E-course

What Is Object-Orientation?

E-course

OBJECT INTERACTION

SEQUENCE DIAGRAMS

In This Lecture You Will Learn:

E-course

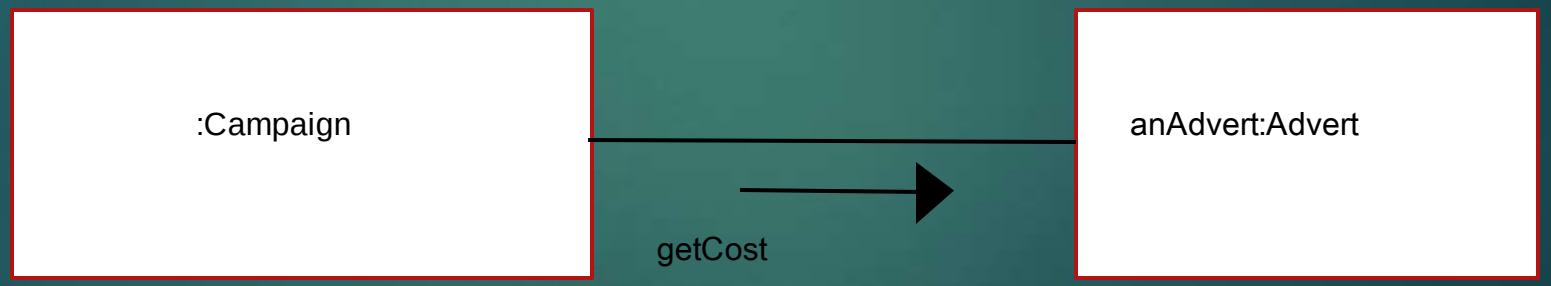
- how to develop object interaction from use cases;
- how to model object interaction using an interaction sequence diagram;
- how to cross-check between interaction diagrams and a class diagram.

Object Messaging

E-course

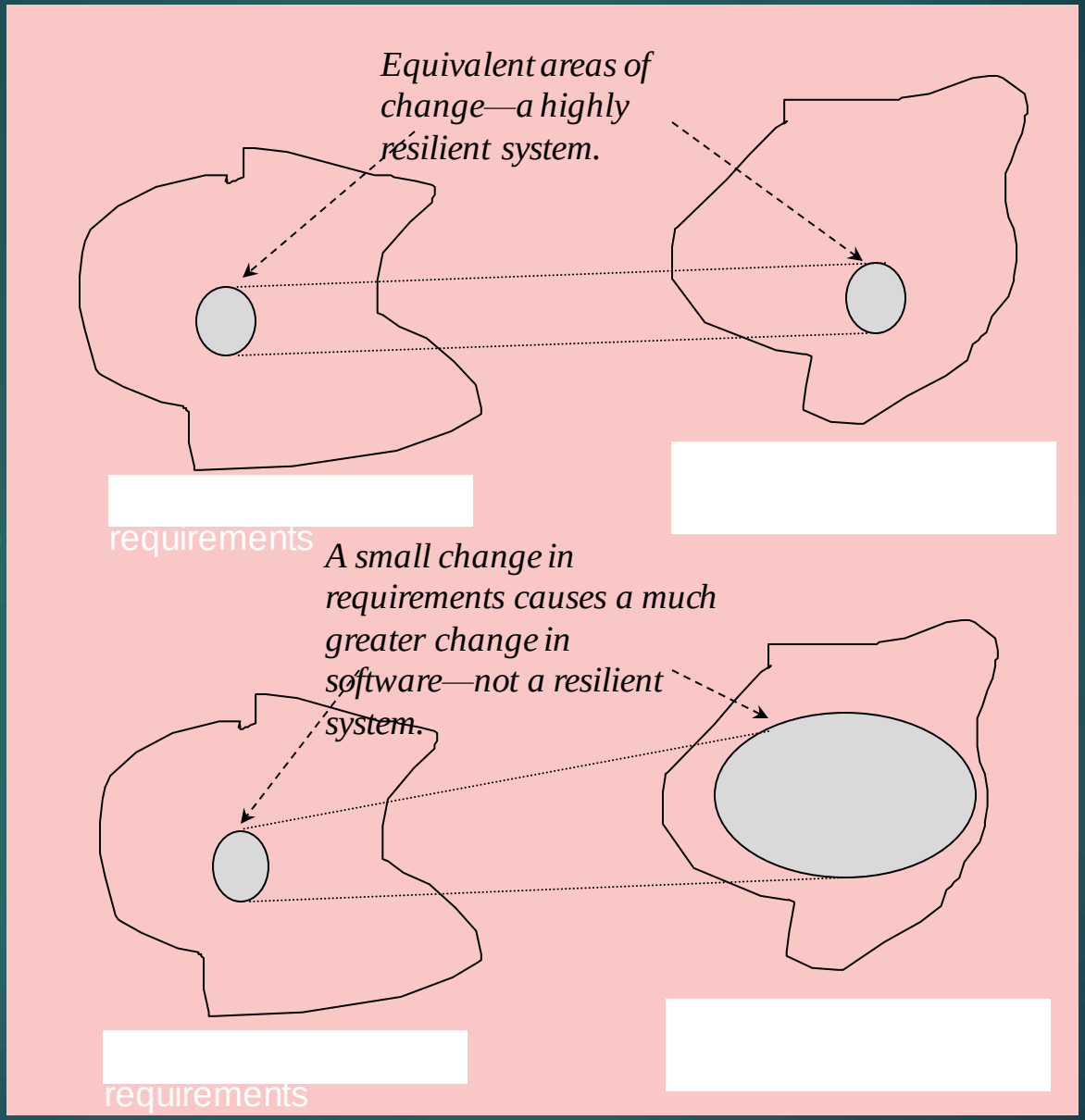
Objects communicate by sending messages.
Sending the message getCost() to an Advert object, might use the following syntax.

currentadvertCost = anAdvert.getCost()



E-course

Resilience of Design



Interaction & Collaboration

E-course

- A collaboration is a group of objects or classes that work together to provide an element of functionality or behaviour.

- An interaction defines the message passing between lifelines (e.g. objects) within the context of a collaboration to achieve a particular behaviour.

Modelling Interactions

E-course

- Interactions can be modelled using various notations
  - Interaction sequence diagrams
  - Communication diagrams
  - Interaction overview diagrams
  - Timing diagrams

Sequence Diagrams

E-course

- An interaction diagram models the behavior of a group of objects that work together to achieve a user goal.
- A sequence diagram helps us identify a set of collaborating objects involved in a scenario of a use case.
- A sequence diagram has two dimensions: the vertical dimension and the horizontal dimension, respectively representing the passage of time and the objects involved in the interaction.
- Object icons are placed horizontally at the top of the sequence diagram, and messages are passed between them.

Sequence Diagrams

E-course

- Shows an interaction between lifelines (e.g. objects) arranged in a time sequence.
- Can be drawn at different levels of detail and to meet different purposes at several stages in the development life cycle.
- Typically used to represent the detailed object interaction that occurs for one use case or for one operation.

Sequence Diagrams

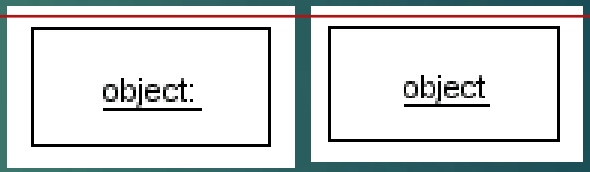
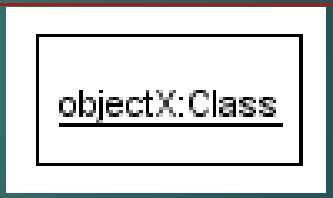
E-course

- Vertical dimension shows time.
- Objects (or subsystems or other connectable objects) involved in interaction appear horizontally across the page and are represented by lifelines.
- Messages are shown by a solid horizontal arrow.
- The execution or activation of an operation is shown by a rectangle on the relevant lifeline.

Common UML Interaction Diagram Notation

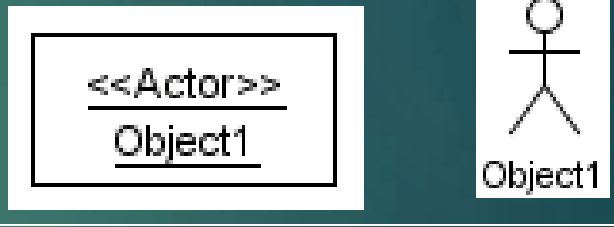
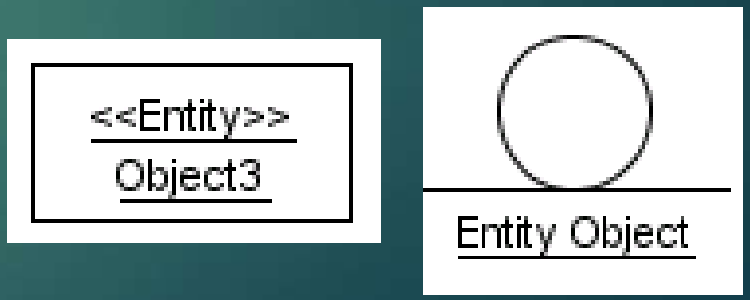
E-course

- **Object Symbol**

Naming Format	Notation
An object of an unspecified class.	
A named object of a specified class.	
An unnamed object of a specified class.	

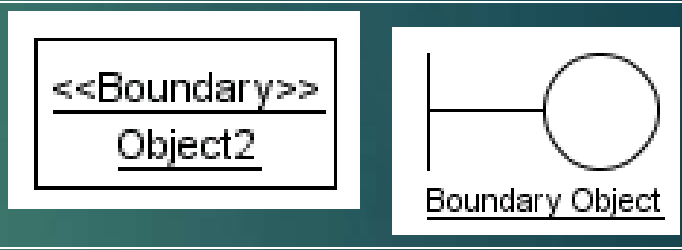
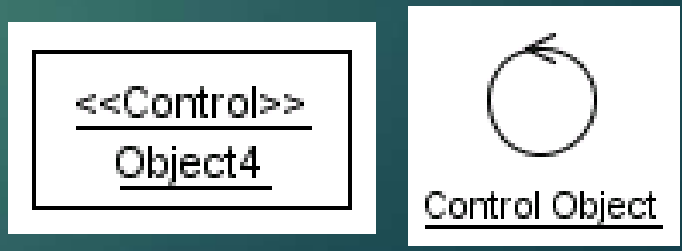
Object Stereotype

E-course

Object Category	Description	Graphical Notations
Actor Object	An external entity that interacts with the system.	
Entity Object	An object that models the data in the system. It often represents an object in the problem domain.	

Object Stereotype (cont'd)

E-course

Object Category	Description	Graphical Notations
Boundary Object	An object that handles the communication between actor objects and the system.	
Control Object	An object that models the flow of control and functionality that do not naturally belong to entity objects or boundary objects.	

Messages

E-course

Message	Description	Notation
Procedure call or other nested flow of control	The message sender waits for the completion of the procedure call of the message receiver.	
Asynchronous communication	The sender dispatches a message and immediately continues with the next step of execution.	

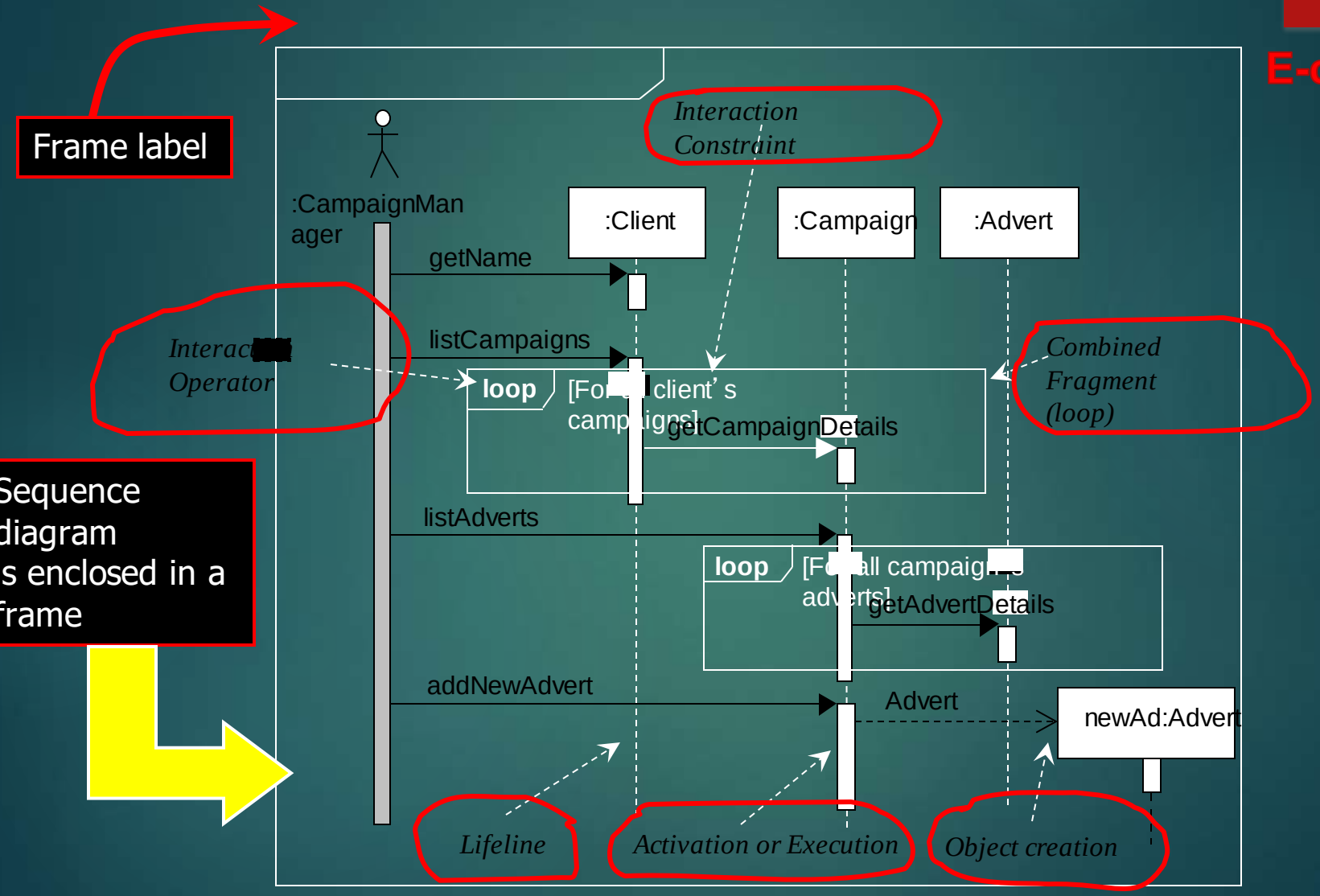
Messages (cont'd)

E-course

Message	Description	Notation
Return message	Message returned from the procedure call.	
Message with travel delay	The message will take a significant amount of time to arrive at the receiving object. (This is only used in sequence diagrams.)	

Sequence Diagram

E-course



Sequence diagram is enclosed in a frame

Sequence Diagram

E-course

- Iteration is represented by *combined fragment* rectangle with the *interaction operator* 'loop' .
- The loop combined fragment only executes if the guard condition in the interaction constraint evaluates as true.
- Object creation is shown with the construction arrow (dashed) going to the object symbol for the Advert lifeline.

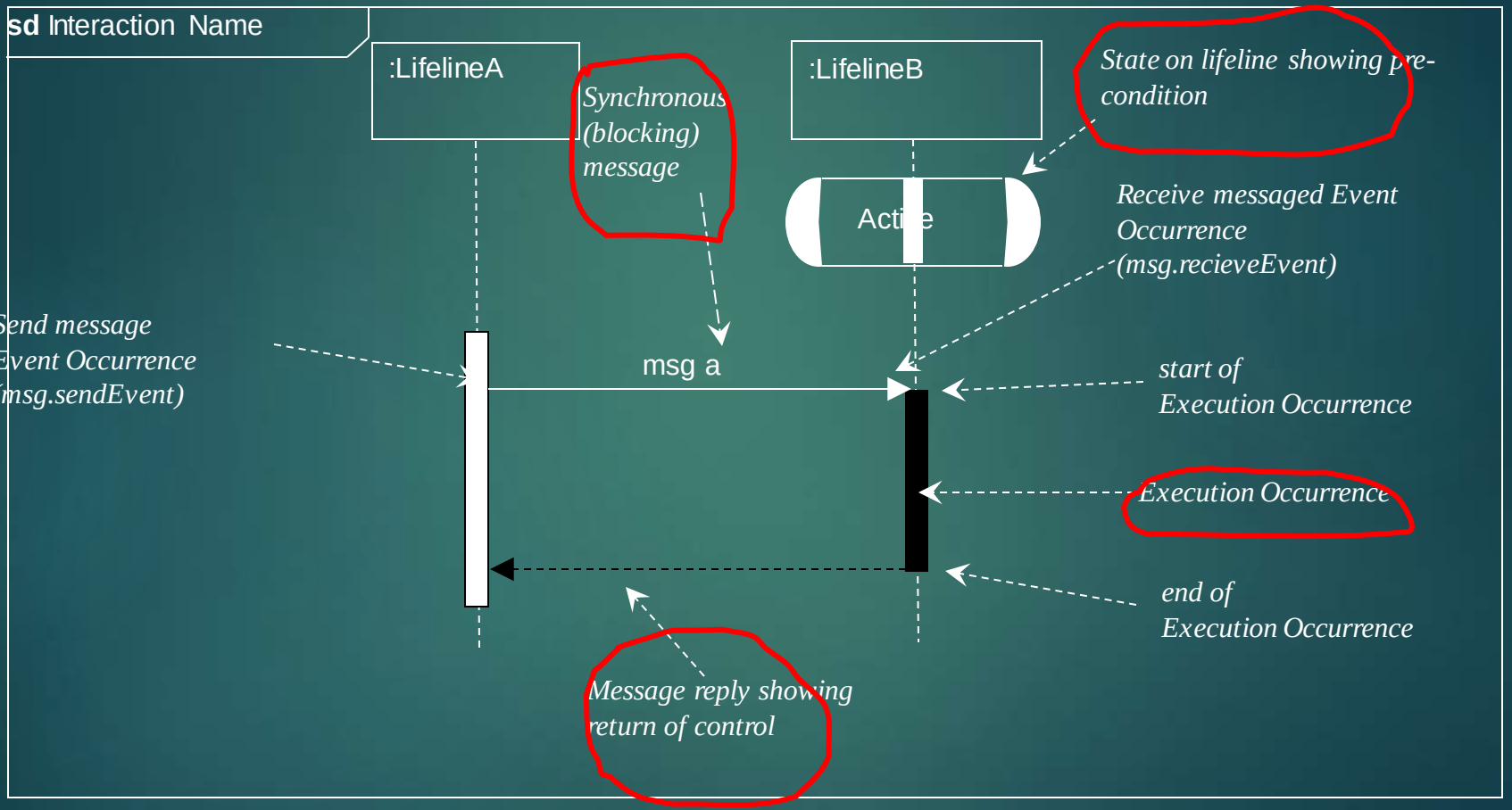
Synchronous Message

E-course

- A *synchronous message* or *procedural call* is shown with a full arrowhead, causes the invoking operation to suspend execution until the focus of control has been returned to it.

Further Notation

E-course

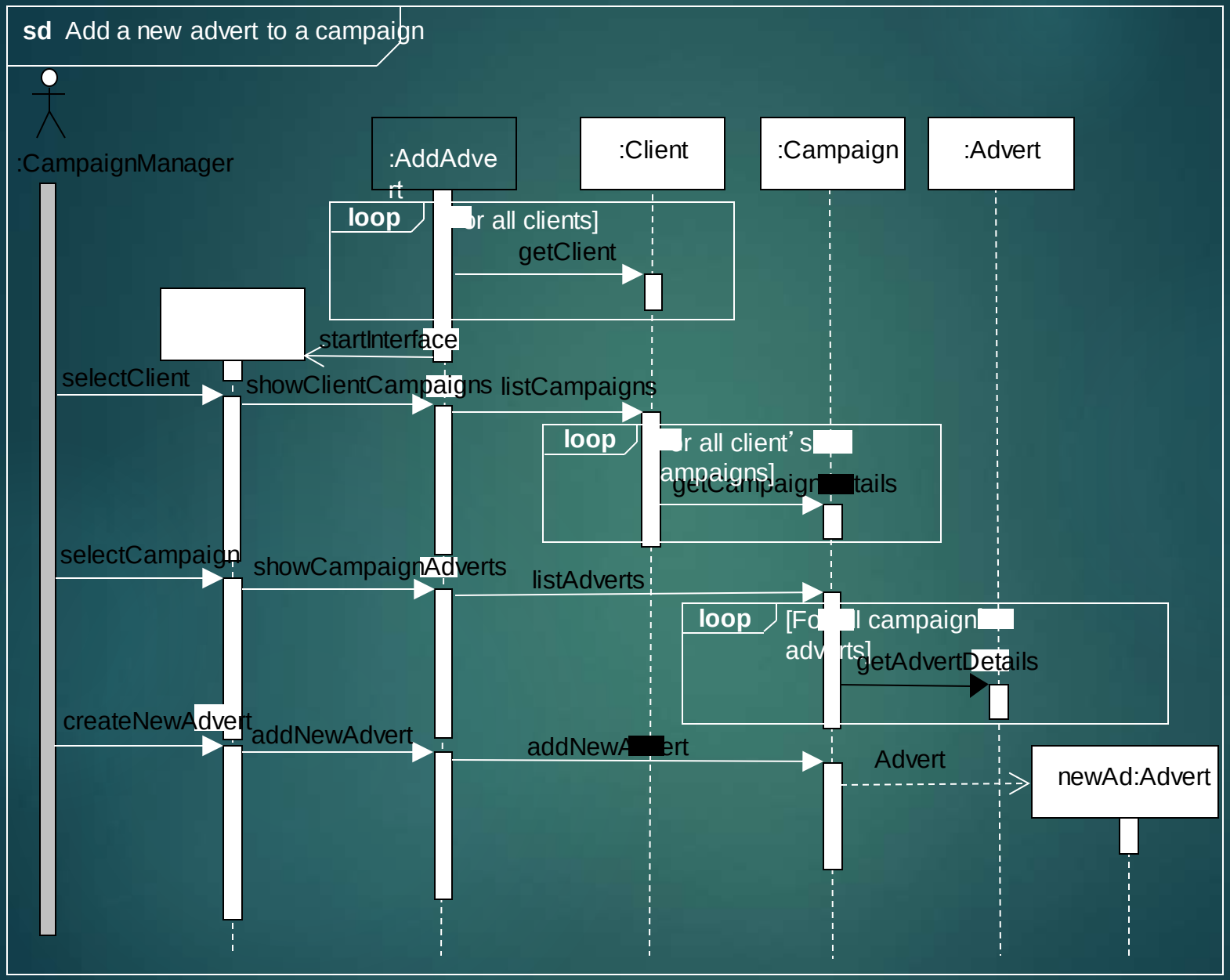


Boundary & Control Classes

E-course

- Most use cases imply at least one boundary object that manages the dialogue between the actor and the system – in the next sequence diagram it is represented by the lifeline :AddAdvertUI

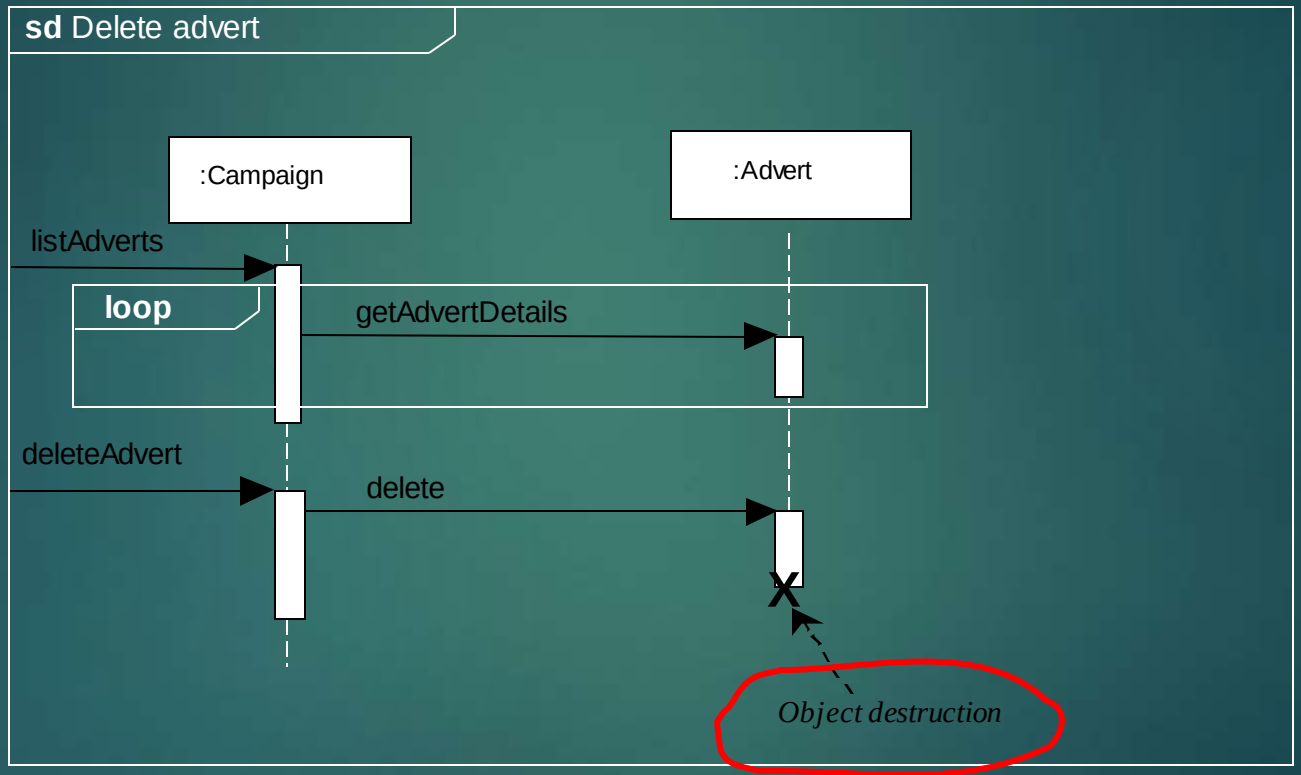
- The control object is represented by the lifeline :AddAdvert and this manages the overall object communication.



E-course

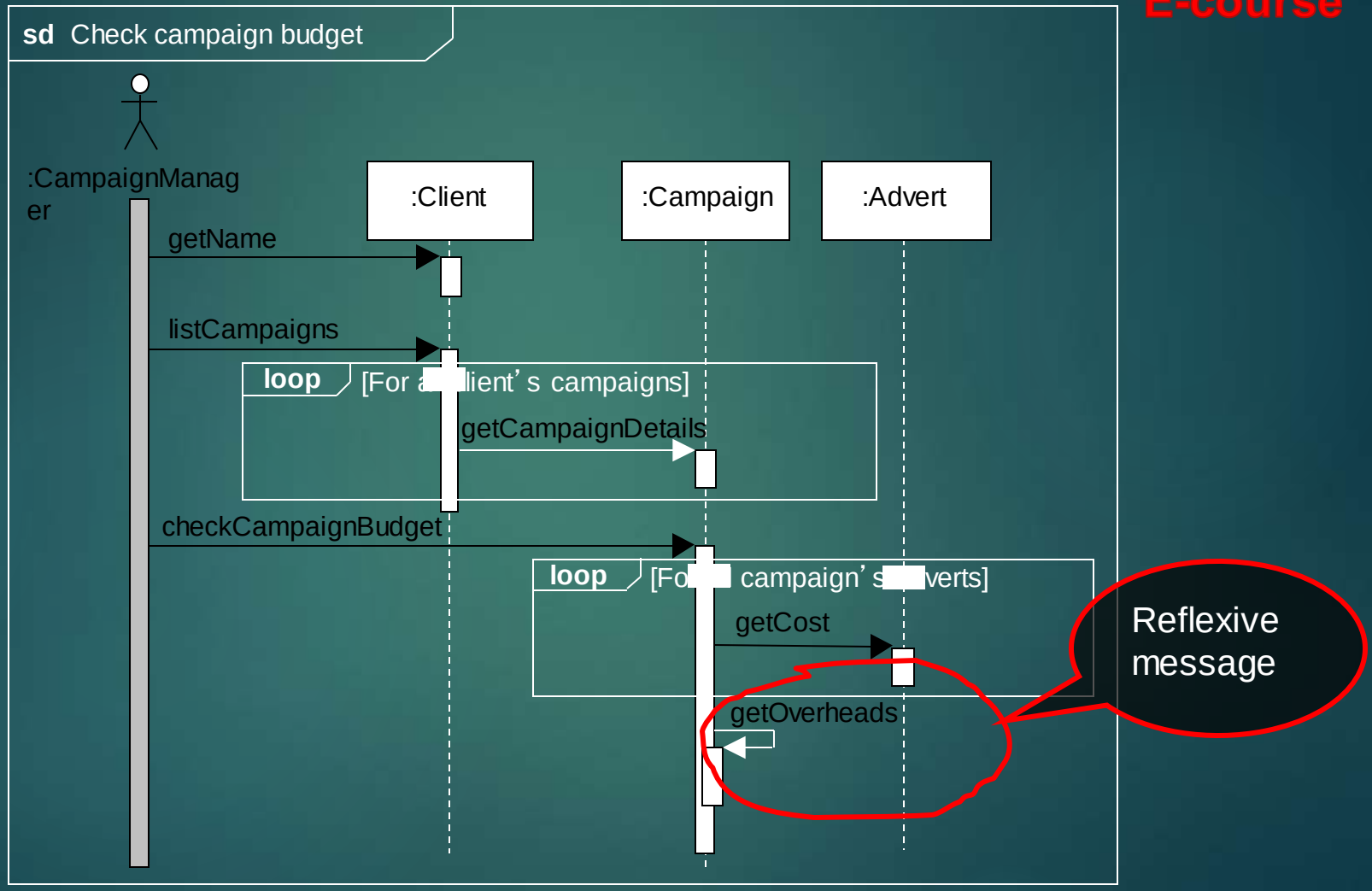
Object Destruction

E-course



Reflexive Messages

E-course



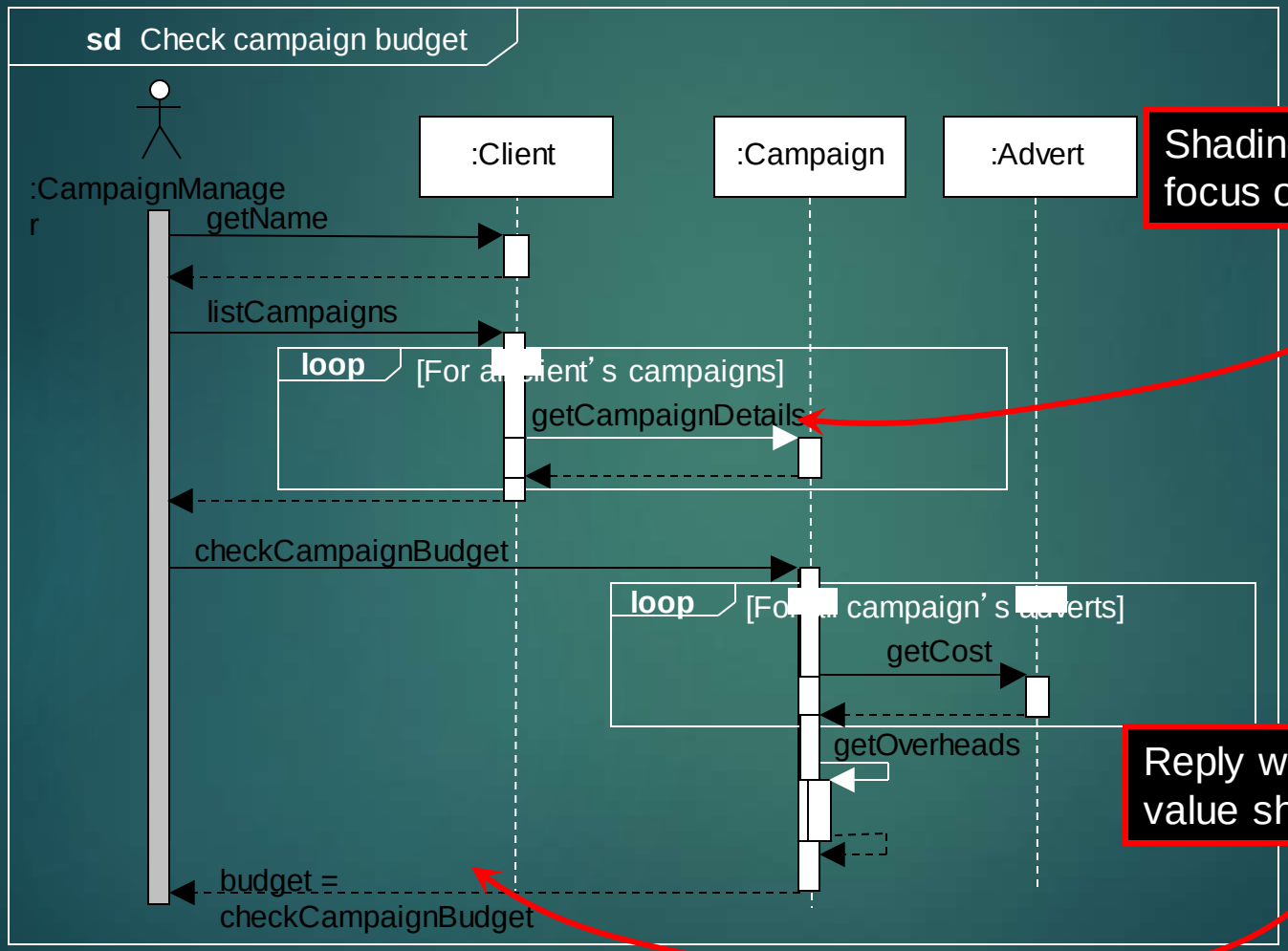
Focus of Control

E-course

- Indicates times during an activation when processing is taking place within that object.
- Parts of an activation that are not within the focus of control represent periods when, for example, an operation is waiting for a return from another object.
- May be shown by shading those parts of the activation rectangle that correspond to active processing by an operation.

Focus of Control

E-course



Reply Message

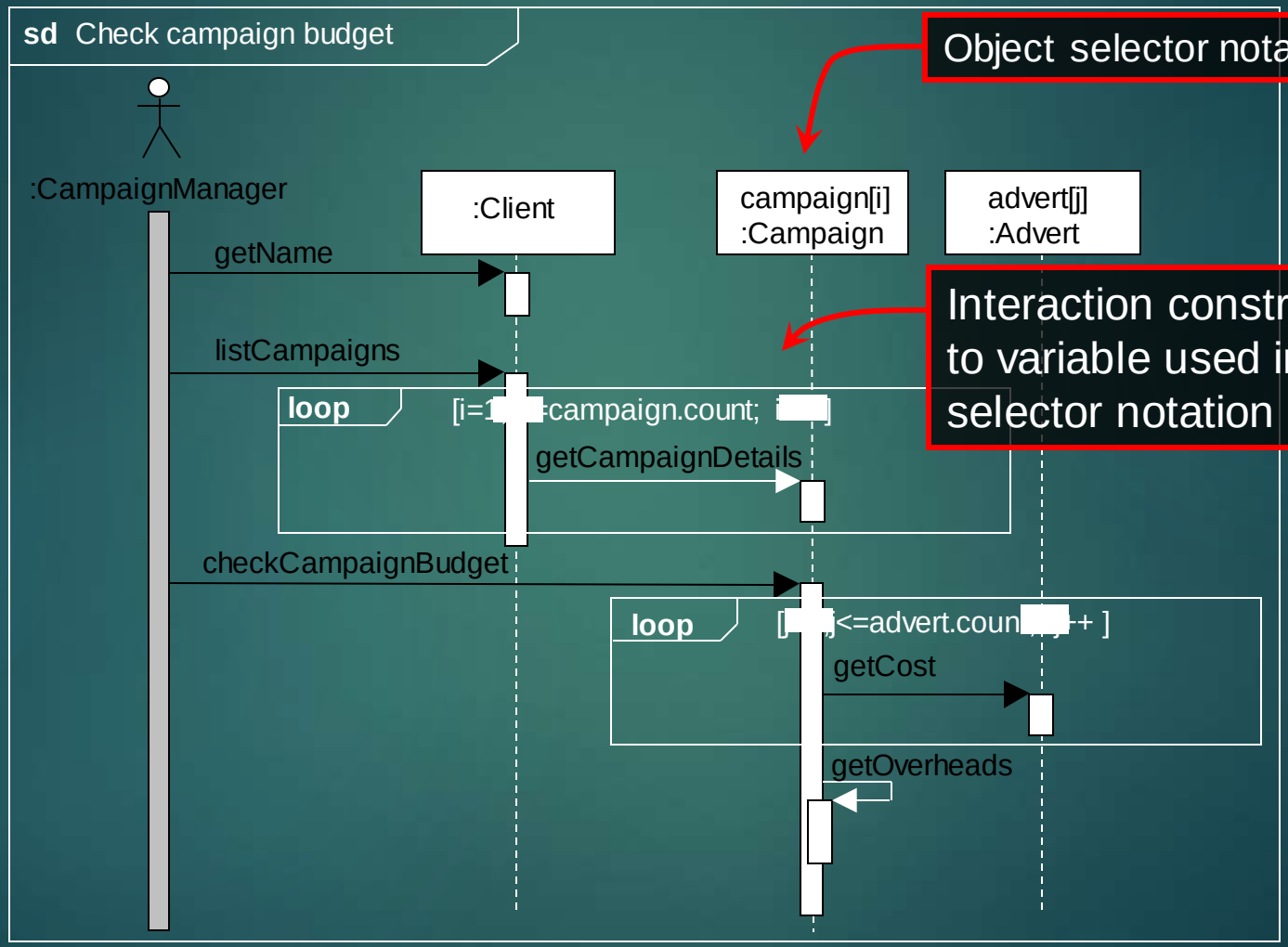
E-course

- A *reply message* returns the control to the object that originated the message that began the activation.

- Reply messages are shown with a dashed arrow, but it is optional to show them at all since it can be assumed that control is returned to the originating object at the end of the activation

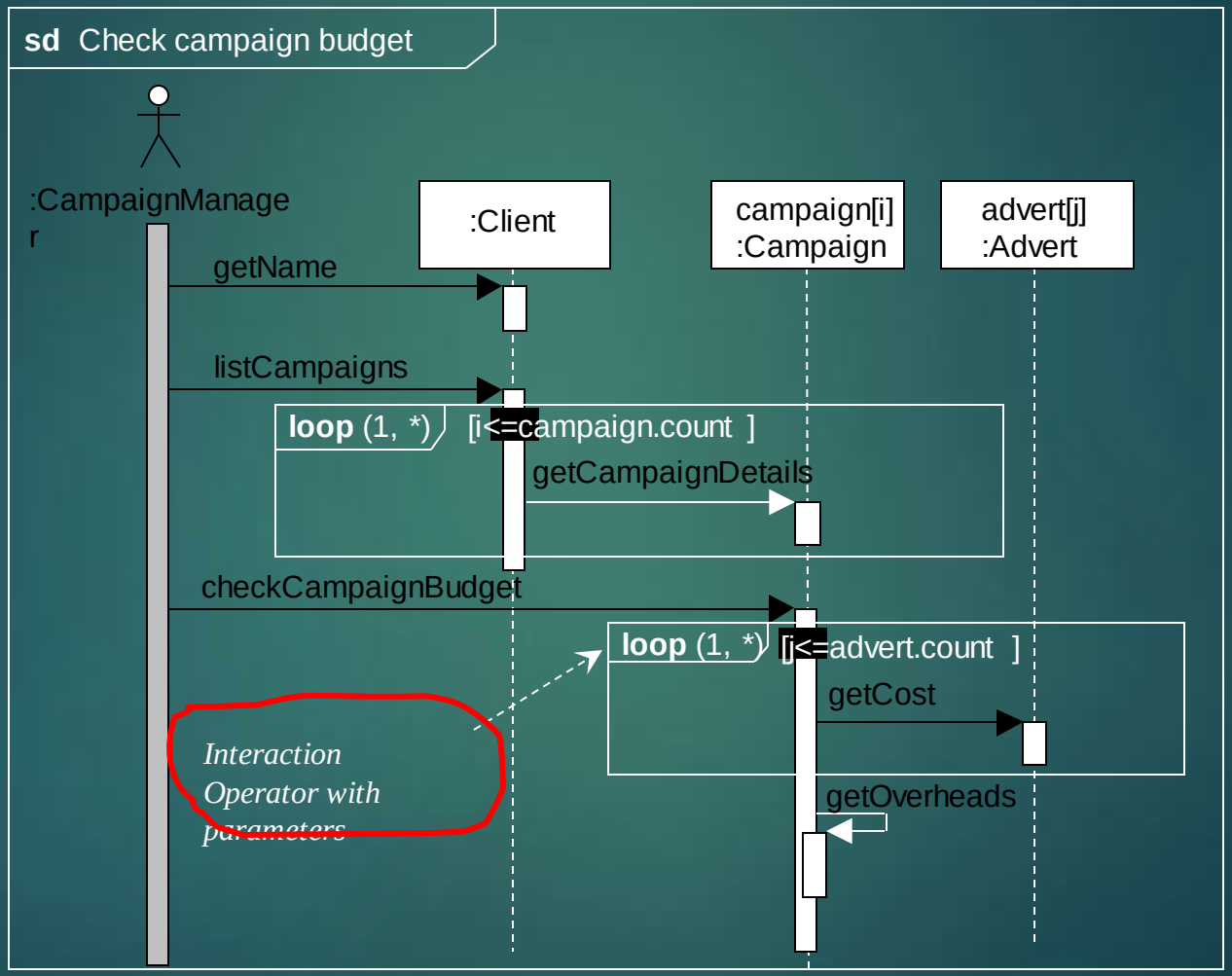
Object Selector Notation

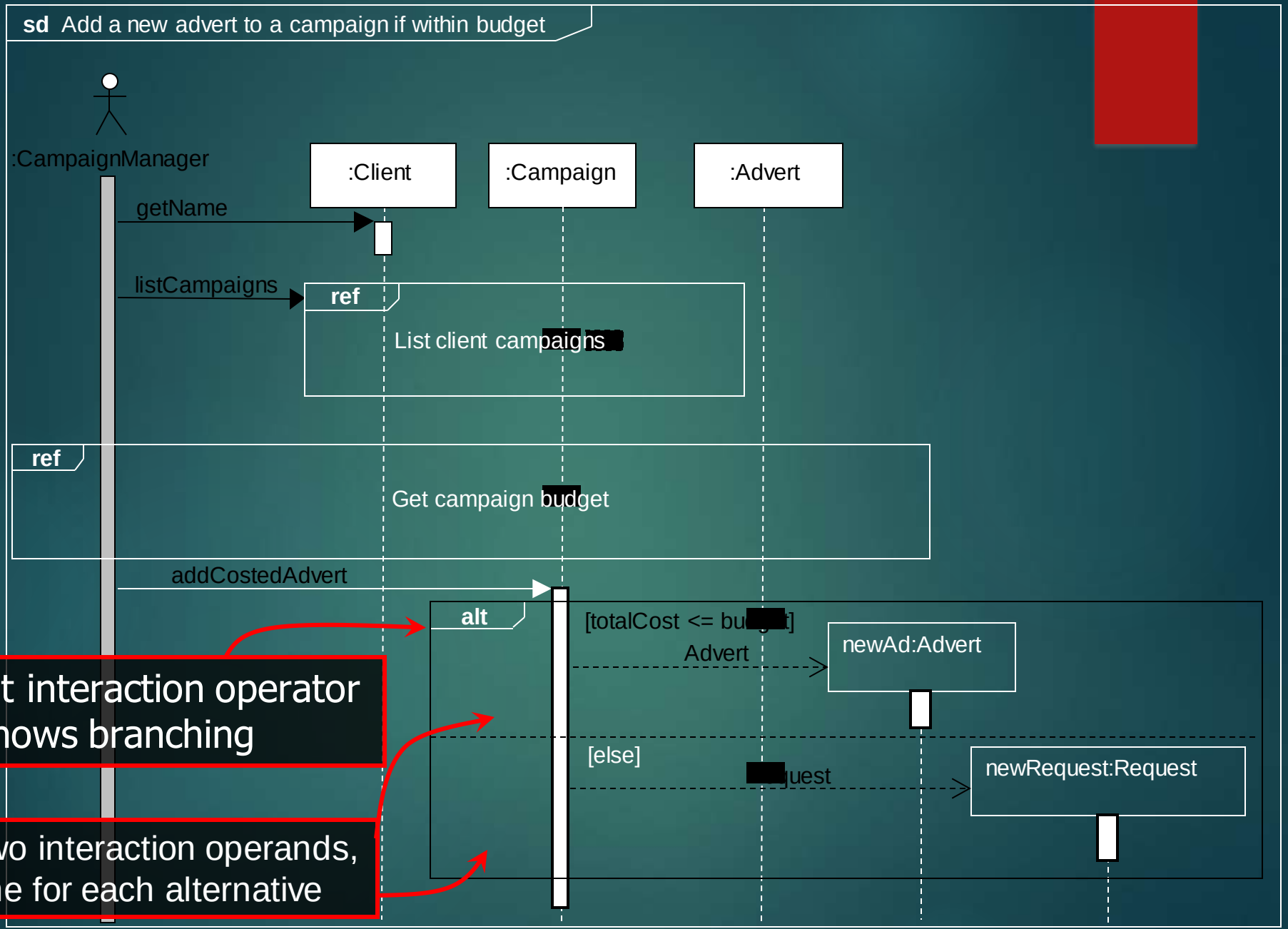
E-course



Interaction Operators

E-course





Handling Complexity

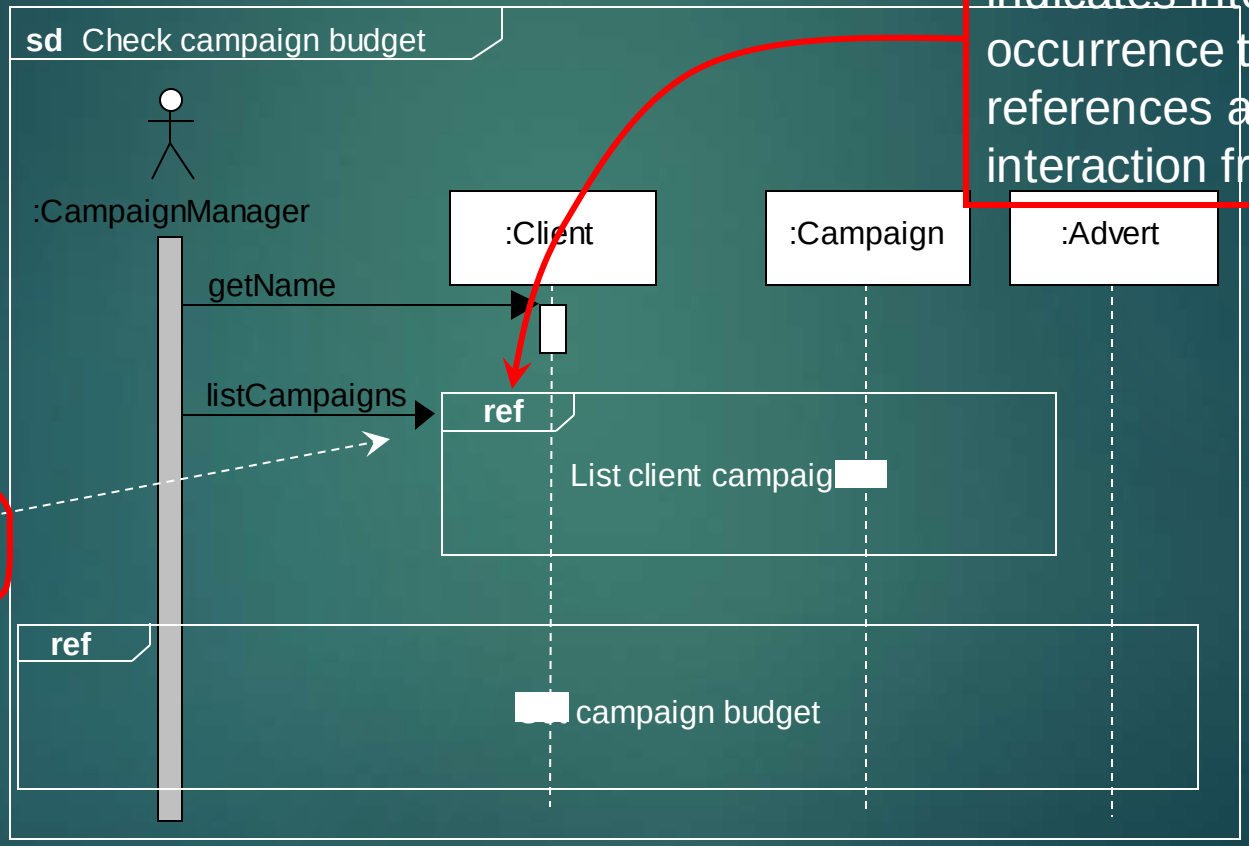
E-course

- Complex interactions can be modelled using various different techniques
  - Interaction fragments
  - Lifelines for subsystems or groups of objects
  - Continuations
  - Interaction Overview Diagrams (see later lecture)

Using Interaction Fragments

E-course

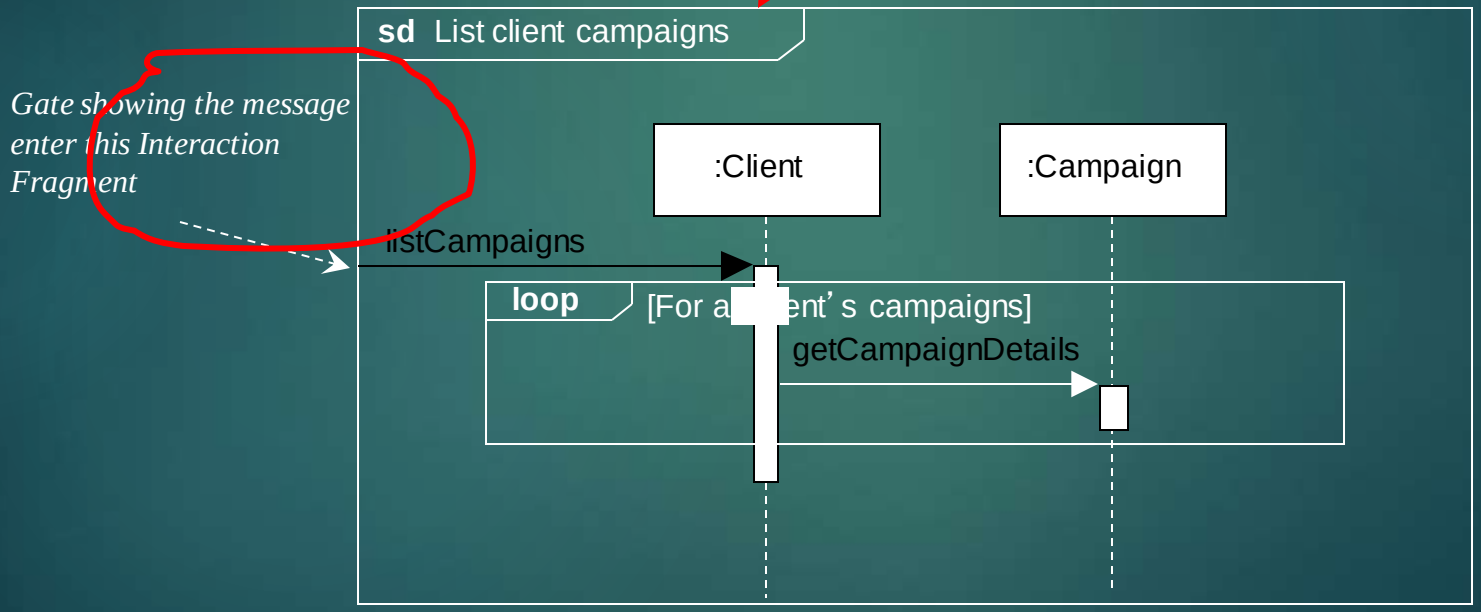
ref interaction operator indicates interaction occurrence that references an interaction fragment



Interaction Fragment

E-course

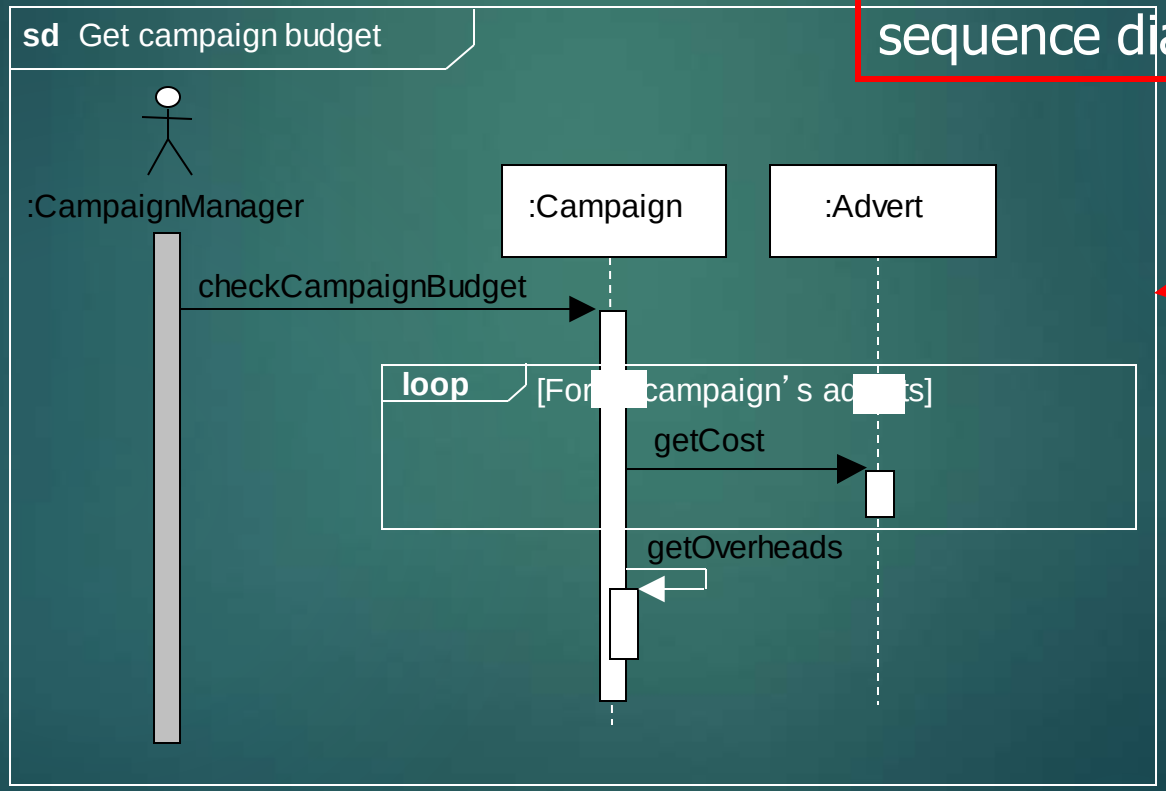
Interaction fragment that is referenced in `Check campaign budget` sequence diagram

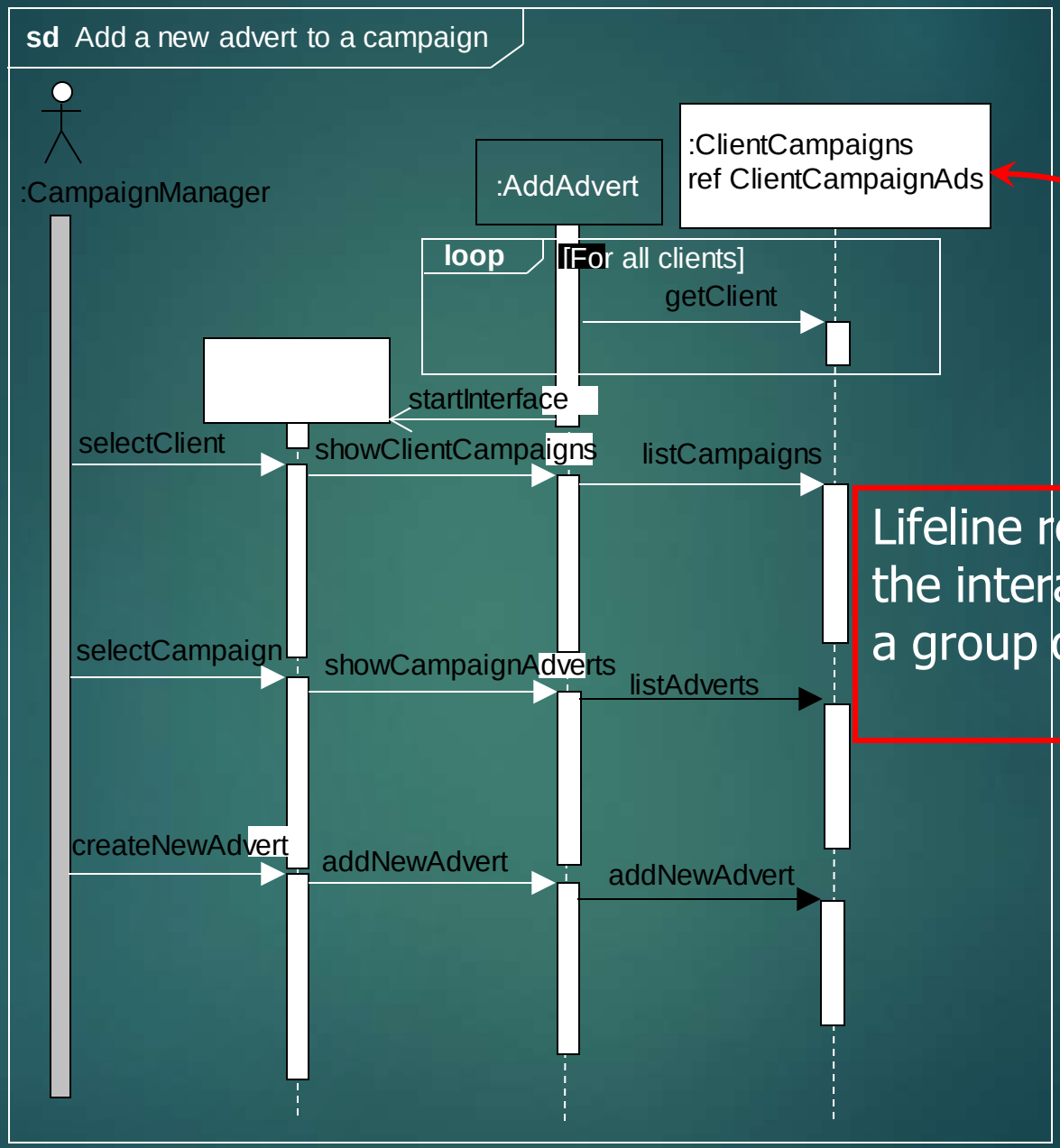


Interaction Fragment

E-course

Interaction fragment that is also referenced in `Check campaign budget` sequence diagram

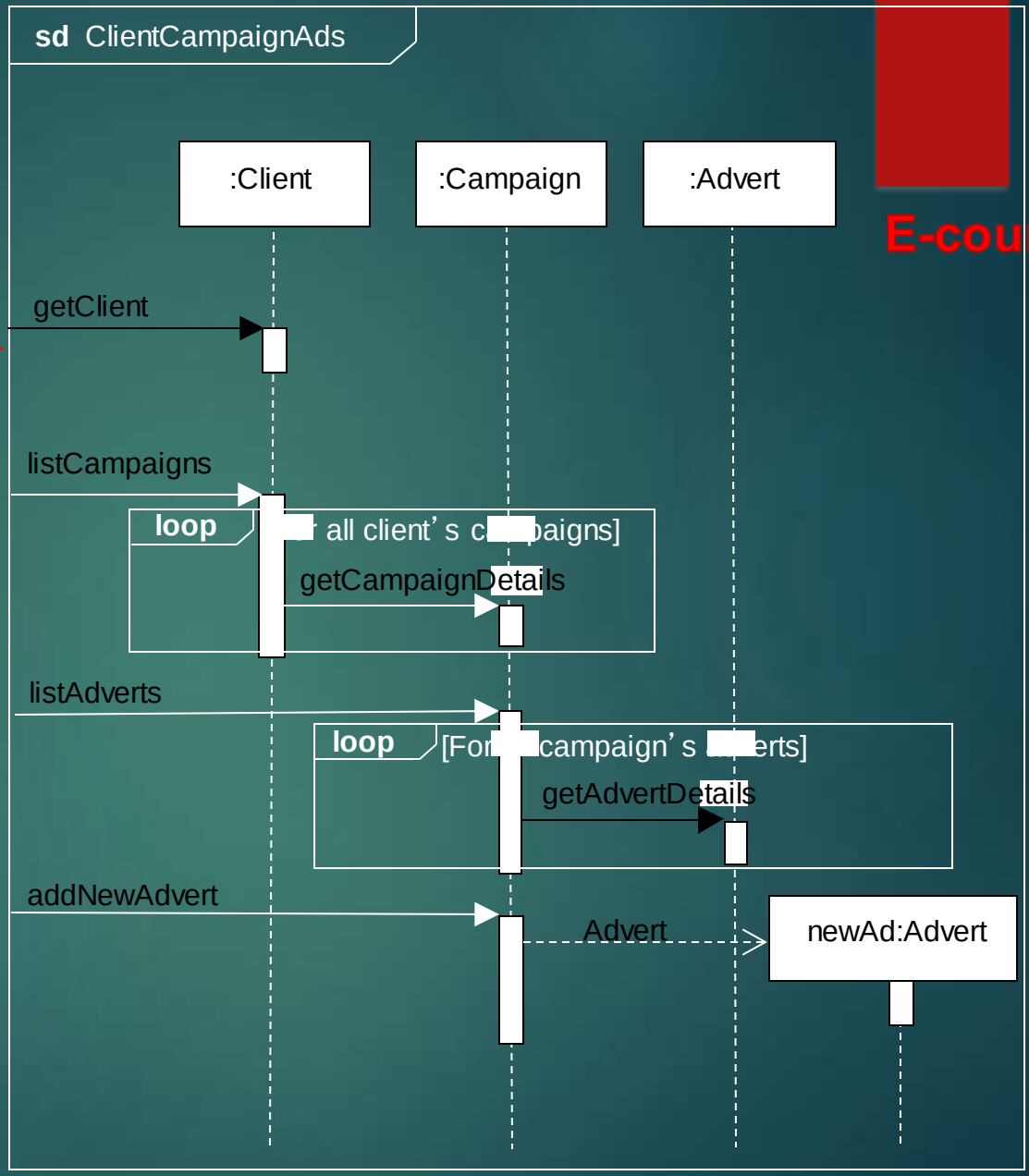




E-course

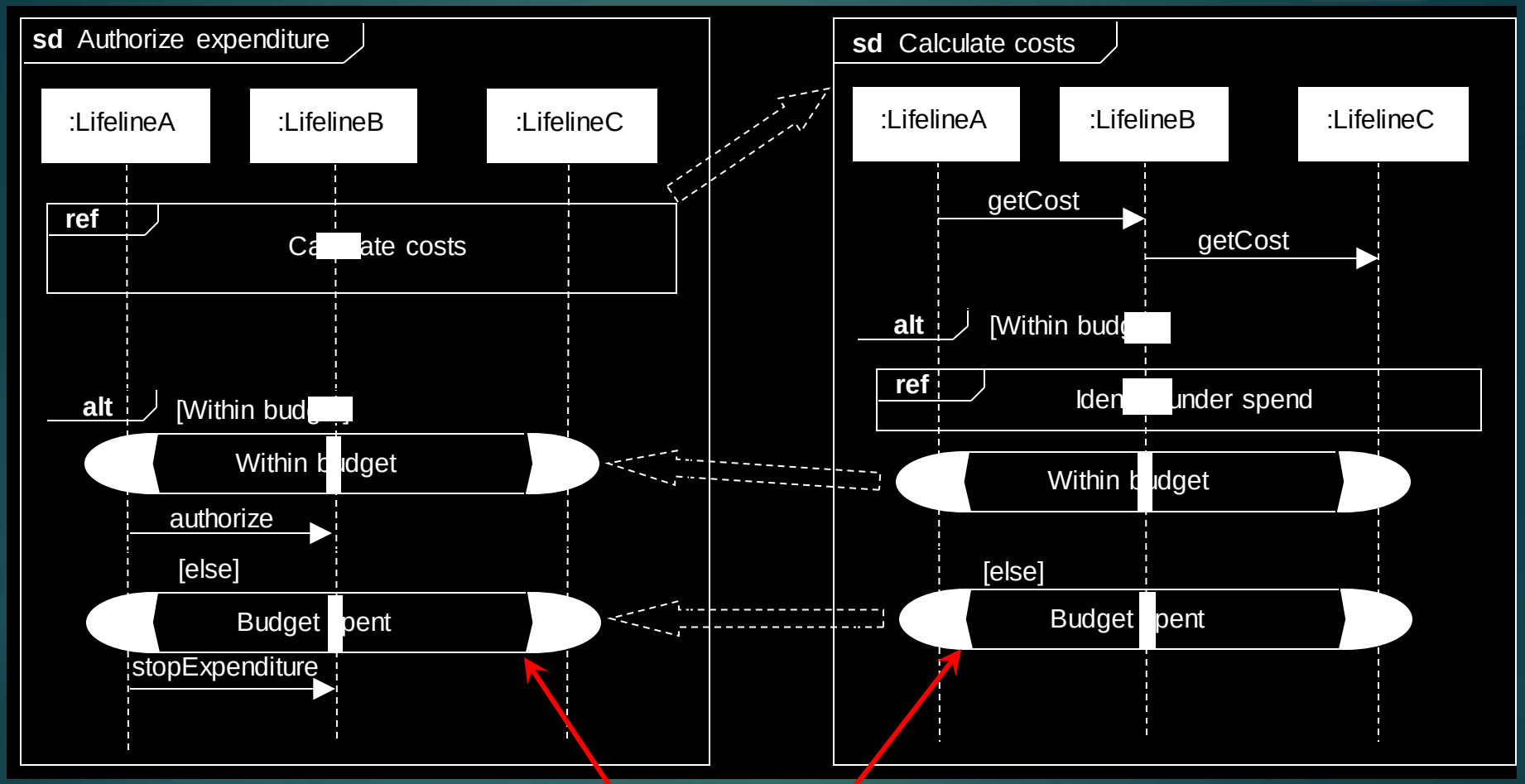
Lifeline representing the interaction between a group of objects

Sequence diagram referenced in the `Add a new advert to a compaign` sequence diagram



E-course

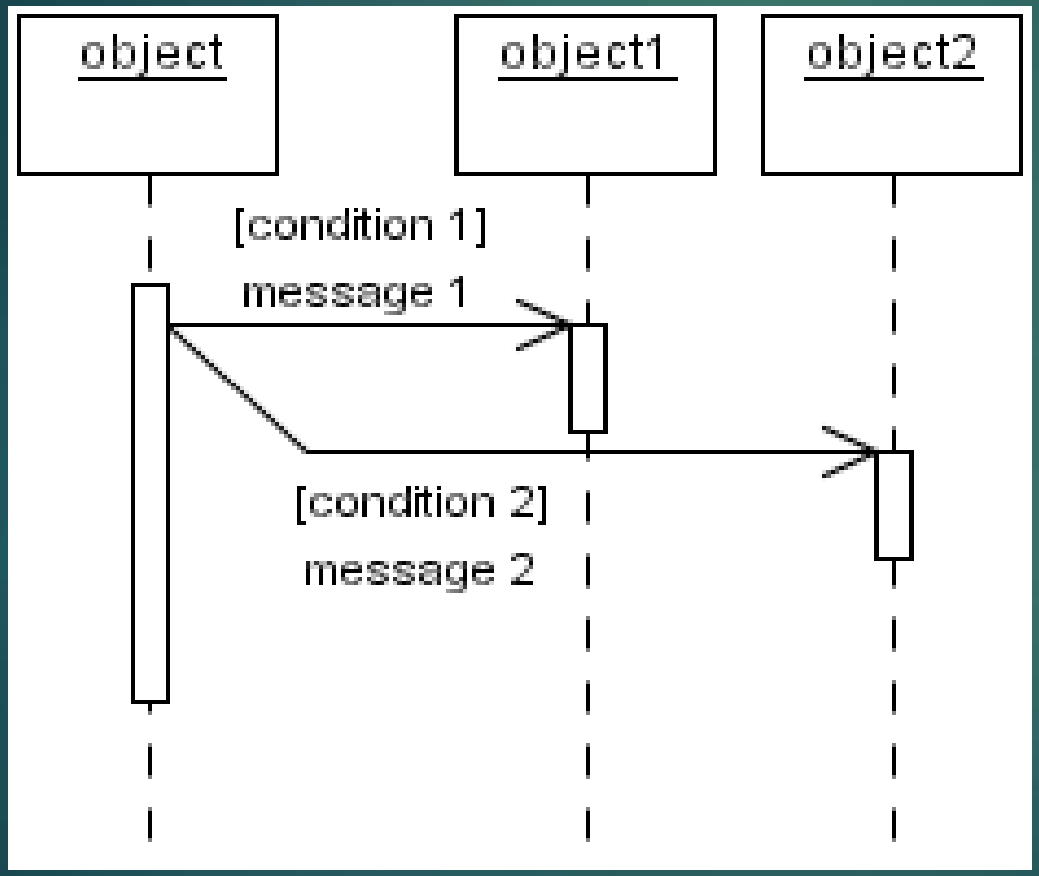
Using Continuations



Continuations are used to link sequence diagrams

Branching (in UML 1.x)

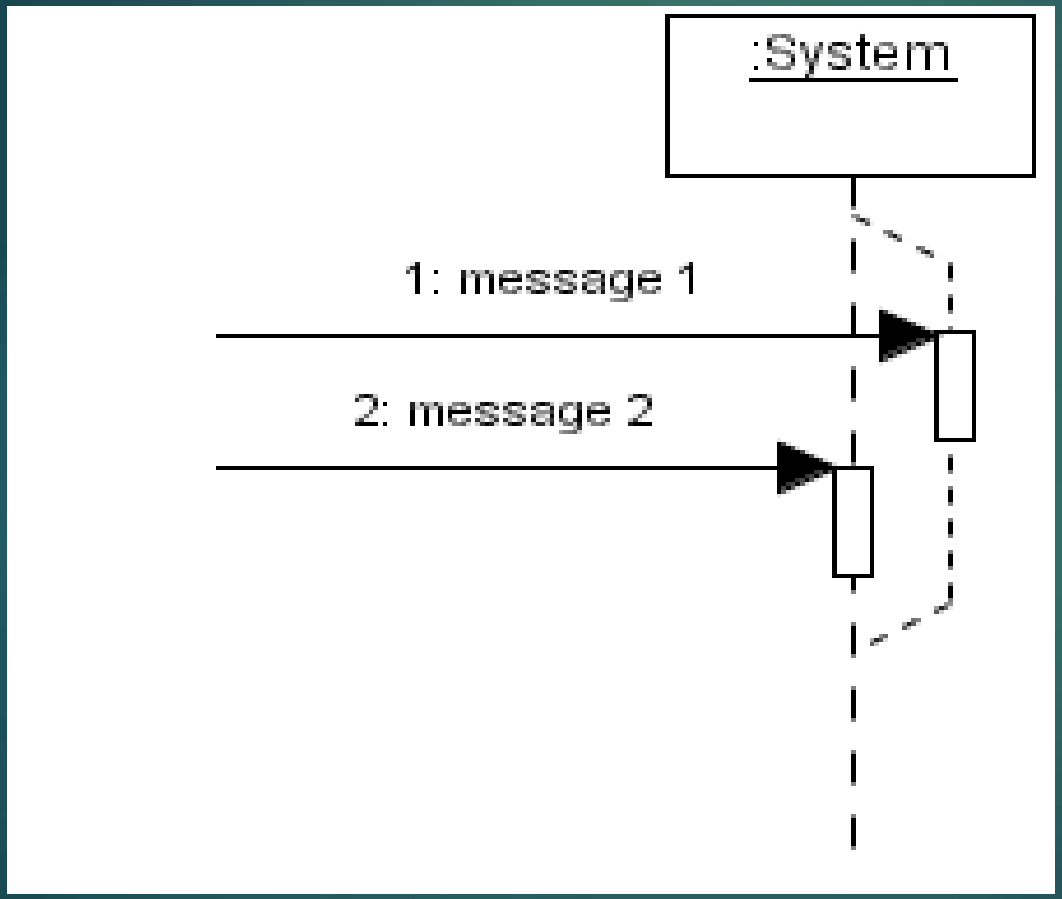
E-course



Conditional Message Transmission

Alternate Message Reception (in UML 1.x)

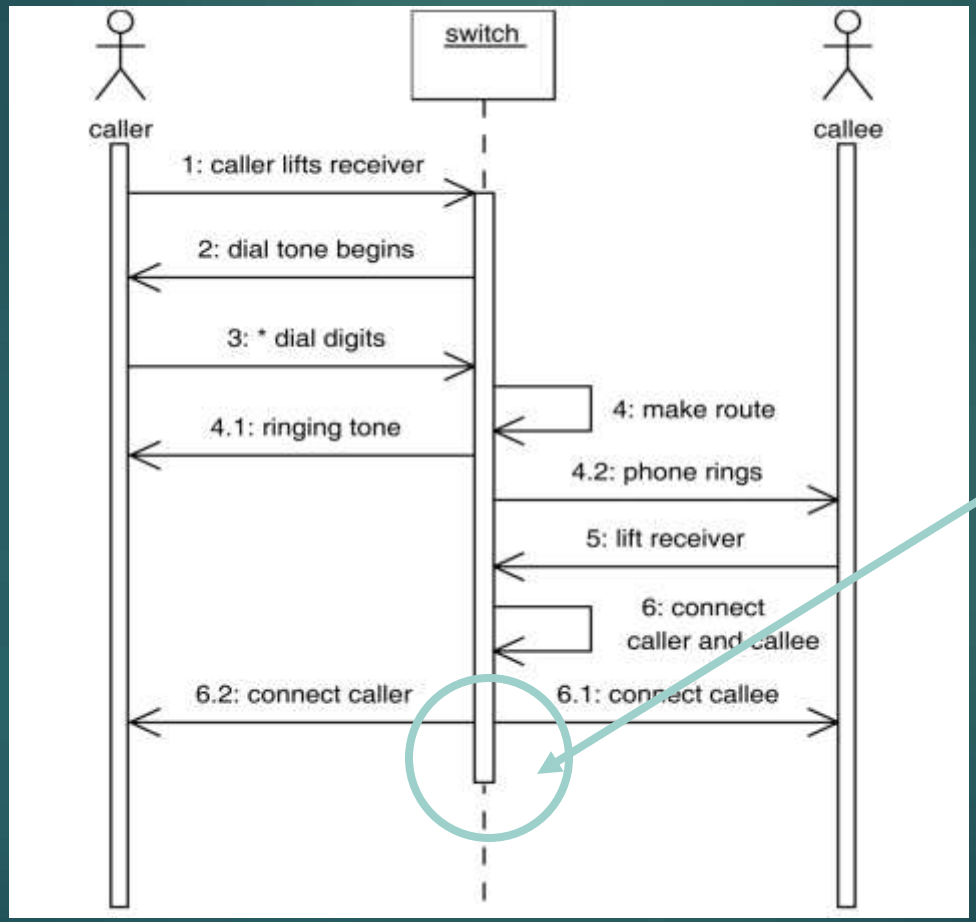
E-course



Alternate Message Reception

Example (in UML 1.x)

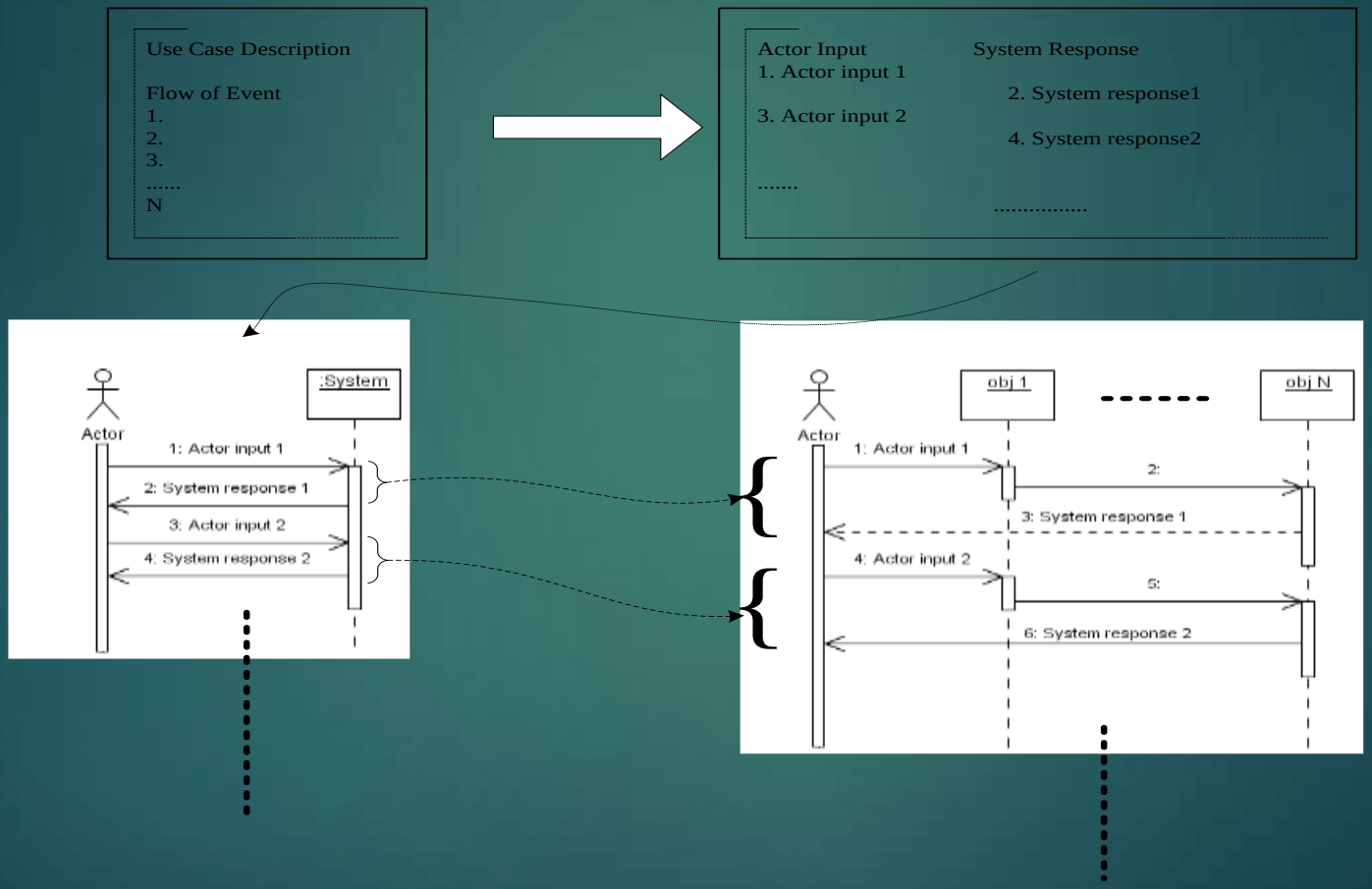
E-course



Concurrent Branch

Tricks and Tips (cont'd)

E-course



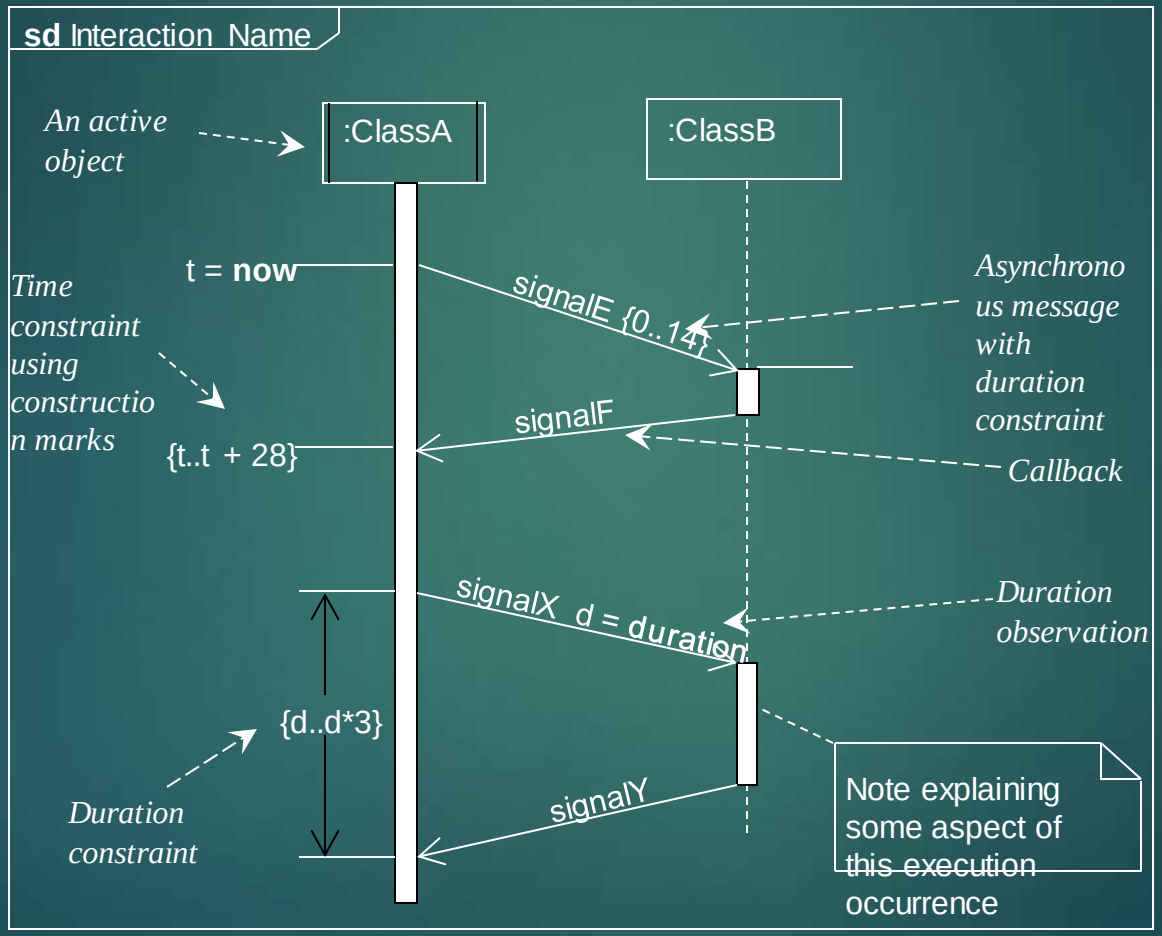
Asynchronous Message

E-course

- An *asynchronous message*, drawn with an open arrowhead, does not cause the invoking operation to halt execution while it awaits a return.

Further Notation

E-course



Interaction Operators

E-course

Interaction Operator	Explanation and use
alt	Alternatives represents alternative behaviours, each choice of behaviour being shown in a separate operand. The operand whose interaction constraint is evaluted as true executes.
opt	Option describes a single choice of operand that will only execute if its interaction contraint evaluates as true.
break	Break indicates that the combined fragment is performed instead of the remainder of the enclosing interaction fragment.
par	Parallel indicates that the execution operands in the combined fragment may be merged in any sequence once the event sequence in each operand is preserved.
seq	Weak Sequencing results in the ordering of each operand being maintained but event occurrence from different operands on different lifelines may occur in any order. The order of event occurrences on common operands is the same as the order of the operands.
strict	Strict Sequencing imposes a strict sequence on execution of the operands but does not apply to nested fragments.
neg	Negative describes an operand that is invalid.
critical	Critical Region imposes a constraint on the operand that none of its event occurrences on the lifelines in the region can be interleaved.
ignore	Ignore indicates the message types, specified as parameters, that should be ignored in the interaction.
consider	Consider states which messages should be consider in the interaction. This is equivalent to stating that all others should be ignored.
assert	Assertion states that the sequence of messaging in the operand is the only valid continuation.
loop	Loop is used to indicate an operand that is repeated a number times until the interaction constraint for the loop is no longer true.

Guidelines for Sequence Diagrams

E-course

1. Decide at what level you are modelling the interaction.
2. Identify the main elements involved in the interaction.
3. Consider the alternative scenarios that may be needed.
4. Identify the main elements involved in the interaction.

Guidelines for Sequence Diagrams

E-course

5. Draw the outline structure of the diagram.

6. Add the detailed interaction.

7. Check for consistency with linked sequence diagrams and modify as necessary.

8. Check for consistency with other UML diagrams or models.

Model Consistency

E-course

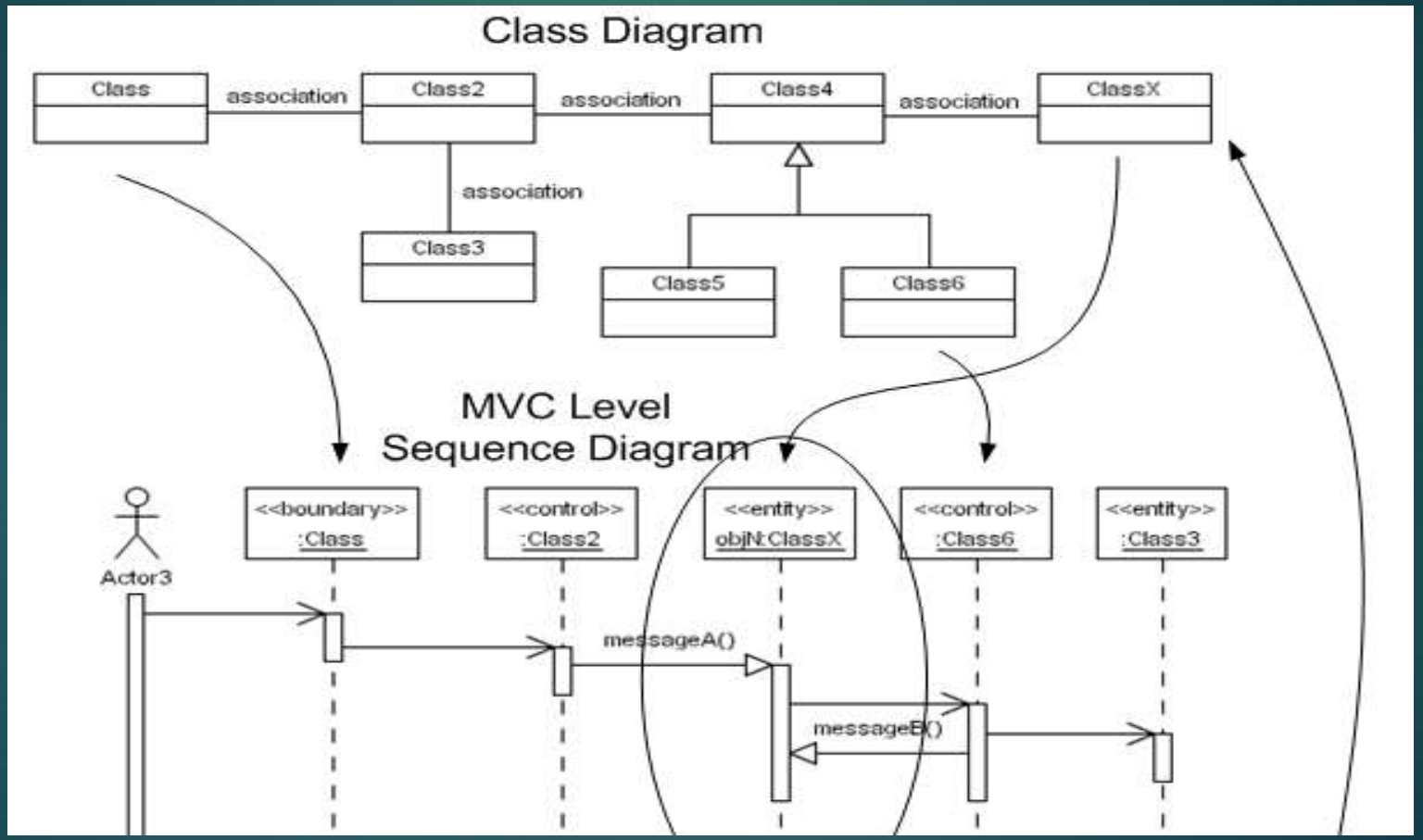
- The allocation of operations to objects must be consistent with the class diagram and the message signature must match that of the operation.
  - Can be enforced through CASE tools.
- Every sending object must have the object reference for the destination object.
  - Either an association exists between the classes or another object passes the reference to the sender.
  - This issue is key in determining association design (See Ch. 14).
  - Message pathways should be carefully analysed.

Model Consistency

E-course

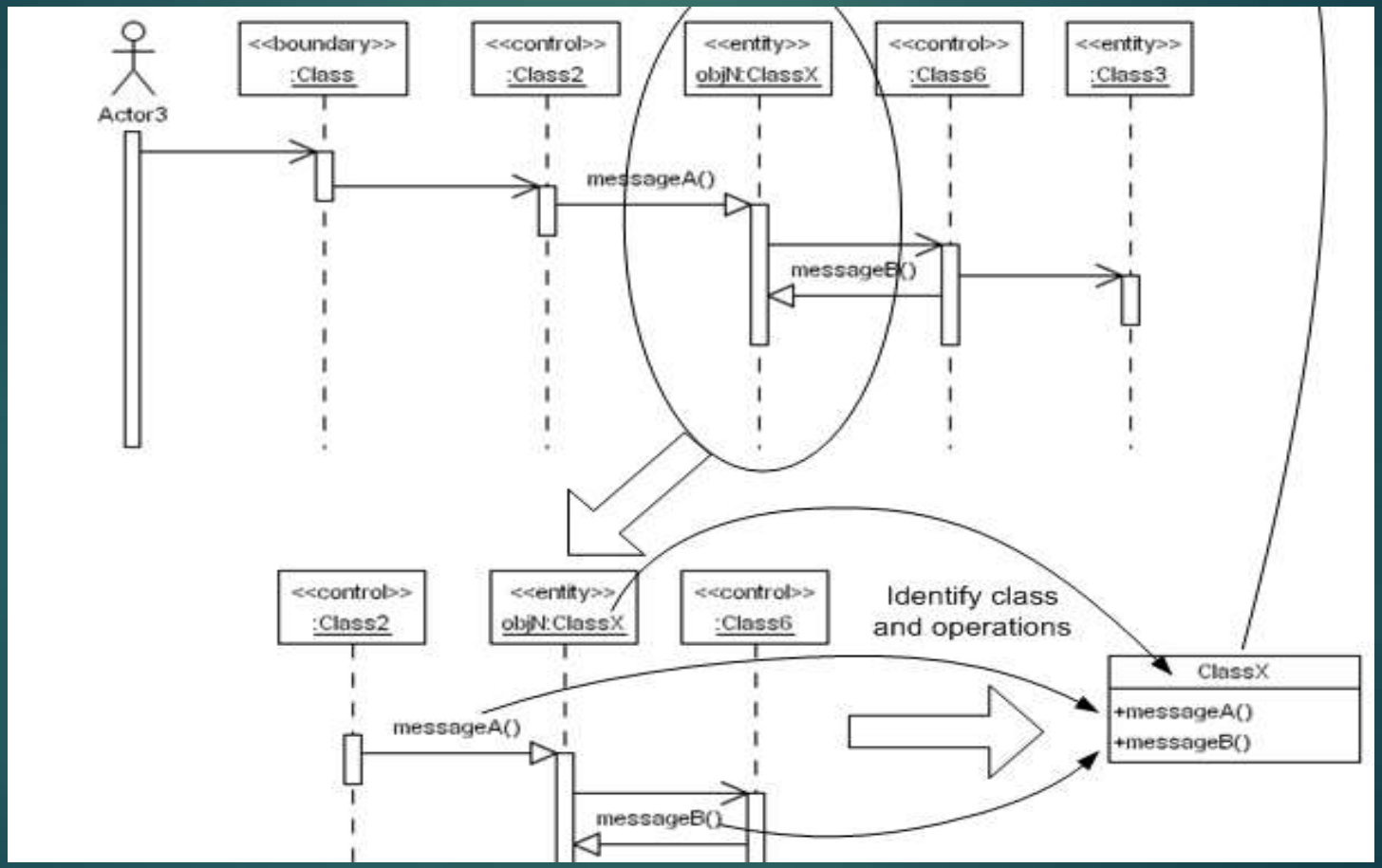
- All forms of interaction diagrams used should be consistent.
- Messages on interaction diagrams must be consistent with the state machine for the participating objects.
- Implicit state changes in interactions diagrams must be consistent with those explicitly modelled in state machine.

Tricks and Tips (cont'd)



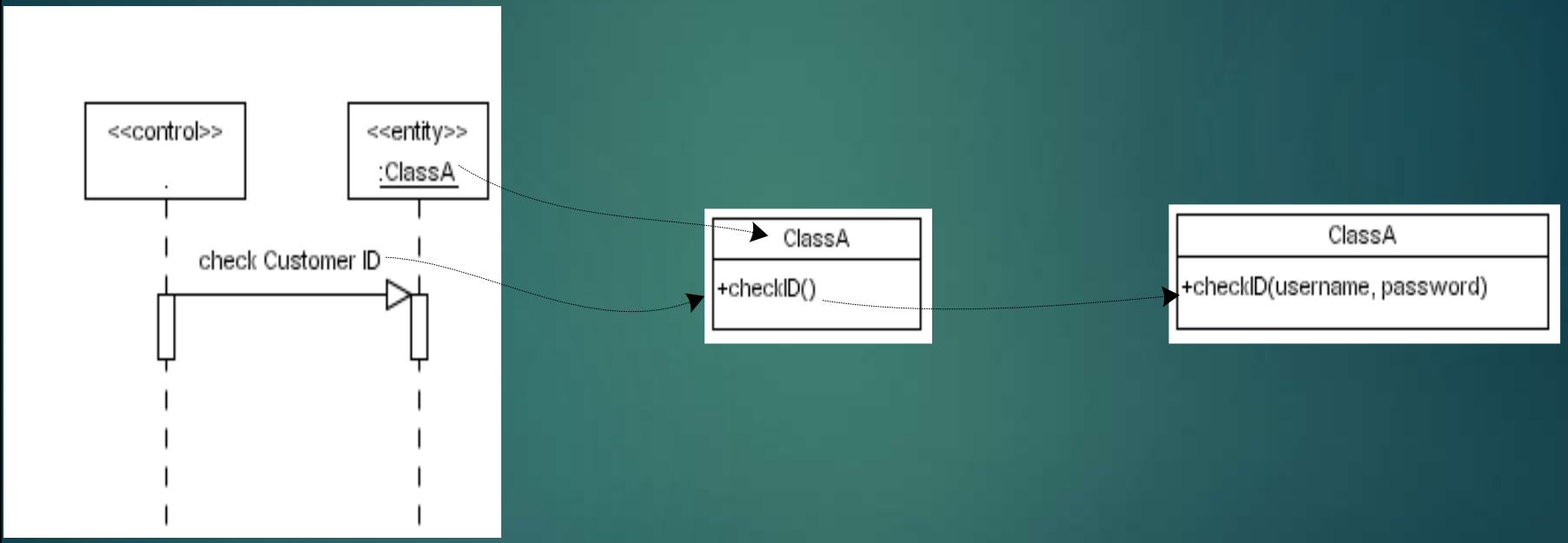
Tricks and Tips (cont'd)

E-course



Tricks and Tips (cont'd)

E-course



References

E-course

- UML Reference Manual (OMG, 2004)
- Bennett, Skelton and Lunn (2005)

(For full bibliographic details, see Bennett, McRobb and Farmer)

Object-Oriented Technology - From Diagram to Code with Visual Paradigm for UML, Curtis H.K. Tsang, Clarence S.W. Lau and Y.K. Leung, McGraw-Hill Education (Asia), 2005

WE FOCUS ON KNOWLEDGE-BASED ON EDUCATION

KSRA of Empowerment is a global non-profit organization committed to bringing empowerment through education by utilizing innovative mobile technology and educational research from experts and scientists. KSRA emerged in 2012 as a catalytic force to reach the hard to reach populations worldwide through Learning management system & E-learning & mobile learning.

The KSRA team partners with local under-served communities around the world to improve the access to and quality of knowledge based on education, amplify and augment learning programs where they exist, and create new opportunities for e-learning where traditional education systems are lacking or non-existent.

E-course