

**END-TERM EXAMINATION  
SIXTH SEMESTER (B-TECH)  
OBJECT-ORIENTED SOFTWARE ENGINEERING**

**Time : 3.00 hrs.**

**M.M. : 75**

**Note:** *Attempt any five questions including Q.no. 1., which is compulsory.*

**Q. 1. Tick True/false with justification for the following questions. (25)**

**(a) In UML, class diagram only addresses the static design view of a system.**

**Ans. (a) True**

These diagrams show a set of classes, interfaces and collaborations and their relationships.

**(b) UML can be optionally used in any of the software life cycle model.**

**Ans. True:** UML can be optimally used in any of the software life cycle model as it is a graphical presentation of a set of elements. It is also called as a connected graph of vertices and arcs.

**(c) UML is used only to visualize, specify and document the artifacts of software systems.**

**Ans. True:** UML is a standard language which are not only for developers but also for business documenting.

**(d) In UML sequence diagram and collaboration diagram is isomorphic in nature.**

**Ans. True.**

**(e) In UML, use case diagram and state chart diagram both capture the behaviour of a system.**

**Ans. True:** The use case diagram and state chart diagram describes the static and dynamic behaviour of the whole system. It shows the functionalities of the overall system.

**(f) During construction process, implementation and test model of a system are produced.**

**Ans. True.**

**(g) The analysis model should only be viewed as a conceptual model of a system.**

**Ans. False:** The analysis model, design model and the structured systems are viewed as a conceptual model of a system.

**Q.(h) The design model is further refined through description of all use cases of a system.**

**Ans. True.**

**Q.(i) The analysis model is used as the primary tool when doing integration testing.**

**Ans. True.**

**Q.(j) During analysis requirement models of a system is built.**

**Ans. True.**

## SECTION A

**Q.2. Name five object oriented methods except UML and compare any three mentioning the strength and weakness while modeling the various aspects of a system. (12.5)**

**Ans.** Five object oriented methods are explained below.

**(a) Rumbaugh's Modeling (OMT)**

OMT stands for Object Modeling Technique and is given by Rumbaugh in 1991. OMT consists of 4 phases.



**Phase-I: Analysis:** Initially, the requirements are stated in a problem statement. From this statement, relevant classes are extracted and also their relations and attributes.

**Phase-II: System Design:** The outputs are a structure of basic system architecture and also high level strategy decisions.

**Phase-III: Object Design:** The output's are a design document consisting of detailed object static, dynamic and final models. It provides detailed specification of object implementations.

**Phase-IV: Implementation:** It provides reusable, extensible and robust code.

**(i) Object Model:** It is a static model used to describe the structure of object in a system, identify attributes, operations and relationships with other objects in the system.

**(ii) Dynamic Model:** It represents the state transition diagrams represents in (OMT) a network of states and events.

**(iii) Functional Model:** It is used to visualize the data flow between different processes in a business. It is based on classical DFD and used to specify global functions that operate on the objects.

**(b) Booch's Modeling:** It is aimed for physical design and programming and not for requirements analysis and conceptual design. It concentrates on static model. Booch's OOD describes diagrams at two level.

**(i) At Design level:** Class diagrams, object diagrams, state transition diagrams and interaction diagrams are at this level.

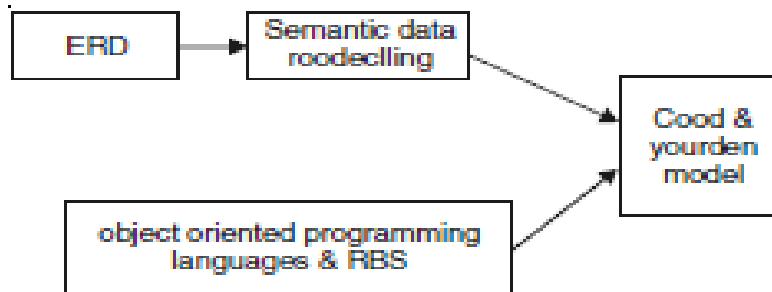
**(ii) At implementation level:** Module diagrams and process diagrams are at this level.

**(c) Goad and Yourdor's Modelling:** OOA/OOD is a single model that describes the static aspects of a system and not on dynamic or functional models. It is on dynamic or functional models. It is an dynamic or functional models. It is extension of ERD to represents object with complete relationship like aggregation, composition, specialisation and generalization.

**Coad and Yourdor's OOA/OOD Model.**

OOA consists of 5 steps:

- Finding classes and objects
- Defining subjects
- Defining services
- Defining attributes
- Identifying structures



**(d) Shalaer and mellor's modeling:** The OOA method consists of 2 parts:

**(i) Relationship Modeling of Data:** It introduces generalization and inheritance concept.

**(ii) Dynamic and Functional Aspects:** So, it is based on three models, i.e., static, dynamic and functional.

- **Object Relational Model/Static Model:** It is represented by the concept of objects and attributes. There is no object encapsulation since the object does not include any methods.

- **Object Dynamic Model:** It is based on STDs. The object by cycle (OWC) is described by a succession of states and the state change occurs when the object responds to messages.

- **Object functional Model/Process Model:** It is represented by the concept of objects and attributes. There is no object encapsulation since the object does not include any methods.

- **Object Dynamic Model:** It is based on STDs. The object by cycle (ohc) is described by a succession of states and the state change occurs when the object responds to messages.

- **Object Functional Model/Process Model:** It explains the activity associated with a state of the object. It is shown by a data flow diagram. So, it includes data flow, data storages, process flow and control flow.

**(e) Coleman's Modelling (FUSION):** The fusion method has been derived from Booch's OOD, Rumbaugh's OMT and other formal methods. It has 3 specialization levels analysis (conceptual model), design (interaction model) and implementation.

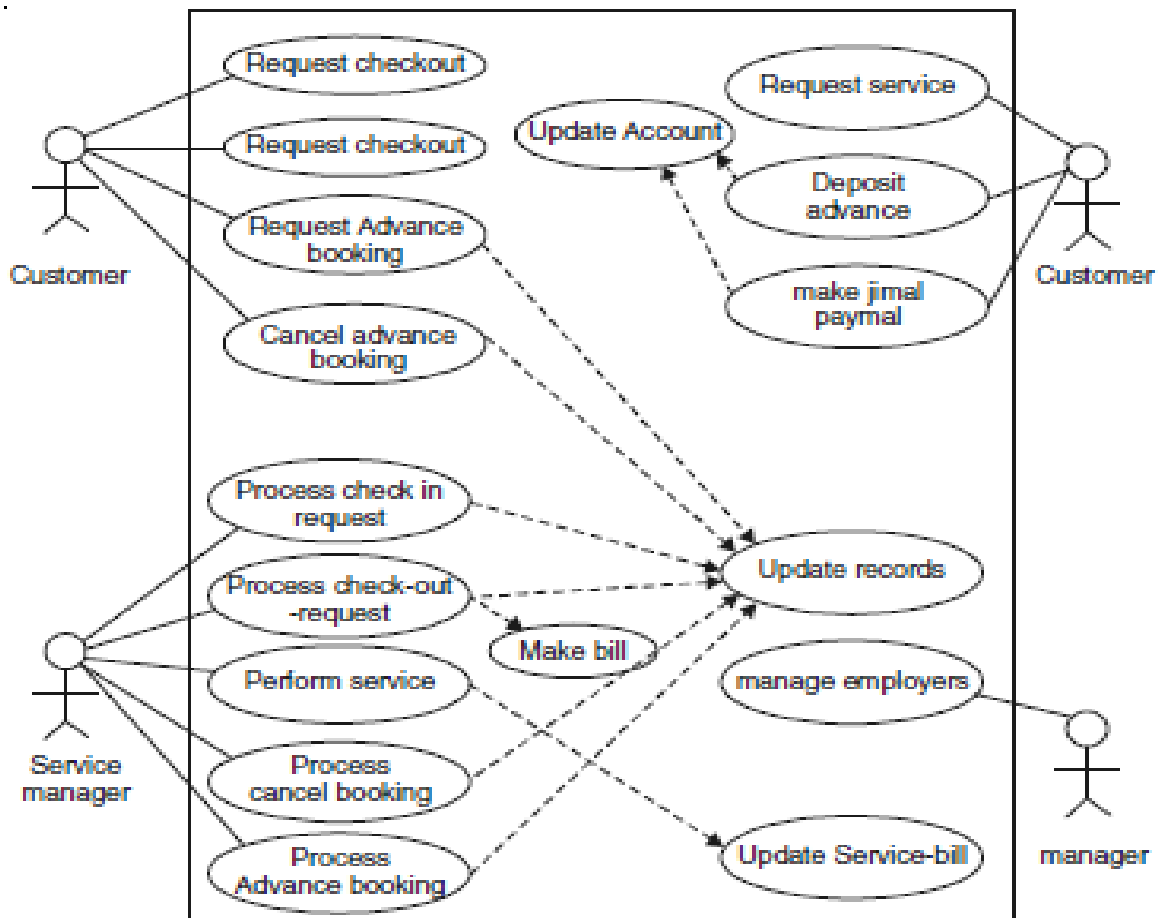
- **Conceptual Analysis Model:** It includes ERD, aggregation and generalization concepts. More emphasis is given to Data base objects rather than or user interface programming objects. This model has a provision to represent both binary and n-ary relationship.

- **Interaction Design Model:** It is used to describe the objects and the message exchanged between them. The focus of this model is on refining of the system operations rather than just mapping of the objects.

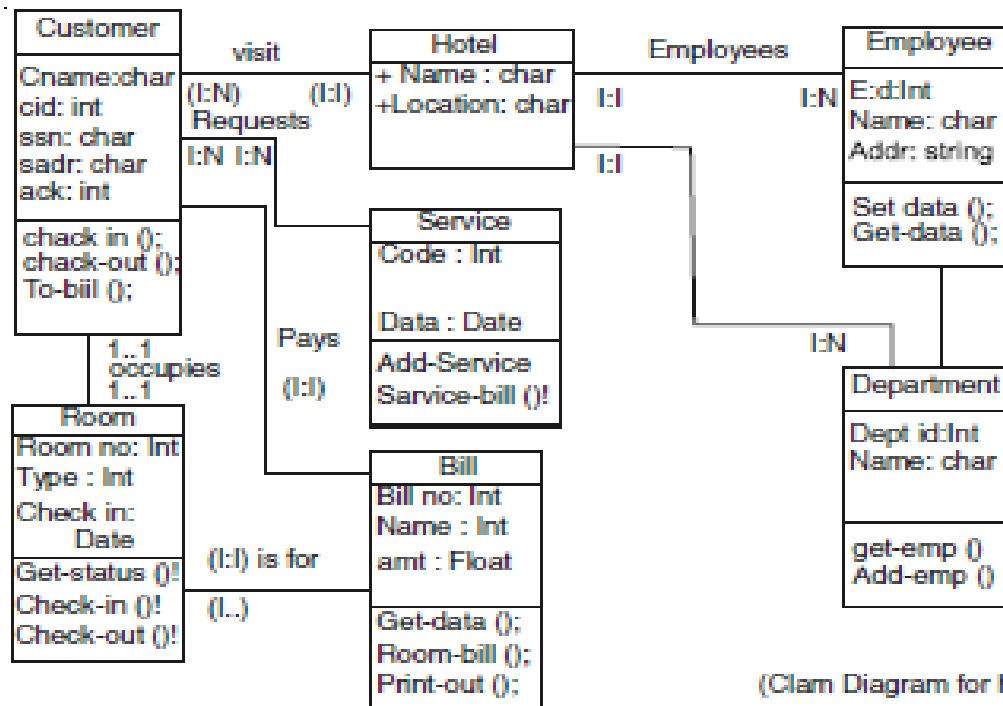
**Q.3. Write down the requirements for a typical hotel management system and draw use case diagram and class diagram. (12.5)**

**Ans.** The requirements for a typical hotel management system are.

- (i) Customer
- (ii) Service Employee
- (iii) Manager.
- (iv) Records.



Use case Diagram for hotel management system



(Class Diagram for hotel)

## SECTION-B

**Q.4. Explain the development processes involved starting from the elicitation of requirements to the analysis models considering on Automaed Teller Machine as a illustrating example. (12.5)**

**Ans.** Requirement elicitation is the practise of collecting the requirements of a system from users, customers and other stakeholders. The most important good of requirement engineering is to find out what user really need. Users need can be identified only if we understand the expectations of the users from the desired software. In this process first of all interviews are conducted. During the intervies both the parties would like to understand each other. The objective of conducting on interviews is to understand the customer expectations from the software. In brainstorming session creative thinking is promoted in order to generate new ideas. Then the FAST facilitated Application Specification Technique) session is conducted to bridge the expectation gap a difference between what developers think they are supposed to build and what the customer think they are going to get. As per the requirement as gathered using FAST or QFD, the software engineer can create a set of sesnaries that provide a description of how the system will be used.

**Ex- ATM system.**

**Elicitation of Requirements:**

Actor Bank Customer–Person who owns one or more  
Accouts on the bank.

Withdraw Money–

The Bank customer specifies a Account and provide credentials to the Bank providing that he or she is authorized to acces the Bank Account.

The Bank Customer specifies the amount of money he or she wishes to withdraw.

The Bank checks if the amount is consistent with the rules of the Bank and the state of the Bank Customer's amount.

If that is the case, the Banks customer receives the money in cash.

**Analyses.**

Use case attributes

Use case withdraw money using ATM

Initiating actor:

- Banks Customer

Proconditions:

- Bank Customer has opend a Bank Account with the Bank and
- Banks customer has received an ATM Card and PIN post conditions.
- Bank Customer has the requested cash or
- Bank customer receives an explanation from the ATM about why the cash could not be dispensed.

**Use Case Flow of Events:**

**Actor Steps:**

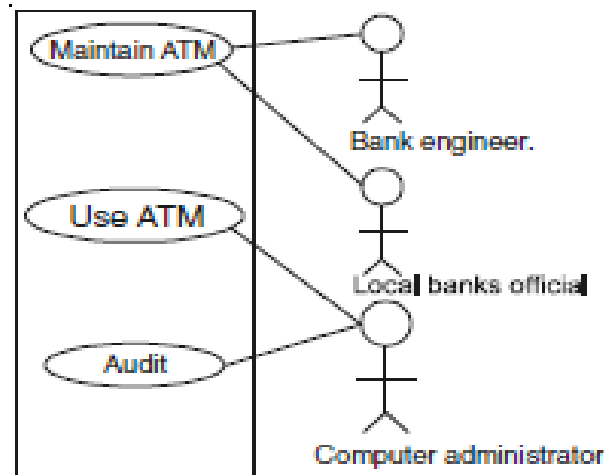
1. The Bank customer inputs the card into the ATM.
3. The Bank Customer types in PIN
5. The Bank customer selects an account
7. The Bank customer inputs an amount.

**System Steps:**

2. The ATM requests the inputs of a four-digit PIN.

4. If several accounts are recorded on the card, the ATM offers a choice of the account numbers for selection by the Bank customer.
6. If only the account is recorded on the card or after the selection, the ATM requests the amount to be withdraws.
8. The ATM outputs the money and a receipt and steps the interaction.

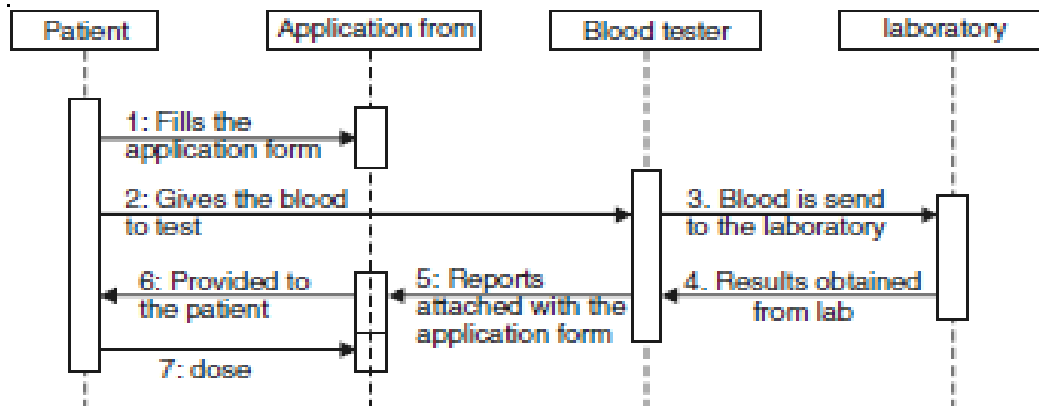
**Use case Diagram for ATM:**



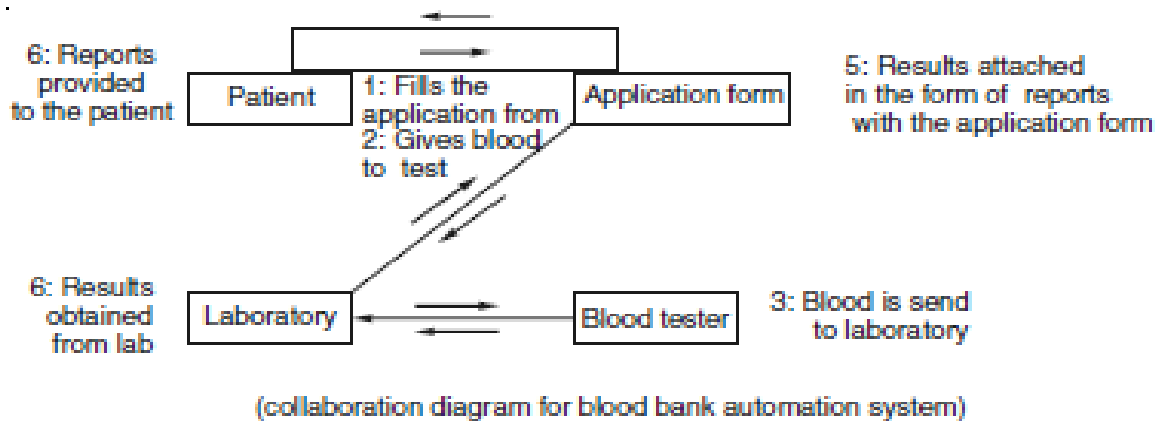
**Q.5. How do you relate sequence diagram and collaboration diagram? Explain with an example of Blood Bank Automation system. (12.5)**

**Ans.** Sequence diagram describes timing sequence of the objects over a vertical time dimension with interactions between object depicted on a horizontal dimension. Whereas the collaboration diagram describes the interactions and relationships between objects and sequences of a system organised on time and space. Numbers are used to show the sequence of messages.

Sequence diagram displays interaction between objects from a temporal stand point and it focuses on expressing interaction on the messages whereas a collaboration diagram represents collaboration between objects and these objects are related in a particular context, interaction.



(Sequence diagram for blood bank automation system)



### SECTION-C

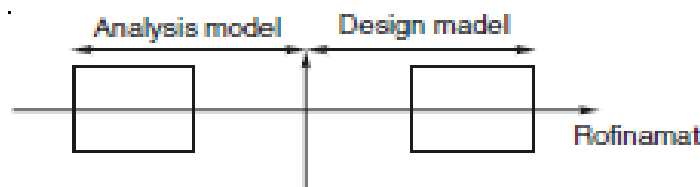
**Q.6. (a) Differentiate between design model and implementation model of a system. What are the main activities that are undertaken to arrive design model from the analysis model, explain briefly. (7.5)**

**Ans. Design Model:** It is mode where each object will be fully specified. The analysis model is refined and formalized to get a design model. It adopt to the actual implementation environment. This means that analysis model is adopted to fit in the implementation model at the same time as we refine it, because we want that writing the software becomes easier but changes cannot be avoided. So a new model is developed.

**Implementation Model:** This model consists of the source code of the specified objects in the design model. It is desirable that a block can be easily translated into the actual object module. In a smooth implementation environment, this is typically done. It consists of the annotated source code. Here OOP language is not required i.e., the technique may be used with any programming language to obtain on OO structure of the system. This it is strongly desirable to have an easy match between a block and the actual object modulule.

#### Main activities involved to arrive at design model from analysis module

The transistion from the analysis model to the design model should be made when the consequences of the implementation environment start.



- (i) The transition from analysis to design model should be made for each specific application.
- (ii) The analysis model is viewed as a conceptual and logical mode of the system.

**Thus the steps included in it are.**

- The changes in the view of design model into an abstraction of the source code to be written latter.
- The design model should be drawn in the way how the source code should be structured, managed and written.
- The objects or blocks to the module concept used in the programming language which are implemented are drawn.

The block is as follows.  Block.

- Blocks relates to the actual code.
- The communication between the blocks are described by stimuli.
- A stimulus is sent from one block to another.
- Interaction diagram is used to describe a sequence of stimuli.

**Q.6. (b) What are the parameters on the basis of which quality of object oriented design can be evaluated? Explain brief. (5)**

**Ans.** Quality of object oriented design can be evaluated by considering the attributes for functionality, reliability, maintainability and portability as the first set of quality attributes.

This the quality of attributed are recognized by.

(i) **Reusability:** This characteristic describes possibility of using designing for new issues without imposing important changes.

(ii) **Flexibility:** This characteristic described possibility to change design. It is ability to design which caused to adopt software for providing related work.

(iii) **Understandability:** The design characteristics caused to ease of easy understanding. It relates directly to the complexity of the design structure.

(iv) **Usage:** The function which is allocated to class and will be given through class public interfaces.

(v) **Extendibility:** It refers to presence and use of properties for current design that facilities exerting of new requirements to design.

(vi) **Efficiency:** This characteristic refers to the ability which is product and show desirable usage by concepts and object-oriented techniques.

Design properties are clear concepts which can be specified by testing internal and external structure, communications and function of compounts, methods of attributes and classes. Evaluation of class definition by considering to their inherited relation with other classes and considering component attributes and their inner methods show main information about structural characteristics and class application and its objects.

Attributes of abstraction, encapsulation, coupling and cobesion are also counted as structural characteristics and object-oriented as well.

**Q.7. (a) Differentiate between state based testing and specification testing. (5)**

**Ans. State based Testing:** Each feature is considered as a mapping from its starting or input states to its resultant or output states affected by any stimuli. Substates are defined which are the values of a data item at a specific point in time. These are then analyzed for specific and general values. Next, the set of states that the  $I^{\text{th}}$  feature actually accepts as input ( $I_i$ ) and the set of states it is able to generate as ( $O_i$ ) are determined for all the features of the class. One test case should allocate one substrate per data item.

**Specification Testing:** In this testing the test case specification identifier is the name of the test cases, used to distinguish it from other test cases. Conventions such as naming the test cases from the features or the component being tested allow developers to more easily refer to test cases. The test case specification lists all the components under test and the features to be exercised. All the Input specifications i.e. (lists the input required for the test cases) and output specifications i.e., (lists of expected output this output is computed manually or with a competing system) are computed so that proper outputs for proper inputs are occuring in the system. It also includes the lists of hardware and software platform needed to execute the test, including any test drives or stubs.

**Q.7. (b) Differentiate between traditional software development bycycle (SDLC) with object oriented SDLC and discuss one of the prominent object oriented SDLC model. (7.5)**



**Ans.**

<b>Traditional Software development (SDLC)</b>	<b>Object Oriented SDLC</b>
<ol style="list-style-type: none"><li>1. Programs are written in a procedural manner.</li><li>2. Functions are used</li><li>3. Lesser security</li><li>4. Sequential (batch-like) programs are written</li><li>5. Non resability of code.</li></ol>	<ol style="list-style-type: none"><li>1. Data is given a higher priority.</li><li>2. Classes are used</li><li>3. More secure as unauthorized access of data is prevented.</li><li>4. Classes that encapsulate data + functions together are written.</li><li>5. More resability of code.</li></ol>

**Object Oriented SDLC Model:**

First of all functional and non-functional requirements are identified.

Then OOA steps are applied to design a system.

**Step 1.** Identify Actors

**Step 2.** Develop a business process model.

**Step 3.** Identify use cases.

**Step 4.** Develop Interaction diagram to identify classes.

**Step 5.** Develop static UMH class diagrams.

- Identify classes
- Identify relationships
- Identify attributes
- Identify methods.

**Step 6.** Iterate and refine the whole design. If required again the step 1 through step 3 is repeated.

**SECTION-D**

**Q.8. (a) List out the techniques for requirements elicitation and discuss two prominent techniques for suitable examples. (5)**

**Ans.** The Requirement Elicitation process are as follows:

(a) Interviews, (b) Brainstorming, (c) FAST, (d) OFD, (e) Use Case

**(a) Interviews:** After receiving the problem statement from the customer, the first step is to arrange a meeting with the customer. During the meeting or interviews, both the parties would like to understand each other. The objective of conducting an interview is to understand the customer expectations from the software. Interviews are one of the most popular technique for understanding the problem domains and this technique is quite successful. Requirements engineer must be open minded and should not approach the interview with preconceived notions about what is required.

**(b) Brainstorming:** Brainstorming used in many business applications, is a group technique to promote creative thinking and can be used during requirements elicitation process to generate new ideas. It has become very popular and is being used by most of the companies. The more requirement that can be identified in the beginning, the better it is for the software development team. It is always easier to select from a long list of ideas or to create a new idea by combining lots of existing ideas. This group technique may be carried out with specialised groups like actual users, middle level managers etc., or with total stakeholders.

**(c) Fast (Facilitated Application Specification Technique):** This approach is similar to Brainstorming sessions and the objectives is to bridge the expectation gap- a difference between what developers think they are supposed to build and what customer think they are going to get.

This technique was developed specifically for collecting requirements. In order to reduce expectation gap, a team oriented approach is developed for requirements gathering and is called FAST. This list is presented in the session for discussion. Participants before starting a session also agree not to debate. After discussion some of the entries from the list is eliminated and new entries are also added to the list. This process is continued till a consensus is reached. The goal is to identify the problem, propose elements of the solution, negotiate different approaches and specify a preliminary set of solution requirements in an atmosphere that is conducive to the accomplishment of the goal.

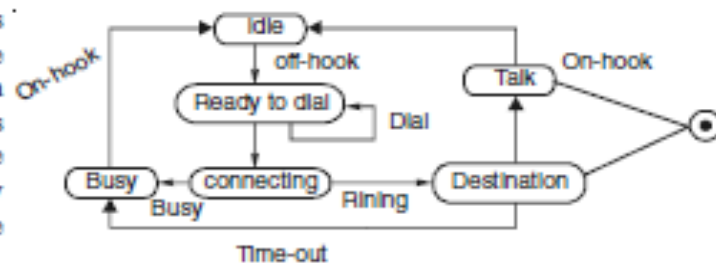
**(d) Use Case :** As requirement are gathered using FAST or QFD, the software engineer can create a set of scenarios that provide a description of how the system will be used. These scenarios are called as use-cases. To create a use case, the analyst must first identify the different types of people or devices that use the system/product. These actors actually represent roles that people play as the system operates.

This approach uses a combination of text and pictures in order to improve the understanding of requirements. The use cases describe "what of a system and not how". They only give functional view of the system. Use cases are structured outline or templates for the description of user requirement, modeled in a structured language like english. An actor or external agent, lies outside the system model, but interacts with it in some way.

**Q.8. (b) What is the significance of state chart diagram while designing the system? Explain structure of state chart diagram and illustrate with working example. (7.5)**

**Ans.** State chart diagram focuses on the changing state of a system driven by events. It describes the sequence, status conditions and appropriate response or actions to conditions during the life of the objects within the system.

**Example:** Here in this diagram various states are explained through the diagram how a connection of a call is established from the source to the destination. If the system is busy how it is identified as the connection is busy.



So, the sequence of the call establishment is described through the state chart diagram.

**State Chart Diagram Components are**

