

CST315/SER394 Lab: UML Sequence and Communication Diagrams

(Fall 2015 revision)

In this module we talked about Sequence diagrams. Sequence diagrams are used in UML to capture the *interactions* between objects. They are often used in conjunction with use case scenarios as the scenario description steps are meant to describe interactions. Thus sequence diagrams are a category of diagrams called *Interaction diagrams* in UML. Their role in the system design process is to help understand how and where objects (component, subsystems, modules) need to talk to one another.

Objectives:

1. Understand and apply flow modeling via sequence diagrams to model interactions between objects.
2. Practice UML syntax and semantic “tools” in the context of a modern tool.

Tools:

1. *Visual Paradigm, Community Edition*
or
2. *Microsoft Visio*

Task 1 (individual): Create a Sequence diagram from a Scenario [50 points total]

A scenario description in a use case is like a scenario description in your UX modeling, only structured a little different. Use case scenarios are written in a step-by-step outlining style, and if properly done map readily to a sequence diagram.

For the grocery store step-by-step scenario on the next page, create a sequence diagram capturing the interactions. Some Hints are provided for you on the next page.

- Properly modeling an appropriate number of objects. [20 points]
- Properly modeling an appropriate number of messages. [30 points]

Task 2: Sequence Diagrams – translating from a Communication diagram [50 points total]

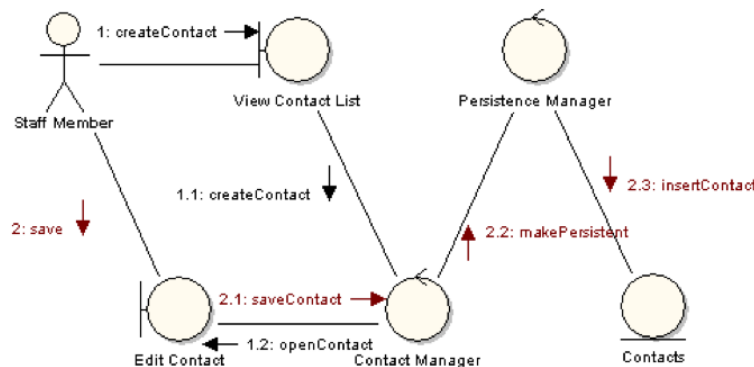
Take the Communication diagram in the figure below and translate it to a Sequence diagram. The general process is to convert each 'plain' object into one that has a lifeline, put them in a row at the top, and then put the communication diagram arrows (messages) into the correct order (see numbering) on top of those lifelines.

Note this diagram uses the UML stereotypes <<boundary>>, <<controller>>, and <<entity>>. In class, we used only the box notation – this only tells us we are working with objects. The stereotypes change how objects look so that it is easier to tell their role in the system.

- ❖ A boundary object (circle with left vertical line), indicates a part of the system that interfaces with an actor.
- ❖ An entity object (circle with bottom line), indicates a part of system that represents data/state.
- ❖ A control object (circle with top arrow), indicates a part of the system forming a layer between boundary objects and entity objects, that contains logic for the boundary object to use the entity objects.

(To use these in VisualParadigm, right-click on the object box when placing a Lifeline on the diagram. You will get a context menu that allows you to choose “Stereotypes”, and on that menu you will find these stereotypes.)

- Properly modeling an appropriate number of messages. [14 points]
- Messages are in an order that matches the communication diagram. [14 points]
- Objects are appropriately included and typed (i.e. use boundary, controller, entry stereotypes, etc). [18 points]
- Use proper stereotype symbols. [4 points]



Submission

- Please use a UML note to include your name.
- Submit a Word document containing exported images of each of the (two) diagrams you created.
- Name the file <asurite>_interactiondiagram.docx.
- Make sure that the diagrams you produce are syntactically and semantically correct. Your UML tool will help, but do not just manipulate the tool to get a picture, realize that the symbols mean something.

Scenario Description for a trip to the grocery store

0. Get in your car and drive to the grocery store
1. Enter the grocery store.
2. Find first item.
3. Verify marked price of first item
4. Put first item into shopping cart.
5. Find second item.
6. Verify marked price of second item
7. Put second item into shopping cart.
8. Can't find third item.
9. Get assistance from Jared the Assistant to find the third item.
10. Verify marked price of third item
11. Put third item into shopping cart.
12. Get into checkout line at register 3.
13. Get into checkout line at register 4 since 5 minutes passed.
14. Jody the Checkout Clerk checks you out.
15. Put your items into your car.
16. Get into your car.
17. Drive back to your place of residence.
18. Park the car there.
19. Take your items into your place of residence.

HINTS:

There doesn't have to be a one-to-one mapping of lines of text to stimuli (messages), but whatever you do, make it possible for the reader to trace back and forth between the two by putting the line number on the left into the message label on your diagram.

Stay at the same level of abstraction as the text. Do not, for example, raise the level to where there are only three or four stimuli, or lower the level by adding details that aren't indicated by the text. This is an exercise in translation, not engineering.

An interaction between 2 objects (you & your car) is represented as a message from one to the other.

A message is a request for service, so its direction is from the requester (a.k.a., client) to the provider.

If information flows in a message, it's common practice to show the info in parentheses after the name of the stimulus; e.g., *drive to (grocery store)*