Mô hình hóa hướng đối tượng với UML

Tài liệu

- Zhiming Liu, *Object-Oriented Software Development with UML*, UNU/IIST Report
- http://uml.org/

Mô hình hóa

- Mô hình là kết quả của phân tích và thiết kế
- Là sự đơn giản hóa thực tế, cho phép hiểu rõ hơn hệ thống cần phát triển
- Mô hình cho phép:
  - Cái nhìn trực quan về hệ thống đang có hoặc hướng tới
  - Kiểm chứng hệ thống bởi khách hàng
  - Cung cấp những chỉ dẫn để xây dựng hệ thống
  - Tài liệu hóa hệ thống

Mô hình hóa

- Tồn tại nhiều cách mô hình hóa một hệ thống
  - Mô hình phù hợp sẽ làm cho việc giải bài toán dễ hơn
- Có nhiều mức chính xác của mô hình
  - Mô hình trừu tượng -> làm mịn -> mô hình chi tiết
- Không có mô hình nào là đầy đủ
  - Cần tiếp cận (hiểu) hệ thống thông qua nhiều mô hình khác nhau
- Mô hình tốt phải là mô hình phù hợp với thế giới thực

Mô hình hóa hướng đối tượng

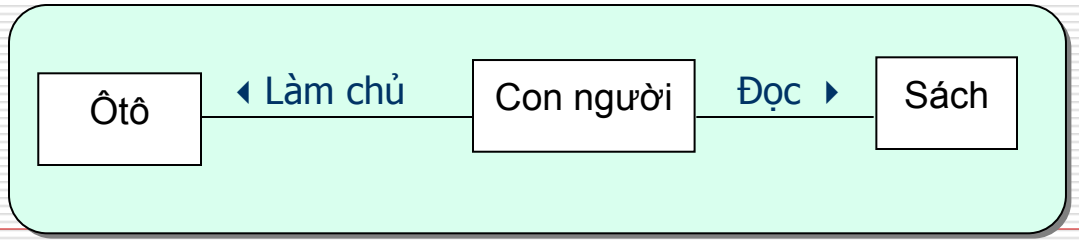
- Tăng tính độc lập của mô hình với các chức năng yêu cầu
- Dễ dàng hơn trong việc thay đổi hoặc thêm bớt các chức năng
- Gần với thế giới thực

Ví dụ về mô hình

Thế giới thực

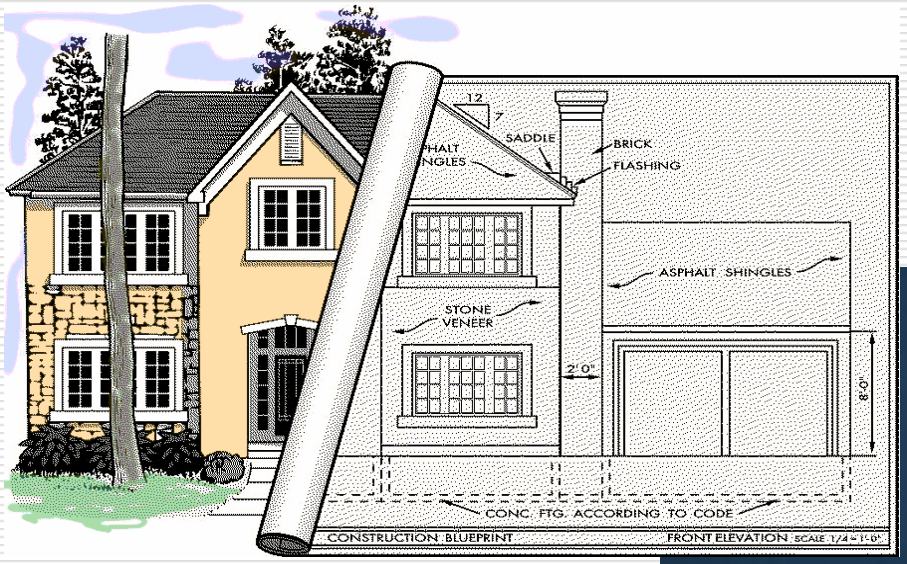
Mô hình: Quả địa cầu học sinh

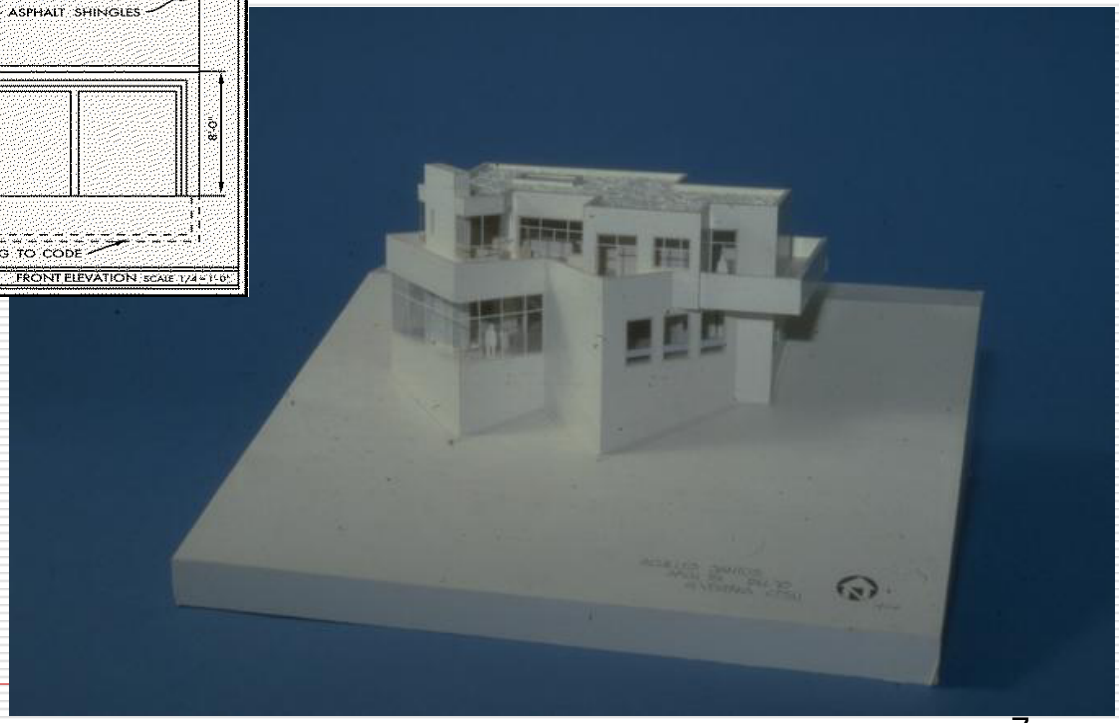
Thế giới thực



Mô hình

Ví dụ

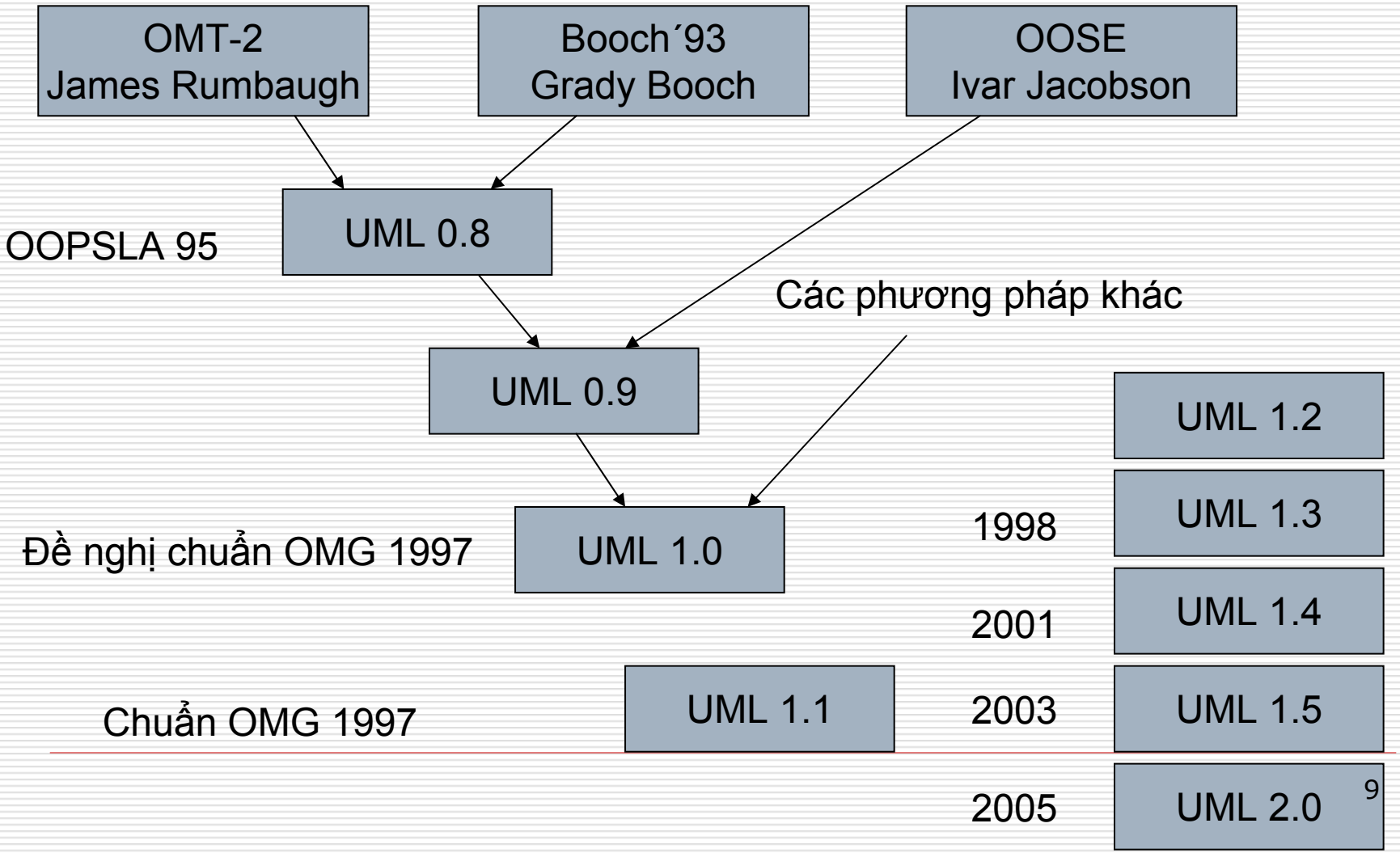

12
7
SADDLE
BRICK
FLASHING
ASPHALT SHINGLES
STONE VENEER
2' 0"
8'-0"
CONC. FTG. ACCORDING TO CODE
CONSTRUCTION BLUEPRINT
FRONT ELEVATION SCALE 1/4 = 1'-0"



Ngôn ngữ mô hình hóa thống nhất UML

- Là ngôn ngữ mô hình hóa hướng đối tượng
- Được thừa nhận như một chuẩn mặc định của ngành CNTT
  - Real-time UML, Agent - UML
- Có nhiều tools và phương pháp dựa trên UML
  - Rational rose
  - Rational Unified Process (RUP)

Unified Modeling Language



Mục đích của UML

- Biểu diễn toàn bộ hệ thống
- Tạo sự liên kết giữa các khái niệm (concepts) về hệ thống và các artifacts thực hiện được
- Vừa trực quan (hướng người dùng) vừa hỗ trợ khả năng tự động hóa (máy)
- Một ngôn ngữ chung:
  - Sử dụng được với nhiều phương pháp khác nhau
  - Đáp ứng tất cả các giai đoạn trong phát triển phần mềm

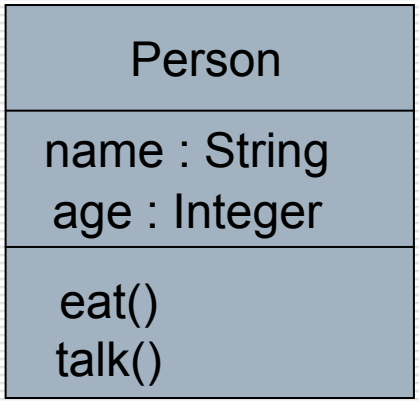
UML cho phép

- Hiển thị
  - Thông qua các ký pháp đồ họa có ngữ nghĩa xác định
- Đặc tả
  - Một cách chính xác và toàn diện
- Xây dựng
  - Các lớp, các quan hệ có thể được ánh xạ vào chương trình thực
- Làm tài liệu
  - Các biểu đồ, các ghi chú, các ràng buộc,…

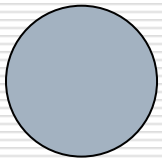
Phạm vi ứng dụng

- Hệ thống thông tin doanh nghiệp
- Ngân hàng và các dịch vụ tài chính
- Viễn thông
- Giao thông
- Quân sự và hàng không (hệ nhúng)
- Khoa học
- Các ứng dụng phân tán trên web

Ký pháp



Class

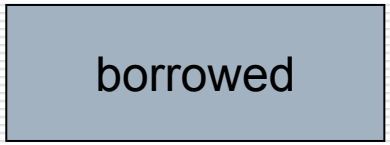


Comparable

Interface

Use case

Message



State

Ký pháp (tiếp)

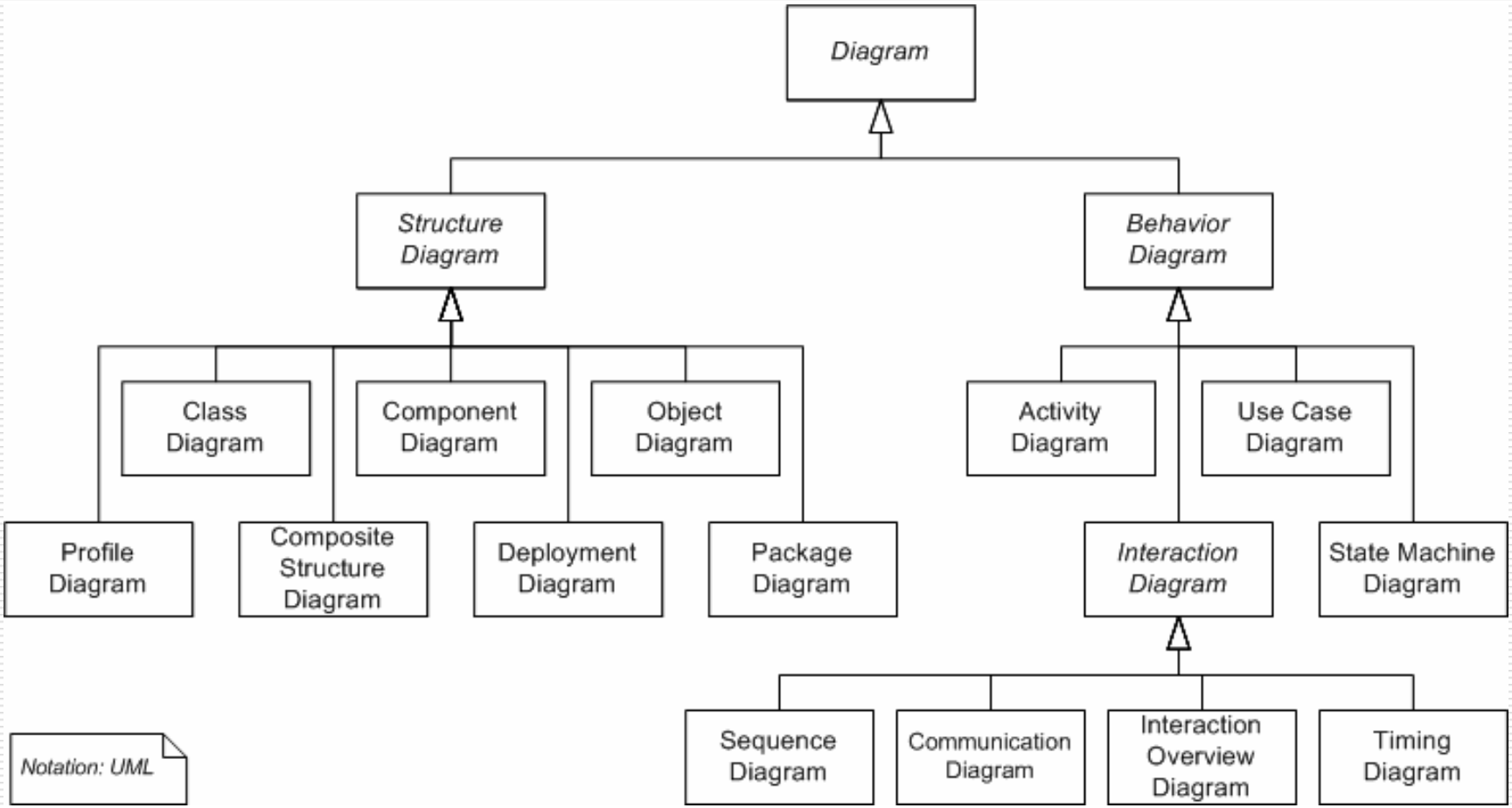
Dependence

Association

Inheritance

Realisation

Phân loại biểu đồ



Phân loại biểu đồ

Mô hình chức năng

Hệ thống làm gì

Biểu đồ ca sử dụng

Biểu đồ tuần tự
Biểu đồ tương tác
Biểu đồ trạng thái
Biểu đồ hoạt động

Mô hình cấu trúc

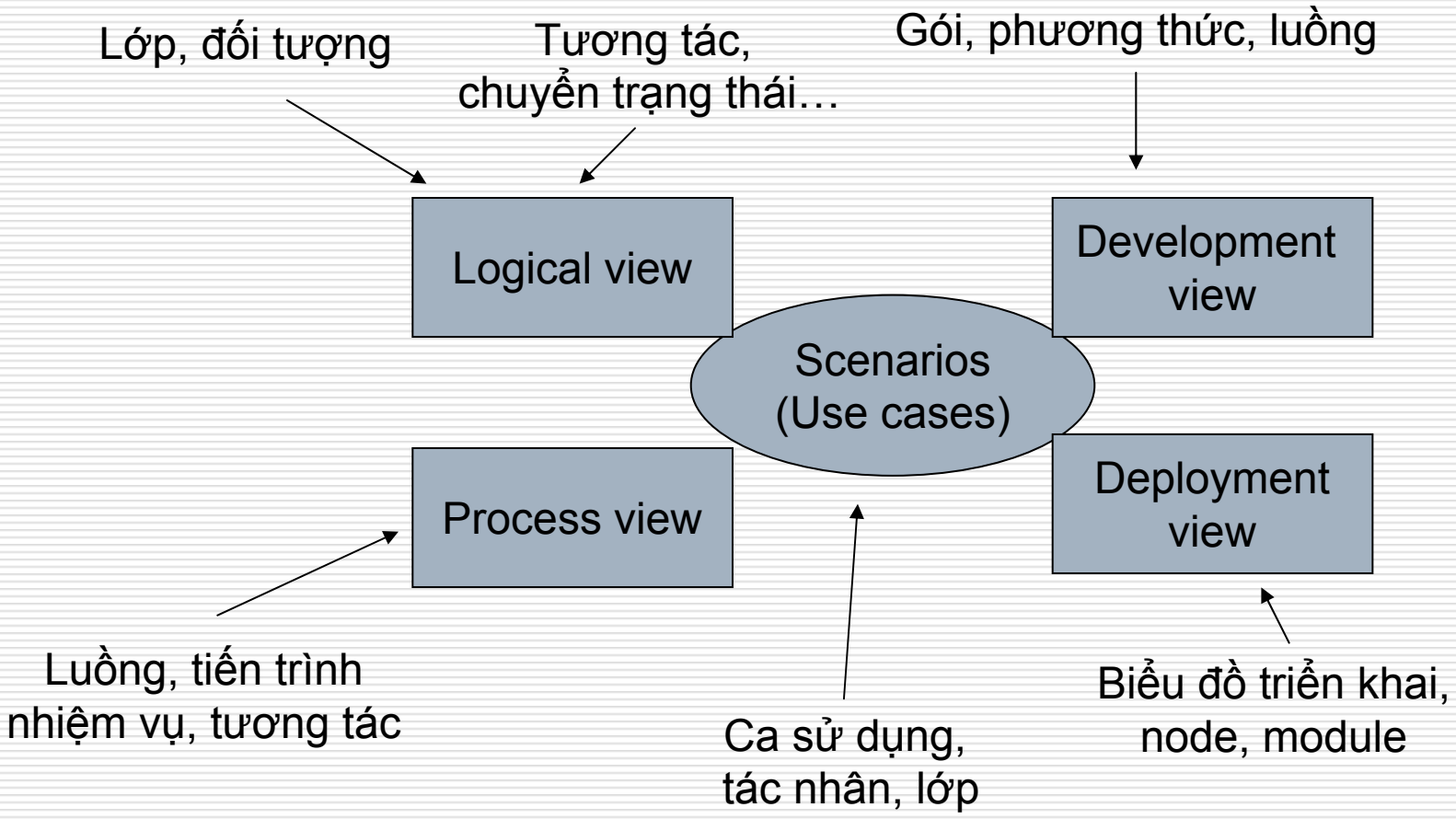
Hệ thống phản ứng với cái gì

Biểu đồ lớp và đối tượng

Mô hình thời gian

Thứ tự giữa các hành động trong hệ thống

4+1 view model



Cách nhìn ca sử dụng

- Nhìn hệ thống bởi những người dùng cuối
  - Tác nhân, chức năng, kịch bản
- Phân loại được các hành vi
  - Chức năng, độ ưu tiên
- Chỉ ra các ràng buộc giữa các hành vi
  - Sử dụng, kế thừa

Cách nhìn logic

- Cung cấp cái nhìn logic về chức năng/hành vi của hệ thống
- Nhìn nhận các đối tượng
  - Các lớp và đối tượng
  - Các gói
  - Các quan hệ: Kết nối, trừu tượng, đa hình, đồng nhất
- Nhìn nhận các trạng thái/luồng công việc
- Nhìn nhận về cách thức tương tác
  - Các kịch bản của các ca sử dụng

Cách nhìn tiến trình

- Phân rã dựa trên nhiệm vụ và tiến trình
  - Phân loại được các nhóm tiến trình
- Cung cấp các thông tin động về hoạt động của hệ thống:
  - Thứ tự thực hiện, cách thức tương tác, ràng buộc thứ tự
  - Phân tán/song song
  - Khả năng tính tích hợp, hiệu năng,…

Cách nhìn phát triển

- Phân rã theo module
- Nhóm thành các module bằng các gói
- Tổ chức thành các hệ thống con để:
  - Tăng độ chắc chắn
  - Giảm sự kết dính và nhìn thấy
- Giúp cho:
  - Dễ dàng phát triển
  - Nhìn thấy khả năng tái sử dụng
  - Quản lý cấu hình

Cách nhìn triển khai

- Phân rã hệ thống thành các nút triển khai
  - Vai trò của nút
  - Liên quan giữa các nút
- Cung cấp các thông tin:
  - Hiệu năng, tính sẵn sàng của hệ thống
  - Cách thức cài đặt, bảo trì

Ví dụ: Hệ thống quản lý thư viện

- Người dùng: Sinh viên (khách hàng), thủ thư (nhân viên)
- Chức năng
  - Tra cứu
  - Mượn
  - Trả

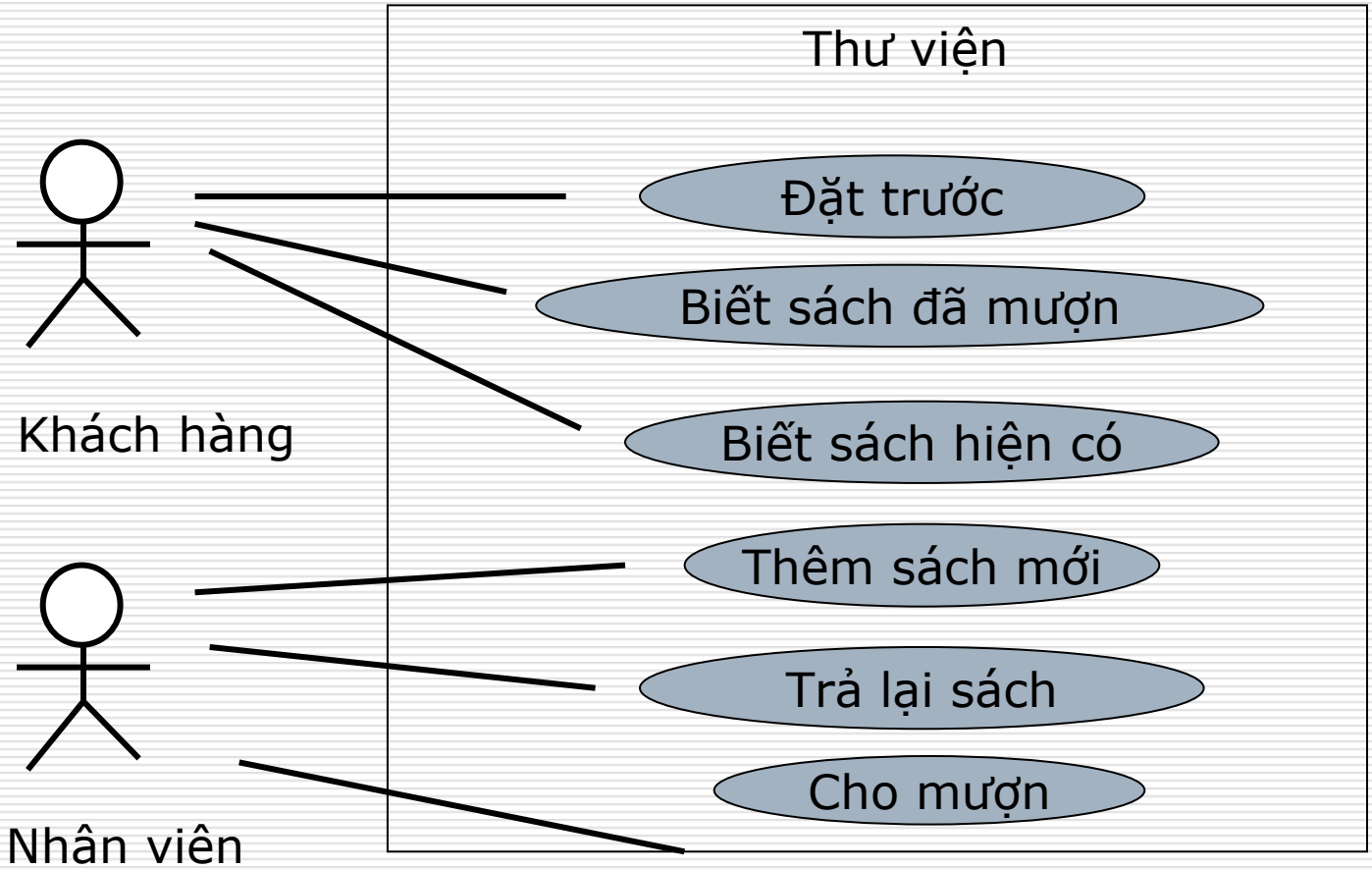
Use Case Diagram – Biểu đồ Ca sử dụng

- Cách nhìn chức năng của người sử dụng
- Xây dựng ở giai đoạn ban đầu (phân tích)
- Mục đích
  - Xác định ngữ cảnh của hệ thống
  - Nắm bắt các yêu cầu của hệ thống
  - Kiểm chứng kiến trúc hệ thống
  - Hướng dẫn phát triển và sinh test cases
- Được tạo bởi người phân tích và chuyên gia miền ứng dụng

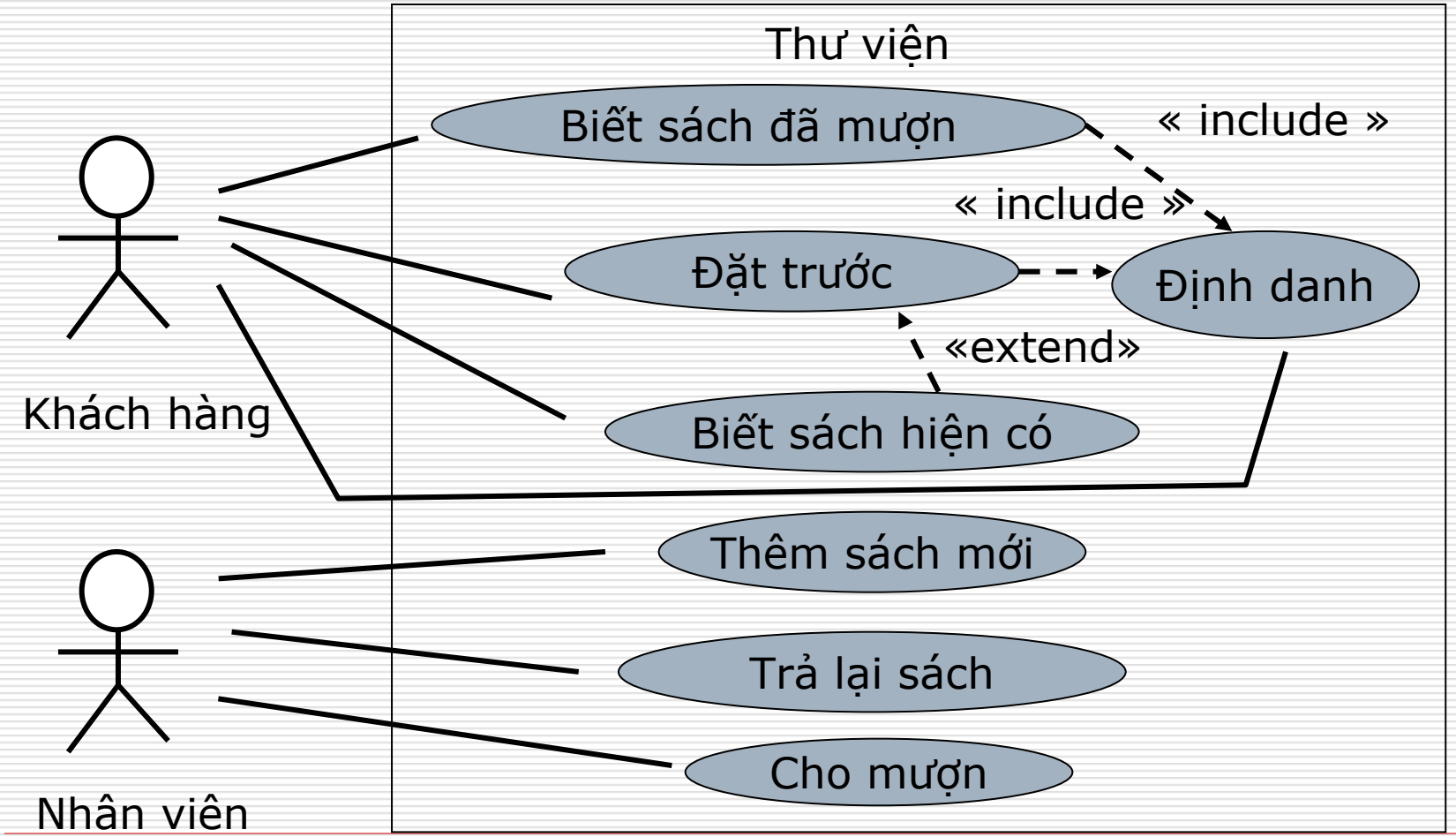
Xác định ca sử dụng

- Xác định các tác nhân sử dụng
  - Người hoặc hệ thống khác
- Với mỗi tác nhân, tìm kiếm các ca sử dụng với hệ thống
  - Cái gì làm thay đổi trạng thái của hệ thống hoặc chờ đợi phản ứng từ hệ thống
- Tổ chức lại các ca sử dụng với các quan hệ sử dụng, mở rộng…

Biểu đồ ca sử dụng



Biểu đồ ca sử dụng



Kịch bản của một ca sử dụng

Đặt trước sách

- Khách hàng đứng trước máy vi tính
  1. Hệ thống hiển thị một thông điệp chào mừng
  2. Khách hàng chọn lựa thao tác đặt trước
  3. Hệ thống yêu cầu đăng nhập
  4. Khách hàng đưa định danh
  5. Hệ thống yêu cầu chọn sách
  6. Khách hàng chọn sách muốn mượn
  7. Hệ thống chuyển trạng thái sách thành đặt trước

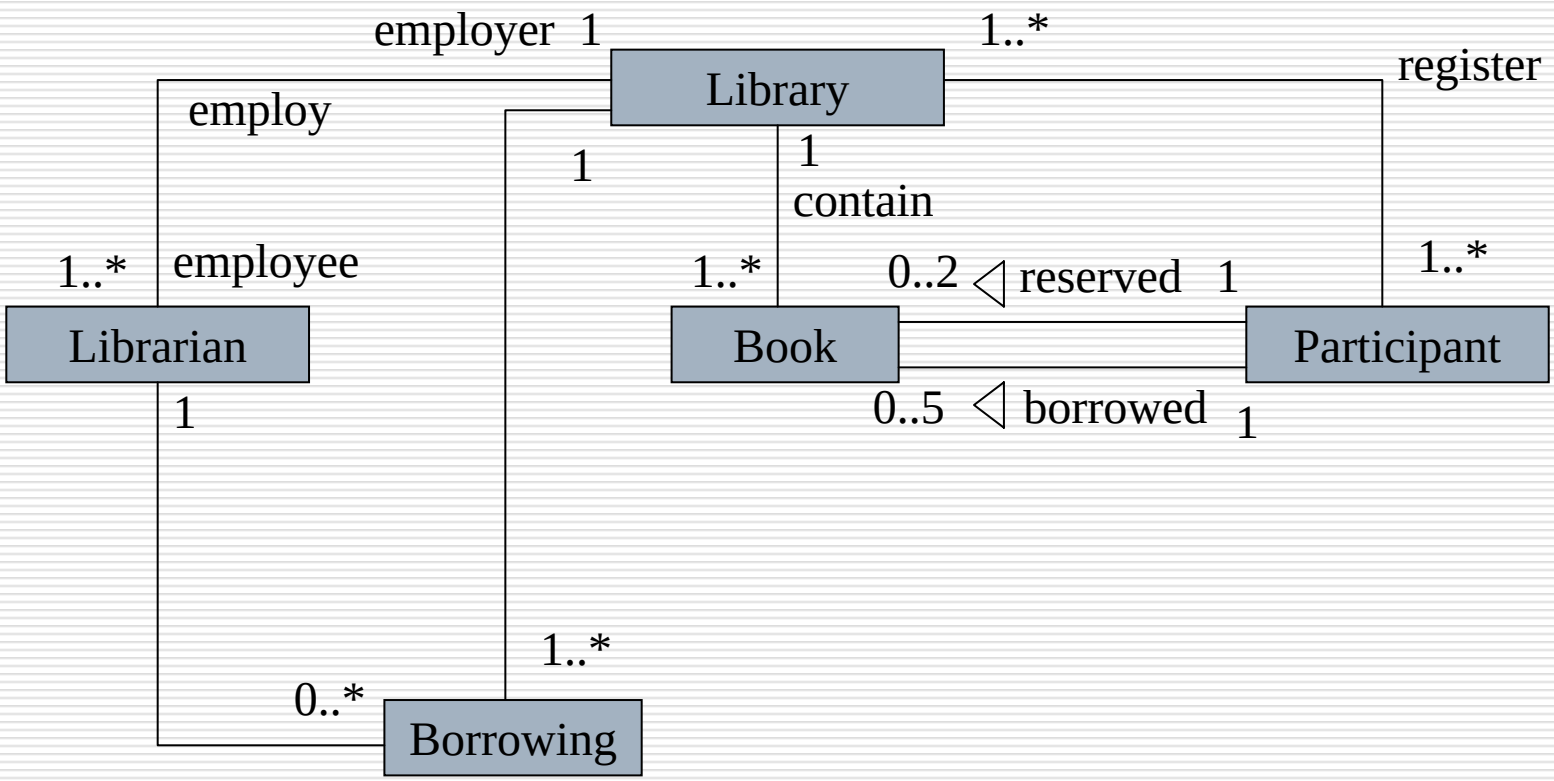
Class diagram – Biểu đồ lớp

- Là mô hình thiết kế đối tượng
  - Hỗ trợ cài đặt thực
- Các bước
  - Tìm kiếm các lớp
  - Xác định liên kết giữa các lớp
  - Xác định các thuộc tính
  - Tổ chức và đơn giản hóa bằng kế thừa
  - Xóa các liên kết thừa
  - Thẩm định: đã thỏa mãn đầy đủ các yêu cầu chưa?
  - Lặp lại và làm mịn mô hình
  - Nhóm các lớp thành các modules (gói)

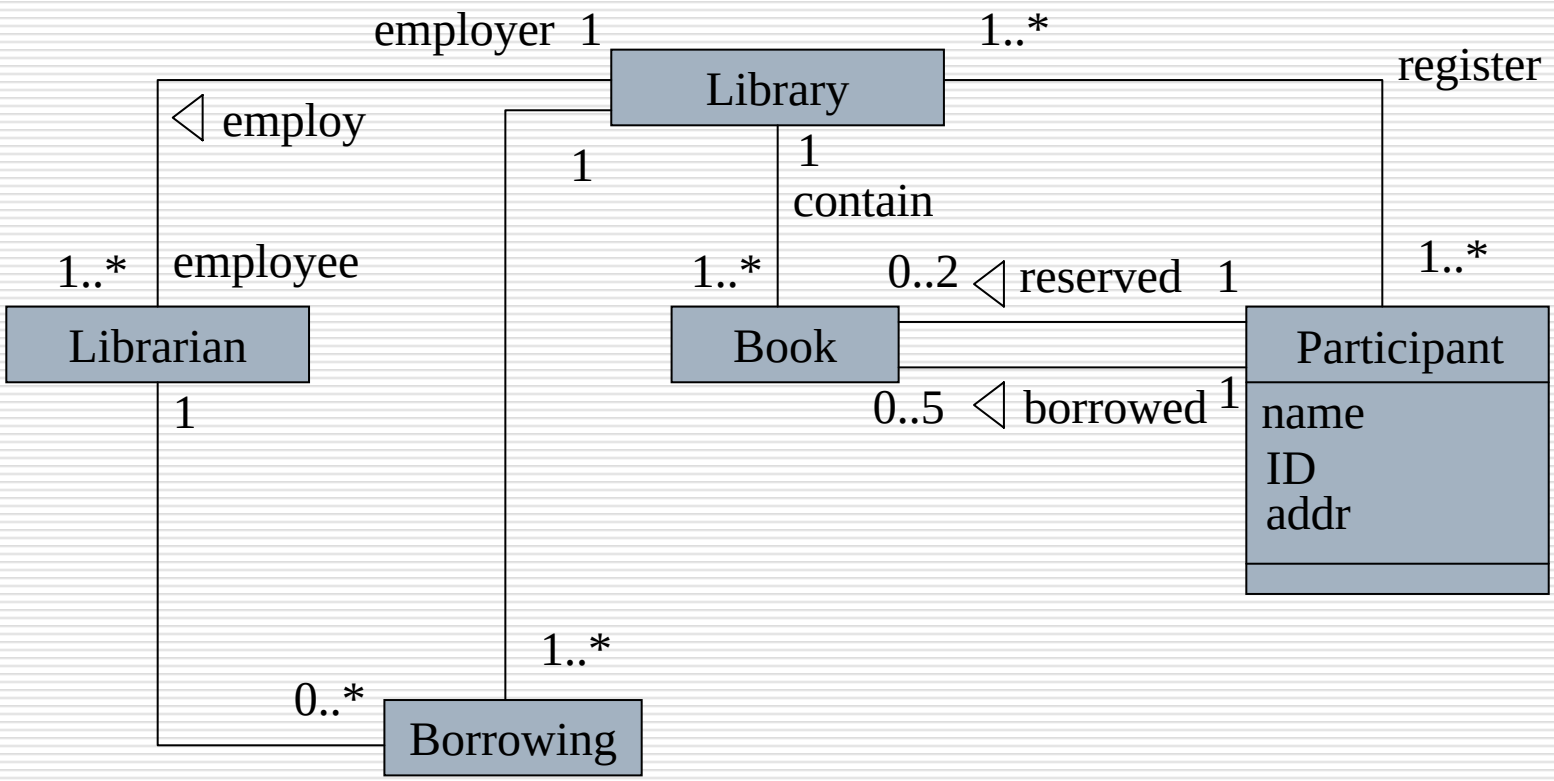
Class diagram: Xác định các lớp

- Người quản lý thư viện mong muốn tự động hóa việc mượn sách
- Cần một phần mềm cho phép người sử dụng biết sách hiện có, có thể đặt mượn 2 quyển sách, những người tham gia mượn sách có thể biết sách nào đã mượn hoặc đã đặt
- Những người tham gia mượn sách sở hữu một password để truy nhập
- Việc mượn sách được thực hiện bởi các thủ thư, sau khi xác định người mượn sách, họ biết được người này có được phép mượn hay không? (tối đa 5 quyển), người này được ưu tiên? (đã đặt trước)

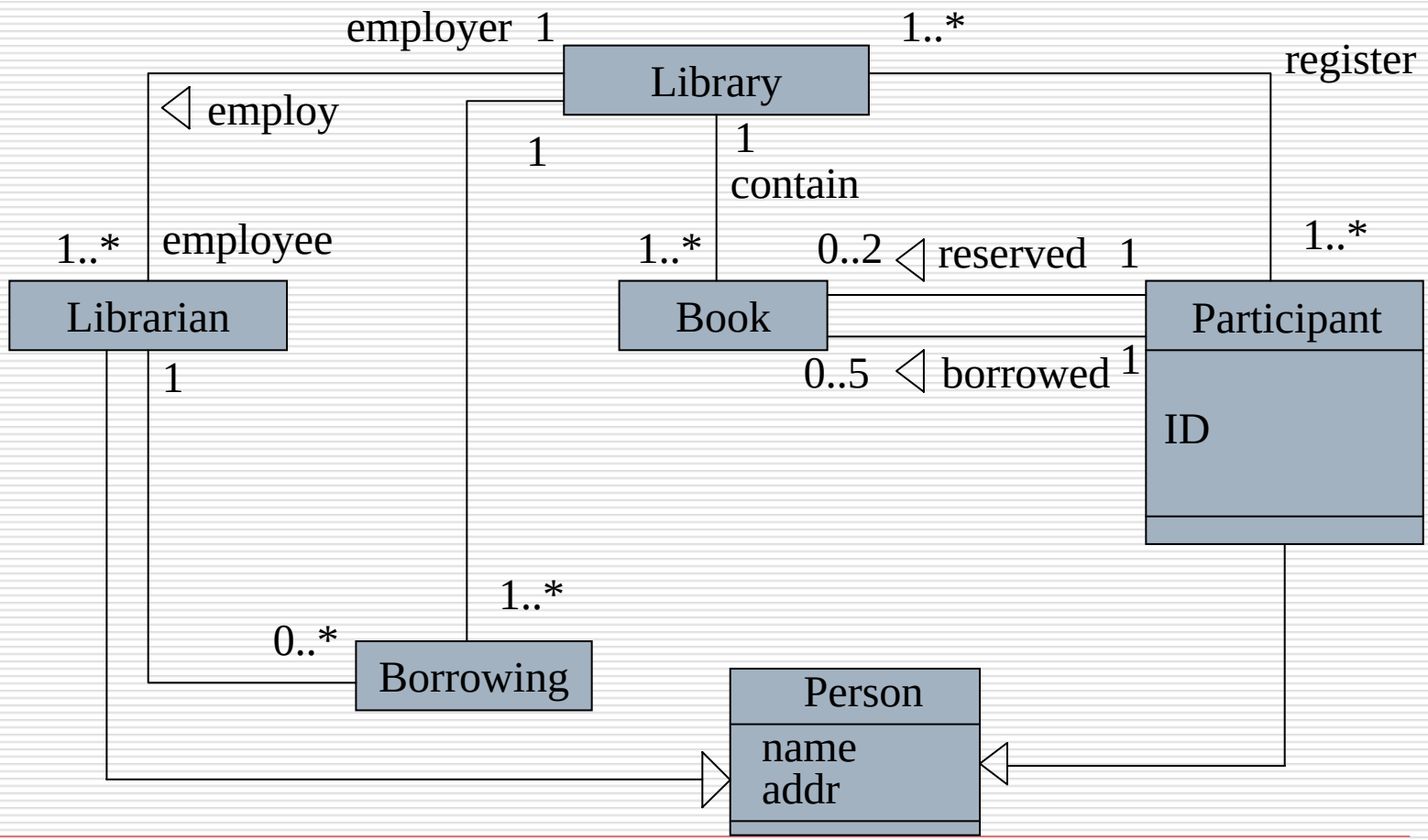
Xác định các liên kết



Xác định các thuộc tính



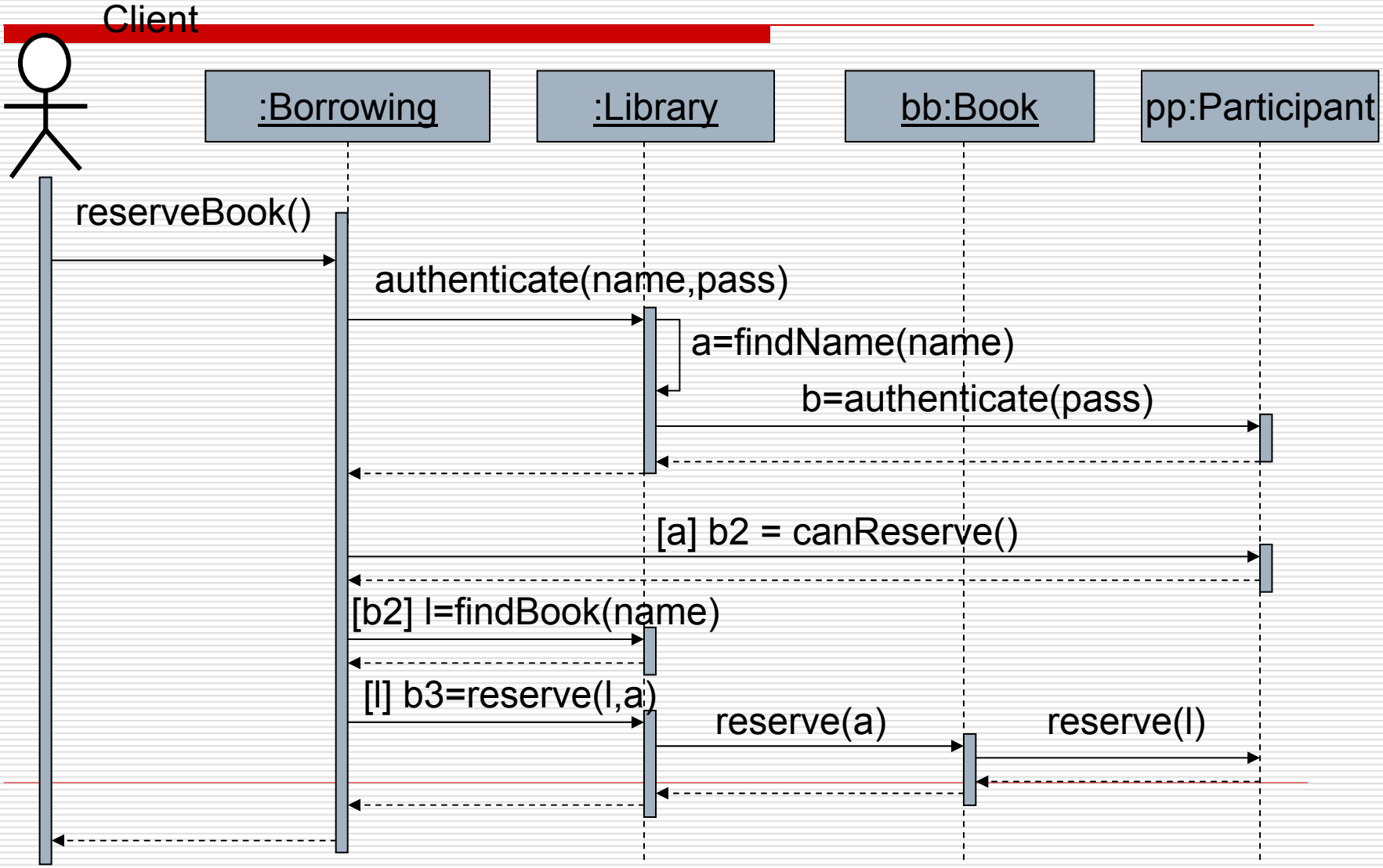
Tổng quát hóa bằng thừa kế



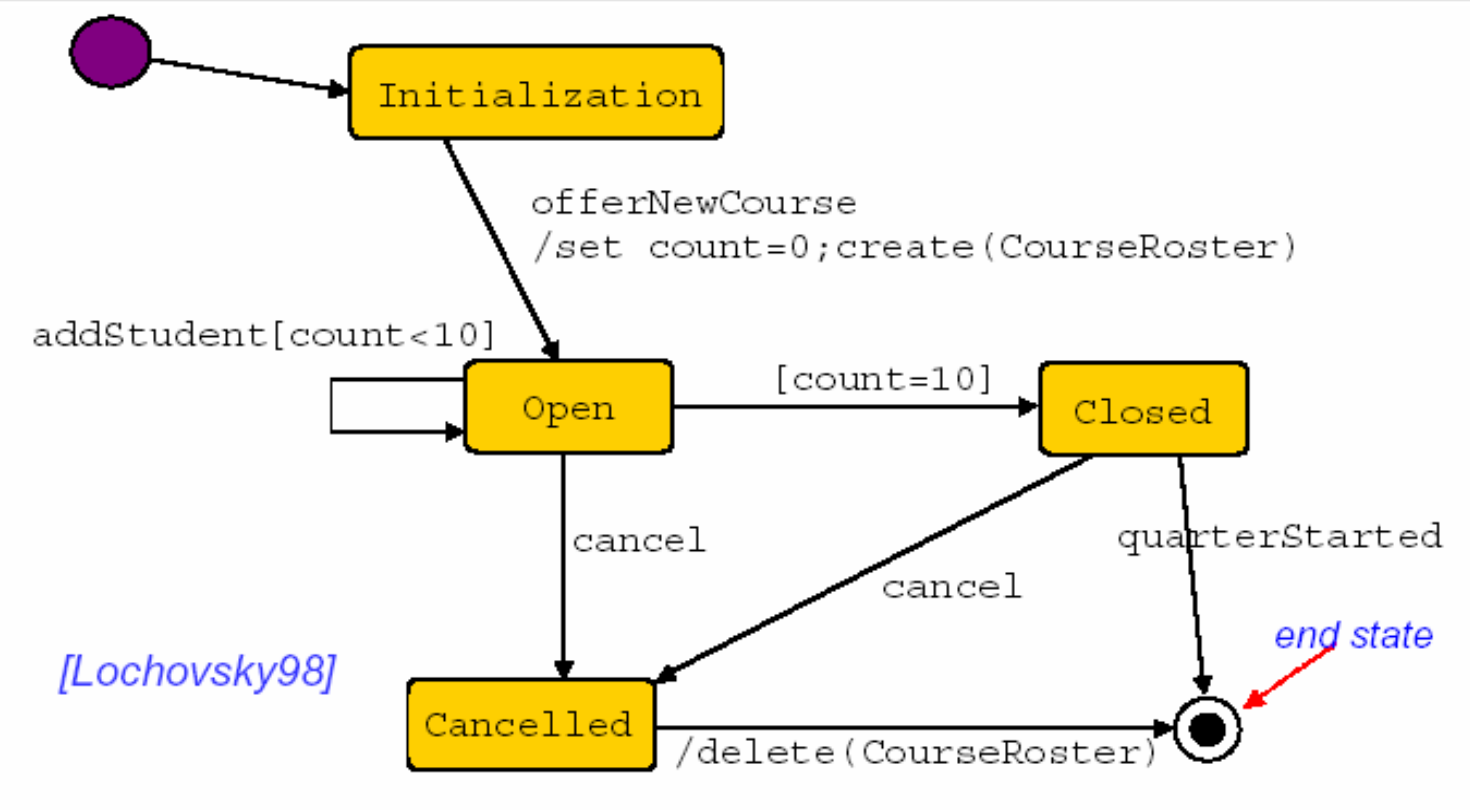
Biểu đồ tuần tự

- Thứ tự tương tác giữa các đối tượng
- Biểu diễn theo thứ tự thời gian
  - Đọc biểu đồ từ đỉnh xuống đáy
  - Mỗi đối tượng có vòng đời (Lifeline)
    - Bắt đầu khi hình thành đối tượng, kết thúc khi phá hủy đối tượng
    - Thông điệp được vẽ giữa hai đối tượng – thể hiện đối tượng gọi phương thức của đối tượng khác
    - Thông điệp phản thân: đối tượng yêu cầu thực hiện phương thức của chính nó

Sequence Diagram: Đặt trước sách

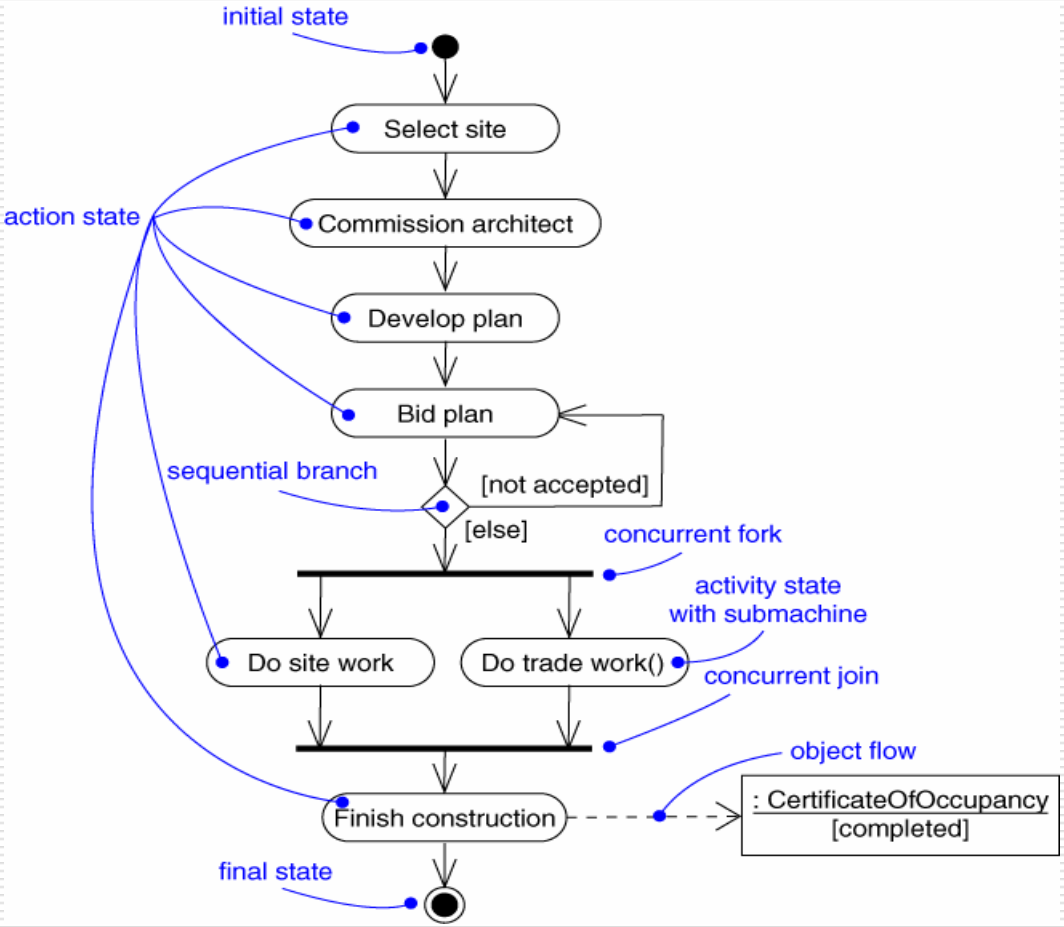


State Diagram – Biểu đồ trạng thái



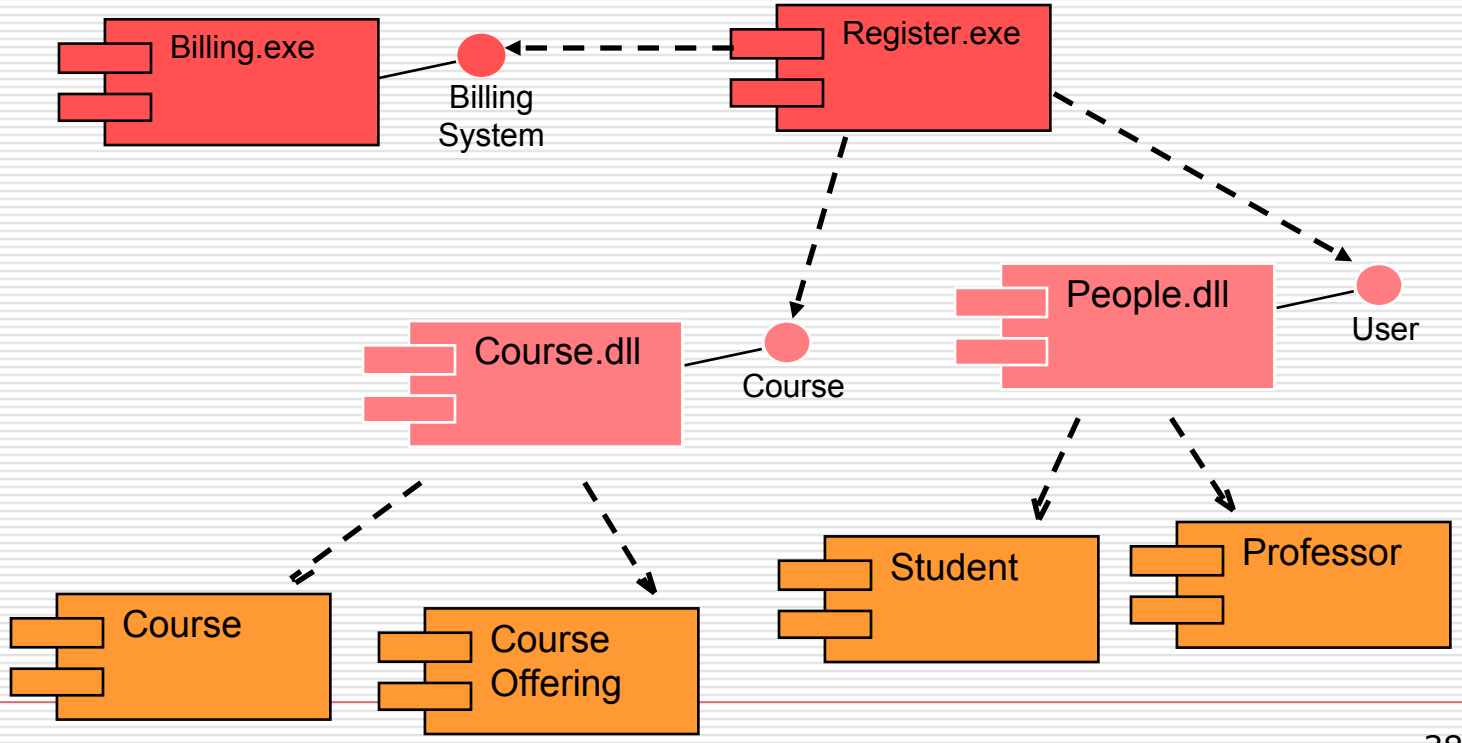
Activity Diagram – Biểu đồ hành động

- Biểu diễn các luồng công việc
- Hướng hoạt động
- Kịch bản cho ca sử dụng



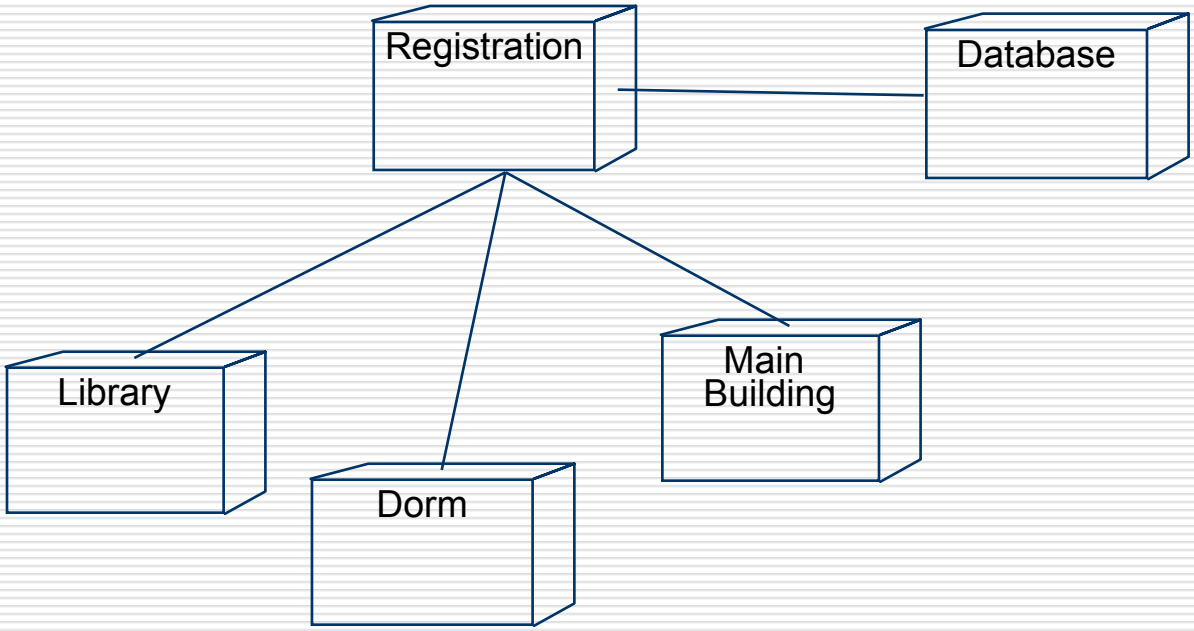
Component Diagram

- Biểu đồ thành phần (Component diagrams) biểu diễn sự tổ chức và phụ thuộc giữa các thành phần phần mềm

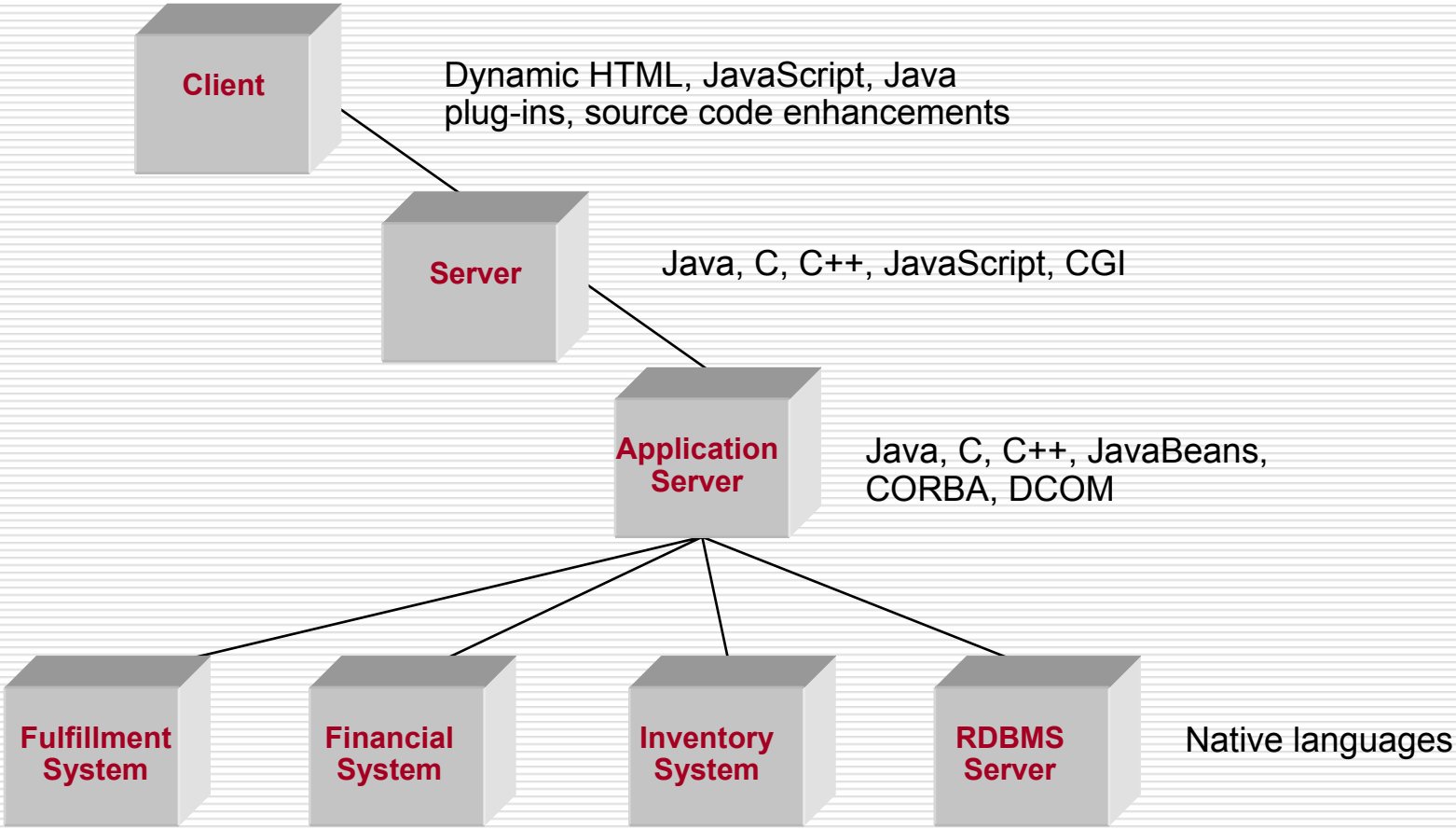


Deployment Diagram

- Biểu đồ triển khai (deployment diagram) biểu diễn cấu hình của các phần tử của hệ thống và mối liên quan



Deployment Diagram



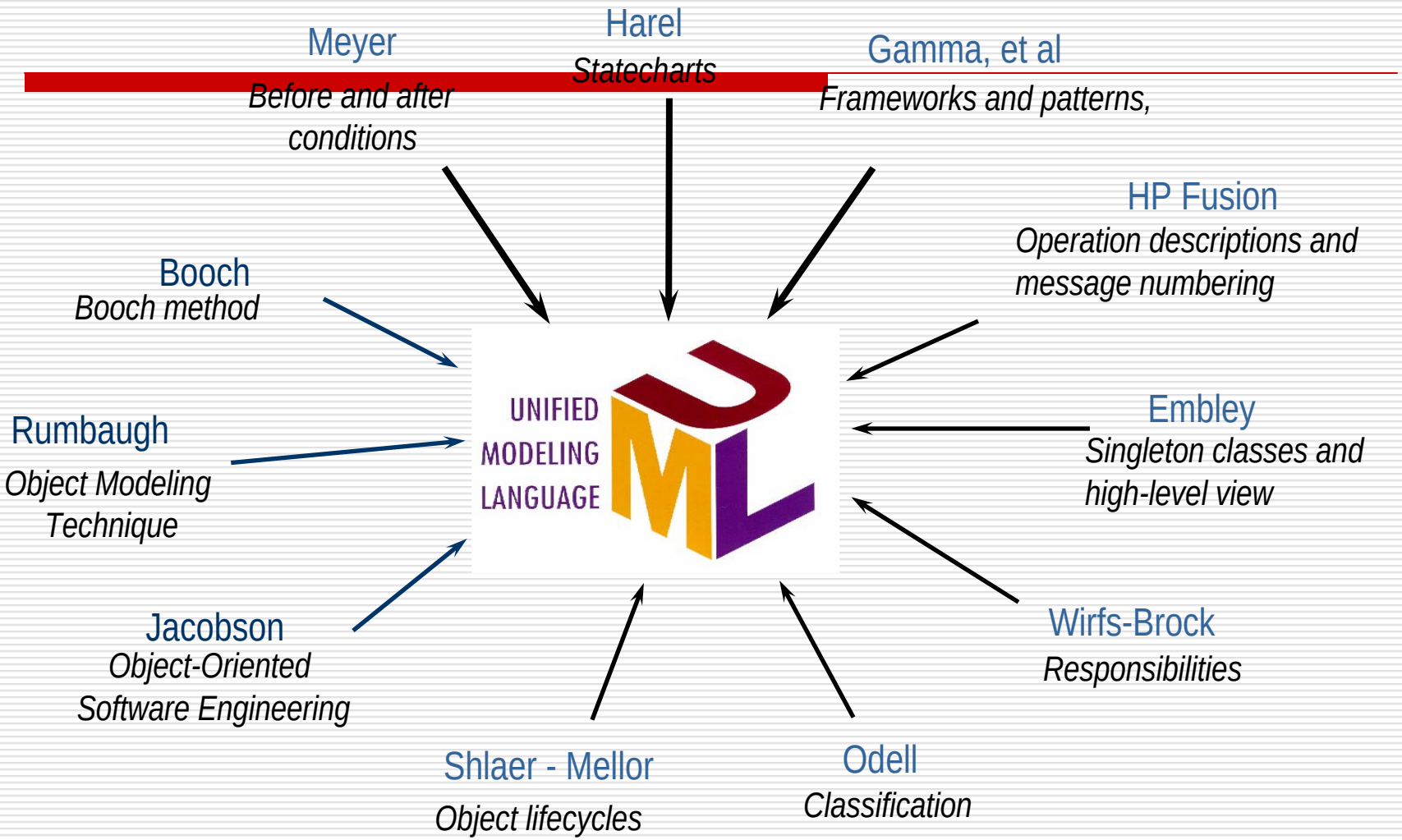
Bài tập

- Tự thực hành Star UML
- Vẽ biểu đồ use case cho bài toán đã được giao phân tích
- Vẽ biểu đồ class cho bài toán đang được giao thiết kế

UML Partners

- Rational Software Corporation
- Hewlett-Packard
- I-Logix
- IBM
- ICON Computing
- Intellicorp
- MCI Systemhouse
- Microsoft
- ObjecTime
- Oracle
- Platinum Technology
- Taskon
- Texas Instruments/Sterling Software
- Unisys

Đóng góp cho ngôn ngữ UML



3 phần tử cơ bản của UML

1. Các khối cơ bản để xây dựng
   - Các thực thể sử dụng
   - Các khái niệm liên quan
   - Các biểu đồ
2. Các luật để sử dụng các khối cơ bản
   - Các luật ngữ nghĩa
   - Các luật trình bày
3. Các cơ chế chung
   - Đặc tả
   - Trình bày
   - Mở rộng mô hình

Object Diagram

- Biểu diễn thực thể và liên kết
- Được xây dựng ở giai đoạn phân tích và thiết kế
- Mục đích
  - Minh họa cấu trúc dữ liệu/đối tượng
  - Đặc tả snapshots