**IX. Sequence and Collaboration Diagrams**

Interaction Diagrams
Sequence Diagrams
Examples
Collaboration Diagrams

**Interaction Diagrams**
- Interactions among objects are modeled by interaction diagrams.
- An interaction between two objects A and B involves object A sending a message requesting an action that object B can perform.
- There are two types of interaction diagrams:
  - Sequence diagrams
  - Collaboration diagrams
- We discuss each in detail in the rest of this lecture unit.

**The Nature of an Interaction**

![Diagram showing Employee and Company with interactions](image)

**Sequence Diagrams**
- Sequence diagrams describe in detail how actors use use cases; they can also model external business processes the new system will support (e.g., processing a book order).
- An interaction is a behavior that consists of a set of messages exchanged between external and system objects.
- Interactions consist of one or more messages. Interactions may be synchronous (e.g., calling someone on the phone), or asynchronous (e.g., sending someone email).
- Sequence diagrams defined during requirements analysis should not:
  - include design objects;
  - specify message signatures in any detail.

**The Basic Idea**

![Diagram showing iteration, participating object, and time](image)

**Example: Add a New Campaign**
- Getting back to the use case “Add a new campaign”
  - Find client by name
  - Create new campaign
  - Find creative staff member by name
  - Assign campaign manager
Add another New Campaign

- Getting back to the use case “Add a new campaign”

Add new campaign
✓ Get client details
✓ Create new campaign
✓ Find creative staff member by name
✓ Assign campaign manager

A More Realistic Example

Add new campaign
✓ Find client by name;
✓ Create new campaign c;
✓ Assign creative staff member to c;
✓ Assign campaign manager;
✓ Inform the creative staff person.

An Even More Realistic Example

Another Example: Print Shop

Flow of Control

Iteration

Iteration (repetition of an operation) is shown with an asterisk
- Each StaffMember will be selected in turn
- Once selected, the CalculateBonus message will be sent to the
  one currently selected
- There is only one loop!
For a particular use case, start by identifying which objects and actors might be involved.

- You may not get this right, but you can always change it.
- Imagine that there is a use case required by Agate called Check Campaign Budget.
- Each Campaign has an EstimatedCost attribute and each Advert has an EstimatedCost attribute.
- The purpose of the use case is to check that the total estimated cost of all the adverts is less than that for the campaign as a whole.
- Which objects are involved here?

Where do we start?

- Select the relevant Campaign, probably using its name.
- How we select it is something we leave for the design phase:
  - it could be from a list box
  - it could involve a separate window on the screen
  - it could involve some kind of index
- These are design issues, which we shall leave for now, although we should document them if the customer expressed a preference at this stage.
We then need to send a message to the Campaign to check its budget. Note there is no Return here. Where does control go?

For each Advert
Get Cost of Advert
Return

What happens next?
Advert returns its cost, in this case the EstimatedCost of the Advert
Once all the Advert's costs have been fetched and totalled up, the total can be taken away from the EstimatedCost of the Campaign.

This has to happen for every Advert in the Campaign, so there's a loop
Once all the Advert's costs have been fetched and totalled up, the total can be taken away from the EstimatedCost of the Campaign.

We could add a new attribute to Advert called ActualCost, which is set when an Advert has been completed.
Now GetCost() can return the ActualCost if it exists, otherwise it uses EstimatedCost().
Collaboration Diagrams

- These diagrams are comparable to sequence diagrams. In fact, you can map every sequence diagram to an equivalent collaboration diagram and vice versa.
- Collaboration diagrams show interaction without the time dimension, but do include object links.
- Like sequence diagrams, collaboration diagrams are intended to model scenarios; each scenario describes a possible sequence of events and actions.
- Sequence diagrams are helpful because they capture visually the sequence of events over time.
- Collaboration diagrams capture more directly the interactions between actors and objects.

**Note:** All operations shown on collaboration and sequence diagrams must be present in the destination classes.

Select Courses to Teach

Add a Course Offering

Additional Readings