

Công nghệ phần mềm

Bài 05: Mô hình hóa hệ thống

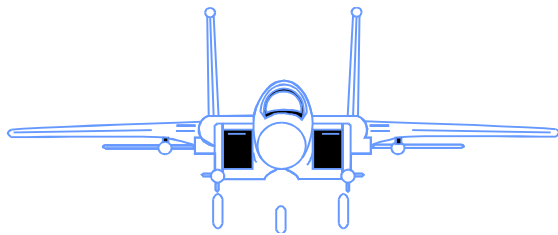
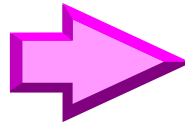
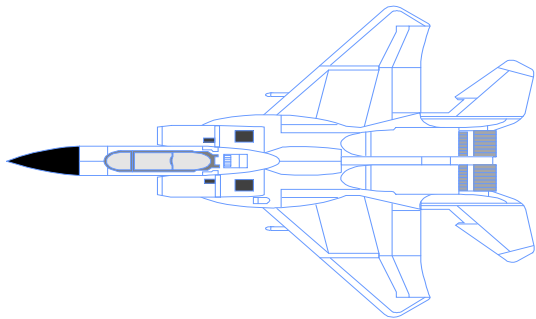
Nội dung

- Khái niệm mô hình hóa hệ thống
- Ngôn ngữ mô hình hóa UML
- Các góc nhìn hệ thống
- Một số loại mô hình hệ thống
 - Mô hình ngữ cảnh
 - Mô hình tương tác
 - Mô hình cấu trúc
 - Mô hình hành vi
- Kỹ nghệ hướng mô hình

Khái niệm cơ bản về hệ thống

- Hệ thống bao gồm các thành phần (thiết bị, phần mềm, con người), hoạt động trong môi trường, và hướng mục đích
- Hệ thống hiện thời (system-as-is): đang vận hành, có vấn đề hoặc nhu cầu cần cải tiến (cơ hội mới)
- Hệ thống được phát triển (system-to-be): cần được phát triển, tiến hóa từ hệ thống hiện thời (giải pháp dựa vào phần mềm – software-to-be).

Mô hình hóa hệ thống



- Mô hình hóa hệ thống: sự đơn giản hóa của chủ thể trong thế giới thực

Mục đích của mô hình hóa hệ thống

- Để hiểu tổng thể hệ thống (chức năng, cấu trúc) và để giao tiếp giữa các bên liên quan
- Mô hình hệ thống hiện thời: giúp cho thu thập và phân tích yêu cầu (kỹ nghệ yêu cầu)
- Mô hình hệ thống mới: (phân tích, thiết kế, cài đặt)
 - đặc tả yêu cầu hệ thống
 - đề xuất thiết kế hệ thống
 - làm tài liệu

Nguyên lý mô hình hóa hệ thống

- Mô hình ảnh hưởng đến cách vấn đề được giải quyết
- Mọi mô hình có thể được biểu diễn ở các mức độ chính xác khác nhau.
- Mô hình tốt nhất là khi được kết nối với thực tế.
- Sử dụng một mô hình duy nhất thường không đủ.

Các góc nhìn hệ thống

- Góc nhìn từ bên ngoài: mô hình hóa ngữ cảnh và môi trường hệ thống
- Góc nhìn tương tác: giữa môi trường và hệ thống, hoặc giữa các thành phần hệ thống
- Góc nhìn cấu trúc: mô hình hóa tổ chức hệ thống, hoặc biểu diễn cấu trúc của dữ liệu hệ thống
- Góc nhìn hành vi: mô hình hóa hành vi động của hệ thống, và cách mà hệ thống phản hồi các sự kiện

Các loại biểu đồ UML

- Biểu đồ hoạt động: biểu diễn các hoạt động diễn ra trong một tiến trình hay một xử lý dữ liệu
- Biểu đồ ca sử dụng: biểu diễn chuỗi tương tác giữa hệ thống và môi trường
- Biểu đồ tuần tự: biểu diễn tương tác giữa tác nhân và hệ thống, hay giữa các thành phần hệ thống
- Biểu đồ lớp: biểu diễn các lớp đối tượng và quan hệ giữa chúng
- Biểu đồ trạng thái: biểu diễn cách hệ thống phản ứng với các sự kiện từ bên trong và bên ngoài

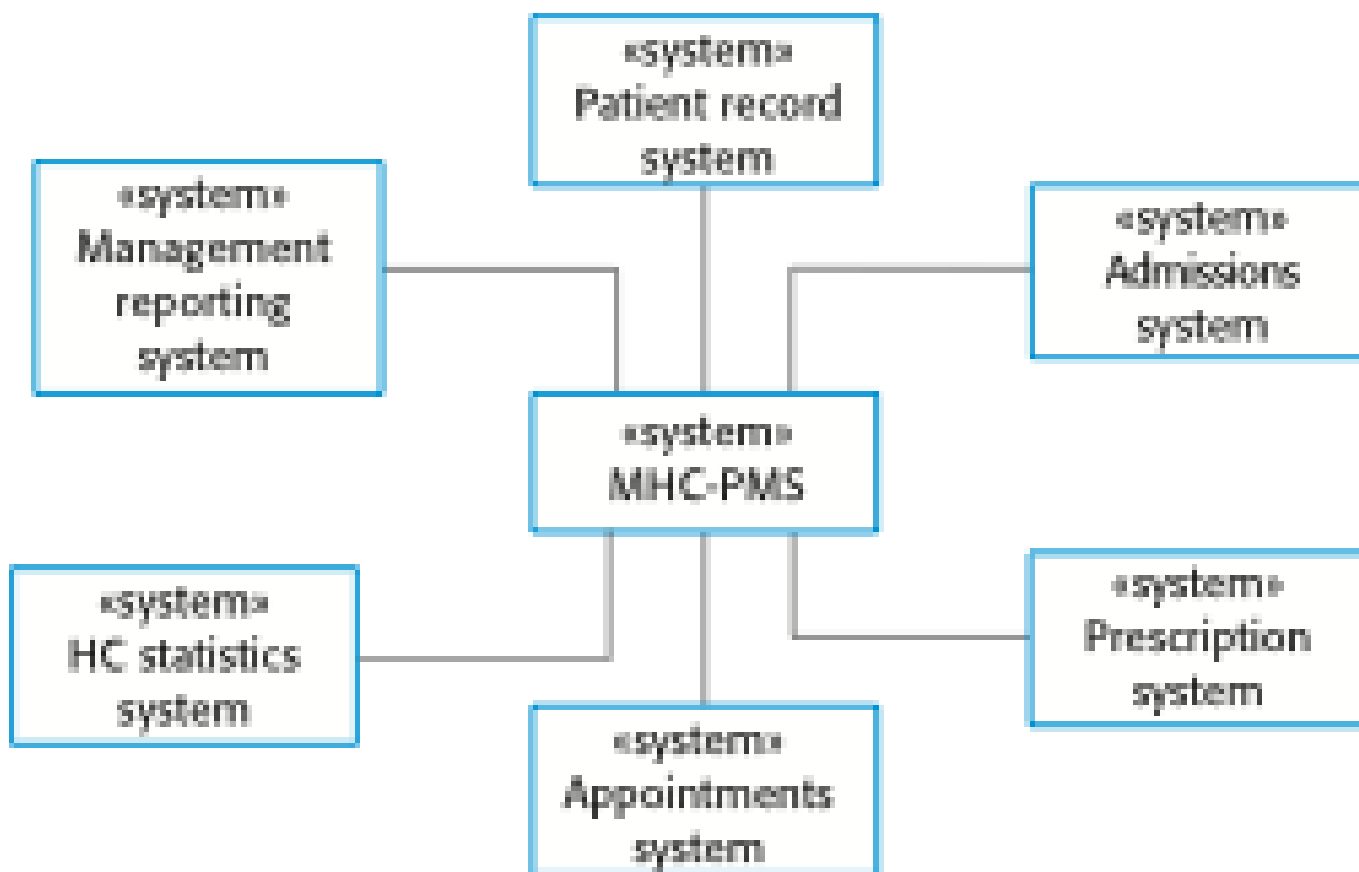
Mô hình ngữ cảnh

- Được dùng để minh họa ngữ cảnh hoạt động của hệ thống (các yếu tố làm nên biên hệ thống)
- Việc xác định biên hệ thống chịu ảnh hưởng của các yếu tố tổ chức và xã hội.
- Là một dạng mô hình kiến trúc chỉ mối quan hệ của hệ thống với các hệ thống khác.

Biên hệ thống

- Biên hệ thống xác định những gì thuộc về hệ thống và những gì thuộc về môi trường
- Biên hệ thống ảnh hưởng sâu sắc đến yêu cầu của hệ thống
- Xác định biên hệ thống mang yếu tố chính trị
- Áp lực để xác định hệ thống, qua đó làm tăng / giảm ảnh hưởng hoặc khối lượng công việc của các thành phần trong tổ chức.

Ví dụ mô hình ngữ cảnh

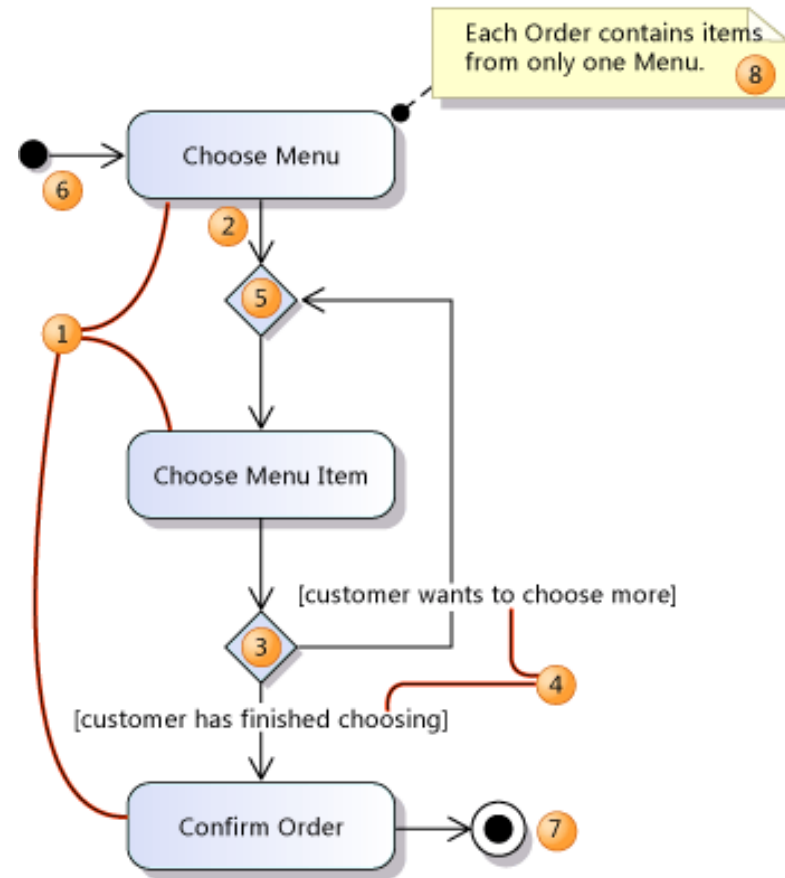


Mô hình quy trình

- Thường được dùng để hỗ trợ cho mô hình ngữ cảnh
- Mô tả sự tương tác giữa hệ thống với môi trường trong ngữ cảnh quy trình nghiệp vụ
- Biểu đồ hoạt động UML có thể được dùng để mô hình hóa quy trình nghiệp vụ

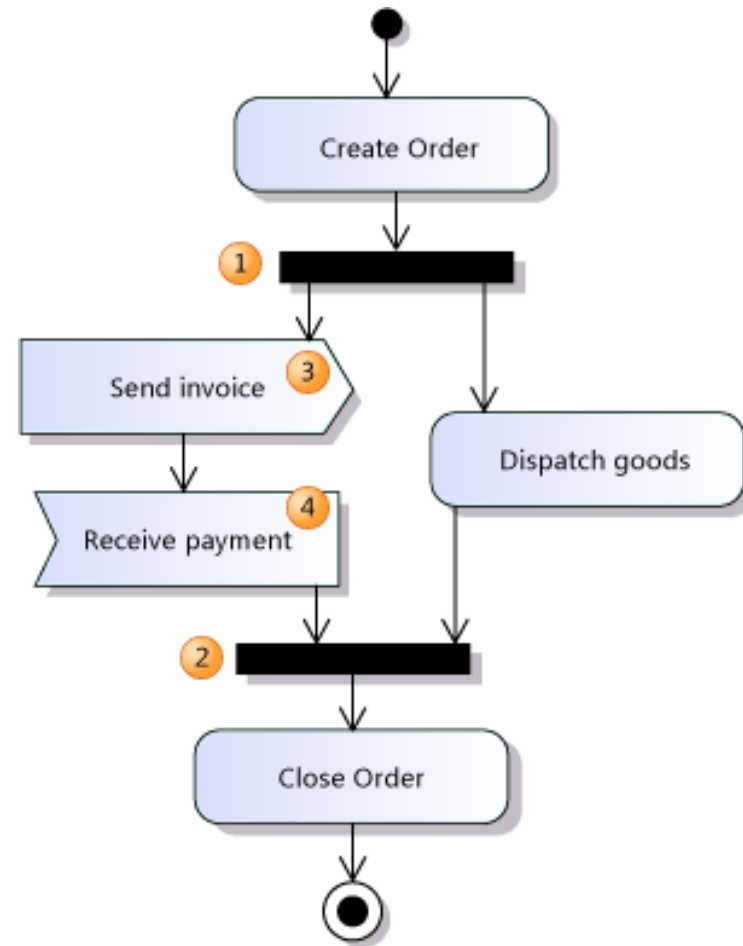
Biểu đồ hoạt động

- Tạo hành động **Action** (1) cho mỗi tác vụ được thực hiện bởi người dùng, hệ thống, hoặc cộng tác cả hai
- Kết nối chuỗi hoạt động bởi các **Connectors** (2)
- Sử dụng **Decision Node** (3) để quyết định bước tiếp theo.
- Đưa vào các điều kiện **guards** (4) để lựa chọn nhánh đi tiếp theo.
- Sử dụng **Merge Node** (5) để hợp nhất nhánh phụ với nhánh chính
- Nút khởi đầu **Initial Node** (6)
- Sử dụng **Activity Final Node** (7) để kết thúc hoạt động

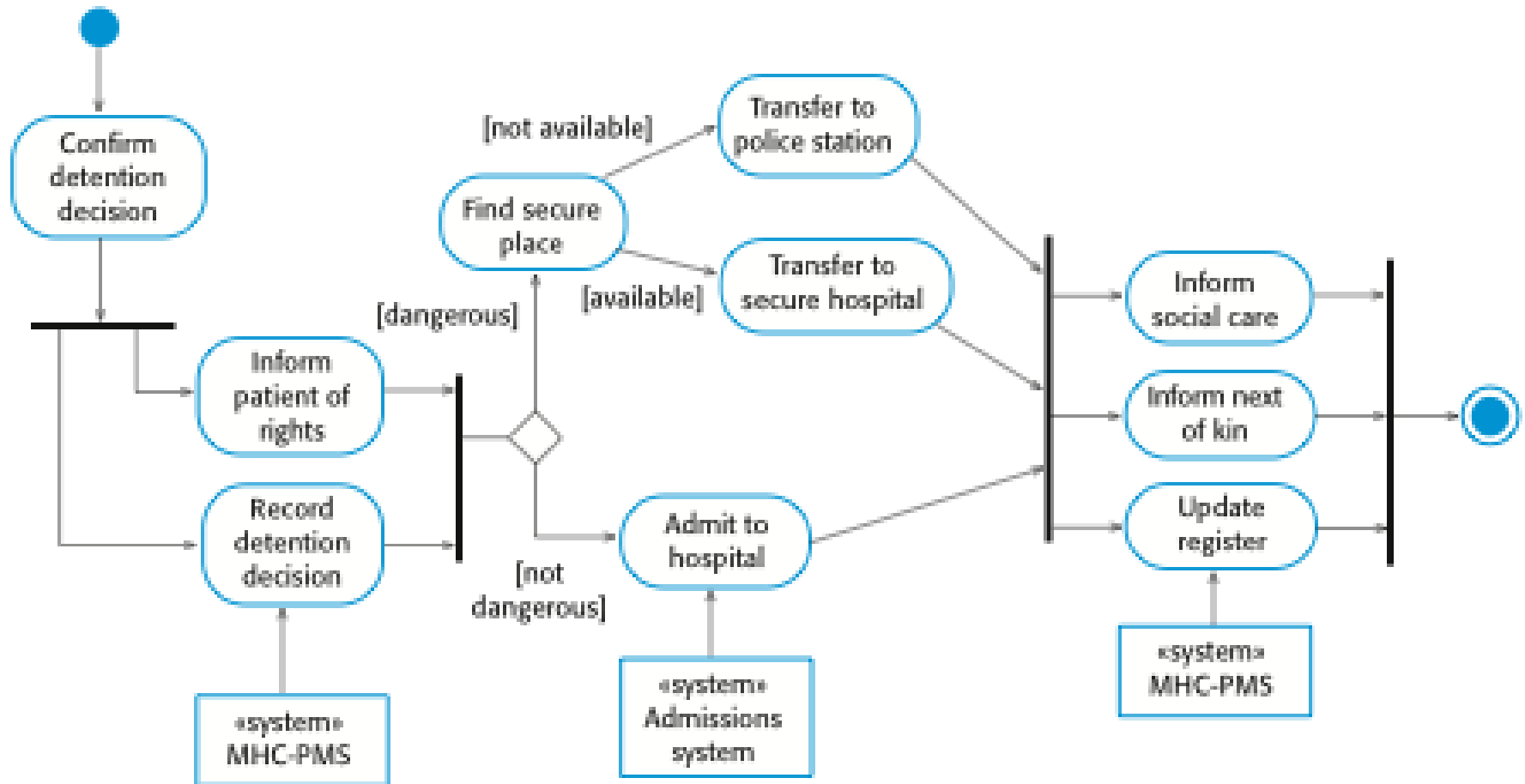


Biểu đồ hoạt động (tiếp)

- **Fork Node (1)** sẽ chia luồng điều khiển thành hai hay nhiều luồng. Khi hành động trước đó kết thúc, tất cả các hành động tương ứng đầu ra của fork node sẽ được kích hoạt.
- **Join Node (2)** sẽ hợp nhất các luồng đồng thời (concurrent threads) với nhau. Hành động sau **Join Node** chỉ được bắt khi tất cả các hành động dẫn đến **Join Node** đều kết thúc.
- Sử dụng **Send Signal Action (3)**, và nhận (4), để chỉ rằng một tín hiệu hoặc thông điệp được gửi tới hoặc nhận từ các hoạt động hay tiến trình khác.



Ví dụ mô hình quy trình



Mô hình tương tác

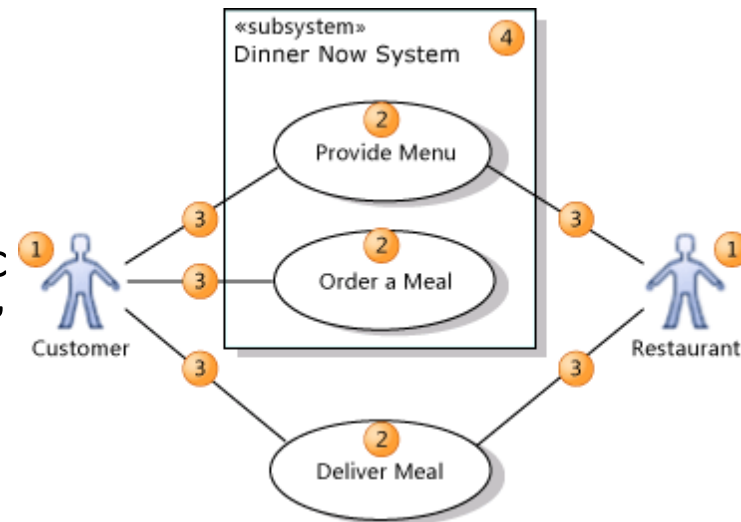
- Mô hình hóa tương tác người dùng giúp xác định các yêu cầu người dùng
- Mô hình hóa tương tác giữa hệ thống và hệ thống giúp làm rõ các vấn đề về giao tiếp (đặc tả giao thức tương tác)
- Mô hình hóa tương tác thành phần giúp hiểu cấu trúc của hệ thống và khả năng đáp ứng các yêu cầu phi chức năng (về hiệu năng và tính tin cậy.)
- Biểu đồ ca sử dụng và biểu đồ tuần tự thường được dùng để mô hình hóa tương tác

Mô hình ca sử dụng

- Ca sử dụng được đề xuất để đặc tả và cấu trúc các yêu cầu hệ thống.
- Mỗi ca sử dụng đại diện cho một tác vụ rời rạc có liên quan đến sự tương tác bên ngoài với một hệ thống.
- Các tác nhân tham gia ca sử dụng có thể là người hoặc các hệ thống khác.
- Mô hình ca sử dụng thường ở dạng kết hợp biểu đồ trực quan và các mô tả văn bản chi tiết.

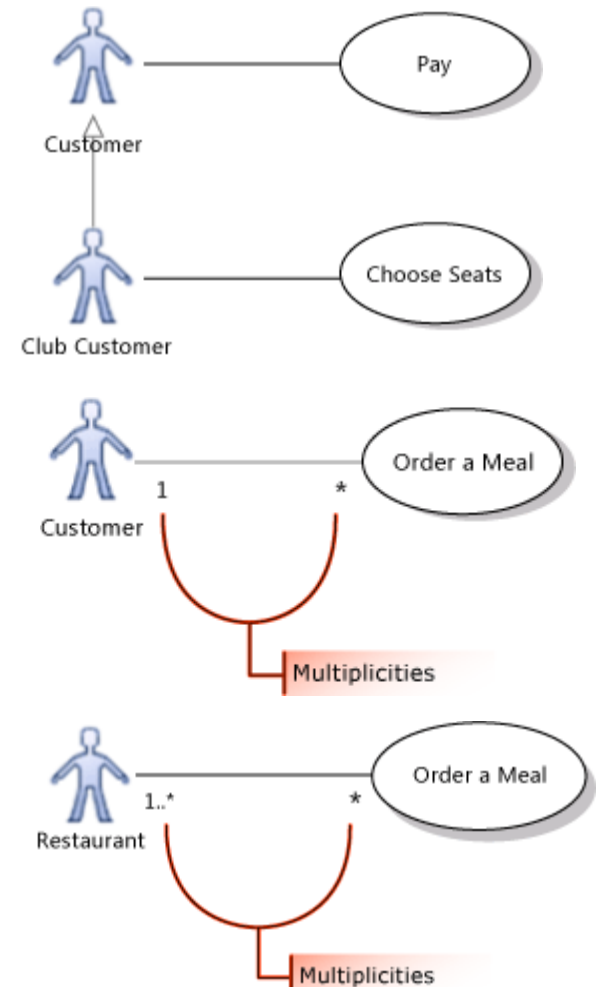
Biểu đồ ca sử dụng

- Tác nhân (1) là một lớp người, tổ chức, thiết bị hoặc thành phần phần mềm bên ngoài tương tác với hệ thống. Trong mô hình ca sử dụng này, các tác nhân bao gồm **Khách hàng** và **Nhà hàng**.
- Ca sử dụng (2) đại diện cho các hành động được thực hiện bởi một hoặc nhiều tác nhân trong việc theo đuổi một mục tiêu cụ thể. Các ca sử dụng ở đây bao gồm là **Đặt bữa ăn (Order Meal)**, **Cập nhật thực đơn (Update Menu)**, **Thanh toán (Process Payment)**.
- Trong biểu đồ ca sử dụng, các ca sử dụng được kết gán (3) với các tác nhân tham gia ca sử dụng.
- Hệ thống (4) là một thành phần phần mềm nhỏ, mà các tác nhân của nó chỉ là các thành phần phần mềm khác; hoặc nó có thể là một ứng dụng hoàn chỉnh.



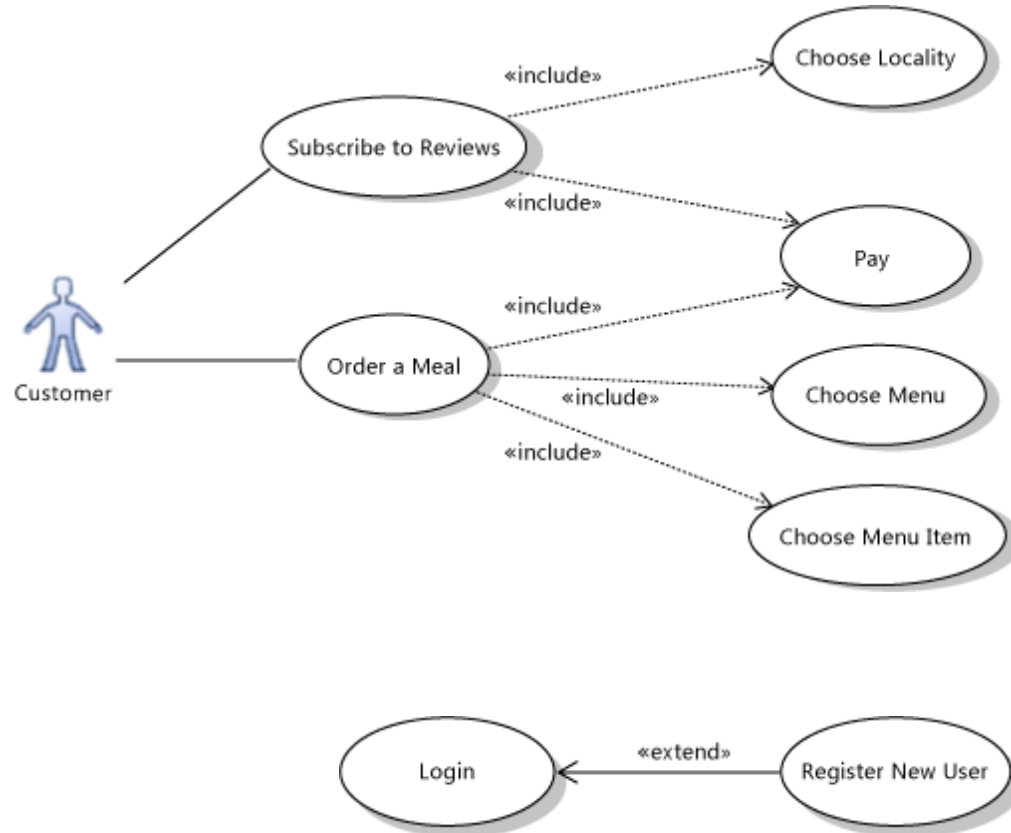
Biểu đồ ca sử dụng (2)

- Quan hệ thống quát hóa (**Generalization**) giữa các tác nhân. Ví dụ, tác nhân Club Customer kế thừa tác nhân Customer.
- Ràng buộc lực lượng giữa tác nhân và ca sử dụng có thể là 1, 1..*, 0..1, hoặc *.



Biểu đồ ca sử dụng (3)

- Quan hệ bao gộp (**Include**): một ca sử dụng diễn đạt một vài chi tiết của ca sử dụng khác. Ví dụ, ca sử dụng **Đặt bữa ăn (Order a Meal)** bao gộp ca sử dụng **Thanh toán (Pay)**, **Chọn thực đơn (Choose Menu)** và **Chọn món (Choose Menu Item)**.
- Quan hệ mở rộng (**extend**): một ca sử dụng có thể thêm chức năng cho ca sử dụng khác trong điều kiện nào đó. Ví dụ, ca sử dụng **Login** sẽ mở rộng sang ca sử dụng **Đăng ký (Register New User)** khi người dùng chưa có tài khoản.



Ví dụ ca sử dụng

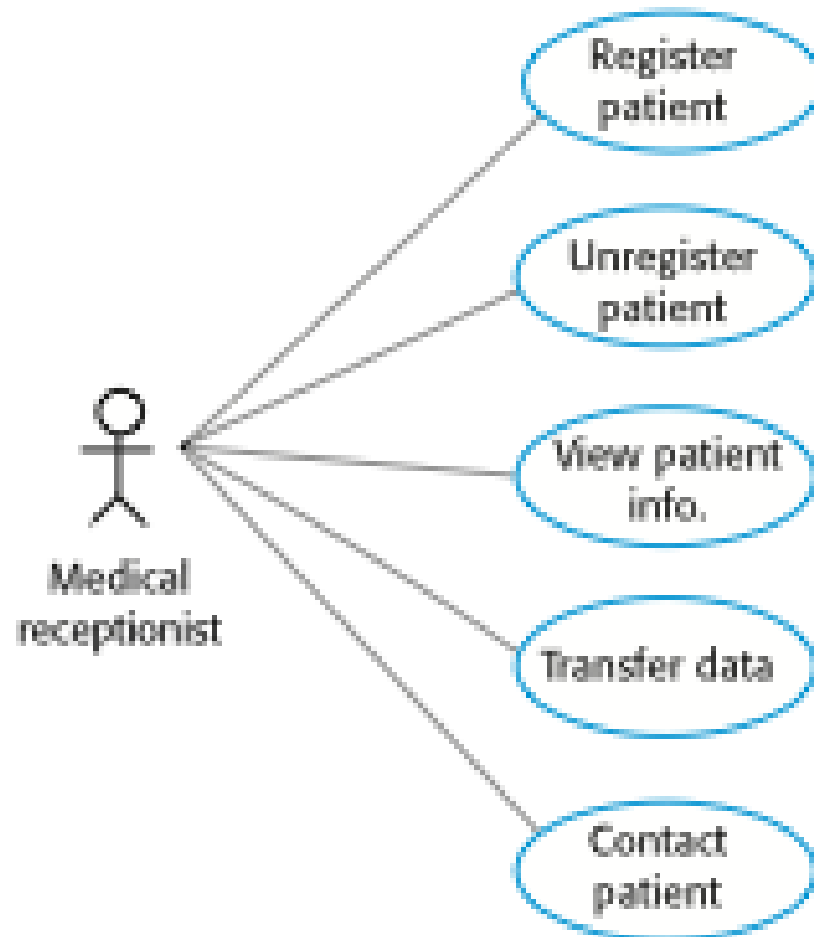
- Ví dụ ca sử dụng truyền dữ liệu trong hệ MHC-PMS



Mô tả ca sử dụng ở dạng bảng

MHC-PMS: Transfer data	
Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the MHC-PMS to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient's diagnosis and treatment.
Data	Patient's personal information, treatment summary
Stimulus	User command issued by medical receptionist
Response	Confirmation that PRS has been updated
Comments	The receptionist must have appropriate security permissions to access the patient information and the PRS.

Ví dụ mô hình ca sử dụng

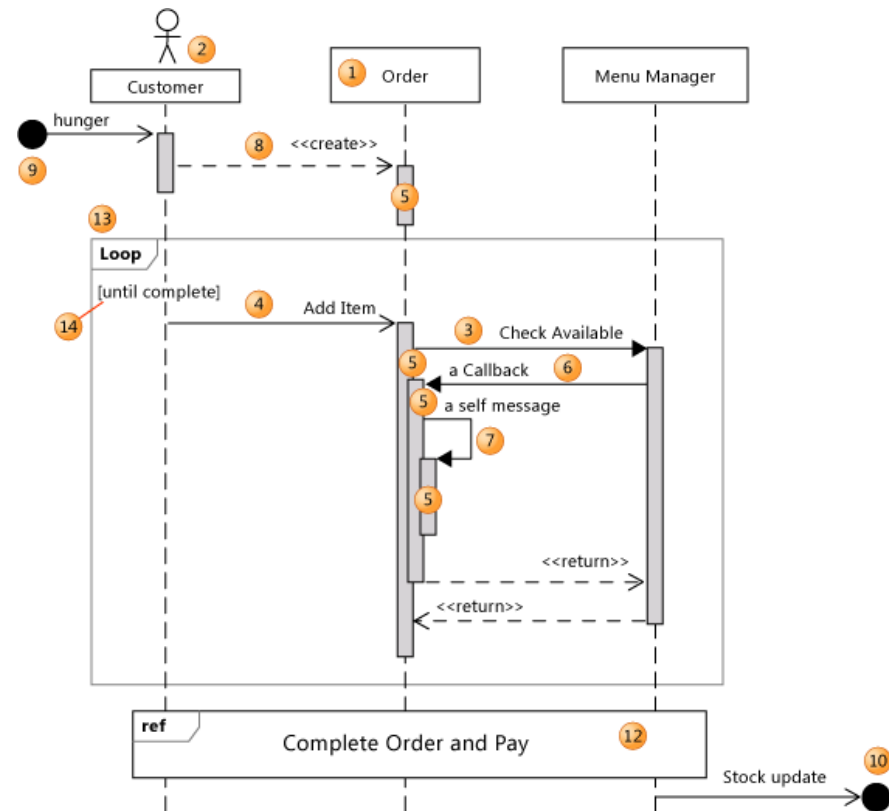


Biểu đồ tuần tự

- Để mô hình hóa tương tác giữa tác nhân và các đối tượng bên trong hệ thống
- Để đặc tả các kịch bản sử dụng

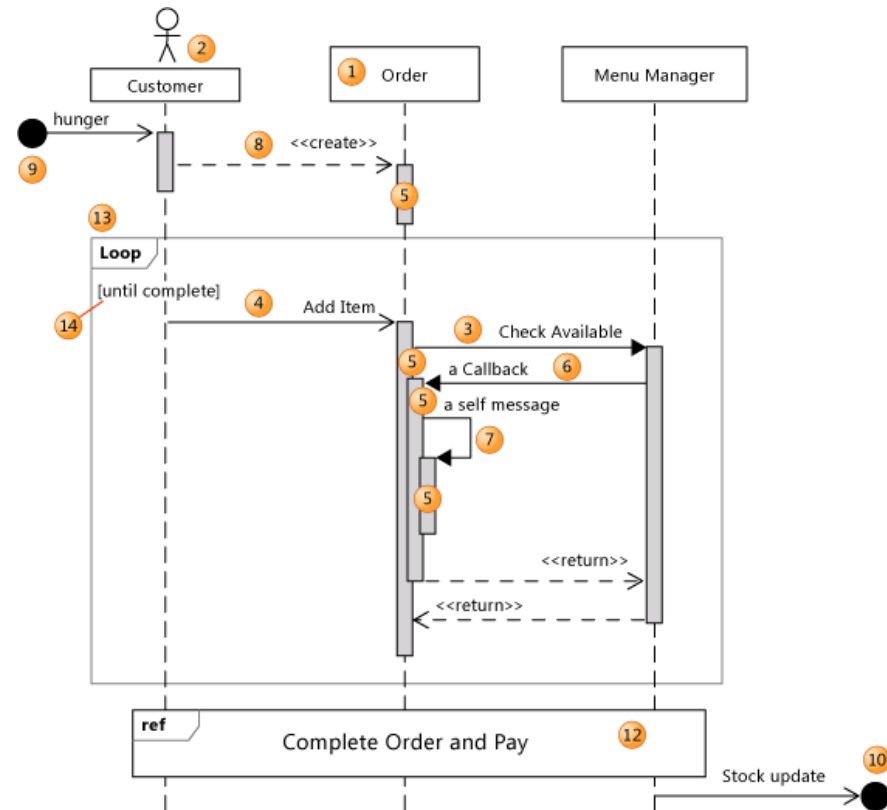
Biểu đồ tuần tự (2)

- Đường sông (**Lifelines**) (1) biểu diễn các thể hiện của lớp, thành phần, tác nhân hay các thiết bị.
- Các thông điệp (3, 4, 6, 7) được gửi từ đối tượng gửi đến đối tượng nhận.
- Một khoảng thực thi (5) xuất hiện ở đối tượng nhận thông điệp.
- Ký pháp (9) để chỉ tình huống một thông điệp được gửi từ nguồn không xác định, và được gửi một cách không đồng bộ.

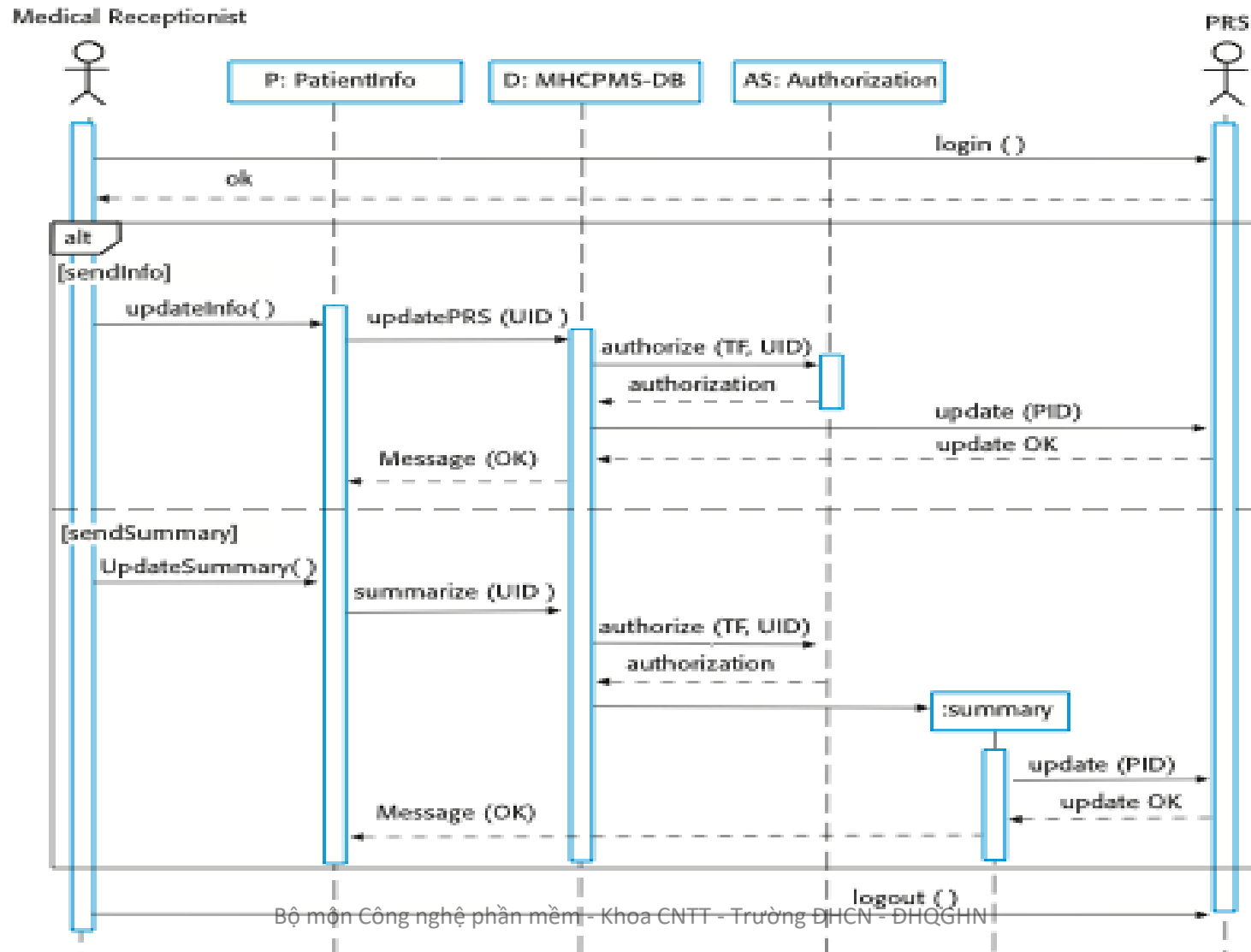


Biểu đồ tuần tự (3)

- Thông điệp đồng bộ (**Synchronous**) (3) mô tả tương tác trong đó đối tượng gửi chờ cho đến khi nhận được phản hồi của đối tượng nhận.
- Mũi tên **<<return>>** là tín hiệu từ đối tượng nhận, để kết thúc một thực thi.
- Mũi tên bất đồng bộ (**Asynchronous**) (4) để chỉ người gửi có thể tiếp tục công việc mà không phải chờ phản hồi từ người nhận.
- Tín hiệu **<<Create>>** (8) để mô tả hành động đối tượng gửi tạo đối tượng nhận.



Ví dụ biểu đồ tuần tự - truyền dữ liệu

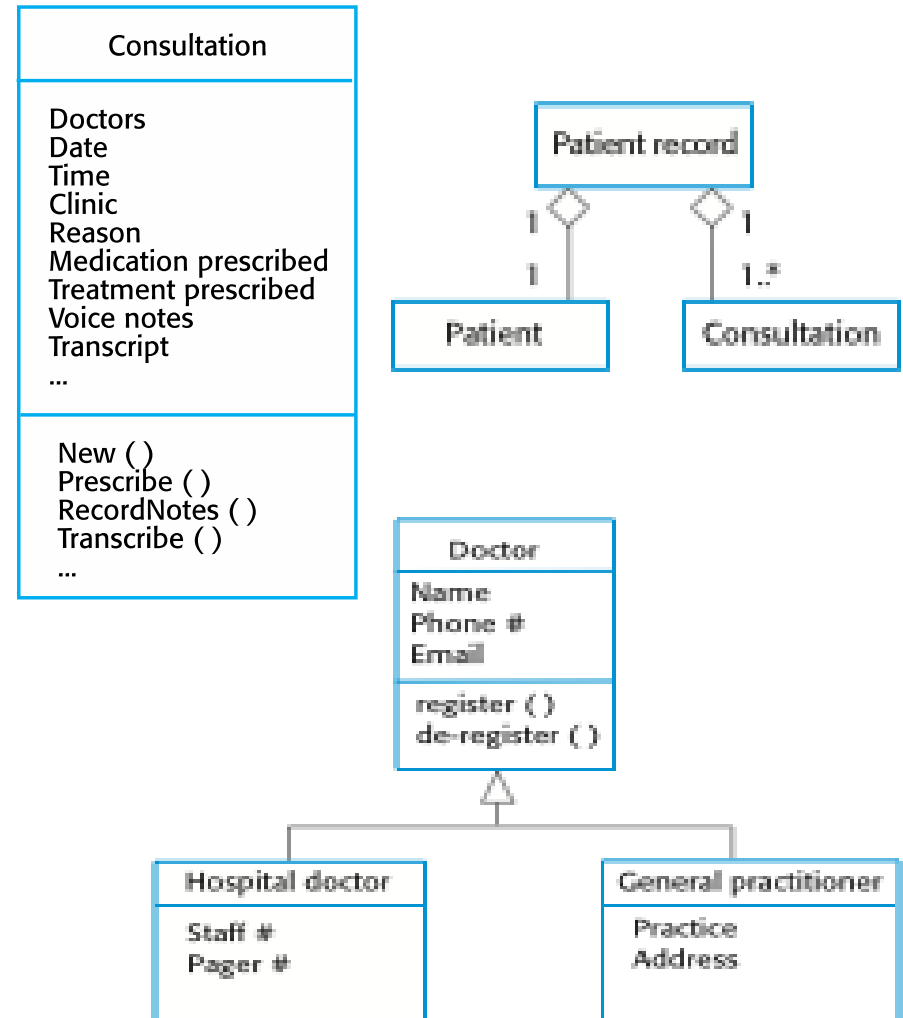


Mô hình cấu trúc

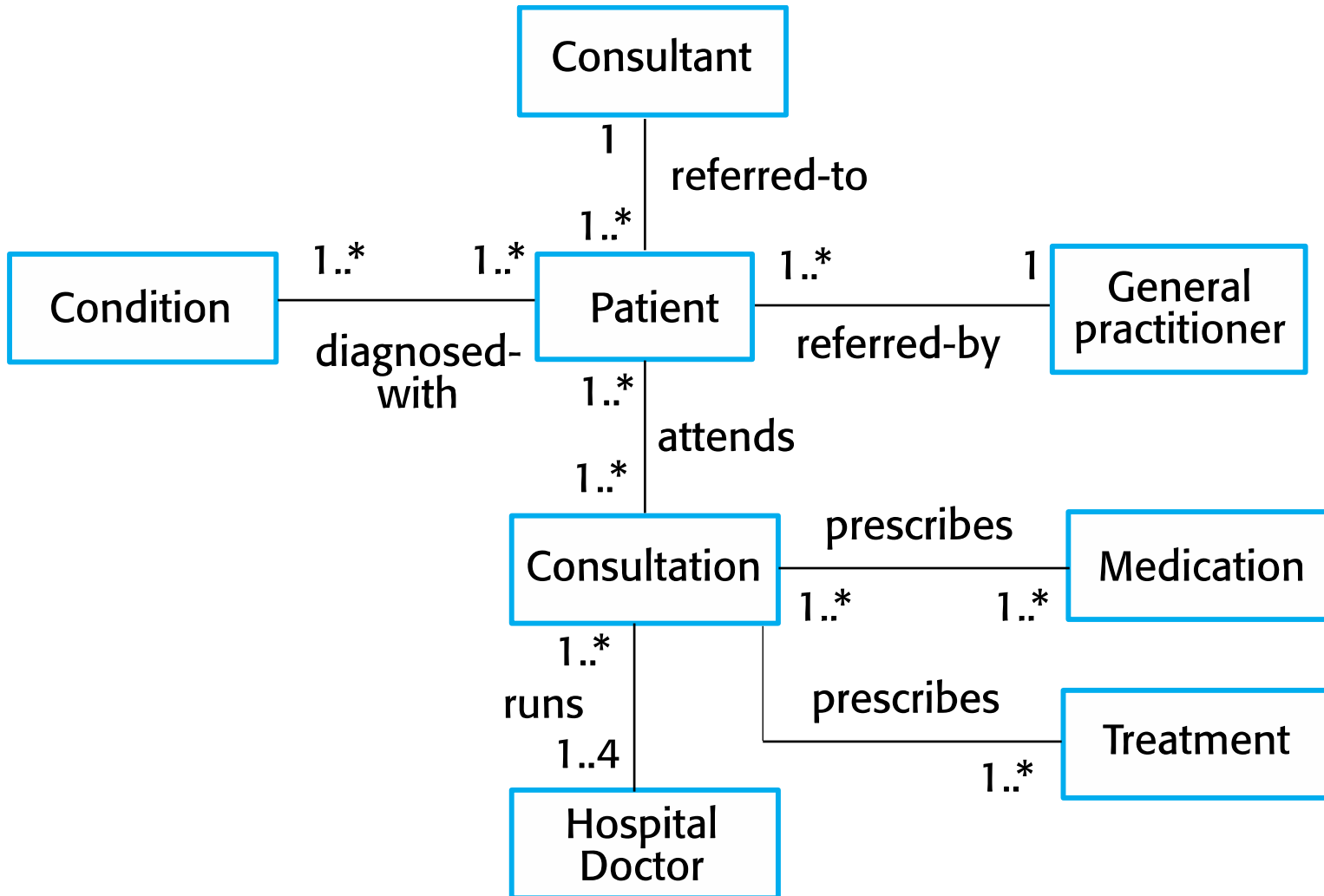
- Được dùng để biểu diễn cách tổ chức của hệ thống, chỉ ra quan hệ giữa các thành phần hệ thống
- Mô hình cấu trúc ở dạng mô hình tĩnh khi mô tả cấu trúc của thiết kế hệ thống hoặc ở dạng mô hình động khi phản ánh cấu trúc hệ thống khi thực thi
- Mô hình này thường được dùng khi thảo luận và thiết kế kiến trúc hệ thống

Biểu đồ lớp

- Được dùng để biểu diễn mô hình cấu trúc hệ thống
- Lớp: tập các đối tượng (thể hiện) cùng đặc điểm, hành vi và ngữ nghĩa
- Quan hệ kết hợp: quan hệ ngữ nghĩa giữa các lớp
- Tổng quát hóa: Lớp con chi tiết hóa lớp cha
- Quan hệ tụ hợp (hợp thành): quan hệ giữa tổng thể và bộ phận, có (không có) thành phần chia sẻ.



Ví dụ biểu đồ lớp



Mô hình hành vi

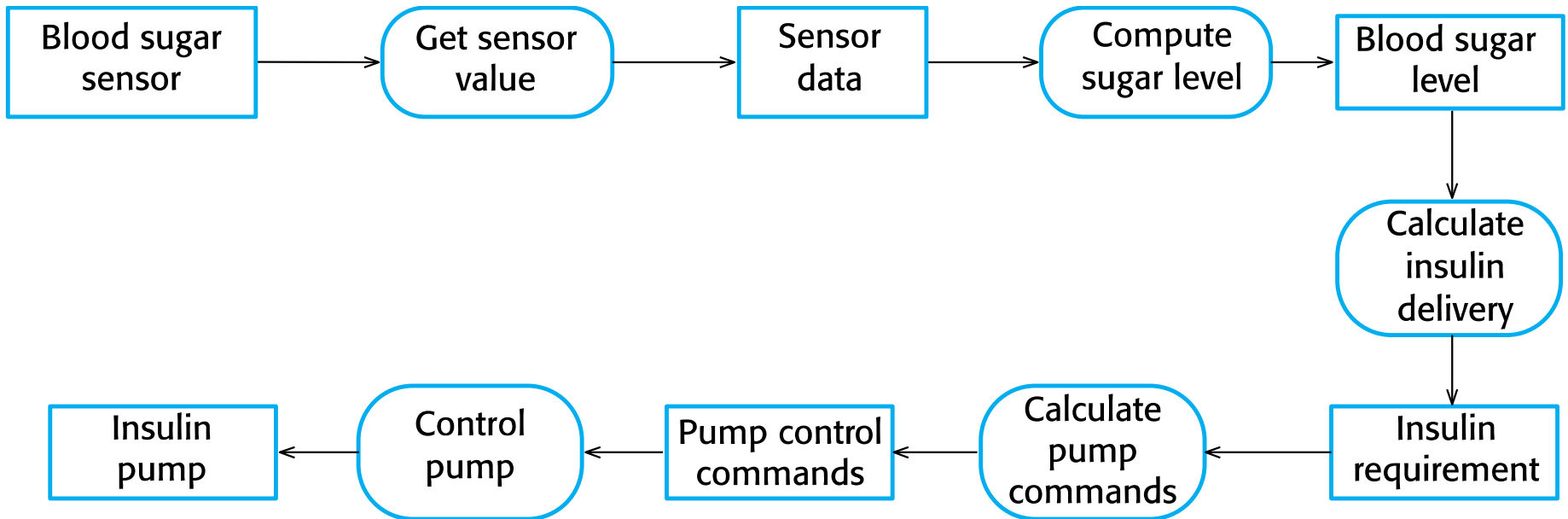
- Mô hình hành vi mô tả hành vi động của hệ thống khi nó thực hiện, qua đó làm rõ cách thức hệ thống phản hồi các kích thích từ môi trường
- Kích thích từ môi trường có 2 dạng:
 - Dữ liệu: được gửi đến và được hệ thống xử lý
 - Sự kiện: kích hoạt các xử lý của hệ thống

Mô hình hóa hướng dữ liệu

- Thường dùng cho các hệ thống nghiệp vụ (business system)
- Mô hình hướng dữ liệu (data-driven models) biểu diễn chuỗi hoạt động xử lý các dữ liệu đầu vào và kết xuất dữ liệu đầu ra tương ứng.
- Được hỗ trợ bởi biểu đồ hoạt động UML 2.0 hoặc biểu đồ tuần tự (ở dạng không tường minh)

Ví dụ mô hình hóa hướng dữ liệu

Biểu đồ luồng dữ liệu (DFD) cho thao tác tiêm Insulin



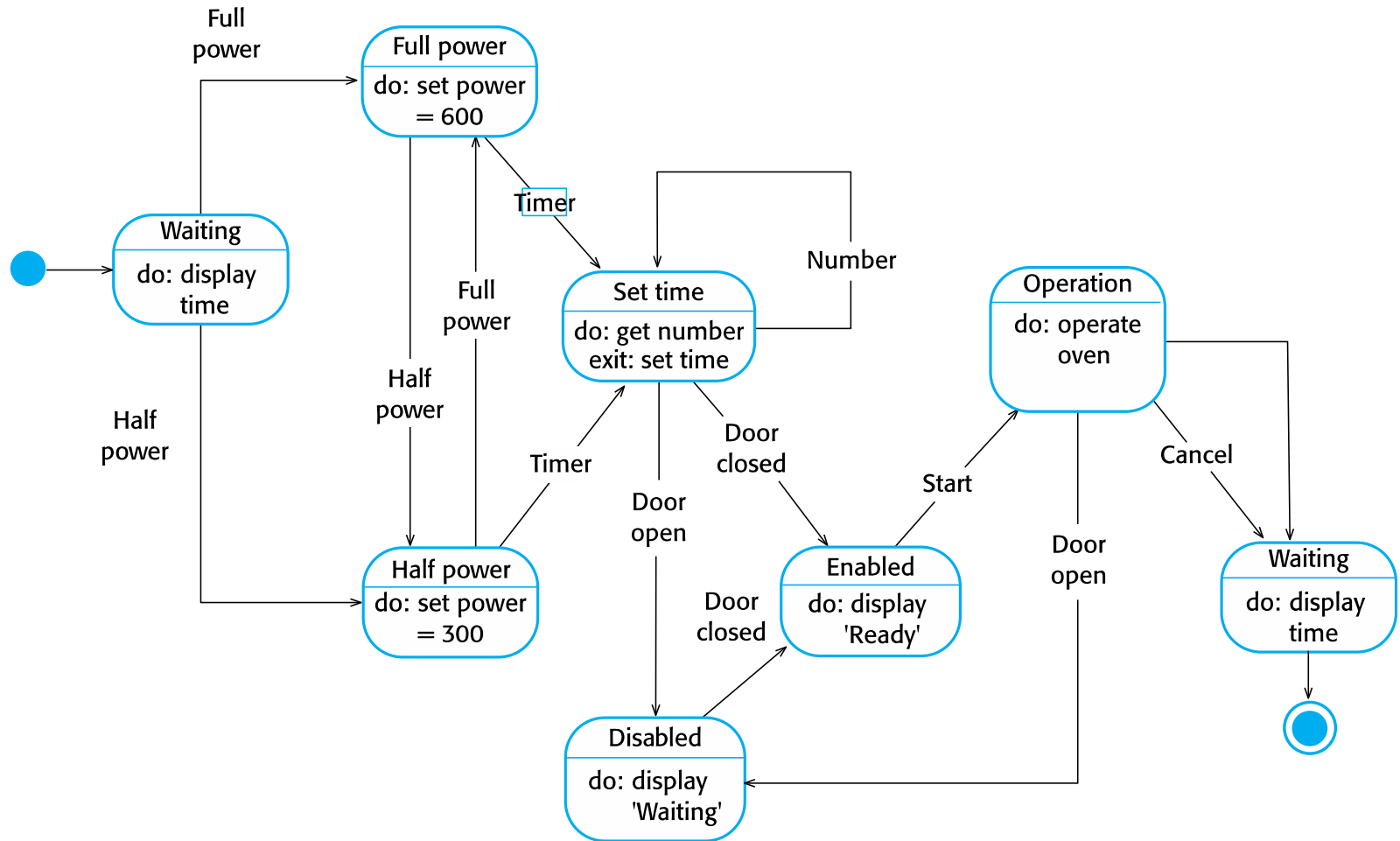
Mô hình hóa hướng sự kiện

- Mô hình hướng sự kiện (event-driven models) biểu diễn cách mà hệ thống phản hồi các sự kiện từ bên ngoài và bên trong.
- Dựa vào giả định cơ bản: Hệ thống có một tập hữu hạn trạng thái và các sự kiện sẽ kích hoạt các chuyển từ trạng thái này sang trạng thái kế tiếp.
- Thường dùng cho các hệ thống thời gian thực (real-time system)

Mô hình máy trạng thái

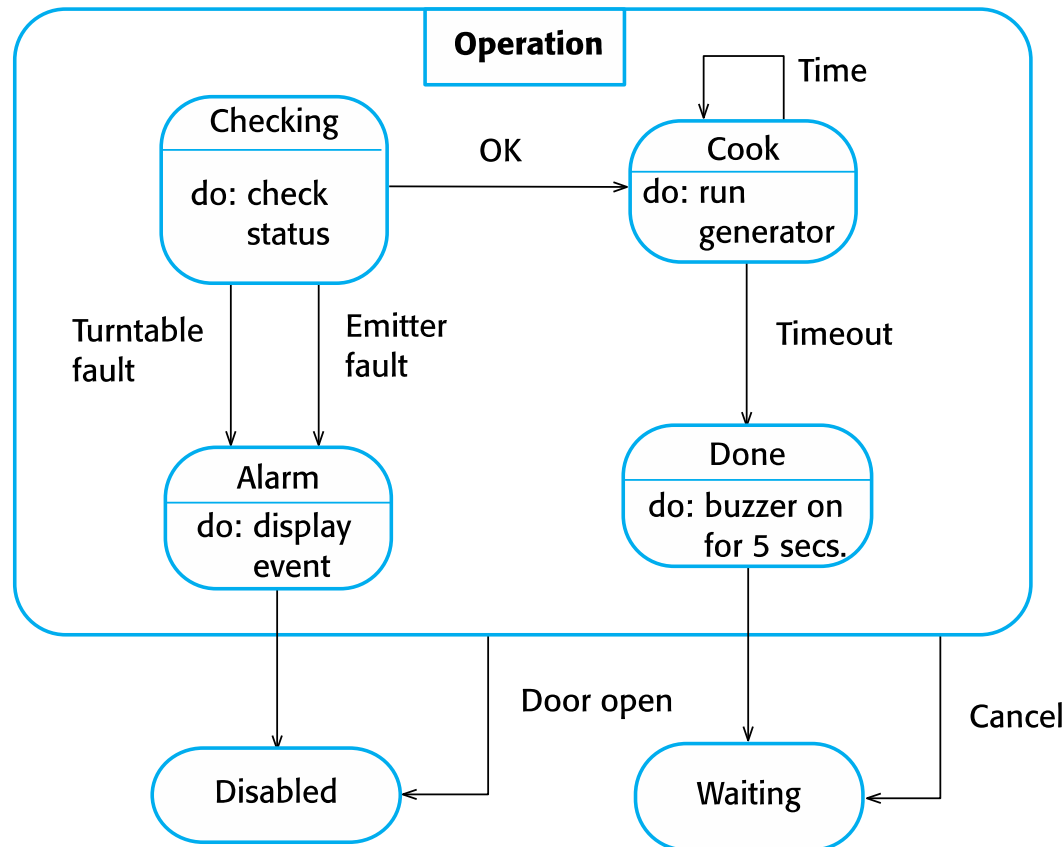
- Được dùng để mô hình hóa hành vi hệ thống. Biểu diễn cách mà hệ thống phản hồi các sự kiện từ bên ngoài và bên trong.
- Mô hình máy trạng thái (trạng thái hệ thống, các sự kiện, các chuyển)
- Biểu đồ trạng thái UML (statecharts) được dùng để biểu diễn các mô hình máy trạng thái.

Ví dụ mô hình máy trạng thái



Mô hình hoạt động của lò vi sóng

Ví dụ mô hình máy trạng thái (2)



Kỹ nghệ hướng mô hình

- Kỹ nghệ hướng mô hình là phương pháp phát triển phần mềm trong đó các mô hình được xem là các chế tác đầu ra cơ bản của quy trình phát triển (chương trình là một dạng mô hình đầu ra)
- Các chương trình tương ứng với nên tảng thực thi nào đó thường được sinh tự động từ các mô hình bởi các chuyển đổi mô hình.
- Hướng tiếp cận này cho phép các kỹ sư làm việc ở cấp độ trừu tượng cao => dễ giao tiếp với chuyên gia miền, giảm nỗ lực ánh xạ thiết kế sang các diễn đạt chi tiết ở cấp độ ngôn ngữ lập trình.

Đặc điểm kỹ nghệ hướng mô hình

- Ưu điểm
 - xem xét hệ thống ở cấp độ trừu tượng cao
 - sinh mã tự động
- Nhược điểm
 - Sai khác giữa các trừu tượng mức mô hình và mức cài đặt
 - chi phí để tạo ra các bộ chuyển đổi mô hình cũng như các bộ sinh mã tự động

Tổng kết

- Khái niệm mô hình hóa hệ thống
- Ngôn ngữ mô hình hóa UML
- Các góc nhìn hệ thống phần mềm
- Một số loại mô hình hệ thống
- Kỹ nghệ hướng mô hình