

OOAD ASSIGNMENT - 1

1. Explain the purpose of the unified modelling language

Unified Modeling Language aka UML is a general

purpose modelling language. The main aim of Uml is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

Uml is not a programming language, it is rather a visual language. We use Uml diagrams to portray the behavior and structure of a system.

UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object management Group - o.m.g adopted Unified Modelling Language as a standard in 1997. It's been managed by o.m.g ever since. International Organization for Standardization - ISO published Uml

as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

2. Explain significance of each UML diagrams.

Activity Diagram

Activity diagrams are probably the most important UML diagrams for doing business process modeling.

In software development, it is generally used to describe the flow of different activities and actions. These can be both sequential and in parallel. They describe the objects used, consumed or produced by an activity and the relationship between the different activities. All the above are essential in business process modeling.

Use Case Diagram

A cornerstone part of the system is the functional

requirements that the system fulfills. Use Case diagrams are used to analyze the systems high-level requirements. These requirements are expressed through different use cases

Sequence UML Diagram

Sequence diagrams are probably the most important UML diagrams among not only the computer science community but also as design-level models for business application development. Lately, they have become popular in depicting business processes, because of their visually self-explanatory nature.

Class Diagram

Class UML diagram is the most common diagram type for software documentation. Since most software being created nowadays is still based on the Object-Oriented Programming paradigm, using class diagrams to document the software turns out to be a common-sense

solution. This happens because OOP is based on classes and the relations between them.

When dealing with documentation of complex systems, component UML diagrams can help break down the system into smaller components. Sometimes it is hard to depict the architecture of a system because it might encompass several departments or it might employ different technologies.

Q Building blocks of uml

As Uml describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. The conceptual model of Uml can be mastered by learning the following three major elements -

Uml building blocks

Rules to connect the building blocks

Common mechanisms of Uml

Structural things define the static part of the model. They represent the physical and conceptual elements. Following are the brief descriptions of the structural things.

Class - Class represents a set of objects having similar responsibilities

Interface - Interface defines a set of operations, which specify the responsibility of a class..

Collaboration - Collaboration defines an interaction between elements.

Use case - Use case represents a set of actions performed by a system for a specific goal.

Component - Component describes the physical part of a system.

Node - A node can be defined as a physical element that exists at run time.

Define the following terms

Multiplicity:

It can be set for attributes, operations, and associations in a UML class diagram, and for associations in a use case diagram. The multiplicity is an indication of how many objects may participate in the given relationship or the allowable number of instances of the element.

UML Association

It is a structural relationship that represents objects can be connected or associated with another object inside the system. Following constraints can be applied to the UML Association relationship.

Aggregation

Aggregation relationship is denoted using a straight line with an empty arrowhead at one end.

Metadata modeling is a type of metamodeling used in software engineering and systems engineering for the analysis and construction of models applicable to and useful for some predefined class of problems.

Meta-modeling is the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for the modeling in a predefined class of problems.

Temporal Event

A temporal event is a time occurrence (direct or referential) which is meaningful to the business.

A temporal event may give rise to a precondition or a trigger.

UML models all of the above events as a simple trigger and has no precondition. You must use a non-UML method specification to model preconditions.

Concurrency is a property of a system in which several behaviors can overlap in time—the ability to perform two or more tasks at once. In the sequential paradigm, the next step in a process can be performed only after the previous has completed; in a concurrent system some steps are executed in parallel.

In UML Activity Diagrams you can fork the flow of control to more than one stream which will be executed simultaneously and you can join them again to unite (maybe before the end of the activity).

Composite States

are composed within the State Machine diagram by expanding a State element, adding Regions if applicable, and dragging further State elements, related elements and connectors within its boundaries. The internal State elements are then referred to as Substates.

Path

A communication path is association between two deployment targets, through which they are able to exchange signals and messages.

Q. Analyst approach to problem solving

As business analysis professionals and change agents, one of our most important skills is problem solving.

Problems present an opportunity to bring value to our customers and organization. Without the crucial skill

of problem solving, we are limited in our contribution to the organization as well as career growth. Many

people conduct problem solving intuitively simply by thinking without pausing to consider what the

underlying process and mechanism for problem solving is.

Problem solving is a discipline; a science of applying

logical and analytical techniques to identify the

underlying cause and recommend solutions that address

the root cause. Recommended 6 Stage Problem Solving

Approach The problem solving approach that Jatt uses

is a simple six stage process. The stages do not

need to be completed sequentially; the individual

stages may repeat and be completed in iterations.

The stages consist of:

Defining the problem statement

Defining scope, Elicit information and resolving ambiguity, Identifying associations and relationships, Root cause analysis, Solution proposal, etc.

The Problem Solving Process Start by creating the problem statement. The problem statement is a well-defined statement or question to frame the context. After you have a clear and unambiguous problem statement, define the scope of the effort.

The scope definition is probably the most important stage since it basically whether or not the problem can be solved satisfactorily. Scope is defined to apply constraints to the domain of consideration. When we have scope we know what to consider and what not to consider. Therefore, all possible solutions are directly dependant on the information within the scope. Once the scope is defined, you can move on to eliciting information and resolving ambiguity. Perform a stakeholder analysis and elicit information from all known stakeholders or sources as a basis for investigation.

You can use workshops, focus groups, interviews, document analysis, and other approaches to elicit information. When we elicit information, we try to remove ambiguity as ambiguity represents the unknown, liability, and risk. To reduce ambiguity, we need to consider the taxonomy of ambiguity to provide a frame of reference to how we will resolve it.

Ambiguity may be:

Missing information, Incorrect information, Duplicate information, Conflicting information, Incomplete information, etc....

The above provide a basis to ask questions concerning all information that is within scope, to challenge this information to be reliable and suitable for use. Context diagrams and domain diagram can help resolve ambiguity. Next, we identify associations and relationships to organize the information so we can derive meaning from it. Information needs to be structured, aligned, and associated that provides an additional level of meaning.

Q. What skills must a system analyst have?

Analytical

At the beginning of a project, a systems analyst has to understand the requirements of the client and study the various ways in which the requirements can be met. Once the various ways are found out, the onus is on the systems analyst to thoroughly analyze the pros and cons of every solution before deciding on a best feasible solution. The system analyst also has to analyze the ways and means of applying a solution which has been zeroed upon.

Technical Knowledge

A systems analyst must have ample technical skills because they have to help the technical team design a product which is to suit the customers needs.

He/She must have a working knowledge of the prevalent operating systems, programming languages and hardware platforms.

A systems analyst is expected to help in coding and debugging of a customized product. Further, he or she should also be able to tweak up the initial design based on the customer's feedback. The systems analyst is expected to constantly help in redesigning till the end product has been delivered to the client and has got the approval.

Man Management

Any systems analyst has to have great interpersonal skills and the ability to manage people. He or she has to coordinate with the different technical teams and get the product customized as per requirement delivered within the deadline agreed upon.

The systems analyst should ensure that no feathers have been ruffled and the whole team functions smoothly.

Efficient Planning and Execution

A systems analyst is expected to have a keen eye for detail and has to plan the various stages in which the project has to be executed.

Q Functional decomposition of information system

It is a term that engineers use to describe a set of steps in which they break down the overall function of a device, system, or process into its smaller parts. This is usually accomplished through thoughtful analysis and team discussions of project information and the result is a chart that describes the problem and or solutions in increasing detail.

It corresponds to the various functional relationships as how the original complex business function was developed. It mainly focusses on how the overall functionality is developed and its interaction between various components.

Large or complex functionalities are more easily understood when broken down into pieces using functional decomposition.

Q. Types of information system

An information system is a set of interrelated components that works together to collect, process, store and breakdown the information to support decision making.

1. Transaction Processing System:

It is information system that processes data resulting from the occurrences of business transactions

Their objectives are to provide transaction in order to update records and generate reports i.e to perform store keeping function

The transaction is performed in two ways: Batching processing and Online transaction processing.

Example: Bill system, payroll system, Stock control system.

Management Information System:

Management Information System is designed to take relatively raw data available through a Transaction Processing System and convert them into a summarized and aggregated form for the manager, usually in a report format. It reports tending to be used by middle management and operational supervisors.

Many different types of report are produced in mis. Some of the reports are a summary report, on-demand report, ad-hoc reports and an exception report.

Example: Sales management systems, Human resource management system.

Decision Support System - DSS:

It is an interactive information system that provides information, models and data manipulation tools to help in making the decision in a semi-structured and unstructured situation.

It comprises tools and techniques to help in gathering relevant information and analyze the options and alternatives, the end user is more involved

in creating DSS than an MIS.

Example: Financial planning systems, Bank loan management systems.

Experts System:

These include expertise in order to aid managers in diagnosing problems or in problem-solving. These systems are based on the principles of artificial intelligence research.

It is a knowledge-based information system. It uses its knowledge about a specificity are to act as an expert consultant to users.

Knowledgebase and software modules are the components of an expert system. These modules perform inference on the knowledge and offer answers to a users question

Q. Explain The Following

1. Unified process

The Unified Software Development Process or Unified Process is an iterative and incremental software development process framework. The best-known and extensively documented refinement of the Unified Process is the Rational Unified Process - RUP. Other examples are OpenUP and Agile Unified Process.

2. Joint application design

Joint Application Development (JAD) is a process that accelerates the design of information technology solutions. JAD uses customer involvement and group dynamics to accurately depict the user's view of the business need and to jointly develop a solution. Before the advent of JAD, requirements were identified by interviewing stakeholders individually.

The ineffectiveness of this interviewing technique, which focused on individual input rather than group consensus, led to the development of the JAD approach.

Q. Steps in conducting interview

- Prepare for the interview
- Understand the STAR interview process
- Describe the company and the position to the interviewee
- Explain the interview process to your interviewee
- Learn about the interviewee's career goals
- Ask questions related to the position
- Gather more information with follow up questions
- Take notes to help you remember important information
- Give candidates the chance to ask questions

Steps for Following Up in the Interview

- Review the class diagram and select the classes that will require state charts
- For each selected class in the group make a list of all status conditions
- Begin building state chart fragments by identifying the transitions that cause an object to leave the identified state
- Sequence the state transition combination in correct order
- Look for additional transitions and review the path and look for independent concurrent path
- Explain each transaction with the appropriate message event guard condition and action expression
- Review and test each state chart