

**Nhập môn**  
**Công nghệ học Phần mềm**  
**Introduction to Software Engineering**

**Department of Software Engineering**  
**Faculty of Information Technology**  
**Hanoi University of Technology**  
**TEL: 04-8682595 FAX: 04-8692906**  
**Email: [cnpm@it-hut.edu.vn](mailto:cnpm@it-hut.edu.vn)**

# Phần III

## Yêu cầu người dùng

### User's Requirements

#### Chương 5: Phương pháp xác định yêu cầu

5.1. Kỹ thuật xác định yêu cầu

5.2. Nội dung xác định yêu cầu

5.3. Các nguyên lý phân tích yêu cầu

# 5.1. Kỹ thuật xác định yêu cầu phần mềm

## SW Requirements Engineering

- **Yêu cầu phần mềm:** là tất cả các yêu cầu về phẩm mềm do khách hàng - người sử dụng phần mềm - nêu ra, bao gồm: các chức năng của phần mềm, hiệu năng của phần mềm, các yêu cầu về thiết kế và giao diện, các yêu cầu đặc biệt khác

- **Thông thường các yêu cầu phần mềm được phân loại theo 4 thành phần của phần mềm:**
  - Các yêu cầu về phần mềm (Software)
  - Các yêu cầu về phần cứng (Hardware)
  - Các yêu cầu về dữ liệu (Data)
  - Các yêu cầu về con người (People, Users)
- ***Mục đích:* mục đích của yêu cầu phần mềm là xác định được phần mềm đáp ứng được các yêu cầu và mong muốn của khách hàng - người sử dụng phần mềm**

# Tại sao cần phải đặt ra yêu cầu phần mềm ?

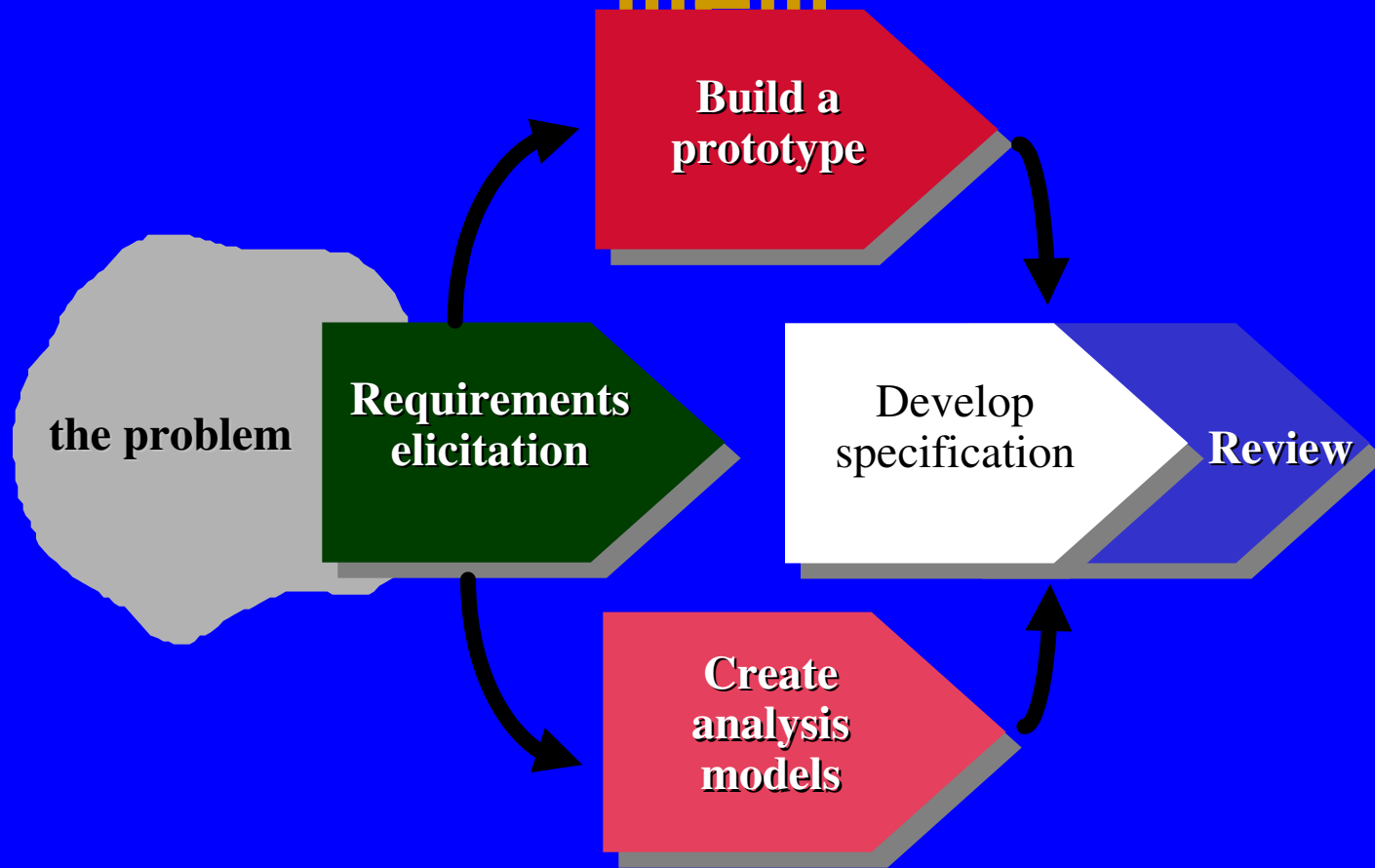
- Khách hàng chỉ có những ý tưởng còn mơ hồ về phần mềm cần phải xây dựng để phục vụ công việc của họ, chúng ta phải sẵn sàng, kiên trì theo đuổi để đi từ các ý tưởng mơ hồ đó đến “Phần mềm có đầy đủ các tính năng cần thiết”
- Khách hàng rất hay thay đổi các đòi hỏi của mình, chúng ta nắm bắt được các thay đổi đó và sửa đổi các mô tả một cách hợp lý

## 5.2. Nội dung xác định yêu cầu phần mềm

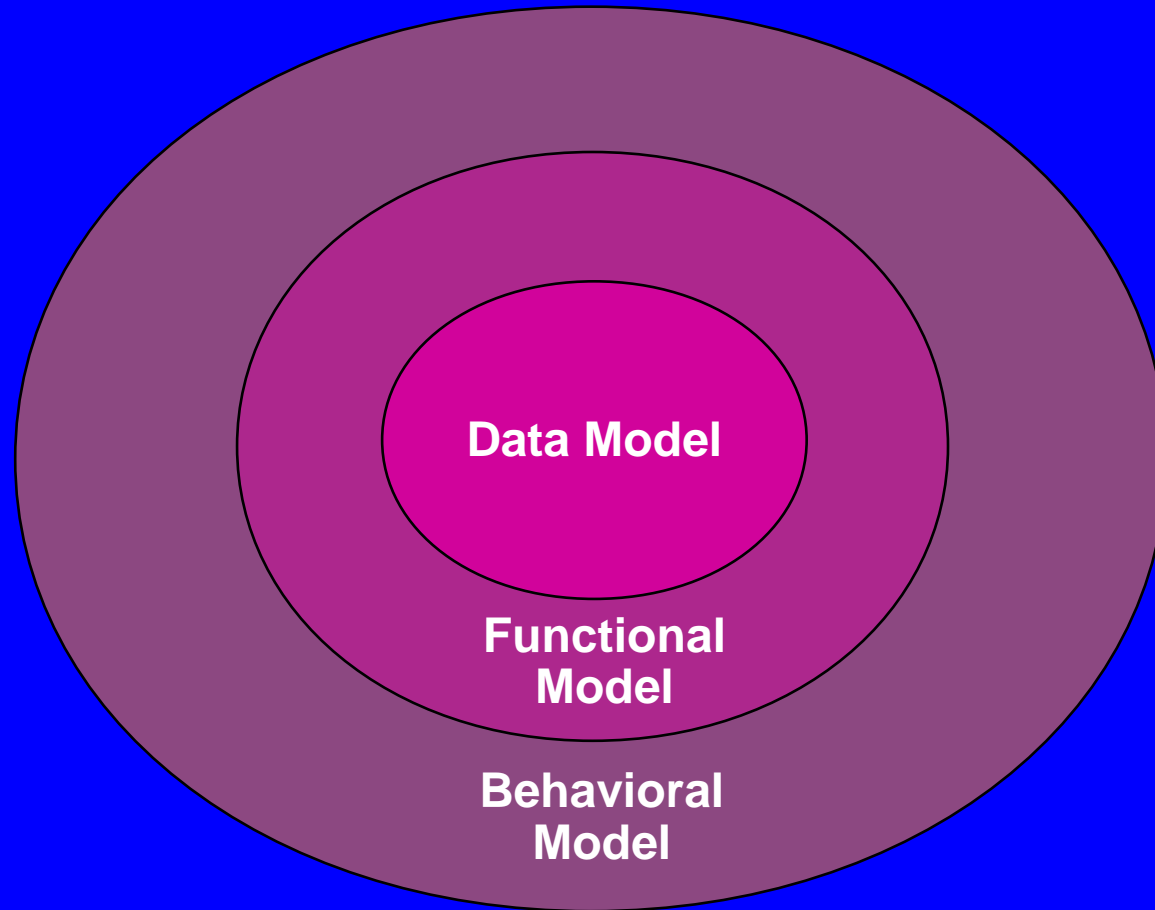
### Contents of Requirements Engineering

- Phát hiện các yêu cầu phần mềm (Requirements elicitation)
- Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (Requirements analysis and negotiation)
- Mô tả các yêu cầu phần mềm (Requirements specification)
- Mô hình hóa hệ thống (System modeling)
- Kiểm tra tính hợp lý các yêu cầu phần mềm (Requirements validation)
- Quản trị các yêu cầu phần mềm (Requirements management)

# Quy trình xác định yêu cầu phần mềm



# The Analysis Model





# 5.2.1. Phát hiện yêu cầu phần mềm

## (Requirements Elicitation)

Các vấn đề của phát hiện yêu cầu phần mềm (Problems)

- Phạm vi của phần mềm (Scope)
- Hiểu rõ phần mềm (Understanding)
- Các thay đổi của hệ thống (Volatility)

# Phương pháp phát hiện yêu cầu phần mềm

## Requirements Elicitation Methodology

- Xác định các phương pháp sử dụng phát hiện các yêu cầu phần mềm: phỏng vấn, làm việc nhóm, các buổi họp, gặp gỡ đối tác, v.v.
- Tìm kiếm các nhân sự (chuyên gia, người sử dụng) có những hiểu biết sâu sắc nhất, chi tiết nhất về hệ thống giúp chúng ta xác định yêu cầu phần mềm
- Xác định “môi trường kỹ thuật - technical environment”
- Xác định các “ràng buộc lĩnh vực domain constraints”
- Thu hút sự tham gia của nhiều chuyên gia, khách hàng để chúng ta có được các quan điểm xem xét phần mềm khác nhau từ phía khách hàng
- Thiết kế các kịch bản sử dụng của phần mềm

# Sản phẩm (output) của “phát hiện yêu cầu phần mềm”

- Bảng kê (statement) các đòi hỏi và chức năng khả thi của phần mềm
- Bảng kê phạm vi ứng dụng của phần mềm
- Mô tả môi trường kỹ thuật của phần mềm
- Bảng kê tập hợp các kịch bản sử dụng của phần mềm
- Các nguyên mẫu xây dựng, phát triển hay sử dụng trong phần mềm (nếu có)
- Danh sách nhân sự tham gia vào quá trình phát hiện các yêu cầu phần mềm - kể cả các nhân sự từ phía công ty-khách hàng

## 5.2.2. Phân tích các yêu cầu phần mềm và



# Requirements Analysis and Negotiation

- Phân loại các yêu cầu phần mềm và sắp xếp chúng theo các nhóm liên quan
- Khảo sát tỉ mỉ từng yêu cầu phần mềm trong mối quan hệ của nó với các yêu cầu phần mềm khác
- Thẩm định từng yêu cầu phần mềm theo các tính chất: phù hợp, đầy đủ, rõ ràng, không trùng lặp

# Requirements Analysis and Negotiation

- Phân cấp các yêu cầu phần mềm theo dựa trên nhu cầu và đòi hỏi khách hàng / người sử dụng
- Thẩm định từng yêu cầu phần mềm để xác định chúng có khả năng thực hiện được trong môi trường kỹ thuật hay không, có khả năng kiểm định các yêu cầu phần mềm hay không
- Thẩm định các rủi ro có thể xảy ra với từng yêu cầu phần mềm

# Requirements Analysis and Negotiation

- **Đánh giá thô (tương đối) về giá thành và thời gian thực hiện của từng yêu cầu phần mềm trong giá thành sản phẩm phần mềm và thời gian thực hiện phần mềm**
- **Giải quyết tất cả các bất đồng về yêu cầu phần mềm với khách hàng / người sử dụng trên cơ sở thảo luận và thương lượng các yêu cầu đề ra**

## 5.2.3. Đặc tả yêu cầu phần mềm

- Đặc tả các yêu cầu phần mềm là công việc xây dựng các tài liệu đặc tả, trong đó có thể sử dụng tới các công cụ như: mô hình hóa, mô hình toán học hình thức (a formal mathematical model), tập hợp các kịch bản sử dụng, các nguyên mẫu hoặc bất kỳ một tổ hợp các công cụ nói trên
- Chất lượng của hồ sơ đặc tả đánh giá qua các tiêu thức
  - Tính rõ ràng, chính xác
  - Tính phù hợp
  - Tính đầy đủ, hoàn thiện



# Requirements Specification

- Các thành phần của hồ sơ đặc tả
  - Đặc tả phi hình thức (Informal specifications) được viết bằng ngôn ngữ tự nhiên
  - Đặc tả hình thức (Formal specifications) được viết bằng tập các ký pháp có các quy định về cú pháp (*syntax*) và ý nghĩa (*semantic*) rất chặt chẽ
  - Đặc tả vận hành chức năng (Operational specifications) mô tả các hoạt động của hệ thống phần mềm sẽ xây dựng
  - Đặc tả mô tả (Descriptive specifications) – đặc tả các đặc tính đặc trưng của phần mềm

# Requirements Specification

- **Đặc tả chức năng (Operational Specifications):** thông thường khi đặc tả các chức năng của phần mềm người ta sử dụng các công cụ tiêu biểu sau
  - **Biểu đồ luồng dữ liệu (Data Flow Diagrams)**
  - **Máy trạng thái hữu hạn (Finite State Machines)**
  - **Mạng Petri (Petri nets)**

# Requirements Specification

- **Đặc tả mô tả (Descriptive Specifications)**
  - **Biểu đồ thực thể liên kết (Entity-Relationship Diagrams)**
  - **Đặc tả Logic (Logic Specifications)**
  - **Đặc tả đại số (Algebraic Specifications)**

# Biểu đồ luồng dữ liệu (DFD)

- **Hệ thống (System):** tập hợp các dữ liệu (data) được xử lý bằng các chức năng tương ứng (functions)
- **Các ký pháp sử dụng:**

**Thể hiện các chức năng (functions)**



**Thể hiện luồng dữ liệu**



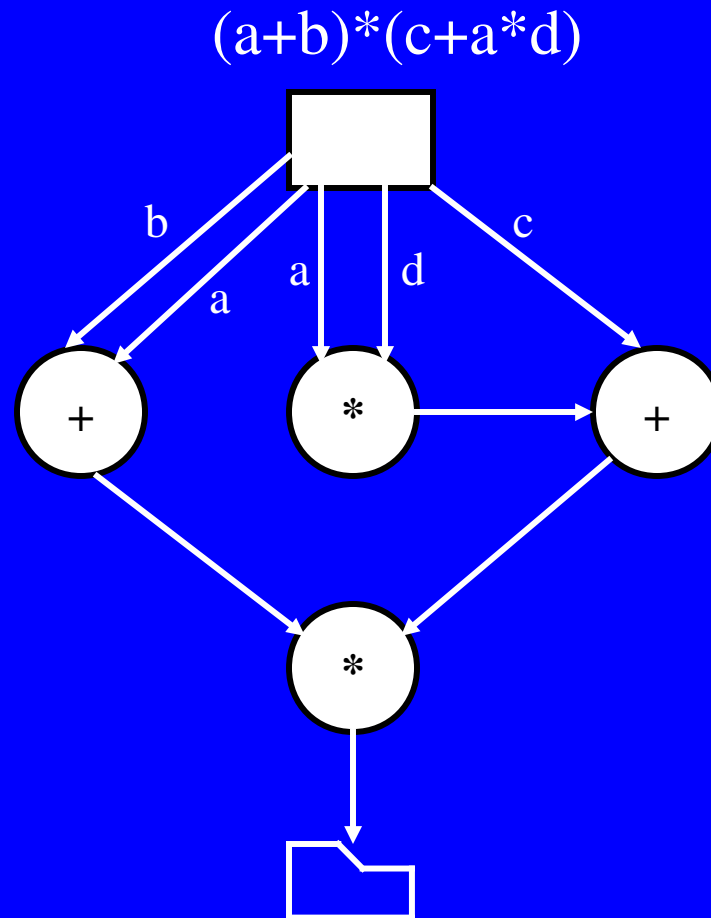
**Kho dữ liệu**



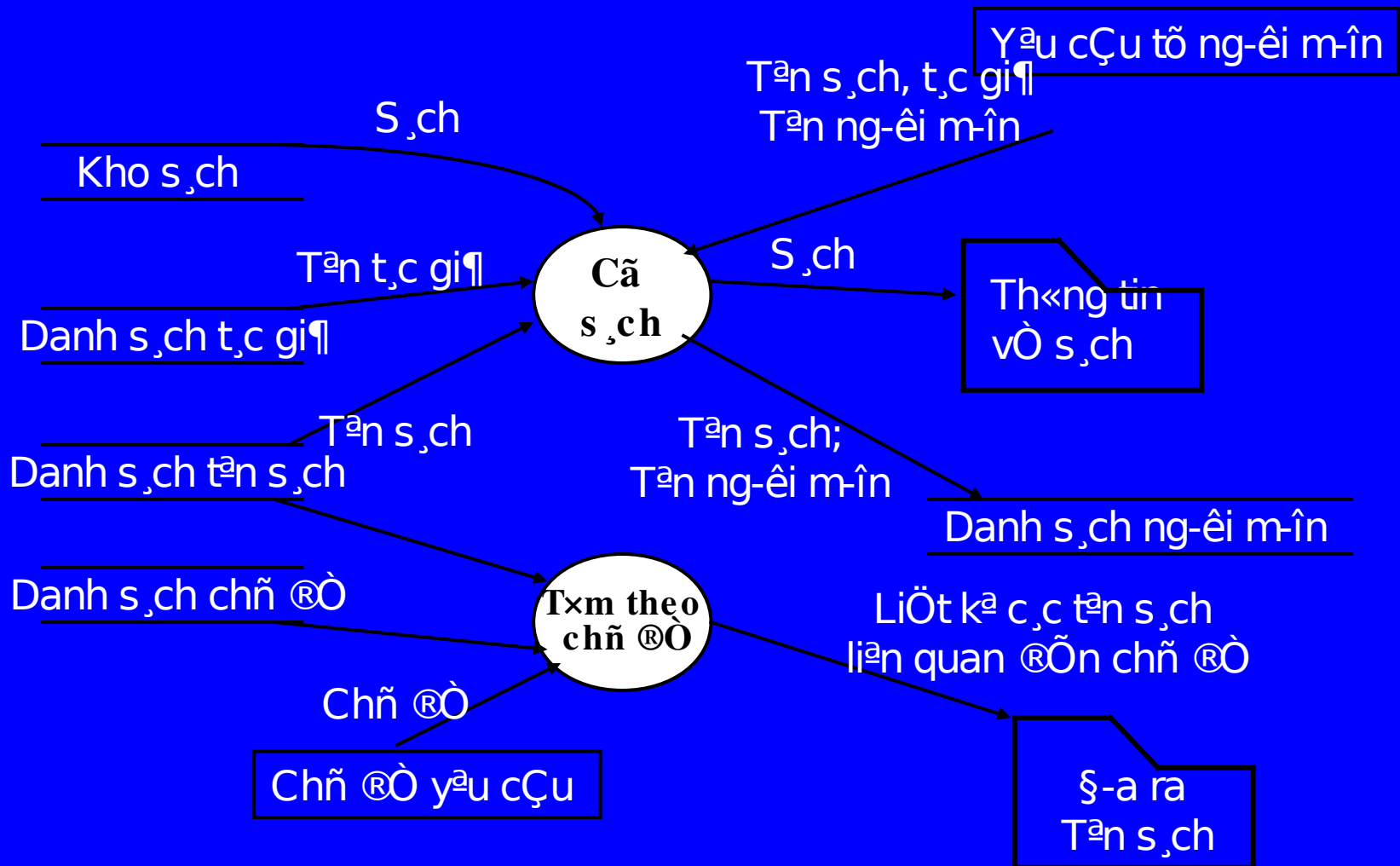
**Vào ra dữ liệu và tương tác giữa hệ thống và người sử dụng**



# Ví dụ mô tả biểu thức toán học bằng DFD



# Ví dụ đặc tả các chức năng của thư viện qua DFD



# Các hạn chế của DFD

- ý nghĩa của các ký pháp sử dụng được xác định bởi các định danh lựa chọn của NSD

Ví dụ của chức năng tìm kiếm:

**If** NSD nhập vào cả tên tác giả và tiêu đề sách **Then**  
tìm kiếm sách tương ứng, không có thì thông báo lỗi

**Elseif** chỉ nhập tên tác giả **Then**

hiển thị danh sách các sách tương ứng với

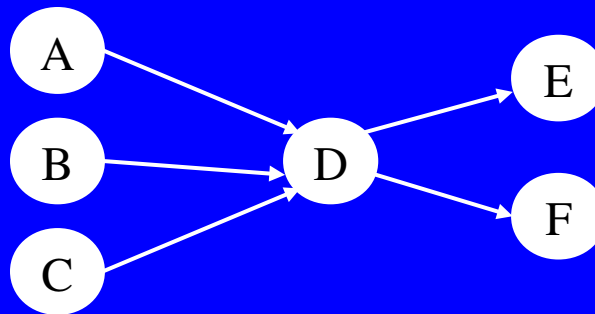
tên tác giả đã nhập và yêu cầu NSD lựa chọn sách

**Elseif** chỉ nhập tiêu đề sách **Then**

...

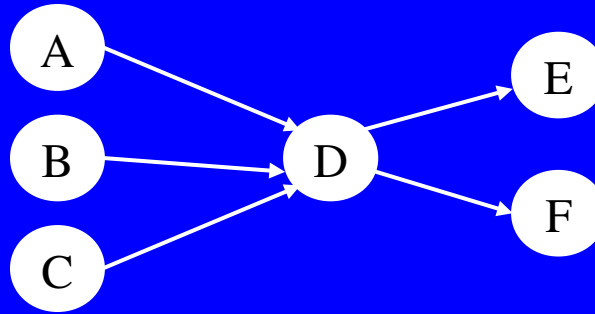
**Endif**

- Trong DFD không xác định rõ các hướng thực hiện (control aspects)

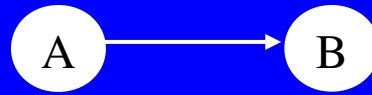


**Biểu đồ DFD này không chỉ rõ đầu vào là gì để thực hiện chức năng D và đầu ra là gì sau khi thực hiện chức năng D.**





- Chức năng D có thể cần cả A, B và C
- Chức năng D có thể chỉ cần một trong A, B và C để thực hiện
- Chức năng D có thể kết xuất kết quả cho một trong E và F
- Chức năng D có thể kết xuất kết quả chung cho cả E và F
- Chức năng D có thể kết xuất kết quả riêng cho cả E và F

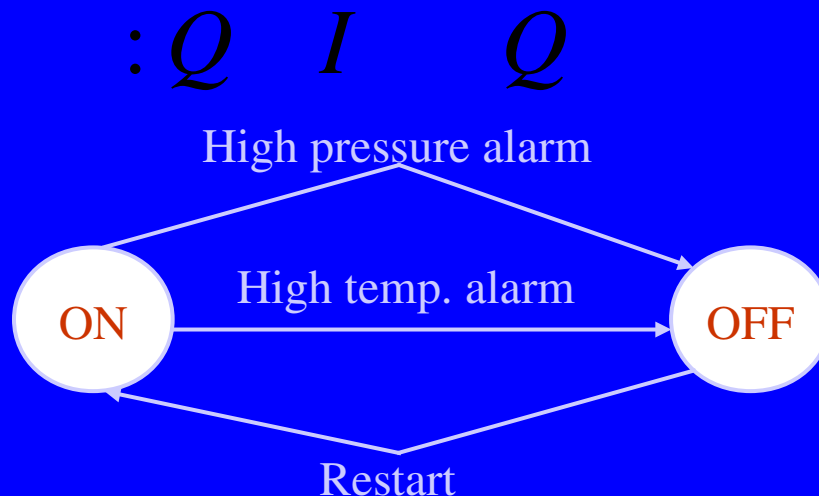


- **DFD không xác định sự đồng bộ giữa các chức năng / mô-đun**
  - **A xử lý dữ liệu và B được hưởng (nhận) các kết quả được xử lý từ A**
  - **A và B là các chức năng không đồng bộ (asynchronous activities) vì thế cần có buffer để ngăn chặn tình trạng mất dữ liệu**

# Finite State Machines (FSM)

FSM chứa

- Tập hữu hạn các trạng thái  $Q$
- Tập hữu hạn các đầu vào  $I$
- Các chức năng chuyển tiếp



# Đặc tả các yêu cầu phần mềm bằng FSM

- Xem xét ví dụ về thư viện với các giao dịch như sau:
  - Mượn sách / Trả sách
  - Thêm đầu sách / Loại bỏ đầu sách
  - Liệt kê danh sách các đầu sách theo tên tác giả hay theo chủ đề
  - Tìm kiếm sách theo các yêu cầu của người mượn
  - Tìm kiếm sách quá hạn trả, . . .

# Đặc tả . . .

- Các yêu cầu đặc biệt của thư viện:
  - Độc giả không được mượn quá một số lượng sách nhất định, trong một thời gian nhất định
  - Một số sách không được mượn về
  - Một số người không được mượn một số loại sách nào đó, . . .

## Các đối tượng –

Tên sách

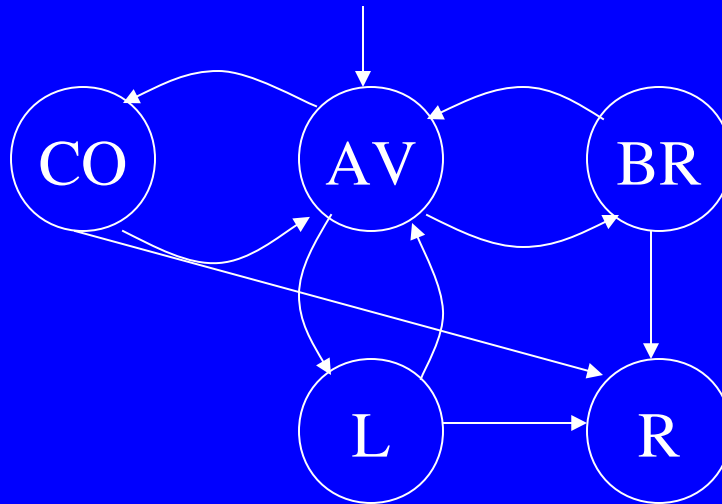
Mã quyển

Nhân viên phục vụ

Người mượn

- Chúng ta cần có tập hợp (danh sách) các tiêu đề sách, danh sách các tác giả cho từng quyển sách, danh sách các chủ đề liên quan của các quyển sách
- Ta có tập hợp các sách (mỗi đầu sách có thể có nhiều quyển sách trong thư viện). Mỗi quyển sách có thể có 1 trong 5 trạng thái sau:
- (AV) - Available được phép mượn, (CO) - (BR) - đã mượn (Check Out; Borrow), (L): Last, (R): Remove

- **FSM đặc tả các trạng thái**



- ii. Có thể có hạn chế về số sách được mượn cho 1 nhóm độc giả hoặc mọi độc giả, . . .

# Mô hình đặc tả :

## Mô hình thực thể liên kết

- Mô hình khái niệm cho phép đặc tả các yêu cầu logic của hệ thống, thường được sử dụng trong các hệ thống dữ liệu lớn

### ER Model

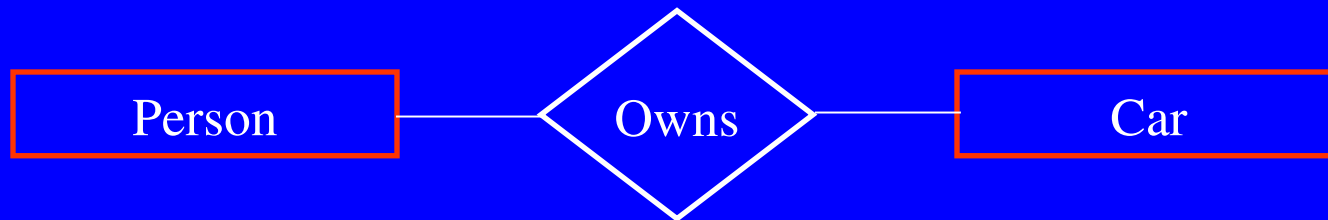
- Thực thể
- Quan hệ
- Thuộc tính

### Biểu đồ thực thể

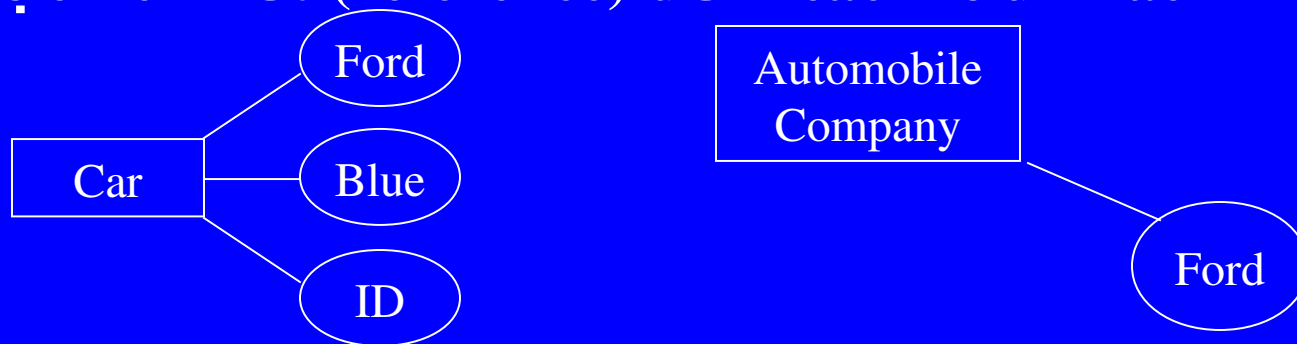


- **Thực thể** – tập hợp các thông tin liên quan cần được xử lý trong phần mềm

Thực thể có thể có mối quan hệ: – **person owns car**

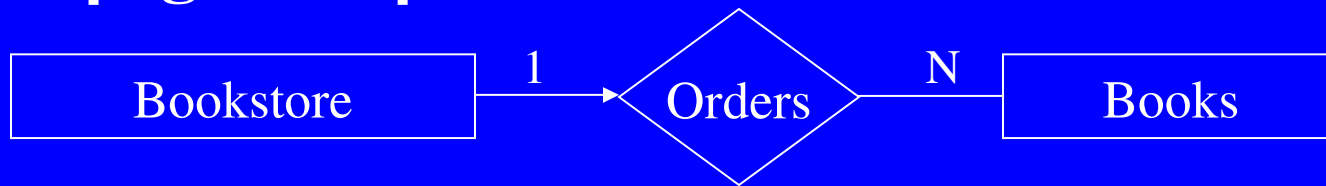


- **Thực thể có các thuộc tính**
- **Thuộc tính: Tính chất của một thực thể hoặc một đối tượng dữ liệu**
  - **đặt tên cho 1 mẫu (instance) của đối tượng dữ liệu**
  - **mô tả mẫu (instance)**
  - **tạo liên kết (reference) đến các mẫu khác**

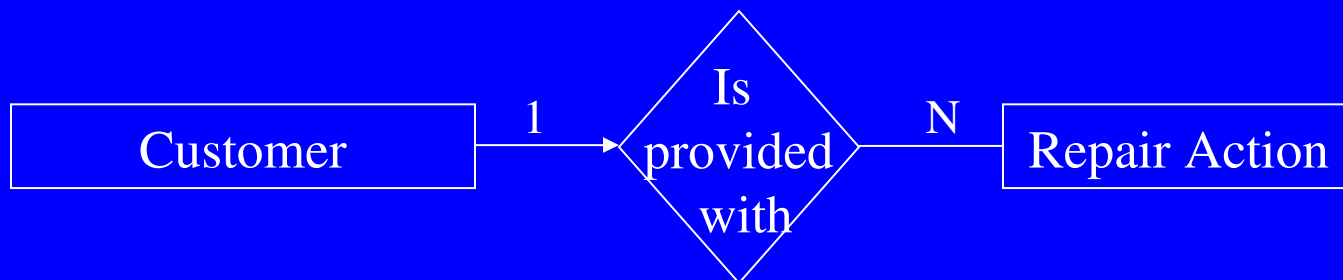


Tập các thuộc tính của 1 đối tượng dữ liệu được xác định thông qua ngữ cảnh của bài toán

- **Quan hệ – chỉ ra mối liên quan giữa các đối tượng dữ liệu**



- **Cardinality** : chỉ ra định lượng của mỗi quan hệ  
**1:1** one-to-one   **1:N** one-to-many   **M:N** many-to-many
- **Modality** : **0** – có thể có, có thể không có quan hệ  
**1** – bắt buộc có quan hệ



# Ví dụ ERD mô tả thư viện



# Các yêu cầu của một đặc tả tốt

- Dễ hiểu với người dùng
- Có ít điều nhập nhằng
- Có ít quy ước khi mô tả, có thể tạo đơn giản
- Với phong cách từ trên xuống (topdown)
- Dễ triển khai cho những pha sau của vòng đời: thiết kế hệ thống và thiết kế chương trình và giao diện dễ làm, đảm bảo tính nhất quán, . . .

## 5.3. Các nguyên lý phân tích yêu cầu sử dụng

- **Nguyên lý I. Mô hình hóa dữ liệu**
  - Xác định các đối tượng dữ liệu
  - Xác định các đặc tính của các đối tượng dữ liệu
  - Thiết lập các mối quan hệ giữa các đối tượng dữ liệu

# Các nguyên lý phân tích yêu cầu sử dụng

- **Nguyên lý II. Mô hình hóa các chức năng**
  - Xác định các chức năng chuyển đổi đối tượng dữ liệu
  - Chỉ ra luồng dữ liệu đi qua hệ thống như thế nào
  - Biểu diễn bộ phận sản sinh dữ liệu và bộ phận tiêu thụ dữ liệu

# Các nguyên lý phân tích yêu cầu sử dụng

- **Nguyên lý III. Mô hình hóa hành vi**
  - Chỉ ra các trạng thái (states) khác nhau của hệ thống
  - Đặc tả các hiện tượng (events) làm hệ thống thay đổi trạng thái



# Các nguyên lý phân tích yêu cầu sử dụng

- **Nguyên lý IV.** *Partition the Models*

Tinh lọc từng mô hình để biểu diễn các mức trừu tượng thấp hơn

- Lọc đối tượng dữ liệu
- Tạo ra phân cấp chức năng
- Biểu diễn hành vi (behavior) ở các mức chi tiết khác nhau

# Các nguyên lý phân tích yêu cầu sử dụng

- Nguyên lý V. *Bản chất (Essence)*

Hãy bắt đầu bằng cách tập trung vào bản chất của vấn đề chứ không xem xét những chi tiết cài đặt (begin by focusing on the essence of the problem without regard to implementation details)

**New Tool: Unified Modeling Language (UML)!**