Practical no: 1

Name: Saloni Vishwakarma
Batch-Roll no: C1-13
Subject: Software Engineering and Project Management Lab
Date of execution: 30 August 2023

Aim: To study Unified Modeling Language (UML) perspective and notations in StarUML.
To-do-task:
1. Study the installation steps for the open-source tool "StarUML"

As of my last knowledge update in September 2021, StarUML is a popular UML modeling tool. To download and install StarUML, follow these steps:
-Visit the StarUML Website
-Choose the Edition
-Download the Installer
-Run the Installer
-Accept the License Agreement
-Choose Installation Options
-Complete the Installation
-Launch StarUML
-Activate/Register (if required)
-Start Using StarUML

oft Bing

star uml

SEARCH CHAT IMAGES VIDEOS MAPS NEWS SHOPPING MORE

33,20,000 Results Date

StarUML
UML diagram software

Reviews Support Developer

staruml.io
https://staruml.io

StarUML

Web Compatible with UML 2.x standard metamodel and diagrams: Class, Object, Use Case, Component, Deployment, Composite Structure, Sequence, Communication, Statechart, ...

Downloads

Default Colors for UML Elements Auto Backup Word Wrap for ...

Docs

Docs - StarUML

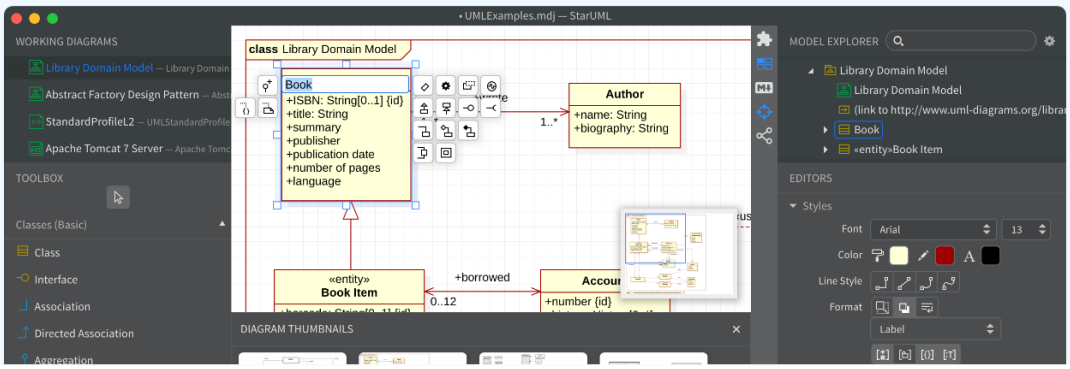
Download Buy Extensions Support Blog Docs

StarUML

A sophisticated software modeler for agile and concise modeling

Download for Windows
(Intel x86)

current version: v5.1.0



2. Study and research what is Unified Modeling Language (UML)

UML stands for "Unified Modeling Language." It is a standardized, general-purpose modeling language used in the field of software engineering and system design. UML provides a set of graphical notations and a methodology for visualizing, specifying, constructing, and documenting software systems and other complex systems.

UML is a powerful tool for communication and collaboration among software developers, architects, designers, and other stakeholders involved in software development projects. It allows them to create visual representations of system structure, behavior, and interactions, making it easier to understand, plan, and design complex systems.

3. Study the types of diagrams that belongs to UML

Unified Modeling Language (UML) includes several types of diagrams, each designed to represent different aspects of a system or software application. These diagrams help software developers, architects, and other stakeholders visualize, specify, and document various aspects of a system. Here are some common types of UML diagrams:

Class Diagram:
-Represents the static structure of a system.
-Shows classes, their attributes, methods, and relationships between classes.
-Used for designing and modeling the object-oriented structure of a software system.

Use Case Diagram:
-Describes the interactions between external actors (users or systems) and a system.
-Shows the different use cases or functionalities of the system.
-Helps in understanding system requirements and user interactions.

Sequence Diagram:
-Illustrates how objects interact with each other over time.
-Represents the dynamic behavior of a system by showing the sequence of messages exchanged between objects.
-Useful for modeling scenarios and understanding the flow of control.

Activity Diagram:
-Represents the workflow or flowchart-like behavior of a system.
-Used to model business processes, system processes, or the behavior of a single object.
-Shows the flow of activities and decisions.

State Machine Diagram:
-Represents the states and state transitions of an object or system.
-Shows how an object or system responds to events and changes in its state.
-Useful for modeling the behavior of complex systems.

Component Diagram:
-Represents the physical components (e.g., classes, libraries, and executables) of a system.
-Shows the dependencies between components.
-Useful for understanding the high-level architecture of a software system.

Deployment Diagram:
-Represents the physical deployment of software components on hardware nodes.
-Shows the relationships between software components and the hardware they run on.
-Useful for system deployment and infrastructure planning.

Package Diagram:
-Organizes and represents the structure of packages and their relationships in a system.
-Helps manage the organization of elements in a larger system.
-Useful for modularization and managing dependencies.

Object Diagram:
-Represents a specific instance of a class diagram.
-Shows objects and their relationships at a particular point in time.
-Useful for debugging and understanding specific system states.

Communication Diagram (formerly Collaboration Diagram):
-Focuses on object interactions and the messages exchanged between objects.
-Similar to a sequence diagram but emphasizes the relationships between objects.
-Useful for visualizing object interactions in a specific scenario.

Interaction Overview Diagram:
-Combines elements of activity and sequence diagrams to provide a high-level view of interactions.
-Useful for showing complex interactions at a high level.

Timing Diagram:
-Represents the behavior of objects with respect to time.
-Shows when events occur and the duration of activities.
-Useful for real-time systems and performance analysis.

4. To design the use case diagram for ATM machine.