

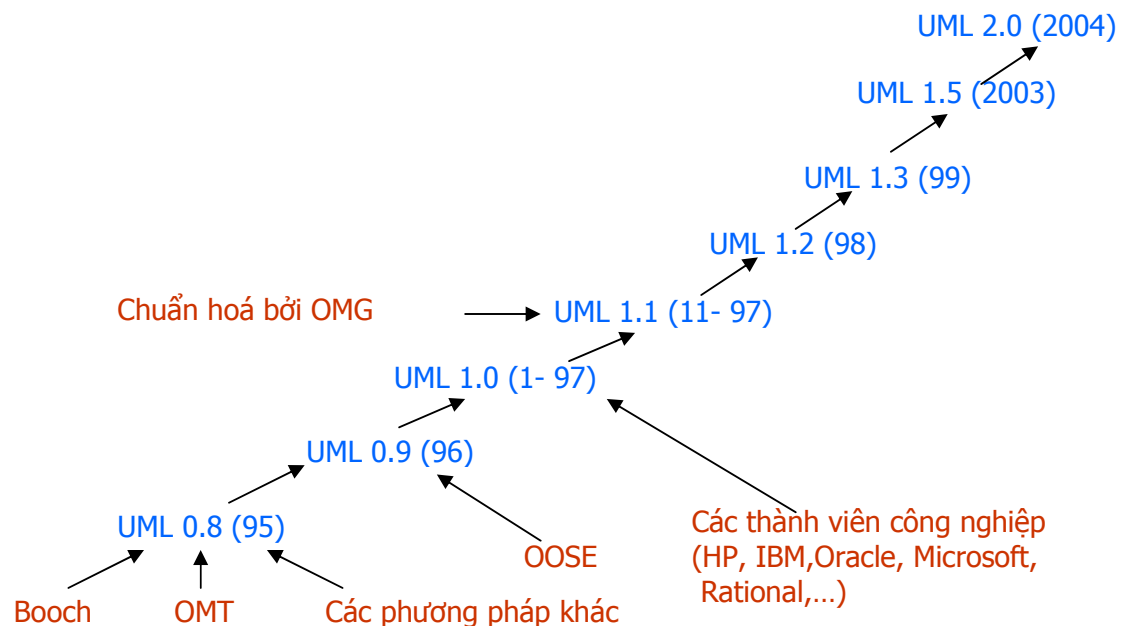
## Chương 3

# UML (UNIFIED MODELING LANGUAGE)

### Lịch sử của UML

- Số lượng các phương pháp luận hướng đối tượng gia tăng từ dưới 10 đến 50 trong khoảng những năm 1989 đến 1994, và do đó nảy sinh vấn đề là làm cho người phát triển khó tìm thấy một phương pháp luận duy nhất thoả mãn đầy đủ nhu cầu của họ.
- Vào tháng mười năm 1994, Rumbaugh đã liên kết với công ty Booch (Rational Software Corporation) để kết hợp phương pháp Booch và phương pháp OMT. Và cho ra một bản phác thảo về phương pháp có tên là Unified Process vào tháng mười năm 1995.
- Cũng trong năm 1995, Jacobson đã nỗ lực tích hợp phương pháp này với OOSE. Và những tài liệu đầu tiên về UML đã được trình làng vào trong năm 1996.
- Phiên bản 1.0 của UML đã được công bố vào tháng giêng 1997, bao gồm các công việc của các thành viên của UML consortium :
 

DEC	MCI Systemhouse
HP	Microsoft
i-Logix	Oracle
Intellicorp	Rational Software
IBM	TI
ICON	Computing Unisys
- Bản thảo về UML phiên bản 1.5 đã được tạo vào tháng ba năm 2003.
- Phiên bản UML 2.0 sẽ được tạo vào 2004.



1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
Rational		UML consortium	OMG						
U M L 0.8	U M L 0.9	U M L 1.0	U M L 1.1	U M L 1.2	U M L 1.3	U M L 1.4		U M L 1.5	U M L 2.0 ??

### UML ?

- UML được tạo ra nhằm chuẩn hoá ngôn ngữ mô hình hoá, UML không phải là một chuẩn về tiến trình và do đó, UML phải được sử dụng kết hợp với một tiến trình phương pháp luận.
- UML là một ngôn ngữ dùng để đặc tả, trực quan hoá, và tư liệu hoá phần mềm hướng đối tượng. Nó không mô tả một tiến trình hay một phương pháp mà trong đó chúng ta dùng nó để mô hình hoá. Ví dụ : Công ty Rational Software đề xuất một quy trình RUP (Rational Unified Process) được xem như là một phương pháp luận phát triển hệ thống và có ngôn ngữ mô hình hoá là UML.
- UML phủ tất cả các mức mô hình hoá khác nhau trong qui trình phát triển bao gồm chín loại sơ đồ, trong đó, năm sơ đồ dùng biểu diễn khía cạnh tĩnh và bốn sơ đồ biểu diễn khía cạnh động của hệ thống.

### Các đặc trưng của một tiến trình sử dụng UML

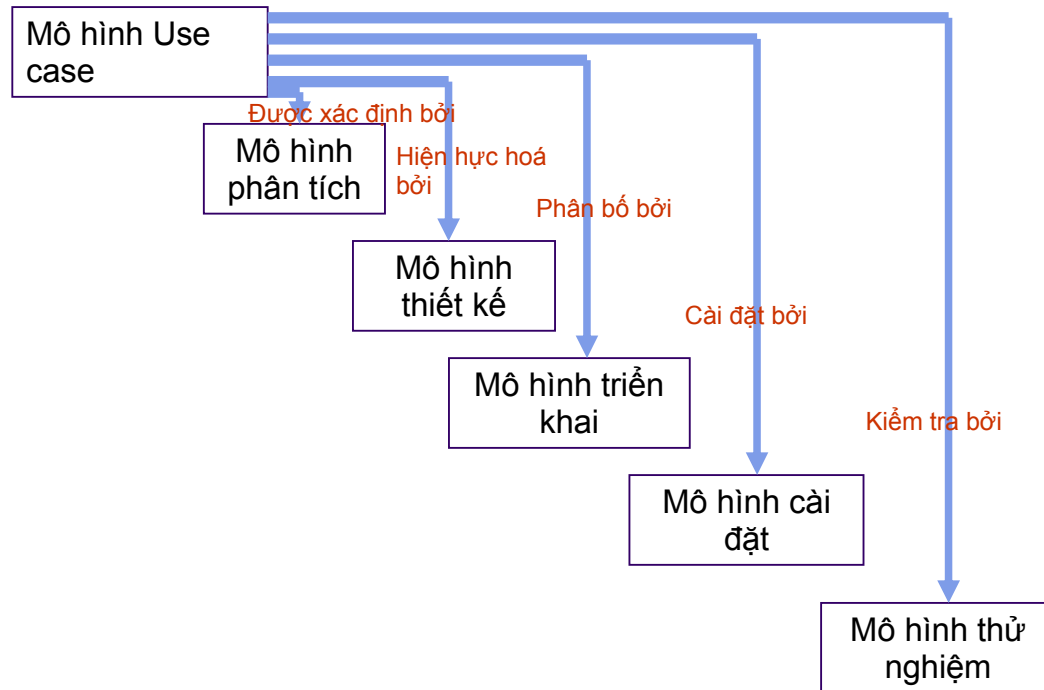
Thuở ban đầu, qui trình tuần tự được xem là phương pháp hợp lý nhất để phát triển hệ thống. Tuy nhiên, trải qua vài thập niên, đã cho thấy các dự án sử dụng qui trình tuần tự thường ít thành công, bởi những nguyên do sau đây:

- Sự giả định ban đầu có sai sót
- Thất bại trong việc kết hợp các nhân tố con người
- Các hệ thống ngày càng lớn và thường hay thay đổi
- Chúng ta vẫn còn đang trong giai đoạn thăm dò của công nghệ phần mềm, và không có nhiều kinh nghiệm. Đây là lý do chính.

Một phương pháp luận sử dụng UML phải kết hợp với một qui trình lặp và điều này sẽ giảm đi các hạn chế của qui trình tuần tự. Tính chất lặp gồm các đặc trưng cơ bản sau :

- Tính lặp (iterative)
  - o Thay vì nỗ lực xác định tất cả các chi tiết của mô hình trong một thời điểm, chúng ta chỉ xác định các chi tiết đã « đáp ứng » cho thời điểm đó để thực hiện, và
  - o Lặp lại một (hoặc nhiều) vòng lặp khác bổ sung thêm các chi tiết
- Gia tăng (incremental)
  - o Hệ thống tiến hoá thông qua một tập các sự gia tăng
  - o Mỗi sự gia tăng sẽ bù đắp thêm vào hệ thống các tính năng khác

- Tập trung người dùng (user – concentrated)
  - o Phân tích viên xác định các tính năng của hệ thống thông qua các use case
  - o Người dùng xác nhận các use case này
  - o Thiết kế viên và người phát triển hiện thực hoá các use case
  - o Người thử nghiệm kiểm tra hệ thống về việc thoả mãn các use case được đặt ra.



- Hướng kiến trúc (well-defined structure)
  - o Hệ thống được phân chia thành các hệ thống con
  - o Mức luận lý và vật lý phải được xác lập một cách tách biệt trong hệ thống

### Các sơ đồ trong UML

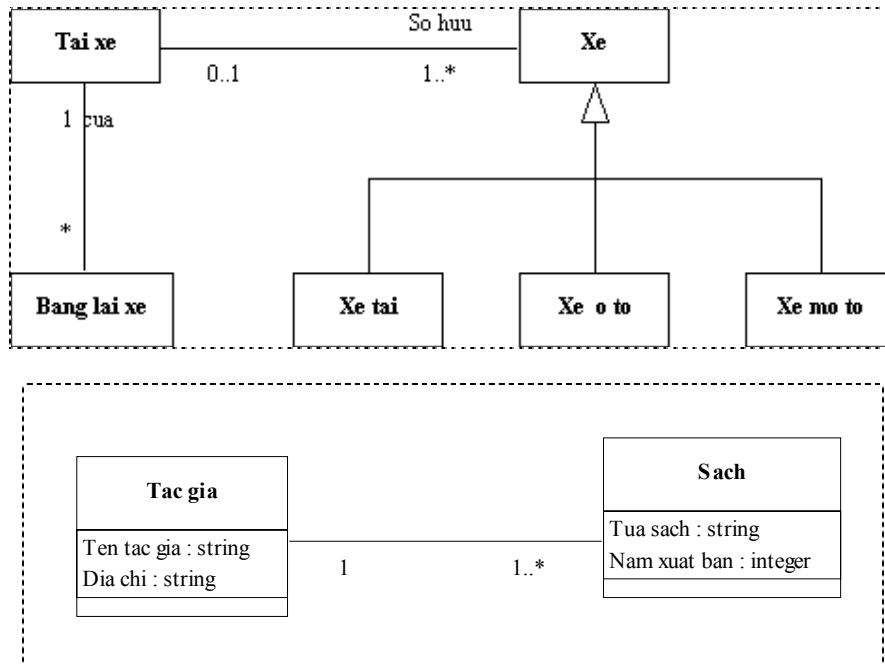
- Các sơ đồ mô tả khía cạnh tĩnh
  - o Sơ đồ đối tượng (object diagram)
  - o Sơ đồ lớp (class diagram)
  - o Sơ đồ use case (use case diagram)
  - o Sơ đồ thành phần (component diagram)
  - o Sơ đồ triển khai (deployment diagram)
- Các sơ đồ mô tả khía cạnh động
  - o Các sơ đồ tương tác (interaction diagram)
    - Sơ đồ tuần tự (sequence diagram)
    - Sơ đồ hợp tác (collaboration diagram)
  - o Sơ đồ hoạt động (activity diagram)
  - o Sơ đồ chuyển dịch trạng thái (state transition diagram)

**Sơ đồ lớp và đối tượng:** được sử dụng để mô hình hoá cấu trúc tĩnh của hệ thống trong quá trình phát triển. Mỗi sơ đồ chứa đựng các lớp và các mối quan hệ giữa chúng (quan hệ kế thừa (heritage), quan hệ kết hợp (association), quan hệ tập hợp (aggregation), quan hệ thành phần (composition)). Chúng ta cũng có thể mô tả các hoạt động của lớp (operation). Sơ đồ đối tượng là một thể hiện của sơ đồ lớp. Nó mô tả trạng thái chi tiết của hệ thống tại một thời điểm cụ thể và là bức tranh của hệ thống tại một thời điểm, do đó, biểu đồ đối tượng

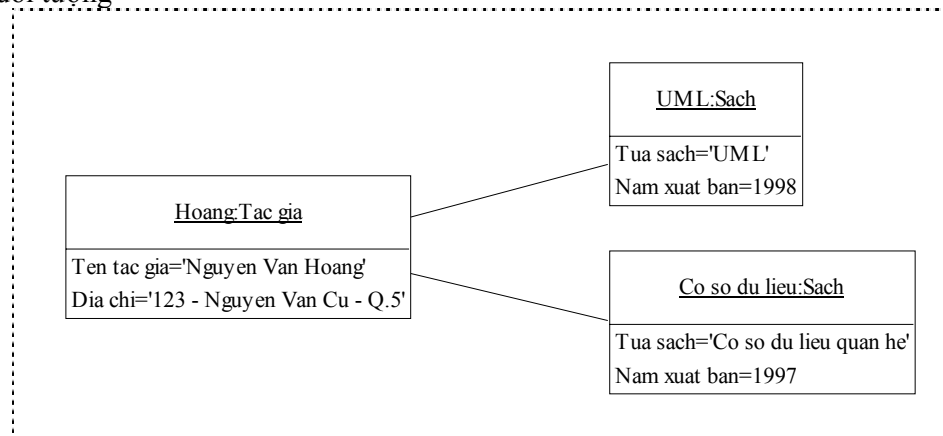
được dùng để minh hoạ một trường hợp thực tế của sơ đồ lớp. Sơ đồ đối tượng có cùng ký hiệu với biểu đồ lớp. Sơ đồ đối tượng được dùng để minh hoạ một trường hợp phức tạp của bức tranh thực tế về hệ thống trong các thể hiện cụ thể.

Ví dụ :

Sơ đồ lớp

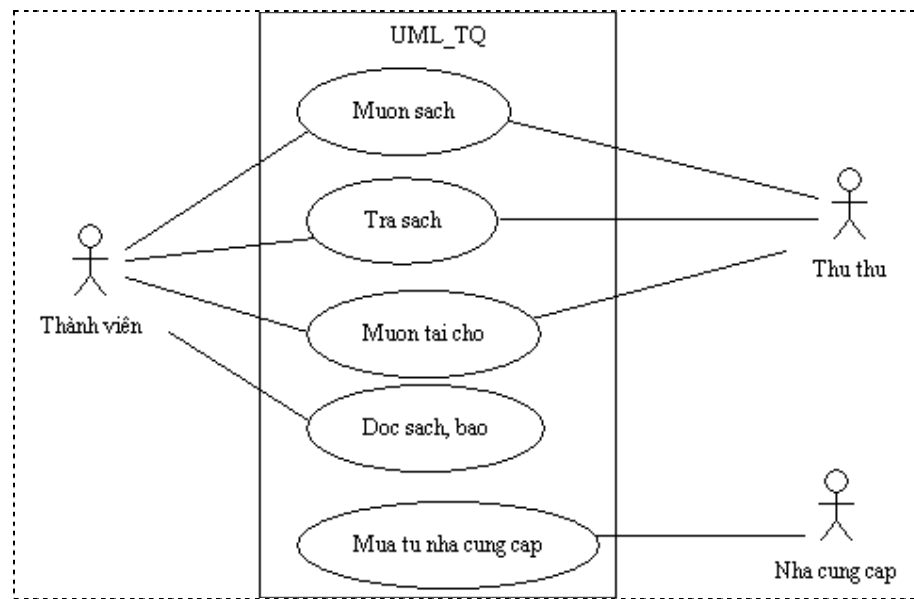


Sơ đồ đối tượng

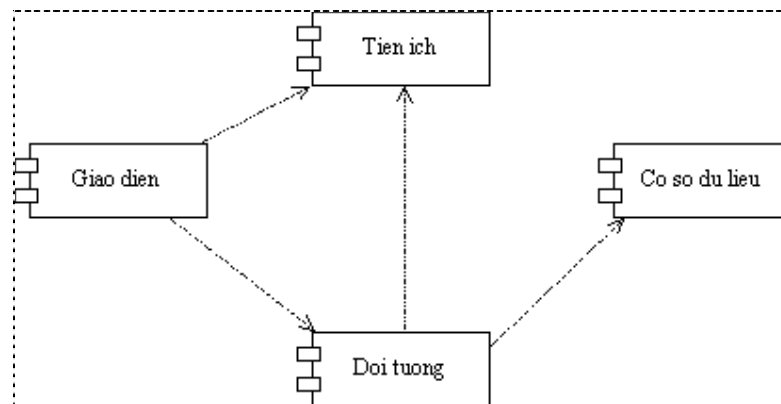


**Sơ đồ use case** : xuất phát từ các mô hình use case của phương pháp OOSE (Jacobson). Nó mô tả giao diện với một hệ thống từ quan điểm và cách nhìn của người sử dụng. Một sơ đồ use case mô tả các tình huống tiêu biểu của việc sử dụng một hệ thống. Nó biểu thị các trường hợp sử dụng (trong việc mô hình hoá các tính năng hệ thống) và các tác nhân (trong việc mô hình hoá các vai trò tham gia bởi các cá nhân tương tác với hệ thống), và mối quan hệ giữa các use case và các tác nhân.

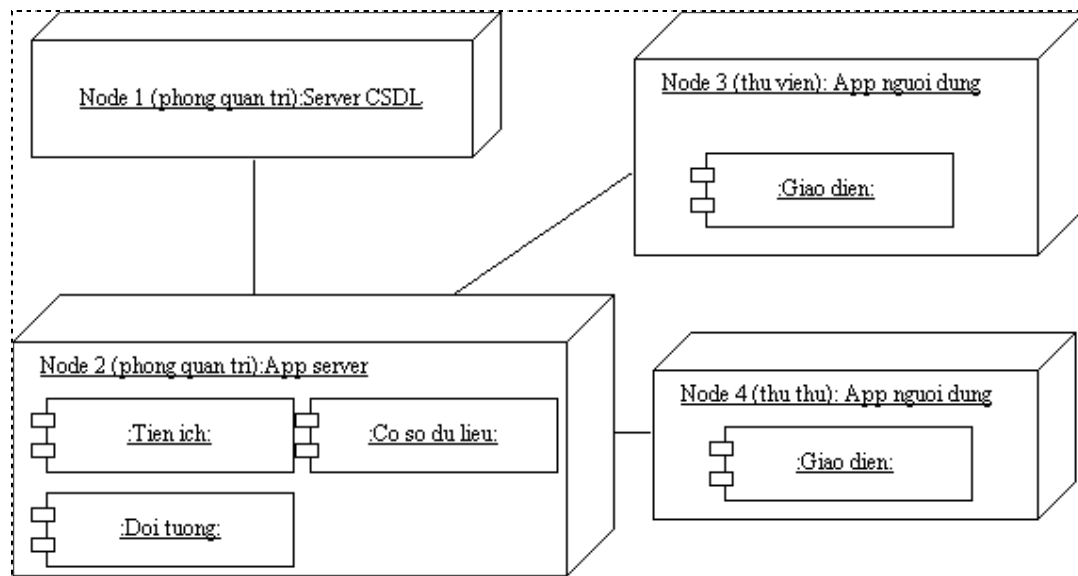
Ví dụ : sơ đồ use case một hệ thống quản lý thư viện



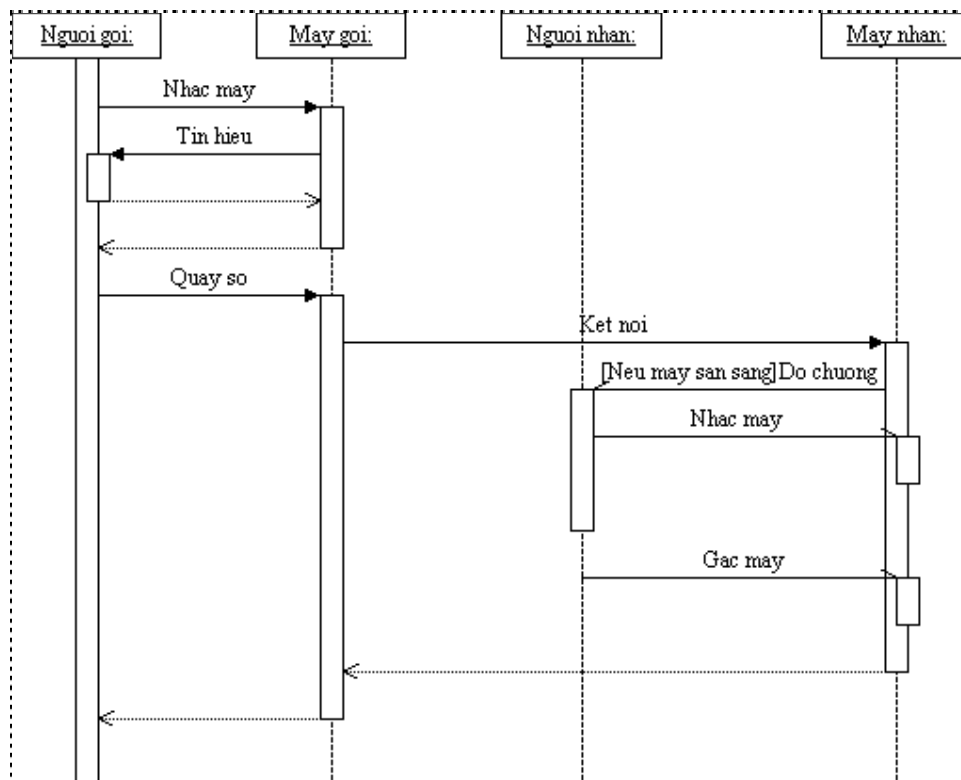
**Sơ đồ thành phần** : được sử dụng để biểu thị các nhìn tĩnh trong việc cài đặt một hệ thống. Mỗi sơ đồ bao gồm các thành phần (component) và các mối quan hệ phụ thuộc giữa chúng trong môi trường cài đặt. Một thành phần đại diện cho một yếu tố cài đặt vật lý của môi trường (mã nguồn, mã thực thi, tập tin, cơ sở dữ liệu, một thư viện hàm,...). Ví dụ : sơ đồ thành phần của một hệ thống phần mềm quản lý thư viện



**Sơ đồ triển khai** : mô tả cách bố trí vật lý các thiết bị và sự phân phối các thành phần trú ngụ tại các thiết bị này. Một sơ đồ triển khai bao gồm các nút (node) đại diện cho các tài nguyên thiết bị và các thành phần được cài đặt trong thiết bị, các liên kết trong sơ đồ dùng để mô tả sự trao đổi giữa các nút. Sơ đồ triển khai biểu thị một sự tương ứng giữa cấu trúc phần mềm của một hệ thống và kiến trúc về bố trí thiết bị của nó. Ví dụ : sơ đồ triển khai của hệ thống quản lý thư viện

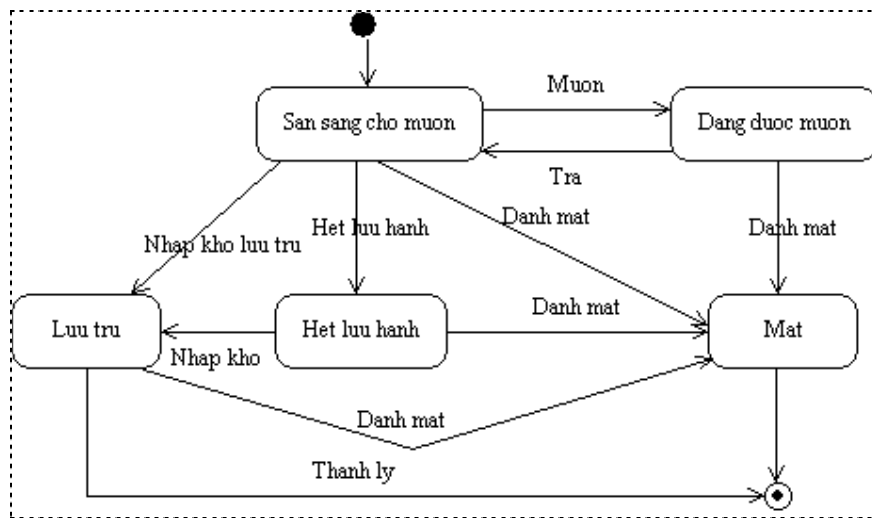


**Sơ đồ tuần tự và sơ đồ hợp tác** : trình bày các cách nhìn động về tương tác giữa các đối tượng của hệ thống trong quá trình phát triển. Sơ đồ hợp tác mô tả sự hợp tác giữa một nhóm các đối tượng trong hoạt động để đạt một mục tiêu cụ thể. Sơ đồ tuần tự thêm vào chiều thời gian nhằm thể hiện trực quan thứ tự trao đổi của các thông điệp (message). Ví dụ : sơ đồ tuần tự mô tả hoạt động của xử lý cuộc gọi của máy điện thoại



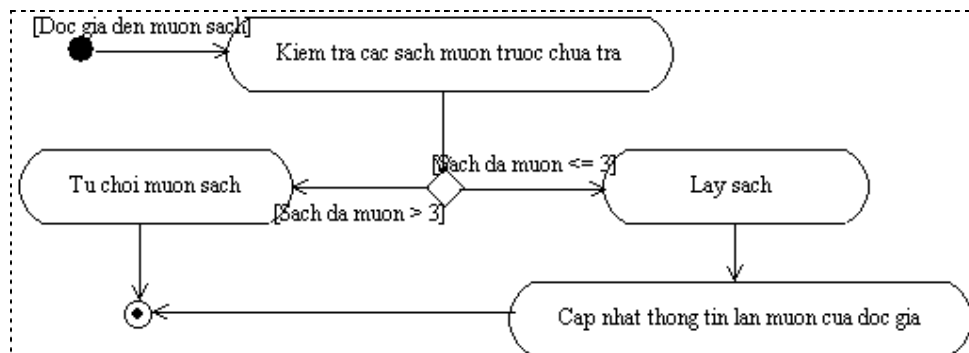
**Sơ đồ chuyển đổi trạng thái** : hình thành từ phương pháp OMT và Booch. Mỗi sơ đồ được dùng có liên quan đến một lớp để biểu thị các trạng thái khác nhau của đối tượng của lớp và các biến cố kích hoạt sự chuyển dịch giữa các trạng thái.

Ví dụ : sơ đồ trạng thái của cuốn sách trong thư viện



**Sơ đồ hoạt động** : dùng để mô hình hoá các dòng hoạt động liên kết tới các lớp như là trong trường hợp của một nhóm các lớp hợp tác cùng thực hiện trong một loại tiến trình. Mỗi lớp sẽ đảm nhiệm các hoạt động và các chuyển dịch như được mô tả trong sơ đồ chuyển dịch trạng thái. Tuy nhiên, một sơ đồ hoạt động có thể liên quan đến nhiều lớp hơn là một lớp. Mặt khác nó mô tả tiến trình tuần tự các hoạt động, sự đồng bộ hoá các dòng điều khiển song song, các điều kiện và quyết định, điểm bắt đầu và điểm kết thúc tiến trình.

Ví dụ : sơ đồ hoạt động đơn giản của hoạt động mượn sách thư viện



## Các hệ thống sử dụng UML trong việc mô hình hoá

Việc sử dụng UML trong quá trình mô hình hoá có thể áp dụng trong các loại hệ thống sau :

**Hệ thống nghiệp vụ (business system)** : mô tả tài nguyên (nhân lực, tài lực, tài sản,...), mục tiêu, luồng công việc, các qui tắc, ràng buộc trong hoạt động sản xuất kinh doanh của doanh nghiệp. Ví dụ : các công ty sản xuất, cửa hàng kinh doanh, y tế, giáo dục, ...

**Hệ thống thông tin (informaton system)** : thu thập và lưu trữ, biến đổi dữ liệu nhằm cung cấp thông tin đáp ứng nhu cầu người nhận trong các tổ chức hoạt động nghiệp vụ. Hệ thống thông tin cũng được chia thành nhiều loại tùy thuộc vào quy mô và độ phức tạp : hệ thống thông tin tác nghiệp : là hệ thống chuyên xử lý việc thu thập và truy tìm thông tin trong môi trường hoạt động nghiệp vụ. Hệ thống thông tin quản lý : xử lý tổng hợp dữ liệu thông qua các thống kê báo cáo nhằm đáp ứng thông tin cho các nhà quản lý theo dõi tình hình hoạt động. Hệ thống thông tin chuyên gia, hệ hỗ trợ ra quyết định : xử lý và tri thức hoá các dữ liệu hiện tại nhằm đáp ứng các nhu cầu nâng cao về mặt thông tin như là hỗ trợ giải đáp tự động, hỗ trợ ra quyết định, dự báo tình hình tương lai,...

**Phần mềm hệ thống (System software)** : xây dựng các công cụ phần mềm cơ sở cho các phần mềm khác sử dụng như là hệ điều hành, hệ quản trị cơ sở dữ liệu, công cụ phát triển,....

**Hệ thống nhúng (embeded system)** : là một loại hệ thống phần mềm được xây dựng gắn trên một loại thiết bị như : điện thoại di động, thiết bị điều khiển,... Các hệ thống nhúng này thường được lập trình dùng ngôn ngữ cấp thấp hoặc chuyên dụng và có bộ nhớ lưu trữ cũng như màn hình.

**Hệ thống kỹ thuật (Technical system)** : xử lý và điều khiển các thiết bị kỹ thuật như hệ thống viễn thông, hệ thống quân sự, hay các quá trình xử lý kỹ thuật công nghiệp (dây chuyền sản xuất, vận hành máy móc,...). Đây là loại thiết bị phải xử lý các giao tiếp đặc biệt , không có phần mềm chuẩn và thường là các hệ thống thời gian thực (real time).