

25  
SOICT

YEARS ANNIVERSARY

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Nhập môn Công nghệ Phần mềm

(Introduction to Software Engineering)

# CHƯƠNG 6\_2

## Mô hình các yêu cầu phần mềm

# Mục tiêu của bài học

Sinh viên sẽ được trang bị các kiến thức sau:

- Các khái niệm liên quan tới mô hình yêu cầu phần mềm
- Giới thiệu về một số mô hình yêu cầu phần mềm
- Áp dụng mô hình yêu cầu phần mềm trong ứng dụng WebApps

# Nội dung

1. Tổng quan – phân loại
2. Một số mô hình thực tế
3. Mô hình cho mô hình các yêu cầu
4. Mô hình yêu cầu cho ứng dụng WebApps

# 1. Tổng quan - phân loại

- Mô hình yêu cầu phần mềm được gọi là *structured analysis (phân tích có cấu trúc)*: xem xét, phân tích yêu cầu theo dữ liệu giữa các tiến trình và sự thay đổi dữ liệu như các thực thể riêng biệt.
  - Data objects (dữ liệu đối tượng) được mô hình bằng cách định nghĩa các trạng thái và quan hệ giữa chúng.
  - Các tiến trình thao tác đối với dữ liệu đối tượng được mô hình để cho thấy làm thế nào biến đổi dữ liệu như dữ liệu đối tượng thông qua hệ thống.
- Mô hình yêu cầu phần mềm được gọi là object-oriented analysis (*phân tích hướng đối tượng*), tập trung vào:
  - Định nghĩa các lớp
  - Cách thức mà chúng liên kết với nhau để thực hiện các yêu cầu của khách hàng.

## 2. Một số mô hình thực tế

### a. Flow-Oriented Modeling – Biểu đồ luồng

- Trình diễn làm thế nào dữ liệu đối tượng được biến đổi thông qua hệ thống **data flow diagram (biểu đồ luồng dữ liệu) (DFD)**
- Được nhiều người nhìn nhận là một cách tiếp cận theo trường phái cũ, tuy nhiên nó vẫn tiếp cung cấp một khía cạnh của hệ thống nên vẫn được sử dụng để bổ sung cho các mô hình phân tích khác.

# The Flow Model

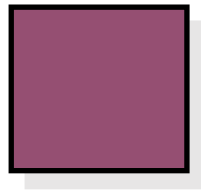
- Mỗi hệ thống máy tính cơ bản là một hệ thống biến đổi thông tin



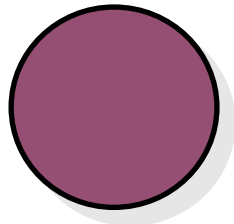


# Các ký hiệu mô hình luồng Dữ Liệu

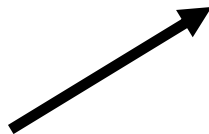
- Mỗi hệ thống máy tính cơ bản là một hệ thống biến đổi thông tin



**Thực thể bên ngoài**



**Tiến trình**



**Luồng dữ liệu**



**Kho dữ liệu**

# Thực thể ngoài

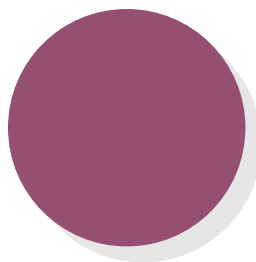
**Một sản xuất hoặc tiêu thụ dữ liệu**

*Ví dụ: một người, một thiết bị, một cảm biến...*

*Ví dụ khác: hệ thống máy tính cơ bản*

*Dữ liệu luôn luôn phải xuất phát từ một nơi nào đó  
và luôn luôn phải được gửi đến một cái gì đó.*

# Tiến trình



**Một sự biến đổi dữ liệu (thay đổi đầu vào qua đầu ra)**

*Ví dụ: tính thuế, xác định diện tích,  
Định dạng báo cáo, hiển thị đồ thị*

*Dữ liệu phải luôn luôn được xử lý bằng cách  
nào đó để đạt được chức năng của hệ thống*

# Luồng dữ liệu

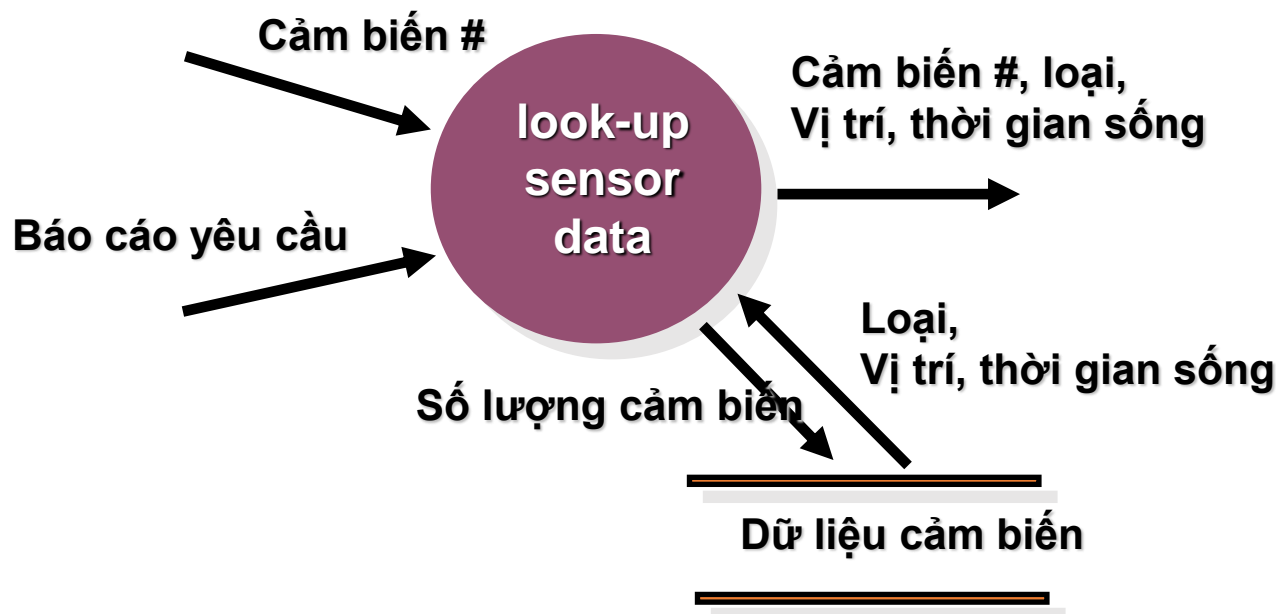


**Luồng dữ liệu thông qua một hệ thống, bắt đầu như đầu vào và biến đổi thành đầu ra.**



# Các kho dữ liệu

**Dữ liệu thường được lưu cho lần sử dụng sau**



# Biểu đồ luồng dữ liệu: hướng dẫn

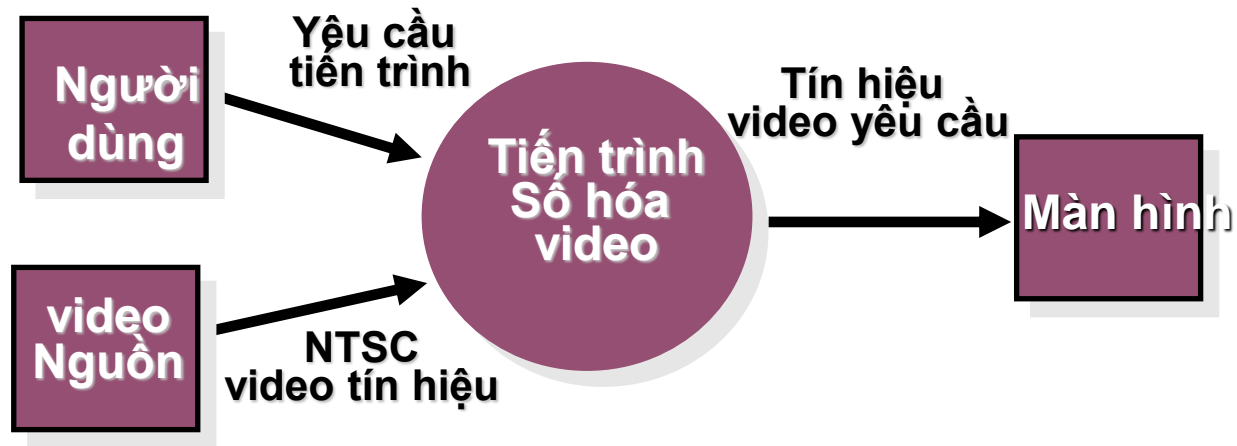
- Tất cả các biểu tượng phải được gán nhãn có nghĩa đầy đủ
- Biểu đồ DFD phát triển thông qua số lượng mức các chi tiết
- Luôn luôn bắt đầu với một mức nội dung (gọi là level 0)
- Luôn luôn chỉ ra các thực thể ngoài tại mức 0
- Luôn luôn gán nhãn cho hướng luồng dữ liệu
- Không thể hiện các thủ tục logic

# Ví dụ: hướng dẫn xây dựng một DFD

- Xem xét kịch bản người dùng và/hoặc để tách biệt các dữ liệu đối tượng và sử dụng một phân tích ngữ pháp để xác định “hoạt động”
- Xác định thực thể ngoài (các sản xuất và tiêu thụ của dữ liệu)
- Tạo mức 0 DFD



# Ví dụ mức 0 DFD



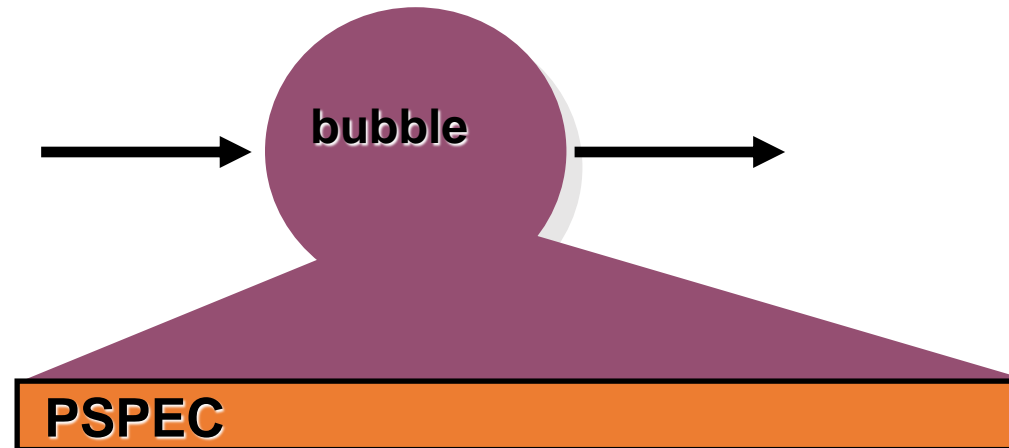


# Các ghi chú

- Mỗi ô tròn là đơn giá trị cho đến khi nó làm gì đấy
- Tỷ lệ mở rộng giảm khi số lượng mức tăng
- Phần lớn hệ thống yêu cầu giữa 3 và 7 cấp cho một mô hình đầy đủ luồng
- Một mục lưu lượng dữ liệu duy nhất (mũi tên) có thể được mở rộng như cấp độ tăng (dữ liệu từ điển cung cấp thông tin)

## 2. Một số mô hình thực tế

### b. Đặc tả tiến trình (PSPEC)



- ☐ Kịch bản
- ☐ Mã giả(PDL)
- ☐ Phương trình
- ☐ Các bảng
- ☐ Biểu đồ và/hoặc

# Mô hình luồng Điều khiển

- Biểu diễn các “**sự kiện**” và các tiến trình mà quản lý các sự kiện
- Một “sự kiện” là một điều kiện đúng/sai mà có thể chắc chắn bởi:
  - Nghe tất cả các cảm biến mà “**đọc**” bởi phần mềm
  - Nghe tất cả các ngắt điều kiện.
  - Nghe tất cả các “ngắt” mà được dẫn động bởi hệ thống.
  - Nghe tất cả các dữ liệu điều kiện.
  - Nhắc lại phân tích danh từ/động từ được áp dụng cho kịch bản sản xuất, xem xét tất cả "kiểm soát mục" như có thể CSPEC đầu vào/đầu ra..

# Đặc tả Điều khiển (CSPEC)

*CSPEC có thể là:*

- Biểu đồ trạng thái  
(biểu đồ tuần tự)
- Bảng chuyển đổi  
trạng thái
- Bảng quyết định
- Bảng kích hoạt

*Tổ hợp*

## 2. Một số mô hình thực tế

### c. Mô hình hành vi

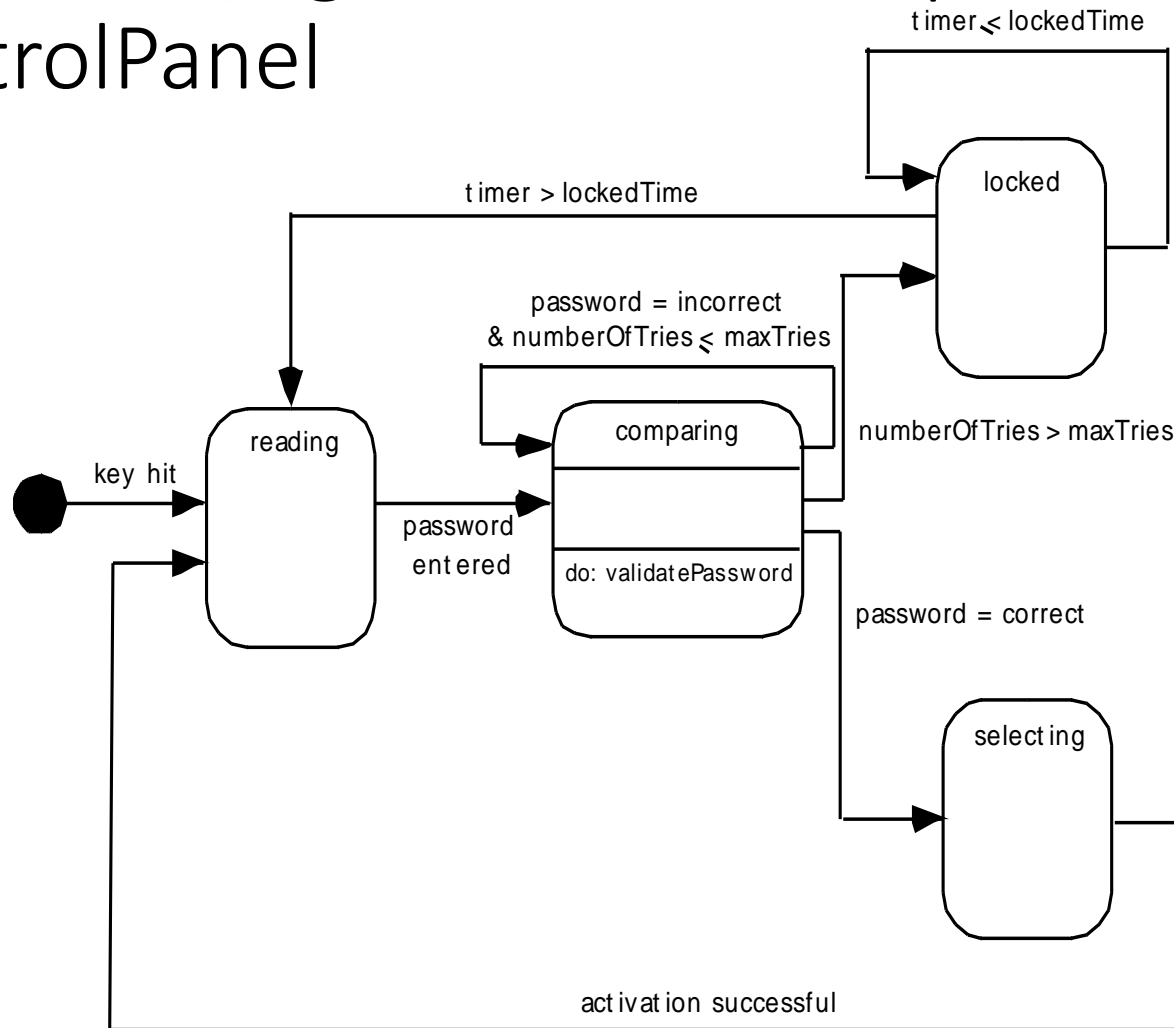
- Mô hình hành vi chỉ ra cách phần mềm sẽ phản ứng với các sự kiện hoặc kích thích bên ngoài (ví dụ biểu đồ User-case). Để tạo ra các mô hình, phải thực hiện theo các bước sau:
  - Đánh giá tất cả users-case để hiểu đầy đủ trình tự của các tương tác trong hệ thống.
  - Xác định sự kiện điều khiển dãy tương tác và hiểu làm thế nào những sự kiện liên quan đến các đối tượng cụ thể.
  - Tạo ra chuỗi cho mỗi use-case.
  - Xây dựng một biểu đồ trạng thái cho hệ thống.
  - Xem lại các mô hình hành vi để xác minh tính chính xác và nhất quán.

## 2. Một số mô hình thực tế

### Biểu diễn trạng thái

- Trong bối cảnh của mô hình hành vi, 2 đặc điểm khác nhau của trạng thái phải được xem xét:
  - Trạng thái của mỗi lớp
  - trạng thái của hệ thống (quan sát từ bên ngoài hệ thống thực hiện chức năng của nó)
- Trạng thái của một lớp có cả hai đặc tính chủ động và thụ động.
  - Một trạng thái thụ động chỉ đơn giản là tình trạng hiện tại của tất cả các thuộc tính của một đối tượng.
  - Các trạng thái hoạt động của một đối tượng cho biết tình trạng hiện tại của đối tượng như nó trải qua một sự biến đổi hoặc qua hoạt động của 1 tiến trình.

# Biểu đồ trạng thái cho các lớp ControlPanel



# Các trạng thái của một hệ thống

- **Trạng thái**—1 tập của các tình huống mà đặc trưng cho hành vi của 1 hệ thống tại một thời điểm nhất định
- **Chuyển đổi trạng thái**—sự di chuyển từ một trạng thái tới trạng thái khác.
- **Sự kiện**—Một biến cố mà nhờ đó hệ thống có thể dự đoán được hành vi
- **Hành động**—tiến trình mà xuất hiện như một chuỗi các biến đổi



# Ví dụ: Mô hình hành vi

- Tạo ra một danh sách của các trạng thái khác nhau của hệ thống (làm thế nào để hệ thống xử lý các hành vi?)
- Chỉ ra cách hệ thống làm cho một trạng thái chuyển đổi từ một này sang một trạng thái khác (Làm thế nào hệ thống chuyển đổi trạng thái?)
  - Chỉ ra sự kiện
  - Chỉ ra hành động
- Vẽ một biểu đồ trạng thái hoặc một biểu đồ trình tự

# Biểu đồ trình tự

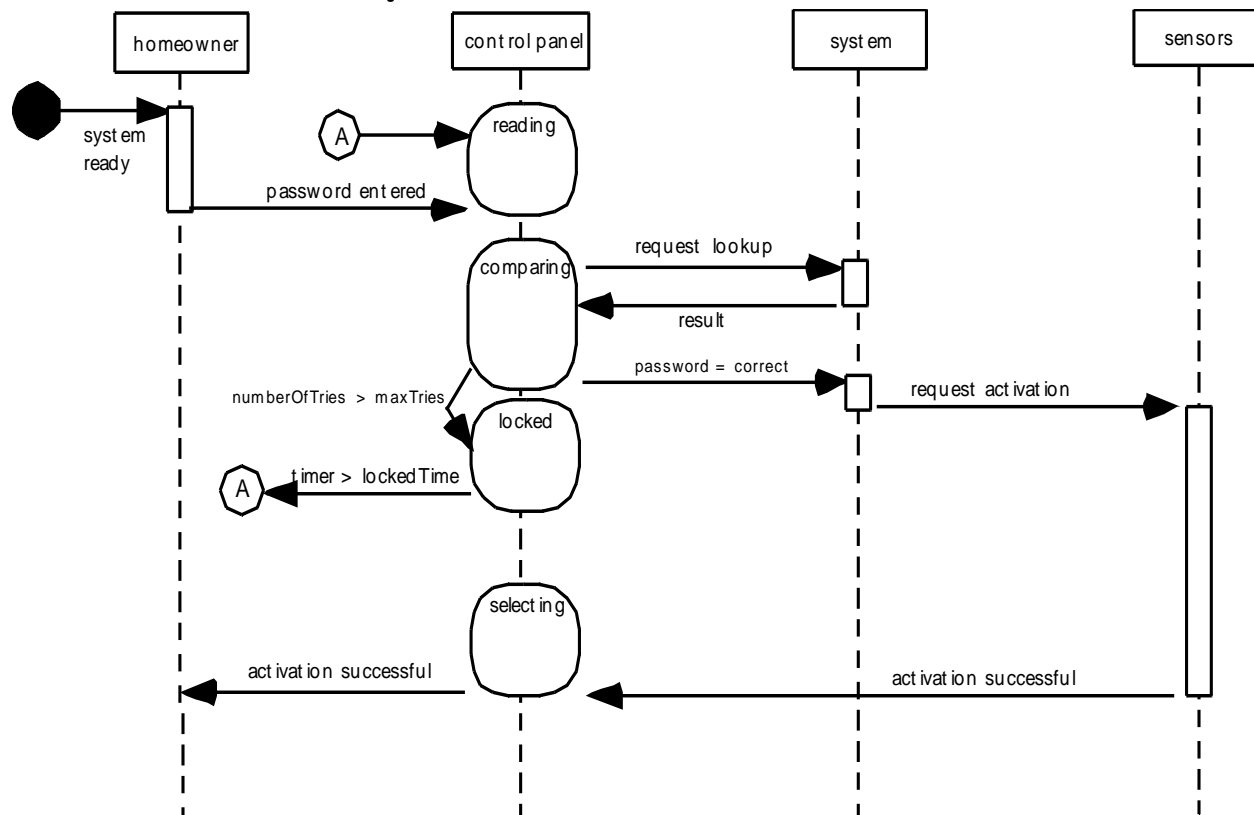


Figure 8.27 Sequence diagram (partial) for SafeHome security function

# Viết các thông số phần mềm



### 3. Mô hình cho mô hình các yêu cầu

- Mô hình phần mềm là một cơ chế để thu nhận tri thức mà cho phép nó được áp dụng lại khi xuất hiện một vấn đề mới
  - Miền kiến thức có thể được áp dụng cho một vấn đề mới trong cùng miền ứng dụng .
  - các miền kiến thức được lưu lại bởi một mô hình có thể được áp dụng bằng cách tương tự vào một miền ứng dụng khác mà hoàn toàn khác nhau.
- Ban đầu, các tác giả của một mô hình phân tích không "tạo ra" các mô hình, nhưng sau đó, họ phát hiện ra đó là công việc yêu cầu kỹ thuật cần được tiến hành.
- Khi mô hình đã được khảo sát, nó đã được chứng thực

# Khảo sát mô hình phân tích

- Các yếu tố cơ bản nhất trong mô tả của một mô hình yêu cầu là các use-case.
- Một tập hợp chặt chẽ các use-case có thể phục vụ là cơ sở cho việc phát hiện một hoặc nhiều các mẫu phân tích thêm.
- Một mô hình phân tích ngữ nghĩa (SAP) "là một mô hình mô tả một tập hợp nhỏ các use-case mạch lạc theo cùng mô tả một ứng dụng cơ bản."

# Một ví dụ

- Hãy xem xét các use-case sơ bộ sau đây cho phần mềm cần thiết để kiểm soát và giám sát một máy ảnh thực sự xem và cảm biến khoảng cách cho một chiếc điện thoại tự động :

**Use case:** Theo dõi chuyển động ngược

**Mô tả:** Khi chiếc xe được đặt tại số lùi, các phần mềm điều khiển cho phép một nguồn cấp dữ liệu video từ một máy quay phim phía sau được đặt để hiển thị bảng điều khiển. Phần mềm điều khiển tính toán các khoảng cách và định hướng trên màn hình hiển thị để các nhà điều hành xe có thể duy trì định hướng khi xe di chuyển theo chiều ngược lại. Phần mềm điều khiển cũng theo dõi một cảm biến để xác định xem một đối tượng nằm trong 10 feet phía sau của chiếc xe. Nó sẽ tự động dừng chiếc xe nếu các cảm biến khoảng cách chỉ ra một đối tượng trong 3 feet phía sau của xe.

# Một ví dụ

- Use-case này với một loạt các chức năng đó sẽ được phân tích và xây dựng (thành một bộ mạch lạc của use-case) trong quá trình thu thập yêu cầu và mô hình hóa.
- Bất kể bao nhiêu xây dựng được thực hiện, use-case(s) đề nghị một SAP-giám sát dựa trên phần mềm đơn giản, chưa áp dụng rộng rãi và kiểm soát các cảm biến và cơ cấu chấp hành trong một hệ thống vật lý.
- Trong trường hợp này, các cảm biến cung cấp thông tin xung quanh khoảng cách và thông tin video. Các "thiết bị truyền động" là hệ thống dừng của xe (gọi nếu một đối tượng là rất gần với xe).
- Nhưng trong trường hợp tổng quát hơn, một mô hình được áp dụng rộng rãi được phát hiện -> (Thiết bị truyền động-cảm biến) **Actuator-Sensor**

## 4. Mô hình yêu cầu cho ứng dụng WebApps

- **Phân tích nội dung:** đầy đủ các nội dung được cung cấp bởi các WebApp (ứng dụng Web) được xác định, bao gồm cả văn bản, đồ họa và hình ảnh, video, âm thanh và dữ liệu. Mô hình hóa dữ liệu có thể được sử dụng để xác định và mô tả một trong các dữ liệu đối tượng.
- **Phân tích Tương tác:** Cách thức mà người dùng tương tác với các WebApp được mô tả chi tiết. Biểu đồ ca sử dụng(Use-Case) có thể được phát triển để cung cấp mô tả chi tiết của tương tác này.
- **Phân tích chức năng:** Các use-case được tạo ra như một phần của phân tích tương tác xác định các hoạt động mà sẽ được áp dụng cho nội dung WebApp và bao hàm các chức năng xử lý khác. Tất cả các hoạt động và chức năng được mô tả chi tiết.
- **Phân tích cấu hình:** Môi trường và cơ sở hạ tầng, trong đó vị trí WebApp được mô tả chi tiết.



# Khi nào chúng ta tiến hành phân tích?

- Trong một số tình huống phân tích và thiết kế hợp nhất. Tuy nhiên, **một phân tích hoạt động rõ ràng xuất hiện khi:**
  - các WebApp sẽ được xây dựng nhiều và/hoặc phức tạp.
  - số lượng các bên liên quan là lớn
  - số lượng kỹ sư Web và đóng góp khác quá lớn
  - các mục tiêu và đối tượng (được xác định trong quá trình lập) cho các WebApp sẽ ảnh hưởng đến “kinh doanh ”
  - sự thành công của các WebApp sẽ có một ảnh hưởng mạnh mẽ đến sự thành công của doanh nghiệp

# Mô hình nội dung

- **Đối tượng nội dung** được lấy từ biểu đồ ca sử dụng
  - kiểm tra các mô tả kịch bản để tham khảo trực tiếp và gián tiếp đến nội dung
- **Các trạng thái** của một đối tượng nội dung được định nghĩa
- Các mối quan hệ giữa các đối tượng nội dung và/hoặc, hệ thống phân cấp của nội dung được duy trì bởi một WebApp
  - Sơ đồ mối quan hệ thực thể-mối quan hệ hoặc UML
  - Cây phân cấp dữ liệu hoặc UML

# Cây dữ liệu

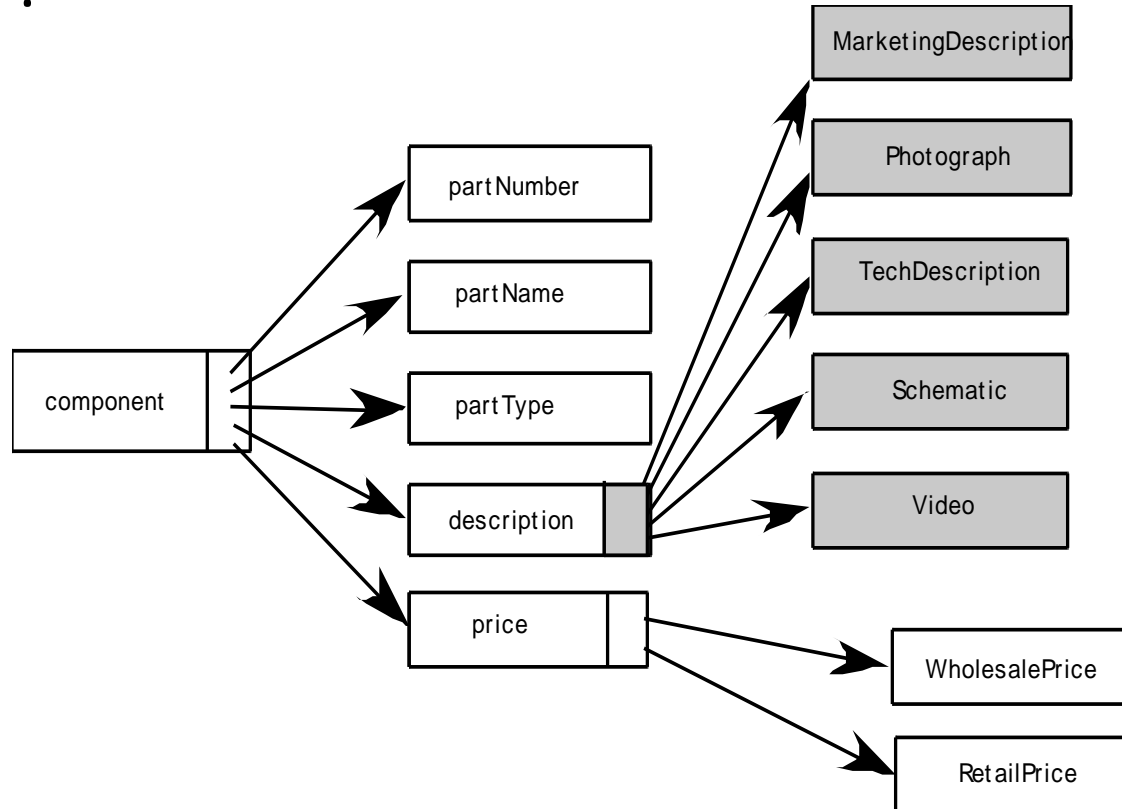


Figure 18.3 Data tree for *aSafeHome* component

# Mô hình tương tác

- Bốn nguyên tố cấu thành:
  - use-cases(biểu đồ ca sử dụng)
  - sequence diagrams(biểu đồ trình tự)
  - state diagrams(biểu đồ trạng thái)
  - a user interface prototype(giao diện người dùng)
- Mỗi trong số này là một nguyên tố UML quan trọng và được mô tả trong Phụ lục I

# Biểu đồ trình tự

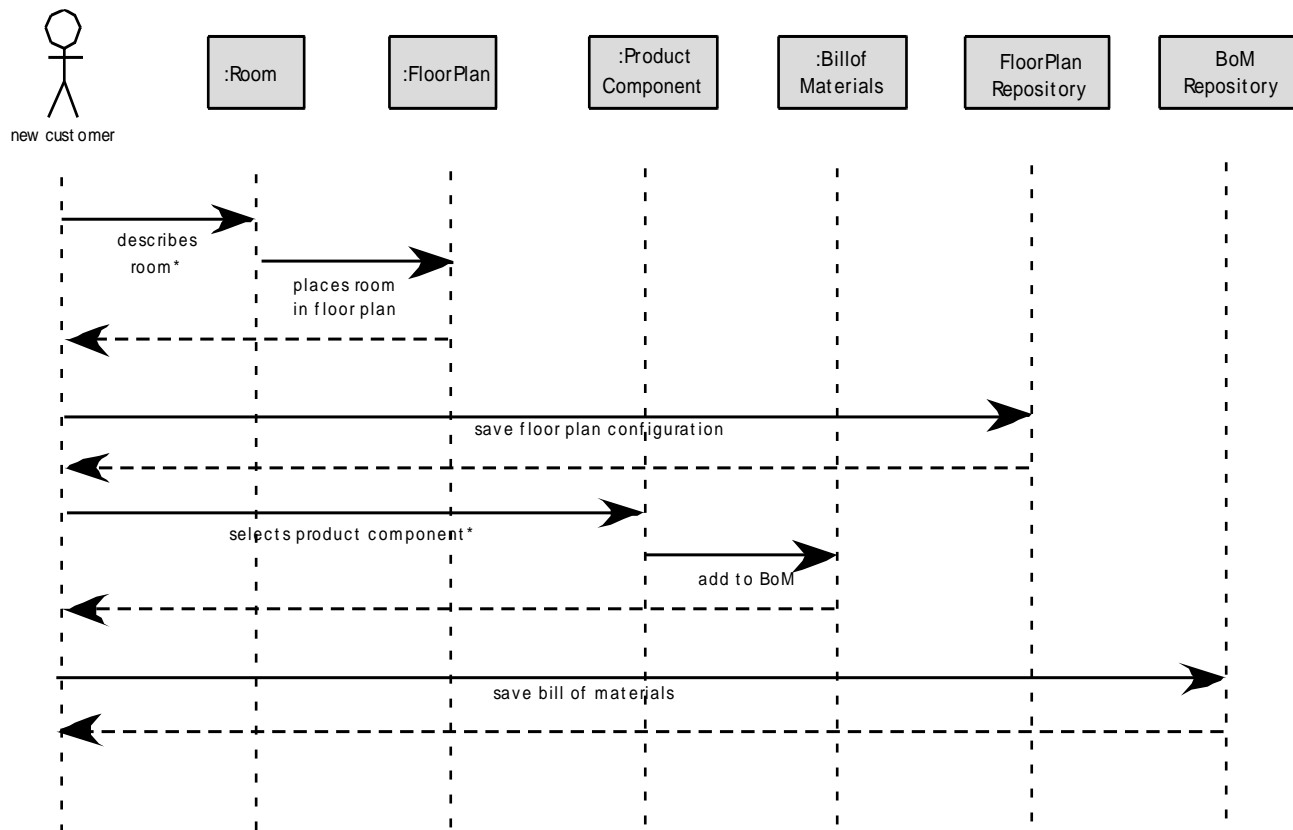


Figure 18.5 Sequence diagram for use-case: *select SafeHome components*

# Biểu đồ trạng thái

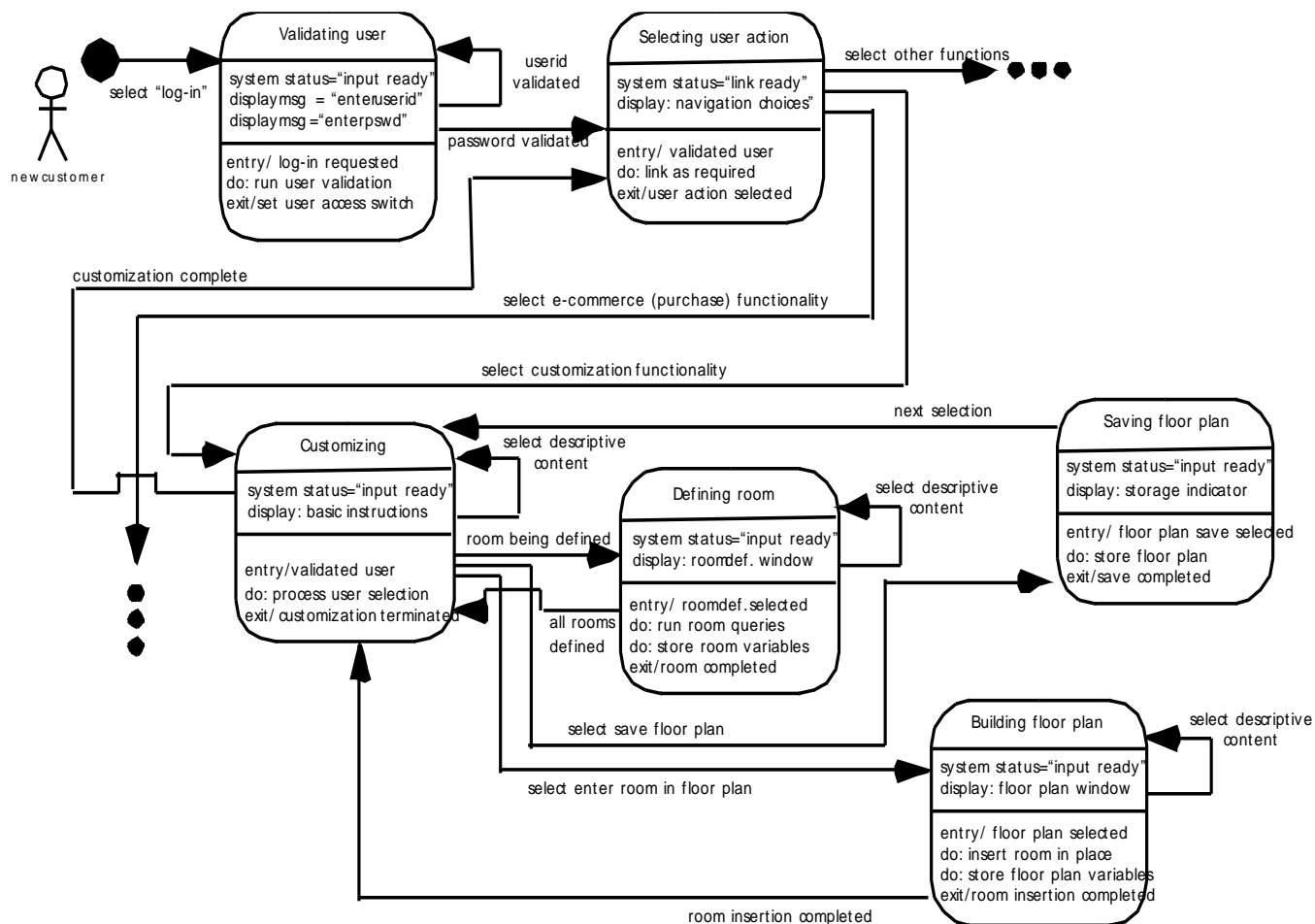


Figure 18.6 Partial state diagram for **new customer** interaction

# Mô hình chức năng

- Các mô hình chức năng giải quyết hai yếu tố xử lý của WebApp
  - Khảo sát chức năng người dùng được phân phối bởi các WebApp cho người dùng cuối
  - các hàm trong lớp thực hiện hành vi kết hợp với lớp.
- Một biểu đồ hoạt động có thể được sử dụng để biểu diễn luồng tiến trình



# Biểu đồ hoạt động

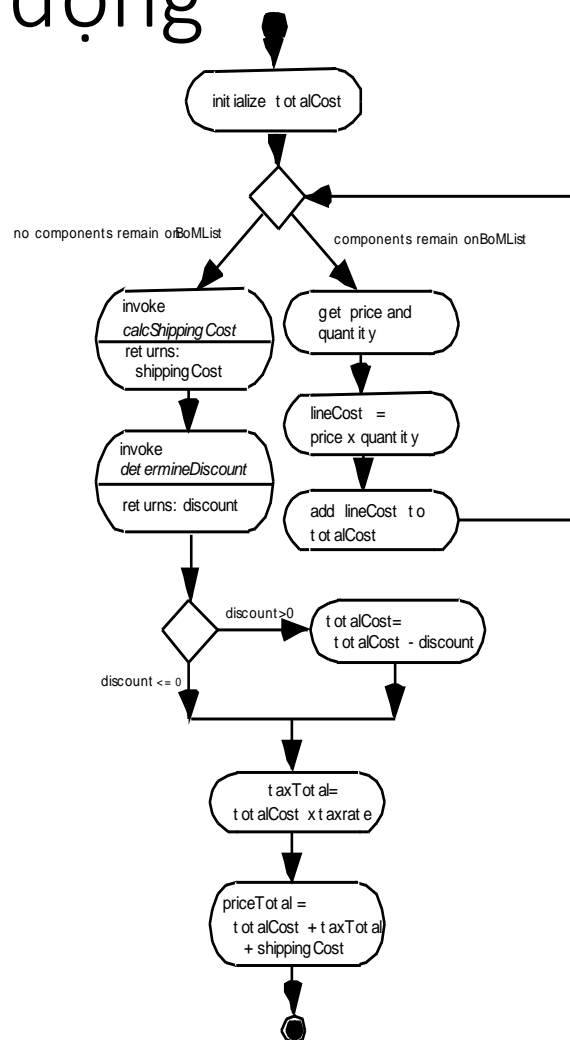


Figure 18.7 Activity diagram for **computePriceOperation**



# Mô hình cấu hình

- Máy chủ
  - Phần cứng máy chủ và môi trường hệ điều hành phải được xác định
  - Cân nhắc khả năng tương tác trên phía máy chủ phải được xem xét
  - Giao diện thích hợp, giao thức truyền thông và thông tin hợp tác liên quan phải được quy định
- Khách hàng(client)
  - Vấn đề cấu hình trình duyệt phải được xác định
  - Yêu cầu thử nghiệm cần được xác định

# Mô hình điều hướng

- Đối với các yếu tố/trạng thái dễ dàng để đạt được (yêu cầu các bước chuyển hướng ít hơn) những người khác? Ưu tiên cho các trình diễn là gì?
- Một số yếu tố cần được nhấn mạnh để buộc người dùng điều hướng theo hướng của họ?
- Làm thế nào lỗi điều hướng cần được xử lý?
- Chuyển hướng tới các nhóm có liên quan cần được ưu tiên hơn hướng đến một yếu tố cụ thể.
- Chuyển hướng nên được thực hiện thông qua các liên kết, thông qua truy cập dựa trên tìm kiếm, hoặc bằng một số phương tiện khác?
- Các yếu tố có nhất định nên giới thiệu đến người dùng dựa vào bối cảnh của hành động chuyển hướng trước?

# Mô hình điều hướng

- Nên một bản đồ chuyển hướng đầy đủ hoặc thực đơn có sẵn ở mọi điểm trong sự tương tác của người dùng?
- Thiết kế chuyển hướng nên được thúc đẩy bởi các hành vi sử dụng dự kiến phổ biến nhất hoặc bởi tầm quan trọng của các yếu tố nhận thức WebApp ?
- Có thể sử dụng một “lưu trữ” chuyển hướng trước đó thông qua các WebApp để tiến hành sử dụng trong tương lai?
- Nhóm người dùng điều hướng tối ưu nên được thiết kế?
- Làm thế nào kết nối các đường link bên ngoài để các WebApp được xử lý? Chồng lên các cửa sổ trình duyệt hiện hành? Mở một cửa sổ trình duyệt mới? Mở một khung riêng biệt?

# Tài liệu tham khảo

- Slide Set to accompany Software Engineering: A Practitioner's Approach, 7/e by Roger S. Pressman
- <http://bit.ly/2ihOLB4>

# Tổng kết

Sau buổi học, sinh viên đã hiểu được:

- Các mô hình đặc tả yêu cầu phần mềm.
- Biết cách sử dụng các mô hình trong quá trình đặc tả yêu cầu
- Hiểu được mô hình thông qua một ví dụ về xác định yêu cầu phần mềm của hệ thống WebApps



25 YEARS ANNIVERSARY  
**SOICT**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

