**PART A**

**EXPERIMENT NO. 1**

**A.1 Aim: - Learning the Strum environment.**

**A.2 Prerequisite**

**1. Concepts of Software Engineering, Model**

**A.3 Outcome**

After successful completion of this training students will be able to use Star UML for creating the UML model with respect to given case study.

Design solution using unified modeling language.

**A.4 Task:**

1. Download Star UML software in your machine and Prepare necessary documentation for doing hands on with Star UML.
2. Select the appropriate case study to design UML diagrams.

**PART B**

|  |  |
| --- | --- |
| Roll No: B017 | Name: Niharika Dalal |
| Class: B. Tech. Comp. Engg. | Batch: B1 |
| Date of Experiment: 4.1.2016 | Date of Submission: 11. 1.2016 |
| Grade: A+ |  |

**B.1 About Star UML**

**………………………………………………………………………………………**

## Documentation

Documentation describes the concepts of tool but on high level vision. A more detailed documentation is available for the diagramming functions. Besides English, documentation exists in Korean, Japanese and Russian.

## Configuration

Some general and diagram configurations options are available from the Tools/Option menu. You will find in this window also the configuration switches for the code generation. The interface is also very configurable as you can select what part of the tool you would like to view or not.

## Features

When you start a new project, StarUML proposes which approach you want to use:

* 4+1 (Krutchen)
* Rational
* UML components (from Cheesman and Daniels book)
* default or empty

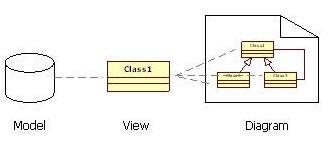
Depending on the approach, profiles and/or frameworks may be included and loaded. If you don't follow a specific approach, the "empty" choice could be used. Although a project can be managed as one file, it may be convenient to divide it into many units and manage them separately if many developers are working on it together.

StarUML makes a clear conceptual distinction between models, views and diagrams.

Model: an element that contains information for a software model.

View: a visual expression of the information contained in a model.

Diagram: a collection of view elements that represent the user’s specific design thoughts.



StarUML is built as a modular and open tool. It provides frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta-model and tool through COM Automation, and it provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

StarUML supports the following diagram types

* Use Case Diagram
* Class Diagram
* Sequence Diagram
* Collaboration Diagram
* State-chart Diagram
* Activity Diagram
* Component Diagram
* Deployment Diagram
* Composite Structure Diagram

The user interface is intuitive. During diagram editing, "wizards" are located around the object that give you the quick shortcuts to main associated tasks with your current operation, like adding an attribute when you create a class for instance.

StarUML has also a model verification feature. You can export diagram in different formats (jpg, bmp, wmf). It also supports a patterns approach and import of Rational Rose files.

StarUML Generator is platform module to generate various artefacts (like as Microsoft Word, Excel, PowerPoint, and Text-based artefacts) by templates depending on UML model elements in StarUML.

The users can define their own templates and can apply many different kinds of templates to the same UML model, so the users can get various artefacts automatically, easily and fast. The tool supports code generation and reverse engineering for Java, C# and C++.

**Conclusion**

StarUML has many powerful features and is certainly more than a "simple" diagramming tool. With its support of MDA (Model Driven Architecture), it is more aimed at people using UML in an intensive way and with some code generations objectives than for simply drawing diagrams to document requirements. However, using StarUML just as a diagramming tool work fine, especially on Windows as the tool is built with Delphi and might execute faster than the Java-based tools.

## System Requirements

The following are the minimum system requirements for running StarUML™.

* Intel® Pentium® 233MHz or higher
* Windows® 2000, Windows XP™, or higher
* Microsoft® Internet Explorer 5.0 or higher
* 128 MB RAM (256MB recommended)
* 110 MB hard disc space (150MB space recommended)
* CD-ROM drive
* SVGA or higher resolution monitor (1024x768 recommended)
* Mouse or other pointing device

## Key Features

StarUML™ has the following new features.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Accurate UML standard model | It adheres to the UML standard specification specified by the OMG for software modelling.  StarUML maximizes itself to order UML 1.4 standard and meaning, and it accepts UML 2.0 notation on the basis of robust meta model**.** |
| Open software model format | Unlike many existing products that manage their own legacy format models inefficiently, manages all files in the standard XML format.  Codes written in easy-to-read structures and their formats can be changed conveniently by using the XML parser. |
| True MDA support | StarUML truly supports UML Profile. This maximizes extensibility of UML, making modelling of applications possible even in areas like finance, defence, e-business, insurance, and aeronautics. |
| Applicability of methodologies and platforms | StarUML™ manipulates the approach concept, creating environments that adapt to any methodologies/processes. Not only the application framework models for platforms like .NET and J2EE, but also basic structures of software models (e.g. 4+1 view-model, etc.) can be defined easily. |
| Software model verification function | Users can make many mistakes during software modelling. Such mistakes can be very costly if left uncorrected until the final coding stage. In order to prevent this problem, StarUML™ automatically verifies the software model developed by the user, facilitating early discovery of errors, and allowing more faultless and complete software development. |
| Useful Add-Ins | StarUML™ includes many useful Add-Ins with various functionalities: it generates source codes in programming languages and converts source codes into models, imports Rational Rose files, exchanges modelling information with other tools using XMI, and supports design patterns. |

**……………………………………………………………………………………….**

**B.2 Problem Statement of the Case Study Selected**

**………………………………………………………………………………………**

We have selected a case study of a mobile application for a bakery shop. We intend to make an android application for a bakery which will allow the user to select from a variety of products available in the application. The app will also allow the user to add items to a cart, where they will be stored temporarily on a database; and when these items are ordered, the cart as well as the database will be emptied.

The basic intention is to create android application using Java which will help to understand the object-oriented concept of software engineering which, makes use of classes and objects.

**……………………………………………………………………………………….**

**B.3 Conclusion**

**………………………………………………………………………………………**

StarUML is a modular and open tool. It provides frameworks for extending the functionality of the tool. It allows access to all functions of the model and tool and provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies.

In this practical, we have studied the working and use of StarUML, a software used to create diagrams in object oriented software engineering.

**……………………………………………………………………………………….**