WIA2002: Software Modelling

Semester 1, Session 2016/17

Lecture 9: Modelling Object Interactions - UML Interaction Diagrams

(Part 2: Communication, Interaction Overview and Timing Diagrams)

Learning Objectives

- Know how to model object interaction using an interaction communication diagram.
- Know how to model interactions using interaction overview diagrams.
- Know how to model interaction using an interaction sequence diagram.
- Know how to use timing diagrams.

2. COMMUNICATION DIAGRAMS

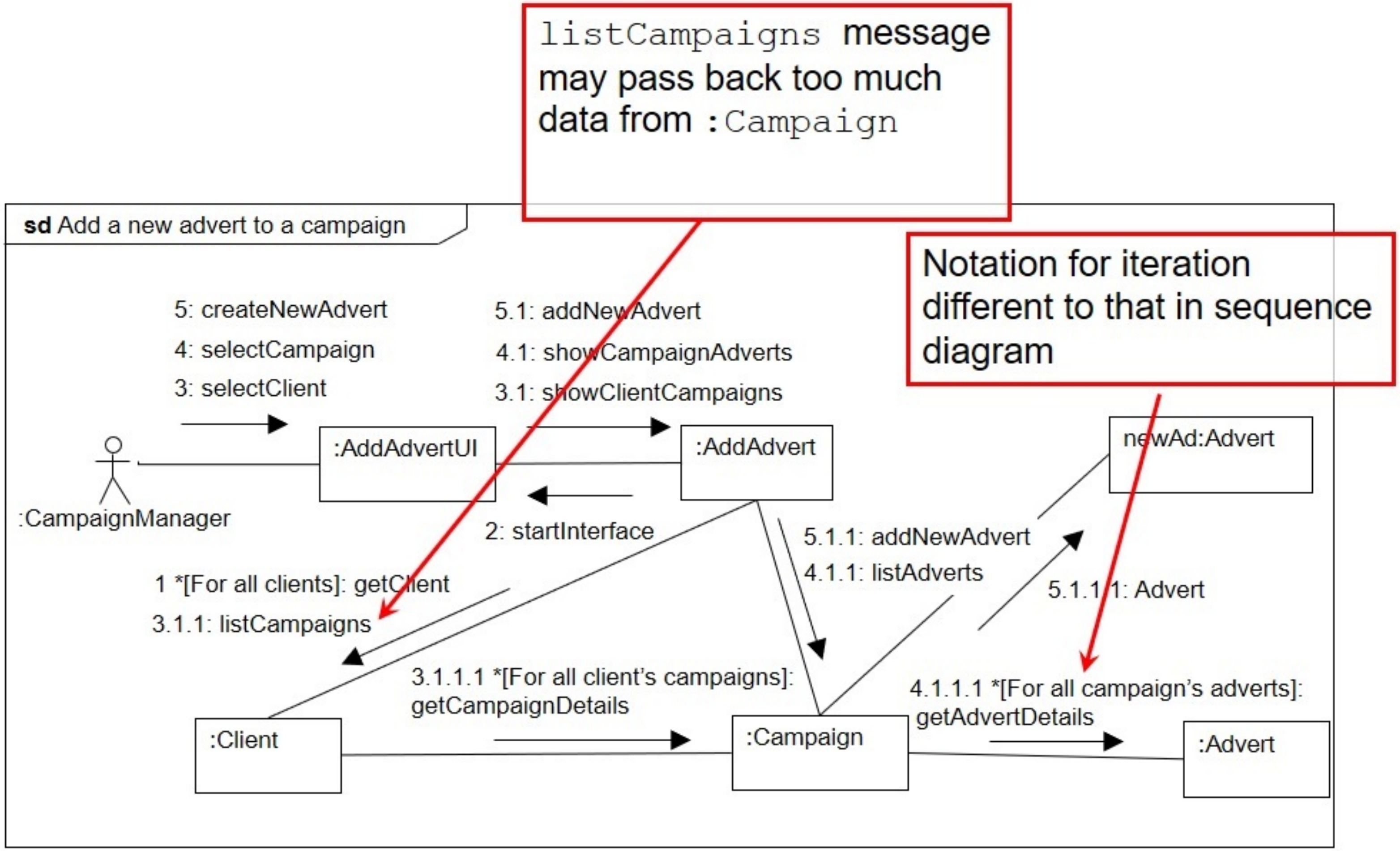
Communication Diagrams

- Hold the same information as sequence diagrams.
- Show links between objects that participate in the collaboration.
- No time dimension, sequence is captured with sequence numbers.
- Sequence numbers are written in a nested style (for example, 3.1 and 3.1.1) to indicate the nesting of control within the interaction that is being modelled.
- Preferred by some developers as they offer a view of object interaction that is easy to relate to the underlying collaboration .
- The visibility of links between the lifelines.

Communication Diagrams

- However, they do not provide the same level of syntax as sequence diagrams and are not suitable for complex interaction.
- Probably useful during analysis activities while sequence diagrams are better at representing design detail.
- Communication diagrams are used to describe analysis use case realizations because typically the messages are not fully specified at this stage. However, when there are many messages between two objects then a collaboration diagram is more difficult to read than the equivalent sequence diagram.

Notations of Communication Diagrams

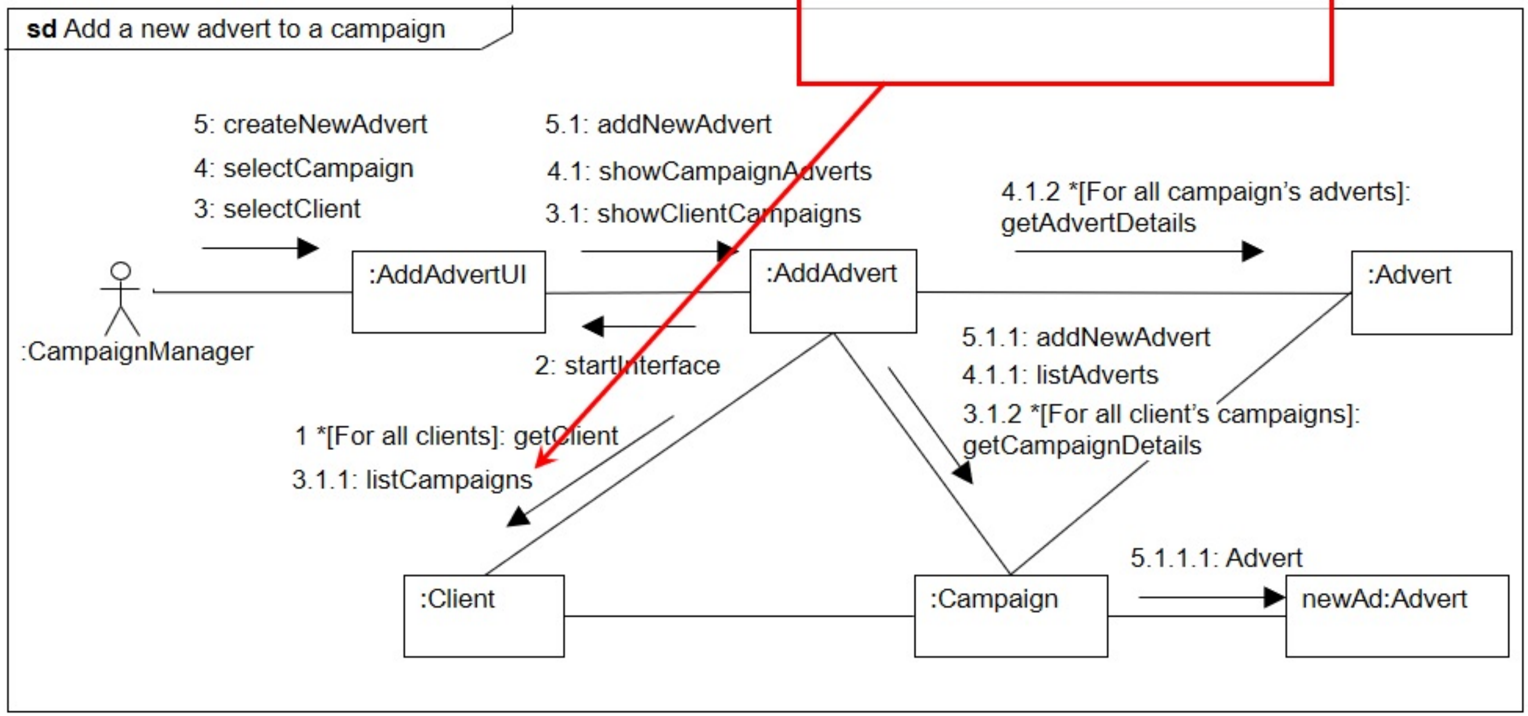


`listCampaigns` message may pass back too much data from `:Campaign`

Notation for iteration different to that in sequence diagram

Alternative Interaction

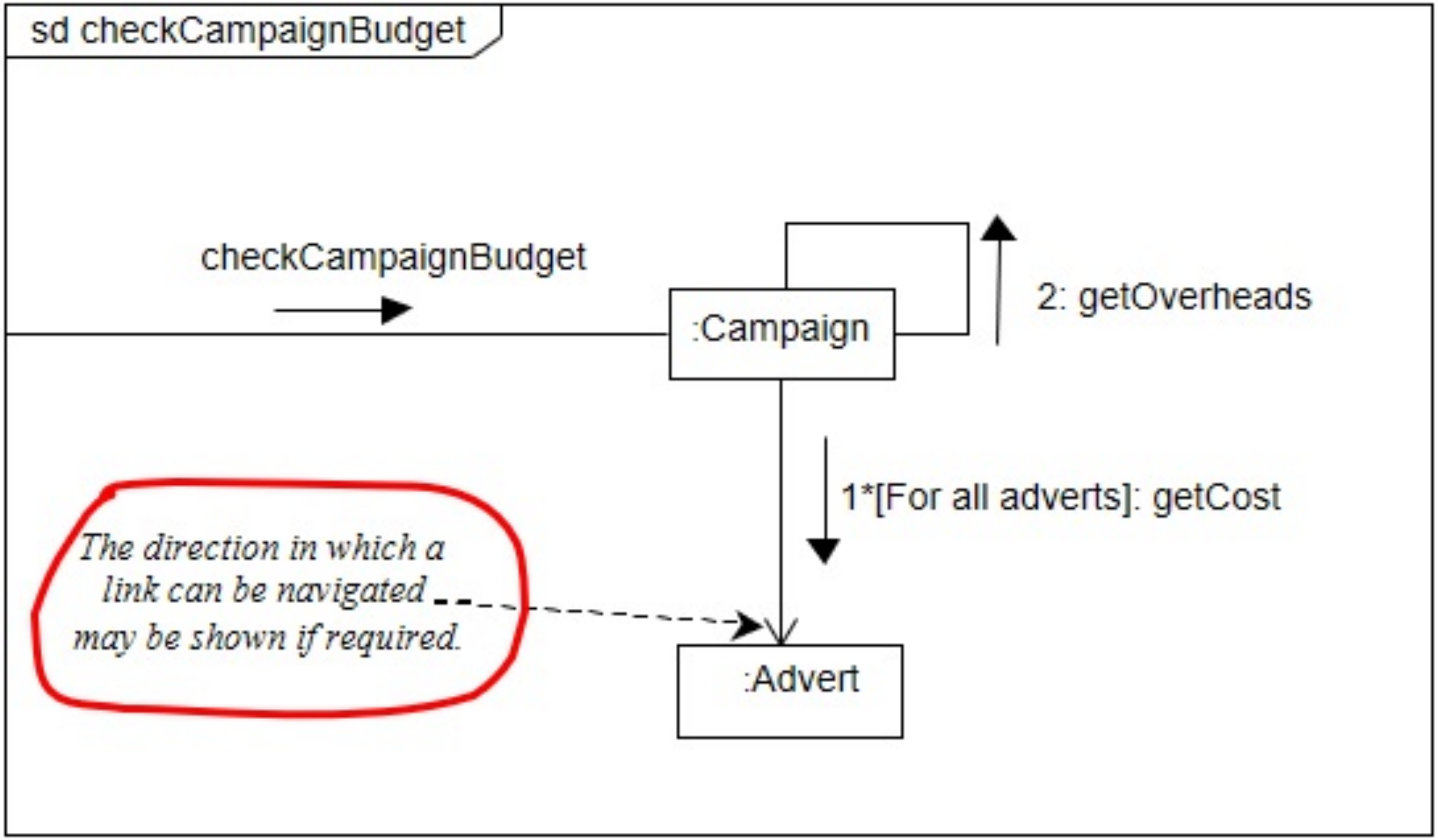
`listCampaigns` message does not pass back any data from `:Campaign` in this alternative interaction



Message Labels

Type of message	Syntax example
1. Simple message.	`4: addNewAdvert`
2. Nested call with return value. *The return value is placed in the variable* `name`.	`3.1.2: name = getName`
3. Conditional message. *This message is only sent if the condition* `[balance > 0]` *is true.*	`5 [balance > 0]: debit(amount)`
4. Iteration	`4.1 *[For all adverts]: getCost`

Navigating Links



The direction in which a link can be navigated may be shown if required.

3. INTERACTION OVERVIEW DIAGRAMS

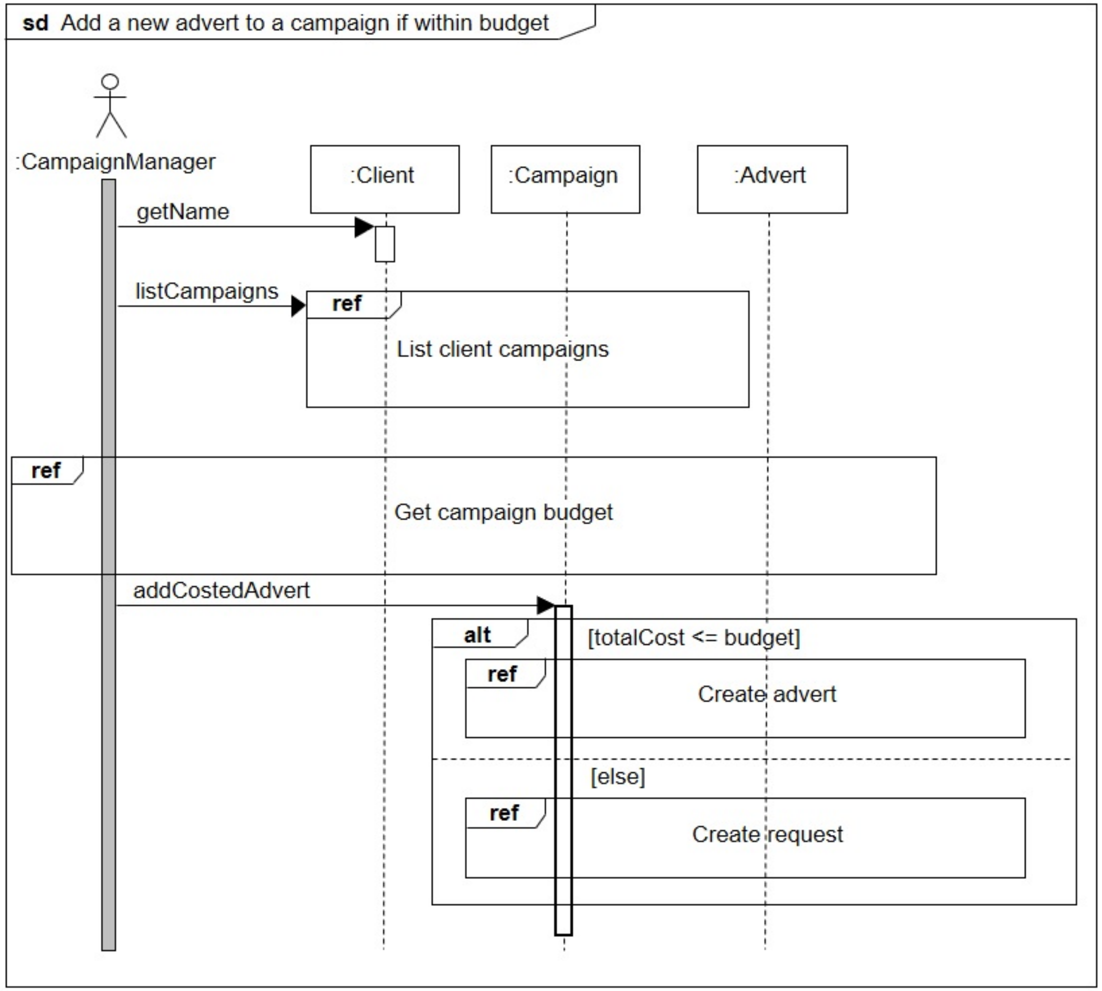
Interaction Overview Diagrams

- *Variants of activity diagrams* and incorporate interaction diagrams.
  - Syntax for activity diagrams is used, including decision and merge nodes
- Focuses on the *flow of control* in an interaction.
- Nodes in the diagram may be interactions or interaction occurrences.
- The detailed messaging of the interaction is *hidden* in the message.
- Interaction needs to be broken down into its key elements.

Drawing Interaction Overview Diagrams

- An alternative version of the sequence diagram **Add a new advert to a campaign if within budget** is shown on the next slide and is used to develop an interaction overview diagram

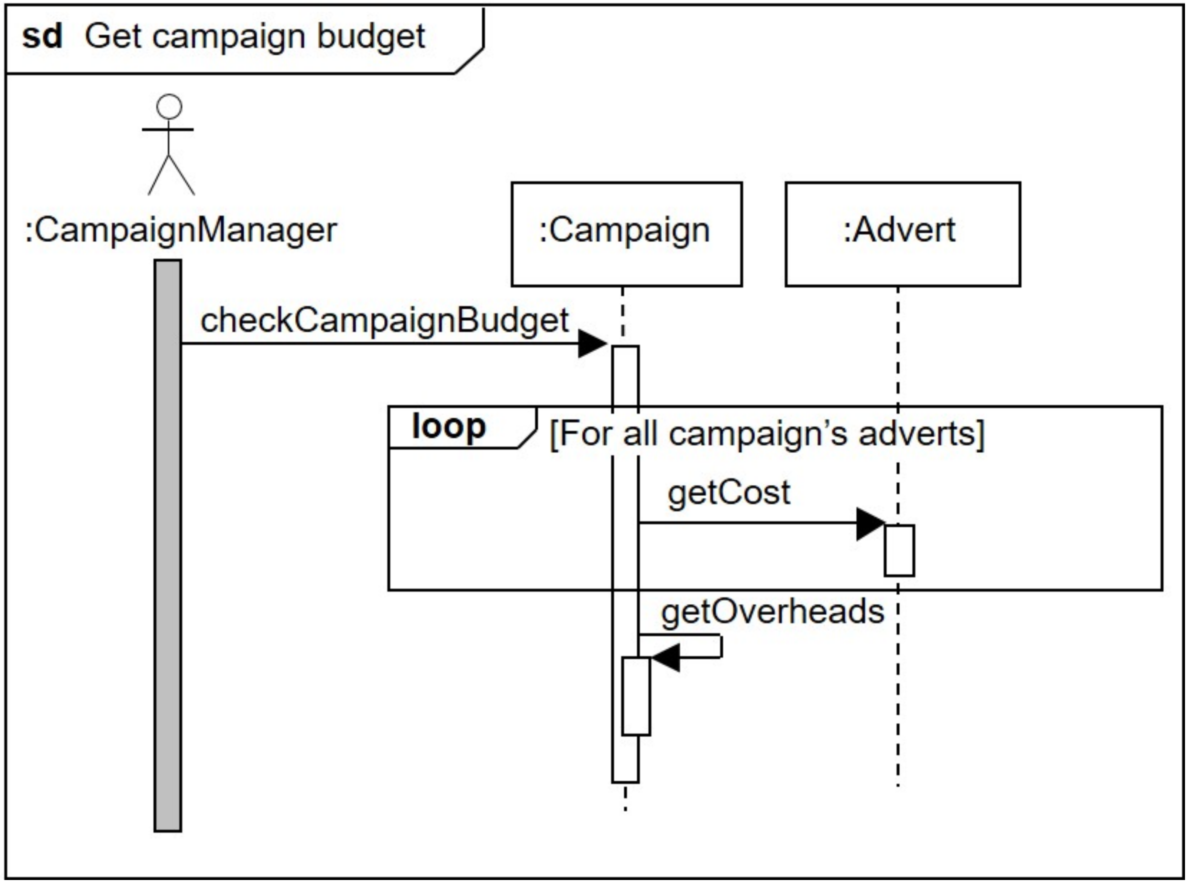
Drawing Interaction Overview Diagrams: Sequence Diagram



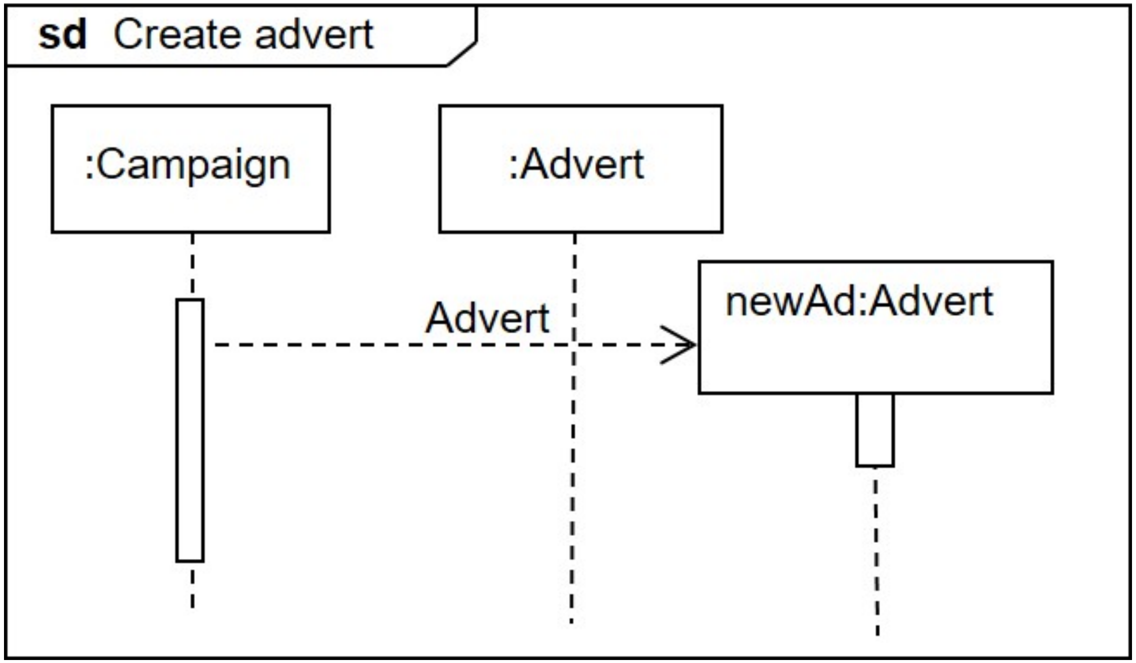
Drawing Interaction Overview Diagrams

- The sequence diagram is broken down to some extent by referring to two interaction fragments
  - List client campaigns
  - Get campaign budget
- The **alt** combined fragment could be broken down further to two more interaction fragments to show the flow of control:
  - Create advert
  - Create request

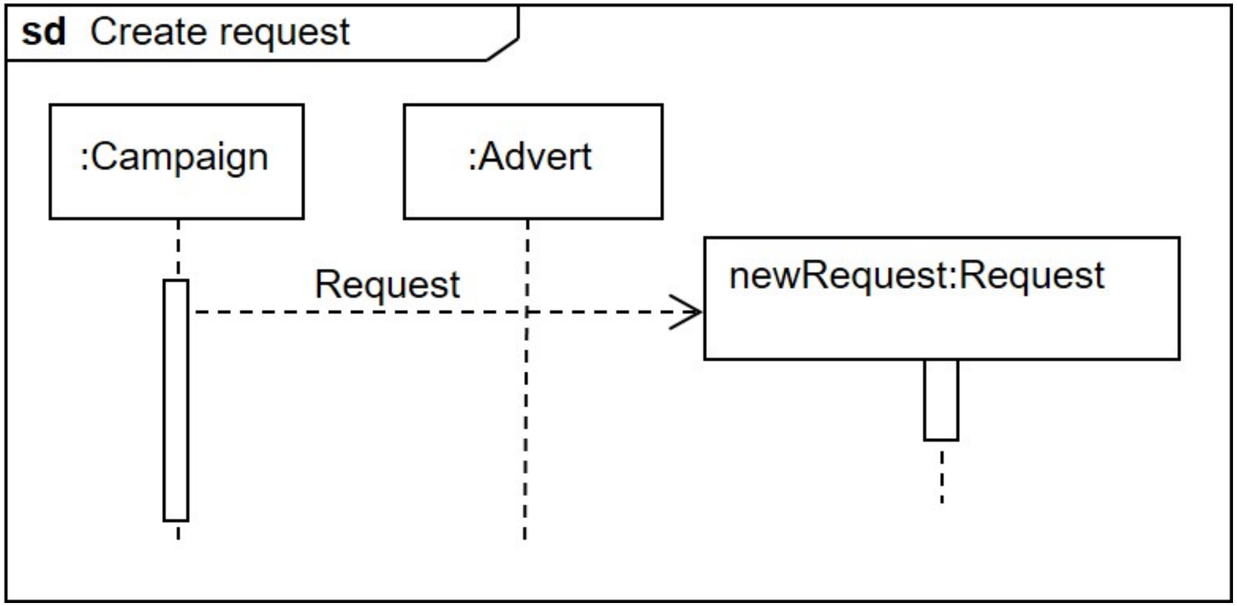
Interaction Fragment Used: Get campaign budget



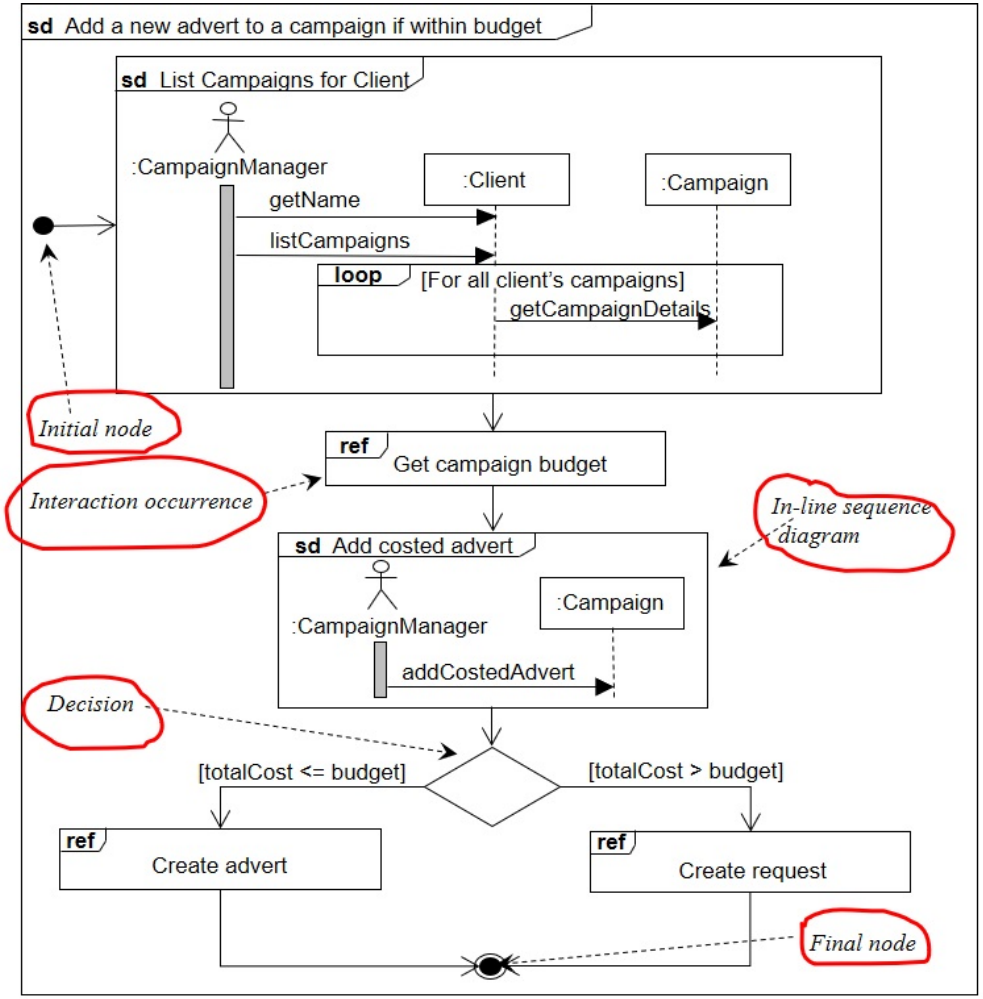
Interaction Fragment Used: Create advert



Interaction Fragment Used: Create request



Notation of Interaction Overview Diagram



Benefits of Interaction Overview Diagrams

- Useful when describing a *complex* interaction
- Particularly when it comprises a series of interaction fragments
- Provide a useful notation to describe *high-level* system interactions

4. TIMING DIAGRAMS

Timing Diagram

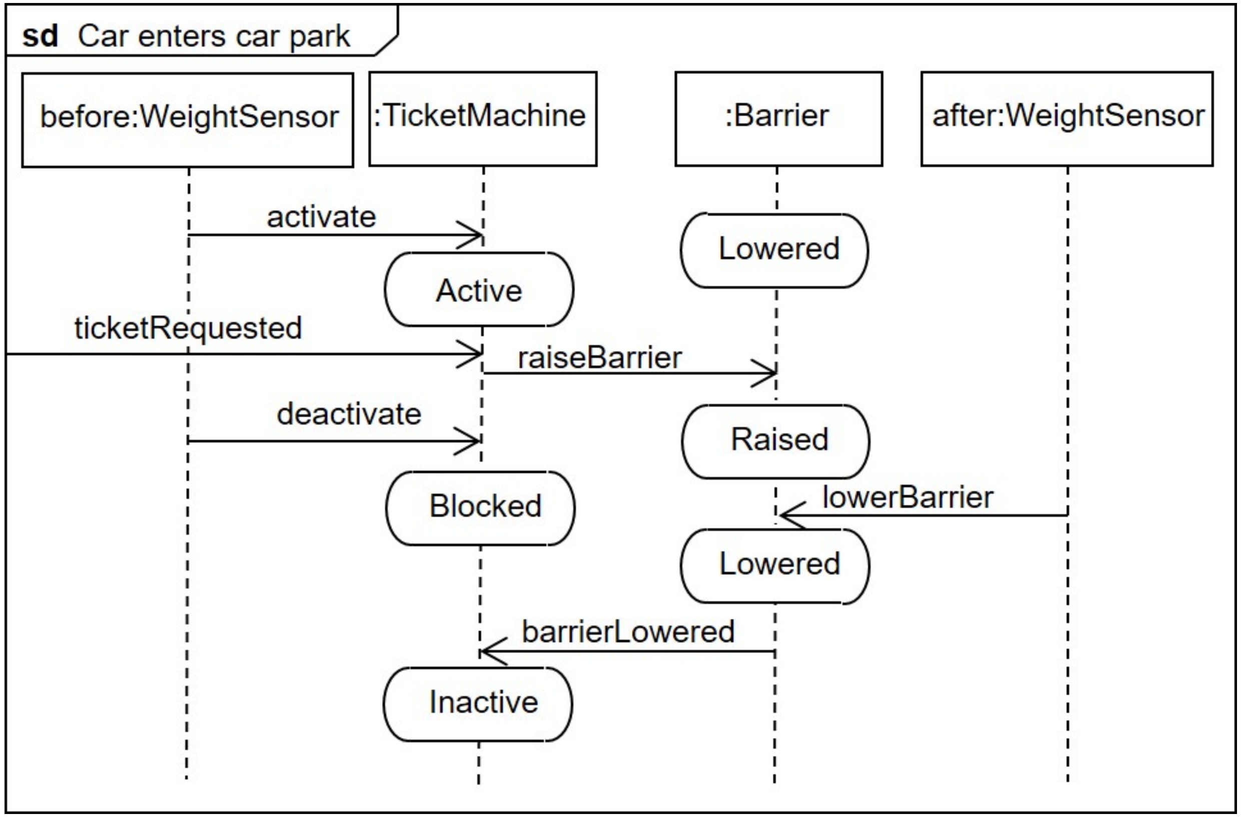
- A new feature in UML 2.0.
- Show how time constraints affect interactions between lifelines.
- It is useful to build the state machines for the key lifelines at the same time.
    - State machines will be discussed in the next lecture.

Timing Diagram

- Timing diagram may be prepared at different levels of detail.
- They include the state changes for one or more lifelines
  - Typically those with important time dependent behaviour are modelled.

Example Sequence Diagram

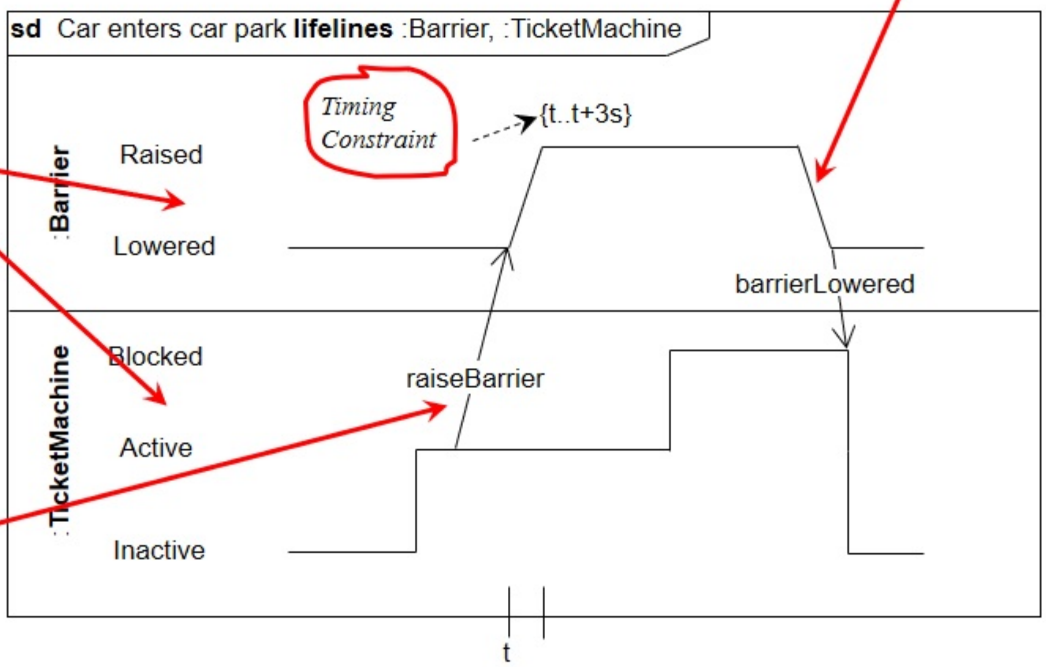
- The following sequence diagram `Car enters car park` is the basis for the subsequent timing diagram



Notation of Timing Diagrams

Sloped line represents duration of state change

Diagram has two instances, one for each lifeline



Message from one lifeline to another

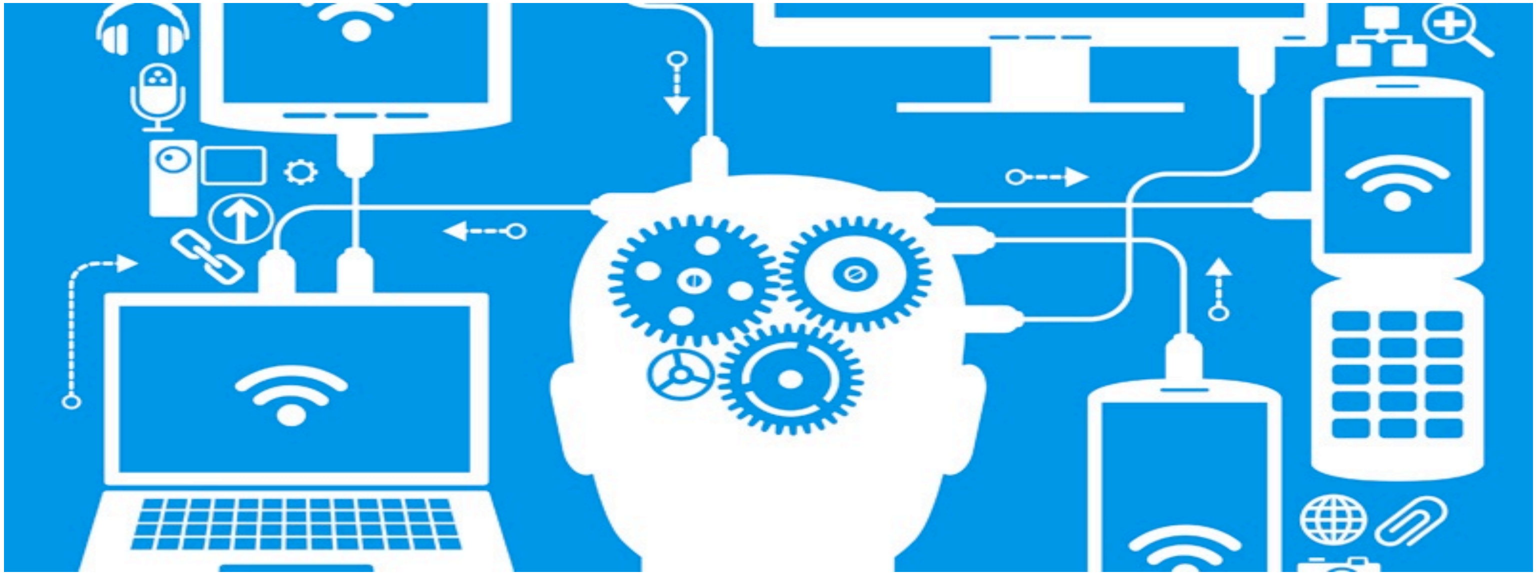
Key points

- Object interaction can be modelled using an interaction communication diagram.
- Interactions can be modelled using interaction overview diagrams.
- Interaction can also be modelled using an interaction sequence diagram.
- Timing diagrams show how time constraints affect interactions between lifelines.

References

- Simon Bennett, Steve McRobb & Ray Farmer. 2010. Object Oriented Systems Analysis and Design using UML 4th Edition, McGraw-Hill.
- UML Reference Manual (OMG, 2009)
- Bennett, Skelton and Lunn (2005)

In the next lecture..



Lecture 10: UML State Machine Diagrams