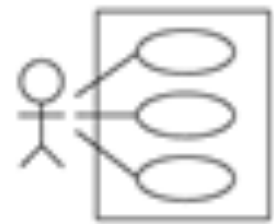


Sequence Diagram

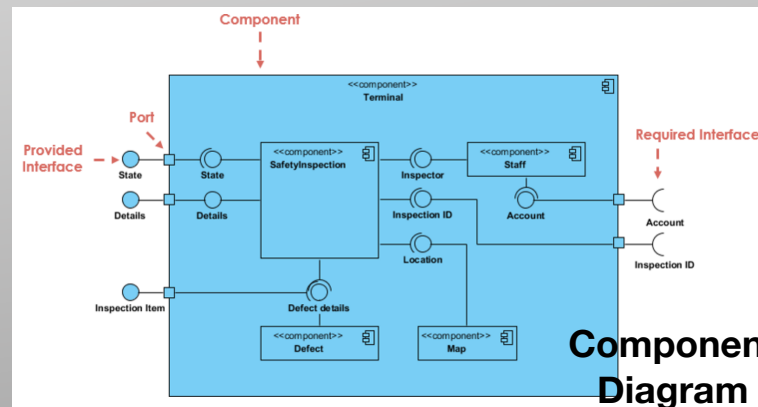
Jin Hyun Kim

UML Models for SLC



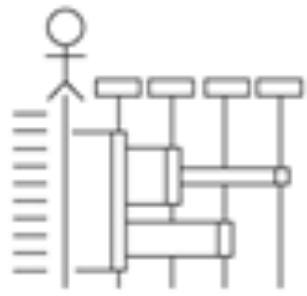
Use Case Model

Requirement
Elicitation and
Analysis

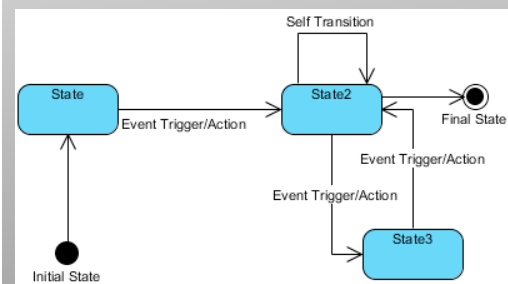


Component Diagram

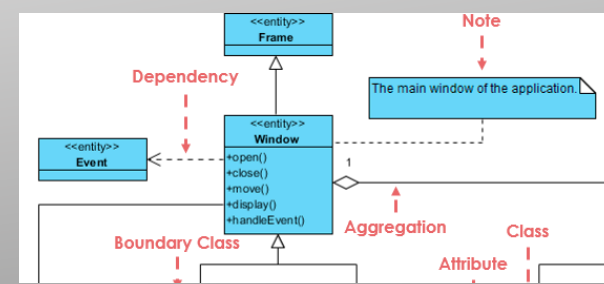
Architecture and
Component Models



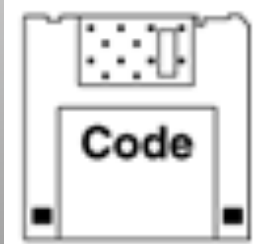
Sequence Diagram



Statecharts Diagram



Class Diagram



Implementation

Design

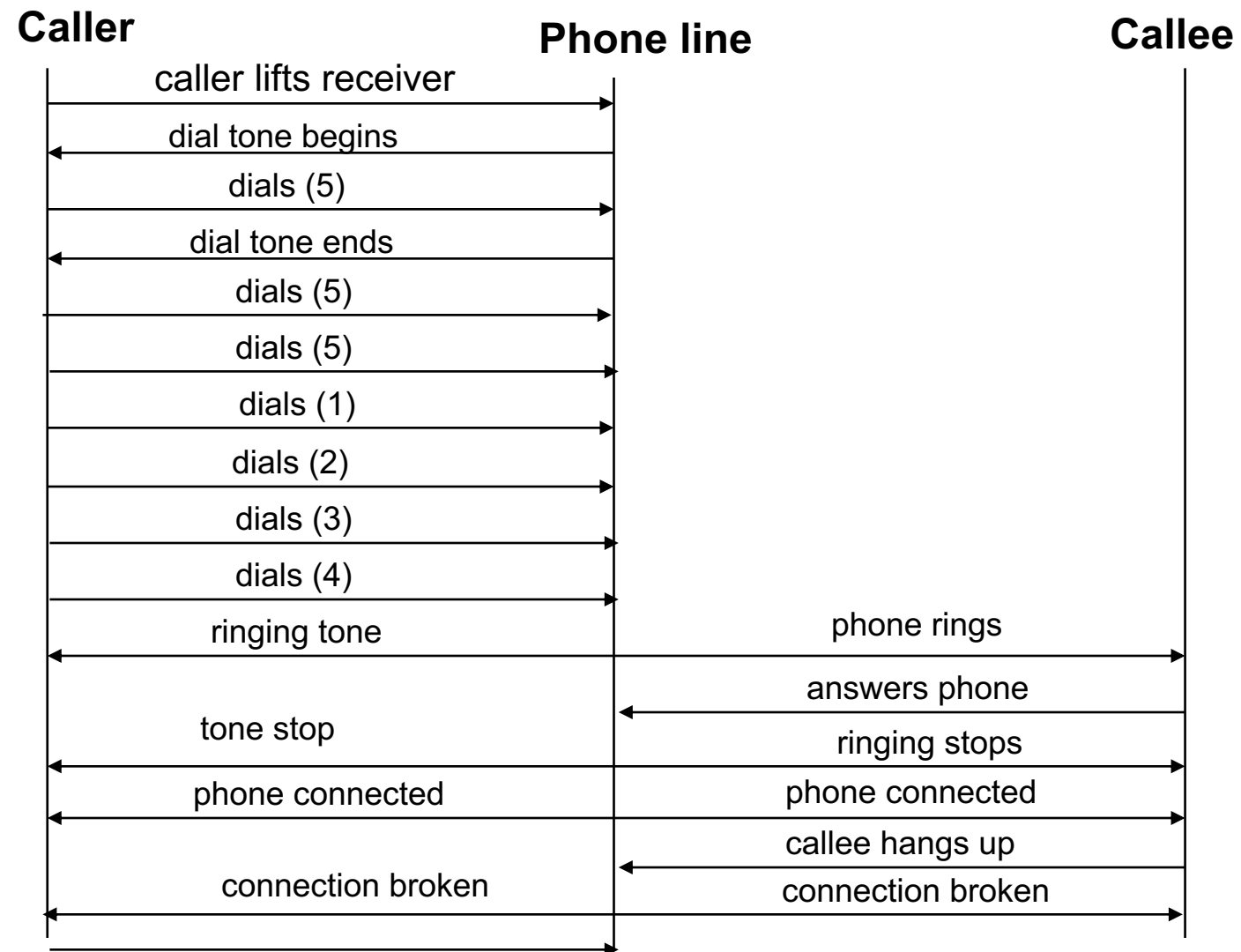
Dynamic Behavior

- Interaction diagram, e.g., sequence diagram, is a scenario-based model that captures how objects or components interact and communicate to achieve functionalities
 - Typically captures the behavior of a single use case
- State-based diagram, e.g., state charts diagram, captures a full behavior of a component in terms of state-transitions

Scenarios

- A scenario is a sequence of events that occurs during one particular execution of a system (시나리오는 시스템이 수행되는 동안 발생하는 일련의 이벤트들의 모임)
- A scenario can include all events in the system or can only include events that are generated by a certain object in that system (시스템의 모든 이벤트를 포함하거나 시스템의 특정 객체가 만드는 이벤트 만을 포함할 수 있음)
- A scenario can be **a historical record of executing** or **simulating the execution of a system or an object** (시나리오는 시스템이나 객체를 실행 혹은 모의시험한 기록)

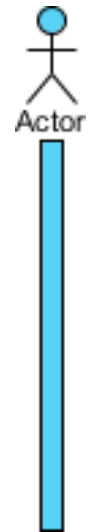
Event Trace Diagram



Sequence Charts

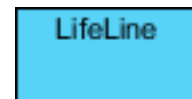
- A sequence shows a series of messages exchanged by
 - A selected set of objects in **temporally limited situation**
 - With emphasis on the chronological (시간순서) course of events.
- Objects are shown by vertical lifelines.
 - Highlights The chronological sequence of the message.
- Time runs from top to bottom.

Syntax



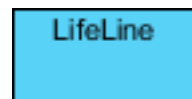
Actor

- a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data)
- external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject).
- represent roles played by human users, external hardware, or other subjects.



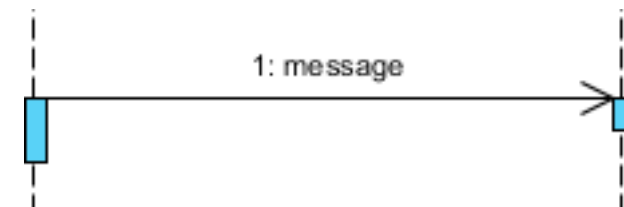
Lifeline

- A lifeline represents an individual participant in the Interaction.



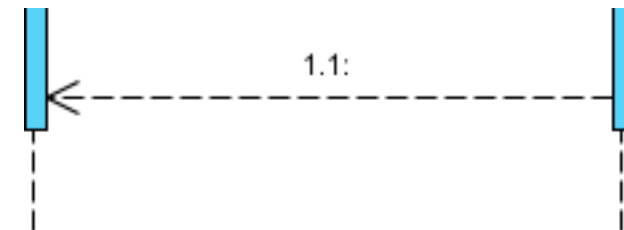
Activations

- A thin rectangle on a lifeline represents the period during which an element is performing an operation.
- The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



Call Message

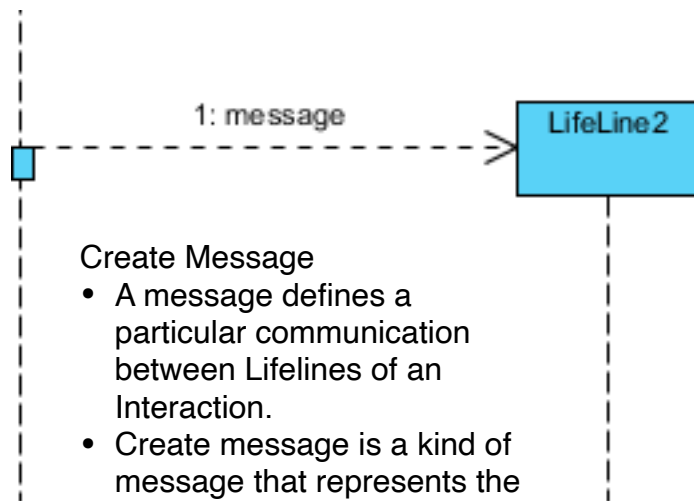
- A message defines a particular communication between Lifelines of an Interaction.
- Call message is a kind of message that represents an invocation of operation of target lifeline



Return Message

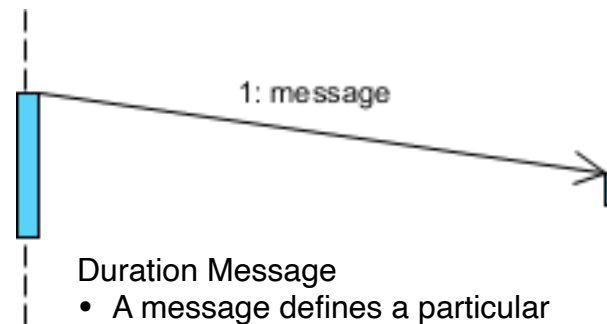
- A message defines a particular communication between Lifelines of an Interaction.
- Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.

Syntax



Create Message

- A message defines a particular communication between Lifelines of an Interaction.
- Create message is a kind of message that represents the instantiation of (target) lifeline.



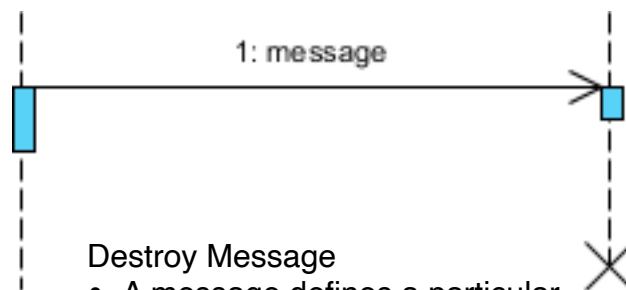
Duration Message

- A message defines a particular communication between Lifelines of an Interaction.
- Duration message shows the distance between two time instants for a message invocation.



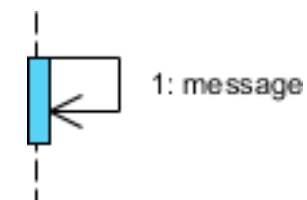
Note

- A note (comment) gives the ability to attach various remarks to elements.
- A comment carries no semantic force, but may contain information that is useful to a modeler.



Destroy Message

- A message defines a particular communication between Lifelines of an Interaction.
- Destroy message is a kind of message that represents the request of destroying the lifecycle of target lifeline.



Self Message

- A kind of message that represents the invocation of message of the same lifeline.

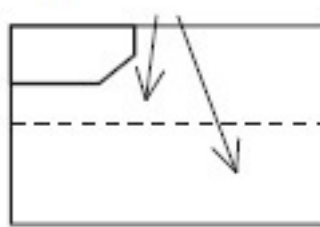
Syntax 3

Combined fragment

Operator



Operands

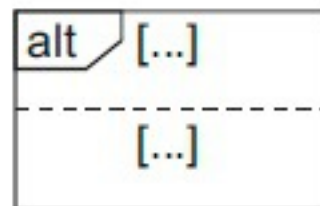


	Operator	Purpose
Branches and loops	alt opt loop break	Alternative interaction Optional interaction Iterative interaction Exception interaction
Concurrency and order	seq strict par critical	Weak order Strict order Concurrent interaction Atomic interaction
Filters and assertions	ignore consider assert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

Syntax

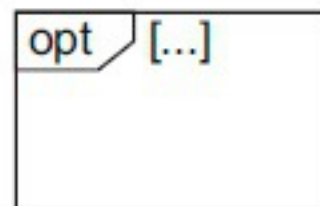
Branches and Loops

alt fragment



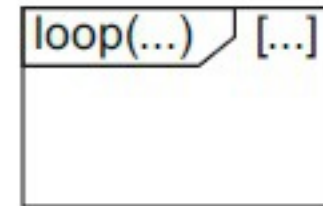
Alternative interaction

opt fragment



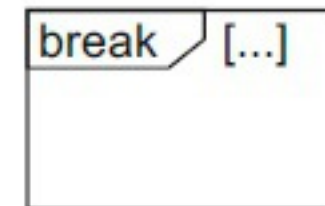
Optional interaction

loop fragment



Repeated interaction

break fragment

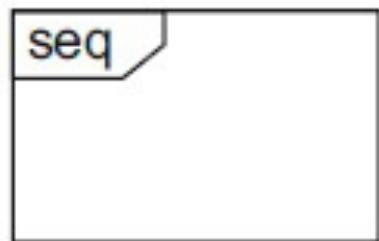


Exception handling

Syntax

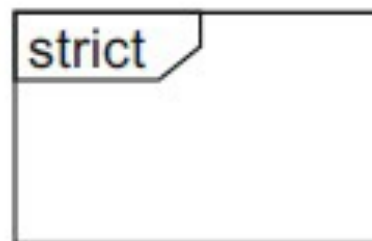
Concurrency and Order

seq fragment



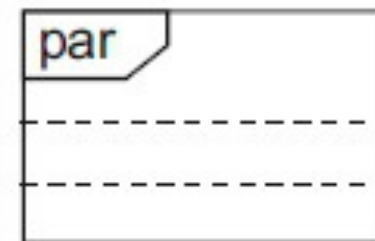
*Sequential interaction
with weak order*

strict fragment



*Sequential interaction
with a strict order*

par fragment



Concurrent interaction

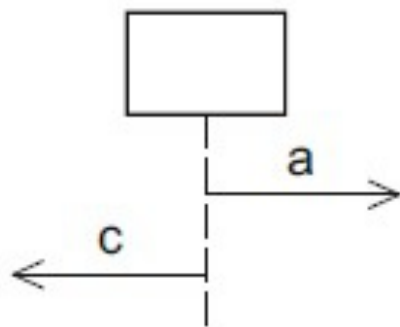
critical fragment



Atomic interaction

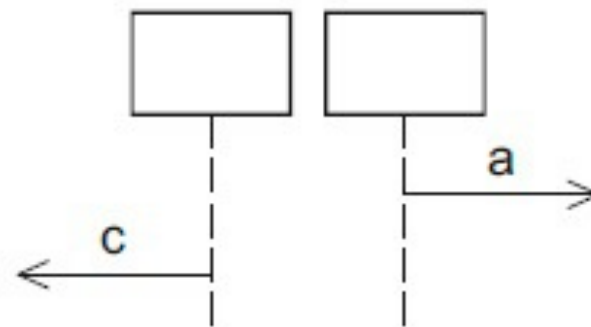
Semantics

- Message sequence



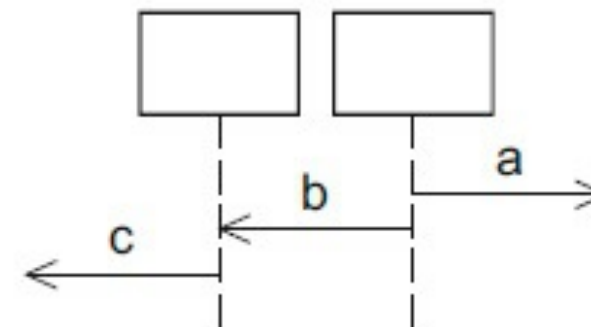
T01: $a \rightarrow c$

(a)



T01: $a \rightarrow c$
T02: $c \rightarrow a$

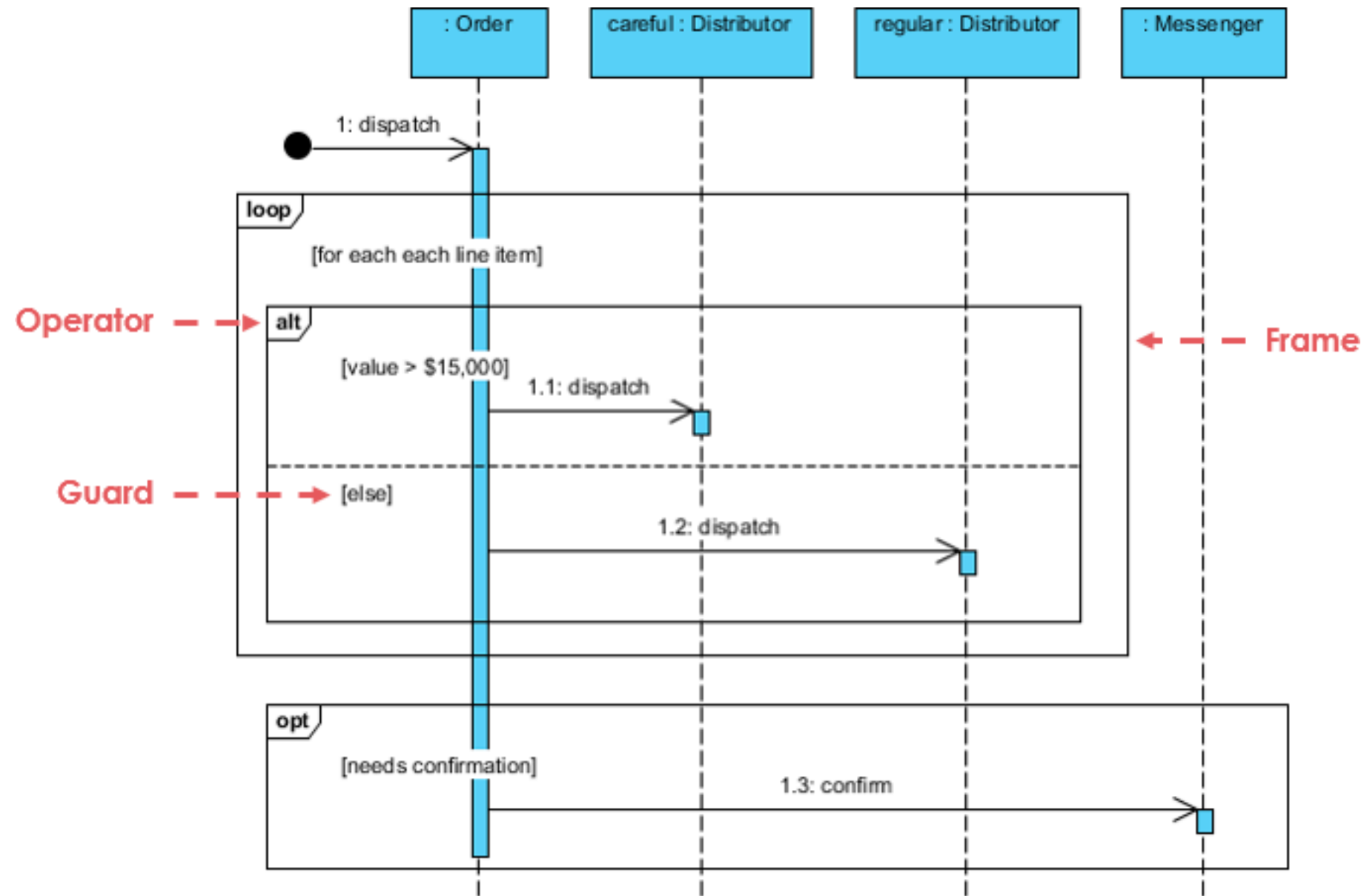
(b)



T01: $a \rightarrow b \rightarrow c$

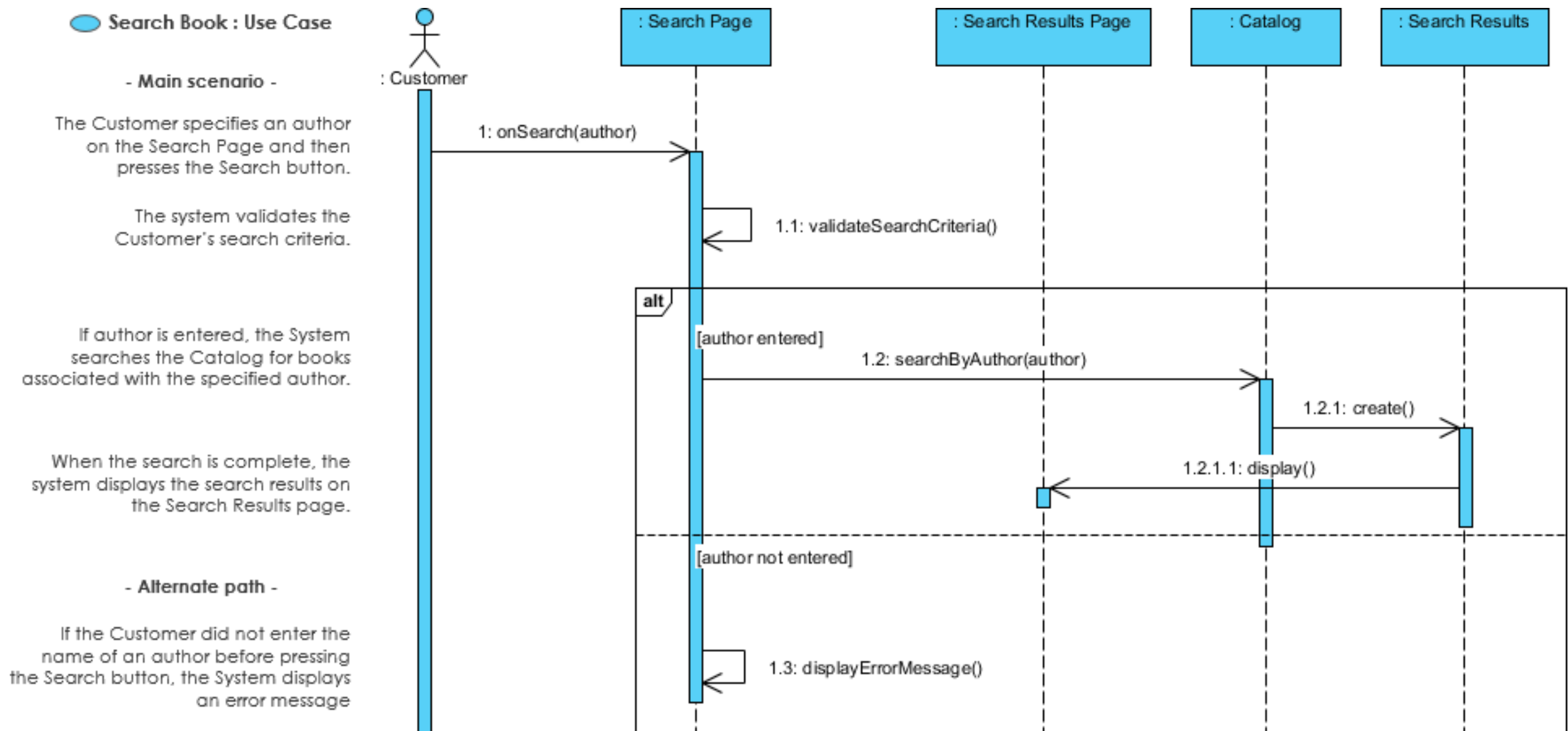
(c)

Combined Example



Example:

Search a Book in Library

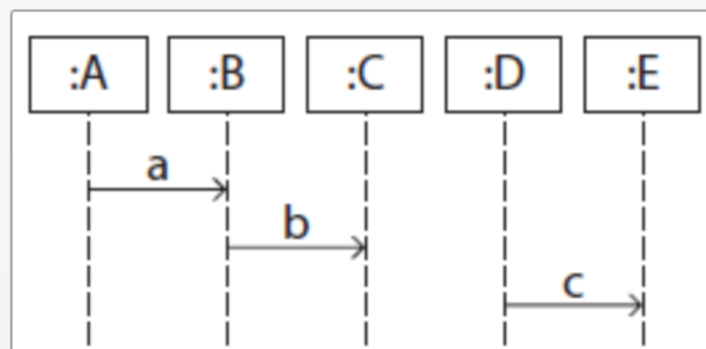


Semantics

- <https://arxiv.org/pdf/1003.1160.pdf>

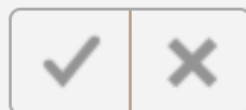
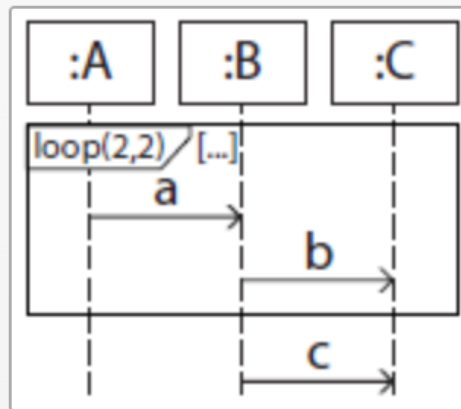
Quiz on Semantics

Question 34: You are given the following sequence diagram. Which traces are possible?

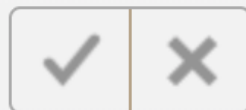


- | | | |
|-------------------------------------|--------------------------|---------------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $c \rightarrow b \rightarrow a$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $a \rightarrow c \rightarrow b$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $b \rightarrow a \rightarrow c$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $a \rightarrow b \rightarrow c$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $c \rightarrow a \rightarrow b$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $b \rightarrow c \rightarrow a$ |

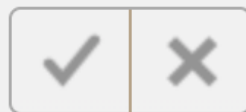
Question 1: You are given the following sequence diagram. Which traces are possible?



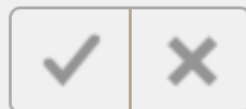
$a \rightarrow b \rightarrow a \rightarrow b \rightarrow c$



$a \rightarrow b \rightarrow c$

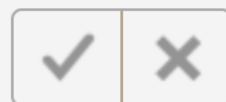
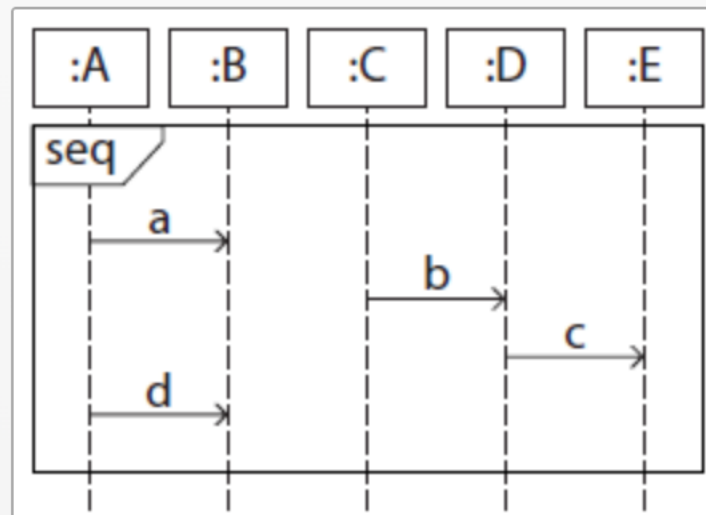


c

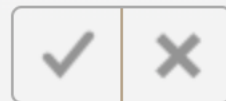


$a \rightarrow b \rightarrow a \rightarrow b$

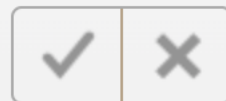
Question 1: You are given the following sequence diagram. Which traces are possible?



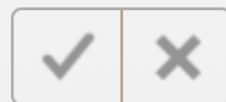
a → c → b → d



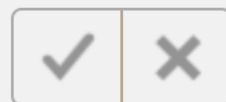
d → a → c → b



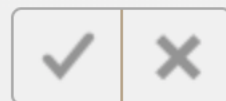
b → c → a → d



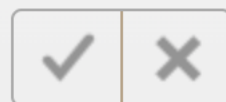
a → d → b → c



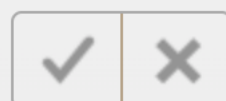
b → a → d → c



a → b → c → d

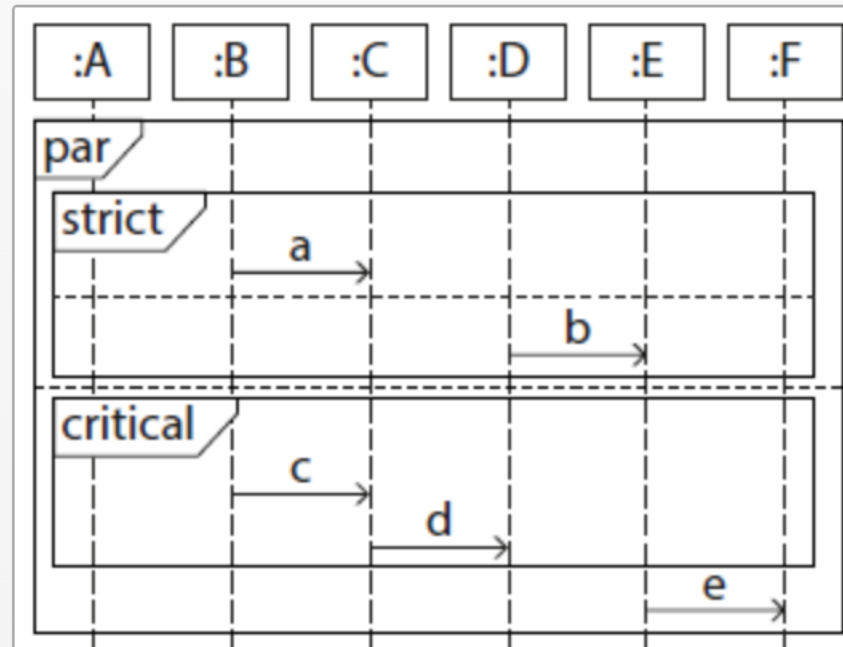


a → d → c → b



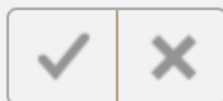
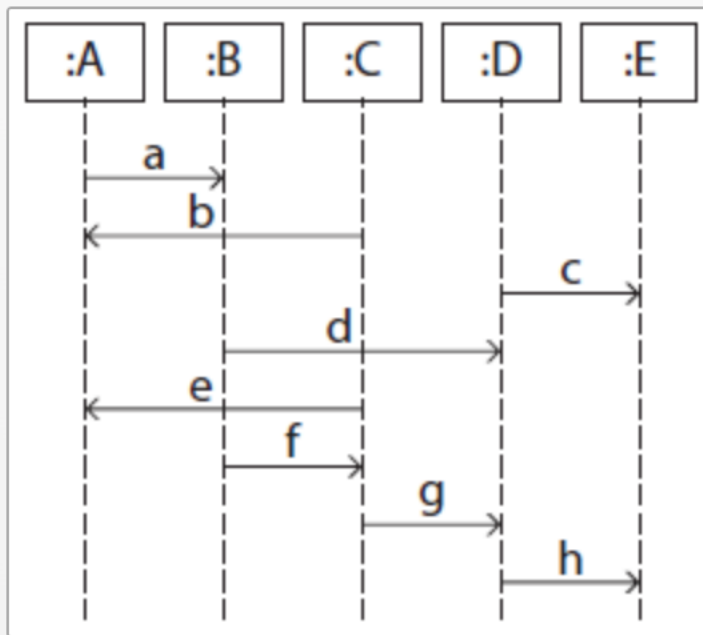
b → a → c → d

Question 6: You are given the following sequence diagram. Which traces are possible?

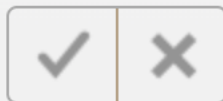


- | | | |
|-------------------------------------|--------------------------|---|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $c \rightarrow d \rightarrow a \rightarrow e \rightarrow b$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $c \rightarrow d \rightarrow a \rightarrow b \rightarrow e$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $a \rightarrow b \rightarrow c \rightarrow e \rightarrow d$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $c \rightarrow a \rightarrow e \rightarrow d \rightarrow b$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $a \rightarrow b \rightarrow e \rightarrow d \rightarrow c$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $e \rightarrow a \rightarrow b \rightarrow c \rightarrow d$ |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | $a \rightarrow c \rightarrow d \rightarrow b \rightarrow e$ |

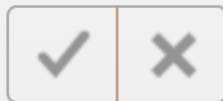
Question 16: You are given the following sequence diagram. How would you change the model supposed e should always be sent after d?



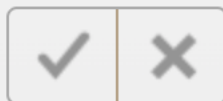
I do not have to change anything, as that is already the case.



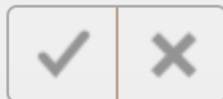
I would put d and e into a `critical` fragment.



I would put e and f into a `critical` fragment.



I would put b and c into a `strict` fragment.



I would put d and e into a `strict` fragment.

More Information

- Example diagrams from: <http://www.ibm.com/developerworks/rational/library/3101.html>
- <https://technical-leader.tistory.com/14>
- UML quiz - <http://elearning.uml.ac.at/>