	CDE STUDY CENTRE ANNA UNIVERSITY, CHENNAI 600 025	044- 22357222

ACKNOWLEDGEMENT

SET-32

SEMESTER : II

Name	HARYISHI E.
Roll No	2232MCA0058
Course	MCA

Semester II

Sl. No.	Sub. Code	Subject Name	Assignments Submitted ( )	Assignments not Submitted (X)
1	DMC6201	Internet Programming	2	
2	DMC6202	Cloud Computing Technologies	2	
3	DMC 6203	Artificial Intellegence & ML	2	
4	DMC 6204	Mobile Application Developement	2	
5	DMC 6205	Cyber Security	2	
6	DMC 6001	Software Project Management	2	
Total No. of Assignments submitted				

Signature of Candidate

Signature of the Staff

SET 32 - MCA

Name : HARYLSH E.

Roll No : 2232MCA0058

Course : Master of Computer Applications

SUBJECT CODE

DMC 6201

SUBJECT NAME:

~~INTERNET~~
INTERNET PROGRAMMING

Marks	
Name and Signature (faculty)	

19-10-2023 | IInd SEMESTER ASSIGNMENT - 1 | 2232MCA0058

2MCA | INTERNET PROGRAMMING | DMC6201

1. A. Formulate a (python) program that works on with JSON.

Creating a program that works with JSON involves tasks like fetching JSON data from remote server, parsing JSON, and possibly sending JSON data to a server. Here we demonstrated a simple example program in python that operates JSON data.

Program 1: Python to fetch the data from web source:

```
import requests
import json

#URL to fetch data from that API.
source-url = "https://example.com/api/data"

try:
    #Fetch JSON data from the API
    response = request.get (source_url)
    if response.status_code == 200:    #checks if we get proper response
        data = response.json()    # parses the JSON response.
        for item in data:
            print(f"ID: {item['id']}, Name : {item['name']}")

    else:
        print(f"Failed to fetch the data. Statuscode:{
                response.status_code}")
```

```
except requests.exceptions.RequestException as e:
    print(f"An error occured : {e}")
```

Practical Application of working with JSON :

Consider a situation of displaying the entire detail of a particular student and taking the phone number of the student from JSON data. The manipulation of data in JSON can be taken with 3 approaches of fetching JSON data. From those we can fetch the relevant details. Here is the program in Javascript of the same :

Program:

```
# Approach 1: Fetching data from API link
async function fetchStudentDataFromAPI(apilink, rollno) {
  try {
      const response = await fetch(apilink);
      const data = await response.json();
      const student = data.find(student => student.rollno == rollno);
      return student;
  }
  catch (error) {
      console.error("Error fetching data from API:", error);
      return Null;
  }
}

# Approach 2: Fetching data from a seperate JSON file
async function loadStudentDataFromFile(filePath, rollno) {
  try {
      const response = await fetch(filepath);
```

```
    const data = await response.json();
    const student = data.find(
                student => student.rollno === rollno);

    return student;
  } catch (error) {
    console.error ("Error loading data from file : ", error);
    return null;
  }
}
```

Approach 3 : Use Internal JSON data

```
const internalStudentData = [
  {
    rollno : "101",
    name : "John Koh",
    contact-number : "123-456-7890"
  }
];

// Function to read student data
function readStudentData (rollno){
  constant student = internalStudentData .find (student =>
    student.rollno === rollno);
  return student;
}
function toRead Contact (rollno){
  constant student = load StudentData fromFile (rollno);
  return student ? student. contactNumber : "Student Not found";
}
```

```
const rollnum = "101";
const student1 = readStudentDataByRollNumber (rollnum);
const studcontact = read ContactNumber (rollnum);
console.log ("Student Data of {%s} is : {%s} ", rollnum, student1);
console.log (" Contact : {%s} ", studcontact);
```

1. B. Summarize AJAX client server architecture in detail.

Ajax refers to a set of technologies and techniques that allow webpages be interactive like desktop application. AJAX stands for Asynchronous Javascript and XML. It uses XHTML for content, CSS for presentation and JS with DOM for dynamic content display, in an intention to develop better, faster and more interactive web application.

AJAX is based on the below standards:

- Browser based presentation using HTML and CSS
- Data is stored in XML format and fetches from server.
- Behind the scene data fetching is done using XML HttpRequest objects in the browser.
- JavaScript to make everything happen.

The asynchronous nature in AJAX dipits that the script will send the request and continuously executes the request-responce recieving without waiting for reply. Soon as the reply is recieved a browser event is fired.

Client Server Architecture:

The Ajax provides a rich and diverse set of products that range from hundreds of suppliers. The diversity provides IT managers and web developers with the ability to choose the optimal architectural approach and best products among multiple vendors.

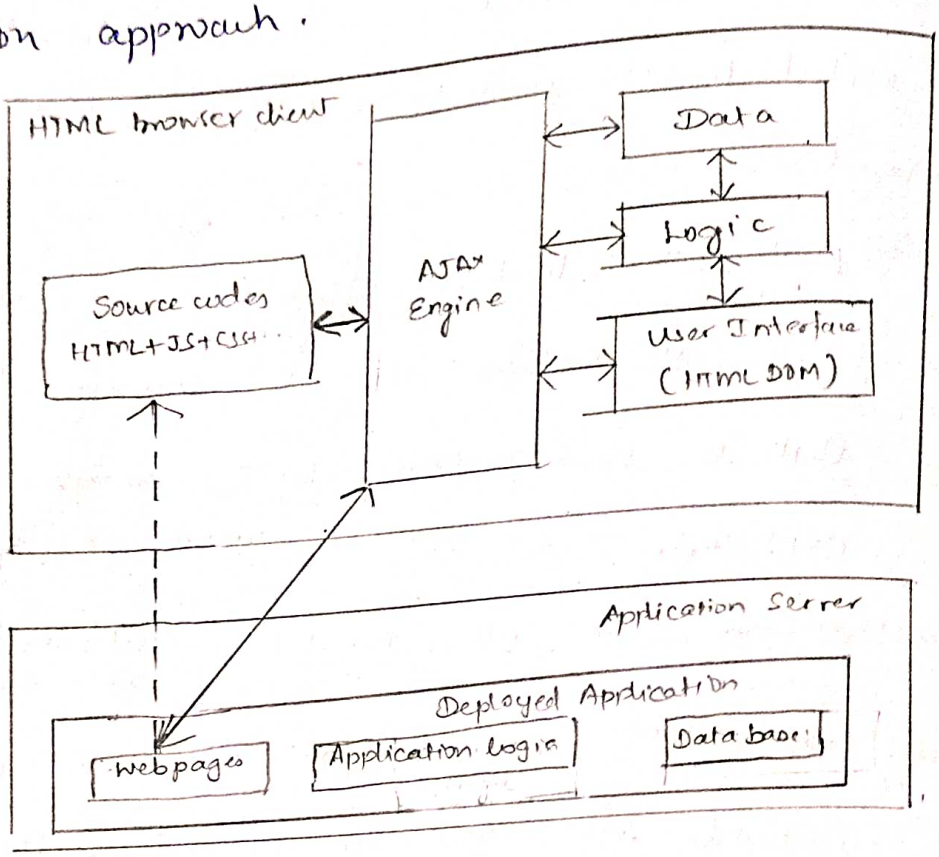
Most AJAX technologies transforms a platform independent definition of the application into the appropriate HTML and JS content that is then processed by the browser to deliver the rich user experience. Some AJAX design platforms most of their transformation on client. Other performs transformation on server.

1. Client-side AJAX transformation:

* With client side AJAX, the engine runs from the client.
* The server delivers web ~~da~~ content (HTML, CSS, JS, etc.) which is processed by the client side AJAX egine into recieved web content
* The browser renders the recieved HTML etc., content that comes out of AJAX engine.
* The application developement team can typically provide the server side components such as web pages, application logic programms and data management.
* Client-side components includes client-side user interface logic such as event handlers.

* Advantage : The independance from server side technology.
* Server code creates and serves the page and response to the client's asynchronous requests.

This way either side can be swapped with another implementation approach.



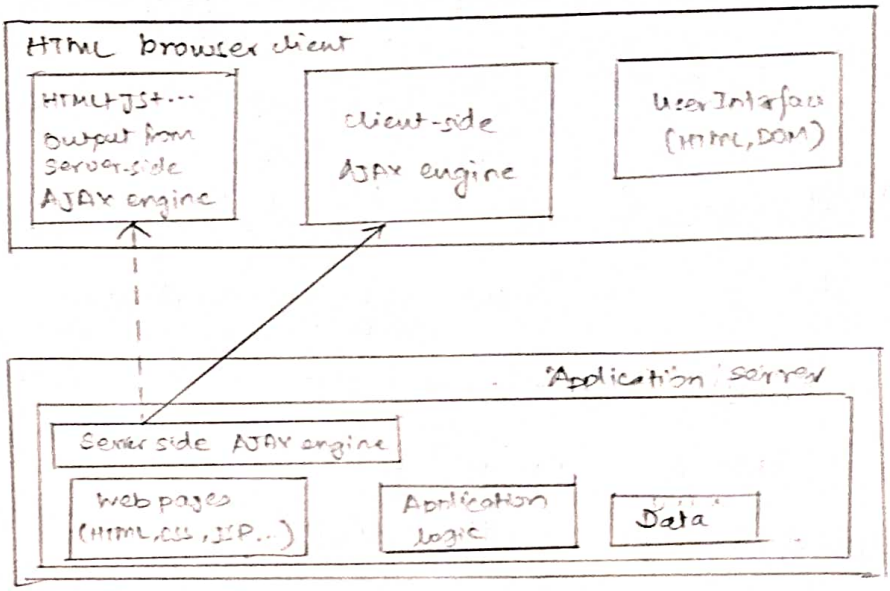
2. Server side AJAX transformation:

* For servers side AJAX, and AJAX server component performs most or all of the AJAX transformations..
* The server component generates the necessary web content (HTML, CSS, JS, etc.) to deliver the desired user experience.
* The server side AJAX toolkit downloads its own client side AJAX library which communicates directly with the toolkit server side AJAX component.
* With this architecture below, the application development

team typically only provides server side components (Web pages, data, and logic) and entrusts client-side logic to Ajax toolkit.

* The main benefit of this approach is that it allows the use of server-side language for debugging, editing and refactoring tools with which developers are familiar with..

* ~~The~~



* The disadvantage of this approach is the dependance on a particular server side technology.

* As a general rule, server side Ajax framework expect app code to be written in the server-side language.

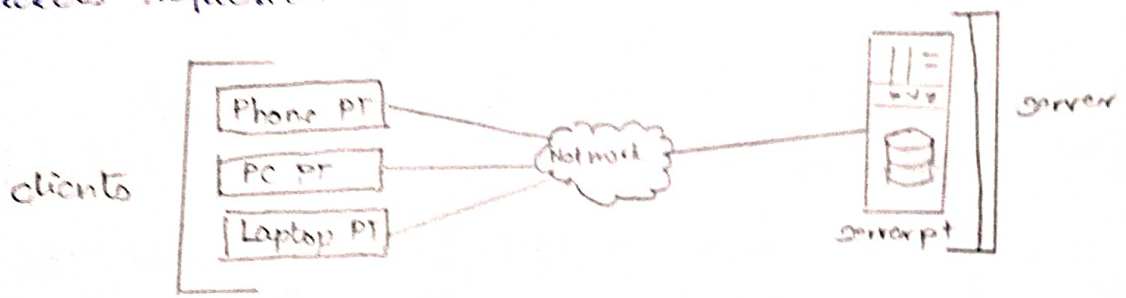
* These frameworks typically hide all the Javascript that runs in the browser inside widgets, including their events.

2. Demonstrate server architecture and explain its working principles.

A server architecture, often refer to server model or server design refer to the overall structure and organization of server within a computing environment. Server architecture can vary widely depending on the specific requirement of an organization or application.

General Mechanism of Server models in client POV :

Basically, in client - server architecture, when client computer sends the request for data to the server through internet, the server accepts the required process and deliver the data packets requested back to the client.



• Interaction of client to Server System:

There are below mentioned steps to follow and interact with the server by the client.,

① Client (user) enters the Uniform Resource Locator of the web site or file. The browser then requests the DNS (Domain Name System) server.

(2) DNS server lookup for the address of the web server.

(3) DNS server responds with the IP address of the web server.

(4) Browser sends over an HTTP/HTTPS request to web server IP (provided by DNS server).

(5) Server sends over the necessary files of the website.

(6) Browser then renders the files and the website is displayed. This rendering is done via DOM interpreter, CSS and JS engine collectively known as JIT compilers.

Varieties of Server Architecture:

1) Client-Server Architecture: Here, multiple clients communicate with the centralized server to request and recieve services or data. The server handles client requests, processes them and sends back responses. Web servers, database servers, email SMTP servers are common examples of single client-server architecture.

2) Peer-to-Peer (P2P) architecture: Here, all nodes (devices) in the network have equal status and can act as both client and servers. They directly communicate and share resources with one another. File sharing application like BitTorrent applies the peer-to-peer architectures so as to maintain connectivity to multiple users. It got benefits such as decentralized, no single point of failure, suitable for small networks.

3) Three-Tier architecture : This devides the application into three layers such as presentation, application logic and data storage management. Each layer serves by a seperate servers or server cluster. This promotes scalability, seperation of concepts and easy maintainance. Full stack web application developers would likely prefer this variant as a system architecture.

4) N-tier architecture : This extends the three tier model to include multiple tiers of layers for improved scalability and mobility. This as well promotes or extends the power of scalability, enhances security and improves maintainability. Large scale enterprise application is the example.

Working Principles of Server architecture :

The working principles involves the following key concepts :

(i) client :

- → Client, often a web browser, is the user interface. User interacts with the client to access web applications.
- → User initiates actions by clicking links, submitting forms or typing URLs in the browser's address bar.
- → The client sends HTTP request to the server when users want to retrive data or perform actions.

(ii) Server Application:

-> The server application is responsible for processing client requests and generating responses.

-> It runs on a web server (eg. Apache, IIS) and hosts your web applications code.

-> When client request is recieved, the server application processes it and may interact with databases and other services.

-> It generates dynamic context, such as HTML, JSON and XML, which is sent back to the client as an HTTP response.

(iii) Request - Response cycle:

-> The client sends an HTTP request to the specific URL on the server.

-> The server /responds/ listens for incoming request, routes them to the appropriate handles and processes the request.

-> It retrives data from the other data sources, performs business logic, and generates the response.

-> The response, typically in the form of HTML for webpages or ~~ABI~~ data for API, is sent back to client as an HTTP responce.

-> This entire process is combinely called as request - response cycle.

(iv) Stable communication (stateless)

-> The communication between the client and server is stateless, meaning each req. from client is independent.

-> To maintain user session state, the server may use cookies, token or session management techniques.

(v) Scalability and load balancing :

In the real world applications, multiple server instances may be used to handle an high volume of request. Load balancers distribute incoming requests across these servers instances to ensure even load balance.

(vi) Security :

Security measure ensures such as encryption (HTTPS) authentication, and authorization are crucial to protect data and ensures that only authorized users can access the resources.

19-10-2023 | IInd SEMESTER ASSEGINMENT - 2. | 2233MCA0058

2 MCA | INTERNET PROGIRAMMING | DMC6201

1. Compare and contrast RESTFUL web services and spring security with it's architecture.

RESTFul web services and spring security are two distinct components of web services application developement, and they serve different purposes.

Similarities:

A. RESTFUL Webservices:

(i) Purpose and functionality:

* RESTFul web services are designed to allow communication processes between different systems over the internet. They provide a way for application to interact with each other by sending and recieving data, often in JSON and XML format.

* They follow the principles of REST, which promotes stateless, resource based interactions, and an uniform interface.

(ii) Architecture:

* RESTful services typically follow a client-server architecture where clients make HTTP request to servers. Servers process those requests and return HTTP responces, often in JSON or XML format.

* RESTful services do not maintain session state on the server, making them stateless. Each request from the client must contain all the necessary information.

* RESTful ~~fllow~~ follows single tier to N-tier architecture style.

3. (iii) Key features:

* Stateless : Each request from a client to the server must contain all the information needed to understand and fulfill the request.

* Resource based: Resources are represented by URLs, and interactions are driven by these resources.

* HTTP Methods : Restful services use standard HTTP method for CRUD such as GET, POST, PUT, DELETE.

(iv) Security :

RESTful services often implement security measures, such as authentication and authorization, to protect the APIs from unauthorized access. These security mechanisms can be implemented using various technologies, including Spring Security.

B. Spring Security :

(i) Purpose and Functionality :

* Spring security is a framework for authorization and authentication in Java-based web application. It provides mechanisms to secure web applications by controlling access to various parts of the application based on user roles and responsibilities.

(ii) Architecture :

* Spring Security can be integerated into various web application architectures, including RESTful services, to control who can access specific endpoint or resources.

* It typically works as a filter in the application's request processing chain, intepreting and intercepting requests and

checking whether users can authenticate and authorize to access the resource requested.

(iii) Key features:

* Authentication: Spring security supports various authentication mechanisms, including username / password, tokens and more.

* Authorization: It enables fine grained control over who can access specific parts of an application based on user roles and permissions.

* Integeration: Spring Security can be integerated with Spring based applications, including spring boot, and offers seamless security features.

(iv) Security:

* Spring Security primarily focuses on authorization and authentication within a web application. It provides a robust set of security features to protect the application from unauthorized access.

* It can also be used to ensure REST webservices by ~~cont~~ controlling access ~~by~~ to API endpoints, making it a suitable choice when RESTful services are part of a broader web application.

<u>Differences</u>:

~~Feature~~: The below table depits the differences between RESTful web services and Spring Security which are typically two different aspects of building web application.

Let us contrast both aspects below:

Points.	RESTFul Webservices	Spring Security
1. Natures	They are designed architecture style for creating networked applications, emphasizing stateless and resource-based interactions.	It is a framework that provides security features for Java applications, often used to secure web applications.
2. Purposes:	They are primarily used for exposing and consuming data via HTTP methods, typically for APIs.	It is used to secure applications including web application, by providing features like authentication, authorization and protection against common security threats.
3. Functionality:	RESTful focuses on data transfer and communication, without inherent security features.	Focuses on user authentication authorization and ensuring that resources are accepted securely.
4. Components:	Typically involves endpoints, data representations and HTTP methods.	Includes components like authentication provides, security filters, users details services and access control.
5. Integration:	Can be built using various frameworks and technologies, not limited to the spring	An Integeral part of the Spring ecosystem, often used in conjuction with Spring MVC for web application.
6. Security Aspect.	Security is not build in this feature but can be implemented seperately using techniques likes API key, OAuth, or custom authentication.	Specialized in provides security for application, with features for authentication, role-based access control, and protection against common web security.

2. Analyse about Highest Order Components with sample program.

Highest Order Components (HOC's) are the design pattern used in React to ensure the reusability and modularity of components. An HOC is an function that takes a component and returns a new component with added functionality. This pattern is particularly useful for implementing cross-cutting concerns such as data fetching, authentication and state management.

Suppose we have a basic React component that displays a list of items.,

```
import React from 'react';
class ItemList extends React.Components{
    state = { items = [], };
    component DidMount (){
      // simulating data fetching from an API
      setTimerOut( () => {
          this.setState({ items: ['Item1', 'Item2', 'Item3'], });
      },1000);
    }
    render () {
      return (
        <div>
          <h1> Item list </h1>
          <ul >
            { this. state. items. map ({item,index} => (
                <li  key = {index } > {item}</li>

            ) }
          </ul>
        </div>
      );
    };
}
```

```
export default ItemList;
```

Now, let's create an higher-order component that adds loading functionality to any component that needs it;

```
import React from 'react';
const withLoading = (WrappedComponent) => {
  return class WithLoading extends React.Components {
    state = { isLoading : true; }
    componentDidMount () { // simulating asynchronous operation.
      setTimeOut(() => { this.setState({isLoading: false}); }, 1000);
      render (){
        return this.state.isLoading ? (
          <div> Loading... </div> ) : (
          <WrappedComponent {...this.props} />);
      }
    };
  };
export default withLoading;
```

In this example,

1. We create the highest order component 'withLoading', which takes a 'Wrapped Component' as an argument.
2. Inside 'withLoading', we render a loading indicator while our asyncronous operation is in progress (simulated with 'TimeOut') and then render the 'Wrapped Component' once the loading is complete.

Now, we can use 'withLoading' HOC to wrap our 'ItemList' component by below program;

```
import React from 'react';
import withLoading from './withLoading'; //importing HOC
class ItemList extends React.Component {
    // .. same as before ...
}
//Wrapping the component using the HOC
const ItemListWithLoading = withLoading(ItemList);
export default ItemListWithLoading;
```

With this setup, 'ItemListWithLoading' becomes a new component that includes the loading behaviour without modifying the original 'Item List' component.

This seperation of concerns makes it easier to manage and reuse the functionality across the different parts of your application.

SET 32 - MCA

Name : HARVISHI E.

Roll No : 22 32 MCA 0058

Course : Master of Computer Applications

SUBJECT CODE

DMC 6202

SUBJECT NAME:

CLOUD COMPUTING TECHNOLOGIES

Marks	
Name and Signature (faculty)	

28-10-2023 | II nd SEMESTER ASSIGNMENT - 1 | 2232MCA0058

2 MCA | CLOUD COMPUTING TECHNOLOGIES | DMC 6202

1. Discuss about Stub generation in distributed computing systems.

A stub is the piece of code that transforms parameters passed between the client and server during a remote procedure call (RPC). An RPCs main purpose is to allow a local computer to call procedure on another computer remotely (server) because the client and server utilize distinct address spaces. That is the reason parameters used in a function (procedure) call must be transfered; otherwise, the values in one computer memory will point to different data on the other computer.

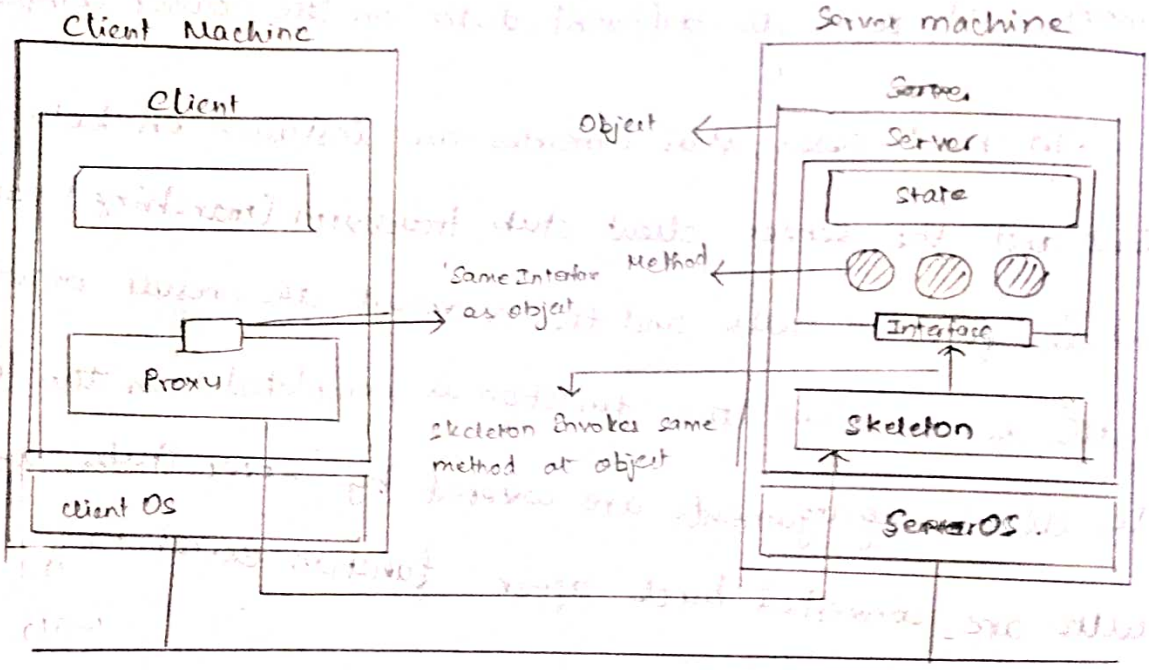
In most cases, stub libraries are installed on both the client and the server. client stub transform (marshing) parameters used in function calls and then reconvert the result returned by the server when the function is completed. On the other side, client arguments are covered by server stubs, and results are converted back after function execution.

Stub Generation:

Stub generation can be created in two different ways:

* **Manual Generation of stub**: In this, the RBC implementer provides a connection of translations functions from which a user can create their own stub using this way. This method is easy to use and can handle a wide range of argument types.

* **Automatic Generation of Stubs**: Here, the client and server interface are defined using Interface Definition Language (IDL). An Interface specification, for example, contains information indicating whether each argument is input, output, or both. Only input arguments must be passed from client to server, while only output elements must be copied from server to client. This is the most popular way of generating stubs.



Architecture:

The above diagram displays the architecture of client server that transfers stubs, which each component is explained below.,

Import Interface : A client program that calls procedures from an interface is said to import the interface.

Export Interface : A server program that implements a procedure in an interface is said to export an interface.

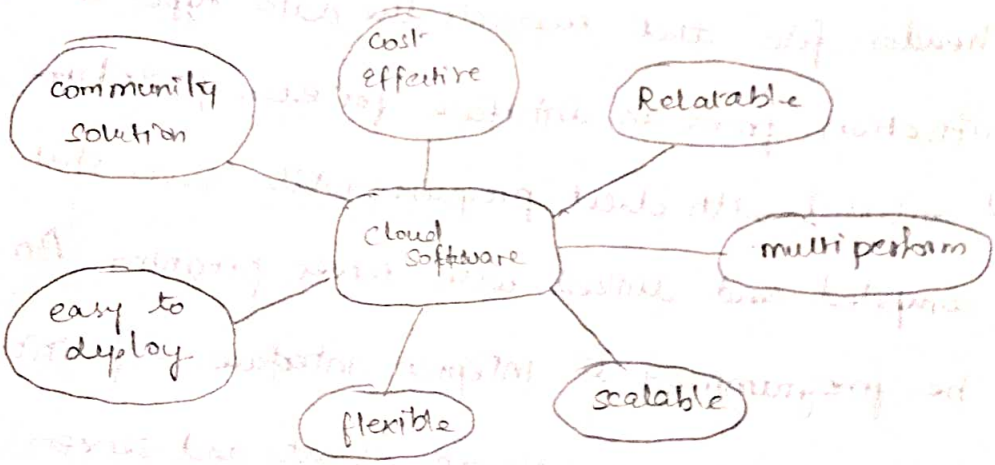
Interface Definition Language (IDL) : when building a distributed application, a programmer first creates an interface definition in IDL, then writes the client program that imports the interface and the server program that exports it. An IDL compiler is used to process the interface definition and generate components that may be coupled with client and server programs without required any change to existing compilers. The compiler generates the required marshalling and unmarshalling operations in each stub procedures and a header file that supports the data types in the interface specification from an interface for each procedure are compiled and unlinked with client program, while server stub procedures are compiled and linked with server program. An IDE compiler can be programmed to interpret interface definitions for a variety of languages, allowing clients and servers written in different language to communicate via Remote Procedure Calls (RPC).

To achieve the semantic transparancy, designers used the concept of stubs in RPC (Remote Procedure Call)

look like LPC (local procedure call). Stub mask the real RPC implementation from the application that interface with the underlying RPC systems.

2) Discuss about Open Source cloud software and explain the brief note on Eucalyptus and Open Nebulla.

An open source cloud is any cloud service or solution that is developed using open source technologies and software. This covers any public, private or hybrid cloud models providing SAAS, IAAS and PAAS that have been built and operate entirely on open source technologies. The below visual displays the advantages of using open source cloud software/service/solution.



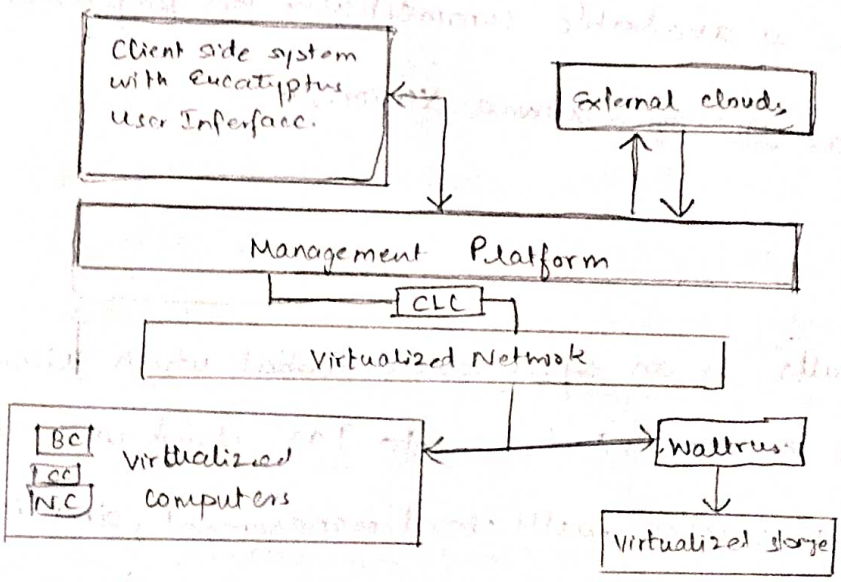
Some of the open source cloud services are Eucalyptus, Open Nebula, Open stack, Aneka, Cloudism, etc., Out of which Eucalyptus and Open Nebula is discussed below.,

(A) Eucalyptus :

Eucalyptus is a product from Eucalyptus system that was developed out of a research project at University of California, Santa Barbara. It ~~was~~ provides an AWS-compliant EC2-based webservice interface for interfacing with the cloud services. It also provides services such as Walrus, and an UI for managing user and image.

Eucalyptus Architecture

Eucalyptus takes many design queues from Amazon's EC_2, and it is an open source environment. The architecture was presented in Eucalyptus white paper. The architecture is displayed below.,



The system supports cloud programmers in VM image management as above diagramed, Essentially, the system has been extended to support the developement of both

computer cloud and storage cloud.

VM Image Management :

Eucalyptus takes many design queues from Amazon's EC2, and its image management system is no longer different. Eucalyptus stores image in Walrus, the block system of storage that is analogous to Amazon S3 services.

As such, any user can bundle their own root file system, upload and then register the image and link it with a particular kernel or ram disk image.

This image is uploaded into a user-defined bucket with Walrus and can be retrived anywhere from any availability zone. This allows users to create specialty virtual applications and deploy them within Eucalyptus with ease.

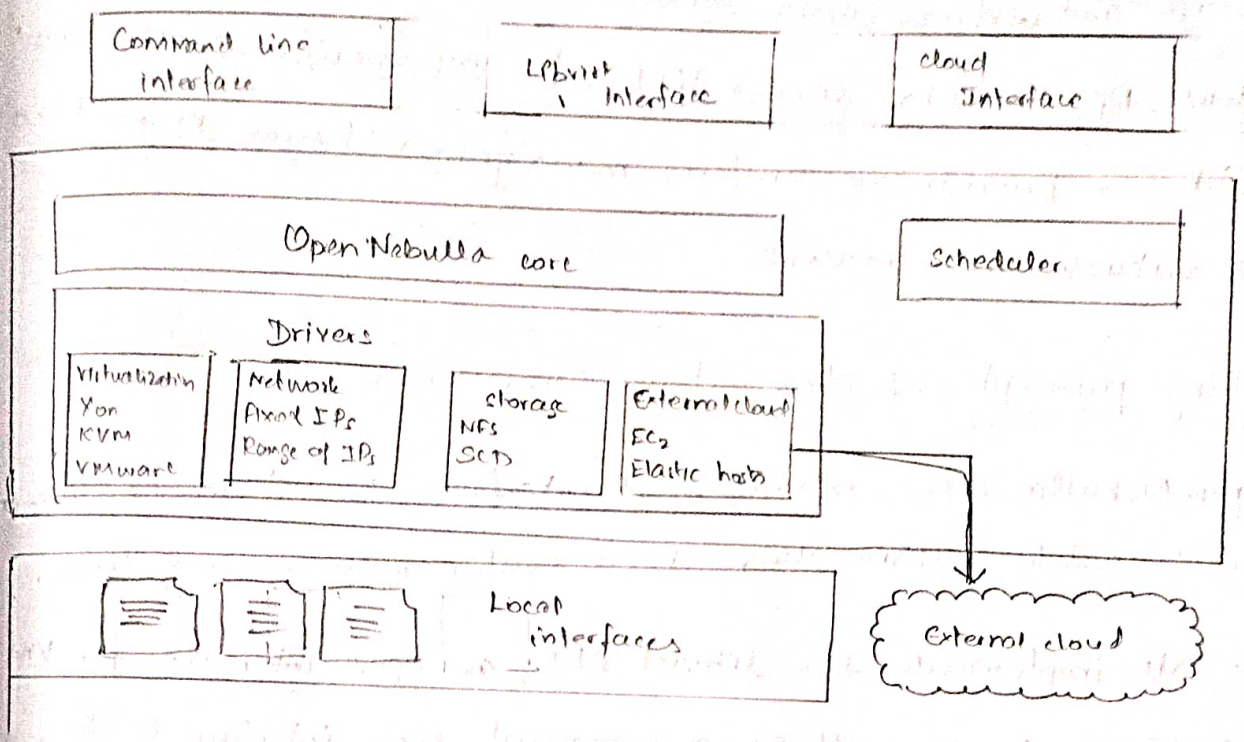
Eucalyptus is available commertially on proprietary version, as well as the open source version.

(B) Open Nebula :

Open Nebulla is an open source toolkit which allows users to transform existing interfaces into IaaS cloud with cloud-like interfaces. It enables multi cloud management, simplicity, Integeration, Community and commodity support and helps users in various scenarios and application areas.

Architecture :

The architecture of open Nebulla has been designed to be flexible and modular to allow integeration with different storage and network interface configurations, and hypervisioner technologies. The simplified diagram is exhibited below.,



* Open Nebulla Core : The core is a centralized component that manages VM full lifecycle, including.,
  - → setting up network dynamically for group of VMs
  - → Managing their capacity storage ~~managem~~ requirements such as VM disk image deployment or noon-the-fly software environment creation.

* Scheduler : Another important component is the storage capacity manager or scheduler. It governs the functionality provided by the core. The default capacity of a scheduler is a requirement or rank matchmaker. However, it is also possible to develop more

complex schedule policies, through a lease model and advanced restrictions

* Access Drivers: The last main components are the access drivers. They provide an abstraction of the underlying interface and infrastructure to expose the basic functionality of the monitoring, storage and virtualization services available in the cluster. Thus, OpenNebula doesn't tied up to any specific environment and can provide an uniform management layer regardless of virtualization concepts.

Working principles of other exterior components:

* OpenNebula offers management interfaces to integerate the core's functionality within other data center management tools.
* It also implements the libvirt API, an open interface for VM management, as well as a command-line interface (CLI). The subtle of this functionality is exposed to external users through a cloud interface.

Salient features:

* OpenNebula is able to adapt to organizations with changing resource needs, including addition, or failure of physical resources.
* Live migration and VM snapshots are the features to support changing environments.

* When local ~~stor~~ resources are insufficient, OpenNebulla can support a hybrid cloud model by using cloud drivers to interface with external clouds.

This lets ~~loc~~ organization suppliments the local ~~interface~~ infrastructure with computing capacity from a public cloud.

* OpenNebulla includes EC_2 driver, which can submit request to Amazon EC_2 and Eucalyptus as well as ElasticHost driver.

* An Image Repository allows users to easily specify disk images from a catalog without worrying about low-level disk configuration attribute or block device mapping.

28-10-2023 II^M SEMESTER ASSIGNMENT -2 23331MCA0058

2 MCA Cloud Computing Technologies DMCB202

1) Google company offers both Software as a Service and platform as a services solutions in cloud computing. Assume your company must deploy Java and PHP solutions. Discuss how your company might use Google App Engine and company potential cost. Discuss the Pros, cons and compare with other popular cloud service providers such as Azure cloud services.

To deploy Java and PHP solutions in the cloud, your company can consider using Google Cloud Platform (GCP) with Google App engine for perform ~~soft~~ as a service (Paas). Here's how we might use Google App Engine and their potential costs.

Using Google App Engine for Java and PHP:

Google App engine is a fully managed platform for develop and deployment of applications. Our company uses this, based on salient supports system below:,

1) Java and PHP support:

Google App engine supports both Java and PHP. You can deploy Java applications using standard servlet container, and for PHP applications, it provides an environment to run PHP code.

2) Scalability :

App Engine automatically scales your application based on traffic. You don't need to worry about provisioning or managing infrastructures, Google handles for you.

3) Data Storage :

Google cloud offers various data storage option, such as google cloud SQL for relational databases and Google Cloud Datastore for NoSQL storage. These can be integerated with our java and PHP applications.

4) Developement frameworks :

You can use popular frameworks for Java and PHP deployment, it make easy to build and deploy application.

Potantial Costs :

The cost of using Google App Engine depends on factors like application usage, data, storage and traffic. Google offers a free tier with limited resources, but beyond that, you will be charged based on actual usage. Costs can vary widely based on your applications demands, so its important to use the google cloud pricing calculator to estimate your specific expences.

There are some prose and cons for using Google App engine on our company to choose for deploymont in java & PHP.

Prose of Google App Engine :

1) Managed ~~reso~~ services : Google App Engine is fully managed, which means you don't have to worry about server management, patching or scalability.

2) Auto Scaling : It automatically scales your application based on demand, ensuring optional performances.

3) Integeration : It integerates well with other Google cloud services, makes it easier to build comprehensive solutions.

Cons of Google App Engine :

1) Vendor Lock-In : As with any cloud platform, there's degree of vendor lock-in, which can make it challenging to migrate to another provider in the future.

2) Limited Language support : While Java and PHP are supported, if your company uses other languages, you might need to consider different option.

Comparison of Google App Engine with other cloud platforms :

* Amazon Web Services :

AWS provides a wide range of services, including elastic beanstalk for PAAS. AWS Elastic Beanstalk is similar to Google App engine and supports Java and PHP. AWS provides a broaded array of services and has a large market share.

* Microsoft Azure :

Azure provides Azure App Services, which is similar to Google App Engine and supports Java and PHP. Azure provides strong integeration compare to Google Engine, with various Microsoft technologies.

Ultimately, the choice between GCP with Google App Engine, AWS or Azure will depend on our company's specific needs, existing technology stack, cost considerations. Each has its strengths and may be a better fit for different use cases, It's advisable to conduct a thorough evaluation and potentially run pilot projects to determine which cloud provider allignes best with our requirement.

2) Social Network is a Saas Solution? Justify whether or not social network as Saas cloud computing solution? Include a description of the most appropriate deployment model for a social media site such as 1) Facebook.

2) Twitter
3) Linked In
4) Google plus +
5) Tagged
6) Flicker.

Social Network as Saas : A comprehensive Analysis

Introduction :

Social Network have become an integral part of our lives, connecting individuals and facilitating communication and sharing of content. In recent years, many social network have traditioned to cloud services, making them prime candidate for Saas deployment. Let us justify why social network fits the cloud computing ~~platform~~ infrastructure of on SAAS.

Justification for Software Network as SAAS :

1) Accessibility : Social network are accessed over the internet, making them inherently cloud-based. Users can access their accounts and data from anywhere ~~to~~ with an internet connection, without the need for local software installations.

2) Multi-Tenancy : Social Network serve millions of users simultaneously, indicating a multitenant architecture, which is characteristic of SAAS. Different users share the same software instance hosted in the cloud.

3) Subscription-based : While many social networks offer free access, they often provide premium feature or ad-free experiences for a fee. This subscription-based revenue model align with the SAAS model.

4) Automatic Updates : Social Networks are updated regularly, and user typically don't need to worry about manually installing updates. The services provider handle software.

5) Scalability : Social network must handle massive user based and varying workloads. They use cloud infrastructure to scale resources up or down on demand.

Analysis of deployment models in Social Networks:

i) Facebook:

Facebook is a social media network application that has benefited with a public cloud deployment model., The following justifies the ~~reson~~ reason:,

1) Scalability : It experiences massive variations in user traffic and content uploads through the day, public cloud provides after the scalability needs to handle these functions effectively.

2) Global research : Facebook has ~~user~~ global user base, and a public cloud's vast network of data centers around the world helps user to access the content for users across different geographical regions:

(ii) Twitter:

Twitter is also a public social media network and is hosted on its own architecture, usually in public cloud. It follows a multi-tenant architecture, and tweets are accessible in public

(iii) Linked In :

LinkedIn is a professional network used for business purposes. It typically operates in a public cloud, allowing open access to most of its feature, but also offers premium subscription services for enhanced feature.

(iv) Google Plus :

Google plus, which is discontinued 2019, was a public social network deployed by google. It operated in the public cloud as a part of Google suite's of services.

(v) Tagged :

Tagged is a public social network, similar to facebook and Twitter, and its typically hosted in a public cloud environment, providing access to a broad user base.

(vi) Flicker :

Flicker is a photo-sharing platform that allows user to store and share photos. It is allowed to be hosted in a public cloud and follows SAAS model, where users can access and manage their photoes online.

Conclusion:

Social networks can indeed be considered (SaaS) software as a service solutions. These platform align with the key ~~characteris~~ characteristics of SAAS, including accessibility, scalability, regular updates, data storage and user management.

23-10-2023 | IInd SEMESTER ASSIGNMENT - 1. | 2232 MCA 0058

ARTIFICAL INTELLEGENCE AND MACHINE LEARNING

MCA | DMC 6203

1. Eloberate how an agent interacts with environment. Give its type of Agents.

An agent is anything that can be viewed as perceiving its environment through effectors.

* Human → Agent → eyes, ears, and other organs for sensors
→ other body parts for effectors.

This enables making decisions and takes action to acheive specific goals. These agents are designed to exhibit intellectual behaviour and adapts to change the circumstances.

An environment refers to the external context within which AI agent can operate. It provides the necessary stimuli and agent feedback to percieve and interact to the world.

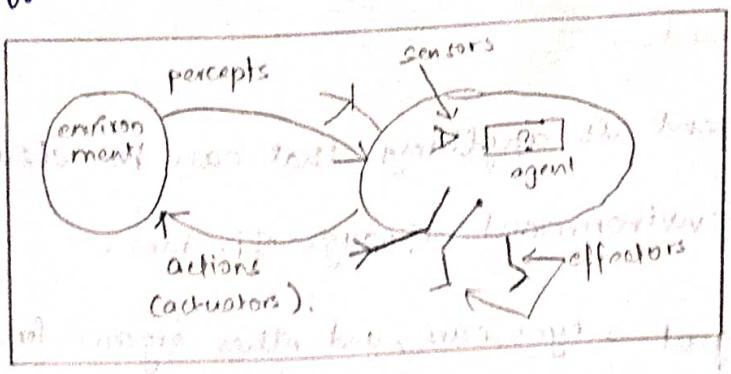
The interaction of agents on the environment depends on the below terminologies / systems ..

* Sensor is a system which detects the change in the environment and sends the information to other electronic devices. An agent observes ~~its sensor~~ its environment through its sensor.

* Actuators performs the action which converts energy into motion. The actuators are only responsible for moving and controlling a system. An actuators can be electric motor, gears, rails etc,

* Effectors are entity that affect the environment. Effectors usually appears as a physical entity.

The generic agent interaction with environment through sensors and effectors in the way as on below diagram,



Structure of an AI agent:

This explains how agent enables itself work on the environment. The task of AI is to design an agent program which implements the agent functions. Before we design an Agent functions, we must have a pretty good idea of the possible percepts and action, what goals or performance measure the agent is supposed to acheive, and what sort of environment it will operate in. These comes in a wide variety.

Agent Type (kind)	Percepts	Actions	Goals	Environment
Medical diagonistic system	symptoms, findings, patients's answ.	questions, tests, treatement	Health patient, min. cost.	patient, hospital.
Interactive english tutor	Typed words	print exercises, suggestions, correction	Maximize student's score on test	set of students.
Refinery controller	Pixels of varying intensity	pick up parts and sort into bins.	place parts in correct bins.	conveyor belt with parts.

Agent type	Percepts	Actions	Goals	Environment
Taxi Driver	camera, speedometer, GPS, sonar, microphone	steer, accellerate, break, talk to passanger	safe, fast, legal, trip, maximum profits	roads, other traffic, customer, pedestrians.

Agent Function / Program skeleton:

An agent function or program will use some internal data structure that will be updated as new percepts arrive. These data structure are operated on by the agent's decision-making decision procedures to generate an action choice, which is then passed to the architecture to be executed.

```
function SKELETON-AGENT percepts return action
  static : memory, the agent's memory of the world
  memory -> UPDATE MEMORY (memory, percept)
  action -> CHOOSE-BEST ACTION (memory)
  memory -> UPDATE MEMORY (memory, action)
return action.
```

Example: Designing an automated taxi driver:

Let us consider "designing an automated taxi driver" as an example of agent and its interaction with percepts, action, goals and environment. The table is based on 3rd case on the above table.

The taxi will need to know where it is, what else on the road and how fast it is going. This information can be obtained from percepts (cameras, speedometer, GPS). To control the vehicle properly, it should have a actuator (accellerator). It will also

need to know the mechanical state of the vehicle, so it will need the usual array of engine and electric system sensors.

It might have instruments that are not available to average human driver: GPS to give it accurate position information. Finally the microphone or keyboard for passangers to tell it their destination.

The action will be more or less same as on normal car drivers.

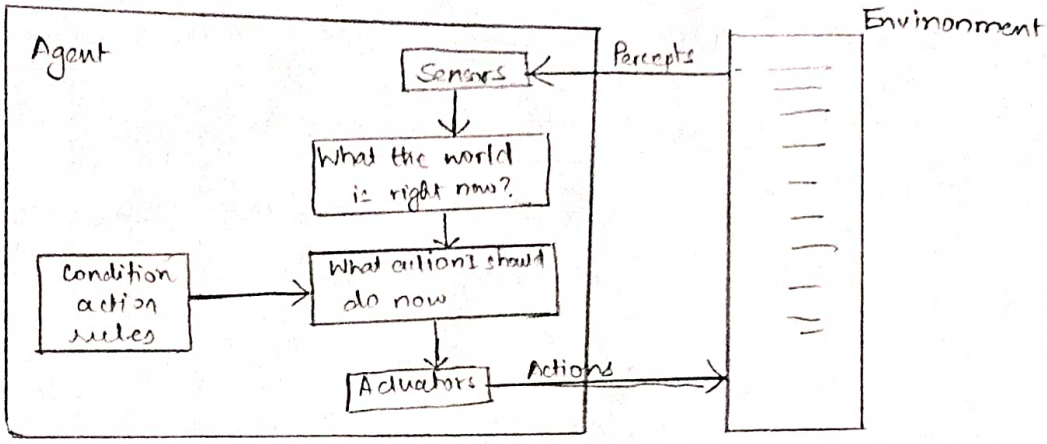
It will additionally need output to a screen or voice syntarizer to talk back to passanger, and perhaps some way to communicate with other vehicles.

Types of Agents:

Agents can be classified into four types based on their degree of perceived intellegence, ~~and~~ capability and agent functional complexity.

1.) Simple Reflex Agent:

Simple reflex Agent ignore the rest of the percept history and based on the current percept. Percept history of all that agent perceives the history up to date. The agent function is based on condition-action rule. This agent function only succeeds when the environment is fully observable. For simple reflex agent operating in particularly observable environment, infinite loop, is often unavoidable.



Problems with Simple reflex agents are:

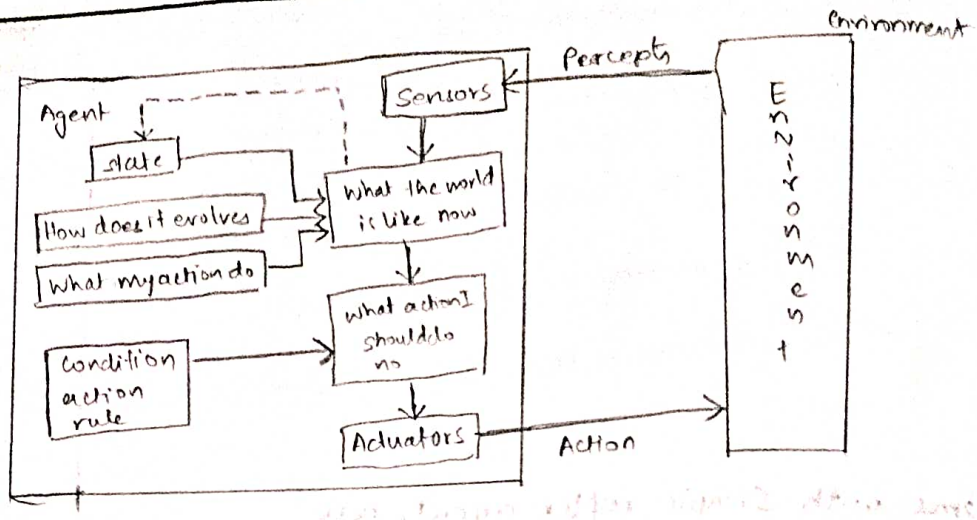
- Very limited intellegent
- No knowledge of non-perceptual parts of the state
- Usually too big to generate and store
- If there occurs any change in the environment, then the collection of rules need to be updated.

2.) Model Based Reflex Agents:

It works by finding a rule whose condition matches the current situation. A model-based agent can be handle partially observed environment by the use of a model about the world. The agent has to keep track ~~to~~ of the internal state which is adjusted by each percept and that depends on percept history. The current state is stored inside the agent which maintains some kind of structure of describing the part of world which cannot be same.

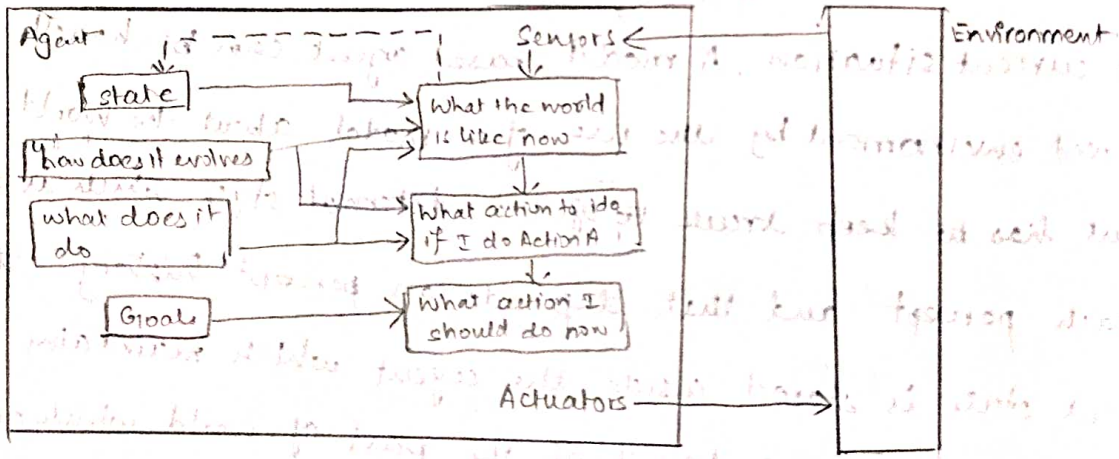
Updating the state requires information about:

- How the world evolves in dependantly from the agent?
- How does the agent's action affect the world?



3) Goal Based Agent :

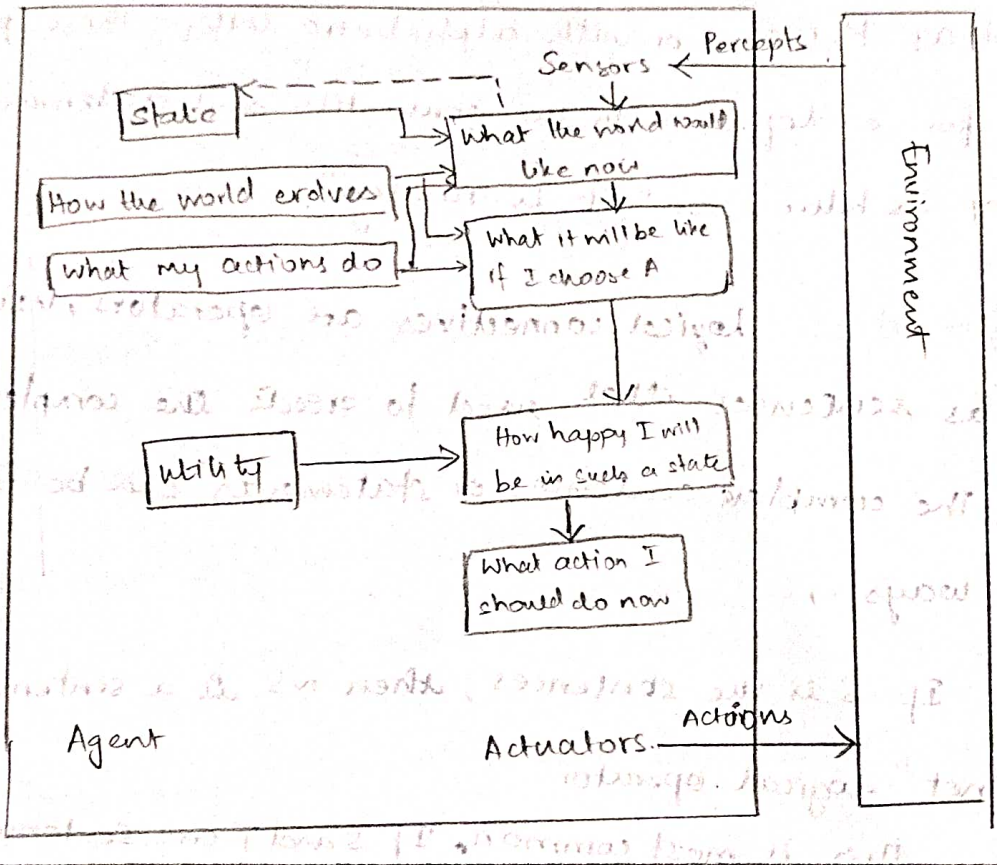
These kinds of ~~ki~~agents take decisions based on how they are currently from their goal. Their every action is intended to reduce their distance from the goal. This allows the agent a way to choose among multiple possibilities, selecting one which reaches the goal state. They usually require research and planning and however, the agent's behaviour can be easily behaved.



4) Utility Based Agents :

The agents which are developed having their end uses as building blocks are called utility based agent are used. They choose action based on the preference (utility) for each state. The 'utility' describes how happy the agent is. Because of the

in the world, an utility agent choose the action that maximizes the expected utility. A utility function maps a state onto a real number which describes the associated degree of happiness.



2. What are the basic components of proportional logic? Explain with an example.

Proportional logic, also known as calculus or semetic logic, is a fundamental component AI and represents the simplest form of logic. It deals with proportion, which are statements that can be (either true or false), boolean. It serves to illustrate many of the concepts of logical just as well as first-order logic. Their basic components are discussed below.,

Components:

(1) Proportional (or variables): Proportions are the atomic statements that can be either true or false. They are represented as variable, often denoted as P, Q, R... or with alphabetic letter. These proportions can stands for or defines with the semantics and statements such as "The sky is blue" or "It is raining".

(2) Logical Connectives: Logical connectives are operators, usually represented as sentences that used to create the complex statements. The complex sentences or statements can be formed in below two ways.,

(i) Unary: If S is the sentences, then ~S is a sentences, where '~' is "not" logical operator.

(ii) Binary: This is most common. If S and T are sentences, then, (S o T) is the complex sentence with below four logical connectors (denoted by o).

~~(a) Conjuction (AND, ∧)~~:

(a) Conjuction (AND, ∧) : P ∧ Q : P and Q

(b) Disjuction (OR, ∨) : P ∨ Q : P or Q.

(c) Implication ('implies', →) : P ⇒ R : P implies R.

(d) Biconditional ("if and only if", ⇔) : P <=> R : P occurs if and only if 'R'.

Example:

Let's use proportional logic to create an simple example.

Consider the following defined proportions:,

P: "The sun is shining"

Q: "It is a clear day"

We can use these clear propositions and logical connectives to represent a more complex statement:

Statement A: "The sun is shining and It is a clear day".

This can be logically converted to PL as,

$A \Rightarrow P \wedge Q$.

This means that A is true only if both P and Q are true, that is the sentence defined in A.

Let us consider another Instance example:,

Let P: 'It is hot' ; Q: 'It is humid', R: "It is raining"

So,

"If it is hot and humid, then it is raining" $\longrightarrow (P \wedge Q) \Rightarrow R$

Similarly,

$Q \Rightarrow P$: If it is humid, it is hot

3. Explain the steps in Bayesian learning with an example.

Bayesian learning is a probablistic approach to machine learning that uses Bayesian inference to update our beliefs about model parameters based on observed data. It is particularly useful for modeling uncertainty and is commonly used in various fields, including statistics and artificial intellegence.

Bayesian learning uses Bayes theorem to determine the conditional probability of a hypothesis given some evidences or observation. Bayes theorem provides the way to calculate the hypothesis 'h' based on its prior probability P(h), it is the probabilities of obtaining various data given the hypothesis and observed data itself.

The collectively steps for implementing bayesian learning is summarized below,

(i) Define a prior probability distribution:

Initiate, as mentioned earlier, by defining a prior probability distribution over the model parameters. The prior distribution represents your beliefs about the parameter before observing any data. It encapsulates any prior knowledge or assumptions you have about the parameters. The prior probability distribution is denoted by $P(\phi)$.

(ii) Data Collection:

Collect a dataset D of observations, which is typically represented as a set of data points. for supervised learning, this data includes input-output pairs (x, y). In the similar fasion to (i) step, write P(D) to denote the prior probability that the training data D is observed.

(iii) Define a likelihood function:

Specify a likelihood function, P(D|h), which represents the probability of observing the data given a particular set of model parameters, that which hypothesis 'h' holds. More generally, we can write P(x|y) to denote the probability of x given y. This likelyhood describes the relationship between the params and the observed data.

(iv) Calculate the posterior Probability distribution:

In machine learning problems, we are interested in the probability P(h|D) that h holds given the observed training data D. This P(h|D) is called as posterior probability of h, because it holds confidence that h holds after we have seen the training data D, in the contrast to the prior probability P(h), which is independant of D.

(v) Apply Baye's Theorem to ~~prior~~ on $P(h|D)$:

Bayes' theorem provides the way to calculate the posterior probability $P(h|D)$, from prior probability $P(h)$, together with $P(D)$ and $P(D|h)$.

$$\boxed{\text{Bayes' theorem: } P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}}$$

In many learning scenarios, the learner considers some set of hypothesis H and is interested in finding the most probable hypothesis $h \in H$ given the dataset D. Any such maximally probable hypothesis is called a maximum posteriori hypothesis (h_{MAP}). This can be computed by,

~~$h_{MAP} = \text{argmax}\, h \in H$~~

$$h_{MAP} = \arg_{max}\, h \in H \,.\, P(h|D)$$

$$= \arg_{max}\, h \in H \,.\left[\frac{P(D|h) \,.\, P(h)}{P(D)}\right]$$

$$\boxed{h_{MAP} = \arg_{max}\, h \in H \,.\, P(D|h) \,.\, P(h)}$$ dropping $P(D)$ as D is constant.

(vi) Make Inferences

Once you have the posterior distribution, you can make inference about the model parameters. Common inferences include calculating point estimates and credible estimate intervals. Bayesian learning provides a full distribution over the parameters.

Example:

Consider a medical diagonistic system in which there are two alternative hypothesis.

(1) The patient has covid

(2) The patient has not.

The available data is from the particular laboratory test with two possible outcomes, positive (+) and negative (-). We have prior knowledge that,

→ over the entire ~~problem~~ population of people, only 8% have the disease.

After tests,

→ The correct positive results in only 0.98% of the cases in which disease are actually present.

→ The correct negative results in only 97 of cases in which disease are not present.

Suppose we observe a new patient from whom lab test returns positive result, our problem is to determine ~~the~~ whether ~~the~~ we can ~~patt~~ diagonize the patient as having covid (C) or not.

The above problem summarizes the below,

$P(C) = 0.08$ $\quad$ $P(C') = 0.992$

$P(+|C) = 0.98$ $\quad$ $P(-|C) = 0.02$

$P(+|C') = 0.03$ $\quad$ $P(-|C') = 0.97$

The hMAP can be found using.

$$P(c|+) = \frac{P(+|c) * P(c))}{(P(+|c) * P(c)) + (P(+|c') * P(c')} = 0.209.$$

While $P(h|D)$ of covid is significantly higher than $P(D|h)$, the most probable hypothesis is still that the patient does not have covid (hypothetically).

4) Discuss about forward chaining and backward chaining with an example.

The inference engine is the component of the intelligent system in artificial intelligence, which applies logical rules to the knowledge base to infer new information from known facts.

Inference engine has commonly proceeds into two modes,

1. forward chaining 2- Backward chaining

(A) forward chaining:

It is also known as forward ~~direct~~ deduction or reasoning method when using an inference engine. It is a form of reasoning which starts with atomic sentences in the knowledge base and applied inference rules (Modus Ponens) in forward direction to extract more data until the goal is satisfied.

The forward chaing algorithm starts from :,

(i) known facts,
(ii) Triggers all rules whose premises are classified
(iii) Add their conclusion to the known facts.
(iv) These process repeats untill the problem is solved.

Properties of Forward chaining :

* It is a down-up approach, as it moves from bottom to up
* It is a process of making a conclusion based on known facts or data, by ~~stro~~ starting from the initial state and reaches the goal state.
* This approach is ~~also~~ also known as data-driven as we reach to the goal using available data.
* It is commonly used in expert systems, such as clips, business and production rule systems.

Example :

Consider the following example,

" As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some hostiles missiles, and all the missiles were sold ~~by~~ to it by Robert, who is an American citizen". The problem is to,

Prove that "Robert is an criminal"

To solve this,
We will convert all the above facts into first-order definite clauses, and then will proceed with this approach.

Facts → First-Order clauses :

- "It is a crime for an american to sell weapon to hostile nations" (Let say p, q, r, are propositions).

  Americans (p) ∧ weapons (q) ∧ ~~sells~~ hostile (r) ∧ sell (p, q, r) → criminal (p). — ①

- "Country A has some missiles"

  ∃p Owns (A, p) ∧ Missile (p). — ②

  → can be writen in difine clauses with T_1 as constant.

  Owns (A, T_1) — ②

  Missile (T_1) — ③

- "All of the missiles were sold to country A by Robert".

  ∃p missiles (p) ∧ Owns (A, p) → sells (Robert, p, A) — ④

- "Missiles are weapons"

  Missiles (p) → weapons (p) — ⑤

- "Enemy of America is known as hostile

  Enemy (p, America) → Hostile (p). — ⑥

- Country A is enemy of American.

  Enemy (A, America). — ⑦

- Robert is American

  Robert (America) — ⑧.

Forward chaining proof:

① Start with the known facts and choose the sentences that doesn't have implications.

American (Robert) | Missile (T_1) | Owns (T_1, T_1) | Enemy (A, America)

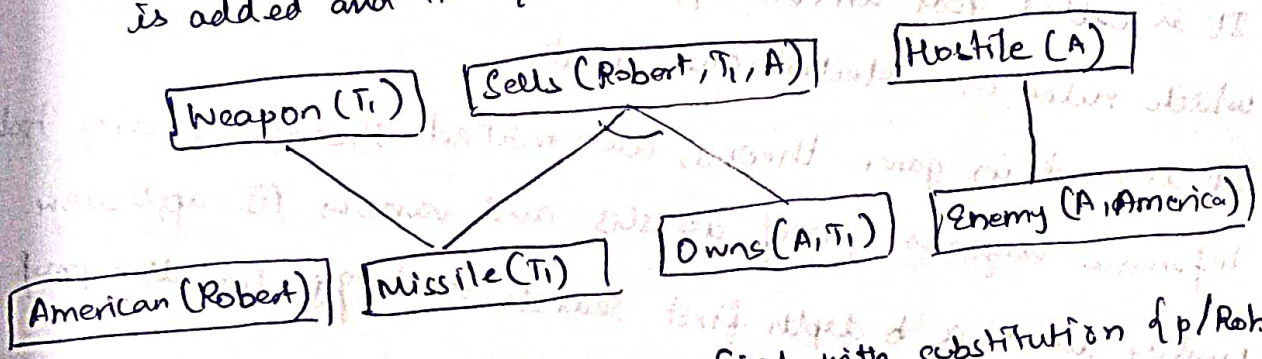
② Look on those facts which infer from available facts and with satisfied premises.

eq. ① doesnot satify the premises, hence cannot be added

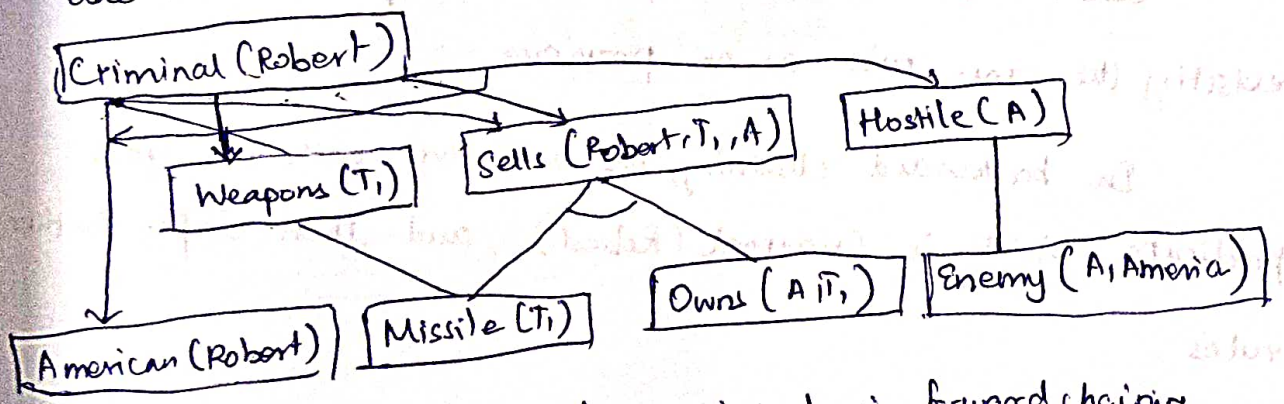
eq ② & ③ are already added.

eq. ④ satisfy the substitution {p/T_1}, so sells (Robert, T_1, A) is added, which infer from conjunctions of eq. ② & ③.

eq. ⑥ is satisfied with the substitution (p/A), so hostile (A) is added and it infers from eq ①.



③ We can now check eq ① is satisfied with substitution {p/Robert, q/T_1, r/A}, so we can add Criminal (Robert) which infers all the available facts. Hence reached the goal statement.



Hence proved that Robert is criminal using forward chaining.

(B) Backward chaining:

Backward chaining, A.k.A backward deduction or reasoning method when using an inference engine. A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find the known fact that supports the goal.

Properties:

* It is a top-down approach.
* It is based on modus ponens Inference rule.
* The goal is broken into sub-goals or sub-goals to prove the facts true.
* It is called goal-driven approach, as a list of goals decides which rules are selected and used.
* It is used in game theory, automated theorem proving tools, inference engines, proof assistants and various AI application.
* Mostly used a depth first search strategy for the proof of substantiation.

Example:

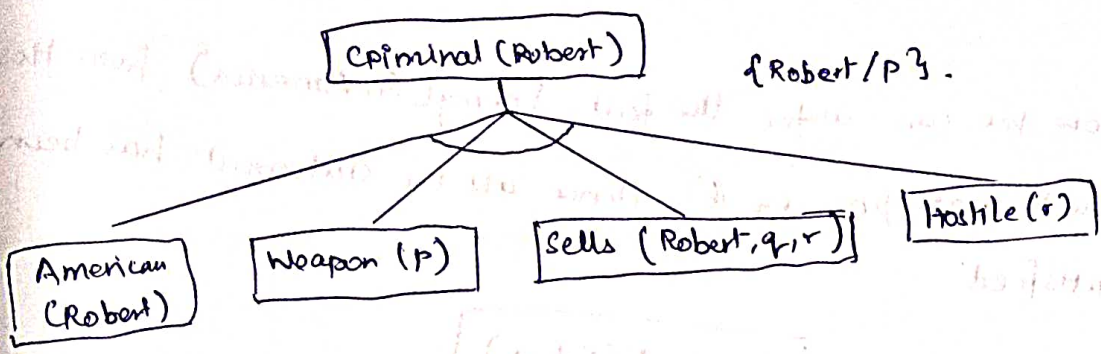
Consider the same example as on forward chaining, revisiting the same rule as on previous.

In backward chaining, we will start with our goal predicate, which is Criminal (Robert), and then infer further rules.

① Consider the goal fact. In this case, as we need to prove the statement, we consider, Criminal (Robert)

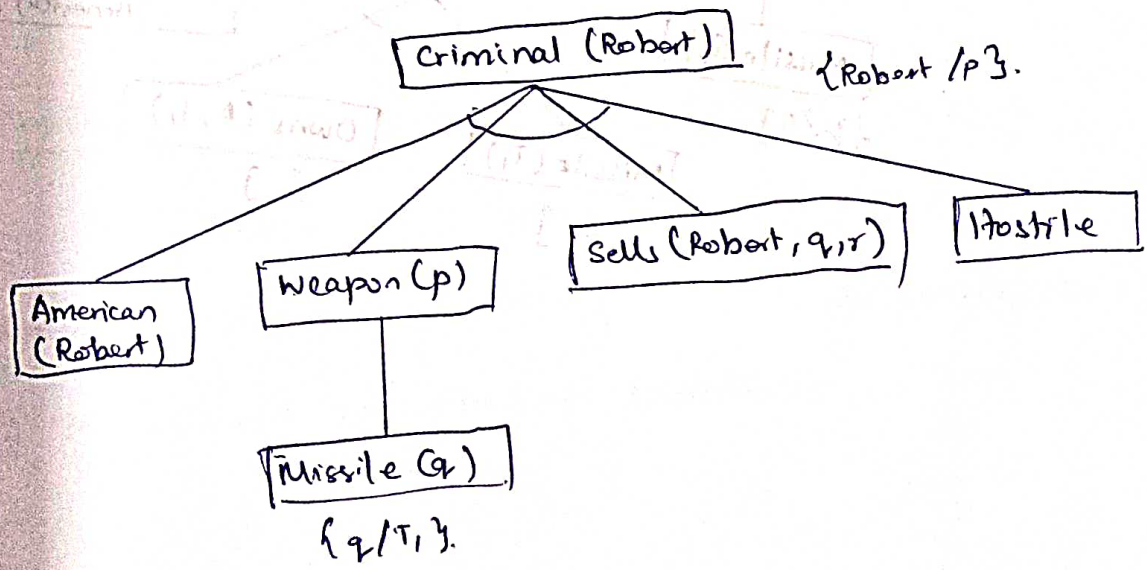
② Infer other facts from goal fact which satisfies the rule. So we can see eq.①, the goal predicate criminal(Robert) is present with substitution {Robert / p}. So we will add all the conjunctive facts below the first level and will replace p with Robert.

Here, American (Robert) is a fact, so it is proved here.

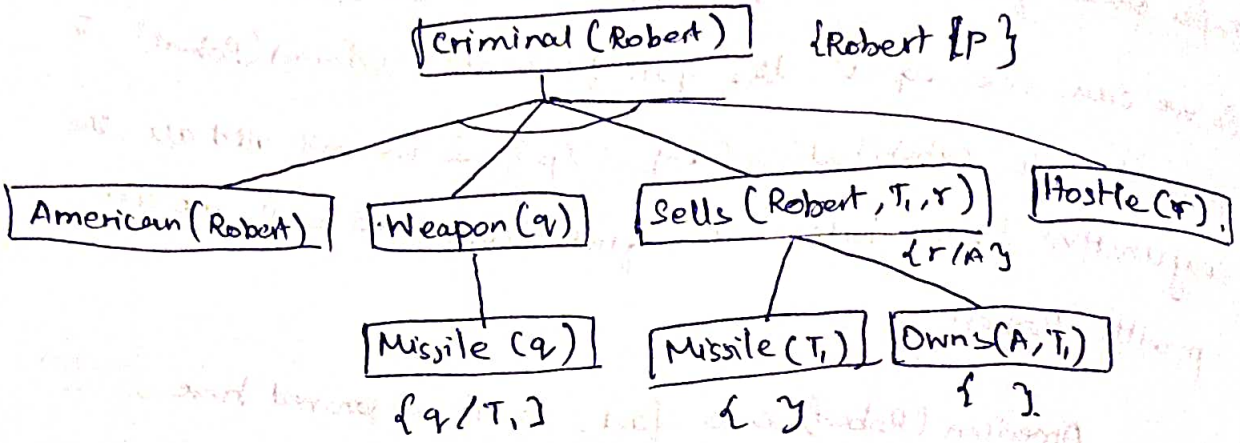


③ We will extract further fact ~~from or~~ from any ~~node~~ rule that infer from prescribed element to satisfy other rule.

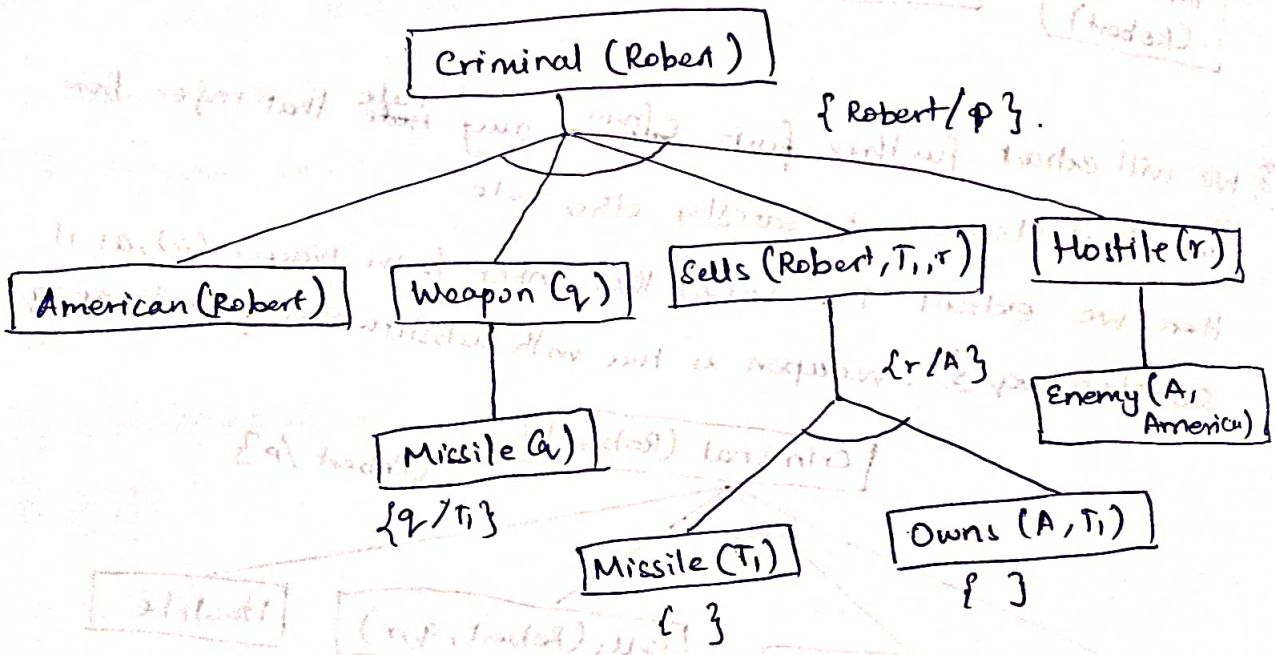
Here we extract Missile(q) that infer from Weapon (p), as it satisfies ~~eq~~ ⑤. Weapon is true with substitution of T_1 at q.



④. Now, infer Missile (T_1) and Owns (A, T_1) from Sells (Robert, T_1, r) which satisfies eq. ④, with substitution of A in place of r. Here we can prove two statements.



⑤ Now we can infer the fact Enemy (A, America) from Hostile(A) which satisfies eq. ⑥. Hence all the statement has been satisfied.



26-04-2023

IInd SEMESTER ASSIGNMENT - 2

2MCA

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING.

2232MCA0058

DMC6203

1) Discuss about the steps in classification using Decision tree with an example.

Decision tree is an supervised learning technique that can be used for both classification problems, where internal nodes represents the feature of the dataset, branches represents the decision tree rules and each leaf node represents the ~~the decision~~ outcome.

It is a graphical representation for getting all the possible solutions to a problem / decision based on given condition.

Purpose:

- ✓ Usually mimic human thinking ability while making decision, so it is easy to understand.
- ✓ Logic behind the tree can be easily understood as it is in a tree like structure.

Generalized steps of classification using decision tree:

Below steps describes the generalized way of expressing, representing and classification to compile the process of creating and understanding decision trees.

Step 1 : Begin the tree with the root node, says S, which contains the complete data set.

Step 2 : Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step 3 : Devide the S into subsets that contains possible values for the best attributes.

Step 4 : Generate the decision tree node, which contains the best attribute.

Step 5 : Recursively make new decision tree using the datasets created in step 3.

Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as leaf node.

Decision Tree Algorithm - ID3 :

ID3 algorithm stands for Iterative Dichotomiser 3, is an classification algorithm that follows a greedy approach of building a decision tree by selecting a best attribute that yields maximum information gain (IG) or minimum entropy (H)

- Entropy (H(S)) : It is an measure of uncertainity in dataset S. In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split S. Entropy = 0 implies it is pure class.

$$\text{Entropy} = \begin{cases} \sum_i -P_i \log_2 P_i, & P_i \rightarrow \text{probability} \Rightarrow \text{impure class} \\ 0 & \Rightarrow \text{pure class.} \end{cases}$$

- Information Gain: IG(A) tells us how much uncertainity in S was reduced after splitting set S on attribute A. In ID3, info. can gain be calculated for each remaining attribute. Attribute with larger information gain is used to split the S set.

$$IG(A) = H(S) - \sum_{i=1}^{T} p(t).H(t)$$

T → Subsets created from splitting S by attribute A such that $S = t \in T \cup t$.

H(S) → Entropy of S

p(t) → proportion of number of elements into the no. of elements in set S.

H(t) → Entropy of set T.

Example :

Suppose you have a set of ~~anir~~animals and their features, and you want to create a decision tree to classify them as ~~ea~~ either "mammel" or reptile. Dataset might look like below lable.,

Animal	Fur	Scales	Gives birth	Class
Dog	✓	×	✓	mammal
Crocodile	×	✓	×	Reptile
Cat	✓	×	✓	Mammal

Animal	Fur	Scale	Birth	Class
Elephant	✓	×	✓	Mammal
Snake	×	✓	×	Reptile.

Now, entropy measures the impurity or disorder in the data

Entropy $H(S) = \sum_{i=1}^{n} -P_i \log P_i$

$= -p(\text{Mammal}) * \log_2(p(\text{Mammal}))$

$- p(\text{Reptile}) * \log_2(p(\text{Reptile}))$

$p(\text{Mammal}) = 3/5$ & $p(\text{reptile}) = 2/5$.

$= -(3/5) * \log(3/5) - (2/5) * \log(2/5)$

$\boxed{H(S) \simeq 0.971.}$

IG for fur $= H(S) - \sum_{i=1}^{T} P_i H(t)$

By calculation, we are getting,

IG for fur $= (0.971) - [(3/5) * 0.918 + (2/5) * 1.0]$

for scales $= (0.971) - [(2/5) * 1.0 + (3/5) * 0.918]$

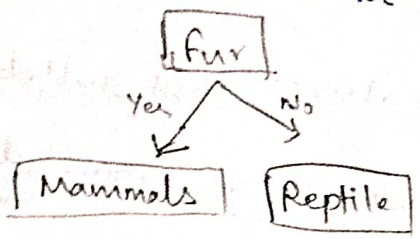
for Birth $= (0.971) - [(3/5) * 0.918 + (2/5) * 1.0]$

From the above information, fur might have highest information gain. Now, creating sub trees based on the feature (fur in this case and repeating ID3 algorithm for each branch.

This will continue recursively until one of the stopping conditions is met such as.,

- All instances in a subset belong to the same class.
- No features are available for splitting.
- A predefined path of the tree is reached.

The decision tree would be like the below.,



2) Explain the steps in classification using support vector machine with the example.

Support Vector Machine is a supervised machine learning algorithm that is used for both classification and ~~g~~regression analysis. The goal is to create the best line or decision boundry that can segregate n-dimentional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundry is called a hyperplane.

SVM chooses the extreme points/vectors that helps in creating the hyperplane. These extreme cases are called support vectors. Based on these information, we have discussed below steps with example on classification via SVM.

(i) Data Collection and processing:

Gather a labeled dataset with features and their corresponding classes for example, you might have a dataset ~~g~~ of flowers with features like petal length and width, Ensure the data is preprocessed, which may involve handling missing values, scaling features and encoding categories variables.

(4).

(i) Feature selection :

Select the relevant features that will be used to classification, In our flower example, you may decide to use petal length and petal width.

(ii) Model selection :

Choose the appropriate SVM kernel, common choises are linear, polynomial or radial basis function RBF kernel. The choise depends on the nature of your data

→ Linear : It can be used as a dot product between any two observations. The mathematical formula will appear as :,

$$k(x, x_i) = \text{sum}(x, x_i).$$

→ Polynomial : It is more generalized form of linear kernel and distinguish curved or non-linear inner ~~sp~~ space. The mathematical formula for this would be :

$$k(x, x_i) = 1 + \text{sum}(x * x_i)^d.$$

d → degree of polynomial.

(iv) Model Training :

Train the SVM model using our labeled dataset. The goal is to find the hyperplane that best separates the classes in your feature space. SVM algorithm will identify support vectors, which are the data point closest to the hyperplane.

(v) Hyper parameter training:

Tune hyperparameters like the regularization parameter (C) and kernel parameters to optimize the model's performance. This can be done through cross-verification.

(vi) Model Evaluation:

Evaluate the model's performance using various metrics like accuracy, precision, recall, F1-score and ROC curves.

You can split the data set into training and verifying sets or using some predefined techniques like k-fold cross-validation.

(vii) Prediction:

Once the model is trained and evaluated, you can use it to make predictions on new, unlabeled data.

For example, if you have a new flower with petal length and width measures, the SVM can classify it into one of the predefined classes.

Example:

Let us consider you have a dataset of this flower with features (sepal length, sepal width, petal length and patal width) and you want to classify them into three classes: "Iris Setosa", "Iris Versicolor", "Iris Virginica".

1) You preprocess the data and select the petal length and petal width as features.

2) Choose an RBF kernel for the SVM.

3) Train the SVM model on your labelled dataset.

4) Tune hyperparameters like C and the kernel parameter through cross-validation

5) Evaluate the model's performance using various metrics.

6) With a new iris flower's petal length and width measures the SVM can classify into one of the three classes.

This SVM finds the hyperplane that best separate the classes and it can be a linear or non-linear boundary, depending on the kernel been used.

3) List and briefly explain the logistic regression or gradient descent.

The goal with gradient descent is to find the optimal weights: minimize the loss function we've defined for the model, explicitly represent the fact that the loss function L is parameterized by the weights, which we'll refer to in machine learning in general as θ and in the case of logistic regression $\theta = w, b$.

Stochastic Algorithm:

Stochastic ~~Algorithm~~ gradient descent is an online algorithm that minimizes the loss function by computing its gradient after each training example, and nudging θ in the right direction i.e., the opposite direction of the gradient, which is presented below algorithm.,

function STOCHASTIC GRADIENT DECENT (L(), f(), x, y) returns θ

L → loss function
f → function parameterized by θ
x → {a : a ∈ training inputs}.
y → {a : a ∈ training outputs}.

$\theta \leftarrow 0$
repeat till done
for each traing tuple (x^i, y^i)
  Optional (for reporting):
    compute $\bar{y}^t = f(x^i, \theta)$
    compute the loss $L(\bar{y}^i, y^i)$
  $g \leftarrow \nabla_\theta L(f(x^i, \theta), y^i)$

  $\theta \leftarrow \theta - \eta g$
return θ.

* Computation of loss is used to report how how well we are doing on the current tuple. The algorithm can terminate when it converges (or when gradient norm <), or when progress halts.

* The learning rate η is a hyperparameter that must be adjusted.
* If it's too high, the learner will take step that are too large, overshooting the minimum of the loss function.
* If it's too low, the learner will take step that is too small, and take too long to get to the maximum.
* It is common to start with a highest learning rate and then slowly becomes decended, so that it is a function of the iteration k of training, ~~and then~~ s
* The notation ηk can be used to mean the value of the learning rate at iteration k.

Composed steps :

Here are the key steps in gradiant descent for logistics regression ;

1) Initialize parameters.
2) Compute hypothesis function
3) Compute cost function
4) update parameter.
5) Repeat iterations
6) Converage the change in cost function.
7) Final parameter recieval.

Example :

Let us walk through a single step of the gradiant descent algorithm. Generally in a word document synonym

analysis function, it seems a single observation x, whose correct value is $y=1$ (this is a positive review), and with only two features:

$x_1 = 3 \rightarrow$ counts of positive lexical words.

$x_2 = 2 \rightarrow$ count of negative lexical words.

Let us assume the initial weights and bias in θ_0 are all set of 0, and the initial learning rate η is 0.1:

$w_1 = w_2 = b = 0$;

$\eta = 0.1$

The single update step requires that we compute the gradient, multiplied by the learning rate.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(f(x^i, \theta), y^i)$$

In our mini example, there are three parameters, so the gradient vector has 3 dimentions, for w_1, w_2 and b. We can compute the first gradient as follows:

$$\nabla_{w,b} = \begin{bmatrix} \frac{\partial L_{CE}(\hat{y}, y)}{\partial w_1} \\ \frac{\partial L_{CE}(\hat{y}, y)}{\partial w_2} \\ \frac{\partial L_{CE}(\hat{y}, y)}{\partial w_2} \end{bmatrix} = \begin{bmatrix} (\sigma(w.x+b) - y)x_1 \\ (\sigma(w.x+b) - y)x_2 \\ (\sigma(w.x+b) - y) \end{bmatrix} = \begin{bmatrix} (\sigma(0-1)x_1) \\ (\sigma(0-1)x_2) \\ (\sigma(0-1)) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} -0.5x_1 \\ -0.5x_2 \\ -0.5 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix}$$

$$\boxed{\nabla_{w,b} = \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix}}$$

Now, that we have a gradient, we compute the new parameter vector θ_1 by moving θ_0 in the opposite direction from the gradiant:

$$\theta_1 = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} - \eta \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 0.15 \\ 0.10 \\ 0.05 \end{bmatrix}$$

So after one step of gradiant descent, the weight have shifted to be

$w_1 = 0.15 \quad w_2 = 0.10 \quad w_3 = 0.05$.

4) Explain in detail about the various activation functions used in neural network with example.

In neural networks, The elementary computation steps in each unit computes the new activation value for the unit by applying the activation function, g, to the result of input function a_i.,

$$\boxed{a_i \leftarrow g(in_i) = g\left(\sum_{j=1}^{n} W_{j,i} \times a_j\right)}$$

Different models are obtained by using different mathematical functions for 'g'. The popular activation functions are given below:

(i) Binary step activation function

(ii) Linear activation function.

(iii) Tan H (Hyperbolic Tangent activation function)

(iv) ~~Sigmoo~~ Sigmoidal activation function.

(v) ReLU (Rectified Linear Unit Activation function)

(vi) Leaky ReLU

(vii) PReLu (Parametric Relu).

Let us explain the concept and scope of ~~each~~ some activation function and example of one such activation function.

(i) Sigmoid Activation function:

The sigmoid function maps ~~may~~ any real-valued number to a value between 0 and 1. It's often used in the output layer of binary classification models.

Formula : $\boxed{\sigma_x = \frac{1}{1+e^{-x}}}$

~~Ep~~ Ilu

(ii) TanH activation function :

~~Tan~~ Hyperbolic Tangent activation function maps real-values numbers to a range between -1 and 1. It is similar to the sigmoid but centered at 0.

Formula: $\boxed{\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}}$

(iii) ReLU [Rectified Linear Unit] :

ReLU is a popular activation function that introduces non linearity by outputting the input for positive values and zero for negative values.

$$ReLU_x = \max(0, x).$$

(iv) Leaky ReLU

Leaky ReLU is an variant of ReLU that allows a small gradient for negative values, preventing neurons from dying during training.

$$\text{Leaky ReLU} = \max(\alpha * x, x)$$

$\alpha \rightarrow 0.01$ or any small positive constant.

(v) Parametric ReLU :

Parametric ReLU is similar to Leaky ReLU but allows the network to learn the slope of negative values rather than using a fixed constant.

$$PReLU_x = \max(\alpha * x, x)$$

$\alpha \rightarrow$ learnable parameter.

In the similar way, we have many activation functions. Now, lets look into the example with concept of Binary Step Activation function.

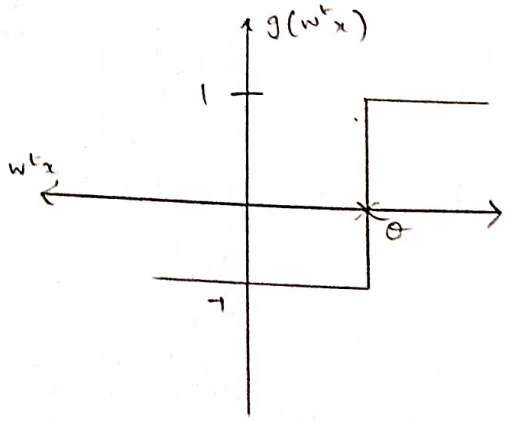
(vi) Binary step Activation function :

This allows network to produce a mathematical conversion

In which the results will be gradiated in two numerics (0, 1, -1). This is also shortly called as "step" function.

The step function has a threshold t such that it outputs as 1 when the input is greater than its threshold, and outputs as 0 otherwise. The biological motivation is that 1 represents the firing of the pulse down the axon, and 0 represent no firing.

The threshold represents the minimum total weight input necessary to cause the neuron to file.



Instead of having threshold for each unit, we add an extra input whose activation a_0 is fixed at -1. The extra weight $w_{0,i}$ associated with a_0 serves the function of a threshold at t, provides that $w_{0,i}$ $a_0 = -t$. Then all units can have a fixed threshold at 0. The mathematical expression of representing two thresholds are.,

$$a_i = step_i \left(\sum_{j=1}^{n} w_{j,i} a_j \right)$$

$$= step_0 \left(\sum_{j=0}^{n} w_{j0} a_j \right) = -1$$

where $w_{0j} = t$ and $a_0 = -1$

SET 32 - MCA

Name : HARYISH E.

Roll No : 2282MCA0058

Course : Master of Computer Application (MCA)

SUBJECT CODE

DMC 6205

SUBJECT NAME:

CYBER SECURITY

Marks	
Name and Signature (faculty)	

22.10.2023 IInd SEMESTER ASSIGNMENT-1 2232MCA0058

2 MCA CYBER SECURITY DMC6204

1. What do you mean by Policy? Why it is developed and revised?

A policy is the formal document or statement that outlines specific rules, guidelines or principles that an organization, institution, or government entity follows to acheive its objective or manage various aspects of its operations. Policies serves as a framework for decision making, governance and standardization within an entity.

These policies are designed to ensure the confidentiality, integrity and availability of data and systems while mitigating risks and addressing compliance requirements.

Reasons for the policies to be developed and revised:

There are many reasons for why the policies are developed and revised. Here are some of the common reasons for them:,

1. Risk mitigation:

Cybersecurity policies are developed to identify and mitigate potential security risks. They outline measures and controls that helps protect an organization's digital aspects from various ~~the~~ threats, such as malware, data breaches and unauthorized access of the system,

2. Compliance: Po

Policies are essential for ensuring an organization's compliance with the relevant laws, regulations and industry standards. Cybersecurity policies help align security practices with legal requirements, which is crucial to avoid penalties and legal issues.

3. Data Protection:

The Policies establishes guidelines for the secure handling of sensitive data. They dictate how data should be stored, transmitted and accessed to prevent data leaks or unauthorized disclosure.

4. Incident Response:

These policies provide a framework for responding to cybersecurity incidents. They outline the steps to be taken, in case of a breach, helping to minimize damage and recover quickly.

5. User Awareness and Training:

Policies can including provisions for user education and wellness regarding cybersecurity best practices. They ~~ensure~~ ensure that employees understand their roles in maintainance security.

6. Technology Usage :

Policies govern the appropriate use of technologies within an organization. This includes guidelines on software installations, device usage and network access.

7. Review and Updates :

Cybersecurity policies need to be regularly reviewed, reviewed and updated because the threat landscape is constantly evolving. New threads and vulnarabilities emerge and technology advances, making it crucial to adapt policies to remain effectives.

8. Auditing and ~~Accord~~ Accountability :

Policy provides a basis for auditing and holding individuals and dependancies accountables for cybersecurity responsibilities. They define role and responsibilities in the organization's security framework.

9. Continuous Improvement :

By revicing reviewing and updating policies, the organization can continuously improve their cybersecurity posture. Feedback from incidents and audits can be used to refine policies and ensure security measures.

In summary, cybersecurity policies are vital for establishing a secure digital environment.

Security policies are formal set of rules which is issued by an organization to ensure that the user who are authorized to access company technology and information assets comply with rules and guidelines related to the security information. On developing and servicing cybersecurity policies, user can experience the following benefits.

Key Benefits:

- By outlining security controls and best practices, policies help mitigate security risks.
- Policies detail how sensitive data should be handled, stored and transmitted, reducing the risk of data breaches and unauthorized access.
- Cyber security policies often includes provisions for ongoing user training. Educating and awaring users are likely less to fall victim to common security threads like phising.
- Effective security policies are help maintain an organization's reputation. A breach can have several reputational consequences, and having policies in place demonstrate a commitment to protect sensitive information.

2) Explain Email Policy ?

An Email Policy or Email security policy is a series of procedures governing the use of emails within a network or an establishment. It details how a category of users interacts with messages that are sent and recieved via email.

Working or functionality of Email Policy :

The default security strength of email isn't so strong. Messages sent via emails are in public space. Hence they can be easily accessed by anyone with average hacking skills. For an email security policy to be effective, it must include the following items :

1. The scope and purpose of the policy.
2. Information about the ownship of content contained in the emails
3. Privacy concerns and expectations of parties using the email.
4. Their responsibilities of the email users.
5. Guidelines for using the organization's email accounts.
6. Tips to detect and avoid email security threats.
7. Specific actions to ~~be~~ take in the events of a suspended email security breach.

Accessibility is the key in the successful implementation of this policy. Team members can only be abreast with the

information in the policy. If they can access the document.

It is advicable to use the workflow tool ~~to~~ with cloud storage and remote access. That may, authorized terms members can access the policy from anywhere at anytime.

Training is another essencial element to successfully implement an email security policy.

Building Effective Email Security policy:

An Email security policy isn't one-size-fit-all, but the cyber threats that endanger the use of emails have similar effects on organizations regardless of their offering and sizes. They are common standards that should be followed in building an standard policy.

Here are some tips of practice for building an email security policy that works.

(a) Adopt a template:

Creating an email security policy from scratch isn't bad idea, but could save yourself some time by adopting an existing template. This is necessary, especially if you aren't familiar with the content of the policy.

Instead of creating irrelevant information, you have vital information for creating a policy that works.

b) Modify the template :

Adopting an existing template doesn't mean you should use it the way it is. The template is to give you an idea of what the policy looks like.

Instead of considering everything in template hook line and sinker, adjust it to suite the unique needs your business. In the end, you'll have an original documents that is tailor made of your organization.

c) Identify User Engagement Terms :

Users of your email may engage in indiscriminate activities if they aren't aware that such activities are prohibited. It's your responsibility to expressly state how they should use your email.

Identify unhealthy email practices that may expose your network to cyber attacks and expose your network warns against involving such activities.

d) Implementing a tool :

Your email security policy is incomplete without implementing a tool that enhances the security of your email. Protection from security threads on email at manual band is insufficient. Match their energy with advanced tools like sandbox, spam filters and malware prevention software.

e). Enforce User Policy Acknowledgment:

The successful implementation of your policy begins with your user's willingness to abide by it. Change comes from some resistance. Team members who aren't familiar with an email security policy may decide to overlook it.

Get users to commit to using the policy by appending their signature. That way, you have proof of their aggrement to ~~use~~ it in case if they failed to.

f). Train users;

Email users may not understands ~~some important~~ some informations in the policy. Leaving them aside at the state of confusion is risky. Ensure to ~~train~~ make them understand the policy by conducting training.

g) Develope an incident Response plan.

Even with all the training on how to implement an email security policy effectively, things might still go wrong. Develop an incident response plan in the event of security breach. Your policy should contain what users should do once they suspect malicious attack or activity. Taking the right action can mitigate the effect of cyberattack.

Now let us look on how to cultivate healthy cyber culture with Email Security Policy.

Cultivate Healthy cyberculture :

Instant messaging may be trendy in communicating with friends and family. But when it comes to business and work, good old email is still relevant. It helps organizations to maintain a sense of order and formality.

You may not able to stop attackers from targetting your emails, but you can nullify their attacks with an email effective security policy.

CONCLUSION :

When everyone using your email understands how to keep how to keep the information safe, cybercriminals will have no opportunity to strike. It's a matter of time before they give up trying to penetrate your network and move on.

22-10-2023 | IIND SEMESTER ASSIGNMENT-2 | 2232MCA0056

2 MCA | CYBER SECURITY | 24MCE205

1) Eloberate Cybercrime play a vital role against person property and government to protect all valuable information and rights?

Cybercrime plays a significant role in posing threats to personal property and government interests, necessitating the protection of valuable information and rights. In the digital age, individuals, organizations, and governments are increasingly reliant on technology and data, making them vulnurable to various forms of cyber threats.

(A) Personal Property :

Cybercriminals often target individuals through activities such as identity theft, phishing and ransomware attacks. These crimes results in financial losses, reputational damage, and personal privacy breaches.

* Identify thefts : Cybercriminals thefts personal information to impersonate individuals, commit fraud, or access financial accounts.

* Phishing : This involves deceptive emails or websites that tricks users into revealing sensitive information, such as logical credentials or financial details.

22-10-2023 IIIrd SEMESTER ASSIGNMENT-2 2024MCA0058

2 MCA CYBER SECURITY SMC6205

1) Elaborate cybercrime play a vital role against person property and government to protect all valuable information and rights?

Cybercrime plays a significant role in posing threats to personal property and government interests, necessitating the protection of valuable information and rights. In the digital age, individuals, organizations, and governments are increasingly reliant on technology and data, making them vulnerable to various forms of cyber threats.

(1) Personal Property :

Cybercriminals often target individuals through activities such as identity theft, phishing and ransomware attacks. These crimes results in financial losses, reputational damage, and personal privacy breaches.

* Identity thefts : Cybercriminals thefts personal information to impersonate individuals, commit fraud, or access financial accounts.

* Phishing : This involves deceptive emails or websites that tricks users into revealing sensitive information, such as logical credentials or financial details.

* Ransomeware: Ransomware encrypts a victim's data, demanding a ransom for the decryption key causing financial losses and potential data exposure.

Protecting personal information, data and online assets is crucial to mitigate these risks.

(B) Government:

Government store a vast amount of sensitive information, from national security data to citizens' personal records. cyberattacks against government entities can have severe consequences consequences, including national security breaches and compromised citizen data. Safeguarding this information is vital to uphold the rule of law and protect citizen's rights.

(C) Valuable and confidential services:

Corporates and institutions also house valuable intellectual property, trade secrets and data protection. Cyber espionage and data breaches can lead to the theft or exposure of this information, causing financial harm and loss of competitive advantage. Protecting such valuable data is crucial for business continuity and innovation. The above mentioned threats are noted as below:,

* Data breaches: Unauthorized access to databases or networks can result in the theft or exposure of sensitive information, including personal, financial and medical records.

* Cyber Espionage: state-sponsered or corporate actors engage in cyber ~~espe~~espionage to steal sensitive data, intellectual property or government secrets.

(D) Rights:

Cybercrimes can infridge upon individual's rights, including privacy and freedom of speech. Cyber bullying, online harassment and censorship attempts are examples of how rights can be violated in the digital realm. Defending these rights is essential for maintaining a free and open internet.

Measurements taken against cybercrime:

To protect against cybercrime, severals measures can be taken:

(a) Cybersecurity department measures:

Implementing robust cybersecurity practices, including firewalls, encryptions and regular software updates, can enhance protection against cyber threats.

(b) Education and awareness:

Increasing awareness about cyber threats and best practices can empower individual and organizations to recognize and respond to potential attacks.

(c) Legislation and Regulations:

Government can enact and enforce cybercrime laws to deter criminals and prosecute them when necessary. International cooperation is often required to combat cross-border cyber crime effectively.

(d) Collaboration:

Public-private partnership can help in sharing threat intelligence and developing joint strategies to combat cyber threats effectively.

In summary, cybercrime poses a significant risk to personal property, government interests, valuable information, and rights. Protecting these assets and rights requires a multifaceted approach involving technology, education, legislation, and International cooperation.

2) Explain in what situation semiconductor law comes in consideration and how it differs from patent law:

Semiconductor law, or more specifically "mask work protection" under semiconductor law, comes into consideration in situations related to the protection of the unique layout design of integerated circuits (ICs). Here are some of the situations where semiconductor law is relevant:

(i) IC Design protection:

Semiconductor law is primarily used to protect the layout designs of integerated design circuits. When a designer or manufactures creates a unique ~~ag~~arrangement of transmitters, capacitors or other electronic components on a semiconductor chip, they may seek mask work protection to prevent ~~users~~ others from copying that specific layout.

(ii) Manufacturing and exportation:

Semiconductor law grants exclusive rights to reproduce, distribute and import the protection layout design. Manufacturers may use mask work protection to safeguard their designs from unauthorized duplication, especially in the context of global semiconductor protection and environment distribution among the region.

(iii) Intellual Property Rights:

Companies or individual involved in semiconductor industry, which is the critical sector for technology, and electronics may use semiconductor laws to estrabilish and protect their intellectual property rights. This is particularly Important when they invest significant resources in creating Innovative chip layouts.

(iv) Preventing Counterfeiting:

In the semiconductor industry, ~~conter~~ counterfeiting and unauthorized replication of ICs is a concern. Semiconductor law can be used to prevent counterfeiting from reproducing exact layout of ICs, helping to maintain quality and reliability in electronic products.

(v) Competitive Advantages:

Competitive companies may use mask work protection to gain a competitive advantages by safeguarding their investment in IC design. It allows them to retain control over the use of and distribution of their unique chip layouts.

(vi) Licensing and Royalties:

Semiconductor laws enables creators to licence their mask work rights to others, ~~gather~~ generating revenue through royalties or fees for the use of the protected layout design.

Key Differences of Semiconductor laws with Patent Laws:

By subject or content, Semiconductor laws specifically addresses intellectual property rights related to semiconductor devices, ICs design and layouts. It focuses on protecting the unique arrangement of electronic components on a micro chip; while Patent law focususes on border range of innovations, processes, machines, composition of matter and useful improvements to existing technologies.

Here are the key differences between patern and semiconductor law.,

(i) Protection:

Semiconductor law primarily involves the protection of mask works, which is protected by Semiconductor Chip protection Act in United States and similar laws in other contries. It is as specific arrangement of transistors and other elements in a semiconductor chip.

Patent protects the functional aspects and principles behind inventions, including semiconductor manufacturing processes, devices and applications. Patent offers broader protection.

(ii) Duration of protection:

Semiconductor law ~~prot~~ mask work protection is typically shorter, usually around 10 years in United states and may vary

in other countries. This protection is more limited to the scope but minimal work of renewal processes.

Patent law, generally provides protection for up to 20 years from the date of filing, granting inventors an exclusive rights to make, use, and sell their Invensions. Patent offers longer and more extensive protection.

(iii) Originality vs Novelty :

In Semiconductor law, to qualify for protection, the layout design must be original and the result of the creator's intellectual effort. It doesn't protect the functionality of the circuit, but rather the arrangement of its ~~ele~~element.

Whereas in Patent law, to obtain a patent, an invention must be novel, non-obvious, and useful. This means the inventions must me new, but not obvious to someone skilled in the field, have a practical applications.

(iv) Functional & Structural :

Semiconductor law protects the structure arrangement of components in ICs and while patent law can protect functional aspects and processes.

In summary, semiconductor law and patent has differs in scope, requirements and IPs they protect. At last, semiconductor law enhanses protection in limited field while patents can apply to various fields.

3) Write a short note on the following:
a) Patent law b) Copyright law c) IPR.

(a) Patent law:

Patent law is a legal framework that provides inventions for a specific period, ~~stop~~ with exclusive rights to their inventions for a specific period, typically 20 years from the date of filing. The primary purpose of patent law is to encourage innovation by rewarding inventors for their creative efforts. By patent law granting inventors a limited monopoly, the law provides the financial incentive to invest in research and a developement.

There are different types of patents and some of the pathent laws are.

→ Utility patents: It covers new and useful processes, machines, articles of manufacture and composition of matter.

→ Design pattents: It protects the ornamental design of a functional item.

→ Plant patents: It protects new and distinct plant varieties.

b) Copy write law:

Copywrite law protects original creative works such as literature, music, art, software, and more. It grants creators exclusive rights to their creations and is designed to

encourage and protect artistics and literacy expressions. This includes the following aspects.

→ The protection of copyrights is automatic upon the creation of a work and does not require formal registration. It covers tangible expression of ideas rather than the ideas themselves.

→ Copyright holders have the exclusive rights to reproduce, distribute, perform, display and adapt their work.

→ The protection lasts for the life of the author + 50-20 yrs. This may vary between countries.

~~(b) Intellectual property rights~~ :

→ Copyright law includes the concept of fair use, which allows limited use of copyrighted material without permission for purposes such as criticism, commentary, news reporting, educating and research.

→ Public works that no longer under copyright protection are said to be in public domain, meaning that they can be used by user as a public.

(c) Intellectual Property Rights :

Intellectual property rights, often abbreviated as IPR, is a board term that encompresses various legal rights granted to individuals and organizations for their intellectual creations and innovations. This is the superset of patent, copyrights, trademarks, trade secrets and more. The key aspects of IPR include the following.,

→ Encouragement of Innovation : IPR plays an important role in forecasting innovation, creativity, and economic growth by offering incentives for individuals and entities to invest in research and deployment, artistic expression and other forms of intellectual works.

→ Legal Protection : IPR laws provides legal means for individuals and organization to property, preventing unauthorize use, copying or exploitation by others.

→ Exclusive Rights : Depending on the type of ~~IPR~~ Intellectual property, IPR grants rights to creators or investors.

→ Economic value : IP can have significant economic value, as it can be bought, sold, licenced, and used as collateral for loans.

→ Balancing Public Interests : The laws aims to balance the interests of IPR holders with the public interests in accessing and benefiting from intellectual creators.

SET 32 - MCA

Name : HARVISH E

Roll No : 2232MCA0038

Course : Master of Computer Application

SUBJECT CODE

DMC 6204

SUBJECT NAME:

MOBILE APPLICATION DEVELOPMENT

Marks	
Name and Signature (faculty)	

13-10-2023 IIND SEMESTER ASSESSMENT - I. DMC 6204

MCA - II Mobile Application Developement 2232MCA0058

1. Explain in detail about the Mobile Information Device profile.

The Mobile Information Device profile (MIDP) is the technology specification that was developed by Java Community Process (JCP) and aimed to standardize platform for running Java application on mobile devices. It plays an crucial role in enabling the developement of mobile application using Java programming language.

Background and Purpose :

MIDP was developed in the early 2000's when the mobile devices were relatively less powerful compared to todays smartphones. It is designed to bring java applications to these devices.

Instead of one-purpose Java, sun provides different flavours of programming languages specification standards, Enterprise, Micro and Java Card. A specification published for the use of Java on embedded devices such as mobile apps, PDAs, Set Top Boxes address the limitation and pervating communication devices.

It combines a resource-combined JVM and set of Java APIs for developement of mobile application. J2ME introduces profiles to reduces the memory requirement. Profile specify language subsets for different group of devices.

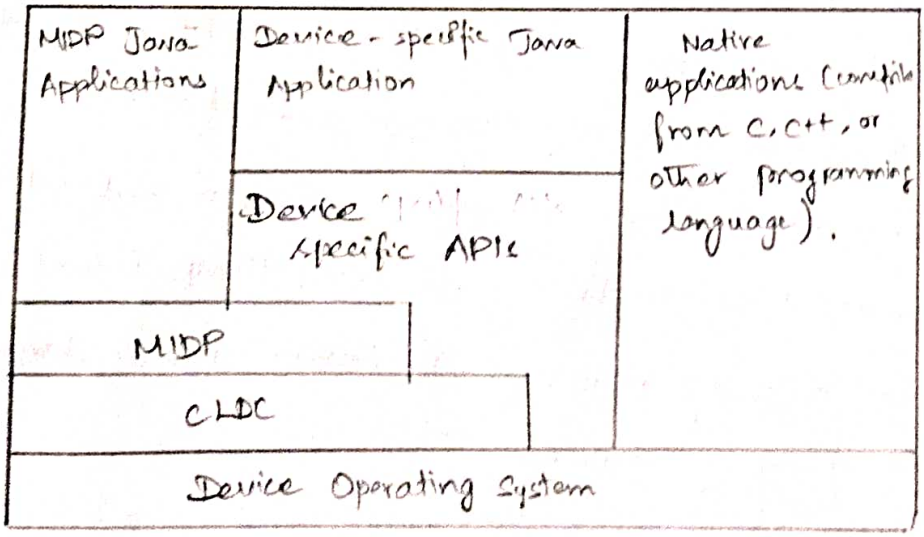
Key Components:

Beyond the core libraries only those Java classes which are actually needed to support typical feature of a devices group are includes into a corresponding profile devices.

J2ME has 2 configurations namely Connected Limited Device Configurations (CLDC) and Connected Devices Configuration (CDC).

MIDP Architecture:

* MIDP ~~architecture~~ is a part of Java ME framework and sits on the top of Connected Limited Device Configuration, a set of lower level programming interfaces.
* MIDP defines a lifecycle model similar to the applet model that is termed as MIDLet.
* MIDLet sits on the top of MIDP APIs and packages, specifically on top of CLDC APIs, CLDC APIs sits on the top of VM and VM sits on top of hardware.
* Native applications runs directly on this layered and represents the world as it is today - Many different kinds of devices, each with its own OS and native applications.



* Layered on the top of the Device OS is CDLC (Including KVM) and the MIDP APIs.
* MIDP applications use only the CIDC and MIDP APIs. ~~Device~~
* Device specific APIs Java Applications may also uses Java APIs supplied by the device vendor.
* The CLDC defines a core of APIs, mostly taken from J2SE. These includes fundamental classes in Java.lang, java stream classes,
  - -> fundamental classes in java.lang.
  - -> stream classes from java.io
  - -> simple collection from java.util
  - -> generalized network API in javax.microedition.io

* MIDP specified the AIPIs like java.microedition.midlet, javax.microedition.lcdui, javax.microedition.rms, javax.microedition.io

MIDP Package classes:

MIDP consists of 5 packages namely user interface packages, persistance package, application lifecycle package, networking package and core package.

1. User Interface package:
   - -> class : javax.microedition.lcdui
   - -> The UI APIs provide a set of features of implementation of user interfaces for MIDP application.

2. Persistence Package:
   - -> class : javax.microedition.rms.
   - -> The MIDP provides a mechanism for MIDlets to persistently store data and later retrive it.

3. Application lifecycle Package :

→ class : java.microedition.midlet

→ The MIDlet package provide MIDP application and the interaction between application and environment in which application runs.

4. Networking packages :

→ class : javax.microedition.io

→ MID profile includes networking supports based on the generic connection framework from the connected limited device configurations.

5. Core packages :

→ classes: javax.io , java.lang, java.util.

→ Provides for system IO through data streams.

MIDP lifecycle (MIDlet) :

The MIDlet lifecycle is the creation of any MIDlet. The life cycle includes the execution states StartApp(), PauseApp(), DestroyApp() and Launch. To manage developing it, there is software term application management software (AMS) providing the runtime environment. This runtime environment enforces the permissions, security and execution states and is provided the required scheduling and system classes.

① When a MIDlet is provided loaded into device and the constructor is called, it is in loaded state.

② After StartApp() is called, the MIDlet is in active state until the program manager calls PauseAPP() or DestroyAPP().

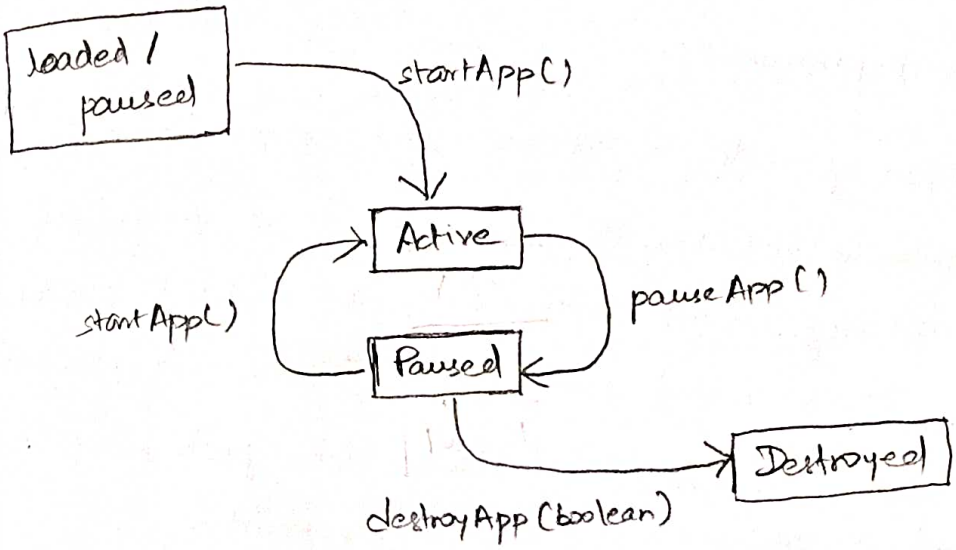
→ PauseApp() pauses the MIDlet

→ DestroyApp() destroys / terminates the MIDlet.

3. All state change call back method should terminate quickly, because the state is not changed competely before method returns.

4. MIDlets can request to resume activity by calling resumeRequest()

5. In order to terminate, a MIDlet can call notifyDestroyed(). Note that System.Exit() is not supported in MIDP.



MIDP Developement steps:

(i) Write your Java Application
(ii) Compile it
(iii) Pre verify
(iv) Package it into Jar file
(v) Create the application description
(vi) Publish the MIDlet suite
(vii) Install on device or emulator.

2. Explain the top reason why mobile apps will become so important for companies as corporate website.

Mobile apps have become increasingly important for corporate websites and businesses for the variety of reasons. Here are the top key reasons why mobile apps have gained prominence and will continue to be significant for corporate websites:

(i) Enhanced User Experience:

Mobile apps provide a superior user experience compared to mobile-responsive websites. They are designed specifically for mobile devices, offering smoother navigation, faster loading times, and intuitive touch based interactions. Apps can also leverage an advantage of device specific features like push notifications, GPS, camera and sensors, enhancing user engagement.

(ii) Personalization:

Mobile apps allow for highly personalized user experience. They can leverage user data, preferences and behaviour to tailor content and recommendations, making the interactions more relevant and engaging. Personalization can lead to increase user retention and conversation rates.

(iii) Offline Access:

Unlike corporate websites, mobile apps can offer limited functionality even when users are offline. This is particularly valuable for corporate application such as productivity tools, reference materials, when users need to

access to content or features regardless of their internet connection. However, to gain some specified benefits above on the application by the customer, he or she has to pay some amount as a part of subscribtion. This kind of business approach makes convinient for ~~bus~~ corporate, in terms to selling mobile application as a product.

(iv) Direct Communication:

Mobile apps ~~allow~~ enable direct and indirect instant communication with users through push notifications. Companies can use notifications to send updates, promotions, news and reminders, which can be highly effective in engaging and re-engaging users. For instance, SMBs can monitor how often customers use app, determine what promotions they might have interest in and review solicied and unsolicied feedback. This information can help build and maintain positive customer relationships.

(v) Brand loyalty and Engagement:

Businesses that operate in high competitive market vertices must put extra work on cultivating customer loyalty. Mobile applications can offer them an advantage, as they can interest their repeat business opportunities and improve referral rates. They can also boost new product and service adoptation rates.

Business applications could also play an important role in raising people's engagement rates in your business. Creating apps on deals, offers or creative ~~comp~~ campaigns to increase people's engagement with your app. There is no limitations in physical location to engage audience.

(vi) Performance and speed :

Mobile applications are typically faster and more responsive to websites. This speed is crucial, especially for e-commerce and traditional applications. This is where slow loading times can leads to cart abandonment or a poor user experience. Improved performance can direct by impact revenue and customer satisfaction.

In conclusion, mobile apps offers many advantages over traditional websites to the corporates.

14.10.2023

IInd SEMESTER ASSIGNMENT - 2.

I MCA

MOBILE APPLICATION DEVELOPMENT

2232MCA0058

DMC6204

1) What are the techniques for composing applications?

Composing mobile applications involves a combination of technical and design techniques to create functional, user-friendly and visually appealing apps. Here are some key techniques for composing mobile applications.

1. Platform section:

Choose the right platform for your mobile app. Decide whether you want to develop for android, ios or both using cross-platform tools like Flutter or react Native.

2. User - Centered Design:

Start with user-centered design approach. Understand your target audience with their needs. Design the app's user interface with a focus on usability, accessibility, and user experience (UX).

3. Wireframing and Prototyping:

Create wireframe and prototypes to visualize the app's layout and interaction. This helps in refining the design and functionality before developement begins.

4. Responsive design:

Ensure that your app's design is responsive, adapting to different screen size and orientation. This is crucial for a

consistant experience across various experience.

5. Intuitive Navigation:

Design an intuitive navigative structure. Users should be able to move through the app with the ease and find what they need without confusion.

6. Performance Optimization:

Pay attention to app performance by optimizing code, minimizing network requests, and using efficient data storage techniques. Slow apps can lead to user frustration and also indirectly it will affect the business.

7. Offline Functionality:

Mobile application developers need to incorporate offline functionality while composing, if possible. Users appreciate apps that will work even in limited or no internet connectivity.

8. Security:

Implement ~~re~~ robust security measures to protect user data. Use encryption, authentication and authorization mechanisms to ensure the app security. Lack or minimal security measures on an application can lead to lack of usage and so becomes loss to the business, while payment and 'checkout' related concepts comes into subject.

9. API Integeration:

Integerate with external APIs for accessing data and services properly handle API requests and responses, and consider rate limiting and cacheing to improve performance. Encouraging data transfer via APIs helps developers develop robost and easily accessible application.

10. Cross platform Developement:

If toargetting multiple platforms, use cross-platforms developement framework and tools to share a significant portion of code and resources across android and IOS.

11. Testing and Quality Assurance:

Rigenously test the app on various devices, screen sizes and operating system versions. Performance functional, usability and security testing to identify and fix issues.

12. Continuous Integeration and Deployment (CI/CD).

Set up CI/CD pipelines to automate the build, testing and deployment processes. This ensures faster and more reliable updates.

13. App Store Guidelines :

Familiarize yourself with the guidelines and requirements of the application provisioning softwares (Google Play store, Apple App store). Ensure your App compiles with the policies.

14. Feedback and Iteration :

Gather user feedback and analytics data to understand how users are engaging and interacting with your application developed and published to the world. Use this information to make iterative improvements.

15. App Monatization :

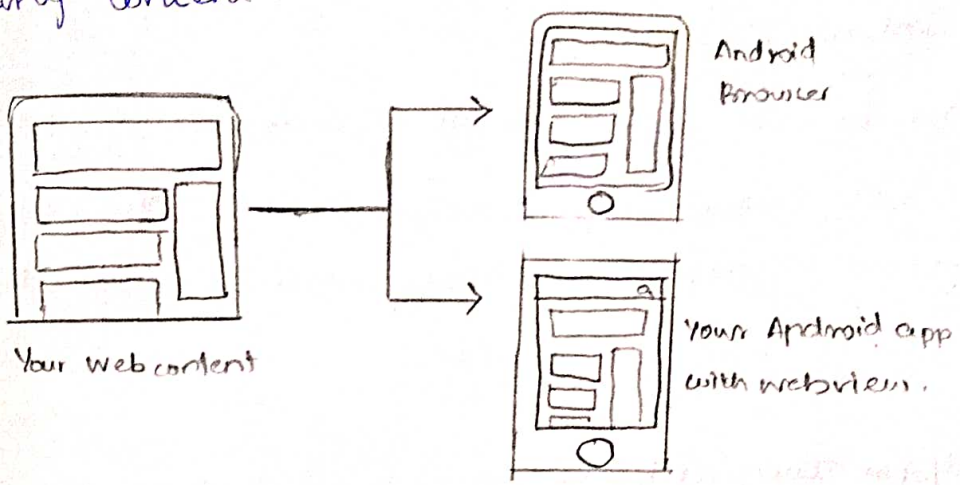
If your app is intended to generate revenue, consider various monatization strategies like in-app advertizing, in-app purchases or subscription models. On publishing application to the app provisioning software, They also provide monatization to the business based on the number of users ~~not~~ installed the application. However, this may not be suggest approach as it doesn't compensate your investments in publishing applications to the app market.

18. User Support and Updates :

Provides user support channels and regular release of new features, resolution of bugs and improved app performance. This promotes maintaining the mobile application at regular basis.

2. Describe the importance of communication via web?

Android offers a variety of ways to present a content to a user. To provide user experience that is consistant with the rest of platform, it is usually best to build a native apps that incorporates framework-provided experiences. Some apps, however may need increased control over the UI. In this case, a webview is a good option for displaying trusted first party content.



In Android, web communication is done by using special layout called webview which is the extention of the view class that helps displaying the web content.

If you want to deliver a small web application as a part of client application, webview layout is the best choise. It does not include any features of a fully developed web browser, such as navigation control or an address bar.

Use of web content developement on Android via "Webview" is a powerful technique that offers several benefits and importance

Here are some of the importance of incorporating web view through android application developement.

1. Hybrid Application Developement:

Andnoid app has capability to create hybrid application by combining native app components with web based content. This approach allows for greater flexibility and the ability to leverage existing web resources.

2. Efficient Content Delivery:

Web content, such as HTML, CSS and JS can be loaded dynamically via web view from remote servers. This enables efficient content delivery and easy updates without requiring users to install new app versions.

3. Cross platform Compatibility:

Web content can be developed once and displayed within the Android App via 'WebView'. This content can also be shared across different platforms, making it easier to maintain a consistent user experience.

4. Rich Media Interactivity:

Web content delivered through 'Webview' can include rich media elements like videos, interactive forms, maps, and animations. It provides an engaging user experience that may not be as easily achievable with native UI component alone.

5. Cost Effective Maintainance:

Content updates are easier to manage and cost effective. Developers can modify the ~~up~~ web content without updating an app by user. This is particularly useful for frequently changing information sources such as news, products lists or social media blogspots.

6. Uniform User Experience:

By rendering web content within the app, you can maintain a ~~same~~ consistant look and feel with your web site or other web-based services. This helps users recognize and navigate the app more easily.

7. Responsive Design:

Web content in Webview can take advantage of responsive design techniques to adapt to various screen sizes and orientations, ensuring a consistant user experience across different android devices.

8. Integeration ~~of~~ of webservices:

Many android apps require integeration with webbased services and APIs. 'Webview' allows developers to embed web forms or service endpoints directly in the app, streaming the user experience.

(16)

9. Support Offline Access:

Developers can design web content that works even when the device is offline. By cacheing data and resources, the app can continue to provide functionality, which is ~~espei~~ especially valuable for users within intermittent connectivity.

10. Third party Integerations:

'Webview' can be used to integerate third party web services, such as payment gateways, social media or authentication providers, into android app. This ~~sp~~ simplifies the process of integerating external services.

In Conclusion, the use and importance of Web communication via 'Webview' class on android developement has provided a valuable technique that combines the best of both native app functionality and web content integeration. However, it is important to implement web content securely and consider the specific use case and UX requirement of your application.

SET 32 - MCA

Name : HARYISH E

Roll No : 2232MCA0058

Course : Master of Computer Applications.

SUBJECT CODE

DMC 6001

SUBJECT NAME:

SOFTWARE PROJECT MANAGEMENT

Marks	
Name and Signature (faculty)	

16-10-2023 II nd SEMESTER ASSIGNMENT - 1. [redacted]2232MCA0058

II MCA [redacted] SOFTWARE PROJECT MANAGEMENT DMC6001

Illustrate Network Models in Software Project Management.

Network Models in software project management are graphical representations that helps in planning, scheduling and managing tasks and activities created and framed by project manager on an assigned software project. These models provides the way to understand visually on the dependancies on a tasks, estimate project duration, allocate resources, and track progress.

There are several network models commonly used in software project management, with two of the most prominent being the PERT (Project Evaluation and Review Technique) and CPM (Critical Path Method).

Critical Path Method:

The Critical Path Network diagram is deterministic, meaning the work duration it lays out are meant to be instructions followed by project team members. This is the reason why the CPM model is best choise for scheduling.

- Purpose: CPM is used to determine the short time it will take to complete a project by identifying the critical path - the sequence of tasks that, if delayed, would delay the entire project.

- Components :
  - Task / Activities : These represents the individual work items in the project, each with its estimated duration.
  - Dependencies : Relationship between the tasks including finish-to-start, start-to-start and finish-to-finish.
  - Network diagram : The ~~graphical~~ visual representation of a ~~graphic~~ of graph diagram where each node is represented as below :,

Early start	duration	Early finish
Activity label		
latest start	Float	Latest finish.

  - steps involved :
    1. Identifying the activities
    2. Defining the tasks and dependancies
    3. Creating Network diagram.
    4. Calculate earliest start and finish times and latest start and finish times of each task.
    5. Determining the critical path, which is the longest path in the network with zero slack or float.

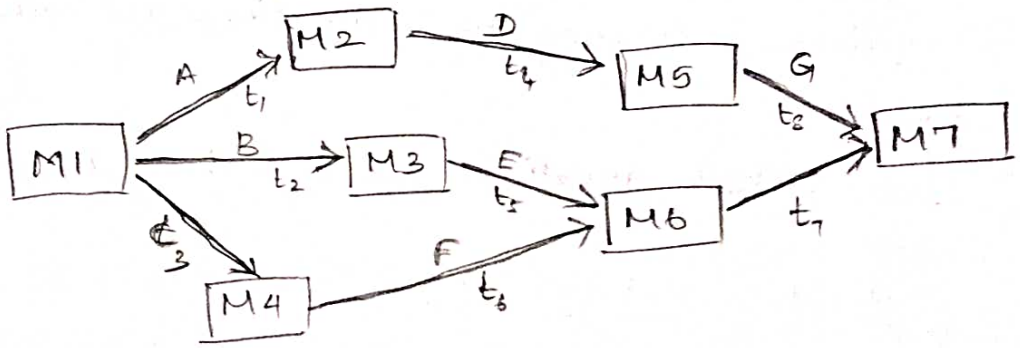
  - Advantages :
    (i) Helps identify the most critical tasks
    (ii) Provides a clear schedule.
    (iii) Supports resource allocation.

  - Limitations :
    (i) Assumes that task duration are deterministic
    (ii) May not account for resource constraints.

Program Evaluation and Review Technique (PERT)

PERT is used to identify the time it takes to finish a particular task or activity. It is a system that helps in proper scheduling and coordination of all tasks throughout the project. It is also helps in keeping ~~tattack~~ track of the progress, or lack thereof, of the overall project.

- ~~Purpose~~ :
- Purpose : PERT is the probabilistic model used to estimate the projects duration when task duration are uncertain.
- Components :
  - * Tasks/ Activities: Similar to CRM, but each task has three duration estimates : optimistic, most likely or greedy, pessimistic.
  - * Dependencies: Task Relationships.
  - * Network diagram:



M_i → Milestones

X → Activity short name/phase

t_i → Time /duration of activity /duration estimates

* Steps to calculate the weightage of the activity.
  1. Gathering tasks and their duration estimates (O/M/P)
  2. Plug into the formula based on estimates.
  3. - substituting duration estimates into formula below,

$$\boxed{\text{Expected Time } (T_E) = \frac{(O+4M+P)}{6}}$$

* Steps of constructing PERM: This is similar to the CPM steps, but instead of calculating start, finish and latest times (3), you can utilize the steps to calculate the weightage of the activity. Laterly, determine the duration of project (estimated) and their standard deviation.

* Advantages:
  1. Accounts for uncertainity in task durations.
  2. Provides a range of possible project duration.
  3.

* Limitations:
  1. More complex than CPM
  2. Requires estimating three durations for each task.
  3. Assumes task durations follow a beta distruction.

Dependancy Structure Matrix (DSM):

The Dependancy Structure Matrix or design structure matrix is a simple, compact and visual representation of a system developement or project in the form of a square matrix. DSM is the great way to visualize and manage dependancies between modules, which is critical to the project management.

- Purpose : DSM is the different approach to modeling project dependancies. It represents tasks in a matrix, makes suitable for project with complex dependencies.

- Components :
  * Tasks / Activities : To frame the architectural diagram in the form of graph that can laterly transform to a matrix of rows and columns.
  * Dependancies : The graphical representation of in a directed fashion and the off-diagonal elements that indicates task relationship.
  * Representation : Matrix format / tabular representation with rows and columns.
    - → Row header indicates the elements / nodes in a graph.
    - → column header represents the same.
    - → Non-Empty cells represents the arrow in a graph that is marked with the node element points to.

- Steps ~~* Steps~~ :
  1. Defining tasks, activities and dependancies.
  2. Creating DSM matrix, indicating dependancies
  3. Analyzing the matrix to identify dependancies and tasks relationship.

* Advantages :
  (i) Useful for managing complex projects with many dependancies.
  (ii) Supports easy visualization of dependancies.

* Limitations :
  (i) May not be as intuitive for smallest projects.
  (ii) Doesn't provide critical path analysis.

Gantt Graph :

Gantt graph is commonly used in project management and it is the useful way of presenting activities (tasks or events) displayed against time. On the left side of the chart is a list of the activities and along the top is a suitable time scale. This was designed by Karol Adamiecki in 1890s.

* Purpose : This chart is a visual representation of a project schedule, showing tasks, durations and task dependancies.

* Components :
  → Tasks / Activities : Represented as a horizontal bars on the timeline.
  → Dependancies : shown as links between tasks on the diagram.
  → Diagram / Representation : Represented in a graphical table where first half of table includes Task id and Task name, starting date/week, ending date/week. The second half of table has bars, the position and the length of the bar reflects the duration of the task with respect to start and end date/week.

- Steps :

① Define the task and activities in a micro milestones.

② List out the task, duration and dependancies.

③ Create a Gantt graph chart, arranging the task in the order they need to be completed.

- Advantages :

(i) Provide a clear visual representation of project schedule.

(ii) Easy to understand for both project management members, managers, and stackholders.

- Limitations :

(i) May not handle complex projects with many dependancies effectively.

(ii) Limited in its ability to calculate critical paths.

In software project management, these networks models are essential tool for planning, scheduling and managing software developement projects. Dependancies depending on the project's complexity and the level of uncertainity in task durations, project managers can choose the most suitable network model to ensure successful project delivery.

16-10-2023 IInd SEMESTER ASSIGNMENT -2 22232MCA0058

2 MCA SOFTWARE PROJECT MANAGEMENT DMC 6001

Write a brief note on the Software Evaluation Technique.

The evaluation of a software project will be based on strategic, technical and economic criteria and will normally be undertaken as a strategic planning or feasibility study of an information system developement. The risk involved in the project implementation and execution is also needed to be evaluated.

Based on these, the Project evaluation is carried out in step o of step wise. The sequence step to step are concerned with managing the project that stems from the project selection.

The below listed are the different project management evaluation techniques done on a software based on project ~~require~~ requirement.

1. Strategic Assessement
2. Technical Assessement
3. Cost - benefit analysis
4. Cash flow forecasting
5. Cost - benefit evaluation technique.
6. Risk Evaluation.

1. Strategic Assessment :

a. Programme management :

* Individual projects are needed to be seen as a a components of a programme and should be evaluated and managed as such.
* A project must be evaluated according to how it contributes to its goal and its visibility, timing, resourcing and final worth can be affected by the programme as a whole.
* Even where there is no explicitly define programme, any proposed project must be evaluated within the context of organization's overall business objective.

b. Portfolio Management :

* where an organization such as software house is developing a software system they could be asked to carry out a strategic and operational assessment on behalf of the customer.
* The proposed project will form part of a portfolio of the ongoing and planned project and project selection must take an account of the possible efforts of other projects in the portfolio and the overall portfolio profile.

2. Technical Assessement :

* Technical assessement of a proposed system consists ~~of~~ of evaluating the required functionality against the hardware and software available.
* Constraints will ~~in~~ influence the cost of the solution and this must be taken into account in the cost-benefit analysis.

3. Cost - Benefit Analysis :

* The most common way of carrying out an economic ~~assessm~~ assesment of a ~~propsd~~ proposed information system, or other developement, it is by comparing the expected cost of developement and operation of the system with benefits of having it in place.
* Assessement is based ~~on~~ upon the questions of whether the costs are necessary to ask whether or not the project ~~p~~ under consideration is the best of a number of opinions.
* Standard way of evaluating cost - benefit analysis of any project is carried ~~on~~ out on the following two steps.,
  (i) Identifying and estimating all of costs and benefits of carrying out the project
  (ii) Expressing these costs and benefits in common units.
* Costs are categories from the monetory terms, into three such as.,
  (i) Developement costs
  (ii) Setup costs
  (iii) Operational costs.

4. Cash Flow Forecasting :

* Cash flow forecasting is a critical financial management practice in software project management.
* It involves estimating and monitoring the inflow and outflow of cash during the cost of a software project.
* Accurate cash flow forecasting is essential to ensure that a project remainds finantially viable and can meet its obligations.
* It has the purpose on the following aspects,
  1. Financial planning : Provides a clear picture of the project's financial health and requirement throughout its lifetime.
  2. Risk management : Identifying potential cash shortages or surplus and take proactive measures to mitigate financial risks.
  3. Resource allocation : Determine the allocation of financial resources, including budgets for developement, personnel, infrastructure.
  4. Decision making : Supports decision-making process by assessing the financial feasibility of a project phases, milestones, and options.

5. Cost - Benefit Evaluation Techniques :

* Cost - Benefit evaluation techniques are essential tools used in software project management to asess the financial and non-financial aspects of a project
* These techniques helps project managers and stackholders informed decision about project initiation, prioritization and resource allocation.

Here are the following techniques of cost - benefit evaluation techniques used in software project management;

a. Net Present Value

NPV calculates the present value of all future cash flows (both costs and benefits) generated by a project. It helps determine whether a project. It helps determine whether a project is financially viable by considering the time value of money.

b. Return of Investment :

ROI measures the profitability of a project by comparing the net benefits to initiate investments. There are variations on the formula used to calculate ROI but the common formula to determine them is;

$$\boxed{ROI = \frac{\text{average annual profit}}{\text{total investment}} \times 100}$$

c. Payback Period :

Payback period represents the time it takes for a project to recoup its initial investment from the generated cost inflows. Normally, the project with the shortest payback period will be chosen as a basis that an organization will wish to minimize the time that a project is 'in debt'.

d. Benefit Cost Ratio (BCR).

BCR assesses the ratio of the total benefits to total costs, helping stackholders evaluate the economic attractiveness

of a project.

e. Cost Effective Analysis:

Cost Effective Analysis compares the costs of alternative project and solutions to determine which provides the best (solution) outcomes for a given costs.

f. Break-even-Analysis:

Break-even analysis identifies the point at which total costs equal total revenues or benefits, indicating the minimum level of outcome required for a project to be financially sustainable.

g. Internal Rate of Return (IRR):

IRR overcomes the disadvantage of NPV that, NPV might not be directly comparable in earning from other investments or the costs of borrowing capital.

6. Risk Evaluation:

* Risk Evaluation is a crucial aspect of software project management that involves assessing potential risks and uncertainities that may impact a project's success.
* By systematically evaluating risks, project Managers can develop effective risk mitigation strategies and make informed.
* Risk Evaluation involves activities such as Risk Identification, aspects, Quantitative analysis, Heat maps, Registers, SWOT, workshops, dependancy analysis, Root cause analysis.

Purpose of Software Project Evaluation:

~~Assessing Success~~:

(i) Evaluation of whether the project acheived its intended goals, met requirements and delivered value to stakeholders.

(ii) Examine the quality of the software product, including its functionality, reliability, performance, security and usability.

(iii) Identify strengths and weaknesses of a project management and developement process to facilitate continuous improvement.

(iv) Analysis of risks and issues encountered during the project and assesses how effectively ~~co~~ they were mitigated.

(v) Capture lessons learned and best practice to share with the organization and apply to further projects.

Conclusion:

Based on the purpose and various techniques of project evaluation (also costing), we could conclude that the project must be evaluated on strategic, technical and economic grounds..