

LỜI NÓI ĐẦU

Những năm 60 của thế kỷ trước, thế giới chưa định hình những pháp rõ rệt cho phân tích và thiết kế hệ thống. Người ta xây dựng hệ thống một cách tùy tiện, theo sở thích và kinh nghiệm cá nhân. Kết quả phân tích là những tài liệu trình bày dưới dạng văn tự dài dòng, khó đọc và khó trao đổi. Từ những năm 70 tới nay, nhiều phương pháp phân tích và thiết kế lần lượt ra đời. Mỗi phương pháp đều có ưu và nhược điểm riêng, có thể được ưu chuộng ở nơi này, nhưng lại được ít ưu chuộng ở nơi khác. Sự phong phú và đa dạng trong phương pháp như vậy cũng có nghĩa là sự không thống nhất, không chuẩn hóa.

Trải qua thời gian, một số phương pháp đã tỏ ra là có một sức sống dẻo dai, bám trụ cho đến tận ngày nay. Trong số này các phương pháp được gọi dưới một tên chung là các phương pháp có cấu trúc¹ được quan tâm và được nói đến nhiều nhất bởi:

- Phương pháp có cấu trúc đã chứng tỏ được tính kinh điển của nó qua thời gian. Học nó trước hết là học cách tư duy nhất quán và chặt chẽ của nó.
- Phương pháp có cấu trúc là phương pháp dung dị, không cầu kỳ như một số phương pháp khác, dễ áp dụng, nhưng lại rất hữu hiệu. Ngày nay nó chưa lạc hậu, mà vẫn phát huy tác dụng tốt. Bằng chứng là một hệ thống lớn và hiện đại như Oracle vẫn tiếp tục sử dụng và phát triển nó.

Bài giảng này đề cập đến phương pháp phân tích thiết kế hệ thống thông tin hướng cấu trúc theo cách tiếp cận phổ biến nhất. Bài giảng này là tài liệu học tập cho sinh viên ngành Công nghệ thông tin Trường Đại học Giao thông Vận tải ở học kỳ 5. Bài giảng là tài liệu tham khảo hữu ích cho sinh viên các chuyên ngành khác và những người quan tâm đến phương pháp phân tích thiết kế hướng cấu trúc.

Mọi ý kiến đóng góp xin gửi về: Bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin, Trường Đại học Giao thông Vận tải, Láng Thượng – Đống Đa – Hà Nội.
Email: bmcpm@utc.edu.vn.

¹ Dưới tên các phương pháp có cấu trúc có nhiều phương pháp của các tác giả hay nhóm tác giả khác nhau. Các phương pháp này nhất quán trong cùng một phương hướng tư duy.

MỤC LỤC

Chương 1: TỔNG QUAN VỀ HỆ THỐNG và HỆ THỐNG THÔNG TIN.....	5
1.1 Định nghĩa Hệ thống.....	5
1.2 Các khái niệm liên quan đến hệ thống	6
1.3 Khái niệm hệ thống thông tin	8
1.4. Một quy trình phát triển hệ thống đơn giản.....	11
Chương 2. PHÁT TRIỂN HỆ THỐNG THÔNG TIN	19
2.1. Quy trình phát triển hệ thống.....	19
2.2. Một quy trình phát triển hệ thống	23
2.3. Các chiến lược phát triển hệ thống	28
2.4. Các kỹ thuật và công cụ tự động hóa	30
2.5. Vai trò của những người tham gia phát triển HTTT	33
2.6. Xây dựng thành công hệ thống thông tin	34
Chương 3. TỔNG QUAN VỀ PHÂN TÍCH HỆ THỐNG	36
3.1. Khái niệm phân tích hệ thống.....	36
3.2. Các hướng tiếp cận phân tích hệ thống	36
3.3. Các giai đoạn phân tích hệ thống	37
3.4. Xác định các yêu cầu của người dùng	42
Chương 4. CÁC PHƯƠNG PHÁP THU THẬP THÔNG TIN	46
4.1. Phương pháp phỏng vấn	46
4.2. Phương pháp dùng phiếu hỏi.....	49
4.3. Phương pháp lấy mẫu	51
4.4. Phân tích tài liệu định lượng/định tính	52
4.5. Phương pháp quan sát.....	53
4.6 Các công việc sau khảo sát.....	55
Chương 5. MÔ HÌNH HÓA CHỨC NĂNG	59
5.1. Mô hình hóa hệ thống.....	59
5.2. Mô hình logic.....	60
5.3. Biểu đồ phân rã chức năng	61

5.4. Biểu đồ luồng dữ liệu (DFD).....	64
5.5. Các phần tử của DFD	65
5.6. Biểu đồ luồng dữ liệu mức ngữ cảnh	69
5.7. Trình tự và quy tắc xây dựng DFD.....	70
Chương 6. MÔ HÌNH HÓA DỮ LIỆU	76
6.1. Mô hình hóa dữ liệu.....	76
6.2. Các phần tử của biểu đồ quan hệ thực thể (ERD)	76
6.3. Xây dựng biểu đồ quan hệ thực thể.....	81
6.4. Xây dựng biểu đồ dữ liệu quan hệ (RDM).....	86
6.5. Từ điển dữ liệu.....	92
Chương 7. TỔNG QUAN VỀ THIẾT KẾ HỆ THỐNG.....	95
7.1. Các hướng tiếp cận thiết kế hệ thống	95
7.2. Các công việc cụ thể trong giai đoạn thiết kế.....	96
Chương 8. KIẾN TRÚC ỨNG DỤNG VÀ VIỆC MÔ HÌNH HÓA	98
8.1. Kiến trúc ứng dụng	98
8.2. Biểu đồ luồng dữ liệu vật lý	98
8.3. Kiến trúc công nghệ thông tin	100
Chương 9. THIẾT KẾ CƠ SỞ DỮ LIỆU	102
9.1. Các phương thức lưu trữ dữ liệu.....	102
9.2. Kiến trúc dữ liệu	102
9.3. Triển khai mô hình dữ liệu logic dựa trên một cơ sở dữ liệu quan hệ	103
Chương 10. THIẾT KẾ ĐẦU VÀO	105
10.1. Tổng quan về thiết kế đầu vào.....	105
10.2. Các điều khiển giao diện cho thiết kế đầu vào	106
10.3. Quy trình thiết kế đầu vào	107
Chương 11. THIẾT KẾ ĐẦU RA	108
11.1. Tổng quan về thiết kế đầu ra	108
11.2. Cách thức thiết kế đầu ra	108
Chương 12. THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG	112

12.1. Tổng quan về giao diện người dùng	112
12.2. Kỹ thuật giao diện người dùng	113
12.3. Các phong cách thiết kế giao diện người dùng	114
12.4. Cách thức thiết kế giao diện người dùng.....	119
TÀI LIỆU THAM KHẢO	121

Chương 1: TỔNG QUAN VỀ HỆ THỐNG và HỆ THỐNG THÔNG TIN

1.1 Định nghĩa Hệ thống

Chúng ta thường dùng chữ hệ thống để nói đến một cái gì đó phức tạp, có nhiều bộ phận có quan hệ với nhau và có nhiều cách hiểu khác nhau về thuật ngữ “hệ thống”. Trên thực tế có nhiều định nghĩa khác nhau về Hệ thống. Ta xem xét 2 định nghĩa sau:

Định nghĩa có tính mô tả [1] được xem là đầy đủ hơn cả:

(1) Hệ thống là một tập hợp bao gồm 3 thành phần chính sau:

- *Các phần tử (elements)*: không cần phân biệt bản chất của nó là cái gì.
- *Các mối quan hệ (relationships)* giữa các phần tử đó. Các mối quan hệ thường có nhiều dạng khác nhau. Có thể kể ra một vài dạng quan hệ tiêu biểu như mối quan hệ cơ học giữa các bộ phận trong một cái máy, quan hệ năng lượng giữa mặt trời và các hành tinh của nó, quan hệ thông tin giữa các tổ chức, ... Các mối quan hệ thường được đặc trưng bằng tên gọi, quy mô, chiều hướng và mức độ của chúng.
- Các mối quan hệ liên kết các phần tử với nhau *tạo thành một thể thống nhất* xác định một cơ chế vận hành để có được 1 chức năng hay mục tiêu của hệ thống mà không một phần tử riêng lẻ nào của nó có thể có được. Thành phần (Đặc trưng) này còn gọi là *tính trời* của hệ thống, vì rằng chỉ có hệ thống do các phần tử liên kết lại thành một thể thống nhất mới tạo ra đặc trưng này.

Ví dụ: Hệ thống mặt trời có các hành tinh là các phần tử của nó, các mối quan hệ là lực hấp dẫn giữa chúng với nhau. Chiếc đồng hồ cơ học là một hệ thống cơ học gồm có các phần tử là các bánh xe, dây cót, kim,... được gắn kết với nhau bằng các quan hệ như các liên kết cơ học và tạo nên chức năng chỉ giờ.....

Một định nghĩa khác xem hệ thống như quá trình xử lý [1]

(2) Hệ thống là tập hợp các phần tử có quan hệ qua lại với nhau cùng hoạt động hướng đến một mục tiêu chung thông qua việc tiếp nhận cái vào và sản sinh cái ra nhờ một quá trình chuyển đổi được tổ chức.

Định nghĩa này là một trường hợp cụ thể của định nghĩa (1) và đúng với hầu hết các hệ thống động.

1.2. Các khái niệm liên quan đến hệ thống

- **Môi trường** (*Environment*): Mọi hệ thống không tồn tại trong trống không, mà luôn tồn tại và hoạt động trong một hệ thống lớn hơn. Hệ thống lớn hơn không kể hệ thống được xét gọi là *môi trường* của nó. Như vậy, môi trường có thể xem là phần còn lại của thế giới sau khi đã “tách” hệ thống khỏi nó. Ví như: môi trường của một xí nghiệp là khách hàng, các tổ chức cung cấp vốn, các tổ chức cung cấp nguyên vật liệu và năng lượng, nguồn năng lượng, nguồn lao động và cả các cơ quan thuế và quản lý thị trường. Trên thực tế, người ta không xem cả phần còn lại của thế giới là môi trường của một hệ thống cụ thể, mà xem môi trường chỉ là những gì không thuộc nó và có quan hệ trực tiếp hay thực sự ảnh hưởng lên chính hệ thống đó.
- **Ranh giới** (*boundary*): Để nhận biết ra môi trường của một hệ thống hay phân biệt nó với môi trường ta cần phải xác định được đâu là *ranh giới* của hệ thống. Nói cách khác là phạm vi của hệ thống đến đâu. Trên thực tế, đối với một hệ thống bất kỳ không phải luôn dễ dàng nhận ra phạm vi hay ranh giới của nó. Một trong những cách để xác định phạm vi của một hệ thống là hãy kể ra các phần tử và các mối quan hệ của nó.
- **Hệ thống tĩnh** (*static system*): Một hệ thống mà tất cả các thành phần tạo nên chúng hầu như không thay đổi theo không gian và thời gian gọi là hệ thống tĩnh. Ví dụ: hệ thống đường xá, hệ thống thiết bị khi không vận hành,
- **Hệ thống động** (*dynamic system*): Hệ thống động là hệ thống có sự thay đổi đáng kể trong không gian và thời gian về các mặt hình dáng, cấu trúc, đặc tính, vị trí,.. của nó. Các hệ thống như các sinh vật, các thiết bị đang vận hành, ... là những hệ thống động.
- **Trạng thái của hệ thống** (*system state*): Đối với một hệ thống động, người ta thường xác định một bộ đặc trưng của nó để phản ánh sự thay đổi của hệ thống

trong không gian và thời gian. Với một bộ giá trị cụ thể của các đặc trưng của hệ thống ở một thời điểm được gọi là một trạng thái của hệ thống. Xét một vật rơi từ trên cao xuống, *khoảng cách* từ vật đến mặt đất có thể xem là một đặc trưng trạng thái của nó.

- **Hệ thống con** (*subsystem*): Một hệ thống lại có thể phân thành những thành phần nhỏ hơn có được mục tiêu hay chức năng riêng của nó. Một phần như vậy là *hệ thống con* của hệ thống đã cho. Một phân xưởng sản xuất là một hệ con của một xí nghiệp sản xuất, động cơ là một hệ con của một ô tô.
- **Phân rã** (*decomposition*): Một hệ thống có các hệ thống con. Đến lượt mình, hệ thống con lại có thể gồm các hệ thống con của nó. Nhờ có đặc tính này mà ta có thể phân chia một hệ thống thành những hệ thống con của nó. Tiếp đó, ta lại phân chia các hệ thống con của nó thành những hệ thống con nhỏ hơn hay thành phần tử của hệ thống. Quá trình này gọi là *phân rã hệ thống*. Phân rã là một công cụ chủ yếu để phân tích hệ thống.
- **Môđun hóa** (*modularity*): Môđun được hiểu là một phần tử của một hệ thống. Môđun hóa là một kết quả trực tiếp của sự phân rã một hệ thống. Nó đề cập đến việc phân chia một hệ thống thành những phần nhỏ với một quy mô tương đối đồng nhất. Môđun hóa thường được sử dụng để phân chia và tổ chức một hệ thống chương trình máy tính thành các thành phần của chúng.
- **Giao diện** (*interface*): Các hệ thống con trong một hệ thống có thể tương tác với nhau, cũng như hệ thống có thể tương tác với môi trường. Chỗ mà tại đó hai hệ thống tương tác với nhau hay tương tác với môi trường gọi là giao diện của nó.
- **Cái vào, cái ra** (*Input, Output*): Khi một hệ thống tương tác với một hệ thống khác (hay môi trường), những cái mà nó nhận được gọi là *cái vào* và những cái mà hệ thống khác nhận được từ nó gọi là *cái ra* của nó.
- **Hệ thống mở, hệ thống đóng** (*open system, closed system*): Một hệ thống có tác động qua lại với các hệ thống khác hay môi trường gọi là *hệ thống mở*. Hệ thống không phải là mở gọi là *hệ thống đóng*.

Ngoài các khái niệm trên ta còn có thể có các khái niệm: hệ thống thích nghi, hệ có điều khiển, ghép nối, kết dính, hệ thống logic, hệ thống vật lý.

1.3 Khái niệm hệ thống thông tin

- **Thông tin** là một loại tài nguyên của tổ chức, phải được quản lý chu đáo giống như mọi tài nguyên khác. Việc xử lý thông tin đòi hỏi chi phí về thời gian, tiền bạc và nhân lực. Việc xử lý thông tin phải hướng tới khai thác tối đa tiềm năng của nó. Thông tin có rất nhiều cách định nghĩa khác nhau.
- **Hệ thống thông tin** (Information System - IS) trong một tổ chức có chức năng thu nhận và quản lý dữ liệu để cung cấp những thông tin hữu ích nhằm hỗ trợ cho tổ chức đó và các nhân viên, khách hàng, nhà cung cấp hay đối tác của nó. Ngày nay, nhiều tổ chức xem các hệ thống thông tin là yếu tố thiết yếu giúp họ có đủ năng lực cạnh tranh và đạt được những bước tiến lớn trong hoạt động. Hầu hết các tổ chức nhận thấy rằng tất cả nhân viên đều cần phải tham gia vào quá trình phát triển các hệ thống thông tin. Do vậy, phát triển hệ thống thông tin là một chủ đề ít nhiều có liên quan tới bạn cho dù bạn có ý định học tập để trở nên chuyên nghiệp trong lĩnh vực này hay không.
- Về mặt kỹ thuật, HTTT được xác định như một tập hợp các thành phần được tổ chức để thu thập, xử lý, lưu trữ, phân phối và biểu diễn thông tin trợ giúp việc ra quyết định và kiểm soát trong một tổ chức. [1]
- Một cách khác, Hệ thống thông tin là một hệ thống bao gồm con người, dữ liệu, các quy trình và công nghệ thông tin tương tác với nhau để thu thập, xử lý, lưu trữ và cung cấp thông tin cần thiết ở đầu ra nhằm hỗ trợ cho một hệ thống. [3]
- Hệ thống thông tin hiện hữu dưới mọi hình dạng và quy mô.

Phân loại hệ thống thông tin:

Các hệ thống thông tin có thể được phân loại theo các tiêu chí khác nhau như: quy mô, đặc tính kỹ thuật,... hay theo chức năng chúng phục vụ:

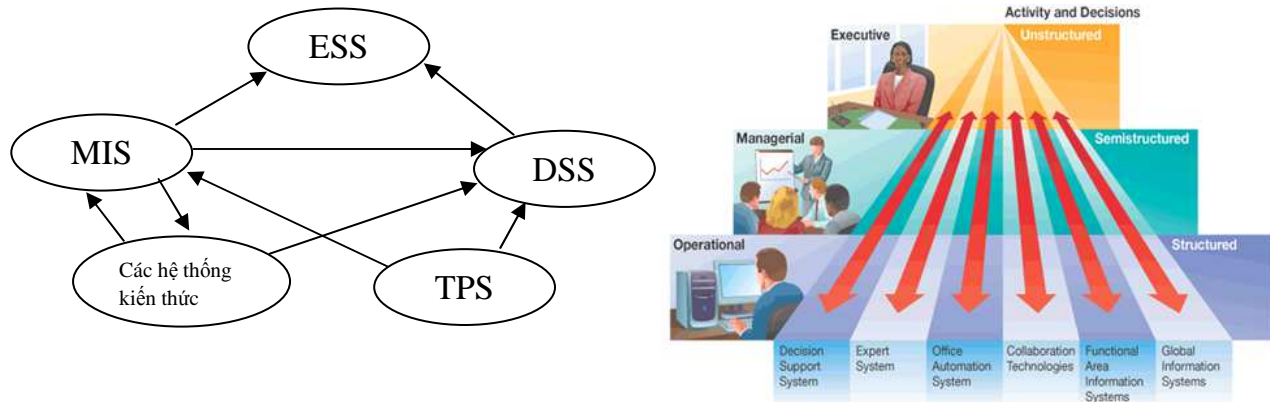
- **Hệ thống xử lý giao dịch** (Transaction processing system – TPS) là một hệ thống thông tin có chức năng thu thập và xử lý dữ liệu về các giao dịch nghiệp vụ. Có

thể hiểu nó là hệ thống máy tính để thực hiện việc ghi nhận các giao dịch hàng ngày cần thiết cho hoạt động nghiệp vụ của tổ chức để giao dịch với khách hàng, nhà cung cấp,...

- **Hệ thống thông tin quản lý** (Management information system - MIS) là một hệ thống thông tin cung cấp thông tin cho việc báo cáo hướng quản lý dựa trên việc xử lý giao dịch và các hoạt động của tổ chức. Hệ thống này trợ giúp các hoạt động quản lý của tổ chức như lập kế hoạch, kiểm tra thực hiện, tổng hợp và làm báo cáo, làm các quyết định quản lý trên cơ sở các quy trình thủ tục cho trước.
- **Hệ thống hỗ trợ quyết định** (Decision support system – DSS) là một hệ thống thông tin vừa có thể trợ giúp xác định các thời cơ ra quyết định, vừa có thể cung cấp thông tin để trợ giúp việc ra quyết định.
- **Hệ thống thông tin điều hành** (Executive Support system – ESS) là một hệ thống thông tin hỗ trợ nhu cầu lập kế hoạch và đánh giá của các nhà quản lý điều hành.
- **Hệ thống chuyên gia** (Expert System) là hệ thống thông tin thu thập tri thức chuyên môn của các chuyên gia rồi mô phỏng tri thức đó nhằm đem lại lợi ích cho người sử dụng bình thường. Sự khác biệt cơ bản của hệ chuyên gia với hệ hỗ trợ quyết định là ở chỗ: Hệ chuyên gia yêu cầu những thông tin xác định (từ người dùng) được đưa vào để đưa ra quyết định có chất lượng cao trong một lĩnh vực hẹp. Còn với hệ hỗ trợ quyết định, tùy nhu cầu mà người dùng đưa thông tin vào để có được những giải pháp trợ giúp tương ứng với phạm vi rộng.
- **Hệ thống truyền thông và cộng tác** (Communication and collaboration system) là một hệ thống thông tin làm tăng hiệu quả giao tiếp giữa các nhân viên, đối tác, khách hàng và nhà cung cấp để củng cố khả năng cộng tác giữa họ. Hệ truyền thông giúp cho việc thực hiện các trao đổi thông tin giữa các thiết bị dưới các hình thức khác nhau ở khoảng cách xa được nhanh chóng, dễ dàng và có chất lượng.
- **Hệ thống tự động văn phòng** (Office automation system) là một hệ thống thông tin hỗ trợ các hoạt động nghiệp vụ văn phòng nhằm cải thiện luồng công việc giữa các nhân viên. Ví dụ như: hệ xử lý văn bản, hệ thư tính điện tử, hệ thống lập lịch làm việc, bảng tin,....

Ngoài các hệ trên, ta còn có thể gặp: hệ cung cấp thông tin thực hiện, hệ trợ giúp làm việc theo nhóm, ...

Mối liên hệ giữa các hệ thống thông tin



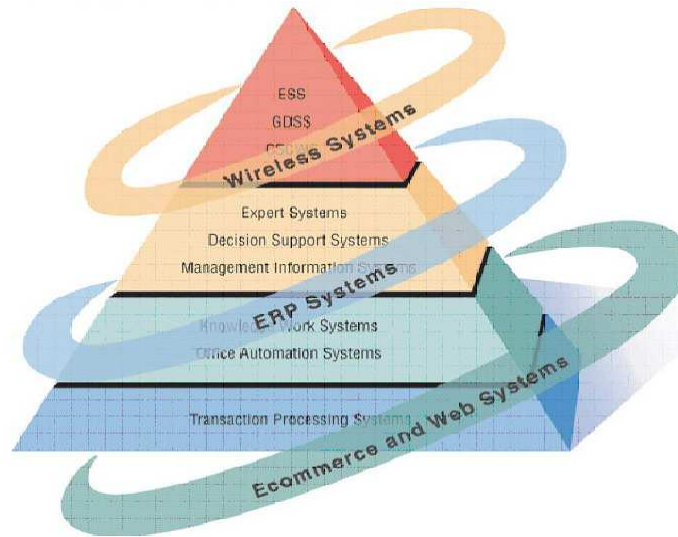
Hình 1.1: Mối quan hệ giữa các HTTT trong một tổ chức

Các loại HTTT trong một tổ chức không làm việc độc lập mà có một sự phụ thuộc lẫn nhau: Các hệ xử lý giao dịch thường là nơi sản sinh ra các thông tin cho các hệ thống khác sử dụng. Đến lượt mình mỗi hệ này lại sản sinh ra các thông tin cho các hệ khác.

Các công nghệ mới:

Các công nghệ mới đang được tích hợp vào các hệ thống truyền thống:

- Thương mại điện tử (e-commerce) sử dụng Web để thực hiện các hoạt động kinh doanh.
- Lập kế hoạch khai thác nguồn tài nguyên doanh nghiệp (ERP-Enterprise Resource Planning) có mục đích tích hợp các hệ thống thông tin khác nhau trong một tổ chức.
- Các thiết bị cảm tay và không dây, bao gồm thương mại di động (m-commerce).
- Phần mềm mã nguồn mở



Hình 1.2: Các công nghệ mới tác động tới tất cả các hệ thống

Phân tích và thiết kế hệ thống là cách tiếp cận có hệ thống tới:

- Việc xác định các vấn đề, cơ hội và mục tiêu
- Việc phân tích các luồng thông tin trong các tổ chức.
- Việc thiết kế các hệ thống thông tin trên máy tính để giải quyết vấn đề

Bài giảng này đề cập tới hai nội dung chính, một là “phân tích” những yêu cầu nghiệp vụ cho các hệ thống thông tin và hai là ”thiết kế” các hệ thống thông tin đáp ứng những yêu cầu đó. Nói một cách khác, sản phẩm của quá trình phân tích và thiết kế hệ thống chính là một hệ thống thông tin.

1.4. Một quy trình phát triển hệ thống đơn giản

Trên đây, bạn đã được giới thiệu về các loại hình hệ thống thông tin khác nhau, một số xu hướng công nghệ có ảnh hưởng tới sự phát triển của các hệ thống thông tin. Trong mục này, bạn sẽ học một khía cạnh nữa về hệ thống thông tin, đó là “quy trình” phát triển một hệ thống thông tin.

Hầu hết các quy trình phát triển hệ thống của các tổ chức đều hướng theo cách tiếp cận giải quyết vấn đề (problem-solving). Cách tiếp cận này thường kết hợp các bước giải quyết vấn đề nói chung sau:

1. Xác định vấn đề
2. Phân tích và hiểu vấn đề
3. Xác định các yêu cầu giải pháp
4. Xác định các giải pháp khác nhau và chọn cách “tốt nhất”
5. Thiết kế giải pháp đã lựa chọn

6. Cài đặt giải pháp đã lựa chọn
7. Đánh giá kết quả (nếu vấn đề vẫn không được giải quyết thì quay lại bước 1 hoặc 2)

Nhằm mục đích đơn giản, chúng tôi sẽ trình bày cách tiếp cận giải quyết vấn đề ban đầu gồm bốn giai đoạn hoặc pha cần phải được hoàn thành đối với bất kỳ một dự án phát triển hệ thống nào – đó là pha khởi đầu hệ thống, phân tích hệ thống, thiết kế hệ thống và cài đặt hệ thống. Bảng dưới đây thể hiện quan hệ giữa các bước giải quyết vấn đề nói chung và quy trình mà chúng tôi trình bày.

Quy trình phát triển hệ thống đơn giản hóa	Các bước giải quyết vấn đề nói chung
Khởi đầu hệ thống	1. Xác định vấn đề. (Đồng thời lập kế hoạch cho giải pháp của vấn đề).
Phân tích hệ thống	1. Phân tích và hiểu vấn đề . 2. Xác định các yêu cầu giải pháp.
Thiết kế hệ thống	1. Xác định các giải pháp khác nhau và chọn cách “tốt nhất” 2. Thiết kế giải pháp đã lựa chọn
Cài đặt hệ thống	1. Cài đặt giải pháp đã lựa chọn 2. Đánh giá kết quả. (Nếu vấn đề vẫn không được giải quyết thì quay lại bước 1 hoặc 2).

Bảng 1-1 Quy trình phát triển hệ thống

Cần lưu ý là bất cứ quy trình phát triển hệ thống nào cũng phải được quản lý trên cơ sở dự án. Phải có ít nhất một nhân sự nhận trách nhiệm làm người quản lý dự án để đảm bảo rằng hệ thống được phát triển đúng thời gian, trong giới hạn ngân sách cho phép và có chất lượng chấp nhận được. Hoạt động quản lý một dự án được gọi là quản lý dự án

Quản lý dự án (Project Management) là hoạt động xác định, lập kế hoạch, điều khiển, kiểm soát một dự án để phát triển một hệ thống chấp nhận được trong khoảng thời gian và ngân sách được giao

Quản lý quy trình (Process Management) là hoạt động liên tục nhằm xác định, cải thiện và kết hợp việc sử dụng phương pháp luận mà tổ chức đã lựa chọn (“quy trình”) với các tiêu chuẩn đối với mọi dự án phát triển hệ thống.

1.2.1. Khởi đầu hệ thống

Các dự án hệ thống thông tin thường phức tạp. Chúng đòi hỏi sự đầu tư, nỗ lực và thời gian đáng kể. Các vấn đề cần giải quyết thường được phát biểu một cách mơ hồ, có nghĩa rằng giải pháp được hình dung ban đầu có thể còn chưa hoàn thiện. Vì vậy, các dự án hệ thống phải được lập kế hoạch cẩn thận. Giai đoạn khởi đầu hệ thống hình thành phạm vi dự án và kế hoạch giải quyết vấn đề. Do đó, pha khởi đầu hệ thống thiết lập phạm vi dự án, mục tiêu, lịch biểu và ngân sách cần thiết để giải quyết vấn đề.

Phạm vi dự án xác định lĩnh vực nghiệp vụ được hướng đến của dự án và các mục tiêu cần đạt được. Phạm vi và mục tiêu về cơ bản đều ảnh hưởng tới các đảm bảo về tài nguyên, cụ thể là lịch biểu và ngân sách, những nhân tố cần được thực hiện để hoàn thành dự án. Bằng việc thiết lập một ngân sách và lịch biểu dựa vào phạm vi và mục tiêu ban đầu, bạn cũng sẽ thiết lập được một ranh giới mà dựa vào đó tất cả các nhân sự đều có thể chấp nhận thực tế là bất cứ thay đổi nào trong tương lai đối với phạm vi hoặc mục tiêu cũng sẽ tác động tới lịch biểu và ngân sách.

Người quản lý dự án, người phân tích hệ thống và người sở hữu hệ thống là những nhân lực chủ yếu trong pha khởi đầu hệ thống.

Khởi đầu hệ thống (System Initiation) là việc lập kế hoạch ban đầu cho một dự án để xác định phạm vi nghiệp vụ, mục tiêu, lịch biểu và ngân sách ban đầu.

Sau khi nghiên cứu hệ thống phải đưa ra được một kế hoạch dự án cơ bản. Kế hoạch dự án này cần được phân tích khả thi trên 3 mặt:

- Khả thi kỹ thuật: xem xét khả năng kỹ thuật hiện có (về thiết bị, về công nghệ và khả năng làm chủ công nghệ) đủ đảm bảo thực hiện giải pháp của CNTT sẽ được áp dụng trong điều kiện có thể.
- Khả thi kinh tế:
 - + Khả năng tài chính của tổ chức cho phép thực hiện dự án: bao gồm nguồn vốn và số vốn có thể huy động trong thời hạn cho phép.
 - + Lợi ích mà hệ thống được xây dựng mang lại, ít nhất là đủ bù đắp chi phí ban đầu phải bỏ ra xây dựng nó.
 - + Những chi phí thường xuyên cho hệ thống hoạt động (chi phí vận hành) là chấp nhận được đối với tổ chức.
- Khả thi về thời gian: dự án được phát triển trong thời gian cho phép. Một tiến trình thực hiện dự án được chỉ ra trong giới hạn đã cho.

1.2.2. Phân tích hệ thống

Bước tiếp theo trong quy trình phát triển hệ thống mà chúng tôi trình bày là giai đoạn phân tích hệ thống. Pha này nhằm cung cấp cho đội dự án hiểu biết thấu đáo hơn về vấn đề và nhu cầu của dự án. Hiểu một cách đơn giản, lĩnh vực nghiệp vụ (phạm vi của dự án – như đã xác định trong pha khởi đầu hệ thống) có thể được nghiên cứu và phân tích để thu được những hiểu biết chi tiết hơn. Pha phân tích hệ thống yêu cầu làm việc với người sử dụng hệ thống để xác định rõ các yêu cầu nghiệp vụ đối với hệ thống sẽ được mua hoặc phát triển.

Sự hoàn thiện của pha phân tích hệ thống thường thể hiện kết quả ở nhu cầu cập nhật các kết quả đã có trước đó ở pha khởi đầu hệ thống. Việc phân tích có thể phát hiện yêu cầu phải xét lại phạm vi hoặc mục tiêu của dự án – ví dụ có thể cảm thấy phạm vi của dự án quá lớn hoặc quá nhỏ. Cuối cùng, tính khả thi của bản thân dự án trở nên đáng ngờ. Dự án có thể bị hủy bỏ hoặc có thể chuyển sang giai đoạn tiếp theo.

Người quản lý dự án, người phân tích hệ thống và người sử dụng hệ thống là những nhân lực cơ bản trong pha phân tích hệ thống.

Phân tích hệ thống (System Analysis) là việc nghiên cứu lĩnh vực vấn đề nghiệp vụ để đề xuất các cải tiến và xác định các yêu cầu nghiệp vụ cũng như thứ tự ưu tiên cho giải pháp.

1.2.3. Thiết kế hệ thống

Sau khi đã có hiểu biết về các yêu cầu nghiệp vụ của một hệ thống thông tin, ta có thể tiến hành pha thiết kế hệ thống. Trong giai đoạn này, trước tiên cần xem xét các giải pháp công nghệ khác nhau. Hiếm khi chỉ có một giải pháp cho một vấn đề.

Một khi một giải pháp đã được lựa chọn và chấp nhận, pha thiết kế hệ thống phát triển các bản đặc tả và thiết kế chi tiết được yêu cầu để cài đặt giải pháp cuối cùng. Các bản đặc tả và thiết kế chi tiết đó sẽ được dùng để cài đặt cơ sở dữ liệu, chương trình, giao diện người dùng và mạng cho hệ thống thông tin. Trong trường hợp ta lựa chọn mua phần mềm thay vì xây dựng nó thì bản thiết kế chi tiết sẽ xác định cách thức phần mềm đó được tích hợp vào hoạt động nghiệp vụ và các hệ thống thông tin khác.

Nhắc lại về các định hướng công nghệ đã trình bày ở trên, các định hướng đó sẽ ảnh hưởng chủ yếu tới quy trình thiết kế hệ thống và ra quyết định. Nhiều tổ chức xác định một kiến trúc công nghệ thông tin chung dựa trên các định hướng công nghệ đó. Nếu vậy, tất cả các pha thiết kế hệ thống cho hệ thống thông tin mới đều phải tuân theo kiến trúc công nghệ thông tin chuẩn.

Người quản lý dự án, người phân tích hệ thống và người thiết kế hệ thống là những nhân lực chính trong pha thiết kế hệ thống.

Thiết kế hệ thống (System Design) là quá trình xác định và xây dựng giải pháp kỹ thuật dựa trên máy tính cho các yêu cầu nghiệp vụ được xác định trong pha phân tích hệ thống.

1.2.4. Cài đặt hệ thống

Bước cuối cùng trong quy trình phát triển hệ thống đơn giản mà chúng tôi trình bày là cài đặt hệ thống. Pha cài đặt hệ thống xây dựng hệ thống thông tin mới và đưa nó vào hoạt động. Trong giai đoạn này, các phần cứng và phần mềm được cài đặt và sử dụng. Các phần mềm ứng dụng được mua và cơ sở dữ liệu được cài đặt và cấu hình. Các phần mềm tùy biến và cơ sở dữ liệu được xây dựng dựa trên các bản đặc tả và thiết kế chi tiết được phát triển ở pha thiết kế hệ thống.

Khi các thành phần hệ thống đã được xây dựng hoặc cài đặt thì chúng phải được kiểm thử riêng rẽ. Sau đó, toàn bộ hệ thống cũng phải được kiểm thử để đảm bảo rằng nó hoạt động chính xác và đáp ứng được các yêu cầu của người dùng. Một khi hệ thống đã được kiểm thử đầy đủ, nó phải được đưa vào hoạt động. Dữ liệu từ hệ thống trước đó có thể phải được chuyển đổi hoặc nhập vào cơ sở dữ liệu khởi đầu và người sử dụng hệ thống phải được đào tạo để sử dụng hệ thống một cách chuẩn xác. Cuối cùng, một số kế hoạch chuyển tiếp từ quy trình nghiệp vụ và hệ thống thông tin cũ có thể phải được tiến hành.

Người quản lý dự án, người phân tích hệ thống và người xây dựng hệ thống là những nhân lực chủ yếu trong giai đoạn cài đặt hệ thống.

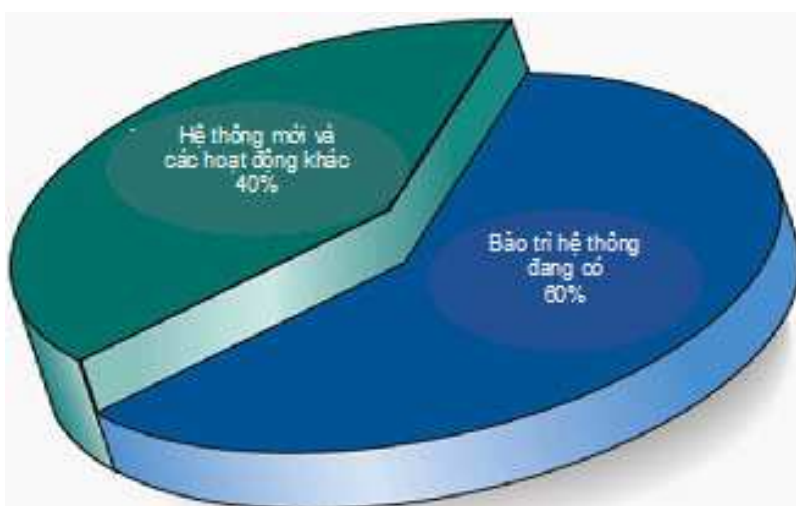
Cài đặt hệ thống (System Implementation) là giai đoạn xây dựng, cài đặt, kiểm thử và triển khai một hệ thống.

1.2.5. Bảo trì hệ thống

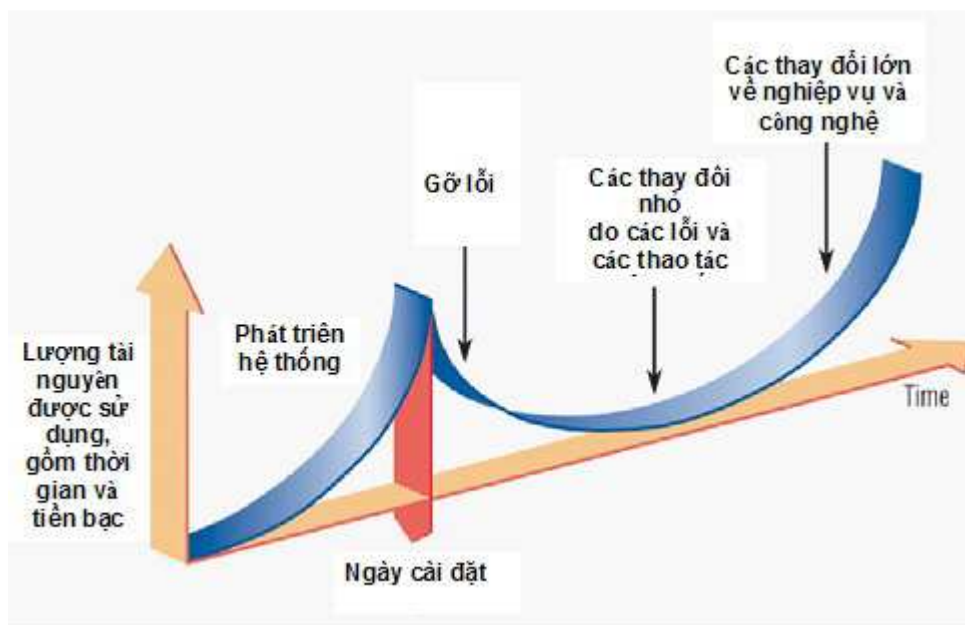
Sẽ thật thiếu sót nếu không khẳng định rằng việc cài đặt hệ thống thông tin sẽ dẫn tới việc phải đối mặt với sự tồn tại của giai đoạn hỗ trợ và cải thiện không ngừng. Các hệ thống thông tin được cài đặt rất hiếm khi hoàn hảo. Những người sử dụng sẽ tìm thấy lỗi và thỉnh thoảng bạn sẽ tìm thấy những sai sót trong thiết kế và cài đặt cần được sửa chữa. Ngoài ra, các yêu cầu nghiệp vụ và của người dùng thay đổi không ngừng. Do đó, sẽ có nhu cầu cải thiện không ngừng bất kỳ hệ thống thông tin nào tới khi nó lỗi thời.

Hỗ trợ và cải thiện hệ thống thuộc về một dự án khác, thường được gọi là dự án bảo trì và nâng cấp. Một dự án như vậy cần tuân theo cùng cách tiếp cận giải quyết vấn

đề như đã được xác định với bất kỳ dự án nào khác. Điểm khác biệt duy nhất là nỗ lực và ngân sách cần để hoàn thành dự án. Nhiều pha sẽ được hoàn thành nhanh hơn nhiều, đặc biệt là nếu nhân lực ban đầu đã tài liệu hóa một cách đúng đắn hệ thống ngay từ giai đoạn đầu. Lẽ tất nhiên, nếu họ không làm như vậy thì dự án cải thiện hệ thống có thể tiêu tốn nhiều thời gian, nỗ lực và tiền bạc hơn.



Hình 1-3: Tỷ lệ thời gian cho việc bảo trì hệ thống



Hình 1-4: Mức sử dụng tài nguyên trong quy trình phát triển hệ thống

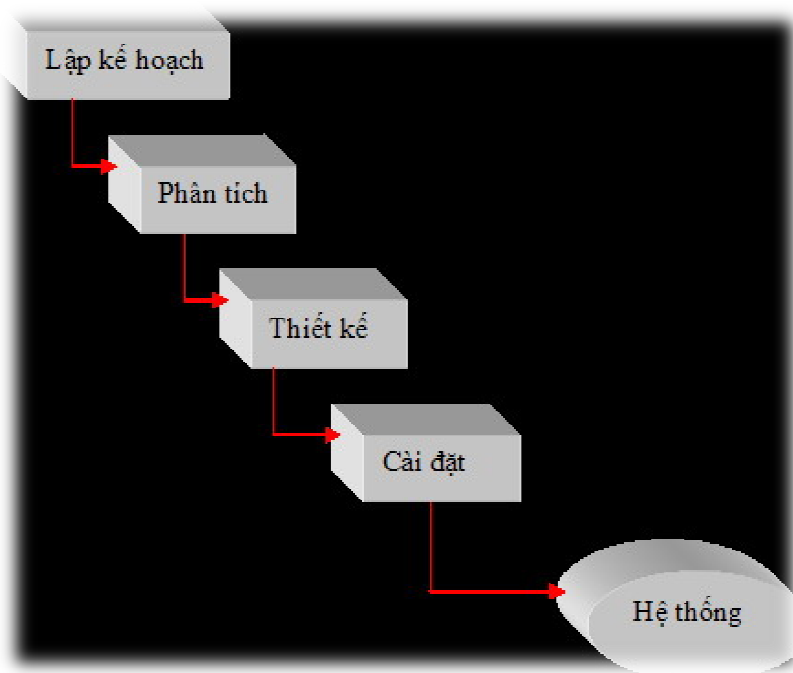
1.2.6. Phát triển tuần tự và phát triển lặp

Tất cả nội dung trình bày ở các mục trên có thể khiến bạn kết luận rằng phát triển hệ thống là một quy trình tuần tự một cách tự nhiên. Trước tiên, bạn khởi đầu dự án, rồi phân tích, thiết kế và cuối cùng là triển khai hệ thống. Điều này không phải là luôn đúng

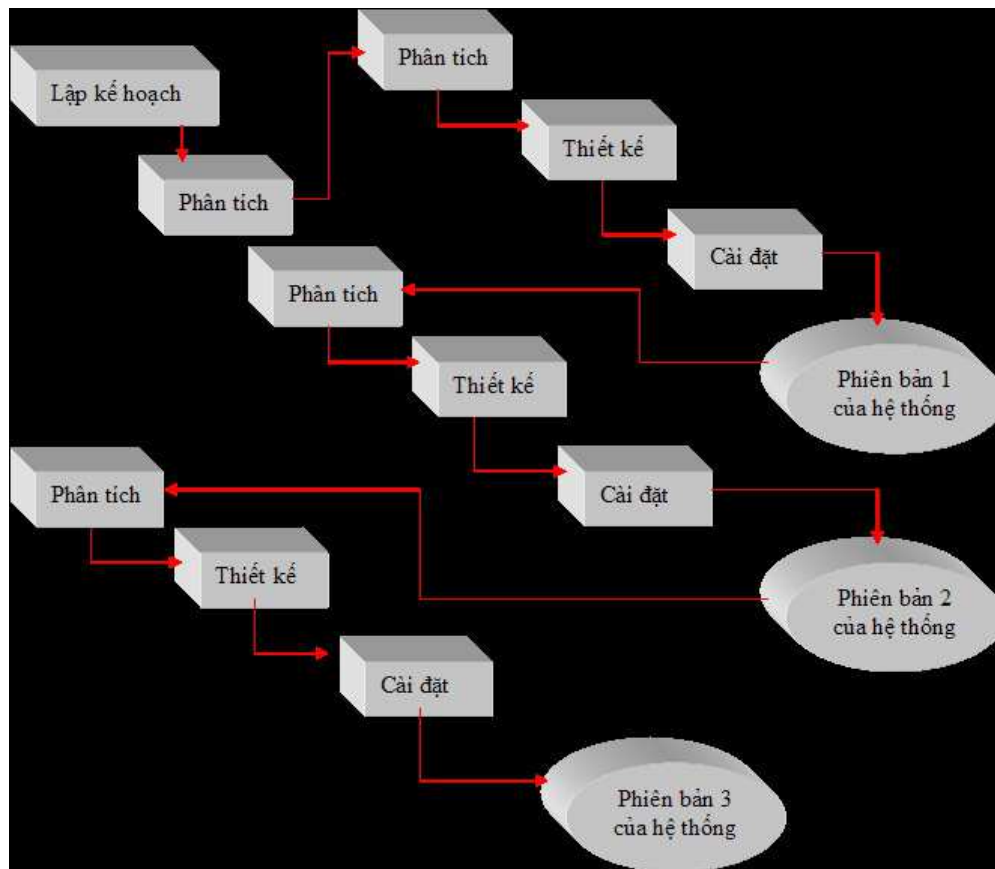
đầu. Có các chiến lược hoặc cách tiếp cận khác nhau để thực hiện quy trình phát triển hệ thống nói chung.

Rõ ràng các quy trình tuần tự là một trong các khả năng. Cách tiếp cận này được minh họa trong hình 1- 4. Chú ý rằng chiến lược này đòi hỏi mỗi pha phải được hoàn thành - cái này tiếp sau cái kia. Sự hoàn thành tuần tự sẽ cho kết quả trong sự phát triển một hệ thống hoàn toàn mới. Hình thức trực quan của cách tiếp cận này giống như một thác nước (waterfall) nên nó thường được gọi là quy trình “phát triển thác nước”. (Trong thực tế, các giai đoạn có thể chồng lấp lên nhau. Ví dụ phân thiết kế hệ thống có thể được bắt đầu trước khi hoàn thành giai đoạn phân tích hệ thống.).

Tuy nhiên, cách tiếp cận thác nước không còn được dùng phổ biến. Vì có một chiến lược phổ biến hơn, thể hiện trong hình 1-5, thường được gọi là quy trình phát triển lặp. Cách tiếp cận này đòi hỏi hoàn thành việc phân tích, thiết kế và cài đặt đủ để phát triển đầy đủ một phần của hệ thống mới và đưa nó vào hoạt động sớm nhất có thể. Một khi “phiên bản” đó của hệ thống được cài đặt, chiến lược tiếp theo là thực hiện thêm một số việc phân tích, thiết kế và cài đặt để tạo ra phiên bản tiếp theo của hệ thống. Quá trình lặp đi lặp lại tới khi tất cả các phần của hệ thống thông tin tổng thể được cài đặt. Sự phổ biến của quy trình lặp này có thể giải thích như sau: Người sở hữu và sử dụng hệ thống phàn nàn về thời gian quá dài cần để phát triển và cài đặt các hệ thống thông tin khi sử dụng cách tiếp cận thác nước. Trong khi đó, cách tiếp cận lặp cho phép đưa vào sử dụng các phiên bản với thời gian ngắn hơn. Điều này sẽ thỏa mãn đòi hỏi của khách hàng.



Hình 1-5 Phương pháp luận phát triển theo mô hình thác nước



Hình 1-6 Phương pháp luận phát triển lặp

Câu hỏi thảo luận

- 1.1. Nêu chức năng và vai trò của hệ thống thông tin trong một tổ chức.
- 1.2. Phân biệt hệ thống quản lý giao dịch (TPS) với hệ thống thông tin quản lý (MIS) và hệ thống thông tin điều hành (EIS).
- 1.3. Nêu ví dụ về hệ thống quản lý giao dịch, hệ thống thông tin quản lý và hệ thống truyền thông cộng tác.
- 1.4. Cho biết các xu thế công nghệ mới đang được đưa vào các hệ thống thông tin?
- 1.5. Nêu các giai đoạn của một quy trình phát triển hệ thống đơn giản.

GỢI Ý: Dựa vào mục 1.2, nêu tóm tắt các giai đoạn trong một quá trình đơn giản để phát triển hệ thống.

Chương 2. PHÁT TRIỂN HỆ THỐNG THÔNG TIN

2.1. Quy trình phát triển hệ thống

2.1.1. Khái niệm

Quy trình phát triển hệ thống – một tập hợp các hoạt động, phương pháp, thực nghiệm, kết quả và các công cụ tự động hóa mà các nhân sự sử dụng để phát triển và cải thiện không ngừng hệ thống thông tin và phần mềm

Một quy trình phù hợp để phát triển hệ thống phải bảo đảm:

- Hiệu quả để cho phép nhà quản lý điều chuyển nguồn lực giữa các dự án
- Tài liệu nhất quán nhằm giảm chi phí thời gian sống để bảo trì hệ thống (bởi các đội phát triển khác) về sau
- Chất lượng nhất quán xuyên suốt các dự án

2.1.2. Mô hình quản lý quy trình CMM

Capability Maturity Model (CMM) là một framework chuẩn hóa để đánh giá mức độ hoàn thiện của các quy trình phát triển hệ thống thông tin, các quy trình quản lý và các sản phẩm của một tổ chức. Mục đích của CMM là để hỗ trợ cho các tổ chức cải thiện tính hoàn chỉnh của các quy trình phát triển hệ thống. Nó bao gồm 5 mức độ hoàn thiện:

- **Mức 1—Khởi đầu:** ở mức này, các dự án phát triển hệ thống không tuân theo quy trình bắt buộc nào. Mỗi đội phát triển lại có những công cụ và phương pháp riêng. Sự thành công hay thất bại thường phụ thuộc vào kỹ năng và kinh nghiệm của đội dự án.
- **Mức 2—Có thể lặp lại:** Các quy trình quản lý và thực hiện dự án được thiết lập để theo dõi chi phí dự án, lịch biểu và tính thiết thực. Các dự án đều sử dụng một quy trình phát triển hệ thống nhưng quy trình đó có thể biến đổi phù hợp với từng dự án. Đội dự án nỗ lực phối hợp để có thể lặp lại những kết quả tốt đã đạt được. Những kinh nghiệm thực tiễn được áp dụng để chuẩn hóa quy trình cho mức kế tiếp.
- **Mức 3—Được định rõ:** Một quy trình phát triển hệ thống chuẩn (một “phương pháp luận”) được mua về hoặc được phát triển. Tất cả các dự án sử dụng một phiên bản của quy trình này để phát triển và bảo trì hệ thống thông tin và phần mềm. Nhờ việc sử dụng quy trình chuẩn mà mỗi dự án đều mang tính nhất quán về tài liệu và kết quả sản phẩm thu được.
- **Mức 4—Được quản lý:** Các mục tiêu đo được về chất lượng và hiệu quả phải được thiết lập. Các kết quả đo chi tiết về chất lượng quy trình phát triển hệ thống

chuẩn và chất lượng sản phẩm luôn được thu thập và lưu trữ vào cơ sở dữ liệu. Đội dự án dựa vào những dữ liệu đó để cải thiện việc quản lý từng dự án.

- **Mức 5—Tối ưu:** Quy trình phát triển hệ thống chuẩn được giám sát và cải thiện không ngừng dựa trên các phép đo và phân tích dữ liệu được thiết lập trong mức 4. Có thể bao gồm việc thay đổi kỹ thuật, công nghệ để thực hiện các hoạt động được đòi hỏi trong quy trình phát triển hệ thống chuẩn, cũng như việc điều chỉnh chính quy trình.

Cần nhận thấy rằng mỗi mức CMM lại là tiền điều kiện cho mức tiếp theo. Hiện tại, trên thế giới, nhiều tổ chức đang nỗ lực để đạt được ít nhất là CMM mức 3.

2.1.3. Phương pháp luận phát triển hệ thống

- **Vòng đời hệ thống** – là sự phân tích vòng đời của một hệ thống thông tin thành hai giai đoạn, (1) phát triển hệ thống và (2) đưa vào hoạt động và bảo trì hệ thống
- **Phương pháp luận phát triển hệ thống** – là một quy trình phát triển chuẩn hóa xác định một tập các hoạt động, phương pháp, thực nghiệm, kết quả và các công cụ tự động hóa mà những người phát triển hệ thống và người quản lý dự án dùng để phát triển và cải thiện không ngừng các hệ thống thông tin và phần mềm
- Các phương pháp luận phát triển hệ thống
 - Phát triển ứng dụng nhanh có kiến trúc (Architected Rapid Application Development - Architected RAD)
 - Phương pháp luận phát triển hệ thống động (Dynamic Systems Development Methodology - DSDM)
 - Phát triển ứng dụng kết hợp (Joint Application Development - JAD)
 - Công nghệ thông tin (Information Engineering - IE)
 - Phát triển ứng dụng nhanh (Rapid Application Development - RAD)
 - Quy trình hợp nhất Rational (Rational Unified Process - RUP)
 - **Phân tích và thiết kế hướng cấu trúc (Structured Analysis and Design)** – đây là phương pháp được trình bày trong bài giảng này
 - Lập trình eXtreme (eXtreme Programming - XP)

2.1.4. Các nguyên lý phát triển hệ thống

- Để người sở hữu và người sử dụng hệ thống tham gia vào tất cả các giai đoạn phát triển của hệ thống
- Sử dụng một cách tiếp cận giải quyết vấn đề
- Thiết lập các giai đoạn và các hoạt động
- Tài liệu hóa suốt quá trình phát triển
- Thiết lập các chuẩn
- Quản lý quá trình và các dự án
- Cân đối hệ thống với vốn đầu tư

- Không né tránh việc hủy bỏ hoặc sửa phạm vi
- Chia để trị
- Thiết kế hệ thống để có thể phát triển và thay đổi

Nguyên lý 1: Để người sở hữu và người sử dụng hệ thống tham gia vào tất cả các giai đoạn phát triển hệ thống

- Sự tham gia của người sử dụng sẽ tạo nên ý thức họ là người làm chủ hệ thống và dẫn đến sự chấp nhận và hài lòng của họ về hệ thống
- Có nghĩa là người sử dụng và người sở hữu hệ thống cũng “sống” trong hệ thống

Nguyên lý 2: Sử dụng một cách tiếp cận giải quyết vấn đề

- Nghiên cứu và tìm hiểu vấn đề trong ngữ cảnh của nó
- Xác định các yêu cầu của giải pháp phù hợp
- Xác định các giải pháp đề cử và chọn giải pháp tốt nhất có thể
- Thiết kế và/hoặc cài đặt giải pháp
- Quan sát và đánh giá tác động của giải pháp, và cải thiện giải pháp một cách phù hợp

Nguyên lý 3: Thiết lập các giai đoạn và các hoạt động

- Xác định phạm vi
- Phân tích vấn đề
- Phân tích yêu cầu
- Thiết kế logic
- Phân tích quyết định
- Thiết kế vật lý và tích hợp
- Xây dựng và kiểm thử
- Cài đặt và đưa vào hoạt động

Các giai đoạn trên xác định các vấn đề, đánh giá, thiết kế và cài đặt giải pháp (Quy trình phát triển hệ thống)

Nguyên lý 4: Tài liệu hóa suốt quy trình phát triển hệ thống

- Là hoạt động liên tiếp để phát hiện điểm mạnh và điểm yếu của hệ thống trong suốt quy trình phát triển
- Củng cố sự truyền đạt thông tin giữa các nhân sự trong hệ thống
- Sự tán thành và giao kèo giữa người sở hữu/người sử dụng với người phân tích/người thiết kế về phạm vi, yêu cầu và tài nguyên của dự án

Nguyên lý 5: Thiết lập các chuẩn về tính nhất quán

- Các chuẩn phát triển hệ thống: tài liệu, phương pháp luận
- Các chuẩn nghiệp vụ: các quy tắc và thực tế nghiệp vụ
- Các chuẩn công nghệ thông tin: kiến trúc và cấu hình chung cho sự phát triển hệ thống nhất quán

Nguyên lý 6: Quản lý quy trình và các dự án

- **Quản lý quy trình** : hoạt động liên tiếp trong đó tài liệu hóa, quản lý, giám sát việc sử dụng và cải thiện phương pháp luận tổ chức đã lựa chọn (“quy trình”) cho việc phát triển hệ thống. Quản lý quy trình quan tâm tới các giai đoạn, các hoạt động, các kết quả và các chuẩn chất lượng nên được áp dụng nhất quán cho mọi dự án.
- **Quản lý dự án** : quy trình xác định phạm vi, lập kế hoạch, bố trí nhân sự, tổ chức, chỉ đạo và điều khiển một dự án để phát triển một hệ thống thông tin với chi phí thấp nhất, trong một khoảng thời gian cụ thể và với chất lượng có thể chấp nhận được.

Nguyên lý 7: Cân đối hệ thống với vốn đầu tư

- **Kế hoạch hệ thống thông tin mang tính chiến lược** phải phù hợp và hỗ trợ cho **kế hoạch hoạt động mang tính chiến lược** của tổ chức
- Có một vài giải pháp có thể, cái đầu tiên không nhất thiết là cái tốt nhất
- Đánh giá tính khả thi của từng giải pháp theo hai tiêu chí:
 - **Hiệu quả chi phí**: phân tích chi phí/lợi ích
 - **Quản lý rủi ro**: xác định, đánh giá và điều khiển những thách thức tiềm ẩn đối với sự hoàn thành một hệ thống

Nguyên lý 8: Không né tránh việc hủy bỏ hoặc sửa phạm vi

- Phạm vi của một dự án có thể tăng lên
- Quy trình phát triển có các điểm kiểm tra đối với các giai đoạn của nó:
- Hủy bỏ dự án nếu nó không khả thi (do tổ chức quyết định)
- Đánh giá lại? điều chỉnh chi phí/phạm vi nếu phạm vi mở rộng thêm (do người phân tích quyết định)
- Thu hẹp phạm vi nếu ngân sách/ lịch biểu bị co lại (do người phân tích quyết định)

Nguyên lý 9: Chia để trị

- Chia một hệ thống phức tạp thành nhiều hệ thống con/thành phần đơn giản hơn
- Quy trình giải quyết vấn đề có thể được làm đơn giản hóa đối với những vấn đề nhỏ hơn
- Các hệ thống con khác nhau ứng với những loại nhân sự khác nhau

Nguyên lý 10: Thiết kế hệ thống để có thể phát triển và thay đổi

- Hệ thống cần được xây dựng sao cho mềm dẻo và dễ thích ứng để có thể thay đổi về sau

2.2. Một quy trình phát triển hệ thống

2.2.1. Động lực của một dự án phát triển hệ thống

Sự ra đời của hầu hết các dự án đều là sự kết hợp của các yếu tố thuộc 3 nhóm sau:

- **Vấn đề** (Problem) – một trạng thái khó khăn trong thực tế ngăn cản tổ chức đạt được đầy đủ mục đích, mục tiêu của nó.
- **Cơ hội** (Opportunity) – một cơ hội để cải thiện tổ chức cho dù không có vấn đề nào được xác định
- **Chỉ thị** (Directive) – một yêu cầu mới được áp đặt bởi nhà quản lý, chính phủ hoặc bộ phận có ảnh hưởng nào đó từ bên ngoài

Các dự án có kế hoạch

- Một **kế hoạch chiến lược hệ thống thông tin** xem xét toàn bộ tổ chức để xác định các dự án phát triển hệ thống, những dự án đó sẽ đem lại giá trị mang tính chiến lược dài hạn cho tổ chức.
- Việc tái cấu trúc quy trình nghiệp vụ (business process redesign) phân tích thấu đáo một chuỗi các quy trình nghiệp vụ để loại bỏ sự dư thừa, thủ tục rườm rà đồng thời cải thiện hiệu quả và giá trị gia tăng. Khi đó, cần thiết kế lại hệ thống thông tin hỗ trợ cho các quy trình nghiệp vụ đã được thiết kế lại đó.

Các dự án không có kế hoạch

- Được kích hoạt bởi một vấn đề, cơ hội hoặc chỉ thị cụ thể xuất hiện trong khi thực hiện nghiệp vụ
- Hội đồng chỉ đạo – một bộ phận quản trị gồm người sở hữu hệ thống và ban điều hành công nghệ thông tin có trách nhiệm lựa chọn dự án phát triển hệ thống phù hợp.
- Backlog – một kho lưu trữ các đề xuất dự án không thể được cấp vốn hoặc bố trí nhân sự vì chúng có mức ưu tiên thấp hơn dự án đã được phê duyệt để phát triển hệ thống.

Cả dự án được định trước hay không định trước đều phải trải qua cùng quy trình phát triển hệ thống cơ bản, chúng ta sẽ xem xét những giai đoạn dự án đó trong phần tiếp theo sau.

2.2.2. Các giai đoạn của dự án thông thường

1. Xác định phạm vi

- Mục đích: xác định các vấn đề, cơ hội và chỉ thị (problems, opportunities, và directives - POD); đánh giá rủi ro của dự án; thiết lập phạm vi, các yêu cầu và ràng buộc sơ bộ, ngân sách và lịch biểu (*nghiên cứu sơ bộ*)
- Vấn đề: Liệu dự án có đáng để xem xét – Xác định phạm vi của dự án
- Kết quả: kế hoạch/biểu đồ dự án
- Kiểm tra tính khả thi: Hủy bỏ dự án / Phê chuẩn để tiếp tục / Thu hẹp hoặc mở rộng phạm vi phù hợp với sự thay đổi ngân sách và lịch biểu

2. Phân tích vấn đề

- Mục đích: nghiên cứu và phân tích hệ thống hiện có từ góc độ của người dùng giống như cách họ nhìn nhận dữ liệu, các quy trình và giao diện
- Vấn đề: Chi phí/lợi ích của việc xây dựng hệ thống mới để giải quyết những vấn đề đó
- Kết quả: các mục tiêu cải thiện hệ thống (các tiêu chuẩn nghiệp vụ để đánh giá hệ thống mới)
- Kiểm tra tính khả thi: Hủy bỏ dự án / Phê chuẩn để tiếp tục / Thu hẹp hoặc mở rộng phạm vi phù hợp với sự thay đổi ngân sách và lịch biểu

3. Phân tích yêu cầu

- Mục đích: tìm hiểu các nhu cầu người dùng không có trong hệ thống mới về dữ liệu, các quy trình và giao diện
- Vấn đề: Xác định các yêu cầu đối với hệ thống mới (NHỮNG GÌ CẦN THỰC HIỆN) mà không cần diễn giải các chi tiết kỹ thuật (LÀM NHƯ THẾ NÀO)
- Các lỗi và sự bỏ sót trong pha phân tích yêu cầu sẽ để lại hậu quả là sự không hài lòng của người dùng về hệ thống cuối cùng và những thay đổi hao tổn chi phí
- Kết quả: báo cáo yêu cầu nghiệp vụ

4. Thiết kế logic

- Mục đích: chuyển các yêu cầu nghiệp vụ của người dùng thành mô hình hệ thống mô tả CẦN LÀM GÌ mà không xác định thiết kế kỹ thuật hoặc cài đặt cụ thể của những yêu cầu đó (*thiết kế khái niệm*)
- Vấn đề: sử dụng mô hình đồ họa của hệ thống để biểu diễn các yêu cầu của người dùng về dữ liệu, các quy trình, giao diện và để đơn giản hóa việc cải thiện sự truyền thông tin giữa các nhân sự
- Chú ý: việc mô hình hóa hệ thống quá thừa sẽ làm chậm đáng kể tiến trình hướng tới việc cài đặt giải pháp hệ thống dự định

- Kết quả: Các mô hình hệ thống logic (DFD, ERD...)

5. Phân tích quyết định

- Mục đích: xác định tất cả các giải pháp đề cử, phân tích tính khả thi của từng giải pháp, tiến cử một hệ thống làm giải pháp mục tiêu
- Vấn đề: phân tích tính khả thi dưới các tiêu chí kỹ thuật, hoạt động, tính kinh tế, lịch biểu (technical, operational, economic, schedule - TOES) và rủi ro
- Kết quả: đề xuất hệ thống được phê duyệt
- Kiểm tra tính khả thi: Hủy bỏ dự án / Chấp nhận đề xuất hệ thống với sự thay đổi ngân sách và lịch biểu / Thu hẹp phạm vi của giải pháp được đề xuất với sự thay đổi ngân sách và lịch biểu
- Các giải pháp đề cử được đánh giá dưới các tiêu chí TOES và rủi ro:
- Tính khả thi kỹ thuật – Liệu giải pháp có thực tế về kỹ thuật? Liệu nhân sự có đủ thành thạo kỹ thuật để thiết kế và xây dựng giải pháp này?
- Tính khả thi hoạt động – Liệu giải pháp có đáp ứng hết các yêu cầu của người dùng? Ở mức độ nào? Liệu giải pháp có thay đổi môi trường làm việc của người sử dụng? Người dùng sẽ cảm nhận thế nào về giải pháp đó?
- Tính khả thi kinh tế – Liệu giải pháp có hiệu quả về chi phí?
- Tính khả thi lịch biểu – Hệ thống có thể được thiết kế và cài đặt trong một khoảng thời gian chấp nhận được?
- Rủi ro – Khả năng cài đặt thành công là như thế nào? (Quản lý rủi ro)

6. Thiết kế vật lý

- Mục đích: chuyển các yêu cầu nghiệp vụ thành các đặc tả thiết kế kỹ thuật cho việc xây dựng
- Vấn đề: kỹ thuật sẽ được sử dụng như thế nào để xây dựng hệ thống về mặt dữ liệu, các quy trình và giao diện
- Kết quả: các đặc tả thiết kế hệ thống (thiết kế chi tiết)
- Kiểm tra tính khả thi: Tiếp tục/ Thu hẹp hoặc mở rộng phạm vi với sự thay đổi ngân sách và lịch biểu

7. Giai đoạn xây dựng

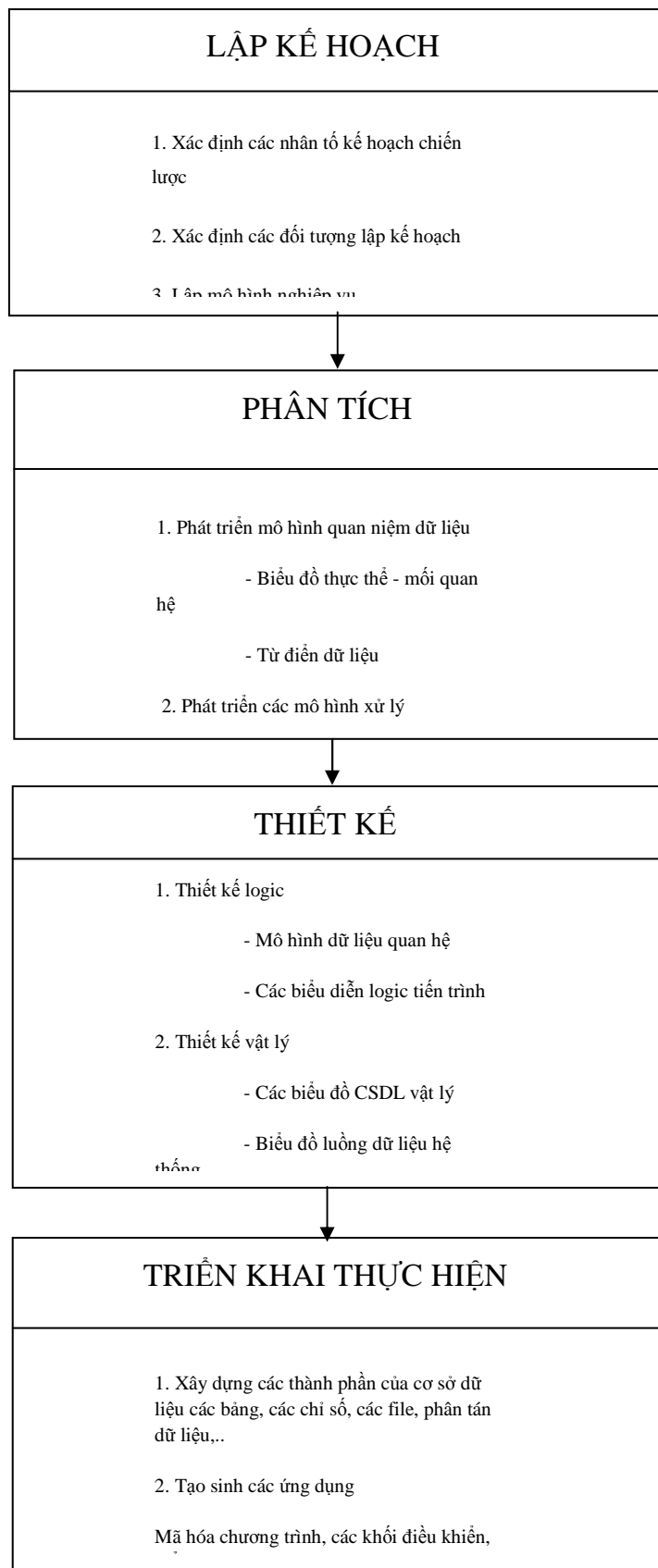
- Mục đích: xây dựng và kiểm thử hệ thống đáp ứng các yêu cầu nghiệp vụ và đặc tả thiết kế; cài đặt giao diện kết nối giữa hệ thống hiện có với hệ thống mới
- Vấn đề: xây dựng cơ sở dữ liệu, các chương trình ứng dụng giao diện người dùng/hệ thống, cài đặt phần mềm được thuê hoặc mua về
- Kết quả: hệ thống được đề xuất trong phạm vi ngân sách và lịch biểu

8. Giai đoạn cài đặt

- Mục đích: đưa hệ thống thu được vào hoạt động
- Vấn đề: huấn luyện người dùng, viết sách hướng dẫn, nạp file, tạo cơ sở dữ liệu, kiểm thử cuối cùng
- Kế hoạch chuyển đổi: từ hệ thống cũ sang hệ thống mới
- Kết quả: hệ thống sẵn sàng để hoạt động

Hoạt động và hỗ trợ

- Hỗ trợ hệ thống không ngừng tới khi hệ thống trở nên lỗi thời và bị thay thế bởi một hệ thống mới.
- Vấn đề: hỗ trợ kỹ thuật cho người dùng; sửa lỗi, kế hoạch phục hồi phù hợp với các yêu cầu nảy sinh.
- **Tóm tắt quy trình phát triển hệ thống:**
 - Giai đoạn xác định phạm vi: Vấn đề nào
 - Giai đoạn phân tích vấn đề: Các kết quả (Thông tin/Dữ liệu, Các quy trình, Các giao diện)
 - Giai đoạn phân tích yêu cầu: Những yêu cầu của người dùng
 - Thiết kế logic: Mô hình khái niệm – Cần làm gì
 - Giai đoạn phân tích quyết định: Giải pháp nào
 - Giai đoạn thiết kế: Mô hình vật lý: Làm thế nào
 - Giai đoạn xây dựng: Thực hiện
 - Giai đoạn cài đặt: Sử dụng



Hình 2.1: Lược đồ tiến trình phát triển HTTT

2.2.3. Các hoạt động xuyên suốt vòng đời

Là bất kỳ hoạt động nào diễn ra tại nhiều hoặc tất cả các giai đoạn của quy trình phát triển hệ thống.

- **Tìm hiểu thực tế (Fact-finding)**
 - Là quy trình sử dụng việc nghiên cứu, phỏng vấn, gặp gỡ, phiếu hỏi, mẫu và các kỹ thuật khác để thu thập thông tin về các vấn đề, yêu cầu của hệ thống.
 - Rất quan trọng vào những giai đoạn đầu của dự án, khi mà đội phát triển tìm hiểu về thuật ngữ chuyên ngành, các vấn đề, cơ hội, ràng buộc, các yêu cầu và mức ưu tiên.
- **Tài liệu hóa và trình bày**
 - Tài liệu hóa – là hoạt động liên tục để ghi lại thông tin và chi tiết kỹ thuật của một hệ thống cho việc tham khảo ở hiện tại và trong tương lai
 - Trình bày – là hoạt động liên tục của việc truyền đạt thông tin, tìm kiếm, đề xuất và cung cấp tài liệu để xem xét bởi người sử dụng và người quản lý
 - Kho chứa – một cơ sở dữ liệu và/hoặc tệp thư mục trong đó người phát triển hệ thống lưu tất cả các tài liệu, kiến thức và các thành phần của một hoặc nhiều dự án hoặc hệ thống thông tin
- **Phân tích tính khả thi**
- **Quản lý dự án và quy trình**

2.3. Các chiến lược phát triển hệ thống

2.3.1. Chiến lược phát triển hướng mô hình

Model-driven development – một chiến lược phát triển hệ thống nhấn mạnh vào việc vẽ các mô hình hệ thống để trợ giúp việc trực quan hóa và phân tích các vấn đề, xác định các yêu cầu nghiệp vụ, và thiết kế các hệ thống thông tin.

- **Mô hình hóa chức năng** – một kỹ thuật lấy quá trình làm trung tâm được phổ biến bởi phương pháp luận **phân tích và thiết kế hướng cấu trúc**, sử dụng các mô hình yêu cầu nghiệp vụ để tạo các thiết kế phần mềm hiệu quả cho một hệ thống.
- **Mô hình hóa dữ liệu** – một kỹ thuật lấy dữ liệu làm trung tâm để mô hình hóa các yêu cầu dữ liệu nghiệp vụ và thiết kế hệ thống cơ sở dữ liệu phù hợp.
- **Mô hình hóa đối tượng** – một kỹ thuật kết nối dữ liệu và quá trình thành các cấu trúc duy nhất gọi là các đối tượng. Các mô hình đối tượng là các biểu đồ tài liệu hóa một hệ thống dưới dạng các đối tượng của nó và các tương tác giữa chúng.

Ưu điểm:

- Kế hoạch dài hạn hơn
- Mô hình hóa hệ thống hiện tại và phân tích yêu cầu trên phạm vi rộng hơn
- Phân tích nhiều giải pháp kỹ thuật khác nhau
- Phù hợp với các hệ thống được hiểu rõ

Nhược điểm:

- Thời gian thực hiện lâu
- Sự tham gia thụ động của người sử dụng hệ thống bởi họ không nhìn thấy sản phẩm
- Các yêu cầu trong mỗi giai đoạn cần được xác định đầy đủ: điều này không thực tế và/hoặc không mềm dẻo

2.3.2. Chiến lược phát triển ứng dụng nhanh

Rapid application development (RAD) – các kỹ thuật nhấn mạnh sự tham gia của người sử dụng trong việc xây dựng tiến hóa nhanh các bản mẫu hoạt động của một hệ thống để đẩy nhanh quy trình phát triển hệ thống đó.

- RAD được dựa trên việc xây dựng các bản mẫu, những bản mẫu này tiến hóa thành các hệ thống hoàn thiện
- Một **bản mẫu** là một mô hình hoạt động hoặc mô hình biểu diễn với tỷ lệ nhỏ hơn của các yêu cầu của người sử dụng hoặc của một thiết kế đề xuất cho một hệ thống thông tin
- Một **time box** là một khoảng thời gian không thể mở rộng, thường là 60-120 ngày mà một hệ thống đề cử phải được đưa vào hoạt động. Các cải thiện sẽ được thực hiện trong những phiên bản ra đời sau đó.

Ưu điểm:

- Xử lý được các yêu cầu không ổn định hoặc không chính xác của người sử dụng
- Sự tham gia chủ động của người sử dụng vào việc xây dựng sản phẩm thực tế: làm tăng sự nhiệt tình, hỗ trợ của họ
- Phát hiện sớm các lỗi hoặc sự bỏ sót: trong quá trình kiểm thử và thay đổi bản mẫu
- Làm giảm rủi ro nhờ lặp đi lặp lại việc làm bản mẫu

Nhược điểm:

- Tăng chi phí thời gian sống để hoạt động, hỗ trợ và bảo trì hệ thống (hoạt động và sửa chữa liên tục)

- Quá trình phân tích vấn đề ngăn ngừa có thể đem lại hệ quả là việc giải quyết những vấn đề sai
- Ngăn cản người phân tích xem xét các kỹ thuật khác thay vì chỉ xét tới kỹ thuật đang được dùng để làm bản mẫu

2.3.3. Chiến lược cài đặt gói ứng dụng thương mại

Commercial application package – một ứng dụng phần mềm có thể mua về và tùy biến cho phù hợp các yêu cầu nghiệp vụ của một số lượng lớn các tổ chức hoặc một ngành nghề cụ thể. Một thuật ngữ khác là *hệ thống thương mại dùng ngay* (commercial off-the-shelf (COTS) system). Khi chọn được một gói phần mềm thích hợp, tổ chức không cần viết chương trình mà chỉ cần cài đặt nó để sử dụng.

Ưu điểm

- Cài đặt nhanh hệ thống mới (nhiều chức năng tương tự nhau giữa các tổ chức khác nhau, không cần thiết phải xây dựng từ đầu)
- Không cần các chuyên gia và nhân sự cho việc phát triển
- Chi phí phát triển thấp (nhưng tốn chi phí tùy biến và cài đặt)
- Người bán chịu trách nhiệm về việc cải thiện phần mềm và sửa lỗi

Nhược điểm

- Phụ thuộc vào người bán
- Việc tùy biến/nâng cấp trong tương lai rất tốn kém
- Một hệ thống thương mại dùng ngay hiếm khi phản ánh được hệ thống lý tưởng được tự phát triển
- Phải thay đổi các quy trình nghiệp vụ hiện tại để phù hợp với hệ thống thương mại

Ngoài các chiến lược trên, người ta còn dùng phương pháp làm bản mẫu, phương pháp thuê bao, phương pháp người sử dụng phát triển hệ thống,....

2.4. Các kỹ thuật và công cụ tự động hóa

2.4.1. Khái niệm CASE

Computer-Assisted Software Engineering - là các công cụ phần mềm tự động hóa hỗ trợ việc vẽ và phân tích các mô hình hệ thống và các đặc tả liên quan. Một số công cụ CASE cũng cung cấp khả năng làm bản mẫu và sinh mã.

- **Kho chứa CASE** – một cơ sở dữ liệu của người phát triển hệ thống trong đó họ có thể lưu các mô hình hệ thống, các đặc tả chi tiết và các sản phẩm khác của việc phát triển hệ thống. Cách gọi khác là **từ điển dữ liệu**.
- **Forward engineering** – một khả năng của công cụ CASE có thể sinh mã phần mềm và cơ sở dữ liệu ban đầu trực tiếp từ hệ thống.
- **Kỹ thuật đảo ngược - Reverse engineering** – một khả năng của công cụ CASE có thể sinh ra mô hình ban đầu của hệ thống từ mã cơ sở dữ liệu hoặc phần mềm.
- Bốn lý do để sử dụng công cụ CASE là:
 - Tăng hiệu suất phân tích
 - Làm đơn giản hóa việc giao tiếp giữa người phân tích và người sử dụng
 - Cung cấp tính liên tục giữa các giai đoạn vòng đời
 - Để đánh giá tác động của việc bảo trì

2.4.2. Phân loại CASE

Công cụ CASE có thể chia thành các loại sau:

- Các công cụ CASE mức cao (còn gọi là front-end CASE) dùng để thực hiện phân tích và thiết kế
- Các công cụ CASE mức thấp (còn gọi là back-end CASE) dùng để sinh mã từ thiết kế CASE đã có
- CASE tích hợp, thực hiện cả hai chức năng của CASE mức cao và mức thấp
- Các công cụ CASE mức cao:
 - Tạo và thay đổi thiết kế hệ thống
 - Lưu dữ liệu trong kho chứa dự án
 - Kho chứa là một tập hợp các bản ghi, phần tử, biểu đồ, hình ảnh, báo cáo và các thông tin khác của dự án
 - Các công cụ CASE đó mô hình hóa các yêu cầu tổ chức và xác định các đường biên của hệ thống
- Các công cụ CASE mức thấp sinh mã nguồn từ thiết kế CASE đã có
- Mã nguồn thường có thể được sinh dưới dạng một số ngôn ngữ lập trình

Ưu điểm của việc sinh mã

- Giảm thời gian phát triển hệ thống mới
- Thời gian để bảo trì mã được sinh ngắn hơn thời gian bảo trì hệ thống truyền thống
- Các chương trình máy tính có thể được sinh thành nhiều ngôn ngữ
- Thiết kế CASE có thể được mua từ một nhà cung cấp thứ 3 và được điều chỉnh phù hợp với các yêu cầu tổ chức
- Việc sinh mã sẽ tránh được các lỗi về mã lập trình

Kỹ thuật đảo ngược

- Là việc sinh ra thiết kế CASE từ mã chương trình máy tính
- Mã nguồn được kiểm tra, phân tích và chuyển thành các thực thể kho chứa
- Kỹ thuật đảo ngược tạo ra (tùy thuộc vào tập công cụ được sử dụng):
- Các cấu trúc dữ liệu và các phân tử, mô tả các file, bản ghi và trường
- Các thiết kế giao diện
- Trình bày báo cáo đối với các chương trình xử lý theo khối
- Một biểu đồ thể hiện sự phân cấp của các môđun trong chương trình
- Thiết kế cơ sở dữ liệu và các quan hệ
- Kỹ thuật đảo ngược có các ưu điểm sau:
 - Giảm thời gian bảo trì hệ thống
 - Tài liệu chương trình được tạo ra bù cho tài liệu đã mất
 - Các chương trình hướng cấu trúc có thể được sinh ra từ các chương trình phi cấu trúc đã có
 - Việc bảo trì hệ thống trong tương lai dễ thực hiện hơn
 - Các phần không được sử dụng của chương trình có thể được loại bỏ

2.4.3. Môi trường phát triển ứng dụng

Application development environments (ADEs) – một công cụ phát triển phần mềm tích hợp cung cấp tất cả các điều kiện cần thiết để phát triển phần mềm ứng dụng mới với chất lượng và tốc độ lớn nhất. Cách gọi khác là *môi trường phát triển tích hợp* (*integrated development environment* - IDE)

- Các thành phần ADE có thể gồm:
 - Các ngôn ngữ lập trình hoặc trình dịch
 - Các công cụ xây dựng giao diện
 - Phần mềm trung gian
 - Các công cụ kiểm thử
 - Các công cụ quản lý phiên bản
 - Các công cụ tạo Help
 - Các liên kết tới kho chứa

2.4.4. Bộ quản lý dự án và quy trình

- **Ứng dụng quản lý quy trình** – một công cụ tự động hóa trợ giúp việc lập tài liệu và quản lý một phương pháp luận và chiến lược, các kết quả của nó và các chuẩn quản lý chất lượng. Một thuật ngữ đang nổi bật là phần mềm phương pháp – *methodware* (ví dụ Visio, Visible Analyst, Rational Rose...)
- **Ứng dụng quản lý dự án** – một công cụ tự động hóa trợ giúp việc lập kế hoạch các hoạt động phát triển hệ thống (tốt nhất là sử dụng phương pháp luận đã được chấp thuận), dự đoán và phân bổ nguồn lực bao gồm con người và chi phí), lập lịch biểu hoạt động và nguồn lực, giám sát tiến trình theo lịch biểu và ngân sách,

điều khiển và sửa đổi lịch biểu và nguồn lực, và báo cáo tiến trình dự án (ví dụ Microsoft Project...)

2.5. Vai trò của những người tham gia phát triển HTTT

2.5.1 Các nhà quản lý HTTT

Các nhà quản lý HTTT có thể tham gia trực tiếp vào quá trình phát triển HTTT nếu một tổ chức nhỏ hoặc họ có mong muốn. Hơn thế nhà quản lý HTTT tham gia vào việc phân phối nguồn lực trước hay thông qua dự án phát triển HTTT. Nhà quản lý HTTT có thể tham dự các cuộc họp xem xét dự án ở những phạm vi có liên quan.

2.5.2 Các nhà phân tích hệ thống

Các nhà phân tích hệ thống là người chủ chốt trong quá trình phát triển HTTT. Để trở thành một nhà phân tích hệ thống thực thụ, họ cần ít nhất có bốn kỹ năng: kỹ năng phân tích, kỹ năng kỹ thuật, , kỹ năng quản lý và kỹ năng giao tiếp.

Kỹ năng phân tích làm cho anh ta có thể hiểu được tổ chức và sự hoạt động của nó, có thể xác định được những cơ hội và các vấn đề đang đặt ra để phân tích và giải quyết chúng. Một điểm quan trọng của kỹ năng phân tích của nhà phân tích là cách suy nghĩ mang tính hệ thống: nhờ cách suy nghĩ này, nó cho phép nhà phân tích nhìn nhận được các mối quan hệ quan trọng giữa các hệ thống thông tin trong một tổ chức cũng như với môi trường mà trong đó tổ chức tồn tại. *Kỹ năng kỹ thuật* cho phép các nhà phân tích hiểu những tiềm năng và hạn chế của công nghệ thông tin. *Kỹ năng quản lý* giúp nhà phân tích quản lý các dự án, các nguồn lực, rủi ro và những thay đổi. *Kỹ năng giao tiếp* giúp nhà quản lý làm việc với những người dùng cuối, với khách hàng cũng như với nhà phân tích khác, hay các nhà lập trình.

2.5.3 Các nhà lập trình (lập trình viên)

Các nhà lập trình có nhiệm vụ chuyển các đặc tả thiết kế hệ thống đã cho thành các cấu trúc mà máy tính hiểu được và vận hành được. Việc viết chương trình máy tính đôi khi còn gọi là viết mã hay mã hóa thiết kế. Nhà lập trình cũng viết các tài liệu chương trình và các chương trình kiểm thử.

2.5.4 Người sử dụng cuối cùng

Những người dùng cuối cùng là các nhân viên nghiệp vụ, họ là các chuyên gia trong lĩnh vực nghiệp vụ của mình. Nhà phân tích cần làm việc với họ để chuyển những hiểu biết nghiệp vụ của họ thành HTTT trợ giúp cho họ. Thông thường người dùng cuối cùng là khách hàng. Đôi khi họ cũng giúp đỡ các đội phát triển như cung cấp các ý kiến chuyên gia về cách thức hoạt động khác nhau. Họ cùng tham gia vào việc kiểm thử hệ thống, đánh giá hệ thống và là người dùng hệ thống sau này. Những đóng góp của họ có ý nghĩa quan trọng đến việc sử dụng hệ thống một cách hiệu quả.

2.5.5 Các nhà quản lý nghiệp vụ

Các nhà quản lý có quyền lực trong việc cấp vốn cho các dự án phát triển, phân phối các nguồn lực (người, vật tư, thiết bị) cần thiết để đảm bảo sự thành công của dự án. Năng lực ra quyết định, các hiểu biết về quy trình hoạt động nghiệp vụ của các nhà quản lý có thể lập nên các yêu cầu chung và các ràng buộc cho các dự án phát triển. Trong những tổ chức lớn, các dự án phát triển hệ thống được các ủy ban tư vấn, các nhà lãnh đạo điều hành tham gia quyết định, và họ thường là thành viên của các nhóm lập kế hoạch. Bởi vậy, họ có quyền định hướng cho việc phát triển, đề xuất và thông qua quyết định và giao nhiệm vụ cho các bộ phận thực hiện.

2.5.6 Các chuyên viên kỹ thuật và các nhà quản lý khác

Đối với một tổ chức lớn, còn có vai trò của người quản trị CSDL, các chuyên gia mạng và truyền thông, bộ phận đào tạo - cán bộ, cán bộ kiểm toán nội bộ.

2.6. Xây dựng thành công hệ thống thông tin

Một hệ thống thông tin như thế nào được xem là thành công? Đó là câu hỏi khó trả lời, và ngay cả đối với một hệ thống thông tin cụ thể thì mọi người không dễ dàng đồng ý với nhau về việc đánh giá và hiệu quả của nó. Tuy nhiên, người ta cũng đã đưa ra một số tiêu chuẩn làm cơ sở cho việc đánh giá một hệ thống thông tin. Một hệ thống thông tin được xem là có hiệu quả nếu nó góp phần nâng cao chất lượng hoạt động quản lý tổng thể của một tổ chức, nó thể hiện trên các mặt:

- *Đạt được các mục tiêu thiết kế* đầu ra của tổ chức
- *Chi phí vận hành là chấp nhận được*
- *Tin cậy, đáp ứng được các chuẩn mực* của một HTTT hiện hành. Chẳng hạn: tính sẵn sàng, thời gian làm việc trong ngày, tuần, thời gian thực hiện một dịch vụ, ...

- *Sản phẩm có giá trị xác đáng:* Thông tin đưa ra là đúng đắn, kịp thời, có ý nghĩa thiết thực đối với hoạt động chức và quản lý, nâng cao chất lượng sản phẩm và dịch vụ của tổ chức, sai sót cho phép.
- *Dễ học, dễ nhớ và dễ sử dụng*
- *Mềm dẻo, dễ bảo trì.*

Câu hỏi thảo luận

- 2.1. Quy trình phát triển hệ thống là gì?
- 2.2. Phân biệt Vòng đời hệ thống với Phương pháp luận phát triển hệ thống.
- 2.3. Giải thích tại sao lại nên để người sử dụng tham gia và tất cả các giai đoạn của quá trình phát triển hệ thống?
- 2.4. Các nguyên nhân có thể dẫn tới việc ra đời một dự án phát triển hệ thống?
- 2.5. Nêu các giai đoạn nói chung của một dự án phát triển hệ thống?
- 2.6. Nêu các hoạt động diễn ra trong suốt vòng đời phát triển hệ thống?
- 2.7. Nêu các ưu nhược điểm của chiến lược phát triển hệ thống hướng mô hình.

Chương 3. TỔNG QUAN VỀ PHÂN TÍCH HỆ THỐNG

3.1. Khái niệm phân tích hệ thống

- **Phân tích hệ thống:** là giai đoạn phát triển trong một dự án, *tập trung vào các vấn đề nghiệp vụ*, ví dụ như những gì hệ thống phải làm về mặt dữ liệu, các thủ tục xử lý và giao diện, *độc lập với kỹ thuật* có thể được dùng để cài đặt giải pháp cho vấn đề đó.
- **Thiết kế hệ thống:** là giai đoạn phát triển tập trung vào việc xây dựng và cài đặt mang tính kỹ thuật của hệ thống (cách thức mà công nghệ sẽ được sử dụng trong hệ thống).

3.2. Các hướng tiếp cận phân tích hệ thống

3.2.1. Các tiếp cận phân tích hướng mô hình

- Nhấn mạnh việc vẽ các mô hình hệ thống dạng đồ họa để tài liệu hóa và kiểm tra hệ thống hiện tại cũng như hệ thống được đề xuất.
- Cuối cùng thì mô hình hệ thống trở thành bản thiết kế chi tiết cho việc thiết kế và xây dựng một hệ thống được cải thiện.
- **Phân tích hướng cấu trúc** (Structured Analysis - SA): thuộc kiểu phân tích hướng mô hình, là kỹ thuật lấy quá trình làm trung tâm để phân tích một hệ thống đang có và xác định các yêu cầu nghiệp vụ cho một hệ thống mới. Phân tích hướng cấu trúc là một trong các tiếp cận chính thống đầu tiên của việc phân tích hệ thống thông tin. Hiện nay, nó vẫn là một trong các cách tiếp cận được áp dụng phổ biến nhất. Phân tích hướng cấu trúc tập trung vào luồng dữ liệu luân chuyển qua các quy trình nghiệp vụ và phần mềm. Nó được gọi là “lấy quá trình làm trung tâm”.
- Mô hình minh họa các thành phần của hệ thống: các quá trình (các chức năng, thao tác) và những thành phần liên quan là đầu vào, đầu ra và các file.
- **Kỹ thuật thông tin** (Information Engineering - IE): là kỹ thuật hướng mô hình và lấy dữ liệu làm trung tâm, nhưng có tính đến quá trình (rõ ràng ngữ cảnh) để lập kế hoạch, phân tích và thiết kế hệ thống thông tin. IE khác với SA ở chỗ, người phân tích sẽ vẽ mô hình dữ liệu trước. IE minh họa và đồng bộ hóa các quá trình và dữ liệu của hệ thống.
- **Phân tích hướng đối tượng** (Object Oriented Analysis - OOA): một kỹ thuật hướng mô hình tích hợp dữ liệu và quá trình liên quan tới việc xây dựng thành các đối tượng. Đây là kỹ thuật mới nhất trong số các hướng tiếp cận. OOA minh họa các đối tượng của hệ thống từ nhiều khung nhìn chẳng hạn như cấu trúc và hành vi.

3.2.2. Các tiếp cận phân tích hệ thống nhanh

Các cách tiếp cận phân tích hệ thống nhanh nhấn mạnh việc xây dựng các bản mẫu để xác định nhanh các yêu cầu nghiệp vụ và của người dùng đối với một hệ thống mới

- **Làm bản mẫu tìm hiểu** (Discovery prototyping) – một kỹ thuật dùng để xác định các yêu cầu nghiệp vụ của người dùng bằng cách để họ phản ứng với một bản cài đặt nhanh-thô của các yêu cầu đó

Ưu điểm

- Các bản mẫu phục vụ cho cách suy nghĩ “Ta sẽ biết cái gì mình muốn khi nhìn thấy nó”, đây là đặc điểm thường gặp của nhiều người quản lý và người dùng.

Nhược điểm

- Có thể bị chi phối bởi việc nhìn nhận và cảm giác quá vội vã
- Có thể khuyến khích sự tập trung quá sớm vào việc thiết kế
- Người dùng có thể lầm tưởng rằng đó là hệ thống hoàn thiện có thể được xây dựng một cách nhanh chóng bằng các công cụ làm bản mẫu
- **Phân tích kiến trúc nhanh** (Rapid architected analysis) – các mô hình hệ thống dẫn xuất từ hệ thống đang có hoặc từ các bản mẫu tìm hiểu
- **Sử dụng kỹ thuật đảo ngược** (Reverse engineering) – là việc sử dụng công nghệ để đọc mã nguồn của một chương trình ứng dụng, cơ sở dữ liệu và/hoặc giao diện người dùng đang có và tự động sinh ra mô hình hệ thống tương ứng

3.2.3. Các phương pháp Agile

Agile method – sự kết hợp của nhiều cách tiếp cận của việc phân tích và thiết kế các ứng dụng được cho là phù hợp với vấn đề đang được giải quyết và hệ thống đang được phát triển.

- Hầu hết các phương pháp luận mang tính thương mại đều không áp đặt một cách tiếp cận duy nhất (phân tích hướng cấu trúc, IE hay OOA) đối với người phân tích hệ thống.
- Thay vào đó, họ tích hợp tất cả các cách tiếp cận phổ biến thành một tập hợp các phương pháp agile.
- Người phát triển hệ thống có thể lựa chọn linh động từ nhiều công cụ và kỹ thuật để hoàn thành nhiệm vụ một cách tốt nhất.

3.3. Các giai đoạn phân tích hệ thống

- Giai đoạn xác định phạm vi **WHAT PROBLEM**: *Liệu có nên xem xét dự án và để làm gì?*
- Giai đoạn phân tích vấn đề **WHAT ISSUES** : *Liệu có nên xây dựng một hệ thống mới và để làm gì?*
- Giai đoạn phân tích yêu cầu **WHAT REQUIREMENTS**: *Người dùng cần gì và muốn gì từ hệ thống mới?*
- Giai đoạn thiết kế Logic **WHAT TO DO**: *Hệ thống mới cần phải làm những gì?*
- Giai đoạn phân tích quyết định **WHAT SOLUTION**: *Giải pháp nào là tốt nhất?*

3.3.1. Giai đoạn xác định phạm vi

Bước 1.1: xác định các vấn đề, cơ hội và yếu tố chi phối theo các tiêu chí sau:

- Tính khẩn cấp: trong khoảng thời gian nào thì vấn đề cần được giải quyết hoặc cơ hội hoặc yếu tố chi phối cần được nhận ra?
- Tính rõ ràng: Mức độ thấy được của của một giải pháp hoặc hệ thống mới đối với khách hàng hoặc người quản lý điều hành?
- Tính hữu ích: Một hệ thống mới hoặc giải pháp có thể tăng lợi nhuận hoặc giảm chi phí hàng năm lên/xuống bao nhiêu?
- Tính ưu tiên: dựa vào những câu trả lời trên, mức ưu tiên giữa các vấn đề, cơ hội và yếu tố chi phối là như thế nào?
- Giải pháp khả thi: vào giai đoạn đầu của dự án, giải pháp khả thi có thể diễn đạt ở dạng giản đơn sau:
 - Để nguyên
 - Sửa nhanh
 - Thay đổi đơn giản để củng cố hệ thống hiện có
 - Thiết kế lại hệ thống hiện có
 - Thiết kế một hệ thống mới

Bước 1.2: Thảo luận sơ bộ phạm vi

- Kết quả: Báo cáo phạm vi dự án (giới hạn của dự án)
- Những loại dữ liệu nào cần nghiên cứu
- Những quy trình nghiệp vụ nào cần đưa vào
- Hệ thống giao tiếp như thế nào với người dùng và các hệ thống khác
- Chú ý: nếu sau này phạm vi thay đổi thì ngân sách và lịch biểu cũng nên được thay đổi phù hợp

Bước 1.3: Đánh giá tính khả thi của dự án

- “Liệu dự án này có đáng được xem xét ?”
- Phân tích chi phí/lợi ích
- Quyết định

- Phê duyệt dự án
- Hủy bỏ dự án
- Xem xét lại phạm vi dự án (với ngân sách và lịch biểu đã được điều chỉnh)

Bước 1.4: lập biểu và lập kế hoạch ngân sách cho dự án

- Kết quả: báo cáo dự án
- Lập kế hoạch chủ đạo cho toàn bộ dự án: lập biểu và phân bổ tài nguyên
- Lập kế hoạch chi tiết và lập biểu để hoàn thiện giai đoạn kế tiếp

Bước 1.5: Trình bày dự án và kế hoạch

- Trình bày và bảo vệ dự án, kế hoạch trước hội đồng thẩm định
- Khởi đầu chính thức dự án và thông báo về dự án, các mục tiêu và lịch biểu
- Kết quả: báo cáo dự án (nhân sự, các vấn đề, phạm vi, phương pháp luận, chỉ thị về các công việc phải hoàn thành, các kết quả, các chuẩn chất lượng, lịch biểu, ngân sách)

3.3.2. Giai đoạn phân tích vấn đề

Bước 2.1: Nghiên cứu lĩnh vực vấn đề

- Tìm hiểu lĩnh vực của vấn đề và các thuật ngữ nghiệp vụ
- Dữ liệu: dữ liệu đang được lưu trữ, các thuật ngữ nghiệp vụ
- Các quá trình: các sự kiện nghiệp vụ hiện có
- Các giao diện: các vị trí và người dùng hiện tại
- Kết quả: xác định về lĩnh vực hệ thống / các mô hình của các hệ thống hiện có

Bước 2.2: Phân tích các vấn đề và cơ hội

- Nghiên cứu các nguyên nhân và hệ quả của từng vấn đề (chú ý: một hệ quả có thể lại là nguyên nhân của những vấn đề khác)
- Kết quả: các báo cáo vấn đề được cập nhật và các phân tích nguyên nhân-hệ quả của từng vấn đề và cơ hội

Bước 2.3: Phân tích các quá trình nghiệp vụ (chỉ dành cho việc tái cấu trúc quy trình nghiệp vụ)

- Đánh giá giá trị gia tăng hoặc giảm bớt của các quá trình đối với toàn bộ tổ chức
- Số lượng đầu vào, thời gian đáp ứng, các khâu đình trệ, chi phí, giá trị gia tăng, các hệ quả của việc loại bỏ hoặc hợp lý hóa quá trình
- Kết quả: các mô hình quá trình nghiệp vụ hiện tại

Bước 2.4: Xác lập các mục tiêu cải thiện hệ thống

- Xác định các mục tiêu cụ thể cải thiện hệ thống và các ràng buộc đối với mỗi vấn đề
- Các mục tiêu phải chính xác, có thể đo được
- Các ràng buộc về lịch biểu, chi phí, công nghệ và chính sách
- Kết quả: các mục tiêu cải thiện hệ thống và báo cáo đề xuất

Bước 2.5: Cập nhật kế hoạch dự án

- Cập nhật dự án:
- Thu hẹp phạm vi, chỉ giữ những mục tiêu ưu tiên cao để phù hợp với thời hạn/ngân sách
- Mở rộng phạm vi và điều chỉnh lịch biểu và ngân sách phù hợp
- Kết quả: kế hoạch dự án đã được cập nhật

Bước 2.6: trình bày các nhận xét và đề xuất

- Kết quả: các mục tiêu cải thiện hệ thống
- Quyết định: tiếp tục/điều chỉnh/hủy bỏ dự án hiện tại

3.3.3. Giai đoạn phân tích yêu cầu

Bước 3.1: xác định các yêu cầu hệ thống

- Các yêu cầu chức năng: các hoạt động và dịch vụ cung cấp bởi hệ thống: các chức năng nghiệp vụ, các đầu vào, đầu ra, dữ liệu được lưu trữ.
- Các yêu cầu phi chức năng: các đặc trưng, đặc điểm xác định một hệ thống thỏa đáng: hiệu suất, tài liệu, ngân sách, tính dễ học và sử dụng, tiết kiệm chi phí, tiết kiệm thời gian, an toàn.
- Kết quả: phác thảo các yêu cầu chức năng và phi chức năng: các mục tiêu cải thiện và đầu vào, đầu ra, các quá trình, dữ liệu được lưu trữ liên quan để đạt được mục tiêu

Bước 3.2: Phân mức ưu tiên cho các yêu cầu

- Các yêu cầu mang tính bắt buộc có ưu tiên cao hơn các yêu cầu khác
- Time boxing: đưa ra hệ thống dưới dạng một tập các phiên bản kế tiếp nhau trong một khoảng thời gian. Phiên bản đầu tiên đáp ứng các yêu cầu thiết yếu và có mức ưu tiên cao nhất.

Bước 3.3: Cập nhật kế hoạch dự án

- Nếu các yêu cầu vượt quá phiên bản đầu tiên: thu hẹp phạm vi hoặc tăng ngân sách
- Kết quả: các yêu cầu hệ thống đã được thống nhất (các yêu cầu và mức ưu tiên đã được bổ sung)

3.3.4. Giai đoạn mô hình hóa logic

Bước 4.1: Phân tích các yêu cầu mang tính chức năng

- Các mô hình hệ thống logic: hệ thống phải làm gì (chứ không phải làm như thế nào)
- Việc tách biệt phần nghiệp vụ với các giải pháp kỹ thuật sẽ giúp cho việc xem xét các cách thức khác nhau để cải thiện các quá trình nghiệp vụ và các khả năng lựa chọn giải pháp kỹ thuật.
- Xây dựng các bản mẫu để xác lập các yêu cầu giao diện người dùng
- Kết quả: các mô hình dữ liệu (ERD), các mô hình quá trình (DFD), các mô hình giao diện (biểu đồ ngữ cảnh, biểu đồ Use case), các mô hình đối tượng (các biểu đồ UML) của hệ thống được đề xuất.

Bước 4.2: Kiểm tra các yêu cầu mang tính chức năng

- Kiểm tra tính đầy đủ, xem xét lại, thực hiện các thay đổi và bổ sung đối với các mô hình hệ thống và các bản mẫu để đảm bảo rằng các yêu cầu đã được xác định thỏa đáng.
- Liên kết các yêu cầu phi chức năng với các yêu cầu mang tính chức năng.

3.3.5. Giai đoạn phân tích quyết định

Là giai đoạn chuyển tiếp giữa phân tích hệ thống và thiết kế hệ thống

Bước 5.1: xác định các giải pháp đề cử

- Xác định tất cả các giải pháp đề cử có thể có
- Kết quả: ma trận các hệ thống (giải pháp) đề cử

Bước 5.2: Phân tích các giải pháp đề cử

- Việc phân tích tính khả thi được thực hiện với từng đề cử mà không quan tâm tới tính khả thi của các đề cử khác. Các tính khả thi về kỹ thuật, tính sẵn sàng hoạt động, tính kinh tế, lịch biểu:
 - **Tính khả thi về kỹ thuật.** Liệu giải pháp có phù hợp với thực tế công nghệ? Liệu đội ngũ dự án có chuyên gia kỹ thuật để thiết kế và xây dựng giải pháp?

- **Tính khả thi về hoạt động.** Liệu giải pháp có thực hiện được yêu cầu của người dùng? Ở mức độ nào? Giải pháp sẽ thay đổi môi trường làm việc của người dùng như thế nào? Người dùng sẽ cảm thấy như thế nào về giải pháp như vậy?
- **Tính khả thi về kinh tế.** Liệu giải pháp có chi phí hiệu quả?
- **Tính khả thi lịch biểu.** Liệu giải pháp có thể được thiết kế và xây dựng trong một khoảng thời gian chấp nhận được hay không?

Bước 5.3: So sánh các giải pháp đề cử

- Chọn giải pháp đề cử có sự kết hợp “toàn diện tốt nhất” của các tính khả thi về kỹ thuật, hoạt động, kinh tế và lịch biểu
- Ma trận tính khả thi
- Kết quả: giải pháp được đề xuất

Bước 5.4: Cập nhật kế hoạch dự án

- Đầu vào: giải pháp đề xuất
- Xem xét và cập nhật lịch biểu mới nhất của dự án và phân bổ tài nguyên
- Kết quả: cập nhật kế hoạch dự án

Bước 5.5: đề xuất một giải pháp

- Kết quả: đề xuất dự án

3.4. Xác định các yêu cầu của người dùng

3.4.1. Giới thiệu

- **Vai trò của việc xác định yêu cầu:**
 - **Yêu cầu hệ thống** (yêu cầu nghiệp vụ) là một mô tả các nhu cầu và mong muốn đối với một hệ thống thông tin. Một yêu cầu có thể mô tả các chức năng, đặc trưng (thuộc tính) và các ràng buộc.
 - **Các yêu cầu mang tính chức năng:** các chức năng hoặc đặc trưng có thể có trong một hệ thống thông tin để nó thỏa mãn nhu cầu nghiệp vụ và có thể chấp nhận được đối với người dùng
 - **Các yêu cầu phi chức năng:** các đặc trưng, đặc điểm và thuộc tính của các hệ thống cũng như bất kỳ các ràng buộc nào có thể giới hạn ranh giới của giải pháp được đề xuất.
- **Hậu quả của yêu cầu không chính xác:**

- Hệ thống có thể tốn nhiều chi phí hơn
- Hệ thống có thể hoàn thiện muộn hơn thời gian đã định
- Hệ thống có thể không phù hợp với những gì người dùng mong muốn và sự hài lòng đó có thể khiến họ không sử dụng nó
- Chi phí bảo trì và củng cố hệ thống có thể quá cao
- Hệ thống có thể không chắc chắn và dễ có lỗi và ngừng hoạt động
- Uy tín của các chuyên gia trong đội dự án có thể bị giảm sút bởi bất kỳ thất bại nào, cho dù là do ai gây ra thì cũng sẽ bị xem là lỗi của cả đội dự án
- **Các tiêu chuẩn xác định yêu cầu hệ thống:**
 - **Nhất quán** – các yêu cầu không mâu thuẫn hay nhập nhằng lẫn nhau.
 - **Toàn diện** – các yêu cầu mô tả mọi đầu vào và đáp ứng có thể có của hệ thống.
 - **Khả thi** – các yêu cầu có thể được thoả mãn dựa trên các tài nguyên và ràng buộc sẵn có.
 - **Cần thiết** – các yêu cầu là thực sự cần thiết và đáp ứng mục đích của hệ thống.
 - **Chính xác** – các yêu cầu được phát biểu chính xác.
 - **Dễ theo dõi** – các yêu cầu ánh xạ trực tiếp tới các chức năng và đặc trưng của hệ thống.
 - **Có thể kiểm tra** – **các yêu cầu đã được vạch rõ nên** việc kiểm tra có thể thực hiện được.

3.4.2. Quy trình xác định yêu cầu

- Phân tích yêu cầu: Phân tích các yêu cầu để giải quyết các vấn đề về:
 - Các yêu cầu bị thiếu
 - Các yêu cầu mâu thuẫn nhau
 - Các yêu cầu không khả thi
 - Các yêu cầu trùng lặp
 - Các yêu cầu mơ hồ
- Chính thức hóa các yêu cầu:
 - Lập tài liệu xác định các yêu cầu
 - Truyền đạt tới các nhân sự tham gia
 - Lập tài liệu yêu cầu - một tài liệu xác định yêu cầu bao gồm:
 - Các chức năng, dịch vụ mà hệ thống nên cung cấp.
 - Các yêu cầu phi chức năng bao gồm các thuộc tính, đặc điểm và đặc trưng của hệ thống.
 - Các ràng buộc giới hạn sự phát triển của hệ thống hoặc theo đó hệ thống phải hoạt động
 - Thông tin về các hệ thống khác mà hệ thống phải giao tiếp
- Quản lý yêu cầu: là quá trình quản lý các thay đổi đối với các yêu cầu
- Trong thời gian diễn ra dự án, việc xuất hiện các yêu cầu mới hay thay đổi những yêu cầu đã có là rất phổ biến

- Các nghiên cứu cho thấy rằng có đến 50% hoặc hơn thế các yêu cầu sẽ biến đổi trước khi hệ thống được hoàn thiện



Hình 3-1 Ngũ cảnh yêu cầu đối với một hệ thống thông tin

3.4.3. Các phương pháp tìm hiểu thực tế

- Lấy mẫu của các cơ sở dữ liệu, biểu mẫu và tài liệu hiện có
- Nghiên cứu và thăm địa điểm của tổ chức.
- Quan sát môi trường làm việc
- Lập phiếu hỏi
- Phỏng vấn
- Làm bản mẫu thăm dò: (Discovery prototyping) – là hoạt động xây dựng một mô hình làm việc hoặc mô hình minh họa quy mô nhỏ đối với các yêu cầu của người dùng để phát hiện hoặc kiểm tra các yêu cầu đó
- **Lập kế hoạch yêu cầu kết hợp** (Joint requirements planning - JRP) – một quá trình trong đó các cuộc họp nhóm làm việc được tổ chức chặt chẽ (có chương trình rõ ràng và những đại diện quan trọng) nhằm mục đích phân tích các vấn đề và xác định các yêu cầu.
- JRP là một tập con của kỹ thuật phát triển ứng dụng kết hợp (Joint Application Development – JAD) bao gồm toàn bộ quá trình phát triển hệ thống.
- Ích lợi của JRP:
- JRP tập hợp những người sử dụng và người quản lý vào một dự án phát triển (khuyến khích họ trở thành “người làm chủ” dự án).
- JRP giảm lượng thời gian cần thiết để phát triển hệ thống.
- JRP kết hợp chặt chẽ những ích lợi của việc làm bản mẫu để làm phương tiện xác nhận các yêu cầu và đạt được sự phê chuẩn cho thiết kế.

Ví dụ tóm tắt kết quả khảo sát để xây dựng hệ thống quản lý bán điện:

Công ty điện X cần quản lý hệ thống bán điện cho người tiêu dùng. Khi người tiêu dùng có nhu cầu sử dụng điện năng cần phải ký hợp đồng với công ty. Hợp đồng được

phân thành nhiều loại như hộ gia đình, hộ kinh doanh, hộ sản xuất ...; ứng với mỗi loại có một đơn giá riêng. Đơn giá này còn phụ thuộc vào số lượng điện tiêu thụ, ví dụ: đơn giá tính theo 50 số điện đầu tiên, 100 số tiếp theo ...

Hàng tháng công ty có nhân viên đi ghi số điện tại công tơ của mỗi hợp đồng mua điện. Sau đó, các thông tin này được ghi vào cơ sở dữ liệu và hệ thống tự động tạo ra các hoá đơn thanh toán của từng hợp đồng mua điện. Hoá đơn này được gửi đến từng hộ mua điện. Khi hoá đơn được thanh toán, hệ thống sẽ xác nhận và cập nhật vào cơ sở dữ liệu. Đầu tháng, hệ thống tự động kiểm tra cơ sở dữ liệu để tìm ra những hoá đơn nào chưa thanh toán. Nếu có hoá đơn nào sau 3 tháng còn chưa được thanh toán, hệ thống tự động tạo ra một phiếu nhắc thanh toán quá hạn. Phiếu này sẽ được gửi đến hộ mua điện có hoá đơn chưa thanh toán. Nếu sau 6 tháng, hoá đơn chưa được thanh toán, hệ thống sẽ tạo ra một phiếu ngừng cung cấp điện. Công ty chỉ tiếp tục bán điện khi các hoá đơn này được thanh toán hết.

Khách hàng có thể yêu cầu tạm ngừng cung cấp điện hoặc huỷ bỏ hợp đồng mua điện với công ty. Khách hàng chỉ được huỷ bỏ hợp đồng khi đã thanh toán hết tất cả các hoá đơn.

Câu hỏi thảo luận C3

3.1. Phân biệt Phân tích hệ thống và Thiết kế hệ thống. Người ta nói: Khi phân tích hệ thống ta PHẢI quan tâm đến các vấn đề kỹ thuật liên quan, Bạn nghĩ gì về nhận xét này.

3.2. Phân biệt kỹ thuật phân tích hướng cấu trúc với phân tích hướng đối tượng.

GỢI Ý: tham khảo mục Các hướng tiếp cận phân tích hệ thống. 3.2.1

3.3. Kể tên các giai đoạn phân tích hệ thống.

3.4. Nêu vai trò của việc xác định yêu cầu.

3.5. Kể tên các phương pháp khảo sát thực tế.

Chương 4. CÁC PHƯƠNG PHÁP THU THẬP THÔNG TIN

4.1. Phương pháp phỏng vấn

- Phỏng vấn là một phương pháp quan trọng để thu thập dữ liệu về các yêu cầu của hệ thống thông tin
- Việc phỏng vấn nhằm phát hiện thông tin về:
 - Các ý kiến của người được phỏng vấn
 - Cảm giác của người được phỏng vấn
 - Trạng thái hiện tại của hệ thống
 - Các mục tiêu của con người và tổ chức
 - Các thủ tục nghiệp vụ không chính thức
- Năm bước lập kế hoạch phỏng vấn là:
 - Đọc các tài liệu cơ bản
 - Thiết lập các mục tiêu phỏng vấn
 - Xác định người đi phỏng vấn
 - Chuẩn bị người được phỏng vấn
 - Quyết định cấu trúc và kiểu câu hỏi
- Có hai kiểu câu hỏi phỏng vấn cơ bản:
 - Câu hỏi mở
 - Câu hỏi đóng



Hình 4-1 So sánh câu hỏi mở và câu hỏi đóng trong phỏng vấn

4.1.1. Dạng câu hỏi mở

- Các câu hỏi phỏng vấn mở cho phép những người được phỏng vấn trả lời những gì họ mong muốn và mức độ mong muốn của họ
- Các câu hỏi mở phù hợp khi người phân tích quan tâm tới độ rộng và sâu của câu trả lời
- Có tám ưu điểm:
 - Làm cho người được phỏng vấn cảm thấy thoải mái
 - Cho phép người phỏng vấn tập trung vào cách biểu đạt của người được phỏng vấn:
 - Phản ánh trình độ văn hóa, các giá trị, thái độ và niềm tin
 - Cung cấp mức độ chi tiết cao
 - Phát hiện các câu hỏi mới mà chưa được khai thác
 - Làm cho người được phỏng vấn thấy thú vị hơn
 - Cho phép tính tự nhiên cao hơn
 - Giúp người phỏng vấn dễ điều chỉnh nhịp độ hơn
 - Hữu ích khi người phỏng vấn không chuẩn bị trước
- Có năm nhược điểm:
 - Có thể thu được quá nhiều chi tiết không liên quan
 - Có thể mất đi tính điều khiển cuộc phỏng vấn
 - Có thể mất quá nhiều thời gian để thu được thông tin có ích
 - Có khả năng thể hiện rằng người phỏng vấn không chuẩn bị
 - Có thể gây ấn tượng rằng người phỏng vấn đang trong “cuộc hành trình đi câu”

4.1.2. Dạng câu hỏi đóng

- Câu hỏi đóng hạn chế số câu trả lời có thể có
- Câu hỏi đóng phù hợp để tạo ra dữ liệu đáng tin cậy và chính xác, dễ dàng để phân tích
- Phương pháp luận hiệu quả và đòi hỏi ít kỹ năng đối với người phỏng vấn
- Sáu ưu điểm:
 - Tiết kiệm thời gian phỏng vấn
 - Dễ dàng so sánh giữa các lần phỏng vấn
 - Dễ đạt đúng mục đích
 - Kiểm soát được cuộc phỏng vấn
 - Bao phủ một phạm vi rộng lớn một cách nhanh chóng
 - Thu hoạch được các dữ liệu liên quan
- Bốn nhược điểm:
 - Nhàm chán đối với người được phỏng vấn
 - Khó thu được nhiều chi tiết
 - Có thể mất đi các ý tưởng chính
 - Khó tạo được mối giao tiếp tốt giữa người phỏng vấn và người được phỏng vấn

4.1.3. Các dạng câu hỏi khác

- Các câu hỏi lưỡng cực:
 - Là những câu hỏi có thể trả lời với các từ “có” hoặc “không” hoặc “đồng ý” hoặc “không đồng ý”
 - Các câu hỏi này chỉ nên dùng khi thật cần thiết
- Các câu hỏi thăm dò:
 - Các câu hỏi thăm dò gọi ra tính chi tiết hơn về câu hỏi trước đó
 - Mục đích của câu hỏi thăm dò là:
 - Thu được nhiều ý nghĩa hơn
 - Làm sáng rõ
 - Khai thác và mở rộng các quan điểm của người được phỏng vấn

4.1.4. Thứ tự đặt câu hỏi

- Ba cách cơ bản để cấu trúc cuộc phỏng vấn là:
 - Kim tự tháp: mở đầu với các câu hỏi đóng và tiếp tục với các câu hỏi mở
 - Hình phễu: mở đầu với các câu hỏi mở và tiếp tục với các câu hỏi đóng
 - Kim cương: mở đầu với các câu hỏi đóng, tiếp tục với các câu hỏi mở và kết thúc bằng các câu hỏi đóng
- Cấu trúc kim tự tháp
 - Mở rất chi tiết, thường là bằng các câu hỏi đóng
 - Mở rộng bằng các câu hỏi mở và những câu trả lời tổng quát hơn
 - Hữu ích nếu người được phỏng vấn cần được khích lệ đi vào chủ đề hoặc tỏ ra không tự nguyện hướng tới chủ đề
- Cấu trúc phễu:
 - Mở đầu với các câu hỏi mở, mang tính tổng quát
 - Kết thúc bằng cách thu hẹp các câu trả lời có thể có bằng việc sử dụng các câu hỏi đóng
 - Cung cấp cách thức dễ dàng, không gây áp lực để bắt đầu một cuộc phỏng vấn
 - Có ích khi người được phỏng vấn cảm thấy hứng khởi với chủ đề
- Cấu trúc kim cương
 - Một cấu trúc hình kim cương mở đầu theo cách rất cụ thể
 - Tiếp theo các vấn đề tổng quát hơn được xem xét
 - Kết thúc với các câu hỏi cụ thể
 - Cấu trúc này kết hợp thế mạnh của cả cấu trúc kim tự tháp và hình phễu
 - Mất nhiều thời gian hơn các cấu trúc khác
- Kết thúc việc phỏng vấn:
 - Luôn luôn hỏi “Liệu còn có gì khác mà bạn muốn bổ sung không?”
 - Tóm tắt và cung cấp phản hồi về ấn tượng của người phỏng vấn
 - Hỏi xem người tiếp theo nên phỏng vấn là ai
 - Thiết lập các cuộc hẹn gặp tiếp theo

- Cảm ơn người được phỏng vấn và bắt tay
- Báo cáo phỏng vấn
 - Viết càng sớm càng tốt ngay sau khi phỏng vấn
 - Cung cấp một bản tóm tắt ban đầu, sau đó thì chi tiết hơn
 - Xem lại báo cáo với người được phỏng vấn

4.2. Phương pháp dùng phiếu hỏi

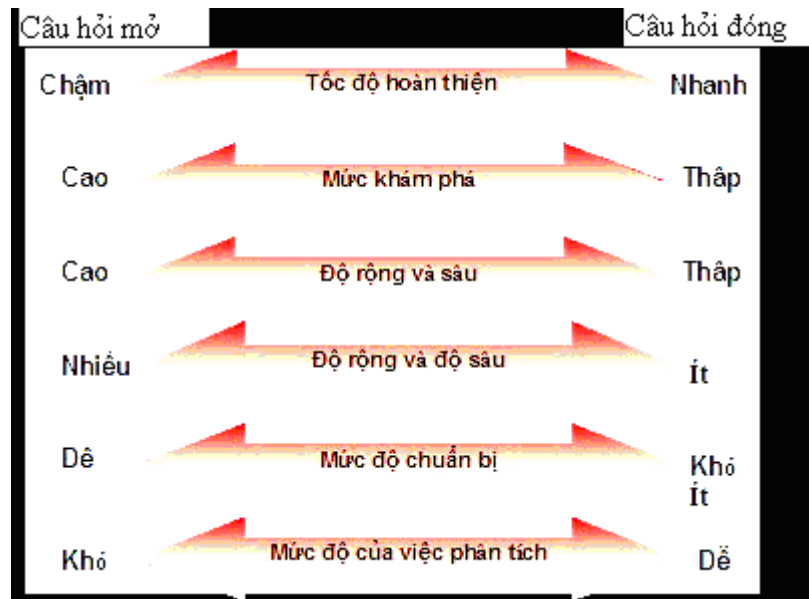
- Phiếu hỏi có ích để thu thập thông tin từ các thành viên chủ đạo trong tổ chức về:
 - Thái độ
 - Niềm tin
 - Hành vi
 - Tính cách
- Phiếu hỏi có giá trị nếu:
 - Các thành viên của tổ chức phân tán rộng
 - Nhiều thành viên tham gia vào dự án
 - Cần việc có tính thăm dò
- Các câu hỏi được thiết kế theo một trong hai kiểu

- Câu hỏi mở

- Cố gắng đoán trước câu trả lời sẽ nhận được
- Phù hợp để thu được các ý kiến

- Câu hỏi đóng

- Sử dụng khi tất cả các lựa chọn đều liệt kê được
- Khi các lựa chọn loại trừ lẫn nhau



Hình 4-2 So sánh câu hỏi mở và câu hỏi đóng khi dùng phiếu hỏi

4.2.1. Thiết kế phiếu hỏi

- Ngôn ngữ dùng trong phiếu hỏi nên:
 - Đơn giản
 - Cụ thể
 - Không thành kiến
 - Không có vẻ bề trên
 - Chính xác về mặt kỹ thuật
 - Hướng đến những người có hiểu biết
 - Phù hợp với khả năng đọc hiểu của người trả lời
- Phiếu hỏi phải chính xác và đáng tin cậy
 - Tính tin cậy thể hiện sự nhất quán trong trả lời – nghĩa là thu được cùng các kết quả nếu như cùng một phiếu hỏi được phân phát trong cùng điều kiện
 - Tính chính xác là mức độ câu hỏi đo được những gì người phân tích muốn đánh giá
- Tỷ lệ câu trả lời tốt có thể có được nhờ sự điều chỉnh phù hợp phiếu hỏi
 - Để ra nhiều khoảng trống
 - Bố trí khoảng trống lớn để viết/gõ câu trả lời
 - Tạo điều kiện cho người trả lời dễ dàng bày tỏ rõ câu trả lời của họ
 - Nhất quán về hình thức
- Thứ tự câu hỏi:
 - Đặt các câu hỏi quan trọng nhất lên đầu tiên
 - Nhóm các câu hỏi có cùng nội dung lại với nhau
 - Đặt các câu hỏi ít gây tranh luận lên trên

4.2.2. Các phương pháp phát phiếu hỏi

- Tập hợp tất cả những người trả lời vào cùng một thời gian
- Phát phiếu hỏi cho từng cá nhân
- Gửi phiếu hỏi qua đường bưu điện
- Phát phiếu hỏi qua Web hoặc thư điện tử, có các ưu điểm:
 - Giảm chi phí
 - Thu thập và lưu trữ các kết quả dễ dàng hơn
- Phiếu hỏi dạng web thường gồm:
 - Hộp văn bản đơn dòng
 - Hộp văn bản cuộn, dùng một hoặc nhiều đoạn văn bản
 - Hộp chọn dành cho các câu trả lời có/không hoặc đúng/sai
 - Nút tùy chọn cho các câu trả lời mang tính loại trừ lẫn nhau có/không hoặc đúng/sai
 - Menu thả để chọn từ một danh sách
 - Nút Submit (xác nhận) hoặc Reset (xác lập lại)

4.3. Phương pháp lấy mẫu

Lấy mẫu là quá trình lựa chọn một cách có hệ thống các phần tử đại diện của một mẫu. Thay vì nghiên cứu tất cả các thể hiện của các biểu mẫu và bản ghi trong các tệp hoặc cơ sở dữ liệu thì người phân tích chỉ cần sử dụng kỹ thuật lấy mẫu để chọn ra một phần đủ lớn các phần tử đại diện phục vụ cho việc xác định thông tin diễn ra trong hệ thống.

Bao gồm hai quyết định quan trọng:

- Những tài liệu và website quan trọng nào nên được lấy mẫu
- Những người nào nên được phỏng vấn và gửi phiếu hỏi

Lý do người phân tích cần lấy mẫu là:

- Giảm chi phí
- Tăng tốc quá trình thu thập dữ liệu
- Cải thiện hiệu quả
- Giảm việc tập trung thu thập dữ liệu

4.3.1. Các bước thiết kế mẫu

- Để thiết kế một mẫu tốt, một người phân tích hệ thống cần tuân theo bốn bước sau:
 - Xác định dữ liệu cần được thu thập hoặc mô tả

- Xác định tập cần được lấy mẫu
- Chọn loại mẫu
- Quyết định kích thước mẫu
- Quyết định kích thước mẫu nên được thực hiện theo những điều kiện cụ thể mà người phân tích hệ thống làm việc:
 - Lấy mẫu dữ liệu trên các thuộc tính
 - Lấy mẫu dữ liệu trên các biến
 - Lấy mẫu dữ liệu định tính

4.3.2. Các kiểu lấy mẫu

- Lấy mẫu tùy ý:
 - Các mẫu không giới hạn, không mang tính xác suất
 - Dễ sắp xếp
 - Không đáng tin cậy nhất
- Lấy mẫu có mục đích
 - Dựa trên sự đánh giá
 - Người phân tích chọn nhóm các cá nhân để lấy mẫu
 - Dựa trên các tiêu chuẩn
 - Mẫu không mang tính xác suất
 - Đáng tin cậy ở mức độ vừa phải
- Lấy mẫu ngẫu nhiên đơn giản
 - Dựa trên danh sách các con số của tập lấy mẫu
 - Mỗi người hoặc tài liệu đều có cơ hội được lựa chọn ngang nhau
- Lấy mẫu ngẫu nhiên phức tạp, có ba hình thức là:
 - Lấy mẫu có hệ thống
 - Là phương pháp đơn giản nhất của lấy mẫu theo xác suất
 - Chọn mọi cá nhân thứ k trong danh sách
 - Không hay nếu danh sách được sắp thứ tự
 - Lấy mẫu phân tầng
 - Xác định các tập lấy mẫu con
 - Chọn các đối tượng hoặc con người để lấy mẫu từ tập lấy mẫu con
 - Bù vào số lượng không cân đối các nhân viên trong một nhóm nhất định
 - Chọn các phương pháp khác nhau để thu thập dữ liệu từ các nhóm con khác nhau
 - Là phương pháp quan trọng nhất đối với người phân tích
 - Lấy mẫu theo nhóm
 - Chọn nhóm các tài liệu hoặc con người để nghiên cứu
 - Chọn các nhóm điển hình đại diện cho số còn lại

4.4. Phân tích tài liệu định lượng/định tính

4.4.1. Phân tích tài liệu định lượng

- Nghiên cứu dữ liệu cứng là một phương pháp hữu hiệu để người phân tích thu thập thông tin
- Dữ liệu cứng có thể thu thập từ:
 - Phân tích các tài liệu định lượng như các hồ sơ được sử dụng để ra quyết định
 - Các báo cáo thực thi
 - Các hồ sơ
 - Các mẫu thu thập dữ liệu
 - Các giao dịch nghiệp vụ

4.4.2. Phân tích tài liệu định tính

- Xem xét các tài liệu định tính để thu được:
 - Các thông tin tiềm ẩn quan trọng
 - Trạng thái tâm lý
 - Những gì được coi là tốt/xấu
 - Hình ảnh, logo, biểu tượng
- Tài liệu định tính bao gồm:
 - Các bản ghi nhớ
 - Dấu hiệu trên các bản tin
 - Website của tổ chức
 - Các tài liệu chỉ dẫn
 - Sổ tay về chính sách của tổ chức

4.5. Phương pháp quan sát

- Việc quan sát cung cấp sự hiểu biết về những gì các thành viên của tổ chức thực sự đang làm
- Nhìn nhận trực tiếp các quan hệ tồn tại giữa những người ra quyết định và các thành viên khác của tổ chức
- Kỹ thuật STROBE:
 - Gọi là kỹ thuật quan sát môi trường có cấu trúc (**STR**uctured **OB**serva**tion** of the **E**nvironment). Là kỹ thuật quan sát môi trường của những người ra quyết định
 - STROBE phân tích bảy phần tử môi trường:
 - Vị trí văn phòng
 - Vị trí bàn làm việc
 - Thiết bị văn phòng
 - Tài sản
 - Các nguồn thông tin bên ngoài

- Màu sắc và ánh sáng văn phòng
- Trang phục của người ra quyết định
- Vị trí văn phòng
 - Những văn phòng dễ thâm nhập
 - Các hành lang chính, cửa mở thông nhau
 - Không gian đi lại lớn
 - Làm tăng tần suất tương tác và các thông điệp không chính thức
 - Những văn phòng khó thâm nhập
 - Có thể nhìn nhận hệ thống theo cách khác
 - Nằm cô lập so với các văn phòng khác
 - Vị trí bàn làm việc
 - Không gian kín, quay lưng vào tường, khoảng rộng sau bàn lớn
 - Thể hiện vị trí có sức mạnh lớn nhất
- Bàn quay mặt vào tường, ghế nằm về một phía
 - Khích lệ nhân viên
 - Khả năng trao đổi, giao tiếp ngang nhau
 - Thiết bị văn phòng
- Tủ hồ sơ và giá sách:
 - Nếu không có những thứ đó thì nhân viên chỉ lưu trữ một số mục thông tin mang tính cá nhân
- Tài sản
 - Máy tính điện tử
 - Máy vi tính
 - Bút mực, bút chì, thước kẻ
- Các nguồn thông tin bên ngoài
 - Báo hoặc tạp chí thương mại thể hiện rằng nhân viên khai thác các thông tin bên ngoài
 - Các báo cáo, sổ ghi nhớ, sổ tay chính sách của công ty thể hiện rằng con người khai thác các thông tin bên trong tổ chức
- Màu sắc và ánh sáng văn phòng
- Ánh sáng chói, âm áp thể hiện:
 - Khuynh hướng hướng tới giao tiếp cá nhân nhiều hơn
 - Nhiều cuộc giao tiếp không chính thức hơn
- Màu tươi, sáng sủa thể hiện:
 - Nhiều sự giao tiếp chính thức hơn (vì vậy nên chú trọng vào sổ ghi nhớ, các báo cáo...)
 - Trang phục
- Nam giới
 - Complê trang trọng thể hiện khả năng đó là người có quyền lực lớn
 - Trang phục bình thường thể hiện nhiều khả năng đó là người tham gia vào việc ra quyết định

- Nữ giới
 - Trang phục trang trọng thể hiện khả năng đó là người có quyền lực
 - Có 5 biểu tượng dùng để đánh giá kết quả quan sát các phần tử của STROBE so với kết quả phỏng vấn thực tế là:
- Một dấu check – kết quả phỏng vấn được xác nhận
- Dấu “X” – kết quả phỏng vấn là ngược lại
- Biểu tượng oval – cần phải xem xét kỹ hơn
- Hình vuông – việc quan sát làm thay đổi kết quả phỏng vấn
- Hình tròn – kết quả phỏng vấn được bổ sung bởi việc quan sát

4.6 Các công việc sau khảo sát

Trừ các trường hợp sử dụng những bộ công cụ tiên tiến để thu thập yêu cầu thông tin, đối với đa số các trường hợp còn lại, các dữ liệu thu được thường vẫn là những dữ liệu thô, là các chi tiết tản mạn cần được xử lý sơ bộ và tổng hợp. Xử lý sơ bộ, phân loại, tổng hợp các dữ liệu thu được là rất cần thiết để tiện theo dõi, quản lý, phục vụ trực tiếp cho quá trình khảo sát và làm tư liệu cho các bước tiếp theo.

4.6.1 Xử lý sơ bộ kết quả khảo sát

Sau khi phỏng vấn, điều tra, nghiên cứu tài liệu ta cần xem lại và hoàn thiện tài liệu thu được, bao gồm việc phân loại, sắp xếp, trích rút dữ liệu, tổng hợp,... dư liệu, làm cho nó trở nên đầy đủ, chính xác, cân đối, gọn gàng dễ kiểm tra và dễ theo dõi. Phát hiện những chỗ thiếu để bổ sung, những chỗ sai hay không logic để sửa đổi. Hoàn chỉnh biểu đồ chức năng phân cấp thu được. Quá trình này thường được lặp lại nhiều lần và tiến hành song song với các hoạt động xác định yêu cầu.

Dự án	Tiêu dự án: Lập đơn hàng	trang số 6
Loại: Phân tích hiện trạng	Mô tả dữ liệu	số tt: 10 Ngày 05/08/2008
Định nghĩa:	Tên dữ liệu: Nhà cung cấp Dùng để chỉ những người cung cấp hàng thường xuyên, nó cho phép xác định mỗi nhà cung cấp	
Cấu trúc và khuôn dạng	Kiểu ký tự, gồm từ 30-40 ký tự, một số chữ đầu hay chữ tắt viết hoa	
Loại hình	Sơ cấp (dữ liệu gốc)	
Số lượng	50 nhà cung cấp (mức tối đa)	
Ví dụ:	Công ty HANOISIMEX, viết tắt: HANOISIMEX	
Lời bình	Tên nhà cung cấp thường có tên đầy đủ và tên viết tắt. Đôi khi con có tên tiếng anh. Đi theo tên nhà cung cấp con có những đặc trưng cho mỗi nhà cung cấp như: địa chỉ, điện thoại, fax, tài khoản,...	

Bảng 4.1: Một ví dụ Bảng mô tả chi tiết tài liệu

Các dữ liệu đưa vào các bảng này thường được rút ra từ các báo cáo, chứng từ, tài liệu và những kết quả từ phỏng vấn hay nghiên cứu tài liệu. Chúng là một hình thức làm

tài liệu để lấy ý kiến của người sử dụng và cũng được dùng như tài liệu chính thức cho các bước tiếp sau này.

Thông thường, ở các bước tiếp theo, những bảng này được xem như những dữ liệu đầu vào cho nó. Chỉ trong trường hợp cần thiết người ta mới quay trở lại xem các thông tin gốc như các bảng phỏng vấn, các báo cáo nghiệp vụ,... Vì rằng, những tài liệu gốc thường quá nhiều. Hơn nữa, nhiều dữ liệu trong các hồ sơ thu thập ban đầu thường là những dữ liệu cụ thể, không cần thiết, không đặc trưng cho cấu trúc hay định dạng của mỗi loại dữ liệu. Trên thực tế, cái mà nhà phân tích và thiết kế cần cho các bước tiếp theo chính là các đặc trưng và cấu trúc của mỗi loại dữ liệu.

Dự án	Tiểu dự án: Lập đơn hàng	trang số 20
Loại: Phân tích hiện trạng	Mô tả công việc	số tt: 10 Ngày 06/08/2008
Điều kiện bắt đầu (kích hoạt)	Công việc: Lập đơn hàng, - Tồn kho dưới mức quy định - Đề nghị hấp dẫn của nhà cung cấp - Có đề nghị cung ứng của khách hàng - Đến ngày lập đơn hàng theo quy định quản lý - Thẻ kho, giấy đề nghị, danh sách nhà cung cấp, đơn chào hàng 01 cuộc điện thoại đặt hàng hoặc 01 đơn đặt hàng được lập và gửi đi (có bản mẫu kèm theo) Nhà cung cấp, bộ phận tài vụ, lưu Tùy thuộc vào ngày trong tuần Thứ 2, 7: không xảy ra; Thứ 3, 5: 10-15 lần. Thứ 4, 6: 0-5 lần. 10 phút/ đơn hàng điện thoại. 60 phút / đơn viết - Những đơn hàng trên 2 triệu phải được trưởng bộ phận thông qua (quản lý) - Số lượng đặt mỗi loại dưới mức quy định cho trước (kỹ thuật) - Quy định một số người cụ thể lập đơn hàng (tổ chức) - Đôi khi phải đặt hàng do những tình huống đột xuất, chẳng hạn có dự báo về sự khan hiếm một số mặt hàng trong thời gian tới. - Mức tồn kho tối thiểu chỉ tính cho một số mặt hàng và cách ước lượng của nó còn mang tính chủ quan,	
Thông tin đầu vào:		
Kết quả đầu ra:		
Nơi sử dụng:		
Tần xuất:		
Thời lượng		
Quy tắc		
Lời bình		

Bảng 4.2: Một ví dụ Bảng mô tả chi tiết công việc

4.6.2 Tổng hợp kết quả khảo sát

Khi ta có một bài toán lớn, phức tạp thường không thể quan sát được tất cả các dữ liệu cùng một lúc. Khi tiến hành xác định yêu cầu, người ta phải tiến hành từng nhóm, theo từng lĩnh vực để quan sát và hu thập thông tin. Lúc này cần lắp ghép lại để có được một bức tranh tổng thể. Việc tổng hợp được tiến hành theo hai loại: Tổng hợp theo các xử lý, tổng hợp theo các dữ liệu.

- Tổng hợp theo các xử lý:

Mục tiêu của tổng hợp xử lý là làm rõ các thiếu sót và sự rời rạc của các yếu tố liên quan đến công việc khi phỏng vấn. Sau đó trình bày tường minh để người sử dụng xem xét, đánh giá và hợp thức hóa, đảm bảo sự chính xác của xử lý. Sau đây là ví dụ một bảng tổng hợp công việc

Số tt	Mô tả công việc	Vị trí làm việc	Tần suất	Hồ sơ vào	Hồ sơ ra
T1	Lập đơn hàng: xuất phát từ yêu cầu cung ứng, thực đơn sản xuất, báo giá. Đơn hành lập và chuyển đi bằng điện thoại (80%), viết (20%), sắp các đơn hàng vào sổ đặt để đối chiếu, theo dõi	Quản lý kho hàng	4-5 đơn/ngày 5-10 dòng/đơn	D1 D2	D3 D4
T2

Bảng 4.3: Một ví dụ Bảng tổng hợp công việc

Việc tổng hợp có thể theo các lĩnh vực hoạt động. Lĩnh vực hoạt động được hiểu là một phần của đối tượng nghiên cứu có thể xác định một hay một số mục tiêu có quan hệ với nhau. Chúng thường được mô tả bằng một tập các hoạt động, các quy tắc tổ chức hay quản lý và có các dữ liệu chung. Các lĩnh vực trong một tổ chức thường được mang tên theo thuật ngữ chức năng như: kế toán, lập kế hoạch, dự báo, bán hàng.....

- Tổng hợp các dữ liệu

Mục tiêu của tổng hợp dữ liệu là liệt kê ra tất cả các dữ liệu có liên quan đến miền khảo sát của tổ chức và sàng lọc để thu được những dữ liệu đầy đủ, chính xác và gán cho tên gọi thích hợp mà mọi người tham gia dự án đồng ý. Kết quả tổng hợp dữ liệu có thể có nhiều loại khác nhau. Tuy nhiên, hai tài liệu không thể thiếu là bảng tổng hợp các hồ sơ có trong tổ chức và bảng từ điển dữ liệu về các mục từ lấy ra từ các tài liệu khảo sát và những đặc trưng của nó.

Số tt	Tên – vai trò	Công việc liên quan
D1	Phiếu vật tư: <i>ghi hàng hóa xuất hay nhập</i>	T1
D2	Số thực đơn: <i>định mức hàng hóa làm một sản phẩm</i>	T1
D3	Đơn đặt hàng: <i>ghi lượng hàng đặt gửi nhà cung cấp</i>	T1
D4	Số đặt hàng: <i>tập hợp các đơn hàng đã đặt</i>	T1, T2
D5	Phiếu giao hàng: <i>ghi số lượng hàng của nhà cung cấp phát ra</i>	T2, T3
Di

Bảng 4.4: Một ví dụ Bảng tổng hợp hồ sơ, tài liệu (thực thể dữ liệu)

Số tt	Tên gọi, ý nghĩa	Kiểu	Cỡ	Khuôn dạng	Lĩnh vực	Quy tắc, ràng buộc
1	Số hóa đơn	Ký tự	8		Kế toán	Chữ hoa hoặc số
2	Tên hàng hóa	Ký tự	20		Kế toán
3	Ngày hóa đơn	Ký tự	8	dd-mm-yy	Kế toán
.....

Bảng 4.5: Một ví dụ Bảng tổng hợp từ điển dữ liệu

4.6.3 Hợp thức hóa kết quả khảo sát

Hợp thức hóa kết quả khảo sát là việc hiểu và thể hiện các thông tin khảo sát ở các dạng khác nhau được những người sử dụng và đại diện tổ chức chấp nhận là đúng đắn và đầy đủ. Mục tiêu của hợp thức hóa kết quả khảo sát là nhằm đảm bảo sự chính xác hóa của thông tin và dữ liệu phản ánh yêu cầu thông tin của hệ thống và đảm bảo tính pháp lý của nó cho việc sử dụng sau này.

Việc hợp thức hóa bao gồm việc phân tích viên hoàn chỉnh và trình diễn các dữ liệu thu được để người sử dụng xem xét và cho ý kiến. Tổng hợp các tài liệu đệ trình để các nhà quản lý và lãnh đạo đánh giá và đề xuất bổ sung. Sau đó các tài liệu được hoàn chỉnh và trình bày lại theo những khuôn mẫu xác định để các nhóm và bộ phận quản lý phát triển hệ thống xem xét, thông qua và quyết định chấp nhận.

Câu hỏi thảo luận

- 4.1. Nêu các mục tiêu của việc phỏng vấn.
- 4.2. Kể tên các dạng câu hỏi có thể dùng trong quá trình phỏng vấn.
- 4.3. Nêu ý nghĩa của việc lấy mẫu.
- 4.4. Nêu các phần tử môi trường được phân tích trong kỹ thuật STROBE.

Chương 5. MÔ HÌNH HÓA CHỨC NĂNG

5.1. Mô hình hóa hệ thống

5.1.1. Các bước mô hình hóa hệ thống

Trong chương 3, bạn đã biết về các hoạt động phân tích hệ thống, những hoạt động đó là nhằm mục đích vẽ các mô hình hệ thống. Các mô hình hệ thống đóng vai trò quan trọng trong phát triển hệ thống. Dù là người sử dụng hay người phân tích hệ thống thì bạn đều phải giải quyết những vấn đề phi cấu trúc. Và một cách để cấu trúc vấn đề là vẽ các mô hình.

- **Mô hình** là một biểu diễn hình tượng của thực tế. Các mô hình có thể được xây dựng cho các hệ thống hiện có để giúp chúng ta hiểu kỹ hơn về những hệ thống đó. Hoặc cũng có thể xây dựng mô hình cho các hệ thống được đề xuất nhằm tài liệu hóa các yêu cầu nghiệp vụ hoặc thiết kế kỹ thuật.
- **Mô hình hóa chức năng** (Process Modeling) với biểu đồ luồng dữ liệu (Data Flow Diagram - DFD)
 - Hệ thống làm gì?: Mô hình hóa chức năng là kỹ thuật dùng để tổ chức và tài liệu hóa cấu trúc và luồng dữ liệu xuyên qua các quá trình của một hệ thống và/hoặc các chức năng được thực hiện bởi các quá trình hệ thống.
- **Mô hình hóa dữ liệu** (Data Modeling) với biểu đồ quan hệ thực thể (Entity Relationship Diagram - ERD)
 - Hệ thống có những dữ liệu nào?: Mô hình hóa dữ liệu là kỹ thuật dùng để tổ chức và mô hình hóa dữ liệu của một hệ thống nhằm xác định các yêu cầu nghiệp vụ cho một cơ sở dữ liệu. Đôi khi mô hình hóa dữ liệu còn được gọi là *mô hình hóa cơ sở dữ liệu*.
- **Mô hình hóa đối tượng** (Object Modeling) với ngôn ngữ mô hình hợp nhất (Unified Modeling Language - UML)

5.1.2. Mục đích của mô hình hóa hệ thống

- Để hiểu rõ hơn về hệ thống: các cơ hội để đơn giản hóa, tối ưu hóa (Tái cấu trúc quy trình)
- Để liên kết các hành vi và cấu trúc của hệ thống (các yêu cầu nghiệp vụ về: thông tin/dữ liệu và chức năng/quy trình)
- Để trực quan hóa và điều khiển kiến trúc hệ thống (thiết kế)
- Để kiểm soát những rủi ro trong quá trình phát triển

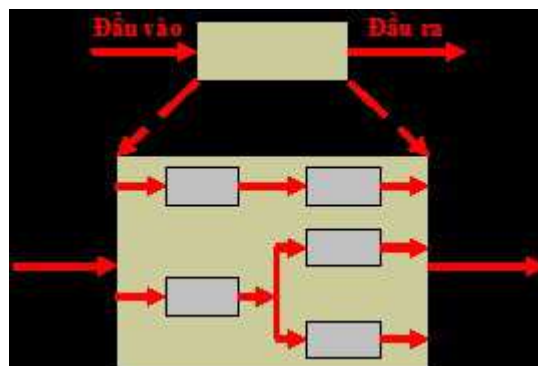
5.1.3. Các thao tác mô hình hóa chức năng

- Lập kế hoạch chiến lược hệ thống

- Các mô hình quá trình nghiệp vụ của tổ chức mô tả các chức năng nghiệp vụ quan trọng
- Tái cấu trúc quy trình nghiệp vụ
 - Các mô hình chức năng “As is” làm đơn giản việc phân tích các điểm yếu (Hệ thống hiện tại).
 - Các mô hình chức năng “To be” làm đơn giản việc cải thiện (Hệ thống mới được đề xuất).
- Phân tích hệ thống
 - Mô hình hóa hệ thống hiện có bao gồm những thiếu sót của nó (DFD logic)
 - Mô hình hóa các yêu cầu logic (các quá trình và luồng dữ liệu cần có dù hệ thống được xây dựng thế nào – DFD logic) của hệ thống được đề xuất.
 - Mô hình hóa các giải pháp kỹ thuật đề cử (DFD vật lý)
 - Mô hình hóa giải pháp được chọn (DFD vật lý)

5.1.4. Khái niệm hệ thống

Một hệ thống tồn tại bằng việc lấy đầu vào từ môi trường, biến đổi (xử lý) đầu vào này và tạo ra một đầu ra. Một hệ thống có thể được phân rã thành nhiều hệ thống con. Một hệ thống con có đầu vào và đầu ra của riêng nó. Đầu ra của một hệ thống con có thể trở thành đầu vào của những hệ thống con khác.



Hình 5-1 Hệ thống và các hệ thống con

- Hệ thống và quá trình
 - **Một hệ thống là một quá trình.** Nó thể hiện một chức năng nghiệp vụ.
 - Một **quá trình** là công việc được thực hiện trên hoặc đáp ứng cho các điều kiện hoặc luồng dữ liệu vào
 - Một quá trình (chức năng) có thể được phân rã thành các quá trình con (các chức năng con, các thao tác).

5.2. Mô hình logic

5.2.1. Phân biệt mô hình logic và mô hình vật lý

- **Mô hình logic** cho biết hệ thống là gì và làm gì. Nó độc lập với việc cài đặt kỹ thuật. Nó minh họa bản chất của hệ thống. Mô hình logic còn có thể được gọi là *mô hình bản chất*, *mô hình khái niệm mô hình nghiệp vụ*.
- **Mô hình vật lý** không chỉ thể hiện hệ thống là gì và làm gì mà còn thể hiện cách thức hệ thống được cài đặt một cách vật lý và kỹ thuật. Nó phản ánh các lựa chọn công nghệ. Mô hình vật lý còn có thể được gọi là *mô hình cài đặt* hay *mô hình kỹ thuật*.

5.2.2. Sự cần thiết của mô hình logic

Các nhà phân tích hệ thống đã nhận thấy giá trị của việc tách riêng việc nghiên cứu nghiệp vụ với việc nghiên cứu kỹ thuật. Đó là lý do tại sao họ sử dụng các mô hình hệ thống logic để minh họa các yêu cầu nghiệp vụ và các mô hình hệ thống vật lý để minh họa các thiết kế kỹ thuật. Các hoạt động của người phân tích hệ thống tập trung chủ yếu vào các mô hình hệ thống logic vì những lý do sau:

- Các mô hình logic loại bỏ tư tưởng thiên lệch do ảnh hưởng bởi cách thức cài đặt hệ thống đã có hoặc ý kiến chủ quan của một người nào đó về cách cài đặt cho hệ thống. Do đó, chúng khuyến khích tính sáng tạo.
- Các mô hình logic làm giảm khả năng bỏ sót các yêu cầu nghiệp vụ trong trường hợp con người bị chi phối quá nhiều vì các kết quả mang tính kỹ thuật. Nhờ việc tách biệt những gì hệ thống phải làm với cách thức hệ thống thực hiện mà chúng ta có thể phân tích tốt hơn các yêu cầu nhằm đảm bảo tính hoàn thiện, chính xác và nhất quán.
- Các mô hình logic cho phép truyền đạt với người dùng cuối dưới dạng ngôn ngữ phi kỹ thuật hoặc ít kỹ thuật hơn.

5.3. Biểu đồ phân rã chức năng

5.3.1. Khái niệm BFD

BFD là sơ đồ phân rã có thứ bậc các chức năng của hệ thống từ tổng thể đến chi tiết. Mỗi chức năng có thể có một hoặc nhiều chức năng con, tất cả được thể hiện trong một khung của sơ đồ.

- Ý nghĩa của BFD:
 - Giới hạn phạm vi của hệ thống cần phải phân tích.
 - Tiếp cận hệ thống về mặt logic nhằm làm rõ các chức năng mà hệ thống thực hiện để phục vụ cho các bước phân tích tiếp theo.
 - Phân biệt các chức năng và nhiệm vụ của từng bộ phận trong hệ thống, từ đó loại bỏ những chức năng trùng lặp, dư thừa.

Tuy nhiên, BFD không có tính động, nó chỉ cho thấy các chức năng mà không thể hiện trình tự xử lý của các chức năng đó cũng như là sự trao đổi thông tin giữa các chức năng. Do đó, BFD thường được sử dụng làm mô hình chức năng trong bước đầu phân tích.

5.3.2. Phương pháp xây dựng BFD

a. Phân mức các chức năng

- BFD được xây dựng theo mô hình phân mức, mỗi một chức năng có thể gồm một hoặc nhiều chức năng con trong sơ đồ.
- Chú ý là với một sơ đồ, không nên có quá 6 mức, thông thường thì 3 mức là phù hợp với các hệ thống trung bình.
- Với mỗi chức năng không nên có quá 6 chức năng con vì như vậy sẽ làm sơ đồ trở nên phức tạp và khó kiểm soát. Nếu gặp trường hợp có quá nhiều chức năng con thì có thể giải quyết bằng cách tạo thêm mức trung gian để nhóm các chức năng con lại.
- Cần đảm bảo tính cân bằng của sơ đồ, nghĩa là các chức năng thuộc cùng một mức nên có sự tương đương nhau về kích thước và độ phức tạp
- Mỗi chức năng phải mang một tên duy nhất, không trùng lặp với chức năng khác; tên phải thể hiện khái quát các chức năng con của nó, phản ánh được thực tế nghiệp vụ mà nó thực hiện. Tên của chức năng phải bắt đầu bằng động từ, ví dụ như “lập đơn hàng”.

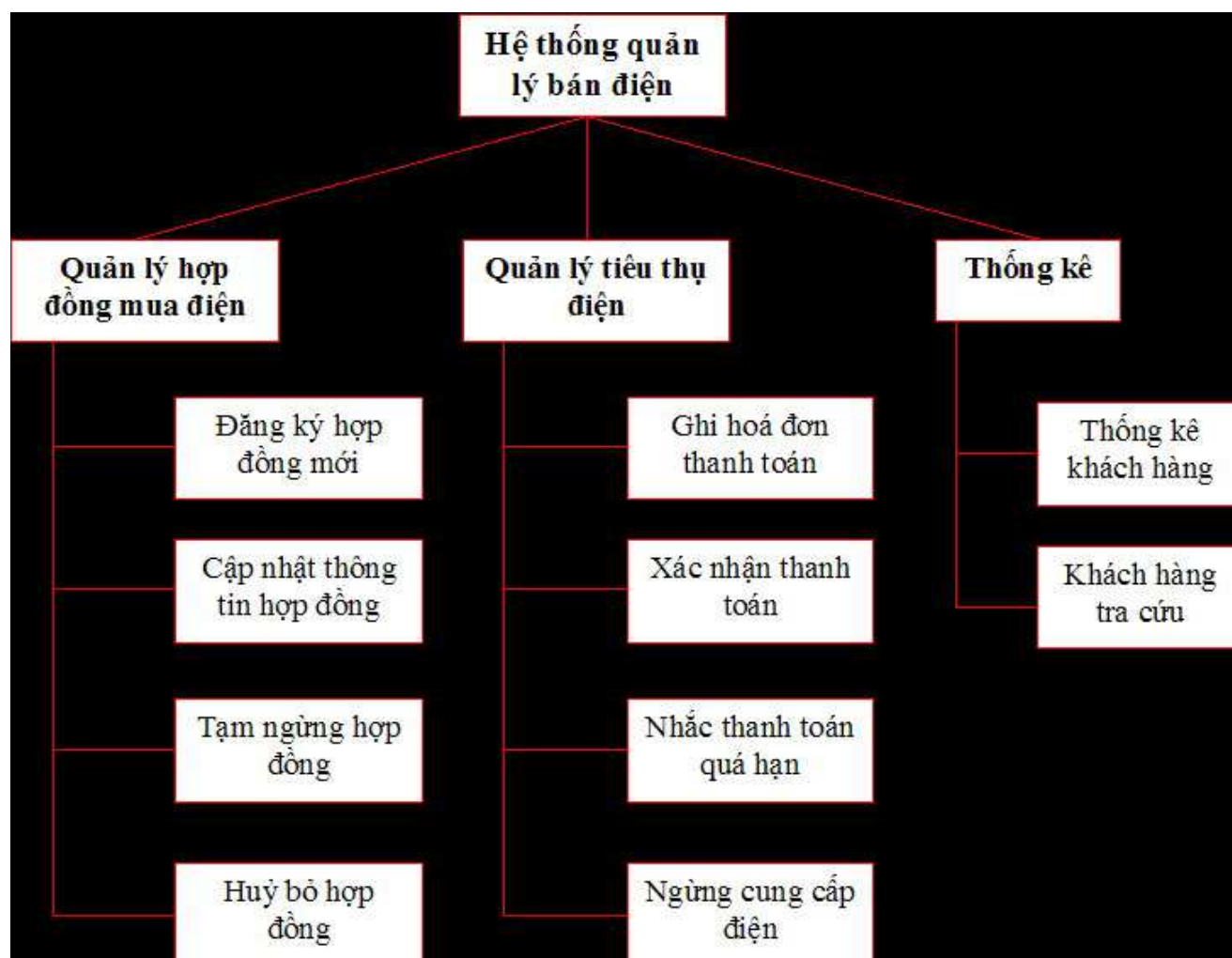
Tuân thủ những nguyên tắc trên sẽ giúp cho việc xây dựng các mô hình dữ liệu tiếp theo được rõ ràng.

b. Xác định các chức năng

- Ở mức cao nhất của nghiệp vụ, chức năng chính có thể là một trong các loại sau:
 - Sản xuất sản phẩm.
 - Cung cấp dịch vụ (bán hàng, bảo dưỡng).
 - Quản lý tài nguyên (tài sản, nguồn nhân lực, con người...).
- Khi đã xác định được loại mà nó thuộc vào thì sẽ đặt tên cho chức năng cao nhất này.
- Tiếp theo, để xác định các chức năng con thì từ chức năng chính, ta đặt nó trong chu kỳ sống gồm các giai đoạn:
 - Xác định nhu cầu.
 - Mua bán.
 - Bảo hành, bảo dưỡng.
 - Thanh lý hoặc chuyển nhượng.

Mỗi giai đoạn có thể có một hoặc nhiều chức năng con. Ví dụ, với chức năng Bán hàng thì ở giai đoạn xác định nhu cầu có thể có chức năng con là Quản lý thông tin khách hàng, ở giai đoạn mua bán thì có thể là Cập nhật đơn hàng...

- Người phân tích phải xác định được mức nào là thấp nhất, khi đó sẽ dừng việc phân tích chức năng. Để nhận biết một chức năng mức thấp nhất bằng cách xét xem có phải chức năng đó chỉ có một nhiệm vụ hoặc một nhóm các nhiệm vụ nhỏ.
- Khi xây dựng BFD cần đảm bảo tính đơn giản, rõ ràng và chính xác của sơ đồ. Với các hệ thống lớn, có thể trình bày BFD trên nhiều trang, trang 1 là BFD mức cao nhất (mức 0), tiếp theo ứng với mỗi chức năng sẽ được phân tích ở các trang sau tới chức năng mức thấp nhất thì dừng.



Hình 5-2: Ví dụ biểu đồ phân cấp chức năng Hệ thống quản lý bán điện

5.4. Biểu đồ luồng dữ liệu (DFD)

5.4.1. Mô hình hóa chức năng với DFD

Trong phần này, chúng ta tập trung vào việc mô hình hóa chức năng logic trong giai đoạn phân tích hệ thống. Như trên đã nói, mô hình hóa chức năng là một kỹ thuật để tổ chức và tài liệu hóa cấu trúc và luồng dữ liệu cũng như logic, đường lối và các thủ tục được thực hiện bởi các quá trình của một hệ thống. Một trong các mô hình chức năng phân tích hệ thống chính là sơ đồ luồng dữ liệu.

- Một sơ đồ luồng dữ liệu (Data Flow Diagram – DFD) là một công cụ đồ họa để mô tả luồng dữ liệu luân chuyển trong một hệ thống và những hoạt động xử lý được thực hiện bởi hệ thống đó. Sơ đồ luồng dữ liệu còn có các tên gọi khác là *biểu đồ bọt*, *biểu đồ biến đổi* và *mô hình chức năng*.
- Tại sao sử dụng DFD?
 - Sự mô tả bằng ngôn ngữ hướng tới sự giải thích, nó có thể bỏ sót những thông tin quan trọng.
 - Sự mô tả đồ họa minh họa được luồng dữ liệu trong một tổ chức thông qua DFD.

Biểu đồ luồng dữ liệu đã rất phổ biến hơn 20 năm nay nhưng lợi ích của DFD đã được đổi mới nhờ vào tính ứng dụng của nó trong việc tái cấu trúc quy trình nghiệp vụ (business process redesign – BPR). Khi mà tổ chức nhận thấy rằng hầu hết các hệ thống xử lý dữ liệu đã trở nên lỗi thời, không hiệu quả và rườm rà về thủ tục thì đó là lúc có thể thu lợi ích mới nhờ vào việc tổ chức lại các quy trình nghiệp vụ. Điều này được tiến hành trước tiên bằng việc mô hình hóa các quy trình nghiệp vụ nhằm mục đích phân tích, thiết kế lại và/hoặc cải thiện chúng. Tiếp theo, công nghệ thông tin có thể được áp dụng một cách sáng tạo cho các quy trình nghiệp vụ đã được cải thiện nhằm tối đa hóa giá trị thu về cho tổ chức.

5.4.2. Vai trò của DFD

DFD tài liệu hóa một thao tác/hoạt động/chức năng nghiệp vụ của một hệ thống thành một quá trình. DFD mô tả cách thức dữ liệu được xử lý trong và tại biên giới của hệ thống. DFD thể hiện chi tiết sự phụ thuộc lẫn nhau giữa các quá trình của hệ thống, các sự dịch chuyển dữ liệu hoặc thông tin giữa các quá trình.

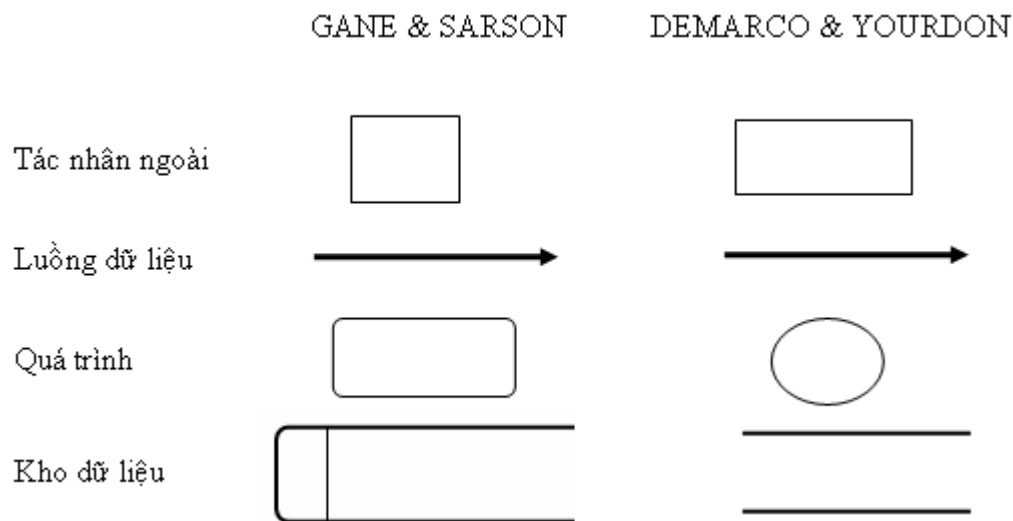
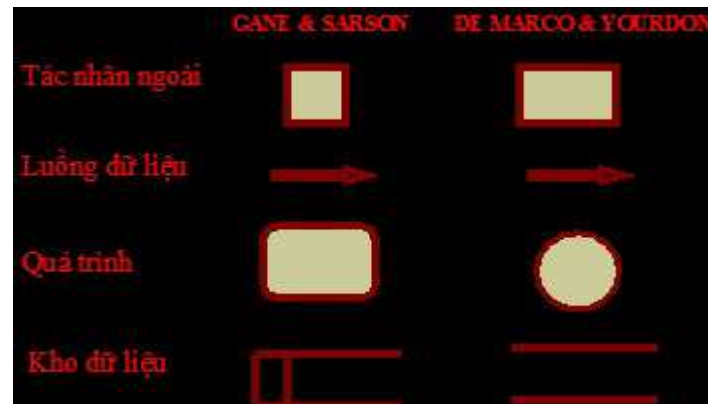
- DFD logic mô tả luồng thông tin của một hệ thống
- DFD vật lý mô tả cách thức một hệ thống thông tin được cài đặt vật lý (ai làm, bằng cách nào, bằng công cụ nào)

- Mô tả cách thức một hệ thống thông tin được cài đặt vật lý (ai làm, bằng cách nào, bằng công cụ nào)

5.5. Các phần tử của DFD

5.5.1. Các bộ ký hiệu DFD

Có một vài bộ ký hiệu DFD mà phổ biến là Gane/Sarson và Demarco/Yourdon. Trong đó, bộ ký hiệu Gane & Sarson được sử dụng phổ biến hơn.



Hình 5-3: Các bộ ký hiệu DFD

5.5.2. Tác nhân ngoài

- Khái niệm: Một tác nhân ngoài là một nguồn cung cấp hoặc nhận thông tin dữ liệu của hệ thống. Một tác nhân ngoài không phải là một phần của hệ thống, nó thể hiện mối quan hệ giữa hệ thống với môi trường bên ngoài

- Nhãn: Tên của tác nhân ngoài phải là một **đanh từ**

Một tác nhân ngoài xác định một người, một đơn vị của tổ chức hay một tổ chức khác nằm ngoài phạm vi của dự án nhưng có tương tác với hệ thống đang được nghiên cứu.

Các tác nhân ngoài xác định “biên giới” hay phạm vi của hệ thống đang được mô hình hóa. Khi phạm vi thay đổi, các tác nhân ngoài có thể trở thành các quá trình và ngược lại

Tác nhân ngoài thường là:

- o Phòng ban, bộ phận trong tổ chức nhưng nằm ngoài phạm vi hệ thống.
- o Một chi nhánh hoặc tổ chức bên ngoài
- o Một hệ thống thông tin khác của tổ chức
- o Người dùng cuối hoặc người quản lý của hệ thống

Ví dụ:

Nhà cung cấp

5.5.3. Kho dữ liệu

- Khái niệm: Một kho dữ liệu là một kho lưu trữ dữ liệu, nó chứa thông tin.

- Kho chứa vật lý là phi vật chất, nó có thể là một tủ hồ sơ, sách hoặc file máy tính.

- Nhãn: Tên của kho dữ liệu phải bắt đầu bằng **đanh từ**, nó nói lên nội dung thông tin.

- Một kho dữ liệu là “dữ liệu tĩnh” khác với luồng dữ liệu là “dữ liệu chuyển động”. Một kho dữ liệu cần biểu diễn cho “những thứ” mà tổ chức muốn lưu trữ dữ liệu, “những thứ” thường là:

o Con người: ví dụ như khách hàng, phòng, nhân viên, thầy giáo, sinh viên, nhà cung cấp...

o Các địa điểm: ví dụ như nơi sinh, tòa nhà, phòng, chi nhánh...

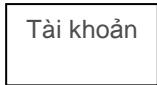
o Các đối tượng: ví dụ như sách, máy móc, sản phẩm, nguyên liệu thô, bản quyền phần mềm, gói phần mềm, công cụ, phương tiện vận tải...

o Các sự kiện (dữ liệu được thu thập về chúng): ví dụ như việc bán hàng, giải thưởng, lớp học, chuyến bay, hóa đơn, đơn hàng, đăng ký, đặt chỗ...

o Các khái niệm (dữ liệu về chúng rất quan trọng): ví dụ như việc giảm giá, tài khoản, khóa học, chất lượng...

Có thể xác định các kho dữ liệu với các yếu tố Tài nguyên – Sự kiện – Tác nhân – Địa điểm. Các kho dữ liệu được mô tả trong một DFD chứa tất cả các thể hiện của các thực thể dữ liệu (được mô tả trong một biểu đồ quan hệ quan hệ ERD).

Ví dụ:



5.5.4. Luồng dữ liệu

- Khái niệm:

Một luồng dữ liệu biểu diễn một sự di chuyển của dữ liệu (thông tin) giữa các quá trình hoặc kho dữ liệu.

Một luồng dữ liệu không biểu diễn một tài liệu hay một vật thể vật lý: nó biểu diễn sự trao đổi thông tin trong tài liệu hoặc về vật thể.

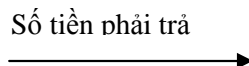
- Nhãn: Phải có tên và không trùng lặp với các luồng dữ liệu khác. Tên phải thể hiện logic của thông tin chứ không phải dạng vật lý của nó và phải bắt đầu bằng danh từ

- Một luồng dữ liệu biểu diễn một đầu vào dữ liệu tới một quá trình hoặc đầu ra dữ liệu từ một quá trình.

- Một luồng dữ liệu cũng có thể được dùng để biểu diễn việc tạo, đọc, xóa hoặc cập nhật dữ liệu trong một file hoặc cơ sở dữ liệu (được gọi là kho dữ liệu).

- Một luồng dữ liệu ghép (gói) là một luồng dữ liệu chứa các luồng dữ liệu khác.

Ví dụ:



5.5.5. Quá trình

- Khái niệm:

Một quá trình là một hoạt động được thực hiện trên luồng dữ liệu vào để tạo một luồng dữ liệu ra.

Là chức năng được thực hiện bởi hệ thống để đáp ứng lại các luồng dữ liệu hoặc điều kiện vào.

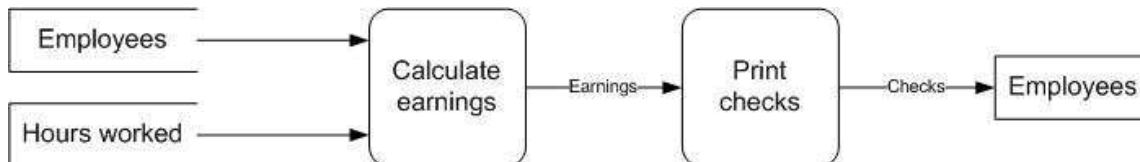
- Nhãn: Sử dụng một **động từ** để gán nhãn cho hành động thực hiện bởi quá trình (không phải là tên của người hay phòng ban thực hiện nó trong DFD vật lý).

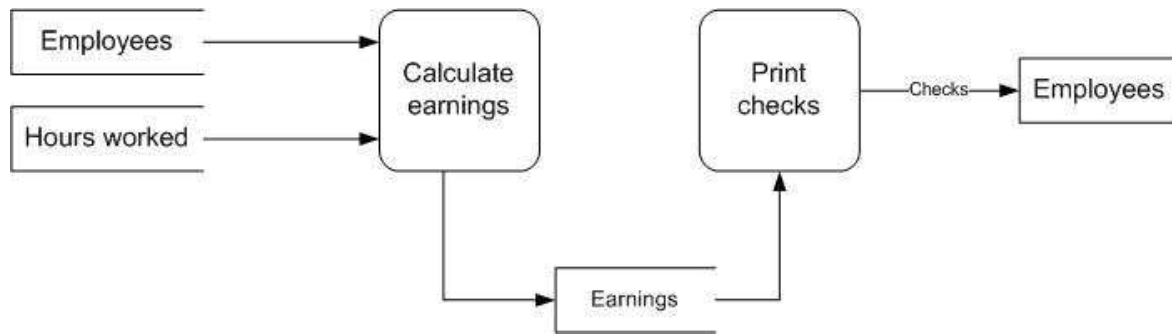
- Một quá trình phải có ít nhất một luồng dữ liệu vào và ít nhất một luồng dữ liệu ra.

Ví dụ:

Thanh toán hóa đơn

- Các loại quá trình logic
 - Chức năng: một tập các hoạt động tiếp diễn liên quan tới một nghiệp vụ; ví dụ như việc bán hàng
 - Sự kiện (hay giao dịch, hoạt động): một công việc phải được hoàn thành toàn bộ (hoặc như một phần của một chức năng); ví dụ như việc thu tiền thanh toán (là một công đoạn trong việc bán hàng)
 - Quá trình cơ bản (hay thao tác): một hoạt động hoặc thao tác chi tiết, rời rạc được yêu cầu để đáp lại một sự kiện. Thông thường, một số thao tác như vậy phải được hoàn thành để đáp ứng một sự kiện; ví dụ như ghi tiền thành toán
- Cách tách các quá trình
 - Mỗi hệ thống có thể được chia thành nhiều quá trình khác nhau bằng các cách khác nhau. Các quá trình có thể được tách nếu có một luồng thông tin đi giữa chúng.
 - Điều kiện để tách: nếu các tiến trình này không thực hiện đồng thời hoặc không cùng một nơi hoặc không do một người thực hiện. Khi đó, ta kiểm tra quá trình tách bằng cách cuối luồng dữ liệu ta đặt câu hỏi:
 - Tiến trình tiếp theo có thể thực hiện ở thời gian khác được không?
 - Tiến trình tiếp theo có thể thực hiện ở nơi khác được không?
 - Tiến trình tiếp theo có thể được thực hiện bởi người khác được không?
- Nếu một trong ba câu hỏi trên là có thì ta tách chúng bằng cách đặt một tệp dữ liệu ở giữa.





Hình 5-4: Ví dụ cách tách các quá trình

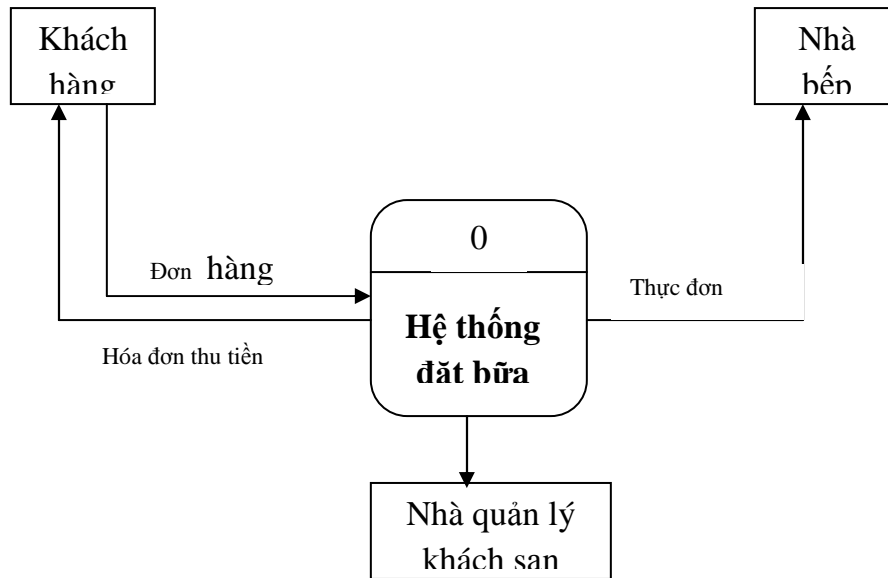
5.6. Biểu đồ luồng dữ liệu mức ngữ cảnh

Biểu đồ luồng dữ liệu mức ngữ cảnh (Context data flow diagram) là một mô hình chức năng được dùng để tài liệu hóa phạm vi của một hệ thống. Nó còn được gọi là *mô hình môi trường*. Để xây dựng biểu đồ ngữ cảnh, cần phải:

- Xác định biên giới của hệ thống
- Xác định các tác nhân ngoài
- Không chi tiết về các quá trình và kho dữ liệu của hệ thống
- Chiến lược cụ thể xây dựng biểu đồ ngữ cảnh:
 - Coi cả hệ thống là một “hộp đen”, xem nó là một chức năng duy nhất và chỉ quan tâm tới phần bên ngoài của nó mà không phải xét tới những hoạt động bên trong.
 - Hỏi người sử dụng cuối của hệ thống về những giao dịch nghiệp vụ mà hệ thống phải đáp ứng. Đó chính là các luồng vào của hệ thống. Với mỗi luồng vào, cần xác định nguồn tương ứng của nó. Các nguồn sẽ trở thành các tác nhân ngoài trong sơ đồ ngữ cảnh.
 - Hỏi người sử dụng cuối về những đáp ứng phải được sinh ra bởi hệ thống. Đó chính là các luồng ra của hệ thống, Với mỗi luồng ra, xác định đích của nó. Các đích đó cũng sẽ trở thành các tác nhân ngoài.
 - Xác định các kho dữ liệu ngoài. Rất nhiều hệ thống đòi hỏi truy nhập vào các tệp hoặc cơ sở dữ liệu của hệ thống khác.

Vẽ một biểu đồ ngữ cảnh dựa trên tất cả các thông tin đã xử lý. Chú ý chỉ minh họa những luồng dữ liệu thể hiện những mục tiêu chính của hệ thống nhằm tránh việc biểu đồ ngữ cảnh có quá nhiều luồng dữ liệu vào/ra.

Trong biểu đồ ngữ cảnh không có các kho dữ liệu



Hình 5-5: Biểu đồ ngữ cảnh hệ thống đặt bữa ăn

5.7. Trình tự và quy tắc xây dựng DFD

5.7.1. Các bước xây dựng DFD

Kỹ thuật phổ biến được dùng để xây dựng DFD là kỹ thuật phân mức. Dựa theo BFD của hệ thống, chúng ta sẽ xây dựng DFD theo nhiều mức, mỗi mức thể hiện trên một hoặc nhiều trang. Nên đặt tên cho mỗi trang bằng tên của chức năng đang được phân tích trên trang đó. Như vậy với trang phân tích tại mức 0 thì tên của nó chính là tên của hệ thống.

Biểu đồ ngữ cảnh

- o Xác định hệ thống và giới hạn của nó (ngữ cảnh)
- o Xác định các tác nhân ngoài (người cung cấp, người nhận thông tin hệ thống của)
- o Xác định các luồng dữ liệu ngoài (đầu vào, đầu ra)
- o Chú ý: bản thân toàn bộ hệ thống là một quá trình (nó nhận đầu vào và biến đổi nó thành đầu ra)

Tóm lại, một biểu đồ ngữ cảnh chỉ gồm 3 thành phần sau:

Một tiến trình duy nhất: (được đánh bằng chữ số 0 và có tên và có tên bên trong) để mô tả toàn bộ hệ thống (hay lĩnh vực) đang xét. Tên của tiến trình này là tên của toàn bộ HTTT (hay một hệ con của nó) mà ta cần xây dựng.

Các tác nhân (bên ngoài hệ thống) có mối quan hệ về thông tin với hệ thống (tác nhân có thể là người, là một tổ chức hay một HTTT khác).

Các luồng dữ liệu đi từ các tác nhân vào hệ thống hay từ hệ thống đến các tác nhân.

-DFD mức 0

o Xác định những gì đang được thực hiện ra giữa từng đầu vào và đầu ra tương ứng

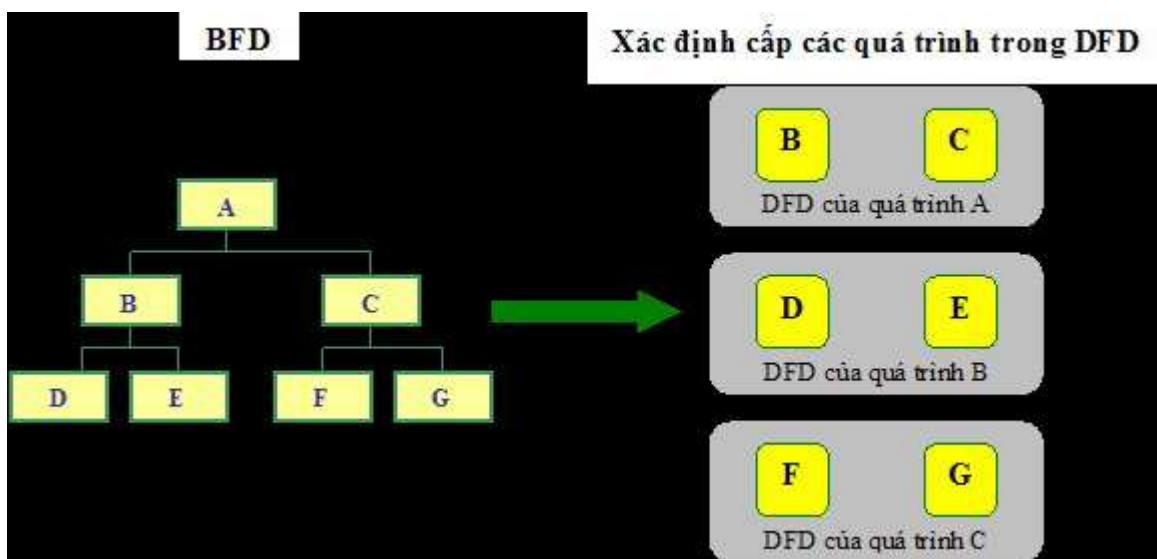
o Xác định các quá trình

o Xác định các luồng dữ liệu ngoài giữa các tác nhân ngoài với các quá trình

o Xác định các luồng dữ liệu ngoài giữa các quá trình với các kho dữ liệu

-Các DFD mức 1

o Là các quá trình con của các quá trình mức 0



Hình 5-6: Kỹ thuật phân mức DFD dựa trên BFD

5.7.2. Các quy tắc xây dựng DFD

- Qui tắc 1: mỗi biểu tượng có riêng một nhãn duy nhất để tránh gây hiểu nhầm
- Qui tắc 2: Sử dụng một ĐỘNG TỪ để gán nhãn cho một quá trình (vì một quá trình là một hành động!!!)
- Qui tắc 3: Mỗi luồng dữ liệu phải được liên kết với ít nhất một quá trình
- Qui tắc 4: Phải có một góc tô đậm trong tất cả các thể hiện của một biểu tượng lặp trong cùng một biểu đồ
- Qui tắc 5: Một quá trình phải luôn có luồng dữ liệu vào và ra
- Qui tắc 6: Không cần có một luồng dữ liệu mà không có sự biến đổi liên kết với một quá trình (vì hoạt động như vậy là vô giá trị)
- Qui tắc 7: Các quá trình cha và các quá trình con tương ứng của nó phải có các luồng dữ liệu vào ra giống nhau (nhưng các quá trình con có thể có luồng dữ liệu của riêng nó)
- Qui tắc 8: Các luồng dữ liệu không thể tự phân tách được
- Qui tắc 9: Một gói dữ liệu có thể gồm nhiều phần tử dữ liệu được truyền đi đồng thời tới cùng một đích
- Qui tắc 10: Không được sử dụng mũi tên hai chiều vì luồng vào (cập nhật) và luồng ra (trích thông tin) của một kho dữ liệu mang nội dung thông tin khác nhau

5.7.3. Phân tích hướng cấu trúc cổ điển (top-down)

1. **Vẽ DFD logic dạng top-down của hệ thống hiện có:** Vẽ các biểu đồ DFD vật lý dạng top-down để biểu diễn sự cài đặt vật lý hiện tại của hệ thống bao gồm các giới hạn của nó.
2. Chuyển đổi các DFD vật lý thành các DFD logic tương ứng của nó.
3. Vẽ các DFD logic dạng top-down biểu diễn một hệ thống được cải thiện.
4. Mô tả tất cả các luồng dữ liệu, kho dữ liệu, quy tắc và thủ tục trong một từ điển dữ liệu.
5. Một cách tùy chọn, đánh dấu các bản sao của DFD logic để biểu diễn các giải pháp vật lý khác nhau.
6. Vẽ các DFD vật lý dạng top-down biểu diễn giải pháp được lựa chọn.

5.7.4. Phân tích hướng cấu trúc hiện đại (hướng sự kiện)

1. Vẽ một DFD ngữ cảnh để xác lập phạm vi ban đầu của dự án.
2. Vẽ một biểu đồ phân rã chức năng để phân chia hệ thống thành các hệ thống con.

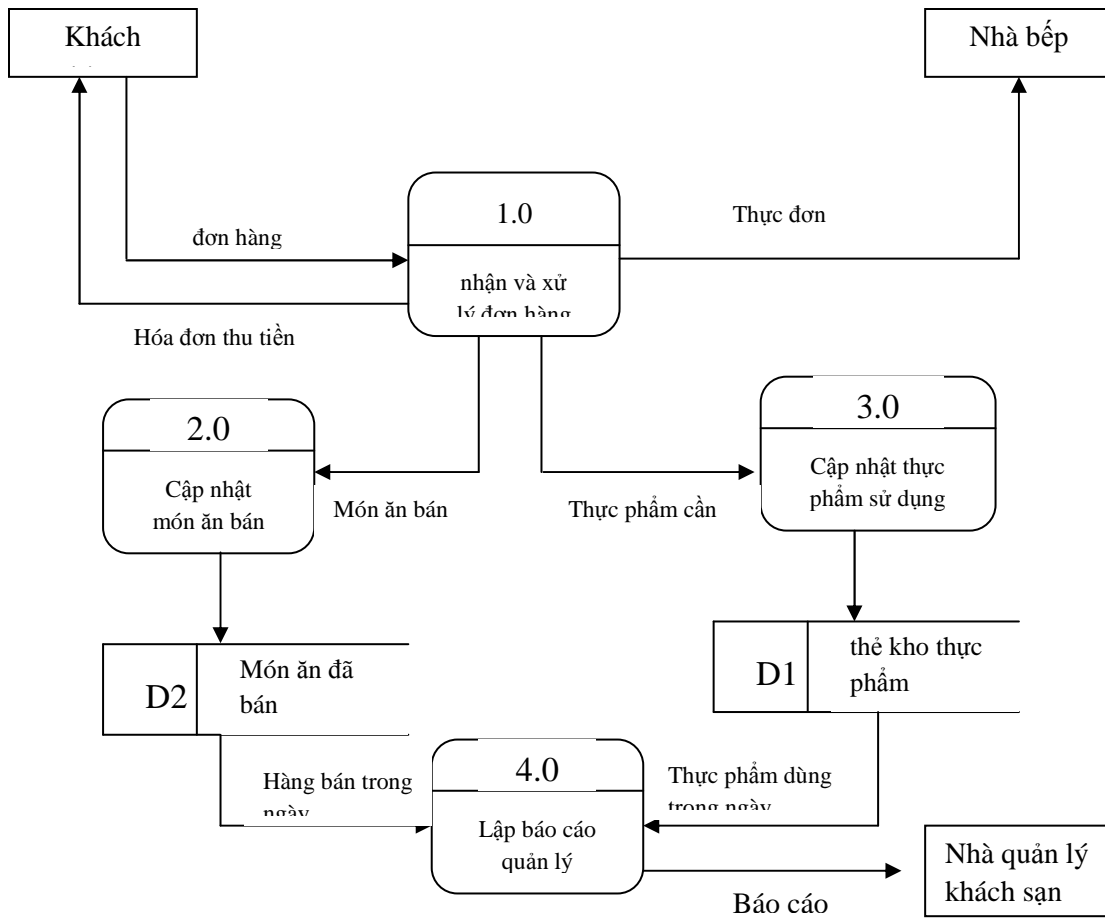
3. Tạo một danh sách các đáp ứng sự kiện hay use-case cho hệ thống để xác định các sự kiện mà hệ thống phải có đáp ứng.
4. Vẽ một DFD sự kiện (hay bộ xử lý sự kiện) cho từng sự kiện.
5. Kết hợp các DFD sự kiện thành một biểu đồ hệ thống (hay đối với các hệ thống lớn thì là các biểu đồ hệ thống con).
6. Vẽ các DFD chi tiết cho các bộ xử lý sự kiện phức tạp hơn.
7. Tài liệu hoá các luồng dữ liệu và quá trình trong từ điển dữ liệu.

5.7.5. So sánh DFD với biểu đồ tiến trình (flowchart)

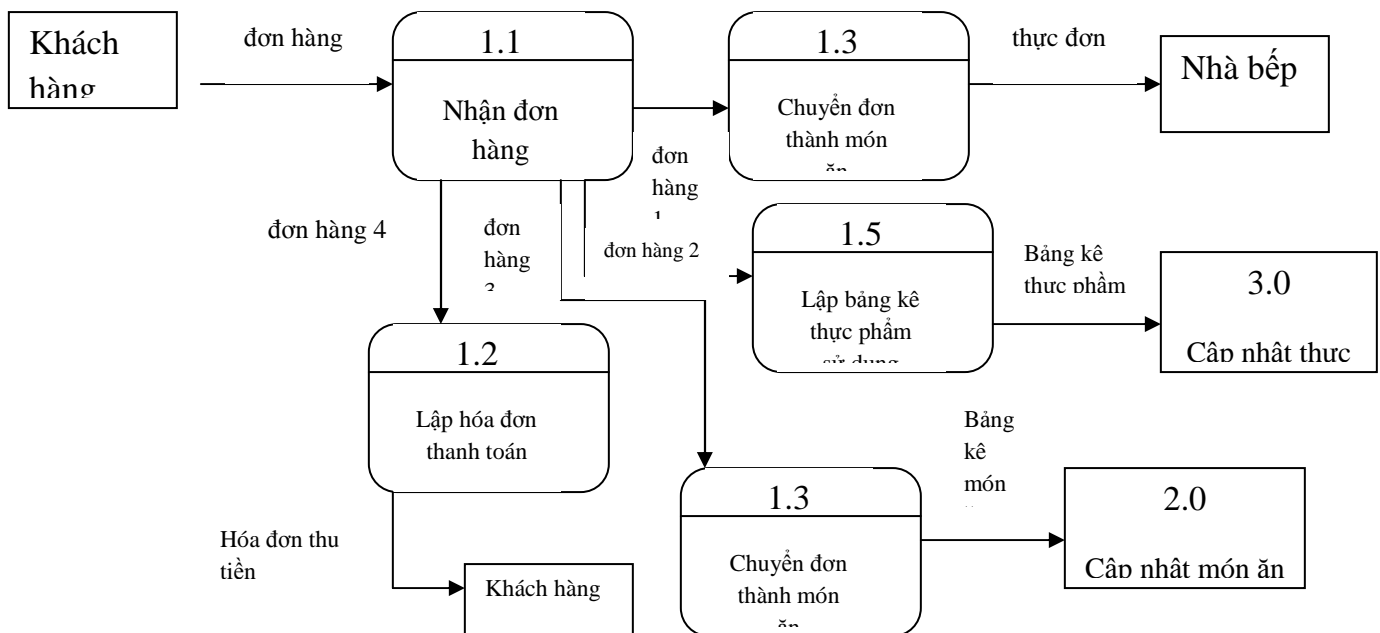
- Các quá trình trong DFD có thể thực hiện song song (vào cùng một thời gian): Các quá trình trong flowchart thực hiện mỗi cái vào một thời điểm
- DFD thể hiện luồng dữ liệu xuyên suốt hệ thống: Flowchart thể hiện luồng điều khiển (trình tự và sự chuyển giao điều khiển)
- Các quá trình trong một DFD có thể có thời gian khác nhau đáng kể (hàng ngày, hàng tuần, cấp bách): Các quá trình trong flowchart là một phần của một chương trình đơn nhất với thời gian nhất quán

5.7.6. Sự bảo toàn dữ liệu

- Là việc bảo đảm rằng một luồng dữ liệu chỉ chứa dữ liệu cần thiết cho quá trình nhận nó.
 - Xác định và loại bỏ những phần không hiệu quả
 - Đơn giản hoá việc giao tiếp giữa các quá trình
 - Phải xác định chính xác kết cấu dữ liệu của từng luồng dữ liệu, được biểu diễn dưới dạng các cấu trúc dữ liệu (trong phần Mô hình hoá dữ liệu)



Hình 5-7: DFD mức 0 của của hệ thống đặt bữa ăn



Hình 5-8: Biểu đồ mức 1 nhận được khi phân rã tiến trình 1.0 từ biểu đồ mức 0

Câu hỏi thảo luận C5

5.1. Hệ thống là gì?

5.2. Nêu mục đích của việc mô hình hóa hệ thống.

5.3. Phân biệt mô hình logic và mô hình vật lý.

GỢI Ý: tham khảo mục Mô hình logic.

5.4. Giải thích tại sao mô hình logic lại đóng vai trò quan trọng trong phân tích hệ thống?

5.5. Ý nghĩa của biểu đồ phân cấp chức năng là gì?

5.6. Tại sao trong BFD, mỗi chức năng không nên có quá 6 chức năng con?

GỢI Ý: tham khảo mục Phương pháp xây dựng BFD.

5.7. Nêu vai trò của DFD.

5.8. Khi xây dựng DFD, có thể gặp những sai sót nào?

5.9. Thảo luận về BFD cho hệ thống quản lý bán điện trong hình 5.3

5.10. Giải thích ví dụ về cách tách các quá trình trong hình 5-5.

5.11. Thảo luận về biểu đồ ngữ cảnh của hệ thống đặt bữa ăn

5.12. Thảo luận về DFD mức 0 của hệ thống quản lý đặt bữa ăn

5.13 Xây dựng các DFD còn lại cho hệ thống quản lý bán điện và hệ thống đặt bữa ăn

Chương 6. MÔ HÌNH HÓA DỮ LIỆU

6.1. Mô hình hóa dữ liệu

6.1.1. Khái niệm

Mô hình hoá dữ liệu (mô hình hoá cơ sở dữ liệu, mô hình hoá thông tin) là một kỹ thuật để tổ chức và tài liệu hoá dữ liệu của hệ thống trong một mô hình. Kỹ thuật này xác định các yêu cầu nghiệp vụ đối với một cơ sở dữ liệu. Mô hình hóa dữ liệu thường được gọi là mô hình hóa cơ sở dữ liệu vì cuối cùng một mô hình dữ liệu luôn được cài đặt thành cơ sở dữ liệu.

Biểu đồ quan hệ thực thể (Entity Relationship Diagram - ERD) mô tả dữ liệu dưới dạng các thực thể và các quan hệ được mô tả bởi dữ liệu. ERD xác định các đơn vị thông tin cơ sở cần thiết cho hệ thống (các thực thể) và các mối quan hệ giữa chúng. Nghĩa là tất cả các dữ liệu chỉ được lưu giữ một lần trong toàn bộ hệ thống.

6.1.2. Từ mô hình dữ liệu tới cài đặt cơ sở dữ liệu

ERD: là một mô hình khái niệm của các thực thể dữ liệu, các thuộc tính (đặc điểm) và các quan hệ (với các thực thể khác) của chúng trong một hệ thống thông tin (độc lập kỹ thuật). (Phân tích)

Mô hình dữ liệu quan hệ (Relational Data Model - RDM): một bản thiết kế cho việc cài đặt của một mô hình dữ liệu khái niệm (ERD) trong môi trường cơ sở dữ liệu quan hệ (độc lập phần mềm). (Thiết kế)

Sơ đồ quan hệ : một sơ đồ thể hiện cách thức một mô hình dữ liệu được cài đặt với hệ quản trị cơ sở dữ liệu (như Microsoft Access hay MS SQL Server...). (Cài đặt)

6.1.3. Vai trò của biểu đồ quan hệ thực thể

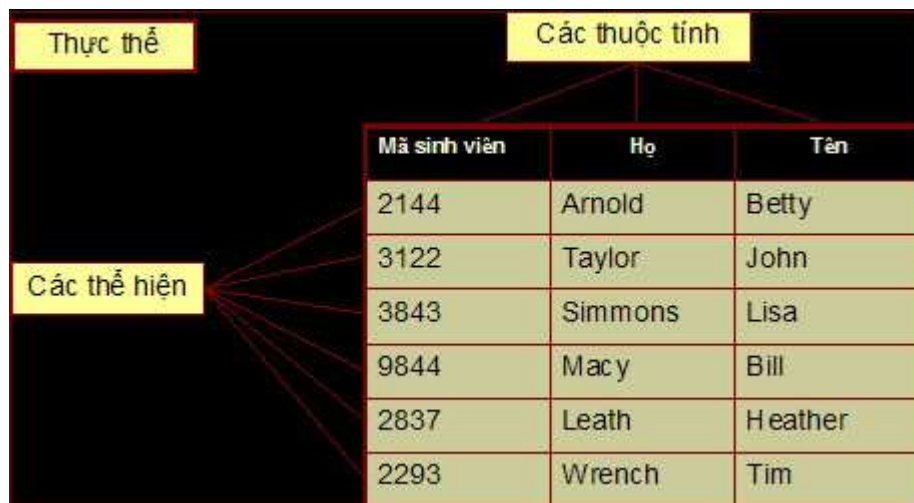
- Cơ sở dữ liệu = dữ liệu + quan hệ
- ERD được dùng để mô hình hoá dữ liệu và quan hệ của chúng
- ERD là một biểu diễn đồ họa của mô hình dữ liệu khái niệm
- ERD là độc lập tài nguyên: nó không gắn với bất cứ môi trường cơ sở dữ liệu cụ thể nào

6.2. Các phần tử của biểu đồ quan hệ thực thể (ERD)

6.2.1. Thực thể

Thực thể là một nhóm các thuộc tính tương ứng với một đối tượng khái niệm mà chúng ta cần thu thập và lưu trữ dữ liệu về nó

- Các vật thể, con người, địa điểm, sự kiện, khái niệm mà sự tồn tại của nó không phụ thuộc vào các thực thể khác
- Thực thể là một tập các thể hiện của đối tượng mà nó biểu diễn
- Thực thể phải có một tên duy nhất (một danh từ số ít), từ định danh duy nhất và ít nhất một thuộc tính (chính là từ định danh)
- Các loại thực thể có thể có:
 - Con người: là những người thực hiện chức năng nào đó trong hoặc ngoài hệ thống. Ví dụ: công ty, khách hàng, phòng ban, bộ phận, nhân viên, giáo viên, sinh viên, nhà cung cấp...
 - Địa điểm: là nơi được sử dụng bởi con người. Ví dụ: nơi bán hàng, toà nhà, phòng, chi nhánh...
 - Vật thể: là những đối tượng vật lý thấy rõ. Ví dụ: sách, tạp chí, sản phẩm, nguyên liệu thô, công cụ...
 - Sự kiện: là những gì xảy ra theo thời gian hoặc theo một quy trình nhất định. Ví dụ: giải thưởng, sự huỷ bỏ, chuyến bay, giờ học, việc lập hoá đơn, việc đặt hàng, việc đăng ký, sự gia hạn, sự đặt chỗ, việc bán hàng...
 - Khái niệm: là những gì không thể nhìn thấy được. Ví dụ: tài khoản, khoảng thời gian, khoá học, nguồn tài chính, quy tắc, luật lệ...
- Trong ERD, thực thể được ký hiệu là một hình chữ nhật, mỗi thực thể tương đương với một bảng dữ liệu trong cơ sở dữ liệu của hệ thống.
- Thể hiện của thực thể: là một thực thể cụ thể. Ví dụ thực thể SinhVien có thể có nhiều thể hiện như Betty, John, Lisa...



Hình 6.1: Ví dụ thực thể và thể hiện

6.2.2. Thuộc tính

Mỗi thực thể bao gồm nhiều thông tin, mỗi thông tin là một thuộc tính của tập thực thể, ứng với một trường trong bảng dữ liệu tương ứng. Ví dụ: khách hàng Nguyễn Văn A có năm sinh là 1981, có số điện thoại là 8534... . Tập thực thể khách hàng sẽ có các thuộc tính “năm sinh”, “số điện thoại”. Một thuộc tính là một đặc tính mô tả hoặc đặc điểm quan tâm của một thực thể.

- Kiểu dữ liệu (Data type) của một thuộc tính xác định kiểu dữ liệu có thể lưu trữ được trong thuộc tính đó
- Phạm vi (Domain) của một thuộc tính xác định các giá trị mà thuộc tính đó có thể chứa một cách hợp lệ
- Giá trị mặc định (default value) của một thuộc tính là giá trị sẽ được ghi vào nếu không được xác định bởi người dùng
- Kiểu dữ liệu

Các kiểu dữ liệu logic điển hình cho các thuộc tính	
Kiểu dữ liệu logic	Ý nghĩa logic
NUMBER	Có thể là số thực hoặc số nguyên
TEXT	Một chuỗi ký tự, bao gồm cả các con số
MEMO	Tương tự như TEXT nhưng có kích thước không xác định. Một số hệ thống đòi hỏi khả năng này để lưu các văn bản dài trong bản ghi cơ sở dữ liệu
DATE	Ngày dưới bất kỳ định dạng nào
TIME	Giờ dưới bất kỳ định dạng nào
YES/NO	Một thuộc tính chỉ có thể nhận một trong hai giá trị bên
VALUE SET	Một tập hữu hạn các giá trị. Trong hầu hết các trường hợp, một lược đồ mã sẽ được xây dựng (ví dụ M=Nam giới, F=Nữ giới)
IMAGE	Các loại hình ảnh

Bảng 6-1: Các kiểu dữ liệu logic điển hình cho các thuộc tính

- Phạm vi dữ liệu

Các phạm vi logic điển hình cho các kiểu dữ liệu logic		
Kiểu dữ liệu	Phạm vi	Ví dụ
NUMBER	Đối với số nguyên, xác định phạm vi	{10-99}
	Đối với số thực, xác định phạm vi và độ chính xác	{1.000-799.999}

TEXT	Kích thước lớn nhất của thuộc tính Các giá trị thực tế thường là vô hạn; tuy nhiên, người dùng có thể xác định các hạn chế nào đó	Text(30)
DATE	Sự biến đổi trên các định dạng MMDDYYYY	MMDDYYYY MMYYYY
TIME	Đối với thời gian AM/PM: HHMMT Đối với thời gian 24 giờ: HHMM	HHMMT HHMM
YES/NO	{YES, NO}	{YES, NO} {ON, OFF}
VALUE SET	{giá trị#1, giá trị#2,...giá trị#n} {bảng các mã và ý nghĩa}	{M=Nam giới F=Nữ giới}

Bảng 6-2: Các phạm vi logic điển hình cho các kiểu dữ liệu logic

- Giá trị mặc định

Các giá trị mặc định chấp nhận được cho các thuộc tính		
Giá trị mặc định	Diễn giải	Ví dụ
Một giá trị hợp lệ từ phạm vi	Đối với một thể hiện của thuộc tính, nếu người dùng không xác định giá trị thì sử dụng giá trị này	0 1.00
NONE hoặc NULL	Đối với một thể hiện của thuộc tính, nếu người dùng không xác định giá trị thì để trống	NONE NULL
Required hay NOT NULL	Đối với một thể hiện của thuộc tính, đòi hỏi người dùng phải nhập vào một giá trị hợp lệ từ phạm vi. (Điều này được dùng khi không có giá trị nào trong phạm vi có thể làm giá trị mặc định nhưng lại nhất thiết phải có giá trị được nhập)	REQUIRED NOT NULL

Bảng 6-3: Các giá trị mặc định chấp nhận được cho các thuộc tính

- Có 3 loại thuộc tính:
 - Thuộc tính khóa: gồm một hoặc nhiều thuộc tính trong thực thể được dùng để gán cho mỗi thể hiện thực thể một cách tham khảo duy nhất. Ví dụ thuộc tính Mã sinh viên trong thực thể Sinh viên
 - Thuộc tính mô tả: là các thuộc tính dữ liệu mô tả về một đối tượng và không được chọn làm thuộc tính khóa. Ví dụ các thuộc tính Tên sinh viên, Địa chỉ...
 - Thuộc tính kết nối: là thuộc tính mà với thực thể này thì là thuộc tính mô tả nhưng với thực thể khác thì là thuộc tính khóa, nó đóng vai trò kết nối các thực thể có quan hệ với nhau

- Ví dụ về các loại thuộc tính:

TÊN THỰC THỂ		KHÁCH HÀNG
- <u>mã thực thể</u>		<u>Customer ID</u>
- thuộc tính 1		Cust_Name
- thuộc tính 2		Cust_Address
-		Cust_Phone
- thuộc tính n		

6.2.3. Mối quan hệ

- Một quan hệ tài liệu hoá một liên kết giữa một, hai hoặc nhiều thực thể. Nó phải có một cái tên (và có thể mang dữ liệu).
- **Quan hệ 1-1:**
 - Là mối quan hệ trong đó *một thực thể của tập thực thể này tương ứng với duy nhất một thực thể* của tập thực thể kia và ngược lại. Ví dụ, một thực thể hóa đơn hàng chỉ ứng với duy nhất một thực thể chi tiết hóa đơn mô tả nó.
 - Quan hệ 1-1 được biểu diễn bằng một mũi tên hai đầu hoặc là một đoạn thẳng...



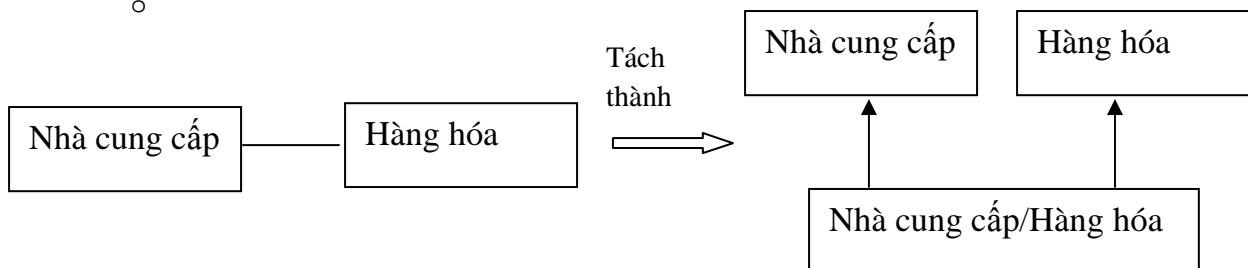
- Quan hệ này sẽ dẫn tới việc nhập chung hai tập thực thể thành một tập thực thể, tập thực thể mới phải bao gồm các thuộc tính của hai tập thực thể cũ.
- **Quan hệ 1-n:**
 - Là mối quan hệ mà trong đó *một thực thể của tập thực thể này có quan hệ với nhiều thực thể của tập thực thể kia*. Ví dụ, một khách hàng có thể đặt nhiều đơn hàng nên một thực thể khách hàng trong tập thực thể khách hàng có quan hệ với nhiều thực thể đơn hàng trong tập thực thể đơn hàng.
 - Quan hệ 1-nhiều được biểu diễn bằng một mũi tên 1 đầu hướng từ bên nhiều tới bên 1 hoặc là một đoạn thẳng với một đầu là trục ba hướng về bên nhiều...



- Quan hệ này đóng vai trò rất quan trọng thể hiện mối liên hệ giữa các thực thể trong mô hình. Ở đây, thuộc tính khóa của bên một sẽ là thuộc tính kết nối của bên nhiều.
- **Quan hệ n-n:**
 - Là mối quan hệ mà trong đó một thực thể của tập thực thể này có quan hệ với nhiều thực thể của tập thực thể kia và ngược lại. Ví dụ, một nhà cung cấp trong tập thực thể nhà cung cấp có thể cung cấp nhiều loại hàng trong tập thực thể Hàng hóa và ngược lại một loại hàng có thể được cung cấp bởi nhiều nhà cung cấp.
 - Quan hệ nhiều - nhiều được biểu diễn bằng một đoạn thẳng hoặc là một đoạn thẳng có trục ba ở cả hai đầu...



- Quan hệ này không thể hiện được mối quan hệ giữa hai thực thể cũng như không cho thấy điều gì về mặt nghiệp vụ, nên thường tách thành hai quan hệ 1- n bằng cách tạo một tập thực thể trung gian có quan hệ 1- n với cả hai tập thực thể đã có. Ví dụ với quan hệ n-n giữa nhà cung cấp và hàng hóa, ta sẽ tạo tập thực thể nhà cung cấp/hàng hóa có quan hệ là một nhà cung cấp gồm nhiều dòng nhà cung cấp/hàng hóa và một hàng hóa lại ứng với nhiều dòng nhà cung cấp/hàng hóa.



6.3. Xây dựng biểu đồ quan hệ thực thể

6.3.1. Các bước mô hình hóa dữ liệu logic

1. Mô hình dữ liệu ngữ cảnh

- Đề thiết lập phạm vi dự án

2. Mô hình dữ liệu dựa trên khoá

- Loại bỏ các quan hệ không cụ thể
- Thêm các thực thể có liên quan
- Bao gồm các khoá chính
- Xác định chính xác số yếu tố

3. Mô hình dữ liệu với thuộc tính đầy đủ

- Tất cả các thuộc tính còn lại
- Các tiêu chuẩn nhóm con

4. Mô hình dữ liệu được chuẩn hoá

- Thế nào là một mô hình dữ liệu tốt?

o Đơn giản

- Các thuộc tính dữ liệu mô tả bất cứ thực thể đã cho nào thì chỉ nên mô tả thực thể đó thôi
- Mỗi thuộc tính của một thể hiện của thực thể chỉ có thể có một giá trị

o Không dư thừa

- Mỗi thuộc tính dữ liệu, không phải là khoá ngoại, mô tả tối đa một thực thể
- Tìm cùng một thuộc tính được ghi lại nhiều lần dưới các tên khác nhau

o Linh động và dễ điều chỉnh cho những nhu cầu phát sinh trong tương lai

6.3.2. Trình tự xây dựng ERD

- Xác định các thực thể (Top-down)
- Xác định bậc của các quan hệ giữa các thực thể (rõ ràng ngữ cảnh)
- Hoàn thiện các quan hệ với các số yếu tố (rõ ràng ngữ cảnh)
- Xây dựng mô hình

Xác định các thực thể

- Cách 1:

Một tập thực thể có thể thuộc một trong 3 loại sau đây.

- Thông tin liên quan tới một giao dịch chủ yếu của hệ thống, ví dụ như hóa đơn bán hàng thuộc về quá trình bán hàng, đơn đặt hàng thuộc về quá trình mua hàng.
- Thông tin liên quan tới thuộc tính hoặc tài nguyên của hệ thống, ví dụ khách hàng, nhà cung cấp, vị trí kho hàng...
- Thông tin đã được khái quát dưới dạng thống kê liên quan tới lập kế hoạch hoặc quản lý như bảng chấm công, lịch trực...

Để nhận ra tập thực thể, phải đặt câu hỏi để ghi nhận thông tin về thực thể:

- Cái gì mà ta cần lưu thông tin về nó?
- Cái gì là cốt yếu trong hệ thống?
- Cái gì mà ta nói về nó trong hệ thống?
- Cái gì có thể dùng để phân biệt sự kiện của tập thực thể này với sự kiện của một tập thực thể khác?

- Cách 2:

Lấy một bản mô tả về hệ thống hiện tại hoặc cần có trong tương lai, xem xét các danh từ có trong đó xem có phải là thông tin cần lưu giữ không. Chú ý loại bỏ các từ đồng nghĩa. Lưu ý là có những danh từ mang tính mô tả nhưng lại không trở thành một tập thực thể, một số khác lại có thể là tập thực thể tiềm năng.

- Ví dụ: chương trình quản lý kho hàng
 - Theo dõi hàng tồn trong một kho nào đó tại một thời điểm nào đó.
 - Theo dõi chi tiết xuất nhập tồn của mỗi loại hàng hóa.
 - In chi tiết xuất nhập vật tư cho mỗi khách hàng.

Vậy các tập thực thể xác định được từ mô tả này chính là: vật tư, kho, hàng, khách hàng.

Xác định mối quan hệ

Quan hệ giữa các tập thực thể thường được diễn tả bởi các động từ, nó xác định sự tác động của các thực thể với nhau. Để xác định được các mối quan hệ giữa các tập thực thể, cần chú ý:

- Nếu cần phải lưu giữ thông tin về tập thực thể này trong tập thực thể kia thì sẽ có một quan hệ xuất hiện để tạo mối liên kết.
- Khi quan hệ giữa hai thực thể là gián tiếp thì ta không cần phải xây dựng mối quan hệ giữa chúng.
- **Ví dụ:**

Hệ thống thông tin quản lý tồn kho gồm 4 thực thể cơ bản sau:

Tên dữ liệu	Bao gồm
Danh mục tồn kho	+ Thẻ giá + Hồ sơ tồn kho + Báo cáo hàng tồn kho cũ + Danh sách đặt hàng bổ sung + Báo cáo mức tồn kho + Bản kiểm kê tồn kho
Danh mục hàng bán	+ Hoá đơn bán hàng + Báo cáo bán hàng
Danh mục nhà cung cấp	+ Hoá đơn nhà cung cấp + Đơn đặt hàng bổ sung
Danh mục bảo hành	+ Đơn bảo hành

Quan hệ giữa Danh mục tồn kho và Danh mục nhà cung cấp là quan hệ n-n vì:

- Mỗi loại sản phẩm trong danh mục tồn kho có thể có hơn một nhà cung cấp trong danh mục nhà cung cấp.
- Mỗi nhà cung cấp trong danh mục nhà cung cấp có thể cung cấp nhiều hơn một loại sản phẩm.

Chúng ta đưa ra tập thực thể mới là **Danh mục đặt hàng** có quan hệ 1 - n với cả hai tập thực thể ban đầu. Việc này nhằm tách 1 quan hệ n-n thành 2 quan hệ 1-n.

Tương tự, có quan hệ $n - n$ giữa danh mục đặt hàng và danh mục tồn kho. Chúng ta đưa ra tập thực thể **Chi tiết đặt hàng**.

Chúng ta đã thu thập được danh sách một số phần tử dữ liệu, cần xác định những thực thể còn thiếu. Bắt đầu với dữ liệu bán hàng, có một vài trường của hoá đơn bán mô tả một khách hàng. Vậy khách hàng là một đối tượng mà dữ liệu tập hợp vào đó. Do vậy **Danh mục khách hàng** là một tập thực thể.

Nhận xét rằng quan hệ giữa danh mục khách hàng và danh mục tồn kho là quan hệ $n-n$ vì:

- Một khách hàng có thể mua nhiều loại sản phẩm.
- Một loại sản phẩm có thể được bán cho nhiều khách hàng.

Với các thuộc tính còn lại trong dữ liệu bán hàng ta có **Danh mục bán hàng**, từ đó chúng ta có quan hệ $n-n$ giữa Danh mục bán hàng với Danh mục tồn kho vì:

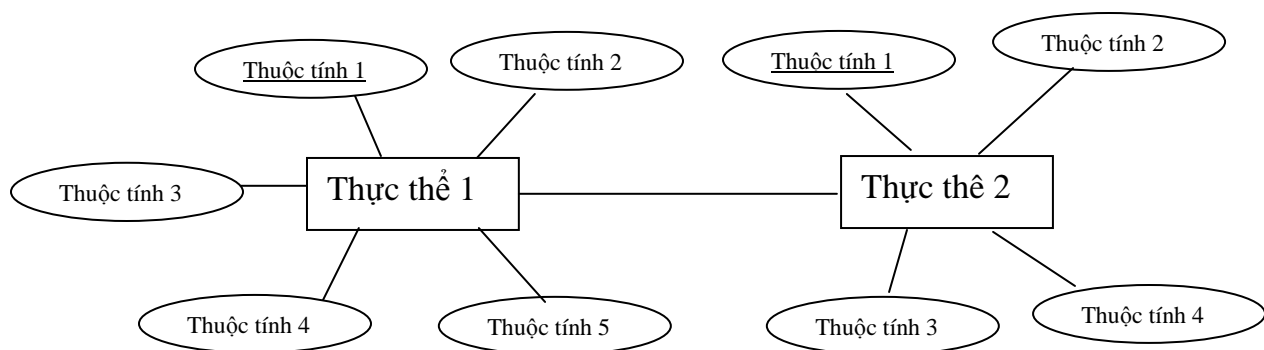
- Một giao dịch bán có thể có nhiều hơn một loại hàng.
- Một loại hàng có thể có trong nhiều giao dịch bán.

Do vậy, ta đưa ra tập thực thể mới là **Chi tiết bán hàng**

6.3.3. Quy tắc xây dựng ERD

- Mỗi thực thể phải có tên
- Mỗi thực thể phải có định danh
- Mỗi thể hiện không thể là một thực thể
- Mỗi quan hệ phải có tên (có thể mang hoặc không mang dữ liệu)
- Số yếu tố phải hợp lý (rõ ràng ngữ cảnh)

6.3.4. Các kiểu ký hiệu ERD khác



- Xác định các mối quan hệ giữa các thuộc tính của các thực thể.

6.4.1. Xác định thuộc tính

- Để xác định các thuộc tính của các thực thể, cần dựa vào những yếu tố sau:
 - Sự hiểu biết về hệ thống đang phân tích.
 - Quá trình phỏng vấn, trao đổi với người sử dụng.
 - Các báo cáo, biểu mẫu được dùng trong hệ thống hiện tại.
- Từ những thông tin thu thập được, ta sẽ lập danh sách các thuộc tính cho các thực thể đã được xác định trong ERD. Phân biệt các thuộc tính khóa bằng cách gạch dưới.

6.4.2. Phân tích và chuẩn hóa dữ liệu

- Phân tích dữ liệu là một quá trình chuẩn bị một mô hình dữ liệu cho việc cài đặt thành một cơ sở dữ liệu đơn giản, không dư thừa, mềm dẻo và dễ thích ứng. Kỹ thuật cụ thể được gọi là sự chuẩn hóa.
- Chuẩn hóa là một kỹ thuật tổ chức các thuộc tính dữ liệu sao cho chúng được nhóm thành các thực thể không dư thừa, ổn định, mềm dẻo và dễ thích ứng:

◦ Không có sự lặp lại các thuộc tính ở các bảng khác nhau, trừ thuộc tính khóa và thuộc tính kết nối

◦ Không chứa các thuộc tính có giá trị là kết quả tính được của các thuộc tính khác. Ví dụ, thuộc tính giá thành là kết quả của thuộc tính số lượng nhân với thuộc tính đơn giá nên cần phải loại bỏ.

◦ Không có vai trò giống nhau giữa các thực thể

- **Khái niệm phụ thuộc hàm:**
 - Phụ thuộc hàm đơn trị: từ 1 giá trị của khóa trong bảng, ta chỉ xác định được 1 giá trị cho các thuộc tính khác. VD: với mỗi mã khách hàng, chỉ có duy nhất một giá trị Họ tên, số điện thoại, địa chỉ...
 - Phụ thuộc hàm đa trị: 1 giá trị của khóa trong bảng lại ứng với nhiều giá trị của các thuộc tính khác. Ví dụ: ứng với một mã số học sinh lại có nhiều môn học khác nhau vì một học sinh có thể học nhiều môn học.
 - Như vậy, nếu có thuộc tính không phụ thuộc hàm vào khóa thì nó phải nằm trong một thực thể khác. Quá trình chuẩn hóa được thực hiện dựa trên khái niệm phụ thuộc hàm nêu trên.
- **Chuẩn hóa dạng 1:**

- Yêu cầu: các thuộc tính nào có thể xuất hiện nhiều lần với cùng một thực thể thì loại bỏ ra.
- Các thuộc tính bị loại ra sẽ cùng với thuộc tính khóa của thực thể ban đầu tạo thành một tập thực thể mới.
- **Chuẩn hóa dạng 2:**
 - Yêu cầu: tất cả các thuộc tính trong thực thể phải phụ thuộc hàm vào toàn bộ khóa.
 - Đối với các thực thể chỉ có một trường là khóa thì đương nhiên thỏa mãn dạng chuẩn 2.
 - Đối với các thực thể có khóa bao gồm 2 thuộc tính trở lên, nếu trong đó có những thuộc tính phụ thuộc hàm đơn trị vào một bộ phận của khóa thì tách các thuộc tính đó ra thành 1 thực thể mới với khóa là bộ phận khóa của thực thể ban đầu mà nó phụ thuộc hàm.
- **Chuẩn hóa dạng 3:**
 - Yêu cầu: tất cả các thuộc tính phải phụ thuộc đơn trị vào khóa và không phụ thuộc hàm đơn trị vào bất kỳ thuộc tính nào không phải là khóa trong thực thể.
 - Tách những thuộc tính phụ thuộc hàm đơn trị vào thuộc tính không phải là khóa, đưa chúng vào thực thể mới có khóa chính là thuộc tính mà nó phụ thuộc hàm.

Ví dụ 6-1: xét quá trình xây dựng các thuộc tính cho các tập thực thể dựa trên mẫu hóa đơn bán hàng của một công ty

Số HĐ:				
HÓA ĐƠN BÁN				
Ngày.....				
Họ tên khách hàng:.....Mã số khách hàng:.....				
Địa chỉ:.....				
Mã số mặt hàng	Tên hàng	Số lượng	Đơn giá	Thành tiền

- Quá trình chuẩn hóa diễn ra như sau:

Thuộc tính ban đầu chưa chuẩn hóa	Chuẩn hóa dạng 1	Chuẩn hóa dạng 2	Chuẩn hóa dạng 3
	1NF	2NF	3NF
Số hiệu đơn	Số hiệu đơn	Số hiệu đơn	Số hiệu đơn

Mã số khách hàng	Mã số khách hàng	Mã số khách hàng	Mã số khách hàng
Ngày đặt hàng	Ngày đặt hàng	Ngày đặt hàng	Ngày đặt hàng
Tên khách hàng	Tên khách hàng	Tên khách hàng	<u>Mã số khách hàng</u>
Địa chỉ	Địa chỉ	Địa chỉ	Tên khách hàng
Mã số mặt hàng	<u>Số hiệu đơn</u>	<u>Số hiệu đơn</u>	Địa chỉ
Tên mặt hàng	<u>Mã số mặt hàng</u>	<u>Mã số mặt hàng</u>	<u>Số hiệu đơn</u>
Số lượng	Tên mặt hàng	Số lượng	<u>Mã số mặt hàng</u>
Đơn giá	Số lượng	Đơn giá	Số lượng
	Đơn giá	<u>Mã số mặt hàng</u>	Đơn giá
		Tên mặt hàng	<u>Mã số mặt hàng</u>
			Tên mặt hàng

- Sau khi chuẩn hóa, thu được các thực thể sau:

Đơn hàng bán (Số hiệu đơn hàng, Mã số khách hàng, Ngày đặt hàng)

Khách hàng (Mã số khách hàng, Tên khách hàng, Địa chỉ khách hàng)

Dòng đơn hàng (Số hiệu đơn hàng, Mã số mặt hàng, Số lượng, Đơn giá)

Mặt hàng (Mã số mặt hàng, Tên mặt hàng)

- Kết hợp các tập thực thể chung.**

Do việc chuẩn hóa xuất phát từ nhiều tài liệu khác nhau nên có thể sau khi chuẩn hóa sẽ xuất hiện các thực thể giống nhau. Cần phải hợp nhất chúng thành một thực thể mà chứa đủ các thuộc tính. Rất có thể sau giai đoạn này thì thực thể thu được sẽ không còn ở dạng chuẩn 3 nên cần phải thực hiện chuẩn hóa lại các thực thể mới.

Ví dụ 6-2: Có 2 tập thực thể đơn đặt hàng được chuẩn hóa từ 2 tài liệu là đơn đặt hàng và tài liệu giao hàng như sau:

Đơn hàng (Số hiệu đơn hàng, Mã số khách hàng, Ngày đặt hàng)

Đơn đặt hàng (Số hiệu đơn hàng, Tình trạng đơn hàng, Địa chỉ giao hàng)

Sau khi kết hợp có:

Đơn hàng (Số hiệu đơn hàng, Mã số khách hàng, Ngày đặt hàng, Tình trạng đơn hàng, Địa chỉ giao hàng)

Thực thể mới không còn thỏa dạng chuẩn 3 vì địa chỉ giao nhận phụ thuộc hàm vào Mã số khách hàng là thuộc tính không phải là khóa của thực thể. Thực hiện chuẩn hóa, tách thuộc tính Địa chỉ giao nhận ra khỏi thực thể ta được thực thể mới:

Đơn hàng (Số hiệu đơn hàng, Mã số khách hàng, Ngày đặt hàng, Tình trạng đơn hàng)

6.4.3. Xác định các mối quan hệ

Theo ví dụ 6-1 và 6-2 sau khi chuẩn hóa, thu được các thực thể sau:

Đơn hàng bán (Số hiệu đơn hàng, Mã số khách hàng, Ngày đặt hàng)

Khách hàng (Mã số khách hàng, Tên khách hàng, Địa chỉ khách hàng)

Dòng đơn hàng (Số hiệu đơn hàng, Mã số mặt hàng, Số lượng, Đơn giá)

Mặt hàng (Mã số mặt hàng, Tên mặt hàng)

Giao nhận (Số hiệu giao nhận, Mã số khách hàng, Ngày giao)

Dòng giao hàng (Số hiệu giao nhận, Số hiệu đơn hàng, Mã số mặt hàng, Số lượng giao)

a. Ma trận thực thể/ khóa.

- Để xác định các mối quan hệ giữa các thực thể, ta cần lập bảng ma trận thực thể/ khóa. Trong đó, các cột liệt kê các tập thực thể, các hàng liệt kê các thuộc tính khóa của các thực thể.
- Ứng với mỗi ô giao giữa hàng và cột, nếu thuộc tính khóa có trong thực thể, ta đánh dấu X, nếu không là khóa của thực thể nhưng có xuất hiện trong đó thì đánh dấu O. Ví dụ:

Thực thể	Đơn hàng	Khách hàng	Dòng đơn hàng	Mặt hàng	Giao nhận	Dòng giao hàng
Thuộc tính khóa						
Số hiệu đơn hàng	X		X			X
Mã số khách hàng	O	X			X	

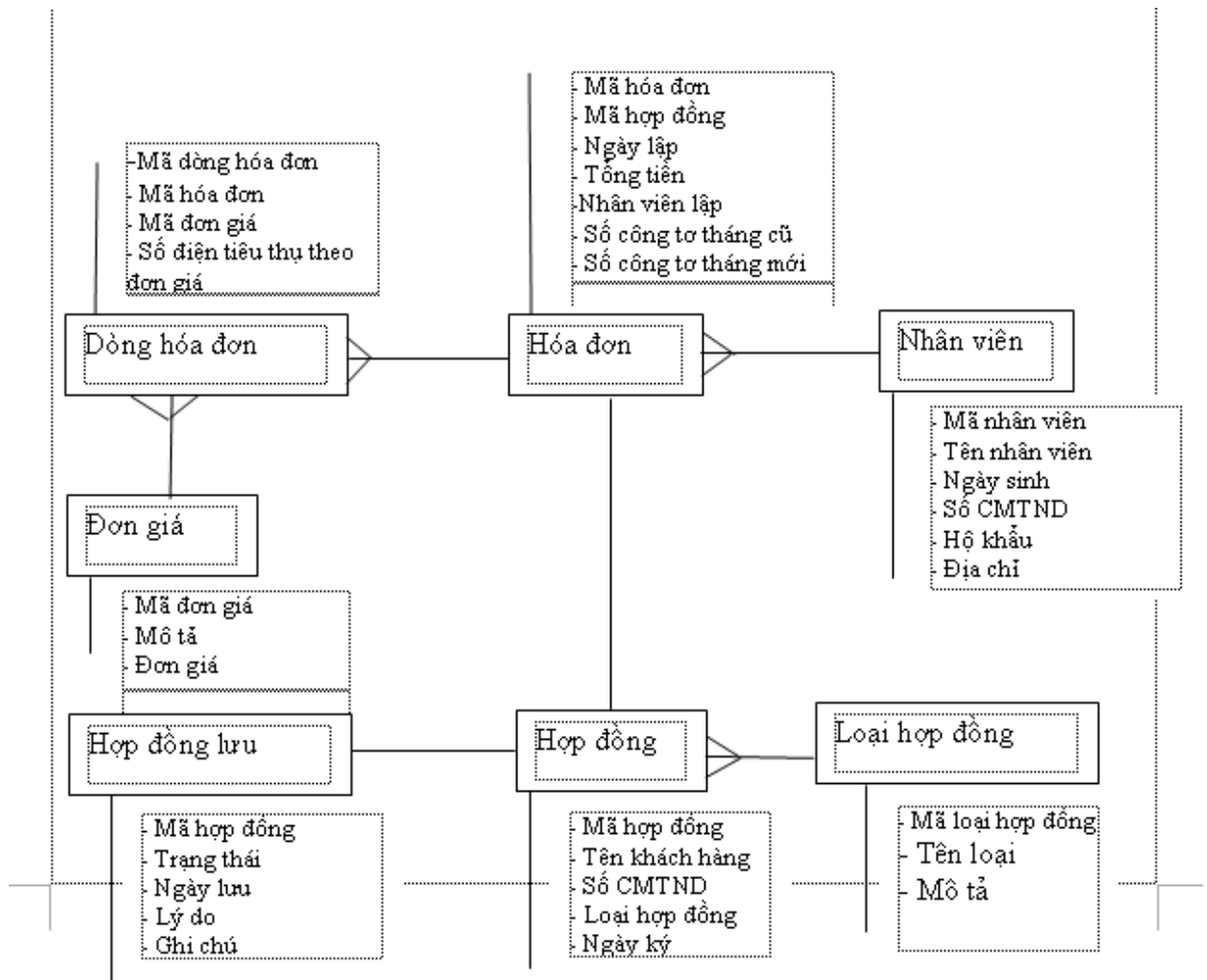
Mã số mặt hàng			X	X		X
Số hiệu giao nhận					X	X

b. Thiết lập các mối quan hệ.

Dựa theo bảng ma trận thực thể/khóa, ta xác định các mối quan hệ bằng cách bắt đầu từ cột thứ nhất, từ ô chứa khóa của nó, ta chiếu qua các ô kế tiếp trên cùng một hàng để xem nếu ô nào có chứa dấu X hoặc O thì ta có một liên kết giữa thực thể đang xét với thực thể có ô chứa dấu trên hàng đó.

6.4.4. Xây dựng mô hình RDM

- Sau khi đã thực hiện xong 3 bước trên, ta lập mô hình quan hệ.
- Tiếp theo, cần điều chỉnh để có được mô hình dữ liệu hoàn chỉnh:
 - So sánh ERD được xây dựng ban đầu với RDM để chỉnh lý những điểm khác biệt sao cho hai mô hình phải phản ánh chính xác lẫn nhau.
 - Trong một số trường hợp, nhà phân tích có thể đưa vào hoặc loại bỏ những quan hệ phụ để làm trong sáng mô hình.



Hình 6-5: Biểu đồ quan hệ dữ liệu của hệ thống bán điện

6.5. Từ điển dữ liệu

6.5.1. Khái niệm

Từ điển dữ liệu là bộ phận của tư liệu trong phân tích thiết kế, nó mô tả nội dung của các đối tượng theo định nghĩa có cấu trúc

Trong DFD các chức năng xử lý, kho dữ liệu, luồng dữ liệu chỉ mô tả ở mức khái quát thường là tập hợp các khoản mục riêng lẻ. Các khái quát này cần được mô tả chi tiết hoá hơn qua công cụ từ điển dữ liệu

Từ điển dữ liệu là sự liệt kê có tổ chức các phần tử dữ liệu thuộc hệ thống, liệt kê các mục từ chỉ tên gọi theo một thứ tự nào đó và giải thích các tên một cách chính xác

chặt chẽ ngắn gọn để cho cả người dùng và người phân tích hiểu chung cái vào, cái ra, cái luân chuyển.

6.5.2. Kí hiệu mô tả nội dung cho từ điển dữ liệu:

Kết cấu dữ liệu	Ký pháp	Ý nghĩa
Định nghĩa	=	Được tạo từ
Tuần tự	+	Và
Tuyển chọn	[]	Hoặc
Lặp	{ } n	Lặp n lần
Lựa chọn	()	Dữ liệu tùy chọn
Giải thích	* Lời chú thích *	Giới hạn chú thích

6.5.3. Ví dụ một từ điển dữ liệu:

- Xác định một từ hoá đơn:

Hoá đơn = Số HD + Ngày bán + Khách hàng + Hàng * n + Số lượng * n + Thành tiền * n + Tổng cộng +
KT trường + Người bán.

- Xác định thông tin về khách hàng

Khách hàng = Họ tên Khách + Tài khoản + Địa chỉ + Điện thoại

- Xác định thông tin về từng mặt hàng

Hàng = Mã hàng + Tên quy cách + Đơn vị tính + Đơn giá

- Họ tên khách cần được tách tên để thuận tiện đối với tên Tiếng Việt

Họ tên khách = Họ đệm + tên

Câu hỏi thảo luận C6

- 6.1. Phân biệt thuộc tính khóa và thuộc tính kết nối.
- 6.2. Quan hệ 1 - 1 xuất hiện trong các trường hợp nào? Hãy cho ví dụ.
- 6.3. Quan hệ nhiều - nhiều được xử lý như thế nào trong mô hình ERD?
- 6.4. Quan hệ 1 - nhiều thường xuất hiện trong những trường hợp nào? Hãy cho ví dụ.
- 6.5. Nêu ý nghĩa của số yếu tố trong ERD?
- 6.6. Phân biệt phụ thuộc hàm đơn trị và phụ thuộc hàm đa trị.
- 6.7. Phân tích mối quan hệ giữa các thực thể trong ERD của hệ thống quản lý kho ở hình 6-4.
- 6.8. Phân tích RDM của hệ thống quản lý bán điện trong hình 6-5.
- 6.9. Xây dựng RDM của hệ thống quản lý kho.

Chương 7. TỔNG QUAN VỀ THIẾT KẾ HỆ THỐNG

7.1. Các hướng tiếp cận thiết kế hệ thống

7.1.1. Các tiếp cận hướng mô hình

Thiết kế hướng mô hình (Model-driven) là một cách tiếp cận thiết kế hệ thống nhấn mạnh vào việc vẽ các mô hình hệ thống để tài liệu hóa các khía cạnh cài đặt và kỹ thuật của một hệ thống. Các mô hình thiết kế thường được dẫn xuất từ các mô hình logic được phát triển trước đó theo cách phân tích hướng mô hình. Cuối cùng thì các mô hình thiết kế hệ thống sẽ trở thành các bản thiết kế phục vụ cho việc xây dựng và cài đặt hệ thống mới.

Trong tiếp cận hướng mô hình có 3 kỹ thuật là thiết kế hướng cấu trúc, kỹ thuật thông tin và thiết kế hướng đối tượng. Ngày nay, các tiếp cận hướng mô hình thường được củng cố nhờ vào việc sử dụng các công cụ tự động hóa. Các công cụ thường dùng:

- Công cụ đi kèm bộ công cụ lập trình: Oracle Designer
- Các công cụ đơn giản: MS.Word, MS.Visio, Smartdraw...
- Các công cụ chuyên dụng: Rational Rose, Rational XDE for platforms...
- **Thiết kế hướng cấu trúc hiện đại (Modern Structured Design):**
 - Là kỹ thuật phân rã chức năng hệ thống ra thành nhiều phần, mỗi thành phần lại được thiết kế chi tiết hơn ở các bước sau. Thiết kế hướng cấu trúc còn được gọi là thiết kế chương trình từ tổng quan đến chi tiết (top- down).
 - Mỗi modul ở mức thấp nhất chỉ thực hiện một phần việc nhất định, ít liên quan đến công việc của các modul khác.
 - Thường được sử dụng vì đơn giản, dễ hiểu, thuận tiện trong triển khai và nâng cấp.
 - Mô hình phần mềm được dẫn xuất từ thiết kế hướng cấu trúc được gọi là biểu đồ cấu trúc (structure chart). Biểu đồ này được xây dựng từ các luồng dữ liệu trong chương trình. Thiết kế hướng cấu trúc được thực hiện trong giai đoạn phân tích hệ thống. ***Tuy nhiên, nó không bao trùm mọi khía cạnh của việc thiết kế, như thiết kế đầu vào/đầu ra hay cơ sở dữ liệu.***
- Các ký hiệu trong biểu đồ cấu trúc:
 - Modul: được biểu diễn bằng hình chữ nhật có nhãn là tên của modul.
 - Dữ liệu được chuyển giao giữa các modul: biểu diễn bởi mũi tên có đầu tròn rỗng.
 - Thông tin điều khiển: biểu diễn bằng mũi tên với đầu tròn đặc.
- **Kỹ thuật thông tin (Information Engineering):**
 - Là cách tiếp cận hướng mô hình và lấy dữ liệu làm trung tâm nhưng , chú trọng đến việc tổ chức các thông tin:
 - nội dung thông tin

- quan hệ giữa các thành phần thông tin.
- Công cụ chủ yếu là sơ đồ mô hình dữ liệu
- Được sử dụng khi thiết kế chương trình dựa trên mô hình cơ sở dữ liệu quan hệ.
- **Thiết kế hướng đối tượng (Object Oriented Design):**
 - Sử dụng cách tiếp cận theo tư duy hướng đối tượng - phân biệt rõ ràng hai yếu tố:
 - dữ liệu (thuộc tính)
 - chức năng (hành vi) và các mối tương tác (sự kiện)
 - Là bước tiếp theo của Phân tích hệ thống hướng đối tượng, do đó đòi hỏi những kết quả từ bước trước: định nghĩa đối tượng, thuộc tính, hành vi, sự kiện...

7.1.2. Phát triển ứng dụng nhanh

Kỹ thuật xây dựng ứng dụng nhanh chóng bằng cách phối hợp sử dụng nhiều kỹ thuật:

- Tổ hợp thông tin
- Làm bản mẫu
- **Làm bản mẫu (Prototyping)**
 - Bản mẫu là một chương trình nhỏ, chưa hoàn chỉnh nhưng đủ để cho người xem hình dung về chức năng, hoạt động của chương trình cần thực hiện:
 - Làm bản mẫu là phương pháp mô hình hoá trên mã nguồn chứ không trên bản vẽ
 - Thuận tiện để làm bản demo cho người dùng cuối xem (không đòi hỏi phải hiểu những ngôn ngữ mô hình hoá)
 - Sớm phát hiện những sai khác về nghiệp vụ
 - Chỉ phù hợp với các dự án nhỏ, ít phức tạp
- **Kỹ thuật phát triển ứng dụng kết hợp (Joint Application Development):** phát triển ứng dụng bằng cách gộp chung hai giai đoạn phân tích và thiết kế. Nhấn mạnh sự tham gia đồng thời của nhà phân tích, thiết kế, người dùng cuối, chuyên gia hệ thống trong quá trình xây dựng. Thường được dùng phát triển các ứng dụng nhỏ trong thời gian ngắn.

7.2. Các công việc cụ thể trong giai đoạn thiết kế

- **Thiết kế kiến trúc ứng dụng**
 - Lựa chọn công nghệ sử dụng cho dự án, Đưa ra mô hình vật lý của hệ thống.
 - Thiết kế cơ sở dữ liệu
 - Đưa ra mô hình dữ liệu.

- Lựa chọn hệ quản trị CSDL và tối ưu hoá mô hình dữ liệu theo hệ quản trị đã lựa chọn
- Thiết kế giao diện hệ thống: đầu ra, đầu vào, giao diện người dùng, báo cáo...
- Đưa ra các đặc tả hệ thống cho lập trình viên

Một số gợi ý cho một thiết kế tốt

Mục tiêu của một thiết kế tốt là tạo ra một hệ thống dễ đọc, dễ lập trình và dễ bảo trì. Các thiết kế hệ thống là kết quả thực hiện các hướng dẫn đưa ra (được xem là tốt). Vì vậy, các hệ thống cần đáp ứng mục tiêu tổng quát của thiết kế hệ thống.

- Hệ thống cần được môđun hóa, tức là nó cần được tổ chức thành một hệ phân cấp với một số đơn vị nhỏ hơn.
- Mỗi môđun cần kiểm soát được các chức năng của một số hợp lý các môđun con ở mức thấp hơn.
- Các môđun cần độc lập tương đối với nhau sao cho không một chức năng của môđun nào có thể ảnh hưởng đến sự làm việc bên trong của các môđun khác, tức là tổng số các truyền thông giữa các môđun phải đạt tối thiểu.
- Mỗi môđun cần có kích thước hợp lý
- Cần làm mịn đến mức sao cho mỗi môđun chỉ thực hiện một và chỉ một chức năng.
- Mã hóa trong một môđun cần khái quát tới mức sao cho mỗi môđun có thể sử dụng được nhiều lần trong hệ thống.

Câu hỏi thảo luận C7

- 7.1. Đặc điểm của thiết kế hướng cấu trúc hiện đại là gì?
- 7.2. Nêu ý nghĩa của việc làm bản mẫu.
- 7.3. Nêu các công việc cụ thể trong giai đoạn thiết kế.

Chương 8. KIẾN TRÚC ỨNG DỤNG VÀ VIỆC MÔ HÌNH HÓA

8.1. Kiến trúc ứng dụng

Kiến trúc hệ thống thông tin (KTHT) là một đặc tả về mặt công nghệ của một hệ thống thông tin. KTHT dùng làm phương tiện để:

- Trao đổi về đặc tính của hệ thống (tập trung hay phân tán, CSDL, tính tích hợp, giao diện hệ thống...).
- Cơ sở để triển khai hệ thống theo thiết kế.
- Cơ sở để bảo trì hệ thống sau này.

8.2. Biểu đồ luồng dữ liệu vật lý

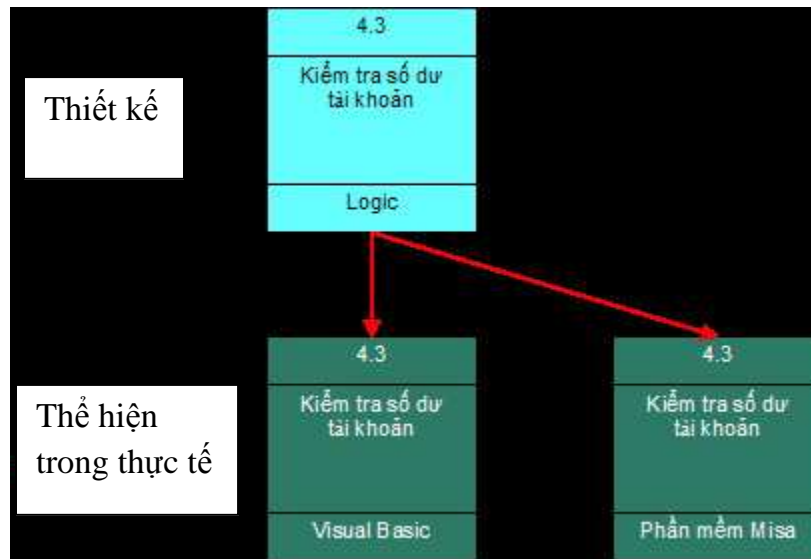
Biểu đồ luồng dữ liệu vật lý (Physical Data Flow Diagram - PDFD) là mô hình chức năng dùng để mô hình hoá kiến trúc hệ thống. PDFD biểu diễn các thuộc tính của từng thành phần trong KTHT cũng như mô tả mối quan hệ, tương tác giữa các thành phần đó. Dưới đây sẽ giới thiệu cách PDFD diễn tả từng đối tượng trong KTHT.

8.2.1. Chức năng vật lý

- Chức năng vật lý là nơi thực hiện các công việc của hệ thống, đó có thể là con người, máy tính cá nhân, server, máy tính cầm tay...
- Mỗi hệ thống cần một hoặc một số chức năng vật lý
- PDFD giúp chúng ta thấy rõ: thông tin được xử lý bởi con người hay máy móc, thông tin được xử lý bởi công nghệ nào...
- Ký hiệu

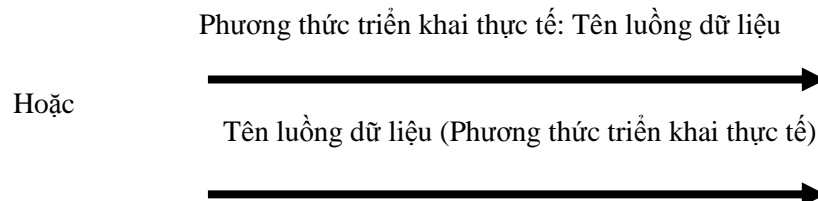
ID (tùy chọn)
Động từ + Danh từ/ Cụm câu
Phương thức cài đặt

- Ví dụ PDFD biểu diễn một chức năng vật lý



8.2.2. Luồng dữ liệu vật lý

- Luồng dữ liệu vật lý:
 - Mô tả các luồng dữ liệu đi luân chuyển trong hệ thống
 - Các lệnh tương tác với CSDL: tạo, đọc, cập nhật, xoá các đối tượng csdl
 - Nhập/xuất các phần tử dữ liệu giữa các thành phần trong mạng
 - Ký hiệu:



8.2.3. Kho dữ liệu vật lý

- Các kho dữ liệu vật lý dùng để mô tả
 - Một cơ sở dữ liệu
 - Một bảng trong cơ sở dữ liệu
 - Một file máy tính
 - File tạm
 - Một phương tiện lưu trữ dự phòng
 - Một dạng lưu trữ dữ liệu phi máy tính (mã vạch, RFID, thẻ từ...)
- Ký hiệu

ID	Phương thức triển khai: Tên kho dữ liệu
----	---

Hoặc

D	Phương thức triển khai: Tên kho dữ liệu
---	---

8.3. Kiến trúc công nghệ thông tin

Kiến trúc công nghệ thông tin (Information technology architecture) là một chủ đề phức tạp. Trong mục này, chúng tôi chỉ tóm tắt những xu thế công nghệ thông tin hiện đại có tác động tới các quyết định trong giai đoạn thiết kế.

8.3.1. Hệ thống phân tán

8.3.1. Hệ thống phân tán

Hệ phân tán (Distributed system) là hệ thống trong đó các thành phần phân tán giữa những địa điểm, mạng, máy tính khác nhau: tính toán lưới (grid-computing, mạng máy tính dựa trên PC...). Đối lập với hệ phân tán là **hệ tập trung** (Centralized system) là hệ thống trong đó các thành phần, các tác vụ xử lý tập trung tại một nơi (Mainframe). Các hệ thống hiện đại là các hệ phân tán, nó giúp phân phối dữ liệu và các dịch vụ đến gần người dùng cuối hơn, cắt giảm sự phức tạp và chi phí đầu tư, bảo trì. Có 3 loại kiến trúc hệ thống phân tán:

- **Kiến trúc máy chủ tệp (File server architecture)**
 - Là một mạng cục bộ (LAN) trong đó có một máy chủ chứa dữ liệu của một hệ thống thông tin. Mạng LAN là mạng nội bộ kết nối các máy tính(PC, Server, PDA...) trong một phạm vi hẹp (văn phòng, toà nhà...). Mạng LAN giúp tổng hợp năng lực các máy tính đơn lẻ trong mạng khi cho phép bất kỳ máy nào cũng có thể là máy chủ, bất kể máy nào cũng có thể là máy khách.
 - Kiến trúc này cho phép nhiều máy tính cá nhân và máy trạm được kết nối để chia sẻ dữ liệu và giao tiếp với nhau.
- **Kiến trúc khách/chủ (Client/Server architecture)**
 - Là kiến trúc trong đó có một hay nhiều máy tính đóng vai trò máy chủ cung cấp các dịch vụ, dữ liệu cho một hay nhiều máy khách.
 - Máy chủ cơ sở dữ liệu (Database server): là máy chủ logic lưu trữ một hay nhiều cơ sở dữ liệu đồng thời cung cấp một hệ thống các câu lệnh cho phép thao tác với những cơ sở dữ liệu nói trên.
 - Máy chủ ứng dụng (Application server): là máy chủ logic lưu trữ phần xử lý logic của một hay nhiều ứng dụng, cho phép các máy khách truy nhập vào để thực thi ứng dụng.

- Máy chủ nhắn tin hoặc phần mềm nhóm (Message hoặc Groupware server): là máy chủ logic cung cấp các dịch vụ như email, lịch làm việc, các chức năng hỗ trợ làm việc nhóm.
- Máy chủ web (Webserver): là máy chủ logic lưu trữ và vận hành các website trên mạng internet hoặc intranet.
- **Kiến trúc tính toán dựa trên Internet (Internet-Based computing architecture)**
 - Là một dạng khác của kiến trúc phân tán đang góp phần định hình lại ý tưởng thiết kế của các nhà phân tích hệ thống và chuyên gia thông tin.
 - Một hệ thống tính toán mạng là hệ thống trong đó các hệ thống thông tin đều chạy trên trình duyệt (ví dụ như hệ thống tài chính, hệ thống quản lý nhân sự...), lấy dữ liệu từ máy chủ web.

8.3.2. Kiến trúc dữ liệu

Cơ sở dữ liệu quan hệ lưu trữ dữ liệu dưới dạng bảng. Mỗi bảng bao gồm nhiều cột (giống các trường trong cơ sở dữ liệu dựa trên file), giao giữa các dòng và cột là các bản ghi (tương tự khái niệm bản ghi trên cơ sở dữ liệu file). Cơ sở dữ liệu quan hệ có một cơ sở toán học vững chắc và được dùng làm cơ sở dữ liệu của hầu hết các hệ thống hiện nay.

Cơ sở dữ liệu quan hệ phân tán là cơ sở dữ liệu quan hệ trong đó một hay nhiều bảng được nhân rộng và phân tán trên nhiều máy chủ cơ sở dữ liệu ở các nơi khác nhau.

Hệ quản trị cơ sở dữ liệu (HQTCSDL) là hệ thống quản lý việc lưu trữ, truy vấn, phân quyền truy nhập một hay nhiều cơ sở dữ liệu. HQTCSDL phân tán là một HQTCSDL làm thêm chức năng quản lý sự đồng bộ, kiểm soát truy nhập đối với các bảng dữ liệu phân tán. Có 2 kỹ thuật:

- Data partitioning: phân mảnh và phân tán một hay nhiều trường dữ liệu giữa các server mà không có hoặc có rất ít sự trùng lặp.
- Data Replication: không phân mảnh mà nhân bội một hay nhiều trường rồi phân tán giữa các server.

8.3.3. Kiến trúc giao diện

Là các kênh giao tiếp giữa các trung tâm xử lý trong hệ thống hay giữa các hệ thống máy tính với nhau. Các kiểu giao tiếp: dữ liệu vào ra trực tuyến/ theo bố, nhập liệu không cần bàn phím (mã vạch, thẻ từ, RFID), nhập liệu bằng bút cảm ứng, dữ liệu EDI, dữ liệu có được thông qua nhận dạng (vân tay, scan...), thông qua middleware...

Chương 9. THIẾT KẾ CƠ SỞ DỮ LIỆU

9.1. Các phương thức lưu trữ dữ liệu

Có hai phương thức lưu trữ dữ liệu phổ biến:

- File
- Cơ sở dữ liệu

9.1.1. File

Là một tập hợp của các bản ghi tương tự nhau. Các file không có liên quan với nhau trừ khi được liên kết trong code của chương trình ngoài

- Ưu điểm:
 - Dễ dàng thiết kế nếu chỉ dùng cho một ứng dụng
 - Tối ưu về hiệu năng nếu chỉ dùng cho một ứng dụng
- Nhược điểm:
 - Khó thích ứng hoặc khó dùng chung giữa nhiều ứng dụng
 - Hay bị dư thừa dữ liệu (cùng một thông tin lại được lưu trữ trên nhiều file khác nhau)

9.1.2. Cơ sở dữ liệu

Là một tập hợp của nhiều files (bảng) có quan hệ với nhau. Bản ghi của một file (hay bảng) có thể có mối quan hệ vật lý với một hay nhiều bản ghi ở các file (hay bảng) khác.

- Ưu điểm:
 - Tách biệt dữ liệu khỏi logic chương trình do đó tăng tính thích ứng, khả chuyển của chương trình.
 - Kiểm soát được quy mô, độ lớn của dữ liệu
 - Tối ưu trong việc chia sẻ dùng chung giữa nhiều ứng dụng
 - Giảm thiểu dư thừa dữ liệu
- Nhược điểm:
 - Phức tạp hơn công nghệ file rất nhiều
 - Ở khía cạnh nào đó truy xuất cơ sở dữ liệu thường chậm hơn so với truy xuất file
 - Cần tuân thủ nhiều nguyên tắc khi thiết kế để có thể khai thác được lợi ích của cơ sở dữ liệu quan hệ
 - Cần có chuyên gia sử dụng hệ quản trị cơ sở dữ liệu

9.2. Kiến trúc dữ liệu

- Kiến trúc dữ liệu mô tả cách thức:
 - sử dụng file/cơ sở dữ liệu để lưu trữ dữ liệu
 - công nghệ file/cơ sở dữ liệu được lựa chọn sử dụng
 - cơ cấu quản lý được thiết lập để quản lý các nguồn dữ liệu
- Thông thường dữ liệu được lưu trữ đồng thời bởi nhiều cách thức, phương tiện:
 - Các files,
 - Cơ sở dữ liệu cá nhân, cơ sở dữ liệu chung của nhóm, cơ sở dữ liệu giao dịch,
 - Nhà kho dữ liệu (tổng hợp các nguồn)...
- Hệ quản trị CSDL:
 - Là một phần mềm dùng để quản lý việc tạo, truy nhập, kiểm soát, quản lý các đối tượng dữ liệu của một hay nhiều cơ sở dữ liệu.
 - Phần nền tảng của một HQTCSĐL là một bộ máy dữ liệu - data engine
 - Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL) là một phần của bộ máy dùng để định nghĩa các bảng, trường, quan hệ
 - Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML) dùng để thêm, sửa, xoá và di chuyển giữa các trường trong cơ sở dữ liệu

9.3. Triển khai mô hình dữ liệu logic dựa trên một cơ sở dữ liệu quan hệ

9.3.1. Cơ sở dữ liệu quan hệ

Là cơ sở dữ liệu lưu trữ và quản lý dữ liệu trong những bảng 2 chiều. Các bảng này có thể có quan hệ với nhau thông qua các trường khoá

- Đặc thù của cơ sở dữ liệu quan hệ:
 - Mô hình dữ liệu vật lý (Schema)
 - DDL và DML được thể hiện bởi ngôn ngữ SQL
 - Triggers là các chương trình được nhúng cùng cơ sở dữ liệu và tự động thực thi khi cơ sở dữ liệu được cập nhật
 - Thủ tục thường trú (Stored procedure) là chương trình được nhúng cùng cơ sở dữ liệu và thực thi từ câu lệnh của ứng dụng

9.3.2. Mô hình hoá dữ liệu

- Một mô hình dữ liệu tốt là mô hình trong đó:
 - Mỗi thuộc tính mô tả một và chỉ một thực thể
 - Mỗi thuộc tính chỉ tồn tại ở duy nhất một thực thể (trừ thuộc tính khoá ngoại)
- Để có được một mô hình dữ liệu tốt, ta tiến hành các bước chuẩn hoá (xem thêm phần phân tích hệ thống)
- Chuẩn hoá dữ liệu - Một thực thể logic hay một bảng vật lý được gọi là:

- Ở dạng chuẩn thứ nhất nếu không có thuộc tính (trường) nào có hai giá trị trong cùng một thể hiện
- Ở dạng chuẩn thứ hai nếu nó đã ở dạng chuẩn thứ nhất và giá trị các trường không phải là khoá chính hoàn toàn phụ thuộc vào khoá chính.
- Ở dạng chuẩn thứ ba nếu nó đã ở dạng chuẩn thứ hai và giá trị các trường không phải khoá chính không phụ thuộc các trường không phải khoá chính khác
- Các bước tạo mô hình dữ liệu vật lý
 - Xem lại mô hình dữ liệu logic
 - Tạo bảng cho mỗi thực thể
 - Tạo trường cho mỗi thuộc tính
 - Tạo chỉ mục (index) cho mỗi khoá
 - Thiết kế khoá ngoại cho các quan hệ
 - Định nghĩa kích thước/kiểu dữ liệu, thuộc tính null, giá trị mặc định
 - Đánh giá và thiết lập các ràng buộc
 - *Chú ý các công nghệ khác nhau cho các kiểu dữ liệu khác nhau*
- Tạo câu lệnh SQL
 - Tuỳ vào từng hệ quản trị CSDL, tạo CSDL tương ứng trên ngôn ngữ DDL

Câu hỏi thảo luận C9

9.1. Nêu ưu nhược điểm của phương thức lưu trữ dữ liệu dạng file.

9.2. Nêu ưu nhược điểm của phương thức lưu trữ dữ liệu sử dụng cơ sở dữ liệu.

Chương 10. THIẾT KẾ ĐẦU VÀO

10.1. Tổng quan về thiết kế đầu vào

10.1.1. Các khái niệm

Có thể phân loại các phương thức nhập liệu theo hai đặc trưng: (1) cách thức dữ liệu được thu thập, đưa vào và xử lý và (2) phương pháp và công nghệ được dùng để thu thập và nhập dữ liệu.

- Thu thập dữ liệu (data capture): nhận dạng và tạo dữ liệu mới từ nguồn tạo tin
- Nhập liệu (data entry): chuyển dữ liệu từ nguồn tạo tin vào máy tính
- Xử lý dữ liệu (data processing): là quá trình biến đổi trực tiếp trên dữ liệu trước khi đưa nó về dạng máy tính có thể đọc được. Xử lý bó là thu thập 1 khối lượng dữ liệu và xử lý đồng thời cả bó. Xử lý trực tuyến là xử lý ngay lập tức dữ liệu vừa thu thập được.

10.1.2. Các phương thức nhập liệu

- Bàn phím
- Chuột
- Màn hình cảm ứng (màn hình tương tác)
- Nhận dạng âm thanh, tiếng nói
- Tự động nhập liệu: mã vạch, nhận dạng quang học, mực từ, thẻ từ, thẻ thông minh, sinh trắc học...

10.1.3. Các nguyên tắc thiết kế đầu vào

- Nên tuân theo những nguyên tắc dưới đây khi thiết kế phương thức nhập liệu:
 - Không nên nhập những dữ liệu có thể tính toán được từ những dữ liệu khác. Ví dụ: Số lượng x Đơn giá = Thành tiền
 - Không nhập những dữ liệu có thể lưu trong máy tính như những hằng số.
 - Sử dụng mã lấy từ cơ sở dữ liệu đối với những thuộc tính phù hợp.
 - Sử dụng các chỉ dẫn nhập liệu khi thiết kế các form nhập liệu (tooltip).
 - Giảm thiểu số lượng ký tự gõ vào để tránh gây sai sót. Thay vào đó, cố gắng dùng các hộp check chọn càng nhiều càng tốt.
 - Dữ liệu nhập vào theo trình tự từ trên xuống dưới, trái qua phải.

10.1.4. Kiểm soát nhập liệu

Việc kiểm soát dữ liệu đầu vào rất cần thiết trong tất cả các hệ thống ứng dụng trên máy tính. Các điều khiển đầu vào đảm bảo rằng dữ liệu đầu vào là chính xác và hệ thống được bảo vệ khỏi các lỗi vô ý hoặc hữu ý.

- Số lượng đầu vào cần phải được theo dõi, đặc biệt là trong trường hợp nhập dữ liệu theo bố:
 - Lưu mã số giao dịch cho bố các dữ liệu nhập liệu theo bố.
 - Ghi các log file cho các dữ liệu được nhập trực tuyến

Phải kiểm soát tính đúng đắn của dữ liệu nhập vào. Phải làm các kiểm tra về: trùng lặp thực thể, kiểu dữ liệu, định dạng, tính ràng buộc với các dữ liệu khác. Ví dụ: Khi nhập liệu thành phố và quốc gia cho một hồ sơ nhân sự, nếu đã chọn quốc gia là Việt Nam thì chỉ cho phép chọn thành phố là Hà Nội, Huế hoặc các thành phố khác ở Việt Nam ... chứ không cho phép chọn thành phố thuộc quốc gia khác như Tokyo chẳng hạn.

10.2. Các điều khiển giao diện cho thiết kế đầu vào

10.2.1. Một số điều khiển phổ biến

- Hộp văn bản (Text box): chứa một hộp hình chữ nhật kèm theo tên, cho phép nhập dữ liệu vào.
- Nút chọn loại trừ (Radio button): chứa một hình tròn nhỏ kèm theo một đoạn văn bản mô tả tương ứng với giá trị lựa chọn. Trong một nhóm các nút này thì chỉ cho phép chọn một nút mà thôi.
- Hộp chọn kiểm tra (Check box): chứa một hộp hình vuông kèm theo đoạn văn bản mô tả trường dữ liệu vào, người dùng sẽ chọn giá trị Yes/No. Trong một nhóm các hộp chọn thì có thể chọn nhiều hộp.
- Hộp danh sách (List box): là một hình chữ nhật chứa một hoặc nhiều dòng dữ liệu.
- Danh sách thả (Drop down list): chứa hộp chọn hình chữ nhật và một nút bên cạnh. Khi nhấn vào nút đó thì danh sách sẽ được thả xuống.
- Hộp thả kết hợp (Combination box): cũng là một danh sách thả nhưng cho phép người dùng nhập thêm dữ liệu ngoài những dữ liệu có sẵn trong đó.
- Nút lệnh (Button): các nút lệnh không phải là điều khiển vào. Chúng không dành cho việc lựa chọn dữ liệu vào. Mục đích của chúng là cho phép người dùng xác nhận rằng tất cả các dữ liệu cần được xử lý hay hủy bỏ một giao dịch hoặc cần gọi chức năng trợ giúp... Tóm lại, nút lệnh đóng vai trò gọi tới một chức năng nào đó.

10.2.2. Một số điều khiển cao cấp

- Hộp lịch thả (Drop down calendar): là một ô dữ liệu có chứa một nút mũi tên. Khi nhấn chuột vào đó thì hộp lịch được thả xuống để chọn ngày.
- Điều khiển hiệu chỉnh trượt (Slider edit control): cho phép lựa chọn giá trị bằng cách trượt con trỏ.
- Điều khiển hiệu chỉnh mặt nạ (Masked edit control): điều khiển này tạo ra định dạng để buộc dữ liệu nhập vào phải tuân theo.
- Hộp danh sách chọn (Check list box): điều khiển này được dùng để kết hợp nhiều hộp chọn kiểm tra.

- Hộp cây chọn (Check tree box): điều khiển này được dùng để biểu diễn các lựa chọn dữ liệu dưới dạng cây phân cấp.

10.3. Quy trình thiết kế đầu vào

- Xác định các dữ liệu đầu vào của hệ thống và các yêu cầu nhập liệu logic.
- Lựa chọn các điều khiển thích hợp.
- Thiết kế, lập cơ chế kiểm soát nhập liệu, lưu vết
- Nếu cần thiết, lập hồ sơ đặc tả đầu vào.

Câu hỏi thảo luận C10

10.1. Nêu tóm tắt các phương thức nhập liệu.

10.2. Nêu các nguyên tắc thiết kế nhập liệu

10.3. Thiết kế một số trường hợp kiểm soát nhập liệu trong các bài toán thông dụng: hồ sơ nhân sự, điểm học sinh – sinh viên,...

Chương 11. THIẾT KẾ ĐẦU RA

11.1. Tổng quan về thiết kế đầu ra

11.1.1. Phân loại đầu ra

Một cách để phân loại đầu ra là dựa vào hình thức phân phối chúng trong hay ngoài tổ chức và đối tượng người sẽ đọc và sử dụng chúng. Hình thức đầu ra chủ yếu là dưới dạng các báo cáo.

- Báo cáo nội bộ: là các báo cáo được cung cấp cho người dùng hệ thống trong tổ chức
 - Báo cáo chi tiết: thông tin trực tiếp truy xuất từ dữ liệu hệ thống, ví dụ: danh sách khách hàng trong hệ thống quản lý khách hàng, danh sách thí sinh trong hệ thống tuyển sinh
 - Báo cáo tóm lược: thông tin sau khi truy xuất đã được sắp xếp theo thứ tự thuận tiện cho người dùng quan sát, đôi khi kết quả được thể hiện dưới dạng đồ họa, ví dụ: khách hàng theo khu vực trong bài toán quản lý khách hàng, danh sách thí sinh theo từng ban tuyển sinh trong bài toán tuyển sinh
 - Báo cáo ngoại lệ: thông tin cảnh báo, đột xuất theo sự kiện thay đổi về chất lượng, điều kiện của hệ thống.
- Báo cáo bên ngoài là các báo cáo cung cấp cho khách hàng, nhà cung cấp, cơ qua pháp luật...
- Báo cáo quay vòng là các loại báo cáo bên ngoài sau đó lại trở về hệ thống như là một phương thức thu thập dữ liệu, chẳng hạn bản điều tra, hoá đơn...

11.1.2. Các phương thức cài đặt đầu ra

- In ra trên giấy
- Hiển thị trên màn hình, trên trang web
- Xuất dưới dạng đa phương tiện
- Gửi thư trực tiếp
- Tạo các đường liên kết
- ...

11.2. Cách thức thiết kế đầu ra

11.2.1 Các nguyên tắc thiết kế đầu ra

- Báo cáo phải đơn giản, dễ hiểu, dễ giải thích:
 - Bao gồm tiêu đề
 - Ghi rõ ngày giờ phát hành
 - Có các phần ghi thông tin chung

- Thông tin phải được thể hiện ở dạng người dùng bình thường không được tùy ý sửa chữa.
- Thông tin hiển thị phải hài hoà giữa các trang
- Cung cấp cách di chuyển giữa các ô thật sự đơn giản
- Thời gian xuất báo cáo phải được kiểm soát
- Một số hình thức báo cáo phải được sự đồng ý của công ty, nhà quản lý

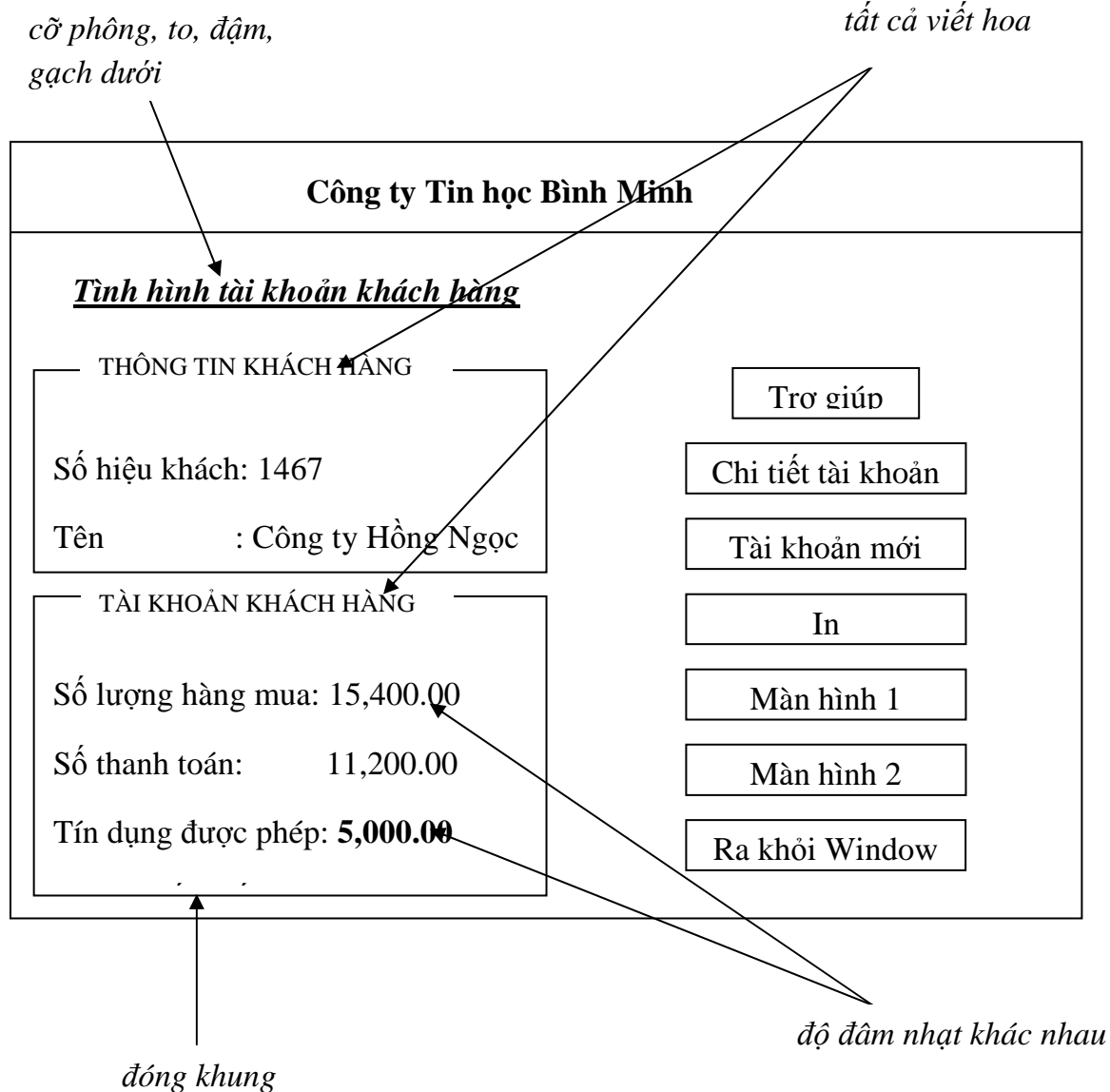
11.2.2 Quy trình thiết kế đầu ra

- Xem xét tất cả các đầu ra của hệ thống và các yêu cầu logic.
- Làm rõ đặc tả (vật lý) của các yêu cầu báo cáo.
- Thiết kế các bản mẫu trước (nếu cần).
- Thiết kế, kiểm thử và kiểm tra đầu ra của báo cáo

Ví dụ: Một khuôn mẫu đặc tả thiết kế

<p>a. Tổng quan sơ lược</p> <ul style="list-style-type: none">- <i>Tên biểu mẫu (báo cáo):</i> Trạng thái tài khoản của khách hàng- <i>Người sử dụng:</i> Người theo dõi tài khoản khách hàng trong văn phòng công ty- <i>Nhiệm vụ:</i> cung cấp các thông tin về tài khoản khách hàng, địa chỉ, số dư tài khoản, các khoản nợ và thanh toán đến thời điểm hiện tại, số tín dụng tối đa, tỉ lệ chiết khấu và tài khoản chung.- <i>Hệ thống:</i> Ms Windows- <i>Môi trường:</i> môi trường văn phòng chuẩn.																			
<p>b. Bản mẫu thiết kế</p> <table border="1"><thead><tr><th colspan="2">Công ty Tin học Bình Minh</th></tr></thead><tbody><tr><td colspan="2"><u>Tình hình tài khoản khách hàng</u></td></tr><tr><td>THÔNG TIN KHÁCH HÀNG</td><td>Trợ giúp</td></tr><tr><td>Số hiệu khách: 1467</td><td>Chi tiết tài khoản</td></tr><tr><td>Tên : Công ty Hồng Ngọc</td><td>Tài khoản mới</td></tr><tr><td>TÀI KHOẢN KHÁCH HÀNG</td><td>In</td></tr><tr><td>Số lượng hàng mua: 15,400.00</td><td>Màn hình 1</td></tr><tr><td>Số thanh toán: 11,200.00</td><td>Màn hình 2</td></tr><tr><td>Tín dụng được phép: 5,000.00</td><td>Ra khỏi Window</td></tr></tbody></table>		Công ty Tin học Bình Minh		<u>Tình hình tài khoản khách hàng</u>		THÔNG TIN KHÁCH HÀNG	Trợ giúp	Số hiệu khách: 1467	Chi tiết tài khoản	Tên : Công ty Hồng Ngọc	Tài khoản mới	TÀI KHOẢN KHÁCH HÀNG	In	Số lượng hàng mua: 15,400.00	Màn hình 1	Số thanh toán: 11,200.00	Màn hình 2	Tín dụng được phép: 5,000.00	Ra khỏi Window
Công ty Tin học Bình Minh																			
<u>Tình hình tài khoản khách hàng</u>																			
THÔNG TIN KHÁCH HÀNG	Trợ giúp																		
Số hiệu khách: 1467	Chi tiết tài khoản																		
Tên : Công ty Hồng Ngọc	Tài khoản mới																		
TÀI KHOẢN KHÁCH HÀNG	In																		
Số lượng hàng mua: 15,400.00	Màn hình 1																		
Số thanh toán: 11,200.00	Màn hình 2																		
Tín dụng được phép: 5,000.00	Ra khỏi Window																		

Sử dụng các kỹ thuật làm sáng rõ các thông tin khác nhau



Sử dụng màu:

- Màu là công cụ rất lợi hại. Việc sử dụng nó có ảnh hưởng đến tính khả dụng của một hệ thống. Khi sử dụng màu thích hợp, nó cung cấp nhiều lợi ích cho các mẫu biểu hay báo cáo. Ngược lại nó cũng mang lại nhiều phản tác dụng

- Nói chung không nên dùng nhiều màu. Dùng màu rất thích hợp cho các sơ đồ, đồ thị. Thiết kế trước hết được thực hiện ở dạng không màu, sau đó cho phép đưa màu vào những vị trí thực sự cần thiết. Thường thì, màu để hỗ trợ làm sáng rõ thông tin hay định dạng thông tin.

Câu hỏi thảo luận C11

11.1. Nêu các phương thức cài đặt đầu ra.

11.2. Nêu các nguyên tắc thiết kế đầu ra

11.3 Thiết kế một số mẫu báo cáo cho các bài toán thông dụng: hóa đơn mua – bán hàng; bảng điểm học sinh – sinh viên;...

11.4 Thiết kế đầu ra của 1 hệ thống tuyển sinh đại học chính quy.

Chương 12. THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

12.1. Tổng quan về giao diện người dùng

- Giao diện người dùng hiệu quả phải phù hợp với trình độ và kinh nghiệm của người dùng. Những nguyên nhân sau đây khiến cho người dùng sử dụng sai hay cảm thấy nhầm chán, lẫn lộn thậm chí hoảng sợ quay sang chối bỏ phần mềm:
 - o Sử dụng nhầm lẫn các thuật ngữ, khái niệm
 - o Giao diện không trực quan
 - o Cách tiếp cận giải quyết vấn đề bị lẫn lộn
 - o Thiết kế giao diện rắc rối
- Các nguyên tắc nên áp dụng khi thiết kế giao diện người dùng:
 - o Phải hiểu rõ trình độ người sử dụng cũng như đặc thù các công việc của họ
 - o Lôi kéo người dùng vào việc thiết kế giao diện
 - o Kiểm tra và thử nghiệm việc thiết kế trên người dùng thật
 - o Áp dụng các quy ước, thói quen trong thiết kế giao diện, tuân thủ style chung cho toàn chương trình.
 - o Người dùng cần được chỉ dẫn những công việc họ sẽ đối mặt tiếp theo:
- Chỉ cho người dùng hệ thống đang mong đợi họ làm gì
- Chỉ cho người dùng dữ liệu họ nhập đúng hay sai
- Giải thích cho người dùng hệ thống đang đứng yên do có công việc cần xử lý chứ không treo
- Khẳng định với người dùng hệ thống đã hay chưa hoàn thành một công việc nào đó
 - o Nên định hình giao diện sao cho các thông điệp, chỉ dẫn luôn xuất hiện tại cùng một vị trí

- o Định hình các thông điệp và chỉ dẫn đủ dài để người dùng có thể đọc được, đủ ngắn để họ có thể hiểu được
- o Các giá trị mặc định cần được hiển thị
- o Lường trước những sai sót người dùng có thể gặp phải để phòng tránh
- o Không cho phép xử lý tiếp nếu lỗi chưa được khắc phục

12.2. Kỹ thuật giao diện người dùng

12.2.1. Hệ điều hành và trình duyệt web

Những hệ điều hành đồ họa phổ biến cho các máy khách hiện nay là Windows, Macintosh, Unix, Linux và cho các máy cầm tay là Palm OS, Windows CE. Tuy nhiên, hệ điều hành ngày càng không còn là nhân tố chính trong thiết kế giao diện người dùng nữa. Các ứng dụng Internet và Intranet chạy trên các trình duyệt web. Hầu hết các trình duyệt có thể chạy trên nhiều hệ điều hành. Điều này cho phép thiết kế giao diện người dùng ít phụ thuộc vào hệ điều hành. Tính năng này được gọi là độc lập nền tảng (platform independence). Thay vì viết giao diện riêng cho từng hệ điều hành thì chỉ cần viết giao diện cho một hoặc hai trình duyệt. Hiện tại, hai trình duyệt phổ biến nhất là Microsoft Internet Explorer và Netscape Navigator nhưng vẫn còn tồn tại một khó khăn khác đó là vấn đề về các phiên bản trình duyệt.

12.2.2. Màn hình hiển thị

Kích thước vùng hiển thị là vấn đề then chốt khi thiết kế giao diện. Không phải màn hình hiển thị nào cũng là dạng màn hình máy tính cá nhân. Có rất nhiều thiết bị hiển thị không phải là máy tính cá nhân.

Đối với màn hình máy tính cá nhân, chúng ta có đơn vị đo lường là độ phân giải đồ họa. Độ phân giải đồ họa được tính theo pixel, đó là số điểm sáng phân biệt được hiển thị trên màn hình. Hiện nay, độ phân giải phổ biến là 800.000 pixel theo chiều ngang và 600.000 pixel theo chiều dọc trong một màn hình 17 inch. Những kích thước hiển thị lớn hơn hỗ trợ nhiều pixel hơn; tuy nhiên, người thiết kế nên thiết kế giao diện theo loại màn hình có độ phân giải phổ biến nhất.

Rõ ràng, các máy tính cầm tay và một số thiết bị hiển thị đặc biệt (ví dụ như màn hình máy rút tiền tự động ATM) hỗ trợ màn hình hiển thị nhỏ hơn nhiều cũng phải được xem xét khi thiết kế giao diện.

Cách thức thể hiện vùng hiển thị đối với người dùng được điều khiển bởi cả khả năng kỹ thuật của màn hình và khả năng của hệ điều hành, Hai cách tiếp cận phổ biến nhất là paging và scrolling. **Paging** hiển thị một màn hình hoàn chỉnh các ký tự vào cùng một lần. Toàn bộ vùng hiển thị được gọi là một trang (hay màn hình). Các trang được hiển thị theo nhu cầu của người dùng bằng cách nhấn nút lệnh, tương tự như lật các trang trong một cuốn sách. **Scrolling** dịch chuyển phần thông tin hiển thị lên hoặc xuống trên màn hình, thường là mỗi lần 1 dòng. Các màn hình máy tính cá nhân còn cho phép nhiều tùy chọn paging và scrolling.

12.2.3. Bàn phím và các thiết bị trợ

Hầu hết (nhưng không phải tất cả) các thiết bị hiển thị và màn hình đều được tích hợp với bàn phím. Những tính năng chủ yếu của bàn phím là tập ký tự và các khóa chức năng.

Tập ký tự của hầu hết các máy tính cá nhân đều theo chuẩn. Những tập ký tự đó có thể được mở rộng với phần mềm để hỗ trợ thêm các ký tự và biểu tượng. Các khóa chức năng nên được sử dụng một cách nhất quán. Nghĩa là, bất kỳ chương trình nào cũng nên sử dụng nhất quán các khóa chức năng cho cùng mục đích. Ví dụ, F1 thường được dùng để gọi chức năng trợ giúp trong cả hệ điều hành và các ứng dụng.

Hầu hết các giao diện (bao gồm các hệ điều hành và trình duyệt) đều sử dụng thiết bị trợ như chuột, bút và màn hình cảm ứng. Tất nhiên, thiết bị trợ phổ biến nhất vẫn là chuột.

Bút đang trở nên quan trọng trong các ứng dụng chạy trên các thiết bị cầm tay. Bởi lý do là những thiết bị đó thường không có bàn phím. Do đó, giao diện có thể cần được thiết kế để cho phép “gõ” trên một bàn phím được hiển thị trên màn hình hoặc sử dụng một chuẩn viết tay như Graffiti hoặc Jot.

12.3. Các phong cách thiết kế giao diện người dùng

12.3.1. Giao diện dựa trên cửa sổ và frame

Phần cơ bản nhất của một giao diện là cửa sổ. Một cửa sổ có thể nhỏ hoặc lớn hơn vùng màn hình hiển thị. Nó thường chứa các điều khiển chuẩn ở góc trên bên phải như phóng to, thu nhỏ hay đóng cửa sổ.

Phần dữ liệu hiển thị bên trong cửa sổ có thể lớn hoặc nhỏ hơn kích thước cửa sổ. Trong trường hợp lớn hơn, có thể dùng thanh cuộn để dịch chuyển.

Một cửa sổ có thể được chia thành các vùng gọi là frame. Mỗi frame có thể hoạt động độc lập với các frame khác trong cùng một cửa sổ. Mỗi frame có thể được xác định để phục vụ cho một mục đích nhất định.

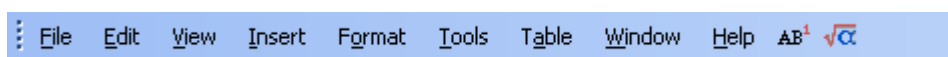
Trong một cửa sổ, chúng ta có thể sử dụng tất cả các điều khiển giao diện đã giới thiệu trong các chương 9 và 10.

12.3.2. Giao diện dựa trên menu

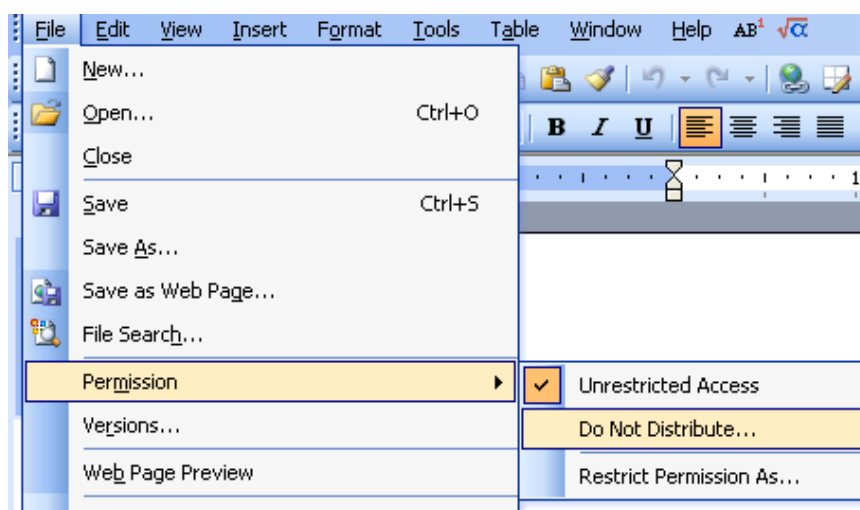
Chiến lược đối thoại phổ biến nhất và cổ điển nhất là menu. Có nhiều loại menu nhưng tư tưởng chung đều là yêu cầu người dùng chọn một hành động từ menu:

- Menu kéo thả, menu xếp tầng
- Menu pop-up
- Thanh công cụ và menu icon
- Menu siêu liên kết

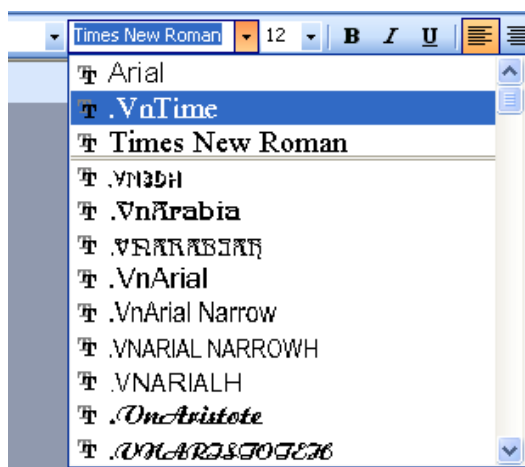
thanh menu bar:



menu xếp tầng:



menu kéo thả



Bảng: Hướng dẫn thiết kế các thực đơn

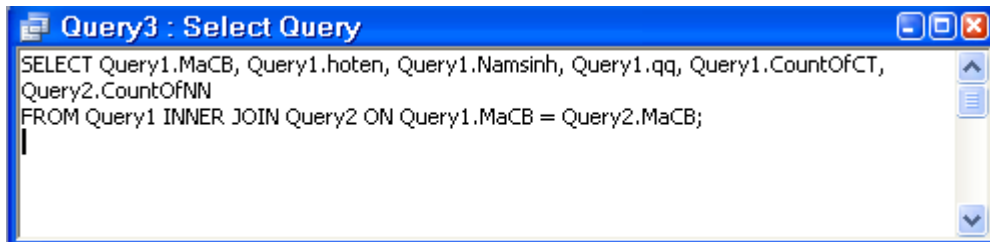
Từ mục	Một thực đơn cần có một tiêu đề có đủ ý nghĩa Câu lệnh rõ ràng, mô tả chính xác thao tác cần thực hiện Các mục thực đơn hiển thị gồm cả chữ hoa, chữ thường, có sự giải thích rõ ràng, không mơ hồ.
Cách tổ chức	Nguyên tắc tổ chức bền vững cần được sử dụng để gắn kết được nhiệm vụ cần thực hiện với người sử dụng nó
Độ dài	Số lượng các tùy chọn không vượt quá khuôn khổ màn hình Sử dụng thực đơn con để phân cấp và rút ngắn thực đơn dài
Sự lựa chọn	Phương pháp lựa chọn và truy nhập phải vững chắc, phản ánh được quy mô của ứng dụng và các triết lý của người dùng Lựa chọn tùy chọn, trình tự của sắp xếp các tùy chọn cần được tổ chức hợp lý phù hợp với thói quen, kinh nghiệm và nhiệm vụ thực hiện
Làm sáng rõ	Việc làm sáng rõ cần hạn chế ở mức tối thiểu và chỉ nên thực hiện với tùy chọn được chọn

12.3.3. Giao diện dựa trên dòng lệnh

Thay cho menu hoặc cũng có thể bổ sung thêm cho menu, một số ứng dụng được thiết kế sử dụng đối thoại dựa trên tệp lệnh (còn gọi là giao diện ngôn ngữ lệnh – command language interface). Tuy nhiên, người sử dụng phải học cú pháp tập lập nên cách tiếp cận này chỉ phù hợp với đối tượng người dùng chuyên gia. Có 3 loại cú pháp, lựa chọn loại nào là phụ thuộc vào công nghệ có thể dùng:

- Cú pháp dựa trên ngôn ngữ (ví dụ như SQL)
- Cú pháp mnemonic: người sử dụng được cung cấp một màn hình giao tiếp trong đó họ có thể nhập các lệnh gọi tới các hành động. Các câu lệnh phải có nghĩa với người sử dụng.
- Cú pháp ngôn ngữ tự nhiên: cho phép người dùng nhập các câu hỏi các lệnh bằng ngôn ngữ tự nhiên. Hệ thống thông dịch các lệnh đó theo cú pháp đã biết và có thể yêu cầu người dùng nhập lại rõ ràng hơn nếu nó không hiểu được ý muốn của người dùng.

Cách thiết kế giao diện dựa trên dòng lệnh từng phổ biến trong các ứng dụng máy tính lớn và các ứng dụng máy tính cá nhân trên DOS trước đây. Nhưng phong cách tương tác này vẫn được sử dụng trong một số ứng dụng hiện nay. Ví dụ như Microsoft Access có phần soạn thảo câu truy vấn như hình dưới đây:



Thiết kế giao diện người dùng là việc đặc tả sự đối thoại giữa người sử dụng chương trình và máy tính. Sự đối thoại này thường cho kết quả là dữ liệu đầu vào và thông tin đầu ra. Có một số phong cách thiết kế giao diện người dùng. Trước đây, những hình thức đó được xem là loại bỏ nhau nhưng ngày nay, chúng đang pha trộn lẫn nhau. Mục này giới thiệu tổng quan một số phong cách và chiến lược được dùng để thiết kế giao diện người dùng và cách thức chúng được kết hợp vào các ứng dụng.

12.3.4. Đối thoại hỏi – đáp

Hình thức đối thoại hỏi đáp được dùng chủ yếu để hỗ trợ cho đối thoại dựa trên menu hoặc dựa trên câu lệnh. Người dùng được gợi ý bằng câu hỏi mà họ cần cho câu trả lời. Câu hỏi đơn giản nhất là Yes/No. Chiến lược này yêu cầu chúng ta phải xét mọi câu trả lời đúng có thể có và chuẩn bị mọi hành động nếu xuất hiện câu trả lời sai. Rõ ràng đây là một hình thức giao diện khó thiết kế. Tuy nhiên, hình thức này phổ biến trong các ứng dụng trên web.

12.3.5. Một số tính năng đặc biệt

- Xác thực và phân quyền

Trong hầu hết các hệ thống, người sử dụng phải được xác thực trước khi họ được phép sử dụng hệ thống. Nói một cách khác, người sử dụng phải đăng nhập vào hệ thống. Hầu hết việc đăng nhập đều yêu cầu tên người dùng (username) và mật khẩu (password). Mỗi người dùng được cấp một quyền hạn sử dụng một số chức năng nhất định. Người dùng có quyền cao nhất thường là người quản trị hệ thống.

Trang chủ

Tài khoản đăng nhập

Tên truy cập *

Hãy điền tên đăng nhập ở Trang thông tin điện tử Trường đại học giao thông vận tải của bạn

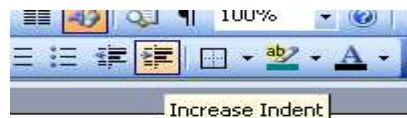
Mật khẩu *

Điền mật khẩu đi kèm với tên đăng nhập.

Hình: Ví dụ đăng nhập một hệ thống

- Trợ giúp
 - Tooltip: xuất hiện khi người dùng đưa chuột vào vị trí biểu tượng (hoặc đối tượng) trên màn hình. Tooltip chứa một đoạn mô tả ngắn gọn về chức năng thể hiện bởi đối tượng tương ứng.

Ví dụ:

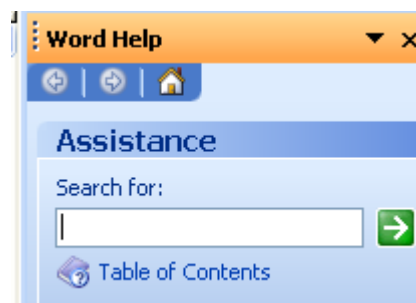


- Help Wizard: hướng dẫn người sử dụng thông qua một quá trình phức tạp bằng cách đưa ra một chuỗi các hộp đối thoại yêu cầu người dùng phải đưa ra đầu vào và trả về phản hồi của hệ thống.

Ví dụ:



- Tác tử (Agents): là các đối tượng phần mềm có thể hoạt động trên nhiều ứng dụng phần mềm và thậm chí là trên các mạng. Ví dụ như tác tử trợ giúp của Microsoft (có thể hiểu là trợ lý) cung cấp một trợ lý chung trong các ứng dụng Office. Nó cho phép người dùng đặt câu hỏi bằng một đoạn ngôn ngữ tự nhiên. Đoạn ngôn ngữ sẽ được thông dịch bởi tác tử để đưa ra đáp ứng phù hợp. Sau đó, người dùng có thể lựa chọn một trong các đáp ứng để chọn ra chỉ mục trợ giúp chi tiết hơn nữa.



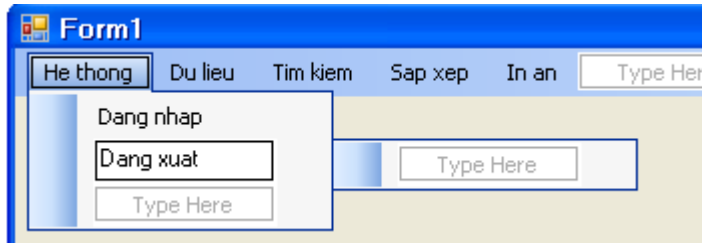
12.4. Cách thức thiết kế giao diện người dùng

12.4.1. Các công cụ tạo giao diện

- Microsoft Access

- CASE Tools
- Visual Basic
- Visio
- ...

Xây dựng menu với Visual Basic .NET:



12.4.2. Quy trình thiết kế giao diện người dùng

Bước 1 - Lập sơ đồ phân cấp giao tiếp người dùng hoặc sử dụng lược đồ biến đổi trạng thái

Bước 2 - Lập bản mẫu đối thoại và giao diện người dùng

Bước 3 - Tham khảo và tiếp thu ý kiến phản hồi của người dùng. Nếu cần thiết quay trở lại bước 1 và bước 2.

Câu hỏi thảo luận C12

12.1. Nêu các nguyên tắc tổng thiết kế giao diện.

12.2. Nêu các kỹ thuật giao diện.

12.3. So sánh các phong cách giao diện.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Phân tích và thiết kế hệ thống thông tin, Nguyễn Văn Ba, Nhà xuất bản ĐHQG Hà Nội, 2006
2. Phân tích và thiết kế hệ thống thông tin, Nguyễn Văn Vy, Nhà xuất bản ĐHQG Hà Nội, 2001
3. Phân tích và thiết kế hệ thống, Ebook, Đại học Sư phạm Hà Nội, 2006
4. Giáo trình Phân tích hệ thống, Ebook, Đại học Cần Thơ
5. Giáo trình Phân tích và thiết kế hệ thống, Phan Huy Khánh, Đại học Bách Khoa Đà Nẵng, 2001.
6. Giáo trình Phân tích thiết kế hệ thống, Đại học Quốc gia Tp Hồ Chí Minh.

Tiếng Anh

7. Information Systems Analysis and Design, Myriam Lewkowicz
8. PPP DFD Editor, The Norwegian University Of Science and Technology