PHÂN TÍCH THIẾT KẾ HỆ THỐNG
Tiếp cận hướng đối tượng

Tháng 9-2007

ThS. Nguyễn Anh Hào

Khái niệm tiếp cận hướng đối tượng

- Tiếp cận hướng xử lý (mô tả các công việc của hệ thống), hay dữ liệu (mô tả thực thể và các quan hệ) là những cách tiếp cận có cấu trúc có đặc điểm chung: Các mô tả (ví dụ: các thành phần của hệ thống) được tạo ra từ quan điểm nhận thức của người phân tích viên, có thể rất khác biệt với thế giới thực.
- Trong cách tiếp cập hướng đối tượng, các thành phần làm việc trong hệ thống được ví như các đối tượng "đang sống"; chúng có thể phát triển tương đối độc lập với hệ thống mà không làm phá vỡ hệ thống.
- Tiếp cận hướng đối tượng là đặt quan điểm tìm hiểu hệ thống từ góc nhìn của từng đối tượng đang tồn tại trong thực tế (hoặc nhận thức phổ biến của mọi người) mà nó có thể tham gia giải quyết những vấn đề của hệ thống.

Khái niệm tiếp cận hướng đối tượng

- Tiếp cận OO là cách trừu tượng hóa về các khái niệm trong thế giới thực : các đối tượng trong thế giới thực được mô hình hóa thành các lớp đối tượng (classes) có trạng thái (state), hành vi (behavior) và các mối quan hệ: tổng quát hóa (generalization), kết tập (aggregation), liên kết (association), cộng tác (collaboration), ứng phó với các sự kiện (sequence, state transition, activity),… của các đối tượng trong một hệ thống đang tồn tại.
- Do đó, ưu điểm của tiếp cận OO là các mô tả đều rất gần gủi với nhận thức của nhiều người nên dể kiểm chứng, dể chia sẽ và khả thi hơn cách tiếp cận hướng cấu trúc.

Mô hình hóa trong OOAD

- Có (ít nhất) là 8 nguyên lý hỗ trợ cho OOAD:

1. Đóng gói (Encapsulation)
2. Phân lớp đối tượng (Classification)
3. Tổng quát hóa / cụ thể hóa (Gen./Spec.)
4. Đa hình (Polymorphism)
5. Kết tập (Aggregation)
6. Kết hợp (Association)
7. Hợp tác (Collaboration)
8. Truyền thông điệp (message passing)

(Practical Object-Oriented Development with UML and JAVA. Richard C.Lee, William M.Tepfenhart, Pearson Education, 2002)

Đối tượng (Object)

- *Đối tượng*: là sự vật có một vai trò trong một bối cảnh ứng dụng, có định danh (tên gọi), trạng thái (thuộc tính), hành vi (phương thức). Một đối tượng có thể quan sát được (con người), hoặc 1 ý niệm (nghề nghiệp, chuyên khoa).
- *Trạng thái*: bao gồm thuộc tính và giá trị của thuộc tính tại mỗi thời điểm.
- *Hành vi*: diễn tả 1 đối tượng hành động hoặc phản ứng như thế nào trong môi trường.

Object

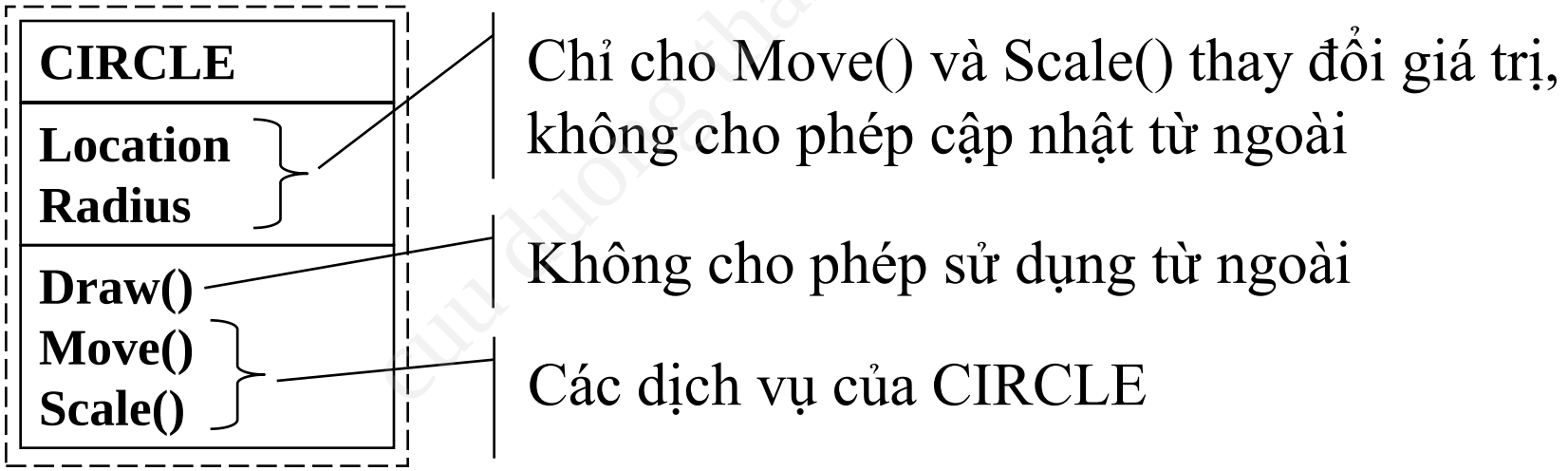
Bill : PERSON
Name= Bill **DOB = 4/15/1978** **Addr = 12 Main St.**
Earn_for_living()

Class

PERSON
Name **DOB** **Addr**
Earn_for_living()

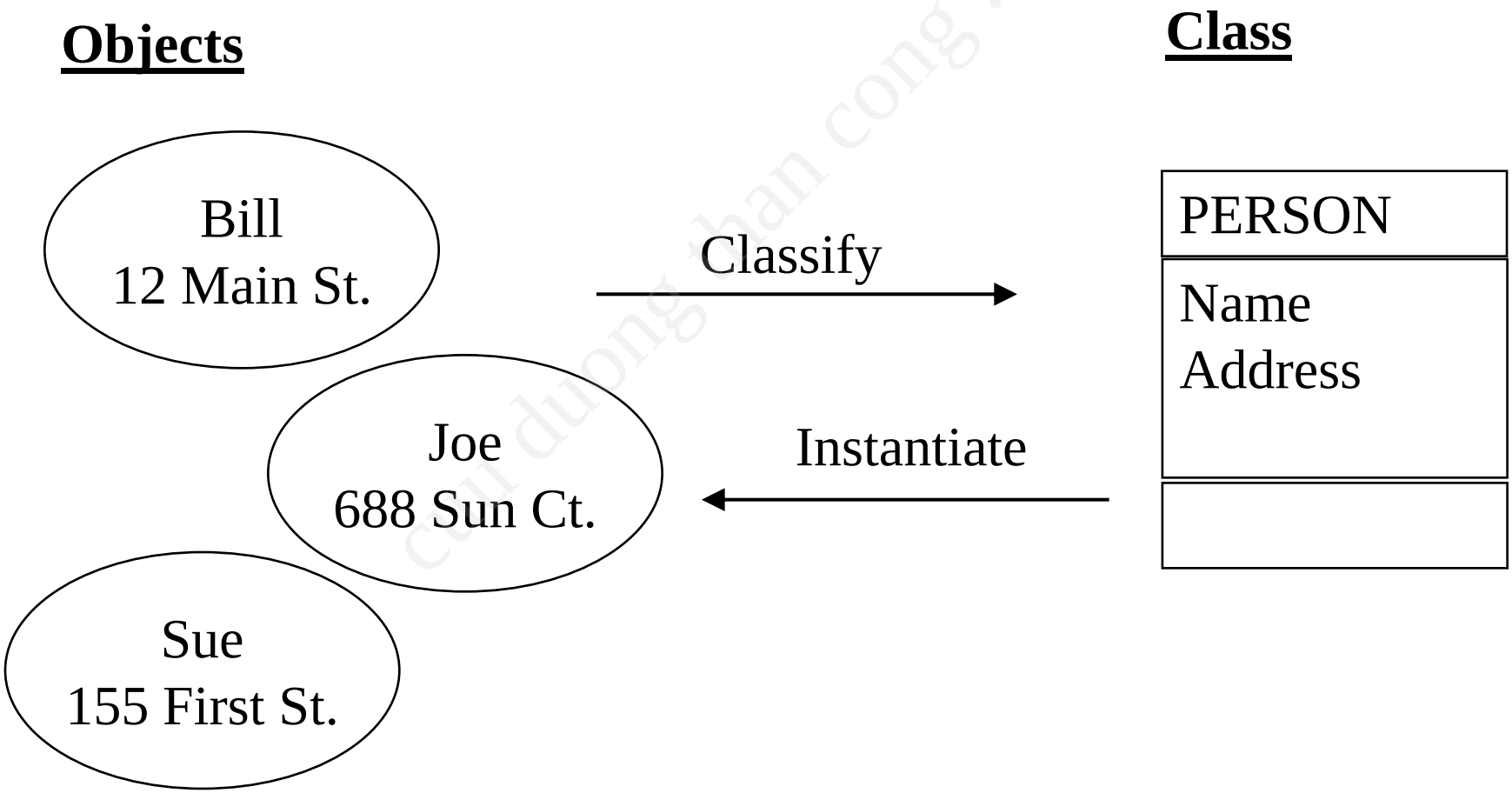
1. Đóng gói (Encapsulation)

- Là cơ chế gắn kết trạng thái của đối tượng và các hành vi của đối tượng vào trong một thể thống nhất (đóng gói). Việc đóng gói giúp phân lập những gì thuộc đối tượng và không thuộc đối tượng, để bảo vệ đối tượng.
- dịch vụ = hành vi cung cấp giá trị sử dụng cho bên ngoài



2. Phân lớp đối tượng

➲ Nhóm các đối tượng cùng chung một số đặc điểm vào trong một lớp = lớp đối tượng.



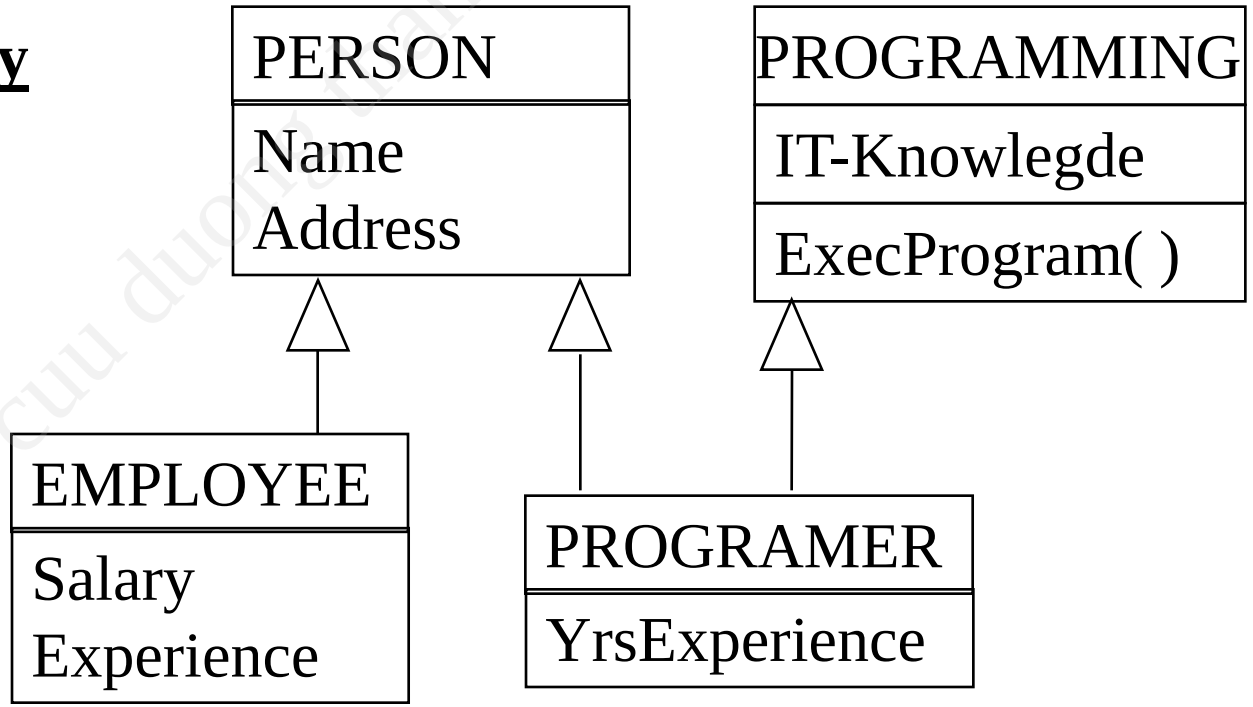
3. Tổng quát hóa/Cụ thể hóa

~ Cấu trúc diễn tả sự giống nhau giữa các lớp (thuộc tính và các dịch vụ), tạo thành sự kế thừa

➲ Tổng quát hóa: tìm điểm giống nhau giữa các lớp (con) để tạo thành lớp tổng quát.

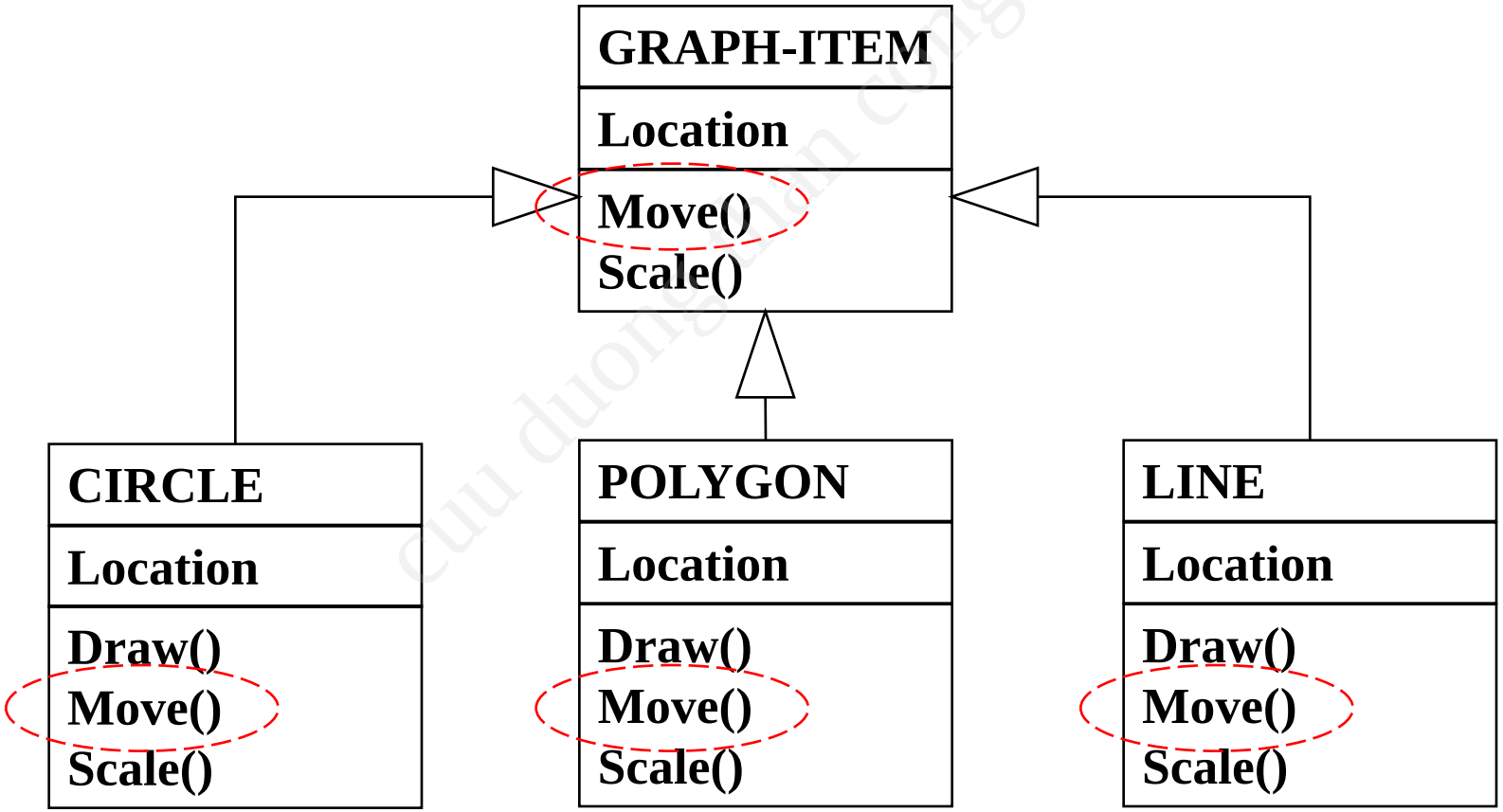
➲ Cụ thể hóa: tìm lớp đối tượng con có đầy đủ đặc điểm của lớp đối tượng cha

Class hierarchy



4. Đa hình hóa (Polymorphism)

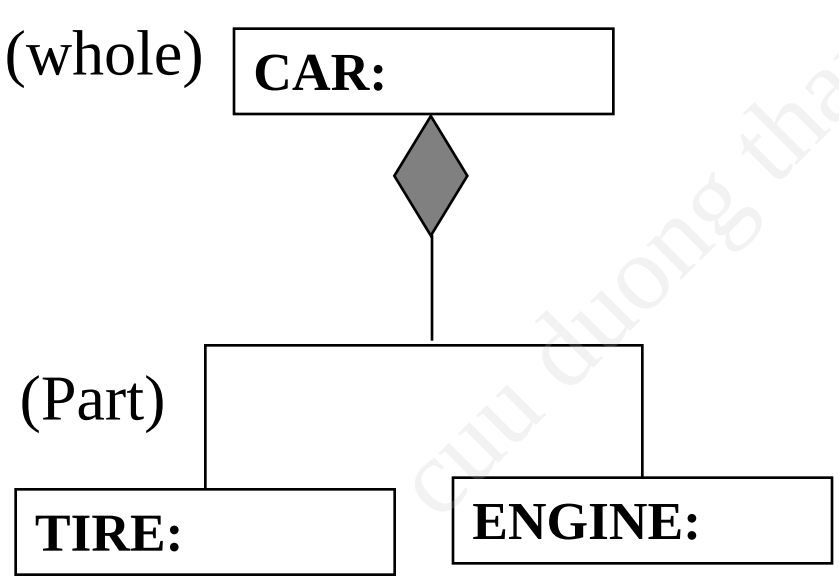
➲ Là cơ chế cho phép đa ngữ nghĩa trong cách sử dụng các dịch vụ của đối tượng; tuy tên gọi cho các dịch vụ rất giống nhau (‘move()’) ở các đối tượng (‘circle’, ‘polygon’, ‘line’), nhưng cách thực hiện các dịch vụ này lại khác nhau.

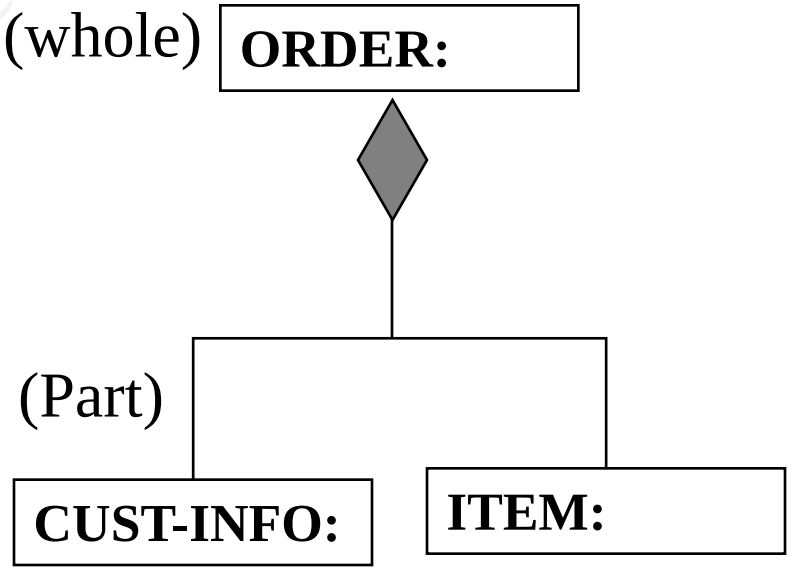


5. Kết tập (Aggregation)

~ một tập các đối tượng hợp thành một đối tượng duy nhất.

- 1 xe hơi bao gồm các bánh xe (tyres) và động cơ (engine).
- 1 đơn đặt hàng gồm thông tin khách hàng (cust-info) và các mặt hàng được yêu cầu (items).





6. Kết hợp (Association)

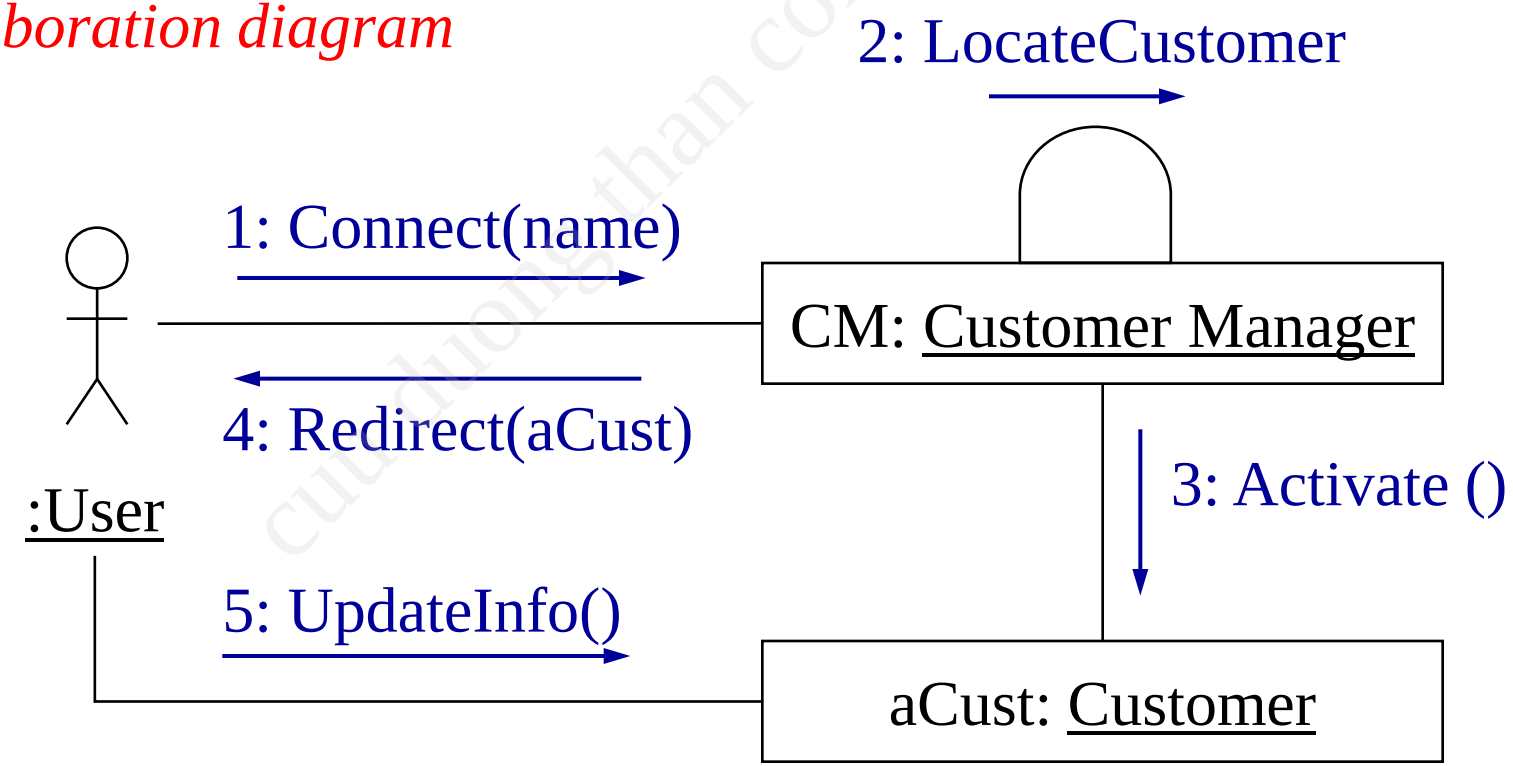
➲ Là mối quan hệ "logic" giữa các đối tượng / lớp đối tượng (giống như quan hệ trong ERD).

7. Hợp tác (Collaboration)

➲ Là sự liên kết dịch vụ giữa các đối tượng, qua cơ chế truyền thông điệp. Nó mô tả sự tương tác giữa các đối tượng để thực hiện một vài hành vi theo một kịch bản nào đó.

User là khách hàng muốn cập nhật dữ liệu của họ vào hệ thống:

Collaboration diagram



8. Truyền thông điệp (Message passing)

- Là cơ chế gửi thông điệp mang yêu cầu từ một đối tượng (Client) đến một đối tượng khác (Agent) để nhờ thực hiện. Đây là cơ chế thực hiện theo nghĩa hợp tác, chứ không phải theo mệnh lệnh:
  1. Việc thông dịch message phụ thuộc ở Agent, và
  2. Agent có thể ủy thác (delegate) cho một Agent khác thực hiện.
- *Sự ủy thác* (Delegation): Thông điệp mang yêu cầu được chuyển đi từ đối tượng này đến đối tượng khác cho đến khi có một đối tượng đáp ứng được yêu cầu.

  Công việc từ Giám đốc chuyển đến Trưởng phòng, Trưởng phòng ủy thác cho chuyên viên thực hiện.

Trình tự thực hiện gồm:

1. Xác định các use cases
2. Tìm các đối tượng xử lý các use cases
3. Thiết lập lược đồ lớp đối tượng (Gen.+Agg.+Ass.)
4. Mô hình hóa cách xử lý của hệ thống
   - Lược đồ tuần tự
   - Lược đồ cộng tác
   - Lược đồ trạng thái
   - Lược đồ hoạt động

- Phân tích, thiết kế: là quá trình gở bỏ hoặc bổ sung dần các thuộc tính, đối tượng và lớp đối tượng trong lược đồ UML cho phù hợp với giải pháp thực tế.

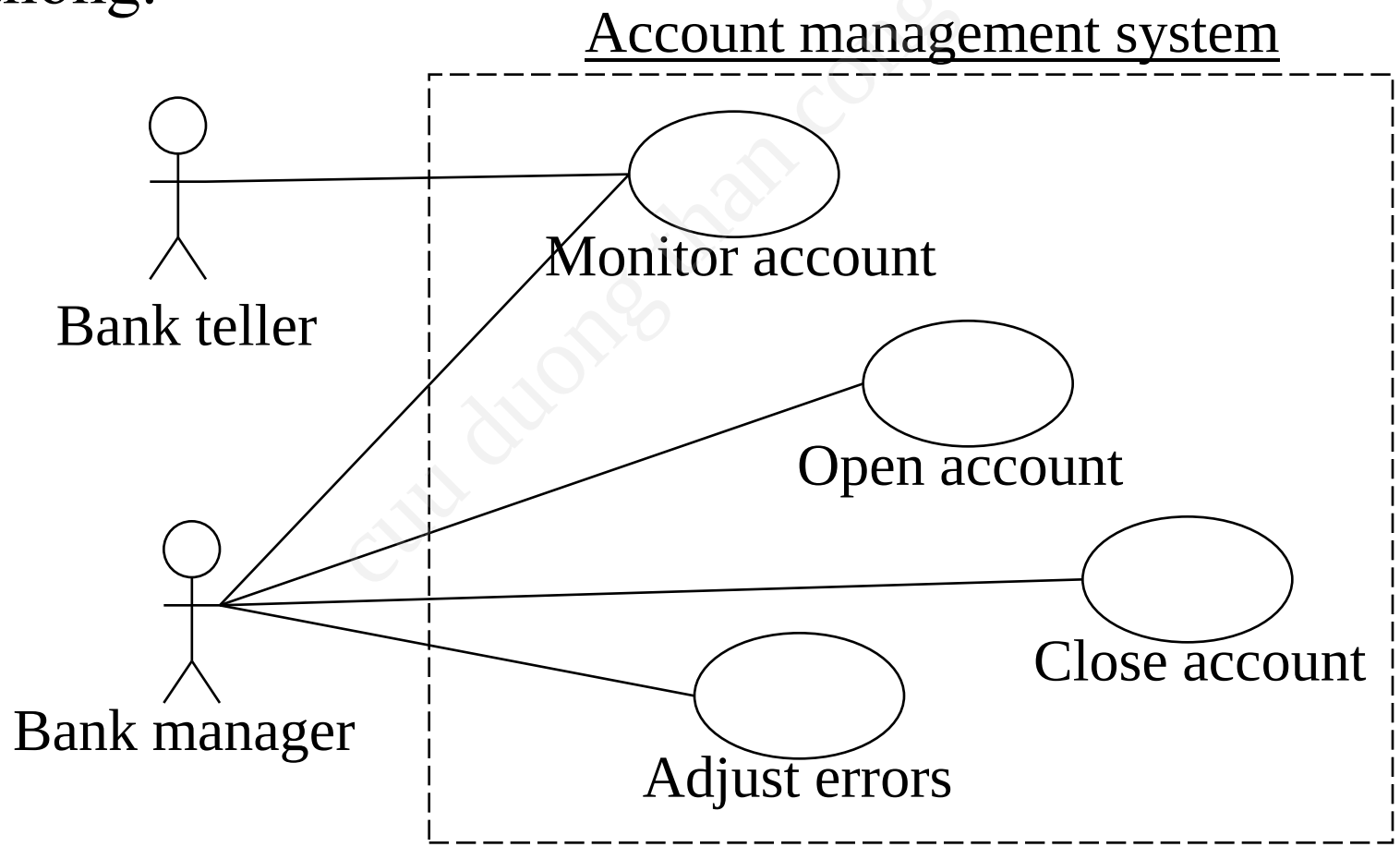
Use case

- Một goal (mục đích) của hệ thống là một giá trị sử dụng (ích lợi) cung cấp cho users (actors) có tương tác với hệ thống đó.
- Một actor (tác nhân) là một đối tượng bên ngoài hệ thống có tương tác với hệ thống.

  - Actor: users, thiết bị ngoại vi, timer,…
- Một use case là một trường hợp tương tác giữa hệ thống với actor, để thực hiện một goal của hệ thống.
  - Nó đặc tả một chuỗi hành động (ie, chức năng) mà hệ thống thực hiện để tạo ra giá trị sử dụng
  - Nó đặc tả một chuỗi các tương tác (ie, kịch bản) giữa hệ thống với một hoặc nhiều actors.

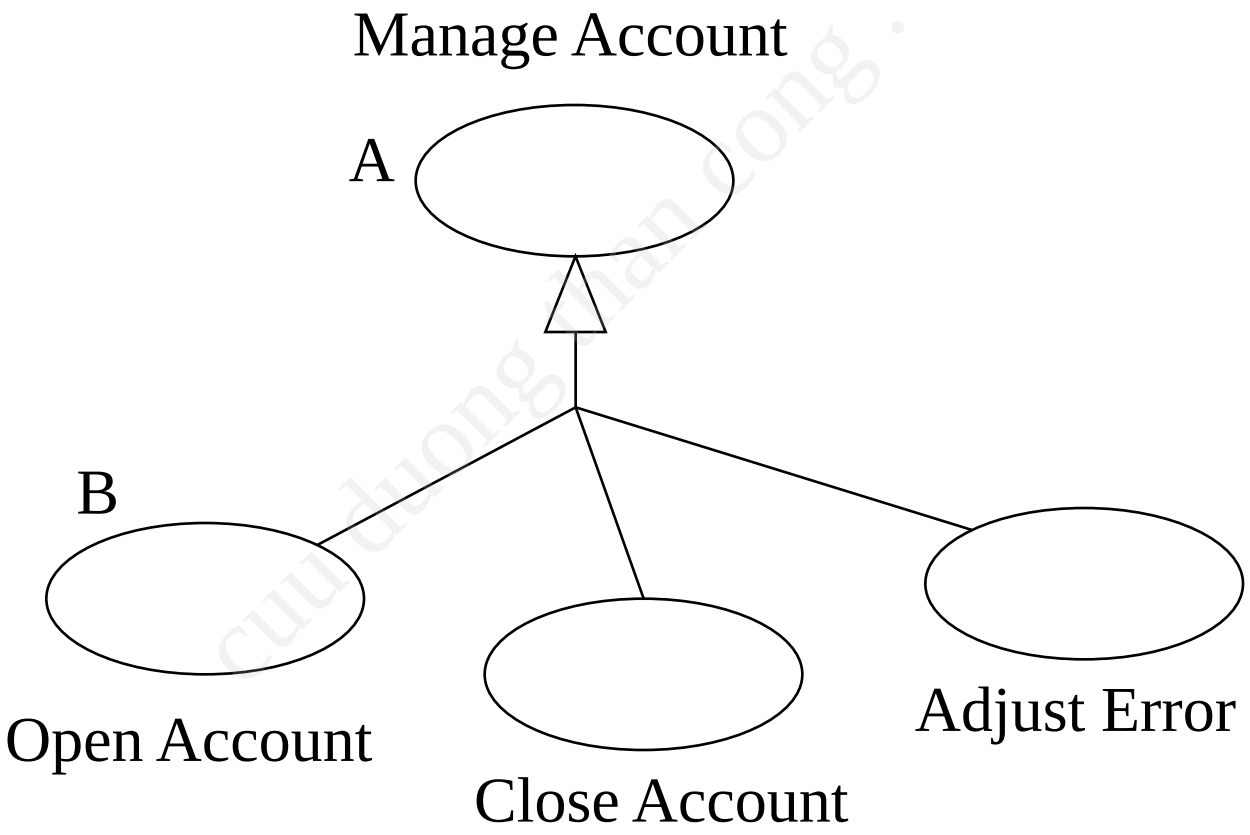
Use case

- Nhìn theo quan điểm của users (actors), hệ thống là một tập hợp các use cases, mỗi use case liên kết với một chức năng do hệ thống cung cấp. Một use case diễn tả một dịch vụ được cung cấp từ một (hoặc một số) đối tượng trong hệ thống.



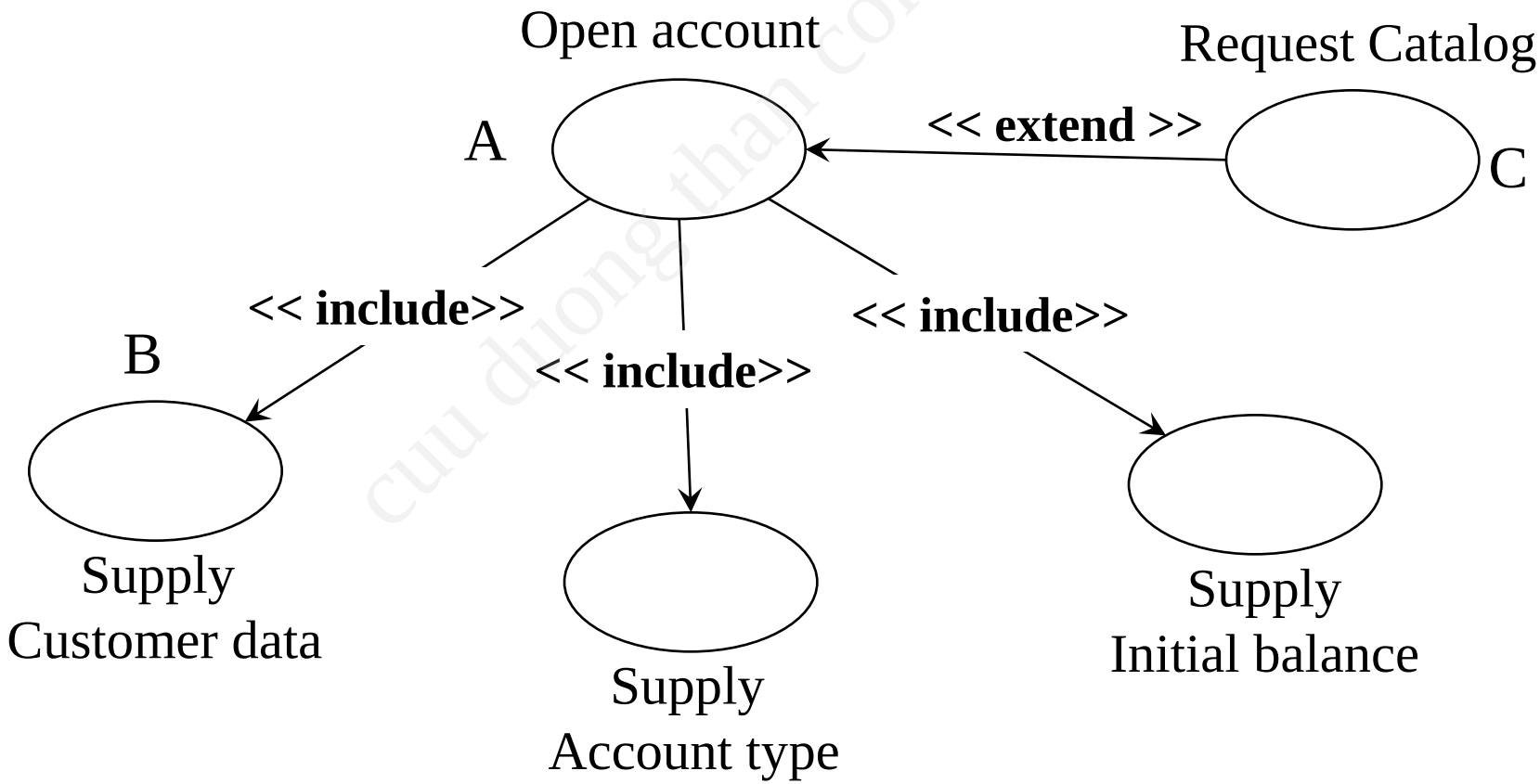
Quan hệ giữa các use case

- ***Tổng quát hóa/cụ thể hóa***: Use case A là một sự tổng quát hóa của use case B nếu use case B là một thể hiện cụ thể hóa của use case A.



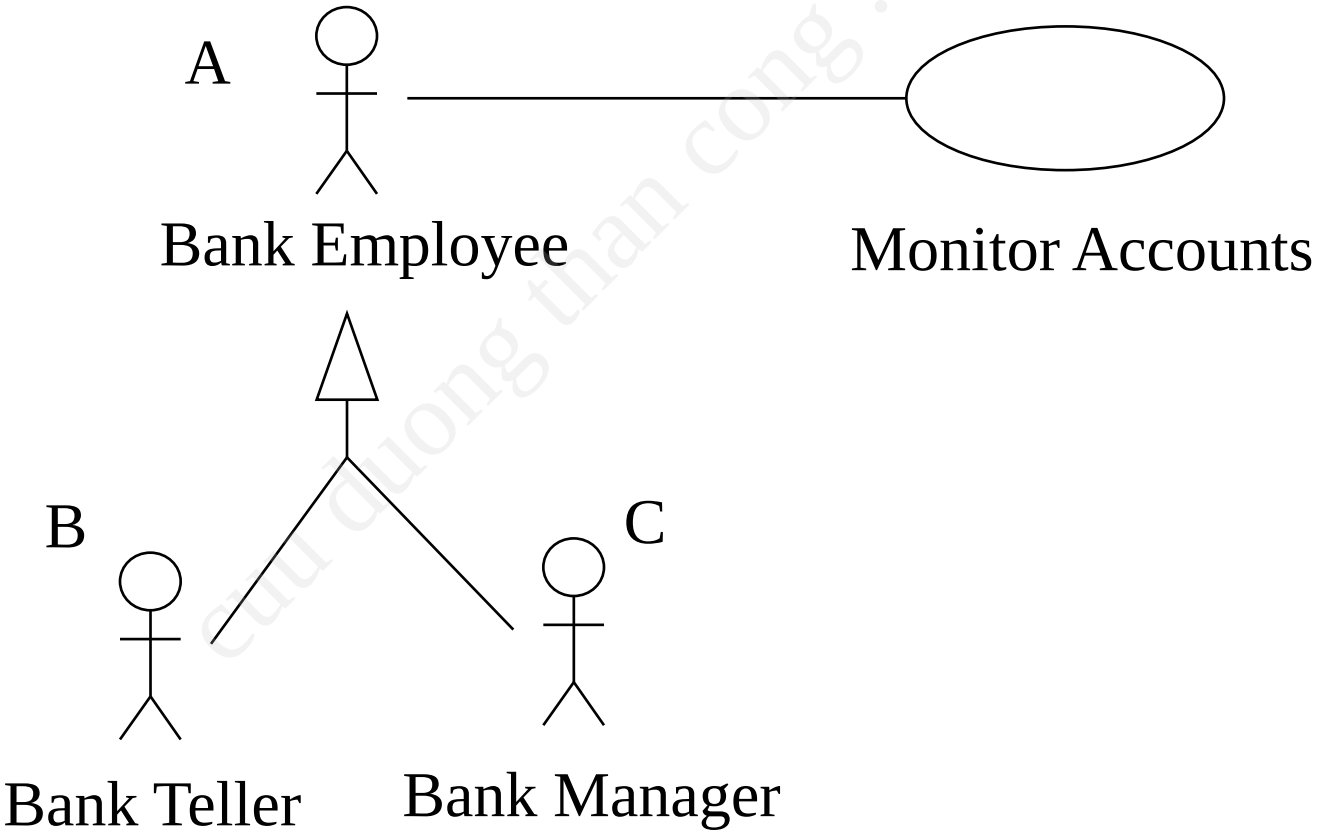
Quan hệ giữa các use case

- ***<<Include>>***: Use case A 'include' use case B khi A cần phải sử dụng use case B (B phải có cho A).
- ***<<Extend>>***: Use case A 'extend' use case C nếu C là một sự mở rộng xử lý của A (A có thể cần C, có thể không).



Quan hệ giữa các actors

- Tổng quát hóa generalation /cụ thể hóa specialization : Nếu Actor A là một sự tổng quát hóa của actor B và C thì những gì A thực hiện được trên hệ thống, các actor B và C cũng làm được.

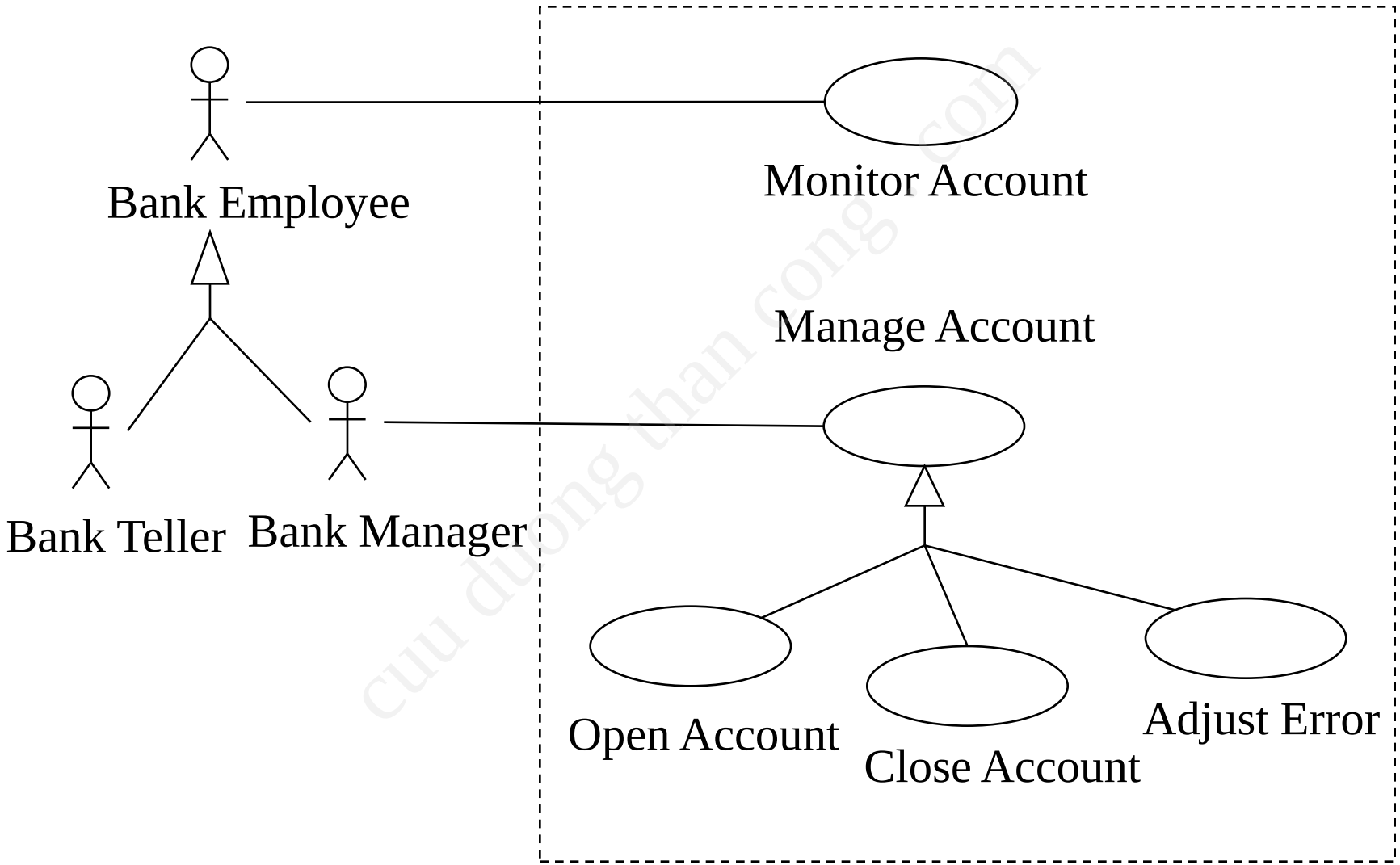


Thiết lập use cases

1. Actors và vai trò trên hệ thống (use cases), có thể là
   - *Users của hệ thống*
   - *Ứng dụng bên ngoài hệ thống*
   - *Thiết bị bên ngoài có tương tác với hệ thống*
   - *Sự kiện kích hoạt hệ thống theo thời gian*
2. Quan hệ giữa các use cases (incl./ext./gen.)
3. Điều kiện để use case được kích hoạt
4. Kết quả mong đợi từ use case
5. Các ngoại lệ (vd: từ chối thực hiện do lỗi)
6. Các ràng buộc (vd: miền giá trị hợp lệ, timeout,..)
7. Sự tùy biến trên các use cases (nếu có)

Lược đồ use case

Lược đồ use case cho hệ thống quản lý tài khoản ngân hàng



Phương pháp tìm các đối tượng

1. Sử dụng vật thể (các khái niệm vật chất quen thuộc)
   - Xác định các vật thể, như con người, vật dụng, tổ chức (trường học, bệnh viện, công ty,…), vị trí địa lý, bản báo cáo,… trong phạm vi của vấn đề đang khảo sát
   - Xác định các đối tượng và lớp đối tượng tương ứng với vật thể
2. Sử dụng cách phân rã đối tượng
   - Tìm kết tập hoặc lớp đối tượng
   - Phân rã (chuyên biệt hóa) thành các đối tượng thành phần topdown
3. Sử dụng cách tổng quát hóa đối tượng
   - Xác định các đối tượng hoặc lớp đối tượng trong vấn đề
   - Tìm các đối tượng có những thuộc tính và dịch vụ tương tự nhau để tổng quát hóa thành lớp đối tượng trừu tượng mà các đối tượng đã biết có thể kế thừa được bott. up

Xác định thuộc tính của đối tượng

➲ **Thuộc tính**: Dữ liệu gì làm đối tượng có thể được nhận biết trong môi trường hoặc sở hữu như tài sản riêng.

1. Đối tượng được mô tả tổng quát như thế nào ?
2. Phần nào trong mô tả là hữu ích cho bài toán ?
3. Vậy đặc tả tối thiểu cho đối tượng trong phạm vi bài toán đang giải quyết là gì ?

Các loại thuộc tính:

- Thuộc tính mô tả: là các facts của đối tượng được nhìn từ bên ngoài (môi trường). Vd: Jan nặng thêm 1 kg.
- Thuộc tính định danh: là các thuộc tính để phân biệt đối tượng; một đối tượng có nhiều tên gọi và khi tên gọi thay đổi, đối tượng vẫn tồn tại như trước

Xác định thuộc tính của đối tượng

1. Thuộc tính của đối tượng phải phù hợp với ý nghĩa của đối tượng trong ngữ cảnh mà đối tượng đang tồn tại. Vd: năm kinh nghiệm và tuổi của một lập trình viên.
2. Có ý nghĩa ở mọi thời điểm.
3. Không chứa cấu trúc bên trong.
4. Là đặc tính của cả thực thể chứ không chỉ của thành phần trong thực thể. VD: Computer gồm CPU, bàn phím, màn hình, con chuột. Kích thước màn hình là thuộc tính của màn hình, không phải của computer.
5. Nếu 1 đối tượng trừu tượng có tương tác với đối tượng khác, thuộc tính của nó phải thể hiện ý nghĩa của chính nó. Vd: mô hình hóa đối tượng bơm xăng: số lít được bơm ≠ lượng xăng trong bình chứa, hoặc dung tích của máy bơm.

Xác định dịch vụ của đối tượng

- **Dịch vụ**: là công việc cần thực hiện cho đối tượng khác, được kích hoạt qua cơ chế truyền thông điệp và được định nghĩa qua giao tiếp, gồm:
  1. Tên gọi của dịch vụ
  2. Thông số: là những gì đối tượng cần (nhưng chưa có sẵn) để thực hiện dịch vụ
  3. Giá trị trả về là giá trị cung cấp cho đối tượng khác
  4. Ngoại lệ: những trường hợp bị từ chối thực hiện

  Trong quá trình phân tích và thiết kế, chỉ có tên gọi của dịch vụ là cần phải cân nhắc kỹ về ý nghĩa của nó trong phạm vi của bài toán; còn các thông số, giá trị trả về và các ngoại lệ sẽ dần dần được tinh chỉnh tùy theo mức độ cài đặt về phương diện kỹ thuật.

Xác định hành vi của đối tượng

- **Hành vi**: là một tập các hành động mà đối tượng cần thực hiện để cung cấp dịch vụ, được định nghĩa trên 3 yếu tố:
  1. Đầu vào (thông số của dịch vụ)
  2. Đầu ra (kết quả trả về)
  3. Các hành động và trạng thái tương ứng

1. Hành vi tĩnh (static behavior)

   Hành vi của đối tượng không bị ảnh hưởng (không thay đổi tính chất xử lý) bởi trạng thái của nó. Vd: lấy căn bậc 2.

   Hành vi tĩnh của đối tượng có thể được diễn tả bằng lược đồ hoạt động (activity diagram).

2. Hành vi động (dynamic behavior)

   Hành vi của đối tượng bị thay đổi tùy theo trạng thái của đối tượng tại thời điểm phát sinh sự kiện kích hoạt.Vd: nhấn nút ‘mod’ của đồng hồ điện tử : time – alarm – eslap – timer – setting.

   Hành vi động của đối tượng được diễn tả bằng lược đồ trạng thái (state diagram).

Xác định quan hệ của đối tượng

- Đối tượng có 3 loại quan hệ cơ bản sau

1. Tổng quát hóa (generalization), ví dụ: cha con.
2. Kết tập (aggregation, composition), ví dụ: xe hơi.
3. Kết hợp (association), ví dụ: hôn nhân.

Xác định các quan hệ giữa các đối tượng là để tìm ra sự hợp tác giữa các đối tượng dựa trên vai trò của chúng trong phạm vi của bài toán đang giải quyết.

Quan hệ tổng quát hóa (generalization)

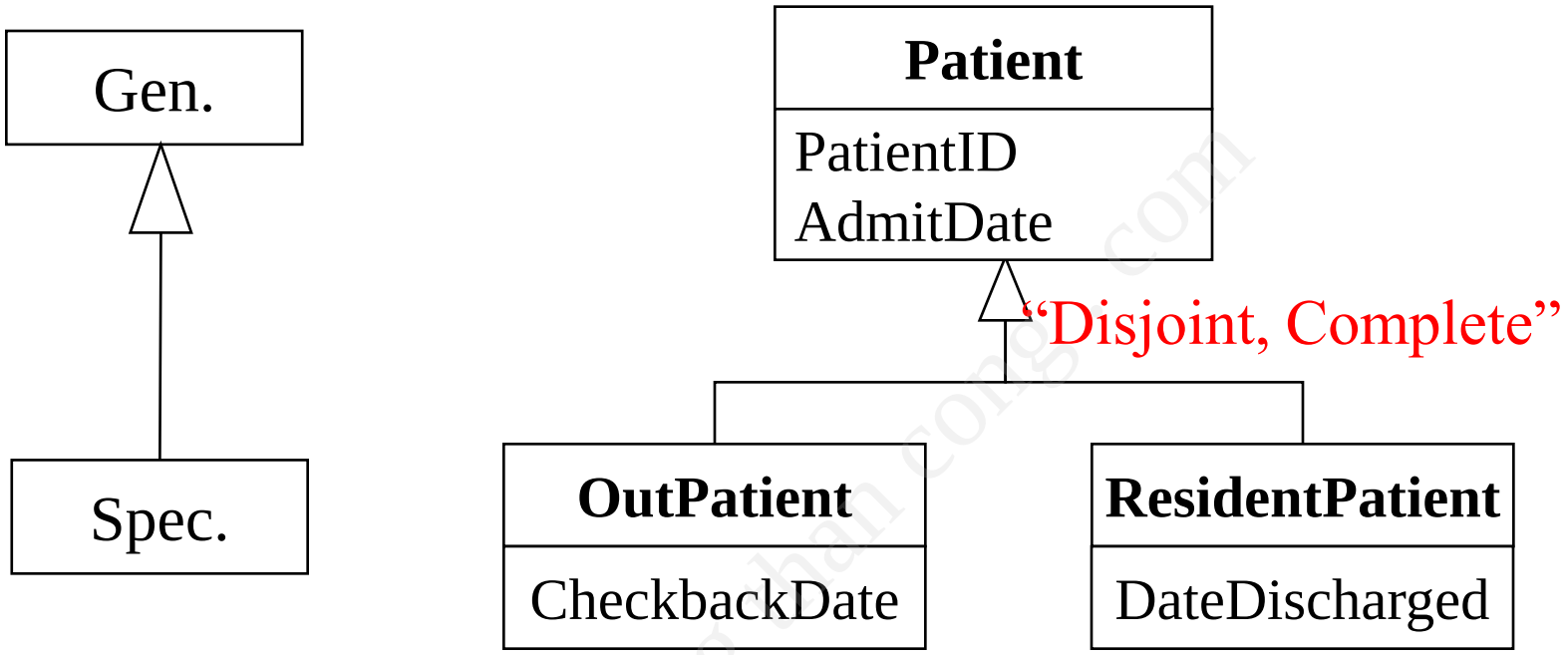
- *Quan hệ* "is_a": nếu đối tượng A có một quan hệ is_a với đối tượng B thì tất cả các thuộc tính và dịch vụ của B đều có ở A. Ví dụ: Employee "is_a" Person.
- Quan hệ tổng quát hóa là một quan hệ is_a: đối tượng B là một sự tổng quát hóa (cha) của đối tượng A (con), theo một quan điểm nào đó. Như vậy, một đối tượng có thể có nhiều cách tổng quát hóa (nhiều cha): trường hợp này là một sự đa thừa kế (multiple inheritance). Ví dụ: lập trình viên có 2 tổng quát hóa: con người, và lập trình.
- Quan hệ tổng quát hóa có thể không tồn tại vĩnh cữu. Nếu 1 nhân viên không còn làm việc nữa thì anh ta không còn thừa kế các thuộc tính của đối tượng "làm công" (lĩnh lương, thăng tiến, nghỉ phép)

Quan hệ tổng quát hóa (generalization)

Các đặc tính của quan hệ tổng quát hóa

1. Đối tượng con thừa hưởng được toàn bộ thuộc tính, quan hệ và dịch vụ của đối tượng cha.
2. Đối tượng con có thể thừa hưởng trọn vẹn hành vi của đối tượng cha, và cũng có thể thay đổi các hành vi này (polymophism).
3. Nếu B là tổng quát hóa của A, thì A không là tổng quát hóa của B (bất đối xứng)
4. Nếu A "is_a" B và B "is_a" C thì A "is_a" C (bắc cầu)

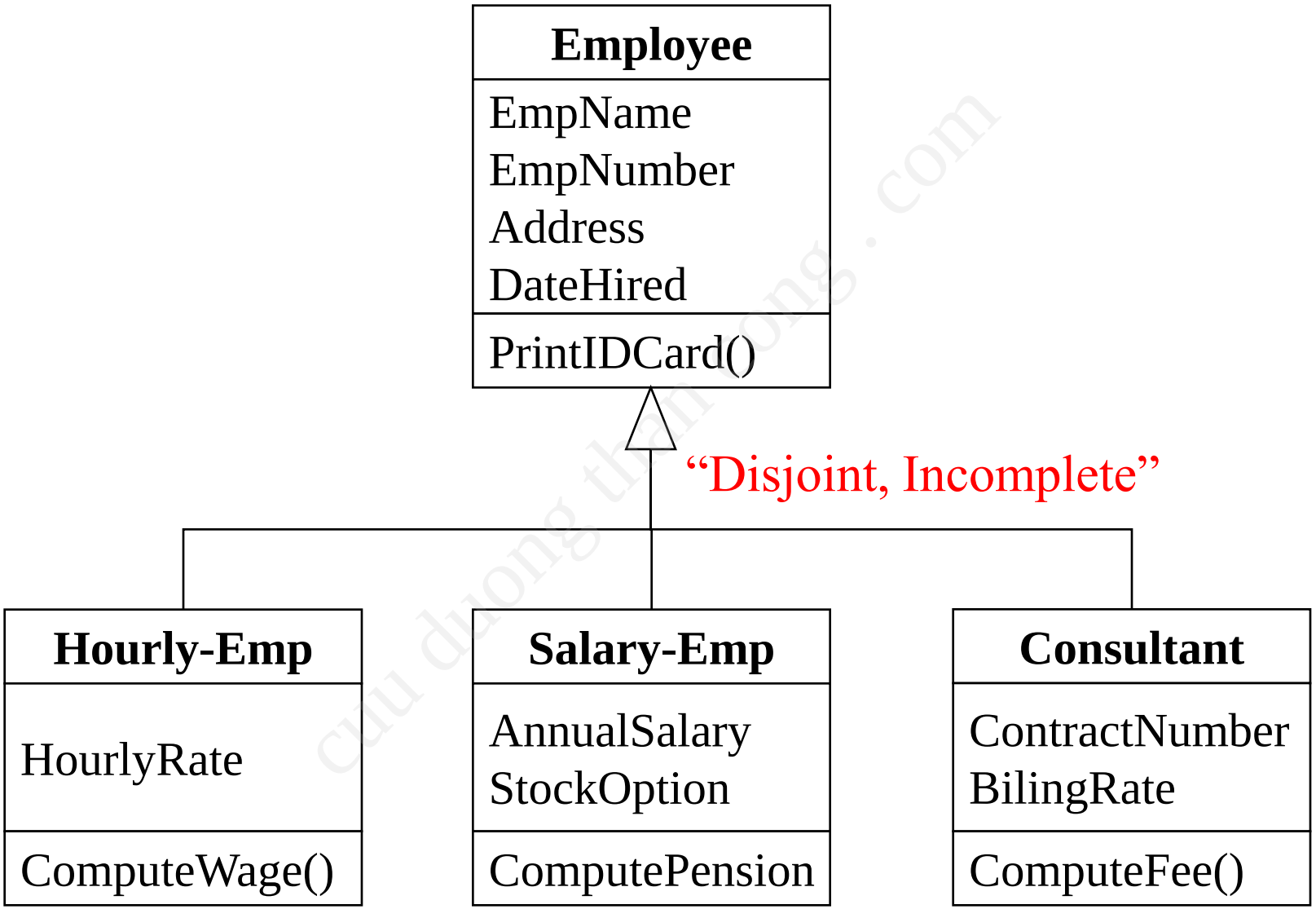
UML cho quan hệ tổng quát hóa



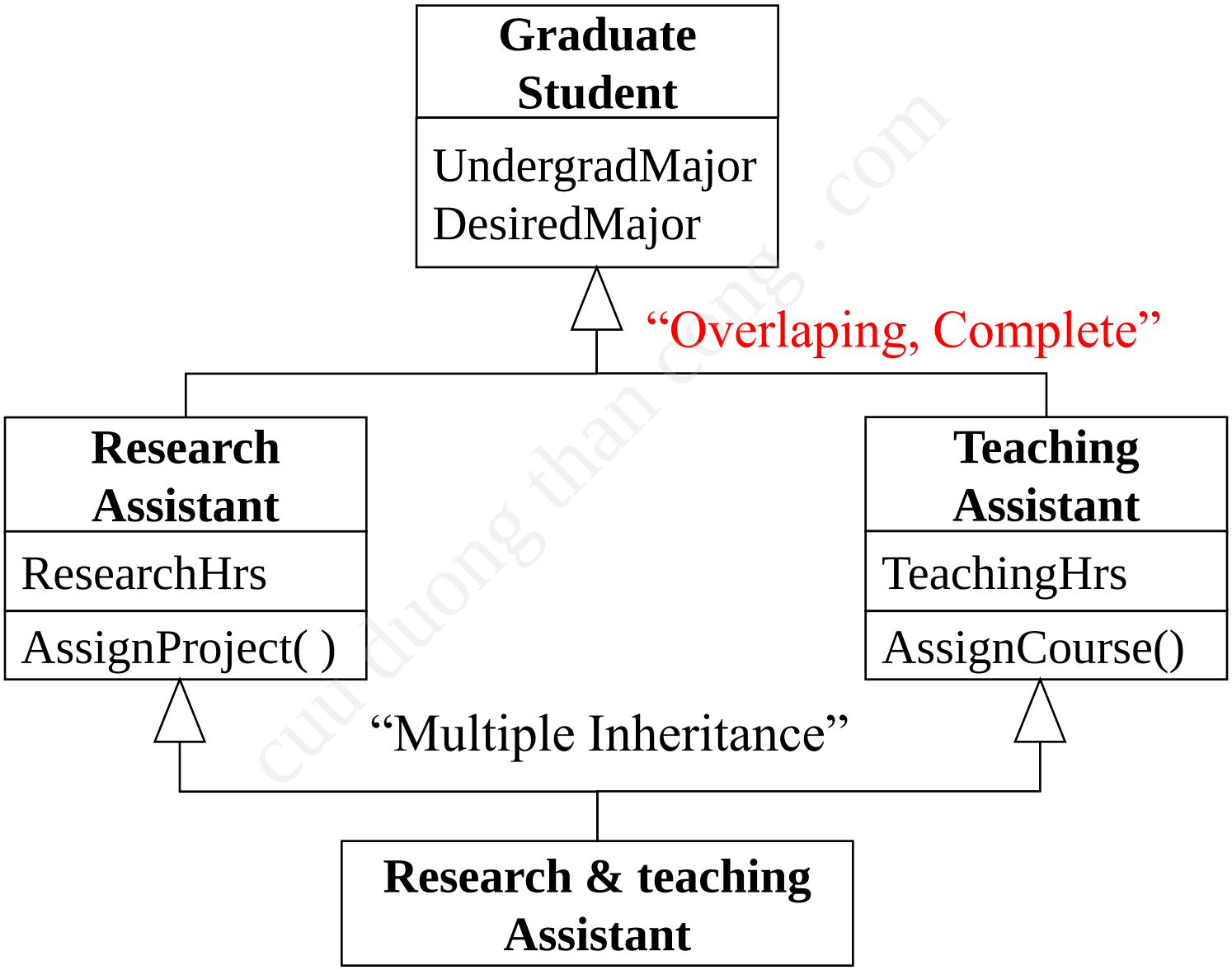
➲ Các đặc tính của lược đồ UML cho qh.tổng quát hóa:

1. ***Disjoint***: không có đối tượng nào thuộc nhiều lớp, ngược lại là Overlaping.
2. ***Complete***: Subclasses được vẽ đầy đủ trên lược đồ, ngược lại là Incomplete (vẫn còn một số subclass chưa xác định).

UML cho quan hệ tổng quát hóa



UML cho quan hệ tổng quát hóa



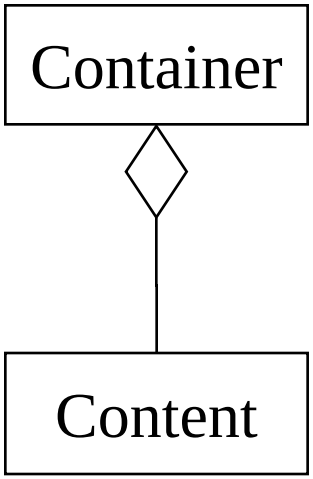
Quan hệ kết tập (aggregation)

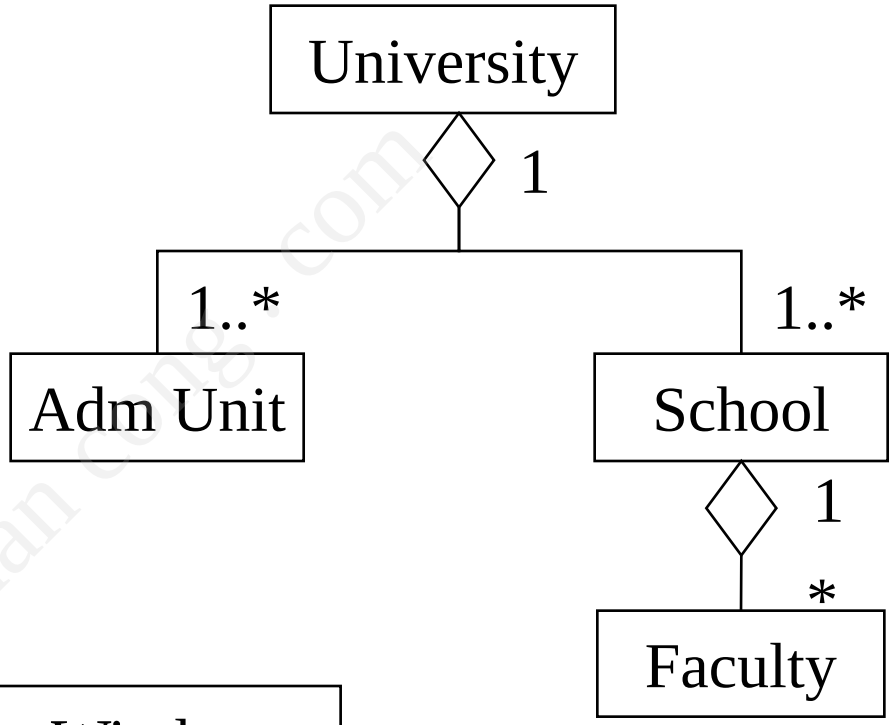
- Là quan hệ diễn tả sự liên kết nhiều đối tượng có chung một mục đích (chức năng) để "tạo thành" một đối tượng duy nhất, dựa trên một số tính chất:
  - Tích hợp các thành phần (component-integral).Vd: bánh xe, máy, khung … tạo thành xe.
  - Vật liệu làm ra vật thể. Vd: sắt, gỗ, nhựa,… làm ra ghế.
  - Nội dung chứa bên trong (container content).Vd: các mục hàng, ngày mua, nơi,… trong yêu cầu đặt hàng.

  Giống như quan hệ tổng quát hóa, kết tập có tính bất đối xứng, bắc cầu và các đối tượng đều có một vài chức năng chung; nhưng trong kết tập không có tính kế thừa.
- UML định nghĩa 2 loại kết tập: *aggregation* và *composition*.

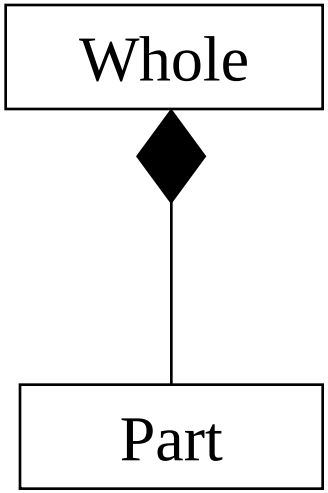
UML cho quan hệ kết tập

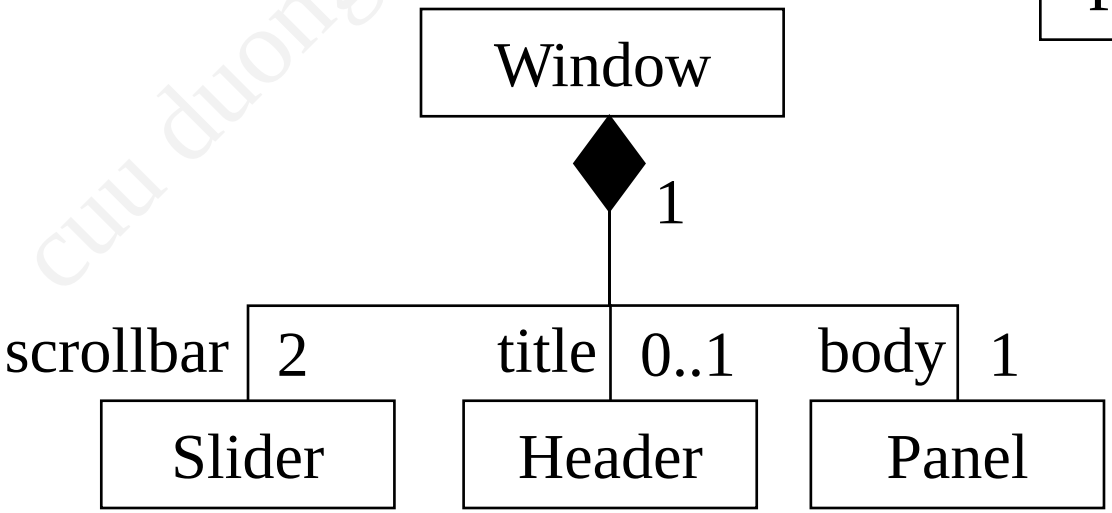
Aggregation





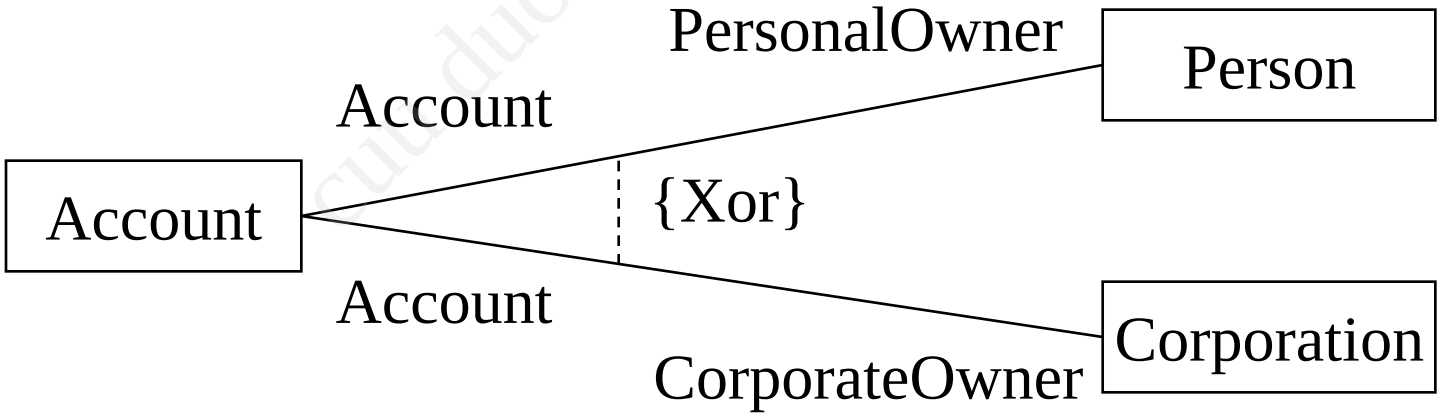
Composition





UML cho quan hệ kết hợp (association)

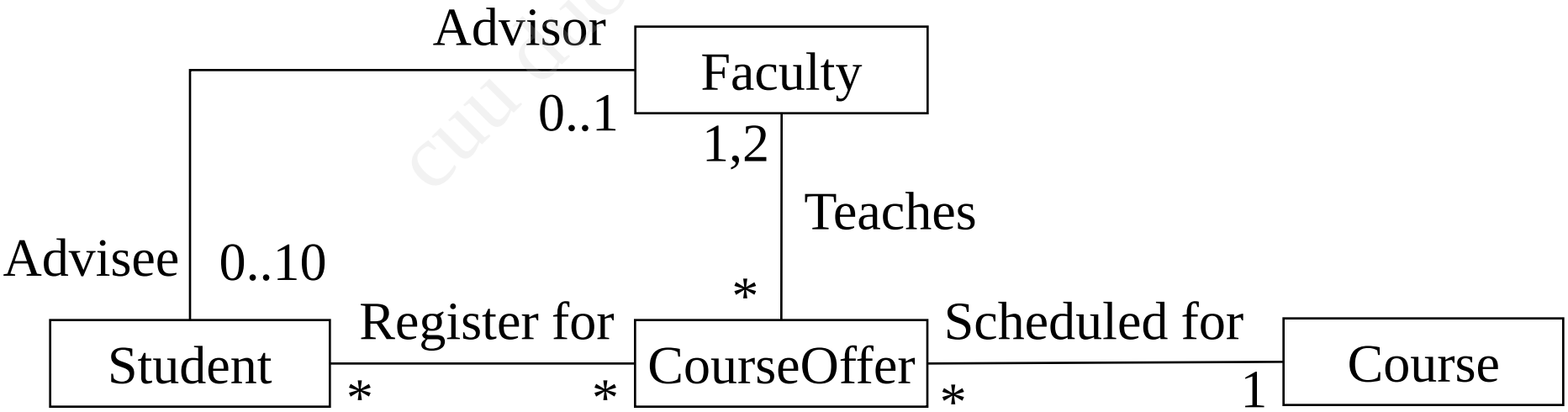
Ví dụ:



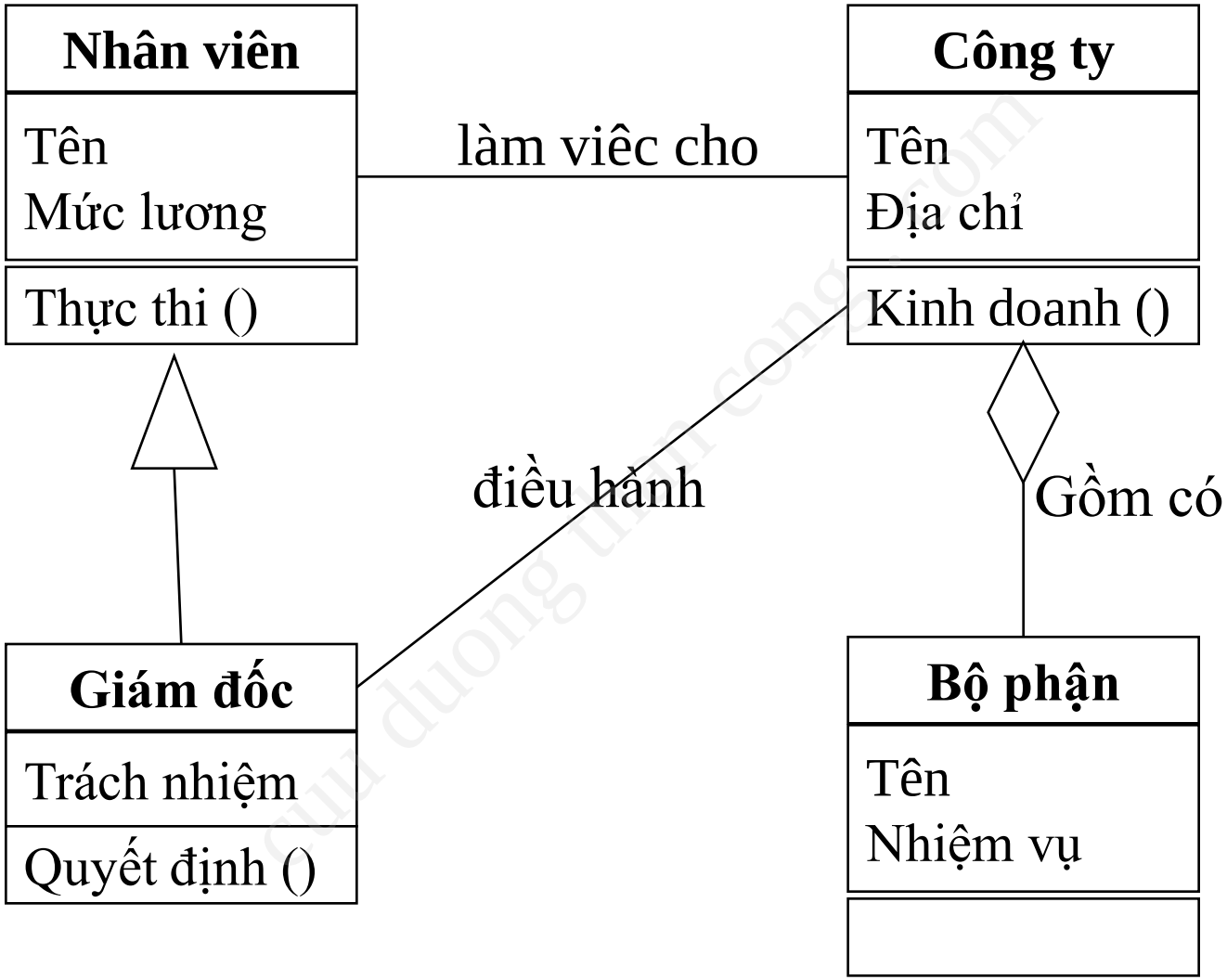
UML cho quan hệ kết hợp

Customer Order của nhà hàng Hoosie Burger

Tổ chức dạy học ở trường đại học

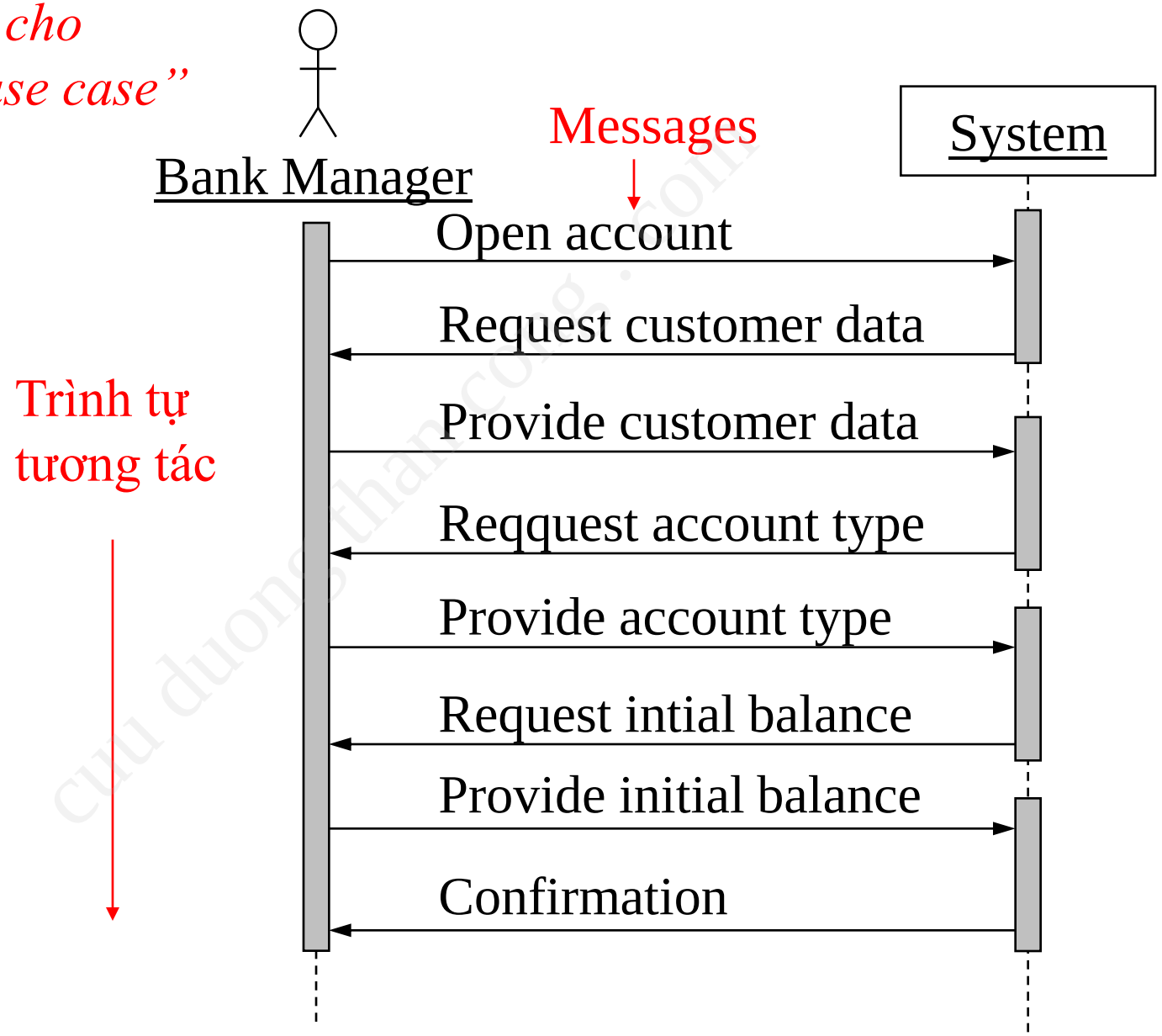


Ví dụ lược đồ class



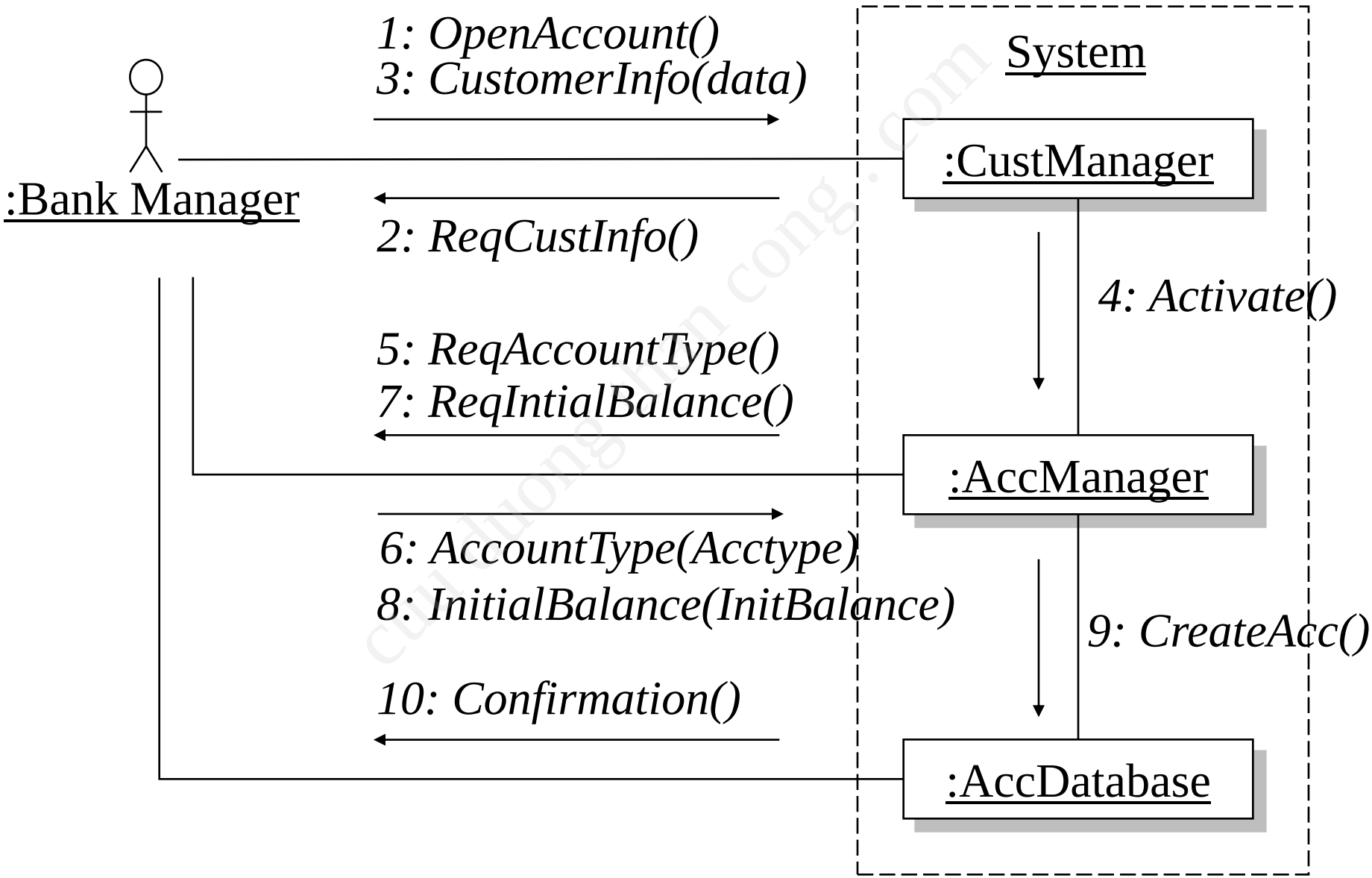
Liên kết use case với lược đồ tuần tự

Lược đồ tuần tự cho "open account use case"



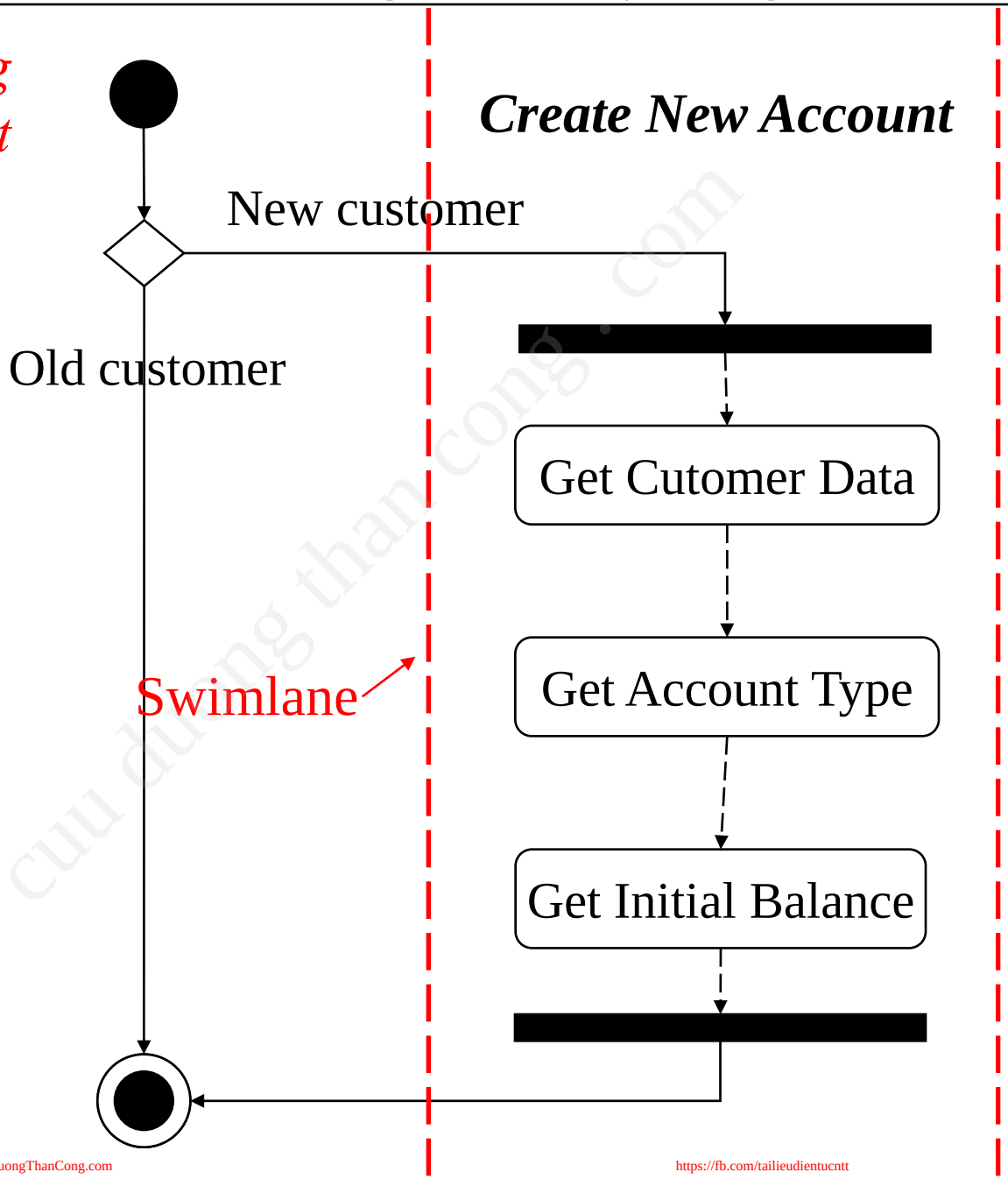
Liên kết use case với lược đồ cộng tác

Lược đồ cộng tác cho "open account use case"



Lược đồ hoạt động (activity diagram)

Lược đồ hoạt động cho "open account use case"

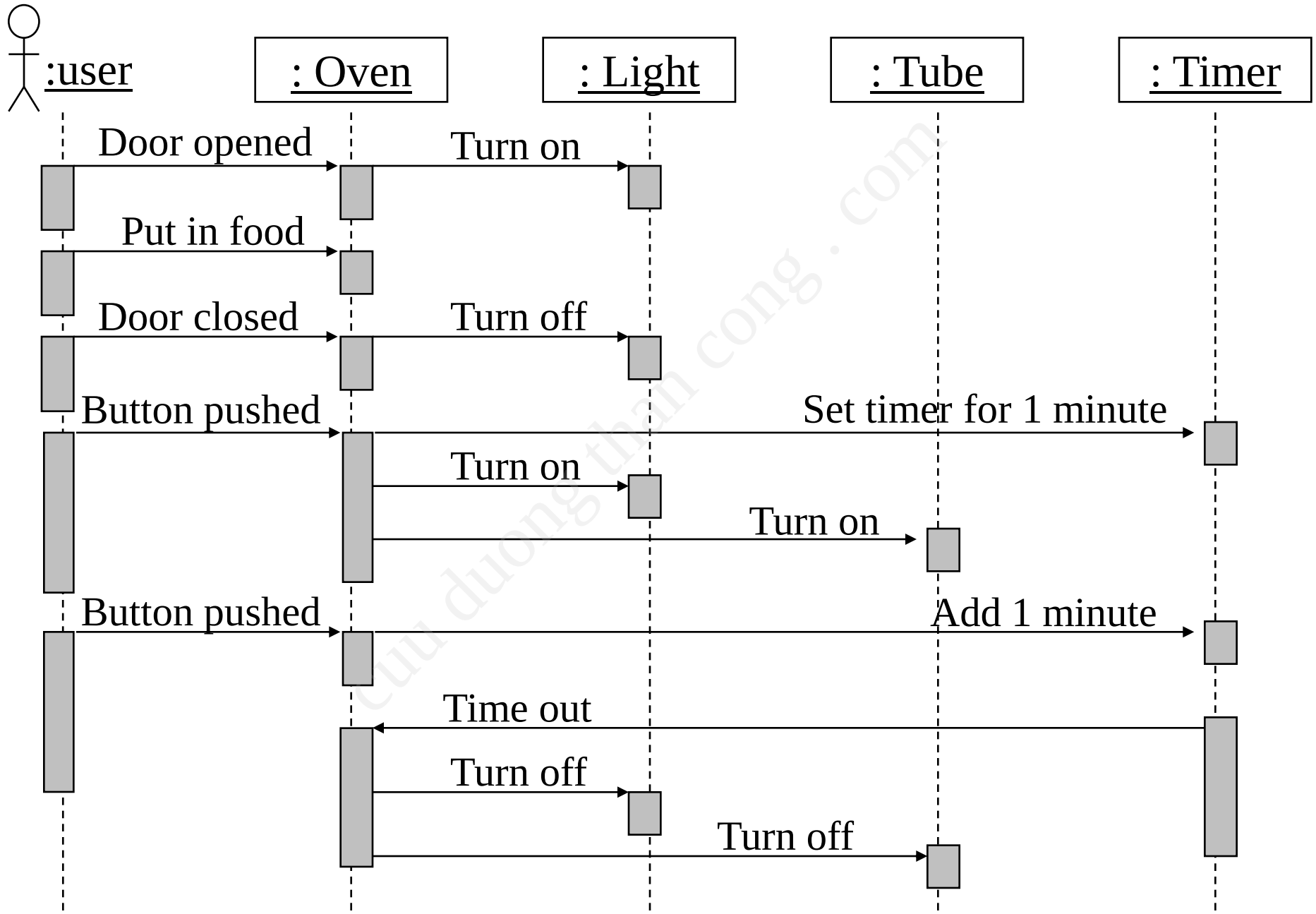


Mô hình trạng thái

- Một mô hình trạng thái diễn tả mối quan hệ giữa các trạng thái, các sự kiện kích hoạt đối tượng chuyển trạng thái, và chuyển trạng thái của đối tượng.

1. Trạng thái: là giá trị của các thuộc tính mô tả đối tượng trong từng khoảng thời gian xác định (trước hoặc sau khi phản ứng với sự kiện).
2. Sự kiện: là điều kiện kích hoạt hành vi của đối tượng.
3. Chuyển trạng thái: là sự thay đổi trạng thái của đối tượng do phản ứng với sự kiện kích hoạt.
4. Hành động: là hoạt động bên trong của đối tượng gây ra sự thay đổi trạng thái.

Ví dụ: lđ tuần tự của lò viba



Lược đồ trạng thái của lò viba