INTERACTION DIAGRAMS CLASS NOTES

Interaction diagrams show how to record details how objects interact to perform task
UML provides three main interaction diagrams

- Sequence diagrams
- Communication diagrams
- Collaboration diagrams

SEQUENCE DIAGRAMS

Sequence diagrams show how objects communicate with one another. They to identity at detailed level the operations that are necessary to implement the functionality of a use case. They include time as important dimension. The idea is that interactions among objects take place in as specified sequence and the sequence takes time to go from the beginning to the end. Interaction specifies the communication pattern amongst sets of objects or systems that participate in the collaboration. Interaction is described is partially ordered sequence of the messages between objects or communicating roles they play and communicate by sending messages. Objects are modeled in terms of the roles the play and communication by sending messages. hey can be drawn at different levels of detailed to meet different purposes at several stages in the development lifecycle. Sequence diagrams consists of objects represented as rectangles (with name underline), messages represented as vertical progressions.

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram. Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications. Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures, in business process engineering as process flow diagrams, as message sequence charts and call flows for telecom/wireless system design, and for protocol stack design and analysis.

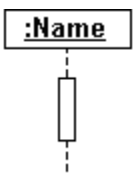
BASIC CONCEPTS AND NOTATIONS

Actor

Represents an external person or entity that interacts with the system

Object

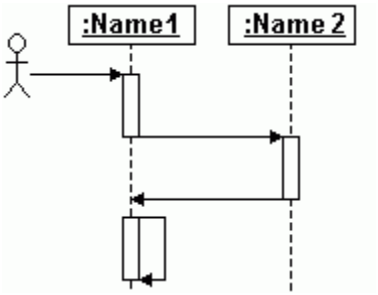
All objects are spread horizontally across the diagram. The vertical dimensions represents time. Each object (classifier role) in a sequence diagrams is represented by a dash-line extending downloads called the object's lifeline.-this is a dash-line with object symbols at the top.



Message
A message is shown by a solid horizontal arrow from one life-line to another and is labeled with messages name.
A message goes from one object to another i.e goes from one object's lifeline to the other object's lifeline. An object can send a message to itself i.e from its lifeline back to its own lifeline-called recursion.
Types messages
Simple message
This is a transfer of control from one object to another.
Synchronous
If an object sends a synchronous message it waits for an answer before it proceeds with its operations.
Asynchronous
If an object sends an asynchronous message it doesn't waits for an answer before it proceeds.

Simple	This is a transfer of control from one object to another.	
synchronous	If an object sends a synchronous message, it waits for an answer to that message before it proceeds with its business.	
asynchronous	If an object sends an asynchronous message, it doesn't wait for an answer before it proceeds.	

Time
Time starts at the top and progresses towards the bottom. A message closer to the top occurs earlier in the time than a message close to the bottom.

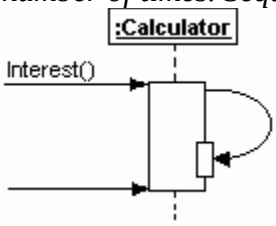


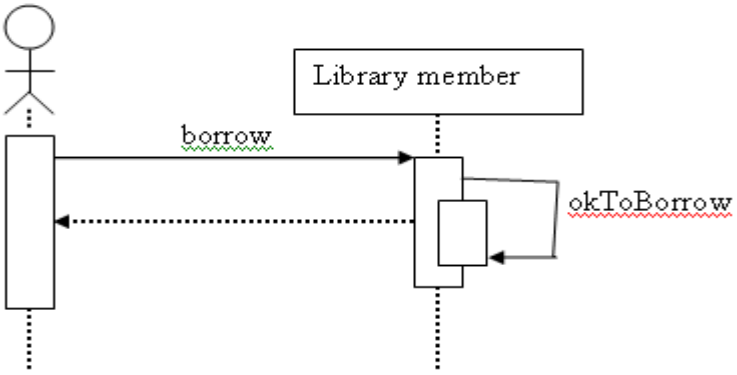
ACTIVATION LIVE/EXECUTION OCCURRENCE.
Once a message is received the operation that has been invoked begins to execute.
The period during which an operation executes is known as activation or execution occurrence (executive live) and sis shown by a narrow rectangle laid along the life-line

Recursion/recursive messages
An object can send a message to itself. This is known as reflexive message and is shown by an arrow that start and finishes at the same object lifeline. Activation invokes itself on the same object and is known as recursion.
In this case the reflexive message invokes a different operation from the operation that sent the message and a new activation symbol is stacked on the original execution occurrence(nested activation) . the object already had a live activation when it sent the message, now it has a new different activation because it is also the receiver of the message.

Suppose one of the objects in your system is a calculator, and suppose one of its operations computes interest. In order to compute compound interest for a timeframe that encompasses

several compounding periods, the object's interest-computation operation has to invoke itself a number of times. Sequence diagram for this is following:

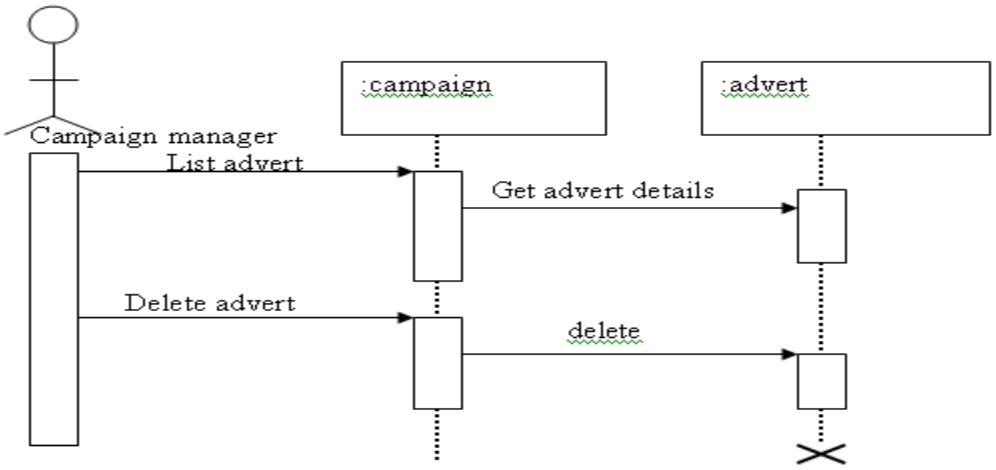


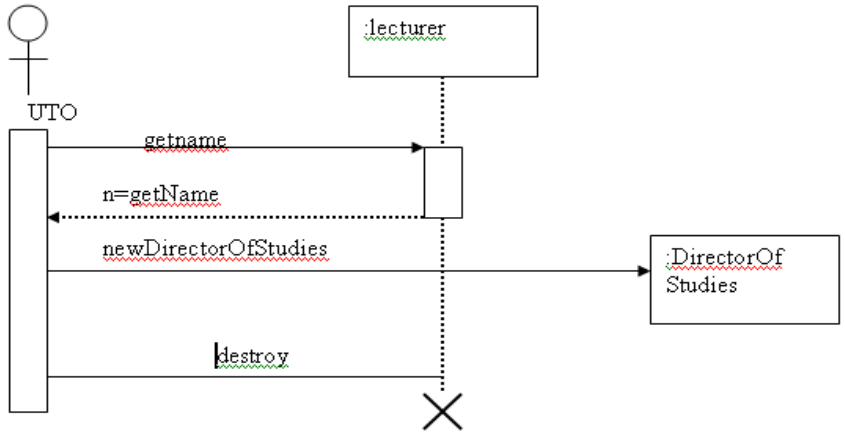


CREATION AND DELETION OF OBJECTS

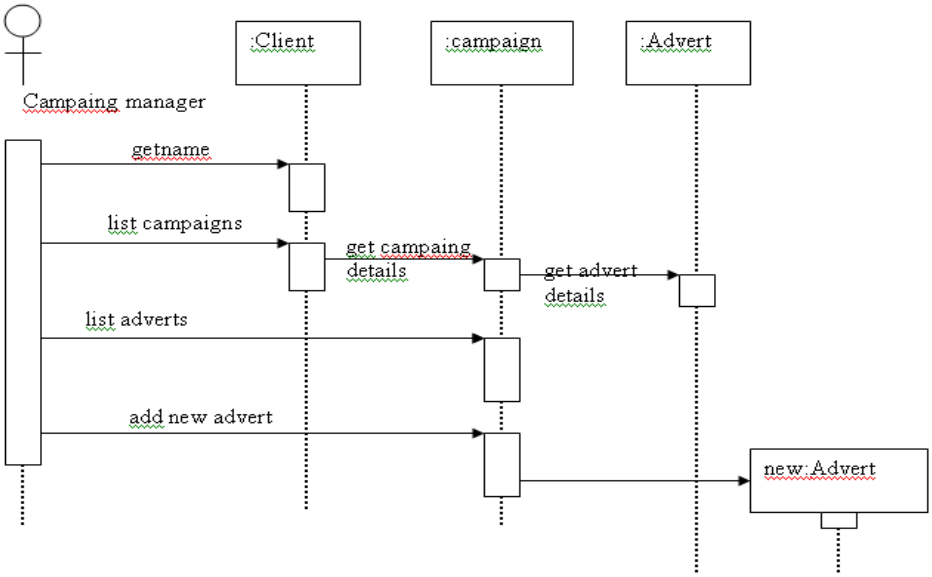
Objects in an interaction diagram are not always static, objects may be created and deleted during an interaction. Sequence diagrams provide notation to show the creation and deletion of objects. Objects may be created or destroyed at different stages during an interaction. Destruction of an object is shown by its activation ending with large X on the lifeline at the point in the interaction when the object is destroyed. An object may either be destroyed when it receives a message or it may be self-destruction at the end of an execution if it is required by the operation that is being executed. The created object is shown at the point where it is created.

Promote a class who is described by an object of class lecturer who is promoted to being director of studies, a new object of class director of studies is created and the old lecturer object must be deleted

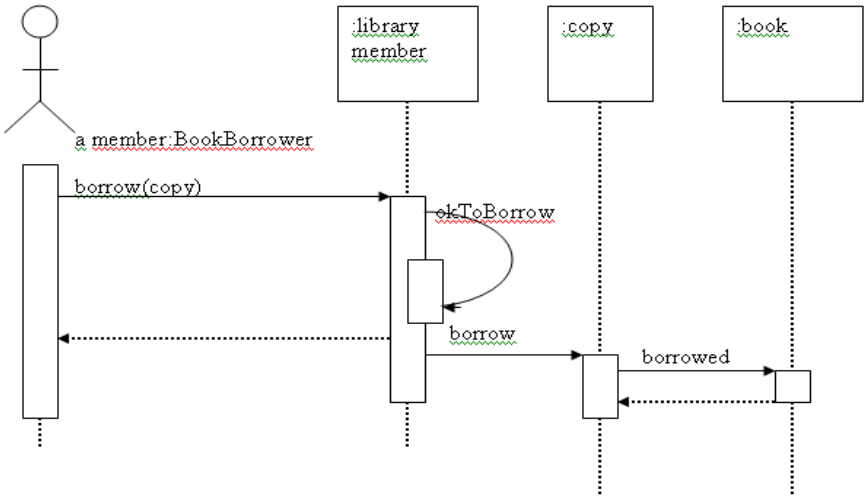


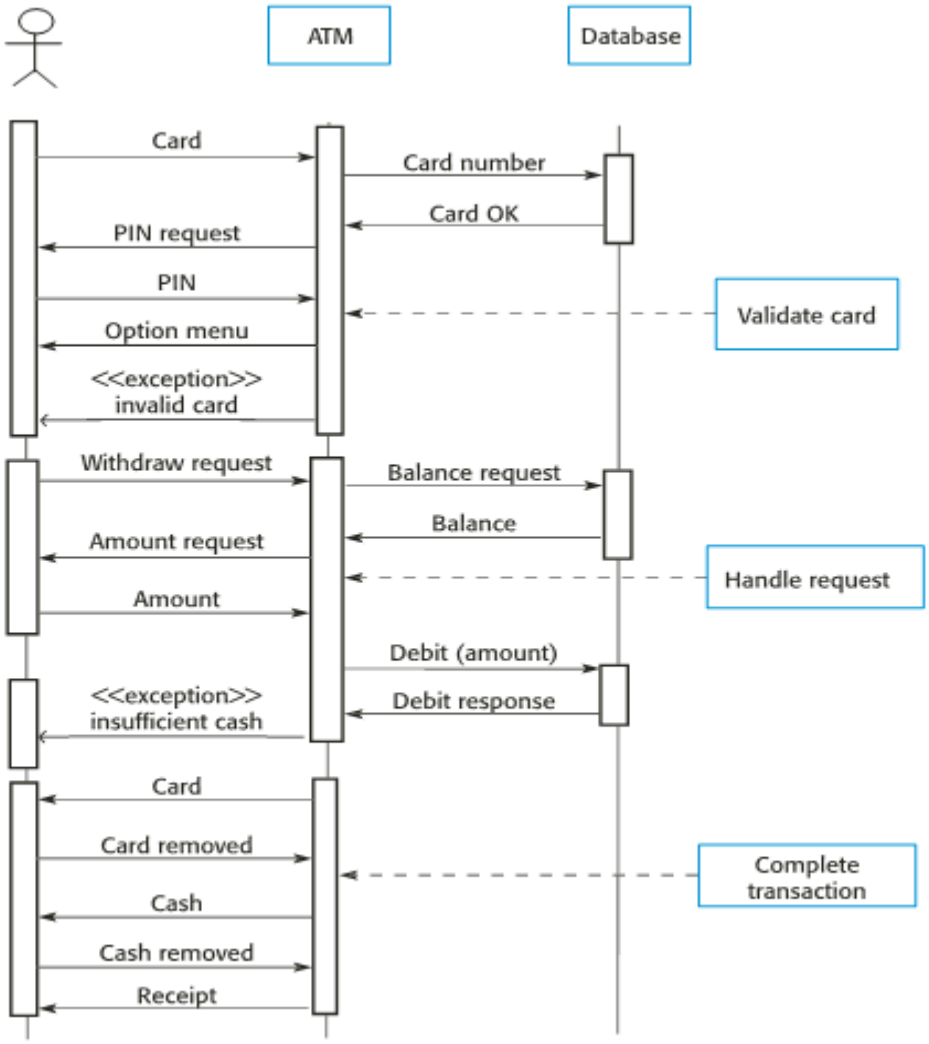


Give the text description of the sequence diagram below.

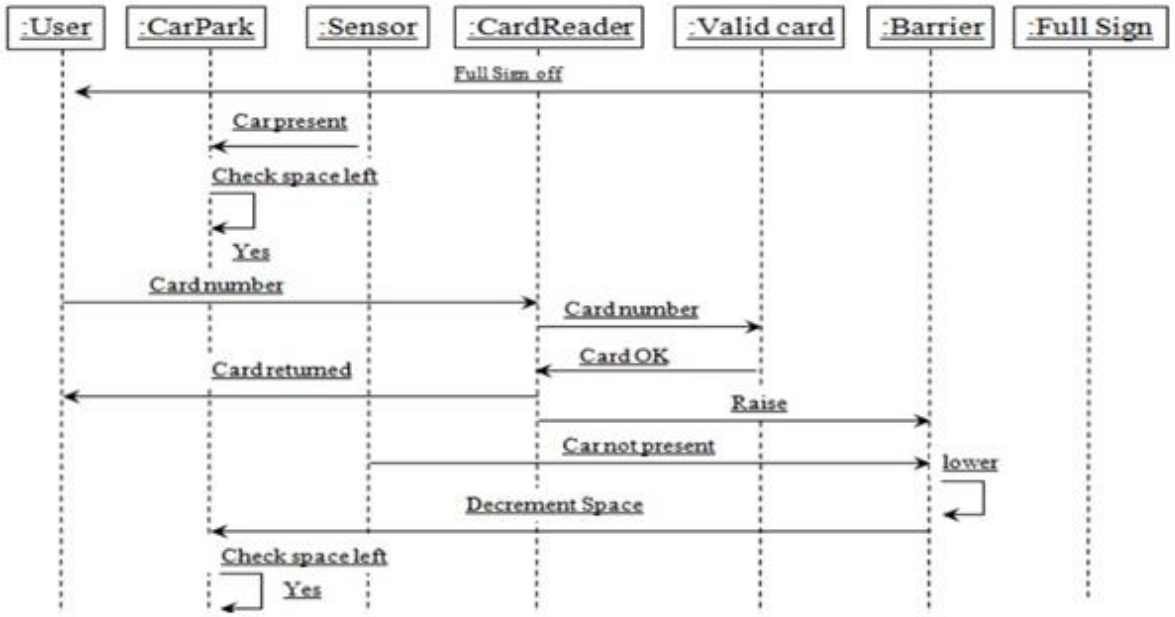


Sequence from the library system scenario



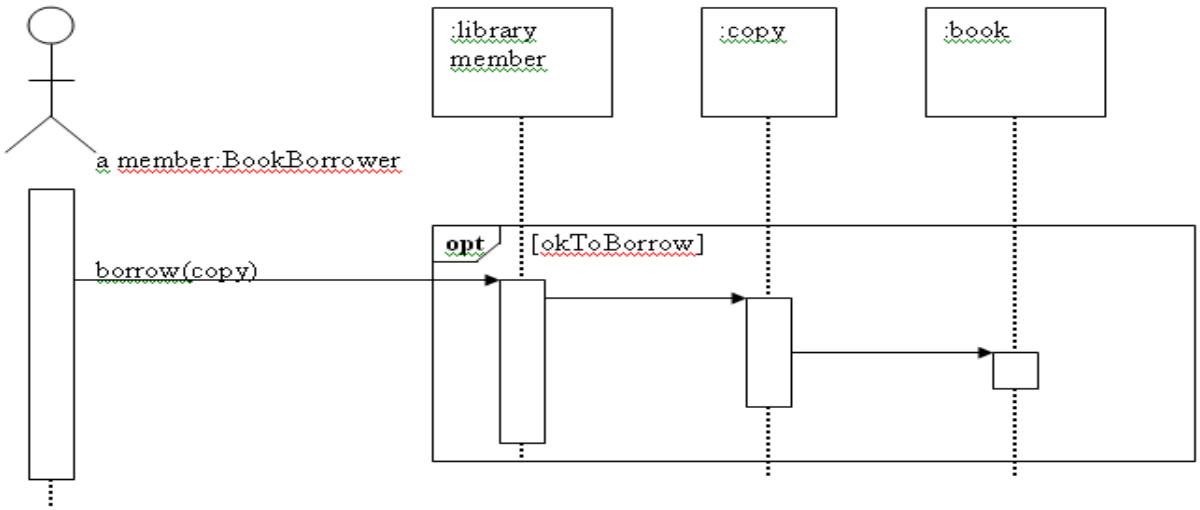


Just a Line management wishes to increase security, both in their building and on site, without antagonizing their employees. They would also like to prevent people who are not part of the company from using the Just a Line car park. It has been decide to issue identity cards to all employees, which they are expected to wear while on the Just a Line site. The cards records the name, department and number of the member of staff, and permit access to the Just a Line car park. A barrier and a card reader are placed at the entrance to the car park. The driver of an approaching car insert his or her numbered card in the card reader, which then checks that the card number is known to the Just a Line system. If the card is recognized, the reader sends a signal to raise the barrier and the car is able to enter the car park. At the exit, there is also a barrier, which is raised when a car wishes to leave the car park. When there are no spaces in the car park a sign at the entrance display “Full” and is only switched off when a car leaves. Special visitor’s cards, which record a number and the current date, also permit access to the car park. Visitor’s cards may be sent out in advance, or collected from reception. All visitor’s cards must be returned to reception when the visitor leaves Just a Line.

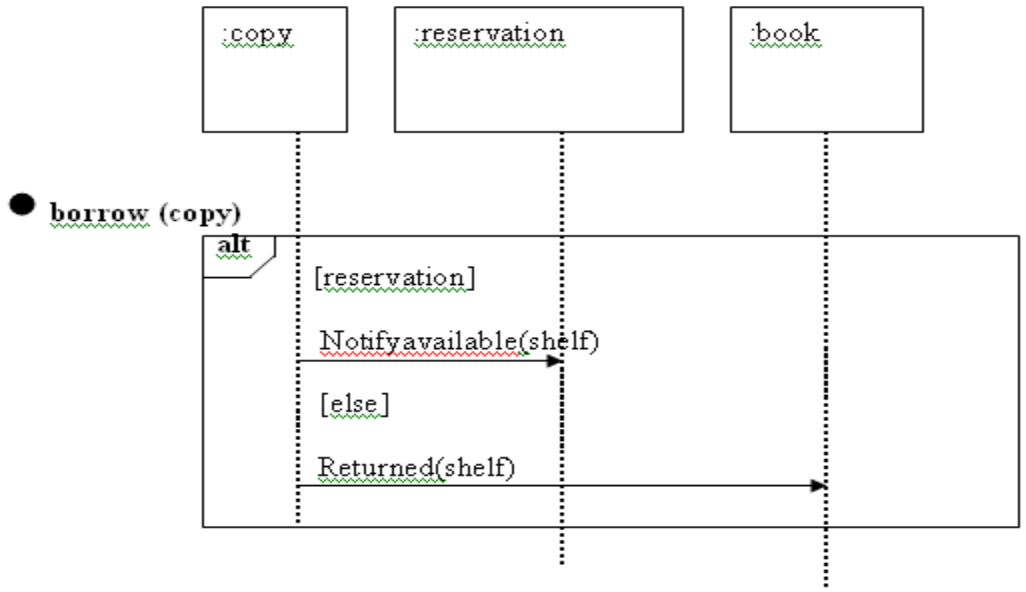


CONDITIONAL BEHAVIOR/ITERATIONS

Use cases can include different scenarios. It is useful to show conditional behavior or a variable number of iterations in an interaction. We can put a guard on an individual message, by writing the condition is square brackets in front of the message. In sequence diagrams conditional behavior is enclosed in a rectangle. The rectangle is labeled opt short for 'optional' and the guard condition is written in square brackets close to the opt label.



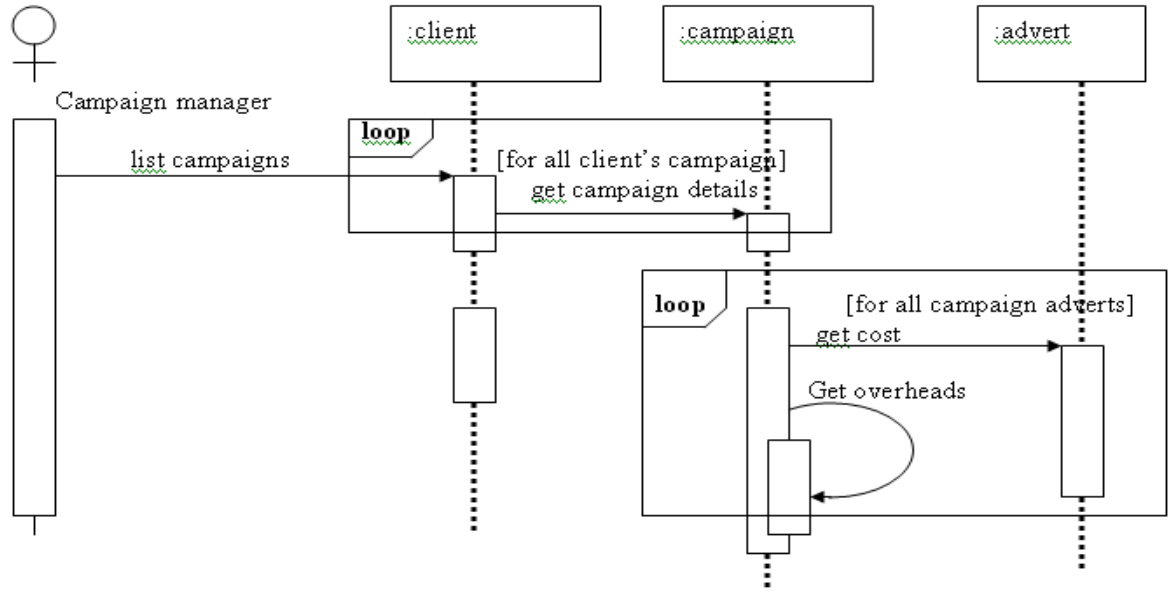
If/then/else statement or a case statement, you can use similar notation but labeled alt rather than **opt**



ITERATION/LOOP.

One object may send a message to another some number of times.

In sequence diagrams you can use a rectangle labeled loop with the iteration clause nearby.



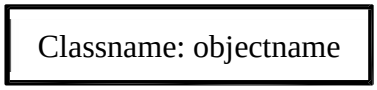
COLLABORATION DIAGRAMS

Collaboration diagrams show how objects interact. Objects interact to perform some task. Objects which interact to perform some task together with the links between them are known as collaborations

Basic concepts and notation

Object

Each object is shown as a rectangle. The name of the object and name of the class are separated by a colon.



Links

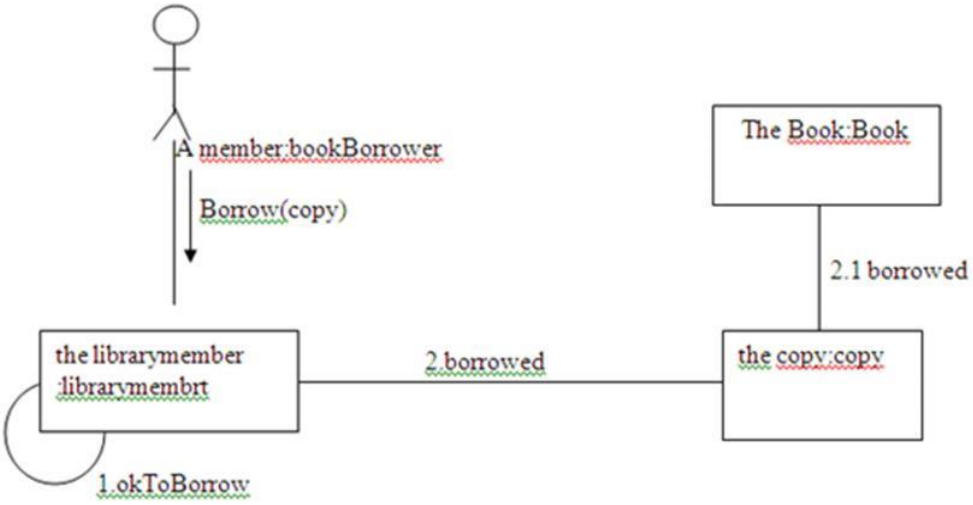
Links between objects are shown like associations in the class diagram. Lines connecting objects are links

Actor

Messages

Messages are represented as arrows that point to the receiving object near the association line between the two objects. The arrow represents messages and are labeled with their names, sequence numbers and assignments. The sequence numbers show the order in which the message occur (we add a number corresponding to the message's orders in the sequence) message numbers are nested so that you can tell which messages are sent from within other messages

Example diagram from the library scenario.



Example

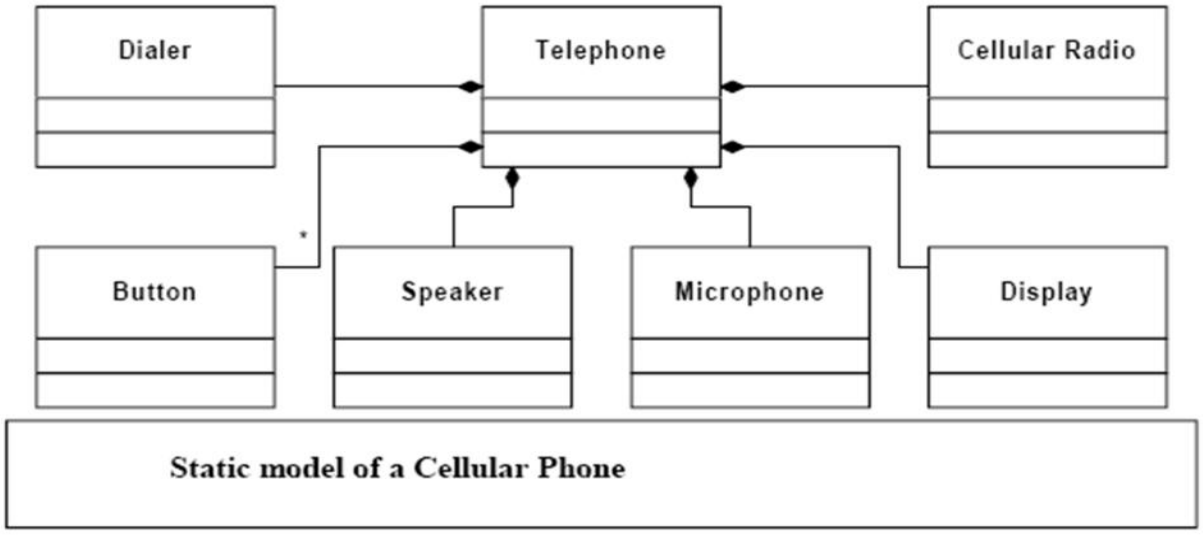
Cellular phone

Consider the structure of a cellular phone handset. The phone has buttons for entering digits, a "send" button for initiating a call, a microphone a speaker, and a display. It also has a "dialer" that gathers the digits together and emits the appropriate tones and a cellular radio that deals with the connection to the network.

The steps involved in making a phone call with such as phone might be summarized as follows

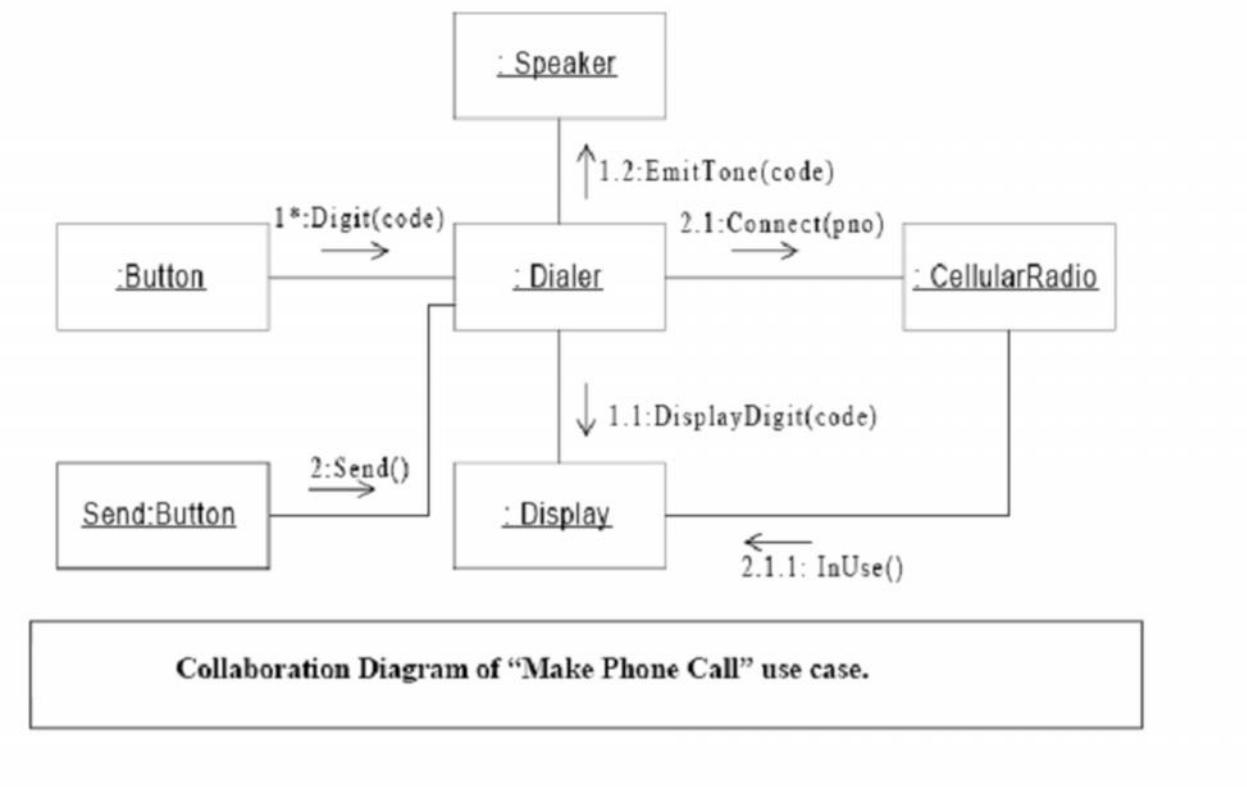
1. The user presses the digit buttons to enter a phone number
2. The display is updated as each digit is added to the phone number
3. The dialer generates a corresponding tone for each digit as it is entered and emits this from the speaker
4. The user presses "Send"
5. The cellular radio establishes a connection to the network
6. The phone number is sent to the network
7. The connection is made to the phone with specified phone number

From this simple spec, we might be tempted to create a static model as shown in the figure below



How does the cellular phone work? Let's just look at how a customer might make a phone call. The use case for this interaction looks like this:

Use case: Make Phone Call
1. User presses the digit buttons to enter the phone number.
2. For each digit, the display is updated to add the digit to the phone number.
3. For each digit, the dialer generates the corresponding tone and emits it from the speaker.
4. User presses "Send"
5. The "in use" indicator is illuminated on the display
6. The cellular radio establishes a connection to the network.
7. The accumulated digits are sent to the network.
8. The connection is made to the called party.



Collaboration Diagram of "Make Phone Call" use case.

Revision Questions

1. A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line item detail. The customer enters payment information, which the system validates the records. The system updates inventory. The customer receives a receipt form the system and then leaves with the items. Model this scenario using a sequence diagram to capture the occurrence of the events and messages in the system
   (10 marks)

2. Create a collaboration diagram for the following interaction. A customer wants to draw money from his bank account. He enters his card into an ATM (automated teller machine). The ATM machine prompts “Enter PIN”. The customer enters his PIN. The ATM (internally) retrieves the bank account number from the card. The ATM encrypts the PIN and the account number and sends it over to the bank. The bank verifies the encrypted Account and PIN number. If the PIN number is correct, the ATM displays “Enter amount”, draws money from the bank account and pays out the amount

   (10 marks)

3. Object oriented systems are said to be systems whose the overall behavior is determined by sum of all effects of collaborating objects through message passing. Describe what happens when any one of these objects receive a message

4. Consider the following scenario. Produce a collaboration diagram showing the messages that are exchanges in the system. Start from selecting bank name: process should be based on select Bank name, select card, verify password, select account, update account. Data flow will be bank, card authorization password, data store will be bank name, account. Control flow will be bad bank name code, invalid card code, bad password, bad account, transaction failed. The actual update process applies the transaction kind and amount to select the account.

5. A customer walks into a baker’s shop and asks the baker what kind of loaves she has for sale. The baker looks under the counter and tells the customer that she has two white loaves and one wholemeal loaf. The customer says that he would like to buy the wholemeal loaf. Now the business transaction takes place: the baker wraps the loaf and offers it up with a request for payment; the customer gives the baker some money; the baker gives the customer some change. The customer leaves, satisfied. Draw a collaboration diagram to show messages passed among the objects involved in the transaction