END TERM EXAMINATION

SIXTH SEMESTER [B.TECH.]

Paper Code : ETCS - 304	*Subject : Object Oriented Software Engineering*
Time : 3 Hours	*Maximum Marks : 75*

Note : Q.1 is compulsory. Rest of the paper contains Four units, attempt One question from each unit.

Q 1) Attempt all the following questions:

(a) Define the term 'cohesion' in the context of object oriented design of systems?
(b) Do you need to develop all the views of the system? Justify your answer?
(c) Can integration testing and the unit testing be performed by the same test cases? Justify your answer?
(d) 'Inheritance often leads to less code', can this also leads to less testing? Explain with justification?
(e) State and justify whether the statement 'UML usage in software development will increase software reliability' is true or false?
(f) What are abstract Use-Cases?
(g) Which statement is most appropriate in OOSE?
- **Use case are identified through actors**
- **Actors are identified through use cases**

(h) Which new dimension is added in design model to the analysis model dimension and why?
(i) Name some commonly used requirement elicitation techniques?
(j) In UML, class diagram only addresses the static design view of a system? State True or False?

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?

Q 3) Write down the requirements for a typical hotel management system and draw use case diagram and class diagram?

Q 4) How do you relate sequence diagram and collaboration diagram? Explain with an example of Blood Bank Automation system?

Q 5) 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?

Q 6) List out the techniques for requirements elicitation and discuss two prominent techniques for suitable examples?

Q 7) Distinguish between implementation view and the environmental view of the system. How both of these views are modeled in UMW. Explain with suitable example?

Solution

Ans 1-a) In Object Oriented design cohesion means design element must be able to do its task on its own. It must generate minimal message to other objects. Hence for a better system the coupling between modules should be low and the cohesion between them should be high.

Ans 1-b) A system has several stakeholders. An ideal system will satisfy the needs of all the users. Users will be differentiated in several manners like educational profile, job profile, business interests etc. also most of the online systems will have anonymous users. So a system may be developed with the view of all the possible stakeholders.

Ans 1-c) No unit testing means testing a module in isolation and integration testing means testing the interface between two modules. So test cases in unit and integration testing must be different. Unit test cases must focus on working of a module where as integration testing's test cases must emphasize on good interface.

Ans 1-d) It is true that inheritance leads to less code but this will not necessarily reduce testing effort as now we will need to test the interface also. It is possible that a class in isolation may be working fine but after getting inherited by some other class or vice-versa can lead towards more errors due to poor interfacing.

Ans 1-e) UML usage will increase the software reliability as things; functions can be easily traced from one to another. Objects are traced from one model to another thus increasing the reliability. Similarly operations are traced from one model to another thus increasing the reliability.

Ans 1-g) Use cases are identified though actors as during system modeling first actors are identified and then based on their functionalities the use-cases are identified.

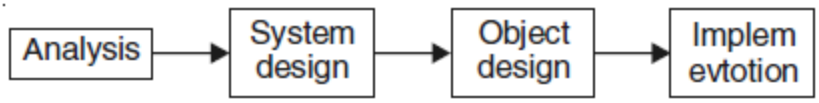
Ans 1-h) 'Implementation' dimension is added in design model to consider the effect of actual implementation environment for system modeling because analysis model consider the ideal environment.

Ans 1-i) Various requirements elicitation techniques are: Brainstorming, JAD Sessions, Group Discussions, QFD (Quality Function Deployment), Interviews, Surveys etc.

Ans 1-j)True, these diagrams show a set of classes, interfaces and collaborations and their relationships.

Ans 2) Five object oriented methods are explained below.

> ***Rambaugh's Modeling (OMT):*** OMT stands for Object Modeling Technique and is given by Rambaugh 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. If 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 represent 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.

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.

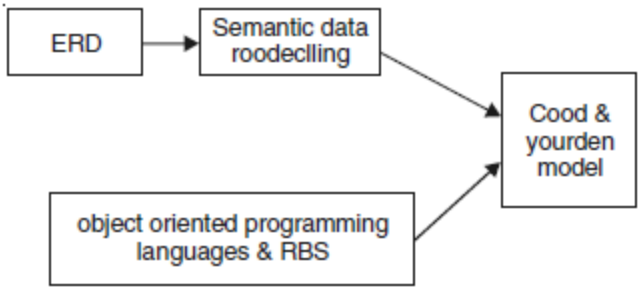
(i) At Design level: Class diagrams, object diagrams, state transition diagrams and interaction diagrams are at this level.

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

Coad 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, specialization 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



Shalaer and mellor's modeling: The OOA method consists of 2 parts:

(i) *Relationship Modeling of Data*: It introduces generatization 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. These 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. These 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 include data flow, data storages, process flow and control flow.

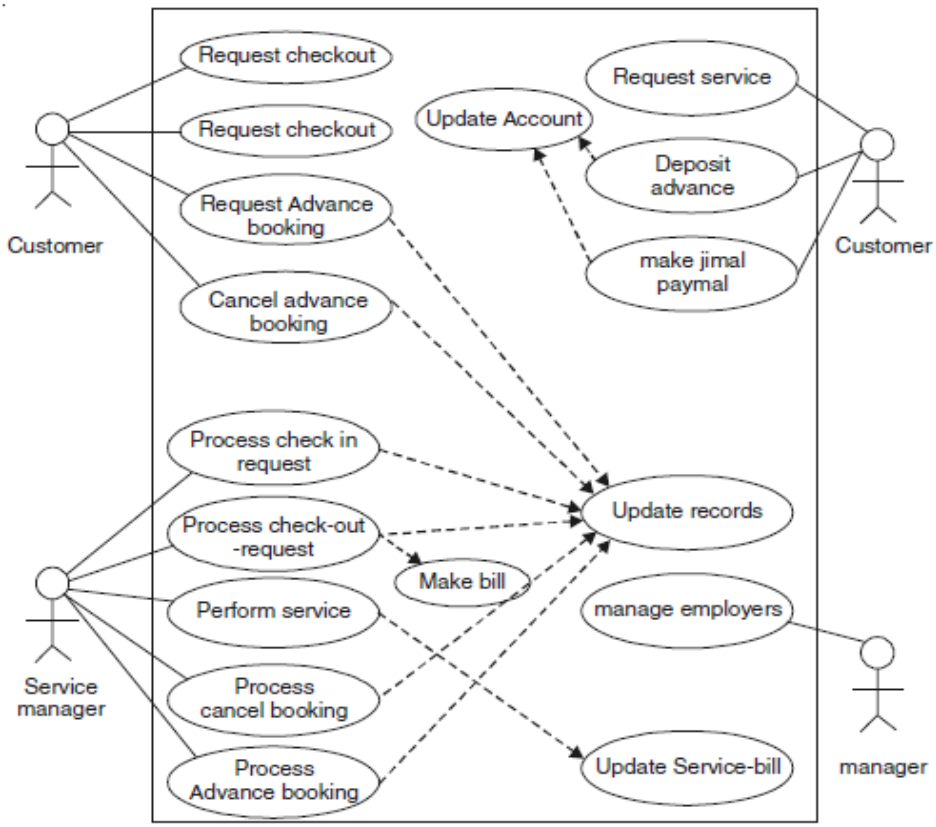
Coleman's Modelling (FUSION): The fusion method has been derived from Booch's OOD, Rambaugh's OMT and other formal methods. It has 3 specialization levels analysis (comceptisalmodel), design (interaction model) and implementation.

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.

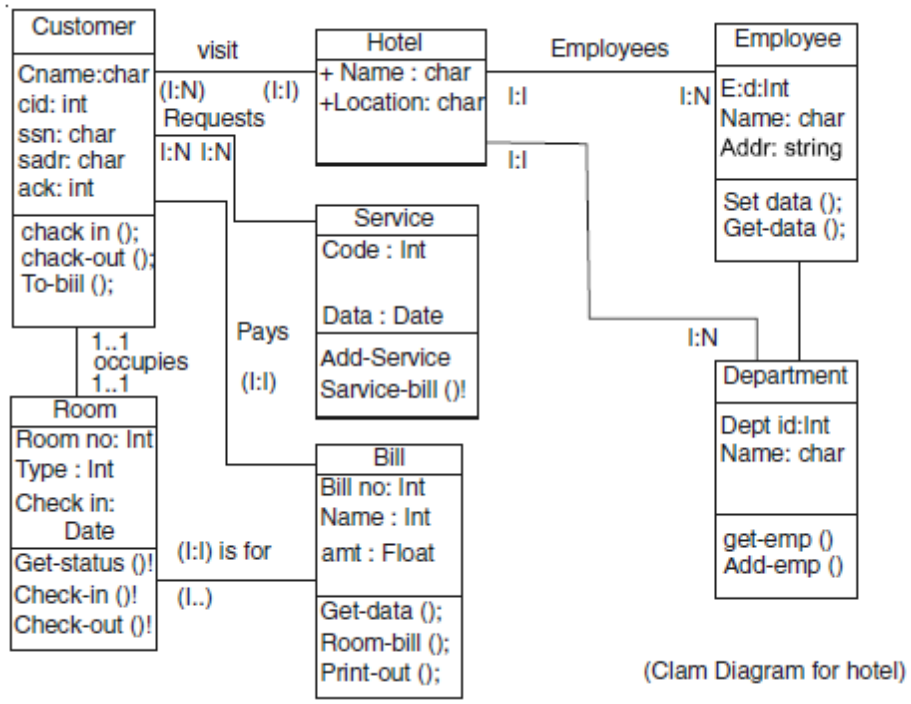
Interaction Design Model: It is used to describe the objects and the message exchanged between them. The focus of this model is on refining of the system operations rather than just mapping of the objects.

Ans 3) The requirements for a typical hotel management system are.

- Customer requirements
- Service Employee requirements
- Manager requirements
- Records requirements



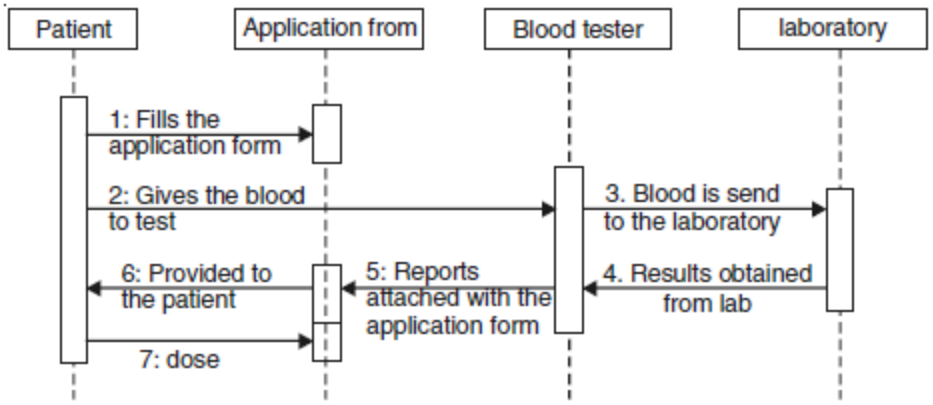
Use case Diagram for hotel management system



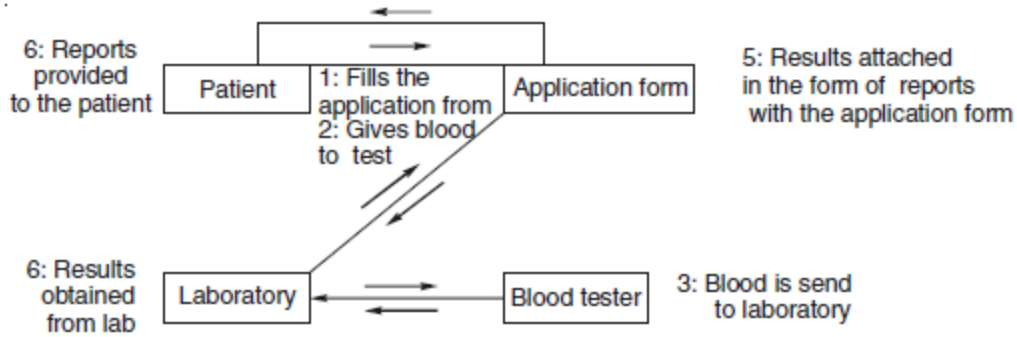
(Clam Diagram for hotel)

Ans 4) 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 organized 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)



(collaboration diagram for blood bank automation system)

Ans 5) 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 module.

Main activities involved to arrive at design model from analysis module

The transition 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.

Ans 6) The Requirement Elicitation processes are as follows:

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

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 techniques 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.

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 specialized groups like actual users, middle level managers etc., or with total stakeholders.

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.

Use Case: As requirements 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 test 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 is some way.

Ans 7) Environmental View: These models describe both the structural and behavioral dimensions of the domain or environment in which the solution is implemented. This view is often also referred to as the deployment or physical view.

Deployment diagrams: The models depicts and describe environmental elements and configuration of runtime processing components, libraries and objects that will reside on them.

Implementation View: The implementation view combines the structural and behavioral dimensions of the solutions realization or implementation. This view is often also referred to as the component or development views.

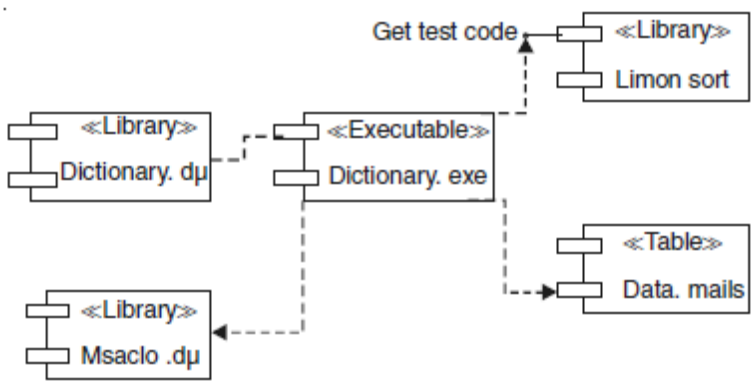
Component Diagrams: These depict the high level organization and dependencies of source code components, binary components and executable components and whether these components exist at compile, lions or runtime.

So, these two views are modeled in UML through the component diagrams and deployment diagrams.

Component diagram:

- Component diagram represents executable code.
- It is used to model the physical implementation of the software.

Example of a Component diagram:



Deployment diagram:

- It describes the physical esources of the system.
- It's focus is on the nodes on which the software will run.

Example of a Deployment diagram: