

PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

Bài 4. UML (tiếp)

Giáo viên: TS. Trần Mạnh Tuấn

Bộ môn: Hệ thống thông tin

Khoa: Công nghệ thông tin

Email: tmtuan@tlu.edu.vn

Điện thoại: 0983.668.841

Nội dung

- ❖ UML - Overview
- ❖ UML – Khối xây dựng (Building Block)
- ❖ UML – Kiến trúc (Architecture)
- ❖ UML – Các loại mô hình hóa (Modeling Types)
- ❖ UML – Các ký hiệu căn bản (Basic Notations)
- ❖ UML – Các biểu đồ chuẩn (Standard Diagrams)

Các dạng biểu đồ UML

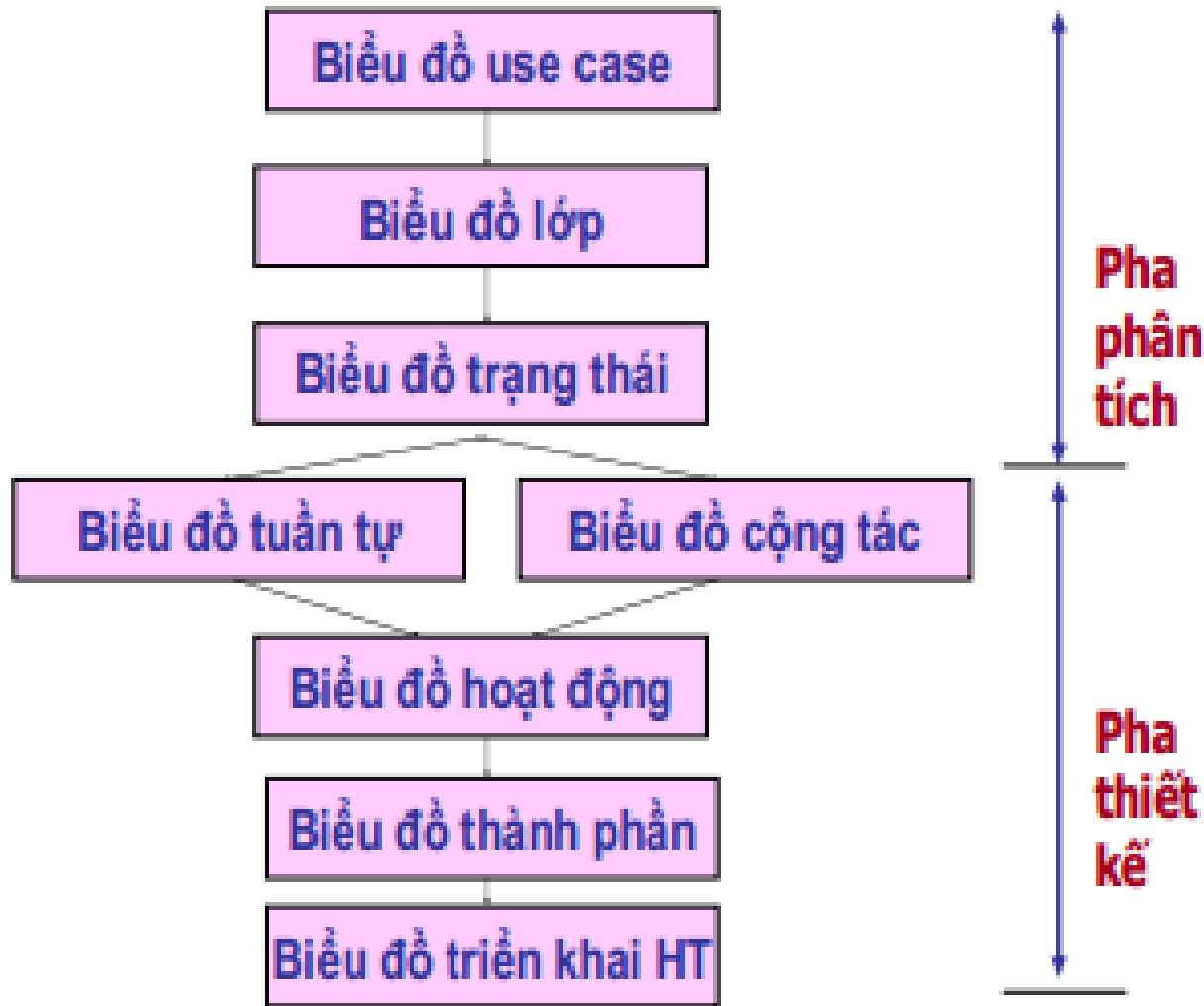
❖ Structural Diagrams (biểu đồ cấu trúc)

- Class Diagram (biểu đồ lớp)
- Object Diagram (biểu đồ đối tượng)
- Component Diagram (biểu đồ thành phần)
- Deployment Diagram (biểu đồ triển khai)

❖ Behavioral Diagrams (biểu đồ hành vi)

- Use case Diagram (biểu đồ UC)
- Sequence Diagram (biểu đồ trình tự)
- Collaboration Diagram (biểu đồ cộng tác)
- Statechart Diagram (biểu đồ trạng thái)
- Activity Diagram (Biểu đồ hoạt động)

Các dạng biểu đồ UML

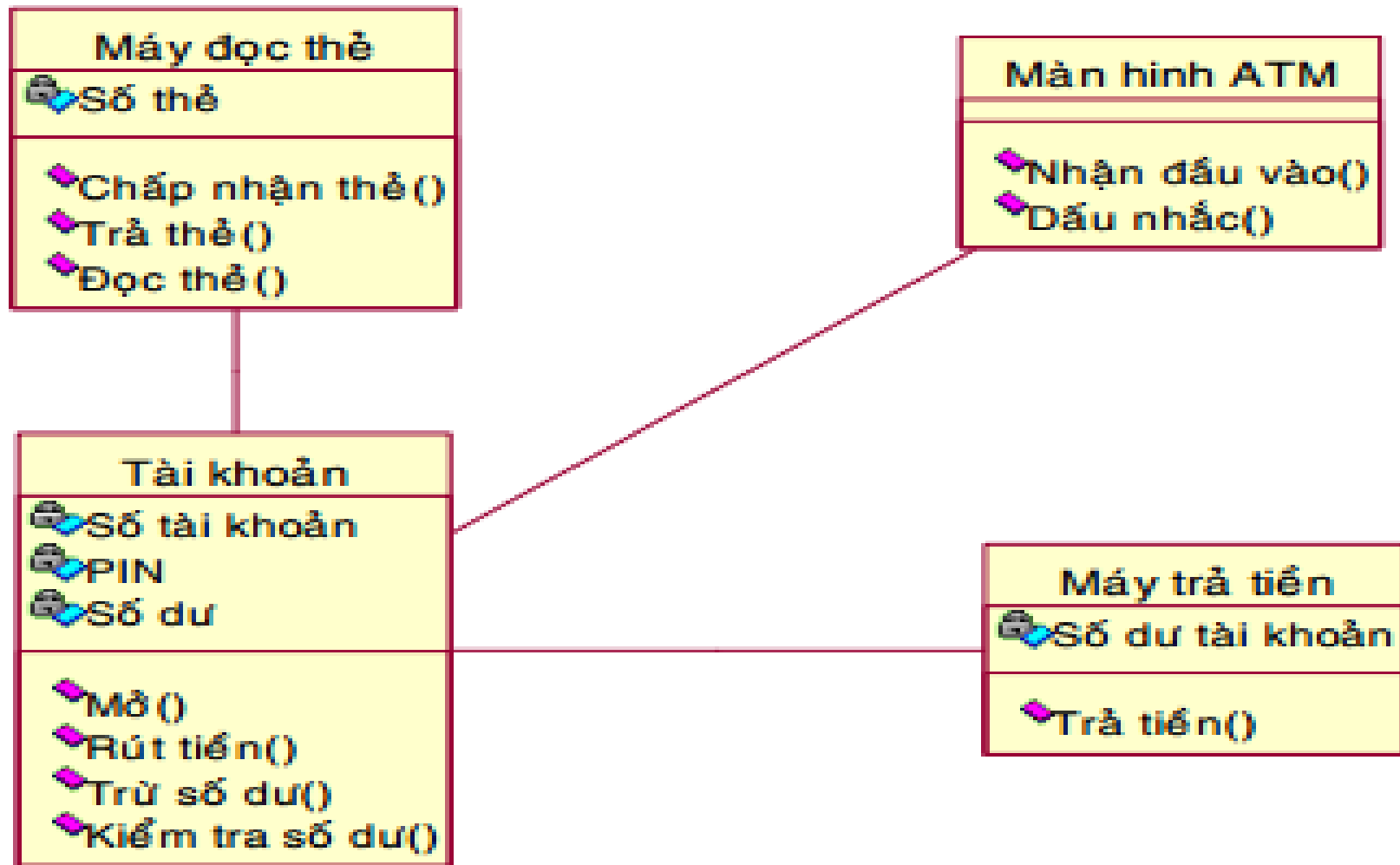


Các dạng biểu đồ UML

❖ Class Diagram:

- Hình ảnh tĩnh của ứng dụng
- Ngoài việc mô tả, hình ảnh hóa và văn bản hóa các khía cạnh của một hệ thống, Class Diagram còn xây dựng các code thực thi của ứng dụng phần mềm.
- Là biểu đồ duy nhất ánh xạ trực tiếp vào ngôn ngữ OO

Các dạng biểu đồ UML



Các dạng biểu đồ UML

❖ Mục đích của Class Diagram:

- Phân tích và thiết kế tầm nhìn tĩnh của một ứng dụng
- Mô tả trách nhiệm của hệ thống
- Cơ sở cho các biểu đồ thành phần và triển khai
- Cơ chế chuyển và nhận trong hệ thống.

❖ Khi nào sử dụng Class Diagram:

- Mô tả hình ảnh tĩnh của hệ thống
- Chỉ ra các hợp tác giữa các phần tử trong tầm nhìn tĩnh
- Mô tả các chức năng thể hiện trong hệ thống
- Xây dựng phần mềm ứng dụng sử dụng ngôn ngữ OO

Các dạng biểu đồ UML

❖ Object Diagram

- Chuyển hóa từ Class Diagram.
- Object Diagram là hình ảnh tĩnh của hệ thống, tuy nhiên hình ảnh này chỉ được thể hiện tại một thời điểm nhất định.

❖ Mục đích của Object Diagram:

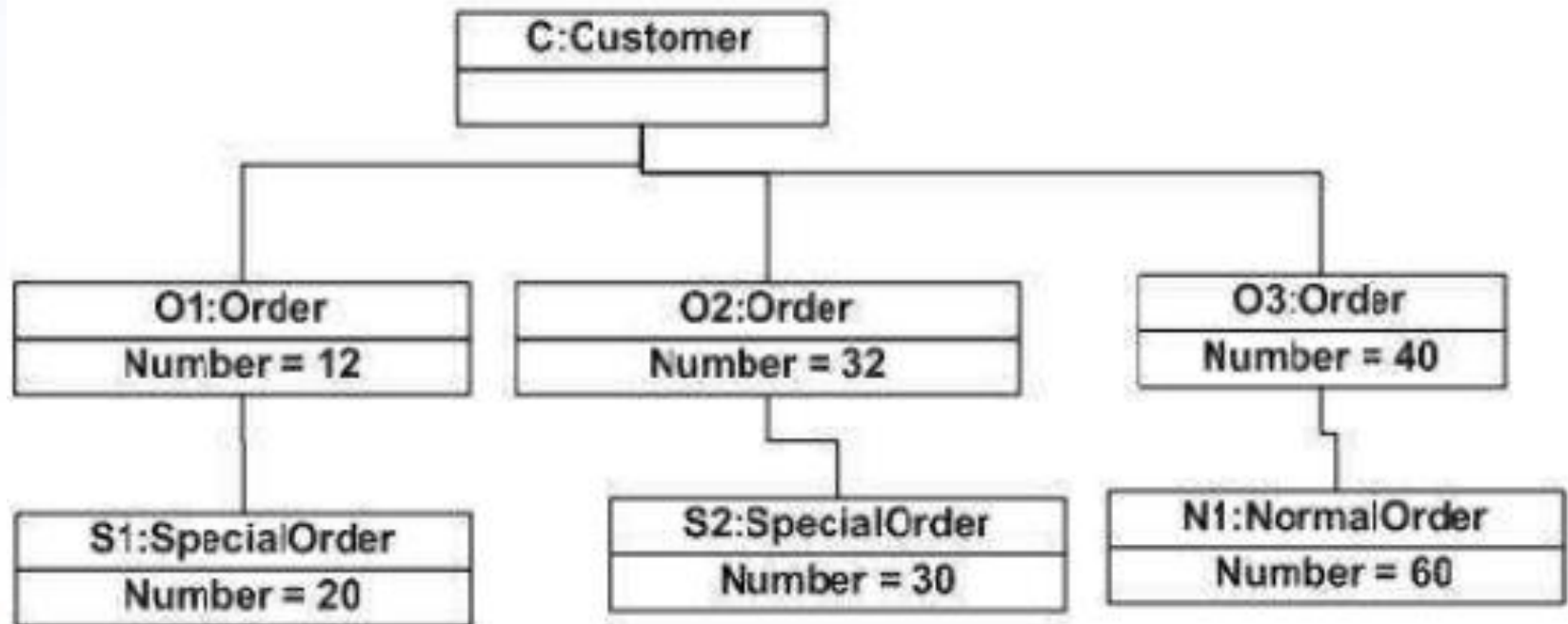
- Cơ chế kỹ thuật gửi và nhận giữa các đối tượng hệ thống
- Mối quan hệ các đối tượng của một hệ thống
- Hình ảnh tĩnh của một tương tác
- Hiểu được hành vi đối tượng và quan hệ của chúng từ khía cạnh thực hành.

❖ **Khi nào sử dụng Object Diagram:**

- Tạo giao thức cho hệ thống
- Cơ chế lưu trữ
- Mô hình hóa cấu trúc dữ liệu phức tạp
- Hiểu hệ thống từ khía cạnh thực hành.

Các dạng biểu đồ UML

Object diagram of an order management system



Các dạng biểu đồ UML

❖ **Component diagram**

- Được sử dụng để mô hình các khía cạnh vật lý của HT.
- Sử dụng mô tả các tác nhân vật lý: files, thư viện, file thực thi,...

❖ **Mục đích của Component Diagram:**

- Hình ảnh hóa các thành phần của hệ thống
- Xây dựng các thực thi bởi sử dụng kỹ thuật chuyển (forward) và đảo ngược (reverse)
- Mô tả tổ chức và các quan hệ của các thành phần (components)

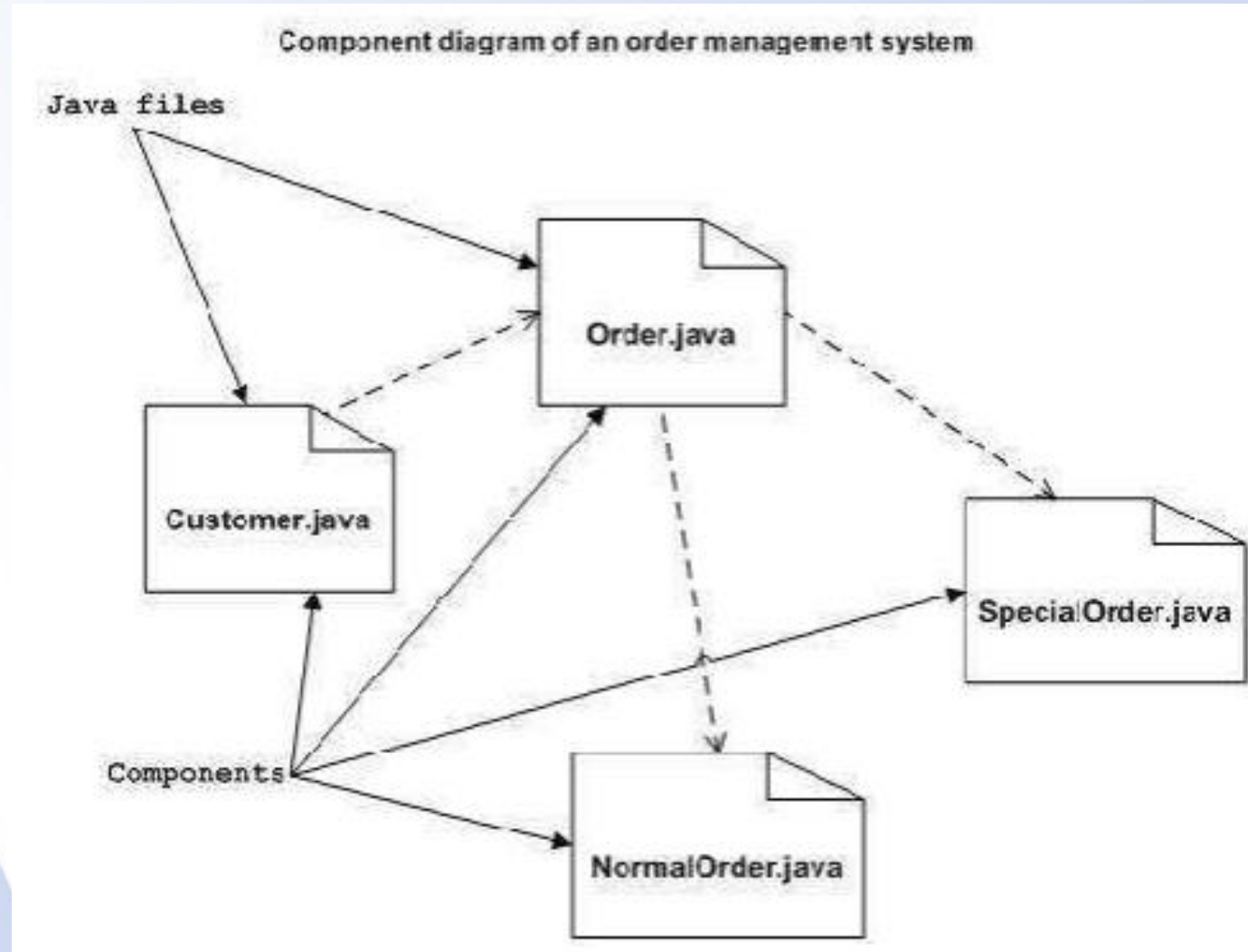
❖ **Vẽ một biểu đồ thành phần:**

- Cần xác định rõ:
 - Các files sử dụng trong hệ thống
 - Thư viện hay các thành phần khác liên quan đến hệ thống
 - Mỗi quan hệ giữa các tác nhân.
 - Sử dụng tên có ý nghĩa cho các tác nhân (thành phần)
 - Sử dụng các lưu ý cho khai báo các điểm quan trọng

❖ **Biểu đồ thành phần (Component Diagrams) được sử dụng:**

- Mô hình các thành phần của hệ thống
- Mô hình Cơ sở dữ liệu
- Mô hình các files thực thi của ứng dụng
- Mô hình source code của hệ thống.

Các dạng biểu đồ UML

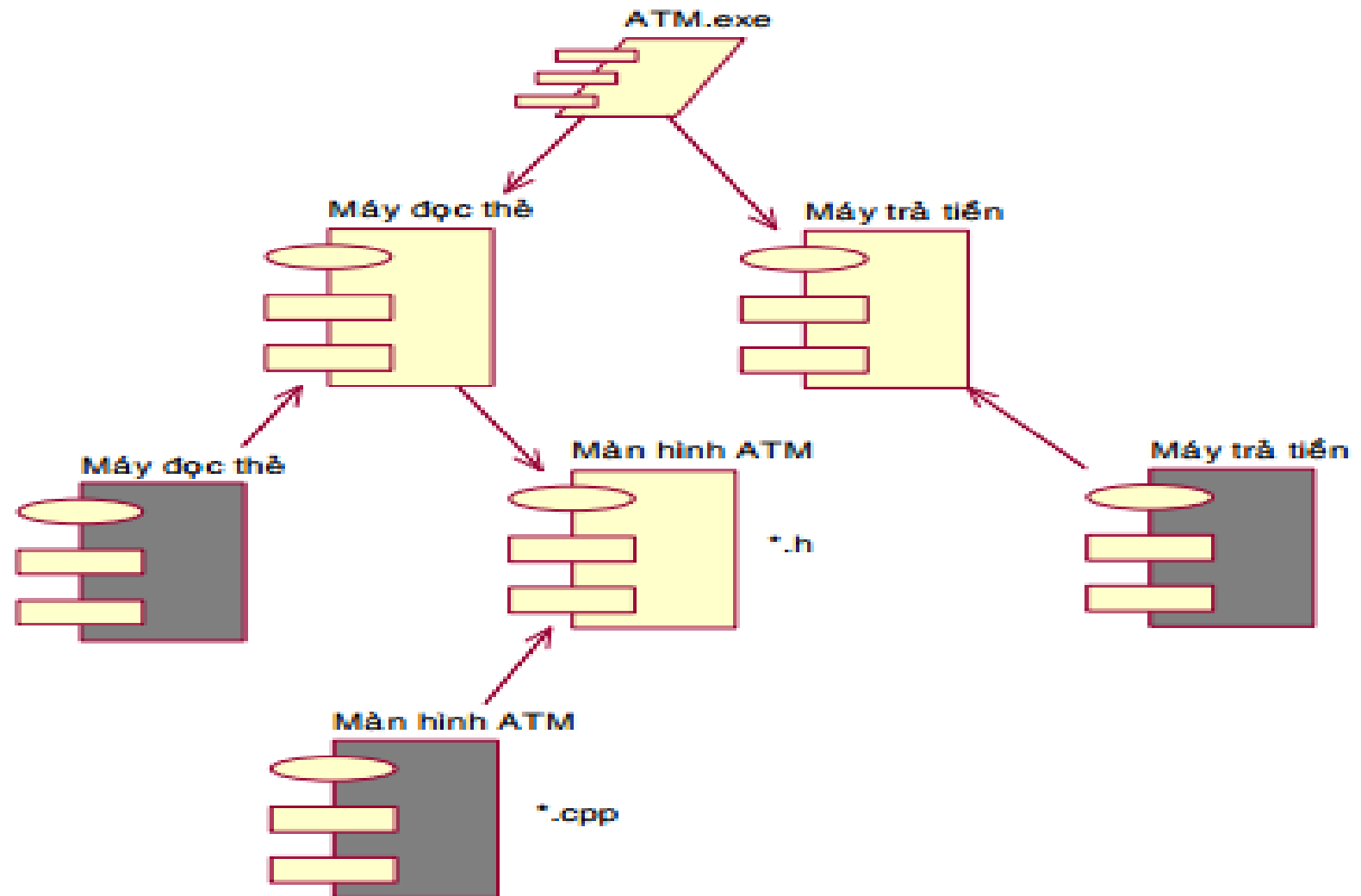


Các dạng biểu đồ UML

Component Diagram (biểu đồ thành phần)

- ❖ Biểu đồ thành phần cho ta cái nhìn vật lý của mô hình. Biểu đồ thành phần cho ta thấy được các thành phần phần mềm trong hệ thống và quan hệ giữa chúng.
- ❖ Xác định cấu trúc vật lý khi thực hiện.
- ❖ Xây dựng đặc tả kiến trúc hệ thống
- ❖ Mục đích
 - Tổ chức mã nguồn (code)
 - Xây dựng bản phác thảo thực thi
 - Xác định cơ sở dữ liệu vật lý
- ❖ Phát triển bởi kỹ sư và lập trình viên.

Các dạng biểu đồ UML



❖ **Deployment Diagram (biểu đồ triển khai)**

- Hình ảnh hóa topo của các thành phần vật lý của một hệ thống, dùng cho phần mềm triển khai trên đó.
- Là hình ảnh tĩnh của HT
- Bao gồm các nodes và các mối quan hệ của chúng.

Các dạng biểu đồ UML

❖ Mục đích của biểu đồ triển khai:

- Mô tả các thành phần phần cứng dùng để triển khai hệ thống phần mềm.
- Component Diagrams mô tả các thành phần, còn Deployment Diagrams chỉ ra chúng được triển khai ntn trên phần cứng. Hai biểu đồ trong UML mô tả phần cứng và phần mềm, trong khi các biểu đồ khác mô tả các tác nhân phần mềm của hệ thống.
- Mô hình hóa topo (cấu trúc liên kết) phần cứng của HT
- Mô tả các thành phần phần cứng sử dụng triển khai phần mềm
- Mô tả các nodes xử lý thực thi.

Các dạng biểu đồ UML

❖ **Vẽ biểu đồ Deployment:**

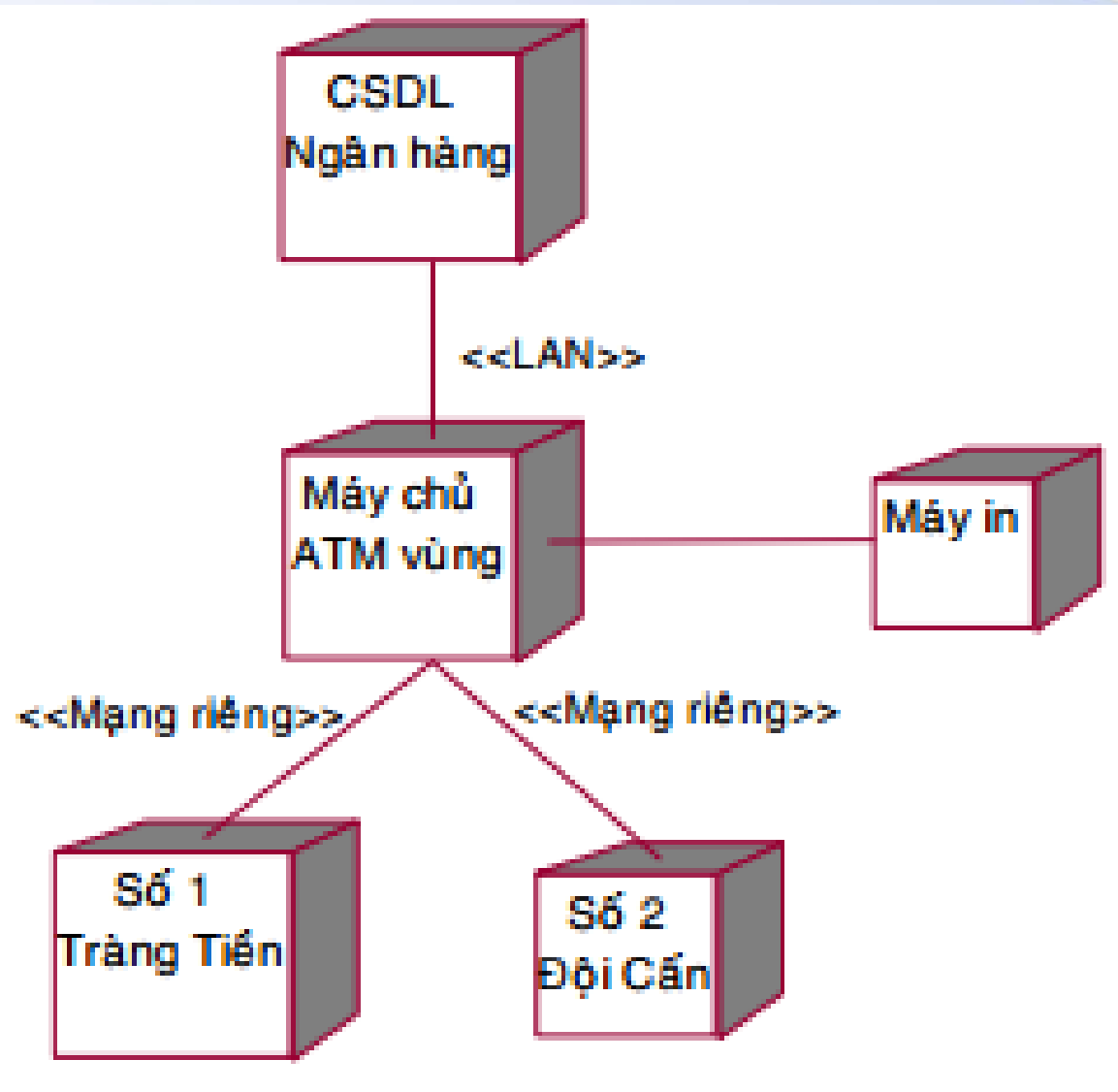
- Các tham số quan trọng trong việc thiết kế biểu đồ:
 - Performance
 - Scalability (mở rộng HT)
 - Maintainability (Bảo trì HT)
 - Portability (Tính linh động)
- Cần xác định rõ:
 - Các node
 - Quan hệ giữa các nodes

Các dạng biểu đồ UML

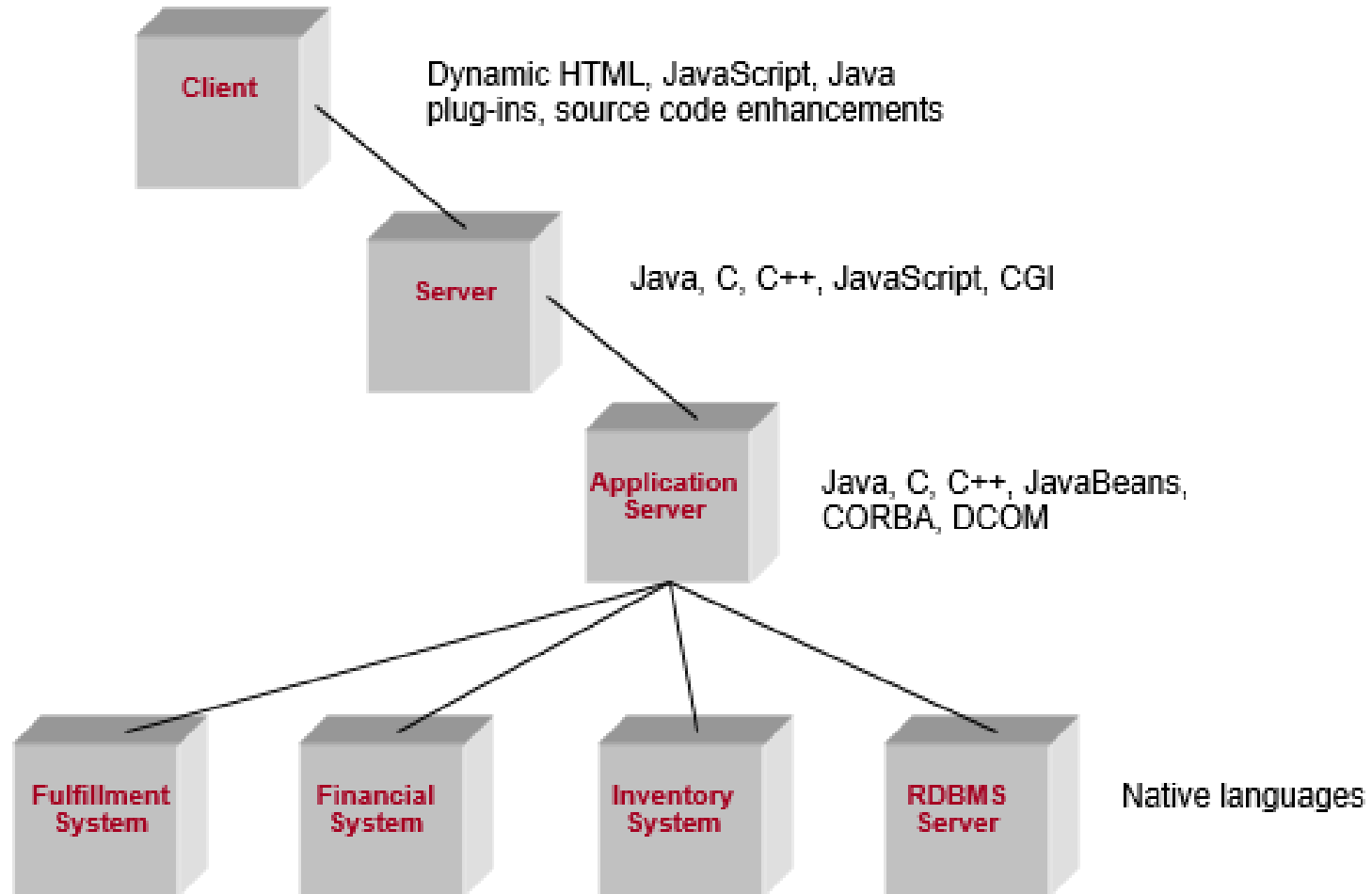
❖ Mô hình triển khai được sử dụng để:

- Mô hình Topo phần cứng của HT
- Mô hình hệ thống ẩn
- Mô hình chi tiết các phần cứng cho hệ thống client/server
- Mô hình chi tiết phần cứng chi tiết của ứng dụng phân phối
- Cho kỹ thuật chuyển và đảo nghịch

Các dạng biểu đồ UML

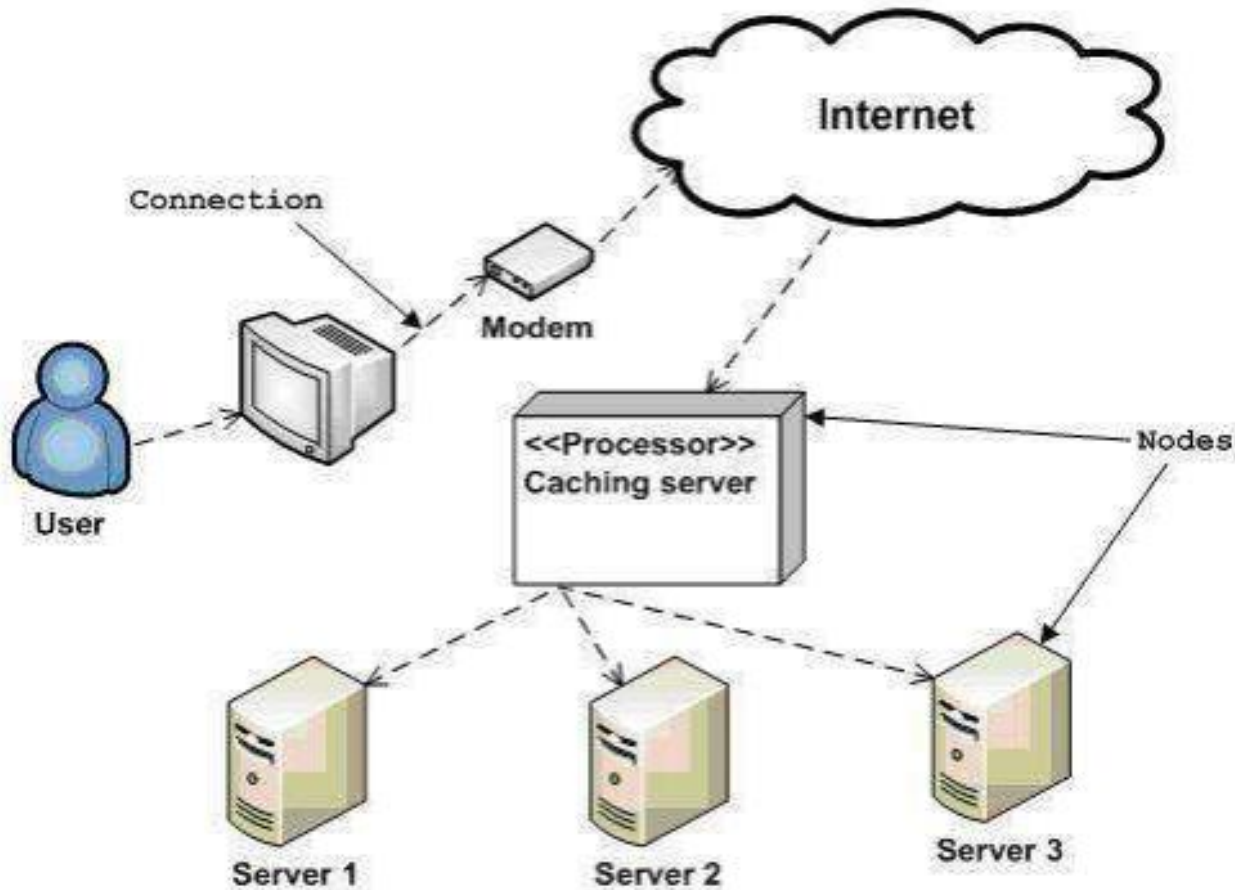


Các dạng biểu đồ UML



Các dạng biểu đồ UML

Deployment diagram of an order management system



Các dạng biểu đồ UML

❖ Use case Diagram

Khía cạnh quan trọng nhất của mô hình hệ thống là biểu diễn các hành vi động – hành vi của hệ thống khi chạy hay vận hành. Use case là một trong 5 biểu đồ động.

- ❖ Thành phần liên quan đến use case là vai trò người dùng (actors).
- ❖ Biểu đồ use case bao gồm actors, các use case và quan hệ giữa chúng. Biểu đồ được dùng cho mô hình hệ thống/hệ thống con của ứng dụng. Một use case đơn thể hiện một chức năng cụ thể của hệ thống.

Các dạng biểu đồ UML

Mục đích:

- Thu thập các yêu cầu của HT
- Mô tả hình ảnh bên ngoài Của hệ thống.
- Xác định các nhân tố trong và ngoài tác động hệ thống
- Chỉ ra tương tác giữa các yêu cầu là các tác nhân.

Các dạng biểu đồ UML

❖ **Vẽ biểu đồ Use case**

- Biểu đồ use case được xem như phân tích yêu cầu cao cấp của hệ thống. Khi các yêu cầu hệ thống được phân tích, các chức năng được thể hiện trong các use case.
- Actors có thể là người dùng, là ứng dụng nội tại, hay là các ứng dụng bên ngoài.
- Khi vẽ cần xác định:
 - Các chức năng được thể hiện như use case
 - Actors
 - Quan hệ giữa các use case và actors.

Các dạng biểu đồ UML

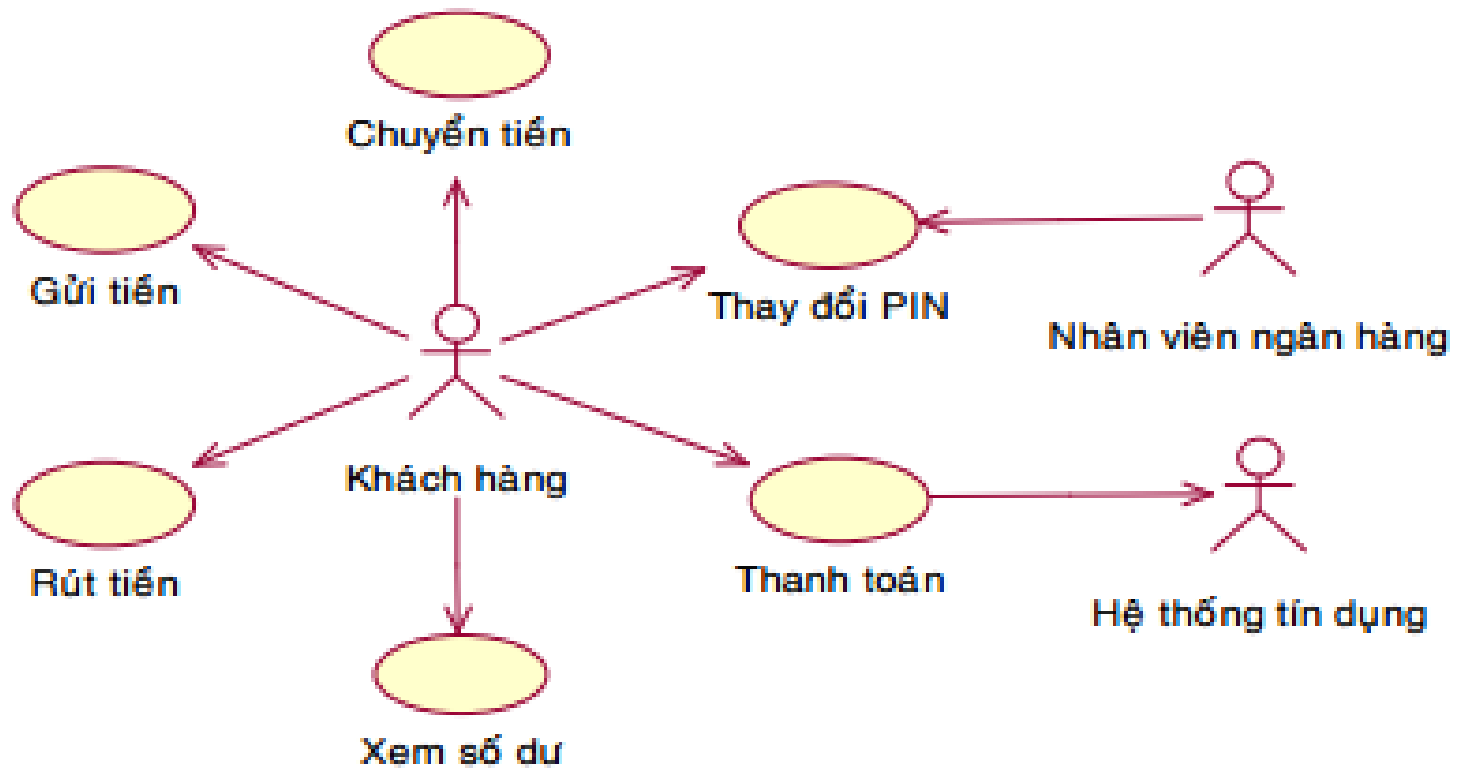
■ Lưu ý:

- Tên của use case rất quan trọng
- Tên thích hợp cho actors
- Hiển thị mối quan hệ và phụ thuộc rõ ràng trong biểu đồ
- Không cần cố đưa ra tất các kiểu quan hệ. Mục đích chính của biểu đồ là xác định các yêu cầu.
- Sử dụng chú ý khi có các điểm quan trọng

❖ **Biểu đồ use case được sử dụng ở đâu?**

- Phân tích yêu cầu và thiết kế cấp độ cao
- Mô hình nội dung của hệ thống
- Kỹ thuật đảo nghịch
- Kỹ thuật chuyển.

Các dạng biểu đồ UML



Các dạng biểu đồ UML

Các biểu đồ tương tác - Interactions

❖ Bao gồm:

- Biểu đồ tuần tự (Sequence diagram)
- Biểu đồ hợp tác (Collaboration diagram)

❖ Mục đích:

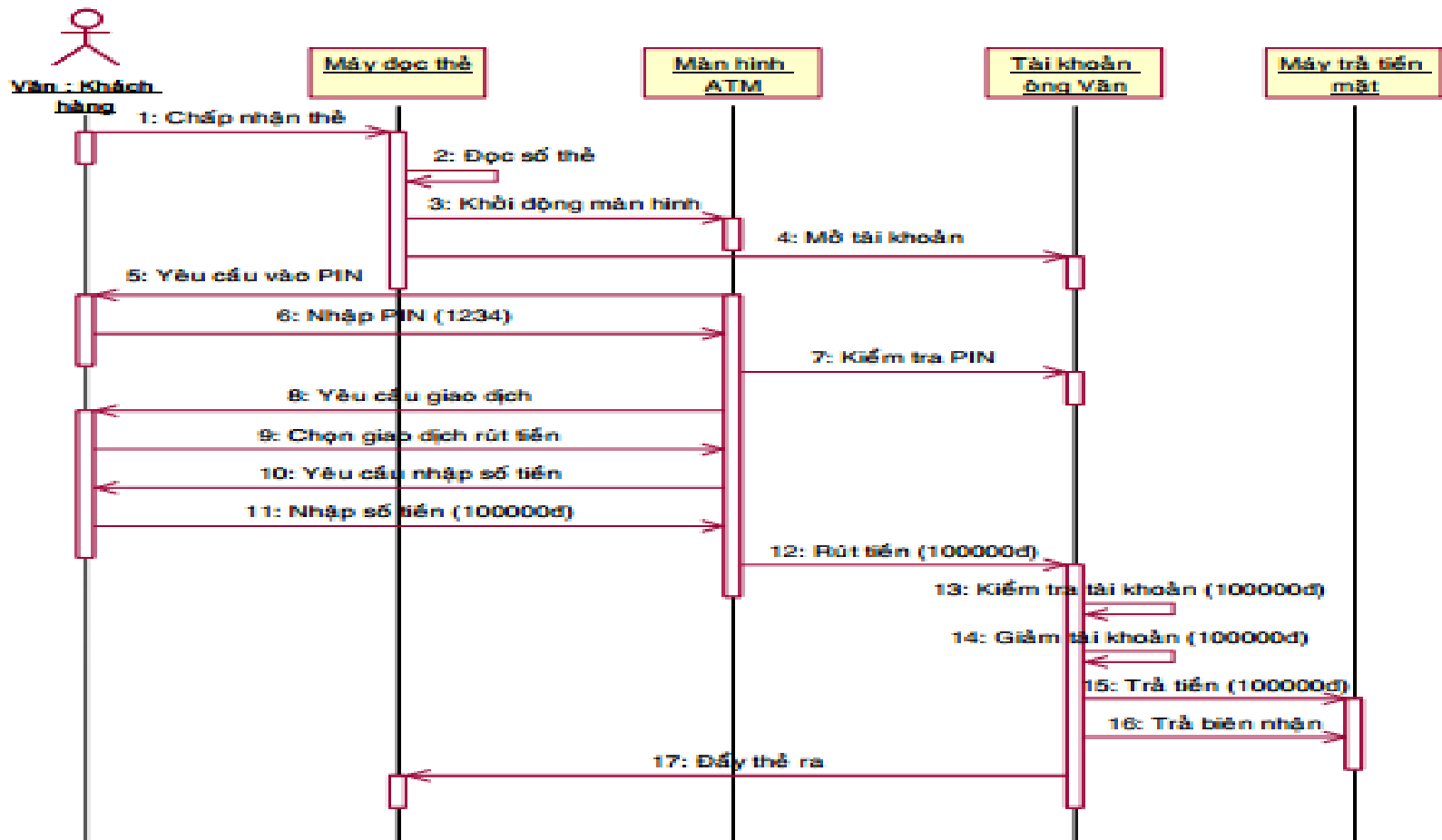
- Hình ảnh hóa các hành vi tương tác của hệ thống.
- Mô tả các luồng thông điệp (messages) trong hệ thống
- Mô tả tổ chức cấu trúc của các đối tượng
- Mô tả tương tác giữa các đối tượng

❖ Vẽ các biểu đồ tương tác:

- Cần xác định rõ trước khi vẽ:
 - Các đối tượng đóng vai trò trong tương tác
 - Các luồng thông điệp (messages) giữa các đối tượng
 - Sự tuần tự trong các luồng thông điệp
 - Tổ chức đối tượng

Các dạng biểu đồ UML

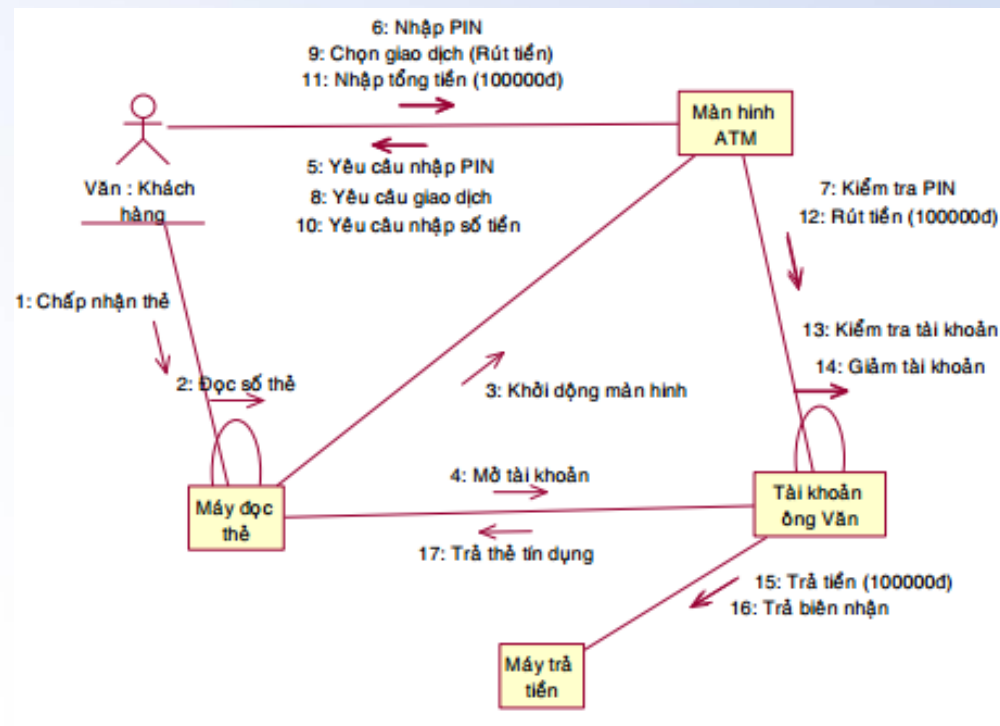
❖ Biểu đồ tuần tự (Sequence diagram)



Các dạng biểu đồ UML

❖ Biểu đồ hợp tác (Collaboration diagram)

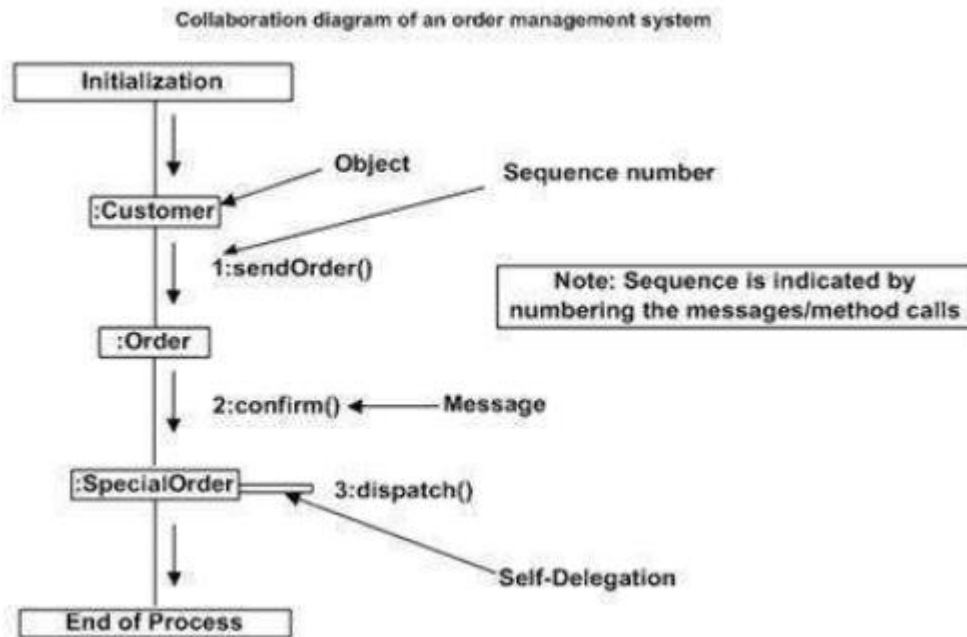
- Đưa ra tổ chức đối tượng mà ở đó phương thức gọi tuần tự với số thứ tự kèm theo.
- Các phương thức tương tự như sequence diagram. Khác là sequence diagram ko mô tả tổ chức đối tượng, còn collaboration diagram thì có.



Các dạng biểu đồ UML

❖ Sử dụng Interaction:

- Mô hình luồng điều khiển theo luồng thời gian
- Mô hình luồng điều khiển theo tổ chức cấu trúc
- Cho kỹ thuật chuyển và đảo nghịch



Biểu đồ trạng thái - Statechart

- ❖ **Mô tả các trạng thái khác nhau của một thành phần (component)/đối tượng trong hệ thống.**
- ❖ **Mục đích:**
 - Mô hình khía cạnh động của hệ thống
 - Mô hình vòng đời của hệ thống tương tác-quá sự kiện
 - Mô tả các trạng thái khác nhau của đối tượng
 - Xác định máy trạng thái để mô hình các trạng thái của đối tượng – (state machine)

Các dạng biểu đồ UML

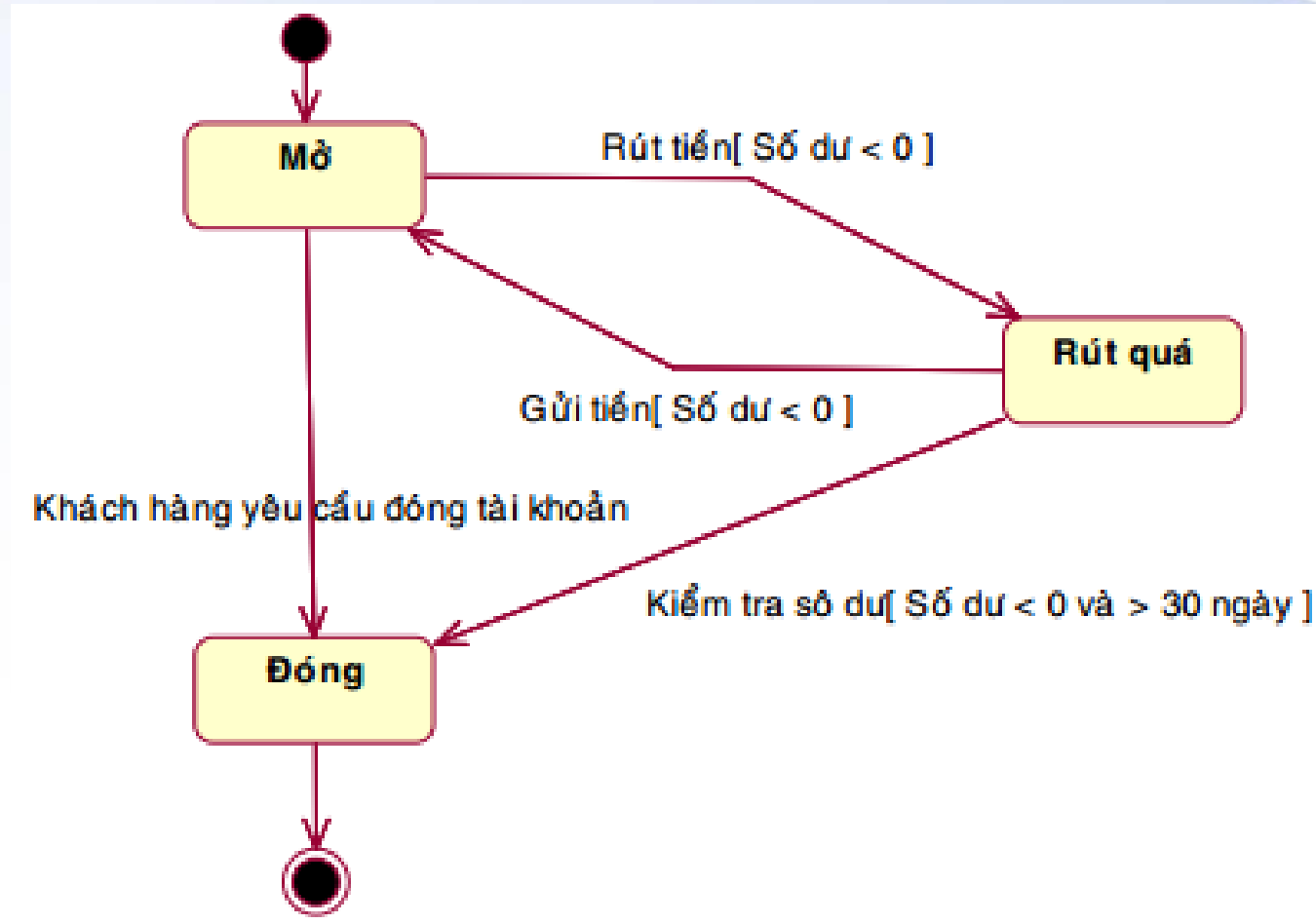
❖ **Vẽ biểu đồ trạng thái (Statechart)**

- Cần xác định:
 - Xác định các đối tượng quan trọng được phân tích
 - Xác định các trạng thái
 - Xác định các sự kiện

❖ **Statechart được sử dụng ở đâu?**

- Được sử dụng để mô hình các trạng thái và các sự kiện vận hành trong hệ thống.
- Để mô hình các trạng thái của hệ thống
- Để mô hình hệ thống phản ứng (bao gồm các đối tượng phản ứng)
- Xác định trách nhiệm các sự kiện khi thay đổi trạng thái
- Kỹ thuật chuyển và đảo nghịch.

Các dạng biểu đồ UML



Biểu đồ hành động (Activity Diagram)

- ❖ Biểu đồ thể hiện luồng từ một hành động tới một hành động khác. Luồng này có thể là tuần tự, rẽ nhánh hay đồng thời.
- ❖ **Mục đích:**
 - Vẽ luồng hành động của hệ thống
 - Mô tả tuần tự từng hành động
 - Mô tả luồng song song, rẽ nhánh và đồng thời của hệ thống.

Các dạng biểu đồ UML

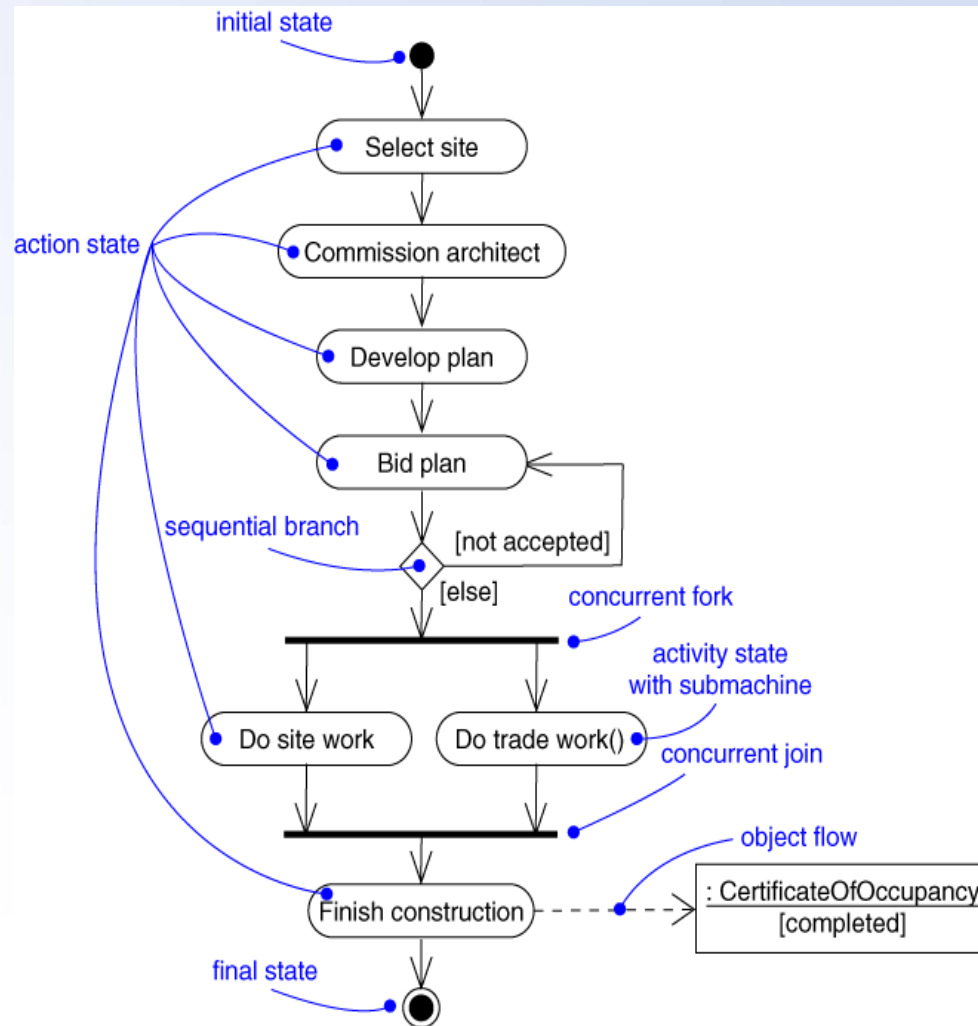
❖ **Vẽ biểu đồ hành động:**

- Xác định các phần tử sau:
 - Các hành động (Activities)
 - Sự kết hợp
 - Các điều kiện
 - Các ràng buộc

❖ **Khi nào sử dụng biểu đồ hành động:**

- Mô hình luồng công việc bởi các hành động được sử dụng (Quy trình nghiệp vụ)
- Mô hình các yêu cầu kinh doanh
- Hiểu sâu hơn về các chức năng hệ thống
- Khảo sát các yêu cầu kinh doanh ở giai đoạn sau.

Các dạng biểu đồ UML



Trao đổi, câu hỏi?