



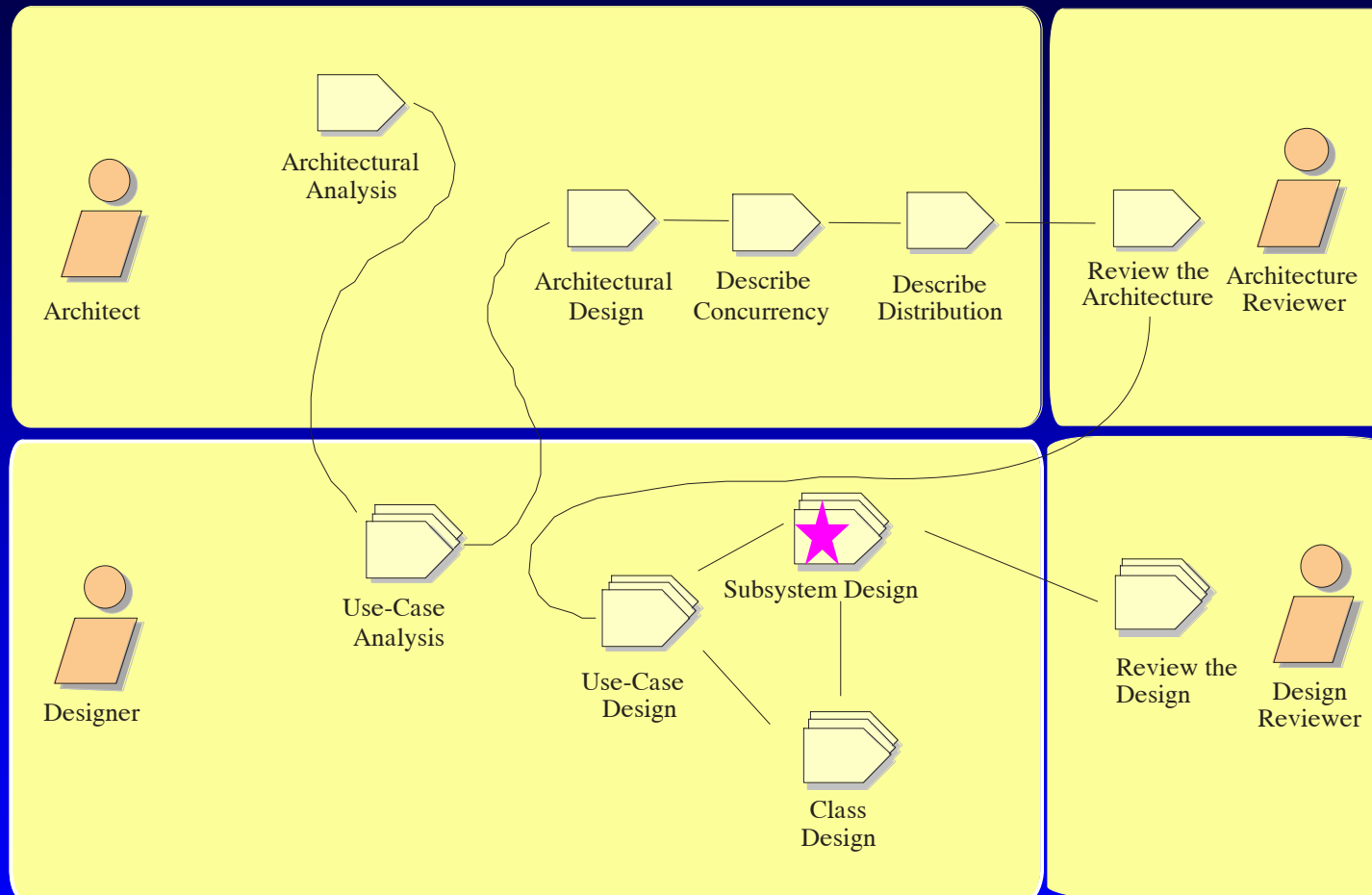
Phân tích và Thiết kế Hướng đối tượng dùng UML

Module 12: Thiết kế Subsystem

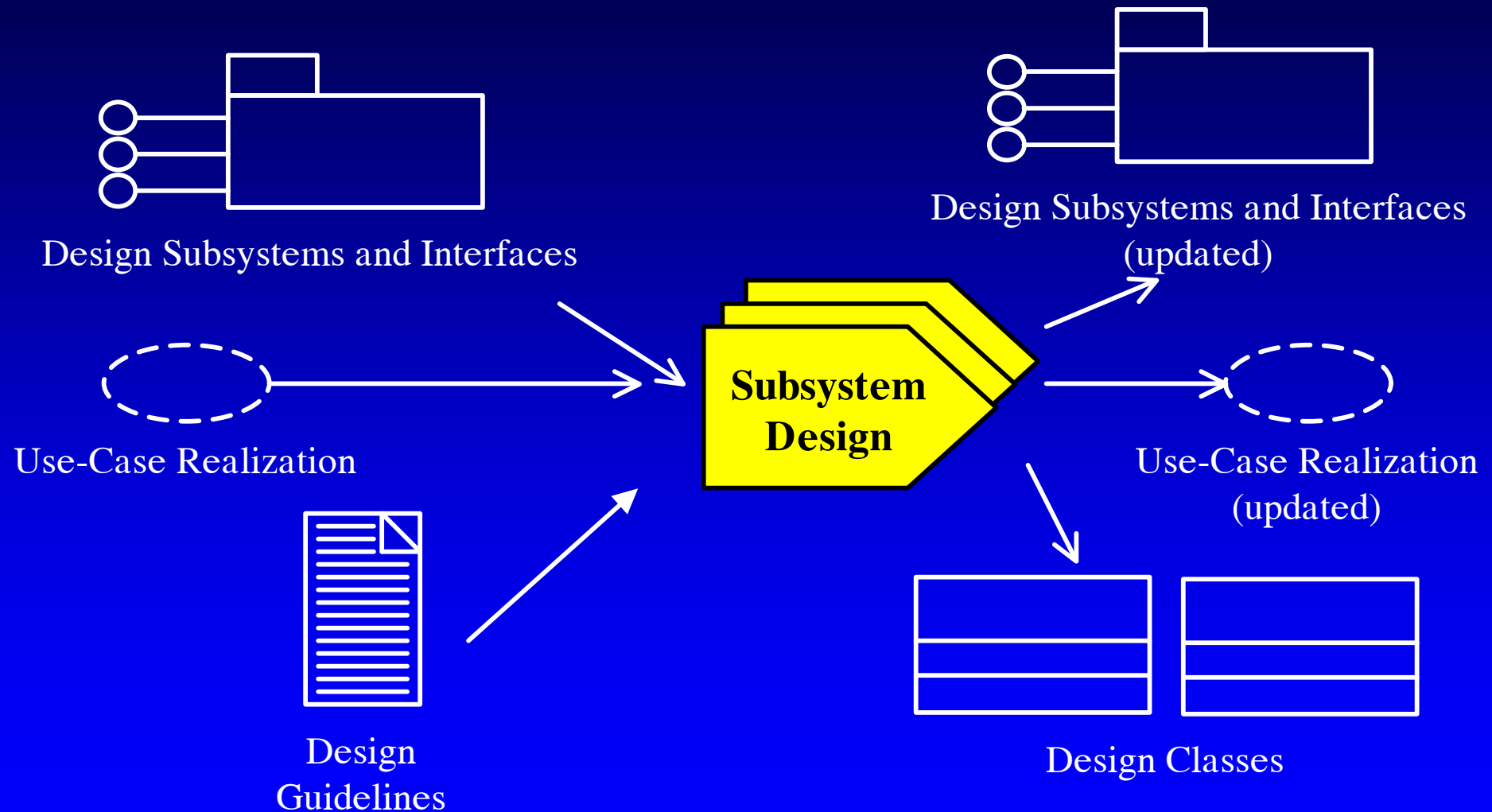
Mục tiêu:

- ◆ Tìm hiểu mục đích của bước thiết kế Subsystem và vị trí của công đoạn này trong qui trình
- ◆ Định nghĩa các hành vi được mô tả trong interfaces của subsystem dưới dạng collaboration của các class chứa bên trong
- ◆ Lập sơ liệu về cấu trúc bên trong của subsystem
- ◆ Xác định các phụ thuộc vào các phần tử bên ngoài subsystem

Vị trí của Thiết kế Subsystem

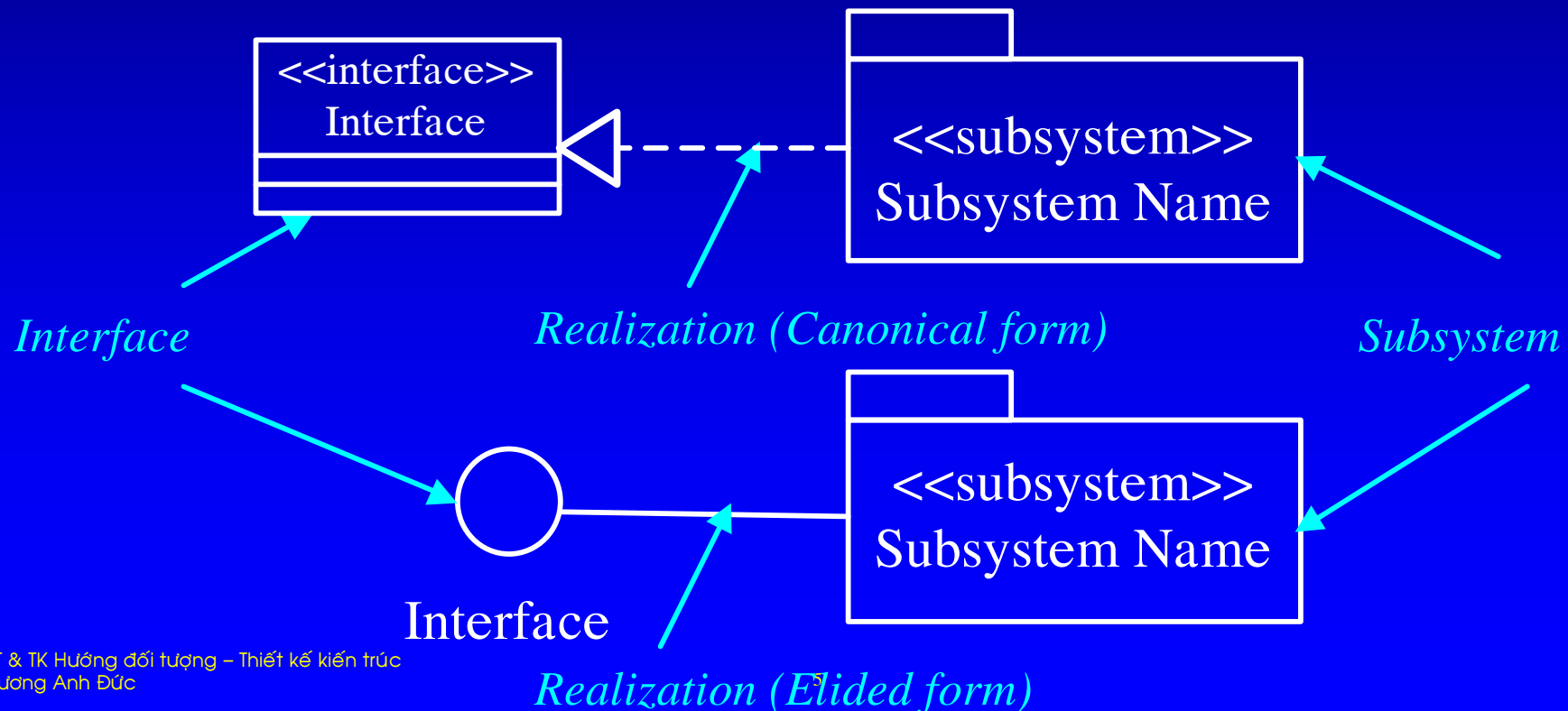


Tổng quan về thiết kế Subsystem



Nhắc lại: Subsystems và Interfaces

- ◆ Subsystem là sự “kết hợp” giữa package và class
- ◆ Hiện thực hóa 1 hoặc nhiều interfaces định nghĩa hành vi của nó



Subsystem Guidelines

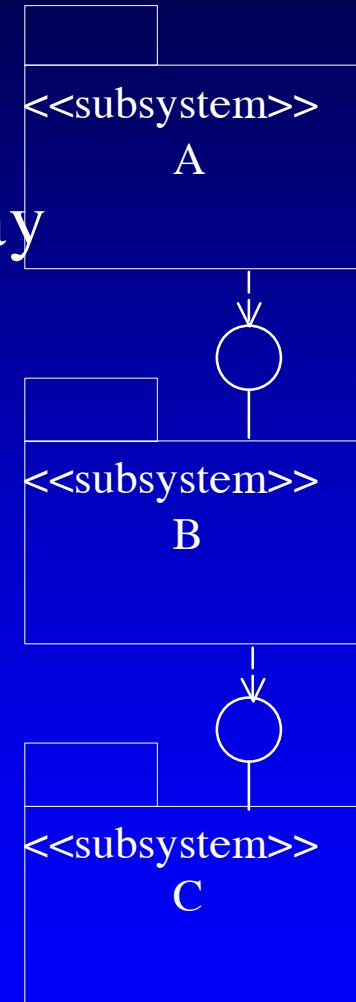
◆ Mục tiêu

- Giảm thiểu kết nối
- Có tính cơ động, tương thích plug-and-play
- Cách ly các thay đổi
- Tiến hóa độc lập

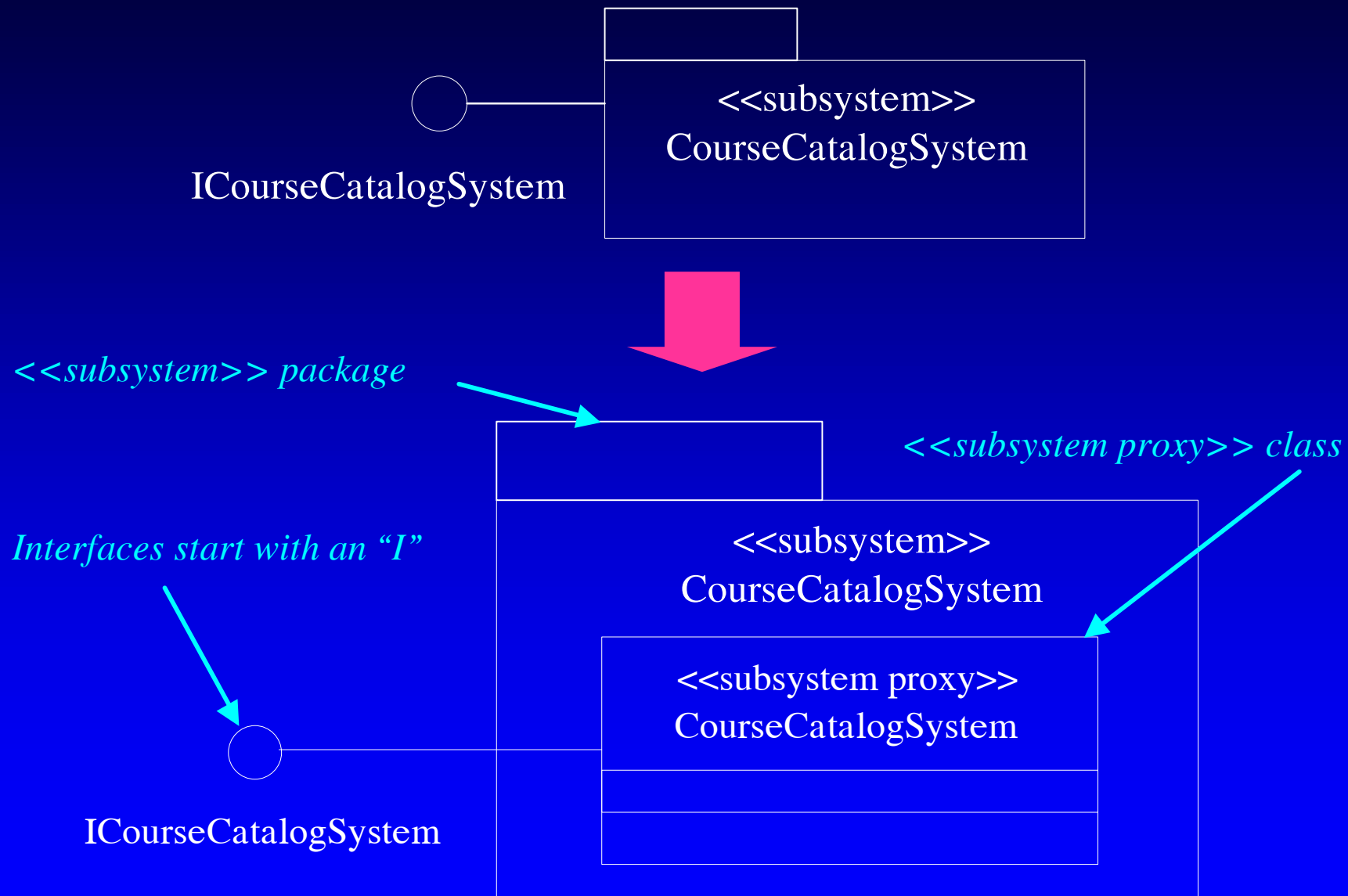
◆ Những đề nghị

- Đừng khảo sát chi tiết, chỉ interfaces
- Chỉ phụ thuộc vào interfaces khác

Chìa khóa là abstraction và encapsulation



Nhắc lại: Các quy ước mô hình hóa



Các bước thiết kế Subsystem

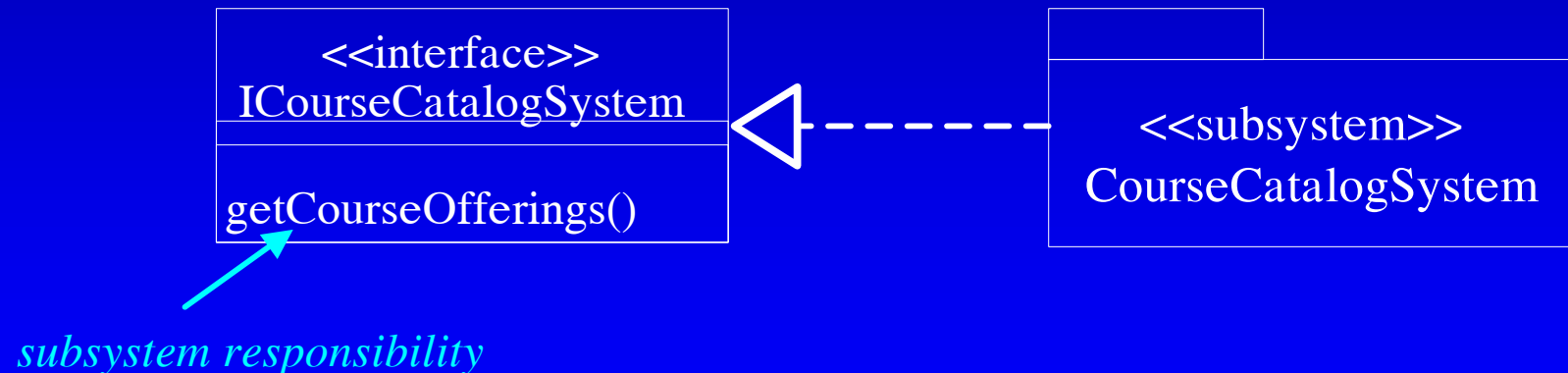
- ◆ Phân bố các hành vi của Subsystem cho các phần tử của nó
- ◆ Lập sơ liệu về các phần tử của Subsystem
- ◆ Mô tả các phụ thuộc của Subsystem
- ◆ Checkpoints

Các bước thiết kế Subsystem

- ★ ♦ Phân bố các hành vi của Subsystem cho các phần tử của nó
- ♦ Lập sơ liệu về các phần tử của Subsystem
- ♦ Mô tả các phụ thuộc của Subsystem
- ♦ Checkpoints

Các nhiệm vụ của Subsystem

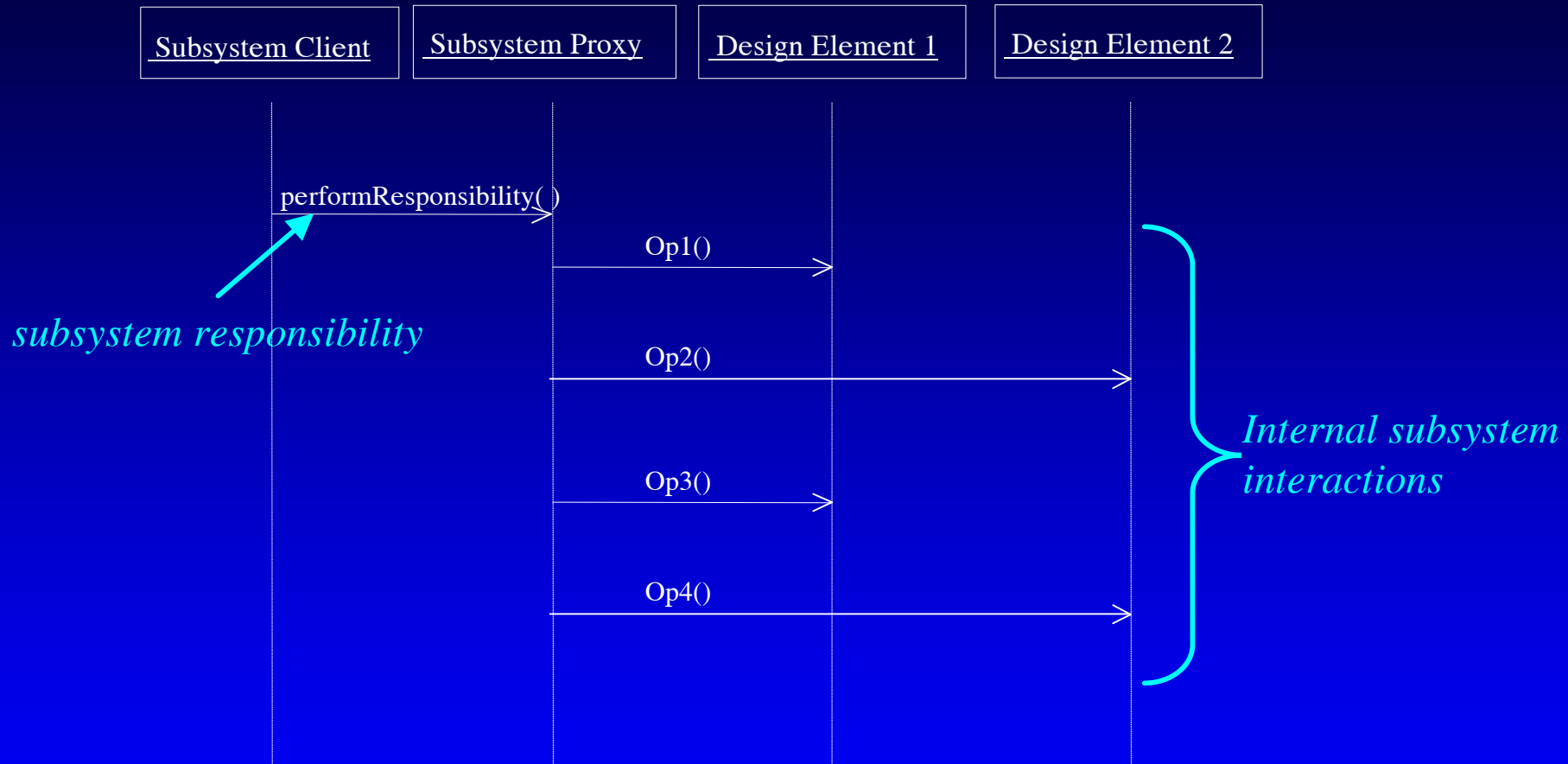
- ◆ Các interface operation xác định nhiệm vụ của Subsystem
- ◆ Interface operations được hiện thực hóa bởi
 - Internal class operations
 - Internal subsystem operations



Phân bổ nhiệm vụ của Subsystem

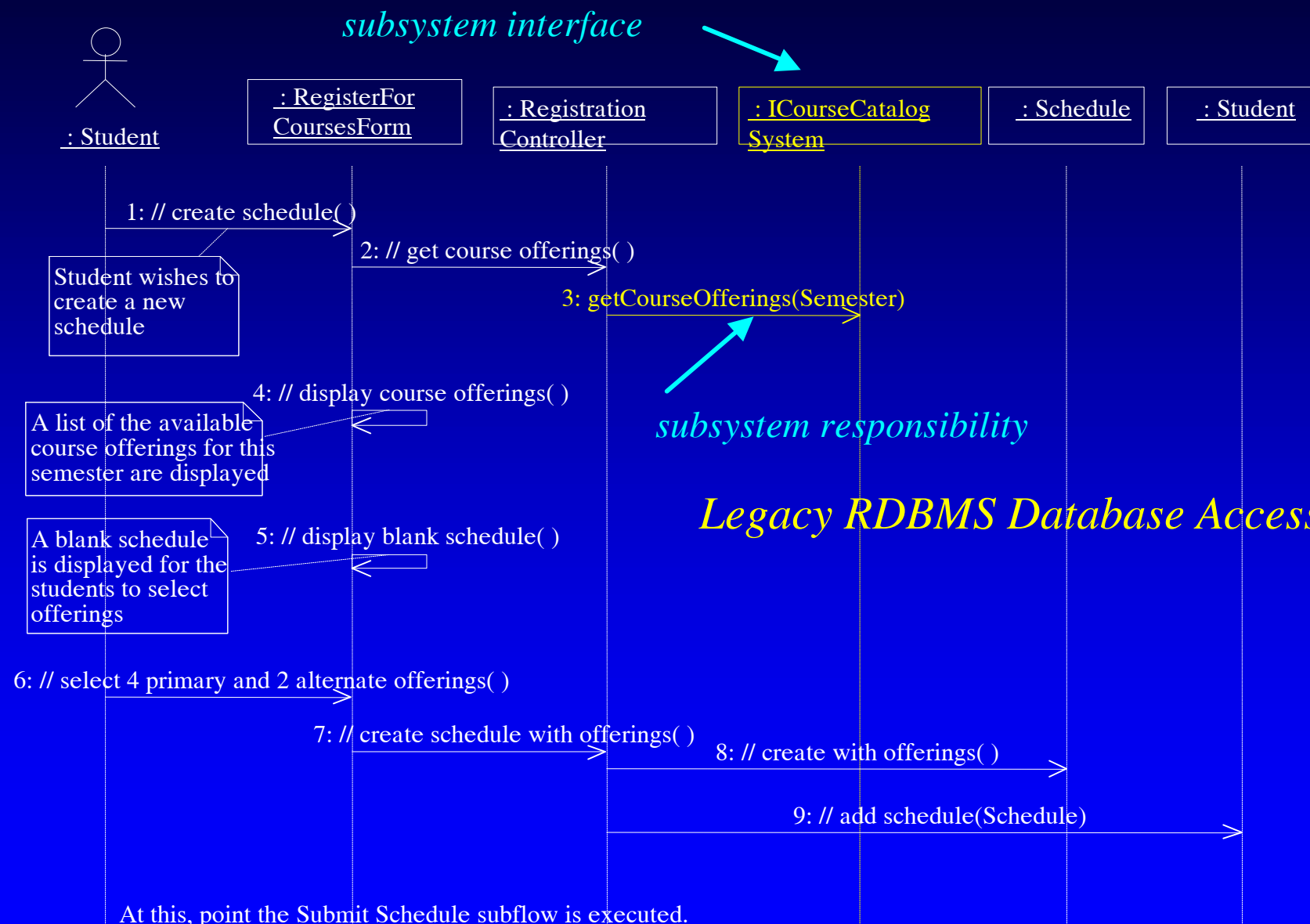
- ◆ Xác định các phân tử thiết kế mới, hoặc dùng lại cái có sẵn, (chẳng hạn, các class hay subsystem)
- ◆ Gán trách nhiệm của subsystem cho các phân tử của nó
- ◆ Tích hợp các cơ chế khả dụng (persistence, distribution, ...)
- ◆ Lập sơ liệu về collaboration giữa các phân tử thiết kế trong “interface realizations”
 - Một hay nhiều interaction diagrams/interface operation
 - Class diagram(s) chứa các quan hệ cần thiết
- ◆ Tham khảo lại phần thiết kế kiến trúc
 - Chỉnh subsystem boundaries và các phụ thuộc, nếu cần

Qui ước: Subsystem Interaction Diagrams



Không hiển thị Subsystem interface

Ví dụ: CourseCatalogSystem Subsystem



Tích hợp cơ chế kiến trúc: Persistency

- ◆ Bảng ánh xạ Analysis-Class với các cơ chế kiến trúc có từ phân tích Use-Case

Analysis Class	Analysis Mechanism(s)	
Student	Persistency, Security	<i>OODBMS Persistency</i>
Schedule	Persistency, Security	
CourseOffering	<i>Persistency, Legacy Interface</i>	<i>RDBMS Persistency</i>
Course	<i>Persistency, Legacy Interface</i>	
RegistrationController	Distribution	

*OODBMS Persistency đã khảo sát
trong Use-Case Design*

Nhắc lại: Các bước tích hợp JDBC

- ♦ Cung cấp khả năng truy suất đến thư viện các class cần để cài đặt JDBC

Ö

- *Cung cấp java.sql package*

- ♦ Tạo các DBClass cần thiết

- *Một DBClass/một persistent class*
- *Course Offering persistent class => DBCourseOffering*

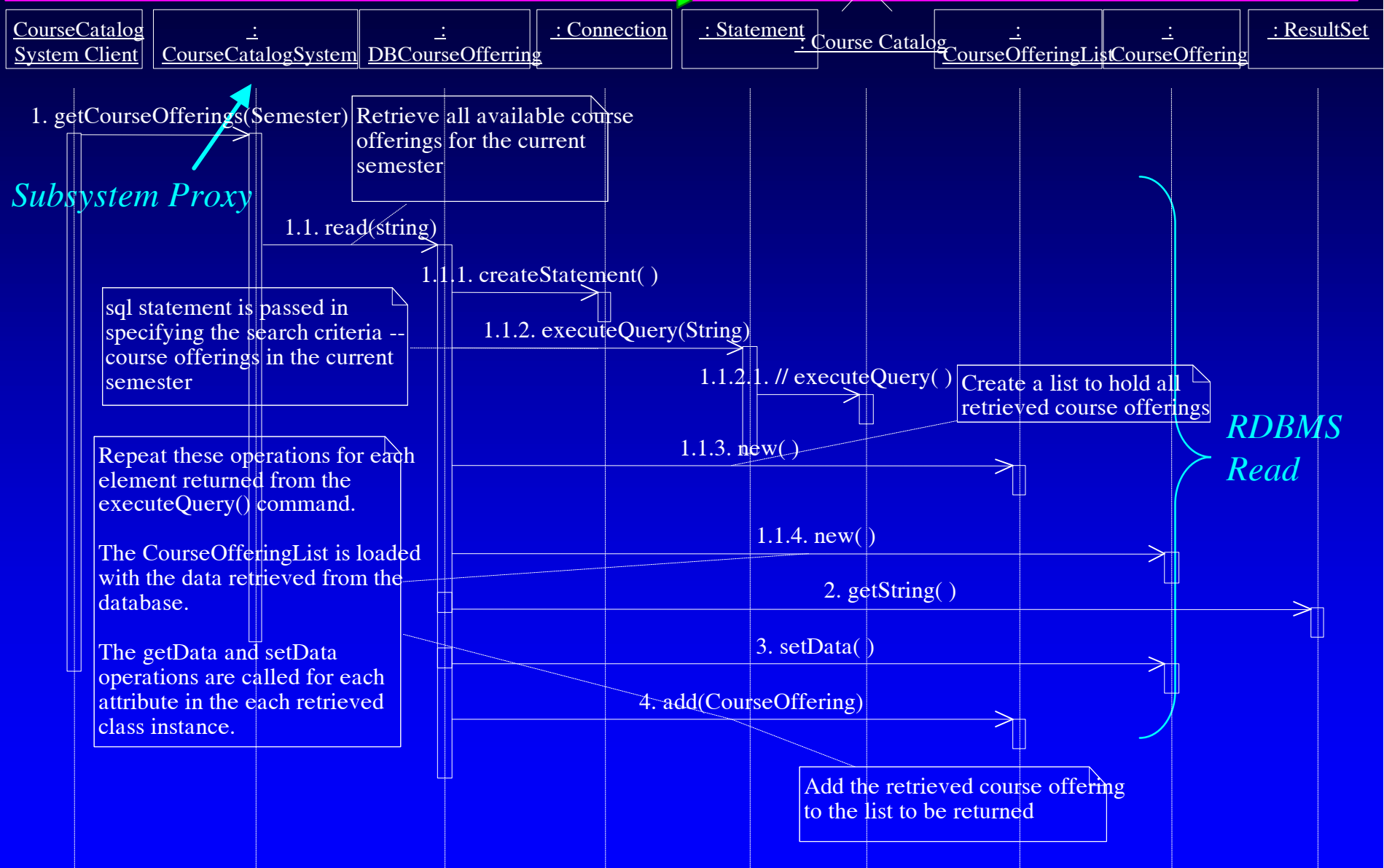
Ö - *Done*

(còn tiếp)

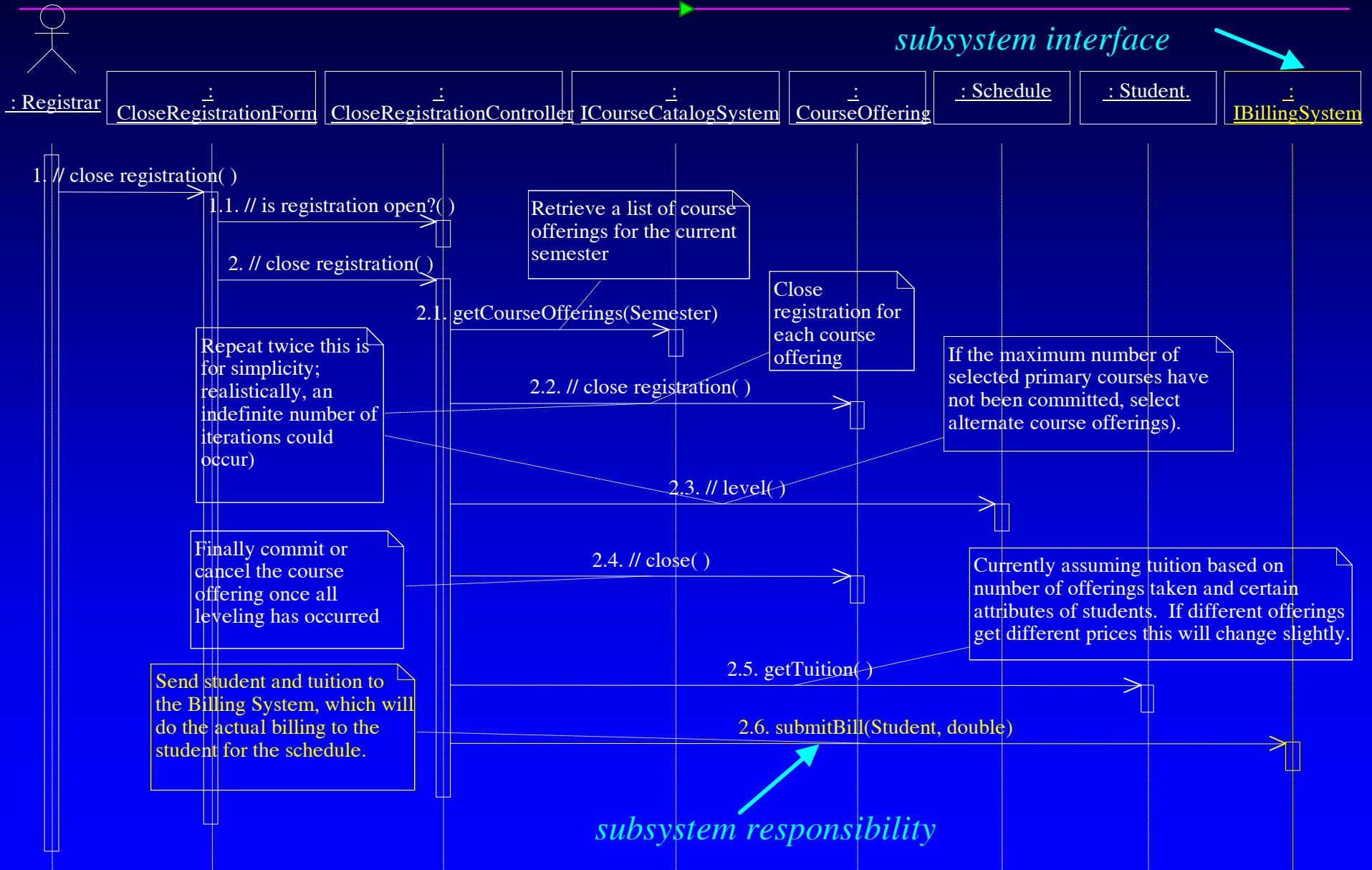
Nhắc lại: Các bước tích hợp JDBC (tt.)

- ◆ Tích hợp các DBClass vào thiết kế
 - Gán cho các package/layer
 - *DBCourseOffering* đặt vào *CourseCatalogSystem subsystem*
 - Thêm các quan hệ với persistency clients
 - *Persistency clients* là *CourseCatalogSystem subsystem clients*
- ◆ Tạo/Hiệu chỉnh các interaction diagram mô tả:
 - Việc khởi tạo Database
 - Truy xuất đến Persistent class: Create, Read, Update, Delete

Ví dụ: Local CourseCatalogSystem Subsystem Interaction

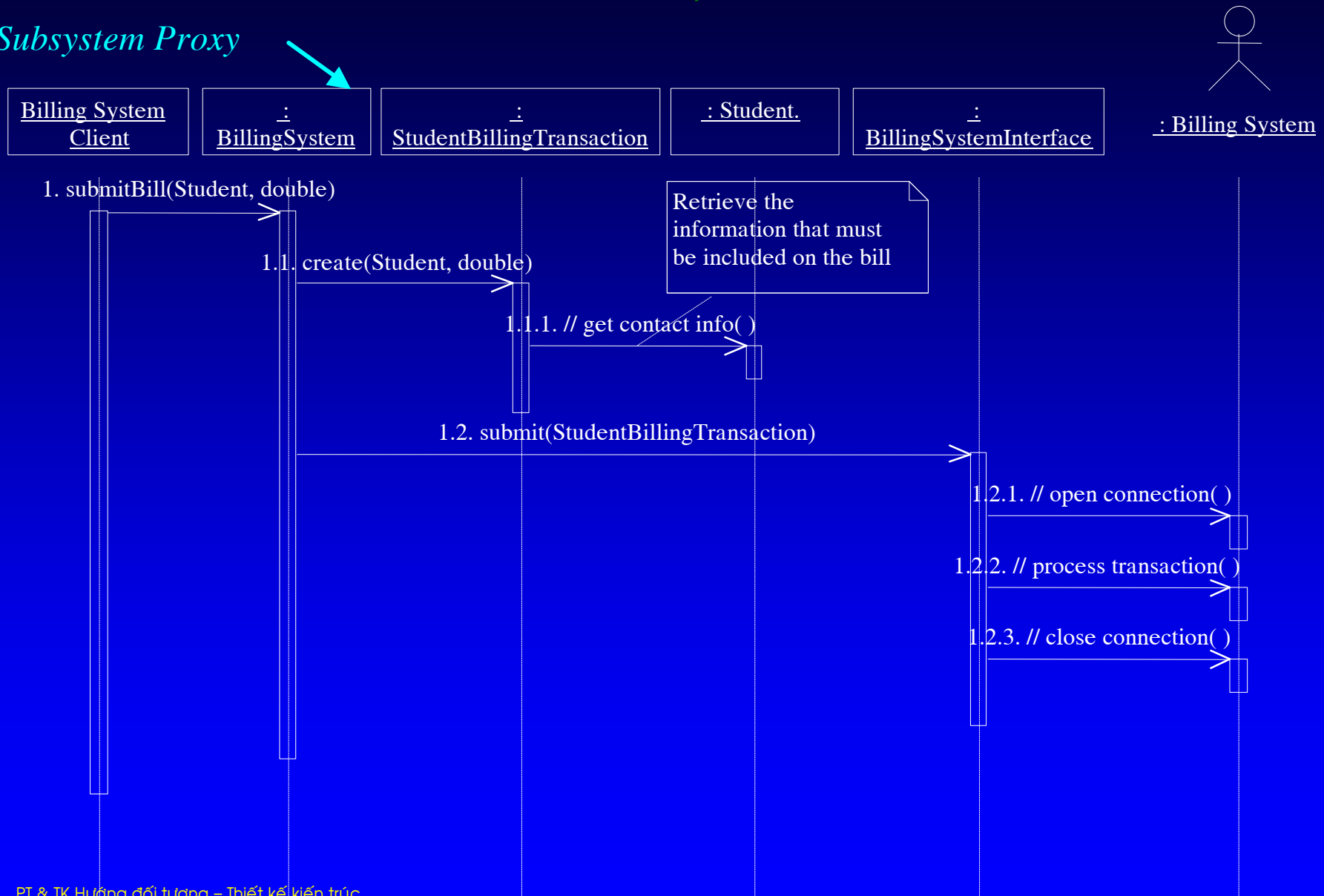


Ví dụ: Billing System Subsystem



Ví dụ: Local BillingSystem Subsystem Interaction

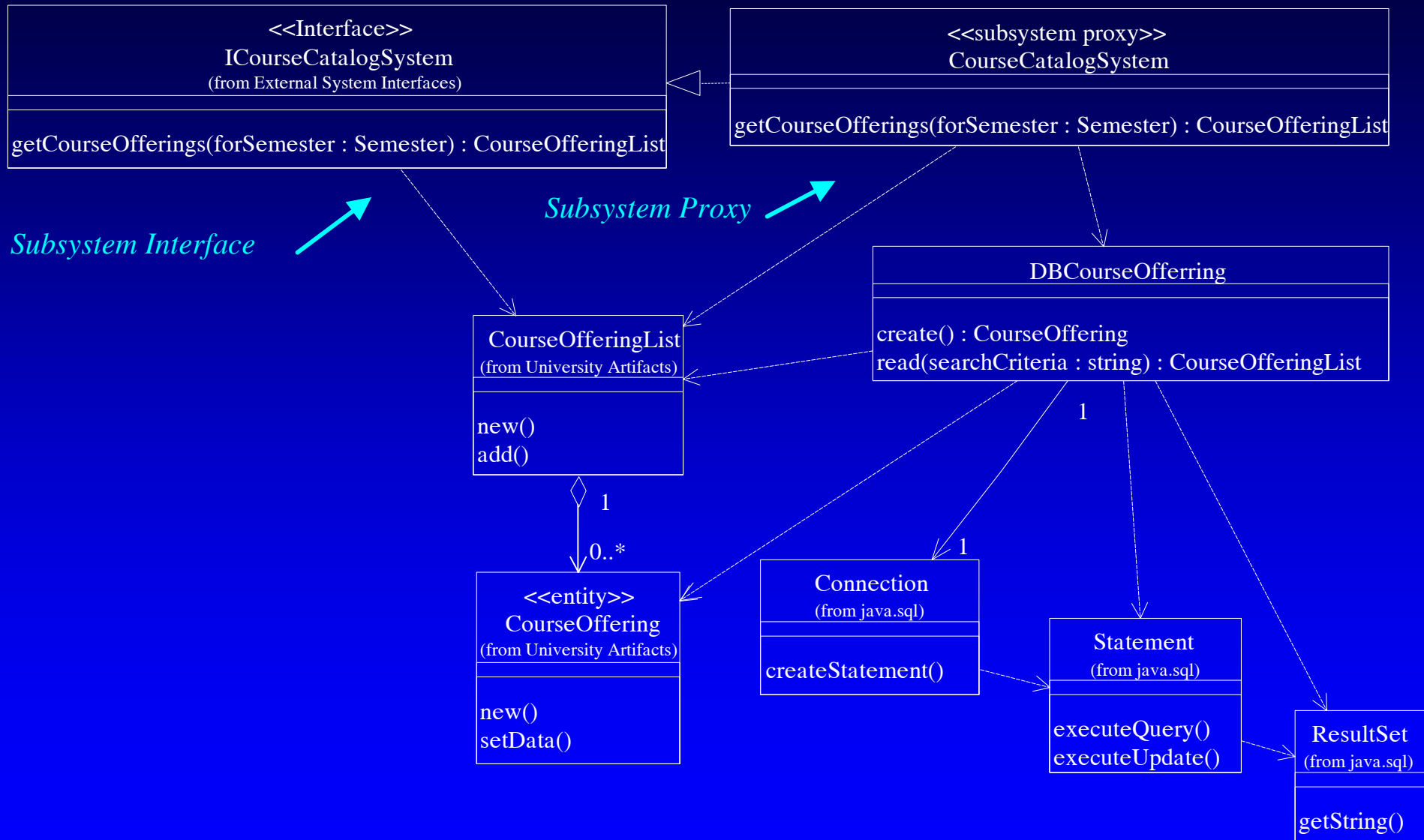
Subsystem Proxy



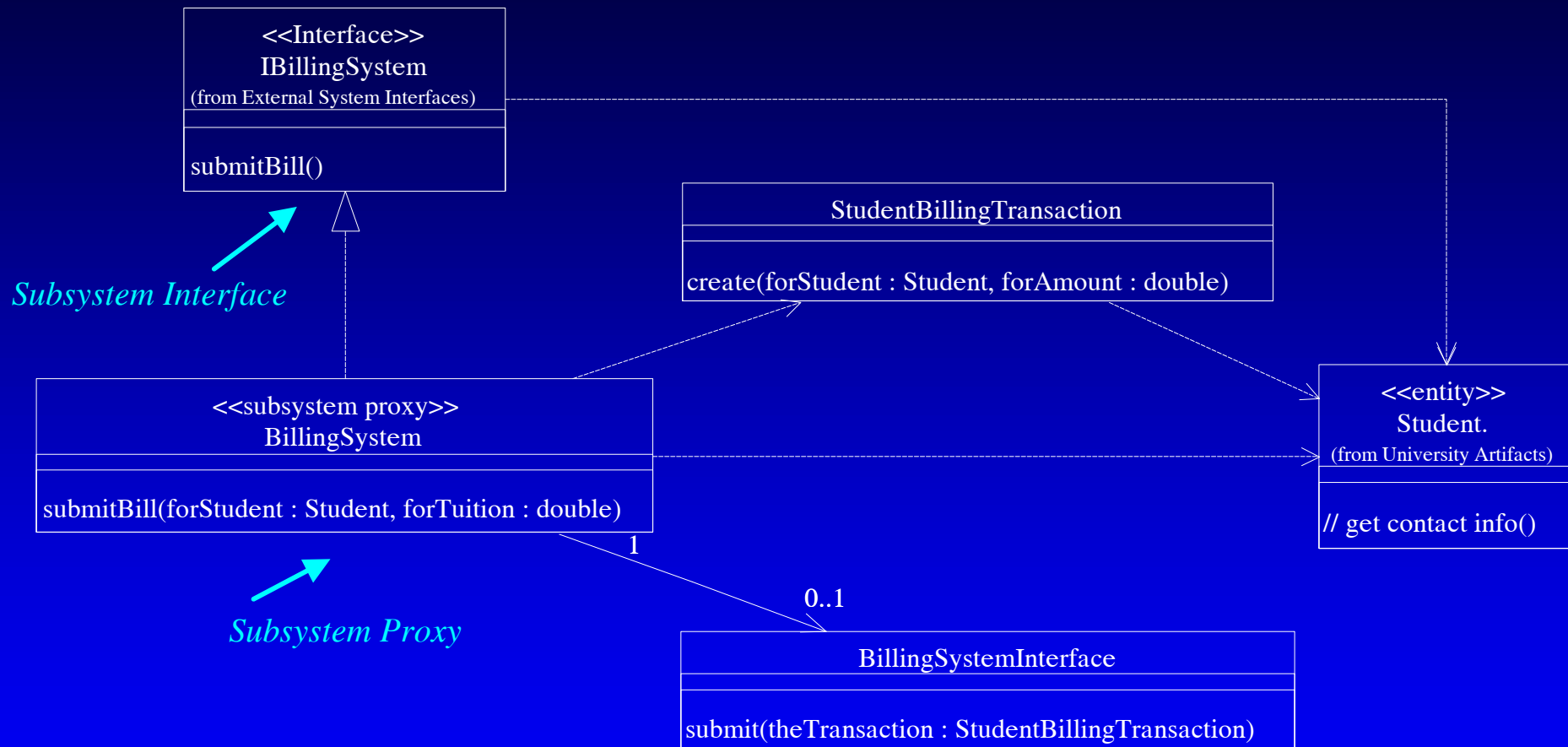
Các bước thiết kế Subsystem

- ◆ Phân bố các hành vi của Subsystem cho các phần tử của nó
- ★ ◆ Lập sơ liệu về các phần tử của Subsystem
- ◆ Mô tả các phụ thuộc của Subsystem
- ◆ Checkpoints

Ví dụ: CourseCatalogSystem Subsystem Elements



Ví dụ: Billing System Subsystem Elements

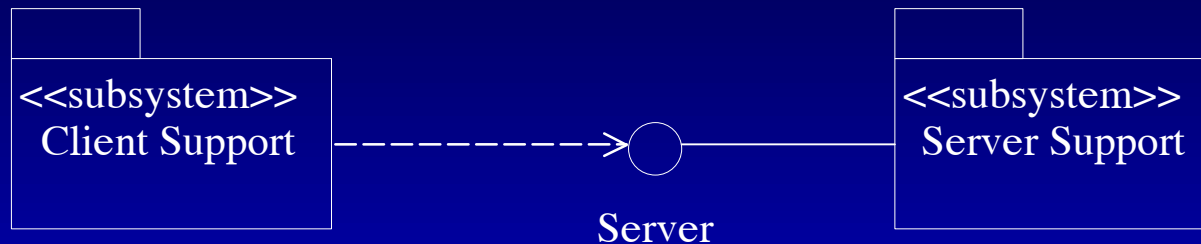


Các bước thiết kế Subsystem

- ◆ Phân bố các hành vi của Subsystem cho các phần tử của nó
- ◆ Lập sơ liệu về các phần tử của Subsystem
- ★ ◆ Mô tả các phụ thuộc của Subsystem
- ◆ Checkpoints

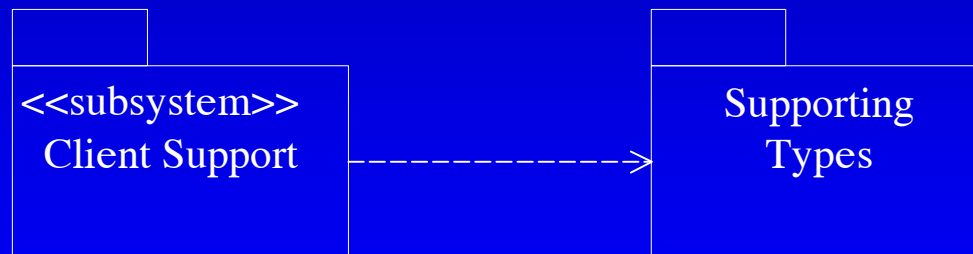
Describing Subsystem Dependencies

- ◆ Subsystem phụ thuộc vào một subsystem



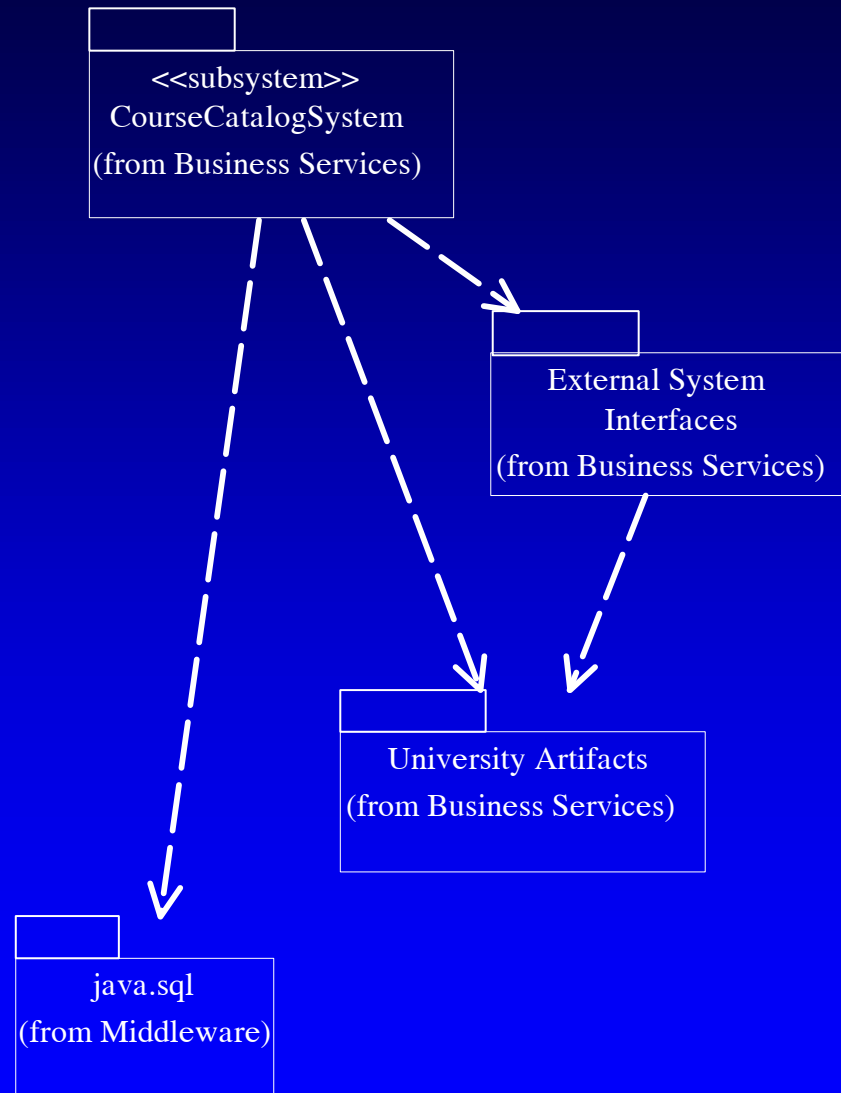
Mềm dẻo

- ◆ Subsystem phụ thuộc vào package

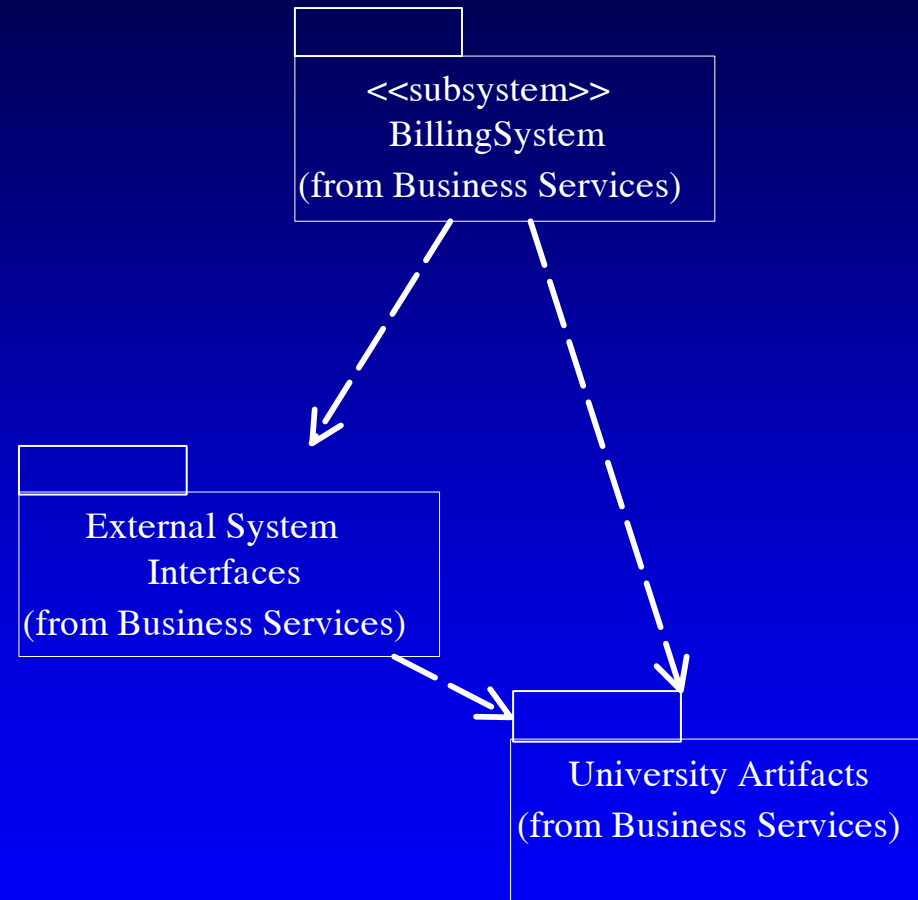


Dùng nhưng cẩn thận

Ví dụ: CourseCatalogSystem Subsystem Dependencies



Ví dụ: BillingSystem Subsystem Dependencies



Các bước thiết kế Subsystem

- ◆ Phân bố các hành vi của Subsystem cho các phần tử của nó
- ◆ Lập sơ liệu về các phần tử của Subsystem
- ◆ Mô tả các phụ thuộc của Subsystem

★ ◆ Checkpoints

Checkpoints: Thiết Subsystems

- ◆ Đã định nghĩa realization association cho mỗi interface đề nghị bởi subsystem?
- ◆ Đã định nghĩa một dependency association cho mỗi interface dùng bởi subsystem chưa?
- ◆ Hãy chắc chắn rằng, không có một phần tử nào của subsystem có public visibility.
- ◆ Mỗi operation trên một interface được hiện thực hóa bởi subsystem đã được mô tả trên một interaction diagram chưa? Nếu chưa, operation có được hiện thực hóa bởi một class đơn không? Khi đó, có dễ thấy đó là ánh xạ 1:1 giữa class operation và interface operation?

Nhắc lại: Thiết kế Subsystem Design

- ◆ Mục đích của thiết kế Subsystem là gì ?
- ◆ Cần tạo ra bao nhiêu interaction diagrams trong khi thiết kế Subsystem?
- ◆ Tại sao phải đặt các phụ thuộc của một subsystem lên subsystem interface?

Bài tập: Thiết kế Subsystem

- ◆ Cho biết các điều sau:
 - Định nghĩa các subsystem, các interface và các quan hệ của chúng với các phần tử thiết kế khác (các lược đồ trong ngữ cảnh của subsystem)

(còn tiếp)

Bài tập: Thiết kế Subsystem (tt.)

- ◆ Với một subsystem cụ thể, hãy xác định:
 - Các phần tử thiết kế chứa trong subsystem và các quan hệ của chúng
 - Các cơ chế kiến trúc khả dụng
 - Các interaction cần để cài đặt các subsystem interface operation

(còn tiếp)

Bài tập: Thiết kế Subsystem Design (tt.)

- ◆ Với một subsystem cụ thể, hãy tạo các lược đồ:
 - “Interface realizations”
 - Interaction diagram cho mỗi interface operation
 - Class diagram chứa các phần tử của subsystem hiện thực hóa các chức năng của interface và quan hệ của chúng
 - Class diagram thể hiện subsystem và tất cả các phụ thuộc lên các external package và subsystem (subclass dependencies class diagram)