WEB TECHNOLOGIES

Module 1

Introduction to the Internet: The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, Multipurpose Internet Mail Extensions, The Hypertext Transfer Protocol. Common Gateway Interface(CGI), Content Management System – Basics *Case Study:* Apache Server, Word Press.

1.1 A Brief Introduction to the Internet:

1.1 Origins

ARPAnet - late 1960s and early 1970s US DoD developed large scale network. Basic requirement was that the network should be sufficiently robust. DoD's ARPA funded for the first project and so the name ARPAnet . ARPANET was available to laboratories & universities that conducted ARPA-funded research.

BITnet(Because It's Time N/W), CSnet(Computer Science Network) :developed in late 1970s & early 1980s.BITnet is began at City University of New York. It was to provide email and file transfer for other institutions. For different reasons BITnet and CSnet were not widely used.

NSFnet(National Science Foundation) – was created in 1986 .Originally for non-DOD funded places. Initially connected NSF funded supercomputer centers at five universities. By 1990, it had replaced ARPAnet for non-military uses. Soon became the network for all (by the early 1990s).By 1992 NSFnet connected more than 1 million computers , around the world **.** NSFnet eventually became known as the Internet by 1995.

1.2What is the Internet ?

A huge collection of computers connected in a communications network. Internet is N/w of N/W's and different devices in the network communicates with each other based on the low level protocol TCP/IP (Transport layer).TCP/IP standard allows a program on one computer to communicate with a program on another computer via internet.Not every computer on internet directly connects to every other computer.Individual computers in an organization can be connected each other in a local network.

1.3 Internet Protocol (IP) Addresses

Every node has a unique numeric address. The Internet Protocol (IP) address of a machine connected to the Internet is a unique 32 bit number.

4 8 bit numbers separated by periods x.x.x.x ranges 0 to 255.

New standard, IPv6, has 128 bits (1998). Organizations are assigned groups of IPs for their computers. For eg.a small organization may be assigned 256 IP addresses such as 191.57.126.0 to 191.57.126.255.Very large organizations such as DoD may be assigned 16 million IP addresses which include IP address with one particular first 8 bit number such as 12.0.0.0 to 12.255.255.255

1.4 Domain names

Domain names are used to identify the machine in a network.It is difficult to remember number to identify machines on net.Domain begin with host name followed by one or more domain names

Form: **host-name.domain-names**

Ex) www.cars.maruthi.org

First domain is the smallest; last is the largest. Last domain specifies the type of organization. Fully qualified domain name means the host name and all of the domain names.

Domain names must be converted to IP address before the message can be transmitted over the internet to the destination. It is done using **Domain Name Servers**. DNS servers - convert fully qualified domain names to IP's. If someone types www.example.com , it will map that name to the corresponding IP address similar to 121.12.12.121. Domain names made up of multiple parts called labels. Top level domain is what appears after the period in the domain name. A few examples of toplevel domains are .com,.org,.edu.Some denote a country code or geographic location such as .us or .ca. Each label denotes another subdomain to the right.

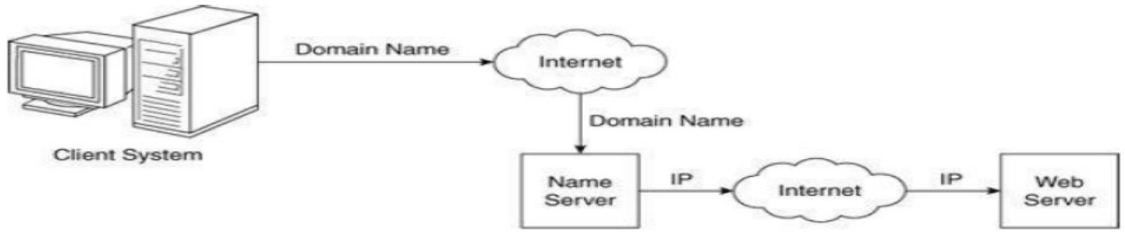


Figure 1.1 Domain name conversion

1.2The World-Wide Web

Origins

Tim Berners-Lee at CERN proposed the Web in 1989. Purpose is to allow scientists around the world to use the internet to have access to many databases of scientific work through their own computers.

Document form: hypertext(text with embeded links to text in other documents. The units of information on the web have different names: pages, documents and resources. Hypermedia means a document contains more than just text(images, sound, etc.)

Web or Internet?

The internet is a collection of computers and other devices connected by equipment that allows them to communicate with eachother.The web is a collection of softwares and protocols that has been installed on most of the computers on the internet .Some of these computers run Web servers, which provide documents, but most run Web clients, or browsers, which request documents from servers and display them to users.

Difference between web and internet

Web

Collection of software and protocols that helps to gather or deliver information during the communication. Web is information repository over internet

Internet

Collection of computers and other devices connected each other and allows them to communicate each other. Internet is the interconnected network of networks

1.3 Web Browsers

It is a software running on client machine.It is called browsers because ,it allows the users to browse the resourses available on server.

- Mosaic :First browser with GUI developed by NCSA (National Center for Supercomputer Applications at Univ.of Illinois), in early 1993 .

Browsers are clients because it always initiate communication with servers and servers react. Most requests are for existing documents. But some requests are for program execution, with the output being returned as a document.Commonly used protocol in the Web is HyperText Transfer Protocol (HTTP). Examples of commonly used browsers are Microsoft IE, Mozilla Firefox, Google Chrome, Opera, Apples Safari.

Static document

A browser requests a static document from a server. The server locates the document among its servable documents and sends it to the browser, which displays it for the user.

Dynamic Document

User supplies the requested input, server perform some computation and then returns the results of the computation. Sometimes a browser directly requests the execution of a program stored on the server. The output of the program is then returned to the browser.

Static document	**Dynamic Document**
• A browser requests a static document from a server. • The server locates the document among its servable documents and sends it to the browser, which displays it for the user • Information change rarely • Database is not used • Not interactive • Contents doesn't change with user.	• user supplies the requested input, server perform some computation and then returns the results of the computation. • Sometimes a browser directly requests the execution of a program stored on the server. The output of the program is then returned to the browser • Information change are rarely • Database is used • Interactive • It generate content based on user • Eg: facebook

1.4 Web Servers

Provide responses to browser requests, either existing documents or dynamically built documents. They act only when requests are made to them by browsers running on other computers on the network. All communications between browsers and servers use Hypertext Transfer Protocol (HTTP). Most commonly used web servers are Apache and IIS.

Apache -65% ,IIS-16% ,nginX-8%

A Web client, or browser, opens a network connection to a Web server, sends information requests and possibly data to the server, receives information from the server, and closes the connection. Other machines exist between browsers and servers on the network are network routers and domain-name servers.

Operation of web servers

Web browsers initiate a network communication with servers by sending a request using URL. When web server begins execution it informs the OS under which it is running , that it is ready to accept incoming n/w

connection through a specific port .While in this running state the server runs as a background process in the OS. Web servers performs operations specified by the commands. All HTTP commands include a URL, which includes the specification of a host server machine.

A URL can specify one of two different things:

The address of a data file stored on the server that is to be sent to the client, or

A program stored on the server that the client wants executed, with the output of the program returned to the client.

The primary task of a Web server is to monitor a communications port on its host machine, accept HTTP commands through that port, and perform the operations specified by the commands and returns the response.

General server characteristics

The file structure of a Web server has two main directories:

1. ***Document root (servable documents*)**: It stores the web document to which the server has direct access and normally serves to clients.

2. ***Server root (server system software):*** stores server and its support software.

The files stored directly in the document root are those available to clients through top-level URLs.. The clients do not access the document root directly in URLs; rather, the server maps requested URLs to the document root, whose location is not known to clients.

For example, the site name is www.tunias.com.

Suppose that the document root is named *topdocs* and is stored in the */admin/web* directory, making its address */admin/web/topdocs*.

A request for a file from a client with the URL http://www.tunias.com/petunias.html will cause the server to search for the file with the file path /admin/web/topdocs/petunias.html.

Many servers allow part of the servable document collection to be stored outside the directory at the document root. The secondary areas from which documents can be served are called ***virtual document trees*** .Sometimes files with different types of content, such as images, are stored outside the document root .Many servers can support more than one site on a computer, thus reducing the cost of each site and making their maintenance more convenient. Such secondary hosts are called ***virtual hosts.*** Some servers can serve documents that are in the document root of other machines on the Web; they are called ***proxy servers***.

Apache

Apache server is open source, fast, reliable, best available UNIX based systems).Apace is controlled and maintained by ***configuration file.*** When Apache begins execution, it reads its configuration information from a file and sets its parameters to operate accordingly. The configuration file can be edited by the managers to change Apache's behaviour.

3 configuration files in Apache Server(httpd.conf,srm.conf,access.conf). httpd.conf- stores the directives that control an Apache server behaviour.

IIS

Most Windows based Webservers use IIS.In IIS, the Server behaviours is modified by changes made through a window based management program called ***IIS Snap-in.***

Under Windows XP and Vista, the IIS snap-in is accessed by going to *Control Panel, Administrative Tools, and IIS Admin*. Clicking on this last selection takes you to a window that allows starting, stopping, or pausing IIS.

1.5 Uniform Resource Locators- URL

URL is used to identify documents (resources) on the internet.

General form: **scheme:object-address**

scheme → communications protocol, such as telnet , http, mailto ,ftp

The http protocol is used to request and send HTML documents. For the http protocol, the object-address is*:*

// fully qualified domain name/doc path

For the file protocol, only the doc path is needed

file://path-to-document

Host name(name of the server) may include a port number(default port is 80): *http:8080//www.xyz.org*

- URLs cannot include spaces or any of a collection of other special characters (semicolons, colons, ...)

-To include space or special character, the character must be coded as % sign , followed by two digit hexadecimal ASCII code of char.

Ex) if domain name is San jose→San%20Jose(20 hexacode for space)

-UNIX Server →path is specified with forward slashes

-Windows Server→ backward slashes.

A URL need not include all directories on the path. A path that includes all directories along the way is called a complete path.

Complete path: http://www.abc.com/files/images/logo.jpg

In most cases, the path to the document is relative to some base path that is specified in the configuration files of the server. Such paths are called *partial paths* .

Eg ,if the server's configuration specifies that the root directory for files it can serve is files/images, the previous URL is specified as

http://www.abc.com/logo.jpg

If the doc path ends with a slash, it means it is a directory.nn

http://www.abc.com/departments/

Sometimes a directory is specified (with the trailing slash) but its name is not given, as in the following example:
http://www.abc.com/

The server then searches at the top level of the directory in which servable documents are normally stored for something it recognizes as a home page. By convention, this page is often a file named *index.html*. The home page usually includes links that allow the user to find the other related servable files on the server.If the directory does not have a file that the server recognizes as being a home page, a directory listing is constructed and returned to the browser.

1.6 Multipurpose Internet Mail Extensions (MIME)

Browser needs some way of determining the format of the document it receives from a web server. Without knowing the form of the document ,browsers would not be able to render it .Because different document format require different rendering software. The form of the document are specified with MIME.

MIME is an internet standard that specifies the data format of the content that the server is transmitting to the browser,so that programs can interpret the data correctly.

MIME is used to specify to the browser the format of a file returned by the server. Web server attaches MIME format specification to the beginning of the document ,to provide to a browser.

When browser receives the document , it uses MIME format specification to determine what to do with the document.

If content is text→MIME code tells the browser that it is text

If content is sound→browser will choose a program to access the transmitted sound.

Type specifications

Form: type/subtype

Common MIME types are : **1) text 2) images 3) video**

SubTypes are: **text → plain ,html**

images→gif, jpeg

video→mpeg, quicktime

Examples: text/plain, text/html, image/gif, image/jpeg

Server gets type from the requested file name's extension (.html tells the server that it should attach text/html to the document before sending it to the browser text/html).Browser gets the type explicitly from the server.

Experimental Document Types

When the MIME type is either text or image, the browser renders the document without any problem. However if the type is video or audio, it cannot renders the document .It has to take the help of other software like media player,win amp etc. These softwares are called helper applications or plugins.

The name of an experimental Subtype begins with **x-** e.g., video/x-msvideo.

If browser does not have helper application/plug-in to render a document, an error message is displayed.

Every browser has a set of MIME specification (file types) it can handle.

All browsers can deal with

Eg) text/plain (unformatted text) ,and text/html (HTML files).

1.7 HyperText Transfer Protocol

HTTP is the application layer protocol used by all Web communications. HTTP consists of two phases: (current version 1.1)

The Request and The Response phase.

Each HTTP communication (request or response) between a browser and a Web server consists of two parts: a header and a body.

The header contains information about the communication; The body contains the data of the communication if there is any. An HTTP session is a sequence of network request-response transactions.

An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a server. An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own.

Request Phase

Form:

a)HTTP method domain part of URL HTTP version.

b)Header fields

c)blank line

d) Message body

a) An example of the first line of a request:

GET /ktu.edu/degrees.html HTTP/1.1

Most commonly used HTTP request methods or simply HTTP methods are :

(i) **GET** - Fetch a data from web server by specifying parameters in the URL portion of the request.This is the main method used for document retrieval.

(ii)**POST** – To send data to the server for eg: file update ,form data etc.Execute the document, using the enclosed data in body

(iii) **HEAD** - Fetch just the header of the document only (not body of the document)

(iv) **PUT** – replace the specific doc/Store a new document on the server at a location specified by the given URL

(v) **DELETE** - Remove a document from the server

b) Header fields-

It includes information about the communication.

format of header field

field-name followed by colon and value

4 categories of header fields

General→for general information like date

Request→included in request headers

Response→for response headers

Entity→used in both request and response

One common **request** field is **Accept** field which specifies a preference of the browser for the MIME type of the requested document

Ex) Accept:text/plain

Accept:image/gif

Accept:text/html

Header provides additional information about the data that will be sent

Ex) content-type:text/html

Ex) *Host:hostname* request field gives the name of the host.

The header of a request must be followed by a blank line, which is used to separate the header from the body of the request.

Example of HTTP Client request

GET /index.html HTTP/1.1

Host: www.example.com

Host indicate that the internet host of the resource (server) being requested.

A client request (in this request line and only one header field) is followed by a blank line, used to separate the header from the body.

Response Phase

The general form of HTTP response is as follows:

a)Status line

b)Response header fields

c)Blank line

d)Response body

a) Status line format:

HTTP version status code explanation

Example: HTTP/1.1 200 OK

- (Current version is 1.1)

- Status code is a three-digit number; first digit specifies the general status

1 => Informational

2 => Success

3 => Redirection

4 => Client error

5 => Server error

404→Not found(requested file could not found)

500→Internal Server error(server has encountered a problem and not able to fulfil the request.

200→OK

b) **After the status line, server sends a response header , contain several lines of information about the response each in the form of a field. The only essential field of the header is**

Content-type:text/html

c) Response header followed by blank line

d) Response data follows the blank line

The following is the response document for the previous request

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

Response fields in header section

ETag (entity tag) header field is used to determine if a cached version of the requested resource is identical to the current version of the resource on the server.

Content-Type specifies the Internet media type of the data conveyed by the HTTP message

Content-Length indicates its length in bytes.

Accept-Ranges*: bytes:* HTTP/1.1 webserver publishes its ability to respond to requests for certain byte ranges of the document by setting the field *Accept-Ranges: bytes*.

Connection*: close* is sent, it means that the web server will close the TCP connection immediately after the transfer of this response.

Last –Modified- indicates the date and time at which the document was last modified.

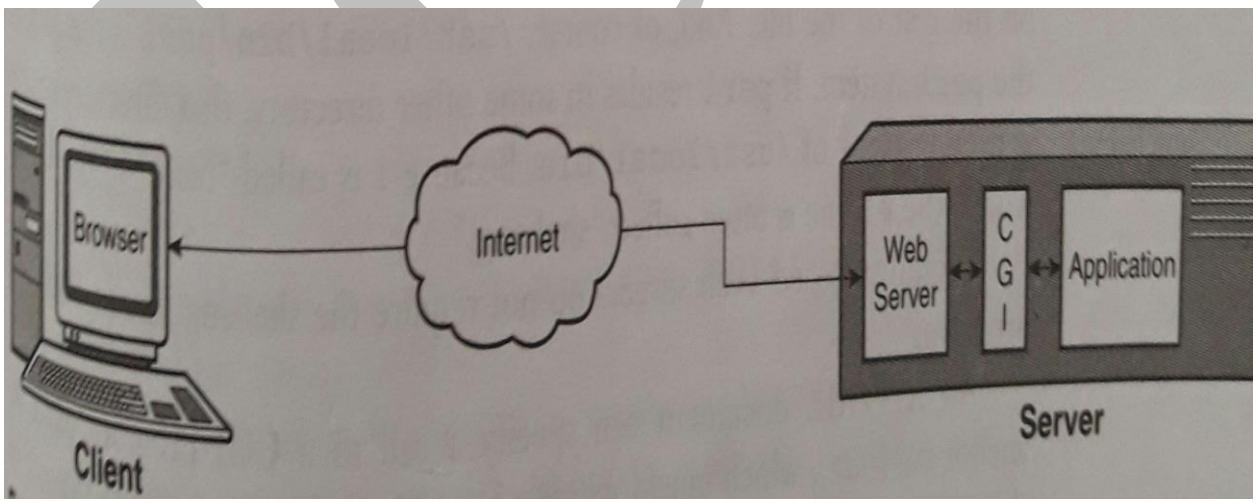
Most of the header lines are optional.

A *Content-Encoding* can be used to compress the transmitted data.

Difference between GET & POST

GET	POST
• Gets or retrieves information from the server • Send information to the server as part of a URL • Visible to human eye • Limited number of characters	• Post or sends data to a server in a secure manner • Sends form data as part of HTTP message • Secure- hides the submitted data • No limitations about the amount of data.

1.8 CGI (COMMON GATEWAY INTERFACE)



CGI act as an Interface between Webserver and Applications. CGI enables the Web server to handle client requests (e.g. from Web Browsers) and passing it to the right application. CGI programs may be written in *any* programming language like c,c++, java etc. A very widespread language that is used for CGI programming is Perl.

The Common Gateway Interface (CGI) is a protocol describing a standard way of providing server-side active web content. Under circumstances determined by the server, an HTTP request will cause a program to run. The output from the program will be the response returned to the client making the request. Servers can identify CGI programs by their addresses on the server or by file extensions.When a server receives a request for a CGI Program, it does not return the file- it executes the program in the file and return that programs output. It may be an html document.

One common way for a browser user to interact with web server is through forms. A form is presented to the user and the filled contents will be sent to the server by clicking submit button. The server decodes the transmitted form contents, perform necessary computations and produce the output.

CGI Linkage

CGI programs often are stored in a directory named cgi-bin. The first line of your program should look like this:

#!/usr/bin/perl –W

The final part contains optional flags for the Perl interpreter. Warnings are enabled by the -w flag.

print "Content-type: text/html\n\n";

This is a *content-type header* that tells the receiving web browser what sort of data it is about to receive . In this case, an HTML document.

```
//first.cgi
 #!/usr/bin/perl -w
 print "Content-type: text/html\n\n";
 print "Hello, world!\n";
```

Example 2

Program : second.cgi

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
print "<html><head><title>Hello  world</title></head>\n";
print "<body>\n";
print "<h2>Hello, world!</h2>\n";
print "</body></html>\n";
```

The CGI.pm Module

It is a standard library module. The CGI program should include the module via the *use* command. This goes after the #!/usr/bin/perl line and before any other code:

```
 use CGI qw(:standard);
```

The qw(:standard) part of this line indicates that we're importing the "standard" set of functions from CGI.pm

The CGI.pm module has many functions;

header;

start_html;

end_html;

The header function prints out the "Content-type" header. With no arguments, the type is assumed to be "text/html".

start_html prints out the

< !DOCTYPE html > <html>, <head>, <title>and <body> tags.

If you call start_html with only a single string argument, it's assumed to be the page title.

Ex) print start_html("Hello World");

The end_html function prints out the closing HTML tags:

</body>

</html>

Example

```
#!/usr/bin/perl -w
use CGI qw(:standard);
print header;
print start_html("Hello World");
print "<h2>Hello, world!</h2>\n";
print end_html;
```

Common CGI.pm Functions

"Shortcut" functions produce tags, using their parameters as attribute values

e.g., h2("Very easy!"); produces

<h2> Very easy! </h2>

Tags can have both content and attributes. Each attribute is passed as a name/value pair ,just as in a hash literal. Attribute names are passed with a preceding dash

```
textarea( -name => "Description",
          -rows => "2",
```

```
            -cols => "35"   );
```

Produces:

```
 <textarea name ="Description" rows=2
          cols=35> </textarea>
a({-href => "fruit.html"},
      "Press here for fruit descriptions");
```

Output:
Press here for fruit descriptions

Using CGI.pm to Parse the Query String

Example1.html

```
<html><head><title>Test Form</title></head>
<body>
<form action="test.cgi" method="GET">
<p>Enter Name: <input type="text" name="fname" size=30>
<input type="submit"></p>
</form>
</body></html>
```

Test.cgi

```
#!/usr/bin/perl -w
use CGI qw(:standard);
print header;
print start_html("Test");
print "Your name is:  param($fname) "<br>\n";
print end_html;
```

Example2.html

```
<html><head><title>Test Form</title></head>
<body>
<form action="get.cgi" method="GET">
<p>First Name: <input type="text" name="firstname" size=30><br />
```

```
Last Name: <input type="text" name="lastname" size=30><br />
<input type="submit"></p>
</form>
</body></html>
```

get.cgi

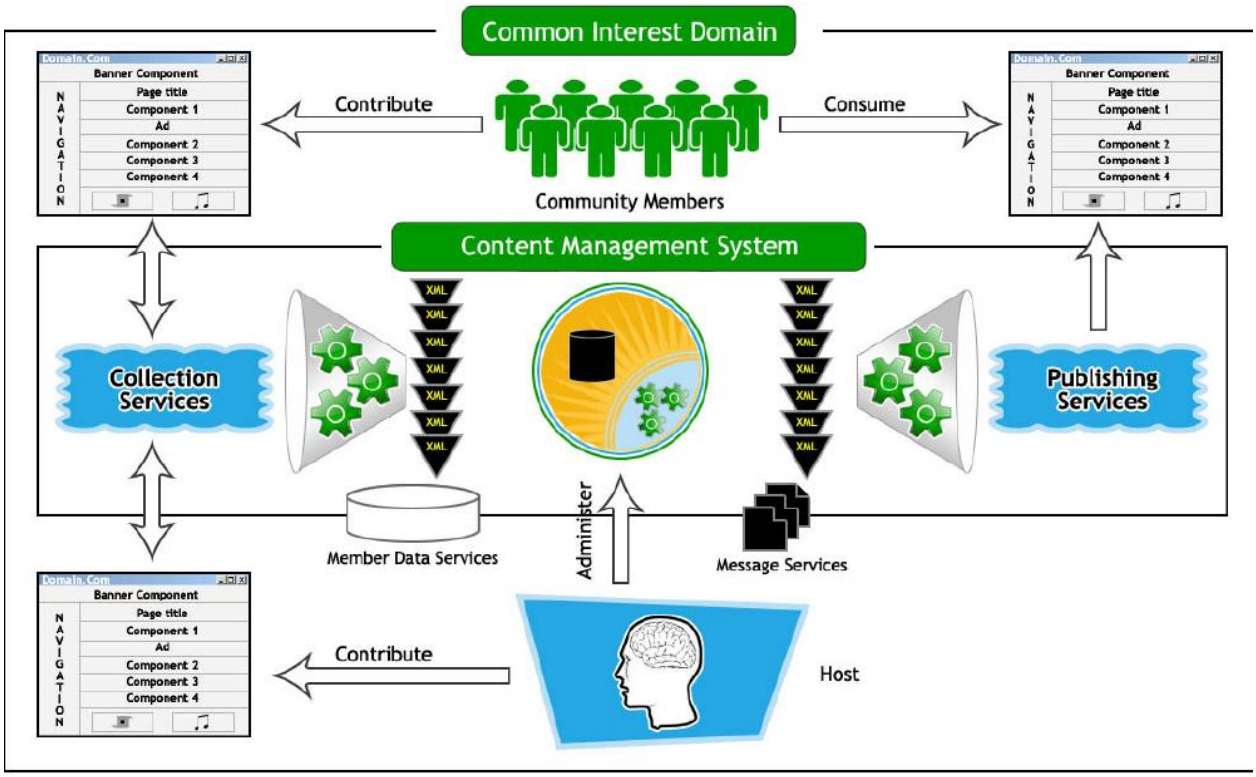
```
#!/usr/bin/perl -w
use CGI qw(:standard);
print header;
print start_html("Get Form");
my $fname,$lname;
$fname=param(“firstname”);
$lname=param(“lastname”);
print "Your Name = $fname\t$lname";
 print end_html;
```

1.9 CONTENT MANAGEMENT SYSTEMS

Content Management Systems was developed as a mechanism to maintain the content of the website.The term "Content Management System" is synonymous with "Web Content Management Systems".**Content** is any type or unit of digital information that is likely to be published across the Internet and/or Intranet .It can be text, images, graphics, video, sound etc. Content Management contributes to the effective management of various kinds of with the support of centralized webmasters and decentralized web authors/editors . They can create, edit, manage and publish all the content of a web page in accordance with a given framework or requirements.

A content management system (CMS) is a system used to manage content, typically for a website. A CMS consists of two elements: the content management application (CMA) and the content delivery application (CDA). The CMA element allows the content manager or author to manage the creation, modification and removal of content from a website without needing the expertise of a webmaster. The CDA element uses the information, compiling it to update the website.

A CMS enables a variety of centralized technical and de-centralized non-technical staff to create, edit, manage and finally publish a variety of content (text, graphics, video etc), under the constraint of a centralized set of rules, processes and workflows that ensure a coherent, validated website appearance.



A Content Management System can be broken down into four categories by function: Content Collection or Authoring, Workflow, Storage or Management, and Publishing.

A CMS system manages the flow of content from authoring to publishing by using a plan of workflow and by providing content storage and integration.

1.Collection/Authoring

The content collection process consists of adding new components to the existing repository.

The collection system includes the tools, procedures and staff that are employed to gather content, and provide editorial and metadata processing.

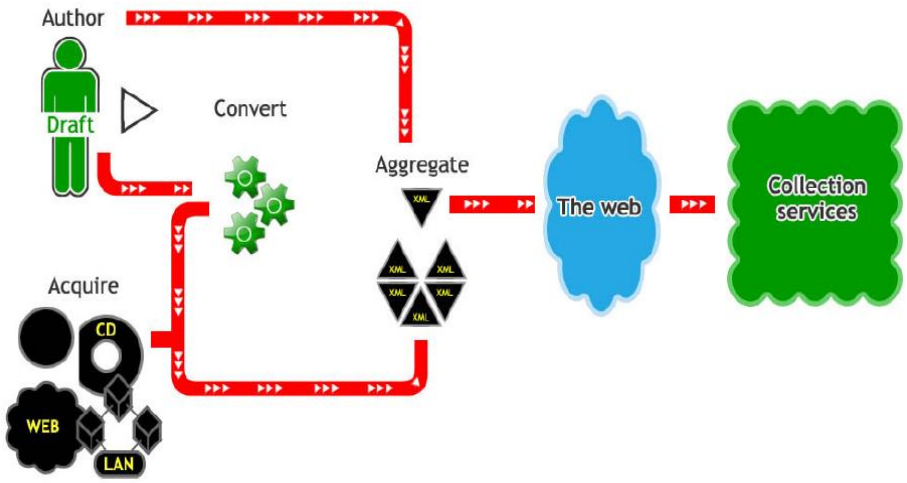
Content collection can be divided into these categories:

Authoring

Aggregation

Conversion

Collection/Authoring

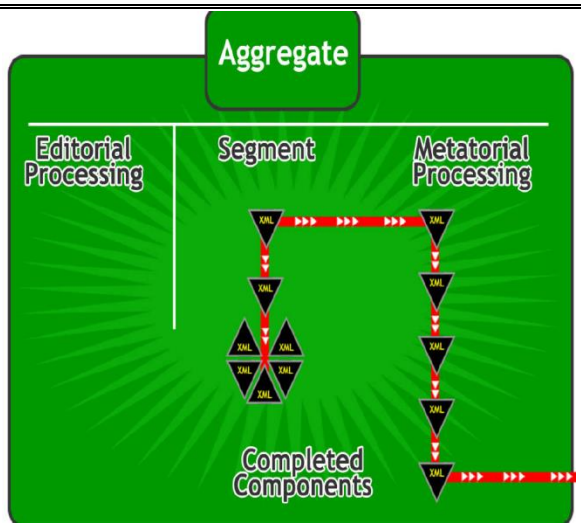


Authoring

This is the process of creating content from scratch. Authors almost always work within an editorial framework that allows them to fit their content into the structures of a target publication.

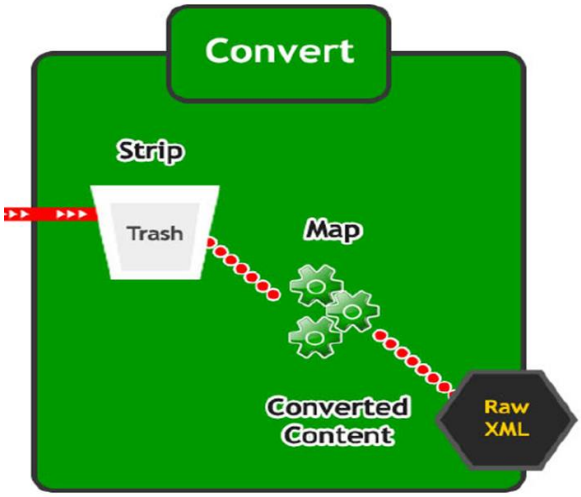
Aggregation

- This is the process of gathering pre-existing content together for inclusion in the system.
- Aggregation is generally a process of format conversion followed by intensive editorial processing and meta-tagging.
- The conversion changes the formatting of the content, while the editorial processing serves to segment and tag *the content for inclusion in the repository.*



Conversion

- This is the process of changing the metadata structure of the content
- During this process, the structural and the format-related codes must be both handled.



II Management

Repositories have the following functions:

• Storing content;

• Selecting content;

• Managing content;

• Connecting to other systems.

The management system is the repository housing all the content and the metadata information, as well as the one providing the processes and the tools needed to access and manage the collected content and metadata information.

III Workflow

- The workflow system includes the **tools and the procedures** that assure that the entire process of collection, storage and publication runs effectively, efficiently, and according to well-defined timelines and actions.
- A **workflow system supports the creation and management of business processes**.
- To be successful, the workflow system should:

• Extend over the entire process. Every step of the process from authoring through to the final deployment of each publication should be modeled and tracked within the same system.

• Represent all of the significant parts of the process including:

o Staff members.

o Standard processes.

o Standard tools and their functions.

IV Publishing

Content publishing is the process through which content is drawn out of the repository and formatted into websites, web services and other publications.

The publishing system must include:

• Publication templates;

• A full programming language;

• Runtime dependency resolution;

• File and directory creation.