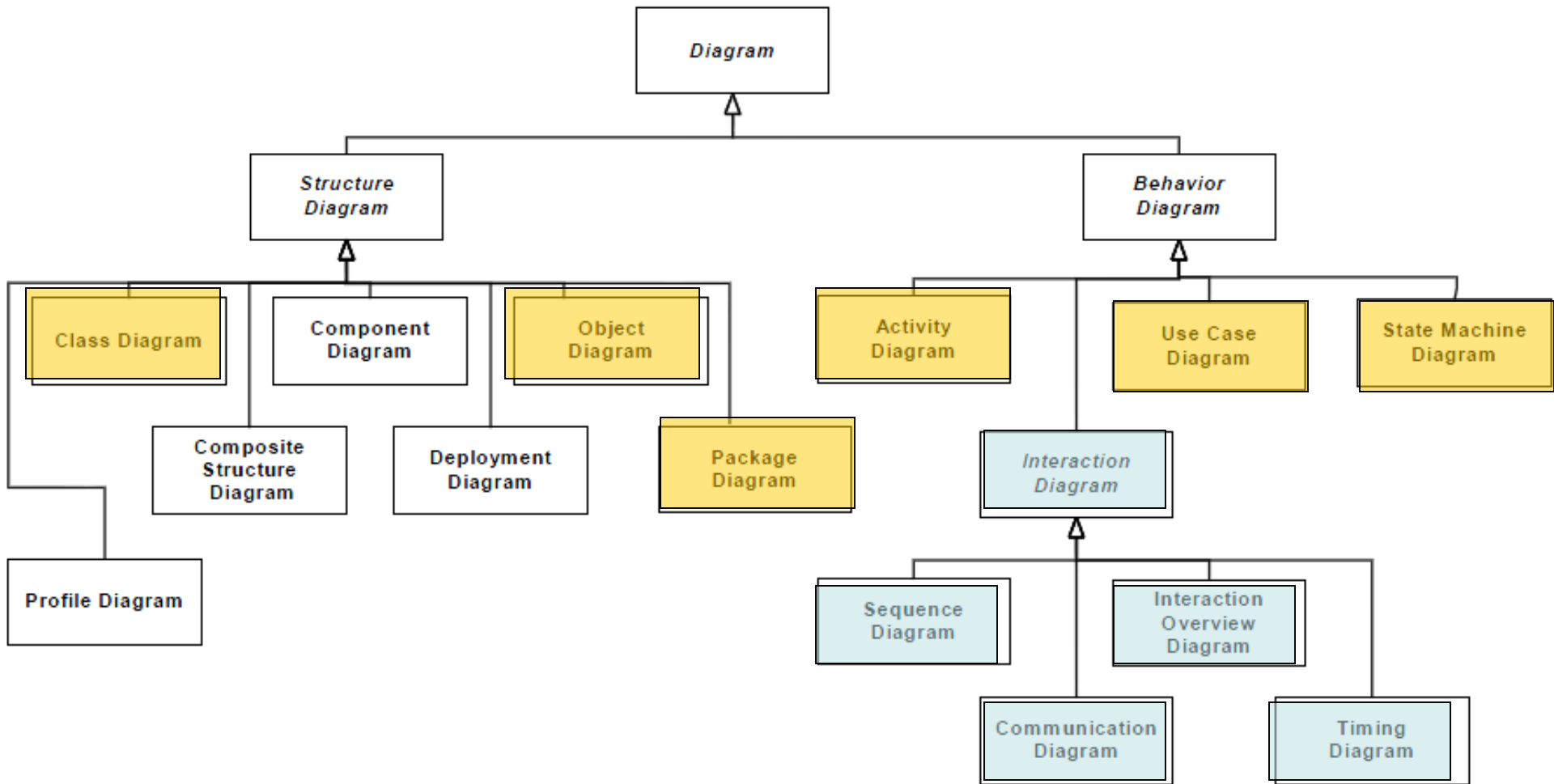


Object-Oriented Technology and UML Interaction Diagram

Interaction Diagram



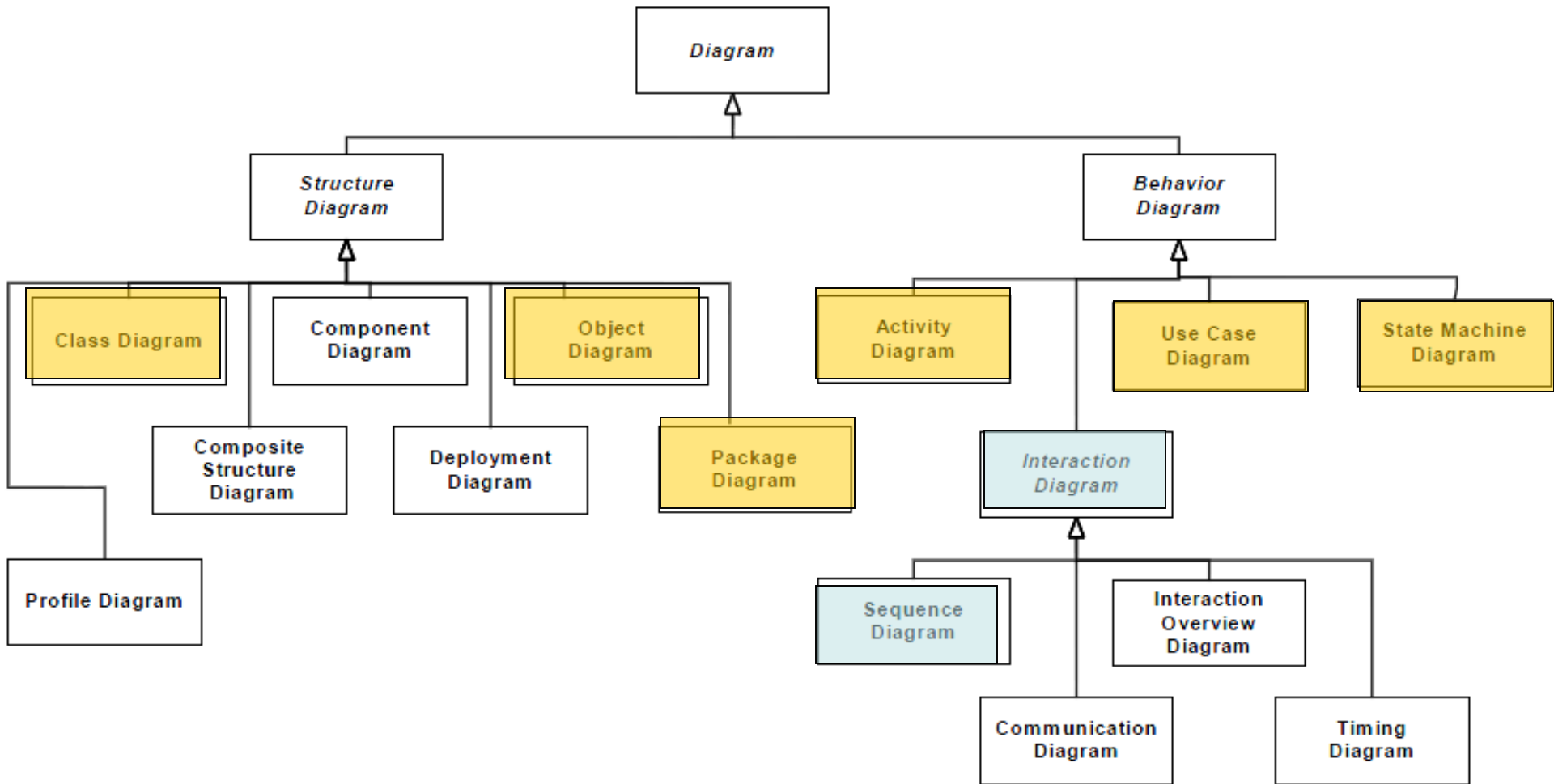
Kinds of Interaction Diagrams

- Sequence Diagram
- Communication Diagram
- Timing Diagram
- Interaction Overview Diagram

Topics

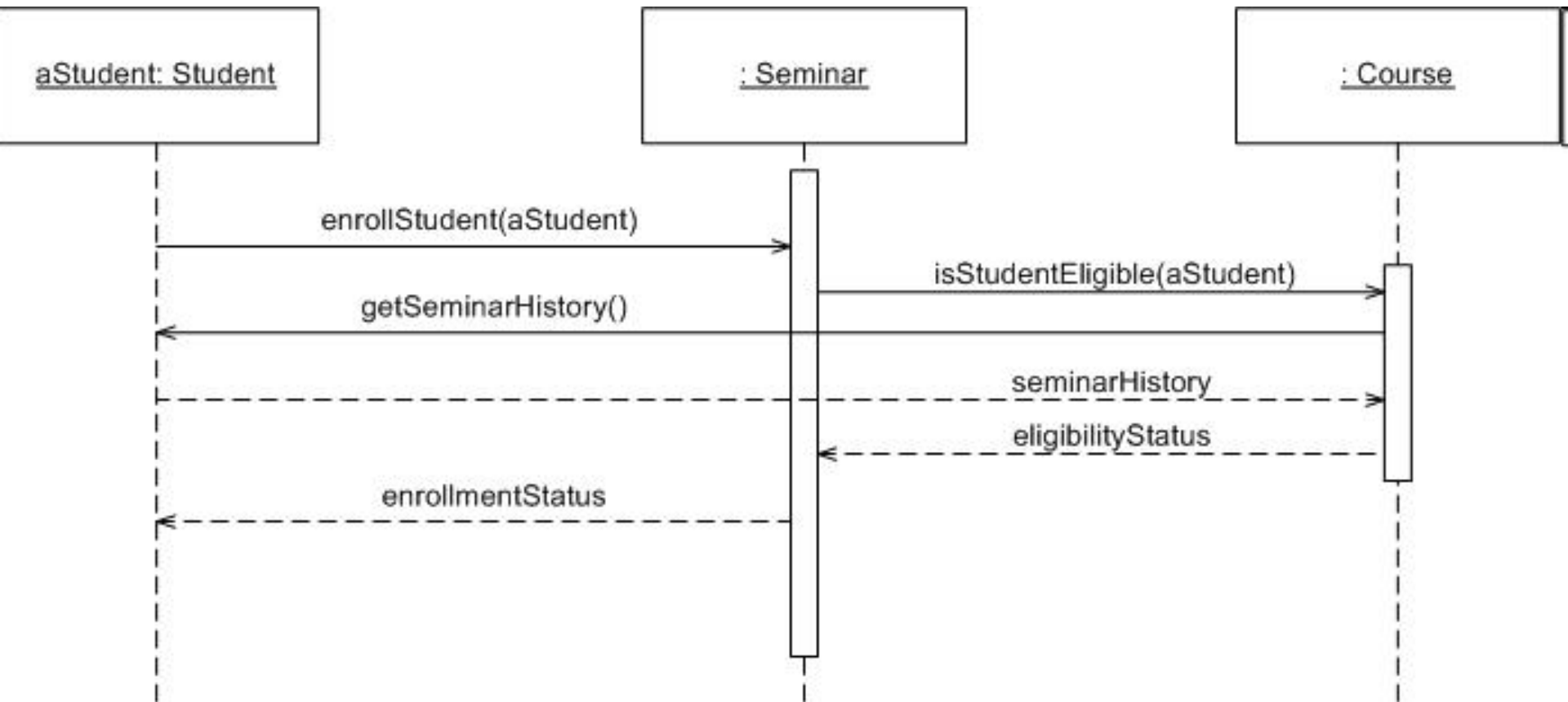
- Basic concepts
- Constituent elements
- Representation
- Reading method
- Modeling

Sequence Diagram

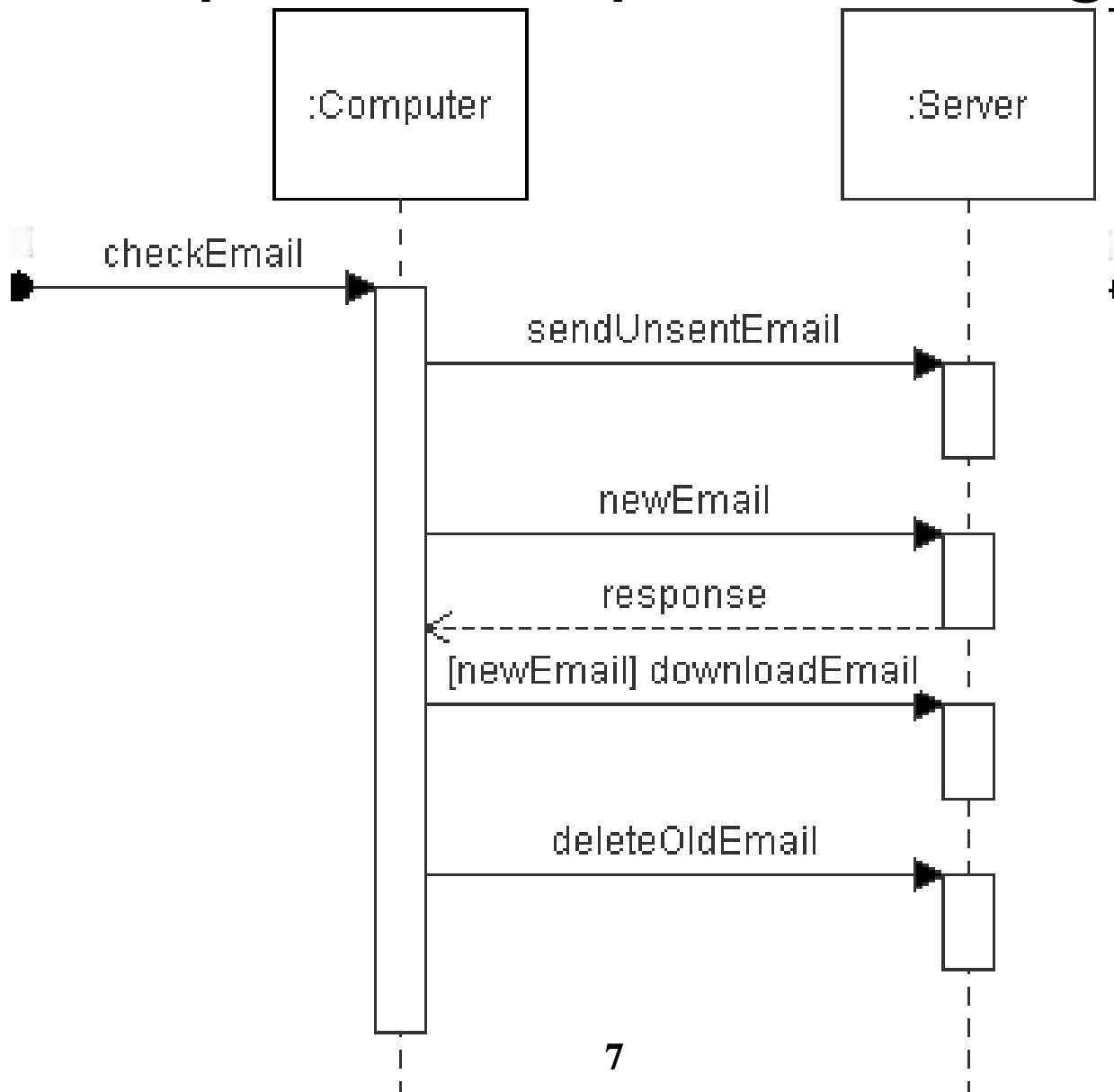


Example of Sequence Diagram

● Enrolling in a seminar



Example of Sequence Diagram

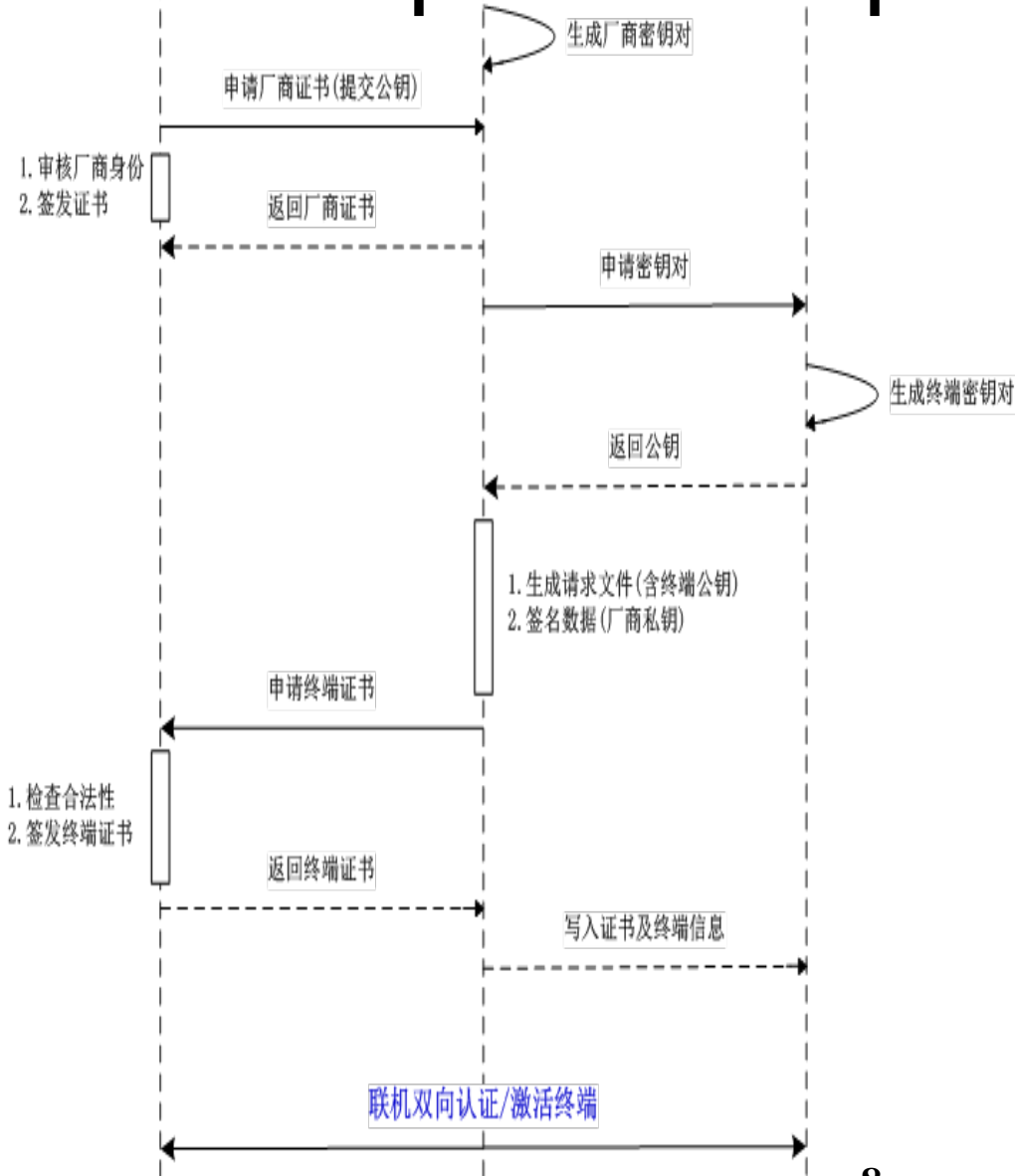


中国银联

终端厂商

支付终端

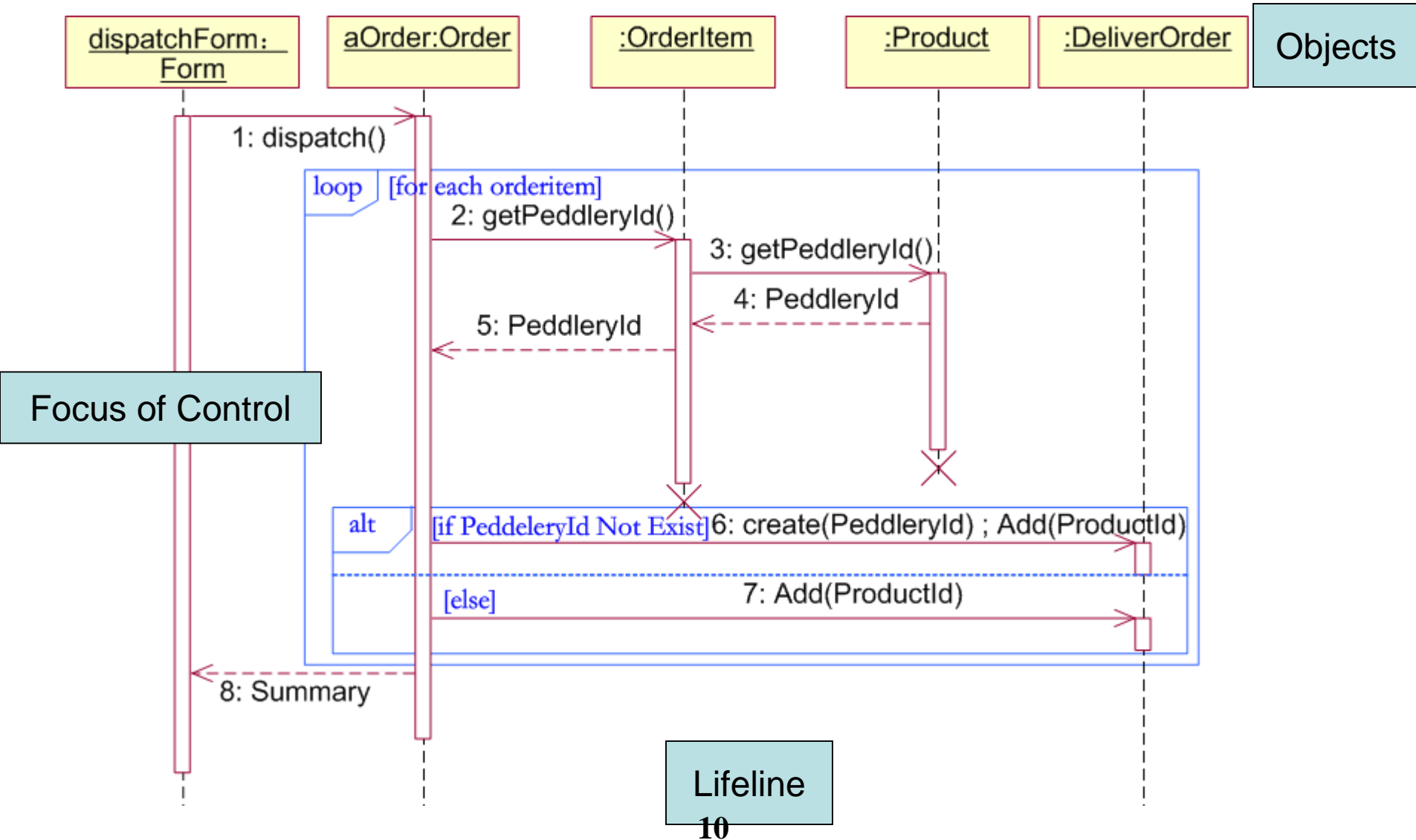
Example of Sequence Diagram



Sequence Diagram

- A sequence diagram is an interaction diagram that **emphasizes the time ordering of messages**
- A sequence diagram shows a set of roles and the messages sent and received by the **instances** playing the roles
- You use sequence diagrams to illustrate the **dynamic** view of a system

Sequence Diagram



Sequence Diagram

- Graphically, a sequence diagram is a table that shows
 - **objects** arranged along the **X axis**
 - and **messages**, ordered in increasing time, along the **Y axis**

Constituent Elements of Sequence Diagram

- Object
- Lifeline
- Focus of control
- Message

Object

- Place the objects or roles that participate in the interaction at the top of your diagram, across the horizontal axis

objectName :
ClassName

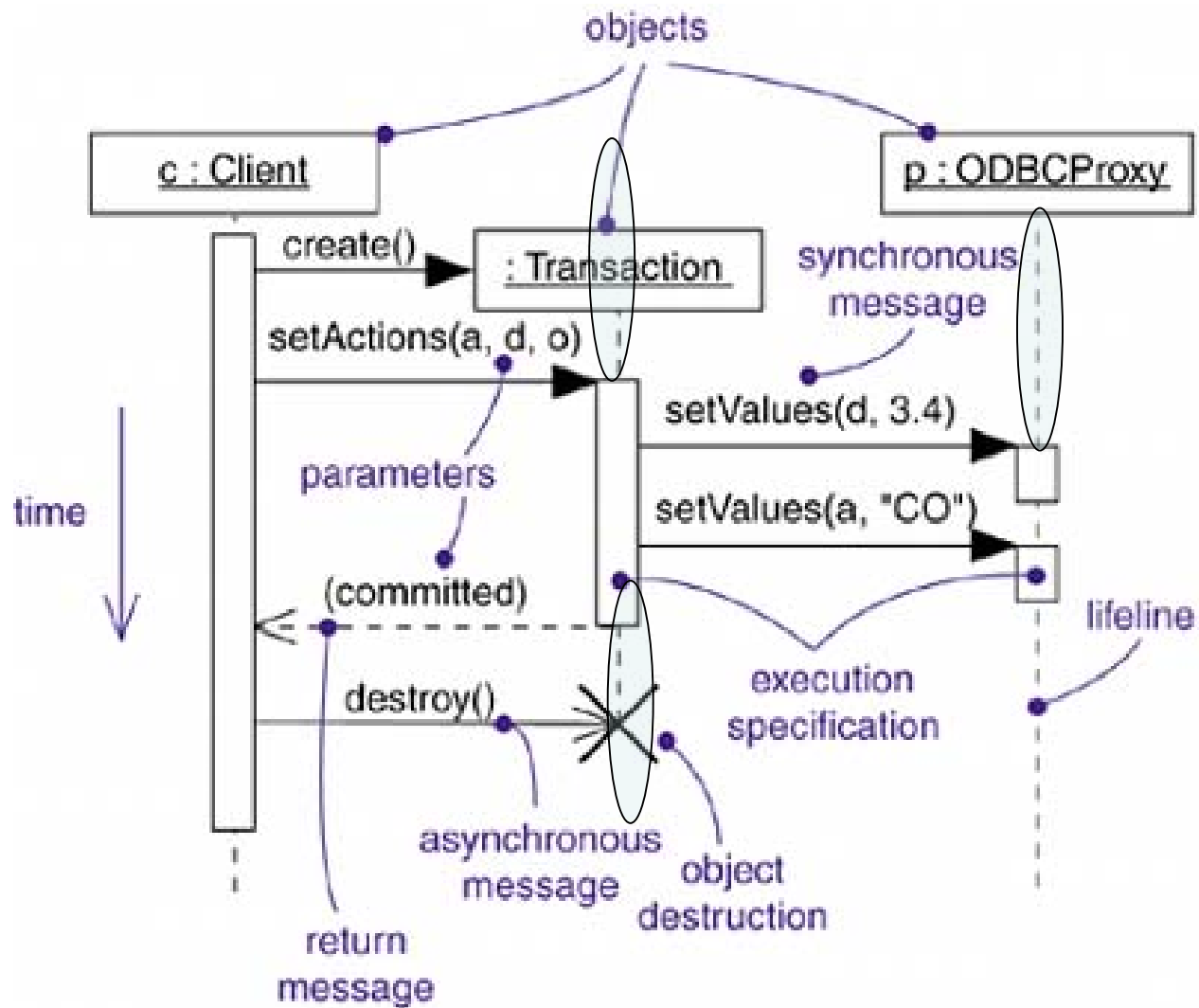
: ClassName

objectName

Lifeline

- An object lifeline is **the vertical dashed line that represents the existence of an object over a period of time**
- Most objects that appear in an interaction diagram will be in existence for the duration of the interaction, so these objects are all aligned at the top of the diagram, with their lifelines drawn from the top of the diagram to the bottom
- Objects may be created during the interaction. Their lifelines start with the receipt of the message create (drawn to box at the head of the lifeline)

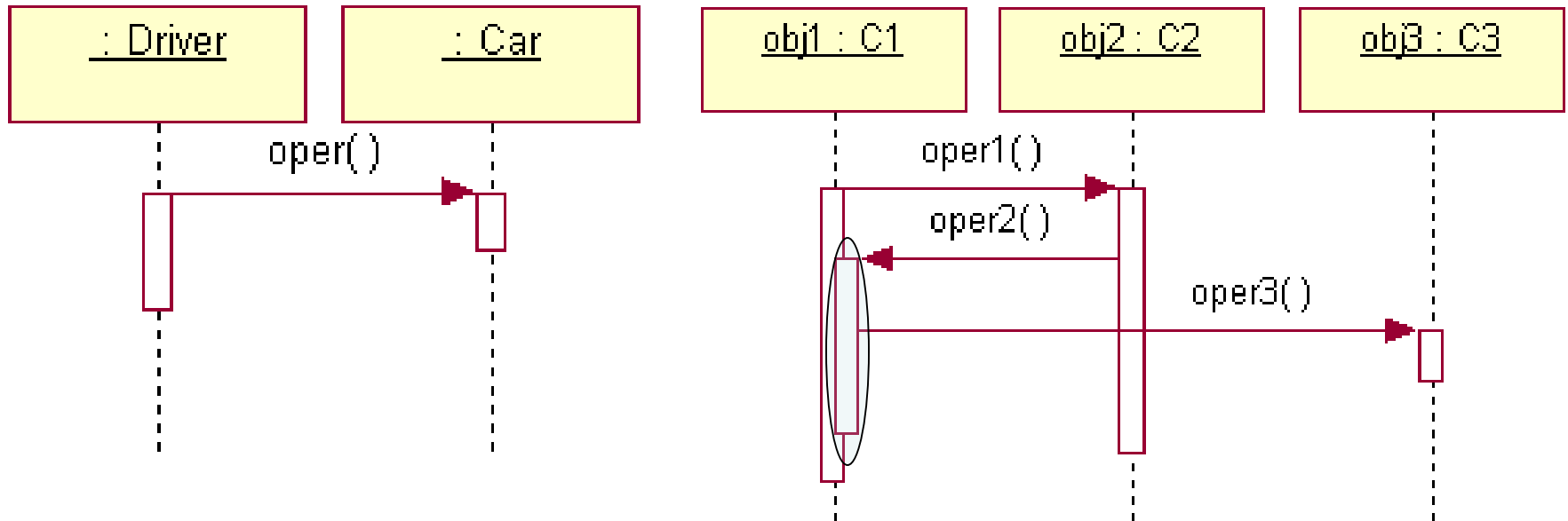
Lifeline



Focus of control

- The focus of control is **a tall, thin rectangle** that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure
- The top of the rectangle is aligned with the start of the action; the bottom is aligned with its completion (and can be marked by a return message)
- You can show the **nesting** of a focus of control (caused by recursion, a call to a self-operation, or by a callback from another object) by stacking another focus of control slightly to the right of its parent (and can do so to an arbitrary depth)

Focus of control



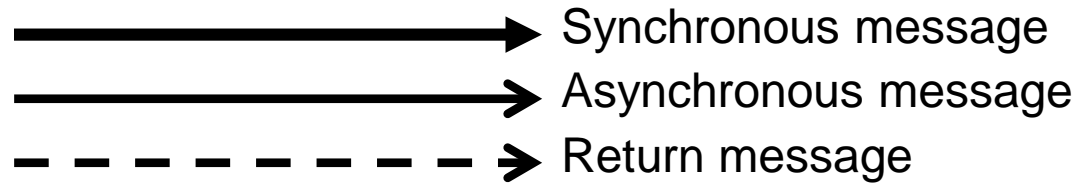
Message

- The main content in a sequence diagram is the messages
- A message is shown by an arrow from one lifeline to another
- The arrowhead points to the receiver

Message

- If the message is **asynchronous**, the line has a **stick arrowhead**
- If the message is **synchronous** (a call), the line has a **filled triangular arrowhead**
- A reply to a **synchronous** message (a return from a call) is shown by a **dashed arrow with a stick arrowhead**
- The return message may be omitted, as there is an implicit return after any call, but it is often useful for showing return values

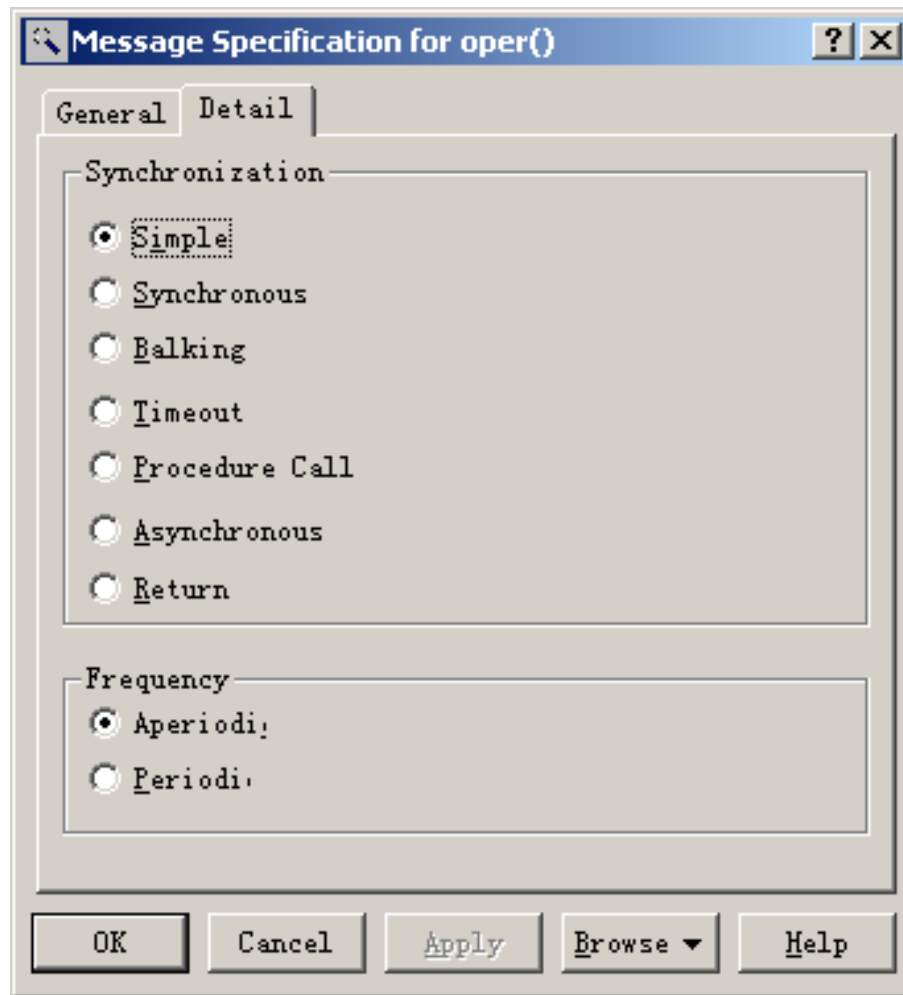
Message



Message

- When you pass a message, an action usually results on its receipt
 - An action may result in a change in state of the target object and objects accessible from it
- In the UML, you can model several kinds of messages
 - Call
 - Invokes an operation on an object; an object may send a message to itself, resulting in the local invocation of an operation
 - Return
 - Returns a value to the caller
 - Send
 - Sends a signal to an object
 - Create
 - Creates an object
 - Destroy
 - Destroys an object; an object may commit suicide by destroying itself

Messages in Rational Rose



Structured Control in Sequence Diagrams

- A sequence of messages is fine for showing a **single, linear sequence**
- But often we need to show **conditionals** and **loops**
- Sometimes we want to show **concurrent** execution of multiple sequences
- This kind of **high-level control** can be shown using **structured control operators** in sequence diagrams

Structured Control in Sequence Diagrams

- A control operator is shown as a rectangular region within the sequence diagram
- It has a tag text label inside a small pentagon in the upper left corner to tell what kind of a control operator it is
- The operator applies to the lifelines that cross it. This is considered the body of the operator

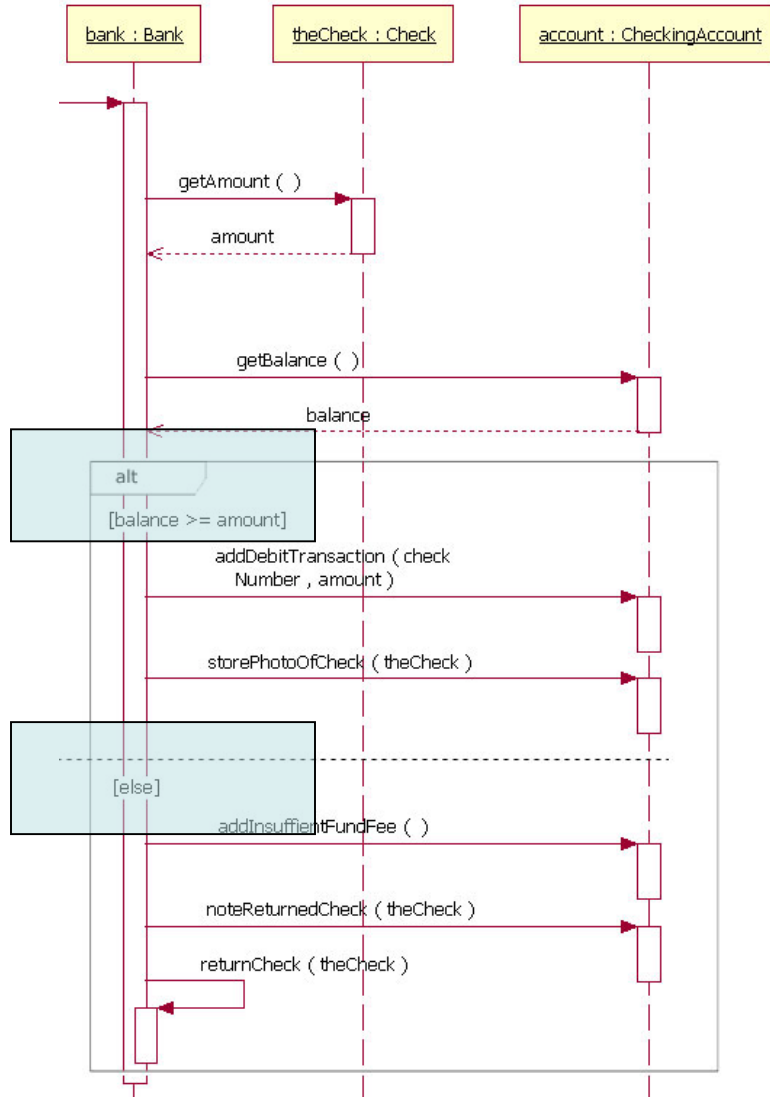
Structured Control: Optional execution

- The tag is **opt**
- The body of the control operator is executed if a guard condition is true when the operator is entered

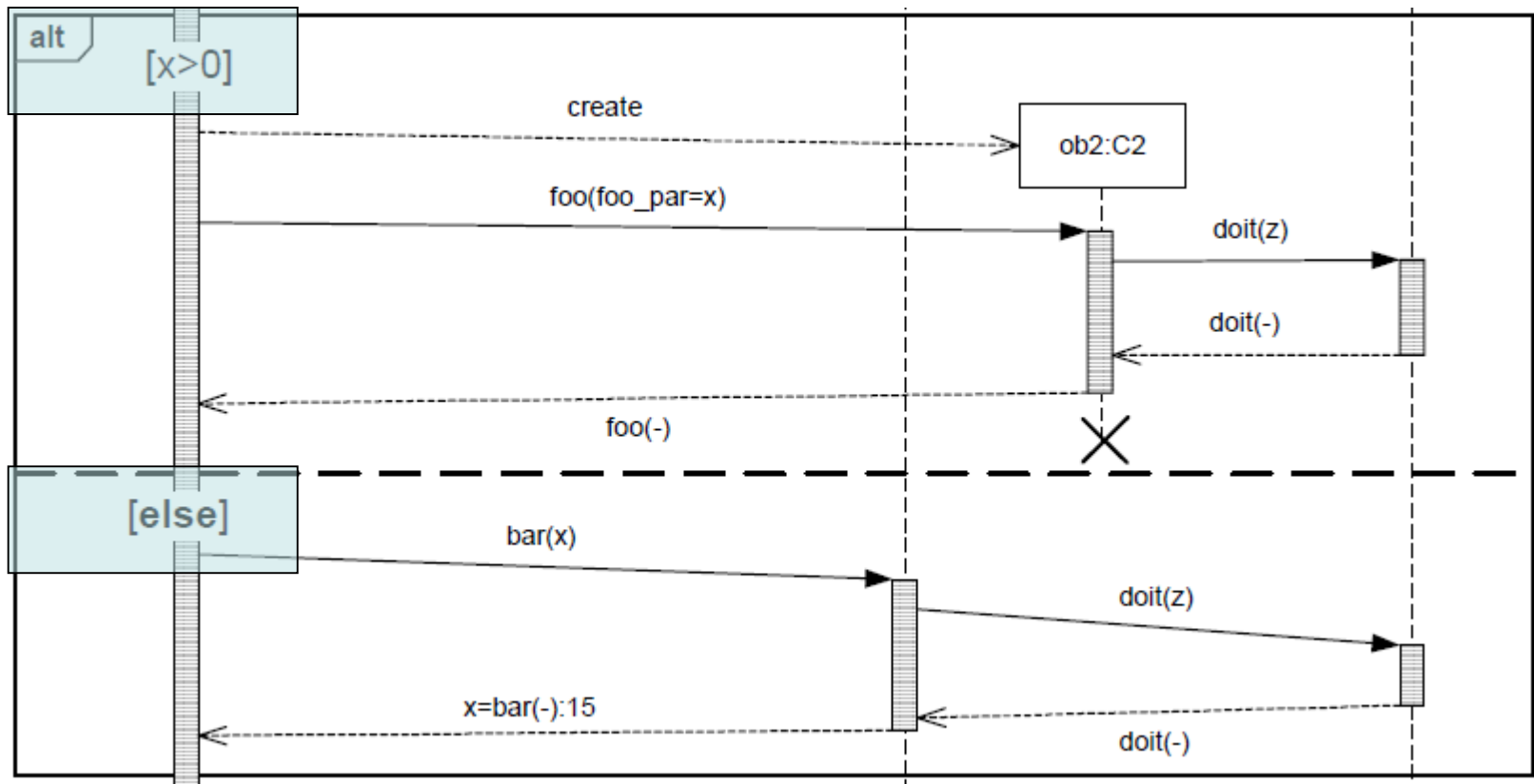
Structured Control: Conditional Execution

- The tag is **alt**
- The body of the control operator is divided into multiple subregions by horizontal dashed lines
- Each subregion represents one branch of a conditional
- Each subregion has a guard condition. If the guard condition for a subregion is true, the subregion is executed. However, at most one subregion may be executed

Structured Control: Conditional Execution



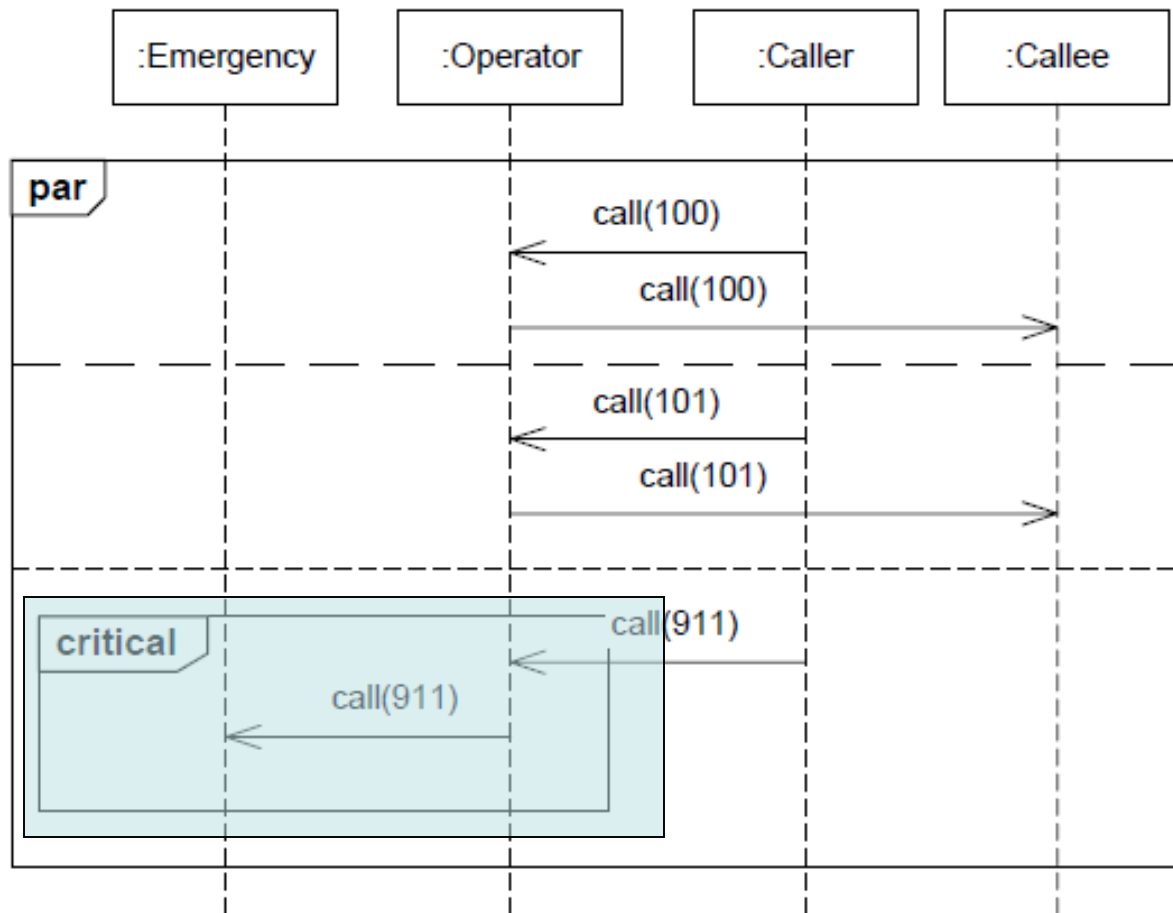
Structured Control: Conditional Execution



Structured Control: Parallel Execution

- The tag is **par**
- The body of the control operator is divided into multiple subregions by horizontal dashed lines
- Each subregion represents a parallel (concurrent) computation
- In most cases, each subregion involves different lifelines. When the control operator is entered, all of the subregions execute concurrently

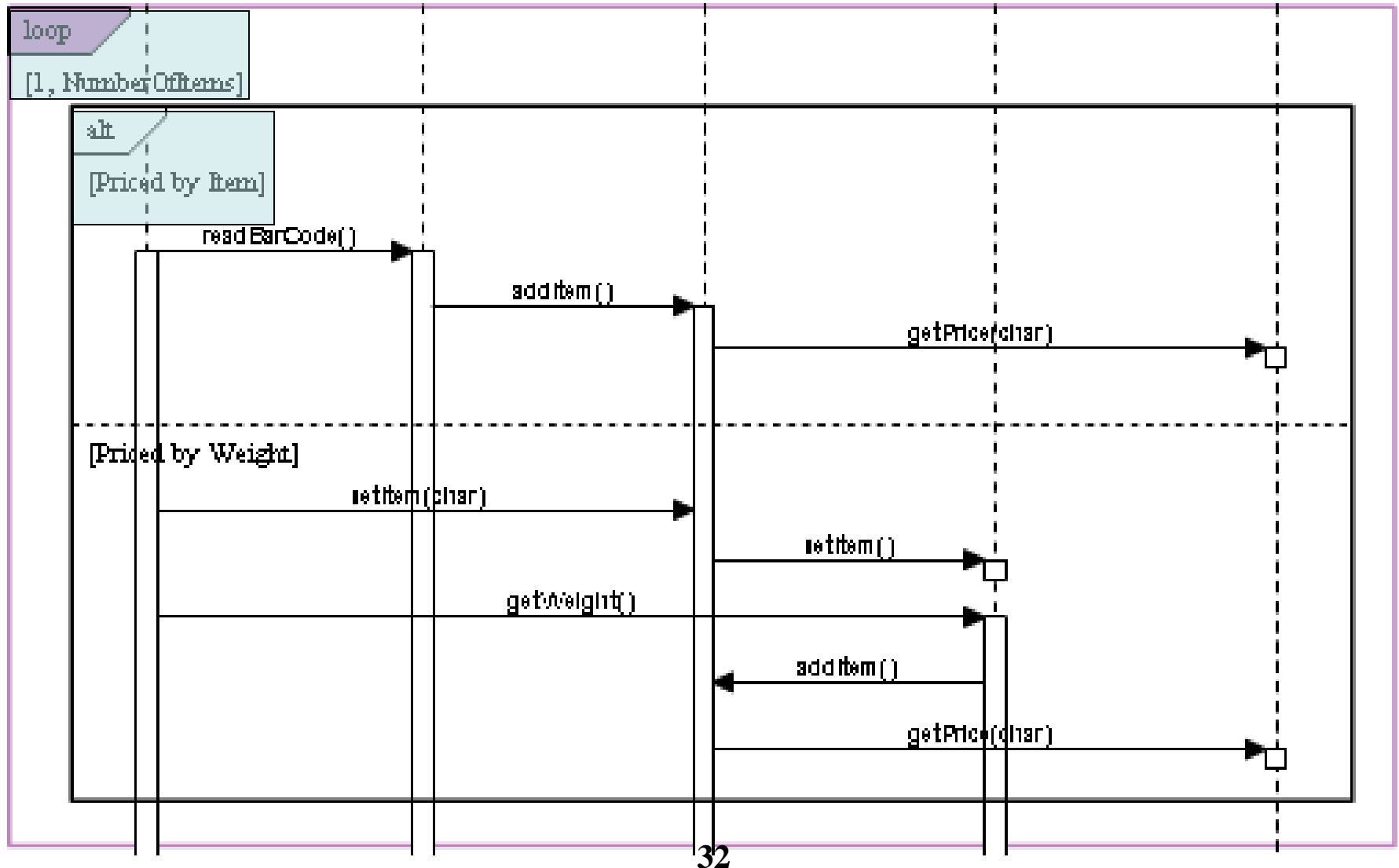
Structured Control: Parallel Execution



Structured Control: Loop (iterative) Execution

- The tag is **loop**
- A guard condition appears at the top of one lifeline within the body. The body of the loop is executed repeatedly as long as the guard condition is true before each iteration
- When the guard condition is false at the top of the body, control passes out of the control operator

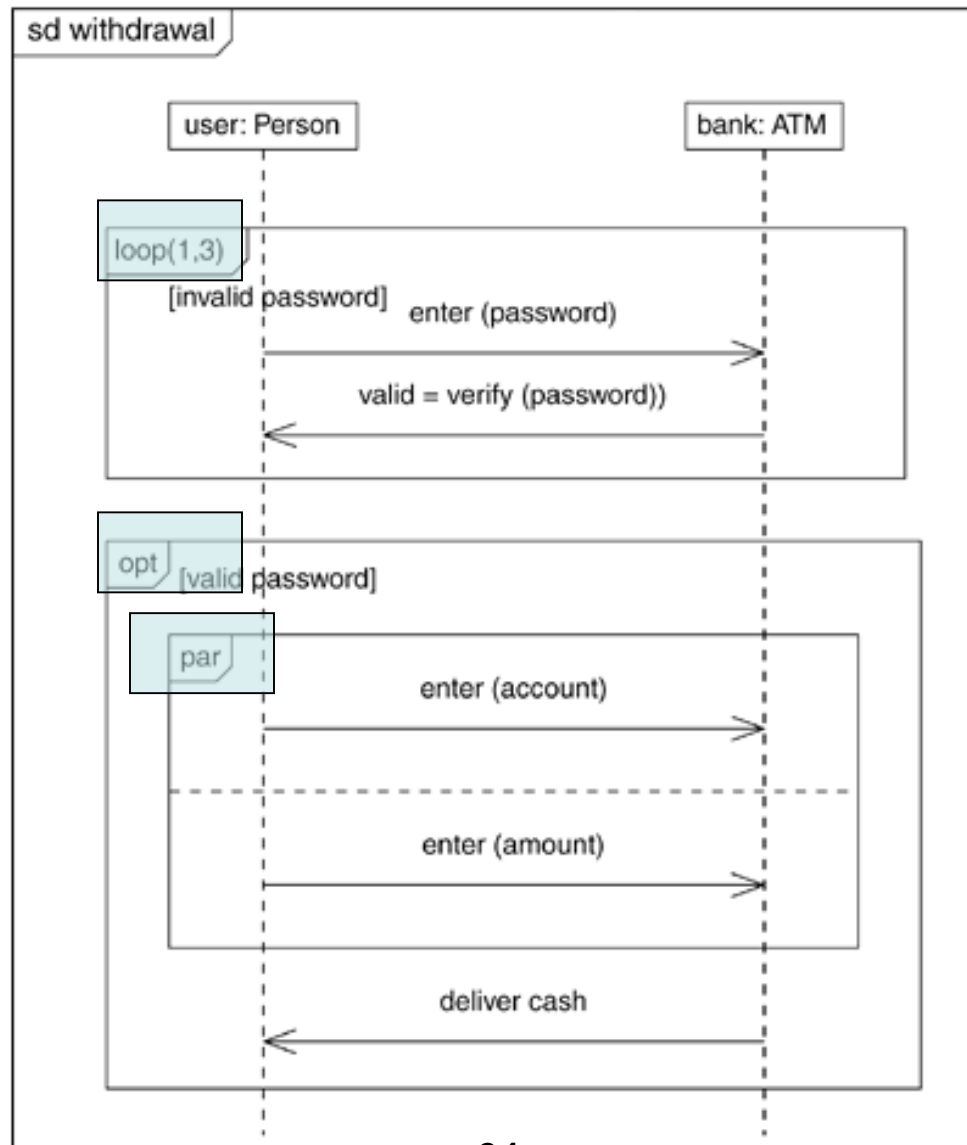
Structured Control: Loop (iterative) Execution



Other Structured Control Operators

- Critical
- Ref

Structured Control in Sequence Diagrams



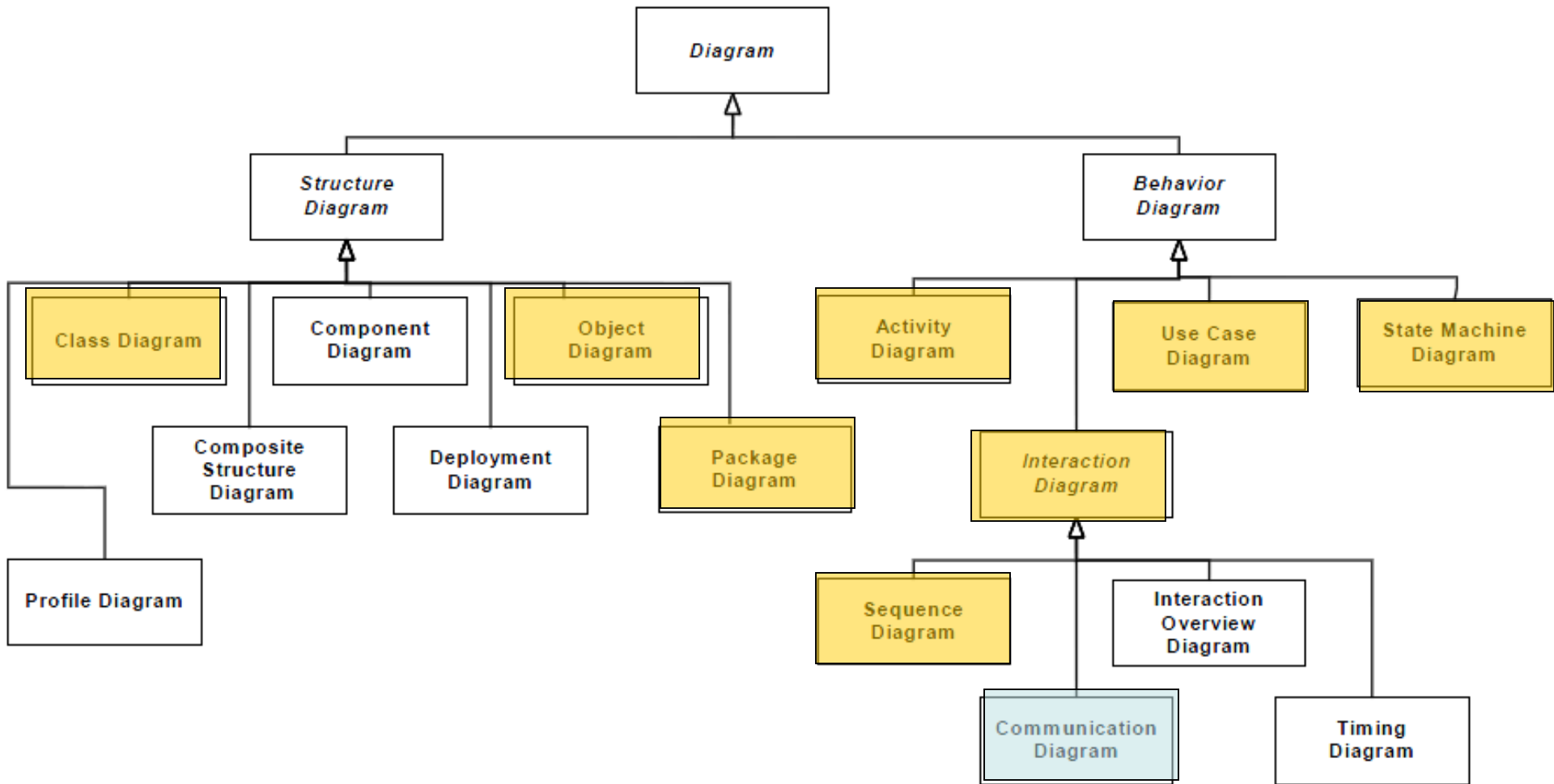
Steps of Sequence Diagram Modeling

- Set the context for the interaction
- Set the stage for the interaction by identifying which objects play a role in the interaction
- Set the lifeline for each object
- Starting with the message that initiates this interaction, lay out each subsequent message from top to bottom between the lifelines
- If you need to visualize the nesting of messages or the points in time when actual computation is taking place, adorn each object's lifeline with its focus of control
- If you need to specify time or space constraints, adorn each message with a timing mark and attach suitable time or space constraints
- If you need to specify this flow of control more formally, attach pre- and postconditions to each message

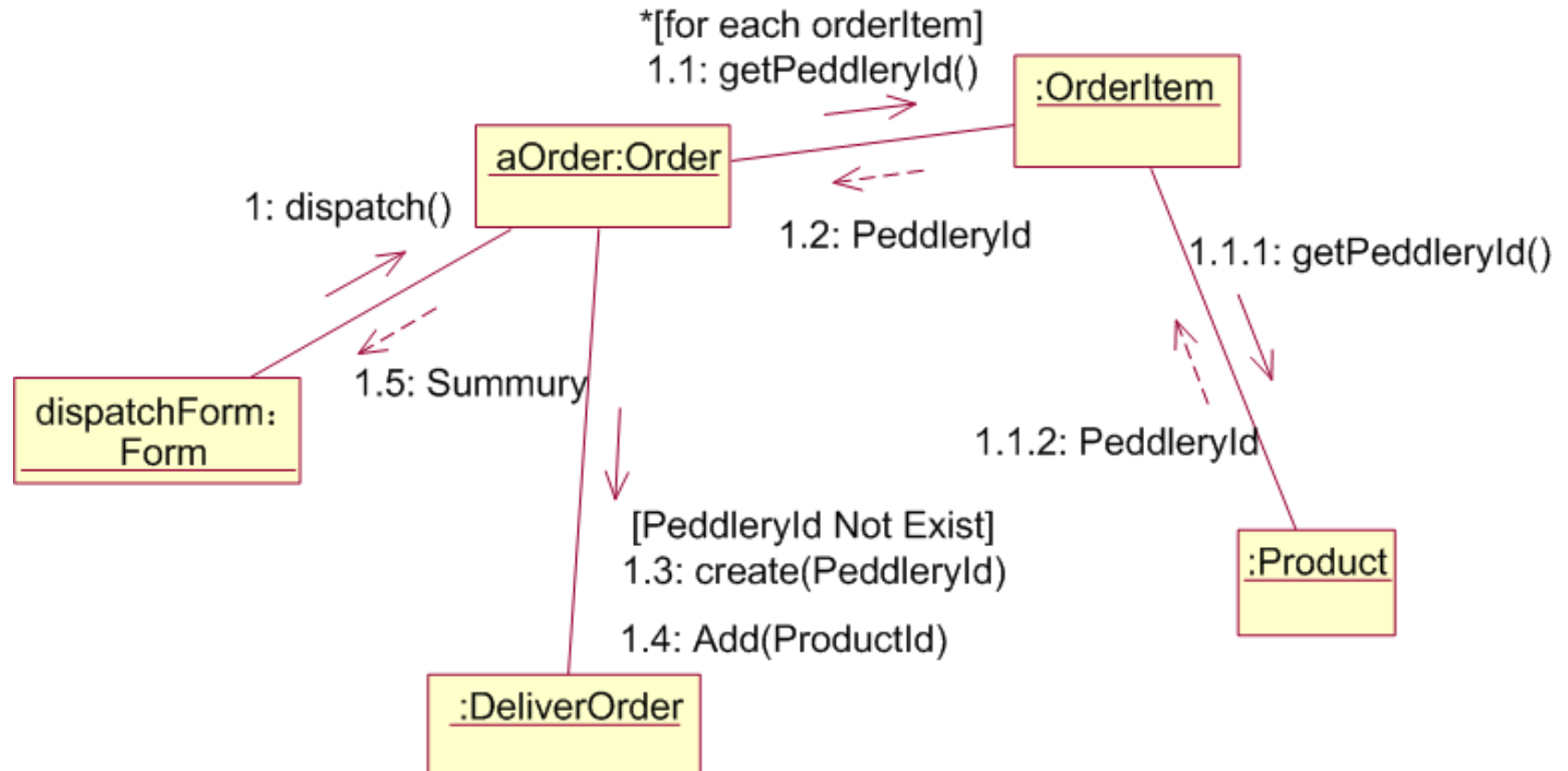
Style of UML Sequence Diagram

- Strive for Left-to-Right Ordering of Messages
- Name Objects When Several of the Same Type Exist
- Focus on Critical Interactions

Communication Diagram



Example of Communication Diagram

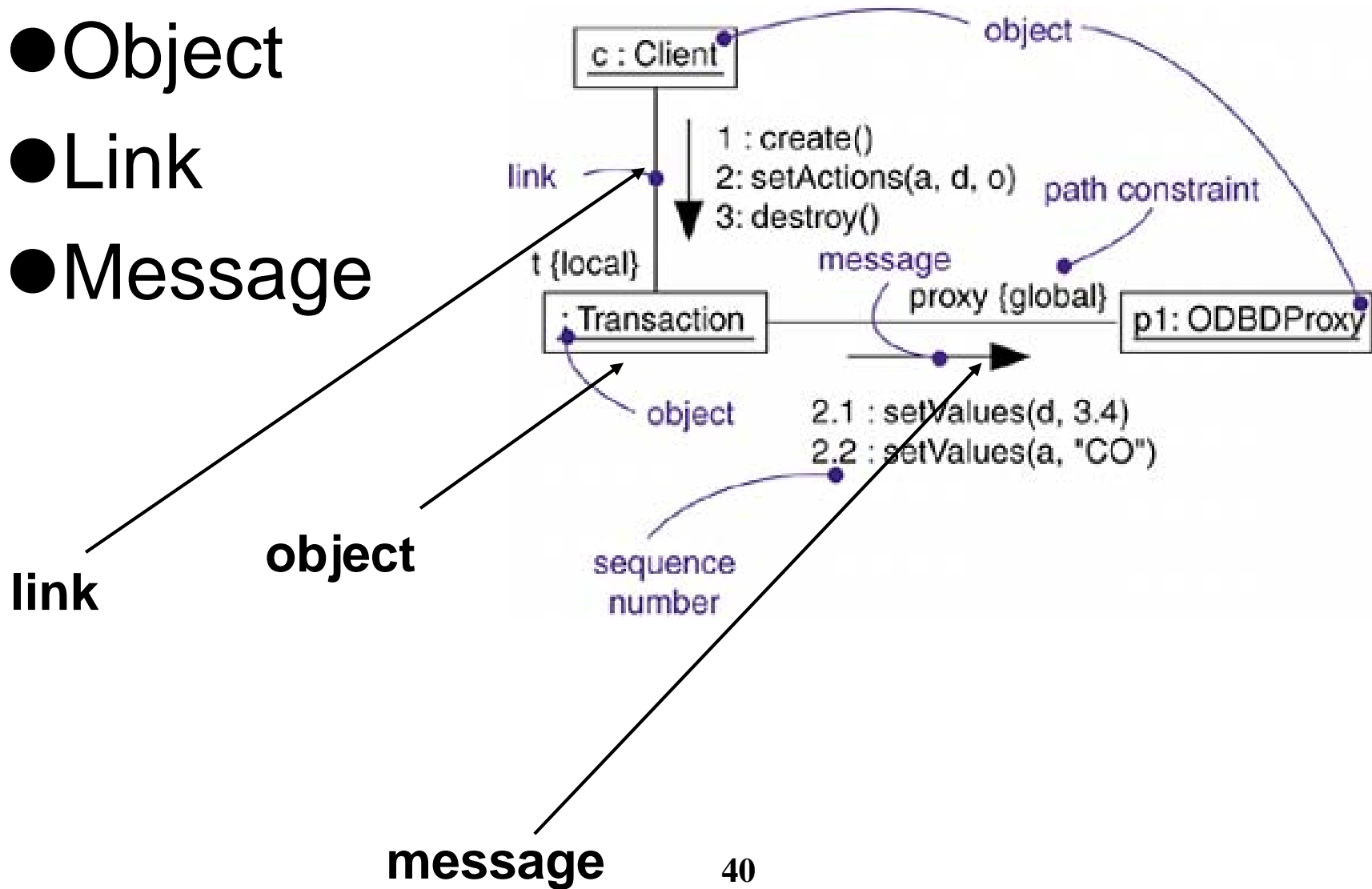


Communication Diagram

- A communication diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages

Constituent Elements of Communication Diagram

- Object
- Link
- Message



Link

- A link is a semantic connection among objects
- In general, a link is an instance of an association

Message

- To indicate the time order of a message, you prefix the message with a number (starting with the message numbered 1), increasing monotonically for each new message in the flow of control (2, 3, and so on)
- To show nesting, you use Dewey decimal numbering (1 is the first message, which contains message 1.1 and message 1.2, and so on)

Steps of Sequence Communication Modeling

- Set the context for the interaction
- Set the stage for the interaction by identifying which objects play a role in the interaction
- Specify the links among these objects, along which messages may pass
- Starting with the message that initiates this interaction, attach each subsequent message to the appropriate link, setting its sequence number, as appropriate
- Show nesting by using Dewey decimal numbering
- If you need to specify time or space constraints, adorn each message with a timing mark and attach suitable time or space constraints
- If you need to specify this flow of control more formally, attach pre- and postconditions to each message

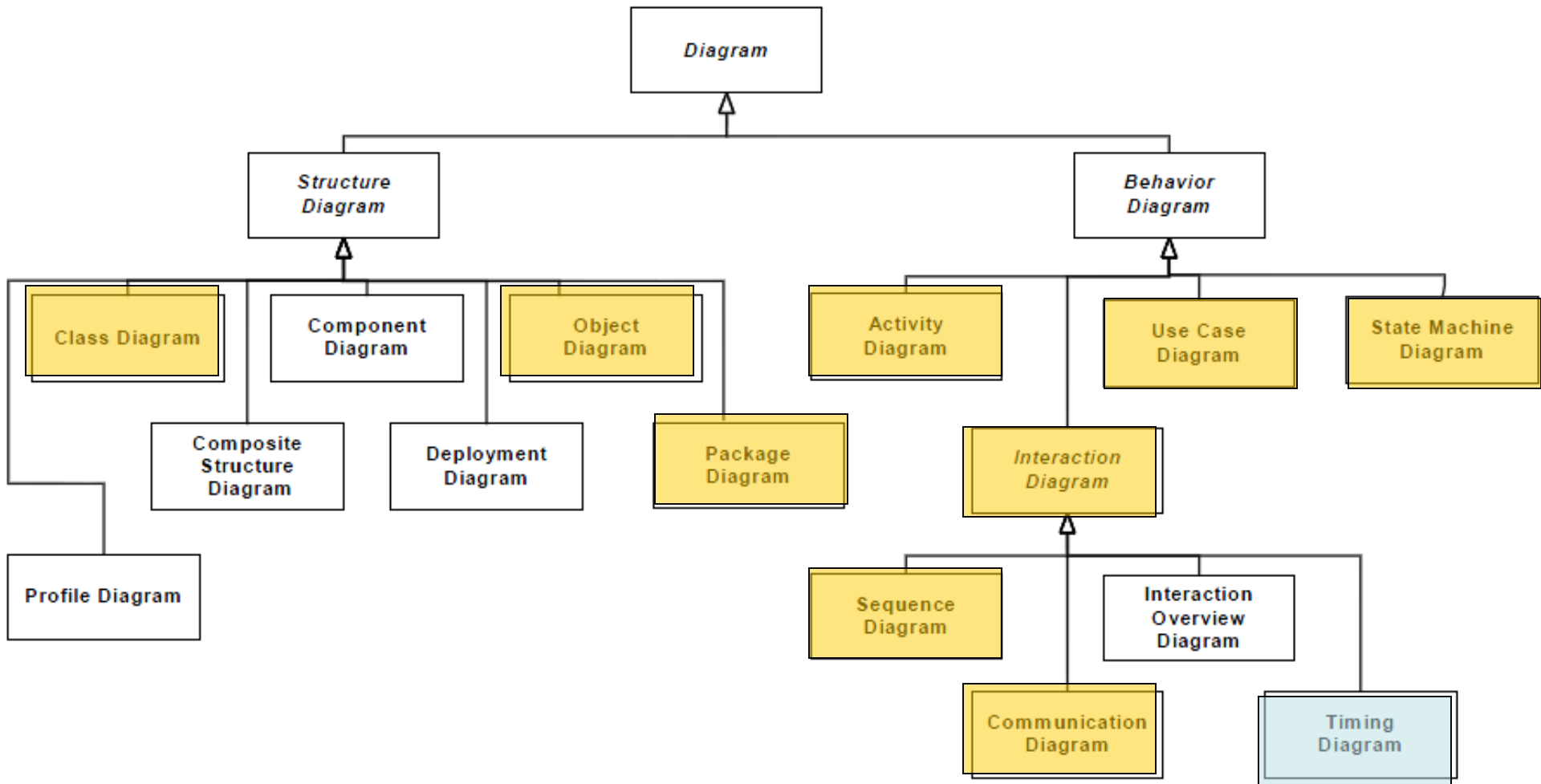
Style of UML Communication Diagram

- Do Not Use Communication Diagrams to Model Process Flow
 - If you want to model process or data flow, then you should consider drawing a UML activity diagram
- Create a Sequence Diagram When Sequence Is Important

Comparison between Sequence and Communication Diagram

- A sequence diagram emphasizes the time ordering of messages. A communication diagram emphasizes the organization of the objects that participate in an interaction
- Because they both derive from the same information in the UML's metamodel, sequence diagrams and communication diagrams are semantically equivalent
- Sequence Diagram
 - Lifeline
 - Focus of control
- Communication Diagram
 - Sequence number

Timing Diagram

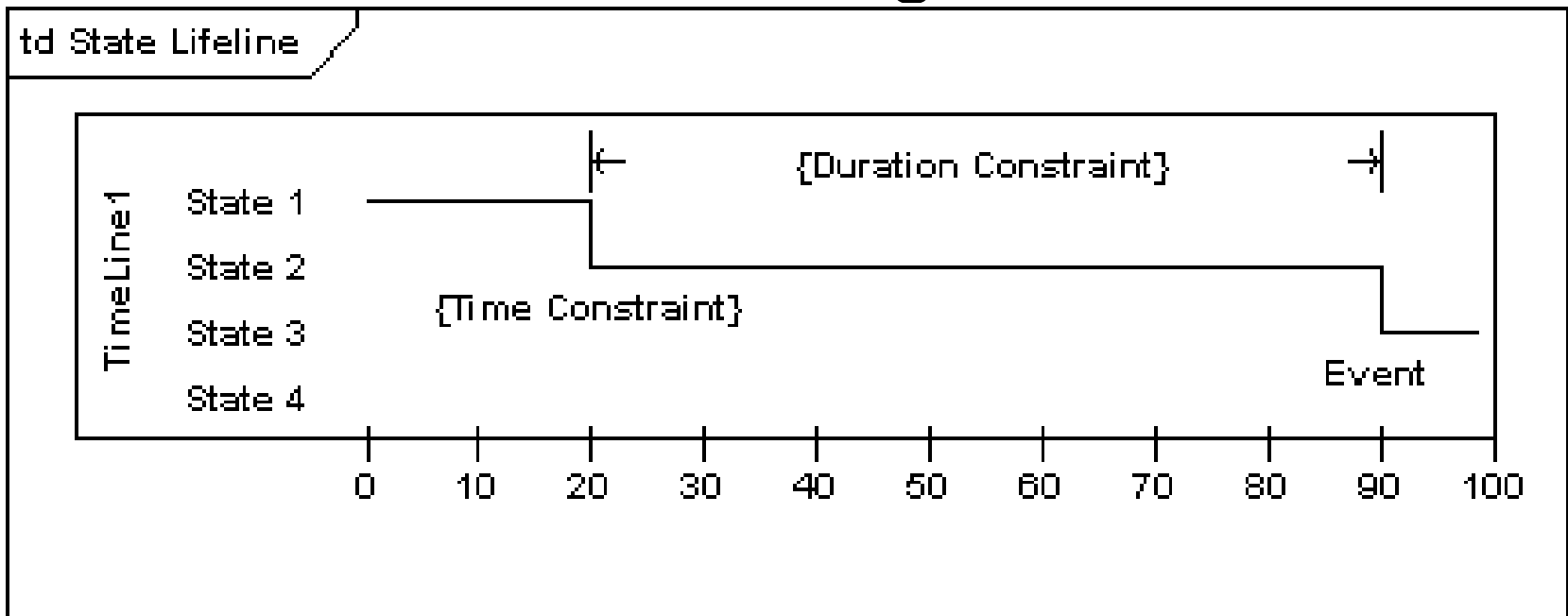


Timing Diagram

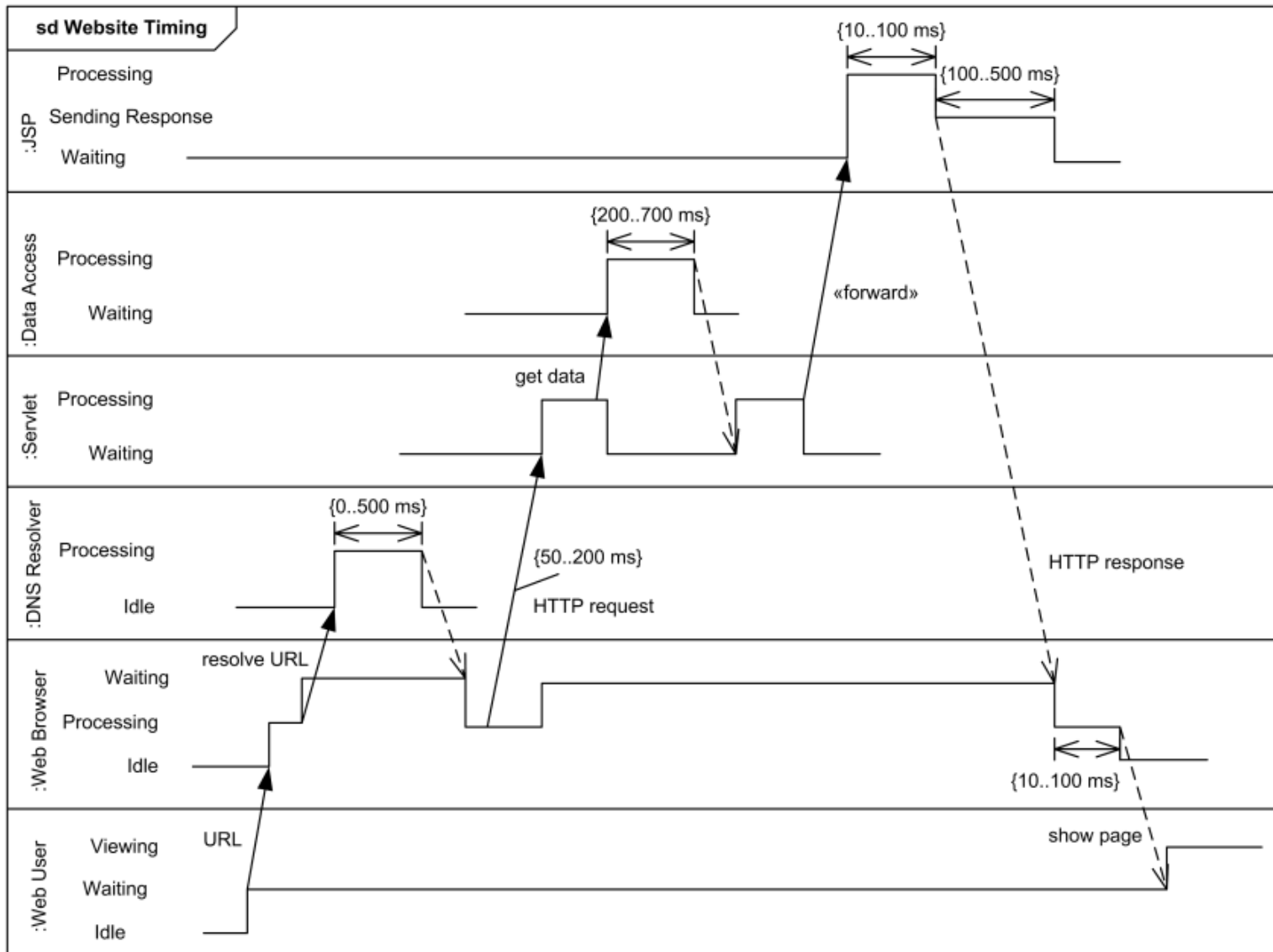
- Timing diagrams focus on conditions changing within and among Lifelines along a linear time axis
- UML timing diagrams are used to display the change in state or value of one or more elements over time. It can also show the interaction between timed events and the time and duration constraints that govern them

Timing Diagram

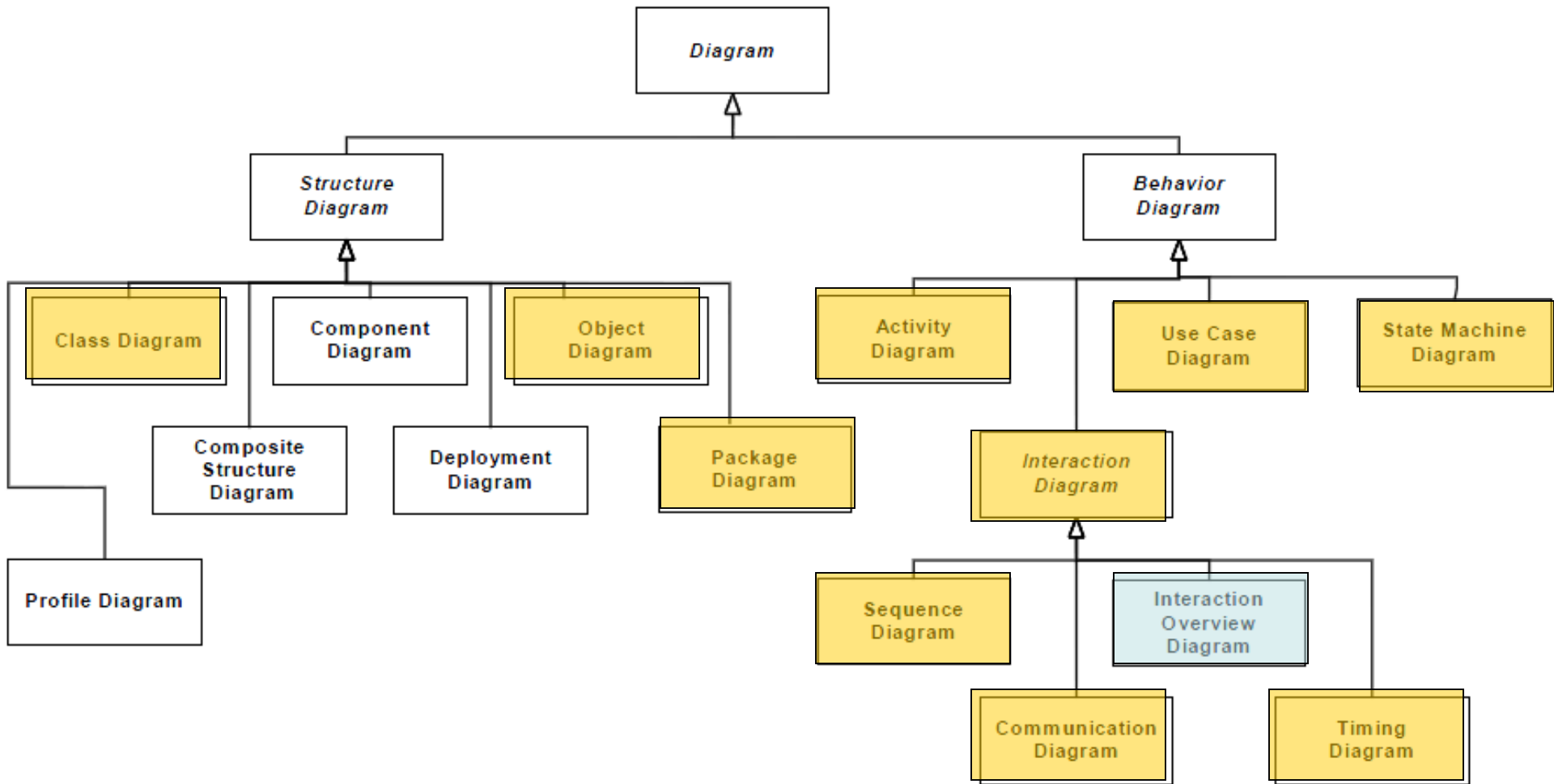
- The X-axis displays elapsed time in whatever units are chosen, while the Y-axis is labelled with a given list of states



Example of Timing Diagram



Interaction Overview Diagram



Interaction Overview Diagram

- Interaction Overview Diagrams define Interactions through a variant of Activity Diagrams in a way that promotes overview of the control flow
 - Interaction Overview Diagrams focus on **the overview of the flow of control** where the nodes are Interactions or InteractionUses
 - **The Lifelines and the Messages do not appear** at this overview level

Interaction Overview Diagram

- In some places relegates these diagrams as interaction diagrams while in other places interaction overview diagrams are referred to as specialization of activity diagrams

Example

