



LAB 4: Interaction Diagram - UML Sequence Diagram

OBJECTIVES

- To understand the role of dynamic models in requirement analysis by reading and constructing UML Sequence Diagram.
 - To identify sequences of interactions between the objects of a system.
 - To identify the distinction between a Sequence Diagram and System Sequence Diagram.
-

INTRODUCTION

Interaction Diagram

- A series of diagram describing the dynamic behavior of an object oriented system.
- Often used to model the way a use case is realized through sequences of messages between objects.
- Purpose of Interaction Diagram:
 - Model interactions between objects.
 - Verify that the Use Case description can be supported by the existing classes.
 - Identify responsibilities / operations and assign them to classes.
- Common types of Interaction Diagram:
 - Sequence Diagram
 - Collaboration Diagram (Covered in next lab)
 - Activity Diagram (Covered in next lab)

Sequence Diagram

- Shows how processes operate with one another in what order.
- Illustrates how objects interact with each other.
- Emphasizes time ordering of the messages.
- Can model simple sequential flow, branching, iteration, recursion and concurrency.
- Key parts of a Sequence Diagram:
 - Participant: An object or entity that acts in the sequence diagram.
 - Message: Communication between the participants / objects.

System Sequence Diagram

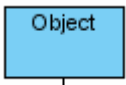

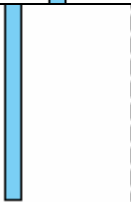
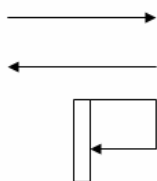
- A type of sequence diagram that shows (for a particular scenario of a use case), the events that external actors generate, their order, and possible inter-system events.
- SSD are visual summaries of the individual use cases.

- An SSD should show the following:
 - External Actors
 - Messages (methods) invoked by these actors.
 - Return values associated with previous messages.
 - Indication of any loop or iteration area.

System Sequence Diagram (SSD) versus Sequence Diagram

- SSD illustrates input and output events related to the system under discussion without concern of how such events are processed (treat it as a black box).
- SSD is typically associated with use-case realization in the logical view of system development.
- An SSD will always include a “system” object in its modeling.

Key Components of System Sequence Diagram

Notation	Display	Description
Object		The primary element of a Sequence Diagram. An object is an instance of a class.
Actor		Role played under the scenario.
Lifeline		Identify the existence of the objects over time / the ensuing scenario.
Message		Indicates communications between objects. Several types of messages exists, where you can represent simple messages, special messages (create and destroy objects) or message responses.

SEQUENCE DIAGRAM AND SSD WITH VP-UML

To illustrate the working flow of SSD and Sequence Diagram, we will base our guidelines to the problem sample below by creating a System Sequence Diagram for Process Sale:

Consider a Sale Process Scenario for a cashier processing sale activities in a



supermarket. The cashier is required to process any purchasing (sale) activity by the customer. The customer will firstly put their goods on the cash counter. The cashier will then enter the item ID (could be the barcode) and its quantity, so that the system can give the description of the item and total price. The process is repeated until all items are scanned and recorded as sale. The process will end afterwards, and will require the customer to pay the amount indicated.

- Describe the main objects (2 in this case):
- Describe the main flow of events / scenario.
- Translate them into system events for communication between the objects:
- Create the SSD.

The Use Case has been thoroughly analyzed and identified below:

1. Cashier starts a new sale.
2. Cashier enters item identifier (should be barcode of item).
3. System records sale item and presents item description, price and running total.
4. Cashier repeats steps 2 to 3 until all items processed.
5. System presents total.
6. Cashier informs the customer the total and asks for payment.
7. Customer pays and System handles payment.

The Use Case 2 column format is shown below:

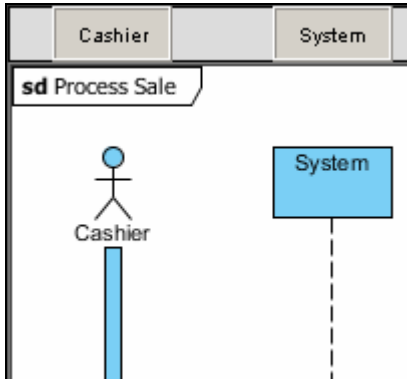
Actor	System Response
Cashier starts a new sale.	
Cashier enters item identifier	
	System records sale item and presents item description, price and running total.
	System presents total.
Cashier process the payment done on behalf of customer.	
	Customer pays and System handles payment.

- Create a new Sequence Diagram and name the diagram as “Process Sale”.
- Create the actors “Customer” and “Cashier”. To do this, click on the Actor icon under the Diagram Pane.

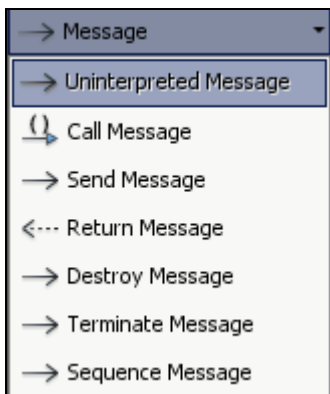
- Subsequently, create an object called “System”. This can be done by clicking the “Lifeline” icon under the Diagram Pane.



You should get the following display:



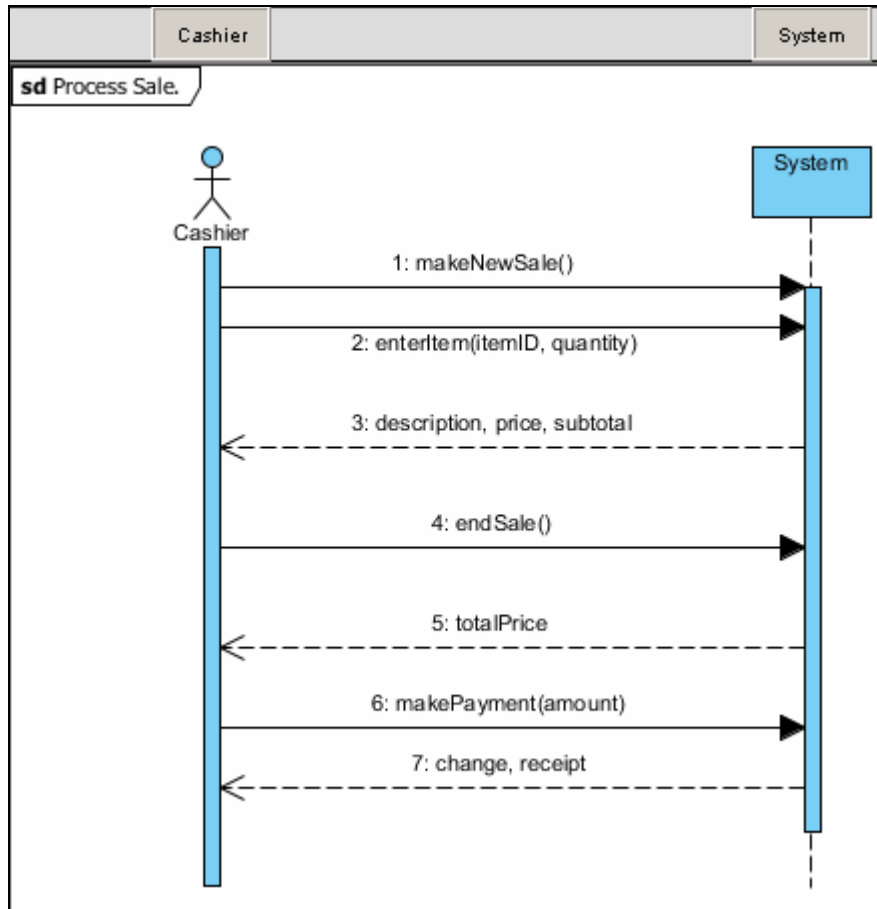
- To create messages between objects / actors, click on the “Message” icon under the Diagram Pane.



- Create message “makeNewSale()” and “enterItem(itemID, quantity)” linking between the actor Cashier and object System.

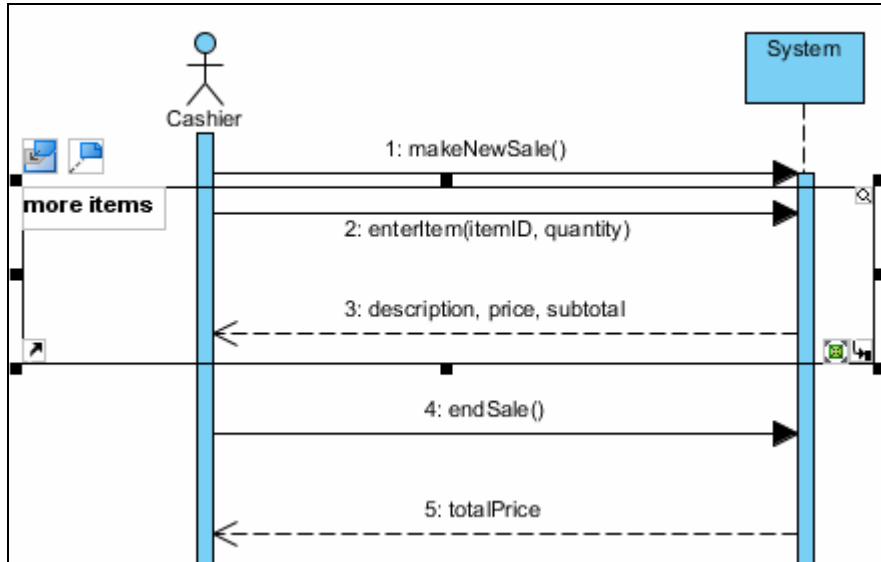
To create a return message, choose “Return Message” under the Message tree in the Diagram Pane.

- Create a return message “description, price, subtotal” from System to Cashier.
- Repeat the above steps to come up with a complete SSD shown below:

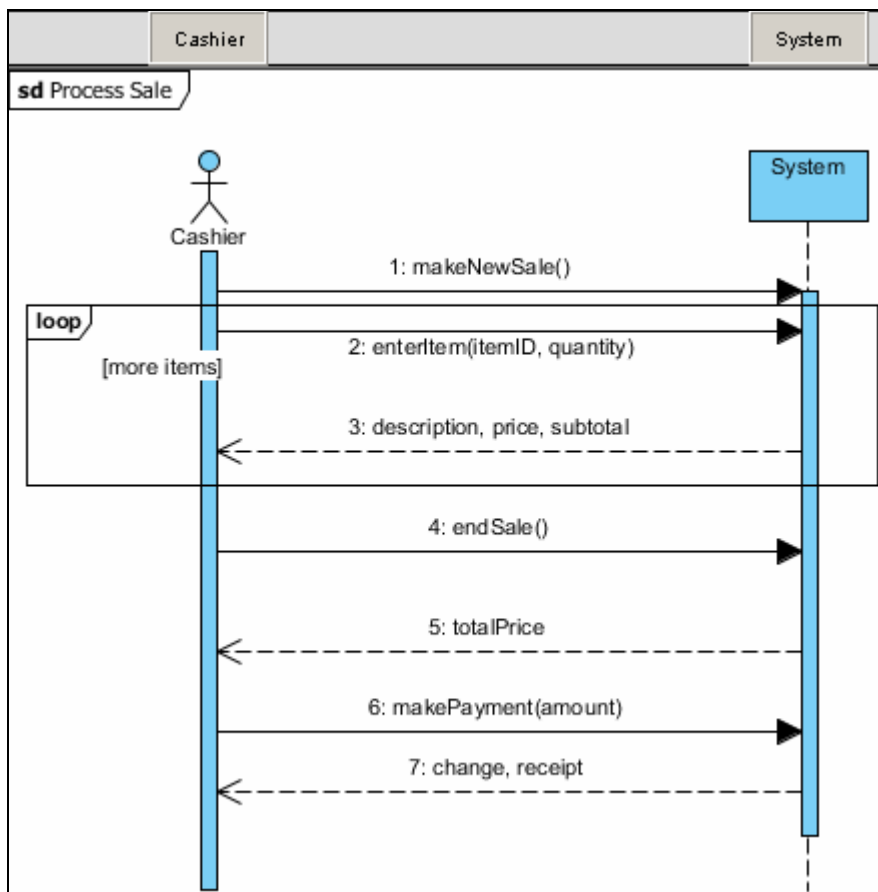


Consider that the cashier is required to process all sale items until completed. We need to model this with a loop fragment.

- To create a loop scenario fragment, click on the “Loop combined Fragment” icon under the Diagram Pane. Left click and drag to cover required selected events as shown below.



- A modified / updated SSD of Process Sale is shown below:



Updated System Sequence Diagram for Process Sale



Now we will model a more detailed Sequence Diagram for the process sale use case scenario.

Analyzing the situation concerned with processing sales items, we have come up with the following use case description:

Identified Actors / Objects: Cashier, Cash Register System, Sale, Product Item

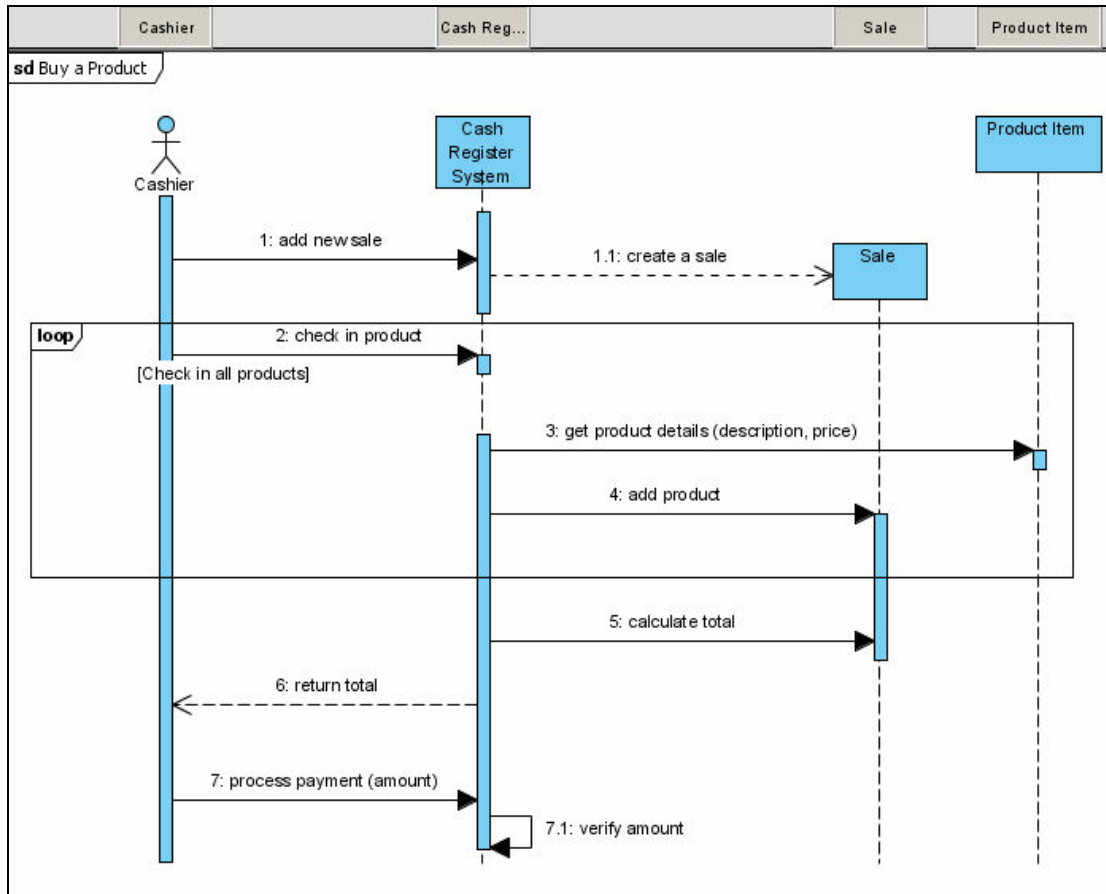
Identified Main Flow:

1. Cashier initiates a new sales record.
2. Cashier checks in product.
3. System retrieves product description and price.
4. System add product to sales record.
5. System repeats check in and adding until all products is checked-in.
6. System calculates total price of sale and indicates to cashier.
7. Cashier process payment by customer.

The Use Case Description 2 column format is shown below:

Actor	System Response
Cashier initiates a new sales record.	
Cashier checks in product.	
	System retrieves product description and price.
	System add product to sales record.
	System repeats check in and adding until all products is checked-in.
	System calculates total price of sale and indicates to cashier.
Cashier process payment by customer.	

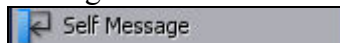
- Create a new Sequence Diagram named “Make New Sale”. Fill it in with the identified objects and messages as shown below.



Sequence Diagram for Make New Sale scenario

Consider that the cash register system must verify the amount being paid by the customer (which is processed by the cashier). This should require the system to invoke a self request for amount verification (done by self messaging).

- To create a self message, click on the “Self Message” icon under the Diagram Pane. Left click under the object “Cash Register System” lifeline to create a new self message.



In-Class Exercise on <<System Sequence Diagram>>

Model a scenario of the Withdraw Money use case of a Bank ATM system. The user is able to make withdrawal of money. The system employs a standard procedure of validating the card and account holder’s password.

- Describe the main objects (2 in this case):
 - Cardholder (or Customer)
 - ATM System



- Describe the main flow of events in this scenario. The first 2 events are listed below:
 - Customer arrives at the ATM machine and inserts a bank card.
 - The system requests for user authentication (password).
- Translate the flows into corresponding system events.
- Construct a SSD based on the information above.

In-Class Exercise on <<Sequence Diagram>>

Now model the scenario of the Withdraw Money use case in more detail. Where the SSD showed only input and output generated by external actors and the system object, this Sequence Diagram will elaborate further by including new internal objects under the withdraw money use case such as Card Controller (to control card management) and Bank (the issuing bank).

- Describe the main objects (few are given below):
 - User
 - ATM System
 - Card Controller
- Describe the main flow of events in this scenario. The first 2 events are listed below:
 - Customer arrives at the ATM machine and inserts a bank card.
 - The card is verified by the Card Controller.
- Translate the flows into corresponding system events (input and response).
- Construct a Sequence Diagram based on the information above.

End of Lab 4.