Brunel
University
London

Software Development and Management
CS2002

Dr Giuseppe Destefanis
giuseppe.destefanis@brunel.ac.uk

Lecture 5

UML Sequence diagrams

At the end of this lecture, and after a period of independent study:

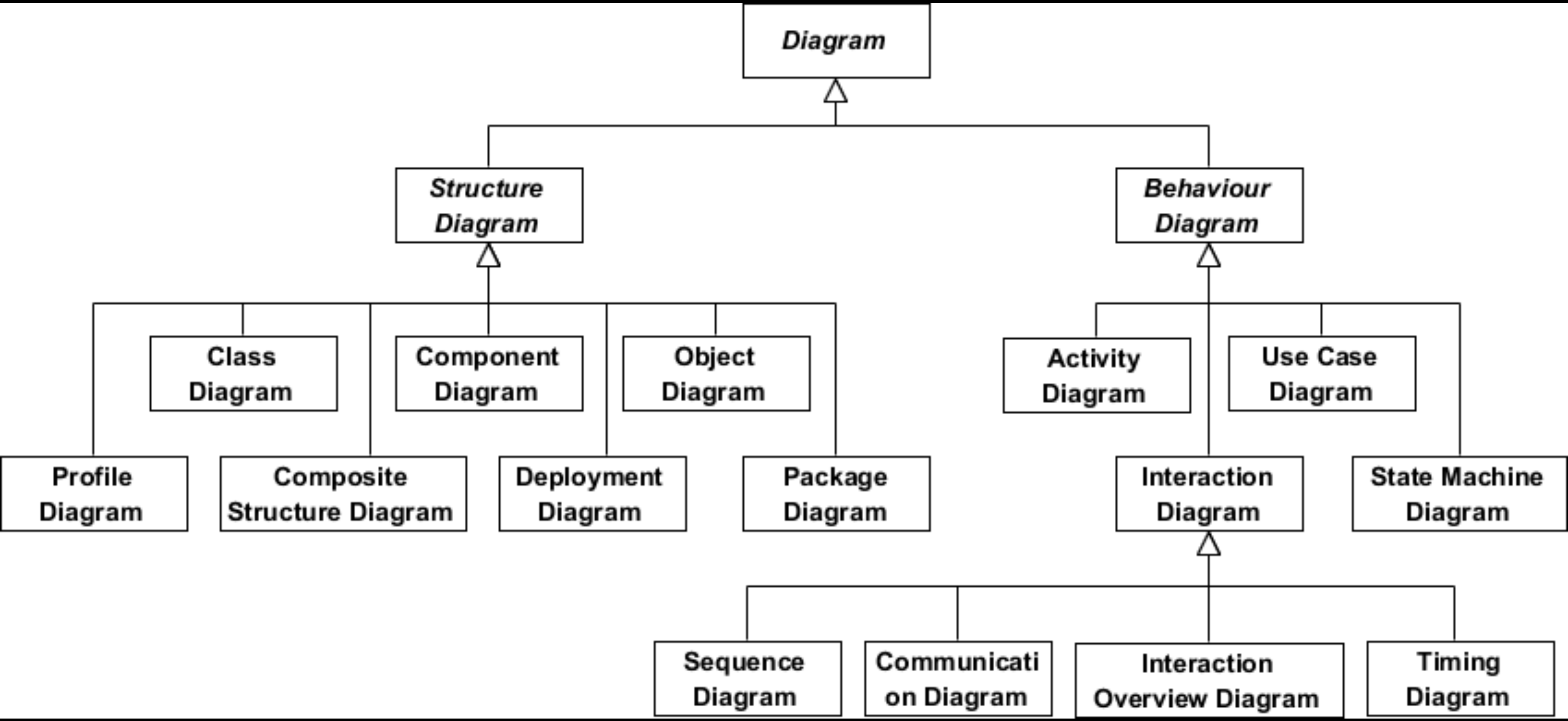
- You should be able to define what a Sequence Diagram is and to generate Sequence Diagrams given a Use Case diagram and a class diagram

Sequence Diagrams captures:

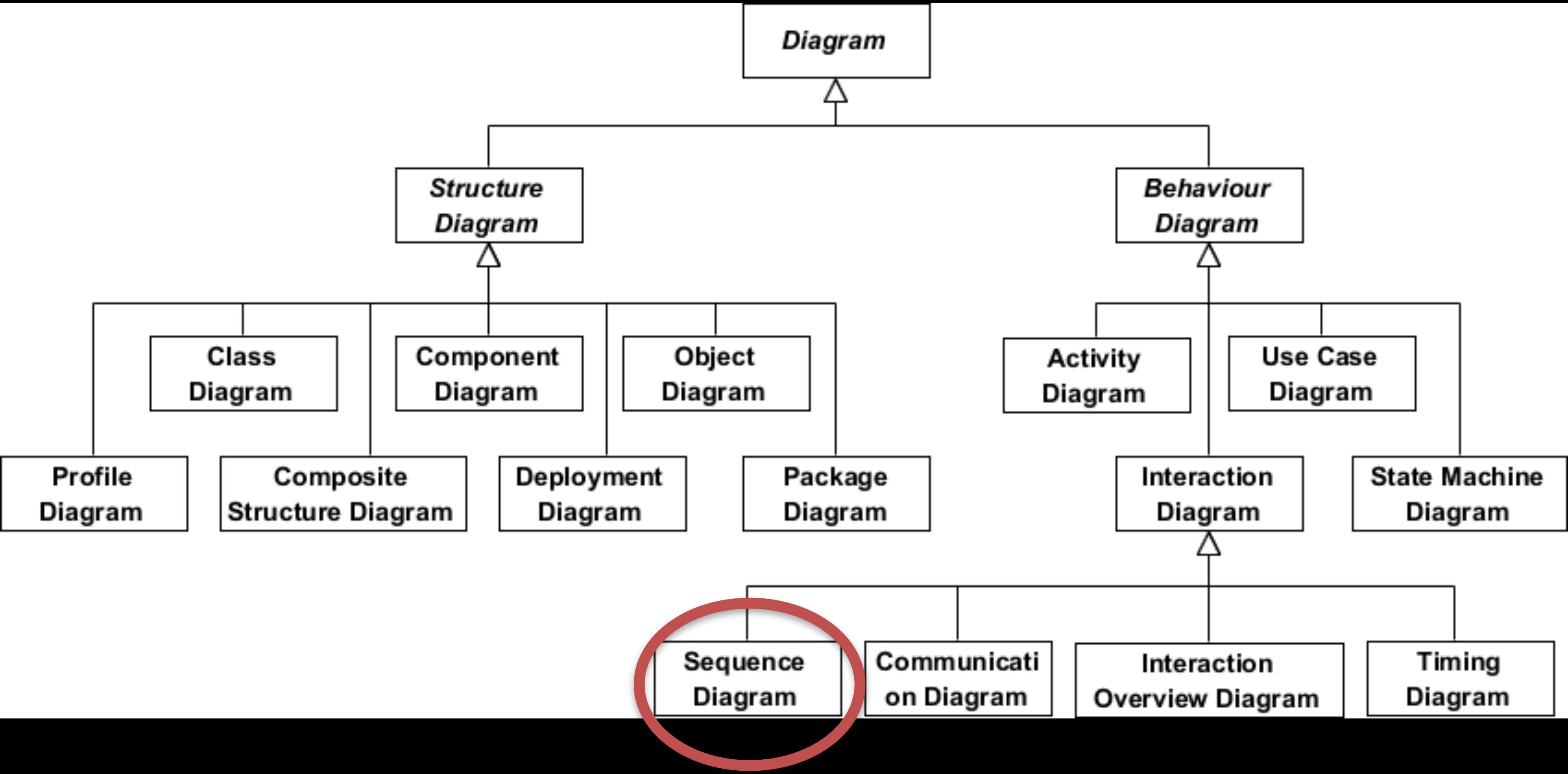
- the interaction that takes place in a collaboration that either realizes a **use case** or an **operation**

- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems

Different diagrams for different viewpoints



Different diagrams for different viewpoints

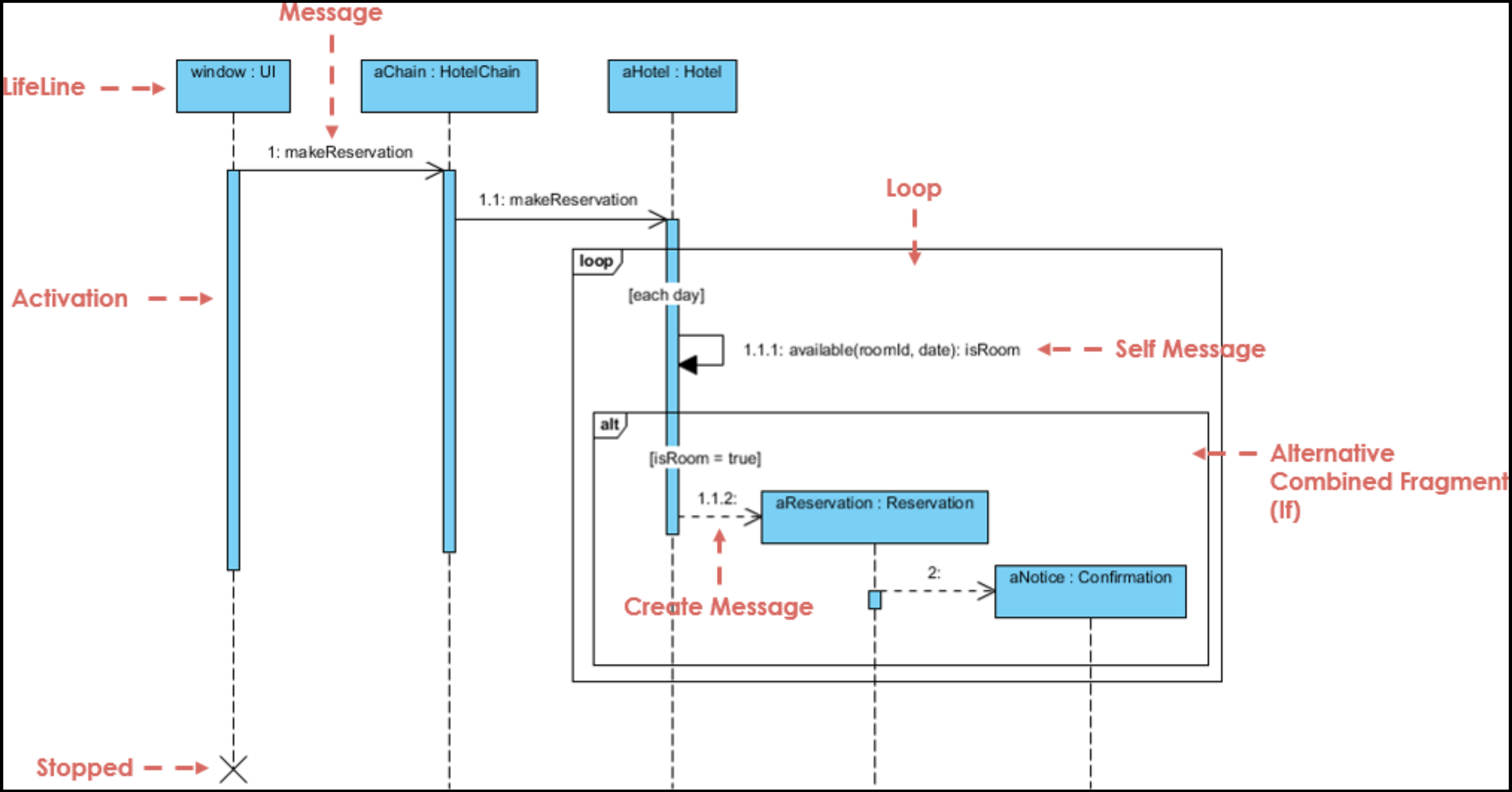


Purpose of a Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

Sequence diagrams

- Sequence Diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically)



Message

LifeLine

Loop

Activation

Self Message

Alternative Combined Fragment (If)

Create Message

Stopped

Object Dimension

- The horizontal axis shows the elements that are involved in the interaction

- Conventionally, the objects involved in the operation are listed from left to right according to when they take part in the message sequence. However, the elements on the horizontal axis may appear in any order

Time Dimension

- The vertical axis represents time proceedings (or progressing) down the page.

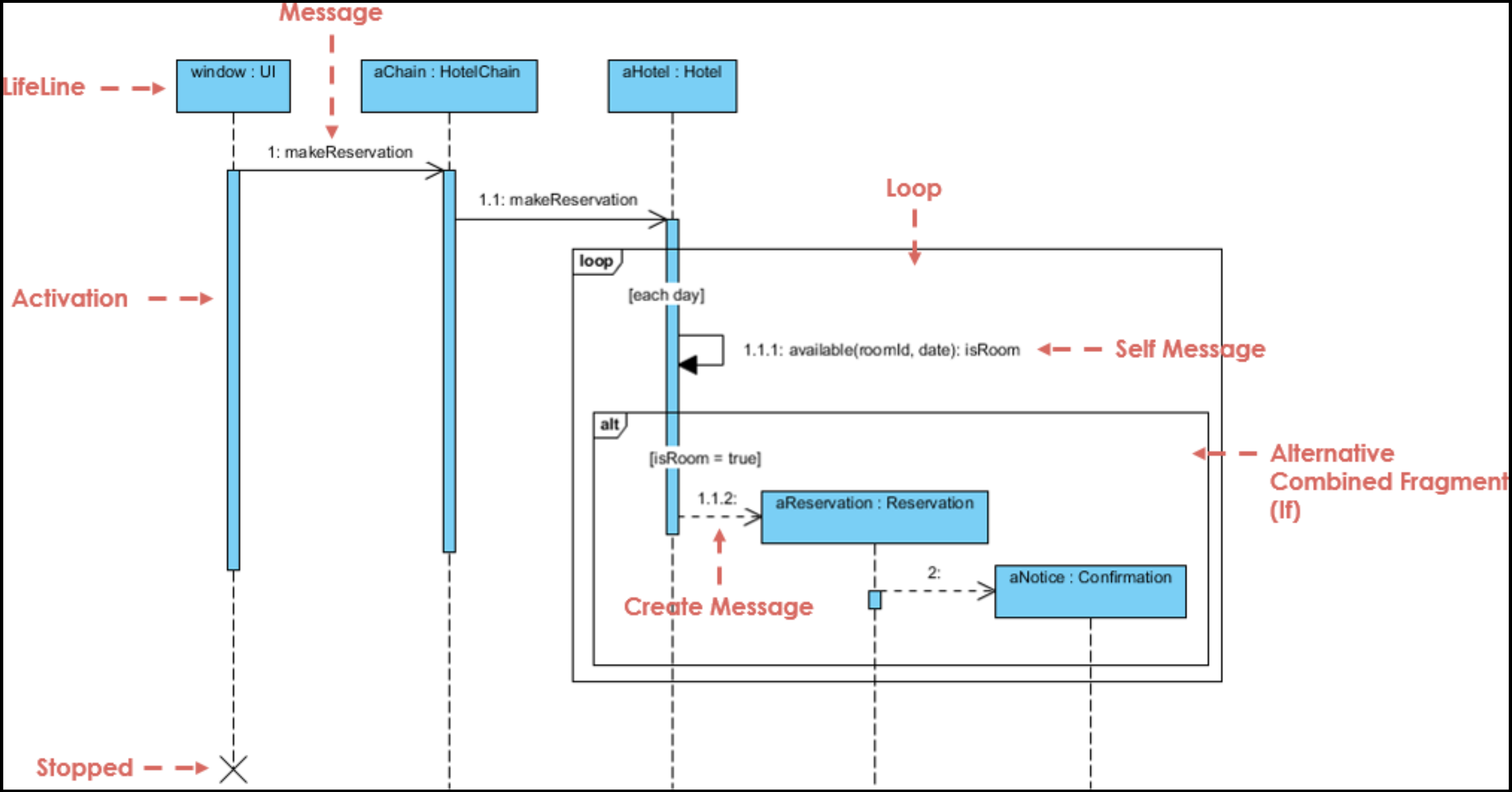
- Time in a sequence diagram is all about ordering, not duration. The vertical space in an interaction diagram is not relevant for the duration of the interaction.

Sequence Diagram is

an interaction diagram that details how operations are carried out (what messages are sent and when).

Sequence diagrams are organized according to time.

The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.



Message

LifeLine

Loop

Activation

Self Message

Alternative Combined Fragment (If)

Create Message

Stopped

Actor

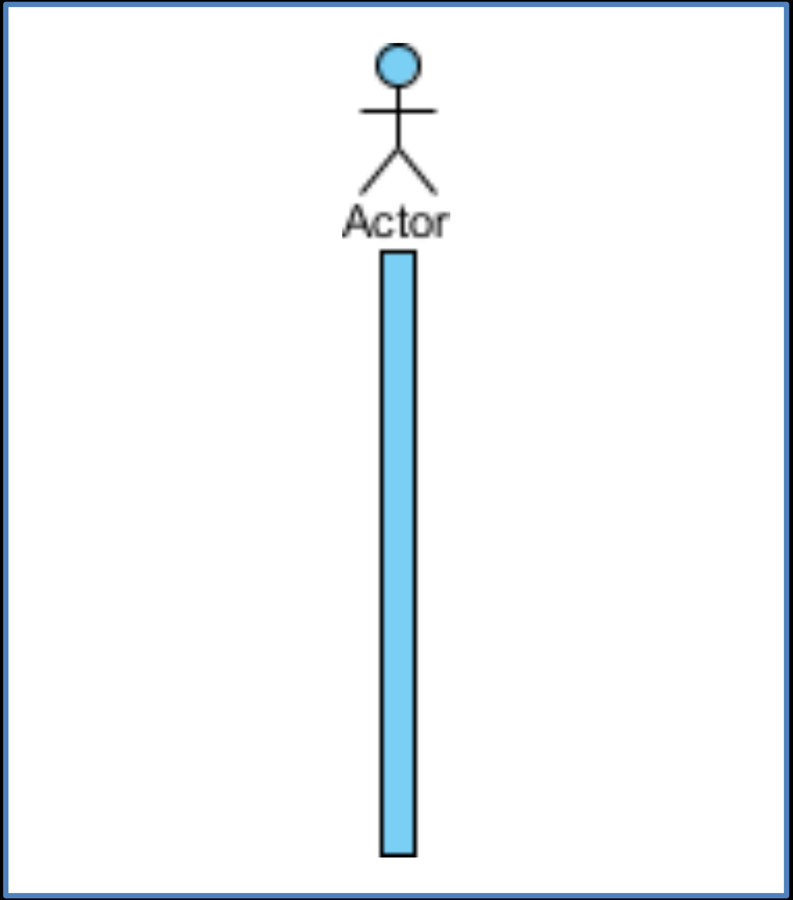
- a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data)

- 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.

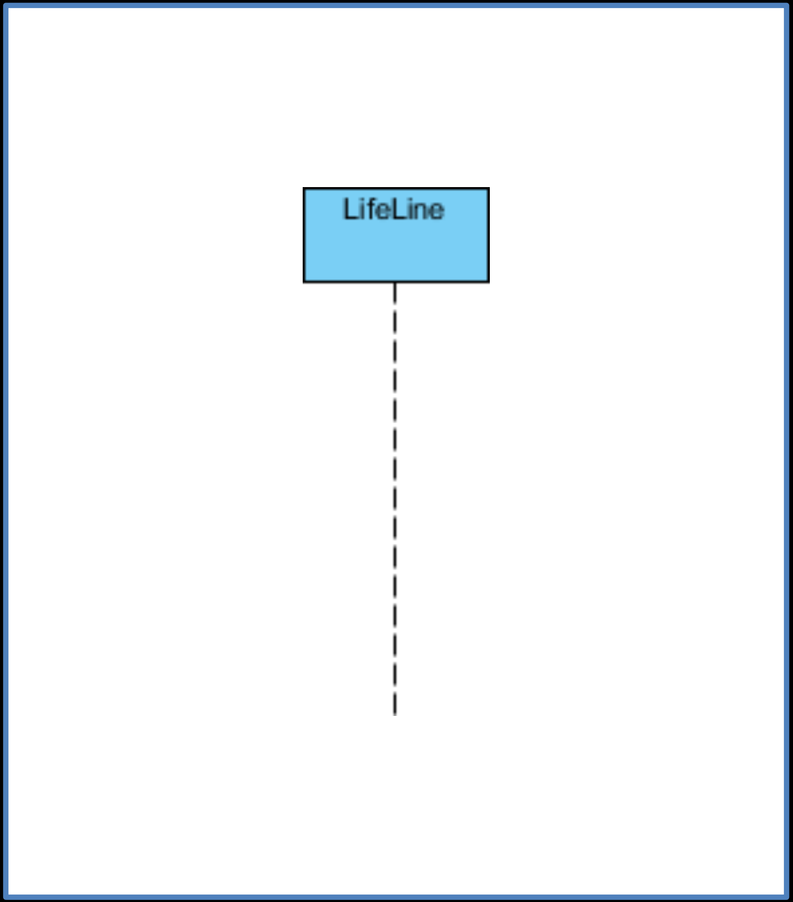
- An actor does not necessarily represent a specific physical entity but merely a particular role of some entity

- A person may play the role of several different actors and, conversely, a given actor may be played by multiple different person.



Lifeline

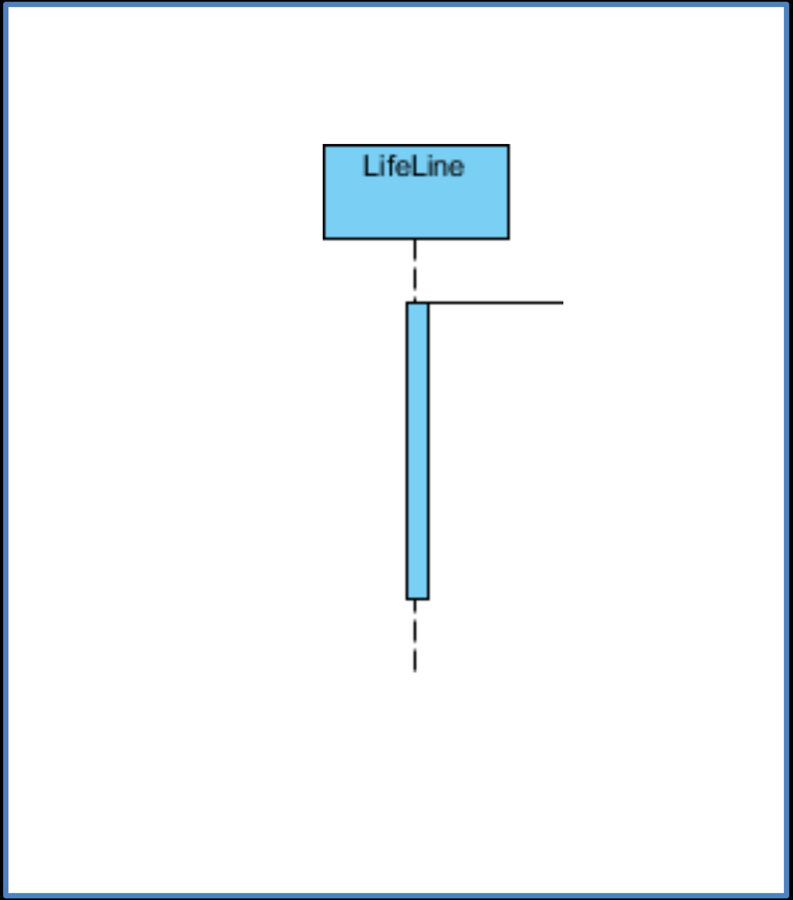
- A lifeline represents an individual participant in the Interaction



Activations

- A thin rectangle on a lifeline) represents the period during which an element is performing an operation.

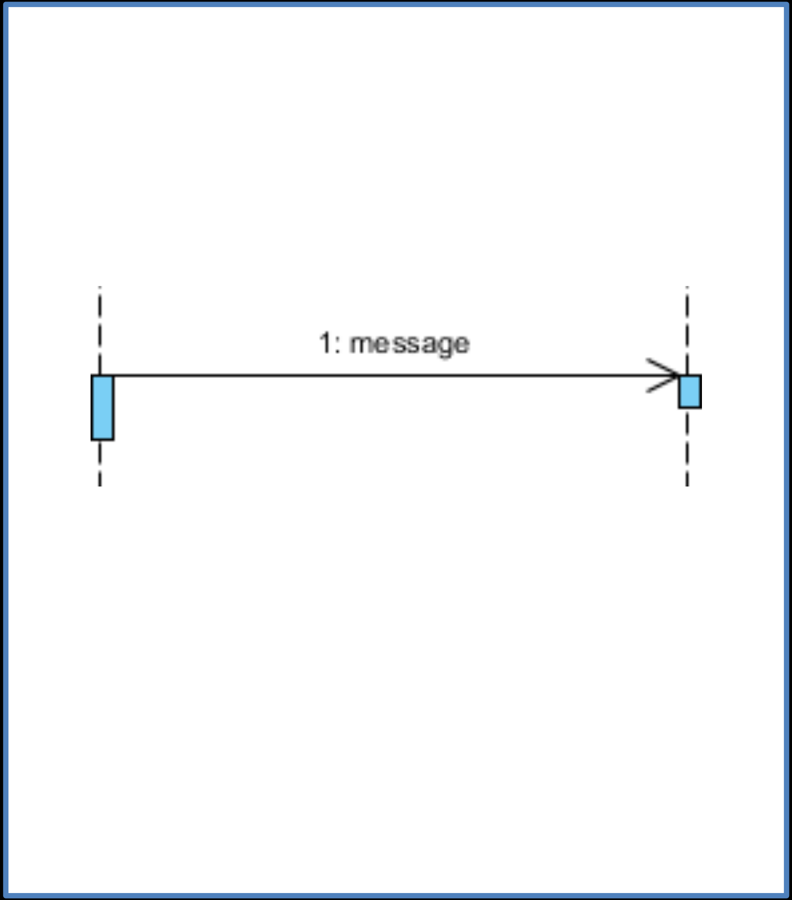
- The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



Call message

- A message defines a particular communication between Lifelines of an Interaction.

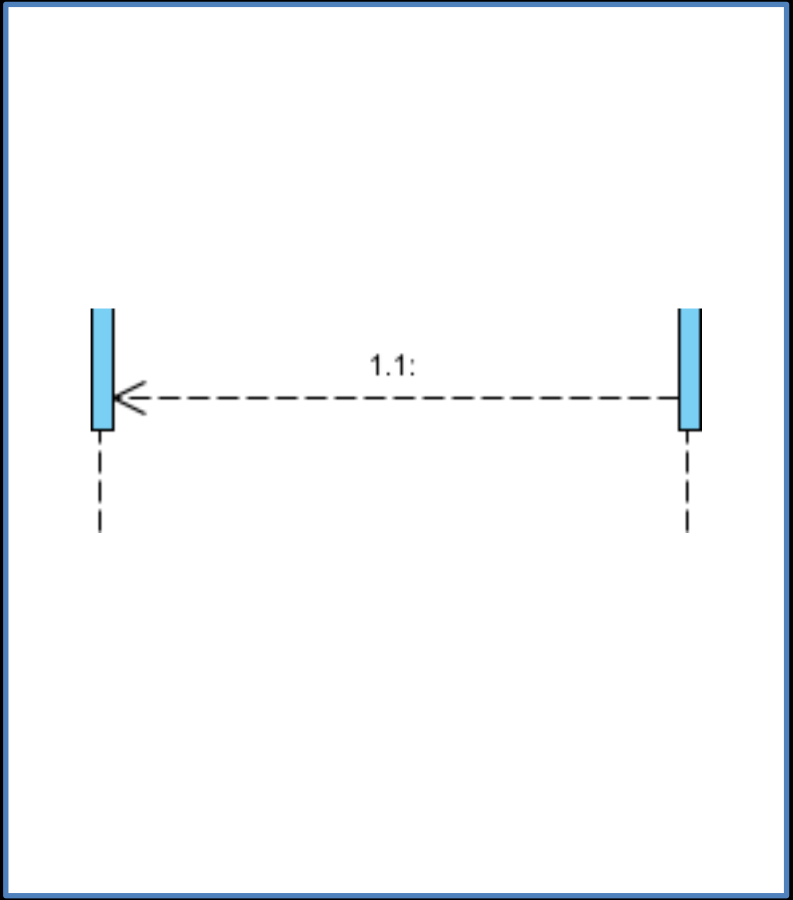
- Call message is a kind of message that represents an invocation of operation of target lifeline.



Return message

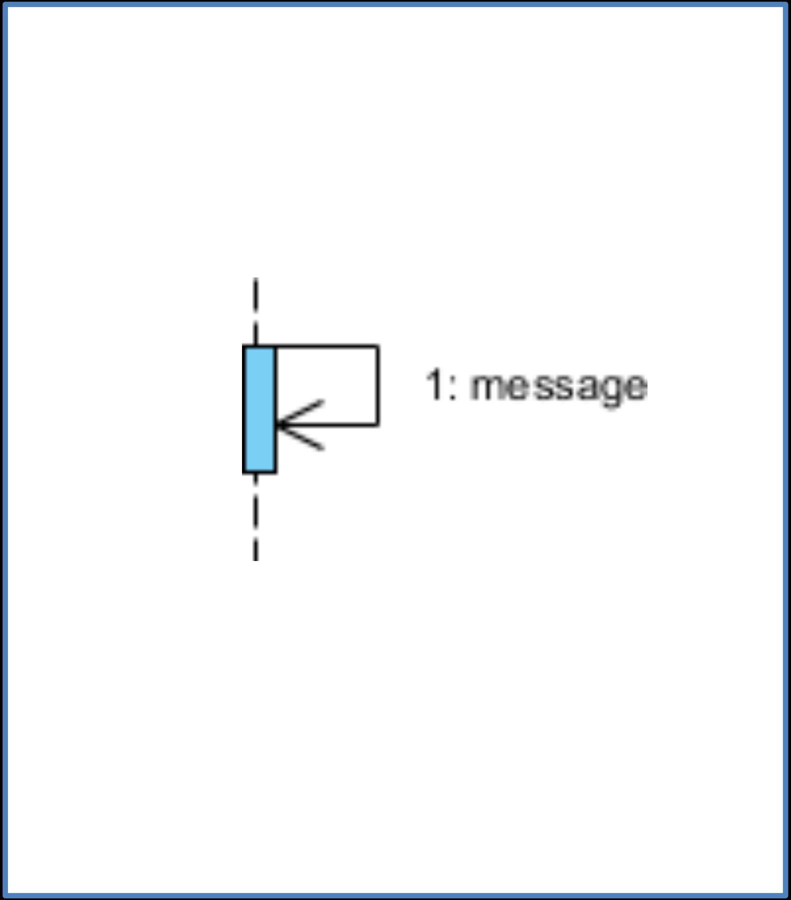
- A message defines a particular communication between Lifelines of an Interaction.

- Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.



Self message

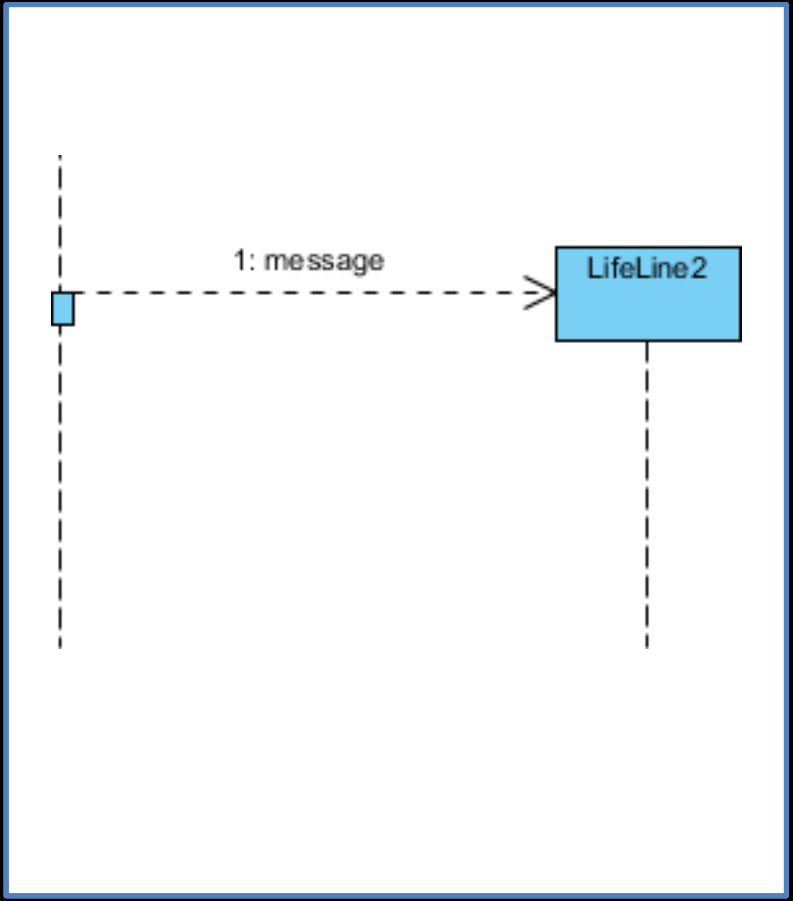
- A message defines a particular communication between Lifelines of an Interaction.
- Self message is a kind of message that represents the invocation of message of the same lifeline.



Create message

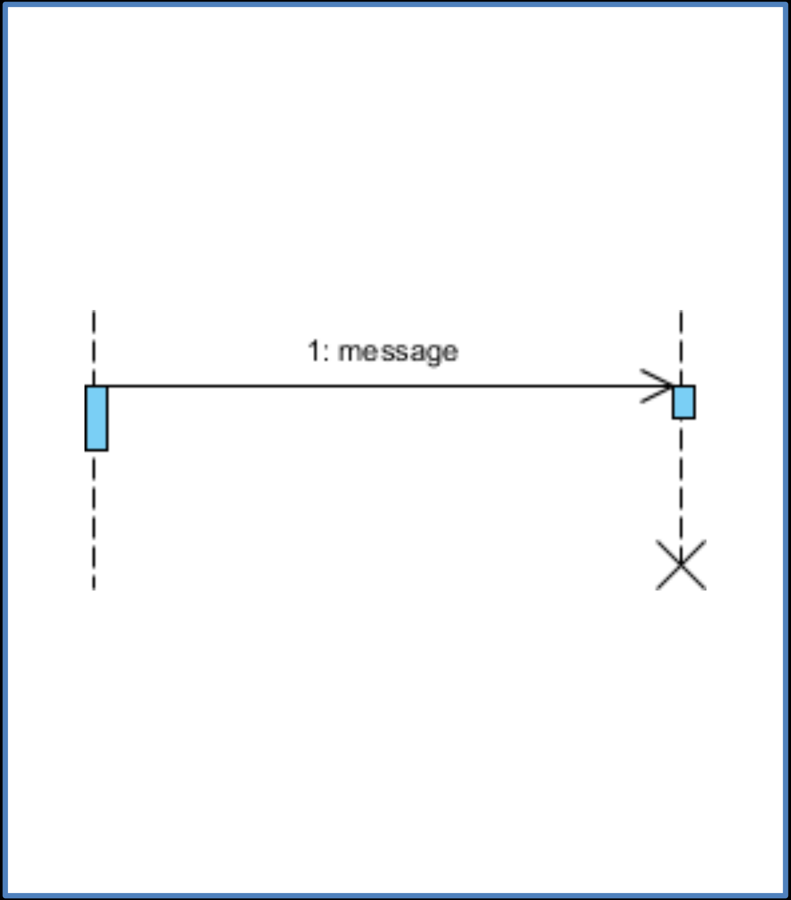
- A message defines a particular communication between Lifelines of an Interaction.

- Create message is a kind of message that represents the instantiation of (target) lifeline.



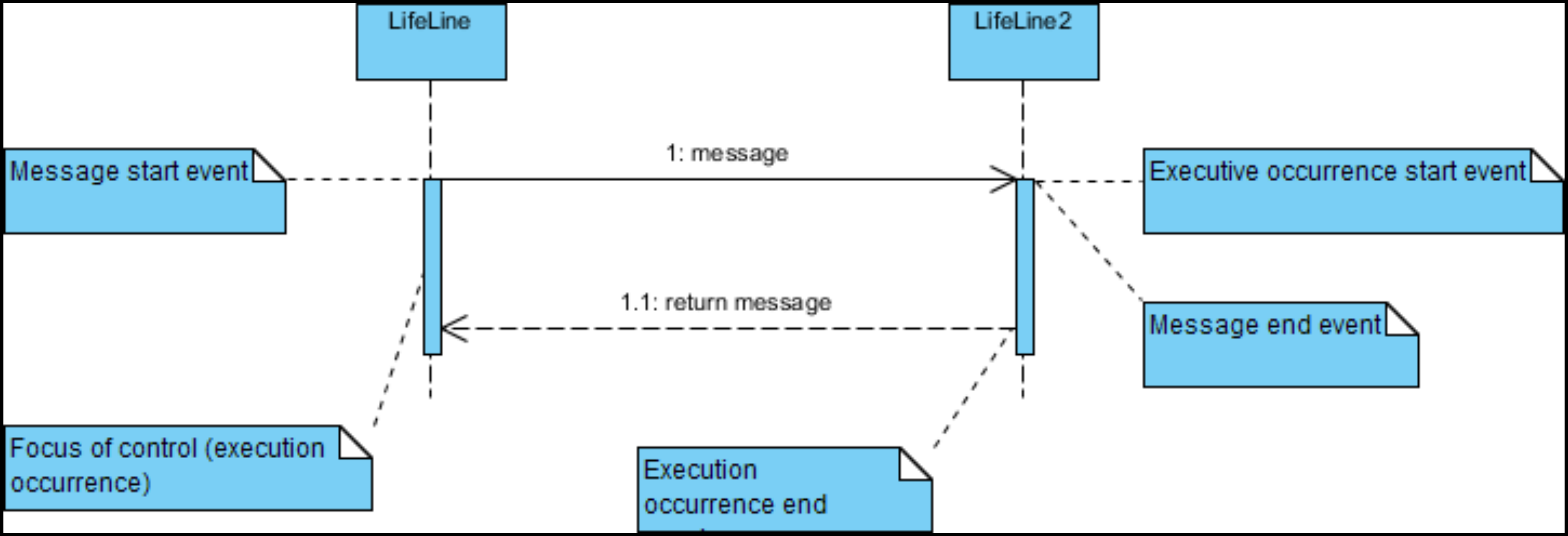
Destroy message

- 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.



Message and Focus of Control

- An Event is any point in an interaction where something occurs.
- Focus of control: also called execution occurrence, an execution occurrence
- It shows as tall, thin rectangle on a lifeline)
- It represents the period during which an element is performing an operation. The top and the bottom of the rectangle are aligned with the initiation and the completion time respectively.



LifeLine

LifeLine2

1: message

1.1: return message

Message start event

Executive occurrence start event

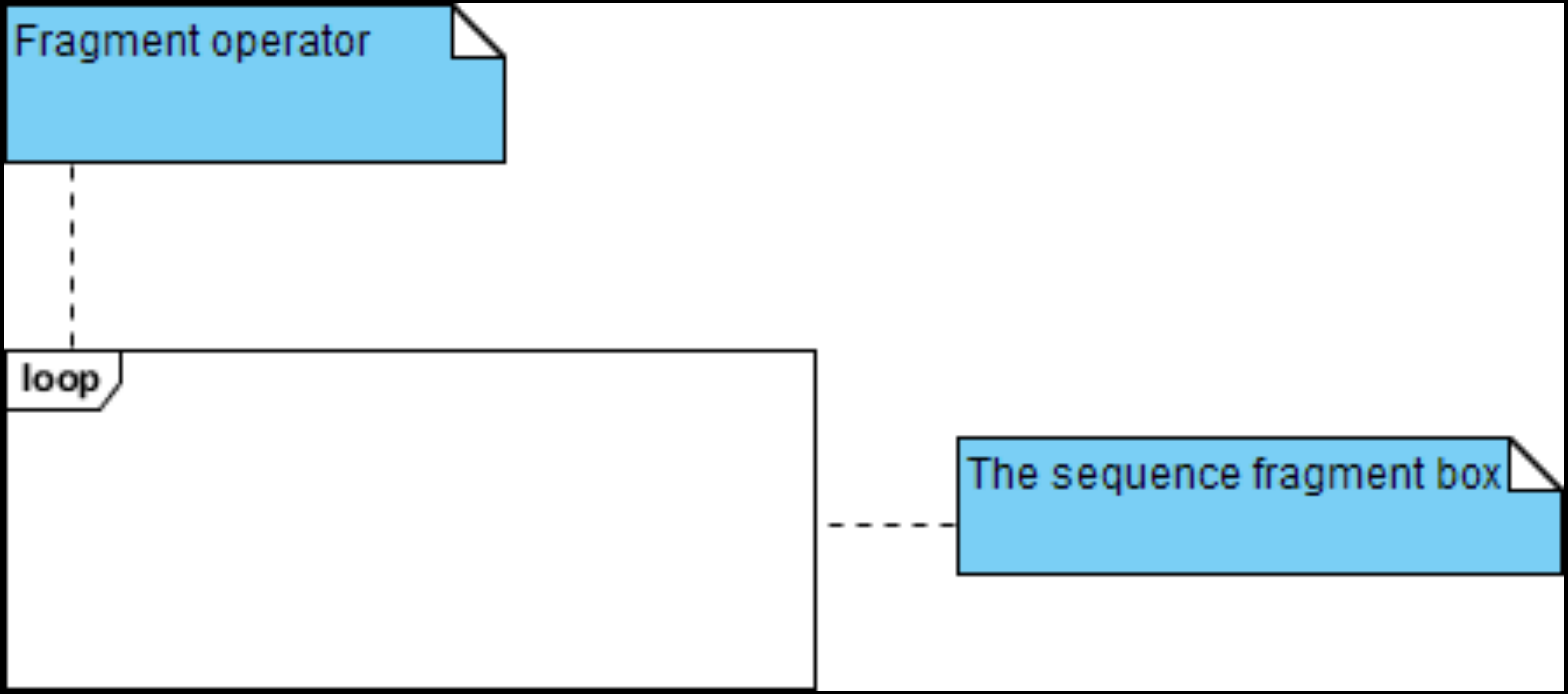
Message end event

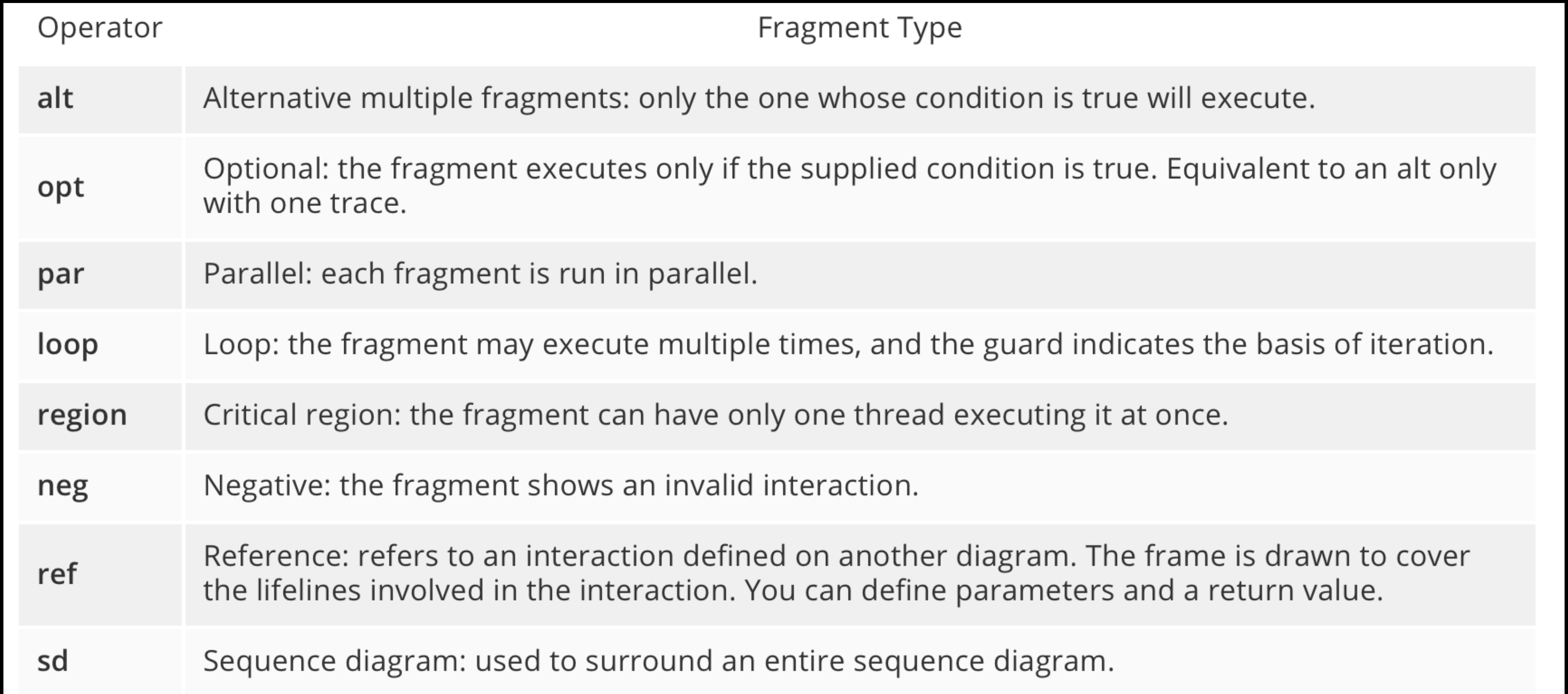
Focus of control (execution occurrence)

Execution occurrence end

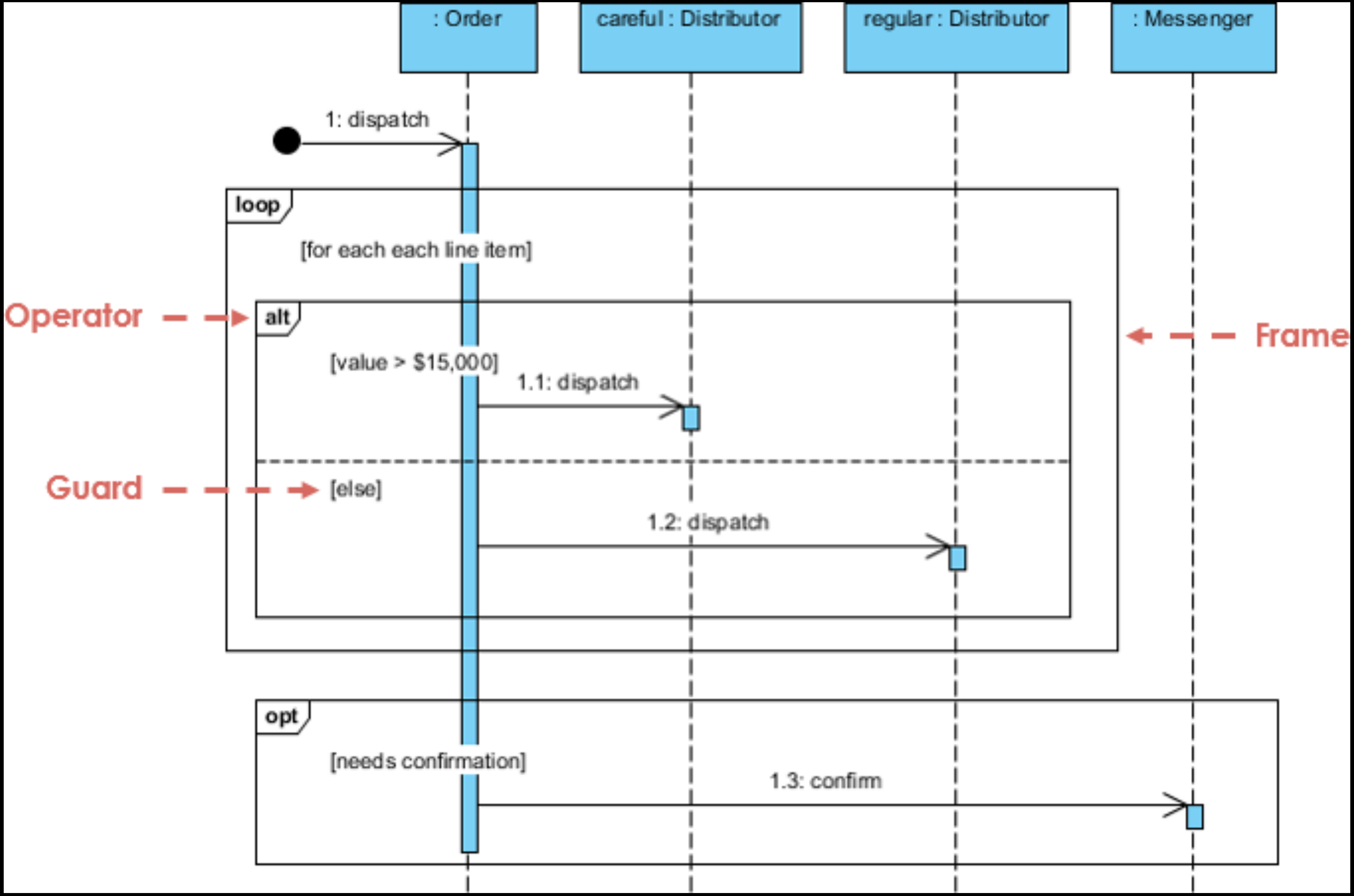
Sequence fragments

- Sequence fragments make it easier to create and maintain accurate sequence diagrams
- A sequence fragment is represented as a box, called a combined fragment, which encloses a portion of the interactions within a sequence diagram
- The fragment operator (in the top left cornet) indicates the type of fragment
- Fragment types: ref, assert, loop, break, alt, opt, neg


Fragment operator
loop
The sequence fragment box



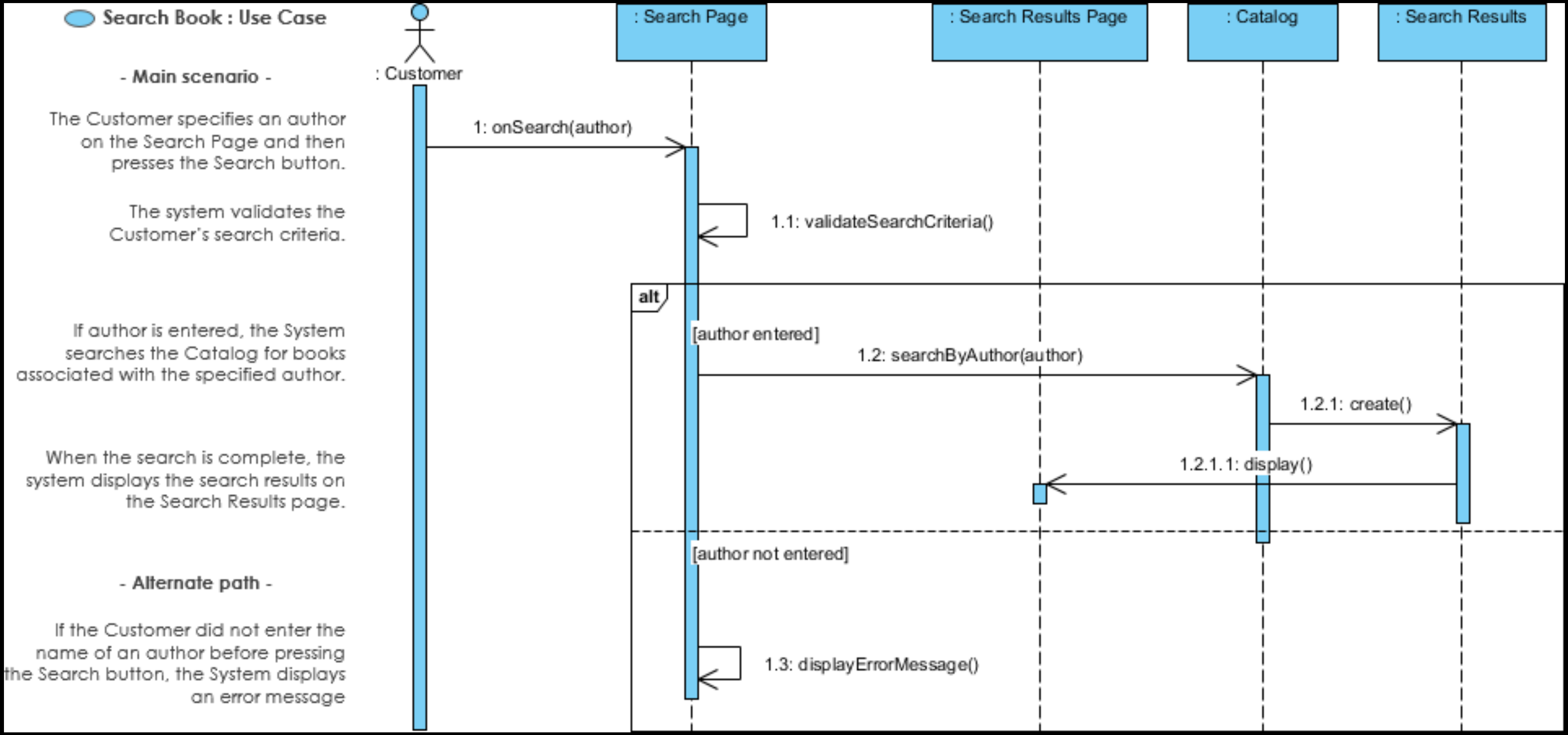
Operator	Fragment Type
alt	Alternative multiple fragments: only the one whose condition is true will execute.
opt	Optional: the fragment executes only if the supplied condition is true. Equivalent to an alt only with one trace.
par	Parallel: each fragment is run in parallel.
loop	Loop: the fragment may execute multiple times, and the guard indicates the basis of iteration.
region	Critical region: the fragment can have only one thread executing it at once.
neg	Negative: the fragment shows an invalid interaction.
ref	Reference: refers to an interaction defined on another diagram. The frame is drawn to cover the lifelines involved in the interaction. You can define parameters and a return value.
sd	Sequence diagram: used to surround an entire sequence diagram.



Operator

Guard

Frame



Search Book : Use Case

- Main scenario -

The Customer specifies an author on the Search Page and then presses the Search button.

The system validates the Customer's search criteria.

If author is entered, the System searches the Catalog for books associated with the specified author.

When the search is complete, the system displays the search results on the Search Results page.

- Alternate path -

If the Customer did not enter the name of an author before pressing the Search button, the System displays an error message

Recap

- Sequence diagrams can be somewhat close to the code level
- A good sequence diagram is still a bit above the level of the real code
- Sequence diagrams are language neutral
- Non-coders can do sequence diagrams
- Easier to do sequence diagrams as a team
- Can be used for testing

Reading

- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/
- https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp
- Chapter 5, Sommerville