

Giai đoạn 2 Phân tích hệ thống

Chương 5 Mô hình hóa đối tượng

Nội dung

- ❖ Giải thích phân tích hướng đối tượng có thể được dùng để mô tả một HTTT như thế nào.
- ❖ Định nghĩa các khái niệm và các thuật ngữ của mô hình hóa đối tượng, bao gồm đối tượng, thuộc tính, phương thức, thông điệp, lớp và thể hiện.
- ❖ Giải thích các mối quan hệ giữa các đối tượng: phụ thuộc, liên kết, bao gộp và thừa kế.

Nội dung

- ❖ **Vẽ sơ đồ liên kết đối tượng.**
- ❖ **Mô tả các kỹ thuật và công cụ ngôn ngữ mô hình hóa hợp nhất (UML): *use case*, sơ đồ *use case*, sơ đồ lớp, sơ đồ tuần tự, sơ đồ chuyển trạng thái và sơ đồ hoạt động.**
- ❖ **Giải thích các ưu điểm của việc sử dụng các công cụ *CASE* trong phát triển mô hình đối tượng.**
- ❖ **Giải thích cách thiết lập mô hình đối tượng như thế nào.**

Các khái niệm hướng đối tượng

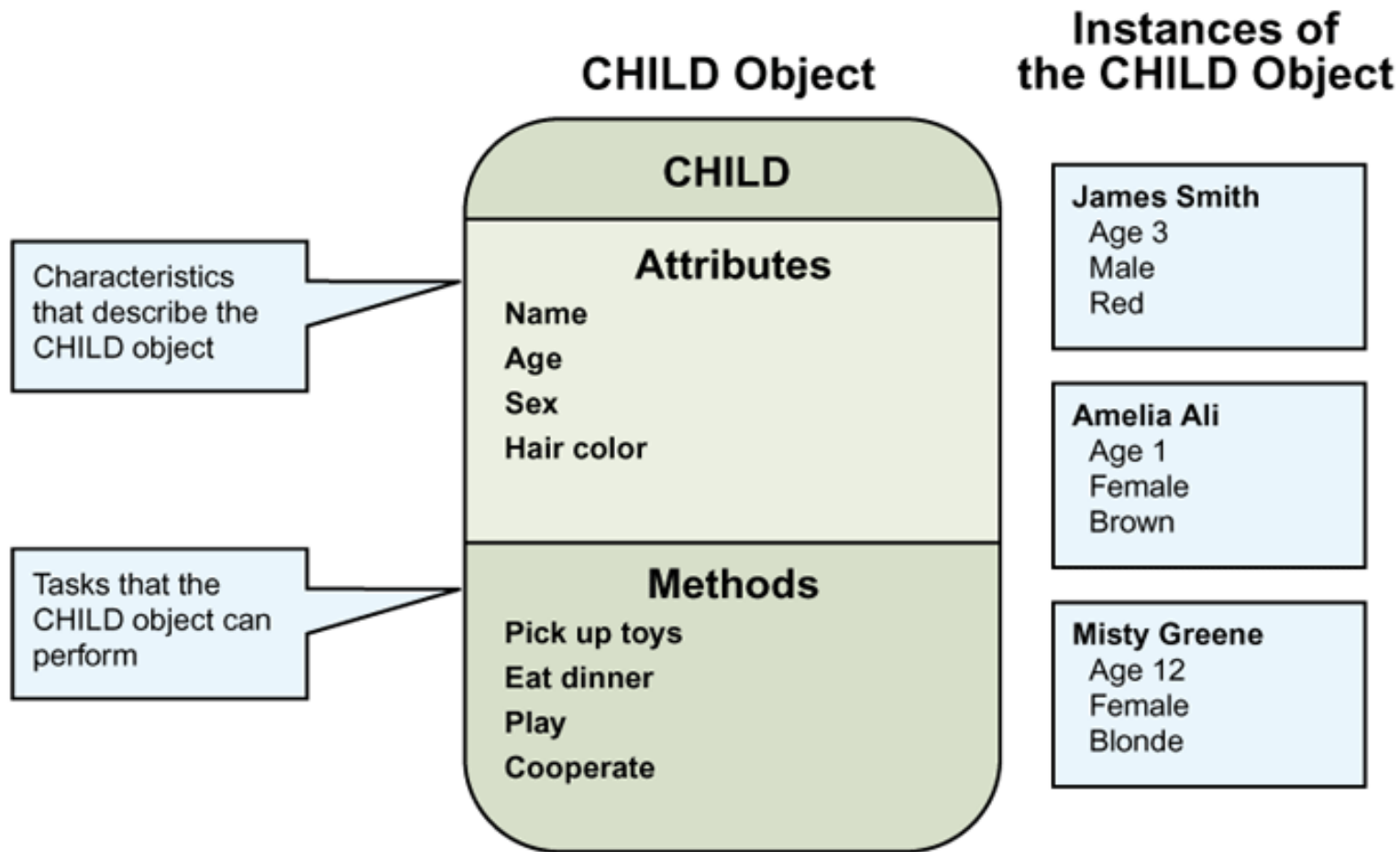
- ❖ Phân tích hướng đối tượng là một cách tiếp cận nhìn hệ thống trên quan điểm các đối tượng.
- ❖ Sản phẩm cuối cùng là mô hình đối tượng.
- ❖ Tổng quan
 - ▶ Sử dụng *Ngôn ngữ mô hình hóa hợp nhất (UML)* để phát triển các mô hình đối tượng.
 - ▶ Các đối tượng bao gồm dữ liệu và các quá trình ảnh hưởng đến dữ liệu này.
 - ▶ Lớp là một nhóm các đối tượng giống nhau.

Các khái niệm hướng đối tượng

❖ Đối tượng

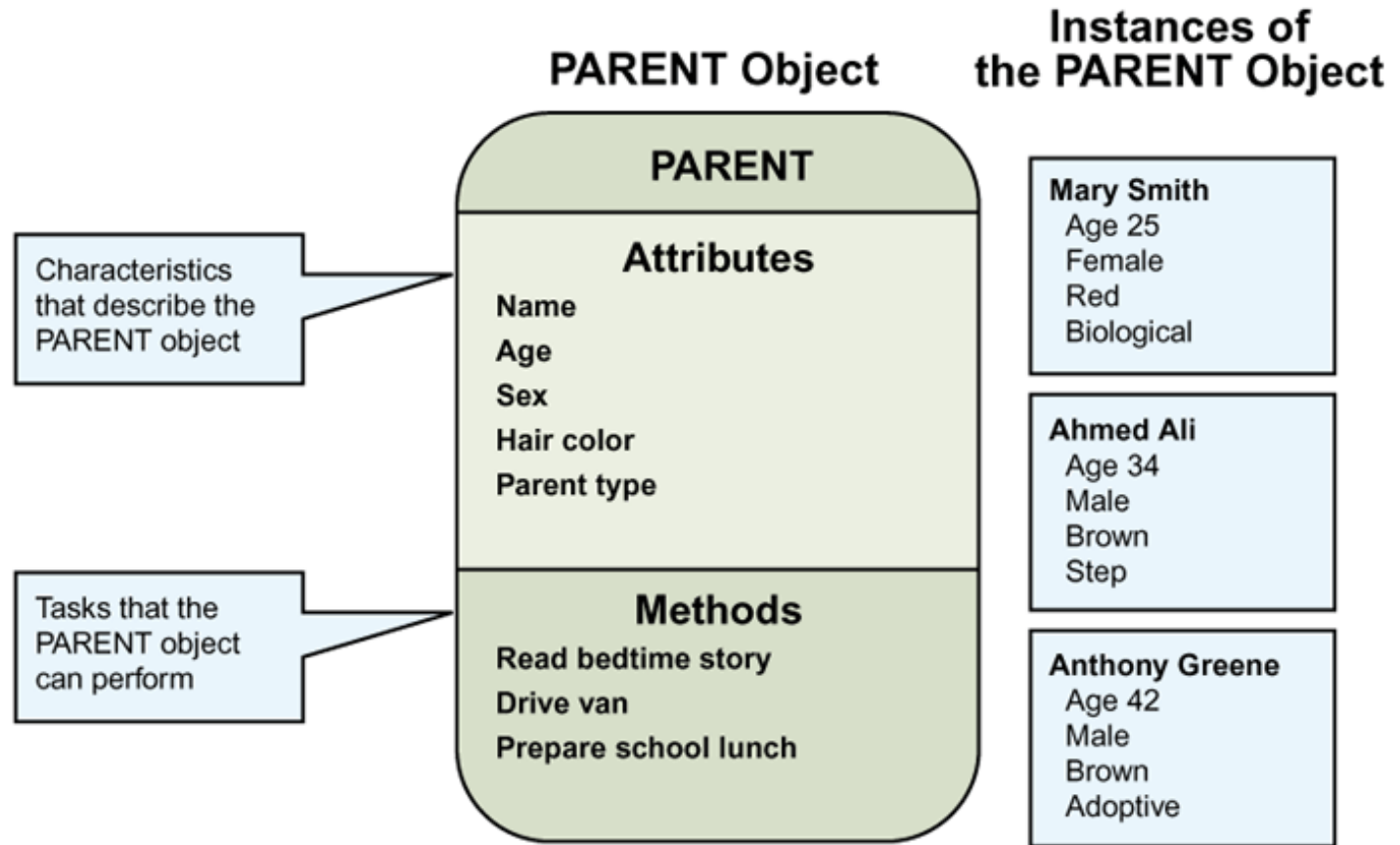
- ▶ *object*
- ▶ Một đối tượng có thể biểu diễn một người, vị trí, sự kiện hoặc giao dịch.
- ▶ *UML* biểu diễn một đối tượng bằng hình chữ nhật có tên đối tượng ở trên, theo sau là các thuộc tính và các phương thức của đối tượng.

Các khái niệm hướng đối tượng



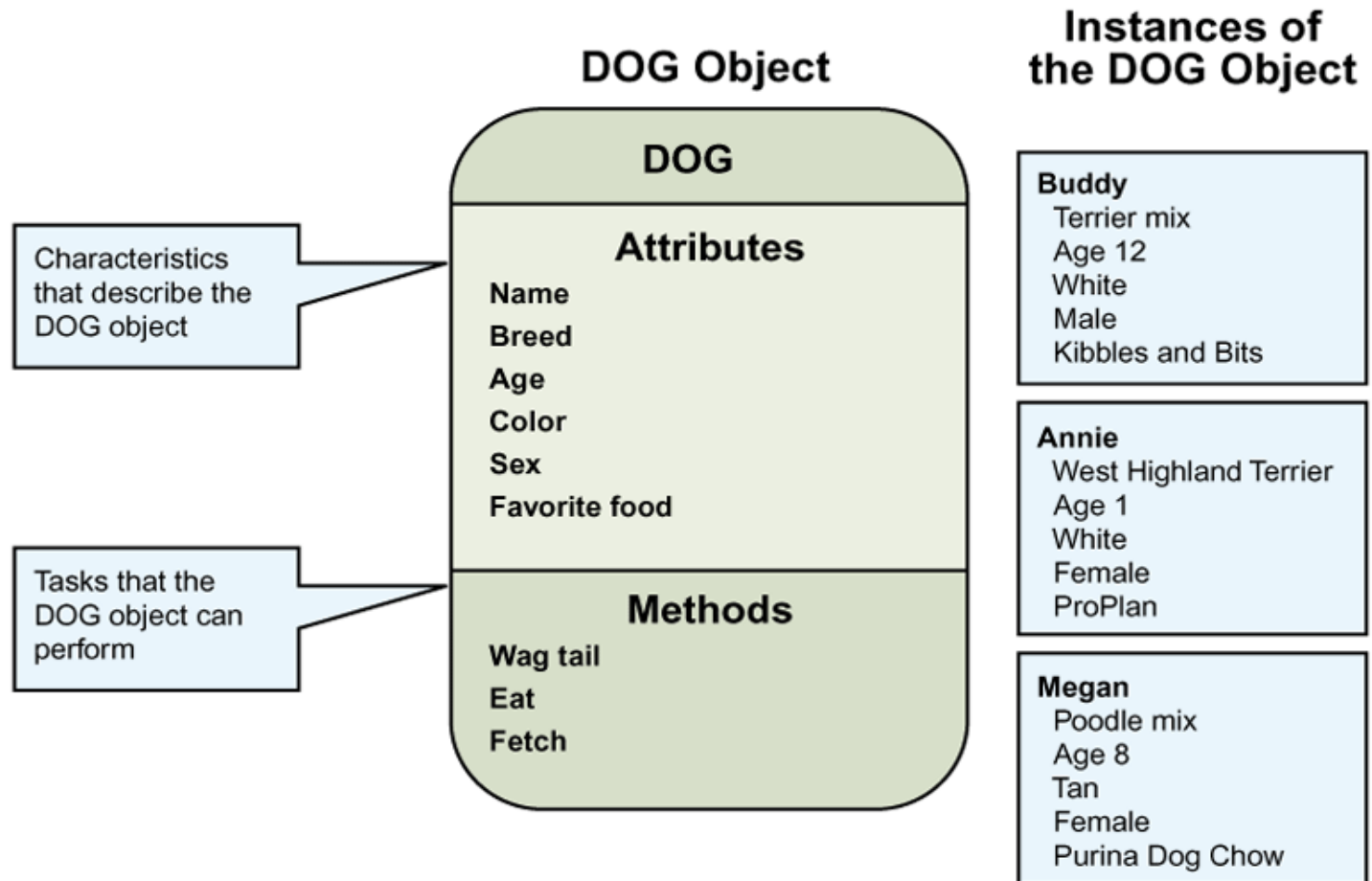
Hình 5.1. Đối tượng *Child* có 4 thuộc tính và 4 phương thức. *James Smith*, *Amelia Ali* và *Misty Greene* là các thể hiện của đối tượng *Child*.

Các khái niệm hướng đối tượng



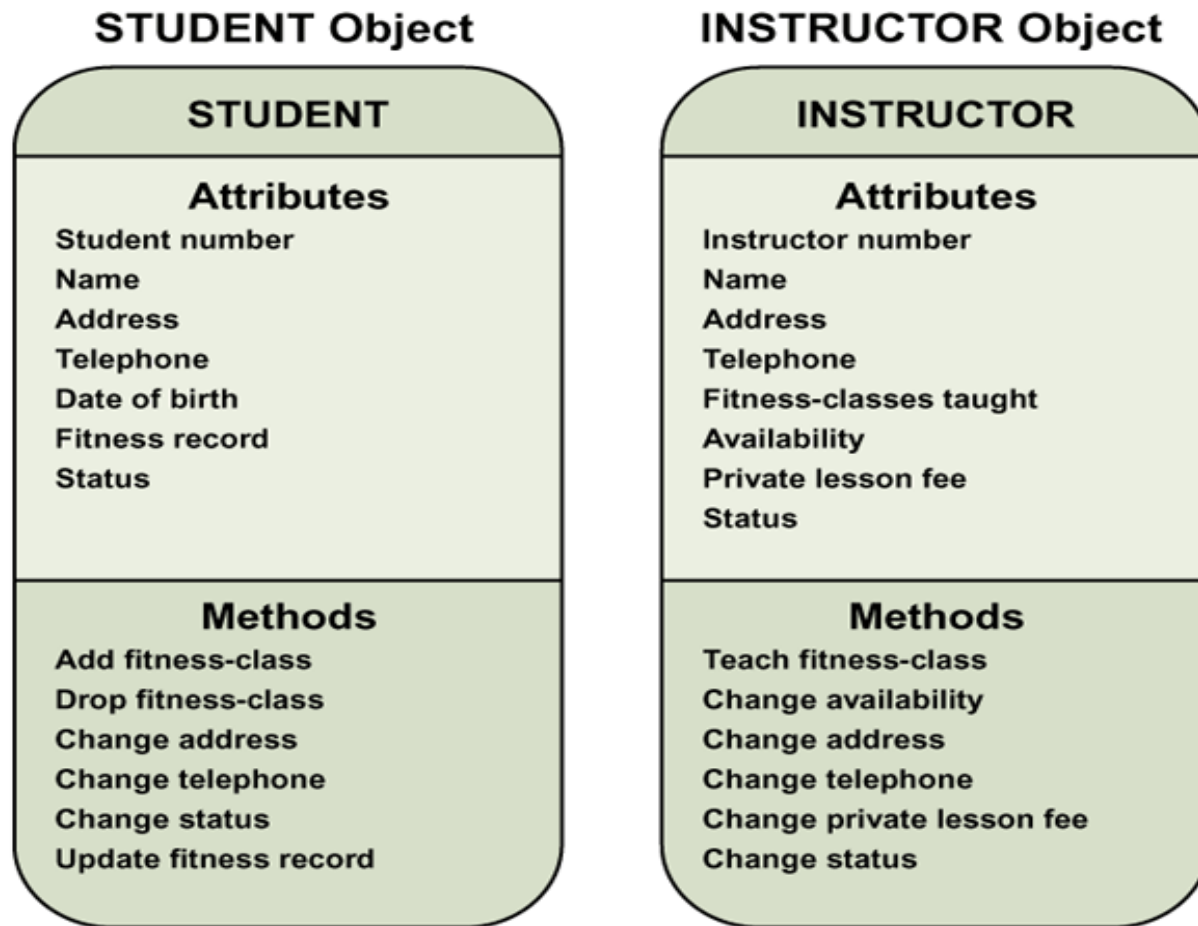
Hình 5.2. Đối tượng *Parent* có 5 thuộc tính và 3 phương thức. *Mary Smith*, *Ahmed Ali* và *Anthony Greene* là các thể hiện của đối tượng *Parent*.

Các khái niệm hướng đối tượng



Hình 5.3. Đối tượng *Dog* có 6 thuộc tính và 3 phương thức. *Budy*, *Annie* và *Megan* là các thể hiện của đối tượng *Dog*.

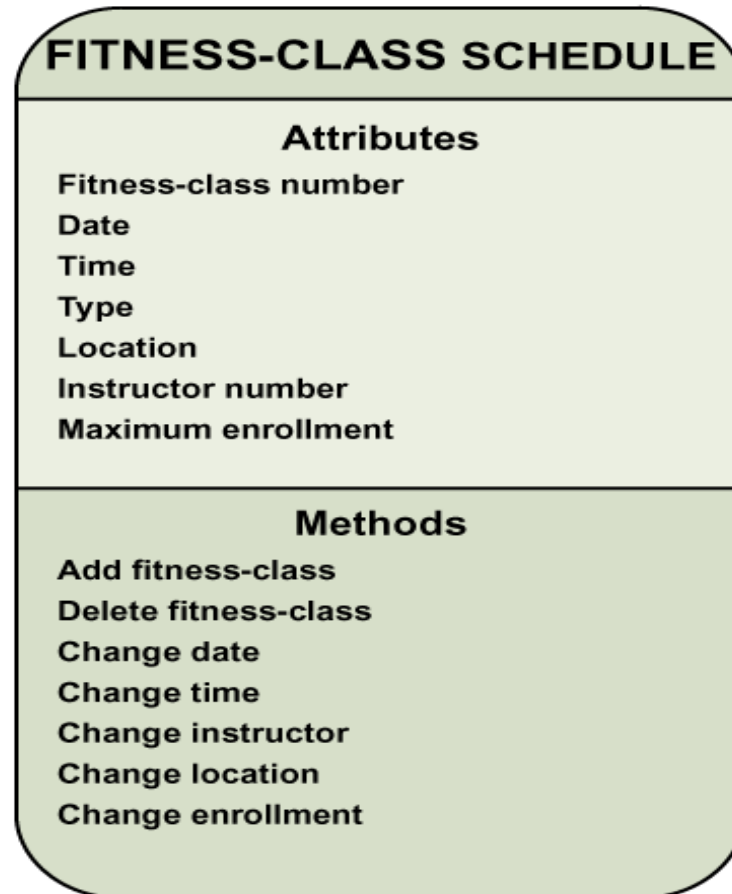
Các khái niệm hướng đối tượng



Hình 5.4. Đối tượng *Student* có 7 thuộc tính và 6 phương thức. Đối tượng *Instructor* có 8 thuộc tính và 6 phương thức.

Các khái niệm hướng đối tượng

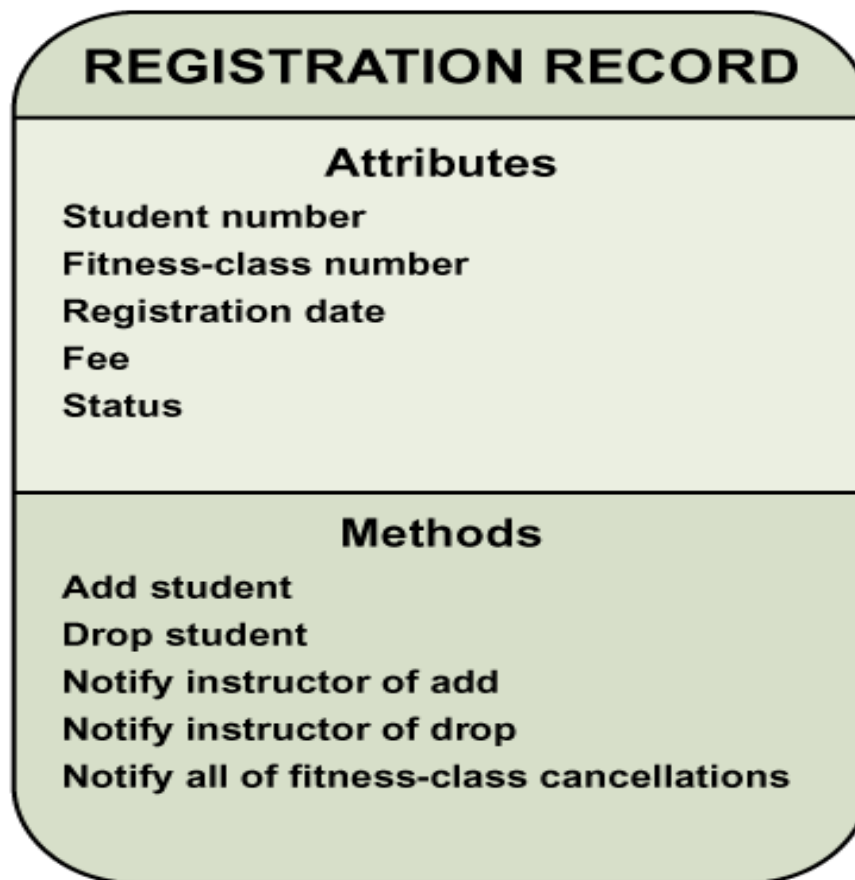
FITNESS-CLASS SCHEDULE Object



Hình 5.5. Đối tượng *Fitness-Class Schedule* có 7 thuộc tính và 7 phương thức.

Các khái niệm hướng đối tượng

REGISTRATION RECORD Object



Hình 5.6. Đối tượng *Registration Record* có 5 thuộc tính và 5 phương thức.

Các khái niệm hướng đối tượng

❖ Thuộc tính

- ▶ *attribute*
- ▶ Thuộc tính là đặc điểm dùng để mô tả đối tượng.
- ▶ Số lượng các thuộc tính cần thiết phụ thuộc vào các yêu cầu nghiệp vụ và người sử dụng của HTTT.
- ▶ Các thuộc tính được thiết kế trong quá trình thiết kế hệ thống.
- ▶ Các đối tượng có thể thừa kế hoặc có được các thuộc tính từ các đối tượng khác.
- ▶ Trạng thái của một đối tượng là một tính từ dùng để mô tả trạng thái hiện tại của đối tượng.

Các khái niệm hướng đối tượng

STUDENT Object

State	Status
Future	Registered, but has not started to attend
Current	Registered, attending one or more fitness-classes
Past	Attended one or more fitness-classes in the past

FITNESS-CLASS Object

State	Status
Open	Fitness-class is open for enrollment
Closed	Fitness-class has reached maximum enrollment
Canceled	Fitness-class has been canceled

BANK ACCOUNT Object

State	Status
Active	Account is open and meets standards for activity
Inactive	Account is dormant for specified period of time
Closed	Account was closed
Frozen	Assets have been legally attached

Hình 5.7. Trạng thái của đối tượng là một thuộc tính cho biết trạng thái của đối tượng.

Các khái niệm hướng đối tượng

❖ Phương thức

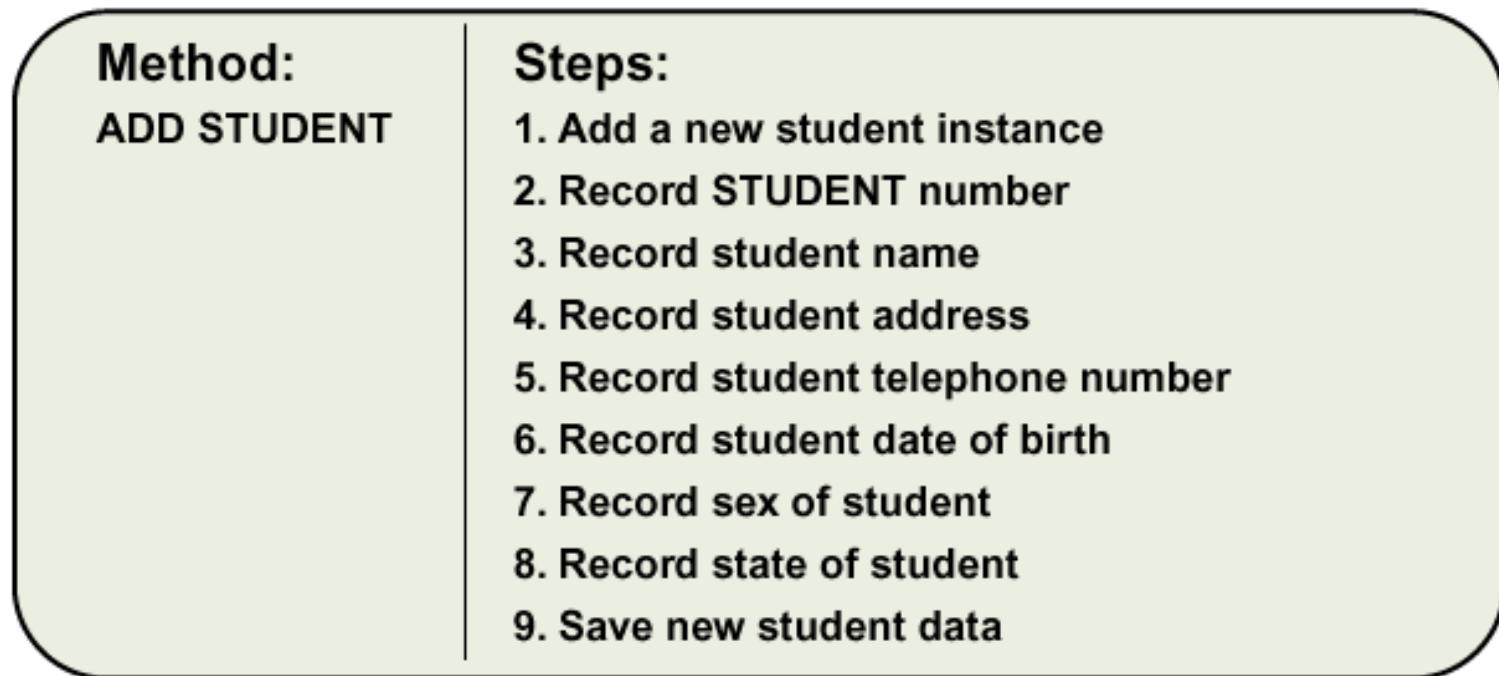
- ▶ *method*
- ▶ Một phương thức xác định các công việc cụ thể mà đối tượng có thể thực hiện.
- ▶ Phương thức mô tả một đối tượng làm **gì** và làm **như thế nào**.
- ▶ Phương thức *constructor* tạo một thể hiện mới của một đối tượng.
- ▶ Phương thức cập nhật dùng để thay đổi dữ liệu hiện tại.
- ▶ Phương thức truy vấn cung cấp thông tin về các thuộc tính của một đối tượng.

Các khái niệm hướng đối tượng

Method: MORE FRIES	Steps: <ol style="list-style-type: none">1. Heat oil2. Fill fry basket with frozen potato strips3. Lower basket into hot oil4. Check for readiness5. When ready raise basket and let drain6. Pour fries into warming tray7. Add salt
------------------------------	---

Hình 5.8. Phương thức *More Fries* yêu cầu thực hiện 7 bước cụ thể.

Các khái niệm hướng đối tượng



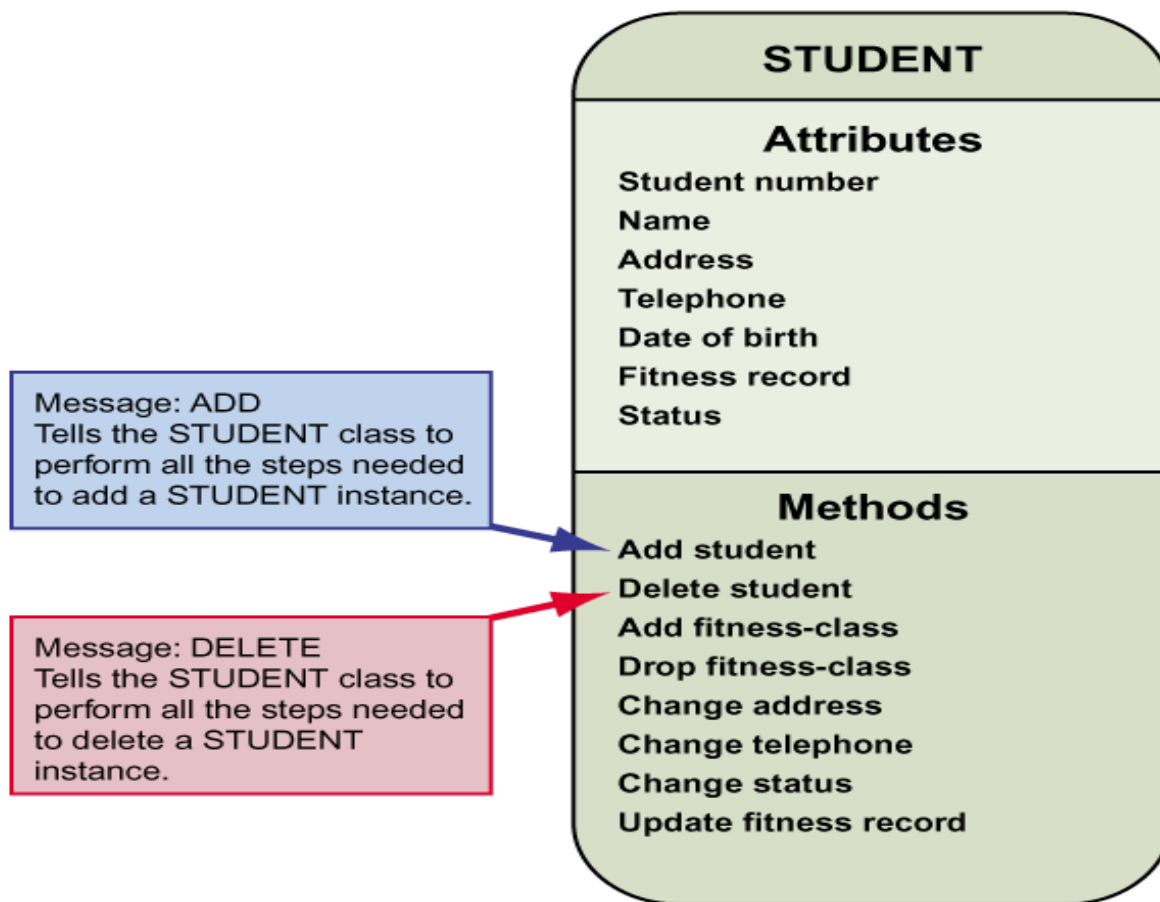
Hình 5.9. Phương thức *Add Student* yêu cầu thực hiện 9 bước cụ thể.

Các khái niệm hướng đối tượng

❖ Thông điệp

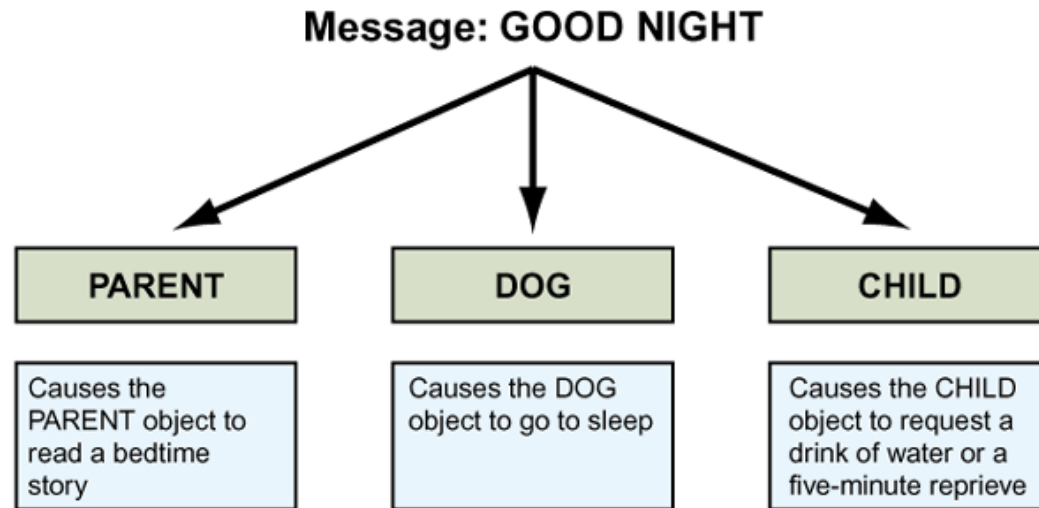
- ▶ *message*
- ▶ Thông điệp là một lệnh báo cho một đối tượng thực hiện một phương thức nào đó.
- ▶ Tính đa hình (*polymorphysm*) là khái niệm mà cùng một thông điệp gửi cho hai đối tượng khác nhau có thể tạo ra các kết quả khác nhau.
- ▶ Khái niệm hộp đen là một ví dụ về tính đóng bao – tất cả dữ liệu và các phương thức đều được đóng kín.

Các khái niệm hướng đối tượng



Hình 5.10. Thông điệp *Add* thông báo lớp *Student* thực hiện phương thức *Add Student*. Thông điệp *Delete* thông báo lớp *Student* thực hiện phương thức *Delete Student*.

Các khái niệm hướng đối tượng



Hình 5.11. Ví dụ về tính đa hình. Thông điệp *Good Night* tạo ra các kết quả khác nhau, tùy thuộc vào đối tượng nào nhận thông điệp này.



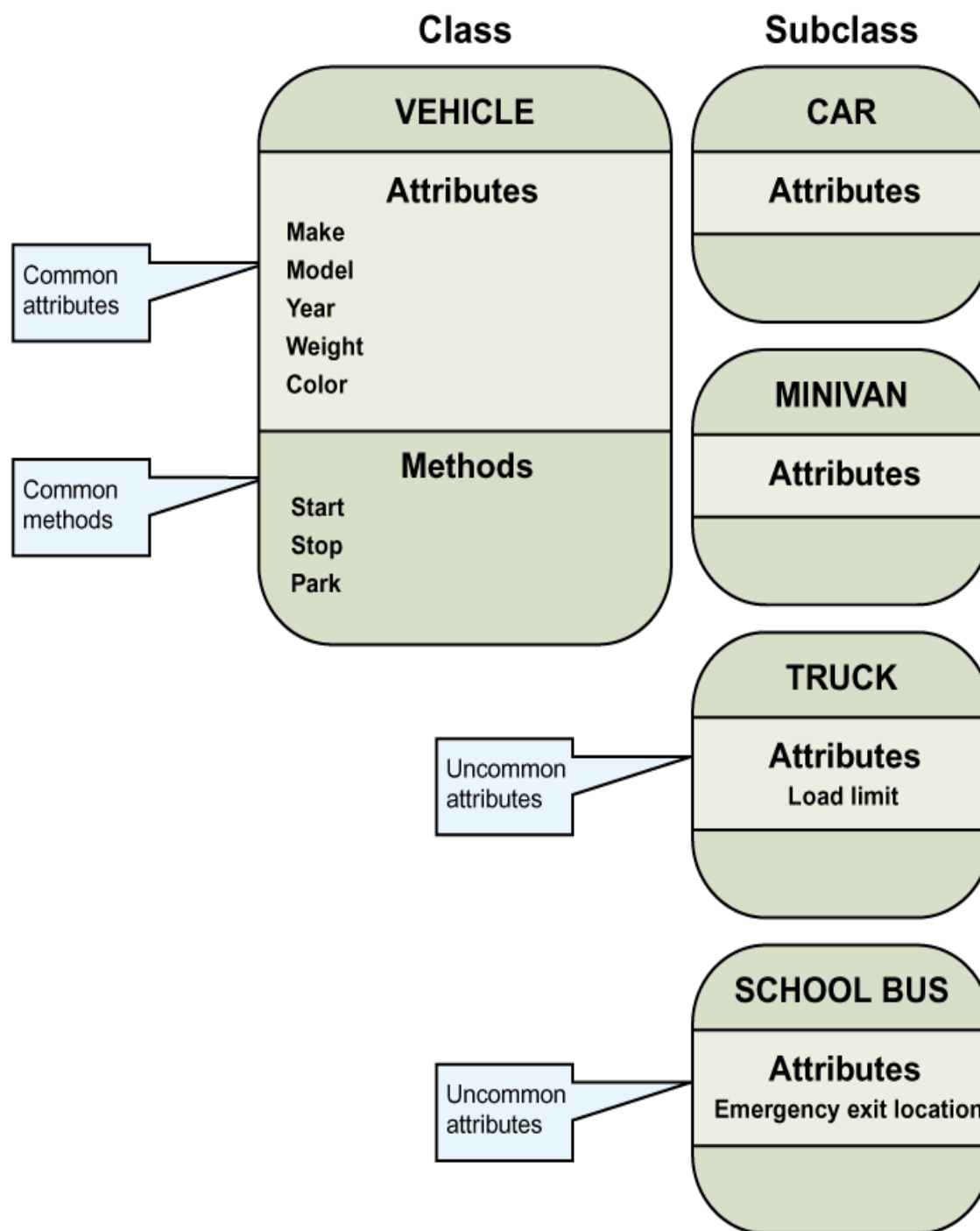
Hình 5.12. *Sales Transaction* gửi một thông điệp cho một thể hiện của lớp *Customer*.



Các khái niệm hướng đối tượng

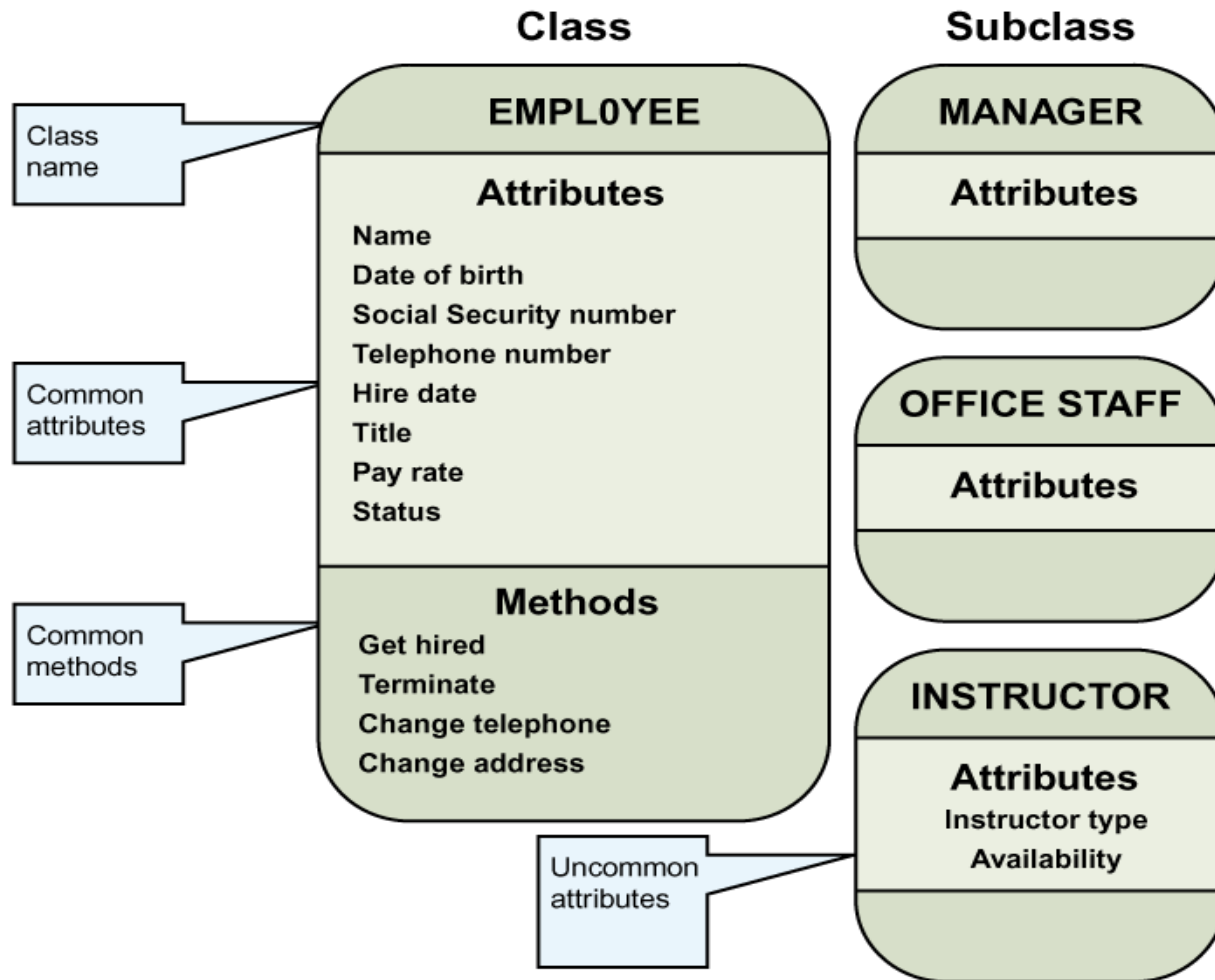
❖ Lớp

- ▶ *class*
- ▶ Một đối tượng thuộc một nhóm hoặc một loại, loại này được gọi là lớp.
- ▶ Các đối tượng trong một lớp có thể được gom nhóm thành các nhóm con.
- ▶ Một lớp có thể thuộc một loại chung hơn, loại này được gọi là lớp cha.



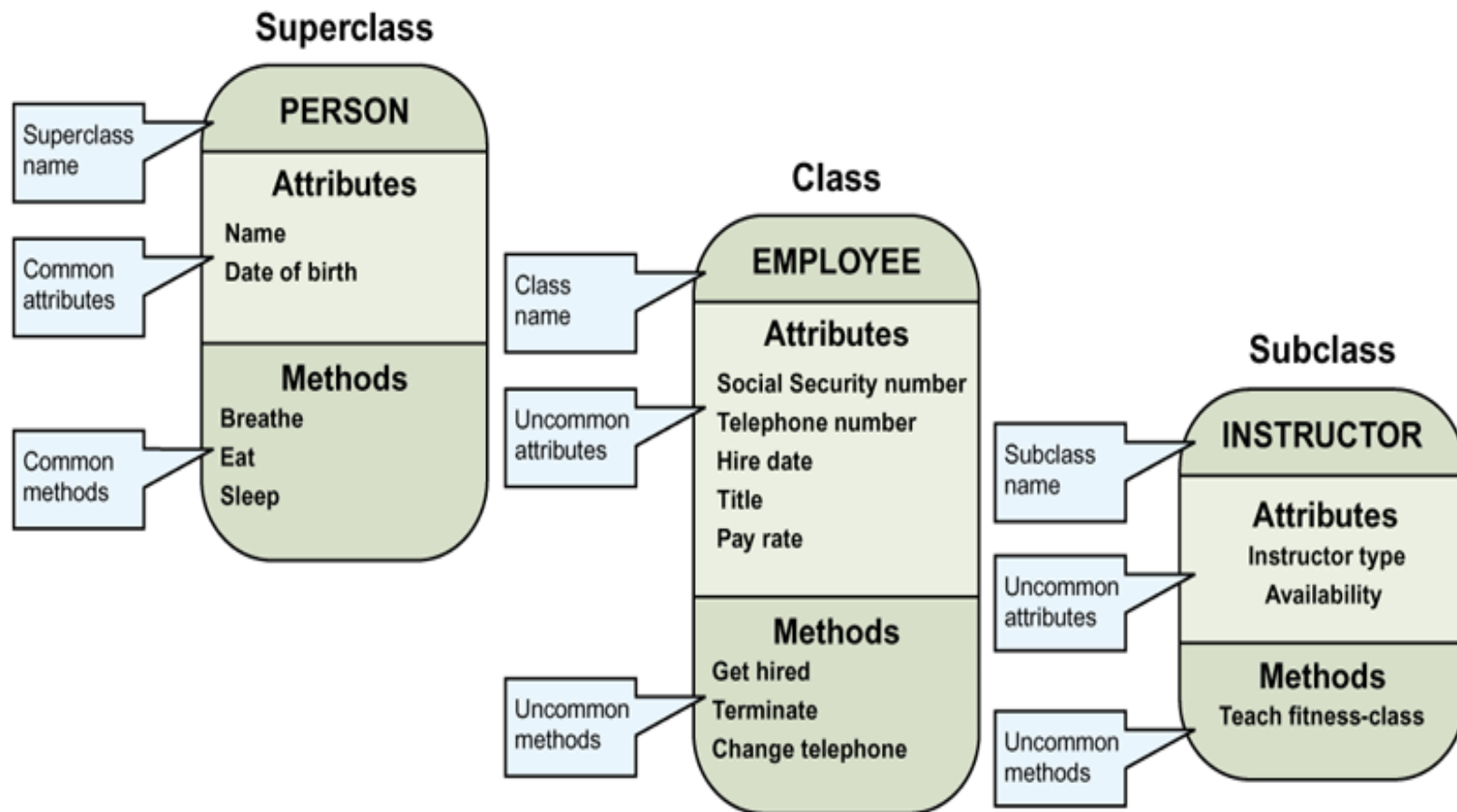
Hình 5.13. Lớp *Vehicle* chứa các thuộc tính và các phương thức chung. *Car*, *Minivan*, *Truck* và *School Bus* là các thể hiện của lớp *Vehicle*.

Các khái niệm hướng đối tượng



Hình 5.14. Lớp *Employee* chứa các thuộc tính và các phương thức chung. *Manager*, *Office Staff* và *Instructor* là các lớp con của lớp *Employee*.

Các khái niệm hướng đối tượng



Hình 5.15. Lớp cha *Person* chứa các thuộc tính và các phương thức chung. *Employee* là lớp con của lớp *Person*. *Instructor* là lớp con của lớp *Employee*.

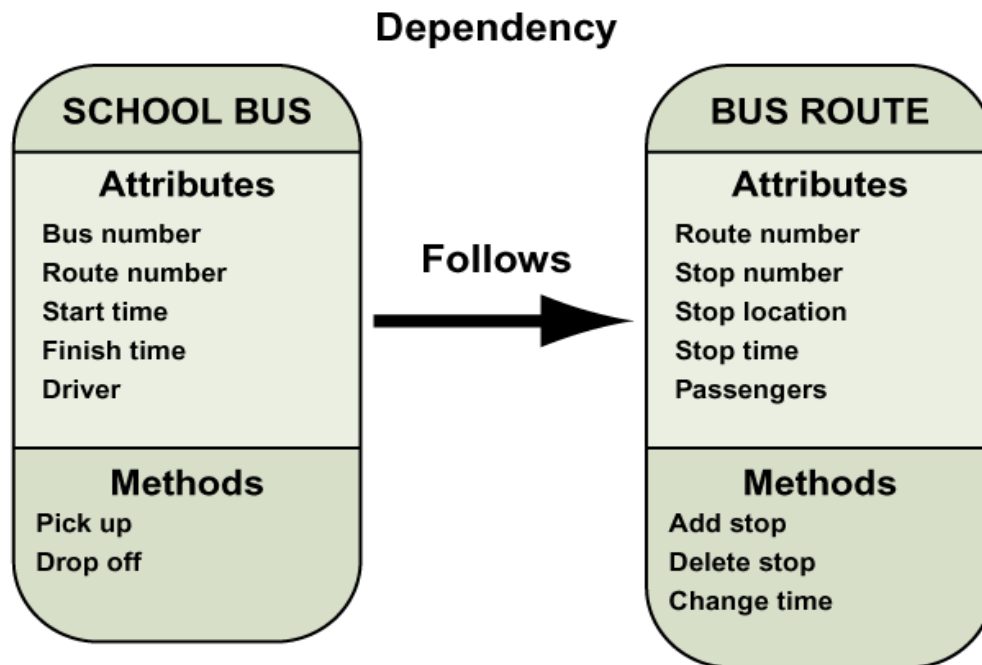
Các mối quan hệ giữa các đối tượng và các lớp

- ❖ Các mối quan hệ cho phép các đối tượng liên lạc và tương tác với nhau.
- ❖ Các mối quan hệ mô tả các đối tượng cần gì để biết lẫn nhau.
- ❖ **Các loại mối quan hệ**
 - ▶ Mối quan hệ phụ thuộc (*dependency*)
 - ▶ Mối quan hệ liên kết (*association*)
 - ▶ Mối quan hệ bao gộp (*aggregation*)
 - ▶ Mối quan hệ thừa kế (*inheritance*)

Các mối quan hệ giữa các đối tượng và các lớp

❖ Mối quan hệ phụ thuộc

- ▶ *dependency*
- ▶ Xảy ra khi một đối tượng phải được thông báo về một đối tượng khác.



Hình 5.16. Mối quan hệ phụ thuộc tồn tại giữa các đối tượng *School Bus* và *Bus Route*. Mọi thay đổi của *Route Bus* sẽ ảnh hưởng đến *School Bus*.

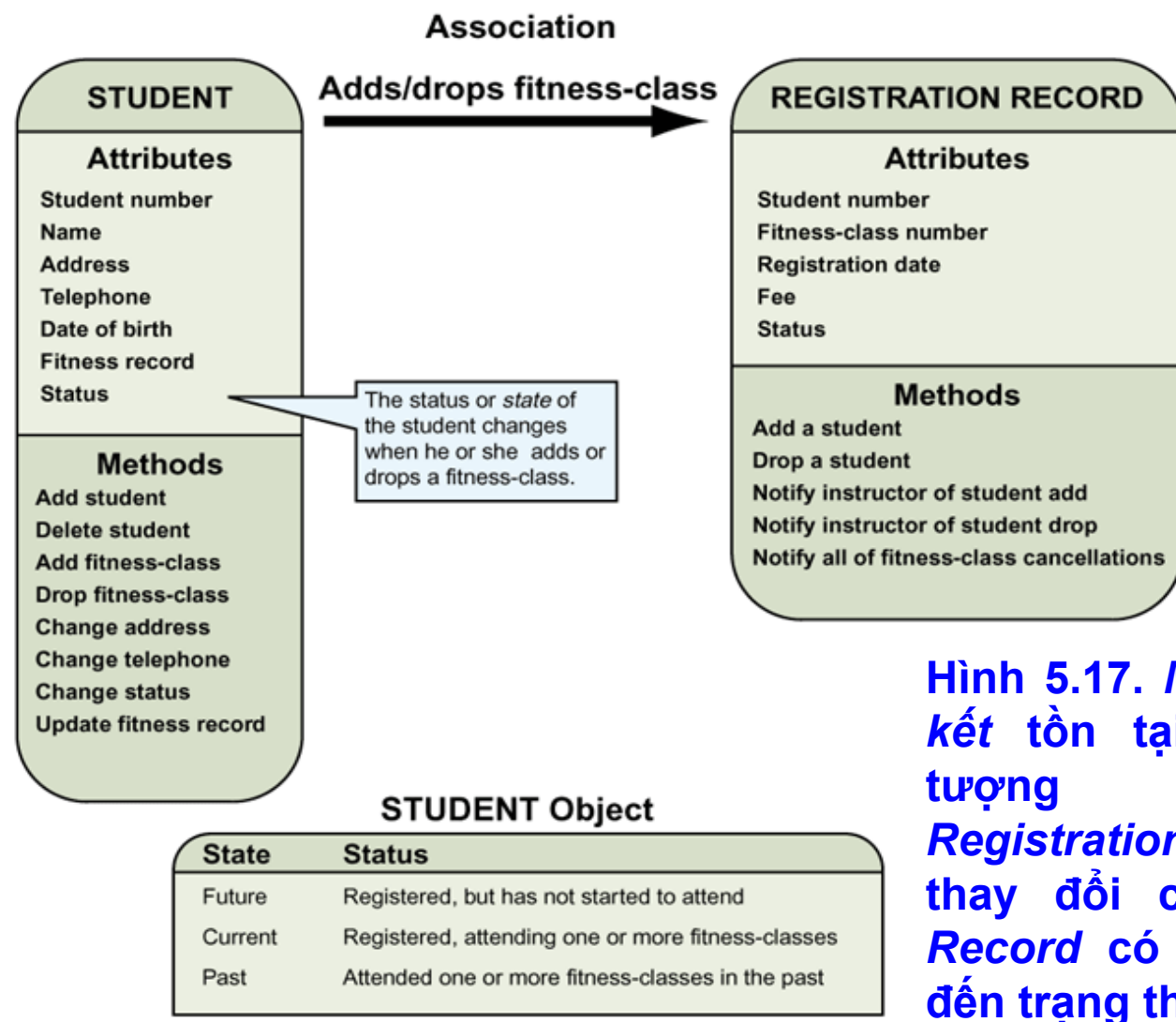


Các mối quan hệ giữa các đối tượng và các lớp

❖ Mối quan hệ liên kết

- ▶ *association*
- ▶ Mạnh hơn mối quan hệ phụ thuộc và xảy ra khi các thuộc tính nào đó của một đối tượng được xác định từ sự tương tác của nó với một đối tượng khác.

Các mối quan hệ giữa các đối tượng và các lớp



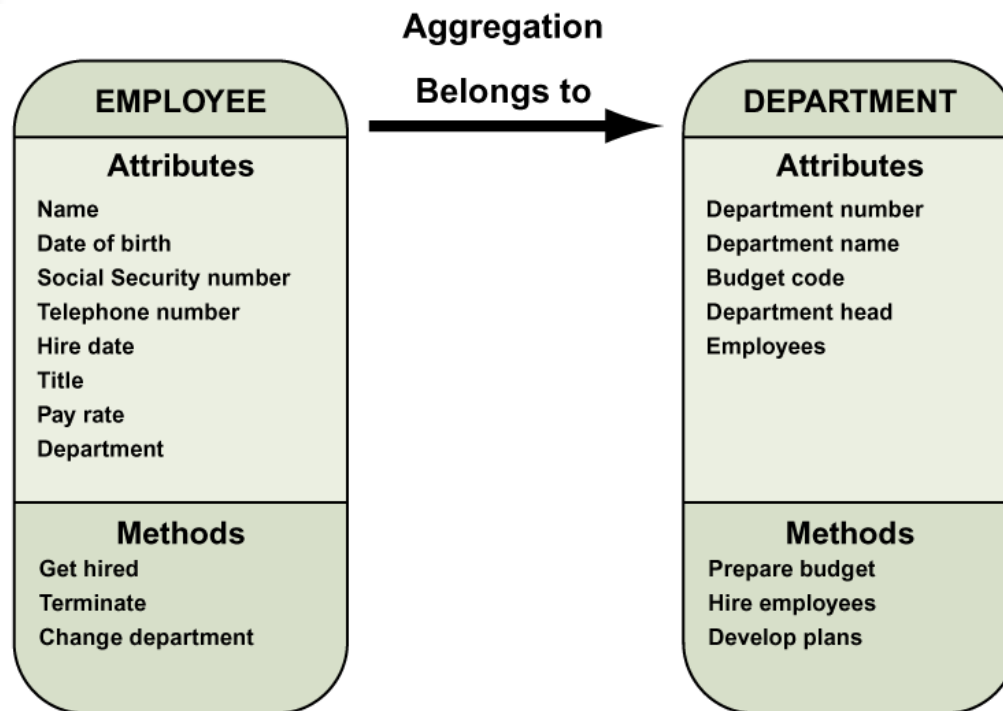
Hình 5.17. Mối quan hệ liên kết tồn tại giữa các đối tượng *Student* và *Registration Record*. Các thay đổi của *Registration Record* có thể ảnh hưởng đến trạng thái của *Student*.

Các mối quan hệ giữa các đối tượng và các lớp

❖ Mối quan hệ bao gộp

► *aggregation*

- Tồn tại khi một đối tượng là một bộ phận của một đối tượng khác.



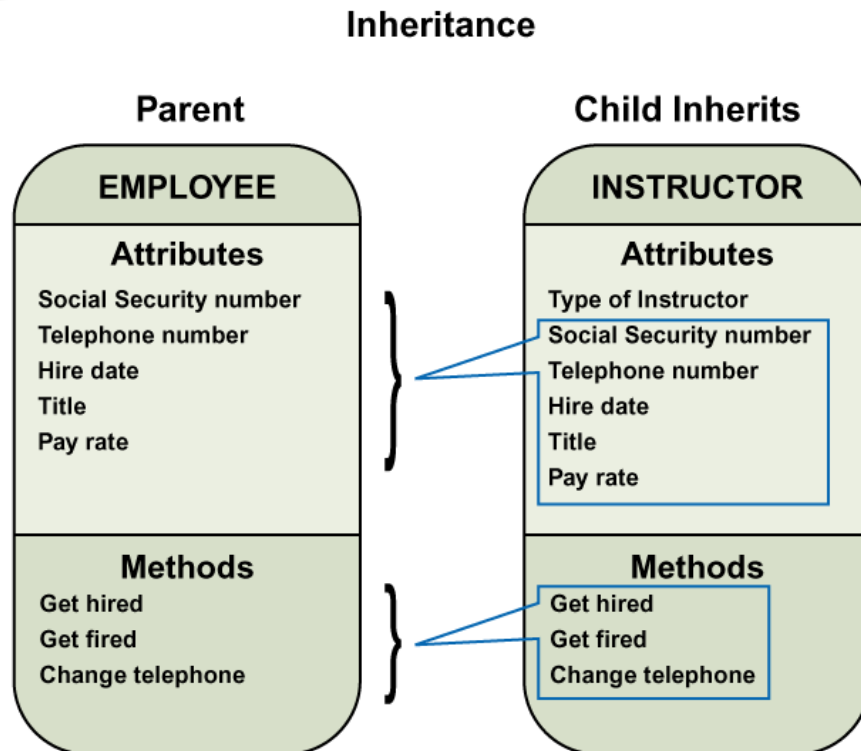
Hình 5.18. Mối quan hệ bao gộp tồn tại giữa các đối tượng *Department* và *Employee*. Tất cả các thành viên của *Department* là bộ phận của *Employee*.

Các mối quan hệ giữa các đối tượng và các lớp

❖ Mối quan hệ thừa kế

► *inheritance*

- Cho phép một đối tượng thừa kế các thuộc tính và các phương thức của một đối tượng khác.



Hình 5.19. Mối quan hệ thừa kế tồn tại giữa các đối tượng *Instructor* và *Employee*. Đối tượng *Instructor* thừa kế các đặc tính của lớp *Employee* và có thể có thêm các thuộc tính riêng.

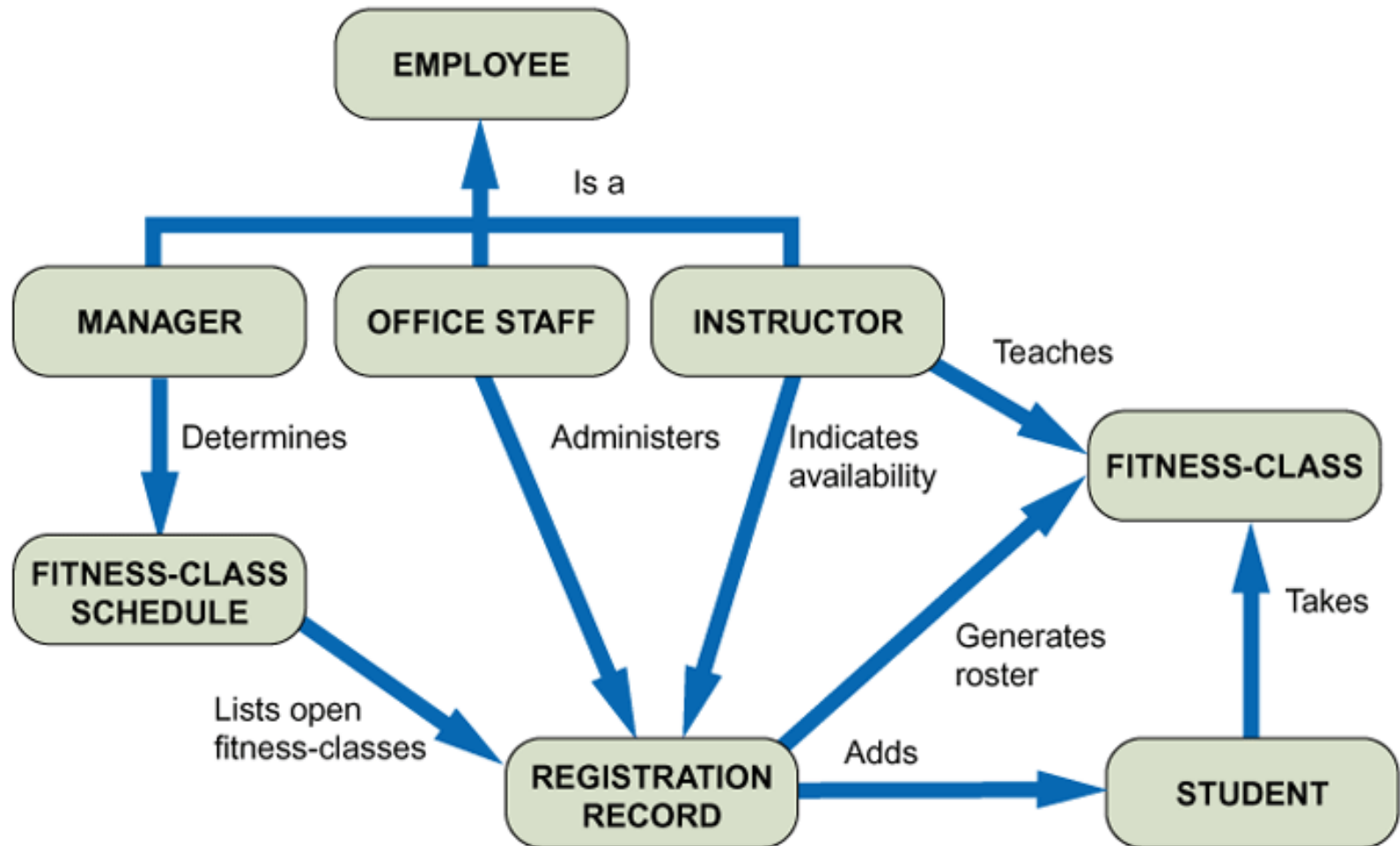


Các mối quan hệ giữa các đối tượng và các lớp

❖ Sơ đồ liên kết đối tượng

- ▶ *object relationship diagram*
- ▶ Cung cấp cái nhìn tổng quát của hệ thống.
- ▶ Hướng dẫn phát triển các sơ đồ và tài liệu tiếp theo.

Các mối quan hệ giữa các đối tượng và các lớp



Hình 5.20. Một ví dụ về sơ đồ liên kết đối tượng.

Mô hình hóa đối tượng dùng UML

- ❖ Người phân tích hệ thống sử dụng *UML* để mô tả hệ thống hướng đối tượng.
- ❖ Sử dụng các ký hiệu để biểu diễn đồ họa các thành phần và các mối quan hệ khác nhau bên trong hệ thống.
- ❖ Được sử dụng chủ yếu để hỗ trợ phân tích hệ thống hướng đối tượng và phát triển các mô hình đối tượng.
- ❖ *Object Management Group* đề nghị *UML* là một chuẩn mô hình đối tượng.

Mô hình hóa đối tượng dùng UML

❖ Sơ đồ *use case*

- ▶ Sơ đồ *use-case* biểu diễn trực quan sự tương tác giữa những người sử dụng với HTTT.
- ▶ Sơ đồ tóm lược trực quan của nhiều *use case* liên quan với nhau ở trong hệ thống.
- ▶ Phạm vi của hệ thống cho thấy cái gì được đưa vào trong hệ thống và cái gì không được đưa vào trong hệ thống.

Mô hình hóa đối tượng dùng UML

❖ Sơ đồ *use case*

- ▶ Một *use case* biểu diễn các bước trong một chức năng hoặc quá trình nghiệp vụ cụ thể.
- ▶ Một tác nhân (thực thể ngoài) khởi động một *use case* bằng cách yêu cầu hệ thống thực hiện một chức năng hoặc quá trình.
- ▶ Một *use case* có thể tương tác với các *use case* khác.
- ▶ Mỗi *use case* đều có phần mô tả.

Mô hình hóa đối tượng dùng UML

❖ Sơ đồ *use case*

- ▶ Phân loại *use-case* dựa vào mục đích và lượng thông tin của *use-case*.
 - *Use-case* khái quát (*overview use-case*) và *use-case* chi tiết (*detail use-case*).
 - *Use-case* chủ yếu (*essential use-case*) và *use-case* thực (*real use-case*).
- ▶ Các thành phần của mô tả *use-case*
 - Thông tin khái quát (*overview information*).
 - Các mối liên kết (*relationship*).
 - Dòng các sự kiện (*flow of events*).
 - Các đặc tính tùy chọn (*optional characteristic*).

Mô hình hóa đối tượng dùng UML

Use-case name:	ID:	Importance level:
Primary actor:	Use-case type:	
Stakeholders and interests:		
Brief description:		
Trigger:		
Type:		
Relationships: (Association, Include, Extend, Generalization)		
Normal flow of events:		
Subflows:		
Alternative/exceptional flows:		

Hình 5.21. Mẫu mô tả *use-case*

Mô hình hóa đối tượng dùng UML

Use-case name: Make appointment	ID: 2	Importance level: High
Primary actor: Patient	Use-case type: Detail, essential	
Stakeholders and interests: Patient - wants to make, change, or cancel an appointment Doctor - wants to ensure patient's needs are met in timely manner		
Brief description: This use case describes how we make an appointment as well as changing or canceling an appointment.		
Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment. Type: External		
Relationships: Association: Patient Include: Make Payment Arrangements Extend: Create New Patient Generalization:		

Hình 5.22. Một ví dụ của mô tả *use-case*

Mô hình hóa đối tượng dùng UML

Normal flow of events:

1. The Patients contacts the office regarding an appointment.
2. The Patient provides the Receptionist with his or her name and address.
3. The Receptionist validates that the Patient exists in the Patient database.
4. The Receptionist executes the Make Payment Arrangements use-case.
5. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment or change an existing appointment.

If the patient wants to make a new appointment,
the S-1: new appointment subflow is performed.

If the patient wants to cancel an existing appointment,
the S-2: cancel appointment subflow is performed.

If the patient wants to change an existing appointment,
the S-3: change appointment subflow is performed.

6. The Receptionist provides the results of the transaction to the Patient.

Hình 5.22. Một ví dụ của mô tả *use-case* (tiếp theo)

Mô hình hóa đối tượng dùng UML

Subflows:

S-1: New Appointment

1. The Receptionist asks the Patient for possible appointment times.
2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment.

S-2: Cancel Appointment

1. The Receptionist asks the Patient for the old appointment time.
2. The Receptionist finds the current appointment in the appointment file and cancels it

S-3: Change Appointment

1. The Receptionist performs the S-2: cancel appointment subflow.
2. The Receptionist performs the S-1: new appointment subflow.

Hình 5.22. Một ví dụ của mô tả *use-case* (tiếp theo)

Mô hình hóa đối tượng dùng UML

Subflows:

S-1: New Appointment

1. The Receptionist asks the Patient for possible appointment times.
2. The Receptionist matches the Patient's desired appointment tiems with available dates and times and schedules the new appointment.

S-2: Cancel Appointment

1. The Receptionist asks the Patient for the oldappointment time.
2. The Receptionist finds the current appointment in the appointment file and cancels it

S-3: Change Appointment

1. The Receptionist performs the S-2: cancel appointment subflow.
2. The Receptionist performs the S-1: new appointment subflow.

Alternative/exceptional flows:

3a: The Receptionist executes the Create New Patient use-case.

S-1, 2a1: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule.

S-1, 2a2: The Patient chooses one of the proposed times or decides not to make an appointment.

Hình 5.22. Một ví dụ của mô tả *use-case* (tiếp theo)

AN ACTOR:

- Is a person or system that derives benefit from and is external to the system
- Is labeled with its role
- Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead
- Is placed outside the system boundary



Actor/Role

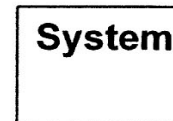
A USE CASE:

- Represents a major piece of system functionality
- Can extend another use case
- Can include another use case
- Is placed inside the system boundary
- Is labeled with a descriptive verb-noun phrase



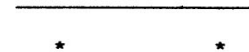
A SYSTEM BOUNDARY:

- Includes the name of the system inside or on top
- Represents the scope of the system



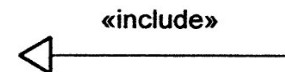
AN ASSOCIATION RELATIONSHIP:

- Links an actor with the use case(s) with which it interacts



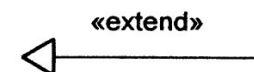
AN INCLUDE RELATIONSHIP:

- Represents the inclusion of the functionality of one use case within another
- The arrow is drawn from the base use case to the used use case



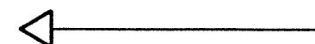
AN EXTEND RELATIONSHIP:

- Represents the extension of the use case to include optional behavior
- The arrow is drawn from the extension use case to the base use case



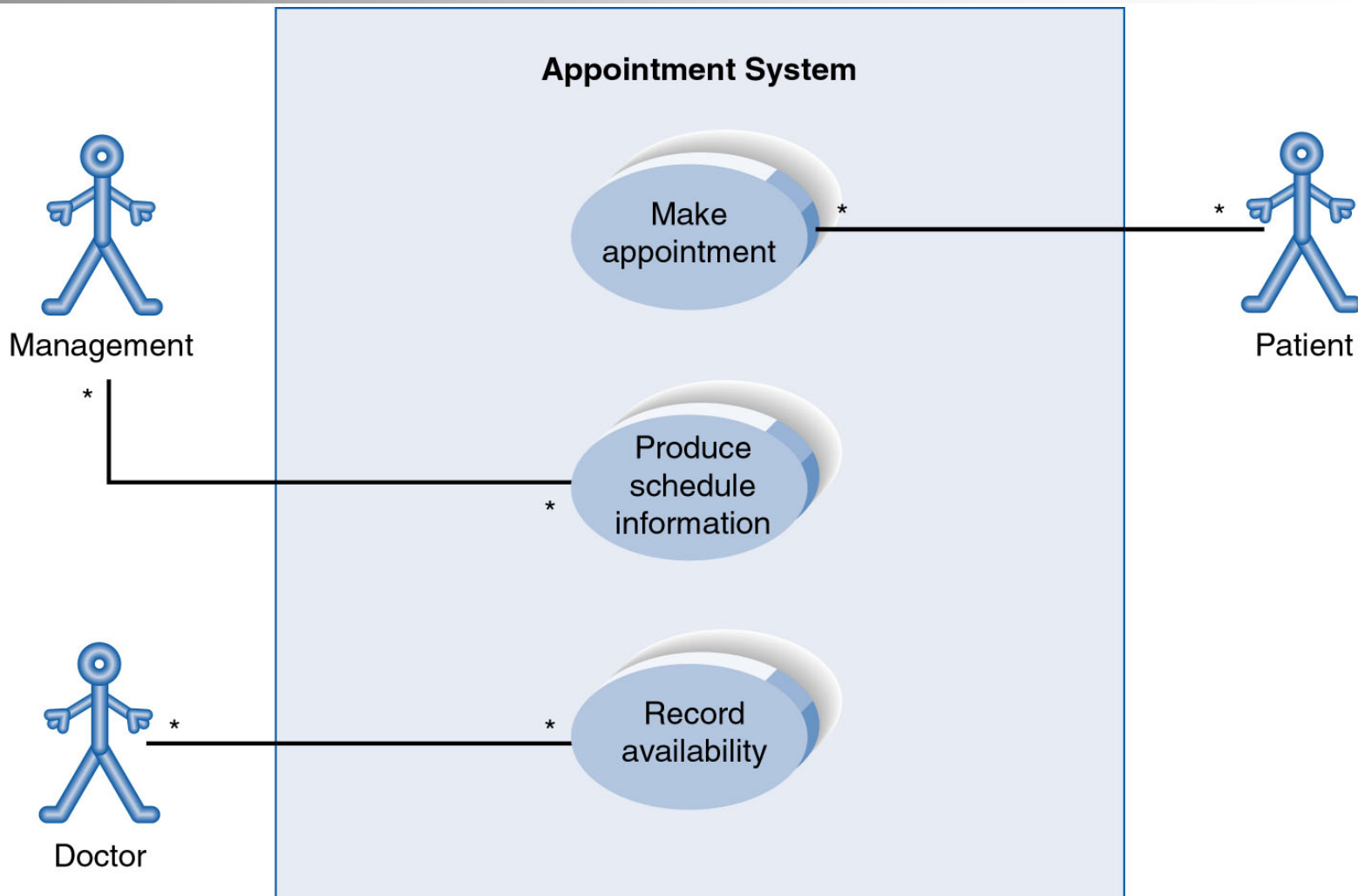
A GENERALIZATION RELATIONSHIP:

- Represents a specialized use case to a more generalized one
- The arrow is drawn from the specialized use case to the base use case



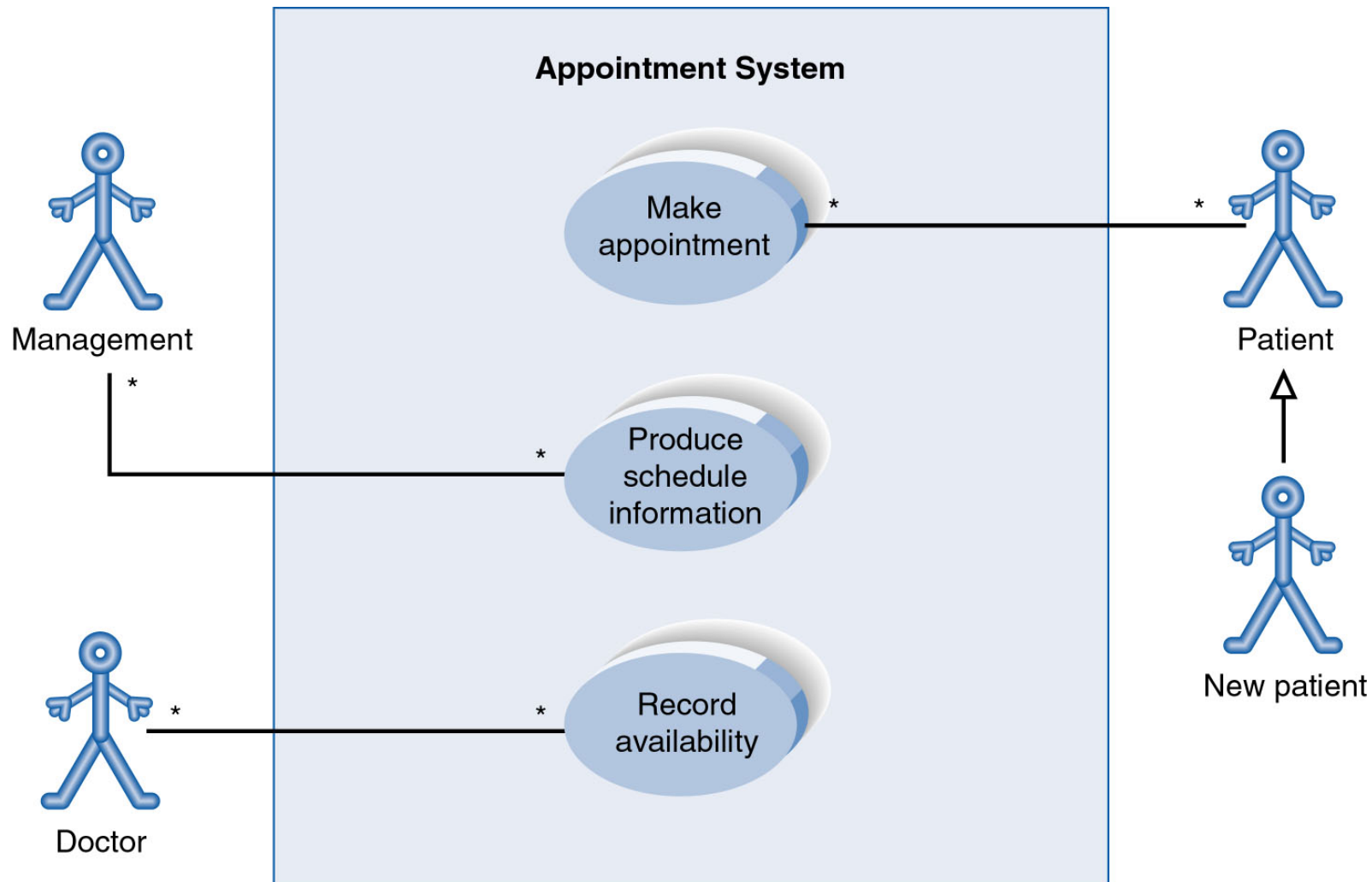
Hình 5.23. Các ký hiệu của sơ đồ use-case.

Mô hình hóa đối tượng dùng UML



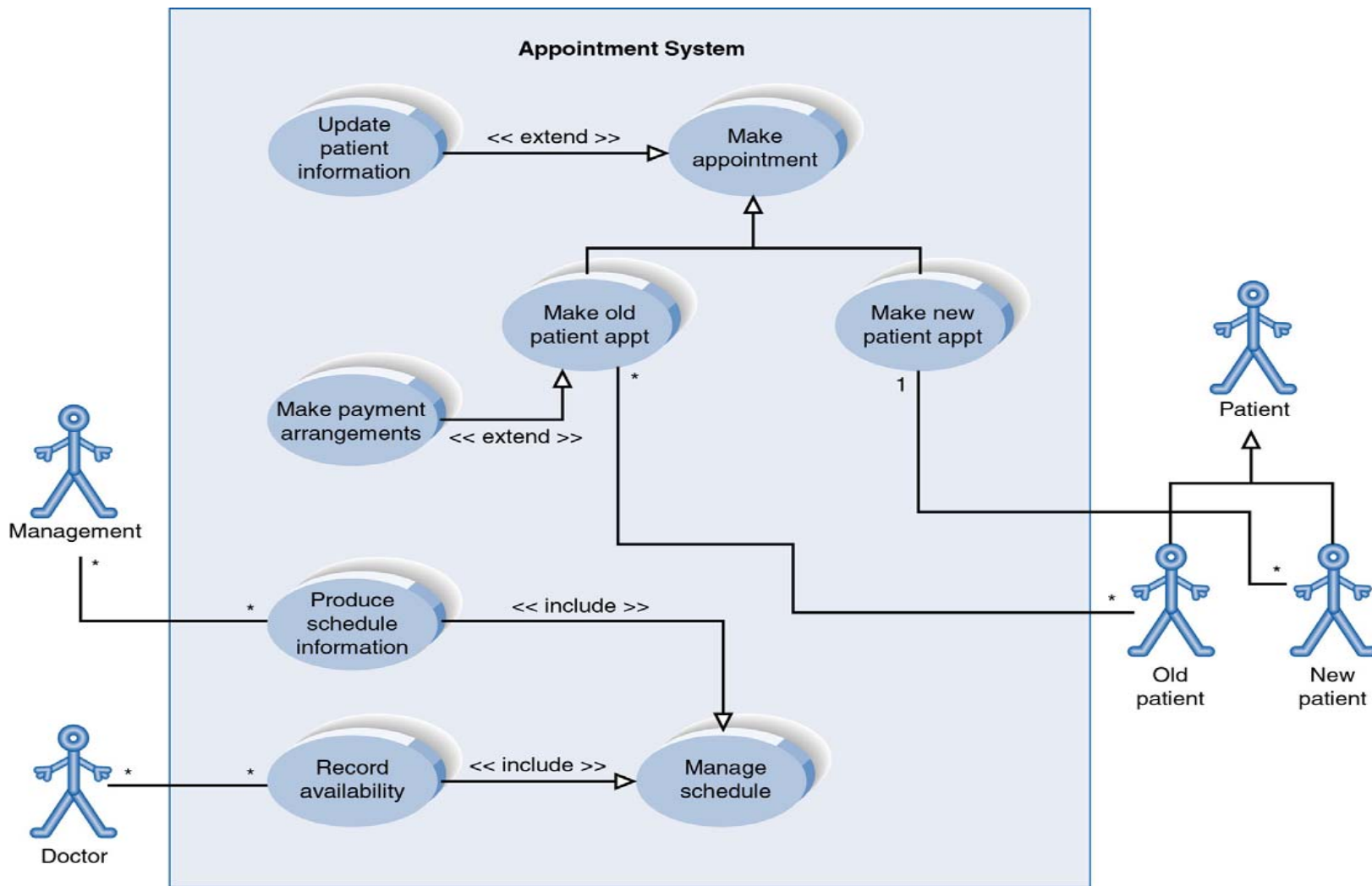
Hình 5.24. Sơ đồ use-case của hệ thống hẹn khám bệnh

Mô hình hóa đối tượng dùng UML



Hình 5.25. Sơ đồ use-case với các tác nhân chuyên biệt.

Mô hình hóa đối tượng dùng UML



Hình 5.26. Các mối liên kết *Extend* và *Include*.

Mô hình hóa đối tượng dùng UML

❖ Sơ đồ lớp

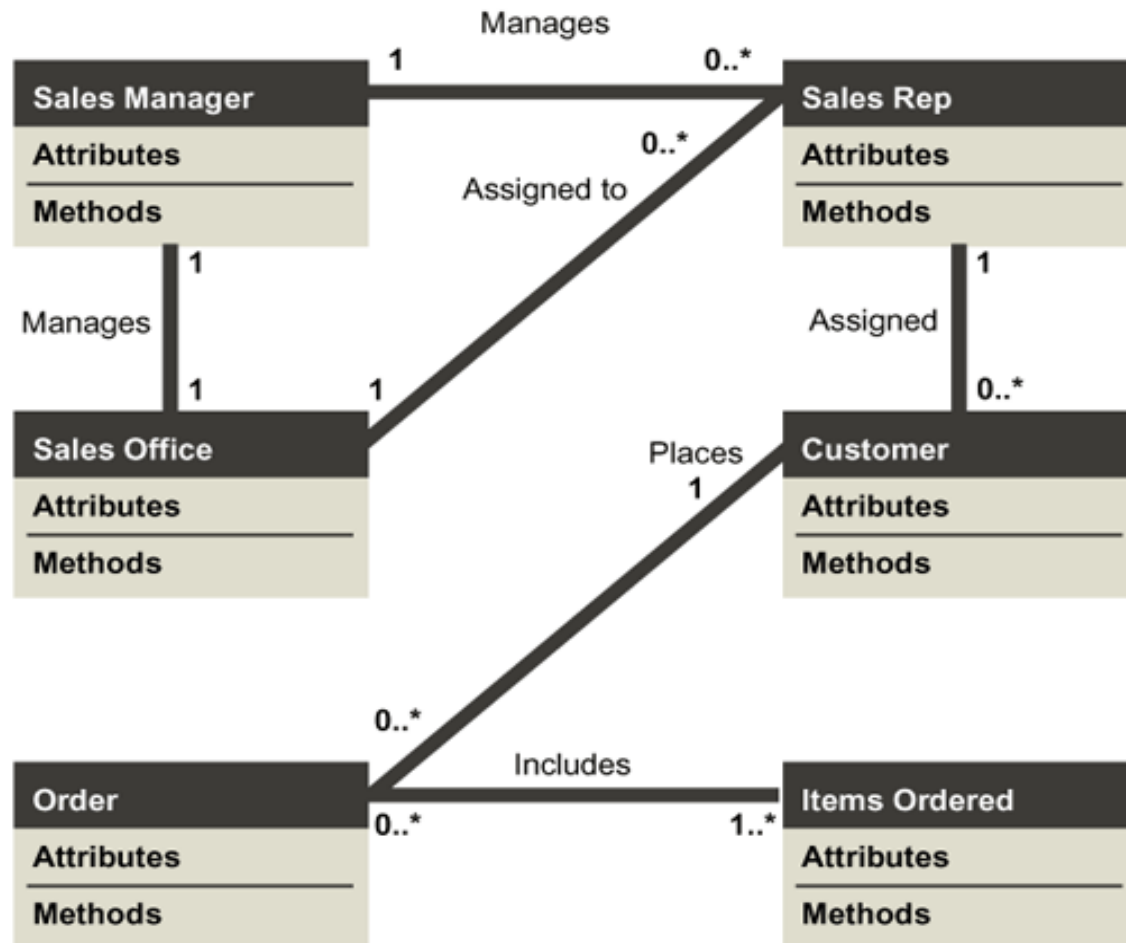
- ▶ *class diagram*
- ▶ Biểu diễn cái nhìn chi tiết của một *use case*.
- ▶ Sơ đồ lớp là một mô hình luận lý, từ đó chuyển thành mô hình vật lý và cuối cùng trở thành HTTT hoạt động.
- ▶ Lượng số mô tả các thể hiện của lớp này quan hệ như thế nào với các thể hiện của lớp khác.

Mô hình hóa đối tượng dùng UML

UML Notation	Nature of the Relationship	Example		Description
0..*	Zero or many	Employee	Payroll Deduction	An employee can have no payroll deductions or many deductions
		1	0..*	
0..1	Zero or one	Employee	Spouse	An employee can have no spouse or one spouse
		1	0..1	
1	One and only one	Office Manager	Sales Office	An office manager manages one and only one office
		1	1	
1..*	One or many	Order	Item Ordered	One order can include one or many items ordered
		1	1..*	

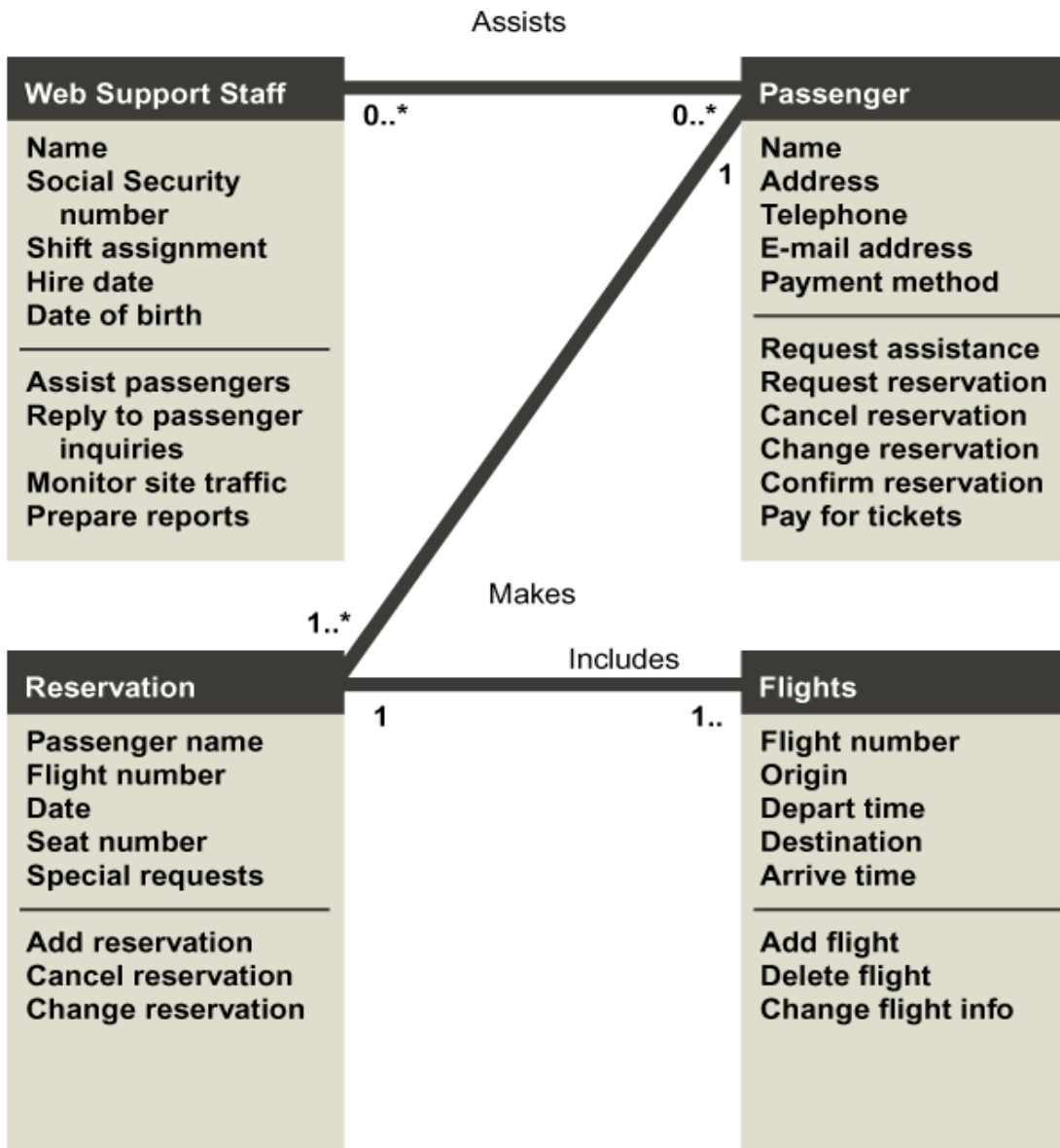
Hình 5.27. Ký hiệu UML cho thấy loại liên kết giữa các thể hiện của một lớp với các thể hiện của một lớp khác.

Mô hình hóa đối tượng dùng UML



Hình 5.28. Sơ đồ lớp cho *use-case Sales Order* (không ghi các thuộc tính và các phương thức).

Mô hình hóa đối tượng dùng UML



Hình 5.29. Sơ đồ lớp cho use-case Airline Reservation.

Mô hình hóa đối tượng dùng UML

❖ Sơ đồ tuần tự

- ▶ *sequence diagram*
- ▶ Sơ đồ tuần tự là mô hình động của một *use case*.
- ▶ Sơ đồ tuần tự cho thấy sự tương tác giữa các đối tượng theo trình tự thời gian.
- ▶ Sơ đồ tuần tự cho thấy tất cả các tác động có thể có hoặc chỉ tập trung vào một tình huống cụ thể.
- ▶ Tương tác đi từ trên xuống dọc theo đường thẳng đứng thời gian.


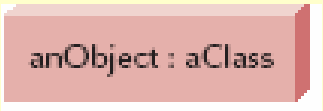
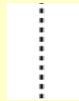
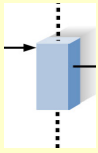
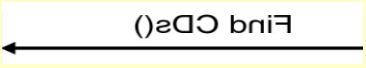



Mô hình hóa đối tượng dùng UML

❖ Sơ đồ tuần tự

- ▶ Các mũi tên nằm ngang biểu diễn các thông điệp từ đối tượng này đến đối tượng kia.
- ▶ Sơ đồ tuần tự bao gồm các ký hiệu biểu diễn các lớp, các đường sống, các thông điệp và các tiêu điểm điều khiển.

Mô hình hóa đối tượng dùng UML

Tác nhân (<i>actor</i>)	
Đối tượng (<i>object</i>)	
Đường sống (<i>lifeline</i>)	
Tiêu điểm điều khiển (<i>focus of control</i>)	
Thông điệp (<i>message</i>)	
Hủy đối tượng (<i>object destruction</i>)	

Hình 5.30. Các ký hiệu của sơ đồ tuần tự.

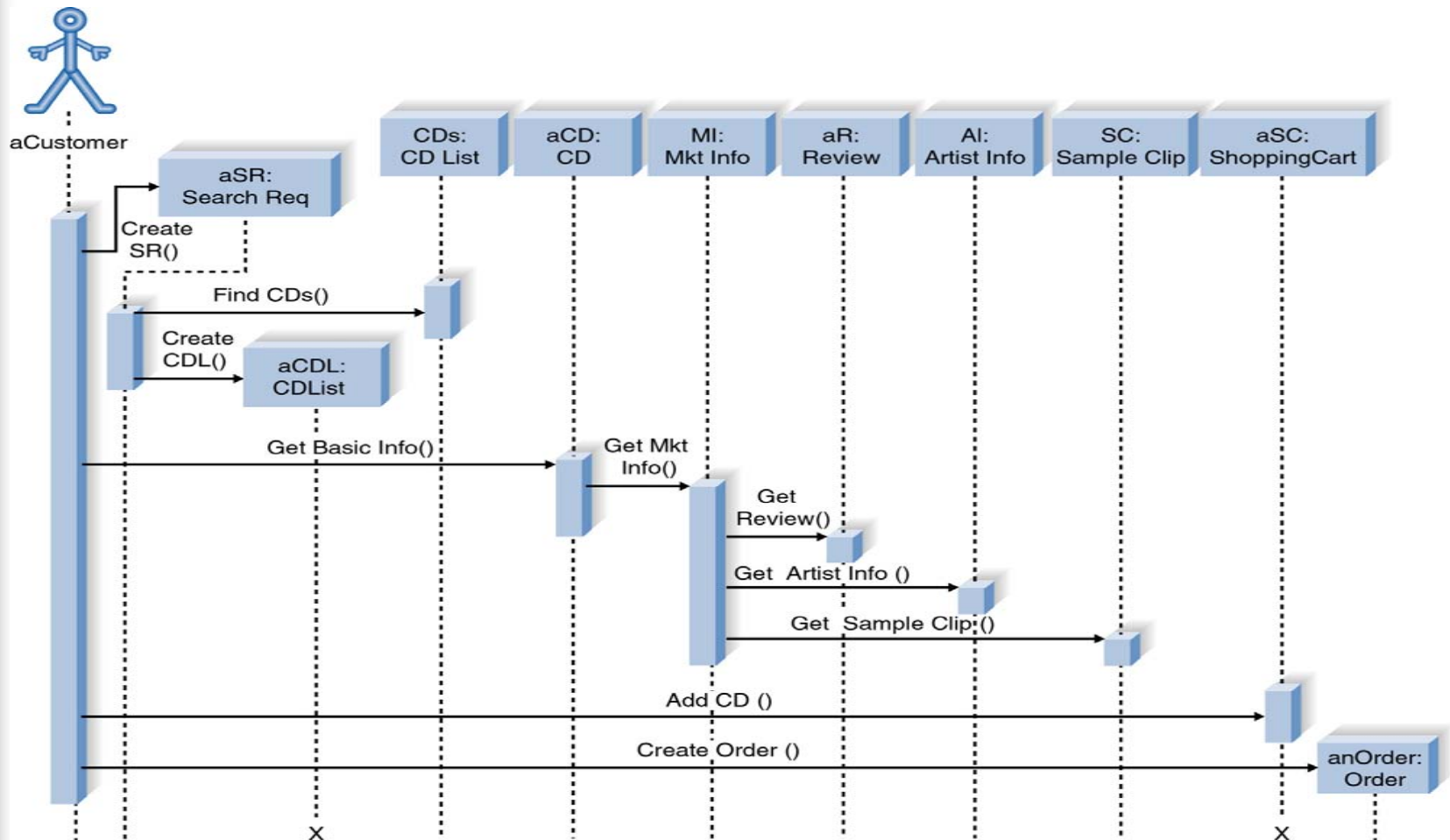
Mô hình hóa đối tượng dùng UML

Normal flow of events:

1. **Customer** submits a search request to the system.
2. The system provides the **customer** a list of recommended CDs.
3. The **customer** chooses one of the CDs to find additional information.
4. The system provides the **customer** with basic information & CD Reviews
5. The **customer** calls the **maintain order use case**.
6. The **customer** iterates over 3 through 5 until finished shopping.
7. The **customer** executes the **checkout use case**.
8. The **customer** leaves the website.

Hình 5.31. Dòng các sự kiện của *use-case Place Order*.

Mô hình hóa đối tượng dùng UML



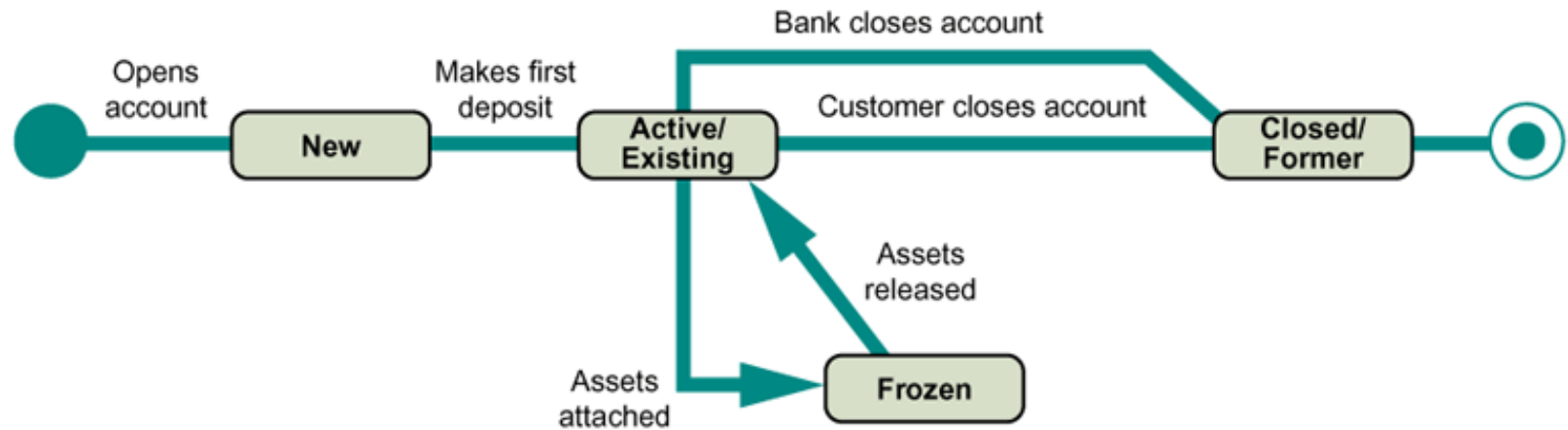
Hình 5.32. Sơ đồ tuần tự của *use-case Place Order*.

Mô hình hóa đối tượng dùng UML

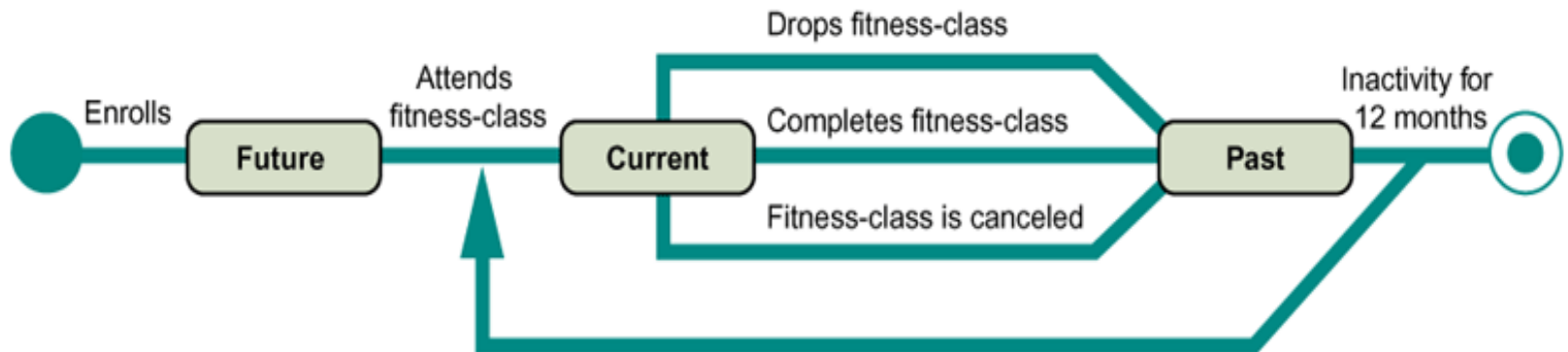
❖ Sơ đồ chuyển trạng thái

- ▶ *state transition diagram*
- ▶ Cho thấy một đối tượng thay đổi như thế nào từ trạng thái này thành trạng thái khác.
- ▶ Tất cả các trạng thái có thể có phải được thể hiện trong sơ đồ chuyển trạng thái.
- ▶ Bao gồm các ký hiệu biểu diễn sự tương tác ban đầu, các trạng thái, các sự kiện gây ra sự thay đổi trạng thái và trạng thái cuối cùng (kết thúc).

Mô hình hóa đối tượng dùng UML



Hình 5.33. Ví dụ về sơ đồ chuyển trạng thái của một tài khoản ngân hàng.



Hình 5.34. Sơ đồ chuyển trạng thái cho thấy trạng thái của sinh viên. Một sinh viên cũ không đăng ký sau 12 tháng được xử lý như sinh viên mới.

Mô hình hóa đối tượng dùng UML

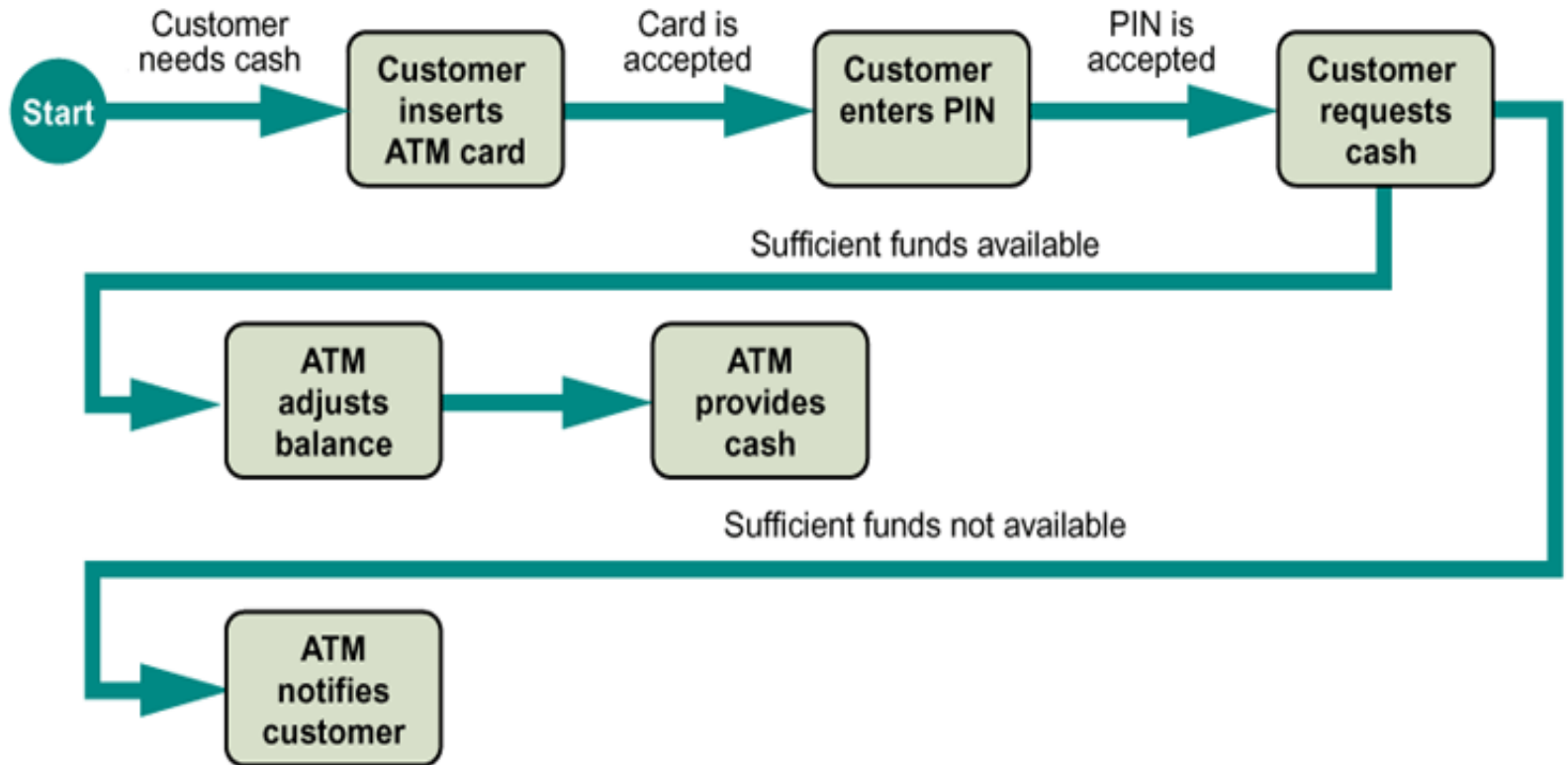
❖ Sơ đồ hoạt động

▶ *activity diagram*

- ▶ Cho thấy thứ tự xảy ra của các hành động và xác định các đường đi ra.

❖ Các sơ đồ tuần tự, sơ đồ chuyển trạng thái và sơ đồ hoạt động là các công cụ mô hình hóa động giúp người phân tích hiểu các đối tượng đối xử và tương tác như thế nào ở bên trong hệ thống.

Mô hình hóa đối tượng dùng UML



Hình 5.35. Sơ đồ hoạt động cho thấy các tác vụ và các sự kiện khi rút tiền tại máy ATM.

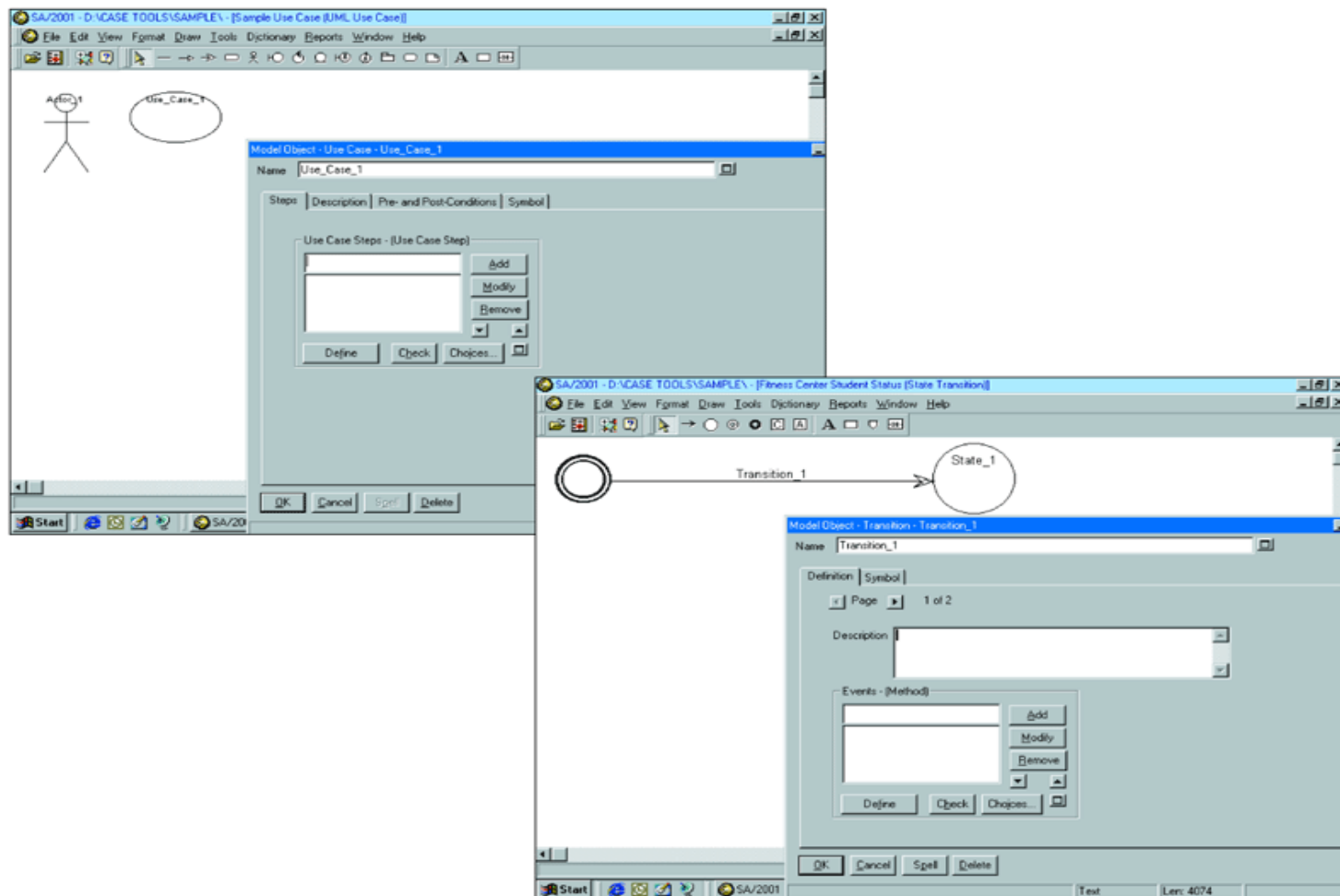


Mô hình hóa đối tượng dùng UML

❖ Công cụ CASE

- ▶ *Computer-Aided Software Engineering*
- ▶ Làm nhanh quá trình lập sơ đồ.
- ▶ Cung cấp khung thức (*framework*) cho việc lập tài liệu về các thành phần của hệ thống.
- ▶ Bảo đảm tính nhất quán và cung cấp các mối liên kết chung.
- ▶ Hệ thống *Architect 2001* là một công cụ CASE.

Mô hình hóa đối tượng dùng UML



Hình 5.36. System Architect 2001 ghi nhận use-case và sơ đồ chuyển trạng thái.

Tổ chức mô hình đối tượng

- ❖ Tổ chức các *use case* và các sơ đồ *use case* để chúng có thể được liên kết thích hợp với các sơ đồ lớp, sơ đồ chuyển trạng thái, sơ đồ tuần tự và sơ đồ hoạt động.
- ❖ Bây giờ chuẩn bị một sơ đồ để dễ dàng thay đổi phần mềm sau này.