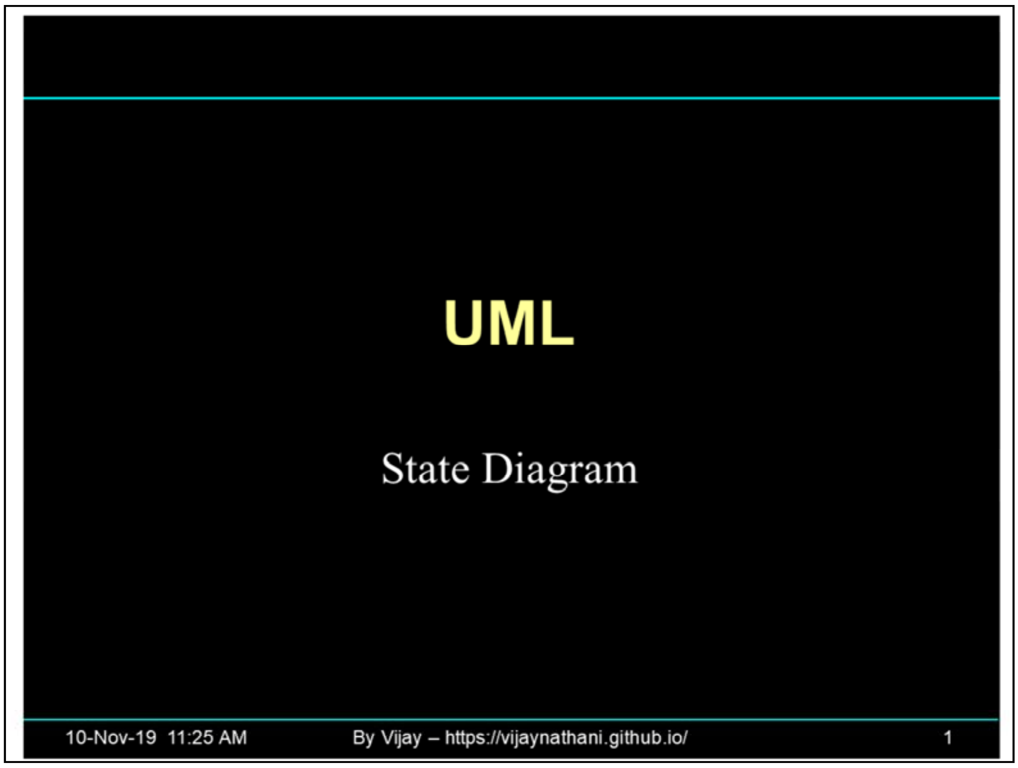
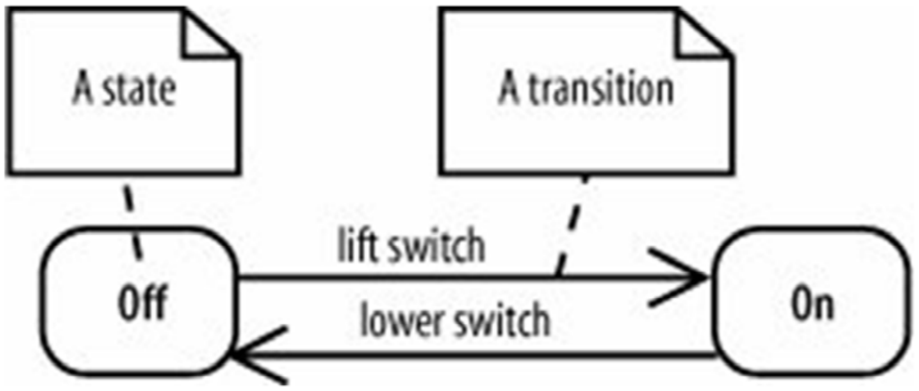


UML

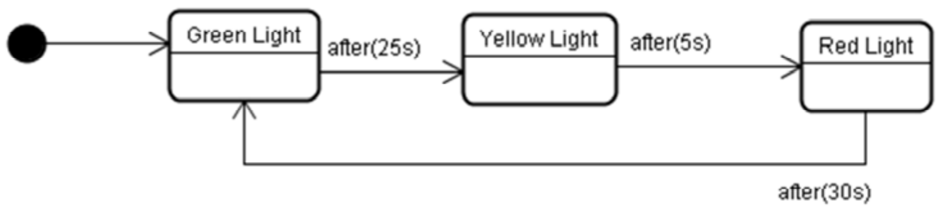
State Diagram

10-Nov-19 11:25 AM By Vijay – https://vijaynathani.github.io/ 1

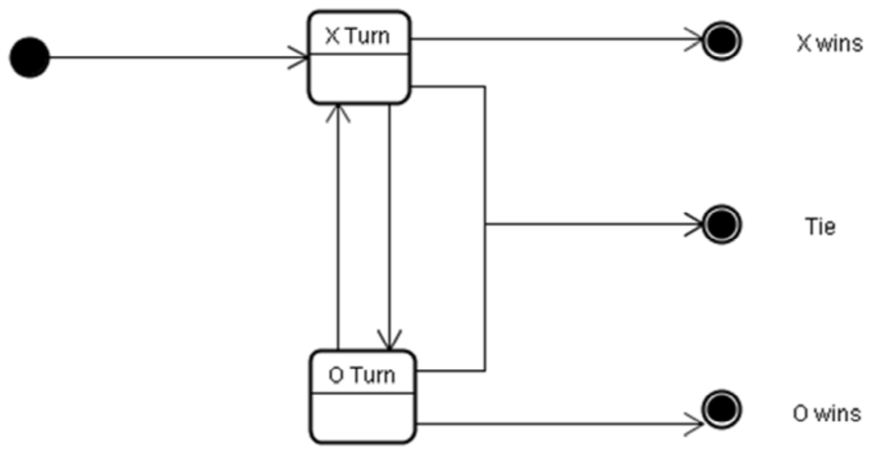
A light switch

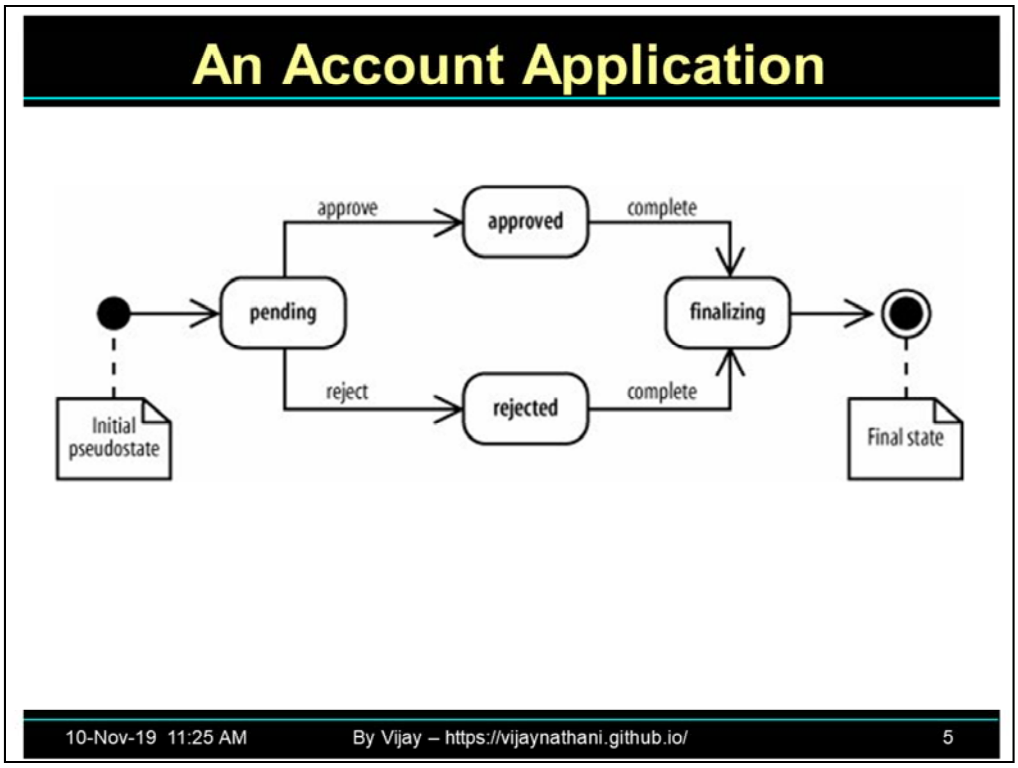


A traffic signal

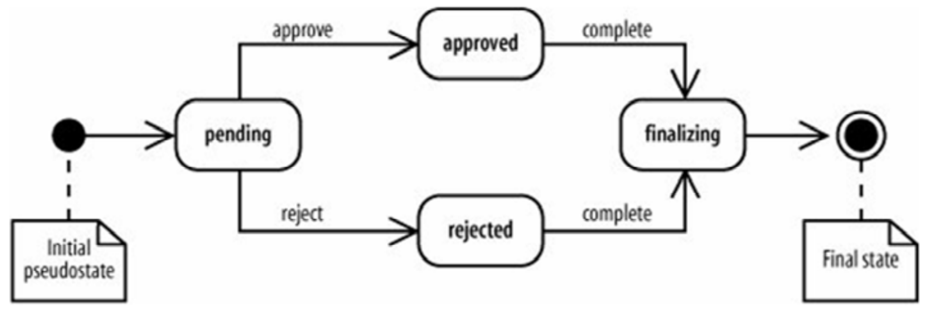


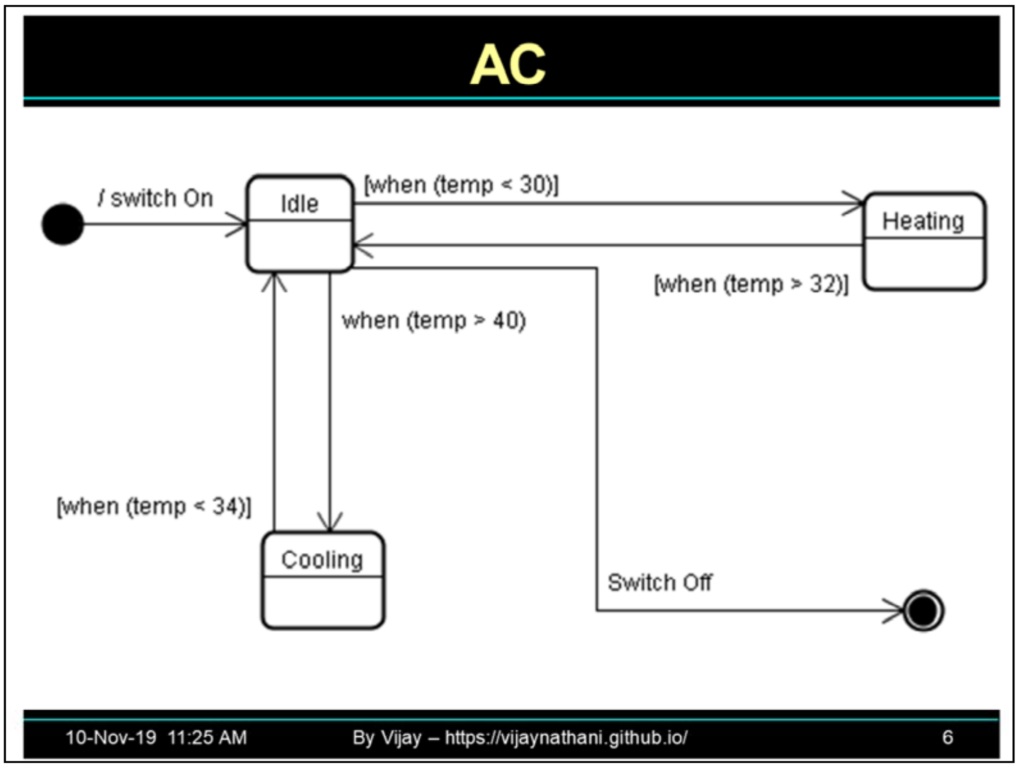
Tic-Tac-Toe Game



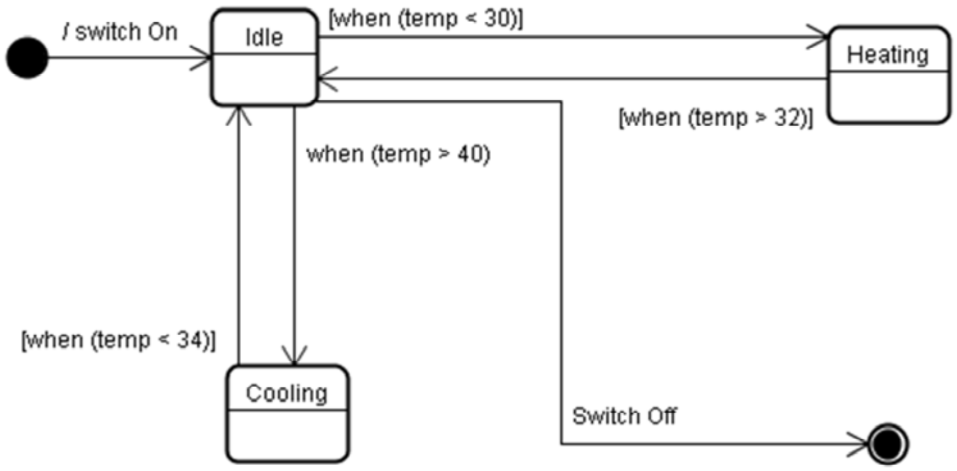


An Account Application

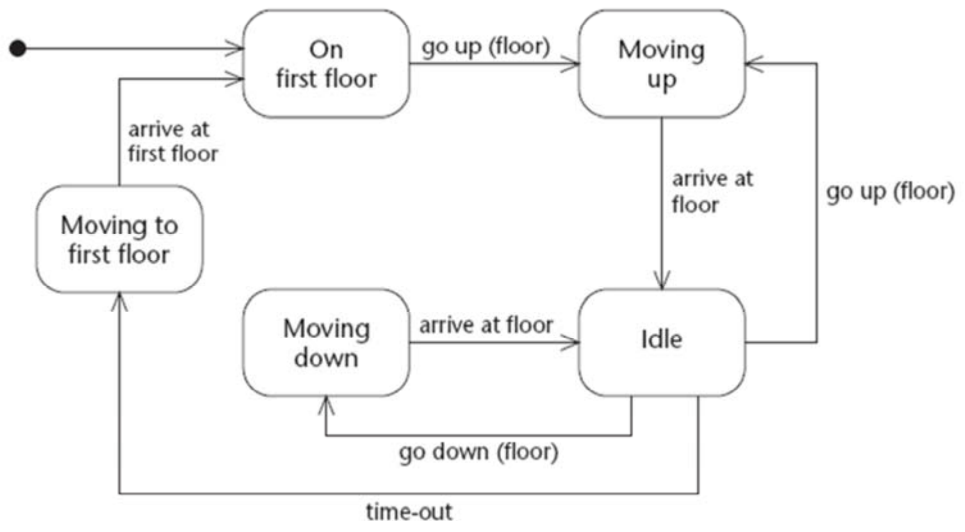


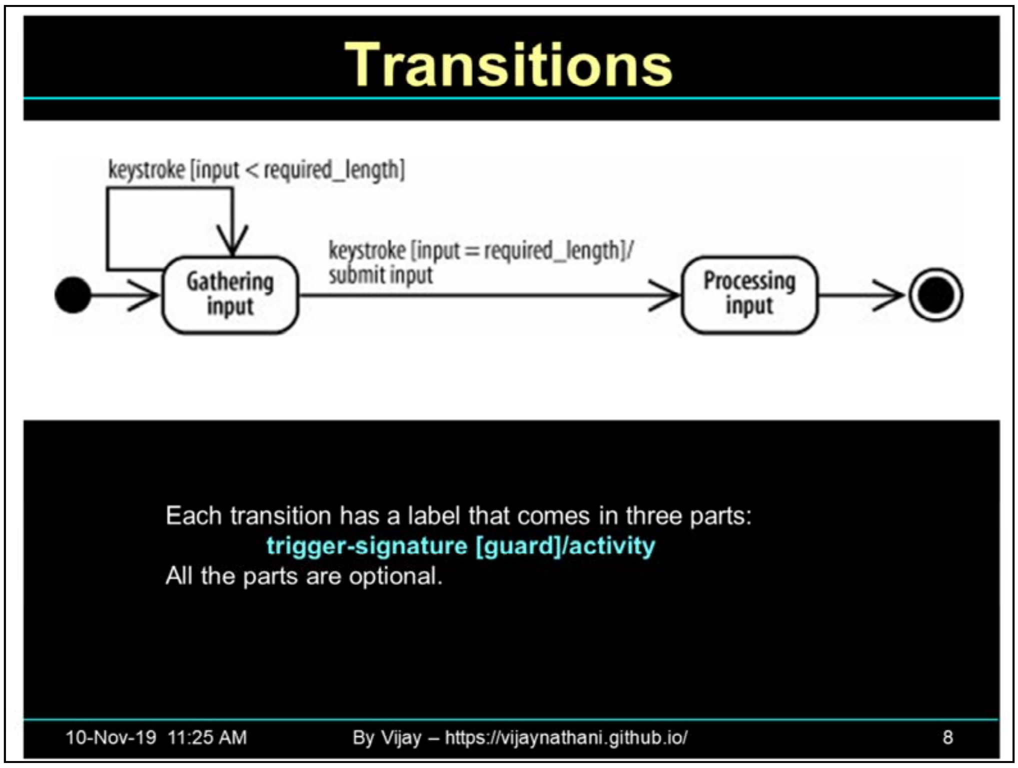


AC

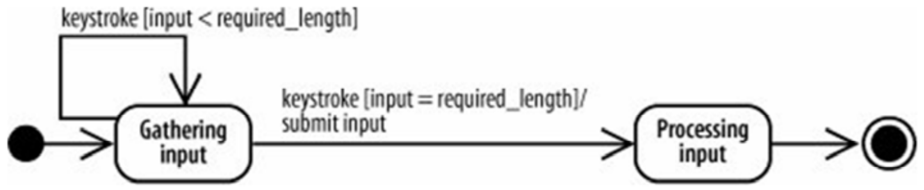

/ switch On
Idle
[when (temp < 30)]
Heating
[when (temp > 32)]
when (temp > 40)
[when (temp < 34)]
Cooling
Switch Off

Elevator


On first floor
go up (floor)
Moving up
arrive at first floor
arrive at floor
go up (floor)
Moving to first floor
Moving down
arrive at floor
Idle
go down (floor)
time-out



Transitions



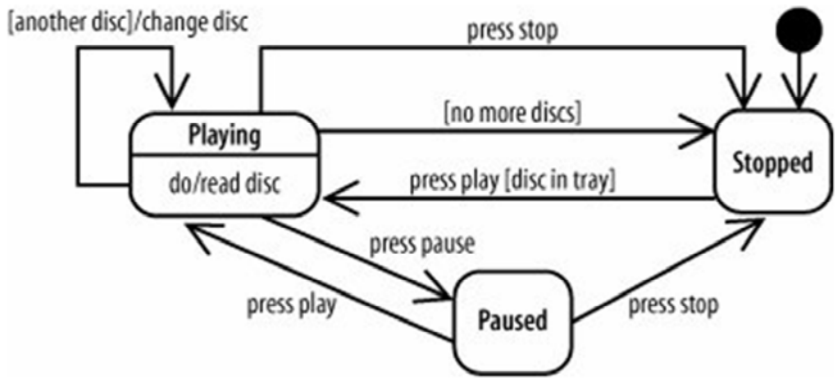
keystroke [input < required_length]
Gathering input
keystroke [input = required_length]/ submit input
Processing input

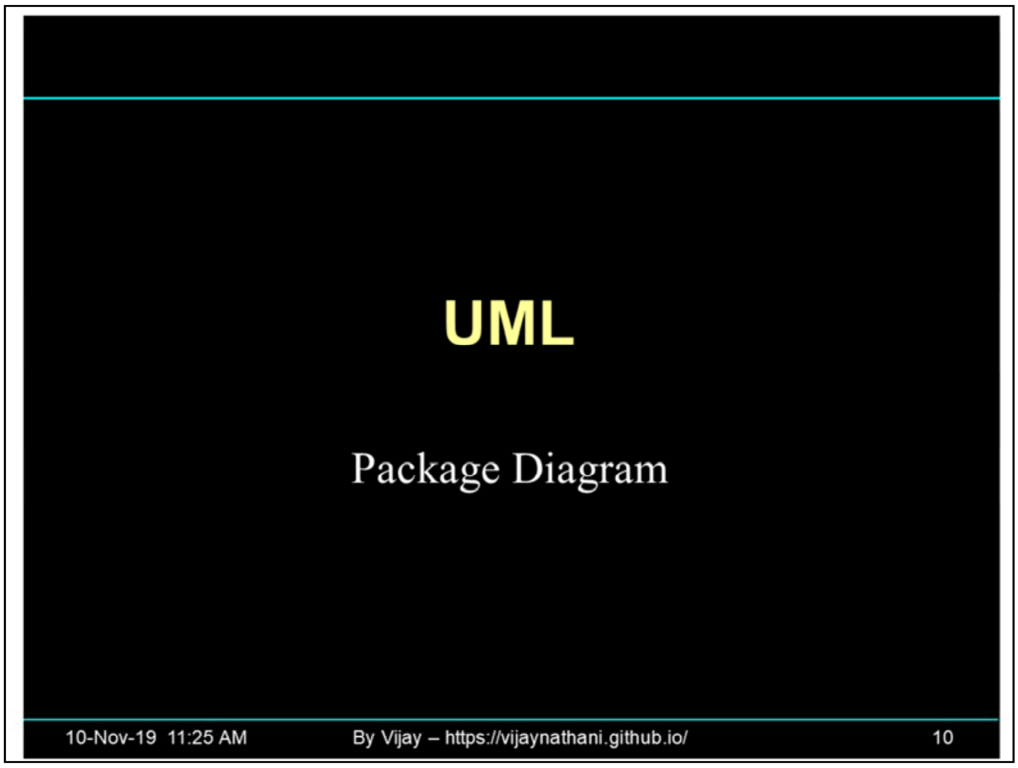
Each transition has a label that comes in three parts:

trigger-signature [guard]/activity

All the parts are optional.

CD Player

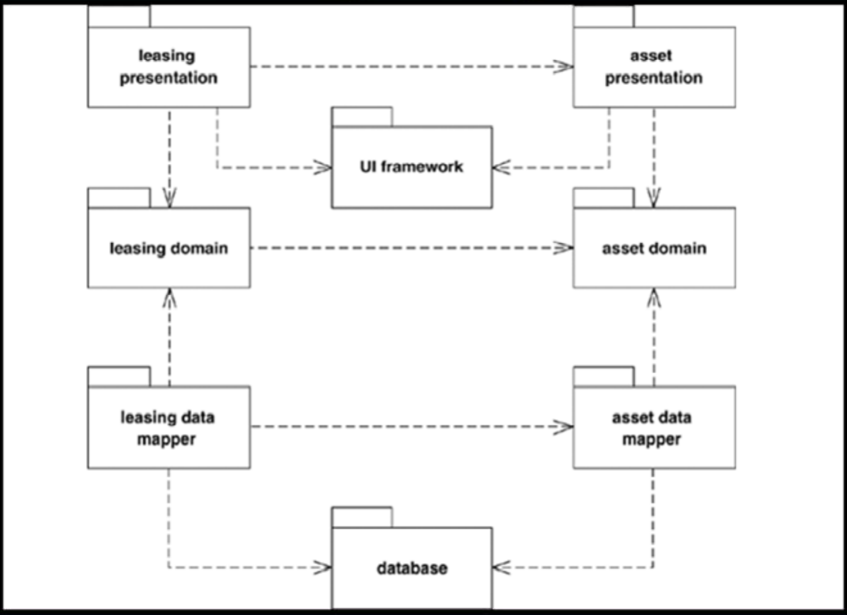

[another disc]/change disc
press stop
[no more discs]
Playing
do/read disc
press play [disc in tray]
Stopped
press pause
press play
Paused
press stop



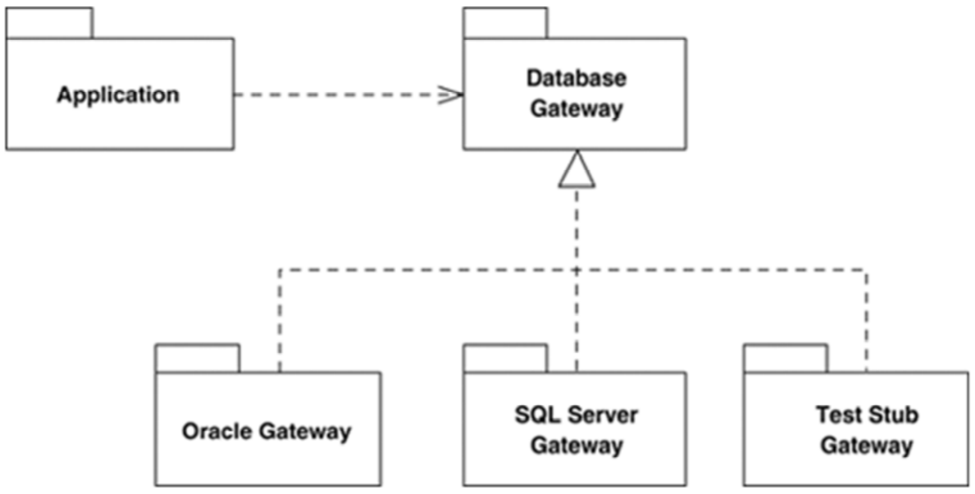
UML

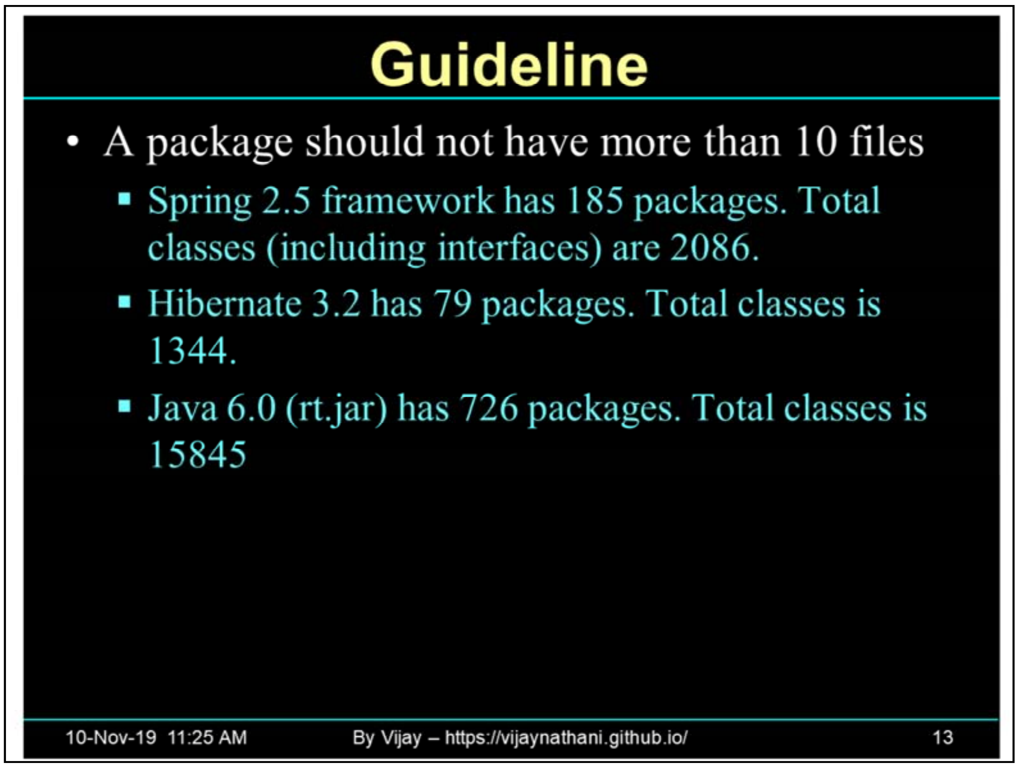
Package Diagram

Package Diagram



Package Diagram

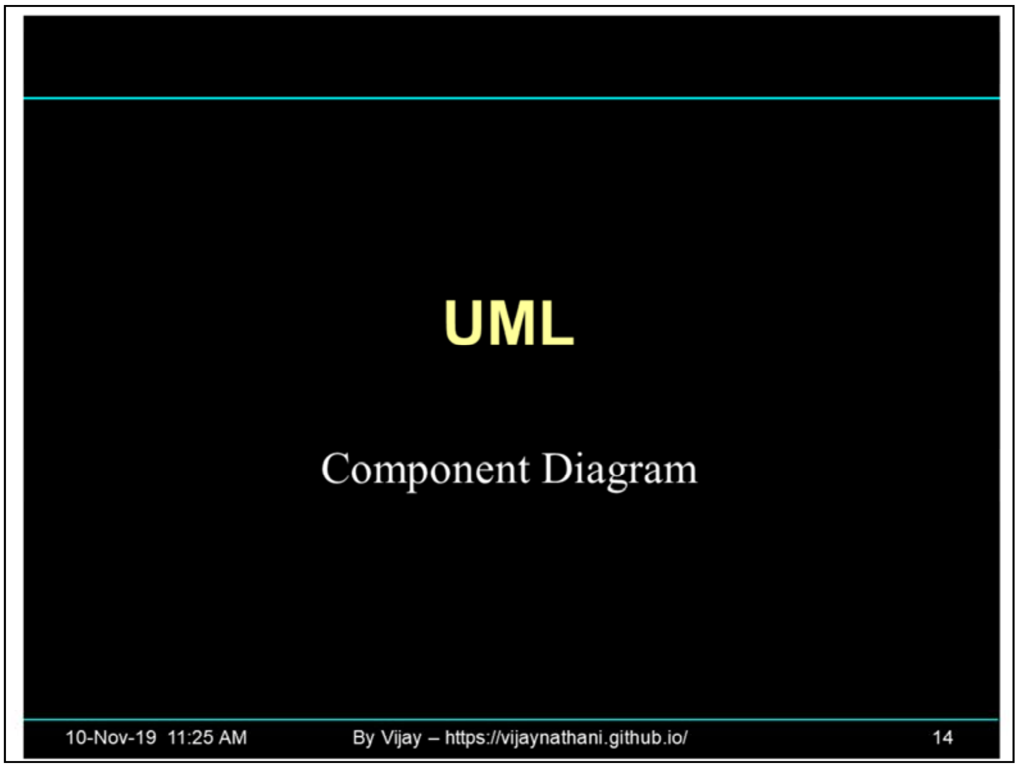




Guideline

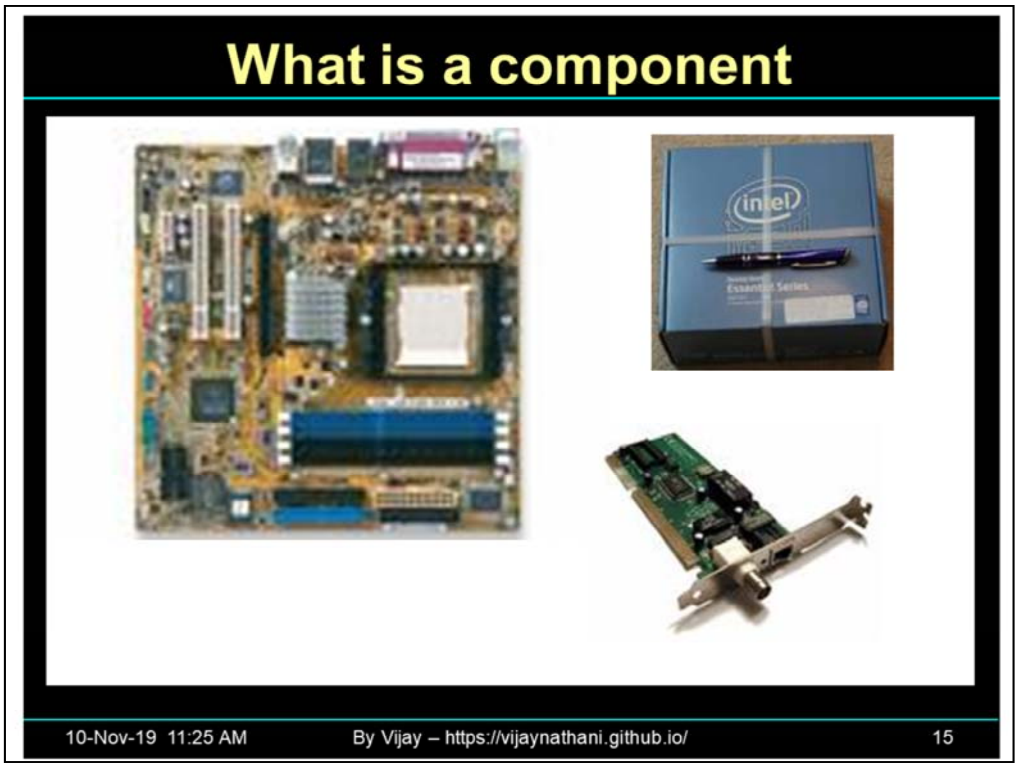
- A package should not have more than 10 files
  - Spring 2.5 framework has 185 packages. Total classes (including interfaces) are 2086.
  - Hibernate 3.2 has 79 packages. Total classes is 1344.
  - Java 6.0 (rt.jar) has 726 packages. Total classes is 15845

10-Nov-19 11:25 AM By Vijay – https://vijaynathani.github.io/ 13



UML

Component Diagram



What is a component

10-Nov-19 11:25 AM By Vijay – https://vijaynathani.github.io/ 15

Components are not a technology.

- Technology people seem to find this hard to understand.

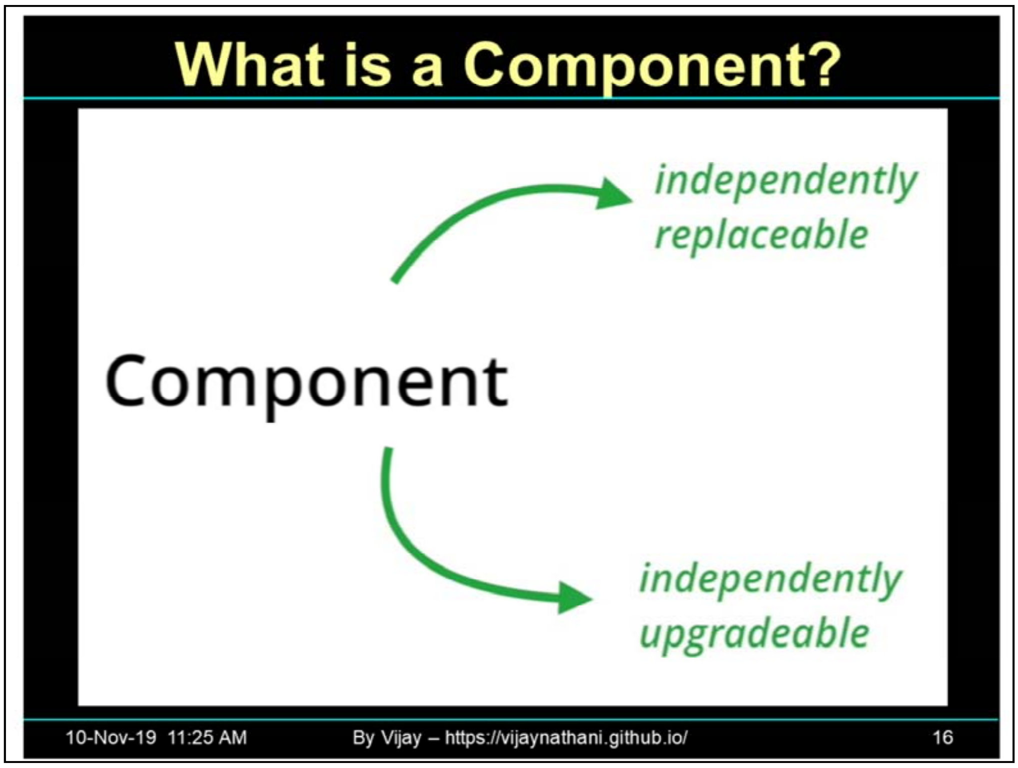
Components are about how customers want to relate to software.

- They want to be able to buy their software a piece at a time, and to be able to upgrade it just like they can upgrade their stereo.
- They want new pieces to work seamlessly with their old pieces, and to be able to upgrade on their own schedule, not the manufacturer's schedule.
- They want to be able to mix and match pieces from various manufacturers.

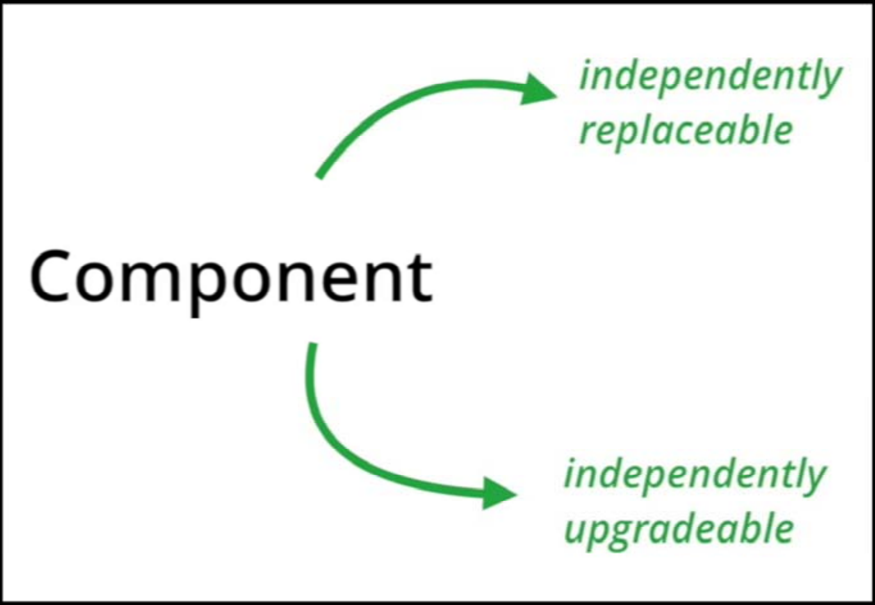
This is a very reasonable requirement. It is just hard to satisfy.

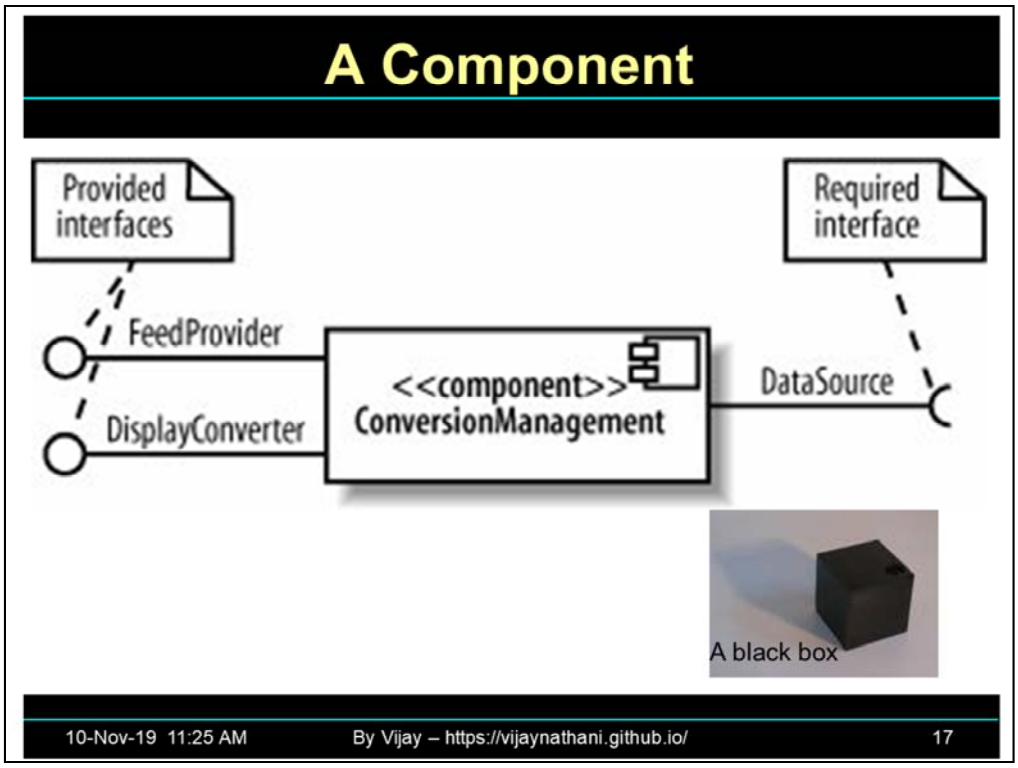
Components / Modules are collection of packages. High cohesion within module. Low coupling with other modules. Modules are a conceptual unit, source management and deployment unit. Every module should have a distinctive role in a large system.

Each layer can have multiple modules. Higher layers depend upon the lower layers and not the other way around.

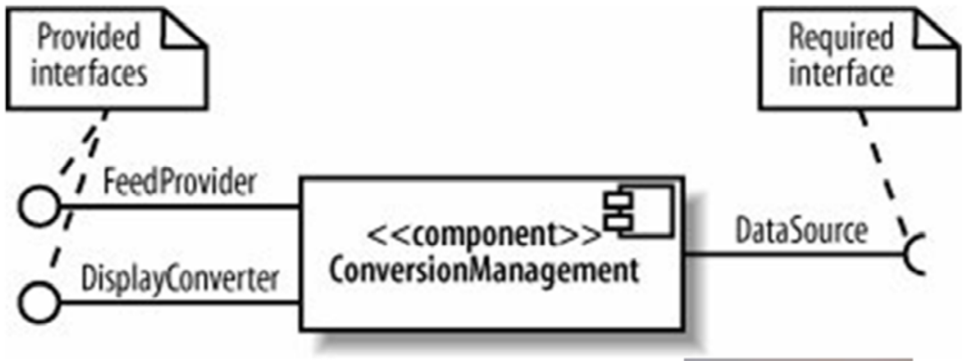


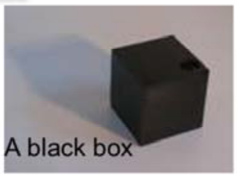
What is a Component?





A Component




A black box

10-Nov-19 11:25 AM By Vijay – https://vijaynathani.github.io/ 17

Components usually have a façade. Expose one or more packages only.

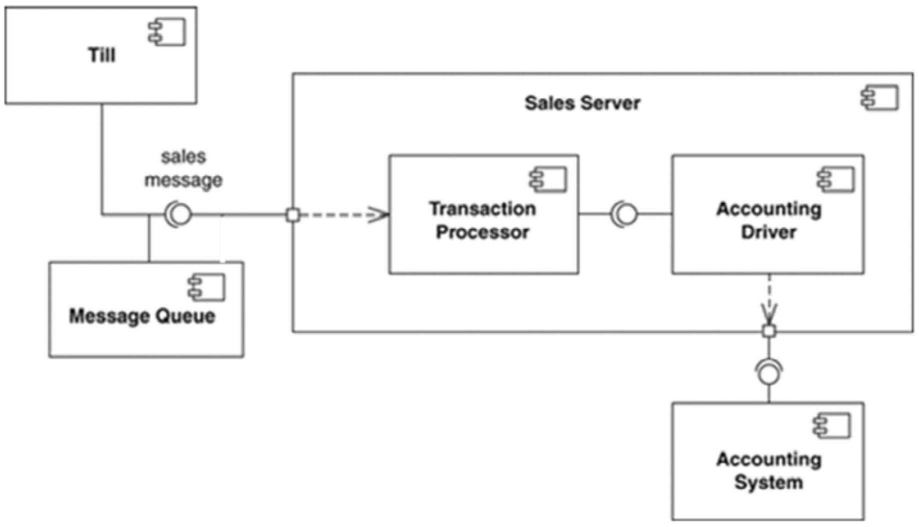
Backward compatibility and architectural quality need not be separate goals.

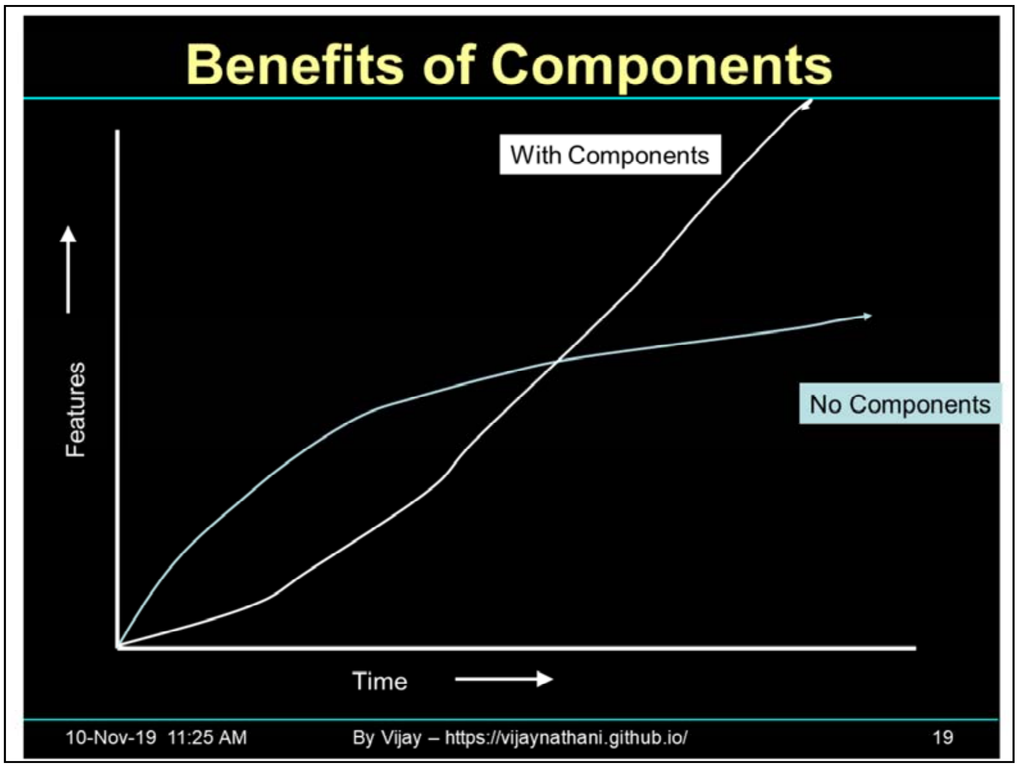
Example: How to retain backward binary compatibility, when the library that we are depending upon is changing. Different users can use different versions of library. e.g. Spring can provide support Hibernate, iBatis, Quartz. Each has evolved over time. Users can use these libraries directly.

Spring achieves this by using reflective checks and invocations.

Rule: Things should be as independent as possible.

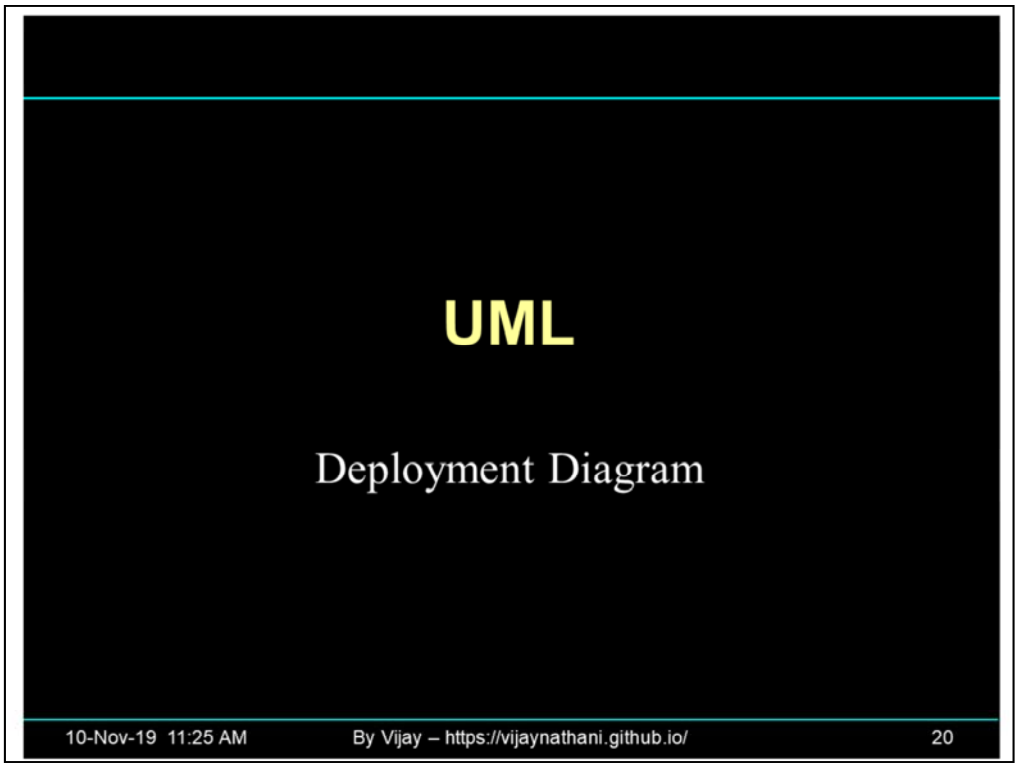
Component Diagram





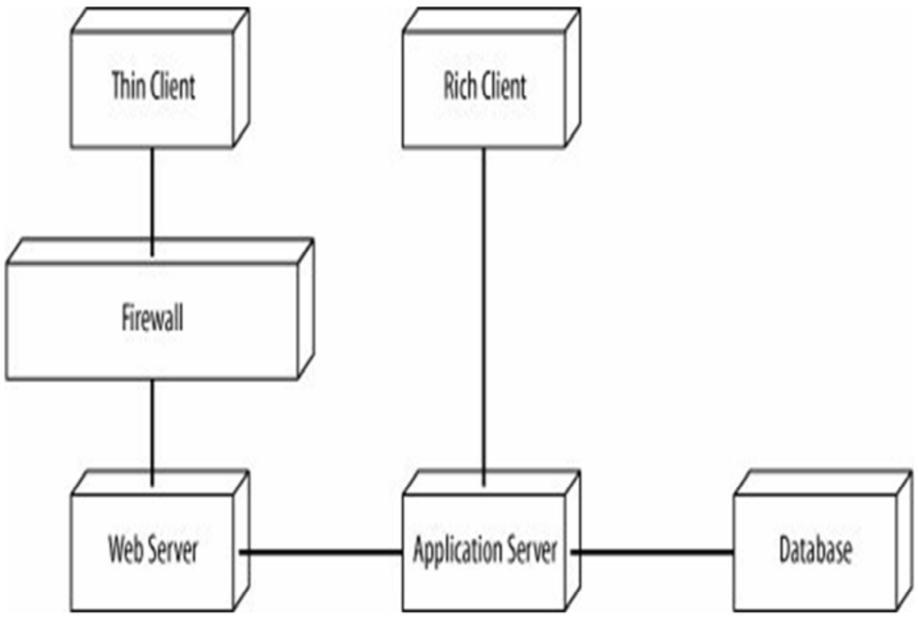
Benefits of Components

10-Nov-19 11:25 AM   By Vijay – https://vijaynathani.github.io/   19

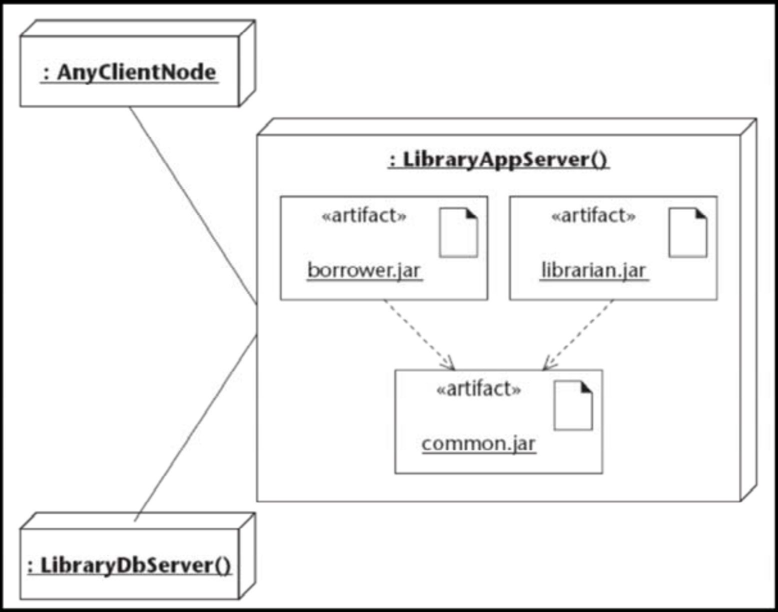


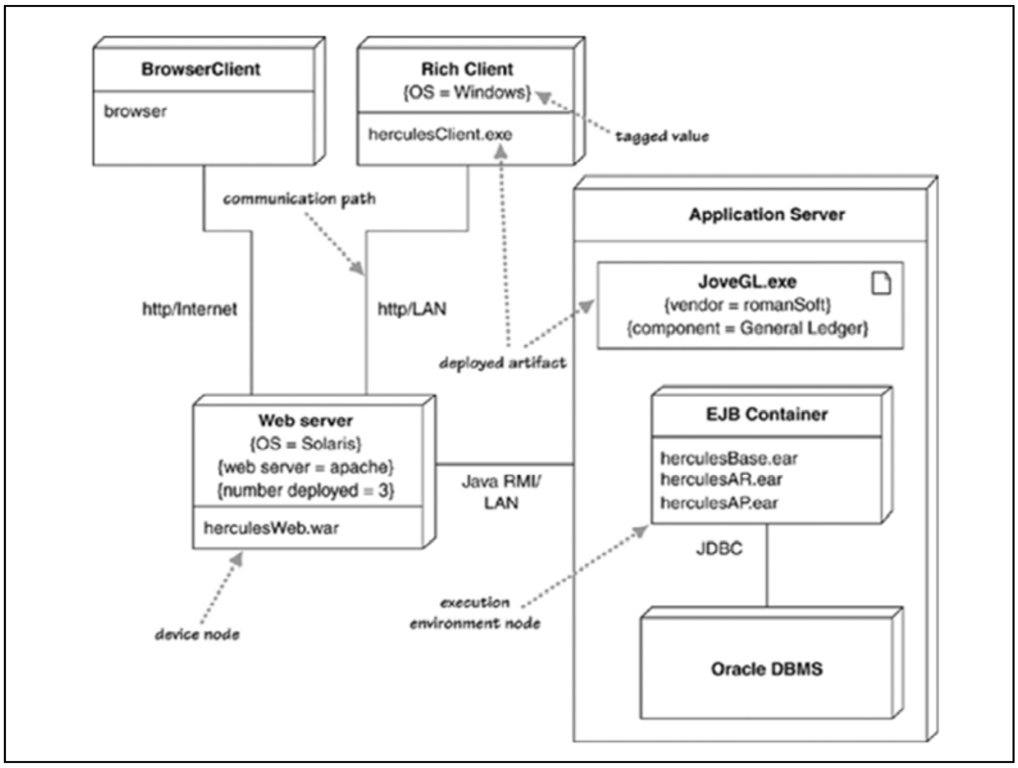
UML

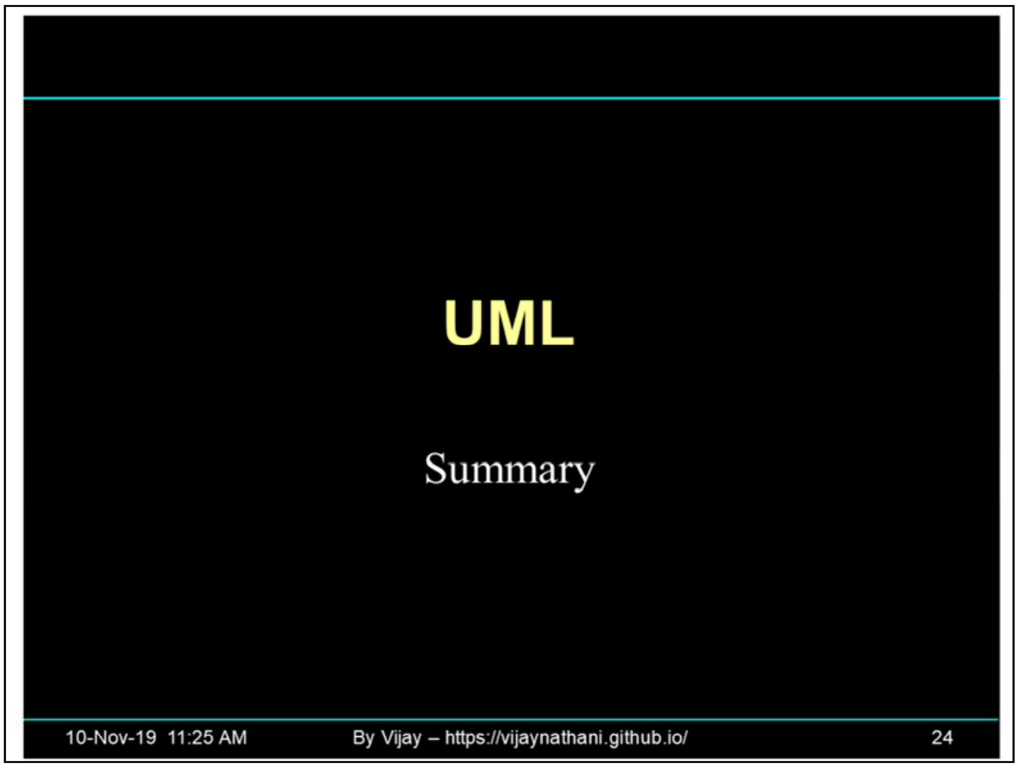
Deployment Diagram


Thin Client
Rich Client
Firewall
Web Server
Application Server
Database

Deployment Diagram



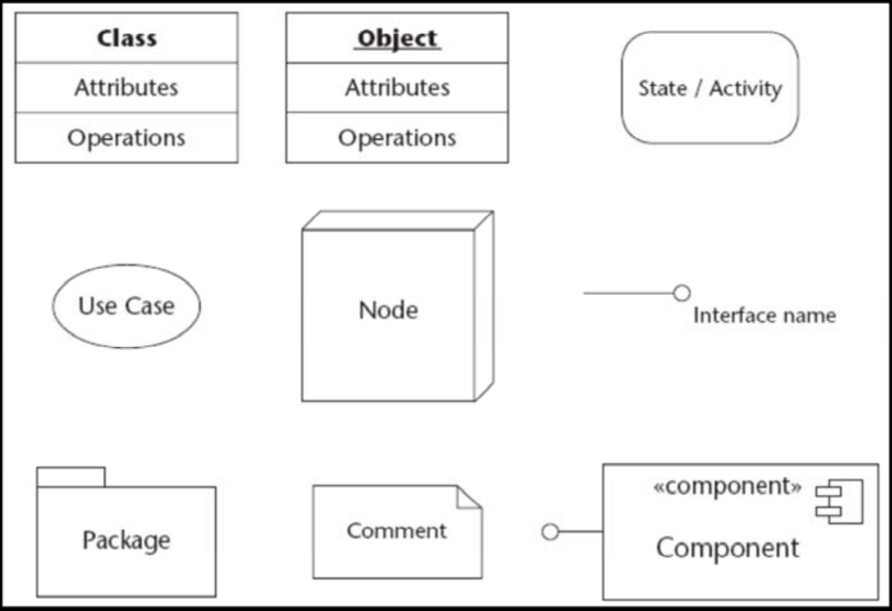

BrowserClient
browser
Rich Client
{OS = Windows}
herculesClient.exe
tagged value
communication path
http/Internet
http/LAN
Application Server
JoveGL.exe
{vendor = romanSoft}
{component = General Ledger}
deployed artifact
Web server
{OS = Solaris}
{web server = apache}
{number deployed = 3}
herculesWeb.war
Java RMI/
LAN
EJB Container
herculesBase.ear
herculesAR.ear
herculesAP.ear
JDBC
execution
environment node
device node
Oracle DBMS



UML

Summary

Summary


Class
Attributes
Operations
Object
Attributes
Operations
State / Activity
Use Case
Node
Interface name
Package
Comment
«component»
Component

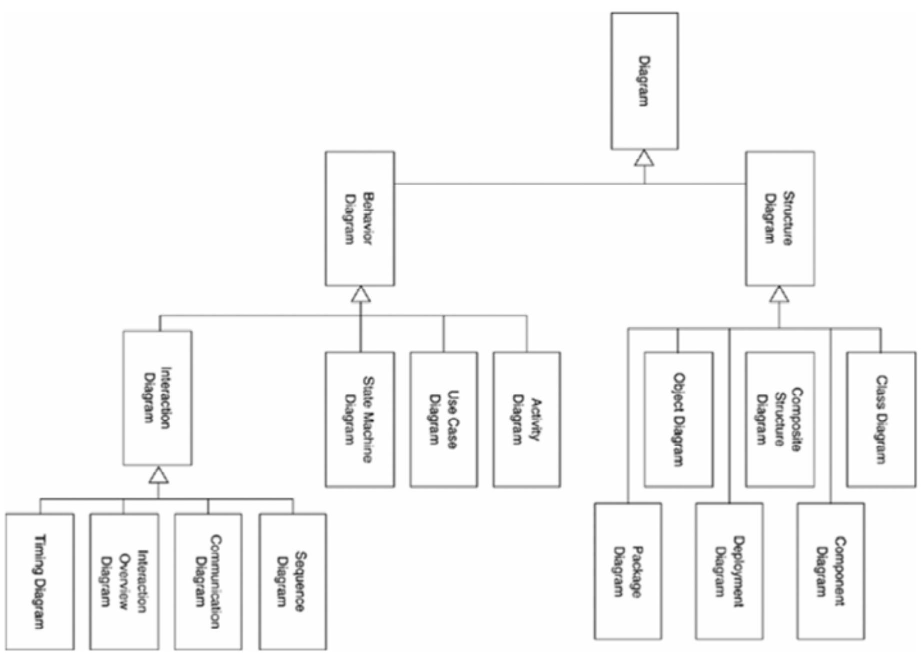
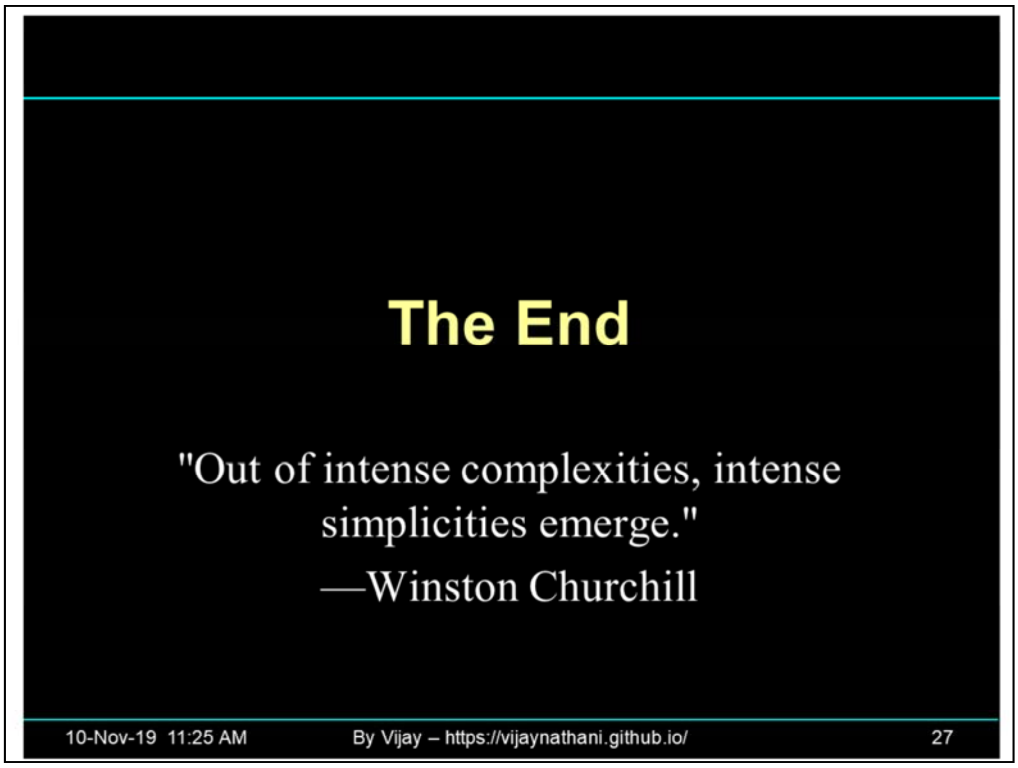


Diagram
Behavior Diagram
Structure Diagram
Interaction Diagram
State Machine Diagram
Use Case Diagram
Activity Diagram
Object Diagram
Composite Structure Diagram
Class Diagram
Timing Diagram
Interaction Overview Diagram
Communication Diagram
Sequence Diagram
Package Diagram
Deployment Diagram
Component Diagram



The End

"Out of intense complexities, intense simplicities emerge."

—Winston Churchill