

Phân tích thiết kế hướng đối tượng

Bài 3: Tổng quan về ngôn ngữ UML

TS. Nguyễn Hiếu Cường

Bộ môn CNPM, Khoa CNTT, Trường ĐH GTVT

Email: cuonggt@gmail.com

Xuất xứ của UML

- UML là một bộ ký pháp mô hình hóa hướng đối tượng
- Đầu những năm 1990, đã có nhiều phương pháp mô hình hóa hướng đối tượng: OOAD của Booch, OMT của Rumbaugh, OOSE/Objectory của Jacobson, OOA/OOD của Coad và Yourdon...
- Năm 1994 Booch và Rumbaugh bắt đầu hợp tác trong khuôn khổ của công ty Rational, nhằm xây dựng một phương pháp hợp nhất
- Năm 1995 Jacobson cũng tham gia hợp tác và tạo nên một nhóm gọi là “ba người bạn” (three amigos)...

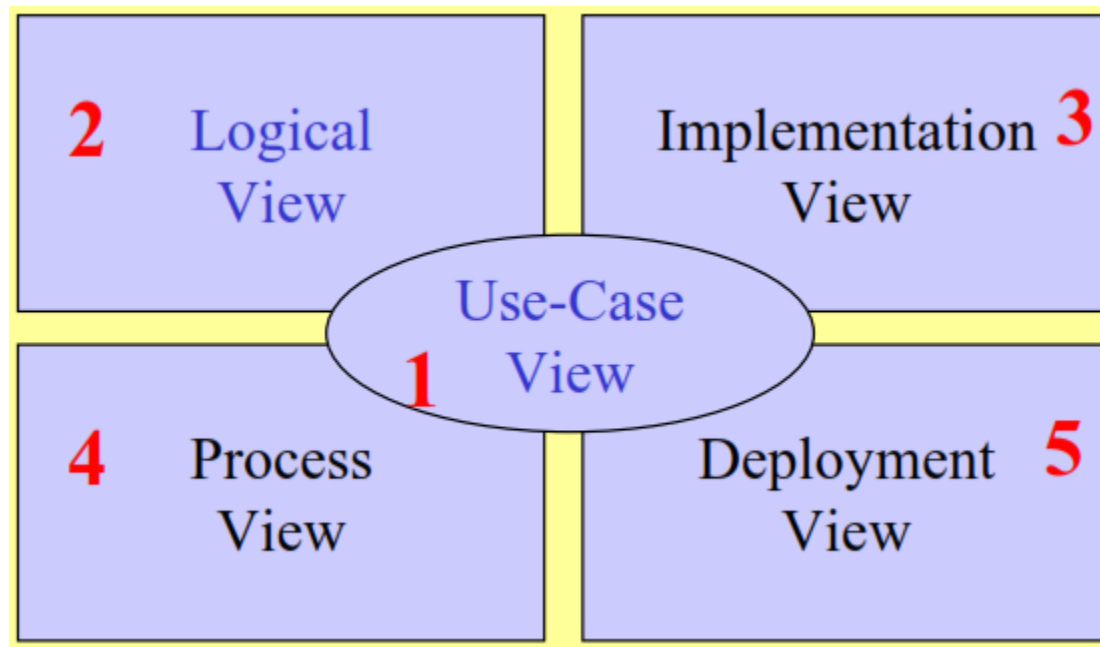
Xuất xứ của UML

- Tháng 10/1995 đưa ra phác thảo đầu tiên, tháng 6/1996 (UML 0.9), tháng 1/1997 (UML 1.1)
- Cuối năm 1997, UML 1.1 đã được OMG (Object Management Group) công nhận là chuẩn cho các ngôn ngữ mô hình hóa
 - Celebrating 25 years of UML 1.1 (<https://www.uml.org/>)
 - Tiêu chuẩn [TCVN 7981-2: 2008](#) (tương đương ISO/TS 17369 - 2: 2005)
- Tháng 6/1998 (UML 1.2), 10/1998 (UML 1.3), 5/2001 (UML 1.4)
- Năm 2004 chính thức ra đời UML 2.0
- Tháng 6/2015 (UML 2.5)...

<https://www.uml-diagrams.org/>

Các góc nhìn của UML

- UML cung cấp năm (4 + 1) góc nhìn, mỗi góc nhìn thể hiện một khía cạnh của hệ thống

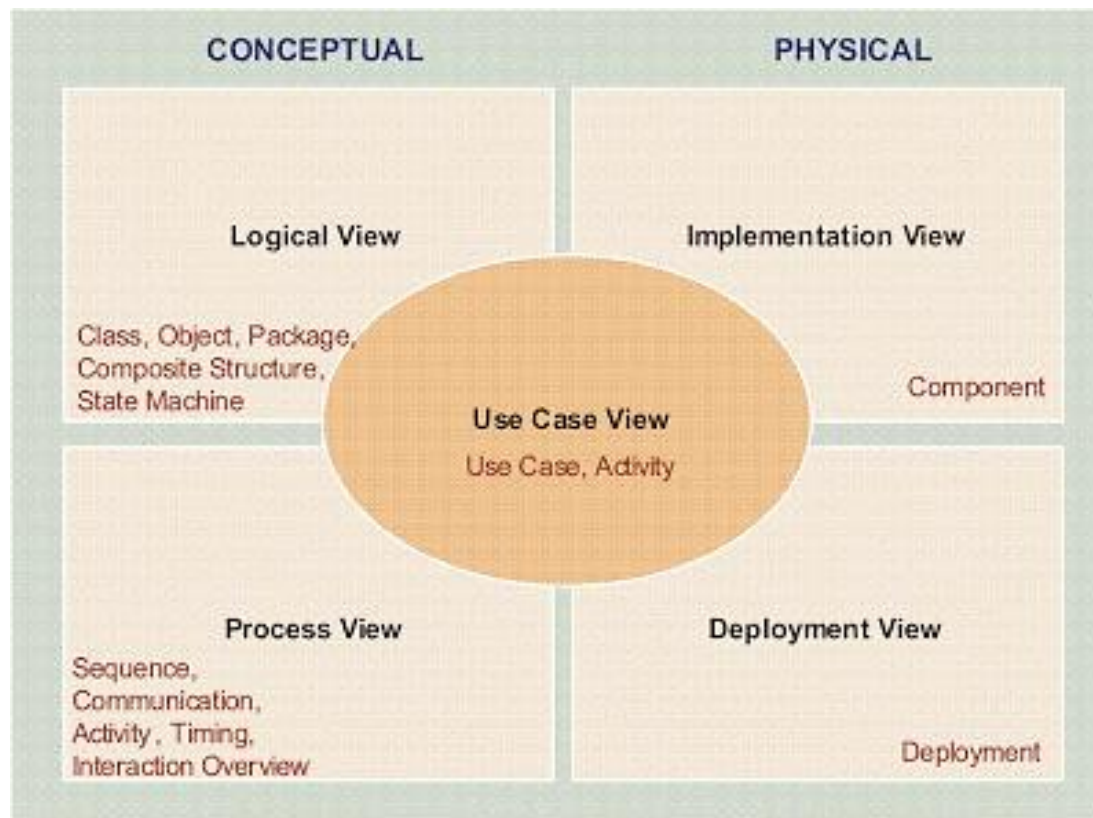


Các góc nhìn của UML

- *Use Case View*: cung cấp góc nhìn về các ca sử dụng giúp chúng ta hiểu hệ thống có gì, ai dùng và dùng nó như thế nào.
- *Logical View*: cung cấp góc nhìn về cấu trúc hệ thống, xem nó được tổ chức như thế nào, bên trong nó có gì.
- *Implementation View*: là một góc nhìn về cấu trúc giúp chúng ta hiểu cách phân bố và sử dụng các thành phần trong hệ thống ra sao.
- *Process View*: cung cấp góc nhìn động về hệ thống, xem các thành phần trong hệ thống tương tác với nhau như thế nào
- *Deployment View*: cung cấp góc nhìn về triển khai hệ thống, nó cũng ảnh hưởng lớn đến kiến trúc hệ thống

Thể hiện các góc nhìn

- Mỗi góc nhìn được thể hiện bằng một hoặc một số loại biểu đồ



Góc nhìn ca sử dụng

- Là góc nhìn từ ngoài vào hệ thống, là cách nhìn của người dùng cuối, người phân tích
- Không phản ánh tổ chức bên trong của phần mềm mà chỉ làm rõ các chức năng lớn mà hệ thống phải đáp ứng
- Cũng dùng để thẩm tra xem hệ thống có đáp ứng đúng yêu cầu
- Là góc nhìn trung tâm vì nó đặt ra nội dung tác động vào các góc nhìn khác
- Biểu đồ UML cần thiết để thể hiện trong góc nhìn này là: biểu đồ ca sử dụng

Góc nhìn thiết kế

- Còn gọi là góc nhìn logic
- Là góc nhìn vào bên trong hệ thống, cho thấy các nhiệm vụ của hệ thống được thiết kế ra sao
- Đó là cách nhìn của người thiết kế hệ thống
- Các biểu đồ UML cần thiết để thể hiện trong góc nhìn này là: biểu đồ lớp, biểu đồ đối tượng

Góc nhìn thực thi

- Còn gọi là góc nhìn thành phần
- Là góc nhìn đối với dạng phát hành của phần mềm, bao gồm các thành phần và tệp tương đối độc lập, có thể lắp ráp để tạo nên hệ thống chạy được
- Các biểu đồ UML cần thiết để thể hiện trong góc nhìn này là: biểu đồ thành phần

Góc nhìn quá trình

- Còn gọi là góc nhìn song hành
- Là góc nhìn để phản ánh các lộ trình điều khiển, các quá trình thực hiện của hệ thống
- Các biểu đồ UML cần thiết để thể hiện trong góc nhìn này là: biểu đồ trình tự, biểu đồ giao tiếp, biểu đồ máy trạng thái

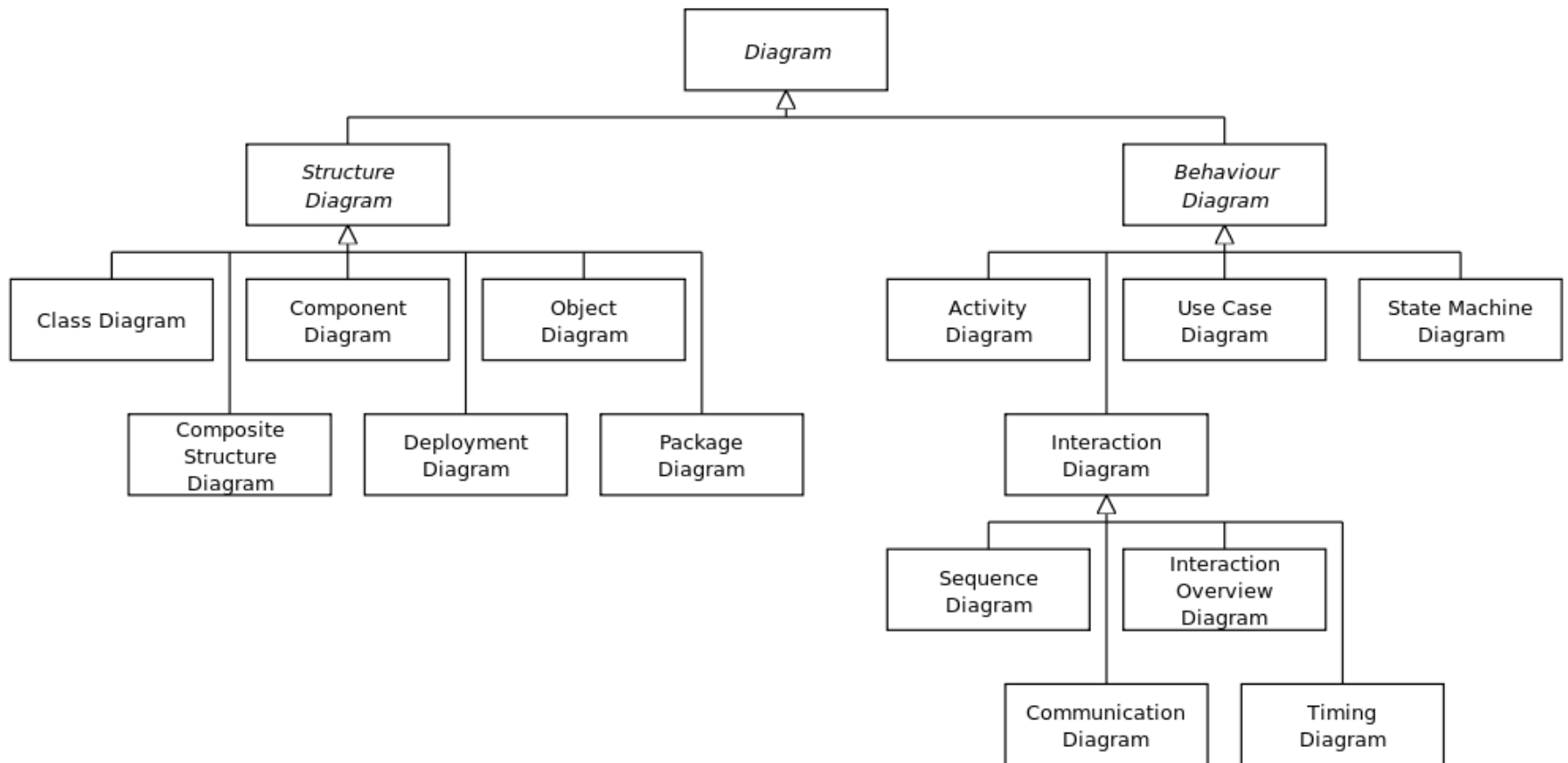
Góc nhìn bố trí

- Là góc nhìn về sự bố trí của phần cứng mà trên đó hệ thống được thực hiện
- Chỉ rõ sự phân bố, sắp đặt các thành phần của hệ thống vật lý
- Các biểu đồ UML cần thiết để thể hiện trong góc nhìn này là: biểu đồ thành phần, biểu đồ bố trí

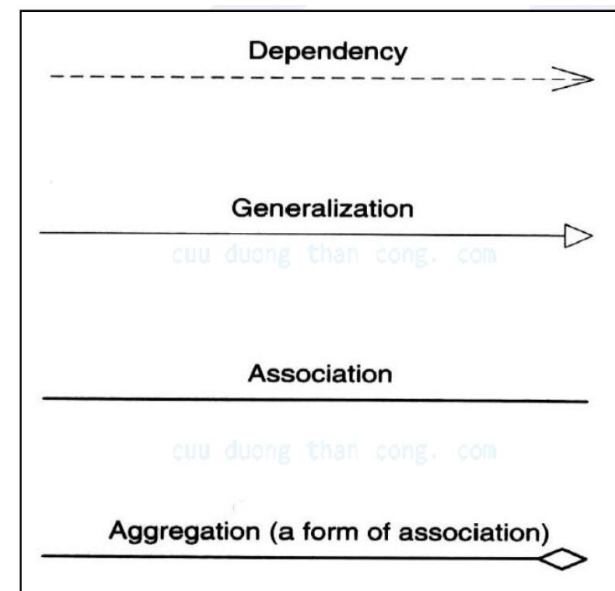
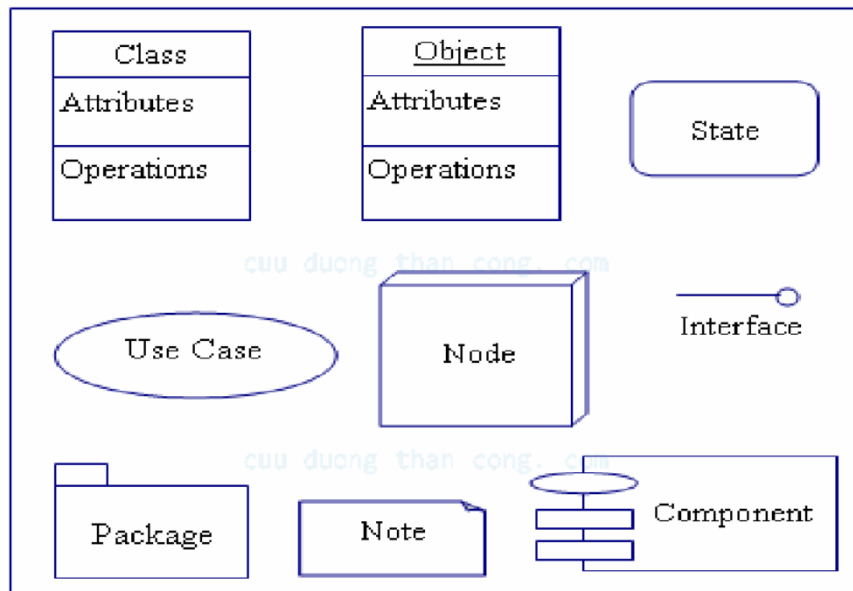
Mô hình hóa với UML

- Để mô hình hóa một hệ thống, thường cần nhiều loại mô hình để diễn tả hệ thống theo nhiều góc nhìn khác nhau
- Nếu một hệ thống nhỏ, có thể chỉ cần mô tả nó trên 2 góc nhìn, là góc nhìn ca sử dụng và góc nhìn thiết kế
- Khi mô hình hóa hệ thống, tùy theo giai đoạn và nhu cầu sử dụng mà các biểu đồ được vẽ ở mức độ trừu tượng hóa cao hay thấp (khái lược hay chi tiết)

Các biểu đồ UML

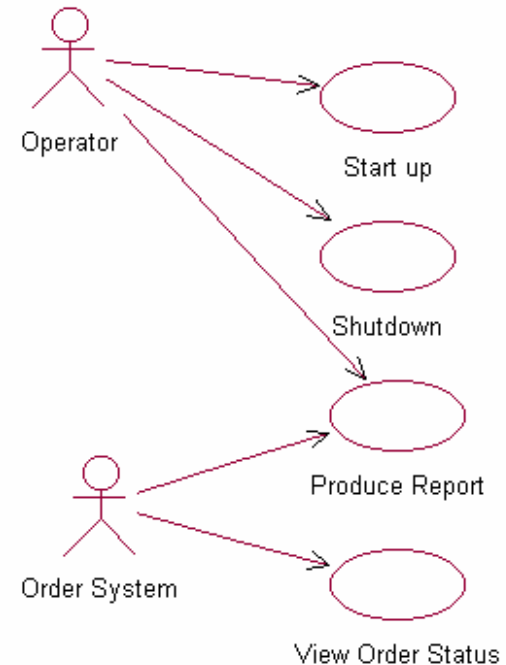


Các phần tử mô hình và quan hệ



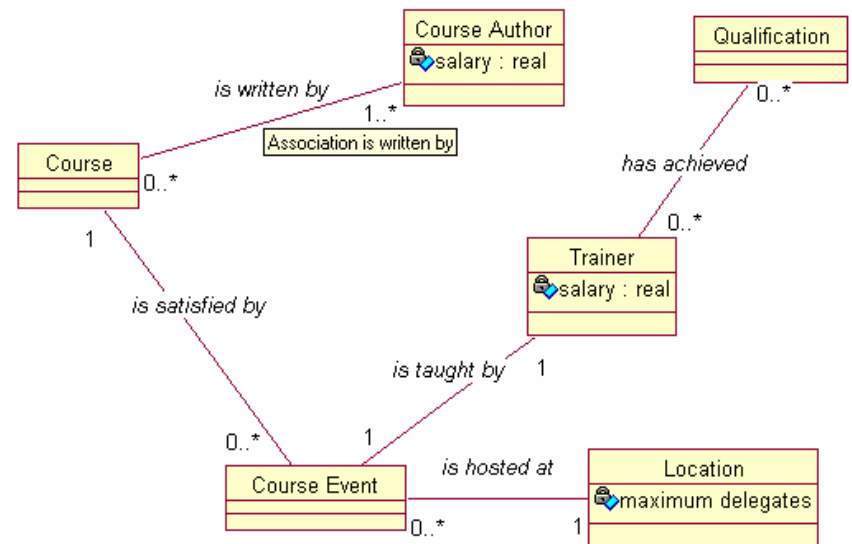
Biểu đồ ca sử dụng

- Use case diagram
- Mô tả hành vi của hệ thống từ góc nhìn của người dùng
- Giúp hiểu các yêu cầu cần làm
- Giúp người phát triển (analysts, developers, coders, tests) và khách hàng trao đổi với nhau



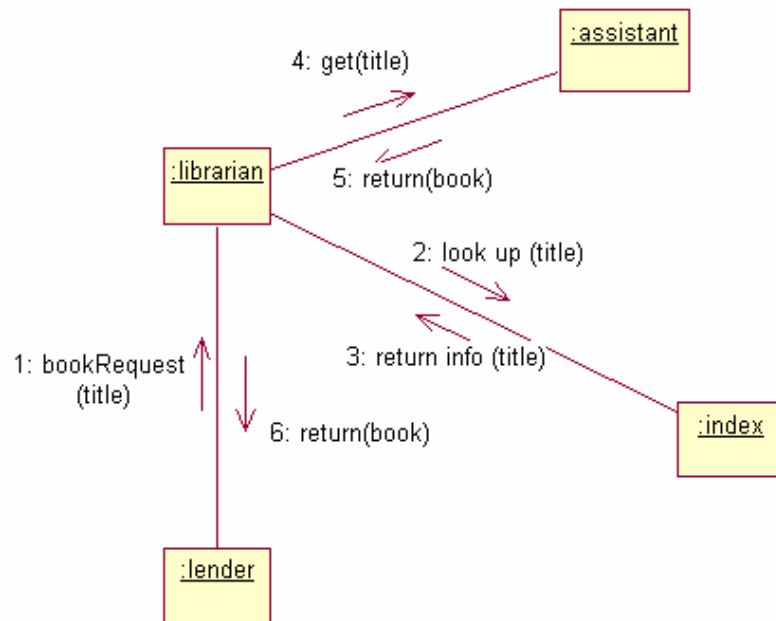
Biểu đồ lớp

- Class diagram
- Biểu đồ quan trọng trong cả bước phân tích và thiết kế
- Khi phân tích, biểu đồ lớp thể hiện mô hình khái niệm (Conceptual model) và cùng với biểu đồ ca sử dụng là công cụ hiệu quả để phân tích yêu cầu



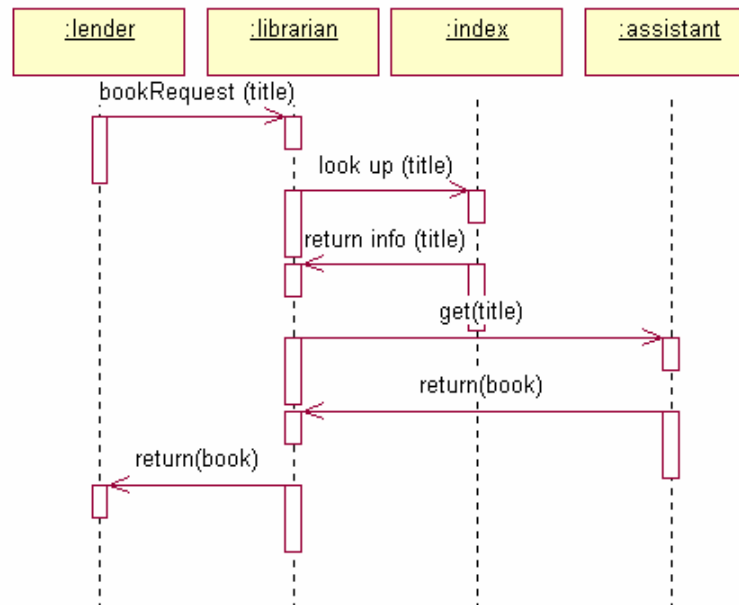
Biểu đồ giao tiếp

- Communication diagram (trong UML 1.0 là Collaboration diagram)
- Mô tả sự tương tác giữa các đối tượng



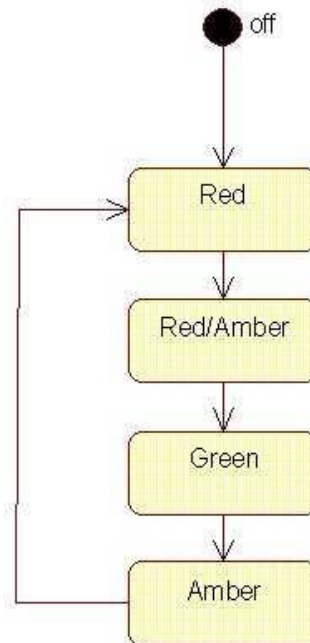
Biểu đồ trình tự

- Sequence diagram
- Tác dụng tương tự như biểu đồ giao tiếp, nhưng rõ hơn về thứ tự thực hiện của các thông điệp



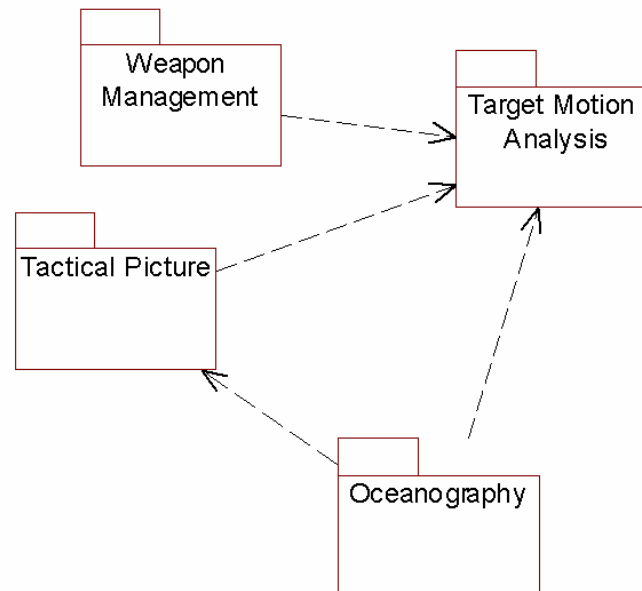
Biểu đồ máy trạng thái

- State machine diagram (trong UML 1.0 là State diagram)
- Một số đối tượng, trong thời điểm nào đó có một trạng thái nhất định



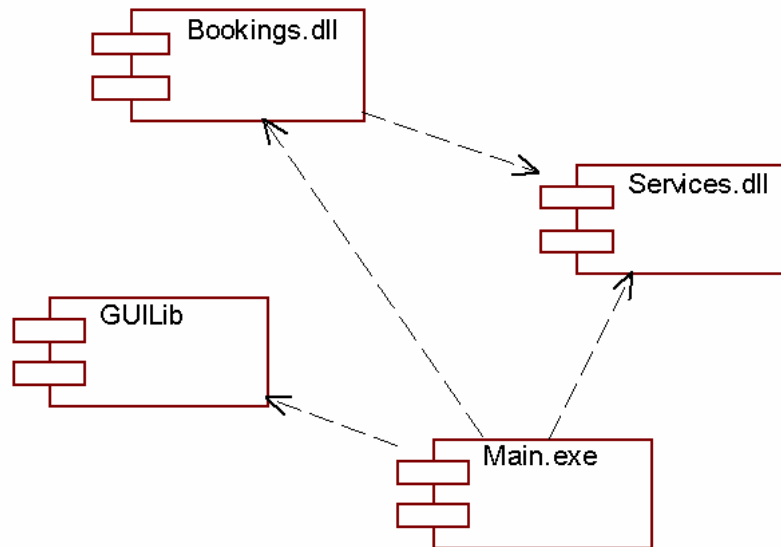
Biểu đồ gói

- Package diagram
- Tương tự như “thư mục” trong quản lý tệp tin
- Có thể có các gói các ca sử dụng, gói các lớp (khi phân tích một hệ thống lớn)



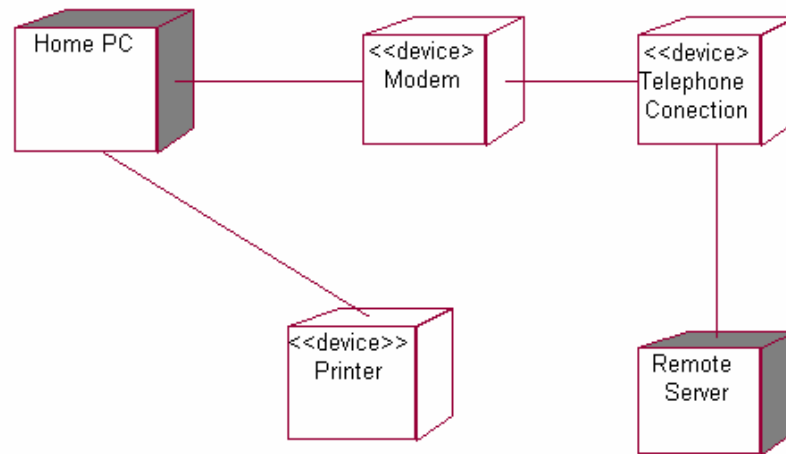
Biểu đồ thành phần

- Component diagram
- Mô tả sự phụ thuộc giữa các thành phần phần mềm (files, link libraries, executables)



Biểu đồ triển khai

- Deployment diagram
- Mô tả về cách phần mềm của chúng ta sẽ được triển khai



Tóm tắt

1. Biểu đồ ca sử dụng: Hệ thống của chúng ta sẽ tương tác với thế giới bên ngoài như thế nào?
 2. Biểu đồ giao tiếp: Các đối tượng sẽ tương tác với nhau như thế nào?
 3. Biểu đồ tuần tự: Các đối tượng sẽ tương tác với nhau như thế nào?
 4. Biểu đồ máy trạng thái: Các đối tượng có các trạng thái nào?
 5. Biểu đồ gói: Chúng ta mô-đun hóa quá trình phát triển của mình như thế nào?
 6. Biểu đồ thành phần: Các thành phần phần mềm của chúng ta sẽ liên quan với nhau như thế nào?
 7. Biểu đồ triển khai: Phần mềm sẽ được triển khai như thế nào?
-