

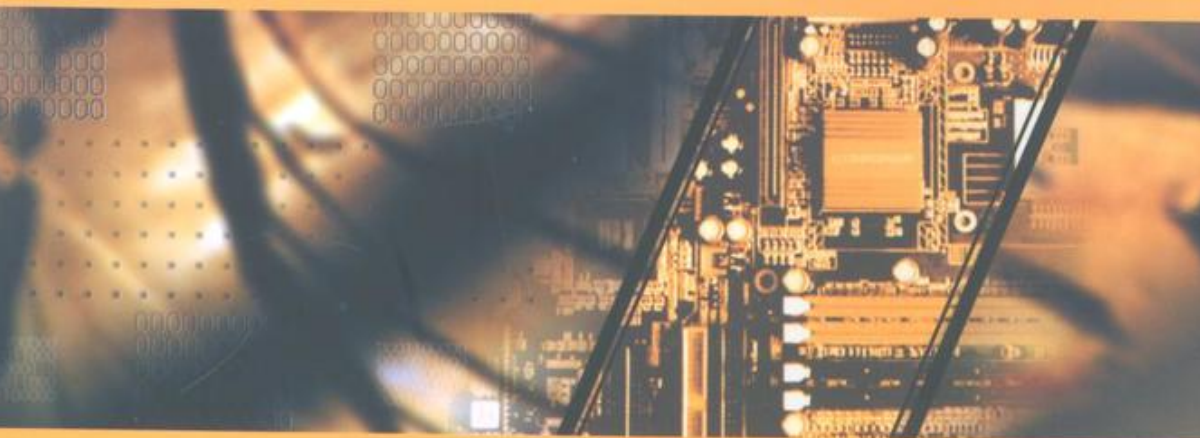


SỞ GIÁO DỤC VÀ ĐÀO TẠO HÀ NỘI

**GIÁO TRÌNH**

# **Phân tích thiết kế hệ thống thông tin quản lý**

DÙNG TRONG CÁC TRƯỜNG TRUNG HỌC CHUYÊN NGHIỆP



NHÀ XUẤT BẢN HÀ NỘI

SỞ GIÁO DỤC VÀ ĐÀO TẠO HÀ NỘI

PHẠM MINH TUẤN

**GIÁO TRÌNH**  
**PHÂN TÍCH THIẾT KẾ HỆ THỐNG**  
**THÔNG TIN QUẢN LÝ**

*(Dùng trong các trường THCN)*

NHÀ XUẤT BẢN HÀ NỘI - 2005

## Lời giới thiệu

---

**N**ước ta đang bước vào thời kỳ công nghiệp hóa, hiện đại hóa nhằm đưa Việt Nam trở thành nước công nghiệp văn minh, hiện đại.

Trong sự nghiệp cách mạng to lớn đó, công tác đào tạo nhân lực luôn giữ vai trò quan trọng. Báo cáo Chính trị của Ban Chấp hành Trung ương Đảng Cộng sản Việt Nam tại Đại hội Đảng toàn quốc lần thứ IX đã chỉ rõ: “Phát triển giáo dục và đào tạo là một trong những động lực quan trọng thúc đẩy sự nghiệp công nghiệp hóa, hiện đại hóa, là điều kiện để phát triển nguồn lực con người - yếu tố cơ bản để phát triển xã hội, tăng trưởng kinh tế nhanh và bền vững”.

Quán triệt chủ trương, Nghị quyết của Đảng và Nhà nước và nhận thức đúng đắn về tầm quan trọng của chương trình, giáo trình đối với việc nâng cao chất lượng đào tạo, theo đề nghị của Sở Giáo dục và Đào tạo Hà Nội, ngày 23/9/2003, Ủy ban nhân dân thành phố Hà Nội đã ra Quyết định số 5620/QĐ-UB cho phép Sở Giáo dục và Đào tạo thực hiện đề án biên soạn chương trình, giáo trình trong các trường Trung học chuyên nghiệp (THCN) Hà Nội. Quyết định này thể hiện sự quan tâm sâu sắc của Thành ủy, UBND thành phố trong việc nâng cao chất lượng đào tạo và phát triển nguồn nhân lực Thủ đô.

Trên cơ sở chương trình khung của Bộ Giáo dục và Đào tạo ban hành và những kinh nghiệm rút ra từ thực tế đào tạo, Sở Giáo dục và Đào tạo đã chỉ đạo các trường THCN tổ chức biên soạn chương trình, giáo trình một cách khoa học, hệ

thống và cập nhật những kiến thức thực tiễn phù hợp với đối tượng học sinh THCS Hà Nội.

Bộ giáo trình này là tài liệu giảng dạy và học tập trong các trường THCS ở Hà Nội, đồng thời là tài liệu tham khảo hữu ích cho các trường có, đào tạo các ngành kỹ thuật - nghiệp vụ và đồng đảo bạn đọc quan tâm đến vấn đề hướng nghiệp, dạy nghề.

Việc tổ chức biên soạn bộ chương trình, giáo trình này là một trong nhiều hoạt động thiết thực của ngành giáo dục và đào tạo Thủ đô để kỷ niệm “50 năm giải phóng Thủ đô”, “50 năm thành lập ngành” và hướng tới kỷ niệm “1000 năm Thăng Long - Hà Nội”.

Sở Giáo dục và Đào tạo Hà Nội chân thành cảm ơn Thành ủy, UBND, các sở, ban, ngành của Thành phố, Vụ Giáo dục chuyên nghiệp Bộ Giáo dục và Đào tạo, các nhà khoa học, các chuyên gia đầu ngành, các giảng viên, các nhà quản lý, các nhà doanh nghiệp đã tạo điều kiện giúp đỡ, đóng góp ý kiến, tham gia Hội đồng phản biện, Hội đồng thẩm định và Hội đồng nghiệm thu các chương trình, giáo trình.

Đây là lần đầu tiên Sở Giáo dục và Đào tạo Hà Nội tổ chức biên soạn chương trình, giáo trình. Dù đã hết sức cố gắng nhưng chắc chắn không tránh khỏi thiếu sót, bất cập. Chúng tôi mong nhận được những ý kiến đóng góp của bạn đọc để từng bước hoàn thiện bộ giáo trình trong các lần tái bản sau.

GIÁM ĐỐC SỞ GIÁO DỤC VÀ ĐÀO TẠO

## **Lời nói đầu**

---

**Y**êu cầu lớn nhất đối với một cán bộ ngành tin học quản lý là phải có khả năng phân tích hiểu được thực trạng nghiệp vụ của các cơ quan để từ đó thiết kế xây dựng ra các hệ thống thông tin dùng máy tính và phương tiện truyền thông đáp ứng yêu cầu quản lý. Tri thức về hệ thống, phân tích và thiết kế hệ thống tin trở thành nhu cầu cấp thiết cho mọi cán bộ tin học.

Đối với học sinh trung cấp tin học, cần được trang bị các kiến thức cơ bản về phân tích, thiết kế một hệ thống thông tin quản lý làm cơ sở cho các em có thể tham gia vào việc thiết kế, xây dựng các hệ thống tin sau này. Đồng thời, nó cũng là các kiến thức cơ sở để các em tiếp tục học tập nâng cao trình độ.

Hiện nay, ở nước ta đã có nhiều sách viết về phân tích thiết kế hệ thống. Tuy nhiên, những cuốn sách này thường được viết chung cho các đối tượng, đặc biệt là thường được dùng cho các sinh viên đại học hoặc cao hơn. Vì vậy chúng tôi biên soạn cuốn sách này với mục đích dành riêng cho các đối tượng là học sinh các trường trung cấp tin học.

Với đối tượng là học sinh trung cấp, cuốn sách này ngoài việc trang bị những kiến thức cơ bản nhất, chúng tôi đưa vào các kỹ thuật cụ thể sử dụng trong việc phân tích, thiết kế hệ thống. Đồng thời chúng tôi cũng cố gắng đưa vào một số ví dụ có phân tích cụ thể để các em dễ hiểu hơn.

Nội dung của cuốn sách được chia làm 4 chương, cuối mỗi chương đều có bài tập lý thuyết cũng như thực hành.

**Chương 1. Một số khái niệm cơ bản về hệ thống**

**Chương 2. Khảo sát hiện trạng và xác lập dự án**

**Chương 3. Phân tích hệ thống**

**Chương 4. Thiết kế hệ thống**

Tôi xin cảm ơn các thầy khoa Công nghệ thông tin - HVKTQS và trường ESTIH cho tôi các ý kiến đóng góp quý báu. Tôi xin cảm ơn trường ESTIH đã tạo điều kiện

cho tôi hoàn thiện giáo trình này.

Mặc dù bản thân đã cố gắng tham khảo các tài liệu và các ý kiến tham gia của các thầy cũng như các bạn đồng nghiệp đã dạy và nghiên cứu môn PTTKHT, song cuốn giáo trình vẫn không tránh khỏi những thiếu sót. Mong bạn đọc và các em học sinh, đóng góp ý kiến.

TÁC GIẢ

# **NHẬP MÔN**

## **CHƯƠNG TRÌNH CHI TIẾT**

### **PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN QUẢN LÝ**

#### **1. Tên chuyên ngành đào tạo: Tin học quản lý**

#### **2. Mục tiêu chung của môn học**

Sau khi học xong môn học này, học sinh nhận biết được các hệ thống thông tin. Có khả năng đọc, hiểu, làm đúng các tài liệu khảo sát, phân tích, thiết kế hệ thống. Biết khảo sát, phân tích, thiết kế các hệ thống thông tin quản lý đơn giản như hệ thống bán hàng, quản lý vật tư,...

#### **3. Điều kiện tiên quyết**

Học sinh phải có hiểu biết sơ lược về quản lý, hoạt động của một số mô hình quản lý.

Học sinh nắm được một hệ quản trị cơ sở dữ liệu.

#### **4. Nội dung tóm tắt**

**Chương 1. Một số khái niệm cơ bản về hệ thống**

**Chương 2. Khảo sát hiện trạng và xác lập dự án**

**Chương 3. Phân tích hệ thống**

**Chương 4. Thiết kế hệ thống**

## **5. Phương pháp dạy học**

Giảng lí thuyết.

Thảo luận theo nhóm.

Lấy ví dụ minh họa.

Chú ý, đưa các ví dụ thực tế.

## **6. Kiểm tra đánh giá**

Cuối mỗi chương có bài kiểm tra 1 tiết.

Kết thúc môn học đánh giá bằng hình thức thi.



# Chương 1

## MỘT SỐ KHÁI NIỆM CƠ BẢN VỀ HỆ THỐNG

### **Mục tiêu**

Làm cho học sinh nhận biết được các yếu tố của một hệ thống: phần tử, mục đích, môi trường.

Phân biệt được các hệ thống quản lí và hệ thống kinh doanh dịch vụ.

Nhận ra tổng quát các lĩnh vực có thể ứng dụng tin học trong một hệ thống.

Kể ra được các giai đoạn phát triển của hệ thống.

### **Nội dung tóm tắt**

Khái niệm về hệ thống.

Các vấn đề của hệ thống.

Các hệ thống kinh doanh, dịch vụ.

## **NỘI DUNG**

### **I. KHÁI NIỆM CHUNG VỀ HỆ THỐNG. HỆ THỐNG KINH DOANH DỊCH VỤ**

#### **1. Khái niệm chung về hệ thống**

Thuật ngữ hệ thống có thể được hiểu 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.

##### **1.1. Các phần tử của hệ thống**

Các phần tử của hệ thống là các thành phần hợp thành hệ thống được hiểu theo nghĩa rộng:

- Các phần tử rất đa dạng, chẳng hạn trong hệ thống mặt trời thì các phần tử là mặt trời, quả đất,... Có khi các phần tử lại là những đối tượng trừu tượng như là một phương pháp, một lập luận, một qui tắc... như trong các hệ thống tư tưởng. Như vậy các phần tử có thể rất khác nhau về bản chất, không những giữa

các hệ thống khác nhau mà có thể ngay trong cùng một hệ thống.

- Các phần tử không nhất thiết là sơ đẳng mà có thể là những thực thể phức tạp, mà có thể lại được xem như những hệ thống. Bởi vậy hệ thống thường có tính phân cấp: Hệ thống được hợp thành từ nhiều hệ thống con, và mỗi hệ thống con lại được hợp thành từ những hệ thống nhỏ hơn...

### **1.2. Quan hệ giữa các phần tử của hệ thống**

Các phần tử của một hệ thống không phải được tập hợp lại một cách ngẫu nhiên, rời rạc, mà giữa chúng luôn tồn tại những quan hệ, những mối ràng buộc lẫn nhau tạo thành một cấu trúc (hay tổ chức). Chẳng hạn, trong một hệ thống hành chính gồm các cán bộ và nhân viên thì giữa họ tồn tại các mối quan hệ, ràng buộc về phân cấp, phân quyền, đoàn thể,... Phần lớn các hệ thống đều có tính biến động song vẫn giữ sự ổn định trong tổ chức, trong các quan hệ giữa các phần tử, nghĩa là vẫn giữ cái bản chất, cái đặc trưng cốt lõi của hệ thống.

### **1.3. Mục đích của hệ thống**

Sự biến động của hệ thống thường thể hiện trên hai mặt:

- *Sự tiến triển*: Các thành phần của hệ thống (các phần tử và các quan hệ) có thể phát sinh, tăng trưởng, suy thoái, mất đi.

- *Sự hoạt động*: Các phần tử của hệ thống trong các mối quan hệ đã định, cùng cộng tác với nhau để thực hiện mục đích chung của hệ thống.

Mục đích của hệ thống thể hiện ở chỗ: Hệ thống biến đổi những cái vào nhất định thành những cái ra nhất định. Chẳng hạn một hệ thống sản xuất thì nhận vào các nguyên vật liệu để sản xuất ra các thành phẩm.

### **1.4. Môi trường của hệ thống**

Một hệ thống luôn tồn tại trong một môi trường nào đó. Hệ thống nhận cái vào từ môi trường và xuất cái ra trả lại môi trường. Để phân biệt hệ thống với môi trường xung quanh cần phải xác định giới hạn của hệ thống (cả về vật lý và khái niệm). Với các loại hệ thống khác nhau cách mô tả hệ thống là rất phong phú và đa dạng. Có thể mô tả hệ thống bằng các phương pháp định tính thông qua mô tả các tính chất, chức năng và nhiệm vụ của hệ thống. Cũng có thể mô tả hệ thống bằng phương pháp định lượng thông qua việc liệt kê danh sách tất cả các phần tử của hệ thống. Việc xác định biên một cách chính xác và hợp lý là rất cần thiết, đặc biệt có ý nghĩa với giai đoạn khảo sát hệ thống. Cần chú ý rằng, giới hạn của hệ thống phụ thuộc chặt chẽ vào mục tiêu của hệ thống đó. Môi trường bên ngoài là tập hợp các phần tử không thuộc vào hệ thống nhưng có mối quan hệ với hệ thống: hoặc là chịu tác động của hệ thống hoặc là tác

động lên hệ thống. Chẳng hạn nếu ta quan niệm công ty là một hệ thống thì khách hàng, nhà cung cấp là các phần tử thuộc môi trường bên ngoài hệ thống. Quan hệ giữa khách hàng và công ty là mua hàng, trong khi đó quan hệ giữa nhà cung cấp và công ty là cung ứng vật tư. Quan hệ giữa môi trường và hệ thống là rất đa dạng và phức tạp. Đôi khi đó là mối quan hệ thông tin (như thông tin về mặt hàng, về giá cả, về giao nhận, về hợp đồng...), nhưng trong một số trường hợp thì mối quan hệ đó lại dường như không phải chỉ là thông tin (như sự tin cậy, uy tín, chất lượng...). Như vậy môi trường chẳng qua lại là một hệ thống nào đó có giao tiếp với hệ thống đang xét.

## **2. Hệ thống kinh doanh dịch vụ và các hệ thống con của nó**

### **2.1. Hệ thống kinh doanh dịch vụ**

Hệ thống kinh doanh dịch vụ là hệ thống mà mục đích là kinh doanh hay dịch vụ.

- Kinh doanh là hoạt động của con người nhằm mang lại lợi nhuận tức là thu giá trị thặng dư.

- Dịch vụ là hoạt động của con người nhằm mang lại lợi ích tức là cung cấp giá trị sử dụng.

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?

*Ví dụ:* Các công ty, nhà máy, dịch vụ.... là các hệ thống kinh doanh vì lợi nhuận. Các trường học, công trình công cộng, bệnh viện.... là các hệ thống dịch vụ vì lợi ích.

Đặc điểm của các hệ thống kinh doanh, dịch vụ là có sự tham gia của con người nên hệ thống mang theo nhiều đặc điểm, ưu điểm, khuyết điểm của con người. Vì vậy các hệ thống kinh doanh dịch vụ có hai nét nổi bật:

- Vai trò của cơ chế điều khiển (trong kinh doanh gọi là sự quản lý) là rất quan trọng nhằm giữ cho hệ thống hướng đúng đích và đạt kết quả với chất lượng cao.

- Vai trò của thông tin cũng rất quan trọng nhằm phục vụ cho nhu cầu giao tiếp, trao đổi của con người.

### **2.2. Các hệ thống con trong hệ thống kinh doanh dịch vụ**

Các hệ thống kinh doanh dịch vụ luôn bao gồm hai hệ thống con:

- Hệ thống tác nghiệp là hệ thống bao gồm con người, phương tiện và các

phương pháp tham gia trực tiếp vào quá trình biến đổi những cái vào thành những cái ra nhằm thực hiện mục tiêu kinh doanh hay dịch vụ. 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.

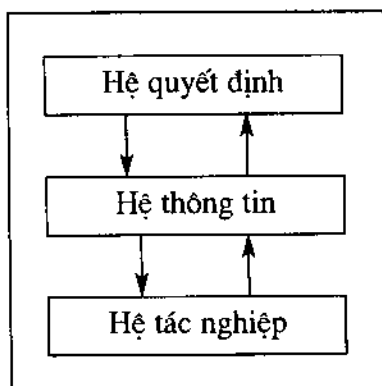
- Hệ thống quản lý là hệ thống bao gồm con người, phương tiện, phương pháp cho phép điều khiển, kiểm soát hoạt động tác nghiệp hướng đúng vào mục đích kinh doanh hay dịch vụ.

Hoạt động quản lý là một dãy nối tiếp của hai việc: đề xuất quyết định kinh doanh và thực thi quyết định kinh doanh. Trước khi quyết định, cần phải thu thập các thông tin liên quan. Các thông tin có ích cho quyết định phải được kết xuất từ nhiều nguồn thông tin phức tạp thông qua các quá trình thu gom, lưu trữ, xử lý. Sau khi ra quyết định, thì quyết định phải được truyền đạt tới nơi thực hiện cùng với những thông tin cần thiết cho việc thực thi quyết định đó. Điều đó có nghĩa là trong quản lý, bên cạnh nhiệm vụ đưa ra các quyết định kinh doanh luôn có nhiệm vụ xử lý thông tin. Vì vậy hệ thống quản lý trong hệ thống kinh doanh dịch vụ lại có thể tách thành hai hệ thống:

+ Hệ quyết định bao gồm con người, phương pháp, phương tiện thực hiện việc đề xuất các quyết định kinh doanh.

+ Hệ thông tin bao gồm con người, phương pháp, phương tiện tham gia vào việc xử lý các thông tin kinh doanh.

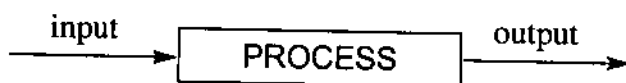
Như vậy ta có thể coi hệ thống kinh doanh dịch vụ chứa ba hệ thống con là: hệ tác nghiệp, hệ quyết định và hệ thông tin. Việc phân chia này chỉ có ý nghĩa phương pháp luận, nhằm cho ta một cách nhìn, một cách nghiên cứu đối với hệ thống, chứ không phải là một sự phân chia về tổ chức.



Hình 1. Các hệ thống con của hệ thống kinh doanh, dịch vụ.

### *Nhiệm vụ và vai trò của hệ thống tin*

Chức năng chính của hệ thống tin là xử lý thông tin của hệ thống. Quá trình xử lý thông tin tương tự như hộp đen gồm bộ xử lý, thông tin đầu vào, thông tin đầu ra.



*Hình 2. Cấu trúc của một hộp đen*

Thông tin trong hệ thống kinh doanh dịch vụ có thể gồm hai loại:

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

### *Nhiệm vụ của hệ thống thông tin*

+ *Đối ngoại*: Hệ thống thông tin thu nhận thông tin từ môi trường ngoài, đưa thông tin ra ngoài. Ví dụ như thông tin về giá cả, thị trường, sức lao động, nhu cầu hàng hoá v.v...

+ *Đối nội*: Hệ thống thông tin là cầu nối liên lạc giữa các bộ phận của hệ thống kinh doanh dịch vụ. Nó cung cấp cho hệ tác nghiệp, hệ quyết định các thông tin gồm hai loại nhằm: phản ánh tình trạng nội bộ của cơ quan, tổ chức trong hệ thống và tình trạng hoạt động 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.

## **2.3. Các thành phần của hệ thống thông tin**

Nếu không kể con người và phương tiện thì HTTT còn lại thực chất gồm hai thành phần chính: 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 thông tin thường phải 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 cô đọng, 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, qui tắc, thủ tục cho phép xử lý các thông tin kinh doanh dịch vụ.

## **II. SỬ DỤNG MÁY TÍNH ĐỂ XỬ LÝ TỰ ĐỘNG CÁC THÔNG TIN**

### **1. Các hệ thống tin học**

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

Sự tham gia của máy tính trong một hệ thống tin học có thể ở nhiều mức độ khác nhau:

- *Mức thấp*: Máy tính chỉ được dùng để giải quyết một số vụ việc đơn lẻ.
- *Mức trung bình*: Máy tính cùng với con người cộng tác, phân công công việc để cùng nhau thực hiện một qui trình quản lý phức tạp.
- *Mức cao*: Máy tính đóng vai trò chủ chốt trong quá trình xử lý thông tin,

con người không can thiệp vào quá trình xử lý mà chỉ có nhiệm vụ cung cấp thông tin đầu vào cho máy tính và nhận các kết quả ra từ máy tính.

## **2. Các phương thức xử lý thông tin của máy tính**

Việc xử lý thông tin bằng máy tính có thể thực hiện theo nhiều phương thức khác nhau:

### **2.1. Xử lý tương tác**

Xử lý tương tác là xử lý thực hiện từng phần, xen kẽ giữa phần thực hiện bởi người và phần thực hiện bởi máy tính: hai bên trao đổi với nhau dưới hình thức đối thoại. Ở đây, con người không những đưa ra yêu cầu xử lý mà còn đưa ra các quyết định dẫn dắt quá trình để đi tới kết quả.

### **2.2. Xử lý theo lô**

Các giao dịch diễn ra, luồng thông tin đến gộp thành nhóm và đợi xử lý theo lô. Ví dụ các giao dịch bán hàng trong một ngày được cập nhật vào cuối mỗi ngày và sau khi các thông tin đó được cập nhật thì hệ thống sẽ thực hiện các thao tác tính tồn kho, tính doanh thu bán ra trong ngày. Ngoài ra các hệ thống xử lý theo lô có thể áp dụng trong các bài toán như tính lương, tuyển sinh và các bài toán giải quyết có tính định kỳ theo chu kỳ thời gian nhất định. Phương thức này thường dùng cho các trường hợp:

- 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.

### **2.3. Xử lý trực tuyến**

Khi giao dịch phát sinh, các thông tin đến được cập nhật tự động và xử lý ngay. Xử lý trực tuyến dùng để hiển thị, chỉnh đốn, sửa chữa các tệp dữ liệu, phục vụ trực tiếp khách hàng tại chỗ. Ví dụ như bán vé máy bay, vé tàu, ...

### **2.4. Xử lý thời gian thực**

Là hành vi của một hệ thống phải thoả mãn một số ràng buộc rất ngặt nghèo về thời gian, chẳng hạn phải chịu hạn định với thời gian trả lời. Phương pháp này phù hợp với các hệ thống điều khiển.

## **3. Một số loại hệ thống tin học**

### **3.1. Hệ thống thông tin quản lý**

Hệ thống tin quản lý là một cấu trúc hợp nhất các cơ sở dữ liệu và dòng

thông tin nhằm làm tối ưu cho việc thu thập, truyền và trình bày thông tin qua tổ chức nhiều cấp, nhiều thành phần, thực hiện nhiều nhiệm vụ để hoàn thành một mục tiêu thống nhất.

*Ví dụ về hệ thống tin quản lý như:* Hệ thống quản lý nhân sự, Hệ thống quản lý sinh viên, Hệ thống quản lý mua bán hàng,...

Hệ thống tin quản lý thường được ứng dụng trong các lĩnh vực hoạt động như thương mại, hành chính, kế toán, tài vụ,...

- Hệ thống tin quản lý có cơ sở dữ liệu hợp nhất nên hệ thống tin quản lý có thể cung cấp cho các nhà quản lý công cụ và khả năng có thể dễ dàng truy cập thông tin.

Hệ thống tin quản lý có các chức năng:

- Thu thập, phân tích và lưu trữ các thông tin một cách hệ thống.
- Xử lý thông tin: Thay đổi, sửa chữa, tiến hành tính toán tạo ra các thông tin kết quả.
- Phân phối, cung cấp thông tin.

Chất lượng của hệ thống tin quản lý được đánh giá qua tính nhanh chóng trong việc đáp ứng các yêu cầu thông tin, tính mềm dẻo của hệ thống và toàn vẹn của thông tin.

### **3.2. Hệ thống tự động hoá sản xuất**

Đó là các hệ thống nhằm xử lý và điều khiển tự động các quá trình vận hành các thiết bị trong sản xuất, viễn thông, quân sự, ... Các hệ thống này đều phải làm việc theo phương thức xử lý thời gian thực.

### **3.3. Phần mềm hệ thống**

Các hệ thống này thiết lập nên hạ tầng kỹ thuật của các hệ thống máy tính phục vụ cho các phần mềm ứng dụng chạy trên đó. Đó là hệ điều hành, chương trình dịch, hệ quản trị cơ sở dữ liệu,...

### **3.4. Các hệ thống tự động hoá văn phòng**

Tự động hoá văn phòng là cách tiếp cận nhằm đưa máy tính vào hoạt động văn phòng, cho phép thu tóm mọi việc tính toán, giao lưu, quản lý thông tin tất cả vào trong các cửa sổ trên màn hình máy tính. Một hệ thống tự động hoá văn phòng thường cung cấp một số trong các chức năng sau:

- Thư tín điện tử.
- Thư tín tiếng nói.
- Lịch biểu, bố trí thời gian.



- Các phương tiện tính toán đơn giản.
- Quản lý tệp.
- Kết nối cửa sổ.
- Xử lý văn bản.

#### **4. Tính mở của hệ thống tin**

Hiện tại, một thực tế là trong một xí nghiệp hoặc cơ quan, hệ thống thông tin là một tập hợp các hệ thống khác nhau trên nền các phần cứng và phần mềm khác nhau. Các thành phần thường là được cung cấp bởi các nơi khác nhau và nói chung là không tương thích. Chúng có thể không tương thích với nhau về giao diện (kiểu chữ, chế độ đồ họa, độ phân giải màn hình hoặc về cấu trúc dữ liệu). Sự không tương thích giữa các hệ thống đã cản trở rất nhiều tới khả năng trao đổi thông tin bên trong tổ chức và sự trao đổi thông tin giữa tổ chức với các hệ thống bên ngoài.

Để có thể thực hiện việc trao đổi dữ liệu hoặc trao đổi các ứng dụng giữa các hệ thống với nhau, các hệ thống này phải được thiết kế theo các chuẩn công nghệ trong lập trình, truyền thông, mạng, quản trị hệ thống, dịch vụ và giao diện.

*Trong tài liệu này bắt đầu từ chương 2 đều nói về hệ thống thông tin quản lý. Bởi vậy từ nay khi nói tới hệ thống thì được hiểu là hệ thống thông tin quản lý.*

### **III. SỰ PHÁT TRIỂN HỆ THỐNG**

#### **1. Các giai đoạn phân tích, thiết kế và cài đặt**

Quá trình phân tích, thiết kế và cài đặt hệ thống có thể chia thành các giai đoạn sau (việc phân giai đoạn này phụ thuộc từng phương pháp và chỉ có tính tương đối.):

*\* Giai đoạn 1: Khảo sát hiện trạng và xác lập dự án.*

Thu thập thông tin, tài liệu, nghiên cứu hiện trạng nhằm làm rõ tình trạng hoạt động của hệ thống tin cũ trong hệ thống thực, từ đó đưa ra giải pháp xây dựng hệ thống tin mới (hệ thống tin quản lý).

*\* Giai đoạn 2: Phân tích hệ thống.*

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

*\* Giai đoạn 3: Thiết kế hệ thống*

Là nhằm đưa ra các quyết định về cài đặt hệ thống, để sao cho hệ thống thoả

mãn được các yêu cầu mà giai đoạn phân tích đã đưa ra, đồng thời lại thích ứng với các điều kiện ràng buộc trong thực tế.

Giai đoạn thiết kế hệ thống được chia thành các giai đoạn con:

- Thiết kế tổng thể (xác định vai trò của máy tính).
- Thiết kế chi tiết (thiết kế giao diện, thiết kế các kiểm soát, thiết kế cơ sở dữ liệu, thiết kế các module, chương trình).

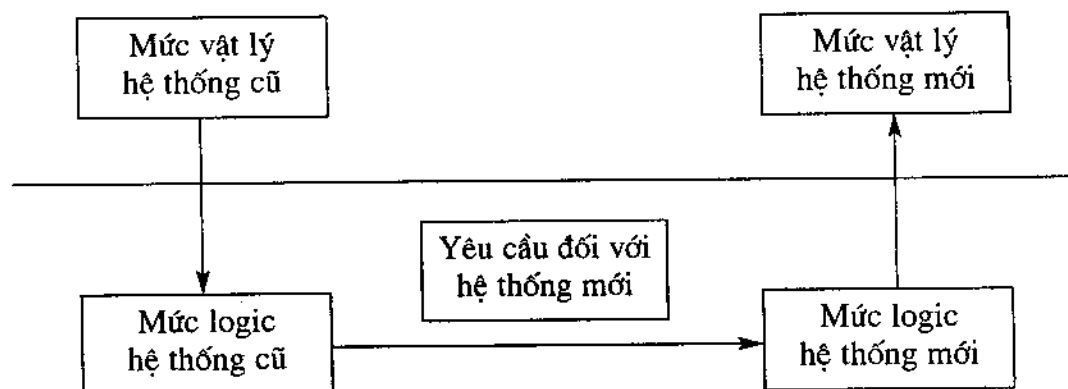
\* *Giai đoạn 4: Cài đặt*

Bao gồm hai công việc chính là lập trình và kiểm định nhằm chuyển các kết quả phân tích và thiết kế trên giấy thành một hệ thống chạy được.

\* *Giai đoạn 5: Khai thác và bảo trì*

Là giai đoạn đưa hệ thống vào sử dụng, đồng thời thực hiện các chỉnh sửa khi phát hiện thấy hệ thống còn có chỗ chưa thích hợp.

Quá trình phân tích và thiết kế hệ thống có thể xem xét qua sơ đồ phân tích thiết kế có cấu trúc gồm bốn bước chính tương ứng với các khối chỉ ra trong sơ đồ:



Hình 3. Các mô hình vật lý và mô hình logic.

## 2. Chu trình phát triển

Có nhiều loại chu trình phát triển khác nhau. Phần này trình bày vắn tắt một số chu trình phát triển.

- Chu trình thác nước.

Chu trình này mô tả sự phát triển hệ thống trên cơ sở phân chia các giai đoạn riêng rẽ với nhau về thời gian. Điểm cốt yếu của chu trình phát triển này là các giai đoạn được phát triển kế tiếp nhau, giai đoạn sau được bắt đầu khi giai đoạn trước đó đã kết thúc, không chồm lên nhau về thời gian. Chu trình phát triển

thác nước không có sự quay lui để chỉnh sửa khi phát hiện sai sót trong một khâu nào đó. Nhưng sự quay lui lại là một nhu cầu rất tự nhiên, vì nhiều khi có vào giai đoạn sau mới phát hiện được những thiếu sót bắt nguồn từ giai đoạn trước.

- Chu trình tăng trưởng.

Ý tưởng chính của phương pháp này là phát triển dần dần, từng bước và chuyển giao từng phần. Công việc nghiệm thu được tiến hành khi tất cả các phần đã được bàn giao. Phương pháp này chỉ thích hợp với những hệ thống có thể chia cắt và chuyển giao theo từng mảnh.

- Chu trình xoắn ốc.

Phương pháp này duy trì quá trình lặp một dãy các tiến trình mà qua mỗi lần thực hiện các tiến trình đó các mẫu được hoàn thiện dần. Có thể chia chu trình xoắn ốc thành các giai đoạn chính sau:

- + Phát hiện các nhu cầu và xác lập các mục tiêu.
- + Đánh giá các phương án.
- + Thiết kế và tạo lập các nguyên mẫu.
- + Thử nghiệm các mẫu.

Các giai đoạn này được thực hiện lặp nhiều lần cho đến khi hệ thống được hoàn thiện.

- Chu trình lắp ráp các thành phần

Thực chất của phương pháp này là lắp ráp các thành phần có sẵn. Công việc chính là xác định các thành phần phù hợp, thu gom chúng và lắp ráp. Có thể chia thành các giai đoạn:

- + Nghiên cứu, hình thành các giải pháp và xác định các thành phần của hệ thống.
- + Đánh giá, lựa chọn các giải pháp và lựa chọn các thành phần của hệ thống
- + Tích hợp các thành phần.

### **3. Mô hình hoá hệ thống**

Các bước phát triển hệ thống như là tìm hiểu nhu cầu, phân tích hệ thống và thiết kế hệ thống tuy có khác nhau về mục tiêu và nhiệm vụ, song đều là những quá trình nhận thức và diễn tả sự phức tạp thông qua các mô hình. Tức là đó đều là những quá trình mô hình hóa.

#### **3.1. Trừu tượng hóa**

Để tìm hiểu thế giới, các khoa học thực nghiệm vận dụng một nguyên lý cơ

bản, đó là sự trừu tượng hóa.

Trừu tượng hóa là một nguyên lý của sự nhận thức, đòi hỏi phải bỏ qua các sắc thái (của một chủ đề) không liên quan tới chủ đề hiện thời để tập trung hoàn toàn vào các sắc thái liên quan tới chủ đề đó.

Nói cách khác, trước một bài toán (hay một chủ đề), tạm quên hay lờ đi các chi tiết có tác dụng rất ít hoặc không có tác dụng gì đối với lời giải bài toán, nhờ đó hình thành được sự diễn tả đơn giản hóa và dễ hiểu cho phép giải quyết được bài toán theo đúng bản chất của nó.

### 3.2. Mô hình

Mô hình là một dạng trừu tượng hóa của một hệ thống thực.

Nói rõ hơn thì mô hình là một hình ảnh (một biểu diễn) của một hệ thống thực, được diễn tả:

- + Ở một mức độ trừu tượng.

- + Theo một quan điểm (hay một góc nhìn).

- + Bằng một hình thức hiểu được (văn bản, phương trình, bảng, đồ thị,...).

Việc dùng mô hình để nhận thức và diễn tả một hệ thống gọi là mô hình hóa.

### 3.3. Hai mức độ mô hình hóa hệ thống

Mọi mô hình đều phản ánh hệ thống theo một mức độ trừu tượng hóa nào đó. Người ta thường phân biệt hai mức độ chính:

- *Mức logic*: Tập trung mô tả bản chất của hệ thống và mục đích hoạt động của hệ thống mà bỏ qua các yếu tố về tổ chức thực hiện, về biện pháp cài đặt. Nói cách khác mô hình logic trả lời câu hỏi “Là gì?”, chẳng hạn chức năng gì?, thông tin gì? mà bỏ qua câu hỏi “Như thế nào?”

- *Mức vật lý*: Trả lời câu hỏi “Như thế nào?”, quan tâm tới các mặt như: phương pháp, phương tiện, biện pháp, công cụ, tác nhân, địa điểm, thời gian, ...

Vì có sự phân biệt hai mức độ mô hình hóa như trên nên mọi quá trình phát triển hệ thống đều phải bao gồm hai giai đoạn là phân tích và thiết kế.

Giai đoạn phân tích hệ thống có mục đích đi sâu vào bản chất và chi tiết của hệ thống. Nó giải đáp câu hỏi “Là gì?” mà bỏ qua câu hỏi “Như thế nào?”. Vậy phân tích quan tâm vấn đề mà không quan tâm giải pháp. Vấn đề ở đây thường là: Chức năng và dữ liệu.

Giai đoạn thiết kế hệ thống lại có mục đích là lựa chọn các giải pháp cài đặt nhằm hiện thực hóa các kết quả phân tích. Do đó thiết kế phải đi sau phân tích

và đi trước lập trình. Nó trả lời câu hỏi “Như thế nào?”. Thiết kế luôn luôn phải tìm sự dung hòa giữa tính hợp lý của các kết quả phân tích với các yêu cầu của thực tiễn như là các ràng buộc, các hạn chế, các ưu tiên,...

## **Bài tập chương 1**

1. Tại sao khi xây dựng một hệ thống thông tin cần phải có phân tích và thiết kế hệ thống?
2. Nêu vai trò của hệ thống thông tin trong hệ thống kinh doanh dịch vụ.
3. Hãy nêu những thành phần chính của hệ thống thông tin quản lý. Nêu ví dụ một hệ thống thông tin quản lý trong thực tế và chỉ ra các thành phần của hệ thống này.
4. Hãy nêu những công việc chính của phân tích và thiết kế một hệ thống thông tin quản lý.
5. Nêu các giai đoạn phân tích, thiết kế và cài đặt một hệ thống thông tin quản lý.

## Chương 2.

# KHẢO SÁT HIỆN TRẠNG VÀ XÁC LẬP DỰ ÁN

### Mục tiêu

Học sinh phải trình bày được các giai đoạn khảo sát hệ thống.

Chỉ ra các nguồn điều tra với một hệ thống quản lí đơn giản.

Phát hiện được các yếu kém của một hệ thống quản lí, kinh doanh dịch vụ.

Xác định được mục tiêu, giới hạn của dự án.

Biết lập kế hoạch triển khai một dự án.

### Nội dung tóm tắt

Các bước khảo sát đánh giá hiện trạng.

Xây dựng dự án.

## NỘI DUNG

### I. KHẢO SÁT VÀ ĐÁNH GIÁ HIỆN TRẠNG

#### 1. Đại cương giai đoạn khảo sát

##### 1.1. Mục đích

Thông thường thì một hệ thống mới được xây dựng là nhằm để thay thế cho một hệ thống cũ đã bộc lộ nhiều điều bất cập. Chính vì vậy mà việc tìm hiểu nhu cầu đối với hệ thống mới thường bắt đầu từ việc khảo sát và đánh giá hệ thống cũ - hiện trạng.

Khảo sát hiện trạng và xác lập dự án là giai đoạn đầu của quá trình phân tích và thiết kế hệ thống. Việc khảo sát thường 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 khảo sát còn có thể coi như "*Nghiên cứu tính khả thi*" hoặc "*Nghiên cứu hiện trạng*".

Mục đích cuối cùng của giai đoạn này là "*kí kết hợp đồng thoả thuận*" để

xây dựng hệ thống thông tin đối với hệ thống kinh doanh của một tổ chức.

## **1.2. Nội dung khảo sát và đánh giá hiện trạng**

Việc khảo sát được thực hiện theo các nội dung sau:

- Tìm hiểu môi trường xã hội, kinh tế và kỹ thuật của hệ thống, nghiên cứu cơ cấu tổ chức của cơ quan chủ quản hệ thống đó.
- Nghiên cứu các chức trách, nhiệm vụ, các trung tâm ra quyết định và điều hành, sự phân cấp các quyền hạn.
- Thu thập và nghiên cứu các hồ sơ, sổ sách, các tệp cùng với các phương thức xử lý các thông tin đó.
- Thu thập và mô tả các qui tắc quản lý tức là các qui định, các công thức do nhà nước hoặc cơ quan đưa ra làm căn cứ cho các quá trình xử lý thông tin.
- Thu thập các chứng từ giao dịch và mô tả các chu trình luân chuyển và xử lý các thông tin và tài liệu giao dịch.
- Thống kê các phương tiện và tài nguyên đã và có thể sử dụng.
- Thu thập các đòi hỏi về thông tin, các ý kiến phê phán về hiện trạng, các dự đoán, nguyện vọng và kế hoạch cho tương lai.
- Đánh giá, phê phán hiện trạng và đề xuất hướng giải quyết.

Qua các nội dung trên có thể thấy mục tiêu của người phân tích và thiết kế cần đạt được trong giai đoạn này là:

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

## **1.3. Các yêu cầu đối với một cuộc điều tra**

Việc khảo sát điều tra phải đạt được các yêu cầu sau:

- Trung thực, khách quan, phản ánh đúng tình hình thực tại.
- Không bỏ sót thông tin.
- Các thông tin thu thập phải được đo đếm (số lượng, tần suất, độ chính xác,...).
- Không trùng lặp, tức là việc điều tra phải được tiến hành theo một trật tự sao cho mỗi người được điều tra không bị nhiều người điều tra hỏi đi hỏi lại về một vấn đề.
- Không gây cảm giác xấu hay phản ứng tiêu cực ở người bị điều tra, phải luôn luôn gợi mở, tế nhị, tuyệt đối không can thiệp vào công việc nội bộ cơ quan

hay làm tăng thêm các mâu thuẫn trong cơ quan.

## **2. Các nguồn điều tra**

Có nhiều nguồn thông tin trong hệ thống có thể được khai thác cho mục đích điều tra. Mỗi nguồn thường cung cấp các loại thông tin khác nhau và đòi hỏi các phương pháp khai thác khác nhau.

### **2.1. Người sử dụng trong hệ thống**

Các người dùng hệ thống (những nhân viên, cán bộ trong cơ quan cũng như những khách hàng, đối tác của cơ quan) là nguồn thông tin cần được điều tra đầu tiên. Từ các người dùng ta có thể tìm hiểu được sự hoạt động của hệ thống hiện tại, xác định được các mục tiêu và yêu cầu của mỗi người dùng. Phương pháp điều tra thường dùng ở đây là phỏng vấn hoặc phiếu điều tra.

### **2.2. Các sổ sách tài liệu**

Các loại sổ sách tài liệu, tệp máy tính,... thường là nguồn thông tin để điều tra về các loại dữ liệu, luồng dữ liệu và giao dịch. Phương pháp khai thác nguồn này là đầu tiên lập một danh sách các tài liệu qua sự tìm hiểu từ các người dùng, rồi sau đó nghiên cứu từng tài liệu để phát hiện các dữ liệu cơ bản và các dữ liệu cấu trúc. Vào lúc này cần phát hiện các dữ liệu trùng lặp hoặc sự thiếu nhất quán trong tên gọi để đảm bảo rằng không có dữ liệu nào xuất hiện dưới hai tên khác nhau.

### **2.3. Các chương trình máy tính**

Các chương trình máy tính có thể được dùng để xác định các chi tiết về các cấu trúc dữ liệu và các quá trình xử lý. Phương pháp tìm hiểu ở đây là đọc kỹ chương trình hoặc tài liệu kèm theo và đôi khi có thể cho chạy chương trình với các dữ liệu kiểm định để thấy nó làm việc thế nào.

### **2.4. Các tài liệu mô tả qui trình, chức trách**

Đó là những tài liệu qui định các qui trình làm việc và chức trách của các cán bộ nhân viên trong cơ quan. Chúng có thể được dùng để người điều tra hiểu thêm chi tiết về các công việc của các người dùng. Các chi tiết như vậy là quan trọng trong bước thiết kế chi tiết sau này. Phương pháp khai thác là đọc tài liệu để thu thập các chi tiết có ích cho bước thiết kế về sau.

### **2.5. Các thông báo**

Các loại thông báo như chứng từ giao dịch, giấy báo,... là nguồn điều tra cho phép tìm hiểu các loại đầu ra cần thiết với các người dùng. Chúng có thể sử dụng làm căn cứ cho các cuộc phỏng vấn người dùng, để từ đó xác định các đòi hỏi mới về đầu ra ở người dùng.



*Chú ý rằng, không một loại nguồn điều tra nào cung cấp đầy đủ các thông tin, mà phải tiến hành điều tra từ tất cả các nguồn trên.*

### **3. Một số phương pháp khảo sát thường dùng**

#### **3.1. Nghiên cứu tài liệu viết**

Đây là một sự quan sát gián tiếp, bởi vì đây cũng là sự khảo sát bằng mắt nhưng không phải ở hiện trường mà trên các tài liệu viết. Thông qua việc nghiên cứu các loại chứng từ giao dịch như hoá đơn, phiếu thanh toán, các loại sổ sách, các tệp máy tính, các tài liệu tổng hợp như kế hoạch, thống kê, biên bản, quyết định,... ta có thể thu thập được nhiều loại thông tin, từ các hoạt động chung của cơ quan, đến các dữ liệu cơ bản, các dữ liệu cấu trúc. Nghiên cứu tài liệu thường kết hợp với phỏng vấn ở mức thấp (mức thao tác, thừa hành) để chi tiết hóa mô hình của hệ thống.

#### **3.2. Phương pháp quan sát**

Đó là phương pháp theo dõi bằng mắt tại hiện trường, nơi làm việc một cách thụ động. Việc quan sát thường đòi hỏi khá nhiều thời gian. Hơn nữa, quan sát tỉ mỉ từng chi tiết nói chung không phải là phương pháp hữu hiệu để thu thập các thông tin cần thiết cho việc phát triển hệ thống máy tính. Một hệ thống mới thường sẽ làm thay đổi cung cách và các chi tiết thao tác, khiến cho các cung cách làm việc cũ không còn mấy ý nghĩa. Một hạn chế lớn nữa là người bị quan sát thường cảm thấy khó chịu, và họ thường thay đổi cách hành động khi bị quan sát (thay đổi theo chiều hướng không tốt). Tuy nhiên kết hợp quan sát với phỏng vấn ngay tại chỗ là một cách làm rất có hiệu quả.

#### **3.3. Phương pháp phỏng vấn**

Đó là cách làm việc tay đôi hoặc theo nhóm, trong đó người điều tra đưa ra các câu hỏi và chất lọc lấy các thông tin từ các câu trả lời. Đây là phương pháp cơ bản cho mọi cuộc điều tra.

Có hai loại câu hỏi:

- *Câu hỏi mở*: Là câu hỏi mà số khả năng trả lời là rất lớn, người hỏi chưa hình dung hết được. Câu hỏi mở là có ích khi người hỏi chưa có ý định rõ ràng, muốn hỏi để thăm dò, để gợi mở vấn đề, và người trả lời phải là người có hiểu biết rộng.

- *Câu hỏi đóng*: Là câu hỏi mà các phương án trả lời có thể dự kiến sẵn, người trả lời chỉ cần khẳng định đó là phương án nào. Câu hỏi đóng là có ích khi người điều tra đã có chủ định và cần biết rõ các chi tiết.

Việc chuẩn bị và sắp xếp các câu hỏi cho hợp lý, phù hợp với chủ định điều tra và khả năng của người trả lời là điều phải cân nhắc kỹ.

### **3.4. Phương pháp sử dụng bảng hỏi, phiếu điều tra**

Đây là một hình thức phỏng vấn gián tiếp. Các câu hỏi được liệt kê trong một mẫu điều tra, và người được điều tra ghi các trả lời của mình vào mẫu đó.

So với phương pháp phỏng vấn thì chỗ chính yếu của phương pháp này là thiếu sự giao tiếp giữa người hỏi và người trả lời. Trong phỏng vấn nhiều khi ngôn ngữ nói không phải là công cụ duy nhất để truyền đạt. Nét mặt, cử chỉ, dáng điệu có thể cho ta biết người được phỏng vấn đang nghĩ gì, những điều mà họ không muốn nói ra bằng lời. Ngược lại phiếu điều tra, bảng hỏi lại có chỗ mạnh là có thể mở rộng diện điều tra và ít tốn kém.

### **4. Qui trình điều tra**

Một qui trình điều tra là một kế hoạch xác định việc khai thác các nguồn điều tra cần được tiến hành theo trật tự nào, theo phương pháp nào và cần thu thập những thông tin nào.

Ta thường phân biệt ba mức điều tra kể từ trên xuống



*Hình 4. Các mức thông tin trong một hệ thống thông tin*

- *Mức 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.

- *Mức điều phối quản lý:* Mức giám sát của những người quản lý trực tiếp.

Họ cung cấp thông tin báo cáo tóm tắt định kì, 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.

- *Mức thao tác, thừa hành*: 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.

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

## **5. Phân loại và biên tập các thông tin điều tra**

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 là các thông tin sơ đẳng, cấu trúc hoá: Các phòng ban, chức vụ, năm sinh v.v

- Các thông tin động thường các thông tin về không gian như các đường di chuyển tài liệu, về thời gian như thời gian xử lý, hạn định chuyển 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.

Sự phân biệt các loại thông tin điều tra như trên cho phép ta sắp xếp lại các thông tin một cách có hệ thống. Như vậy sau khi rà soát, phân loại, biên tập lại các thông tin thu thập được trong quá trình điều tra, ta hình thành được một bản trình bày mô tả các thành phần và sự hoạt động của hệ thống. Bản trình bày này

có thể kèm theo một số đồ thị, bảng biểu minh họa (chẳng hạn lưu đồ hệ thống, bảng quyết định, ...). Bản trình bày dùng để trao đổi trong nhóm những người triển khai hệ thống cũng như để trao đổi với các người dùng và cơ quan chủ quản hệ thống.

## **6. Phát hiện các yếu kém của hiện trạng và các yêu cầu cho tương lai**

Các yếu kém của hiện trạng thường thể hiện ở:

\* 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 tổn phí cần được đánh giá theo một tiêu chuẩn và khía cạnh nào đó như yếu tố thời gian, con người, quá trình.

Các yêu cầu nảy sinh:

- Những nhu cầu về thông tin chưa được đáp ứng.

- Các nguyện vọng của nhân viên.

- Dự kiến, kế hoạch của lãnh đạo.

## **III. XÂY DỰNG DỰ ÁN**

### **1. Xác định mục tiêu, phạm vi 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.

- Phạm vi (Scope).

Trước hết cần có sự thỏa thuận giữa cơ quan chủ quản và những người phát triển hệ thống về phạm vi, ranh giới giữa các vấn đề đặt ra với dự án. Phạm vi có thể bao trùm cả cơ quan hay chỉ dừng chạm một vài bộ phận nhỏ của cơ quan, bao quát công tác quản lý toàn diện hay chỉ giải quyết một vài công tác quản lý riêng biệt nào đó.

Trong một cơ quan dù lớn hay nhỏ thì hệ thống quản lý thường bao gồm nhiều hệ con (như quản lý nhân sự, quản lý tài chính, quản lý vật tư,...), và mỗi

hệ con lại gồm nhiều hệ nhỏ hơn (chẳng hạn quản lý kho, xử lý đơn hàng,...). Thông thường thì ta cần có một nghiên cứu tổng quát về chiến lược tin học hóa công tác quản lý của cơ quan, qua đó xác định các hệ con và các giao diện giữa chúng, rồi sau đó chọn một số hệ con đưa vào phạm vi giải quyết của dự án.

- Hạn chế

+ Về tài chính: Mức độ đầu tư và chi phí dành cho dự án.

+ Về con người: Khả năng quản lý, nắm bắt kỹ thuật mới, khả năng về đào tạo, tác vụ.

+ Về thiết bị: Các khả năng về thiết bị và kỹ thuật có thể đáp ứng.

+ Về môi trường: Các yếu tố ảnh hưởng về môi trường, xã hội.

+ Về thời gian: Các ràng buộc của hệ thống về thời gian hoàn thành.

- Mục tiêu của hệ thống.

Thông thường thì một hệ thống thông tin được xây dựng là nhằm vào các mục tiêu sau:

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

+ Mang lại lợi ích nghiệp vụ: Tăng khả năng xử lý, đáp ứng yêu cầu nghiệp vụ một cách nhanh chóng, thuận lợi, chính xác, an toàn.

+ Mang lại lợi ích kinh tế: Giảm biên chế cán bộ, giảm chi phí hoạt động, tăng thu nhập.

Khi vạch các mục tiêu cho một dự án xây dựng hệ thống thông tin nên chọn những mục cụ thể để sau này có thể kiểm soát được mức độ hoàn thành của dự án.

## **2. Phác họa và nghiên cứu tính khả thi của giải pháp**

Sau khi khảo sát và đánh giá sơ bộ hệ thống cũ, cần đưa ra một giải pháp tuy chưa chi tiết, song phải đủ để thuyết phục được bên sử dụng về khả năng đáp ứng nhu cầu của nó và đủ rõ để chứng tỏ được tính khả thi.

Thường thì ban đầu, ta nên đề xuất nhiều giải pháp, rồi đánh giá tính khả thi của từng giải pháp, để từ đó tiến hành cân nhắc, so sánh và thỏa thuận với bên sử dụng chọn lấy một giải pháp thỏa đáng nhất.

*Các phác họa đề xuất nhằm vào các điều kiện sau:*

- Thỏa mãn các yêu cầu bên A (bên chủ đầu tư) hay không? Thường các yêu cầu này được đưa ra dưới dạng câu hỏi cốt yếu mà nhà phân tích cần trả lời.

- Định hướng giải quyết, thực hiện như thế nào.

- Về thiết bị: Cần đưa ra các chủng loại, tính năng, giá cả thời gian cung cấp vì chúng thường phải dự trữ sớm.

*Xác định các mức độ tự động hoá khác nhau.*

- Tổ chức lại các hoạt động thủ công.

- Tự động hoá một phần, nghĩa là có máy tính trợ 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*

Tính khả thi có thể được cân nhắc trên các mặt sau:

- Khả thi về kỹ thuật: Các yêu cầu về kỹ thuật và công nghệ của giải pháp có thể đáp ứng được không?

- Khả thi về nghiệp vụ: Có đáp ứng các yêu cầu về nghiệp vụ của bên sử dụng không? (cung cấp đúng các thông tin nghiệp vụ cần thiết vào đúng lúc yêu cầu và đến đúng nơi).

- Khả thi về kinh tế: Chi phí cho giải pháp có thể đáp ứng được không? Có thoả đáng so với lợi ích thu lại không?

### **3. Lập dự trù và kế hoạch triển khai dự án.**

Một dự án xây dựng hệ thống thông tin muốn thành công cần thiết phải có kế hoạch thực thi và lập dự trù. Kế hoạch tổng thể có thể chỉ ra một số bước quan trọng:

*a. Giai đoạn hình thành hợp đồng:* Quyết định hệ thống khả thi hay không và thoả thuận các điều khoản sơ bộ dẫn đến một hợp đồng ký kết.

*b. Lập dự trù thiết bị:* Thời gian chuẩn bị mua sắm thiết bị thường diễn ra khá lâu nên nhất thiết cần dự trù về thiết bị sớm. Dự trù được dựa trên một số căn cứ đã có thể ước lượng được:

- Khối lượng dữ liệu cần lưu trữ lâu dài.

- Các dạng làm việc (xử lý theo lô, tương tác, trực tuyến,...)

- Số lượng người dùng tối đa.

- Khối lượng thông tin cần thu thập.

- Khối lượng thông tin cần kết xuất, các tài liệu cần in ra.

Trên các căn cứ này, ta đã có thể xác định được cấu hình của thiết bị bao gồm:

- Mạng hay máy lẻ.

- Các thiết bị ngoại vi, đặc biệt là các thiết bị ngoại vi đặc dụng.

- Các phần mềm (hệ điều hành, hệ quản trị cơ sở dữ liệu, các ngôn ngữ lập trình, ...)

*c. Dự trù kinh phí:* Ngoài kinh phí cho thiết bị và xây dựng địa điểm còn phải có kinh phí cho quá trình triển khai dự án. Kinh phí này phải được thỏa thuận của đôi bên tham gia dự án, căn cứ trên các mặt:

- Khối lượng công việc và số người tham gia thực hiện qua các giai đoạn.
- Mức độ của các đòi hỏi về chất lượng sản phẩm, về thời hạn hoàn thành và về bảo hành sản phẩm.

Kinh phí cho dự án thường được phân bổ cho các giai đoạn triển khai dự án.

*d. Xây dựng triển khai dự án:*

- Tổ chức nhóm làm việc.
- Sự điều hành dự án: Ban điều hành gồm một số lượng hạn chế các thành viên, trong đó có cả nhà tin học và cả người dùng, được tổ chức bên trên nhóm làm việc của dự án và chịu trách nhiệm về việc đưa ra các quyết định có tính định hướng cho dự án.

- Tiến trình của dự án: Cần phải dự kiến tiến trình triển khai dự án.

Ở giai đoạn xác lập dự án này, vấn đề là chọn một phương pháp và một tiến trình triển khai cho thích hợp với hệ thống cần xây dựng, cũng như thích hợp với kinh nghiệm, thói quen và với cả công cụ sẵn có của nhóm thực hiện dự án. Trên cơ sở đó mà xác định một lịch biểu cho dự án.

## **Bài tập chương 2**

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

2. Trình bày các phương pháp khảo sát hiện trạng hệ thống. Có nhất thiết khi khảo sát hiện trạng hệ thống, phân tích viên phải trực tiếp đến tận nơi để khảo sát không? Tại sao?

3. Hãy chọn một tổ chức (chẳng hạn như một thư viện, một hệ thống quản lý sinh viên của một khoa, một hệ thống quản lý vật tư, hệ thống tuyển sinh đại học,...) để khảo sát. Mô tả các nhiệm vụ của hệ thống, các thành phần của nó cùng với nhu cầu xử lý thông tin trong đó.

## Chương 3

# PHÂN TÍCH HỆ THỐNG

### Mục tiêu

Học sinh trình bày được các mô hình và phương tiện diễn tả chức năng.  
Biết đọc và phân tích sơ đồ phân cấp chức năng của một hệ thống kinh doanh đơn giản.  
Biết đọc và phân tích sơ đồ dòng dữ liệu của một hệ thống đơn giản.  
Phân biệt và chuyển đổi được mô hình vật lí và mô hình logic.  
Biết đọc và trình bày được các mô hình và phương tiện diễn tả dữ liệu.  
Biết phân tích mô hình dữ liệu và mô hình thực thể liên kết cho một hệ thống quản lí đơn giản.

### Nội dung tóm tắt

Phân tích sơ đồ chức năng.  
Phân tích sơ đồ dòng dữ liệu.  
Phân tích mô hình thực thể.  
Phân tích mô hình quan hệ.

## NỘI DUNG

### I. CÁC MÔ HÌNH VÀ PHƯƠNG TIỆN DIỄN TẢ CHỨC NĂNG

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 trao đổi trong quá trình phát triển hệ thống. Mỗi mô hình là một khuôn dạng để nhận thức về hệ thống và nó mang tính chủ quan.

Mục tiêu của phân tích mô hình xử lý là đưa ra cách xác định các yêu cầu của người dùng trong quá trình phát triển hệ thống, những yêu cầu này bám sát vào một loạt các sự kiện mà người phân tích thu được qua phỏng vấn, đặt câu hỏi, đọc tài liệu và 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. *Functional Hierarchical Decomposition Diagram (FHD)*:



- Biểu đồ luồng dữ liệu. *Data Flow Diagram (DFD)*.
- Các ký hiệu mở rộng của hãng IBM.
- Sơ đồ thuật toán (*Algorithm*).
- Ngôn ngữ giả trình (*Pseudo Code*).
- Các đặc tả quy tắc quản lý.

## 1. Các mức độ diễn tả chức năng

Chức năng ở đây là các tiến trình thông tin. Việc diễn tả chức năng có thể thực hiện ở những mức độ khác nhau, tùy theo hoàn cảnh và yêu cầu.

### 1.1. Diễn tả vật lý và diễn tả logic

Sự diễn tả chức năng ở mức độ logic chỉ tập trung trả lời câu hỏi “Là gì?”, nghĩa là chỉ diễn tả mục đích, bản chất của quá trình xử lý, mà bỏ qua các yếu tố về thực tại, về cài đặt (thường gọi là các yếu tố vật lý) như phương pháp, phương tiện, tác nhân, địa điểm, thời gian,...

Sự diễn tả chức năng ở mức vật lý đòi hỏi phải nói rõ cả mục đích và cách thực hiện của quá trình xử lý, nói cách khác phải trả lời cả hai câu hỏi: “Là gì?” và “Làm như thế nào?”. Câu hỏi “Làm như thế nào?” thể hiện ở các khía cạnh như: dùng phương pháp gì?, biện pháp gì?, dùng công cụ gì?, ai làm?, ở đâu?, lúc nào?,...

Trong giai đoạn khảo sát sơ bộ một hệ thống phải ghi nhận những gì đang diễn ra trong thực tế. Vậy lúc đó phải dùng sự diễn tả ở mức độ vật lý. Tuy nhiên các yếu tố vật lý thường che khuất bản chất của hệ thống, làm lu mờ các điểm bất hợp lý của hệ thống (làm lầm tưởng là hợp lý). Để nói rõ sự bất hợp lý và chỉ rõ bản chất của hệ thống, ở giai đoạn phân tích hệ thống phải loại bỏ mọi yếu tố vật lý, và diễn tả các chức năng ở mức độ logic. Đối với hệ thống mới thì một sự mô tả logic là cần thiết trước khi tính đến các biện pháp về cài đặt.

### 1.2. Diễn tả đại thể và diễn tả chi tiết

Ở mức độ đại thể thì một chức năng được mô tả dưới dạng “hộp đen”. Nội dung bên trong hộp đen không được chỉ rõ, nhưng các thông tin vào và ra hộp đen thì được chỉ rõ.

Ở mức độ chi tiết thì nội dung của quá trình xử lý phải được chỉ rõ hơn. Thông thường thì chỉ cần chỉ ra các chức năng con và mối liên hệ về thông tin và điều khiển giữa các chức năng đó. Vì các chức năng con vẫn còn phức tạp, nên lại phải diễn tả chúng một cách chi tiết hơn, thông qua các chức năng nhỏ

hơn. Cứ tiếp tục như thế, sẽ có một sự phân cấp trong mô tả. Ở mức cuối cùng thì các chức năng khá đơn giản, lúc đó có thể diễn tả trực tiếp quá trình xử lý của nó, không cần phải chia thành các chức năng nhỏ hơn. Sự mô tả trực tiếp một chức năng được gọi là sự đặc tả. Sự mô tả chi tiết, đại thể, hay đặc tả được sử dụng tùy lúc, tùy nơi trong quá trình phân tích và thiết kế hệ thống.

## 2. Biểu đồ phân cấp chức năng (FHD)

FHD là công cụ khởi đầu để mô tả hệ thống qua chức năng, là một trong những mô hình tương đối đơn giản, dễ hiểu, thân thiện với người sử dụng, kỹ thuật mô hình hoá lại không quá phức tạp, nó rất có ích cho các giai đoạn sau. Một khâu quan trọng trong khi xây dựng biểu đồ phân cấp chức năng là xác định chức năng.

*Chức năng* là khái niệm dùng để mô tả công việc cần thiết sao cho công tác nghiệp vụ được thực hiện. Trong chức năng không cần thiết nêu ra rằng nghiệp vụ đó được thực hiện ở đâu?, như thế nào?, bởi ai?, có nghĩa là nó không quan tâm đến các yếu tố vật lý của vấn đề mà chỉ quan tâm tới khía cạnh hình thức, logic của vấn đề.

Ví dụ: Chức năng *lập thời khoá biểu* dùng để mô tả cho công tác nghiệp vụ của một trường học nào đó có nhiệm vụ *thu thập thông tin về số lớp học, sĩ số của từng lớp, số phòng học, phân công nhiệm vụ giảng dạy của từng giáo viên để từ đó sắp xếp tạo ra một thời khoá biểu chung cho toàn trường*.

Để hiểu rõ hơn các chức năng cần phân rã, hay nói cách khác là mô tả chúng chi tiết hơn nữa. Để làm việc này chúng ta có thể sử dụng biểu đồ phân cấp chức năng.

*Biểu đồ phân cấp chức năng* là sự phân rã có thứ bậc các chức năng của hệ thống trong phạm vi được xem xét. Nó cho phép phân rã dần dần các chức năng từ chức năng mức cao thành chức năng chi tiết nhỏ hơn.

Thành phần của biểu đồ bao gồm:

- Các chức năng:

+ Tên chức năng: Mỗi chức năng phải có một tên duy nhất dưới dạng động từ - bổ ngữ.

Ví dụ: Lấy đơn hàng, mua hàng, lập thời khoá biểu.

Mua hàng

Hình 5. Chức năng mua hàng.

+ Ký hiệu: Chức năng được ký hiệu bằng hình chữ nhật bên trong có tên chức năng.

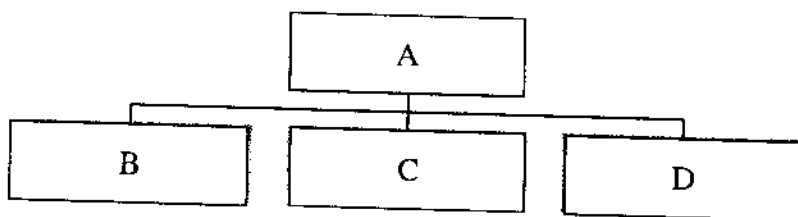
*Ví dụ:*

- Kết nối: Kết nối giữa các chức năng mang tính chất phân cấp và được ký hiệu bằng đoạn thẳng nối chức năng “cha” với các chức năng “con”.

*Ví dụ:* Chức năng A phân rã thành các chức năng B, C, D.

Đặc điểm của FHD

- Các chức năng được nhìn thấy một cách khái quát nhất, trực quan, dễ hiểu, thể hiện tính cấu trúc của phân rã chức năng.



*Hình 6. Sơ đồ phân cấp chức năng.*

- 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.

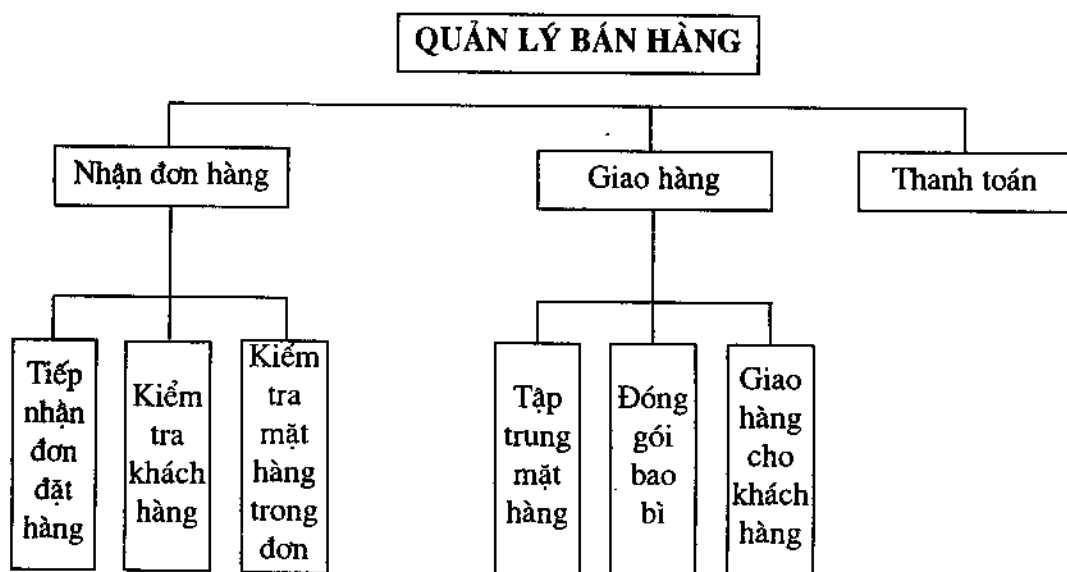
- Rất gần gũi với sơ đồ tổ chức nhưng ta không đồng nhất nó với sơ đồ tổ chức: Phần lớn các tổ chức của doanh nghiệp nói chung thường gắn liền với chức năng.

Vì những đặc điểm trên mà FHD thường được sử dụng làm mô hình chức năng trong bước đầu phân tích.

*Ví dụ:*

Xét hệ thống quản lý bán hàng tại một công ty cung ứng vật tư. Khi có nhu cầu mua hàng, khách hàng gửi đơn đặt hàng cho bộ phận nhận đơn hàng. Bộ phận này xem xét tính hợp lệ của đơn đặt hàng (xem xét về khách hàng, về mặt hàng trong đơn). Nếu đơn hàng được chấp nhận thì sẽ được chuyển sang bộ phận thanh toán và khách hàng trả tiền cho bộ phận thanh toán. Bộ phận giao hàng căn cứ vào đơn hàng đã được chấp nhận gom hàng từ các kho, đóng gói và giao hàng cho khách hàng.

Biểu đồ phân cấp chức năng của hệ thống này được mô tả như sau:



Hình 7. Sơ đồ phân cấp chức năng của hệ thống quản lý bán hàng

### 3. Biểu đồ luồng dữ liệu (DFD)

#### 3.1. Khái niệm

Biểu đồ luồng dữ liệu (DFD) là một loại biểu đồ nhằm mục đích diễn tả một quá trình xử lý thông tin với các yêu cầu sau:

- Sự diễn tả là ở mức logic, nghĩa là nhằm trả lời câu hỏi “Làm gì?” mà bỏ qua câu hỏi “Làm như thế nào?”.
- Chỉ rõ các chức năng (con) phải thực hiện để hoàn tất quá trình xử lý cần mô tả.
- Chỉ rõ các thông tin được chuyển giao giữa các chức năng đó và qua đó phần nào thấy được trình tự thực hiện của chúng.

DFD 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 dữ liệu.

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 ký 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.

### 3.2. Các thành phần trong DFD

DFD gồm có các thành phần sau:

- Tiến trình (Process).
- Luồng dữ liệu (Data Flows).
- Kho dữ liệu (Data Store).
- Tác nhân ngoài (External Entity).
- Tác nhân trong (Internal Entity).

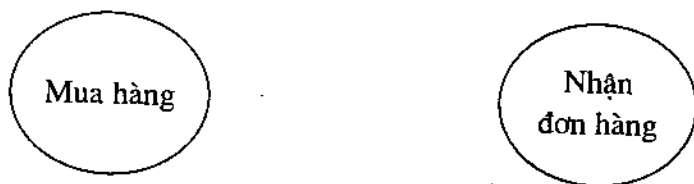
#### *Tiến trình (Process)*

+ Khái niệm: *Tiến trình* được hiểu là các quá trình biến đổi thông tin, từ thông tin vào nó biến đổi, tổ chức lại thông tin, bổ sung thông tin hoặc tạo ra thông tin mới tổ chức thành thông tin đầu ra phục vụ cho hoạt động của hệ thống như lưu vào kho dữ liệu hoặc gửi cho các chức năng khác.

+ Biểu diễn: Tiến trình được biểu diễn bằng hình tròn hay hình ôvan, trong đó có ghi tên của tiến trình.

+ Tên tiến trình: Bởi vì tiến trình là các thao tác nên tên phải được dùng là một động từ cộng với bổ ngữ nếu cần, cho phép hiểu một cách vắn tắt chức năng làm gì.

*Ví dụ:* Chức năng “Mua hàng”, “Nhận đơn hàng”



*Hình 8. Các chức năng*

Cần chú ý rằng tên của tiến trình trong biểu đồ luồng dữ liệu phải trùng với tên đã được đặt trong biểu đồ phân cấp chức năng. Khi xây dựng biểu đồ luồng dữ liệu, nếu có tiến trình nào không tạo ra thông tin mới thì nó không phải là tiến trình trong biểu đồ luồng dữ liệu và khi đó cần phải xem xét lại biểu đồ phân cấp chức năng. Thông thường nên xem xét đến khả năng chức năng này đã bị tách ra khỏi chức năng khác một cách không hợp lý.

#### *Luồng dữ liệu*

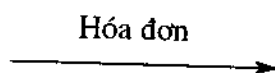
+ Khái niệm: Luồng dữ liệu là luồng thông tin vào hay ra của một tiến trình. Bởi vậy luồng dữ liệu được coi như các giao diện giữa các thành phần của biểu đồ.

+ **Biểu diễn:** *Luồng dữ liệu* trên biểu đồ được biểu diễn bằng mũi tên có hướng trên đó có ghi tên là tên thông tin mang theo. Mũi tên để chỉ hướng của luồng thông tin.

+ **Tên luồng dữ liệu:** Vì thông tin mang trên luồng, nên tên là danh từ cộng với tính từ nếu cần thiết, cho phép hiểu một cách vắn tắt nội dung của dữ liệu được chuyển giao.

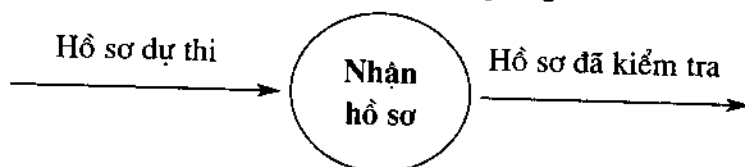
Các luồng dữ liệu và tên được gán cho chúng là các thông tin “logic” chứ không phải là các tài liệu vật lý - giá mang thông tin. Tuy nhiên trong một số trường hợp tên dòng dữ liệu trùng (hoặc ta đã quen dùng) với tên tài liệu vật lý.

Ví dụ: “Hoá đơn”, “Hoá đơn đã kiểm tra”, “Điểm thi”, “Danh sách thi lại”, “Phiếu nhập”, “Hồ sơ dự thi”.



Hình 9. Dòng dữ liệu.

Ví dụ về tiến trình và luồng dữ liệu tương ứng



Hình 10. Tiến trình và luồng dữ liệu.

### *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 mang tin (băng từ, đĩa từ v.v...) 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 cặp đoạn thẳng song song trên đó có ghi tên của kho.

+ **Tên:** Bởi vì kho chứa các dữ liệu nên tên của kho 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.

Ví dụ: Kho “Hồ sơ Cán bộ”, “Độc giả” v.v...

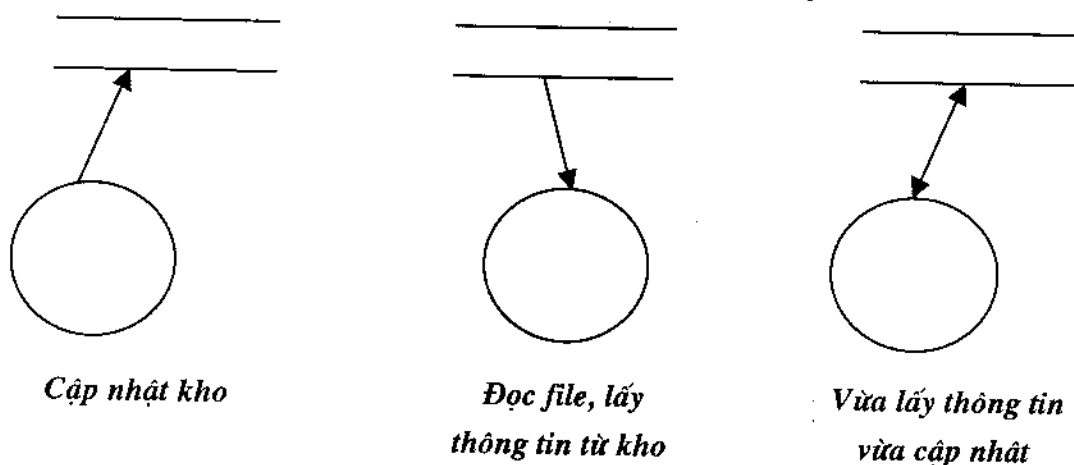
Hồ sơ độc giả

Độc giả

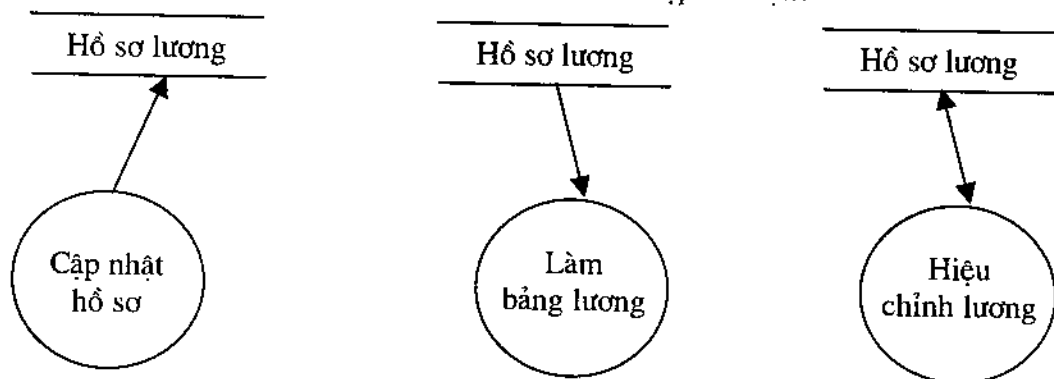
*Hình 11. Các tệp dữ liệu.*

Biểu diễn mối liên quan giữa chức năng xử lý, kho dữ liệu, luồng dữ liệu trong biểu đồ luồng dữ liệu:

Ví dụ việc quản lý lương của cán bộ, công nhân viên tại một cơ quan. Khi có nhân viên mới về cơ quan phải gửi các thông tin về lương của cá nhân cho bộ phận này. Bộ phận này kiểm tra tính đầy đủ và chính xác của các thông tin và lưu trữ các thông tin này. Theo định kỳ làm các bảng lương, đồng thời thường xuyên cập nhật các thông tin mới về lương cho cán bộ công nhân viên.



*Hình 12. Các thao tác trên tệp dữ liệu.*



*Hình 13. Các thao tác vào ra tệp Hồ sơ lương.*

Ta có thể mô tả các công việc đó của bộ phận quản lý lương như sau:

- 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.

- Đối với kho dữ liệu phải có ít nhất một luồng dữ liệu 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”.

- Trong biểu đồ luồng dữ liệu ta có thể đặt một kho dữ liệu ở nhiều nơi để thuận lợi cho việc theo dõi (nhưng phải hiểu là một).

#### *Tác nhân ngoài*

Tác nhân ngoài còn được gọi là đối tác, là một người, một nhóm người hay một tổ chức ở bên ngoài lĩnh vực nghiên cứu của hệ thống nhưng có tiếp xúc, trao đổi thông tin với hệ thống. Sự có mặt của 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 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 tên.

+ Tên: Được xác định bằng danh từ kèm theo tính từ nếu cần thiết.

Ví dụ:

THÍ SINH

KHÁCH HÀNG

*Hình 14. Tác nhân ngoài*

- Trong một biểu đồ luồng dữ liệu ta có thể đặt một tác nhân ngoài ở nhiều chỗ để dễ đọc, dễ hiểu (nhưng phải hiểu chỉ là một).

#### *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, nhưng có trao đổi thông tin với



các phần tử thuộc trang hiện tại 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 ký 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 trong có ghi tên.

+ **Tên tác nhân trong:** Được biểu diễn bằng động từ kèm bổ ngữ nếu cần

Khi xây dựng biểu đồ một tác nhân trong có thể được đặt ở nhiều nơi trong biểu đồ cho dễ đọc, dễ hiểu.

QUẢN LÝ KHO

TÍNH LƯƠNG

*Hình 15. Tác nhân trong.*

## **4. Các phương tiện đặc tả chức năng**

### **4.1. Khái niệm về đặc tả**

Một điểm chung trong việc sử dụng FHD và DFD là để diễn tả một chức năng phức tạp ta phân rã nó ra thành nhiều chức năng con đơn giản hơn. Nói cách khác, là từ một “hộp đen”, ta giải thích nó bằng cách tách nó ra thành nhiều “hộp đen”. Có vẻ như đó là một sự luẩn quẩn, song thực ra là đã có sự tiến bộ vì các chức năng con thu được là đơn giản hơn trước. Muốn đẩy tới sự tiến bộ đó, ta tiếp tục phân rã các chức năng con này. Sự lặp lại quá trình phân rã (thông qua các FHD hay DFD) đương nhiên tới một lúc phải dừng lại. Các chức năng thu được ở mức cuối cùng, đã là rất đơn giản, cũng vẫn cần được giải thích (nếu không thì vẫn cứ là “hộp đen”). Bấy giờ sự giải thích chức năng phải được thực hiện bởi những phương tiện diễn tả trực tiếp. Đó gọi là sự đặc tả chức năng, thường gọi tắt là P-Spec (Process Specification).

Một đặc tả chức năng thường được trình bày một cách ngắn gọn, không vượt quá một trang A4, và gồm hai phần:

Phần đầu đề gồm:

- Tên chức năng.
- Các dữ liệu vào.
- Các dữ liệu ra.

Phần thân mô tả nội dung xử lý, ở đó thường sử dụng các phương tiện mô tả

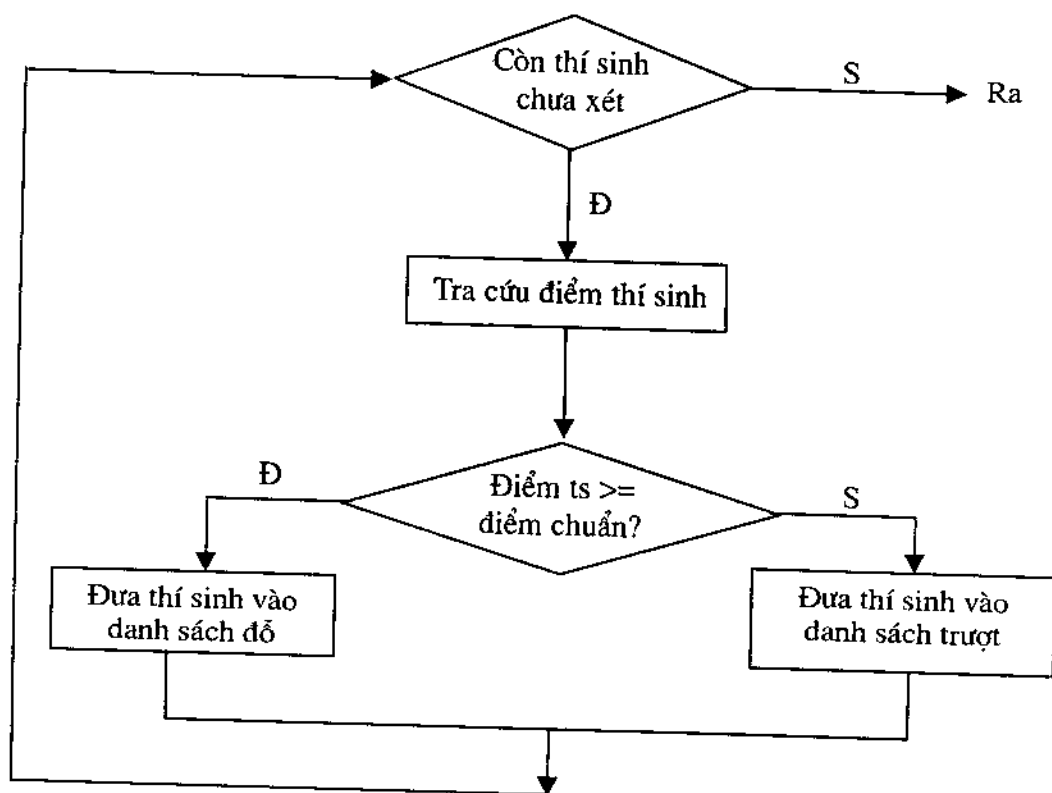
sau đây (liệt kê theo trật tự ưu tiên giảm dần):

- Các phương trình toán học.
- Các sơ đồ khối.
- Các ngôn ngữ tự nhiên cấu trúc hoá.

#### 4.2. Sơ đồ khối

Sơ đồ khối là loại biểu đồ diễn tả giải thuật quen thuộc và ưa dùng với những người mới học lập trình, vì nó đơn giản dễ hiểu. Với lập trình nâng cao, thì nó bộc lộ nhiều nhược điểm, cho nên nó lại ít được ưa dùng: nó khuyến khích việc sử dụng tràn lan GO TO. Nó không thể hiện rõ ba cấu trúc điều khiển cơ bản (tuần tự, chọn, lặp), nó hỗ trợ kém cho lập trình trên xuống và càng tỏ ra gượng ép với lập trình đệ quy v.v... Tuy nhiên với nhiệm vụ đặc tả các chức năng đơn giản mà ta cần ở đây, thì nó đáp ứng được yêu cầu.

Dưới đây là một ví dụ dùng sơ đồ khối để đặc tả chức năng “lập danh sách trúng tuyển và danh sách trượt”.



Hình 18. Một sơ đồ khối

### 4.3. Các ngôn ngữ có cấu trúc

Ngôn ngữ có cấu trúc là một ngôn ngữ tự nhiên bị hạn chế:

- Chỉ được phép dùng các câu đơn sai khiến hay khẳng định (thể hiện các lệnh hay các điều kiện).

- Các câu đơn này được ghép nối nhờ một từ khoá thể hiện các cấu trúc điều khiển chọn và lặp.

Như vậy ngôn ngữ có cấu trúc có những đặc điểm của một ngôn ngữ lập trình, song nó không chịu những hạn chế và quy định ngặt nghèo của các ngôn ngữ lập trình, cho nên được dùng thoải mái hơn. Tuy nhiên nó cũng không quá phóng túng như một ngôn ngữ tự do.

Dưới đây là đặc tả của chức năng “lập danh sách trúng tuyển và danh sách trượt” ở dạng ngôn ngữ có cấu trúc.

Lặp: Lấy một thí sinh từ kho các thí sinh

    Tra cứu điểm của thí sinh đó.

    Nếu Điểm của thí sinh  $\geq$  điểm chuẩn.

    Thì Đưa thí sinh vào danh sách đỗ

*Không thì* Đưa thí sinh vào danh sách trượt

Đến khi Hết thí sinh.

## II. PHƯƠNG PHÁP PHÂN TÍCH HỆ THỐNG VỀ CHỨC NĂNG

Phương pháp phân tích có cấu trúc đặc biệt có tác dụng với hệ thống thông tin quản lý. Đặc điểm của phương pháp này là đơn giản, dễ theo dõi, nhưng không quá sơ lược, mà cũng không cầu kỳ. Công cụ biểu diễn chính được sử dụng trong quá trình phân tích là biểu đồ luồng dữ liệu (DFD).

Mục đích của nó là tiến hành phân tích chức năng của hệ thống để thành lập một mô hình logic của một hệ thống mới, dưới dạng một DFD (hay đúng hơn là một tập hợp các DFD). Nó vận dụng ba kỹ thuật:

- Kỹ thuật phân mức (phân tích từ trên xuống);
- Kỹ thuật chuyển đổi DFD vật lý thành DFD logic;
- Kỹ thuật chuyển đổi DFD hệ thống cũ thành DFD hệ thống mới.

### 1. Kỹ thuật phân mức

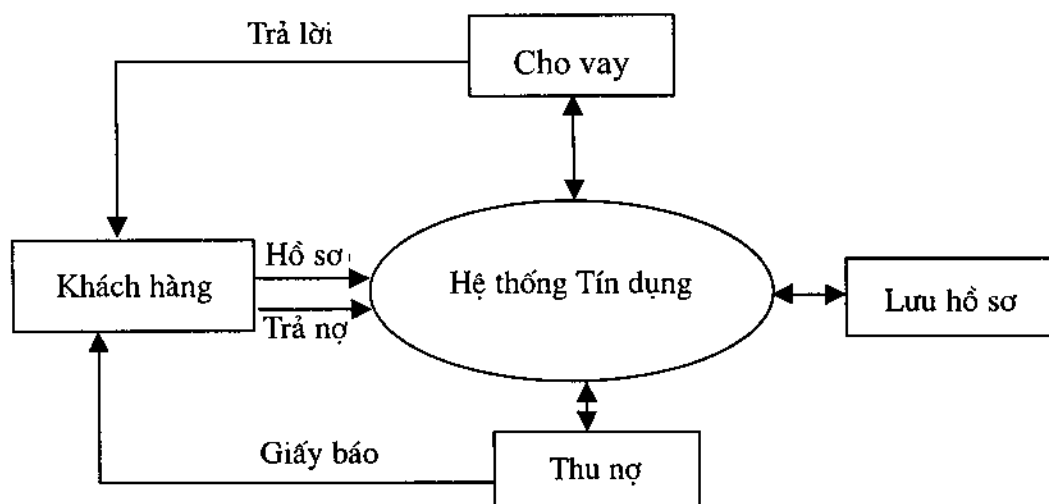
Kỹ thuật phân mức hay còn gọi là “phân tích từ trên xuống” (top - down-

nanalysis) tiến hành sự phân tích chức năng của hệ thống bằng cách đi dần từ một mô tả cụ thể đến những mô tả chi tiết thông qua nhiều mức. Sự chuyển dịch từ một mức tới mức tiếp theo thực chất là sự phân rã chức năng ở mức trên thành một số các chức năng ở mức dưới, vậy đây là quá trình triển khai theo một cây, và chính vì vậy mà phương pháp này còn có tên là phương pháp phân tích có cấu trúc.

Có hai cách vận dụng kỹ thuật phân mức: dùng biểu đồ phân cấp chức năng (FHD) và dùng biểu đồ luồng dữ liệu (DFD).

Với FHD, thì phân tích trên xuống thực hiện bằng cách triển khai dần cây phân cấp từ gốc đến ngọn lần lượt qua các mức (mỗi mức bao gồm một tập hợp các chức năng). Để triển khai từ một mức đến mức tiếp theo ta xem xét từng chức năng và đặt câu hỏi: để hoàn thành chức năng đó thì phải hoàn thành các chức năng con nào. Nhờ đó ta phát hiện các chức năng thuộc mức tiếp theo mà mỗi liên quan với các chức năng ở mức trên là quan hệ bao hàm (hay cha-con). Tên của mỗi chức năng (ở mỗi nút của cây) được chọn lựa để có thể phản ánh một cách ngắn gọn nội dung của chức năng đó. Chức năng ở gốc (mức 0) thể hiện nhiệm vụ tổng quát của hệ thống.

*Ví dụ:* Xét hệ thống hoạt động tín dụng của ngân hàng. Đây là một trong lĩnh vực hoạt động của ngân hàng bên cạnh các hệ thống tiết kiệm, chuyển khoản.



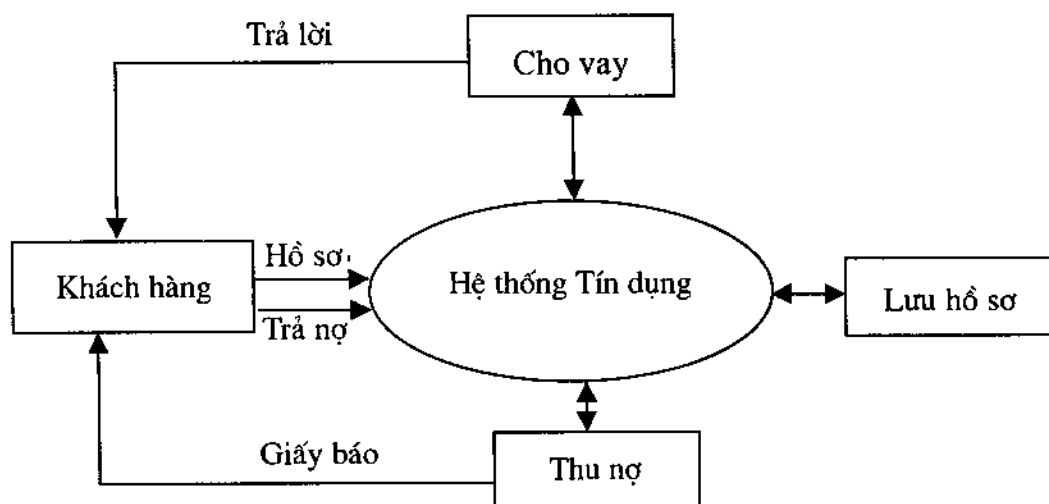
Hình 19. Sơ đồ luân chuyển thông tin của hệ thống

nanalysis) tiến hành sự phân tích chức năng của hệ thống bằng cách đi dần từ một mô tả cụ thể đến những mô tả chi tiết thông qua nhiều mức. Sự chuyển dịch từ một mức tới mức tiếp theo thực chất là sự phân rã chức năng ở mức trên thành một số các chức năng ở mức dưới, vậy đây là quá trình triển khai theo một cây, và chính vì vậy mà phương pháp này còn có tên là phương pháp phân tích có cấu trúc.

Có hai cách vận dụng kỹ thuật phân mức: dùng biểu đồ phân cấp chức năng (FHD) và dùng biểu đồ luồng dữ liệu (DFD).

Với FHD, thì phân tích trên xuống thực hiện bằng cách triển khai dần cây phân cấp từ gốc đến ngọn lần lượt qua các mức (mỗi mức bao gồm một tập hợp các chức năng). Để triển khai từ một mức đến mức tiếp theo ta xem xét từng chức năng và đặt câu hỏi: để hoàn thành chức năng đó thì phải hoàn thành các chức năng con nào. Nhờ đó ta phát hiện các chức năng thuộc mức tiếp theo mà mỗi liên quan với các chức năng ở mức trên là quan hệ bao hàm (hay cha-con). Tên của mỗi chức năng (ở mỗi nút của cây) được chọn lựa để có thể phản ánh một cách ngắn gọn nội dung của chức năng đó. Chức năng ở gốc (mức 0) thể hiện nhiệm vụ tổng quát của hệ thống.

*Ví dụ:* Xét hệ thống hoạt động tín dụng của ngân hàng. Đây là một trong lĩnh vực hoạt động của ngân hàng bên cạnh các hệ thống tiết kiệm, chuyển khoản.



Hình 19. Sơ đồ luân chuyển thông tin của hệ thống

Đối với khách hàng đến vay tiền ở ngân hàng thì phải có một hồ sơ (gồm đơn xin vay, 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 hàng 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 với 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.

### *Xây dựng biểu đồ phân cấp chức năng của hệ thống*

Hệ thống bao gồm có các mức sau:

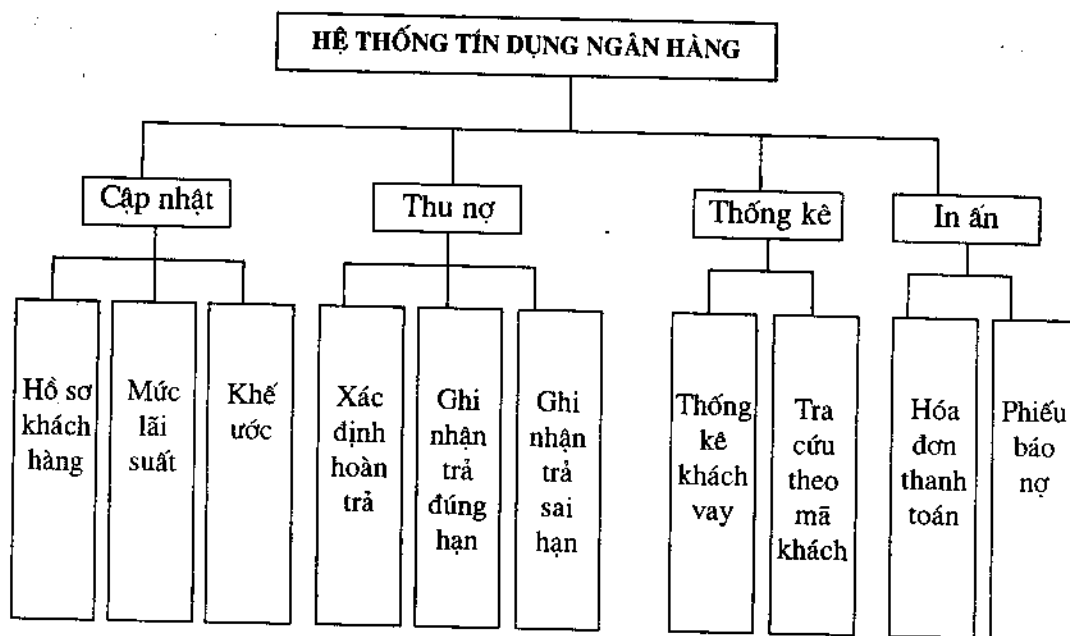
Mức 0: Hoạt động của hệ thống tín dụng.

Mức 1: Gồm các chức năng sau:

- Cập nhật dữ liệu.
- Thu nợ.
- Thống kê và tra cứu.
- In ấn.

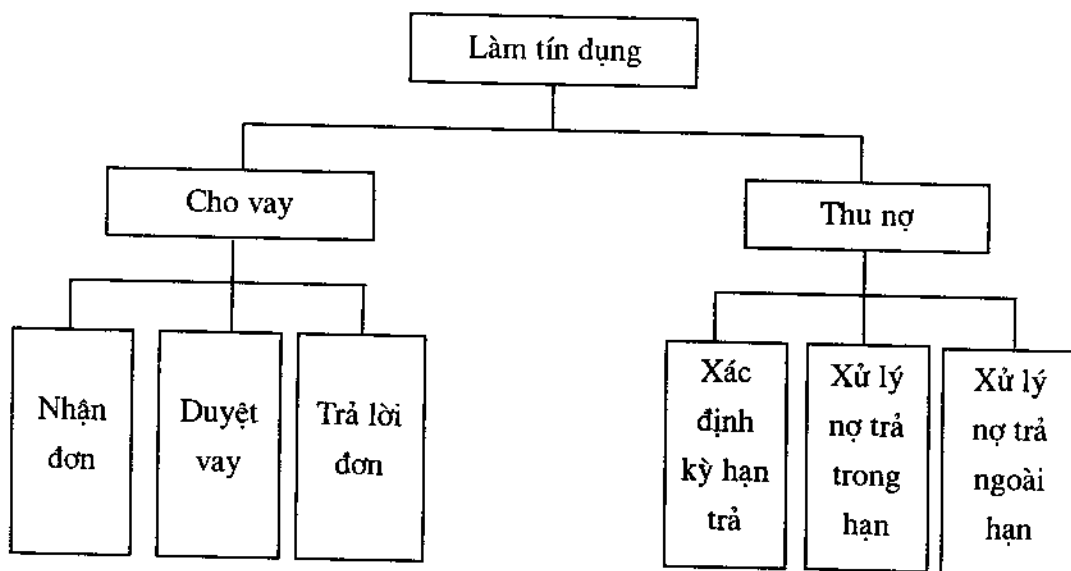
Mức 2: Gồm các chức năng cập nhật khách hàng, xác định mức lãi suất, lập khế ước là các chức năng con của chức năng Cập nhật dữ liệu; Xác nhận hoàn trả, ghi nhận trả đúng hạn, ghi nhận trả sai hạn là các chức năng con của chức năng Thu nợ; thống kê khách vay, thống kê nợ quá hạn, tra cứu theo mã khách là các chức năng con của chức năng tra cứu theo mã khách; In hóa đơn thanh toán, in phiếu báo nợ là các chức năng con của chức năng in ấn

Qua phân tích các chức năng ở trên ta xây dựng được biểu đồ phân cấp chức năng của hệ thống như sau:



Hình 20. Một sơ đồ phân cấp chức năng hệ thống tín dụng

Tuy nhiên qua một cách phân tích khác chúng ta có thể nhận được FHD khác như dưới đây. Điều đó không có mâu thuẫn gì do quan niệm của yêu cầu hệ thống và cách phân tích.



Hình 21. Một sơ đồ phân cấp chức năng của hệ thống tín dụng khác

Để thấy quá trình phân tích với FHD, mà thực chất là sự phân rã dần dần các chức năng là tự nhiên và dễ làm. Tuy nhiên kết quả thu được có thể nói là nghèo nàn: Hệ thống chỉ được diễn tả bằng một tập hợp các chức năng rời rạc (ở một mức của FHD). Vì vậy cách làm này chỉ thích hợp cho sự phân tích bước đầu, hoặc cho các hệ thống đơn giản.

Với DFD thì quá trình phân tích trên xuống lại là quá trình thành lập dần dần các DFD diễn tả các chức năng của hệ thống theo từng mức. Mỗi mức là một tập hợp các DFD:

- Mức bối cảnh hay khung cảnh (còn gọi là mức 0) chỉ có một DFD, trong đó chỉ có một chức năng duy nhất - chức năng tổng quát của hệ thống và các luồng thông tin trao đổi với các tác nhân ngoài.

- Mức đỉnh (còn gọi là mức 1) cũng chỉ có một DFD gồm các chức năng chính của hệ thống.

- Các mức dưới đỉnh (còn gọi là các mức 2,3,4,...) , mỗi mức gồm nhiều DFD được thành lập như sau:

Cứ mỗi chức năng mức trên, ta thành lập một DFD ở mức dưới, gọi là DFD định nghĩa (hay giải thích), chức năng đó, theo cách sau:

- Phân rã chức năng đó thành nhiều chức năng con.

- Vẽ tại các luồng dữ liệu vào và ra chức năng trên nhưng nay phải vào hay ra ở các chức năng con thích hợp;

- Nghiên cứu các quan hệ về dữ liệu giữa các chức năng con, nhờ đó bổ sung các luồng dữ liệu nội bộ hoặc các kho dữ liệu nội bộ.

### **Cách đánh số các chức năng**

Các chức năng được đánh số theo ký pháp chấm theo dõi vệt triển khai từ trên xuống.

- Các chức năng ở mức đỉnh được đánh số 1.2.3,....

- Các chức năng ở các mức dưới đỉnh, chẳng hạn các chức năng mức 2 được đánh số là: 1.1, 1.2,....,2.1, 2.2,.... Các chức năng mức 3 được đánh số là: 1.1.1, 1.1.2,....,2.1.1, 2.1.2,....

Khi triển khai dần các mức như trên ta phải tuân thủ một số quy tắc sau đây, gọi là các quy tắc về sự tương hợp giữa các mức:

- Khi định nghĩa một chức năng ở mức trên thành một DFD ở mức dưới thì các luồng dữ liệu vào và ra chức năng đó phải được bảo toàn, nghĩa là phải được vẽ lại đầy đủ trong DFD định nghĩa. Khi bảo toàn từ mức trên xuống mức dưới như vậy thì một luồng dữ liệu có thể phân rã ra thành nhiều luồng con, nếu cần.

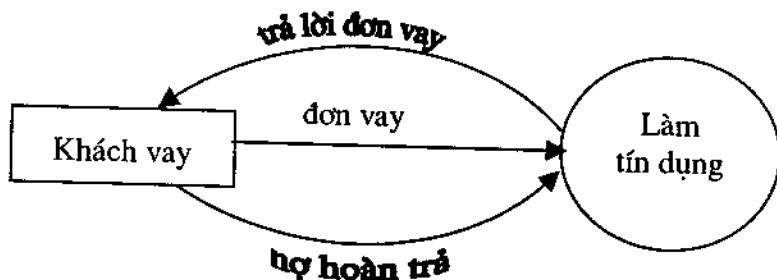


- Các tác nhân ngoài phải xuất hiện toàn bộ trong DFD bối cảnh, và không được phát sinh mới ở các mức dưới. Tuy nhiên có thể vẽ lại một tác nhân ngoài ở mức dưới nếu thấy cần (chẳng hạn để thấy nơi xuất phát hoặc nơi đến của một luồng dữ liệu nào đó).

- Các kho dữ liệu không xuất hiện trong DFD bối cảnh. Tuy nhiên chúng dần dần phát sinh ở các mức dưới khi cần đến.

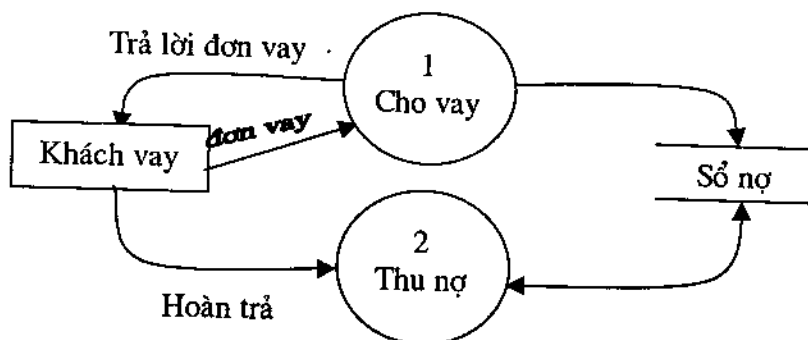
Trở lại ví dụ về hệ thống thông tin ở cơ sở tín dụng thì quá trình phân tích trên xuống là như sau:

- Mức khung cảnh (mức 0): Chức năng tổng quát của hệ thống là: Làm tín dụng. Tác nhân ngoài của hệ thống là khách vay (cũng có thể còn có những đối tác khác như là ngân hàng, chính quyền địa phương v.v..., song tạm thời lược bỏ cho đơn giản). Bổ sung các luồng dữ liệu trao đổi giữa hệ thống và tác nhân ngoài, ta có biểu đồ luồng dữ liệu khung cảnh như sau:



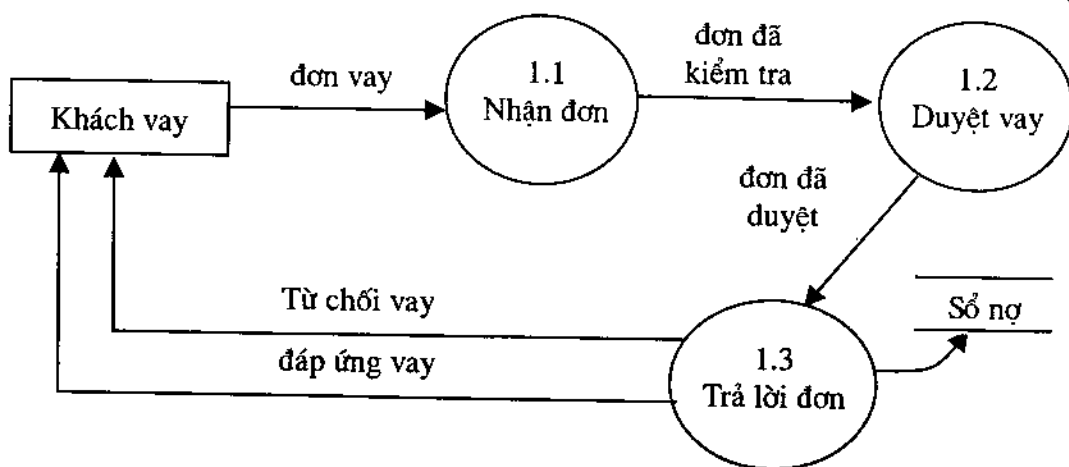
Hình 22. Sơ đồ luồng dữ liệu mức ngữ cảnh

- Mức khung cảnh (mức 1): Chức năng làm tín dụng có thể phân rã thành hai chức năng con là Cho vay và thu nợ. Ngoài ba luồng dữ liệu vào / ra ở mức khung cảnh được bảo toàn, thì ta thấy luồng thông tin trao đổi giữa hai chức năng cho vay và thu nợ là không trực tiếp, mà phải thông qua một kho dữ liệu, là sổ nợ. Từ đó có DFD mức đỉnh như sau:



Hình 24. Sơ đồ dòng dữ liệu mức đỉnh

- Mức dưới đỉnh (mức 2): Chức năng 1 được phân rã thành ba chức năng: Nhận đơn, duyệt vay và trả lời đơn. Còn chức năng 2 thì được phân rã thành 3 chức năng: Xác định kỳ hạn trả, xử lý nợ trả trong hạn và xử lý trả nợ ngoài hạn. Bảo toàn các luồng dữ liệu vào / ra và thêm các luồng dữ liệu nội bộ, ta lập được hai DFD định nghĩa hai chức năng 1 và 2 như sau:

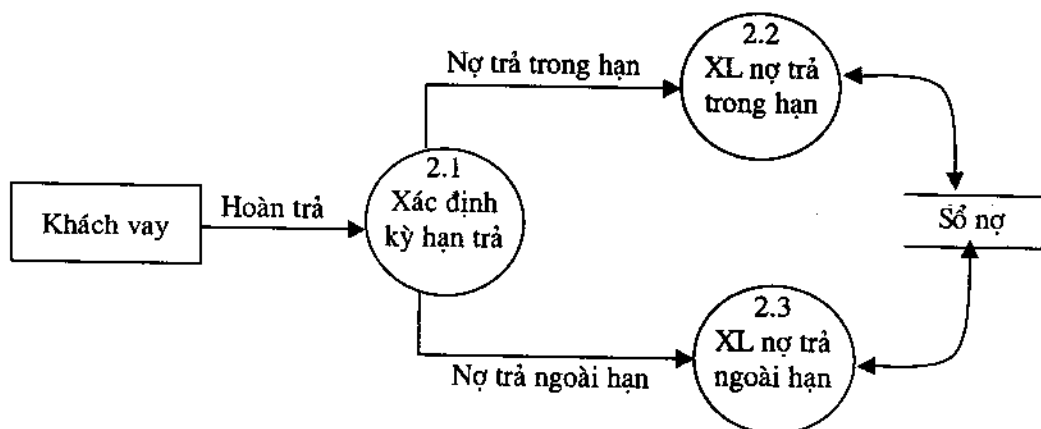


Hình 25. Sơ đồ luồng dữ liệu chức năng 1

Chú thích:

Nếu cùng một hệ thống, mà ta tiến hành phân tích bằng cả hai cách: dùng FHD và dùng DFD thì đương nhiên giữa hai mô hình phải có mối liên quan ăn khớp: Luôn luôn phải có ánh xạ 1-1 giữa các chức năng trong FHD với các chức năng trong các DFD tương ứng.

Quá trình triển khai trên xuống không thể kéo dài mãi, mà phải dừng sau một số mức. Ta quyết định dừng quá trình, khi có những biểu hiện sau:



Hình 26. Sơ đồ luồng dữ liệu chức năng 2

Các chức năng đã là khá đơn giản.

Với các hệ thống vừa và nhỏ số mức thường là 3, với các hệ thống lớn số mức cũng chỉ khoảng 6.

Khi dùng việc triển khai bằng DFD, thì mỗi chức năng trong các DFD ở mức cuối cùng, ta phải cho một đặc tả trực tiếp.

Quá trình triển khai đã làm phát sinh rất nhiều tên gọi: Tên chức năng, tên luồng dữ liệu, tên kho dữ liệu. Các tên đó rất dễ gây lầm lẫn cho các người dùng cũng như các người thiết kế sau này. Vậy phải lập một từ điển dữ liệu để giải thích các tên gọi đó. Nội dung và cấu trúc của từ điển dữ liệu sẽ được đề cập sau.

## **2. Kỹ thuật chuyển đổi DFD vật lý thành DFD logic**

Mục đích của phân tích là đi tới một mô hình logic của hệ thống. Quá trình phân mức nói ở mục trên chỉ có thể dẫn ta vào những mô tả chi tiết, chứ không phải là đã đưa đến những mô tả logic. Bởi vì trong mô tả mà thu được dưới dạng DFD thì vẫn còn lẫn các yếu tố vật lý. Nhất là khi sự mô tả thành lập dựa trên sự khảo sát một hệ thống đang tồn tại, đang hoạt động, các yếu tố vật lý vốn tồn tại trong thực tế, sẽ có nhiều cơ hội lọt qua sự sàng lọc của người phân tích để đi vào các DFD.

Có ba loại yếu tố vật lý có thể lẫn vào các DFD:

(1) Các yếu tố vật lý xuất hiện tường minh trong ngôn từ hay hình vẽ ở trong biểu đồ như là:

- Các phương tiện, phương thức được dùng để thực hiện các chức năng (như máy tính, bàn phím, máy in, xử lý thủ công v.v...)

- Các giá mang thông tin (như đĩa từ, sổ sách, chứng từ trên giấy, đường điện thoại v.v...)

- Các tác nhân thực hiện chức năng (như giám đốc, kế toán viên, thủ kho v.v...)

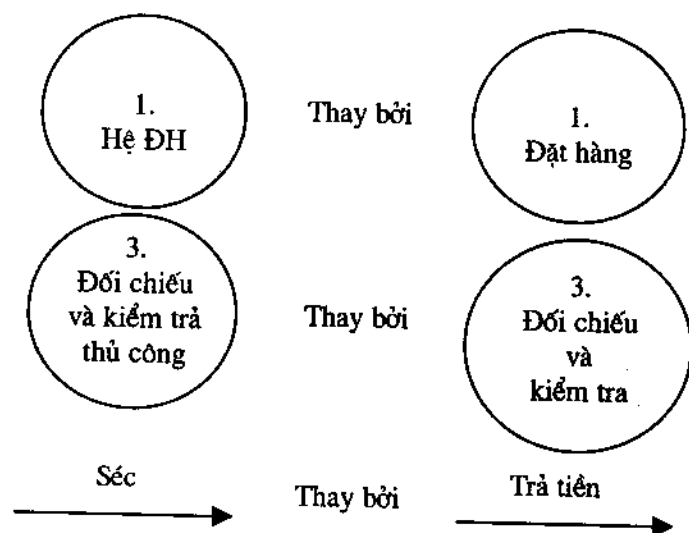
(2) Các chức năng vật lý, đó là những chức năng gắn liền với một công cụ hay một biện pháp xử lý nhất định, và sẽ không còn lý do tồn tại khi công cụ hay biện pháp đó bị thay đổi; chẳng hạn chức năng nhập dữ liệu vào máy tính sẽ không tồn tại nữa khi ta không dùng máy tính, và đó là một chức năng vật lý.

(3) Cấu trúc vật lý, ấy là cấu trúc chung của biểu đồ đang còn phản ánh trực tiếp cách bố trí, tổ chức hay cài đặt hiện tại, mà chưa phản ánh rõ bản chất logic

của hệ thống, chưa thoát ra ngoài mọi cách cài đặt cụ thể, chẳng hạn ở hệ cung ứng vật tư ở nhà máy Z có 3 tổ công tác là: đặt hàng, nhận hàng/phát hàng và đối chiếu. Do đó một cách tự nhiên, DFD mức đỉnh được thành lập với 3 chức năng tương ứng với 3 tổ chức đó. Cấu trúc như vậy là cấu trúc chủ quan, vì nó bị áp đặt bởi một hình thức tổ chức công việc có thể là còn tạm thời và chưa hẳn đúng thực chất.

Các yếu tố vật lý nói trên xen lẫn vào DFD, làm cho DFD phản ánh không đúng bản chất của hệ thống, cần phải gạt bỏ chúng khỏi biểu đồ.

Để loại bỏ các yếu tố vật lý loại (1), ta chỉ cần loại bỏ ra khỏi biểu đồ các phân ngôn từ hay hình vẽ biểu thị phương tiện, giá mang thông tin hay tác nhân, và chỉ giữ lại sự diễn tả nội dung của chức năng hay thông tin mà thôi, chẳng hạn:



Để loại bỏ các yếu tố loại (2), tức là các chức năng vật lý, trước hết phải làm cho chúng xuất hiện trong biểu đồ, thường thì các chức năng vật lý là các chức năng nhỏ, chưa xuất hiện ở các DFD mức trên. Vậy ta phải triển khai các DFD xuống các mức thấp. Khi phân rã mỗi chức năng thành các chức năng nhỏ, ta sẽ tách được các chức năng vật lý rời khỏi các chức năng logic. Từ đó ta loại các chức năng vật lý ra khỏi biểu đồ.

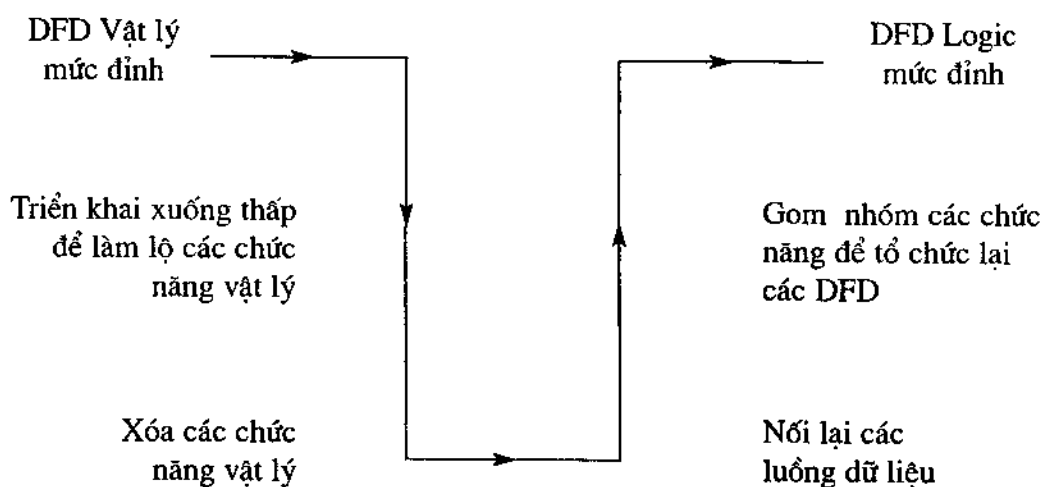
Sau khi loại bỏ hết các chức năng vật lý, thì các chức năng còn lại là các chức năng logic và ta đang ở một mức thấp nào đó. Bây giờ ta sẽ loại bỏ yếu tố

vật lý loại (3), tức là cấu trúc vật lý, bằng cách tổ chức lại các biểu đồ, từ dưới lên trên, xuất phát từ các chức năng logic nói trên, theo các bước như sau:

- Trước hết do có các chức năng vật lý bị loại, một số luồng dữ liệu đã bị đứt quãng, ta phải tìm cách nối chúng lại cho liên tục.

- Tiếp đó xem xét nội dung các chức năng logic, tìm cách gom cụm các chức năng gần gũi và hợp tác với nhau trong một mục đích xử lý vào một chức năng lớn, cho dù trước đây chúng bị chia lìa bởi các lý do cài đặt. Làm như thế, đang ở mức dưới, ta lại thành lập được DFD ở mức trên, và cứ thế tiếp tục, ta tổ chức lại các DFD ở các mức, cho đến mức đỉnh. Cấu trúc của các biểu đồ bấy giờ sẽ không còn mang tính chất vật lý nữa.

Tóm lại, từ DFD vật lý mức đỉnh, ta triển khai xuống các mức thấp, rũ bỏ các chức năng vật lý, rồi lại trở về các mức cao để chỉnh đốn lại cấu trúc của các biểu đồ. Rốt cục ta trở lại về mức đỉnh với một DFD logic. Quá trình đó được khái quát như sau:



Hình 28. Quá trình xây dựng một sơ đồ logic

### 3. Kỹ thuật chuyển từ DFD của hệ thống cũ sang DFD của hệ thống mới

Thường thì ít khi phải xây dựng mới hoàn toàn DFD logic của hệ thống mới, mà chỉ xây dựng nó từ DFD của hệ thống cũ, qua các bước thêm, bớt hay chỉnh đốn lại mà thôi.

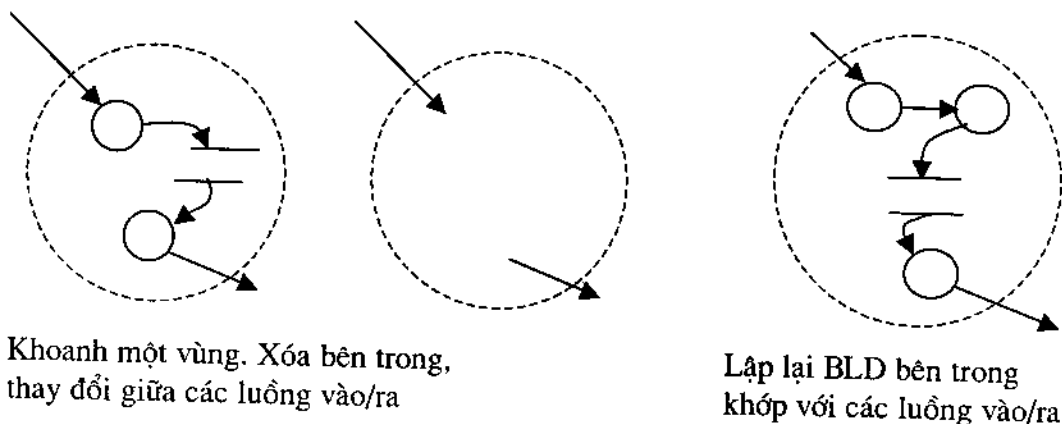
Để thực hiện sự chuyển đổi đó, trước hết cần xem lại các điểm phê phán đã đưa ra trước đây đối với hệ thống cũ, cũng như xem lại các mục tiêu và ưu tiên đã đề xuất đối với hệ thống mới. Nhờ sự phân tích chi tiết vừa tiến hành, nay có thể khẳng định lại, bổ sung hay sửa đổi các phê phán và các mục tiêu hay ưu tiên đó.

Căn cứ vào đó, đối chiếu với DFD logic của hệ thống cũ, ta phát hiện trong đó có những chỗ thiếu sót, dư thừa hay cần sửa đổi lại. Khoanh từng vùng đó lại và gọi đó là các vùng thay đổi.

Đối với từng vùng thay đổi đó, ta tiến hành các bước biến đổi như sau:

- Xóa phần DFD bên trong vùng, song vẫn giữ lại các luồng vào / ra (các luồng đi qua ranh giới của vùng);
- Xác định chức năng tổng quát mới của vùng thay đổi;
- Thiết lập một trung tâm biến đổi mới (tức là một DFD) thực hiện chức năng tổng quát nói trên, gồm: các chức năng hợp thành, các kho dữ liệu cần thiết và các luồng dữ liệu liên kết các chức năng và các kho đó;
- Nối trung tâm biến đổi đó với các luồng vào/ ra của vùng, nếu cần có thể bổ sung các chức năng biến đổi trung gian các luồng vào/ ra đó, hoặc thêm luồng vào / ra mới.

Mô tả các bước tiến hành trên:



Hình 29. Các bước chuyển từ DFD cũ sang DFD mới

### III. CÁC MÔ HÌNH VÀ PHƯƠNG TIỆN DIỄN TẢ DỮ LIỆU

#### 1. Khái niệm diễn tả dữ liệu

Một hệ thống trong trạng thái vận động bao gồm hai yếu tố là các tiến trình

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 mục này chúng ta chỉ tập trung 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 với nhau. Trong phần này chúng ta đề cập đến 4 công cụ chủ yếu:

- Mã hoá dữ liệu.
- Từ điển dữ liệu.
- Mô hình dữ liệu.
- Mô hình quan hệ.

## **2. Sự mã hoá**

### **2.1. 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 đề đặt ra là 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 hoá dữ liệu để phân loại dữ liệu lưu trữ và tìm kiếm có hiệu quả và bảo mật dữ liệu, đặc biệt trong các hệ thống thông tin xử lý bằng máy tính.

*Một số ví dụ về mã hoá:*

- 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.

### **2.2. Chất lượng và yêu cầu đối với mã hoá**

Trong thực tế ta gặp rất nhiều đối tượng cần mã hoá như mã hoá ngành nghề cần 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 hoá không được nhập nhằng: Thể hiện ánh xạ 1-1 giữa mã hoá và giải mã mỗi đối tượng được xác định rõ ràng và duy nhất với một mã nhất định.

- Thích ứng với phương thức sử dụng: Việc mã hoá 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 vào cuối các mã đã có.

- + Xen mã mới vào giữa các mã đã có, thường mã xen phải dùng phương pháp cóc nhảy, nhảy đều đặn dựa vào thống kê để tránh tình trạng “bùng nổ ” mã.

- 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.

- Cách mã cần xác định sao cho tối thiểu hoá sai sót khi mã và giảm tính dư thừa của mã.

### **2.3. Các kiểu mã hoá**

#### **(1) Mã hoá liên tiếp (Serial Coding):**

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 (Range Coding):**

Sử dụng các số nguyên như mã hoá liên tiếp nhưng phân ra theo lát cho từng loại đối tượng, trong mỗi lát dùng mã liên tiếp.

Ví dụ: Mã hoá các đối tượng là các hàng ngũ kim.

Vùng 1: 0001 – 0999 để mã hóa các hàng ngũ kim bé, trong đó:

0001 – 0099 để mã hóa các loại vít

0100 – 0299 để mã hóa các loại ốc

0300 – 0499 để mã hóa các loại bulong

0500 – 0599 để mã hóa các loại đinh

....

Vùng 2: 1000 – 1999 để mã hóa các chi tiết kim loại, trong đó

1000 – 1099 để mã hóa các loại sắt chữ U

....



**Ưu điểm:** Không nhập nhằng, đơn giản, có thể mở rộng và xen thêm được.  
**Nhược điểm:** Vẫn phải dùng bảng tương ứng.

**(3) Mã phân đoạn:**

Mã được phân thành nhiều đoạn, mỗi đoạn mang một ý nghĩa riêng. Ví dụ, số đăng kí xe máy có biển số xe là **29 F6 6956** 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.

**Chương 1**

**1.1 Bài 1**

**1.2 Bài 2**

**1.3 Bài 3**

**Chương 2**

**2.1 Bài 4**

**2.1.1 Mục 1**

**2.1.2 Mục 2**

**2.2 Bài 5**

**2.3 Bài 6**

....

**Ưu điểm:** Các ưu điểm tương tự như mã hoá phân đoạn. ngoài ra việc tìm kiếm mã dễ dàng.

**Khuyết điểm:** Tương tự như các nhược điểm của mã phân đoạn.

**(5) Mã diễn nghĩa (Memonic Coding):** Bằng cách gán một tên ngắn gọn nhưng hiểu được cho một đối tượng.

Ví dụ: Đội bóng các nước tham gia giải Tiger cup được mã bằng cách lấy ba kí tự đầu như sau:

VIE: Việt Nam, THA: Thailand, SIN: Singapore, IND: Indonesia, MAL: Malaysia.

**Ưu điểm:** Tiện dùng cho xử lý bằng thủ công.

**Khuyết điểm:** Không giải mã được bằng máy tính.

### ***Các chú ý khi lựa chọn sự mã hoá***

Như đã nêu ở trên, có nhiều phương pháp mã hoá khác nhau, có thể sử dụng một kiểu mã nào đó, cũng 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 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 nào được dùng trước đó cho các đối tượng này để kế thừa.
- Thỏa thuận với người dùng cách mã.
- Thử nghiệm trước khi dùng chính thức để chỉnh lý kịp thời.

## **3. Từ điển dữ liệu**

### **3.1. Khái niệm**

Từ điển dữ liệu là một tư liệu tập trung mọi tên gọi của mọi đối tượng được dùng trong hệ thống trong cả các giai đoạn phân tích, thiết kế, cài đặt và bảo trì. nó là văn phạm giả hình thức mô tả nội dung của các sự vật, đối tượng theo định nghĩa có cấu trúc. Chẳng hạn trong biểu đồ luồng dữ liệu (DFD) các chức năng xử lý, kho dữ liệu, luồng dữ liệu chỉ mô tả ở mức khái quát thường là tập hợp các khoản mục riêng lẻ. Các khái quát này cần được mô tả chi tiết hơn qua công cụ từ điển dữ liệu.

### **3.2. 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, thiết kế, cài đặt và bảo trì hiểu chung cái vào, cái ra, cái luân chuyển. Ký 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	[1]	hoặc
	< @ <sup>n</sup>	Lặp n lần
	()	Dữ liệu tùy chọn
	*Lời chú thích*	Giới hạn chú thích

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

Số HD: 124 hoá đơn bán hàng Ngày 01-10-04 Bán cho ông/bà: <b>Trần Văn Ngọc</b> Tài khoản: LTM010254 Địa chỉ: 534 Trần Hưng Đạo Tel: (04) 8226465/8692205						
Số TT	Mã hàng	Tên, quy cách	Đơn vị	Đơn giá	Số lượng	Thành tiền
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						
Bằng chữ						
Kế toán trưởng		Người nộp tiền			Người bán hàng	

\* *Xác định thông tin về hoá đơn:*

Hoá đơn = Số HD + Ngày bán + Khách hàng + {Mặt hàng + Số lượng + Thành tiền}<sup>n</sup> + Tổng cộng + kế toán trưởng + 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

## 4. Mô hình thực thể

### 4.1. Khái niệm

Phân tích dữ liệu là một phương pháp xác định các đơn vị thông tin cơ sở có ích cho hệ thống, và định rõ mối quan hệ bên trong hoặc các tham khảo giữa chúng. Điều này có nghĩa là mọi phần dữ liệu sẽ chỉ được lưu giữ một lần trong toàn bộ hệ thống của tổ chức và có thể thâm nhập được từ bất kì chương trình nào; phải có chỗ cho mọi thứ đều ở đúng chỗ của nó.

Thông tin trong nó được cấu trúc và tổ chức như thế nào? Câu trả lời là

bảng: Mọi thông tin trong cơ sở dữ liệu về mặt logic đều có thể được giữ trong các bảng đơn giản, mỗi bảng đều có một số cột cố định. Kỹ năng phân tích dữ liệu chính là kỹ thuật gán cho từng phần thông tin những bảng thích hợp nhất.

Ở giai đoạn phân tích phải định ra được các loại bảng cần cho dữ liệu và phải chỉ ra mối quan hệ tự nhiên giữa các thực thể trong các bảng khác nhau, công cụ sử dụng cho việc này chính là mô hình thực thể.

Ví dụ về một số bảng:

Khách hàng

Số hiệu khách hàng	Tên khách hàng	Tín dụng	Địa chỉ
A137	J.SMITH	A	10 HIGH ST BUBE
A257	CATESBY	B	20 LOW ST BATH
B340	BOB&CO	A	15 DIPLOCK SALE
B751	DEVANE%SON	A	8 THE QUANGARE RYDE

Đơn hàng

Số đơn hàng	Số hiệu khách hàng	Ngày tháng	Địa chỉ giao nhận
2001	B137	5.7.95	Z5 TIP TON WICK
2010	B340	7.7.95	7 ANDOVER MEWS
2013	A137	8.7.95	25 TIPTON WICK

Hình 30. Một số bảng

Hình 31 cho một ví dụ về mô hình thực thể. Mỗi hộp trên sơ đồ biểu thị cho một bảng riêng biệt (tên hộp mô tả cho bản chất của mỗi ô hoặc hàng trong bảng), và các dòng và mũi tên chỉ ra mối quan hệ. Mô hình thực thể được xây dựng từ các khái niệm logic chính:

1. Thực thể
2. Kiểu thực thể
3. Thuộc tính
4. Quan hệ



Hình 31. Một mô hình thực thể

*a. Thực thể:*

Thực thể là một chủ điểm, một nhiệm vụ, một đối tượng, hay một sự kiện đáng quan tâm đối với tổ chức, kể cả những thông tin mà nó giữ. Sau đây là thí dụ về các thực thể.

Công ty TNHH Lưu Lan (khách hàng)

Đơn hàng số 0234

Mục kho AB23

Một thực thể tương đương với một dòng trong một bảng nào đó.

*b. Kiểu thực thể:*

Kiểu thực thể là việc nhóm tự nhiên một số thực thể lại, mô tả cho một loại thông tin chứ không phải là bản thân thông tin. Chẳng hạn, “Tiền Lợi” là một thực thể, được để ý tới vì ông ta mua hàng của công ty, ông ta là khách hàng. “Khách hàng” là một kiểu thực thể vì nó mô tả cho một số thực thể và dựa trên đó thông tin được lưu giữ.

Kiểu thực thể tương đương với bảng và có dạng hộp trong sơ đồ mô hình thực thể.

Các kiểu thực thể quan trọng nhất thường rơi vào một trong ba phạm trù sau:

1. Thông tin liên quan tới một trong các giao dịch chủ yếu của hệ thống (chẳng hạn đơn hàng bán, đơn hàng mua...)

2. Thông tin liên quan tới các thuộc tính hoặc tài nguyên của hệ thống (chẳng hạn kho tàng, khách hàng, nhà cung cấp, đội ngũ cán bộ, nguyên vật liệu v.v...)

3. Thông tin đã khái quát, thường dưới dạng thống kê, liên quan tới vạch kế hoạch hoặc kiểm soát (chẳng hạn như dự đoán, ngân sách, tính lương, lịch điều xe v.v...)

Cách tiếp cận đơn giản nhất trong việc quyết định xem liệu một kiểu thông tin có nên đưa vào như một kiểu thực thể trong mô hình hay không và việc tự hỏi liệu bằng các thông tin như vậy có ích lợi gì cho hệ thống không và trong

trường hợp có ích thì phải xác định rõ các thực thể cơ bản sẽ tạo ra một dòng trong bảng.

*c. Thuộc tính.*

Sau khi đã xác định được kiểu thực thể thích hợp (bảng) và bản chất của thực thể (dòng), bước tiếp theo là xác định những thông tin nào cần phải được lưu giữ cho mỗi thực thể. Thuộc tính là các đặc trưng của thực thể, biểu thị bằng các trường hoặc cột của bảng. Chẳng hạn, trong hình 30 các thuộc tính của khách hàng là:

Số hiệu khách hàng

Tên khách hàng

Số tiền tín dụng

Địa chỉ khách hàng

Có ba kiểu thuộc tính khác nhau và bất kỳ một kiểu thực thể đặc biệt nào cũng đều có thể có một số thí dụ về cả ba kiểu này.

Các kiểu đó là:

1. Tên gọi

2. Mô tả

3. Kết nối

*Thuộc tính tên gọi.*

Mỗi thực thể trong bảng đều phải được xác định duy nhất. Không thể có hai hoặc nhiều phần tử có cùng tên gọi. Một tên gọi (còn gọi là “khóa”) là một hoặc nhiều thuộc tính trong kiểu thực thể được dùng để gán cho mỗi thực thể một cách tham khảo duy nhất.

Chẳng hạn xem hình 30. Số hiệu khách hàng là tên gọi cho kiểu thực thể “khách hàng”. Điều này có nghĩa là mọi khách hàng trong bảng đều phải có một số hiệu khách hàng khác nhau. Thuộc tính tên khách hàng có thể là một ứng cử viên cho tên gọi, nhưng bao giờ cũng có khả năng trùng tên giữa hai hoặc nhiều khách hàng, khi đó sẽ phải có sự trùng nhau trong các dòng tương ứng.

Số hiệu đơn hàng rõ ràng là một tên gọi cho kiểu thực thể đơn hàng.

Tuy nhiên, các thực thể trong “dòng đơn hàng” (chính là một bảng chứa mọi mặt hàng trong các đơn hàng trong hệ thống) không được xác định duy nhất từ một thuộc tính. Cả số hiệu đơn hàng lẫn số hiệu mặt hàng đều được dùng độc lập; có thể có nhiều mặt hàng từ cùng một đơn hàng và cùng một mặt hàng được yêu cầu ở nhiều đơn hàng. Tên gọi hợp lệ duy nhất cho kiểu thực thể này là việc

tổ hợp cả hai thuộc tính số hiệu đơn hàng và số hiệu mặt hàng.

*Thuộc tính mô tả.*

Hầu hết các thuộc tính trong một kiểu thực thể đều có thể là mô tả. Đây là những đơn hàng thông tin mô tả cho thực thể được tham trỏ tới. Thông tin này làm tăng hiểu biết của ta về thực thể và sẽ phục vụ cho các mục đích có ích bên trong hệ thống. Thí dụ về các mô tả trong hình 30 là:

Đối với kiểu thực thể khách hàng.

Tên khách hàng

Số tiền tín dụng

Địa chỉ

Đối với kiểu thực thể đơn hàng

Ngày

Địa chỉ giao hàng

Đối với kiểu thực thể mặt hàng

Số lượng giao hàng

Đơn vị của mặt hàng

*Thuộc tính kết nối.*

Mục đích của thuộc tính kết nối là chỉ ra mối quan hệ giữa một thực thể đã có và một thực thể khác trong một bảng khác. Thuộc tính kết nối rất giống với thuộc tính mô tả thông thường trong bản thân thực thể đó nhưng ở một thực thể khác trong mô hình thì nó lại là một tên gọi.

Tốt nhất là tên minh họa bằng một thí dụ.

Trong hình 30 bảng Đơn hàng chứa thuộc tính số hiệu khách hàng:

Hiển nhiên nó là quan trọng vì nó cho biết ai đã đặt đơn, do đó thông tin này phải có mặt. Tuy nhiên, cùng một thuộc tính số hiệu khách hàng này lại xuất hiện trong bảng khách hàng, trong bảng đó nó là thuộc tính tên gọi. Điều này chỉ ra mối quan hệ giữa một khách hàng và một đơn hàng (đơn hàng phải do một khách hàng nào đó đặt). Nó cũng nói lên rằng không cần phải giữ mọi thông tin khách hàng khác (chẳng hạn địa chỉ vận đơn) trong bảng thực thể đơn hàng. Nếu ta muốn biết địa chỉ đó thì ta có thể dùng số hiệu khách hàng trong đơn hàng để dò tìm tiếp trong bảng khách hàng. Đó chính là cách làm việc của bộ kết nối.

Hình 33 minh họa rõ thêm về hiệu quả của thuộc tính kết nối trong bảng, việc sử dụng các thuộc tính như vậy có liên hệ chặt chẽ với việc xác định chủ

đề tiếp của chúng ta và toàn bộ khái niệm này phải trở thành rõ ràng dần trong việc triển khai tiếp chương này.

Khách hàng

Số hiệu khách hàng	Tên khách hàng	Tín dụng
A137	Tiến Lợi	A
A257	Giang Hồng	B

Đơn hàng bán.

Số hiệu đơn hàng	Số hiệu khách hàng	Ngày tháng
2001	A137	15/3/2004
2010	B340	17/3/2004

Hàng hoá.

Số hiệu hàng	Tên hàng	Đơn vị
63B	Tăng trọng H02	bao
54A	Tăng trọng H03	túi

Dòng đơn hàng

Số hiệu đơn hàng	Số hiệu mặt hàng	Số lượng giao
2001	63B	25
2001	47B	14

*Hình 33. Bảng thực thể thuộc tính kết nối cao*

Như ta đã biết, mối quan hệ tự nhiên xuất hiện giữa các thực thể các kiểu khác nhau. Chẳng hạn ta có thể thấy rằng các mối quan hệ tồn tại giữa một khách hàng và một đơn hàng (khách hàng đặt đơn hàng), một đơn hàng và một mặt hàng trong đơn hàng, một sản phẩm và một nhà cung cấp (nhà cung cấp bán sản phẩm). Bản chất của các mối quan hệ này là tổ chức và tạo nên cách sử dụng trong việc điều khiển hoạt động nghiệp vụ.

Những mối quan hệ như vậy được biểu diễn trên mô hình thực thể bằng các đường có mũi tên hoặc dấu tam giác.

Có ba kiểu quan hệ chính được sử dụng dưới các dạng đơn giản nhất của Mô hình thực thể:

1. Một - một



2. Một - nhiều

3. Nhiều - nhiều

Ba kiểu này liên quan với số các thực thể trong một bảng có quan hệ với một hoặc nhiều thực thể trong các bảng khác. Các quan hệ này được định nghĩa là thông qua các thí dụ.

*Quan hệ một - một.*

Giả sử có hai bảng thực thể A và B, mỗi quan hệ một - một là tồn tại nếu:

- Với mỗi thực thể trong bảng A đều có một thực thể tương ứng trong bảng B.
- Với mỗi thực thể trong bảng B đều có một thực thể tương ứng trong bảng A.

Kí hiệu



*Quan hệ một - nhiều*

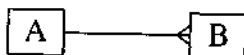
Giả sử có cùng một bảng thực thể A và B trên, quan hệ một - nhiều là tồn tại nếu:

\* Với mỗi dòng trong bảng thực thể A đều có nhiều dòng trong bảng B nhưng

\* Với mỗi dòng trong bảng B chỉ có một và chỉ một dòng trong bảng A.

Một quan hệ một - nhiều được vẽ trong mô hình thực thể bằng một đường hình tam giác ở một đầu; đầu không đánh dấu chỉ ra rằng bảng với một dòng còn bảng ở đầu tam giác sẽ chứa nhiều dòng.

Kí hiệu



*Quan hệ nhiều - nhiều*

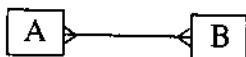
Ta vẫn dùng hai bảng thực thể tương tượng A và B, mỗi quan hệ nhiều - nhiều tồn tại nếu:

\* Với mỗi thực thể trong bảng A có nhiều thực thể trong bảng B

\* Với mỗi thực thể trong bảng B có nhiều thực thể trong bảng A

Quan hệ nhiều - nhiều được biểu diễn trên mô hình bởi một đường với kí hiệu tam giác ở cả hai đầu.

Kí hiệu



## 4.2. Phân tích mô hình thực thể

Có nhiều cách tiếp cận khác nhau để xây dựng mô hình dữ liệu ban đầu. Cách tiếp cận được nói tới ở đây là một cách tiếp cận đơn giản nhất về mặt kỹ

thuật, và có thể nói rằng cách tiếp cận này phụ thuộc vào mức độ hiểu biết của độc giả về các thực thể và các mối quan hệ thiết lập giữa chúng. Nó cho phép các nhà xây dựng mô hình dữ liệu có kinh nghiệm tiến rất nhanh, nhưng nó không đưa ra được, như một số cách tiếp cận đã làm, một loại bước hình thức để những người tập sự có thể theo được.

Có hai giai đoạn trong quá trình xây dựng mô hình

1. Xác định các kiểu thực thể.

2. Xác định các mối quan hệ.

Từng giai đoạn một trên đây sẽ cung cấp tài liệu xuất phát cho những phân tích chi tiết hơn trong quá trình xây dựng mô hình.

*Xác định kiểu thực thể.*

Giai đoạn này bao gồm việc xác định các bảng chính để giữ thông tin về hệ thống trong lĩnh vực nghiên cứu. Không cần phải xác định ngay lập tức mọi kiểu thực thể, thực tế quá trình xây dựng mô hình sẽ “tháo gỡ” dần các bảng rắc rối. Cho nên mục tiêu của giai đoạn đầu này là nêu ra được các ứng cử viên hiển nhiên để xem xét và mở rộng về sau.

Một cách tiếp cận có ích trong việc xác định các kiểu thực thể ban đầu này là dùng ba phạm trù đã nói ở đầu chương, và xem xét toàn bộ lĩnh vực nghiên cứu làm nảy sinh vấn đề.

Bảng thực thể nào cần có để giữ các chi tiết về:

Các giao tác chính trong hệ thống?

(Chẳng hạn đơn hàng, phiếu giao hàng...)

Các tài sản và tài nguyên chính được dùng trong hệ thống?

(Chẳng hạn nhà cung cấp, nhân dân...)

Các thông tin về lập kế hoạch và kiểm soát?

(Chẳng hạn thời gian biểu, dự toán chi tiết...)

Cách tiếp cận thứ hai và thường được dùng nhiều hơn là lấy một bản mô tả về hệ thống hiện tại hoặc hệ thống cần có rồi xem xét cẩn thận từng danh từ và động từ mô tả, bằng cách đặt ra các câu hỏi:

Liệu tổ chức có cần lưu giữ thông tin về điều này không, dưới dạng một dòng trong bảng hoặc nguyên cả một bảng?

Tốt nhất là ta nên minh họa cho điều này thông qua một thí dụ. Hình 8.16 nêu ra một mô tả tóm tắt cho hệ thống xử lý vật tư của công ty sản xuất giấy. Hình 3.17 nêu ra cùng bản mô tả đó nhưng đã có xác định các danh từ và động từ như các kiểu thực thể ứng cử viên, được khoanh tròn lại.

Một số danh từ trong đoạn mô tả này không tạo nên bảng thực thể. Chẳng hạn “kho tàng” không phải là bảng thực thể, vì chỉ có một kho hàng trong hệ thống. Các từ như “chất lượng” và “sản xuất” có thể cuối cùng cũng có liên quan

tới một kiểu thực thể nào đó, nhưng ở giai đoạn này định nghĩa về chúng quá rộng đối với ta. Điều đó cũng đáng lưu tâm là động từ “đã giao nhận” được xác định như một kiểu thể tiềm năng ta có thể dựa trên tri thức về nghiệp vụ của mình để quyết định về điều đó, thừa nhận sự tồn tại của một giao tác “giao nhận” trong hệ thống như vậy. Các kiểu thực thể khác cũng được làm rõ ràng ra; ta có thể giữ một bảng tất cả các nhà cung cấp, bảng tất cả các đơn mua hàng, bảng các vật tư trong kho.v.v... có thể là ở giai đoạn này một số kiểu thực thể quan trọng nhất còn chưa được xác định ra, nhưng ta cũng không phải bận tâm lắm về điều này. Chúng sẽ được phát hiện ra trong các giai đoạn sau của quá trình mô hình hoá.

Vật tư được các nhà cung cấp giao hàng, và khi chúng được gửi tới, người ta sẽ kiểm tra lại chúng theo bản sao của đơn hàng mua gửi cho nhà cung cấp. Vật tư được giám định chất lượng và giả sử rằng chúng thoả mãn yêu cầu, khi đó sẽ phát sinh một thẻ kho thích hợp ghi việc cất giữ nó. Vật tư nằm trong kho cho tới khi công việc sản xuất yêu cầu tới nó và lúc đó sẽ phát sinh một lệnh sản xuất cần tới vật tư. Vật tư đó sẽ được lấy ra và phân phát cho phân xưởng có yêu cầu.

Một mô tả về hệ thống được đề đạt nghiên cứu

Vật tư được các nhà cung cấp giao hàng, và khi chúng được gửi tới, người ta sẽ kiểm tra lại chúng theo bản sao của đơn hàng mua gửi cho nhà cung cấp. Vật tư được giám định chất lượng giả sử rằng chúng thoả mãn yêu cầu, khi đó sẽ phát sinh một thẻ kho thích hợp ghi việc cất giữ nó. Vật tư nằm trong kho cho tới khi công việc sản xuất yêu cầu tới nó và lúc đó sẽ phát sinh một lệnh sản xuất cần tới vật tư. Vật tư đó sẽ được lấy ra và phân phát cho phân xưởng có yêu cầu.

Vật tư

Giao hàng

Nhà cung cấp

Đơn mua hàng

Thẻ kho

Lệnh sản xuất

Phân xưởng sản xuất

Các kiểu thực thể được chính xác

### *Xác định các quan hệ*

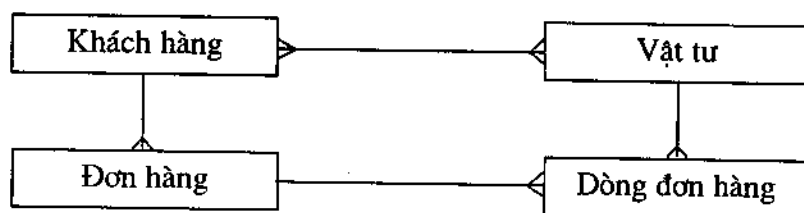
Sau khi đã xác định các kiểu thực thể chính, vấn đề bây giờ xác định các liên kết tự nhiên giữa chúng, và phải ghi các liên kết này dưới dạng quan hệ một - nhiều. Ba phát biểu sau có thể được lấy làm hướng dẫn cho việc tìm các quan hệ đó:

1. Một quan hệ tồn tại giữa hai thực thể nếu cần phải giữ thông tin trong thực thể này về kiểu thực thể kia. Lý do cho việc giữ thông tin kết nối này là bản chất của quan hệ.

2. Trong mọi quan hệ một – nhiều, thực thể giữ thông tin kết nối theo định nghĩa là ở đầu “nhiều”.

3. Các quan hệ “gián tiếp” nên được bỏ qua. Một quan hệ gián tiếp là quan hệ mà hai thực thể có liên quan được kết nối thông qua thực thể thứ ba. Chẳng hạn một khách hàng có thể đặt một dòng đơn hàng đặc biệt, nhưng mối quan hệ giữa khách hàng và dòng đơn hàng không phải trực tiếp là một - nhiều; mối quan hệ đúng là giữa Khách hàng và Đơn hàng, rồi giữa Đơn hàng và Dòng đơn hàng.

Nhà phân tích sẽ chọn hai hoặc ba kiểu thực thể là kiểu thực thể quan trọng nhất, thông thường các kiểu thực thể này đều có liên quan tới tiêu chuẩn Tài sản hoặc Tài nguyên; (trong thí dụ của ta đó là Nhà cung cấp, Vật tư). Rồi sau đó cố gắng xác định ra mối quan hệ giữa chúng. Ta có thể thấy ngay rằng một nhà cung cấp có thể cung cấp nhiều loại vật tư khác nhau, và mọi dạng vật tư đều có thể được cung ứng bởi nhiều nhà cung cấp. Công ty phải lưu giữ một danh sách những nhà cung cấp vật tư, thì dựa trên đó mới lên được Đơn hàng mua, và danh sách này được biểu thị bằng kiểu thực thể nhà cung cấp/vật tư.



## 5. Mô hình quan hệ

Dạng mô hình hoá này là một phần của cách tiếp cận tổng thể tới việc phân tích dữ liệu. Mô hình quan hệ được sử dụng như việc kế tục của quá trình mô

hình hoá dữ liệu, nhằm kiểm tra, cải tiến và mở rộng mô hình dữ liệu đã xây dựng.

Quá trình tạo ra một mô hình quan hệ và dùng nó để kiểm tra mô hình dữ liệu bao gồm việc thực hiện một số bước, sẽ được thảo luận chi tiết hơn trong chương này:

1. Xác định tất cả các thuộc tính cần dùng tới trong hệ thống định xây dựng.
2. Xác định kiểu thực thể để đặt từng thuộc tính nhằm giảm thiểu việc sao chép và tránh dư thừa. (Kỹ thuật được dùng cho việc này có tên là chuẩn hóa).
3. Xác định các quan hệ vốn tiềm ẩn bên trong các danh sách thuộc tính đã được thiết lập cho từng kiểu thực thể. Điều này được thực hiện bằng cách ghi ra những thuộc tính nào trong từng bảng là thuộc tính kết nối. Mọi thuộc tính kết nối đều biểu diễn cho đầu “nhiều” của quan hệ “một-chiều”.
4. Với các thuộc tính, kiểu thực thể và quan hệ đã biết, bây giờ có thể xây dựng một sơ đồ cùng kiểu như Mô hình dữ liệu trực giác. Khi đó ta có thể So sánh các mô hình và trích ra được từ việc so sánh đó một mô hình duy nhất có chứa các đặc trưng tốt nhất của cả hai.

5. Sau khi đã xác định được khuôn dạng tốt nhất cho Mô hình dữ liệu để biểu diễn các yêu cầu của hệ thống, ta có thể phải ước lượng về khối lượng thực thể cho từng bảng, và điều này cũng được ghi lại trong mô hình.

Mặc dù việc tiến hành các bước trên có thể tốn thời gian và các nguồn tài nguyên khác, nhưng ích lợi của cách tiếp cận chặt chẽ thường lớn hơn giá phải trả nhiều. Tuy nhiên cũng có khi hệ thống đơn giản tới mức chẳng cần phải “xử lý đầy đủ” như vậy. Nhưng phân tích cần có cách đánh giá riêng của mình trong trường hợp như vậy.

### **5.1. Xác định thuộc tính**

Nhà phân tích có thể dựa vào ba nguồn cung cấp cơ bản để lấy được các chi tiết về những ‘thuộc tính’ tiềm năng của các thực thể trong hệ thống:

1. Từ tri thức của chính mình về thực tế công việc chung trong lĩnh vực đang nghiên cứu mà ‘dự đoán’ hoặc xác định một cách trực cảm về các thuộc tính,
2. Từ những người sử dụng, thông qua quá trình phỏng vấn; (chẳng hạn “tổ chức lưu giữ những thông tin gì về các nhà cung cấp?”),
3. Từ việc xem xét các khuôn dạng biểu báo và các tài liệu khác được sử dụng thường xuyên trong lĩnh vực đang nghiên cứu. Vậy, với từng kiểu thực thể trong mô hình dữ liệu, nhà phân tích sẽ phải xây dựng một danh sách các

thuộc tính dự tuyển. Một số thuộc tính có thể được xem như thuộc vào nhiều kiểu thực thể, trong trường hợp đó, ở giai đoạn này, chúng cần được đặt cả vào danh sách thích hợp; vấn đề về sự sao chép này được xem xét lại ở giai đoạn sau của quá trình.

Có một tập các chuẩn được sử dụng để ghi ra những danh sách này. Tất cả các chuẩn đó kết hợp cùng nhau tạo nên mô hình quan hệ.

Phải viết ra tên kiểu thực thể (viết chữ hoa),

Viết ra các thuộc tính theo bất kỳ thứ tự ở các dạng tiếp theo, mỗi thuộc tính cách nhau một dấu phẩy,

Các thuộc tính khoá phải được in nghiêng (thông thường chúng được đặt ở đầu danh sách cho tiện).

Khách hàng

*Số liệu khách hàng*, Tên khách hàng, Địa chỉ

Đơn đặt hàng

*Số hiệu đơn*, *Số hiệu khách hàng*, Ngày đơn hàng

Dòng đơn hàng

*Số hiệu đơn*, *Số hiệu mặt hàng*, Số lượng, Đơn vị

Mặt hàng

*Số hiệu mặt hàng*, Tên hàng, Đơn vị

Giao nhận

*Số hiệu giao nhận*, *Số hiệu khách hàng*, Ngày giao

Dòng giao nhận

*Số hiệu giao nhận*, *Số hiệu đơn hàng*, *Số hiệu mặt hàng*, Lượng giao nhận

Hình 35. Trích từ mô hình quan hệ

## 5.2. Xác định các kiểu thực thể: chuẩn hoá

### a. Mô tả

Chuẩn hoá là quá trình khảo sát các danh sách thuộc tính, và áp dụng một tập các quy tắc phân tích vào các danh sách đó, chuyển chúng thành dạng:

Tối thiểu việc lặp lại,

tránh dư thừa,

xác định và giải quyết sự nhập nhằng.

Sự lặp, tình trạng cùng một thuộc tính có mặt ở nhiều bảng thực thể, chỉ xuất hiện đối với các thuộc tính tên gọi và kết nối, và là cần thiết để thể hiện các mối quan hệ.

Dư thừa có thể xuất hiện dưới dạng thuộc tính không khoá trong nhiều bảng, hoặc có thể trở tới dữ liệu 'suy diễn'. Các thuộc tính có các giá trị là kết quả có từ tính toán đơn giản được thực hiện trên các thuộc tính khác cần phải được loại trừ khỏi mô hình. Thí dụ về một thuộc tính như thế là 'giá thành' trong danh sách: Số hiệu mặt hàng, số lượng, đơn giá, giá thành (Trong đó 'giá thành' chỉ là 'số lượng' nhân với 'đơn giá').

Việc lưu trữ các giá trị suy diễn được này không chỉ làm tốn phí không gian lưu trữ mà còn tạo khả năng làm xuất hiện những sự không nhất quán. Nếu một dòng của bảng có các giá trị:

Số lượng 25,

Đơn giá 4,

Giá thành 124,

thì bản thân thực thể đã mang lỗi ngay từ đầu.

Nhập nhằng có thể xuất hiện trong tình huống có nhiều người sử dụng thông tin. Nó có thể xuất hiện dưới dạng một thuộc tính chưa được hiểu rõ hoặc dưới dạng mang nhiều nghĩa khác nhau đối với nhiều người. Quá trình chuẩn hoá buộc người ta phải xem xét rất cẩn thận về ý nghĩa của từng thuộc tính và ta chỉ có thể xây dựng được mô hình quan hệ chừng nào đã hiểu rõ tất cả các thuộc tính liên quan. Nhà phân tích thường hỏi những câu hỏi đại loại như: Liệu 'số khách hàng' có trùng với 'số hiệu tài khoản khách hàng' không?

Có quan hệ nào giữa 'kiểu sản phẩm' và 'lớp kho' không?

#### *b. Phụ thuộc hàm*

Quá trình chuẩn hoá được thực hiện dựa trên khái niệm "phụ thuộc hàm" và mô hình được chuẩn hoá đầy đủ, lý tưởng là mô hình ở đây mỗi thuộc tính trong mỗi bảng thực thể đều có một phụ thuộc hàm trực tiếp vào các thuộc tính khoá của bảng. Phụ thuộc hàm nghĩa là:

Với mọi giá trị của khoá tại mọi thời điểm được xét, chỉ có một giá trị cho từng thuộc tính khác trong bảng.

Chẳng hạn, trong bảng khách hàng,

Số hiệu khách hàng, tên khách hàng, địa chỉ khách hàng.

Khách hàng có thể thay đổi địa chỉ của mình, nhưng vào mọi lúc, với khoá

đã cho (Số hiệu khách hàng) ta có thể nói rằng có một giá trị cho tên và địa chỉ.

Nếu một thuộc tính không phụ thuộc hàm vào khoá (theo định nghĩa của chúng ta), thì nó phải nằm ở bảng thực thể khác. Chẳng hạn, trong bảng thực thể đơn hàng.

Số hiệu đơn hàng, Ngày đặt hàng, Số hiệu khách hàng, mặt hàng ta không thể nói rằng, với khoá đã cho (Số hiệu đơn hàng), chỉ có một giá trị cho thuộc tính Khoản mục đơn hàng; kiến thức nghiệp vụ thông thường cho ta biết rằng một đơn hàng có thể có nhiều mặt hàng đặt hàng.

Vì thuộc tính Khoản mục đơn hàng không phụ thuộc hàm vào khoá nên nó không là thuộc tính hợp lệ cho kiểu thực thể này và phải được cất ở bảng thực thể khác.

### *c. Ba dạng chuẩn*

Ta gọi danh sách các thuộc tính của một bảng thực thể đặc biệt là 'chưa chuẩn hoá' nếu nó chưa được xét qua trong quá trình chuẩn hoá. Quá trình chuẩn hóa bao gồm việc áp dụng ba quy tắc kiểm tra liên tiếp nhau.

Nếu danh sách các kiểu thực thể qua được quy tắc kiểm tra thứ nhất thì nó được gọi là có 'dạng chuẩn thứ nhất' (1NF)

Nếu nó qua được quy tắc kiểm tra thứ hai thì nó sẽ được gọi là có 'dạng chuẩn thứ hai' (2NF)

Nếu nó qua được quy tắc kiểm tra thứ ba thì nó sẽ được gọi là có 'dạng chuẩn thứ ba' (3NF), và được xem như đã được chuẩn hoá đầy đủ.

(Phải nhấn mạnh rằng còn có một số 'dạng chuẩn' nữa, nhưng chúng không được sử dụng rộng rãi trong việc phân tích các hệ thống nghiệp vụ).

Nếu một danh sách các kiểu thực thể không qua được một phép kiểm tra nào thì điều này có nghĩa là cần phải bỏ bớt đi một hoặc nhiều thuộc tính bên trong nó, và tạo nên một bộ phận của các thực thể khác. Điều này có nghĩa là nhà phân tích có thể bắt đầu với một danh sách các thuộc tính dự định đối với một kiểu thực thể, và sau khi áp dụng ba quy tắc chuẩn hoá anh ta có thể thấy rằng nên gán một số trong các thuộc tính đó cho các kiểu thực thể khác. Cho đến hết các giai đoạn chuẩn hoá thì anh ta không chỉ có kiểu thực thể gốc mà còn có cả kiểu thực thể mới được xác định và tất cả chúng đều chuẩn hoá hoàn toàn.

### *d. Chọn khoá*

Nhiệm vụ đầu tiên, khi có trước một danh sách các thuộc tính đối với một



kiểu thực tế, là chọn một khoá. Khoá đó gồm một hoặc nhiều thuộc tính có giá trị cung cấp một địa danh duy nhất cho mọi dòng trong bảng: không có hai thực thể nào trong một kiểu thực thể có cùng khoá.

Thường thì việc chọn khoá là trực tiếp nhưng đôi khi cũng cần phải suy nghĩ. Một số thuộc tính, thoáng nhìn thì có vẻ cho được tính duy nhất nhưng sau khi xem xét kỹ thì sẽ thấy rằng nó có thể gây nguy cơ trùng lặp. Chẳng hạn,

1. Trong bảng thực thể Khách hàng, tên “khách hàng” hình như là một khoá đầy hứa hẹn. Tuy vậy, hiển nhiên là trong hầu hết các tổ chức, rất có khả năng có hai khách hàng trùng tên cho nên tên khách hàng không thể đáng được coi là khoá.

2. Trong bảng thực thể Khoản mục nhà cung cấp, bảng thực thể này, mô tả các mặt hàng khác nhau do nhà cung cấp bán, chứa thông tin cần cho việc cần xây dựng đơn hàng mua. Một trong các thuộc tính của bảng này là “Số hiệu mặt hàng của nhà cung cấp”, xác định một mặt hàng như một dòng trong danh mục của nhà cung cấp. Bản thân số này không phải là ứng cử viên cho khoá vì có thể hai nhà cung cấp khác nhau cùng hệ thống ghi số hiệu để mã hoá các danh mục trong các danh mục khác nhau. Trong thực tế cũng hiếm khi tìm được một khoá từ các thuộc tính có giá trị nằm ngoài sự kiểm soát của chúng ta (chẳng hạn trong tay các hãng bên ngoài).

Nói chung, ứng cử viên khoá duy nhất có thể được là tổ hợp của nhiều thuộc tính.

Chẳng hạn,

1. Trong kiểu thực thể khoản mục của nhà cung cấp, việc tổ hợp của “Số hiệu nhà cung cấp” và “Số hiệu mặt hàng của nhà cung cấp” sẽ có thể dùng được như một khoá.

2. Trong kiểu thực thể Khoản mục đơn hàng bán, tổ hợp của “Số hiệu đơn hàng bán” và “Số hiệu mặt hàng bán” có thể là khoá. Khi có sự lựa chọn giữa các khoá có thể có, quy tắc chung là chọn khoá nhỏ nhất và nên quan tâm tới các ứng cử viên số hơn là ứng cử viên chữ. Vì mô hình quan hệ là mô hình “logic” của dữ liệu có sẵn trong hệ thống nên không cần phải xem xét việc xử lý khi chọn khoá.

Nếu không có sẵn khoá duy nhất trong danh sách các thuộc tính, thì có thể đây không phải là bảng thực thể đích thực. Trong trường hợp tồi nhất, có thể cần phải thiết kế một khoá duy nhất cho kiểu thực thể. Các kỹ năng cần cho việc

thiết kế mã như vậy nằm ngoài phạm vi của tài liệu này.

Để minh hoạ quá trình chuẩn hoá, ta nên xét một ví dụ về một giao tác nghiệp vụ thông thường. Hình 36 minh hoạ một dạng đơn hàng đơn giản, mà có lẽ đã từng được nhà phân tích sử dụng để trợ giúp lên danh sách ban đầu các thuộc tính cho kiểu thực thể đơn hàng.

<b>TÀI LIỆU ĐƠN HÀNG BÁN</b>		
Số hiệu đơn.....		
Số hiệu khách hàng.....		Địa chỉ giao hàng
Tên khách hàng.....		
Ngày trên đơn.....		
Số hiệu mặt hàng	Mô tả mặt hàng	Số lượng

Hình 36. Một tài liệu đơn hàng chuẩn

<b>TÀI LIỆU ĐƠN HÀNG BÁN</b>		
Số hiệu đơn.....		
Số hiệu khách hàng.....		Địa chỉ giao hàng
Tên khách hàng.....		
Ngày trên đơn.....		
Số hiệu mặt hàng	Mô tả mặt hàng	Số lượng

TÀI LIỆU/ KIỂU THỰC THỂ		ĐƠN HÀNG BÁN.....		
Chưa chuẩn hoá	1NF	2NF	3NF	
Số hiệu đơn				
Số hiệu khách hàng				
Ngày đặt hàng				
Tên khách hàng				
Địa chỉ				
Số hiệu mặt hàng				
Mô tả mặt hàng				
Số lượng mặt hàng				

Hình 37. Khuôn dạng trợ giúp cho việc chuẩn hoá

Hình 37 mô tả cho cách bố trí khuôn dạng mà sẽ còn được dùng để hướng dẫn các nhà phân tích trong quá trình chuẩn hoá và chúng ta sẽ xem cách điền khuôn dạng đó trong phần tiếp của chương này. Ta có thể thấy rằng các thuộc tính được xác định từ dạng đơn này ở hình 37 đã được đưa vào ở “chưa chuẩn hoá”, và rằng “số hiệu đơn hàng” đã được xác định là khoá duy nhất.

TÀI LIỆU/ KIỂU THỰC THỂ		ĐƠN HÀNG BÁN.....	
Chưa chuẩn hoá	1NF	2NF	3NF
Số hiệu đơn	Số hiệu đơn	Số hiệu đơn	Số hiệu đơn
Số hiệu khách hàng	Số hiệu khách hàng	Số hiệu khách hàng	Số hiệu khách hàng
Ngày đặt hàng	Ngày đặt hàng	Ngày đặt hàng	Ngày đặt hàng
Tên khách hàng	Tên khách hàng	Tên khách hàng	Số hiệu khách hàng
Địa chỉ	Địa chỉ	Địa chỉ	Tên khách hàng
Số hiệu mặt hàng	Số hiệu mặt hàng		Địa chỉ
Mô tả mặt hàng	Tên hàng	Số hiệu đơn	Số hiệu đơn
Số lượng mặt hàng	Đơn vị	Số hiệu mặt hàng	Số hiệu mặt hàng
	Số lượng mặt hàng	Tên hàng	Số lượng mặt hàng
		Đơn vị	Số hiệu mặt hàng
		Số lượng mặt hàng	Tên hàng
			Đơn vị

Hình 38. Chuẩn hoá

Kết quả chuẩn hoá ta được các kiểu thực thể:

KHACH HANG : Số hiệu khách hàng, Tên khách hàng, Địa chỉ.

DON HANG : Số hiệu đơn hàng, Số hiệu khách hàng, Ngày đặt hàng.

DONG DON HANG: Số hiệu đơn hàng, Số hiệu mặt hàng, Số lượng mặt hàng.

HANG HOA : Số hiệu mặt hàng, Tên hàng, Đơn vị.

e. Gộp các kiểu thực thể chung

Thông thường khi chuẩn hoá một số danh sách kiểu thực thể, ta đưa tới cùng một kiểu thực thể.

Chẳng hạn, Kiểu thực thể khách hàng cũng đã được xác định từ việc xem xét chẳng hạn Phiếu đăng ký khách hàng.

Khi đó ta ở vào tình thế có hai kiểu thực thể giống nhau, cả hai đều ở dạng chuẩn 3NF nhưng mỗi kiểu thực thể có lẽ có một số thuộc tính khác nhau.

*f. Xác định những mối quan hệ*

Ở giai đoạn này, một mô hình quan hệ đầy đủ cần được phát triển. Bất cứ mô hình quan hệ nào cũng sẽ chứa trong nó những chi tiết không tương minh về mối quan hệ giữa các thực thể trong mô hình (dưới dạng các thuộc tính “kết nối”). Những quan hệ này có thể được lấy ra từ mô hình bằng một công cụ gọi là MA trận kiểu thực thể/ khoá.

Đây là một công cụ hữu ích để giúp cho phân tích việc xác định tất cả các mối quan hệ được chứa một cách tiềm ẩn trong mô hình quan hệ. Ta làm việc này bằng cách khảo sát lần lượt từng việc thực tế, và sử dụng các tên gọi duy nhất của nó để tìm kiếm đối với bất kỳ “thuộc tính kết nối” nào trong các kiểu thực thể khác. Nếu một bộ kết nối được tìm thấy, thì nó cho thấy sự có mặt của một quan hệ một - nhiều giữa hai kiểu thực thể liên quan. Khi đó một danh sách có thể được tạo ra bởi tất cả các mối quan hệ vốn có trong mô hình, và nhờ đó có thể xây dựng được một mô hình tương tự với Mô hình dữ liệu.

Ma trận kiểu thực thể/ khoá được xây dựng như sau:

	KHÁCH HÀNG	ĐƠN HÀNG	ĐONG ĐƠN HÀNG	HÀNG HOA
Số hiệu đơn hàng		K	K	
Số hiệu khách hàng	K	C		
Số hiệu mặt hàng			K	K

*Hình 39. Ma trận kiểu thực thể/khoá*

Mô hình được tạo ra bằng cách điền tên của các kiểu thực thể trong những cột phía trên, tên các thuộc tính khoá ở các dòng. Xác định cách chuyển danh sách của các kiểu thực thể vào các cột, và đánh dấu các tên của thuộc tính khoá khi chúng xuất hiện. Rõ ràng rằng cùng một thuộc tính khoá có thể xuất hiện đối với nhiều kiểu thực thể;

Những ô của ma trận sẽ được biểu diễn theo cách:

1. Chữ “K” được dùng biểu thị sự kết hợp của một kiểu thực thể và các thuộc tính khoá của nó được đưa vào những ô giao nhau giữa cột và những kiểu thực thể và hàng là một trong những thuộc tính khoá của nó: (ô giao nhau của cột)

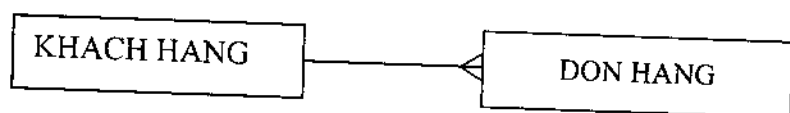
“đơn hàng” với hàng “số hiệu đơn hàng”

2. Chữ “C” được dùng để chỉ sự có mặt của một thuộc tính nổi trong một kiểu thực thể nhất định. Nhưng điều này được xác định bằng cách làm lần lượt với một kiểu thực thể, kiểm tra mô hình quan hệ về sự có mặt bất kỳ thuộc tính nào trong các hàng của ma trận.

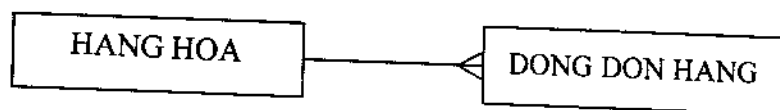
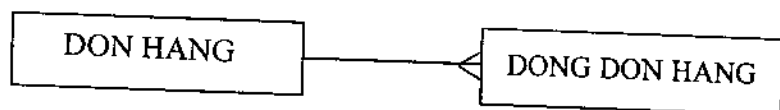
*Rút ra các mối quan hệ*

Phương pháp được dùng để rút ra các mối quan hệ từ ma trận thì không rắc rối chút nào cả. Làm lần lượt với từng kiểu thực thể, các ô “K” được xác định cho nó. Rồi nhà phân tích chỉ việc nhìn vào cột khác mà các thuộc tính có mặt trong đó, (bằng những chữ “K” hoặc “C”). Nếu một cột như vậy được tìm thấy, kiểu thực thể của nó được đánh dấu là xuất hiện tại đầu “nhiều” của mối quan hệ một - nhiều, với kiểu thực thể ban đầu xuất hiện tại đầu “một” (của quan hệ đó).

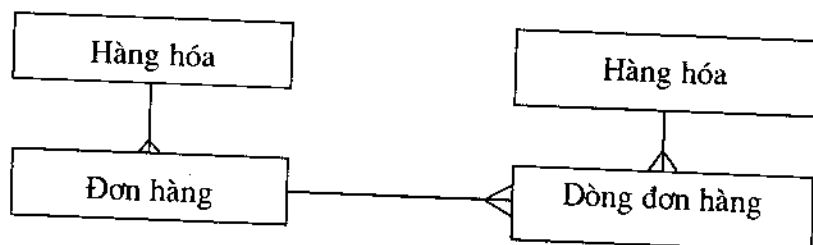
Trong ma trận ở hình 40, cột thứ nhất biểu diễn thực thể “khách hàng” và ở “K” chỉ thuộc tính khoá “số hiệu khách hàng” của nó. Xem xét các cột dọc theo “Số hiệu khách hàng” chúng ta sẽ nhận ra rằng cột “đơn hàng” có một giá trị tại ô của nó. Điều này cho thấy rằng kiểu thực thể “khách hàng” có mối quan hệ một - nhiều với kiểu thực thể “đơn hàng”.



Tương tự ta có:



Từ đó xác định mô hình:



### Bài tập chương 3

1. Khi xây dựng biểu đồ phân cấp chức năng ta dựa vào những yếu tố nào? Cần phải chú ý gì khi xây dựng biểu đồ phân cấp chức năng.
2. Cơ sở để xây dựng biểu đồ luồng dữ liệu các mức: khung cảnh, đỉnh, dưới đỉnh. Giữa DFD và FHD có mối liên hệ gì?
3. Biểu đồ luồng dữ liệu mức vật lý và mức logic khác nhau ở những điểm nào?
4. Ý nghĩa của các tác nhân ngoài đối với hệ thống. Có hệ thống nào không có tác nhân ngoài không? Tại sao?
5. Thế nào là quan hệ 1 - 1, quan hệ 1 - nhiều, quan hệ nhiều - nhiều. Cho các ví dụ minh họa.
6. Một quan hệ khi nào được gọi là có dạng chuẩn 1NF, dạng chuẩn 2NF, dạng chuẩn 3NF? Hãy đưa ra một quan hệ ở dạng chuẩn 1NF nhưng không ở dạng chuẩn 2NF, một quan hệ ở dạng chuẩn 2NF nhưng không ở dạng chuẩn 3NF.
7. So sánh phương pháp xây dựng mô hình dữ liệu và mô hình quan hệ.

## Chương 4

# THIẾT KẾ HỆ THỐNG

### Mục tiêu

Học sinh phải trình bày được các giai đoạn thiết kế.

Biết thiết kế các hệ thống con, các giao diện người máy.

Chỉ ra các điểm cần kiểm soát và thiết kế các kiểm soát tại các điểm đó.

Biết thiết kế CSDL. Nhận biết được độ phức tạp tính toán đối với các câu hỏi và biết cách bổ sung các trường cần thiết làm giảm độ phức tạp tính toán, đồng thời biết cách khắc phục sự không toàn vẹn dữ liệu tại các trường đó.

Biết thiết lập các lược đồ chương trình và đánh giá chất lượng một lược đồ.

### Nội dung tóm tắt

Thiết kế tổng thể.

Thiết kế giao diện.

Thiết kế cơ sở dữ liệu.

Thiết kế chương trình.

## NỘI DUNG

Thiết kế là nhằm chuyển các đặc tả logic của hệ thống (về chức năng, về dữ liệu) thành các đặc tả vật lý của hệ thống, có tính tới các yêu cầu và ràng buộc vật lý. Như vậy, nếu phân tích nhằm trả lời câu hỏi: “Làm gì?”, thì thiết kế nhằm trả lời câu hỏi: “Như thế nào?”

Đầu vào của công việc thiết kế bao gồm:

- Các đặc tả logic về hệ thống, có được từ giai đoạn phân tích.
- Các yêu cầu và ràng buộc về các điều kiện vật lý cụ thể, như là các hình trạng phần cứng, phần mềm, các tài nguyên, các yêu cầu về thời gian thực hiện,

thời gian trả lời, về xử lý sai lỗi, về chi phí bảo trì ....

Đầu ra của công việc thiết kế sẽ là các quyết định về:

- Một kiến trúc tổng thể của hệ thống.
- Các hình thức trao đổi trên biên của hệ thống (các mẫu thu thập, các tài liệu in ra, các giao diện người-máy).
- Các kiểm soát, nhằm phòng ngừa các sự cố vật lý và các ý đồ phá hoại.
- Tổ chức vật lý của cơ sở dữ liệu theo các phương án sử dụng tệp hay sử dụng hệ quản trị cơ sở dữ liệu có sẵn.
- Tổ chức chương trình theo modun.

Các quyết định về thiết kế luôn là một sự thỏa hiệp giữa hai mặt: tính hợp lý đơn thuần (logic) và sự hạn chế của điều kiện cụ thể (vật lý).

## **I. THIẾT KẾ TỔNG THỂ VÀ THIẾT KẾ GIAO DIỆN**

### **1. Thiết kế tổng thể**

#### **1.1. Mục đích**

Mục đích của thiết kế tổng thể là nhằm đưa ra một kiến trúc tổng thể của hệ thống. Kiến trúc này thể hiện sự phân chia hệ thống thành nhiều hệ thống con và sự chia tách phần thực hiện bằng thủ công với phần thực hiện bằng máy tính (một máy tính hay nhiều máy tính) trong mỗi hệ thống con đó.

#### **1.2. Phân chia hệ thống thành các hệ thống con**

*Hệ thống con* là một sự gom nhóm các chức năng trong một hệ thống xung quanh một nhiệm vụ hay một mục đích nào đó.

Sự phân chia hệ thống thành các hệ thống con là nhằm giảm thiểu sự phức tạp, công kênh, hoặc nhằm tạo ra những thuận lợi cho quá trình thiết kế cũng như khai thác, bảo dưỡng sau này.

Sự phân chia hệ thống thành các hệ thống con được tiến hành ngay trên biểu đồ luồng dữ liệu của hệ thống lập từ giai đoạn phân tích. Ta dùng một đường ranh (một đường đứt nét chẳng hạn) để tách các chức năng trong biểu đồ luồng dữ liệu thành nhóm, mỗi nhóm là một hệ thống con. Thông thường thì các DFD cấp cao (mức đỉnh hay mức dưới đỉnh) cho ta những gợi ý tốt: mỗi chức năng lớn xuất hiện trong các DFD mức cao, đó là đại diện cho một hệ thống con, gồm các chức năng phân rã từ nó trong các DFD mức thấp. Tuy nhiên sự phân chia dựa theo chức năng đó phải được xem xét dựa theo hai tiêu chuẩn:



- **Tính cố kết cao:** Cố kết là sự gắn bó về logic hay về mục đích của các chức năng trong cùng một hệ thống con. Sự cố kết của các chức năng trong cùng một hệ thống con phải càng cao càng tốt.

- **Tính tương liên yếu:** Tương liên là sự trao đổi thông tin và tác động lẫn nhau giữa các hệ thống con. Một sự phân chia tốt đòi hỏi sự tương liên lẫn nhau giữa các hệ thống con phải càng lỏng lẻo, càng đơn giản càng tốt.

Sự phân chia hệ thống thành hệ thống con thực ra không phải dựa hoàn toàn vào căn cứ chức năng thuần túy, mà còn có nhiều căn cứ khác phải tham khảo đến, đặc biệt là các căn cứ thực tế. Bởi vậy sẽ có nhiều cách gom nhóm mà ta có thể kể ra một số cách như sau:

(1) *Gom theo thực thể:* Gom tụ vào hệ thống con những chức năng liên quan tới cùng một (hay một số) kiểu thực thể nhất định.

*Ví dụ:* Hệ con Khách Hàng bao gồm mọi chức năng liên quan tới khách hàng, như các chức năng về xử lý đơn hàng, làm hoá đơn, phát hàng, thanh toán, xử lý nợ đọng vv... Hệ con KHO VẬT TƯ quy định các chức năng xuất, nhập hàng, đặt bổ sung, kiểm kê ...

(2) *Gom theo sự kiện giao dịch:* Gom tụ vào một hệ thống con những chức năng được khởi động (trong thời gian) khi có một sự kiện giao dịch nào đó xảy ra.

*Ví dụ:* Khi một đơn hàng đến (Một sự kiện giao dịch) thì một loạt các chức năng sẽ được khởi động, như Ghi nhận đơn hàng, Kiểm tra khả năng đáp ứng của kho hàng, kiểm tra tư cách khách hàng, Xử lý các yêu cầu của đơn hàng... Các chức năng này được gom vào một hệ thống con xử lý đơn hàng.

(3) *Gom theo trung tâm biến đổi:* Nếu phát hiện trong DFD có một nhóm các chức năng cộng tác với nhau để thực hiện một tính toán hay một sự biến đổi thông tin đặc biệt nào đó, thì tách chúng ra thành một hệ thống con.

(4) *Gom theo một lý do thiết thực nào đó:*

Đó có thể là những lý do nhằm:

- Thuận lợi cho cấu trúc kinh doanh;
- Thuận lợi cho vị trí của công sở;
- Thích ứng với cấu hình phần cứng vốn có;
- Phù hợp với trình độ đội ngũ cán bộ;
- Phù hợp với sự phân cấp trách nhiệm công tác;
- Tạo ra khả năng bảo mật tốt hơn...

### 1.3. Phân định phần thực hiện thủ công với phần thực hiện bằng máy tính

Cho đến nay, qua bước phân tích, ta đã mô tả hệ thống với nhiều chức năng, nhiều kho dữ liệu, nhưng ta chưa hề bao giờ đề cập đến các câu hỏi như chức năng nào sẽ do máy tính thực hiện, chức năng nào vẫn do con người thực hiện, kho dữ liệu nào được lưu trong máy tính (dưới dạng tệp hoặc cơ sở dữ liệu) và kho dữ liệu nào vẫn được quản lý bằng tay (dưới dạng hồ sơ, sổ sách).

Giờ đây đã đến lúc ta phải quyết định rõ về vai trò của máy tính trong hệ thống, bằng cách phân định trên biểu đồ luồng dữ liệu phần thực hiện bằng máy tính và phần thực hiện thủ công; đồng thời cũng quyết định phương thức xử lý theo lô được vận dụng ở những nơi nào và phương thức xử lý trực tuyến được vận dụng ở những nơi nào.

Sự phân định đó cũng được thực hiện trong DFD để quyết định chức năng nào sẽ thực hiện bằng máy tính, chức năng nào sẽ thực hiện bởi con người. Hai trường hợp xảy ra.

#### (1) Đối với các chức năng:

Xem xét từng chức năng trong DFD để quyết định chức năng nào sẽ thực hiện bằng máy tính, chức năng nào sẽ được thực hiện bằng con người. Hai trường hợp xảy ra:

- Một số chức năng có thể được quyết định chuyển trọn vẹn sang phần xử lý bằng máy tính hay phần xử lý thủ công. Tên của chúng sẽ vẫn được giữ nguyên.

- Một số chức năng xét thấy cần tách một phần xử lý bằng máy tính, một phần xử lý thủ công. Vậy phải phân rã thêm một mức để tách phần xử lý bằng máy tính với phần xử lý thủ công. Chọn tên thích hợp cho các chức năng con đó.

#### (2) Đối với các kho dữ liệu: Cũng lần lượt xem xét từng kho dữ liệu có mặt trên DFD.

- Kho dữ liệu chuyển sang phần máy tính sẽ là các kiểu thực thể tiếp tục có mặt trong mô hình dữ liệu, để sau này trở thành tệp hay cơ sở dữ liệu.

- Kho dữ liệu chuyển sang phần thủ công sẽ là:

- + Các tệp thủ công (sổ sách, bảng biểu...).

- + Các hồ sơ từ văn phòng.

- Các kiểu thực thể tương ứng với các kho dữ liệu xử lý thủ công phải loại ra khỏi mô hình dữ liệu (như vậy trong mô hình dữ liệu mới, một số kiểu thực thể biến mất, một số kiểu thực thể giảm thuộc tính). Ngược lại, do yêu cầu xử lý máy tính, một số kiểu thực thể mới có thể được thêm vào mô hình dữ liệu,

chẳng hạn Catalo cung cấp (hay kiểu thực thể NCC/mặt hàng) cần phải thêm vào mô hình dữ liệu, thay cho cách chào hàng trực tiếp của NCC.

- Chọn các phương án thể hiện khác nhau:

- Thường thì có nhiều cách sử dụng máy tính để thực hiện các chức năng. Đặc biệt có hai phương thức xử lý thông dụng là: Xử lý theo lô và xử lý trực tuyến. Vậy khi thành lập DFD đã chia tách thành phần máy tính và phần thủ công, ta phải thể hiện rõ xu hướng sử dụng máy tính đã được chọn lựa.

## **2. Thiết kế các nhiệm vụ thủ công và các giao diện người - máy**

Sau khi tách phần máy tính với phần thủ công, thì hai vấn đề thiết kế đặt ra ngay:

- Thiết kế các nhiệm vụ thủ công, tức là các quy trình do con người đảm trách;

- Thiết kế các giao lưu trên biên, nghĩa là các luồng dữ liệu thông qua đường ranh giới giữa hai phần máy tính và thủ công. Các giao lưu này được thể hiện thành các loại giao diện, như là các mẫu thông tin thu thập, các tài liệu in ra từ máy tính, các màn hình.

### **2.1. Thiết kế các chức năng thủ công**

*a) Gom các chức năng thủ công thành các công việc và nhiệm vụ*

Các chức năng nằm trong phần thủ công của DFD được đem ra nghiên cứu, mô tả lại chúng thành các quy trình, công việc. Thường các quy trình hay công việc đó là rất đơn giản, ta lại gom chúng thành các nhiệm vụ lớn hơn để giao cho một người hay một nhóm người thực hiện, một lần nữa việc gom nhóm lại có thể được quyết định theo nhiều tiêu chí khác nhau:

- Theo giao dịch;

- Theo kho dữ liệu (như thế người phụ trách công tác cũng phụ trách luôn cả dữ liệu);

- Theo địa điểm ;

- Theo thời gian xử lý ;

- Theo sự phân công chức trách ...

Nội dung và hình thức thực hiện các nhiệm vụ lại phụ thuộc vào phương thức làm việc giữa con người với máy tính, làm việc theo mẻ hay làm việc theo trực tuyến.

*b) Các yêu cầu đối với việc thiết kế các thủ tục thủ công*

Việc thiết kế các thủ tục thủ công phải đạt các yêu cầu sau:

- Miêu tả nội dung công việc rõ ràng: Mục đích cần đạt, các bước thực hiện, yêu cầu của mỗi bước.

- Ấn định độ chính xác phải đạt.

- Ấn định mức năng suất cần thiết, mức độ khéo léo cần có.

- Hướng dẫn mức xử lý khi có sai sót.

## **2.2. Thiết kế các biểu mẫu và tài liệu xuất**

### *a) Các biểu mẫu và các tài liệu in*

Đó là các hình thức trình bày các thông tin để nhập vào máy tính hay xuất từ máy tính, bao gồm:

Các biểu mẫu thu thập thông tin như là:

- + Các tờ khai.

- + Các phiếu điều tra.

Các tài liệu in ra từ máy tính như là:

- + Các bảng biểu thống kê, tổng hợp.

- + Các chứng từ giao dịch (đơn hàng, hoá đơn...).

### *b) Yêu cầu về thiết kế các biểu mẫu và tài liệu in*

- Phải bao gồm đầy đủ các thông tin cần thiết.

- Các thông tin phải chính xác, và do đó thông tin thu thập phải được kiểm tra trước khi đưa vào máy, thông tin xuất phải được kiểm tra trước khi giao cho nơi sử dụng.

- Phải dễ đọc, dễ hiểu, dễ sử dụng.

### *c) Cách trình bày các biểu mẫu và tài liệu in*

Nói chung mọi mẫu biểu hay tài liệu in đều gồm có 3 phần chính:

- Phần đầu: Gồm tên tài liệu, tên cơ quan chủ quản.

- Phần thân: Gồm các thông tin cần thu thập hay cần xuất.

- Phần cuối: Gồm ngày nhập tài liệu và chữ ký những người có trách nhiệm.

Ngoài ra, các phần khai phiếu điều tra thường có thêm phần ghi chú hay phân hướng dẫn cho người khai (thường để ở mặt sau tài liệu).

Các thông tin trình bày ở phần thân thường được gom theo nhóm có liên kết chặt chẽ với nhau. Các thông tin có cấu trúc (chẳng hạn một khoản đặt hàng, gồm mã mặt hàng, mô tả mặt hàng, số lượng, đơn giá, thành tiền) thường được trình bày thành các bảng gồm nhiều cột, nhiều dòng. Tên của các cột thường phải được cân nhắc sao cho vừa rõ nghĩa vừa không quá dài để có thể viết gọn được trong một ô đầu cột.

Thứ tự của các nhóm thông tin trình bày trong phần thân của tài liệu có thể được quyết định dựa trên nhiều căn cứ khác nhau, chẳng hạn:

- Theo thứ tự ưu tiên.
  - Theo thứ tự quen dùng hay dễ diễn.
  - Bình thường thì các bảng (thông tin có cấu trúc) để sau các thông tin đơn.
- Về các biểu mẫu thu thập thông tin, có ba cách lấy tin:
- Khung để điền.
  - Các trường hợp để chọn (như các mục 2, 9 trong tài liệu trên).
  - Các câu hỏi để trả lời. Có hai loại câu hỏi:

+ Câu hỏi đóng, là câu hỏi mà các phương án trả lời đã được dự kiến sẵn (thực chất cũng là các trường hợp để chọn).

+ Câu hỏi mở, là câu hỏi mà câu trả lời hoàn toàn tự do, không dự kiến trước được. Loại thông tin thu được bằng câu hỏi mở là khó xử lý bằng máy tính, vậy nên ít dùng.

### **2.3. Thiết kế màn hình và đơn chọn**

#### *a) Mục đích của việc sử dụng màn hình*

Màn hình (cùng với bàn phím) được sử dụng để đối thoại giữa người và máy.

Đặc điểm của sự đối thoại này là:

- Vào / ra gần nhau (có thể nói là xen kẽ).
- Thông tin cần đến là tối thiểu (có thể nói là cần đâu lấy đấy, không cần phòng dư hay tích lũy trước).

#### *b) Yêu cầu thiết kế màn hình*

- Màn hình phải sáng sủa, không lộn xộn;
- Thể hiện rõ cái gì là được yêu cầu, cái gì là cần làm.

#### *c) Các màn hình đối thoại*

Có ba hình thức thực hiện đối thoại giữa người/máy:

- Câu lệnh và câu nhắc: Máy hỏi hay nhắc, người đáp lại.
- Điền mẫu: Người dùng điền thông tin vào các ô trống trong một khung mẫu trên màn hình

- Đơn chọn: Người dùng chọn một việc (hay một khoản) trong nhiều việc (nhiều khoản) trình bày trong một đơn chọn (menu) trên màn hình.

Thay vì một danh sách các khoản trong đơn chọn, người ta có thể dùng một tập hợp các biểu tượng (icons) giàu hình ảnh và gây ấn tượng hơn.

Nếu các khoản trong đơn chọn là quá nhiều, người ta có thể dùng các đơn

chọn phân cấp (mở một khoản có thể mở ra một đơn chọn mới). Khi một đơn chọn có quá nhiều cấp để đi sâu thì việc quay lui hay ra khỏi đơn chọn phải chỉ dẫn cẩn thận.

*d) Các hướng dẫn cho việc thiết kế giao diện người/ máy*

Ngày nay các giao diện người/ máy thường được thực hiện trên các màn hình dưới dạng các cửa sổ (window). Hiện nay có khá nhiều hệ thống sản sinh các giao diện, giúp ta thực hiện khá thuận lợi các giao diện đó. Tuy nhiên các giao diện cần được thiết kế cẩn thận trước khi thực hiện nó.

Việc thiết kế giao diện người/ máy nên thực hiện theo các bước sau đây:

- Phân loại các người dùng theo:
  - Khả năng khéo léo, tinh tế;
  - Phân cấp trong tổ chức;
  - Các nhóm chuyên môn mà họ tham gia.
- Mô tả, nhu cầu, đặc điểm và kịch bản của mỗi loại người dùng.
- Thiết kế sự phân cấp các lệnh:
  - Đưa ra danh sách các lệnh theo nhu cầu của người dùng;
  - Chính đốn lại hệ thống các lệnh đó bằng cách :
- + Sắp xếp thứ tự:
  - Việc thường làm được đặt ra trước.
  - Sắp xếp theo trật tự công việc quen thuộc với người dùng.
- + Phân đoạn cho dễ nhìn:

Phân danh sách thành từng đoạn nhỏ, giúp cho người dùng có thể thu tóm ngay sau một cái liếc mắt.

- Thiết kế các chi tiết tương tác:

Dựa vào các tiêu chí sau:

- Thích hợp: Chọn ngôn từ thích hợp, số bước thích hợp, các hành động thích hợp.
- Ít bước: Tối thiểu hoá việc bấm chuột, gõ phím. Tối thiểu hoá thời gian chờ kết quả.
- Lập lại: Tạo khả năng lập lại (Undo) khi người dùng lầm lẫn điều gì đó.
- Người dùng không cần phải nhớ: Đừng để người dùng phải ghi nhớ điều gì (trong óc hay trên giấy) khi chuyển từ cửa sổ này sang cửa sổ khác.
- Dễ hiểu, dễ đọc: Tạo cảm giác thoải mái cho người dùng.

## **II. THIẾT KẾ CÁC KIỂM SOÁT**

### **1. Mục đích**

Mục đích của việc thiết kế các kiểm soát là đề xuất các biện pháp nhằm làm cho hệ thống đảm bảo được:

- Tính chính xác
- Tính an toàn
- Tính nghiêm mật
- Tính riêng tư

Tính chính xác của hệ thống thể hiện trước hết ở chỗ hệ thống làm việc luôn luôn đúng đắn, không đưa ra các kết quả tính toán sai lạc, không dẫn tới các kết quả kinh doanh sai lạc (chẳng hạn quyết định thực hiện giao hàng trong khi khách hàng đã có yêu cầu huỷ đơn hàng, mà giấy yêu cầu này lại đang tồn đọng đâu đó trong hệ thống). Các sai lầm đó có thể bắt đầu từ quá trình phân tích trước đây chưa thật kín kẽ, cũng có thể do lập trình sai lỗi. Các loại kiểm định (kiểm định đơn nguyên, kiểm định tích hợp, kiểm định hệ thống) có thể phát hiện các lỗi sai đó.

Tính chính xác còn thể hiện ở chỗ dữ liệu được sử dụng trong hệ thống là xác thực. Ở đây, ta sẽ đề cập các vấn đề kiểm tra các thông tin thu thập và các thông tin xuất từ hệ thống, nhằm đảm bảo tính xác thực của dữ liệu.

Tính an toàn của hệ thống thể hiện ở chỗ hệ thống không bị xâm hại (hay bị xâm hại không nhiều) khi có sự cố kỹ thuật.

Tính nghiêm mật của hệ thống thể hiện ở chỗ hệ thống có khả năng ngăn ngừa các xâm phạm vô tình hay cố ý.

Tính riêng tư của hệ thống thể hiện ở chỗ hệ thống đảm bảo được các quyền truy nhập riêng tư đối với các loại người dùng khác nhau.

### **2. Kiểm tra các thông tin thu thập và các thông tin xuất**

Để đảm bảo tính xác thực của hệ thống thông tin thu thập để đưa vào máy tính cũng như các thông tin xuất từ máy tính, nhất thiết phải thiết lập các biện pháp kiểm tra đối với các thông tin đó.

Nơi thực hiện kiểm tra thường là:

- Ở nơi thu thập thông tin đầu vào.
- Ở trung tâm máy tính.
- Ở nơi phân phối các thông tin đầu ra.

*Mục đích kiểm tra là:*

- Phát hiện lỗi.
- Chữa lỗi.

*Hình thức kiểm tra có thể được chọn lựa giữa nhiều phương án:*

- Kiểm tra tay hay kiểm tra máy (tự động).
- Kiểm tra đầy đủ hay kiểm tra không đầy đủ: Kiểm tra đầy đủ đương nhiên là tốt, song lại tốn kém, bởi vậy chỉ nên tập trung chủ yếu vào một số thông tin chủ yếu để kiểm tra.

- Kiểm tra trực tiếp hay kiểm tra gián tiếp;
- Kiểm tra trực tiếp là sự kiểm tra không cần dùng thông tin phụ. Chẳng hạn kiểm tra khuôn dạng của thông tin hay kiểm tra giá trị của thông tin có ở trong khoảng cho phép không, đều là các kiểm tra trực tiếp.

- Kiểm tra gián tiếp là kiểm tra qua việc so với các thông tin khác .

*Thứ tự kiểm tra thường là:*

- Kiểm tra trực tiếp trước,
- Kiểm tra gián tiếp sau.

Việc kiểm tra tự động (bằng chương trình máy tính) có thể cài đặt theo hai hình thức:

Chương trình kiểm tra khi nhập thông tin theo mẻ. Cách dùng này thường dùng cho các đợt thu thập thông tin với khối lượng lớn.

Chương trình kiểm tra thông tin nhập/xuất trong việc làm đối thoại người/máy. Chương trình loại này thường được cài đặt dưới dạng các trigger, tức là các chương trình tự kích hoạt bất cứ khi nào có sự loại bỏ, bổ sung, cập nhật dữ liệu.

### **3. Phân tích và thiết kế các kiểm soát**

*a) Các sự cố làm gián đoạn chương trình và sự phục hồi*

- Nguyên nhân, tác hại và biện pháp khắc phục sự gián đoạn chương trình. Khi một chương trình bị gián đoạn do một lý do nào đó, thì tác hại dẫn tới là:
  - Làm mất thì giờ, lại phải cho chạy chương trình lại từ đầu,
  - Làm mất hoặc sai lạc thông tin, chẳng hạn thông tin trên tệp bị sai lạc vì đang cập nhật dở dang.

Việc mất thời gian nói chung là không nghiêm trọng lắm, nhưng thông tin sai lệch là điều nguy hiểm, cần được khắc phục. Có nhiều cách để đảm bảo sự an toàn của thông tin.



- **Khoá từng phần cơ sở dữ liệu:** CSDL được phân hoạch thành các đơn vị để cập nhật. Các đơn vị có thể là trường, bản ghi, tệp hoặc một số phần rộng hơn của CSDL. Khi một bản sao của một đơn vị được cập nhật, thì bản gốc phải khoá lại, ngăn mọi truy nhập đến nó. Khi cập nhật kết thúc, phiên bản của đơn vị thay thế phiên bản cũ và sự cập nhập được hoàn thành. Nếu trong quá trình được cập nhập, hệ thống bị hỏng, thì bản gốc vẫn còn nguyên vẹn.

- **Các tệp sao lục:** Các tệp sao lục bao gồm các tệp nhật ký và các tệp lưu, Tệp nhật ký là những tệp tuần tự chứa các bản sao (hoặc hình ảnh) của các CSDL trước và sau khi chúng được cập nhập. Các tệp lưu gồm các bản sao toàn bộ CSDL được thực hiện mỗi tuần một lần.

- **Các thủ tục phục hồi:** Thủ tục phục hồi là thủ tục nhằm đưa CSDL trở về trạng thái đúng đắn mà trước khi nó bị hỏng vì một sự gián đoạn chương trình.

Sự gián đoạn chương trình có thể xảy ra do các nguyên nhân sau:

- **Hỏng hóc phần cứng:** Chẳng hạn hỏng một mạch ở trung tâm hay hỏng một thiết bị ngoại vi. Các hỏng hóc loại này nhiều khi không ngắt hẳn chương trình và để chương trình tiếp tục chạy sai lạc. Vậy phục hồi là vô ích, mà phải chữa lại máy và cho chạy chương trình từ đầu (có thể là một phiên bản dự trữ của một tệp đang dùng).

- **Giá mang của tệp có sự cố:** Chẳng hạn đĩa từ bị hỏng hay bị rạch. Chương trình bị gián đoạn vì thao tác vào ra không thực hiện được. Vậy có thể sử dụng thủ tục phục hồi.

- **Hỏng hóc môi trường máy tính:** Chẳng hạn mất điện, độ ẩm cao. Nếu điện mất hẳn, chương trình bị gián đoạn có thể sử dụng thủ tục phục hồi. Nhưng khi điện chỉ chập chờn hoặc máy bị sai lạc do nóng ẩm, chương trình bị ngắt hẳn. Phải chạy lại chương trình từ đầu, với một phiên bản dự trữ của tệp, mà không thể phục hồi.

- **Hỏng hóc hệ điều hành, hiếm thấy ở hệ điều hành một người dùng.** Chương trình bị ngưng toàn bộ và định vị được, vậy có thể sử dụng thủ tục phục hồi.

- **Thực hiện sai quy định của hệ điều hành:** Chẳng hạn trong hệ điều hành nhiều người dùng, xảy ra sự cố tràn bộ nhớ hay thời gian quy định, chương trình bị ngưng. Trường hợp này có thể sử dụng thủ tục phục hồi.

- **Lỗi do lập trình sai:** Chẳng hạn gặp trường hợp chưa dự kiến trong chương trình. Chương trình dừng nhưng định vị được. Vậy có thể phục hồi

thông tin (sau khi đã sửa lại chương trình).

- Nhầm lẫn thao tác: Sự phục hồi là có ích.

- Nguyên tắc hoạt động của các thủ tục phục hồi

Sao lục và phục hồi thường được các hệ thống nền (cứng và mềm) hỗ trợ, ở đây ta đề cập chúng ở mức độ người dùng và người lập trình ứng dụng.

Chương trình có khả năng phục hồi phải thực hiện được các công việc sau:

- Khi chạy bình thường, chương trình phải lưu trữ một cách định kỳ các giá trị của một số các “biến mốc”, dùng để trở tình trạng đang tiến triển của chương trình. Thường thì là chỉ số của các lệnh đang được thực hiện, các giá trị trung gian nhận được... Các giá trị này đương nhiên phải ghi trên một tệp ngoại vi mà không được để ở bộ nhớ trong để chúng còn lưu lại sau sự cố.

- Khi chạy phục hồi sau sự cố gián đoạn, chương trình thực hiện các bước sau:

- + Đọc các giá trị cuối cùng của các biến mốc.

- + Định vị đầu đọc của các tệp đang dùng.

- + Xử lý (nếu cần) một số lỗi trên các tệp vận động, do đó có một số lỗi đó chưa rõ là trước khi bị gián đoạn đã được xử lý chưa.

- + Khởi động lại chương trình từ chỗ bị ngắt.

*b) Các xâm phạm con người và cách phòng tránh*

- Sự xâm phạm từ con người

Con người cũng là một đầu mối gây tác hại đáng chú ý đối với hệ thống. Sự xâm phạm từ phía con người có thể là:

- Sự xâm hại vô tình, do nhầm lẫn hoặc bởi sự tò mò không có ác ý.

- Sự xâm hại cố tình của những kẻ có mục đích rõ rệt tấn công vào hệ thống nhằm lấy cắp thông tin, làm hệ thống tê liệt, phá hoại thông tin nhằm gây ra các quyết định sai lạc hoặc gây ra thất thoát, lãng phí tài sản.

Như vậy để bảo vệ hệ thống trước các sự xâm hại từ phía con người, ta phải nhằm vào mục đích sau:

- Bảo vệ tính bí mật: Bảo đảm thông tin không bị lộ hoặc bị khám phá bởi các cá nhân không được phép;

- Bảo vệ tính toàn vẹn: Bảo đảm tính nhất quán của dữ liệu, đặc biệt ngăn chặn việc tạo, thay đổi bất hợp pháp hoặc phá hoại dữ liệu.

- Bảo vệ tính khả dụng: Người dùng hợp pháp không bị từ chối truy nhập tới thông tin và các tài nguyên một cách không đúng.

- Bảo vệ tính riêng tư: Bảo đảm các tài nguyên không bị sử dụng bởi các cá nhân không có quyền hoặc theo các cách không hợp pháp.

- Các điểm hở và các đe dọa từ các điểm hở

Các điểm hở là các chỗ mà một tác nhân ngoài, vô tình hay cố ý có thể gây ra một tác động tiêu cực lên hệ thống gây thiệt hại cho cơ quan chủ quản hệ thống. Người phân tích, người kiểm soát, trước hết phải chỉ ra mọi điểm hở khả năng đối với hệ thống mà mình phải bảo vệ. Các điểm hở có thể là:

- Mọi luồng dữ liệu đi và đến một đối tác của hệ thống;
- Mọi luồng dữ liệu cắt ngang qua ranh giới giữa phần máy tính và phần thủ công trong biểu đồ luồng dữ liệu, các kho hoặc tệp;
- Ở mỗi điểm hở như vậy người phân tích cần phải dự đoán các mối đe dọa tiềm ẩn, thậm chí phải hình dung các kịch bản đe dọa có thể xảy ra. Từ đó có thể đánh giá mức độ đe dọa từ mỗi điểm hở là cao, vừa hay là thấp, như thế có thể tập trung sự chú ý nhiều hơn vào những điểm nhạy cảm nhất.

- Các biện pháp bảo mật

Có nhiều biện pháp có thể áp dụng để ngăn chặn các âm mưu làm hại đến hệ thống. Tuy nhiên mỗi biện pháp đó đều có hạn chế. Bởi vậy ta phải phối hợp đồng thời nhiều biện pháp thì mới có thể đạt hiệu quả mong muốn. Các biện pháp có thể sắp xếp theo các mức độ, từ đơn giản đến phức tạp:

- Bảo mật vật lý: Đó là các biện pháp sử dụng các công cụ vật lý, hoặc tác động lên thiết bị, thay vì các phần mềm. Để làm điều đó có thể dùng khoá hay hệ thống báo động để ngăn chặn và phát hiện các kẻ xâm nhập trái phép.

- Mật khẩu: Đó là các mã xưng danh, do người dùng tự đặt và đăng ký với hệ thống, nhưng giữ kín đối với người khác, để có thể xuất trình và chứng minh sự đích thực khi hệ thống kiểm tra. Mật khẩu là biện pháp phổ biến và đặc dụng. Song sự giấu kín mật khẩu, vì lý do này hay lý do khác, thường kéo dài, làm cho sự kiểm soát chóng mất tác dụng.

- Mật mã: Tạo mật mã là sự biến đổi dữ liệu từ dạng nhận thức được thành dạng không nhận thức được (gọi là mật mã). Quá trình ngược là sự giải mã. Có nhiều phương pháp tạo mật mã (với khoá bí mật, khoá công khai...) đem lại kết quả tốt. Tuy nhiên cách làm này thường khó bảo trì và tốn kém.

- Bảo mật bằng gọi lại: Sự truy nhập thực hiện một cách gián tiếp, qua mỗi trạm kiểm soát, tương tự cách gọi điện thoại qua tổng đài, thường được áp dụng ở các cơ quan có yêu cầu bảo mật cao. Kiểm soát được thật chặt song người

dùng phải chịu thêm phiền phức.

- Tường lửa: Đó là một hệ thống (phần cứng + phần mềm) được đặt giữa mạng của một tổ chức, một công ty, một quốc gia (intranet) và internet, hay giữa các mạng có độ bảo mật khác nhau. Chức năng của tường lửa là ngăn chặn các thâm nhập trái phép (theo danh sách truy nhập đã xác định trước) và thậm chí có thể lọc bỏ những gói tin mà ta không muốn gửi đi hay nhận vào vì một lý do nào đó. Tuy nhiên tường lửa không đủ thông minh như con người để có thể đọc hiểu từng loại thông tin và phân tích nội dung tốt hay xấu của nó.

- *Phân quyền người sử dụng*

Phân quyền người sử dụng là việc phân loại các người sử dụng để:

- Gán cho mỗi loại người sử dụng một số quyền truy nhập nhất định.
- Cho phép một số người sử dụng được phép uỷ quyền, tức là giao quyền truy nhập cho người khác.

### **III. THIẾT KẾ CƠ SỞ DỮ LIỆU**

#### **1. Mục đích**

Cơ sở dữ liệu là nơi lưu giữ lâu dài các dữ liệu của hệ thống ở bộ nhớ ngoài. Các dữ liệu này phải được tổ chức tốt theo hai tiêu chí :

- Hợp lí, nghĩa là phải đủ dùng không dư thừa ;
- Truy nhập thuận lợi, nghĩa là tìm kiếm, cập nhập, bổ sung và loại bỏ các thông tin sao cho nhanh chóng và tiện dùng .

Ở giai đoạn phân tích, ta đã nghiên cứu dữ liệu theo tiêu chí hợp lí (đủ và không thừa). Kết quả là đã thành lập được một lược đồ dữ liệu theo mô hình thực thể - liên kết hay mô hình quan hệ. Thường gọi đó là lược đồ khái niệm về dữ liệu, vì nó chỉ dừng ở yêu cầu đủ và không thừa, mà bỏ qua yêu cầu nhanh và tiện.

- Sang giai đoạn thiết kế, ta phải biến đổi lược đồ khái niệm nói trên thành lược đồ vật lý, tức là một cấu trúc lưu trữ thực sự của dữ liệu trong bộ nhớ ngoài. Có nhiều cấu trúc lưu trữ, mỗi cấu trúc đều có ưu điểm và nhược điểm riêng, buộc người thiết kế phải cân nhắc khi lựa chọn một cấu trúc thích hợp với hoàn cảnh của hệ thống mà mình xây dựng .Sự cân nhắc này dựa theo hai hướng nghiên cứu sau:

- Nghiên cứu các yêu cầu truy nhập dữ liệu của mỗi chức năng trong hệ thống, làm sao cho các truy nhập đó phải nhanh và tiện.

- Nghiên cứu các đặc điểm và ràng buộc của cấu hình vật lý của hệ

thống (các phần cứng và phần mềm sử dụng) sao cho thích ứng được với cấu hình đó.

Vì có hai việc phải nghiên cứu như vậy, mà người ta thường tách việc thiết kế dữ liệu thành hai bước:

(i) Thông qua việc nghiên cứu các yêu cầu truy nhập mà biến đổi lược đồ khái niệm thành một dạng trung gian gọi là lược đồ logic về dữ liệu.

(ii) Thông qua việc nghiên cứu cấu hình của hệ thống, đặt biệt là các ngôn ngữ lập trình, các hệ quản trị cơ sở dữ liệu đã được chọn dùng mà biến đổi lược đồ logic thành lược đồ vật lí thích hợp với cấu hình đó.

## **2. Thành lập lược đồ logic**

### **2.1. Lược đồ logic**

Mọi cấu trúc vật lý đều tạo nên từ các đơn vị cơ sở là các bản ghi (một bản ghi cấu tạo từ nhiều trường). Vì vậy người ta chọn lược đồ logic, một dạng trung gian trước khi đến lược đồ vật lí, là một cấu trúc các kiểu bản ghi. Mỗi kiểu bản ghi là một tập hợp những bản ghi có cấu trúc trường giống nhau, thường được gọi cho gọn là một bảng. Bảng được biểu diễn bởi một hình chữ nhật có mang tên của bảng (viết bên ngoài) và danh sách các trường (viết bên trong). Giữa hai bảng ta thiết lập một kết nối, diễn tả bằng một mũi tên, nếu hai bảng đó có một trường chung. Tên trường này được lấy làm nhãn cho kết nối, hơn nữa trường này phải có giá trị duy nhất cho mỗi bản ghi trong bảng ở gốc mũi tên.

Bước thiết kế logic xuất phát từ một lược đồ khái niệm về dữ liệu dưới dạng mô hình quan hệ đã chuẩn hoá (về 3NF). Dễ thấy rằng một mô hình thực thể - liên kết hạn chế là một cấu trúc các kiểu bản ghi, còn mỗi liên kết một - nhiều trở thành một kết nối. Cũng tương tự, một mô hình quan hệ cũng là một cấu trúc các kiểu bản ghi, trong đó mỗi quan hệ trở thành một kiểu bản ghi, còn các kết nối sẽ được thêm vào khi có mặt các khoá ngoài.

Tuy nhiên điều ngược lại thì không nhất thiết đúng: Một kiểu bản ghi không nhất thiết phải hàm nghĩa một kiểu thực thể, cũng như một kiểu bản ghi không nhất thiết tương ứng với một quan hệ đã ở dạng chuẩn 3. Mục đích thiết kế logic là, xuất phát từ lược đồ khái niệm ta đi tìm một cấu trúc các kiểu bản ghi phù hợp với các yêu cầu truy nhập của các tiến trình trong hệ thống. Chính vì vậy, bước thiết kế logic có xu hướng giạt lùi so với quá trình phân tích: Chấp nhận sự dư thừa, và có thể hạ chuẩn, miễn là sự lợi hại của bước giạt lùi này đã được cân nhắc rất thận trọng.

## **2.2. Đưa thêm các thuộc tính tính toán, tình thế và đánh giá các khối lượng**

Trước đây, trong giai đoạn phân tích hệ thống, ta đã đưa các thuộc tính tính toán, các thuộc tính tình thế ra khỏi lược đồ dữ liệu. Sở dĩ thế vì các thuộc tính này là dư thừa bởi giá trị của chúng là tính toán được từ các thuộc tính khác có mặt trong lược đồ. Sự có mặt của chúng trong lược đồ chứa chấp một nguy cơ tiềm ẩn vi phạm tính toàn vẹn của dữ liệu.

Tuy nhiên các thuộc tính này lại rất có ích và quen dùng trong quản lý. Chúng phản ánh trực tiếp một tình trạng nào đó của một khu vực quản lý (chẳng hạn số dư tài khoản, lượng hàng tồn kho ...), thường được tra cứu luôn, mà mỗi lần dùng thì không còn phải tính toán vòng vo nữa. Chính vì vậy mà ở giai đoạn thiết kế hệ thống này, khi ta phải quan tâm đến nhu cầu truy cập dữ liệu làm sao cho tiện nhanh, thì ta lại phải xem lại các thuộc tính tình thế, thuộc tính nào dùng nhiều thì được thêm vào các bảng. Thậm chí, nhiều thuộc tính tình thế cùng liên quan tới một chủ đề nào đó có thể được gom lại thành một bảng mới, gọi là bảng tình thế.

Một việc khác cần làm để chuẩn bị cho bước nghiên cứu các yêu cầu truy nhập tiếp theo là đánh giá số bản ghi cho mỗi bảng trong lược đồ. Số này được gọi là khối lượng của bảng và được ghi vào một ngăn bổ sung ở phía dưới bảng. Khối lượng của mỗi bảng thường là một số biến động theo thời gian. Bởi vậy ta cần tìm hiểu trong thực tế ứng dụng của hệ thống và lấy số trung bình các bản ghi của bảng trong một khoảng thời gian nào đó. Chẳng hạn trong một thư viện nhỏ, số lượng trung bình các bạn đọc là 100, số lượng trung bình các cuốn sách trong thư viện là 2000 và số lượng trung bình các lần cho mượn sách là 5000 (trong một năm). Những khối lượng này sẽ được bổ sung vào các bảng.

## **2.3. Nghiên cứu các yêu cầu truy nhập**

Khi nói rằng cơ sở dữ liệu phải được thiết kế sao cho có thể đáp ứng các yêu cầu truy nhập một cách nhanh chóng và tiện lợi, thì sự đáp ứng này không phải hướng tới bất kì yêu cầu truy nhập nào, mà chủ yếu hướng tới các yêu cầu truy nhập của các tiến trình thông tin có mặt trong hệ thống.

Bởi vậy ta phải lần lượt xét từng chức năng trong DFD của hệ thống, để tìm trong mỗi tiến trình đó có những yêu cầu truy nhập nào.

Một yêu cầu truy nhập, thể hiện một sự tìm kiếm cập nhật, bổ sung hay loại bỏ một số thông tin trong CSDL, thường gồm hai vế:

- (i) Biết một số thuộc tính nào đó,

(ii) Tra cứu một số trường khác .

Thông thường thì các trường biết trước gọi là khoá tìm kiếm và các trường tra cứu là không ở trên cùng một bảng, cho nên yêu cầu truy nhập được thực hiện bởi một đường truy nhập tạo nên một dãy các bước truy nhập.

Đối với mỗi bước truy nhập, ta cần chỉ ra bốn đặc điểm sau:

- Bảng cần được truy nhập,
- Khoá tìm kiếm,
- Trường cần tra cứu,
- Tần suất truy nhập.

Các nhận xét này cho phép ta chia cắt lại cấu trúc các kiểu bản ghi.

*Chia cắt lại các kiểu bản ghi*

Cấu trúc các kiểu bản ghi được chỉnh sửa lại cho phù hợp với các kết quả nghiên cứu về yêu cầu truy nhập ở trên :

- Các kết nối không hề được sử dụng (tức là nhãn của kết nối không được làm khoá tìm kiếm cho một bước truy nhập nào cả) thì sẽ được dỡ bỏ (cùng với khoá ngoài tương ứng).

- Căn cứ trên sự phát hiện các cụm trường thường được tra cứu đồng thời mà thực hiện các biến đổi sau:

- + Nếu có một cụm nằm rải ra trên hai bản, thì nên gộp hai bản đó thành một, để bớt số bước truy nhập.

- + Nếu nhiều cụm rời nhau lại nằm trên một bảng lớn, thì nên cắt bảng đó thành nhiều bảng nhỏ theo cụm để các bảng gọn nhẹ hơn.

- + Có thể lập lại một trường ở một bảng khác (tức là lập một bản sao của nó), nếu thấy như thế tiện tra cứu hơn.

Nếu thấy có một bảng nào đó được truy nhập nhiều theo một khoá tìm kiếm nào đấy, thì ta nên thiếp lập cho nó một đường truy nhập đặc biệt (lập tệp chỉ dẫn).

### **3. Thành lập lược đồ vật lý**

Lược đồ vật lý là cấu trúc lưu trữ thực sự của dữ liệu trong bộ nhớ ngoài, phụ thuộc theo cấu hình của hệ thống (các ngôn ngữ lập trình, các hệ quản trị cơ sở dữ liệu...). Có hai phương án lựa chọn chính là các tệp và các cơ sở dữ liệu, theo đó mà ta phải chuyển đổi lược đồ logic thu được từ trên thành lược đồ vật lý thích hợp.

Điểm khác biệt cơ bản giữa tệp và cơ sở dữ liệu là: Đối với tệp, người dùng

phải trực tiếp thành lập nó và khai thác nó, nghĩa là phải hiểu rõ và chịu trách nhiệm trực tiếp về nó. Như thế thì chương trình và dữ liệu ràng buộc chặt chẽ với nhau. Đối với CSDL thì sự có mặt của hệ quản trị CSDL đã cách ly chương trình của người dùng với cấu trúc lưu trữ của dữ liệu làm cho chương trình có ưu điểm là độc lập đối với dữ liệu. Đối lại thì các ngôn ngữ định nghĩa dữ liệu cũng như các ngôn ngữ thao tác dữ liệu cung cấp bởi các hệ QTCSDL lại phải nhúng được vào ngôn ngữ lập trình được chọn để cài đặt hệ thống, điều này không phải bao giờ cũng được thuận lợi.

### 3.1. Các tệp

Tệp là một dãy các bản ghi cùng kiểu. Mỗi bản ghi gồm nhiều trường và chiếm một số từ (hay byte) trong bộ nhớ.

Chương trình muốn tìm kiếm các bản ghi trong tệp thì phải đọc lần lượt các bản ghi từ đầu tệp cho đến khi gặp đủ các bản ghi cần tìm. Cách làm này được gọi là sự truy nhập tuần tự. Trường hợp cần tìm có một bản ghi thôi mà phải duyệt tuần tự như vậy thì rõ ràng là rất tốn thời gian, nhất là với các tệp lớn. Lúc đó tệp cần có khả năng truy nhập trực tiếp. Sự truy nhập trực tiếp được thực hiện nhờ cài thêm các tệp chỉ dẫn. Tệp chỉ dẫn có cùng số bản ghi như bản chính, song đó là tệp “nhẹ”, bởi mỗi bản ghi của nó chỉ có hai trường, một trường là khoá chỉ dẫn có giá trị trùng với một trường trong tệp chính được dùng làm khoá tìm kiếm cho yêu cầu truy nhập, và trường còn lại chứa con trỏ tới bản ghi tương ứng trong tệp chính. Nhờ có con trỏ lập sẵn đó mà với khoá chỉ dẫn đã biết ta có thể trực tiếp truy nhập vào bản ghi mà không cần tìm kiếm tuần tự.

Để chuyển đổi từ lược đồ logic về cấu trúc tệp, ta tiến hành như sau:

- Mỗi bảng trong lược đồ logic chuyển thành một tệp.
- Thêm các tệp chỉ dẫn đối với các khoá tìm kiếm có tần suất sử dụng cao.

Không thể lập tệp chỉ dẫn cho mọi khoá tìm kiếm được, bởi vì có một sự hạn chế là: hệ nào cũng có một số tối đa các tệp được mở tại cùng một thời điểm. Khi một tệp chính được mở sẽ kéo theo các tệp chỉ dẫn liên quan với nó cũng được mở, làm cho số tệp mở đồng thời có thể tăng vọt, vượt ra ngoài mức hạn chế, làm cho hệ thống bị tắc nghẽn. Vậy không thể lập chỉ dẫn tràn lan được, mà chỉ hạn chế vào những nhu cầu thực sự cấp thiết mà thôi.

### 3.2. Các cơ sở dữ liệu

Cấu trúc lưu trữ vật lý phụ thuộc theo mô hình của CSDL: Mô hình quan hệ, mô hình mạng hay mô hình phân cấp. Ở đây ta chỉ xét mô hình quan hệ. Với



mô hình này mỗi bảng trong lược đồ logic chuyển đổi trực tiếp thành một quan hệ (mỗi trường thành một thuộc tính của quan hệ). Không có cài đặt gì đặc biệt đối với các kết nối giữa các bảng, ngoại trừ sự có mặt (vốn có) của khoá ngoài. Một ngôn ngữ định nghĩa dữ liệu cho phép định nghĩa các quan hệ, và các lệnh trong một ngôn ngữ thao tác dữ liệu cho phép lưu trữ và tìm kiếm các dữ liệu dựa trên các phép toán của đại số quan hệ (thay cho sự lần dò theo các con trỏ).

## **IV. THIẾT KẾ CHƯƠNG TRÌNH**

### **1. Mục đích**

Tới đây thì các kết quả thu được qua các giai đoạn phân tích, thiết kế tổng thể, và thiết kế chi tiết (về các giao diện, kiểm soát và cơ sở dữ liệu) dù là khá phong phú, nhưng vẫn còn là chưa đủ để có thể chuyển sang lập trình được. Các yếu tố còn thiếu là:

- Các chức năng xuất hiện trong các DFD chỉ là các chức năng logic (thuộc lĩnh vực bài toán) mà chưa có các chức năng phụ trợ cần thiết, như là:

- + Các chức năng đối thoại với người dùng.

- + Xử lý lỗi.

- + Xử lý vào/ra.

- + Tra cứu CSDL.

- + Các chức năng điều hành (nhằm liên kết các chức năng khác).

- Các liên quan giữa các chức năng trong DFD chỉ là các chuyển giao dữ liệu mà không phải là các chuyển giao điều khiển (tức là chuyển giao sự thực hiện khi thi hành). Thế mà một đặc trưng không thể thiếu trong một chương trình là đặc trưng điều khiển (sự tuần tự, chọn, lặp và đặc biệt là lời gọi giữa các chương trình con). Đặc trưng này chưa hề có trong các DFD.

Vì các thiếu sót này mà các DFD thu được từ giai đoạn phân tích còn phải được biến đổi, bổ sung thêm chi tiết, thì mới trở thành đầu vào thực sự cho việc lập trình được. Vì vậy phải có thêm một giai đoạn thiết kế chi tiết, đó là thiết kế chương trình. Chú ý rằng đây cũng chỉ là một giai đoạn của thiết kế, nhằm đưa ra các quyết định về cài đặt, chứ chưa phải là cài đặt, chưa phải là lập trình thực sự.

Đầu vào cho việc thiết kế chương trình là DFD của từng hệ thống con (đã xác định trong thiết kế tổng thể) cùng với các quyết định về giao diện, kiểm soát và CSDL đã được chọn trong các bước thiết kế chi tiết trước đây.

Đầu ra của thiết kế chương trình là một miêu tả về nội dung các chương

trình sẽ được cài đặt, bao gồm:

+ Một lược đồ chương trình (LCT) cho mỗi hệ thống con. Lược đồ chương trình được trình bày dưới dạng một đồ thị có hướng thường gọi là lược đồ cấu trúc, trong đó:

- Mỗi nút là một môđun chương trình,
- Mỗi cung là một lời gọi (lời gọi của môđun ở gốc đối với môđun ở ngọn của cung).

+ Đặc tả nội dung của từng môđun trong LCT.

+ Phân bổ các môđun trong LCT thành các chương trình (hay môđun tải).

## 2. Lược đồ cấu trúc

*Lược đồ cấu trúc* (còn gọi là *lược đồ chương trình*) là một biểu diễn dưới dạng đồ thị của một tập hợp các môđun cùng với các giao diện giữa các môđun đó (bao gồm sự chuyển giao điều khiển và chuyển giao dữ liệu).

### 2.1. Các môđun chương trình

Trong định nghĩa của lược đồ cấu trúc nói trên, thì môđun có thể được hiểu là:

Một chương trình con.

Có khi chỉ là một cụm câu lệnh nằm trong chương trình.

Nhiều ngôn ngữ lập trình có các UNIT, CLASS, OBJECT. Đó thực chất là các nhóm môđun chương trình (thường gọi là các phương thức) tập hợp xung quanh một cấu trúc dữ liệu.

Nói chung thì môđun có bốn thuộc tính cơ bản sau:

- Vào/Ra:

+ Cái vào: Nó nhận được thông tin gì từ chương trình gọi nó.

+ Cái ra: Nó trả lại thông tin gì cho chương trình gọi nó.

- Chức năng: Ấy là ánh xạ từ cái vào thành cái ra.

- Cơ chế: Ấy là phương thức cụ thể để biến đổi cái vào thành cái ra.

- Dữ liệu cục bộ: Ấy là các chỗ nhớ hay cấu trúc dữ liệu dùng riêng cho nó.

Vào/ra + Chức năng tạo thành đặc trưng bề ngoài của một môđun. Còn cơ chế + Dữ liệu cục bộ tạo thành đặc trưng bên trong của một môđun. Phương pháp thiết kế có cấu trúc lưu ý trước hết đặc trưng bề ngoài của môđun và lùi việc xem xét các đặc trưng bên trong của môđun lại sau.

Ngoài bốn thuộc tính cơ bản trên, môđun chương trình còn có một số thuộc tính khác cũng cần lưu ý khi thiết kế:

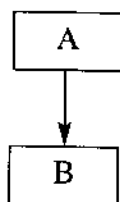
- Tên của môđun, cho phép dùng tên để gọi môđun (tức là huy động nó).
- Chỗ chiếm của nó (ở trong văn bản chương trình, hay trong bộ nhớ).

## 2.2. Các yếu tố hợp thành LCT

(i) Các môđun: Biểu diễn bởi một hộp chữ nhật với tên môđun ở bên trong. Tên môđun phản ánh tóm tắt chức năng của môđun.

Nếu là các môđun đã định nghĩa sẵn (trong hệ thống, trong thư viện chương trình), thì các cạnh bên được vẽ kép.

(ii) Kết nối các môđun: Các môđun có thể được kết nối với nhau bằng các lời gọi, diễn tả bởi một mũi tên (cung)



Môđun A gọi môđun B;

Môđun B thực hiện chức năng của mình rồi trả điều khiển cho A ở vị trí sau lời gọi.

Trong cách biểu diễn các lời gọi bằng các cung giữa các môđun như trên thì ta có thể hiểu thêm như sau:

- Vị trí của lời gọi (trong môđun A), cũng như số lần gọi đều không được chỉ rõ. Điều đó có nghĩa là ở giai đoạn lập LCT thì các chi tiết này có thể tạm bỏ qua, để giảm bớt sự phức tạp.

- Trật tự trước sau của nhiều lời gọi xuất phát từ cùng một môđun được thể hiện bởi trật tự từ trái qua phải.

Như vậy trong dãy các môđun được gọi bởi cùng một môđun thường phân biệt ba khu vực: bên trái là khu vực gồm các môđun lấy các thông tin vào, ở giữa là khu vực gồm các môđun xử lý các thông tin, và bên phải là khu vực gồm các môđun đưa các thông tin ra.

- Đôi khi người ta thể hiện phép chọn giữa một số lời gọi bằng một hình thoi nhỏ, và thể hiện phép lặp một nhóm lời gọi bằng một mũi tên vòng. Chẳng hạn ở hình dưới đây, ta hiểu là môđun A:

Gọi môđun B hoặc môđun C (tùy thuộc một điều kiện nào đó).

Và tiếp đó thực hiện vòng lặp các lời gọi D, E và thỉnh thoảng F.

(iii) Thông tin trao đổi giữa các môđun:

Các thông tin được gửi đi kèm với lời gọi (các tham số) và thông tin trả về sau khi thực hiện lời gọi (trả lời) được thể hiện bằng các mũi tên nhỏ vẽ dọc

theo cung biểu diễn cho lời gọi, có kèm theo tên của thông tin.

Dưới đây là một ví dụ về một lược đồ cấu trúc (LCT)

### **3. Cách chuyển đổi DFD thành LCT**

#### **3.1. Yêu cầu chung**

Đối với mỗi DFD của một hệ thống con, ta phải lập một LCT tương ứng. LCT này phải đạt các yêu cầu sau:

- Nhiệm vụ của mọi tiến trình trong DFD phải được chuyển hết vào các môđun chương trình của LCT. Đó có thể là sự chuyển đổi mỗi tiến trình thành một môđun chương trình, mà cũng có thể là sự phân tán hay tổ hợp các tiến trình vào các môđun chương trình khác nhau.

- Phải thêm các môđun vào - ra (giao diện với người dùng hay truy nhập CSDL), và đặc biệt là thêm các môđun điều khiển làm nhiệm vụ dẫn dắt quá trình xử lý. Trong số các môđun điều khiển đó có một môđun đặt ở gốc của LCT, thường được gọi là môđun chính (main).

- Thiết lập các lời gọi (kèm với các thông tin chuyển giao) giữa các môđun, phản ánh được quá trình thực thi của chương trình.

Trước khi xem xét các cách thức chuyển đổi DFD thành LCT, ta hãy xét một ví dụ cụ thể. Hình dưới đây cho một DFD nhỏ (đó là một phần của hệ thống con 2.2 của hệ QLCUVT đã cho) và LCT được chuyển đổi từ DFD đó. Qua đây ta thấy sự khác biệt về cấu trúc của hai loại công cụ biểu diễn này.

#### **3.2. Triển khai trên xuống**

Một LCT với các môđun và các lời gọi của nó tạo nên một sự phân cấp. Bởi vậy cách thành lập LCT tốt nhất là triển khai dần từ trên xuống. Sự triển khai trên xuống đó lại có thể kết hợp chặt chẽ với sự phân mức các DFD mà ta thu được từ giai đoạn phân tích. Chẳng hạn DFD mức đỉnh có dạng như ở hình dưới, trong đó chức năng B triển khai thêm một mức trở thành một DFD mức dưới đỉnh với các chức năng B1, B2, B3, B4. Căn cứ trên DFD mức đỉnh (ở đây có dạng tuyến tính), ta dễ dàng lập được phần chót của LCT (xem hình tiếp theo), rồi sau đó triển khai B với các lời gọi tới các môđun mới B1, B2, B3, B4.

Cách làm như trên là khá đơn giản, song LCT thu được thường là quá rườm rà. Và lại, khi gặp một DFD (ở một mức nào đó) có dạng phức tạp, thì việc chuyển nó thành LCT không phải là dễ như ở ví dụ trên. Các trường hợp phức tạp thường thể hiện dưới dạng điển hình: cấu trúc chế biến và cấu trúc giao dịch. Do đó có hai hướng thiết kế khác nhau sẽ được trình bày dưới đây.

### *a) Thiết kế hướng theo chế biến*

Thiết kế hướng theo chế biến áp dụng cho trường hợp DFD có nhiệm vụ chế biến một số thông tin lấy từ một số nguồn phát, thành một số thông tin gửi tới nơi nhận.

Các bước thực hiện như sau:

(i) Trước hết xác định nhóm các chức năng góp phần thực sự và chủ yếu vào nhiệm vụ chế biến thông tin của DFD. Gọi nhóm này là trung tâm chế biến. Phần còn lại của biểu đồ gồm:

- Các tuyến lấy thông tin vào, mỗi tuyến vào này xuất phát từ một nguồn phát, dẫn thông tin qua một số bước sơ chế (nghĩa là những xử lý "nhẹ" như là biên tập, sắp xếp,...) để cuối cùng trở thành một đầu vào của trung tâm chế biến.
- Các tuyến chuyển thông tin ra, mỗi tuyến này xuất phát từ một đầu ra của trung tâm chế biến, đưa thông tin đi qua một số bước gia công (nghĩa là những xử lý nhằm chỉnh sửa chút ít, như là chuyển thành dạng dễ hiểu, dễ dùng hơn) để đến một nơi nhận.

(ii) Vẽ hai mức cao nhất của LCT, trong đó môđun đỉnh gọi đến:

- + Một môđun lấy thông tin cho mỗi tuyến vào,
- + Một môđun tạo thông tin cho trung tâm chế biến,
- + Một môđun chuyển giao thông tin cho mỗi tuyến ra.

Kết quả thực hiện bước này đối với DFD cho trong hình trên được trình bày trong hình dưới đây.

### *Hai mức đầu của LCT chế biến*

(iii) Triển khai mỗi môđun ở mức 2 (vào, ra hay chế biến) thành các môđun ở các mức thấp, cho đến khi chuyển hết mọi nhiệm vụ xử lý của DFD vào LCT. Kết quả cuối cùng của ví dụ trên là LCT ở hình sau:

### *b) Thiết kế hướng theo giao dịch*

Đặc trưng của cấu trúc giao dịch là có một chức năng phân loại cho phép xác định loại của dữ liệu vào, để rồi cứ mỗi loại sẽ cung cấp một cách xử lý (hay dịch vụ) riêng. Thiết kế hướng theo giao dịch áp dụng cho trường hợp DFD thể hiện một cấu trúc giao dịch như vậy.

Các bước thực hiện như sau:

(i) Trước hết xác định trung tâm giao dịch trong DFD, bao gồm chức năng phân loại dữ liệu vào cùng với các chức năng tham gia vào các quá trình xử lý cho từng loại dữ liệu. Phần còn lại của biểu đồ gồm:

- Một tuyến lấy thông tin vào, dẫn thông tin từ một nguồn phát, qua một số chức năng sơ chế để đến chức năng phân loại thuộc trung tâm giao dịch.

- Một (hay một số) tuyến đưa thông tin ra, dẫn thông tin đưa ra từ các xử lý theo trường hợp, qua một số chức năng gia công thêm để tới một (hay một số) nơi nhận.

(ii) Vẽ hai mức cao nhất của LCT, trong đó môđun đỉnh gọi đến:

- Một môđun lấy thông tin vào tuyến vào,

- Một môđun xử lý (dịch vụ) cho mỗi trường hợp của dữ liệu vào; các lời gọi này được kết nối qua một phép chọn,

- Một môđun chuyển giao thông tin cho mỗi tuyến ra.

Kết quả thực hiện bước này cho DFD được trình bày trên.

(iii) Triển khai mỗi môđun ở mức 2 ( vào, ra hay dịch vụ) thành các môđun ở các mức thấp, cho đến khi chuyển hết mọi nhiệm vụ xử lý của DFD vào LCT.

#### **4. Chất lượng của LCT**

LCT lập được từ các cách làm trình bày ở trên chưa nên xem là dạng cuối cùng để chấp nhận. Trái lại ta chỉ có thể xem đây là phác thảo bước đầu của thiết kế môđun. Ta còn phải tinh chỉnh nó lại, bằng cách gộp, tách hay san sẻ lại nhiệm vụ giữa các môđun để đạt được các tiêu chuẩn về chất lượng nói ở sau đây.

##### **4.1. Sự tương liên**

Sự tương liên nói lên mức độ ảnh hưởng lẫn nhau giữa các môđun. Đương nhiên không thể không có tương liên giữa các môđun trong cùng một LCT. Tuy nhiên để có một LCT tốt, sự tương liên giữa các môđun phải càng lỏng lẻo, càng đơn giản càng tốt. Có nhiều loại tương liên, ta hãy xét vài loại chính:

**Tương liên nội dung:** Đó là trường hợp môđun này can thiệp nội dung của môđun khác. Ví dụ một môđun làm thay đổi nội dung của môđun khác, rẽ nhánh sang một môđun khác, hay sử dụng dữ liệu cục bộ (không phải dữ liệu phản hồi) của môđun được gọi. Tương liên nội dung là tương liên xấu nhất, cần phải loại bỏ.

- **Tương liên điều khiển:** Đó là trường hợp một môđun này chuyển một thông tin điều khiển cho một môđun khác. Thông tin điều khiển là thông tin dùng vào việc điều khiển quá trình, chứ không là chất liệu của các xử lý. Chẳng hạn các cờ, trạng thái, giá trị Boolean dùng để tính điều kiện của một phép chọn v.v... Tương liên điều khiển vi phạm nguyên tắc che giấu thông tin. Bởi vì khi một môđun chuyển một thông tin điều khiển cho một môđun bị gọi, thì như vậy nó đã phải biết phương thức bên trong của môđun bị gọi. Một thay đổi với môđun

bị gọi sẽ kéo theo sự thay đổi đối với môđun gọi. Vậy tương liên điều khiển cũng nên tránh. Tuy nhiên ta không thể tránh tương liên điều khiển một cách triệt để được, vì nhiều khi nó vẫn có ích, vẫn cần thiết.

- *Tương liên dữ liệu*: Đó là trường hợp hai môđun trao đổi dữ liệu cho nhau. Đó là loại tương liên cần phải được chấp nhận. Song sự trao đổi dữ liệu cũng phải càng đơn giản càng tốt, nghĩa là:

- + Ít dữ liệu tốt hơn là nhiều dữ liệu trao đổi;

- + Nếu có cách trao đổi chuẩn (chẳng hạn chuyển giao tham số) thì nên dùng nó;

- + Nên chuyển dữ liệu thật, hơn là chuyển con trỏ đến dữ liệu.

#### 4.2. Sự cố kết

Sự cố kết nói lên sự gắn bó giữa các phần bên trong một môđun. Về phương diện này, thì một môđun nên thực hiện một chức năng logic duy nhất hơn là nhiều chức năng khác nhau. Môđun càng cố kết, thì chức năng của nó càng dễ thấy, logic của nó rành mạch, do đó dễ phát hiện lỗi, dễ bảo trì.

#### 4.3. Hình thái

Hình thái của lược đồ cũng là một biểu hiện của chất lượng. Khi đi từ một mức trên xuống mức dưới, thì các nhánh của LCT có thể xoè ra (rẽ thành nhiều nhánh) hay chụm vào (chập vào cùng một môđun). Nhìn một cách toàn thể thì LCT nên xoè ở trên và chụm lại ở dưới. Thông thường thì trong một LCT, các môđun càng ở phía trên thì các chức năng của chúng càng nặng về điều khiển và càng nhẹ về xử lý. Trái lại các môđun càng ở thấp phía dưới thì càng ít tính chất điều khiển và càng nặng về xử lý hay dịch vụ. Bởi vậy càng xuất hiện nhiều các điểm chụm ở phía dưới, có nghĩa là có nhiều dịch vụ được sử dụng đi sử dụng lại nhiều lần, ở nhiều chỗ. Sử dụng lại là một trong những mục tiêu của thiết kế môđun, thể hiện trong LCT ở sự chụm lại đó.

Ta gọi phạm vi điều khiển của một môđun là phần LCT bao gồm môđun đó và những môđun phụ thuộc (được gọi) trực tiếp hay gián tiếp từ nó. Ta gọi phạm vi ảnh hưởng của một quyết định là phần LCT bao gồm mọi môđun chịu ảnh hưởng của quyết định đó. Chẳng hạn trong LCT cho trong hình dưới thì:

- Phạm vi điều khiển của A là A, B, C;

- Chẳng hạn trong B có một quyết định q1 và kết quả của quyết định này được dùng trong A, E, F, thì phạm vi ảnh hưởng của q1 là A, E, F.

Một thiết kế tốt phải tạo ra LCT, trong đó:

- Các quyết định phải có miền ảnh hưởng càng hẹp càng tốt;
- Mỗi phạm vi ảnh hưởng phải nằm trong phạm vi điều khiển tương ứng.

## 5. Đặc tả các môđun

Sau khi lập được LCT (cho mỗi hệ thống con), thì ta còn phải đặc tả mỗi môđun trong đó, tức là miêu tả rõ nội dung của môđun.

Công việc này cũng giống như trước đây, khi ta cần phải đặc tả tiến trình khi đã thành lập DFD. Các phương tiện biểu diễn dùng cho đặc tả môđun vẫn là các phương tiện dùng cho đặc tả chức năng (đặc biệt là các sơ đồ khối hay các ngôn ngữ tựa NNLT). Tuy nhiên điểm khác biệt cơ bản là ở mức độ của sự miêu tả. Đặc tả chức năng của DFD chỉ dừng lại ở mức độ miêu tả logic của xử lý, bỏ qua mọi điều kiện cài đặt cụ thể. Còn đặc tả môđun chương trình thì phải cụ thể hơn, bao gồm cả các chi tiết như là: các tham số chuyển giao, các đối thoại với người dùng, các xử lý lỗi, thực hiện vào/ra, tra cứu cơ sở dữ liệu. Vậy ở đây ta phải vận dụng các kết quả thiết kế trước đó, như là thiết kế các thủ tục thủ công, các giao diện, các tệp hay CSDL.

## 6. Đóng gói thành môđun tải

Một môđun chương trình muốn thực hiện được phải đưa vào bộ nhớ trong (gọi là được nạp, hay được tải). Mỗi lần tải phải tốn một phần thời gian của hệ thống.

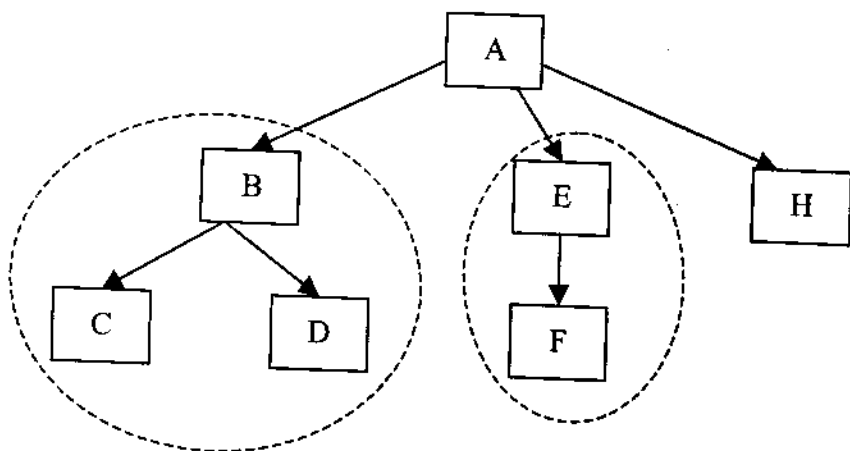
Người ta gọi môđun tải (load modul) là một nhóm môđun chương trình được tải vào bộ nhớ trong đồng thời. Lý tưởng nhất là tất cả các môđun trong một LCT hợp thành một môđun tải. Như thế thì thời gian tiêu tốn cho việc tải chương trình là ít nhất, và chương trình chạy một cách thuận lợi, vì gọi môđun nào là môđun đó đã sẵn sàng. Tuy nhiên như vậy đòi hỏi bộ nhớ lớn, nhiều khi không đáp ứng nổi. Ngược lại, nếu mỗi môđun chương trình là một môđun tải, thì tiết kiệm được chỗ nhớ, nhưng lại tốn nhiều thời gian. Vì vậy cần tìm giải pháp trung gian cho môđun tải. Bởi vậy nhiệm vụ cuối cùng trong thiết kế chương trình là cắt LCT thành các môđun tải.

Nói chung thì việc thiết kế môđun tải phải căn cứ trên các yếu tố như là kích cỡ bộ nhớ, kích cỡ các môđun, tần suất các cuộc gọi môđun và một môđun tải phải bao gồm các môđun gắn kết với nhau nhiều nhất.

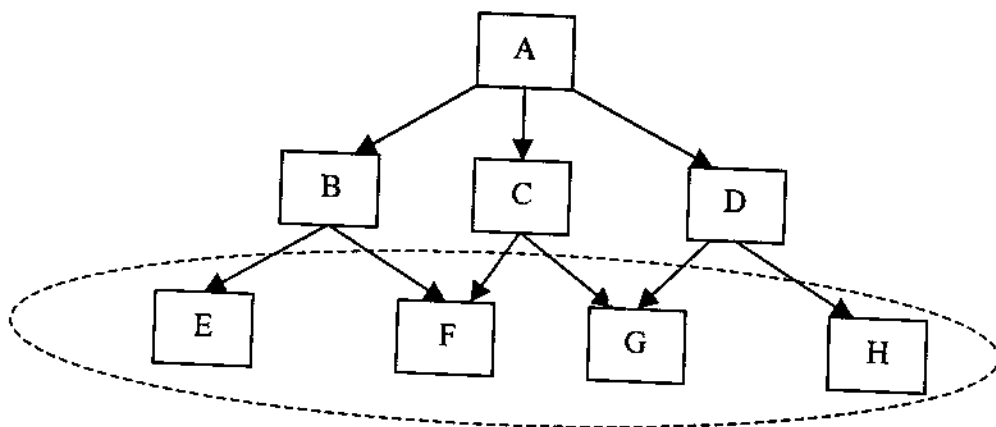
Tuy nhiên không có một phương pháp rõ rệt cho việc thiết kế môđun tải. Các hình (a), (b), (c) dưới đây cho một số cách làm, cũng chỉ mang tính gợi ý.



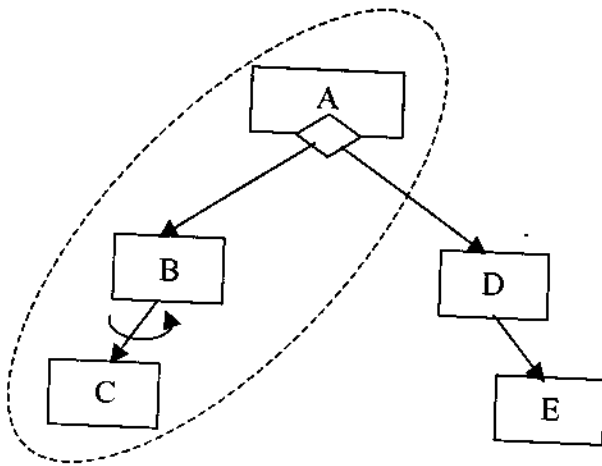
Hình (a) gom các môđun theo phạm vi điều khiển (tức là theo dòng các lời gọi). Hình (b) lại gom mọi môđun ở mức thấp nhất vào một môđun tải. Đương nhiên cách làm này chỉ có ích khi các môđun ở mức thấp này là những môđun dùng chung và dùng luôn cho các môđun ở những mức trên, chúng cần phải thường trực ở bộ nhớ trong. Hình (c) lại đi sâu hơn vào các cấu trúc chọn hay lặp. Ở đây A thường xuyên chọn B và thỉnh thoảng mới chọn D, mà B lại gọi lặp C nhiều lần. Vậy gom A, B, C vào một môđun tải (nếu bộ nhớ là eo hẹp).



(a) Gom môđun tải theo dòng điều khiển



(b) Gom môđun tải theo mức thấp nhất



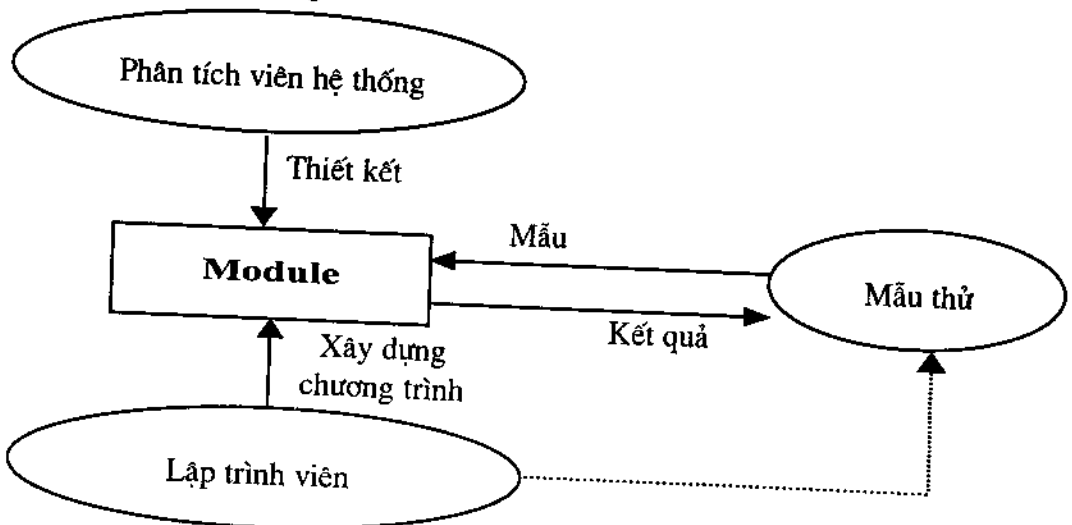
(c) Gom môđun tải theo tần số gọi

## 7. Thiết kế các mẫu thử

### 7.1. 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.

- 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”

## 7.2. Các loại mẫu thử

*Loại mẫu thử hoàn chỉnh/ không hoàn chỉnh*

Mẫu thử hoàn chỉnh đảm bảo dự kiến mọi trường hợp có mặt trong chương trình. Mẫu thử không hoàn chỉnh khi ta chỉ cần kiểm tra các điểm mốc 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.

*Loại mẫu thử ngẫu nhiên/ Không ngẫu nhiên.*

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

## 7.3. 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 giá trị trung gian.
- Kiểm tra các giá trị trung gian.
- Kiểm tra vệt chương trình.
- Thử hiệu năng: Các mẫu thử lớn, phải có một thời gian để thực hiện.

## Bài tập chương 4

1. Nêu vai trò của việc thiết kế kiểm soát và bảo mật hệ thống.
2. Có thể tránh được mọi sai sót và rủi ro đối với hệ thống không? Cách lựa chọn và cách khắc phục như thế nào?
3. Khi thiết kế các file dữ liệu ta dựa vào đâu? Các căn cứ nào cho phép ta xác định các thuộc tính của file: tên file, tên thuộc tính, các khóa và các thuộc tính kết nối....
4. Các đường truy nhập vào file dựa vào liên kết nào của mô hình thực thể liên kết.
5. Tại sao khi thiết kế các file, đôi khi người ta phá vỡ chuẩn 3NF? Điều đó có gây nên những lỗi cấm không? Cho ví dụ minh họa.

6. Mục đích của file chỉ dẫn để làm gì?
7. Thiết kế file dữ liệu và lựa chọn phần mềm là nhiệm vụ của người phân tích, thiết kế hay người lập trình.
8. Thiết kế các file dữ liệu cho các hệ thống: Hệ thống quản lý tuyển sinh, Hệ thống quản lý thư viện, Hệ thống quản lý học tập.
9. Từ biểu đồ luồng dữ liệu hãy xây dựng lược đồ cấu trúc chương trình cho hệ thống (con) mượn trả sách của thư viện.

## TÀI LIỆU THAM KHẢO

1. *Phân tích và thiết kế Tin học hệ thống Quản lý - kinh doanh - nghiệp vụ*. Ngô Trung Việt. Nhà xuất bản Giao thông vận tải - 2000
2. *Phân tích thiết kế, cài đặt hệ thống tin quản lý*. Viện Tin học. Năm 1990.
3. *Phân tích và thiết kế hệ thống thông tin*. Nguyễn Văn Ba. Nhà xuất bản Đại học quốc gia Hà Nội - 2003.
4. *Phân tích và thiết kế hệ thống thông tin*. Thạc Bình Cường. Nhà xuất bản Khoa học và Kỹ thuật – 2002.
5. *Phân tích và thiết kế hệ thống thông tin*. Đào Thanh Tĩnh. Học viện Kỹ thuật Quân sự - 2004.
6. *Cơ sở dữ liệu*. Đỗ Trung Tuấn. Nhà xuất bản Giáo dục - 1998.
7. *Nhập môn cơ sở dữ liệu quan hệ*. Lê Tiến Vương. Nhà xuất bản Khoa học và kỹ thuật - 1995.

## MỤC LỤC

<i>Lời giới thiệu</i> .....	3
<i>Lời nói đầu</i> .....	5
<i>Bài mở đầu</i> .....	7
<b>Chương 1. MỘT SỐ KHÁI NIỆM CƠ BẢN VỀ HỆ THỐNG</b>	
<i>I. Khái niệm chung về hệ thống. Hệ thống kinh doanh dịch vụ</i> .....	9
1. Khái niệm chung về hệ thống .....	9
2. Hệ thống kinh doanh dịch vụ và các hệ thống con của nó .....	11
<i>II. Sử dụng máy tính để xử lý tự động các thông tin</i> .....	14
1. Các hệ thống tin học .....	14
2. Các phương thức xử lý thông tin của máy tính .....	15
3. Một số loại hệ thống tin học .....	15
4. Tính mở của hệ thống tin .....	17
<i>III. Sự phát triển hệ thống</i> .....	17
1. Các giai đoạn phân tích, thiết kế và cài đặt .....	17
2. Chu trình phát triển .....	18
3. Mô hình hoá hệ thống .....	19
<b>Chương 2. KHẢO SÁT HIỆN TRẠNG VÀ XÁC LẬP DỰ ÁN</b>	
<i>I. Khảo sát và đánh giá hiện trạng</i> .....	22
1. Đại cương giai đoạn khảo sát .....	22
2. Các nguồn điều tra .....	24
3. Một số phương pháp khảo sát thường dùng .....	25
4. Quy trình điều tra .....	26
5. Phân loại và biên tập các thông tin điều tra .....	27
6. Phát hiện các yếu kém của hiện trạng và các yêu cầu cho tương lai .....	28

<b>II. Xây dựng dự án</b>	<b>28</b>
1. Xác định mục tiêu, phạm vi và hạn chế của dự án	28
2. Phác họa và nghiên cứu tính khả thi của giải pháp	29
3. Lập dự trù và kế hoạch triển khai dự án	30
<b>Chương 3. PHÂN TÍCH HỆ THỐNG</b>	
<b>I. Các mô hình và phương tiện diễn tả chức năng</b>	<b>32</b>
1. Các mức độ diễn tả chức năng	33
2. Biểu đồ phân cấp chức năng (FHD)	34
3. Biểu đồ luồng dữ liệu (DFD)	36
4. Các phương tiện đặc tả chức năng	41
<b>II. Phương pháp phân tích hệ thống về chức năng</b>	<b>43</b>
1. Kỹ thuật phân mức	43
2. Kỹ thuật chuyển đổi DFD vật lý thành DFD logic	50
3. Kỹ thuật chuyển từ DFD của hệ thống cũ sang DFD của hệ thống mới	52
<b>III. Các mô hình và phương tiện diễn tả dữ liệu</b>	<b>53</b>
1. Khái niệm diễn tả dữ liệu	53
2. Sự mã hoá	54
3. Từ điển dữ liệu	57
4. Mô hình thực thể	58
5. Mô hình quan hệ	67
<b>Chương 4. THIẾT KẾ HỆ THỐNG</b>	
<b>I. Thiết kế tổng thể và thiết kế giao diện</b>	<b>79</b>
1. Thiết kế tổng thể	79
2. Thiết kế các nhiệm vụ thủ công và các giao diện người-máy	82
<b>II. Thiết kế các kiểm soát</b>	<b>86</b>
1. Mục đích	86
2. Kiểm tra các thông tin thu thập và các thông tin xuất	86
3. Phân tích và thiết kế các kiểm soát	87
<b>III. Thiết kế cơ sở dữ liệu</b>	<b>91</b>
1. Mục đích	91
2. Thành lập lược đồ logic	92
3. Thành lập lược đồ vật lý	94

<i>IV. Thiết kế chương trình</i> .....	96
1. Mục đích .....	96
2. Lược đồ cấu trúc .....	97
3. Cách chuyển đổi DFD thành LCT .....	99
4. Chất lượng của LCT .....	101
5. Đặc tả các môđun .....	103
6. Đóng gói thành môđun tải .....	103
7. Thiết kế các mẫu thử .....	105
<i>Tài liệu tham khảo</i> .....	108



**NHÀ XUẤT BẢN HÀ NỘI**  
**4 - TỐNG DUY TÂN, QUẬN HOÀN KIẾM, HÀ NỘI**  
Điện thoại: (04)8.257063; 8.252916. Fax: (04)8.257063

---

**GIÁO TRÌNH**  
**PHÂN TÍCH THIẾT KẾ HỆ THỐNG**  
**THÔNG TIN QUẢN LÝ**  
**NHÀ XUẤT BẢN HÀ NỘI - 2005**

Chịu trách nhiệm xuất bản:  
**NGUYỄN KHẮC OÁNH**

Biên tập:  
**PHẠM QUỐC TUẤN**

Bìa:  
**TRẦN QUANG**

Trình bày, kỹ thuật vi tính:  
**HOÀNG LAN HƯƠNG**

Sửa bản in:  
**PHẠM QUỐC TUẤN**

---

In 1.560 cuốn, khổ 17 x 24 cm, tại Công ty In Khoa học Kỹ thuật - Hà Nội.  
Số in: 185. Giấy phép xuất bản số: 116GT/407 CXB cấp ngày 29/3/2005.  
In xong và nộp lưu chiểu quý II năm 2005.

**BỘ GIÁO TRÌNH XUẤT BẢN NĂM 2005**  
**KHOA TRƯỜNG TRUNG HỌC KINH TẾ KỸ THUẬT TIN HỌC**

1. THUẬT TOÁN LẬP TRÌNH
2. ĐÁNH MÁY VI TÍNH
3. SOẠN THẢO VÀ ĐÁNH MÁY VĂN BẢN
4. NGHIỆP VỤ THƯ KÝ
5. KẾ TOÁN MÁY
6. MARKETING
7. NGÔN NGỮ LẬP TRÌNH C
8. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI (C++)
9. BẢNG TÍNH ĐIỆN TỬ (EXCEL)
10. VISUAL BASIC
11. CẤU TRÚC MÁY TÍNH
12. GIAO TIẾP
13. ACCESS
14. MẠNG MÁY TÍNH
15. THIẾT KẾ WEB
16. BẢO TRÌ PC
17. HỆ ĐIỀU HÀNH
18. CƠ SỞ DỮ LIỆU
19. PHÂN TÍCH THIẾT KẾ HỆ THỐNG
20. KỸ THUẬT SỐ
21. PHOTOSHOP
22. CAD/CAM

