INTRODUCING THE UML

LAB MANUAL 1
Lab Objective: Learn to Know

- What is UML
- Why the UML is necessary
- How to represent UML components in diagrams

Tool used: MS Visio 2003

Lab Description: The Unified Modeling Language (UML) is a visual modeling language that enables system builders to create blueprints that capture their visions in a standard, easy-to-understand way, and provides a mechanism to effectively share and Communicate these visions with others. The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model. UML model describes what a system is supposed to do. It doesn’t tell how to implement the system.

Class Diagram:

A class is a category or group of things that have the same attributes and the same behaviors. A rectangle is the icon that represents the class. It’s divided into three areas. The uppermost area contains the name, the middle area holds the attributes, and the lowest area holds the operations. i.e

Anything in the class of washing machines has attributes such as brand name, model, serial number, and capacity. Behaviors for things in this class include the operations “accept clothes,” “accept detergent,” “turn on,” and “turn off.”

In UML, a multiword classname has initial capital letters for all the words and eliminates white space between each word (for example, WashingMachine). Attribute names and operation-names follow the same convention, but the first letter of the first word isn’t capitalized (for example, acceptClothes()).

Object Diagram:

An object is an instance of a class—a specific thing that has specific values of the class’s attributes. The icon is a rectangle, just like the class icon, but the name is underlined. In
the icon on the left, the name of the specific instance is on the left side of a colon, and the name of the class is on the right side of the colon. The name of the instance begins with a lowercase letter. It’s also possible to have an anonymous object, as the icon on the right of Figure shows. This just means that you don’t supply a specific name for the object, although you do show the class it belongs to.

myWasher:WashingMachine :WashingMachine

Use Case Diagram:

A use case is a description of a system’s behavior from a user’s standpoint. For system developers, the use case is a valuable tool: It’s a tried-and-true technique for gathering system requirements from a user’s point of view. Obtaining information from the user’s point of view is important if the goal is to build a system that real people can use.

The little stick figure that corresponds to the washing machine user is called an actor. The ellipse represents the use case. The actor—the entity that initiates the use case—can be a person or another system. The use case is inside a rectangle that represents the system, and the actor is outside the rectangle.

State Diagram:

At any given time, an object is in a particular state. This transition from one state to the next state is represented by state diagram.i.e A person can be a newborn, infant, child, adolescent, teenager, or adult. An elevator is either moving or stationary. A washing machine can be either in the soaking, washing, rinsing, spinning, or off state. The symbol at the top of the figure represents the start state and the symbol at the bottom represents the end state.
Communication Diagram:

Both the sequence diagram and the communication diagram show interactions among objects. For this reason, the UML refers to them collectively as interaction diagrams.

Rather than represent time in the vertical dimension, this diagram shows the order of messages by attaching a number to the message label.
Sequence Diagram:

Objects interact with one another, and these interactions occur over time. The UML sequence diagram shows the time-based dynamics of the interaction.

![Sequence Diagram]

Each arrow represents a message that goes from one object to another. Time, in this diagram, proceeds from top to bottom. So the first message is timeSoak(), which the timer sends to itself. The second message is sendWater(), which the timer sends to the water pipe. The final message, stopRotating(), goes from the timer to the drum.

Activity Diagram:

The activities that occur within a use case or within an object’s behavior typically occur in a sequence is represented by the activity diagram.
Deployment Diagram:

The UML deployment diagram shows the physical architecture of a computer based system. It can depict the computers, show their connections with one another, and show the software that sits on each machine. Each computer is represented as a cube, with interconnections between computers drawn as lines connecting the cubes.

Notes:

It often happens that one part of a diagram doesn’t present an unambiguous explanation of why it’s there or how to work with it. When that’s the case, the UML note is helpful. Think of a note as the graphic equivalent of a yellow sticky. Its icon is a rectangle with a folded corner. Inside the rectangle is explanatory text.
Composite Structure Diagram:

Each component class inside the whole is called the composite structure diagram. i.e a person consists of a mind and a body. It consists of lines and symbols that join the Person class to the Mind class and to the Body class.

Timing Diagram:

Timing diagram is designed to show how long an object is in a state.

Interaction Overview Diagram:

If you replaced some of the activities with sequence diagrams or communication diagrams (or a combination of the two), you’d have UML 2.0’s new interaction overview diagram.
Assignment 1:

1) Define model. Explain the importance of modeling?
2) Explain briefly four basic principles of modeling?
3) What are the main differences in solving a problem in object oriented way and traditional (algorithm/structured) way?
4) Why do you prefer object oriented modeling over traditional (algorithmic) modeling?
5) What are the four main aims of UML. Explain them briefly?
6) Why is it necessary to have a variety of diagrams in a model of a system?

Lab performed on (date): __________ Reg #: _____________

Checked by: ________________________ Date: ______________

Marks Awarded: ____________