



MÔ HÌNH HOÁ PHẦN MỀM

TUẦN 5: SEQUENCE DIAGRAM

GVLT: NGUYỄN THỊ MINH TUYỀN

NỘI DUNG

1. Giới thiệu
2. Các tương tác và các bên tương tác
3. Thông điệp
4. Các combined fragment
 - a. Branches and loops
 - b. Concurrency and order
 - c. Filters and assertions
5. Các thành phần khác
6. Các loại biểu đồ tương tác khác

NỘI DUNG

1. Giới thiệu

2. Các tương tác và các bên tương tác

3. Thông điệp

4. Các combined fragment

- a. Branches and loops
- b. Concurrency and order
- c. Filters and assertions

5. Các thành phần khác

6. Các loại biểu đồ tương tác khác

GIỚI THIỆU

- Mô hình hoá hành vi giữa các đối tượng (inter-object behavior)

- Tương tác

- Chỉ ra cách các thông điệp (message) và dữ liệu (data) được trao đổi giữa các thành phần tương tác với nhau.

- Các bên tương tác (interaction partners)

- Human (student, lecturer, administrator, ...)
- Non-human (server, printer, executable software, ...)

- Ví dụ:

- Conversation between persons
- Message exchange between humans and a software system
- Communication protocols
- Sequence of method calls in a program
- ...

NỘI DUNG

1. Giới thiệu

2. Các tương tác và các bên tương tác

3. Thông điệp

4. Các combined fragment

a. Branches and loops

b. Concurrency and order

c. Filters and assertions

5. Các thành phần khác

6. Các loại biểu đồ tương tác khác

CÁC BIỂU ĐỒ TƯƠNG TÁC

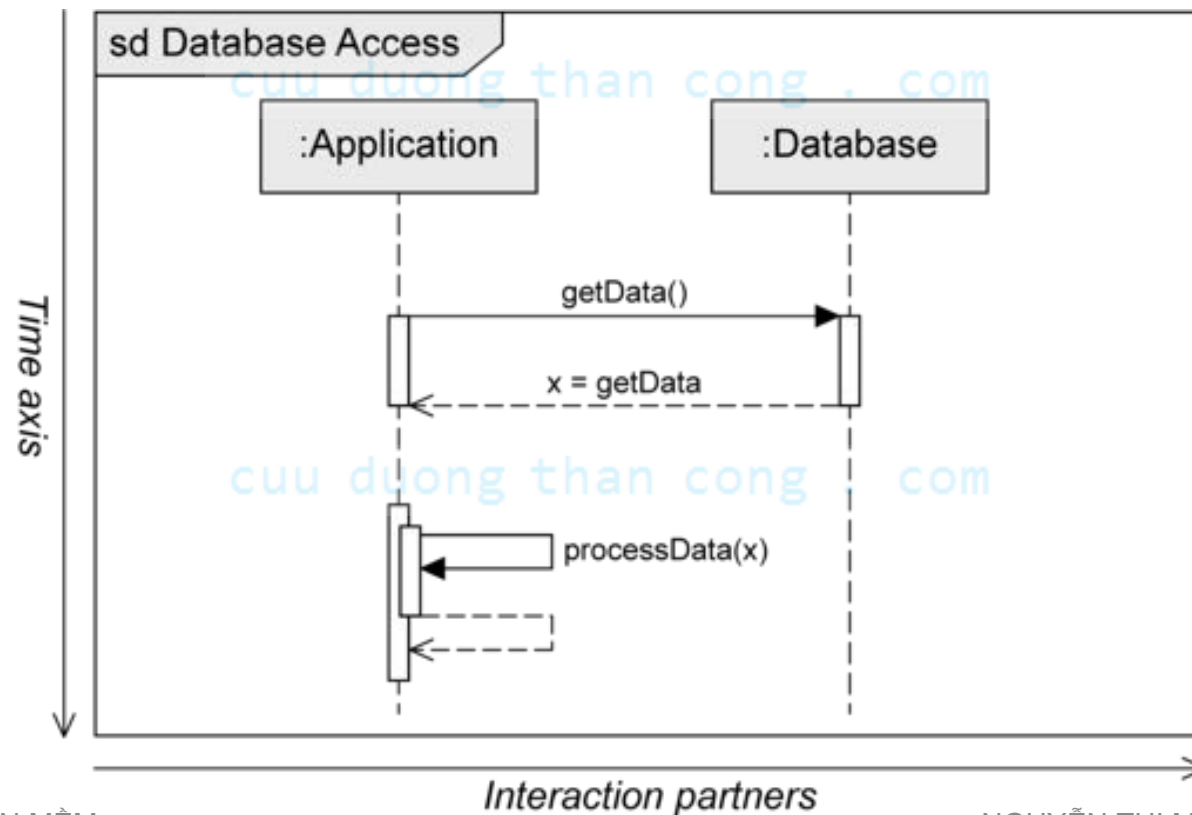
- Được dùng để chỉ ra các tương tác
- Mô hình hoá các kịch bản cụ thể
- Mô tả các chuỗi giao tiếp tại các mức chi tiết khác nhau
- Các biểu đồ tương tác biểu diễn:
 - Tương tác của một hệ thống trong môi trường của nó
 - Tương tác giữa các phần hệ thống để biểu diễn cách mà một use case có thể được cài đặt
 - Giao tiếp giữa các tiến trình trong đó các thành phần tương tác phải tuân thủ một số giao thức nhất định
 - Giao tiếp tại mức lớp (triệu gọi thao tác, hành vi giữa các đối tượng)
- Biểu đồ tuần tự là biểu đồ tương tác thông dụng nhất.

BIỂU ĐỒ TUẦN TỰ

- Sequence Diagram

- Biểu đồ hai chiều

- Tương tác = chuỗi các đặc tả sự kiện



CÁC BÊN TƯỞNG TÁC

Lifeline

r:C

- Các bên tương tác (Interaction partners) được mô tả dưới dạng vòng đời (lifeline)
- Phần đầu của vòng đời:
 - Biểu thức chứa **roleName:Class**
 - Vai trò là khái niệm tổng quát hơn các đối tượng
 - Đối tượng có thể có nhiều vai trò khác nhau trong suốt vòng đời
- Phần thân của vòng đời:
 - Chiều đứng, biểu diễn bằng đường nét đứt
 - Biểu diễn lifetime của đối tượng liên kết với nó

Head of the lifeline →

lecturer

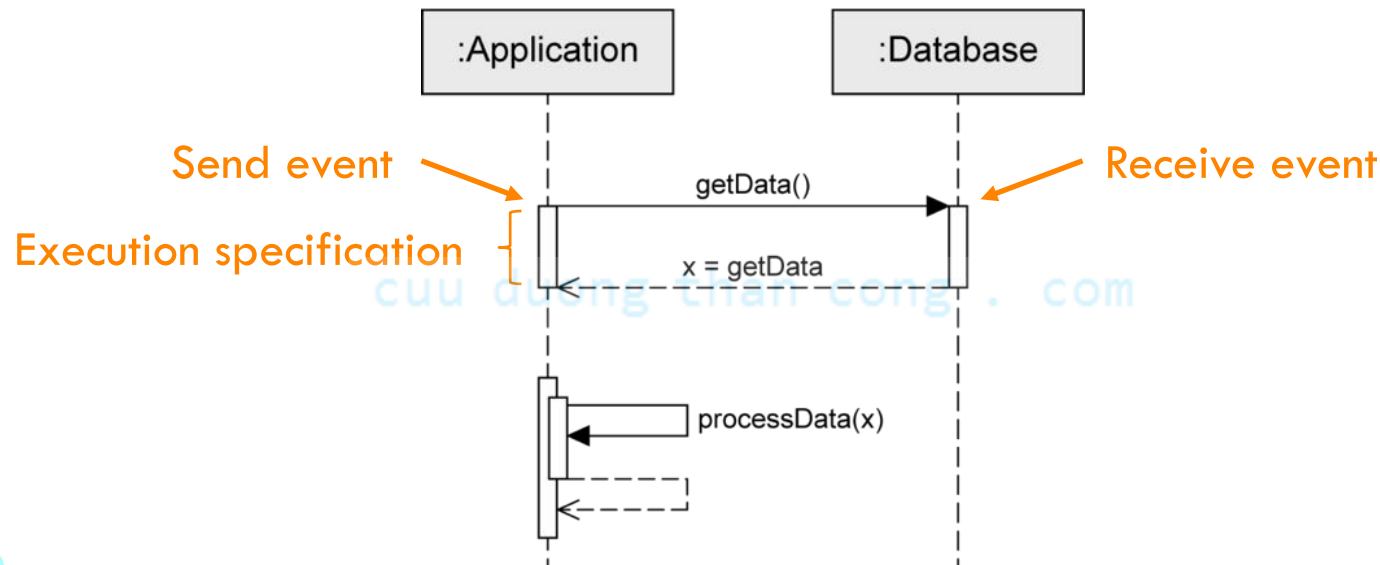
:Professor

lecturer
:Professor

Body of the Lifeline →

TRAO ĐỔI THÔNG ĐIỆN (1/2)

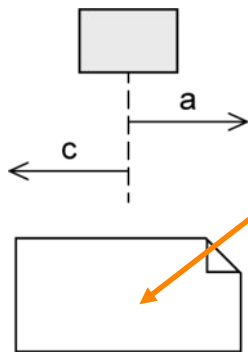
- Tương tác được xem như một chuỗi các đặc tả sự kiện
- Đặc tả sự kiện bao gồm việc gửi và nhận thông điệp hoặc các sự kiện xảy ra dựa vào thời gian
- Đặc tả việc thực thi (Execution specification)
 - Continuous bar
 - Được dùng để hiển thị khi một thành phần tương tác thực hiện một số hành vi



TRAO ĐỔI THÔNG điệp (2/2)

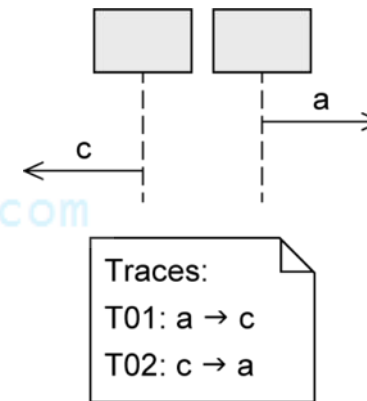
• Thứ tự của các thông điệp

... on one lifeline

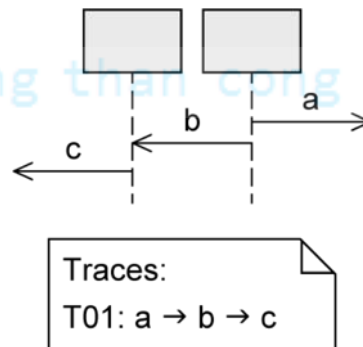


"Happens before"

... on different lifelines



... on different lifelines which exchange messages

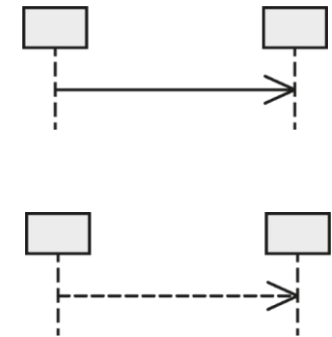
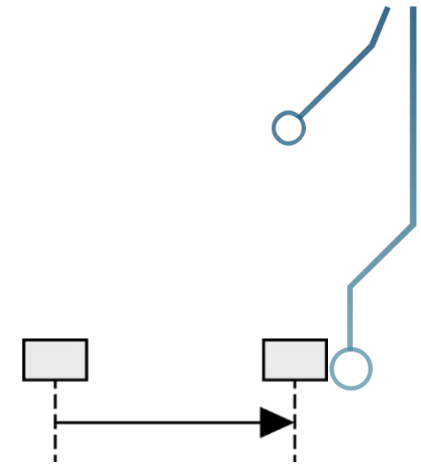


NỘI DUNG

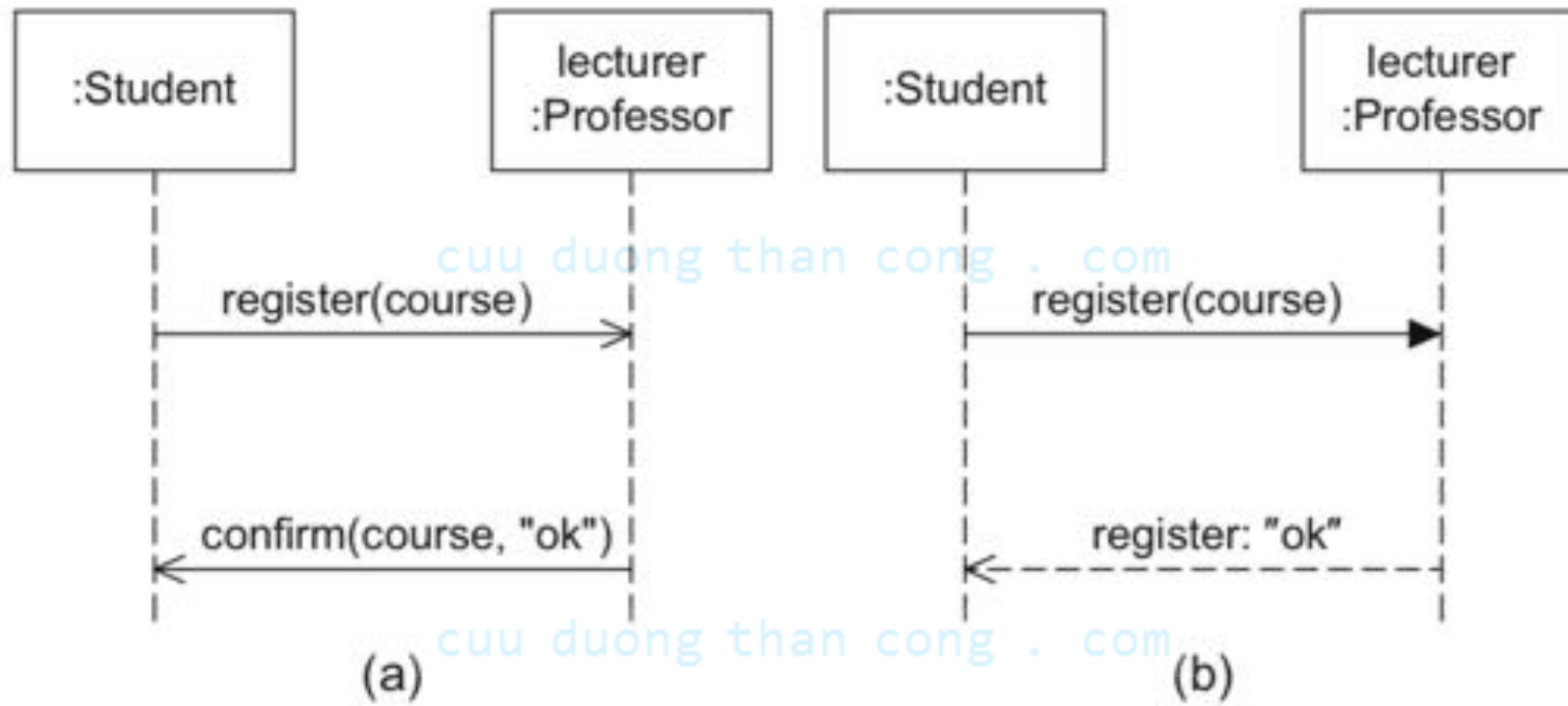
1. Giới thiệu
2. Các tương tác và các bên tương tác
3. Thông điệp
4. Các combined fragment
 - a. Branches and loops
 - b. Concurrency and order
 - c. Filters and assertions
5. Các thành phần khác
6. Các loại biểu đồ tương tác khác

THÔNG ĐIỆN (1/3)

- Thông điệp đồng bộ (Synchronous message)
 - Bên gửi thông điệp chờ cho tới khi nhận được một thông điệp trả lời trước khi tiếp tục
 - Cú pháp của tên thông điệp: **msg(par1, par2)**
 - **msg**: tên thông điệp
 - **par**: tham số cách nhau bởi dấu
- Thông điệp bất đồng bộ (Asynchronous message)
 - Bên gửi thông điệp không cần chờ thông điệp trả lời
 - Cú pháp của tên thông điệp : **msg(par1, par2)**
- Thông điệp trả lời (Response message)
 - Có thể bỏ qua nếu nội dung và điểm mà thông điệp trả về được gửi và nhận có ngữ cảnh rõ ràng
 - Cú pháp: **att=msg(par1, par2):val**
 - **att**: giá trị trả về có thể gán cho một biến nào đó
 - **msg**: tên thông điệp
 - **par**: các tham số cách nhau bởi dấu ,
 - **val**: giá trị trả về



VÍ DỤ



THÔNG ĐIỆN (2/3)

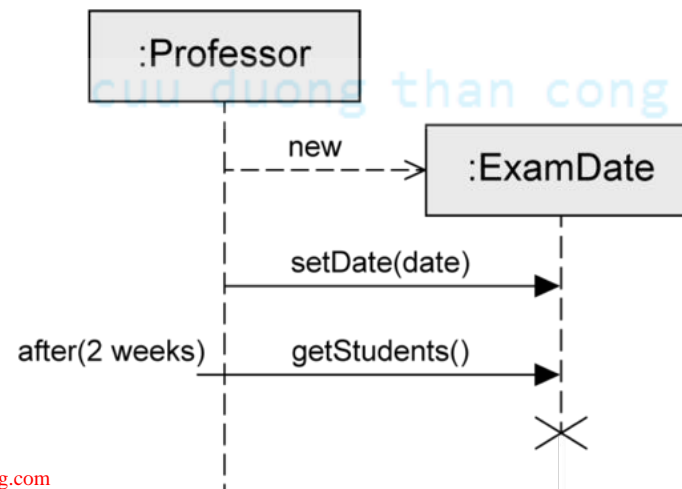
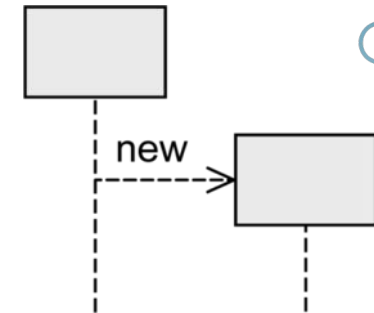
- Tạo đối tượng

- Đường mũi tên nét đứt
- Đầu mũi tên hướng từ đầu của lifeline của đối tượng được tạo

- Từ khoá **new**

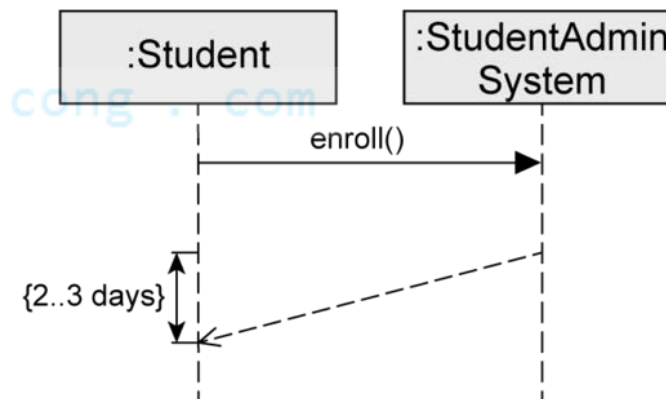
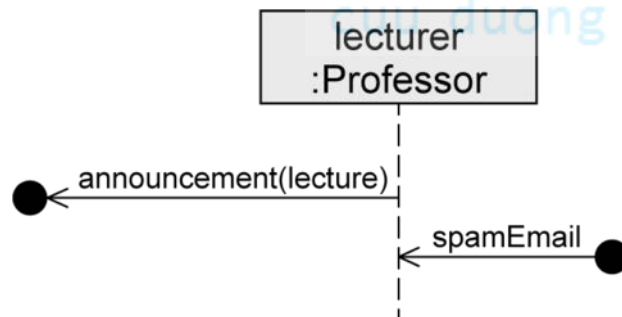
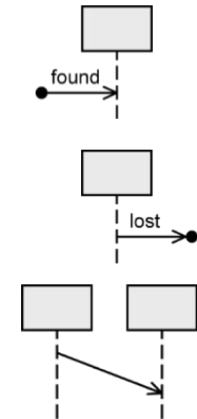
- Huỷ đối tượng

- Đối tượng bị xoá bỏ
- Dấu chéo (×) tại điểm cuối của lifeline



THÔNG ĐIỆNP (3/3)

- Found message
 - Bên gửi thông điệp không biết hoặc không liên quan
- Lost message
 - Bên nhận thông điệp không biết hoặc không liên quan
- Thông báo tiêu tốn thời gian (Time-consuming message)
 - "Thông điệp với một khoảng thời gian"
 - Thường các thông điệp được giả sử truyền đi mà không mất thời gian
 - Biểu diễn thời gian giữa gửi và nhận thông điệp



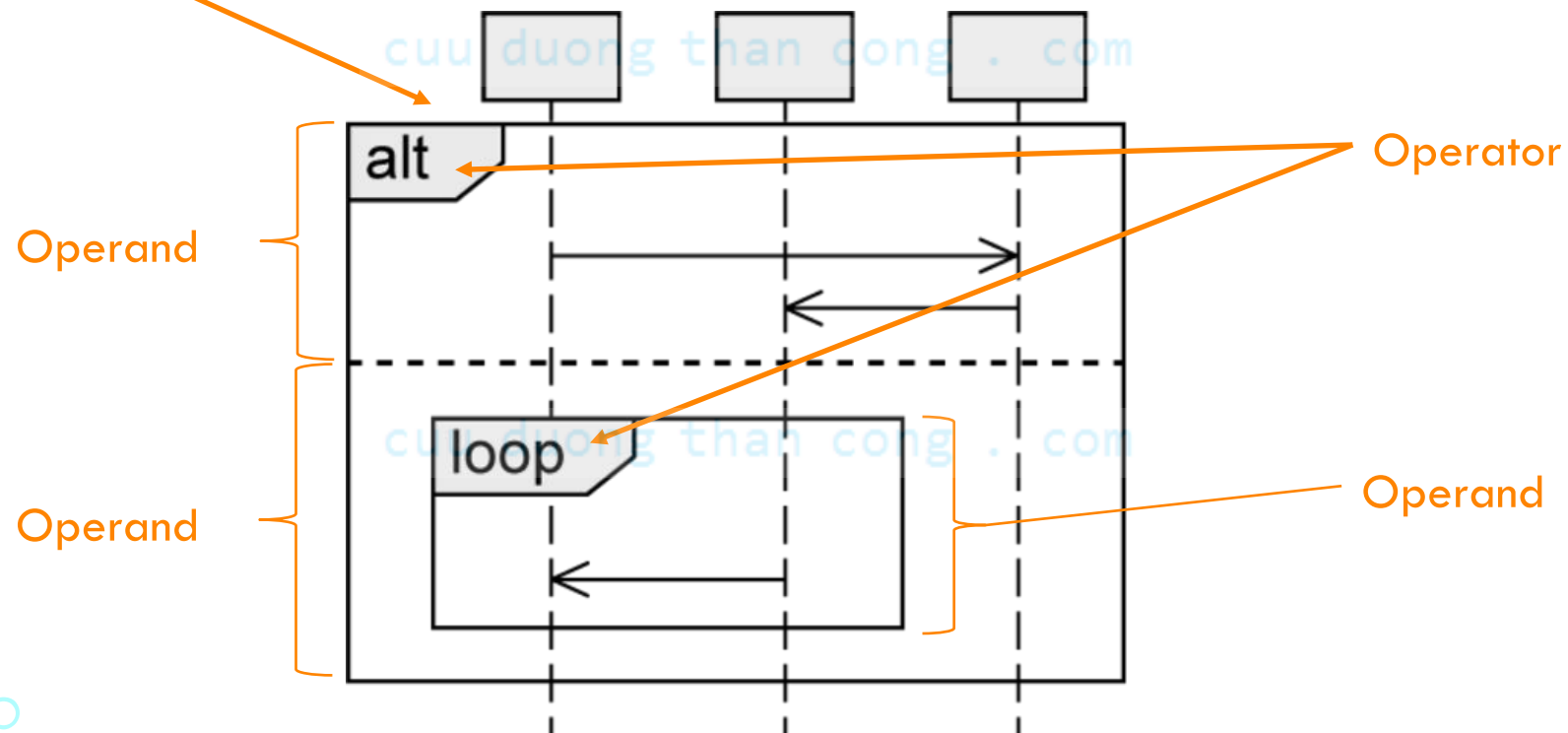
NỘI DUNG

1. Giới thiệu
2. Các tương tác và các bên tương tác
3. Thông điệp
4. Các combined fragment
 - a. Branches and loops
 - b. Concurrency and order
 - c. Filters and assertions
5. Các thành phần khác
6. Các loại biểu đồ tương tác khác

COMBINED FRAGMENTS

- Mô hình các cấu trúc điều khiển khác nhau
- Có 12 loại toán tử (operator) được định nghĩa trước

Combined Fragment



CÁC LOẠI COMBINED FRAGMENTS

	Operator	Purpose
Branches and loops	alt	Alternative interaction
	opt	Optional interaction
	loop	Repeated interaction
	break	Exception interaction
Concurrency and order	seq	Weak order
	strict	Strict order
	par	Concurrent interaction
	critical	Atomic interaction
Filters and assertions	ignore	Irrelevant interaction
	consider	Relevant interaction
	assert	Asserted interaction
	neg	Invalid interaction

NỘI DUNG

1. Giới thiệu

2. Các tương tác và các bên tương tác

3. Thông điệp

4. Các combined fragment

a. Branches and loops

b. Concurrency and order

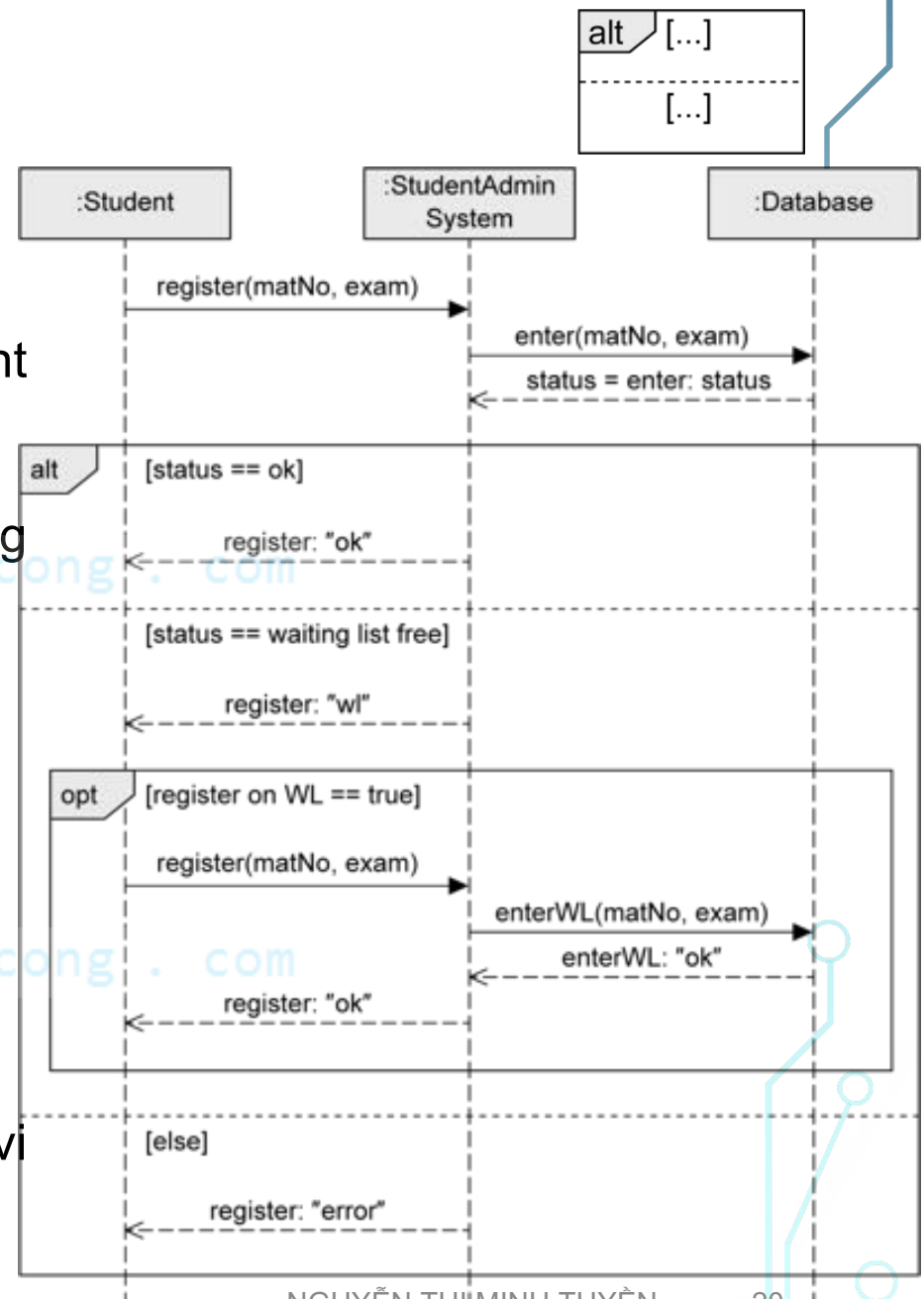
c. Filters and assertions

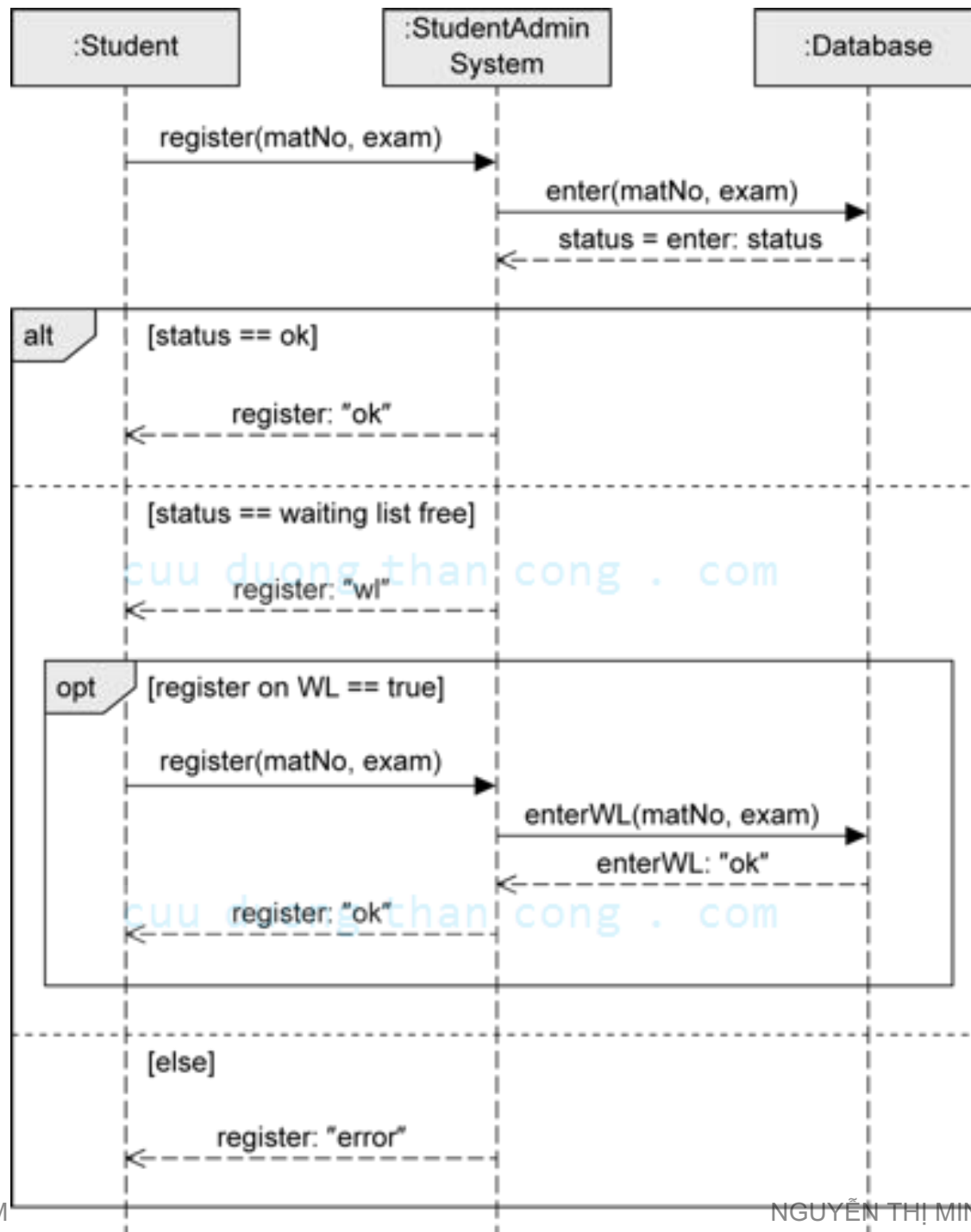
5. Các thành phần khác

6. Các loại biểu đồ tương tác khác

alt FRAGMENT

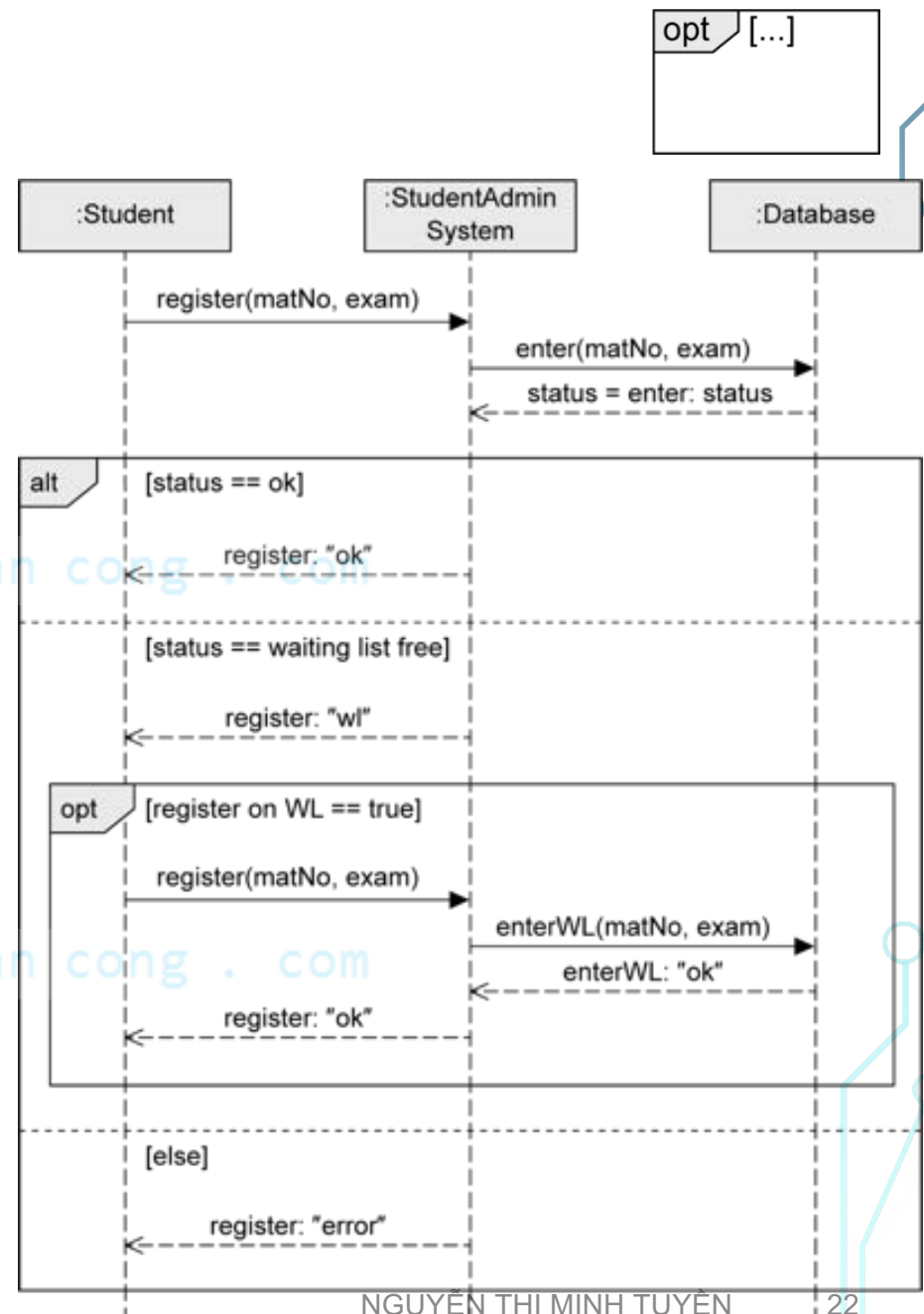
- Biểu diễn các chuỗi thay thế
- Tương tự câu lệnh switch statement trong Java
- Guards được dùng để chọn một đường đi được thực thi
 - Được đặt giữa hai dấu []
 - mặc định: true
 - đã định nghĩa trước: [else]
- Nhiều toán hạng
- Guards phải rời rạc để tránh hành vi không xác định



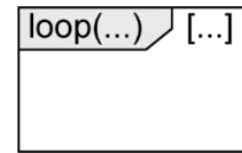


opt FRAGMENT

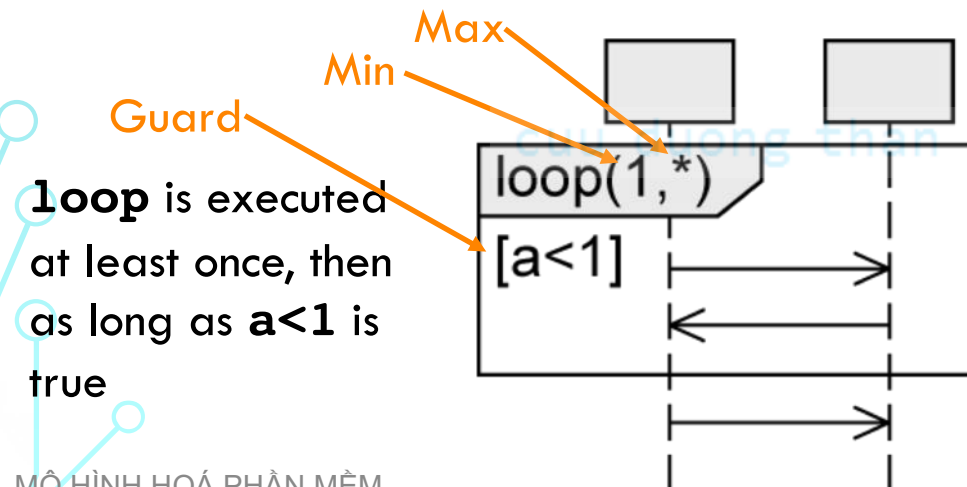
- Để mô hình hoá một chuỗi tùy chọn
- Việc thực thi thực sự tại thời điểm chạy phụ thuộc vào guard
- Có chính xác một toán hạng
- Tương tự câu lệnh **if** mà không có nhánh **else**
- Tương đương với **alt** fragment với hai toán hạng, một cái bỏ trống



loop FRAGMENT



- Để biểu diễn rằng một chuỗi được thực thi lặp lại
- Chính xác một toán hạng
- Từ khoá loop được theo sau bởi các số vòng lặp min và max (**min..max**) hoặc (**min,max**)
 - mặc định: (*) .. không có giới hạn trên
- Guard
 - Được ước lượng ngay khi số lần lặp tối thiểu được tiến hành
 - Được kiểm tra cho mỗi lần lặp trong giới hạn (**min,max**)
 - Nếu guard sai, việc thực thi sẽ bị kết thúc.



Các ký hiệu thay thế:

$\text{loop}(3, 8) = \text{loop}(3..8)$
 $\text{loop}(8, 8) = \text{loop}(8)$
 $\text{loop} = \text{loop} (*) = \text{loop}(0, *)$

break FRAGMENT

- Hình thức xử lý ngoại lệ đơn giản
- Chính xác một toán hạng với một guard
- Nếu guard đúng :
 - Các tương tác trong toán hạng được thực thi
 - Các thao tác còn lại quanh fragment bị bỏ qua
 - Tương tác tiếp tục ở fragment có level cao hơn
 - Hành vi khác với **opt** fragment

- Chính xác một toán hạng với một guard

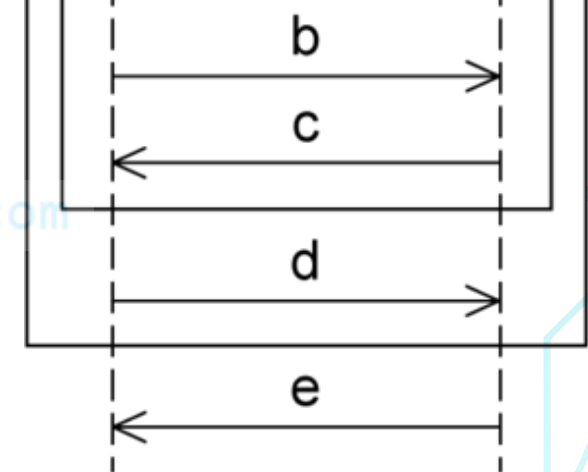
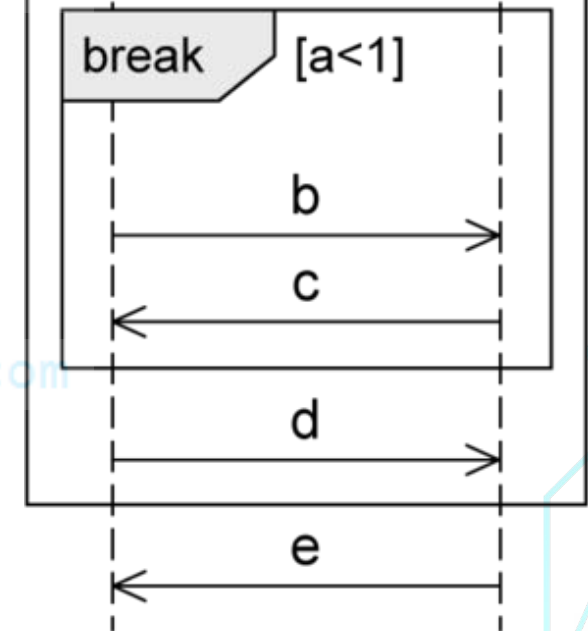
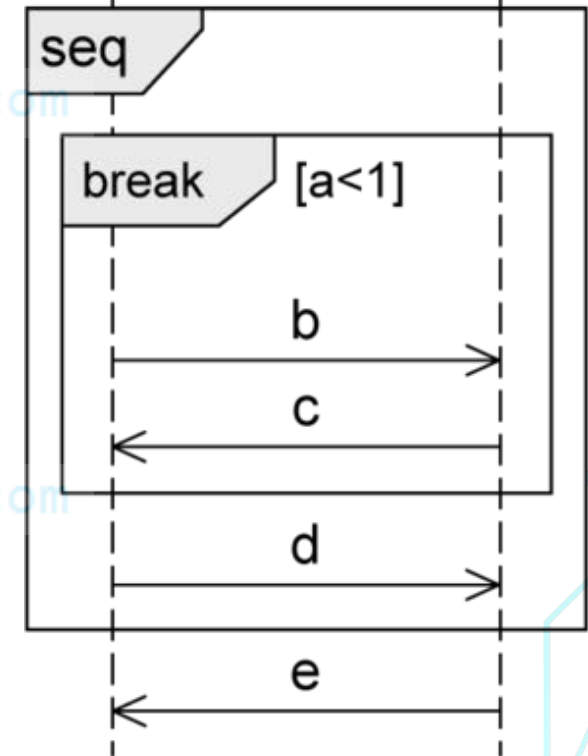
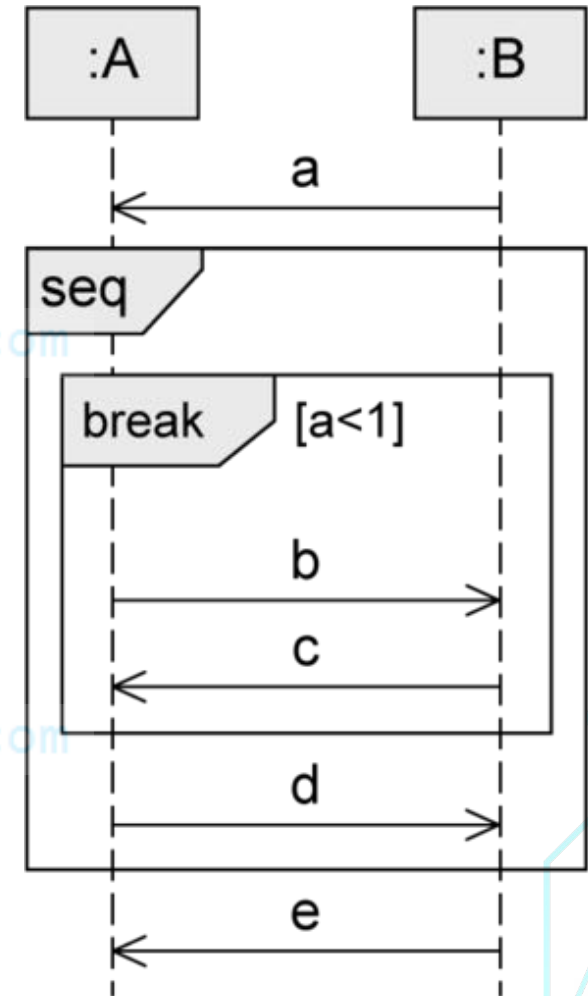
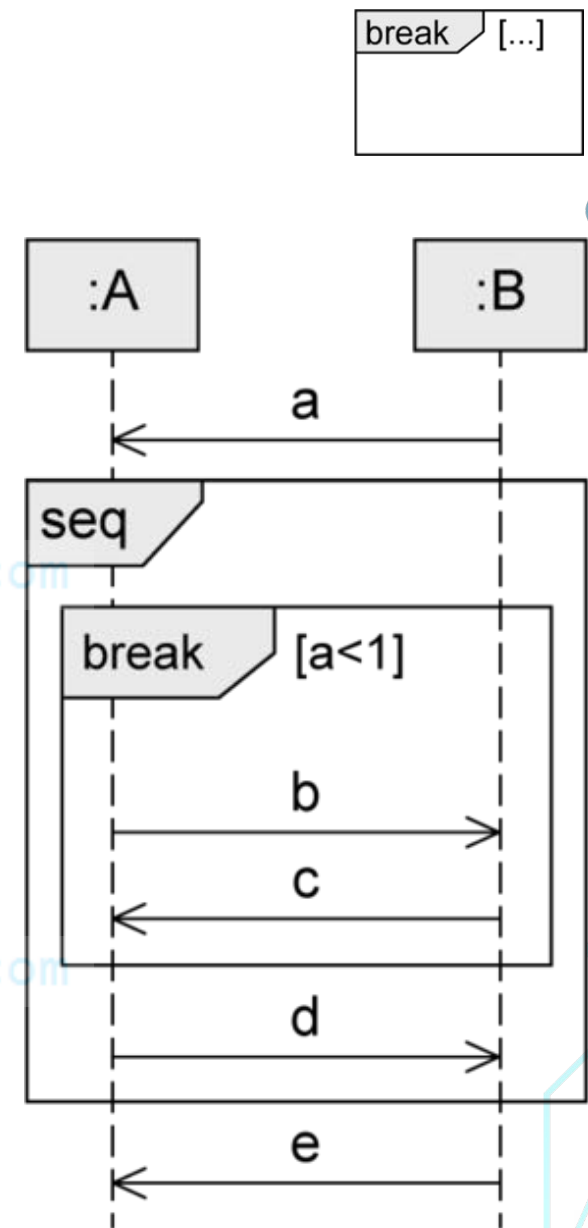
- Nếu guard đúng :

- Các tương tác trong toán hạng được thực thi
- Các thao tác còn lại quanh fragment bị bỏ qua
- Tương tác tiếp tục ở fragment có level cao hơn
- Hành vi khác với **opt** fragment

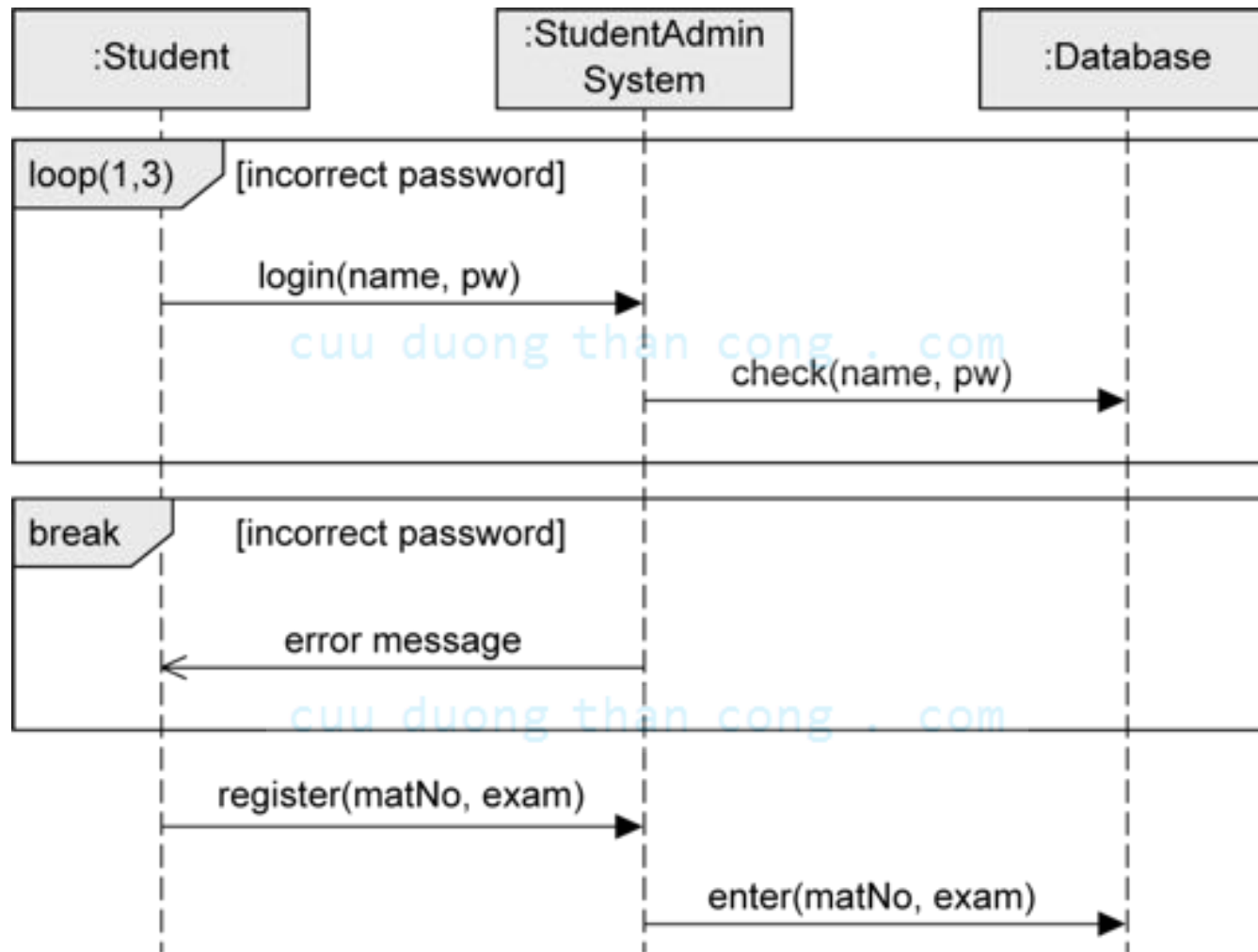
- Các thao tác còn lại quanh fragment bị bỏ qua

- Tương tác tiếp tục ở fragment có level cao hơn

- Hành vi khác với **opt** fragment



loop VÀ break FRAGMENT – VÍ DỤ



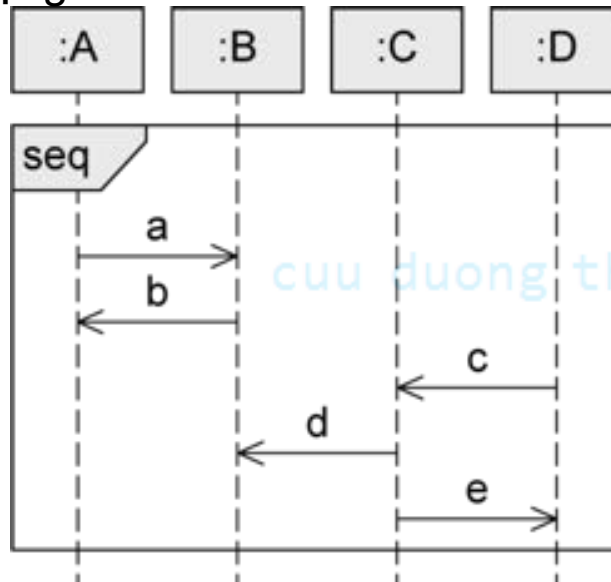
NỘI DUNG

1. Giới thiệu
2. Các tương tác và các bên tương tác
3. Thông điệp
- 4. Các combined fragment**
 - a. Branches and loops
 - b. Concurrency and order
 - c. Filters and assertions
5. Các thành phần khác
6. Các loại biểu đồ tương tác khác

seq FRAGMENT

- Biểu diễn thứ tự mặc định
- Trình tự yếu:

1. Thứ tự các sự kiện trong mỗi toán hạng được duy trì trong kết quả.
2. Các sự kiện trong các lifeline từ các toán hạng khác nhau có thể theo thứ tự bất kỳ.
3. Các sự kiện trong cùng lifeline từ các toán hạng khác nhau được sắp xếp sao cho một sự kiện của toán hạng thứ nhất sẽ xảy ra trước sự kiện của toán hạng thứ 2.



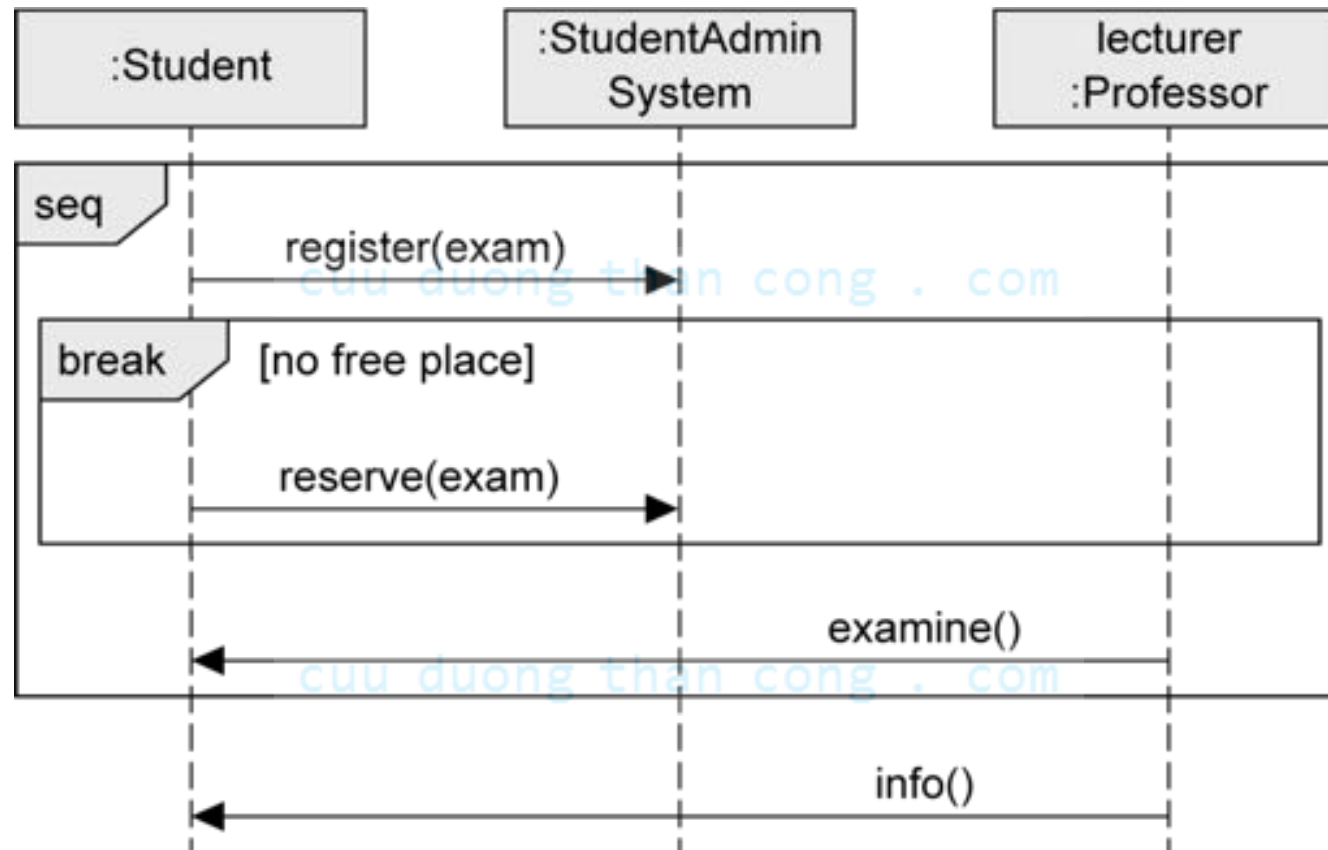
Traces:

T01: a → b → c → d → e

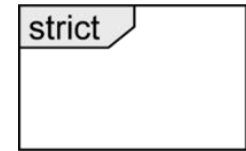
T02: a → c → b → d → e

T03: c → a → b → d → e

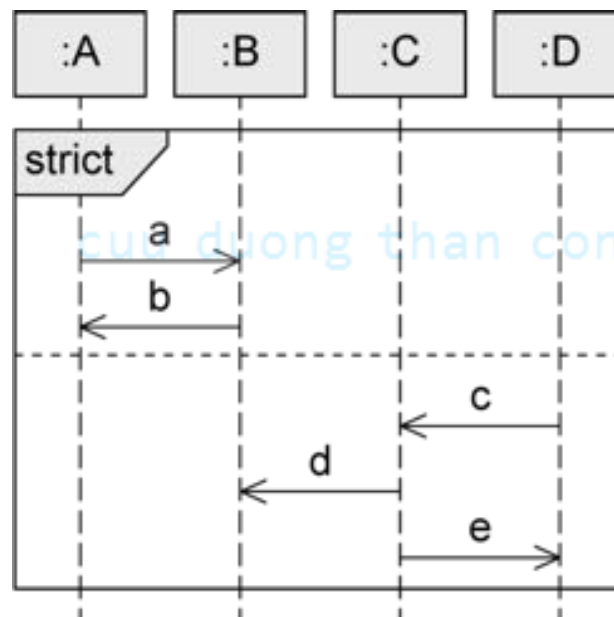
seq FRAGMENT – VÍ DỤ



strict FRAGMENT

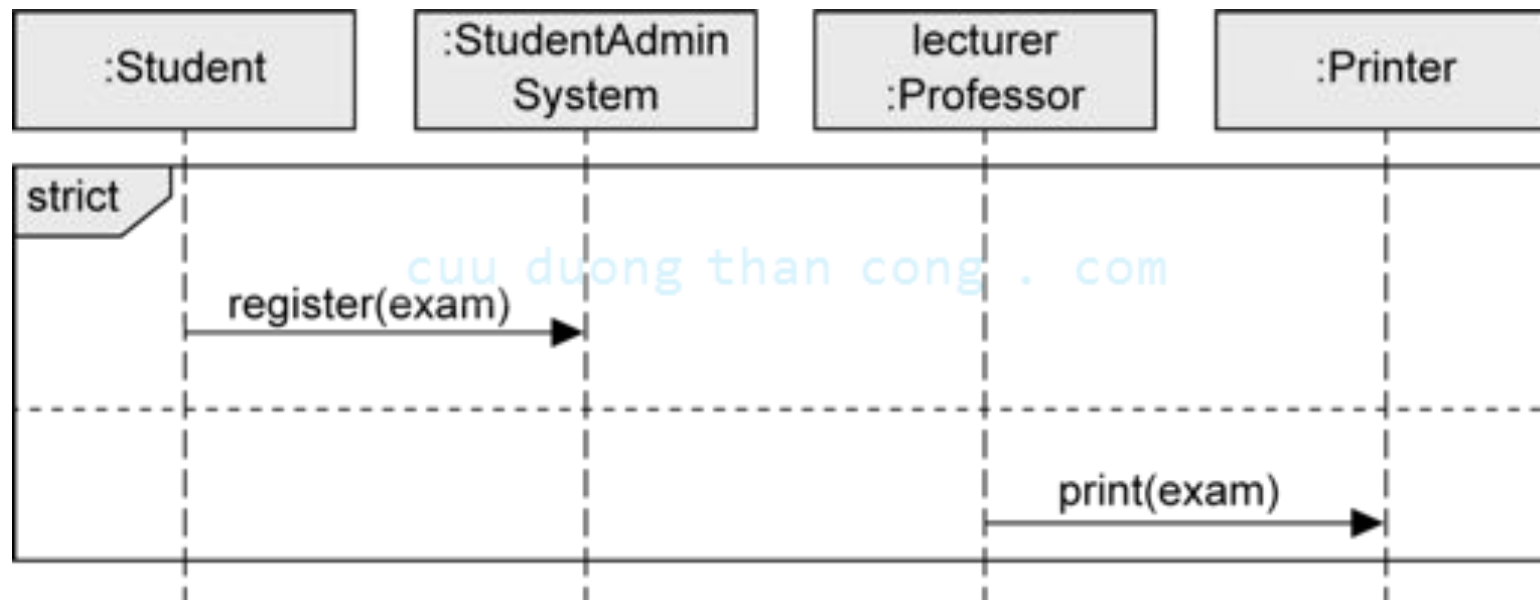


- Tương tác tuần tự với thứ tự
- Thứ tự của sự kiện diễn ra trên các lifeline khác nhau giữa các toán hạng khác nhau là quan trọng.
 - Các thông điệp trong một toán hạng cao hơn trên trục tung luôn luôn được trao đổi trước khi các thông điệp của toán hạng ở thấp hơn

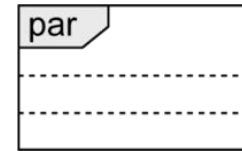


Traces:
T01: $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$

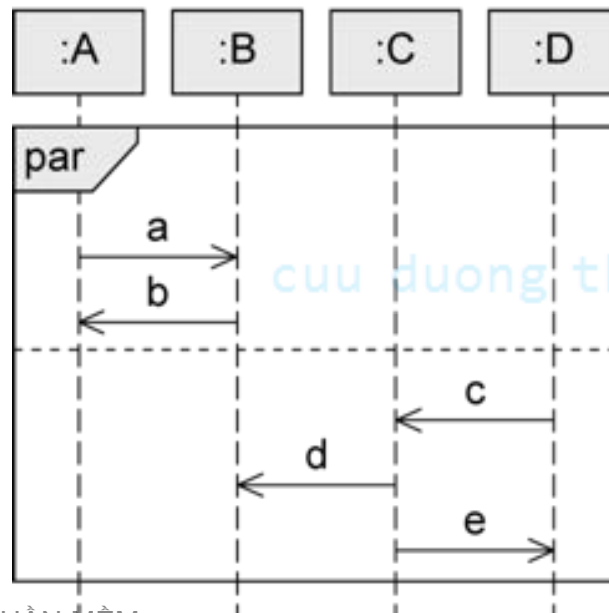
strict FRAGMENT – VÍ DỤ



par FRAGMENT



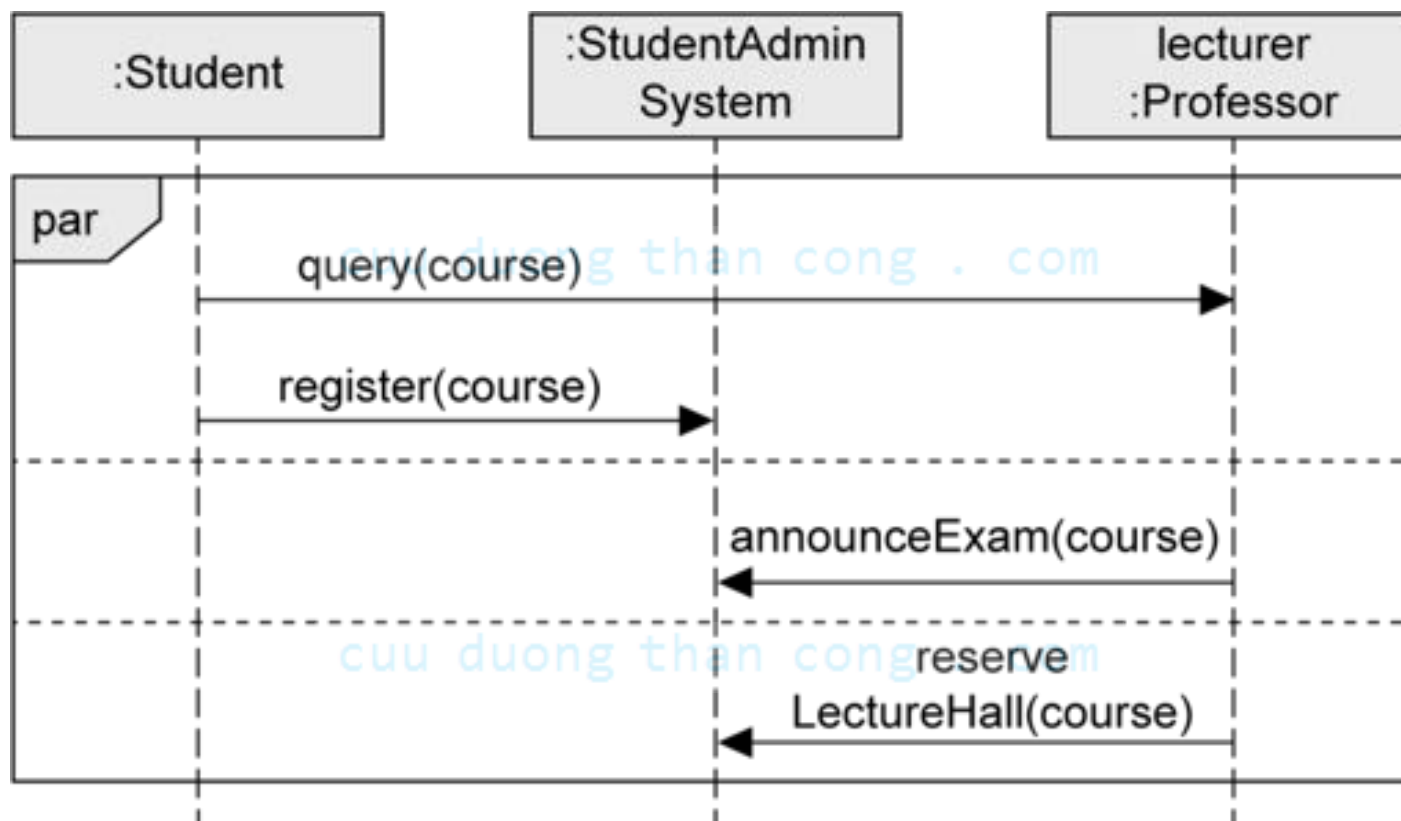
- Cho phép bỏ qua thứ tự thời gian giữa các thông điệp khác nhau trong các toán hạng khác nhau.
- Các đường thực thi của toán hạng khác nhau có thể đan xen nhau miễn là các ràng buộc của riêng mỗi toán hạng được tôn trọng
- Thứ tự của các toán hạng khác nhau không liên quan nhau
- Đồng thời, không song song thực sự, chỉ diễn tả sự đồng thời



Traces:

T01: a → b → c → d → e
T02: a → c → b → d → e
T03: a → c → d → b → e
T04: a → c → d → e → b
T05: c → a → b → d → e
T06: c → a → d → b → e
T07: c → a → d → e → b
T08: c → d → a → b → e
T09: c → d → a → e → b
T10: c → d → e → a → b

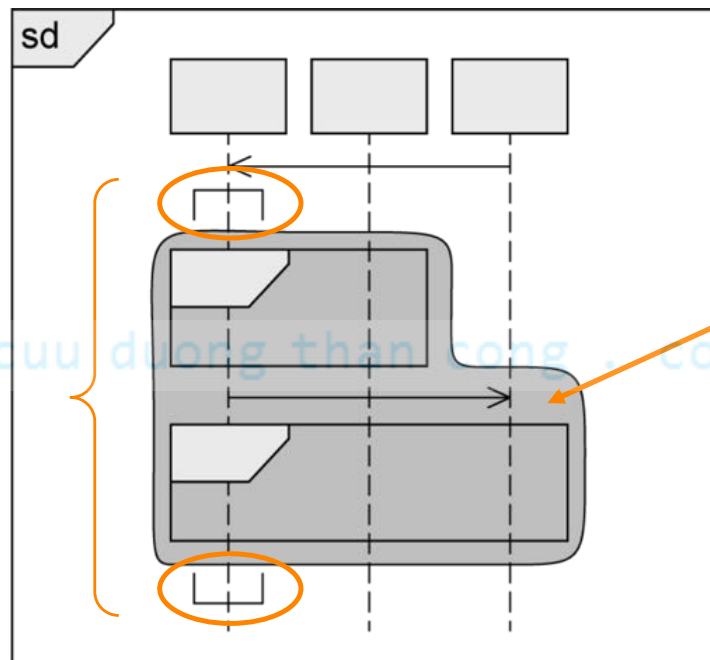
par FRAGMENT – VÍ DỤ



COREGION

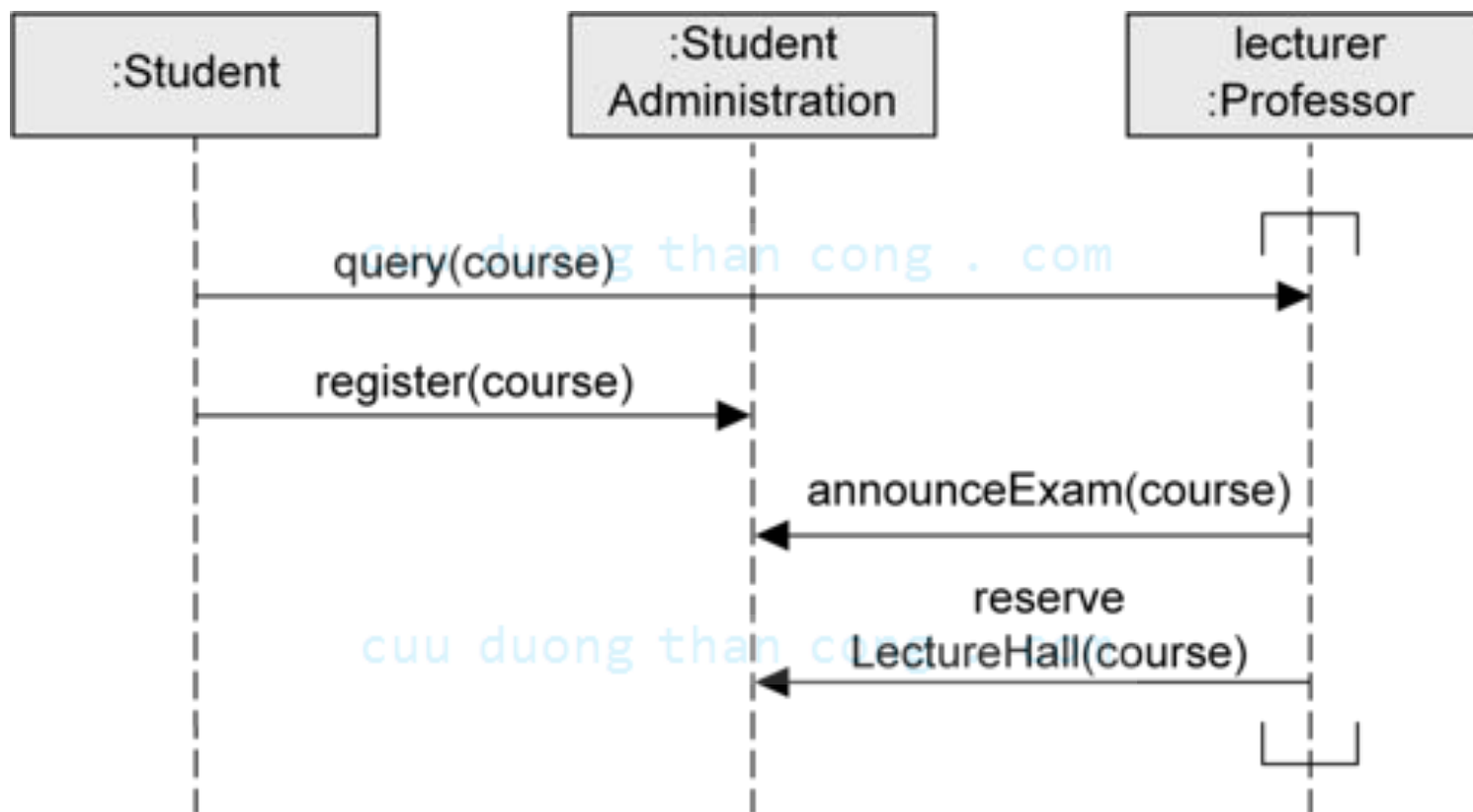
- Để mô hình hoá các sự kiện đồng thời của một lifeline duy nhất
- Thứ tự xảy ra sự kiện trong một coregion không bị hạn chế
- Khu vực của lifeline được bao phủ bởi coregion được đánh dấu bởi hai dấu [] xoay ngang 90 độ

Coregion

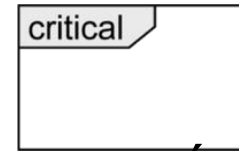


Impact of
coregion

COREGION – VÍ DỤ

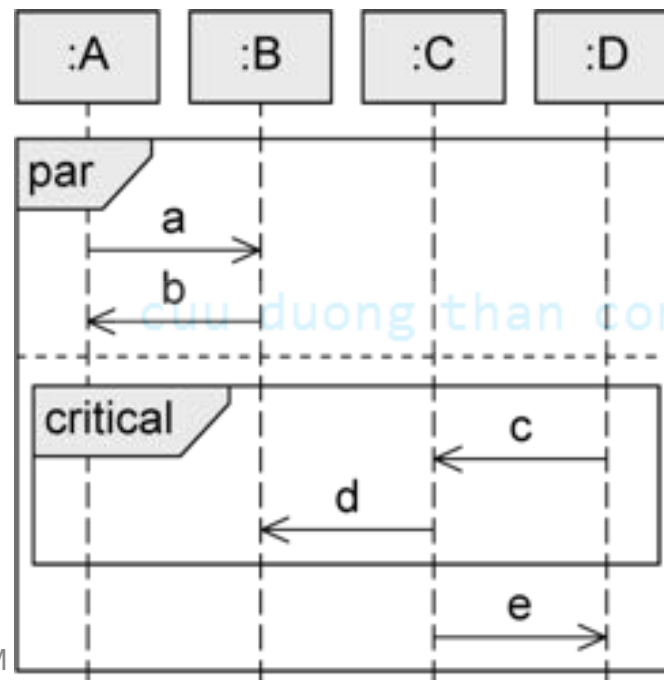


critical FRAGMENT



- Để đảm bảo rằng các phần nào đó của một tương tác không bị ngắt bởi các sự kiện không mong đợi
- Đặt trong **critical**: thứ tự mặc định **seq**

cuu duong than cong . com



Traces:

T01: a → b → c → d → e
T02: a → c → d → b → e
T03: a → c → d → e → b
T04: c → d → a → b → e
T05: c → d → a → e → b
T06: c → d → e → a → b

critical FRAGMENT – VÍ DỤ

