NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

CS3005 – Software Design & Architecture Lab

LAB Instructors: Sobia Iftikhar "Sobia.iftikhar@nu.edu.pk"

Lab 10

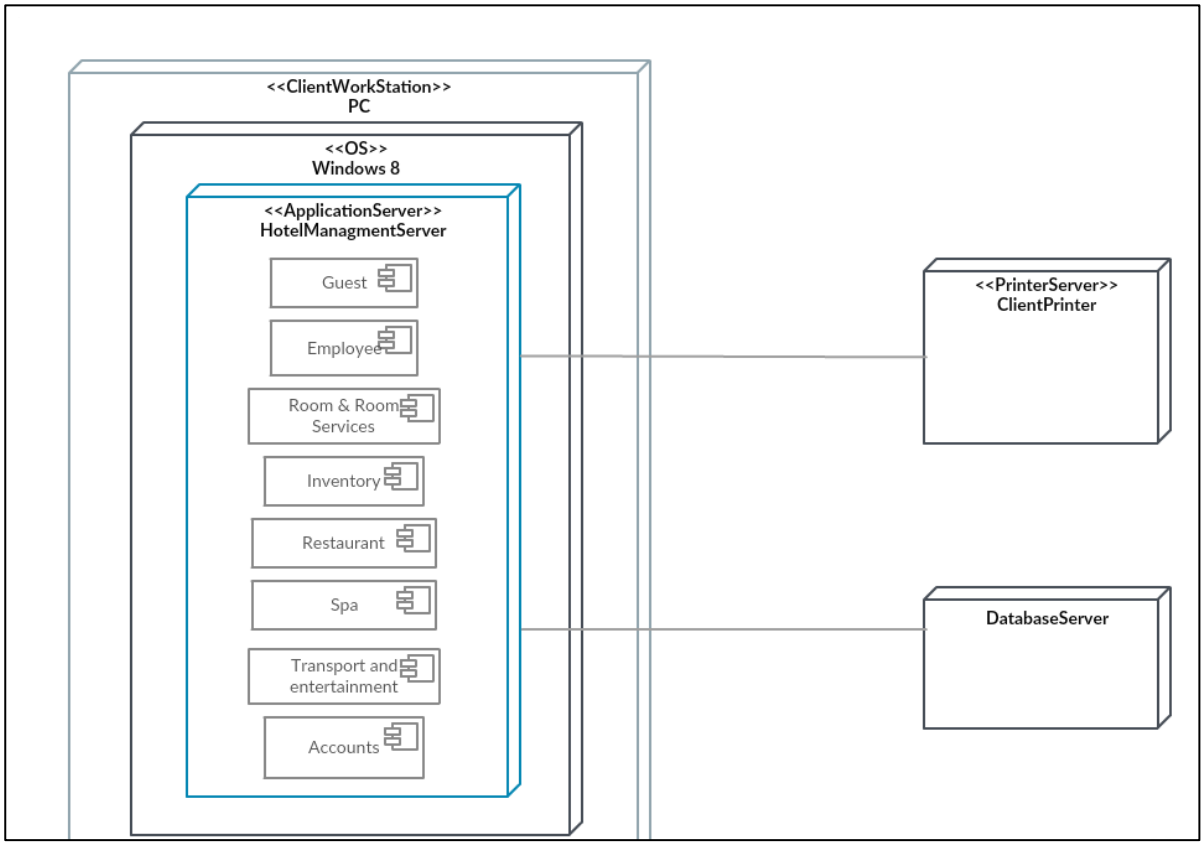
Objective:

1. **Deployment Diagram**
2. **State Diagram**
3. **Exercise**

Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

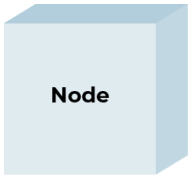


Deployment Diagram Notations

In order to draw a deployment diagram, you need to first become familiar with the following deployment diagram notations and deployment diagram elements.

1. Node

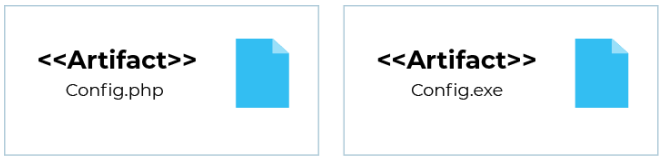
A node is the main tool in the deployment diagram. It describes the execution of the codes and how the system components interact with one another. A node can be a software or hardware object. Nodes house and execute the artifacts. Mostly, a node is a physical entity.



Nodes have two main types:

- A node can be a physical machine that performs the code computations. This type of node is usually a device, such as a personal computer or an external server.
- A node can be an environment that performs the code execution. People often call this an execution environment node. An example is the JVM or Java virtual machine.

2. Artifact



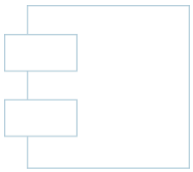
This notation depicts the real software product the developers are working on. The symbol for an artifact is a rectangle with the label “artifact” and the artifact name. They don’t stand on their own; instead, they’re generally deployed on a node.

You can use artifacts to represent…

- an executable file
- a framework used during software development
- a source file
- an output file
- documentation

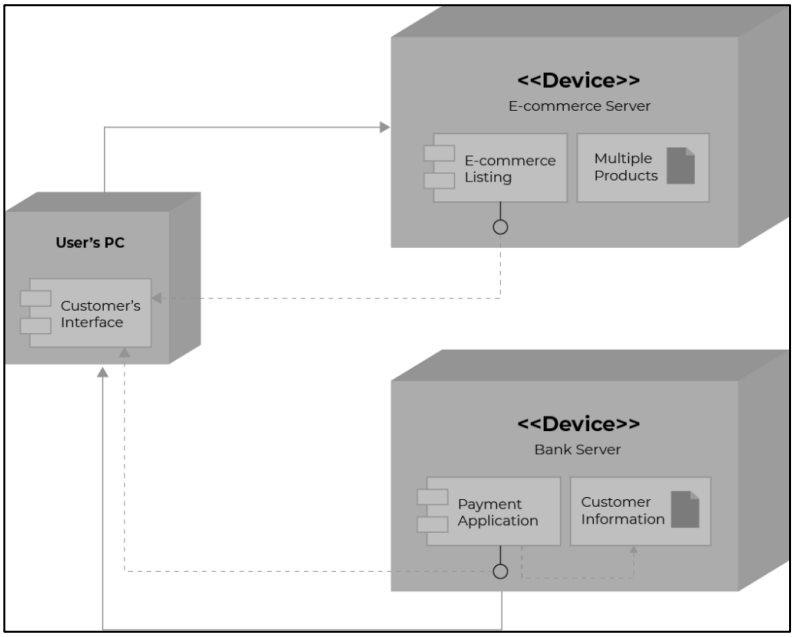
3. Component

This notation represents other software elements present in the system. The symbol for the component notation is a rectangle with two tabs. For instance, a bank application running on an Android device is a component of the node (Android device).



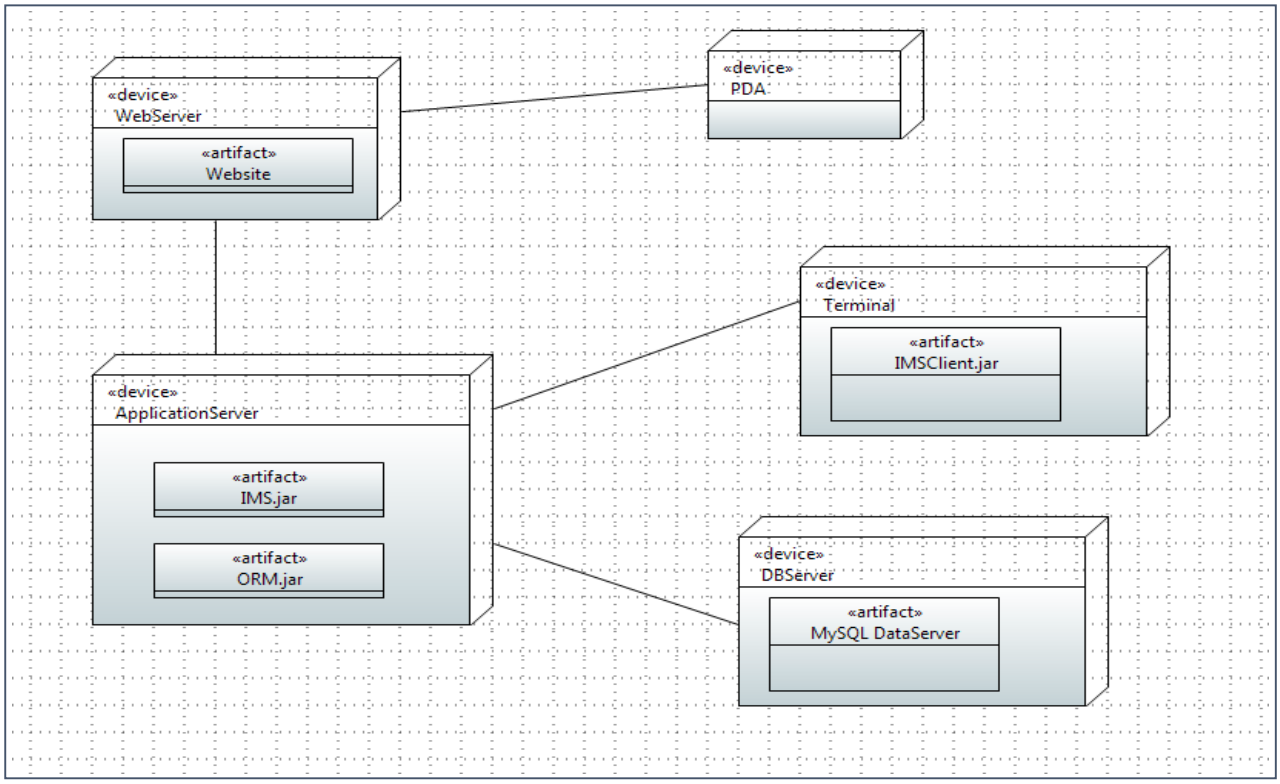
Examples:

This example shows how an e-commerce platform receives payment from a customer for a product. There's an interaction between the user's PC, the e-commerce server, and the bank server to get the payment done. You can see three nodes: the user's PC, the bank server, and the e-commerce server.



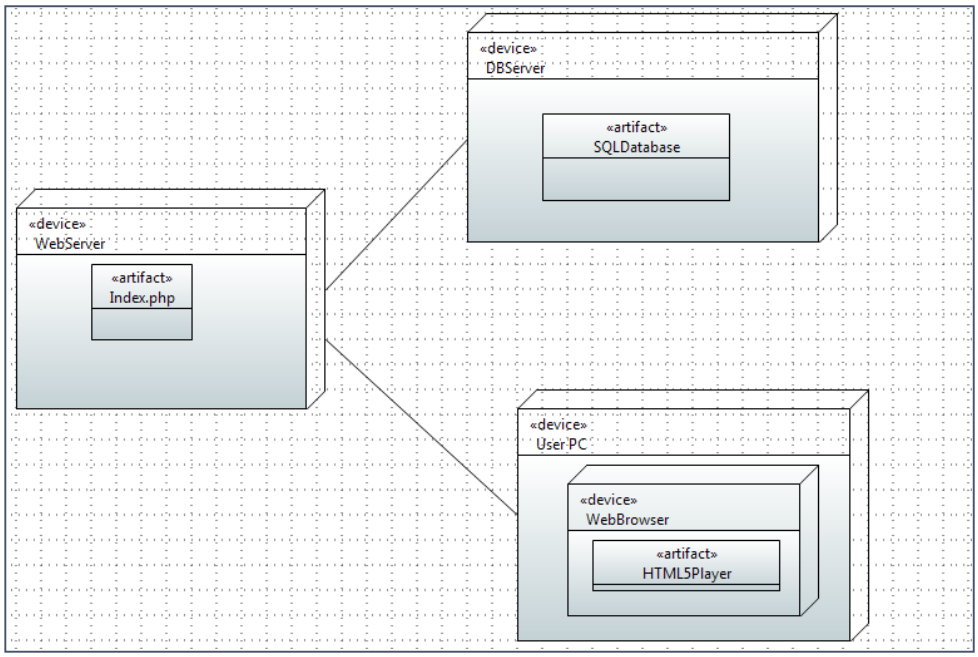
Example 02

The example shows the topology of a human resources system, which follows a classical client/server architecture



Example 03

Following deployment diagram represents the working of HTML5 video player in the browse.



State chart Diagram

They define different states of an object during its lifetime. States are defined as a condition in which an object exists and it changes when some event is triggered
These states are changed by events.

So the most important purpose of Statechart diagram is to model life time of an object from creation to termination. Statechart diagrams are useful to model reactive systems Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state.

Purpose

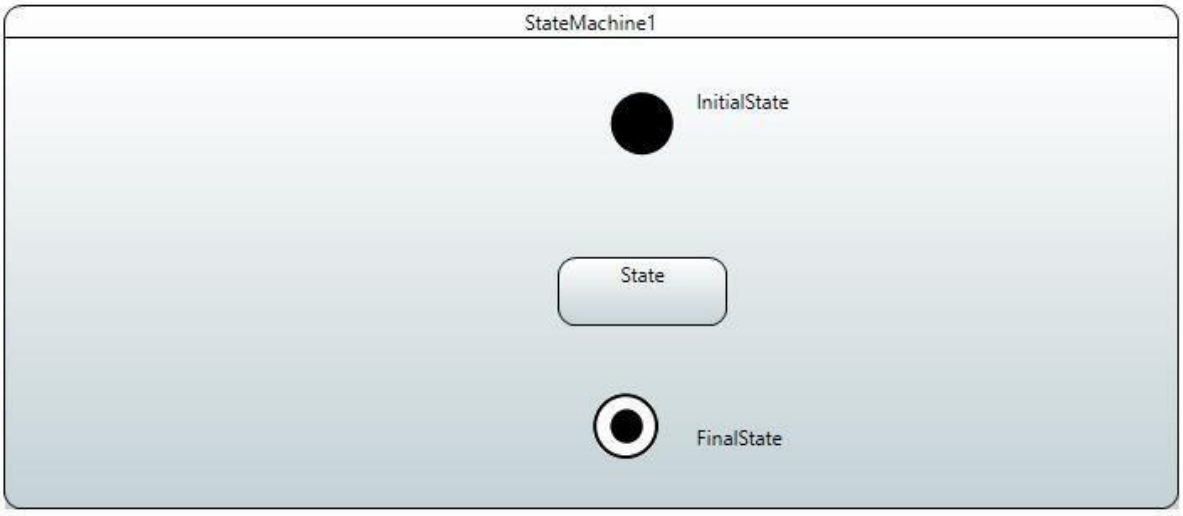
1. Following are the main purposes of using Statechart diagrams To model dynamic aspect of a system.
2. To model life time of a reactive system.
3. To describe different states of an object during its life time. Define a state machine to model states of an object.

How to draw state charts

Before drawing a Statechart diagram we must have clarified the following points: Identify important objects to be analyzed.

Identify the states. Identify the events.

Elements of state chart diagrams:



Initial state

The initial state symbol is used to indicate the beginning of a state machine diagram.

State box

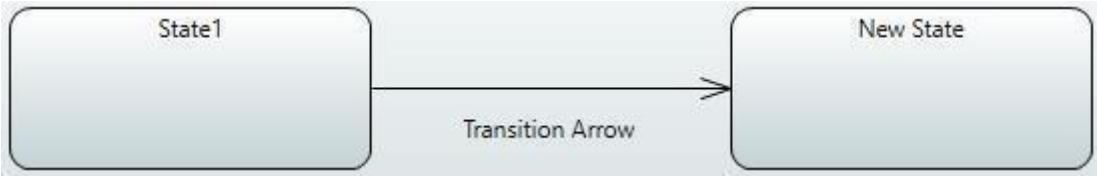
It is a specific moment in the lifespan of an object. It is defined using some condition or a statement within the classifier body. It is used to represent any static as well as dynamic situations.

Final state

This symbol is used to indicate the end of a state machine diagram.

Transition

A transition is a change in one state into another state which is occurred because of some event. A transition causes a change in the state of an object.

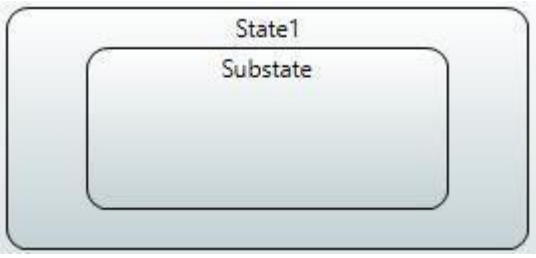


Trigger

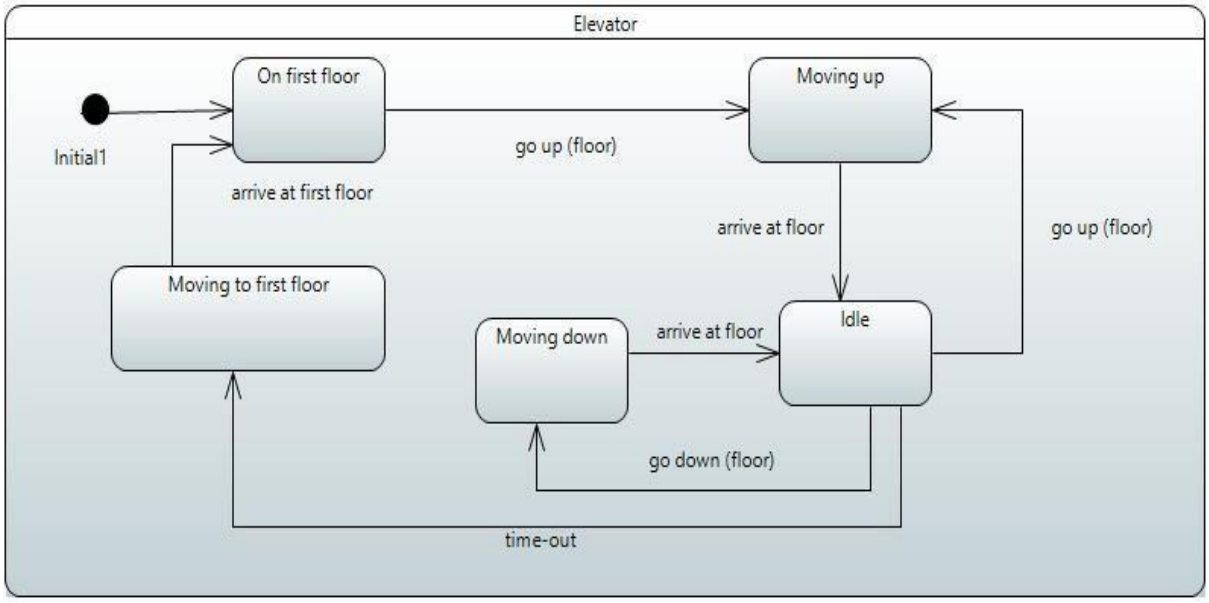
A type of message that actively moves an object from state to state, written above the transition arrow.

Substate

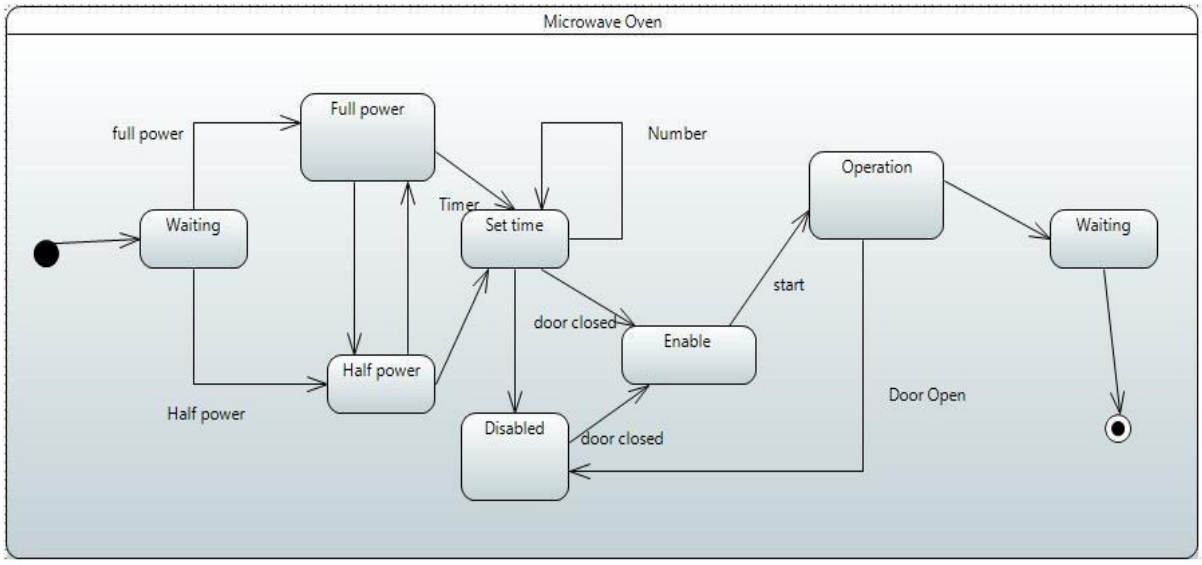
A state contained within a composite state's region.



Elevator state chart diagram example

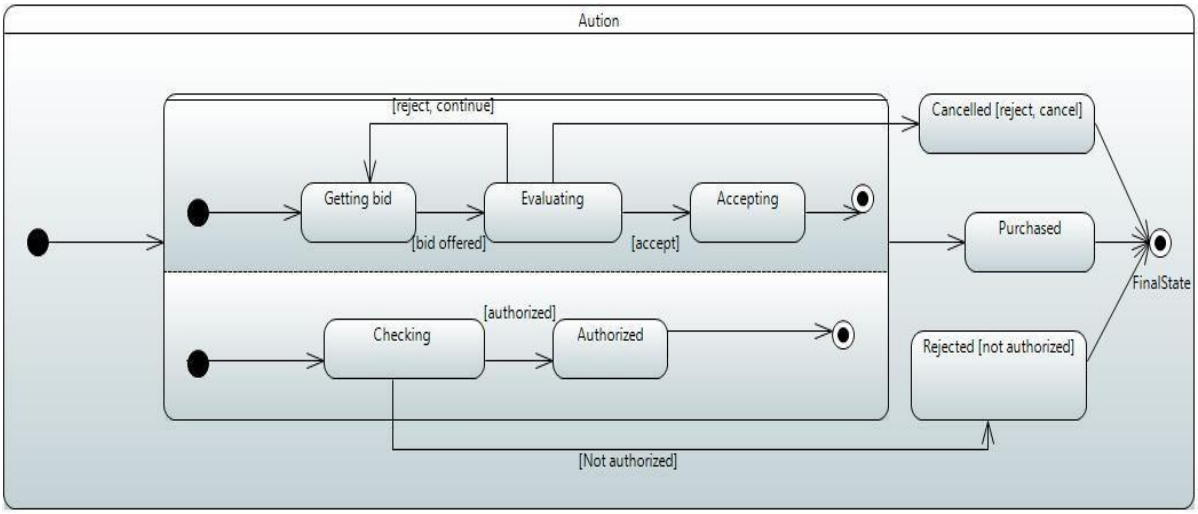


Microwave oven state diagram example



concurrent State Machine Diagram Example — Auction Process

In this example, the state machine first entering the Auction requires a fork at the start into two separate start threads. Each substate has an exit state to mark the end of the thread. Unless there is an abnormal exit (Canceled or Rejected), the exit from the composite state occurs when both substates have exited.



University state diagram example

This state diagram shows the process of enrollment and classes at a university. The composite state "Enrollment" is made up of various substates that will lead students through the enrollment process.

Once the student has enrolled, they will proceed to "Operational state" that is Teaching "Final exams" and finally to "Vacation