



# CÔNG NGHỆ PHẦN MỀM

---

## Chương 5 – Mô hình phần mềm

TUẦN 5, 6



# Các nội dung trình bày

- Mô hình ngữ cảnh (Context models)
- Mô hình tương tác (Interaction models)
- Mô hình cấu trúc (Structural models)
- Mô hình hành vi (Behavioral models)
- Kỹ thuật dẫn xuất từ mô hình (Model-driven engineering)



# Các mô hình phần mềm

- Để hiểu rõ ràng và chi tiết phần mềm lớn và phức tạp, ta phải nhìn nó từ nhiều góc nhìn khác nhau.
- Ta dùng khái niệm mô hình để miêu tả các thông tin về phần mềm theo 1 góc nhìn đặc biệt nào đó.
- Tập các mô hình khác nhau tạo thành tài liệu đặc tả chi tiết và đầy đủ về phần mềm mà ta phải xây dựng trong quá trình phát triển phần mềm.



# Các mô hình phần mềm

- Trước hết nên nhìn phần mềm theo 2 góc nhìn đối lập : góc nhìn từ ngoài và góc nhìn nội bộ trong phần mềm.
- **Mô hình use-case** cho ta đặc tả phần mềm theo góc nhìn từ ngoài vào.



# Các mô hình phần mềm

- Riêng góc nhìn nội bộ, vì phần mềm thường rất phức tạp, ta nên nhìn nó theo nhiều cấp độ trừu tượng khác nhau như :
  - Trừu tượng : ta dùng **mô hình phân tích**.
  - Cụ thể : ta dùng **mô hình thiết kế** và **mô hình triển khai**
  - Cụ thể đã hiện thực bằng các công nghệ hiện có : ta dùng **mô hình hiện thực**.
  - Để phục vụ kiểm thử, ta dùng **mô hình kiểm thử**.
- Tóm lại, mọi người đều đồng ý là nên đặc tả phần mềm theo ít nhất 6 mô hình liệt kê ở trên.



# Các mô hình phần mềm

- Mỗi mô hình cho phần mềm lớn thường chứa rất nhiều thông tin, nên nó không phải là đơn thể.
- Hiện nay, mô hình là phần tử phức hợp chứa nhiều bộ phận nhỏ, mỗi bộ phận nhỏ được gọi là lược đồ.
- UML cho ta xây dựng tối thiểu 9 loại lược đồ, chúng được chia làm 2 góc nhìn : cấu trúc tĩnh và hành vi động.



# Các mô hình phần mềm

- Để thể hiện góc nhìn về cấu trúc tĩnh của phần mềm, ta dùng 1 trong 5 lược đồ :
  - lược đồ Use-case
  - lược đồ Class
  - lược đồ Đối tượng
  - lược đồ Thành phần
  - lược đồ Triển khai.



# Các mô hình phần mềm

- Để thể hiện góc nhìn về hành vi động của phần mềm, ta dùng 1 trong 4 lược đồ :
  - lược đồ trình tự
  - lược đồ cộng tác
  - lược đồ trạng thái
  - lược đồ hoạt động



# Các ký hiệu UML

- Ngôn ngữ UML được dùng để xây dựng các lược đồ.
- Ngôn ngữ UML là ngôn ngữ đồ họa, nó chỉ có 3 ký hiệu cơ bản :
  - Hình chữ nhật miêu tả 1 phần tử mô hình nào đó
  - Hình tròn miêu tả 1 phần tử mô hình nào đó
  - Đoạn thẳng để miêu tả mối quan hệ giữa 2 phần tử.

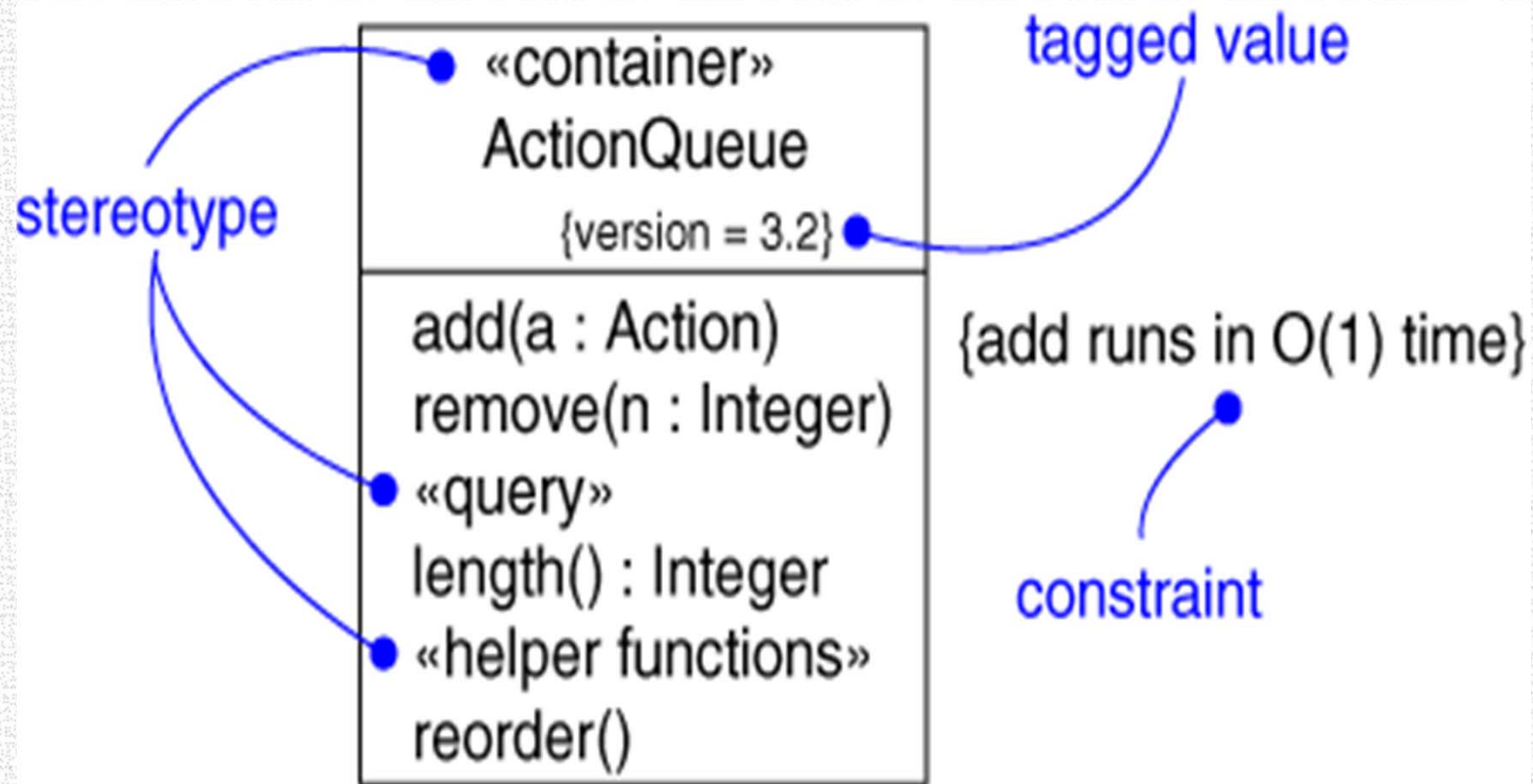


# Các ký hiệu UML

- Để miêu tả nhiều loại phần tử mô hình khác nhau, 2 ký hiệu là chưa đủ.
- Tương tự, có nhiều loại quan hệ giữa 2 phần tử như Tổng quát hóa, Phụ thuộc, Hiện thực, Kết hợp... Do đó 1 ký hiệu đoạn thẳng là chưa đủ để miêu tả các loại mối quan hệ này.
- Do đó UML giới thiệu thêm 1 số cơ chế nói rộng :
  - Tagged value : giá trị tag theo dạng {1.2}
  - Constraint : thông tin ràng buộc theo dạng {x<5}
  - Stereotype : các mẫu rập khuôn theo 1 trong 2 dạng :
    - Theo dạng đồ họa
    - Theo dạng thêm văn bản



# Các ký hiệu UML





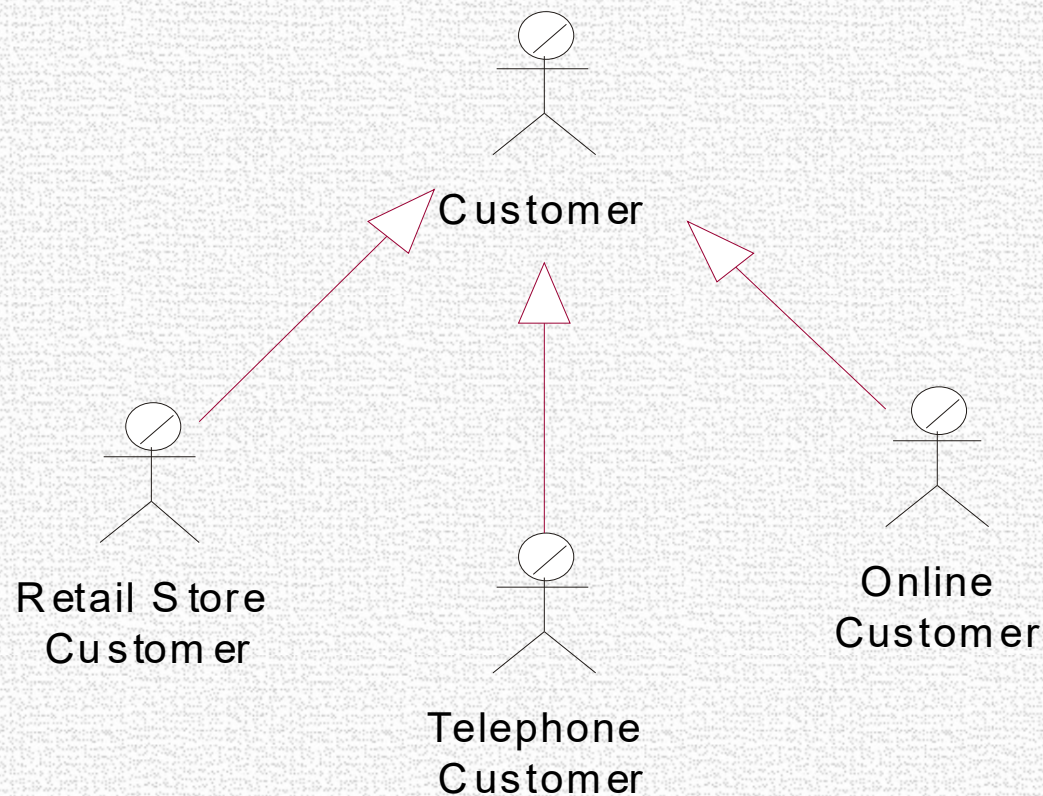
# Lược đồ use-case

- Miêu tả 1 bộ phận chức năng của phần mềm mà 1 số actor bên ngoài tương tác trực tiếp.
- Gồm 2 loại phần tử mô hình : actor và use-case. Có thể chứa thêm phần tử note để ghi chú thêm cho phần tử khác.
- Có 1 số mối quan hệ giữa actor-actor : generalization
- Có 1 số mối quan hệ giữa actor-use-case : association
- Có 1 số quan hệ giữa use-case và use-case :
  - Bao gộp (include)
  - Mở rộng (extend)



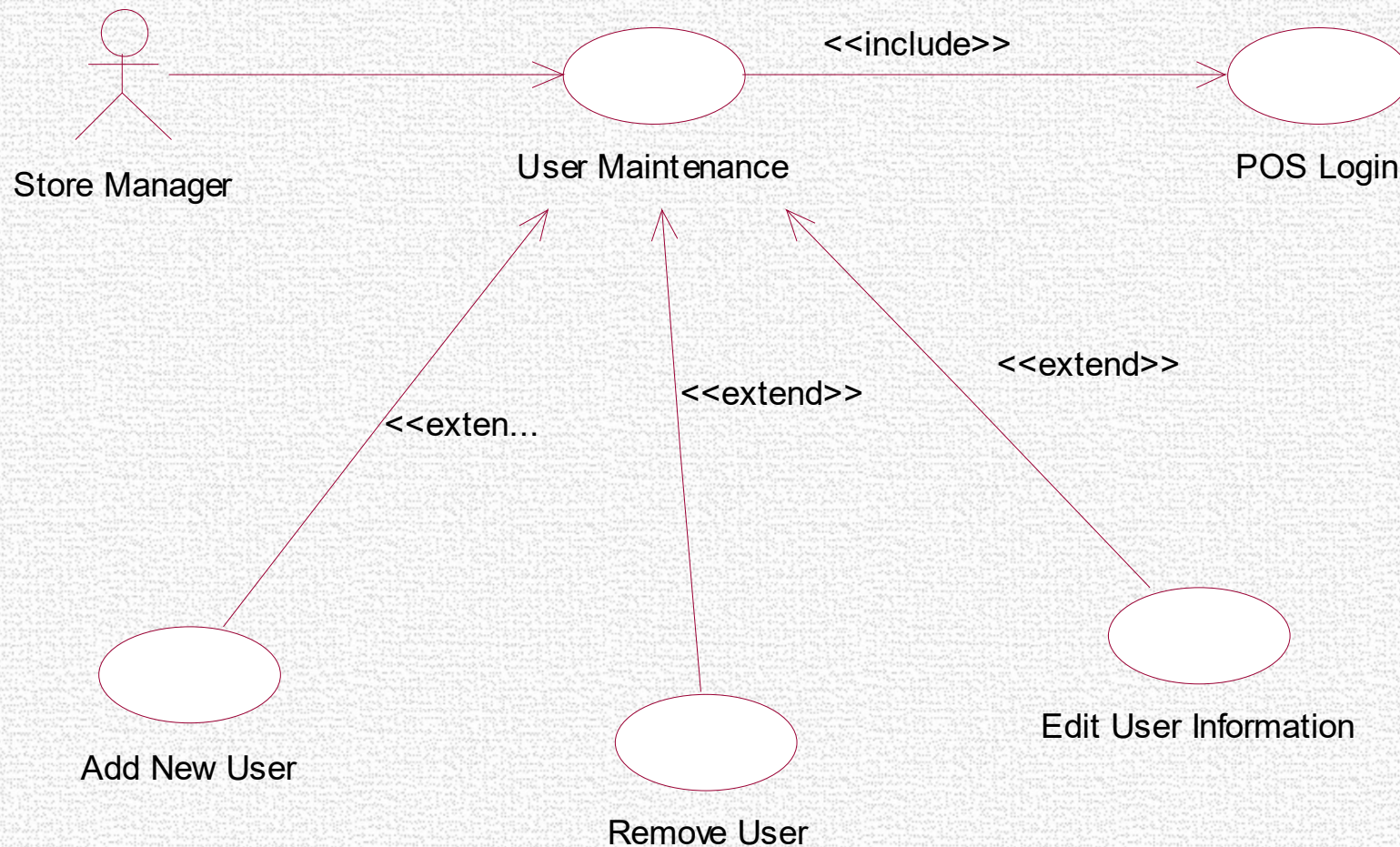
# Lược đồ use-case

## Customers





# Lược đồ use-case : Thí dụ



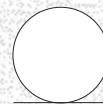


# Lược đồ class ở cấp phân tích

- Miêu tả bản phát họa việc thực hiện 1 use-case nào đó
- Gồm 2 loại phần tử mô hình : actor và class ở cấp phân tích.  
Có thể chứa thêm :
  - phần tử note để ghi chú thêm cho phần tử khác.
  - Interface của class
- Có 3 loại class ở cấp phân tích :
  - Class biên : miêu tả phần tử phục vụ tương tác giữa phần mềm với actor.
  - Class điều khiển : miêu tả hoạt động cần thực thi
  - Class thực thể : miêu tả dữ liệu được xử lý



Boundary class



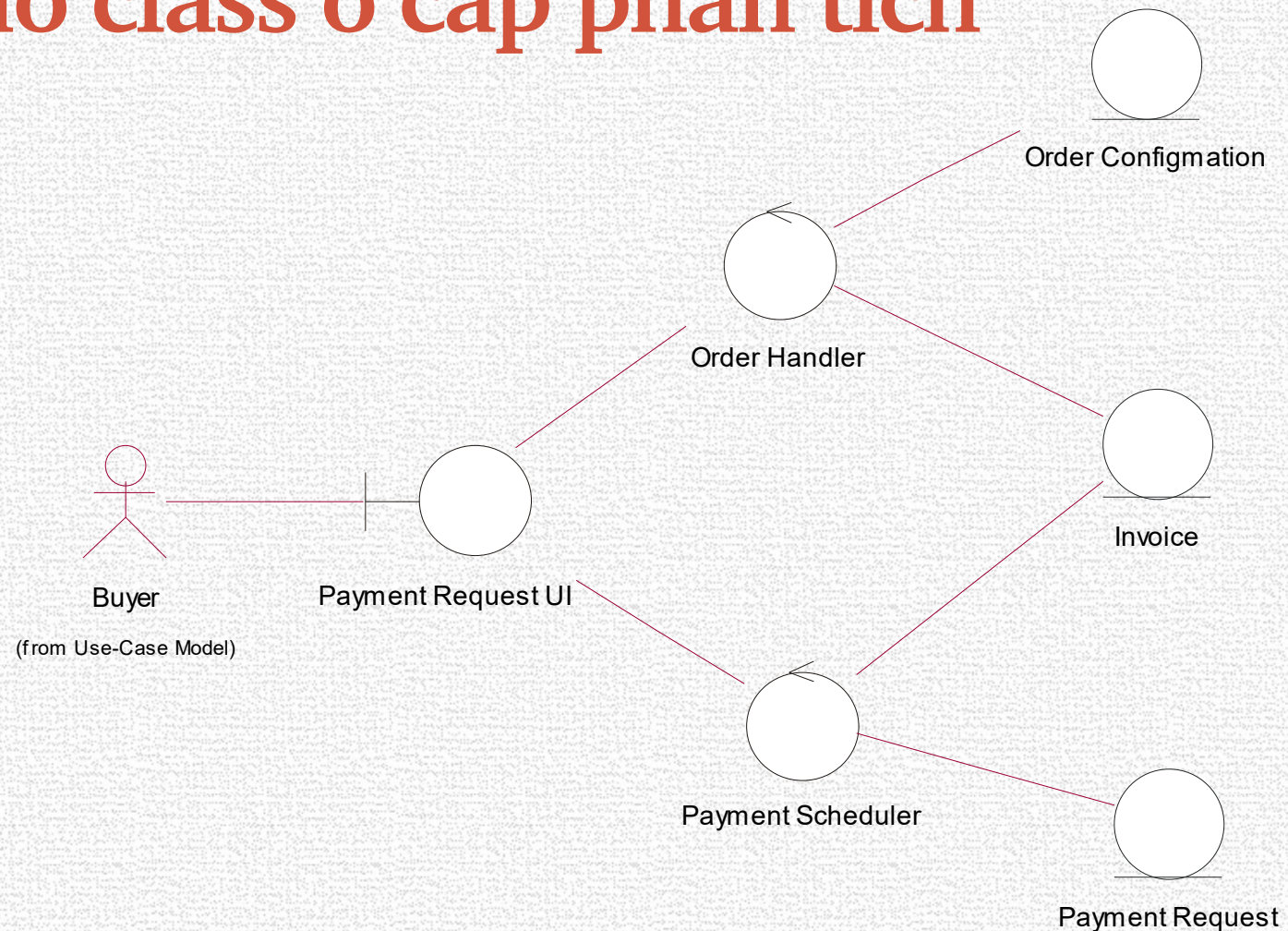
Entity class



Control class



# Lược đồ class ở cấp phân tích



Lược đồ class ở cấp phân tích phát họa cách thực hiện use-case "Pay Invoice"

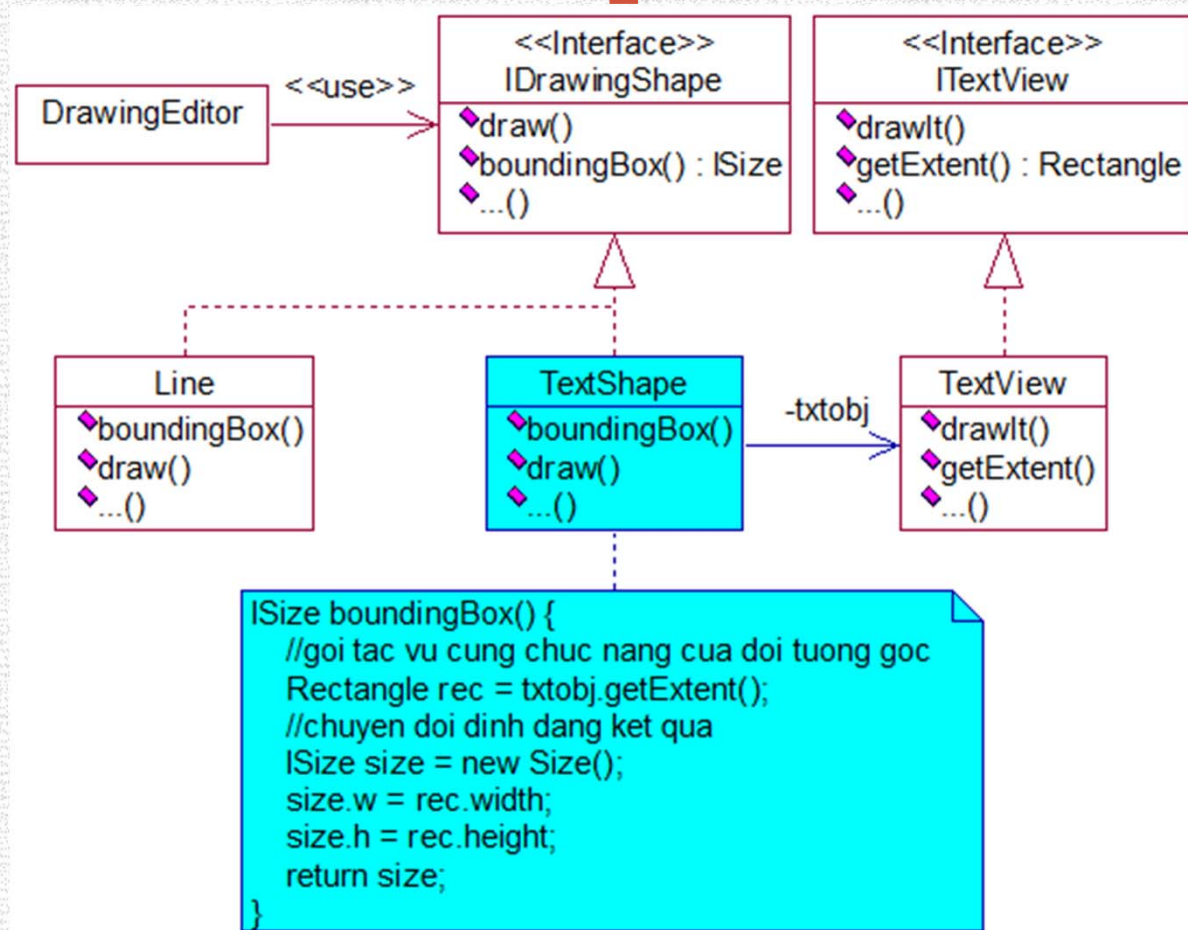


# Lược đồ class ở cấp thiết kế

- Miêu tả chi tiết việc thực hiện 1 chức năng nào đó
- Gồm các loại phần tử mô hình : actor, class ở cấp thiết kế, interface, note.
- Quan hệ giữa class và interface : hiện thực.
- Quan hệ giữa các class :
  - Thừa kế.
  - Bao gộp.
  - Kết hợp.
- Cần miêu tả đầy đủ các thông tin về từng mối quan hệ.



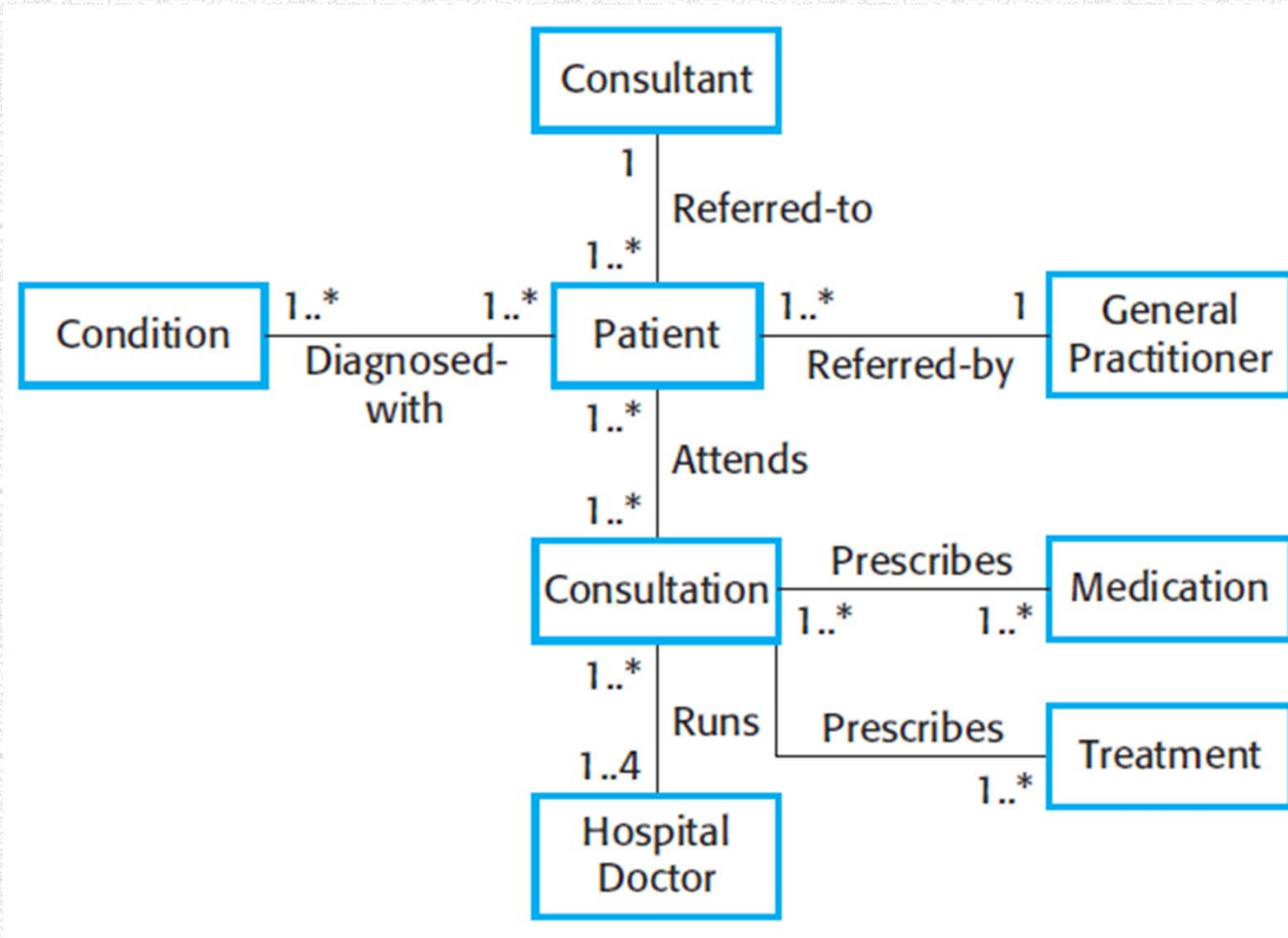
# Lược đồ class ở cấp thiết kế



Lược đồ class ở cấp thiết kế miêu tả cách dùng lại đối tượng sẵn có nhưng không cần biết interface sử dụng của nó

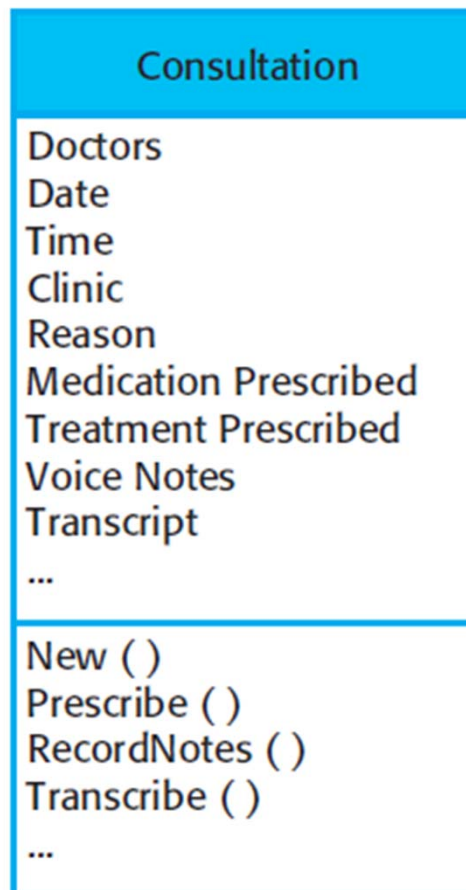


# Class và mối quan hệ kết hợp trong MHC-PMS





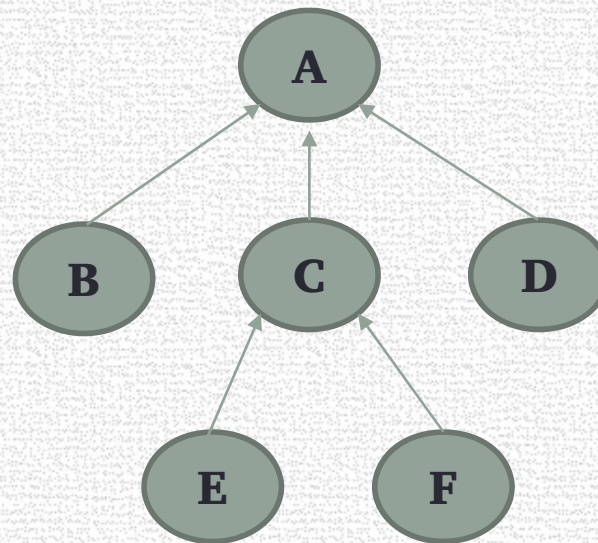
# Class Consultation trong MHC-PMS





# Mối quan hệ tổng quát hoá

- Là mối quan hệ ngược với thừa kế.
- Phần tử tổng quát hóa chứa những thông tin chung mà các phần tử khác cũng có.
- Phần tử cụ thể hóa có mọi thông tin của phần tử tổng quát và nhiều thông tin riêng của mình.



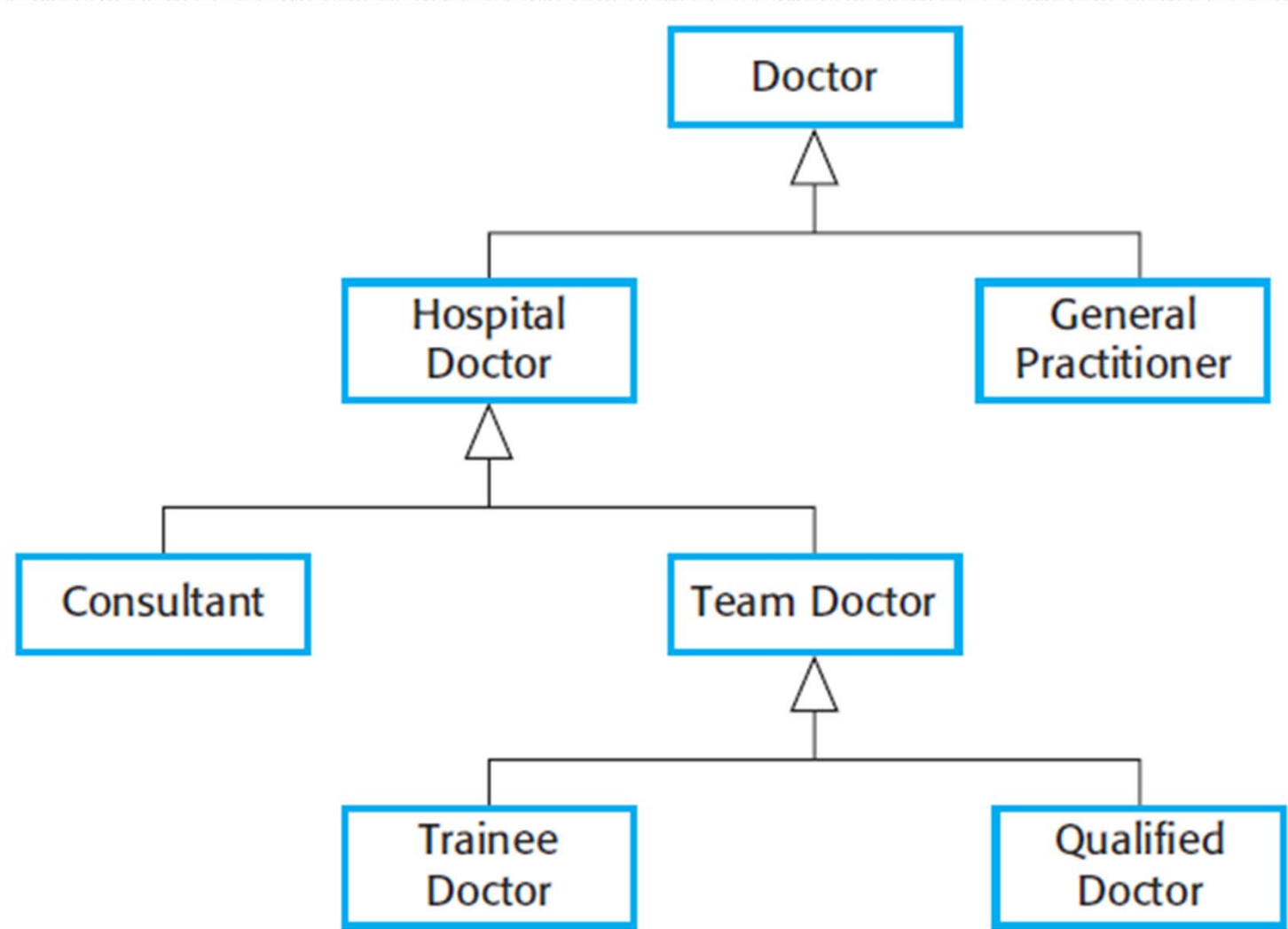


# Mối quan hệ tổng quát hoá

- Các ngôn ngữ hướng đối tượng phổ dụng như Java, C# đều hỗ trợ tính thừa kế để hiện thực mối quan hệ tổng quát hóa trong mô hình phần mềm.
- Nếu cần thay đổi thông tin chung của nhiều class, ta chỉ cần thay đổi nó 1 lần trong class tổng quát hóa.
- Các class con chỉ cần thừa kế class tổng quát hóa rồi thêm những thông tin riêng của mình.

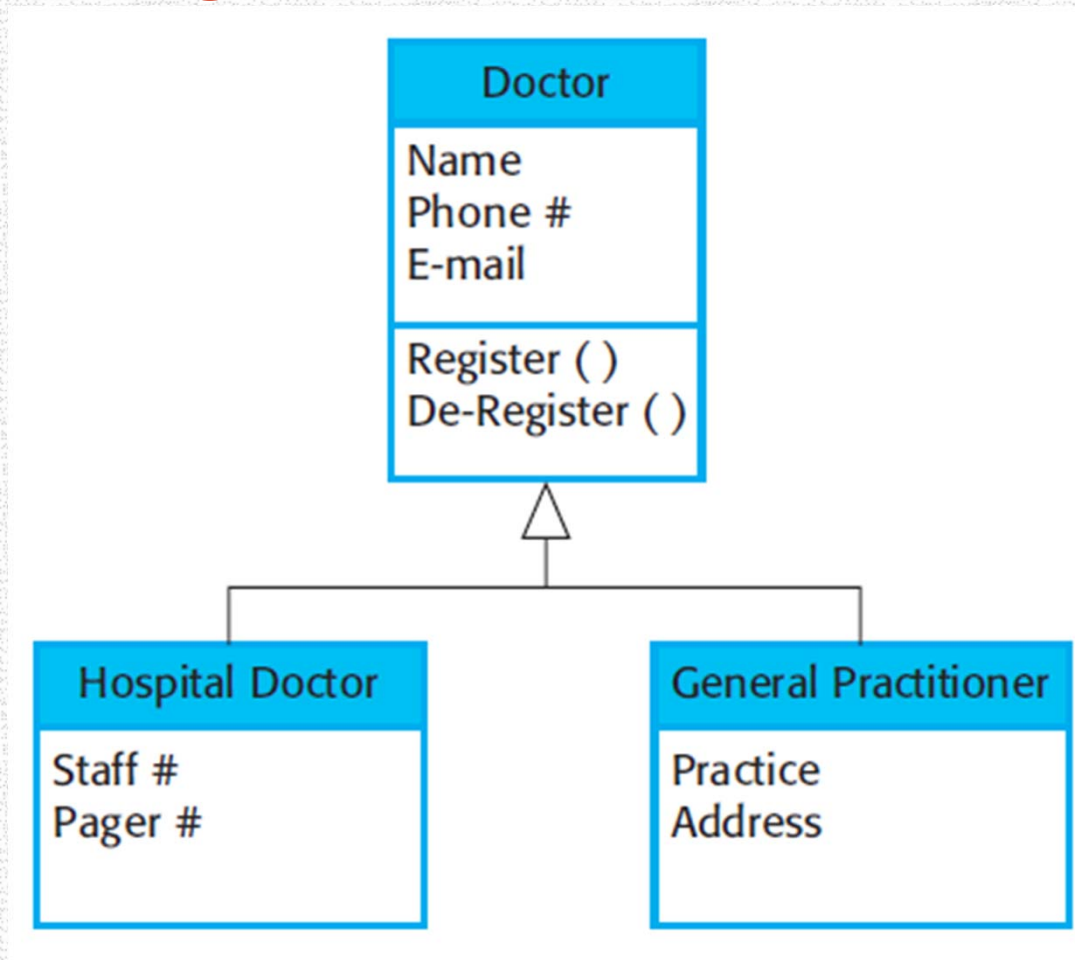


# Cây phân cấp do mối quan hệ tổng quát hóa trong MHC-PMS



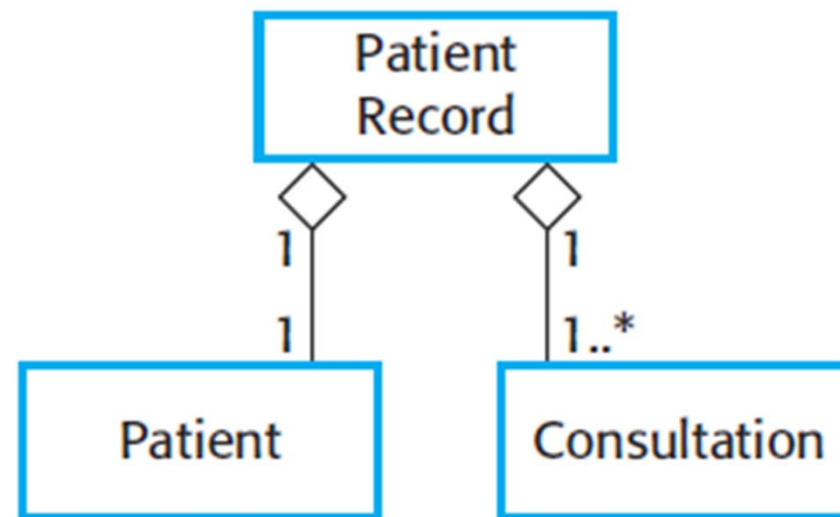


# Mối quan hệ tổng quát hóa với các thông tin tăng cường





# Mỗi quan hệ bao gồm



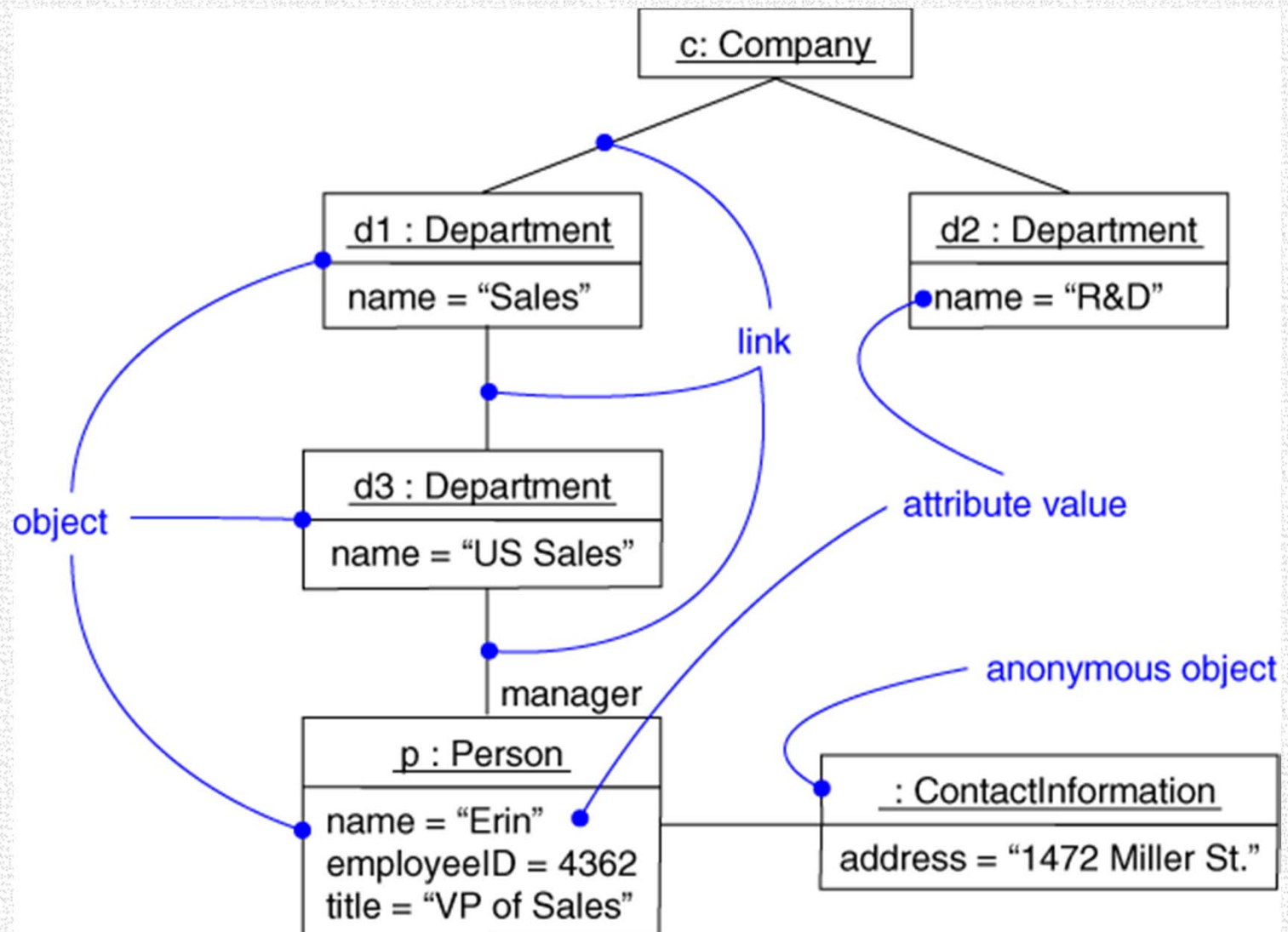


# Lược đồ đối tượng

- Cho thấy các đối tượng tham gia thực hiện 1 chức năng tương ứng, trạng thái của chúng và mối quan hệ giữa các đối tượng này.
- Được xây dựng trong hoạt động phân tích và thiết kế.
- Mục đích :
  - Cho thấy cấu trúc đối tượng và cấu trúc dữ liệu.
  - Đặc tả snapshots.
- Phân tích viên, nhà thiết kế và lập trình viên sẽ hợp tác với nhau để xây dựng và duy trì lược đồ thành phần.



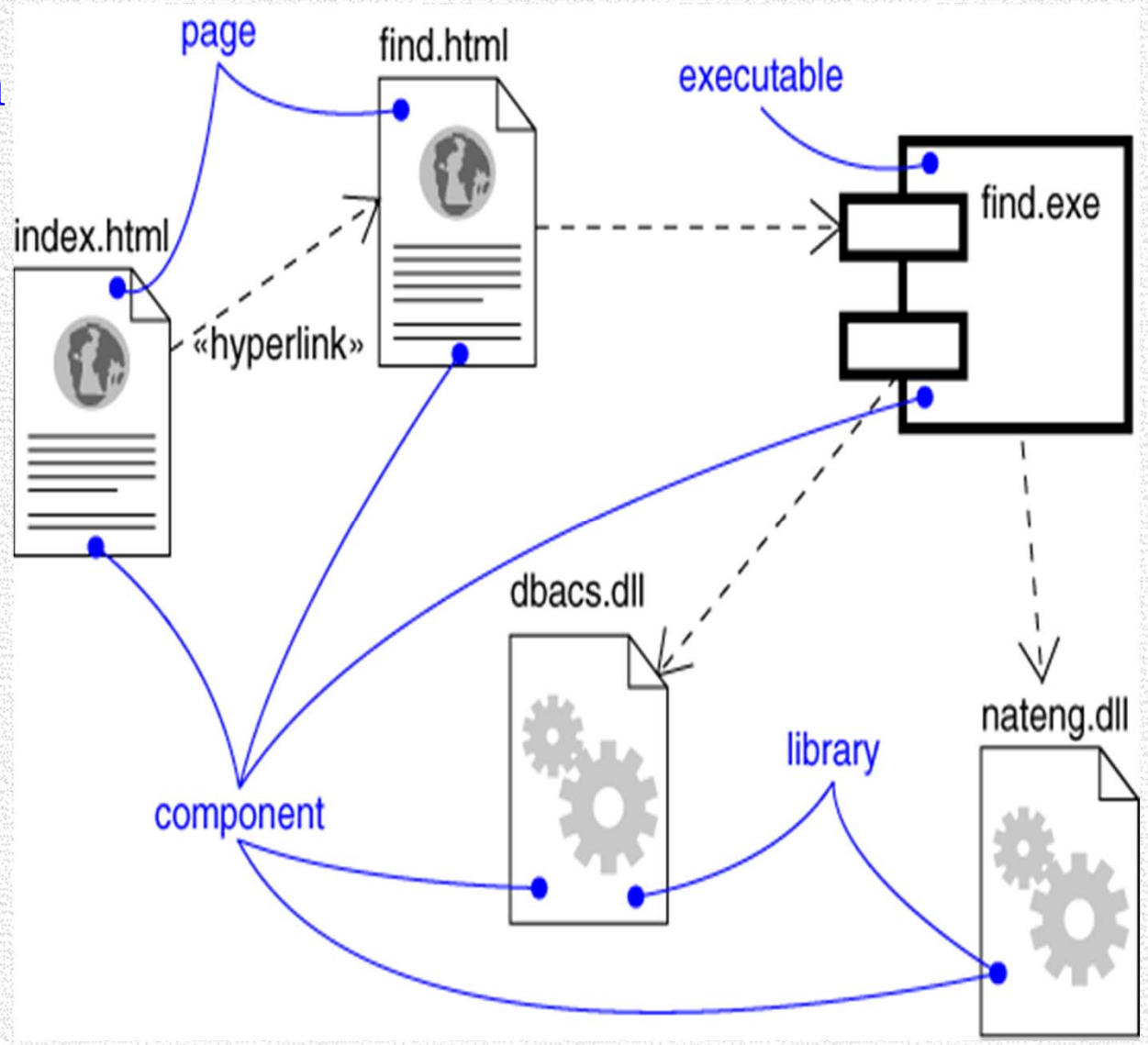
# Lược đồ đối tượng





# Lược đồ thành phần

- Miêu tả chi tiết hiện thực việc thực hiện 1 chức năng nào đó gồm nhiều thành phần thực tế và mối quan hệ giữa chúng.
- Thông tin quan trọng nhất của mỗi thành phần là công nghệ xây dựng nó : trang Web, file \*.exe, thư viện \*.dll,...





# Lược đồ thành phần

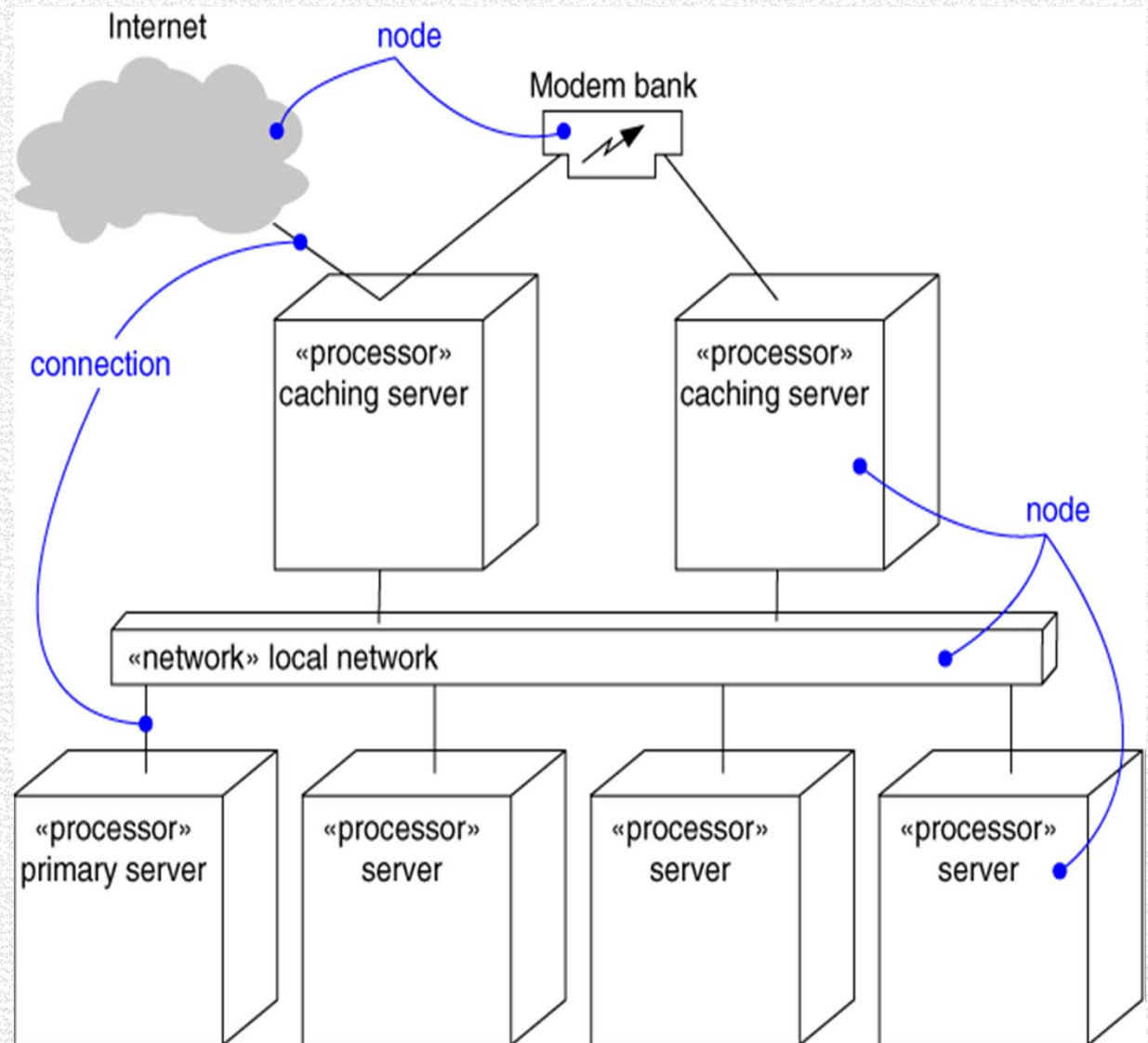
- Cho thấy cấu trúc vật lý của bộ phận phần mềm thực hiện 1 chức năng xác định.
- Được xây dựng như là 1 phần của đặc tả kiến trúc phần mềm.
- Mục đích :
  - Giúp tổ chức mã nguồn.
  - Xây dựng release chạy được.
  - Đặc tả database vật lý.
- Kiến trúc sư và lập trình viên sẽ hợp tác với nhau để xây dựng và duy trì lược đồ thành phần.





# Lược đồ triển khai

- Miêu tả cấu trúc liên kết giữa các tài nguyên vật lý hầu phục vụ việc chạy phần mềm.
- Mỗi phần tử miêu tả 1 tài nguyên và vai trò, chức năng của tài nguyên đó.





# Lược đồ triển khai

- Cho thấy cấu trúc liên kết của các tài nguyên phần cứng.
- Được xây dựng như là 1 phần của đặc tả kiến trúc phần mềm.
- Mục đích :
  - Giúp đặc tả sự phân bố các thành phần phần mềm.
  - Nhận dạng các vị trí cổ nút chai ảnh hưởng đến hiệu suất.
- Kiến trúc sư, kỹ sư mạng và kỹ sư phần mềm sẽ hợp tác với nhau để xây dựng và duy trì lược đồ triển khai.



# Lược đồ trạng thái

- Cho thấy hành vi động của một đối tượng/một thành phần phần mềm theo góc nhìn chuyển trạng thái của nó theo các sự kiện ngoài xảy ra.
- Mục đích :
  - Mô hình đời sống đối tượng.
  - Mô hình các đối tượng phản ứng theo sự kiện (phần tử giao diện người dùng, thiết bị I/O,...).
- kỹ sư phần mềm sẽ xây dựng và duy trì lược đồ trạng thái.



# Lược đồ trạng thái của lò vi sóng

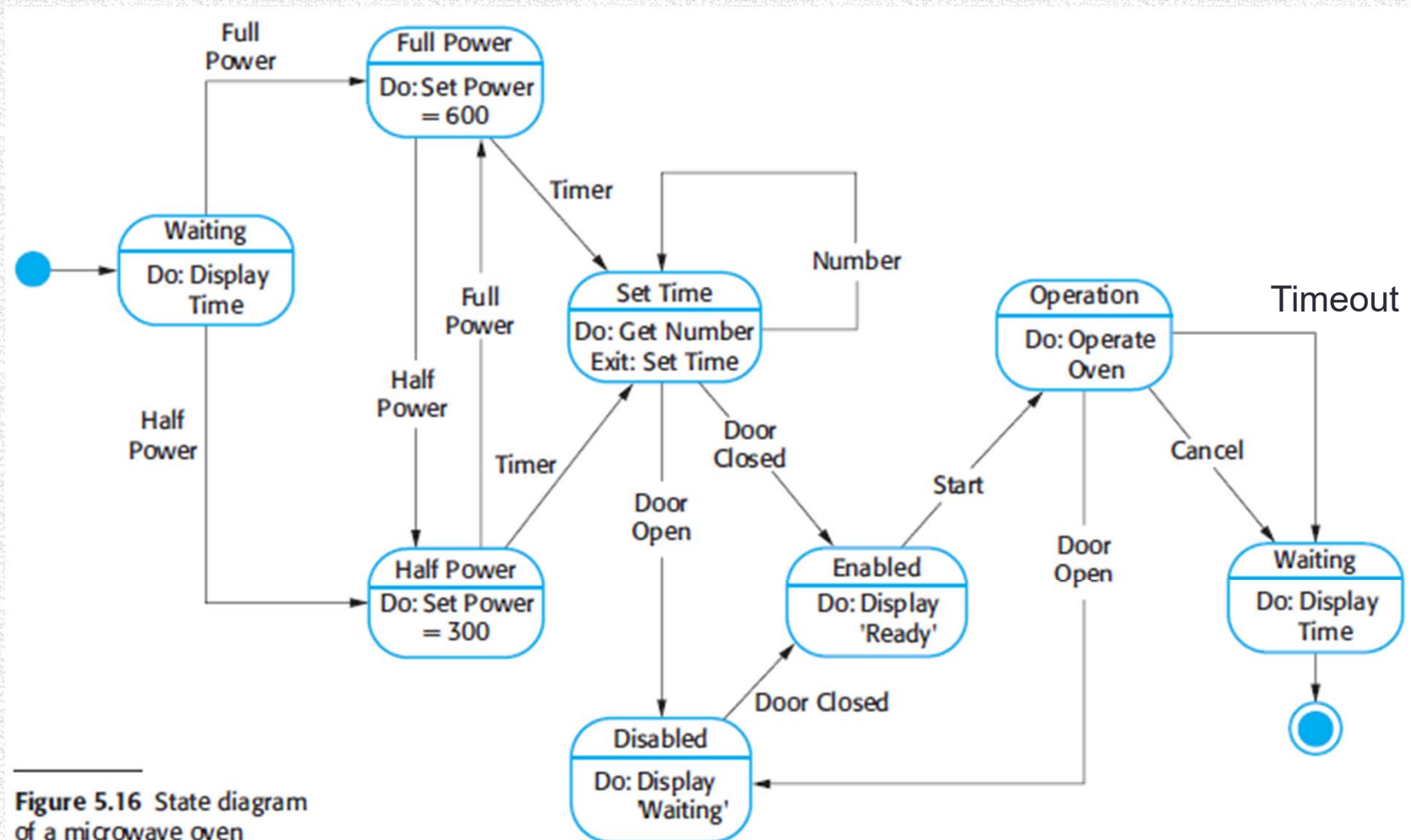


Figure 5.16 State diagram of a microwave oven



# Các trạng thái của lò vi sóng

State	Description
Waiting	Lò đợi nhập, nó hiển thị thời gian hiện hành.
Half power	Công suất nấu được thiết lập ở 300 watts. Lò hiển thị chữ 'Half power'.
Full power	Công suất nấu được thiết lập ở 600 watts. Lò hiển thị chữ 'Full power'.
Set time	Người dùng nhập từng ký số miêu tả thời gian lò sẽ nấu. Màn hình hiển thị giá trị được nhập và hiệu chỉnh theo yêu cầu người dùng.
Disabled	Lò bị cấm hoạt động, đèn bên trong sáng, lò hiển thị chữ 'Not ready'.
Enabled	Lò được phép hoạt động, đèn bên trong tắt, lò hiển thị chữ 'Ready to cook'.
Operation	Lò đang nấu, đèn bên trong sáng, lò hiển thị giá trị timer giảm dần. Khi nấu xong, đèn bên trong sáng, lò hiển thị chữ 'Cooking complete' và tạo âm báo vang lên 5 giây.

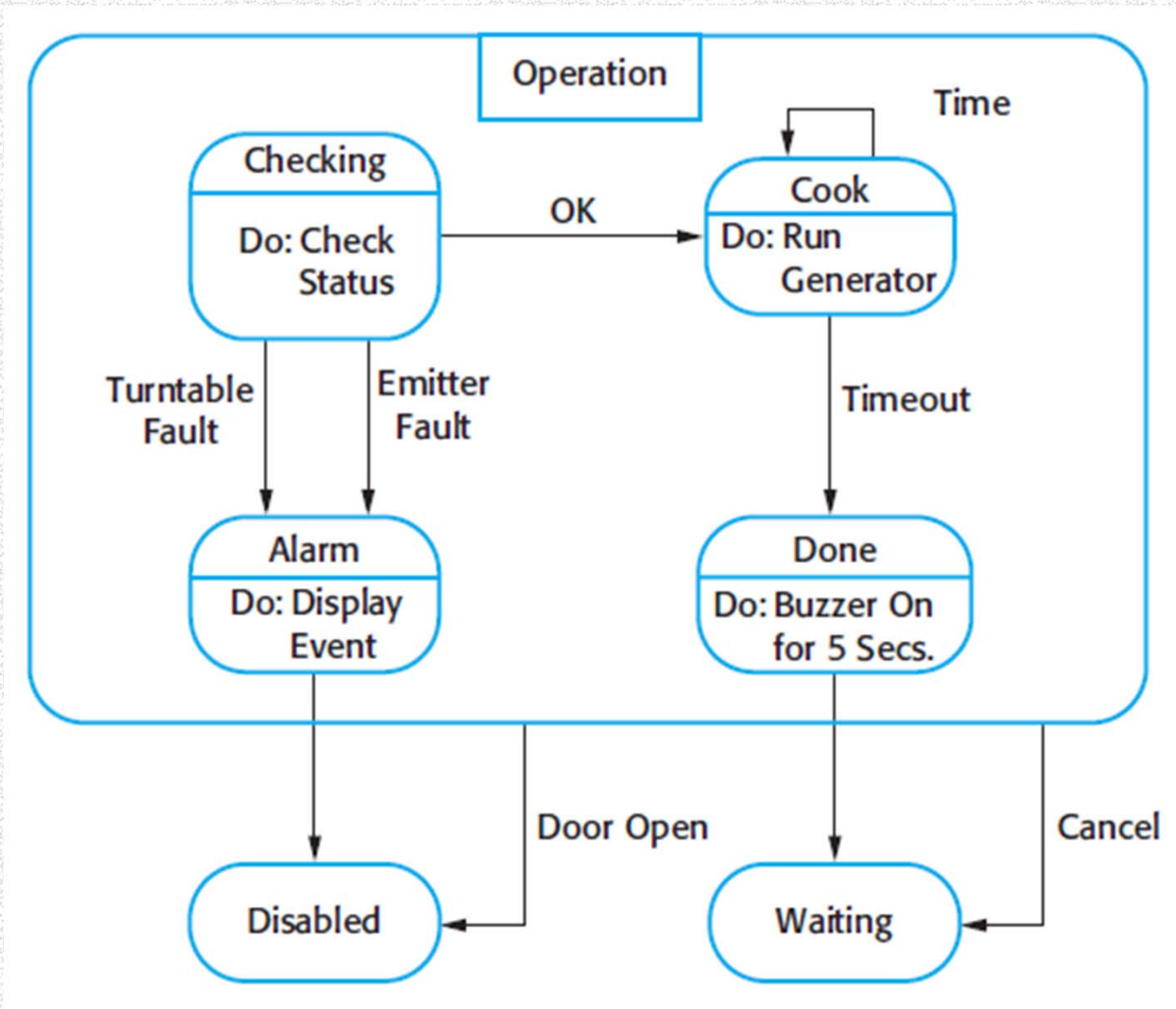


# Các kích thích trong lò vi sóng

Stimulus	Description
Half power	Người dùng vừa ấn button “half-power”.
Full power	Người dùng vừa ấn button “full-power”.
Timer	Người dùng vừa ấn 1 trong các button của timer.
Number	Người dùng vừa ấn nút số.
Door open	Cửa lò vừa bị mở.
Door closed	Cửa lò vừa bị đóng.
Start	Người dùng vừa ấn button “Start”.
Cancel	Người dùng vừa ấn button “Cancel”.



# Hoạt động của lò vi sóng



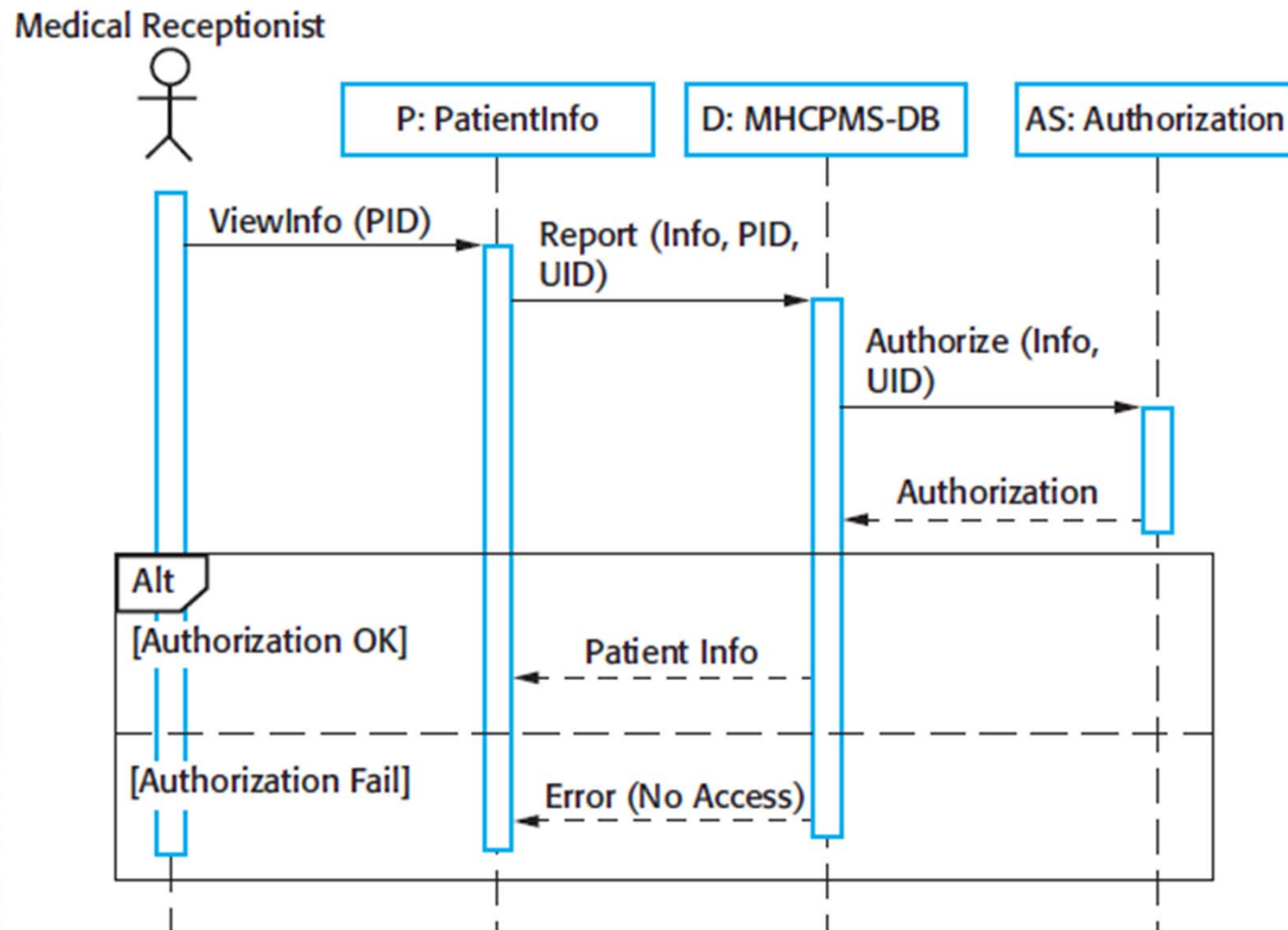


# Lược đồ trình tự (Sequence diagrams)

- Lược đồ trình tự miêu tả trình tự tương tác cụ thể theo thời gian giữa actor với các đối tượng phần mềm và giữa các đối tượng phần mềm để thực hiện 1 use-case tương ứng.
- Các actor và đối tượng có liên quan thường được vẽ theo chiều ngang ở hàng trên cùng, mỗi phần tử có 1 đường miêu tả đời sống từ giữa nó thẳng dọc xuống dưới.
- Mỗi tương tác giữa 2 thành phần được vẽ bằng đoạn thẳng có mũi tên giữa 2 thành phần đó.



# Lược đồ trình tự của use-case “View patient information”





# Xây dựng lược đồ trình tự

1. Xác định use-case cần vẽ lược đồ
2. nhận dạng phần tử khởi động use-case
  - Có thể là user, hay
  - Đối tượng của phần mềm
    - Xác định tên class
    - Xác định tên đối tượng
3. Vẽ hình chữ nhật miêu tả phần tử này ở vị trí trên trình tự lược đồ
  - Dùng ký hiệu UML là object
4. Vẽ đường thẳng đứng ở giữa và dưới hình chữ nhật để miêu tả đời sống của phần tử tương ứng.
5. Vẽ mũi tên hướng về bên phải từ nó để miêu tả việc gửi thông báo nhờ vả đến thành phần khác.





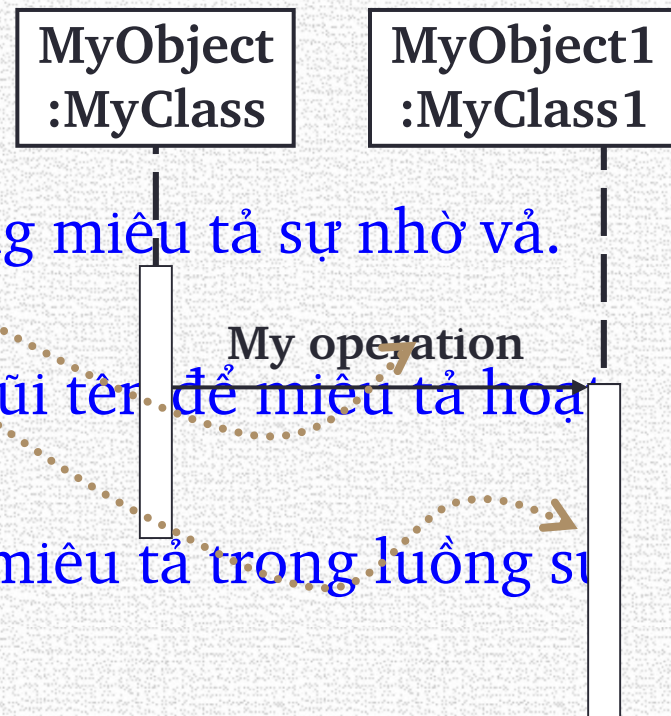
# Xây dựng lược đồ trình tự

## 6. Xác định phần tử xử lý hoạt động được nhờ và

- Đối tượng của phần mềm

Xác định tên class

Xác định tên đối tượng



## 7. Nhập tên hoạt động cho đoạn thẳng miêu tả sự nhờ vả.

- Không miêu tả thông tin trả về?

## 8. Vẽ hình chữ nhật kéo dài từ đầu mũi tên để miêu tả hoạt động bắt đầu.

## 9. ... tiếp tục với mỗi nội dung được miêu tả trong luồng sự kiện của use-case.



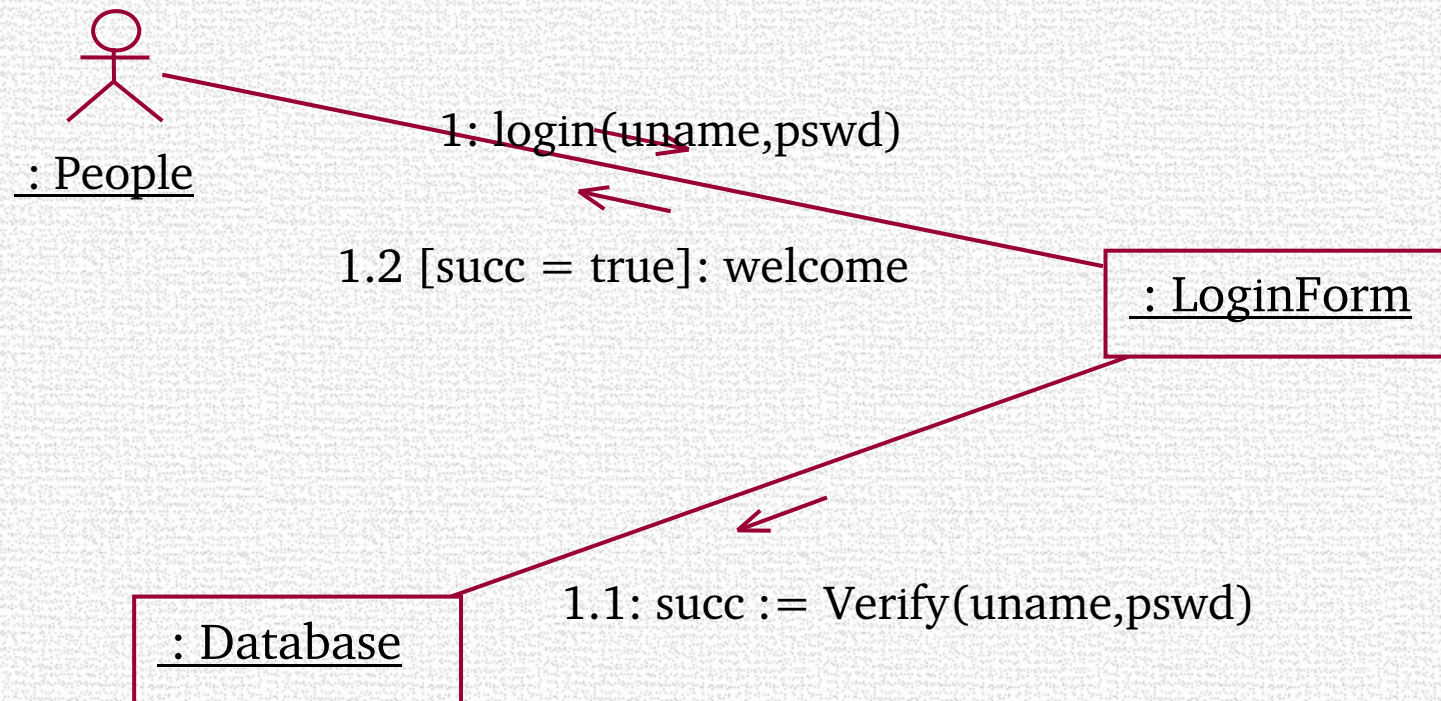
# Lược đồ cộng tác (Collaboration diagrams)

- Lược đề cộng tác miêu tả các tương tác giữa actor với các đối tượng phần mềm và giữa các đối tượng phần mềm để thực hiện 1 use-case tương ứng.
- Lược đồ cộng tác có thể chứa các phần tử :
  - Actor, đối tượng
  - Đoạn thẳng nối giữa 2 phần tử miêu tả từ 1 đến n hoạt động tương tác giữa 2 phần tử đó.



# Thí dụ lược đồ cộng tác

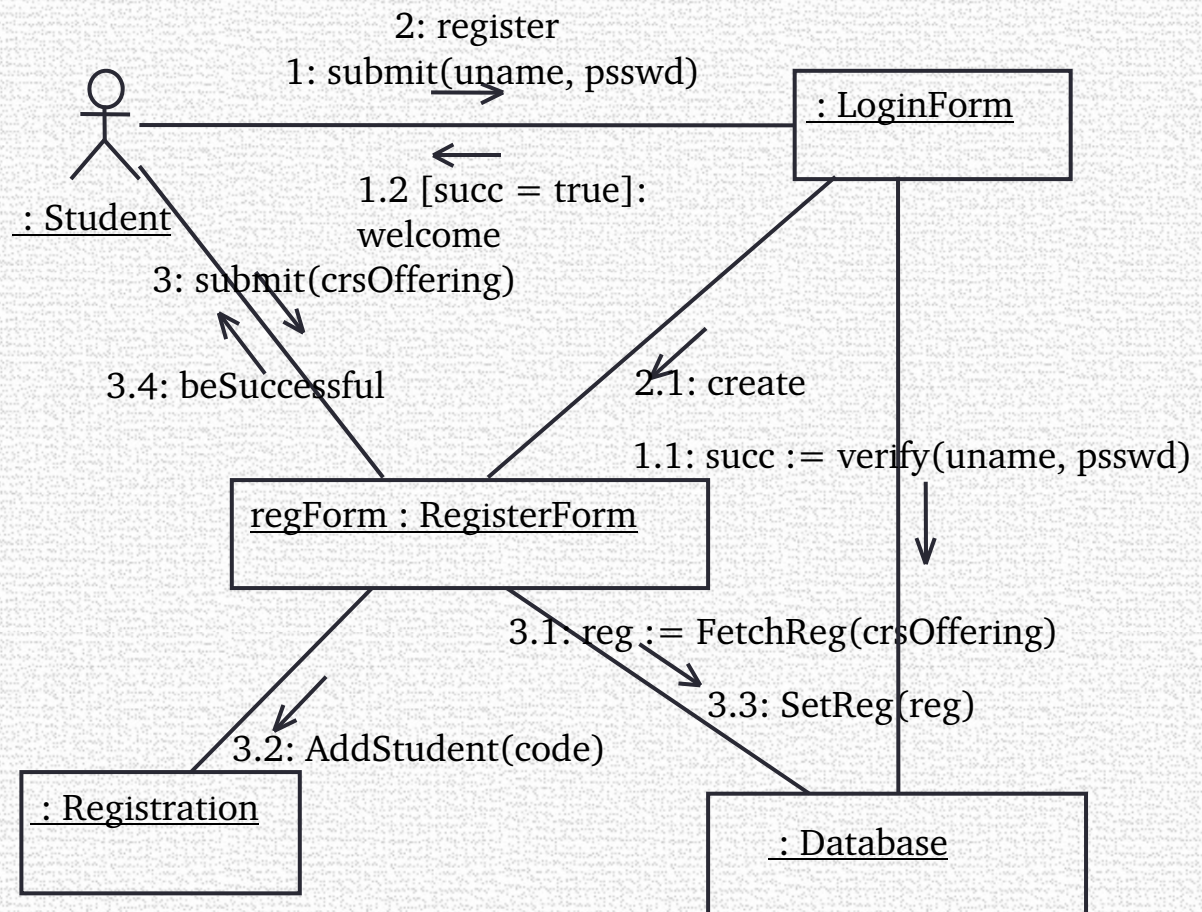
- Lược đồ cộng tác cho use-case “Login”





# Thí dụ lược đồ cộng tác

- Thí dụ lược đồ cộng tác cho use-case “Registers course”





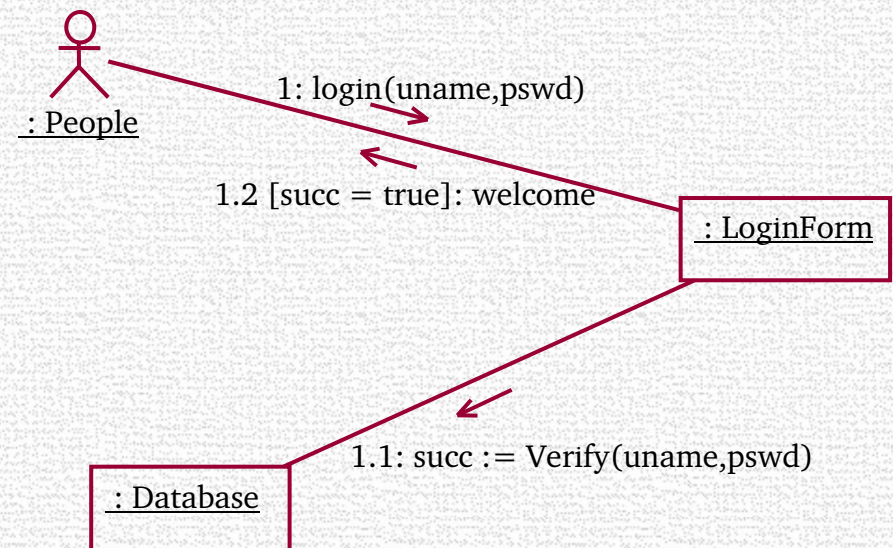
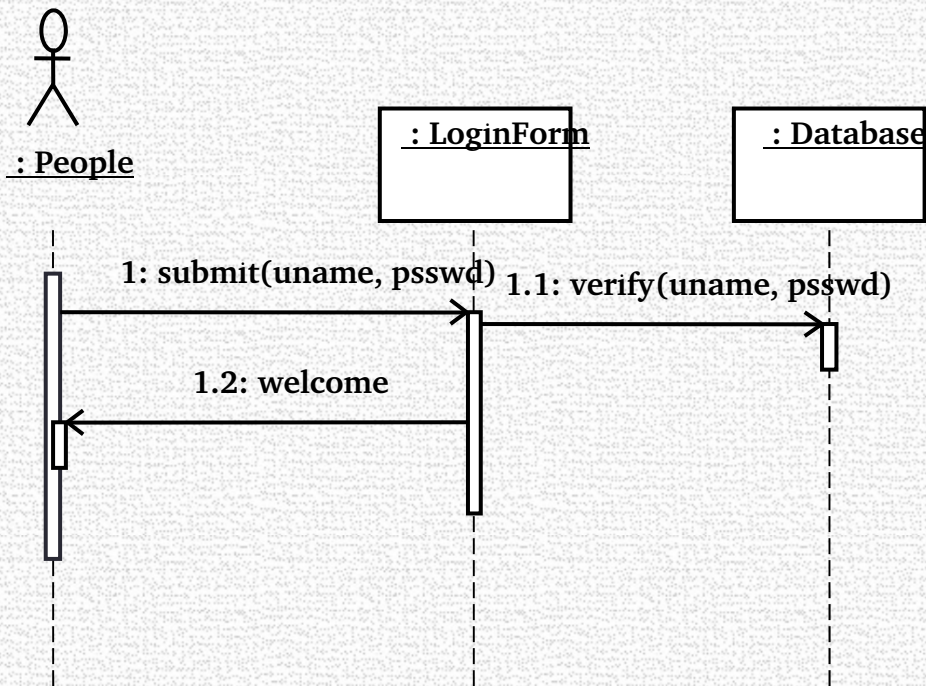
# Lược đồ cộng tác & trình tự

- Lược đồ cộng tác và lược đồ trình tự cùng miêu tả các thông tin giống nhau nhưng thể hiện theo 2 cách khác nhau : hoặc chi tiết theo thời gian các tương tác hoặc tổng quát các tương tác giữa các đối tượng khác nhau.
- Từ lược đồ trình tự, ta luôn tạo được lược đồ cộng tác.
- Từ lược đồ cộng tác, nếu có đầy đủ các tương tác theo thời gian, ta có thể xây dựng lược đồ trình tự.



# Lược đồ cộng tác & trình tự

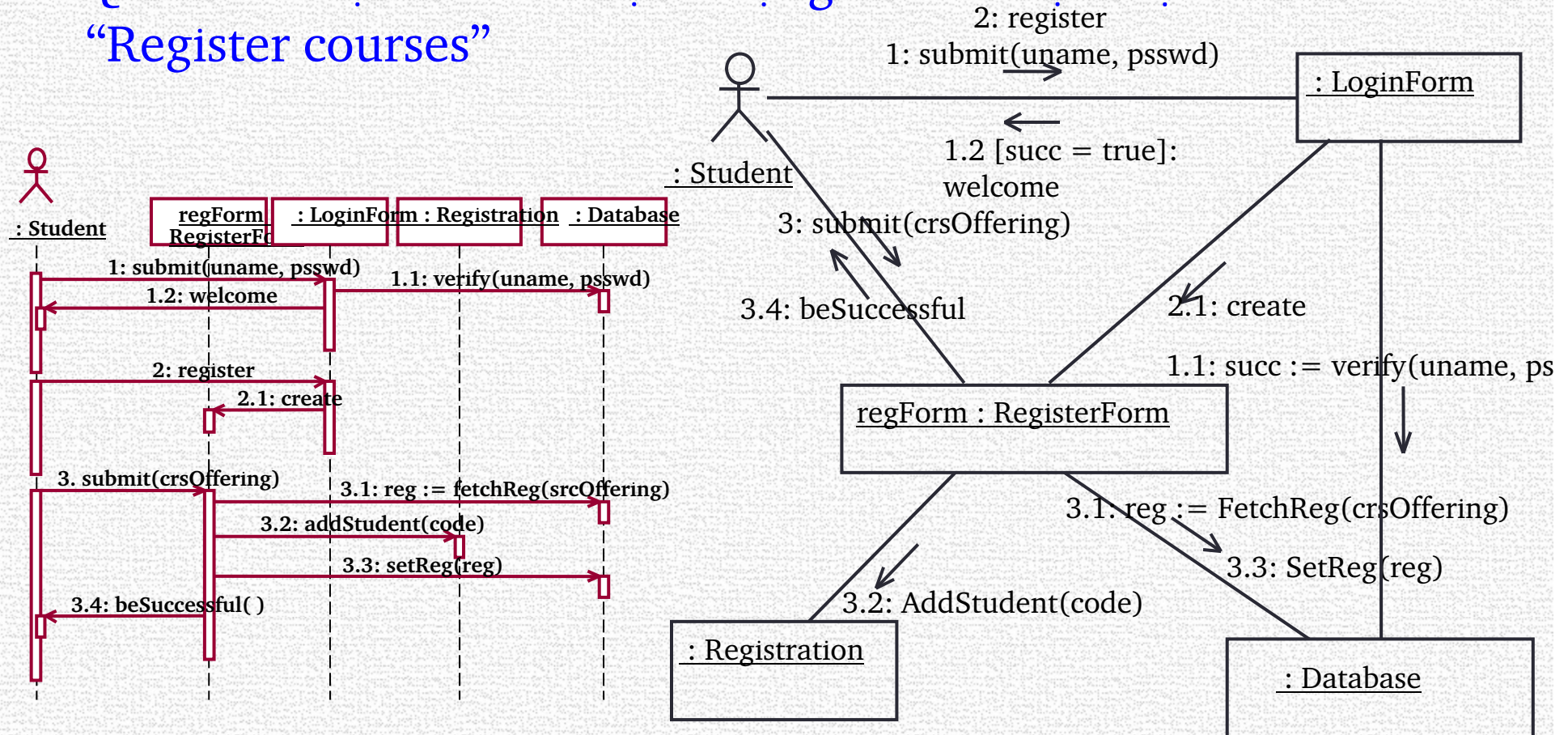
- Quan sát lược đồ trình tự và cộng tác để thực hiện use-case “Login”





# Thí dụ lược đồ cộng tác & trình tự

- Quan sát lược đồ trình tự và cộng tác để thực hiện use-case “Register courses”





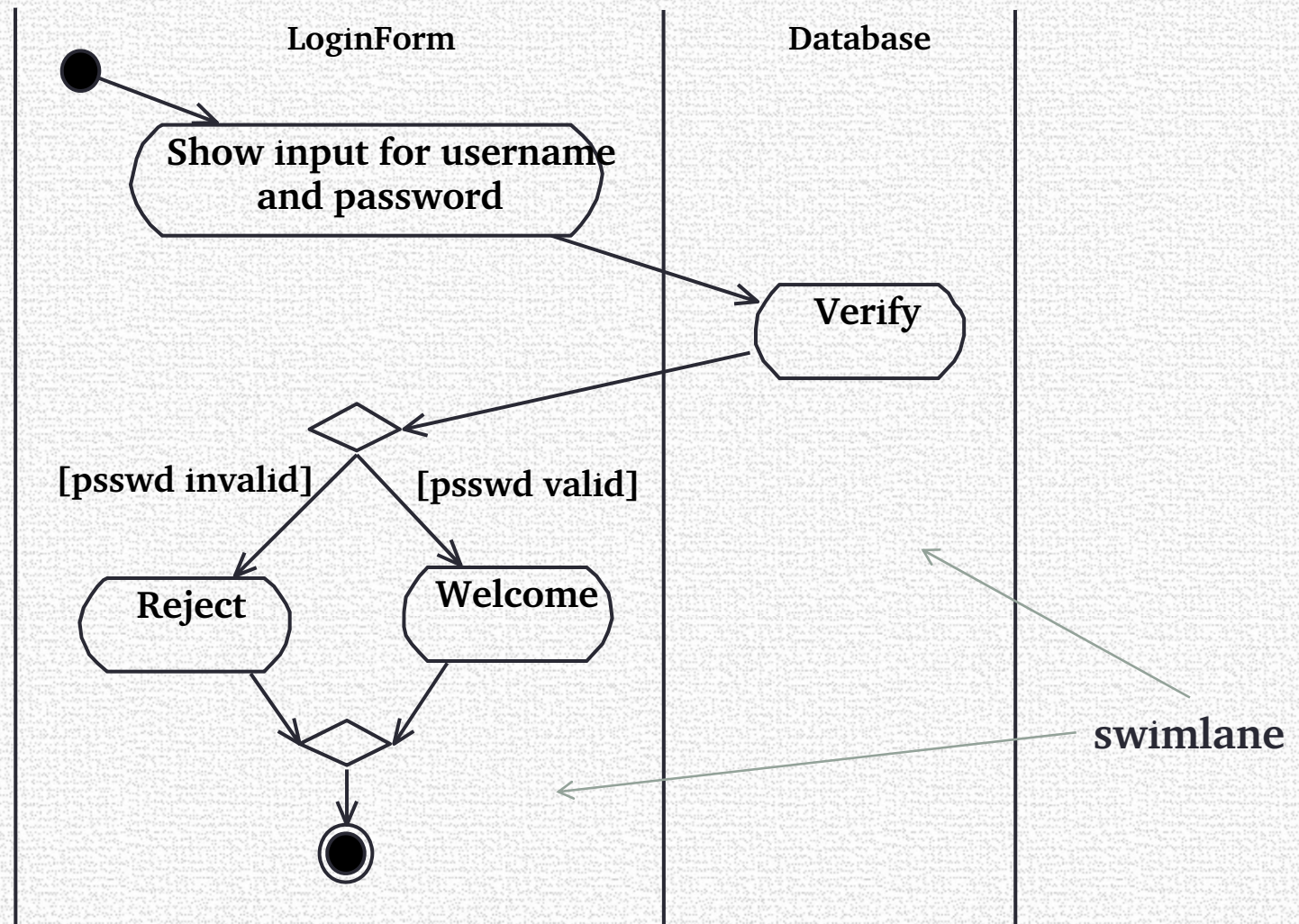
# Lược đồ hoạt động

- Lược đồ hoạt động miêu tả thuật giải thực hiện chức năng tương ứng. Nó là sự mở rộng của flowchart trước đây.
- Lược đồ hoạt động miêu tả góc nhìn động của hệ thống, đặc biệt quan trọng trong việc mô hình chức năng của phần mềm, nó nhấn mạnh luồng điều khiển giữa các đối tượng.
- Lược đồ hoạt động miêu tả :
  - Thuật giải : có thể ở dạng tuần tự hay song song.
  - Các đối tượng tham gia : mỗi đối tượng thực hiện các công việc nào.



# Thí dụ về lược đồ hoạt động

- Lược đồ hoạt động thực hiện chức năng đăng nhập bằng form Login

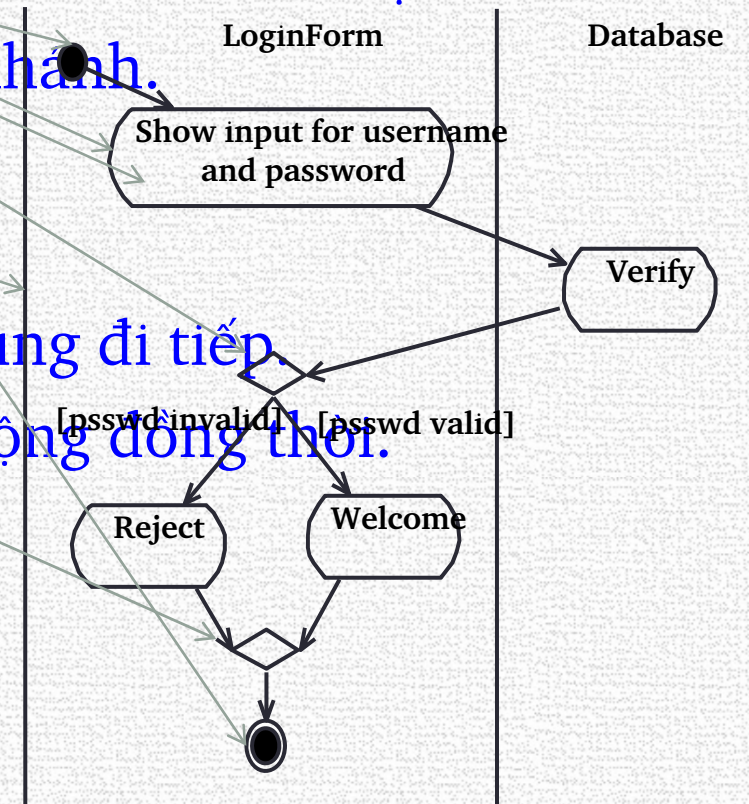




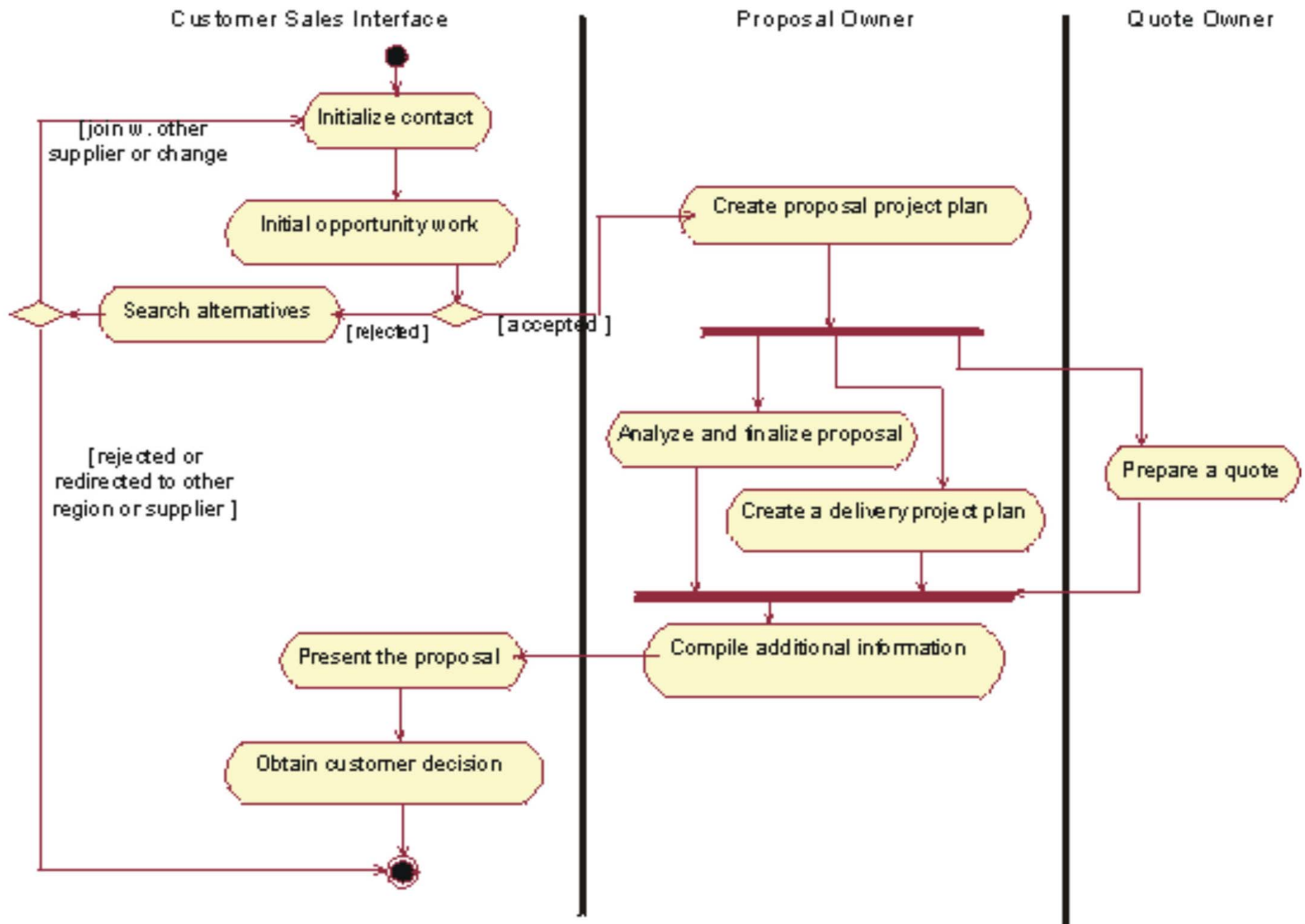
# Thí dụ về lược đồ hoạt động

Lược đồ hoạt động có thể chứa các loại phần tử :

- Nút bắt đầu và nút kết thúc : tròn nhỏ đen, tròn chứa tròn nhỏ đen
- Nút miêu tả hành động cần thực hiện : hình chữ nhật bo tròn.
- Nút quyết định theo điều kiện để rẽ nhánh.
- Cung chuyển hoạt động
- Làn bơi cho từng đối tượng tham gia
- Nút hợp nhiều cung chuyển vào để cùng đi tiếp.
- Thanh bắt đầu và kết thúc các hoạt động đồng thời.







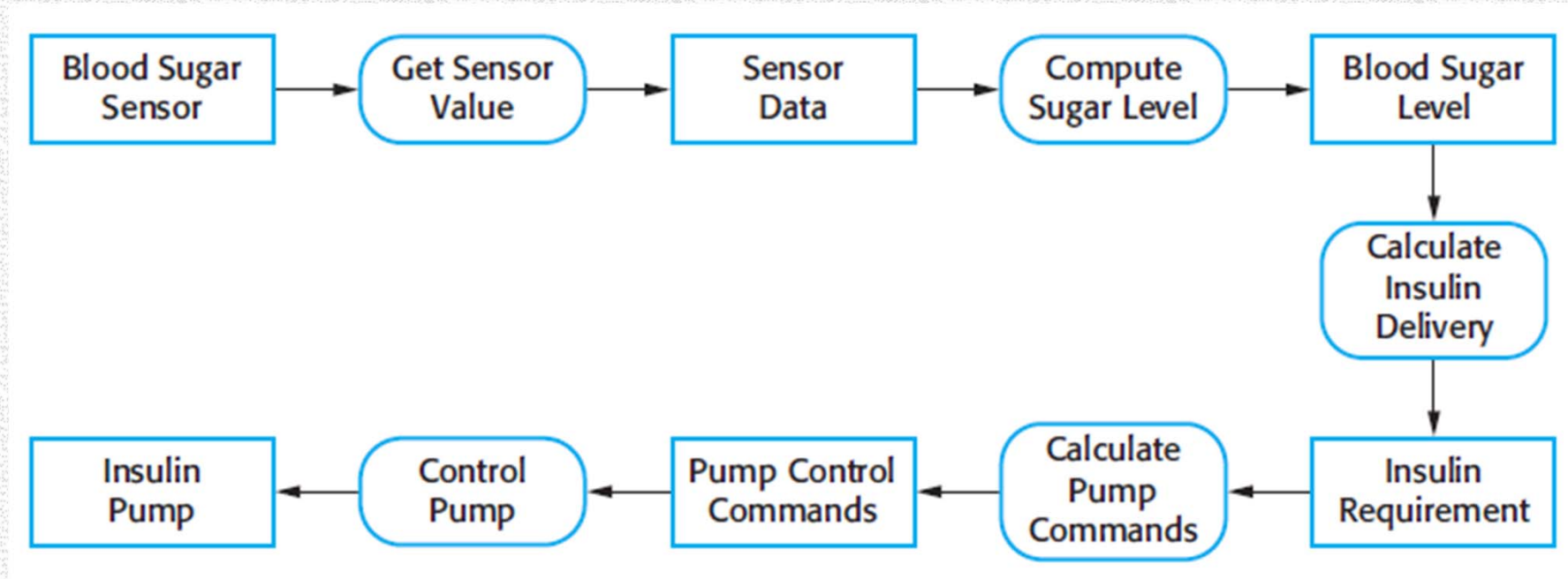


# Mô hình dẫn xuất dữ liệu (Data-driven)

- Nhiều phần mềm nghiệp vụ là phần mềm xử lý dữ liệu mà được dẫn xuất bởi dữ liệu là chủ yếu. Chúng bị điều khiển bởi dữ liệu nhập chứ ít xử lý sự kiện từ ngoài.
- Mô hình dẫn xuất dữ liệu cho thấy trình tự các hoạt động liên quan đến việc xử lý dữ liệu nhập và tạo ra kết xuất tương ứng.
- Mô hình dẫn xuất dữ liệu đặc biệt hữu dụng trong khi phân tích các yêu cầu vì chúng có thể được dùng để trình bày việc xử lý end-to-end trong phần mềm.



# Mô hình hoạt động của “bơm insulin”



Dữ liệu



Hoạt động



## Mô hình dẫn xuất sự kiện (Event-driven)

- Các phần mềm real-time thường được dẫn xuất sự kiện chứ xử lý dữ liệu rất ít. Thí dụ, phần mềm chuyển kênh điện thoại đáp ứng với sự kiện “receiver off hook” bằng cách tạo tín hiệu quay số.
- Mô hình dẫn xuất sự kiện trình bày cách phần mềm đáp ứng với các sự kiện bên trong hay bên ngoài phần mềm.
- Nó dựa trên giả thuyết là phần mềm có 1 số trạng thái hữu hạn và sự kiện có thể chuyển phần mềm từ trạng thái này sang trạng thái khác.



# Mô hình máy trạng thái (State machine)

- Mô hình này sẽ miêu tả hành vi phần mềm khi đáp ứng với các sự kiện trong/ngoài phần mềm.
- Nó cho thấy các đáp ứng của phần mềm với các kích thích, do đó thường được dùng cho các phần mềm thời gian thực.
- Mô hình máy trạng thái dùng :
  - các nút miêu tả các trạng thái của phần mềm
  - Các cung chuyển có mũi tên nối giữa 2 nút miêu tả các sự kiện. Khi sự kiện xảy ra, phần mềm sẽ chuyển từ trạng thái này sang trạng thái kia.
- Lược đồ trạng thái trong UML được dùng để miêu tả mô hình máy trạng thái.



# Kết chương

- **Mô hình** là 1 góc nhìn trừu tượng về phần mềm, nó bỏ qua các chi tiết. Ta thường xây dựng nhiều mô hình khác nhau để miêu tả phần mềm theo nhiều góc nhìn khác nhau.
- **Mô hình ngữ cảnh (hay mô hình use-case)** miêu tả vị trí của phần mềm trong ngữ cảnh xung quanh nó, những actor nào có liên quan đến phần mềm, chúng tương tác với phần mềm thông qua các chức năng gì ? Lược đồ use-case và lược đồ trình tự được dùng chủ yếu để miêu tả các tương tác giữa actor với phần mềm.
- **Mô hình cấu trúc** trình bày tổ chức và kiến trúc phần mềm. Lược đồ class ở cấp phân tích, lược đồ class ở cấp thiết kế, lược đồ thành phần, lược đồ đối tượng, lược đồ triển khai cho chúng ta mô hình cấu trúc phần mềm theo nhiều mức độ trừu tượng khác nhau.





# Kết chương

- **Mô hình hành vi** trình bày góc nhìn động của phần mềm khi chạy. Các lược đồ trình tự, lược đồ trạng thái, lược đồ hoạt động, lược đồ cộng tác giúp ta đặc tả được mô hình hành vi theo nhiều góc nhìn khác nhau.
- **Kỹ thuật xây dựng phần mềm được dẫn xuất từ mô hình (Model-driven engineering)** là cách tiếp cận mới, ở đó con người chỉ cần đặc tả các mô hình phần mềm, tiện ích máy tính sẽ dịch tự động các mô hình ra phần mềm chạy được.