

# Giai đoạn Phân tích Thiết kế

Công nghệ phần mềm

# Tổng quan

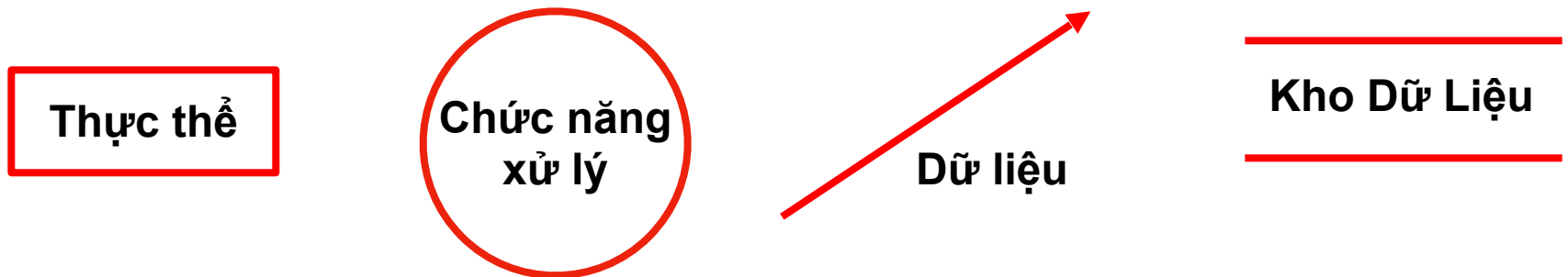
- Phân tích, thiết kế các xử lý của hệ thống
  - Theo hướng chức năng (DFD)
  - Theo hướng đối tượng (UseCase, Class)
- Phân tích, thiết kế dữ liệu (ERD)
- Phân tích, thiết kế giao diện
- Các công cụ hỗ trợ

# I.1. Phân tích các xử lý của hệ thống theo hướng chức năng

- Dùng lược đồ dòng dữ liệu (DFD) để mô hình hóa các xử lý.
- Xây dựng DFD
- Các phương pháp phân tích.
- Các mức mô hình hóa
- Sơ đồ tổng quát
- Các ví dụ

# LƯỢC ĐỒ DÒNG DỮ LIỆU (DFD)

- ▶ DFD được xây dựng từ 4 phần tử chính
  - **Thực thể**: tạo hoặc tiêu thụ thông tin, nằm bên ngoài phạm vi hệ thống
  - **Chức năng xử lý**: thực hiện chức năng nào đó, tiêu thụ và tạo thông tin, nằm bên trong phạm vi hệ thống
  - **Thông tin hay dữ liệu**
  - **Kho dữ liệu**: lưu trữ dữ liệu mà được sử dụng bởi nhiều chức năng xử lý



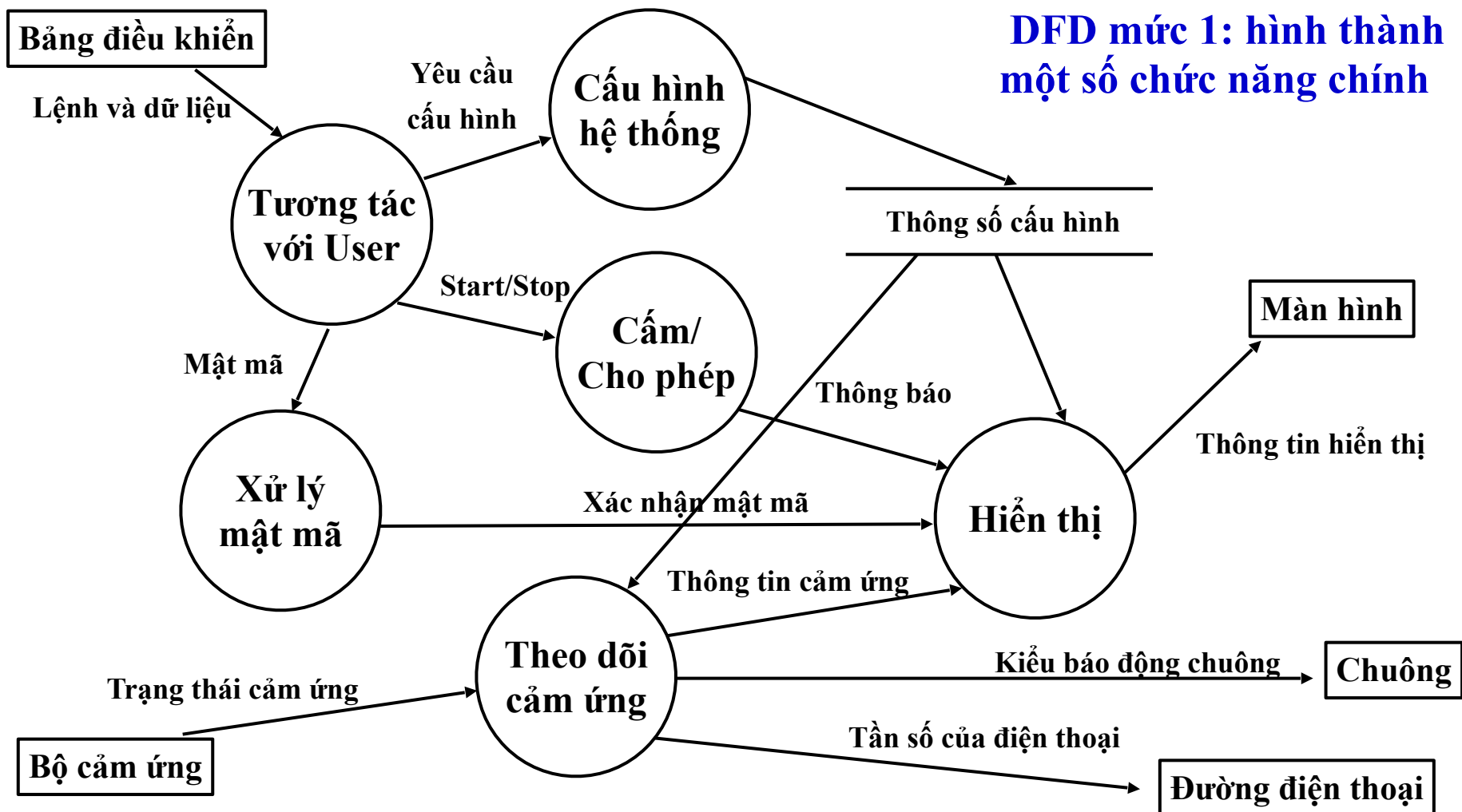
# Xây dựng DFD

- Cái vào của một tiến trình phải khác với cái ra của nó.
- Các đối tượng phải có tên duy nhất. Tuy nhiên các tác nhân ngoài hay các kho dữ liệu có thể vẽ tại nhiều vị trí khác nhau để dễ quan sát.
- Mỗi tiến trình phải có một ít nhất một cái vào (input) và ra (output).
- Luồng dữ liệu không thể đi trực tiếp từ kho này đến kho khác.
- Dữ liệu không thể di chuyển trực tiếp từ một tác nhân đến một kho và ngược lại.

# Xây dựng DFD

- Dữ liệu không thể đi trực tiếp từ tác nhân đến tác nhân.
- Một luồng dữ liệu không thể quay lại nơi nó vừa đi ra.
- Trong quá trình phân rã luôn luôn phải đảm bảo tính cân bằng: mọi luồng dữ liệu vào và ra, các tác nhân ngoài và các kho dữ liệu phải được bảo toàn trong sơ đồ ở mức sau.
- Quá trình phân rã dừng khi đạt được luồng dữ liệu sơ cấp (một tiến trình là một tính toán hay một thao tác dữ liệu đơn giản).

# Xây dựng DFD – Ví dụ



# Các phương pháp phân tích

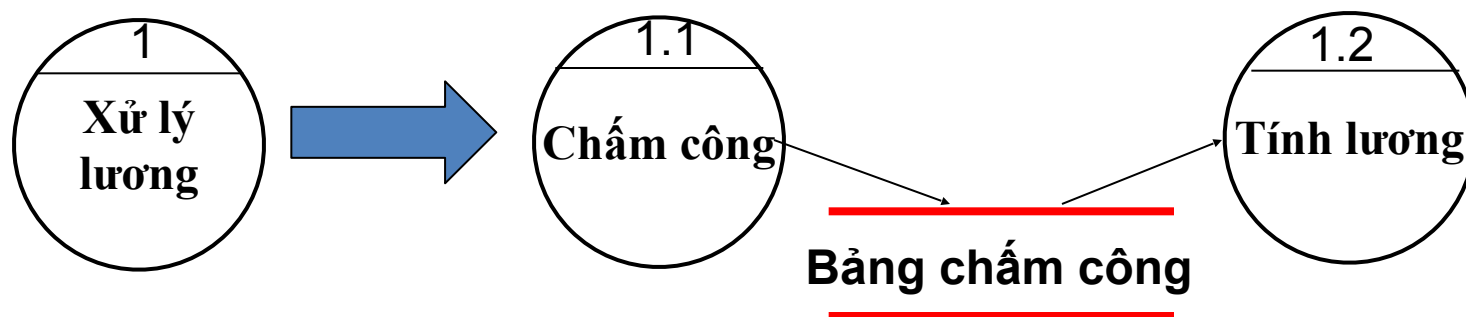
- Luật 1: Nếu xử lý đầu nhằm chuyển tiếp thông tin, và xử lý sau sử dụng thông tin đó.
- Thì xử lý ban đầu có thể phân tích thành một cặp xử lý kết nối với nhau bằng dòng dữ liệu.





# Các phương pháp phân tích

- Luật 2: Một xử lý có thể phân tích thành một cặp xử lý và kho dữ liệu. Vì 2 xử lý có thời điểm khác nhau, nên dữ liệu chuyển đổi phải được lưu lại trong một kho dữ liệu.



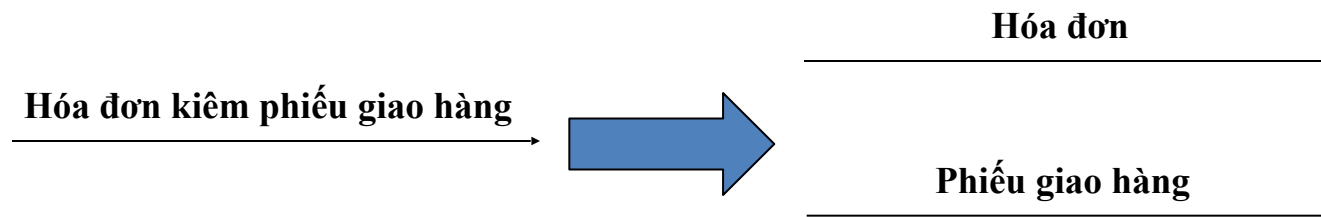
# Các phương pháp phân tích

- Luật 3: Một xử lý có thể phân tích thành một cặp xử lý riêng biệt. Vì 2 xử lý không có nhu cầu trao đổi thông tin.



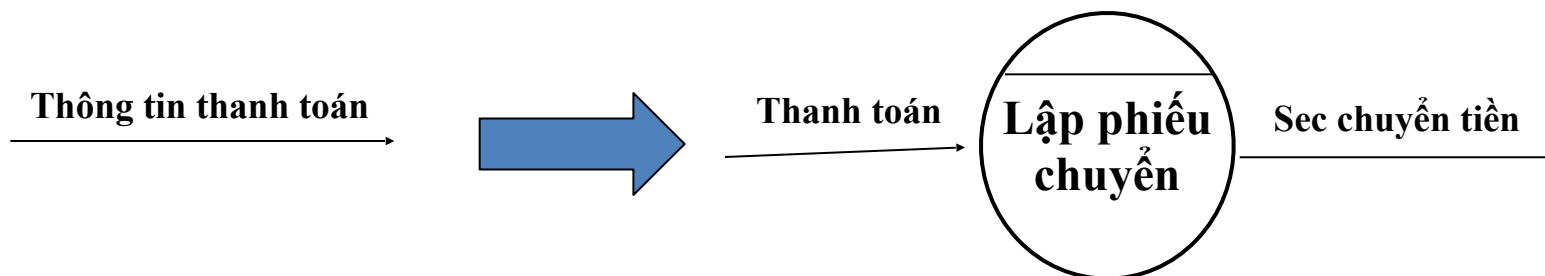
# Các phương pháp phân tích

- Luật 4: Tách một dòng dữ liệu thành tập các dòng dữ liệu. Để chuyển một dòng dữ liệu tổng hợp thành các dòng dữ liệu có nội dung độc lập.



# Các phương pháp phân tích

- Luật 5: Tách một dòng dữ liệu thành 2 dòng dữ liệu và một ô xử lý. Khi nhận ra dòng dữ liệu có chuyển đổi nội dung thông tin.

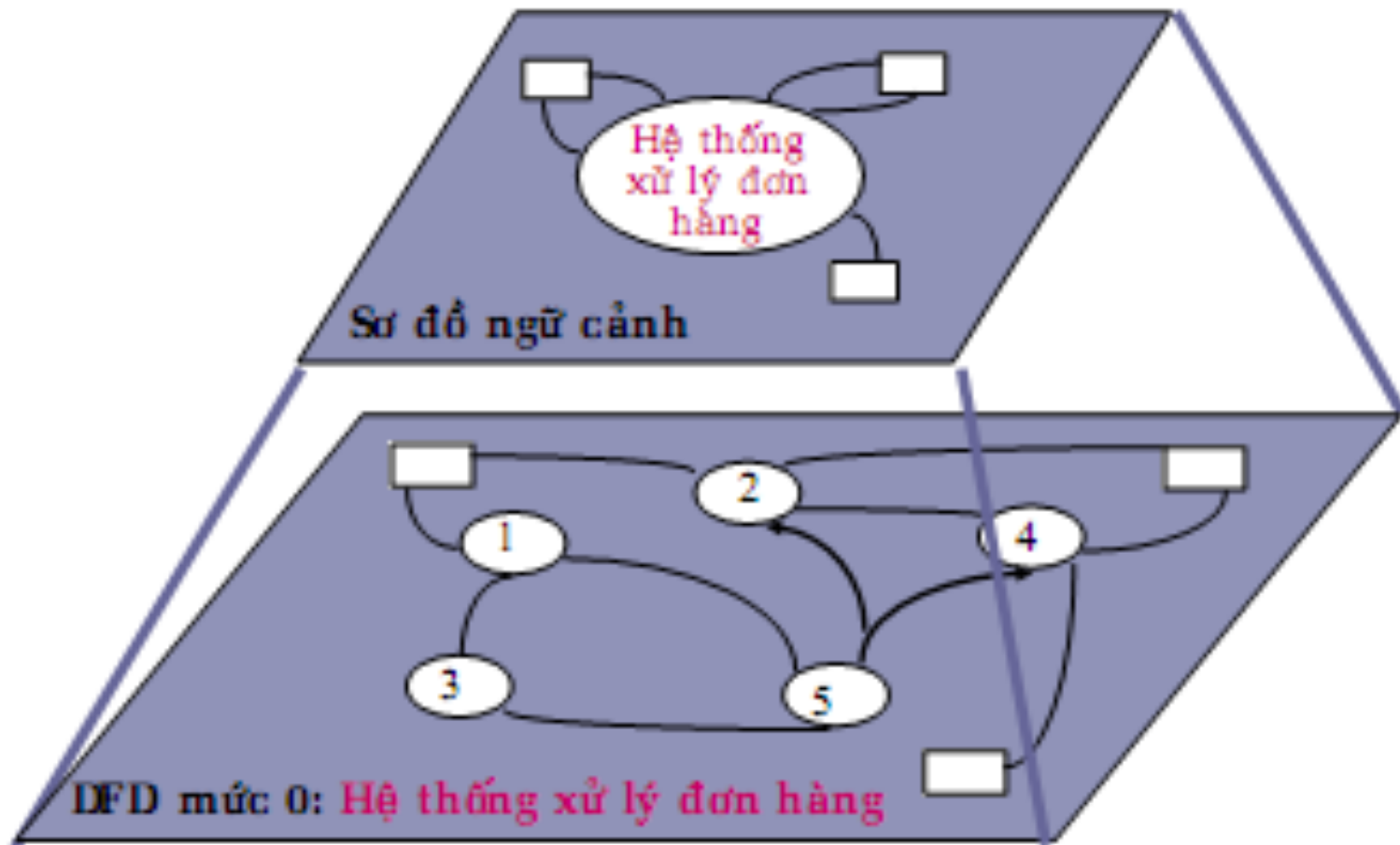


# Các phương pháp phân tích

- Thông thường dùng phương pháp phân tích dữ liệu từ trên xuống.

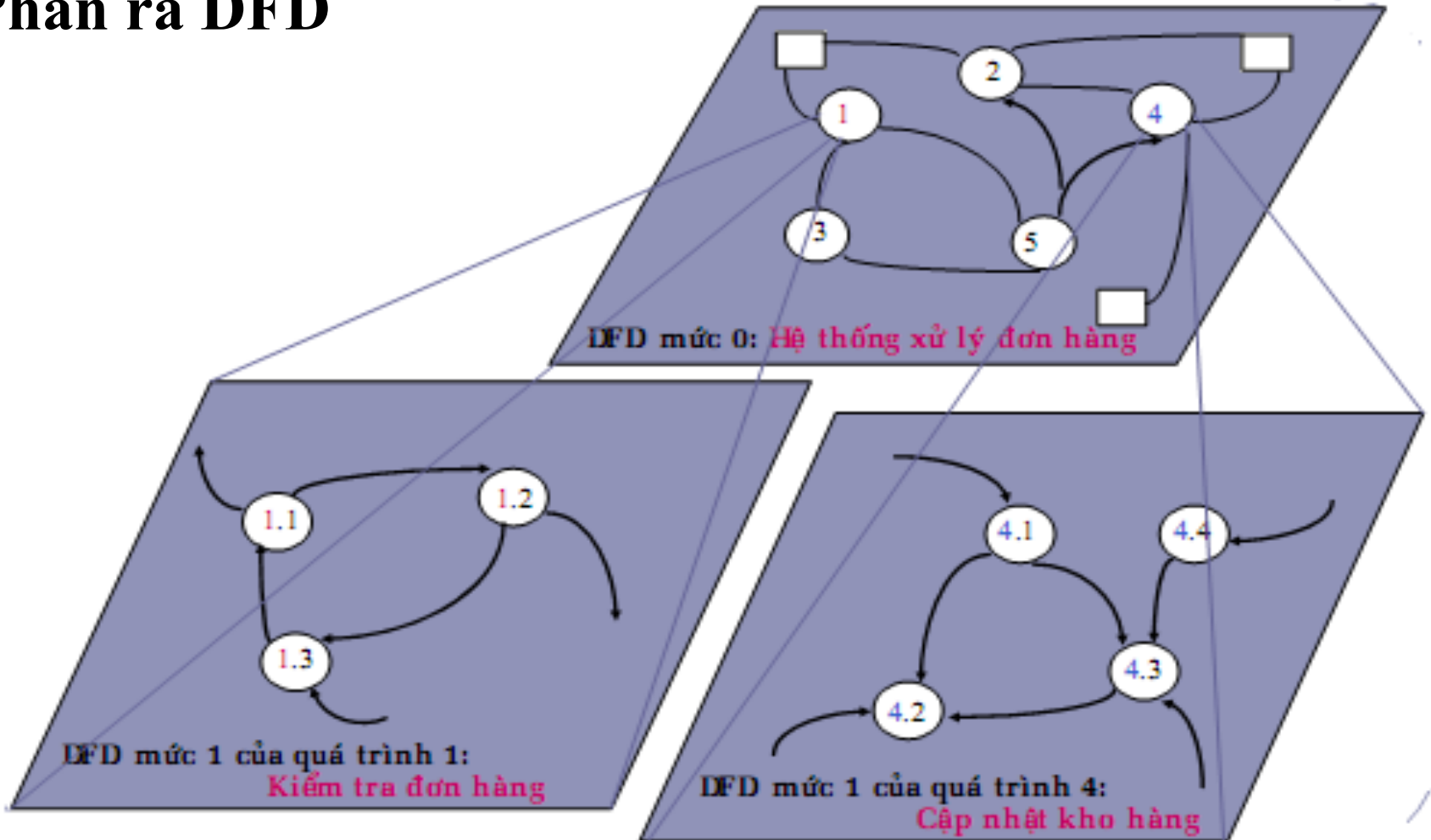
# Phân tích từ trên xuống

## Phân rã DFD

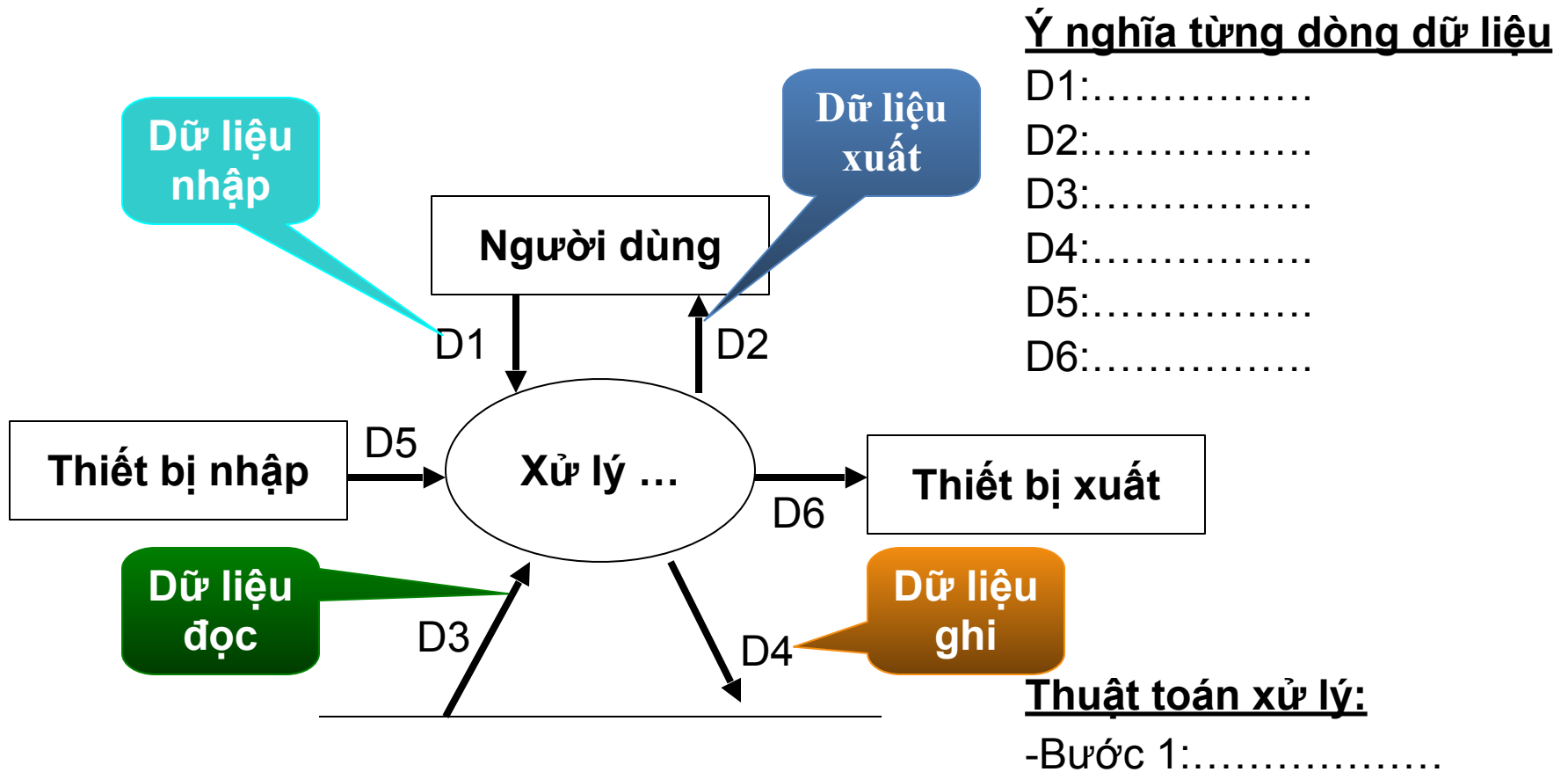


## +

# Phân rã DFD



# Sơ đồ tổng quát



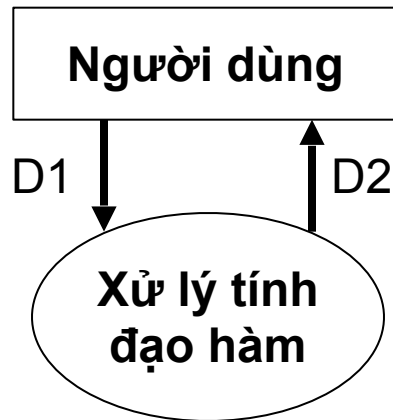


# Ví dụ 1

- Xét chức năng tính đạo hàm của một đơn thức

## 1. Phân tích

Sơ đồ luồng dữ liệu

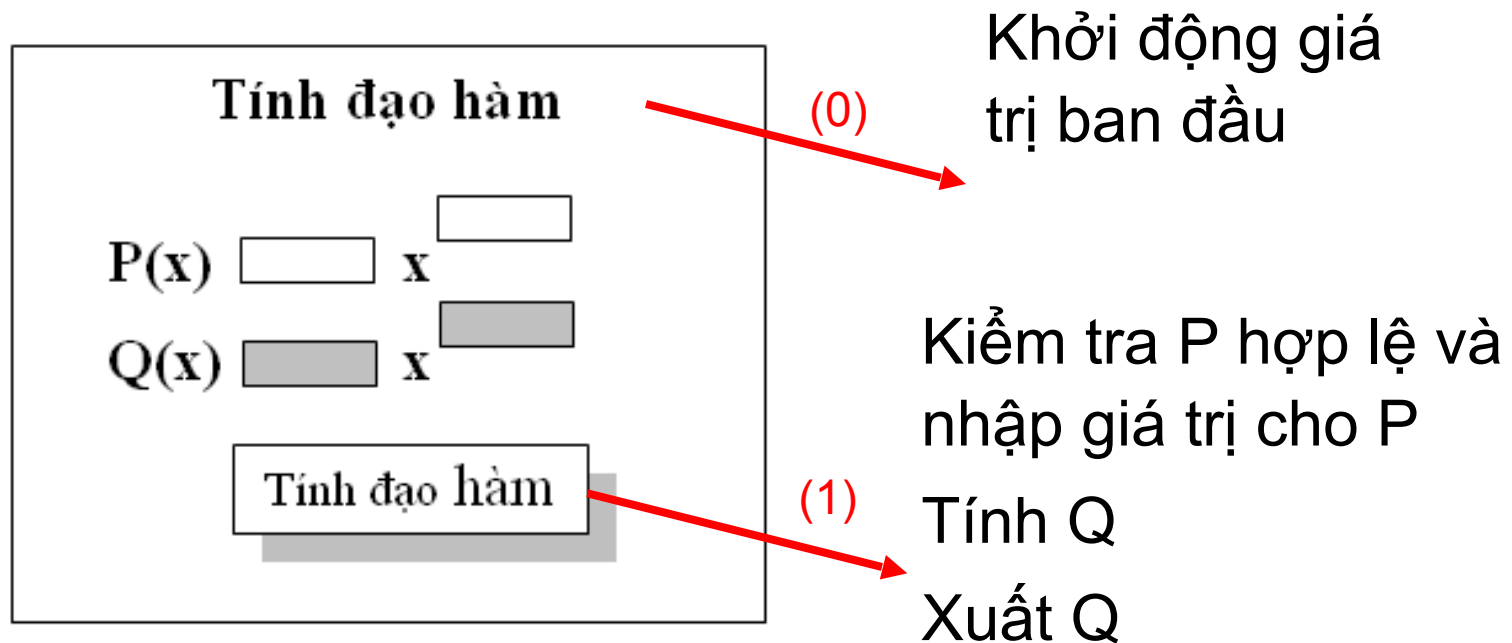


- ❖ D1: Đơn thức cần tính đạo hàm P
- ❖ D2: Đơn thức kết quả Q
- ❖ Thuật toán xử lý
  - Nhập và kiểm tra D1 ( $a \neq 0$ )
  - Tính và xuất D2

# Ví dụ 1

## 2. Thiết kế

➤ Mô tả chi tiết cách thức giao diện



# Ví dụ 1

## 2. Thiết kế

- Mô tả chi tiết kiểu dữ liệu: Sử dụng kiểu cấu trúc DON\_THUC với hai thành phần:
  - Hệ số có kiểu số thực
  - Số mũ có kiểu số nguyên (không âm)
- Mô tả chi tiết các hàm xử lý
  - Hàm xử lý biến cố 1
  - Hàm kiểm tra hệ số, Hàm kiểm tra số mũ
  - Hàm nhập đơn thức
  - Hàm tính đạo hàm, Hàm xuất đơn thức

# Ví dụ 1

## 3. Lập trình

‘Khai báo kiểu cấu trúc (C)

```
typedef struct DON_THUC
```

```
{
```

```
    int Heso;
```

```
    int Somu;
```

```
};
```

‘Khai báo biến

```
DON_THUC P, Q;
```

# Ví dụ 1

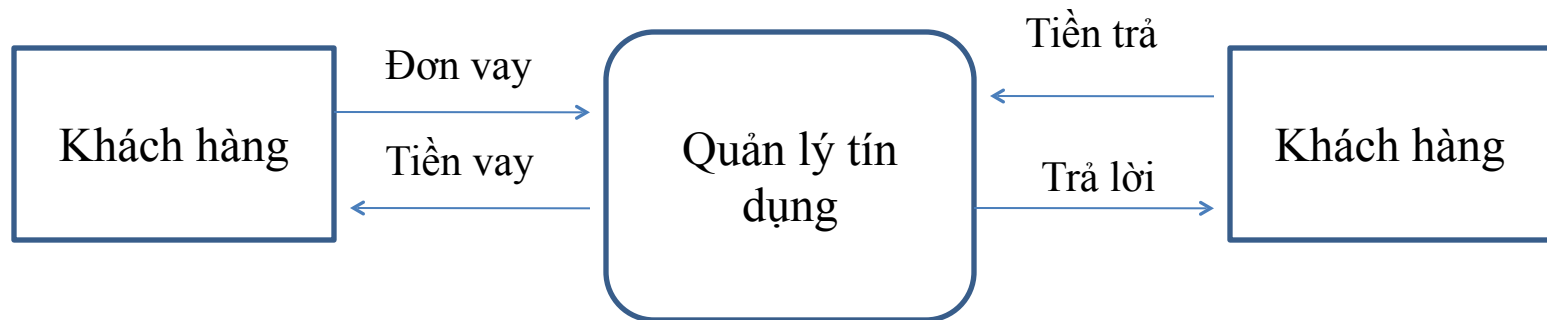
## 3. Lập trình

‘Hàm xử lý biến cố tính đạo hàm trên màn hình

```
void cmdDaoham_Click()  
    if (Kiem_Tra_He_So() && Kiem_Tra_So_Mu()){  
        Nhap();  
        DaoHam();  
        Xuat();  
    }  
}
```

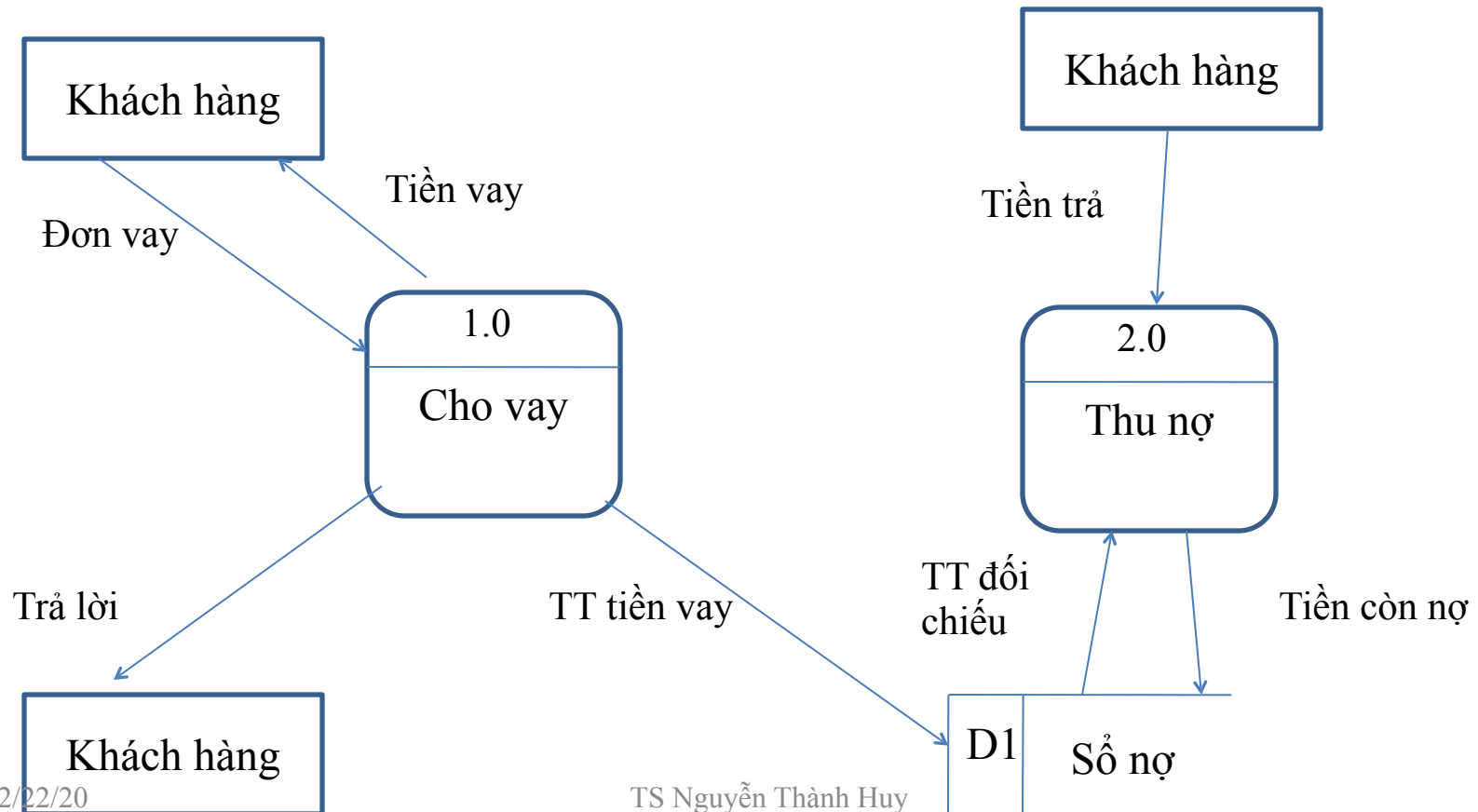
# Ví dụ DFD: Quản lý tín dụng

- Sơ đồ ngữ cảnh



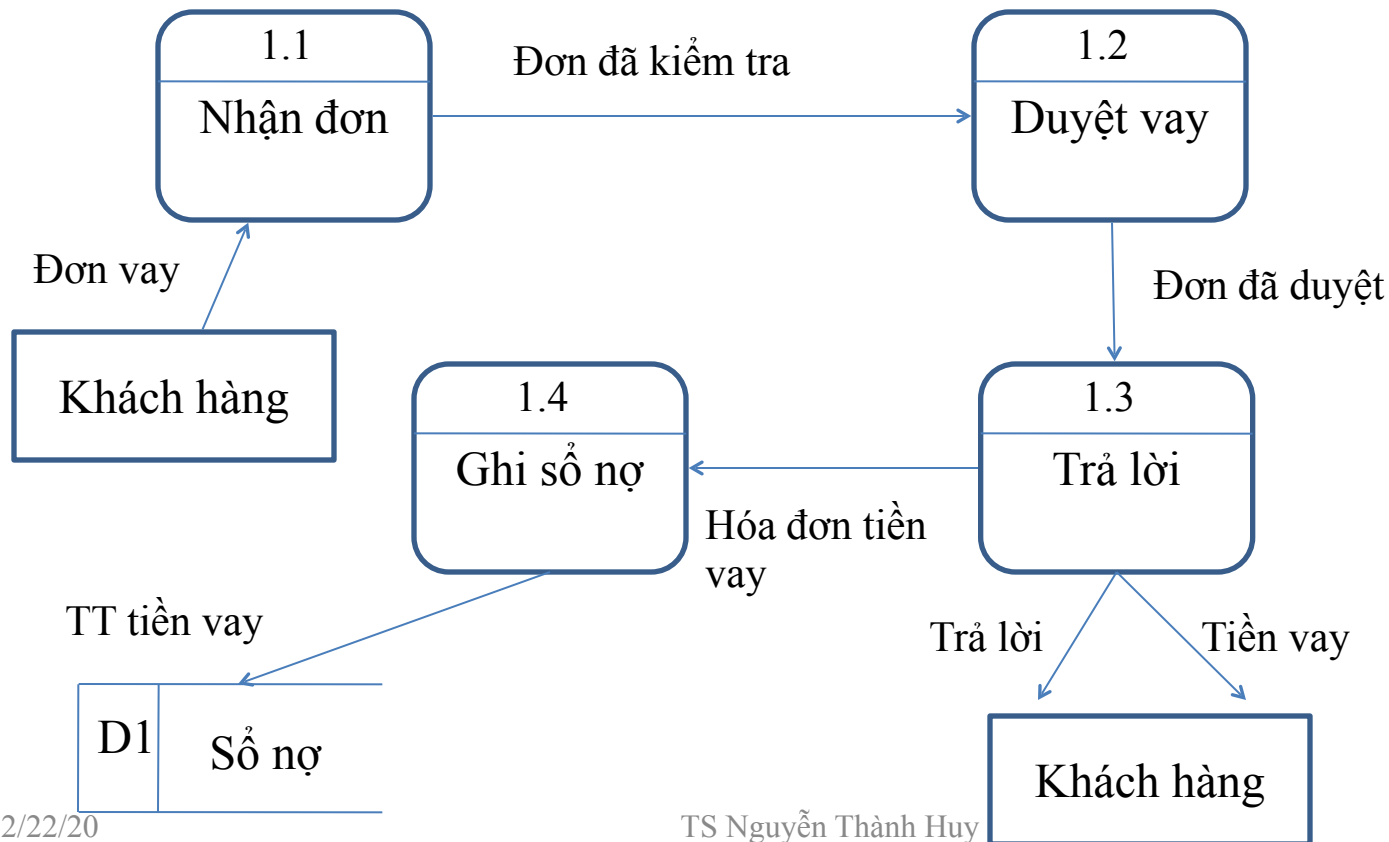
# Ví dụ DFD: Quản lý tín dụng

- Sơ đồ mức 0



# Ví dụ DFD: Quản lý tín dụng

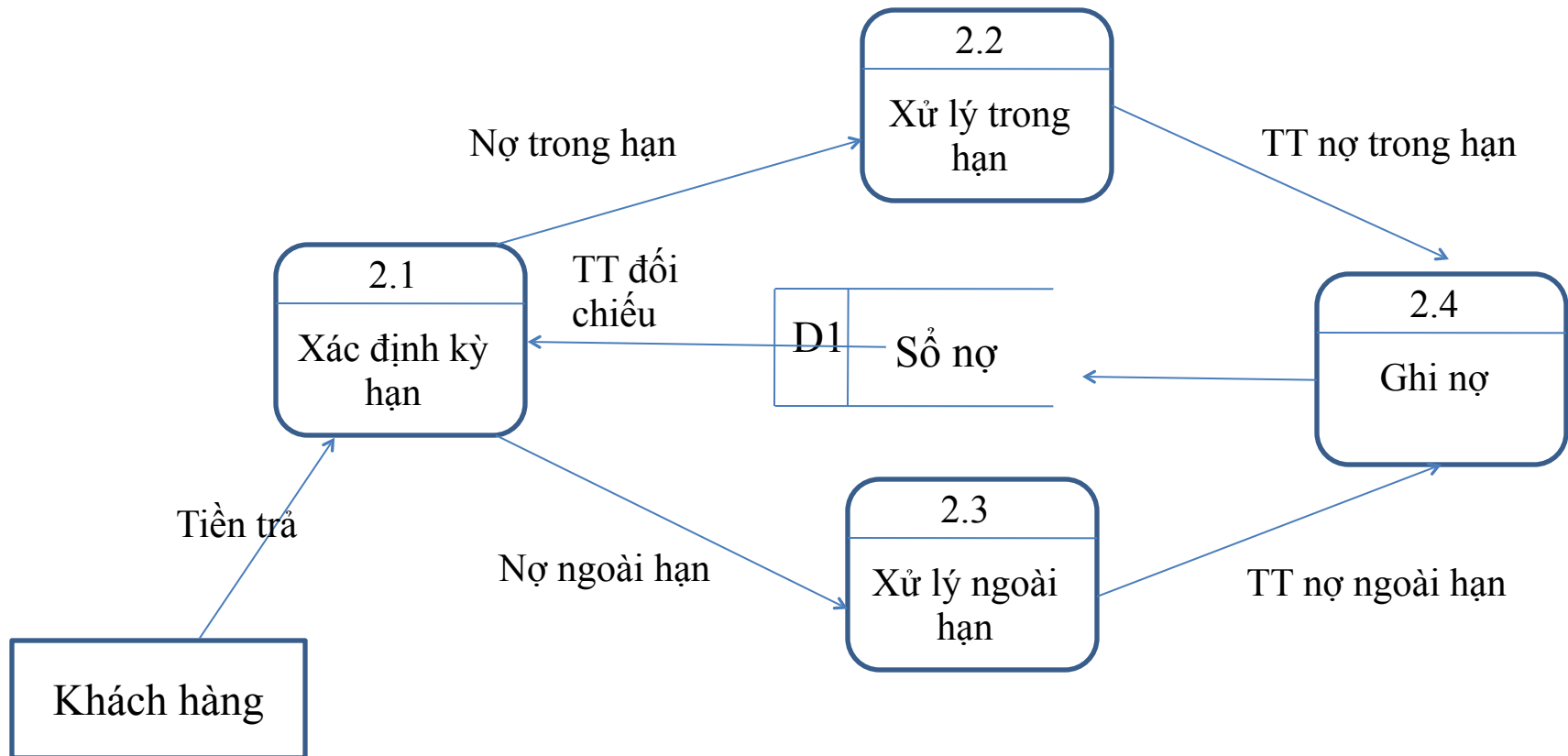
- Sơ đồ mức 1 của 1.0



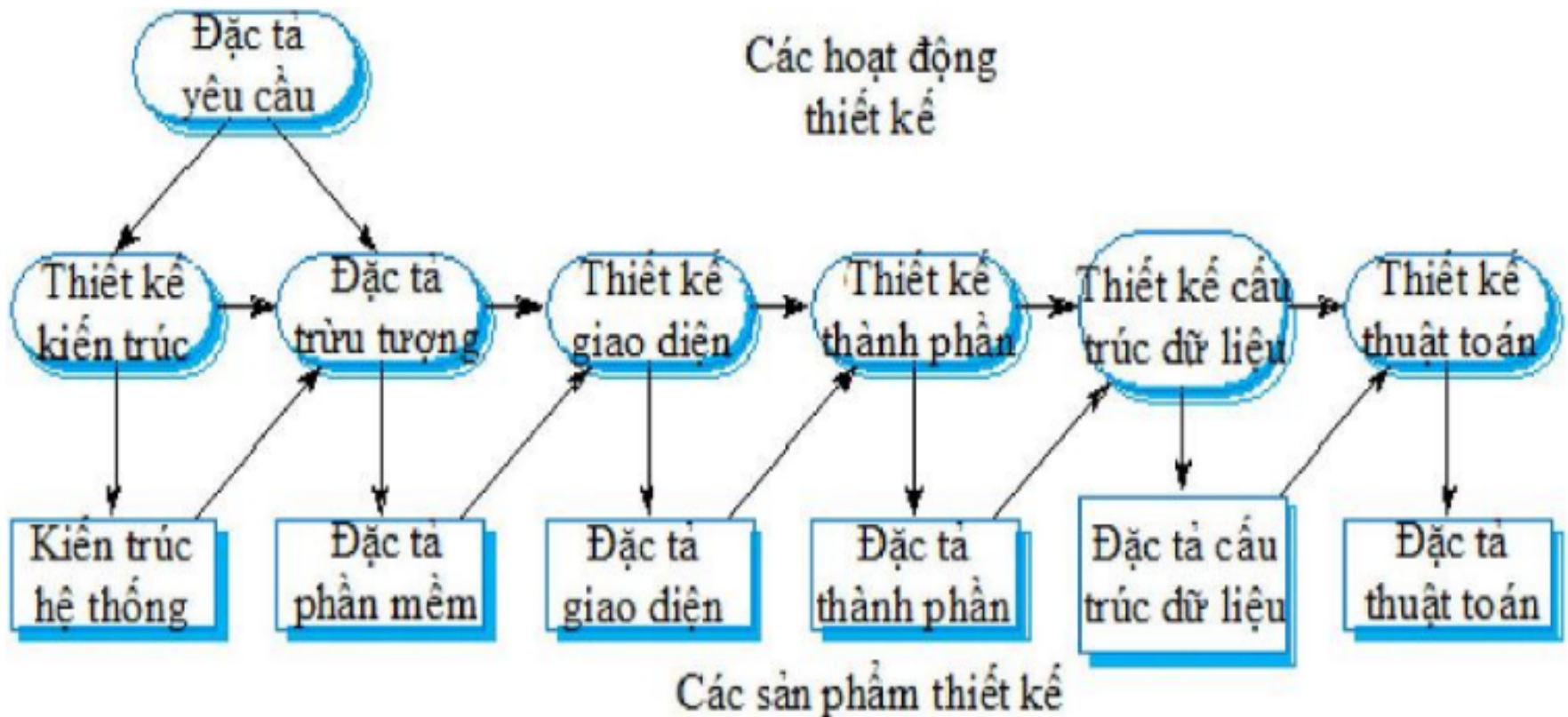


# Ví dụ DFD: Quản lý tín dụng

- Sơ đồ mức 1 của 2.0



# Các hoạt động chính của thiết kế



# Các hoạt động chính của thiết kế

- Thiết kế kiến trúc:

Xác định hệ tổng thể phần mềm bao gồm các hệ con và các quan hệ giữa chúng và ghi thành tài liệu

- Đặc tả trừu tượng:

Các đặc tả trừu tượng cho mỗi hệ con về các dịch vụ mà nó cung cấp cũng như các ràng buộc chúng phải tuân thủ.

# Các hoạt động chính của thiết kế

- Thiết kế giao diện:

Giao diện của từng hệ con với các hệ con khác được thiết kế và ghi thành tài liệu; đặc tả giao diện không được mơ hồ và cho phép sử dụng hệ con đó mà không cần biết về thiết kế nội tại của nó

- Thiết kế các thành phần:

Các dịch vụ mà một hệ con cung cấp được phân chia cho các thành phần hợp thành của nó

# Các hoạt động chính của thiết kế

- Thiết kế cấu trúc dữ liệu:

Các cấu trúc dữ liệu được dùng trong việc thực hiện hệ thống được thiết kế chi tiết và đặc tả

- Thiết kế thuật toán:

Các thuật toán được dùng cho các dịch vụ được thiết kế chi tiết và được đặc tả.

# III. Thiết kế giao diện

- Giao diện người dùng (User Interface)
  - Giao diện người dùng cần phải được thiết kế sao cho phù hợp với kỹ năng, kinh nghiệm và sự trông đợi của người sử dụng nó.
  - Người sử dụng hệ thống thường đánh giá hệ thống thông qua giao diện hơn là chức năng của nó.
  - Giao diện của hệ thống nghèo nàn có thể khiến người sử dụng tạo ra các lỗi hết sức nghiêm trọng.

# Thiết kế giao diện

- Tác nhân con người trong thiết kế giao diện
  - Khả năng nhớ tức thời của con người hạn chế.
  - Nhu cầu của người dùng hệ thống
  - Kinh nghiệm, năng lực
    - khả năng dùng bàn phím, mouse,...
    - tốc độ phản ứng, khả năng nhớ thao
  - Sở thích, văn hóa, lứa tuổi
    - màu sắc, ngôn ngữ, biểu tượng
    - các loại tương tác khác nhau: hình ảnh, văn bản, âm thanh...

# Thiết kế giao diện

- Các nguyên tắc thiết kế giao diện
  - Sự quen thuộc của người sử dụng:  
Giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính.  
Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy ... mà không nên sử dụng những khái niệm như thư mục, danh mục ...



# Thiết kế giao diện

- Nhất quán:  
hệ thống nên hiển thị ở mức thống nhất thích hợp. Ví dụ: các câu lệnh và menu nên có cùng định dạng ...
- Tối thiểu hoá sự bất ngờ:  
Nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tự.

# Thiết kế giao diện

- Khả năng phục hồi:  
hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi.
- Hướng dẫn người sử dụng:  
Như hệ thống trợ giúp, hướng dẫn trực tuyến ...
- Tính đa dạng:  
Hỗ trợ nhiều loại tương tác cho nhiều loại người sử dụng khác nhau.

# Thiết kế giao diện

- Kỹ thuật: hai vấn đề chính cần giải quyết
  - Tương tác người dùng: cách người dùng đưa thông tin vào cho hệ thống
  - Biểu diễn thông tin: cách hệ thống trình diễn thông tin cho người dùng
  - Giải pháp được xem xét theo góc độ
    - Thiết bị tương tác người dùng
    - Cách hệ thống trình diễn - chủng loại giao diện
    - Mô hình tương tác

# Thiết kế giao diện

- Thiết bị tương tác
  - Màn hình
  - Bàn phím
  - Mouse, bút từ, ...
  - Màn hình cảm biến
  - Mic/Speaker
  - Smart cards,...
- Cả thiết bị lẫn phương thức đều đang tiến hóa:
  - Nhận dạng tiếng nói, chữ viết...

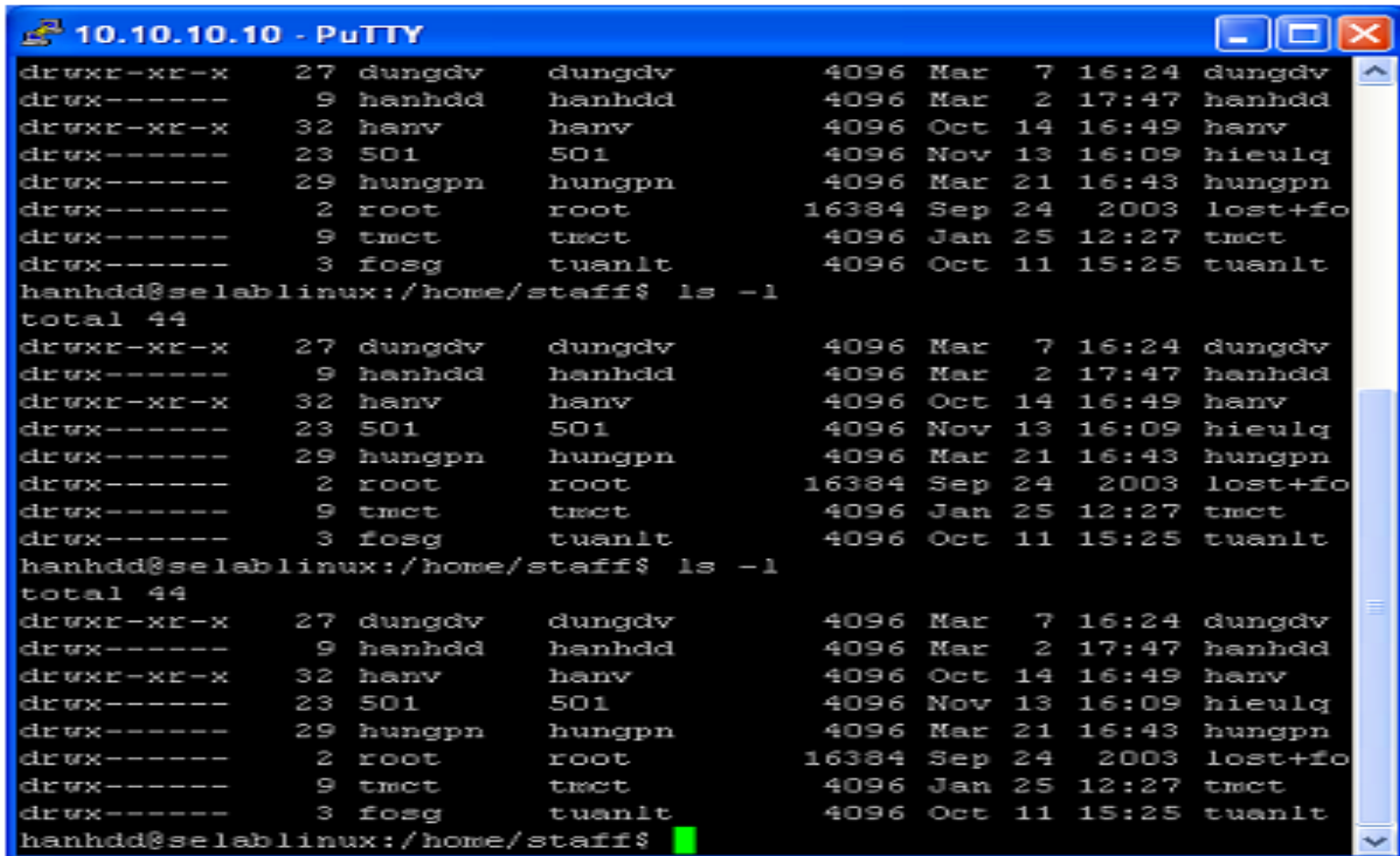
# Thiết kế giao diện

- Các loại giao diện:
  - Giao diện dòng lệnh:
    - Là phương thức tương tác đầu tiên Nhập lệnh/dữ liệu từ bàn phím
    - Dễ cài đặt so với GUI
      - thực hiện thông qua hàm chuẩn của ngôn ngữ
      - không tốn tài nguyên hệ thống

# Thiết kế giao diện

- Có khả năng tổ hợp lệnh để tạo các lệnh phức tạp
  - phối hợp các filter, tạo các lô xử lý (batch)
  - có thể lập trình bằng (Unix) shell
  - có thể tự động hóa
- Thao tác thực hiện tuần tự
  - khó sửa lỗi thao tác trước đó
- Không phù hợp với người dùng ít kinh nghiệm
  - khó học, khó nhớ dễ nhầm
  - đòi hỏi kỹ năng sử dụng bàn phím

# Thiết kế giao diện



The screenshot shows a PuTTY terminal window titled "10.10.10.10 - PuTTY". The terminal displays a directory listing command and its output. The command is executed twice, showing the same results. The output is a table of files and directories with columns for permissions, owner, group, size, date, and filename.

```
drwxr-xr-x 27 dungdv dungdv 4096 Mar 7 16:24 dungdv
drwx----- 9 hanhdd hanhdd 4096 Mar 2 17:47 hanhdd
drwxr-xr-x 32 hanv hanv 4096 Oct 14 16:49 hanv
drwx----- 23 501 501 4096 Nov 13 16:09 hieulq
drwx----- 29 hungpn hungpn 4096 Mar 21 16:43 hungpn
drwx----- 2 root root 16384 Sep 24 2003 lost+fo
drwx----- 9 tmct tmct 4096 Jan 25 12:27 tmct
drwx----- 3 fosg tuanit 4096 Oct 11 15:25 tuanit
hanhdd@selablinux:/home/staff$ ls -l
total 44
drwxr-xr-x 27 dungdv dungdv 4096 Mar 7 16:24 dungdv
drwx----- 9 hanhdd hanhdd 4096 Mar 2 17:47 hanhdd
drwxr-xr-x 32 hanv hanv 4096 Oct 14 16:49 hanv
drwx----- 23 501 501 4096 Nov 13 16:09 hieulq
drwx----- 29 hungpn hungpn 4096 Mar 21 16:43 hungpn
drwx----- 2 root root 16384 Sep 24 2003 lost+fo
drwx----- 9 tmct tmct 4096 Jan 25 12:27 tmct
drwx----- 3 fosg tuanit 4096 Oct 11 15:25 tuanit
hanhdd@selablinux:/home/staff$ ls -l
total 44
drwxr-xr-x 27 dungdv dungdv 4096 Mar 7 16:24 dungdv
drwx----- 9 hanhdd hanhdd 4096 Mar 2 17:47 hanhdd
drwxr-xr-x 32 hanv hanv 4096 Oct 14 16:49 hanv
drwx----- 23 501 501 4096 Nov 13 16:09 hieulq
drwx----- 29 hungpn hungpn 4096 Mar 21 16:43 hungpn
drwx----- 2 root root 16384 Sep 24 2003 lost+fo
drwx----- 9 tmct tmct 4096 Jan 25 12:27 tmct
drwx----- 3 fosg tuanit 4096 Oct 11 15:25 tuanit
hanhdd@selablinux:/home/staff$
```

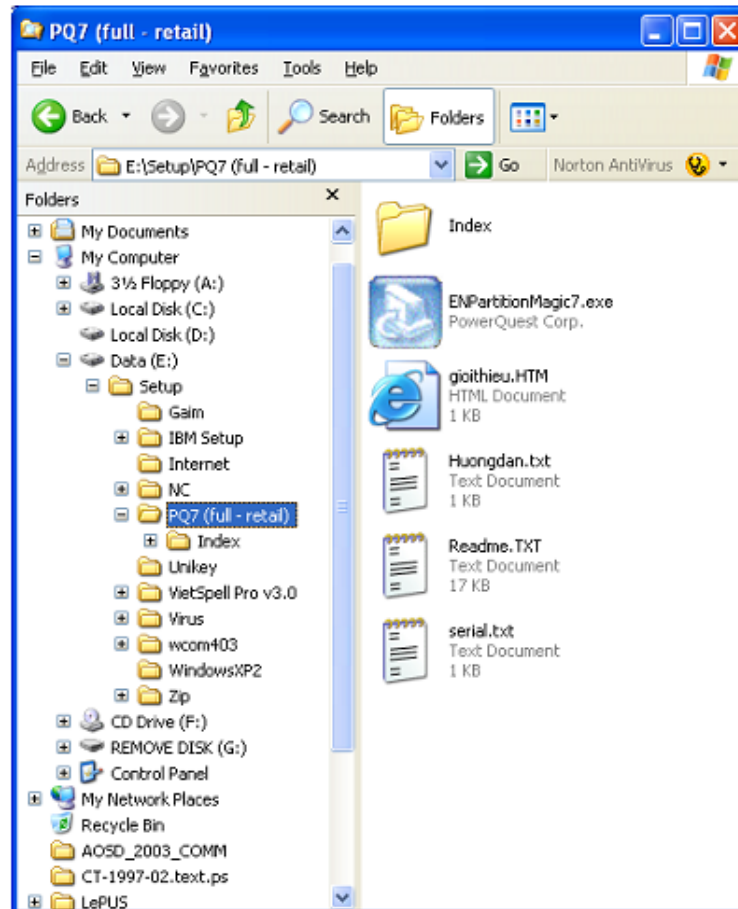
# Thiết kế giao diện

## – Giao diện đồ họa (GUI)

- Là giao diện thông dụng trên PC, Apple, Unix WS
- Dễ học, dễ sử dụng, thuận tiện với người ít kinh nghiệm
- Có nhiều cửa sổ, có thể tương tác song song trên nhiều cửa sổ mà không bị mất thông tin
- Có thể hiển thị, tương tác dữ liệu trên nhiều vị trí trong cửa sổ



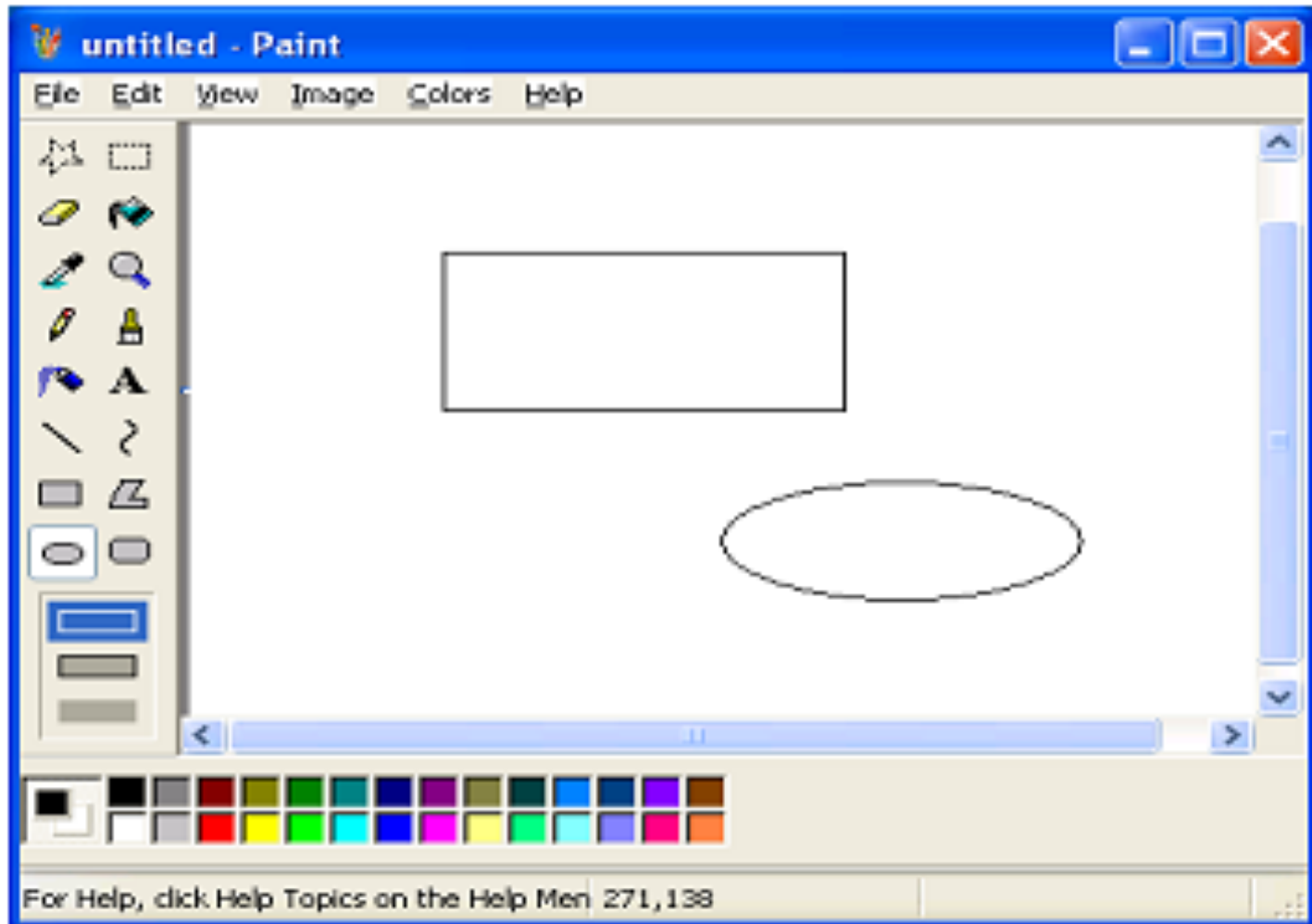
# Thiết kế giao diện



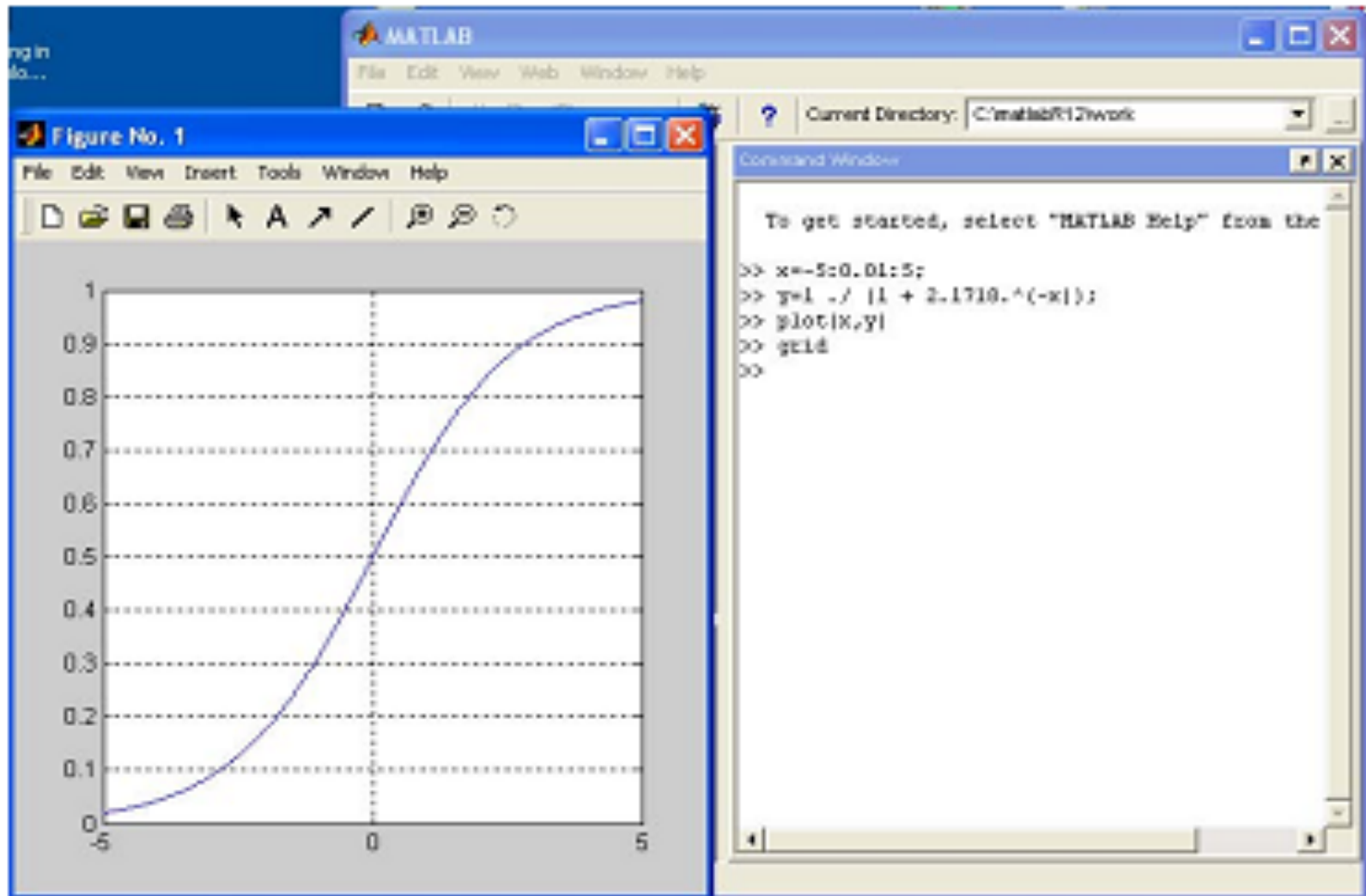
# Thiết kế giao diện

- Hình thức tương tác
  - Tương tác trực tiếp với thông tin ví dụ: soạn thảo; nhập dữ liệu vào các form...
    - dễ học, dễ sử dụng
    - nhận được tức thời kết quả thao tác
    - cài đặt phức tạp, tốn tài nguyên phần cứng
  - Tương tác gián tiếp ví dụ: chọn lệnh từ menu, giao diện dòng lệnh
    - kém trực quan
    - thuận tiện khi lặp lại thao tác phức tạp

# Thiết kế giao diện

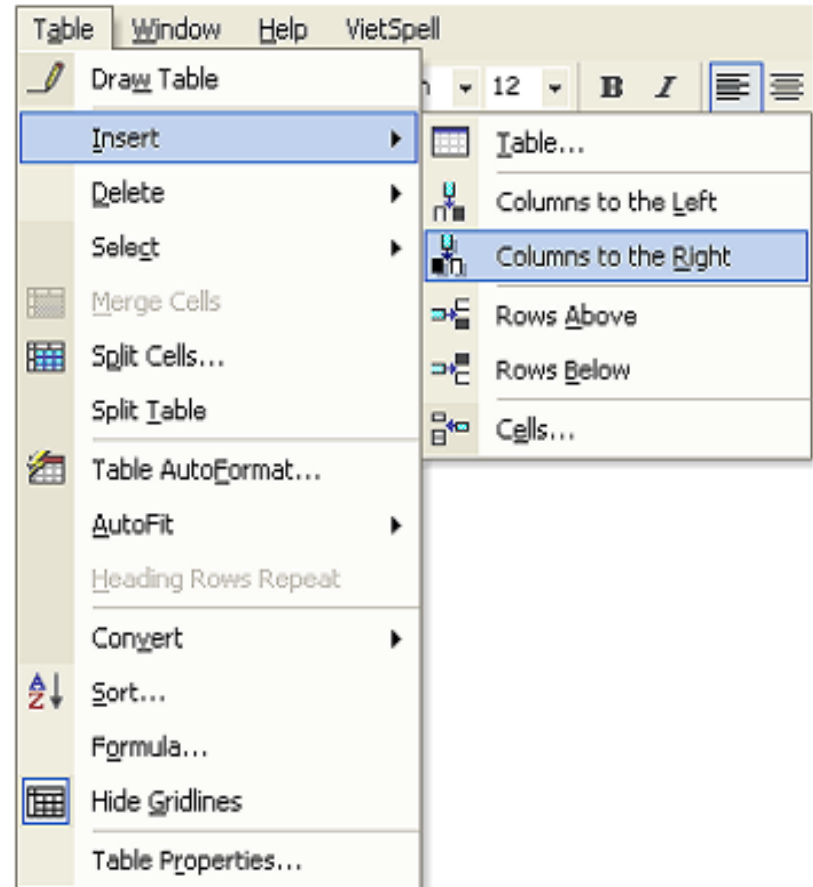


# Thiết kế giao diện



# Thiết kế giao diện

- Sử dụng thực đơn (menu)
  - Không cần nhớ lệnh
  - Tối thiểu hóa dùng bàn phím
  - Tránh các lỗi như sai lệnh, sai tham số
  - Dễ dàng tạo các trợ giúp theo ngữ cảnh



# Thiết kế giao diện

- Hộp thoại (Dialog)
  - Dùng khi cần người dùng đưa ra lựa chọn quyết định



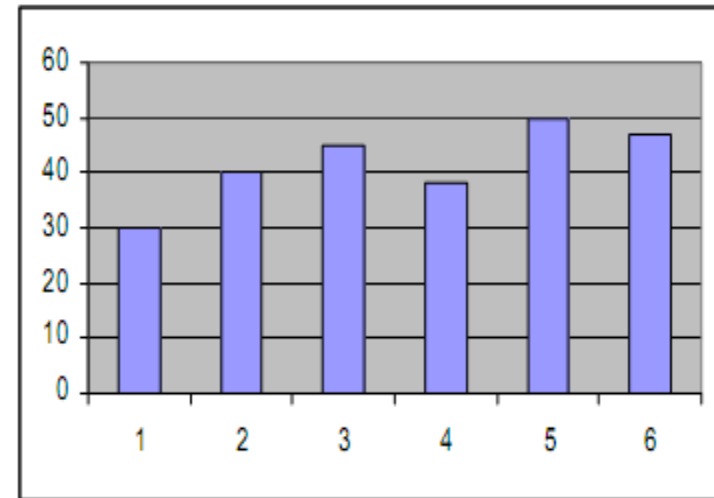
# Thiết kế giao diện

- Các vấn đề
  - Phương pháp hiển thị thông tin
    - Hiển thị bằng văn bản (text)
      - chính xác
      - dễ cài đặt
    - Hiển thị bằng đồ họa (graphic)
      - trực quan
      - dễ dàng nhìn nhận các mối quan hệ

Tháng 1: 30

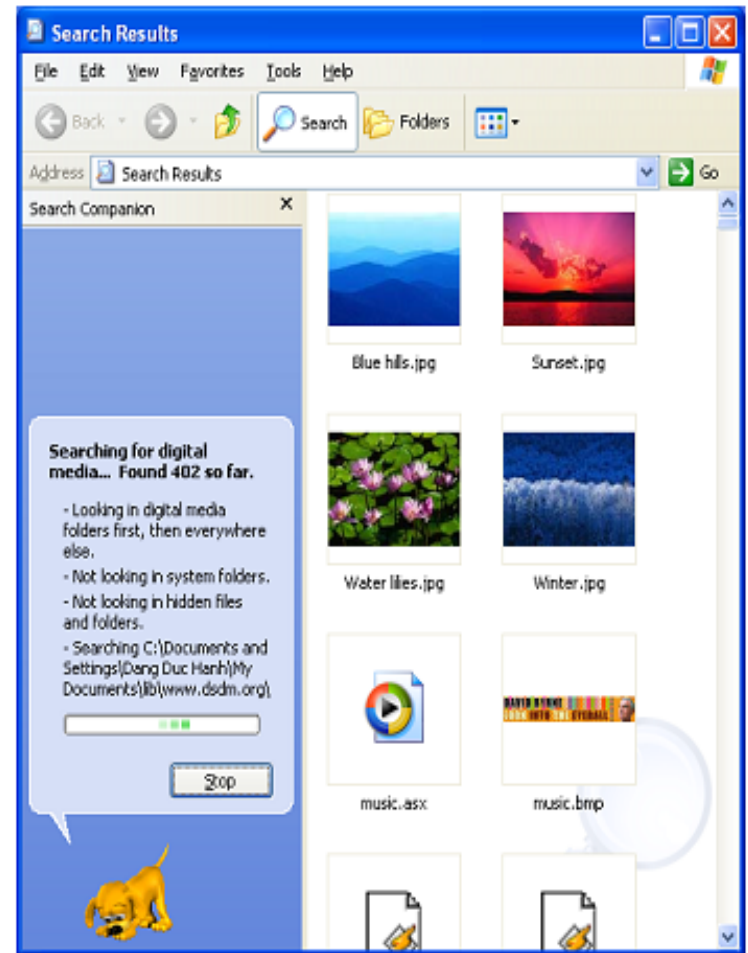
Tháng 2: 40

....



# Thiết kế giao diện

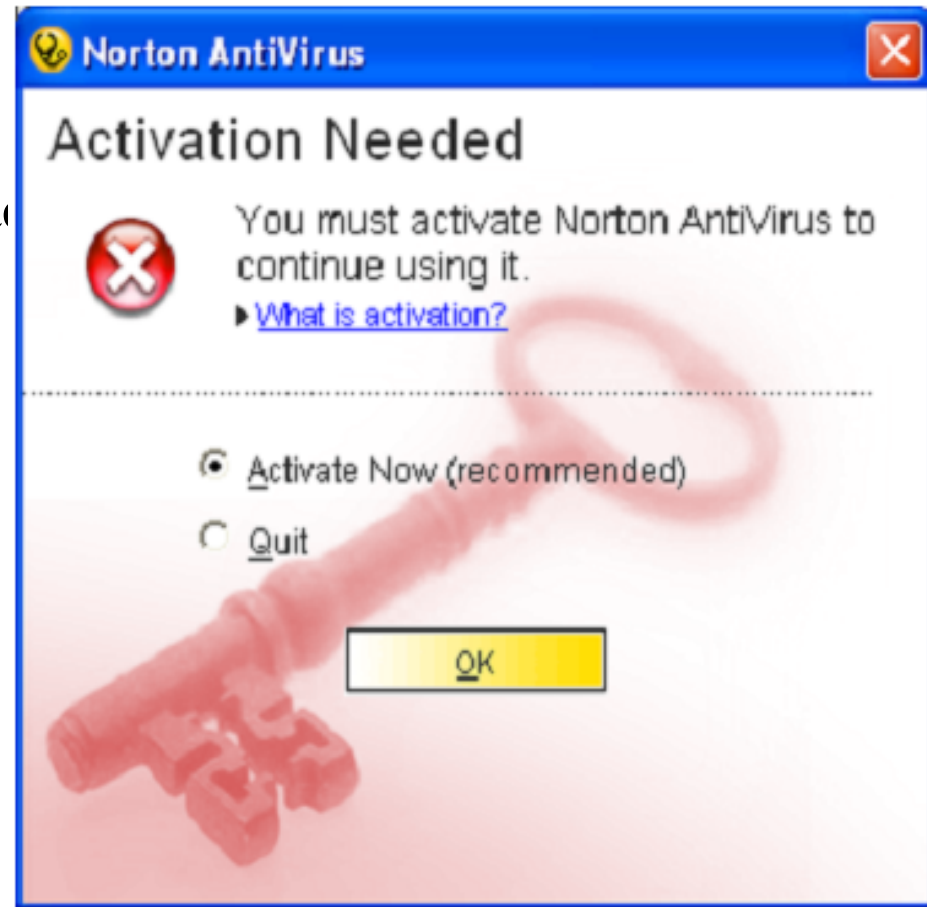
- Thời gian phản hồi
  - Thời gian trung bình
    - thời gian trung bình phản hồi với thao tác
    - người dùng không thể đợi quá lâu ( $< 3s$ )
    - cần chứng tỏ hệ thống đang hoạt động
  - Độ biến thiên thời gian phản hồi
    - Gây cảm giác hệ thống gặp lỗi





# Thiết kế giao diện

- Thông báo
  - Phản hồi của hệ thống đối với thao tác
  - Cần có nghĩa, dễ hiểu, đưa ra các
    - thông tin hữu ích với người dùng
    - tránh đưa ra các số hiệu
    - định dạng thông báo phải nhất quán
- Thông báo lỗi
  - chính xác
  - có tính xây dựng (nguyên nhân, cách khắc phục,...)

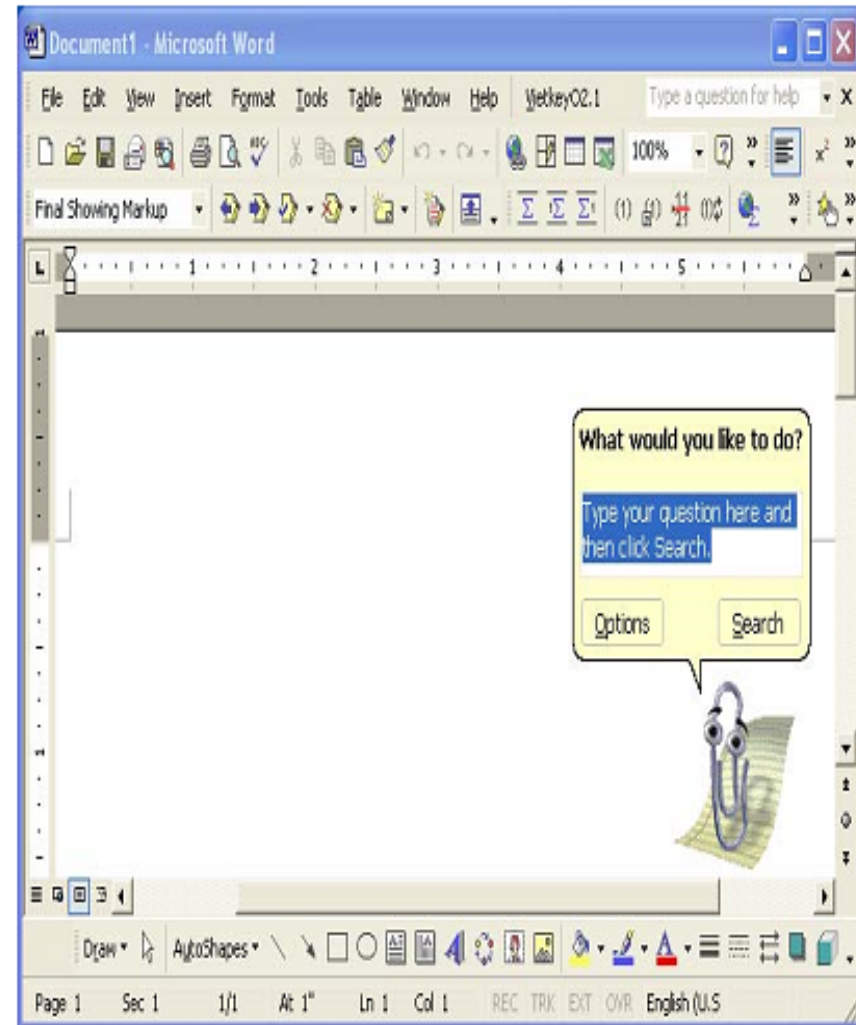


# Thiết kế giao diện

- Số lượng thông báo:
  - Đưa ra càng nhiều càng tốt = càng thân thiện
  - Đưa ra một lượng tối thiểu = im lặng là vàng
- Thời điểm và thứ tự đưa ra thông báo
- Yêu cầu phản hồi đối với thông báo

# Thiết kế giao diện

- Tiện ích (trợ giúp)
  - Cần có các tiện ích trợ giúp người sử dụng
  - Tiện ích tích hợp: trợ giúp trực tuyến, theo ngữ cảnh
    - chú giải thao tác, giao diện
  - Các tài liệu trực tuyến
    - tra cứu chức năng hệ thống
  - Các macro: tự động hóa thao tác
    - ví dụ: MS Word macro

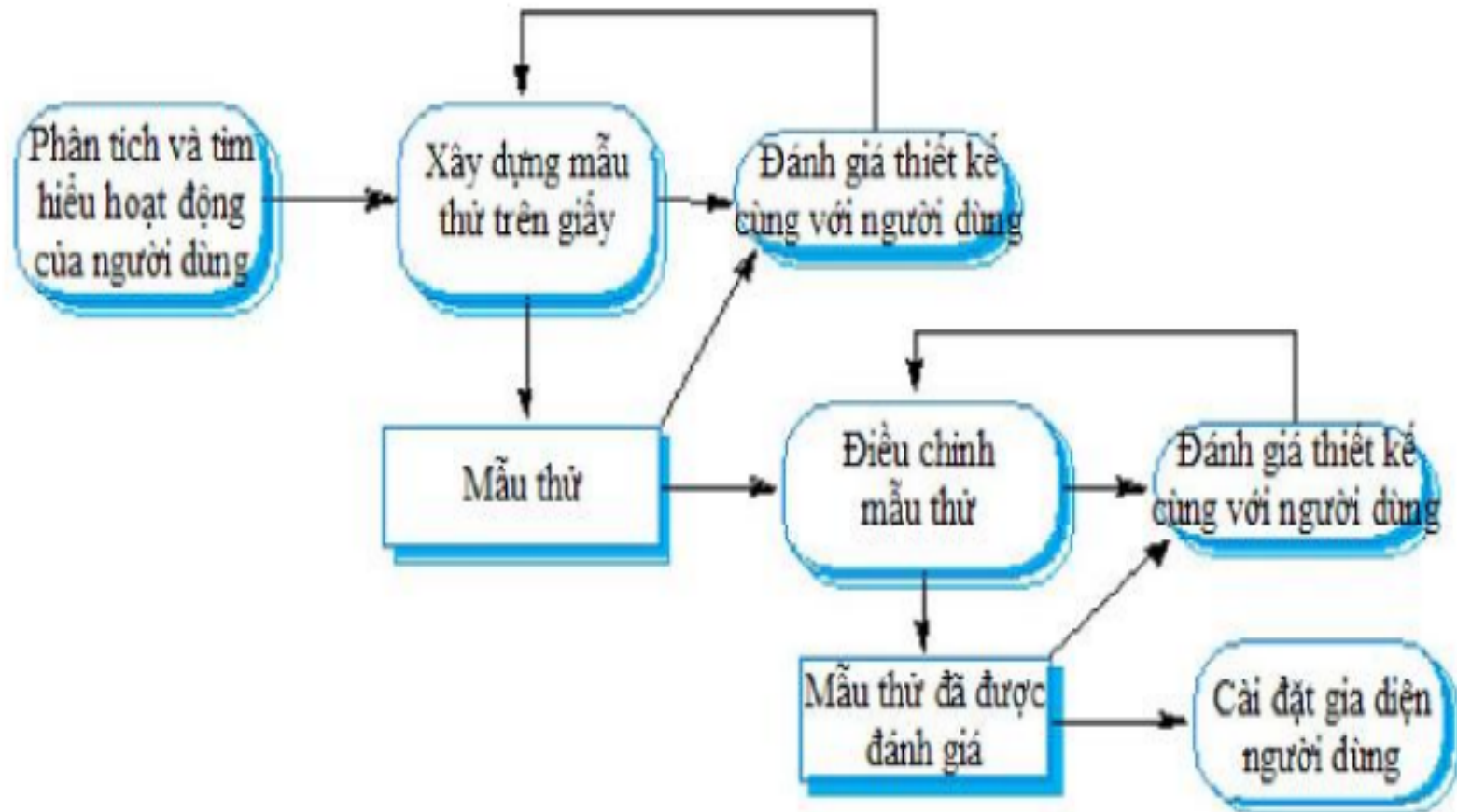


# Thiết kế giao diện

- Tính công nghệ
  - Giao diện là phần tử dễ thay đổi
    - thay đổi quy trình, phương thức thao tác
    - thay đổi môi trường (phần cứng, hệ điều hành)
    - nâng cấp (đẹp hơn, dễ sử dụng hơn...)
  - Giao diện phải dễ sửa đổi
  - Giao diện phải có tính khả chuyển
  - Giao diện nên độc lập với xử lý

# Thiết kế giao diện

- Quy trình thiết kế giao diện



# Thiết kế giao diện

- Quy trình thiết kế giao diện
  - Phân tích người sử dụng:  
Tìm hiểu những gì người sử dụng sẽ làm với hệ thống.
  - Lập mẫu thử hệ thống:  
Xây dựng một tập các mẫu thử để thử nghiệm
  - Đánh giá giao diện:  
Thử nghiệm các mẫu thử cùng với người sử dụng.

# Thiết kế giao diện

- Phân tích người sử dụng
  - Các kỹ thuật phân tích
    - Phân tích nhiệm vụ: mô hình hoá các bước cần thực hiện để hoàn thành một nhiệm vụ.
    - Phân tích nhiệm vụ phân cấp.
    - Phỏng vấn và trắc nghiệm: hỏi người sử dụng về những gì mà họ làm.
    - Mô tả: quan sát người sử dụng làm việc và hỏi họ về những cách mà không được biết tới

# Thiết kế giao diện

- Lập mẫu thử giao diện người dùng
  - Mẫu thử cho phép người sử dụng có được những kinh nghiệm trực tiếp với giao diện.
  - Nếu không có những kinh nghiệm trực tiếp như vậy thì không thể đánh giá được khả năng có thể sử dụng được của giao diện.
  - Quy trình :
    - Lập các mẫu thử trên giấy.
    - Tinh chỉnh mẫu thử và xây dựng chúng



# Thiết kế giao diện

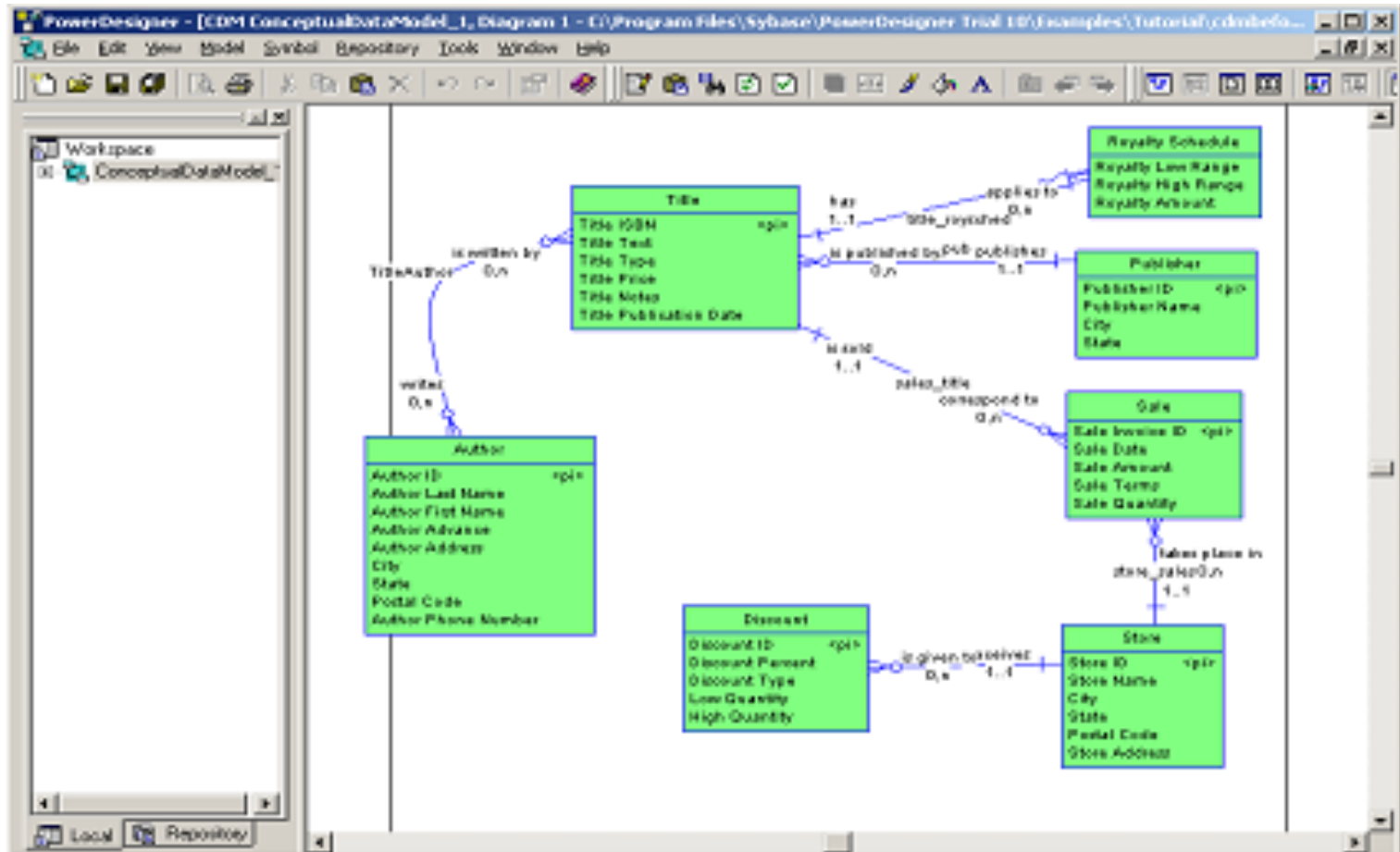
- Các kỹ thuật lập mẫu thử
  - Mẫu thử hướng nguyên mẫu: sử dụng công cụ như Macromedia Director để xây dựng một tập hợp các nguyên mẫu và màn hình. Khi người sử dụng tương tác với chúng thì màn hình sẽ thay đổi để hiển thị trạng thái kế tiếp.
  - Lập trình trực quan: sử dụng các ngôn ngữ được thiết kế cho việc phát triển nhanh như Visual Basic.
  - Mẫu thử dựa Internet: sử dụng web browser và script

# Thiết kế giao diện

- Đánh giá giao diện người dùng
  - Các kỹ thuật:
    - Trắc nghiệm lại các phản hồi của người sử dụng
    - Ghi lại quá trình sử dụng mẫu thử của hệ thống và đánh giá nó.
    - Lựa chọn những thông tin về việc sử dụng dễ dàng và các lỗi của người sử dụng.
    - Cung cấp mã lệnh trong phần mềm để thu thập những phản hồi của người sử dụng một cách trực tuyến.

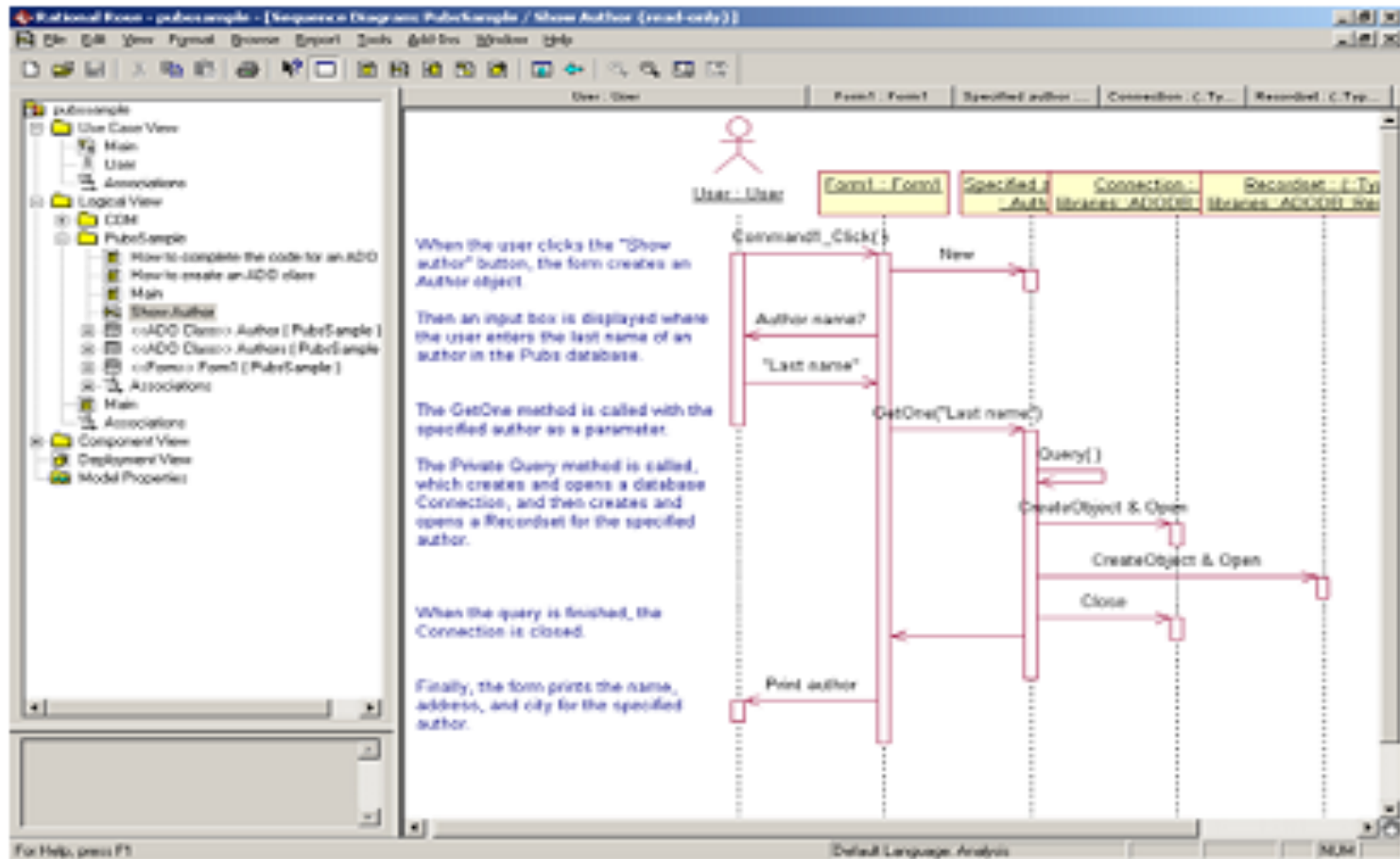
# Công cụ thiết kế

## – Power Designer



# Công cụ thiết kế

## – Rational Rose



# Thảo luận

- Quan hệ giữa hướng chức năng và hướng đối tượng?
- Phân biệt mức quan niệm, logic, vật lý?
- Ai là người thực hiện công việc phân tích, thiết kế trong dự án phần mềm?
- Kết quả của giai đoạn phân tích, thiết kế để làm gì?