



Phân tích thiết kế hệ thống thông tin quản lý

Chương 1. Đại cương về hệ thống thông tin quản lý

Giới thiệu sơ lược một số phương pháp phân tích thiết kế

Phân tích thiết kế hệ thống thông tin là phương pháp luận để xây dựng và phát triển hệ thống thông tin bao gồm các lý thuyết, mô hình, phương pháp và các công cụ sử dụng trong quá trình phân tích và thiết kế hệ thống. Có nhiều phương pháp phân tích khác nhau. Ở đây chúng ta tóm lược một vài phương pháp quan trọng để làm phương tiện so sánh và đối chiếu tham khảo các tài liệu khác.

Phương pháp Sadt (Structured Analysis and Design Technique) - Kỹ thuật phân tích và Thiết kế cấu trúc Phương pháp này xuất phát từ Mỹ, ý tưởng cơ bản của nó là: phân rã một hệ thống lớn thành các phân hệ nhỏ và đơn giản.

SADT được xây dựng dựa trên 7 nguyên lý sau đây.

- Sử dụng một mô hình
- Phân tích đi xuống (top down)
- Dùng một mô hình chức năng và một mô hình quan niệm (còn được gọi là " Mô hình thiết kế ")
- Thể hiện tính đối ngẫu của hệ thống
- Sử dụng các biểu diễn dưới dạng đồ họa.
- Phối hợp hoạt động của nhóm.
- Ưu tiên tuyệt đối cho hồ sơ viết.

SADT được định nghĩa là phương pháp sử dụng các kỹ thuật :

- Dòng dữ liệu hay còn gọi là biểu đồ luồng dữ liệu (Data Flow Diagrams)
- Từ điển dữ liệu (Data Dictionary)
- Anh ngữ có cấu trúc (Structured English)
- Bảng quyết định
- Cây quyết định

Phương pháp SADT có nhiều ưu điểm như dựa vào nguyên lý phân tích cấu trúc, thiết kế theo lối phân cấp, dựng trên các lưu đồ chức năng, tạo được các liên hệ " Một cha nhiều con " (One parent to many children relationship), bảo đảm từ một dữ liệu vào sản xuất nhiều dữ liệu ra.

Nhưng nhược điểm của nó là không bao gồm toàn bộ tiến trình phân tích và nếu không thận trọng sử dụng

SADT có thể đưa tiến trình trùng lặp thông tin.

Phương pháp này được dùng khá phổ biến, truyền thống do tính logic của nó.

Tài liệu này sẽ bám sát phương pháp thiết kế SADT và tham khảo các phương pháp khác

Phương pháp MERISE (Methode pour Rassembler les Ideés Sans Effort)

Phương pháp đề tập hợp các ý tưởng không cần cố gắng

Phương pháp MERISE là phương pháp phân tích có nguồn gốc từ Pháp, ra đời từ những năm cuối thập niên 70. Nó là kết quả nghiên cứu của nhiều tập thể nghiên cứu tin học nhằm đáp ứng các chờ đợi của người sử dụng, ý thức được về sự lạc hậu của các phương pháp phân tích cổ điển thế hệ thứ nhất.

Ý tưởng cơ bản của phương pháp MERISE là xuất phát từ ba mặt cơ bản sau :

Mặt thứ nhất :

Quan tâm đến chu kỳ sống của hệ thống thông tin, trải qua nhiều giai đoạn: " **Thai nghén**"(Gestation) - Quan niệm/ý niệm - Quản trị - chết . Chu kỳ sống này đối với hệ thống tổ chức lớn có thể kéo dài từ 10-15 năm.

Mặt thứ hai :

Đề cập tới chu kỳ đặc tả của hệ thống thông tin còn được gọi là chu kỳ trừu tượng.

Hệ thống thông tin tựu trung lại như một toàn thể được miêu tả bởi nhiều tầng (Couche): "**Bộ nhớ**" của hệ thống thông tin được mô tả trên bình diện quan niệm, kế đó trên bình diện logic và cuối cùng trên bình diện vật lý. "**Qui trình xử lý**" được mô tả trên bình diện quan niệm, kế tiếp là trên bình diện tổ chức và cuối cùng là trên bình diện tác nghiệp.

Mỗi tầng được mô tả dưới dạng mô hình tập trung tập hợp các thông số chính xác. Theo đó kho những thông số của tầng dưới tầng trưởng, tầng đang mô tả không biến đổi và nó chỉ thay đổi khi các tham số của mình thay đổi.

Mỗi mô hình được mô tả thông qua một hình thức dựa trên các nguyên tắc, nguyên lý ngữ vựng và cú pháp xác định. Có những qui tắc chuyển cho phép chuyển từ mô hình này sang mô hình khác một cách tự động nhiều hay ít.

Mặt thứ ba:

Mặt này có liên quan đến chu kỳ của các quyết định (Cycle des Decisions) cần phải ra trong suốt chu kỳ sống của sản phẩm. Những quyết định có liên quan đến nội dung của những mô hình khác nhau của chu kỳ trừu tượng, đến các hình thái của quan niệm và liên quan đến sự phát triển của hệ thống .

Đặc trưng cơ bản của phương pháp MERISE là :

- + Nhìn toàn cục
- + Tách rời các dữ liệu và xử lý
- + Tiếp vận theo mức

Có thể tóm tắt nội dung thứ hai và nội dung thứ ba thể hiện qua việc nhận thức và xây dựng các loại mô hình trong quá trình phân tích và thiết kế bằng bảng sau :

Mức	Dữ liệu	Xử lý
Quan niệm	Mô hình quan niệm dữ liệu	Mô hình quan niệm xử lý
Tổ chức	Mô hình Logic dữ liệu	Mô hình tổ chức xử lý
Kỹ thuật	Mô hình vật lý dữ liệu	Mô hình tác vụ xử lý

Ưu điểm của phương pháp MERISE là có cơ sở khoa học vững chắc. Hiện tại nó là một trong những phương pháp phân tích được dùng nhiều ở Pháp và các nước Châu Âu khi phải phân tích và thiết kế các hệ thống lớn. Nhược điểm của phương pháp này là công kênh, do đó, để giải quyết các áp dụng nhỏ việc sử dụng phương pháp này nhiều lúc đưa đến việc kéo dài thời gian, nặng nề không đáng có.

Phương pháp MCX (Methode de xavier castellani)

Phương pháp phân tích MCX có nguồn gốc từ Pháp, do giáo sư của Viện tin học xí nghiệp (IIE - Institut Informatique d' entreris) sáng tạo. Phương pháp này khá thông dụng và thoả các điều kiện của các phương pháp phân tích thế hệ thứ hai .

Có thể nêu một số nét cơ bản về phương pháp phân tích MCX

- Cho phép xây dựng được một mô hình tổng quát, chính xác, biểu diễn hệ thống thông tin hoặc các phân hệ thông tin.
- Cho phép phân tích, nắm được dữ liệu, quá trình xử lý và truyền thông các hệ thống thông tin.
- Cho phép biểu diễn các xử lý với các lưu đồ và các chương trình, soạn thảo bởi một ngôn ngữ giải thuật dùng ở các mức khác nhau.
- Cho phép lượng hoá các xử lý.
- Phương pháp MCX đưa ra các giai đoạn cơ bản của quá trình phân tích.
- Phân tích macro
- Phân tích sơ bộ
- Phân tích quan niệm
- Phân tích chức năng
- Phân tích cấu trúc

Phương pháp phân tích này khá hữu hiệu, thích hợp với việc thực hành. Nhược điểm là rườm rà.

Phương pháp GALACSI (Groupe d' Animation et de Liaison pour l' Analyse et la Conception de système d' Information)

Phương pháp GALACSI có nguồn gốc tại Pháp, do một nhóm các giáo sư của các học viện công nghệ IUT (Institut Universitaire de Technologic) và MIAGE (Maitrise de Methodes Informatiques Applyquées à la Gestion - Cao học về phương pháp tin học áp dụng vào quản lý).

Phương pháp GALACSI chính thức ra đời vào tháng 04 năm 1982. Nội dung cơ bản của phương pháp trình bày một tập hợp các công cụ và "nguyên liệu" để tiến hành các giai đoạn cơ bản sau đây của quá trình phân tích.

1. Nghiên cứu các hệ thống tổ chức và hệ thống thông tin tương ứng:

- * Nghiên cứu hiện trạng
- * Nghiên cứu khả thi

2. Phân tích chức năng

- * Mô hình dữ liệu
- * Mô hình xử lý

3. Phân tích cấu trúc

- * Tổ chức dữ liệu : ở mức logic và vật lý
- * Tổ chức xử lý : Xử lý theo lô, xử lý theo thời gian thực (đối thoại người với máy)

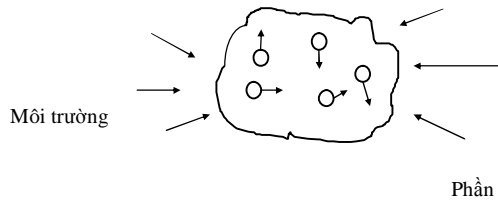
- * Môi trường tiếp nhận : Máy vi tính, mạng máy tính, ngôn ngữ, các phần mềm chuyên dụng.
- * Giao diện người - máy : Công thái học, ngôn ngữ giao tiếp.

4. Lập trình:

Giải thuật, ngôn ngữ lập trình, kiến trúc các môi trường đặt thù. Do phần lớn các tác giả là các giáo sư nên phương pháp được dùng để giảng dạy trong nhiều học viện (IUT) . Nhược điểm của phương pháp là chưa thử nghiệm nhiều trong thực tế.

1. Các hệ thống kinh doanh :

Hệ thống : Là một tập hợp có tổ chức của nhiều phần tử có những mối ràng buộc lẫn nhau và cùng hoạt động chung cho một mục đích nào đó.



Hệ thống kinh doanh là hệ thống có mục đích phục vụ cho kinh doanh (Business). Kinh doanh có thể vì lợi ích hoặc vì lợi nhuận. Việc phân định này chỉ mang tính tương đối và nó thật sự cần thiết để sau này khi xây dựng hệ thống ta có thể kiểm nghiệm hệ thống đã đạt được yêu cầu và mục tiêu chưa.

Thí dụ : Các công ty, nhà máy, dịch vụ ... là hệ thống kinh doanh vì lợi nhuận

Các trường học, công trình công cộng, bệnh viện ... là hệ thống kinh doanh vì lợi ích

Đặc điểm của hệ thống kinh doanh là có sự tham gia của con người nên mang theo nhiều đặc điểm, ưu điểm, khuyết điểm của con người.

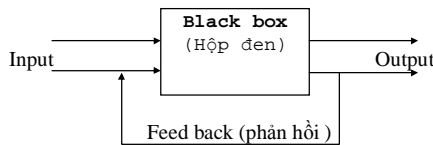
Các thành phần của hệ thống kinh doanh: Một hệ thống kinh doanh có thể phân làm 3 hệ thống con :

- Hệ thống quyết định là hệ thống bao gồm: con người, phương tiện và các phương pháp tham gia đề xuất quyết định.
- Hệ thống thông tin là hệ thống bao gồm: con người, phương tiện và các phương pháp tham gia xử lý thông tin kinh doanh (hệ quản trị).
- Hệ tác nghiệp là hệ thống bao gồm: con người, phương tiện và các phương pháp tham gia trực tiếp thực hiện mục tiêu kinh doanh (sản xuất trực tiếp). Một cách tổng quát hệ tác nghiệp là các hoạt động nhằm thực hiện có tính cách cạnh tranh để đạt được mục tiêu đã xác định của hệ quyết định.

Lưu ý rằng nhiệm vụ của môn học xây dựng hệ thống thông tin nên tránh sa đà nhằm lẫn với hệ tác nghiệp

2. Nhiệm vụ và vai trò của hệ thống thông tin

Hệ thống thông tin là phân hệ con của hệ thống kinh doanh. Chức năng chính của nó là xử lý thông tin của hệ thống. Sự phân chia này có tính phương pháp luận chứ không phải là sự chia mang tính vật lý. Quá trình xử lý thông tin tương tự như hộp đen gồm bộ xử lý, thông tin đầu vào, thông tin đầu ra và thông tin phản hồi của hệ thống



Thông tin kinh doanh có 2 loại sau đây.

- Thông tin tự nhiên là loại thông tin giữ nguyên dạng khi nó phát sinh : tiếng nói, công văn, hình ảnh ... Việc xử lý thông tin này thuộc về công tác văn phòng với kỹ thuật mang đặc điểm khác nhau
- Thông tin có cấu trúc là thông tin được cấu trúc hoá với khuôn dạng nhất định thường dạng sổ sách, bảng biểu quy định

- Nhiệm vụ của hệ thống thông tin:
 - Đối ngoại: Hệ thống thông tin thu nhận thông tin từ môi trường ngoài, đưa thông báo ra ngoài. Ví dụ như thông tin về giá cả, thị trường,...

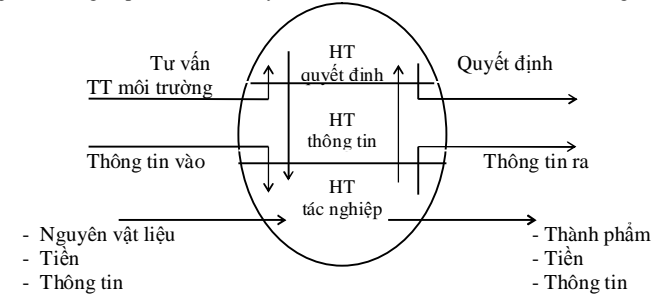
- Đối nội: Hệ thống thông tin là cầu nối liên lạc giữa các bộ phận của hệ kinh doanh. Nó cung cấp cho hệ tác nghiệp, hệ quyết định thông tin nhằm 2 loại sau :

§ Phản ánh tình trạng nội bộ của cơ quan, tổ chức trong hệ thống .

§ Tình trạng hoạt động kinh doanh của hệ thống.

§ Vai trò của hệ thống thông tin :

Hệ thống thông tin đóng vai trò trung gian giữa hệ thống và môi trường, giữa hệ thống con quyết định và hệ thống con tác nghiệp. Sơ đồ dưới đây cho ta các nhìn nhận vai trò của hệ thống thông tin



3. Các thành phần hợp thành của hệ thống thông tin:

a) Đặc điểm của HTTT quản lý:

HTTT là hệ thống được tổ chức thống nhất từ trên xuống dưới có chức năng tổng hợp các thông tin giúp các nhà quản lý tốt cơ sở của mình và trợ giúp ra quyết định hoạt động kinh doanh. Một hệ thống quản lý được phân thành nhiều cấp từ trên xuống dưới và chuyển từ dưới lên trên.

b) Các thành phần cơ bản của HTTT

Nếu không kể con người và phương tiện thì HTTT còn lại thực chất gồm 2 bộ phận: Dữ liệu và xử lý.

- Các dữ liệu: Các thông tin có cấu trúc. Với mỗi cấp quản lý lượng thông tin xử lý có thể rất lớn, đa dạng và biến động cả về chủng loại, về cách thức xử lý. Thông tin cấu trúc bao gồm luồng thông tin vào và luồng thông tin ra.

Luồng thông tin vào

Có thể phân loại các thông tin cần xử lý thành ba loại sau:

- Thông tin cần cho tra cứu: Các thông tin dùng cho tra cứu là thông tin dùng chung cho hệ thống và ít bị thay đổi. Các thông tin này thường được cập nhật một lần và chỉ dùng cho tra cứu trong việc xử lý thông tin sau này.
- Thông tin luân chuyển chi tiết: Các thông tin luân chuyển chi tiết là loại thông tin chi tiết về hoạt động của đơn vị, khối lượng, khối lượng thông tin thường rất lớn, cần phải xử lý kịp thời.
- Thông tin luân chuyển tổng hợp: Các thông tin luân chuyển tổng hợp là loại thông tin được tổng hợp từ hoạt động của các cấp thấp hơn, thông tin này thường có động, xử lý theo kỳ, theo lô.

Luồng thông tin ra

- Thông tin đầu ra được tổng hợp từ các thông tin đầu vào và phụ thuộc vào nhu cầu quản lý trong từng trường hợp cụ thể, từng đơn vị cụ thể. Thông tin ra là việc tra cứu nhanh về một đối tượng cần quan tâm đồng thời phải đảm bảo chính xác kịp thời.
- Các thông tin đầu ra quan trọng nhất được tổng hợp trong quá trình xử lý là các báo cáo tổng hợp, thống kê, thông báo. Các mẫu biểu báo cáo thống kê phải phản ánh cụ thể trực tiếp, sát với một đơn vị.
- Ngoài những yêu cầu được cập nhật thông tin kịp thời cho hệ thống, luồng thông tin ra phải được thiết kế linh hoạt mềm dẻo. Đây là chức năng thể hiện tính mở, tính giao diện của hệ thống thông tin đầu ra gắn với chu kỳ thời gian tùy ý theo yêu cầu của bài toán cụ thể, từ đó ta có thể lọc bớt được thông tin thừa trong quá trình xử lý.

Các xử lý là các quy trình, phương pháp, chức năng xử lý thông tin được lưu giữ lâu dài nhưng luôn tiến triển do 2 nguồn gốc.

- Tự nhiên tiến hoá: Thông tin làm thay đổi tình trạng về nội bộ.
- Tự nhiên hoạt động: Thông tin làm thay đổi tình trạng hoạt động kinh doanh của doanh nghiệp.

4. Các hệ thống tự động hoá :

Hệ thống tự động là hệ thống có sự tham gia của máy tính để xử lý thông tin, có nhiều mức độ xử lý khác nhau.

* Mức độ tự động hoá :

- Toàn bộ: Con người chỉ đóng vai trò phụ trong hệ thống.

- **Một phần:** Chia công việc xử lý giữa người (thủ công) và máy tính.
- Việc tự động hoá một hệ thống kinh doanh có thể làm với 2 cách
- Sử dụng máy tính tập trung bao trùm toàn bộ (Phương pháp hồ)
 - Áp dụng máy tính cho từng bộ phận riêng rẽ, cho từng phạm vi (Phương pháp giếng)

*** Phương thức xử lý thông tin:**

- **Xử lý mẻ (Batch Processing):** Thông tin đến đợi theo mẻ. Thí dụ như tính lương, tuyển sinh, các bài toán giải quyết có tính định kỳ theo chu kỳ thời gian nhất định
- **Xử lý trực tuyến (on-line processing):** Thông tin đến xử lý ngay. Thí dụ như bán vé máy bay, vé tàu, hệ INTERNET

Phương thức này thường dùng cho các trường hợp sau :

- In các báo cáo, kết xuất, thống kê.
- In các giấy tờ giao dịch có số lượng lớn

Xử lý có tính chất định kỳ thường dùng khi:

- Vào ra và xử lý một số lượng nhỏ các giao dịch
- Hiện thị, chỉnh đốn, sửa chữa các tệp
- Phục vụ trực tiếp khách hàng tại chỗ

Ngày nay người ta có xu hướng dùng xử lý trực tuyến nhiều do máy có giá thành thấp, nhưng điều đó không hẳn là hay.

*** Ưu điểm trực tuyến**

- Giảm được công việc giấy tờ, các khâu trung gian.
- Kiểm tra được sự đúng đắn của dữ liệu ngay khi thu nhập.
- Người dùng hiểu rõ được qui trình xử lý (người dùng tham gia với vai trò tích cực).
- Cho trả lời nhanh chóng.

*** Nhược điểm :**

- Đắt hơn (cả về phần cứng và phần mềm)
- Xây dựng tốn công, tốn thì giờ hơn
- Sử dụng CPU không kinh tế (phải thường trực ngay cả lúc không làm gì)
- Xử lý chậm khi khối lượng lớn.
- Khó bảo đảm tính tin cậy (Reliability).
- Khó phục hồi dữ liệu (vì dữ liệu ở trên đồng).
- Đòi hỏi nhiều biện pháp đặc biệt dữ liệu.

5. Các giai đoạn phân tích, thiết kế và cài đặt :

Các công việc cần hoàn thành .

- Xác định vấn đề và yêu cầu.
- Xác định mục tiêu, ưu tiên.
- Thiết kế logic (trả lời câu hỏi làm gì ? hoặc là gì ? What ?).
- Thiết kế vật lý (đưa những biện pháp, phương tiện, How ?).
- Cài đặt (lập trình).
- Khai thác và bảo trì.

Việc phân giai đoạn tùy từng phương pháp và chỉ có tính tương đối.

- * **Giai đoạn 1:** - Khảo sát hiện trạng và xác lập dự án.
- Tìm hiểu phê phán đề đưa ra giải pháp

- * **Giai đoạn 2:** - Phân tích hệ thống.

Phân tích sâu hơn các chức năng, các dữ liệu của hoạt động cũ để đưa ra mô tả hoạt động mới (giai đoạn thiết kế logic).

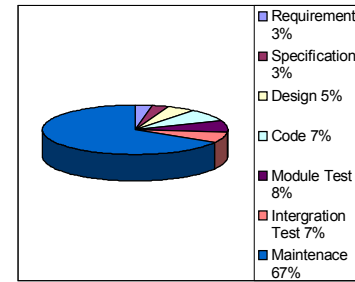
- * **Giai đoạn 3:** Thiết kế tổng thể (xác lập vai trò của môi trường một cách tổng thể trong hệ thống).

- * **Giai đoạn 4:** Thiết kế chi tiết :
- Thủ công.
- Kiểm soát phục hồi.
- Thiết kế cơ sở dữ liệu.
- Chương trình.

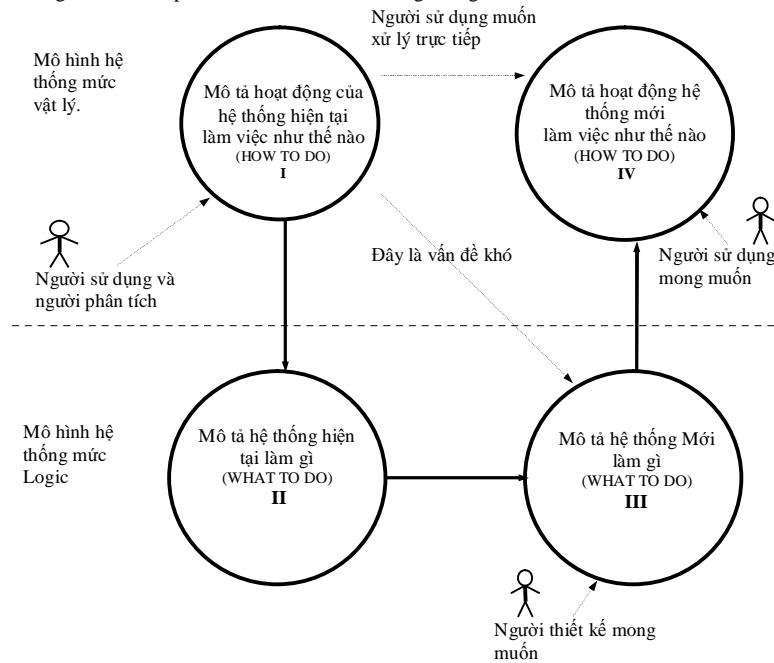
- * **Giai đoạn 5:** Cài đặt, lập trình

- * **Giai đoạn 6:** Khai thác và bảo trì

Theo một thống kê tính về chi phí cho các giai đoạn này được thể hiện bằng sơ đồ sau (Zelkowitz 1978, Trang 202)

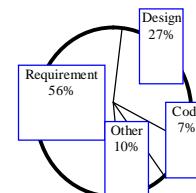


Các giai đoạn của phân tích và thiết kế hệ thống thông tin



Bài tập chương 1:

- 1.1 Tại sao khi xây dựng các phần mềm cần phải phân tích và thiết kế hệ thống ?
- 1.2 Nêu vai trò hệ thống thông tin trong hệ thống kinh doanh.
- 1.3 Nêu các giai đoạn của quá trình phân tích và thiết kế hệ thống.
- 1.4 Những lĩnh vực ứng dụng nào phù hợp với phương thức xử lý thông tin theo lô (batch), và lĩnh vực nào phù hợp xử lý theo trực tuyến (on-line)
- 1.5 Phân biệt hệ thống tin quản lý (MIS) với hệ trợ giúp quyết định (DSS) và hệ chuyên gia (ES)
- 1.6 Hãy thảo luận sơ đồ phân bố các sự cố sai sót của vòng đời hệ thống



Chương 2. Các công cụ diễn tả xử lý

2.1 Đại cương: Phân tích thiết kế hệ thống nói chung là sự nhận thức và mô tả một hệ thống; bởi vậy người ta thường dùng các mô hình, các biểu đồ để trừu tượng hoá và là công cụ giúp con người trao đổi với nhau trong quá trình phát triển hệ thống. Mỗi mô hình là một khuôn dạng để nhận thức về hệ thống và nó mang ý thức chủ quan.

Mục tiêu của phân tích mô hình xử lý là đưa ra một cách xác định các yêu cầu của người dùng trong quá trình phát triển hệ thống; những yêu cầu này được bám sát từ một loạt các sự kiện mà người phân tích thu được qua phỏng vấn, đặt câu hỏi, đọc tài liệu và qua các phép đo thử nghiệm

Có một số công cụ chính để diễn tả chức năng của hệ thống:

- Biểu đồ phân cấp chức năng (BPC) - Functional Hierarchical Decomposition Diagram (FHD):
- Biểu đồ luồng dữ liệu (BLD) - Data Flow Diagram (DFD)
- Các kí hiệu mở rộng của hãng IBM
- Sơ đồ thuật toán (Algorithm)
- Ngôn ngữ giả trình (Pseudo Code)
- Các đặc tả các qui tắc quản lý
- Từ điển định nghĩa chức năng xử lý

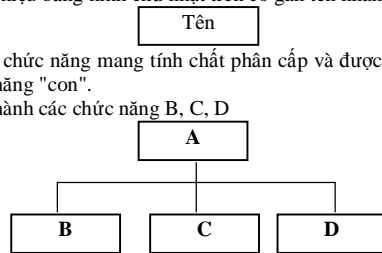
2.2. Biểu đồ phân cấp chức năng (BPC)

BPC là công cụ khởi đầu để mô tả hệ thống qua chức năng do công ty IBM phát triển vì vậy cho đến nay nó vẫn còn được sử dụng. Nó cho phép phân rã dần dần các chức năng từ chức năng mức cao thành chức năng chi tiết nhỏ hơn; và kết quả cuối cùng ta thu được một cây chức năng. Cây chức năng này xác định một cách rõ ràng để hiểu cái gì xảy ra trong hệ thống.

Thành phần của biểu đồ bao gồm :

- **Các chức năng:** được kí hiệu bằng hình chữ nhật trên có gắn tên nhãn
- **Kết nối:** kết nối giữa các chức năng mang tính chất phân cấp và được kí hiệu bằng đoạn thẳng nối chức năng "cha" tới các chức năng "con".

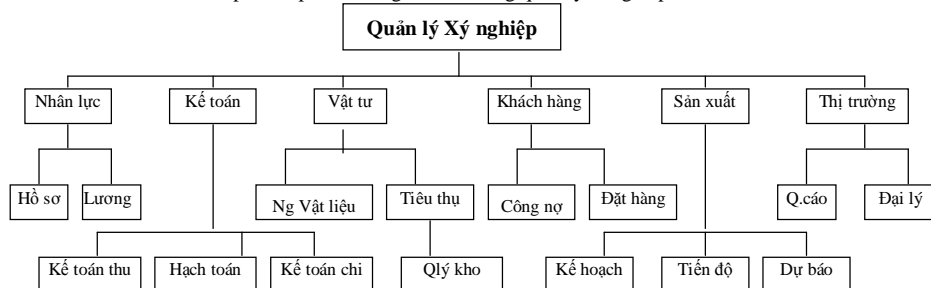
Thí dụ : Chức năng A phân rã thành các chức năng B, C, D



Đặc điểm của BPC :

- Các chức năng được nhìn một cách khái quát nhất, trực quan dễ hiểu, thể hiện tính cấu trúc của phân rã chức năng (Functionally Decomposed)
- Dễ thành lập vì tính đơn giản : Nó trình bày hệ thống phải làm gì hơn là hệ thống làm như thế nào?
- Mang tính chất tĩnh vì bỏ qua mối liên quan thông tin giữa các chức năng. Các chức năng không bị lặp lại và không dư thừa
- Rất gần gũi với sơ đồ tổ chức nhưng ta không đồng nhất nó với sơ đồ tổ chức: Phần lớn các tổ chức của doanh nghiệp nói chung thường gắn liền với chức năng .

Thí dụ : Hình 2.2 Biểu đồ phân cấp chức năng của hệ thống quản lý xí nghiệp.



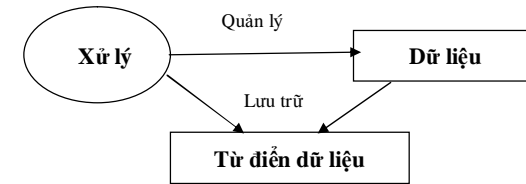
2.3 Biểu đồ luồng dữ liệu (BLD)

a. Mục đích: Diễn tả tập hợp các chức năng của hệ thống trong các mối quan hệ trước sau trong tiến trình xử lý, trong bàn giao thông tin cho nhau. Mục đích của biểu đồ luồng dữ liệu là giúp chúng ta thấy được đằng sau những cái gì thực tế xảy ra trong hệ thống (cái bản chất), làm rõ những chức năng và thông tin nào cần thiết cho quản lý

Biểu đồ này dựa vào phương pháp phát triển hệ thống có cấu trúc bao gồm 3 kỹ thuật phân tích chính:

- **Sơ đồ dòng dữ liệu (Data Flow Diagram)** mô tả quan hệ giữa quá trình xử lý và các dòng dữ liệu
- **Từ điển định nghĩa dữ liệu (Data Dictionary Definitions)** một tá các phần tử dòng dữ liệu
- **Xác định quá trình xử lý (Process Specifications)** mô tả quá trình xử lý một cách chi tiết

Mối quan hệ giữa 3 thành phần là bức tranh sinh động của hệ thống được thể hiện qua sơ đồ sau:



b. Tác dụng: BLD là công cụ chính của quá trình phân tích, nhằm mục đích thiết kế trao đổi và tạo lập dữ liệu. Nó thể hiện rõ ràng và khá đầy đủ các nét đặc trưng của hệ thống trong các bước phân tích, thiết kế và trao đổi tư liệu

c. Các mức diễn tả:

- **Mức vật lí:** Mô tả hệ thống làm như thế nào? (How to do?)
- **Mức khái niệm (logic):** Mô tả hệ thống làm gì?(What to do?); ở đây không nói đến biện pháp công cụ...

Hình thức biểu diễn : Trong một số tài liệu khác nhau với các phương pháp tiếp cận khác nhau (MEIN, SSADM) người ta thường dùng các kí hiệu không hoàn toàn giống nhau. Tuy vậy các thành phần cơ bản không thay đổi và nó được sử dụng nhất quán trong các quá trình phân tích, thiết kế

d. Các thành phần của biểu đồ

Mỗi biểu đồ luồng dữ liệu gồm 5 thành phần :

- Chức năng xử lý (Process)
- Luồng thông tin (Data Flows)
- Kho dữ liệu (Data Store)
- Tác nhân ngoài (External Entity)
- Tác nhân trong (Internal Entity)

1. Chức năng xử lý (Process)

- **Khái niệm:** Chức năng xử lý là chức năng biểu đạt các thao tác, nhiệm vụ hay tiến trình xử lý nào đó. Tính chất quan trọng của chức năng là biến đổi thông tin. Tức là nó phải làm thay đổi thông tin từ đầu vào theo một cách nào đó như tổ chức lại thông tin, bổ sung thông tin hoặc tạo ra thông tin mới
- **Biểu diễn:** Chức năng xử lý được biểu diễn bằng đường tròn hay ô van, trong đó có ghi nhãn (tên) của chức năng. Việc dùng kí hiệu đường tròn chỉ là qui ước, được kế thừa từ các phương pháp luận dựa trên tiến trình trước đây. Nhiều phương pháp luận đã chấp nhận những kí hiệu khác cho mục đích này chẳng hạn như hình chữ nhật hay hình vuông tròn các góc tiện lợi cho soạn thảo văn bản. Bởi vậy khi tham khảo các tài liệu khác ta nên chú ý; còn trong tài liệu này ta sử dụng nhất quán kí hiệu đường tròn
- **Nhãn (tên) chức năng:** Bởi vì chức năng là các thao tác nên tên phải được dùng là một "Động từ" cộng với "bổ ngữ". Chú ý rằng trong tiếng Việt động từ và danh từ đôi khi chung một từ nên cần thiết ta phải thêm từ xác định "sự" nếu muốn nhấn mạnh đó là danh từ.

Ví dụ : Chức năng "Ghi nhận hoá đơn", "Theo dõi mượn trả", "Xử lý thi lại"



2. Luồng dữ liệu

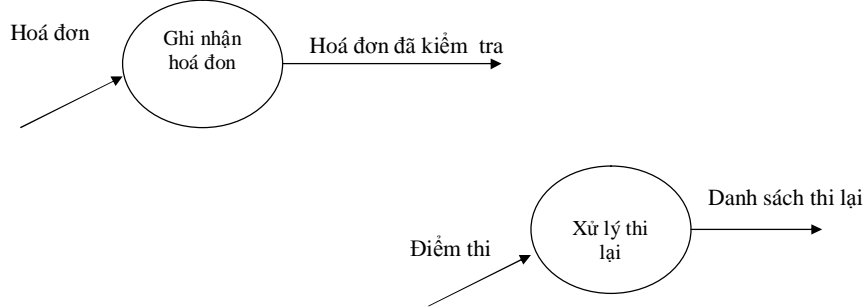
- **Khái niệm:** Luồng dữ liệu là luồng thông tin vào hay ra của một chức năng xử lý. Bởi vậy luồng dữ liệu được coi như các giao diện giữa các thành phần của biểu đồ
- **Biểu diễn:** Luồng dữ liệu trên biểu đồ được biểu diễn bằng mũi tên có hướng trên đó có ghi tên nhãn là tên luồng thông tin mang theo. Mũi tên để chỉ hướng của luồng thông tin

- **Nhãn (tên) luồng dữ liệu:** Vì thông tin mang trên luồng, nên tên là “danh từ” cộng với “tính từ” nếu cần thiết.

Thí dụ “Hoá đơn”, “Hoá đơn đã kiểm tra”, “Điểm thi”, “Danh sách thi lại”

Các luồng dữ liệu và tên được gán cho chúng là các thông tin “logic” chứ không phải là các tài liệu vật lý

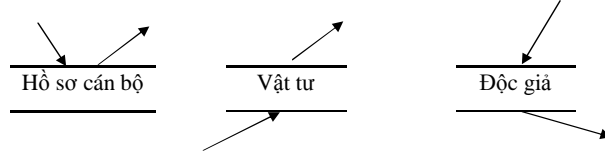
Thí dụ về chức năng xử lý và luồng dữ liệu tương ứng



3. Kho dữ liệu

- **Khái niệm:** Kho dữ liệu là các thông tin cần lưu giữ lại trong một khoảng thời gian, để sau đó một hay một vài chức năng xử lý, hoặc tác nhân trong sử dụng. Nó bao gồm một nghĩa rất rộng các dạng dữ liệu lưu trữ: Dưới dạng vật lý chúng có thể là các tài liệu lưu trữ trong văn phòng hoặc các file trên các thiết bị mạng tin (băng từ, đĩa từ) của máy tính; nhưng ở đây ta quan tâm đến thông tin chứa trong đó tức là dạng logic của nó (trong cơ sở dữ liệu)
- **Biểu diễn:** Kho dữ liệu được biểu diễn bằng hình chữ nhật hở hai đầu hay (cặp đoạn thẳng song song) trên đó ghi nhãn của kho.
- **Nhãn:** Bởi vì kho chứa các dữ liệu nên tên của nó là danh từ kèm theo tính từ nếu cần thiết, nó nói lên nội dung thông tin chứ không phải là giá mang thông tin

Thí dụ : Kho “Hồ sơ Cán bộ”, “Vật tư”, “Phòng”, “Độc giả”

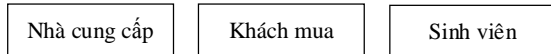


4. Tác nhân ngoài: Người ta còn gọi là Đối tác (External Entities) là một người, nhóm hay tổ chức ở bên ngoài lĩnh vực nghiên cứu của hệ thống nhưng đặc biệt có một số hình thức tiếp xúc, trao đổi thông tin với hệ thống. Sự có mặt các nhân tố này trên sơ đồ chỉ ra giới hạn của hệ thống, và định rõ mối quan hệ của hệ thống với thế giới bên ngoài. Điều đáng chú ý là hiểu nghĩa “ngoài lĩnh vực nghiên cứu” không có nghĩa là bên ngoài tổ chức, chẳng hạn như đối với hệ thống xử lý đơn hàng thì bộ phận kế toán, bộ phận mua hàng và các bộ phận kho tàng vẫn là tác nhân ngoài. Đối với hệ thống tuyển sinh đại học thì tác nhân ngoài vẫn có thể là thí sinh, giáo viên chấm thi và hội đồng tuyển sinh.

Tác nhân ngoài là phần sống còn của hệ thống, chúng là nguồn cung cấp thông tin cho hệ thống cũng như chúng nhận các sản phẩm thông tin từ hệ thống

- **Biểu diễn:** Bằng hình chữ nhật, có gán nhãn.
- **Nhãn (tên):** Được xác định bằng danh từ kèm theo tính từ nếu cần thiết

Thí dụ :



5. Tác nhân trong

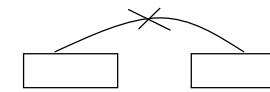
- **Khái niệm:** Tác nhân trong là một chức năng hay một hệ thống con của hệ thống được mô tả ở trang khác của biểu đồ. Thông thường mọi biểu đồ có thể bao gồm một số trang, đặc biệt là trong các hệ thống phức tạp và với khuôn khổ giấy có hạn thông tin được truyền giữa các quá trình trên các trang khác nhau được chỉ ra nhờ ký hiệu này. Ý nghĩa của tác nhân trong với kí hiệu tương tự như nút tiếp nối của sơ đồ thuật toán.

- **Biểu diễn:** Tác nhân trong biểu diễn bằng hình chữ nhật hở một phía và trên có ghi nhãn.
- **Nhãn (tên) tác nhân trong:** Được biểu diễn bằng Động từ kèm bổ ngữ

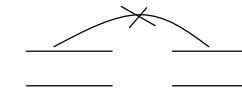


Một số chú ý khi xây dựng biểu đồ BLD :

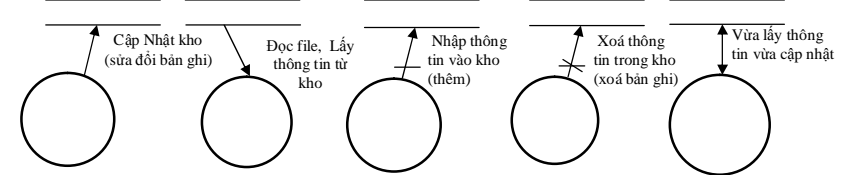
- Trong biểu đồ không có hai tác nhân ngoài trao đổi trực tiếp với nhau



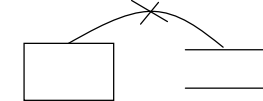
- Không có trao đổi trực tiếp giữa hai kho dữ liệu mà không thông qua chức năng xử lý.



- Nói chung kho đã có tên nên luồng dữ liệu vào ra kho không cần tên, chỉ khi việc cập nhật, hoặc trích từ kho chỉ một phần thông tin ở kho, người ta mới dùng tên cho luồng dữ liệu.
- Vì lí do trình bày nên tác nhân ngoài, tác nhân trong và kho dữ liệu sử dụng nhiều lần có thể vẽ được vẽ lại ở nhiều nơi trong cùng biểu đồ để cho dễ đọc, dễ hiểu hơn
- Mối liên quan giữa chức năng xử lý, kho dữ liệu và luồng dữ liệu :



- Đối với kho dữ liệu phải có ít nhất một luồng vào và ít nhất một luồng ra. Nếu kho chỉ có luồng vào mà không có luồng ra là kho “vô tích sự”, nếu kho chỉ có luồng ra mà không có luồng vào là kho “rỗng”
- Tác nhân ngoài không trao đổi với kho dữ liệu mà phải thông qua chức năng xử lý



Thí dụ (Case Study): Sau đây ta xét một ví dụ tổng quát, thí dụ này sẽ có đầy đủ các đặc thù được xem xét các khía cạnh xuyên suốt trong các chương về sau

Hệ thống cung ứng vật tư của nhà máy X.

Nhà máy X bao gồm các phân xưởng, sản xuất một số sản phẩm nhất định. Trong quá trình sản xuất các phân xưởng sử dụng vật tư. Nhà máy có bộ phận quản lý cung ứng vật tư. Hiện tại hệ thống gồm có 2 bộ phận tách rời: Mua hàng (ĐH) và Tiếp nhận hàng, Phát hàng (PH)

Hai bộ phận này đã lập riêng hai hệ thống xử lý trên 2 máy tính và 2 máy tính này không tương thích nên không nối với nhau được. Cấu trúc tương ứng của 2 bộ phận là

a) Hệ đặt hàng (ĐH) nhằm giải quyết các dự trữ vật tư của các phân xưởng

- Chọn người cung ứng
- Thương lượng với nhà cung cấp
- Lập đơn hàng (SH -đơn)
- Sao lưu đơn hàng và cất trong file “Đơn hàng”.

File sử dụng : “Người cung cấp” chứa thông tin về người cung cấp với các thông tin cần quản lý: Mã người cung cấp, Tài khoản, Địa chỉ, Điện thoại, Các mặt hàng và khả năng cung cấp.

Chú ý :

- Mỗi bản dữ trữ vật tư có thể đáp ứng bởi những người cung cấp khác nhau. Tuy nhiên mỗi mặt hàng trên một bản dự trữ chỉ do một người cung cấp cung ứng.

- Mỗi đơn hàng lại có thể chứa nhiều mặt hàng do nhiều phân xưởng tiêu thụ yêu cầu, lưu ý rằng trên đơn hàng không có lưu thông tin nơi người dự trữ vì vậy cần lưu thông tin Dự trữ - Đơn hàng (DT/DH)
- Hệ Phát hàng (PH):** Theo dõi hàng từ khi nhận về, nhập vào kho đến khi phát hàng về phân xưởng
- Hàng về kèm phiếu giao hàng: Thông tin trên phiếu giao hàng kèm theo nơi cất (tạm) hàng lưu ở file "Nhận hàng". Thông tin trên phiếu giao hàng không lưu thông tin người sử dụng hàng
- Bộ phận thủ công: Làm nhiệm vụ đối chiếu, các công việc tiến hành như sau:
 - Hàng ngày bộ phận thu hàng nhận hàng, in các danh sách hàng nhận về gửi đến bộ phận đối chiếu, trong danh sách đều có ghi SH- đơn
 - Đối chiếu SH-đơn để tìm địa chỉ phát hàng để bộ phận nhận hàng phát cho nơi nhận
 - Đối chiếu nhận hoá đơn với danh sách hàng về, nếu khớp chuyển cho tài vụ để trả tiền, nếu không khớp thì trao đổi về các bất nhất giữa Đơn hàng-Nhận hàng-Hoá đơn (ĐH/NH/HD)

Việc vẽ biểu đồ luồng dữ liệu BLD có thể vẽ ở các mức độ thô hay tinh dần

Ban đầu căn cứ vào 4 chức năng chính:

- Đặt hàng
- Nhận, phát hàng
- Đối chiếu
- Trả tiền

Tác nhân ngoài :

- Phân xưởng
- Người cung cấp

*SHđơn - SHMH - SH Dự trữ

*SHGH - SHMH - SH Đơn hàng.

2.4. Các thể hiện khác của biểu đồ luồng dữ liệu:

Trên đây ta đã nghiên cứu 2 phương pháp biểu diễn chức năng xử lý của hệ thống: Biểu đồ phân cấp chức năng (BPC) và Biểu đồ luồng dữ liệu (BLD)

Các phương pháp này đôi khi chưa sáng tỏ với thực tế vì thực chất mô hình còn giản lược, chưa lột tả hết các khía cạnh chi tiết của quá trình

Bởi vậy cần thiết phải đưa ra một số khái niệm để trừu tượng hoá, lấy được bản chất của vấn đề

a) Sự đồng bộ hoá:

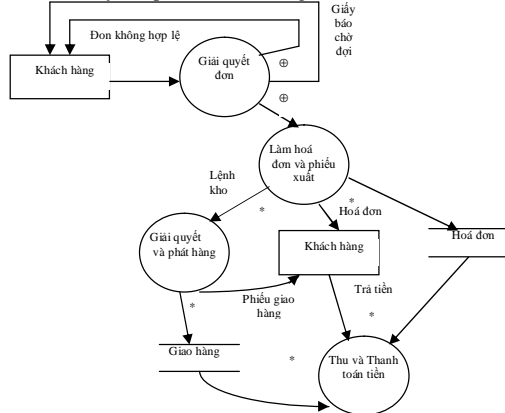
Sự đồng bộ hoá thể hiện quá trình diễn ra đồng thời hoặc lựa chọn của các dòng dữ liệu vào hoặc ra từ các chức năng xử lý. Để thực hiện điều này ta bổ sung một số kí hiệu bên cạnh luồng dữ liệu

Kí hiệu : * và (AND)

hoặc loại trừ (XOR)

() hoặc không loại trừ (OR)

Thí dụ : Hãy xét phân hệ bán hàng



b) Phương pháp của MERISE: Mô tả chi tiết các chức năng

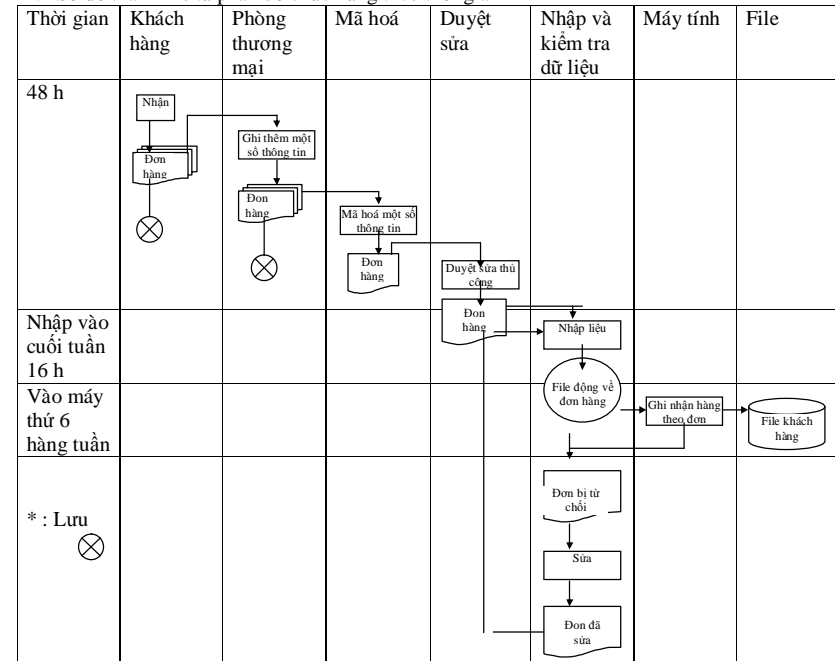
Biểu đồ luồng dữ liệu BLD chỉ giới hạn mô tả các chức năng trong tiến trình xử lý nhưng chưa diễn tả thời gian và địa điểm thực hiện. Phương pháp Merise cho rằng như vậy không đủ cần xây dựng bảng gồm các công việc và thời gian phân bổ thực hiện, làm mịn hoá tiến trình xử lý (xem hình 2.1).

c) Sơ đồ công việc theo theo các thanh:

Đây là phương pháp để mô tả thô các công việc theo bảng. Với cột chỉ thời gian và hàng chỉ các công việc cv_k. Các thanh xác định công việc cv_k từ thời điểm t_i tới t_j.

	t ₁	t ₂			t _{n-1}	t _n
cv ₁ .	█					
cv ₂ .		█				
cv ₃ .			█			
					█	
cv _k						█

Hình 2.2 Sơ đồ thanh mô tả phân bổ chức năng theo thời gian



Hình 2.1 Tiến trình thực hiện chi tiết các công việc

d) Đưa thêm các kí hiệu vật lý vào biểu đồ:

Để làm rõ các các chức năng và phân biệt các giá thông tin đối với các nguồn dữ liệu và kho dữ liệu ta đưa thêm các kí hiệu và các quy ước dùng trong giáo trình. Đây là các qui định của hãng IBM, tuy các kí hiệu này tương đối cổ điển nhưng ngày nay người ta vẫn dùng do thói quen và tính trực quan của nó. Các kí hiệu này có tác dụng khi ta muốn chi tiết thêm các biểu đồ.

Biểu diễn thông tin

Thông tin tổng quát



Đĩa từ

Hồ sơ / Kết xuất



Bảng từ

Tài liệu in



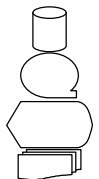
Màn Hình

Trống từ

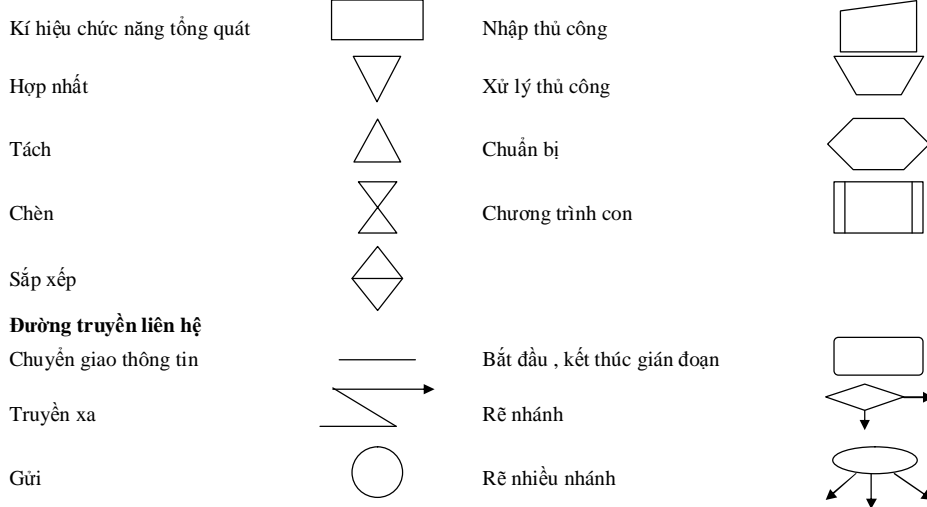


Tài liệu vào

Đĩa mềm



Biểu diễn xử lý



Hình 2.3 Các kí hiệu bổ sung của IBM

2.4. Đặc tả các chức năng

a) *Khái niệm về đặc tả:* Trong biểu đồ phân cấp chức năng BPC, biểu đồ luồng dữ liệu BLD, các chức năng dù có chi tiết đến đâu (tới mức không phân nhỏ được nữa) cũng chỉ xác định nhờ tên của nó.

Quá trình phân tích từ trên xuống dưới, với mục đích phân rõ dần từng bước sẽ ngừng ở một mức nào đó vì có phân tích sâu thêm sẽ vượt qua câu hỏi "Hệ thống là gì" để lần sang giai đoạn thiết kế trả lời câu hỏi "Hệ thống như thế nào" hoặc là chức năng thu được đã đơn giản tới mức có thể mô tả và lời là rõ.

Bởi vậy cần thiết các chức năng có thể được mô tả một cách chi tiết (mức mô tả thấp nhất) hơn bằng một số phương pháp khác gọi là đặc tả chức năng P-Spec (Process Specification).

Một đặc tả gồm 2 phần (thường không quá 1 trang A4):

- Phần đầu đề:
 - Tên chức năng
 - Các dữ liệu vào
 - Các dữ liệu ra

Phần thân: Mô tả nội dung xử lý.

b) *Các phương tiện có thể sử dụng để đặc tả*

- Các biểu đồ, lược đồ, sơ đồ khối
- Các phương trình toán học
- Các bảng, cây quyết định
- Các ngôn ngữ tự nhiên cấu trúc hoá

1. Phương pháp đặc tả bằng sơ đồ khối (Flow Chart, Diagram): Phương pháp này khá cổ điển nhưng trực quan và thường áp dụng cho các hệ thống đơn giản. Một sơ đồ khối gồm: Các khối bắt đầu, kết thúc, Thao tác, rẽ nhánh, và khối vòng lặp. Phần này chúng ta có thể tham khảo trong phần tin học đại cương.

2. Phương pháp đặc tả bằng ngôn ngữ có cấu trúc (Pseudo Code): Đây là ngôn ngữ đặc tả hay còn gọi là ngôn ngữ giả trình vì nó rất gần với ngôn ngữ lập trình và chuyển đổi sang ngôn ngữ lập trình một cách dễ dàng. Ngôn ngữ giả trình được đặc tả bằng lời thông qua một ngôn ngữ nào đó với cú pháp không chặt chẽ để diễn tả các bước với các hành động cơ sở (Primitive Actions), cấu trúc tuần tự (Sequences), lựa chọn (Selections), và thao tác lặp (Iterations). Tuy nhiên ta không nên dùng ngôn ngữ tự do

Thí dụ: Cấu trúc lựa chọn IF ... THEN

```

READ-FILE STOCK-DETAILS
IF < điều kiện >
    < hành động >
ELSE
    < hành động >
    
```

Cấu trúc đa lựa chọn

```

CASE
WHEN < điều kiện > < hành động >
    WHEN < điều kiện > < hành động >
    ...
    ...
Cấu trúc lặp
DO WHILE < điều kiện >
    < hành động >
    
```

```

REPEAT
    < hành động >
UNTIL < điều kiện >
    
```

3. Những qui định và qui tắc về quản lí: Các quy định được thể hiện qua các công thức tính toán, các phép biến đổi.

Thí dụ

- Qui định tính lãi suất tín dụng và tiền gửi
- Qui định cách tính lương,
- Qui định đánh thuế thu nhập cao
- Qui định tính điểm trung bình chung học tập

4. Phương pháp đặc tả sử dụng bảng quyết định: Bảng quyết định là bảng biểu diễn các điều kiện, các hành động và dưới điều kiện nào thì hành động sẽ được tiến hành. Bảng thường phân thành các trường hợp một cách rành rẽ và không bỏ sót các trường hợp. Bảng quyết định gồm bốn góc một phần tư có dạng sau:

Các điều kiện có thể xảy ra	Các qui tắc áp dụng
Các hành động có thể có	Các hành động xảy ra

Thí dụ: Bài toán phát biểu như sau:

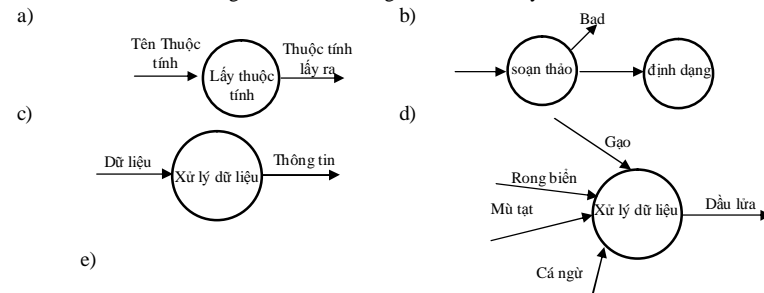
Giả sử có 3 người tù đi làm khổ sai với tên tương ứng A,B,C. Ông cai tù đặt điều kiện: Có 5 cái mũ, gồm 2 mũ trắng (T), và 3 mũ đỏ (Đ). Mỗi người chỉ xem được 2 mũ trên đầu 2 người kia. Hãy đoán xem mình đội mũ màu gì? Nếu người nào đoán đúng màu thì được thưởng, không phải đi làm, nếu đoán sai thì ăn đòn và vẫn phải đi làm, và nếu không đoán thì đi làm bình thường

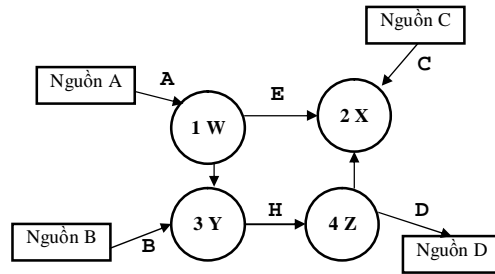
- A : thua, xin không đoán
- B : thua, xin không đoán
- C : (mũ) đoán được ()

Trường hợp	Người			Hành động
	A	B	C	
1	T	T	T	Không có 3 mũ trắng
2	T	T	Đ	
3	T	Đ	T	B, loại
4	T	Đ	Đ	
5	Đ	T	T	A, loại
6	Đ	T	Đ	
7	Đ	Đ	T	B loại
8	Đ	Đ	Đ	

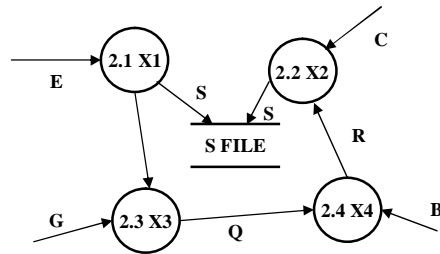
Bài tập Chương 2

2.1 Tìm chỗ sai trong các biểu đồ luồng dữ liệu dưới đây:

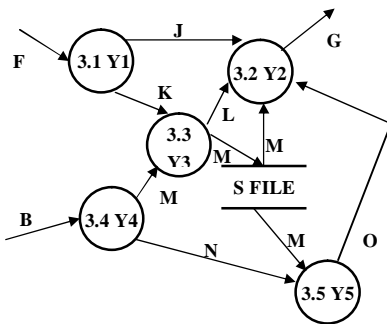




f)



g)



2.2. Tại sao luồng dữ liệu vào/ra từ kho dữ liệu đôi khi không có tên ?

2.3. Chức năng sơ cấp là gì? Trong BLD, chức năng sơ cấp đòi hỏi điều gì mà thành phần khác không nhất thiết phải có?

2.4. Trong biểu đồ luồng dữ liệu có khi nào không có tác nhân ngoài không? Tại sao?

2.5. Trong biểu đồ luồng dữ liệu những sai sót nào hay gặp phải. Hãy giải thích?

2.6. Biểu đồ luồng dữ liệu cho ta biết mối quan hệ gì giữa các thành phần của hệ thống?

Chương 3. Các phương tiện và mô hình diễn tả dữ liệu

3.1. Khái niệm diễn tả dữ liệu

Một hệ thống trong trạng thái vận động bao gồm hai yếu tố là các chức năng xử lý và dữ liệu. Giữa xử lý và dữ liệu có mối quan hệ mật thiết chặt chẽ và bản thân dữ liệu có mối liên kết nội bộ không liên quan đến xử lý đó là tính độc lập dữ liệu. Mô tả dữ liệu được xem như việc xác định tên, dạng dữ liệu và tính chất của dữ liệu. Dữ liệu không phụ thuộc vào người sử dụng đồng thời không phụ thuộc vào yêu cầu tìm kiếm và thay đổi thông tin. Trong chương này để thuận tiện cho phương pháp nghiên cứu chúng ta chỉ tập trung đề cập đến các phương tiện và mô hình diễn tả dữ liệu. Đó là các thông tin được quan tâm đến trong quản lý, nó được lưu trữ lâu dài, được xử lý và sử dụng trong hệ thống thông tin quản lý.

Có nhiều công cụ để mô tả dữ liệu. Các công cụ này là các cách trừu tượng hoá dữ liệu đặc biệt là mối quan hệ của dữ liệu nhằm phổ biến những cái chung nhất mà con người ta có thể trao đổi lẫn nhau. Trong phần này chúng ta đề cập tới 4 công cụ chủ yếu:

- Mã hoá dữ liệu (coding)
- Từ điển dữ liệu (Data Dictionary)
- Mô hình thực thể liên kết ER (Entity- Relationship)
- Mô hình quan hệ (Relational Data Base Modeling)

3.2. Sự mã hoá

a) *Khái niệm mã hoá:* Mã là tên viết tắt gán cho một đối tượng nào đó hay nói cách khác mỗi đối tượng cần có tên và vấn đặt ra ta sẽ đặt tên cho đối tượng như thế nào. Trong mỗi đối tượng gồm nhiều thuộc tính khác nhau thì yêu cầu mã hoá cho các thuộc tính cũng là yêu cầu cần thiết. Ngoài ra mã hoá còn là hình thức chuẩn hóa dữ liệu và bảo mật dữ liệu đặc biệt trong các hệ thống thông tin xử lý bằng máy tính

Một số thí dụ về mã hóa: Khi ta cần xác định một công dân thì số chứng minh thư hoặc số hộ chiếu là mã của công dân đó. Khi cần xác định xe ô tô hay xe máy thì biển số xe là mã của xe đó.

b) *Chất lượng của việc mã hoá:* Trong thực tế ta gặp rất nhiều đối tượng cần mã hoá như mã hoá ngành nghề đào tạo, mã hoá các bệnh, mã số điện thoại, mã thẻ sinh viên, thẻ bảo hiểm y tế,... Chúng ta có nhiều phương pháp mã khác nhau. Do vậy cần xác định một số tiêu chí để đánh giá chất lượng của việc mã hoá:

- **Mã không được nhập nhằng:** Thể hiện ánh xạ 1 - 1 giữa mã hoá và giải mã, mỗi đối tượng được xác định rõ ràng với một mã nhất định
- **Thích ứng với phương thức sử dụng:** Việc mã có thể tiến hành bằng thủ công nên cần phải dễ hiểu, dễ giải mã, và việc mã hoá bằng máy đòi hỏi cú pháp chặt chẽ.
- **Có khả năng mở rộng mã:**
 - Thêm phía cuối (sau) của các mã đã có
 - Xen mã mới vào giữa các mã đã có, thường mã xen phải dùng phương pháp cóc nhảy, nhảy đều dựa vào thống kê để tránh tình trạng “bùng nổ” mã. Chẳng hạn như mã dòng lệnh trong ngôn ngữ lập trình BASIC
- **Mã phải ngắn gọn làm giảm kích cỡ của mã:** đây cũng là mục tiêu của mã hoá. Tuy nhiên điều này đôi khi mâu thuẫn với khái niệm mở rộng mã sau này.
- **Mã có tính gợi ý:** Thể hiện tính ngữ nghĩa của mã. Đôi khi tính gợi ý là yêu cầu đối với mã công khai, và làm cho việc mã hoá thuận tiện dễ dàng

c) *Các kiểu mã hoá:*

(1) **Mã hoá liên tiếp:** Ta dùng các số nguyên liên tiếp 000, 001, 002. . . để mã hoá. Phương pháp này thường để đánh số thứ tự trong danh sách các đối tượng

Ưu điểm: Không nhập nhằng, đơn giản, thêm phía sau

Khuyết điểm: Không xen được, thiếu tính gợi ý vì cần phải có bảng tương ứng và không phân theo nhóm

(2) **Mã hoá theo lát:** Sử dụng các số nguyên như mã hoá liên tiếp nhưng phân ra từng lát (lớp) cho từng loại đối tượng, trong mỗi lát dùng mã liên tiếp

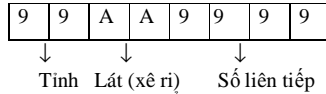
Thí dụ: Mã hoá Ngũ kim

Vùng 1	0001 - 0999	ngũ kim bé
Vùng 2	0001 - 0099	vít
Vùng 3	0100 - 0299	ê cu
	0300 - 0499	bulong
	0500 - 0599	đinh
	1000 - 1999	chi tiết kim loại
Vùng n	1000 - 1099	sắt U

Ưu điểm: Không nhập nhằng, đơn giản, có thể mở rộng xen thêm được

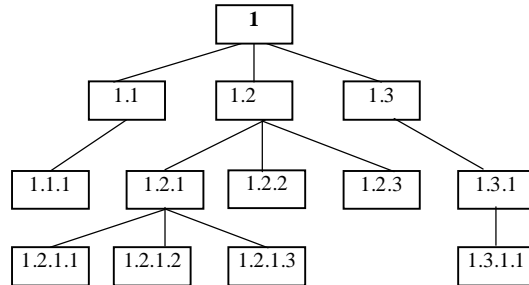
Nhược điểm: Thiếu gợi ý

(3) **Mã phân đoạn:** Bản thân mã được phân thành nhiều đoạn mỗi đoạn mang một ý nghĩa riêng
 Thí dụ: Số đăng kí xe máy



Thí dụ : Biển số xe máy của ông X là 29 F6 696 là biển xe đăng kí tại Hà nội (mã tỉnh là 29)
Ưu điểm: Không nhập nhằng, mở rộng, xen thêm được và được dùng khá phổ biến. Loại mã này cho phép thiết lập các phương thức kiểm tra gián tiếp đối với mã của các đối tượng bằng cách trích rút các đoạn mã để kiểm tra
Nhược điểm: Mã quá dài nên thủ tục mã nặng nề, không cố định và vẫn có thể bị bão hoà mã
(4) Mã phân cấp: Các đối tượng được mã hoá theo chế độ phân cấp các chi tiết nhỏ dần. Một hình ảnh khá quen thuộc của mã hoá phân cấp là đánh số chương, tiết, mục trong một quyển sách.

1. Chương 1
 - 1.1 Bài 1
 - 1.2 Bài 2
2. Chương 2
 - 2.1 Bài 3
 - 2.1.1 Mục 1
 - 2.1.2 Mục 2
 - 2.2 Bài 4
 - 2.3 Bài 5



Ưu điểm: Các ưu điểm tương tự như mã hoá phân đoạn. Ngoài ra việc tìm kiếm mã dễ dàng
Khuyết điểm: Tương tự các nhược điểm của mã kiểu phân đoạn
(5) Mã điển nghĩa: Bằng cách gán một tên ngắn gọn nhưng hiểu được cho một đối tượng
 Thí dụ : Đội bóng các nước tham gia giải Tiger cup được mã bằng cách lấy 3 kí tự đầu như sau
 VIE : Vietnam, THA: Thailand, SIN : Singarpore, IND: Indonesia, MAL: Malaysia .

Ưu điểm: Tiện dùng cho xử lí bằng thủ công và số lượng đối tượng được mã ít
Khuyết điểm: Không giải mã được bằng máy tính.

d) **Cách lựa chọn sự mã hoá:**

Có nhiều phương pháp mã hoá khác nhau, có thể sử dụng kết hợp nhiều kiểu để đạt chất lượng mã tốt nhất.
 Việc lựa chọn mã hoá cần dựa vào các yếu tố sau:

- Nghiên cứu việc sử dụng mã sau này
- Nghiên cứu số lượng các đối tượng được mã hoá để lường trước được sự phát triển
- Nghiên cứu sự phân bố thống kê các đối tượng để phân bố theo lớp
- Tìm xem đã có những mã hoá nào được dùng trước đó cho các đối tượng này để kế thừa
- Thỏa thuận người dùng cách mã
- Thử nghiệm trước khi dùng chính thức để chỉnh lý kịp thời

3.3. Từ điển dữ liệu

a) **Khái niệm:** Từ điển dữ liệu (còn gọi từ điển yêu cầu) là bộ phận của tư liệu trong phân tích thiết kế, nó là vận phạm giả hình thức mô tả nội dung của các sự vật, đối tượng theo định nghĩa có cấu trúc. Trong biểu đồ luồng dữ liệu (BLD) 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.

b) **Cấu tạo từ điển:** 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. Kí pháp mô tả nội dung cho từ điển dữ liệu tuân theo bảng sau:

Kết cấu dữ liệu	Ký pháp	Ý nghĩa
Tuần tự	=	được tạo từ
Tuyển chọn	+	và
Lặp	[] { } ⁿ ()	hoặc Lặp n lần dữ liệu tùy chọn

	* Lờn chú thích *	giới hạn chú thích
--	-------------------	--------------------

Thí dụ : Giả sử có tờ hoá đơn bán hàng như sau

Số HD: 123		HOÁ ĐƠN BÁN HÀNG		Ngày: 01-01-99			
Bán cho ông/bà : Trần Tĩnh Mịch		Đơn vị		Tài khoản : LTM010254			
Địa chỉ : 534 Hàng Mành,		Đơn giá		Tel : (04) 8226465/8692205			
Số TT	Mã hàng	Tên, quy cách	Đơn vị	Đơn giá	Số lượng	Thành tiền	Ghi chú
1	X30	Xi măng	bao	47000	200	9400000	
2	Y10	Quạt thông gió	chiếc	100000	6	600000	
3	Z20	Nồi cao áp	chiếc	2500000	1	2500000	
4	X10	Đinh 20 phân	kg	5500	100	550000	
Tổng cộng							
Bảng chữ							
<i>Kế toán trưởng</i>			<i>Người nộp tiền</i>		<i>Người bán hàng</i>		

Ta có một phần tử điển dữ liệu sau

* **Xác định một tờ hoá đơn như sau***

Hoá đơn = Số HD + Ngày bán + Khách hàng +

{ + Hàng
+ Số lượng
+ 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

Một ví dụ khác là từ điển xác định số điện thoại:

Số điện thoại = [số máy phụ | số bên ngoài]

Số máy phụ = [2001 | 2002 | ... | 2999]

Số bên ngoài = 9 + [số nội hạt | số đường dài]

Số nội hạt = số đầu + số thâm nhập

Số đường dài = (1) + mã vùng + số nội hạt

Số đầu = [795 | 799 | 874 | 877]

Số thâm nhập = *Bất kỳ xấu bốn chữ số*

3.4) Mô hình thực thể liên kết

a) **Khái niệm:** Mô hình thực thể liên kết là công cụ thành lập lược đồ dữ liệu hay gọi là biểu đồ cấu trúc dữ liệu (BCD), nhằm xác định khung khái niệm về các thực thể, thuộc tính, và mối liên hệ ràng buộc giữa chúng. Mục đích của mô hình xác định các yếu tố:

- Dữ liệu nào cần xử lý
- Mối liên quan nội tại (cấu trúc) giữa các dữ liệu

b) **Thực thể và kiểu thực thể**

Thực thể là một đối tượng được quan tâm đến trong một tổ chức, một hệ thống, nó có thể là đối tượng cụ thể hay trừu tượng. Thực thể (theo Mein II, 1991) phải tồn tại, cần lựa chọn có lợi cho quản lí và phân biệt được

Thí dụ : Các khách hàng đều có tài khoản của họ và các nhà cung cấp cung cấp các mặt hàng, ở đây các đối tượng được quan tâm:

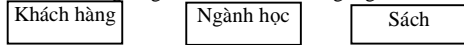
<i>Tài khoản</i>	}	là đối tượng cụ thể
<i>Khách hàng</i>	}	Đối tượng trừu tượng
<i>Nhà cung cấp</i>		
<i>Mặt hàng</i>		

Để định nghĩa một cách chính xác hơn ta đưa ra khái niệm: Kiểu thực thể (entity type) và thể hiện thực thể (entity instance).

Kiểu thực thể là tập hợp các thực thể hoặc một lớp các thực thể có cùng đặc trưng cùng bản chất. Thể hiện thực thể là một thực thể cụ thể, nó là một phần tử trong tập hợp hay lớp của kiểu thực thể. Sau này trong các ứng dụng để tránh sử dụng nhiều khái niệm ta đồng nhất thực thể và kiểu thực thể.

Thí dụ : ông "Nguyễn Văn Bích", Hoá đơn số "50", Mặt hàng "X30" là các thực thể cụ thể. Nhưng "Khoa Công nghệ thông tin", "Ngành xử lý nước thải" là các thực thể trừu tượng vì ta không xác định rõ ràng các tiêu chuẩn của nó

Với các thực thể nêu trên ta có kiểu thực thể tương ứng : Khách hàng, hoá đơn, hàng, khoa, ngành
 Biểu diễn thực thể : Kiểu thực thể được biểu diễn bằng hình hộp chữ nhật trong đó ghi nhãn tên kiểu thực thể
 Giả sử ta có các kiểu thực thể tương ứng các nhân khách hàng, ngành học, sách

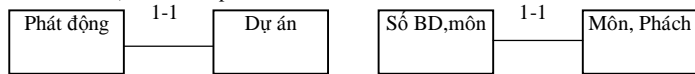


Nhận xét : Trong một bảng dữ liệu ta hình dung cả bảng là kiểu thực thể, mỗi dòng ứng với các bản ghi là thể hiện thực thể, các cột ứng với các thuộc tính của thực thể.

c) Liên kết và kiểu liên kết

Liên kết là sự kết nối có ý nghĩa giữa hai hay nhiều thực thể phản ánh một sự ràng buộc về quản lí.
 Thí dụ: Ông Nguyễn Văn An làm việc ở phòng tài vụ; Hoá đơn số 50 gửi cho khách hàng Lê Văn ích; Sinh viên Trần tinh Mịch thuộc lớp Tin
 Kiểu liên kết là tập các liên kết cùng bản chất. Giữa các kiểu thực thể có thể tồn tại nhiều mối liên kết, mỗi mối liên kết xác định một tên duy nhất.
 Biểu diễn các liên kết bằng đoạn thẳng nối giữa hai kiểu thực thể.
 Các dạng kiểu liên kết : Giả sử ta có các thực thể A,B, C, D ...Kiểu liên kết là sự xác định có bao nhiêu thể hiện của kiểu thực thể này có thể kết hợp với bao nhiêu thể hiện của thực thể kia.

- Liên kết một-một (1-1)** giữa hai kiểu thực thể A, B là ứng với một thực thể trong A có một thực thể trong B và ngược lại. Liên kết này còn gọi là liên kết tầm thường và ít xảy ra trong thực tế, thông thường liên kết này mang đặc trưng bảo mật hoặc cần tách bạch một kiểu thực thể phức tạp thành các kiểu thực thể nhỏ hơn; chẳng hạn một chiến dịch quảng cáo (phát động) cho một dự án; một số báo danh (ứng với một môn thi) có một số phách.

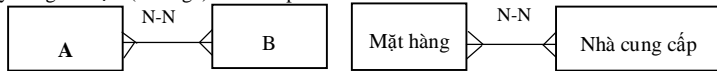


- Liên kết một - nhiều (1-N)** giữa hai kiểu thực thể A, B là ứng với một thực thể trong A có nhiều thực thể trong B và ngược lại ứng với một thực thể trong B chỉ có một thực thể trong A. Liên kết này biểu diễn kết bằng đoạn thẳng giữa hai kiểu thực thể và thêm trạc 3 (hay còn gọi chân gà) về phía nhiều.

Thí dụ : Một lớp có nhiều sinh viên (sinh viên thuộc vào một lớp); Một khách hàng có nhiều tài khoản (tài khoản thuộc về một khách hàng).



- Liên kết nhiều - nhiều (N-N)** giữa hai kiểu thực thể A, B là ứng với một thực thể trong A có nhiều thực thể trong B và ngược lại ứng với một thực thể trong B có nhiều thực thể trong B. Biểu diễn liên kết này bằng ba trạc (chân gà) ở cả hai phía.



Liên kết nhiều nhiều rất khó cài đặt trong các hệ quản trị cơ sở dữ liệu sẵn có. Để dễ biểu diễn người ta dùng phương pháp thực thể hoá bằng cách bổ sung thực thể trung gian để biến đổi liên kết nhiều - nhiều thành hai liên kết một - nhiều



Ở đây A/B là thực thể trung gian giữa A và B, MH/NCC là kiểu thực thể trung gian giữa kiểu thực thể "Mặt hàng" và "Nhà cung cấp"

Biểu diễn: xem liên kết này như một thực thể (thực thể hoá)

Liên kết nhiều bên (nhiều phía): Một kiểu thực thể có thể liên kết với nhiều kiểu thực thể. Liên kết này cũng biểu diễn dưới dạng một thực thể trung gian

d) Các thuộc tính

Định nghĩa Thuộc tính là giá trị thể hiện một đặc điểm nào đó của một thực thể hay một liên kết
 Hoá đơn số : ngày 20/5/94, tổng số tiền 4000000 đồng

Văn phòng phẩm Hồng Hà - bút bi : giá 1000đ/chiếc, đóng gói 10cái
 Thiên Long - bút bi : giá 1100đ/chiếc, đóng gói 12 cái

Kiểu thuộc tính :

- Tên gọi**
- Mô tả:** dữ liệu gắn liền với thực thể là thuộc tính Không khoá
- Kết nối:** nhận diện kiểu thực thể trong thực hệ hay mối liên kết. Thuộc tính kết nối là khoá ở quan hệ này, là mô tả ở quan hệ khác.
- Khoá:** dùng để phân biệt các thực thể hay liên kết nên không được cập nhật

4. Mô hình quan hệ :

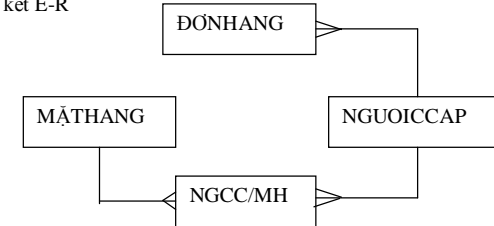
Biểu diễn bằng bảng của các kiểu thực thể liên kết :

- Mỗi bảng 2 chiều là kiểu thực thể. Mỗi cột là một thuộc tính, mỗi dòng là thực thể
- Trong mô hình vẽ chính là thể hiện những đường truy nhập vì nó thể hiện các kết nối và phải trôi theo các mối nối
- Trong mô hình quan hệ khái niệm xuất phát là bảng (file)

Các khái niệm cơ bản : Đại số quan hệ quan niệm theo nghĩa rộng không nhất thiết là thực thể. Các Phụ thuộc hàm, chuẩn hoá 1NF, 2NF, 3NF chúng ta sẽ đề cập chi tiết trong chương sau.

Ví dụ về quan hệ : ĐƠNHANG - MẶTHANG - NGUOICCAP - NGCC/MH

Mô hình thực thể liên kết E-R



Bảng quan hệ thể hiện như sau:

Người cung cấp

SH-NCC	Tên NCC	Địa chỉ	Tài khoản
C300	Hồng Hà	HN	3420
C301	Thiên Long	HCM	4218
A18	Gang thép TNg	TN	2937
A20	Gang thép BHoà	BH	4812

Mặt hàng

SH-MH	Tên Hàng	Mô tả	Đóng gói
425	Bút bi		1000 cái
449	Mực		100 lọ
M97	Sắt Φ8		1 tạ
M70	Sắt Φ20		2 tạ

Người cung cấp -Mặt hàng

SH-NCC	SH-MH	Đơn giá
C300	425	1.000.000
C300	449	40.000
C301	425	1.200.000
A18	M97	800.000

Đơn hàng

SH	HD	Ngày	SH-NCCØ	P t vận chuyển
2142		20/05/99	C300	ô tô
2143		25/05/99	A18	ô tô
2239		15/07/93	C300	Tàu hoả

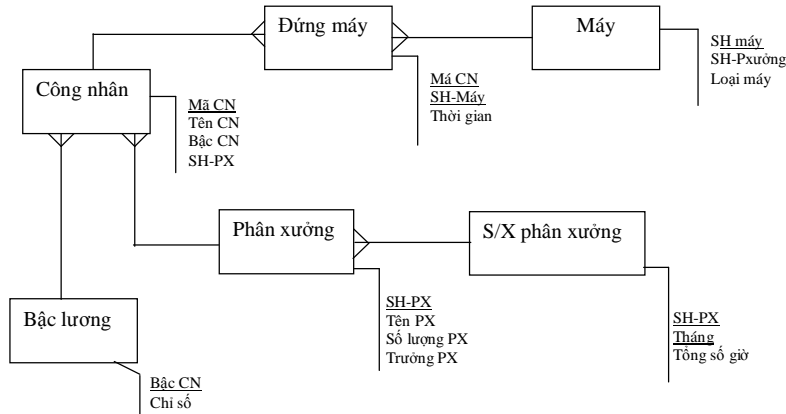
Ví dụ Khi chấm công dùng bảng

(c, t, m, p, r, g)

Công nhân có số hiệu c, tên là t làm trên máy có số hiệu m, ở phân xưởng p mà ông r là trưởng phân xưởng, với số giờ tích lũy là g.

Các thuộc tính có thể thu thập như sau :

Tên thuộc tính	Giải thích ý nghĩa
SH-máy	Số hiệu máy
Loại-máy	Chủng loại
SH-PX	Số hiệu phân xưởng
Tên-PX	Tên phân xưởng
Trưởng-PX	Tên của trưởng phân xưởng
Tên-CN	Tên công nhân
Bậc-CN	Tay nghề của công nhân
Chỉ số	Chỉ số lương cho những bậc thợ
Thời gian	Thời gian làm việc
Tổng số giờ	Số giờ tổng cộng của các máy đã chạy của một phân xưởng
Tháng	Tên của tháng hiện thời



Bài tập chương 3

1. Khách sạn Steak- Acclaim không nhận thanh toán bằng séc cá nhân hay thẻ tín dụng mà chỉ thanh toán bằng tiền mặt hoặc séc luân chuyển (hoặc cả hai loại). Hãy xác định việc thanh toán bằng từ điển dữ liệu.

Thanh toán = ?

2. Trong định nghĩa bằng từ điển sau, hãy chỉ chỗ sai:

Tổng giá của mặt hàng = giá bán + thuế giá trị gia tăng.

3. Xây dựng mô hình thực thể liên kết E-R cho hệ thống quản lý thư viện. Hệ thống gồm các thực thể sau:

- Độ giá
- Sách
- Mượn trả

4. Xây dựng mô hình thực thể liên kết E-R cho hệ thống quản lý sử dụng vận tư của xí nghiệp. Hệ thống gồm các thực thể sau:

- Phân xưởng
- Vật tư
- Sử dụng vật tư

5. Thuật toán tách 1 lược đồ quan hệ thành dạng chuẩn 3NF được phát biểu như sau:

Cho $U = \{ \text{Tập thuộc tính} \}$, $F = \{ \text{Tập phụ thuộc hàm} \}$

Kết luận : $R(U)$ được tách thành $R_1(U_1), R_2(U_2), \dots, R_k(U_k)$. Với R_i ở dạng 3NF.

Phương pháp tách $\rho = (R_1, R_2, \dots, R_k)$.

Thuật toán :

Bước 1: Tìm phủ tối thiểu

a) Tách về phải của các phụ thuộc hàm thành các thuộc tính đơn

$X \rightarrow A_1 A_2$ thì $X \rightarrow A_1$ và $X \rightarrow A_2$

b) Loại bỏ phụ thuộc hàm dư thừa (do tính bắc cầu)

Có nghĩa là : Nếu có $A \rightarrow B, B \rightarrow C$ và $A \rightarrow C$ thì

bỏ $A \rightarrow C$ và chỉ giữ lại $A \rightarrow B, B \rightarrow C$

Bước 2: Nhóm các phụ thuộc hàm cùng về trái (gộp lại):

Nếu có: $X \rightarrow A_1, X \rightarrow A_2 \dots X \rightarrow A_n$ thì $X \rightarrow A_1, A_2 \dots A_n$

Xây dựng các tập $U_i = \{ \text{Tập các thuộc tính không liên quan đến về trái, về phải của mọi phụ thuộc hàm, có nghĩa là không có mặt trong mọi phụ thuộc hàm} \}$ ta gọi là " Các thành thuộc tính bơ vơ "

$U_1 = \{ \text{Tập các thuộc tính trong phụ thuộc hàm 1} \}$

$U_2 = \{ \text{Tập các thuộc tính trong phụ thuộc hàm 2} \}$

$U_3 = \{ \text{Tập các thuộc tính trong phụ thuộc hàm 3} \}$

$U_n = \{ \text{Tập các thuộc tính trong phụ thuộc hàm n} \}$

Với mỗi quan hệ $R_i(U_i)$ xác định khoá K_i

$R_1(U_1)$ xác định khoá $K_1, R_n(U_n)$ xác định khoá K_n

Bước 3 : Tìm khoá tối thiểu K cho tập thuộc tính U_0 nếu có

$K = \{ \text{Hợp các thuộc tính khoá } K_1 K_2 K_r \} / \text{ loại bỏ thuộc tính bắc cầu } \}$

$R_0 = \{ K \cup U_0 \}$

Hãy chuẩn hoá lược đồ quan hệ sau

a) Cho $U = \{ A, B, C, D, E, F, G, H \}$

$F = \{ A \rightarrow CB, C \rightarrow D, EG \rightarrow FH \}$

b) Cho $U = \{ C\#, I, D, B, K, F, L, M, G, \}$

$F = \{ C\# \rightarrow IDBKF, D \rightarrow B, K \rightarrow F \}$

c) Cho $U = \{ A, B, C, D \}$

$F = \{ AB \rightarrow C, D \rightarrow B, C \rightarrow ABD \}$

Chương 4 : Khảo sát hiện trạng và xác lập dự án

4.1 Đại cương giai đoạn khảo sát

a) *Mục đích:* Khảo sát hiện trạng và xác lập dự án là giai đoạn đầu của quá trình phân tích và thiết kế hệ thống (Giai đoạn I của 4 bước phân tích thiết kế cấu trúc). Việc khảo sát thường được tiến hành qua hai giai đoạn:

- Khảo sát sơ bộ nhằm xác định tính khả thi của dự án
 - Khảo sát chi tiết nhằm xác định chính xác những gì sẽ thực hiện và khẳng định những lợi ích kèm theo
- Giai đoạn này còn có tên gọi như "Nghiên cứu tính khả thi (feasibility study)" hoặc "Nghiên cứu hiện trạng (survey of existing system)"

Mục đích cuối cùng của giai đoạn này là "ký kết được hợp đồng thỏa thuận" để xây dựng hệ thống thông tin đối với hệ thống kinh doanh, một tổ chức

b) *Yêu cầu thực hiện của giai đoạn:*

- Khảo sát đánh giá sự hoạt động của hệ thống cũ
- Đề xuất mục tiêu, ưu tiên cho hệ thống mới
- Đề xuất ý tưởng cho giải pháp mới
- Vạch kế hoạch cho dự án

4.2 Tìm hiểu và đánh giá hiện trạng

Tim hiểu và đánh giá hiện trạng nhằm phát hiện những nhược điểm cơ bản của hệ thống cũ, đồng thời cũng định hướng cho hệ thống mới cần giải quyết "cải tạo cái cũ xây dựng cái mới"

a) *Phương pháp khảo sát hiện trạng:*

Các mức khảo sát: cho dù là khảo sát sơ bộ, được phân biệt 4 mức theo thứ tự

- **Thao tác, thừa hành (Tác vụ):** Người sử dụng làm việc trực tiếp với các thao tác của hệ thống và họ thường xuyên nhận ra những khó khăn và những vấn đề nảy sinh ít người được biết. Những công việc này có ảnh hưởng rất lớn do có sự thay đổi các thủ tục và những thay đổi khác kèm theo khi có hệ thống mới
- **Điều phối, quản lý (Điều phối):** Mức giám sát của các những người quản lý trực tiếp. Họ cung cấp thông tin báo cáo tóm tắt định kỳ, các thông tin chi tiết mà họ quản lý tại mọi thời điểm. Tuy nhiên họ không nhìn vấn đề xa được, và không phải là người trực tiếp ra quyết định.
- **Quyết định, lãnh đạo:** Quan sát ở mức tổ chức, lãnh đạo ra quyết định, những ý tưởng mang tính chiến lược phát triển lâu dài quyết định xu hướng phát triển của hệ thống.
- **Chuyên gia cố vấn (Tư vấn):** Mức này bao gồm cố vấn và những người chuyên nghiệp. Vai trò của họ tư vấn về chuyên môn sâu và có thể phê phán hoặc chấp nhận hệ thống. Họ có thể quan trọng hay không tùy thuộc vào đánh giá của mức quyết định.



Mỗi một mức ở trên có vai trò và ảnh hưởng đến hoạt động và sự phát triển chung của hệ thống nên phải được khảo sát đầy đủ

Hình thức khảo sát: Có nhiều hình thức khảo sát, chúng được sử dụng kết hợp để nâng cao hiệu quả, tính xác thực, tính khách quan, tính toàn diện của phương pháp luận:

- Quan sát theo dõi:
 - *Chính thức:* Có chuẩn bị, có thông báo trước
 - *Không chính thức:* Lưu ý rằng với quan sát không chính thức thường cho ta các kết luận khách quan hơn. Quá trình theo dõi có ghi chép và sử dụng các phương pháp để rút ra các kết luận có tính thuyết phục và khoa học
- Phỏng vấn (Interview)
 - Sử dụng các câu hỏi trực tiếp (đóng)
 - Sử dụng các câu hỏi mở
 - Sử dụng các bảng hỏi, mẫu điều tra

a) *Phân loại thông tin:*

Các thông tin thu thập được cần phải phân loại theo các tiêu chí

• *Hiện tại / tương lai:*

- Thông tin cho hiện tại phản ánh chung về môi trường, hoàn cảnh, các thông tin có lợi ích cho nghiên cứu hệ thống quản lý.
- Các thông tin cho tương lai được phát biểu từ các mong muốn, phàn nàn, các dự kiến kế hoạch. Các thông tin cho tương lai có thể có ý thức nhưng không được phát biểu cần được gợi ý hoặc các thông tin vô ý thức cần được dự đoán.

• *Tĩnh / động / biến đổi:*

- Các thông tin tĩnh có thể các thông tin sơ đẳng, cấu trúc hoá, Các phòng ban, chức vụ v.v..
- Các thông tin động thường các thông tin về không gian như các đường đi chuyên tài liệu, về thời gian như thời gian xử lý, hạn định chuyên giao thông tin
- Các thông tin biến đổi: Quy tắc quản lý, các quy định của nhà nước, của cơ quan làm nền cho việc xử lý thông tin. Các thủ tục, những công thức tính toán cũng như các điều kiện khởi động công việc. Các quy trình xử lý v.v...

• *Môi trường / nội bộ:*

- Phân biệt các thông tin của nội bộ hoặc từ môi trường có tác động với hệ thống
- Một điểm đáng lưu ý trong việc phân loại là chú trọng việc đánh giá các tiêu chuẩn như tần suất xuất hiện (điểm đỉnh, điểm trùng), độ chính xác và thời gian sống

b) *Phát hiện các yếu kém của hiện trạng và các yêu cầu cho tương lai:*

Yếu kém:

- **Thiếu, vắng:** Thiếu một chức năng nào đó, thiếu phương tiện xử lý thông tin, thiếu con người thực hiện, quản lý v.v.
- **Kém hiệu lực (hiệu suất thấp)** do các yếu tố
 - Phương pháp xử lý không chặt chẽ
 - Cơ cấu tổ chức bất hợp lý
 - Lưu chuyển thông tin bất hợp lý, dài lòng vòng
 - Giấy tờ tài liệu trình bày kém
 - Sự ùn tắc, quá tải.
- **Tồn phí cao:** Thực chất sự tồn phí cần được đánh giá theo một tiêu chuẩn và khía cạnh nào đó như yếu tố thời gian, con người, quá trình

Yêu cầu này sinh:

- Những nhu cầu về thông tin chưa được đáp ứng
- Các nguyện vọng của nhân viên
- Dự kiến, kế hoạch của lãnh đạo

4.3. Xác định phạm vi, mục tiêu và hạn chế của dự án.

Một hệ thống thông tin thường khá phức tạp mà không thể thực hiện trong một thời gian nhất định bởi vậy cần hạn chế một số ràng buộc để hệ thống mang tính khả thi nhất định. Tại thời điểm này cần xác định các mục tiêu cho dự án, và chính các mục tiêu này là thước đo để kiểm chứng và nghiệm thu dự án sau này.

a. *Phạm vi (Scope):*

Phạm vi là khoảng vùng dự án cần thực hiện với các phương pháp

- *Phương pháp giếng (well) theo chiều sâu:* hạn chế phạm vi hẹp và đi sâu. Phương pháp này dễ nhưng không giải quyết được tổng thể và sau này khó phát triển các hệ con thành nhất thể.
- *Phương pháp hồ (lake) theo chiều rộng:* giải quyết tổng thể, nhất quán, mang tính tập trung hoá cao có định hướng lâu dài.

Trên thực tế thường chọn giải pháp trung hoà cả 2 phương pháp này

b. *Mục tiêu của hệ thống*

- Khắc phục những yếu kém hiện tại
- Đáp ứng những nhu cầu trong tương lai, thể hiện chiến lược phát triển lâu dài của cơ quan
- Thể hiện các hạn chế về thời gian, chi phí, con người

c. *Hạn chế*

- *Tài chính:* Kinh phí cho phép triển khai
- *Con người:* Khả năng quản lý, nắm bắt kỹ thuật mới, khả năng về đào tạo, tác vụ
- *Thiếu bị:* Các kỹ thuật cho phép
- *Môi trường:* Các yếu tố ảnh hưởng về môi trường, xã hội

- **Thời gian:** Các ràng buộc của các hệ thống thời gian hoàn thành, phân phối tài liệu

4.4 Phác họa và nghiên cứu tính khả thi của giải pháp

Sau khi khảo sát và đánh giá sơ bộ hệ thống cũ cũng như đưa ra giải pháp cho hệ thống mới, giai đoạn phác họa tính khả thi cực kỳ quan trọng. Nó quyết định hệ dự án hệ thống này có trở thành hiện thực hay không?.

Phác họa này nhằm vào các điều kiện sau:

- Thoả mãn các yêu cầu bên A (bên chủ đầu tư) hay không?: Thường các yêu cầu này được đưa ra dưới các dạng câu hỏi cốt yếu - TOR (Term of references) mà nhà phân tích cần phải trả lời
- Định hướng giải quyết, thực hiện như thế nào?.
- Thiết bị: Cần đưa ra các chủng loại, tính năng, giá cả, thời gian cung cấp và chúng thường phải dự trữ sớm

Xác định các mức tự động hoá khác nhau:

- Tổ chức lại các hoạt động thủ công
- Tự động hoá một phần, nghĩa là có máy tính tự giúp nhưng không đảo lộn cơ cấu tổ chức
- Tự động hoá làm thay đổi về cơ cấu tổ chức

Phân tích tính hiệu quả và đánh giá tính khả thi: (Chi phí/ lợi ích)

- Khả thi về kỹ thuật
- Khả thi về tác vụ (về xử lý thông tin)
- Khả thi về kinh tế

Tóm lại nhà phân tích thường đưa ra một loạt giải pháp để tiện việc so sánh, đánh giá rồi chọn lựa một giải pháp tối ưu chấp nhận được

4.5. Xét thí dụ (Case Study)

Hệ thống cung ứng vật tư của một xí nghiệp đã được trình bày ở chương trước. Hệ thống gồm phân hệ đặt hàng và mua hàng (gọi tắt hệ ĐH), hệ nhận hàng từ nhà cung cấp và phát hàng tới các phân xưởng dự trữ (gọi tắt là hệ PH) và bộ phận đối chiếu thủ công thanh toán với nhà cung cấp. Qua giai đoạn khảo sát ta đánh giá hiện trạng của hệ thống

Yếu kém:

- Thiếu vắng: Không có kho hàng thông dụng để lưu tạm thời các mặt hàng nhập về và tạm thời chưa sử dụng
- Kém hiệu lực do:
 - o Giải quyết đơn hàng, dự trữ quá chậm do cách đối chiếu thủ công và cách lấy thông tin
 - o Theo dõi việc thực hiện đơn hàng không sát, xảy ra nhiều sai sót do phân tán về quản lý
- Tồn phí: Do quá trình đối chiếu thủ công dễ khớp từ khi dự trữ, đơn hàng, hàng nhận, phiếu giao hàng và hoá đơn thanh toán

Mục tiêu

- Thêm kho hàng thông dụng
- Khắc phục 2 điều kém hiệu lực bằng cách tổ chức lại để rút ngắn quá trình giải quyết 1 dự trữ hàng
- Tổ chức lại để theo dõi thực hiện đơn hàng chặt chẽ, tránh sai sót
- Cố gắng tận dụng phần mềm và phần cứng đã có

Phác họa giải quyết: ở đây ta đưa ra 5 giải pháp để cân nhắc lựa chọn

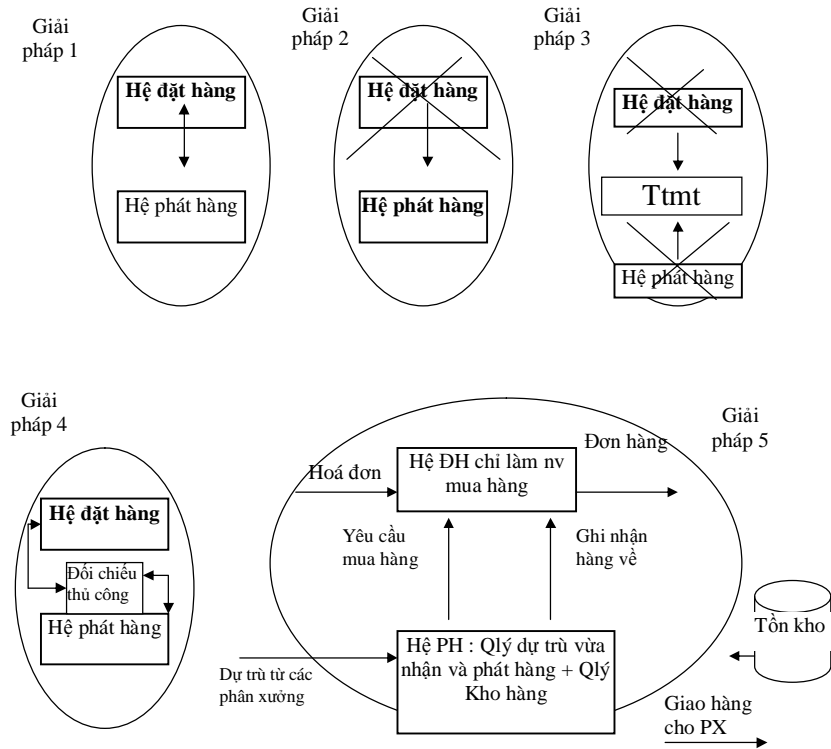
Giải pháp 1: Tạo kênh liên lạc để kết nối hai phân hệ. Giải pháp này vi phạm tính khả thi về kỹ thuật vì giải thiết 2 máy không tương thích

Giải pháp 2: Gộp hệ đặt hàng vào hệ phát hàng hay ngược lại nhằm loại bỏ một máy tính vi phạm thao tác (tác nghiệp)

Giải pháp 3: Loại bỏ 2 máy tính đưa các toàn bộ các nhiệm vụ vào trung tâm máy tính của xí nghiệp. Thực chất của giải pháp này là trang bị máy tính mới, viết lại phần mềm, xử lý tập trung.. Giải pháp này đòi hỏi chi phí lớn, tốn kém hơn. Nó chỉ có lợi khi điều kiện kinh tế cho phép.

Giải pháp 4: Giữ nguyên hiện trạng vốn đang có, vẫn dùng bộ máy tính cũ, chương trình cũ. Thực chất không phát triển hệ thống. Thực chất giải pháp này không có ý nghĩa gì nhưng đôi khi chưa tìm được giải pháp nào hay hơn thì tạm thời chấp nhận.

Giải pháp 5: Chuyển nhiệm vụ nhận dự trữ từ hệ Đặt hàng sang hệ Phát hàng. Như vậy hệ ĐH chỉ làm nhiệm vụ mua hàng. Hệ PH vừa quản lý dự trữ, vừa nhận và phát hàng



Trên đây ta đưa ra 5 giải pháp có tính tương đối và không có chuẩn mực nào cả. Nếu xét chi tiết hơn, nhà phân tích cần thiết phải tính toán cụ thể về nhiều khía cạnh để khẳng định việc lựa chọn một giải pháp và phủ định các giải pháp còn lại

4.6 Lập dự trữ và kế hoạch triển khai dự án

a) *Giai đoạn hình thành hợp đồng*

b) *Dự trữ thiết bị*

c) *Kế hoạch triển khai dự án:*

- Tổ chức
- Tiến độ

Bài tập chương 4

4.1 Tại sao phải khảo sát hiện trạng của hệ thống cũ khi xây dựng hệ thống thông tin mới ?.

4.1 Trình bày các phương pháp khảo sát hiện trạng hệ thống mà anh chị biết. Có nhất thiết khi khảo sát hệ thống người phân tích phải trực tiếp đến tận nơi để khảo sát không ? Tại sao?

4.2 Hãy thực tập khảo sát hệ thống thông tin phục vụ quản lý ở cơ quan và viết các yêu cầu mục tiêu của dự án tin học hoá, giả sử ta chọn một trong các dự án sau

- Hệ thống quản lý thư viện
- Hệ thống quản lý nhân sự/đảng viên
- Hệ thống nhân sự/tính lương
- Hệ thống quản lý vật tư.
- Hệ thống quản lý khen thưởng
- Hệ thống quản lý học tập của học sinh trường phổ thông
- Hệ thống quản lý học tập sinh viên đại học, cao đẳng
- Hệ thống tuyển sinh đại học

- Hệ thống quản lý kinh doanh trong lĩnh vực nào đó
- Hệ thống quản lý Mini Lab
- Hệ thống quản lý khách sạn
- Hệ thống quản lý sản xuất của nhà máy
- Hệ thống quản lý tín dụng/tiết kiệm
- Hệ thống quản lý thuế
- Hệ thống hạch toán kế toán
- Hệ thống quản lý bệnh án của bệnh viện.
- Hệ thống mạng máy tính
- Hệ thống quản lý đối tượng, vụ án
- Hệ thống quản lý các dự án v. v ...

Chương 5. Phân tích hệ thống về xử lý

5.1 Đại cương

Phân tích hệ thống theo nghĩa chung nhất là khảo sát nhận diện và phân định các thành phần của một phức hợp và chỉ ra các mối liên quan giữa chúng. Theo nghĩa hẹp phân tích hệ thống là giai đoạn 2, đi sau giai đoạn khảo sát sơ bộ, là giai đoạn bản lề giữa khảo sát sơ bộ và giai đoạn đi sâu vào các thành phần hệ thống.

Kết quả của giai đoạn này ta xây dựng được các biểu đồ mô tả logic chức năng xử lý của hệ thống. Giai đoạn này gọi là giai đoạn thiết kế logic chuẩn bị cho giai đoạn thiết kế vật lý. Yêu cầu đòi hỏi thiết kế logic một cách hoàn chỉnh trước khi thiết kế vật lý

Đường lối thực hiện:

- **Phân tích trên xuống (Top-down):** Phân tích từ đại thể đến chi tiết, thể hiện phân rã các chức năng ở biểu đồ phân cấp chức năng và ở cách phân mức ở BLD
- **Đi từ hệ thống cũ sang hệ thống mới:** (Từ II - III)
- **Chuyển từ mô tả vật lý sang mô tả logic:** (Từ I - II).

Sau đây ta chi tiết hoá từng phương pháp với các giai đoạn tương ứng

5.2. Phân tích hệ thống từ trên xuống:

Phương pháp phân tích từ trên xuống dưới áp dụng cho việc xây dựng hai loại biểu đồ liên quan đến chức năng xử lý : Biểu đồ phân cấp chức năng và Biểu đồ luồng dữ liệu

a) *Xây dựng biểu đồ phân cấp chức năng BPC:*

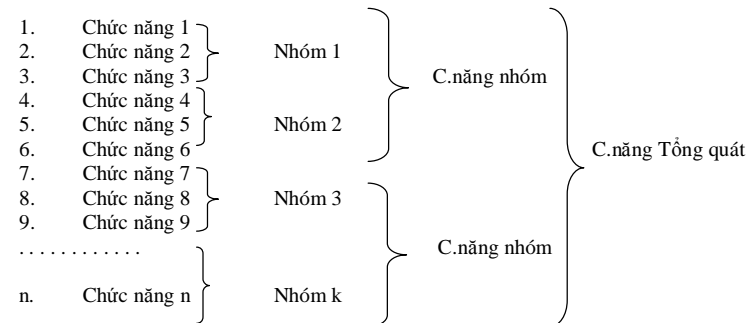
Đây là biểu đồ mô tả tĩnh. Bảng kỹ thuật phân mức ta xây dựng biểu đồ dưới dạng cây. Trong đó mỗi nút tương ứng với một chức năng.

Tại giai đoạn khảo sát sơ bộ hệ thống ta liệt kê các chức năng của hệ thống; Các chức năng này phản ánh hệ thống làm gì chẳng hạn như cập nhật dữ liệu, tra cứu, thống kê, tính toán xử lý. ... Các chức năng được phân thành từng nhóm chức năng có liên quan với nhau và chúng được xếp gần nhau. Các chức năng được đánh số theo thứ tự và theo nhóm.

Một điểm lưu ý rằng các phân tích viên thường gặp các sai lầm khi vẽ biểu đồ này:

- Các đường nối từ mức trên xuống mức dưới không có mũi tên vì bản thân các mức đã thể hiện tính phân cấp
- Biểu đồ này thuần tuý là chức năng xử lý, các tiến trình nên không có mô tả dữ liệu, hoặc mô tả các thuộc tính
- Lưu ý rằng đây là các chức năng của hệ thống thông tin chứ không phải là chức năng của hệ tác nghiệp.

Việc phân tích liệt kê các chức có dạng như sau

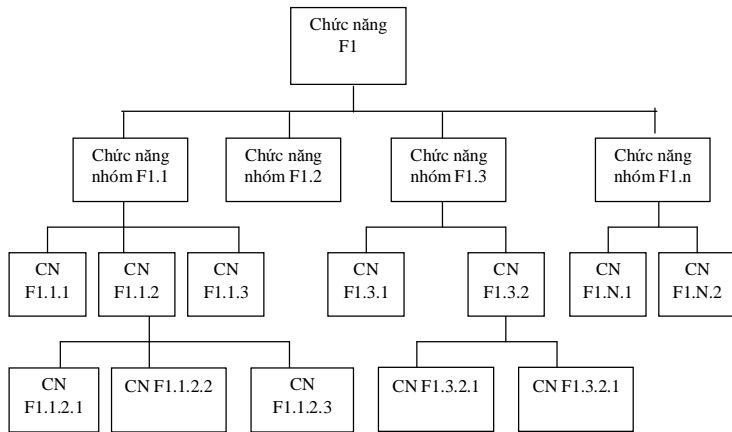


Mức 1: Nút gốc là chức năng tổng quát của hệ thống

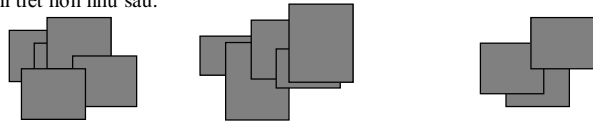
Mức 2: Phân rã ở chức năng thấp hơn là chức năng nhóm.

Các mức tiếp theo được phân rã (Decomposition) tiếp tục và mức cuối cùng là chức năng nhỏ nhất không phân chia được nữa.

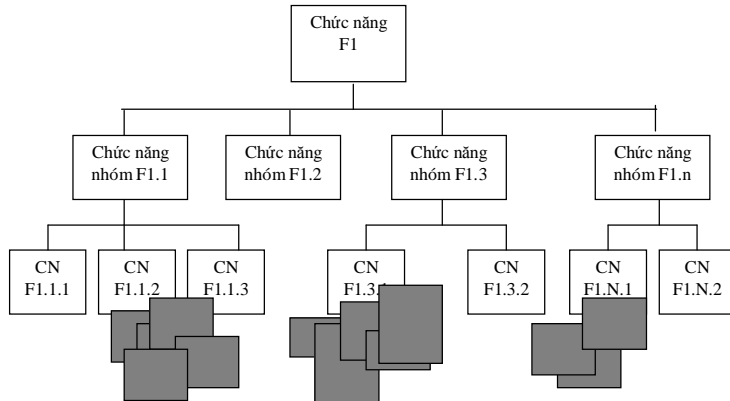
Cuối cùng ta có biểu đồ BPC sau:



Phương pháp kết hợp từ dưới lên trên (Bottom-up): Một cách để xác định các công việc cụ thể của một chức năng nào đó ta sử dụng kết hợp phương pháp “Kiểm soát” từ dưới lên trên. Thực chất của phương pháp này ngược với phương pháp trên để tạo thành một sơ đồ hoàn chỉnh. Mỗi chức năng nhỏ gom tụ từ một nhóm các công việc cụ thể chi tiết hơn như sau:



Từ 2 biểu đồ trên ta có BPC kết hợp như sau



Biểu đồ luồng dữ liệu:

Mô tả các chức năng của hệ thống theo tiến trình (process) (biểu đồ động)
 Phương pháp cấu trúc biểu đồ luồng dữ liệu: Biểu đồ luồng dữ liệu đối với hệ thống nhỏ, đơn giản thông thường được xây dựng dễ dàng, không công kênh để xem xét; tuy nhiên đối với hệ thống lớn phức tạp chẳng hạn như các hệ kinh doanh thì cách tốt nhất là nên tuân theo các hướng dẫn đơn giản để có được một biểu đồ tốt:

Xác định các thành phần tĩnh trong hệ thống, có nghĩa là các đối tượng có chứa dữ liệu
 Xác định các thao tác xử lý chính mà nó sử dụng và dữ liệu sinh ra, đồng thời xác định các dòng dữ liệu giữa chúng
 Mở rộng - Khai triển và làm mịn dần các tiến trình của biểu đồ

Chính lý lại biểu đồ từng bước thích hợp và bảo đảm tính logic
 Một kỹ thuật sử dụng khá phổ biến để phân rã (decompose) biểu đồ là kỹ thuật phân mức.
 Có 3 mức cơ bản được đề cập đến :

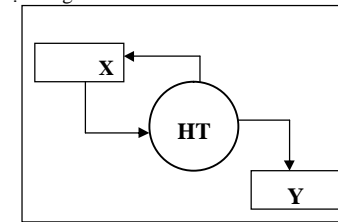
Mức 1: Biểu đồ luồng dữ liệu mức khung cảnh (Context Data Flow Diagram)

Mức 2: Biểu đồ luồng dữ liệu mức đỉnh (Top Level Data Flow Diagram)

Mức 3: Biểu đồ luồng dữ liệu mức dưới đỉnh (Levelling Data Flow Diagram)

BLD mức ngữ cảnh (mức 1): Đây là mô hình hệ thống ở mức tổng quát nhất, ta xem cả hệ thống như một chức năng. Tại mức này hệ thống chỉ có duy nhất một chức năng. Các tác nhân ngoài và đồng thời các luồng dữ liệu vào ra từ tác nhân ngoài đến hệ thống được xác định..

Thí dụ : Hệ thống được xác định 1 chức năng HT, Tác nhân ngoài là X và Y, 2 luồng dữ liệu từ hệ thống ra X và Y. Một luồng dữ liệu từ X vào hệ thống.



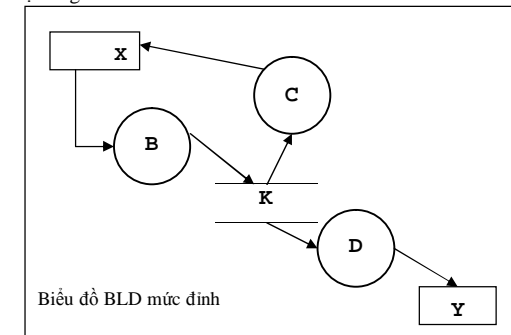
Biểu đồ luồng dữ liệu mức khung cảnh

Mức 2 : BLD mức đỉnh (BLD nhiều chức năng) được phân rã từ BLD mức ngữ cảnh với các chức năng phân rã tương ứng mức 2 của BPC. Các nguyên tắc phân rã :

- Các luồng dữ liệu được bảo toàn
- Các tác nhân ngoài bảo toàn
- Có thể xuất hiện các kho dữ liệu
- Bổ sung thêm các luồng dữ liệu nội tại nếu cần thiết.

Với thí dụ trên giả sử hệ thống được định nghĩa như sau:

Chức năng HT = B ⊕ C ⊕ D
 Dữ liệu : Xuất hiện kho K



Biểu đồ BLD mức đỉnh

Mức 3: BLD mức dưới đỉnh phân rã từ BLD mức đỉnh. Các chức năng được định nghĩa riêng từng biểu đồ hoặc ghép lại thành một biểu đồ trong trường hợp biểu đồ đơn giản. Các thành phần của biểu đồ được phát triển như sau

- Về Chức năng : phân rã chức năng cấp trên thành chức năng cấp dưới thấp hơn
- Luồng dữ liệu:

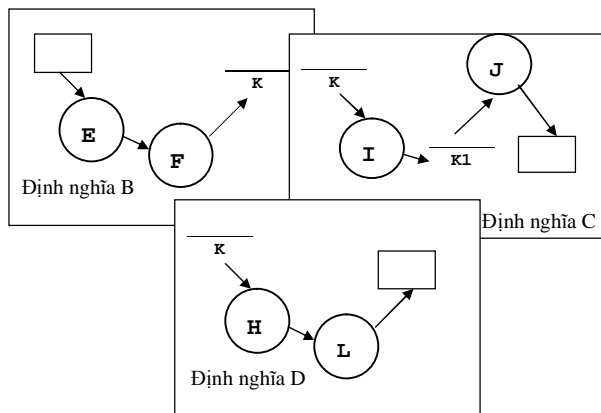
Vào/ra mức trên thì lặp lại (bảo toàn) ở mức dưới (phân rã)

Thêm luồng nội bộ

- Kho dữ liệu : dần dần xuất hiện theo nhu cầu nội bộ
- Tác nhân ngoài: Xuất hiện đầy đủ ở mức khung cảnh, ở mức dưới không thể thêm gì .

Thí dụ : Từ biểu đồ mức đỉnh trên ta có BLD mức dưới đỉnh với định nghĩa sau

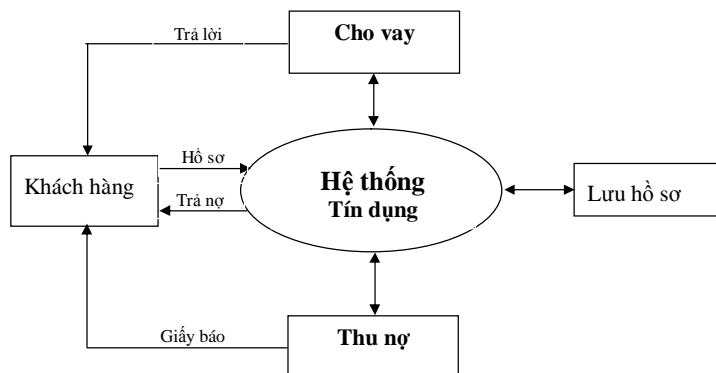
B = E ⊕ F; C = I ⊕ J; D = H ⊕ L; Thêm kho dữ liệu K1



Nhận xét: số mức phân rã thông thường là 7 ± 2 mức tùy độ phức tạp của hệ thống

Thí dụ : Hoạt động tín dụng

Sơ đồ luân chuyển thông tin



Đối với khách hàng đến vay tiền ở ngân hàng thì phải có một hồ sơ (gồm: Đơn xin vay, giấy chứng minh thư) và yêu cầu được vay. Nếu hồ sơ hợp lệ hoặc không hợp lệ thì hệ thống sẽ trả lời khách hàng.

Đối với ngân hàng nếu yêu cầu và hồ sơ của khách vay hợp lệ tức là yêu cầu của khách được đáp ứng thì ngân hàng lập một tài khoản tương ứng với khế ước vay mà ngân hàng quy định về số tài khoản, thời gian vay, mức lãi suất và ngày hoàn trả. Khách vay phải thanh toán (gốc + lãi) cho ngân hàng theo đúng hạn ghi trên khế ước vay, nếu quá hạn khách hàng không đến trả ngân hàng thì hệ thống sẽ thông báo tới khách hàng đồng thời áp dụng mức lãi suất quá hạn.

Đến kỳ hạn hoàn trả khách vay đến thanh toán (trả nợ) bộ phận thu nợ tính ra số tiền mà khách hàng phải trả, căn cứ vào ngày vay, ngày hoàn trả và lãi suất. Sau đó hệ thống đối chiếu với tài khoản gốc, in hoá đơn thanh toán và thông báo tới khách hàng.

3. Xây dựng biểu đồ phân cấp chức năng mới của hệ thống.

Sơ đồ phân cấp (rã) chức năng thể hiện cái nhìn tổng quát về hệ thống, đây là bước phân tích hệ thống về xử lý.

Gồm có các mức sau:

Mức 1: Quản lý toàn bộ hệ thống thông tin tín dụng

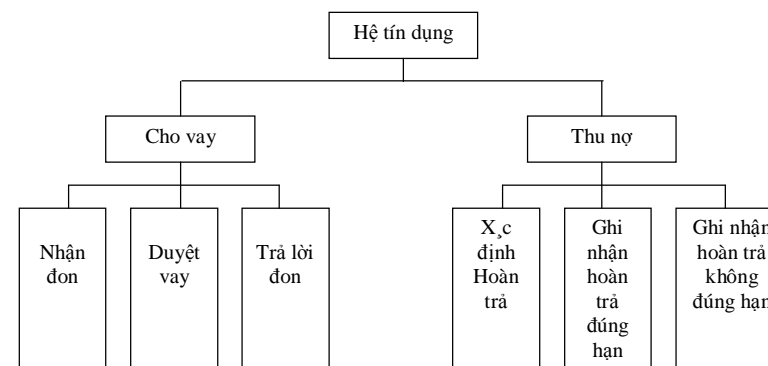
Mức 2: Gồm các chức năng sau:

- Cập nhật dữ liệu
 - + Khách hàng
 - + Lập khế ước
- Thu nợ
 - + Đúng hạn

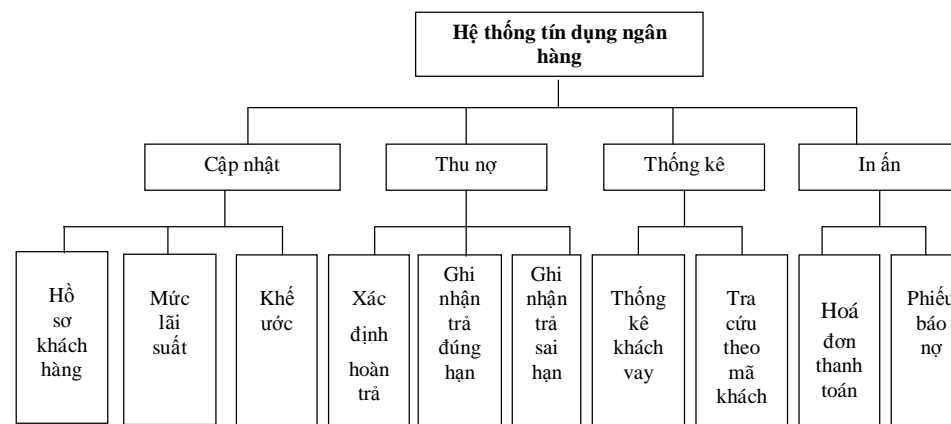
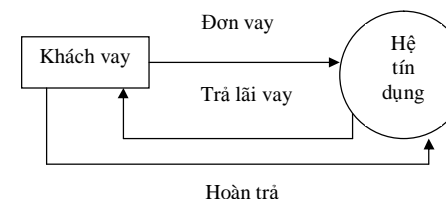
- + Quá hạn
- + Xác nhận hoàn trả
- Thống kê và tra cứu
 - + Thống kê khách vay
 - + Thống kê nợ quá hạn
 - + Tra cứu theo mã khách
- In ấn

- + In hoá đơn thanh toán
- + In phiếu báo nợ

Hoạt động tín dụng chủ yếu cho vay và thu nợ.

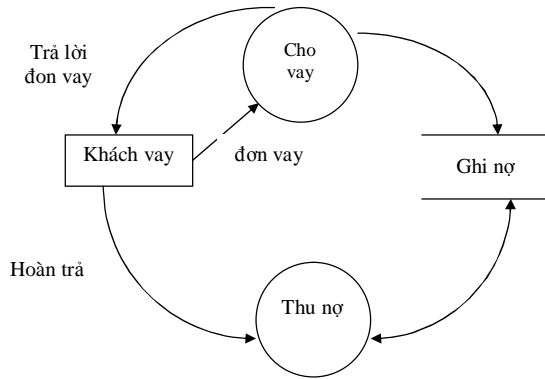


• BLD khung cảnh:



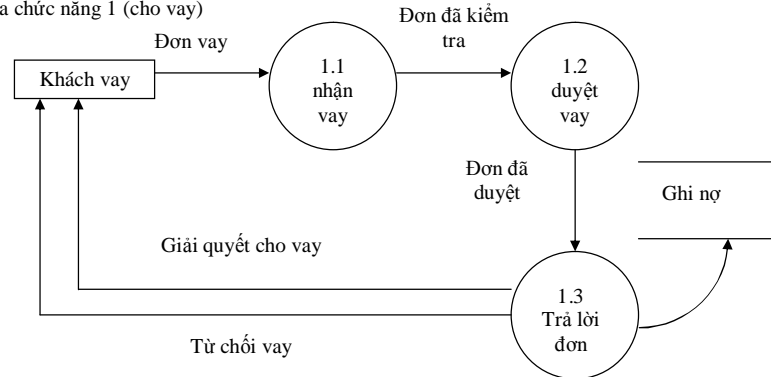
Biểu đồ phân rã chức năng.(BPC)

* BLD mức đỉnh:

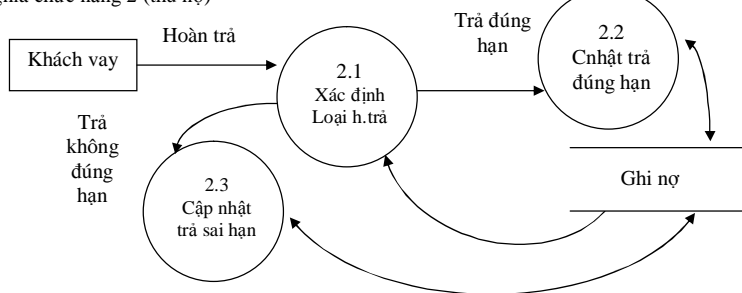


Biểu đồ luồng dữ liệu mức dưới đỉnh

• Định nghĩa chức năng 1 (cho vay)



• Định nghĩa chức năng 2 (thu nợ)



5.3. Chuyển từ BLD mức vật lý sang BLD mức logic

a) *Khái niệm BLD mức vật lý, mức logic:* Do phương pháp phân tích hệ thống có cấu trúc nên trong quá trình phát triển hệ thống một ưu thế quan trọng nhất trong thực hành là tách bạch chính thức cách nhìn “Vật lý” và cách nhìn “Logic” của hệ thống. ở bước trước chúng ta đã đề cập xây dựng BLD mức vật lý và nó làm tiền đề cho nhà phân tích thiết kế chuyển đổi thành biểu đồ BLD mức logic.

BLD mức vật lý của hệ thống mô tả cách thức hệ thống thực hiện các nhiệm vụ của nó, ai làm gì, làm ở đâu, mất bao nhiêu thời gian v.v...

Trong khi đó BLD mức logic bỏ qua những ràng buộc, các yếu tố vật lý, nó chỉ quan tâm chức năng nào là cần cho hệ thống và thông tin nào là cần để thực hiện cho chức năng đó. Nói một cách đơn giản BLD mức vật lý thường được dùng trong khảo sát hệ thống hiện tại (hệ thống cũ) và trong thiết kế hệ thống mới (khối I,IV trong 4 bước phân tích thiết kế có cấu trúc), còn các BLD logic được dùng cho việc phân tích các các yêu cầu của hệ thống cũ lẫn mới (Khối II,III trong 4 bước phân tích thiết kế có cấu trúc).

Sự phân biệt hai khái niệm vật lý và logic là nhân tố chủ yếu trong mọi phương pháp luận của hệ thống có cấu trúc

b) *Phương pháp chuyển đổi BLD mức vật lý sang mức logic:*

Xuất phát từ biểu đồ luồng dữ liệu mức vật lý ta tiến hành loại bỏ các yếu tố vật lý từ biểu đồ này. Đây là quá trình trừu tượng hoá các thành phần của biểu đồ, lược bỏ các yếu tố vật lý để giữ lại các tính chất tinh túy nhất mà vẫn không làm thay đổi bản chất của hệ thống.

Khi loại bỏ một số chức năng, dữ liệu và chỉ giữ những thành phần gắn liền với mục đích trong BLD, ta cần lưu ý loại bỏ theo các tiêu chí sau:

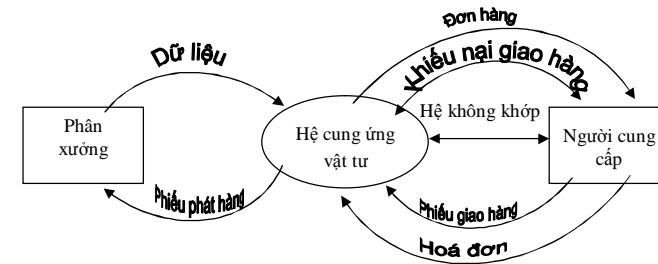
- Loại bỏ các chức năng do chính con người, thiết bị, và hệ thống thực hiện. Các chức năng này thuần túy chỉ là các thao tác vật lý, nên không tin học hoá được
- Phát hiện và loại bỏ những chức năng gắn liền với các biện pháp xử lý. Ở đây các chức năng này chỉ tồn tại tạm thời do những biện pháp quy định. Khi thay đổi biện pháp, các chức năng này không còn phù hợp nữa
- Loại bỏ các cấu trúc BLD gắn liền với biện pháp xử lý

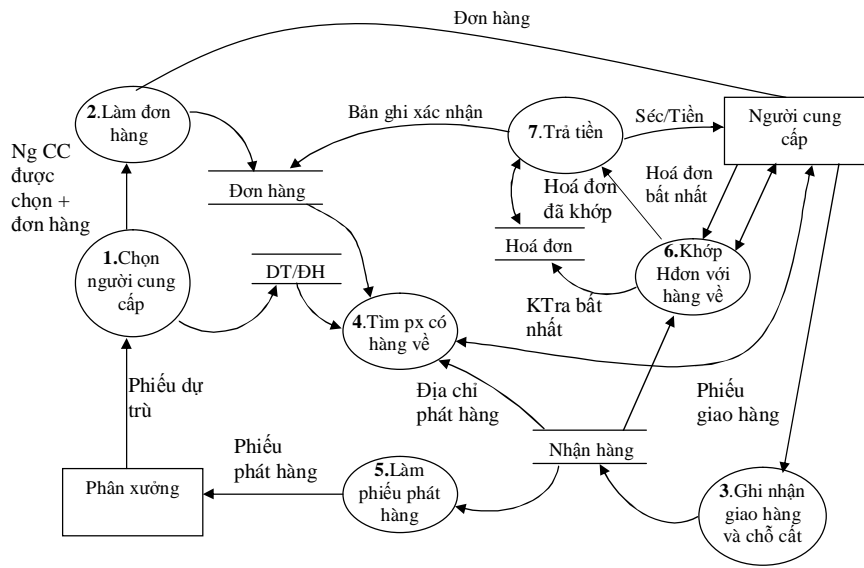
Biện pháp loại bỏ: Chúng ta có thể loại bỏ trên BLD bằng cách xoá bỏ các chức năng cần loại bỏ (xoá bỏ ngôn từ); thay thế chuyển đổi các luồng dữ liệu cho thích hợp khi loại bỏ một số chức năng và dữ liệu; ghép phối một số chức năng gắn liền thành cụm và cuối cùng là tổ chức lại biểu đồ bằng cách đánh số lại các chức năng. Trong trường hợp phát hiện một chức năng nào đó chưa rõ vật lý hay logic, cách tốt nhất là phân rã chức năng này thành các chức năng chi tiết hơn để việc loại bỏ được thực hiện.

Chú ý rằng việc chuyển đổi BLD từ mức vật lý thành mức logic chỉ diễn ra đối với BLD mức đỉnh và mức dưới đỉnh, không áp dụng cho BLD mức ngữ cảnh vì biểu đồ này chỉ có một chức năng duy nhất và không có kho dữ liệu.

Thí dụ B: Xây dựng BLD đối với hệ thống cung ứng vật tư của nhà máy ở mức logic. Đầu tiên ta xây dựng BLD ở mức vật lý

Biểu đồ luồng dữ liệu mức khung cảnh





Biểu đồ luồng dữ liệu mức logic

Trong biểu đồ luồng dữ liệu mức dưới đỉnh các chức năng đối chiếu thủ công bị loại bỏ. Triển khai chức năng 3 với nhiều đường vào ra. Cách tổ chức lại biểu đồ như sau:

- Chức năng 1.3 và 2.2 tuân tụy lệ lí được loại bỏ.
- Tiến hành ghép một số chức năng và đánh số lại ta có 7 chức năng sau:

1.1 thành	1
1.2 thành	2
2.1 thành	3
3.1, 3.2, và 3.3	4
2.3 thành	5
3.4 thành	6
4.1 và 1.4 thành	7

5. 4. Chuyển từ BLD của hệ thống cũ sang BLD của hệ thống mới.

Một câu hỏi đặt ra: "Vi sao ta cần thiết chuyển BLD hệ thống cũ sang hệ thống mới ở mức logic?". Trả lời là: Lý do để hệ thống mới thừa hưởng những cốt lõi tinh túy của hệ thống cũ, không làm biến đổi cái bản chất của hệ thống cũ, khắc phục các nhược điểm và kế thừa những cái đã có ưu điểm, khác về cài đặt.

Trong khi chuyển ta cần phải xem lại:

1. Những nhược điểm của hệ thống cũ như Thiếu chức năng, hiệu suất thấp, lãng phí. Những nhược điểm này cần được khắc phục
2. Các yêu cầu, mục tiêu của hệ thống mới: Đây là các yêu cầu ưu tiên cần bổ sung vào các chức năng của biểu đồ.

Việc biến đổi có thể thực hiện bằng cách khoanh lại một số vùng là các vùng thay đổi. Đối với những vùng thay đổi sẽ được sắp xếp lại sao cho:

- Luồng dữ liệu vào, ra: Đó là giao diện đối với những vùng còn lại phải bảo toàn.
- Xác định chức năng tổng quát của vùng thay đổi để khi biến đổi vẫn giữ nguyên được chức năng chính của nó; không làm cho chức năng này bị biến dạng.
- Xóa một phần BLD cần thay đổi bên trong và lập lại các chức năng từ nhỏ chi tiết, các chức năng biến đổi trung gian (kiểm tra, thêm...) và các trung tâm biến đổi
- Bổ sung các nhu cầu về kho dữ liệu, lập các luồng dữ liệu.

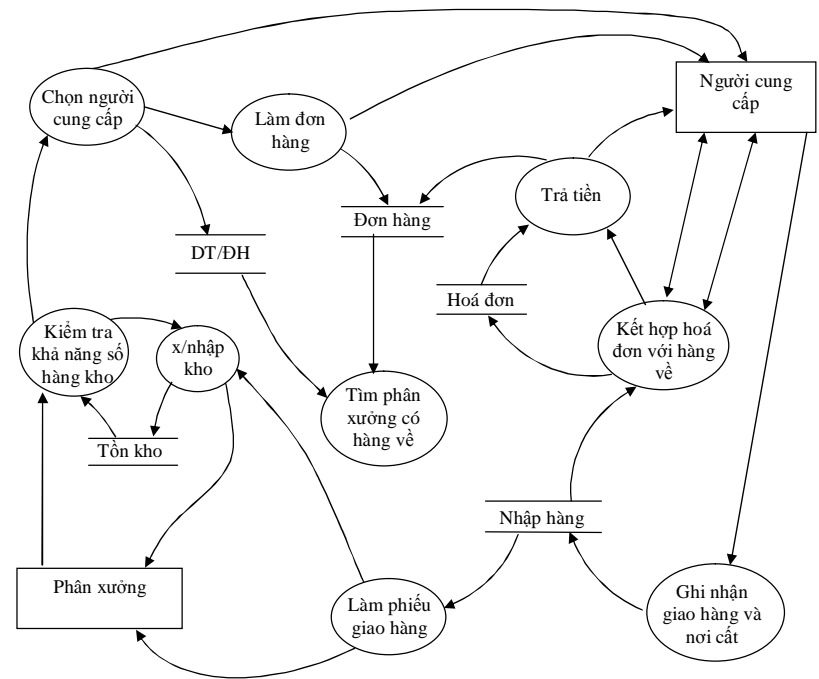
☞ Trở lại thí dụ (Case study): Quản lý sử dụng vật tư

1. Nhược điểm hệ thống cũ :

- Thiếu kho hàng thông dụng; Thiếu hẳn một chức năng trong BLD
- Tốc độ xử lý chậm: Do đối chiếu thủ công rất nhiều; lỗi này do cài đặt hệ thống ban đầu trên hai máy không tương thích, nên không thấy thể hiện ở BLD.
- Theo dõi thực hiện đơn hàng có nhiều sai sót : Từ các khâu làm đơn hàng đến việc nhận hàng và trả tiền có thể gây ảnh hưởng một phần.
- Sự lãng phí: Lý do chính là đối chiếu thủ công và cũng không thấy được ở BLD

Sau đây ta có thể vẽ lại Biểu đồ luồng dữ liệu của hệ thống mới ở mức logic. Biểu đồ này là biểu đồ cuối cùng của giai đoạn phân tích hệ thống về chức năng xử lý. Nó nhất thiết cần được rà soát nhiều lần để biểu đồ được hoàn thiện trước khi đưa ra thiết kế các module chương trình và dữ liệu.

Biểu đồ luồng dữ liệu của hệ thống mới mức logic



Bài tập chương 5

1. Ý tưởng cơ bản của phân tích hệ thống về xử lý là gì, gồm các bước và tiêu chuẩn nào?
2. Khi xây dựng biểu đồ phân cấp chức năng ta dựa vào các yếu tố nào?
3. Cơ sở để xây dựng biểu đồ luồng dữ liệu các mức: Khung cảnh, mức đỉnh và mức dưới đỉnh. Giữa biểu đồ BLD và biểu đồ BPC có mối liên hệ gì?. Tiêu chuẩn nào đánh giá biểu đồ xây dựng được là hợp lý và có tính logic
4. Biểu đồ luồng dữ liệu ở mức vật lý và mức logic khác nhau ở những điểm nào?
5. Tại sao các chức năng của biểu đồ BLD được gán nhãn (đánh số) theo dạng phân cấp?
6. Hãy thực hiện việc phân tích về chức năng xử lý của các hệ thống đưa ra trong phần bài tập chương 4 (câu 4.3)

Chương 6. Phân tích hệ thống về dữ liệu

6.1. Đại cương: Phân tích hệ thống bao gồm việc phân tích về xử lý và phân tích về dữ liệu. Chúng ta tách tạm thời việc phân tích dữ liệu vì dữ liệu có tính độc lập tương đối. Dữ liệu là đối tượng của xử lý. Mục đích của giai đoạn phân tích hệ thống về dữ liệu là: Lập lược đồ dữ liệu hay gọi là biểu đồ cấu trúc dữ liệu BCD. Hệ thống dữ liệu lưu giữ lâu dài:

- Thông tin gì, bao gồm dữ liệu gì.
- Mối liên quan: xác định liên quan giữa các dữ liệu.

Phương pháp thực hiện: thể hiện 2 cách tiếp cận:

- Mô hình thực thể liên kết: phương pháp này trực quan hơn đi từ trên xuống dưới, bằng cách xác định các thực thể, mối liên kết giữa chúng rồi đến các thuộc tính. Phương pháp này bao trùm được nhiều thông tin, tuy nhiên kết quả hay thừa.
- Mô hình quan hệ: xuất phát từ danh sách các thuộc tính rồi đi đến các lược đồ quan hệ. Phương pháp này đi từ dưới lên, kết quả là vừa đủ cho những kết xuất xử lý.

Chúng ta nên làm theo 2 cách để so sánh.

6.2. Thành lập BCD theo mô hình thực thể liên kết

Trước tiên ta phải thu nhập thông tin theo 3 yếu tố:

- Kiểu thực thể (Entities Type)
- Kiểu liên kết (Entities Relationship Type)
- Các thuộc tính (Attributes)

a. Phát hiện các kiểu thực thể: để phát hiện các kiểu thực thể ta thường tìm từ 3 nguồn:

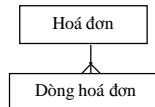
- Các tài nguyên: vật tư, tài chính, con người, môi trường
- Các giao dịch: đó là các thông tin đến từ môi trường bên ngoài nhằm kích động một chuỗi các hoạt động nào đó của hệ thống chẳng hạn như đơn hàng, hoá đơn...
- Các thông tin đã cấu trúc hoá: sổ sách, hồ sơ, các bảng biểu quy định..

b. Phát hiện các kiểu liên kết: trên thực tế có rất nhiều các liên kết giữa các kiểu thực thể nhưng ta chỉ ghi nhận các kiểu liên kết có ích cho công tác quản lý và các liên kết giữa các kiểu thực thể mà ta vừa phát hiện ở trên.

- **Liên kết 1 - nhiều:** đó là các liên kết thường hay gặp nhất, thường thông qua các đường truy nhập, không phải một bước mà được lần theo khoá có thể qua nhiều tệp khác nhau. Các liên kết 1 - nhiều thường là:

Chứng từ / Dòng chứng từ, hóa đơn / dòng hóa đơn

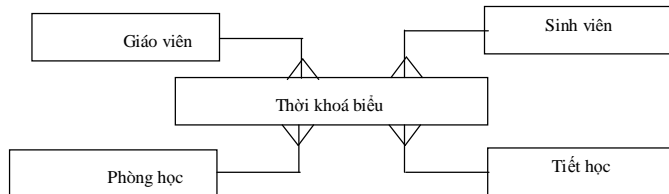
Ví dụ:



Mối liên quan thường được diễn tả bằng các giới từ sở hữu “cho, thuộc, bởi, của...”.

- **Liên kết nhiều - nhiều:** có thể liên kết nhiều bên nhiều phía. Trên biểu đồ nó phải được thể hiện bằng một kiểu thực thể trung gian với khoá là tổ hợp khoá của các bên tham gia.

Ví dụ:



c. Phát hiện các thuộc tính:

Thuộc tính khoá nhận diện (khoá đơn hoặc khoá kép)

Các thuộc tính mô tả chỉ xuất hiện ở mỗi kiểu thực thể, dùng để mô tả các đặc trưng của thực thể.

Thuộc tính kết nối: đó là thuộc tính thể hiện vai trò kết nối giữa 2 kiểu thực thể. Nó là thuộc tính khoá nhận diện ở thực thể này và đồng thời xuất hiện là thuộc tính mô tả ở thực thể khác.

Ví dụ: Việc xuất nhập vật tư của một cơ sở sản xuất được tiến hành như sau:

Khi xuất hoặc nhập vật tư kèm theo phiếu nhập/xuất kho được nhập vào máy.

Phiếu có dạng sau:

Đơn vị		Phiếu nhập/xuất kho		Quyển số			
Số.....							
Ngày.....tháng..... năm.....							
Tên người lập:.....Bộ phận:.....							
Nhập vào kho:.....Ghi có tài khoản.....							
Số TT	Tên hàng	Đơn vị tính	Số lượng		Giá đơn vị	Thành tiền	Ghi chú
			Xin nhập	Thực nhập			
Cộng							
Cộng thành tiền (Viết bằng chữ).....							
Người nhập/xuất		Thủ kho		Kế toán trưởng		Thủ trưởng đơn vị	

Ở đây sẽ xuất hiện 2 kiểu thực thể là phiếu nhập/xuất và dòng phiếu nhập/xuất.

Chúng ta xem xét lại ví dụ về hệ thống Quản lý vật tư:

Các thực thể

(1) Tài nguyên:

- Người cấp
- Phần xưởng
- Tồn kho
- Mặt hàng

(2) Giao dịch :

- Đơn hàng - dòng đơn hàng
- Giao hàng - dòng giao hàng
- Hoá đơn - dòng hoá đơn
- Phát hàng - dòng phát hàng
- Dự trữ - dòng dự trữ

(3) Thông tin cấu trúc: (đa số là các liên kết phản ánh bằng sổ sách)

- Xuất/nhập kho
- Các liên kết:
- Dự trữ/ Đơn hàng (nhiều - nhiều)
- Mặt hàng / người cấp (1- nhiều)

Qua đó sơ bộ ta vẽ được biểu đồ sau:

Thông tin tình huống (đưa vào trong tương lai)

- Danh sách những thông tin cơ bản
- Xuất phát từ 1 hay một số tài liệu xuất ra của các hệ thống, nhất các tiêu thức (cái ra → cái cần)" Phương pháp cái ra" Method Sortir.
- **Bước 2:** Tu chỉnh lại danh sách:
 - Loại bỏ các tên đồng nghĩa
 - Loại bỏ các thuộc tính tính toán.

Ví Dụ: thành tiền = đơn giá * số lượng (trong một hoá đơn)
 (Vì đây là giai đoạn logic chỉ tính đến đây đủ và hợp lí chưa nói đến tiện lợi, sau này đến giai đoạn thiết kế có thể ta lại bổ sung thêm thuộc tính này).

- Loại bỏ các thuộc tính tích lũy (thực chất cũng từ thuộc tính tính toán)
 Ví dụ: số hàng tồn kho = Σ nhập = Σ xuất
- Thay thế các thuộc tính không đơn bởi các thuộc tính đơn.
- **Bước 3:** Tìm các phụ thuộc hàm có trong danh sách nói trên.
 Việc xác định các phụ thuộc hàm thực chất phải dựa vào ý nghĩa trên thực tế.
- Rà từng cặp thuộc tính trong những danh sách trên, hoặc máy móc và đơn giản hơn là lập bảng 2 chiều.
- Tìm các phụ thuộc hàm về phải không đơn (gồm nhiều thuộc tính)

- **Bước 4:** Chuẩn hoá: có 2 phương pháp

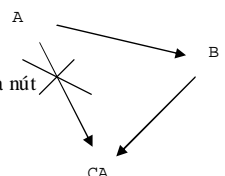
• **PP1 Phân rã:**
 Ban đầu ta coi tất cả các thuộc tính nằm trong một quan hệ rồi tiến hành phân rã quan hệ này...

- i) **Dạng 1NF:** Tách nhóm các thuộc tính lặp
 - o Phần còn lại có thể tạo thành một quan hệ, tìm khoá cho nó
 - o Phần tách ra + khoá trên lập thành quan hệ khác. Tìm khoá
- ii) **Dạng 2NF:** Loại bỏ phụ thuộc hàm bộ phận vào khoá bằng cách tách ra cộng với bộ phận của khoá nói trên (thông thường khoá là bộ phận nói trên) để tạo ra các quan hệ mới.
- iii) **Dạng 3NF:** Loại bỏ phụ thuộc hàm không khoá bằng cách tách những phụ thuộc hàm không có khoá tham gia, tách ra + thuộc tính ở về trái (khóa). Hay nói khác đi là tách các nhóm thuộc tính phụ thuộc hàm vào thuộc tính không phải là khoá, nhóm tách ra là một quan hệ có khóa mới.

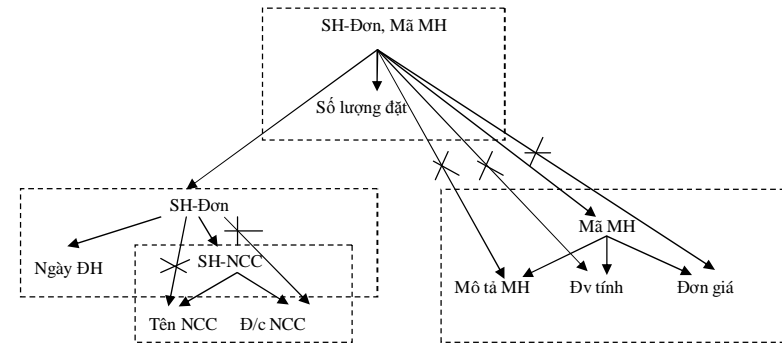
VD:

Danh sách các thuộc tính	Dạng 1NF	Dạng 2NF	Dạng 3NF
SH- Đơn	SH- Đơn	SH- Đơn	SH- Đơn
SH-NCC	SH-NCC	SH-NCC	SH-NCC
Tên-NCC	Tên-NCC	Tên-NCC	Ngày-ĐH
Đ/C-NCC	Đ/C-NCC	Đ/C-NCC	
Ngày-ĐH	Ngày-ĐH	Ngày-ĐH	SH-NCC
Mã-MH			Tên-NCC
Mô tả-MH	SH- Đơn	SH- Đơn	Đ/C-NCC
Đơn vị tính	Mã-MH	Mã-MH	
Đơn giá	Mô tả-MH	Số lượng đặt	SH- Đơn
Số lượng đặt	Đơn vị tính		Mã-MH
Thành tiền	Đơn giá	Mã-MH	Số lượng đặt
Tổng cộng	Số lượng đặt	Mô tả-MH	
		Đơn vị tính	Mã-MH
		Đơn giá	Mô tả-MH
			Đơn vị tính
			Đơn giá

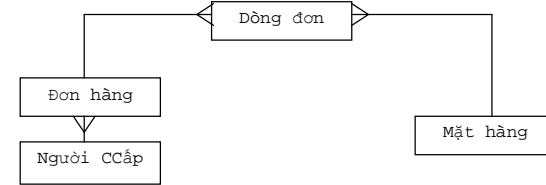
- **PP2: Phương pháp Tổng hợp**
- i) **Lập một đồ thị có hướng gọi là đồ thị phụ thuộc hàm:**
 - o Mỗi thuộc tính là một nút
 - o Mỗi một nhóm thuộc tính là về trái của 1 phụ thuộc hàm cũng là nút
 - o Có 1 phụ thuộc hàm A → B thì ta vẽ 1 cung A đến B.
- ii) **Loại bỏ các cung khép kín** (loại các phụ thuộc hàm không trực tiếp).
- iii) **Dùng các hình chữ nhật để khoanh vùng:**
 Mỗi nút trong (có con) lấy làm khoá gộp cùng với các con của nó lập thành 1 quan hệ.



- Ví dụ:
- 1 SH- Đơn → SH - Nguồn cung cấp, Ten Nccap, Đ/chi - Nccap, ngày - ĐH
 - 2 SH - Nccap → Ten - Nccap, Đ/c - Nccap
 - 3 Mã MH → Mô tả - MH, Đơn vị tính, Đơn giá
 - 4 SH - Mã MH → SH - Đơn, Mã MH, Mô tả MH, ĐV tính, Đơn giá, số lượng đặt



Vậy lập được 4 quan hệ
 MATHANG (ma - MH, Mota MH, ĐV tính, Đơn giá)
 NGCCAP (SH- NCC, Ten- NCC, ĐC - NCC)
 ĐONHG (SH-Đơn, Ngày ĐH, SH - NCC)
 Đòng ĐH (SH - Đơn, Mã MH, Số lượng đặt)
 Trong sơ đồ thực thể liên kết mô hình:



- **Bước 5:** Lập lại các bước từ 1 -4 trên các danh sách xuất phát khác...
 Nếu cùng 1 kiểu thực thể, từ các danh sách xuất phát khác nhau có các quan hệ danh sách thuộc tính khác nhau. Gộp lại: khi gộp lại có thể xuất hiện phụ thuộc hàm bậc cao, khi gộp xong phải tiếp tục cho chuẩn hoá quan hệ vừa gộp lại.

Ví dụ:
 Đơn hàng (SH - Đơn, SH - Khách hàng, Ngày ĐH)
 Từ phiếu giao hàng Đơn hàng (SH - Đơn, Tình trạng đơn, Đ/c giao hàng)
 Gộp lại, nghiên cứu các phụ thuộc hàm này sinh bên trong.
 Giả sử mỗi khách hàng chỉ có 1 địa chỉ giao hàng duy nhất, khi ấy có
 SH - khách hàng → Đ/c giao hàng
 Chuẩn hoá để được Đơn hàng (SH - Đơn, SH - khách hàng, ngày Đh, tình trạng đơn)
 Khách hàng (Sh - khách hàng, Đ/c giao hàng)

- * **Kết luận:**
 Phân tích hệ thống chức năng:
 + BPC chức năng (tính)
 + BLD luồng dữ liệu (động)
 Phân tích hệ thống dữ liệu:
 BCD:
 + Mô hình thực thể liên kết
 + Mô hình quan hệ :
 - Phân rã
 - Tổng hợp

Sau đó so sánh các phương pháp để xem có sai sót gì không?

Chỉ giữ lại những liên kết 1 - nhiều cần để làm các đường truy nhập vì các liên kết nhiều - nhiều được tách ra, các liên kết 1 - 1 ít sử dụng.

Bài tập chương 6

6.1 So sánh Phương pháp xây dựng lược đồ cấu trúc dữ liệu bằng mô hình thực thể liên kết và mô hình quan hệ.

6.2 Liên kết 1-1 xuất hiện trong các trường hợp nào? Cho ví dụ

6.3 Liên kết nhiều- nhiều (N-N) được xử lý như thế nào trong mô hình thực thể liên kết E-R.

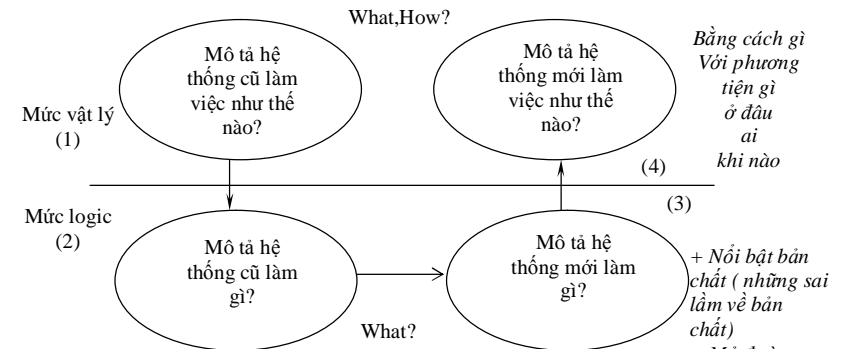
6.4 Liên kết 1-nhiều (1-N) thường gặp trong các trường hợp nào?. Cho ví dụ

6.5 Hãy thực hành xây dựng các mô hình thực thể liên kết E-R trong các hệ thống cho ở bài tập chương 4 (4.3).

6.6 Thuộc tính khoá, thuộc tính kết nối của mỗi thực thể được xác định như thế nào?

Chương 7 Thiết kế tổng thể và thiết kế giao diện

Mở đầu : Ta xem lại 4 bước của thiết kế có cấu trúc



1. Đại cương về giai đoạn thiết kế:

Xuất phát: mô tả logic của hệ thống mới

- BPC (phân rã chức năng - mô tả tính)

- BLD (mô tả động đặt trong mối liên quan về dữ liệu đối với nhau) đã phân mức

- BCD cấu trúc dữ liệu : Thông tin / liên kết

Quan hệ giữa các thực thể và các thuộc tính

Nhiệm vụ giai đoạn này là Chuyển biểu đồ ở mức logic → mức Vật lý

- Biện pháp

- Phương tiện

- Cách cài đặt.....

Các bước :

+ *Thiết kế tổng thể:* - Phân định ranh giới phần thực hiện bởi Máy Tính và thủ công

- Phân định các hệ thống con MT (khu vực trong biểu đồ luồng dữ liệu được xử lý bằng

MT)

+ *Thiết kế giao diện:* (thiết kế đầu ra và đầu vào) thường thiết kế đầu ra trước rồi thiết kế đầu vào

+ *Thiết kế các kiểm soát:* - Các vấn đề về bảo mật

- Vấn đề về bảo vệ

+ *Thiết kế các tệp* (khi thiết kế logic BCD chỉ quan tâm: đủ ? không trùng lặp?

- Tiện

- Nhanh khi truy nhập

(Từ mô hình lí tưởng → cài đặt thực tế)

+ *Thiết kế về chương trình*

Có nhiều phương thức sử dụng MT: - Tập trung hay phân tán ?

- Các hình thức xử lý theo mẻ hay theo kiểu trực tuyến?

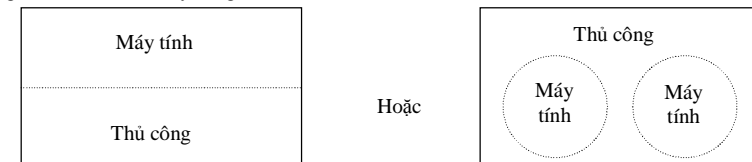
- Phân định MT/ thủ công

- Phân định hệ thống con MT

2. Phân định hệ thống MT và hệ thống thủ công

BLD ở một mức nào đó, kết quả vẫn BLD đó có thêm ranh giới giữa MT và thủ công, trên hình vẽ phân định bằng các đường nét đứt đoạn....., có thể các vùng không liên thông.

Ví dụ trong biểu đồ ta nhìn thấy đáng điệu sau



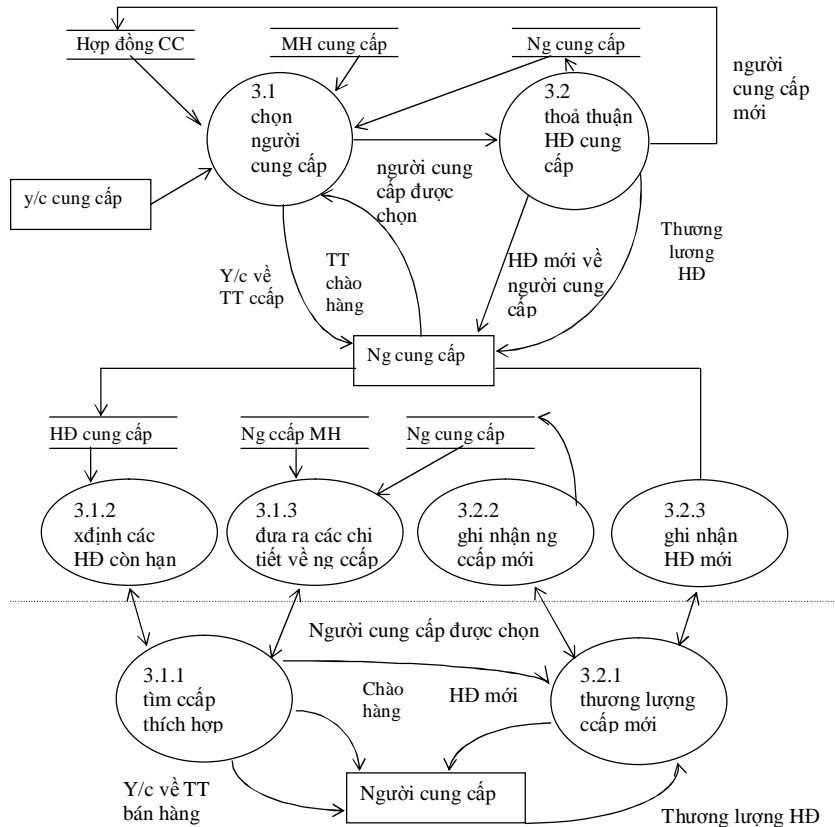
a) Đối với các chức năng xử lý

+ Dồn về hẳn một bên các chức năng thực hiện bằng máy tính, điều đó khá dễ
 + Nếu trong trường hợp các chức năng không hẳn về 1 bên ta tiếp tục phân rã nhỏ đi sao cho sau khi phân rã được tiếp sự phân biệt rõ ràng giữa MT và thủ công

Ví dụ: Trong hệ thống cung ứng vật tư, giao diện giữa người cung ứng vật tư và khách hàng có những việc thực hiện tự động hoá bằng máy tính, tuy nhiên có những việc thực hiện thuần túy bằng thủ công mà không thể thay thế bằng máy tính được. Chức năng đối chiếu cũng tương tự như vậy. Bởi vậy các chức năng đó được phân rã tách thành các chức năng chi tiết hơn để có thể tách rời các phần MT/TC

Một số chú ý:

1. Việc phân định các chức năng MT/TC đôi khi dẫn đến sự nhầm lẫn giữa hệ thống thông tin và hệ tác nghiệp. Thực chất ta đang xét MT/TC ngay trong hệ thống thông tin
2. Trong một chức năng đôi khi có những phần vừa máy tính vừa thủ công, cái khó là làm sao ta có thể tách chúng ra được mà vẫn giữ nguyên được hình dáng biểu đồ của hệ thống.
3. Việc tách phần MT/TC nhằm gợi ý cho người thiết kế chú ý đến thiết kế giao diện người dùng tại biên giới MT/TC
4. Đối với các hệ thống dùng phương thức trực tuyến thì phần làm bằng máy tính sẽ là chủ yếu. Phần thực hiện thủ công chỉ mang tính theo dõi kiểm tra



b) Đối với các kho dữ liệu

Nếu kho dữ liệu khi chuyển sang thực hiện bằng Máy tính, nó biến thành các kiểu thực thể, liên kết và sau đó khi cài đặt nó là các file dữ liệu, các cơ sở dữ liệu cần phải so sánh lại với biểu đồ cấu trúc đã có phải có mặt trong BCD

Kho dữ liệu nếu thực hiện bằng thủ công: chẳng hạn hồ sơ tài liệu, thì cần loại ra khỏi BCD.

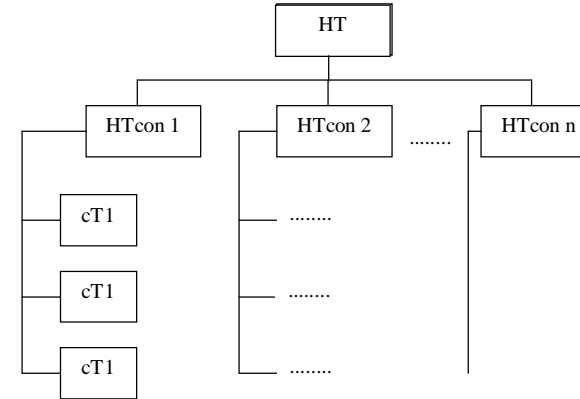
Đôi khi ta buộc phải thêm 1 thực thể hay 1 liên kết vào BCD để thuận tiện cho việc cài đặt (Chẳng hạn bảng giá, catalo cung cấp, Số hiệu đơn hàng v. v...)

c) *Chọn lựa phương thức và sử dụng máy tính:* Các phương thức thể hiện đối với hệ thống có thể là hệ thống mở, hệ thống trực tuyến.

Ví dụ: Kiểm tra kho hàng tuần, hàng ngày và bất cứ lí do nào

3. Phân định các hệ thống con MT

Hệ thống con thực chất là 1 bộ riêng lẻ chương trình



Nguyên tắc phân định không nhất thiết chỉ căn cứ vào chức năng thuần túy

Các căn cứ để phân định:

(1) Theo thực thể: gom tùm những chức năng xung quanh 1 kiểu thực thể hay (tập) 1 nhóm kiểu thực thể (tập hợp tập)

VD hệ thống con khách hàng
 hệ thống con kho vật tư

(2) Theo giao dịch: là thông tin về kinh doanh khi xuất hiện sẽ khởi động 1 loạt chức năng để cập nhật thông tin VD Đơn hàng là 1 trong những giao dịch

Gom các chức năng được khởi động bởi giao dịch đó vào 1 nhóm, dựa vào phương pháp lan truyền đối với các luồng dữ liệu xuất phát từ thông tin vào khởi động bằng sự giao dịch

(3) Theo thông tin biến đổi:

Nếu nhận thấy trong BLD có 1 khu vực tập trung xử lí thông tin chủ yếu (Central) thì gom những chức năng này lại "nhắc lên" kéo theo những gì liên quan.

VD Tính lương:

Khi cần tính Lương sẽ kéo theo những đầu vào: cấp bậc, thâm niên..., đồng thời đối với đầu ra sẽ bảng lương chi tiết, bảng tổng hợp lương...

(4) Gộp theo tính thiết thực: chẳng hạn

- + Cấu trúc kinh doanh của cơ quan
- + Vị trí cơ sở
- + Sự tồn tại của phần cứng
- + Trình độ đội ngũ của cán bộ nhân viên thừa hành
- + Trách nhiệm công tác (thường là quyền ưu tiên xâm nhập vào dữ liệu) (private)

4. Thiết kế chi tiết về các thủ tục người dùng và các giao diện

a) Chức năng thủ công: có nhiều loại

- Không liên quan đến MT
- Có máy tính trợ giúp:
 - + batch
 - + On - line
- Có những chức năng thủ công mới xuất hiện để phục vụ MT: "ăn theo" sự xuất hiện MT

- + Mã hoá thông tin thu nhập
- + Kiểm soát và sửa chữa thông tin
- + Nhập thông tin
- + Kiểm tra tài liệu xuất
- + Phân phối tài liệu xuất

Yêu cầu: chức năng thủ công

- Đáp ứng đòi hỏi hệ thống
- Thông tin chính xác
- Dễ dùng, dễ hiểu
- Gõ phím ít nhất ngắn gọn đủ ý
- Nội dung phải làm (không gian, thời gian)
- Yêu cầu năng suất (qualification)
- Cách xử lí các sai sót gặp phải

b) *Thiết kế các tài liệu xuất*

Xuất dữ liệu theo các phương tiện:

- Giấy
- Màn hình
- Đĩa
- v. v ...

Phương thức xuất: - Lập tức
- Trì hoãn

Tài liệu Xuất ra gồm

- Tài liệu có cấu trúc (thông thường là thừa)
- Thông tin tuỳ tiện (theo kiểu may - đo) trả lời nhu cầu (inquiry), loại tài liệu phổ cập, nhưng loại này không định dạng. Trong trường hợp này yêu cầu người dùng phải hiểu được ngôn ngữ thể hệ 4 trong việc hỏi đáp (Nếu không phải dùng 1 trung tâm "phiên dịch" làm trung gian cho người dùng và hệ thống)
- Yêu cầu đối với tài liệu xuất phải đủ, chính xác (kiểm tra không nhập nhầm), dễ hiểu, dễ đọc

Có 2 hình thức in ra: Khung in sẵn/ không có khung in sẵn

- Trình bày:
- Đầu (Heading)
 - Thân (bao gồm những nội dung cơ bản, gom nhóm có mối liên hệ logic với nhau)
 - Cuối

c) *Thiết kế các màn hình và đơn chọn:*

Mục đích sử dụng màn hình là đối thoại: đặc điểm của đối thoại là

- Vào / Ra gần nhau
- Thông tin thường tối thiểu (Cần đâu lấy đấy, không đưa sẵn)

Yêu cầu thiết kế - Sáng sủa (dễ nhìn, dễ đọc)
- Lệnh phải rành mạch (muốn gì? Làm gì?)

Hình thức đối thoại trên màn hình: Thiết kế màn hình liên quan đến hình thức, định dạng, thiết lập, trình bày các thông tin trên màn hình. Bước đầu tiên của thiết kế là phân tích đối thoại giữa người dùng và máy tính. Việc phân tích này đòi hỏi cần xác định nhóm logic của đối thoại liên quan đến các hành vi đơn giản chẳng hạn như các yêu cầu người dùng hoặc hiển thị chi tiết về dữ liệu. Các dạng hội thoại thường được đề cập

- + Câu lệnh, câu nhắc
- + Đơn chọn (Menu): Ngày nay người ta dùng đơn chọn phân cấp, nên chú ý lối thoát của mỗi cấp. Kết hợp với đơn chọn là các hộp chiếu sáng để tăng tính hấp dẫn
- + Điền mẫu
- + Sử dụng các biểu tượng (Icon) để tăng tính trực quan

d) *Thiết kế cái vào*

- 1) Chọn phương thức thu nhập thông tin: - On-line
- Trì hoãn (đưa qua thời gian, cập nhật sau)
- Từ xa

2) **Xác định khuôn mẫu thu nhập thông tin**

Mẫu có 2 kiểu - Khung (để điền)
- Câu hỏi (câu hỏi đóng: trả lời xác định trước, câu hỏi mở: gợi ý)

- Yêu cầu mẫu - Thuận tiện cho người điều tra
- Thuận tiện mã hoá
- Thuận tiện người gõ phím

- Nội dung đơn giản, rõ ràng, chính xác

Tóm lại:

Thiết kế giao diện là một trong những phần thiết yếu của hệ thống để hệ thống trình bày một phần các thông tin mà người sử dụng cần biết. Bởi vậy mục tiêu của nó cần được người thiết kế tiến hành một cách hết sức cẩn thận. Các yêu cầu chính cần được xem xét:

- Loại thiết bị phương tiện giao diện được sử dụng
- Thiết kế hội thoại người dùng - hệ thống
- Bản chất của dữ liệu và phương cách mã hoá dữ liệu
- Các yêu cầu về kỹ thuật đánh giá dữ liệu
- Thiết lập định dạng màn hình và các báo cáo.

Bài tập chương 7

7.1 Hãy thiết kế giao diện cho chương trình cập nhật dữ liệu khi có độc giả yêu cầu mượn sách trong hệ thống thư viện.

7.2 Đối hệ thống tuyển sinh vào các trường đại học hãy phân định hệ thống máy tính và thủ công cho hợp lý và logic

7.3 Phân định hệ thống quản lý sản xuất của xí nghiệp thành các hệ thống con: Nhân sự, vật tư, lương, kế toán, kế hoạch, tiếp thị

7.4 Phân định hệ thống kinh doanh tiền tệ tại ngân hàng với các chức năng: Tín dụng, tiết kiệm, kế toán ngân hàng.

7.5 Thiết kế dữ liệu đầu vào của hệ thống:

Quản lý nhân sự của trường đại học

Hoá đơn thanh toán và các phiếu xuất nhập của hệ thống kinh doanh.

Hồ sơ bệnh án trong các bệnh viện.

7.6 Thiết kế tổng thể thực hiện các nhiệm vụ gì?

7.7 Thế nào là hệ thống con? Có phải mọi hệ thống đều phải phân định thành các hệ thống con? Cho ví dụ minh hoạ

Chương 8 Thiết kế các kiểm soát

1. Giới thiệu

Ở một số giai đoạn trong quá trình phát triển của hệ thống bao giờ cũng cần tiến hành các kiểm tra cần thiết để đảm bảo việc thực hiện đúng đắn cho hệ thống dự định.

Việc kiểm soát hệ thống nhằm tránh một số nguy cơ:

- Sai lỗi do đó phải tiến hành kiểm tra các thông tin thu nhập.
- Sự cố kỹ thuật do vậy phải tiến hành bảo vệ (an toàn).
- Ý đồ xấu do đó phải tiến hành bảo mật
- Rủi ro về môi trường: cháy, bão lụt.

Ba khía cạnh cơ bản của hệ thống cần được bảo vệ bằng cách kiểm soát đó là:

- **Độ chính xác:** phải kiểm tra xem các giao tác đang được tiến hành có được thực hiện chính xác hay không và các thông tin được giữ trong cơ sở dữ liệu của công ty có đúng đắn không.
- **Độ an toàn:** có một yêu cầu bao trùm về việc gìn giữ tài sản của công ty, để đảm bảo rằng không xảy ra mất mát dù có ý hay vô tình, dù do chệnh mảng hay rủi ro.
- **Độ riêng tư:** cũng có nhu cầu kiểm tra xem các quyền của cá nhân và công ty khác có được bảo vệ không. Có lẽ khía cạnh quan trọng nhất của vấn đề này là đảm bảo rằng hệ thống dự kiến sẽ tuân thủ những hạn chế do Luật bảo vệ dữ liệu áp đặt.

2. Nghiên cứu việc kiểm tra các thông tin thu nhập hay xuất ra

- Mục đích: bảo đảm tính xác thực của thông tin
- Yêu cầu: Mọi thông tin xuất ra hay nhập vào đều phải qua kiểm tra
- Nơi kiểm tra:
 - Nơi thu nhập
 - Trung tâm máy tính / nơi phân phát
- Nội dung kiểm tra: phát hiện lỗi và sửa lỗi
- Hình thức kiểm tra:
 - Tay (thủ công): đầy đủ / không đầy đủ
 - Máy (tự động): trực tiếp / gián tiếp, tham khảo các thông tin khác.
- Thứ tự kiểm tra: Kiểm tra trực tiếp trước, gián tiếp sau.
 - Trực tiếp: Sự cố mật
Khuôn dạng
Kiểu
Miền giá trị
 - Gián tiếp: Kiểm tra 1 thông tin khi mà các thông tin dùng cho việc kiểm tra đó đã được kiểm tra.
 - Kiểm tra tự động: kiểm tra sự ràng buộc toàn vẹn (integrity constraint)

2. Cách giai đoạn tiếp cận kỹ thuật phân tích các kiểm soát

a) **Xác định các điểm hở trong hệ thống:** Điểm hở là điểm mà tại đó thông tin của hệ thống có tiềm năng bị thâm nhập bởi những người trong hoặc ngoài tổ chức. Điều này không chỉ nói tới dạng đầu ra, như đơn mua hàng và bảng kiểm kê, mà còn nói tới mọi thông tin bên trong công ty mà nếu bị dùng sai thì có thể làm cho tài sản công ty chịu rủi ro. Mỗi khi xác định được điểm hở, cần phải tiến hành ba hoạt động sau:

- **Xác định kiểu đe dọa từ chỗ hở:** Các kiểu đe dọa này bao gồm từ các hành động cố ý như ăn cắp hoặc phá hoại cho tới các nguy cơ mất mát tài sản và ảnh hưởng tới công việc kinh doanh của công ty, chẳng hạn như các quyết định quản lý tồi. Mức độ đe dọa dưới dạng thiệt hại tiềm năng cho hệ thống cũng cần được xem xét và tính toán.
- **Đánh giá các đe dọa:** mức độ cao, thấp, vừa. Đe dọa cao là mối đe dọa lớn đến hệ thống có thể bị tổn thất nghiêm trọng nếu tình huống xấu nhất xuất hiện. Vừa có nghĩa là hệ thống có thể bị thất thoát trong những trường hợp tồi nhất nhưng vẫn có thể chịu đựng được mà không ảnh hưởng đến nền kinh doanh. Thấp có nghĩa là hệ thống có thể dự kiến được mỗi đe dọa và chuẩn bị được một số phương tiện để ngăn cản.
- **Xác định tình trạng đe dọa:** Sau khi thấy được các mối đe dọa có thể có, nhóm kiểm tra có thể kiểm tra lại xem những đe dọa này xuất hiện như thế nào. Điều này bao gồm việc dùng mô hình DFD, theo dõi ngược lại điểm hở, rà soát các hoàn cảnh được biểu thị bởi từng quá trình và lỗi tiềm năng từ mỗi dòng dữ liệu. Giai đoạn này của việc phân tích điều khiển đòi hỏi rất nhiều trí tưởng tượng và óc sáng tạo. Một khía cạnh khác cần kiểm tra tại giai đoạn này là xác suất xuất hiện tình huống đe dọa. Thông tin này, cùng với các chi tiết trước đây về "mức độ đe dọa" có thể làm cho nhóm kiểm soát quyết định

được về tầm quan trọng của mỗi nguy hiểm và giúp cho họ quyết định được tầm mức kiểm soát cần thực hiện.

b) **Thiết kế các kiểm soát cần thiết:** Sau khi đã nắm chắc được mức độ thiệt hại phát sinh từ điểm hở, nhà thiết kế phải quyết định các kiểm soát vật lý để ngăn cản hoặc làm giảm thiểu thiệt hại này.

c) **Phân tích các nguy cơ thất thoát dữ liệu:** bao gồm việc phát hiện các điểm hở thường là các chỗ vào ra như các file, màn hình, phân tích các đe dọa từ chỗ hở như: phá hoại, lấy cắp gây sự lãng phí, làm sai lệch thông tin.

d) **Các mức bảo mật:**

- Bảo mật vật lý. Khoá, báo động
- Nhận dạng nhân sự
- Mật khẩu
- Tạo mật mã: biến đổi dữ liệu từ dạng nhận thức được sang dạng mã. Phương pháp này tốn kém khó bảo trì nhưng phù hợp cho việc truyền dữ liệu và giải mã.
- Bảo mật bằng gọi lại

e) **Phân biệt riêng tư (Privacy)**

Phân biệt quyền truy nhập khác nhau đối với người dùng và cho phép uỷ quyền

Biện pháp:

- Dùng tên mỗi người làm tiền tố cho mọi đối tượng

Cài đặt (Sequel):

2 thủ tục : Giao quyền (Grant của Sequel)

Rút quyền (Revoke của Sequel)

+ Giao quyền Grant

Đối tượng: Dữ liệu quyền là

- + Đọc (Read)
- + Chèn (Insert)
- + Loại bỏ (Delete)
- + Điều chỉnh giá trị thuộc tính (Update)
- + Thêm thuộc tính (Expand)
- + Loại tệp (DROP)
- + Tạo tệp chỉ dẫn (Index)

Chương trình : Quyền là RUN (chạy)

Dạng chung lệnh Grant:

GRANT <các quyền> ON <đối tượng> TO <danh sách người dùng>[WITH GRANT OPTION] {được uỷ quyền cho người khác}

Để chạy GRANT đưa thêm và CDL các quan hệ

- Quyền: (người cho, người nhận, tên quan hệ, Read, Insert, Delete, Expand, Drop (có/ không), Index, Update, Option (có/ không)).

Cập nhật: Tất cả, Không có gì, 1 số

+ Cập nhật (...) các thuộc tính được cập nhật

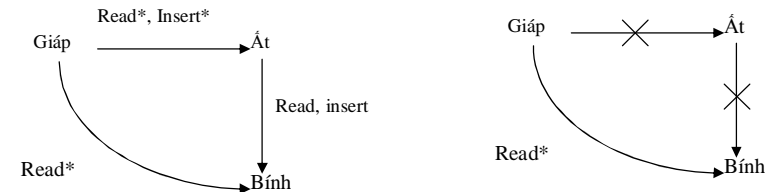
+ Chạy chương trình: Quyền thực hiện chương trình

Ví dụ: Sử dụng

Giáp: Grant read insert on hoá đơn to Át with Grant option

Át: Grant read insert on hoá đơn to Bính with Grant option

Giáp: Grant read insert on hoá đơn to Bính with Grant option



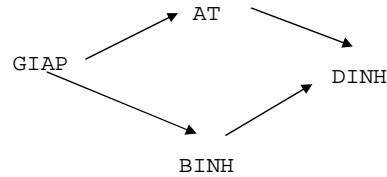
Dấu * là quyền trao trực tiếp cho người sử dụng.

- Rút quyền REVOKE

REVOKE <các quyền> on <đối tượng> from <danh sách người dùng>

Ví dụ : Giáp: Revoke Read, insert on hoá đơn from Át

Các trường hợp khúc mắc
+ N nguồn cho 1 quyền



+Ủy quyền vòng quanh
GIÁP → ÁT → ĐÌNH → BÌNH

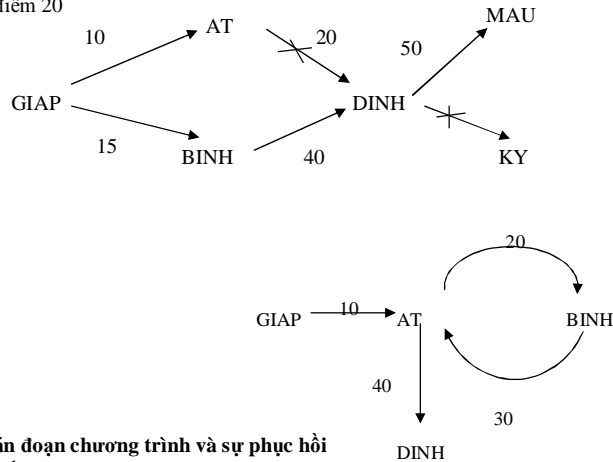
⇒ Cần thêm thông tin

Gán cho mỗi quyền 1 "con dấu" (thực chất giá trị ghi nhận thời điểm)

* Quy tắc rút quyền: Nếu A bị rút quyền mà A đã ủy quyền cho B thì B cũng bị rút quyền nếu B không bị nơi khác ủy quyền vào thời điểm trước khi A nhận được quyền đó.

Ví dụ:

a) Át rút quyền Đình ở thời điểm 20



b) Loại: Giáp - Át
Át - Bình
Bình - Át
Át - Đình

4. Nghiên cứu khả năng gián đoạn chương trình và sự phục hồi

a) Các gián đoạn chương trình

- Nguyên nhân:
 - Hỏng phần cứng
 - Giá mang tệp có sự cố
 - Môi trường
 - Hệ điều hành
 - Nhầm lẫn thao tác
 - Lập trình sai
- Hậu quả:
 - Mất thì giờ
 - Mất thông tin

b) Cài đặt các thủ tục phục hồi

- Chương trình theo mẻ (mất thời gian)
- Chương trình trực tuyến (on - line): phục hồi

Nguyên tắc phục hồi sao lục:

Khi chạy chương trình, bình thường định kỳ ghi lại 1 số biến mốc quan trọng.

Khi gián đoạn: Khởi động lại chương trình với giá trị biến mốc gần nhất.

Bài tập chương 8

8.1 Nêu vai trò của việc thiết kế kiểm soát và bảo mật hệ thống

8.2 Có thể tránh được mọi sai sót và rủi ro đối với hệ thống không? Cách lựa chọn và khắc phục như thế nào?

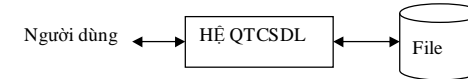
8.3 Hãy chỉ ra nguyên tắc phân quyền và ủy quyền đối với hệ thống

Chương 9 Thiết kế các file dữ liệu

1. Đại cương

a) Thiết kế dữ liệu phải dựa vào:

- Biểu đồ cấu trúc dữ liệu BCD như mô hình quan hệ, mô hình thực thể liên kết E-R, dựa vào biểu đồ luồng dữ liệu trong đó đặc biệt lưu tâm đến kho dữ liệu.
- Hệ Quản trị CSDL có sẵn.
- Khi thiết kế các file phải đảm bảo sao cho các dữ liệu phải đủ, không trùng lặp, việc truy cập đến các file dữ liệu phải thuận tiện, tốc độ nhanh.
- Mỗi hệ quản trị CSDL có ngôn ngữ định nghĩa dữ liệu.



Tuy nhiên khi cài đặt cụ thể để cho tiện lợi ta có thể bổ xung thêm một số thuộc tính tính toán, lặp lại một số thuộc tính, ghép một số thực thể thành một file....

b) File: Người dùng phải biết tổ chức file của mình, đương nhiên có hệ quản lí file đầu sao nó cũng chỉ giúp quản lí file chứ không phải quản lí CSDL.

Fox, Access cũng mới chỉ là hệ quản lí file.

Nếu có máy tính lí tưởng (tốc độ I/O tương ứng CPU) thì không cần phải làm gì chỉ thực thể và liên kết thì ta xây dựng được các các file. Vấn đề làm sao để truy nhập các file nhanh và thuận tiện.

Chú ý: Nhiều khi đã đạt chuẩn 3 NF nhưng để nhanh tiện, 3 NF có thể bị phá vỡ.

2. Phương pháp thực hiện:

Từ BCD để nhanh và thuận tiện ta thực hiện các bước sau:

- Thêm những thuộc tính tình huống (thường là tính toán được, tích lũy được)
- Gộp các kiểu thực thể, kiểu liên kết vào 1 file (có thể dư thừa) để bớt số lần truy nhập, tách thành nhiều file vì không phải bao giờ cũng dùng hết các kiểu thực thể liên kết trong một lần truy nhập.
- Lặp lại các thuộc tính từ file khác.
- Lập các file chỉ dẫn (Index) để truy nhập được nhanh, căn cứ vào xử lí (nhu cầu sử dụng).

3. Đưa thêm các thuộc tính tình huống và lặp lại các thuộc tính từ các file khác:

Các thuộc tính tình huống là các thuộc tính tính toán hoặc các thuộc tính tích lũy:

Ví dụ:

Tính toán: Thành tiền = số lượng * đơn giá

Tổng hợp đồng = ∑ thành tiền

Tích lũy Số dư tiết kiệm, lượng hàng tồn kho, số dư tài khoản.

Nhiều khi ta phải lập những file tình huống và chấp nhận sự dư thừa.

4. Nghiên cứu các đường truy nhập

Mỗi một đường truy cập gắn liền với chức năng xử lí khi ta thấy có yêu cầu truy nhập bằng cách xem lại BLD.

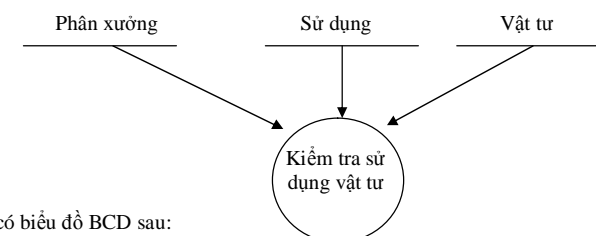
Mỗi xử lí ta cần chỉ ra các vấn đề sau:

- Truy nhập file nào ?
- Sử dụng khoá nào ?
- Tra cứu gì ?
- Tần số truy nhập

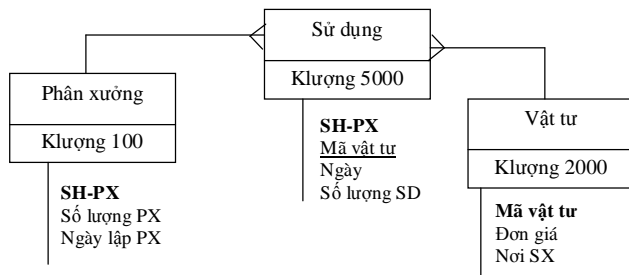
Nếu khoá và tra cứu trong cùng 1 file ta nói là truy cập trực tiếp. Còn các trường hợp còn lại nói chung là truy cập gián tiếp. Việc truy cập gián tiếp thông qua đường truy cập bằng cách lần theo các mối liên kết 1 - nhiều.

Ví dụ:

Xét xử lí: Kiểm tra sử dụng vật tư ta có một phân biểu đồ BLD sau đây:



Tương ứng ta có biểu đồ BCD sau:



Hãy xét 3 yêu cầu truy nhập tương ứng các câu hỏi sau:

- A - Tìm Soluong PX của 1 PX cho biết SH-PX
- B - Tìm đơn giá của các vật tư sử dụng bởi 1 PX biết SH-PX.
- C - Tìm soluong PX của các PX đã sử dụng 1 vật tư đã cho, biết mã VT

Mỗi yêu cầu tạo ra 1 đường truy nhập gồm nhiều bước:

- File gì?
- Tên đường truy nhập (A, B, C)
- Bước số mấy?
- Tra cứu gì?
- Tần số

Yêu cầu A:

Thực hiện 1 bước: A/1 truy nhập vào file "Phân xưởng"

Khoá: SH - PX

Tra cứu soluong PX

Tần số : (50 lần / ngày)

Yêu cầu B thực hiện 2 bước:

* B/1

Truy cập từ file "sử dụng"

Với khoá SH-PX

Tra cứu mã VT

Tần số 150 lần/ ngày, mỗi lần 50 bản ghi (5000/100, trung bình một phân xưởng sử dụng 50 lần)

* B/2

Truy cập từ file "Vật tư"

Khoá: mã VT

Tra cứu: đơn giá

Tần số 7500 lần/ ngày = (1500 × 50)

Yêu cầu C

* C/1

Truy cập file "sử dụng"

Khoá: mã VT

Tra cứu SH-PX

Tần suất 20 lần/ ngày, mỗi lần 2,5 bản ghi (=5000/2000).

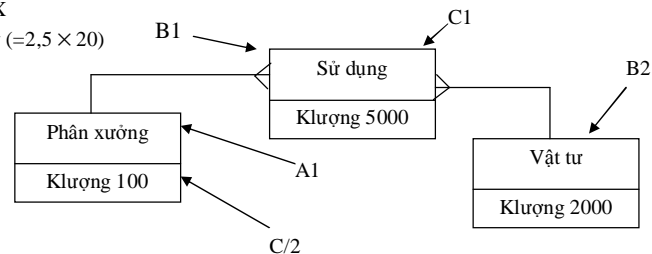
* C/2

Truy cập từ file "Phân xưởng"

Khoá: SH-PX

Tra cứu số lượng PX

Tần số 50 lần / ngày (=2,5 × 20)



5. Chuyển mô hình thực thể liên kết (hay mô hình quan hệ) thành các file

Nguyên tắc: nói chung mỗi 1 kiểu thực thể liên kết thành 1 file (thêm thuộc tính tình huống)

Khi cần có thể phân rã một thực thể thành những cụm thực thể hay dùng đối với những quan hệ quá lớn. Ngược lại có thể gộp các thực thể thành 1 file để hạn chế những đường truy cập gián tiếp, tất nhiên nó sẽ phá vỡ tính chất chuẩn hoá.

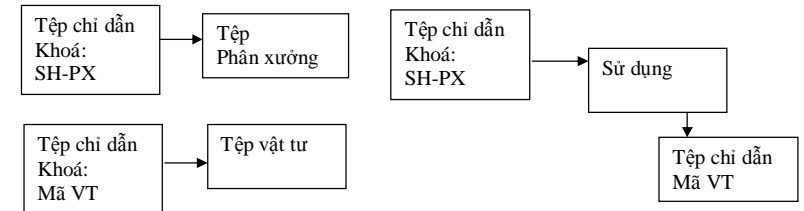
Các phương pháp truy cập để lập file chỉ dẫn:

- Tuần tự
- Trực tiếp
- Hàm trực tiếp: mỗi giá trị của khoá là 1 địa chỉ (hay gây sự dụng độ và giải quyết bằng móc nối ra vùng tràn).
- Tuần tự có chỉ dẫn: các phần tử đặt liên tiếp trên giá mang, có tổ chức, các chỉ dẫn để truy nhập trực tiếp.
- Móc nối (pointer): ở bộ nhớ ngoài, các phần tử kế tiếp không liền kề, các phần tử tự do cũng móc nối với nhau.

Hạn chế: phải tổ chức trên nền có sẵn do đó chỉ đối chiếu tương đối....

Lập file chỉ dẫn căn cứ vào đường truy nhập:

Trở lại ví dụ trên



Chú ý: Các đường truy nhập thông qua

- Thuộc tính kết nối
- Mối liên kết 1 - nhiều không được vật lí hoá trong mô hình quan hệ (chỉ trong mô hình mạng và phân cấp).

Bài tập chương 9

9.1 Khi thiết kế các file dữ liệu ta dựa vào biểu đồ nào. Các căn cứ nào cho ta xác định các thuộc tính của file: Tên file, Tên thuộc tính, các khoá và thuộc tính kết nối..

9.2 Thiết kế các file trên hệ quản trị cơ sở dữ liệu như FOX, ACCESS có phải là thiết kế mô hình thực thể liên kết E-R không? Tại sao.

9.3 Các đường truy cập vào file dựa vào liên kết nào của mô hình thực thể liên kết E-R?

9.4 Tại sao khi thiết kế các file đôi khi người ta phá vỡ chuẩn hoá 3NF? Điều đó có gây nên những lỗi cảm không? Cho ví dụ minh hoạ.

9.5 Mục đích của file chỉ dẫn để làm gì? Các kỹ thuật xây dựng file chỉ dẫn. Khi xây dựng các file chỉ dẫn ta chịu thêm chi phí gì (những nhược điểm của nó)

9.6 Thiết kế file dữ liệu và lựa chọn phần mềm là nhiệm vụ của người phân tích thiết kế hay người lập trình

9.7 Thiết kế các file dữ liệu cho hệ thống:

- Hệ thống tuyển sinh
- Hệ thống quản lý học tập
- Hệ thống quản lý thư viện
- Hệ thống kinh doanh các thiết bị máy tính
- Hệ thống quản lý khách sạn
- Hệ thống quản lý xe máy (có lưu lại chủ cũ sử dụng)

Chương 10. Thiết kế chương trình

10.1. Đại cương thiết kế chương trình

Thiết kế chi tiết bao gồm các thiết kế:

- Giao diện
- Kiểm soát
- Tập (File CSDL)
- Chương trình.

Như vậy thiết kế các module chương trình là công việc chính của giai đoạn thiết kế chi tiết. Trong kết quả phân tích thiết kế đến nay ta đã có BLD của hệ thống diễn tả các chức năng xử lý logic của hệ thống đồng thời liên quan thừa kế dữ liệu, còn chương trình là liên quan điều khiển và cơ sở dữ liệu đã thiết kế ở chương 9

Ngoài ra các chức năng khác như sau cũng cần được thể hiện trong thiết kế chương trình :

- Chức năng đối thoại
- Chức năng xử lý lỗi
- Chức năng xử lý vào/ ra
- Chức năng tra cứu CSDL
- Chức năng Module điều hành

Chú ý rằng trong phần này ta quan tâm thiết kế nội dung chương trình mà không phải viết chương trình cụ thể, vì nhiệm vụ này là của người lập trình viên. Người lập trình khi có bản thiết kế trong tay không nhất thiết phải hiểu cả hệ thống mà lập trình theo thiết kế được giao.

Nội dung chủ yếu trong giai đoạn này

Xác định cấu trúc tổng quát

- | | |
|-----------|---|
| Tổng quát | <ul style="list-style-type: none"> - Phân định các Module CT - Xác định mối liên quan giữa các Module đó (thông qua lời gọi và các thông tin trao đổi) - Đặc tả các Module chương trình - Gộp các Module thành chương trình (Module tài) - Thiết kế các mẫu thử (Test CT, chú ý đây cũng là việc của người thiết kế) |
|-----------|---|

10.2. Module chương trình

a) Định nghĩa: Module chương trình có thể hiểu dưới các dạng sau

- 1 Chương trình con: Dạng Procedure, Function, Subroutine....
- 1 cụm lệnh trong chương trình
- hoặc những ngôn ngữ dùng có UNIT, CLASS, OBJECT...

b) Các thuộc tính của mô đun chương trình

Tóm lại 1 Module CT có 4 thuộc tính cơ bản

- | | |
|-----------------|--|
| Đặc trưng ngoài | <ul style="list-style-type: none"> • Vào: thông tin từ CT gọi nó, Ra: Thông tin trả lại cho CT gọi nó • Chức năng hàm biến đổi từ vào → ra |
| Đặc trưng trong | <ul style="list-style-type: none"> • Cơ chế: Phương thức cụ thể để thực hiện chức năng trên • Dữ liệu cục bộ: chỗ nào nhớ, dữ liệu dùng riêng cho nó |

Đặc trưng ngoài: Các module gọi nó chỉ cần biết đặc trưng này

Đặc trưng trong thể hiện sự cài đặt của Module

Việc tách đặc trưng ngoài và đặc trưng trong để tạo độc lập cho sự cài đặt module đối với những module ngoài nó.

Các loại chương trình thường có trong hệ thống quản lý:

- Chương trình đơn chọn (menu program)
- Chương trình nhập dữ liệu (data entry program)
- Chương trình biên tập kiểm tra dữ liệu vào (edit program)
- Chương trình cập nhật dữ liệu (update program)
- Chương trình hiển thị, tra cứu (display or inquiry program)
- Chương trình tính toán (compute program)
- Chương trình in (print program)

c) Thiết kế cấu trúc

Thiết kế có cấu trúc là phương pháp tiến hành phân định các module theo kiểu trên xuống và làm mịn dần từng bước

Phân ánh LT có cấu trúc, tuy nhiên có khác biệt

- Trong LT có cấu trúc
 - Hướng tới các phương tiện của ngôn ngữ lập trình (mịn dần)

- Mức trên viết CT bằng ngôn ngữ LT có xen thêm ngôn ngữ pseudo code thay cho lời gọi sau này như vậy tại 1 bước nào đó mỗi module đã được đặc tả
- LT có cấu trúc mịn dần nhưng không có chỉ rõ phương pháp mịn dần như thế nào không có hướng dẫn từ mức này xuống mức kia

- Thiết kế có cấu trúc

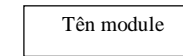
- Phân định module về logic
- Chỉ mô tả như những cái vào/ ra, chuyển giao dữ liệu, chứ nội dung chưa được đề cập.
- Có hướng dẫn các phân định và ý nghĩa của module

10.3. Công cụ để diễn tả cấu trúc CT (lược đồ cấu trúc (LCT))

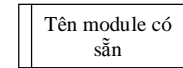
Lược đồ cấu trúc: LCT là công cụ ở đây hết sức thô sơ, thô sơ một cách cố tình để trừu tượng hoá nhằm đi tới cách viết các chương trình cụ thể và chi tiết hơn

a) Biểu diễn các module

Module được biểu diễn bằng hình chữ nhật trên có ghi nhãn là tên module.

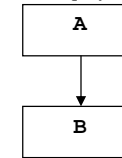


Trường hợp đặc biệt module đã có sẵn ta biểu diễn thêm hai đường gạch dọc



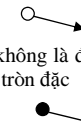
b) Kết nối các module: thể hiện bằng lời gọi.

A gọi B, B thực hiện chức năng của mình rồi quay về A ở vị trí sau lời gọi



c) Thông tin chuyển giao giữa các module:

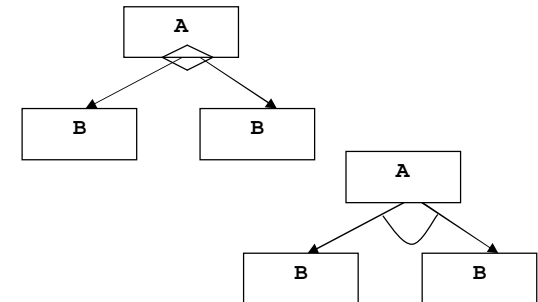
Các module chuyển giao bằng dữ liệu và điều khiển
Dữ liệu chuyển giao kí hiệu mũi tên và đầu tròn rỗng



Những thông tin điều khiển (không là đối tượng để xử lý mà dùng trong quá trình điều khiển thực hiện chương trình). Kí hiệu mũi tên và đầu tròn đặc

d) Một số trường hợp đặc biệt

- Chọn lựa gọi B hay C



- Lặp A gọi B nhiều lần

VD

Note: Nhận xét những module phía trên là module điều khiển càng đi xuống tính chất điều khiển giảm dần, thực sự xử lý, biến đổi thông tin

Nếu triển khai thêm xuống dưới sẽ xuất hiện những module chỉ chế biến thông tin và được gọi từ nhiều module khác.

Trên "xoè ra" Dưới "chụm vào"

10. 4. Chất lượng của lược đồ cấu trúc (LCT)

Một trong những nguyên tắc cơ bản của việc thiết kế có cấu trúc đó là từ một hệ thống lớn ta phân thành từng module có thể quản lý được. Tuy nhiên, điều quan trọng là việc chia nhỏ nên thực hiện theo một cách mà các module đó thể độc lập với nhau. Các module này có thể tương tác (coupling) hoặc là cố kết (cohesion) với nhau.

a) Có sự tương tác (coupling):

Một trong những phạm vi chất lượng thiết kế là sự tương tác, tức là độ phụ thuộc giữa hai module với nhau. Đối tượng cần bàn ở đây là sự tương tác tối thiểu, tức là tạo một module có độ độc lập có thể được. Độ tương tác thấp giữa các module chỉ ra sự phân chia tốt trong hệ thống và các module có thể đạt được theo một trong ba cách sau:

- Lược bỏ những mối quan hệ không cần thiết.
- Giảm bớt các quan hệ cần thiết.
- Bỏ đi các mối quan hệ lỏng lẻo cần thiết.

Một trong những điểm chủ yếu của sự tương tác thấp là không có một module nào lo lắng về bất kỳ những chi tiết cấu tạo bên trong nó. Các module này có các chức năng và sự xuất hiện các chức năng bên trong nó như một hộp đen.

Tóm lại, sự tương tác thấp nhằm thỏa mãn:

- Sự kết nối giữa hai module càng ít càng tốt, sự thay đổi trong module này không làm ảnh hưởng đến module kia.
- Khi ta muốn thay đổi trong một module thì độ rủi ro rất thấp cần thay đổi module khác
- Khi quản lý một module, ta không lo lắng về những chi tiết bên trong của các module khác; tức là ta muốn hệ thống đơn giản và dễ hiểu.

10.4.1. Các nguyên tắc của sự tương tác:

Thực ra, làm giảm sự tương tác giữa các module tức là làm giảm đi sự kết nối phức tạp giữa hai module. Các nguyên tắc làm giảm sự tương tác gồm:

- Tạo các sự kết nối hẹp.
- Tạo các sự kết nối trực tiếp.
- Tạo các sự kết nối cục bộ (toàn cục).
- Tạo các sự kết nối rõ ràng.
- Tạo các sự kết nối mềm dẻo.

10.4.1.1 Các sự kết nối hẹp:

Độ rộng về sự giao tiếp giữa hai module là có nhiều kết nối cần thiết liên kết giữa hai module đó. Một sự tương tác hẹp giữa hai module (là tốt) là một cặp module chỉ có một mẫu dữ liệu kết nối với nhau duy nhất, và ngược lại nếu có nhiều mẫu dữ liệu kết nối giữa hai module là không tốt.

10.4.1.2 Các kết nối trực tiếp:

Giao tiếp giữa hai module là để nhận biết nhau nếu một người nào đó lĩnh hội được nó một cách trực tiếp mà không cần tham khảo tới nhiều mẫu dữ liệu khác nhau khi tiếp xúc lần đầu. Chẳng hạn, một module nói về chi tiết của một khách hàng nào đó (module CUST-DETAILS) mà đã được định nghĩa gồm có các mẫu tin như: tên khách hàng (CUST-NAME), số tài khoản khách hàng (CUST-ACCOUNT-NUM), địa chỉ khách hàng (CUST-ADDRESS), bản thanh toán của khách hàng (CUST-BALANCE), thì lúc đó ta dễ nhận biết chúng giao tiếp với thế giới bên ngoài bằng mẫu dữ liệu nào, có lẽ qua trường mẫu tin địa chỉ khách hàng (CUST-ADDRESS).

10.4.1.3. Các sự giao tiếp cục bộ (toàn cục):

Nếu tất cả các thông tin giao tiếp yêu cầu để hiểu biết về kết nối giữa hai module là chính nó, thì các thông tin đó được gọi là cục bộ. Thông tin về kết nối toàn cục là xuyên qua toàn mẫu dữ liệu. Trong trường hợp này, thông tin về sự kết nối giữa hai module có lẽ có hàng trăm cách móc nối khác nhau từ module đang gọi hoặc là module đã gọi.

10.4.1.4. Các sự kết nối rõ ràng:

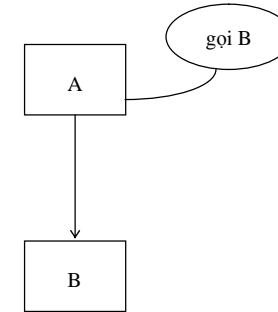
Sự kết nối rõ ràng giữa hai module là không có lặp lại, không có tính tối nghĩa. Ví dụ có một đoạn trình tự module A giao tiếp với module B bằng cách thay đổi nội dung trong đoạn trình B, điều này là sự kết nối không rõ ràng.

10.4.1.5. Sự kết nối mềm dẻo:

Bảo trì một hệ thống máy tính thường bao gồm nhiều thay đổi các liên kết trong số các module trong hệ thống.

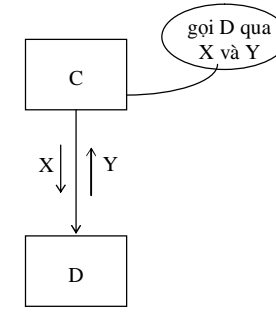
10.4.2. Tương tác bình thường:

Hai module, A và B gọi là tương tác bình thường nếu như A gọi được B và ngược lại B gọi được A, tất cả các thông tin truy cập giữa chúng là các tham số được gọi chính chúng. Tất nhiên, đây là mô tả trường hợp bình thường trong sơ đồ có cấu trúc. Hình vẽ 4.2 và 4.3 mô tả trường hợp trên



A và B tương tác bình thường với nhau, nhưng không có gì để nói về nhau

Hình 4.2



C và D tương tác bình thường với nhau, nhưng giao tiếp với nhau qua dữ liệu X và Y

Hình 4.3

Trong hình 4.2 A gọi tới B nhưng A không truy cập bất cứ điều gì tới B và cũng không nhận được bất cứ điều gì từ B. Trường hợp này đánh dấu một điểm zero trong tỉ lệ tương tác. A gọi là tương tác tới B khi và chỉ khi A là tên của B. (Như thể khi B thay đổi tên thì A cũng sẽ thay đổi theo).

10.4.2.1 Tương tác dữ liệu:

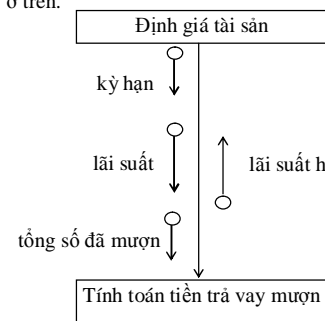
Trong hình 4.3 biểu diễn rộng hơn trong số các kiểu tương tác thông thường, tương tác bình thường cũng có nghĩa là tương tác dữ liệu.

Hai module gọi là tương tác dữ liệu nếu chúng giao tiếp với nhau bằng các tham số, mỗi tham số là một phần trong mẫu dữ liệu. Dữ liệu tương tác là sự giao tiếp cần thiết giữa nhiều module. Khi nhiều module phải giao tiếp với nhau thì dữ liệu tương tác là không thể tránh khỏi và dữ liệu tương tác này không làm ảnh hưởng đến các module miễn là nó được tối thiểu hoá. Ví dụ trong hình 4.4 có tất cả bốn mẫu dữ liệu tương tác gồm: tổng số mượn, lãi suất, thời hạn, lãi suất hoàn trả là cần thiết.

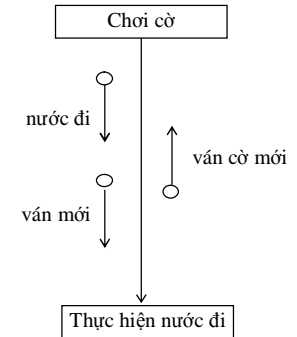
Mặt khác, các thông tin phụ trội khác không cần như thêm vào tên khách hàng làm tăng thêm độ phức tạp, không dùng để tính toán tiền trả nợ.

Tương tác dữ liệu thể hiện tất cả các đặc tính tốt nhất của sự tương tác. Nếu như ta giao tiếp giữa các module với nhau bằng những thông tin không cần thiết thì sự tương tác trở nên bị thu hẹp lại. Tương tác dữ liệu cũng có nghĩa là khi giao tiếp giữa hai module muốn gì được nấy, hay là các đoạn mã tương tác dữ liệu được thể hiện dọc theo các module gọi và các module chuẩn bị các module khác. Có hai điều cần chú ý trong sự tương tác dữ liệu:

- Với sự tương tác dữ liệu càng nhỏ là càng tốt.
- Với sự tương tác dữ liệu, trong trường hợp có nhiều module tương tác với nhau, thì những thông tin dư thừa (không rõ ràng) sẽ làm cho sự tương tác kém hiệu quả và vì phạm đến năm nguyên lý của sự tương tác ở trên.



Hình 4.4



Hình 4.6

10.4.2.2 Tương tác stamp

Thông thường hai module được gọi là tương tác stamp nếu như module này tương tác tới module khác nhờ vào dữ liệu kết nối chung, dữ liệu kết nối này có đầy đủ tính cấu trúc bên trong nó. Ví dụ: dữ liệu kết nối có thể là một bản ghi khách hàng gồm có nhiều trường, hình 4.5 thể hiện sự tương tác stamp.

Trong hình 4.6 có ba tham số: bàn cờ, nước đi, ván cờ mới, tất cả có đầy đủ tính cấu trúc, vì vậy mà chúng thể hiện sự tương tác stamp. Sự tương tác này xảy ra khi các dữ liệu cấu trúc lựa chọn có tính chất tự nhiên tới các ứng dụng và không có tính mật mờ. Chúng ta hãy xem kỹ trong hình 4.6, sự định nghĩa nước đi của bàn cờ. Khi có sự tương tác quanh co thì nên dùng sự tương tác dữ liệu hơn là dùng sự tương tác stamp.

Mặc dù, khi người thiết kế giỏi cảm thấy dùng sự tương tác stamp là tốt nhưng người thiết kế kém hơn thì cho rằng tương tác stamp là không tốt cho cùng một hệ thống, cho nên có hai khuyến nghị được nêu ra đây cho tương tác stamp:

- Dùng bao giờ truy cập tới các bản ghi có quá nhiều trường, tới các module mà chỉ một hoặc hai trường trong số các trường đó. Xem hình 4.7 mô tả ba module tương tác stamp với nhau

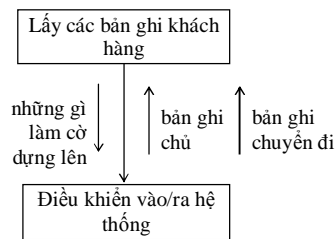
Bản ghi khách hàng thuê gồm có các trường: số bảng, thành phần cấu lạc bộ Marlin, số câu lạc bộ Marlin, xăng đã sử dụng, loại xe hơi, số dặm đi được, số ngày sử dụng... Mặc dù, module tính toán tiền thuê cơ sở chỉ yêu cầu ba trường cuối cùng, khi nó nhận tất cả các thông tin về khách hàng thuê. Bất kỳ sự thay đổi nào trong bản ghi về khách hàng thuê, hoặc là khuôn mẫu hoặc là cấu trúc bản ghi, sẽ làm ảnh hưởng tới tất cả các module tham trò tới nó, ngay cả các module không tham trò tới các trường thay đổi.

Như ví dụ đơn giản trên nhìn vào hình vẽ 4.7 khi mà trường số câu lạc bộ Marlin thay đổi về khuôn dạng, thì cả hai module tính toán tiền thuê cơ sở, module tính toán tiền phải trả cho loại ga sẽ phải thay đổi theo, hoặc là tối thiểu biên dịch lại mặc dù các module đó không tham trò tới các thành phần của cấu lạc Merlin.

- Nếu như ta muốn gói dữ liệu thành bó thì dùng sự tương tác stamp rất có hiệu quả.

10.4.2.3 Tương tác điều khiển

Hai module được gọi là tương tác điều khiển, nếu như module này truy cập tới module kia thông một mảnh thông tin kết nối và mảnh thông tin kết nối đó lại tham gia vào sự điều khiển logic của một module khác nữa. Hình 4.2.3 thể hiện hai module tương tác điều khiển với nhau.



Hình 4.2.3 Hai module tương tác điều khiển

Giá trị làm cho cờ dựng lên để chỉ ra rằng hệ thống đang được điều khiển đọc các bản ghi vào ra. Chẳng hạn khi cờ có giá trị bằng 1 có nghĩa là lấy bản ghi chủ kế tiếp, khi cờ bằng 2 thực hiện việc duy chuyển bản ghi kế tiếp, khi cờ bằng 3 thực hiện cả hai bước trên, khi giá trị bằng 4 có nghĩa là điều khiển hệ thống in các header... Trong hình 4.2.3, module lấy các bản ghi khách hàng quyết định một cách rõ ràng đến các thành phần điều khiển vào ra của hệ thống. Để mà một module gọi thực hiện một quyết định thì nó phải tính logic của module bị gọi tổ chức như thế nào. Chẳng hạn, để chọn đúng giá trị dựng cờ, thì module lấy các bản ghi khách hàng phải biết tính logic của hệ thống điều khiển vào ra. Khi hệ thống có nhiều module tương tác với nhau thì sự tương tác điều khiển không con thích hợp nữa, vì nó thường chỉ ra sự hiện diện của các module khác làm quan hệ trong hệ thống trở nên rối rắm và khó khăn cho việc thiết kế hệ thống.

10.4.3 Tương tác chung (common coupling)

Hai module được gọi là tương tác chung nếu chúng đều tham trò đến vùng dữ liệu toàn cục giống nhau

a) Nguyên tắc việc lựa chọn sự tương tác

- Chọn sự tương tác càng lỏng lẻo càng tốt
- Chọn sự tương tác càng đơn giản càng tốt
- Do sau này hệ thống sẽ phải sửa chữa đỡ "Rút giây động rừng" (xấu nhất)
- Tương tác nội dung: Module này can thiệp vào nội dung của module khác

- Tương tác điều khiển: module này chuyển 1 thông tin điều khiển cho 1 module khác (cờ,...) (khi gửi 1 thông tin điều khiển thực chất module cấp trên đã biết nội dung module cấp dưới như vậy vì phạm nguyên tắc "che dấu").

Cần thì vẫn phải dùng tương tác này song tránh nếu được (hạn chế)

- Tương tác dữ liệu: trao đổi dữ liệu cho nhau (cần chấp nhận tương tác này, tuy nhiên chọn tương tác này càng đơn giản càng tốt: chuyển giao qua các phân tích chuẩn: danh sách tham số)

b) Sự cố kết (Cohension): Sự gắn bó về mặt logic các phần trong nội bộ của module càng cao càng tốt (mỗi module chỉ nên giao 1 nhiệm vụ logic, dùng giao những nhiệm vụ phân tán)

c) Hình thái: Trên xoè ra → thể hiện sự tinh tế

Dưới chụm vào → thể hiện?

Ở mỗi điểm xoè ra chỉ nên 7 ± 2

Có 2 khái niệm

- Phạm vi điều kiện của 1 module: Module đó cùng với những module phụ thuộc (được gọi)
- Phạm vi ảnh hưởng của 1 quyết định: là mọi module (chịu ảnh hưởng) có sử dụng kết quả quyết định đó

VD

- Chẳng hạn trong B có 1 quyết định q1 mà kết quả của nó AEF thì phạm vi ảnh hưởng:

AEF

Phạm vi điều kiện của A là A, B, C

- Một thiết kế tốt thì:
 - o Phạm vi ảnh hưởng nằm trong phạm vi điều khiển
 - o Các quyết định có miền ảnh hưởng càng bé càng tốt

10.5. Cách thức chuyển BLD thành LCT:

Thực chất chuyển BLD của hệ thống con thành BLD công đoạn (Job) ở mức bé nhất

Có 2 phương thức định hướng cho việc chuyển BLD thành LCT

- Phương thức theo biến đổi (Transform analysis)
- Phương thức theo thao tác (Transaction analysis)

Hai phương thức này không đối lập và có thể kết hợp với nhau

Ở đây chúng ta chỉ đưa ra những gợi ý, định hướng cho nhà phân tích thiết kế

a) Phương thức theo biến đổi: Dựa theo sự phát hiện trung tâm biến đổi thông tin chủ (tính toán, kết xuất)

Trung tâm như vậy: có tính chất

- Các phần còn lại: sẽ bị cắt rời không còn liên kết được với nhau sau khi ta cắt đi trung tâm biến đổi nếu "xách" trung tâm biến đổi lên sẽ kéo theo phần còn lại
- Thượng lưu: luồng thông tin vào
- Hạ lưu: luồng thông tin ra

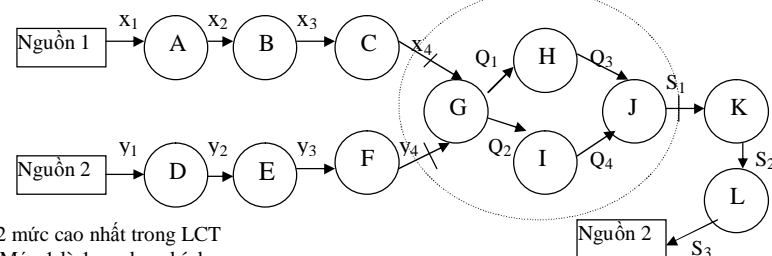
Có 5 bước thực hiện

(1) Dời theo luồng dữ liệu vào (thượng lưu) vượt qua các chức năng biến đổi thông tin sơ bộ cho đến khi dữ liệu được biến đổi trừu tượng nhất hoặc đến lúc không xem nó là dữ liệu vào được nữa thì chúng ta ngắt (đánh dấu) luồng vào từ vị trí đó

(2) Xác định nguồn dữ liệu ra, đi ngược dòng vượt qua các chức năng chế biến dạng thông tin cho đến khi không xem được đó là dữ liệu ra, thì dừng lại và đánh dấu...

(3) Căn cứ vào các điểm đánh dấu khoanh vùng để cô lập trung tâm biến đổi

Ví dụ :



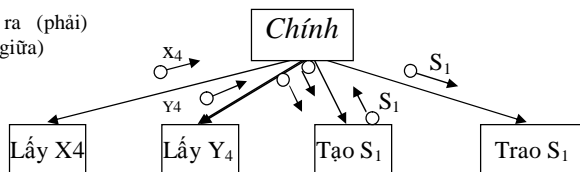
(4) Vẽ 2 mức cao nhất trong LCT

Mức 1 là 1 modul chính

Mức 2 tiếp theo gồm 3 modul

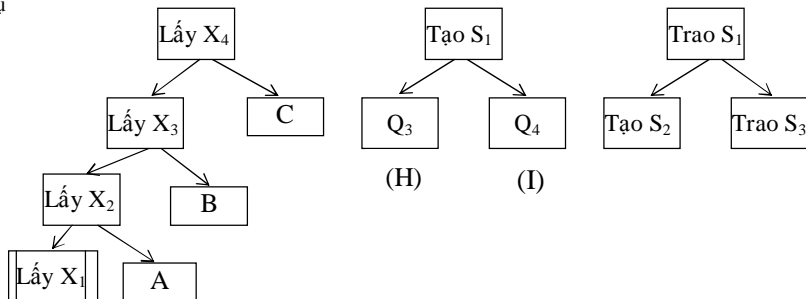
1 modul vào cho mỗi luồng dữ liệu vào (trái)

1 modul ra cho mỗi luồng dữ liệu ra (phải)
và 1 modul thông tin biến đổi (giữa)
Quay lại Ví dụ trên



(5) Triển khai mỗi module (vào, ra, biến đổi) ở mức trên thành mức thấp hơn làm xuất hiện dần các module tương ứng với chức năng xử lý trong BLD

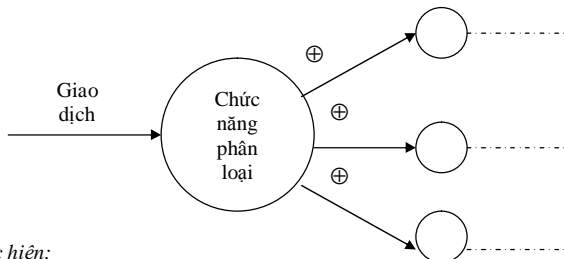
Ví dụ



b) Phân tích theo thao tác (giao dịch): [Transaction Analysis] Đó là các thông tin mà khi xuất hiện thì nó khởi động một loạt các chức năng trong BLD. Một giao tác bao gồm:

- Các sự kiện trong môi trường hệ thống (event)
- Tác nhân kích thích (stimulus)
- Các hành động (activity)
- Các phản ứng, đáp ứng của hệ thống (response)
- Những kết quả, ảnh hưởng của giao tác (effect)

VD: Đơn hàng đến khởi động một loạt các chức năng; đặc điểm là luôn có một chức năng phân loại thông tin giao dịch



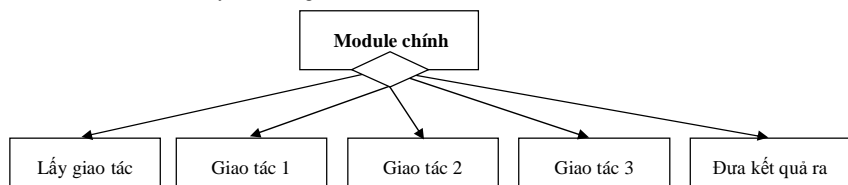
Các bước thực hiện:

- (1) Phát hiện 1 chức năng xử lý trong BLD: nhận 1 luồng dữ liệu vào và cho ra nhiều dữ liệu loại trừ lẫn nhau
- (2) Xác định các loại giao tác khác nhau tương ứng với các luồng ra của chức năng nói trên và các chức năng được khởi động từ các giao tác đó
- (3) Vẽ LCT ở 2 mức cao nhất

Mức 1: 1 module chính

Mức 2: 1 module cho mỗi loại giao tác và các module giao tác này được module chính gọi qua phép chọn. Cũng có thể thêm các module lấy các thông tin vào/ra

VD

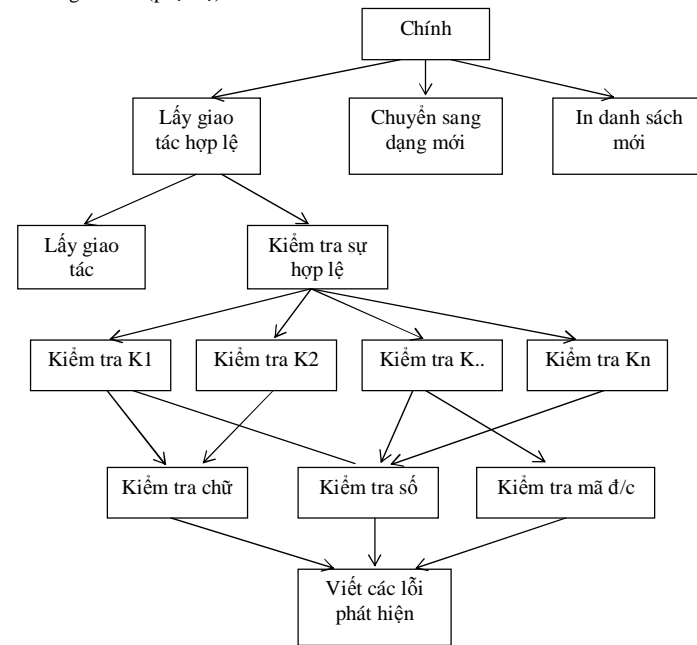


(4) Triển khai các module xuống mức thấp

Các mức thấp hơn có thể phối hợp theo cả hai phương pháp

+ Phân tích theo biến đổi chính

+ Phân tích theo các giao tác (phụ trợ)



c) Cấu trúc lại hệ thống:

Xem lại toàn bộ hệ thống xem có phù hợp với các yêu cầu đề ra hay không để chỉnh lý kịp thời

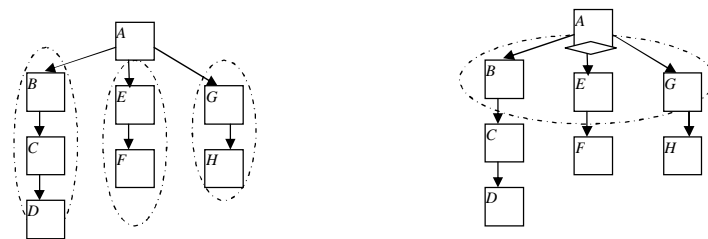
10.6. Đóng gói thành module tải

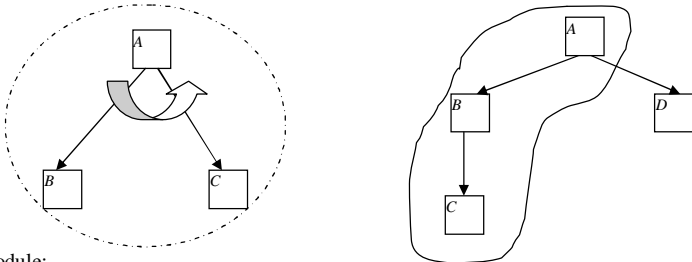
Đây là giai đoạn cuối của khâu thiết kế các module để dẫn đến lập trình được. Ta có thể coi LCT là 1 chương trình cũng được. Nhưng thường chương trình như vậy lớn quá nên có nhu cầu đóng gói, tải dần từng module vào bộ nhớ trong

Có một vài cách đóng gói

- Đóng gói theo dòng dữ liệu vào (đóng gói theo các phạm vi điều khiển) có hình dáng chế dọc lược đồ, chuyển giao theo nguồn dữ liệu hoặc
- Đóng gói chế ngang theo mức LCT thường đối với các module lựa chọn.
- Đóng gói theo 1 Thư viện CT
- Đóng theo Module gọi lặp thường xuyên và ghép chung vào module gọi

Nếu phép chọn buộc phải cắt ra thì nên khảo sát phép chọn cân đối hay không, gộp nhánh được gọi luôn (nhánh nặng thoả điều kiện nằm ngay sau if) vào chương trình con





Đặc tả các module:

Đặc tả các module nhằm đề cập đến nội dung chi tiết của từng module bằng một ngôn ngữ giải thuật nào đó chẳng hạn

- Sơ đồ khối (flowchart)
- Ngôn ngữ giả trình (Pseudo code).

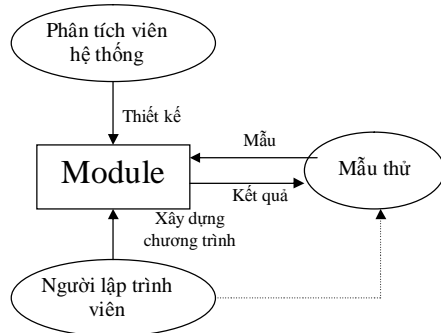
Dựa trên đặc tả này người xây dựng chương trình sẽ mã hoá thành các chương trình ứng dụng một cách dễ dàng. Phương pháp và kỹ thuật đặc tả các module được đề cập đến trong các môn học trước: Tin học đại cương, cấu trúc dữ liệu và giải thuật, Kỹ thuật lập trình, Công nghệ phần mềm...

10.7. Lập các mẫu thử (test)

Người thiết kế hệ thống sau khi thiết kế các module còn có trách nhiệm thiết kế và đưa ra các mẫu thử nhằm đảm bảo tính khách quan. Các mẫu thử này chính là các yêu cầu người lập trình phải đảm bảo thực hiện đúng các chức năng và yêu cầu khái quát của hệ thống cũng như các yêu cầu chi tiết của từng module chương trình.

Test : -Từng chương trình

-Toàn bộ hệ thống



Hiện nay "test" gần như là biện pháp duy nhất để kiểm tra chương trình. Về lý thuyết chúng ta đã biết là có các phương pháp chứng minh sự đúng đắn, độ phức tạp, thời gian thực hiện và không gian lưu trữ, cũng như tính hiệu quả của chương trình nhưng các công cụ này hiện chưa khả thi về ứng dụng. Như Diskjstra đã phát biểu: "Mẫu thử chỉ chứng minh sự có mặt của lỗi chứ không chứng minh được sự vắng mặt của lỗi"

a) Các loại mẫu thử

1. Loại mẫu thử hoàn chỉnh / không hoàn chỉnh

Mẫu thử hoàn chỉnh bảo đảm dự kiến mọi trường hợp có mặt trong chương trình. Mẫu thử không hoàn chỉnh khi ta chỉ cần kiểm tra các điểm mấu chốt quan trọng, còn các phần thứ yếu, không quan trọng có thể cho phép bỏ qua không ảnh hưởng sai lệch đến tính chất của hệ thống cũng như từng module riêng lẻ

2. Loại mẫu thử Ngẫu nhiên / không ngẫu nhiên.

Trước tiên ta nên thử không ngẫu nhiên, sau đó tiến hành những mẫu thử ngẫu nhiên. Có nhiều cách sinh các mẫu ngẫu nhiên; thường sinh theo luật xác suất Baux hoặc phương pháp Von Newman

Ví dụ: Lấy dữ liệu 4 con số đặt là x_0

Sau đó lấy 4 con số ở giữa của bình phương x_0 (x_0^2) đặt là x_1 .

Cứ tiếp tục như vậy với các x_i

Chẳng hạn $x_0 = 1147$
 $x_0^2 = 1315609$ $x_1 = 3156$
 $x_1^2 = 98012763$ $x_2 = 0127$

Dãy Fibonacy: $F(n+2) = F(n+1) + F(n)$

- Phương pháp thương: Lấy 2 số A, B rất lớn

$$x_{i+1} = A * x_i - B * q, \quad q: \text{thương số của phép chia.}$$

Như vậy ta xem x_{i+1} là số dư của phép chia $A * x_i$ với B. Đây này, ngẫu nhiên và tuần hoàn

- Chọn ngẫu nhiên (chữ, chữ pha số)

3. Mẫu thử đa dạng, phong phú và đủ lớn

b) Trình bày mẫu thử:

- Mẫu thử có thể được trình bày theo bảng có dạng sau

Mẫu thử (1)		Kết quả thu được (2)			Kết quả mong đợi (dự đoán) (3)			Sai lệch thực tế giữa (2) và (3)		Nhận xét	
D	d2			kn	k1	k2		kn	%		%
1											

-Mẫu thử có thể sinh bằng các "bộ sinh" tự động bằng cách chỉ ra công thức sinh

c) Các cách thử chương trình bằng mẫu thử

- Thử tính đúng đắn.
- So kết quả thu được với kết quả chờ đợi
- Nếu trong quá trình phức tạp, yêu cầu chương trình in các trị trung gian.
- Kiểm tra các giá trị trung gian.
- Kiểm tra vết chương trình.
- Thử hiệu năng: các mẫu thử, lớn, phải cho 1 thời gian để thực hiện

Bài tập chương 10

1. Từ biểu đồ luồng dữ liệu hãy xây dựng lược đồ cấu trúc chương trình cho hệ thống :

- Tính lương
- Check out cho khách
- Giao dịch mượn trả sách

2. Thông tin bàn giao giữa các module là gì, chỉ ra các nguyên tắc cụ thể

3. Trong các hệ thống hệ quản trị CSDL các tương tác giữa các module có xảy ra hay không? Cách khắc phục

4. Các phương pháp thử đánh giá hệ thống ở một số đặc tính sau

- + Đúng đắn và ổn định của module chương trình
- + Tính thời gian thực hiện
- + Độ phức tạp
- + Tính thân thiện
- + Tính dễ sửa chữa
- + Tính mở

Hãy bàn luận về các tính chất trên để làm rõ các nguyên tắc cơ bản khi thiết kế

Chương 11. Lập trình - Chạy thử - Bảo trì

1. Lập trình

a. Thành lập tổ lập trình

Tổ lập trình là một nhóm tham gia việc viết các Môđul và được lắp ghép thành hệ thống. Việc thiết kế hệ thống càng chi tiết bao nhiêu và mang tính hệ thống cao sẽ giúp cho việc thực hiện cài đặt và phát triển hệ thống hoàn thiện bấy nhiêu.

- Một chương trình ứng dụng trung bình có từ 8000 đến 15.000 câu lệnh và trung bình người ta có thể viết được 30 câu lệnh 1 ngày.

- Từ cơ sở trên tạo nhóm lập trình bao gồm bao nhiêu người trong khoảng thời gian bao lâu.

b. Chọn ngôn ngữ lập trình

Những ngôn ngữ mang tính hệ thống viết được ra môi trường thường dùng là C, C++, Pascal và môi trường chuyên dùng: Cobol, Fox, Access, VB, Lotus Notes. Môi trường điển hình hiện nay là: HQT CSDL (ORACLE).

c. Cài đặt các tệp, biết các đoạn chương trình chung

d. Soạn thảo chương trình cho từng đơn vị xử lý

- Yêu cầu đối với các chương trình:

- + Vào ra phải đúng đắn
- + Dễ đọc, dễ hiểu để còn bảo trì
- + Dễ sửa, dễ nâng cấp
- + Chạy phải nhanh, tiết kiệm bộ nhớ có hiệu quả không gian, thời gian.
- + Tối ưu hoá về mã: thể hiện ở thời gian và chỗ chiếm bộ nhớ.

2. Chạy thử và ghép nối

Chạy thử và ghép nối để cho ra một mẫu thử hệ thống

3. Thành lập các tài liệu hướng dẫn sử dụng

Tài liệu hướng dẫn đóng vai trò quan trọng với người sử dụng

a. Đại cương

Mục đích của tài liệu là để trao đổi, liên lạc. Nhà phân tích tham gia phát triển hệ thống cần trao đổi với một số người trước, trong và sau tiến trình phân tích và thiết kế đã được thảo luận ở đây. Thông tin thu được cần phải được ghi lại theo khuôn dạng làm thuận tiện cho việc thâm nhập và tìm kiếm. Kết quả của hoạt động phân tích và các ý tưởng được xem xét trong giai đoạn thiết kế (cả những ý tưởng được chấp thuận cũng như bị loại bỏ) đều cần được thu tóm dưới dạng văn bản nào đó, trước hết để giúp làm đầy đủ tiến trình phát triển rồi thử nữa để hỗ trợ cho việc chạy và bảo trì hệ thống khi nó đi vào hoạt động.

Về cơ bản có hai khuôn dạng tài liệu. Chúng liên quan tới hai nhóm người tham gia trong việc phát triển, và các nhu cầu thông tin khác nhau:

+ **Người dùng.** (Thuật ngữ được dùng ở đây bao hàm cả nhà quản lý, người chủ và người vận hành hệ thống). Tài liệu cho những người này phải được chuẩn bị một cách chính thức bởi nhóm phát triển (một số trong họ cũng chính là người dùng). Tài liệu này được xem như một phần của việc bàn giao hệ thống. Trong phương pháp luận Systemscraft, các tài liệu bàn giao bao gồm

- Đặc tả yêu cầu nghiệp vụ
- Đặc tả thiết kế hệ thống
- Tài liệu cho người dùng
- Hướng dẫn vận hành

+ **Người phát triển.** (Thuật ngữ được dùng ở đây bao hàm cả nhà phân tích, người thiết kế, người làm bản mẫu, người lập trình, người quản lý dự án, chuyên gia CSDL... đã tham gia vào tiến trình phát triển. Ta cũng có thể kể cả một số người dùng có tham gia nhiều vào phát triển hệ thống) Tài liệu cho những người này trong suốt thời kỳ nghiên cứu. Các tài liệu này thường được gọi là *Hồ sơ giấy tờ làm việc*.

b. Các hướng dẫn chung

1. Phân cứng và phần mềm ứng dụng.
2. Hướng dẫn về các phương thức khai báo
3. Về các người sử dụng
4. Các hướng dẫn dùng khác

c. Giới thiệu chương trình, trình tự khai thác

1. Danh sách các chương trình
2. Mô tả chi tiết
3. Trình tự khai thác

d. Đặc trưng của các đầu vào: đưa ra các mẫu

e. Đặc trưng của các tệp

1. Đặc trưng chung
2. Cấu trúc tệp

e3. Các tệp chi dẫn

f. Đặc trưng của các đầu ra

1. Đặc trưng chung
2. Cấu trúc lúc trình bày

g. Hướng dẫn cho các nhân viên điều hành hệ thống

4. Bảo trì hệ thống

- Song song với quy trình kiểm tra thì ta phải tiến hành bảo trì hệ thống.

- + Sửa các lỗi
- + Điều chỉnh theo yêu cầu mới
- + Cải thiện hiệu năng của hệ thống. Muốn vậy ta phải hiểu được chương trình từ những tài liệu để lại, phải lần ngược đầu vết khi phát hiện lỗi.

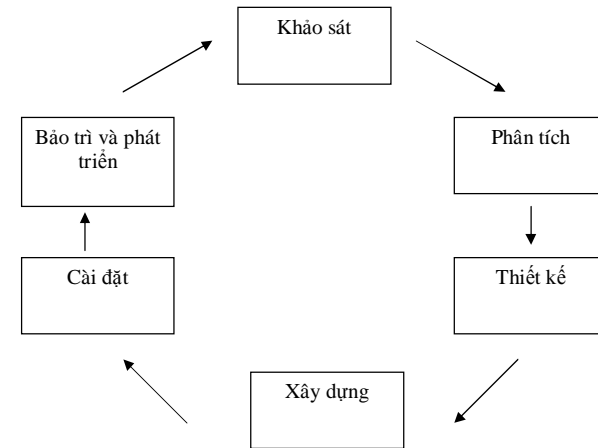
- Bảo trì gồm 4 mức:

- + Mức 0: Giới hạn trong chương trình
- + Mức 1: Bảo trì mức vật lý: liên quan đến phần cứng
- + Mức 2: Mức truy nhập tổ chức
- + Mức 3: Mức quan niệm, khái niệm hay logic

- Các loại bảo trì:

- + Bảo trì sửa chữa: 17% đến 20%
- + Bảo trì thích ứng: 18% đến 25%
- + Bảo trì hoàn thiện: cải tiến hệ thống để nó chạy tốt hơn, ổn định hơn, nhanh hơn... chiếm từ 50% đến 60%.

Tóm lại chu trình phát triển của hệ thống truyền thông như sau:



Câu hỏi ôn tập

1. Trình bày các bước chạy thử và test hệ thống
2. Có thể áp dụng phương pháp luận PTTK hệ thống thông tin cho các bài toán kỹ thuật được không? Có áp dụng cho các dự án xã hội được không?
3. Tại sao nói phân tích thiết kế hệ thống là một công việc cực kỳ quan trọng trong quá trình xây dựng hệ thống tin quản lý. Anh chị hiểu như thế nào câu nói “Sự lãng phí, rui ro trên giấy còn hơn xảy ra trong thực tiễn”
4. Những công cụ diễn tả xử lý. Sự khác nhau giữa biểu đồ phân rã chức năng (BPC) và biểu đồ luồng dữ liệu (BLD). Chúng có mối quan hệ với nhau như thế nào
5. Vòng đời của sản phẩm phần mềm tin học quản lý là gì? Giai đoạn nào quan trọng nhất.
6. Các điều tối kỵ (sai cơ bản để phát hiện) khi vẽ BLD, BPC.
7. Những thành phần cấu thành BLD, thành phần nào sử dụng nhân là động từ? vì sao? Những thành phần nào sử dụng nhân là danh từ? Tại sao? Có hệ thống nào mà BLD không có tác nhân ngoài không? Tại sao? Số tác nhân ngoài tối đa là bao nhiêu?
8. Tại sao cần các thể hiện khác của biểu đồ luồng dữ liệu? Chúng có thể là những cái gì? Các công thức, quy định, thủ tục dùng để làm gì?
9. Mối quan hệ giữa mô hình thực thể liên kết và mô hình CSDL quan hệ. Phân biệt thực thể và kiểu thực thể, liên kết và kiểu liên kết, thuộc tính và giá trị thể hiện của thuộc tính? Cho các thí dụ minh họa.
10. Vai trò của phụ thuộc hàm (PTH) trong phân tích dữ liệu? PTH sơ đẳng, bộ phận, phụ thuộc hàm trực tiếp, bắc cầu. Ý nghĩa của với việc chuẩn hoá dữ liệu.
11. Tại sao phải khảo sát hệ thống hiện trạng trước khi tiến hành phân tích và thiết kế hệ thống mới.
12. Xây dựng Mô hình thực thể liên kết của các hệ thống phổ biến sau:
 - QL Thư viện
 - QL Kết quả học tập
 - QL Khách sạn
 - QL Nhân sự
 - QL Tuyển sinh
 - QL Vật tư
 - QL Kinh doanh
 - QL Xe máy
 - QL Dịch vụ nhà cho thuê
13. Sự khác nhau cơ bản khi thành lập biểu đồ cấu trúc dữ liệu (BCD) theo mô hình thực thể liên kết và mô hình quan hệ.
14. Phân định ranh giới hệ thống phần thực hiện bằng máy tính và thủ công để làm gì. Cách chủ đạo của phương pháp này.
15. Kỹ thuật chính khi thu thập thông tin và các bước thực hiện của nó.
16. Mã hoá dữ liệu: Các phương pháp mã cơ bản, phương pháp nào coi là tốt nhất.. Hãy mã hoá sinh viên bằng số thẻ, xác định mã này
17. Điểm hờ là gì. Tại sao cần nghiên cứu các điểm hờ và các phương pháp bảo mật thông tin. Mật khẩu và mật mã khác nhau thế nào? Quyền ưu tiên là gì trong PTTK.
18. Biểu đồ cấu trúc dữ liệu (BCD) dạng mô hình thực thể liên kết E-R có liên quan đến các bảng dữ liệu trong FOXPRO và ACCESS như thế nào, chúng có thay thế được nhau không?
19. Thiết kế FILE dựa vào những phần gì trước đó? Các bước của một đường truy cập FILE đối với mỗi yêu cầu là gì? Diễn giải các bước đối với các yêu cầu khi truy cập các hệ Thư viện (SACH, ĐOCCIA, MUONTRA), hệ khách sạn (PHONG, KHACH, CHECK_IN_OUT), vv...
20. Module chương trình là gì? Các thuộc tính cơ bản của một module. Đặc trưng trong và đặc trưng ngoài.
21. Phân biệt các biểu đồ sau (rất hay nhầm):
 - a. Biểu đồ BPC
 - b. Giao diện MENU hệ thống
 - c. Sơ đồ tổ chức
 - d. Lược đồ cấu trúc chương trình
22. Chất lượng của một LCT.
23. Tại sao nói rằng thiết kế mẫu thử là nhiệm vụ của người PTTK hệ thống, Các tiêu chuẩn của mẫu thử cần đạt được.
24. Tại sao luồng dữ liệu vào/ ra từ kho dữ liệu đôi khi không có tên?
25. Chức năng sơ cấp là gì? Trong BLD, chức năng sơ cấp đòi hỏi điều gì mà thành phần khác không nhất thiết phải có?
26. Anh chị có nhận xét gì khi học xong môn PTTK hệ thống thông tin (ngắn gọn, gạch đầu dòng)