có thể không thích hợp. Việc kiểm soát tính đầy đủ và hợp lệ có thể bị thiếu đối với dữ liệu vào và việc cập nhật. Việc kiểm toán, kiểm soát vận hành, kiểm soát dữ án và các chuẩn cho giao diện ổn định giữa các hệ con có thể bị bỏ sót.

Việc kiểm soát trên dữ liệu có thể bị mất trong hệ thống vốn ở ngoài bộ phận hệ thống tin truyền thống. Khi người dùng tạo ra ứng dụng và tệp của riêng mình, thì vấn đề sẽ trở thành tăng thêm khó khăn cho việc xác định nơi dữ liệu được định vị và để đảm bảo rằng cùng mẫu thông tin (như số hiệu sản phẩm hay thu nhập hàng năm) được dùng một cách nhất quán trong toàn tổ chức.

CHÍNH SÁCH VÀ THỦ TỤC ĐỂ QUẢN LÝ TÍNH TOÁN NGƯỜI DÙNG CUỐI

Làm sao các tổ chức có thể làm cực đại ích lợi của việc phát triển ứng dụng người dùng cuối trong khi vẫn giữ nó dưới việc kiểm soát quản lý? Bây giờ chúng ta xem xét vài chiến lược.

Một cách vừa tạo điều kiện thuận lợi vừa quản lý việc phát triển ứng dụng người dùng cuối là thiết lập trung tâm thông tin. **Trung tâm thông tin** là một tiện nghi đặc biệt cung cấp việc huấn luyện và hỗ trợ cho tính toán người dùng cuối. Trung tâm thông tin có phần cứng, phần mềm, và các chuyên viên kỹ thuật vốn cung cấp cho người dùng cuối các dụng cụ, huấn luyện và lời khuyên chuyên gia để cho họ có thể tạo ra các ứng dụng hệ thống tin của riêng mình. Các nhà tư vấn trung tâm thông tin luôn có sẵn để hướng dẫn

người dùng và để trợ giúp trong việc phát triển những ứng dụng phức tạp hơn. Nhân viên trung tâm thông tin tổ hợp tri thức chuyên gia về phần cứng, phần mềm, và cơ sở dữ liệu cho các ứng dụng người dùng cuối bằng các kĩ năng trao đổi liên con người. Họ làm việc chủ yếu như thầy giáo và cố vấn cho người dùng, nhưng họ cũng tham gia vào việc phân tích, thiết kế, và lập trình cho nhiều ứng dụng phức tạp. Các dịch vụ điển hình do nhân viên trung tâm thông tin cung cấp bao gồm những điều sau:

- Ước lượng phần cứng và phần mềm mới
- Tư vấn về các dụng cụ và phương pháp luận thích hợp cho việc phát triển ứng dụng.
- Huấn luyện trong ngôn ngữ cấp cao và dụng cụ phát triển
- Sinh ra và sửa đổi bản mẫu.
- Trợ giúp trong chương trình gỡ lỗi
- Trợ giúp trong việc truy nhập dữ liệu
- Trợ giúp cho các ứng dụng, vấn đáp và báo cáo có yêu cầu các ngôn ngữ lập trình
- Cung cấp mối liên hệ với các nhóm xử lí thông tin khác (như các chuyên viên cơ sở dữ liệu) vốn hỗ trợ về tài nguyên trung tâm thông tin.

Trung tâm thông tin cung cấp nhiều ích lợi quản lí. Họ có thể giúp người dùng cuối tìm ra các dụng cụ và ứng dụng sẽ làm cho họ năng suất hơn. Họ ngăn cản việc tạo ra các ứng dụng dư thừa. Họ thúc đẩy việc dùng chung dữ liệu và tối thiểu hoá vấn đề toàn vẹn. Họ đảm bảo rằng ứng dụng được phát triển bởi người dùng cuối đáp ứng việc kiểm toán, chất lượng dữ liệu và chuẩn an toàn.

Một ích lợi khác của trung tâm thông tin là ở chỗ họ có thể giúp thiết lập và áp đặt các chuẩn công ti về phần cứng và phần mềm để cho người dùng cuối không đưa các công nghệ không tương hợp và tách biệt vào trong tổ chức. Trung tâm thông tin sẽ chỉ trợ giúp cho người dùng bằng phần cứng và phần mềm mà đã được cấp quản lí chấp thuận.

Bên cạnh việc dùng các trung tâm thông tin, người quản lý có thể theo đuổi các chiến lược khác để đảm bảo rằng tính toán người dùng cuối phục vụ cho các mục tiêu lớn hơn của tổ chức. Cấp quản lý nên kiểm soát việc phát triển các ứng dụng người dùng cuối bằng việc tổ hợp chúng vào trong kế hoạch hệ thống chiến lược. Phương pháp luận cho việc thiết lập các yêu cầu thông tin qui mô cả tổ chức, đã được mô tả ở đầu chương này có thể giúp xác định các ứng dụng người dùng cuối có lợi cho toàn bộ tổ chức. Cấp quản lý cũng nên phát triển các kiểm soát về việc phát triển người dùng cuối chủ chốt, như nhấn mạnh vào việc biện minh chi phí về các dự án hệ thống tin người dùng cuối và thiết lập các chuẩn phân cứng, phần mềm và chất lượng cho các ứng dụng người dùng phát triển.

Khoán ngoài các hệ thống tin

Nếu tổ chức không muốn dùng tài nguyên nội bộ của mình để xây dựng và vận hành hệ thống tin, thì nó có thể thuê một tổ chức ngoài vốn chuyên môn cung cấp các dịch vụ để làm việc này. Tiến trình chuyển giao việc vận hành trung tâm máy tính, mạng viễn thông, hay việc phát triển ứng dụng của một tổ chức cho người bán các dịch vụ này ở bên ngoài được gọi là **khoán ngoài**. Trong tổ chức mà chi phí cho vận hành của hệ thống thông tin đã tăng lên nhanh chóng, thì người quản lý cũng quay sang việc khoán ngoài để kiểm soát chi phí đó.

ƯU VÀ NHƯỢC ĐIỂM CỦA KHOÁN NGOÀI

Khoán ngoài đang trở thành phổ biến bởi vì một số tổ chức nhận thấy điều đó có tỉ lệ chi phí-lợi ích hơn là nó phải duy trì trung tâm máy tính riêng của mình và nhân viên hệ thống tin. Người cung cấp dịch vụ khoán ngoài có thể được lợi do việc tiết kiệm qui mô (cùng tri thức, kĩ năng và khả năng có thể được chia sẻ cho nhiều khách hàng khác nhau) và rất có thể đặt giá rất cạnh tranh về dịch vụ hệ thống tin. Việc khoán ngoài cho phép một tổ

chức với nhu cầu thăng giáng về xử lý thông tin chỉ phải trả cho đúng điều nó dùng thay vì phải xây dựng trung tâm máy tính riêng của mình mà lại dùng không hết công suất khi chưa tới cao điểm tải. Một số tổ chức làm khoán ngoài bởi vì nhân viên hệ thống tin nội bộ của họ không thể giữ cùng nhịp với sự thay đổi công nghệ hay bởi vì họ muốn giải phóng tài năng khan hiếm và đắt đỏ cho các hoạt động được trả lương cao hơn.

Nhưng không phải tất cả các tổ chức đều được lợi từ việc khoán ngoài, và nhược điểm của khoán ngoài có thể tạo ra những vấn đề nghiêm trọng cho tổ chức nếu chúng không được hiểu rõ và quản lí. Khi một tổ chức cho thuê trách nhiệm phát triển và vận hành hệ thống tin của nó, thì nó có thể mất kiểm soát về chức năng hệ thống tin của mình. Khoán ngoài có thể đặt người bán vào vị trí có lợi thế, nơi mà khách hàng phải chấp nhận bất kì cái gì người bán làm ra và bất kì phí nào mà người bán đặt ra. Sự phụ thuộc này cuối cùng gây ra chi phí cao hay mất kiểm soát về chiều hướng kĩ thuật. Các bí mật thương mại hay thông tin tài sản riêng có thể rò rỉ sang kẻ cạnh tranh bởi vì hệ thống tin của tổ chức đang được vận hành hay phụ thuộc vào người ngoài. Điều này có thể đặc biệt có hại nếu tổ chức cho phép người khoán ngoài phát triển hay vận hành các ứng dụng vốn cho nó một kiểu ưu thế cạnh tranh nào đó.

KHI NÀO THÌ DÙNG KHOÁN NGOÀI

Nếu việc phát triển hệ thống và chức năng vận hành được quản lí tốt và có hiệu quả, thì người bán bên ngoài chẳng có thể cung cấp được ích lợi gì ngay. Tuy nhiên, có một số hoàn cảnh mà việc khoán ngoài lại rất có nghĩa:

- *Khi có cơ hội hạn chế cho tổ chức để phân biệt bản thân nó về mặt cạnh tranh qua một ứng dụng hệ thống tin đặc biệt hay cả loạt ứng dụng.*
- *Khi tính dự đoán được về dịch vụ hệ thống tin không gián đoạn là không quan trọng lắm.*

- *Khi khả năng hệ thống tin hiện có của tổ chức là giới hạn, không hiệu quả hay yếu kém về kỹ thuật. Chẳng hạn họ có thể dùng người khoán ngoài để giúp mình chuyển dịch từ tính toán dựa trên máy lớn truyền thống sang môi trường tính toán khách/nguồn phục vụ.*

Lập chiến lược công nghệ là một miền mà tổ chức không nên để cho người ngoài làm. Nhiệm vụ chiến lược tốt nhất là phải giữ trong nhà.

Bảng 3.5 so sánh ưu và nhược điểm của từng phương án xây dựng hệ thống đã được mô tả trong chương này.

Bảng 3.5 So sánh các cách tiếp cận phát triển hệ thống

Cách tiếp cận	Tính năng	Ưu điểm	Nhược điểm
Vòng đời hệ thống	Tiến trình tuần tự từng bước Các đặc tả viết và chấp thuận Vai trò giới hạn của người dùng	Cần thiết cho các hệ thống và dự án lớn phức tạp	Chậm và tốn kém Không khuyến khích thay đổi Giấy tờ nhiều để quản lý
Làm bản mẫu	Yêu cầu được xác định một cách động bằng hệ thống thực nghiệm	Nhanh và không đắt Có ích khi yêu cầu không chắc chắn hay khi giao diện người dùng cuối là quan trọng Thúc đẩy sự tham gia của người dùng	Không thích hợp cho các hệ thống lớn phức tạp Có thể che đậy các bước quan trọng phân tích, làm tài liệu và kiểm thử

Bảng 3.5 (tiếp theo)

Gói phần mềm ứng dụng	Phần mềm thương mại làm giảm nhu cầu về các chương trình phần mềm được phát triển nội bộ	Công việc thiết kế, lập trình, cài đặt và bảo trì được giảm bớt	Có thể không đáp ứng các yêu cầu duy nhất của tổ chức
		Có thể tiết kiệm thời gian và chi phí khi phát triển các ứng dụng nghiệp vụ chung	Có thể không thực hiện nhiều chức năng nghiệp vụ tốt
		Giảm bớt nhu cầu tài nguyên hệ thống tin nội bộ	Việc chuyên biệt hoá nhiều làm tăng chi phí phát triển
Phát triển người dùng cuối	Hệ thống được tạo ra bằng các dụng cụ phần mềm thể hệ bốn	Người dùng kiểm soát việc xây dựng hệ thống	Có thể dẫn tới sinh sôi nảy nở hệ thống tin không kiểm soát được
	Nhanh và không hình thức	Tiết kiệm thời gian và chi phí phát triển	Hệ thống không phải bao giờ cũng đáp ứng các chuẩn đảm bảo chất lượng
	Vai trò tối thiểu của chuyên viên hệ thống tin	Giảm sự ỳn đồng ứng dụng	
Khoán ngoài	Hệ thống được xây dựng và đôi khi được vận hành bởi người bán bên ngoài	Có thể làm giảm hay kiểm soát chi phí	Mất kiểm soát về chức năng hệ thống tin
		Có thể tạo ra hệ thống khi tài nguyên nội bộ không sẵn có hay thiếu hụt về mặt kĩ thuật	Tuỳ thuộc vào hướng kĩ thuật và sự thành công của người bán bên ngoài

3.4 XÂY DỰNG HỆ THỐNG THÀNH CÔNG: CHẤT LƯỢNG VÀ THỰC HIỆN

Trong gần như mọi tổ chức, hệ thống tin chiếm nhiều thời gian và tiền bạc để thực hiện hơn việc hình dung ban đầu, hay hệ thống được hoàn thành thì lại làm việc không đúng. Bởi vì quá nhiều hệ thống tin gây ra rắc rối, nên những người thiết kế, người xây dựng và người dùng hệ thống tin nên hiểu tại sao chúng lại thành công hay thất bại. Các nhân tố quản lý và tổ chức cũng đóng vai trò mạnh mẽ trong kết quả hệ thống bởi vì việc thực hiện một hệ thống tin là một tiến trình thay đổi tổ chức.

Chúng ta đã cố gắng thám hiểm nhiều phương diện của hệ thống tin thành công trong cả cuốn sách này, nhưng vấn đề về chất lượng và sự thành công của hệ thống tin là tiêu điểm của mục này. Sau khi xem xét ban đầu về chất lượng hệ thống tin và các tính năng của hệ thống thành công, chúng ta sẽ nêu đại cương điểm đặc thù của vấn đề về hệ thống tin. Thế rồi chúng ta mô tả các phương pháp luận truyền thống và mới cùng các dụng cụ để cải tiến chất lượng phần mềm và tính hiệu quả hệ thống tổng thể. Cuối cùng chúng ta sẽ thám hiểm các nhân tố quản lý và tổ chức vốn ảnh hưởng tới kết quả của hệ thống bằng việc nhìn vào tiến trình thực hiện.

3.4.1 HỆ THỐNG TIN THÀNH CÔNG LÀ GÌ?

Điều gì tạo ra hệ thống tin thành công? Đây không phải bao giờ cũng là câu hỏi dễ trả lời bởi vì không phải ai cũng có thể đồng ý về giá trị của tính hiệu quả của hệ thống tin đặc biệt. Nhưng nếu chúng ta phải tổng quát hoá, thì hệ thống thành công sẽ là chính xác, tin cậy, sẽ làm việc như được dự định, và sẽ được sử dụng rộng rãi. Cái giá cho việc phát triển hệ thống sẽ nằm trong những giới hạn tài chính và ngân sách đã được tổ chức thiết lập. Hệ thống sẽ có chất lượng cao.

Một hệ thống có chất lượng sẽ làm những điều sau:

- Đạt tới các mục tiêu nghiệp vụ do bộ phận người dùng nói ra.
- Vận hành với giá chấp nhận được, xứng với giá trị được tạo ra cho tổ chức.
- Đáp ứng các tiêu chuẩn hiệu năng đã được xác định cẩn thận (như thời gian đáp ứng và tính sẵn có hệ thống).
- Tạo ra cái ra chính xác, tin cậy.
- Dễ học và dùng
- Mềm dẻo

Hệ thống tin thành công giúp thúc đẩy chất lượng toàn diện trong toàn tổ chức, đóng góp cho việc quản lý chất lượng toàn diện. Quản lý chất lượng toàn diện (TQM) là một khái niệm được người Nhật phổ biến, vốn làm cho chất lượng là trách nhiệm được chia đều cho mọi người trong tổ chức, với thành tựu của việc kiểm soát chất lượng xem như mục đích tự thân nó. Mọi người đều được trông đợi đóng góp cho việc cải tiến chất lượng toàn diện - người kỹ sư tránh lỗi thiết kế, người công nhân sản xuất phát hiện khiếm khuyết, người đại diện bán hàng giới thiệu sản phẩm đúng đắn, và người thư ký tránh những lỗi gõ sai. Quản lý chất lượng toàn diện bao gồm tất cả các chức năng của tổ chức.

Quản lý chất lượng toàn diện được người Nhật phổ biến, người trao trách nhiệm về sự nhất quán chất lượng cho các công nhân, những người thực tế tạo ra sản phẩm và dịch vụ, đối lập với bộ phận kiểm tra chất lượng. Cấp quản lý Nhật chấp nhận mục tiêu khiếm khuyết không có, hội tụ vào việc cải tiến sản phẩm và dịch vụ của họ trước khi chuyển giao hơn là sửa lại chúng sau khi chuyển giao. Các khảo cứu đã thường được chỉ ra rằng một vấn đề càng được phát hiện sớm sửa trong chu trình nghiệp vụ thì càng ít gây tổn kém cho cho tổ chức để khử bỏ nó. Do vậy cách tiếp cận chất lượng của người Nhật đem lại không chỉ sự dịch chuyển trong

việc tập trung vào công nhân và tôn trọng ngày càng nhiều về chất lượng sản phẩm và dịch vụ, mà còn làm giảm giá thành nữa.

Cách hệ thống tin đóng góp cho việc quản lí chất lượng toàn diện

Hệ thống tin có thể đóng góp cho chất lượng tổng thể bởi vì hệ thống này dính líu sâu sắc với công việc hàng ngày của các bộ phận khác trong toàn bộ tổ chức. Chẳng hạn người phân tích hệ thống tin thường giữ vai trò dẫn đầu trong việc thiết kế những hệ thống phòng ban đa dạng như tính lương, nghiên cứu bằng phát minh và kiểm soát các tiến trình hoá học. Các chuyên gia hệ thống tin duy trì tri thức của họ về các phòng ban đó qua cả sự hỗ trợ tiếp diễn của mình cho những hệ thống đó lẫn sự tham dự của họ vào việc hoạch định thông tin cho các phòng ban. Bên cạnh đó, thông thường chỉ các nhân viên hệ thống tin mới biết dữ liệu nào đó gốc gác ở đâu, các phòng ban khác dùng chúng và cất giữ chúng thế nào, và các chức năng nào khác sẽ có ích từ việc truy nhập vào chúng. Với hiểu biết rộng rãi này về việc tích hợp chức năng của cả tổ chức, nhân viên hệ thống tin có thể là thành viên có giá trị cho bất kì tổ dự án chất lượng nào.

Nhân viên trong các bộ phận hệ thống tin tốt có những kĩ năng vốn là nhân tố chủ chốt cho sự thành công của chương trình chất lượng. Những người phân tích hệ thống tin giỏi đều được huấn luyện và có kinh nghiệm trong việc phân tích và thiết kế lại các tiến trình kinh doanh. Điều đó sau rốt chính là nghề của họ. Nhân viên hệ thống tin là nguồn các ý tưởng về ứng dụng công nghệ cho các vấn đề chất lượng và thường là những người làm cho công nghệ đó thành có sẵn cho dự án chất lượng.

Phân nhiều các công việc hệ thống tin là rất kĩ thuật. Hậu quả là các chuyên viên hệ thống tin có thể quá dễ dàng nhìn công nghệ như câu trả lời cho mọi vấn đề chất lượng. Trong khi công nghệ có

nhiều đóng góp, thì chất lượng thường có nghĩa là hội tụ vào các vấn đề nghiệp vụ. Sau rốt công nghệ chỉ là dụng cụ hay tập hợp các dụng cụ mà, nếu được dùng một cách đúng đắn, thì sẽ hỗ trợ cho các mục tiêu nghiệp vụ của phòng ban hay tổ chức.

Vấn đề với hệ thống tin

Không may là có đến 75 phần trăm các hệ thống tin lớn vận hành sai lỗi đến mức không thể giúp được cho tổ chức đáp ứng cho mục tiêu chất lượng của nó. Mặc dầu những hệ thống này vẫn đang trong sản xuất, chúng làm tốn quá nhiều thời gian và tiền bạc phụ để thực hiện hay khiếm khuyết về chức năng đến mức phía nghiệp vụ không thể nào thu hoạch được ích lợi trông đợi. Nhiều sai hỏng hệ thống tin không nhất thiết là dẫn tới sụp đổ, nhưng chúng rõ ràng hoặc không được dùng theo cách chúng đã được dự định, hoặc chúng chẳng được dùng chút nào. Người dùng phải phát triển các thủ tục thủ công song song để làm cho các hệ thống này làm việc cho đúng. Những sai hỏng này có thể lần được về bốn vấn đề hệ thống tin cơ sở: thiết kế, dữ liệu, chi phí và vận hành.

THIẾT KẾ

Thiết kế thực tế của hệ thống không thấu tóm được các yêu cầu nghiệp vụ bản chất. Thông tin có thể không được hệ thống cung cấp đủ nhanh hay dưới dạng dùng được. Nó thậm chí còn có thể cung cấp những phần thông tin sai. Một số hệ thống trở thành vô dụng bởi vì chúng đã được thiết kế với **giao diện người dùng** quá nghèo nàn (phần của hệ thống mà người dùng tương tác với). Chẳng hạn, một mẫu đưa vào hay màn hình vào dữ liệu trực tuyến có thể được bố trí quá nghèo nàn đến mức chẳng ai muốn đệ trình dữ liệu cả. Hay các thủ tục yêu cầu tìm thông tin trực tuyến có thể không thông minh.

DỮ LIỆU

Dữ liệu trong hệ thống có một mức độ không chính xác cao hay không đầy đủ, hay chúng không thể được chia ra đúng theo mục tiêu nghiệp vụ.

CHI PHÍ

Một số hệ thống vận hành hoàn toàn ổn thoả về chi phí tới mức việc thực hiện hay vận hành hệ thống tin mới không thể nào biện minh được về giá trị đối với nghiệp vụ.

VẬN HÀNH

Thông tin không được cung cấp đúng hạn và theo cách có hiệu quả bởi vì việc vận hành máy tính xử lí thông tin có thể bị gián đoạn. Bởi vì phải chạy đi chạy lại chương trình quá nhiều, lịch biểu bị trễ, và thời gian dành cho việc sửa chữa chương trình, nên nhân viên hệ thống tin không có thời gian để làm việc với các giải pháp dài hạn. Hệ thống trực tuyến có thể vận hành không thích hợp bởi vì thời gian đáp ứng quá lâu.

Những vấn đề này có thể do các nhân tố kĩ thuật, quản lí và tổ chức gây ra. Trong mục này chúng ta tập trung vào các nhân tố kĩ thuật, đặc biệt là các nhân tố chất lượng phần mềm suy yếu. Các mục sau sẽ xét tới các nhân tố quản lí và tổ chức đằng sau những vấn đề hệ thống này.

Nhu cầu về đảm bảo chất lượng phần mềm

Các vấn đề hệ thống thường phản ánh vào phần mềm vốn có phẩm chất kém. Một lỗi không được phát hiện ra trong phần mềm tin dụng của một công ti hay phần mềm kiểm soát tiến trình có thể gây ra tổn thất tới hàng triệu đô la. Hành khách hiện đại và xe cộ thương mại ngày càng phụ thuộc nhiều vào các chương trình máy tính với các chức năng chủ chốt. Một khiếm khuyết phần mềm ẩn

trong hệ thống phanh có thể gây ra tổn thất nhân mạng. Một hồng hóc trong hệ thống kiểm soát không lưu vào những khoảnh khắc căng thẳng có thể cản trở người kiểm soát ngăn chặn hai máy bay hành khách đâm nhau, gây cho hàng trăm người chết.

LỖI VÀ KHIẾM KHUYẾT

Vấn đề chính với phần mềm là sự hiện diện của lỗi ẩn, hay khiếm khuyết mã chương trình. Các nghiên cứu đã chỉ ra rằng thực tế không thể nào khử bỏ được tất cả các lỗi khỏi một chương trình lớn. Nguồn gốc chính của lỗi là do độ phức tạp của mã ra quyết định. Ngay cả một chương trình tương đối nhỏ vài trăm dòng lệnh cũng chứa tới hàng chục quyết định dẫn tới hàng trăm hay thậm chí hàng nghìn đường chạy khác nhau. Các chương trình quan trọng bên trong hầu hết các tổ chức thường lớn hơn thế nhiều lắm, chứa tới mười nghìn hay thậm chí hàng triệu dòng mã, mỗi dòng lại chứa nhiều chọn lựa và đường chạy của những chương trình nhỏ hơn. Độ phức tạp như thế thì khó mà làm tài liệu và thiết kế - người thiết kế làm tài liệu cho một số phản ứng sai hay không xét tới một số khả năng nào đó.

Không khiếm khuyết, một mục tiêu của phong trào quản lí chất lượng toàn diện, không thể nào đạt tới được trong các chương trình lớn. Việc kiểm thử đầy đủ đơn giản là không thể làm được. Các chương trình kiểm thử đầy đủ có chứa tới hàng nghìn chọn lựa và hàng triệu đường chạy sẽ đòi hỏi mất hàng nghìn năm tính toán. Chúng ta không thể nào khử bỏ được hết tất cả các lỗi, mà chúng ta thậm chí cũng không thể nào biết được chắc chắn độ nghiêm trọng của những lỗi vẫn còn lại. Ngay cả khi tìm được lỗi, thì cũng lại khó mà loại bỏ chúng đi được. Trong nhiều trường hợp, nỗ lực để sửa lỗi sẽ lại đưa vào lỗi hoàn toàn mới hay cả loạt lỗi.

KHÔNG CÓ KHẢ NĂNG ĐÁP ỨNG NHU CẦU NGƯỜI DÙNG

Việc phát triển phần mềm thực tế đáp ứng cho nhu cầu người dùng có thể là khó khăn trong một tiến trình mà người dùng cuối giao phó sản phẩm đó cho người làm trước khi sản phẩm đó được xây dựng. Trong thực tế, hệ thống cuối cùng đã "được mua" trước, mua mà "chưa hề thấy". Việc xác định nhu cầu người dùng và đánh giá chất lượng của hệ thống được hoàn thành đã chứng tỏ là những thách thức lớn lao. Phần lớn các dự án phát triển hệ thống đều bắt đầu bằng việc xác định các yêu cầu thông tin người dùng và các đặc tả dưới dạng phân tích hệ thống và tài liệu thiết kế.

Một vấn đề là ở chỗ việc đáp ứng các đặc tả không nhất thiết đảm bảo chất lượng. Hệ thống được hoàn thành trong thực tế có thể đáp ứng cho các đặc tả nhưng lại không thoả mãn nhu cầu người dùng. Điều này xuất hiện bởi vì các đặc tả không chính xác, không đầy đủ hay không làm chi tiết đúng đắn; các chức năng bị bỏ quên không nói tới trong đặc tả; hay việc thay đổi nhu cầu của người dùng trong thời kỳ phát triển. Thường chất lượng không được định lượng trong đặc tả cho nên việc đánh giá về chất lượng hệ thống trở thành mang tính chủ quan. Kiểu tình huống này có thể dễ dàng suy đồi thành việc trở mặt trách móc giữa nhóm làm hệ thống tin và cộng đồng người dùng, mỗi bên lại công kích bên kia trong khi không dả động gì với phẩm chất hệ thống.

Đặc tả thường không xét tới hệ thống theo góc nhìn của người dùng. Trong khi người thiết kế sẽ tập trung vào tính chức năng, thì họ thường lơ là tính dễ học và dễ dùng, độ chính xác và độ tin cậy, hay tốc độ đáp ứng. Tất cả những nhân tố này đều quan trọng với người dùng hệ thống và do vậy quan trọng cho sự thành công của hệ thống.

Thời gian đáp ứng hệ thống là một thí dụ chung về một đặc tả chi tiết vốn vẫn bị bỏ qua hay không được xác định thích hợp. Một đại diện dịch vụ khách hàng, người dùng cuối trong chức năng dịch vụ khách hàng có thể cần một thời gian đáp ứng hỏi đáp không

nhiều hơn năm giây, nhưng các đặc tả có thể không chứa tham chiếu nào tới thời gian đáp ứng. Nếu vào lúc chuyển giao thời gian đáp ứng là mười giây, thì hệ thống sẽ đáp ứng đặc tả nhưng không đáp ứng nhu cầu người dùng. Có lẽ thời gian đáp ứng năm giây được xác định và được đạt tới trong khi kiểm thử với một hay hai người dùng đồng thời. Hệ thống sẽ thoả mãn đặc tả này nếu thời gian đáp ứng giảm xuống 20 giây khi 100 người dùng đồng thời thì có được không? Nếu hệ thống đáp ứng trong năm giây cho 80 phần trăm thời gian khi 100 người đang trong hệ thống, thì hệ thống có đáp ứng đặc tả của nó không nếu nó không đáp ứng phần đặc biệt của 20 phần trăm thời gian? Việc thiếu các chuẩn được nhất trí, do được trong đặc tả dẫn tới người dùng không thoả mãn và hệ thống không đáp ứng cho nhu cầu người dùng.

ÁC MỘNG BẢO TRÌ

Phần mềm máy tính có thể tạo ra các vấn đề hệ thống tin bởi vì về truyền thống nó đã phức tạp cho bảo trì đến thế rồi. Bảo trì, tiến trình sửa đổi hệ thống trong việc dùng sản xuất, là pha tốn kém nhất của tiến trình phát triển hệ thống. Bảng 3.6 chỉ ra kích cỡ của vấn đề bảo trì. Trong một phần năm các bộ phận hệ thống tin, 85 phần trăm giờ nhân viên được dành cho bảo trì, để lại rất ít thời gian cho việc phát triển hệ thống mới. Trong phần lớn các cửa hiệu, gần một nửa thời gian nhân viên chuyên môn được dành cho việc bảo trì các hệ thống hiện có.

Tại sao giá bảo trì lại cao thế? Một lí do chính là ở sự thay đổi tổ chức. Tổ chức có thể trải qua những thay đổi nội bộ lớn về cấu trúc hay quyền lãnh đạo, hay thay đổi có thể tới từ môi trường xung quanh. Những thay đổi này ảnh hưởng tới yêu cầu thông tin. Một nguyên nhân thông thường cho vấn đề bảo trì lâu dài là việc phân tích và thiết kế hệ thống có lỗi, đặc biệt là phân tích yêu cầu thông tin. Một số khảo cứu về các hệ thống lớn cho thấy rằng đa số lỗi hệ thống - 64 phần trăm - nảy sinh từ lỗi phân tích ban đầu.

Bảng 3.6 VẤN ĐỀ BẢO TRÌ

Giờ nhân viên hàng năm	
Bảo trì và nâng cao hệ thống hiện tại	48.0 %
Phát triển hệ thống mới	46.1 %
Các vấn đề khác	5.9 %
20% việc điều tra dành 85% nỗ lực của nó cho việc bảo trì và nâng cao	
Tần số hoạt động	
Lỗi - khẩn cấp	17.4 %
Thay đổi - dữ liệu, cái vào, tệp, phần cứng	18.2 %
Cải tiến - nâng cao, hiệu quả, tài liệu	60.3 %
Các vấn đề khác	4.1%

Nếu lỗi được phát hiện sớm, trong khi phân tích và thiết kế, thì chi phí cho nỗ lực phát triển hệ thống là nhỏ. Nhưng nếu chúng không được phát hiện ra mãi tới sau khi lập trình, kiểm thử hay việc chuyển đổi đã hoàn tất, thì chi phí có thể lên vọt theo con số thiên văn. Chẳng hạn một lỗi logic nhỏ, chỉ mất một giờ để sửa trong giai đoạn phân tích và thiết kế có thể mất 10, 40 và 90 lần lâu hơn để sửa trong các giai đoạn lập trình, chuyển đổi và hậu cài đặt tương ứng.

Để có thể giải quyết việc bảo trì một cách nhanh chóng và không tốn kém, hệ thống phần mềm phải mềm dẻo. Hệ thống mềm dẻo có thể nhanh chóng và dễ dàng được sửa chữa khi vấn đề xuất hiện. Một hệ thống như vậy cũng có thể được sửa đổi khi yêu cầu nghiệp vụ thay đổi hàng năm. Chẳng hạn hệ thống bán hàng phải có khả năng điều tiết theo các sản phẩm mới, các nhân viên bán hàng mới và thậm chí các văn phòng mới với vài hay không vấn đề gì. Tính không mềm dẻo cũng là một vấn đề chung, đặc biệt bởi vì nhiều người thiết kế không xem xét tới nhu cầu thay đổi khi họ thiết kế hệ thống mới.

Tuy nhiên, cho dù họ có thiết kế hệ thống mềm dẻo, thì tính mềm dẻo có thể dường như là đắt đỏ và tốn thời gian. Lợi ích của nó không phải bao giờ cũng được người dùng hiểu rõ và đánh giá đúng. Ngày tháng hoàn thành và giới hạn chi phí thường buộc người phát triển phải hi sinh tính mềm dẻo để hoàn thành một dự án như đã hứa. Không may là tổ chức có thể phải trả giá cao hơn nhiều về sau.

3.4.2 GIẢI PHÁP CHO VẤN ĐỀ CHẤT LƯỢNG PHẦN MỀM

Hệ thống tin là phức tạp, và giải pháp cho vấn đề hệ thống tin cũng phức tạp tương tự. Mục này tập trung vào các giải pháp kỹ thuật cho vấn đề liên quan tới chất lượng phần mềm. Mục sau sẽ mô tả các giải pháp quản lý và tổ chức cho vấn đề hệ thống tin.

Các giải pháp cho vấn đề chất lượng phần mềm bao gồm việc dùng một phương pháp luận phát triển hệ thống thích hợp, cấp phát tài nguyên đúng đắn trong việc phát triển hệ thống, dùng các các độ đo, chú ý tới kiểm thử và dùng các dụng cụ chất lượng.

Dụng cụ chất lượng và phương pháp luận

Phương pháp luận phát triển thực tế là một tuyển tập các phương pháp, một hay nhiều phương pháp cho mọi hoạt động bên trong mọi pha của dự án phát triển. Người dùng một phương pháp luận hiếm khi dùng mọi phương pháp trong nó trong một dự án bởi vì hầu hết các dự án không đòi hỏi mọi hoạt động có thể có. Nhiều phương pháp luận có ích đã tồn tại, một số thích hợp với những công nghệ đặc biệt, số khác phản ánh các triết lý phát triển khác nhau. Bộ phận hệ thống tin, tiếp nối với cấp quản lý hay các phòng ban khác, chọn phương pháp luận được dùng với các công nghệ khác. Tuy nhiên, chìa khoá cho việc phát triển chất lượng là chọn ra một phương pháp luận thích hợp và thế rồi bắt buộc dùng nó. Một số phương pháp luận nhấn mạnh và lưu đồ và biểu đồ, số khác dùng mô tả lời. Phương pháp luận phải yêu cầu các tài liệu yêu cầu hệ

thống và tài liệu đặc tả vốn đầy đủ, chi tiết và chính xác, và được làm tài liệu theo định dạng mà cộng đồng người dùng có thể hiểu được trước khi chấp thuận nó. Các đặc tả cũng phải bao gồm cách đo đã được thoả thuận về chất lượng hệ thống để cho hệ thống có thể được đánh giá một cách khách quan trong khi nó được phát triển và khi nó được hoàn tất. Chúng ta không thể quá cường điệu rằng các đặc tả chất lượng - các cách biểu diễn rõ ràng, chính xác cho nhu cầu người dùng - là điều mấu chốt cho sự phát triển một hệ thống chất lượng.

CẤP PHÁT TÀI NGUYÊN TRONG PHÁT TRIỂN HỆ THỐNG

Cái nhìn về **cấp phát tài nguyên** trong khi phát triển hệ thống đã thay đổi rất có ý nghĩa trong nhiều năm qua. Việc cấp phát tài nguyên xác định ra cách thức chi phí, thời gian, và con người được phân bổ cho các pha khác nhau của dự án. Trong thời kỳ đầu, những người phát triển tập trung vào việc lập trình, vốn chỉ quăng 1 phần trăm thời gian và chi phí của dự án được dành cho phân tích hệ thống (xác định đặc tả). Khi các nhà chuyên môn hệ thống tin đã đi tới gần viên cảnh nghiệp vụ hay người dùng về chất lượng, thì họ cũng đi tới hiểu ra vai trò trung tâm của đặc tả. Hậu quả là các tài nguyên dự án được dịch chuyển ngay từ sớm vào chu trình dự án. Nhiều thời gian hơn được dành cho đặc tả và phân tích hệ thống, giảm dần tỉ lệ thời gian lập trình và làm giảm nhu cầu nhiều thời gian bảo hành thế. Tài liệu hiện tại gợi ý rằng quăng một phần tư thời gian và chi phí dự án nên được chỉ tiêu cho đặc tả và phân tích, mà có lẽ 50 phần trăm tài nguyên của nó được cấp phát cho thiết kế và lập trình. Việc thiết đặt và hậu cài đặt lí tưởng thì nên đòi hỏi chỉ một phần tư tài nguyên của dự án.

ĐỘ ĐO PHẦN MỀM

Độ đo phần mềm có thể đóng vai trò sống còn trong việc làm tăng chất lượng của dự án. **Độ đo phần mềm** là việc định giá khách quan về hệ thống dưới dạng cách đo định lượng. Việc sử dụng độ

đo tiếp tục cho phép bộ phận hệ thông tin và người dùng cùng đo hiệu năng của hệ thống và xác định vấn đề khi chúng xuất hiện. Thí dụ về độ đo phần mềm bao gồm một số các giao tác vốn có thể được xử lý trong một đơn vị thời gian xác định, thời gian đáp ứng trực tuyến, số các séc trả lương được in ra trong một giờ, và số các lỗi đã biết trên hàng trăm dòng mã. Không may là phần lớn các biểu lộ về chất lượng lại không dễ xác định dưới dạng độ đo. Trong những trường hợp đó người phát triển phải tìm cách đo gián tiếp. Chẳng hạn cách đo khách quan về tính dễ dùng có thể là số lần gọi điện nhờ giúp đỡ mà nhân viên hệ thông tin nhận được hàng tháng từ hệ thống của người dùng.

Để các độ đo thành công, chúng phải được thiết kế cẩn thận, hình thức và khách quan. Chúng phải đo các khía cạnh có ý nghĩa của hệ thống. Tuy nhiên cũng có lời cảnh báo: Độ đo phần mềm sẽ chỉ có hiệu lực trong việc đánh giá chất lượng của hệ thống nếu người dùng đồng ý về cách đo đó từ trước. Bên cạnh đó, các độ đo chẳng có giá trị gì chừng nào mà chúng chưa được dùng một cách nhất quán. Một vài bộ phận hệ thông tin ngày nay vẫn dùng rộng rãi các độ đo khách quan, hình thức cũ mặc dầu các khảo cứu chỉ ra chúng có thể đã được cải tiến chất lượng có ý nghĩa.

KIỂM THỬ

Việc kiểm thử ngay từ đầu, đều đặn và thấu đáo sẽ đóng góp một cách có ý nghĩa cho chất lượng hệ thống. Nói chung, kiểm thử phần mềm thường hay bị hiểu lầm. Nhiều người coi kiểm thử là cách để chứng minh tính đúng đắn của công việc mà họ đã làm. Trong thực tế, chúng ta biết rằng tất cả các phần mềm có kích cỡ đều có lẫn khuất lỗi ở đâu đó, và chúng ta phải kiểm thử để phơi bày ra những lỗi này. Bất kì động cơ nào khác cũng đều sẽ làm cho tiến trình kiểm thử thành kém thấu đáo hơn.

Kiểm thử bắt đầu tại pha thiết kế. Bởi vì chưa có việc mã hoá nào cả nên việc kiểm thử thông thường được dùng là xét duyệt -

việc duyệt xét lại tài liệu đặc tả hay thiết kế trong một nhóm nhỏ những người được lựa chọn cẩn thận dựa trên kỹ năng cần thiết cho các mục tiêu đặc biệt được kiểm thử. Một khi việc mã hoá bắt đầu, thì xét duyệt mã hoá cũng có thể được dùng để duyệt xét mã chương trình. Tuy nhiên, mã phải được kiểm thử bằng việc chạy máy tính. Khi phát hiện ra lỗi, nguồn gốc lỗi được tìm ra và khử bỏ thông qua tiến trình được gọi là **gỡ lỗi**.

Phần trước đã mô tả các giai đoạn khác nhau của kiểm thử cần để đưa hệ thống tin vào vận hành - kiểm thử chương trình, kiểm thử hệ thống và kiểm thử chấp nhận. Kiểm thử sẽ thành công chỉ nếu được lập kế hoạch đúng đắn. Bản kế hoạch phải bao gồm các trường hợp kiểm thử để cho người phát triển có thể chắc chắn rằng họ đã kiểm thử một phạm vi xấp xỉ các cái vào hợp lệ và không hợp lệ. Cái vào không hợp lệ phải được kiểm thử để chắc chắn hệ thống giải quyết lỗi một cách thích hợp. Việc kiểm thử cũng phải được may đo theo kỹ thuật được kiểm thử.

DỤNG CỤ CHẤT LƯỢNG

Cuối cùng, chất lượng hệ thống có thể được nâng cao một cách có ý nghĩa bằng việc dùng các dụng cụ chất lượng. Ngày nay, nhiều dụng cụ đã được phát triển để đề cập tới mọi khía cạnh của tiến trình phát triển hệ thống. Các nhà chuyên môn hệ thống tin đang dùng phần mềm quản lý dự án để quản lý dự án của mình. Các sản phẩm đã có để làm tư liệu cho đặc tả và thiết kế hệ thống dưới dạng văn bản và đồ hoạ. Các dụng cụ lập trình bao gồm từ điển dữ liệu, các thư viện để quản lý modul chương trình và dụng cụ thực tế tạo ra mã chương trình. Tập hợp các dụng cụ mới nhất tự động hoá cho nhiều việc chuẩn bị cho việc kiểm thử toàn diện. Công nghệ dụng cụ vẫn còn tương đối mới, và trong nhiều trường hợp giá trị của chúng vẫn còn chưa được chứng minh. Dầu sao đi chăng nữa, dụng cụ cũng có tác động có ý nghĩa lên chất lượng hệ thống và lên chi phí phát triển. Chúng ta sẽ thảo luận về các kiểu dụng cụ ở cuối mục này.

Phương pháp luận có cấu trúc

Trong những ngày đầu của lập trình, đã có vài phương pháp luận. Đặc tả người dùng được thu thập qua đối thoại không chính thức rồi được viết ra thành các lời kể khó tuân theo. Chương trình được viết trong mã phức tạp và lẫn lộn, còn được nói tới là món **mì ống** vì nó không thể đọc được tới mức luồng logic dường như là xoắn xuýt như một nồi mì ống. Hệ thống không thể thay đổi được, việc bảo trì gần như không thể được.

Để đáp lại cho những vấn đề ban đầu này, các phương pháp luận mới đã nổi lên trong những năm 1970. Những phương pháp luận này tổ hợp một phạm vi các phương pháp hay kĩ thuật mà thường được mô tả bằng thuật ngữ *có cấu trúc* và *trên xuống*. **Có cấu trúc** nói tới sự kiện là các kĩ thuật này làm dần từng bước, mỗi bước lại xây dựng dựa trên bước trước đó. **Trên xuống** nói tới một cách tiếp cận tiến dần từ mức độ cao nhất, trừu tượng nhất xuống tới mức độ chi tiết thấp nhất - từ tổng quát tới đặc thù. Chẳng hạn, mức cao nhất của mô tả trên xuống về hệ thống tài nguyên con người chỉ ra các chức năng tài nguyên con người chính, như nhân sự, phúc lợi, nhân công và cơ hội kinh tế bình đẳng. Mỗi một trong những điều này rồi lại được mở ra hay phân tách ra thành tầng tiếp. Chẳng hạn phúc lợi có thể bao gồm lương hưu, tiết kiệm nhân viên, chăm sóc sức khoẻ và bảo hiểm. Mỗi một trong những đường này lại được chia nhỏ ra, hết tầng nọ tới tầng kia, cho tới khi chất liệu tại mức thấp nhất dễ dàng được hiểu thấu và làm tư liệu. Trên xuống được dùng cho phân tích, thiết kế và lập trình.

Các phương pháp luận có cấu trúc truyền thống đều hướng xử lí chứ không hướng dữ liệu. Trong khi mô tả dữ liệu là một phần của những phương pháp này, thì các phương pháp luận hướng tiến trình hội tụ vào cách dữ liệu được biến đổi chứ không vào bản thân dữ liệu. Các phương pháp luận này phần lớn là tuyến tính - mỗi pha đều phải được hoàn thành trước khi pha tiếp có thể bắt đầu. Phương pháp luận trên xuống, có cấu trúc đã được dùng để phát triển một

số lớn các hệ thống trong hơn hai thập kỷ nay, cho nên rất nhiều hệ thống hiện nay đã được phát triển theo cách này. Mặc dầu có mối quan tâm ngày càng tăng tới các phương pháp luận khác, chúng vẫn còn là cách tiếp cận phương pháp luận thống trị ngày nay.

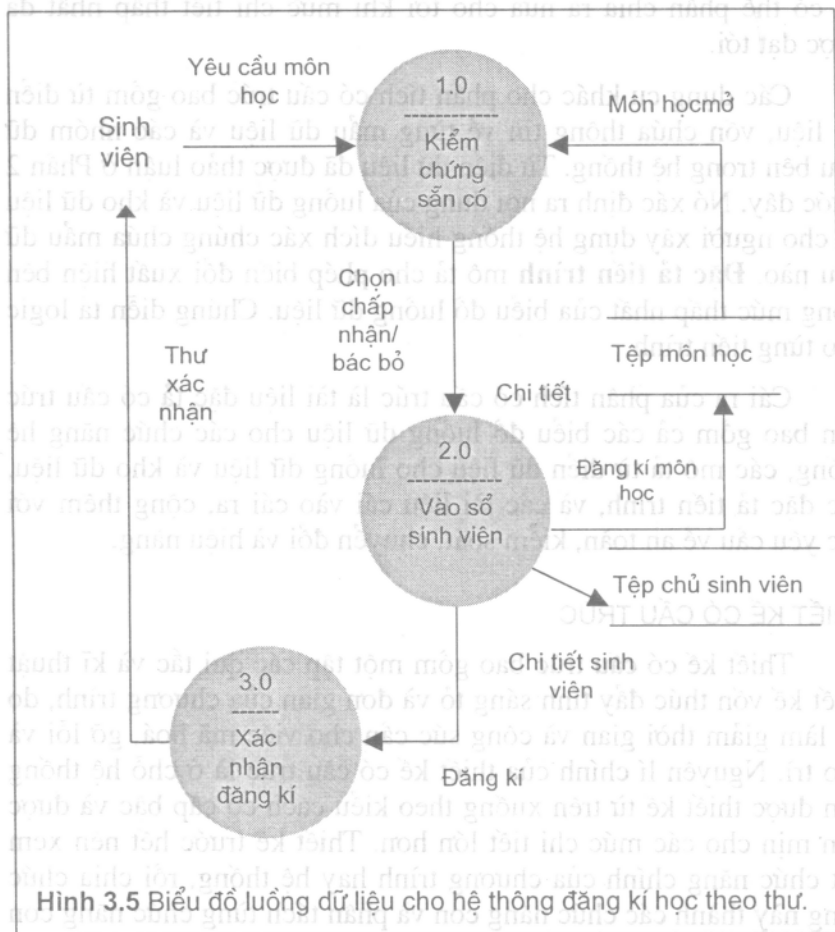
PHÂN TÍCH CÓ CẤU TRÚC

Phân tích có cấu trúc là phương pháp luận trên xuống được dùng rộng rãi để định nghĩa cái vào, xử lý và cái ra của hệ thống. Nó đưa ra một mô hình đồ họa logic về luồng thông tin, phân hoạch hệ thống thành các modul vốn biểu lộ các mức chi tiết quản lý được. Nó xác định nghiêm ngặt các tiến trình hay phép biến đổi vốn xuất hiện bên trong từng modul và các giao diện vốn tồn tại giữa chúng. Dụng cụ chính của nó là **biểu đồ luồng dữ liệu (DFD)**, một biểu diễn đồ họa của các tiến trình và giao diện cấu phần của hệ thống (luồng dữ liệu) giữa chúng.

Hình 3.5 vẽ một biểu đồ luồng dữ liệu đơn giản cho hệ thống đăng ký theo môn học ở đại học mở theo gửi thư. Các hình tròn biểu thị cho tiến trình, vốn họa lên việc biến đổi dữ liệu. Hình chữ nhật mở biểu diễn cho một thực thể ngoài, vốn là nơi phát hay nhận thông tin nằm ngoài biên giới của hệ thống đang được mô hình. Hai đường song song biểu thị cho kho dữ liệu, vốn hoặc là các kho dữ liệu thủ công hay tự động hoá. Mũi tên biểu thị cho luồng dữ liệu, vốn chỉ ra sự chuyển động giữa các tiến trình, các thực thể ngoài và các kho dữ liệu. Chúng bao giờ cũng chứa các gói dữ liệu, với tên hay nội dung của từng dữ liệu được liệt kê dưới mũi tên.

Luồng dữ liệu này chỉ ra rằng sinh viên đệ trình mẫu đăng ký có tên, số hiệu của họ và số hiệu môn học được lựa chọn mà họ muốn lấy. Trong tiến trình 1.0 hệ thống kiểm chứng rằng mỗi môn học được chọn vẫn đang mở trong tệp môn học của trường. Tệp này phân biệt các môn học vẫn đang nhận học sinh với các môn đã bị cắt bỏ hay đã kín rồi. Tiến trình 2.0 ghi danh sinh viên vào trong các môn học mà người đó đã được chấp nhận. Nó cập nhật tệp môn

học của trường theo số hiệu và tên sinh viên và tính lại kích cỡ lớp. Nếu số sinh viên đã tới số tối đa thì môn học này được dựng cờ là đóng. Tiến trình 2.0 cũng cập nhật tệp sinh viên chính theo thông tin về sinh viên mới hay thay đổi địa chỉ. Tiến trình 3.0 gửi cho mỗi sinh viên xin học một thư xác nhận đăng kí, liệt kê ra các môn học mà người đó đã được đăng kí và lưu ý việc lựa chọn môn học có thể chưa kín.



Các biểu đồ có thể được dùng để mô tả cho các tiến trình mức cao hơn cũng như các chi tiết mức thấp. Thông qua các biểu đồ luồng dữ liệu có mức độ, một tiến trình phức tạp có thể được phân chia ra thành các mức chi tiết kế tiếp. Toàn bộ hệ thống có thể được phân chia thành các hệ con với một biểu đồ luồng dữ liệu mức cao. Mỗi hệ con, đến lượt nó, lại có thể được phân chia thành các hệ con phụ với biểu đồ luồng dữ liệu mức hai, và các hệ con mức thấp hơn lại có thể phân chia ra nữa cho tới khi mức chi tiết thấp nhất đã được đạt tới.

Các dụng cụ khác cho phân tích có cấu trúc bao gồm từ điển dữ liệu, vốn chứa thông tin về từng mẫu dữ liệu và các nhóm dữ liệu bên trong hệ thống. Từ điển dữ liệu đã được thảo luận ở Phần 2 trước đây. Nó xác định ra nội dung của luồng dữ liệu và kho dữ liệu để cho người xây dựng hệ thống hiểu đích xác chúng chứa mẫu dữ liệu nào. **Đặc tả tiến trình** mô tả cho phép biến đổi xuất hiện bên trong mức thấp nhất của biểu đồ luồng dữ liệu. Chúng diễn tả logic cho từng tiến trình.

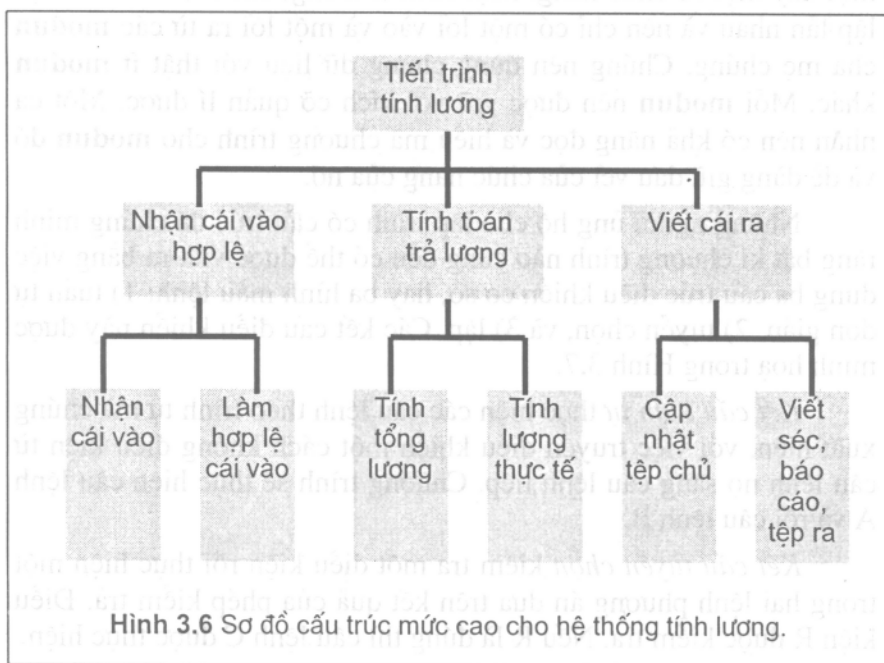
Cái ra của phân tích có cấu trúc là tài liệu đặc tả có cấu trúc vốn bao gồm cả các biểu đồ luồng dữ liệu cho các chức năng hệ thống, các mô tả từ điển dữ liệu cho luồng dữ liệu và kho dữ liệu, các đặc tả tiến trình, và các tài liệu cái vào cái ra, cộng thêm với các yêu cầu về an toàn, kiểm soát, chuyển đổi và hiệu năng.

THIẾT KẾ CÓ CẤU TRÚC

Thiết kế có cấu trúc bao gồm một tập các qui tắc và kỹ thuật thiết kế vốn thúc đẩy tính sáng tỏ và đơn giản của chương trình, do đó làm giảm thời gian và công sức cần cho việc mã hoá, gỡ lỗi và bảo trì. Nguyên lý chính của thiết kế có cấu trúc là ở chỗ hệ thống nên được thiết kế từ trên xuống theo kiểu cách có cấp bậc và được làm mịn cho các mức chi tiết lớn hơn. Thiết kế trước hết nên xem xét chức năng chính của chương trình hay hệ thống, rồi chia chức năng này thành các chức năng con và phân tách từng chức năng con

cho tới khi đạt tới mức chi tiết thấp nhất. Các modul mức thấp nhất mô tả cho việc xử lý thực tế vẫn xuất hiện. Theo cách này, mọi logic mức cao và mô hình thiết kế đều được phát triển trước khi mã chương trình chi tiết được viết ra. Nếu phân tích có cấu trúc đã được thực hiện, thì tài liệu đặc tả có cấu trúc có thể dùng làm cái vào cho tiến trình thiết kế.

Khi thiết kế được hình thành, thì nó được làm tài liệu trong sơ đồ cấu trúc. **Sơ đồ cấu trúc** là sơ đồ trên xuống, chỉ ra từng mức thiết kế, mối quan hệ của nó với các mức khác, và vị trí của nó trong cấu trúc thiết kế tổng thể. Hình 3.6 vẽ một sơ đồ cấu trúc có thể được dùng cho hệ thống tính lương. Nếu thiết kế có quá nhiều mức không xếp được vào một sơ đồ cấu trúc, thì nó có thể được chia ra thành các sơ đồ cấu trúc chi tiết hơn.



LẬP TRÌNH CÓ CẤU TRÚC

Lập trình có cấu trúc mở rộng các nguyên lý cai quản cho thiết kế có cấu trúc sang việc viết chương trình. Nó cũng dựa trên nguyên tắc **modun** hoá, đi theo phân tích và thiết kế trên xuống.

Lập trình có cấu trúc là một phương pháp tổ chức và mã hoá chương trình vốn làm đơn giản hoá các đường điều khiển để cho chương trình có thể dễ hiểu và sửa đổi. Lập trình có cấu trúc làm giảm độ phức tạp được tạo ra khi các lệnh chương trình nhảy tới nhảy lui sang các phần khác của chương trình, làm mờ tối logic và luồng chương trình.

Mỗi một trong các hộp trong sơ đồ cấu trúc biểu diễn cho một **modun** cấu phần. Nó tạo nên một đơn vị logic vốn thực hiện một hay một số chức năng. Một cách lí tưởng, các **modun** nên độc lập lẫn nhau và nên chỉ có một lối vào và một lối ra từ các **modun** cha mẹ chúng. Chúng nên dùng chung dữ liệu với thật ít **modun** khác. Mỗi **modun** nên được giữ với kích cỡ quản lí được. Một cá nhân nên có khả năng đọc và hiểu mã chương trình cho **modun** đó và dễ dàng giữ dấu vết của chức năng của nó.

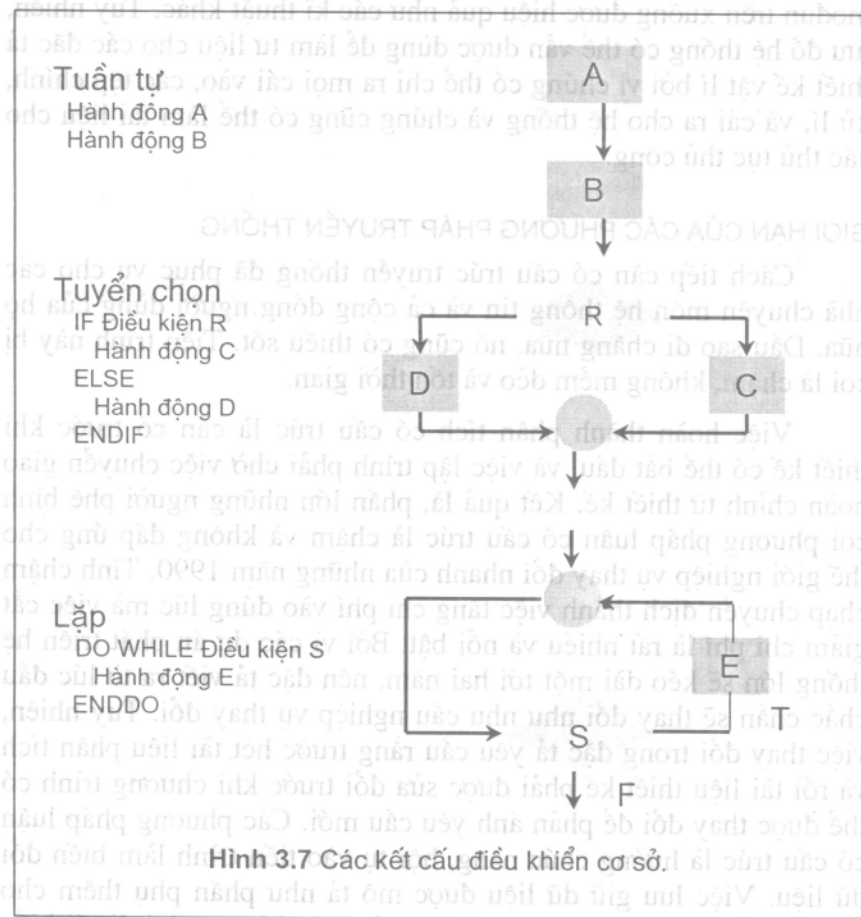
Những người ủng hộ cho lập trình có cấu trúc đã chứng minh rằng bất kì chương trình nào cũng đều có thể được viết ra bằng việc dùng ba cấu trúc điều khiển cơ sở, hay ba hình mẫu lệnh: 1) tuần tự đơn giản, 2) tuyển chọn, và 3) lặp. Các kết cấu điều khiển này được minh hoạ trong Hình 3.7.

Kết cấu tuần tự thực hiện các câu lệnh theo trình tự mà chúng xuất hiện, với việc truyền điều khiển một cách không điều kiện từ câu lệnh nọ sang câu lệnh tiếp. Chương trình sẽ thực hiện câu lệnh A và rồi câu lệnh B.

Kết cấu tuyển chọn kiểm tra một điều kiện rồi thực hiện một trong hai lệnh phương án dựa trên kết quả của phép kiểm tra. Điều kiện R được kiểm tra. Nếu R là đúng thì câu lệnh C được thực hiện.

Nếu R là sai thì câu lệnh D được thực hiện. Tiếp đó điều khiển truyền qua câu lệnh tiếp.

Kết cấu lặp lập lại một đoạn mã chừng nào mà phép kiểm tra điều kiện vẫn còn đúng. Điều kiện S được kiểm tra. Nếu S đúng, thì câu lệnh E được thực hiện và điều khiển quay về kiểm tra S. Nếu S là sai thì E bị nhảy qua và điều khiển truyền sang câu lệnh tiếp.



LƯU ĐỒ

Lưu đồ là một dụng cụ thiết kế cổ vẫn còn được dùng. Lưu đồ hệ thống làm chi tiết luồng dữ liệu trong toàn thể hệ thống tin. Lưu đồ chương trình mô tả cho các tiến trình xảy ra bên trong một chương trình riêng lẻ trong hệ thống và trình tự mà theo đó chúng phải được thực hiện. Việc lập lưu đồ không còn được khuyến cáo cho thiết kế chương trình nữa bởi vì nó không cung cấp cấu trúc modun trên xuống được hiệu quả như các kĩ thuật khác. Tuy nhiên, lưu đồ hệ thống có thể vẫn được dùng để làm tư liệu cho các đặc tả thiết kế vật lí bởi vì chúng có thể chỉ ra mọi cái vào, các tệp chính, xử lí, và cái ra cho hệ thống và chúng cũng có thể làm tài liệu cho các thủ tục thủ công.

GIỚI HẠN CỦA CÁC PHƯƠNG PHÁP TRUYỀN THỐNG

Cách tiếp cận có cấu trúc truyền thống đã phục vụ cho các nhà chuyên môn hệ thống tin và cả cộng đồng người dùng của họ nữa. Dẫu sao đi chăng nữa, nó cũng có thiếu sót. Tiến trình này bị coi là chậm, không mềm dẻo và tốn thời gian.

Việc hoàn thành phân tích có cấu trúc là cần có trước khi thiết kế có thể bắt đầu, và việc lập trình phải chờ việc chuyển giao hoàn chỉnh từ thiết kế. Kết quả là, phần lớn những người phê bình coi phương pháp luận có cấu trúc là chậm và không đáp ứng cho thế giới nghiệp vụ thay đổi nhanh của những năm 1990. Tính chậm chạp chuyển dịch thành việc tăng chi phí vào đúng lúc mà việc cắt giảm chi phí là rất nhiều và nổi bật. Bởi vì các dự án phát triển hệ thống lớn sẽ kéo dài một tới hai năm, nên đặc tả viết ra từ lúc đầu chắc chắn sẽ thay đổi như nhu cầu nghiệp vụ thay đổi. Tuy nhiên, việc thay đổi trong đặc tả yêu cầu rằng trước hết tài liệu phân tích và rồi tài liệu thiết kế phải được sửa đổi trước khi chương trình có thể được thay đổi để phản ánh yêu cầu mới. Các phương pháp luận có cấu trúc là hướng chức năng, hội tụ vào tiến trình làm biến đổi dữ liệu. Việc lưu giữ dữ liệu được mô tả như phần phụ thêm cho những tiến trình này. Vậy mà cấp quản lí nghiệp vụ lại đi tới hiểu

ra rằng phần có giá trị nhất của hệ thống tin là dữ liệu, rằng phần lớn hệ thống tin phải là hướng dữ liệu. Hệ thống hội tụ vào dữ liệu có thể nhỏ hơn và mềm dẻo hơn nhiều, làm cho chúng dễ dàng thay đổi hơn và mang tính đáp ứng nhiều hơn với việc thay đổi nhu cầu nghiệp vụ.

Bên cạnh các phương pháp luận và dụng cụ truyền thống, người xây dựng hệ thống đang quay sang phát triển hướng sự vật, kỹ nghệ phần mềm có máy tính hỗ trợ (CASE), và tái kỹ nghệ phần mềm để giúp bao quát các vấn đề chất lượng hệ thống tin.

Phát triển phần mềm hướng sự vật

Trong Phần 2 chúng ta đã giải thích rằng lập trình hướng sự vật tổ hợp dữ liệu và những thủ tục xác định vốn vận hành trên những dữ liệu đó vào trong một "sự vật". Lập trình hướng sự vật là một phần của cách tiếp cận lớn hơn tới việc phát triển hệ thống được gọi là phát triển phần mềm hướng sự vật. Phát triển phần mềm hướng sự vật khác với các phương pháp luận truyền thống ở chỗ nó dịch chuyển sự hội tụ từ việc mô hình hoá tách biệt các tiến trình nghiệp vụ và dữ liệu sang việc tổ hợp dữ liệu và thủ tục vào trong các sự vật thống nhất. Hệ thống được xem như một tuyển tập các sự vật và mối quan hệ giữa các sự vật. Các sự vật này được định nghĩa, lập trình, làm tài liệu và cất giữ để được dùng như các khối xây dựng cho những ứng dụng tương lai.

Các sự vật mô hình cho cách người dùng thực tế làm việc. Chẳng hạn, nhân viên giữ tài khoản nhận có khuyến hướng nghỉ ở mức khách hàng, giới hạn vay nợ, và hoá đơn - cùng mức mà các sự vật được xây dựng. Phân tích hướng sự vật (OOA) và thiết kế hướng sự vật (OOD) đều dựa trên các sự vật này và do vậy mô hình hoá cho thế giới thực gần gũi hơn các phương pháp trước.

ÍCH LỢI CỦA CÁCH TIẾP CẬN HƯỚNG SỰ VẬT

Bởi vì sự vật là dễ dùng lại được, nên việc phát triển phần mềm hướng sự vật trực tiếp đề cập tới vấn đề về tính dùng lại và được trông đợi làm giảm thời gian và chi phí viết phần mềm. Tất nhiên, không tổ chức nào thấy được sự tiết kiệm này từ tính tái dụng chừng nào mà nó còn chưa xây dựng lên một thư viện các sự vật để từ đó mà rút ra. Kinh nghiệm phát triển phần mềm hướng sự vật vẫn còn rất hạn chế, cho nên cũng còn quá sớm để mà đánh giá cách tiếp cận này. Dẫu sao đi chăng nữa, các khảo cứu ban đầu cũng vẫn có hứa hẹn. Kinh nghiệm đã chỉ ra rằng năng suất lập trình thu được tốt hơn 10:1 là có thể. Công ti Electronic Data System Corp. (EDS) đã khảo cứu ích lợi này bằng việc xây dựng hai hệ thống quản lí bảo trì, một dùng các kĩ thuật có cấu trúc và một dùng lập trình hướng sự vật. EDS dùng mức độ kĩ năng cân bằng cho hai tổ dự án và cho họ làm việc từ cùng một đặc tả. Họ thấy sự cải tiến năng suất là 14:1 với việc dùng lập trình hướng sự vật (IDC White Paper, 1992). Chi phí bảo trì cũng được hạ thấp bằng việc giảm nhiều thay đổi bảo trì. Chẳng hạn, khi hệ thống bưu điện Mỹ thay đổi mã zip từ năm chữ số lên chín chữ số, thì mỗi chương trình truyền thống bên trong công ti này đều phải bị thay đổi. Nếu các chương trình của công ti mà là hướng sự vật, thì người lập trình chỉ phải sửa đổi mã của sự vật, và thay đổi này sẽ được phản ánh trong mọi chương trình dùng sự vật đó.

Việc phát triển phần mềm hướng sự vật dẫn tới những thay đổi khác trong phương pháp. Một khi thư viện sự vật tồn tại, thì việc thiết kế và lập trình thường có thể bắt đầu mà không phải đợi các tài liệu phân tích. Về lí thuyết, thiết kế và lập trình cả hai sẽ bắt đầu ngay khi các yêu cầu được hoàn tất qua việc dùng cách lặp lại làm bản mẫu nhanh. Các phương pháp hướng sự vật nên làm tăng sự tham dự của người dùng bởi vì họ có thể thấy các sự vật dễ hiểu hơn và làm việc tự nhiên hơn là các dụng cụ có cấu trúc truyền thống và bởi vì việc làm bản mẫu lặp lại đặt họ vào trung tâm của thiết kế và thậm chí của lập trình.

CHƯỞNG NGẠI CHO VIỆC DÙNG CÁC KỸ THUẬT HƯỚNG SỰ VẬT

Mặc dầu nhu cầu về huấn luyện trong các kỹ thuật và các dụng cụ lập trình hướng sự vật, việc phát triển phần mềm hướng sự vật vẫn trong giai đoạn non trẻ và còn chưa được thuyết phục với hầu hết các cơ quan để chấp nhận nó. Chưa có phương pháp luận phát triển hướng sự vật nào được thống nhất phổ biến, mặc dầu đã có nhiều đề nghị. Mới đây ngôn ngữ mô hình hoá thống nhất UML đang dần trở thành một phương pháp luận hướng sự vật được đa số người chấp nhận. Tuy nhiên, nhiều công ti vẫn còn ngần ngại chưa dám thử vì phương pháp luận hướng sự vật đòi hỏi việc huấn luyện lại cho nhân viên một cách cơ bản và toàn diện, một sự thay đổi tư duy to lớn. Cấp quản lí cũng nhận biết rằng việc chuyển hoàn toàn sang phát triển hướng sự vật sẽ mất nhiều thời gian. Phần lớn các công ti đã đầu tư rất nhiều và các hệ thống có cấu trúc hiện tại mà vẫn phải bảo trì cho tới lúc họ có thể cần thay thế chúng. Cho đến lúc đó thì bộ phận hệ thống tin chắc cũng sẽ có được tri thức chuyên gia về cả các phương pháp có cấu trúc và hướng sự vật.

Kỹ nghệ phần mềm có máy tính trợ giúp (CASE)

Kỹ nghệ phần mềm có máy tính trợ giúp (CASE) - đôi khi còn được gọi là kỹ nghệ hệ thống có máy tính hỗ trợ - là việc tự động hoá các phương pháp luận từng bước cho việc phát triển phần mềm và hệ thống để làm giảm khối lượng công việc lặp lại mà người phát triển cần làm. Việc chấp nhận nó có thể giải phóng cho người phát triển để làm những nhiệm vụ giải quyết vấn đề sáng tạo hơn. Các dụng cụ CASE cũng tạo điều kiện thuận lợi cho việc tạo ra tài liệu rõ ràng và sự điều phối các nỗ lực của tổ phát triển. Các thành viên tổ có thể chia sẻ công việc dễ dàng hơn bằng việc truy nhập vào các tệp của nhau để xét duyệt hay sửa đổi điều đã được làm. Hệ thống được phát triển bằng CASE đã tỏ ra là tin cậy hơn và ít đòi hỏi sửa chữa hơn. Nhiều dụng cụ CASE là dựa trên máy vi tính, với những khả năng đồ hoạ mạnh.

Bảng 3.7 Các phần tử khoá của CASE

Dụng cụ vẽ biểu đồ: Các dụng cụ đồ hoạ để vẽ các kí hiệu cho biểu đồ luồng dữ liệu, sơ đồ cấu trúc, biểu đồ thực thể-quan hệ, hay các kiểu biểu đồ khác liên kết với một phương pháp luận đặc biệt.

Bộ kiểm chứng cú pháp: Kiểm chứng độ chính xác và tính đầy đủ của thông tin được đưa vào trong hệ thống tuân thủ với các qui tắc của phương pháp luận có cấu trúc đặc biệt.

Dụng cụ làm bản mẫu: Các bộ sinh màn hình, báo cáo và đơn chọn cho phép người phân tích vẽ ra các màn hình mong muốn và bố trí báo cáo hay đường dẫn đơn chọn qua hệ thống mà không cần đặc tả định dạng phức tạp hay lập trình.

Kho thông tin: Một cơ sở dữ liệu trung tâm vốn phục vụ như một cơ chế để cất giữ mọi kiểu tài sản phần mềm - các bố trí màn hình và báo cáo, các biểu đồ, định nghĩa dữ liệu, mã chương trình, lịch dự án và các tài liệu khác. Kho này phối hợp, tích hợp và chuẩn hoá các mảnh mẫu thông tin khác nhau để cho chúng có thể dễ dàng được truy nhập tới, được dùng chung cho người phân tích và được dùng lại cho công việc phần mềm tương lai.

Bộ sinh mã: Các bộ sinh này có thể sinh ra các modul mã thực hiện được từ các đặc tả cấp cao. Một số dụng cụ CASE dùng biểu tượng để chỉ ra các chức năng chương trình khác nhau và dịch các biểu tượng đó thành chương trình.

Phương pháp luận phát triển: Một số sản phẩm CASE chứa danh sách kiểm tra hay lời thuật chỉ tiết về toàn bộ phương pháp luận phát triển vốn giúp điều phối và kiểm soát toàn bộ dự án phát triển hệ thống.

Dụng cụ quản lí dự án: Một số dụng cụ CASE tích hợp các cấu phần của chúng với các dụng cụ riêng lẻ phổ biến cho việc lập lịch dự án và ước lượng tài nguyên, trong khi số khác lại tổ hợp phần mềm quản lí dự án vào trong bộ dụng cụ CASE.

Các dụng cụ CASE cung cấp các tiện nghi đồ hoạ tự động hoá cho việc tạo ra các sơ đồ và biểu đồ, các bộ sinh màn hình và báo cáo, các từ điển dữ liệu, các tiện nghi báo cáo tăng cường, các

dụng cụ phân tích và kiểm tra, bộ sinh mã, và bộ sinh tài liệu. Phần lớn các dụng cụ CASE đều dựa trên một hay nhiều phương pháp luận có cấu trúc phổ biến. Một số đang bắt đầu hỗ trợ cho phát triển hướng sự vật. Nói chung, các dụng cụ CASE cố gắng làm tăng năng suất và chất lượng bằng việc làm những điều sau:

- Bắt buộc dùng phương pháp luận phát triển chuẩn và kỉ luật thiết kế
- Cải tiến trao đổi giữa người dùng và chuyên viên kĩ thuật
- Tổ chức và làm liên quan các cấu phần thiết kế và cung cấp việc truy nhập nhanh chóng vào chúng qua kho thiết kế.
- Tự động hoá các phần buồn tẻ và hay sinh lỗi của phân tích và thiết kế.
- Tự động hoá việc kiểm thử và kiểm soát việc giới thiệu hệ thống.

Các phân tử khoá của CASE được mô tả trong Bảng 3.7.

THÍ DỤ VỀ CÁC DỤNG CỤ CASE

Dụng cụ CASE đã được phân loại dưới dạng liệu chúng có hỗ trợ cho các hoạt động ở đầu trước hay đầu sau của tiến trình phát triển hệ thống. Các dụng cụ CASE đầu trước tập trung vào việc nắm bắt thông tin phân tích và thiết kế trong những giai đoạn sớm của việc phát triển hệ thống, trong khi các dụng cụ CASE đầu sau lại đề cập tới việc mã hoá, kiểm thử, và hoạt động bảo trì. Các dụng cụ đầu sau giúp chuyển đổi các đặc tả tự động thành mã chương trình. Các sản phẩm CASE như Andersen Consulting's Foundation và KnowledgeWare's Application Development Workbench biểu thị cho nhiều dụng cụ tích hợp đầy đủ mà bắt đầu hỗ trợ cho toàn bộ tiến trình phát triển hệ thống, kể cả quản lí dự án và sinh tự động mã chương trình.

Người phân tích dùng dụng cụ CASE để giúp nắm bắt các yêu cầu và đặc tả bằng việc lưu giữ thông tin trong cơ sở dữ liệu

CASE, tại đó có thể dễ dàng dò tìm và cải biên chúng. Các dụng cụ CASE tạo điều kiện thuận lợi cho công việc thiết kế và phân tích phía trước, cho nên có ít lỗi hơn phải sửa về sau. Trình soạn thảo văn bản và đồ hoạ CASE giúp người phân tích tạo ra các biểu đồ đúng đắn về mặt kĩ thuật, các mô tả tiến trình, và các mục từ điển dữ liệu.

Các dụng cụ CASE tự động gắn các phần tử dữ liệu với các tiến trình nơi chúng được dùng. Nếu một biểu đồ luồng dữ liệu bị thay đổi từ tiến trình này sang tiến trình khác, thì các phần tử trong từ điển dữ liệu cũng sẽ được thay đổi một cách tự động để phản ánh sự thay đổi trong biểu đồ này. Các dụng cụ CASE cũng chứa các tính năng để làm hợp lệ các biểu đồ thiết kế và đặc tả. Một số bộ dụng cụ chứa các tính năng làm bản mẫu như bộ vẽ màn hình và báo cáo, vốn cho phép người phân tích vẽ ra các định dạng màn hình hay báo cáo cho người dùng xét duyệt. Dụng cụ CASE vậy hỗ trợ cho việc thiết kế lặp bằng việc tự động hoá việc duyệt lại và thay đổi và cung cấp tiện nghi làm bản mẫu.

Một phần tử trung tâm trong dụng cụ CASE là kho thông tin, vốn cất giữ mọi thông tin được người phân tích định nghĩa trong dự án. Kho này bao gồm các biểu đồ luồng dữ liệu, các sơ đồ cấu trúc, các biểu đồ thực thể-quan hệ, các định nghĩa dữ liệu, các đặc tả tiến trình, các định dạng màn hình và báo cáo, các chú thích và bình luận, và kết quả kiểm thử. Cơ sở dữ liệu CASE có thể được nhiều thành viên của tổ dự án dùng chung và có chứa các tính năng để hạn chế những thay đổi chỉ do người phân tích đặc biệt duy nhất thực hiện.

THÁCH THỨC CỦA VIỆC DÙNG CASE

Để được dùng một cách có hiệu quả, dụng cụ CASE đòi hỏi nhiều kỉ luật tổ chức hơn cách tiếp cận thủ công. Mọi thành viên của dự án phát triển phải tôn trọng triệt để một tập chung các qui ước đặt tên, các chuẩn, và phương pháp luận phát triển. Không có

kỉ luật này, người phân tích và thiết kế sẽ đơn giản cố gắng tổ hợp dụng cụ CASE vào các tiến trình phát triển cũ, không thích hợp. Dụng cụ CASE tốt nhất áp đặt các phương pháp và chuẩn chung, điều vốn có thể làm nản lòng việc dùng chúng trong những tình huống thiếu kỉ luật của tổ chức.

CASE không phải là liều thuốc bách bệnh thần kì. Nó không có khả năng làm cho hệ thống được thiết kế một cách tự động hay không đảm bảo rằng các yêu cầu nghiệp vụ là được đáp ứng. Người thiết kế hệ thống vẫn cần phải hiểu nhu cầu nghiệp vụ của tổ chức mình là gì và cách thức nghiệp vụ vận hành. Phân tích và thiết kế hệ thống vẫn còn phụ thuộc vào kĩ năng phân tích của người phân tích/thiết kế. Nếu tổ chức thiếu phương pháp luận, thì dụng cụ CASE có thể được dùng để tự động hoá việc thực hành tạp nham và thường không tương hợp hơn là tích hợp hay chuẩn hoá cách tiếp cận phát triển hệ thống của tổ chức. Bảng 3.8 mô tả sức mạnh và giới hạn của dụng cụ CASE.

Bảng 3.8 ĐIỀU CASE CÓ THỂ VÀ KHÔNG THỂ LÀM ĐƯỢC

Dụng cụ CASE có thể:

1. Tự động hoá nhiều công việc thủ công của việc phát triển hệ thống
2. Thúc đẩy việc chuẩn hoá dựa trên một phương pháp luận
3. Thúc đẩy sự nhất quán và phối hợp lớn hơn trong một dự án phát triển.
4. Sinh ra một phần lớn tài liệu cho hệ thống, như biểu đồ luồng dữ liệu, mô hình dữ liệu, sơ đồ cấu trúc, hay các đặc tả khác.

Dụng cụ CASE không thể:

1. Tự động cung cấp hệ thống vận hành, có liên quan. Nó chỉ dễ dàng tạo ra một hệ thống tồi như tạo hệ thống tốt bằng việc dùng dụng cụ CASE.
2. Giao tiếp dễ dàng với cơ sở dữ liệu và ngôn ngữ thế hệ bốn
3. Tự động hoá buộc người phân tích phải dùng một phương pháp luận đã mô tả trước hay tạo ra phương pháp luận khi nó không tồn tại.
4. Biến đổi triệt để tiến trình phân tích và thiết kế hệ thống.

Tái kĩ nghệ phần mềm

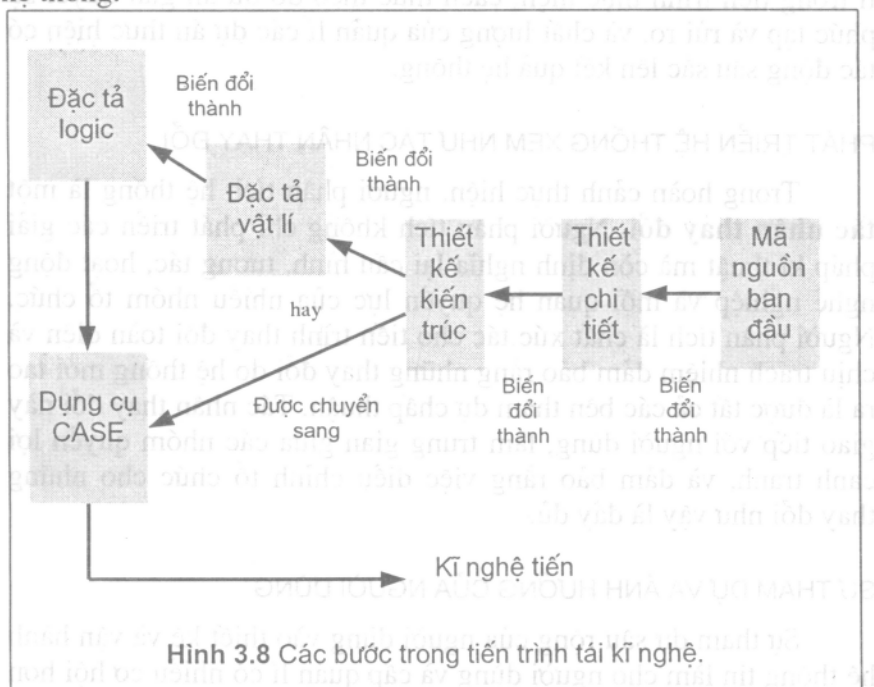
Tái kĩ nghệ phần mềm là một phương pháp luận để cập tới vấn đề phần mềm lão hoá. Rất nhiều phần mềm mà các tổ chức đang dùng đã được viết ra nhưng chẳng được lợi của phương pháp luận như phân tích, thiết kế và lập trình có cấu trúc. Phần mềm như vậy khó bảo trì hay cập nhật. Tuy nhiên, thông thường phần mềm sẽ phục vụ cho tổ chức đủ tốt để tiếp tục được dùng nữa, chỉ nếu nó có thể dễ bảo trì hơn. Mục đích của tái kĩ nghệ là cứu lấy những phần mềm như thế bằng việc nâng cấp nó sao cho người dùng có thể tránh được dự án thay thế tốn kém và lâu dài. Về bản chất, người phát triển dùng việc tái kĩ nghệ để trích rút ra phần thông minh trong thiết kế và lập trình từ hệ thống hiện có, từ đó tạo ra hệ thống mới mà không bắt đầu từ đầu. Tái kĩ nghệ bao gồm ba bước: kĩ nghệ ngược, sửa lại thiết kế và đặc tả chương trình, và kĩ nghệ tiến.

Kĩ nghệ ngược đòi hỏi trích rút các đặc tả nghiệp vụ nền tảng từ hệ thống hiện có. Các hệ thống cũ hơn, không cấu trúc, không có tài liệu thích hợp về tiến trình nghiệp vụ, thiết kế hệ thống hay chương trình. Công cụ kĩ nghệ ngược đọc và phân tích mã, tệp và mô tả cơ sở dữ liệu hiện có của chương trình và tạo ra tài liệu có cấu trúc về hệ thống. Cái ra sẽ chỉ ra các cấu phần mức thiết kế, như các thực thể, thuộc tính, và tiến trình. Với tài liệu có cấu trúc để làm việc, tổ dự án có thể sửa lại thiết kế và đặc tả để đáp ứng cho yêu cầu nghiệp vụ hiện thời. Trong bước cuối cùng, **kĩ nghệ tiến**, các đặc tả đã được sửa lại được dùng để sinh ra mã mới, có cấu trúc cho một hệ thống có cấu trúc và bây giờ bảo trì được. Trong Hình 3.8 bạn có thể theo tiến trình tái kĩ nghệ. Lưu ý rằng dụng cụ CASE có thể được dùng trong bước kĩ nghệ tiến.

3.4.3 QUẢN LÝ VIỆC THỰC HIỆN HỆ THỐNG

Bên cạnh các nhân tố kĩ thuật liên quan tới chất lượng phần mềm, các vấn đề hệ thống tin có thể còn bị gây ra bởi cách thức

quản lý sự thay đổi tổ chức xoay quanh hệ thống tin mới. Như nhiều nhà nghiên cứu hệ thống tin quản lý đã chỉ ra, việc đưa vào hay thay đổi một hệ thống tin có tác động mạnh về mặt tổ chức và hành vi. Nó làm biến đổi cách thức các cá nhân và nhóm khác nhau thực hiện và tương tác. Những thay đổi trong cách thông tin được định nghĩa, truy nhập và sử dụng để quản lý tài nguyên của tổ chức thường dẫn tới sự phân phối mới về quyền lực. Sự thay đổi tổ chức bên trong này làm phát sinh kháng cự và chống đối và có thể đưa tới việc mất đi của một hệ thống tốt khác. Một đặc trưng quan trọng của phần lớn các hệ thống tin là ở chỗ các cá nhân được đề nghị hay được yêu cầu thay đổi hành vi của mình để thực hiện chức năng hệ thống.



Nhiều khảo cứu đã thấy rằng trong các tổ chức với môi trường và tính năng định chế tương tự, thì cùng đổi mới sẽ thành

công ở tổ chức này nhưng lại thất bại ở tổ chức khác. Tại sao? Một cách tiếp cận để giải thích sự khác biệt này tập trung vào những hình mẫu thực hiện khác nhau.

Các nguyên nhân của thực hiện thành công hay thất bại

Trong hoàn cảnh quản lí thay đổi, việc **thực hiện** nói tới tất cả các hoạt động tổ chức cùng làm việc hướng tới sự chấp thuận, quản lí và qui trình hoá một đổi mới như hệ thống tin mới. Các nhân tố quản lí và tổ chức, như vai trò của người dùng và cấp quản lí trong tiến trình thực hiện, cách thức theo đó dự án giải quyết sự phức tạp và rủi ro, và chất lượng của quản lí các dự án thực hiện có tác động sâu sắc lên kết quả hệ thống.

PHÁT TRIỂN HỆ THỐNG XEM NHƯ TÁC NHÂN THAY ĐỔI

Trong hoàn cảnh thực hiện, người phân tích hệ thống là một **tác nhân thay đổi**. Người phân tích không chỉ phát triển các giải pháp kĩ thuật mà còn định nghĩa lại cấu hình, tương tác, hoạt động nghề nghiệp và mối quan hệ quyền lực của nhiều nhóm tổ chức. Người phân tích là chất xúc tác cho tiến trình thay đổi toàn diện và chịu trách nhiệm đảm bảo rằng những thay đổi do hệ thống mới tạo ra là được tất cả các bên tham dự chấp thuận. Tác nhân thay đổi này giao tiếp với người dùng, làm trung gian giữa các nhóm quyền lợi cạnh tranh, và đảm bảo rằng việc điều chỉnh tổ chức cho những thay đổi như vậy là đầy đủ.

SỰ THAM DỰ VÀ ẢNH HƯỞNG CỦA NGƯỜI DÙNG

Sự tham dự sâu rộng của người dùng vào thiết kế và vận hành hệ thống tin làm cho người dùng và cấp quản lí có nhiều cơ hội hơn để lên khuôn hệ thống tương ứng với ưu tiên của họ và các yêu cầu nghiệp vụ. Bên cạnh đó, họ rất có thể phản ứng tích cực với hệ

thống được hoàn thành bởi vì họ đã là thành viên tích cực trong tiến trình thay đổi.

Bảng 3.9 LỖ HỔNG TRAO ĐỔI NGƯỜI DÙNG-NGƯỜI THIẾT KẾ

Quan tâm của người dùng	Quan tâm của người thiết kế
Hệ thống có chuyển giao thông tin tôi cần cho công việc hay không?	Tệp chủ sẽ chiếm bao nhiêu không gian đĩa?
Tôi có thể truy nhập dữ liệu nhanh thế nào?	Để thực hiện chức năng này cần bao nhiêu dòng mã chương trình?
Tôi có thể dò tìm dữ liệu dễ dàng đến đâu?	Chúng ta có thể cắt giảm thời gian CPU khi cho chạy hệ thống không?
Tôi cần sự hỗ trợ của thư kí đến đâu để đưa dữ liệu vào hệ thống?	Cách cất giữ mẫu dữ liệu này hiệu quả nhất là gì?
Việc vận hành hệ thống khớp với lịch làm việc hàng ngày của tôi thế nào?	Chúng ta nên dùng hệ quản trị cơ sở dữ liệu nào?

Vấn đề trao đổi giữa người dùng cuối và người thiết kế là lí do chính tại sao các yêu cầu của người dùng lại không được tổ hợp đúng vào trong hệ thống tin và tại sao người dùng lại bị gạt ra ngoài tiến trình thực hiện. Người dùng và các chuyên viên hệ thống tin có khuynh hướng mang kiến thức kinh nghiệm, mối quan tâm và ưu tiên khác nhau, và thường theo đuổi những mục tiêu khác nhau. Điều này thường được nói tới là **lỗ hổng trao đổi người dùng-người thiết kế**. Sự khác biệt này được biểu lộ trong sự trung thành với tổ chức khác nhau, khác nhau trong cách tiếp cận tới việc giải quyết vấn đề, và khác nhau về vốn từ. Chẳng hạn các chuyên viên hệ thống tin thường có khuynh hướng máy móc hay kĩ thuật cao đối với việc giải quyết vấn đề. Họ tìm các giải pháp kĩ thuật thanh nhả và tinh vi mà trong đó hiệu quả phân cứng và phần mềm được tối ưu nhưng lại ảnh hưởng tới tính dễ dùng hay tính hiệu quả tổ chức. Mặt khác, người dùng lại ưa chuộng hệ thống có thiên hướng giải quyết các vấn đề nghiệp vụ hay tạo điều kiện cho các nhiệm vụ

tổ chức. Thường khuynh hướng của cả hai nhóm này lạc lõng tới mức dường như họ nổi tiếng khác nhau. Những khó khăn này được minh họa trong Bảng 3.9, vốn mô tả cho mỗi quan tâm điển hình của người dùng cuối và các chuyên viên kỹ thuật (người thiết kế hệ thông tin) đối với việc phát triển hệ thông tin mới.

HỖ TRỢ CỦA CẤP QUẢN LÝ

Nếu dự án hệ thông tin có sự hậu thuẫn và chấp thuận của cấp quản lý tại các mức khác nhau, thì rất có thể nó được cảm nhận tích cực từ cả người dùng và nhân viên dịch vụ thông tin kỹ thuật. Cả hai nhóm này sẽ cảm thấy rằng sự tham gia của họ vào tiến trình phát triển sẽ nhận được sự chú ý ở mức cao hơn, được ưu tiên và thưởng. Việc hậu thuẫn của cấp quản lý cũng đảm bảo rằng dự án hệ thống sẽ nhận được đủ ngân quỹ và tài nguyên để thành công. Hơn nữa, tất cả những thay đổi trong thói quen và thủ tục làm việc và việc tái tổ chức liên kết với hệ thống mới phụ thuộc sự hậu thuẫn của cấp quản lý để được tuân theo có hiệu lực.

MỨC ĐỘ PHỨC TẠP VÀ RỦI RO

Các hệ thống khác nhau rất lớn về kích cỡ, phạm vi, mức độ phức tạp và các cấu phần tổ chức và kỹ thuật. Một số dự án phát triển hệ thống rất có thể thất bại bởi vì chúng mang một mức độ rủi ro cao hơn số khác. Các nhà nghiên cứu đã xác định ra ba chiều hướng chủ chốt ảnh hưởng tới mức độ rủi ro dự án: kích cỡ dự án, cấu trúc dự án, và kinh nghiệm với kỹ thuật.

Kích cỡ dự án. Dự án càng lớn - như được chỉ ra theo số tiền chi ra, số lượng nhân viên thực hiện, thời gian cấp cho việc thực hiện, và số đơn vị tổ chức bị ảnh hưởng - thì rủi ro càng lớn.

Cấu trúc dự án. Các dự án càng có cấu trúc cao thì chạy càng ít rủi ro hơn những dự án có yêu cầu tương đối không xác định, hay thay đổi, và thường xuyên thay đổi; khi các yêu cầu là rõ ràng và trực tiếp, thì cái ra và tiến trình có thể dễ dàng được xác định.

Người dùng trong các dự án có cấu trúc cao có khuynh hướng biết đích xác họ muốn gì và điều gì hệ thống nên làm; có khả năng rất thấp cho việc họ đổi ý kiến.

Kinh nghiệm với công nghệ. Rủi ro của dự án sẽ dâng lên nếu tổ dự án và nhân viên hệ thống tin không quen thuộc với phần cứng, phần mềm hệ thống, phần mềm ứng dụng, hay hệ quản trị cơ sở dữ liệu được đề nghị cho dự án.

Ba chiều này của rủi ro dự án sẽ hiện diện trong các tổ hợp khác nhau cho từng nỗ lực thực hiện. Mức độ rủi ro càng cao thì càng có nhiều khả năng nỗ lực thực hiện sẽ thất bại.

QUẢN LÝ TIẾN TRÌNH THỰC HIỆN

Xung khắc và không chắc chắn cố hữu trong bất kì nỗ lực thực hiện nào sẽ được khuếch đại lên khi một dự án thực hiện bị quản lí và tổ chức tồi. Dưới việc quản lí tồi, các yếu tố cơ sở của thành công có thể bị bỏ qua. Việc huấn luyện để đảm bảo rằng người dùng cuối là thoải mái với hệ thống mới và hiểu hoàn toàn việc sử dụng tiềm năng của nó thì thường bị hi sinh đi, một phần bởi vì ngân sách đã bị căng thẳng vào lúc cuối của dự án. Dự án phát triển hệ thống mà không được quản lí đúng thì rất có thể phải chịu việc chi tiêu quá mức, trượt thời gian chính, và hiệu năng kĩ thuật tụt xuống thấp hơn mức đã ước lượng.

Các dự án bị quản lí tồi như thế nào? Về trung bình, các dự án khu vực tư nhân đều bị ước lượng sai đến một nửa về ngân sách và thời gian cần chuyển giao hệ thống đầy đủ như đã hứa hẹn trong kế hoạch hệ thống. Một số rất lớn các dự án được chuyển giao mà thiếu chức năng (hứa sẽ chuyển giao trong phiên bản sau). Các dự án của chính phủ cũng phải chịu cùng mức độ thất bại, thậm chí còn tệ hơn.

Cải tiến việc quản lí thực hiện

Không phải tất cả các khía cạnh của tiến trình thực hiện là đều có thể dễ dàng được kiểm soát hay lập kế hoạch. Tuy nhiên, những cơ hội cho việc thành công hệ thống có thể được tăng lên bởi việc dự kiến các vấn đề thực hiện tiềm năng và áp dụng các chiến lược sửa chữa thích hợp. Các chiến lược cũng phải được phát minh ra để đảm bảo rằng người dùng giữ một vai trò thích hợp trong toàn bộ thời kì thực hiện và cho việc quản lí tiến trình thay đổi tổ chức. Nhiều phương pháp luận quản lí dự án, thu thập yêu cầu, và lập kế hoạch đã được phát triển cho những loại vấn đề đặc biệt.

TĂNG SỰ THAM DỰ CỦA NGƯỜI DÙNG

Mức độ tham dự của người dùng nên thay đổi, tùy theo cả phương pháp luận phát triển được dùng và mức độ rủi ro của dự án. Các dụng cụ để đưa người dùng tham dự, **dụng cụ tích hợp ngoài**, bao gồm những cách sau để móc nối công việc của tổ thực hiện với người dùng ở mọi mức độ của tổ chức:

- Người dùng có thể được chọn làm người lãnh đạo dự án hay như sếp phó của tổ dự án.
- Ban chỉ đạo của người dùng có thể được tạo ra để đánh giá thiết kế hệ thống và làm việc quản lí toàn diện dự án.
- Người dùng có thể trở thành thành viên tích cực của tổ dự án.
- Dự án có thể yêu cầu việc xét duyệt chính thức và chấp thuận đặc tả của người dùng.
- Biên bản của tất cả các cuộc họp thiết kế chủ chốt có thể phân phối rộng rãi cho người dùng.
- Người dùng có thể chuẩn bị báo cáo trạng thái cho cấp quản lí cao hơn.
- Người dùng có thể được trao trách nhiệm huấn luyện và thiết đặt.

- Người dùng có thể chịu trách nhiệm kiểm soát thay đổi, phản hồi tất cả những thay đổi vô nghĩa với hệ thống một khi đặc tả thiết kế cuối cùng đã được hoàn tất.

VƯỢT QUA SỰ CHỐNG ĐỐI CỦA NGƯỜI DÙNG

Việc phát triển hệ thống không phải là một tiến trình hoàn toàn trôi chảy. Người dùng chỉ đạo các hoạt động thiết kế thường dùng vị trí của mình để đẩy mạnh quyền lợi riêng tư và để thu được quyền lực hơn là thúc đẩy các mục tiêu tổ chức. Hơn nữa, việc tham gia vào các hoạt động thực hiện có thể không đủ để vượt qua vấn đề về sự chống đối của người dùng. Tiến trình thực hiện yêu cầu thay đổi tổ chức. Sự thay đổi như vậy có thể bị chống đối lại bởi vì người sử dụng khác nhau có thể bị hệ thống ảnh hưởng tới theo những cách khác nhau. Trong khi một số người có thể đón chào hệ thống mới bởi vì nó đem lại những thay đổi họ cảm nhận như là có lợi cho mình, thì những người khác có thể lại chống đối những thay đổi này vì họ tin sự dịch chuyển đó có hại cho quyền lợi của họ.

Nếu việc dùng hệ thống là tự nguyện, thì người dùng có thể chọn việc tránh né nó; nếu việc dùng là bắt buộc, thì chống đối sẽ xảy ra dưới dạng tỉ lệ lỗi tăng lên, sự chia rẽ, thay người và thậm chí phá hoại. Do đó, chiến lược thực hiện phải không chỉ khuyến khích sự tham dự và tham gia của người dùng; nó cũng còn đề cập tới vấn đề chống lại thực hiện. **Việc chống lại thực hiện** là chiến lược thận trọng để cản trở việc thực hiện hệ thống tin hay một đổi mới trong tổ chức.

Các chiến lược vượt qua sự chống đối của người dùng bao gồm sự tham dự của người dùng (để kêu gọi ra cam kết cũng như cải tiến thiết kế), đào tạo người dùng (huấn luyện), bố buộc quản lý (chỉ thị, chính sách), và khuyến khích người dùng. Sự chống đối của người dùng có thể được đề cập tới qua những thay đổi với hệ thống mới, như nhân tố con người được cải thiện (giao diện người dùng/hệ thống) và sửa đổi gói phần mềm để tuân thủ với các thủ

tục của tổ chức. Cuối cùng, người dùng sẽ hợp tác nhiều hơn nếu các vấn đề của tổ chức cũng được giải quyết trước khi đưa hệ thống mới vào.

QUẢN LÝ ĐỘ PHỨC TẠP KỸ THUẬT

Các dự án với *mức độ công nghệ cao* được lợi từ các **dụng cụ tích hợp nội bộ**. Sự thành công của các dự án như vậy tùy thuộc vào việc độ phức tạp kỹ thuật có thể được quản lý tốt đến đâu. Người lãnh đạo dự án cần cả kinh nghiệm kỹ thuật và quản lý cực kỳ nhiều. Họ phải có khả năng thấy trước vấn đề và phát triển mối quan hệ làm việc êm thấm trong tổ thiên về kỹ thuật.

- Các thành viên tổ nên là người có nhiều kinh nghiệm.
- Tổ nên ở dưới sự lãnh đạo của một người quản lý giàu kinh nghiệm về kỹ thuật và quản lý dự án.
- Các cuộc họp tổ nên diễn ra thường xuyên, với việc phân phối thường lệ biên bản cuộc họp liên quan tới những quyết định thiết kế chủ chốt.
- Tổ nên duy trì đều đặn các cuộc xét duyệt trạng thái kỹ thuật.
- Số lớn nhân viên của tổ nên có lịch sử quan hệ làm việc tốt lẫn nhau.
- Các thành viên tổ nên tham gia vào việc đặt mục tiêu và thiết lập thời hạn đích.
- Kỹ năng kỹ thuật tinh tuý hay tri thức chuyên gia mà không có sẵn bên trong tổ chức thì nên được tìm kiếm ở bên ngoài tổ chức.

DỤNG CỤ LẬP KẾ HOẠCH VÀ KIỂM SOÁT CHÍNH THỨC

Các dự án sẽ được lợi từ việc dùng thích hợp các **dụng cụ lập kế hoạch chính thức** và **dụng cụ kiểm soát chính thức**. Với các kỹ thuật quản lý dự án như sơ đồ PERT (kỹ thuật đánh giá và xét duyệt chương trình) và sơ đồ Gantt, có thể phát triển một kế hoạch

chi tiết. (PERT liệt kê ra các hoạt động đặc thù vốn tạo nên dự án, thời hạn của chúng, và các hoạt động phải được hoàn tất trước khi một hoạt động đặc thù có thể bắt đầu. Sơ đồ Gantt biểu diễn trực quan trình tự và thời gian cho các nhiệm vụ khác nhau trong một dự án phát triển, cũng như các yêu cầu tài nguyên của chúng.) Các nhiệm vụ có thể được định nghĩa và tài nguyên được lập ngân sách. Các kĩ thuật quản lí dự án này có thể giúp cho người quản lí xác định chỗ thắt cổ chai và xác định tác động của vấn đề sẽ có vào thời điểm hoàn thành dự án. Tùy theo phương pháp luận phát triển được dùng và kích cỡ cùng rủi ro của dự án mà các kĩ thuật quản lí dự án sau đây có thể được chấp thuận:

- Chia pha theo cột mốc có thể được chọn.
- Đặc tả có thể được phát triển từ nghiên cứu khả thi.
- Chuẩn đặc tả có thể được thiết lập.
- Các tiến trình cho việc chấp thuận dự án có thể được phát triển.

Các kĩ thuật kiểm soát chuẩn có thể được dùng để vẽ sơ đồ tiến độ dự án so với ngân sách và hạn thực hiện, để cho sai lệch có thể được phát hiện ra và tổ thực hiện có thể điều chỉnh để đáp ứng lịch gốc của mình.

- Kì luật có thể được duy trì để kiểm soát hay chốt lại thiết kế.
- Các báo cáo thường kì so với kế hoạch sẽ chỉ ra phạm vi của tiến độ.
- Độ lệch so với kế hoạch có thể được phát hiện ra.

Việc thực hiện có thể được cải tiến thêm nữa bằng việc điều chỉnh chiến lược quản lí dự án theo mức độ rủi ro cố hữu trong từng dự án. Để kiểm soát rủi ro, dự án với *cấu trúc tương đối ít* có thể bao gồm người sử dụng hoàn toàn vào tất cả các giai đoạn, trong khi dự án chính thức hơn có thể cần điều chỉnh việc tham dự của người dùng tương ứng theo pha dự án. Có thể có những tình huống mà trong đó sự tham gia của người dùng là không thích hợp. Chẳng

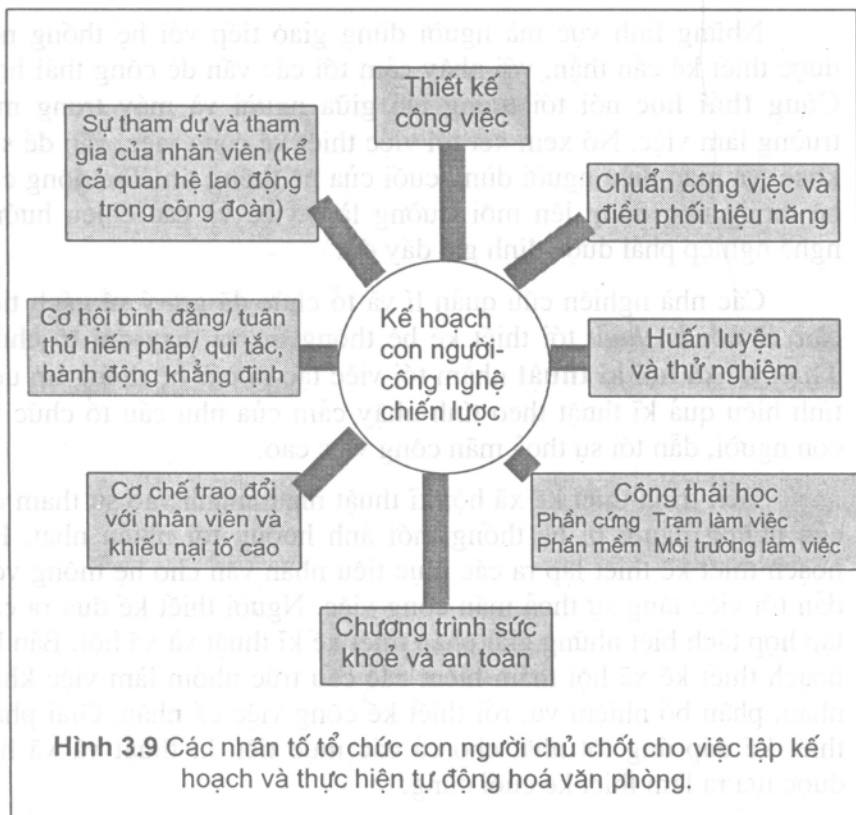
hạn, một số người dùng có thể phản ứng tiêu cực với thiết kế mới mặc dầu ích lợi toàn diện là cân nặng hơn nhược điểm của nó. Một số cá nhân có thể chống lại việc mất quyền lực xem như kết quả của quyết định thiết kế. Trong trường hợp này, việc tham dự vào thiết kế thực tế có thể làm trầm trọng thêm sự oán giận và chống đối.

Các dự án dùng công nghệ mới phức tạp thì bị rủi ro hơn và đòi hỏi sự nhấn mạnh hơn vào các dụng cụ tích hợp nội bộ. Các dự án lớn có thể làm giảm rủi ro bằng việc dùng nhiều các dụng cụ lập kế hoạch và kiểm soát.

Thiết kế cho tổ chức

Toàn bộ tiến trình phát triển hệ thống có thể được xem như việc thay đổi tổ chức có kế hoạch, vì mục đích của hệ thống mới là để cải tiến hiệu năng của tổ chức. Tiến trình phát triển phải đề cập tường minh tới cách thức mà theo đó tổ chức sẽ thay đổi khi hệ thống mới được thiết lập. Bên cạnh những thay đổi về thủ tục, những biến đổi về chức năng nghề nghiệp, cấu trúc tổ chức, quan hệ quyền lực, và hành vi tất cả đều cần phải được lập kế hoạch cẩn thận. Chẳng hạn, Hình 3.9 minh hoạ cho các chiều của tổ chức mà sẽ cần được đề cập tới cho việc lập kế hoạch và thực hiện hệ tự động hoá văn phòng.

Mặc dầu các hoạt động phân tích và thiết kế hệ thống được hỗ trợ để bao gồm việc phân tích tác động vào tổ chức, lĩnh vực này về mặt truyền thống đã bị bỏ quên lãng. **Phân tích tác động tổ chức** giải thích hệ thống được đề xuất sẽ ảnh hưởng tới cấu trúc tổ chức, thái độ, việc ra quyết định và vận hành của tổ chức như thế nào. Để tích hợp hệ thống tin thành công vào với tổ chức, việc định giá tác động tổ chức với tư liệu đầy đủ và thấu đáo cần phải được chú trọng nhiều hơn trong nỗ lực phát triển.



CHÚ Ý TỚI NHÂN TỐ CON NGƯỜI

Chất lượng của hệ thông tin phải được lượng tính dưới dạng tiêu chí người dùng chứ không phải chỉ là tiêu chí kỹ thuật của nhân viên hệ thông tin. Chẳng hạn, một mục tiêu dự án có thể là các nhân viên vào dữ liệu có khả năng học các thủ tục và mã cho bốn màn hình vào dữ liệu trực tuyến mới trong phiên huấn luyện nửa ngày.

Những lĩnh vực mà người dùng giao tiếp với hệ thống nên được thiết kế cẩn thận, với nhạy cảm tới các vấn đề công thái học. **Công thái học** nói tới tương tác giữa người và máy trong môi trường làm việc. Nó xem xét tới việc thiết kế công việc, vấn đề sức khoẻ, và giao diện người dùng cuối của hệ thống tin. Tác động của hệ thống ứng dụng lên môi trường làm việc và các chiều hướng nghề nghiệp phải được định giá đầy đủ.

Các nhà nghiên cứu quản lí và tổ chức đã gợi ý về cách tiếp cận *xã hội kĩ thuật* tới thiết kế hệ thống tin và thay đổi tổ chức. **Thiết kế xã hội kĩ thuật** nhằm tới việc tạo ra các hệ thống tin uốn tính hiệu quả kĩ thuật theo tính nhạy cảm của nhu cầu tổ chức và con người, dẫn tới sự thoả mãn công việc cao.

Tiến trình thiết kế xã hội kĩ thuật nhấn mạnh vào sự tham dự của những người bị hệ thống mới ảnh hưởng tới nhiều nhất. Kế hoạch thiết kế thiết lập ra các mục tiêu nhân văn cho hệ thống vốn dẫn tới việc tăng sự thoả mãn công việc. Người thiết kế đưa ra các tập hợp tách biệt những giải pháp thiết kế kĩ thuật và xã hội. Bản kế hoạch thiết kế xã hội thám hiểm các cấu trúc nhóm làm việc khác nhau, phân bổ nhiệm vụ, rồi thiết kế công việc cá nhân. Giải pháp thiết kế đáp ứng tốt nhất cho cả các mục tiêu kĩ thuật và xã hội được lựa ra làm thiết kế cuối cùng.

PHÂN TÍCH THIẾT KẾ CÓ CẤU TRÚC

Phương pháp luận phân tích thiết kế có cấu trúc cung cấp một phương pháp phát triển hệ thống cùng với một bộ dụng cụ tối thiểu. Sự nhấn mạnh cũng được đặt vào tính dễ hiểu đối với người sử dụng, đặc biệt trong các giai đoạn ban đầu. Phương pháp luận này về cơ bản bao gồm một số dụng cụ và kỹ thuật mô hình hoá tích hợp cần cho việc phân tích nghiệp vụ và thiết kế hệ thống máy tính.



Hình A.1 Các giai đoạn phát triển của phương pháp luận .

Hình A.1 minh họa cho các giai đoạn khác nhau có trong một dự án phát triển tiến hóa, và cho ấn tượng về mức độ gối nhau có thể giữa các giai đoạn này. Trong phương pháp luận này (cũng như trong nhiều phương pháp luận có cấu trúc tiến bộ nhất) người ta không xét tới sự cần thiết phải xây dựng các mô hình hệ thống vật lý hiện tại mặc dù đôi khi cũng có lợi khi sử dụng mô hình để làm sáng tỏ một thủ tục vật lý phức tạp. Quan điểm ở đây là phần lớn các thông tin về hệ thống vật lý đang có không cần thiết phải 'đưa vào' theo cách truyền thống trong quá trình khảo sát bởi vì phần lớn trong chúng không thích hợp đối với hệ thống mới và bởi vì người sử dụng, các chuyên gia trong hệ thống đó, là một bộ phận trong

nhóm phát triển và họ luôn luôn sẵn sàng tiếp tay cung cấp các thông tin cần thiết.

Điểm quan trọng nhất rút ra từ biểu đồ này là sự kiện hai giai đoạn chính sẽ được vận hành đồng thời cho phần lớn dự án; không cần phải hoàn tất mọi phần phân tích nghiệp vụ trước khi bắt đầu thiết kế. Điều này có nghĩa là làm bản mẫu, điều được xem là một phần của quá trình thiết kế, có thể bắt đầu từ giai đoạn tương đối sớm (có lẽ trong vòng một vài ngày đầu tiên).

A.1 GIAI ĐOẠN PHÂN TÍCH NGHIỆP VỤ

Mục đích của giai đoạn này là đưa ra phân tích logic về hệ thống hiện thời, rút ra từ đó các yêu cầu không tường minh của nghiệp vụ, điều hiển nhiên phải tính đến trong thiết kế hệ thống mới. Hình A.2 minh họa các sản phẩm của phân tích logic này, được biết với tên Mô hình nghiệp vụ. Có bốn kỹ thuật lập mô hình có cấu trúc đặc biệt nổi tiếng được dùng trong giai đoạn này và chúng được sử dụng ở dạng đơn giản nhất. Đó là:

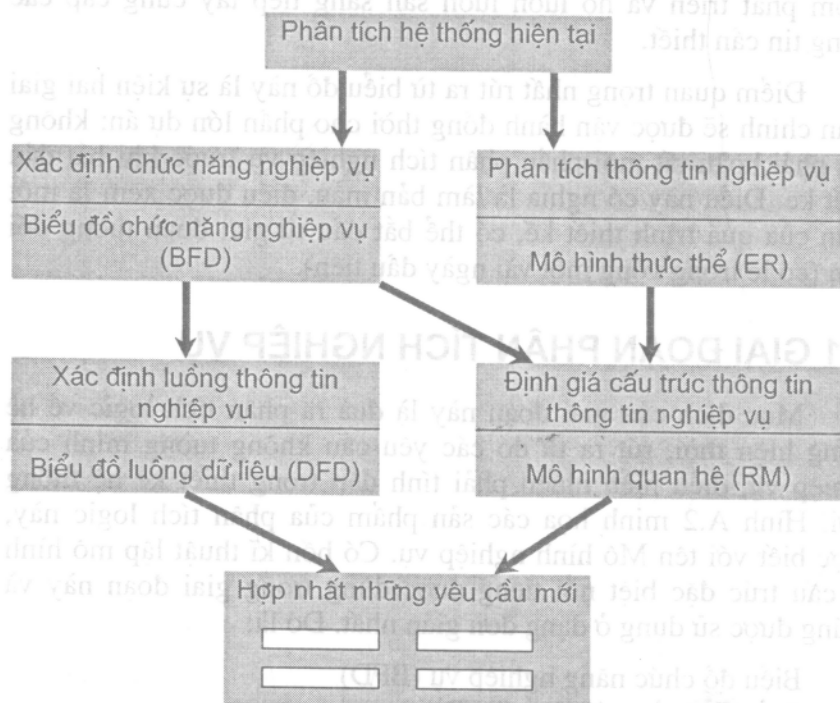
Biểu đồ chức năng nghiệp vụ (BFD)

Biểu đồ luồng dữ liệu (DFD)

Mô hình thực thể - quan hệ (ER)

Mô hình quan hệ (RM)

Mục đích của giai đoạn này còn là để đưa vào trong yêu cầu nghiệp vụ mọi tiện nghi phụ mà người sử dụng xác định rằng chưa có trong hệ thống hiện tại. Chúng được tích hợp vào nhóm thứ hai của các lược đồ tích hợp, Mô hình nghiệp vụ mới, trong thực tế nó thường là bản sao đã được chỉnh lại một chút của mô hình nghiệp vụ trước đó. Rất hiếm khi người sử dụng đòi hỏi các tiến trình hoặc dữ liệu mới; thông thường mối quan tâm chính là các yêu cầu hiện tại được xử lý nhanh hơn và chính xác hơn. Tất nhiên có một số phát triển chưa có trong hệ thống hiện tại.



Hình A.2 Giai đoạn phân tích nghiệp vụ.

Phân tích chức năng nghiệp vụ

Một trong những vấn đề khó khăn nhất mà các người phân tích phải đương đầu là việc xác định đúng những yêu cầu logic đằng sau hiện thực vật lí của hệ thống hiện thời. Để làm cho nhiệm vụ này dễ dàng hơn, phương pháp luận bắt buộc tiến hành phân rã một cách có thứ bậc các chức năng nghiệp vụ bên trong lĩnh vực nghiên cứu (biểu đồ chức năng nghiệp vụ). Việc nhấn mạnh vào chức năng hơn vào tiến trình gây khó khăn cho người phân tích làm chệch hướng ghi nhận cách thức các tiến trình được thực hiện về mặt vật lí. Ý tưởng này lại được chuyển tiếp sang hoạt động mô

hình hoá tiếp sau, vì các chức năng của BFD trở thành các tiến trình của biểu đồ luồng dữ liệu DFD. Các chuẩn được dùng cho DFD là của DeMarco, nhưng với một chút ít bổ sung, người ta đã cố gắng giữ cho các dụng cụ ban đầu này thân thiện tối đa với người sử dụng. Hai mô hình BFD và DFD kiểm tra chéo lẫn nhau và được lặp đi lặp lại cho đến khi đạt tới một mô hình các yêu cầu chức năng thoả đáng đơn giản và thẩm mỹ.

Một khía cạnh quan trọng khác của việc dùng biểu đồ chức năng nghiệp vụ BFD làm dụng cụ lập mô hình đầu tiên là ở chỗ người phân tích thường có thể xác định toàn bộ các chức năng tích hợp, rồi tiến hành khảo sát, phân tích và thiết kế một chức năng này một cách tách biệt đối với các chức năng khác. Điều này có nghĩa là người phân tích đôi khi có khả năng đặt phân cấp chức năng vào giai đoạn thiết kế hệ thống trong khi công việc phân tích còn đang được thực hiện cho các bộ phận khác của hệ thống. Đây là điểm chủ chốt trong chiến lược cung cấp bản mẫu sớm.

Phân tích các yêu cầu thông tin nghiệp vụ

Để xem xét chi tiết các yêu cầu thông tin của hệ thống, cần dùng thêm hai mô hình nữa; chúng được dẫn ra đây, vẫn với lí do để đơn giản hoá, là mô hình dữ liệu và mô hình quan hệ. Mô hình dữ liệu cơ bản là mô hình quan hệ-thuộc tính-thực thể đạt được thông qua cách tiếp cận từ trên xuống dưới (thuật ngữ mô hình dữ liệu và mô hình thực thể được dùng đồng nghĩa ở đây). Mô hình quan hệ được xây dựng từ những thuộc tính được xác định trong mô hình trước và được chuyển qua quá trình 'chuẩn hóa'. Hai mô hình này hiển nhiên được dùng để kiểm tra chéo lẫn nhau nhưng chúng cũng được tích hợp với các mô hình chức năng mà trong đó các kiểu thực thể trong bản cuối cùng của mô hình dữ liệu trở thành 'kho dữ liệu' trong bản cuối cùng của biểu đồ luồng dữ liệu.

Phải thừa nhận là riêng bốn mô hình này không thể tạo nên tư liệu đầy đủ cho đặc tả yêu cầu; cần phải có giải thích thêm, các

mục từ trong từ điển dữ liệu, các mô tả tiến trình mức thấp của DFD, các thông tin nền tảng.v.v.. Tuy nhiên, mức và nội dung của những tài liệu trợ giúp này là chủ đề cần đánh giá giữa các người phân tích, người sử dụng, nhà quản lí dự án. Tất cả những gì phương pháp luận cung cấp là chủ chốt để đảm bảo hướng dẫn cho một phân tích đầy đủ.

A.2 GIAI ĐOẠN THIẾT KẾ HỆ THỐNG

Trong khi giai đoạn phân tích nghiệp vụ thuần túy xử lí cho quan điểm logic về hệ thống và chẳng bao giờ liên quan tới cách thức có thể thoả mãn cho các yêu cầu, thì giai đoạn thiết kế hệ thống (Hình A.3) lại bao gồm việc xem xét ngay lập tức các khả năng cài đặt các yêu cầu nghiệp vụ này bằng cách sử dụng máy tính. Cần phải lưu ý rằng giai đoạn thiết kế đối với một vài phần của hệ thống nên bắt đầu càng sớm càng tốt để lợi dụng được những ưu thế của các cơ hội xây dựng bản mẫu.

Quá trình thiết kế dùng tất cả đặc tả yêu cầu được xây dựng trong quá trình phân tích làm đầu vào chính, bao gồm:

- Biểu đồ chức năng nghiệp vụ
- Biểu đồ luồng dữ liệu
- Mô hình thực thể
- Mô hình quan hệ

Các tài liệu trợ giúp gồm

- Các mô tả tiến trình
- Biểu đồ các yêu cầu vật lí
- Từ điển dữ liệu

Thêm vào đó cần phải tiến hành đối thoại liên tục và đầy đủ với người sử dụng và người chủ trong suốt quá trình thiết kế. Điều

này có thể thực hiện thông qua các cuộc phỏng vấn tìm hiểu sự kiện, những cuộc họp xét duyệt và trình diễn. Cách quan trọng nhất để tiến hành đối thoại là thông qua bản mẫu và phương pháp luận cũng hướng vào việc cung cấp các cơ hội làm bản mẫu ở một số giai đoạn thiết kế.

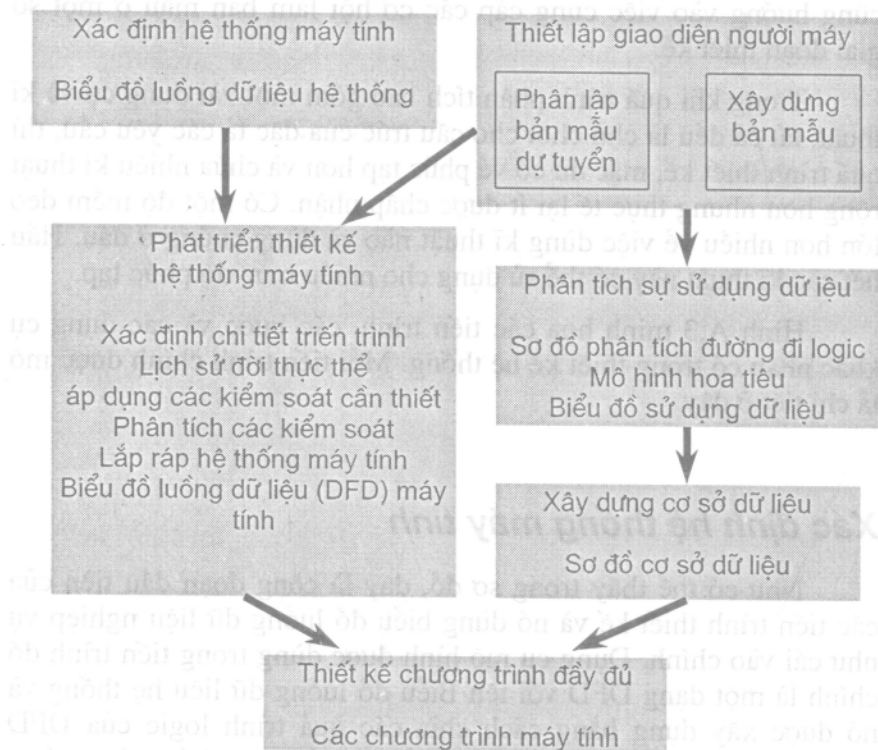
Trong khi quá trình phân tích bao gồm một vài dụng cụ và kỹ thuật, tất cả đều là chủ chốt cho cấu trúc của đặc tả các yêu cầu, thì quá trình thiết kế, mặc dù có vẻ phức tạp hơn và chứa nhiều kỹ thuật rộng hơn nhưng thực tế lại ít được chấp nhận. Có một độ mềm dẻo lớn hơn nhiều về việc dùng kỹ thuật nào và dùng chúng ở đâu. Hầu hết các kỹ thuật này có thể sử dụng cho nhiều mức độ phức tạp.

Hình A.3 minh họa các tiến trình, các bước và các dụng cụ khác nhau có trong thiết kế hệ thống. Mỗi tiến trình chính được mô tả chi tiết ở đây.

Xác định hệ thống máy tính

Như có thể thấy trong sơ đồ, đây là công đoạn đầu tiên của các tiến trình thiết kế và nó dùng biểu đồ luồng dữ liệu nghiệp vụ như cái vào chính. Dụng cụ mô hình được dùng trong tiến trình đó chính là một dạng DFD với tên Biểu đồ luồng dữ liệu hệ thống và nó được xây dựng bằng cách chia các quá trình logic của DFD nghiệp vụ thành các tiến trình vật lý, một số trong chúng được thực hiện bằng máy tính và số khác bởi con người. Đường đứt quãng đi qua trung tâm của biểu đồ DFD thường được dùng để phân tách các tiến trình máy tính ra khỏi các tiến trình thủ công. Tuy nhiên, chắc chắn có luồng dữ liệu liên lạc giữa các tiến trình máy tính và thủ công, và các luồng dữ liệu này biểu cho các tài liệu, các định dạng, các báo cáo hoặc các màn hình tạo ra giao diện người- máy của hệ thống.

Mô hình nghiệp vụ của hệ thống cần có



Hình A.3 Giai đoạn thiết kế hệ thống.

Từng người thiết kế riêng biệt có thể làm việc trên các lĩnh vực chức năng khác nhau của DFD nghiệp vụ, mỗi lĩnh vực có lẽ chỉ đại diện cho một phần nhỏ của toàn bộ hệ thống. Người thiết kế sẽ sử dụng kinh nghiệm và óc suy xét để xác định một hay nhiều cách dùng máy tính để hỗ trợ thực hiện các tiến trình trên DFD. Các cách chọn khác nhau này sẽ được mô hình hóa bằng cách sử dụng dụng cụ DFD hệ thống, sau đó được bàn luận với người chủ/

người sử dụng và đi đến thống nhất lấy ra một phương án được ưa chuộng hơn cả. Các cách chọn này có thể dùng cách tiếp cận xử lý theo lô hoặc trực tuyến và có thể chứa nhiều mức độ tham gia của máy tính, từ việc ghi nhận đơn giản những sự kiện nghiệp vụ xảy ra cho tới cài đặt đầy đủ trên máy tính tất cả các hoạt động đó.

Xác nhận các cái vào, cái ra

Một khi người sử dụng nhất trí với một cách dùng máy tính cho một phần của hệ thống, thì người thiết kế có khả năng xác định các khu vực của hệ thống có thể thích hợp với cách tiếp cận làm bản mẫu. Xây dựng nhiều bản mẫu nhỏ cho các phần khác nhau của hệ thống là thực tế thông dụng hơn là xây dựng một bản mẫu lớn cho toàn bộ. Các bản mẫu tiềm năng tốt xuất hiện ở nơi giao diện người-máy phức tạp và bao gồm các đối thoại trực tuyến. (Các tiến trình xử lý theo lô thông thường không đưa ra các cơ hội làm bản mẫu có giá trị).

Các dự tuyến bản mẫu riêng biệt được đánh dấu trên DFD hệ thống và có thể được trao cho từng người phân tích/lập trình, những người sẽ xây dựng bản mẫu của mình dùng các chi tiết lấy từ các đặc tả yêu cầu, từ những bản sao tài liệu hệ thống cũ và từ những bàn luận kỹ với người sử dụng. DFD hệ thống sẽ chỉ ra kho dữ liệu logic nào được sử dụng bởi tiến trình đang được làm bản mẫu. Như thế có thể xây dựng các tệp dữ liệu 'giả' để cho phép mô phỏng các truy nhập này. Có thể xây dựng một bản mẫu đầu tiên rất nhanh như vậy rồi có thể dùng các phản ứng của người sử dụng để gọt dũa bản mẫu cho đến khi đáp ứng được chất lượng đòi hỏi.

Điều gần như không tránh khỏi là người làm mẫu cần phải truy nhập tới môi trường thế hệ thứ tư để vận hành bản mẫu thành công.

yêu cầu vật lí của người sử dụng và cung cấp các mô hình rõ ràng để từ đó có thể xây dựng được các hệ thống con, các chương trình và các chương trình con.

Chính đặc trưng quan trọng của phương pháp luận này là ở chỗ nó phải đủ mềm dẻo và qui mô để trợ giúp mọi dạng phát triển hệ thống nghiệp vụ. Hiển nhiên những hệ thống nghiệp vụ nhỏ hơn không đòi hỏi cùng mức độ cố gắng thiết kế như các hệ thống lớn hơn, cho nên nhiều kĩ thuật rút từ các tiến trình con chỉ nên được dùng khi bản chất của phát triển đòi hỏi.

Xác nhận chi tiết tiến trình máy tính

Mục đích của tiến trình con này là để đảm bảo rằng không một khía cạnh kém hiển nhiên nào của hệ thống mới dự kiến bị bỏ sót. DFD là một dụng cụ rất có ích cho phân tích và thiết kế nhưng nó thiếu cơ chế tự kiểm tra toàn bộ. Để cho phép kiểm tra tính đầy đủ của DFD hệ thống, ta sử dụng một dụng cụ có tên là Lịch sử đòi thực thể.

Cách tiếp cận này bao hàm việc kiểm tra từng kiểu thực thể từ Mô hình dữ liệu, và tạo ra một danh sách liên tiếp các sự kiện mà các hoạt động sẽ ảnh hưởng tới nó. Chẳng hạn từng thực thể phải có một sự kiện để tạo ra thể nghiệm. Cũng như vậy cần có sự kiện gây ra hành động xóa một thể nghiệm riêng biệt. Mỗi thuộc tính trong thực thể có thể nhận một giá trị mới hoặc thay đổi giá trị cũ bởi một hay nhiều hành động. Tất cả các hành động hướng sự kiện có ảnh hưởng tới từng kiểu thực thể riêng biệt vậy sẽ được kiểm tra lại đối với các tiến trình trên DFD; tất cả các hành động có thuộc một quá trình DFD nào đó không? Nếu không thì DFD cần phải xem xét lại.

Trong một vài hệ thống, trật tự xuất hiện của các sự kiện có liên quan tới một thực thể đặc biệt có thể rất quan trọng. Chẳng hạn trước khi thanh toán, không những chỉ phải có đơn hàng, mà phải có xác nhận đã nhận được hàng; nếu việc thanh toán được thực hiện

không có xác nhận thì sai sót có thể xảy ra. Ở những nơi cần xem xét, thuộc tính 'chỉ báo trạng thái' được thêm vào kiểu của thực thể, để giữ các chi tiết của hành động cuối cùng đã xảy ra. Các thủ tục sai cho các dây sai cũng phải được thiết kế và đưa vào các mô hình khác để tạo ra đặc tả.

Áp dụng các kiểm soát cần thiết

Tiến trình con này bao gồm việc kiểm tra toàn bộ hệ thống được vẽ trong DFD hệ thống (kể cả những phần không có máy tính) và xác định những kiểm soát nào cần phải được áp dụng. Hành động này nên bao gồm sự cam kết và tham gia của những kiểm tra viên nội bộ cũng như của những người sử dụng tương lai của hệ thống đã hoàn tất bởi vì điều này có liên quan tới việc xác định các khu vực trọng yếu trong hệ thống dễ bị xâm phạm bởi:

- Trộm cắp
- Thất thoát tài sản của tổ chức
- Các quyết định nghiệp vụ lầm lẫn
- Chi phí quá mức
- Thiệt hại do lộ thông tin tổ chức

Khi một khu vực như thế được tìm ra, cần phải có các quyết định về mức của các kiểm soát cần thiết để vô hiệu hóa điểm yếu đó và điều này sẽ phụ thuộc vào tình thế gay gắt tới mức nào đối với tổ chức, nó xảy ra như thế nào và cơ chế kiểm soát tổn kém bao nhiêu.

Phương pháp được sử dụng bao gồm việc kiểm tra các phân lọ ra của hệ thống như được vẽ trong DFD hệ thống. Những cái ra có tiềm năng bị tổn thương là:

1. Dữ liệu đưa ra tác nhân bên ngoài
2. Dữ liệu lưu trữ bên trong hệ thống (có thể có việc truy nhập không chính thức đến những dữ liệu này)

Những tình huống và hoàn cảnh có thể gây ra thiệt hại cho từng cái ra này cần được kiểm tra bằng cách lần ngược lại qua mô hình DFD. Khi xác định được từng điểm yếu trên mô hình, cần phải quyết định có một vài hoạt động kiểm soát và sẽ đưa vào trong đặc tả.

Gộp nhóm các thành phần của hệ thống máy tính

Cho tới bây giờ tất cả các tiến trình của hệ thống cần được xử lí trên máy tính đã được xử lí như những hoạt động riêng biệt. Chúng được vẽ trên DFD hệ thống như trong mối quan hệ với các tiến trình khác tham gia vào việc thực hiện chức năng nghiệp vụ mức cao hơn. Tuy nhiên ở giai đoạn này cần phải phối hợp các tiến trình máy tính riêng biệt thành các hệ thống con, các bộ chương trình, các loạt chương trình, các chương trình, các modul và các chương trình con và những tổ hợp này hầu như không theo các đường chức năng thuần túy.

Dụng cụ chính được dùng trong việc gộp nhóm các thành phần của máy tính có tên là Biểu đồ luồng dữ liệu máy tính. Nó hoàn toàn là một DFD vật lí, tương phản với DFD nghiệp vụ logic và DFD hệ thống nửa vật lí. Kỹ thuật này cũng có khả năng tương đương với khả năng mô hình hoá cho các tình huống trực tuyến và xử lí theo lô, và nó thay thế cho cả hai dụng cụ truyền thống hơn, cho Biểu đồ thực hiện máy tính và Biểu đồ mạng. Công dụng bội của DFD bên trong phương pháp luận này ăn khớp với chính sách chung giảm tối đa số các kiểu kỹ thuật lập mô hình trong 'hộp dụng cụ' của người phân tích.

Thiết kế cơ sở dữ liệu vật lí

Đây là mô tả cho tiến trình mà theo đó người thiết kế tệp/cơ sở dữ liệu tạo ra các định nghĩa dữ liệu cho hệ thống dự kiến và

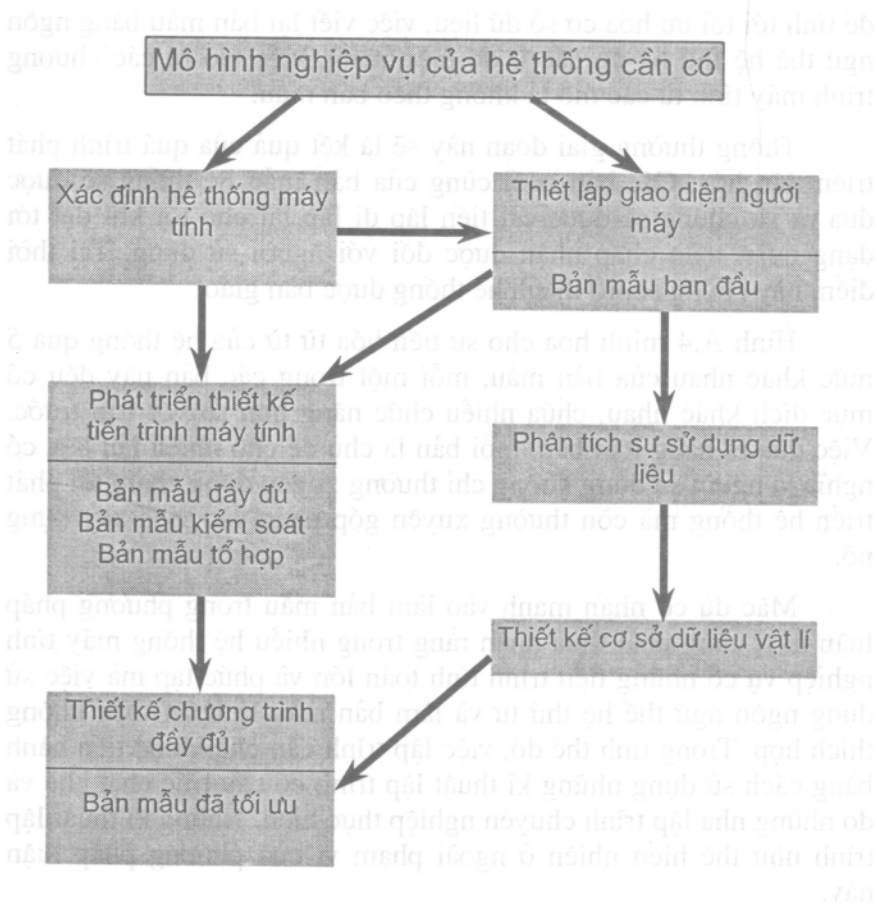
thiết lập các cấu trúc tệp chủ sẵn sàng cho cài đặt. Các thông tin sau đây là cần thiết để có thể tiến hành quá trình này:

1. Các hạn chế hiệu năng của người sử dụng dưới dạng phần cứng, phần mềm, thời gian đáp ứng v.v.
2. Các chi tiết từ phân tích sự sử dụng dữ liệu được tiến hành trước đó như là:
 - Mô hình dữ liệu
 - Mô hình quan hệ
 - Sơ đồ phân tích đường đi
 - Mô hình hoa tiêu
 - Biểu đồ sử dụng dữ liệu

Người thiết kế thông thường bắt đầu bằng việc áp dụng điều được gọi là 'qui tắc cắt thứ nhất' đối với mô hình dữ liệu logic, để chuyển chúng thành tập hợp sơ bộ các tệp tương ứng với phần mềm xử lý tệp cụ thể đang được sử dụng trong tổ chức. Từ đó trở đi những tệp này được 'nắn' và làm tối ưu cho đến khi chúng ăn khớp với yêu cầu hiệu năng của hệ thống.

Để bắt đầu, tiến trình tối ưu này có thể là một bài tập trên giấy mà ở đó các mô hình sử dụng dữ liệu được điều chỉnh và tính toán lại. Tuy nhiên trong các hệ thống căng thẳng về thời gian thường cần xây dựng 'bản mẫu hiệu năng' của một hoặc nhiều tệp quan trọng nhất, để chắc chắn rằng các ràng buộc tệp cần có được đáp ứng.

Hiển nhiên những thủ tục chính xác được người thiết kế tệp/cơ sở dữ liệu chấp nhận phụ thuộc rất nhiều vào phần mềm đặc biệt được sử dụng và vào phần cứng sẵn có. Kết quả là không thể đặc trưng được cho tiến trình này; mọi điều ta có thể đưa ra là những hướng dẫn và thí dụ.



Hình A.4 Bản mẫu đang tiến hoá.

Hoàn thiện thiết kế chương trình

Chi tiết của tiến trình này phụ thuộc vào phần mềm được sử dụng, vào kiểu của hệ thống đưa vào và phụ thuộc vào cách tiếp cận phân tích và thiết kế được tiến hành trong các giai đoạn trước. Chẳng hạn nó có thể bao gồm 'việc làm gọn gàng' bản mẫu tiến hóa

để tính tới tối ưu hóa cơ sở dữ liệu, việc viết lại bản mẫu bằng ngôn ngữ thể hệ thứ ba do các lý do hiệu năng, việc tạo ra các chương trình máy tính từ các mô tả không theo bản mẫu.

Thông thường giai đoạn này sẽ là kết quả của quá trình phát triển tiến hóa. Các bản cuối cùng của bản mẫu hệ thống sẽ được đưa ra và chúng sẽ được cải tiến lặp đi lặp lại cho tới khi đạt tới dạng hoàn toàn chấp nhận được đối với người sử dụng. Tại thời điểm này chúng sẽ trở thành hệ thống được bàn giao!

Hình A.4 minh họa cho sự tiến hóa từ từ của hệ thống qua 5 mức khác nhau của bản mẫu, mỗi một trong các bản này đều có mục đích khác nhau, chứa nhiều chức năng hơn so với bản trước. Việc đưa ra hàng loạt bản, mỗi bản là chủ đề cho nhiều lần lặp, có nghĩa là người sử dụng không chỉ thường xuyên động chạm tới phát triển hệ thống mà còn thường xuyên góp phần vào việc xây dựng nó.

Mặc dù cố nhấn mạnh vào làm bản mẫu trong phương pháp luận này, cũng phải thừa nhận rằng trong nhiều hệ thống máy tính nghiệp vụ có những tiến trình tính toán lớn và phức tạp mà việc sử dụng ngôn ngữ thể hệ thứ tư và làm bản mẫu là hoàn toàn không thích hợp. Trong tình thế đó, việc lập trình cần phải được tiến hành bằng cách sử dụng những kỹ thuật lập trình có cấu trúc chặt chẽ và do những nhà lập trình chuyên nghiệp thực hiện. Những kỹ thuật lập trình như thế hiển nhiên ở ngoài phạm vi của phương pháp luận này.

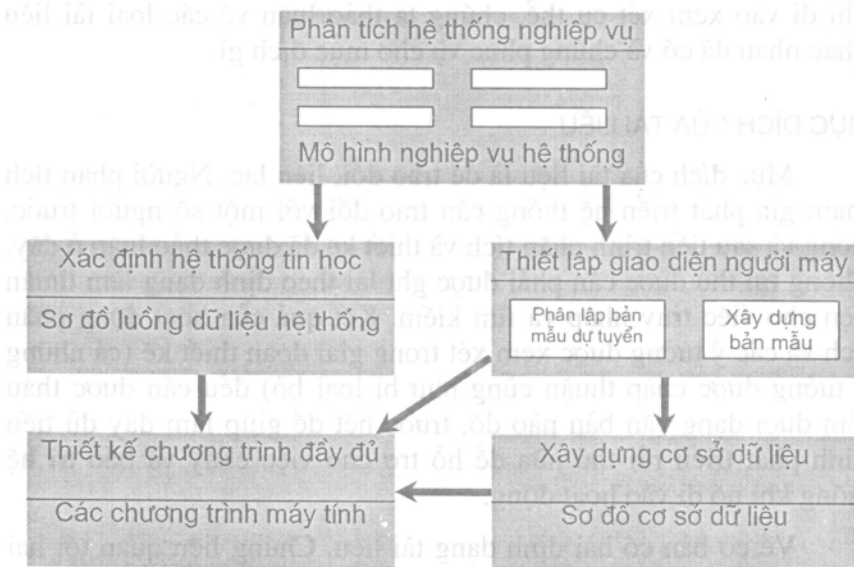
Tóm tắt

Phương pháp luận phân tích thiết kế hệ thống tiến hoá được thiết kế để cung cấp một tập hợp đầy đủ các dụng cụ lập mô hình cho việc phân tích và thiết kế các hệ thống tin học nghiệp vụ. Đặc biệt trong giai đoạn thiết kế có nhiều dụng cụ, không phải tất cả chúng sẽ được sử dụng trong tất cả các hệ thống. Chẳng hạn khi hệ thống là nhỏ, khi nó đơn giản về mặt logic, khi thời gian trả lời

chưa gấp gáp hoặc khi thông tin của hệ thống không nhạy cảm với việc bảo mật, thì một vài giai đoạn thiết kế có thể chỉ đòi hỏi một xử lý tượng trưng. Hình A.5 vẽ ra lộ trình của phương pháp luận ở dạng bé nhất, chỉ chứa các dụng cụ và các giai đoạn chủ chốt nhất đối với bất kỳ một sự phát triển hệ thống nào.

Ngay cả khi kỹ thuật lập mô hình được sử dụng thì vẫn có thể có những mức độ phức tạp và chi tiết khác nhau có thể áp dụng. Trong khi việc ít sử dụng các dụng cụ phát triển hệ thống có thể thành công đối với các hệ thống kém chất lượng, thì việc sử dụng thái quá chúng dẫn đến lãng phí tài nguyên cũng là điều không tốt.

Người phân tích mang trách nhiệm phát triển phải sử dụng óc suy xét trong việc quyết định chính xác cần dùng những kỹ thuật nào, mức độ áp dụng ra sao để cân bằng hai nguy cơ trên. Độ mềm dẻo và tính qui mô này được xem là một trong những ích lợi chính của phương pháp luận này.



Hình A.5 Phương pháp luận phân tích thiết kế tiến hoá dạng đơn giản nhất.

A.3 CÁC KHÍA CẠNH PHỤ CỦA PHƯƠNG PHÁP LUẬN

Điều cần thiết bây giờ là xem xét một cách tóm tắt một số chủ đề tổng quát vốn không thích hợp để trình bày trong các phần trên nhưng thiếu chúng thì mô tả về phương pháp luận sẽ không đầy đủ. Các chủ đề đó là:

1. Làm tư liệu phát triển hệ thống.
2. Bảo trì hệ thống đã phát triển bằng cách dùng cùng phương pháp luận.

Làm tư liệu phát triển hệ thống

Có nhiều kiểu báo cáo, tài liệu, tệp và đặc tả, tất cả cùng tạo nên tập đầy đủ các tài liệu cho hệ thống mới được dự định. Trước khi đi vào xem xét cụ thể, chúng ta thảo luận về các loại tài liệu khác nhau đã có và chúng phục vụ cho mục đích gì.

MỤC ĐÍCH CỦA TÀI LIỆU

Mục đích của tài liệu là để trao đổi, liên lạc. Người phân tích tham gia phát triển hệ thống cần trao đổi với một số người trước, trong và sau tiến trình phân tích và thiết kế đã được thảo luận ở đây. Thông tin thu được cần phải được ghi lại theo định dạng làm thuận tiện cho việc truy nhập và tìm kiếm. Kết quả của hoạt động phân tích và các ý tưởng được xem xét trong giai đoạn thiết kế (cả những ý tưởng được chấp thuận cũng như bị loại bỏ) đều cần được thu tóm dưới dạng văn bản nào đó, trước hết để giúp làm đầy đủ tiến trình phát triển rồi thứ nữa để hỗ trợ cho việc chạy và bảo trì hệ thống khi nó đi vào hoạt động.

Về cơ bản có hai định dạng tài liệu. Chúng liên quan tới hai nhóm người tham gia trong việc phát triển, và các nhu cầu thông tin khác nhau:

1. **Người dùng.** (Thuật ngữ được dùng ở đây bao hàm cả người quản lí, người chủ và người vận hành hệ thống.) Tài liệu cho những người này phải được chuẩn bị một cách chính thức bởi nhóm phát triển (một số trong họ cũng chính là người dùng). Tài liệu này được xem như một phần của việc bàn giao hệ thống. Trong phương pháp luận Phân tích thiết kế tiến hoá, các tài liệu bàn giao bao gồm

Đặc tả yêu cầu nghiệp vụ
Đặc tả thiết kế hệ thống
Tài liệu cho người dùng
Hướng dẫn vận hành

2. **Người phát triển.** (Thuật ngữ được dùng ở đây bao hàm cả người phân tích, người thiết kế, người làm bản mẫu, người lập trình, người quản lí dự án, chuyên gia CSDL v.v. đã tham gia vào tiến trình phát triển. Ta cũng có thể kể cả một số người dùng có tham gia nhiều vào phát triển hệ thống.) Tài liệu cho những người này bao gồm mọi thông tin có liên quan tới việc xây dựng hệ thống trong suốt thời kì nghiên cứu. Các tài liệu này thường được gọi là Hồ sơ giấy tờ làm việc.

HỒ SƠ GIẤY TỜ LÀM VIỆC

Hồ sơ giấy tờ làm việc là tập hợp đầy đủ các thông tin liên quan tới dự án của người phân tích. Nó là nguồn gốc để tạo ra tài liệu chính thức cho người dùng (mặc dầu các tài liệu người dùng này bản thân chúng cũng trở thành một phần của hồ sơ giấy tờ làm việc).

Hồ sơ này cần phải được tổ chức theo cách làm cho mọi phần thông tin liên quan tới việc phát triển hệ thống được lưu trữ và tìm kiếm hiệu quả nhất có thể được. Cấu trúc của hồ sơ này cũng cần phải đủ lỏng để cho phép lưu giữ đủ mọi kiểu thông tin liên quan tới hệ thống (và điều này có thể thay đổi rất lớn tuỳ theo bản chất

hệ thống). Đồng thời cấu trúc này cũng phải đủ xác định để đưa ra chỉ dẫn rõ ràng về các kiểu thông tin khác nhau nên được cất giữ ở đâu hay có thể tìm được chúng từ đâu. Điều này phải được thực hiện với việc tối thiểu trùng lặp dữ liệu.

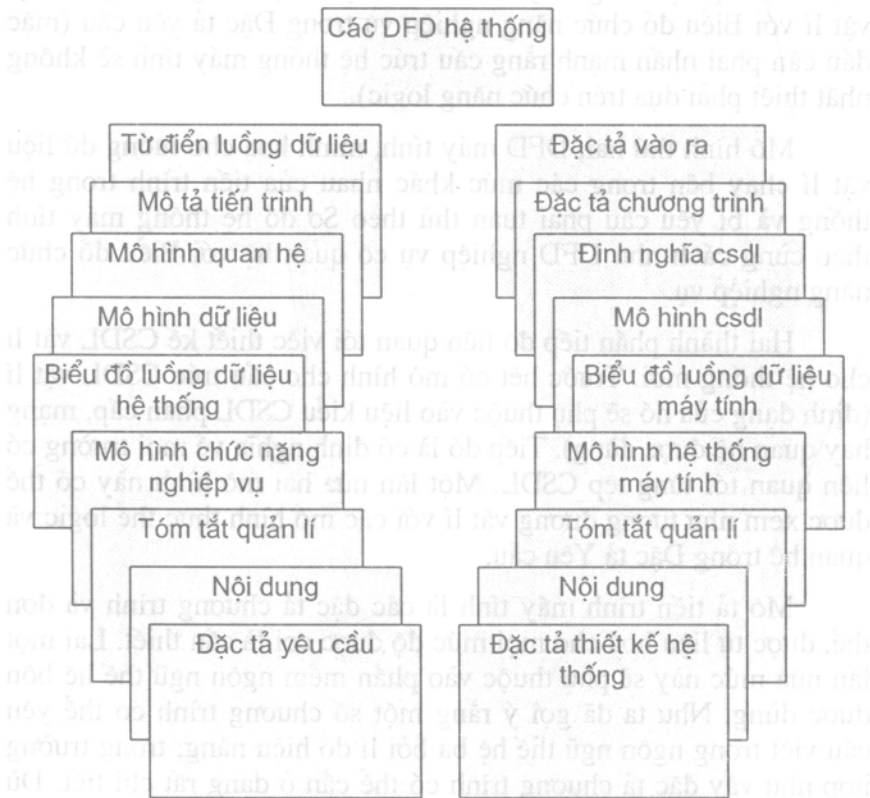
Đây là một lĩnh vực hiển nhiên các hệ thống tìm kiếm văn bản (và đồ hoạ) tỏ ra rất có ích. Người ta có xu hướng nghĩ tới dụng cụ CASE cho vấn đề này nhưng phần lớn các dụng cụ như vậy liên quan nhiều tới việc phát triển đồ hoạ cho các mô hình có cấu trúc. Chúng rõ ràng đóng một phần vai trò chính trong việc bảo trì tài liệu chính thức nhưng chúng không thể xử lý cho số lượng rất lớn và sự đa dạng của thông tin nhiều mặt vốn tích lũy trong Hồ sơ giấy tờ làm việc. Nghiên cứu hiện đại gợi ý rằng các hệ siêu văn bản rất có thể đưa ra được giải pháp. Tuy nhiên dường như là cũng phải vài năm nữa thì mới có được các hệ siêu văn bản thương mại cho việc bảo trì giấy tờ làm việc trong dự án phát triển hệ thống.

Hiện tại, khi toàn bộ tiến trình phát triển hệ thống vẫn đang được xem xét lại thì khó mà gợi ý được cấu trúc tốt nhất cho hồ sơ giấy tờ làm việc. Điều này có thể sẽ khác nhau cho từng tổ chức, từng dự án. Người quản lý phát triển hệ thống, người định xác định các chuẩn và các hướng dẫn cho công việc này sẽ phải làm việc nhiều để xem xét các kiến nghị được nêu ra như một phần của giáo trình Phân tích hệ thống cơ bản.

Đặc tả thiết kế hệ thống

Ngay khi kết thúc giai đoạn phân tích nghiệp vụ cần phải viết ra một báo cáo chính thức 'Đặc tả yêu cầu', còn khi hoàn tất thiết kế hệ thống cũng cần đưa ra mô tả chính thức cho thiết kế đó trong báo cáo có tên 'Đặc tả thiết kế hệ thống'. Hai báo cáo đặc tả này phải có cấu trúc cơ sở chung, theo đó một sẽ mô tả quan điểm 'logic' về hệ thống mới, dưới dạng người dùng yêu cầu cái gì, còn báo cáo kia thì mô tả biểu thị vật lý của hệ thống đó.

Hai báo cáo này không chỉ có cấu trúc đặc tả tương tự nhau mà các dụng cụ mô hình hoá thực tế được dùng cho các thành phần tương đương cũng có quan hệ với nhau rất chặt chẽ. Có thể quan sát điều này bằng cách so sánh sơ đồ minh họa cho hai cấu trúc (xem Hình A.6)



Hình A.6 Các thành phần của Đặc tả yêu cầu nghiệp vụ và thiết kế hệ thống.

Ngoài phần Nội dung và Tóm lược về quản lí, Đặc tả thiết kế hệ thống được tạo nên từ các mô hình đã xây dựng trong toàn bộ tiến trình thiết kế.

Mô hình đầu tiên trong số này là Biểu đồ hệ thống máy tính, chính là mô tả có cấp bậc cho các dãy, hệ con, chương trình và modun trong hệ thống máy tính mới. Đây là sự tương đương về mặt vật lí với Biểu đồ chức năng nghiệp vụ trong Đặc tả yêu cầu (mặc dầu cần phải nhấn mạnh rằng cấu trúc hệ thống máy tính sẽ không nhất thiết phải dựa trên chức năng logic).

Mô hình thứ hai, DFD máy tính, minh hoạ cho luồng dữ liệu vật lí chạy bên trong các mức khác nhau của tiến trình trong hệ thống và bị yêu cầu phải tuân thủ theo Sơ đồ hệ thống máy tính theo cùng cách như DFD nghiệp vụ có quan hệ với Biểu đồ chức năng nghiệp vụ.

Hai thành phần tiếp đó liên quan tới việc thiết kế CSDL vật lí cho hệ thống mới. Trước hết có mô hình cho cấu trúc CSDL vật lí (định dạng của nó sẽ phụ thuộc vào liệu kiểu CSDL phân cấp, mạng hay quan hệ được dùng). Tiếp đó là có định nghĩa về mọi trường có liên quan tới từng tệp CSDL. Một lần nữa hai mô hình này có thể được xem như tương đương vật lí với các mô hình thực thể logic và quan hệ trong Đặc tả Yêu cầu.

Mô tả tiến trình máy tính là các đặc tả chương trình và đơn thể, được tư liệu hoá cho mọi mức độ được coi là cần thiết. Lại một lần nữa mức này sẽ phụ thuộc vào phần mềm ngôn ngữ thế hệ bốn được dùng. Như ta đã gợi ý rằng một số chương trình có thể yêu cầu viết trong ngôn ngữ thế hệ ba bởi lí do hiệu năng; trong trường hợp như vậy đặc tả chương trình có thể cần ở dạng rất chi tiết. Dù các đặc tả đó lấy dạng nào chúng cũng phải được hỗ trợ thêm trong phần phụ lục bởi bản in chương trình.

Đặc tả vào/ra nói tới các màn hình và báo cáo máy tính. (Chú ý: chúng không nói đến các tài liệu thư kí được gõ vào trong hệ thống. Những tài liệu này sẽ được cất riêng trong 'Tài liệu người

dùng'.) Không có qui định về phương pháp mô hình hoá đối thoại người-máy trong phương pháp luận Phân tích thiết kế tiến hoá; ở đây chỉ thử và buộc dùng cách tiếp cận đơn giản và một tập các chuẩn thông thường trong việc tạo ra đối thoại đó. Tuy nhiên sẽ có một số dự án trong đó việc mô hình hoá cho các đối thoại phức tạp là chủ yếu. Khi điều này xảy ra, cần phải lưu ý rằng phương pháp sơ đồ dựa theo Sơ đồ cấu trúc Jackson nên được dùng. (Phương pháp này có ưu điểm rất giống với kỹ thuật được dùng cho mô hình hoá lịch sử đời thực thể.)

Cần phải nhấn mạnh rằng một tập tối thiểu các mô hình đã được đưa vào trong đặc tả. Các sơ đồ như Lịch sử đời thực thể và Sơ đồ phân tích đường đi logic có thể có tầm quan trọng chủ chốt trong việc phát triển thiết kế, nhưng không cần phải nêu ra cách xây dựng hệ thống đã được thoả thuận sau cùng. Các mô hình như vậy sẽ còn lại như một phần của hồ sơ giấy tờ làm việc.

Còn một ngoại lệ quan trọng cho qui tắc này. Mặc dầu DFD hệ thống không phải là bộ phận thực sự của cả Đặc tả yêu cầu lẫn Đặc tả thiết kế hệ thống, chúng về bản chất là cầu nối giữa hai đặc tả này. Khi phải bảo trì hệ thống mới, mọi sửa đổi dự định đều phải được xem xét lại cả ở mức độ logic lẫn vật lý, và ghép nối duy nhất là thông qua DFD hệ thống. Do đó có gợi ý là mô hình DFD hệ thống nên được giữ trong Đặc tả yêu cầu nghiệp vụ, thay thế cho thành phần 'Yêu cầu vật lý' trước đây (xem Hình A.6).

Bảo trì hệ thống đã xây dựng

Quan điểm truyền thống về phát triển hệ thống là ở chỗ chương trình được xây dựng để xử lý các yêu cầu nghiệp vụ như khi được nhìn nhận vào một thời điểm đặc biệt. Khi thời gian trôi qua, phải thực hiện những thay đổi cho các chương trình này để tìm lỗi trong thiết kế gốc và để đưa thêm vào các yêu cầu mới. Những thay đổi này được 'vá' vào chương trình (tại các vị trí thích hợp có thể tìm được). Cuối cùng khi đã gán vào quá nhiều miếng vá thì cấu

trúc cơ bản bị phá hoại nghiêm đến mức nó trở thành gần như không thể tổ hợp thêm được cái gì nữa vào; hệ thống máy tính không còn là một biểu diễn chấp nhận được cho yêu cầu nghiệp vụ, và cần phải nghiên cứu, phân tích và thiết kế lại hệ thống từ đầu. Đây là bản chất cố hữu của việc phát triển hệ thống theo vòng đời.

Quan điểm hiện đại coi hệ thống và chương trình cùng tiến hoá trong toàn bộ cuộc đời. Những thay đổi được thực hiện qua thời gian không được gây ra 'hao mòn' cấu trúc của hệ thống máy tính mà phải làm cải thiện nó lên. Sau hết hệ thống tiến hoá về việc đạt tới tiềm năng của nó. Những thay đổi được tiến hành không chỉ đơn thuần bao gồm việc thay thế chương trình mà còn tự động bao gồm cả việc xác định lại và đặc tả lại các yêu cầu.

Nguồn gốc cho sự thay đổi này về mặt triết học nằm ở các khái niệm hiện đại về mô hình hoá hệ thống có cấu trúc, kiến trúc CSDL và phát triển thế hệ bốn. Nó có quan hệ trực tiếp tới sự khác biệt giữa việc người ta yêu cầu hệ thống máy tính cái gì và cách thức thực hiện yêu cầu đó ra sao.

Quan điểm truyền thống đã ra đời trong thời kì xử lý theo lô còn thống trị và dữ liệu được lưu giữ trên băng từ theo kiểu tổ chức tệp tuần tự. Phần lớn độ phức tạp của các hệ thống đó đều bắt nguồn từ vấn đề tạo ra một giải pháp cho các vấn đề nghiệp vụ tương đối đơn giản và chuẩn bằng cách dùng một môi trường phần cứng không thân thiện, nguyên thủy. Người ta đã phải đưa vào các thủ tục xử lý theo lô, các chương trình sắp xếp và các tệp trung gian. Bởi bản chất của xử lý theo lô, phần lớn hoạt động đáng ra được thực hiện bởi con người tốt hơn là bởi máy tính thì lại bị buộc phải đưa vào hệ thống máy tính công kênh làm cho nó lại càng thêm phức tạp và nặng nề.

Nhờ có các phát triển hiện đại trong phần cứng, phần mềm và kĩ thuật thiết kế hệ thống, chúng ta đang chuyển ngày càng gần hơn tới tình huống mà cấu trúc của cái được yêu cầu từ hệ thống máy tính chính trùng với cấu trúc của cái mà hệ thống sẽ cung cấp (một

ngạn ngữ tin học hiện đại 'cái bạn thấy chính là cái bạn có được'). Tiềm nghi tương tác trực tuyến, chuyển đổi tự động từ 'đặc tả' thành chương trình máy tính, dùng các kĩ thuật hướng sự vật để ánh xạ 'thế giới thực' vào hệ thống, tất cả đều có vai trò trong việc làm hẹp lại lỗ hổng giữa yêu cầu và việc cung cấp.

Điều này có nghĩa là những điều chỉnh cho hệ thống trong cuộc đời làm việc của nó thường bây giờ có thể được diễn tả dưới dạng các yêu cầu logic chứ không phải là chương trình vật lí. Như vậy những điều chỉnh đó phải không gây ra thoái hoá cho cấu trúc hệ thống và do đó cũng không cần phải thiết kế lại toàn bộ hệ thống trong vòng sáu hay bảy năm!

Ý nghĩa của sự kiện này trong phương pháp luận Phân tích thiết kế tiến hoá là ở chỗ tiến trình thay đổi hệ thống sau khi nó đã đi vào 'hoạt động' chủ yếu chính là tiến trình phát triển ban đầu cho nó. Hệ thống làm việc có thể được coi như bản mẫu và người dùng sẽ thảo luận về sửa đổi yêu cầu với người phân tích theo cùng thuật ngữ như họ đã dùng trong khi luyện tập bản mẫu. Vậy có thể áp dụng các sửa đổi và các vòng lặp mới có thể được thực hiện hay các bản mới được đưa ra. Điều này hiển nhiên là việc làm quá đơn giản cho toàn bộ tiến trình sửa đổi hệ thống sống; cần phải có một dạng kiểm soát thay đổi, thậm chí cả kiểm soát quản lí dự án. Tất cả những điều này đều được nêu ở đây chính là nguyên lí mà việc phát triển tiến hoá tiếp tục tiến hành trong suốt cuộc đời của hệ thống, và là nguyên lí tương đối ít có sự khác biệt giữa việc phát triển ban đầu và việc bảo trì khi dùng phương pháp luận phát triển tiến hoá.

Mặc dầu đã trình bày tất cả các điều trên, bao giờ cũng có một số tình huống cần phải phân tích, thiết kế và phát triển lại hệ thống từ đầu, không tương ứng với yêu cầu người sử dụng ngay cả khi phát triển ban đầu đã được tiến hành bằng cách dùng các dụng cụ và kĩ thuật hiện đại như vừa thảo luận. Lí do cho điều này có thể thấy trong sự khác biệt giữa thay đổi liên tục và không liên tục. Với những thời kì dài trong bất kì hệ thống nào, tiến bộ có thể mang

nghĩa là 'càng nhiều càng tốt'. Tuy nhiên đôi khi có việc dịch chuyển chiều hướng bất ngờ, không lường trước được, mà điều đó dẫn đến phủ định hay lật nhào nhiều khái niệm và thành phần chính của hệ thống. Khi tính logic nằm sau mọi phần của hệ thống đã bị phá hoại ngầm theo cách này thì cách tiếp cận có thể duy nhất là quay trở lại với các nguyên lý đầu tiên.

Với tâm niệm như vậy, mục đích của việc xem xét này về bản chất tiến hoá của sự thay đổi trong hệ thống máy tính không phải là phủ quyết tính hợp lệ của việc phát triển lại hệ thống hoàn toàn mà đơn thuần chỉ đặt ra vấn đề đó như một giai đoạn thường lệ không tránh khỏi trong vòng đời hệ thống.

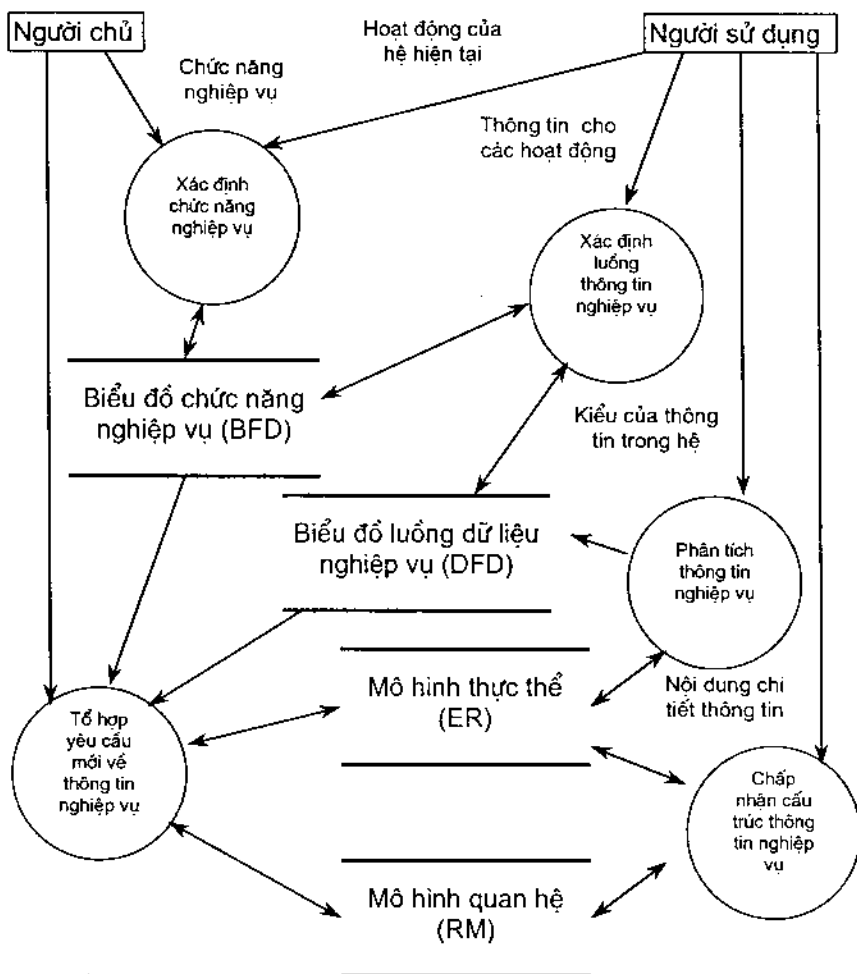
Tổng quan về phương pháp luận Phân tích thiết kế tiến hoá

Phương pháp luận Phân tích thiết kế tiến hoá đã được thiết kế để đưa ra một tập đầy đủ các dụng cụ mô hình hoá cho việc phân tích và thiết kế hệ thống máy tính nghiệp vụ. Trong các giai đoạn phân tích và thiết kế nói riêng, có nhiều dụng cụ, không phải tất cả trong chúng đều được dùng trong mọi hệ thống. Ngay cả khi một kỹ thuật mô hình hoá được sử dụng thì cũng còn nhiều độ phức tạp và chi tiết khác nhau cần phải được áp dụng.

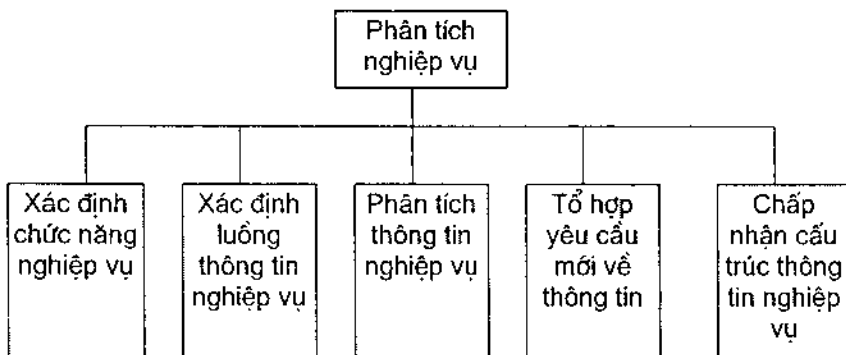
Vì phương pháp luận này chứa các dụng cụ cho việc mô hình hoá hệ thống và vì phương pháp luận này bản thân nó cũng là một loại hệ thống nên dụng cụ này cũng có thể được dùng để mô hình hoá cho chính nó. Sau đây ta sẽ xem xét các biểu đồ biểu diễn cho chính phương pháp luận này.

MÔ HÌNH CHO GIAI ĐOẠN PHÂN TÍCH HỆ THỐNG

Trong giai đoạn phân tích hệ thống, chúng ta có các biểu đồ chức năng nghiệp vụ BFD và biểu đồ luồng dữ liệu DFD sau:



Biểu đồ luồng dữ liệu cho giai đoạn phân tích

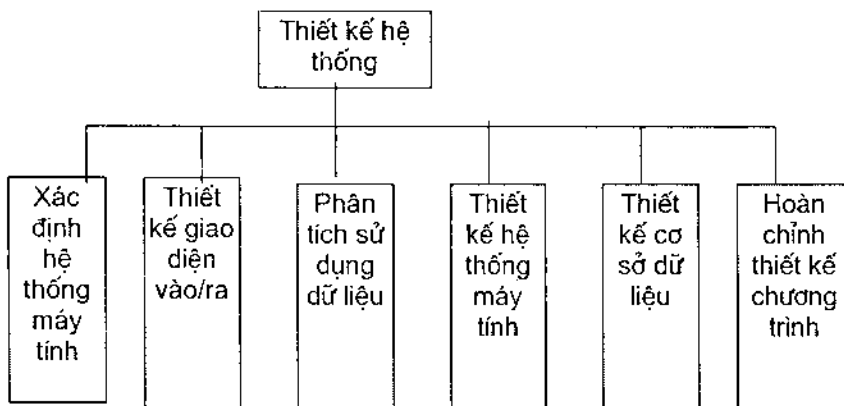


Biểu đồ chức năng nghiệp vụ cho giai đoạn phân tích

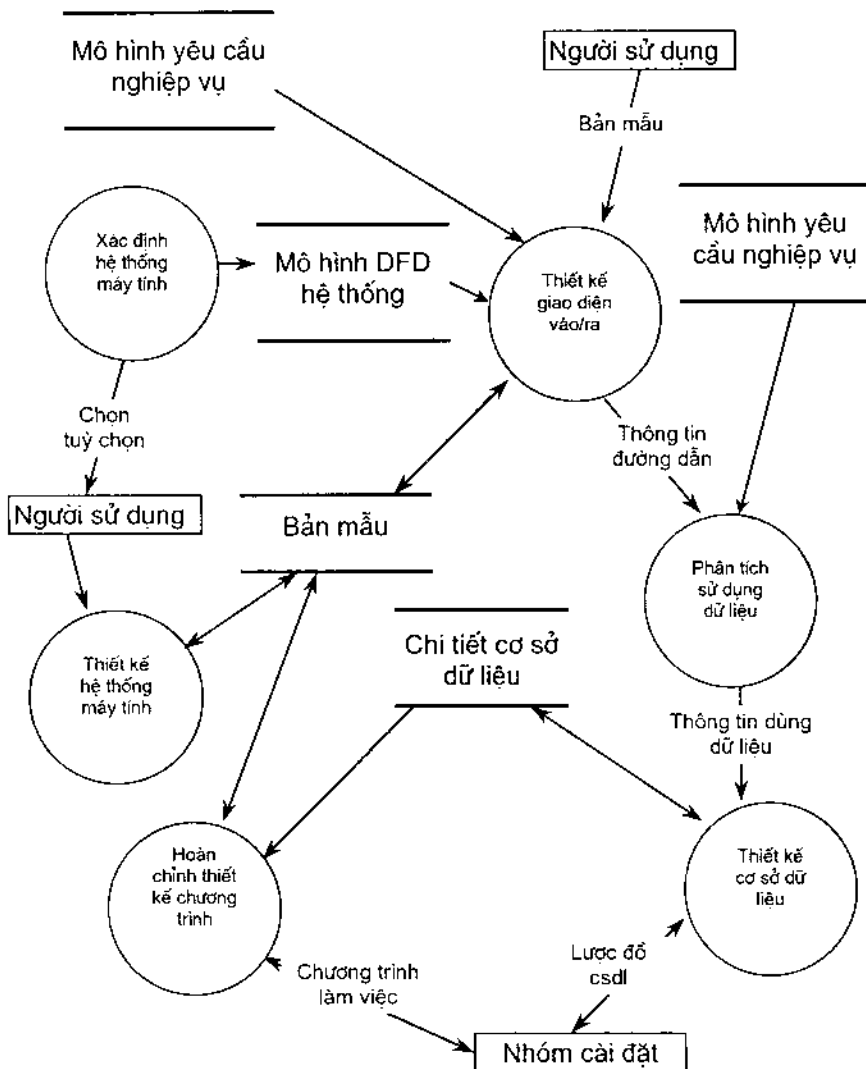
Hình A.7 Các biểu đồ BFD và DFD cho giai đoạn phân tích.

MÔ HÌNH CHO GIAI ĐOẠN THIẾT KẾ HỆ THỐNG

Để hoàn chỉnh mô hình này cho toàn bộ phương pháp luận, Hình A.8 được dùng để minh họa cho các chức năng của giai đoạn thiết kế hệ thống.



Biểu đồ chức năng nghiệp vụ cho giai đoạn thiết kế



Biểu đồ luồng dữ liệu cho giai đoạn thiết kế hệ thống
Hình A.8 Các BFD và DFD cho giai đoạn thiết kế hệ thống.

Tiện nghi mà phương pháp luận này đã mô hình hoá cho chính bản thân nó có thể rất có ích cho các người phân tích cấp cao, người phải quyết định về cách thức may đo phương pháp luận cho dự án cụ thể.

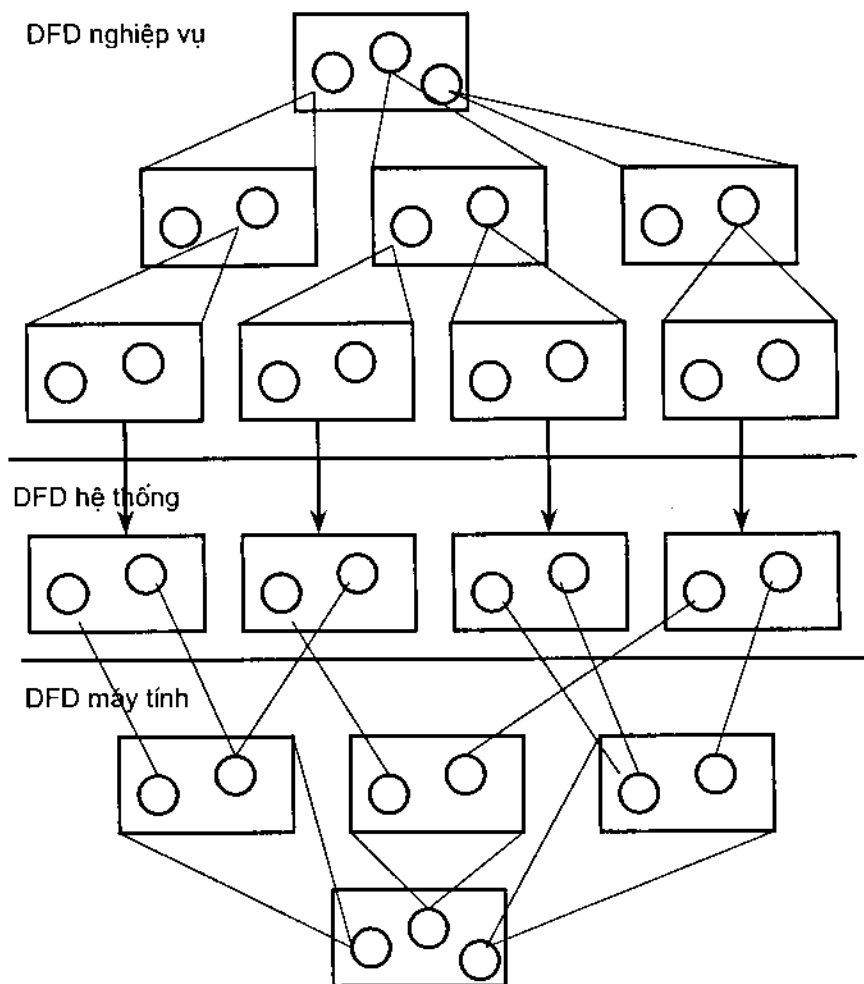
DÙNG TẬP KỸ THUẬT TỐI THIỂU

Một trong những nguyên tắc chỉ đạo được dùng khi cần phải chọn lựa các kỹ thuật mô hình hoá cho phương pháp luận là ở chỗ chỉ nên đưa vào một số tối thiểu các kiểu kỹ thuật khác nhau, và cùng một kỹ thuật nên được dùng cho nhiều giai đoạn khác nhau cho việc phát triển với các mục đích khác nhau. Điều này làm cho phương pháp luận trở nên dễ học hơn và tạo điều kiện xây dựng tri thức chuyên gia cho từng kỹ thuật.

Trong Phân tích thiết kế tiến hoá cách tiếp cận này đã được chấp thuận rộng rãi. Chẳng hạn, kỹ thuật phân tích đường đi logic có thể được thích ứng để minh hoạ cho việc truy nhập vật lý vào hệ CSDL mới, Biểu đồ sử dụng dữ liệu có thể được chuyển cho việc dùng Biểu đồ sử dụng tệp vật lý, và một biến thể của kỹ thuật mô hình hoá kiểu thực thể bản thân nó thường rất hay được sử dụng để trình diễn cấu trúc vật lý cuối cùng của CSDL.

Thí dụ quan trọng nhất về kỹ thuật mô hình hoá đa năng bên trong Phân tích thiết kế tiến hoá tất nhiên là DFD, được dùng để giúp phân tích dòng nghiệp vụ, xác định giới hạn hệ thống máy tính giúp thiết kế cấu trúc hệ thống máy tính tổng thể.

Mối quan hệ giữa chức năng và tiến trình được vẽ trên ba bản của mô hình này là mấu chốt, theo đó người thiết kế, người phải duy trì hệ thống máy tính, phải có thể xác định được mỗi tiến trình máy tính được tạo ra phục vụ cho chức năng nào. Hình A.9 minh hoạ cho sự liên nối này.



Hình A.9 Ba dạng khác nhau của biểu đồ luồng dữ liệu .

Bài tập

1. Thực hành về phương pháp phân tích hệ thống có cấu trúc: Lập biểu đồ chức năng nghiệp vụ theo tài liệu mô tả sau đây. Sau đó lập tất cả các biểu đồ luồng dữ liệu theo các mức của biểu đồ chức năng nghiệp vụ. Xác định các kiểu thực thể có thể có và xác định mối quan hệ giữa các thực thể đó

1. Mở đầu

Công ti Plumbers Part là một công ti chuyên cung cấp các mặt hàng ống dẫn nước và vật liệu xây dựng. Công ti mua với số lượng lớn vật tư từ nhiều nhà máy và cung cấp cho khách hàng qua tổng kho nằm bên cạnh trụ sở công ti.

Công ti Plumbers Part đã được thành lập từ hơn 10 năm trước và đang phát triển thịnh vượng. Gần đây một giám đốc mới được bổ nhiệm, ông ta có ý định tin học hóa công việc mua bán và các hoạt động quản lý kho của công ti. Ông ta yêu cầu anh/chị chuẩn bị các gợi ý cho việc sử dụng một máy tính như thế nào để giải quyết một số vấn đề mà ông đã đề ra. Nhằm giúp đỡ anh/chị, ông đã chuẩn bị một báo cáo ngắn giới thiệu các thủ tục và phương pháp làm việc của Công ti.

Trong bài tập thực hành này, anh(chị) sẽ nghiên cứu tình trạng hiện thời của Công ti Plumbers Part và gợi ý những cải tiến có thể được, với máy tính hoặc không, liên quan đến các hoạt động mua bán và kho tàng.

2. Báo cáo về các phương pháp và thủ tục của Công ti Plumbers Part

Người chuẩn bị: Giám đốc Công ti

Tháng 3/1990

Tôi chuẩn bị báo cáo này để giúp ông hiểu một số vấn đề mà Công ti Plumbers Part đang gặp phải. Vấn đề cơ bản của chúng tôi là do sự phát triển mạnh mẽ của các hoạt động mua bán đã dẫn chúng tôi đến tình trạng cho thấy các thủ tục đã có của chúng tôi để xử lý các đơn hàng của khách từ khâu nhận đến giao hàng là không thỏa đáng. Điều này đã gây bất lợi cho Công ti và làm tăng giá bán của Công ti Plumbers Part.

Bằng chứng rõ ràng nhất nói lên sự cần thiết phải có sự thay đổi thể hiện khi có một lượng lớn giấy tờ chuyển đến các văn phòng. Tôi đã nhận thấy rằng việc tìm kiếm một đơn hàng cụ thể là rất khó khăn.

Vấn đề cũng xuất hiện khi ghi hàng có trong kho thể hiện qua số lần phải điều chỉnh cho những phiếu xuất loại hàng không còn trong kho nữa. Cũng còn phải sử dụng quá nhiều phiếu điều chỉnh kho.

Việc xử lý các đơn hàng là không thỏa đáng, với việc xuất hàng chậm và một số đơn hàng phải gửi trả lại vì hết hàng trong kho.

Về mặt tài chính, sự kiểm soát vay nợ còn yếu. Đặt biệt với những khách hàng đang nợ nhưng vẫn còn phép mua hàng và việc cộng vào sổ nợ còn tồn đọng. Có thể sự kiểm soát đầy đủ về vay nợ nên chuyển sang bộ phận Kế toán và Tài vụ.

Mục tiêu của nghiên cứu

Ông cần đưa ra phương thức của Công ti Plumbers Part có thể:

- Rút ngắn thời gian đáp ứng các đơn hàng của khách
- Giảm bớt công việc bàn giấy và thời gian làm giấy tờ
- Phân phối thông tin trong công ti nhanh chóng và chính xác hơn.
- Cho phép kiểm soát quản lý cao hơn dựa trên việc cung cấp thông tin chính xác và kịp thời cho các báo cáo quản lý.
- Cung cấp thông tin quản lý bằng cách đưa ra những dữ liệu quá khứ để làm quyết định trong tương lai.
- Đưa ra các phương thức liên kết nhanh hơn và tốt hơn.

Tôi chắc chắn rằng Công ti Plumbers Part sẵn sàng sử dụng các hệ thống Tin học. Thực tế các ban Tiếp nhận và Kế toán của chúng ta đã sử dụng một máy vi tính PC rất có kết quả. Tôi tin rằng nhân viên của chúng ta trong các ban khác cũng có thể sử dụng một máy tính tương tự.

Giám đốc Công ti Plumbers Part.

3. Các thủ tục của Công ti Plumbers Part

3.1 Nhận đơn hàng

Đơn hàng được nhận trực tiếp từ khách hàng và các đại diện thương mại. Khách hàng cần ghi rõ số hiệu sản phẩm cũng như mô tả sản phẩm mà họ yêu cầu.

75% số đơn hàng nhận được ở dạng Đơn hàng tiêu chuẩn. 25% nhận được qua điện thoại và được chuyển thành đơn hàng bởi Phòng Bán hàng.

Khi nhận đơn hàng, các chi tiết được nhân viên bán hàng rà soát lại cho chính xác và đầy đủ. Những đơn hàng đầy đủ được chuyển

sang Kiểm soát viên Vay nợ. Đơn hàng của khách thường xuyên và có tin nhiệm sẽ được đi qua, những khách hàng mà kiểm soát viên không biết sẽ được giữ lại để kiểm soát vay nợ. Để làm việc đó, anh ta đối chiếu trong danh mục phiếu ghi khách hàng.

Nếu đây là một khách hàng mới, kiểm soát viên sẽ chuyển cho Giám đốc kinh doanh để xin ý kiến. Giám đốc kinh doanh xem xét các hoàn cảnh cụ thể, đặt một giới hạn vay nợ và tạo một phiếu ghi khách hàng mới.

Nếu đơn hàng từ một khách hàng cũ thì phiếu ghi khách hàng sẽ cho biết giới hạn vay nợ của khách và hiện nay ông ta còn đang nợ bao nhiêu. Kiểm soát viên đánh giá giá trị của đơn hàng mới nếu tổng số không vượt quá giới hạn thì đơn hàng sẽ được đưa sang phần xử lý. Nếu đơn hàng mới vượt quá giới hạn cho vay thì sẽ được chuyển giám đốc xử lý đơn hàng để xem xét. Khoảng 20 đơn hàng mỗi tuần được chuyển xem xét như vậy. 70% trong số đó bị chối cho đến khi nợ của khách được giảm bớt. Năm qua công ti đã mất 55.000 U \$ do khách hàng quỵt nợ.

3.2 Ghi kho

Một khi đơn hàng đã được kiểm tra thì lượng hàng còn trong kho cũng được kiểm tra. Ba thủ kho và một trưởng ban phải tra cứu trong tệp Sổ cái của kho để khẳng định điều đó. Tệp Sổ cái của kho gồm có 3035 phiếu lưu trữ trong ba ngăn kim loại. Mỗi phiếu ghi chi tiết một mặt hàng cùng với lịch sử mua bán mặt hàng đó. Mỗi ngăn do một thủ kho kiểm soát và gồm những phiếu theo số thứ tự của Mã mặt hàng. Mỗi đơn hàng có thể phải chuyển qua cả ba thủ kho để xem tất cả các mặt hàng yêu cầu có còn đủ hay không. Mỗi thủ kho phải xem trong đơn hàng có những mặt hàng anh ta chịu trách nhiệm và kiểm tra xem mặt hàng đó trong kho có còn đủ đáp

ứng đơn hàng hay không. Nếu còn dư, thủ kho đánh dấu mặt hàng vào đơn hàng và sửa đổi vào phiếu kho tương ứng.

Nếu không có đủ hàng thì thủ kho vẫn cấp phát lượng đang có và điều chỉnh vào Phiếu xuất hàng cho khách.

Nếu lượng hàng thậm chí không đủ cho cả mức tối thiểu thì thủ kho phải gửi một Yêu cầu Mua cho Ban Mua hàng để cung cấp lượng tối thiểu yêu cầu.

Khi đã kiểm tra trong kho xong, đơn hàng được chuyển cho kiểm soát viên ghi kho. Anh ta kiểm soát lại đơn hàng và nếu mọi việc ổn thỏa thì đơn hàng được giao cho các nhân viên Xử lý hóa đơn để chuẩn bị giao hàng cho khách.

Nếu không có mặt hàng nào trong đơn hàng có thể được cung cấp đầy đủ thì kiểm soát viên ghi kho sẽ ghi một Phiếu yêu cầu sau trong đó anh ta ghi lại toàn bộ hay một phần những mặt hàng có thể được cung cấp về sau. Phiếu xuất hàng cho khách đã điều chỉnh được chuyển qua Ban Tài vụ. Những phiếu yêu cầu sau cũng sẽ được điền đầy đủ khi Ban kế toán thông báo cho kiểm soát viên ghi kho về những điều chỉnh đã được thực hiện cho một đơn hàng trong khi xuất kho hoặc giao hàng.

Khi Ban ghi kho được thông báo về hàng nhập kho thì tệp Yêu cầu sau được soát lại và mỗi mặt hàng có thể cấp được đều được làm ngay bằng cách đưa ra một Đơn hàng Phụ được xử lý theo cách thông thường.

Tốc độ kiểm tra phiếu kho của các thủ kho trung bình từ 70 đến 80 phiếu kho/giờ. Sự tăng số lượng mặt hàng và đơn hàng do sự phát triển của công ti cùng với tính chất lặp lại đơn điệu của công việc đã dẫn đến chất lượng công việc kém ở khâu này.

Khi đơn hàng đã được kiểm soát viên ghi hàng chuẩn y, thì các nhân viên Xử lý hóa đơn tra trong Biểu giá để kiểm tra giá ghi trên

đơn hàng. Sau đó, họ sao các chi tiết của đơn hàng thành 7 bản. Đơn hàng gốc được ghi lại để tham khảo về sau.

7 bản sao của đơn hàng được phân phối như sau:

Mẫu Đỏ (Phiếu gom hàng) cho Kho
Mẫu Hồng (Phiếu lịch biểu) cho Kho
Mẫu Da cam (Phiếu giao hàng) cho Kho
Mẫu Vàng (Thông báo xuất) cho Kho
Mẫu Trắng (Hóa đơn) cho Kế toán
Mẫu Xanh lơ (Sao hóa đơn) cho Kế toán
Mẫu Xanh (Sao cho người mua) cho Vật tư

Kế toán và tài vụ

Ban kế toán và Tài vụ nhận hai bản sao đơn hàng. Hai bản sao này (màu Trắng và Xanh lơ) dưới dạng hóa đơn được giữ lại cho đến khi phiếu xuất đã có chữ ký tương ứng được gửi đến.

Phiếu xuất đã ký này và cặp hóa đơn được so theo Số đơn hàng. Những thay đổi được ghi trong phiếu xuất cũng được làm thay đổi theo trong hai hóa đơn. Nếu một mặt hàng nào được đánh dấu "chưa hết" thì một Đơn hàng còn thiếu sẽ được lập và được ghi cho bộ phận Ghi kho.

Bản hóa đơn chính được tính tổng và gửi lại cho khách hàng còn bản sao được lưu trữ lại.

Một bản chụp hóa đơn được gửi cho bộ phận Marketing với mục đích thông tin.

Các bản sao của các Đơn hàng Mua do Ban mua hàng gửi đến sẽ được giữ cho đến khi phiếu báo giao hàng của bên bán hàng được gửi đến. Phiếu báo giao hàng của bên cung cấp hàng sẽ được kiểm tra và trả tiền. Tất cả giấy tờ liên quan đến Đơn mua hàng được lưu trữ trong 2 năm sau đó mới được hủy đi.

Hai nhân viên trong Ban Kế toán và Tài vụ chịu trách nhiệm về tài vụ của công ti. Sổ cái được vào khi Hóa đơn gửi đến và Kế toán và Tài vụ chuyển một danh sách hàng ngày lượng tiền mặt đã nhận được và đã làm hóa đơn cho bộ phận Xử lý đơn hàng để cập nhật vào tệp Khách hàng. Một bộ chương trình tài vụ đã được sử dụng trên máy vi tính PC của họ.

Ban Kế toán và Tài vụ cũng chịu trách nhiệm đưa ra Danh mục giá sửa đổi cho các bộ phận Marketing và Xử lý đơn hàng.

Một việc liên quan chính trong Ban này là vấn đề khách hàng trả tiền sau cho các hóa đơn. Đặc biệt trong trường hợp những khách hàng này vẫn còn có thể mua thêm hàng mặc dầu đã vượt quá giới hạn nợ cho phép là một điều lo ngại. Ông giám đốc Kế toán và Tài vụ tin rằng vấn đề này có thể giải quyết hoàn hảo nếu bộ phận xử lý đơn hàng kiểm tra trạng thái hiện thời của một tài khoản khách hàng cùng với ông ta cho tất cả đơn hàng nhận được.

Bởi tầm quan trọng đặc biệt của Ban này nên tôi mong muốn công việc tài vụ của ban này không được bị đảo lộn. Mọi khuyến cáo của ông đều phải chú ý đến điều đó.

3.3 Mua hàng

Một bản sao của mỗi đơn hàng (bản màu xanh) được gửi cho ông Giám đốc Mua hàng để ông ta đánh giá xu hướng mua của khách hàng với các mặt hàng trên cơ sở đó thay đổi việc mua hàng định kỳ từ các cơ sở.

Ông ta nhận được các yêu cầu mua từ bộ phận ghi kho khi một mặt hàng nào đó còn nhỏ hơn mức yêu cầu của đơn hàng và khi có những yêu cầu xuất hàng mới.

Một bản sao Yêu cầu mua hàng được gửi cho Tài vụ và Kế toán để trả tiền và cho kho để trừ tính việc nhập hàng.

3.4 Công việc của nhà kho hàng

Khi các bản sao đơn hàng cho kho hàng (Đỏ, Hồng, Da cam và Vàng) được các nhân viên xử lý đơn hàng gửi đến thì bản lịch biểu được gửi cho Kiểm soát viên giao hàng trong phiếu Giao hàng, thông báo giao hàng và phiếu gom hàng được gửi cho kiểm soát viên chuẩn bị hàng xuất. Tất cả các mặt hàng trong đơn hàng sẽ được gộp lại và thu xếp để chuyển giao cho khách hàng.

1) Chuẩn bị hàng xuất.

Một số nhân viên nhà kho được sử dụng hoàn toàn vào công việc đóng gói tất cả các mặt hàng của một đơn hàng vào một khối. Họ đi trong kho theo thứ tự cho bởi phiếu gom hàng. Đơn hàng đã đóng gói được chuyển vào khu vực cho giao hàng cùng với phiếu giao hàng và phiếu thông báo giao hàng. Phiếu gom hàng được gửi trả lại cho Kiểm soát viên lấy hàng và được hủy bỏ.

Nếu người ta phát hiện ra rằng không có đủ mặt hàng trong kho cho đơn hàng thì đơn hàng sẽ được điều chỉnh với đánh dấu "chưa đủ" vào mặt hàng còn chưa đủ. Một phiếu điều chỉnh kho chỉ ra lượng hàng còn lại trong kho sau cấp phát này được gửi cho bộ phận ghi kho.

Các phiếu điều chỉnh kho còn được sử dụng để thông báo cho bộ phận ghi kho tất cả mất mát hàng hóa đổ vỡ, hỏng và mất trộm. Kiểm kê kho được thực hiện năm một lần vào tháng hai và kết quả được gửi cho bộ phận ghi kho để tạo ra các phiếu ghi kho mới.

Kiểm soát viên lấy hàng cho rằng vấn đề lớn nhất là do bộ phận ghi kho không duy trì được thông tin ghi kho đúng. Người ta đã mất nhiều thì giờ khi mà kho nhận được đơn hàng cho các mặt hàng không có trong kho. Cũng xảy ra việc kho hàng bị đầy khi Ghi kho yêu cầu mua thêm mặt hàng vẫn còn có nhiều trong kho. Khi những

điều này xảy ra thì một phiếu điều chỉnh kho được gửi cho Ghi kho để thông báo nơi có vấn đề.

2) Giao hàng

Việc giao hàng cho khách được làm bằng xe tải hoặc xe tải hàng. Các phương tiện được chất hàng theo lịch biểu của Kiểm soát viên giao hàng. Không có một phương pháp tuyệt đối nào cho việc sắp xếp lịch biểu giao hàng và đôi khi một số xe chỉ được chất nửa thể tích và có đơn hàng bị giao chậm, Phiếu lịch biểu được giữ cho đến khi giao hàng xong, sau đó chúng được lưu vào tệp.

Nếu khách hàng không nhận toàn bộ hay một phần đơn hàng thì phiếu hàng được sửa lại để phản ánh đúng số hàng đã thực sự được giao.

Kiểm soát viên giao hàng than phiền rằng anh ta được thông báo về đơn hàng không hợp lý. Nếu anh ta nhận được tờ phiếu hồng Lịch biểu trước các phiếu giao hàng thì anh ta có thể sắp xếp việc giao hàng sớm hơn và có thể sử dụng một vận tải xe hợp lý hơn.

3) Hàng nhập kho

Khi hàng nhận được từ các nơi cung cấp thì chúng được tháo dỡ và kiểm tra theo tờ sao của Phiếu mua hàng do bộ phận mua hàng gửi đến. Khi hàng nhập đã được kiểm tra đầy đủ, phiếu giao hàng của chủ hàng sẽ được ký nhận và hàng được chuyển vào kho, Một bản sao phiếu giao hàng của chủ hàng được gửi cho Kế toán Tài vụ sau khi một phiếu Nhận hàng đã được điền đầy đủ. Một bản lưu phiếu này được Kiểm soát viên giao hàng lưu giữ và các bản gốc được đóng vào với nhau và chuyển bộ phận ghi kho hai lần một ngày.

Trung bình có khoảng 4 lần nhập 3 chủng loại mặt hàng mỗi ngày.

2. Thực hành quá trình chuẩn hóa dữ liệu cho một chứng từ thực tế

Đơn vị hóa đơn kiêm phiếu xuất kho Mẫu số: 7A VT

..... Số:

Ngày tháng năm 19....

Tài khoản

Họ tên người nhận: CMT số: Nợ Có

Tên và địa chỉ khách hàng

Theo hợp đồng số: ngày ... tháng ... năm ...

Phương thức bán xuất tại kho

Hình thức thanh toán:

T.T	Tên, nhãn hiệu, qui cách vật tư sản phẩm	Đơn vị tính	Mã vật tư sản phẩm	Số lượng	Đơn giá	Thành tiền
A	B	C	D	1	2	3

Cộng thành tiền (bằng chữ)

Xuất ngày tháng ... năm ...

Thủ trưởng Phụ trách Kế toán

đơn vị

cung tiêu trưởng

Người nhận

Thủ kho

NGÔN NGỮ MÔ HÌNH HOÁ THỐNG NHẤT (UML)

B.1 MÔ THỨC HƯỚNG SỰ VẬT

Khi những năm 1980 tới gần, “mô thức hướng sự vật” cho kỹ nghệ phần mềm bắt đầu chín muồi thành một cách tiếp cận có hiệu lực và thực tế tới việc phát triển phần mềm. Người ta đã tạo ra các thiết kế cho những ứng dụng thuộc đủ mọi loại bằng cách dùng định ý hướng sự vật, và đã cài đặt (mã hoá) chương trình dùng các ngôn ngữ và kỹ thuật lập trình hướng sự vật. Nhưng việc phân tích yêu cầu lại bị chậm lại sau.

Ngày nay, phân tích hướng sự vật (OOA) đang dần dần nhưng tiến vững chắc thành một phương pháp phân tích yêu cầu theo đúng nghĩa riêng của nó và như một phần bổ sung cho các phương pháp phân tích khác. Thay vì xem xét vấn đề bằng cách dùng mô hình vào - xử lý - ra (luồng thông tin) hay mô hình suy dẫn chuyên cho các cấu trúc thông tin cấp bậc, OOA đưa vào một số khái niệm mới. Những khái niệm mới này dường như bất thường với nhiều người, nhưng chúng lại hoàn toàn tự nhiên. Coad và Yourdon xem xét vấn đề này khi họ viết:

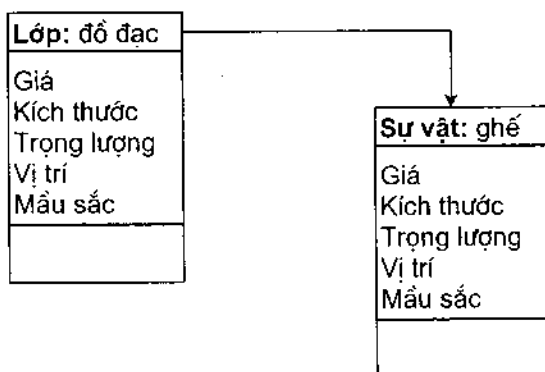
OOA - phân tích hướng sự vật - dựa trên những khái niệm mà chúng ta đã biết trước hết trong vườn trẻ: các vật và các thuộc tính, lớp và thành phần, toàn thể và bộ phận. Tại sao chúng ta lại phải mất thời gian lâu đến thế để áp dụng các khái niệm này vào việc phân tích và đặc tả hệ thống tin là một phỏng đoán của một ai đó - có lẽ vì chúng ta đã quá bận bịu “đi theo luồng” trong những ngày hoàng kim của phân tích có cấu trúc mà không xem xét tới các phương án khác.

Điều quan trọng cần lưu ý là không có sự đồng thuận phổ quát về “các khái niệm” nền tảng của OOA. Nhưng một số giới hạn các ý tưởng chủ chốt dường như lặp lại và đó là những khái niệm chúng ta sẽ xem xét trong mục này.

Khái niệm hướng sự vật

Bất kì thảo luận nào về phân tích hướng sự vật (OOA) cũng đều phải bắt đầu bằng cách nói tới thuật ngữ “hướng sự vật.” Quan điểm hướng sự vật là gì? Tại sao một phương pháp lại được xem là hướng sự vật? Sự vật là gì? Có nhiều ý kiến khác nhau về câu trả lời đúng đắn cho những câu hỏi này. Trong những thảo luận sau đây, chúng ta cố gắng tổng hợp phần chung nhất của những ý kiến này.

Để hiểu quan điểm hướng sự vật ta hãy xét một thí dụ về sự vật thể giới thực - cái bàn đang ngồi hiện tại - cái ghế. Ghế là một thành viên (thuật ngữ “thể nghiệm” đôi khi cũng được dùng) của một *lớp* các sự vật rộng hơn mà ta gọi là **đồ đạc**. Một tập các *thuộc tính* có thể được liên kết với mọi sự vật trong lớp **đồ đạc**. Chẳng hạn, mọi đồ đạc đều có **giá**, **kích thước**, **trọng lượng**, **vị trí** và **màu sắc**, trong nhiều thuộc tính có thể có. Những thuộc tính này áp dụng được dù cho ta nói về một cái bàn hay ghế, một cái tràng kỉ hay cái tủ. Bởi vì ghế là một *thành viên* của lớp **đồ đạc** nên ghế *kế thừa* mọi thuộc tính được xác định cho lớp. Khái niệm này được minh hoạ sơ đồ trong Hình B.1.



Hình B.1 Kế thừa từ lớp tới sự vật..

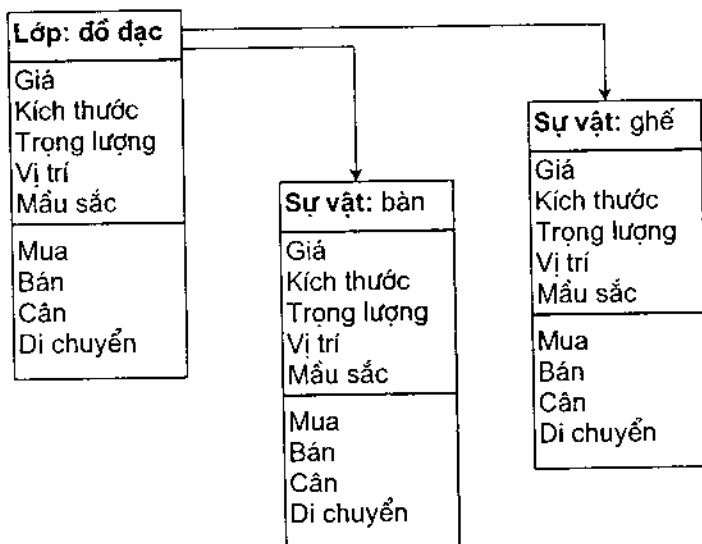
Một khi lớp đã được xác định thì các thuộc tính có thể được dùng lại khi những thể nghiệm mới của lớp được tạo ra. Chẳng hạn, giả sử rằng chúng ta định xác định một vật mới tên là **ghế-bàn** (chéo giữa ghế và bàn) mà vốn là thành viên của lớp **đồ đạc**. **Ghế-bàn** kế thừa các thuộc tính của **đồ đạc**.

Chúng ta định đưa ra một định nghĩa phụ về lớp bằng cách mô tả các thuộc tính của nó, nhưng còn thiếu một cái gì đó. Mọi sự vật trong lớp **đồ đạc** đều có thể được thao tác theo nhiều cách khác nhau. Người ta có thể mua hay bán chúng, có thể thay đổi chúng về mặt vật lí (như bạn có thể cưa bớt một chân hay đánh véc ni), hay chuyển chúng từ nơi này sang nơi khác. Mỗi một trong các *thao tác* (các thuật ngữ khác là *dịch vụ* hay *phương pháp*) sẽ làm thay đổi một hay nhiều thuộc tính của sự vật. Chẳng hạn, nếu thuộc tính **vị trí** thực tế bao gồm khoản mục dữ liệu được xác định là:

vị trí = toà nhà + tầng + phòng

thì một thao tác có tên **di chuyển** sẽ làm thay đổi một hay nhiều khoản mục dữ liệu (**toà nhà**, **tầng**, hay **phòng**) có chứa trong thuộc tính **vị trí**. Để làm điều này, **di chuyển** phải có “tri thức” về những khoản mục dữ liệu đó. Thao tác **di chuyển** có thể được ghế hay bàn sử dụng khi cả hai là thể nghiệm của lớp **đồ đạc**. Mọi thao tác hợp lệ (như **mua**, **bán**, **cân nặng**) cho lớp **đồ đạc** đều được “nối” với định nghĩa sự vật như được vẽ trong Hình B.2 và được kế thừa cho tất cả các thể nghiệm của lớp.

Sự vật **ghế** (và tất cả các sự vật nói chung) đều *bao bọc* dữ liệu (các giá trị thuộc tính xác định ra chiếc ghế), các *thao tác* (những hành động được áp dụng để làm thay đổi các thuộc tính của ghế), các sự vật khác (các sự vật hợp thành có thể được định nghĩa, các hằng (giá trị tập hợp), và các thông tin có liên quan khác. Bao bọc nghĩa là tất cả những thông tin này đều được đóng gói dưới một cái tên và có thể được dùng lại như một đặc tả hay thành phần chương trình.



Hình B.2 Kế thừa của các thao tác từ các lớp tới sự vật.

Chú ý rằng chúng ta đã giới thiệu một vài khái niệm mới, một định nghĩa hình thức hơn về “hướng sự vật” sẽ tỏ ra có ý nghĩa nhiều hơn. Coad và Yourdon định nghĩa thuật ngữ theo cách này:

hướng sự vật = sự vật + phân lớp + kế thừa + truyền thông

XÁC ĐỊNH SỰ VẬT

Nếu bạn nhìn quanh phòng, có một tập hợp các sự vật vật lí có thể dễ dàng xác định, phân lớp và định nghĩa (dưới dạng thuộc tính và thao tác). Nhưng khi bạn “nhìn quanh” không gian vấn đề của ứng dụng phần mềm, có thể sẽ khó xác định ra sự vật hơn.

Chúng ta có thể bắt đầu xác định các sự vật bằng cách xem xét việc phát biểu vấn đề hay thông qua việc thực hiện “phân tích

văn phạm” trên lời thuật xử lí cho hệ thống cần xây dựng. Các sự vật được xác định bằng việc gạch dưới các danh từ hay cụm danh từ và đưa nó vào trong một bảng đơn giản. Cũng nên lưu ý đến cả các từ đồng nghĩa. Nếu cần một sự vật để cài đặt cho một giải pháp thì nó là một phần của *không gian giải pháp*; nếu không, nếu một sự vật cần thiết chỉ để mô tả cho một giải pháp thì nó là một phần của *không gian vấn đề*.

Các sự vật có thể là:

- *Thực thể ngoài* (như hệ thống khác, thiết bị, con người) tạo ra hay tiêu thụ thông tin được dùng bởi một hệ thống tin học hoá
- *Các vật* (như báo cáo, thư, tín hiệu) vốn là một phần của miền thông tin cho vấn đề
- *Sự xuất hiện hay biến cố* (như chuyển sở hữu hay hoàn thành một loạt các bước chuyển robot) vốn xuất hiện bên trong hoàn cảnh vận hành hệ thống
- *Vai trò* (như người quản lí, kĩ sư, người bán hàng) do những người có tương tác với hệ thống đảm nhiệm.
- *Đơn vị tổ chức* (phòng, nhóm, đội) có liên quan tới một ứng dụng
- *Nơi chốn* (tầng sản xuất hay nơi dỡ hàng) tạo nên hoàn cảnh của vấn đề và toàn bộ chức năng của hệ thống.
- *Cấu trúc* (bộ cảm biến, xe cộ bốn bánh hay máy tính) xác định ra lớp các sự vật hay các lớp sự vật có liên quan.

Cũng cần chú ý tới cái không phải là sự vật. Nói chung, một sự vật không bao giờ có “tên thủ tục ra lệnh”. Chẳng hạn, nếu người phát triển phần mềm cho hệ thống tạo ảnh thuộc định nghĩa một sự vật với tên **đạo ảnh**, thì họ có thể phạm phải một sai lầm tinh vi. **Ảnh** thu được từ phần mềm có thể, tất nhiên, là một sự vật (đó là sự vật vốn là một phần của miền thông tin). **Đạo** một ảnh là một thao tác được áp dụng cho sự vật. Rất có thể là việc đạo sẽ

được xác định như một thao tác cho sự vật **ảnh**, nhưng nó không thể được định nghĩa như một sự vật tách biệt bao hàm “đảo ảnh.” Như Cashman đã nói: “... nội dung của hướng sự vật là để bao bọc, nhưng vẫn phải giữ phân tách, dữ liệu và các thao tác trên dữ liệu.”

Coad và Yourdon gợi ý ra sáu đặc trưng chọn lựa nên sử dụng khi người phân tích xem xét từng sự vật tiềm năng để đưa vào trong mô hình phân tích :

- *Thông tin được sử dụng.* Sự vật tiềm năng sẽ có ích trong khi phân tích chỉ nếu thông tin về nó phải được ghi nhớ sao cho hệ thống có thể vận hành.
- *Dịch vụ cần tới.* Sự vật tiềm năng phải có một tập các thao tác xác định được mà có thể làm thay đổi giá trị của các thuộc tính của nó theo cách nào đó.
- *Đa thuộc tính.* Trong khi phân tích yêu cầu, mỗi quan tâm phải dồn vào thông tin “chính”; một sự vật với một thuộc tính trong thực tế có thể có ích khi thiết kế nhưng có lẽ sẽ được biểu diễn tốt hơn như một thuộc tính của một sự vật khác trong hoạt động phân tích.
- *Thuộc tính thông thường.* Tập hợp các thuộc tính có thể được định nghĩa cho sự vật tiềm năng và những thuộc tính này áp dụng cho mọi lần xuất hiện của sự vật.
- *Thao tác thông thường.* Một tập các thao tác có thể được định nghĩa cho sự vật tiềm năng và những thao tác này áp dụng cho mọi lần xuất hiện của sự vật.
- *Các yếu cầu bản chất.* Các thực thể ngoài xuất hiện trong không gian vấn đề và tạo ra hay tiêu thụ thông tin là bản chất cho việc tiến hành bất kì giải pháp nào cho hệ thống gần như bao giờ cũng được định nghĩa như các sự vật trong mô hình yêu cầu.

Để được xem xét như một sự vật hợp pháp để đưa vào trong mô hình yêu cầu, một sự vật tiềm năng phải thoả mãn tất cả (hay gần như tất cả) các đặc trưng trên. Việc quyết định đưa các sự vật

tiềm năng vào trong mô hình phân tích là chủ quan theo nghĩa nào đó, và việc đánh giá về sau có thể làm cho một sự vật bị loại bỏ hay thiết lập lại. Tuy nhiên, bước đầu tiên của OOA phải là một định nghĩa về các sự vật, và phải tiến hành quyết định đưa sự vật vào mô hình (dù chủ quan).

XÁC ĐỊNH THUỘC TÍNH

Các thuộc tính mô tả cho một sự vật đã được chọn lựa để đưa vào trong mô hình phân tích. Về bản chất, chính thuộc tính xác định ra sự vật - làm rõ rệt sự vật mang ý nghĩa gì trong hoàn cảnh của không gian vấn đề. Chẳng hạn, nếu chúng ta định xây dựng một hệ thống theo dõi kết quả bóng chày cho các cầu thủ bóng chày chuyên nghiệp thì các thuộc tính của sự vật **player** sẽ hoàn toàn khác với các thuộc tính của cùng sự vật đó khi được dùng trong hoàn cảnh hệ thống trả lương bóng chày chuyên nghiệp. Trong hoàn cảnh đấu, các thuộc tính như tên, vị trí, trung bình đánh bóng, phần trên sân, năm thi đấu, số trận đã chơi đều có thể có liên quan. Với hoàn cảnh sau, một số trong các thuộc tính này sẽ có nghĩa, nhưng số khác sẽ bị thay thế bởi các thuộc tính như lương trung bình, tiền đặt cọc giao trước, tiền trợ cấp hàng tháng dự kiến, địa chỉ gửi thư v.v..

Để phát triển một tập có ý nghĩa các thuộc tính cho một sự vật, người phân tích lại có thể nghiên cứu lời thuật xử lý (hay phát biểu về phạm vi) cho vấn đề và chọn lựa những điều có thể “thuộc về” sự vật một cách hợp lý. Bên cạnh đó, cần phải trả lời các câu hỏi sau cho từng sự vật: Khoản mục dữ liệu nào (hợp thành và/hoặc sơ cấp) xác định hoàn toàn sự vật này trong hoàn cảnh của vấn đề trong tay?

XÁC ĐỊNH THAO TÁC

Một thao tác làm thay đổi một sự vật theo cách nào đó. Đặc biệt hơn, nó làm thay đổi một hay nhiều giá trị thuộc tính chứa

trong sự vật. Do đó, một thao tác phải có “tri thức” về bản chất của thuộc tính và phải được cài đặt theo cách thức làm cho nó thao tác được cấu trúc dữ liệu đã được suy dẫn từ thuộc tính.

Mặc dầu có nhiều kiểu thao tác tồn tại, nói chung có thể chia chúng thành ba loại chính: (1) các thao tác *tác động* lên dữ liệu theo một cách nào đó (như thêm, xoá, định dạng lại, chọn); (2) các thao tác thực hiện một *tính toán*; (3) các thao tác *điều phối* một sự vật cho việc xuất hiện của sự kiện điều khiển.

Xem như “nhất cắt đầu tiên” suy ra tập các thao tác cho các sự vật của mô hình phân tích, người phân tích lại có thể nghiên cứu lời thuật xử lí (hay phát biểu về phạm vi) về vấn đề và chọn ra những thao tác “thuộc” một cách hợp lí vào sự vật. Để thực hiện điều này, lại phải nghiên cứu về phân tích văn phạm và cô lập ra các động từ. Một số động từ này sẽ là các thao tác hợp pháp và có thể dễ dàng được nối vào một sự vật xác định.

Bên cạnh việc phân tích văn phạm, chúng ta có thể thu được cái nhìn phụ vào các thao tác khác bằng cách xem xét việc *truyền thông* xuất hiện giữa các sự vật. Các sự vật truyền thông bằng cách truyền thông báo lẫn nhau. Trước khi tiếp tục với đặc tả thao tác chúng ta tìm hiểu vấn đề này thêm một chút chi tiết.

TRUYỀN THÔNG GIỮA CÁC SỰ VẬT

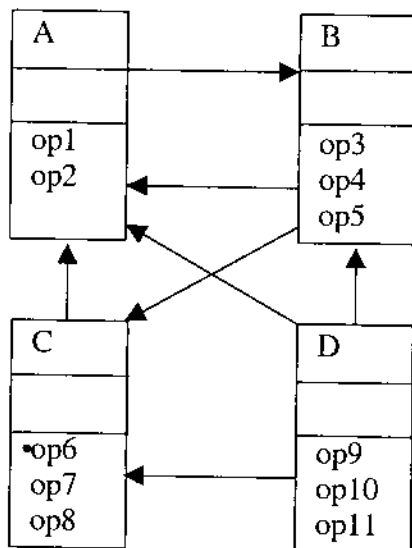
Xác định các sự vật trong hoàn cảnh của mô hình phân tích có thể là đủ để đặt nền tảng cho thiết kế. Nhưng vẫn còn phải thêm một cái gì đó khác (hoặc trong khi phân tích hoặc trong khi thiết kế) để có thể xây dựng được hệ thống - phải thiết lập một cơ chế truyền thông giữa các sự vật. Cơ chế này được gọi là *thông báo*.

Việc dùng các thông báo được minh hoạ theo sơ đồ trong Hình B.3. Với các sự vật A, B, C và D, truyền thông giữa chúng được thực hiện thông qua việc truyền thông báo. Chẳng hạn, nếu sự

vật B cần xử lí liên quan tới thao tác op10 của sự vật D thì nó sẽ truyền cho D một thông báo có dạng:

thông báo: (nơi nhận, thao tác, các đối)

với “nơi nhận” xác định ra sự vật (trong trường hợp này là sự vật D) để nhận thông báo, “thao tác” nói tới thao tác chuẩn bị nhận thông báo (op10), còn “đối” cung cấp thông tin cần cho thao tác thực hiện thành công. Xem như một phần của việc thực hiện của op10, sự vật D có thể gửi một thông báo cho sự vật C dưới dạng:



Hình B.3 Truyền thông báo.

thông báo: (C, op8, <dữ liệu>)

C tìm thấy op08, thực hiện nó và rồi trả lại điều khiển cho D. Thao tác op10 hoàn thành và điều khiển trả về cho B.

Cox mô tả sự trao đổi giữa các sự vật theo cách sau:

Một sự vật được yêu cầu thực hiện một trong những thao tác của nó bằng cách gửi cho nó một thông báo cho sự vật này điều phải làm. Nơi nhận [sự vật] đáp ứng cho thông báo bằng cách đầu tiên chọn thao tác ứng với tên thao tác trong thông báo, thực hiện thao tác này và rồi trả lại điều khiển cho nơi gọi.

Thông báo là quan trọng cho việc cài đặt hệ thống hướng sự vật, nhưng không cần phải xét tới về chi tiết trong khi phân tích yêu cầu. Trong thực tế, mối quan tâm duy nhất của ta tại giai đoạn này

là dùng khái niệm này để giúp xác định các thao tác ứng cử viên cho một sự vật xác định.

HOÀN TẤT ĐỊNH NGHĨA SỰ VẬT

Định nghĩa về thao tác là bước cuối trong việc hoàn thành đặc tả cho một sự vật. Chúng ta chọn lựa các thao tác từ việc phân tích văn phạm của lời thuật xử lý cho hệ thống. Các thao tác phụ có thể được xác định bằng cách xem xét “lịch sử cuộc đời” của một sự vật và những thông báo được truyền qua các sự vật được xác định cho hệ thống.

Lịch sử cuộc đời tổng quát của một sự vật có thể được xác định bằng cách nhận ra rằng sự vật phải được tạo ra, sửa đổi, thao tác hay đọc vào theo cách khác và có thể bị xoá đi. Với sự vật **hệ thống**, điều này có thể được mở rộng để phản ánh hoạt động đã biết xuất hiện trong cuộc đời của nó. Một số thao tác có thể được xác quyết từ việc truyền thông có thể có giữa các sự vật.

B.2 MÔ HÌNH HOÁ PHÂN TÍCH HƯỚNG SỰ VẬT

Một số sơ đồ mô hình khác nhau cho OOA đã được nhiều người gọi ý. Tất cả đều dùng định nghĩa sự vật đã được trình bày trong Mục B.1, nhưng mỗi người lại đưa ra một kí pháp riêng, theo kinh nghiệm trực giác và một triết lí riêng. Để minh hoạ cho cách tiếp cận mô hình OOA, chúng ta sẽ dùng kí pháp của Ngôn ngữ mô hình hoá thống nhất UML.

B.2.1 TỔNG QUAN VỀ UML

Ngôn ngữ mô hình hoá thống nhất UML có một phổ sử dụng rộng rãi. Nó có thể được dùng cho việc mô hình hoá nghiệp vụ, mô hình hoá phần mềm trong mọi pha của việc phát triển và với mọi kiểu hệ thống, và việc mô hình hoá tổng quát cho bất kì kết cấu nào vốn có cả cấu trúc tĩnh và hành vi động. Để đạt tới khả năng rộng rãi này, ngôn ngữ này được định nghĩa đủ mang tính mở rộng và tổng quát để cho phép mô hình hoá các hệ thống đa dạng như vậy, tránh việc quá đặc thù hay quá phức tạp.

Mục này đưa ra một tổng quan về UML để biểu lộ phạm vi và cấu trúc của nó. Các phần tử của ngôn ngữ này được mô tả vắn tắt ở đây; những giải thích sâu sắc hơn và các chi tiết được trình bày trong cuốn *UML Toolkit* của Hans-Erik Eriksson và Magnus Penker. Độc giả không nhất thiết phải hiểu đầy đủ các biểu đồ được vẽ trong phần này, thay vì thế chỉ để có một thoáng nhìn về UML và những khả năng của nó.

Bản tổng quan này mô tả các phần khác nhau của UML:

- *Cái nhìn*: Cái nhìn chỉ ra các khía cạnh khác nhau của hệ thống được mô hình hoá. Cái nhìn không phải là một đồ thị, mà là sự trừu tượng bao gồm một số biểu đồ. Chỉ bằng việc định nghĩa một số cái nhìn, mỗi cái nhìn chỉ ra một khía cạnh đặc biệt của hệ thống, thì bức tranh đầy đủ về hệ thống mới được xây dựng lên. Các cái nhìn cũng móc nối ngôn ngữ mô hình với phương pháp/tiến trình được chọn cho việc phát triển.
- *Biểu đồ*: Biểu đồ là các đồ thị mô tả cho nội dung trong một cái nhìn. UML có chín kiểu biểu đồ khác nhau được dùng một cách tổ hợp để cung cấp mọi cái nhìn về hệ thống.
- *Phần tử mô hình*: Các khái niệm được dùng trong các biểu đồ đều là các phần tử mô hình vốn biểu thị cho các khái niệm hướng sự vật chung như lớp, sự vật và thông báo, và mối quan

hệ giữa các khái niệm đó bao gồm liên kết, tính phụ thuộc và tính tổng quát hoá. Một phần tử mô hình được dùng trong nhiều biểu đồ khác nhau, nhưng nó bao giờ cũng có cùng ý nghĩa và kí hiệu.

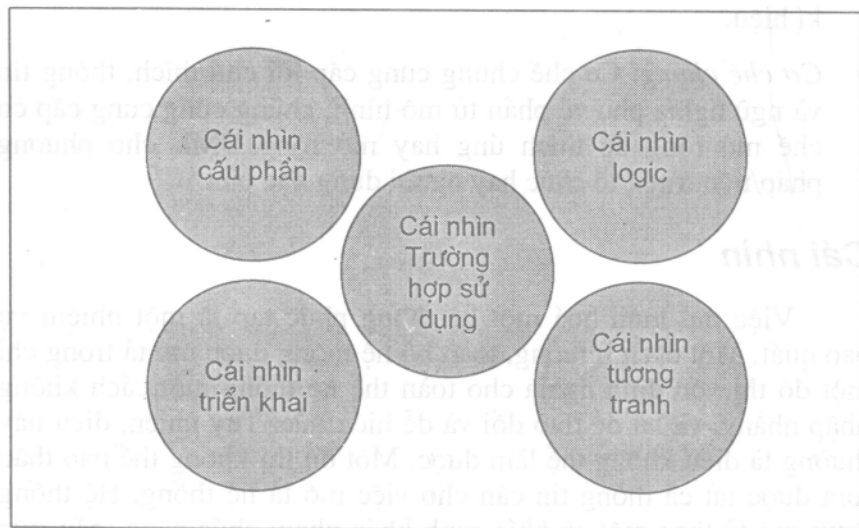
- *Cơ chế chung*: Cơ chế chung cung cấp lời chú thích, thông tin và ngữ nghĩa phụ về phần tử mô hình; chúng cũng cung cấp cơ chế mở rộng để thích ứng hay mở rộng UML cho phương pháp/tiến trình, tổ chức hay người dùng đặc biệt.

Cái nhìn

Việc mô hình hoá một hệ thống phức tạp là một nhiệm vụ bao quát. Một cách lí tưởng, toàn bộ hệ thống được mô tả trong chỉ một đồ thị vốn định nghĩa cho toàn thể hệ thống một cách không nhập nhằng, và lại dễ trao đổi và dễ hiểu nữa. Tuy nhiên, điều này thường là điều không thể làm được. Một đồ thị không thể nào thu tóm được tất cả thông tin cần cho việc mô tả hệ thống. Hệ thống được mô tả theo một số khía cạnh khác nhau: chức năng (cấu trúc tĩnh và tương tác động của nó), phi chức năng (yêu cầu thời gian, độ tin cậy, cách triển khai, v.v..) và khía cạnh tổ chức (tổ chức công việc, tương ứng với các modul mã, v.v..). Vậy hệ thống được mô tả theo một số cái nhìn, với mỗi cái nhìn biểu thị cho một cách phóng chiếu mô tả hệ thống đầy đủ, chỉ ra một khía cạnh đặc biệt của hệ thống.

Mỗi cái nhìn lại được mô tả theo một số các biểu đồ có chứa thông tin vốn nhấn mạnh vào một khía cạnh đặc thù của hệ thống. Cũng hơi có sự chòem lấp lên nhau, để cho biểu đồ thực tế có thể là một phần của nhiều cái nhìn. Bằng cách nhìn vào hệ thống từ các cái nhìn khác nhau, có thể tập trung vào một khía cạnh của hệ thống mỗi lúc. Biểu đồ trong một cái nhìn đặc biệt nên đủ đơn giản để dễ trao đổi, nhưng vẫn nhất quán với các biểu đồ và cái nhìn khác (qua các biểu đồ tương ứng của chúng). Biểu đồ chứa các kí

hiệu đồ hoạ biểu diễn cho các phần tử mô hình của hệ thống. Hình B.4 chỉ ra các cái nhìn của UML. Chúng là :



Hình B.4 Cái nhìn trong UML.

- *Cái nhìn trường hợp sử dụng*: Cái nhìn biểu lộ chức năng của hệ thống như tác nhân ngoài cảm nhận.
- *Cái nhìn logic*: Cái nhìn biểu lộ cách thức chức năng được thiết kế bên trong hệ thống, dưới dạng cấu trúc tĩnh và hành vi động của hệ thống.
- *Cái nhìn cấu phần*: Cái nhìn biểu lộ việc tổ chức cấu phần mã.
- *Cái nhìn tương tranh*: Cái nhìn biểu lộ sự tương tranh trong hệ thống, đề cập tới vấn đề với truyền thông và đồng bộ hoá vốn có mặt trong hệ thống tương tranh.
- *Cái nhìn triển khai*: Cái nhìn biểu lộ cách triển khai hệ thống vào trong kiến trúc vật lí với máy tính và các thiết bị được gọi là nút.

Khi bạn chọn một dụng cụ để vẽ biểu đồ, hãy chắc nó là dụng cụ tạo dễ dàng cho việc chuyển từ cái nhìn này sang cái nhìn khác. Bên cạnh đó, để thấy cách chức năng được thiết kế làm việc bên trong một biểu đồ, dụng cụ này nên làm dễ dàng cho việc chuyển hoặc sang cái nhìn trường hợp sử dụng để người dùng ngoài xem chức năng này được mô tả như thế nào hay sang cái nhìn triển khai để xem chức năng này được phân bố thế nào trong cấu trúc vật lí. Nói cách khác, tìm ra máy tính nào là có sẵn.

Lưu ý rằng các cái nhìn khác có thể được dùng bao gồm cái nhìn tĩnh-động, logic-vật lí, luồng công việc, và những cái nhìn khác. UML không yêu cầu rằng những cái nhìn này phải được dùng, nhưng chúng là những cái nhìn mà người thiết kế UML có trong tâm trí, cho nên rất có thể là phần lớn các dụng cụ ấy sẽ dựa trên những cái nhìn này.

CÁI NHÌN TRƯỜNG HỢP SỬ DỤNG

Cái nhìn trường hợp sử dụng mô tả chức năng mà hệ thống cần chuyển giao, như các tác nhân ngoài cảm nhận. Tác nhân tương tác với hệ thống; đây có thể là người dùng hay hệ thống khác. Cái nhìn trường hợp sử dụng dành cho khách hàng, người thiết kế, người phát triển, và người kiểm thử; nó được mô tả trong các biểu đồ trường hợp sử dụng và đôi khi trong biểu đồ hoạt động. Việc dùng mong muốn của hệ thống được mô tả như một số các trường hợp sử dụng trong cái nhìn trường hợp sử dụng, nơi một trường hợp sử dụng là một mô tả chung về việc dùng hệ thống (một chức năng được yêu cầu).

Cái nhìn trường hợp sử dụng là trung tâm, vì nội dung của nó dẫn hướng sự phát triển của các cái nhìn khác. Mục tiêu cuối cùng của hệ thống là cung cấp chức năng được mô tả theo cái nhìn này - cùng với một số tính chất phi chức năng - do đó cái nhìn này ảnh hưởng tới tất cả các cái nhìn khác. Cái nhìn này cũng được dùng để làm hợp lệ và cuối cùng kiểm chứng hệ thống bằng việc kiểm thử

cái nhìn trường hợp sử dụng với khách hàng (bằng việc hỏi "Đây có phải là điều bạn muốn không?") và với hệ thống cuối cùng (bằng việc hỏi "Hệ thống này có làm việc như đã được xác định không?").

CÁI NHÌN LOGIC

Cái nhìn logic mô tả cách cung cấp chức năng hệ thống. Nó chủ yếu được dành cho người thiết kế và người phát triển. Tương phản với cái nhìn trường hợp sử dụng, cái nhìn logic nhìn vào bên trong hệ thống. Nó mô tả cho cả cấu trúc tĩnh (lớp, sự vật, và mối quan hệ) và sự hợp tác động vốn xuất hiện khi các sự vật gửi thông báo lẫn nhau để cung cấp một chức năng nào đó. Các tính chất như sự bền vững và tương tranh cũng được định nghĩa, cũng như giao diện và cấu trúc nội bộ của lớp đó.

Cấu trúc tĩnh được mô tả trong các biểu đồ lớp và sự vật. Việc mô hình hoá động được mô tả theo các biểu đồ trạng thái, trình tự, hợp tác và hoạt động.

CÁI NHÌN CẤU PHẦN

Cái nhìn cấu phần là một mô tả về các modul cài đặt và sự phụ thuộc của chúng. Nó chủ yếu dành cho người phát triển, và bao gồm biểu đồ cấu phần. Các cấu phần, vốn là các kiểu modul mã khác nhau, được vẽ với cấu trúc và sự phụ thuộc của chúng. Thông tin phụ về các cấu phần, như cấp phát tài nguyên (trách nhiệm cho một cấu phần), hay thông tin quản trị khác, như báo cáo tiến độ cho công việc phát triển, cũng có thể được thêm vào.

CÁI NHÌN TƯƠNG TRANH

Cái nhìn tương tranh giải quyết cho việc phân chia hệ thống thành các tiến trình và bộ xử lí. Khía cạnh này, vốn là tính chất phi chức năng của hệ thống, cho phép sử dụng tài nguyên hiệu quả, thực hiện song song, và giải quyết các sự kiện không đồng bộ từ môi trường. Bên cạnh việc phân chia hệ thống ra thành các mạch

điều khiển thực hiện tương tranh, cái nhìn này phải giải quyết với việc truyền thông và đồng bộ hoá các mạch đó.

Cái nhìn tương tranh được dành cho người phát triển và người tích hợp hệ thống, và nó bao gồm các biểu đồ động (biểu đồ trạng thái, trình tự, hợp tác và hoạt động) và biểu đồ cài đặt (biểu đồ cấu phần và triển khai).

CÁI NHÌN TRIỂN KHAI

Cuối cùng, cái nhìn triển khai biểu lộ cách triển khai vật lý của hệ thống, như máy tính và thiết bị (nút) và cách chúng nối lẫn nhau. Cái nhìn triển khai được dành cho người phát triển, người tích hợp, và người kiểm thử, và được biểu thị bằng biểu đồ triển khai. Cái nhìn này cũng bao gồm một ánh xạ chỉ ra cách các cấu phần được triển khai trong kiến trúc vật lý; chẳng hạn, chương trình hay sự vật nào thực hiện trên máy tính tương ứng.

Biểu đồ

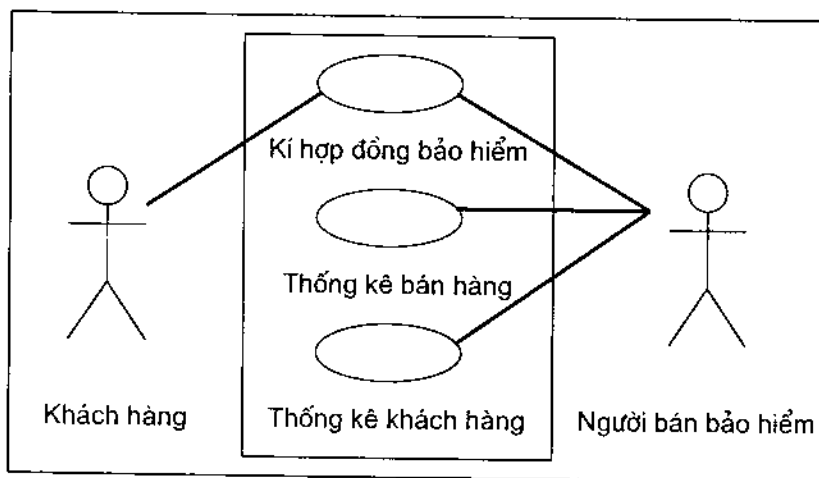
Biểu đồ là các đồ thị thực tế chỉ ra các kí hiệu phần tử mô hình được bố trí để minh hoạ cho một phần hay khía cạnh đặc biệt của hệ thống. Mô hình hệ thống về cơ bản có nhiều biểu đồ thuộc từng kiểu. Biểu đồ là một phần của cái nhìn đặc biệt; và khi nó được vẽ ra, thì nó thường được cấp phát cho một cái nhìn. Một số kiểu biểu đồ có thể là bộ phận của nhiều cái nhìn, tùy theo nội dung của biểu đồ đó.

Mục này mô tả cho các khái niệm cơ bản đằng sau từng biểu đồ. Các biểu đồ được lấy từ các kiểu hệ thống khác nhau để chỉ ra tính đa dạng của UML.

BIỂU ĐỒ TRƯỜNG HỢP SỬ DỤNG

Biểu đồ trường hợp sử dụng chỉ ra một số tác nhân ngoài và mối nối của họ với các trường hợp sử dụng mà hệ thống cung cấp

(xem Hình B.5). Trường hợp sử dụng là một mô tả về một chức năng (một cách dùng đặc biệt của hệ thống) mà hệ thống đó cung cấp. Mô tả này về trường hợp sử dụng thực tế thông thường được tiến hành trong văn bản như một tính chất tài liệu của kí hiệu trường hợp sử dụng, nhưng nó cũng có thể được mô tả bằng việc dùng một biểu đồ hoạt động. Các trường hợp sử dụng được mô tả chỉ như được tác nhân ngoài xem xét bên ngoài (hành vi hệ thống như người dùng cảm nhận nó), và không mô tả cách chức năng này được cung cấp bên trong hệ thống. Các trường hợp sử dụng xác định các yêu cầu chức năng của hệ thống.



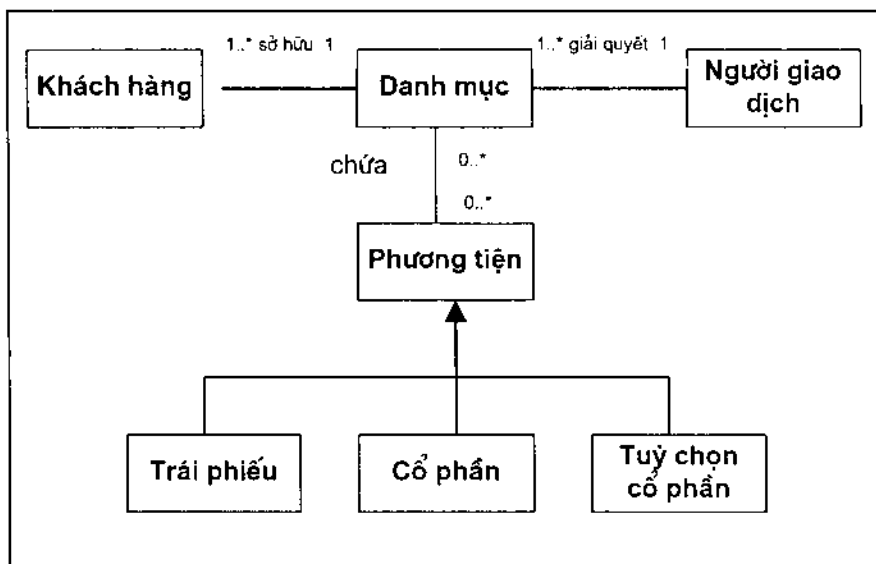
Hình B.5 Biểu đồ trường hợp sử dụng cho kinh doanh bảo hiểm.

BIỂU ĐỒ LỚP

Biểu đồ lớp chỉ ra cấu trúc tĩnh của các lớp trong hệ thống (xem Hình B.6). Các lớp biểu thị cho "sự vật" vốn được giải quyết trong hệ thống. Các lớp có thể có quan hệ với nhau theo một số cách: liên kết (được nối lẫn nhau), phụ thuộc (lớp này phụ thuộc/dùng lớp kia), đặc biệt hoá (lớp này là đặc biệt hoá của lớp

kia), hay được đóng gói (được gộp nhóm lại như một đơn vị). Tất cả những mối quan hệ này đều được vẽ trong biểu đồ lớp cùng với cấu trúc nội bộ của các lớp dưới dạng thuộc tính và phép toán. Biểu đồ này được coi là tĩnh trong đó cấu trúc được mô tả bao giờ cũng hợp lệ tại bất kì điểm nào trong vòng đời hệ thống.

Một hệ thống về cơ bản có một số biểu đồ lớp - không phải tất cả các lớp đều được đưa vào trong một biểu đồ lớp - và một lớp có thể tham gia vào nhiều biểu đồ lớp.

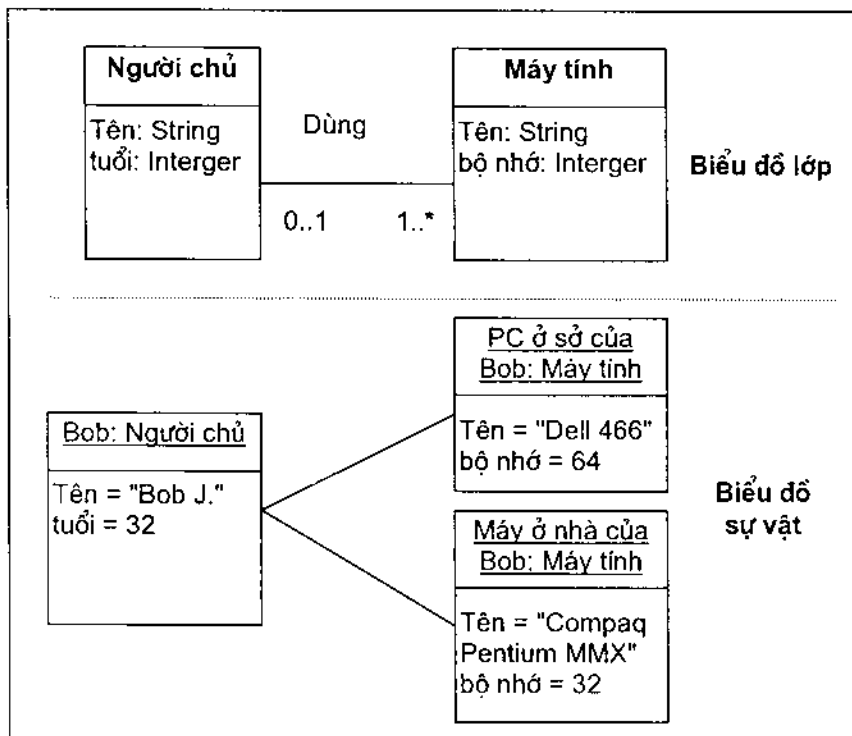


Hình B.6 Biểu đồ lớp cho giao dịch tài chính.

BIỂU ĐỒ SỰ VẬT

Biểu đồ sự vật là một biến thể của biểu đồ lớp và dùng kí pháp gần như đồng nhất. Sự khác biệt giữa hai biểu đồ này là ở chỗ biểu đồ sự vật chỉ ra một số thể nghiệm sự vật của lớp, thay vì là lớp thực tại. Biểu đồ sự vật do vậy là một thí dụ của biểu đồ lớp vốn

chỉ ra bức ảnh chụp nhanh có thể của việc thực hiện hệ thống - điều hệ thống có thể trông thế nào tại điểm nào đó. Cùng kí pháp cho biểu đồ lớp cũng được dùng, với hai ngoại lệ: các sự vật được viết với tên có gạch dưới và tất cả các thể nghiệm trong quan hệ được vẽ. (Xem Hình B.7)



Hình B.7 Biểu đồ lớp và biểu đồ sự vật .

Biểu đồ sự vật không quan trọng bằng biểu đồ lớp, nhưng chúng có thể được dùng để minh họa cho một biểu đồ lớp phức tạp bằng cách chỉ ra các thể nghiệm và mối quan hệ thực tại có thể trông thế nào. Biểu đồ sự vật cũng được dùng như một phần của biểu đồ hợp tác, trong đó vẽ ra sự hợp tác động giữa các sự vật.

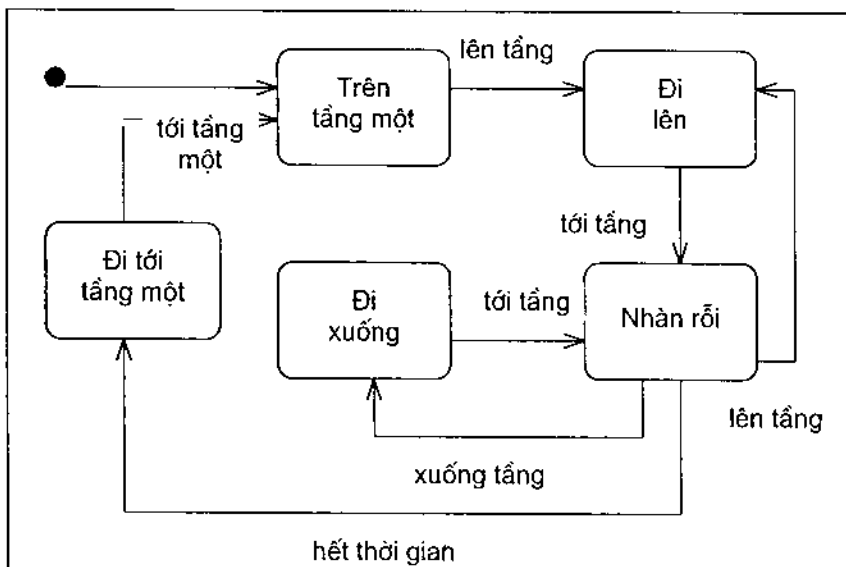
BIỂU ĐỒ TRẠNG THÁI

Biểu đồ trạng thái về cơ bản là phần bổ sung cho việc mô tả lớp. Nó chỉ ra tất cả trạng thái có thể mà sự vật thuộc lớp đó có thể có, và những sự kiện nào gây ra việc đổi trạng thái (xem Hình B.8). Một biến cố có thể là một sự vật khác vốn gửi một thông báo tới cho nó - chẳng hạn, một khoảng thời gian xác định đã trôi qua - hay có thể là một điều kiện nào đó đã được hoàn thành. Việc thay đổi trạng thái được gọi là việc *chuyển*. Việc chuyển cũng có thể có một hành động gắn với nó vốn xác định ra điều cần phải làm có liên quan tới việc chuyển trạng.

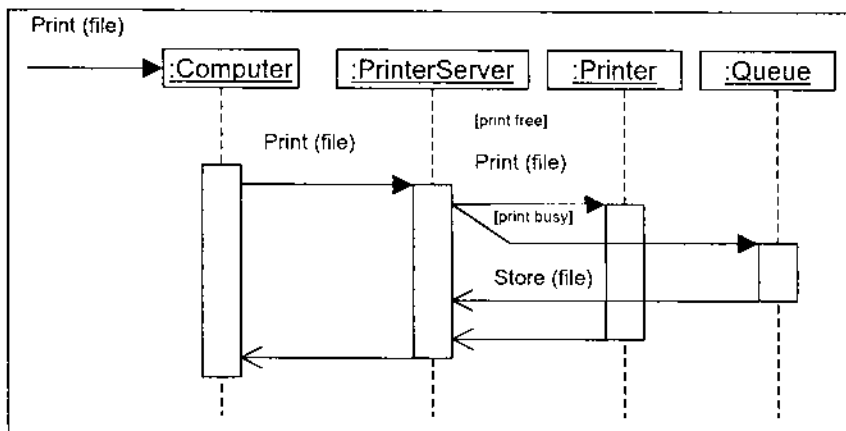
Biểu đồ trạng thái không được vẽ cho mọi lớp, chỉ vẽ cho những lớp có một số trạng thái đã xác định rõ và nơi mà hành vi của lớp đó bị ảnh hưởng và thay đổi bởi các trạng thái khác nhau. Biểu đồ trạng thái cũng có thể được vẽ cho toàn thể hệ thống.

BIỂU ĐỒ TRÌNH TỰ

Biểu đồ trình tự chỉ ra sự hợp tác động giữa một số các sự vật, như được vẽ trong hình B.9. Khía cạnh quan trọng của biểu đồ này là để chỉ ra trình tự các thông báo được gửi giữa các sự vật. Nó cũng chỉ ra tương tác giữa các sự vật, điều gì đó xảy ra tại một điểm xác định trong thực hiện hệ thống. Biểu đồ này bao gồm một số các sự vật được chỉ ra bằng đường vạch đứng. Thời gian được tính tới trong biểu đồ này, và biểu đồ chỉ ra trao đổi thông báo giữa các sự vật khi thời gian trôi qua theo trình tự hay chức năng. Thông báo được vẽ như các đường với mũi tên thông báo giữa các đường sự vật đứng. Thời gian và các chú thích được viết ở lề của biểu đồ.



Hình B.8 Biểu đồ trạng thái cho thang máy.



Hình B.9 Biểu đồ trình tự cho nguồn phục vụ in.

BIỂU ĐỒ HỢP TÁC

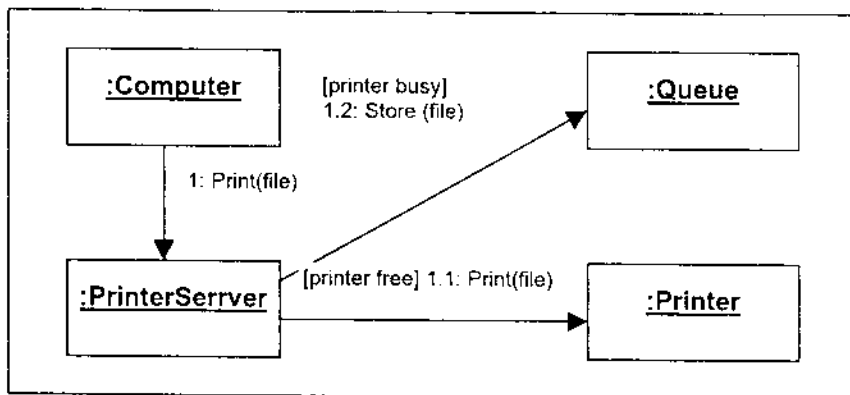
Biểu đồ hợp tác chỉ ra sự hợp tác động, cũng giống như biểu đồ trình tự. Thường thì có việc chọn lựa giữa việc biểu thị sự hợp tác như hoặc biểu đồ trình tự hoặc như biểu đồ hợp tác. Bên cạnh việc chỉ ra sự trao đổi thông báo (được gọi là *tương tác*), biểu đồ hợp tác chỉ ra các sự vật và mối quan hệ giữa chúng (đôi khi còn nói tới như *hoàn cảnh*). Việc dùng biểu đồ trình tự hay biểu đồ hợp tác có thể được quyết định bởi: Nếu thời gian hay trình tự là khía cạnh quan trọng nhất cần nhấn mạnh, thì chọn biểu đồ trình tự; nếu hoàn cảnh là quan trọng cần nhấn mạnh thì chọn biểu đồ hợp tác. Tương tác giữa các sự vật được biểu thị trong cả hai biểu đồ này.

Biểu đồ hợp tác được vẽ như biểu đồ sự vật, trong đó một số sự vật được vẽ cùng với mối quan hệ của chúng (bằng việc dùng kí pháp của biểu đồ lớp/sự vật). Mũi tên thông báo được vẽ giữa các sự vật để chỉ ra luồng thông báo giữa các sự vật. Các nhãn được đặt lên các thông báo, mà trong số các điều khác, chỉ ra trật tự theo đó thông báo được gửi. Nó cũng có thể biểu thị cho điều kiện, lặp, giá trị cho lại vân vân. Khi quen thuộc với cú pháp nhãn thông báo thì người phát triển có thể đọc sự hợp tác và đi theo luồng thực hiện và sự trao đổi thông báo. Biểu đồ hợp tác cũng có thể chứa các sự vật tích cực, các sự vật thực hiện tương tranh với các sự vật tích cực khác (xem Hình B.10).

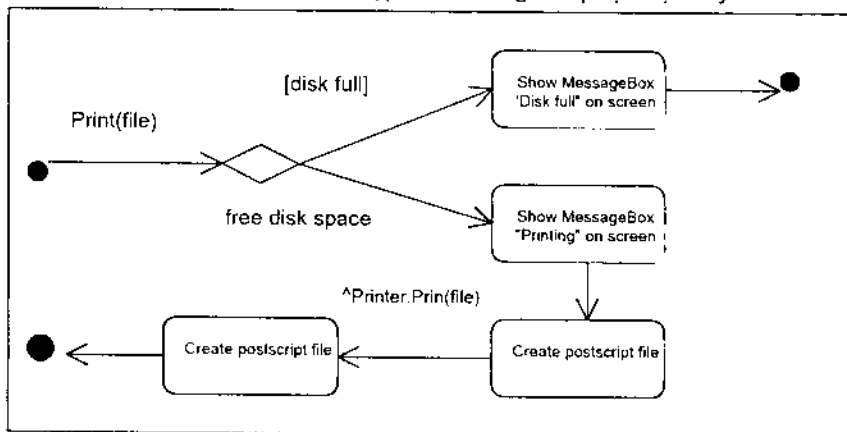
BIỂU ĐỒ HOẠT ĐỘNG

Biểu đồ hoạt động chỉ ra luồng trình tự các hoạt động, như được vẽ trong Hình B.11. Biểu đồ hoạt động về cơ bản được dùng để mô tả cho các hoạt động được thực hiện trong một thao tác, mặc dầu nó cũng có thể được dùng để mô tả cho các luồng hoạt động khác, như trường hợp sử dụng hay tương tác. Biểu đồ hoạt động bao gồm các trạng thái hành động, vốn bao gồm một đặc tả cho một hoạt động cần được thực hiện (một hành động). Một trạng thái hành động sẽ để lại trạng thái khi hành động đã được thực hiện (trạng

thái trong biểu đồ trạng thái cần một biến cố tường minh trước khi nó để lại trạng thái). Vậy điều khiển chảy giữa các trạng thái hành động, làm cho các trạng thái này được nối với nhau. Các quyết định và điều kiện, cũng như việc thực hiện song song của các trạng thái hành động, cũng có thể được vẽ trong biểu đồ này. Biểu đồ này cũng có thể chứa đặc tả về thông báo được gửi hay nhận xem như một phần của các hành động được thực hiện.



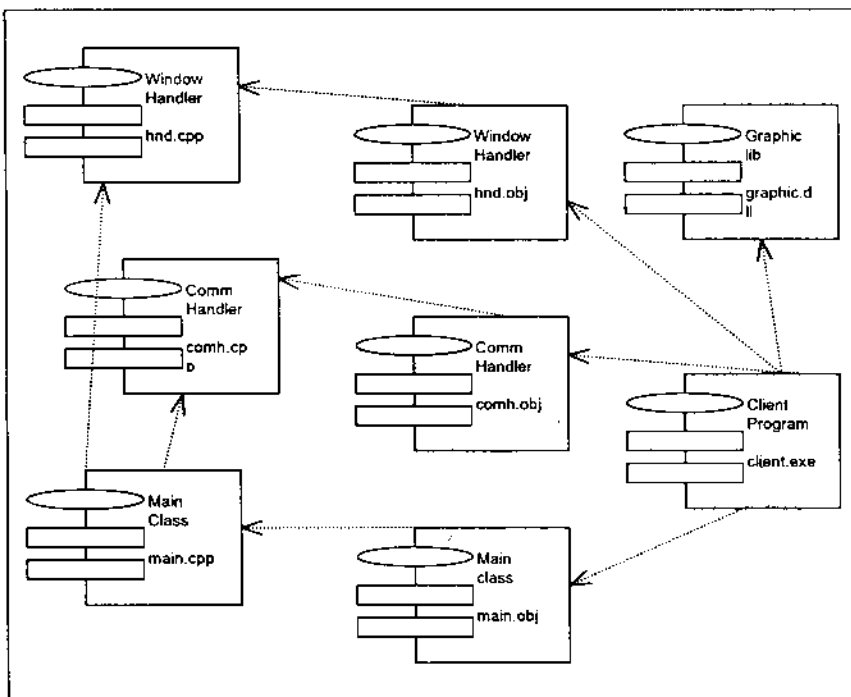
Hình B.10 Biểu đồ hợp tác cho nguồn phục vụ máy in.



Hình B.11 Biểu đồ hoạt động cho nguồn phục vụ máy in.

BIỂU ĐỒ CẤU PHẦN

Biểu đồ cấu phần chỉ ra cấu trúc vật lý của mã dưới dạng các cấu phần mã. Một cấu phần có thể là một cấu phần mã nguồn, cấu phần nhị phân, hay cấu phần thực hiện được. Cấu phần chứa thông tin về một hay nhiều lớp logic mà nó cài đặt, do vậy tạo ra một ánh xạ từ cái nhìn logic sang cái nhìn cấu phần. Sự phụ thuộc giữa các cấu phần được vẽ ra, làm cho dễ phân tích các cấu phần khác bị ảnh hưởng thế nào bởi việc thay đổi trong một cấu phần. Các cấu phần cũng có thể được vẽ với bất kì giao diện nào mà chúng để lộ ra, như giao diện OLE/COM; và chúng có thể được gộp nhóm với nhau trong các gói. Biểu đồ cấu phần được dùng trong công việc lập trình thực tế (xem Hình B.12).

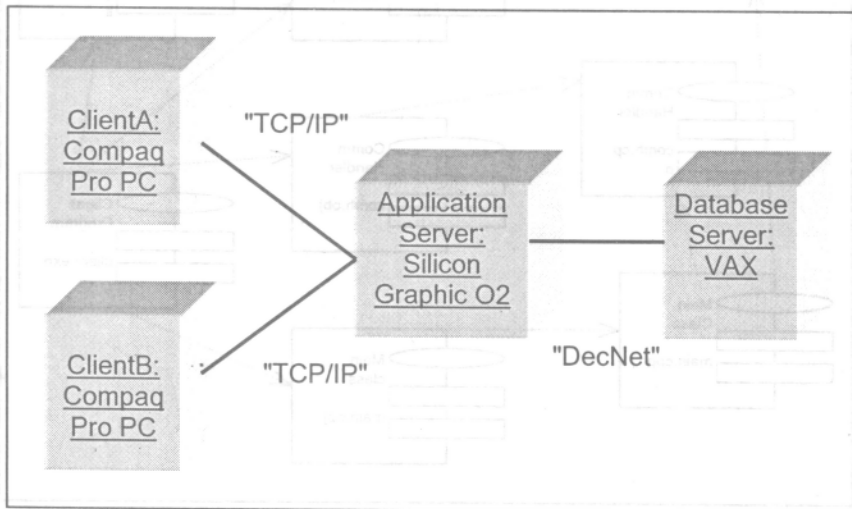


Hình B.12 Biểu đồ cấu phần chỉ ra sự phụ thuộc giữa cấu phần mã.

BIỂU ĐỒ TRIỂN KHAI

Biểu đồ triển khai chỉ ra kiến trúc vật lí của phần cứng và phần mềm trong hệ thống. Bạn có thể biểu thị các máy tính và thiết bị thực tế (các nút), cùng với các ghép nối giữa chúng; bạn cũng có thể biểu thị kiểu của ghép nối. Bên trong các nút, các cấu phần thực hiện được và các sự vật được cấp phát để chỉ ra các đơn vị phần mềm nào là thực hiện được trên nút nào. Bạn cũng có thể chỉ ra sự phụ thuộc giữa các cấu phần.

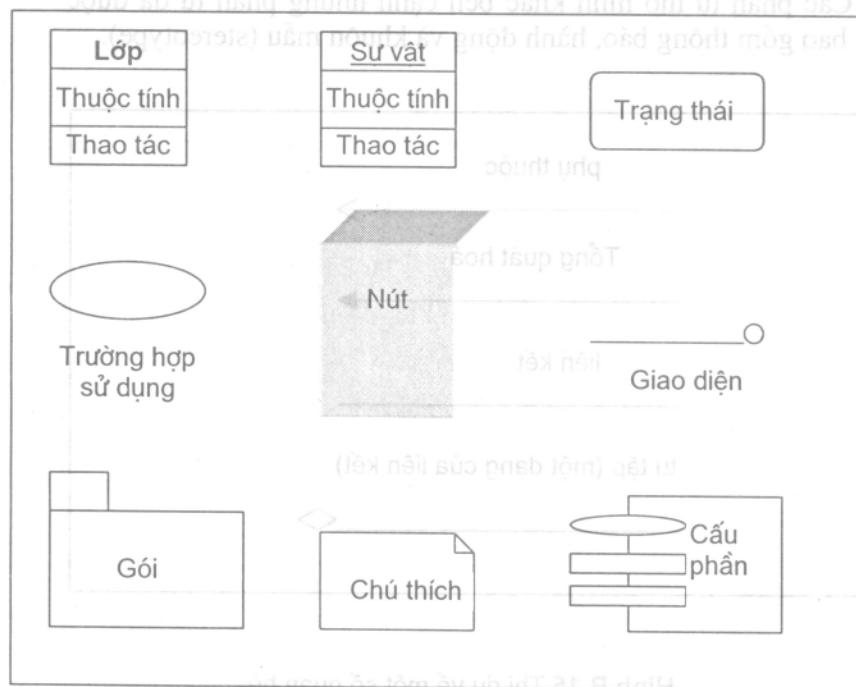
Như đã nói trước đây, biểu đồ triển khai, biểu thị cái nhìn triển khai, mô tả cho kiến trúc vật lí thực tế của hệ thống. Điều này xa với mô tả chức năng trong cái nhìn trường hợp sử dụng. Tuy nhiên, với một mô hình đã xác định rõ, có thể đi qua tất cả các đường từ một nút trong kiến trúc vật lí tới các cấu phần của nó tới các lớp nó cài đặt tới các tương tác mà các sự vật của lớp tham gia vào và cuối cùng tới trường hợp sử dụng. Các cái nhìn khác nhau về hệ thống được dùng để cho một mô tả nhất quán về hệ thống như một toàn thể (xem Hình B.13).



Hình B.13 Biểu đồ triển khai chỉ ra kiến trúc vật lí của hệ thống.

Phần tử mô hình

Các khái niệm được dùng trong các biểu đồ được gọi là phần tử mô hình. Một phần tử mô hình được định nghĩa bằng ngữ nghĩa, định nghĩa hình thức của phần tử đó hay đích xác ý nghĩa của điều nó biểu diễn trong các câu không nhập nhằng. Phần tử mô hình cũng có phần tử cái nhìn tương ứng, vốn là biểu diễn đồ hoạ cho phần tử đó hay kí hiệu đồ hoạ được dùng để biểu diễn cho phần tử này trong các biểu đồ. Một phần tử mô hình có thể tồn tại trong nhiều kiểu biểu đồ, nhưng có những quy tắc cho các phần tử có thể được vẽ trong từng kiểu biểu đồ. Một số phần tử mô hình điển hình là lớp, sự vật, trạng thái, nút, gói và cấu phần, như được vẽ trong Hình B.14.

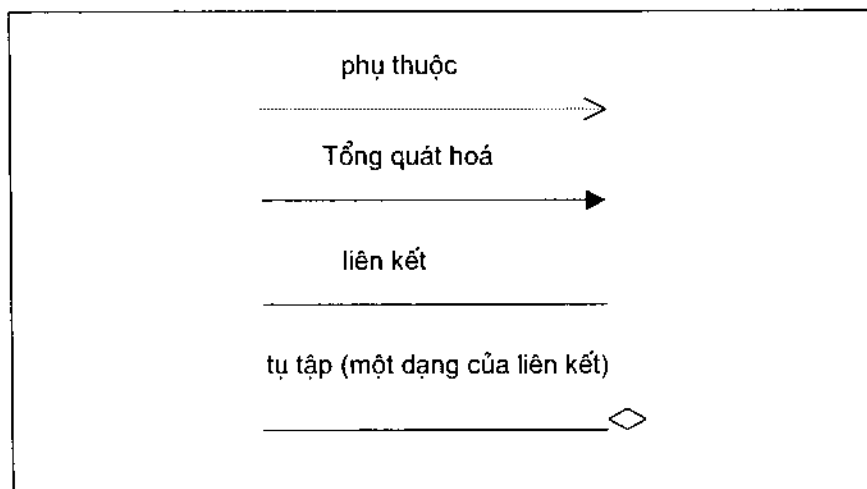


Hình B.14 Một số phần tử mô hình chung.

Hình B.15 vẽ ra các thí dụ về các quan hệ, mà cũng là phần tử mô hình, và chúng được dùng để nối các phần tử mô hình khác với nhau. Một số quan hệ khác nhau là:

- *Liên kết*: Ghép nối các phần tử và móc nối các thể nghiệm.
- *Tổng quát hoá*: Cũng còn được gọi là kế thừa, có nghĩa là một phần tử có thể là đặc biệt hoá của phần tử khác
- *Phụ thuộc*: Chỉ ra rằng phần tử này phụ thuộc theo cách nào đó vào phần tử khác.
- *Tụ tập*: Một hình thức của liên kết mà trong đó một phần tử chứa các phần tử khác.

Các phần tử mô hình khác bên cạnh những phần tử đã được mô tả bao gồm thông báo, hành động và khuôn mẫu (stereotype).



Hình B.15 Thí dụ về một số quan hệ.

Các cơ chế chung

UML sử dụng một số cơ chế chung trong tất cả các biểu đồ, dành cho thông tin phụ trong một biểu đồ, về cơ bản là những thông tin không thể biểu diễn được bằng việc dùng các khả năng cơ sở của phần tử mô hình.

TRANG TRÍ

Các trang trí đồ hoạ có thể được gắn vào các phần tử mô hình trong các biểu đồ. Các trang trí bổ sung thêm ngữ nghĩa cho từng phần tử. Thí dụ về trang trí là kĩ thuật được dùng để tách kiểu với thể nghiệm. Khi một phần tử biểu diễn cho một kiểu thì tên của nó được hiển thị bằng kiểu in đậm. Khi cùng phần tử đó biểu diễn cho thể nghiệm của kiểu này, thì tên của nó được gạch dưới và có thể xác định cả tên của thể nghiệm cũng như tên kiểu. Các trang trí khác là các đặc tả về tính nhiều quan hệ, với tính nhiều là số hay phạm vi chỉ ra bao nhiêu thể nghiệm của kiểu được ghép nối có thể được tham gia vào quan hệ. Các trang trí được viết gần với phần tử mà chúng bổ sung thêm thông tin. Tất cả các trang trí đều được mô tả nối tiếp với mô tả của phần tử chúng ảnh hưởng tới.

CHÚ THÍCH

Không phải mọi điều đều có thể được định nghĩa trong ngôn ngữ mô hình hoá, dù là dùng hết mức của nó. Để có khả năng bổ sung thêm thông tin vào mô hình mà vốn không thể nào biểu diễn được, UML cung cấp khả năng chú thích. Một chú thích có thể được đặt vào bất kì đâu trong bất kì biểu đồ nào, và nó có thể chứa bất kì kiểu thông tin nào. Kiểu thông tin của nó là xấu không được UML diễn giải. Chú thích về cơ bản được gắn với một phần tử nào đó trong biểu đồ bằng đường chấm chấm vốn xác định phần tử nào được giải thích hay làm chi tiết, cùng với thông tin trong chú thích.

Chú thích thường chứa lời giải thích hay câu hỏi từ người làm mô hình xem như điều nhắc nhở để giải quyết một sự việc khó xử vào thời gian sau. Chú thích cũng có thể có khuôn mẫu vốn mô tả cho kiểu chú thích.

ĐẶC TẢ

Các phân tử mô hình có các tính chất vốn giữ các giá trị dữ liệu về phân tử đó. Tính chất được xác định qua tên và một giá trị được gọi là *giá trị có gắn thẻ*, mà thường mang kiểu đặc biệt như interger hay string. Có một số các tính chất định sẵn như Tài liệu, Trách nhiệm, Bền vững và Tương tranh.

Các tính chất được dùng để bổ sung thêm các đặc tả phụ về các thể nghiệm phân tử mà thường không được vẽ trong biểu đồ. Về cơ bản lớp được mô tả bằng một văn bản nào đó cụ thể hoá một cách không hình thức trách nhiệm và khả năng của lớp đó. Kiểu đặc tả này thường không được vẽ trong chính bản thân biểu đồ.

Mở rộng UML

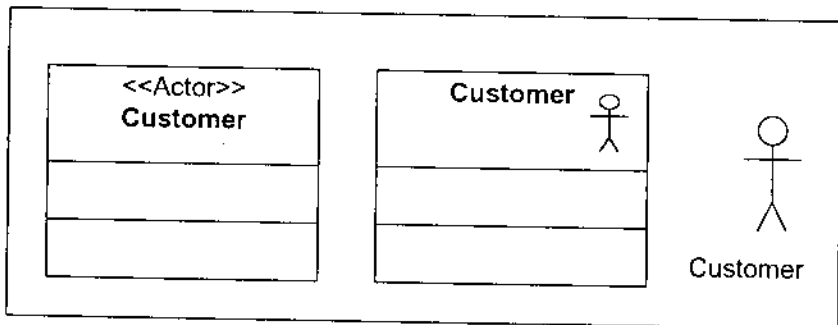
UML có thể được mở rộng hay thích ứng cho một phương pháp, tổ chức hay người dùng đặc biệt. Chúng ta động chạm tới ba cơ chế mở rộng ở đây: khuôn mẫu, giá trị có gắn thẻ, và ràng buộc.

KHUÔN MẪU

Cơ chế mở rộng khuôn mẫu xác định một loại phân tử mô hình mới dựa trên một phân tử mô hình đã có. Vậy, khuôn mẫu là "tựa như" một phân tử hiện tồn tại, cộng thêm một số ngữ nghĩa phụ vốn không có trong phân tử trước. Khuôn mẫu của một phân tử có thể được dùng trong cùng tình huống mà trong đó phân tử gốc được dùng. Các khuôn mẫu đều dựa trên tất cả các kiểu phân tử - lớp, nút, cấu phần và chú thích, cũng như các quan hệ như liên kết, tổng quát hoá và phụ thuộc. Một số khuôn mẫu được xác định trước trong UML, và chúng được dùng để điều chỉnh phân tử mô hình

hiện có thay vì định nghĩa phần tử mới. Điều này giữ cho ngôn ngữ UML đơn giản.

Một khuôn mẫu được mô tả bằng việc đặt tên của nó như một xâu - chẳng hạn, <<Tên khuôn mẫu>> - quanh tên của của phần tử như được vẽ trong Hình B.16. Khuôn mẫu cũng có thể có biểu diễn đồ họa riêng của nó, như một biểu tượng được ghép nối với nó. Một phần tử của một khuôn mẫu xác định có thể được vẽ trong dạng biểu diễn thông thường của nó với tên khuôn mẫu ở ngay trước tên này, như một biểu tượng đồ họa biểu diễn cho khuôn mẫu, hay như việc tổ hợp của cả hai. Bất kì khi nào một phần tử có tên khuôn mẫu hay biểu tượng được nối với nó, thì nó được đọc là một kiểu phần tử của khuôn mẫu đặc biệt.

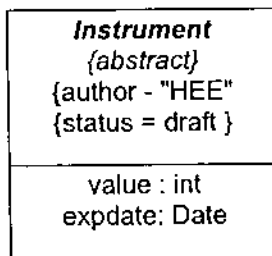


Hình B.16 Khách hàng là lớp với khuôn mẫu <<Actor>>.

GIÁ TRỊ CÓ GẮN THỂ

Như đã được mô tả trước đây, các phần tử có thể có các tính chất chứa cặp tên-giá trị về thông tin về chúng (Hình B.17). Những tính chất này cũng còn được gọi là *giá trị có gắn thể*. Một số các tính chất được xác định sẵn trong UML, nhưng các tính chất cũng có thể được người dùng xác định để giữ thông tin phụ về phần tử. Bất kì kiểu thông tin nào cũng có thể được gắn với các phần tử: thông tin xác định phương pháp, thông tin quản trị về tiến trình mô

hình hoá, thông tin được dùng bởi các dụng cụ khác, như dụng cụ sinh mã, hay bất kì loại thông tin nào mà người dùng muốn gắn vào phần tử đó.



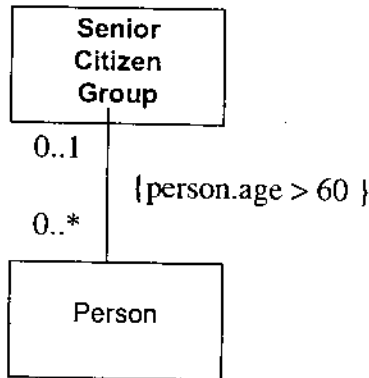
Hình B.17 Các tính chất trên lớp Instrument. Abstract là tính chất định sẵn; author và status là các giá trị có mang thể do người dùng xác định.

RÀNG BUỘC

Ràng buộc là một hạn chế lên một phần tử vốn giới hạn việc dùng phần tử đó hay ngữ nghĩa (nghĩa) của phần tử đó. Ràng buộc hoặc được khai báo trong dụng cụ và được dùng lặp lại trong nhiều biểu đồ, hoặc là được định nghĩa và áp dụng khi cần trong biểu đồ.

Hình B.18 chỉ ra liên kết giữa lớp Senior Citizen Group và lớp Person, chỉ ra rằng nhóm này có thể có người liên kết với nó. Tuy nhiên, để diễn tả rằng chỉ những người già hơn 60 tuổi mới có thể được gắn vào nó, một ràng buộc được định nghĩa để giới hạn việc tham dự chỉ cho những người có thuộc tính độ tuổi lớn hơn 60. Định nghĩa này ràng buộc người nào được dùng trong liên kết. Không có nó, có thể có hiểu lầm khi diễn giải biểu đồ này. Trong kịch bản tồi nhất, điều đó có thể dẫn tới việc thực hiện không đúng cho hệ thống.

Trong trường hợp này, ràng buộc đó được xác định và áp dụng một cách trực tiếp cho biểu đồ mà trong đó nó được dùng, nhưng nó cũng có thể được định nghĩa như một ràng buộc với tên và đặc tả như "Senior Citizen" và "Person.Age > 60" và được dùng trong nhiều biểu đồ.



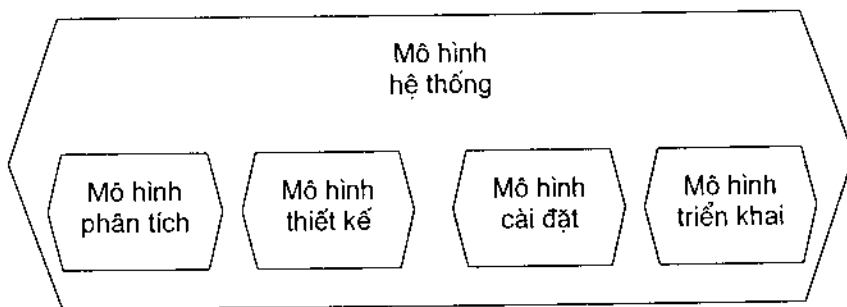
Hình B.18 Ràng buộc hạn chế các sự vật Person có thể tham dự trong liên kết.

Mô hình hoá bằng UML

Khi xây dựng hệ thống bằng UML, không chỉ một mô hình được xây dựng. Có các mô hình phân biệt trong các pha phát triển, và mục đích của các mô hình này là tách biệt. Trong pha phân tích, mục đích của mô hình là thu tóm các yêu cầu về hệ thống và mô hình các lớp "thế giới thực" cơ bản cùng sự hợp tác. Trong pha thiết kế, mục đích của mô hình là mở rộng mô hình phân tích thành giải pháp kĩ thuật làm việc với xem xét tới môi trường cài đặt. Trong pha cài đặt, mô hình thực tế là mã nguồn vốn được lập trình và dịch thành chương trình. Và cuối cùng, trong mô hình triển khai, một mô tả giải thích cách thức hệ thống được triển khai trong kiến trúc vật lí. Việc theo dõi giữa các pha và mô hình được duy trì qua các tính chất hay mối quan hệ làm mịn.

Mặc dầu các mô hình là khác nhau, chúng thông thường được xây dựng bằng việc mở rộng nội dung của các mô hình trước đó. Bởi vì điều này, tất cả các mô hình nên được cất giữ để cho dễ quay lại làm lại hay mở rộng mô hình phân tích ban đầu, và rồi dần dần

đưa thay đổi vào trong các mô hình thiết kế và cài đặt (xem Hình B.19)



Hình B.19 Hệ thống được mô tả trong nhiều mô hình.

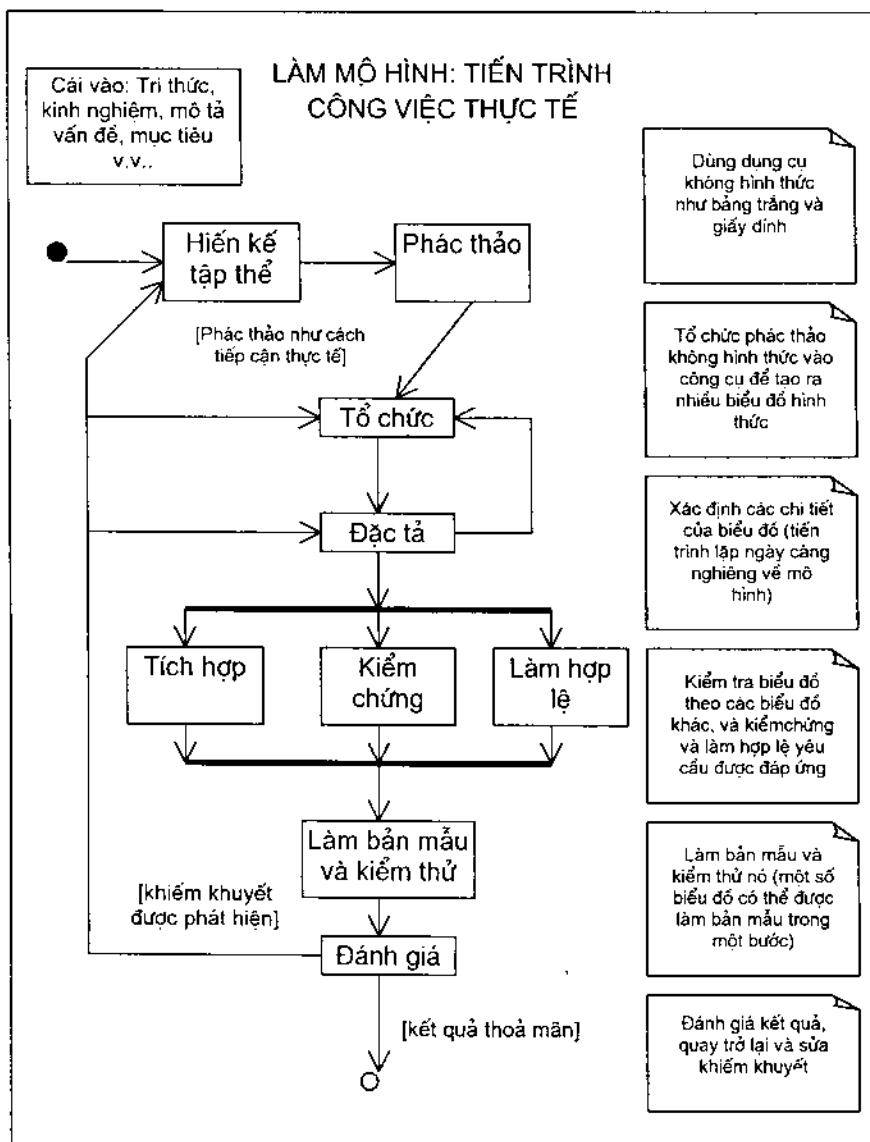
Khi làm mô hình hoá bằng UML, công việc nên được cai quản bằng một phương pháp hay tiến trình vốn nêu đại cương các bước khác nhau cần tiến hành và cách thức các bước này được thực hiện. Một tiến trình như thế về cơ bản phân chia công việc thành những việc lập liên tiếp của các pha phân tích yêu cầu / phân tích / thiết kế / cài đặt / triển khai. Tuy nhiên cũng có một tiến trình nhỏ vốn quan tâm tới công việc mô hình hoá thực tại. Thông thường, khi tạo ra một mô hình hay một biểu đồ, công việc được bắt đầu bằng tập hợp một nhóm người thích hợp, người trình bày vấn đề và mục tiêu; họ tham gia vào các phiên thảo luận hiến kế tập thể và phác hoạ mà trong đó ý tưởng về mô hình có thể được đưa ra trao đổi. Các dụng cụ được dùng là rất không hình thức - thông thường là các tờ giấy dính và bảng trắng. Phiên này tiếp tục cho tới khi những người tham dự cảm thấy họ có một gợi ý thực tế về cơ sở của mô hình (một giả thuyết sớm sủa). Kết quả này được đưa vào trong dụng cụ; mô hình giả thuyết được tổ chức, và một biểu đồ thực tế được xây dựng tương ứng với các qui tắc của ngôn ngữ mô hình. Tiếp đó, mô hình này được làm chi tiết thông qua công việc lặp lại, qua đó nhiều chi tiết về giải pháp được phát hiện ra và được làm tài liệu. Khi càng nhiều thông tin được thu thập về vấn đề và giải pháp

của nó, thì giả thuyết dần trở thành một chẩn đoán cho mô hình dùng được. Khi mô hình gần được hoàn thành, thì bước tích hợp và kiểm chứng được tiến hành, điều đó đưa tới mô hình hay biểu đồ này được tích hợp với các biểu đồ hay mô hình khác trong cùng dự án để đảm bảo rằng không có điều gì không nhất quán. Mô hình này cũng được làm hợp lệ để kiểm chứng rằng nó giải quyết đúng vấn đề (xem Hình B.20).

Cuối cùng, mô hình này được cài đặt vào trong một loại bản mẫu nào đó để được đánh giá về bất kì khiếm khuyết nào trong giải pháp thực tế. Các khiếm khuyết bao gồm những điều như thiếu chức năng, hiệu năng kém, hay chi phí phát triển cao. Những khiếm khuyết đó nên đưa người phát triển quay trở về các bước tương ứng để loại bỏ chúng. Nếu vấn đề là chủ yếu, thì người phát triển có thể phải đi theo mọi đường quay lại pha hiển kế tập thể phác hoạ. Nếu vấn đề là nhỏ, thì người phát triển có thể sẽ chỉ phải thay đổi các bộ phận của tổ chức hay đặc tả của mô hình. Lưu ý rằng bước làm bản mẫu không thể được thực hiện ngay lập tức sau khi biểu đồ kết thúc; nó nên được tiến hành khi một số biểu đồ có thể được làm bản mẫu cùng nhau. Bản mẫu có thể là bản để vứt đi, được xây dựng chỉ để đánh giá, hay nếu bước làm bản mẫu mà thành công thì nó trở thành việc lặp trong tiến trình phát triển thực.

Dụng cụ

Việc dùng một ngôn ngữ mô hình hoá phức tạp và bao quát như UML đòi hỏi sự hỗ trợ của các dụng cụ. Ngay cả nếu phác hoạ đầu tiên của mô hình có được làm bằng việc dùng bảng trắng (vẽ mô hình bằng tay), thì công việc duy trì, đồng bộ hoá và cung cấp sự nhất quán trong một số biểu đồ cũng gần như là không thể được nếu không có dụng cụ.



Hình B.20 Tiến trình làm việc mô hình hoá thực tế.

Thị trường dụng cụ mô hình hoá hay dụng cụ CASE vẫn còn chưa chín muồi đáng ngạc nhiên từ khi đưa ra tầm nhìn đầu tiên về chương trình được dùng để tạo ra chương trình. Nhiều dụng cụ cũng vẫn chưa hơn dụng cụ vẽ, với một vài kiểm tra nhất quán hay tri thức về phương pháp hay trình bày ngôn ngữ mô hình hoá. Dầu sao thì cũng có những cải tiến, và ngày nay dụng cụ đang gắn với tầm nhìn ban đầu. Nhưng nhiều dụng cụ vẫn còn chưa tinh tế, vẫn còn lỗi hay những điểm đặc thù mà ứng dụng thông thường không có, như vấn đề với cắt và dán. Những dụng cụ này cũng còn bị giới hạn bởi sự kiện là chúng tất cả đều có ngôn ngữ mô hình hoá riêng, hay ít nhất thì cũng là định nghĩa riêng của chúng về ngôn ngữ. Với việc đưa ra UML, bây giờ người bán dụng cụ dành nhiều thời gian hơn để cải tiến về dụng cụ và ít thời gian để định nghĩa phương pháp và ngôn ngữ mới.

Dụng cụ CASE hiện đại nên cung cấp các chức năng sau:

- *Vẽ biểu đồ*: Dụng cụ phải hỗ trợ cho việc dễ dàng tạo lại các biểu đồ trong ngôn ngữ mô hình. Dụng cụ nên đủ thông minh để hiểu mục đích của các biểu đồ và biết ngữ nghĩa và qui tắc đơn giản để cho nó có thể cảnh báo hay ngăn cấm việc dùng không thích hợp hay không đúng các phần tử mô hình.
- *Hành động như kho chứa*: Dụng cụ phải hỗ trợ cho một kho chung để cho thông tin được thu thập về mô hình được cất giữ ở một chỗ. Nếu tên của một lớp bị thay đổi trong một biểu đồ, thì thay đổi đó phải được phản ánh trong tất cả các biểu đồ có dùng lớp đó.
- *Hỗ trợ hoa tiêu*: Dụng cụ nên làm cho nó dễ dàng lèo lái qua mô hình, theo dấu một phần tử từ biểu đồ này sang biểu đồ khác, hay để mở rộng mô tả về một phần tử.
- *Cung cấp sự hỗ trợ đa người dùng*: Dụng cụ nên hỗ trợ cho nhiều người dùng, và có khả năng làm cho họ làm việc trên mô hình mà không can nhiễu hay quấy rối lẫn nhau.

- *Sinh mã*: Dụng cụ tiên tiến nên có khả năng sinh mã, tất cả thông tin trong mô hình được dịch thành bộ khung mã vốn để được dùng như cơ sở cho pha cài đặt.
- *Đảo ngược kỹ nghệ*: Dụng cụ tiên tiến nên có khả năng đọc mã hiện có và tạo ra các mô hình từ nó. Vậy, một mô hình có thể được tạo nên từ mã hiện có; hoặc người phát triển có thể lặp giữa việc làm việc trong dụng cụ mô hình và lập trình.
- *Tích hợp với dụng cụ khác*: Dụng cụ nên có khả năng tích hợp với các dụng cụ khác, cả hai đều cùng với môi trường phát triển như trình soạn thảo, trình biên dịch và gỡ lỗi, và với các dụng cụ khác của công ti như hệ thống quản lý cấu hình và kiểm soát phiên bản.
- *Bao trùm mô hình cho mọi mức trừu tượng*: Dụng cụ nên dễ lái từ mô tả mức đỉnh của hệ thống (như một số gói) xuống tới mức mã. Rồi để truy nhập vào mã cho một thao tác xác định trong một lớp, bạn chỉ cần bấm vào tên thao tác trong biểu đồ.
- *Trao đổi mô hình*: Mô hình hay biểu đồ riêng lẻ từ mô hình nên có khả năng xuất khẩu từ dụng cụ này rồi nhập khẩu từ dụng cụ khác, như mã Java được tạo ra trong dụng cụ này và rồi được dùng trong dụng cụ khác. Cùng việc trao đổi cũng nên áp dụng cho các mô hình trong ngôn ngữ xác định rõ.

B.2.2 TIẾN TRÌNH SỬ DỤNG UML

UML chứa các kí pháp và qui tắc làm cho nó có thể diễn tả cho các mô hình hướng sự vật. Tuy nhiên nó không mô tả trước cách công việc nên được tiến hành thế nào; tức là tiến trình hay phương pháp làm việc với ngôn ngữ này. Thay vì vậy, nó được thiết kế để dùng cho nhiều tiến trình, khác nhau cả về phạm vi và mục đích. Dẫu sao đi chăng nữa, để dùng UML thành công, một loại tiến trình nào đó là cần thiết, đặc biệt khi thiết kế các mô hình lớn đòi hỏi nỗ lực tổ. Công việc của mọi người phải được phối hợp và

mọi người phải nhằm tới cùng hướng. Việc dùng tiến trình cũng làm hiệu quả hơn cho việc đo tiến độ và kiểm soát và cải tiến công việc. Bên cạnh đó, tiến trình, đặc biệt trong lĩnh vực kỹ nghệ phần mềm, sẽ có khả năng dùng lại nhiều, cả dưới dạng bản thân tiến trình và các bộ phận (mô hình, cấu phần, kết luận, v.v..)

Một cách đơn giản, tiến trình mô tả cho cái phải làm, cách làm nó, khi nào làm nó và tại sao lại phải làm nó. Nó mô tả một số hoạt động cần được tiến hành theo trật tự nào đó. Xem như kết quả của tiến trình, tài liệu (mô hình và các mô tả khác) về hệ thống được xây dựng sẽ được tạo ra, và sản phẩm giải quyết cho vấn đề ban đầu được đưa vào và chuyển giao. Như ngôn ngữ mô hình cần dụng cụ hỗ trợ, tiến trình cần dụng cụ hỗ trợ cho các hoạt động có hữu của nó. Không may những dụng cụ như vậy còn chưa có sẵn rộng rãi như các dụng cụ cho việc mô hình hoá.

Khi các hoạt động có tên của một tiến trình được hoàn tất, thì mục tiêu tường minh được đạt tới. Tiến trình nên tạo ra kết quả có giá trị với khách hàng (người dùng) của nó. Để đạt được mục đích đó, tiến trình tiêu thụ tài nguyên dưới dạng con người, máy tính, dụng cụ, thông tin, v.v.. Có những qui tắc cho việc dùng tài nguyên này, và một phần quan trọng của việc mô tả tiến trình là định nghĩa các qui tắc này. Tiến trình thông thường được chia thành các tiến trình con lồng nhau. Tại mức đáy, tiến trình là nguyên tử, nghĩa là nó không thể nào bị phân chia hay bể nhỏ thêm nữa.

Tương phản lại, phương pháp thông thường cũng được coi như một tập các hoạt động có quan hệ với nhau, nhưng lại không có mục tiêu, tài nguyên và qui tắc tường minh. Trong kỹ nghệ phần mềm, cả phương pháp và tiến trình đều cần một ngôn ngữ mô hình hoá xác định rõ để diễn đạt và trao đổi kết quả của chúng. Một số nguồn tài liệu còn dự kiến định nghĩa ý nghĩa của "tiến trình" và "phương pháp", bao gồm những khác biệt giữa các thuật ngữ này. Tuy nhiên, với mục đích của chúng ta thì sẽ không có sự khác biệt có ý nghĩa nào giữa hai thuật ngữ này, mặc dầu tiến trình sẽ được

dùng chủ yếu. Một số tiến trình có mối quan tâm chung, như tiến trình quản lí, tiến trình chất lượng, tiến trình chế tạo và tiến trình phát triển. Chúng ta sẽ thảo luận về các tiến trình phát triển phần mềm có dùng UML.

Định nghĩa và hiểu về tiến trình kĩ nghệ phần mềm

Việc định nghĩa tiến trình về kĩ nghệ phần mềm không phải là dễ dàng. Nó đòi hỏi hiểu cơ chế đằng sau công việc phát triển phần mềm và tri thức về cách công việc này *nên* được tiến hành. Trong tình hình này, một số khía cạnh tổng quát của tiến trình kĩ nghệ phần mềm cần phải được giới thiệu. Trong mục này chúng ta thảo luận về kĩ nghệ phần mềm và sự hỗ trợ tiến trình/phương pháp. Bao giờ cũng có nhu cầu về cách thức có cấu trúc để làm việc và suy nghĩ, dù tưởng mình như được hướng dẫn bởi tiến trình hay không tưởng mình như được hướng dẫn bởi phương pháp. Với suy nghĩ đó, tiến trình có thể được xét theo các khía cạnh sau:

- *Hoàn cảnh tiến trình*, mô tả cho miền vấn đề mà trong đó tiến trình có thể được dùng
- *Người dùng tiến trình*, nêu hướng dẫn về cách tiến trình nên được chấp thuận và sử dụng
- *Các bước tiến trình*, mô tả cho các bước (hoạt động) cần được tiến hành trong tiến trình.
- *Đánh giá tiến trình*, mô tả cách đánh giá kết quả (tài liệu, sản phẩm, kinh nghiệm, v.v..)

HOÀN CẢNH TIẾN TRÌNH

Tiến trình phải mô tả cho hoàn cảnh của nó, miền vấn đề mà trong đó nó áp dụng được. Lưu ý rằng người ta có thể không muốn (hay thậm chí không thể) phát triển hay chọn lựa một tiến trình tổng quát vốn giải quyết mọi vấn đề tiềm năng. Điều quan trọng chỉ

là các vấn đề trong một miền vấn đề đặc biệt được giải quyết đúng bằng tiến trình này. Tuy nhiên điều này tạo ra điều khó xử, là ở chỗ thường miền vấn đề lại không hoàn toàn được biết tới chừng nào mà tiến trình còn chưa được hoàn tất.

Miền vấn đề, cũng còn được gọi là hoàn cảnh vấn đề, là một phần của một hoàn cảnh tổ chức hay nghiệp vụ bao gồm con người, quyền lực, chính trị, văn hoá và nhiều thứ. Nghiệp vụ có mục tiêu, và hệ thống tin cần phát triển nên hỗ trợ cho các mục tiêu đó. Có ít nhất bốn lí do để xem xét miền vấn đề (có liên quan tới hệ thống tin) xem như một phần của tổ chức hay nghiệp vụ:

- Tính hiệu quả trong hệ thống tin có thể được đo chỉ bởi đóng góp của nó cho nghiệp vụ.
- Để có khả năng phát triển hệ thống tin, người phát triển phải tương tác với những người bên trong nghiệp vụ đó.
- Để giải quyết một vấn đề, người giải quyết vấn đề (người dùng tiến trình) phải được giới thiệu với nghiệp vụ và hiểu cách nó vận hành.
- Khi người giải quyết vấn đề được giới thiệu với nghiệp vụ, thì mối quan hệ liên con người được hình thành giữa người trong nghiệp vụ và người giải quyết vấn đề

Khi xác định một nghiệp vụ, thuật ngữ này nên được xem xét theo nghĩa tổng quát, bởi vì không có mô hình hay tiến trình nào có thể thu tóm được tất cả mọi khía cạnh của nghiệp vụ và giới hạn này phải được làm tài liệu bên trong tiến trình kĩ nghệ phần mềm. Một kĩ thuật nổi tiếng cho việc mô tả nghiệp vụ là mô tả nó cùng với các biểu dụ như máy móc, cơ chế, cấu trúc kiểm soát hợp lí, văn hoá xã hội, hệ thống chính trị, nhà tù tâm lí, luồng và biến đổi, công cụ chi phối và nhiều thứ khác.

NGƯỜI DÙNG TIẾN TRÌNH

Tiến trình kĩ nghệ phần mềm phải bao hàm các hướng dẫn về việc dùng nó. Những hướng dẫn này không nên chỉ nói tới bản thân

tiến trình, mà còn cần nói tới con người dùng tiến trình đó, những người giải quyết vấn đề được dự định. Con người chúng ta có kết cấu tâm trí ảnh hưởng tới cách ta nghĩ, bao gồm cách ta nghĩ về việc dùng dụng cụ chẳng hạn như một tiến trình. Các kết cấu tâm trí này có thể được chia ra theo mức độ:

- Tiến trình cảm nhận (cách chúng ta lấy thông tin)
- Giá trị / đạo đức
- Động cơ và định kiến
- Khả năng lập luận
- Kinh nghiệm
- Kỹ năng và tập trí thức
- Khả năng cấu trúc hoá
- Vai trò (trong xã hội và nghiệp vụ)
- Hình mẫu, mô hình và khuôn khổ có "trong đầu" chúng ta

Tất cả những kết cấu này ảnh hưởng tới cách diễn giải của chúng ta về mô tả tiến trình. Chẳng hạn, kinh nghiệm của chúng ta về một tiến trình có thể ảnh hưởng tới cách chúng ta cấu trúc mọi sự cho dù tiến trình này cho chúng ta một hướng khác. Các giá trị và đạo đức của chúng ta có thể hạn chế chúng ta không lấy các bước tiến trình nào đó, như tổ chức lại hay thương lượng (đặc biệt trong các tiến trình phát triển và cải tiến nghiệp vụ). Một tiến trình được xác định tốt vậy phải hướng dẫn người dùng của nó, để tránh dùng lầm gây ra bởi các kết cấu tâm trí.

Các bước tiến trình

Phần lớn các tiến trình cho việc phát triển phần mềm đều bao gồm ít nhất ba pha cơ bản (một số tiến trình có thể có nhiều hơn). Chúng là *phát biểu vấn đề*, *thiết kế giải pháp*, và *thiết kế cài đặt*. Pha phát biểu vấn đề giúp phát hiện và phát biểu các vấn đề; pha thiết kế giải pháp phát biểu ra giải pháp cho vấn đề; còn pha cài đặt thì đưa vào và cài đặt cho giải pháp, do đó (một cách lí tưởng) khử

bỏ đi vấn đề. Trong tiến trình kĩ nghệ phần mềm cho các hệ thống hướng sự vật, các bước này có thể dễ dàng dịch thành *phân tích, thiết kế và cài đặt*.

PHÁT BIỂU VẤN ĐỀ

Phát biểu vấn đề bao gồm năm bước chung, vốn tất cả đều là các tiến trình và phương pháp đã được định nghĩa rõ cho kĩ nghệ hệ thống:

1. Hiểu miền các mối quan tâm.
2. Hoàn chỉnh chẩn đoán
3. Xác định đại cương dự đoán
4. Xác định vấn đề
5. Suy dẫn ra hệ thống ý niệm

Bước thứ nhất là để hiểu miền vấn đề; nếu không có thì sẽ khó mô tả cho vấn đề về sau. Điều quan trọng là phải nghiên cứu bản chất của vấn đề để tránh việc phát biểu vấn đề mà lại không bao hàm các vấn đề quan trọng nhất, điều có thể xảy ra. nếu vấn đề được mô tả chỉ như chúng xuất hiện. Lưu ý rằng các vấn đề có liên quan lẫn nhau, và kết quả là không phải bao giờ cũng dễ dàng thu tóm được vấn đề nền tảng vốn có thể là căn nguyên của những điều hiển nhiên. Các vấn đề có thể được mô tả bên trong hệ thống cấp bậc vấn đề. Chẳng hạn một vấn đề có thể là ở chỗ hệ thống hậu cần trong tổ chức không làm việc. Nhưng vấn đề *nền tảng* lại có thể là ở chỗ các tác nhân giải quyết hậu cần không có đủ tri thức; trong trường hợp đó thì chẳng hệ thống tin nào giúp được cho việc giải quyết vấn đề chừng nào mà tác nhân đó còn chưa được huấn luyện lại. Việc huấn luyện có thể được tiến hành bằng phương tiện hệ thông tin, nhưng phần mềm đó không thể được xây dựng chừng nào mà vấn đề thực còn chưa được phát hiện và mô tả.

Bước thứ hai là hoàn chỉnh chẩn đoán, nghĩa là miền hiện tại nên được mô tả dưới dạng điều đã được phát hiện trong bước thứ nhất. Lần nữa, bước trước phải biết và hiểu miền các mối quan tâm;

bước này là để tóm tắt và làm tư liệu cho nó. Tài liệu này được gọi là bản chẩn đoán. Bản chẩn đoán có thể là ở chỗ hệ thống của chúng ta ở những khía cạnh nào đó không được hiệu quả nếu so sánh với hệ thống của những đối thủ cạnh tranh.

Bước thứ ba là phát biểu đại cương dự đoán, trong đó miên tương lai mong muốn được mô tả. Việc mô tả này nên liệt kê ra nơi ta muốn tới, tại sao chúng ta muốn tới đó, và khi nào chúng ta muốn ở đó. Bản đại cương dự đoán có thể là ở chỗ chúng ta cần hệ thống hiệu quả hơn (theo các khía cạnh đã được chẩn đoán ở bước 2) các đối thủ cạnh tranh, vì chúng ta tin rằng tính hiệu quả như vậy là một sức mạnh cạnh tranh. Việc dự đoán nên liệt kê ra những khía cạnh nào trong hệ thống của chúng ta nên được cải tiến, điều này sẽ đem lại cho chúng ta những ưu thế nào, và khi nào chúng ta muốn điều này được đạt tới.

Bước thứ tư là mô tả cách đi từ miên hiện tại sang miên tương lai mong muốn. Bước này thông thường đòi hỏi cụ thể hoá các bước đi và mục tiêu con cần để đạt tới mục tiêu cuối cùng, miên tương lai mong muốn. Những bước và mục tiêu con này có thể, chẳng hạn, diễn đạt các chức năng cần thêm vào hay cải tiến trong hệ thống, và cách thức cài đặt những chức năng đó nhanh nhất có thể được.

Bước cuối cùng là suy dẫn ra hệ thống ý niệm. Hệ thống ý niệm là một hệ thống được phát biểu theo các kết cấu tâm trí của chúng ta, mà, nếu được thiết kế và cài đặt, thì chúng ta tin sẽ khử bỏ đi các vấn đề đã được xác định bên trong miên vấn đề.

THIẾT KẾ GIẢI PHÁP

Thiết kế giải pháp thông thường bao gồm hai bước chính: hoàn chỉnh thiết kế quan niệm/logic, và vẽ ra thiết kế vật lí. Các mô hình thiết kế quan niệm và logic (diễn đạt) hệ thống ý niệm bằng việc dùng ngôn ngữ mô hình hoá. Thiết kế vật lí được xây dựng từ các khía cạnh phi chức năng, như hiệu năng khả năng vận vận. Mục

mục đích của thiết kế quan niệm và logic là để tạo ra một mô hình vốn dễ dàng cho việc cài đặt; chẳng hạn, với ngôn ngữ lập trình hướng sự vật. Mục đích của thiết kế vật lý là để tạo ra một mô hình vốn thu tóm các khía cạnh phi chức năng và đưa tới việc dùng hiệu quả các tài nguyên của chúng ta; chẳng hạn, máy tính, máy in, mạng vân vân. UML làm việc hoàn hảo cho cả hai bước này; việc cài đặt do vậy là trực tiếp.

THIẾT KẾ CÀI ĐẶT

Bước cuối cùng là cài đặt các mô hình đã được tạo ra trong pha thiết kế giải pháp; theo thuật ngữ kỹ nghệ phần mềm, chúng phải được lập trình. Cài đặt cũng có nghĩa là hệ thống phải được giới thiệu cho khách hàng, cho nên bước này cũng bao quát việc làm tài liệu, trợ giúp trực tuyến, sách tra cứu, huấn luyện người dùng, chuyển đổi các bộ phận của hệ thống cũ và dữ liệu cũ và nhiều việc khác.

ĐÁNH GIÁ TIẾN TRÌNH

Điều quan trọng là phải đánh giá mọi thứ đã được hoàn thành và mọi thứ có thể ảnh hưởng tới công việc của chúng ta. Chỉ bằng việc đánh giá mà chúng ta mới có thể học được từ công việc của mình và tiếp tục cải tiến. Vậy người dùng tiến trình, bản thân tiến trình, và kết quả tiến trình nên liên tục được đánh giá, bởi vì chúng ta chưa bao giờ biết liệu vấn đề thực đã được thu tóm hay chưa (để thu tóm triệu chứng hơn nguyên nhân); hơn nữa, những vấn đề mới có thể tự lộ ra trong công việc. Thông thường, sau khi khách hàng hay người dùng mô tả miên vấn đề, thì người phát triển có kinh nghiệm sẽ nghiên cứu liệu mô tả của họ có thực sự thu tóm tất cả các vấn đề hay không (như câu của khách hàng).

Cả trong và sau dự án, các kết quả nên được đánh giá. Kết quả có thể được coi như sản phẩm mà dự án chuyển giao (mô hình, cấu phần phần mềm v.v..) Các sản phẩm được xác định và phát

triển để khử bỏ những vấn đề đã được phát hiện trong miền vấn đề, cho nên chúng phải được đánh giá dưới dạng chúng giải quyết vấn đề tốt đến mức nào. Nói tóm lại, nếu miền vấn đề không được đánh giá cẩn thận, thì sản phẩm không thể nào được đánh giá; còn nếu sản phẩm không thể nào được đánh giá, thế thì khách hàng/người dùng có thể không được thoả mãn với hệ thống cuối cùng.

Để cải tiến công việc của mình, tiến trình và người dùng tiến trình cũng cần được đánh giá, nói riêng, dưới hoàn cảnh hiện thời. Việc đánh giá kết quả (sản phẩm) cùng với phản ứng và kinh nghiệm của người dùng tiến trình (người phát triển) có thể xác định điều này. Điều quan trọng là đánh giá cách người dùng thực tế dùng tiến trình. Hãy hỏi các câu hỏi như: Người dùng có tuân theo tất cả các bước tiến trình không? Người dùng có tuân theo ý định tiến trình không? Kết quả có là điều trông đợi không? Vấn đề nào cá nhân người dùng kinh nghiệm thấy khi cài đặt tiến trình? Không may là việc đánh giá người dùng tiến trình không được thực hiện trên cơ sở đều đặn trong nhiều tiến trình kĩ nghệ phần mềm ngày nay.

Cơ sở cho tiến trình UML

Mặc dầu UML được coi là tổng quát cho việc mô hình hoá hướng sự vật, người thiết kế của nó phải có một loại tiến trình nào đó trong tâm trí khi họ thiết kế nó. Một cách tự nhiên các tiến trình họ có trong tâm trí đã được suy dẫn ra từ các ý tưởng trong phương pháp của Booch, OMT và OOSE/Objectory. Hiện thời họ đang làm việc để định nghĩa một tiến trình tích hợp mới (mà nghe nói sẽ mang tên Objectory) vốn sẽ dùng UML. Tuy nhiên, bất kì phương pháp hay tiến trình hướng sự vật nào cũng có thể dùng UML làm ngôn ngữ mô hình hoá. Các đặc trưng cơ sở mà những người thiết kế có trong tâm trí cho một tiến trình dùng UML là ở chỗ nó nên được điều khiển bởi trường hợp sử dụng, hướng kiến trúc, lặp lại và tăng dần. Các đặc trưng này bây giờ sẽ được khảo sát lần lượt.

HỆ THỐNG ĐƯỢC ĐIỀU KHIỂN THEO TRƯỜNG HỢP SỬ DỤNG

Trong UML trường hợp sử dụng thu tóm các yêu cầu chức năng của hệ thống; chúng "điều khiển" việc phát triển trong mọi giai đoạn tiếp theo sau phân tích yêu cầu. Vậy trường hợp sử dụng được cài đặt để đảm bảo rằng tất cả các chức năng đều được thực hiện trong hệ thống, và để kiểm chứng và kiểm thử hệ thống. Bởi vì trường hợp sử dụng chứa các mô tả về chức năng, nên chúng ảnh hưởng tới mọi pha và mọi cái nhìn, như được vẽ trong Hình B.21. Trong phân tích, chúng được dùng để thu tóm chức năng được yêu cầu và để làm hợp lệ chức năng này với khách hàng. Trong thiết kế và cài đặt, trường hợp sử dụng phải được hiểu trong việc xây dựng. Và cuối cùng, trong kiểm thử, trường hợp sử dụng kiểm chứng hệ thống; chúng trở thành cơ sở cho các trường hợp kiểm thử.

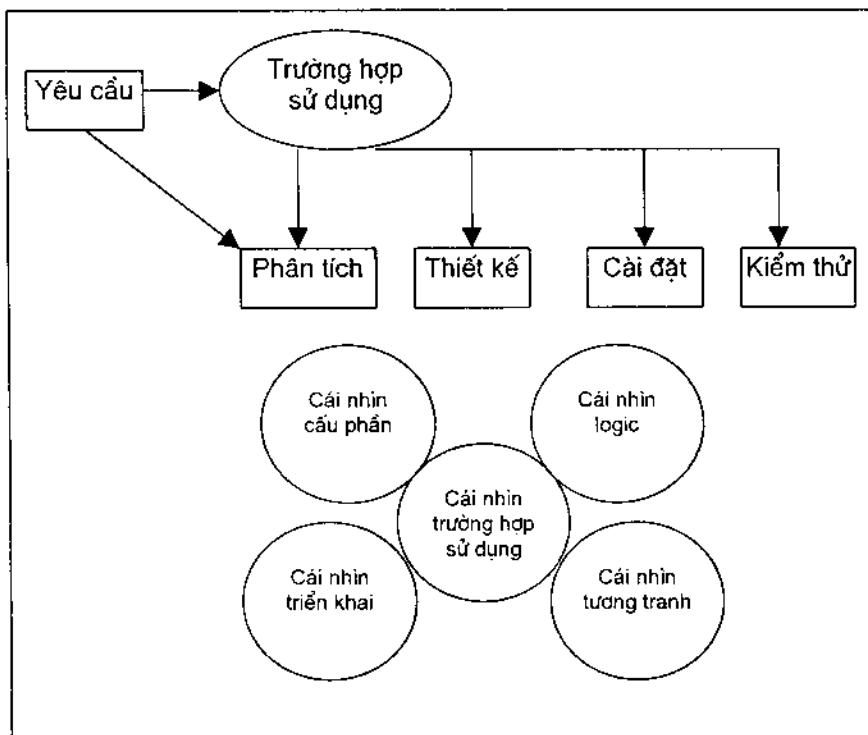
Các tiến trình trong đó trường hợp sử dụng là khái niệm thống trị, như Objectory, cũng mô tả sẵn rằng tổ chức làm việc phải dựa trên các trường hợp sử dụng. Một người phải chịu trách nhiệm cho từng trường hợp sử dụng, để theo sát nó từ đặc tả của nó cho tới kiểm thử cuối cùng.

HƯỚNG KIẾN TRÚC

Tiến trình dùng UML là hướng kiến trúc. Điều này có nghĩa là bạn coi một kiến trúc hệ thống cơ sở, xác định rõ là quan trọng, và có nghĩa là bạn cố gắng thiết lập một kiến trúc như vậy sớm sủa trong tiến trình. Kiến trúc hệ thống được phản ánh bởi các cái nhìn khác nhau về ngôn ngữ mô hình, và nó thực tế được phát triển qua nhiều việc lặp. Tuy nhiên điều quan trọng là định nghĩa kiến trúc cơ sở sớm sủa trong dự án, rồi làm bản mẫu và đánh giá nó, và cuối cùng, làm mịn nó trong tiến trình dự án.

Kiến trúc làm việc như bản đồ của hệ thống vốn định nghĩa ra các phần khác nhau của hệ thống, mối quan hệ và tương tác của chúng, cơ chế truyền thông của chúng, và các qui tắc toàn thể cho các bộ phận có thể được bổ sung hay thay đổi. Kiến trúc tốt dễ cập

tới cả các khía cạnh chức năng và phi chức năng. Nó đặc biệt quan trọng cho việc định nghĩa hệ thống vốn có thể được sửa đổi, vốn có thể được hiểu một cách trực giác, vốn có thể dùng lại được (hoặc trong hay ngoài hệ thống).



Hình B.21 Trường hợp sử dụng ảnh hưởng tới mọi pha và cái nhìn.

Khía cạnh quan trọng nhất của việc tạo ra một kiến trúc tốt là phân chia hệ thống một cách logic thành các hệ con (các gói UML), trong đó sự phụ thuộc giữa các gói khác nhau là đơn giản và nhận biết được. Sự phụ thuộc thường nên mang kiểu khách/nguồn phục vụ, với gói này biết về gói kia, nhưng không có việc ngược lại (nếu chúng cả hai đều phụ thuộc lẫn nhau, thì chúng rất khó tách

biệt). Các gói cũng thường được tổ chức theo tầng, với các tầng khác nhau giải quyết với cùng một thứ nhưng theo các mức độ trừu tượng khác nhau (tầng sự vật nghiệp vụ có thể ở trên đỉnh của tầng giải quyết bền vững, mà tầng này lại nằm trên tầng sinh SQL,v.v..)

LẬP

Xây dựng các mô hình bằng UML tốt nhất được tiến hành theo một số lần lặp nhỏ; tức là thay vì cố gắng định nghĩa tất cả các chi tiết của mô hình hay biểu đồ ngay một lúc, thì việc phát triển là một trình tự các bước, trong đó mỗi lần lặp lại thêm một số thông tin hay chi tiết mới. Thế rồi mỗi lần lặp lại được đánh giá, hoặc trên giấy hoặc trong bản mẫu làm việc, và được dùng để tạo ra cái vào cho lần lặp tiếp. Vậy một tiến trình dùng việc lặp thì sẽ cung cấp phản hồi liên tục vốn làm cải tiến chính bản thân tiến trình cũng như sản phẩm cuối.

Khi quyết định cái gì cần đưa vào trong việc lập, sự tập trung là vào việc lập vốn gây ra tác động lớn nhất lên hệ thống hay có mức độ rủi ro cao nhất. Việc lập với tác động lớn nhất làm cho hệ thống dùng được có thể được chuyển giao cho người dùng sớm hơn; việc lập với mức độ rủi ro cao nhất sẽ giải quyết các vấn đề thực sự động chạm một cách dễ dàng trong dự án, do đó làm giảm rủi ro toàn bộ của dự án. Mỗi việc lập nên xử trí các vấn đề lớn, không trì hoãn chúng. Những việc lập sớm cũng có thể được coi như bản mẫu, có khả năng bị vứt đi nếu chúng không tạo ra kết quả mong muốn.

TĂNG DẦN

Một hay một tập hợp các lần lặp có thể được xem như việc làm tăng dần hệ thống. Việc tăng là một bước trong tiến hoá của hệ thống, đôi khi được gọi là một phiên bản của hệ thống. Khi phát triển lặp lại và tăng lên, thì việc phát triển dựa trên định nghĩa về một số bước. Một bước được "chuyển giao" và được đánh giá bởi

ích lợi kĩ thuật, kinh tế và tiến trình. Khi hệ thống được chuyển giao, điều đó không nhất thiết có nghĩa là phần mềm được gửi tới khách hàng; trong nhiều trường hợp, nó là bản đưa ra nội bộ cho bộ phận phần mềm. Khi chuyển giao, tuy nhiên, tổ phát triển nên trao hệ thống trong trạng thái hiện tại của nó cho ai đó vốn là người nhận nó.

Bản tăng dần được chuyển giao vậy được kiểm thử và đánh giá, về cơ bản bởi tổ phần mềm có tham gia vào tiến trình đánh giá. Những câu hỏi này được đặt ra:

- Hệ thống có các chức năng như được giả thiết phải có trong bước này không; và chúng có làm việc như trông đợi không?
- Hệ thống có thoả mãn các thuộc tính phi chức năng như hiệu năng thích hợp, độ tin cậy và thân thiện người dùng không?
- Bước này đã được phát triển trong thời gian ước lượng không; và trong ngân sách tài nguyên dành cho nó không - bước này có được phát triển bên trong các ràng buộc kinh tế không?
- Liệu có vấn đề trong tiến trình / phương pháp trong bước này không? Có phải làm rõ ràng tiến trình/phương pháp không, hay các hành động có phải được thực hiện không, chẳng hạn thay thế hay bổ sung tài nguyên mới

Các câu hỏi khác có thể được thêm vào. Điều quan trọng là các bước này được dùng để đánh giá toàn bộ tiến trình phát triển - các khía cạnh chức năng và phi chức năng của sản phẩm, cũng như tiến trình và công việc của tổ phần mềm. Sau mỗi bước, các hành động có thể được tiến hành để sửa chữa bất kì vấn đề gì; và kinh nghiệm từ mỗi bước có thể được đưa vào cho bước tiếp. Một khi một vấn đề được xác định, thì không nhất thiết dẫn tới hành động đột ngột. Nó đơn giản có thể chỉ ra rằng các ước lượng quá lạc quan và không đúng đã được đưa ra, và chúng là lí do cho thất bại của

việc tăng thêm. Trong tiến trình dự án, các bước có nghĩa là điều chỉnh sản phẩm và tiến trình sao cho kết quả là thành công.

Mọi bước đều nên thêm chức năng hay thuộc tính mới vào cho sản phẩm. Nội dung của chúng nên được lập kế hoạch; chúng không nên chỉ là cái mà người phát triển cảm thấy thích. Như đã thảo luận trước đây, các chức năng vốn có tác động lớn nhất lên người dùng cuối hay đặt ra rủi ro lớn nhất thì nên được đặt ra trước. Ưu thế của hành động này là ở chỗ ý kiến người dùng về sản phẩm và giao diện là được đón nhận đủ sớm để được tính tới.

Những người chống đối việc phát triển lặp và tăng dần thường tuyên bố rằng phương pháp này làm cho khó lập kế hoạch và loại trừ việc lập kế hoạch dài hạn. Nhưng phát triển lặp và tăng dần lại kinh tế và dễ dự đoán hơn là mô hình thác nước, và khi được thực hiện đúng đắn, thì tạo ra tiến trình tự cải tiến của toàn bộ việc phát triển phần mềm.

Đối lập với phát triển tăng dần là mô hình thác nước đã nói trước đây, trong đó mỗi pha lại được thực hiện chỉ một lần và hệ thống được đưa ra trong cái gọi là "vụ nổ lớn". Toàn bộ việc sinh ra hệ thống đều được phân tích trong pha phân tích, được thiết kế trong chỉ mỗi pha thiết kế, và được cài đặt và kiểm thử trong pha xây dựng. Rủi ro gây ra từ mô hình này là rõ ràng:

- Phản hồi và kinh nghiệm trong tiến trình thiết kế không thể được tính tới (ít nhất cũng không theo cách thức có tổ chức). Các kinh nghiệm như vậy có thể liên quan tới cả sản phẩm và cách thức nó được tạo ra (tiến trình phát triển).
- Việc kiểm thử để xác định liệu các giả thiết phân tích và thiết kế có đúng không thường chỉ được tiến hành rất muộn trong dự án, khi có thể là quá trễ không sửa được chúng.
- Việc bảo trì lịch biểu có thể rất khó khăn bởi vì, trong pha xây dựng, ai đó có thể hiểu rằng bản kế hoạch là quá lạc

quan, mà vào lúc đó thì lại quá trễ không thể sửa chữa lại bản kế hoạch được.

- Rủi ro của dự án không được đẩy tới trước, và do vậy chúng có thể được sửa chữa rất muộn mằn trong dự án

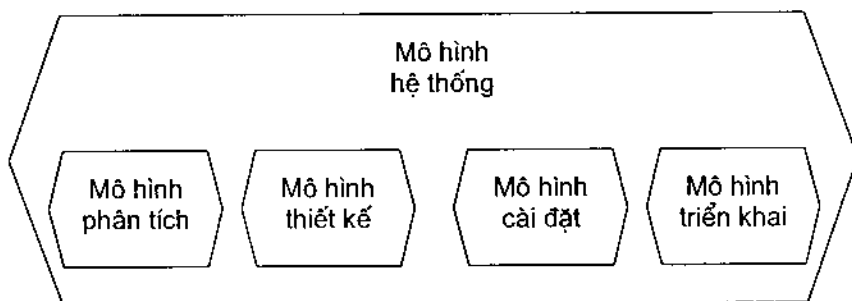
Một số quan sát thực tế: Cho dù những người ủng hộ phương pháp hướng sự vật vẫn nói và biện hộ cho phát triển tăng dần và lặp, trong thực tế, nhiều người vẫn dùng phương pháp thác nước. Chỉ bởi vì hệ thống đang được lập trình trong một số bản đưa ra (như kiểm thử alpha, beta, thử nghiệm v.v..) không có nghĩa là việc phát triển là tăng dần. Bạn hãy nhớ, mỗi lần tăng lên lại phải bổ sung thêm các khía cạnh chức năng hay phi chức năng mới cho hệ thống, và không phải chỉ là sửa một tập lỗi. Mỗi bước phải được đánh giá đúng và được dùng làm cái vào cho bước tiếp.

Phương pháp hướng sự vật truyền thống

Mục này nhìn vào các hoạt động khác nhau về cơ bản được thực hiện khi dùng phương pháp hướng sự vật, như được vẽ trong hình B.22. Đây không phải là nỗ lực để tạo ra phương pháp mới, mà cũng không thu tóm một phương pháp đặc biệt về chi tiết. Nó là mô tả tổng quát về các hoạt động thường được thực hiện trong một phương pháp. Phương pháp được nói tới ở đây là Booch, OMT, Fusion và vân vân. Các bước chính trong những phương pháp này là phân tích, thiết kế, xây dựng và kiểm thử, thông thường được gọi là các pha, mặc dầu chúng ta vẫn nói chúng là các hoạt động. Mục tiếp giới thiệu một tiến trình vĩ mô có chứa các pha này; để tránh lẫn lộn, phân tích, thiết kế, xây dựng và kiểm thử được gọi là các *hoạt động chính*, từng hoạt động trong chúng có một số *hoạt động con*.

Mỗi một trong các hoạt động chính lại tạo ra mô hình riêng của nó vốn chứa một số biểu đồ. Có thể đưa vào các chức năng mới bằng việc bổ sung chúng vào mô hình phân tích và rồi vào từng mô hình khác nữa. Bên cạnh các hoạt động chính, các hoạt động khác

vẫn hiện diện trong toàn bộ dự án, bao gồm hợp xét duyệt và xem xét, xác định vai trò cho tất cả các thành viên tổ phần mềm, biện pháp chất lượng vãn vãn. Tiến trình cũng có thể chứa các mô tả về cách các thủ tục chiều đứng này được thực hiện.



Hình B.22 Các pha và mô hình trong phương pháp/tiến trình truyền thống.

PHÂN TÍCH YÊU CẦU

Phân tích yêu cầu làm phát sinh một thoả thuận giữa khách hàng và người cung cấp hệ thống. Khách hàng có thể là người dùng bên trong cùng tổ chức với người cung cấp hay là tổ chức tách biệt nơi mà các hợp đồng nghiệp vụ chi tiết được viết ra dựa trên tài liệu yêu cầu. Các yêu cầu nên thật chi tiết, mặc dầu thường thì không thể định nghĩa mọi thứ trong một tài liệu như vậy. Khi không thể đi vào chi tiết trong phân tích yêu cầu, thì tài liệu nên diễn tả một ý định hay một ý tưởng cần được gọi ra khi phát sinh bất đồng giữa khách hàng và người cung cấp. Việc phân tích yêu cầu thường được tích hợp với mô hình hoá nghiệp vụ, trong đó tài nguyên nghiệp vụ, qui tắc, mục tiêu và hành động được mô hình hoá.

Phân tích yêu cầu dùng các trường hợp sử dụng, tiến trình nghiệp vụ, hay văn bản để mô tả cho các chức năng cần cho hệ thống. Hệ thống được xét từ bên ngoài; nó không nghiên cứu sâu

vào cách mọi thứ được thực hiện về mặt kĩ thuật. Công việc thực tế bao gồm thảo luận và thương lượng giữa khách hàng và người cung cấp.

Cùng với các khía cạnh chức năng, điều quan trọng phải nhớ tới các yêu cầu phi chức năng nữa. Các vấn đề như hiệu năng và độ tin cậy nên được thảo luận, cho dù là khó tại giai đoạn này để viết ra hợp đồng nghiệp vụ dựa trên một số điều ở đây. Những ràng buộc khác về hệ thống cũng nên được tiến hành: kích cỡ, môi trường kĩ thuật, việc tích hợp cần thiết với hệ thống thừa tự, sản phẩm và ngôn ngữ được dùng, và những điều khác. Các ước lượng chi phí toàn bộ cho phát triển hệ thống cần phải được tiến hành.

Phân tích yêu cầu cho kết quả trong bản đặc tả mà theo đó khách hàng và người cung cấp đều đồng ý. Để đạt được điều đó, một danh mục thuật ngữ đơn giản và /hoặc mô hình quan niệm về các thực thể cơ sở được dùng trong hệ thống là có ích; và một cách tự nhiên, nếu trường hợp sử dụng đã được dùng để thu tóm các yêu cầu chức năng thì chúng được bao gồm vào trong tài liệu nữa.

Các biểu đồ UML được tạo ra trong phân tích yêu cầu bao gồm biểu đồ trường hợp sử dụng, một số biểu đồ lớp đơn giản, và có thể một số biểu đồ hoạt động.

PHÂN TÍCH

Phân tích sinh ra các mô hình của miền vấn đề: lớp, sự vật, và tương tác vốn mô hình hoá cho các thực thể "thế giới thực". Việc phân tích nên thoát khỏi mọi chi tiết kĩ thuật hay cài đặt, và chứa một mô hình lí tưởng, vì nó là việc phát biểu cho vấn đề cần được giải quyết, và tiến hành việc thu nhận tri thức cần thiết về miền này.

Một số hoạt động tiêu biểu trong phân tích là:

- Tri thức miền được thu nhận từ các đặc tả yêu cầu, các mô hình về các tiến trình nghiệp vụ, danh mục thuật ngữ, mô tả hệ thống hiện tại, và phỏng vấn người dùng và bất

kì bên nào khác đáng quan tâm về hệ thống. Đây là hoạt động nghiên cứu.

- Nếu phân tích yêu cầu chính thức đã được tiến hành - các trường hợp sử dụng đã được định nghĩa cho hệ thống - thì nó trở thành một phần của cái vào cho phân tích. Nếu không thì việc định nghĩa chức năng hệ thống nên được tiến hành bằng các trường hợp sử dụng hay kỹ thuật hình thức tương tự.
- Tìm các ứng cử viên cho các lớp thích hợp về cơ bản được tiến hành trong phiên hiến kế tập thể, trong đó các lớp có thể có được liệt kê ra. Khi phiên kết thúc, một cuộc họp xem xét lại tất cả các ứng cử viên được tiến hành và một số lớp bị loại đi khỏi danh sách bởi một số lí do (như chúng là chức năng; chúng là các lớp trùng lặp với tên khác nhau; chúng không có các đặc trưng của lớp; chúng không bên trong miền; chúng không thể được định nghĩa theo cách đúng đắn; v.v..) Danh sách các lớp trong hệ thống về cơ bản thay đổi qua toàn bộ sự phát triển, khi kinh nghiệm mới dẫn tới việc thêm vào hay xoá đi các lớp.
- Các mối quan hệ tĩnh giữa các lớp được mô hình hoá dưới dạng các liên kết, tụ tập, tổng quát hoá và phụ thuộc. Các biểu đồ lớp được dùng để làm tư liệu cho các lớp, đặc tả của chúng, và mối quan hệ của chúng.
- Hành vi và sự hợp tác giữa các sự vật của lớp được mô tả bằng việc dùng các biểu đồ trạng thái, hợp tác, trình tự, hoạt động. Về cơ bản trường hợp sử dụng (hay kịch bản của trường hợp sử dụng) được mô hình hoá trong các biểu đồ trình tự hay hợp tác. Chú ý rằng không giải pháp kĩ thuật nào được mô hình hoá cả; tức là logic kĩ thuật hay nhân tố kĩ thuật như truy nhập cơ sở dữ liệu không được mô tả.

- Khi tất cả các biểu đồ đã được phát triển (thông thường công việc rất lặp lại ở nơi mọi sự thường thay đổi), thì mô hình toàn thể được kiểm chứng bằng việc cho chạy hệ thống "trên giấy." Toàn bộ mô hình được trình bày cho các chuyên gia miền và được thảo luận. Kịch bản được "diễn" và các chuyên gia được hỏi liệu đây có phải là mô hình tự nhiên cho việc giải quyết vấn đề hay không.
- Giao diện người dùng cơ sở có thể được làm bản mẫu, mặc dầu không nhất thiết đi vào chi tiết (như bố trí của sổ cuối cùng). Cấu trúc toàn thể - cách đi qua các cửa sổ, các biểu dụ được dùng và các nội dung cơ sở của cửa sổ chính - đều được làm bản mẫu, được kiểm thử và được thảo luận với đại diện của người dùng.

Tài liệu phân tích bao gồm một mô hình vốn mô tả cho miền vấn đề cần được xử trí trong hệ thống, cùng với hành vi cần thiết của các lớp miền để cung cấp chức năng cần thiết. Lần nữa, tài liệu nên mô tả cho hệ thống "lí tưởng", không đưa môi trường kĩ thuật và các chi tiết của nó vào xem xét.

Các biểu đồ UML được tạo ra trong phân tích là biểu đồ lớp, trình tự, hợp tác, trạng thái và hoạt động, và sự tập trung của chúng là vào miền vấn đề, không vào giải pháp kĩ thuật cụ thể.

THIẾT KẾ

Thiết kế là việc mở rộng và thích nghi kĩ thuật của kết quả phân tích. Các lớp, quan hệ và sự hợp tác từ pha phân tích được bổ sung thêm các phần tử mới, bây giờ tập trung vào cách cài đặt hệ thống trong máy tính. Tất cả các chi tiết về cách thức mọi thứ nên làm việc với nhau về mặt kĩ thuật và những ràng buộc của môi trường cài đặt đều được tính tới. Kết quả của phân tích được mang sang thiết kế và được bảo trì tại trung tâm của hệ thống. Các lớp trong phân tích (các sự vật nghiệp vụ) không nên bị làm xáo trộn trừ phi tuyệt đối cần thiết, để duy trì các tính chất cơ sở và hành vi

của chúng. Thay vì thế, các lớp của phân tích nên được cấy vào kết cấu nền kỹ thuật, nơi các lớp kỹ thuật giúp cho chúng trở thành bền vững, để trao đổi, để trình bày chúng trong giao diện người dùng, vãn vãn. Bằng việc tách bạch các lớp của phân tích ra khỏi kết cấu nền kỹ thuật, sẽ dễ dàng hơn nhiều cho việc thay đổi hay cập nhật chúng. Trong thiết kế, cùng các kiểu biểu đồ lại được dùng như trong phân tích, mặc dầu các biểu đồ mới phải được tạo ra và mô hình hoá để chỉ ra giải pháp kỹ thuật.

Các hoạt động điển hình trong thiết kế là:

- Pha kiến trúc, trong đó các lớp của phân tích được chia thành các gói chức năng (nếu chưa được làm trong phân tích). Các gói mới cho các vùng kỹ thuật như giao diện người dùng, giải quyết cơ sở dữ liệu, và truyền thông được bổ sung vào. Gói phân tích có thể dùng các dịch vụ kỹ thuật từ những gói này, nhưng điều ngược lại nên vẫn còn được giữ không bị ảnh hưởng nhiều nhất có thể được. Cơ chế truyền thông giữa các gói khác nhau được thiết lập (cố gắng cho các quan hệ khách / nguồn phục vụ mà trong đó các gói phân tích chức năng là nguồn phục vụ).
- Nhu cầu tương tranh được xác định và mô hình hoá thông qua các lớp tích cực, các thông báo không đồng bộ, và các kỹ thuật đồng bộ hoá để giải quyết tài nguyên dùng chung.
- Định dạng chi tiết của cái ra từ hệ thống được xác định: giao diện người dùng, báo cáo, và các giao tác được gửi sang hệ thống khác. Thiết kế giao diện người dùng có thể được xem như hoạt động thiết kế tách biệt, khi tính tới tầm quan trọng của nó.
- Các thư viện lớp và cấu phần cần thiết vốn nâng cao kiến trúc và làm tối thiểu công việc cài đặt cần tiến hành.
- Nếu dùng cơ sở dữ liệu quan hệ thì các lớp trong hệ thống được ánh xạ vào các bảng trong mô hình quan hệ truyền

thống. Cơ chế cho việc đọc từ cơ sở dữ liệu cũng được thiết lập trong thiết kế kiến trúc.

- Xem xét đặc biệt được tính tới để giải quyết các ngoại lệ và lỗi trong hệ thống. Điều này thông thường bao gồm cả việc giải quyết lỗi "bình thường" (lỗi có thể đoán được trong tiến trình thực hiện chức năng hệ thống) và lỗi bất thường (lỗi không thể đoán được và phải được giải quyết bằng cơ chế ngoại lệ tổng quát nào đó).
- Các lớp được cấp phát cho các cấu phần mã nguồn, và cấu phần thực hiện được cấp phát cho các nút, bằng việc dùng biểu đồ cấu phần và biểu đồ triển khai.

Hoạt động thiết kế chi tiết nên bao gồm đặc tả cho tất cả các lớp, kể cả các thuộc tính cài đặt cần thiết, giao diện chi tiết của chúng, và mô tả về các phép toán (trong giả mã hay văn bản rõ). Đặc tả nên được làm đủ chi tiết để cho, cùng với các biểu đồ trong mô hình, chúng cung cấp mọi thông tin cần thiết cho việc mã hoá.

Trong pha thiết kế, bạn cần nhớ:

- *Tính theo dõi được*: Phải làm tài liệu cho mọi quyết định để thấy rõ chúng được đưa ra trên cơ sở nào và từ đâu mà yêu cầu gốc được sinh ra. Phải tách biệt các mô hình phân tích, thiết kế và xây dựng lẫn nhau.
- *Giao diện*: Tạo ra giao diện đơn giản, đầy đủ và nhất quán sao cho tất cả các dịch vụ cấu phần có thể dễ dàng được hiểu và dùng.
- *Hiệu năng*: Không quá nhấn mạnh vào hiệu năng ở giai đoạn sớm (để tăng hiệu năng trong hệ thống làm việc hơn là cải tiến hệ thống nhanh mà không chạy).
- *Đơn giản*: Cố gắng tạo ra các cấu trúc đơn giản vốn chắc là được hiểu và dùng bởi mọi người phát triển, thay vì tạo ra các giải pháp "tài tình" mà chỉ vài người mới có thể hiểu nổi.

- *Tài liệu:* Giữ các chú thích về mọi thứ xảy ra trong tiến trình phát triển để cho tất cả các biến cố đều được làm tài liệu và tất cả các vấn đề đều có thể được theo dõi dấu vết.

Các biểu đồ UML được tạo ra trong pha thiết kế là biểu đồ lớp, trình tự, hợp tác, trạng thái, hoạt động, cấu phần và triển khai. Sự hội tụ của chúng là giải pháp kỹ thuật chi tiết, để cung cấp cơ sở cho pha cài đặt.

CÀI ĐẶT

Hoạt động cài đặt thực tế là việc viết mã. Nếu thiết kế đã được thực hiện đúng và với đủ chi tiết, thì việc mã hoá sẽ là nhiệm vụ đơn giản. Bước này bao gồm việc ra quyết định cuối cùng và dịch các biểu đồ thiết kế cùng các đặc tả vào cú pháp của ngôn ngữ lập trình được chọn (mà hi vọng là ngôn ngữ hướng sự vật). Nó cũng bao gồm tiến trình phát triển thực tế, dịch, móc nối và gỡ lỗi các cấu phần lập đi lập lại.

Công việc này được hỗ trợ bởi các qui tắc lập trình vốn dự định chuẩn hoá mã đã được người lập trình khác phát triển, và để ngăn cản các kết cấu nguy hiểm hay không thích hợp trong ngôn ngữ. Các cuộc giám định hay xét duyệt mã, chính thức hay không chính thức, tạo thuận lợi cho các chuẩn được tuân thủ và cải tiến chất lượng toàn diện của mã.

Hoạt động cài đặt thường rất phổ biến trong những người lập trình và quản lí. Người lập trình cảm thấy rằng đây là lĩnh vực mà mình quen thuộc (làm mô hình thường bị coi là trừu tượng và không cần thiết), còn người quản lí lại cảm thấy rằng chừng nào việc mã hoá còn chưa bắt đầu thì chưa công việc thực nào được làm cả. Khi phân tích và thiết kế hướng sự vật được đưa vào, điều khá thông thường đối với nhà quản lí là cứ bước quanh và nói, "Các anh đã bắt đầu mã hoá chưa?" hay "Bắt đầu mã hoá đi chứ!"

Trong phần lớn các trường hợp, việc chạy xô vào bắt đầu mã hoá là sai lầm. Hiển nhiên, các quyết định phân tích và thiết kế phải được tiến hành tại điểm nào đó; sự chọn lựa là ra quyết định thông qua tổ có thành phần đủ trong tiến trình mô hình hoá có cấu trúc hay phải để chúng được đưa ra một cách không tường minh bởi người lập trình trong khi mã hoá. Những quyết định khôn ngoan hơn thường được đưa ra theo cách có cấu trúc. Điều đó nói hãy nhớ tiến hành tiến trình phát triển toàn diện theo cách lặp lại: phân tích một chút, thiết kế một chút, mã hoá một chút và rồi quay lại phân tích nữa. Nếu bất kì một trong những hoạt động này kéo dài quá lâu (như phân tích đến hàng tháng), thì điều đó có thể dẫn tới quá nhiều việc quản lí và tài liệu, tiếp theo là giảm tính hiệu quả của phân tích.

Rất ít biểu đồ mới được tạo ra trong pha cài đặt; thay vì vậy, các biểu đồ được tạo ra trong pha thiết kế được làm chi tiết hay sửa chữa khi cần thiết.

KIỂM THỬ

Mục tiêu của kiểm thử là xác định lỗi trong mã. Việc tìm ra lỗi do vậy được coi là thành công, không phải thất bại. Kiểm thử bao gồm một số các trường hợp kiểm thử, chỗ mà các khía cạnh khác nhau của một phần dưới kiểm thử được kiểm tra. Mỗi trường hợp kiểm thử cho biết phải làm gì, dùng dữ liệu nào, kết quả nào được trông đợi. Khi tiến hành kiểm thử, kết quả - kể cả bất kì sai lệch nào so với trường hợp kiểm thử đã lên kế hoạch - đều được ghi lại trong biên bản kiểm thử. Thông thường sự sai lệch chỉ ra một lỗi trong hệ thống (mặc dầu đôi khi trường hợp kiểm thử cũng có thể sai và hệ thống đúng). Lỗi được ghi lại và được mô tả trong báo cáo kiểm thử, theo đó người lập trình có trách nhiệm được phân bổ để sửa lại lỗi cho phiên bản tiếp của hệ thống. Lỗi có thể là lỗi chức năng (như thiếu chức năng hay chức năng không đúng), phi chức năng (như hiệu năng quá thấp), hay lỗi logic (như chi tiết giao diện người dùng không được coi là logic).

Tiến trình kiểm thử trở nên ngày một được tự động hoá hơn, và thường bao gồm cả dụng cụ hỗ trợ cho việc xác định và tiến hành kiểm thử cũng như việc quản trị toàn bộ tiến trình kiểm thử. Có một số kiểu kiểm thử khác nhau. Kiểm thử đơn vị là kiểm thử cho một cấu phần hay một tập hợp các cấu phần, thường được tiến hành bởi người phát triển cấu phần đó. Kiểm thử tích hợp là kiểm thử một trong các gói đã được gắn với nhau, trong đó giao diện của các gói và sự hợp tác của chúng là hợp lệ. Kiểm thử tích hợp về cơ bản được tiến hành bởi tổ kiểm thử tích hợp chuyên môn, có các thành viên có tri thức sâu về kiến trúc của hệ thống. Kiểm thử hệ thống là kiểm thử chức năng của toàn bộ hệ thống như được người dùng xem xét, và về cơ bản được tiến hành với đặc tả yêu cầu (với trường hợp sử dụng) làm cơ sở. Việc kiểm thử hệ thống kiểm chứng rằng hệ thống chuyển giao chức năng đặc biệt cho người dùng cuối. Một biến thể của kiểm thử hệ thống là kiểm thử chấp nhận, vốn được tiến hành bởi người nhận hệ thống để xác định liệu có chấp nhận phiên bản bàn giao hay không.

Kiểm thử hồi qui không phải là pha kiểm thử mới, mà là kỹ thuật để giải quyết những thay đổi trong hệ thống. Kiểm thử hồi qui được cho chạy sau khi những thay đổi đã được tiến hành cho hệ thống; nó thực tế là một loạt các kiểm thử được chạy trên toàn hệ thống để xác định liệu chức năng nào khác đã bị ảnh hưởng không đúng bởi thay đổi đó. Điều rất thông thường là việc thay đổi ở phần này của hệ thống lại gây ra hiệu quả không mong muốn ở phần nào đó khác (mặc dầu rủi ro này giảm đi nhiều với công nghệ hướng sự vật). Kiểm thử hồi qui liên tục sẽ làm lộ ra những vấn đề như vậy. Kiểm thử hồi qui thường được tự động hoá; chúng được một dụng cụ cho chạy, và dụng cụ này báo hiệu bất kì khác biệt nào trong quan hệ với kết quả dự kiến.

Các biểu đồ trường hợp sử dụng được tạo ra trong phân tích yêu cầu được dùng trong pha kiểm thử để kiểm chứng lại hệ thống. Các biểu đồ triển khai, trình tự và hợp tác được tạo ra trong phân

tích và thiết kế về cơ bản được dùng làm cơ sở cho kiểm thử tích hợp.

Dụng cụ kiểm thử

Một cách lí tưởng, các dụng cụ được dùng trong tiến trình phát triển nên hỗ trợ cho ngôn ngữ mô hình hoá và ngôn ngữ lập trình, cũng như tiến trình. Ngày nay, phần lớn các dụng cụ đều dựa nhiều vào sự hỗ trợ cho một ngôn ngữ lập trình, mặc dầu các dụng cụ mới đang được xây dựng để hỗ trợ cho ngôn ngữ mô hình hoá trực quan như UML. Dẫu vậy, việc hỗ trợ cho tiến trình này vẫn thiếu trong phần lớn các dụng cụ (mà có thể được giải thích bởi sự kiện là khó định nghĩa các tiến trình tổng quát vốn có thể được mọi người dùng hơn là định nghĩa ngôn ngữ lập trình tổng quát và ngôn ngữ mô hình hoá).

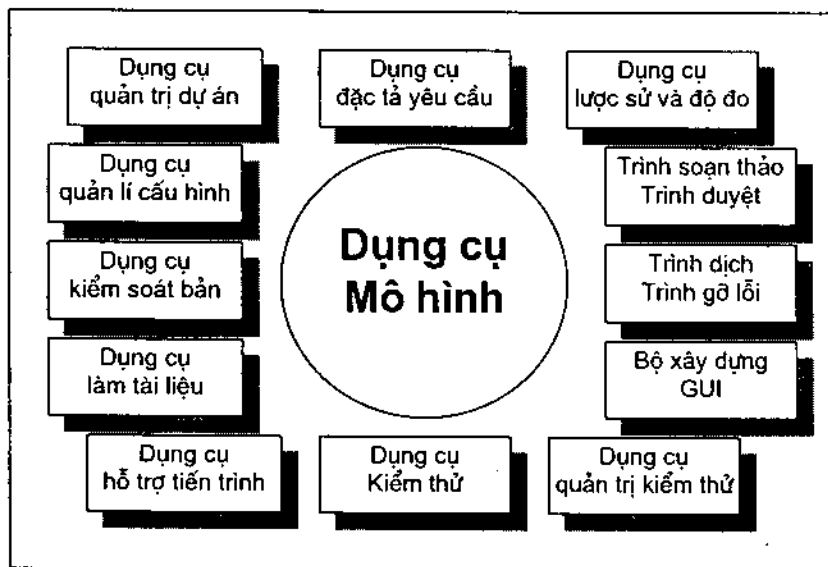
Với sự hỗ trợ cho một tiến trình trong một dụng cụ, các tính năng sau được dự tính sẵn:

- *Tri thức về các pha tiến trình:* Dụng cụ nên biết về các pha của tiến trình; và nếu dụng cụ được dùng trong nhiều pha thì nó nên thích ứng hành vi của mình và cung cấp sự hỗ trợ cho pha mà trong đó nó hiện được dùng.
- *Trợ giúp trực tuyến và hỗ trợ hướng dẫn:* Dụng cụ nên có khả năng cung cấp sự hỗ trợ trực tuyến, trích dẫn một danh sách các hoạt động cần thực hiện trong pha hiện tại, cùng với những hướng dẫn về cách làm chúng.
- *Hỗ trợ cho phát triển lặp:* Dụng cụ nên có khả năng lặp lại công việc, nơi mà dụng cụ hỗ trợ cho cả loạt việc lặp. Trong dụng cụ mô hình hoá, điều này có thể có nghĩa là hỗ trợ cho việc sinh mã và kĩ nghệ ngược cho mã đã sửa đổi.
- *Hỗ trợ tổ:* Dụng cụ nên hỗ trợ cho công việc tổ, làm cho từng thành viên tổ làm việc về phần mình mà không quấy rối người khác.

- *Kho hệ thống chung*: Dụng cụ nên có khả năng chia sẻ một kho chung với các dụng cụ khác, để cho tất cả các dụng cụ đều có hình ảnh toàn thể chung về hệ thống.
- *Tích hợp với các dụng cụ khác*: Nên có khả năng tích hợp dụng cụ này một cách dễ dàng với các dụng cụ khác.

Một cách tự nhiên một số dụng cụ có nhiệm vụ rất đặc thù và được đưa vào chỉ một pha hay một hoạt động. Trong trường hợp đó, dụng cụ này không cần tất cả mọi khả năng được mô tả ở đây, nhưng nó nên hỗ trợ cho cách thức mà pha này hay hoạt động này thường được thực hiện; chẳng hạn, một cách lặp lại, với cái vào từ pha trước, kiểm tra tính nhất quán vùn vùn. Hình B.23 vẽ các dụng cụ thường được đưa vào trong tiến trình phát triển. Chúng là:

- *Môi trường ngôn ngữ*: Trình soạn thảo, trình biên dịch và gỡ lỗi cho ngôn ngữ lập trình được chọn.
- *Kiểm soát cấu hình và phiên bản*: Dụng cụ để giải quyết các cấu hình và phiên bản khác nhau của sản phẩm, kể cả hỗ trợ cho việc giải quyết phát triển tương tranh bởi nhiều người phát triển.
- *Dụng cụ làm tư liệu*: Hỗ trợ cho việc sản xuất tự động và dễ dàng các tài liệu phát triển hay người dùng.
- *Dụng cụ quản lý dự án*: Giúp đỡ cho người quản lý dự án và theo dõi tiến trình phát triển.
- *Dụng cụ kiểm thử*: Hỗ trợ cho các phép kiểm thử khác nhau và cho việc quản trị tiến trình kiểm thử.
- *Bộ xây dựng giao diện người dùng đồ họa GUI*: Dụng cụ để xây dựng và làm bản mẫu cho giao diện người dùng và để triển khai chúng trong hệ thống.
- *Dụng cụ đặc tả yêu cầu*: Hỗ trợ các dụng cụ để thu nhận và mô tả các yêu cầu về hệ thống.



Hình B.23 Dụng cụ được dùng trong tiến trình phát triển.

Tài liệu tham khảo

1. Kenneth C. Laudon, Jane P. Laudon, Management Information Systems, Prentice Hall, 1995
2. James I. Cash, Robert G. Eccles, Nitin Nohria, Richard L. Nolan, Building the Information-Age Organization: Structure, Control, and Information Technologies, IRWIN, 1994
3. Roger S. Pressman, Software Engineering, McGraw-Hill Inc., 1994 - Bản dịch của Ngô Trung Việt, Kỹ nghệ phần mềm, NXB Giáo dục, Tập 1 - 1998, Tập 2- 1999, Tập 3 - 2000.
4. Grady Booch, Object-Oriented Analysis and Design with Applications, The Benjamin/Cumming Publishing Company, Inc., 1994
5. Hans-Erik Eriksson, Magnus Penker, UML Toolkit, John Wiley&Sons, Inc., 1998
6. Steve Cook, John Daniels, Designing Object Systems, Prentice Hall, 1994
7. Ngô Trung Việt, Phân tích và thiết kế tin học hệ thống quản lý - kinh doanh - nghiệp vụ, NXB Giao thông vận tải, 1994
8. Louis Rigand, Thiết lập hệ thống thông tin phục vụ chỉ đạo và quản lý các tổ chức, Dunod 1984, bản dịch tiếng Việt, năm 1988.

PHÁT TRIỂN HỆ THỐNG THÔNG TIN

Góc nhìn của người quản lí

Biên soạn: Ngô Trung Việt

Chịu trách nhiệm xuất bản: PGS. TS TÔ ĐĂNG HẢI

Biên tập: NGUYỄN HUY TIẾN

Sửa bài: QUANG HUY

Trình bày bìa: HƯƠNG LAN

Nhà xuất bản Khoa học và Kỹ thuật

70 Trần Hưng Đạo - Hà Nội

In 1000 cuốn, khổ 14,5 x 20,5 cm, tại Nhà in Đại học Quốc gia Hà Nội

Giấy phép xuất bản số: 123-209-16/5/2001

In xong và nộp lưu chiểu tháng 7 năm 2001

201105

phát triển hệ thống thông tin



1 001102 300233
35.000 VNĐ

Giá: 35.000đ