
PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

Nội dung

Phần I: Tổng quan

Chương 1 – Tổng quan về HTTT

Chương 2 – Mô hình và phương pháp mô hình hóa HTTT

Phần II: Phân tích

Chương 3 – Xác định yêu cầu và chọn lựa phương án

Chương 4 – Mô hình hóa dữ liệu

Chương 5 – Mô hình hóa xử lý

Phần III: Thiết kế

Chương 6 – Thiết kế dữ liệu

Chương 7 – Thiết kế hệ thống

Chương 8 – Thiết kế giao diện

Chương 7 – Thiết kế hệ thống

- 1. Phân chia hệ thống thành các hệ thống con**
- 2. Xây dựng mô hình thiết kế xử lý**
- 3. Thiết kế chức năng phần mềm**

1. Phân chia hệ thống thành các hệ thống con

Mục tiêu:

- Giảm sự phức tạp của hệ thống.
- Dễ dàng hơn cho việc thiết kế và bảo dưỡng hệ thống.

1. Phân chia hệ thống thành các hệ thống con (tt)

Tiêu chí phân chia:

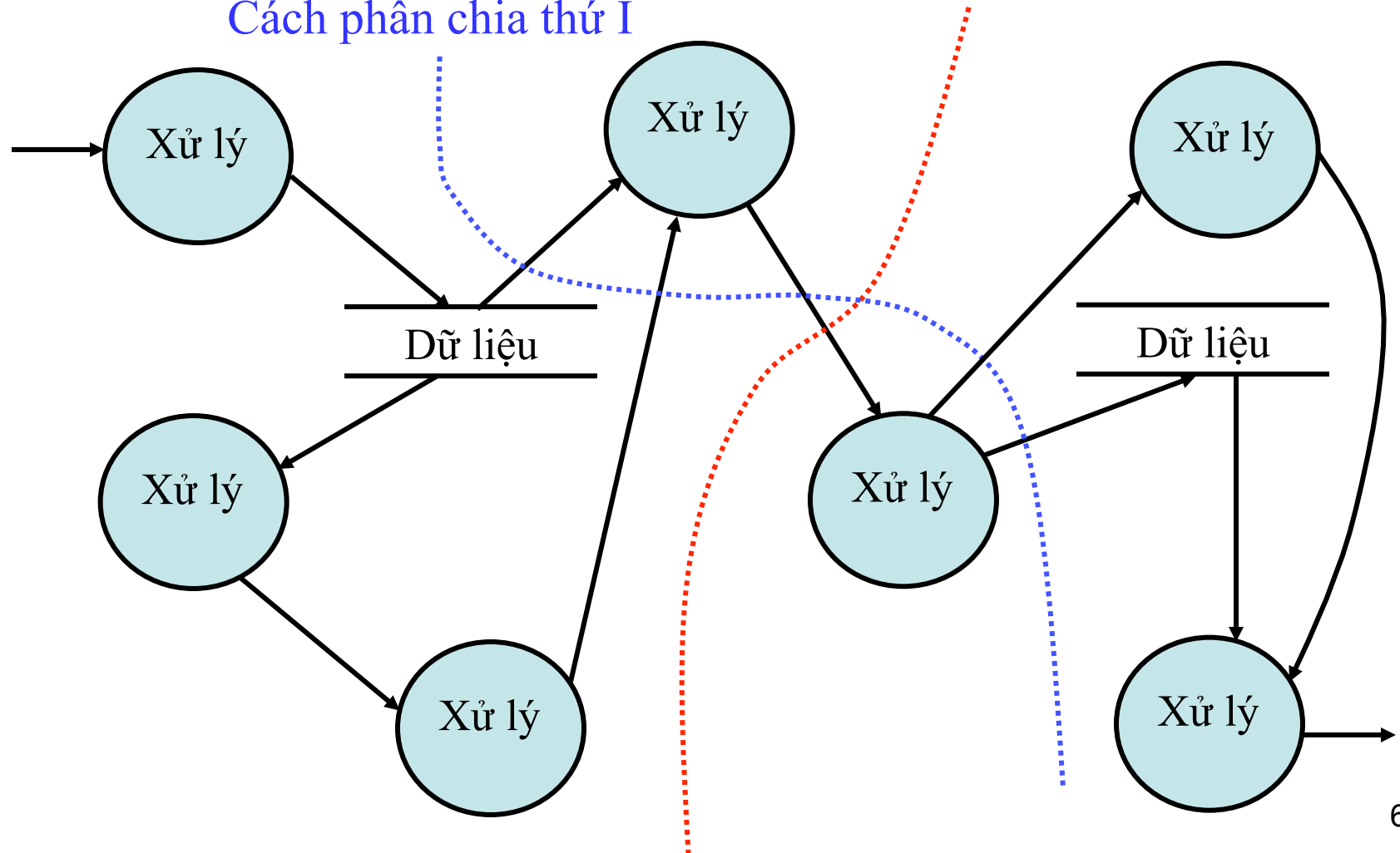
- ***Tính kết dính (cohesion)***: sự gắn bó về luận lý hoặc mục đích của các xử lý trong một hệ thống con.
→ Tính kết dính càng cao càng tốt.
- ***Tính liên kết (coupling)***: Sự trao đổi thông tin và tác động lẫn nhau giữa các hệ thống con.
→ Sự liên kết này càng ít càng tốt.

1. Phân chia hệ thống thành các hệ thống con (tt)

Tiêu chí phân chia (tt)

Cách phân chia thứ II

Cách phân chia thứ I



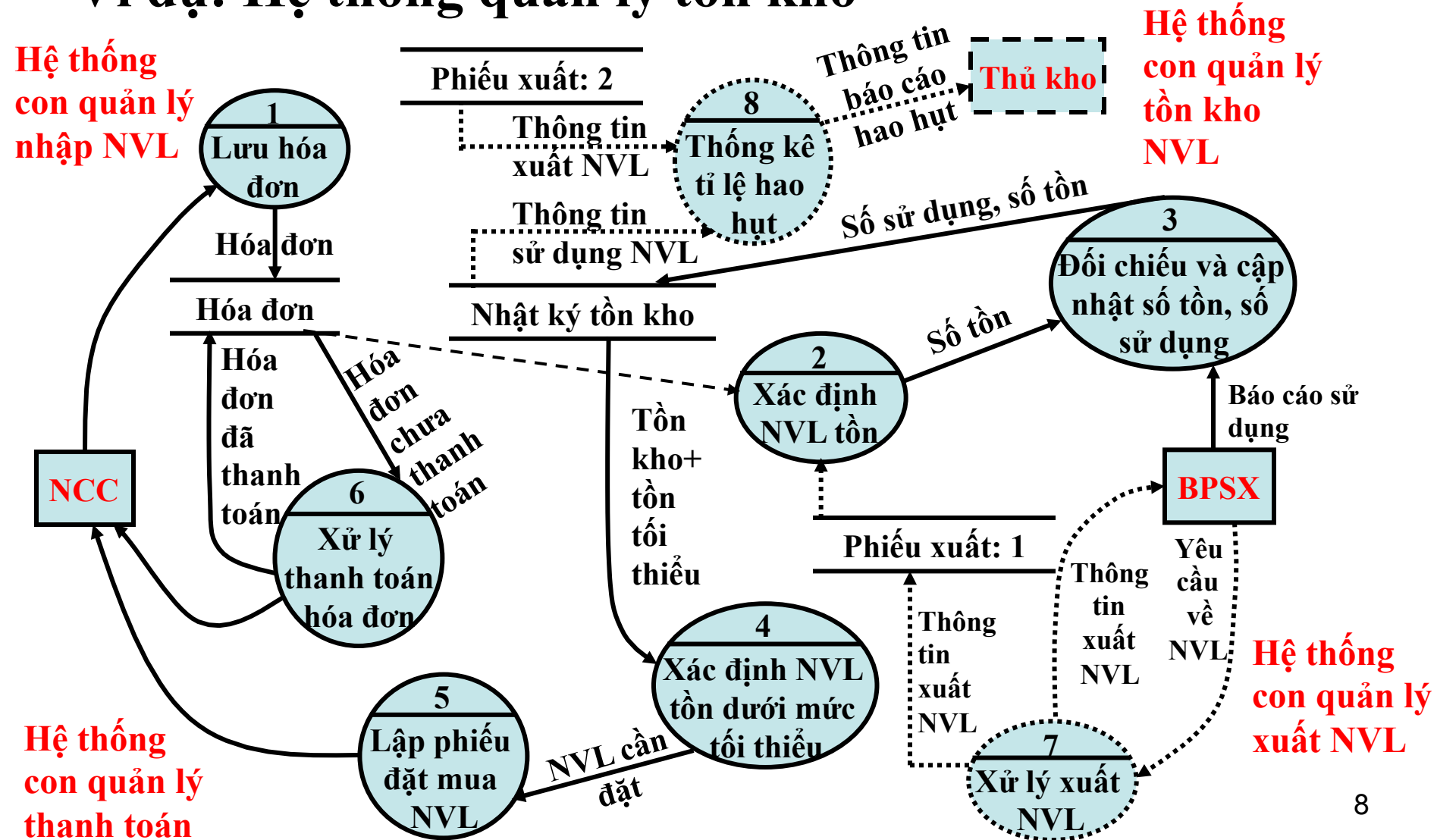
1. Phân chia hệ thống thành các hệ thống con (tt)

Các yếu tố gợi ý phân chia:

- Gom theo *thực thể*: các xử lý liên quan đến một thực thể.
 - Gom theo *sự kiện giao tác*: các xử lý đáp ứng cho một sự kiện xảy ra.
 - Gom theo *vai trò thực hiện* hoặc *vị trí tổ chức*.
- ...

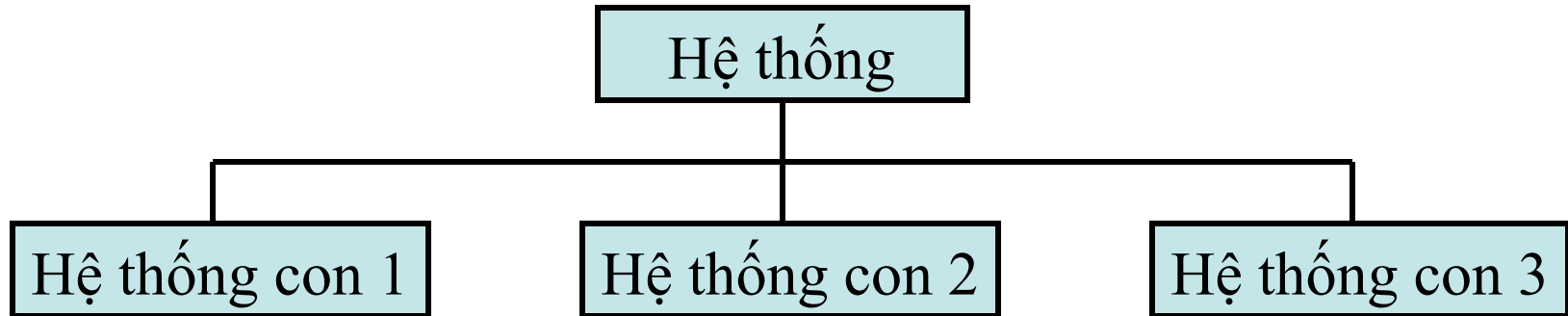
1. Phân chia hệ thống thành các hệ thống con (tt)

Ví dụ: Hệ thống quản lý tồn kho



1. Phân chia hệ thống thành các hệ thống con (tt)

Mô tả hệ thống con:



Hệ thống: ABC

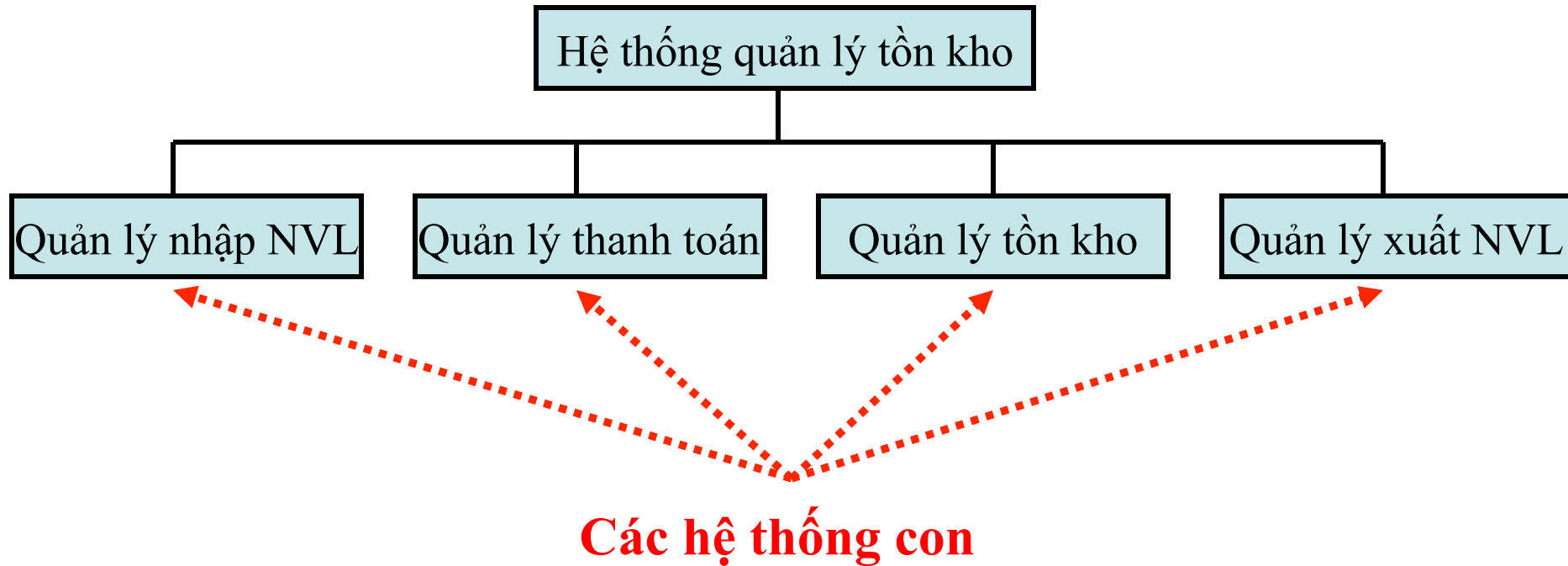
Dòng dữ liệu vào:

Dòng dữ liệu ra:

STT	Hệ thống con	Xử lý	Kho dữ liệu

1. Phân chia hệ thống thành các hệ thống con (tt)

Ví dụ: Hệ thống quản lý tồn kho và các hệ thống con



1. Phân chia hệ thống thành các hệ thống con (tt)

Hệ thống: Quản lý tồn kho NVL

Dòng dữ liệu vào:

Dòng dữ liệu ra:

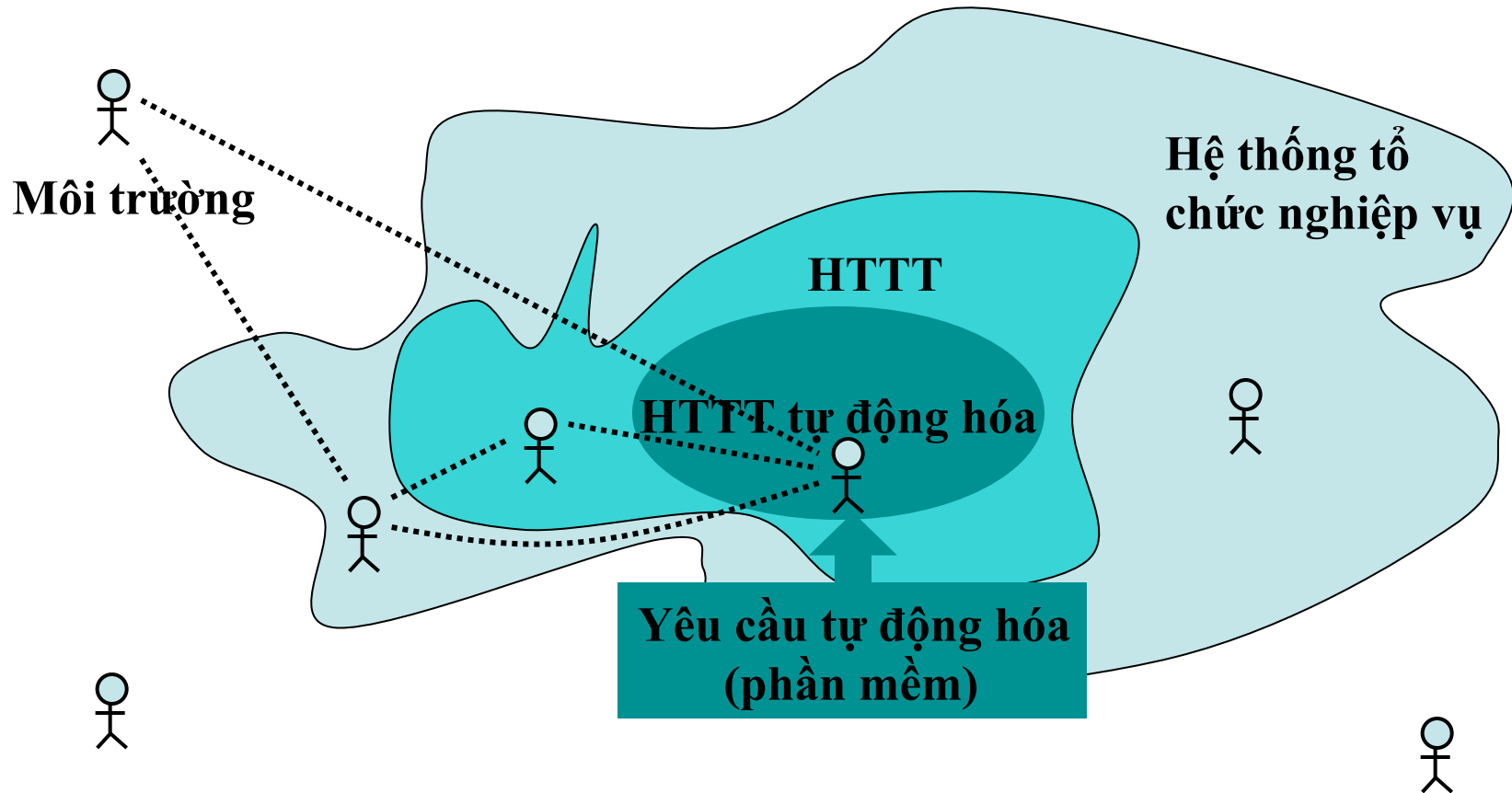
STT	Hệ thống con	Xử lý	Kho dữ liệu
1	Quản lý nhập NVL	<ul style="list-style-type: none">- Lưu hóa đơn;- Xác định NVL tồn dưới mức tối thiểu;- Lập phiếu đặt mua NVL.	<ul style="list-style-type: none">- Hóa đơn- Nhật ký tồn kho
2	Quản lý thanh toán	<ul style="list-style-type: none">- Xử lý thanh toán hóa đơn.	<ul style="list-style-type: none">- Hóa đơn
3	Quản lý tồn kho	<ul style="list-style-type: none">- Xác định NVL tồn;- Đối chiếu và cập nhật số lượng tồn, số sử dụng;- Thống kê tỉ lệ hao hụt.	<ul style="list-style-type: none">- Hóa đơn, phiếu xuất- Nhật ký tồn kho- Phiếu xuất, Nhật ký tồn kho
4	Quản lý xuất NVL	<ul style="list-style-type: none">- Xử lý xuất NVL.	<ul style="list-style-type: none">- Phiếu xuất

2. Xây dựng mô hình thiết kế xử lý

- **Phân chia hoạt động thủ công và tự động**
- **Thiết kế các vai trò**
- **Thiết kế xử lý trực tuyến – theo lô**
- **Mô hình hóa xử lý mức thiết kế**

2. Xây dựng mô hình thiết kế xử lý (tt)

- Phân chia hoạt động thủ công và tự động:

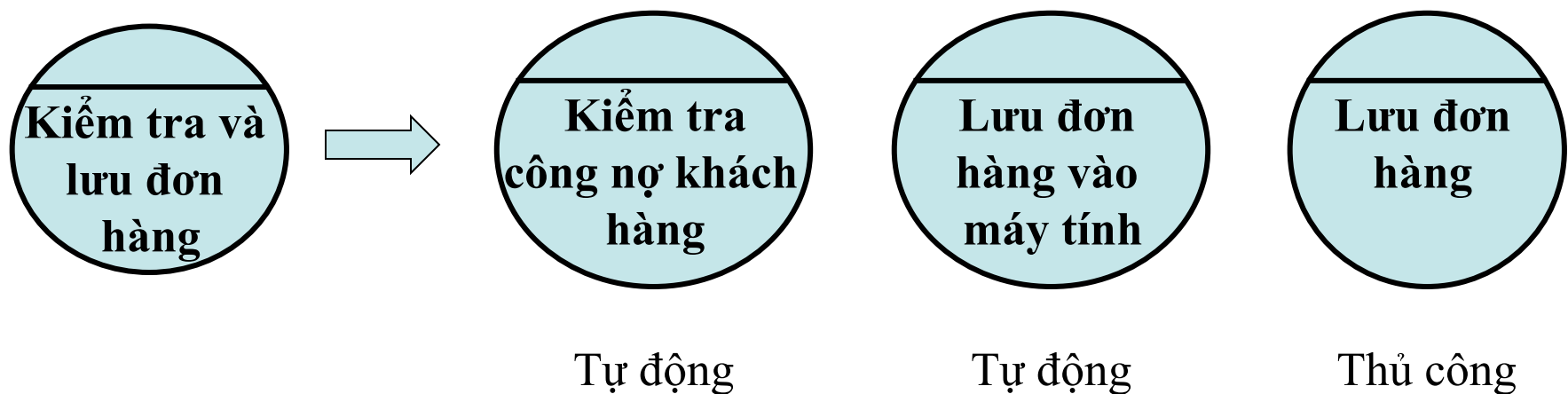


2. Xây dựng mô hình thiết kế xử lý (tt)

- **Phân chia hoạt động thủ công và tự động (tt)**

Nguyên tắc phân chia:

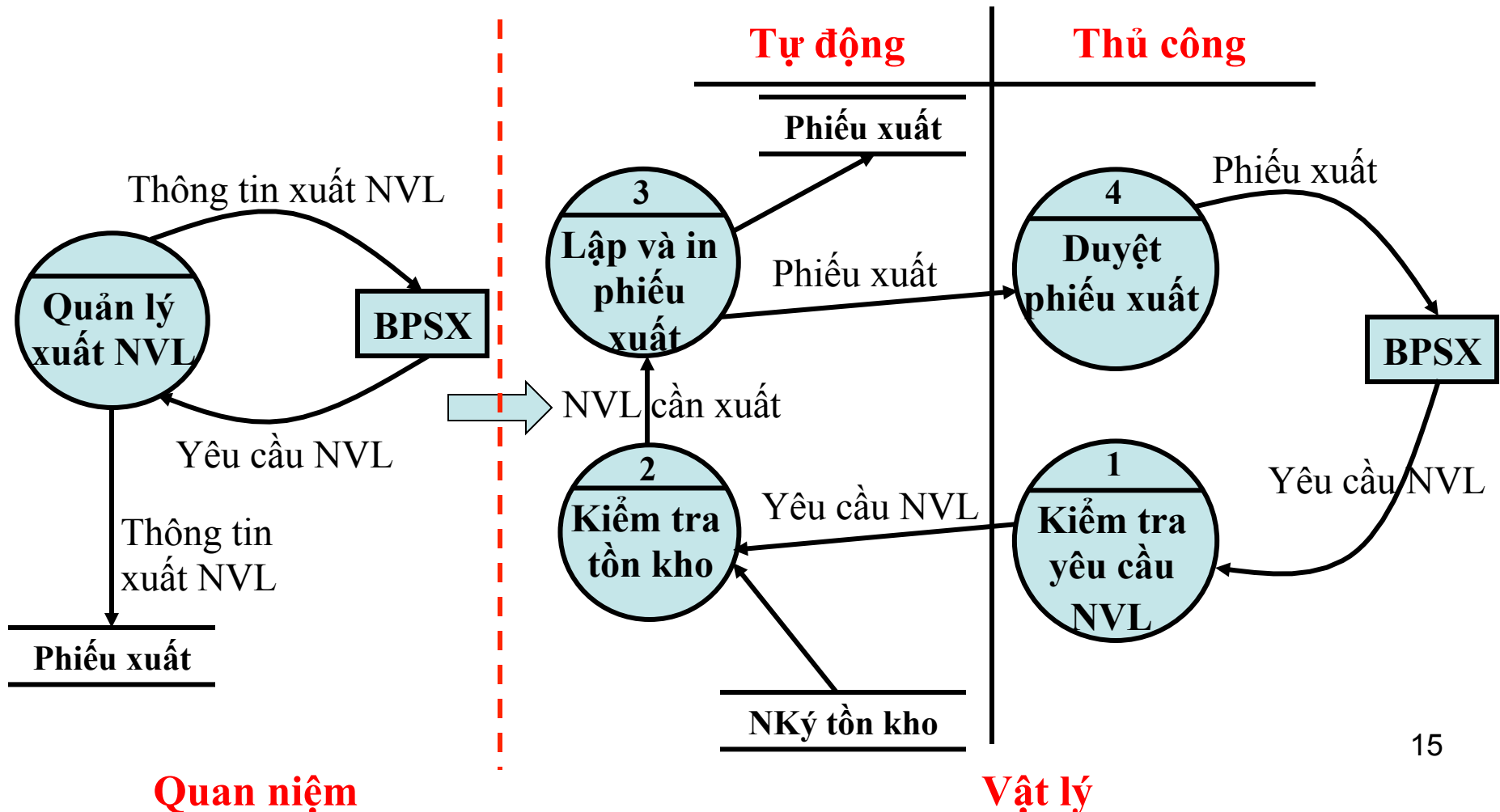
- Các xử lý sau khi phân chia sẽ hoàn toàn thủ công hoặc hoàn toàn tự động.
- Sự phân chia chấp nhận việc trùng lặp nội dung giữa tự động và thủ công.



2. Xây dựng mô hình thiết kế xử lý (tt)

- Phân chia hoạt động thủ công và tự động (tt)

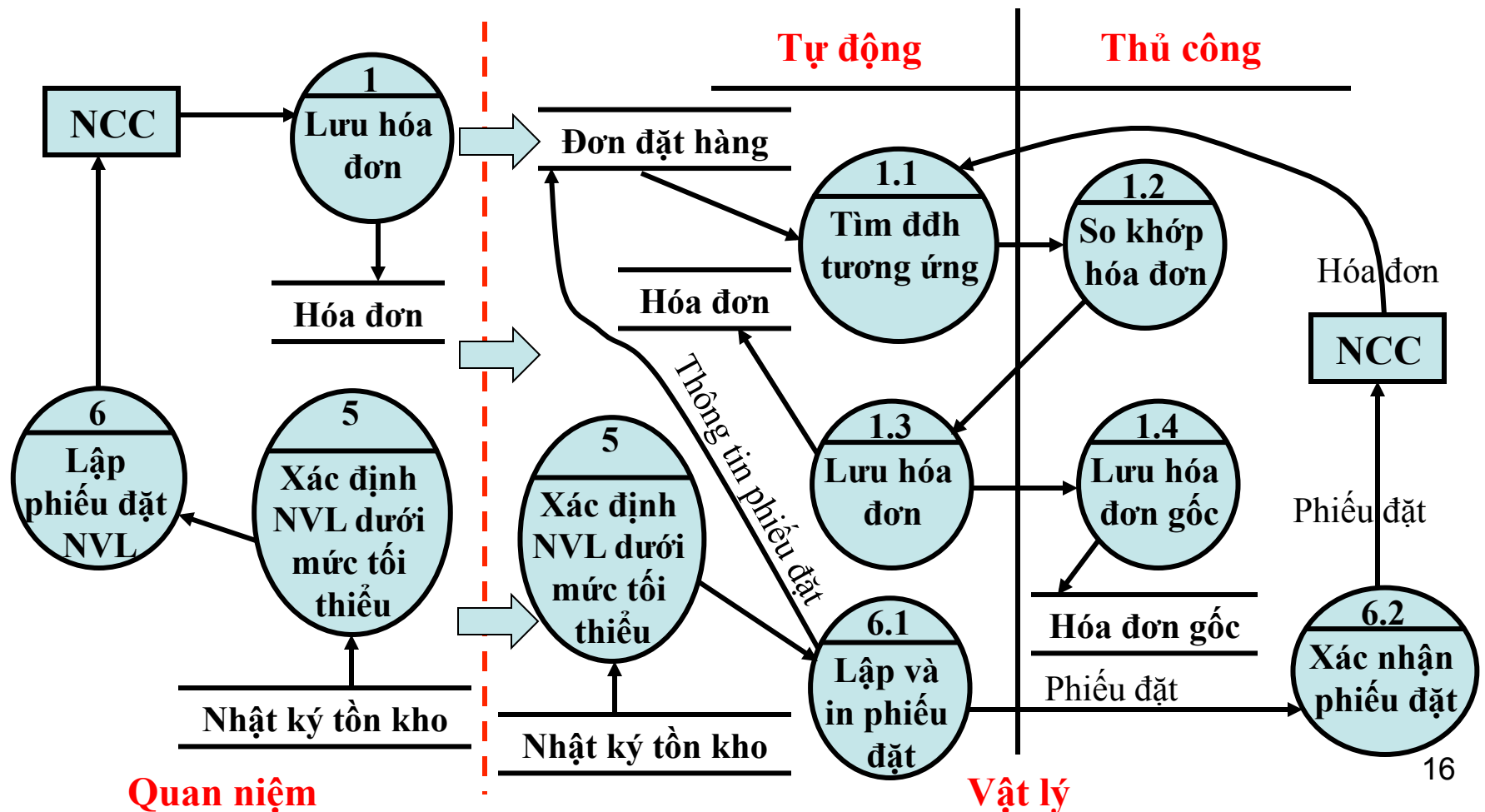
Ví dụ: Quản lý xuất NVL



2. Xây dựng mô hình thiết kế xử lý (tt)

- Phân chia hoạt động thủ công và tự động (tt)

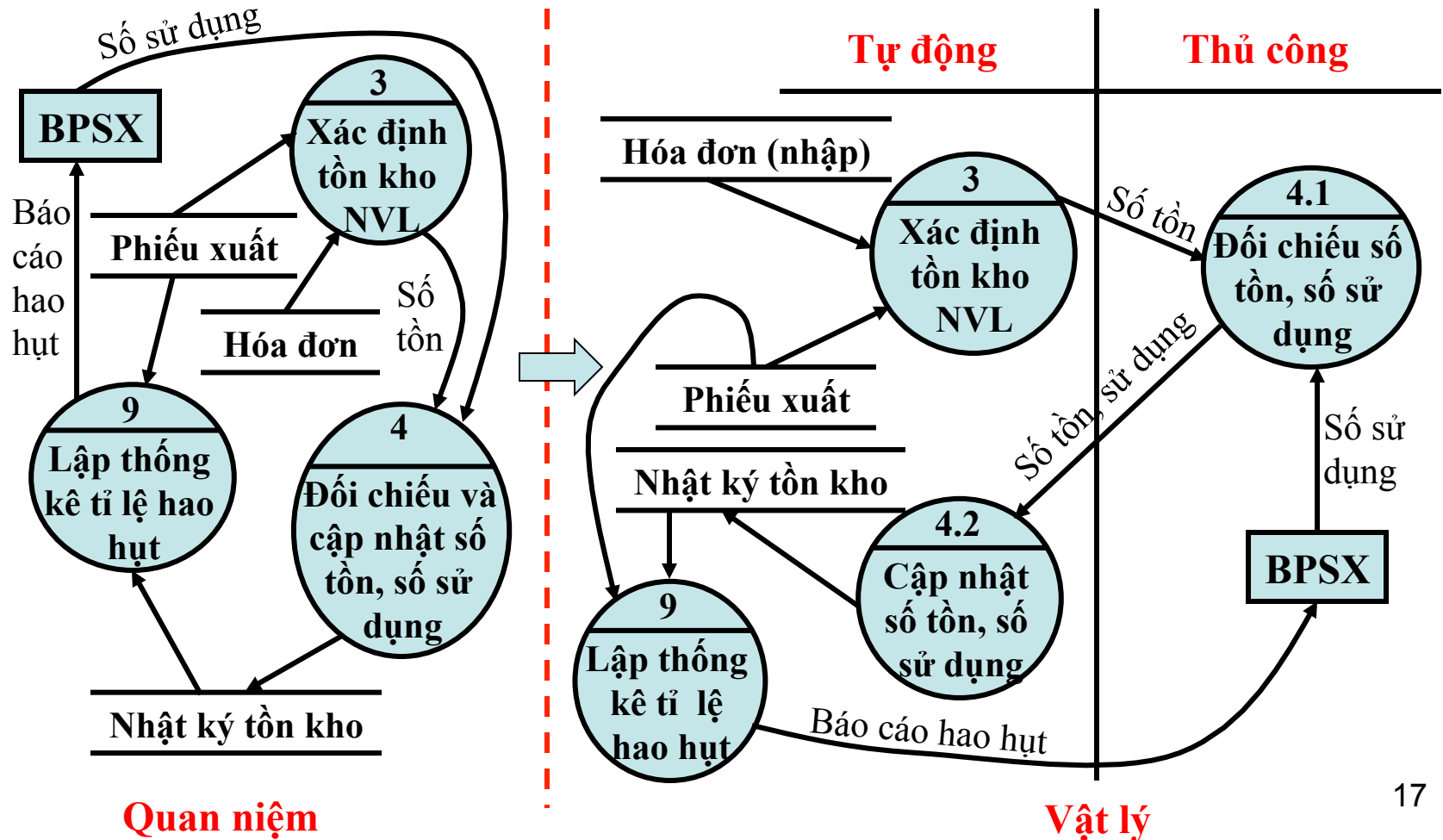
Ví dụ: Quản lý nhập NVL



2. Xây dựng mô hình thiết kế xử lý (tt)

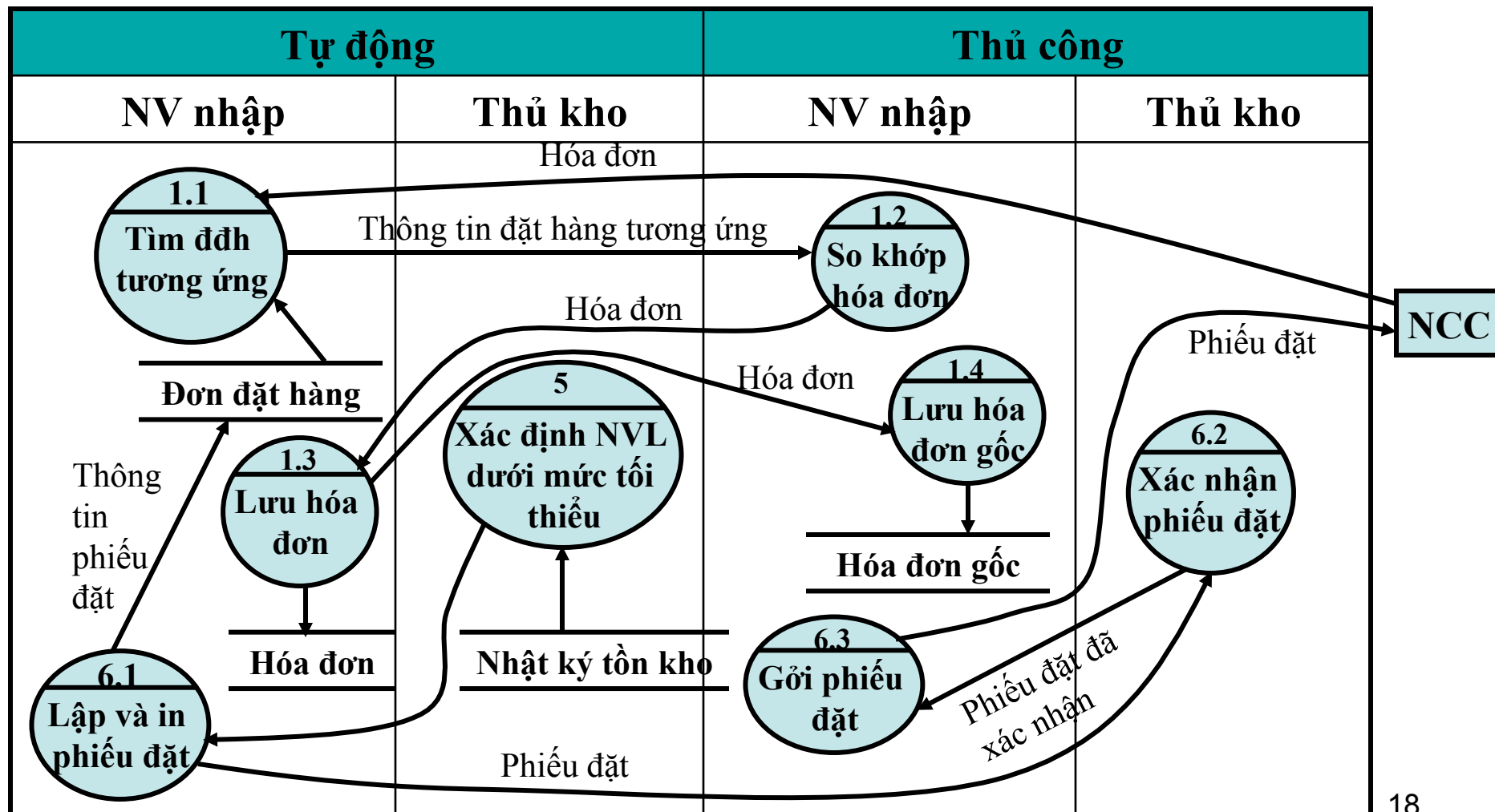
- Phân chia hoạt động thủ công và tự động (tt)

Ví dụ: Quản lý Tồn kho



2. Xây dựng mô hình thiết kế xử lý (tt)

- Thiết kế các vai trò – Ví dụ: Quản lý nhập NVL



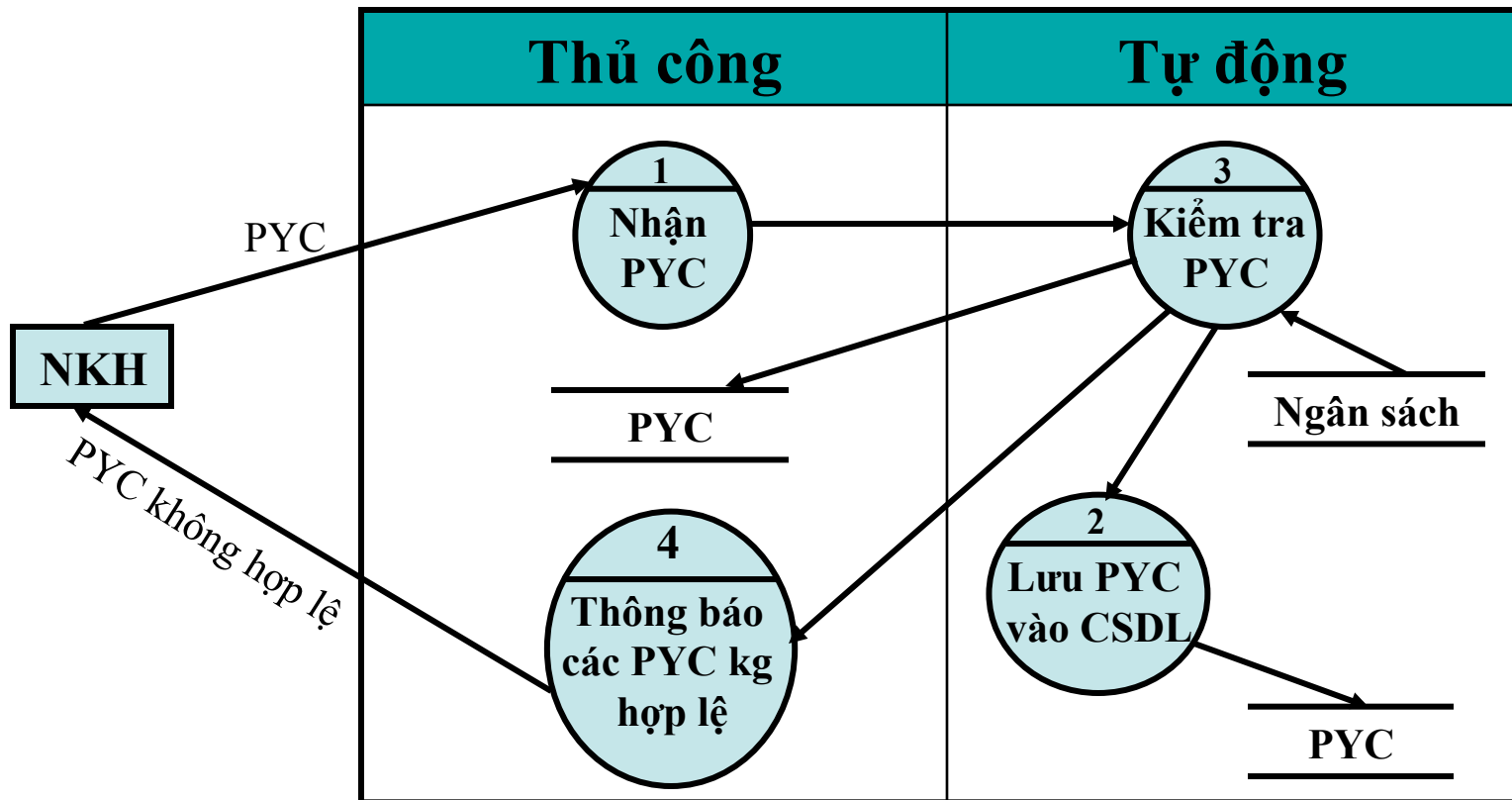
2. Xây dựng mô hình thiết kế xử lý (tt)

- **Thiết kế xử lý trực tuyến:**
 - Xử lý trực tuyến (on-line processing): là các xử lý thu thập và phân phối thông tin mới nhất về hệ thống thông qua một trạm làm việc trực tuyến.
 - Xử lý trực tuyến thường được thiết kế khi:
 - Truy cập và nắm bắt những thông tin xảy ra một cách ngẫu nhiên.
 - Định dạng và kiểu thông tin là không nhất quán.
 - Thông tin đang tiếp tục thay đổi và thông tin mới nhất là cần thiết cho xử lý hiện tại và hỗ trợ ra quyết định.
 - Người dùng ở vị trí dễ dàng truy cập tới HTTT.

2. Xây dựng mô hình thiết kế xử lý (tt)

- Thiết kế xử lý trực tuyến (tt)

Ví dụ: Quản lý yêu cầu sách của Nhà khoa học



2. Xây dựng mô hình thiết kế xử lý (tt)

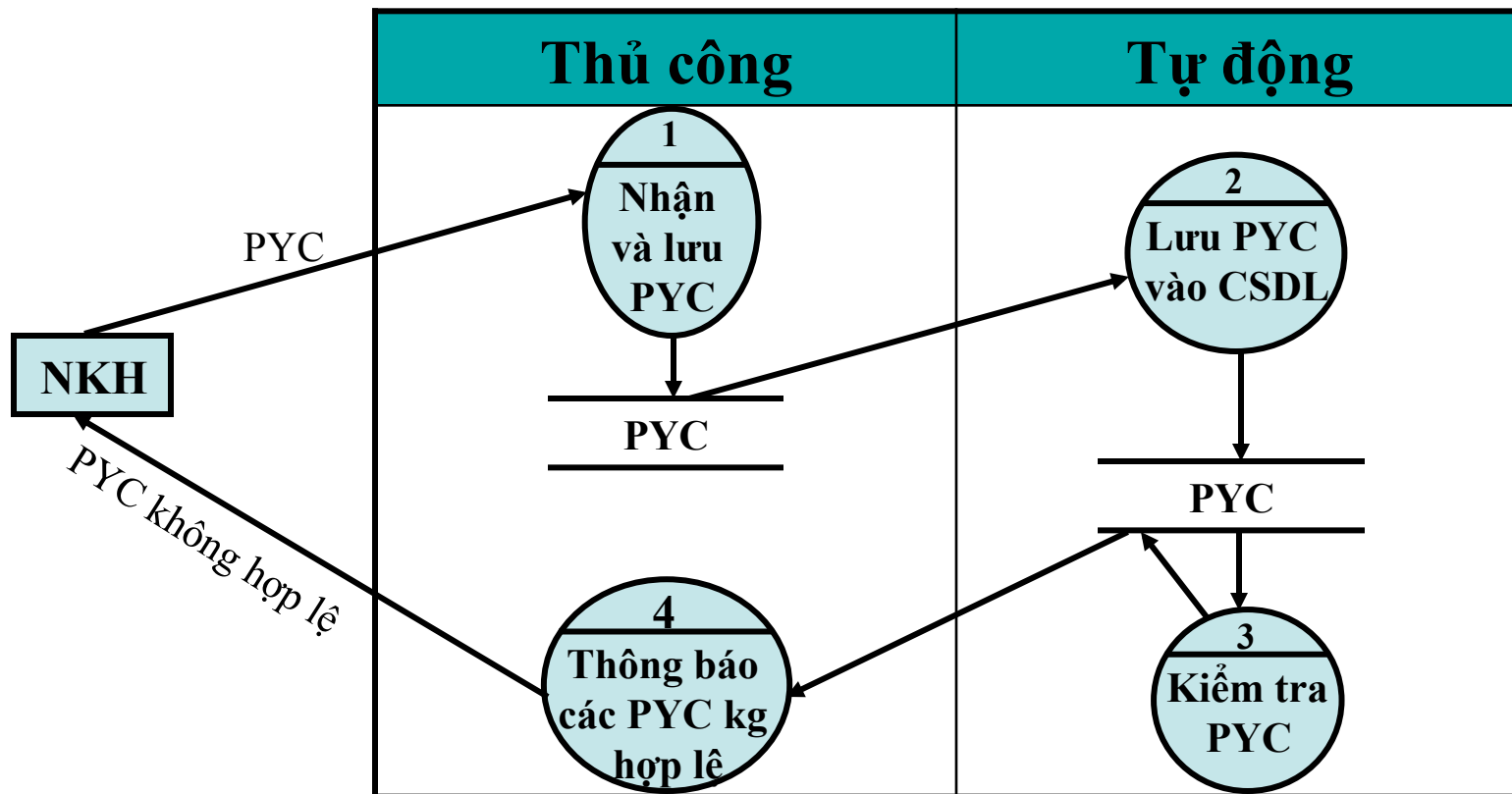
- **Thiết kế xử lý theo lô:**

- Xử lý theo lô (batch processing): việc xử lý đầu vào và đầu ra hệ thống theo một thời điểm xác định.
- Xử lý theo lô thường được thiết kế khi:
 - Truy cập thông tin có định kỳ.
 - Định dạng và kiểu thông tin là nhất quán.
 - Thông tin ổn định trong một khoảng thời gian hoặc nhu cầu thông tin của người dùng không cần phải mới nhất.
 - Người dùng không ở vị trí có thể truy cập HTTT một cách trực tuyến.

2. Xây dựng mô hình thiết kế xử lý (tt)

- Thiết kế xử lý theo lô (tt)

Ví dụ: Quản lý yêu cầu sách của Nhà khoa học



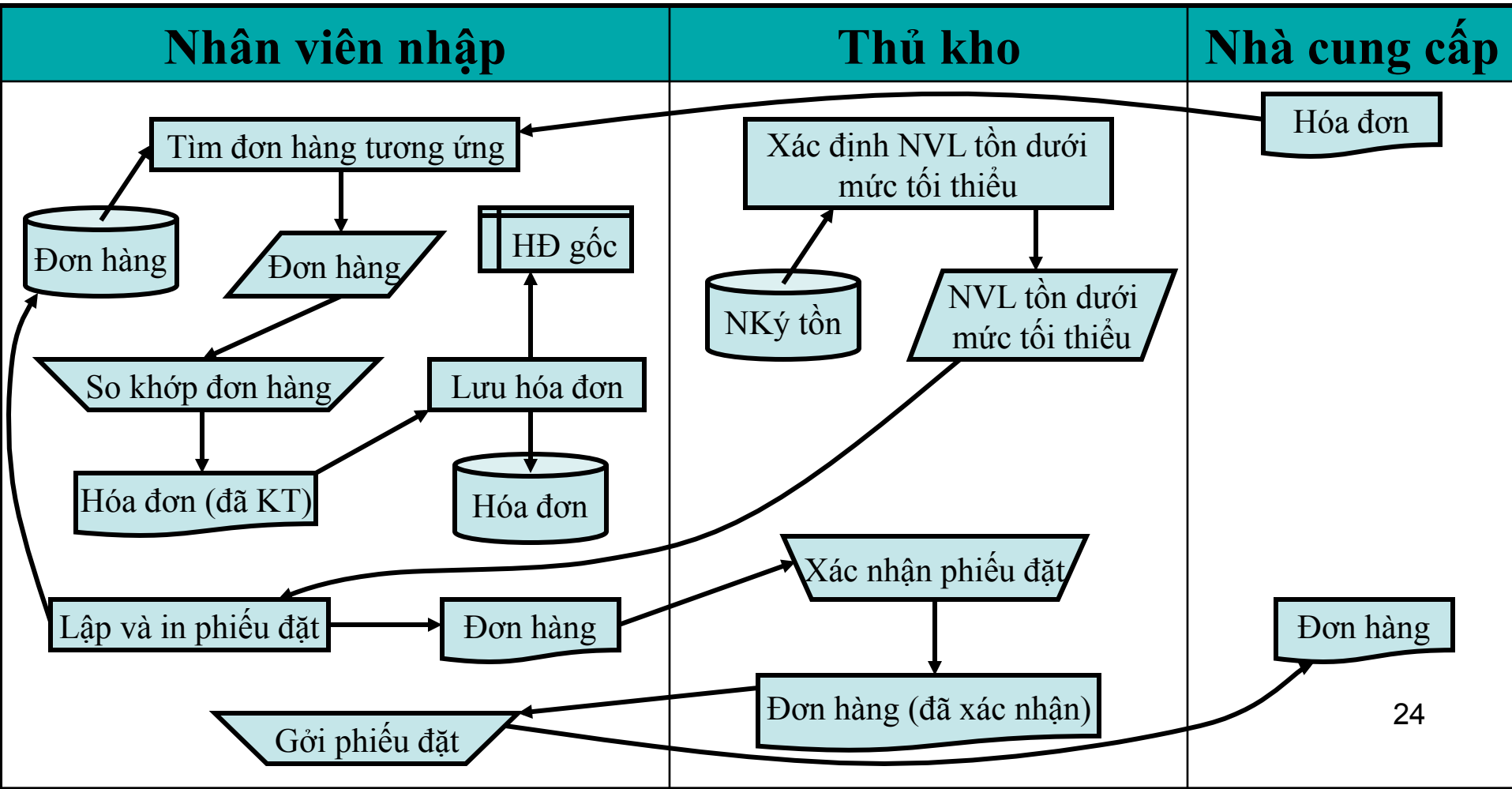
2. Xây dựng mô hình thiết kế xử lý (tt)

- **Mô hình hóa xử lý mức thiết kế:**
 - **Mục tiêu:**
 - Biểu diễn các xử lý của hệ thống trong một môi trường cụ thể.
 - Làm nổi bật các yếu tố vật lý của hệ thống: hình thức, không gian, thời gian, tổ chức, thủ công – tự động, ...
 - **Các mô hình có thể dùng:**
 - Mô hình DFD (mức vật lý)
 - Flowchart
 - Mô hình xử lý tựa Merise

2. Xây dựng mô hình thiết kế xử lý (tt)

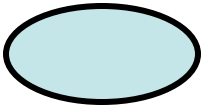
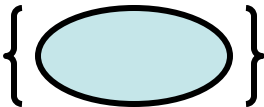
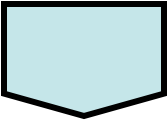

- **Mô hình hóa xử lý mức thiết kế (tt)**

Ví dụ: Flowchart (Quản lý nhập NVL)



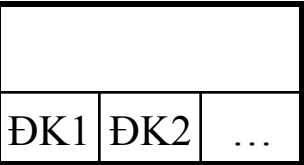
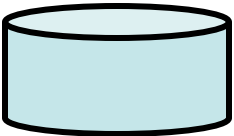

2. Xây dựng mô hình thiết kế xử lý (tt)

- **Mô hình hóa xử lý mức thiết kế (tt)**
 - **Mô hình xử lý tựa Merise – Ký hiệu:**

Ký hiệu	Tên gọi
	Biến cố. Gồm hai loại: biến cố vào và biến cố ra. Biến cố vào gồm hai loại: biến cố phát động và biến cố điều kiện.
	Tập các biến cố.
	Điều kiện phát động biến cố.
	Quy tắc quản lý (QTQL).

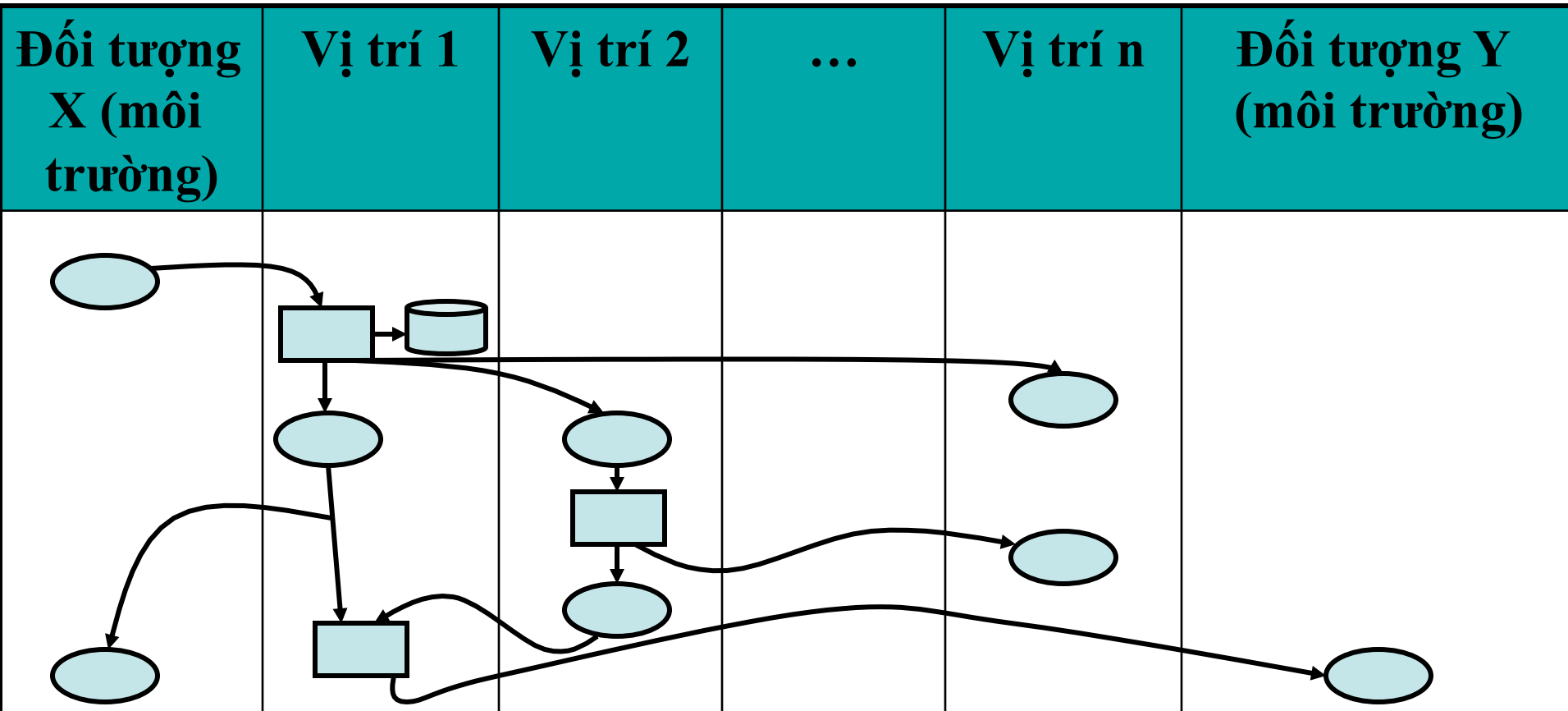
2. Xây dựng mô hình thiết kế xử lý (tt)

- **Mô hình hóa xử lý mức thiết kế (tt)**
 - **Mô hình xử lý tựa Merise – Ký hiệu (tt)**

Ký hiệu	Tên gọi
	Quy tắc quản lý có điều kiện ra.
	Dữ liệu. Gồm hai loại: dữ liệu vào và dữ liệu ra.
	Phương tiện biểu diễn dữ liệu trên giấy.

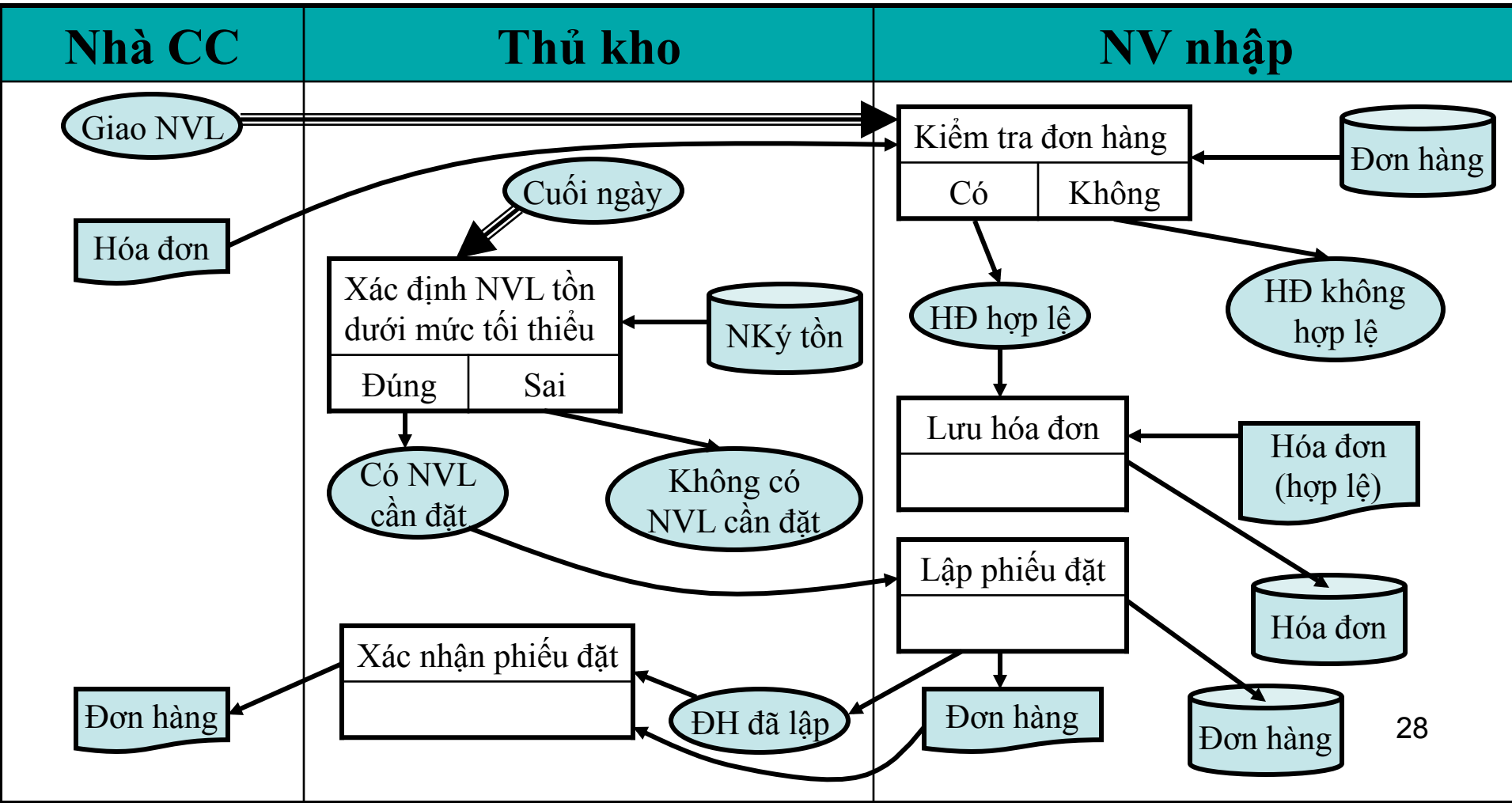
2. Xây dựng mô hình thiết kế xử lý (tt)

- Mô hình hóa xử lý mức thiết kế (tt)
 - Mô hình xử lý tựa Merise (tt)



2. Xây dựng mô hình thiết kế xử lý (tt)

- Mô hình hóa xử lý mức thiết kế (tt)
 - Mô hình xử lý tựa Merise – Ví dụ:



2. Xây dựng mô hình thiết kế xử lý (tt)

- **Mô hình hóa xử lý mức thiết kế (tt)**
 - **Mô hình xử lý tựa Merise – Bảng mô tả chức năng:**

STT	Nguồn gốc QTQL (1)	Thủ tục chức năng (2)	Bản chất (3)	Vị trí (4)	Thời gian thực hiện (5)

(1) Các thủ tục chức năng được lấy từ nguồn gốc QTQL nào.

(2) Thứ tự các công việc trong qui trình xử lý.

(3) Bản chất công việc được thực hiện là thủ công hay tự động.

(4) Thủ tục chức năng đó được thực hiện ở vị trí nào.

(5) Khi nào thực hiện công việc đó (thực hiện ngay, cuối ngày, ...).

2. Xây dựng mô hình thiết kế xử lý

- **Mô hình hóa xử lý mức thiết kế (tt)**

- **Mô hình xử lý tựa Merise – Bảng mô tả chức năng – Ví dụ:**

STT	Nguồn gốc QTQL	Thủ tục chức năng	Bản chất	Vị trí	Thời gian thực hiện
1	Kiểm tra đơn hàng	Xác định đơn hàng của hóa đơn	Tự động	NV nhập	Thực hiện ngay
2	Kiểm tra đơn hàng	So khớp với NVL nhập	Thủ công	NV nhập	Thực hiện ngay
3	Lưu hóa đơn	Lưu hóa đơn	Tự động	NV nhập	Thực hiện ngay
4	Xác định NVL tồn dưới mức tối thiểu	Xác định NVL tồn dưới mức tối thiểu	Tự động	Thủ kho	Cuối ngày
5	Lập đặt mua NVL	Lập đặt mua NVL	Tự động	NV nhập	Cuối ngày
6	Lập đặt mua NVL	In phiếu đặt	Tự động	NV nhập	Cuối ngày
7	Xác nhận phiếu đặt	Xác nhận phiếu đặt	Thủ công	Thủ kho	Cuối ngày

3. Thiết kế chức năng phần mềm

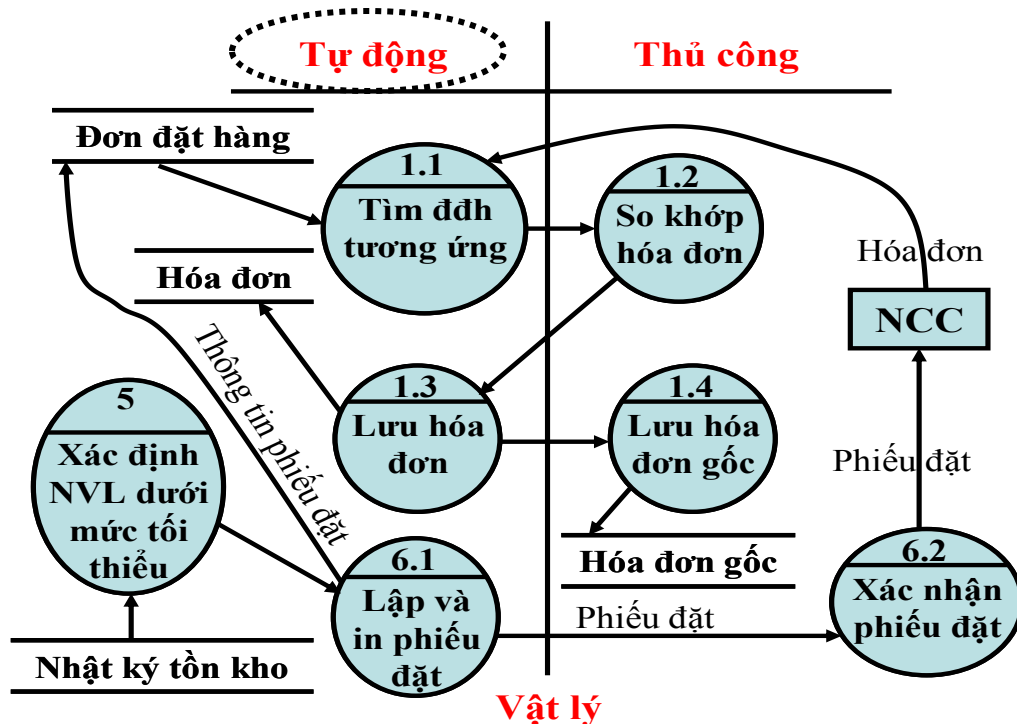
- **Xác định các chức năng phần mềm**
- **Xây dựng cấu trúc chức năng phần mềm**
- **Kiến trúc phần mềm**
- **Thiết kế chức năng phần mềm theo kiến trúc 3 tầng**
- **Các mục tiêu trong thiết kế hệ thống**

3. Thiết kế chức năng phần mềm (tt)

- **Xác định các chức năng phần mềm:**

Từ các *xử lý tự động hóa*: các xử lý được quyết định là tự động hóa trong sơ đồ vật lý sẽ là ứng viên của chức năng phần mềm: **1/n xử lý \Leftrightarrow 1/n chức năng phần mềm**

Ví dụ: Quản lý nhập NVL



Chức năng phần mềm:

Tìm đơn đặt hàng

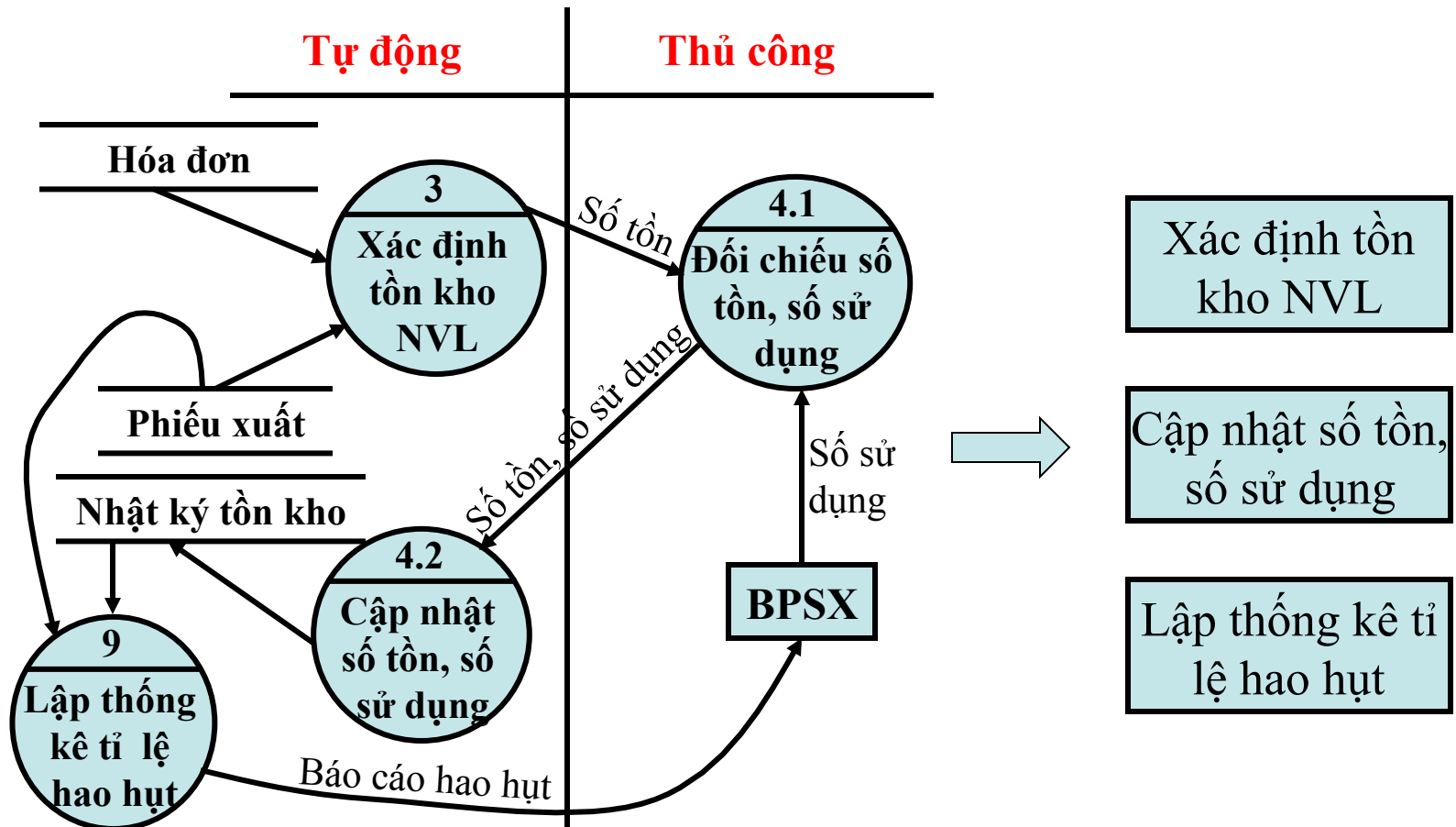
Lưu hóa đơn

Lập và in phiếu đặt

Xác định NVL tồn dưới mức tối thiểu

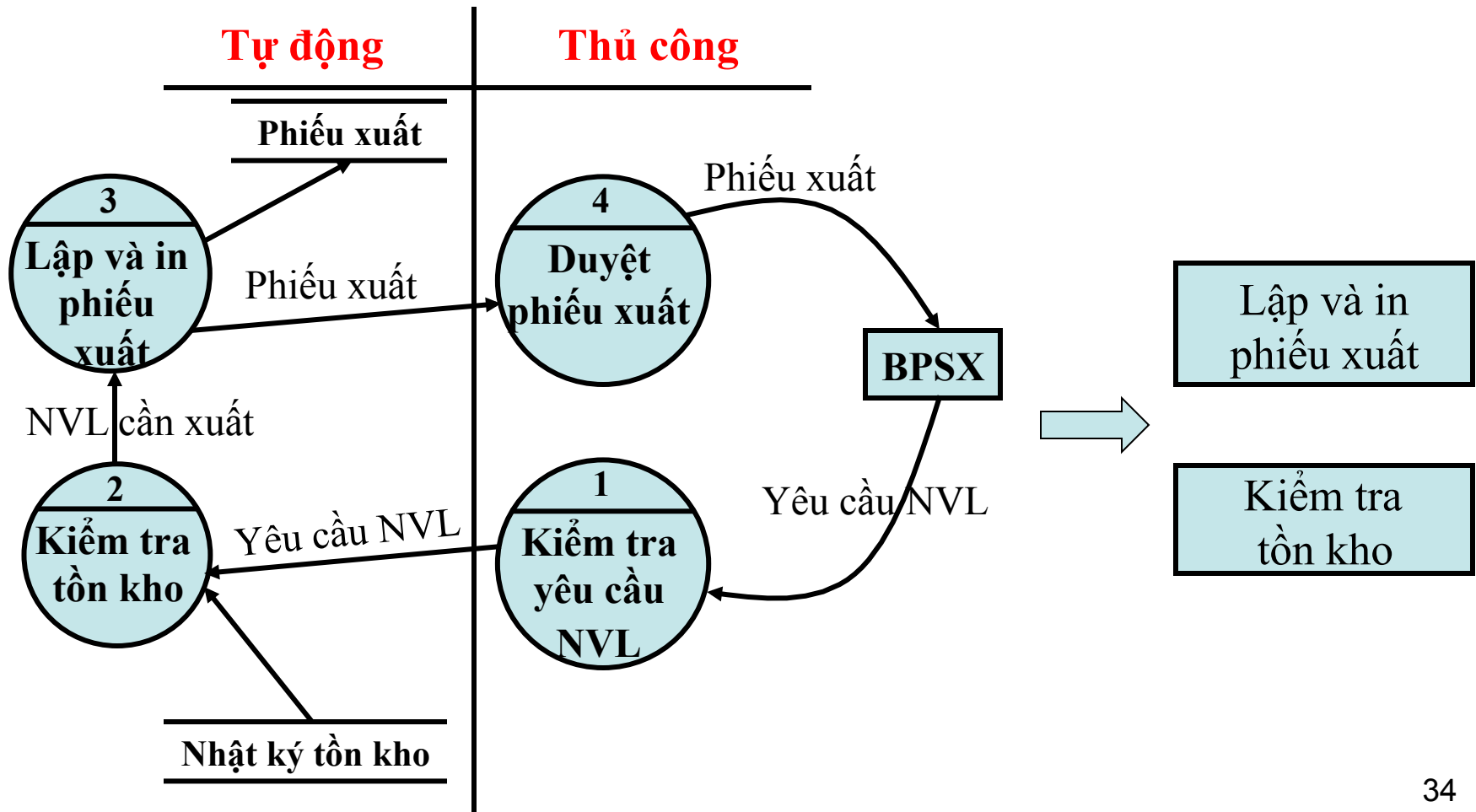
3. Thiết kế chức năng phần mềm (tt)

- **Xác định các chức năng phần mềm (tt)**
Ví dụ: Quản lý Tồn kho



3. Thiết kế chức năng phần mềm (tt)

- **Xác định các chức năng phần mềm (tt)**
Ví dụ: Quản lý xuất NVL



3. Thiết kế chức năng phần mềm (tt)

- **Xác định các chức năng phần mềm (tt)**
Ví dụ: Hệ thống Quản lý tồn kho NVL

Tìm đơn hàng

Lưu hóa đơn

Lập và in phiếu đặt

Xác định NVL tồn
dưới mức tối thiểu

Xác định tồn kho
NVL

Cập nhật số tồn, số
sử dụng

Lập thống kê tỉ lệ
hao hụt

Lập và in phiếu
xuất

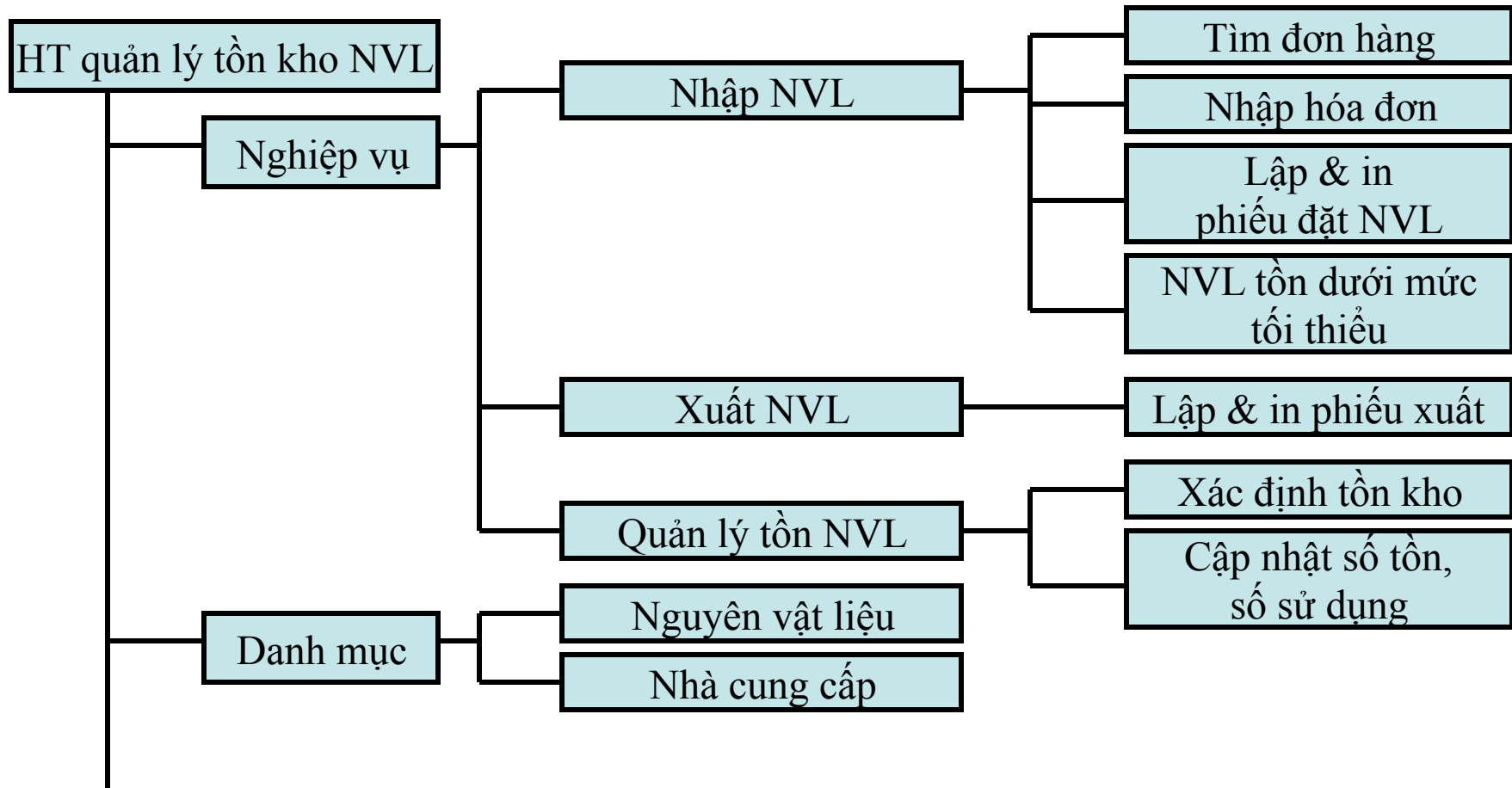
Kiểm tra tồn kho

3. Thiết kế chức năng phần mềm (tt)

- **Xác định các chức năng phần mềm (tt)**
 - Các chức năng quản trị danh mục: NCC, NVL, ...
 - Các chức năng hệ thống:
 - An toàn:
 - + Quản lý người dùng
 - + Backup/restore DB
 - + Mã hóa
 - Thông số hệ thống: thư mục, ngày/biến hệ thống, ...
 - Các chức năng khai thác dữ liệu:
 - Tìm kiếm
 - Thống kê, báo cáo, ...
 - Các tiện ích: máy tính, lịch, forum, ...
 - Hướng dẫn sử dụng.

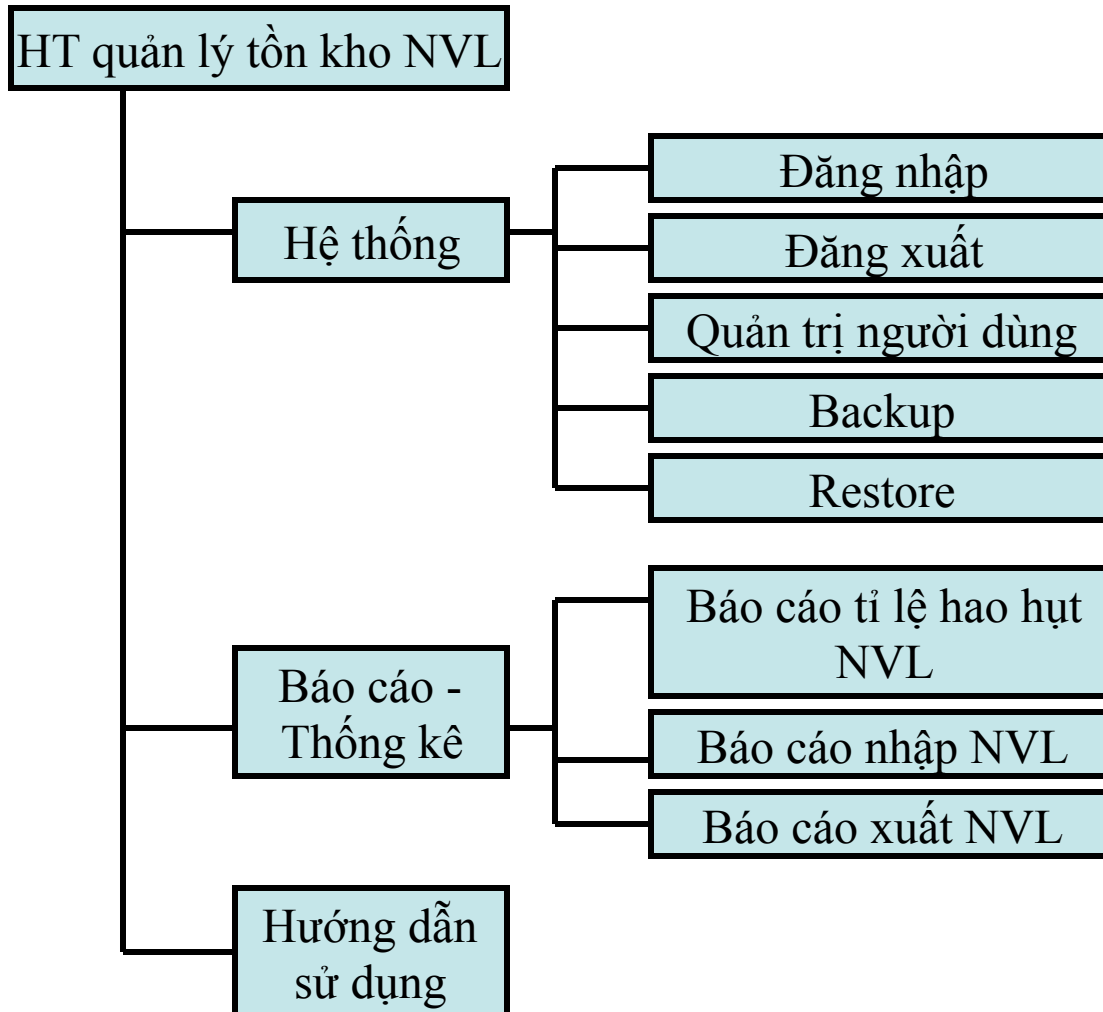
3. Thiết kế chức năng phần mềm (tt)

- Xây dựng cấu trúc chức năng phần mềm – Cách 1:



3. Thiết kế chức năng phần mềm (tt)

- Xây dựng cấu trúc chức năng phần mềm – Cách 1 (tt)



3. Thiết kế chức năng phần mềm (tt)

- Xây dựng cấu trúc chức năng phần mềm – Cách 2:
Dùng biểu đồ cấu trúc (Structured chart):

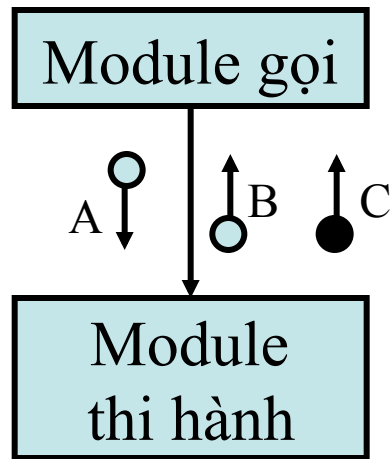
B1: Phân chia thành các module.

- Module: là một đơn vị của hệ thống được xác định bởi chức năng của nó, tất cả các lệnh trong module đều nhằm thực hiện chức năng đó.
- Module có thể là một đoạn chương trình, một thủ tục/hàm, một form, menu, ...

B2: Xác định mối quan hệ giữa các module và thành lập biểu đồ.

3. Thiết kế chức năng phần mềm (tt)

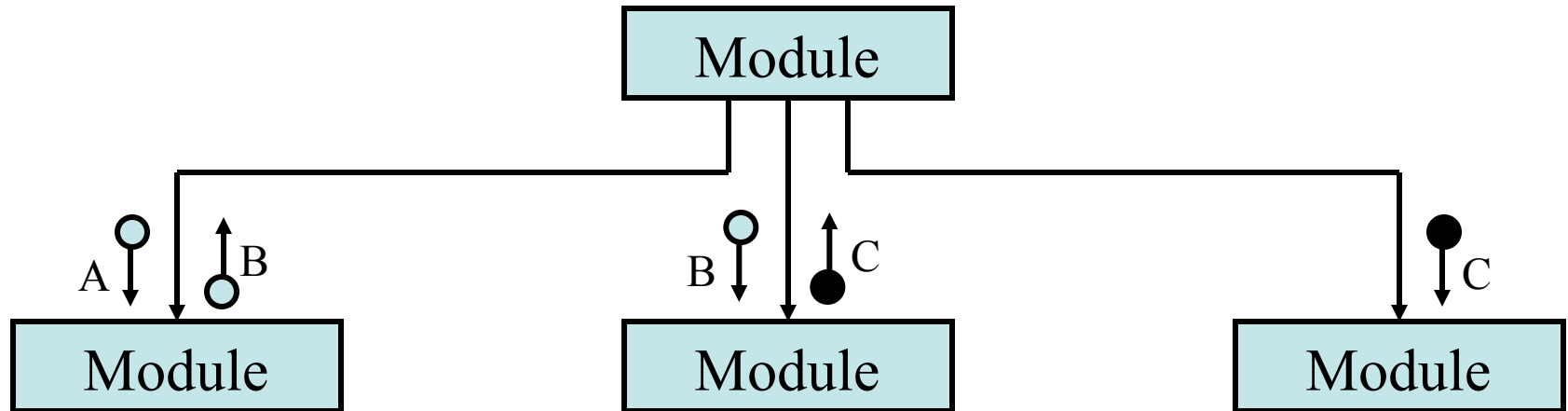
- Xây dựng cấu trúc chức năng phần mềm – Dùng biểu đồ cấu trúc (tt)
Ký hiệu:



A – giá trị tham số truyền
B – giá trị kết quả nhận
C – cờ hiệu

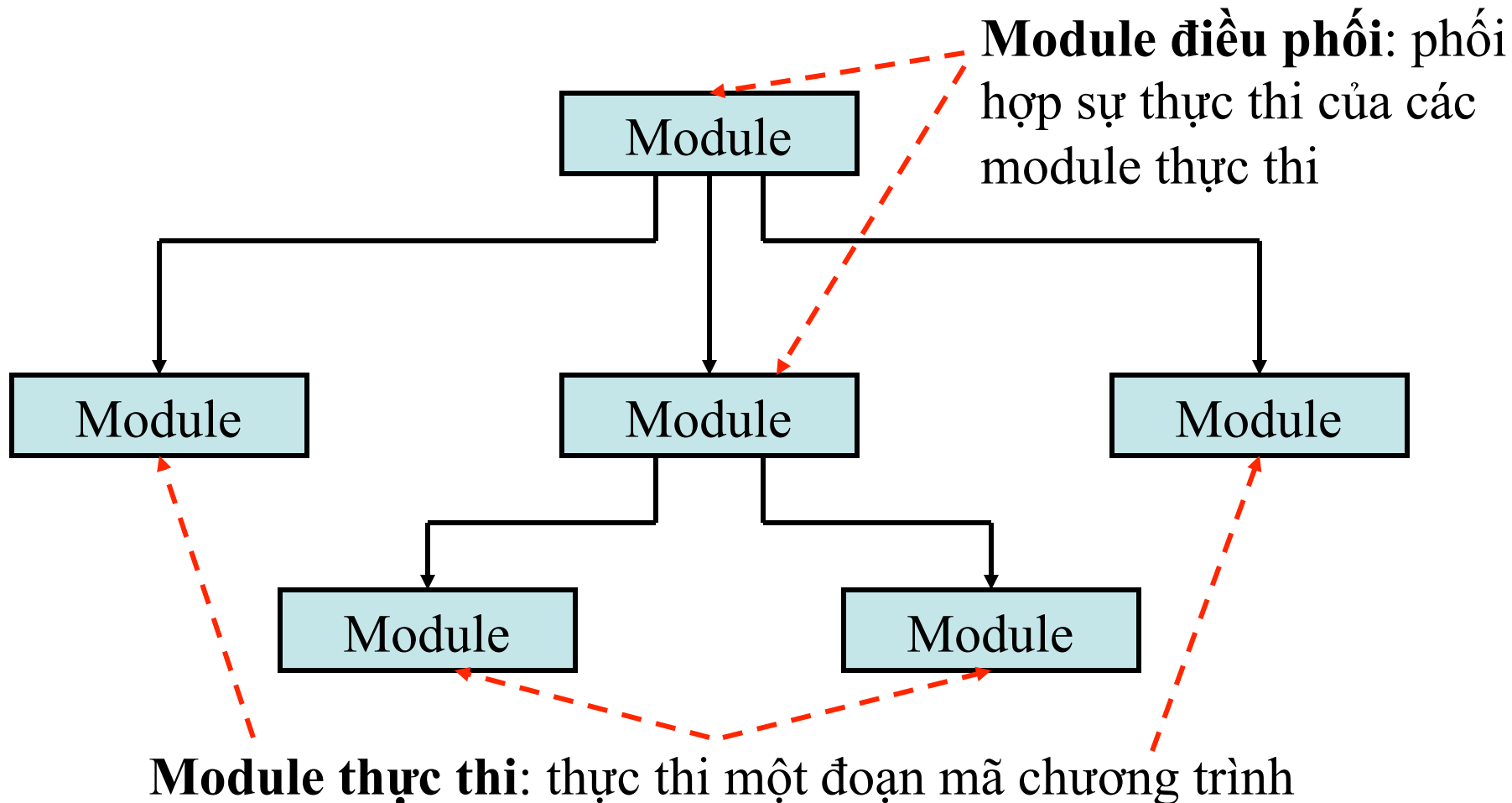
3. Thiết kế chức năng phần mềm (tt)

- Xây dựng cấu trúc chức năng phần mềm – Dùng biểu đồ cấu trúc (tt)



3. Thiết kế chức năng phần mềm (tt)

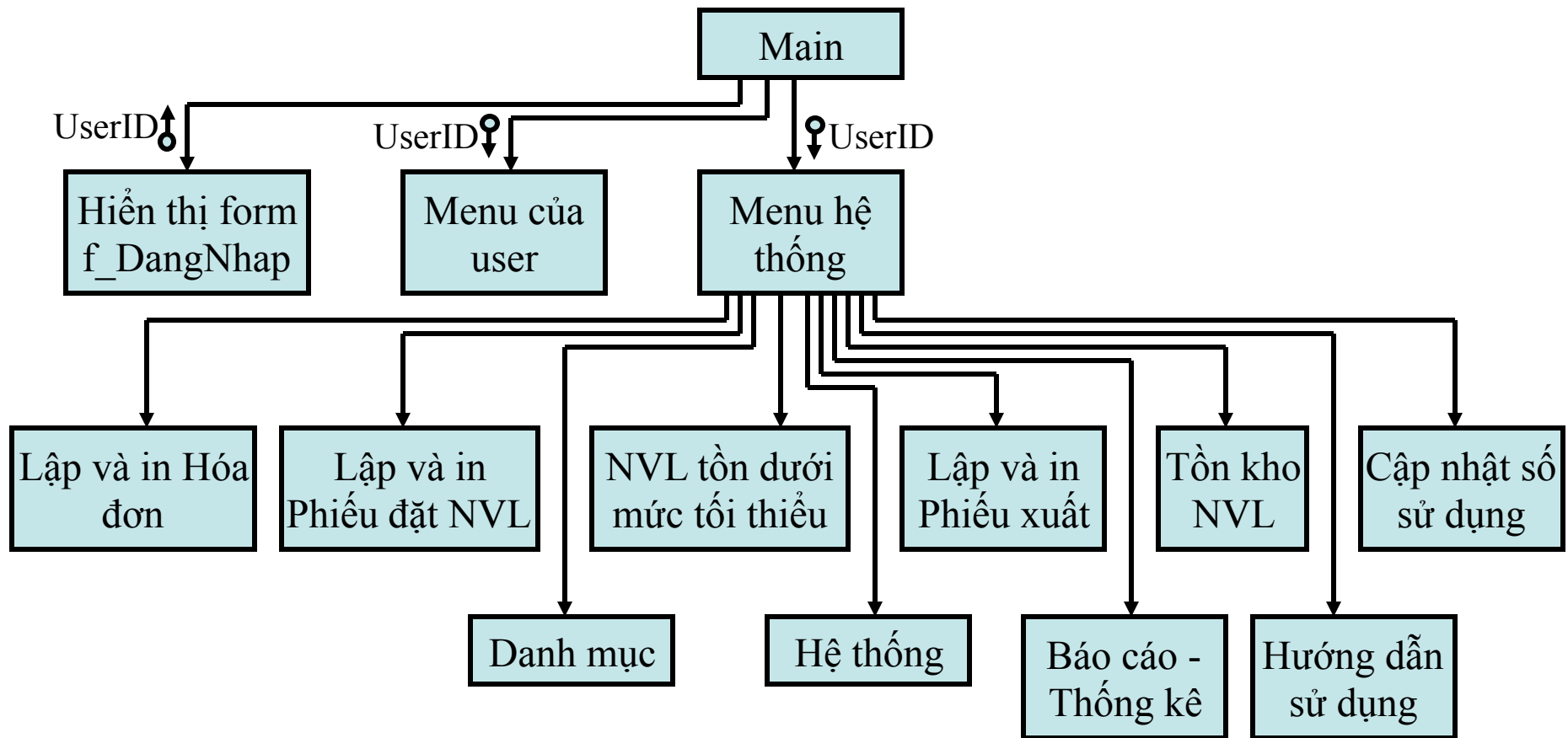
- Xây dựng cấu trúc chức năng phần mềm – Dùng biểu đồ cấu trúc (tt)



3. Thiết kế chức năng phần mềm (tt)

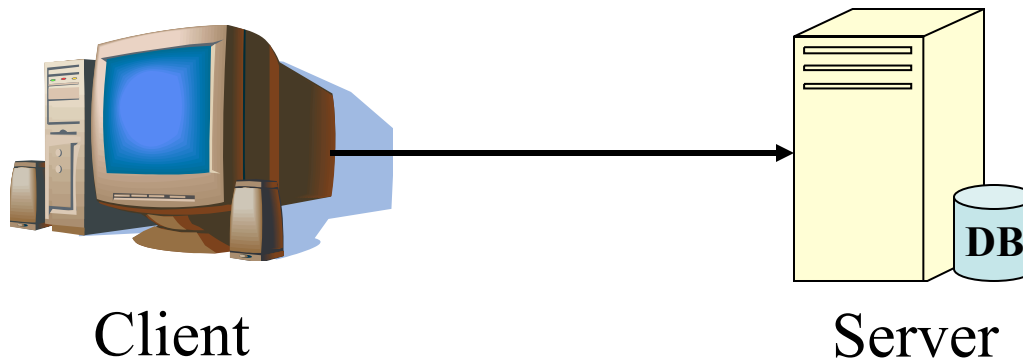
- Xây dựng cấu trúc chức năng phần mềm – Dùng biểu đồ cấu trúc (tt)

Ví dụ:



3. Thiết kế chức năng phần mềm (tt)

- **Kiến trúc phần mềm:**
 - 1) **Kiến trúc client-server:**



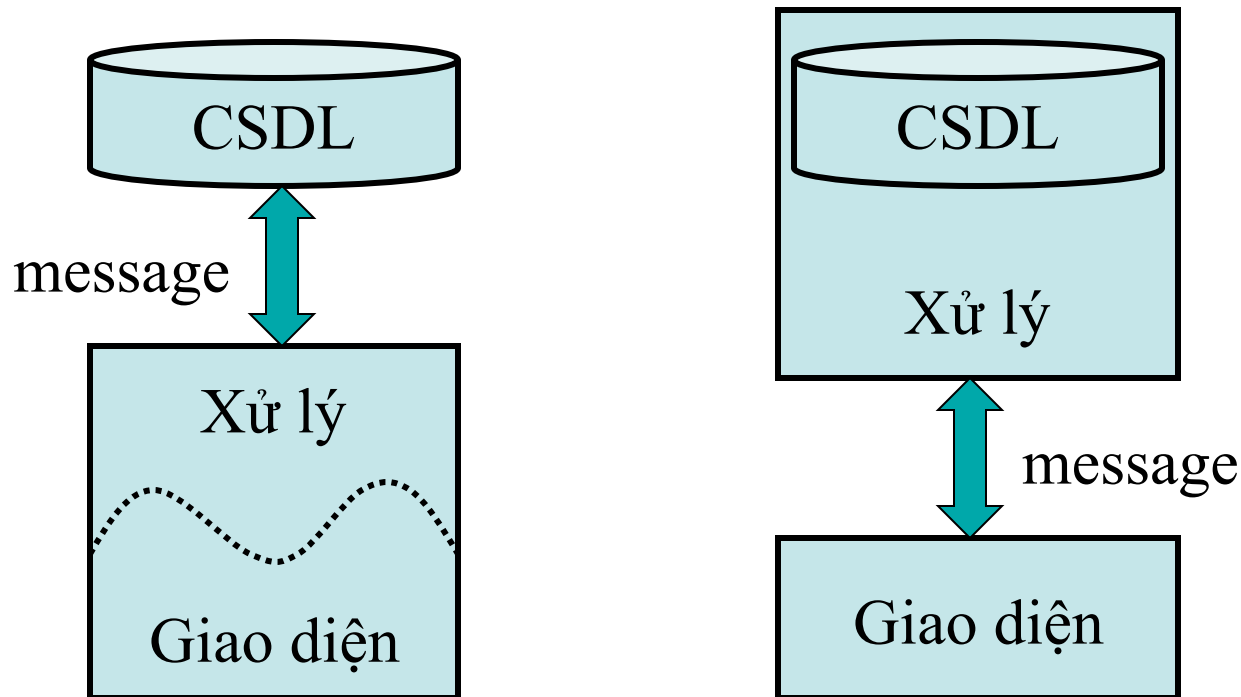
- **Client:** giao diện và chương trình xử lý được viết trực tiếp trong giao diện.
- **Server:** quản trị cơ sở dữ liệu.

Hạn chế: các thành phần của hệ thống phụ thuộc lẫn nhau dẫn đến khó cải tiến, bảo trì và tái sử dụng.

3. Thiết kế chức năng phần mềm (tt)

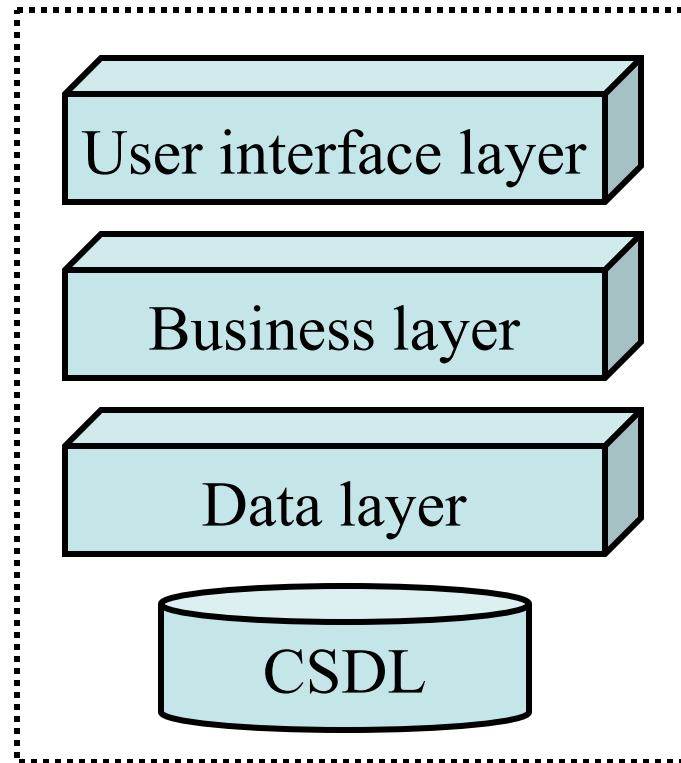
- **Kiến trúc phần mềm (tt)**

1) Kiến trúc client-server (tt)–Một số mô hình client-server:



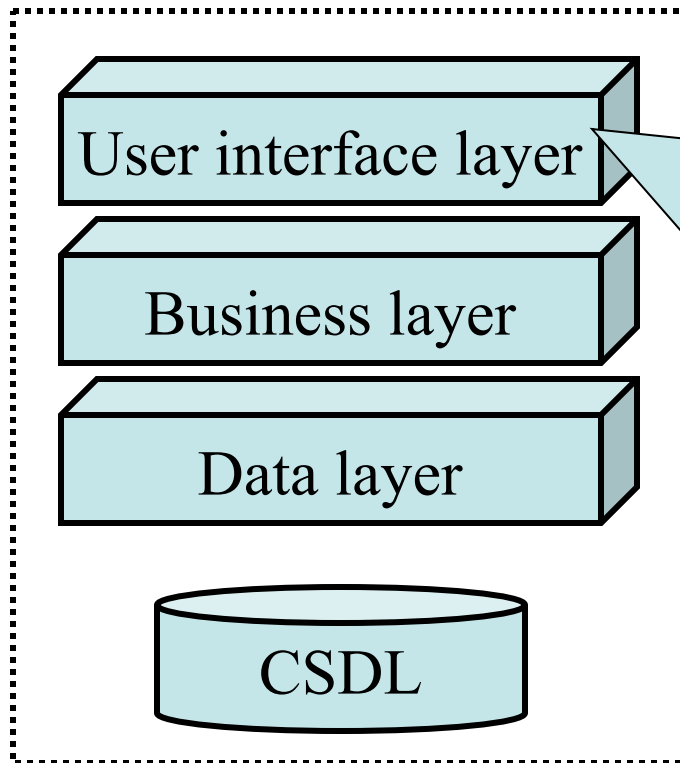
3. Thiết kế chức năng phần mềm (tt)

- Kiến trúc phần mềm (tt)
 - 2) Kiến trúc 3 tầng:



3. Thiết kế chức năng phần mềm (tt)

- Kiến trúc phần mềm (tt)
 - 2) Kiến trúc 3 tầng (tt)

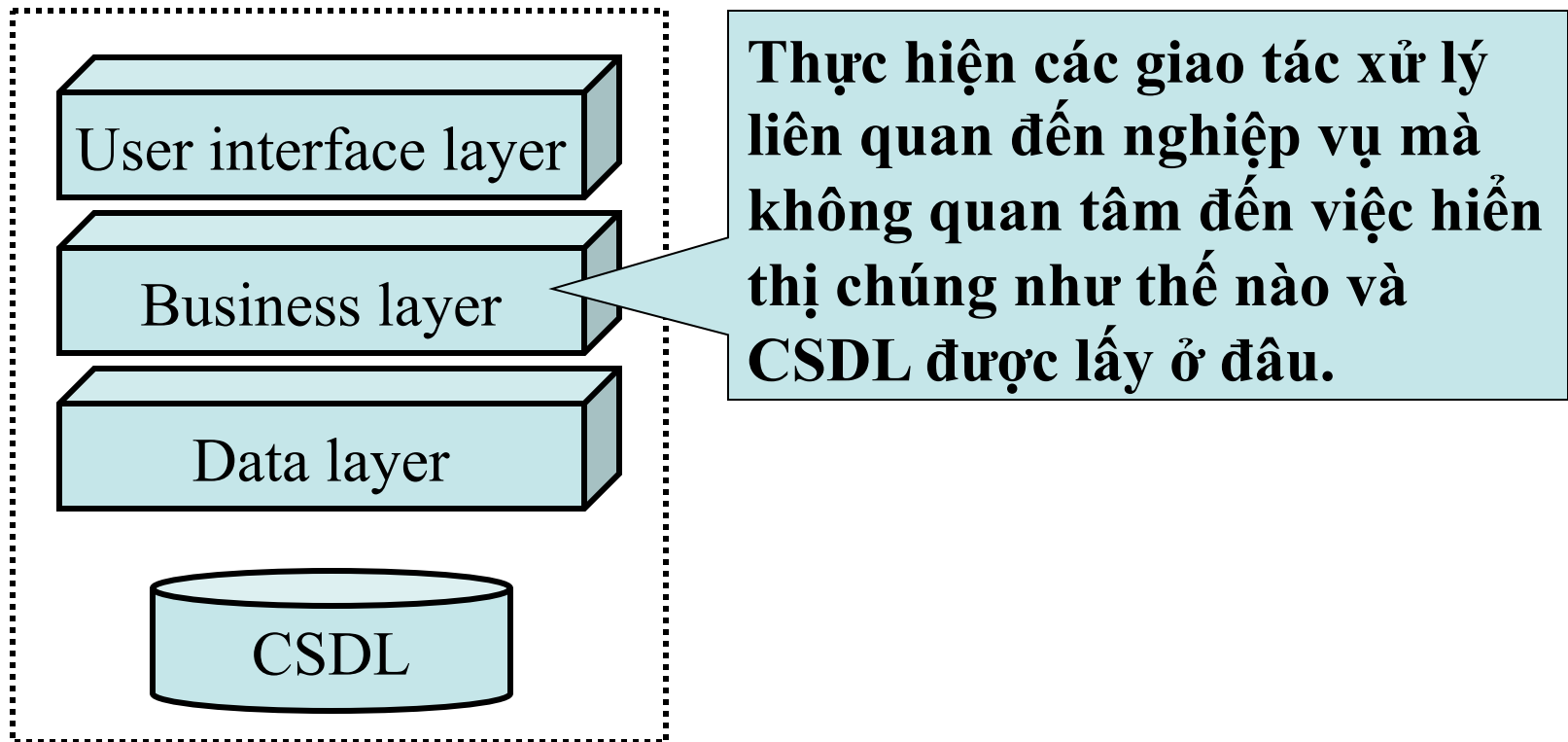


- **Trả lời tương tác người dùng:** Chuyển dịch những hành động của người dùng tới một tình huống xử lý phù hợp.

- **Hiển thị các đối tượng tác nghiệp:** trình bày một hình ảnh tốt nhất các đối tượng tác nghiệp tới người dùng trong một giao diện.

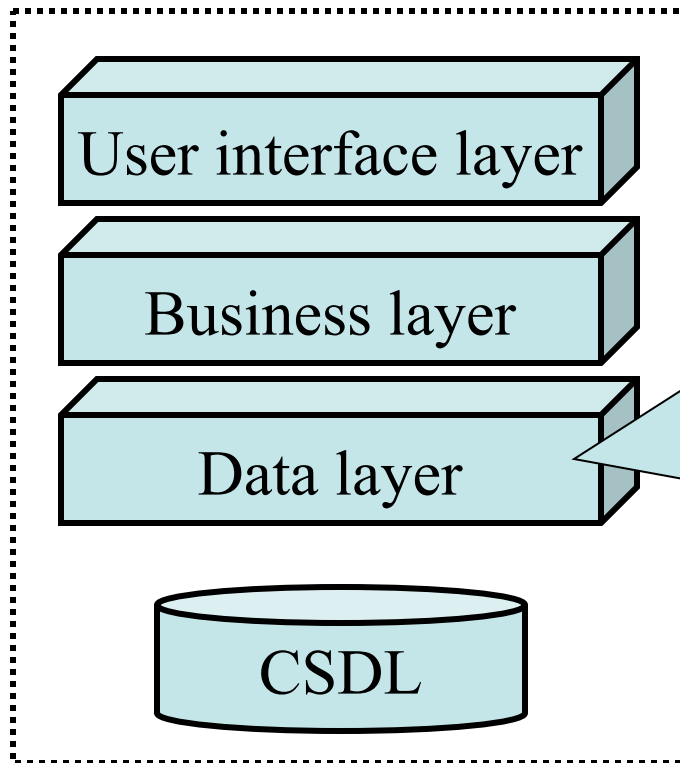
3. Thiết kế chức năng phần mềm (tt)

- Kiến trúc phần mềm (tt)
 - 2) Kiến trúc 3 tầng (tt)



3. Thiết kế chức năng phần mềm (tt)

- Kiến trúc phần mềm (tt)
 - 2) Kiến trúc 3 tầng (tt)



- **Chuyển dịch yêu cầu:** chuyển dịch tất cả các yêu cầu liên quan đến dữ liệu từ tầng tác nghiệp đến một phương thức truy cập dữ liệu thích hợp (dạng SQL, truy xuất file, ...).

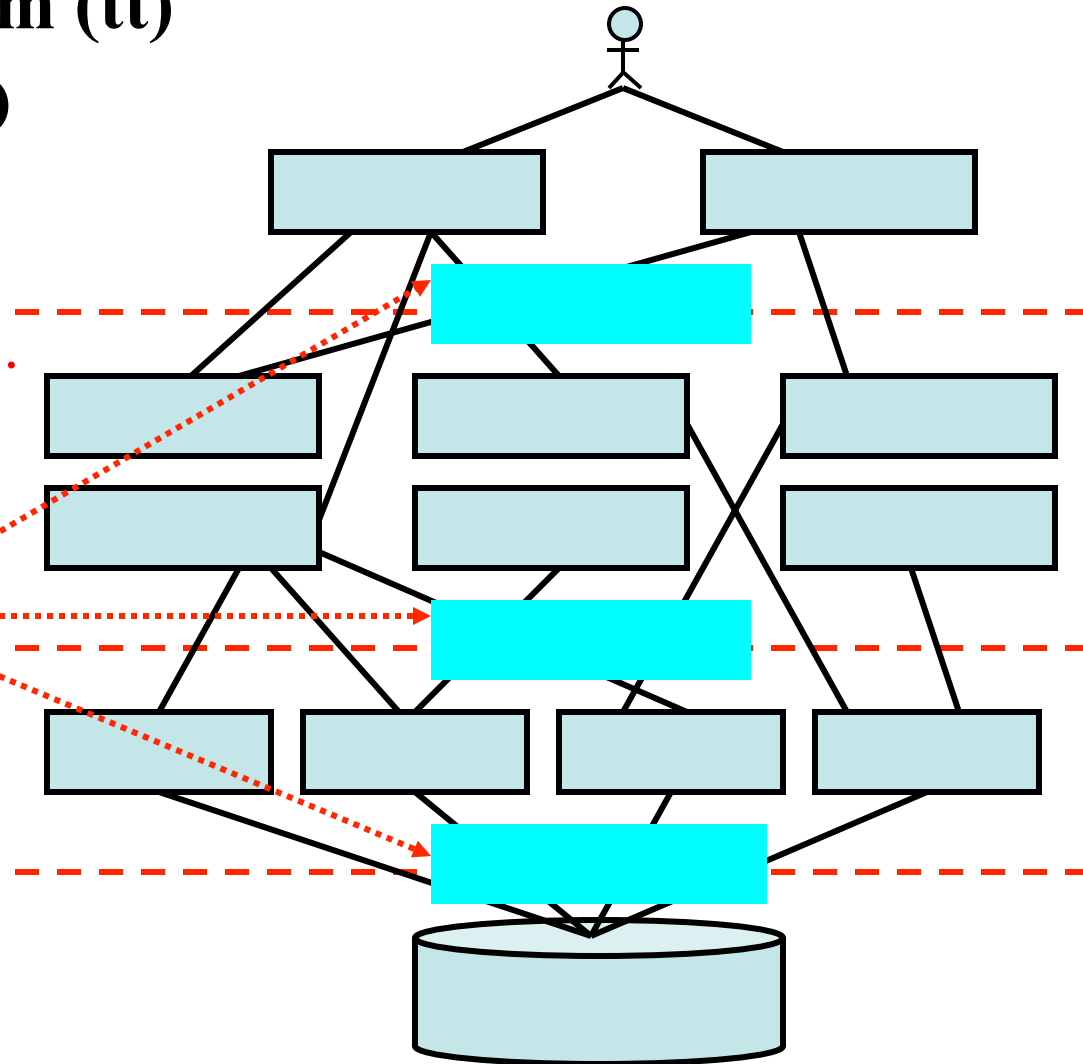
- **Chuyển dịch kết quả.**

3. Thiết kế chức năng phần mềm (tt)

- Kiến trúc phần mềm (tt)
 - 2) Kiến trúc 3 tầng (tt)

Sự phân tầng tạo ra sự độc lập -> dễ cải tiến, bảo trì, ...

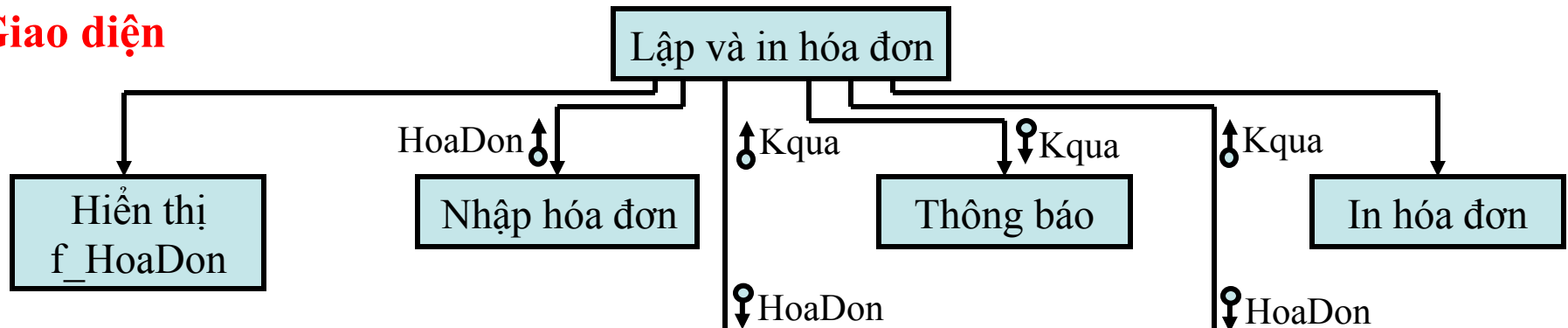
Tạo ra những module trung gian -> các tầng độc lập lẫn nhau tốt hơn.



3. Thiết kế chức năng phần mềm (tt)

- Thiết kế chức năng phần mềm theo kiến trúc 3 tầng:

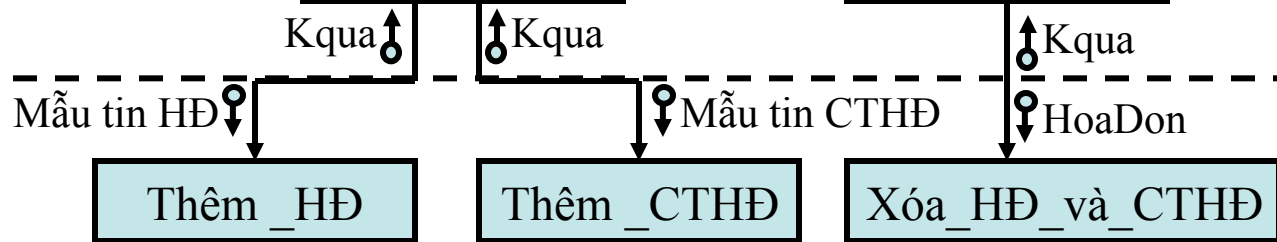
Giao diện



Nghệp vụ



Truy cập CSDL

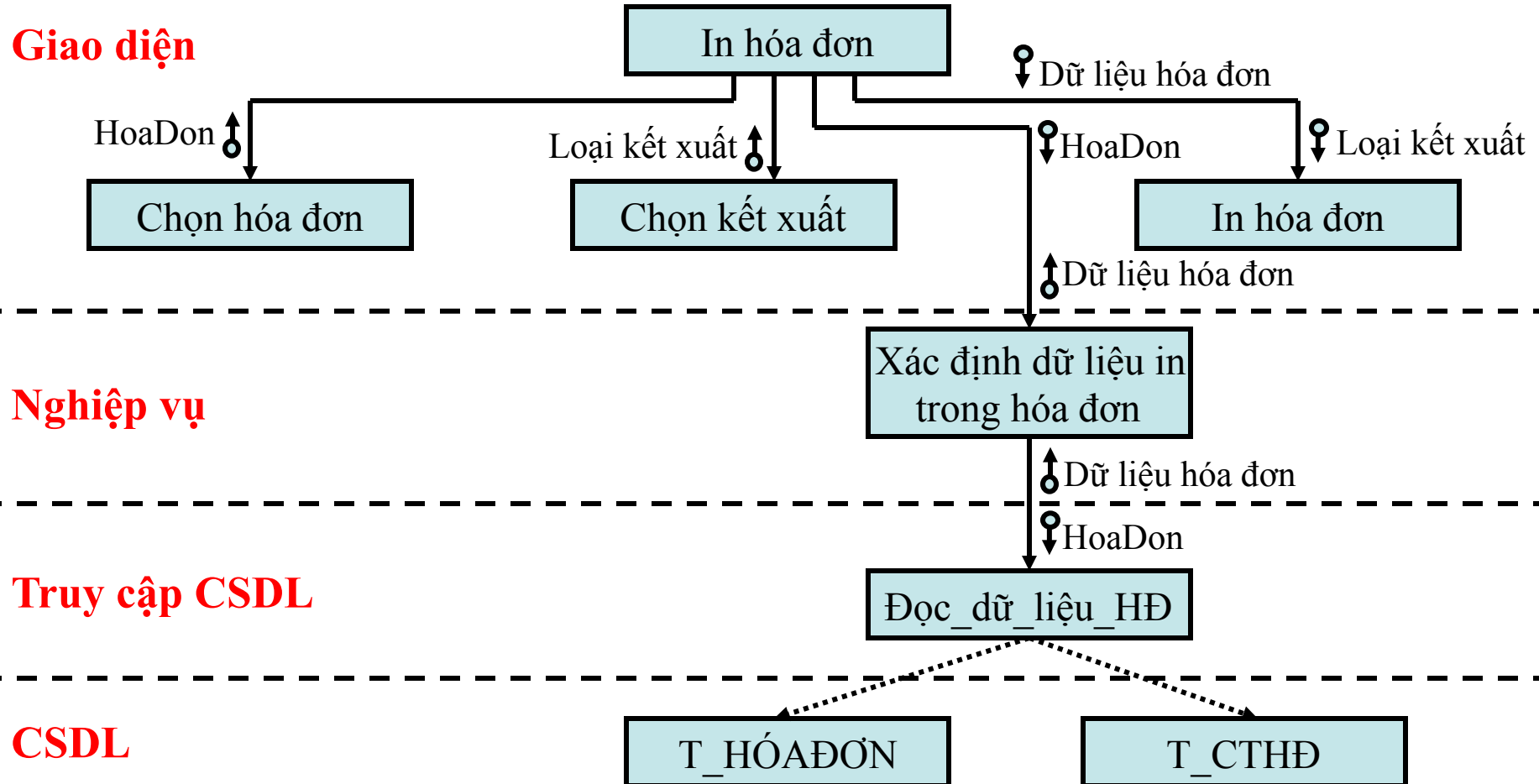


CSDL



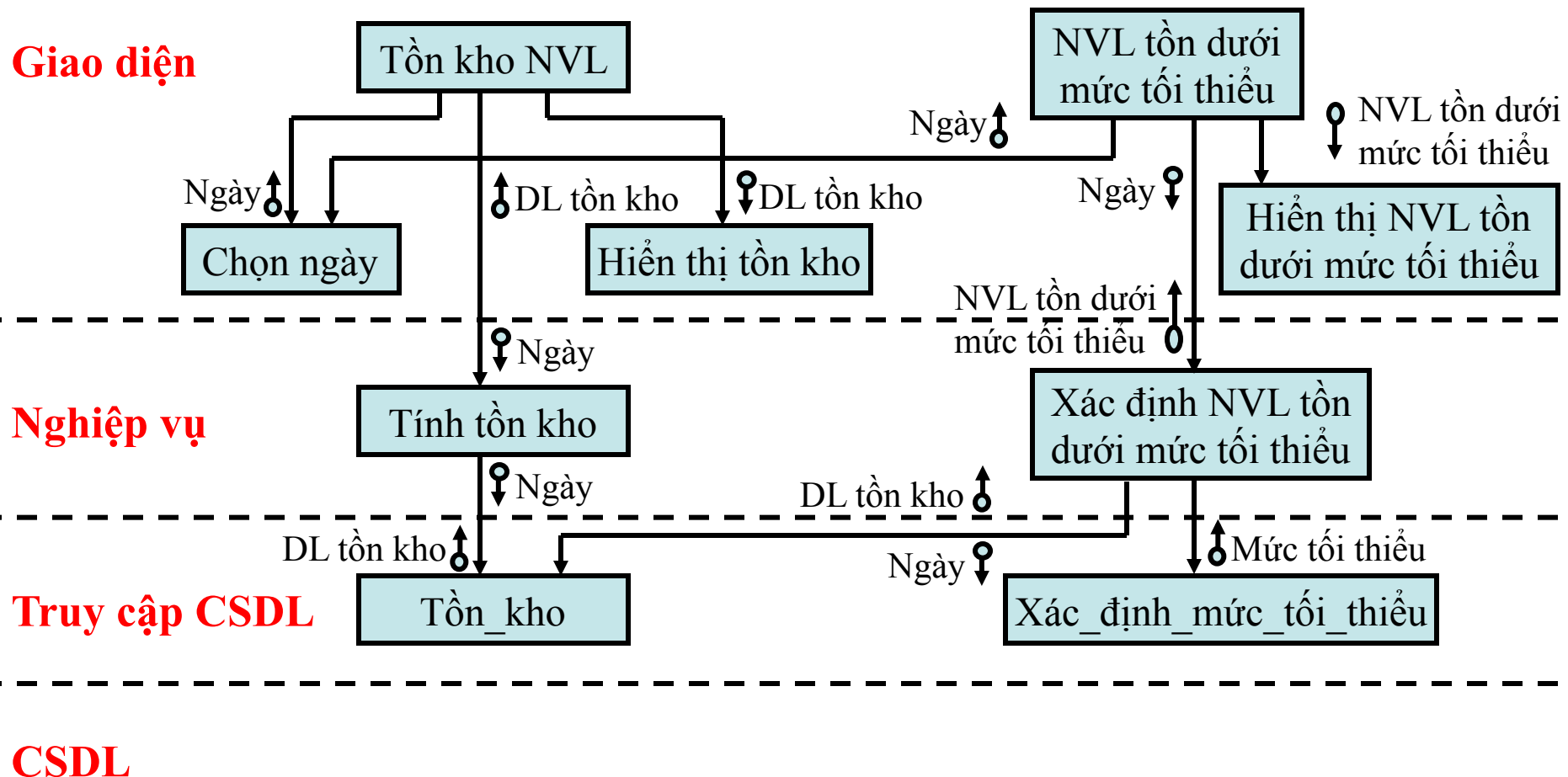
3. Thiết kế chức năng phần mềm (tt)

- Thiết kế chức năng phần mềm theo kiến trúc 3 tầng (tt)



3. Thiết kế chức năng phần mềm (tt)

- Thiết kế chức năng phần mềm theo kiến trúc 3 tầng (tt)



3. Thiết kế chức năng phần mềm (tt)

- **Các mục tiêu trong thiết kế hệ thống:**
 - 1) Hệ thống nên được tổ chức thành một cấu trúc phân cấp các module.
 - 2) Mỗi module nên điều khiển một số lượng hợp lý các module mức dưới.
 - 3) Các module nên độc lập với nhau, sự độc lập này càng cao càng tốt.
 - 4) Mỗi module nên có một kích thước vừa phải.
 - 5) Mỗi module chỉ nên đảm nhận một chức năng.

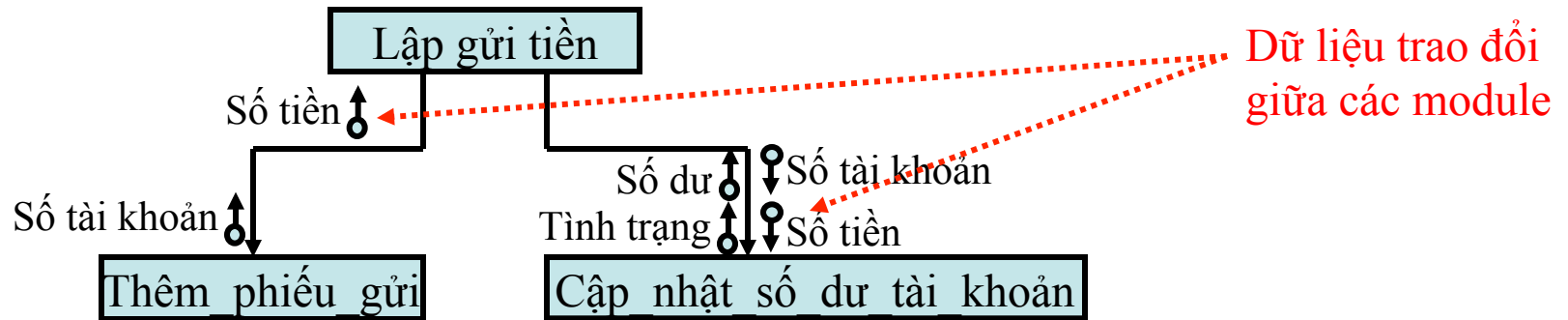
3. Thiết kế chức năng phần mềm (tt)

-> Một số hướng dẫn để đạt được mục tiêu:

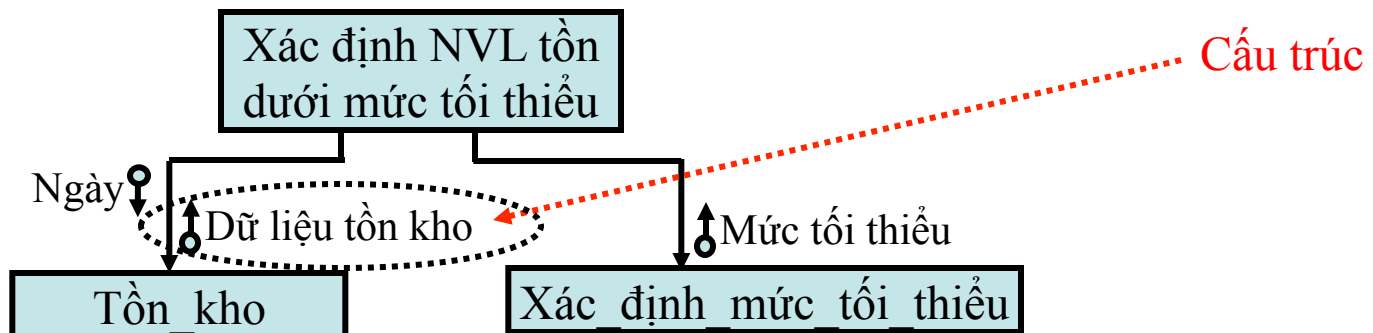
- 1) Dùng biểu đồ cấu trúc để biểu diễn cấu trúc chức năng phần mềm.
- 2) Mỗi module điều phối không nên điều khiển nhiều hơn 7 module thực thi.
- 3) **Coupling**: giảm tối đa sự phụ thuộc lẫn nhau giữa các module bằng cách giảm lượng truyền thông giữa các module. (*sự phụ thuộc càng ít càng tốt*)
- 4) Kích thước 1 module nên giới hạn từ 50->100 dòng lệnh.
- 5) **Cohesion**: các lệnh trong một module nên gắn liền tới cùng một chức năng. (*tính kết dính càng cao càng tốt*)
- 6) **Tái sử dụng**: các module ở mức thấp nên được sử dụng bởi nhiều module ở mức trên.

Phụ lục 1: Năm loại coupling

1) **Data coupling:** các module trao đổi với nhau thông qua cờ dữ liệu đơn hoặc cờ hiệu thông tin.



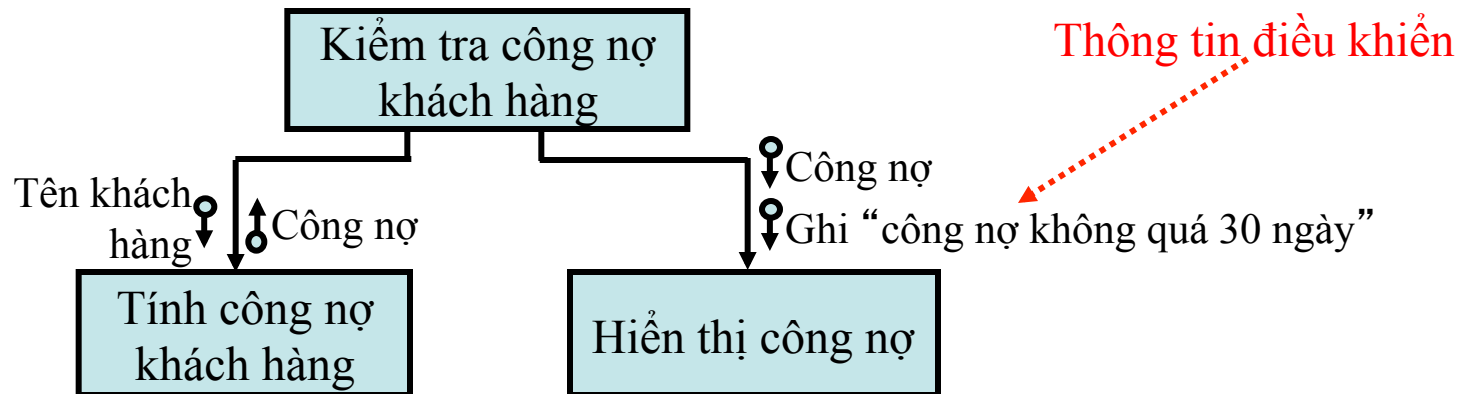
2) **Stamp coupling:** dữ liệu trao đổi giữa các module là một cấu trúc hoặc toàn bộ mẫu tin.



Phụ lục 1: Năm loại coupling (tt)

3) Control coupling: module này chuyển thông tin điều khiển đến module khác.

- **Thông tin điều khiển:** cờ hiệu thông báo cho module biết hành động nào nên thực hiện.



4) Common coupling: hai module cùng tham khảo đến một dữ liệu toàn cục.

5) Content coupling: module này có thể tham khảo đến nội dung của một module khác.

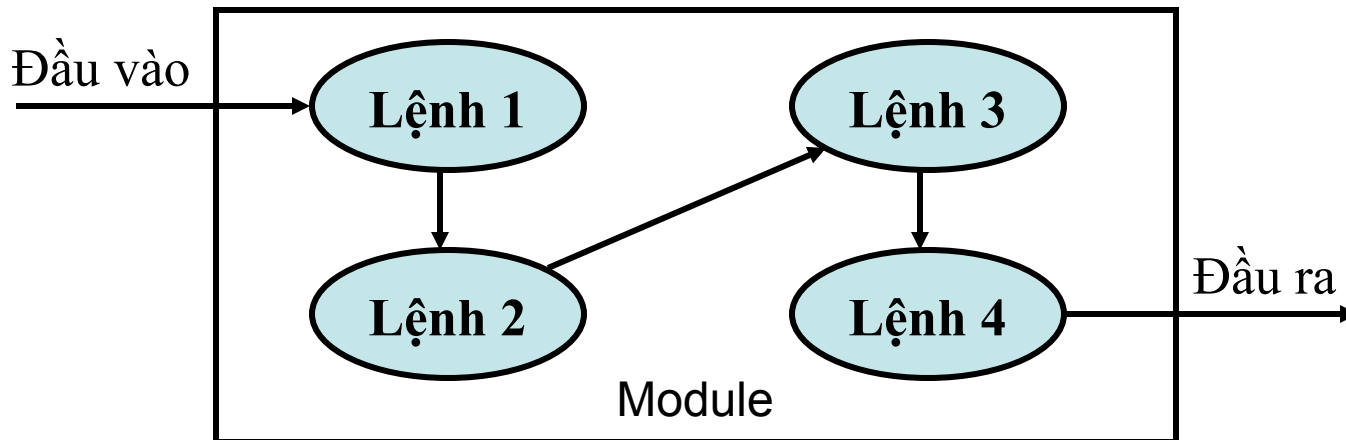
Phụ lục 1: Năm loại coupling (tt)

Xếp hạng coupling:

Tên coupling	Xếp hạng phụ thuộc
Data coupling	Rất thấp
Stamp coupling	Thấp
Control coupling	Trung bình
Common coupling	Cao
Content coupling	Rất cao

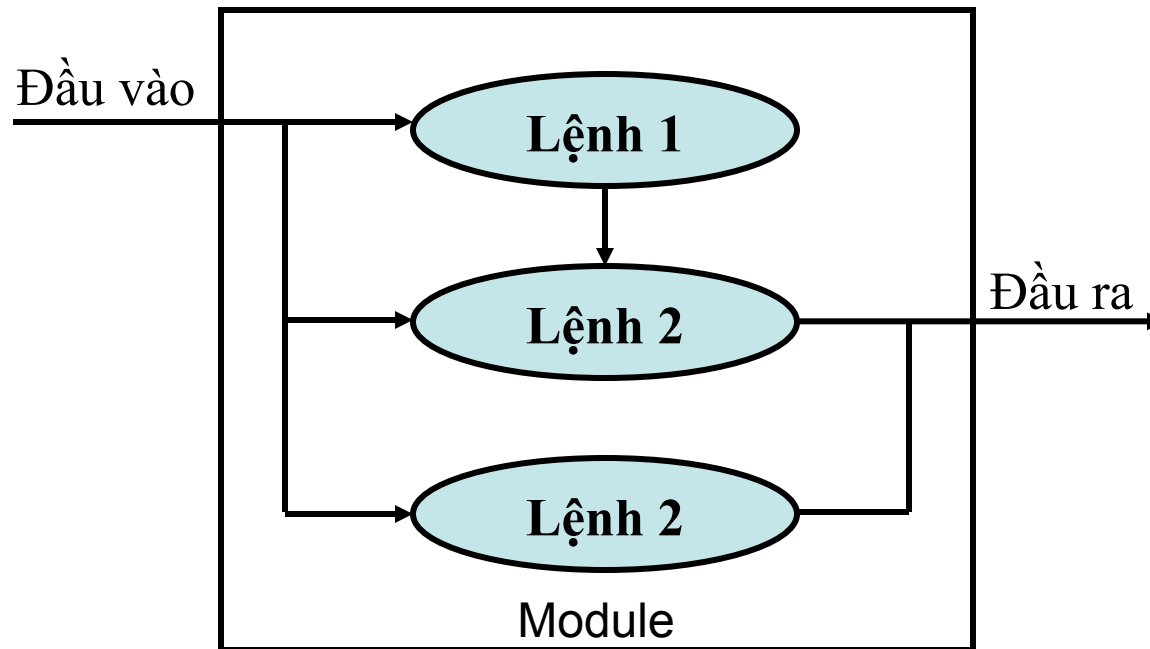
Phụ lục 2: Bảy loại Cohesion

- 1) **Cohesion về chức năng:** các lệnh trong một module gắn liền tới cùng một chức năng hoặc một nhiệm vụ.
- 2) **Cohesion tuần tự:** tuần tự đầu ra của lệnh này chính là đầu vào của lệnh khác trong một module.



Phụ lục 2: Bảy loại Cohesion (tt)

3) **Cohesion trao đổi:** các lệnh trong module liên quan với nhau thông qua việc sử dụng cùng dữ liệu vào hoặc liên quan đến cùng dữ liệu ra.



Phụ lục 2: Bảy loại Cohesion (tt)

- 4) **Cohesion thủ tục:** các lệnh trong module liên kết với nhau thông qua dòng điều khiển.
- 5) Các lệnh trong module liên quan với nhau thông qua việc thực hiện cùng thời điểm hoặc gần thời điểm với nhau.
- 6) **Cohesion luận lý:** bao gồm tập các lệnh được xác định từ bên ngoài module nên có thể trùng lặp một phần chức năng → khó bảo trì.
- 7) **Cohesion ngẫu nhiên:** tất cả các lệnh không liên quan đến những cái khác.