

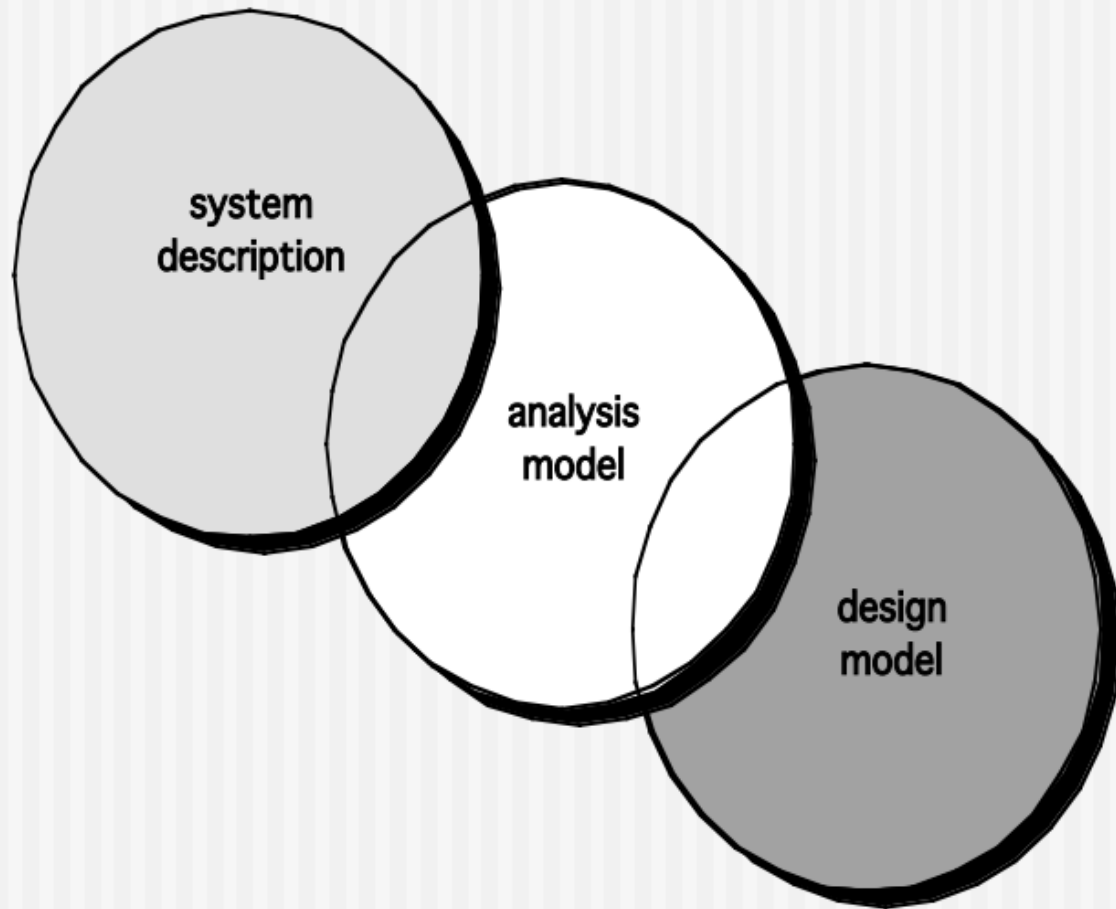
Chương 6

Mô hình phân tích, kịch bản
và phân tích lớp.

Phân tích yêu cầu:

- Phân tích yêu cầu:
 - Xác định đặc điểm của phần mềm
 - Giao diện để phần mềm giao tiếp với các thành phần khác của hệ thống
 - Thiết lập các điều kiện mà phần mềm phải đáp ứng
- Phân tích yêu cầu cho phép kĩ sư phần mềm làm những việc sau:
 - Xây dựng mô hình miêu tả kịch bản sử dụng, chức năng, vấn đề của mỗi lớp, mối quan hệ giữa các lớp, luồng dữ liệu...

Cầu nối



Nguyên tắc ngón tay cái

- Mô hình nên tập chung vào một số yêu cầu nào đó để có thể thấy ngay được vấn đề hoặc những điều cần được đề cập đến.
- Mỗi phần tử thuộc mô hình phân tích cần được thêm vào hiểu biết tổng thể của yêu cầu và cung cấp cái nhìn sâu sắc về miền thông tin, chức năng và hành vi của phần mềm.

Nguyên tắc ngón tay cái

- Đảm bảo rằng mô hình phân tích phục vụ cho tất cả các bên tham gia.
- Đảm bảo mô hình là đơn giản nhất có thể.
- Trì hoãn việc xem xét các cơ sở hạ tầng và các mô hình phi chức năng cho đến khi thiết kế.
- Giảm thiểu sự kết nối trên toàn hệ thống.

Miền phân tích

- Phân tích miền là việc xác định, phân tích và đặc tả các yêu cầu từ một miền ứng dụng cụ thể, thường là để tái sử dụng trong nhiều dự án trong miền ứng dụng. . . Phân tích hướng miền đối tượng là việc xác định, phân tích khả năng tái sử dụng trong một miền ứng dụng cụ thể, về đối tượng phổ biến, các lớp...

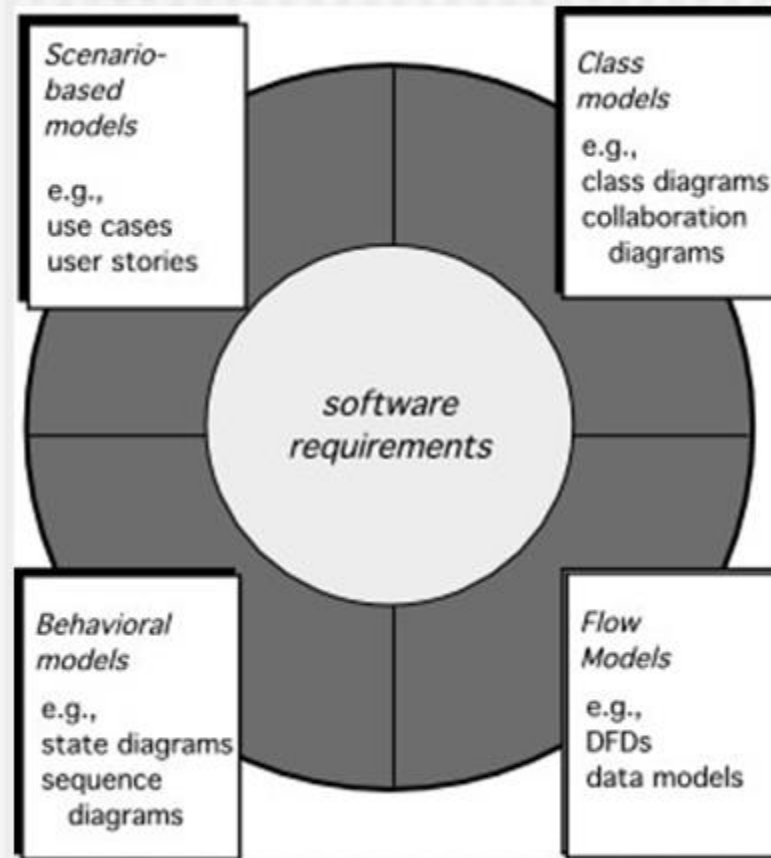


Miền phân tích

- Xác định miền được khảo sát.
- Thu thập các mẫu ứng dụng trong miền.
- Phân tích các ứng dụng thu thập được.
- Phát triển mô hình phân tích cho các đối tượng.



Các thành phần của phân tích yêu cầu.



Mô hình kịch bản cơ sở

- Sơ đồ use-case được thiết kế để xác định xem thứ gì diễn ra bên ngoài hệ thống (tác nhân) và thứ gì được thực hiện bởi hệ thống.
- Những điều nên cân nhắc khi xây dựng sơ đồ use-case:
 - Viết về cái gì?
 - Nên viết bao nhiêu về nó?
 - Miêu tả tới mức độ chi tiết thế nào?
 - Tổ chức các miêu tả thế nào?
-

Viết về cái gì?

- Bắt đầu và gợi ý: cung cấp thông tin bạn cần để bắt đầu vẽ các use-case
- Thu thập thông tin từ các cuộc họp, QFD, các yêu cầu về kĩ thuật để:
 - Xác định các bên liên quan
 - Xác định phạm vi của vấn đề
 - Xác định mục tiêu tổng thể
 - Thiết lập các ưu tiên
 - Vạch ra các yêu cầu chức năng được yêu cầu
 - Đưa ra các đối tượng được điều khiển bởi hệ thống
- Để bắt đầu vẽ các use-case hãy liệt kê các chức năng hoặc hoạt động thực hiện bởi một tác nhân cụ thể.

Viết bao nhiêu?

- Use-case được viết cho mỗi chức năng được ghi nhận
- Nói chung, use-case được viết đầu tiên theo một cách không chính thức.
- Nếu cần thiết, use-case sẽ được vẽ lại theo một cấu trúc đã được định trước.



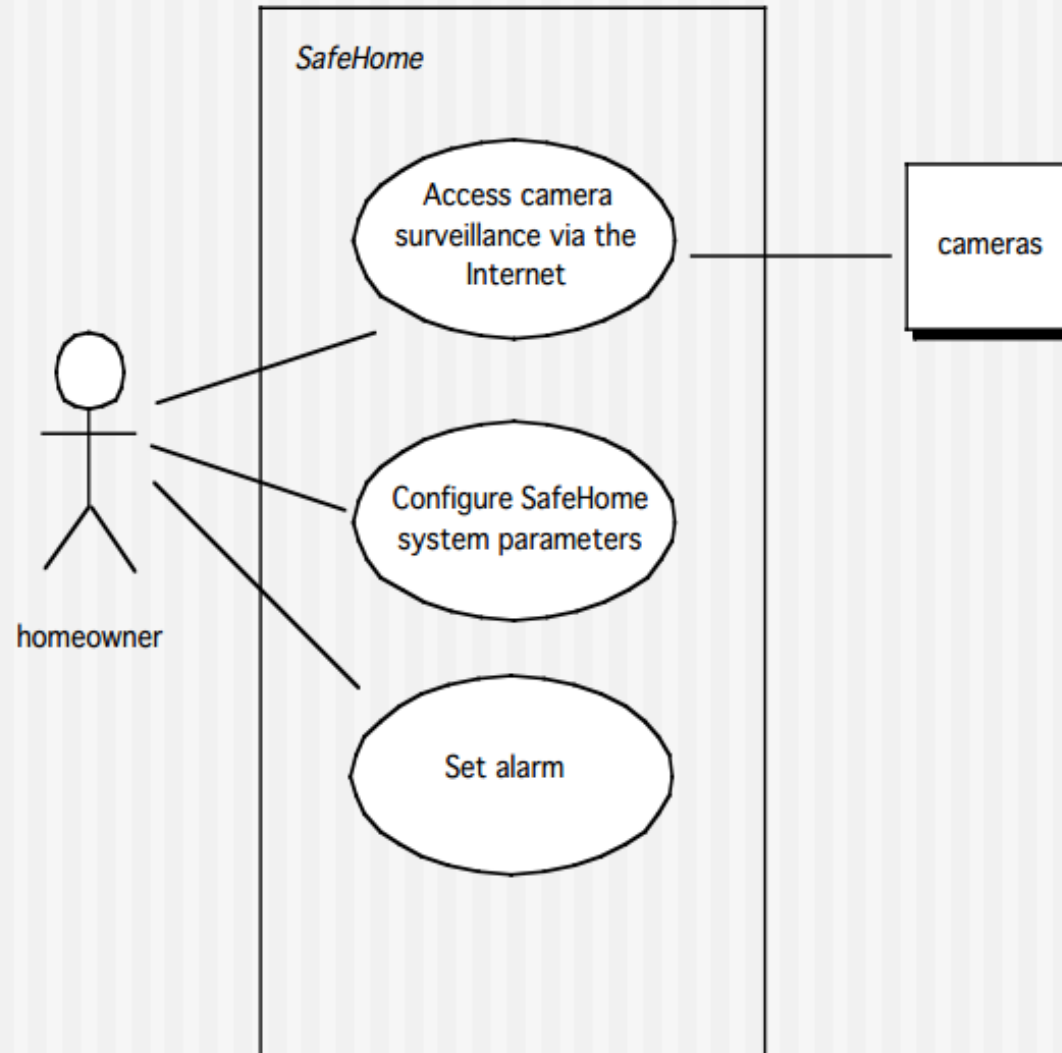
Use-case

- Kịch bản mô tả luồng sử dụng hệ thống
- Tác nhân: đóng vai trò là người hoặc hệ thống khác sử dụng các chức năng của hệ thống cần phát triển
- Người dùng có thể đóng một số vai trò khác nhau trong kịch bản

Xây dựng biểu đồ use-case

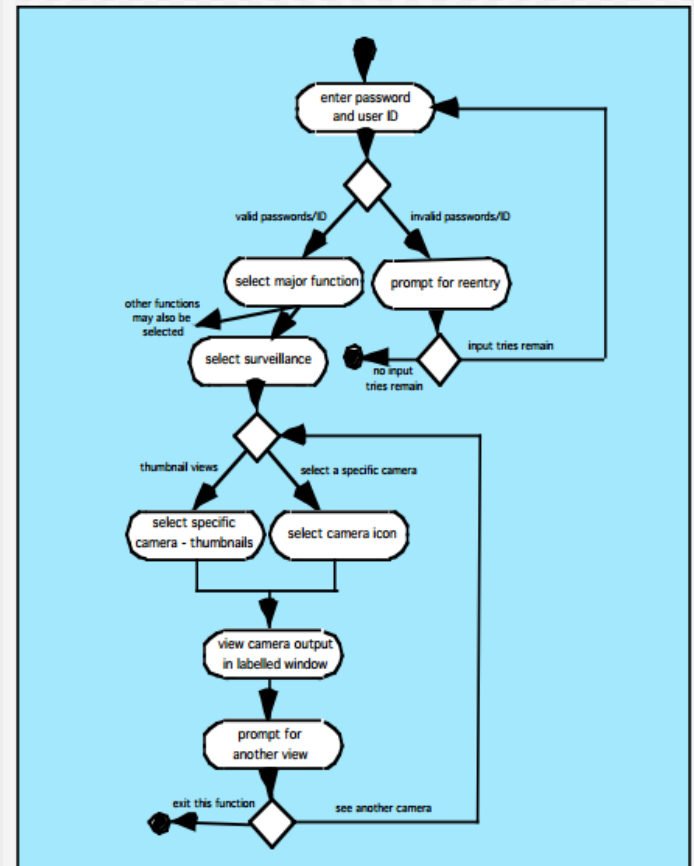
- Nhiệm vụ/chức năng được thực hiện bởi tác nhân là gì?
- Thông tin hệ thống mà tác nhân có được, thay đổi hoặc tạo ra?
- Tác nhân thông báo cho hệ thống biết về thay đổi bên ngoài?
- Thông tin tác nhân mong muốn từ hệ thống?
- Tác nhân mong muốn được thông báo về thay đổi bất ngờ?

Biểu đồ use-case



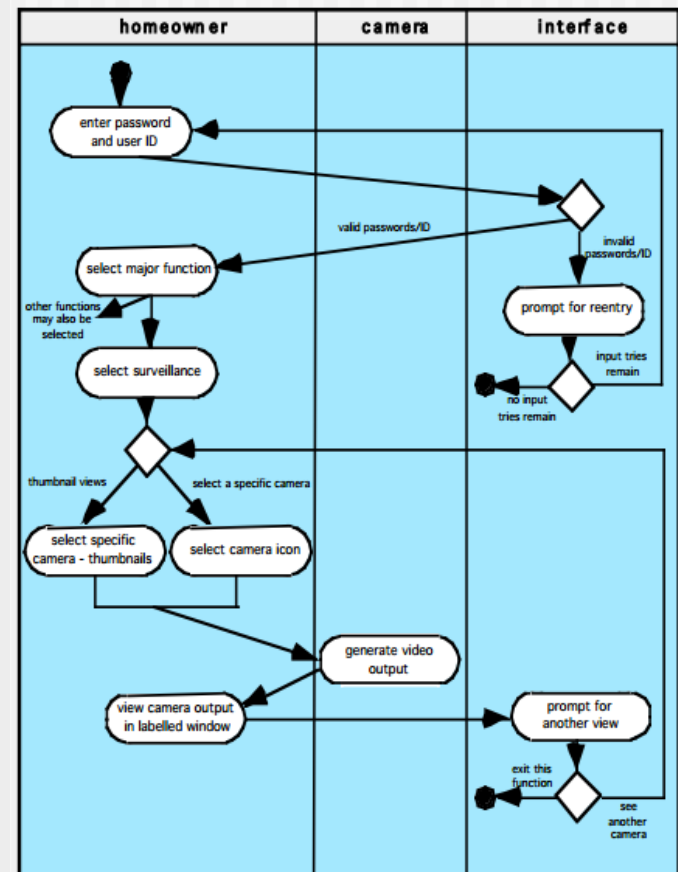
Biểu đồ activity

Biểu đồ activity bổ xung cho biểu đồ use-case bằng cách cung cấp đồ họa miêu tả luồng tương tác trong một kịch bản cụ thể.



Biểu đồ Swimlane

Cho phép mô hình thể hiện luồng activity được miêu tả bằng use-case và đồng thời chỉ ra tác nhân.



Mô hình dữ liệu

- Kiểm tra dữ liệu độc lập với việc xử lí
- Tập trung vào miền dữ liệu
- Tạo ra một mô hình trừu tượng ở cấp độ khách hàng
- Chỉ ra các dữ liệu liên quan đến nhau theo cách nào?



Đối tượng dữ liệu là gì?

- Là thông tin tổng hợp được hiểu bởi phần mềm
 - thông tin tổng hợp: tập các thuộc tính khác nhau
- Có thể là một thực thể bên ngoài (thực thể tạo ra hoặc xử lý thông tin), sự vật...
- Việc mô tả đối tượng dữ liệu kết hợp đối tượng dữ liệu và tất cả thuộc tính của nó.
- Một đối tượng dữ liệu chỉ bao gồm dữ liệu, không bao gồm các thao tác trên dữ liệu đó.

Đối tượng dữ liệu và thuộc tính.

- Một đối tượng dữ liệu bao gồm một tập các thuộc tính và được hiểu như là khía cạnh, đặc điểm của đối tượng.

object: automobile

attributes:

make

model

body type

price

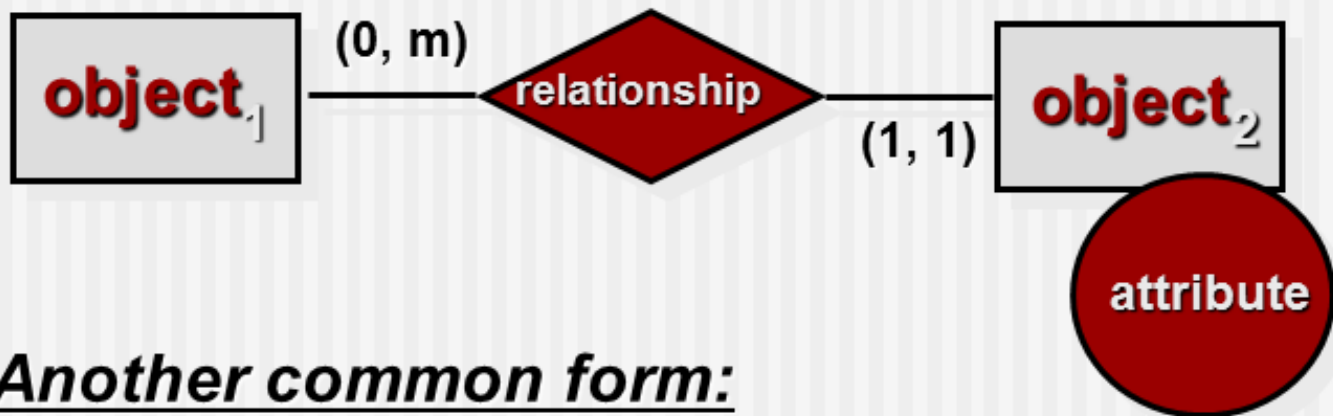
options code

Mối quan hệ

- Một đối tượng dữ liệu kết nối với một đối tượng dữ liệu khác theo các cách khác nhau.
 - Mối quan hệ giữa đối tượng người và xe được thiết lập bởi có quan hệ : người sở hữu xe và xe được lái bởi người.
- Mối quan hệ sở hữu và lái thiết lập mối liên quan giữa xe và người.
- Một vài thể hiện của kết nối có thể tồn tại
- Và các đối tượng liên quan tới nhau theo nhiều cách.

Kí hiệu ERD

One common form:



Another common form:

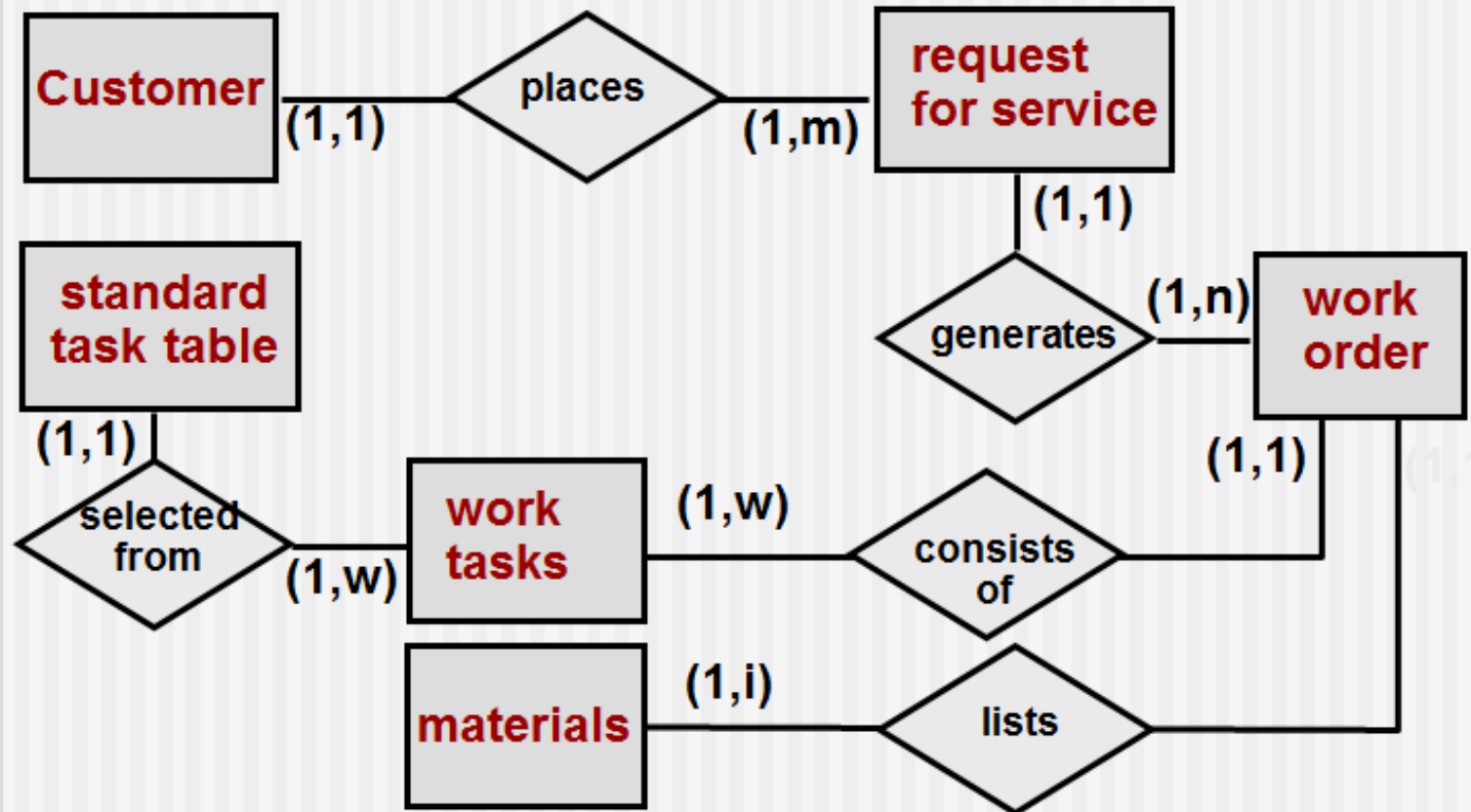


Xây dựng ERD

- Level 1: tất cả các đối tượng và kết nối giữa chúng.
- Level 2: mô hình thực thể và mối quan hệ
- Level 3: mô hình thực thể, các mối quan hệ và thuộc tính



Ví dụ về ERD



Mô hình lớp cơ sở

- Mô hình lớp cơ sở đại diện cho:
 - Đối tượng mà hệ thống thao tác
 - Chức năng cung cấp cho các đối tượng để thực hiện các thao tác
 - Quan hệ giữa các đối tượng
 - Sự cộng tác giữa các lớp được định nghĩa.
- Các yếu tố của một mô hình lớp cơ sở bao gồm các lớp, đối tượng, thuộc tính, chức năng, mô hình CRC, sơ đồ cộng tác, các gói...

Xác định phân tích lớp

- Kiểm tra việc kịch bản sử dụng như là một yêu cầu trong mô hình phân tích.
 - Các lớp học được xác định bằng cách gạch dưới mỗi danh từ hoặc cụm danh từ và đưa nó vào một bảng đơn giản.
 - Các lớp tương đồng cần được chú ý.
 - Nếu một lớp được yêu cầu để giải quyết vấn đề, nó là một phần của giải pháp, nếu một lớp cần thiết để mô tả giải pháp thì nó là một phần của vấn đề.
- Chúng ta nên tìm kiếm gì khi các lớp đã được phân tách?

Biểu hiện của phân tích lớp

- Biểu hiện của phân tích lớp thể hiện theo những cách sau:
 - Thực thể bên ngoài (tạo ra hoặc xử lí thông tin)
 - Sự vật (một phần miền thông tin của vấn đề)
 - Sự kiện xảy ra trong khi hệ thống hoạt động
 - Vai trò của những người tương tác với hệ thống
 - Bối cảnh của vấn đề và chức năng tổng thể
 - Cấu trúc định nghĩa các đối tượng

Lớp tiềm năng

- Lưu giữ thông tin: lớp tiềm năng sẽ có ích trong quá trình phân tích khi thông tin của nó phải được sử dụng bởi hệ thống.
- Dịch vụ cần thiết: lớp tiềm năng phải có một tập hợp các chức năng để có thể thay đổi giá trị của các thuộc tính của nó trong một số cách.
- Nhiều thuộc tính: trong phân tích yêu cầu, nên tập trung vào thông tin quan trọng; một lớp có thể có một thuộc tính
- Thuộc tính chung. Một tập hợp các thuộc tính có thể được định nghĩa cho lớp tiềm năng và những thuộc tính này áp dụng đối với tất cả các thể hiện của lớp.

Lớp tiềm năng

- Chức năng chung: Một tập hợp các chức năng có thể được xác định cho các lớp tiềm năng và các hoạt động này áp dụng đối với tất cả các thể hiện của lớp.
- Yêu cầu thiết yếu. Thực thể bên ngoài (xuất hiện trong vấn đề, sản xuất hoặc tiêu thụ thông tin cần thiết cho hoạt động của bất kỳ giải pháp nào trong hệ thống) sẽ luôn luôn được định nghĩa như là các lớp trong mô hình yêu cầu.

Định nghĩa thuộc tính

- Thuộc tính mô tả lớp có trong mô hình phân tích
- Vd: xây dựng 2 lớp cho người chơi bóng truyền chuyên nghiệp



Xác định các chức năng:

- Chức năng có thể chia làm 4 loại lớn:
 - Chức năng xử lý dữ liệu theo cách nào đó
 - Chức năng thực hiện tính toán
 - Chức năng cần đến trạng thái của các đối tượng
 - Chức năng theo dõi đối tượng



Mô hình CRC

- Class-responsibility-collaborator (CRC): Mô hình này cung cấp một phương tiện đơn giản để xác định và tổ chức các lớp có liên quan đến hệ thống hoặc sản phẩm yêu cầu
- Mô hình CRC là một tập hợp các thẻ đại diện cho các lớp, các thẻ gồm 3 phần, phần đầu tiên ghi tên lớp, phần thứ hai ghi chức năng của lớp và phần thứ 3 ghi các lớp cộng tác với nó.

Mô hình CRC

Class: FloorPlan	
Description:	
Responsibility:	Collaborator:
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

Kiểu lớp

- Lớp thực thể: được xây dựng trực tiếp từ các vấn đề.
- Lớp bound: được dùng để tạo ra các giao diện để các phần khác tương tác với nhau.
- Lớp điều khiển: quản lí các thao tác từ đầu đến cuối :
 - Tạo và cập nhật các đối tượng
 - Thể hiện các đối tượng lớp bound khi có sự giao tiếp qua giao diện
 - Giao tiếp phức tạp giữa các đối tượng
 - Xác nhận dữ liệu trao đổi giữa các đối tượng với đối tượng hoặc với người dùng

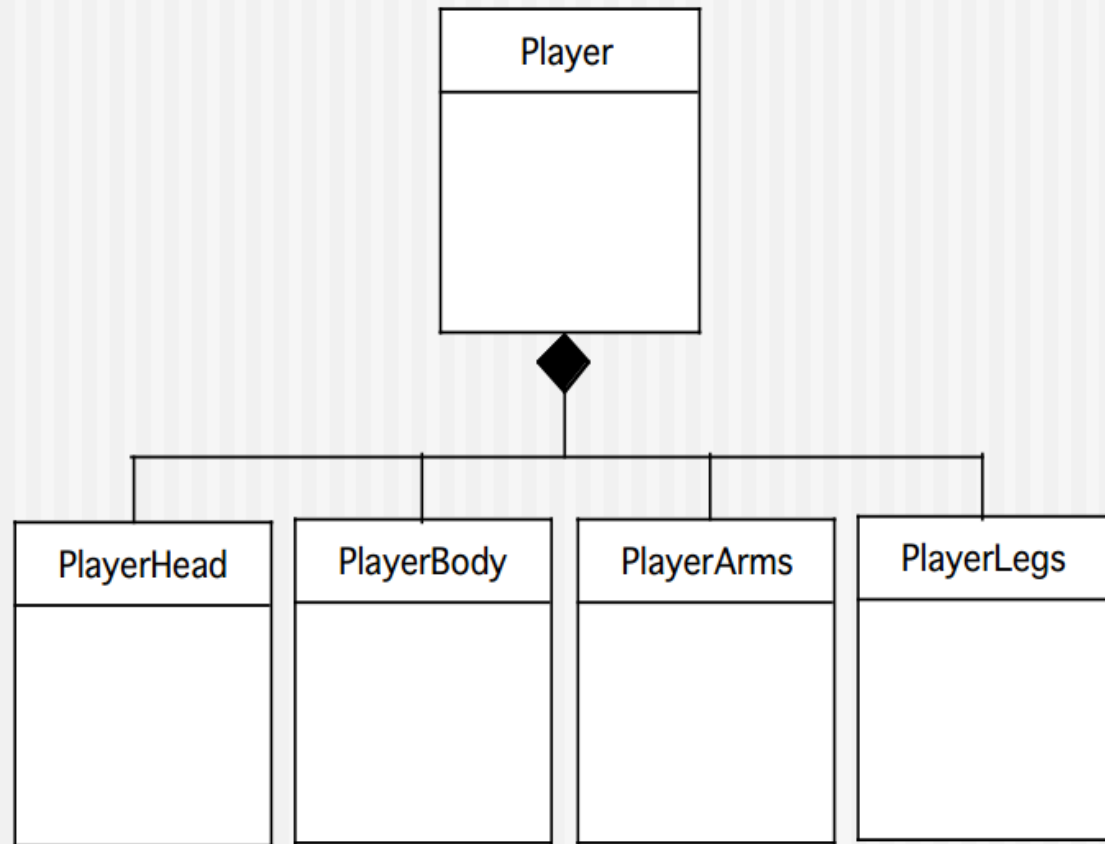
Trách nhiệm

- Một hệ thống thông minh nên được xây dựng từ các lớp để giải quyết hiệu quả vấn đề.
- Thông tin và hành vi liên quan đến nó nên đặt trong cùng một lớp
- Thông tin về một vật thể nên được đặt trong một lớp, tránh phân bố ra nhiều lớp.
- Trách nhiệm cần được chia sẻ giữa các lớp liên quan

Kết hợp

- Một lớp thực hiện trách nhiệm của mình theo 2 cách:
 - Sử dụng chức năng của mình để biến đổi thông tin của mình
 - Kết hợp với một lớp khác.
- Kết hợp xác định mối quan hệ giữa 2 lớp
- Kết hợp được xác định bằng cách xem xét một lớp có thể tự giải quyết được trách nhiệm hay không?
- 3 quan hệ logic giữa các lớp
 - Is a part of
 - has-knowledge-of
 - depends-upon

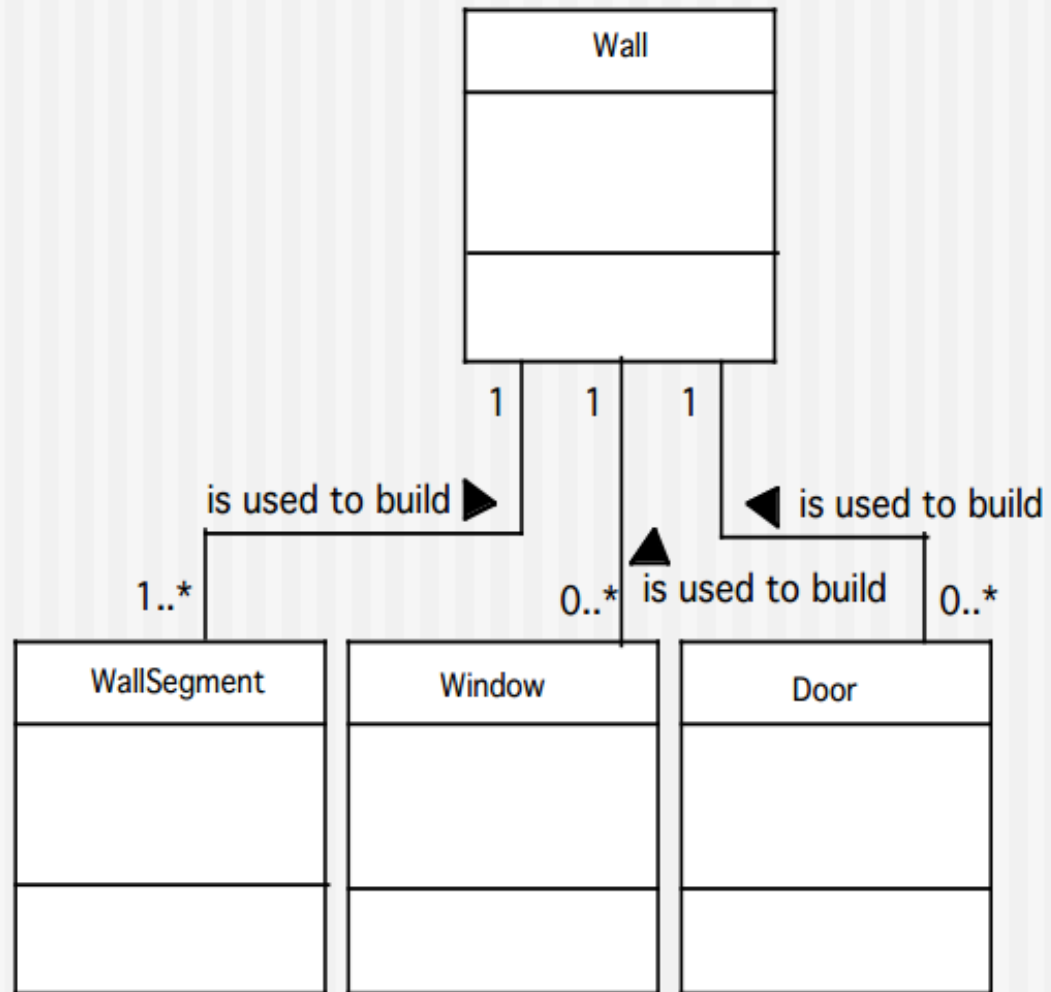
Kết tập lớp



Liên kết và phụ thuộc

- Hai lớp thường liên quan đến nhau theo một số cách
 - Trong uml các mối liên quan này gọi là liên kết
 - Liên kết có thể thể hiện tính nhiều
- Trong nhiều trường hợp, tồn tại mối quan hệ client-server giữa hai lớp
 - Trong trường hợp này, một lớp client phụ thuộc vào một lớp server và quan hệ phụ thuộc được hình thành.

Liên kết bội



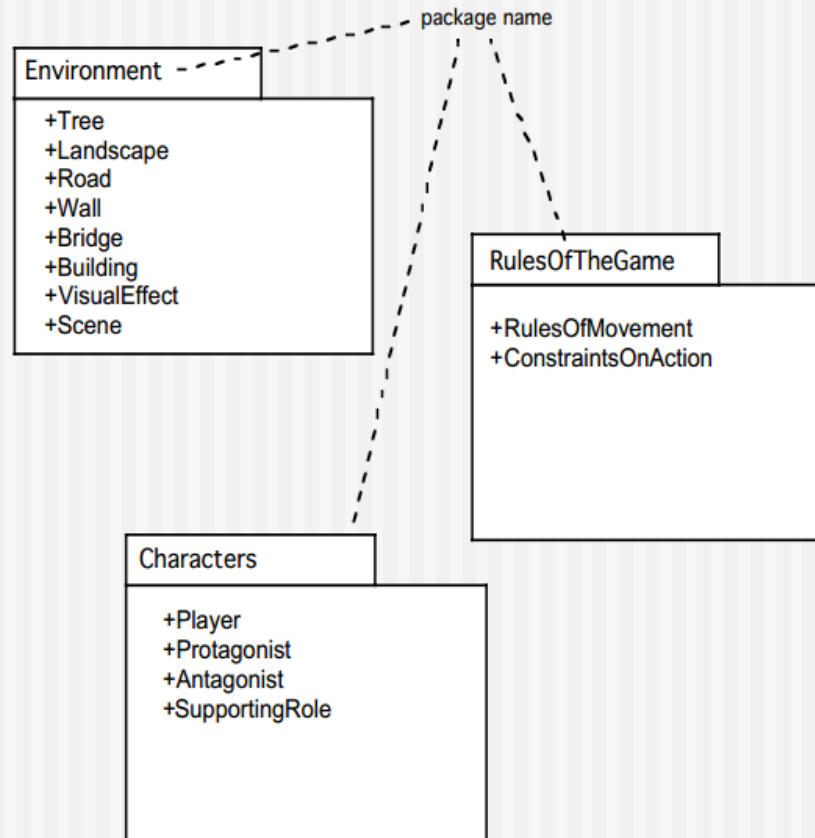
Phụ thuộc



Phân tích gói

- Các thành phần của các mô hình phân tích được phân loại theo cách thức đóng gói chúng (theo nhóm)
- Các dấu cộng trước tên lớp phân tích trong mỗi gói chỉ ra rằng các lớp đó có thể truy cập từ các gói khác.
- Một dấu trừ chỉ ra rằng một phần tử bị ẩn đối với các gói khác và một biểu tượng # chỉ ra rằng một yếu tố chỉ có thể truy cập đến các lớp chứa trong một gói đó.

Phân tích gói



Review mô hình CRC
